Linear Quadratic Control

deterministic case and stochastic case

2024.04.02
모바일로보틱스팀, 로보틱스랩
안재성 연구원

Basic Problem Setup

- **Linear Deterministic System**

$$x(k+1) = Ax(k) + Bu(k)$$

$$y(k) = Cx(k)$$

$\longrightarrow$ **deterministic system이므로** $w(k), v(k)$ **와 같은 term 사용 x**

유도한 결과의 특성을 파악하기 위해,
간단하게 linear time-invariant system 시스템으로 가정하자. (즉, A, B, C가 상수)

For a linear state feedback controller

$$u(k) = -L(k)x(k)$$

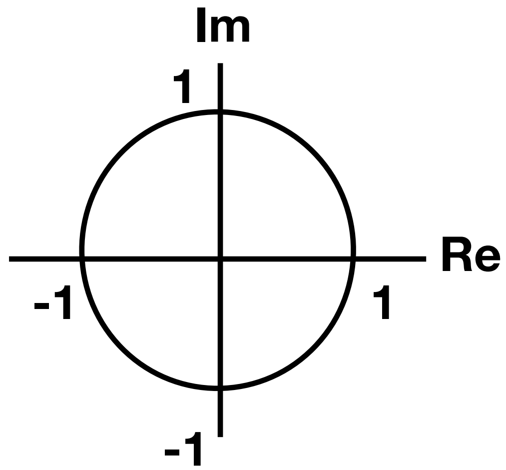
$\downarrow$

Output feedback이 아닌 state feedback인 것에 유의

The closed-loop response is

$$x(k+1) = (A - BL(k))x(k)$$

위 State feedback controller 가 system을 stabilize하기 위해서는 $A - BL(k)$ **의 모든 고유값이 unit disk 안에 있어야 한다.**



Objective of LQ

- **A system visits a sequence of states of** $x(0), x(1), \ldots, x(p)$

  and desired sequence of states $x(0), \bar{x}(1), \ldots \bar{x}(p)$

임의의 set-point가 아닌 0으로 해놓고 생각해보자
0으로 가정해도 문제가 없음 왜냐하면
0으로 해도 문제의 property, solution이 바뀌지 않고 나중에 평행 이동하면 된다

- **Objective function**

$$min \sum_{k=0}^{p-1} [x^T(k)Qx(k) + u^T(k)Ru(k)] + x^T(p)Q_t x(p)$$

Q, R **: symmetric positive definite**
Q_t **: positive semi-definite**

이걸 최소화하는 u를 구하자! -> x, u 모두 0에 가까워져야함
Output error와 input value의 non-zero가 경쟁하지 않음

반면 MPC 같은 경우엔 $min \sum_{k=0}^{p-1} [\underbrace{x^T(k)Qx(k)}_{r-y} + \underbrace{u^T(k)Ru(k)}_{\Delta u^T}] + x^T(p)Q_t x(p)$ **경쟁해서 integral action이 있다고함.**
만약 u를 delta u 안쓰면 off-set free가 아님(= set-point에 도달 불가)

Open-Loop Control vs. Feedback Control

- **Optimal open-loop control problem**

시간 0, 1, 2 지나면서 나오는 system으로부터의 information(output, input이든)을 못얻어서 활용 안하는 것
ex) 전자레인지

$u(0), u(1), ..., u(k)$ **를** $x(0)$ **갖고서만**

- **Optimal feedback control problem**

input을 어떤 information 함수로 쓰겠다.

Find the optimal feedback law $u(k) = f(x(k))$ **Or**

$$u(k) = f(y(k)), y(k-1), ...)$$

- **둘 중 뭐가 좋냐?**

만약 시스템이 deterministic 하다면, open-loop 결과 = feedback-loop 결과

-> Model = Plant

HYUNDAI

Least Squares Solution

- **Open-loop Optimal Feedback Control**

Using $x(k+1) = Ax(k) + Bu(k)$ **recursively gives,**

$$
\begin{aligned}
x(k+1) &= Ax(k) + Bu(k) + A(Ax(k-1) + Bu(k-1)) + Bu(k) \\
&= A^2x(k-1) + Bu(k) + ABu(k-1) \\
&= \vdots \\
&= A^{k+1}x(0) + (Bu(k) + ABu(k-1) + \ldots + A^kBu(0)
\end{aligned}
$$

Then we can write,

$$
\underbrace{\begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(p) \end{bmatrix}}_{\mathcal{X}} = \underbrace{\begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^p \end{bmatrix}}_{S^x} x(0) + \underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 \\ B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ A^{p-1}B & A^{p-2}B & \cdots & B \end{bmatrix}}_{S^u} \underbrace{\begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ \vdots \\ u(p-1) \end{bmatrix}}_{\mathcal{U}}
$$

terminal state

decision variable

Least Squares Solution

- **System equation**

$$\mathcal{X} = S^x x(0) + S^u \mathcal{U}$$

- **Quadratic cost function**

$$V_0(x(0);\mathcal{U}) = \sum_{k=0}^{p-1} [x^T(k)Qx(k) + u^T(k)Ru(k)] + x^T(p)Q_t x(p)$$
$$= \mathcal{X}^T \Gamma^x \mathcal{X} + \mathcal{U}^T \Gamma^u \mathcal{U}$$

$$\Gamma^x = blockdiag\{Q,\ldots,Q,Q_t\} \quad \Gamma^u = blockdiag\{R,\ldots,R\}$$

- **Optimal cost**

$$V_0(x(0)) = \min_{\mathcal{U}} \{x^T(0)S^{xT}\Gamma^x S^x x(0) + \mathcal{U}^T[S^{uT}\Gamma^x S^u + \Gamma^u]\mathcal{U} + 2x^T(0)S^{xT}\Gamma^x S^u \mathcal{U}\}$$

$\mathcal{U}$ 의 quadratic function

- **Optimal solution (input에 제약조건 없다면)**

$$\mathcal{U}^* = -H^{-1}g = -[S^{uT}\Gamma^x S^u + \Gamma^u]^{-1} S^{uT}\Gamma^x S^x x(0)$$

OLOFC

그러면 이제 이 optimal control action $\mathscr{U}*$ 을 다시 cost-to-go function에 넣어보자!

$$\mathscr{U}* = -[S^u\Gamma^x S^u + \Gamma^u]^{-1} S^{uT}\Gamma^x S^x x(0)$$

- **Optimal cost-to-go function**

$$V_0^*(x(0)) = x^T(0)\left[S^{xT}\Gamma^x S^x - S^{xT}\Gamma^x S^u(S^{uT}\Gamma^x S^u + \Gamma^u)^{-1}S^{uT}\Gamma^x S^x\right]x(0)$$

→ **Initial state의 quadratic function**

★ 즉, optimal cost-to-go 함수의 form을 알 수 있다.
Dynamic Programming 풀 때, optimal cost-to-go 함수의 form을 찾기 어려워
neural network를 활용하여 approximation을 쓰는 것이고 이에 반해
OLOFC는 form을 알 수 있다!

즉, 우리가 $x(0)$ 만 알면 $\mathscr{U}*(0), \mathscr{U}*(1), \mathscr{U}*(2)\ldots\mathscr{U}*(p)$ 다 알 수 있다는 뜻이다.
but, 실제에선 model과 plant가 다를 수 있어서 부담스러움

그래서 보통 Receding horizon control 방법을 활용하여 구현한다. (control action 계산엔 state feedback 사용 X)
—> 매 시간 state 정보 들어오는 것을 그나마 활용할 수 있다.

단점
1. 계산적으로 효율적이지 않다.
2. Stochastic case에서는 효과적이지 않다.

Closed-loop Optimal Feedback Control (CLOFC)

- **매 시간마다 system information을 받아서 input을 계산한다.**
- **이런 formulation을 풀 수 있는게 Dynamic Programming**

Dynamic Programming

- **What is DP?** **Multi-stage 문제를 Single-stage 문제로 쪼개서 풀자**

$$\mathcal{X} = S^x x(0) + S^u \mathcal{U}$$

At the stage p-1, Bellman's equation is

$$V_{p-1}(x(p-1)) = \min_{u(p-1)} \{x^T(p-1)Qx(p-1) + u^T(p-1)Ru(p-1) + x^T(p)S(p)x(p)\}$$

where, $S(p) = Q_t$

Noting that $x(p) = Ax(p-1) + Bu(p-1),$

We get

$$V_{p-1}(x(p-1)) = \min_{u(p-1)} \{x^T(p-1)(A^T S(p)A + Q)x(p-1) +$$

$$2x^T(p-1)A^T S(p)Bu(p-1) +$$

$$u^T(p-1)(B^T S(p)B + R)u(p-1)\}$$

Dynamic Programming

그러면 이제 이 optimal control action u^* 을 구할 수 있고 (unconstraint라고 가정하고 풀면)

$$u^*(p-1) = -\underbrace{(B^T S(p) B + R)^{-1} B^T S(p) A}_{L(p-1)} x(p-1)$$

위 식은 State feedback controller!

또 이걸 cost-to-go function에 넣으면

$$V_{p-1}(x(p-1)) = x^T(p-1) S(p-1) x(p-1)$$

$\longrightarrow$ 또 $x(p-1)$ 의 quadratic function 이다!

이것이 LQ Control Problem unconstraint 조건으로 풀었을 때, 굉장한 이점이 된다.

Where $S(p-1)$ **is given by the following Riccati Equation**

$$S(p-1) = A^T S(p) A + Q - A^T S(p) B (B^T S(p) B + R)^{-1} B^T S(p) A$$

Dynamic Programming

Stage: p - 2

stage-wise
single-stage
cost

$$
\begin{aligned}
V_{p-2}(x(p-2)) &= \min_{u(p-2)} \{x^T(p-2)Qx(p-2) + u^T(p-2)Ru(p-2) + \\
&\qquad V_{p-1}(x(p-1))\} \\
&= \min_{u(p-2)} \{x^T(p-2)Qx(p-2) + u^T(p-2)Ru(p-2) + \\
&\qquad x^T(p-1)S(p-1)x(p-1)\}
\end{aligned}
$$

cost-to-go func (optimal)

"Bellman's optimality eqn"

This equation is in the same for as (6). The optimal solution is

$$
u^*(p-2) = -\underbrace{\left(B^T S(p-1)B + R\right)^{-1} B^T S(p-1)A}_{L(p-2)}\, x(p-2)
$$

Generalization

Successively solving for cost-to-go $V_k(x(k))$ **, we get:**

$$u^*(k) = -L(k)x(k), \qquad for k = p-1,\ldots,0$$

매시간마다 optimal control action은 state에 gain을 곱한 형태가 됨.

where

$$L(k) = (B^T S(k+1)B + R)^{-1} B^T S(k+1)A$$

$$S(k) = A^T S(k+1)A + Q - \\ A^T S(k+1)B(B^T S(k+1)B + R)^{-1} B^T S(k+1)A$$

LQ 제어는 Cost-to-go function의 recursive equation이 Riccati Difference Equation이다.
Cost-to-go function이 state의 quadratic function이다.

결론

- **Deterministic case의 경우, Open loop optimal feedback control과 Closed loop optimal feedback control은 같은 결과를 낸다.**

- **The optimal p-stage cost is** $V_0(x(0)) = x^T(0)S(0)x(0)$

- **Receding horizon solution은 연산 소요가 많다**