16-745 Optimal Control Lecture 8

Reid Graves

February 7, 2025

1 Last Time

- Deterministic Optimal Control
- Pontryagin
- Indirect Shooting

2 Today

- LQR Problem
- LQR as a QP
- Riccati Recursion

3 LQR Problem

$$
\begin{aligned}
\min_{x_{1:N},u_{1:N-1}} \quad & J = \sum_{k=1}^{N-1} \frac{1}{2}x_k^T Q_k x_k + \frac{1}{2}u_k^T R u_k + \frac{1}{2}x_N^T Q_N x_N \\
s.t. \quad & x_{k+1} = A_k x_k + B_k u_k \\
& Q_k \geq 0, \quad R > 0
\end{aligned}
$$

R needs to be greater than 0 for problem to be well posed.

3.1 Example

- "Double Integrator"

$$\dot{x} = \begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

- Think of this as a brick sliding on ice (no friction)

$$
\begin{aligned}
x_{k+1} &= \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \begin{bmatrix} q_k \\ \dot{q}_k \end{bmatrix} + \begin{bmatrix} \frac{1}{2}h^2 \\ h \end{bmatrix} u_k \\
x_{k+1} &= Ax_k + Bu_k
\end{aligned}
$$

Where h is the timestep

4 LQR as a QP

- Assume x_1 (initial state) is given (not a decision variable)
- Define

$$z = \begin{bmatrix} u_1 \\ x_2 \\ u_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix}$$

$$H = \begin{bmatrix} R_1 & & & & \\ & Q_2 & & 0 & \\ & & R_2 & & \\ & 0 & & \ddots & \\ & & & & Q_N \end{bmatrix}$$

such that $J = \frac{1}{2} Z^T H Z$

- Define C and d:

$$\begin{bmatrix} B_1 & -I & 0 & \dots & \dots & & 0 \\ 0 & A_2 & B_2 & -I & 0 & \dots & 0 \\ & & & \ddots & & & \\ & & & & A_{N-1} & B_{N-1} & -I \end{bmatrix} \begin{bmatrix} u_1 \\ x_2 \\ u_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} -Ax_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$\Rightarrow Cz = d$

- Now we can write LQR as a standard QP:

$$\min_z \quad \frac{1}{2} z^T H z$$
$$s.t. \quad Cz = d$$

- The Lagrangian of this QP is:

$$L(z, \lambda) = \frac{1}{2} z^T H z + \lambda^T [Cz - d]$$

- KKT conditions:

$$\nabla_z L = Hz + C^T \lambda = 0$$
$$\nabla_\lambda L = Cz - d = 0$$
$$\Rightarrow \begin{bmatrix} H & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} z \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ d \end{bmatrix}$$

- We get the exact solution by solving one linear system!

4.1 Example

- Much better than shooting!

4.2 A closer look at the LQR QP

- The KKT system for LQR is very sparse (lots of zeros in matrix) and has lots of structure:

$$\begin{bmatrix} R & & & & & & B^T & & \\ & Q & & & & & I & A^T & \\ & & R & & & & & B^T & \\ & & & Q & & & & -I & A^T \\ & & & & R & & & & B^T \\ & & & & & Q_N & & & -I \\ B & -I & & & & & & & \\ & A & B & -I & & & & & 0 \\ & & & A & B & -I & & & \end{bmatrix} \begin{bmatrix} u_1 \\ x_2 \\ u_2 \\ x_3 \\ u_3 \\ x_4 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} = \begin{bmatrix} \\ 0 \\ \\ \\ \\ \\ -Ax \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{aligned} & Q_N x_4 - \lambda_4 = 0 \Rightarrow \lambda_4 = Q_N x_4 \\ & Ru_3 + B^T \lambda_4 = Ru_3 + B^T Q_N x_4 = 0 \text{ Plug in dynamics for } x_4 \\ & \Rightarrow Ru_3 + B^T Q_N (Ax_3 + Bu_3) = 0 \\ & \Rightarrow u_3 = -(R + B^T Q_N B)^{-1} B^T Q_N A x_3 \text{ call big matrix before } x_3 \; K_3 \\ & Qx_3 - \lambda_3 + A^T \lambda^4 = 0 \text{ Plug in } \lambda_4 \\ & Qx_3 - \lambda_3 + A^T Q_N x_4 = 0 \text{ Plug in dynamics} \\ & Qx_3 - \lambda_3 + A^T Q_N (Ax_3 + Bu_3) \text{ plug in } u_3 = -K_3 x_3 \\ & \Rightarrow \lambda_3 = (Q + A^T Q_N (A - BK_3)) x_3 \text{ Call big matrix before } x_3 \; P_3 \end{aligned}$$

- Now we have a recursion for K and P:

$$\begin{aligned} P_N &= Q_N \\ K_k &= (R + B^T P_{k+1} B)^{-1} B^T P_{n+1} A \\ P_k &= Q + A^T P_{n+1} (A - BK_k) \end{aligned}$$

- This is called the Riccati equation/recursion
- We can solve the QP by doing a backward Riccati pass followed by a forward rollout to compute $x_{1:N}$ and $u_{1:N-1}$
- General (dense) QP has complexity $O([N(n+m)]^3)$, Horizon N, state dimension n, control dim m
- Riccati solution is $O(N(n+m)^3)$

4.3 Example

- Riccati exactly matches QP
- Feedback policy lets us change x_0 and reject noise/disturbances

4.4 Infinite Horizon LQR

- For time invariant LQR converge to constants
- For stabilization problems we usually use constant K
- Backward recursion for P:

$$
\begin{aligned}
k_k &= (R + B^T P_{n+1} B)^{-1} B^T P_{n+1} A \\
P_k &= Q + A^T P_{k+1} (A + BK_k)
\end{aligned}
$$

- Infinite horizon limit $\Rightarrow P_{k+1} = P_k = R_{\text{inf}} \Rightarrow$