Model Predictive Control
Unconstrained and Constrained Optimal Control

F. Borrelli, M. Morari, C. Jones

UC Berkeley

ETH Zürich

EPFL Lausanne

Fall Semester 2015

Table of Contents

1. Optimal Control
1.1 Introduction
1.2 Batch Approach
1.3 Recursive Approach
1.4 The Dynamic Programming algorithm
1.5 Comments
2. Linear Quadratic Optimal Control
2.1 Batch Approach
2.2 Recursive Approach
2.3 Infinite Horizon Optimal Control
3. Constrained Linear Optimal Control
3.1 Problem formulation
3.2 Feasible Sets
4. Constrained Optimal Control: 2-Norm
4.1 Problem Formulation
4.2 Construction of the QP with substitution
4.3 Construction of the QP without substitution
4.4 2-Norm State Feedback Solution
5. Constrained Optimal Control: 1-Norm and ∞-Norm
5.1 Problem Formulation

Table of Contents

1. Optimal Control
1.1 Introduction
1.2 Batch Approach
1.3 Recursive Approach
1.4 The Dynamic Programming algorithm
1.5 Comments

Optimal Control Introduction (1/2)

- Discrete-time *optimal control* is concerned with choosing an optimal input sequence $U_{0\to N} \triangleq [u_0', u_1', \ldots]'$ (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* $p(x_N)$:

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

- The states $\{x_k\}_{k=0}^{N}$ must satisfy the system dynamics

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

and there may be state and/or input constraints

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

Optimal Control Introduction (2/2)

- In the finite horizon case, there may also be a constraint that the final state x_N lies in a set $\mathcal{X}_f$

$$x_N \in \mathcal{X}_f$$

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

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

General Problem Formulation

Consider the nonlinear time-invariant system

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

subject to the constraints

$$h(x(t), u(t)) \leq 0, \ \forall t \geq 0$$

with $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$ the state and input vectors. ***Assume that*** $g(0,0) = 0$, $h(0,0) \leq 0$.

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 $x_0 = x(0)$,
- $U_{0\to N} := [u_0', \ldots, u_{N-1}']' \in \mathbb{R}^s, \ s := mN$,
- $q(x_k, u_k)$ and $p(x_N)$ are the *stage cost* and *terminal cost*, respectively.

General Problem Formulation

Consider the **C**onstrained **F**inite **T**ime **O**ptimal **C**ontrol (CFTOC) problem.

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

- $\mathcal{X}_f \subseteq \mathbb{R}^n$ is a *terminal region*,
- $\mathcal{X}_{0\to N} \subseteq \mathbb{R}^n$ 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 $U^*_{0\to N}$ one of the minima

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 $N \to \infty$,
  2. the properties of this solution.
  3. approximation of the solution by using a *receding horizon* technique.

- ***Uncertainty***. We will discuss how to extend the problem description and consider uncertainty

Table of Contents

1. Optimal Control
1.1 Introduction
1.2 Batch Approach
1.3 Recursive Approach
1.4 The Dynamic Programming algorithm
1.5 Comments

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

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

is a general Non Linear Programming (NLP) problem with variables $u_0, \dots, u_{N-1}$ and $x_1, \dots, x_N$.

Table of Contents

1. Optimal Control
1.1 Introduction
1.2 Batch Approach
1.3 Recursive Approach
1.4 The Dynamic Programming algorithm
1.5 Comments

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. $x_j^*, x_{j+1}^*, \ldots, x_N^*$, $0 \leq j \leq N-1$, must be optimal.

Suppose that the fastest route from Los Angeles to Boston passes through Chicago. 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.

Solution via Recursive Approach

Principle of optimality

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

Define the cost from j to N

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

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

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

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

Solution via Recursive Approach

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

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

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.

Table of Contents

1. Optimal Control
1.1 Introduction
1.2 Batch Approach
1.3 Recursive Approach
1.4 The Dynamic Programming algorithm
1.5 Comments

Solution via Recursive Approach

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

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

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

Table of Contents

1. Optimal Control
1.1 Introduction
1.2 Batch Approach
1.3 Recursive Approach
1.4 The Dynamic Programming algorithm
1.5 Comments

Solution via Recursive Approach: Comments

- DP algorithm is appealing because at each step j only u_j is computed.
- Need to construct the optimal cost-to-go $J^*_{N-j}(x_j)$, a *function* defined over $\mathcal{X}_{j\to N}$.
- In a few special cases we know the type of function and we can find it efficiently.
- "brute force" approach. Construct $J_{j-1\to N}$ by griding the set $\mathcal{X}_{j-1\to N}$ and computing the optimal cost-to-go function on each grid point.
- A nonlinear *feedback* (time varying) control law is implicitly defined:

$$\begin{aligned} u^*_j(x_j) = \quad & \arg\min_{u_j} \quad q(x_j,u_j) + J^*_{j+1\to N}(g(x_j,u_j)) \\ & \text{subj. to} \quad h(x_j,u_j) \le 0, \\ & \qquad\quad g(x_j,u_j) \in \mathcal{X}_{j+1\to N} \end{aligned}$$

Notation

For the sake of simplicity we will use the following shorter notation

$$
\begin{aligned}
&J_j^*(x_j) := J_{j\to N}^*(x_j),\ j = 0, \dots, N \\
&J_\infty^*(x_0) := J_{0\to\infty}^*(x_0) \\
&\mathcal{X}_j := \mathcal{X}_{j\to N},\ j = 0, \dots, N \\
&\mathcal{X}_\infty := \mathcal{X}_{0\to\infty} \\
&U_0 := U_{0\to N}
\end{aligned}
$$

Outline

1. Optimal Control

2. Linear Quadratic Optimal Control
2.1 Batch Approach
2.2 Recursive Approach
2.3 Infinite Horizon Optimal Control

3. Constrained Linear Optimal Control

4. Constrained Optimal Control: 2-Norm

5. Constrained Optimal Control: 1-Norm and ∞-Norm

Linear Quadratic Optimal Control

- In this section, only *linear* discrete-time time-invariant systems

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

and *quadratic* cost functions

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

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
  1. *Batch Approach*, which yields a series of *numerical values* for the input
  2. *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.

Unconstrained Finite Horizon Control Problem

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

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

- $P \succeq 0$, 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 $x_0, \ldots, x_N$ and $u_0, \ldots, u_{N-1}$ are *optimization variables* that are constrained to obey the system dynamics and the initial condition.

Table of Contents

2. Linear Quadratic Optimal Control
2.1 Batch Approach
2.2 Recursive Approach
2.3 Infinite Horizon Optimal Control

Final Result

- 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 $x(0)$.

Solution approach 1: Batch Approach (1/4)

- 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

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

Solution approach 1: Batch Approach (2/4)

- Define

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

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:

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

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

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

Solution approach 1: Batch Approach (3/4)

- 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:

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

  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

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

Solution approach 1: Batch Approach (4/4)

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

Table of Contents

2. Linear Quadratic Optimal Control
2.1 Batch Approach
2.2 Recursive Approach
2.3 Infinite Horizon Optimal Control

Final Result

- The problem is unconstrained
- Using the Dynamic Programming Algorithm we have

$$u^*(k) = F_k x(k)$$

  which is a linear, time-varying state-feedback controller.
- the optimal cost-to-go $k \to N$ is

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

  which is a positive definite quadratic function of the state at time k
- F_k is computed by using P_{k+1}
- Each P_k is related to P_{k+1} by a recursive equation (Riccati Difference Equation)

Solution approach 2: Recursive Approach (1/8)

- 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

Solution approach 2: Recursive Approach (2/8)

- 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)

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

Solution approach 2: Recursive Approach (3/8)

- 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

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

Solution approach 2: Recursive Approach (4/8)

- 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 $j = 0, \ldots, N$

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

$$\text{s.t.} \quad x_{j+1} = A x_j + B u_j$$

- *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.*

Solution approach 2: Recursive Approach (5/8)

- 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

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

Solution approach 2: Recursive Approach (6/8)

- 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, $j = N-1, \cdots, 0$.

Solution approach 2: Recursive Approach (7/8)

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

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

  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

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

Solution approach 2: Recursive Approach (8/8)

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 (1/2)

- 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.

Comparison of Batch and Recursive Approaches (2/2)

- 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*.

Table of Contents

2. Linear Quadratic Optimal Control
2.1 Batch Approach
2.2 Recursive Approach
2.3 Infinite Horizon Optimal Control

Infinite Horizon Control Problem: Optimal Sol'n (1/2)

- In some cases we may want to solve the same problem with an infinite horizon:

$$J_\infty(x(0)) = \min_{u(\cdot)} \left\{ \sum_{k=0}^{\infty} [x_k' Q x_k + u_k' R u_k] \right\}$$

$$\text{subject to} \quad x_{k+1} = Ax_k + Bu_k, \quad k = 0, 1, 2, \ldots, \infty,$$

$$x_0 = x(0)$$

- As with the Dynamic Programming approach, the optimal input is of the form

$$u^*(k) = -(B'P_\infty B + R)^{-1} B'P_\infty Ax(k) \triangleq F_\infty x(k)$$

and the infinite-horizon cost-to-go is

$$J_\infty(x(k)) = x(k)'P_\infty x(k)\,.$$

Infinite Horizon Control Problem: Optimal Sol'n (2/2)

- The matrix P_∞ comes from an infinite recursion of the RDE, from a point infinitely far into the future.
- Assuming the RDE does converge to some constant matrix P_∞, it must satisfy the following (from (7), with $P_k = P_{k+1} = P_\infty$)

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

  which is called the *Algebraic Riccati Equation (ARE)*.
- The constant feedback matrix F_∞ is referred to as the asymptotic form of the *Linear Quadratic Regulator (LQR)*.
- In fact, if (A, B) is controllable and (Q, A) is observable, then the RDE (initialized with Q at $k = \infty$ and solved for $k \searrow 0$) converges to the unique positive definite solution P_∞ of the ARE.

Stability of Infinite-Horizon LQR

- In addition, the closed-loop system with $u(k) = F_\infty x(k)$ is guaranteed to be asymptotically stable, under the stabilizability and detectability assumptions of the previous slide.
- The latter statement can be proven by substituting the control law $u(k) = F_\infty x(k)$ into $x(k+1) = Ax(k) + Bu(k)$, and then examining the properties of the system

$$x(k+1) = (A + BF_\infty)x(k)\,. \tag{8}$$

- The asymptotic stability of (8) can be proven by showing that the infinite horizon cost $J^*_\infty(x(k)) = x(k)'P_\infty x(k)$ is actually a Lyapunov function for the system, i.e. $J^*_\infty(x(k)) > 0,\ \forall k \neq 0,\ J^*_\infty(0) = 0$, and $J^*_\infty(x(k+1)) < J^*_\infty(x(k))$, for any $x(k)$. This implies that

$$\lim_{k\to\infty} x(k) = 0\,.$$

Outline

1. Optimal Control

2. Linear Quadratic Optimal Control

3. Constrained Linear Optimal Control
3.1 Problem formulation
3.2 Feasible Sets

4. Constrained Optimal Control: 2-Norm

5. Constrained Optimal Control: 1-Norm and ∞-Norm

Table of Contents

3. Constrained Linear Optimal Control
3.1 Problem formulation
3.2 Feasible Sets

Constrained Linear Optimal Control

Cost function

$$J_0(x(0), U_0) = p(x_N) + \sum_{k=0}^{N-1} q(x_k, u_k)$$

- $U_0 \triangleq [u_0', \dots, u_{N-1}']'$
- Squared Euclidian norm: $p(x_N) = x_N' P x_N$ and $q(x_k, u_k) = x_k' Q x_k + u_k' R u_k$.
- $p = 1$ or $p = \infty$: $p(x_N) = \|Px_N\|_p$ and $q(x_k, u_k) = \|Qx_k\|_p + \|Ru_k\|_p$.

Constrained Finite Time Optimal Control problem (CFTOC)

$$\begin{aligned} J_0^*(x(0)) = \quad & \min_{U_0} \quad && J_0(x(0), U_0) \\ & \text{subj. to} && x_{k+1} = Ax_k + Bu_k, \;\; k = 0, \dots, N-1 \\ & && x_k \in \mathcal{X}, \;\; u_k \in \mathcal{U}, \;\; k = 0, \dots, N-1 \\ & && x_N \in \mathcal{X}_f \\ & && x_0 = x(0) \end{aligned} \tag{9}$$

N is the time horizon and $\mathcal{X}$, $\mathcal{U}$, $\mathcal{X}_f$ are polyhedral regions.

Table of Contents

3. Constrained Linear Optimal Control
3.1 Problem formulation
3.2 Feasible Sets

Feasible Sets

Set of initial states $x(0)$ for which the optimal control problem (9) is feasible:

$$\begin{aligned}\mathcal{X}_0 = \quad & \{x_0 \in \mathbb{R}^n | \ \exists (u_0, \dots, u_{N-1}) \text{ such that } x_k \in \mathcal{X}, \ u_k \in \mathcal{U}, \\ & k = 0, \dots, N-1, \ x_N \in \mathcal{X}_f, \ \text{where } x_{k+1} = Ax_k + Bu_k\}\end{aligned}$$

In general $\mathcal{X}_i$ is the set of states x_i at time i for which (9) is feasible:

$$\begin{aligned}\mathcal{X}_i = \quad & \{x_i \in \mathbb{R}^n | \ \exists (u_i, \dots, u_{N-1}) \text{ such that } x_k \in \mathcal{X}, \ u_k \in \mathcal{U}, \\ & k = i, \dots, N-1, \ x_N \in \mathcal{X}_f, \ \text{where } x_{k+1} = Ax_k + Bu_k\},\end{aligned}$$

The sets $\mathcal{X}_i$ for $i = 0, \dots, N$ play an important role in the the solution of the CFTOC problem. They are independent of the cost.

Outline

1. Optimal Control

2. Linear Quadratic Optimal Control

3. Constrained Linear Optimal Control

4. Constrained Optimal Control: 2-Norm
4.1 Problem Formulation
4.2 Construction of the QP with substitution
4.3 Construction of the QP without substitution
4.4 2-Norm State Feedback Solution

5. Constrained Optimal Control: 1-Norm and ∞-Norm

Table of Contents

4. Constrained Optimal Control: 2-Norm
4.1 Problem Formulation
4.2 Construction of the QP with substitution
4.3 Construction of the QP without substitution
4.4 2-Norm State Feedback Solution

Problem Formulation

Quadratic cost function

$$J_0(x(0), U_0) = x_N' P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k \tag{10}$$

with $P \succeq 0,\ Q \succeq 0,\ R \succ 0$.

Constrained Finite Time Optimal Control problem (CFTOC).

$$\begin{aligned} J_0^*(x(0)) = \quad & \min_{U_0} \quad && J_0(x(0), U_0) \\ & \text{subj. to} && x_{k+1} = Ax_k + Bu_k,\ \ k = 0, \dots, N-1 \\ & && x_k \in \mathcal{X},\ \ u_k \in \mathcal{U},\ \ k = 0, \dots, N-1 \\ & && x_N \in \mathcal{X}_f \\ & && x_0 = x(0) \end{aligned} \tag{11}$$

N is the time horizon and $\mathcal{X}$, $\mathcal{U}$, $\mathcal{X}_f$ are polyhedral regions.

Table of Contents

4. Constrained Optimal Control: 2-Norm
4.1 Problem Formulation
4.2 Construction of the QP with substitution
4.3 Construction of the QP without substitution
4.4 2-Norm State Feedback Solution

Construction of the QP with substitution

- **Step 1**: Rewrite the cost as (see previous slides)

$$\begin{aligned} J_0(x(0), U_0) &= U_0' H U_0 + 2x(0)' F U_0 + x(0)' Y x(0) \\ &= \begin{bmatrix} U_0' & x(0)' \end{bmatrix} \begin{bmatrix} H & F' \\ F & Y \end{bmatrix} \begin{bmatrix} U_0' & x(0)' \end{bmatrix}' \end{aligned}$$

- **Step 2**: Rewrite the constraints compactly as (details provided on the next slide)

$$G_0 U_0 \leq w_0 + E_0 x(0)$$

- **Step 3**: Rewrite the optimal control problem as

$$\begin{aligned} J_0^*(x(0)) = \min_{U_0} \quad & \begin{bmatrix} U_0' & x(0)' \end{bmatrix} \begin{bmatrix} H & F' \\ F & Y \end{bmatrix} \begin{bmatrix} U_0' & x(0)' \end{bmatrix}' \\ \text{subj. to} \quad & G_0 U_0 \leq w_0 + E_0 x(0) \end{aligned}$$

Solution

$$J_0^*(x(0)) = \min_{U_0} \quad [U_0' \; x(0)'] \begin{bmatrix} H & F' \\ F & Y \end{bmatrix} [U_0' \; x(0)']'$$
$$\text{subj. to} \quad G_0 U_0 \leq w_0 + E_0 x(0)$$

For a given x(0) U_0^* can be found via a QP solver.

Construction of QP constraints with substitution

If $\mathcal{X}$, $\mathcal{U}$ and $\mathcal{X}_f$ are given by:

$$\mathcal{X} = \{x \,|\, A_x x \leq b_x\} \qquad \mathcal{U} = \{u \,|\, A_u u \leq b_u\} \qquad \mathcal{X}_f = \{x \,|\, A_f x \leq b_f\}$$

Then G_0, E_0 and w_0 are defined as follows

$$G_0 = \begin{bmatrix} A_u & 0 & \dots & 0 \\ 0 & A_u & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_u \\ 0 & 0 & \dots & 0 \\ A_x B & 0 & \dots & 0 \\ A_x AB & A_x B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_f A^{N-1} B & A_f A^{N-2} B & \dots & A_f B \end{bmatrix}, E_0 = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -A_x \\ -A_x A \\ -A_x A^2 \\ \vdots \\ -A_f A^N \end{bmatrix}, w_0 = \begin{bmatrix} b_u \\ b_u \\ \vdots \\ b_u \\ b_x \\ b_x \\ b_x \\ \vdots \\ b_f \end{bmatrix}$$

Table of Contents

4. Constrained Optimal Control: 2-Norm
4.1 Problem Formulation
4.2 Construction of the QP with substitution
4.3 Construction of the QP without substitution
4.4 2-Norm State Feedback Solution

Construction of the QP without substitution

To obtain the QP problem

$$J_0^*(x(0)) = \min_{U_0} \quad [U_0' \; x(0)'] \begin{bmatrix} H & F' \\ F & Q \end{bmatrix} [U_0' \; x(0)']'$$
$$\text{subj. to} \quad G_0 U_0 \leq w_0 + E_0 x(0)$$

we have substituted the state equations

$$x_{k+1} = Ax_k + Bu_k$$

into the state constraints $x_k \in \mathcal{X}$.

It is often more efficient to keep the explicit equality constraints.

Construction of the QP without substitution

We transform the CFTOC problem into the QP problem

$$\begin{aligned} J_0^*(x(0)) = \min_z \quad & [z' \ x(0)'] \begin{bmatrix} \bar{H} & 0 \\ 0 & Q \end{bmatrix} [z' \ x(0)']' \\ \text{subj. to} \quad & G_{0,\text{in}} z \leq w_{0,\text{in}} + E_{0,\text{in}} x(0) \\ & G_{0,\text{eq}} z = E_{0,\text{eq}} x(0) \end{aligned}$$

- Define variable:

$$z = \begin{bmatrix} x_1' & \dots & x_N' & u_0' & \dots & u_{N-1}' \end{bmatrix}'$$

- Equalities from system dynamics $x_{k+1} = Ax_k + Bu_k$:

$$G_{0,\text{eq}} = \left[\begin{array}{ccccc:cccc} I & & & & & -B & & & \\ -A & I & & & & & -B & & \\ & -A & I & & & & & -B & \\ & & \ddots & \ddots & & & & & \ddots \\ & & & -A & I & & & & -B \end{array}\right], E_{0,\text{eq}} = \begin{bmatrix} A \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Construction of the QP without substitution

If $\mathcal{X}$, $\mathcal{U}$ and $\mathcal{X}_f$ are given by:

$$\mathcal{X} = \{x \,|\, A_x x \leq b_x\} \qquad \mathcal{U} = \{u \,|\, A_u u \leq b_u\} \qquad \mathcal{X}_f = \{x \,|\, A_f x \leq b_f\}$$

Then matrices $G_{0,\text{in}}$, $w_{0,\text{in}}$ and $E_{0,\text{in}}$ are:

$$G_{0,\text{in}} = \left[\begin{array}{ccccc|ccccc} 0 & & & & & 0 & & & & \\ A_x & & & & & 0 & & & & \\ & \ddots & & & & & \ddots & & & \\ & & A_x & & & & & 0 & & \\ & & & A_f & & & & & 0 & \\ \hline 0 & & & & & A_u & & & & \\ 0 & & & & & A_u & & & & \\ & \ddots & & & & & \ddots & & & \\ & & 0 & & & & & A_u & & \\ & & & 0 & & & & & A_u & \end{array}\right] \qquad w_{0,\text{in}} = \left[\begin{array}{c} b_x \\ b_x \\ \vdots \\ b_x \\ b_f \\ \hline b_u \\ b_u \\ \vdots \\ b_u \\ b_u \end{array}\right]$$

$$E_{0,\text{in}} = \begin{bmatrix} -A_x' & 0 & \cdots & 0 \end{bmatrix}'$$

Construction of the QP without substitution

Build cost function from MPC cost $x_N' P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k$

$$\bar{H} = \left[\begin{array}{ccc|ccc} Q & & & & & \\ & \ddots & & & & \\ & & Q & & & \\ & & & P & & \\ \hline & & & & R & \\ & & & & & \ddots \\ & & & & & R \end{array}\right]$$

Matlab hint:

```
barH = blkdiag(kron(eye(N-1),Q), P, kron(eye(N),R))
```

Table of Contents

4. Constrained Optimal Control: 2-Norm
4.1 Problem Formulation
4.2 Construction of the QP with substitution
4.3 Construction of the QP without substitution
4.4 2-Norm State Feedback Solution

2-Norm State Feedback Solution

Start from QP with substitution.

- **Step 1**: Define $z \triangleq U_0 + H^{-1}F'x(0)$ and transform the problem into

$$\begin{aligned} \hat{J}^*(x(0)) = \quad & \min_{z} \quad && z'Hz \\ & \text{subj. to} && G_0 z \leq w_0 + S_0 x(0), \end{aligned}$$

where $S_0 \triangleq E_0 + G_0 H^{-1} F'$, and
$\hat{J}^*(x(0)) = J_0^*(x(0)) - x(0)'(Y - FH^{-1}F')x(0)$.

The CFTOC problem is now a **multiparametric quadratic program (mp-QP)**.

- **Step 2**: Solve the mp-QP to get explicit solution $z^*(x(0))$

- **Step 3**: Obtain $U_0^*(x(0))$ from $z^*(x(0))$

2-Norm State Feedback Solution

Main Results

1. The **Open loop optimal control function** can be obtained by solving the mp-QP problem and calculating $U_0^*(x(0)), \ \forall x(0) \in \mathcal{X}_0$ as $U_0{}^* = z^*(x(0)) - H^{-1}F'x(0)$.
2. The first component of the multiparametric solution has the form

$$u^*(0) = f_0(x(0)), \quad \forall x(0) \in \mathcal{X}_0,$$

$f_0 : \mathbb{R}^n \to \mathbb{R}^m$, is continuous and PieceWise Affine on Polyhedra

$$f_0(x) = F_0^i x + g_0^i \quad \text{if} \quad x \in CR_0^i, \quad i = 1, \ldots, N_0^r$$

3. The polyhedral sets $CR_0^i = \{x \in \mathbb{R}^n | H_0^i x \leq K_0^i\}$, $i = 1, \ldots, N_0^r$ are a partition of the feasible polyhedron $\mathcal{X}_0$.
4. The value function $J_0^*(x(0))$ is convex and piecewise quadratic on polyhedra.

Example

Consider the double integrator

$$\begin{cases} x(t+1) &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) \end{cases}$$

subject to constraints

$$-1 \leq u(k) \leq 1, \ k = 0, \dots, 5$$

$$\begin{bmatrix} -10 \\ -10 \end{bmatrix} \leq x(k) \leq \begin{bmatrix} 10 \\ 10 \end{bmatrix}, \ k = 0, \dots, 5$$

Compute the **state feedback** optimal controller $u^*(0)(x(0))$ solving the CFTOC

problem with $N = 6$, $Q = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, $R = 0.1$, P the solution of the ARE, $\mathcal{X}_f = \mathbb{R}^2$.

Example

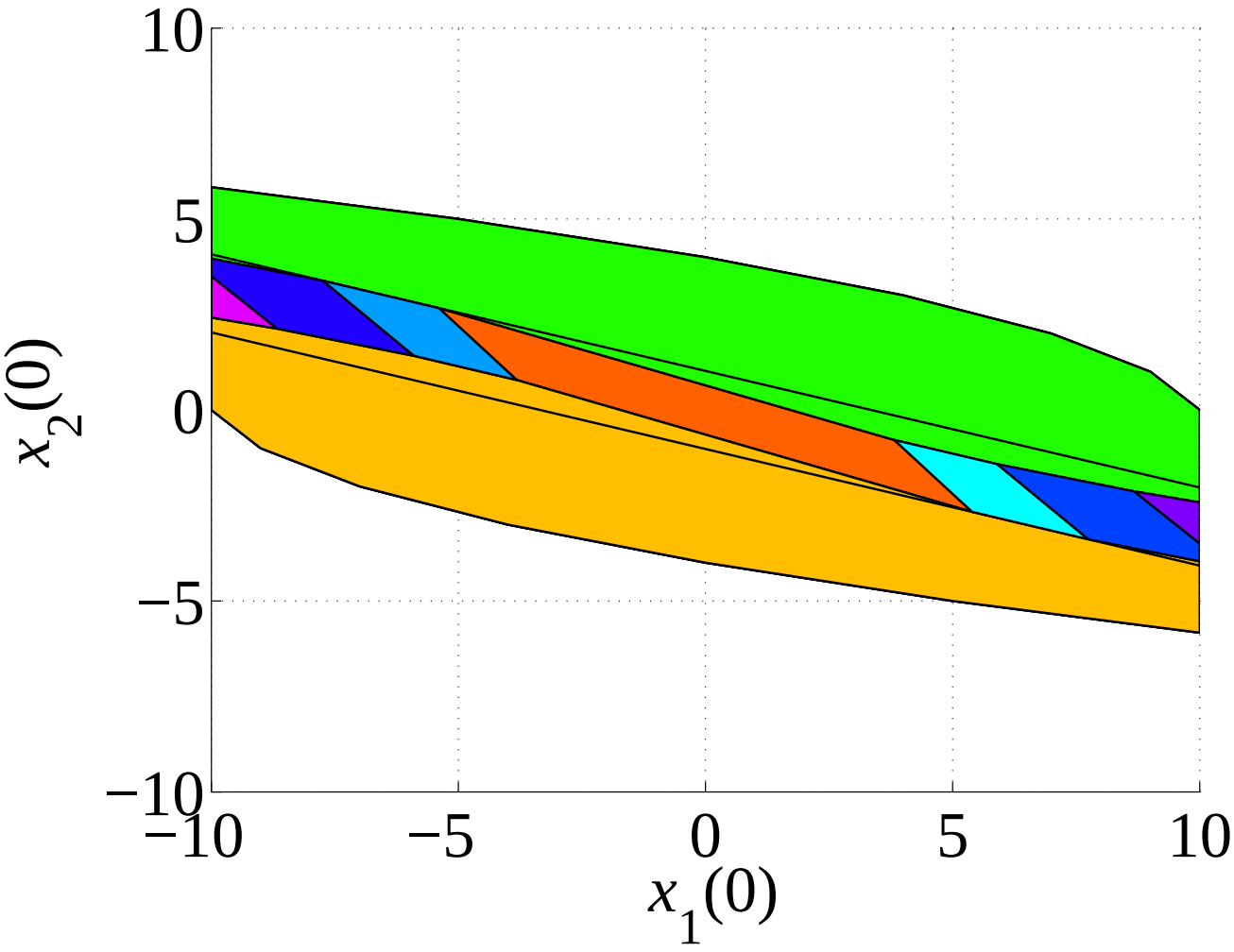

Figure: Partition of the state space for the affine control law $u^*(0)$ $(N_0^r = 13)$

Outline

1. Optimal Control

2. Linear Quadratic Optimal Control

3. Constrained Linear Optimal Control

4. Constrained Optimal Control: 2-Norm

5. Constrained Optimal Control: 1-Norm and ∞-Norm
5.1 Problem Formulation
5.2 Construction of the LP with substitution
5.3 1- /∞-Norm State Feedback Solution

Table of Contents

5. Constrained Optimal Control: 1-Norm and ∞-Norm
5.1 Problem Formulation
5.2 Construction of the LP with substitution
5.3 1- /∞-Norm State Feedback Solution

Problem Formulation

Piece-wise linear cost function

$$J_0(x(0), U_0) := \|Px_N\|_p + \sum_{k=0}^{N-1} \|Qx_k\|_p + \|Ru_k\|_p \tag{12}$$

with $p = 1$ or $p = \infty$, P, Q, R full column rank matrices

Constrained Finite Time Optimal Control Problem (CFTOC)

$$\begin{aligned} J_0^*(x(0)) = \quad & \min_{U_0} \quad && J_0(x(0), U_0) \\ & \text{subj. to} && x_{k+1} = Ax_k + Bu_k, \;\; k = 0, \dots, N-1 \\ & && x_k \in \mathcal{X}, \;\; u_k \in \mathcal{U}, \;\; k = 0, \dots, N-1 \\ & && x_N \in \mathcal{X}_f \\ & && x_0 = x(0) \end{aligned} \tag{13}$$

N is the time horizon and $\mathcal{X}$, $\mathcal{U}$, $\mathcal{X}_f$ are polyhedral regions.

Table of Contents

5. Constrained Optimal Control: 1-Norm and ∞-Norm
5.1 Problem Formulation
5.2 Construction of the LP with substitution
5.3 1- /∞-Norm State Feedback Solution

Construction of the LP with substitution

The problem results in the following standard LP

$$\begin{array}{ll} \min\limits_{z_0} & c_0' z_0 \\ \text{subj. to} & \bar{G}_0 z_0 \leq \bar{w}_0 + \bar{S}_0 x(0) \end{array}$$

where $z_0 := \{\varepsilon_0^x, \ldots, \varepsilon_N^x, \varepsilon_0^u, \ldots, \varepsilon_{N-1}^u, u_0', \ldots, u_{N-1}'\} \in \mathbb{R}^s$, $s \triangleq (m+1)N + N + 1$ and

$$\bar{G}_0 = \begin{bmatrix} G_\varepsilon & 0 \\ 0 & G_0 \end{bmatrix}, \; \bar{S}_0 = \begin{bmatrix} S_\varepsilon \\ S_0 \end{bmatrix}, \; \bar{w}_0 = \begin{bmatrix} w_\varepsilon \\ w_0 \end{bmatrix}$$

For a given $x(0)$ U_0^* can be obtained via an LP solver (the $1-$norm case is similar).

Construction of the LP with substitution - details

Recall that the $\infty-$norm problem can be equivalently formulated as

$$
\begin{aligned}
\min_{z_0} \quad & \varepsilon_0^x + \ldots + \varepsilon_N^x + \varepsilon_0^u + \ldots + \varepsilon_{N-1}^u \\
\text{subj. to} \quad & -\mathbf{1}_n \varepsilon_k^x \leq \pm Q \left[A^k x_0 + \sum_{j=0}^{k-1} A^j B u_{k-1-j} \right], \\
& -\mathbf{1}_r \varepsilon_N^x \leq \pm P \left[A^N x_0 + \sum_{j=0}^{N-1} A^j B u_{N-1-j} \right], \\
& -\mathbf{1}_m \varepsilon_k^u \leq \pm R u_k, \\
& A^k x_0 + \sum_{j=0}^{k-1} A^j B u_{k-1-j} \in \mathcal{X}, \;\; u_k \in \mathcal{U}, \\
& A^N x_0 + \sum_{j=0}^{N-1} A^j B u_{N-1-j} \in \mathcal{X}_f, \\
& k = 0, \ldots, N-1 \\
& x_0 = x(0)
\end{aligned}
$$

Table of Contents

5. Constrained Optimal Control: 1-Norm and ∞-Norm
5.1 Problem Formulation
5.2 Construction of the LP with substitution
5.3 1- /∞-Norm State Feedback Solution

1- /∞-Norm State Feedback Solution

Main Results

1. The **Open loop optimal control function** can be obtained by solving the mp-LP problem and calculating $z_0^*(x(0))$
2. The component $u_0^* = [0 \ \dots 0 \ I_m \ 0 \ \dots \ 0] z_0^*(x(0))$ of the multiparametric solution has the form

$$u^*(0) = f_0(x(0)), \quad \forall x(0) \in \mathcal{X}_0,$$

$f_0 : \mathbb{R}^n \to \mathbb{R}^m$, is continuous and PieceWise Affine on Polyhedra

$$f_0(x) = F_0^i x + g_0^i \quad \text{if} \quad x \in CR_0^i, \quad i = 1, \dots, N_0^r$$

3. The polyhedral sets $CR_0^i = \{x \in \mathbb{R}^n | H_0^i x \leq K_0^i\}$, $i = 1, \dots, N_0^r$ are a partition of the feasible polyhedron $\mathcal{X}_0$.
4. In case of multiple optimizers a PieceWise Affine control law exists.
5. The value function $J_0^*(x(0))$ is convex and piecewise linear on polyhedra.