Expectation:

$$E[x] = \int_{-\infty}^{\infty} f(x)p(x)dx$$

probability density

Linearity:

$$E[x+y] = E[x] + E[y]$$

scalar

$$E[\alpha x] = \alpha E[x]$$

Multi-Variate Gaussian:

$$p(x) = \frac{\exp\left(-\frac{1}{2}(x-\mu)^T P^{-1}(x-\mu)\right)}{\underbrace{\sqrt{(2\pi)^n \det(P)}}_{\text{normalization constant}}}$$

mean: $\mu = E[x] \in \mathbb{R}^n$

covariance: $P = E[(x-\mu)(x-\mu)^T] \in \mathcal{S}^n_{++}$

Stochastic Linear Systems:

* Discrete-time Linear-time-varying (LTV) system w/AWGN

$$\underset{\text{state}}{x_{k+1}} = A_k x_k + B_k u_k + \underbrace{w_k}_{\text{“process noise”}}$$

$$\text{measurement} \rightarrow y_k = C_k x_k + \underbrace{v_k}_{\text{“measurement noise”}}$$

$$w_k \sim \mathcal{N}(0, Q) \qquad\qquad v_k \sim \mathcal{N}(0, R)$$

What is a Kalman Filter?

- Optimal recursive state estimator for a linear system

What are we trying to optimize?

- Minimum mean-squared error (MMSE)

$$E[(x-\bar{x})^T(x-\bar{x})]$$

$$= E[\mathrm{Tr}((x-\bar{x})(x-\bar{x})^T)] = Tr[P]$$

* This is basically least-squares

KF As Recursive MMSE Estimation:

- Assume we have an estimate of the state and error covariance that includes measurements up to timestep t_k:

$$\bar{x}_{k|k} = E[x_k | y_{1:k}]$$

$$P_{k|k} = E[(x_k - \bar{x}_{k|k})(x_k - \bar{x}_{k|k})^T]$$

- We want to update $\bar{x}$ and P to include a new measurement at timestep t_{k+1}

Prediction Step:

$$\bar{x}_{k+1|k} = E[A_k x_k + B_k u_k + w_k | y_{1:k}]$$

using linearity ↓

$$= A_k E[x_k | y_{1:k}] + B_k u_k + \underbrace{\cancel{E[w_k]}}_{0}$$

$$= A_k \bar{x}_{k|k} + B_k u_k$$

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

Innovation:

This is what we feed back in the filter

$$z_{k+1} = y_{k+1} - C_{k+1}\bar{x}_{k+1|k}$$

$$S_{k+1} = E[z_{k+1}z_{k+1}^T] = C_{k+1}P_{k+1|k}C_{k+1}^T + R$$

Measurement Update Step:

- State Update:

(Linear feedback on innovation)

$$\bar{x}_{k+1|k+1} = \bar{x}_{k+1|k} + \underbrace{L_{k+1}}_{\text{Kalman gain}} z_{k+1}$$

- Covariance Update:

$$P_{k+1|k+1} = E[(x_{k+1} - \bar{x}_{k+1|k+1})(x_{k+1} - \bar{x}_{k+1|k+1})^T]$$

$$= \underbrace{(I - L_{k+1}C_{k+1})P_{k+1|k}(I - L_{k+1}C_{k+1})^T + L_{k+1}RL_{k+1}^T}_{\text{"Joseph form" - you may see others}}$$

Kalman Gain:

- Minimize Mean-Squared Error

$$\implies \frac{\partial \operatorname{Tr}(P_{k+1|k+1})}{\partial L_{k+1}} = 0$$

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

Kalman Filter Algorithm:

1) Initialize:

$$\bar{x}_{0|0} \qquad P_{0|0} \qquad Q \qquad R$$

2) Predict:

repeat

$$\bar{x}_{k+1|k} = A_k \bar{x}_{k|k} + B_k u_k \qquad P_{k+1|k} = A_k P_{k|k} A_k^T + Q$$

3) Calculate Innovation:

$$z_{k+1} = y_{k+1} - C_{k+1} \bar{x}_{k+1|k} \qquad S_{k+1} = C_{k+1} P_{k+1|k} C_{k+1}^T + R$$

4) Calculate Kalman Gain:

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

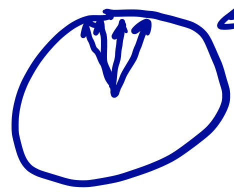
5) Update:

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

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

Attitude Statistics:

- Error covariance for quaternions

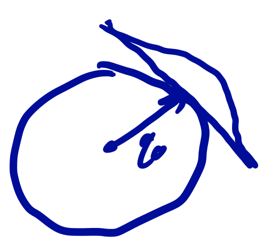


- Due to unit norm constraint, a 4×4 covariance will always be (nearly) singular
- A 4D Gaussian is a bad description of the error statistics for 3DOF rotations.
- We'll play some tricks to use a 3D Gaussian:

$$\mu = q_0 \in \mathbb{H} \quad \text{(quaternion)}$$

$$\delta x_i = x_i - \mu \quad \longrightarrow \quad P = \sum_i \delta x_i \, \delta x_i^T$$

$$\delta q_i = q_0^+ q_i = \begin{bmatrix} \sqrt{1-\phi^T\phi} \\ \phi \end{bmatrix} \quad \longrightarrow \quad P = \sum_i \phi_i \phi_i^T$$



- The mean is a 4D quaternion but the covariance is 3×3 representing a Gaussian in the tangent plane

Dynamics Model:

- If the gyro is very good we don't need to filter it
- Rember we also need to estimate the gyro bias

$$\underbrace{x_n}_{\text{state}} = \begin{bmatrix} q_n \\ \underbrace{b_n}_{\text{bias}} \end{bmatrix} \qquad \underbrace{u_n}_{\text{"control input"}} = \underbrace{\omega_n}_{\text{true}} + \underbrace{b_n}_{\text{bias}}$$

- Discrete-time Dynamics

$$x_{n+1} = f(x_n, u_n) + w_n$$

$$q_{n+1} = q_n \underbrace{\begin{bmatrix} \cos(\theta/2) \\ r\sin(\theta/2) \end{bmatrix}}_{s_n = q_n^+ q_{n+1}}, \quad \theta = \|u_n - b_n\| \delta t, \quad r = \frac{u_n - b_n}{\|u_n - b_n\|}$$

$$b_{n+1} = b_n$$

- Linearization:

$$\delta x_{n+1} = A_n \delta x_n + \underbrace{B_n \delta u_n}_{\text{don't need in filter}}, \quad \delta x = \begin{bmatrix} \phi \\ \delta b \end{bmatrix} \in \mathbb{R}^6$$

$$A_n = \left[\begin{array}{c|c} \frac{\partial \phi_{n+1}}{\partial \phi_n} & \frac{\partial \phi_{n+1}}{\partial b_n} \\ \hline \frac{\partial b_{n+1}}{\partial \phi_n} & \frac{\partial b_{n+1}}{\partial b_n} \end{array}\right]$$

* $\frac{\partial b_{n+1}}{\partial \phi_n} = 0$
* $\frac{\partial b_{n+1}}{\partial b_n} = I$

* $\dfrac{\partial \phi_{n+1}}{\partial \phi_n}$:

$$q_{n+1}\begin{bmatrix}1\\ \phi_{n+1}\end{bmatrix} \approx q_n\begin{bmatrix}1\\ \phi_n\end{bmatrix}\underbrace{\begin{bmatrix}\cos(\theta/2)\\ r\sin(\theta/2)\end{bmatrix}}_{s_n}$$

$$L(q_{n+1})V^T\phi_{n+1} \approx L(q_n)R(s_n)V^T\phi_n$$

$$\Rightarrow \phi_{n+1} = VL^T(q_{n+1})L(q_n)R(s_n)V^T\phi_n$$

$$= \underbrace{VL^T(s_n)R(s_n)V^T}_{\frac{\partial \phi_{n+1}}{\partial \phi_n}}\phi_n$$

* $\dfrac{\partial \phi_{n+1}}{\partial b_n}$

$$q_{n+1}\begin{bmatrix}1\\ \phi_{n+1}\end{bmatrix} \approx q_n\begin{bmatrix}1\\ \frac{1}{2}(\omega_n-b_n)\delta t-\frac{1}{2}\delta b_n\delta t\end{bmatrix}$$

$$q_{n+1} + q_{n+1}\begin{bmatrix}0\\ \phi_{n+1}\end{bmatrix} \approx q_{n+1} - q_{n+1}\begin{bmatrix}0\\ \frac{1}{2}\delta b_n\delta t\end{bmatrix}$$

$$\Rightarrow \boxed{\frac{\partial \phi_{n+1}}{\partial b_n} \approx \frac{1}{2}\delta t I}$$

Measurement Model:

$$\underbrace{y_k = \begin{bmatrix} {}^B r_1 \\ {}^B r_2 \\ \vdots \end{bmatrix}}_{\text{vector observations from sensors}} = \underbrace{\begin{bmatrix} {}^B Q_k^N & & \\ & {}^B Q_k^N & \\ & & \ddots \end{bmatrix} \begin{bmatrix} {}^N r_1 \\ {}^N r_2 \\ \vdots \end{bmatrix}}_{\text{predicted measurements}}$$

$$\delta y_k = \underbrace{\left[\frac{\partial y_k}{\partial \phi_k} \,\vdots\, \frac{\partial y_k}{\partial b_k} \right]}_{C_k} \begin{bmatrix} \phi_k \\ \delta b_k \end{bmatrix} \qquad * \ \frac{\partial y_k}{\partial b_k} = 0$$

$${}^B r + \delta {}^B r = ({}^N Q^B (I + 2\hat{\phi}))^T \, {}^N r = \underbrace{{}^B Q^N \, {}^N r}_{{}^B r} + 2\hat{\phi}^T \underbrace{{}^B Q^N \, {}^N r}_{{}^B r}$$

$$\Rightarrow \quad \delta {}^B r = 2\hat{\phi}^T \, {}^B r = 2 \, {}^B\hat{r} \, \phi$$

$$\Rightarrow \quad C_k = \begin{bmatrix} 2 \, {}^B\hat{r}_1 & \vdots & 0 \\ 2 \, {}^B\hat{r}_2 & \vdots & 0 \\ \vdots & \vdots & \vdots \end{bmatrix}$$

Basic MEKF Algorithm:

1) Initialize with $\bar{x}_{0|0} = \begin{bmatrix} \bar{q}_{0|0} \\ \bar{b}_{0|0} \end{bmatrix}$, $P_{0|0}$

2) Predict:

$$x_{k+1|k} = f(x_k, u_k), \quad P_{k+1|k} = A_k P_{k|k} A_k^T + Q$$

3) Innovation:

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

$$C_{k+1} = \left[\begin{array}{c|c} 2\,{}^B\hat{r}_1 & 0 \\ 2\,{}^B\hat{r}_2 & 0 \\ \vdots & \vdots \end{array}\right]$$

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

4) Kalman Gain:

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

5) Update:

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

$$\bar{q}_{k+1|k+1} = \bar{q}_{k+1|k} \begin{bmatrix} \sqrt{1 - \phi_{k+1}^T \phi_{k+1}} \\ \phi_{k+1} \end{bmatrix}$$

$$\bar{b}_{k+1|k+1} = b_{k+1|k} + \delta b_{k+1}$$

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

6) Go To 2)