Lecture 6: Constrained Finite Time Optimal Control

I. Constrained Optimal Control

- **Initial Remarks: Objectives of Constrained Optimal Control**

In the constrained finite time optimal control (CFTOC), we are given the dynamical system:

$$x^+ = f(x,u) \qquad (x,u) \in \mathcal{X},\ \mathcal{U}$$

Our objective is to design the control law $u = \kappa(x)$ such that system:

1. Satisfies constraints: $\{x_i\} \subset \mathcal{X},\ \{u_i\} \subset \mathcal{U}$
2. Is asymptotically stable: $\lim_{i\to\infty} x_i = 0$
3. Optimizes "performance"
4. Maximizes the set $\{x_0 \mid \text{Conditions } 1-3 \text{ are met}\}$

Note: In real applications, a common approach is two-step: first, design the unconstrained controller and then, patch the controller (e.g. chop out the exceeding part, also known as the anti-windup design) when the constrained are triggered, but such kind of design is not enough for complex and high requirement scenarios. We here want to explicitly consider the constraint in our design of the controller.

- **Limitations of Linear Controllers**

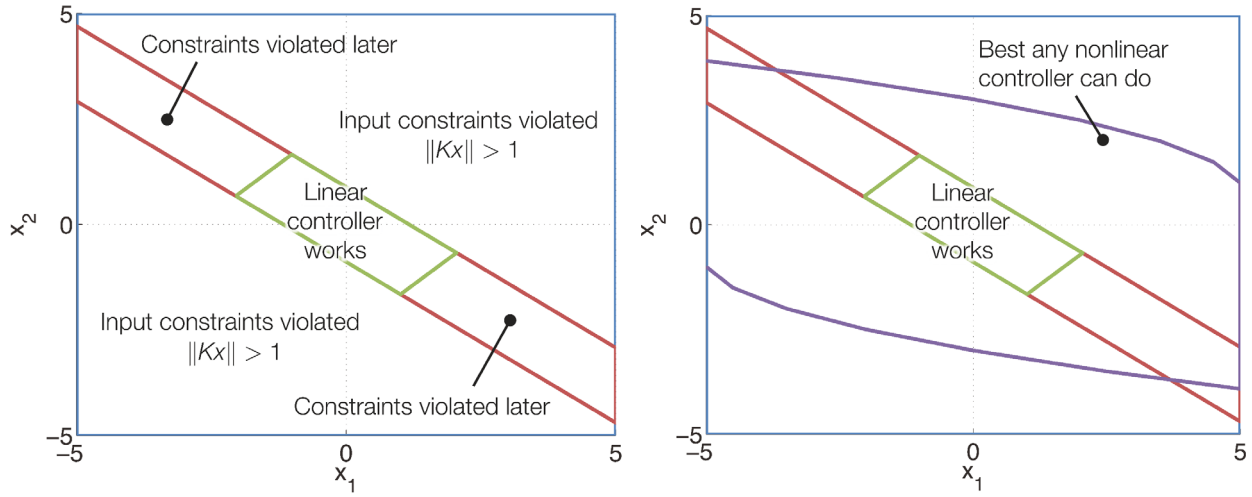
Consider the system given as:

$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u$$

with constraints:

$$\mathcal{X} := \{x \mid \|x\|_\infty \le 5\}$$
$$\mathcal{U} := \{u \mid \|u\|_\infty \le 1\}$$

Consider an LQR controller, with $Q = I, R = 1$, we get: $\Rightarrow K = [0.52 \quad 0.94]$



Does linear control work? Yes, but the region where it works (green) is very small.

→ Use nonlinear control (MPC) to increase the region of attraction.

Note: Explanation for the plot

LQR is linear, so the working region should first be bounded by the red lines $\|u\| = \|Kx\| = 1$ to ensure not violate the input constraint at the initial state, and the true working region is even smaller since even the initial state satisfies the constraints, as the dynamics rolled out, it might still violate the constraint. In contrast, if we use a nonlinear controller, we get much more freedom.

II. Basic Ideas of Predictive Control

- **Initial Remarks: Infinite time and finite time**

what we would like to solve is the constrained infinite time optimal control:

$$J_0^*(x(0)) = \min \sum_{k=0}^{\infty} q(x_k, u_k)$$

$$\text{s.t. } x_{k+1} = Ax_k + Bu_k,\ k = 0, \ldots, N-1$$

$$x_k \in \mathcal{X}, u_k \in \mathcal{U},\ k = 0, \ldots, N-1$$

$$x_0 = x(0)$$

where **stage cost** $q(x,u)$ is the cost of being in the state x and applying the input u

Optimizing over a trajectory provides a **tradeoff between short- and long-term benefits of actions,** such a control law has many beneficial properties, but we can't compute it: infinite number of variables.

Instead, consider the following problem: constrained **finite time** optimal control

$$J_t^*(x(t)) = \min_{U_t} p(x_{t+N}) + \sum_{k=0}^{N-1} q(x_{t+k}, u_{t+k})$$

$$\text{s.t. } x_{t+k+1} = Ax_{t+k} + Bu_{t+k},\ k = 0, \ldots, N-1$$

$$x_k \in \mathcal{X}, u_k \in \mathcal{U},\ k = 0, \ldots, N-1$$

$$x_{t+N} \in \mathcal{X}_f$$

$$x_0 = x(0)$$

Where $U_t = \{u_t \ldots, u_{t+N-1}\}$

Note: That means, since we can not solve the infinite time problem, we solve the finite time problem until time $N-1$ and try to let the system still satisfy what we want (i.e. optimize the cost while satisfying the constraints) through:

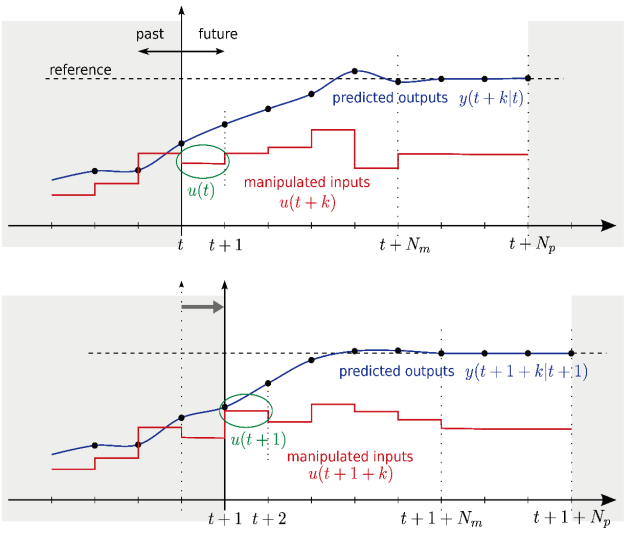
$p(x_{t+N})$ Approximation of the 'tail' of the cost and $\mathcal{X}_f$ Approximation of the 'tail' of the constraints

This idea actually lead to the online receding horizon control (MPC) formulation as shown below

- **On-line Receding Horizon Control**

1. At each sampling time, solve a CFTOC.
2. Apply the optimal input only during $[t,\ t+1]$
3. At $t+1$ solve a CFTOC over a shifted horizon based on new state measurements
4. The resulting controller is Receding Horizon Controller or Model Predictive Controller (MPC)

A clear diagram of MPC and corresponding sudo code is shown below:



1. MEASURE the state $x(t)$ at time instant t
2. OBTAIN $U_t^*(x(t))$ by solving the CFTOP optimization problem
3. IF $U_t^*(x(t)) = \emptyset$; THEN 'problem infeasible' STOP
4. APPLY the first element u_t^* of U_t^* to the system
5. WAIT for the new sampling time $t+1$, GOTO 1

Note: Need a constrained optimization solver for step 2

- **MPC Features (Pros and Cons)**

Pros:

Any model:

- linear
- nonlinear
- single/multivariable
- time delays
- constraints

Any objective

- sum of squared errors
- sum of absolute errors (i.e., integral)
- worst error over time
- economic objective

Cons:

- Computationally demanding in the general case
- May or may not be stable
- May or may not be feasible

III. Constrained Linear Optimal Control

- **Constrained Linear Optimal Control**

Definition (CLOP): The Constrained Linear Optimal Control (CLOP) is the CFTOC

$$J_0^*(x(0)) = \min_{U_0} J_0(x(0), U_0)$$

$$\begin{aligned} \text{subj. to. } & x_{k+1} = Ax_k + Bu_k,\ k = 0, \ldots, N-1 \\ & x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ k = 0, \ldots, N-1 \\ & x_N \in \mathcal{X}_f \\ & x_0 = x(0) \end{aligned}$$

with cost function of the following form:

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

where N is the time horizon, $U_0 \triangleq [u_0^{\mathrm{T}}, \ldots, u_{N-1}^{\mathrm{T}}]^{\mathrm{T}}$ is the control sequence from initial states,

$p(x_N)$, $q(x_k, u_k)$ are generally the squared Euclidian norm:

$$p(x_N) = x_N^{\mathrm{T}} P x_N,\ q(x_k, u_k) = x_k^{\mathrm{T}} Q x_k + u_k^{\mathrm{T}} R u_k$$

cost can also be changed to 1-norm or ∞-norm: $p(x_N) = \|Px_N\|_p,\ q(x_k, u_k) = \|Qx_k\|_p + \|Ru_k\|_p$ $p = 1, \infty$ and with the feasible set $\mathcal{X}$, $\mathcal{U}$, $\mathcal{X}_f$ are **polyhedral regions**.

- **Definition (Feasible set):**

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

$$\mathcal{X}_0 = \{x_0 \in \mathbb{R}^n \mid \exists (u_0, \ldots u_{N-1}) \text{ s.t. } x_k \in \mathcal{X},\ u_k \in \mathcal{U},$$
$$k = 0, \ldots, N-1,\ x_N \in \mathcal{X}_f, \text{ where } x_{k+1} = Ax_k + Bu_k\}$$

and in general the feasible set at time i is the set of states x_i at time i for which the optimal control problem is feasible:

$$\mathcal{X}_i = \{x_i \in \mathbb{R}^n \mid \exists (u_i, \ldots u_{N-1}) \text{ s.t. } x_k \in \mathcal{X},\ u_k \in \mathcal{U},$$
$$k = i, \ldots, N-1,\ x_N \in \mathcal{X}_f, \text{ where } x_{k+1} = Ax_k + Bu_k\}$$

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

Note: Usually $\mathcal{X}_{N-1} \subseteq \mathcal{X}_{N-2} \subseteq \ldots\ldots \subseteq \mathcal{X}_0$, i.e. the feasible becomes smaller as time goes by, this can be considered through reachability analysis. The other perspective is that as time goes by, the systems is more constrained since we have applied control on the system in previous steps.

- **Unconstrained Solution**

Recall that in previous lecture, we have derived that for quadratic cost (squared Euclidian norm) and no state and input constraints:

$$\{x \in \mathcal{X},\ u \in \mathcal{U}\} = \mathbb{R}^{n+m}, \mathcal{X}_f = \mathbb{R}^n$$

We have the **time-varying** linear control law for **finite time** problem:

$$u^*(k) = F_k x(k),\ k = 0, \ldots, N-1$$

If $N \to \infty$, we have the **time-invariant** linear control law for **infinite-time** problem:

$$u^*(k) = F_\infty x(k),\ k = 0, \ldots$$

Next we will compare them with the constrained solutions.

IV. Constrained Linear Optimal Control: 2-Norm

- **Problem Formulation**

$$J_0^*(x(0)) = \min_{U_0}\ x_N^{\mathrm{T}} P x_N + \sum_{k=0}^{N-1} x_k^{\mathrm{T}} Q x_k + u_k^{\mathrm{T}} R u_k$$

$$\text{subj. to. } x_{k+1} = Ax_k + Bu_k,\ k = 0, \ldots, N-1$$
$$x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ k = 0, \ldots, N-1$$
$$x_N \in \mathcal{X}_f$$
$$x_0 = x(0)$$

with $P \succeq 0, Q \succeq 0, R \succ 0, N$ is the time horizon, and the feasible set $\mathcal{X}$, $\mathcal{U}$, $\mathcal{X}_f$ are polyhedra.

- **Batch Approach by QP with substitution (AKA Shooting)**

Solution Overview

Step 1: Rewrite the cost as:

$$\begin{aligned} J_0(x(0), U_0) &= x_N^{\mathrm{T}} P x_N + \sum_{k=0}^{N-1} x_k^{\mathrm{T}} Q x_k + u_k^{\mathrm{T}} R u_k \\ &= U_0^{\mathrm{T}} H U_0 + 2x(0)^{\mathrm{T}} F U_0 + x(0)^{\mathrm{T}} Y x(0) \\ &= \begin{bmatrix} U_0^{\mathrm{T}} & x(0)^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} H & F^{\mathrm{T}} \\ F & Y \end{bmatrix} \begin{bmatrix} U_0^{\mathrm{T}} & x(0)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \end{aligned}$$

Note: $\begin{bmatrix} H & F^{\mathrm{T}} \\ F & Y \end{bmatrix} \succeq 0$ since $J_0(x(0), U_0) \geq 0$ by assumption.

Step 2: Rewrite the constraints compactly as

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

Step 3: Rewrite the optimal control problem as the standard QP form:

$$J_0^*(x(0)) = \min_{U_0} \begin{bmatrix} U_0^{\mathrm{T}} & x(0)^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} H & F^{\mathrm{T}} \\ F & Y \end{bmatrix} \begin{bmatrix} U_0^{\mathrm{T}} & x(0)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$$

$$\text{subj. to. } G_0 U_0 \leq w_0 + E_0 x(0)$$

Thus, for a given $x(0)$, U_0^* can be found via a QP solver.

Solution Detail: Converting Constraints

Note that the feasible set $\mathcal{X}$, $\mathcal{U}$, $\mathcal{X}_f$ are all polyhedra., in other words, they are given by:

$$\mathcal{X} = \{x \mid A_x x \leq b_x\} \quad \mathcal{U} = \{u \mid A_u u \leq b_u\} \quad \mathcal{X}_f = \{x \mid A_f x \leq b_f\}$$

So the G_0, E_0 and w_0 in the above formulation is constructed as below:

$$G_0 = \begin{bmatrix} \underbrace{\begin{matrix} A_u & 0 & \dots & 0 \\ 0 & A_u & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_u \end{matrix}}_{\mathcal{U}} \\ \underbrace{\begin{matrix} 0 & 0 & \dots & 0 \\ A_x B & 0 & \dots & 0 \\ A_x AB & A_x B & \ddots & 0 \\ \vdots & \vdots & \dots & \vdots \\ A_x A^{N-2} B & A_x A^{N-3} B & \dots & 0 \end{matrix}}_{\mathcal{X}} \\ \underbrace{\begin{matrix} A_f A^{N-1} B & A_f A^{N-2} B & \dots & A_f B \end{matrix}}_{\mathcal{X}_f} \end{bmatrix}, \quad E_0 = \begin{bmatrix} \underbrace{\begin{matrix} 0 \\ 0 \\ \vdots \\ 0 \end{matrix}}_{\mathcal{U}} \\ \underbrace{\begin{matrix} -A_x \\ -A_x A \\ -A_x A^2 \\ \vdots \\ -A_x A^{N-1} \end{matrix}}_{\mathcal{X}} \\ \underbrace{\begin{matrix} -A_f A^N \end{matrix}}_{\mathcal{X}_f} \end{bmatrix}, \quad w_0 = \begin{bmatrix} \underbrace{\begin{matrix} b_u \\ b_u \\ \vdots \\ b_u \end{matrix}}_{\mathcal{U}} \\ \underbrace{\begin{matrix} b_x \\ b_x \\ b_x \\ \vdots \\ b_x \end{matrix}}_{\mathcal{X}} \\ \underbrace{\begin{matrix} b_f \end{matrix}}_{\mathcal{X}_f} \end{bmatrix}$$

Note:

1. The definition of H, F and Y should refer to last lecture's notes, where we do dynamics rollout and the system is expressed as:

$$\mathcal{X} = \mathcal{S}^x x(0) + \mathcal{S}^u U_0$$

With $\bar{Q} = \text{blockdiag}(Q, \ldots, Q, P)$ and $\bar{R} = \text{blockdiag}(R, \ldots, R)$, we get $H = (\mathcal{S}^u)^{\mathrm{T}} \bar{Q} \mathcal{S}^u + \bar{R}$, $F = (\mathcal{S}^x)^{\mathrm{T}} \bar{Q} \mathcal{S}^u$ and $Y = (\mathcal{S}^x)^{\mathrm{T}} \bar{Q} \mathcal{S}^x$

2. This method is also called shooting method because the states $x(k)$ are found by taking input and explicit integrated the dynamics (i.e. simulation).

- **Batch Approach by QP without substitution (AKA Collocation)**

Instead of rollout the system dynamics, we can also explicitly keep the equality constraints but need to introduce additional variable.

Solution Overview

Step 1: Introduce and define the variable: $z = [x_1^{\mathrm{T}} \ \ldots \ x_N^{\mathrm{T}} \ u_0^{\mathrm{T}} \ \ldots \ u_{N-1}^{\mathrm{T}}]^{\mathrm{T}}$

Step 2: Rewrite the cost as:

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

$$= [z^{\mathrm{T}} \ \ x(0)^{\mathrm{T}}] \begin{bmatrix} \bar{H} & 0 \\ 0 & Q \end{bmatrix} [z^{\mathrm{T}} \ \ x(0)^{\mathrm{T}}]^{\mathrm{T}}$$

Step 3: Rewrite the constraints compactly as

$$G_{0,\text{in}} z \le w_{0,\text{in}} + E_{0,\text{in}} x(0)$$

$$G_{0,\text{eq}} z = E_{0,\text{eq}} x(0)$$

Step 4: Rewrite the optimal control problem as the standard QP form:

$$J_0^*(x(0)) = \min_z \ [z^{\mathrm{T}} \ \ x(0)^{\mathrm{T}}] \begin{bmatrix} \bar{H} & 0 \\ 0 & Q \end{bmatrix} [z^{\mathrm{T}} \ \ x(0)^{\mathrm{T}}]^{\mathrm{T}}$$

$$\text{subj. to. } G_{0,\text{in}} z \le w_{0,\text{in}} + E_{0,\text{in}} x(0)$$

$$G_{0,\text{eq}} z = E_{0,\text{eq}} x(0)$$

Thus, for a given $x(0)$, z^* (and therefore U_0^*) can be found via a QP solver.

Solution Detail: Converting Objective and Constraints

From the cost and definition of z, it is easy to derive the $\bar{H}$ in the objective matrix as:

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

The equality constraints (system dynamics) are converted as below:

$$x_{k+1}=Ax_k+Bu_k \Rightarrow G_{0,\text{eq}}=\left[\begin{array}{ccccc:cccc} I & & & & & -B & & & \\ -A & I & & & & & -B & & \\ & -A & I & & & & & -B & \\ & & \ddots & \ddots & & & & & -B \\ & & & -A & I & & & & \end{array}\right],\ E_{0,\text{eq}}=\begin{bmatrix} A \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Solution Detail: Converting Constraints

Note that the feasible set $\mathcal{X},\ \mathcal{U},\ \mathcal{X}_f$ are all polyhedra., in other words, they are given by:

$$\mathcal{X}=\{x \mid A_x x \le b_x\} \quad \mathcal{U}=\{u \mid A_u u \le b_u\} \quad \mathcal{X}_f=\{x \mid A_f x \le b_f\}$$

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

$$G_{0,\text{in}}=\left[\begin{array}{ccccc:ccccc} 0 & & & & & 0 & & & & \\ A_x & & & & & 0 & & & & \\ & \ddots & & & & 0 & & & & \\ & & A_x & & & & 0 & & & \\ & & & A_f & & & & 0 & & \\ \hdashline 0 & & & & & A_u & & & & \\ 0 & & & & & A_u & & & & \\ & 0 & & & & & \ddots & & & \\ & & 0 & & & & & A_u & & \\ & & & 0 & & & & & A_u & \\ & & & & 0 & & & & & A_u \end{array}\right] \quad w_{0,\text{in}}=\left[\begin{array}{c} b_x \\ b_x \\ \vdots \\ b_x \\ b_f \\ \hdashline b_u \\ b_u \\ \vdots \\ b_u \\ b_u \end{array}\right] \quad E_{0,\text{in}}=[-A_x^{\mathrm{T}} \ \ 0 \ \ \cdots \ \ 0]^{\mathrm{T}}$$

Note:

1. Even though this method has much more variables and the matrix blocks are larger than the previous methods, in practice it is solved much faster. Again, it is because the substitution method requires to compute the power of the system matrix, which is not numerically ideal.
2. This method is also called collocation, which introduce the states also as decision variable and the dynamics is handled through an implicit integration way.

- **2-Norm State Feedback Solution for CFTOC via Batch Approach**

In this section, we show that when formulating CFTOC as a multiparametric QP using substitution method, we can explicitly get a control law that depend on the initial condition $x(0)$, as described in the following steps:

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

$$\hat{J}(x(0))=\min_z \ z^{\mathrm{T}}Hz$$

$$\text{subj. to. } G_0 z \le w_0+S_0 x(0)$$

Where $S_0 \triangleq E_0+G_0H^{-1}F^{\mathrm{T}}$ and $\hat{J}^*(x(0))=J_0^*(x(0))-x(0)^{\mathrm{T}}(Y-FH^{-1}F^{\mathrm{T}})x(0)$. 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 $z^*(x(0))$

Note:

1. Why this problem is called multiparametric quadratic program?
   It is because in the transformed problem, we can see that the decision variable z is subjected to the **constraint** $G_0 z \leq w_0 + S_0 x(0)$ **which is dependent on multiparameter (since states can be multi -dimensional)** $x(0)$.
2. More detailed introduction of multiparametric programming would be introduced in lecture 11.

Main result:

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 $U_0^* = z(x(0)) - H^{-1}F^{\mathrm{T}}x(0)$
2. The first component of the multiparametric solution has the form:
   $$u^*(0) = f_0(x(0)),\ \forall x(0) \in \mathcal{X}_0$$
   And $f_0 : \mathbb{R}^n \to \mathbb{R}^m$ is continuous and PieceWise Affine on Polyhedra
   $$f_0(x) = F_0^i x + g_0^i \text{ if } x \in CR_0^i,\ i = 1, \ldots, N_0^r$$
3. The polyhedral sets $CR_0^i = \{x \in \mathbb{R}^n \mid H_0^i x \leq K_0^i\},\ i = 0 \cdots, N_0^r$ 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.

Note:

1. The above results are saying: what we get is the control action U_0 as a function of $x(0)$, so it provides a state feedback control law $u^*(k) = f_k(x(k))$ for $k = 0$ and also provides the open-loop optimal control $u^*(k, x(0))$ as a function of initial state for $k = 1, \ldots, N-1$.
2. Then how can we calculate the state feedback solution for other $k = 1, \ldots, N-1$? It can be calculated by solving the CFTOC over the shrinking horizon $k \to N$, $k = 1, \ldots, N$, which requires solving multiple mp-QPs, similar idea is mentioned in discussions in the previous lecture on unconstrained control. For examples, refer to homework 6, Ex 3, though it is for ∞ norm case.
3. We can't get state feedback solution through dynamic programming (recursive approach). It is too complicated, see the discussion below when introducing the $1/\infty$ norm case

Example: simple double Integrator.

Consider the double integrator:

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

Subject to constraints:

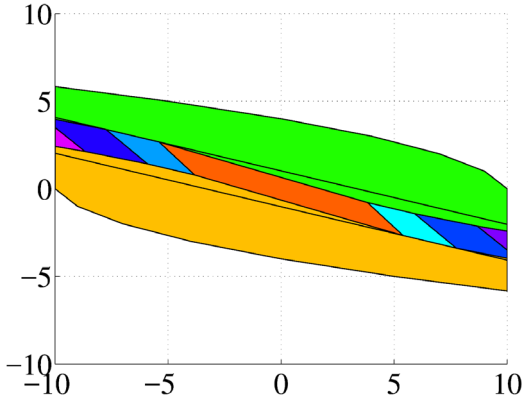
$$-1 \leq u(k) \leq 1,\ k = 0, \ldots, 5$$

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

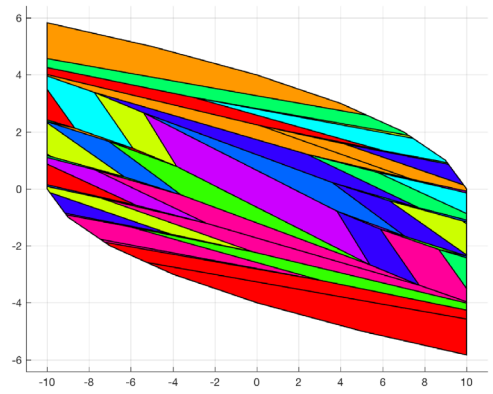
Compute the **state feedback** optimal controller $u^*(0)\ (x(0))$ solving the CFTOC problem with:

$N=6,\ Q=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix},\ R=0.1,\ \mathcal{X}_f=\mathbb{R}^2$ and let P be a solution of ARE

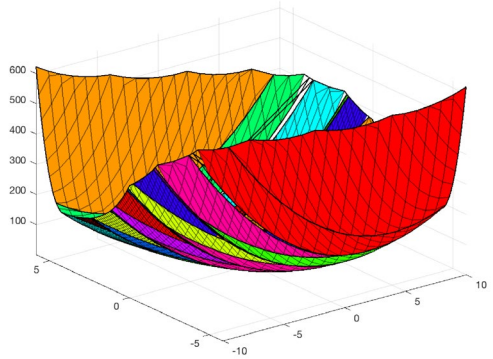
We can get the solution as the following figures shows:



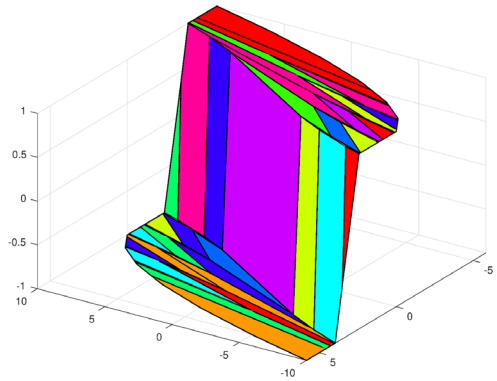
Merged Partition of the state space for the affine control law $u^*(0)\ (N_0^r=13)$



Original Partition of the state space for the affine control law $u^*(0)\ (N_0^r=61)$



Value function for the affine control law $u^*(0)\ (N_0^r=61)$



Optimal control input for the affine control law $u^*(0)\ (N_0^r=61)$

Note: each region corresponds to one kind of constraints (combination) to be active.

V. Constrained Linear Optimal Control: 1-Norm and inf-Norm

- **Problem Formulation**

Piece-wise linear cost function

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

With $p=1$ or $p=\infty$, P,Q,R are full column rank matrices and the CFTOC is:

$$J_0^*(x(0))=\min_{U_0}\ J_0(x(0),U_0)$$

$$\begin{aligned}\text{subj. to. }\ & x_{k+1}=Ax_k+Bu_k,\ k=0,\ \ldots,N-1\\ & x_k\in\mathcal{X},\ u_k\in\mathcal{U},\ k=0,\ \ldots,N-1\\ & x_N\in\mathcal{X}_f\\ & x_0=x(0)\end{aligned}$$

Note: in 2-norm, we require $P\succeq 0, Q\succeq 0, R\succ 0$, now we require them to all be full column rank.

- **Batch Approach by LP with substitution for ∞-norm**

Recall the optimization lecture, the ∞-norm minimization problem can be constructed as a LP with introducing new slack variables, here after substitution the dynamics, we can rewrite the CFTOC as:

$$\min_{z_0} = \varepsilon_0^x + \ldots + \varepsilon_N^x + \varepsilon_0^u + \ldots + \varepsilon_{N-1}^u$$

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

- **1-Norm/∞-Norm State Feedback Solution for CFTOC via Batch Approach**

By stacking up the decision variable in the problem formulation, we can transform the problem into a standard LP:

$$\min_{z_0} \; c_0^{\mathrm{T}} z_0$$

$$\text{subj. to. } \bar{G}_0 z_0 \le \bar{w}_0 + \bar{S}_0 x(0)$$

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

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

This is a multiparametric linear program (mp-LP). For a given $x(0)$, U_0^* can be obtained via an LP solver, and 1-norm case is similar but would introduce more slack variable and the problem would be relatively large (but it still can be solved quickly enough!)

Main result:

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 $u_0^* = [0 \;\; \ldots \;\; 0 \;\; I_m \;\; 0 \;\; \ldots \;\; 0] z_0^*(x(0))$ of the multiparametric solution has the following form:

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

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

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

3. The polyhedral sets $CR_0^i = \{x \in \mathbb{R}^n \mid H_0^i x \leq K_0^i\},\ i = 0 \cdots, N_0^r$ are a partition of the feasible polyhedron $\mathcal{X}_0$
4. In case of multiple optimizers, a PieceWise Affine control law exists. (Recall that if the objective in LP is parallel to the constraint, then it might be infinitely mainly solutions)
5. The value function $J_0^*(x(0))$ is convex and piecewise linear on polyhedra.

Note:

Similar to the 2-norm case, state feedback solution for other $k = 1, \ldots, N-1$ can be calculated by solving the CFTOC over the shrinking horizon $k \to N$, $k = 1, \ldots, N$, which requires solving multiple mp-LPs now.

- **State Feedback Solution via Recursive Approach (Dynamic Programming): Discussions**

In the above section when we talked about the state feedback solution for CFTOC, we use batch approach, formulate the mp-QP/LP, and the state-feedback solution is derived from solving a series of mp-QP/LP over the shrinking horizon. However, recall the lecture for unconstrained optimal control, for unconstrained case, we actually do not use batch approach to compute the state-feedback because the recursive approach (dynamic programming) is much more convenient and computation efficient (the dimension for each sub-problem is low).

The big problem is when dynamic programming is used in the unconstrained cases, we **know the form of the value function, they are usually quadratic or linear, so we can solve a single QP/LP from end to start iteratively**. However, for constrained ones, we make the following statements.

2-Norm State Feedback Solution via Recursive Approach

Consider the dynamic programming formulation of the CFTOC:

$$J_j^*(x_j) = \min_{u_j} \ x_j^{\mathrm{T}} Q x_j + u_j^{\mathrm{T}} R u_j + J_{j+1}(Ax_j + Bu_j)$$

$$\text{subj. to.} \quad x_j \in \mathcal{X}, u_j \in \mathcal{U}$$

$$Ax_j + Bu_j \in \mathcal{X}_{j+1}$$

For $j = 0, \ldots, N-1$, with boundary conditions

$$J_N^*(x_N) = x_N^{\mathrm{T}} P x_N, \quad \mathcal{X}_N = \mathcal{X}_f$$

Note that according to our result mentioned above, the value function $J_{j+1}^*(Ax_j + Bu_j)$ is **piecewise quadratic (not quadratic!)** for $j < N-1$. Therefore, each single backward step described above is **not simply an mp-QP**. In other words, to derive the 2-norm state feedback solution through dynamic programming is hard and the computational advantage of the iterative over the batch approach is not obvious. Generally this method is not recommended.

However, unlike the 2-norm case, the dynamic programming formulation of $1/\infty$-Norm CFTOC can be solved explicitly.

1-Norm/∞-Norm State Feedback Solution via Recursive Approach

Consider the dynamic programming formulation of the CFTOC:

$$J_j^*(x_j) = \min_{u_j} \ \|Qx_j\|_p + \|Ru_j\|_p + J_{j+1}(Ax_j + Bu_j)$$

$$\text{subj. to.} \quad x_j \in \mathcal{X}, u_j \in \mathcal{U}$$

$$Ax_j + Bu_j \in \mathcal{X}_{j+1}$$

For $j = 0, \ldots, N-1$, with boundary conditions

$$J_N^*(x_N) = \|Px_N\|_p, \quad \mathcal{X}_N = \mathcal{X}_f$$

And the norm coefficient $p = 1/\infty$

Theorem: The state feedback piecewise affine solution of CFTOC with $1/\infty$-Norm cost can obtained by solving the above dynamic programming via N mp-LPs.

For the complete proof, refer to Section 11.4.3 of the book. A rough and direct intuition on the theorem can be concluded as follows:

For $j = N-1$, one can always calculate the J_{N-1}^* through a single mp-LP. Note that according to our result mentioned above, the value function J_{N-1}^* is a convex and piecewise affine function of x_{N-1}, the corresponding optimizer u_{N-1}^* is piecewise affine and continuous, and the feasible set $\mathcal{X}_{N-1}$ is a polyhedron. In fact, the convex piecewise affine J_{N-1}^* can be described as:

$$J_{N-1}^*(x_{N-1}) = \max_{i=1,\ldots,n_{N-1}} \{c_i^{\mathrm{T}} x_{N-1} + d_i\}$$

where n_{N-1} is the number of affine components comprising the value function. Therefore, consider $x_{N-1} = Ax_{N-2} + Bu_{N-2}$, J_{N-1}^* becomes:

$$J_{N-1}^*(x_{N-1}) = \max_{i=1,\ldots,n_{N-1}} \{c_i^{\mathrm{T}}(Ax_{N-2} + Bu_{N-2}) + d_i\}$$

For $j = N-2$, the problem becomes:

$$J_{N-2}^*(x_j) = \min_{u_{N-2}} \ \|Qx_{N-2}\|_p + \|Ru_{N-2}\|_p + \max_{i=1,\ldots,n_{N-1}} \{c_i^{\mathrm{T}}(Ax_{N-2} + Bu_{N-2}) + d_i\}$$

$$\text{subj. to.} \quad x_{N-2} \in \mathcal{X}, u_{N-2} \in \mathcal{U}$$

$$Ax_{N-2} + Bu_{N-2} \in \mathcal{X}_{N-1}$$

Recall the lecture on convex optimization, the convex piecewise affine minimization problem can be reformulate into a LP by introducing the slack variables. So this problem is still an mp-LP problem! Similar process propagates all through $j = N-3 \ldots j = 0$, hence the theorem holds.

Note: To sum up, considering recursive approach (dynamic programming), the most critical difference between 2-norm and $1/\infty$-norm case is the form of the value function. Both of the cases can finish the first tail calculation ($j = N-1$) from a simple mp-QP or mp-LP. However, starting from $j < N-1$, i.e. $j = N-2$, since the value function is piecewise quadratic for 2-norm case, the successive sub-problem can no longer be formulate as an mp-QP, while for 1-norm case, the successive sub-problem

is still an mp-LP since the value function is piecewise affine and piecewise affine minimization can be reformulated into a LP. Homework 7, Ex 3 gives an example of CFTOC solved by recursive approach.

VI. Receding Horizon Control (Model Predictive Control) Notation

As is mentioned in the last lecture, receding horizon control (model predictive control) takes the middle ground of batch approach and recursively approach and is a way to effectively derive the state-feedback solution of optimal control problem. It is especially powerful in constrained cases. All the rest of this course would focus on MPC theories and applications, here we first introduce the unified notations and problem formulations to simplify further discussions.

- **RHC (MPC) Notation: General Formulation**

In receding horizon control, we assume linear dynamics with domain:

$$x(t+1)=Ax(t)+Bu(t)$$
$$y(t)=Cx(t)$$
$$x(t)\in\mathcal{X},\ u(t)\in\mathcal{U},\ \forall t\geq 0$$

And we solve the receding-horizon-fashion CFTOC:

$$J_t^*(x(t))=\min_{U_{t\to t+N|t}}\ p(x_{t+N|t})+\sum_{k=0}^{N-1}q(x_{t+k|t},u_{t+k|t})$$
$$\text{subj. to}\ \ x_{t+k+1|t}=Ax_{t+k|t}+Bu_{t+k|t},\ k=0,\ldots,N-1$$
$$x_{t+k|t}\in\mathcal{X},\ u_{t+k|t}\in\mathcal{U},\ k=0,\ldots,N-1$$
$$x_{t+N|t}\in\mathcal{X}_f$$
$$x_{t|t}=x(t)$$

with $U_{t\to t+N|t}=\{u_{t|t},\ldots,u_{t+N-1|t}\}$. In this notation:

1. $x(t)$ is the state of the system at time t
2. $x_{t+k|t}$ is the state of the model at time $t+k$, **predicted at time t, obtained by starting from the current state $x_{t|t}=x(t)$ and applying the input sequence $u_{t|t},\ldots,u_{t+k-1|t}$ to the system model $x_{t+1|t}=Ax_{t|t}+Bu_{t|t}$**

   Note: For example, $x_{3|1}$ represents the predicted state at time 3 when the prediction is done at time $t=1$ starting from the current state $x(1)$, It is different, in general, from $x_{3|2}$ which is the predicted state at time 3 when the prediction is done at time $t=2$ starting from the state $x(2)$.
3. Similarly $u_{t+k|t}$ is read as "the input u at time $t+k$ computed at time t"
4. Let $U_{t\to t+N|t}^*=\{u_{t|t}^*,\ldots,u_{t+N-1|t}^*\}$ be the optimal solution. **The first element of $U_{t\to t+N|t}^*$ is applied to system, i.e.**

   $$u(t)=u_{t|t}^*(x(t))$$
5. The CFTOC problem is reformulated and solved at time $t+1$, based on the new state $x_{t+1|t+1}=x(t+1)$

To sum up, the receding horizon law is:

$$f_t(x(t)) = u^*_{t|t}(x(t))$$

And the closed loop system is:

$$x(t+1) = Ax(t) + Bf_t(x(t)) \triangleq f_{cl}(x(t)),\ t \geq 0$$

- **RHC (MPC) Notation: Time-invariant Systems**

As **the system, the constraints and the cost function (i.e. the weight matrix) are time-invariant**, the solution $f_t(x(t))$ becomes a time-invariant function of the initial state $x(t)$. Thus, we can simplify the notation as:

$$J^*_t(x(t)) = \min_{U_0}\ p(x_N) + \sum_{k=0}^{N-1} q(x_k, u_k)$$

$$\begin{aligned} \text{subj. to.}\ \ & x_{k+1} = Ax_k + Bu_k,\ k = 0, \ldots, N-1 \\ & x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ k = 0, \ldots, N-1 \\ & x_N \in \mathcal{X}_f \\ & x_0 = x(t) \end{aligned}$$

where $U_0 = \{u_0, \ldots, u_{N-1}\}$

The control law and closed loop system are time-invariant as well, and we write $f_0(x_0)$ for $f_t(x(t))$

Note:

1. This basically means for time-invariant systems every horizon you span can be treated in the same way. It is because the system, constraints and weight matrices remain unchanged. Therefore, we don't have to specify the time we make prediction, every calculation $t \to t+N$ is exactly $0 \to N$, the only difference is that the initial condition is different $x_0 = x(t)$.
2. Compare this MPC problem formulation and the CFTOC problem formulation, the only difference is that MPC problem is solved for $x_0 = x(t), t \geq 0$ rather than for $x_0 = x(0)$. In other words, recall our methods to generate state feedback solution of CFTOC using batch approach: we need to consider problems with the shrinking horizon $k \to N,\ k = 1, 2, \ldots N-1$ (tail fixed). However, in MPC formulation we keep spanning the same horizon length forward: $1 \to N+1$, $2 \to N+2 \ldots$ It is because MPC essentially want to get the infinite horizon suboptimal controller by repeatedly solving finite time problems, so there is no reason it has to stick to a fixed tail.
3. The most important thing about MPC is that **the feedback policy is implicitly included in the multi-parametric programming formulation with receding horizon fashion.** Recall what we have mentioned before: what we get from the multi-parametric programming provides a state feedback control law $u^*(k) = f_k(x(k))$ for $k = 0$. Since we are doing receding horizon control, every x_k is exactly the new $x(0)$! So in the end, it would give us a feedback controller (policy) for every time k.