کنترل پیش بین

Model Predictive Control

ارائه کننده: امیرحسین نیکوفرد
مهندسی برق و کامپیوتر دانشگاه خواجه نصیر

Optimal Control Introduction and Unconstrained Linear Quadratic Control

- Optimal Control
  - Introduction
  - Batch Approach
  - Recursive Approach
  - The Dynamic Programming algorithm
- Linear Quadratic Optimal Control
  - Batch Approach
  - Recursive Approach
  - Receding Horizon
  - Infinite Horizon Optimal Control

Optimal Control Introduction

- Discrete-time optimal control is concerned with choosing an optimal input sequence $U_{0\to N} \triangleq \begin{bmatrix} u_0' & u_1' & \cdots \end{bmatrix}'$ (as measured by some objective function), over a finite or infinite time horizon, in order to apply it to a system with a given initial state x(0).
- The objective, or cost function is often defined as a sum of stage costs **$q(x_k,u_k)$** and, when the horizon has finite length N, a terminal cost **$p(x_N)$**

$$J_{0\to N}(x_0, U_{0\to N}) \triangleq p(x_N) + \sum_{k=0} q(x_k, u_k)$$

- The states $\{x_k\}$ must satisfy the system dynamics

$$x_{k+1} = g(x_k, u_k),\ k = 0, \ldots, N-1$$
$$x_0 = x(0)$$

Optimal Control Introduction

and there may be state and/or input constraints

$$h(x_k, u_k) \le 0, \quad k = 0, \ldots, N-1.$$

- In the finite horizon case, there may also be a constraint that the final state x_N lies in a set χ_f

$$x_N \in \chi_f$$

- A general finite horizon optimal control formulation for discrete-time systems is therefore

$$\begin{aligned} J^*_{0\to N}(x(0)) \triangleq \min_{U_{0\to N}} \; & J_{0\to N}(x(0), U_{0\to N}) \\ \text{subject to} \quad & x_{k+1} = g(x_k, u_k),\ k = 0, \ldots, N-1 \\ & h(x_k, u_k) \le 0,\ k = 0, \ldots, N-1 \\ & x_N \in \mathcal{X}_f \\ & x_0 = x(0) \end{aligned}$$

Optimal Control Introduction

General Problem Formulation

- Consider the nonlinear time-invariant system

$$x(t+1) = g(x(t), u(t)),$$

subject to the constraints $h(x(t), u(t)) \leq 0, \forall t \geq 0$

Consider the following objective or cost function

$$J_{0\to N}(x_0, U_{0\to N}) := p(x_N) + \sum_{k=0}^{N-1} q(x_k, u_k)$$

where

- N is the time horizon,
- $x_{k+1} = g(x_k, u_k), k = 1, \ldots, N-1$ and $x_0 = x(0)$
- $q(x_k, u_k)$ and $p(x_N)$ are the stage cost and terminal cost, respectively

Optimal Control Introduction

General Problem Formulation

- Consider the Constrained Finite Time Optimal Control (CFTOC) problem.

$$
\begin{aligned}
J^*_{0\to N}(x_0) = \quad & \min_{U_{0\to N}} \quad J_{0\to N}(x_0, U_{0\to N}) \\
& \text{subj. to} \quad x_{k+1} = g(x_k, u_k),\ \ k = 0, \ldots, N-1 \\
& \qquad\qquad h(x_k, u_k) \leq 0,\ \ k = 0, \ldots, N-1 \\
& \qquad\qquad x_N \in \mathcal{X}_f \\
& \qquad\qquad x_0 = x(0)
\end{aligned}
$$

- $\chi_f \subseteq R^n$ is a terminal region,
- $\chi_{0\to N} \subseteq R^n$ to is the set of feasible initial conditions x(0)
- the optimal cost $J^*_{0\to N}(x_0)$ is called value function,
- assume that there exists **a minimum**
- denote by $U^*_{0\to N}(x_0)$ one of **the minima**

Optimal Control Introduction

Objectives

- Solution.

1. a general nonlinear programming problem (batch approach),
2. recursively by invoking Bellman's Principle of Optimality (recursive approach).

- Infinite horizon. We will investigate if

1. a solution exists as $N \rightarrow \infty$,
2. the properties of this solution.
3. approximation of the solution by using a receding horizon technique.

Batch Approach

Solution via Batch Approach. NLP formulation

Write the equality constraints from system constraints as

$$\begin{aligned} x_1 &= g(x(0), u_0) \\ x_2 &= g(x_1, u_1) \\ &\vdots \\ x_N &= g(x_{N-1}, u_{N-1}) \end{aligned}$$

then the optimal control problem is a general Non Linear Programming (NLP) problem

$$\begin{array}{rll} J^*_{0\to N}(x_0) = & \min_{U_{0\to N}} & p(x_N) + \sum_{k=0}^{N-1} q(x_k, u_k) \\ & \text{subj. to} & x_1 = g(x_0, u_0) \\ & & x_2 = g(x_1, u_1) \\ & & \vdots \\ & & x_N = g(x_{N-1}, u_{N-1}) \\ & & h(x_k, u_k) \leq 0, \;\; k = 0, \ldots, N-1 \\ & & x_N \in \mathcal{X}_f \\ & & x_0 = x(0) \end{array}$$

Recursive Approach

Solution via Recursive Approach

Principle of optimality

For a trajectory $x_0, x_1^*, \ldots, x_N^*$ to be optimal, the trajectory starting from any intermediate point x_j^*, i.e. $x_j^*, x_{j+1}^*, \ldots, x_N^*$, $0 \le j \le N-1$, must be optimal.

Define the cost from j to N

$$J_{j\to N}(x_j, u_j, u_{j+1}, \ldots, u_{N-1}) := p(x_N) + \sum_{k=j}^{N-1} q(x_k, u_k),$$

also called the j-step *cost-to-go*. Then the *optimal cost-to-go* $J^*_{j\to N}$ is

$$\begin{array}{lll} J^*_{j\to N}(x_j) := & \min_{u_j, u_{j+1}, \ldots, u_{N-1}} & J_{j\to N}(x_j, u_j, u_{j+1}, \ldots, u_{N-1}) \\ & \text{subj. to} & x_{k+1} = g(x_k, u_k),\ k = j, \ldots, N-1 \\ & & h(x_k, u_k) \le 0,\ k = j, \ldots, N-1 \\ & & x_N \in \mathcal{X}_f \end{array}$$

Note that $J^*_{j\to N}(x_j)$ depends only on the initial state x_j.

Recursive Approach

Solution via Recursive Approach

By the ***principle of optimality*** the cost $J^*_{j-1\to N}$ can be found by solving

$$\begin{array}{rll} J^*_{j-1\to N}(x_{j-1}) = & \min\limits_{u_{j-1}} & \overbrace{q(x_{j-1},u_{j-1})}^{\text{stage cost}} + \overbrace{J^*_{j\to N}(x_j)}^{\text{optimal cost-to-go}} \\ & \text{subj. to} & x_j = g(x_{j-1},u_{j-1}) \\ & & h(x_{j-1},u_{j-1}) \leq 0 \\ & & x_j \in \mathcal{X}_{j\to N}. \end{array} \quad (1)$$

Note that

- the only decision variable is u_{j-1},
- the inputs $u^*_j, \ldots, u^*_{N-1}$ have already been selected optimally to yield the optimal cost-to-go $J^*_{j\to N}(x_j)$.
- in $J^*_{j\to N}(x_j)$, the state x_j can be replaced by $g(x_{j-1},u_{j-1})$
- The set $\mathcal{X}_{j\to N}$ is the set of states x_j for which (1) is feasible. We will study these sets later in this class.

Dynamic Programming Algorithm

➢ Solution via Recursive Approach

The following (recursive) dynamic programming algorithm can be used to compute the optimal control law.

$$
\begin{aligned}
J^*_{N\to N}(x_N) &= p(x_N)\\
\mathcal{X}_{N\to N} &= \mathcal{X}_f,
\end{aligned}
$$

$$
\begin{array}{lll}
J^*_{N-1\to N}(x_{N-1}) & = \min\limits_{u_{N-1}} & q(x_{N-1},u_{N-1}) + J^*_{N\to N}(g(x_{N-1},u_{N-1}))\\
& \text{subj. to} & h(x_{N-1},u_{N-1}) \le 0,\\
& & g(x_{N-1},u_{N-1}) \in \mathcal{X}_{N\to N}\\
\vdots & & \\
J^*_{0\to N}(x_0) & = \min\limits_{u_0} & q(x_0,u_0) + J^*_{1\to N}(g(x_0,u_0))\\
& \text{subj. to} & h(x_0,u_0) \le 0,\\
& & g(x_0,u_0) \in \mathcal{X}_{1\to N}\\
& & x_0 = x(0).
\end{array}
$$

Linear Quadratic Optimal Control

Consider only linear discrete-time time-invariant systems

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

and quadratic cost functions

$$J_0(x_0, U_0) \triangleq x_N' P x_N + \sum_{k=0}^{N-1} [x_k' Q x_k + u_k' R u_k] \qquad (2)$$

are considered, and we consider only the problem of regulating the state to the origin, without state or input constraints.

The two most common solution approaches will be described here

- Batch Approach, which yields a series of numerical values for the input
- Recursive Approach, which uses Dynamic Programming to compute control policies or laws, i.e. functions that describe how the control decisions depend on the system states.

Linear Quadratic Optimal Control

Unconstrained Finite Horizon Control Problem

- **Goal:** Find a sequence of inputs $U_0 \triangleq [u_0', \dots, u_{N-1}']'$ that minimizes the objective function

$$\begin{aligned} J_0^*(x(0)) \triangleq \min_{U_0} \quad & x_N' P x_N + \sum_{k=0}^{N-1} [x_k' Q x_k + u_k' R u_k] \\ \text{subject to} \quad & x_{k+1} = A x_k + B u_k,\ k = 0, \dots, N-1 \\ & x_0 = x(0) \end{aligned}$$

- $P \succeq 0$, with $P = P'$, is the *terminal* weight
- $Q \succeq 0$, with $Q = Q'$, is the *state* weight
- $R \succ 0$, with $R = R'$, is the *input* weight
- N is the horizon length
- Note that $x(0)$ is the current state, whereas $x_0, \dots, x_N$ and $u_0, \dots, u_{N-1}$ are *optimization variables* that are constrained to obey the system dynamics and the initial condition.

Linear Quadratic Optimal Control

Batch Approach

- The problem is unconstrained
- Setting the gradient to zero:

$$U_0^*(x(0)) = \mathbf{K}x(0)$$

- which implies

$$u^*(0)(x(0)) = K_0 x(0), \ldots, u^*(N-1)(x(0)) = K_{N-1}x(0)$$

which is a linear, open-loop controller function of the initial state $x(0)$.

- The optimal cost is

$$J_0^*(x(0)) = x(0)' P_0 x(0)$$

which is a positive definite quadratic function of the initial state $x(0)$.

Linear Quadratic Optimal Control

Batch Approach

- The batch solution explicitly represents all future states x_k in terms of initial condition x_0 and inputs $u_0, \ldots, u_{N-1}$.
- Starting with $x_0 = x(0)$, we have $x_1 = Ax(0) + Bu_0$, and $x_2 = Ax_1 + Bu_1 = A^2x(0) + ABu_0 + Bu_1$, by substitution for x_1, and so on. Continuing up to x_N we obtain:

$$\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} I \\ A \\ \vdots \\ \vdots \\ A^N \end{bmatrix} x(0) + \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \ddots & \ddots & 0 \\ A^{N-1}B & \cdots & AB & B \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ \vdots \\ u_{N-1} \end{bmatrix}$$

- The equation above can be represented as

$$\mathcal{X} \triangleq \mathcal{S}^x x(0) + \mathcal{S}^u U_0 \,. \tag{3}$$

Linear Quadratic Optimal Control

Batch Approach

- Define

$$\overline{Q} \triangleq \text{blockdiag}(Q, \dots, Q, P) \quad \text{and} \quad \overline{R} \triangleq \text{blockdiag}(R, \dots, R)$$

Then the finite horizon cost function (2) can be written as

$$J_0(x(0), U_0) = \mathcal{X}'\overline{Q}\mathcal{X} + U_0{}'\overline{R}U_0\,. \tag{4}$$

- Eliminating $\mathcal{X}$ by substituting from (3), equation (4) can be expressed as:

$$\begin{aligned} J_0(x(0), U_0) &= (\mathcal{S}^x x(0) + \mathcal{S}^u U_0)'\overline{Q}(\mathcal{S}^x x(0) + \mathcal{S}^u U_0) + U_0{}'\overline{R}U_0 \\ &= U_0{}'HU_0 + 2x(0)'FU_0 + x(0)'\mathcal{S}^{x\prime}\overline{Q}\mathcal{S}^x x(0) \end{aligned}$$

where $H \triangleq \mathcal{S}^{u\prime}\overline{Q}\mathcal{S}^u + \overline{R}$ and $F \triangleq \mathcal{S}^{x\prime}\overline{Q}\mathcal{S}^u$.

- Note that $H \succ 0$, since $R \succ 0$ and $\mathcal{S}^{u\prime}\overline{Q}\mathcal{S}^u \succeq 0$.

Linear Quadratic Optimal Control

Batch Approach

- Since the problem is unconstrained and $J_0(x(0), U_0)$ is a positive definite quadratic function of U_0 we can solve for the optimal input U_0^* by setting the gradient with respect to U_0 to zero:

$$\begin{aligned}\nabla_{U_0} J_0(x(0), U_0) &= 2HU_0 + 2F' x(0) = 0 \\ \Rightarrow U_0^*(x(0)) &= -H^{-1} F' x(0) \\ &= -(\mathcal{S}^{u\prime} \overline{Q} \mathcal{S}^u + \overline{R})^{-1} \mathcal{S}^{u\prime} \overline{Q} \mathcal{S}^x x(0)\,,\end{aligned}$$

  which is a linear function of the initial state $x(0)$.
  Note H^{-1} always exists, since $H \succ 0$ and therefore has full rank.

- The optimal cost can be shown (by back-substitution) to be

$$\begin{aligned}J_0^*(x(0)) &= -x(0)' FHF' x(0) + x(0)' \mathcal{S}^{x\prime} \overline{Q} \mathcal{S}^x x(0) \\ &= x(0)' (\mathcal{S}^{x\prime} \overline{Q} \mathcal{S}^x - \mathcal{S}^{x\prime} \overline{Q} \mathcal{S}^u (\mathcal{S}^{u\prime} \overline{Q} \mathcal{S}^u + \overline{R})^{-1} \mathcal{S}^{u\prime} \overline{Q} \mathcal{S}^x) x(0)\,,\end{aligned}$$

Linear Quadratic Optimal Control

Batch Approach

Summary

- The Batch Approach expresses the cost function in terms of the initial state $x(0)$ and input sequence U_0 by eliminating the states x_k.
- Because the cost $J_0(x(0), U_0)$ is a strictly convex quadratic function of U_0, its minimizer U_0^* is unique and can be found by setting $\nabla_{U_0} J_0(x(0), U_0) = 0$. This gives the optimal input sequence U_0^* as a linear function of the intial state $x(0)$:

$$U_0^*(x(0)) = -(\mathcal{S}^{u'}\overline{Q}\mathcal{S}^u + \overline{R})^{-1}\mathcal{S}^{u'}\overline{Q}\mathcal{S}^x x(0)$$

- The optimal cost is a quadratic function of the initial state $x(0)$

$$J_0^*(x(0)) = x(0)'(\mathcal{S}^{x'}\overline{Q}\mathcal{S}^x - \mathcal{S}^{x'}\overline{Q}\mathcal{S}^u(\mathcal{S}^{u'}\overline{Q}\mathcal{S}^u + \overline{R})^{-1}\mathcal{S}^{u'}\overline{Q}\mathcal{S}^x)x(0)$$

- If there are state or input constraints, solving this problem by matrix inversion is not guaranteed to result in a feasible input sequence

Recursive Approach

- Alternatively, we can use dynamic programming to solve the same problem in a recursive manner.
- Define the "j-step optimal cost-to-go" as the *optimal* cost attainable for the step j problem:

$$
\begin{aligned}
J_j^*(x(j)) \triangleq \min_{u_j,\ldots,u_{N-1}} \quad & x_N' P x_N + \sum_{k=j}^{N-1} [x_k' Q x_k + u_k' R u_k] \\
\text{subject to} \quad & x_{k+1} = A x_k + B u_k,\ k = j, \ldots, N-1 \\
& x_j = x(j)
\end{aligned}
$$

This is the minimum cost attainable for the remainder of the horizon after step j

Linear Quadratic Optimal Control

Recursive Approach

- Consider the 1-step problem (solved at time $N-1$)

$$J^*_{N-1}(x_{N-1}) = \min_{u_{N-1}} \{x'_{N-1} Q x_{N-1} + u'_{N-1} R u_{N-1} + x'_N P_N x_N\} \tag{5}$$

$$\text{subject to} \quad x_N = A x_{N-1} + B u_{N-1} \tag{6}$$

$$P_N = P$$

where we introduced the notation P_j to express the optimal cost-to-go $x'_j P_j x_j$. In particular, $P_N = P$.

- Substituting (6) into (5)

$$\begin{aligned} J^*_{N-1}(x_{N-1}) = \min_{u_{N-1}} \{ & x'_{N-1}(A' P_N A + Q) x_{N-1} \\ & + u'_{N-1}(B' P_N B + R) u_{N-1} \\ & + 2 x'_{N-1} A' P_N B u_{N-1}\} \end{aligned}$$

Linear Quadratic Optimal Control

Recursive Approach

- Solving again by setting the gradient to zero leads to the following optimality condition for u_{N-1}

$$2(B'P_NB+R)u_{N-1}+2B'P_NAx_{N-1}=0$$

Optimal 1-step input:

$$\begin{aligned} u^*_{N-1} &= -(B'P_NB+R)^{-1}B'P_NAx_{N-1} \\ &\triangleq F_{N-1}x_{N-1} \end{aligned}$$

1-step cost-to-go:

$$J^*_{N-1}(x_{N-1}) = x'_{N-1}P_{N-1}x_{N-1}\,,$$

where

$$P_{N-1} = A'P_NA+Q-A'P_NB(B'P_NB+R)^{-1}B'P_NA\,.$$

Linear Quadratic Optimal Control

Recursive Approach

- The recursive solution method used from here relies on Bellman's **Principle of Optimality**
- For any solution for steps 0 to N to be optimal, any solution for steps j to N with $j \geq 0$, taken from the 0 to N solution, must itself be optimal for the j-to-N problem
- Therefore we have, for any $j = 0, \ldots, N$

$$J_j^*(x_j) = \min_{u_j}\{J_{j+1}^*(x_{j+1}) + x_j' Q x_j + u_j' R u_j\}$$

$$\text{s.t.} \quad x_{j+1} = Ax_j + Bu_j$$

- *Suppose that the fastest route from Los Angeles to Boston passes through Chicago. Then the principle of optimality formalizes the obvious fact that the Chicago to Boston portion of the route is also the fastest route for a trip that starts from Chicago and ends in Boston.*

Linear Quadratic Optimal Control

Recursive Approach

- Now consider the 2-step problem, posed at time $N-2$

$$J^*_{N-2}(x_{N-2}) = \min_{u_{N-1},u_{N-2}} \left\{ \sum_{k=N-2}^{N-1} x_k' Q x_k + u_k' R u_k + x_N' P x_N \right\}$$

$$\text{s.t.} \quad x_{k+1} = Ax_k + Bu_k, \quad k = N-2, N-1$$

- From the Principle of Optimality, the cost function is equivalent to

$$\begin{aligned} J^*_{N-2}(x_{N-2}) &= \min_{u_{N-2}} \{ J^*_{N-1}(x_{N-1}) \\ &\quad + x_{N-2}' Q x_{N-2} + u_{N-2}' R u_{N-2} \} \\ &= \min_{u_{N-2}} \{ x_{N-1}' P_{N-1} x_{N-1}' \\ &\quad + x_{N-2} Q x_{N-2} + u_{N-2}' R u_{N-2} \} \end{aligned}$$

Linear Quadratic Optimal Control

Recursive Approach

- As with 1-step solution, solve by setting the gradient with respect to u_{N-2} to zero

Optimal 2-step input

$$\begin{aligned} u_{N-2}^* &= -(B'P_{N-1}B + R)^{-1}B'P_{N-1}Ax_{N-2} \\ &\triangleq F_{N-2}x_{N-2} \end{aligned}$$

2-step cost-to-go

$$J_{N-2}^*(x_{N-2}) = x_{N-2}'P_{N-2}x_{N-2}\,,$$

where

$$P_{N-2} = A'P_{N-1}A + Q - A'P_{N-1}B(B'P_{N-1}B + R)^{-1}B'P_{N-1}A$$

- We now recognize the recursion for P_j and u_j^*, $j = N-1, \cdots, 0$.

Linear Quadratic Optimal Control

Recursive Approach

- We can obtain the solution for any given time step k in the horizon

$$u^*(k) = -(B'P_{k+1}B + R)^{-1}B'P_{k+1}Ax(k)$$
$$\triangleq F_k x(k) \quad \text{for } k = 1, \ldots, N$$

where we can find any P_k by recursive evaluation from $P_N = P$, using

$$P_k = A'P_{k+1}A + Q - A'P_{k+1}B(B'P_{k+1}B + R)^{-1}B'P_{k+1}A \tag{7}$$

This is called the *Discrete Time Riccati Equation* or *Riccati Difference Equation (RDE)*.

- Evaluating down to P_0, we obtain the N-step cost-to-go

$$J_0^*(x(0)) = x(0)'P_0x(0)$$

Linear Quadratic Optimal Control

Recursive Approach

Summary

- From the Principle of Optimality, the optimal control policy for any step j is then given by

$$u^*(k) = -(B'P_{k+1}B + R)^{-1}B'P_{k+1}Ax(k) = F_k x(k)$$

and the optimal cost-to-go is

$$J_k^*(x(k)) = x_k' P_k x(k)$$

- Each P_k is related to P_{k+1} by the Riccati Difference Equation

$$P_k = A'P_{k+1}A + Q - A'P_{k+1}B(B'P_{k+1}B + R)^{-1}B'P_{k+1}A\,,$$

which can be initialized with $P_N = P$, the given terminal weight

Comparison of Batch and Recursive Approaches

- Fundamental difference: Batch optimization returns a sequence $U_0^*(x(0))$ of *numeric values* depending only on the initial state $x(0)$, while dynamic programming yields *feedback policies* $u^*(k) = F_k x(k)$, $k = 0, \ldots, N-1$ depending on each $x(k)$.
- If the state evolves exactly as modelled, then the sequences of control actions obtained from the two approaches are identical.
- The recursive solution should be more robust to disturbances and model errors, because if the future states later deviate from their predicted values, the exact optimal input can still be computed.
- The Recursive Approach is computationally more attractive because it breaks the problem down into single-step problems. For large horizon length, the Hessian H in the Batch Approach, which must be inverted, becomes very large.

Linear Quadratic Optimal Control

Comparison of Batch and Recursive Approaches

- Without any modification, both solution methods will break down when inequality constraints on x_k or u_k are added.

- The Batch Approach is far easier to adapt than the Recursive Approach when constraints are present: just perform a constrained minimization for the current state.

- Doing this at every time step within the time available, and then using only the first input from the resulting sequence, amounts to receding horizon control.