Bayes Filter

Core Concept: Maintaining a belief about a state (position, velocity, etc.) in the face of uncertainty using a recursive linear state estimator.

1. The Big Picture - Uncertainty:

- Robots live in a 'noisy' world
  - ↳ Actuators are noisy: You tell a motor to move robot 1m fwd; it might move 1.05 m or 0.98 m.
  - ↳ Sensors are noisy: An ultrasonic distance sensor might return 2.02m or 1.97 m when wall is 2.00m away.
- A Bayes Filter doesn't track a single coordinate. It tracks a PDF (probability density function) $f_X(x) = P(X = x)$ where f forms a point cloud of where the robot may be.

2. The Variables:

- The following variables are used in prediction & correction:
  - ↳ x_t : the state at time t (e.g. attitude, position, velocity, etc.)
  - ↳ u_t : the control input at time t (e.g. drive fwd 2 m)
  - ↳ z_t : the measurement at time t (e.g. wall detected 3m away)
  - ↳ $Bel(x_t)$: the Belief (PDF representing point cloud of curr state).

3. The Recursive Estimation Algorithm: two-step cycle $\forall t$

- Step A: Prediction (The Motion Update):
  - ↳ We calculate the Prior Belief ($\overline{Bel}$) by shifting the previous Belief point cloud based on the control input at t.

$$\boxed{(1) \quad \overline{Bel}(x_t) = \int P(x_t \mid u_t, x_{t-1})\, Bel(x_{t-1})\, dx_{t-1}}$$

↳ Logic: We sum up every possible prev position (x_{t-1}) in our old point cloud and see where we would land if we moved u_t, our current control input.

↳ Visual: The PDF shifts and becomes wider/flatter (i.e. more uncertain) because of motor noise.

∘ Step B: Correction (The Sensor/Measurement Update):

↳ We sharpen our blurry prediction using sensor data to find the Posterior Belief, which becomes $Bel(x_t)$.

$$\boxed{(2) \quad Bel(x_t) = \eta P(z_t \mid x_t)\, \overline{Bel}(x_t)}$$

↳ Logic: For every point x in the cloud, we multiply our 'blurry prediction' probability by the 'sensor likelihood'.

↳ Eta (η) Factor: A normalization constant, scales the result so the total integral under the curve stays $= 1$. Generally, $\eta = 1 / \int_{-\infty}^{\infty} P(z_t \mid x_t)\, \overline{Bel}(x_t)\, dx_t$.

4. Why Multiplication (Information Fusion):

• Multiplying two PDFs (ex. Gaussians) is the mathematical way to combine or fuse two independant opinions.

1. Agreement: The result is high only where both PDFs are also high.
2. Uncertainty Reduction: The product of two Gaussians is always skinnier than either original. More data naturally means less uncertainty/error.
3. Weighted Average: The new 'peak' (mean) will be pulled towards the 'skinnier' (more certain) distribution.

• Aside: When you multiply two Gaussian densities $N_1(x; \mu_1, \sigma_1^2)$ and $N_2(x; \mu_2, \sigma_2^2)$ the result is $N_3(x; \mu_3, \sigma_3^2)$ scaled by constant $\left(\frac{1}{\eta}\right)$.

$$\Rightarrow \mu_3 = \frac{\mu_1 \sigma_2^2 + \mu_2 \sigma_1^2}{\sigma_1^2 + \sigma_2^2} \quad \text{and} \quad \sigma_3^2 = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

5. Concrete Example: 1D Linear Robot

Scenario: A robot is in a 10m hallway. We track the position x and the uncertainty σ^2 (i.e. variance).

Phase 1: Initial State

The robot starts at 5m but isn't perfectly sure: $Bel(x_{t-1}) = N(5,1)$.

Phase 2: The Prediction (Moving)

The robot moves 2m fwd ($u_t = 2$) with motor variance of 0.5.
Thus we simply take $N(5,1) + N(2, 0.5)$ as $\overline{Bel}(x_t)$.

- New Mean: $5.0 + 2.0 = 7.0$
- New Variance: $1.0 + 0.5 = 1.5$
- Prior Belief: $\overline{Bel}(x_t) = N(7, 1.5)$

Phase 3: The Correction (Sensing)

The sensor predicts we are at 8.0m due to measuring a distance of 2.0m from the end of the hall. Sensor variance is 0.2.
We multiply the prior belief $N(7, 1.5)$ by sensor PDF $N(8, 0.2)$.

- New Mean: $\mu_{new} = \dfrac{\mu_1 \sigma_2^2 + \mu_2 \sigma_1^2}{\sigma_1^2 + \sigma_2^2} = \dfrac{(7.0 \times 2.0) + (8.0 \times 1.5)}{1.5 \times 0.2} \approx 7.88\,m$

  (positions are averaged weighted by inverse of variance).

- New Variance: $\sigma_{new}^2 = \dfrac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2} = \dfrac{1.5 \times 0.2}{1.5 + 0.2} \approx 0.18$

  (new uncertainty always smaller than the originals).

- Posterior Belief: $Bel(x_t) = N(7.88, 0.18)$.

Conclusion: Even though our prediction (prior belief) was at 7m, the result (7.88 m) is much closer to the sensor (8m) due to its lower variance. The correction naturally "pulled" the result towards the more reliable information.

6. The Evolution of Filters

a. Bayes Filter: The theoretical "gold standard." Can handle any shape of clouds (even multimodal) but involves hard integrals.

b. Kalman Filter: Assumes all probability clouds are Gaussians. This replaces integrals with much faster matrix algebra. Only works for Linear Systems.

c. Extended Kalman Filter: (EKF) Uses Taylor Series (Jacobians) to "linearize" the world. This is what runs on most flight controllers (ex. ArduPilot, ZeroPilot) to handle nonlinear curves/rotations.