

# Localización de robot con EKF con balizas

Alvaro, Poma

arpoma167@gmail.com

André, Enes

andre.banenes@gmail.com

Febrero 2023

## 1. Introducción

En este proyecto se trata de desarrollar técnicas de estimación de posición 3D para un robot aéreo basados en filtro de Kalman extendido (EKF) que integra las medidas tomadas por las balizas.

Se dispone de un escenario donde se han desplegado en posiciones conocidas balizas estáticas que miden su distancia a un robot aéreo en dicho escenario. En la realización de este proyecto se han llevado con Webots y ROS, utilizando Python y c++ como lenguaje de programación para la comunicación entre los distintos elementos y implementación del filtro de Kalman extendido.

Este proyecto se divide en :

- Desarrollo de EKF para un sistema de balizas
- Implementación
- Resultados y conclusiones

## 2. Desarrollo de EKF

Antes de empezar con el Filtro de Kalman Extendido (EKF) hablaremos un poco de filtro de Kalman, ya que este es una variación que permite utilizar el Filtro de Kalman en sistemas no lineales.

### 2.1. Filtro de Kalman

El filtro de Kalman es un algoritmo de estimación utilizado en el procesamiento de señales y control automático para estimar el estado de un sistema dinámico a partir de mediciones incompletas y ruidosas.

El filtro de Kalman utiliza la representación en espacio de estados de un modelo matemático: el modelo de estado, que describe cómo el estado del sistema evoluciona en el tiempo, y el modelo de medición, que describe cómo se relacionan las mediciones con el estado del sistema, para predecir el estado futuro del sistema, actualizar la estimación del estado actual del sistema y corregir las mediciones ruidosas.

$$x_{k+1} = A_d x_k + B_d e_k + w_k \quad (1)$$

$$y_k = C_d x_k + D_d u_k + v_k \quad (2)$$

Donde:

- $x_{k+1}$  es el vector de estados predecido en el tiempo  $k+1$
- $x_k$  es el vector de estados en el tiempo  $k$
- $u_k$  es el vector de control, la entrada del sistema

- $w_k$  es el ruido del proceso  $v_k$
- $A_d$  es la matriz de estados
- $B_d$  es la matriz de entrada
- $C_d$  es la matriz de salida
- $D_d$  es la matriz de transmisión directa
- $w_k$  la matriz de covarianza que describe el ruido del proceso
- $v_k$  la matriz de covarianza que describe la incertidumbre de la medición

### 2.1.1. Algoritmo del Filtro de Kalman

Filtro de Kalman es un algoritmo recursivo que se utiliza para estimar el estado de un sistema dinámico a partir de mediciones incompletas y ruidosas. El algoritmo se compone de dos fases principales: la fase de predicción y la fase de actualización.

En la fase de predicción se realiza la predicción del estado y la predicción de la covarianza del sistema y en la fase de actualización se realiza el cálculo de la ganancia de Kalman, actualización del estado y actualización de la covarianza del error de estimación como se indica en las siguientes ecuaciones:

Fase de predicción

$$\begin{aligned}\bar{x} &= Ax + Bu \\ \bar{P} &= APA^T + Q\end{aligned}$$

Fase de actualización

$$\begin{aligned}y &= z - C\bar{x} \\ K &= \bar{P}H^T(C\bar{P}C^T + R)^{-1} \\ x &= \bar{x} + Ky \\ P &= (I - KC)\bar{P}\end{aligned}$$

## 2.2. Filtro de Kalman Extendido

El Filtro de Kalman utiliza ecuaciones lineales por lo que no funciona con problemas no lineales, por lo que el Filtro extendido de Kalman o EKF es una modificación al filtro de Kalman que permite que este pueda utilizar en sistemas no lineales mediante la linealización de la ecuación, para sistemas no lineales  $Ax + Bu$  es remplazada por la función no lineal  $f(x_k, u_k)$  y la expresión lineal  $Cx$  es remplazada por la ecuación no lineal  $h(x_k, u_k)$

Modelo de predicción:

$$x_{k+1} = f(x_k, u_k) + w_k \quad (3)$$

Modelo de observación:

$$y_k = h(x_k, u_k) + v_k \quad (4)$$

El EKF no altera las ecuaciones del filtro de Kalman, se linealizan las ecuaciones no lineales para usar el filtro de Kalman. El EKF utiliza una aproximación lineal de las ecuaciones del modelo alrededor de un punto operativo o estimado tomando las derivadas parciales de cada una de las ecuaciones no lineales y evaluándolas en  $x$  o  $u$  correspondiente.

$$F_k := \frac{\partial f_d(x, u)}{\partial x^T} \Big|_{\hat{x}_{k|k}, u_k} \quad (5)$$

$$H_k := \frac{\partial h(x, u)}{\partial x^T} \Big|_{\hat{x}_{k|k}, u_k} \quad (6)$$

A las derivadas parciales de una matriz se le llama Jacobiano.

$$\frac{\partial \vec{y}}{\partial \vec{x}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_n}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_n}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_n}{\partial x_1} & \frac{\partial y_n}{\partial x_2} & \cdots & \frac{\partial y_n}{\partial x_n} \end{bmatrix} \quad (7)$$

### 2.2.1. Algoritmo del Filtro de Kalman extendido

Las ecuaciones para el filtro de Kalman son las mismas del filtro de Kalman extendido donde solo se añade el cálculo del Jacobiano para las ecuaciones no lineales.

Fase de predicción :

$$F = \left. \frac{\partial f(x_t, u_t)}{\partial x} \right|_{x_t, u_t}$$

$$\bar{x} = f(x, u)$$

$$\bar{P} = F P F^T + Q$$

Fase de actualización:

$$H = \left. \frac{\partial h(\bar{x}_t)}{\partial \bar{x}} \right|_{\bar{x}_t}$$

$$y = z - h(\bar{x})$$

$$K = \bar{P} H^T (H \bar{P} H^T + R)^{-1}$$

$$x = \bar{x} + K y$$

$$P = (I - K H) \bar{P}$$

## 2.3. Determinación de posición usando Balizas

Para el desarrollo de la siguiente se parte se basa de un robot que mueve en un espacio de 3 dimensiones de forma lineal y en el cual se encuentra instalado un receptor de ondas de radio que mediante la calidad de la señal recibida de unas emisores o beacons puede conocer la distancia a cada uno de ellos.

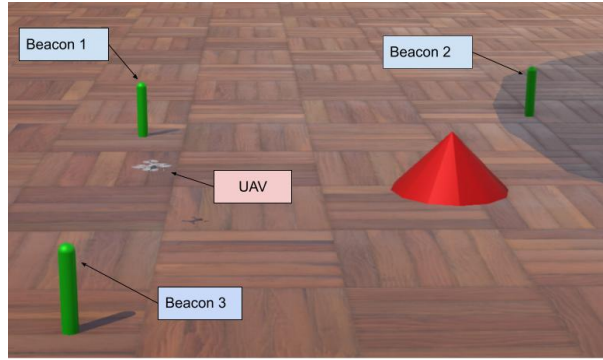


Figura 1: Balizas y UAV

### 2.3.1. Diseño de variables de estado

Para el diseño se supone que el UAV pose una velocidad constante en x,y,z, para lo cual necesitaremos 6 variables de estado distancia en x , distancia en y, distancia en z , velocidad en x , velocidad en y y velocidad en z. El sistema tiene un modelo de movimiento lineal pero posee un modelo del sensor no lineal.

$$X = \begin{bmatrix} distancia_x \\ distancia_y \\ distancia_z \\ velocidad_x \\ velocidad_y \\ velocidad_z \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ z_k \\ Vx_k \\ Vy_k \\ Vz_k \end{bmatrix} \quad (8)$$

### 2.3.2. Diseño el modelo de proceso

Para el Diseño del modelo se asume que el movimiento del UAV es un movimiento rectilíneo uniforme MRUV siguiendo las ecuaciones cinemáticas de Newton.

(9)

$$F = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

Cada una de las variables de distancia es independiente a la otra y posee una velocidad.

$$\dot{x} = v_x$$

$$\ddot{a}_x = \dot{v}_x = 0$$

Colando las ecuaciones diferenciales en forma de variables de estados

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + w_k$$

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ z_{k+1} \\ Vx_{k+1} \\ Vy_{k+1} \\ Vz_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_k \\ y_k \\ z_k \\ Vx_k \\ Vy_k \\ Vz_k \end{bmatrix} + \begin{bmatrix} q^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & q^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & q^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & q^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & q^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & q^2 \end{bmatrix} \quad (11)$$

### 2.3.3. Diseño de modelo de mediciones

La función de medición toma la estimación del vector del estos y la convierte en una medida de la distancia del UAV a cada una de las balizas usando el Teorema de Pitagoras.

Obtener la posición de un objeto de interés a partir de las distancias a posiciones de localización conocidas. La localización basándose en la medición de distancias a ciertos puntos de referencia de localización conocida (balizas ), para la localización de un objeto móvil.

$$\begin{aligned} r_2^2 &= (x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 \\ r_3^2 &= (x_1 - x_3)^2 + (y_1 - y_3)^2 + (z_1 - z_3)^2 \\ r_4^2 &= (x_1 - x_4)^2 + (y_1 - y_4)^2 + (z_1 - z_4)^2 \end{aligned} \quad (12)$$

El modelo de observación esta dado por la ecuación de las balizas

$$z_k = \begin{bmatrix} rB1_k \\ rB2_k \\ rB3_k \end{bmatrix} = \begin{bmatrix} \sqrt{(x_k - xB_1)^2 + (y_k - yB_1)^2 + (z_k - zB_1)^2} \\ \sqrt{(x_k - xB_2)^2 + (y_k - yB_2)^2 + (z_k - zB_2)^2} \\ \sqrt{(x_k - xB_3)^2 + (y_k - yB_3)^2 + (z_k - zB_3)^2} \end{bmatrix} \quad (13)$$

Como se vio anteriormente el vector de estado es de dimensión 6 por lo que se ampliara la matriz para poder tener un modelo que sea totalmente observable mediante la observación de un estado pasado.

$$Vx_k = \frac{x_k - x_{k-1}}{T}$$

$$x_{k-1} = x_k - TVx_k$$

y esta ultima ecuación remplazando en la distancia anterior a cada baliza en cada una de las dimensiones tenemos

$$rB1_{k-1} = \sqrt{(x_{k-1} - xB_1)^2 + (y_{k-1} - yB_1)^2 + (z_{k-1} - zB_1)^2}$$

Obtenemos el nuevo modelo de observación

$$h(x_k) = \begin{bmatrix} rB1_k \\ rB2_k \\ rB3_k \\ rB1_{k-1} \\ rB2_{k-1} \\ rB3_{k-1} \end{bmatrix} = \begin{bmatrix} \sqrt{(x_k - xB_1)^2 + (y_k - yB_1)^2 + (z_k - zB_1)^2} \\ \sqrt{(x_k - xB_2)^2 + (y_k - yB_2)^2 + (z_k - zB_2)^2} \\ \sqrt{(x_k - xB_3)^2 + (y_k - yB_3)^2 + (z_k - zB_3)^2} \\ \sqrt{(x_k - xB_1 - TVx_k)^2 + (y_k - yB_1 - TVy_k)^2 + (z_k - zB_1 - TVz_k)^2} \\ \sqrt{(x_k - xB_2 - TVx_k)^2 + (y_k - yB_2 - TVy_k)^2 + (z_k - zB_2 - TVz_k)^2} \\ \sqrt{(x_k - xB_3 - TVx_k)^2 + (y_k - yB_3 - TVy_k)^2 + (z_k - zB_3 - TVz_k)^2} \end{bmatrix} \quad (14)$$

La realización entre la distancia y la posición del UAV no es lineal debido a la raíz cuadrada, por lo cual se linealiza y evalúa su derivada parcial.

$$H_k := \frac{\partial h(x, u)}{\partial x^T} \Big|_{\hat{x}_k, u_k}$$

$$H_k = \begin{bmatrix} \frac{\partial rB1_k}{\partial x_k} & \frac{\partial rB1_k}{\partial y_k} & \frac{\partial rB1_k}{\partial z_k} & \frac{\partial rB1_k}{\partial Vx_k} & \frac{\partial rB1_k}{\partial Vy_k} & \frac{\partial rB1_k}{\partial Vz_y} \\ \frac{\partial rB2_k}{\partial x_k} & \frac{\partial rB2_k}{\partial y_k} & \frac{\partial rB2_k}{\partial z_k} & \frac{\partial rB2_k}{\partial Vx_k} & \frac{\partial rB2_k}{\partial Vy_k} & \frac{\partial rB2_k}{\partial Vz_y} \\ \frac{\partial rB3_k}{\partial x_k} & \frac{\partial rB3_k}{\partial y_k} & \frac{\partial rB3_k}{\partial z_k} & \frac{\partial rB3_k}{\partial Vx_k} & \frac{\partial rB3_k}{\partial Vy_k} & \frac{\partial rB3_k}{\partial Vz_y} \\ \frac{\partial rB4_k}{\partial x_k} & \frac{\partial rB4_k}{\partial y_k} & \frac{\partial rB4_k}{\partial z_k} & \frac{\partial rB4_k}{\partial Vx_k} & \frac{\partial rB4_k}{\partial Vy_k} & \frac{\partial rB4_k}{\partial Vz_y} \\ \frac{\partial rB5_k}{\partial x_k} & \frac{\partial rB5_k}{\partial y_k} & \frac{\partial rB5_k}{\partial z_k} & \frac{\partial rB5_k}{\partial Vx_k} & \frac{\partial rB5_k}{\partial Vy_k} & \frac{\partial rB5_k}{\partial Vz_y} \\ \frac{\partial rB6_k}{\partial x_k} & \frac{\partial rB6_k}{\partial y_k} & \frac{\partial rB6_k}{\partial z_k} & \frac{\partial rB6_k}{\partial Vx_k} & \frac{\partial rB6_k}{\partial Vy_k} & \frac{\partial rB6_k}{\partial Vz_y} \end{bmatrix} \quad (15)$$

Debido a que las ecuaciones no  $rB1_k, rB2_k, rB3_k$  no están en función de  $Vx_k, Vy_k, Vz_k$  la derivada parcial de estos es 0.

$$H_k = \begin{bmatrix} \frac{\partial rB1_k}{\partial x_k} & \frac{\partial rB1_k}{\partial y_k} & \frac{\partial rB1_k}{\partial z_k} & 0 & 0 & 0 \\ \frac{\partial rB2_k}{\partial x_k} & \frac{\partial rB2_k}{\partial y_k} & \frac{\partial rB2_k}{\partial z_k} & 0 & 0 & 0 \\ \frac{\partial rB3_k}{\partial x_k} & \frac{\partial rB3_k}{\partial y_k} & \frac{\partial rB3_k}{\partial z_k} & 0 & 0 & 0 \\ \frac{\partial rB4_k}{\partial x_k} & \frac{\partial rB4_k}{\partial y_k} & \frac{\partial rB4_k}{\partial z_k} & \frac{\partial rB4_k}{\partial Vx_k} & \frac{\partial rB4_k}{\partial Vy_k} & \frac{\partial rB4_k}{\partial Vz_y} \\ \frac{\partial rB5_k}{\partial x_k} & \frac{\partial rB5_k}{\partial y_k} & \frac{\partial rB5_k}{\partial z_k} & \frac{\partial rB5_k}{\partial Vx_k} & \frac{\partial rB5_k}{\partial Vy_k} & \frac{\partial rB5_k}{\partial Vz_y} \\ \frac{\partial rB6_k}{\partial x_k} & \frac{\partial rB6_k}{\partial y_k} & \frac{\partial rB6_k}{\partial z_k} & \frac{\partial rB6_k}{\partial Vx_k} & \frac{\partial rB6_k}{\partial Vy_k} & \frac{\partial rB6_k}{\partial Vz_y} \end{bmatrix} \quad (16)$$

donde las derivadas parciales se definen como:

$$\frac{\partial rB1_k}{\partial x_k} = \frac{x_k - xB_1}{\sqrt{(x_k - xB_1)^2 + (y_k - yB_1)^2 + (z_k - zB_1)^2}}$$

$$\frac{\partial rB1_{k-1}}{\partial Vx_k} = \frac{-T(x_k - xB_1 - TVx_k)}{\sqrt{(x_k - xB_3 - TVx_k)^2 + (y_k - yB_3 - TVy_k)^2 + (z_k - zB_3 - TVz_k)^2}}$$

$$\frac{\partial rB1_{k-1}}{\partial x_k} = \frac{(1 - Ta)(x_k - xB_1 - TVx_k)}{\sqrt{(x_k - xB_3 - TVx_k)^2 + (y_k - yB_3 - TVy_k)^2 + (z_k - zB_3 - TVz_k)^2}}$$

### 2.3.4. Diseño del proceso y ruido de medida

Para la medición de las balizas se basó en balizas comerciales específicamente en Baliza HW v4.9-IMU de Marvelmind la cual posee un sistema de navegación interior de alta precisión (+ -2 cm) y radio de cobertura de

una baliza: hasta 50 metros por lo cual se usa  $\phi_{range} = 0,02\text{metros}$

$$R = \begin{bmatrix} r^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & r^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & r^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & r^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & r^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & r^2 \end{bmatrix} \quad (17)$$

Para el valor de q hay que tener en cuenta que los primero elementos son la posición, seguidos de la velocidad, en este caso de colocara el mismo valor para todo y se determinara mediante experiencia y pruebas del comportamiento del filtro ante distintos valores de q.

Modelo de estado

$$W_k = \begin{bmatrix} q^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & q^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & q^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & q^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & q^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & q^2 \end{bmatrix} \quad (18)$$

Donde se tiene la siguiente ecuaciones en espacio de estados:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ z_{k+1} \\ Vx_{k+1} \\ Vy_{k+1} \\ Vz_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_k \\ y_k \\ z_k \\ Vx_k \\ Vy_k \\ Vz_k \end{bmatrix} + \begin{bmatrix} q^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & q^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & q^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & q^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & q^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & q^2 \end{bmatrix} \quad (19)$$

Modelo de Observación

$$z_k = C\mathbf{x}_k + v_k$$

$$\begin{bmatrix} rB1_k \\ rB2_k \\ rB3_k \\ rB1_{k-1} \\ rB2_{k-1} \\ rB3_{k-1} \end{bmatrix} = \begin{bmatrix} \frac{\partial rB1_k}{\partial x_k} & \frac{\partial rB1_k}{\partial y_k} & \frac{\partial rB1_k}{\partial z_k} & 0 & 0 & 0 \\ \frac{\partial rB2_k}{\partial x_k} & \frac{\partial rB2_k}{\partial y_k} & \frac{\partial rB2_k}{\partial z_k} & 0 & 0 & 0 \\ \frac{\partial rB3_k}{\partial x_k} & \frac{\partial rB3_k}{\partial y_k} & \frac{\partial rB3_k}{\partial z_k} & 0 & 0 & 0 \\ \frac{\partial rB4_k}{\partial x_k} & \frac{\partial rB4_k}{\partial y_k} & \frac{\partial rB4_k}{\partial z_k} & \frac{\partial rB4_k}{\partial Vx_k} & \frac{\partial rB4_k}{\partial Vy_k} & \frac{\partial rB4_k}{\partial Vz_k} \\ \frac{\partial rB5_k}{\partial x_k} & \frac{\partial rB5_k}{\partial y_k} & \frac{\partial rB5_k}{\partial z_k} & \frac{\partial rB5_k}{\partial Vx_k} & \frac{\partial rB5_k}{\partial Vy_k} & \frac{\partial rB5_k}{\partial Vz_k} \\ \frac{\partial rB6_k}{\partial x_k} & \frac{\partial rB6_k}{\partial y_k} & \frac{\partial rB6_k}{\partial z_k} & \frac{\partial rB6_k}{\partial Vx_k} & \frac{\partial rB6_k}{\partial Vy_k} & \frac{\partial rB6_k}{\partial Vz_k} \end{bmatrix} * \begin{bmatrix} x_k \\ y_k \\ z_k \\ Vx_k \\ Vy_k \\ Vz_k \end{bmatrix} + \begin{bmatrix} r^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & r^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & r^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & r^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & r^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & r^2 \end{bmatrix} \quad (20)$$

### 3. Implementación

Para la implementación se decidió realizar la simulación en Webots donde se poseen nodos de tipo recepción y emisor que será utilizado en las balizas y en UAV para detectar la distancia del UAV a cada una de estas, y estos datos se pasaran mediante ros a un fichero de Python que donde se realizara el filtro de Kalman para la estimación.

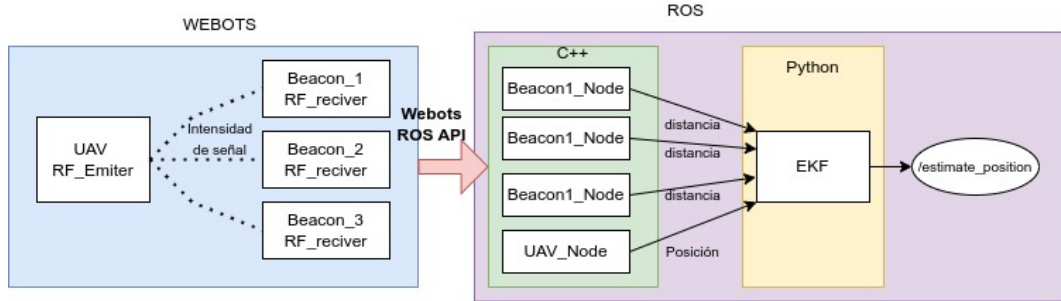


Figura 2: Modelo utilizado para las balizas

#### 3.1. Implementación de EKF con Python

Para el desarrollo de un modulo del EKF para la localización de un robot utilizando balizas se utilizo Python debido a la facilidad e interoperabilidad del mismo, para el desarrollo del EKF se uso de la librería Filterpy que ya implementa las ecuaciones de Kalman y donde se configuraron cada una de mas matrices del modelo de espacios de estados y la actualización del Jacobiano.

Además se desarrollo otra clase para la simulación de la posición del robot y calculo de las distancias de las balizas para pruebas del funcionamiento del filtro, también se desarrollo un archivo Jupyter Notebook para la visualización de los resultados.

#### 3.2. Webots

##### 3.2.1. Webots vs. Gazebo

Aunque *Gazebo* es el simulador más utilizado con la interfaz ROS, su utilización requiere más recursos de la CPU y *Webots* tiene dos modelos de quadcopters fácilmente modificables, *DJI Mavic2Pro* y *Crazyflie*. En un proyecto futuro, *Gazebo* es más recomendable, porque todo se basa en ROS y en *Webots*, la mayor parte de los modelos ya desarrollados no usan ROS y su integración no es tan cómoda.

##### 3.2.2. Quadcopter

El modelo de quadcopter elegido fue el *DJI Mavic2Pro*. Su controlador básico se basa en GPS, pero en *Webots* es posible cambiar los modelos y añadir un *Emitter* que funciona como que emite simula la emision de una señal de radio frecuencia es detectada por el *receiver*. El *receiver* tiene la función `wb_receiver_get_signal_strength` y la capacidad de recibir señales de *emitters* diferentes. Con estas funcionalidades, es posible determinar la distancia de cada baliza hasta el quadcopter.



Figura 3: Modelo *DJI Mavic2Pro*

### 3.2.3. Balizas

Para simular las balizas, se utiliza el sensor *receiver*, ya que nos permite de forma simple tener la distancia al *emitter* mediante `wb_receiver_get_signal_strength`. Es posible simular una señal IR o RF, pero para esta aplicación tiene más sentido utilizar RF. Para la transmisión también es posible simular y controlar el ruido gaussiano de las señales.

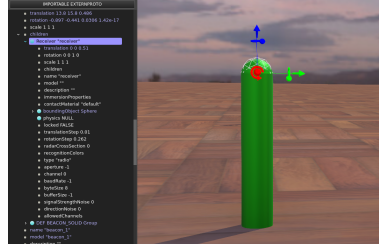


Figura 4: Modelo utilizado para las balizas

la función `wb_receiver_get_signal_strength` devuelve la intensidad de señal simulada en el momento en que se transmitió el paquete. Esta intensidad de señal es igual a la inversa de la distancia entre el emisor y el receptor al cuadrado. En otras palabras,  $s = 1 / r^2$ , donde  $s$  es la intensidad de la señal y  $r$  es la distancia entre el emisor y el receptor, de esta forma podremos simular la baliza.

### 3.2.4. Integración con ROS

La integración de ros se realiza mediante el cambio de controlador de los robots a ros, este lo que realiza es levantar servicios de cada uno de los elementos que tiene el robot. En el caso del UAV se tiene que agregar un robot al uav para colocar el controlador ROS.

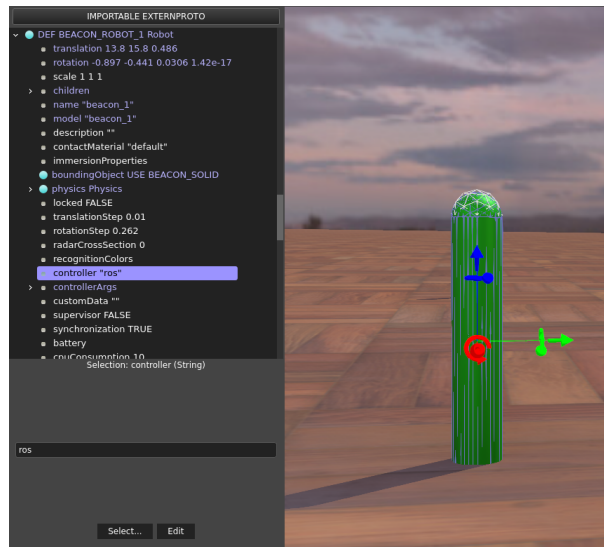


Figura 5: Estructura de un modelo de un robot en Webots

## 3.3. ROS

Para la obtención de los datos de Webots se realizaron un nodo para cada baliza, un nodo para el UAV y un nodo para el EKF.

### 3.3.1. Baliza

Mediantes los servicios de ros que se publica al tener el controlador de ros cada robot en Webots se llama a la intensidad de la señal para calcular la distancia al UAV para publicar a un topic la distancia al UAV en



metros.

### 3.3.2. UAV

Se publica la posición del UAV en un topic para poder comparar la posición del UAV con la posición estimada, lo cual servirá para las pruebas, además este nodo se encarga de publicar datos en el emisor para sean enviados al receptor y este pueda saber la posición a la que se encuentra el UAV mediante la intensidad de la señal que recibe el receptor.

### 3.3.3. EKF

El nodo del EKF se encarga de obtener los datos de cada uno de las balizas y para poder estimar la posición del UAV

## 4. Pruebas

Para el desarrollo de las pruebas se tomaron distintas situaciones de interés, donde se pretende demostrar el funcionamiento del filtro. Como base para las pruebas se toma el recorrido de un UAV en forma de Z, donde se utilizo Jupyter notebook para como herramienta de visualización para demostrar resultados.

### 4.1. Prueba 1

Para la ejecución de la prueba se tienen los siguientes parámetros:

Posición de balizas  $[[1.0, 1.0, 1.5], [10.0, 7.0, 1.0], [1.0, 5.0, 0.0]]$

Ruido de medida 0.1 metros.

Ruido de modelo 0.044

Numero de balizas 3

Sin perdida de baliza

Posición del robot y la estimación:

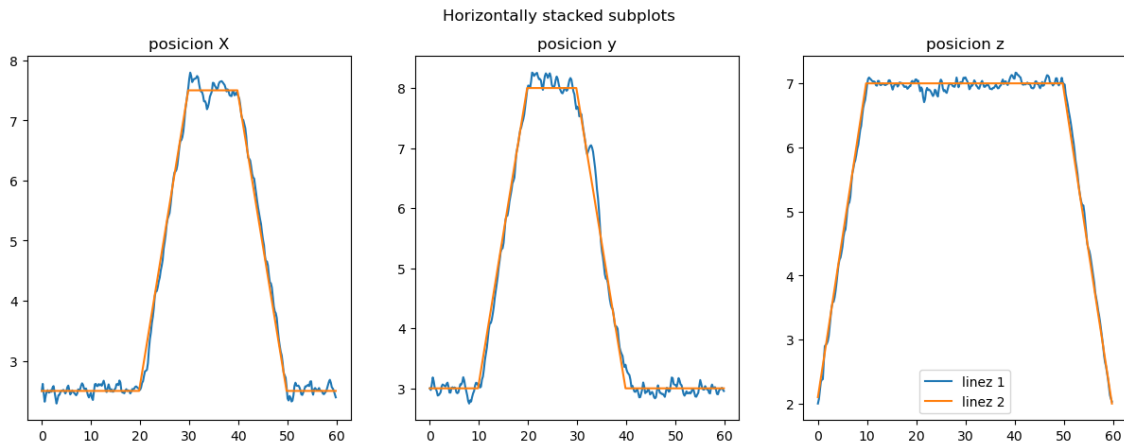


Figura 6: Prueba 1 estimación de posición

Ubicación de robot en 3D

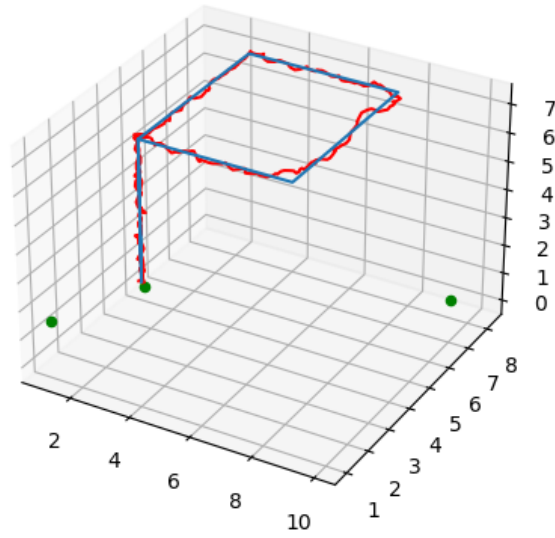


Figura 7: Prueba 1 estimación de posición 3D

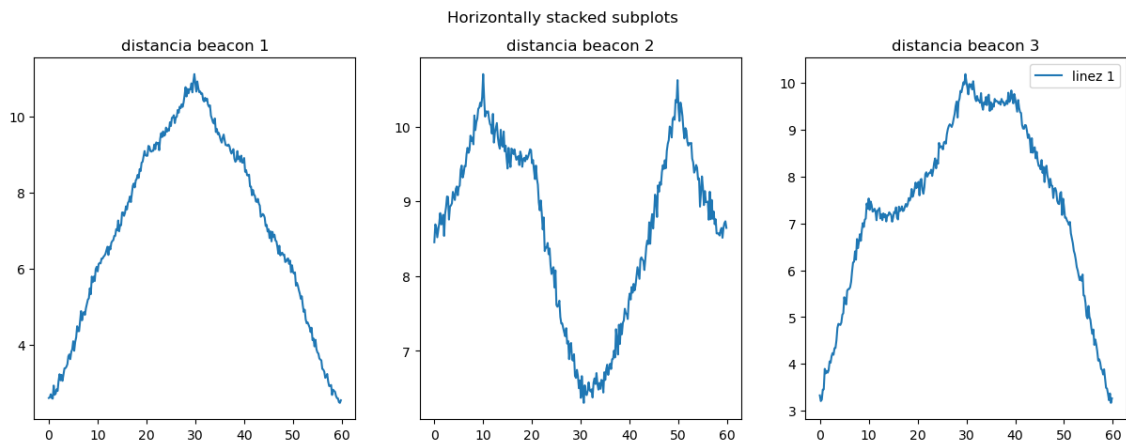


Figura 8: Prueba 1 medidas de balizas

## 4.2. Prueba 2

Para la ejecución de la prueba se tienen los siguientes parámetros:

Posición de balizas  $[[1.0, 1.0, 1.5], [10.0, 7.0, 1.0], [1.0, 5.0, 0.0]]$

Ruido de medida 0.1 metros.

Ruido de modelo 1

Numero de balizas 3

Sin perdida de baliza

Posición del robot y la estimación:

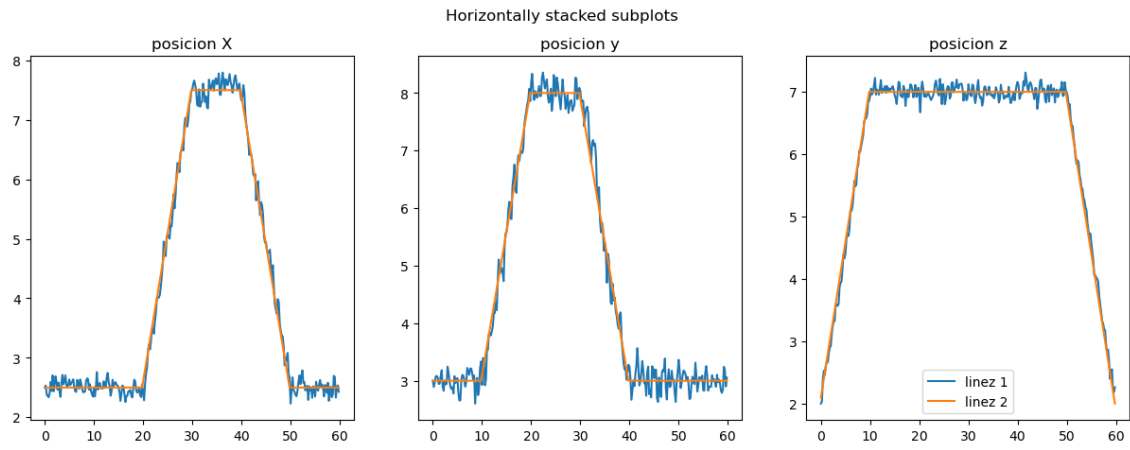


Figura 9: Prueba 2 estimación de posición

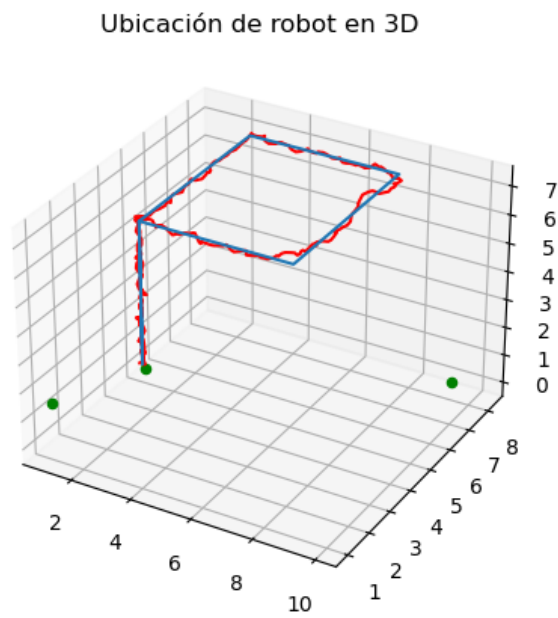


Figura 10: Prueba 2 estimación de posición 3D

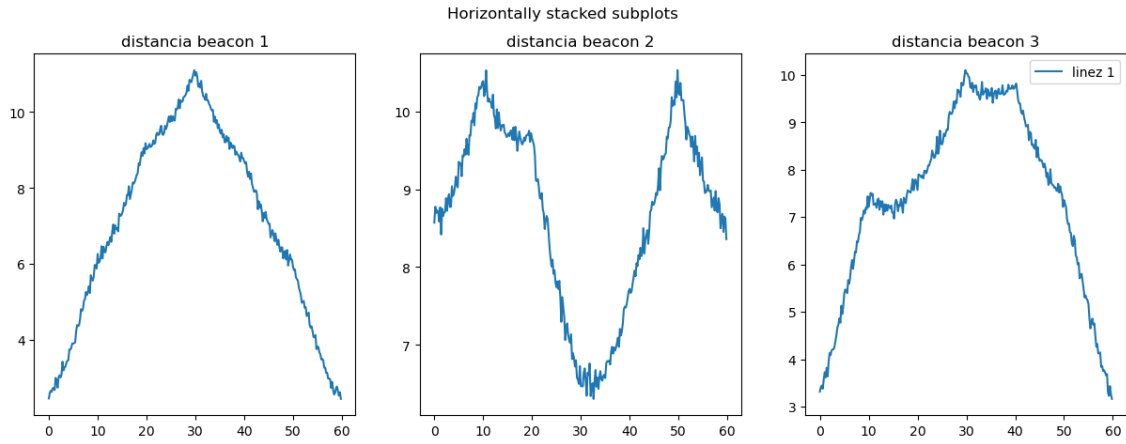


Figura 11: Prueba2 medidas de balizas

Se pude visualizar que con respecto al filtro anterior la estimación es menos suave, a lo que la posición se mantiene constante esto se debe a que el modelo del estado indica que va estar en movimiento y al ser  $Q$  mayor que  $R$  el filtro sigue mas el comportamiento del modelo en su estimación.

### 4.3. Prueba 3

Para la ejecución de la prueba se tienen los siguientes parámetros:

Posición de balizas  $[[1.0, 1.0, 1.5], [10.0, 7.0, 1.0], [1.0, 5.0, 0.0], [5.0, 0.0, 0.0]]$

Ruido de medida 0.1 metros.

Ruido de modelo 0.044

Numero de balizas 4

Sin perdida de baliza

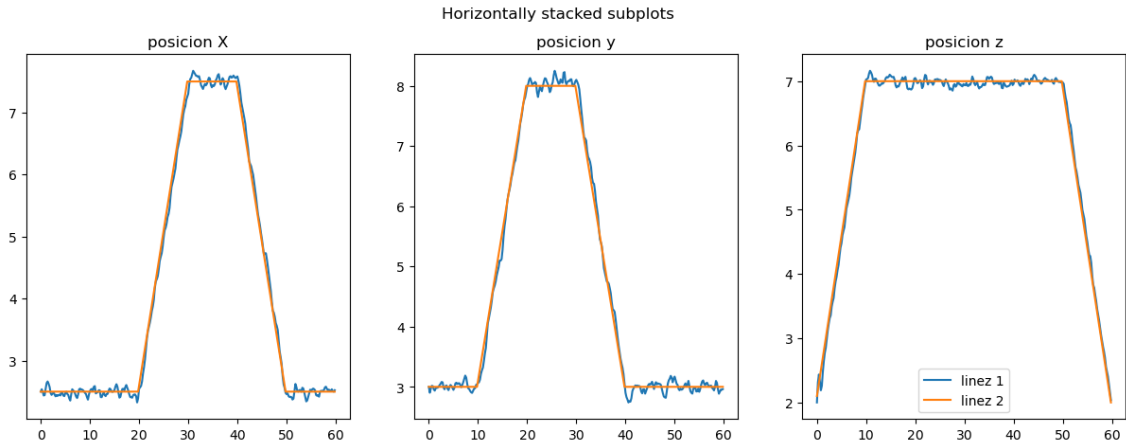


Figura 12: Prueba 3 estimación de posición

Ubicación de robot en 3D

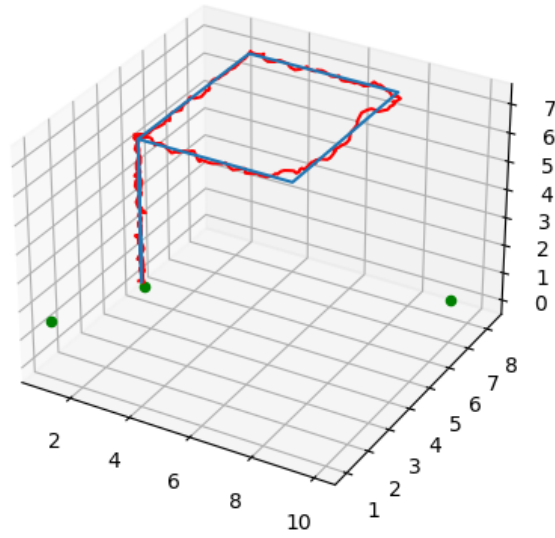


Figura 13: Prueba 3 estimación de posición 3D

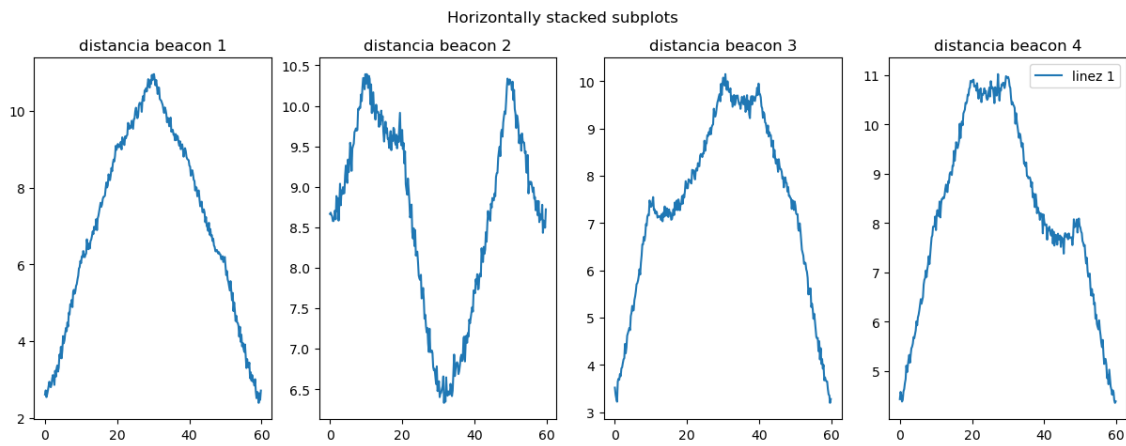


Figura 14: Prueba3 medidas de balizas

Se puede visualizar que el error en la medida disminuyó con respecto a la posición real del robot comparando con la prueba del 1.

#### 4.4. Prueba 4

Para el caso de la pérdida de la señal de las balizas se considera la baliza no detectar la señal de radio se supone que el robot se encuentra a una distancia mayor de 50 metros por lo que el filtro de Kalman no se ejecuta la fase de actualización y solo la de predicción.

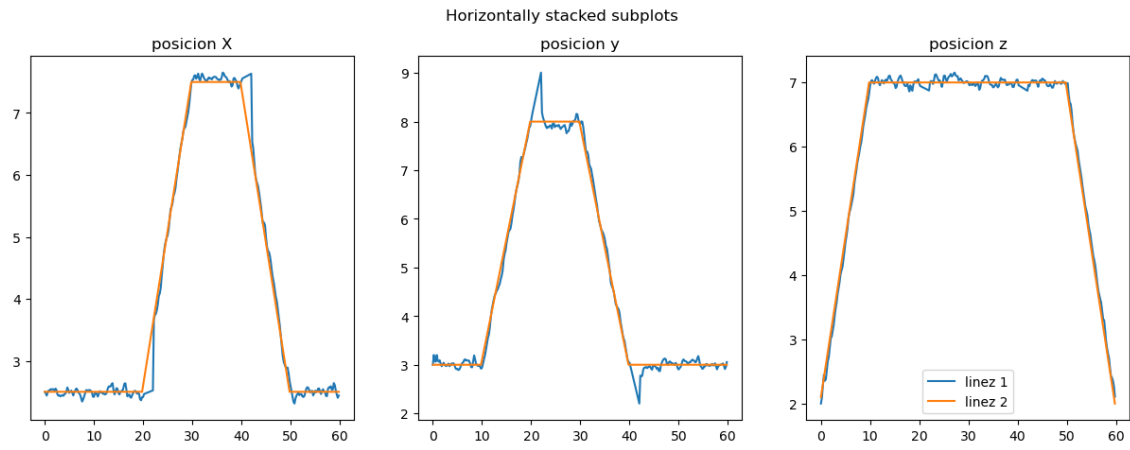


Figura 15: Prueba 4 estimación de posición

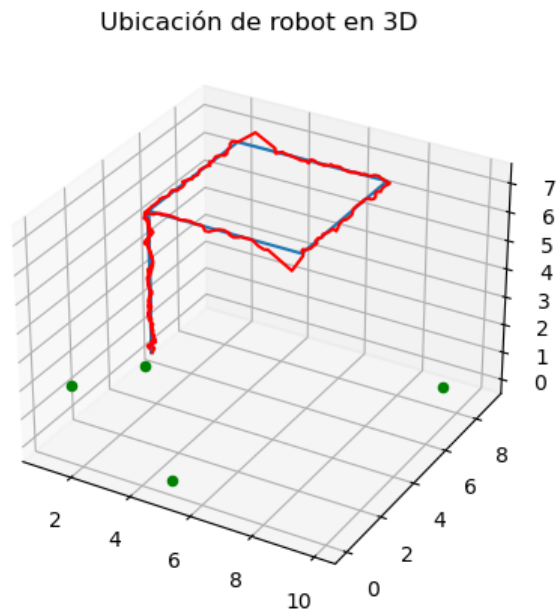


Figura 16: Prueba 4 estimación de posición 3D

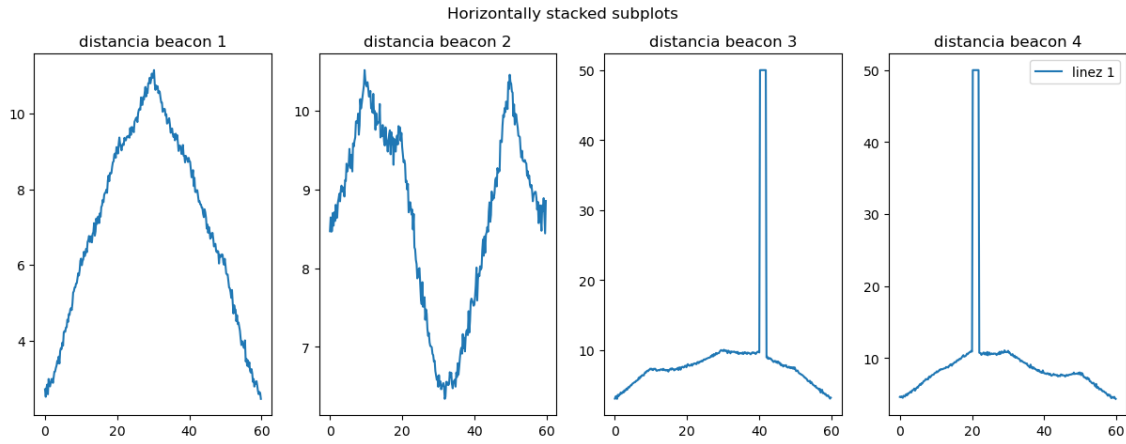


Figura 17: Prueba 4 medidas de balizas

En este caso donde el filtro de Kalman presenta perdida de datos se notan unos picos ya que trata de seguir el comportamiento del modelo.

#### 4.5. Prueba 5

Para la ejecución de la prueba se tienen los siguientes parámetros:

Posición de balizas  $[[1.0, 1.0, 0.0], [2.0, 3.0, 0.0], [0.0, 0.0, 0.0]]$

Ruido de medida 0.1 metros.

Ruido de modelo 0.044

Numero de balizas 3

Sin perdida de baliza

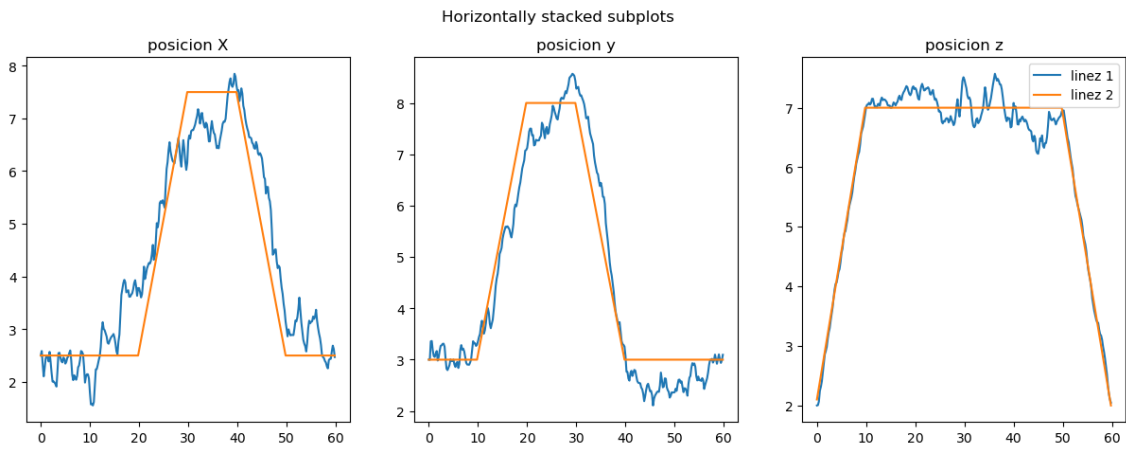


Figura 18: Prueba 5 estimación de posición

Ubicación de robot en 3D

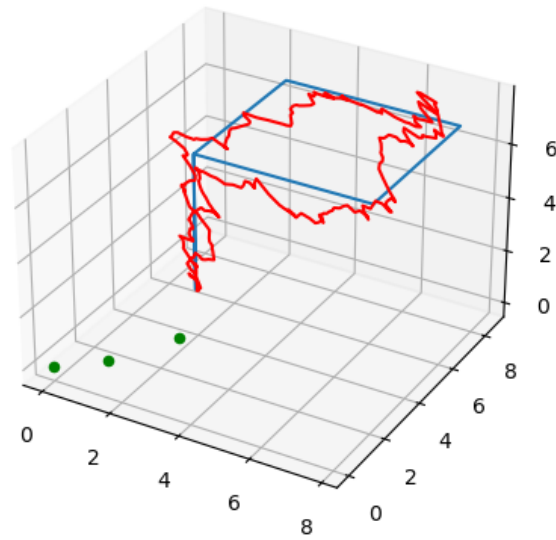


Figura 19: Prueba 5 estimación de posición 3D

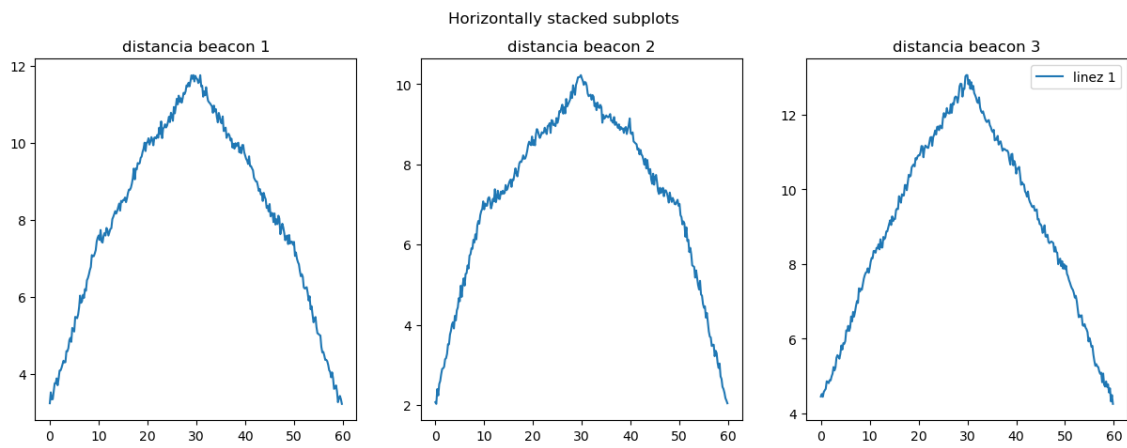


Figura 20: Prueba 5 medidas de balizas

En esta prueba se puede visualizar que el comportamiento del filtro depende de la posición de las balizas ya que al estar muy cerca entre ellas están aportan menos información al filtro para estimar la localización del UAV.



## 5. Conclusiones

El filtro de Kalman extendido que la estimación del UAV depende de la información que se tiene del modelo, mientras mas balizas se encuentren en el modelo preciso es la estimación, como información también se tiene que tener en cuenta la distribución de las balizas ya que mientras mas las posiciones entre estas sean mas diferentes mejor sera la estimación del filtro.

La estimación del Filtro de Kalman depende directamente del comportamiento de modelo, mientras mejor sea el modelo mejor sera la estimación, que se puede cambiar el comportamiento de la estimación dependa mas del modelo o de la medición según la relación de la covarianza del modelo y la covarianza de la medición siendo la covarianza del modelo la que se ha obtenido mediante la realización de pruebas y la covarianza de la medición es la de la baliza, ya que si  $Q > R$  el Filtro tratara de comportarse mas como el modelo y si  $Q < R$  el filtro seguirá mas el comportamiento de medidas observadas.

Cabe destacar que Webots es un sistema con una gran cantidad de modelos y sensores y actuadores que se puede simular en robot y dispone de una gran cantidad de robots ya probados y permite la integración con ros lo que lleva a ser muy complicado en un inicio, debido a que se debe implementar un controlador en Webots para cada robot.

## 6. Referencias

- [https://www.kalmanfilter.net/ES/kalman1d\\_es.html](https://www.kalmanfilter.net/ES/kalman1d_es.html)
- <https://www.sciencedirect.com/science/article/pii/S1569190X22001046>
- Thrun, Sebastian, Burgard, Wolfram and Fox, Dieter. Probabilistic robotics. Cambridge, Mass.: MIT Press, 2005.
- <https://cyberbotics.com/doc/reference/receiver>
- <https://cyberbotics.com/doc/reference/emitter>
- [https://github.com/cyberbotics/webots\\_ros](https://github.com/cyberbotics/webots_ros)