CS294-40 Learning for Robotics and Control Lecture 6 - 9/16/2008

Linear Quadratic Regulators

Lecturer: Pieter Abbeel *Scribe: Ankur Mehta*

1 Lecture outline

- LQR
- LQR Extensions
- Coming up...

2 LQR (Finite Horizon Value Iteration Case Study)

We assume a linear dynamical system:

$$x_{t+1} = A_t x_t + B_t u_t, \tag{1}$$

where $x(t)$ denotes the state at time t and $u(t)$ denotes the intput at time t.

We assume a quadratic cost function:

$$J = \sum_{t=0}^{H-1} (x_t^T Q_t x_t + u_t^T R_t u_t) + x_H^T P_H x_H. \tag{2}$$

with for all t $Q_t \succ 0, R_t \succ 0$. (For a square matrix X we have $X \succ 0$ if and only if for all vectors z we have $z^\top X z > 0$.)

Our goal is to find the input sequence $\{u_0, u_1, ..., u_{H-1}\}$ that minimize the cost, i.e.,

$$\begin{aligned}
& \min_{u_0 \ldots u_{H-1}} J \\
= & \min_{u_0 \ldots u_{H-1}} \sum_{t=0}^{H-1} (x_t^T Q_t x_t + u_t^T R_t u_t) + x_H^T P_H x_H \\
= & \min_{u_0 \ldots u_{H-2}} \sum_{t=0}^{H-2} (x_t^T Q_t x_t + u_t^T R_t u_t) + x_{H-1}^T Q_{H-1} x_{H-1} + \min_{u_{H-1}} u_{H-1}^T R_{H-1} u_{H-1} + x_H^T Q_H x_H.
\end{aligned} \tag{3}$$

We first consider the minimization over u_{H-1}. From Eqn. (1) we have:

$$u_{H-1}^T R_{H-1} u_{H-1} + x_H^T Q_H x_H = u_{H-1}^T R_{H-1} u_{H-1} + (A_{H-1} x_{H-1} + B_{H-1} u_{H-1})^T Q_H (A_{H-1} x_{H-1} + B_{H-1} u_{H-1}) \tag{4}$$

This expression is a convex quadratic in u_{H-1}, and we can find the minimum by setting the gradient equal to zero:

$$\nabla_{u_{H-1}}(\cdot) = 0 = 2R_{H-1} u_{H-1} + 2B_{H-1}^T Q_H A_{H-1} x_{H-1} + 2B_{H-1}^T Q_H B_{H-1} u_{H-1},$$

This gives us the following expression for u_{H-1}:

$$\begin{aligned}
u_{H-1} &= -(R_{H-1} + B_{H-1}^T Q_H B_{H-1})^{-1} B_{H-1}^T Q_H A_{H-1} x_{H-1} && (5) \\
&= K_{H-1} x_{H-1}, && (6)
\end{aligned}$$

for

$$K_{H-1} = -(R_{H-1} + B_{H-1}^T Q_H B_{H-1})^{-1} B_{H-1}^T Q_H A_{H-1}. \tag{7}$$

Plugging this back into Eqn (3) we obtain:

$$
\begin{aligned}
& \min_{u_0 \dots u_{H-1}} J \\
= & \min_{u_0 \dots u_{H-1}} \sum_{t=0}^{H-1} (x_t^T Q_t x_t + u_t^T R_t u_t) + x_H^T P_H x_H \qquad (8) \\
= & \min_{u_0 \dots u_{H-2}} \sum_{t=0}^{H-2} (x_t^T Q_t x_t + u_t^T R_t u_t) + x_{H-1}^T P_{H-1} x_{H-1}, \qquad (9)
\end{aligned}
$$

for

$$P_{H-1} = Q_{H-1} + K_{H-1}^T R_{H-1} K_{H-1} + (A_{H-1} + B_{H-1} K_{H-1})^T Q_H (A_{H-1} + B_{H-1} K_{H-1}). \qquad (10)$$

We see that Eqn. (9) is exactly of the same format as the original problem, which was written out again in Eqn. (8). Hence, we can repeat this procedure for each time step.

This gives us the following dynamic programming algorithm to find the optimal controllers for a linear system with quadratic costs:

for $t = H - 1$ to 0
  $K_t \leftarrow -(R_t + B_t^T P_{t+1} B_t)^{-1} B_t^T P_{t+1} A_t$
  $P_t \leftarrow Q_t + K_t^T R_t K_t + (A_t + B_t K_t)^T P_{t+1} (A_t + B_t K_t)$
next t

For all times t we can compute the optimal inputs as follows: $u_t = K_t x_t$. The total cost (under the optimal policy) starting in the state x at time 0 is given by $J(x) = x^T P_0 x$. More generally, the optimal cost-to-go (=cost incurred in all future steps) for being in state x_t at time t is given by $x_t^\top P_t x_t$.

3 Extensions to LQR

For the extensions below, we could redo the steps above in each special case to re-derive a slight modified version of the above dynamic programming algorithm. However, in this lecture, we adjust the state and/or input representation to mold the dynamical system and cost into the standard form for LQR, which we have covered so far.

Concretely, in the examples below, we will define a new state representation $\bar{x}$ and a new input representation $\bar{u}$ to attain the following standard LQR form:

$$
\begin{aligned}
\bar{x}_{t+1} &= \bar{A}_t \bar{x}_t + \bar{B}_t \bar{u}_t \qquad (11) \\
J &= \sum_{t=0}^{H-1} (\bar{x}_t^T \bar{Q}_t \bar{x}_t + \bar{u}_t^T \bar{R}_t \bar{u}_t) + \bar{x}_H^T \bar{P}_H \bar{x}_H. \qquad (12)
\end{aligned}
$$

3.1 Affine system

Possibly from the output of a linearization of the dynamics and a quadratization of the cost function, we have:

$$
\begin{aligned}
x_{t+1} &= A_t x_t + B_t u_t + a_t \qquad (13) \\
J &= \sum x_t^T Q_t x_t + 2 q_t^T x_t + u_t^T R_t u_t + 2 r_t^T u_t \qquad (14)
\end{aligned}
$$

We can set:

$$\bar{A}_t = \begin{pmatrix} A_t & a_t - B_t u_t^* \\ 0 & 1 \end{pmatrix} \quad (15)$$

$$\bar{x}_t = \begin{pmatrix} x_t \\ 1 \end{pmatrix} \quad (16)$$

$$\bar{B}_t = \begin{pmatrix} B_t \\ 0 \end{pmatrix} \quad (17)$$

$$\bar{Q}_t = \begin{pmatrix} Q_t & q_t \\ q_t & c \end{pmatrix} \quad (18)$$

$$\bar{u}_t = u_t - u_t^*, \quad (19)$$

where c is any constant s.t. $\bar{Q}_t > 0$ and $u_t^* = -R_t^{-1} r_t$.

3.2 Penalize input change

Sometimes we want a penalty for changing the input u_t. We can set:

$$\bar{A}_t = \begin{pmatrix} A_t & B_t \\ 0 & 1 \end{pmatrix} \quad (20)$$

$$\bar{x}_t = \begin{pmatrix} x_t \\ u_{t-1} \end{pmatrix} \quad (21)$$

$$\bar{B}_t = \begin{pmatrix} B_t \\ 1 \end{pmatrix} \quad (22)$$

$$\bar{u}_t = u_t - u_{t-1}. \quad (23)$$

3.3 Follow a trajectory

If we already know a state trajectory x^* and input u^* that we would like to follow, we can penalize the deviation from the trajectory:

$$\bar{x}_t = x_t - x_t^* \quad (24)$$

$$\bar{u}_t = u_t - u_t^*, \quad (25)$$

and set the cost $J = \sum \bar{x}_t^T Q_t \bar{x}_t + \bar{u}_t^T R_t \bar{u}_t$.

4 Coming up...

1. Add noise: $x_{t+1} = A_t x_t + B_t u_t + w_t$
2. Unknown state, observation z_t instead (Separation principle!)
3. More general $x_{t+1} = f(x_t, u_t); \text{cost } = g(x_t, u_t)$
4. Receeding horizon DDP