Kalman Filter: A Probabilistic Approach to Signal Estimatio

Probability & Statistics Course — Final Project

Interactive Demo: `https://leonathn.github.io/FinalProjectProbability/`

The **Kalman Filter** is an optimal recursive algorithm that estimates the state of a dynamic system from noisy measurements. This project presents an interactive web-based visualization demonstrating how the Kalman Filter applies fundamental probability concepts—including **Gaussian distributions**, **Bayes' theorem**, and **variance reduction**—to achieve superior noise filtering.

Keywords: Kalman Filter, Bayesian Estimation, Gaussian, Variance, Signal Processing

In real-world applications, measurements are always corrupted by **noise**. The challenge is to estimate the *true* underlying value.

Problem Statement:

$$\boxed{z_k = x_k + v_k}$$

where:

- z_k = noisy measurement at time k
- x_k = true (unknown) value
- $v_k \sim \mathcal{N}(0, R)$ = measurement noise

Goal: Find the best estimate $\hat{x}_k$ that minimizes the estimation error variance.

The Kalman Filter is built on key probability concepts:

Gaussian (Normal) Distribution:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

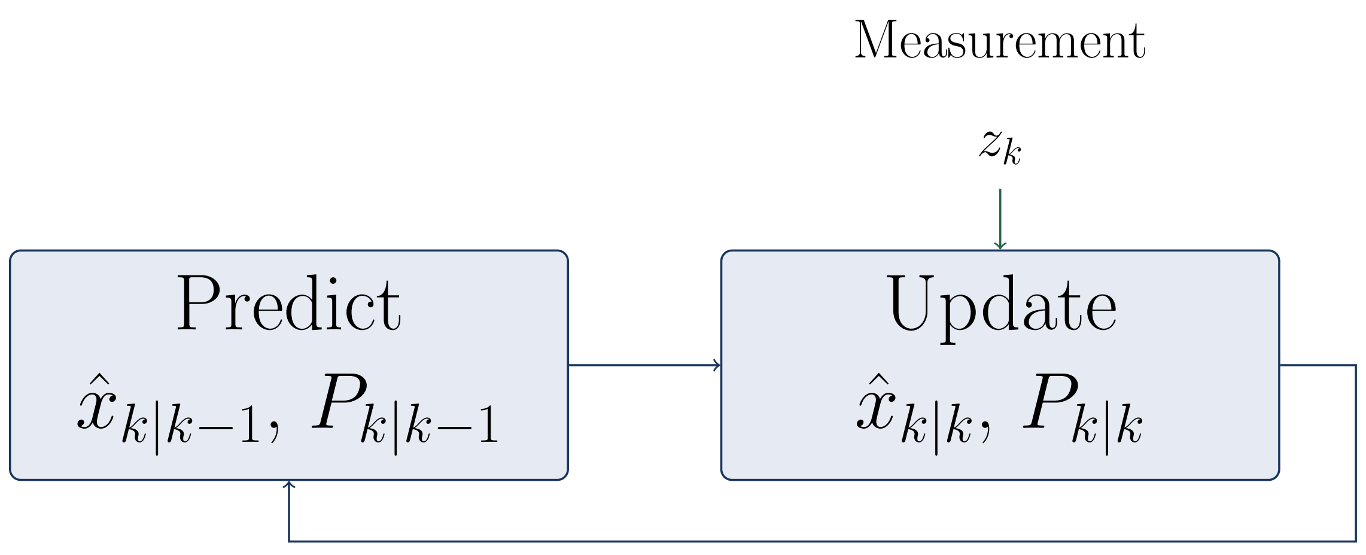
Variance as Uncertainty:

$$\text{Var}(X) = \sigma^2 = E[(X-\mu)^2]$$

Bayes' Theorem:

$$P(\text{state}|\text{meas}) = \frac{P(\text{meas}|\text{state}) \cdot P(\text{state})}{P(\text{meas})}$$

The filter operates in two phases: **Predict** and **Update**.



Step 1: Prediction

$$\hat{x}_{k|k-1} = \hat{x}_{k-1|k-1}$$
$$P_{k|k-1} = P_{k-1|k-1} + Q$$

Step 2: Kalman Gain

$$\boxed{K_k = \frac{P_{k|k-1}}{P_{k|k-1} + R}}$$

K determines trust: measurement vs prediction

Step 3: State Update

$$\boxed{\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - \hat{x}_{k|k-1})}$$

Weighted average toward measurement

Step 4: Covariance Update

$$\boxed{P_{k|k} = (1 - K_k)P_{k|k-1}}$$

Uncertainty **decreases** after measurement!

Given: $\hat{x}_{k-1} = 100$, $P_{k-1} = 4$, $z_k = 105$, $Q = 1$, $R = 10$

Step	**Formula**	**Result**
Predict	$P_{k\|k-1} = 4 + 1$	$P = 5$
Gain	$K = \frac{5}{5+10}$	$K = \mathbf{0.333}$
Update	$100 + 0.333(105 - 100)$	$\hat{x} = \mathbf{101.67}$
Covariance	$(1 - 0.333) \times 5$	$P = \mathbf{3.33}$

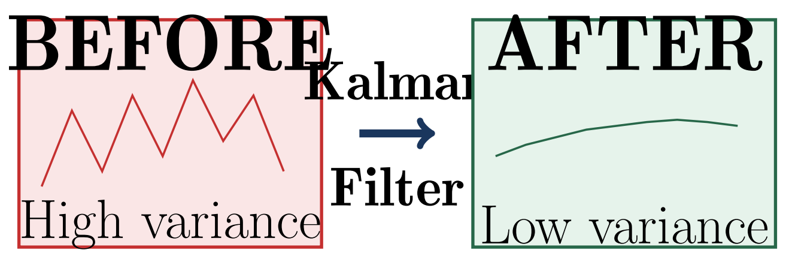
Result: Uncertainty reduced from 5 to 3.33 (**33% reduction!**)

Our web-based demo at:

`https://leonathn.github.io/FinalProjectProbability/`

Features:

- **5 Data Sources:** Bitcoin, Temperature, Stocks, Sensor, GPS
- **Before/After Comparison:** Visual noise reduction
- **Adjustable Parameters:** R and Q sliders
- **Interactive Calculator:** Step-by-step computation



Metric	Raw	Filtered
Variance	High	**-40 to -70%**
Trend Preserved	–	✓
Responsiveness	–	✓
Lag/Delay	–	Minimal

Key Findings:

1. Noise reduction consistent across all data types
2. Filter adapts via Kalman Gain
3. Real-time performance achieved

- **Navigation:** GPS, drones, spacecraft
- **Finance:** Stock prediction, trading
- **Robotics:** Sensor fusion, SLAM
- **Medical:** ECG/EEG processing

Apollo 11 used Kalman Filter to reach the Moon!

LaTeX TikZposter