

Last Time:

- Deterministic OC summary
- LQR vs. MPC
- DDP vs. DIRCOL
- Quaternions

Today:

- Optimization with Quaternions

* Quaternion Recap:

- 4D Unit vectors
- Multiplication rule

$$q_1 * q_2 = \begin{bmatrix} s_1 \\ v_1 \end{bmatrix} * \begin{bmatrix} s_2 \\ v_2 \end{bmatrix} = \begin{bmatrix} s_1 s_2 - v_1^T v_2 \\ s_1 v_2 + s_2 v_1 + v_1 \times v_2 \end{bmatrix}$$

$$L(q_1) = \begin{bmatrix} s_1 & v_1^T \\ v_1 & s_1 I + \tilde{v}_1 \end{bmatrix} \Rightarrow q_1 * q_2 = \frac{L(q_1) q_2}{R(q_2) q_1}$$

- Conjugate

$$q^+ = \begin{bmatrix} s \\ -v \end{bmatrix} = T q \quad , \quad T = \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$$

- Identity

$$q^I = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

- "Hat map" for Quaternions:

$$\hat{\omega} = \begin{bmatrix} 0 \\ \omega \end{bmatrix} = H\omega, \quad H = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

* Geometry of Quaternions

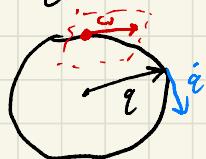


q lives on a sphere in \mathbb{R}^4

\dot{q} lives in the tangent plane at q

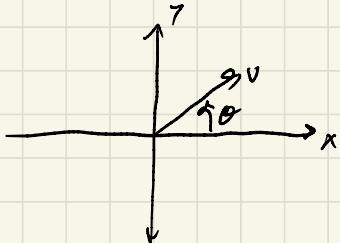
- Kinematics

$$\dot{q} \in \mathbb{R}^4, \quad \omega \in \mathbb{R}^3, \quad \dot{q} = \frac{1}{2}q \hat{\omega} = \begin{bmatrix} 0 \\ \omega \end{bmatrix} = \frac{1}{2}L(q)H\omega$$



ω is always written in the tangent plane at the identity, then kinematics rotates to tangent plane at q .

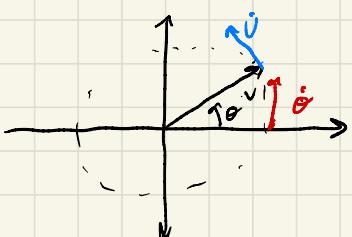
- Analogy with unit complex numbers in 2D



$$v = \cos(\theta) + i \sin(\theta)$$

$$= \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} \Rightarrow v^T v = 1$$

$$\dot{v} = \frac{dv}{d\theta} \dot{\theta} = \begin{bmatrix} -\sin(\theta) \\ \cos(\theta) \end{bmatrix} \dot{\theta} = \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}}_{\text{rotation matrix}} \begin{bmatrix} 0 \\ \dot{\theta} \end{bmatrix}$$



$2D$ "hat map"

Kinematics rotates $\dot{\theta}$ from tangent plane at $\theta=0$ to tangent at current v

* Differentiating Quaternions

- Two key facts

1) Derivatives are really 3D tangent vectors

2) Rotations compose by multiplication, not addition

- Infinitesimal Rotation

$$\delta q = \begin{bmatrix} \cos(\theta/2) \\ \sin(\theta/2) \end{bmatrix} \approx \begin{bmatrix} 1 \\ \frac{1}{2}\theta \end{bmatrix} \approx \begin{bmatrix} 1 \\ \frac{1}{2}\phi \end{bmatrix}$$

↗ small axis-angle vector

$$= \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 0 \\ \phi \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \frac{1}{2} H\phi$$

- Compose with q :

$$\begin{aligned} q' &= q * \delta q = L(q) \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} + \frac{1}{2} H\phi \right) \\ &= q + \frac{1}{2} \underbrace{L(q) H}_{G(q)} \phi \end{aligned}$$

$G(q) \in \mathbb{R}^{4 \times 3}$

"Attitude Jacobian"

- Note: we can use any 3-parameter rotation representation we want for ϕ . They all linearize the same (up to a permutation/scaling).

$$q = \underbrace{\begin{bmatrix} \cos(\theta/2) \\ \frac{\phi}{\|\phi\|} \sin(\theta/2) \end{bmatrix}}_{\text{axis-angle}} = \underbrace{\begin{bmatrix} \sqrt{1-\phi^T\phi} \\ \phi \end{bmatrix}}_{\text{vector part of } q} = \frac{1}{\sqrt{1+\phi^T\phi}} \begin{bmatrix} 1 \\ \phi \end{bmatrix}$$

vector part
of q

Gibbs/Rodrigues

- We'll use the vector part of q in class
 - This lets us differentiate w.r.t. quaternions by inserting $G(q)$ in the right places:
- $f(q) : \mathbb{H} \rightarrow \mathbb{R}$ (gradient of a scalar-valued function)
- \uparrow quaternions
("Hamilton")

$$\nabla f = \frac{\partial f}{\partial q} \frac{\partial q}{\partial \theta} = \frac{\partial f}{\partial q} G(q)$$

$f(q) : \mathbb{H} \rightarrow \mathbb{H}$ (Jacobian of a quaternion-valued function)

$$\phi' = \left[G(f(q))^T \frac{\partial f}{\partial q} G(q) \right] \phi$$

$\underbrace{\qquad\qquad\qquad}_{\nabla f \in \mathbb{R}^{3x3}}$

\uparrow transform output \uparrow transform input

- Hessian of $f(q) : \mathbb{H} \rightarrow \mathbb{R}$

$$\nabla^2 f = G(q)^T \frac{\partial^2 f}{\partial q^2} G(q) + \underbrace{I \left(\frac{\partial f}{\partial q} q \right)}_{\substack{3 \times 3 \\ \text{comes from } \frac{\partial f}{\partial q}}} \underbrace{\qquad\qquad\qquad}_{\text{scalar}}$$

- Now we can do Newton's method and DDP and SQP with quaternions

* Example : Posc Estimation

- Given a bunch of vectors to known landmarks in the environment, determine robot's attitude.

- Called "Wahba's Problem"

$$\min_q J(q) = \sum_{n=1}^m \|{}^N X_n - Q(q)^T {}^B X_n\|_2^2 = \|r(q)\|_2^2$$

Known vectors
in world frame
(from map)
Observed vectors
in body frame
(from camera)

$r(q)^T r(q)$
 \Downarrow
 "residual" vector

- ${}^N X_n$ and ${}^B X_n$ are unit vectors ("directions")

$$r(q) = \begin{bmatrix} {}^N X_1 - Q(q)^T {}^B X_1 \\ {}^N X_2 - Q(q)^T {}^B X_2 \\ \vdots \\ {}^N X_m - Q(q)^T {}^B X_m \end{bmatrix} \Rightarrow \underbrace{\frac{\partial r}{\partial q}}_{3m \times 3} = \underbrace{\frac{\partial r}{\partial q}}_{3m \times 4} \underbrace{G(q)}_{4 \times 3}$$

- Background: Gauss - Newton for Least-squares:

$$\min_x J(x) = \frac{1}{2} \|r(x)\|_2^2 = \frac{1}{2} r(x)^T r(x)$$

$$\frac{\partial J}{\partial x} = r(x)^T \frac{\partial r}{\partial x}$$

$$\frac{\partial^2 J}{\partial x^2} = \left(\frac{\partial r}{\partial x} \right)^T \left(\frac{\partial r}{\partial x} \right) + (I \otimes r(x)^T) \frac{\partial^2 \text{vec}(r)}{\partial x^2}$$

$$\Rightarrow \left(\frac{\partial J}{\partial x} \right) \nabla J \approx \left[\left(\frac{\partial r}{\partial x} \right)^T \left(\frac{\partial r}{\partial x} \right) \right]^{-1} \frac{\partial r}{\partial x}^T r(x)$$

throw this out

* Gauss-Newton for Wahba's Problem:

$$q \leftarrow q_0 \quad (\text{initial guess})$$

do :

$$\nabla r(q) = \frac{\partial r}{\partial q} b(q)$$

$$\phi = -[(\nabla r^T \nabla r)^{-1} \nabla r^T] r(q)$$

$$q \leftarrow q + \begin{bmatrix} \sqrt{1-\phi^T \phi} \\ \phi \end{bmatrix} = L(q) \begin{bmatrix} \sqrt{1-\phi^T \phi} \\ \phi \end{bmatrix}$$

(multiplicative update)

(in general, do line search)

while $\|r(q)\| > tol$