Last Time:

- Linear Kalman Filter

Today:

- Extended Kalman Filter
- Multiplicative EKF

Kalman Filters for Nonlinear Systems:

- Last time we assumed a stochastic LTV system:

$$x_{k+1} = A_k x_k + B_k u_k + w_k, \quad w_k \sim N(0, W)$$

$$y_k = C_k x_k + v_k, \quad v_k \sim N(0, V)$$

- What happens if we have a nonlinear system?

$$x_{k+1} = f(x_k, u_k) + w_k$$

$$y_k = g(x_k) + v_k$$

(we're still assuming additive noise for simplicity)

- Basic idea: linearize $f(x,u)$ and $g(x)$ about your current estimate wherever necessary in the original KF algorithm

EKF Algorithm:

1) Initialize: $\bar{x}_{0|0}$, $P_{0|0}$

2) Predict:

$$\bar{x}_{k+1|k} = f(\bar{x}_{k|k}, u_k)$$

$$A_k = \left.\frac{\partial f}{\partial x}\right|_{\bar{x}_{k|k}}, \quad B_k = \left.\frac{\partial f}{\partial u}\right|_{\bar{x}_{k|k}}$$

$$P_{k+1|k} = A_k P_{k|k} A^T + W$$

3) Innovation:

$$z_{k+1} = y_{k+1} - g(\bar{x}_{k+1|k})$$

$$C_{k+1} = \frac{\partial g}{\partial x}\Big|_{\bar{x}_{k+1|k}}$$

$$S_{k+1} = C_{k+1} P_{k+1|k} C_{k+1}^T + V$$

4) Kalman Gain:

$$L_{k+1} = P_{k+1|k} C_{k+1}^T S_{k+1}^{-1}$$

5) Update:

$$\bar{x}_{k+1|k+1} = \bar{x}_{k+1|k} + L_{k+1} z_{k+1}$$

$$P_{k+1|k+1} = (I - L_{k+1} C_{k+1}) P_{k+1|k} (I - L_{k+1} C_{k+1})^T + L_{k+1} V L_{k+1}^T$$

6) Go to 2

Attitude Statistics:

- How do we write down a covariance for a rotation matrix or quaternion?

- Due to the unit norm constraint, the 4×4 covariance computed for random quaternions will be (practically) singular
- More importantly, A 4D Gaussian is a bad description of the error statistics of 3 DOF rotations
- FYI there are "intrinsic" distributions on $SO(3)$ e.g. the Bingham distribution.

- To use a Gaussian with rotations, we take advantage of matrix/quaternion multiplication and the exponential:

$$\mu = Q_0 \in SO(3) \quad \text{(or unit quaternion)}$$

$$\delta x = x - \mu \rightarrow P = \sum_i \delta x \, \delta x^T$$

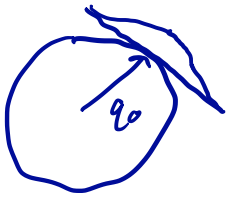
$$\underbrace{\delta Q = Q_0^T Q}_{Q = \tilde{Q}_0^B \delta Q} \rightarrow P = \sum_i \check{\log}(\delta Q) \, \check{\log}(\delta Q)^T \qquad P \in S^3_{tr}$$

Matrix Exponential:

$$\text{axis-angle vector } \phi \rightarrow Q = e^{\hat{\phi}}$$

$$\Rightarrow \phi = \check{\log}(Q)$$

- The mean is a rotation, but the covariance is defined using vectors in $\mathbb{R}^3$. Geometrically, we can think of this as representing a Gaussian in the (hyper) plane tangent to the sphere:



- Other 3-parameter representations (e.g. Gibbs vectors) can be used. They're all equivalent to 1st order.

The Simplest Possible Linearized Dynamics:

- Many spacecraft have very good gyros
- If we assume the gyro is "perfect", we don't need to worry about Euler's equation, just the kinematics:

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

- If our sample rate is much faster than the dynamics, we can assume ω is constant over a sample period:

$$q_{k+1} \approx q_k \begin{bmatrix} r \sin(\theta/2) \\ \cos(\theta/2) \end{bmatrix}, \quad r = \frac{\omega_k}{\sqrt{\omega_k^T \omega_k}}, \quad \theta = \omega_k \delta t$$

- For very small ω and/or δt we can use:

$$q_{k+1} \approx q_k \begin{bmatrix} \omega_k \delta t/2 \\ 1 \end{bmatrix} \simeq q_k + q_k \begin{bmatrix} \omega_k \delta t/2 \\ 0 \end{bmatrix}$$

- From a linear system perspective, ω is effectively treated as a control input:

$$\underbrace{q_{k+1}}_{x_{k+1}} = \underbrace{I}_{A} \underbrace{q_k}_{x_k} + \underbrace{\begin{bmatrix} \delta t/2 \, (s_k I + \hat{v}_k) \\ -\delta t/2 \; v_k^T \end{bmatrix}}_{B_k} \underbrace{\omega_k}_{u_k}$$

- Since $A = I$, covariance prediction is trivial

Measurement Model for Unit Vectors:

- Assume we know ${}^{N}r$ perfectly and we have noisy measurements of ${}^{B}r$

$$y_{k+1} = {}^{B}r = {}^{B}Q^{N}\,{}^{N}r = g(x)$$

- We need to linearize this w.r.t. an axis-angle vector

$${}^{B}r + \delta{}^{B}r = \left(\check{Q}^{B}_{k+1|k}\, e^{\hat{\phi}}\right)^{T} {}^{N}r \simeq (I - \hat{\phi})\,{}^{B}\check{Q}^{N}_{k+1|k}\,{}^{N}r$$

$$\Rightarrow \delta{}^{B}r = -\hat{\phi}\,{}^{B}\check{Q}^{N}_{k+1|k}\,{}^{N}r = \underbrace{\left({}^{B}\check{Q}^{N}_{k+1|k}\,{}^{N}r\right)}_{C_{k+1}}\phi$$

$$C_{k+1} = \underbrace{{}^{B}\check{r}_{k+1|k}}_{\text{stack for each } {}^{B}r_i}$$

Simplest MEKF Algorithm:

1) Initialize: $\bar{q}_{0|0}$, $P_{0|0}$

2) Predict:

$$\bar{q}_{k+1|k} = \bar{q}_{k|k} \begin{bmatrix} r \sin(\theta/2) \\ \cos(\theta/2) \end{bmatrix}, \quad r = \frac{\omega_n}{\sqrt{\omega_n^T \omega_n}}, \quad \theta = \|\omega_n\| \delta t$$

3) Innovation:

$$z_{k+1} = \underbrace{\begin{bmatrix} {}^B r_1 \\ {}^B r_2 \\ \vdots \end{bmatrix}}_{y_{k+1}} - \underbrace{\begin{bmatrix} {}^B Q^N_{k+1|k} & & 0 \\ & \ddots & \\ 0 & & {}^B Q^N_{k+1|k} \end{bmatrix}}_{3N \times 3N} \underbrace{\begin{bmatrix} {}^N r_1 \\ {}^N r_2 \\ \vdots \end{bmatrix}}_{3N \times 1}$$

$$C_{k+1} = \underbrace{\begin{bmatrix} {}^B\hat{Q}^N_{k+1|k} \, {}^N r_1 \\ \vdots \\ {}^B\hat{Q}^N_{k+1|k} \, {}^N r_N \end{bmatrix}}_{3N \times 3}$$

$$S_{k+1} = C_{k+1} P_{k+1|k} C_{k+1}^T + V$$

4) Kalman Gain:

$$L_{k+1} = P_{k+1|k} C_{k+1}^T S_{k+1}^{-1}$$

5) Update:

$$\bar{\phi}_{k+1|k+1} = L_{k+1} z_{k+1}$$

$$\bar{q}_{k+1|k+1} = \bar{q}_{k+1|k} \begin{bmatrix} r \sin(\theta/2) \\ \cos(\theta/2) \end{bmatrix}, \quad \theta = \|\bar{\phi}_{k+1|k+1}\|, \; r = \frac{\bar{\phi}_{k+1|k+1}}{\theta}$$

$$P_{n+1|n+1} = (I - L_{n+1} C_{n+1}) P_{n+1|n} (I - L_{n+1} C_{n+1})^T + L_{n+1} V L_{n+1}^T$$

6) Go to 2