ME 234(b): Constrained Optimal Control

Anushri Dixit

Spring 2022

Slides adapted from CMS 159 (by U. Rosolia) and Berkeley ME231 (by F. Borrelli, M. Morari, C. Jones)

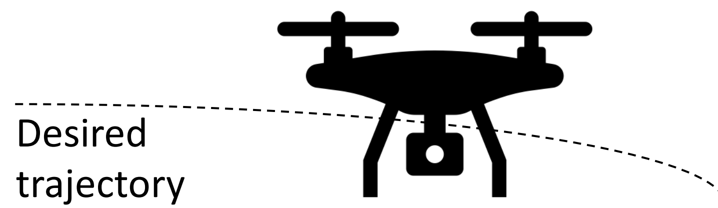
Caltech

Recap: Finite-horizon unconstrained OCP

$$\min_{U_0} \quad x_N^T P x_N + \sum_{i=0}^{N-1} \left(x_i^T Q x_i + u_i^T R u_i \right)$$

$$\text{s.t.} \quad x_{k+1} = A x_k + B u_k$$

$$x_0 = x(0)$$



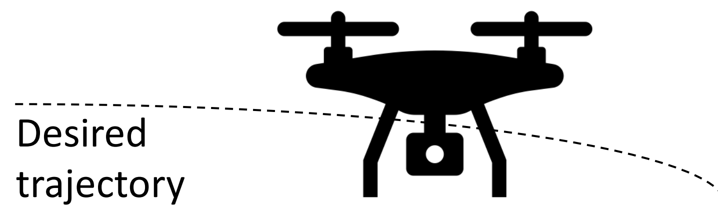
Caltech

Recap: Finite-horizon unconstrained OCP

$$\min_{U_0} \quad x_N^T P x_N + \sum_{i=0}^{N-1} \left(x_i^T Q x_i + u_i^T R u_i \right)$$

$$\text{s.t.} \quad x_{k+1} = A x_k + B u_k$$

$$x_0 = x(0)$$



The solution of the **Batch Approach** is,

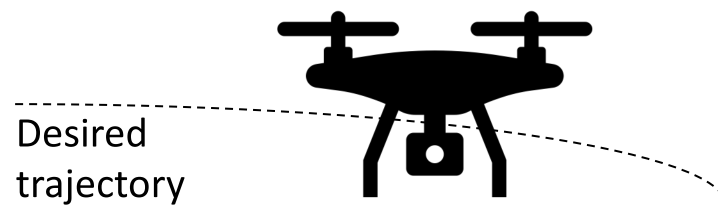
$$U_0^*\big(x(0)\big) = -(\mathcal{S}_u^T \bar{Q} \mathcal{S}_u + \bar{R})^{-1} \mathcal{S}_u^T \bar{Q} \mathcal{S}_x x(0)$$

We obtained this by substituting all the equality constraints into the cost and then solving the unconstrained minimization by taking the gradient.

Caltech

Recap: Finite-horizon unconstrained OCP

$$
\begin{aligned}
\min_{U_0} \quad & x_N^T P x_N + \sum_{i=0}^{N-1} \left(x_i^T Q x_i + u_i^T R u_i \right) \\
\text{s.t.} \quad & x_{k+1} = A x_k + B u_k \\
& x_0 = x(0)
\end{aligned}
$$



The solution of the **Recursive Approach,**

$$
\begin{aligned}
u^*(k) &= -(B^T P_{k+1} B + R)^{-1} B^T P_{k+1} A x(k) \\
&\triangleq F_k x(k),
\end{aligned}
$$

$$
\begin{aligned}
P_k &= A^T P_{k+1} A + Q - A^T P_{k+1} B (B^T P_{k+1} B + R)^{-1} B^T P_{k+1} A \\
P_N &= P
\end{aligned}
$$

We used the Principle of Optimality to solve a one step solution backwards.

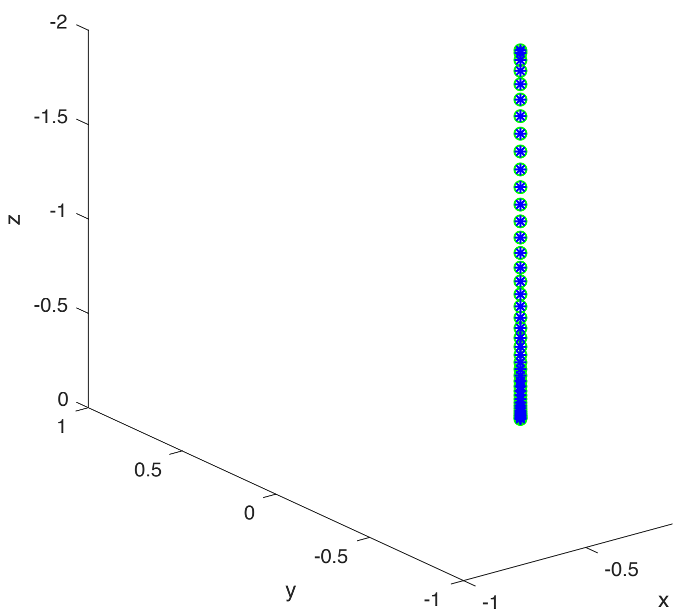
Let's take a look at some code for the drone example.

Caltech

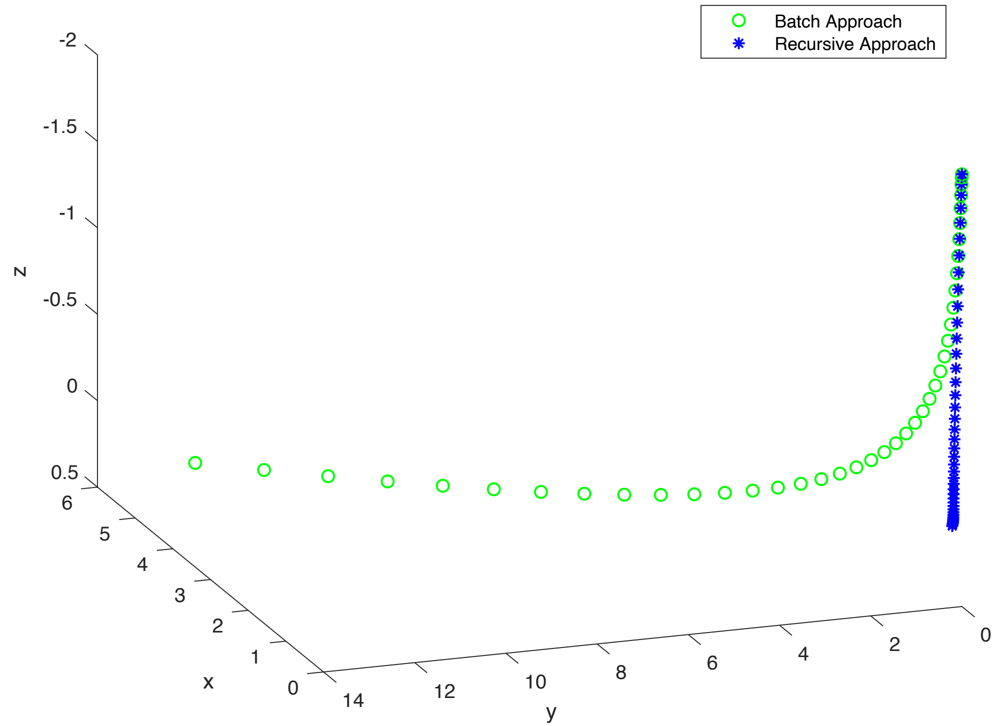
Recap: Finite-horizon unconstrained OCP

Let's look at some code for the drone example we discussed in the last lecture.

Recall that our task is to converge to the origin.



No disturbances



In presence of disturbances

Caltech

Aside: Infinite-horizon unconstrained OCP

Let's briefly consider the case when N→∞

$$J^*_\infty(x(0)) = \min_{u_0, u_1, \ldots} \sum_{k=0}^{\infty} \left(x_k^T Q x_k + u_k^T R u_k\right)$$

The solution to the Batch Approach becomes intractable as N grows.

However, the Recursive Approach can be solved to convergence by finding the matrix P_∞, such that,

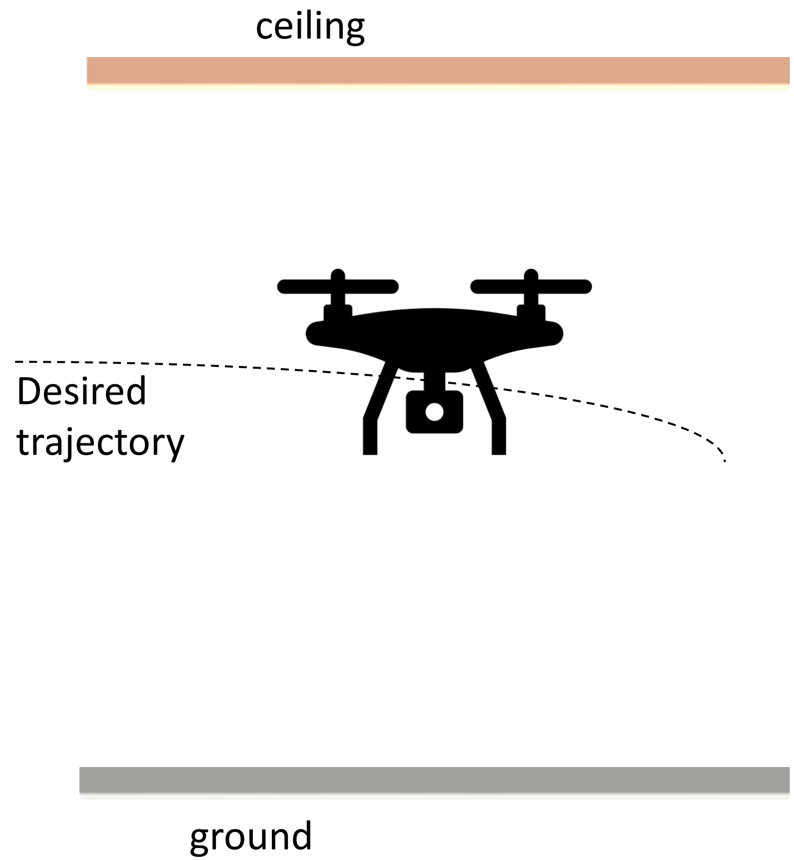
$$P_\infty = A^T P_\infty A + Q - A^T P_\infty B(B^T P_\infty B + R)^{-1} B^T P_\infty A$$

The above equation is the Algebraic Riccati Equation (ARE) and the corresponding optimal controller is asymptotically stabilizing.

Caltech

Adding constraints

$$
\begin{aligned}
\min_{U_0} \quad & x_N^T P x_N + \sum_{i=0}^{N-1} \left(x_i^T Q x_i + u_i^T R u_i \right) \\
\text{s.t.} \quad & x_{k+1} = A x_k + B u_k \\
& x_k \in \mathcal{X}, u_k \in \mathcal{U} \\
& x_N \in \mathcal{X}_F \\
& x_0 = x(0)
\end{aligned}
$$



where, the state constraint set is

$$\mathcal{X} = \{x \in \mathbb{R}^{n_x} : F_x x \leq b_x\}$$

the control constraint set is

$$\mathcal{U} = \{u \in \mathbb{R}^{n_u} : F_u u \leq b_u\}$$

and the terminal set is given by

$$\mathcal{X}_F = \{x \in \mathbb{R}^{n_x} : F_f x \leq b_f\}$$

Caltech

Adding constraints: How to solve?

$$
\begin{aligned}
\min_{U_0} \quad & x_N^T P x_N + \sum_{i=0}^{N-1} \left(x_i^T Q x_i + u_i^T R u_i \right) \\
\text{s.t.} \quad & x_{k+1} = A x_k + B u_k \\
& x_k \in \mathcal{X}, u_k \in \mathcal{U} \\
& x_N \in \mathcal{X}_F \\
& x_0 = x(0)
\end{aligned}
$$

We've looked at two approaches to solve unconstrained OCP: batch and recursive approaches.

The Dynamic Programming based approach (recursive) is hard to solve with constraints → involves gridding the allowable sets and solving for the cost-to-go for each point in the grid (expensive).

The Batch Approach is far easier to adapt to a constrained setting.

1. Batch Approach: With substitution

Recall: We write the dynamics constraints (equality constraints) in terms of the initial condition and the control input as,

$$\underbrace{\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}}_{X_0} = \underbrace{\begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}}_{\mathcal{S}_x} x(0) + \underbrace{\begin{bmatrix} 0 & 0 & \dots & 0 \\ B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \dots & B \end{bmatrix}}_{\mathcal{S}_u} \underbrace{\begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{bmatrix}}_{U_0}$$

Hence, all the dynamics constraints can be written in batch form as,

$$X_0 = \mathcal{S}_x x(0) + \mathcal{S}_u U_0$$

Also let, $\bar{Q} = \text{blkdiag}(\underbrace{Q, Q, \dots, Q}_{\text{N times}}, P)$ and $\bar{R} = \text{blkdiag}(\underbrace{R, R, \dots, R}_{\text{N times}})$

Caltech

1. Batch Approach: With substitution

Hence, we can substitute the equality constraints into the cost to get,

$$\min_{U_0} \quad (\mathcal{S}_x x(0) + \mathcal{S}_u U_0)^T \bar{Q} (\mathcal{S}_x x(0) + \mathcal{S}_u U_0) + U_0^T \bar{R} U_0$$

We now also have the following inequality constraints,

$$x_k \in \mathcal{X} = \{x \in \mathbb{R}^{n_x} : F_x x \leq b_x\}$$

$$u_k \in \mathcal{U} = \{u \in \mathbb{R}^{n_u} : F_u u \leq b_u\}$$

$$x_N \in \mathcal{X}_F = \{x \in \mathbb{R}^{n_x} : F_f x \leq b_f\}$$

Let's write all of these in terms of the initial condition and the control input.

Caltech

1. Batch Approach: With substitution

Like the equality constraints, we can write the inequality constraints in terms of the initial condition and the control inputs as,

$$\underbrace{\begin{bmatrix} F_u & 0 & \dots & 0 \\ 0 & F_u & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & F_u \\ 0 & 0 & \dots & 0 \\ F_x B & 0 & \dots & 0 \\ F_x AB & F_x B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ F_x A^{N-2} B & F_x A^{N-3} B & \dots & 0 \\ F_f A^{N-1} B & F_f A^{N-2} B & \dots & F_f B \end{bmatrix}}_{G_0} \underbrace{\begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}}_{U_0} \leq \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -F_x \\ -F_x A \\ -F_x A^2 \\ \vdots \\ -F_x A^{N-1} \\ -F_f A^N \end{bmatrix}}_{E_0} x(0) + \underbrace{\begin{bmatrix} b_u \\ b_u \\ \vdots \\ b_u \\ b_x \\ b_x \\ b_x \\ \vdots \\ b_x \\ b_f \end{bmatrix}}_{w_0}$$

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

Caltech

1. Batch Approach: With substitution

Hence, the original optimization problem given by

$$\begin{aligned} \min_{U_0} \quad & x_N^T P x_N + \sum_{i=0}^{N-1} \left(x_i^T Q x_i + u_i^T R u_i \right) \\ \text{s.t.} \quad & x_{k+1} = A x_k + B u_k \\ & x_k \in \mathcal{X}, u_k \in \mathcal{U} \\ & x_N \in \mathcal{X}_F \\ & x_0 = x(0) \end{aligned}$$

can instead be written as,

$$\begin{aligned} J_0^*(x(0)) = \min_{U_0} \quad & (\mathcal{S}_x x(0) + \mathcal{S}_u U_0)^T \bar{Q} (\mathcal{S}_x x(0) + \mathcal{S}_u U_0) + U_0^T \bar{R} U_0 \\ \text{s.t.} \quad & G_0 U_0 \leq E_0 x(0) + w_0 \end{aligned}$$

The above problem can be solved using off-the-shelf optimization solvers like quadprog (in MATLAB), Gurobi, Mosek, etc.

Caltech

2. Batch Approach: Without substitution

Let's look at another way to solve the same optimization problem.

We want to rewrite the optimization problem in terms of U_0, X_0.

- Equality constraints

$$\underbrace{\begin{bmatrix} I & 0 & \dots & 0 & -B & 0 & \dots & 0 \\ -A & I & \dots & 0 & 0 & -B & \dots & 0 \\ & \ddots & \ddots & & & & \ddots & \\ 0 & \dots & -A & I & 0 & \dots & 0 & -B \end{bmatrix}}_{G_{0,\text{eq}}} \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_N \\ u_0 \\ \vdots \\ u_{N-1} \end{bmatrix}}_{\begin{bmatrix} X_0^T & U_0^T \end{bmatrix}^T} = \underbrace{\begin{bmatrix} A \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{E_{0,\text{eq}}} x(0)$$

Caltech

2. Batch Approach: Without substitution

We want to rewrite the optimization problem in terms of U_0, X_0 .

- Inequality constraints

$$
\underbrace{\begin{bmatrix}
0 & 0 & \cdots & 0 & 0 & \cdots & & 0 \\
F_x & 0 & \cdots & 0 & \ddots & & & \vdots \\
0 & F_x & \cdots & 0 & \ddots & \ddots & & 0 \\
0 & 0 & \ddots & 0 & & \cdots & & 0 \\
\vdots & \vdots & \ddots & F_x & 0 & \cdots & & \\
 & & & 0 & F_f & & & \vdots \\
 & & & & 0 & F_u & & \\
 & & & & & \ddots & \ddots & \\
0 & 0 & & & \cdots & & 0 & F_u
\end{bmatrix}}_{G_{0,\text{ineq}}}
\underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_N \\ u_0 \\ \vdots \\ u_{N-1} \end{bmatrix}}_{\begin{bmatrix} X_0^T & U_0^T \end{bmatrix}^T}
\leq
\underbrace{\begin{bmatrix} -F_x \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{E_{0,\text{ineq}}} x(0) +
\underbrace{\begin{bmatrix} b_x \\ b_x \\ \vdots \\ b_x \\ b_f \\ b_u \\ \vdots \\ b_u \end{bmatrix}}_{w_{0,\text{ineq}}}
$$

2. Batch Approach: Without substitution

The original optimization problem,

$$\begin{aligned} \min_{U_0} \quad & x_N^T P x_N + \sum_{i=0}^{N-1} \left(x_i^T Q x_i + u_i^T R u_i \right) \\ \text{s.t.} \quad & x_{k+1} = A x_k + B u_k \\ & x_k \in \mathcal{X}, u_k \in \mathcal{U} \\ & x_N \in \mathcal{X}_F \\ & x_0 = x(0) \end{aligned}$$

Can be written in terms of the optimization variables U_0, X_0 as,

$$\begin{aligned} J_0^*(x(0)) = \min_{X_0, U_0} \quad & \begin{bmatrix} X_0^T & U_0^T \end{bmatrix} \begin{bmatrix} \bar{Q} & 0 \\ 0 & \bar{R} \end{bmatrix} \begin{bmatrix} X_0 \\ U_0 \end{bmatrix} \\ \text{s.t.} \quad & G_{0,\text{eq}} \begin{bmatrix} X_0 \\ U_0 \end{bmatrix} = E_{0,\text{eq}} x(0) \\ & G_{0,\text{ineq}} \begin{bmatrix} X_0 \\ U_0 \end{bmatrix} \leq E_{0,\text{ineq}} x(0) + w_{0,\text{ineq}} \end{aligned}$$

Caltech

Batch Approach: Summary

We reformulated the original finite-horizon, constrained, optimal control problem in two way using the batch method.

1. With substitution of the equality constraints to obtain an optimization problem with just the control inputs as the optimization variables.
2. Without substitution of the equality constraints to obtain an optimization problem with both the control input and states as the optimization variables.

What is the difference?

Next Lecture: Why did we pose the optimization problem in the two ways that we did? Discussion on convexity. More examples.