Linear Quadratic Regulator(LQR) for Discrete-Time System

Baboo J. Cui

June 10, 2019

Contents

LQR is related to optimal control problem, many problems can be formulated into it. It's one of the fundamental way to achieve optimal control.

1 Problem Formulation

Given a discrete LTI system:

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

given a time horizon $k \in \{0, 1, \ldots, N\}$, where N may be infinity, find the optimal input sequence $U = \{u[0], u[1], \ldots, u[N-1]\}$ that minimize the **cost function**:

$$J(U) = \sum_{k=0}^{N-1} \left(x^T[k]Qx[k] + u^T[k]Ru[k] \right) + x^T[N]Q_f x[N]$$

- **state weight matrix**: $Q = Q^T \succeq 0$
- **control weight matrix**: $R = R^T \succ 0$, indicate that there is no free control input
- **final state weight matrix**: $Q_f = Q_f^T \succeq 0$
- **running cost**: the value of the first term in $J(u)$
- **terminal cost**: the value of the second term in $J(u)$
- **infinite case**: N is infinity, in this case, $Q_f = 0$

Note that all these case can be generalized into time-varying cases.

2 Examples of Implementations

Many problem can be formulated into LQR form, and here are some examples, though they look differently.

2.1 Energy Efficient Stabilization

Starting from $x[0] = x_0$, find control sequence U that minimize

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

to make it into LQR form, choose:

- $Q = \beta I$
- $R = \alpha I$
- $Q_f = \beta I$

Note that:

- cost function try to make state trajectory stay close to zero and use the least control energy simultaneously
- α and β determine the emphasis

Sometime state cannot be obtained directly, in this case, system output y can be used for evaluating running cost. Suppose output equation(Du part can be eliminate) is

$$y = Cx$$

in this case choose $Q = \beta C^T C$. Here is the proof:

$$\begin{aligned}
\beta \sum_{k=0}^{N} ||y[k]||^2 &= \sum_{k=0}^{N} y^T[k]\beta I y[k] \\
&= \sum_{k=0}^{N} (Cx[k])^T \beta I C x[k] \\
&= \sum_{k=0}^{N} x^T[k] C^T \beta I C x[k] = \sum_{k=0}^{N} x^T[k](\beta C^T C)x[k]
\end{aligned}$$

this is a very import conclusion.

2.2 Minimum Energy Steering

Starting from $x[0] = x_0$, find control sequence U to use least energy to steer the final state to $x[N] = 0$ without lost generosity, the cost is:

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

to make it into LQR form, choose:

- $Q = 0$
- $R = I$
- $Q_f = \infty I$

By setting $Q_f \to \infty I$, there is a big penalty if $X[N]$ is far from 0, note that this won't lead to a analytic solution, but the **approximation** is good enough.

2.3 LQR for Tracking(VIP TOPIC)

Find efficient sequence U for the state to track a given **reference trajectory** x_k^*(may be time-varying):

$$J(U) = \alpha \sum_{k=0}^{N-1} ||u[k]||^2 + \beta \sum_{k=0}^{N} ||x[k] - x_k^*||^2$$

note that $||x[k]-x_k^*||^2$ is not homogeneous quadratic, it should be formulate. It can be expanded(refer math proof in last part):

$$\begin{aligned} ||x[k]-x_k^*||^2 &= x^T[k]x[k] - 2x^T[k]x_k^* + (x_k^*)^T x_k^* \\ &= \begin{bmatrix} x[k] & 1 \end{bmatrix} \begin{bmatrix} I & x_k^* \\ (x_k^*)^T & (x_k^*)^T x_k^* \end{bmatrix} \begin{bmatrix} x[k] \\ 1 \end{bmatrix} \quad \text{dimension augmentation} \end{aligned}$$

construct new state variable $\tilde{x}[k] = [x[k] \quad 1]^T$, new system dynamic will be:

$$\tilde{x}[k+1] = \begin{bmatrix} A & 0 \\ 0 & 1 \end{bmatrix} \tilde{x}[k] + \begin{bmatrix} B \\ 0 \end{bmatrix} u[k]$$

and the origin cost can be reformed as:

$$J(U) = \alpha \sum_{k=0}^{N-1} ||u[k]||^2 + \beta \sum_{k=0}^{N} \tilde{x}^T[k]\tilde{Q}_k\tilde{x}[k]$$

where

$$\tilde{Q}_k = \begin{bmatrix} I & x_k^* \\ (x_k^*)^T & (x_k^*)^T x_k^* \end{bmatrix}$$

clearly, the system is LTI and the cost function is LTV.

2.4 LQR for System with Perturbation

Suppose system is:

$$x[k+1] = Ax[k] + Bu[k] + w[k]$$

To achieve LQR formulation, new state vector is constructed as:

$$\tilde{x}[k] = [x[k] \quad z[k]] \quad \text{dimension augmentation}$$

recall that $x \in \mathbb{R}^n$, and $z[k] \in \mathbb{R}$, set $z[k] = z[k+1] = 1$, new system dynamic will be:

$$\tilde{x}[k+1] = \begin{bmatrix} A & w[k] \\ 0 & 1 \end{bmatrix} \tilde{x}[k] + \begin{bmatrix} B \\ 0 \end{bmatrix} u[k]$$

and system initial condition is $\tilde{x}[0] = [x[0] \quad 1]$. R will be the original one and $\tilde{Q}$ is:

$$\tilde{Q}_k = \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix}$$

clearly, the system is LTV and the cost function is LTI. In this case, u is not changed, x is augmented.

3 Direct Approach to Solve LQR

LQR problem can be directly formulated as a least square problem

4 Extra

4.1 Matlab Functions

- lqrd(): for discrete-time system
- lqr(): for continuous-time system

4.2 Quadratic Expansion

The general length of a vector $x \in \mathbb{R}^n$ is also called the L_2 norm. It is defined as:

$$||x||^2 = x^T x = \sum_{i=1}^{n} x_i^2, \text{ where } x_i \in \mathbb{R}$$

if another vector $y \in \mathbb{R}^n$, the norm of the difference is:

$$\begin{aligned} ||x-y||^2 &= ||y-x||^2 \quad \text{identity property} \\ &= (x-y)^T(x-y) \quad \text{definition} \\ &= x^T x - x^T y - y^T x + y^T y \quad \text{distributive property} \\ &= ||x||^2 - 2x^T y + ||y||^2 \end{aligned}$$

recall that:

$$x^T y = y^T x \quad \text{property of inner product}$$

4.3 Direct Approach

Formulate the LQR problem as a least square problem: Under the constraint:

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

Minimize the function:

$$X^T \underbrace{\begin{bmatrix} Q & & & \\ & Q & & \\ & & \ddots & \\ & & & Q_f \end{bmatrix}}_{\widetilde{Q}} X + U^T \underbrace{\begin{bmatrix} R & & & \\ & R & & \\ & & \ddots & \\ & & & R \end{bmatrix}}_{\widetilde{R}} U$$

4.4 Limitations of Direct Approach

- Matrix inversion needed to find optimal control
- Problem(matrices) dimension increases with time horizon N
- Impraricalfor 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
- Explpoit this relation to design an iterative solution procedure

Dynamic programming approach

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

4.5 Movitating Example

- Start from point A
- Try to reach point B
- Each step only move right
- Cost labeled on each edge

Problem: The least costly path from A to B?

4.6 Formulated as an Optimal Control Problem

- $A = (0,0),\ B = (3,3)$
- State $x[k]$ with
$$x[0] = A,\ x[6] = B$$
- Control $u[k] = \pm 1$
- Dynamics:
$$x[k+1] = \begin{cases} x[k] + (0,1) & u[k] = 1 \\ x[k] + (1,0) & u[k] = -1 \end{cases}$$
- Cost to be minimized:
$$\sum_{k=0}^{5} \underbrace{w(x[k], u[k])}_{\text{edge weight}}$$

4.7 Direct Solution

Enumerate all possible legal from A to B and compare their costs to find the one with the least cost.

- A total of 20 possible paths

  For ℓ-by-ℓ grid, the total number of legal paths is
$$\frac{(2\ell)!}{(\ell!)^2}$$
- Grows extremely fast as problem size ℓ increases
- Solution impractical for large ℓ

4.8 Value Function

Definition: At any point z, the value function(optimal cast-to-go) $V(z)$ is the least possible cost to reach B from z.

- Obtained by solve shorter time horizon problems

Original problem is to find $V(A)$

4.9 Value Function Property

Principle of Optimality: If a least-cost path from A to B is

$$x_0^* = A \to x_1^* \to x_2^* \to \cdots \to x_6^* = B,$$

Then any truncation of it at the end:

$$x_t^* \to x_{t+1}^* \to \cdots \to x_6^* = B$$

is also a least-cost path from x_t^* to B.
As a result, value function at any point z satisfies

$$\begin{aligned} V(z) &= \min\left\{w_u + V\left(z_u'\right), w_d + V\left(z_d'\right)\right\} \\ &= \min_{u \in \pm 1}\left[w(z, u) + V\left(z'\right)\right] \end{aligned}$$

- $V(z)$: Cost-to-go from current position
- $w(z, u)$: Running cost of current step
- $V(z')$: Cost-to-go from next state position

4.10 Value Function Iteration

Idea: Use above to iteratively evaluate $V(z)$ from right to left

4.11 Value Function Iteration

Idea: Use above to iteratively evaluate $V(z)$ from right to left

4.12 Value Function Iteration

Conclusion: The least cost from A to B is 40

4.13 Recover the Optimal Control

Optimal control $u[0]$ is recovered from $V(A) = \min\{5 + 35,\ 7 + 36\}$

4.14 Advantages of Dynamic Programming

Reduced computational complexity: for ℓ-by-ℓ grid

- Only need to compute ℓ^2 value functions
- No need to enumerate $\frac{(2\ell)!}{(\ell!)^2}$ paths
- Solve an optimization problem of fixed size in each iteration

Provide solutions to a family of optimal control problems

- Even if starting from a different initial position (e.g. due to perturbation), there is no need for re-computation

4.15 Back to LQR Problem

A discrete-time LTI system

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

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

$$J(U) = \sum_{k=0}^{N-1} \underbrace{(x[k]^T Qx[k] + u[k]^T Ru[k])}_{running\ cost} + \underbrace{x[N]^T Q_f x[N]}_{terminal cost}$$

Quenstions: Can we apply dynamic programming method to LQR problem?

4.16 Value Function of LQR Problem

The value function at time $t \in \{0, 1, ..., N\}$ and state $x \in \mathbb{R}^n$ is

$$V_t(x) = \min_{u[t],...,u[N-1]} \sum_{k=t}^{N-1} (x[k]^T Qx[k] + u[k]^T Ru[k]) + x[N]^T Q_f x[N]$$

with the initial condtion $x[t] = x$

- **Cost-to-go**, namely, optimal cost of the LQR problem over the time horizon $\{t, t+1, ..., N\}$, starting from $x[t] = x$.

4.17 Solution of LQR Problem via Value Functions

Preview of results:

- The value function at the final time is quadratic: $V_N(x) = x^T Q_f x$
- We will see that the value function at any time t is also quadratic: $V_t(x) = x^T P_t x$ for some $P_t \geqslant 0$
- P_t can be obtained from P_{t+1}

Solution algorithm:

(1) Start from $P_N = Q_f$ at time $t = N$

(2) For $t = N - 1 : 0$ do

- Compute P_t from P_{t+1} by the above recursion

(3) Recover optimal control sequence from value functions

4.18 How are Value Functions Related?

(Hamilton-Jacobi-)Bellman equation:

$$\begin{aligned} V_t(x) &= \min_{u[t]=v} [x^T Q x + v^T R v + V_{t+1}(Ax + Bv)] \\ &= x^T Q x + \min_{u[t]=v} [v^T R v + V_{t+1}(Ax + Bv)] \end{aligned}$$

Optimality principle: For optimal case, cost-to-go form next state $x[t+1]$ should also be optimal, i.e., $V_{t+1}(x[t+1])$.

4.19 $t = N$ case

Value function at time N is quadratic:

$$V_N(x) = x^T P_N x, \quad \forall x \in \mathbb{R}^n, \quad \text{where } P_N = Q_f$$

4.20 $t = N - 1$ case

Value function at time $N - 1$ is:

$$\begin{aligned} V_{N-1}(x) &= x^T Q x + \min_v [v^T R v + V_N(Ax + Bv)] \\ &= x^T Q x + \min_v [v^T R v + (Ax + Bv)^T P_N (Ax + Bv)] \end{aligned}$$