Last Time:

- Deterministic Optimal Control Recap
- LQR vs. MPC
- DDP vs. DIRCOL
- Quaternion Intro

Today:

- Optimization with Quaternions

---

* Quaternion Recap:

- 4D unit vectors
- Multiplication rule

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

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

- Quaternion Conjugate

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

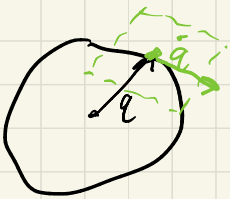
- Identity:

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

- "Hat map" for Quaternions

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

* Geometry of Quaternions:

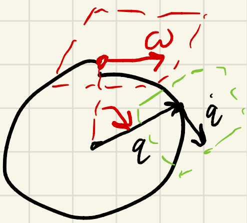


q lives on a sphere in $\mathbb{R}^4$

$\dot{q}$ lives in the tangent plane to the sphere at q

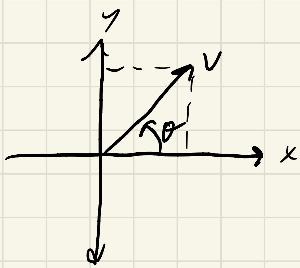
- Kinematics:

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



ω is always written down in the tangent plane at the identity, then kinematics rotate to the tangent plane centered on q.

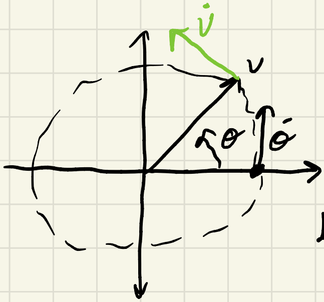
- Analogy with unit complex numbers in 2D:



$$v = \underbrace{\cos(\theta)}_{x \text{ component}} + \underbrace{i\sin(\theta)}_{y \text{ component}} \Rightarrow$$

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

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



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

* Differentiating Quaternions

- Two key facts

1) Derivatives are really 3D "tangent vectors"

2) Rotations compose by multiplication, not addition

- Infinitesimal Rotation

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

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:

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

$$= q + \tfrac{1}{2}\underbrace{L(q)H}_{\substack{G(q)\,\in\,\mathbb{R}^{4\times 3} \\ \text{"Attitude Jacobian"}}}\phi$$

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

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

- 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:

$f(q): \mathbb{H} \to \mathbb{R}$ (gradient of a scalar-valued function)

quaternion ("Hamilton")

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

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

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

transform output

transform input

$$q' = f(q) \Rightarrow q'\delta q' = f(q\delta q)$$

Another way to think about what we're doing with these 3D derivatives

$$q' * \begin{bmatrix} \sqrt{1-\phi'^T\phi'} \\ \phi' \end{bmatrix} = f\left(q * \begin{bmatrix} \sqrt{1-\phi^T\phi} \\ \phi \end{bmatrix}\right)$$

$$\Rightarrow \frac{\partial \phi'}{\partial \phi} = G(f(q))^T \frac{\partial f}{\partial q} G(q)$$

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

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

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

Example: Pose Estimation

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

- Called "Wahba's Problem"

$$\min_q \; J(q) = \sum_{n=1}^{m} \|\underbrace{{}^N x_n}_{\substack{\text{known vectors in} \\ \text{inertial/world frame} \\ \text{(e.g. from a map)}}} - Q(q)\underbrace{{}^B x_n}_{\substack{\text{measured vectors} \\ \text{in body frame} \\ \text{(e.g. from camera)}}}\|_2^2 = \|\overbrace{r(q)}^{\text{"residual"}}\|_2^2$$

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

$$r(q) = \underbrace{\begin{bmatrix} {}^N x_1 - Q\,{}^B x_1 \\ {}^N x_2 - Q\,{}^B x_2 \\ \vdots \\ {}^N x_m - Q\,{}^B x_m \end{bmatrix}}_{3m \times 1} \Rightarrow \underbrace{\nabla r(q)}_{3m \times 3} = \underbrace{\frac{\partial r}{\partial q}}_{3m \times 4} \underbrace{G(q)}_{4 \times 3}$$

* Background: Gauss-Newton for Least-Squares

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

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

throw this term out

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

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

* Gauss-Newton Method for Wahba's Problem:

$q \leftarrow q_0$ (initial guess)

do:

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

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

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

(in general, do a line search)

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