Last Time:

- Dynamic Balance
- Attitude Sensors

Today:

- Static Attitude Determination
- TRIAD
- Wahba's Problem

TRIAD

- First ever attitude determination algorithm (early 60's)
- US Transit Satellites (predecessor to GPS)

* Need at least 2 unit vectors to uniquely determine attitude

- Assume 2 unit vectors known in both body and inertial frames (must be linearly independent)

$$[{}^N r_1 \ \ {}^N r_2] = {}^N Q^B \, [{}^B r_1 \ \ {}^B r_2]$$

- Make a 3rd linearly independent vector by taking a cross product:

$$\underbrace{{}^N[r_1 \,\vdots\, r_2 \,\vdots\, (r_1 \times r_2)]}_{{}^N R} = {}^N Q^B \, \underbrace{{}^B[r_1 \,\vdots\, r_2 \,\vdots\, (r_1 \times r_2)]}_{{}^B R}$$

- As long r_1 and r_2 are linearly independent, R will be invertable

$$\Rightarrow \boxed{{}^N Q^B = {}^N R \, {}^B R^{-1}}$$

- There is a trick to avoid the matrix inverse

$$ {}^N T = \Big[\underbrace{{}^N r_1}_{{}^N t_1} \;\vdots\; \underbrace{\frac{{}^N r_1 \times {}^N r_2}{\| {}^N r_1 \times {}^N r_2 \|}}_{{}^N t_2} \;\vdots\; \underbrace{\frac{{}^N t_1 \times {}^N t_2}{\| {}^N t_1 \times {}^N t_2 \|}}_{{}^N t_3} \Big], \quad {}^B T = \Big[{}^B r_1 \;\vdots\; \frac{{}^B r_1 \times {}^B r_2}{\| {}^B r_1 \times {}^B r_2 \|} \;\vdots\; \frac{{}^B t_1 \times {}^B t_2}{\| {}^B t_1 \times {}^B t_2 \|} \Big] $$

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

- Since T is orthogonal:

$$ \boxed{ {}^N Q^B = {}^N T \, {}^B T^T } $$

- This is the classic TRIAD method
- Very simple to implement
- Limitations:
  - Can only handle 2 observations
  - Static: doesn't account for dynamics
  - No "covariance" or "error bound" information

Wahba's Problem:

- What happens if we want to use more than 2 observations to improve our estimate?
- Define a least-squares cost function for attitude:

$$ L = \sum_i \underbrace{w_i}_{\text{scalar weights}} \| {}^N r_i - {}^N Q^B \, {}^B r_i \|_2^2 $$

- Weights should be inversely proportional to sensor variance to give maximum likelihood answer

- Find attitude by solving the following optimization problem:

$$\min_Q \; L(Q)$$
$$\text{s.t.} \quad Q \in SO(3)$$

- The SO(3) constraint makes this interesting
- Many solution techniques in the litterature
- We will talk about 3:

1) Convex Optimization
2) SVD Method
3) q-Method

Convex Optimization Solution:

- Look closer at Wahba cost function:

$$L = \sum_i w_i ({}^N r_i - Q\,{}^B r_i)^T ({}^N r_i - Q\,{}^B r_i)$$

$$= \sum_i w_i (\underbrace{\cancel{{}^N r_i^T\,{}^N r_i}}_{\text{const}} - 2\,{}^N r_i^T Q\,{}^B r_i + \underbrace{\cancel{{}^B r_i^T Q^T Q\,{}^B r_i}}_{\text{const}})$$

$$= \sum_i w_i (-2\,{}^N r_i^T Q\,{}^B r_i)$$

$$= -2\,\mathrm{Tr}\Big[\underbrace{\Big(\sum_i w_i\,{}^B r_i\,{}^N r_i^T\Big)}_{B = \text{"attitude profile matrix"}} Q\Big]$$

cost function is linear in Q!

* Trace trick: $x^T A y = \mathrm{Tr}[x^T A y] = \mathrm{Tr}[y x^T A]$

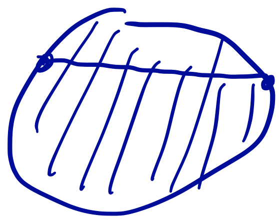
- We can now re-write Wahba's Problem as:

$$\max_{Q} \quad \mathrm{Tr}[BQ]$$

$$\text{s.t.} \quad Q^T Q = I$$

$$\det(Q) = 1$$

- The objective is linear but constraints are non-convex



$x^T x = 1$ non convex

$x^T x \leq 1$ convex

- Replace equality with inequalities to "relax" into a convex problem:

$$\max_{Q} \quad \mathrm{Tr}(BQ)$$

$$\text{s.t.} \quad I - Q^T Q \succeq 0$$

- Because the objective is linear, we will always end up on the boundary of the set, satisfying the constraint with equality
- Called a "tight relaxation"
- This is a semidefinite program (SDP)
- There are lots of good solvers available for SDPs (e.g. SEDUMI in Matlab)

SVD Solution:

- The rotation matrix can be written:

$$ {}^N Q^B = {}^N r_1 {}^B r_1^T + {}^N r_2 {}^B r_2^T + {}^N r_3 {}^B r_3^T $$

where the r_i are mutually-orthogonal unit vectors

- We've already seen a special case:

$$ {}^N Q^B = \begin{bmatrix} {}^B n_1^T \\ {}^B n_2^T \\ {}^B n_3^T \end{bmatrix} = {}^N n_1 {}^B n_1^T + {}^N n_2 {}^B n_2^T + {}^N n_3 {}^B n_3^T $$

* What happens if r_i are not mutually orthogonal?

- Attitude Profile Matrix:

$$ B = \sum_i w_i {}^B r_i {}^N r_i^T $$

- The optimal ${}^N Q^B$ is the closest special-orthogonal matrix to B^T

* Singular Value Decomposition

$$ M = U \Sigma V^T, $$

U and V are orthogonal
Σ is diagonal

- Diagonal elements of Σ are called "singular values" $\sigma_i \geq 0$
- Sort of like an eigendecomposition but more general
- We can make a special orthogonal matrix by setting $\sigma_i = \pm 1$

* SVD Algorithm:

1) form B from measurements
2) Compute $U \Sigma V^T = \mathrm{svd}(B^T)$

3) $${}^{N}Q^{B} = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(U)\det(V) \end{bmatrix} V^T$$