* Attitude Dynamics + Simulation

- What is attitude?

- Rotation from some body-fixed frame to some inertial frame (e.g. ECI)

- We can parameterize this many ways:

  - Rotation matrices (good)

  - Quaternions (good)

  - Euler angles (roll, pitch, yaw) (bad)

  - Axis-angle vectors (OK-ish)

---

* Rotation Matrices:

- Rotate components of a vector from body to inertial frames:

$$ {}^{N}x = {}^{N}Q^{B}\,{}^{B}x $$

- Rows are n_i basis vectors expressed in B components:

$$ \begin{bmatrix} {}^{N}x_1 \\ {}^{N}x_2 \\ {}^{N}x_3 \end{bmatrix} = \begin{bmatrix} {}^{B}n_1^T \\ {}^{B}n_2^T \\ {}^{B}n_3^T \end{bmatrix} {}^{B}x $$

- Columns are b_i basis vectors in N components:

$$ {}^{N}x = \begin{bmatrix} {}^{N}b_1 & {}^{N}b_2 & {}^{N}b_3 \end{bmatrix} \begin{bmatrix} {}^{B}x_1 \\ {}^{B}x_2 \\ {}^{B}x_3 \end{bmatrix} $$

$$\Rightarrow \quad QQ^T = Q^TQ = I \quad \Rightarrow \quad Q^T = Q^{-1}$$

$\Rightarrow$ Q is an orthogonal matrix ($Q \in O(3)$)

$\det(Q) = 1$ "special"

$$\Rightarrow \quad Q \in SO(3)$$

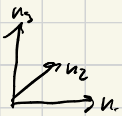
---

* Rotation Kinematics:

- How do we integrate a gyro?

$$\omega(t) \xrightarrow[?]{} \dot{Q}(t) \longrightarrow Q(t)$$

- Velocities in a rotating frame:



$${}^N\dot{r} = {}^NQ^B\left({}^B\dot{r} + {}^B\omega \times {}^Br\right)$$

$$\underbrace{{}^B\dot{r} = Q^T\,{}^N\dot{r} - {}^B\omega \times {}^Br}_{\text{"Kinematic transport theorem"}}$$

- Think about a vector fixed in body frame:

$${}^Nr = Q\,{}^Br \Rightarrow {}^N\dot{r} = \dot{Q}\,{}^Br + \cancelto{0}{Q\,{}^B\dot{r}} = Q\left({}^B\omega \times {}^Br\right)$$

- Define skew-symmetric "hat" matrix:

$$\omega \times r = \hat{\omega} r \Rightarrow \hat{\omega} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$$

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

- This is a linear 1st order ODE. For constant ω:

$$Q(t) = Q_0 e^{\hat{\omega} t}$$

matrix exponential (expm in Matlab)

- We can also define axis-angle vectors:

$$\hat{\phi} = \log(Q), \quad \phi = a\theta$$

unit-vector axis

angle in radians

---

* Rigid-Body Dynamics:

- For a rigid body, the derivative of the angular momentum in the N frame equals torque (just like $F = ma = \dot{p}$)

$$^N\dot{h} = {}^N\tau$$

- Apply Kinematic transport theorem:

$$^N\dot{h} = Q({}^B\dot{h} + {}^B\omega \times {}^Bh) = {}^N\tau$$

$$\Rightarrow {}^B\dot{h} + {}^B\omega \times {}^Bh = {}^B\tau$$

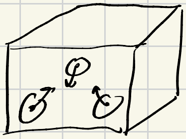
- Plug in definition of angular momentum:

$$^Bh = {}^BJ\,{}^B\omega \Rightarrow {}^B\dot{h} = {}^BJ\,{}^B\dot{\omega}$$

$$\Rightarrow \boxed{J\dot{\omega} + \omega \times J\omega = \tau} \leftarrow \text{"Euler's Equation"}$$

* Gyrostats:

- A collection of rigid bodies who's relative motion doesn't change the total inertia of the system:



- Idealized model of a spacecraft with reaction wheels.

- Slightly modify Euler's equation:

$$ {}^{B}h = {}^{B}J\,{}^{B}\omega + \underbrace{{}^{B}\rho}_{\text{wheel momentum}} $$

- Plug into $\dot{h} + \omega \times h = \tau$:

$$ \left. J\dot{\omega} + \underbrace{\dot{\rho}}_{\text{wheel torque}} + \omega \times (J\omega + \rho) = \tau \right\} \text{"Gyrostat Equation"} $$

* Simulating Attitude Dynamics:

- We can simulate attitude dynamics with e.g. RK4:

$$ x = \begin{array}{cl} \begin{bmatrix} Q \\ \omega \\ \rho \end{bmatrix} & \begin{array}{l} \leftarrow \text{rotation matrix} \\ \leftarrow \text{angular velocity} \\ \leftarrow \text{wheel momentum} \end{array} \end{array} \qquad u = \begin{bmatrix} \dot{\rho} \\ \tau \end{bmatrix} $$

$$ \dot{x} = f(x,u) = \begin{bmatrix} \dot{Q} \\ \dot{\omega} \\ \dot{\rho} \end{bmatrix} = \begin{bmatrix} Q\hat{\omega} \\ -J^{-1}(\dot{\rho} + \omega \times (J\omega + \rho) - \tau) \\ \dot{\rho} \end{bmatrix} $$

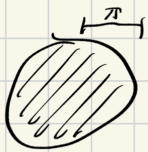
* Problem: Q quickly drifts such that $Q^T Q \neq I$

- Can re-project with SVD, but this is expensive

* Quaternion Geometry:

- Set of all possible axis-angle vectors $-\pi < \|\phi\| \leq \pi$ is a ball in $\mathbb{R}^3$

- Visualize as a disk in $\mathbb{R}^2$:



- Discontinuous jump when we cross $\pm\pi$ that causes "kinematic singularity"

1) Stretch disk up out of plane into hemisphere:

2) Make a copy

3) Rotate copy and glue it on underneath to make a sphere.

- Now instead of jumping, we can smoothly continue onto the "southern hemisphere"

- There are 2 quaternions for every rotation matrix ("double cover of SO(3)")

* Useful Quaternion Stuff:

- Points on the unit sphere in 4D:

$$q = \begin{bmatrix} \cos(\theta/2) \\ a \sin(\theta/2) \end{bmatrix}, \quad \begin{matrix} a = \text{axis (unit vector)} \\ \theta = \text{angle (radians)} \end{matrix}$$

$$= \begin{bmatrix} s \\ \hline v \end{bmatrix} \begin{matrix} \leftarrow \text{"scalar part"} \\ \leftarrow \text{"vector part"} \end{matrix}$$

- This is the quaternion exponential

* Identity Quaternion

$$\theta = 0 \Rightarrow q_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

* Quaternion Conjugate (opposite rotation)

$$q^{+} = \begin{bmatrix} \cos(-\theta/2) \\ a \sin(-\theta/2) \end{bmatrix} = \begin{bmatrix} s \\ \hline -v \end{bmatrix}$$

* Quaternion Multiplication:

- Works just like multiplying rotation matrices!

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

$$= \underbrace{\begin{bmatrix} s_1 & -v_1^T \\ v_1 & s_1 I + \hat{v}_1 \end{bmatrix}}_{L(q_1)} \begin{bmatrix} s_2 \\ v_2 \end{bmatrix} = \underbrace{\begin{bmatrix} s_2 & -v_2^T \\ v_2 & s_2 I - \hat{v}_2 \end{bmatrix}}_{R(q_2)} \begin{bmatrix} s_1 \\ \hline v_1 \end{bmatrix}$$

$$L(q^{+}) = L^T(q), \quad R(q^{+}) = R^T(q)$$

- See "Planning with Attitude" paper

* Rotating Vectors:

$$\begin{bmatrix} 0 \\ {}^N x \end{bmatrix} = q * \begin{bmatrix} 0 \\ {}^B x \end{bmatrix} * q^\dagger = \underbrace{L(q) R^T(q)}\begin{bmatrix} 0 \\ {}^B x \end{bmatrix} = \underbrace{R^T(q) L(q)}\begin{bmatrix} 0 \\ {}^B x \end{bmatrix}$$

Lower-right Block is ${}^N Q^B$

* Quaternion Kinematics

- For small angles / Δt:

$$\phi = a\theta \approx \omega \Delta t$$

- Let's look at a small rotation:

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

$$\approx q * \begin{bmatrix} 1 \\ \frac{1}{2}\omega \Delta t \end{bmatrix} = q * \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{2}\omega \Delta t \end{bmatrix} \right)$$

$$= q + \frac{1}{2}\Delta t \, q * \begin{bmatrix} 0 \\ \omega \end{bmatrix}$$

$$\Rightarrow \frac{q' - q}{\Delta t} = \frac{1}{2} q * \begin{bmatrix} 0 \\ \omega \end{bmatrix} \Rightarrow \dot{q} = \frac{1}{2} q * \begin{bmatrix} 0 \\ \omega \end{bmatrix}$$

- In matrix notation:

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

* Simulating with Quaternions:

$$x = \begin{bmatrix} q \\ \omega \\ \rho \end{bmatrix}, \quad u = \begin{bmatrix} \dot{\rho} \\ \tau \end{bmatrix}$$

$$\dot{x} = f(x, u) = \begin{bmatrix} \dot{q} \\ \dot{\omega} \\ \dot{\rho} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} L(q) H \omega \\ \vdots \\ \vdots \end{bmatrix}$$

- Integrate with e.g. RK4
- Renormalize $q \leftarrow \frac{q}{\|q\|}$ at each step (easy!)