Last Time:

q-Method

Today:

- Calculating rotation errors
- Notes on random sampling
- Algorithm Implementation

How to Compute Rotation Errors:

- Given an estimate ${}^{N}Q^{B}_{est}$ and the true value ${}^{N}Q^{B}_{true}$ we want the error in degrees
- If we had vectors we'd subtract: $e = x_{est} - x_{true}$
- We will use matrix multiplication instead

$$E = {}^{B_{true}}Q^{B_{est}} = \left({}^{N}Q^{B_{true}}\right)^{T} {}^{N}Q^{B_{est}}$$

- Several choices for ordering/frame to represent error in. This is an arbitrary convention.
- Convert E to an axis-angle vector:

$$e^{\hat{\phi}} = Q \Rightarrow e = \overbrace{\log(E)}^{\text{Vhat operator turns skew-symmetric matrix into vector}}$$

matrix log (logm in Matlab)

- Scalar error in degrees is:

$$\theta_e = \left(\frac{180}{\pi}\right)\|e\|$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \Rightarrow \hat{x} = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$$

How to Generate Random Rotations:

* We know how to generate random vectors:
  - randn.m: Gaussian distribution with $\mu = 0$, $P = I$
  - For a different mean + covariance:

$$x_{samp} = \sqrt{P}\,\text{randn}(n, 1) + \mu$$

  - There are several different Matrix square roots. The most common one in applications is Cholesky:

$$\text{chol}(P) = LL^T$$

* How about random rotation matrices?
  - Generate random axis-angle vectors with the desired covariance:

$$\phi_{samp} = \sqrt{P}\,\text{randn}(3, 1)$$

  - Use matrix exponential to convert to rotation matrix:

$$Q = e^{\hat{\phi}} = \underbrace{\text{expm}(\hat{\phi})}_{\text{matlab}}$$

  - If you want a non-zero (non-Identity) mean, use matrix multiplication:

$${}^{N}Q^{B} = Q_0\, e^{\hat{\phi}}$$

  - For quaternions, do the same thing but convert the axis-angle vector to a quaternion using:

$$q = \begin{bmatrix} \dfrac{\phi \sin(\|\phi\|/2)}{\sqrt{\phi^T\phi}} \\ \cos(\|\phi\|/2) \end{bmatrix}$$