16-745 Optimal Control Lecture 15

Reid Graves

March 13, 2025

1 Last Time

- Deterministic Optimal Control summary
- Linear Quadratic Regulator (LQR) vs. Model Predictive Control (MPC)
- Differential Dynamic Programming (DDP) vs. Direct Collocation (DIRCOL)

2 Today

- Optimization with Quaternions

Quaternion Recap

- **4D Unit Vectors**
- **Multiplication Rule**

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

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

- **Conjugate**

$$q^\dagger = \begin{bmatrix} s \\ -v \end{bmatrix} = Tq, \quad T = \begin{bmatrix} 1 & 0 \\ 0 & -I \end{bmatrix}$$

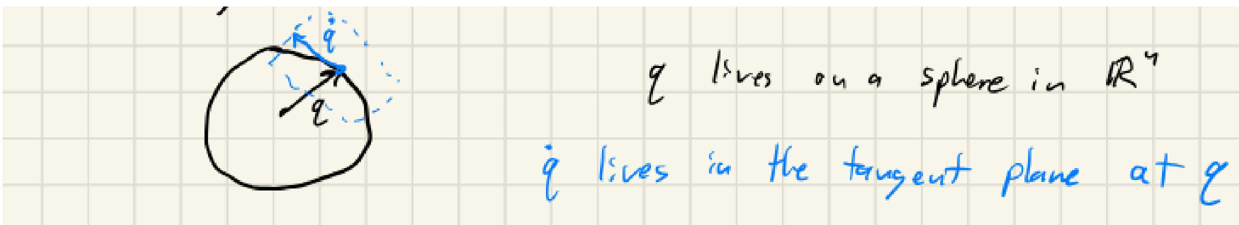
- **Identity**

$$q_I = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

- "Hat map" for Quaternions:

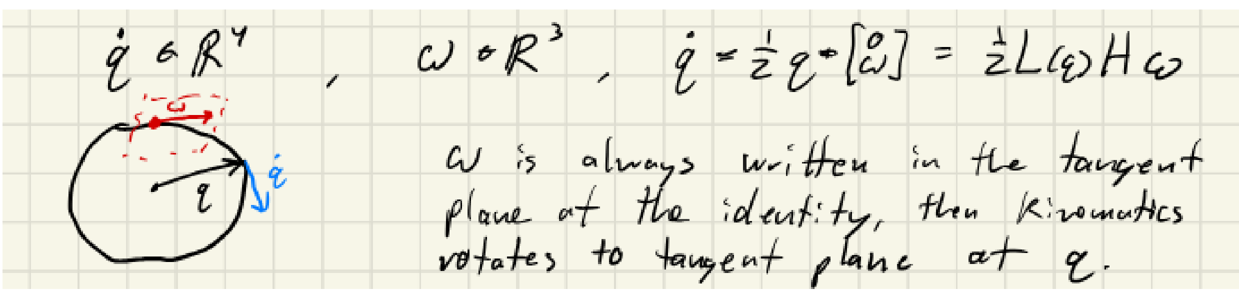
$$\hat{w} = \begin{bmatrix} 0 \\ \omega \end{bmatrix} = H\omega, \quad H = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

3 Geometry of Quaternions



- q lives on a sphere in $\mathbb{R}^4$
- $\dot{q}$ lives in the tangent plane at q

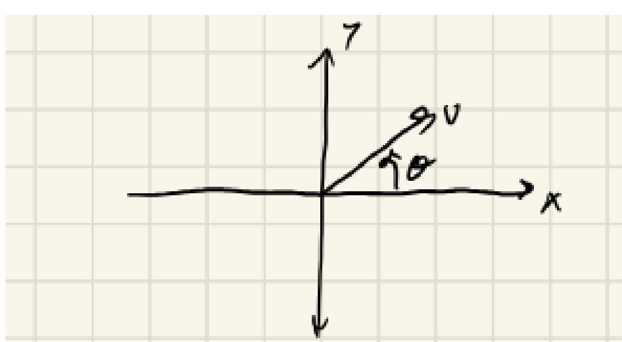
3.1 Kinematics



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

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

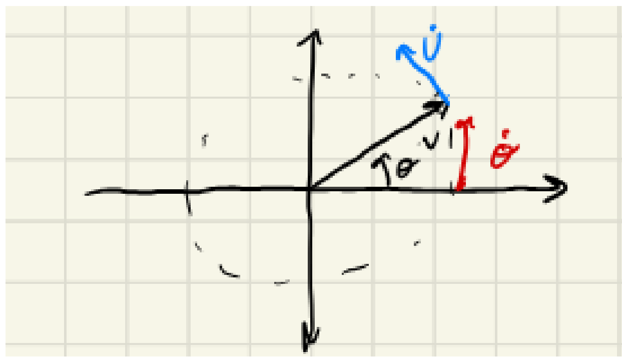
3.2 Analogy with unit complex numbers in 2D



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

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

$$\dot{v} = \frac{\partial v}{\partial \theta}\dot{\theta} = \begin{bmatrix}-\sin(\theta)\\ \cos(\theta)\end{bmatrix}\dot{\theta} = \underbrace{\begin{bmatrix}\cos(\theta) & -\sin(\theta)\\ \sin(\theta) & \cos(\theta)\end{bmatrix}}_{\text{rotation matrix}} \quad \underbrace{\begin{bmatrix}0\\ \dot{\theta}\end{bmatrix}}_{\text{2D ``hat map''}}$$



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

4 Differentiating Quaternions

- Two key facts
  1. Derivatives are really 3D tangent vectors
  2. Rotations compose by multiplication, not addition

4.1 Infinitessimal Rotation

$$\delta q = \begin{bmatrix}\cos\left(\frac{\theta}{2}\right)\\ a\sin\left(\frac{\theta}{2}\right)\end{bmatrix} \approx \begin{bmatrix}1\\ \frac{1}{2}a\theta\end{bmatrix} \approx \underbrace{\begin{bmatrix}1\\ \frac{1}{2}\phi\end{bmatrix}}_{\text{small axis-angle vector}}$$

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

4.2 Compose with q

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

$$= q + \frac{1}{2}\underbrace{L(q)H}_{G(q)\in\mathbb{R}^{4\times 3}}$$

"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)

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

- 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:

$$f(q) : \underbrace{|\mathbb{H}|}_{\text{Quaternions ("Hamilton")}} \to \mathbb{R} \text{ (Gradient of a scalar-valued function)}$$

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

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

$$\phi' = \underbrace{\left[G(f(q))^T \tfrac{\partial f}{\partial q} G(q)\right]}_{\nabla f \in \mathbb{R}^{3\times 3}} \phi$$

$$G(f(q))^T : \text{transform output}$$

$$G(q) : \text{transform input}$$

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

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

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

5 Example: Pose Estimation

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

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

$$= r(q)^T \underbrace{r(q)}_{\text{"residual"}}$$

$${}^N x_k : \text{Known vectors in world frame (from map)}$$

$${}^B x_k : \text{Observed vectors in body frame (from camera)}$$

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

$$r(q) = \begin{bmatrix} {}^{N}x_1 - Q(q)\ {}^{B}x_1 \\ {}^{N}x_2 - Q(q)\ {}^{B}x_2 \\ \vdots \\ {}^{N}x_m - Q(q)\ {}^{B}x_m \end{bmatrix} \Rightarrow \underbrace{\nabla r(q)}_{3m\times 3} = \underbrace{\frac{\partial r}{\partial q}}_{3m\times 4} \underbrace{G(q)}_{4\times 3}$$

- Background: Gauss-Newton for Least-Squares:

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

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

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

throw this out: $(I \otimes r^T(x))\dfrac{\partial^2 vec(r)}{\partial x^2}$

$$\Rightarrow \left(\frac{\partial J}{\partial x^2}\right)^{-1} \nabla J \approx \left[\left(\tfrac{\partial r}{\partial x}\right)^T \left(\tfrac{\partial r}{\partial x}\right)\right]^{-1} \frac{\partial r}{\partial x}^T r(x)$$

6 Gauss-Newton For Wahba's Problem

$q \leftarrow q_0$ (initial guess)

do:

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

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

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

(multiplicative update)

(in general, do line search)

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