ECE 602: LUMPED LINEAR SYSTEMS

Professor Jianghai Hu

Linear Quadratic Regulation: Problem Formulation

Linear Quadratic Regulation Problem

A discrete-time LTI system

$$x[k+1] = Ax[k] + Bu[k], \quad x[0] = x_0$$

LQR Problem: Given a time horizon $k \in \{0, 1, \ldots, N\}$, find the optimal input sequence $U = \{u[0], \ldots, u[N-1]\}$ that minimizes the cost function

$$J(U) = \sum_{k=0}^{N-1} \underbrace{\left(x[k]^T Q x[k] + u[k]^T R u[k]\right)}_{\text{running cost}} + \underbrace{x[N]^T Q_f x[N]}_{\text{terminal cost}}.$$

- State weight matrix $Q = Q^T \succeq 0$ (could by time-varying)
- Control weight matrix $R = R^T \succ 0$ (could by time-varying)
- Final state weight matrix $Q_f = Q_f^T \succeq 0$
- Time horizon N (could be infinity: $N = \infty$)
- Easily generalized to time-varying $A[k]$, $B[k]$, $Q[k]$, $R[k]$ case

Example: Energy Efficient Stabilization

Starting from $x[0] = x_0$, find control U that minimizes the cost

$$J(U) = \alpha \cdot \underbrace{\sum_{k=0}^{N-1} \|u[k]\|^2}_{\text{control energy}} + \beta \cdot \underbrace{\sum_{k=0}^{N} \|x[k]\|^2}_{\text{state deviation from 0}}$$

- LQR problem with $Q = Q_f = \beta I$, $R = \alpha I$
- Simultaneously keep the state close to zero and use less control energy
- Weights $\alpha \geq 0$ and $\beta \geq 0$ determine the emphasis
- Output regulation: replace $x[k]$ in the cost function with $y[k] = Cx[k]$:

$$Q = Q_f = \beta C^T C, \quad R = \alpha I$$

Example: Minimum Energy Steering

Starting from $x[0] = x_0$, find control U with the least energy

$$J(U) = \sum_{k=0}^{N-1} \|u[k]\|^2$$

that can steer the state to its final value $x[N] = 0$

- (Approximate) LQR formulation: $Q = 0$, $R = I$, $Q_f = \rho I$ for ρ very large

Example: Optimal Tracking

Find efficient control U for the state to track a given trajectory

$$J(U) = \alpha \cdot \underbrace{\sum_{k=0}^{N-1} \|u[k]\|^2}_{\text{control energy}} + \beta \cdot \underbrace{\sum_{k=0}^{N} \|x[k] - x_k^*\|^2}_{\text{tracking error penalty}}$$

- $x_0^*, x_1^*, \ldots, x_N^*$ is the reference trajecotry to be tracked

Formulated as a (time-varying) LQR problem

- Augment the state x to $\tilde{x} = \begin{bmatrix} x \\ z \end{bmatrix}$ with $z \in \mathbb{R}$; let $\tilde{x}[0] = \begin{bmatrix} x_0 \\ 1 \end{bmatrix}$
- Augmented state dynamics: $\tilde{x}[k+1] = \begin{bmatrix} A & 0 \\ 0 & 1 \end{bmatrix} \tilde{x}[k] + \begin{bmatrix} B \\ 0 \end{bmatrix} u[k]$
- Choose $\tilde{Q}_k = \beta \begin{bmatrix} I & -x_k^* \\ -(x_k^*)^T & \|x_k^*\|^2 \end{bmatrix}$, $\tilde{R} = \alpha I$, $\tilde{Q}_f = \tilde{Q}_N$

Practical Applications

- Control of (ground,aerial,under-water,space) vehicles
- Robotics (robotic arms, mobile robots)
- Process control (chemical, biological, etc.)
- Energy and power systems
- Transportation systems
- ...

Direct Approach

LQR problem as a least square problem:

$$\text{minimize} \quad J(\mathbf{u}) = \mathbf{x}^T \underbrace{\begin{bmatrix} Q & & & \\ & Q & & \\ & & \ddots & \\ & & & Q_f \end{bmatrix}}_{\mathbf{Q}} \mathbf{x} + \mathbf{u}^T \underbrace{\begin{bmatrix} R & & & \\ & R & & \\ & & \ddots & \\ & & & R \end{bmatrix}}_{\mathbf{R}} \mathbf{u}$$

subject to the constraint:

$$\underbrace{\begin{bmatrix} x[1] \\ x[2] \\ \vdots \\ x[N] \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} B & 0 & \cdots & \\ AB & B & 0 & \cdots \\ \vdots & \vdots & \ddots & \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}}_{\mathbf{G}} \underbrace{\begin{bmatrix} u[0] \\ u[1] \\ \vdots \\ u[N-1] \end{bmatrix}}_{\mathbf{u}} + \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}}_{\mathbf{H}} x_0$$

Solution: $\mathbf{u}^* = -(\mathbf{R} + \mathbf{G}^T\mathbf{Q}\mathbf{G})^{-1}\mathbf{G}^T\mathbf{Q}\mathbf{H}x_0$

Limitations of Direct Approach

- Matrix inversion needed to find optimal control
- Problem (matrices) dimension increases with time horizon N
- Impractical for large N let alone infinite horizon case
- Sensitivity of solutions to numerical errors

Observations:

- Problem easier to solve for shorter time horizon N
- $(N+1)$-horizon solution related to N-horizon solution

Dynamic programming approach

- Reuse results for smaller N to solve for larger N case
- In each iteration only need to deal with a problem of fixed size