16-745 Optimal Control Lecture 13

Reid Graves

March 11, 2025

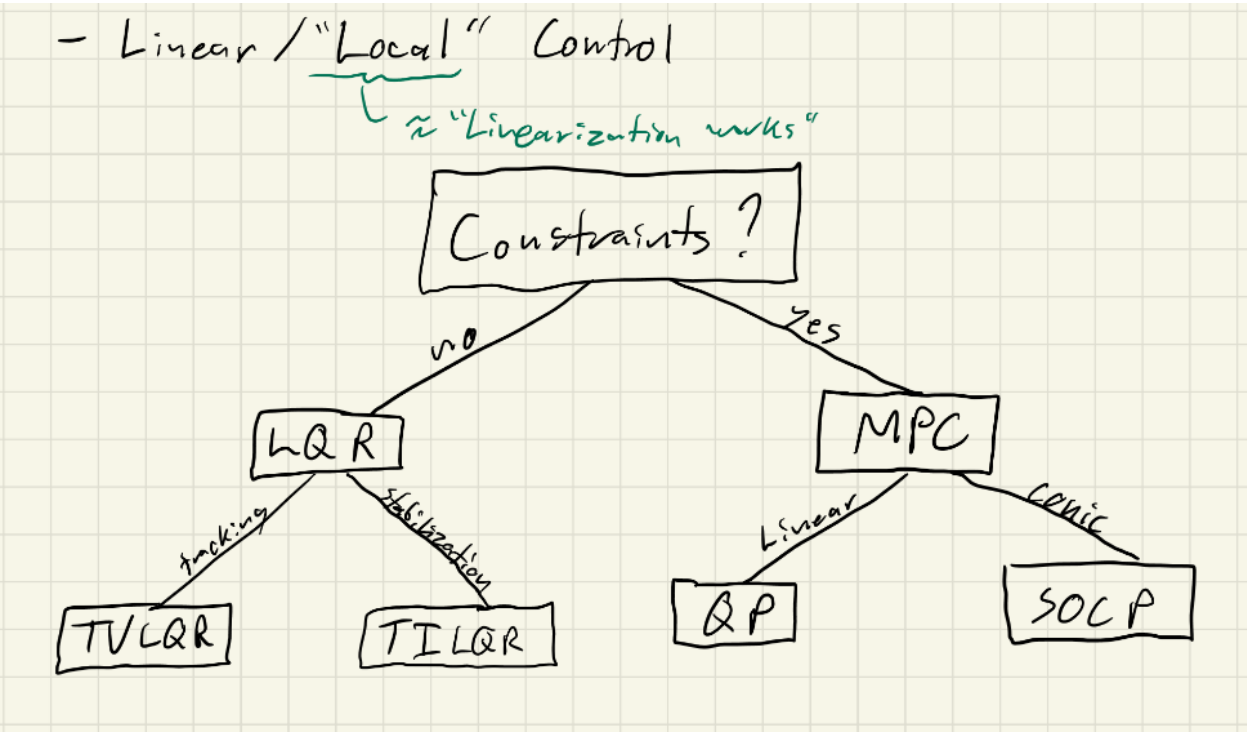
Last Time:

- SQP
- Direct Collocation

Today:

- Algorithm Recap
- Dealing with Attitude (3D rotations)

Recap: Deterministic Optimal Control Algorithms



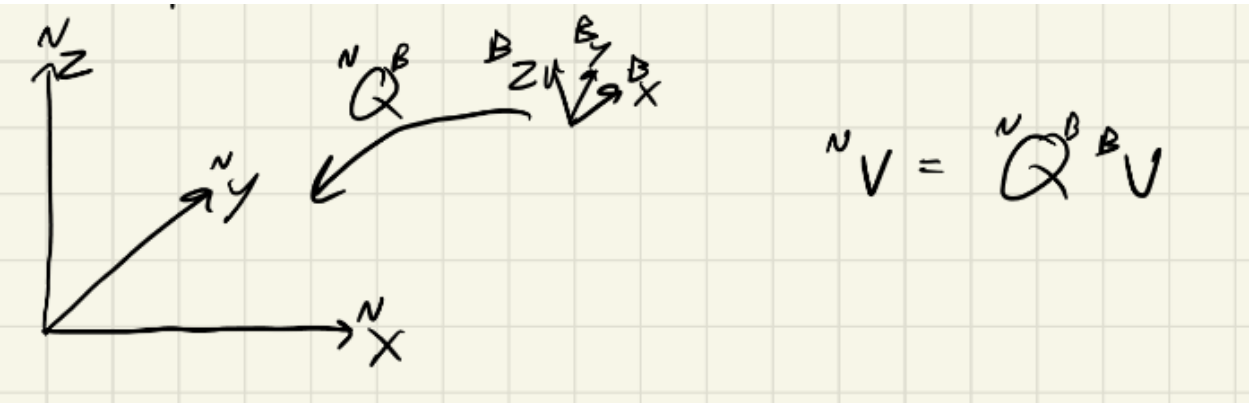
Nonlinear Trajectory Optimization/Planning

DIRCOL	DDP
Only respects dynamics at convergence	Always dynamically feasible
Can use infeasible guess	Can only guess controls
Can handle arbitrary constraints	Hard to handle constraints
Tracking controller must be designed separately	TVLQR tracking controller is free
Typically not as fast	Very fast (local) convergence
Difficult to implement large-scale SQP solver	Easy to implement on embedded systems
Numerically robust	Has issues with ill-conditioning

- **DDP** is often a good choice for *online/real-time* applications where speed is critical and constraint tolerance is not critical.
- **DIRCOL** is often a good choice for *offline* trajectory design, especially over long horizons and/or with complex constraints.

Attitude

- Many robotic systems undergo **large rotations** (quadrotors, airplanes, spacecraft, underwater vehicles, legged robots).
- Naïve angle-based parameterizations (Euler angles) have **singularities** that cause failures and/or require hacks. **Never use them**
- Rotation matrices and quaternions are **singularity-free**, but optimizing over them requires some extra tricks. Both are over parameterized. using 9 numbers for rotation matrices and 4 numbers for quaternions, so have 6 and 1 constrains respectively because there are only 3 degrees of freedom.
- **What is Attitude?** rotation from some body frame on the robot to some fixed world frame. Left superscripts for frame being rotated to, right superscript for frame being rotated from



- Rotation from body frame to world frame
- 3DOF, but there is no globally nonsingular 3-parameter attitude representation.

Rotation / "Direction-Cosine" Matrix

$$\begin{bmatrix} {}^N X_1 \\ {}^N X_2 \\ {}^N X_3 \end{bmatrix} = \underbrace{\begin{bmatrix} {}^B n_1^T \\ {}^B n_2^T \\ {}^B n_3^T \end{bmatrix}}_{{}^N Q^B} \begin{bmatrix} {}^B X_1 \\ {}^B X_2 \\ {}^B X_3 \end{bmatrix}$$

$$= \begin{bmatrix} {}^N b_1 & {}^N b_2 & {}^N b_3 \\ & & \end{bmatrix} \begin{bmatrix} {}^B X_1 \\ {}^B X_2 \\ {}^B X_3 \end{bmatrix}$$

$$\Rightarrow Q^T Q = I \quad \Rightarrow Q^{-1} = Q^T$$
$$\Rightarrow \text{"Orthogonal"}$$
$$\Rightarrow det(Q) = 1 \quad \text{"special"}$$
$$Q \in SO(3) \quad \text{"Special Orthogonal group in 3D"}$$

Where a group is just a class of matrices with certain properties

Kinematics (How do I integrate a gyro?)

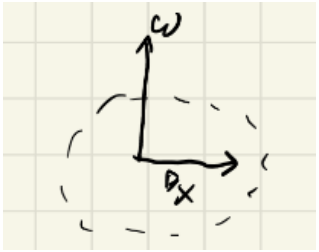


Figure 1: A record spinning at some constant ω

$$
\begin{aligned}
{}^N X = Q(t)\ {}^B X, \quad {}^N \dot{X} &= {}^N \omega \times {}^N X \\
&= Q({}^B \omega \times {}^B X)
\end{aligned}
$$

Apply chain rule:

$$
\begin{aligned}
{}^N X = Q\, {}^B X \quad \Rightarrow \quad {}^N \dot{X} &= \dot{Q}\, {}^B X + Q\, {}^B \dot{X} \\
{}^B \dot{X} &= 0 \\
{}^N \dot{X} &= \dot{Q}\, {}^B X
\end{aligned}
$$

$$
\Rightarrow \quad \dot{Q}\, {}^B X = Q({}^B \omega \times {}^B X)
$$

$$
\omega \times x = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \hat{\omega} x
$$

$$
\Rightarrow \quad \dot{Q}\, {}^B X = Q \hat{\omega}\, {}^B X
$$

$$
\Rightarrow \quad \boxed{\dot{Q} = {}^N Q^B\ {}^B \hat{\omega}}
$$

- We could do dynamics with rotation matrices in our state, but this has a lot of redundancy.
- Quaternions are more compact/efficient.

Quaternions

- Define axis of rotation (unit vector) $\mathbf{a}$.
- Define angle of rotation (scalar, radians) θ.

$$
\boldsymbol{\phi} = \mathbf{a}\theta
$$

3D vector whos direction is θ and magnitude is $\mathbf{a}$

$$
\|\boldsymbol{\phi}\| = \theta, \quad \frac{\boldsymbol{\phi}}{\|\boldsymbol{\phi}\|} = \mathbf{a}
$$

- For constant ω (for short h), we can think of $\boldsymbol{\phi}$ as the integral of ω:

$$
\boldsymbol{\phi} \approx \int_0^h \omega(t)\, dt
$$

(not true in general)

- In terms of $\boldsymbol{\phi}$, we can define quaternions:

$$q = \begin{bmatrix} \cos(\Theta/2) \\ \mathbf{a}\sin(\Theta/2) \end{bmatrix} \leftarrow \begin{bmatrix} \text{“scalar part”} \\ \text{“vector part”} \end{bmatrix}$$

- $q^T q = 1 \quad \Rightarrow \quad$ Valid rotations correspond to “unit quaternions”. Easy to normalize.
- q and $-q$ correspond to the same rotation (adding 2π to Θ). Called a “double cover”.
- Operations on quaternions are analogous to rotation matrices.

Quaternion Multiplication

$$q_1 * q_2 = \begin{bmatrix} s_1 \\ V_1 \end{bmatrix} * \begin{bmatrix} s_2 \\ V_2 \end{bmatrix} = \begin{bmatrix} s_1 s_2 - V_1^T V_2 \\ s_1 V_2 + s_2 V_1 + V_1 \times V_2 \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} s_1 & -\hat{V}_1 \\ v_1 & s_1 I + \hat{V}_1 \end{bmatrix}}_{L(q_1)} \begin{bmatrix} s_2 \\ V_2 \end{bmatrix} = \underbrace{\begin{bmatrix} s_2 & -\hat{V}_2^T \\ V_2 & sI - \hat{V}_2 \end{bmatrix}}_{R(q_2)} \begin{bmatrix} s_2 \\ V_2 \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} s_2 I_{3\times 3} - \hat{V}_2 \end{bmatrix}}_{R(q_2)} \begin{bmatrix} s_1 \\ V_1 \end{bmatrix}$$

Quaternion Conjugate

$$q^\dagger = \begin{bmatrix} s \\ -V \end{bmatrix} = Tq = \begin{bmatrix} 1 & 0 \\ 0 & -I \end{bmatrix} \begin{bmatrix} s \\ V \end{bmatrix}$$

Rotate a Vector

$$\begin{bmatrix} 0 \\ X \end{bmatrix} = q * \begin{bmatrix} 0 \\ {}^B X \end{bmatrix} * q^\dagger$$

$$= H^T L(q) R(q) H \; {}^B X$$

$$H^T R^T(q) L(q) H \; {}^B X$$

(gives $Q(q)$)

$$HX = \begin{bmatrix} 0 \\ X \end{bmatrix} \Rightarrow H = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

Quaternion Kinematics

$$\dot{q} = \frac{1}{2} q * \begin{bmatrix} 0 \\ \omega \end{bmatrix} = \frac{1}{2} L(q) H \omega$$

$$\underbrace{L(q)H}_{4\times 3}$$

- Now we can simulate dynamics with quaternions.