# Actividad: Filtro de Kalman

Jennifer Avendaño

May 6, 2025

# 1 Introducción

El Filtro de Kalman es un algoritmo de estimación recursiva que combina modelos dinámicos con mediciones ruidosas para estimar el estado real de un sistema. Propuesto por Rudolf E. Kalman en 1960, se usa ampliamente en áreas como navegación, seguimiento por radar y robótica.

Este filtro resuelve el problema de estimar variables ocultas, como la posición y velocidad de un objeto, utilizando sensores con ruido. El filtro tiene dos etapas principales: **Predicción** y **Actualización**.

# 2 Conceptos Estadísticos Fundamentales

- **Media** ($\mu$): Valor central de los datos.
- **Valor Esperado** ($E[x]$): Mejor estimación de una variable aleatoria.
- **Varianza** ($\sigma^2$) y **Desviación Estándar** ($\sigma$): Miden la dispersión de los datos.

# 3 Funcionamiento del Filtro de Kalman

El filtro de Kalman se compone de dos etapas:

1. **Predicción**: Estima el siguiente estado a partir del modelo.
2. **Actualización**: Corrige la predicción con la medición más reciente.

El filtro actúa como un *promediador inteligente*, ponderando las fuentes según su precisión.

## Precisión vs. Exactitud

- Un sensor preciso tiene baja varianza pero puede ser inexacto.
- Un sensor exacto mide cerca del valor verdadero, pero puede tener alta dispersión.

# 4 Filtro $\alpha$-$\beta$-($\gamma$)

Este filtro es una versión simplificada del Filtro de Kalman para sistemas donde el modelo es conocido y el error es constante.

### Funcionamiento

- **Predicción**: Basada en el modelo del sistema.
- **Actualización**: Ajusta el estado con las mediciones, usando coeficientes $\alpha$, $\beta$, y $\gamma$.

### Relación con el Filtro de Kalman

El filtro $\alpha$-$\beta$-($\gamma$) es más simple y ligero, pero menos preciso en sistemas complejos.

# 5 Filtro de Kalman Multivariable

Para sistemas multivariables, como el de un avión con posición, velocidad y aceleración, el filtro de Kalman utiliza matrices para representar el estado del sistema:

$$x = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}$$

El filtro predice y corrige el estado utilizando las matrices $P$, $F$, $H$, $Q$, y $R$.

# 6 Filtro de Kalman No Lineal

Los sistemas no lineales requieren aproximaciones, y se usan variantes como el Filtro de Kalman Extendido (EKF) y el Filtro de Kalman No Perfumado (UKF).

### Filtro de Kalman Extendido (EKF)

El EKF linealiza el sistema no lineal alrededor del punto de operación usando derivadas parciales.

### Filtro de Kalman No Perfumado (UKF)

El UKF utiliza la **Transformada No Perfumada** (Unscented Transform) para manejar no linealidades sin derivadas.

# 7 Comparación entre EKF y UKF

| **Característica** | **EKF** | **UKF** |
|---|---|---|
| Manejo de no linealidades | Linealización analítica | Aproximación estadística |
| Precisión | Buena en no linealidad moderada | Mejor en alta no linealidad |
| Complejidad matemática | Alta (usa derivadas) | Moderada (sin derivadas) |
| Facilidad de implementación | Requiere más experiencia | Más directa |

Table 1: Comparación entre el EKF y el UKF

# 8 Fusión de Sensores

Los sistemas reales, como vehículos autónomos, pueden usar múltiples sensores (LiDAR, radar, GPS, INS), y los filtros de Kalman son clave para fusionar estos datos de manera precisa.

# 9 Implementación del Filtro de Kalman

En este proyecto se implementó un Filtro de Kalman para realizar el seguimiento de una pelota detectada en video. La detección se realiza utilizando el modelo YOLOv8, y el seguimiento se gestiona mediante un filtro de Kalman individual por cada objeto detectado.

## 9.1 Modelo del Filtro

Se utilizó un filtro de Kalman lineal con un estado de 4 dimensiones que representa la posición y velocidad en el plano 2D:

$$\mathbf{x} = [x, y, \dot{x}, \dot{y}]^T$$

La matriz de transición de estado $F$ considera un modelo de movimiento con velocidad constante:

$$F = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La matriz de observación $H$ asume que sólo se observan las posiciones $x$ e $y$:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Se inicializó el filtro con:

- Una gran incertidumbre en la matriz de covarianza del estado: $P = 1000 \cdot I$
- Un ruido de medición moderado: $R = 70 \cdot I$
- Un ruido del proceso pequeño: $Q = 0.1 \cdot I$

## 9.2 Inicialización en Código

La siguiente porción de código muestra cómo se define el filtro de Kalman para cada objeto:

```
self.kf = KalmanFilter(dim_x=4, dim_z=2)
self.kf.F = np.array([[1, 0, 1, 0],
                      [0, 1, 0, 1],
                      [0, 0, 1, 0],
                      [0, 0, 0, 1]])
self.kf.H = np.array([[1, 0, 0, 0],
                      [0, 1, 0, 0]])
self.kf.R *= 70
self.kf.P *= 1000
self.kf.Q *= 0.1
```

La posición inicial del objeto se obtiene a partir del centro del *bounding box* generado por YOLOv8 y se asigna al estado del filtro.

## 9.3 Ciclo de Predicción y Actualización

Cada tracker realiza dos pasos principales:

- **Predicción**: Se estima la próxima posición de la pelota con base en su estado actual.
- **Actualización**: Si una nueva detección tiene una Intersección sobre la Unión (IoU) suficientemente alta con la predicción, se actualiza el filtro con la nueva medición.

Cuando no hay coincidencias con las detecciones actuales, el tracker continúa prediciendo su estado, y se elimina tras un tiempo sin ser actualizado.

## 9.4 Ventajas de la Implementación

Este enfoque permite rastrear múltiples pelotas simultáneamente, incluso ante oclusiones temporales o detecciones perdidas. Además, se asigna a cada objeto un identificador y un color únicos, facilitando la visualización y el análisis individual de cada trayectoria.