Last Time:

- Optimization with Quaternions

Today:

- LQR Quaternions
- Quadrotor Control

---

## * LQR with Quaternions

- Naively linearizing a system with a quaternion state results in an uncontrollable linear system.
- We'll apply our quaternion differentiation tricks to LQR to make this work.
- Given a reference $\bar{x}_n$, $\bar{u}_n$ for a discrete-time system $f(x_n, u_n)$:

$$\bar{x}_{n+1} + \Delta x_{n+1} = f(\bar{x}_n + \Delta x_n, \bar{u}_n + \Delta u_n)$$

$$\approx f(\bar{x}_n, \bar{u}_n) + \underbrace{A_n}_{\frac{\partial f}{\partial x}} \Delta x_n + \underbrace{B_n}_{\frac{\partial f}{\partial u}} \Delta u_n$$

- For the quaternion part of the state, we apply the attitude Jacobian to convert $\Delta \rightarrow \phi \in \mathbb{R}^3$

$$x = \begin{bmatrix} r \\ q \\ v \\ \dot{r} \\ \omega \\ \theta \end{bmatrix} \quad \begin{matrix} x[1:3] \\ x[4:7] \\ x[8:n] \\ \vdots \\ \vdots \end{matrix}$$

$$\underbrace{\begin{bmatrix} \delta x_{n+1}[1:3] \\ \phi_{n+1} \\ \delta x_{n+1}[8:n] \end{bmatrix}}_{\delta \tilde{x}_{n+1}} = \underbrace{\begin{bmatrix} I & 0 \\ 0 & G(q_{n+1}) & 0 \\ & & I \end{bmatrix}^T}_{E(\bar{x}_{n+1})} A_n \underbrace{\begin{bmatrix} I & 0 \\ 0 & G(\bar{q}_n) & 0 \\ & & I \end{bmatrix}}_{E(\bar{x}_n)} \begin{bmatrix} \delta x_n[1:3] \\ \phi_n \\ \delta x_n[8:n] \end{bmatrix}$$

$$+ E^T(\bar{x}_{n+1}) B_n \delta u_n$$

- Once we have these "reduced" Jacobians $\tilde{A}_n$, $\tilde{B}_n$:

$$\tilde{A}_n = E(\bar{x}_{n+1})^T A_n E(\bar{x}_n), \quad \tilde{B}_n = E^T(\bar{x}_{n+1}) B_n$$

we compute the LQR controller as usual.

- When we run the controller, we calculate $\delta \tilde{x}$ before multiplying by $K$:

given $x_n$, $\delta \tilde{x}_n = \begin{bmatrix} x_n[1:3] - \bar{x}_n[1:3] \\ \phi(L(\bar{q}_n)^T q_n) \\ x_n[8:n] - \bar{x}_n[8:n] \end{bmatrix}$

whatever 3-parameter representation you like

$$u_n = \bar{u}_n - K_n \delta \tilde{x}_n$$

* Computing error/delta rotations:

- many possible conventions
- We will write it as rotation from body frame $B$ to the reference/desired body frame $R$:

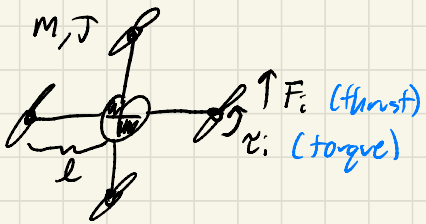
$$\Rightarrow {}^N Q^B = {}^N Q^R \, {}^R Q^B \quad \Rightarrow \quad ({}^N Q^R)^{-1} \, {}^N Q^B = {}^R Q^B$$

$$Q = \bar{Q} \, \delta Q \quad \Rightarrow \quad \delta Q = \bar{Q}^T Q$$

- Using quaternions

$$\Omega q = \vec{q}^{*} * q = L^{T}(q)\, q$$

* 3D Quadrotor




$$F_i = K_T u_i, \quad u \in \mathbb{R}^4$$
$$\tau_i = K_\tau u_i$$

- State:

$$x = \begin{bmatrix} {}^{N}r \in \mathbb{R}^3 & \text{position in } N \text{ frame} \\ {}^{N}q^{B} \in \mathbb{H} & \text{attitude } B \to N \\ {}^{B}V \in \mathbb{R}^3 & \text{linear velocity in } B \text{ frame} \\ {}^{B}\omega \in \mathbb{R}^3 & \text{angular velocity in } B \text{ frame} \end{bmatrix}$$

- Kinematics

$${}^{N}\dot{r} = {}^{N}V = Q\,{}^{B}V$$

$$\dot{q} = \frac{1}{2} q * \hat{\omega} = \frac{1}{2} L(q) H\, {}^{B}\omega = \frac{1}{2} G(q)\, {}^{B}\omega$$

- Translation Dynamics

$$m\,{}^{N}\dot{V} = {}^{N}F$$

${}^{N}F$ ← total force

↖ need to rotate into B frame

$${}^{N}V = Q\,{}^{B}V \Rightarrow {}^{N}\dot{V} = \underbrace{\dot{Q}}_{Q\hat{\omega}}\,{}^{B}V + Q\,{}^{B}\dot{V} = Q\hat{\omega}\,{}^{B}V + Q\,{}^{B}\dot{V}$$

$$\Rightarrow {}^{B}\dot{V} = \overbrace{Q^{T}\,{}^{N}\dot{V}}^{\text{rotate into } B \text{ frame}} - {}^{B}\omega \times {}^{B}V$$

← extra term from spin

$$\Rightarrow {}^B\dot{V} = \frac{1}{m}{}^B F - {}^B\omega \times {}^B V$$

$${}^B F = Q^T \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ k_T & k_T & k_T & k_T \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}$$

- Rotation Dynamics:

$$J^B\dot{\omega} + {}^B\omega \times J^B\omega = {}^B\tau$$ ← total torque

$J$: Inertia matrix

"Euler's equation"

$${}^B\tau = \begin{bmatrix} l k_T (u_2 - u_4) \\ l k_T (u_3 - u_1) \\ k_2 (u_1 - u_2 + u_3 - u_3) \end{bmatrix}$$

* Example:

- LQR (or convex MPC) with some quaternion tricks is very effective.