



UNIVERSIDAD DE EXTREMADURA

FACULTAD DE CIENCIAS

**Grado en FÍSICA**

**MEMORIA DEL TRABAJO FIN DE GRADO**

**ESTUDIO DEL POSICIONAMIENTO DE  
UN ROBOT MÓVIL MEDIANTE SENSORES  
"ULTRA WIDE BAND" (UWB)**

DANIEL ALFONSO DURÁN GARCÍA  
SEPTIEMBRE, 2020



Carlos Javier García Orellana, profesor del Departamento de Ingeniería Eléctrica, Electrónica y Automática de la Universidad de Extremadura.

INFORMA:

Que D. Daniel Alfonso Durán García ha realizado bajo su dirección el Trabajo Fin de Grado. Consideran que la memoria reúne los requisitos necesarios para su evaluación.

Badajoz, día de mes de 2020

Fdo. Carlos Javier García Orellana

# Índice

<b>1. Introducción y objetivos</b>	<b>4</b>
1.1. Introduccion . . . . .	4
1.2. Objetivos . . . . .	6
<b>2. Antecedentes</b>	<b>7</b>
2.1. Técnicas de estimación de la posición . . . . .	7
2.1.1. <i>Fingerprinting</i> . . . . .	7
2.1.2. Técnicas geométricas . . . . .	7
2.1.2.1. Tiempo de llegada (TOA) . . . . .	8
2.1.2.2. Diferencia de tiempo de llegada (TDOA) . . . . .	10
2.1.2.3. Intensidad de la señal recibida (RSSI) . . . . .	11
2.1.2.4. Ángulo de llegada (AOA) . . . . .	12
2.1.3. Mitigación en casos no ideales . . . . .	12
2.2. Tecnologías . . . . .	13
2.2.1. Sonido . . . . .	13
2.2.2. Luz . . . . .	13
2.2.3. Visión por computador . . . . .	14
2.2.4. Campo magnético . . . . .	14
2.2.5. Wi-Fi . . . . .	14
2.2.6. Bluetooth . . . . .	15
2.2.7. Otras tecnologías . . . . .	15
2.3. Ultra-Wide Band . . . . .	16
2.3.1. Ventajas . . . . .	16
2.3.2. Pulsos de UWB . . . . .	17
2.4. Robot . . . . .	20
2.5. Sensores UWB . . . . .	22
<b>3. Metodología</b>	<b>25</b>
3.1. Simulación previa . . . . .	25
3.2. Toma de medidas . . . . .	26
3.3. Laboratorio de Robótica del ICCAEx . . . . .	27
3.4. Edificio de la Facultad de Física . . . . .	29
<b>4. Resultados</b>	<b>31</b>
4.1. Laboratorio de Robótica del ICCAEx . . . . .	32
4.1.1. 4 balizas . . . . .	32
4.1.2. 6 balizas . . . . .	34

4.2. Edificio de Física . . . . .	37
4.2.1. 4 balizas . . . . .	37
4.2.2. 6 balizas . . . . .	39
4.2.3. 8 balizas . . . . .	40
<b>5. Conclusiones</b>	<b>44</b>
<b>6. Bibliografía</b>	<b>46</b>
<b>7. Apéndice 1</b>	<b>49</b>
7.1. Medidas en el laboratorio de Robótica del ICCAEx . . . . .	49
7.1.1. Medidas con 4 balizas . . . . .	49
7.1.2. Medidas de cada trayectoria con 4 balizas . . . . .	50
7.1.3. Medidas con 6 balizas . . . . .	53
7.1.4. Medidas de cada trayectoria con 6 balizas . . . . .	54
7.2. Medidas en el edificio de la Facultad de Física . . . . .	57
7.2.1. Medidas con 8 balizas . . . . .	57
7.2.2. Medidas con 6 balizas . . . . .	58
7.2.3. Medidas con 4 balizas . . . . .	59
<b>8. Apéndice 2</b>	<b>60</b>

# Resumen

Los sistemas de posicionamiento global, basados en la señal de satélites, no presentan una gran precisión en ambientes interiores debido a la falta de visión directa y la atenuación que conlleva, por lo que en estos escenarios es necesario emplear técnicas distintas para lograr un posicionamiento similar.

Dentro de las distintas tecnologías empleadas el Ultra-wide Band (UWB), que usa señales de al menos 500MHz de ancho de banda, está ganando poco a poco popularidad por ser una tecnología poco intrusiva y efectiva en un espectro electromagnético muy poblado, con una precisión en el orden del centímetro, pero muy dependiente del entorno donde se emplea.

Así, en este trabajo se plantean dos entornos para evaluar, en condiciones de uso ideales y no ideales, la precisión del kit comercial de UWB DecaWave MEDK1001 con la ayuda de un robot. Con varias configuraciones de los sensores en ambos escenarios, se comparó el desempeño del kit en diversas situaciones para determinar cómo afectan los obstáculos a las estimaciones de posición que proporciona.

En ambos escenarios se pudo comprobar que la disposición de las balizas es el principal factor a la hora de obtener el menor error posible en el posicionamiento.

# Abstract

Global positioning systems, based on satellite signals, are not able to provide a high level of precision in indoor environments due to the lack of direct line of sight and the attenuation that this entails. In these scenarios it is necessary to use different techniques in order to achieve a similar positioning.

Among the technologies used, Ultra-wide Band, which uses pulses of at least 500MHz bandwidth, has been gaining traction as it is lowly intrusive and highly effective in a saturated electromagnetic spectrum giving a precision in the order of centimeters depending on the scenario.

In this work two locations are chosen, with ideal and non-ideal conditions, to evaluate the precision of the DecaWave MDEK1001 commercial kit using a robot. Testing various setups in both scenarios, the performance of the kit in several situations was compared to determine how obstacles affect the position estimations it provides.

In both scenarios, it can be seen that the anchors setup is the main factor in obtaining the least possible positioning error.

# 1. Introducción y objetivos

## 1.1. Introduccion

El éxito de los sistemas de posicionamiento global como el GPS ha iniciado en los últimos años el desarrollo de tecnologías y experiencias basadas en el control y el seguimiento de la posición de objetos o personas.

En ambientes exteriores y abiertos, la precisión de este tipo de sistemas, con el equipamiento adecuado, puede llegar a ser de centímetros. No es el caso en situaciones interiores, donde la nula línea de visión directa entre los medidores y los satélites hacen que esta precisión sea imposible, por lo que su uso, en general, es inviable.

Debido a estos inconvenientes han aparecido nuevas técnicas para conseguir un posicionamiento similar, pero de forma local. Aunque su objetivo es el mismo, los entornos interiores requieren una precisión mayor y tienen, en general, un espectro electromagnético más poblado, por lo que se deben emplear tecnologías distintas compatibles con estos escenarios.

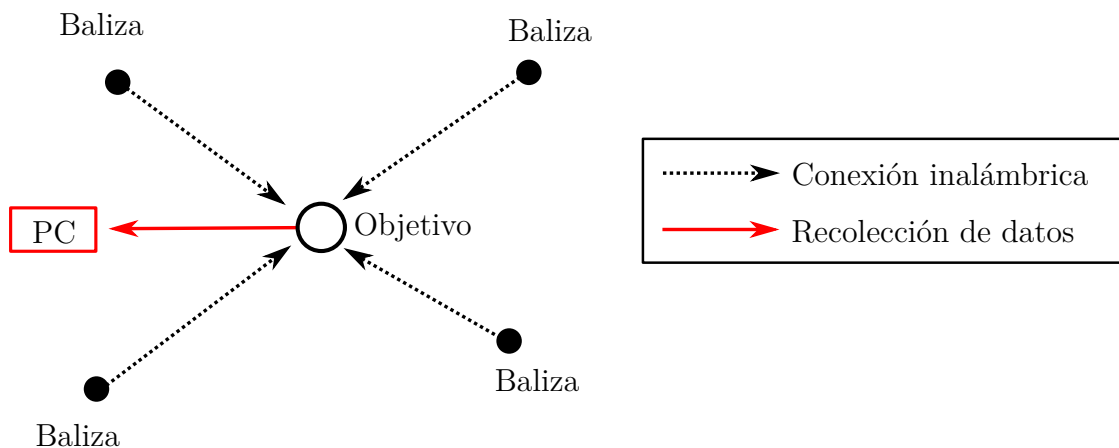


Figura 1: Esquema general de un sistema de posicionamiento local.

A diferencia del caso del posicionamiento global, existen dos tipos de algoritmos: unos basados en argumentos geométricos a partir de emisores con posición conocida, y otros consistentes en un mapeado previo de la zona de interés. Estas técnicas y las tecnologías sobre las que se implementan son las principales diferencias entre las alternativas comerciales disponibles.

La proliferación de la comunicación inalámbrica ha provocado una saturación en el espectro electromagnético, de tal forma que nuevos usos que puedan emplear este tipo de ondas se vean seriamente afectados por interferencias en la frecuencias en uso.

Aunque su planteamiento se remonta varias decenas de años en el tiempo, es cada vez más habitual la mención al Ultra-wide Band (UWB), una tecnología que usa un espectro de ondas de radio con un ancho de banda superior al 20 % de la frecuencia

central o mayor que 500 MHz.

Su gran ancho de banda permite la emisión de pulsos muy cortos en el tiempo y de baja energía, lo que hace que su implantación tenga una alta eficiencia. Además no requiere diseños complejos para la emisión ni para la recepción de pulsos, con lo que el equipamiento necesario para su uso tendrá en general un bajo coste.

La baja tolerancia que proporciona un uso tan amplio de frecuencias a las interferencias –a diferencia del Bluetooth o Wi-Fi, con canales con ancho de banda mucho más pequeño– la hacen muy atractiva para su uso masivo teniendo en cuenta que es posible asegurar un buen funcionamiento independientemente del uso simultáneo de otras tecnologías.

El estudio de la precisión disponible del UWB es el principal foco de este trabajo. La baja potencia de los pulsos usados para el posicionamiento hace que la presencia de obstáculos sea su principal punto débil, con una capacidad de penetración muy baja.

En escenarios con una configuración en la que el objetivo del posicionamiento no tenga línea visión directa de las balizas no será posible tener mediciones excesivamente precisas, y determinar los límites en dichos casos será crítico para evaluar el uso del UWB o de alguna de sus alternativas.

Para automatizar la toma de datos en la medida de lo posible se ha usado un robot en el que colocar el sensor de UWB a posicionar, de tal forma que ha sido posible una toma de datos con un volumen mayor a comparación de una metodología manual.

El robot utilizado ha sido el Turtlebot 2, un pequeño robot con fines educativos que permite el uso de cámaras y sensores láser para labores de movimiento autónomo, de tal forma que permite un movimiento total en un entorno controlado. Aunque es posible una navegación libre a partir de los sensores mencionados, para este trabajo se creó un mapa de los dos escenarios en los que se produjo la toma de medidas para que su posicionamiento local fuese lo más preciso posible.

Dicho robot fue empleado en dos escenarios para poder comparar el desempeño de los sensores de UWB en dos situaciones distintas. Así, en la elección los dos escenarios se primó la posibilidad de comparar una situación ideal, con visión directa y distancias cortas; y una situación más parecida a un uso habitual, con un escenario en el que, en ocasiones, pueda haber obstáculos que dificulten el traslado de la señal de UWB.

La primera toma de medidas se llevó a cabo en uno de los laboratorios de los Institutos Universitarios de Investigación en el que se disponía de una superficie amplia y libre; la segunda localización tuvo lugar en el edificio B de la Facultad de Física, donde por razones estructurales existen varias columnas en sus plantas y en ocasiones aparece un leve tráfico de personas.



## 1.2. Objetivos

En este trabajo se trata de evaluar la capacidad de la tecnología Ultra-wide Band para las tareas de posicionamiento local. Para este acometido es fundamental tener en cuenta el entorno en el que se intenta conseguir dicho posicionamiento y la disposición de todo el entramado de sensores para encontrar la configuración óptima.

Así, los objetivos de este trabajo son

- Evaluar la precisión de los sensores de Ultra-wide Band en un entorno favorable, de tal manera que es posible encontrar los límites de una forma fácilmente parametrizable y sin incluir fuentes adicionales de error.
- Evaluar de nuevo la precisión de los sensores en un segundo escenario menos favorable, con obstáculos que puedan interferir en las tareas de posicionamiento y así poder comprobar cómo afectan estas nuevas variables a los errores en la posición obtenida.
- Comparar el desempeño de la tecnología en ambos escenarios, y las discrepancias frente a los valores teóricos.
- En ambos casos, realizar medidas con distintas configuraciones de los sensores para evaluar cuáles de ellas arrojan mejores resultados y así encontrar el mejor equilibrio entre su desempeño y su complejidad.

## 2. Antecedentes

### 2.1. Técnicas de estimación de la posición

Existen dos alternativas para estimar la posición de un objetivo a partir de la posición conocida de un cierto número de balizas con coordenadas conocidas: la primera, denominada *fingerprinting*, se basa en la obtención de un modelo de posicionamiento a partir de la medición de la potencia de las balizas; la segunda, obtiene las estimaciones a partir de técnicas geométricas [1].

#### 2.1.1. *Fingerprinting*

La primera de las opciones para estimar la posición de un objetivo se basa en la medición de la potencia de la señal recibida desde cada una de las balizas.

Esta técnica requiere un mapeado previo de la zona de interés que permita obtener un modelo contra el que comparar en su uso posterior. La dependencia en un modelo previo provoca que cualquier modificación del entorno –como por ejemplo tránsito de personas– hace que la estimación sea en el mejor de los casos ligeramente distinta y en el peor, imposible.

La comparación contra el modelo hace uso de algoritmos de *machine learning* como *k-nearest-neighbor* o *support vector regression*, que buscan minimizar la diferencia entre las medidas y dicho modelo. Requieren, de forma general, una capacidad computacional mayor a las alternativas geométricas, por lo que son usados habitualmente en entornos con obstáculos estáticos y sin demasiadas modificaciones, donde su ventaja es significativa.

#### 2.1.2. Técnicas geométricas

La multilateración – también llamada *trilateración* en el caso de usar tres puntos de referencia – es una técnica para conocer la posición de un objetivo con argumentos geométricos.

Se basa en la búsqueda de la intersección de ciertas áreas a partir de la posición de varias balizas con posición conocida. Estas áreas están definidas por distancias o ángulos a las balizas de referencia, valor obtenible con distintas técnicas.

En general, la posición del objetivo vendrá dado, en  $n$  dimensiones, por

$$\mathbf{r} = \mathbf{f}(x_1, x_2, \dots, x_n) + \mathbf{n} \quad (1)$$

donde  $\mathbf{f}$  será la función característica de cada algoritmo y  $\mathbf{n}$  un vector que representa el ruido de la señal, con un valor medio nulo.

Este ruido está relacionado con la generación y el traslado de la señal desde la baliza hasta el objetivo y tiene un carácter aleatorio, por lo que su caracterización habitual es la de ruido blanco gaussiano, con valor medio cero.

Esta caracterización, unida al muestreo discreto de la señal por parte de los dispositivos electrónicos, resulta en un límite en la precisión del posicionamiento, incluso en los casos más ideales.

El límite más usual viene dado por la cota inferior de Cramer-Rao (CRLB), definido como la traza de la matriz de Fisher inversa. La matriz de Fisher se construye, en el caso de la parametrización del ruido con características gaussianas, como [2]

$$[\mathcal{I}(\boldsymbol{\theta})]_{ij} = \frac{1}{\sigma^2} \sum_{n=0}^{N-1} \frac{\partial s[n, \boldsymbol{\theta}]}{\partial \theta_i} \frac{\partial s[n, \boldsymbol{\theta}]}{\partial \theta_j} \quad (2)$$

donde  $\boldsymbol{\theta}$  indica el vector con los parámetros a evaluar,  $\sigma$  la desviación del ruido blanco gaussiano,  $N$  el número de balizas utilizadas y  $s[n, \boldsymbol{\theta}]$  la señal emitida por la baliza  $n$ , sin ningún ruido.

Así, la varianza de cada uno de los parámetros a estimar vendrá dada como

$$\sigma^2(\theta_i) \geq [\mathcal{I}(\boldsymbol{\theta})^{-1}]_{ii} \quad (3)$$

### 2.1.2.1 Tiempo de llegada (TOA)

El primer algoritmo considera una velocidad de transmisión conocida y una línea de visión directa entre las balizas y el objetivo. Teniendo en cuenta que el medio de propagación será el aire, la velocidad de la señal será la velocidad de la luz en el caso de ondas electromagnéticas. Esta velocidad no tendrá variaciones significativas al cambiar la zona de interés, por lo que será válida para prácticamente cualquier escenario.

Denotando al objetivo como OBJ y con tres balizas llamadas  $B_i$  con  $i = 1, 2, 3$ , la distancia recorrida por cada señal hasta el objetivo será <sup>1</sup>

$$\begin{aligned} d_i &= c(t_i) = c(\tau_i - t_0) \\ &= \sqrt{(x_i - x)^2 + (y_i - y)^2} + ct_0 \end{aligned} \quad (4)$$

siendo  $t_0$  el instante en el que la baliza  $i$  emitió el pulso. Así, la función característica de este algoritmo será, en dos dimensiones

$$f_i(x, y) = \sqrt{(x_i - x)^2 + (y_i - y)^2} \quad (5)$$

---

<sup>1</sup>Durante todo el desarrollo de la teoría de posicionamiento en esta sección se supondrá un sistema bidimensional. Esto no puede ser así en la implementación *real*, pero teniendo en cuenta que en el posicionamiento local se trata con sistemas euclídeos la adición de la tercera componente en todos los cálculos es trivial.

para cada baliza, definiendo una circunferencia de radio  $d_i$  como se muestra en la Figura 2.

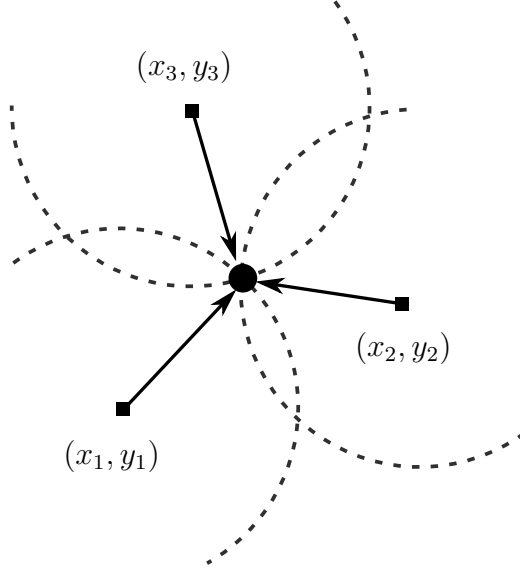


Figura 2: Posicionamiento por tiempo de vuelo (TOA).

A partir de las ecuaciones de (4) la posición de OBJ será [3]

$$\begin{aligned} x &= \frac{(y_2 - y_1)\gamma_1 + (y_2 - y_3)\gamma_2}{2[(x_2 - x_3)(y_2 - y_1) + (x_1 - x_2)(y_2 - y_3)]} \\ y &= \frac{(x_2 - x_1)\gamma_1 + (x_2 - x_3)\gamma_2}{2[(x_2 - x_1)(y_2 - y_3) + (x_2 - x_3)(y_1 - y_2)]} \end{aligned} \quad (6)$$

donde

$$\begin{aligned} \gamma_1 &= x_2^2 - x_3^2 + y_2^2 - y_3^2 + d_3^2 - d_2^2 \\ \gamma_2 &= x_1^2 - x_2^2 + y_1^2 - y_2^2 + d_2^2 - d_1^2 \end{aligned} \quad (7)$$

En este algoritmo es crítica una correcta sincronización entre los relojes de las balizas y el objetivo. Es posible minimizar este problema usando solo uno de los puntos para medir la distancia esperando una reemisión: emitiendo, desde una de las balizas o el objetivo, un pulso y esperando una respuesta similar del objetivo, con lo que el tiempo total será el doble del tomado por la señal original en llegar a la posición requerida.

El CRLB en el caso de TOA viene dado por [2]

$$\text{CRLB} = \sigma^2(\tau) \geq \frac{1}{8\pi^2 \text{SNR} \beta^2} \quad (8)$$

donde  $\tau$  es la distancia estimada y  $\beta$  el ancho de banda efectivo dado por

$$\beta^2 = \frac{\int_{-\infty}^{\infty} \omega^2 |S(\omega)|^2 d\omega}{\int_{-\infty}^{\infty} |S(\omega)|^2 d\omega} \quad (9)$$

con  $S(\omega)$  como la transformada de Fourier de la señal  $s(t)$ .

SNR es la razón entre la señal y el ruido –*signal-to-noise ratio*, en inglés–, de la que se tiene una relación inversamente proporcional. Es decir, la relación esperada: el SNR proporciona la relación entre la energía de la señal y la energía del ruido que la corrompe, así que un valor mayor indica menos ruido y por tanto una incertidumbre menor en la intersección de los pulsos de las balizas.

En este caso, el CRLB toma valores demasiado bajos cuando el SNR es muy bajo. Por ello en dichas situaciones es habitual el uso de la cota inferior de Ziv-Zakai (ZZLB) [4], que no sufre este problema, aunque no siempre es posible evaluar de forma analítica.

### 2.1.2.2 Diferencia de tiempo de llegada (TDOA)

Para eliminar la sincronización de relojes entre las balizas y el objetivo es posible usar también la diferencia temporal entre las señales de las balizas, teniendo así independencia en el sistema del objetivo. Aunque se consiga la independencia del reloj del objetivo, esta configuración sigue requiriendo que los relojes de las balizas estén sincronizados entre sí.

Así, la diferencia entre las distancias de las señales de dos balizas, teniendo en cuenta los tiempos en la llegada de la señal de dichas balizas será

$$d_{ij} = c(t_i - t_j) = c[(\tau_i + t_0) - (\tau_j + t_0)] = c(\tau_i - \tau_j) \quad (10)$$

válida para  $i = 1, 2, 3$  y  $j = 1, 2, 3$  con  $i \neq j$ . Así, la función característica de este algoritmo será

$$f_i(x, y) = \sqrt{(x_i - x)^2 + (y_i - y)^2} - \sqrt{(x_j - x)^2 + (y_j - y)^2} \quad (11)$$

para cada baliza  $i$  respecto de cualquiera  $j$  de las otras.

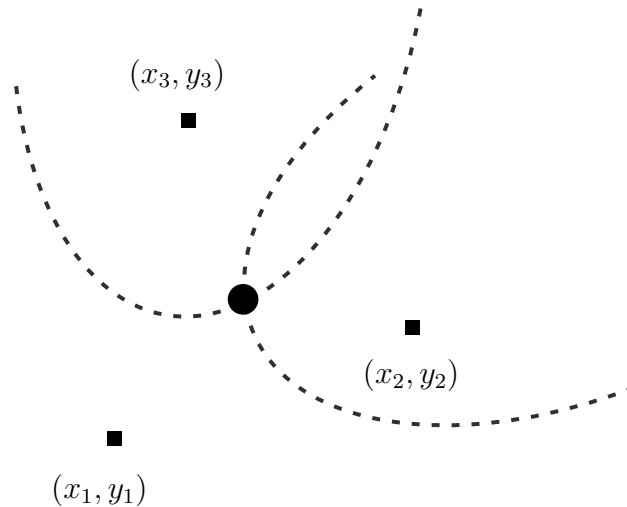


Figura 3: Posicionamiento por diferencia tiempo de vuelo (TDOA)

La función característica (11) hace que se definan parábolas en torno a cada una de las balizas, como se muestra de forma cualitativa en la Figura 3.

Así, tomando la baliza 1 como referencia tenemos que las coordenadas del objetivo vienen dadas por el sistema de ecuaciones

$$\begin{cases} d_{12} &= \sqrt{(x_1 - x)^2 + (y_1 - y)^2} - \sqrt{(x_2 - x)^2 + (y_2 - y)^2} \\ d_{13} &= \sqrt{(x_1 - x)^2 + (y_1 - y)^2} - \sqrt{(x_3 - x)^2 + (y_3 - y)^2} \end{cases} \quad (12)$$

con el que se obtiene de forma general dos soluciones para el posicionamiento: en la mayoría de casos una de ellas será un punto fuera de los límites del recinto en que se trabaja por lo que se elegirá el más cercano a las balizas colocadas.

En el caso de TDOA el CRLB tiene la misma expresión que en el TOA (8), teniendo en cuenta que el ruido añadido a las señales de las balizas tendrá la misma caracterización.

### 2.1.2.3 Intensidad de la señal recibida (RSSI)

Para evitar problemas con la sincronización de los relojes necesaria en las técnicas anteriores es posible aprovechar el decaimiento de la intensidad de la señal con la distancia según la relación

$$p(r) = p(r_0) - 10n \log\left(\frac{r}{r_0}\right) \quad (13)$$

donde  $p(r_0)$  y  $p(r)$  son la intensidad de la señal en el origen y el objetivo respectivamente y  $n$  el exponente de atenuación de la señal.

Esta técnica es muy susceptible ante el ruido ambiental y las posibles interferencias que puedan ocurrir entre las señales de distintas balizas o de fuentes externas. La disposición geométrica es similar a la vista en la técnica de tiempo de llegada (Figura 2), donde a partir de una medición de la señal de una baliza es posible definir una circunferencia de centro en dicha baliza y radio dado por (13).

A diferencia de los casos de TOA y TDOA el CRLB en esta técnica es independiente del ancho de banda de la señal, dado por [1]

$$\sigma^2(d_i) = \frac{\ln(10)\sigma_{sh}d}{10n} \quad (14)$$

donde  $\sigma_{sh}$  indica la desviación estándar del desvanecimiento de la señal.

La independencia de las cotas inferiores del ancho de banda de la señal hacen que esta técnica no aproveche la principal ventaja que da el UWB, por lo que su uso se restringe a otras tecnologías.

#### 2.1.2.4 Ángulo de llegada (AOA)

Otra opción para evitar la sincronización de los relojes entre las balizas consiste en usar el ángulo de incidencia de la señal en el objetivo como se muestra en la Figura 4, de tal forma que, para una baliza  $B_i$  dicho el objetivo se encuentra en la posición

$$\begin{aligned} x &= d_i \cos(\phi_i) + x_i \\ y &= d_i \sin(\phi_i) + y_i \end{aligned} \quad (15)$$

por lo que la función característica de este algoritmo será la distancia de cada baliza al objetivo, a partir de (15)

$$f_i = \arctan(\phi_i) = \arctan\left(\frac{y - y_i}{x - x_i}\right) \quad (16)$$

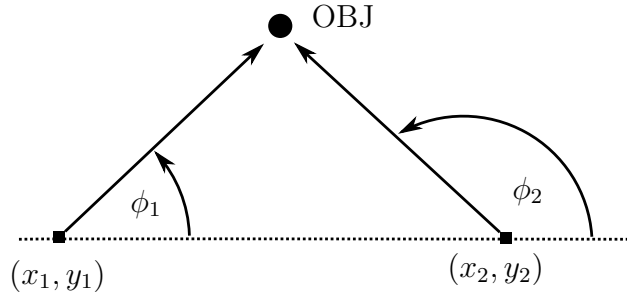


Figura 4: Posicionamiento por ángulo de llegada (AOA).

Esta técnica requiere el uso de antenas direccionales, lo que hace que aumente el coste de su implantación.

En este caso solo necesitamos el ángulo respecto a las balizas, por lo que su implantación será más adecuada en situaciones en las que el cálculo de la distancia entre objetivo y balizas no es fiable o como apoyo o corrección junto con TOA o TDOA.

Para el AOA el CRLB, que nos dará el mínimo de varianza para el ángulo de incidencia vendrá dado, en el caso de balizas colocadas a lo largo de un eje de la forma mostrada en la Figura 4 por [2, 5]

$$\sigma^2(\phi) = \frac{c^2}{2\pi^2 \text{SNR} \beta^2 N (N^2 - 1) d \cos(\phi)} \quad (17)$$

donde  $N$  indica el número de balizas utilizadas y  $d$  la distancia entre ellas.

#### 2.1.3. Mitigación en casos no ideales

En el planteamiento de los algoritmos geométricos se suponía una línea de visión directa entre las balizas y el objetivo para determinar los parámetros de interés, pero en condiciones reales esta suposición no siempre podrá ser respetada.

La presencia de obstáculos que impidan un trayecto de la señal parametrizable o generen rebotes en la zona de interés pueden provocar un posicionamiento erróneo o, en el peor de los casos, imposible.

Para mitigar la modificación en la señal en estos casos se recurre a técnicas estadísticas que se dividen en dos grupos: técnicas paramétricas, en las que se realiza un ajuste del error a una distribución de probabilidad conocida, y técnicas no paramétricas, en la que no se asume ninguna distribución a priori.

## **2.2. Tecnologías**

Todos los algoritmos geométricos mencionados necesitan un soporte físico para determinar las variables posicionales del objetivo que requieren, para lo que es posible el uso de varias tecnologías, como se recoge en [6], de donde procede toda la información de esta sección.

### **2.2.1. Sonido**

Una primera propuesta puede ser el uso de ondas de sonido, normalmente fuera del rango de audición humano. Dependiendo de la frecuencia utilizada son referidos a veces como ultrasonidos.

La baja velocidad de propagación en el aire hace que la incertidumbre en la medida de los tiempos de llegada sea muy baja, por lo que su uso se asocia casi exclusivamente a TOA y TDOA.

Esta tecnología tiene su principal punto débil en la dependencia de la propagación del sonido con la temperatura y la humedad del aire: la precisión ganada por la baja velocidad de propagación se ve compensada con un rango máximo muy bajo, lo que hace que su implantación en entornos amplios no sea óptimo. Además, el equipamiento necesario tiene un coste mayor al de sus alternativas, por lo que su uso se restringe actualmente a situaciones concretas.

### **2.2.2. Luz**

La popularización de la iluminación LED, capaz de cambiar la intensidad de emisión en tiempos muy cortos ha provocado el desarrollo de tecnologías que modulen información a tasas de tiempo mucho más altas de la que el ojo humano puede captar.

Su principal ventaja es la posibilidad de aprovechar las infraestructuras de iluminación ya existentes en ambientes interiores, aunque esta tecnología es muy dependiente de las condiciones en dichas situaciones.

La desventaja más directa es su total reflexión en paredes u objetos opacos, lo que hace que la atenuación sea importante en interiores donde la disposición desfavorezca



una iluminación uniforme o simplemente un tráfico intenso de personas. Además, es muy sensible a fuentes externas como luz solar.

### **2.2.3. Visión por computador**

El desarrollo en los últimos años de técnicas para el reconocimiento de imágenes ha hecho posible la obtención de técnicas de posicionamiento con cámaras.

Esta técnica ha tenido especial auge en el campo de vehículos autónomos, aunque su uso se complementa de forma habitual con otras tecnologías al necesitar buenas condiciones lumínicas para su correcto funcionamiento.

Al igual que en el caso de la luz, su principal ventaja reside en el aprovechamiento de la infraestructura ya existente. En este caso es posible utilizar una tecnología ya asentada y madura sin realizar ninguna modificación sin más que usar cámaras con la resolución suficiente para los distintos casos de uso.

### **2.2.4. Campo magnético**

Aunque hay algunos trabajos que desarrollan el sistema de posicionamiento sobre un campo magnético artificial, la mayoría de aplicaciones se basan en el campo magnético terrestre para determinar la posición del objetivo.

Estas técnicas requieren un estudio previo para determinar la intensidad de dicho campo magnético terrestre a lo largo del espacio en los que implementar el posicionamiento, aunque tras ello el equipamiento necesario es mínimo al requerir únicamente un magnetómetro.

Este tipo de técnicas tiene una precisión muy baja, en torno a metros, y es muy susceptible a cualquier modificación del campo magnético sobre el que se realizó el estudio previo por lo que su aplicación se reduce a situaciones muy concretas.

### **2.2.5. Wi-Fi**

En el espectro electromagnético las señales de Wi-Fi son también una alternativa a la hora de determinar distancias por el decaimiento de su intensidad con la distancia. Su atractivo principal es, como en otras tecnologías, el aprovechamiento de las infraestructuras y el desarrollo ya existente.

Aunque también es posible su uso con TOA y AOA, su uso principal se restringe a técnicas de *fingerprinting*, ya que habitualmente los puntos de acceso Wi-Fi usados para la conexión a internet no suelen superar la unidad por planta o edificio.

Esta escasez de balizas es la principal desventaja: en su uso original un solo emisor puede dar servicio a multitud de dispositivos en una planta o incluso un edificio, por lo que lo habitual será encontrar numerosos obstáculos entre emisor y objetivo. Colocar

más balizas puede añadir interferencias entre las señales teniendo en cuenta que los canales disponibles tienen un ancho de banda bajo.

Además, el uso extendido de esta tecnología inalámbrica hace que su espectro esté ya, en algunas ocasiones, saturado. Su precisión en ambientes ideales está en torno al metro, así que podría ser una buena opción en entornos abiertos o con pocas redes que puedan causar interferencias.

### 2.2.6. Bluetooth

En el caso del Bluetooth nos encontramos de nuevo con una tecnología con una gran infraestructura ya existente teniendo en cuenta que su uso en teléfonos móviles se da desde hace más de una década.

Entre las distintas generaciones se encuentra un protocolo llamado Bluetooth Low Energy (BLE) centrado en obtener un funcionamiento con el mínimo gasto energético posible, lo que lo hace idóneo para usar como sistema para localización como el que se trata en este trabajo.

Técnicamente comparte el mismo rango de frecuencias con el Wi-Fi y las técnicas de posicionamiento son en general parecidas, aprovechando el decaimiento de la señal con técnicas de *fingerprinting*. Pero a diferencia del Wi-Fi, la baja energía de los emisores BLE permiten la colocación de un número mayor de balizas, lo que abre la puerta al uso de algoritmos de multilateración.

Los mejores resultados sobre esta tecnología arrojan una precisión entre 1 y 3 metros, aunque estos resultados dependen en gran medida del número de balizas empleadas.

### 2.2.7. Otras tecnologías

Además de los soportes físicos ya mencionados, existe un enorme abanico de otras tecnologías. La más ampliamente usada y no mencionada se encuentra en el espectro electromagnético dedicado a las redes móviles, aunque las grandes distancias a las que se encuentran las balizas –torres de comunicación en este caso– hace que la precisión, en las mejores ocasiones, se encuentre en torno a los 50 metros, inviable para una localización en interiores.

Es posible que el desarrollo de nuevas generaciones de este tipo de redes como el 5G permitan tener una mayor precisión para permitir su uso.

Entre otras alternativas se encuentran también de comunicaciones a muy corta distancia como el *Near-Field Communication* (NFC), aunque este sistema requiere una participación activa del objetivo –debe acercarse mucho a alguna de las balizas– y no permite localización de objetos en movimiento.

## 2.3. Ultra-Wide Band

El Ultra-Wide Band (UWB) se define como una emisión de radiofrecuencia con un ancho de banda de al menos 500 MHz o más del 20% de la frecuencia central. Su definición técnica se recoge en el estándar 802.15.4 del IEEE junto con otras redes inalámbricas de área personal.

Los canales usados y las potencias permitidas tienen distintas legislaciones según el país que se trate. En el caso de Europa, la regulación del Instituto Europeo de Normas de Telecomunicaciones (ETSI) permite emisiones en los rangos de frecuencia 3,1 – 4,8GHz, destinado a localización de servicios de emergencia y 6,0 – 9,0GHz para localización general [7].

### 2.3.1. Ventajas

La principal característica del UWB, su gran ancho de banda, le da varias ventajas en su uso principal y que se discute en este trabajo: servicios de localización a partir de técnicas geométricas. Aunque sea el principal uso no es el único, ya que también permite transmisión de datos de una forma rápida y puede ser una alternativa en este tipo de situaciones.

El espectro tan amplio de frecuencia provoca que el decaimiento de la señal, dependiente de la frecuencia, sea fácilmente identificable al estar formado de componentes que sufren este hecho de forma acusada.

En el caso de que se produzca propagación multicamino, en la que los rebotes creen nuevos frentes de onda, la resolución del camino principal se vuelve sencilla teniendo en cuenta que el espacio adicional que recorran las trayectoria sin visión directa –fruto de la reflexión en paredes u obstáculos– hacen que estos caminos secundarios se verá rápidamente desfavorecidos.

Además, el amplio ancho de banda hace que las interferencias entre los propios pulsos fruto de los distintos caminos sean mínimas, nulas si el tiempo de propagación adicional supera el periodo entre la emisión de dos pulsos.

Este último hecho es muy probable en el caso del UWB. El tiempo de transmisión de un pulso viene dado por[1]

$$T_{tx} = \left( N_d + \frac{N_d}{N_{mP} N_0 N_d} \right) T_b T_{pr} \quad (18)$$

donde  $N_d$ ,  $N_{mP}$  y  $N_0$  indican los tamaños de datos, tasa máxima y *overhead* respectivamente; mientras que  $T_b$  y  $T_{pr}$  indican la duración y el tiempo de propagación de cada bit, respectivamente.

Si el tiempo de propagación es inversamente proporcional a la tasa de datos, muy amplia en el caso del UWB, dicho tiempo de propagación será por tanto muy bajo, lo

que no solo evita las interferencias sino que permite una resolución temporal muy alta.

La tasa de datos viene dada por capacidad de un canal, dada en bits/sec como[1]

$$C = B \log_2(1 + \text{SNR}) \quad (19)$$

siendo  $B$  el ancho de banda de dicho canal y considerando ruido blanco gaussiano en el transporte de la señal.

Esta gran capacidad es la que convierte al UWB una de las mejores alternativas para el transporte de datos, aunque su adopción en este ámbito no es hasta el momento demasiado amplia.

Otra de las características del UWB es su eficiencia energética. Aunque es posible entender este término de distintas formas, a la hora de realizar comparaciones se suele recurrir a la capacidad por unidad de potencia o al tiempo de uso permitido con sensores alimentados con baterías.

La gran capacidad de transmisión de los canales de UWB hace posible jugar con el periodo de emisión de los pulsos, lo que permite un consumo menor. Por ello, la densidad de energía espectral es muy baja con tasas de transmisión altas; así, la capacidad de detección de estas emisiones es complicada, lo que abre la puerta a aplicaciones militares.

Esta baja densidad de energía espectral también hace que las interferencias con canales con ancho de banda menor tengan un efecto mínimo. Un gran ancho de banda permite el uso de varias bandas del espectro electromagnético a la vez, por lo que el uso en ambientes con señales de Wi-Fi o Bluetooth –tecnologías con un ancho de banda estrecho– es posible. Aun así, es habitual el uso de técnicas de detección y evasión (DAA), en las que antes de la puesta en marcha de la transmisión se evalúan los canales en uso para evitar en la medida de lo posible las bandas ya ocupadas.

### 2.3.2. Pulsos de UWB

En el diseño de sistemas de UWB la elección de la forma del pulso es crucial, ya que determina la respuesta en frecuencias como se muestra en la Figura 5. Los pulsos mayoritariamente tienen perfiles gaussianos, aunque con distintas características [8].

La primera opción es el uso de una forma gaussiana clásica

$$y(t) = A \exp(-a^2 t^2) \quad (20)$$

donde  $A$  indica la máxima amplitud del pulso y  $a$  la constante que indica la pendiente del pulso por lo general elegida de modo que el pulso tenga una duración en el orden de centenas de picosegundos o nanosegundos.

Su respuesta espectral, dada por la transformada de Fourier será

$$Y(\omega) = \frac{A}{a\sqrt{2}} \exp\left(-\frac{\omega^2}{4a}\right) \quad (21)$$

con pico en  $f_0 = 0$ , a partir del cual es posible determinar el ancho de banda teniendo en cuenta que debe ser 3dB menor que dicho pico como

$$\Delta f = 0,8326 \frac{a\sqrt{2}}{2\pi} \quad (22)$$

La segunda opción, llamada gaussiana monocíclica, consiste en la primera derivada temporal de (20)

$$y(t) = -2a^2 A t \exp(-a^2 t^2) \quad (23)$$

con la respuesta espectral

$$Y(\omega) = \frac{i\omega A}{a\sqrt{2}} \exp\left(-\frac{\omega^2}{4a}\right) \quad (24)$$

con frecuencia máxima

$$f_0 = \frac{a\sqrt{2}}{2\pi} \quad (25)$$

y ancho de banda

$$\Delta f = 1,155 \frac{a\sqrt{2}}{2\pi} \quad (26)$$

Este pulso suele ser el utilizado para el diseño de los sistemas de UWB. A diferencia del pulso gaussiano simple, este pulso tiene un nodo con amplitud cero, lo que elimina las posibles componentes continuas y añaden una referencia en la señal, además de una menor energía en señales de baja frecuencia, transmitidas de forma muy ineficiente por las antenas de los emisores de UWB.

Es también idónea teniendo en cuenta las técnicas de modulación de la información usadas, para las que se necesita una referencia en la amplitud del pulso o una forma característica que sea posible invertir.

Por último es posible tomar la segunda derivada de (20) para llegar al pulso gaussiano de doblete, caracterizado como

$$y(t) = -2a^2 A \exp(-a^2 t^2)(1 - 2a^2 t^2) \quad (27)$$

con respuesta espectral

$$Y(\omega) = \frac{-\omega^2 A}{a\sqrt{2}} \exp\left(-\frac{\omega^2}{4a}\right) \quad (28)$$

con frecuencia máxima

$$f_0 = \frac{a}{\pi} \quad (29)$$

y ancho de banda

$$\Delta f = 1,155 \frac{a\sqrt{2}}{2\pi} \quad (30)$$

igual que en el caso monocíclico aunque con frecuencia central distinta.

Este pulso tiene características muy parecidas al monocíclico: un ancho de banda similar y la ausencia de posibles componentes de corriente continua al contar ahora con dos puntos donde la amplitud es nula.

La principal diferencia radica en la frecuencia en la que se da la respuesta máxima, ahora con un valor más alto. Sucesivas derivaciones de (20) tendrán el mismo resultado, con igual ancho de banda y valores crecientes de  $f_0$ , por lo que este tipo de pulsos de mayor orden es quizá adecuado para canales cercanos al límite superior del espectro [9].

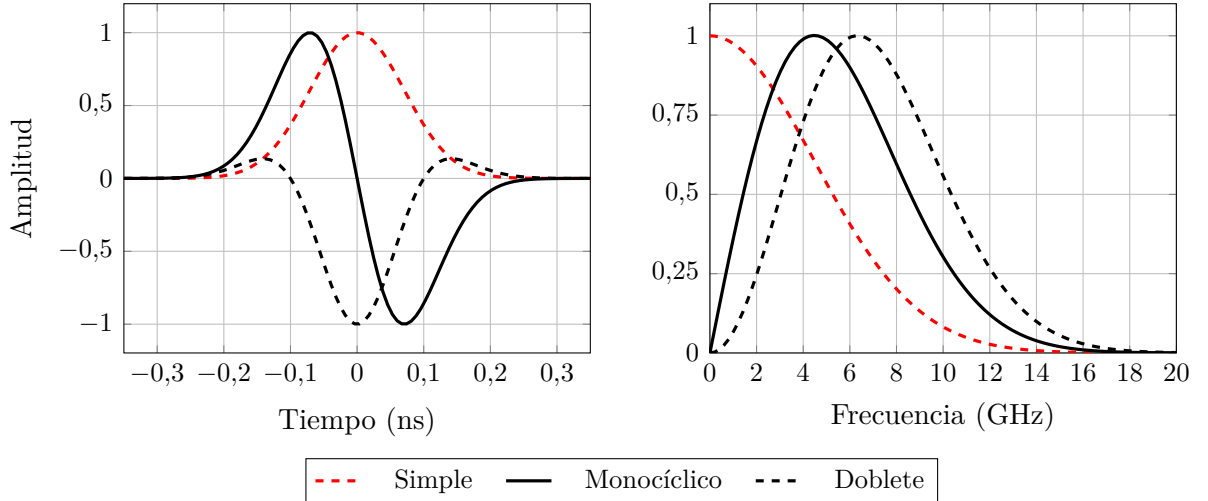


Figura 5: Respuesta en los dominios de tiempo y de frecuencia de los distintos tipos de pulsos de UWB de 500ps, normalizados en ambos casos

La señal generada por las balizas será una sucesión de los pulsos anteriores, de tal manera que puede modelizarse como

$$s(t) = \sum_{k=-\infty}^{\infty} a_m(k)p(t - kT_f) \quad (31)$$

con  $T_f$  como el periodo entre pulsos y  $p(t)$  la forma del pulso elegida, con los coeficientes  $a_m(k)$  dependientes de la modulación de la señal.

Con la adición de ruido blanco gaussiano y teniendo en cuenta las técnicas geométricas de posicionamiento la ecuación (31) pasa a ser la ecuación (1).

## 2.4. Robot

Con el fin de automatizar la toma de medidas para hacerla lo más eficiente posible se usó un robot móvil capaz de desplazarse mediante navegación autónoma en un entorno conocido.

El elegido en este caso fue el robot TurtleBot 2, un robot con fines educativos y de investigación capaz de desplazarse y orientarse con total libertad en superficies llanas, como eran los en el que se desarrolló este trabajo. En la Figura 6 se puede ver el robot durante una de las tomas de medidas.



Figura 6: Turtlebot en una de las tomas de medidas.

El TurtleBot funciona con ROS –*Robot Operating System*, en inglés–, un entorno de trabajo enfocado a los sistemas robóticos encargado del soporte para todos los dispositivos hardware del robot como motores, encoders o cámaras. Cuenta con diversas librerías para distintos casos de uso, entre las que se encuentra la navegación en un entorno controlado como la que se da en el caso de este trabajo.

Dentro de los paquetes disponibles, el usado en el desarrollo del trabajo es el encargado de la navegación del robot, que permite desplazar al robot a cualquier punto del entorno de trabajo.

Es posible acoplar al robot un sistema de visión llamado Kinect que, mediante luz, permite obtener un mapa de profundidad del entorno que lo rodea. Fue desarrollado en primera instancia para el uso en videojuegos pero su uso también se ha extendido a labores de investigación al permitir el posicionamiento de objetos y paredes, de tal

manera que permite al robot evitar obstáculos dinámicos en el caso de que interpongan en su camino.

La funcionalidad de navegación aprovecha estos sensores realizando una fusión de sus resultados con los datos de movimiento de los motores que realizan el movimiento de tal forma que es posible corregir cualquier error en casos donde el movimiento de las ruedas no se traslade directamente en movimiento del robot, como puede ocurrir al rotar sobre sí mismo.

Para conseguir el posicionamiento en el mapa el paquete de navegación construye un *planner* global sobre el que es posible determinar las rutas a seguir por el robot para llegar a un punto dado sorteando obstáculos y paredes. Junto a él trabaja un *planner* local, en el que gracias a los sensores incorporados se evalúan de forma continua los alrededores del robot. Así, es posible conseguir de forma continua una evaluación de los posibles obstáculos que no se encuentran en el mapa del planner global, para lo que se construye un mapa de costes con el fin de abortar el movimiento en el caso de que sea imposible alcanzar el punto objetivo.

Con la información actualizada del *planner* local, el *planner* global es capaz de proporcionar la posición del robot en el mapa. Es posible observar un esquema del funcionamiento de estos sistemas en la Figura 7.

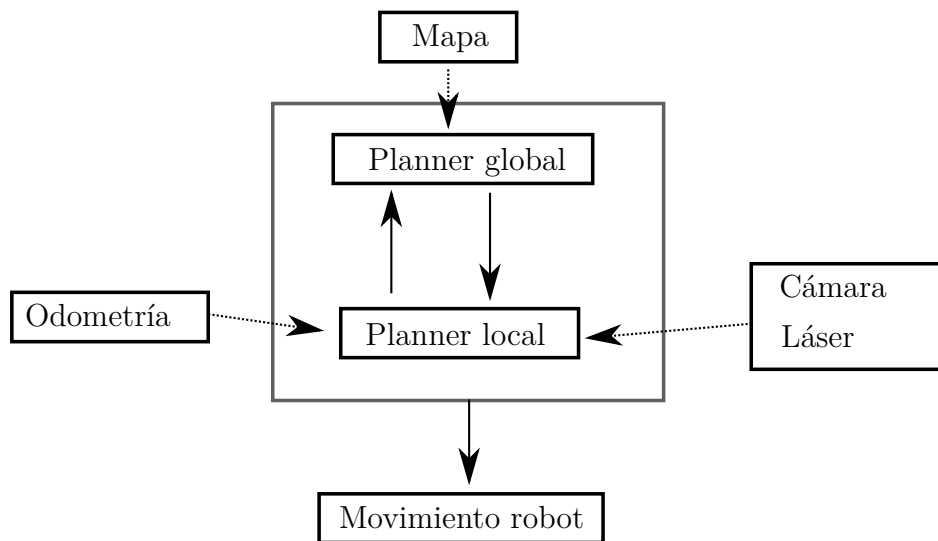


Figura 7: Esquema del funcionamiento del paquete de navegación de ROS.

Con esta herramienta la librería de navegación permite un mapeado autónomo tomando como referencia los datos de odometría que proporcionan los motores propulsores del robot para determinar las dimensiones del entorno.

En el caso de disponer de antemano de dichas dimensiones es posible proporcionar un mapa al sistema de navegación y evitar el paso de reconocimiento del entorno. Esto no solo ahorra tiempo, sino que además minimiza las posibles discrepancias entre los datos de odometría y los desplazamientos reales del robot al realizar el mapeado de



forma autónoma.

La opción de realizar un mapa previo fue la elegida en este caso, ya que el robot cuenta con rutinas para el reposicionamiento en el mapa en dicho caso. Así, a partir de los límites establecidos y comprobados de forma manual, las posibles discrepancias en la odometría del robot se ven continuamente compensadas y corregidas.

Para facilitar el uso de ROS existe la posibilidad de usar el simulador Stage, capaz de crear un mundo virtual a partir de un mapa en dos dimensiones en el que colocar el Turtlebot y simular su funcionamiento de forma total sin tener acceso al robot de forma física. Es posible observar su interfaz en la Figura 8.

Además, ROS también permite el uso de una herramienta de visualización de la posición del robot en el mapa y de todos los sensores que incorpora llamada *rviz*. Aunque es compatible con Stage, sus funcionalidades brillan al usar el robot en entornos reales, donde es posible comprobar de forma continua que su posicionamiento es correcto y que sus sensores funcionan como es debido.

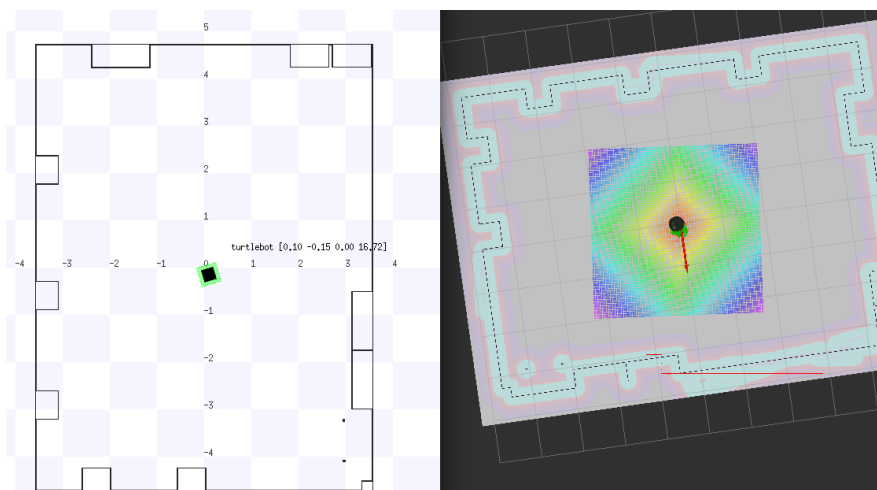


Figura 8: Captura del simulador Stage (a la izquierda) y la herramienta de visualización *rviz* (a la derecha).

## 2.5. Sensores UWB

Aunque el uso de UWB para el posicionamiento no está, al escribir este trabajo, utilizado de forma masiva, existen varias implementaciones comerciales que hacen uso de las técnicas geométricas mencionadas en secciones anteriores. A pesar de que el uso de sensores propios es habitual, la adaptación de teléfonos móviles para la estimación de la posición también ha sido explorada por empresas como BeSpoon, a la espera de una implantación mayoritaria de hardware compatible en el diseño de estos dispositivos.

Los sensores utilizados han sido los pertenecientes al kit MDEK1001 ofertado por la empresa DecaWave. Se trata de un sistema que ofrece, con una configuración mínima, la posibilidad de usar cualquiera de los componentes como baliza u objetivo. Además,

implementa un algoritmo de posicionamiento con al menos tres balizas de, según el fabricante, hasta 10cm de precisión con visión directa [10].

Este kit está formado por los módulos de desarrollo DW1001, encapsulados en una carcasa de plástico que permite colocarlos encima de una superficie plana o su colocación en paredes, como se puede observar en la Figura 9. Permiten su uso con baterías, como fue el uso en este trabajo para las balizas, o a partir de alimentación a través del puerto USB utilizado para la recolección de datos.



Figura 9: Módulo de sensor UWB DWM1001 dentro de la carcasa que lo contiene.

Algunas alternativas comerciales cuentan con antenas direccionales que permiten usar técnicas de AOA, pero no es el caso del kit utilizado que solo permite el posicionamiento con TOA y TDOA.

Los sensores, además de la comunicación por UWB, también permiten la comunicación por Bluetooth para su configuración como objetivos o balizas y en este último caso, la posición que ocupan. También permite configurar estas posiciones de las balizas de forma automática, aunque requiere que todas se coloquen en el mismo plano y en general una configuración manual proporcionará unos mejores resultados.

Para la comunicación por UWB los sensores utilizan el canal 5, con frecuencia central de 6,5GHz y ancho de banda de 499,2MHz que le permiten una tasa de datos de 6,81Mbps [10].

El inicio de comercialización de dicho kit se remonta a 2017, por lo que no existe una amplia bibliografía sobre su uso en entornos reales más allá de los datos proporcionados por el fabricante.

En trabajos anteriores es posible encontrar mediciones tanto en entornos sin obstáculos con distancias bajas, en las que la precisión se encuentra en torno a los 18cm [11]; como en escenarios con distancias amplias y ciertas situaciones sin línea de visión

directa, donde la discrepancias crecen hasta unos 50cm [12, 13].

DecaWave ofrece para su configuración y uso una aplicación para teléfonos móviles Android en la que representa en un mapa la posición de los sensores marcados como objetivos, pero también es posible una conexión a un ordenador como puerto serie.

Esta última opción permite obtener cuáles de las cuatro balizas –en el caso de que se configuren un número mayor de ellas– se están usando en cada momento para realizar la estimación de la posición junto con la distancia calculada entre el objetivo y cada una de dichas balizas.

Además de la posición estimada, el algoritmo de posicionamiento proporciona un número entre 0 y 100 llamado «factor de calidad». Debido a la naturaleza propietaria de este algoritmo, DecaWave no publica detalles sobre la obtención de este dato más allá de ser una comparación entre el posicionamiento obtenido con tres de las balizas y el obtenido al introducir la cuarta.

### 3. Metodología

Para la toma de datos se consideraron dos escenarios: el primero el Laboratorio de Robótica 0L3 del Instituto de Computación Científica Avanzada de la Universidad de Extremadura (ICCAEx), situado en los Institutos Universitarios de Investigación de la Universidad de Extremadura en Badajoz, en el que se disponía de una superficie amplia y libre de objetos para tomar medidas en un escenario lo más cercano a lo ideal.

El segundo escenario fue la segunda planta del edificio B de la Facultad de Física de la Universidad de Extremadura, elegido para acercarse a una disposición más general, con obstáculos entre las balizas y el objetivo.

#### 3.1. Simulación previa

Antes de proceder a la toma de medidas en las dos localizaciones escogidas se escribió el código responsable de los nodos de ROS que desplazaban al robot y recogían los datos de posicionamiento con la ayuda del simulador Stage. Todo este código y su estructura figura en el Apéndice 2.

Además del código del nodo, también fue necesario diseñar el mapa sobre el que el paquete de navegación de ROS construye el *planner* global para conocer la posición y controlar el movimiento del robot. En el caso del laboratorio, dicho mapa estaba ya construido al ser localización habitual de investigación con el Turtlebot.

No fue así en para el edificio de Física. En una primera instancia se recurrió a los planos de construcción, a la espera de posibles correcciones con la obtención de las dimensiones reales de la planta con un medidor láser. Finalmente estas correcciones fueron necesarias, ya que dichos planos no respetaban completamente las medidas reales de los pasillos de la planta que debía recorrer el robot.

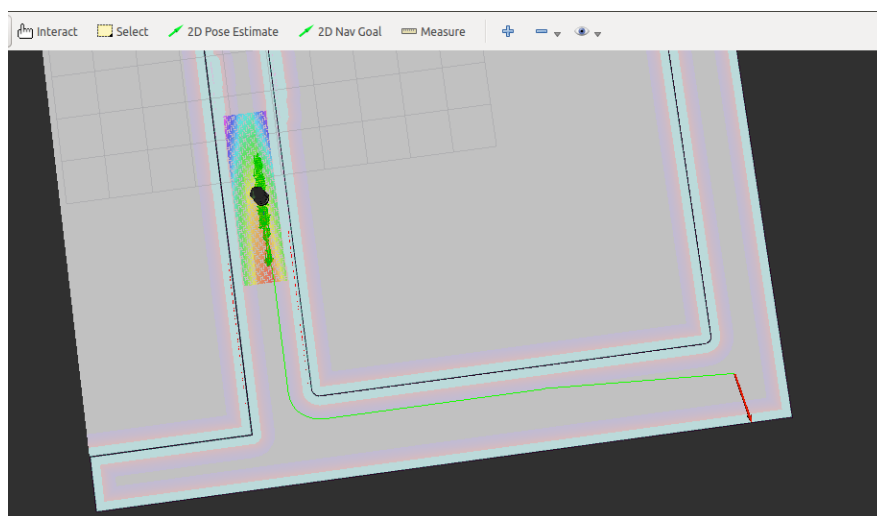


Figura 10: Captura de *rviz* con el primer mapa del edificio de Física, sin columnas.

Además, por simplicidad la primera aproximación del mapa no incluía las columnas de la planta, como se puede observar en la Figura 10. Aunque la superficie que ocupan en el plano es pequeña, una primera toma de medidas con el robot hizo evidente que su reconocimiento por parte de los sensores que incorpora llevaba a las rutinas de posicionamiento local a equívocos a la hora de relacionar estas columnas con las paredes del mapa proporcionado. Esto hacía que los valores obtenidos por el posicionamiento local del robot tuviesen discrepancias de hasta 50cm respecto a la posición real, medida de forma manual.

Por ello se procedió a crear un nuevo mapa con dichas columnas. En este caso el posicionamiento del robot mejoró de forma significativa, siendo capaz de identificar las columnas con solvencia y reduciendo las discrepancias observadas a unos 10cm, mismos valores que se obtenían en el laboratorio.

### 3.2. Toma de medidas

En todos los casos la toma de medidas siguió el mismo procedimiento: en base a una trayectoria fijada que atraviesa una serie de puntos en cada escenario se toman un número prefijado de datos a partir de los sensores, con el robot parado en cada uno de dichos puntos.

Para determinar el número de valores a recoger en cada caso se realizaron tomas de medidas previas para determinar la naturaleza de los datos obtenidos. En todos los casos se obtuvo una distribución de tipo gaussiano, como se puede ver en la figura 11, por lo que se tomaron 50 datos en el caso del laboratorio y 250 en el caso del edificio de Física con el objetivo tener una dispersión al menos menor de 2cm tomando el mínimo número de datos.

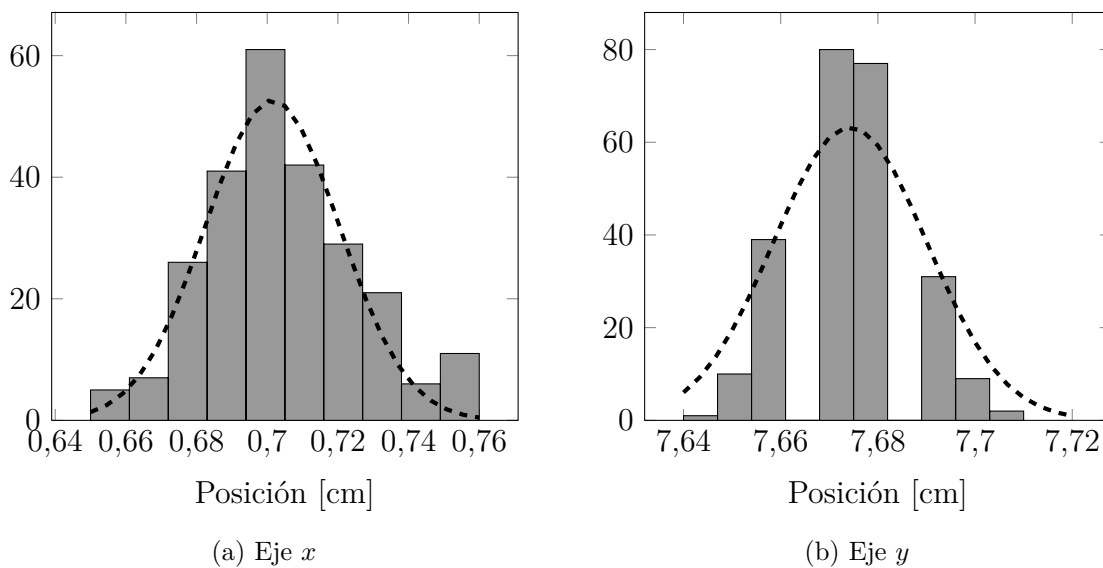


Figura 11: Histograma de los datos de posicionamiento de los sensores en uno de los puntos del edificio de Física, con un ajuste a una curva gaussiana en cada caso.

Además, para eliminar cualquier tipo de anomalía estadística cada trayectoria se repitió un mínimo de tres veces, tras las que se evaluaron los datos. En los casos en los que la dispersión fue excesiva, se procedió a repetir de nuevo estas mediciones.

En las primeras tomas, donde se orientaba el robot hacia el siguiente punto de tal forma que su desplazamiento tomase el menor tiempo posible, se observó que el diseño de las antenas de los sensores utilizados, con una ganancia ligeramente dependiente del ángulo de incidencia [14], hacía que, al girar el sensor asignado como objetivo, la estimación de su posición variase.

Esta observación también ha sido recogida en trabajos previos [15], donde se intentaba corregir a partir de medidas con distintos ángulos de incidencia. Esto no es posible en el caso de este trabajo, ya que no versa sobre el desarrollo de un algoritmo de posicionamiento, sino sobre el estudio la precisión del ya implementado. El algoritmo utilizado es propiedad de la empresa fabricante de los sensores, que no ha liberado detalles sobre su implementación, lo que hace imposible cualquier corrección.

Así, para minimizar este obstáculo todas las medidas que figuran en los resultados están tomadas colocando al robot con el mismo ángulo respecto a las balizas, de tal manera que no es necesario considerar variaciones en la posición obtenida debido a su orientación.

### 3.3. Laboratorio de Robótica del ICCAEx

La primera toma de medidas tuvo lugar en el laboratorio 0L3 del ICCAEx, donde se disponía de una superficie de aproximadamente 5 metros por 7 metros completamente libre para usar por el robot como se puede ver en la Figura 12.

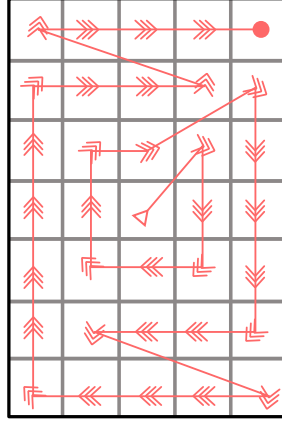


Figura 12: Fotografías del laboratorio de Robótica del ICCAEx.

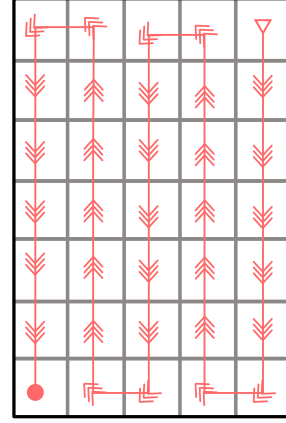
Debido a la libertad a la hora de recorrer dicha superficie se eligieron tres trayectorias que cubriesen todos los puntos a evaluar, establecidos en una cuadrícula con una separación de 1 metro entre cada uno de ellos de tal forma que se evitase un posible sesgo en el movimiento del robot.

31	32	33	34	35
26	27	28	29	30
21	22	23	24	25
16	17	18	19	20
11	12	13	14	15
6	7	8	9	10
1	2	3	4	5

(a) Puntos a evaluar



(b) Trayectoria en espiral



(c) Trayectoria en vertical

Figura 13: Disposición del laboratorio.

La primera de estas trayectorias tenía un recorrido con origen en el centro del laboratorio, acercándose a las paredes exteriores en forma de espiral, con algunas modificaciones para asegurar el paso por todos los puntos.

El segundo trayecto hacía que el robot avanzase de forma recta siguiendo líneas verticales –en el sentido tal y como se ha planteado la representación en dos dimensiones en la figura 13c–, con separación de un metro entre cada punto.

Por último se diseñó una trayectoria en la que robot pasase de forma aleatoria por los 35 puntos a medir, de tal forma que en cada intento el orden fuese distinto.

Para comprobar en este escenario ideal el efecto de la colocación de las balizas se repitió la toma de medida en dos configuraciones, colocando 4 y 6 sensores respectivamente. Teniendo en cuenta que el sistema empleado solo utiliza cuatro de las balizas para el posicionamiento del objetivo, el uso de la configuraciones con 6 de los sensores solo afecta a la posición de dichas baliza en los cálculos, usando siempre los cuatro sensores más cercanos.

La altura a la que fueron colocadas los sensores que actúan como balizas se vio restringida por el mobiliario del laboratorio, de modo que la mayoría se encontraban a unos 180cm del suelo con una de ellas a 80cm y otra a 200cm, aunque el sistema de los sensores permite estas diferencias de alturas en el modo de configuración manual de las balizas.

En la figura 14 es posible comprobar la posición respecto a los puntos de medida de las balizas en ambas configuraciones.

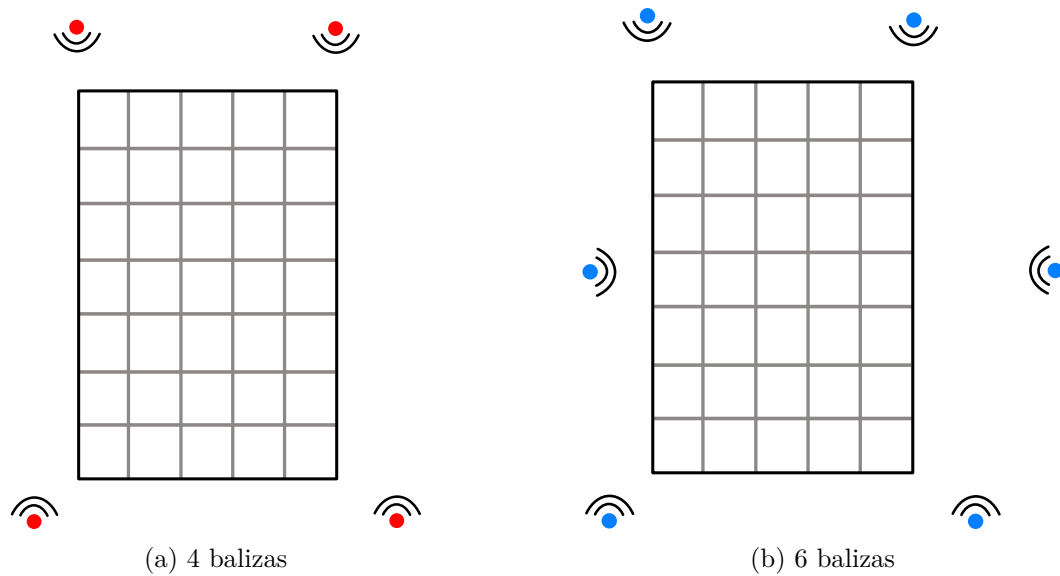


Figura 14: Configuraciones de las balizas en el laboratorio.

### 3.4. Edificio de la Facultad de Física

El segundo escenario se encontraba en la primera planta del edificio B de la Facultad de Física, en la que por razones estructurales se encuentran una serie de columnas como se puede observar en la Figura 15.

En este escenario aparecen elementos más cercanos a una implementación de un sistema de posicionamiento local habitual: obstáculos en las balizas, que no es posible colocar en cualquier lugar, y un tránsito de personas habitual que podía interferir en la señal del sistema utilizado.



Figura 15: Fotografías del edificio B de la Facultad de Física.

En este caso todas las medidas fueron tomadas utilizando la misma trayectoria, que se puede observar en la Figura 16, que seguía los pasillos de la planta del edificio al no poder optar a más variaciones.



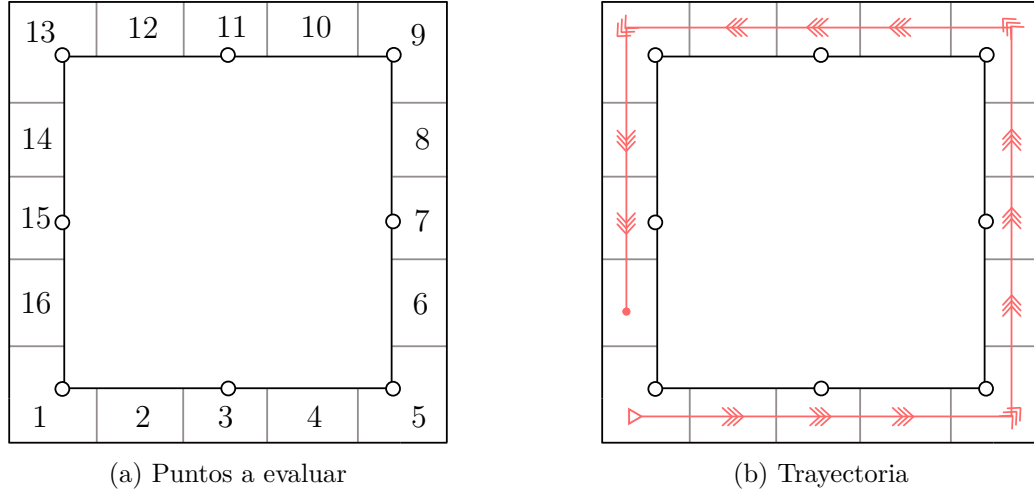


Figura 16: Disposición de la planta del edificio de Física.

Al igual que en el caso del laboratorio, se repitieron las medidas con distintas configuraciones de las balizas, con tres alternativas en este caso: 4, 6 y 8 balizas repartidas de la forma más uniforme posible tal y como se indica en la figura 17.

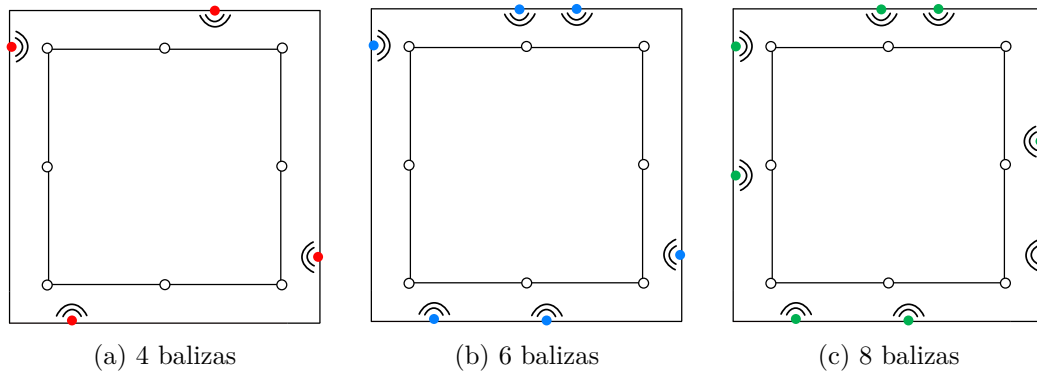


Figura 17: Configuraciones de las balizas en el edificio de Física.

Como en el caso del laboratorio, el kit de sensores UWB utilizado solo usa 4 de las balizas disponibles para estimar la posición del objetivo, por lo que el número mayor de sensores colocados permite una mayor libertad al elegir las balizas con mejor recepción de la señal pero no afinar la estimación con un número mayor de muestras para determinar la intersección de las curvas que definen.

Como es posible ver en la figura 17 la posición de los sensores no es completamente uniforme, debido a que no toda la superficie de la planta estaba disponible por la disposición de los despachos y laboratorios. En este caso la altura de todas las balizas fue similar, a 170cm respecto al suelo.

## 4. Resultados

Tal y como viene recogido en la sección anterior, en la toma de medidas se recabaron datos desde dos fuentes: el posicionamiento local del robot y los cálculos del kit de sensores de UWB.

Para el análisis de los resultados se han considerado a los datos obtenidos por el posicionamiento local del robot como *ground truth*, es decir, como la posición verdadera del sensor establecido como objetivo, que está colocado encima del robot. A partir de esta consideración, es posible definir el error en la posición para cada uno de los ejes cartesianos como

$$\text{Error eje} = \text{Valor robot} - \text{Valor sensores} \quad (32)$$

El kit utilizado permite el posicionamiento en tres dimensiones, aunque no se presentarán y estudiarán los resultados en el eje vertical, ya que el sensor a posicionar se encontraba en todo momento encima del robot, a la misma altura.

Además de las diferencias en cada eje se evaluará la discrepancia en la posición, a partir del vector definido como la diferencia de los vectores de posición del robot y los sensores como se indica en la Figura 18.

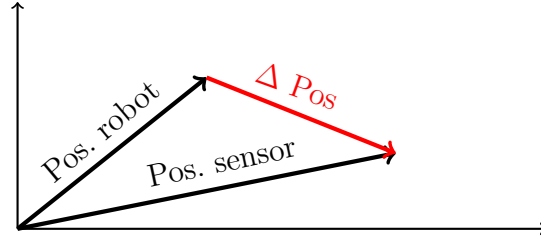


Figura 18: Definición del vector de error de posición.

Así, las componentes de dicho vector son las definidas en (32) y su módulo, que será interpretado como el error en posición se define como

$$\text{Error en posición} = \sqrt{(\text{Error } x)^2 + (\text{Error } y)^2} \quad (33)$$

A la hora de interpretar los resultados a partir de la toma de medidas y con el tratamiento especificado cabe recordar que en (32) se incluyen dos fuentes de error, por un lado el posicionamiento local del robot y por otro el posicionamiento por trilateración de los sensores de UWB.

La evaluación de la bondad del posicionamiento de los sensores, teniendo en cuenta la posibilidad de tomar con ellos tantas medidas como sea posible, es directa, dando en todos los casos una distribución gaussiana, como era de esperar y como se podía ver en la Figura 11; aun así, según los datos proporcionados por el fabricante y contrastados en trabajos previos nos da unos errores de como mínimo 10cm.

En el caso del robot los sistemas de posicionamiento local propios, basados en la fusión de datos de odometría y de posicionamiento de objetos arrojan una precisión de 10cm con un perfil también normal [16].

La adición de ambas fuentes de error, siguiendo el teorema del límite central, nos dará una distribución que tenderá con un número creciente de medias hacia otra distribución gaussiana con el error total en la posición del robot. Teniendo en cuenta que los errores de posicionamiento no son absolutos, será posible encontrar casos puntuales en los que la adición de ambas fuentes de error proporcione un error total con un valor menor que cada uno de los errores por separado, compensándose entre sí.

La toma de numerosas medidas con distintas rutas trata de minimizar este hecho, de tal forma que, de media, es razonable esperar un error en el posicionamiento en torno a los 20cm, en el caso de (33).

## **4.1. Laboratorio de Robótica del ICCAEx**

Como se comentaba en la sección de metodología, se diseñaron varias trayectorias a recorrer por el robot para así eliminar cualquier sesgo en los errores de su posicionamiento local.

Los resultados expuestos en esta sección son fruto de la composición de las medidas de las tres trayectorias con las que se tomaron medidas para cada configuración de balizas. Los datos con cada una ellas por separados se encuentran en el Apéndice 1.

### **4.1.1. 4 balizas**

Con la configuración de cuatro balizas definida en la Figura 14a se obtuvieron los datos recopilados en las Tablas 7 para la trayectoria vertical, 9 para la trayectoria en espiral y 8 para la trayectoria aleatoria.

Los datos acumulados de las tres trayectorias están recogidos en la Tabla 6, y representados en Figura 21

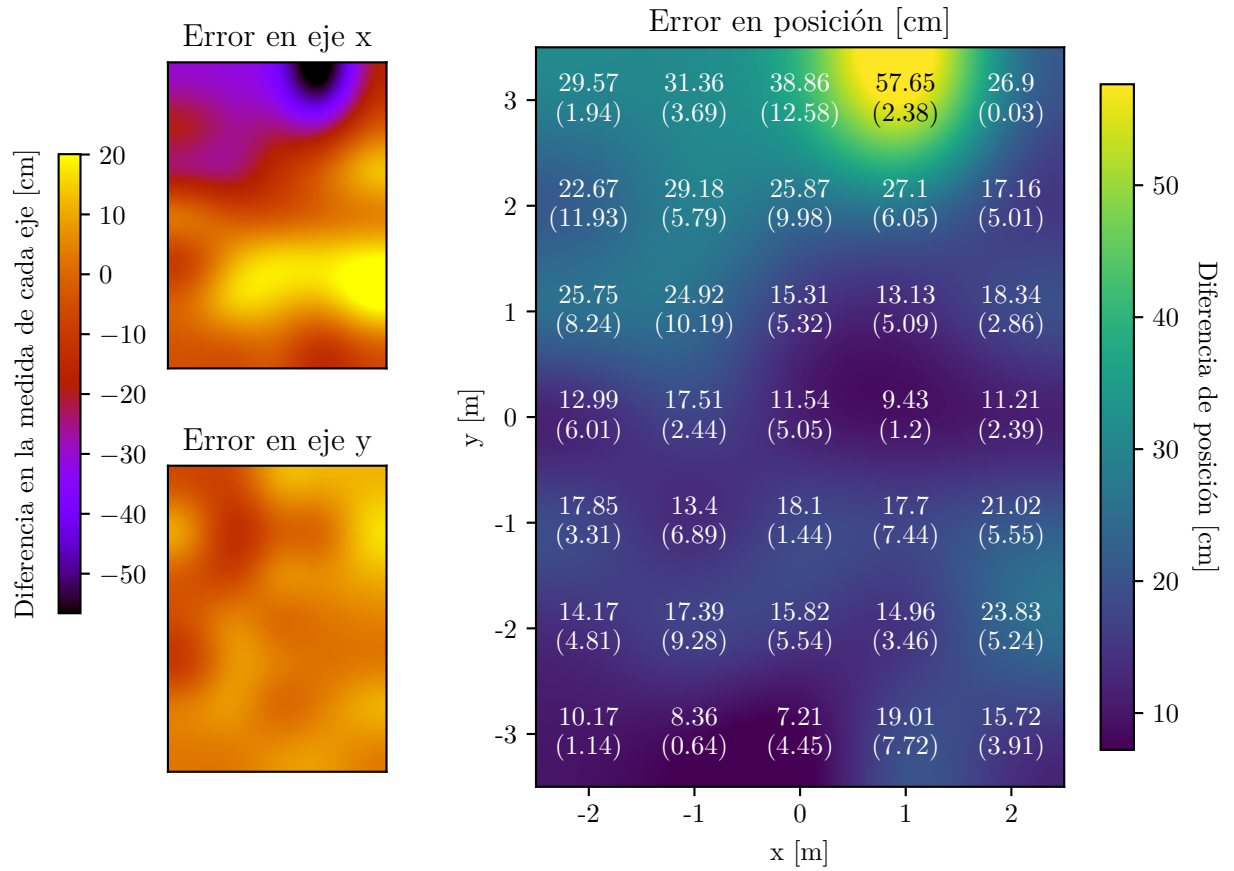


Figura 19: Resultados de las medidas en el laboratorio con 4 balizas. En esta representación y en todas las sucesivas figuran los valores de error obtenidos en los puntos del laboratorio donde se tomaron las medidas. A partir de ellos se ha realizado una interpolación con splines bicúbicos para obtener la distribución continua que se ha representado. Entre paréntesis se indica la desviación estándar de los errores obtenidos en cada punto.

Es posible observar un valor muy alto en la discrepancia en el eje  $x$  en el punto 34 –definido en la Figura 13a–. Teniendo en cuenta que en los puntos adyacentes no ocurre, es posible achacar esta alta discrepancia a puras fluctuaciones estadísticas o a una falta de visión directa entre objetivo y balizas, teniendo en cuenta que no se repite en más puntos.

Para tratar de comprobar el comportamiento general de la combinación de sensores y robot más allá de puntos con valores especialmente altos o bajos se recoge la Tabla ?? con los valores medios y dispersión del error a lo largo de cada uno de los ejes y del error en el posicionamiento en el caso de uso de 4 balizas.

	Eje $x$ (cm)	Eje $y$ (cm)	Eje $x$ (absoluto) (cm)	Eje $y$ (absoluto) (cm)	Posición (cm)
Media	-6.7	3.1	14.6	5.77	20.0
Mediana	-3.3	4.1	13.1	4.9	17.7
Desviación estándar	17.8	5.9	12.2	3.4	9.6
Máximo	20.1	14.4	56.6	14.4	57.6
Mínimo	-56.6	-12.4	0.5	0.7	7.2

Tabla 1: Resumen de los errores en el laboratorio con 4 balizas.

La cercanía entre los valores de media y mediana en todos los casos sugiere una distribución uniforme de los datos, lo que denotaría una densidad de probabilidad gaussiana, como se planteaba al inicio de la sección. Es posible comprobar este hecho con un histograma de los valores obtenidos en la Figura 20.

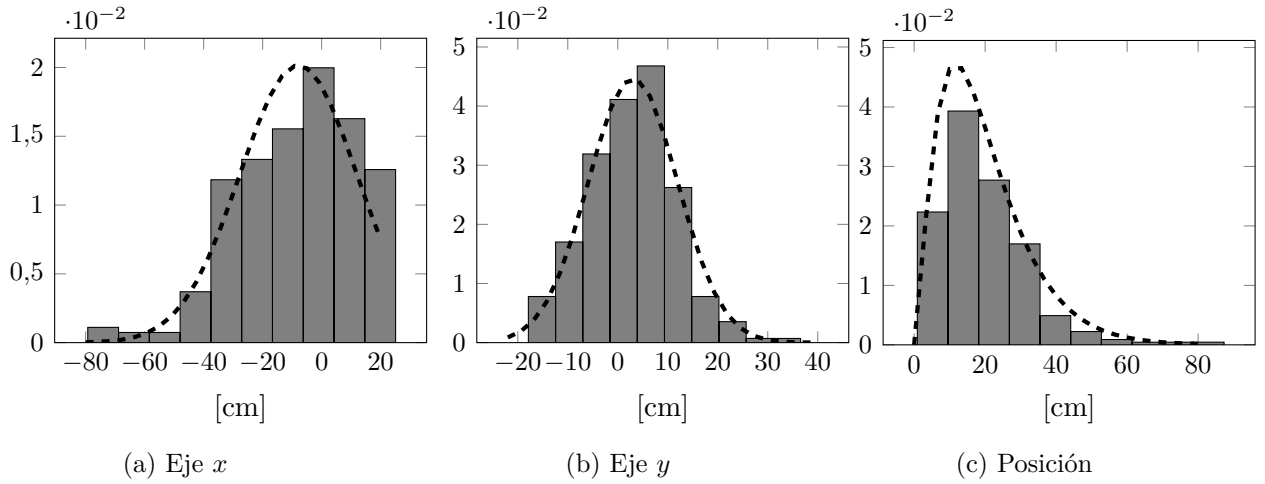


Figura 20: Histogramas con todas las medidas realizadas con 4 balizas, normalizados.

En los histogramas es claramente apreciable la naturaleza gaussiana de los valores obtenidos, por lo que es posible ajustar los valores a distribuciones de probabilidad normales y permite utilizar la media y la desviación estándar como parámetros principales para analizar el rendimiento del sistema establecido entre robot y sensores, aunque un volumen mayor de medidas hubiese resultado en mejores ajustes.

En el caso del error en la posición, fruto de la combinación de distribuciones gaussianas de acuerdo a (33) la parametrización usual es la de una función gamma [12].

#### 4.1.2. 6 balizas

Tras la colocación de dos balizas adicionales a lo largo de los extremos en el eje  $x$  tal y como se indicaba en la Figura 14b se obtuvieron los siguientes resultados recogidos en la Figura 21 y en la Tabla 10.

Al igual que en el caso de 4 balizas, los datos recopilados se encuentran en las Tablas 11 para la trayectoria vertical, 13 para la trayectoria en espiral y 12 para la trayectoria

aleatoria.

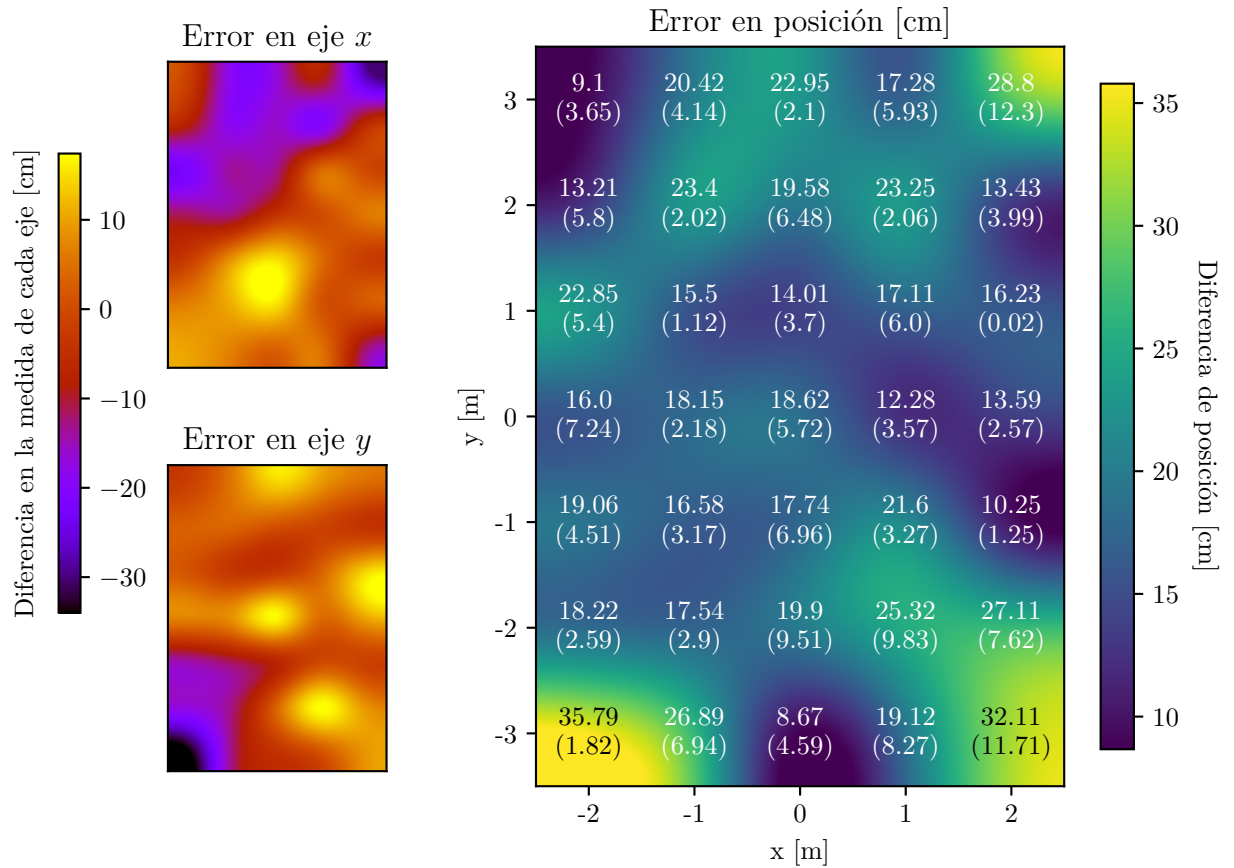


Figura 21: Resultados de las medidas del laboratorio con 6 balizas.  
Entre paréntesis se indica la desviación estándar de los errores obtenidos en cada punto.

En este caso es posible observar de nuevo valores extremales en los puntos 1 y 4, pero no en el 34, lo que confirma que aquel caso se trataba de un error por fluctuaciones estadísticas y no por un error en la configuración en la medida. Esta misma explicación es aplicable a este caso, que incluye valores máximos mucho más cercanos a la media, como se recoge en la tabla de valores medios.

	Eje $x$ (cm)	Eje $y$ (cm)	Eje $x$ (absoluto) (cm)	Eje $y$ (absoluto) (cm)	Posición (cm)
Media	-3.7	0.7	9.6	7.7	19.2
Mediana	-2.2	2.08	8.8	6.0	18.2
Desviación estándar	11.0	10.3	6.5	6.8	6.1
Máximo	15.9	17.4	24.9	33.9	35.8
Mínimo	-24.9	-33.9	0.1	0.1	8.6

Tabla 2: Resumen de los errores en el laboratorio con 6 balizas.

Los resultados de la toma de medidas con 6 balizas no suponen un cambio espe-

cialmente significativo en cuanto a los valores medios de las discrepancias en el posicionamiento, algo esperable teniendo en cuenta que el kit sigue utilizando solo 4 de las balizas en todo momento y las distancias entre sensores no llegan en ningún punto a ser excesivamente altas.

Sí que se puede observar no obstante una mejora en la desviación de las medidas, haciéndolas más estables. Esto se debe principalmente a la menor distancia entre las balizas y el objetivo en todos los puntos, fruto de una distribución más uniforme de dichas balizas.

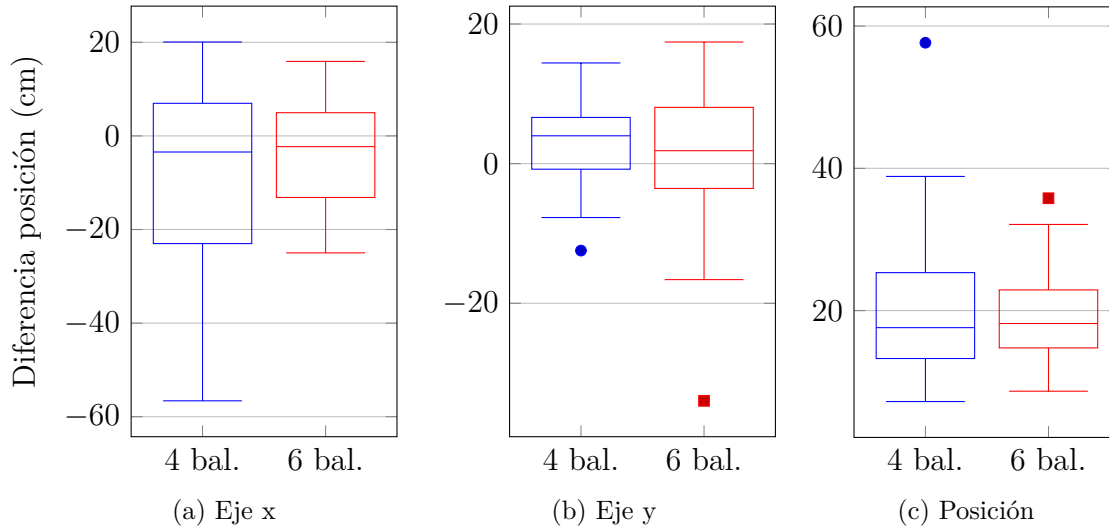


Figura 22: Comparación del rendimiento de las configuraciones de balizas en el laboratorio.

Aunque es posible observar en la Figura 22 una ligera dispersión mayor en el eje  $y$  en el caso de uso de 6 balizas, la diferencia en los valores extremos son despreciables, con las discrepancias en el eje  $x$  como componente claramente superior en la contribución al error en el posicionamiento total.

La razón es la colocación de las dos balizas añadidas. Como se puede observar en la Figura 14b, fueron colocadas en los extremos de este eje, por lo que es de esperar que la mejora en los resultados recaiga principalmente en esos valores.

Así, la adición de nuevas balizas no supondrá un beneficio en el posicionamiento si no se tiene en cuenta dónde se colocan, independientemente del número de nuevos sensores. Una distribución uniforme supondrá, usando este kit, mejores resultados que el uso de numerosas balizas colocadas de forma aleatoria.

Esta colocación fue elegida para tratar de tener una distribución lo más uniforme posible, acortando la mayor distancia entre las balizas en su configuración con 4 sensores. Es posible que un mayor número de balizas proporcione mejores resultados, especialmente si se añaden en los extremos del eje  $y$ ; no obstante, teniendo en cuenta que la dispersión en ese caso ya era lo suficientemente baja –teniendo sensores en los extremos pero con una distancia entre ellos de solo unos 5 metros–, no se procedió a

su análisis en beneficio de tomas de datos de mayor interés.

## **4.2. Edificio de Física**

Una vez obtenido los resultados en el laboratorio se repitió la toma de medidas de la misma forma en el edificio B de la Facultad de Física, en un entorno con obstáculos en los que las tomas de medidas se produjeron en todo momento en zonas cercanas a paredes y con un mismo recorrido al no tener opción por la disposición de la planta a colocar el robot en otros lugares.

Así, la presentación de los siguientes datos corresponden a tres medidas de la misma trayectoria en cada caso.

Las medidas en este escenario se realizaron colocando el origen del sistema de referencia en una de las esquinas de la superficie en las que se realizaron, ya que a diferencia del laboratorio, no era posible acceder al centro para establecer ahí dicho origen. Esto se ve reflejado en la presentación de los resultados, en los que ahora no aparecen coordenadas negativas.

### **4.2.1. 4 balizas**

Colocando 4 sensores como balizas con la disposición indicada en la Figura 17a se obtuvieron los resultados representados en la Figura 23 y recogidos en la Tabla 16.



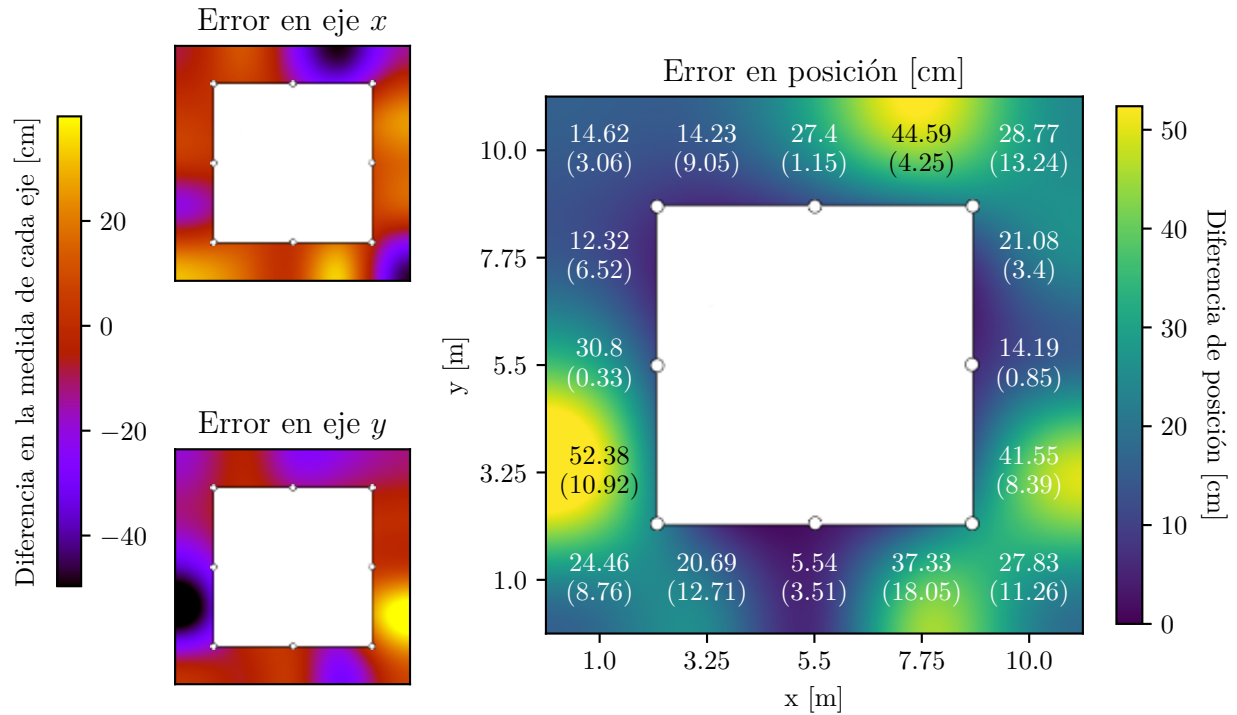


Figura 23: Resultados de las medidas en el edificio de Física con 4 balizas.  
Entre paréntesis se indica la desviación estándar de los errores obtenidos en cada punto.

En general, a lo largo de toda la sección se pueden apreciar unas desviaciones estándar mayores que en el caso del laboratorio: esto se debe a una disposición menos favorable y a un menor volumen de medidas en cada punto.

	Eje $x$ (cm)	Eje $y$ (cm)	Eje $x$ (absoluto) (cm)	Eje $y$ (absoluto) (cm)	Posición (cm)
Media	0.5	-8.6	15.3	15.4	26.1
Mediana	3.4	2.08	13.7	11.2	25.9
Desviación estándar	18.4	10.2	18.7	13.8	12.5
Máximo	27.8	39.9	40.8	49.6	52.3
Mínimo	-40.8	-49.6	2.3	0.5	5.5

Tabla 3: Resumen de los errores en el edificio de Física con 4 balizas.

A comparación de las medidas con el mismo número de balizas en el laboratorio se obtienen en este caso tanto un valor medio como una dispersión mayor, fruto del entorno menos favorable.

Las distancias entre objetivo y balizas es, en este entorno, muy superior con al

menos una de dichas balizas a más de 10 metros de forma constante. Así, los resultados obtenidos aquí coinciden con las conclusiones extraídas de las medidas del laboratorio, donde una configuración de balizas más alejadas proporcionaba peores resultados.

#### 4.2.2. 6 balizas

Añadiendo dos balizas como se indicaba en 17b se obtuvieron los resultados que se han representado en la Figura 24 y recogidos en la Tabla 15.

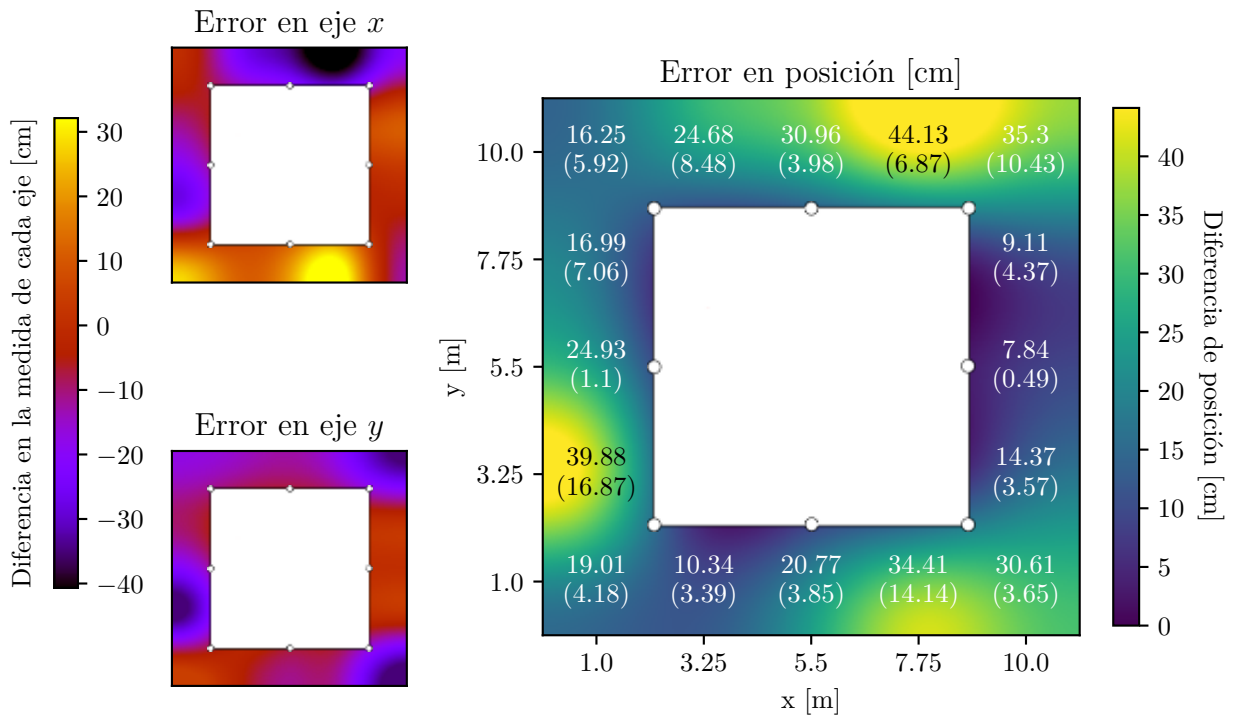


Figura 24: Resultados de las medidas en el edificio de Física con 6 balizas. Entre paréntesis se indica la desviación estándar de los errores obtenidos en cada punto.

Aunque los resultados en este caso son menores, se sigue encontrando el máximo en el punto 10. Es más, los valores desde el punto 9 al 13 no tienen una mejora significativa con 6 balizas debido a la posición de la baliza adicional en esa pared.

Este efecto ya fue discutido en el análisis de las medidas del laboratorio. Una de las conclusiones en aquel caso era que una distribución uniforme de las balizas proporcionaba mejores resultados, y no simplemente un número mayor sin considerar su posición.

Debido a la distribución de la planta las dos balizas más cercanas a los puntos mencionados se colocaron a poco más de dos metros entre ellas. Esta baja separación hace que las medidas que las usan no presenten una gran mejora respecto a los casos en los que solo una de ellas participaba en los cálculos.

	Eje $x$ (cm)	Eje $y$ (cm)	Eje $x$ (absoluto) (cm)	Eje $y$ (absoluto) (cm)	Posición (cm)
Media	-4.7	-11.9	16.0	12.1	23.7
Mediana	-3.8	-10.4	17.7	10.3	22.7
Desviación estándar	18.7	10.5	10.8	10.1	10.9
Máximo	32.1	1.1	40.8	30.8	44.1
Mínimo	-40.7	-30.9	3.3	0.4	7.8

Tabla 4: Resumen de los errores en el edificio de Física con 6 balizas.

A pesar de las escasas mejoras en los puntos mencionados anteriormente, los valores medios recogidos en la Tabla 4 sí que presentan mejoras significativas en este caso con más balizas.

#### 4.2.3. 8 balizas

Por último, se añadieron otros dos sensores para tener 8 balizas tal y como se muestra en la Figura 17c, tras lo que se obtuvieron los resultados representados en la Figura 25 y recogidos en la Tabla 14.

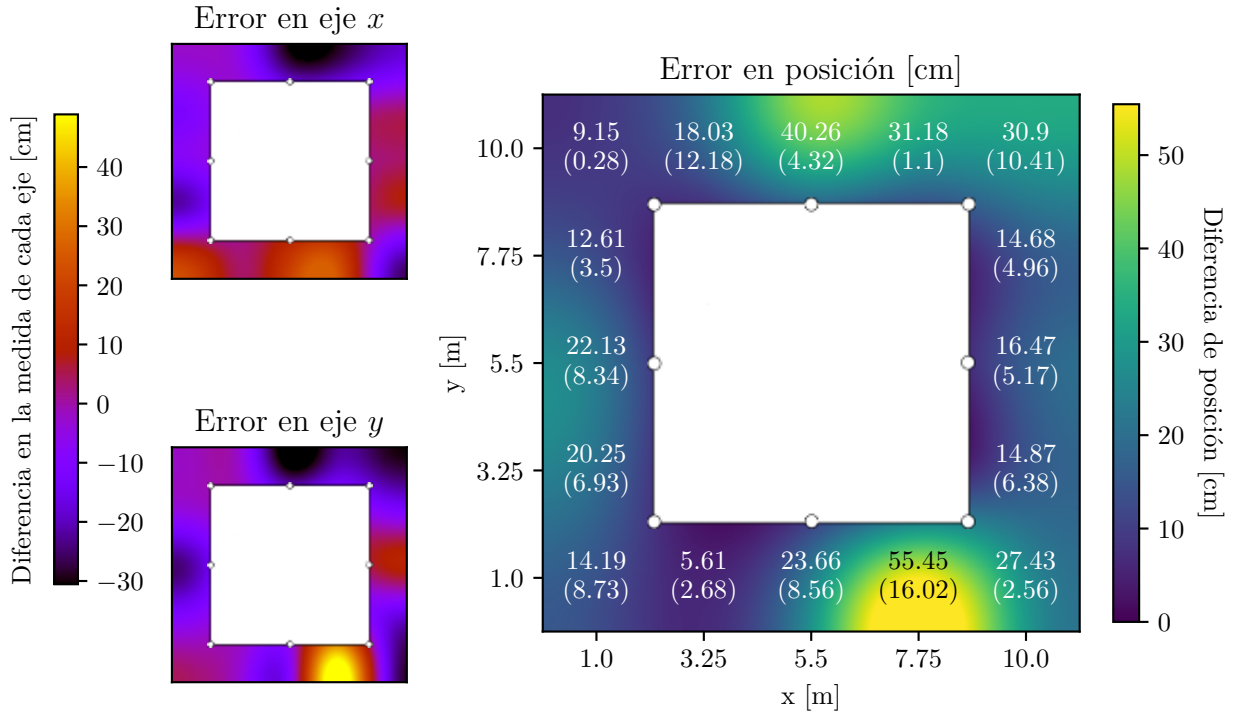


Figura 25: Resultados de las medidas en el edificio de Física con 8 balizas.  
Entre paréntesis se indica la desviación estándar de los errores obtenidos en cada punto.

En este caso es posible observar una mejora generalizada tanto en valores de discrepancia como en sus desviaciones, salvo en el punto 4. Tras revisar los datos el valor tan alto se debe a un error mayor en el posicionamiento de los sensores, que indicaban en el eje  $y$  una posición que colocaría al robot fuera de la planta.

Este hecho se repite a lo largo de las tres medidas realizadas, aunque con distintos valores –de ahí también el alto valor en la desviación estándar de las medidas en el punto–, por lo que es posible que se trate de un caso en el que la tercera baliza utilizada se haya visto ocultada por una de las columnas.

	Eje $x$ (cm)	Eje $y$ (cm)	Eje $x$ (absoluto) (cm)	Eje $y$ (absoluto) (cm)	Posición (cm)
Media	-3.5	-6.2	11.9	13.7	22.3
Mediana	-4.7	-7.1	11.4	9.6	19.1
Desviación estándar	13.6	12.3	7.5	12.3	12.2
Máximo	21.0	48.9	25.8	48.9	55.5
Mínimo	-40.7	-30.5	3.3	1.5	5.6

Tabla 5: Resumen de los errores en el edificio de Física con 8 balizas.

Los valores medios recogidos en la Tabla 5, debido al número bajo de datos, se ven claramente influenciados por los valores anormalmente altos del punto 4, también con una desviación estándar más alta que en el caso de 6 balizas. La diferencia entre media y mediana lo muestra, aunque ambos valores son los mejores obtenidos en el edificio de Física.

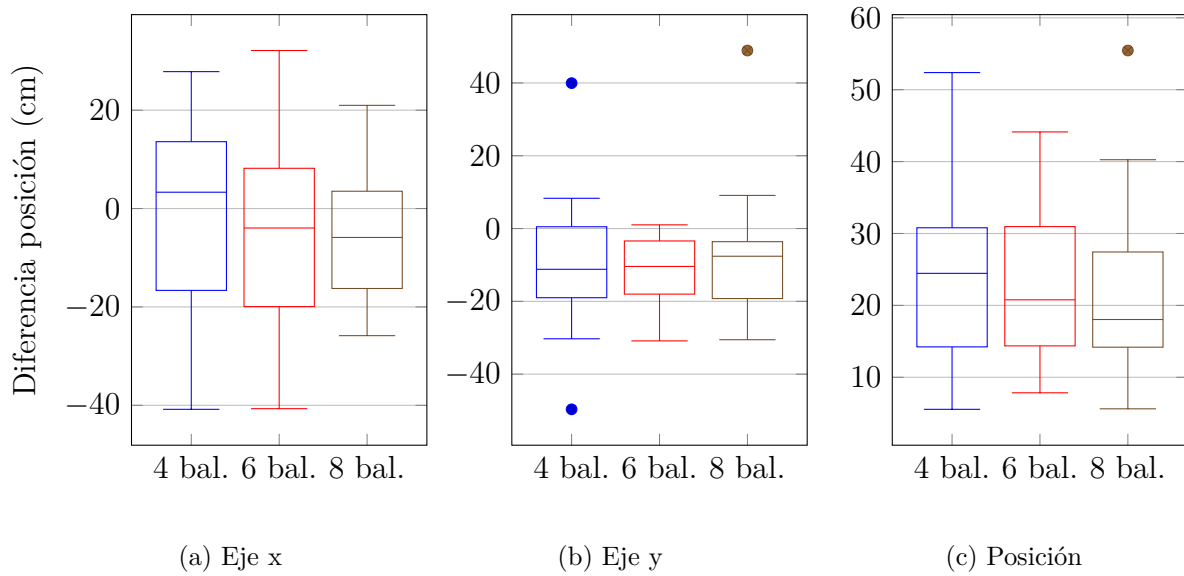


Figura 26: Comparación del rendimiento de las configuraciones de balizas en el edificio de Física.

Los diagramas de cajas de la Figura 26 muestran unos resultados similares a los obtenidos en el laboratorio: un número mayor de balizas resulta en unos valores menores en el error del posicionamiento del objetivo.

Además, confirma la conclusión expuesta en la comparación de aquellos resultados: debido a la utilización de 4 balizas en todo momento del kit la posición en la que se colocan es crítica.

Es posible observar este hecho en los diagramas de cada eje. En la configuración con 6 balizas se añadieron en los extremos del eje  $y$ , y es en dicho eje donde se observan unas mejoras en mayor medida, a pesar de que, como se ha comentado, una de las balizas tuviera una colocación mejorable.

El mismo fenómeno se da en el caso del empleo de 8 balizas, donde la colocación de balizas adicionales en los extremos del eje  $x$  proporciona unos valores con una dispersión mucho menor en este eje.

Aunque los resultados en este escenario no ideal son, como era de esperar, peores que en el ideal, la diferencia relativa en el caso de 4 balizas es del 23,4 % para los valores medios y 18,9 % para las desviaciones estándar del posicionamiento, que equivalen a unos pocos centímetros. Estas diferencias se deben principalmente a la mayor distancia entre balizas y objetivo, y no tanto a la presencia de obstáculos. La anchura de las columnas no parece ser suficiente para evitar la línea de visión directa entre las balizas

y el objetivo, aunque sí que es posible apreciar su efecto puntualmente como en el caso del uso de 8 balizas, dando en uno de los puntos un resultado muy alejado de la media.

La diferencia en los casos con 6 balizas es similar, con una diferencia relativa en los valores medios del 18,9% y 44,4% para las desviaciones estándar del error en el posicionamiento, de nuevo achacables a la mayor distancia entre balizas y objetivo.

## 5. Conclusiones

Para determinar la precisión en el posicionamiento de un kit de sensores de Ultra-wide Band se ha procedido a tomar varias medidas con la ayuda de un robot capaz de desplazarse de forma autónoma en dos escenarios, uno libre de objetos y un segundo con una superficie mayor y con obstáculos que pudieran impedir en ciertas ocasiones el correcto desempeño del posicionamiento del kit utilizado.

Además de las distintas localizaciones, en cada uno de los dos escenarios se han recogido datos con distintas configuraciones de balizas, de tal forma que ha sido posible determinar el efecto de su colocación en el posicionamiento del objetivo.

Las medidas en los dos escenarios han arrojado resultados similares: una distribución uniforme de balizas proporciona mejores resultados fruto de dar opción al kit a utilizar sensores más cercanos para el posicionamiento del objetivo.

Así, para maximizar el rendimiento de este kit no solo se deberán usar un número mayor de balizas, sino que será necesario un estudio que permita obtener una configuración en la zona de interés tal que los objetivos a posicionar se encuentren a la menor distancia posible de al menos 4 balizas.

En ninguno de los dos escenarios contemplados en este trabajo se han obtenido valores medios de error en el posicionamiento mayores a 30cm, teniendo en cuenta además que en todos ellos, además del error de precisión dado por la tecnología de UWB, se añadía el posible error de posicionamiento del robot. Es decir, es posible encontrar en superficies del orden de decenas de metros cuadrados, errores medios de, al menos, dos órdenes de magnitud menores.

Estos resultados abren la puerta a su empleo en el posicionamiento de robots teniendo en cuenta que dicho error entrará en la mayoría de casos en sus propias dimensiones o con humanos, siendo 30cm la distancia equivalente a dar un paso.

En superficies aún mayores que las tratadas se deberían obtener unos resultados parecidos siempre que se realice una colocación de un número mayor de balizas respetando las condiciones de homogeneidad en las distancias ya discutida.

Aun así, el problema del posicionamiento local no está completamente resuelto. Los resultados expuestos en este trabajo son valores medios obtenidos a partir de numerosas medidas de un kit de UWB, pero en algunas de las aplicaciones donde se necesite realizar el posicionamiento de forma veloz la naturaleza aleatoria de las señales empleadas pueden provocar un pobre rendimiento de estas técnicas.

Obstáculos dinámicos también juegan un gran papel en la precisión del posicionamiento local. En el caso del UWB la visión directa entre los sensores es crítica para su correcto funcionamiento y la posición de las balizas debe contemplarla, por lo que escenarios con obstáculos impredecibles evitarán su correcto funcionamiento.

La proliferación de la tecnología UWB da una nueva alternativa a los sistemas de

posicionamiento local ya establecidos y discutidos en este trabajo. Tratando de evitar en la medida de lo posible los casos donde aparecen sus puntos débiles, proporciona una precisión y robustez muy superior a sus alternativas, por lo que se presenta en estos momentos como una de la mejores opciones en este ámbito.



## 6. Bibliografía

- [1] F. Mazhar, M. G. Khan, and B. Sällberg, “Precise indoor positioning using UWB: A review of methods, algorithms and implementations,” *Wireless Personal Communications*, vol. 97, pp. 4467–4491, 2017.
- [2] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation theory*. Prentice-Hall, 1993.
- [3] Z. Şahinoğlu, S. Gezici, and I. Güvenç, *Ultra-wideband Positioning Systems*. Cambridge University Press, 2008.
- [4] H. Soganci, S. Gezici, and H. V. Poor, “Accurate positioning in ultra-wideband systems,” *IEEE Wireless Communications*, vol. 18, 2011.
- [5] J.-N. Shim, H. Park, C.-B. Chae, D. K. Kim, and Y. C. Eldar, “Cramér–Rao Lower Bound on AoA Estimation Using an RF Lens-Embedded Antenna Array,” *IEEE Antennas and Wireless Propagation Letters*, vol. 17, no. 12, pp. 2359–2363, 2018.
- [6] G. M. Mendoza-Silva, J. Torres-Sospedra, and J. Huerta, “A meta-review of indoor positioning systems,” *Sensors*, vol. 19, p. 4507, 2019.
- [7] Decawave Ltd, *UWB Regulations: A Summary of Worldwide Telecommunications Regulations governing the use of Ultra-Wideband radio*, 1.2 ed., 2015.
- [8] K.-L. Du and M. N. S. Swamy, *Wireless Communication Systems*. Cambridge University Press, 2010.
- [9] C. Nguyen and M. Miao, *Design of CMOS RFIC Ultra-Wideband Impulse Transmitters and Receivers*. SpringerBriefs in Electrical and Computer Engineering, 2017.
- [10] Decawave Ltd, *DWM1001 System Overview*, 1.0 ed., 2017.
- [11] R. Simedroni, E. Puschita, T. Palade, P. Dolea, C. Codau, R. Buta, and A. Pastrav, “Indoor positioning using Decawave MDEK1001,” in *2020 International Workshop on Antenna Technology (iWAT)*, pp. 1–4, 2020.
- [12] A. R. J. Ruiz and F. S. Granja, “Comparing Ubisense, BeSpoon, and DecaWave UWB Location Systems: Indoor Performance Analysis,” *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 8, pp. 2106–2117, 2017.

- [13] J. Kulmer, S. Hinteregger, B. Großwindhager, M. Rath, M. S. Bakr, E. Leitinger, and K. Witrissal, “Using DecaWave UWB transceivers for high-accuracy multipath-assisted indoor positioning,” in *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1239–1245, 2017.
- [14] Decawave Ltd, *DWM1001 Datasheet*, 1.4 ed., 2017.
- [15] M. Krumbein, “Heading estimation of a mobile robot using multiple UWB position sensors,” Master’s thesis, Case Western Reserve University, 2019.
- [16] A. P. López, “Posicionamiento local de robots móviles.” Trabajo de Fin de Grado, 2018.



## 7. Apéndice 1

### 7.1. Medidas en el laboratorio de Robótica del ICCAEx

#### 7.1.1. Medidas con 4 balizas

Tabla 6: Valor medio y desviación estándar de los errores para cada eje y para el posicionamiento en cada punto para el conjunto de trayectorias con 4 balizas.

Punto	Error en eje $x$ [cm]		Error en eje $y$ [cm]		Error en posición [cm]	
	Valor medio	Desviación estándar	Valor medio	Desviación estándar	Valor medio	Desviación estándar
1	-3,34	7,01	3,05	4,27	10,17	1,14
2	-1,86	4,18	3,81	0,56	8,36	0,64
3	-1,15	1,62	6,23	4,60	7,21	4,45
4	-10,54	6,32	7,90	1,02	19,01	7,72
5	-3,02	14,44	5,38	2,51	15,72	3,91
6	1,13	13,95	3,88	1,40	14,17	4,81
7	13,74	10,62	7,81	4,45	17,39	9,28
8	13,13	8,71	0,81	0,80	15,82	5,54
9	10,46	7,18	4,16	2,82	14,96	3,46
10	20,06	7,80	10,14	4,09	23,83	5,24
11	-6,46	13,53	-7,72	4,90	17,85	3,31
12	9,53	8,38	7,00	2,06	13,40	6,89
13	16,21	2,61	4,09	3,91	18,10	1,44
14	16,48	7,87	2,04	3,47	17,70	7,44
15	19,82	6,56	3,28	2,61	21,02	5,55
16	0,97	5,73	-3,62	4,71	12,99	6,01
17	-4,32	9,59	4,44	6,48	17,51	2,44
18	-3,54	7,69	0,93	0,61	11,54	5,05
19	0,55	6,68	2,42	3,69	9,43	1,20
20	4,41	6,45	4,58	5,20	11,21	2,39
21	-22,66	10,77	-2,74	9,29	25,75	8,24
22	-23,33	10,62	-7,30	1,17	24,92	10,19
23	-13,40	5,05	4,93	4,44	15,31	5,32
24	-2,70	7,63	5,93	4,94	13,13	5,09
25	11,06	8,65	10,84	3,29	18,34	2,86
26	-20,22	9,67	5,91	9,25	22,67	11,93
27	-25,63	7,46	-12,45	3,00	29,18	5,79
28	-24,55	10,38	-2,28	6,19	25,87	9,98
29	-25,92	6,64	0,69	3,35	27,10	6,05
30	-5,83	7,97	14,42	1,77	17,16	5,01
31	-28,22	1,59	-4,31	2,22	29,57	1,94
32	-30,45	3,22	-6,87	2,63	31,36	3,69
33	-37,37	11,18	8,23	6,68	38,86	12,58
34	-56,61	2,56	9,84	0,47	57,65	2,38
35	-22,70	1,35	12,11	2,80	26,90	0,03

### 7.1.2. Medidas de cada trayectoria con 4 balizas

Tabla 7: Valor medio y desviación estándar de los errores para cada eje y para el posicionamiento en cada punto para la trayectoria vertical con 4 balizas.

Punto	Error en eje $x$ [cm]		Error en eje $y$ [cm]		Error en posición [cm]	
	Valor medio	Desviación estándar	Valor medio	Desviación estándar	Valor medio	Desviación estándar
1	3,67	5,51	7,32	2,80	9,03	4,87
2	2,31	3,98	4,37	4,52	7,72	1,05
3	0,47	1,37	1,64	2,28	2,76	1,54
4	-4,23	6,32	8,92	2,79	11,30	4,19
5	11,42	3,11	2,87	1,53	11,81	3,35
6	1,17	4,63	5,68	1,05	7,45	0,88
7	8,55	0,87	1,69	1,48	8,84	0,80
8	14,94	5,66	-0,12	2,76	15,23	5,55
9	11,21	1,43	7,04	5,43	13,72	4,29
10	12,11	6,93	15,50	3,10	20,38	5,42
11	-15,30	1,25	-2,38	1,69	15,56	1,43
12	1,65	5,99	4,17	1,33	7,09	2,74
13	15,61	2,76	0,98	5,61	16,67	2,40
14	21,30	2,82	6,28	5,68	23,03	1,68
15	21,63	0,99	5,58	4,13	22,66	1,85
16	2,86	1,48	2,98	3,84	4,58	3,61
17	-16,40	9,40	-0,71	9,11	19,37	8,12
18	-2,22	2,91	1,40	3,99	5,34	1,64
19	0,86	2,75	3,36	7,29	7,90	3,23
20	5,53	3,47	11,09	2,83	12,66	3,64
21	-28,84	4,45	9,96	1,73	30,58	4,29
22	-29,03	0,96	-7,00	4,47	30,16	1,74
23	-11,58	3,47	11,11	5,93	16,21	6,49
24	-3,57	5,38	1,65	2,03	6,22	3,13
25	17,79	0,89	11,86	0,51	21,39	0,69
26	-28,98	3,90	18,98	8,79	35,92	1,47
27	-33,58	5,71	-8,23	2,34	34,79	4,85
28	-17,81	1,29	6,48	5,23	19,66	1,31
29	-29,12	1,42	-3,66	1,90	29,40	1,65
30	0,10	2,28	14,71	1,54	14,87	1,62
31	-26,63	6,00	-2,09	6,56	27,63	5,44
32	-33,67	3,54	-9,50	1,58	35,05	3,25
33	-26,19	3,44	1,54	1,55	26,28	3,46
34	-54,04	10,78	10,31	3,93	55,28	10,15
35	-21,35	1,10	14,92	9,48	26,93	6,67

Tabla 8: Valor medio y desviación estándar de los errores para cada eje y para el posicionamiento en cada punto para la trayectoria aleatoria con 4 balizas.

Punto	Error en eje $x$ [cm]		Error en eje $y$ [cm]		Error en posición [cm]	
	Valor medio	Desviación estándar	Valor medio	Desviación estándar	Valor medio	Desviación estándar
1	-10,35	9,50	-1,22	4,13	11,31	9,37
2	-6,04	3,68	3,26	5,38	9,01	2,91
3	-2,77	3,14	10,83	1,11	11,67	0,05
4	-16,86	30,78	6,89	2,17	26,73	23,87
5	-17,46	7,98	7,89	1,31	19,63	6,87
6	-15,97	2,35	3,69	2,76	16,63	2,29
7	4,13	2,48	12,12	0,46	13,03	0,74
8	1,67	8,39	0,71	4,47	9,34	2,51
9	1,32	0,90	0,34	12,86	11,49	6,01
10	17,41	4,00	9,33	0,81	19,87	3,48
11	-16,74	9,43	-14,22	3,13	22,54	8,56
12	5,80	2,81	7,81	0,89	10,11	1,03
13	13,35	6,74	9,61	1,92	17,55	3,41
14	5,39	3,86	2,05	3,42	7,18	2,87
15	11,03	2,82	4,62	6,12	13,55	2,20
16	-6,81	11,06	-7,66	7,11	16,10	4,32
17	-3,60	11,56	0,44	7,27	14,06	1,35
18	-13,56	5,96	0,08	12,01	17,72	7,03
19	-7,77	3,73	-2,50	4,99	9,56	3,76
20	-4,00	3,38	-1,65	5,70	7,84	1,09
21	-31,63	6,36	-6,20	3,03	32,52	5,57
22	-32,51	6,77	-8,86	4,17	33,92	6,92
23	-20,29	4,63	2,76	5,93	21,32	4,62
24	-11,58	4,85	3,29	7,37	14,86	1,33
25	-1,15	2,73	14,26	9,47	14,52	9,54
26	-24,94	4,16	-0,11	2,60	25,09	4,08
27	-27,66	1,91	-14,92	2,11	31,55	0,65
28	-39,22	10,62	-6,71	1,63	39,96	10,10
29	-31,98	6,41	4,49	6,61	33,09	5,69
30	-17,09	3,49	16,43	5,56	24,11	4,91
31	-29,81	11,72	-6,53	8,45	31,50	12,15
32	-27,24	2,39	-4,23	1,98	27,67	2,03
33	-48,55	23,16	14,91	15,56	51,44	26,68
34	-59,17	13,47	9,37	5,40	60,03	14,00
35	-24,04	5,29	9,31	6,30	26,87	3,21

Tabla 9: Valor medio y desviación estándar de los errores para cada eje y para el posicionamiento en cada punto para la trayectoria en espiral con 4 balizas.

Punto	Error en eje $x$ [cm]		Error en eje $y$ [cm]		Error en posición [cm]	
	Valor medio	Desviación estándar	Valor medio	Desviación estándar	Valor medio	Desviación estándar
1	5,67	2,21	7,35	2,13	8,45	3,87
2	-2,67	2,44	3,26	10,74	9,42	6,12
3	4,20	1,51	8,69	3,13	7,98	5,42
4	-3,04	3,18	3,56	5,38	8,01	4,95
5	-8,02	5,15	7,10	2,4	13,42	7,42
6	18,19	3,01	2,28	2,28	18,44	3,25
7	28,54	2,99	9,62	2,43	30,30	2,04
8	22,77	1,04	1,83	1,14	22,88	0,94
9	18,87	1,91	5,10	1,95	19,69	1,32
10	30,66	0,15	5,59	2,04	31,23	0,51
11	12,65	6,79	-6,56	2,90	15,46	4,32
12	21,13	5,44	9,03	1,90	22,98	5,75
13	19,67	1,84	1,70	3,74	20,07	2,12
14	22,76	1,40	-2,21	1,02	22,88	1,49
15	26,79	0,33	-0,37	1,82	26,86	0,36
16	6,85	16,46	-6,18	2,53	18,28	5,31
17	7,05	11,79	13,57	1,49	19,08	3,30
18	5,14	11,36	1,32	0,84	11,54	4,97
19	8,57	1,99	6,40	0,62	10,83	1,21
20	11,69	1,70	4,30	4,95	13,14	3,13
21	-7,52	2,43	-11,98	4,64	14,15	5,22
22	-8,45	1,08	-6,04	3,68	10,67	2,94
23	-8,34	0,99	0,91	0,15	8,39	0,97
24	7,05	12,58	12,85	10,78	18,31	12,40
25	16,54	0,36	6,40	7,64	19,10	2,88
26	-6,74	4,40	-1,13	0,50	7,00	4,15
27	-15,65	1,20	-14,22	1,20	21,21	0,08
28	-16,64	3,74	-6,60	3,48	18,00	4,73
29	-16,68	6,09	1,24	7,81	18,82	4,89
30	-0,49	2,68	12,13	3,50	12,49	3,29
31	7,85	2,46	-5,18	3,53	11,28	4,31
32	4,65	2,99	7,47	5,29	11,59	2,57
33	-9,55	8,15	7,84	4,67	12,18	5,43
34	-5,52	3,13	-7,78	2,64	9,15	4,25
35	-10,35	6,21	7,13	2,69	8,97	3,16

### 7.1.3. Medidas con 6 balizas

Tabla 10: Valor medio y desviación estándar de los errores para cada eje y para el posicionamiento en cada punto para el conjunto de trayectorias con 6 balizas.

Punto	Error en eje $x$ [cm]		Error en eje $y$ [cm]		Error en posición [cm]	
	Valor medio	Desviación estándar	Valor medio	Desviación estándar	Valor medio	Desviación estándar
1	10,09	3,31	-33,97	1,29	35,79	1,82
2	7,05	15,39	-12,29	13,18	26,89	6,94
3	3,86	7,59	-3,78	1,57	8,67	4,59
4	5,15	18,16	2,09	7,05	19,12	8,27
5	-11,09	26,34	9,05	7,68	32,11	11,71
6	6,64	1,82	-16,61	2,46	18,22	2,59
7	11,75	0,42	-12,35	4,38	17,54	2,90
8	15,92	11,34	3,53	2,44	19,90	9,51
9	1,54	16,51	17,42	3,40	25,32	9,83
10	2,33	23,59	8,91	9,46	27,11	7,62
11	-2,34	7,62	-14,30	6,77	19,06	4,51
12	8,82	4,65	-11,12	6,35	16,58	3,17
13	14,78	4,29	-6,63	7,68	17,74	6,96
14	4,76	14,34	-0,11	9,35	21,60	3,27
15	-1,87	6,31	-1,20	6,67	10,25	1,25
16	-11,64	7,88	7,30	6,28	16,00	7,24
17	-12,50	2,37	8,82	3,16	18,15	2,18
18	-3,97	4,00	16,99	5,76	18,62	5,72
19	0,14	5,11	4,60	2,95	12,28	3,57
20	5,85	2,99	10,47	0,29	13,59	2,57
21	-21,27	5,53	1,62	5,88	22,85	5,40
22	-13,58	1,56	-2,99	3,06	15,50	1,12
23	-12,73	3,61	-2,21	2,80	14,01	3,70
24	4,17	6,30	3,71	15,29	17,11	6,00
25	0,23	5,87	14,77	1,19	16,23	0,02
26	-8,69	6,61	3,23	7,42	13,21	5,80
27	-18,76	4,02	3,25	12,47	23,40	2,02
28	-15,75	5,81	0,15	11,02	19,58	6,48
29	-20,11	3,82	-1,26	6,22	23,25	2,06
30	-5,28	2,59	-3,33	6,18	13,43	3,99
31	-2,25	7,30	-1,30	5,45	9,10	3,65
32	-19,10	5,26	5,95	2,03	20,42	4,14
33	-16,61	3,77	13,83	1,08	22,95	2,10
34	-9,55	10,15	8,84	4,20	17,28	5,93
35	-24,99	14,88	4,33	6,81	28,80	12,30



#### 7.1.4. Medidas de cada trayectoria con 6 balizas

Tabla 11: Valor medio y desviación estándar de los errores para cada eje y para el posicionamiento en cada punto para la trayectoria vertical con 6 balizas.

Punto	Error en eje $x$ [cm]		Error en eje $y$ [cm]		Error en posición [cm]	
	Valor medio	Desviación estándar	Valor medio	Desviación estándar	Valor medio	Desviación estándar
1	8,36	2,23	-32,17	5,51	33,34	5,33
2	1,09	12,31	-29,85	17,80	34,64	12,67
3	-2,90	1,82	-1,60	3,99	5,07	2,13
4	2,52	7,31	4,70	3,51	9,40	2,41
5	-10,02	23,69	7,90	11,90	27,52	10,42
6	8,94	3,01	-19,59	1,64	21,75	1,50
7	12,00	1,53	-17,60	1,91	21,39	1,36
8	11,90	3,75	0,24	5,46	13,39	2,52
9	-14,33	6,69	16,22	1,84	22,10	5,29
10	-10,48	3,44	21,39	4,73	24,19	4,04
11	-4,99	9,47	-9,33	5,45	14,96	2,72
12	2,25	8,16	-17,17	2,22	18,91	3,75
13	9,86	1,10	-3,44	4,71	11,15	2,84
14	7,96	15,87	4,31	6,44	18,95	4,01
15	-5,31	2,51	7,69	7,57	11,36	4,69
16	-22,62	2,69	9,64	1,14	24,67	2,07
17	-14,11	6,05	8,45	1,68	17,04	4,41
18	-8,62	0,68	14,05	2,29	16,52	2,12
19	-2,17	2,24	3,26	10,74	9,42	6,86
20	7,20	1,31	10,69	4,13	12,98	4,07
21	-29,00	10,10	-1,22	7,50	30,22	9,36
22	-14,10	11,20	-6,81	7,53	16,25	12,78
23	-7,75	2,74	0,77	4,51	9,04	2,59
24	13,04	11,25	-16,90	8,48	22,37	12,40
25	-4,90	0,89	15,29	4,11	16,19	3,63
26	-17,71	6,19	10,20	4,82	21,30	5,08
27	-19,92	10,45	0,44	5,23	20,69	10,28
28	-15,95	2,99	10,13	4,71	19,66	1,21
29	-14,86	8,63	-10,02	6,48	20,43	4,50
30	-3,77	6,45	5,38	13,95	15,83	5,35
31	2,27	4,12	-0,67	2,24	4,65	2,44
32	-21,46	4,79	5,40	0,72	22,14	4,79
33	-11,45	8,84	15,26	1,89	20,12	6,39
34	-5,22	8,81	3,64	6,32	12,20	3,06
35	-17,35	4,21	4,13	1,69	17,97	3,96

Tabla 12: Valor medio y desviación estándar de los errores para cada eje y para el posicionamiento en cada punto para la trayectoria aleatoria con 6 balizas.

Punto	Error en eje $x$ [cm]		Error en eje $y$ [cm]		Error en posición [cm]	
	Valor medio	Desviación estándar	Valor medio	Desviación estándar	Valor medio	Desviación estándar
1	7,19	6,03	-35,09	1,07	36,33	1,02
2	-8,10	6,33	-8,91	13,63	17,80	7,35
3	0,02	2,48	-5,25	1,38	5,80	1,42
4	-15,66	4,94	9,11	3,36	18,33	5,29
5	-43,87	14,88	18,97	7,07	48,18	15,31
6	6,48	3,90	-13,57	3,19	15,62	2,77
7	12,09	2,45	-6,88	3,53	14,40	2,12
8	4,49	12,06	4,29	2,94	12,96	4,97
9	-5,35	3,39	14,00	2,62	15,22	3,37
10	-17,94	3,43	6,85	4,27	19,58	3,88
11	8,03	1,48	-23,87	4,64	25,34	3,97
12	11,75	3,97	-2,35	2,52	12,09	4,41
13	14,17	1,07	0,77	3,77	14,71	0,55
14	-14,17	17,80	8,47	1,13	19,64	14,32
15	-7,29	3,14	-2,92	2,16	8,50	1,98
16	-4,46	2,12	-1,28	5,37	6,96	2,54
17	-14,23	1,37	5,15	5,66	16,22	0,27
18	-4,44	7,41	25,04	8,05	26,44	8,19
19	-4,64	4,54	8,69	5,67	10,11	6,90
20	1,70	0,55	10,65	1,43	10,79	1,50
21	-16,38	15,12	-3,73	7,60	17,43	16,28
22	-15,18	5,28	0,67	5,03	16,32	4,19
23	-14,26	2,92	-1,44	4,53	15,08	2,67
24	0,37	4,30	19,67	4,26	20,24	3,70
25	-2,87	2,02	15,89	3,08	16,25	3,21
26	-6,32	1,30	6,53	6,29	10,34	4,10
27	-13,35	2,34	19,73	6,56	23,95	6,50
28	-8,54	5,67	5,52	2,36	11,59	2,61
29	-23,87	1,48	2,40	1,64	24,05	1,48
30	-8,92	6,10	-8,34	12,64	16,65	8,30
31	-12,55	4,69	5,03	3,41	13,59	5,64
32	-11,81	6,54	8,67	3,72	14,72	7,39
33	-20,35	4,84	13,57	7,64	25,17	6,82
34	0,14	1,91	13,93	2,39	14,05	2,43
35	-11,83	12,76	12,76	8,25	22,42	5,56

Tabla 13: Valor medio y desviación estándar de los errores para cada eje y para el posicionamiento en cada punto para la trayectoria en espiral con 6 balizas.

Punto	Error en eje $x$ [cm]		Error en eje $y$ [cm]		Error en posición [cm]	
	Valor medio	Desviación estándar	Valor medio	Desviación estándar	Valor medio	Desviación estándar
1	14,72	3,96	-34,66	4,63	37,70	5,80
2	28,17	2,04	1,89	0,41	28,23	2,00
3	14,46	0,42	-4,48	0,41	15,14	0,52
4	28,58	0,89	-7,55	1,69	29,62	0,43
5	20,62	2,18	0,27	0,40	20,63	2,19
6	4,49	1,31	-16,67	2,53	17,28	2,78
7	11,16	0,25	-12,57	0,65	16,82	0,32
8	31,38	3,93	6,06	10,29	33,34	5,57
9	24,31	13,37	22,05	15,40	38,64	0,38
10	35,41	0,75	-1,51	12,37	37,55	0,21
11	-10,06	12,18	-9,71	0,82	16,87	7,73
12	12,46	2,81	-13,83	6,16	18,74	6,41
13	20,32	1,71	-17,21	9,41	27,37	7,19
14	20,50	2,37	-13,12	12,40	26,21	8,06
15	6,98	2,90	-8,36	3,49	10,89	4,53
16	-7,84	4,90	13,55	0,99	16,36	1,53
17	-9,14	10,74	12,87	9,25	21,19	0,99
18	1,15	5,00	11,89	1,19	12,91	1,55
19	7,23	8,44	1,84	14,09	17,32	5,02
20	8,64	0,90	10,06	12,74	17,00	7,09
21	-18,42	0,95	9,81	1,80	20,90	1,68
22	-11,46	4,08	-2,85	7,90	13,92	4,98
23	-16,19	4,32	-5,97	3,49	17,92	2,74
24	-0,90	2,27	8,36	0,46	8,72	0,20
25	8,44	1,80	13,12	9,68	16,24	8,76
26	-2,04	2,92	-7,04	1,31	7,99	0,41
27	-23,00	11,05	-10,43	1,08	25,55	10,38
28	-22,77	3,32	-15,21	4,82	27,47	5,43
29	-21,61	22,15	3,84	2,94	25,28	18,49
30	-3,15	2,54	-7,01	2,52	7,81	3,28
31	3,54	0,73	-8,27	1,20	9,07	0,81
32	-24,03	3,64	3,78	2,73	24,42	4,00
33	-18,05	0,28	12,66	9,72	23,57	5,01
34	-23,58	8,13	8,96	1,45	25,60	6,98
35	-45,79	6,95	-3,91	1,44	46,00	6,79

## 7.2. Medidas en el edificio de la Facultad de Física

A diferencia de la sección anterior, las medidas en el edificio de la Facultad de Física solo se tomaron con una trayectoria, por lo que en este caso no figurarán tablas adicionales.

### 7.2.1. Medidas con 8 balizas

Tabla 14: Valor medio y desviación estándar de los errores para cada eje y para el posicionamiento en cada punto para la trayectoria con 8 balizas.

Punto	Error en eje $x$ [cm]		Error en eje $y$ [cm]		Error en posición [cm]	
	Valor medio	Desviación estándar	Valor medio	Desviación estándar	Valor medio	Desviación estándar
1	13,45	8,07	1,60	5,37	14,19	8,73
2	3,35	1,56	-3,61	3,45	5,61	2,68
3	16,32	8,48	-7,61	15,39	23,66	8,56
4	20,98	19,12	48,91	11,56	55,45	16,02
5	-13,31	10,38	-12,74	17,65	27,43	2,56
6	5,99	3,59	-12,67	7,23	14,87	6,38
7	3,32	6,03	9,11	12,95	16,47	5,17
8	3,52	3,93	-6,62	12,97	14,68	4,96
9	-16,25	10,04	-25,59	6,59	30,90	10,41
10	-24,47	1,90	-19,24	0,99	31,18	1,10
11	-25,84	3,39	-30,54	5,22	40,26	4,32
12	-5,88	15,51	-3,81	13,57	18,03	12,18
13	-3,51	4,03	-1,50	7,28	9,15	0,28
14	-9,51	5,74	-5,20	4,55	12,61	3,50
15	-7,56	3,01	-20,64	8,19	22,13	8,34
16	-17,45	6,78	-10,03	2,65	20,25	6,93

### 7.2.2. Medidas con 6 balizas

Tabla 15: Valor medio y desviación estándar de los errores para cada eje y para el posicionamiento en cada punto para la trayectoria con 6 balizas.

Punto	Error en eje $x$ [cm]		Error en eje $y$ [cm]		Error en posición [cm]	
	Valor medio	Desviación estándar	Valor medio	Desviación estándar	Valor medio	Desviación estándar
1	18,34	4,33	0,37	4,83	19,01	4,18
2	9,73	3,13	-3,41	1,50	10,34	3,39
3	18,33	4,42	-9,37	1,70	20,77	3,85
4	32,13	12,24	-10,36	9,71	34,41	14,14
5	-3,96	1,37	-30,33	3,51	30,61	3,65
6	-3,31	4,39	1,01	13,71	14,37	3,57
7	3,37	3,28	0,45	6,27	7,84	0,49
8	8,19	4,29	0,40	4,05	9,11	4,37
9	-19,93	10,23	-28,52	6,26	35,30	10,43
10	-40,69	8,59	-15,75	4,15	44,13	6,87
11	-29,05	3,06	-10,41	3,50	30,96	3,98
12	-20,93	9,78	-12,10	0,94	24,68	8,48
13	-3,68	5,21	-15,05	5,65	16,25	5,92
14	-8,08	13,67	-7,85	4,97	16,99	7,06
15	-17,14	1,95	-18,02	0,45	24,93	1,10
16	-19,89	4,43	-30,87	22,53	39,88	16,87

### 7.2.3. Medidas con 4 balizas

Tabla 16: Valor medio y desviación estándar de los errores para cada eje y para el posicionamiento en cada punto para la trayectoria con 4 balizas.

Punto	Error en eje $x$ [cm]		Error en eje $y$ [cm]		Error en posición [cm]	
	Valor medio	Desviación estándar	Valor medio	Desviación estándar	Valor medio	Desviación estándar
1	21,40	11,67	-7,73	4,59	24,46	8,76
2	13,59	19,58	-4,59	0,68	20,69	12,71
3	3,32	3,29	2,07	4,10	5,54	3,51
4	27,84	17,32	-24,11	7,95	37,33	18,05
5	-26,22	12,02	8,31	0,21	27,83	11,26
6	10,12	2,95	39,95	9,47	41,55	8,39
7	13,84	0,85	3,10	0,11	14,19	0,85
8	20,98	3,36	0,49	2,05	21,08	3,40
9	-13,24	26,34	-11,22	2,87	28,77	13,24
10	-40,82	6,31	-16,84	4,06	44,59	4,25
11	-18,65	5,45	-19,00	3,69	27,40	1,15
12	9,70	13,38	-3,31	0,34	14,23	9,05
13	-2,64	0,39	-14,38	3,04	14,62	3,06
14	3,48	0,97	-11,19	7,47	12,32	6,52
15	2,32	5,08	-30,29	0,05	30,80	0,33
16	-16,64	4,01	-49,67	10,18	52,38	10,92

## 8. Apéndice 2

En este apéndice figura el código utilizado para construir los nodos encargados del movimiento del robot y de la recogida de datos de posicionamiento, tanto de odometría como de los sensores de UWB.

Para todo el código se preparó un archivo de base con funciones comunes. En este archivo figura el todo el código excepto la generación de las posiciones de cada trayectoria, para las que se programó una función en un archivo adicional para cada una, compilando un nodo para cada caso.

El archivo común, llamado `base.h`, contiene el siguiente código:

```
1  #include <string>
2  #include <vector>
3  #include <array>
4  #include <fstream>
5  #include <chrono>
6  #include <thread>
7  #include <ros/ros.h>
8  #include <std_msgs/Float64.h>
9  #include <move_base_msgs/MoveBaseAction.h>
10 #include <geometry_msgs/PoseWithCovarianceStamped.h>
11 #include <actionlib/client/simple_action_client.h>
12 #include <tf/transform_datatypes.h>
13 #include <tf/transform_listener.h>
14
15 typedef geometry_msgs::Pose Pose;
16 typedef geometry_msgs::PoseWithCovarianceStamped PoseCov;
17
18 const std::string ruta("/home/tb2b/catkin_ws/src/"); //Ruta en el
    ordenador del robot
19
20 const unsigned int MAX_PTS_SENSOR = 50; //Puntos a recoger de los
    sensores.
21 const unsigned int MAX_INT = 3; // Intentos en caso de que el robot
    no pueda llegar al objetivo
22 const unsigned int REP = 3; //Repeticiones de cada trayectoria.
23 const bool SALIDA_TF = false;
24
25 // Posicion del robot.
26 Pose actual_pose;
27 boost::array<double,36> actual_pose_covariance;
28
29 std::vector<Pose> path();
30 std::string get_tag_name();
31
32 std::string get_sensor_data() {
```

```

33     std::ifstream consola;
34     std::ofstream file;
35     std::string linea;
36     unsigned int num_datos = 0;
37
38     while (num_datos < MAX_PTS_SENSOR){
39
40         consola.open("/dev/ttyACM0");
41
42         std::getline(consola, linea);
43
44         if (linea.size() > 115 && linea.size() < 180) {
45             // A veces se vuelve un poco loco al leer asi que no
46             // guardaremos los datos que
47             // por estar con menos caracteres no nos sirven.
48
49             std::this_thread::sleep_for(std::chrono::milliseconds
50                 (100));
51
52             file.open(ruta + "turtle_tfg/src/datos/sensor/" + tag + ".txt",
53                 std::ios_base::app);
54             file << linea << "\n";
55             file.close();
56
57             num_datos++;
58         }
59
60         consola.close();
61     }
62
63     //TODO
64     return("0.0\n");
65 }
66
67 std::string get_date(){
68     /*
69     Devuelve fecha y hora.
70     */
71
72     auto now = std::chrono::system_clock::to_time_t(std::chrono::
73         system_clock::now());
74
75     //Fecha a texto
76     char text[50];
77     std::strftime(text, sizeof(text), "%Y-%m-%d-%Hh%Mm", std::
78         localtime(&now));
79
80

```



```

75         std::string string_return(text); // Devolvemos std::string en vez
           de char*
76         return(string_return);
77     }
78
79     void log_position(const std::string& file_name, const Pose& pos){
80         /*
81             Escribe, en el archivo file_name, la posicion del robot y de
               los sensores.
82         */
83         std::string log_pos; //Posicion a escribir en el archivo
84
85         log_pos = std::to_string(pos.position.x) + "\t" + std::to_string(
           pos.position.y) + "\t"; // Log de la posicion teorica
86
87         //Posicion real, tf
88         if (SALIDA_TF){
89             // En un principio no vamos a usar tf para conseguir la posicion
               asi que lo escondemos detras de un bool en false.
90
91             tf::TransformListener listener;
92             tf::StampedTransform trans;
93
94             bool dato_OK = true;
95             while(dato_OK){
96                 //Reintentamos hasta que damos con el topic.
97                 try{
98                     listener.lookupTransform("map", "base_link", ros::
                       Time(), trans);
99
100                     std::cout << "Posicion_tf_" << std::endl;
101
102                     std::cout << trans.getOrigin().getX() << "," << trans
                       .getOrigin().getY() << std::endl;
103
104                     double roll, pitch, yaw;
105                     trans.getBasis().getRPY(roll, pitch, yaw);
106                     std::cout << "Yaw_" << yaw << std::endl;
107                     dato_OK = false; //Salimos.
108
109                 }
110                 catch(tf::TransformException& ex) {
111                     //std::cout << "Buscando transformacion \r";
112                 }
113             }}
114
115     //amcl_pose

```

```

116
117     std::cout << "Posicion amcl" << std::endl;
118     std::cout << actual_pose.position.x << "," << actual_pose.
        position.y << std::endl;
119     std::cout << actual_pose_covariance[0] << "," <<
        actual_pose_covariance[7] << std::endl;
120
121     std::cout << " " << std::endl;
122
123     log_pos = log_pos
124         + std::to_string(actual_pose.position.x) + "\t"
125         + std::to_string(actual_pose_covariance[0]) + "\t"
126         + std::to_string(actual_pose.position.y) + "\t"
127         + std::to_string(actual_pose_covariance[7]) + "\t"
128         + get_sensor_data(); // Log de la posicion de amcl
129
130     // Al archivo
131     std::ofstream file(file_name, std::ios_base::app);
132     file << log_pos;
133     file.close();
134 }
135
136 bool move_to_goal(const Pose& pos){
137     /*
138         Mueve el robot hacia una posicion en el mapa.
139         Devuelve si el resultado ha sido exitoso.
140     */
141
142     actionlib::SimpleActionClient<move_base_msgs::MoveBaseAction> ac(
        "move_base", true);
143     while(!ac.waitForServer(ros::Duration(5.0))){
144         ROS_INFO("Iniciando server");
145     }
146
147     move_base_msgs::MoveBaseGoal goal;
148
149     goal.target_pose.header.frame_id = "map";
150     goal.target_pose.header.stamp = ros::Time::now();
151     goal.target_pose.pose = pos;
152
153     //Movimiento
154     ac.sendGoal(goal);
155     ac.waitForResult();
156     return(ac.getState() == actionlib::SimpleClientGoalState::
        SUCCEEDED);
157 }
158

```

```

159
160 void update_position(const PoseCov& position){
161     actual_pose = position.pose.pose;
162     actual_pose_covariance = position.pose.covariance;
163 }
164
165 int main(int argc, char** argv){
166
167     ros::init(argc, argv, "map_navigation");
168     ros::NodeHandle n;
169     ros::Subscriber sub = n.subscribe("amcl_pose",500,
170         update_position);
171     bool suc;
172
173     std::string date = get_date();
174
175     std::string file = ruta + "/turtle_tfg/datos/" + get_date() + ".
176         txt"; // Nombre del archivo para guardar los puntos.
177
178     auto posiciones = path(); // Vector con las posiciones de la
179         trayectoria.
180
181     for (int k=1; k <= REP; k++){
182         file = ruta + "turtle_tfg/src/datos/"
183             + get_tag_name() + "-" + date + "-" + std::to_string(k
184                 ) + ".txt";
185
186         for (int i=0; i < posiciones.size(); i++){
187             suc = move_to_goal(posiciones[i]);
188             if (!suc){
189                 ROS_INFO("El robot no ha llegado a su destino ,
190                     reintentando");
191                 for (int j=0; j < MAX_INT; j++){
192                     suc = move_to_goal(posiciones[i]);
193                     if (suc) break;
194                 }
195             }
196             ros::spinOnce();
197
198             std::string tag = get_tag_name() + std::to_string(i) + date +
199                 "-" + std::to_string(k); //Tag de cada punto.
200             log_position(file, posiciones[i], tag);
201         };
202     };
203
204     ROS_INFO("Final de la trayectoria.");

```

```

200         return 0;
201     }

```

Junto con el archivo base se escribió un archivo adicional para cada una de las trayectorias con la definición de la función `path()`, que proporciona los puntos a recorrer por el robot.

En el caso de la trayectoria en espiral tal y como se indica en la Figura 13b del laboratorio dicho archivo fue

```

1  #include <base.h>
2  void log_position(const Pose& pos);
3  bool move_to_goal(const Pose& pos);
4  void update_position(const PoseCov& position);
5
6  std::string get_tag_name(){
7      return("ESP");
8  }
9
10 std::vector<Pose> path() {
11     std::vector<Pose> ruta;
12     Pose position;
13
14     const int MAX = 2;
15
16     for(int i=0; i <= MAX; i++){
17
18         int x = i;
19         int y = i;
20
21         position.position.x = x;
22         position.position.y = y;
23
24         position.orientation = tf::createQuaternionMsgFromYaw(0); //0
                                grados
25
26         ruta.push_back(position);
27
28         // Trayecto hacia abajo
29         for(int j=i; j>=1; j--){
30             y -= 1;
31
32             position.position.x = x;
33             position.position.y = y;
34
35             position.orientation = tf::createQuaternionMsgFromYaw(0);
                                //0 grados
36

```

```

37         ruta.push_back(position);
38     }
39
40     // Trayecto hacia la izquierda
41     for(int j=i; j>=i; j--){
42         x -= 1;
43
44         position.position.x = x;
45         position.position.y = y;
46
47         position.orientation = tf::createQuaternionMsgFromYaw(0);
48         //0 grados
49
50         ruta.push_back(position);
51     }
52
53     // Trayecto hacia arriba
54     for(int j=i; j>=i; j--){
55         y += 1;
56
57         position.position.x = x;
58         position.position.y = y;
59
60         position.orientation = tf::createQuaternionMsgFromYaw(0);
61         //0 grados
62
63         ruta.push_back(position);
64     }
65
66     // Trayecto hacia la derecha
67     for(int j=i; j>=i+1; j--){
68         x += 1;
69
70         position.position.x = x;
71         position.position.y = y;
72
73         position.orientation = tf::createQuaternionMsgFromYaw(0);
74         //0 grados
75
76         ruta.push_back(position);
77     }
78
79     return(ruta);
80 }

```

En el caso de la trayectoria en vertical que se indica en la Figura 13c archivo contenía el siguiente código:

```

1  #include "base.h"
2
3  std::string get_tag_name(){
4      return("VERT");
5  }
6
7  std::vector<Pose> path(){
8      std::vector<Pose> ruta;
9      Pose position;
10
11     const int ANCHO = 2;
12     const int ALTO = 3;
13
14     bool orient = false; //Comenzamos hacia abajo.
15
16     for (int i=ANCHO; i >= -ANCHO; i--){
17         for (int j=ALTO; j >= -ALTO; j--){
18             if (orient){
19                 position.position.x = i;
20                 position.position.y = -j;
21
22                 position.orientation = tf::createQuaternionMsgFromYaw(0);
23                 //0 grados
24             } else {
25                 position.position.x = i;
26                 position.position.y = j;
27
28                 position.orientation = tf::createQuaternionMsgFromYaw(0);
29                 //0 grados
30             }
31             ruta.push_back(position);
32         }
33     }
34     orient = !orient;
35     return(ruta);
36 }

```

La tercera y última trayectoria hacía que el robot recorriese todo el laboratorio de forma aleatoria. Se planteó que en cada trayectoria el orden cambiase, como se indica en el código que la genera:

```

1  #include "base.h"
2  // Trayectoria que recorre el laboratorio de forma aleatoria.
3
4  std::string get_tag_name(){
5      return("RAND");
6  }
7

```

```

8  std::vector<Pose> path(){
9      std::vector<Pose> ruta;
10     Pose position;
11
12     const unsigned int ANCHO = 2;
13     const unsigned int ALTO = 3;
14
15     //Distribuciones pseudo-aleatorias
16     std::random_device rd;
17     std::mt19937 gen(rd());
18
19     // Generamos todos los datos posibles.
20     for (int i = -ANCHO; i <= ANCHO; i++){
21         for (int j = -ALTO; j <= ALTO; j++){
22             position.position.x = i;
23             position.position.y = j;
24             position.orientation = tf::createQuaternionMsgFromYaw(0);
25
26             ruta.push_back(position);
27         }
28     }
29
30     // Mezcla aleatoria del vector.
31     std::shuffle(ruta.begin(), ruta.end(), g);
32
33     return(ruta);
34 }

```

La última trayectoria fue la utilizada en el edificio de la Facultad de Física, con el siguiente código:

```

1  #include <base.h>
2
3  void log_position(const Pose& pos);
4  bool move_to_goal(const Pose& pos);
5  void update_position(const PoseCov& position);
6
7  std::string get_tag_name(){
8      return("FIS");
9  }
10
11  std::vector<Pose> path() {
12      std::vector<Pose> ruta;
13      Pose position;
14
15      float min_x = 0.7;
16      float min_y = -0.5;
17

```

```

18     float max_x = 10.0;
19     float max_y = 10.0;
20
21     const int MAX_PTS_TRAY = 3;
22
23     //Primera parte
24     for (int n=0; n <= MAX_PTS_TRAY; n++){
25         position.position.x = min_x + (max_x - min_x)*n/(MAX_PTS_TRAY
26             + 1);
27         position.position.y = min_y ;
28
29         position.orientation = tf::createQuaternionMsgFromYaw(0); //0
30         grados
31         ruta.push_back(position);
32     }
33
34     //Segunda parte
35     for (int n=0; n <= MAX_PTS_TRAY; n++){
36         position.position.x = max_x ;
37         position.position.y = min_y + (max_y - min_y)*n/(MAX_PTS_TRAY
38             + 1);
39
40         position.orientation = tf::createQuaternionMsgFromYaw(0); //0
41         grados
42         ruta.push_back(position);
43     }
44
45     //Tercera parte
46     for (int n=0; n <= MAX_PTS_TRAY; n++){
47         position.position.x = max_x - (max_x - min_x)*n/(MAX_PTS_TRAY
48             + 1);
49         position.position.y = max_y ;
50
51         position.orientation = tf::createQuaternionMsgFromYaw(0); //0
52         grados
53         ruta.push_back(position);
54     }
55
56     //Cuarta parte
57     for (int n=0; n <= MAX_PTS_TRAY; n++){
58         position.position.x = min_x ;
59         position.position.y = max_y - (max_y - min_y)*n/(MAX_PTS_TRAY
60             + 1);
61
62         position.orientation = tf::createQuaternionMsgFromYaw(0); //0
63         grados

```



```
57         ruta.push_back(position);
58     }
59
60     return(ruta);
61 }
```