

# Documentação do Projeto 9: Filtro de Kalman (Rastreamento de Veículo)

## 1. Visão Geral do Projeto

Este projeto implementa um **Filtro de Kalman** para rastrear o movimento de uma moto em uma estrada. O problema abordado é comum em robótica e sistemas de navegação: sensores como o GPS não são perfeitos; eles possuem "ruído" e imprecisão. Além disso, sensores de GPS geralmente informam apenas a posição, mas muitas vezes precisamos saber também a velocidade do veículo.

O objetivo do sistema é estimar com precisão o **estado real** da moto (sua posição exata e sua velocidade) combinando duas fontes de informação incertas:

1. **Modelo Físico (Predição):** Sabemos que uma moto em movimento tende a continuar em movimento (inércia).
2. **Medição (Correção):** Recebemos leituras ruidosas de um GPS a cada intervalo de tempo.

O filtro de Kalman funde essas informações para reduzir o ruído e inferir variáveis ocultas (neste caso, a velocidade, que não é medida diretamente pelo sensor simulado).

## 2. Como Foi Desenvolvido

O projeto foi desenvolvido em **Python**, utilizando bibliotecas numéricas para álgebra linear e visualização de dados.

- **Linguagem:** Python.
- **Bibliotecas:**
  - `numpy` : Essencial para as operações matriciais intensivas do Filtro de Kalman (multiplicação de matrizes, inversão, transposição).
  - `matplotlib` : Utilizada para gerar gráficos comparativos que mostram a trajetória real, a leitura ruidosa do GPS e a estimativa suavizada do filtro.
- **Simulação:** O código não apenas implementa o filtro, mas também cria um **cenário de simulação** completo ("Ground Truth"), gerando uma trajetória de moto com acelerações variáveis e injetando ruído gaussiano para simular um sensor GPS realístico.

## 3. Modelo e Algoritmo Utilizado

### O Filtro de Kalman

O Filtro de Kalman é um algoritmo recursivo que opera em um ciclo de dois passos: **Predição** e **Atualização**.

#### A. Definição do Estado ( $x$ )

O estado do sistema é um vetor contínuo contendo:

- **Posição ( $p$ ):** Onde a moto está (em metros).
- **Velocidade ( $v$ ):** Quão rápido ela se move (em m/s).

## B. Passo 1: Predição (Modelo de Movimento)

O algoritmo projeta o estado atual para o futuro usando as leis da física (cinemática).

- Nova Posição = Posição Antiga + (Velocidade \* Tempo) + (0.5 \* Aceleração \* Tempo<sup>2</sup>).
- Nova Velocidade = Velocidade Antiga + (Aceleração \* Tempo).
- Nesta etapa, a incerteza do sistema (representada pela matriz de covariância P) aumenta, pois o futuro é incerto.

## C. Passo 2: Atualização (Correção via Medição)

O algoritmo recebe uma nova leitura do GPS ("z").

- Ele compara o que o GPS diz com o que ele previu que aconteceria. A diferença é chamada de **Resíduo** ou *Inovação*.
- **Ganho de Kalman (K):** O filtro calcula um "peso" para decidir em quem confiar mais.
  - Se o GPS for muito preciso (ruído baixo), o Ganho "K" é alto (confia mais no sensor).
  - Se o GPS for ruim, o Ganho "K" é baixo (confia mais na predição do modelo físico).
- O estado é corrigido com base nesse ganho, e a incerteza diminui, pois recebemos nova informação.

## Resultados Observados

Os gráficos gerados pelo projeto demonstram a eficácia do filtro:

1. **Posição:** A linha do Filtro de Kalman é muito mais suave e próxima da linha "Verdadeira" do que os pontos dispersos do GPS.
2. **Velocidade:** O filtro consegue estimar a velocidade da moto com razoável precisão, mesmo que o sensor "GPS" nunca tenha informado a velocidade diretamente (apenas a posição).

## 4. Referências Bibliográficas

- Russell, S. J., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach*. 4ª ed.