What is Attitude?

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

- Canonical representation: Rotation matrix (many other choices)

Rotation Matrices:

- Rotates components of a vector from body to inertial frame:

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

- Rows are $\underline{n}_i$ basis vectors expressed in B components

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

- Columns are $\underline{b}_i$ basis vectors in N-components

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

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

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

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

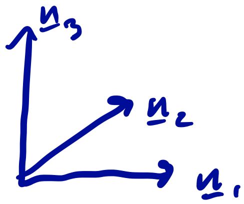
$$ \Rightarrow Q \in SO(3) $$

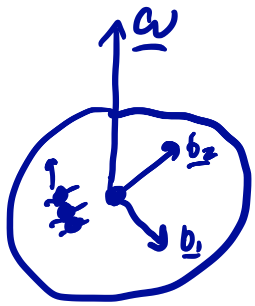
Rotation Kinematics:

- How do we integrate a gyro?

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

- Velocities in a rotating frame





$${}^N\dot{x} = {}^N Q^B \left({}^B\dot{x} + {}^B\omega \times {}^B x\right)$$

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

- Think about a vector fixed in the body frame:

$${}^N x = Q\,{}^B x \Rightarrow {}^N\dot{x} = \dot{Q}\,{}^B x + \cancel{Q\,{}^B\dot{x}}^{\,0}$$
$$= Q\left({}^B\omega \times {}^B x\right)$$

* Define skew-symmetric "hat" opperator

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

$$\Rightarrow \boxed{\dot{Q} = Q\hat{\omega}}$$

* Note this is a linear 1^{st} order ODE. For constant ω:

$$Q(t) = Q_0 \underbrace{e^{\hat{\omega} t}}_{\text{matrix exponential (expm in Matlab)}} \approx Q_0\left(I + \hat{\omega} t\right)$$

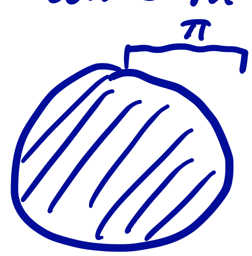
* We can also define "axis-angle" vectors:

$$\vec{\phi} = \underbrace{\log}_{\substack{\text{matrix log} \\ \text{(logm in matlab)}}}(Q) \; , \quad \phi = \underbrace{r}_{\substack{\text{unit vector axis} \\ \text{of rotation}}} \underbrace{\theta}_{\text{angle in radians}}$$

Quaternions:

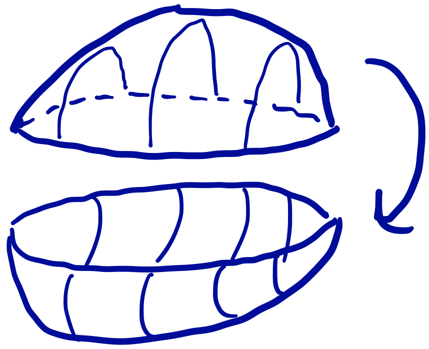
- Naive integration of $\dot{Q} = Q\hat{\omega}$ will lead to $Q \notin SO(3)$ due to numerical error (try it)
- Projecting back onto $SO(3)$ is a pain (need SVD)
- Quaternions are much nicer for simulations.

* Geometry
  - Set of all possible axis-angle vectors $-\pi < \|\phi\| \leq \pi$ is a ball in $\mathbb{R}^3$
  - Visualize as a disk in $\mathbb{R}^2$:



- There's a discontinuous jump when we cross $\pm\pi$ that causes "kinematic singularity"
- We want to get rid of the jump:

1) Stretch disk up out of plane into a hemisphere



2) Make a copy

3) Rotate the copy and glue it underneath the original to make a sphere

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

- There are 2 quaternions for every rotation matrix. We call this a "double cover" of SO(3)

Useful Quaternion Stuff:

* Points on the unit sphere in 4D are given by:

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

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

- turns out this is the quaternion exponential

* Identity Quaternion

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

* Quaternion "Conjugate" (Inverse)

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

* Quaternion Multiplication

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

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

- $L(q^+) = L^T(q)$, $R(q^+) = R^T(q)$

* How to Rotate Vectors:

$$\begin{bmatrix} 0 \\ {}^N x \end{bmatrix} = q \begin{bmatrix} 0 \\ {}^B x \end{bmatrix} q^+$$

* Quaternions Work Just Like Rotation Matrices:

$$Q_3 = Q_2 Q_1 \longleftrightarrow q_3 = q_2 q_1$$

$$Q_1 = Q_2^T Q_3 \longleftrightarrow q_1 = q_2^+ q_3$$

$${}^N\hat{x} = Q\, {}^B\hat{x}\, Q^T \longleftrightarrow \underbrace{\begin{bmatrix} 0 \\ {}^N x \end{bmatrix} = q \begin{bmatrix} 0 \\ {}^B x \end{bmatrix} q^+}$$

skew symmetric matrix

$\updownarrow$

"pure vector" quaternion

$$\begin{bmatrix} 0 \\ {}^N x \end{bmatrix} = \underbrace{L(q) R^T(q)}_{\text{rotation matrix is lower right block}} \begin{bmatrix} 0 \\ {}^B x \end{bmatrix}$$

* Quaternion Kinematics :

$$q_2 = q_1 \delta q = q_1 \begin{bmatrix} \cos(\delta\theta/2) \\ r\sin(\delta\theta/2) \end{bmatrix} \approx q_1 \begin{bmatrix} 1 \\ r\frac{\delta\theta}{2} \end{bmatrix}$$

$$\approx q_1 + q_1 \begin{bmatrix} 0 \\ \frac{1}{2} r\,\delta\theta \end{bmatrix}$$

$$\dot{q} \approx \frac{q_2 - q_1}{\delta t} = \frac{q_1 \begin{bmatrix} 0 \\ \frac{1}{2} r\,\delta\theta \end{bmatrix}}{\delta t} = \frac{1}{2} q_1 \begin{bmatrix} 0 \\ r\,\delta\theta/\delta t \end{bmatrix}$$

$$\Rightarrow \boxed{\dot{q} = \frac{1}{2} q \begin{bmatrix} 0 \\ \omega \end{bmatrix}}$$