

Trabajo Fin de Grado

Ingeniería de las Tecnologías de Telecomunicación

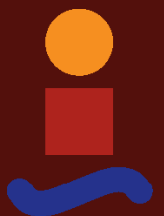
Mitigación de los errores por obstrucción de la línea de visión directa en un sistema de localización

Autor: Carmelo García García

Tutora: Eva María Arias de Reyna Domínguez

Dep. Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2019



Trabajo Fin de Grado
Ingeniería de las Tecnologías de Telecomunicación

Mitigación de los errores por obstrucción de la línea de visión directa en un sistema de localización

Autor:

Carmelo García García

Tutora:

Eva María Arias de Reyna Domínguez

Profesora Titular

Dep. Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2019

Trabajo Fin de Grado: Mitigación de los errores por obstrucción de la línea de visión directa en un sistema de localización

Autor: Carmelo García García

Tutora: Eva María Arias de Reyna Domínguez

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Agradecimientos

Gracias, en primer lugar, a Dña. Eva María Arias de Reyna Domínguez por otorgarme la oportunidad de llevar a cabo este proyecto y por su constante ayuda durante la realización del mismo.

A mis padres, porque la razón por la que he llegado hasta aquí ha sido su infinito apoyo.

A mis hermanos, por ser mis modelos a seguir y por su ayuda cada vez que me he sentido perdido.

A todos y cada uno de mis amigos del grado, especialmente a Álvaro y Omar, pues habéis hecho de estos años los mejores de mi vida.

A todas las personas que compartieron mi experiencia en Swansea, gracias por hacerme ver el mundo desde otro punto de vista.

Gracias a todos.

*Carmelo García García
Sevilla, 2019*

Resumen

La tecnología UWB (*Ultra Wide-Band*) tiene su origen en la década de los sesenta para uso militar, y hoy en día aún se encuentra en desarrollo. Sin embargo, progresivamente se está haciendo hueco en el mercado, llegando a ser actualmente una de las tecnologías más prometedoras para la aplicación de localización en interiores gracias a su precisión a la hora de estimar distancias incluso con obstáculos, su amplio ancho de banda y su bajo consumo y coste.

Conforme esta tecnología ha ido mejorando, se fue regulando y estandarizando. Primeramente se reguló, de forma que se pudiera permitir la emisión de señales UWB sin licencia bajo ciertas condiciones (fue entonces cuando se comenzaron a crear compañías). Como también es deseable la estandarización, se han creado estándares como el IEEE 802.15.4a.

El objetivo de este proyecto es proporcionar información acerca de cómo funciona esta tecnología y qué función tiene en cuanto a los sistemas de localización en interiores, para luego evaluar varios algoritmos que reduzcan los errores provocados por la obstrucción de línea de visión directa en una serie de medidas reales, obtenidas gracias al kit EVK1000 de la compañía *DecaWave*. Por último, se realizará un análisis con los resultados obtenidos, incluyendo la indicación de posibles proyectos futuros.

Abstract

The UWB (*Ultra Wide-Band*) technology has its origin in the decade of the sixties for military use, and nowadays it is still in development. However, its presence in the commercial sector is increasing progressively, becoming one of the most promising technologies for indoor localization applications, thanks to its accuracy when estimating distances with obstacles, its wide bandwidth and its low consumption and cost.

As this technology has been improving, it was regulated and standardized. Firstly it had its regulation, so the UWB signals emission was allowed without license under some conditions (and so the companies started emerging). As standarization is also wanted, standards such as IEEE 802.15.4a have been created.

The aim of this project is to provide information about how this technology works and its function in terms of indoor location systems, and to develop some algorithms to reduce the errors caused by direct line of sight obstruction in a series of real measurements, obtained thanks to the EVK1000 kit from the *DecaWave* company. Finally, an analysis will be carried out with the obtained results, including some possible future projects indications.

Índice

<i>Resumen</i>	III
<i>Abstract</i>	V
1 Introducción	1
1.1 Objetivo	1
1.2 Estructura	2
2 Fundamentos teóricos	3
2.1 UWB	3
2.1.1 Ancho de banda	3
2.1.2 Densidad espectral de potencia	4
2.1.3 Ventajas y desventajas	5
2.1.4 Aplicaciones	6
2.2 Sistemas de localización en interiores	6
2.2.1 RTLS	6
2.2.2 Tipos de medidas en un algoritmo de localización	7
2.2.3 Técnica de localización de Mínimos Cuadrados Ponderados	8
3 Algoritmos empleados	11
3.1 Tipos de identificación	11
3.1.1 Sin identificar	11
3.1.2 Conocimiento perfecto de la condición LOS/NLOS	11
3.1.3 Varianza de las medidas de distancia	11
3.1.4 Algoritmo basado en estadísticos de la señal [CGWI07]	11
3.2 Tipos de mitigación en el algoritmo WLS	12
3.2.1 Sin mitigar	12
3.2.2 <i>Identify and discard</i>	12
3.2.3 Inversa de la varianza	12
3.2.4 β_{LOS} igual a uno y β_{NLOS} distinto de cero	12
3.3 Algoritmos empleados	13
3.3.1 Algoritmo sin identificación	13
3.3.2 Algoritmo con conocimiento perfecto de la condición de visión	13
3.3.3 Algoritmo basado en la varianza de las medidas	13
3.3.4 Algoritmo basado en estadísticos de la señal recibida	13
4 Desarrollo práctico	15

4.1	Entorno de trabajo	15
4.2	Resultados para los algoritmos de referencia	17
4.3	Resultados para el algoritmo basado en la varianza de las medidas	21
4.4	Resultados para el algoritmo basado en estadísticos de la señal recibida	22
4.5	Gráfica RMSE	24
5	Conclusiones	27
	Apéndice A Configuración del software	29
	Apéndice B Códigos de Matlab	33
B.1	Algoritmo sin identificación	33
B.2	Algoritmo con conocimiento perfecto de la condición de visión	35
B.3	Algoritmo basado en la varianza de las medidas	38
B.4	Algoritmo basado en estadísticos de la señal recibida	41
B.5	Ejecución de algoritmos	44
B.6	Histograma de comparación	47
	Apéndice C Ficheros de datos	49
C.1	Posiciones reales del nodo móvil	49
C.2	Posiciones de los nodos ancla	50
C.3	Distancias medidas por cada nodo ancla	51
C.4	Desviación típica	52
C.5	Condición de línea de visión	53
	<i>Índice de Figuras</i>	57
	<i>Índice de Tablas</i>	59
	<i>Índice de Códigos</i>	61
	<i>Bibliografía</i>	63

1 Introducción

Actualmente, la sociedad se encuentra en un auge tecnológico donde la ciencia no para de mejorar nuestra calidad de vida, incluso cuando ya creemos que lo tenemos todo. Un claro ejemplo lo podemos encontrar en la geolocalización, pues a día de hoy gracias a los sistemas GNSS (*Global Navigation Satellite Systems*), como GPS (*Global Positioning System*) o *Galileo*, podemos saber dónde estamos, qué camino tomar para ir de la forma más rápida a un destino concreto, etc.

Esta tecnología está incorporada en todo tipo de dispositivos inteligentes, desde automóviles hasta teléfonos móviles, proporcionando una precisión de varios metros de diferencia respecto a nuestra posición real. Esta precisión está muy bien cuando, por ejemplo, queremos desplazarnos en coche de una ciudad a otra y queremos saber qué carretera tomar. El problema llega cuando nos encontramos en el interior de unas oficinas y queremos ir a un despacho en concreto, pues la degradación de la señal en interiores hace que no se nos localice de forma precisa y por tanto obtengamos indicaciones erróneas.

Existen diferentes soluciones a esta situación, como puede ser el uso de la tecnología Wi-Fi, pero siguen existiendo numerosas limitaciones en cuanto al coste, alcance e interferencias con otras señales.

Imaginemos que trabajamos en un hospital, y queremos informarnos acerca de las constantes vitales de un paciente gracias a un dispositivo médico inalámbrico que tiene implantado. Si hubiera otros pacientes con otros dispositivos implantados cerca, podríamos obtener una información errónea, pues debido a las estructuras metálicas de los equipos médicos y el gran número de interferencias que pueden existir, la precisión de la localización del paciente es muy baja. Es por ello que necesitamos una solución que soporte estos inconvenientes aportando una precisión de centímetros.

Otro ejemplo con la misma necesidad de solución existe en los entornos industriales, donde se quiere tener localizada la instrumentación u otros elementos, como productos de venta en grandes almacenes. Al haber una gran densidad de items a localizar, se requiere una precisión de localización de centímetros.

UWB (*Ultra Wide-Band*) es una posible solución, pues se ha demostrado que puede llegar a proporcionar una precisión de escasos centímetros. Esta tecnología está aún en desarrollo, pero su bajo coste y su alta resistencia a interferencias hace de esta una alternativa muy atractiva para la localización en interiores.

1.1 Objetivo

El objetivo de este proyecto es la implementación y estudio de varios algoritmos de localización en interiores con tecnología UWB que mitiguen los errores provocados por los efectos causados al existir obstrucción de línea de visión directa. Para ello, contamos con el kit EVK1000 de la

compañía *DecaWave*, que contiene dos transmisores UWB (uno de ellos lo usaremos como nodo móvil y otro como nodo ancla) y el software *DecaRanging* proporcionado por la misma compañía.

1.2 Estructura

El presente documento está compuesto por los siguientes capítulos:

- **Introducción:** Es el capítulo en el que nos encontramos, donde se muestra una breve información acerca del proyecto, junto con su motivación y estructura.
- **Fundamentos teóricos:** Capítulo donde se explica teóricamente qué es la tecnología UWB, cómo funciona y cuales son sus principales ventajas, desventajas y aplicaciones en la actualidad. Posteriormente se centrará en cuál es el funcionamiento de un sistema de localización en interiores y de la técnica de localización que usaremos.
- **Algoritmos empleados:** En este tercer capítulo se comentarán los tres algoritmos utilizados para la mitigación de errores por condición NLOS (*Non-Line of Sight*, Sin Línea de Visión) en nuestro sistema de localización.
- **Desarrollo práctico:** Parte experimental del proyecto donde se analizarán los resultados obtenidos según las medidas y algoritmo utilizados.
- **Conclusiones:** Capítulo que se centrará en una opinión final acerca de los resultados obtenidos en el capítulo anterior.
- **Apéndice A. Configuración del software:** Breve información acerca de cómo se han configurado los parámetros a la hora de utilizar el software *DecaRanging*.
- **Apéndice B. Códigos de Matlab:** Contiene los diferentes códigos utilizados para la implementación de los algoritmos usados en el proyecto.
- **Apéndice C. Ficheros de datos:** Último capítulo que muestra los datos obtenidos durante la toma de medidas en el Departamento con la estructura necesaria para poder importarlos en Matlab.

2 Fundamentos teóricos

La tecnología UWB (*Ultra Wide-Band*) actualmente se encuentra en un proceso de intensa investigación para proporcionar todas las mejoras en telecomunicación que promete. Entre otras innovaciones, destaca la localización en interiores con el uso de esta tecnología, y es por ello que en las próximas páginas veremos las principales características de UWB y posteriormente su influencia en la localización en interiores.

2.1 UWB

Se han usado las siguientes referencias para el desarrollo de esta sección: [AGI] [CC01] [dlCal17] [dEco] [dRoj18] [INT] [Mad19] [MCAW18] [MGM+10]

Hoy en día, Ultra Wide-Band es una tecnología de transmisión vía radio conocida por sus aplicaciones en redes inalámbricas de área personal (*WPAN, Wireless Personal Area Network*), como pueden ser transferencia de archivos, conexión de periféricos de ordenador, etc. hasta sistemas radar y de localización en tiempo real. No obstante, esta tecnología tiene su origen en el ámbito militar para aplicaciones radar y comunicaciones alrededor de los años 60.

Ultra Wide-Band, antes conocido como radio basada en impulsos (*IR, Impulse Radio*), se basa en la transmisión de una secuencia de pulsos electromagnéticos, en los que cada bit de información está representado por uno o más de estos pulsos. Los pulsos son muy estrechos, del orden de un nanosegundo, con diferentes tasas de repetición según su aplicación (alta para comunicaciones, baja para escáneres o radar).

2.1.1 Ancho de banda

Tras varias definiciones a lo largo de los últimos años, tanto la Comisión Federal de Comunicaciones (*FCC, Federal Communications Commission*) de Estados Unidos como el Sector de Radiocomunicación de la Unión Internacional de Telecomunicaciones (*ITU-R, International Telecommunication Union Radiocommunication Sector*) establecen la definición de UWB como la transmisión desde una antena para la cual el ancho de banda de la señal emitida, medido a un nivel de -10 dB del valor máximo de la densidad espectral de potencia (*PSD, Power Spectral Density*), es superior al menor de estos dos valores:

- Ancho de banda absoluto $B > 500 \text{ MHz}$:

$$B = f_H - f_L > 500 \text{ MHz} \quad (2.1)$$

donde f_H y f_L son las frecuencias superior e inferior, respectivamente, a -10 dB del valor máximo de la densidad espectral de potencia (*PSD*), tal y como se muestra en la figura 2.1.

- Ancho de banda relativo $B_{rel} > 20\%$. Para ello, sabiendo que el valor de la frecuencia central f_c es:

$$f_c = \frac{f_H + f_L}{2} \quad (2.2)$$

Y que el ancho de banda relativo B_{rel} es:

$$B_{rel} = \frac{B}{f_c} \quad (2.3)$$

Sustituimos y tendremos como resultado final:

$$B_{rel} = \frac{2(f_H - f_L)}{f_H + f_L} > 20\% \quad (2.4)$$

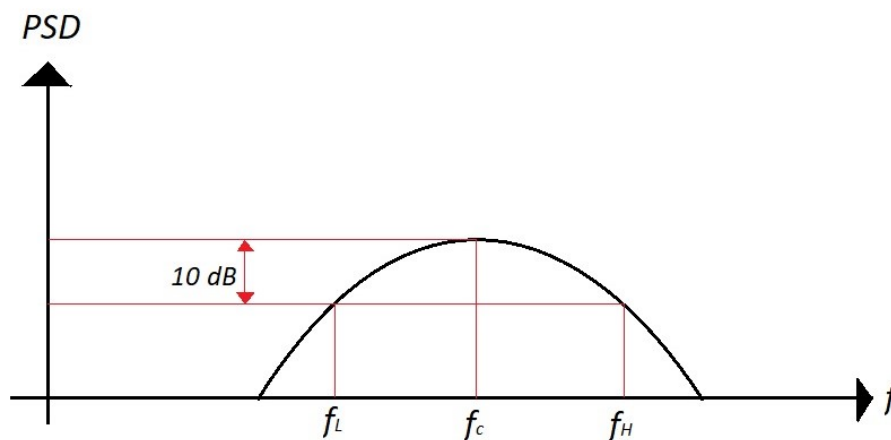


Figura 2.1 Señal Ultra Wide-Band en el dominio de la frecuencia.

2.1.2 Densidad espectral de potencia

A primera vista un gran ancho de banda puede conllevar grandes dificultades a la hora de ocupar un canal, por lo que esta tecnología se ve obligada a coexistir con otras sin provocar interferencias. Por tanto, la potencia es el principal parámetro regulado según la porción del espectro en la que se trabaja.

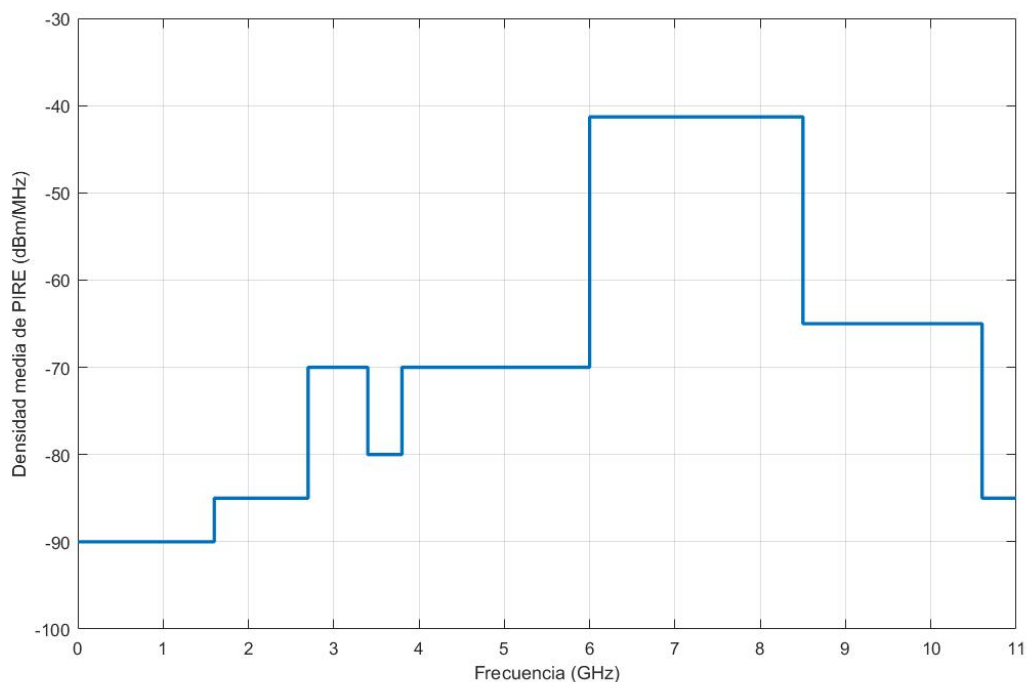
En Europa, la ITU-R regula la densidad media de PIRE (*Potencia Isotrópica Radiada Equivalente*) con valor máximo de -41.3 dBm/MHz en el rango de frecuencias que va desde 6 GHz hasta 8.5 GHz, un total de 2.5 GHz, existiendo una restricción mayor en el resto del espectro.

En Estados Unidos, en cambio, la densidad media de PIRE está regulada por la FCC no pudiendo ser superior a -41.3 dBm/MHz en la banda que comienza en 3.1 GHz y acaba en 10.6 GHz, abarcando un total de 7.5 GHz, siendo este espectro el menos restrictivo de este país en cuanto a tecnología UWB, y a la vez mucho más amplio que en Europa.

Concretamente en España, el uso de UWB viene recogido en la nota *UN-137* del CNAF (Cuadro Nacional de Atribución de Frecuencias) [dEco], según la siguiente tabla, mostrada de una forma más representativa en la figura 2.2

Tabla 2.1 Máscara espectral del CNAF para el uso de UWB en España .

Banda de frecuencias (GHz)	Máxima densidad media de PIRE (dBm/MHz)
Frecuencias < 1.6	-90
1.6 a 2.7	-85
2.7 a 3.4	-70
3.4 a 3.8	-80
3.8 a 6	-70
6 a 8.5	-41.3
8.5 a 10.6	-65
Frecuencias > 10.6	-85

**Figura 2.2** Máscara espectral del CNAF para el uso de UWB en España.

2.1.3 Ventajas y desventajas

Ventajas

- Gran capacidad de transferencia de datos a altas velocidades, del orden de 1225 Mbps en W-USB (*Wireless Universal Serial Bus*).
- Gracias a su muy baja densidad espectral de potencia, no causa ningún tipo de interferencia y por tanto reutiliza bandas de frecuencias asignadas a otros servicios, aportando así un bajo coste y consumo.
- Inmunidad alta frente al efecto multirayecto.
- Precisión de centímetros en cuanto a su uso en sistemas de localización.
- Escasa probabilidad de interceptación por otros equipos gracias a su baja densidad espectral de potencia, camuflándose por debajo del ruido.

- Capacidad de atravesar obstáculos debido a su amplio rango de frecuencias.

Desventajas

- La principal desventaja es el alcance de la señal, alcanzando un máximo de decenas de metros debido a sus limitaciones en potencia.
- La precisión en la medida de distancias se ve también afectada por la presencia de obstáculos que atenúan o incluso obstruyen completamente la señal durante la localización en interiores, como pueden ser los objetos metálicos. Este proyecto se centra principalmente en la mitigación de los errores producidos por estos obstáculos.

2.1.4 Aplicaciones

A pesar de que esta tecnología fuera desarrollada en los 60 en el ámbito militar, no ha sido hasta varias décadas después que ha comenzado a utilizarse, esto es porque a partir de 2002 la FCC en Estados Unidos permitió la emisión de señales UWB sin licencia, cumpliendo con la máscara de emisión y suponiendo un gran avance en las redes WPAN (*Wireless Personal Area Network*, Red Inalámbrica de Área Personal) y en la precisión de sistemas de localización. No obstante, esta tecnología sigue investigándose para poder proporcionar la mayor cantidad de servicios que se cree que puede ofrecer.

Hoy en día, UWB tiene un precio relativamente alto debido a la escasa competencia, pero al estar haciéndose hueco en el mercado, esto implicará una mayor competencia y desarrollo de sus capacidades. Entre sus principales aplicaciones, podemos destacar:

- Redes de sensores.
- Redes WPAN: Transferencia de datos a alta velocidad desde terminales móviles, videocámaras, etc., conexión de periféricos, impresión digital.
- Localización en interiores. Pese a no tener gran alcance, proporciona una gran precisión en tiempo real, pues gracias a su baja densidad espectral de potencia se minimizan las interferencias sobre otras emisiones radioeléctricas, y las interferencias sobre UWB provocadas por otras emisiones radioeléctricas se minimizan por el gran ancho de banda y técnicas utilizadas, por ejemplo *time hopping*.
- Aplicaciones radar en el ámbito militar y gubernamental: detección de intrusos, comunicaciones a bordo de vehículos militares con alta seguridad frente a intercepciones, etc.

2.2 Sistemas de localización en interiores

En esta segunda parte del capítulo de Fundamentos Teóricos, vamos a hablar acerca de qué es un sistema de localización y de qué técnica usaremos durante el proyecto para la estimación de posiciones.

2.2.1 RTLS

Se han usado las siguientes referencias para el desarrollo de esta sección: [Ema13] [FF17] [Gar08] [GS04] [GC09] [GCWI07] [MCAW18] [PKG16] [Per10] [TPHH19] [Vic] [ZM19]

Un sistema de localización en tiempo real (*RTLS*, *Real-Time Locating System*), como su propio nombre indica, es un sistema que mediante un algoritmo de rastreo e identificación estima la localización de personas u objetos en tiempo real, típicamente en áreas cerradas o interior de edificios. El sistema usualmente está formado por un número de nodos ancla tomados como puntos

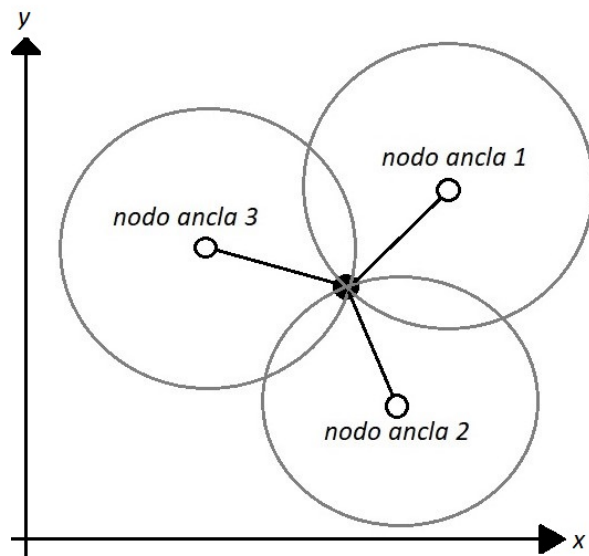


Figura 2.3 Ejemplo de un sistema de localización con tres nodos ancla y un nodo móvil.

de referencia que recibirán señales inalámbricas del objeto o persona, el cual poseerá el nodo móvil, y estimará su posición en base a la información recibida.

Podemos observar un ejemplo en la figura 2.3, donde existen tres nodos ancla representados con un círculo blanco y un nodo móvil representado con un círculo negro. Cada nodo ancla estima una distancia respecto al nodo móvil, pero no sabe la dirección en la que está; por tanto, se genera un sistema de ecuaciones con la siguiente estructura:

$$(x - x_i)^2 + (y - y_i)^2 = \hat{d}_i^2, \quad i = 1, 2, \dots, N \quad (2.5)$$

donde x e y son las coordenadas del nodo móvil, x_i e y_i las coordenadas del nodo ancla i , \hat{d}_i la distancia estimada medida por el nodo ancla i y siendo N el número de nodos ancla (en el caso del ejemplo sería igual a 3). Este sería un ejemplo ideal, pues las anclas han calculado la distancia exacta al móvil, pero un error en dichas medidas provocaría un sistema de ecuaciones donde no exista una única solución.

Debido a al gran número de incógnitas que existen en un lugar de interior (como por ejemplo en un edificio donde puede existir el paso concurrencido de personas, materiales metálicos que sean difíciles de atravesar para la señal, interferencias con otros sistemas de radio, etc.) existen diferentes tecnologías aplicadas en este ámbito, como pueden ser el empleo de infrarrojos (*IR*), ultrasonidos, Wi-Fi (*Wireless Fidelity*), RFID (*Radio Frequency IDentification*), BLE (*Bluetooth Low Energy*) o UWB (*Ultra Wide-Band*).

La localización de un nodo móvil puede estimarse mediante diferentes técnicas, una de ellas y en la que nos centraremos se basa en la medida de distancias.

2.2.2 Tipos de medidas en un algoritmo de localización

Para estimar la posición del nodo móvil, debemos de recibir primeramente las medidas realizadas por cada nodo ancla. Estas medidas pueden ser, por ejemplo, una distancia o un ángulo, calculadas mediante los métodos que se describen a continuación.

Tiempo de llegada

El primer método es muy sencillo y robusto, basado en la medición del tiempo de llegada (*ToA*, *Time of Arrival*), es decir, el tiempo que una señal emitida por el nodo ancla tarda en llegar al nodo

móvil a una velocidad conocida.

$$d_{est} = v \cdot ToA \quad (2.6)$$

donde d_{est} es la distancia estimada en metros, v la velocidad de la señal en metros por segundo y ToA el tiempo de llegada medido en segundos.

Este método es el más usado en UWB, pues aporta una gran precisión gracias a su amplio ancho de banda (y por tanto, su alta resolución temporal) y su resistencia al efecto multirayecto. Sin embargo, cuando la condición de la medida es NLOS y no existe un camino directo, la única forma es atravesando materiales y es ahí cuando se produce un efecto de retraso en el tiempo de llegada, pues la velocidad de propagación es menor dentro del obstáculo, de forma que la ecuación (2.6) dará una estimación de distancia mayor de la real.

Además, puede darse el caso en el que el rayo directo esté completamente obstruido. Entonces, el primer rayo secundario que llegue se interpretará como el directo, dando lugar a grandes errores. Los métodos que se evaluarán en este proyecto están enfocados sobre todo a mitigar este tipo de error.

Diferencia de tiempos de llegada

A diferencia del método ToA, este método se basa en la diferencia de tiempos de llegada (TDoA, *Time Difference of Arrival*) respecto de dos nodos ancla, precisando de una sincronización muy estricta.

De esta forma, el nodo móvil no necesita una conexión individual con cada nodo ancla, sino que envía un mensaje *blink* para ser localizado por los nodos ancla sincronizados. [Sew]

Ángulo de llegada

En este caso, el parámetro a medir es el ángulo de llegada (*AoA*, *Angle of Arrival*, también llamado *DoA*, *Direction of Arrival*), que es el ángulo formado por la dirección de propagación de la señal y una dirección de referencia.

Fuerza de la señal recibida

El último método a considerar para obtener la distancia es a través de la fuerza de la señal recibida (*RSS*, *Received Signal Strength*), basándose en la atenuación de la misma.

Durante la realización de este proyecto nos vamos a centrar en una familia de métodos basados en medidas de distancia, obtenidas mediante estimaciones de *ToA* con el kit EVK1000 de *DecaWave* [Dec]. A continuación, se va a explicar la técnica de localización utilizada durante el proyecto.

2.2.3 Técnica de localización de Mínimos Cuadrados Ponderados

Esta técnica es conocida por las siglas WLS (*Weighted Least Squares*, también llamada *Weighted Linear Regression*), y consiste en una especialización de la técnica de Mínimos Cuadrados.

Supongamos que tenemos un total de N nodos ancla, siendo \mathbf{x} las coordenadas de la posición real del nodo móvil, \mathbf{x}_i las coordenadas de la posición del nodo ancla i , \hat{d}_i la distancia estimada por el nodo ancla i y β_i el peso que le daremos a la medida del nodo ancla i , podemos estimar la posición del nodo móvil ($\hat{\mathbf{x}}$) como el argumento para que el sumatorio del cuadrado de los residuos multiplicados por sus pesos sea mínimo. En nuestro caso, el residuo es la diferencia entre la distancia estimada por el nodo ancla y la distancia real entre el nodo móvil y ancla (calculada a partir de la diferencia entre las posiciones de ambos).

Por tanto, la estimación de la posición de el nodo móvil mediante la técnica WLS [GCWI07] puede expresarse según la ecuación:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \left\{ \sum_{i=1}^N \beta_i (\hat{d}_i - \|\mathbf{x} - \mathbf{x}_i\|)^2 \right\} \quad (2.7)$$

Uno de los principales inconvenientes de esta técnica es la minimización de esta ecuación, pues al tener un sistema de ecuaciones no lineal es necesario un método que evite converger en un mínimo local en lugar del mínimo global. Una solución posible, que de hecho aplicaremos durante el proyecto, es evaluar la función de la ecuación (2.7) en un set de coordenadas posibles (x,y) y hallar el par (x,y) para que el valor de la función sea mínimo; cuantas más posiciones probemos, mayor será la precisión que aporte esta solución pero más tiempo de cálculo se necesitará.

3 Algoritmos empleados

En este capítulo estudiaremos qué tipos de identificación de la condición LOS/NLOS y de mitigación de las medidas identificadas como NLOS hemos utilizado en el algoritmo WLS, para posteriormente explicar los diferentes algoritmos que se han implementado y aplicado durante el proyecto para la mitigación de errores, teniendo como objetivo final destacar qué algoritmo proporcionará la mayor precisión a la hora de determinar la posición (x,y) de un nodo móvil con ubicación desconocida.

3.1 Tipos de identificación

3.1.1 Sin identificar

Como su propio nombre indica, este no es un algoritmo de identificación, sino un caso con el que comparar los resultados obtenidos al aplicar nuestros algoritmos, pues no diferenciaremos la condición LOS de la condición NLOS.

3.1.2 Conocimiento perfecto de la condición LOS/NLOS

Este realmente tampoco es un algoritmo de identificación, sino un algoritmo ideal que usaremos para comparar con otros algoritmos de identificación realistas. Concretamente, se basa en el conocimiento previo de la condición LOS/NLOS de la medida de cada nodo ancla.

3.1.3 Varianza de las medidas de distancia

En este caso, para cada medida de distancia realizada con el kit EVK1000, obtendremos además la desviación típica de la misma y la elevaremos al cuadrado para obtener su varianza. Este tipo de identificación no da como resultado una identificación LOS/NLOS, pero lo utilizamos para ponderar la fiabilidad de las medidas.

Cabe destacar que para utilizar el valor de desviación típica proporcionado automáticamente por el kit EVK 1000, es necesario realizar al menos 50 medidas de distancia en un mismo par de posiciones nodo ancla-móvil.

3.1.4 Algoritmo basado en estadísticos de la señal [CGWI07]

En este trabajo no implementamos el algoritmo de identificación propiamente dicho, sino que dicho algoritmo se encapsula en los valores de probabilidad de correcta identificación previamente estimados por [CGWI07].

Estas probabilidades han sido calculadas con tests de razón de verosimilitud (*Likelihood-ratio tests*), tal y como se explica en la sección 3.3 de [CGWI07], para diferentes modelos de canal (LOS/NLOS en zona residencial, interior de oficinas, exterior e industrial). En nuestro caso, tomaremos las probabilidades correspondientes a los modelos de canal LOS/NLOS en interior de oficinas.

3.2 Tipos de mitigación en el algoritmo WLS

En esta sección, explicaremos los distintos tipos de mitigación de las medidas identificadas como NLOS en el algoritmo WLS mediante la sustitución de β_i en la ecuación (2.7) por un valor concreto.

3.2.1 Sin mitigar

En este primer caso, estableceremos todos los valores de β_i a 1. Este no es un algoritmo de mitigación, sino un algoritmo con el que comparar los demás.

3.2.2 Identify and discard

Utiliza la técnica de identificar y descartar, es decir, asignaremos el valor de β_i a 1 cuando la condición del canal sea LOS y 0 cuando esta sea NLOS (por lo que, al multiplicar en la ecuación (2.7) por 0, las medidas NLOS quedan descartadas) [GCWI07], tal y como se muestra a continuación:

$$\beta_i = \begin{cases} 1 & \text{si la medida es LOS} \\ 0 & \text{si la medida es NLOS} \end{cases} \quad (3.1)$$

Pese a que aparentemente esta técnica es la ideal, en el caso en el que no haya al menos tres nodos ancla con visión directa, existirá ambigüedad en la función a evaluar, provocando un error considerable. El motivo es que en dos dimensiones cada medida de distancia determina una circunferencia centrada en el ancla correspondiente, y harán falta 3 circunferencias para evitar la ambigüedad. Esta técnica de mitigación puede combinarse con los tipos de identificación de las secciones 3.1.2 y 3.1.4.

3.2.3 Inversa de la varianza

En este caso, por definición, hay que combinarlo con la técnica de identificación de la sección 3.1.3, donde obtenemos la varianza de las medidas de distancia para luego establecer el valor de β_i como la inversa de la misma, tal y como se muestra en la siguiente ecuación.

$$\beta_i = \frac{1}{\sigma_i^2}, i = 1, \dots, N \quad (3.2)$$

donde σ_i indica la desviación típica de las medidas de distancia respecto del nodo ancla i , siendo la varianza el cuadrado de esta y N el número total de nodos ancla.

3.2.4 β_{LOS} igual a uno y β_{NLOS} distinto de cero

En este último tipo de mitigación, tras identificar previamente la condición LOS/NLOS con una de las técnicas de identificación de las secciones 3.1.2 o 3.1.4, estableceremos el valor de β_{LOS} a 1 y el valor de β_{NLOS} a un valor constante menor que 1 y distinto de 0.

$$\beta_i = \begin{cases} 1 & \text{si la medida es LOS} \\ \beta_{NLOS} & \text{si la medida es NLOS} \end{cases} \quad (3.3)$$

Para elegir el valor de β_{NLOS} adecuado, habrá que probar varios valores para ver cómo cambia el resultado y seleccionar el valor óptimo.

3.3 Algoritmos empleados

A continuación, se van a explicar los cuatro algoritmos implementados para la realización del proyecto.

3.3.1 Algoritmo sin identificación

Este primer algoritmo combina el tipo de identificación de la sección 3.1.1 (Sin identificar) con el tipo de mitigación de la sección 3.2.1 (Sin mitigar). De esta forma, asignaremos todos los valores de β_i a 1 y la ecuación (2.7) queda de la siguiente forma:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \left\{ \sum_{i=1}^N (\hat{d}_i - \|\mathbf{x} - \mathbf{x}_i\|)^2 \right\} \quad (3.4)$$

Este algoritmo se usará como referencia para comparar los algoritmos de mitigación reales.

3.3.2 Algoritmo con conocimiento perfecto de la condición de visión

En este caso combinaremos la identificación de la sección 3.1.2 (Conocimiento perfecto de la condición LOS/NLOS) con la mitigación de las secciones 3.2.2 (*Identify and discard*) y 3.2.4 (β_{LOS} igual a uno y β_{NLOS} distinto de cero).

Este es un algoritmo ideal, pues conocemos perfectamente la condición LOS/NLOS de las medidas de distancia y, al igual que haremos con el algoritmo sin identificación, lo usaremos como referencia para comparar los algoritmos de mitigación reales, que se explican a continuación.

3.3.3 Algoritmo basado en la varianza de las medidas

Este será el primero de los dos algoritmos reales que implementemos, basado en la identificación mediante la técnica "Varianza de las medidas de distancia" y la mitigación "Inversa de la varianza" antes explicadas, de forma que, tras sustituir el valor de β_i en la ecuación (2.7), obtenemos la siguiente ecuación:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \left\{ \sum_{i=1}^N \frac{1}{\sigma_i^2} (\hat{d}_i - \|\mathbf{x} - \mathbf{x}_i\|)^2 \right\} \quad (3.5)$$

En [GS04] y [CS98], sus autores establecieron el uso de la inversa de la varianza como solución de máxima verosimilitud (*Maximum Likelihood Solution*) para términos de ruido independiente y distribución Gaussiana. No obstante, como observaremos en la sección 4.3, para la estimación de puntos fijos, este algoritmo puede no ser lo suficientemente efectivo.

3.3.4 Algoritmo basado en estadísticos de la señal recibida

Como último algoritmo empleado, usaremos para la identificación el "Algoritmo basado en estadísticos de la señal [CGWI07]" y mitigaremos, al igual que con el algoritmo con conocimiento perfecto de la condición de visión, con las técnicas de las secciones 3.2.2 y 3.2.4.

Concretamente, aplicaremos los porcentajes de identificación que obtendremos de la tabla 4 de [GCWI07], donde para un modelo de canal LOS de interior de oficinas, la probabilidad de identificación correcta de LOS será de un 97.9%, siendo el 2.1% restante una identificación incorrecta, estableciéndose como NLOS; para el modelo de canal NLOS de interior de oficinas, esta probabilidad de identificación correcta pasa a ser 95.9%, mientras que el 4.1% restante corresponde a una identificación incorrecta, estableciéndose como LOS.

- Para el caso LOS:

$$\beta_i = \begin{cases} \beta_{LOS} & \text{si } k \leq 97.9\% \\ \beta_{NLOS} & \text{si } k > 97.9\% \end{cases} \quad (3.6)$$

siendo k un número aleatorio entre 0% y 100%.

- Para el caso NLOS:

$$\beta_i = \begin{cases} \beta_{NLOS} & \text{si } k \leq 95.9\% \\ \beta_{LOS} & \text{si } k > 95.9\% \end{cases} \quad (3.7)$$

siendo k un número aleatorio entre 0% y 100%.

De esta forma, β_i tendrá dos valores diferentes, uno para el caso LOS y otro para el caso NLOS, según la técnica de mitigación que utilicemos. Al ser las probabilidades tan altas, son escasas las posibilidades de identificación incorrecta, pero si por ejemplo establecemos un peso muy diferente entre los casos LOS y NLOS y se identifica erróneamente una medida, esto provocará una estimación imprecisa de la posición del nodo móvil.

Existe una particularidad en cuanto a la evaluación de este algoritmo: la necesidad de conocer el entorno que rodea a cada nodo ancla, sabiendo así para qué posiciones el canal puede tener la condición de línea de visión directa o no.

Cabe destacar que para la realización de este algoritmo durante el proyecto hemos establecido las probabilidades previamente comentadas, sin embargo estas no tienen por qué ser las correctas en todas las situaciones, pues dependen del algoritmo de identificación utilizado y los valores que hemos tomado son los resultados en [CGWI07] para una campaña de medidas concreta. Como se comentará en la sección 5, una investigación futura podría consistir en la comparación del RMSE (*Root Mean Square Error*) en función de las probabilidades de identificación correcta.

4 Desarrollo práctico

Para la evaluación práctica de los algoritmos empleados para la mitigación de errores en un sistema de localización UWB con obstrucción de línea de visión directa, se ha tomado una serie de medidas con el equipo *EVK1000 Evaluation Kit* de la compañía DecaWave, en un lugar de interior. Este lugar de estudio está situado en el Departamento de "Teoría de la Señal y Comunicaciones" de la Escuela Técnica Superior de Ingeniería de Sevilla, y está formado por un pasillo y una sala.

El lugar incluye diferentes obstáculos para la obstrucción de línea de visión directa, tales como paredes, sillas de escritorio, un armario metálico, estanterías metálicas, mesas de madera y paso esporádico de personas, provocando así un alto número de diferentes medidas y configuraciones según la condición del canal.

4.1 Entorno de trabajo

El equipo *EVK1000 Evaluation Kit* ha sido el elegido para realizar la toma de medidas durante el desarrollo del proyecto, incluyendo [Dec]:

- Dos cables USB 2.0. Uno conectará la placa que actúe como nodo móvil a una batería externa, y el otro conectará la placa restante, que actuará como nodo ancla, con el PC.
- Dos placas EVB1000, de dimensiones 7x7 centímetros. Estas placas tienen en su parte frontal una pantalla LCD que muestra información acerca de la distancia media medida y la desviación típica, no obstante a la hora de tomar estos datos usaremos el software proporcionado por la propia empresa *DecaWave*, configurado según el apéndice A al final de este documento.
- Dos antenas.
- Manual de uso.

A la hora de realizar el estudio, se ha diseñado un plano según el dibujo de las losetas del departamento, teniendo estas una forma cuadrada de 50 centímetros de lado. Se ha definido la posición de cinco nodos ancla y una serie de puntos en los que se han tomado medidas, disponiendo de diferentes situaciones para poder realizar un análisis más completo. La distribución del plano y la posición de los cincuenta y siete puntos (representados con circunferencias azules) y los cinco nodos ancla (cruces rojas) pueden observarse en la Figura 4.1.

Durante la toma de medidas, se ha de destacar que la puerta de la sala siempre se ha encontrado abierta, de forma que el Nodo Ancla 1 tiene visión directa con algunos puntos situados en el interior de la sala, al igual que los Nodos Ancla 3 y 5 con algunos puntos del exterior de la sala.

Los Nodos Ancla 1, 2 y 4 se encuentran situados en el exterior de la sala de la siguiente manera: el Nodo 2 está colocado en la parte vertical del pasillo según el plano, teniendo visión directa con

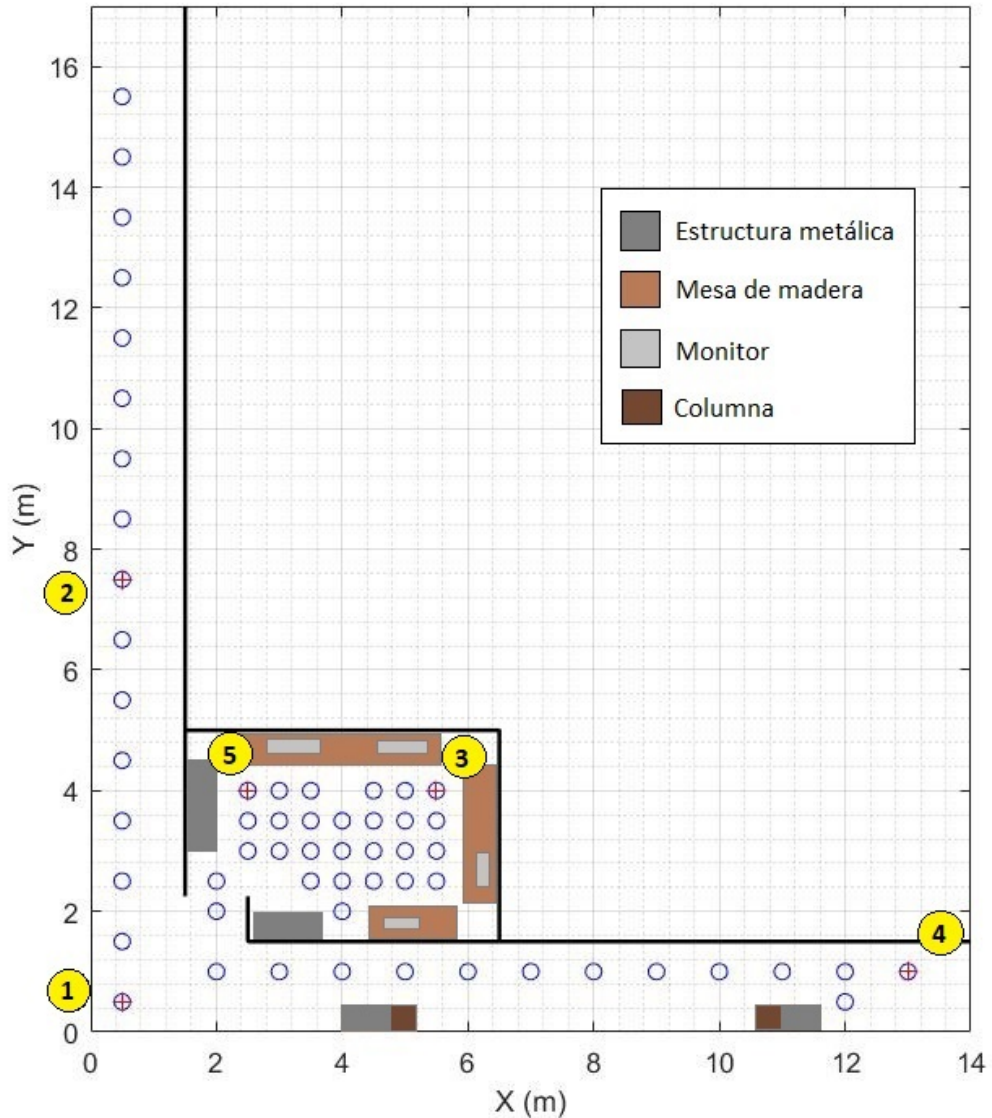


Figura 4.1 Distribución de nodos ancla y puntos en la sala y pasillo.

los puntos de esta misma parte del pasillo; el Nodo 4, en cambio, está en la parte horizontal del pasillo, teniendo visión directa con los puntos de esta misma zona horizontal; y por último, el Nodo 1 se encuentra en la intersección de ambas partes del pasillo, teniendo visión directa con las posiciones de las dos partes más algunas posiciones del interior de la sala. Para el estudio de los puntos colocados en el interior de la sala, se tiene influencia de muros, mesas de madera, armario y estantería metálicos y ordenadores, según desde qué nodo ancla se realice la medida.

Por otra parte, los Nodos Ancla 3 y 5 están colocados en el interior de la sala, teniendo visión directa para todos los puntos de la sala y algunos del pasillo, cerca de la entrada. Respecto al resto de puntos del pasillo con los que tienen obstrucción de línea de visión directa (NLOS), se tiene influencia de los ya mencionados muros, mesas de madera, armario y estantería metálicos y ordenadores, según las posiciones de los puntos a medir.

Para poder obtener el mayor número de resultados posibles en este estudio, se han realizado

medidas de distancia en un total de cincuenta y siete posiciones del nodo móvil, donde al menos tres nodos ancla podían obtener dichas medidas, ya que en algunas situaciones, al haber demasiados obstáculos entre el nodo ancla y móvil, no llegaba siquiera a haber comunicación entre ambos.

En el apéndice C se pueden ver los datos utilizados para este desarrollo práctico, donde se encuentran las posiciones reales del nodo móvil, posiciones de los nodos ancla, distancias medidas por cada nodo ancla (tomando una medida por cada par nodo móvil-ancla), desviación típica de la medida y condición de línea de visión. Concretamente, si nos fijamos en los datos de la sección C.5 (Condición de línea de visión), observaremos una casuística variada, pues en las distintas posiciones del nodo móvil tenemos distinto número total de anclas con comunicación a ese punto, y de las anclas que tenemos, el número total de medidas LOS es variable. Esto será luego clave a la hora de interpretar los resultados.

A continuación, se va a realizar el análisis de los diferentes resultados obtenidos según el algoritmo utilizado (mediante los códigos del apéndice B), mostrando finalmente las conclusiones alcanzadas tras el estudio.

4.2 Resultados para los algoritmos de referencia

Primeramente, se han mostrado tres ejemplos de cómo sería la solución a la técnica de localización descrita en la sección 2.2.3, mostrando una gráfica del plano del departamento en la que aparecen la posición real de cada punto (circunferencias verdes), su posición estimada según la técnica de localización utilizada (equis azules), la distancia entre la posición estimada y su correspondiente posición real (líneas negras) y por último la posición de cada nodo ancla (cruces rojas). En el caso en el que la posición del nodo móvil coincida con la posición de un nodo ancla, no se realizará esta medida, pues cuando la distancia entre ellos es muy corta (del orden de varios centímetros) la medida es muy imprecisa.

Para cada ejemplo se van a mostrar dos gráficas, puesto que para hallar el argumento mínimo de una función en Matlab pueden aplicarse, entre otras técnicas, las dos siguientes: por un lado se puede usar la función *fminsearch*, que devuelve un mínimo local de la función pasada (pudiendo haber un mínimo global que no considere)[Mata] mediante el uso del algoritmo simplex de Nelder-Mead [Matb] [LRWW98]; por otro lado, se puede realizar una rejilla o *grid* del plano del departamento, creando un conjunto de puntos con todas las posiciones posibles de dicho plano (cuanto menor sea la distancia entre estos puntos, mayor precisión tendrá la técnica utilizada pero requiriendo un tiempo mayor en la ejecución de la misma) y evaluando la función en cada uno de estos puntos, quedando con la posición en la que el valor de la función sea mínimo. Durante la ejecución de las pruebas con la técnica de *grid*, se ha establecido una distancia entre puntos del mismo de 25 centímetros, ya que es submúltiplo de la distancia mínima entre dos posiciones reales (50 centímetros) y no exige un tiempo muy alto de ejecución del código en Matlab.

En este primer ejemplo usaremos el algoritmo sin identificación (descrito en la sección 3.3.1), donde no se ha contemplado la identificación de si la condición de línea de visión es directa o no, dando un valor de 1 a todos los pesos de las medidas (en otras palabras, se está evaluando también un algoritmo que no trata de mitigar las medidas NLOS, para compararlo con los algoritmos de mitigación objeto de este trabajo). Para ello, se ha llamado a las funciones creadas en Matlab correspondientes a los códigos B.1 y B.2 del Anexo B. La primera utiliza la función *fminsearch* y la segunda, la técnica de la rejilla o *grid* previamente comentada. Estas funciones tienen como único parámetro la ruta donde se encuentran los datos de las medidas y de las posiciones de los nodos. Para cada posición del nodo móvil se ha utilizado el conjunto de medidas disponible, es decir, un total de $N \leq 5$ nodos ancla utilizados para cada posición del nodo móvil.

Como puede observarse en la Figura 4.2, esta técnica aportaría una solución ineficaz, pues al no reducir el peso de las medidas sin línea de visión directa, las cuales son las que mayor error poseen,

la estimación de las posiciones es diferente a la posición real de cada punto. Además, cuando las anclas no rodean la posición a localizar, el error es grande.

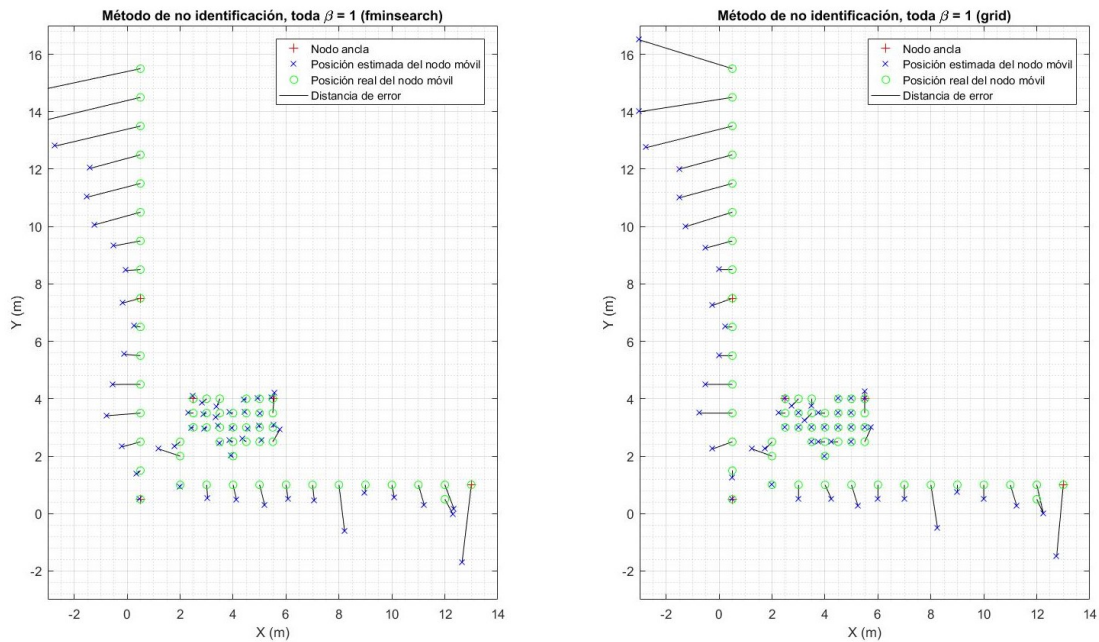


Figura 4.2 Distribución de posiciones estimadas sin mitigación.

Como segundo ejemplo, se ha utilizado el algoritmo con conocimiento perfecto de la condición de visión, comentado en la sección 3.3.2. Para este caso, se han ejecutado las dos funciones creadas en Matlab correspondientes a los códigos B.3 y B.4 a las que se le han pasado como parámetros la ruta en la que se encuentran los datos de las medidas y de las posiciones de los nodos, y el peso que queremos darle a la β_i cuando la medida sea de tipo NLOS (en este caso el valor será 0, por lo que se realizará la mitigación *Identify and discard*).

En la Figura 4.3, observamos una mejora considerable respecto al ejemplo anterior mostrado, sin embargo esta no es una técnica lo suficientemente precisa debido al descarte completo de las medidas NLOS. Tras descartar las medidas NLOS puede que no queden suficientes medidas para lograr la localización (en 2 dimensiones harían falta al menos 3 medidas).

Un ejemplo de este error lo podemos ver en la estimación del nodo móvil en la posición (13,1) (medida en metros), para la que sólo tenemos una medida LOS (nodo ancla 1) y tres medidas NLOS (nodos ancla 2, 3 y 5) y una medida descartada (nodo ancla 4, pues la posición del nodo móvil y ancla coinciden, tal y como se explica en la sección C.5). Al descartar las medidas NLOS, solo tenemos el nodo ancla 1 como recurso para estimar la posición del nodo móvil, lo cual es insuficiente.

Por último, como tercer ejemplo se va a llamar de nuevo a los dos códigos del segundo ejemplo, pero esta vez el valor de β para la condición de NLOS será 0.001, pues se ha comprobado que tras ejecutar el código con un rango de valores de β_{NLOS} desde 0 hasta 1 (primero con intervalos de 0.1 y posteriormente afinando en torno a los valores que han ido resultando mejores), éste ha sido el que menor error ha devuelto como resultado. De esta forma, las medidas con esta condición no serán completamente descartadas y aportarán información, pero con un peso mil veces menor que las medidas LOS. En la Figura 4.4 se muestra la distribución de posiciones estimadas con esta técnica.

En este caso observamos cómo la estimación del nodo móvil en la posición (13,1) (medida en metros) mejora considerablemente al no descartar las medidas NLOS.

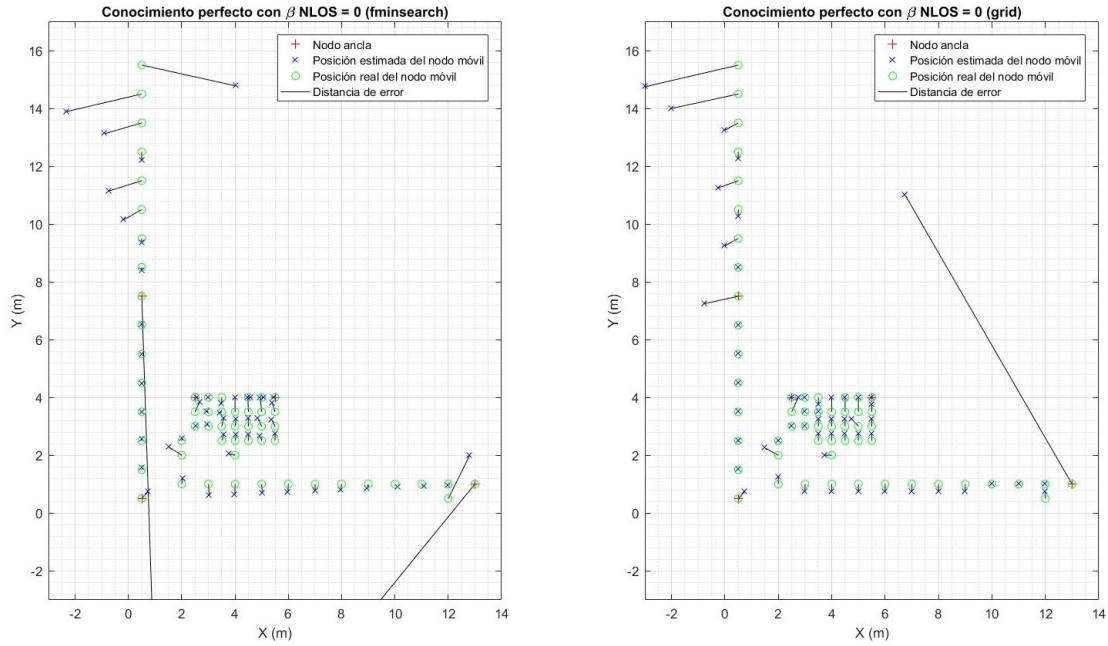


Figura 4.3 Distribución de posiciones estimadas con conocimiento perfecto de la condición LO-S/NLOS y $\beta_{NLOS} = 0$.

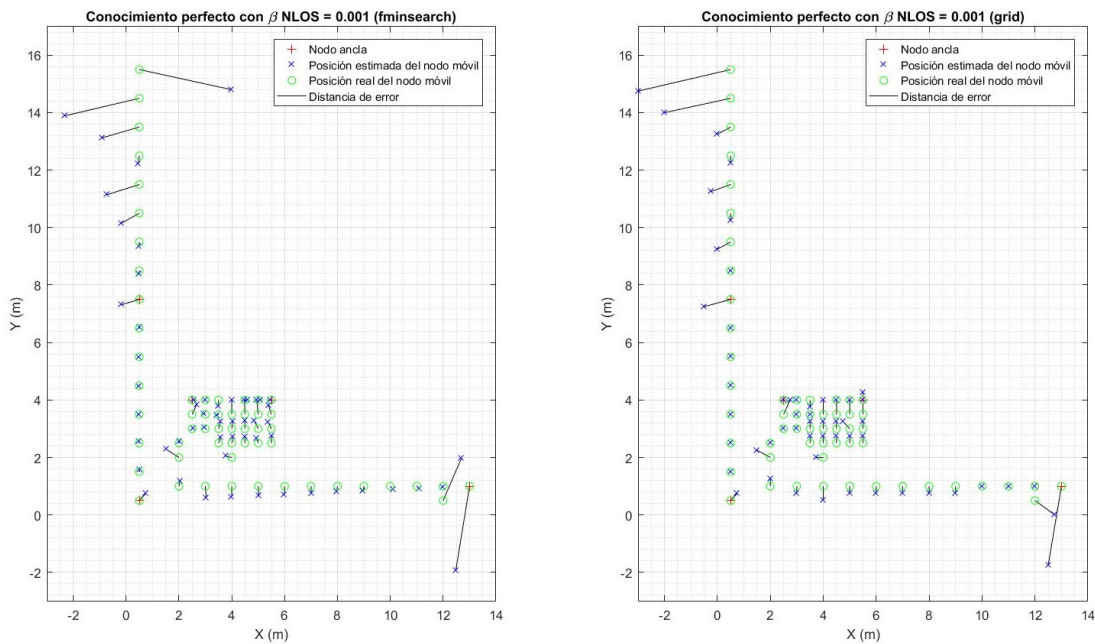


Figura 4.4 Distribución de posiciones estimadas con conocimiento perfecto de la condición LO-S/NLOS y $\beta_{NLOS} = 0.001$.

Un ejemplo de cómo minimizamos la función de la ecuación (2.7) en Matlab se puede ver en las figuras 4.5 y 4.6. Para ello, vamos a obtener la función en el punto (3,4) (metros) y vamos a evaluarla para x desde 0 hasta 14 metros y para y desde 0 hasta 17 metros. En estas figuras se nos indica que cuanto más claro sea el color, mayor será el valor de la función, y cuanto más oscuro, menor.

Podemos observar en la figura 4.6 cómo la zona cercana al punto (3,4) (metros) es más oscura, pues el valor de la función es muy bajo y por tanto ahí es donde se encontrará el mínimo. Pueden existir mínimos locales, ya que la zona azulada es muy extensa y con valores muy similares, de modo que el método *fminsearch* puede no ser eficaz, mientras que el uso del *grid* antes comentado nos asegurará encontrar el mínimo global.

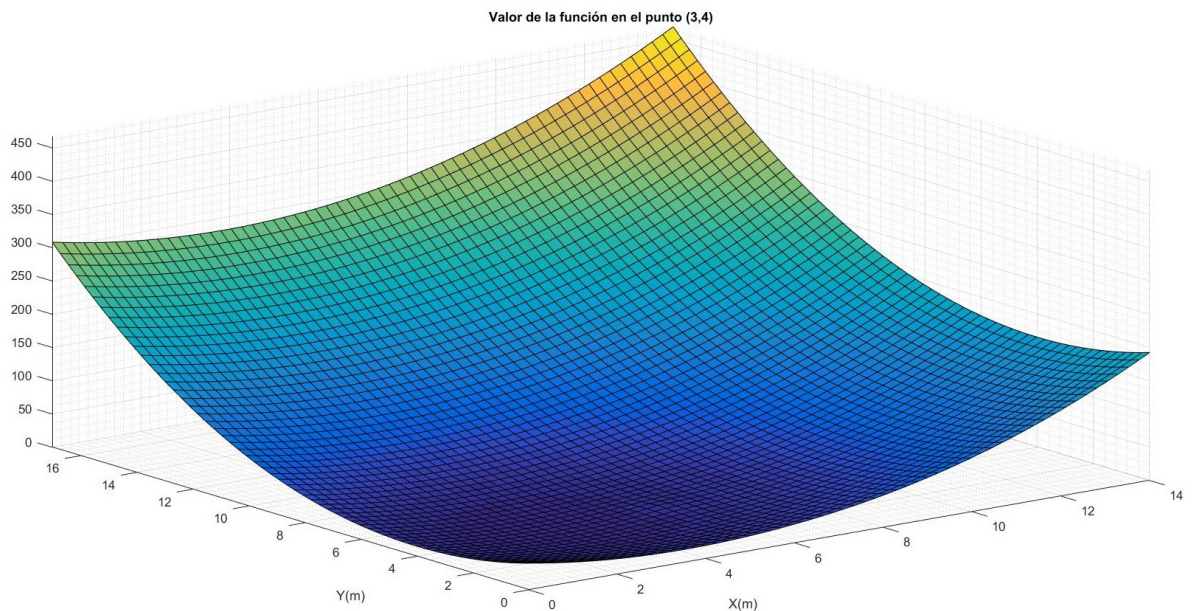


Figura 4.5 Representación en 3D del valor de la función en el punto (3,4).

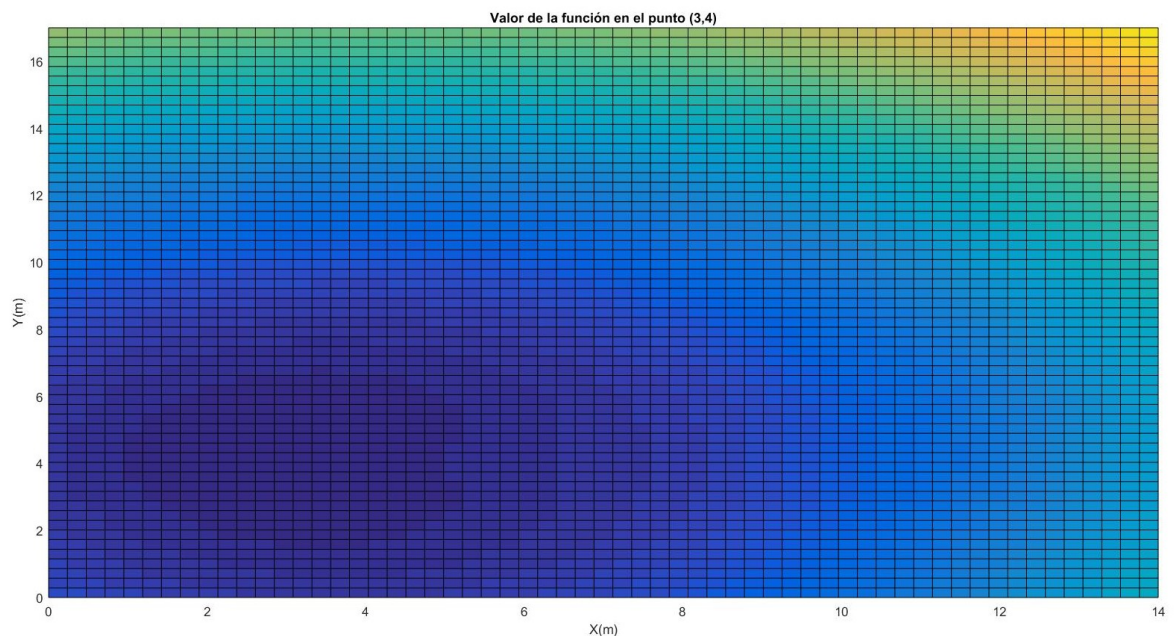


Figura 4.6 Representación en 2D del valor de la función en el punto (3,4).

A continuación, se van a exponer los resultados de los algoritmos desarrollados, mostrando junto a sus gráficas correspondientes las gráficas de la derecha de la Figura 4.2 y la Figura 4.4

para tomarlas como referencia de no tratar de mitigar y de mitigar con información perfecta de la condición de visión, respectivamente. Comparando ambas gráficas se ve que en algunas zonas la presencia de NLOS no ha perjudicado mucho a la estimación, pero en otras zonas sí hay bastante margen de mejora.

4.3 Resultados para el algoritmo basado en la varianza de las medidas

En este primer algoritmo de mitigación, tal y como se ha comentado en la sección 3.3.3, daremos un peso a cada medida según el valor de su varianza, obtenida a partir de su desviación típica (obtenida en centímetros) según la ecuación (3.2). Al igual que se ha hecho con los casos de referencia, se han creado dos funciones en Matlab, una que minimiza la función con la técnica de *fminsearch* y otra con la técnica de la rejilla o *grid*. Estas funciones tienen como único parámetro la ruta donde se encuentran los datos de las medidas y los nodos, mostradas en el Anexo B, códigos B.5 y B.6.

Al ejecutar estos códigos junto a los dos de referencia mencionados anteriormente, observamos en la figura 4.7 cuatro gráficas de las cuales las dos superiores se corresponden con los resultados del algoritmo de esta sección. La gráfica superior izquierda usa la técnica de minimización *fminsearch*, observándose a simple vista la obtención de un peor resultado en comparación con la técnica de minimización con *grid*, mostrada en la gráfica superior derecha. Pese a que posteriormente en la sección 4.5 se comentarán los errores provocados por cada algoritmo, en estas gráficas destaca cómo el algoritmo de varianza con *grid* mejora notablemente con respecto al método de no identificación (gráfica inferior izquierda), pero aún existe una imprecisión en la estimación de posiciones con respecto al método de conocimiento perfecto (gráfica inferior derecha).

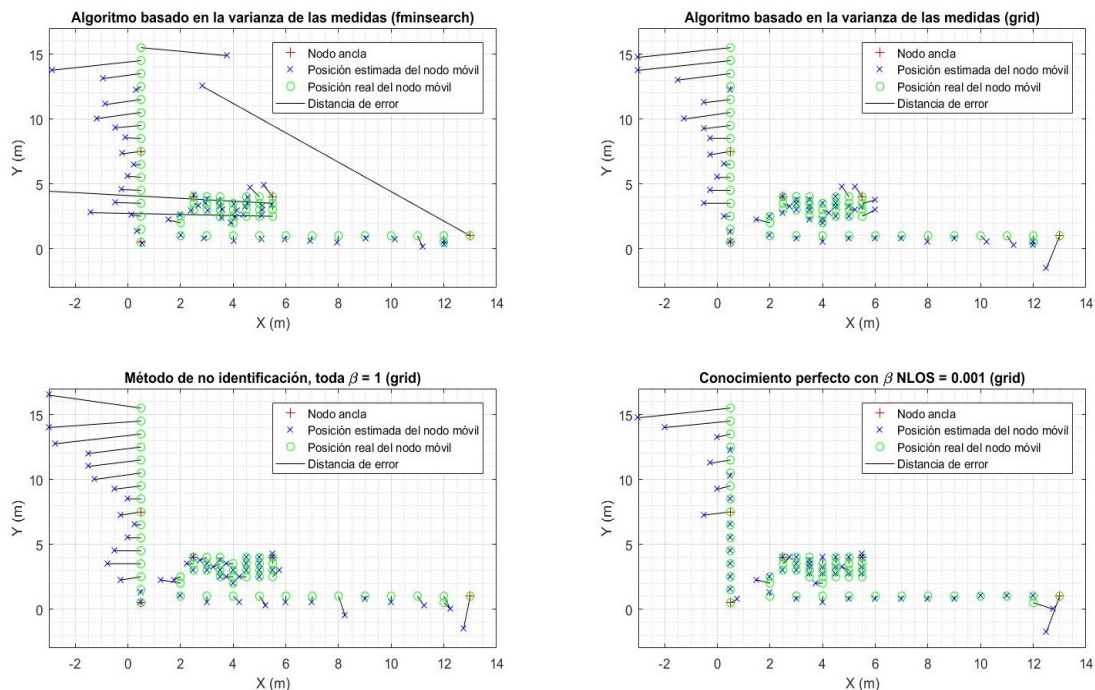


Figura 4.7 Distribución de posiciones estimadas con el algoritmo basado en la varianza de las medidas.

4.4 Resultados para el algoritmo basado en estadísticos de la señal recibida

Para evaluar el algoritmo basado en estadísticos de la señal recibida, explicado en la sección 3.3.4, en la función que ha sido creada en Matlab generaremos para cada medida un valor aleatorio entre 0 y 1 y actuaremos de la siguiente manera: si la condición real de esta medida es LOS y el valor aleatorio obtenido es igual o inferior a 0.979, la identificación es correcta y se le asignará el valor establecido como β_{LOS} (en nuestro caso, 1), y si por contrario es superior a 0.979, se le asignará el valor de β_{NLOS} (pasado por parámetro de la función); si la condición real de la medida es NLOS, este valor aleatorio tendrá que ser igual o inferior a 0.959 para ser identificado como NLOS, o superior para ser erróneamente identificado como LOS. Los códigos de las funciones (el primero usa la técnica de minimización con *fminsearch* y el segundo con *grid*) se muestran en el Anexo B, códigos B.7 y B.8.

Como estas dos funciones dependen de un valor aleatorio, podríamos ejecutar dos veces la misma función y el resultado que se obtendría podría ser distinto, luego para el cálculo de la raíz del error cuadrático medio sería conveniente hacer la media de errores de un número alto de ejecuciones de la misma función. Sin embargo, para mostrar una gráfica de la estimación de posiciones como se ha hecho en las secciones anteriores, tomaremos una sola ejecución como ejemplo ilustrativo, y en el apartado 4.5 se ofrecerán resultados de RMSE.

Para todas las funciones, el valor de β_{LOS} se ha establecido como 1, sin embargo desconocemos el valor óptimo de β_{NLOS} . Por tanto, este algoritmo de probabilidades ha sido ejecutado con diferentes valores hasta observar que el mejor resultado se obtiene para una β_{NLOS} con valor en torno a 0.001. Aun así, se van a mostrar gráficas de estimación de posiciones para ejecuciones con valores de β_{NLOS} igual a 0.8, 0.4 y 0.001 en las Figuras 4.8, 4.9 y 4.10, respectivamente. Además, se mostrarán en la parte inferior de las figuras las dos gráficas de referencia comentadas anteriormente.

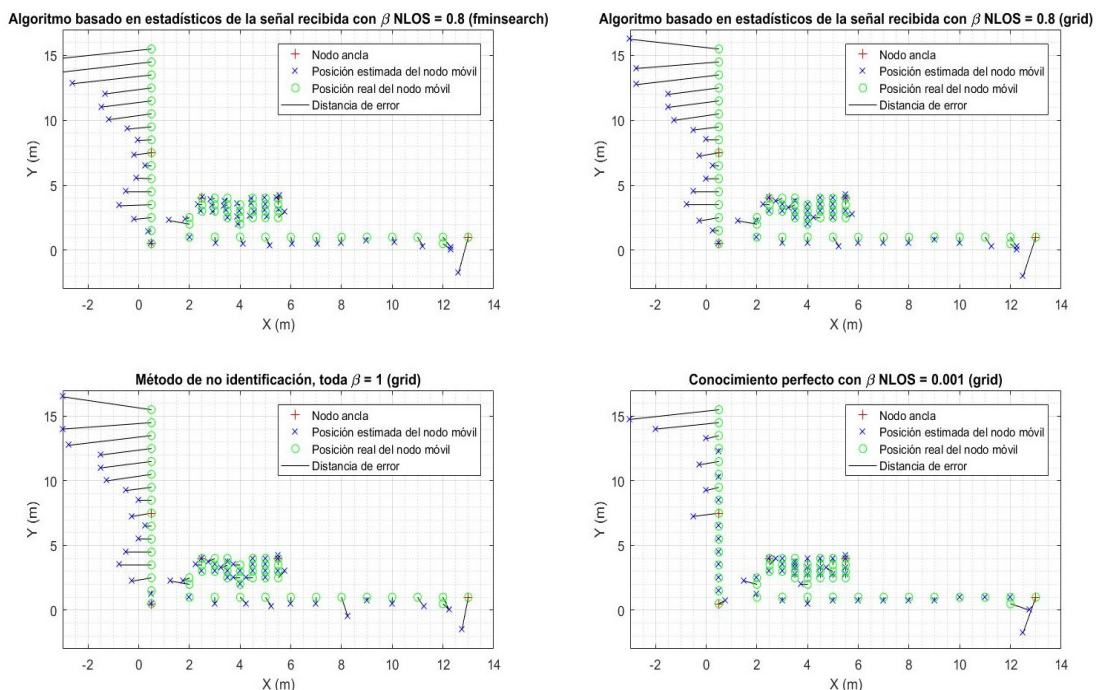


Figura 4.8 Distribución de posiciones estimadas con algoritmo basado en estadísticos de la señal recibida y $\beta_{NLOS} = 0.8$.

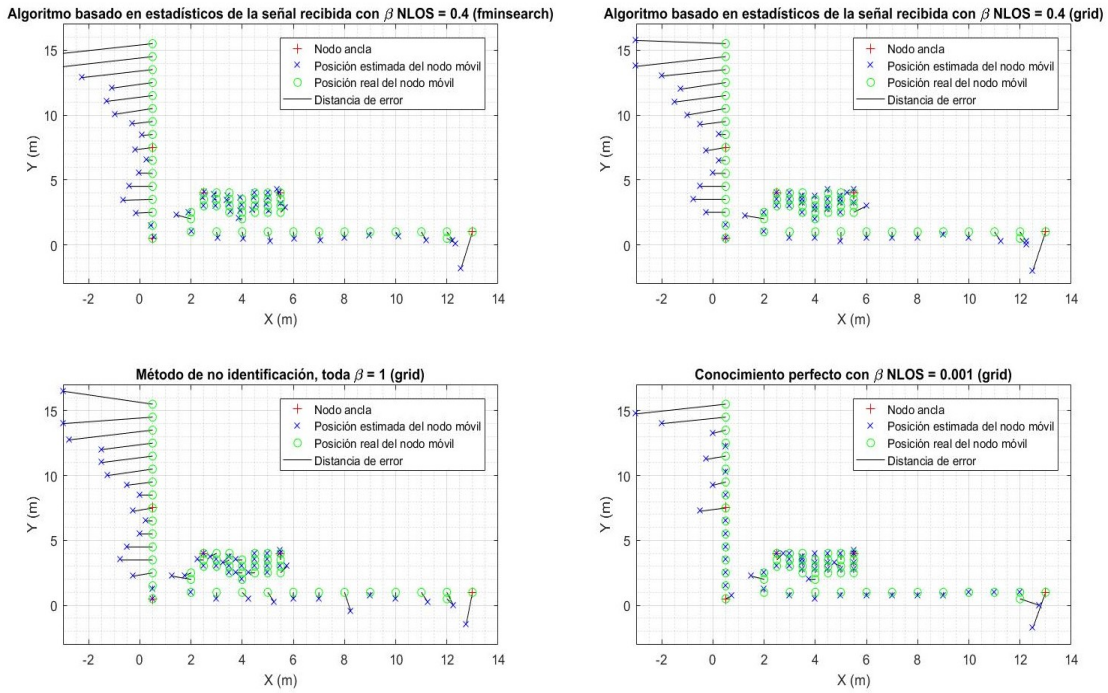


Figura 4.9 Distribución de posiciones estimadas con algoritmo basado en estadísticos de la señal recibida y $\beta_{NLOS} = 0.4$.

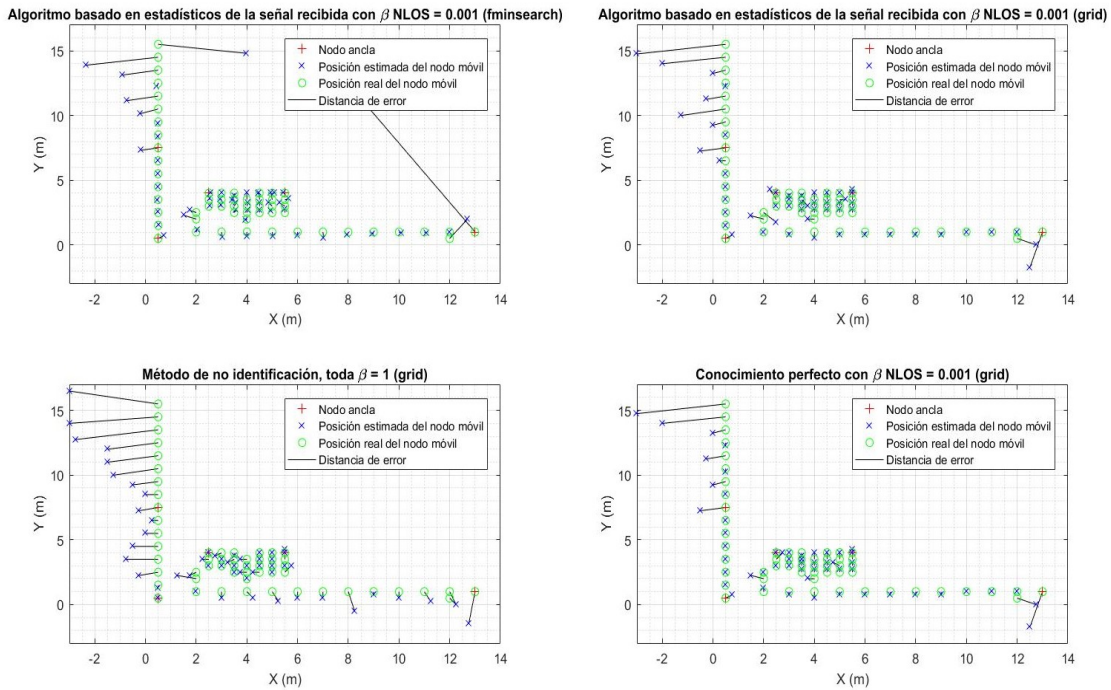


Figura 4.10 Distribución de posiciones estimadas con algoritmo basado en estadísticos de la señal recibida y $\beta_{NLOS} = 0.001$.

Si observamos con detenimiento estas figuras, observamos cómo en la parte vertical del pasillo, la estimación de posiciones mejora conforme el valor de β_{NLOS} va disminuyendo. Como se ha comentado previamente, estas gráficas se corresponden con una de muchas ejecuciones que deberían realizarse para obtener un promedio estadístico de los resultados del algoritmo.

En la figura 4.8 vemos que los resultados de la gráfica superior derecha y de la gráfica inferior izquierda son muy parecidos. Esto se debe a que los valores de β_{NLOS} (0.8 y 1, respectivamente) son muy parecidos, aportando una estimación de posiciones muy similar.

En la figura 4.10 vemos un gran parecido entre las dos gráficas de la derecha, pues solo se diferencian en el tipo de identificación. No obstante, tenemos un claro ejemplo de diferente estimación del nodo móvil en la posición (0.5,10.5) (metros), pues en la gráfica superior la estimación es errónea. Esto se debe a que ha habido una identificación incorrecta de la condición LOS/NLOS en alguna de las medidas de los nodos ancla.

4.5 Gráfica RMSE

Como análisis final, se va a realizar un histograma que compara los distintos valores de RMSE obtenidos. El RMSE (*RootMeanSquareError*, Raíz del Error Cuadrático Medio) no es más que la raíz de la media de los errores (distancia entre la posición real y la estimada) al cuadrado [Sta], calculado a partir de la siguiente fórmula:

$$RMSE \approx \sqrt{\frac{1}{n} \sum_{t=1}^n e_t^2} \quad (4.1)$$

donde e_t es la diferencia en metros entre la posición real del punto y su posición estimada, siendo n el número total de puntos en los que se han tomado medidas.

En todos los métodos, al tener una sola medida por cada posición, el sumatorio se ha realizado sobre las distintas posiciones. Como el número de posiciones, n , es relativamente pequeño, no se pretende dar una validez general al resultado que salga de RMSE. Es más, la ecuación anterior da como resultado una aproximación al RMSE, pues incluye la media muestral del error cuadrático, y no la esperanza del error cuadrático [dRoj18].

Además, en el método basado en estadísticos de la señal recibida para cada posición real no se va a tener una única estimación de posición, pues al depender de valores de probabilidad, el algoritmo debe ejecutarse un gran número de veces para que se puedan dar los casos de identificación LOS/NLOS correcta e incorrecta. Por tanto, para obtener una aproximación al RMSE representativo de este método, se ha ejecutado $k = 500$ veces para cada caso en el que se utiliza este algoritmo (mediante el código B.10). De esta forma, el sumatorio de la ecuación (4.1) para el método basado en estadísticos de la señal tendrá nk términos.

Esta ecuación ha sido aplicada para todos los algoritmos previamente explicados, dando como resultados los siguientes valores mostrados en la Tabla 4.1.

Para una mejor comprensión de estos valores, a continuación en la Figura 4.11 se muestra un histograma con dos conjuntos de columnas, a la izquierda se encuentran las columnas para las que el valor de RMSE obtenido ha sido calculado a partir de la técnica de *fminsearch* ya mencionada, mientras que en el conjunto de la derecha se ha usado la técnica de *grid*. A primera vista se contempla una clara mejora cuando usamos la técnica de *grid*, sin embargo esto requiere un mayor tiempo de ejecución del algoritmo.

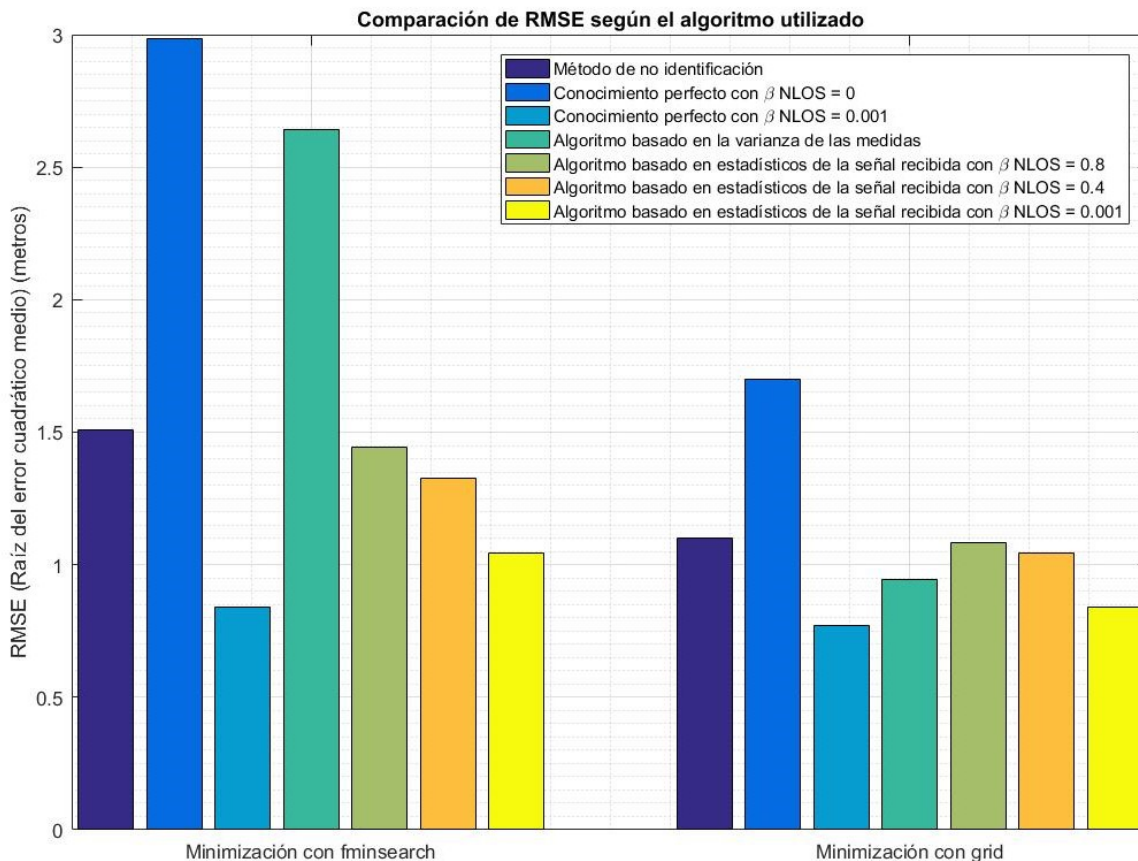
Como ya se comentó en el apartado 4.2, puede chocar que el método con conocimiento perfecto y $\beta_{NLOS} = 0$ no sea el mejor, a veces puede funcionar mal porque al descartar completamente las medidas NLOS queden menos de 3 medidas. Por eso hay una diferencia grande entre $\beta_{NLOS} = 0$ y $\beta_{NLOS} = 0.001$.

Tabla 4.1 Tabla de valores RMSE (metros) según algoritmo utilizado.

Algoritmo	Técnica <i>fminsearch</i>	Técnica <i>grid</i>
No identificación	1.5088 m	1.1002 m
Conocimiento perfecto con $\beta_{NLOS} = 0$	2.9848 m	1.6978 m
Conocimiento perfecto con $\beta_{NLOS} = 0.001$	0.8411 m	0.7716 m
Varianza	2.6414 m	0.9424 m
Estadísticos de la señal con $\beta_{NLOS} = 0.8$	1.4449 m	1.0844 m
Estadísticos de la señal con $\beta_{NLOS} = 0.4$	1.3277 m	1.0447 m
Estadísticos de la señal con $\beta_{NLOS} = 0.001$	1.0428 m	0.8408 m

Si nos fijamos detenidamente en la columna de Conocimiento perfecto con $\beta_{NLOS} = 0.001$ del conjunto de la derecha, observamos que es la que menor RMSE tiene, y por tanto, el mejor método. Sin embargo, este es un método de referencia ideal, de manera que el siguiente método con el menor valor para RMSE lo encontramos en la columna amarilla, que aplica el algoritmo basado en estadísticos de la señal recibida con $\beta_{NLOS} = 0.001$.

También se observa que mitigar con $\beta_{NLOS} = 0.8$ resulta muy parecido a no mitigar (como era de esperar) y que mitigar con el algoritmo basado en estadísticos de la señal con $\beta_{NLOS} = 0.001$ sí consigue avanzar en el margen de mejora que hay entre el método de no identificación y el método con conocimiento perfecto y $\beta_{NLOS} = 0.001$, sin llegar a alcanzar a este último debido a la posibilidad de identificación incorrecta.

**Figura 4.11** Comparación de RMSE según el algoritmo utilizado.

5 Conclusiones

Queda demostrado que la aplicación de la tecnología UWB en sistemas de localización de interiores es un éxito, sin embargo actualmente aún falta por avanzar y mejorar, especialmente en cuanto a los errores provocados por la obstrucción de línea de visión directa con ciertos materiales. Este proyecto ha contribuido en esta mejora mediante la evaluación de dos algoritmos para mitigar dichos errores.

- Por un lado tenemos el algoritmo basado en la varianza de las medidas, el cual aporta una buena mitigación pero no tenemos certeza de si en un caso real funcionaría de la misma manera, y es que para obtener la desviación típica con el equipo EVK1000 es necesario tomar al menos cincuenta medidas. Estas cincuenta medidas, durante la fase experimental del proyecto, se realizaron manteniendo la posición estática en ambos nodo ancla y móvil, mientras que en un caso real el nodo móvil podría estar en movimiento durante el cálculo de la desviación típica y podría tener un valor muy alto incluso en casos LOS. A primera vista parece un método asequible pero necesita perfeccionamiento.
- Por otro lado tenemos el algoritmo basado en estadísticos de la señal recibida, aportando la mejor solución de entre estos dos algoritmos. No obstante, se ha demostrado que la precisión de la tecnología UWB puede alcanzar cifras de escasos centímetros, mientras que nuestra mejor solución aportaba una precisión de unos 83 centímetros. Una mejora de este algoritmo podría basarse en la división de la zona de medidas (en nuestro caso, el plano del Departamento) en distintas zonas, para las que solo se apliquen en el algoritmo las medidas de los nodos ancla incluidos en dicha zona. Esto evitaría los datos aportados por nodos ancla lejanos que, pese a estimar una distancia respecto al punto, son imprecisos y solo empeoran la estimación de la posición. El inconveniente es la necesidad de un conocimiento previo del alcance de los nodos ancla en el plano total para la posterior división antes comentada.

Por último, para simular este algoritmo hemos usado las probabilidades de la tabla 4 de [GCWI07], modelo de canal CM3 y CM4 (interior de oficinas LOS y NLOS, respectivamente) y, al ser las probabilidades de identificación correcta tan altas (97.9% 95.9% respectivamente), los resultados son muy parecidos al caso de conocimiento perfecto usado como referencia. Una línea futura de investigación podría consistir en la prueba de este algoritmo con diferentes valores de probabilidad de identificación hasta poder responder a la pregunta "¿Qué valores de probabilidad de identificación de la condición LOS/NLOS son necesarios para un RMSE deseado?".

Otra idea de línea futura consistiría en poner a prueba el método de identificación de [GCWI07] para ver si en distintos entornos, las probabilidades de identificación correcta salen tan elevadas.

Al comparar las técnicas de minimización utilizadas, observamos cómo el método *grid* es mejor que el método *fminsearch*, pues el problema de los mínimos locales ha afectado a la precisión de los algoritmos de localización, disminuyéndola. Al usar el método *grid* nos aseguramos encontrar el mínimo global.

Tal y como se ve en la figura 4.11, en la versión con técnica *grid*, tanto el algoritmo basado en varianzas como el basado en estadísticos de la señal logran una mejora respecto del método que no trata de mitigar.

Respecto al uso de la técnica de mitigación *Identify and discard*, descartar completamente una medida no siempre es bueno, pues la estimación de la posición del nodo móvil depende del número de medidas que queden, y el error aumenta cuando este número es menor que 3.

Como conclusión final, hay que destacar que pese a la gran aportación de la tecnología UWB en cuanto a localización en interiores, aún existen muchas mejoras pendientes para su completa implantación. Es cierto que aún no es ampliamente conocida, sin embargo esta tecnología se está haciendo hueco en el mercado, por lo que la competencia y el avance serán mayores, de forma que en los próximos años será de gran ayuda en un amplio rango de aplicaciones: desde la localización de visitantes en museos para reproducir archivos de audio acerca de obras según la proximidad a ellas, hasta el control de equipos médicos implantados.

Apéndice A

Configuración del software

En este primer anexo vamos a explicar cómo debe ser configurado el software proporcionado por DecaWave para el correcto funcionamiento del kit EVK1000 durante el proyecto.

Primeramente, deberemos instalar el driver *STM32 Virtual COM Port Driver*, el cual permitirá la conexión entre el PC y el equipo EVK1000, estando disponible en la página de ST Microelectronics: <https://www.st.com/en/development-tools/stsw-stm32102.html>

Por otro lado, tendremos que instalar el software *DecaRanging* propio del kit EVK1000, el cual podemos encontrar en la página oficial de DecaWave, siguiendo el enlace:

<https://www.decawave.com/software/>.

Además de la documentación oficial [Dec], pueden servir de gran ayuda las secciones 3 y 9 de [ICal17], donde encontraremos información detallada acerca de cómo funciona este equipo y sus posibles configuraciones.

Durante la toma de medidas en la realización de la parte experimental del proyecto, antes de usar el equipo se ha tenido que configurar tanto el nodo móvil como el nodo ancla para que fueran emparejados de la siguiente manera:

- Para el nodo móvil, basta con conectarlo a una batería externa mediante un cable USB, de forma que comenzará a enviar mensajes *blink*, esperando a ser escuchado y posteriormente respondido por un nodo ancla, estableciéndose así el emparejamiento.
- Para el nodo ancla, necesitamos configurar los parámetros en el software instalado para operar en el mismo canal y así escuchar estos mensajes *blink* y poder responderlos.

En primer lugar, tras conectar el nodo ancla al PC y ejecutar el software *DecaRanging*, deberemos establecer el rol de *Listener* (el cual viene por defecto) y abrir la ventana de configuración *Channel Setup*, como se muestra en la figura A.1, obtenida de [ICal17].

La configuración que aparece en la ventana *Channel Setup* de la figura A.1 va a ser la utilizada durante el proyecto, donde observamos una frecuencia central de 6.5 GHz, ancho de banda de 500 MHz, longitud de preámbulo de 1024 repeticiones de símbolo, frecuencia de repetición de pulso igual a 16 MHz, velocidad de datos de 110 kbits/s y código de preámbulo igual a 3.

Una vez configurado, veremos cómo aparecen en pantalla los mensajes *blink* enviados por el nodo móvil. Ahora sólo basta con cambiar el rol del nodo ancla a *Anchor*, nos aparecerá un mensaje en pantalla donde se nos dará opción a elegir a qué nodo móvil queremos emparejar este ancla, seleccionaremos la dirección de nuestro nodo móvil y ya finalizará el proceso de emparejamiento.

Una vez configurado el equipo, veremos cómo aparecen por pantalla los datos de las medidas como pueden ser la distancia instantánea, una media de las distancias calculadas o la desviación típica (la cuál valdrá 0 hasta alcanzar cincuenta medidas, entonces obtendremos su valor en centímetros).

TX=0 RX=5 CE=0 RSE=0 HE=0 STO=2 FFE=0 TO=0 L=0

Instant: ToF: 0.000 ns Dist: 0.000 m

Mean[0]: Dist: 0.00 m STDEV: 0.00 cm

Long Term Average = 0.000000 m (over 0)

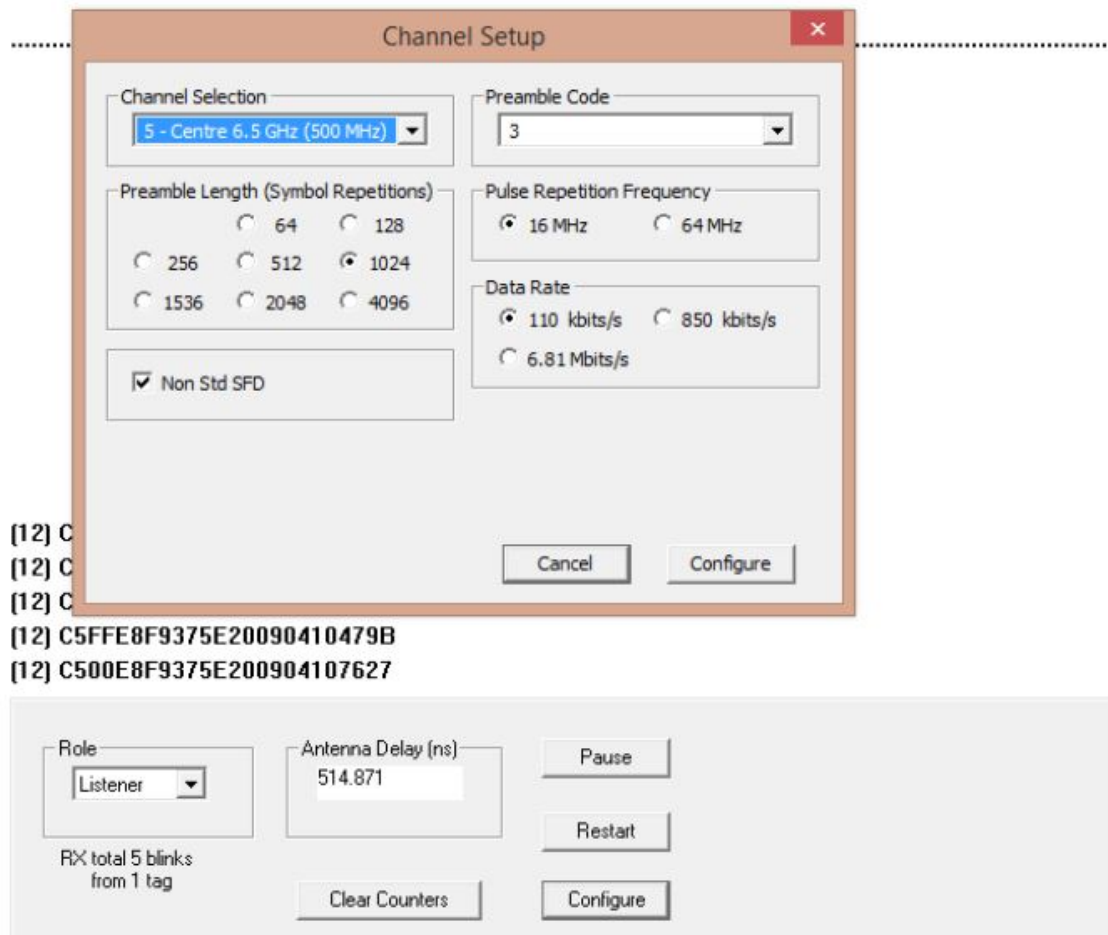


Figura A.1 Configuración del nodo ancla.

A continuación, se muestran una serie de fotografías tomadas durante el proceso de medición.

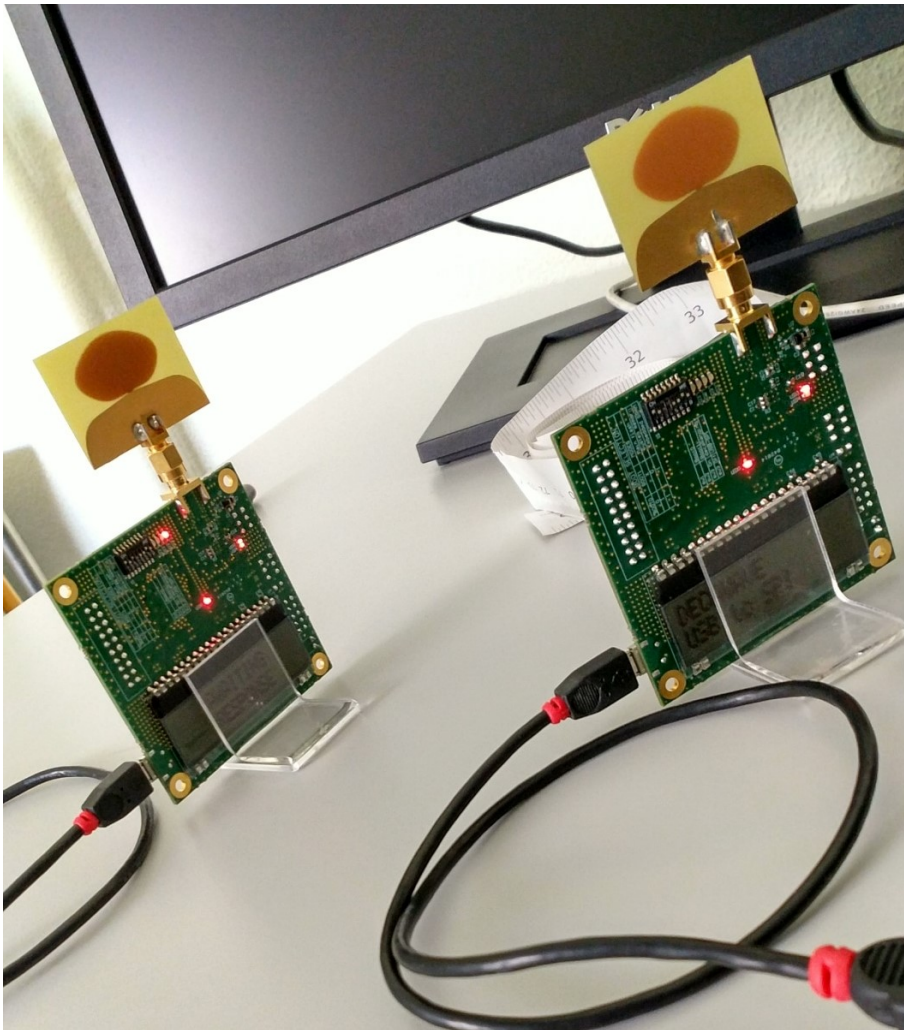


Figura A.2 Proceso de configuración del equipo.



Figura A.3 Proceso de medición de distancia en el pasillo del departamento.

Apéndice B

Códigos de Matlab

A continuación se van a mostrar los códigos de Matlab utilizados para la obtención de las representaciones gráficas expuestas en el capítulo 4 (Desarrollo práctico).

B.1 Algoritmo sin identificación

Código B.1 Algoritmo sin identificación y *fminsearch*.

```
function errors = WLSLocalizationConTodasLasMedidas(path)
realPositions=(load(strcat(path,'/realPositions.txt'))).*0.5;
fixedNodesPosition=(load(strcat(path,'/fixedNodesPosition.txt'))).*0.5;
measuredDistance=load(strcat(path,'/measuredDistance.txt'));
N=length(fixedNodesPosition(:,1)); %Número de nodos ancla
M=length(measuredDistance(:,1)); %Número de posiciones medidas
beta=ones(1,N);
estimPositions = ones(M,2);
errors=ones(1,M);
for k=1:1:M %Bucle para para cada medida
    funIndiv = @(x,y) 0;
    funTotal = @(x,y) 0;
    for i=1:1:N %Bucle para cada nodo ancla
        if(measuredDistance(k,i) ~= 0)
            funIndiv = @(x,y) beta(i)*(measuredDistance(k,i)-norm([x,y]-
                fixedNodesPosition(i,:)))^2;
            funTotal = @(x,y) funTotal(x,y) + funIndiv(x,y);
        end
    end
    x0=[0,0];
    [xmin, fval] = fminsearch(@(x)funTotal(x(1),x(2)), x0); %Evaluación
        de la función
    estimPositions(k,:) = xmin;
    errors(k)=norm(realPositions(k,:)-estimPositions(k,:));
    errorPlot = plot([realPositions(k,1) estimPositions(k,1)], [
        realPositions(k,2) estimPositions(k,2)], 'k');
    hold on
```

```

end
fixedPlot = plot(fixedNodesPosition(:,1),fixedNodesPosition(:,2),'+', '
    MarkerEdgeColor','red');
hold on
estimPlot = plot(estimPositions(:,1), estimPositions(:,2), 'x', '
    MarkerEdgeColor','blue');
hold on
realPlot = plot(realPositions(:,1), realPositions(:,2), 'o', '
    MarkerEdgeColor','green');
hold on
legend([fixedPlot, estimPlot, realPlot, errorPlot],{'Nodo ancla', '
    Posición estimada del nodo móvil', 'Posición real del nodo móvil', '
    Distancia de error'})
ylim([-3,17]);
xlim([-3,14]);
xlabel('X (m)');
ylabel('Y (m)');
title('Método de no identificación, toda \beta = 1 (fminsearch)');
grid on
grid minor
end

```

Código B.2 Algoritmo sin identificación y *grid*.

```

function errors = WLSLocalizationConTodasLasMedidasyGrid(path)
limx = [-3 20];
limy = [-3 20];
separator = 0.25;
realPositions=(load(strcat(path,'/realPositions.txt'))).*0.5;
fixedNodesPosition=(load(strcat(path,'/fixedNodesPosition.txt'))).*0.5;
measuredDistance=load(strcat(path,'/measuredDistance.txt'));
N=length(fixedNodesPosition(:,1)); %Número de nodos ancla
M=length(measuredDistance(:,1)); %Número de posiciones medidas
beta=ones(1,N);
estimPositions = ones(M,2);
errors=ones(1,M);
for k=1:1:M %Bucle para para cada medida
    funIndiv = @(x,y) 0;
    funTotal = @(x,y) 0;
    for i=1:1:N %Bucle para cada nodo ancla
        if(measuredDistance(k,i) ~= 0)
            funIndiv = @(x,y) beta(i)*(measuredDistance(k,i)-norm([x,y]-
                fixedNodesPosition(i,:)))^2;
            funTotal = @(x,y) funTotal(x,y) + funIndiv(x,y);
        end
    end
end
fmin = inf;
for xx=limx(1):separator:limx(2)
    for yy=limy(1):separator:limy(2) %Creación del grid

```



```

        fval = funTotal(xx,yy); %Evaluación de la función
        if fval < fmin %Comprobación de si es o no mínimo
            fmin = fval;
            estimPositions(k,:) = [xx,yy];
        end
    end
end
errors(k)=norm(realPositions(k,:)-estimPositions(k,:));
errorPlot = plot([realPositions(k,1) estimPositions(k,1)], [
    realPositions(k,2) estimPositions(k,2)], 'k');
hold on
end
fixedPlot = plot(fixedNodesPosition(:,1),fixedNodesPosition(:,2),'+', '
    MarkerEdgeColor','red');
hold on
estimPlot = plot(estimPositions(:,1), estimPositions(:,2), 'x', '
    MarkerEdgeColor','blue');
hold on
realPlot = plot(realPositions(:,1), realPositions(:,2), 'o', '
    MarkerEdgeColor','green');
hold on
legend([fixedPlot, estimPlot, realPlot, errorPlot],{'Nodo ancla', '
    Posición estimada del nodo móvil', 'Posición real del nodo móvil','
    Distancia de error'})
ylim([-3,17]);
xlim([-3,14]);
xlabel('X (m)');
ylabel('Y (m)');
title('Método de no identificación, toda \beta = 1 (grid)');
grid on
grid minor
end

```

B.2 Algoritmo con conocimiento perfecto de la condición de visión

Código B.3 Algoritmo con conocimiento perfecto y *fminsearch*.

```

function errors = WLSLocalizationConConocimientoPerfecto(path, betaNLOS)
realPositions=(load(strcat(path,'/realPositions.txt'))).*0.5;
fixedNodesPosition=(load(strcat(path,'/fixedNodesPosition.txt'))).*0.5;
measuredDistance=load(strcat(path,'/measuredDistance.txt'));
sightCondition=load(strcat(path,'/sightCondition.txt'));
N=length(fixedNodesPosition(:,1)); %Número de nodos ancla
M=length(measuredDistance(:,1)); %Número de posiciones medidas
beta = ones(M,N);
estimPositions = ones(M,2);
errors=ones(1,M);
betaLOS = 1;

```

```

for k=1:1:M %Bucle para para cada medida
    for j=1:1:N
        if sightCondition(k,j) == 1 %Si la medida es LOS (Line of Sight)
            beta(k,j) = 1;
        elseif sightCondition(k,j) == 0 %Si la medida es NLOS (Non-Line
            of Sight)
            beta(k,j) = betaNLOS;
        else
            beta(k,j) = 0;
        end
    end
    funIndiv = @(x,y) 0;
    funTotal = @(x,y) 0;
    for i=1:1:N %Bucle para cada nodo ancla
        funIndiv = @(x,y) beta(k,i)*(measuredDistance(k,i)-norm([x,y]-
            fixedNodesPosition(i,:)))^2;
        funTotal = @(x,y) funTotal(x,y) + funIndiv(x,y);
    end
    x0=[0,0];
    [xmin, fval] = fminsearch(@(x)funTotal(x(1),x(2)), x0); %Evaluación
        de la función
    estimPositions(k,:) = xmin;
    errors(k)=norm(realPositions(k,:)-estimPositions(k,:));
    errorPlot = plot([realPositions(k,1) estimPositions(k,1)], [
        realPositions(k,2) estimPositions(k,2)], 'k');
    hold on
end
fixedPlot = plot(fixedNodesPosition(:,1),fixedNodesPosition(:,2),'+',
    MarkerEdgeColor','red');
hold on
estimPlot = plot(estimPositions(:,1), estimPositions(:,2), 'x',
    MarkerEdgeColor','blue');
hold on
realPlot = plot(realPositions(:,1), realPositions(:,2), 'o',
    MarkerEdgeColor','green');
hold on
ylim([-3,17]);
xlim([-3,14]);
grid on
grid minor
legend([fixedPlot, estimPlot, realPlot, errorPlot],{'Nodo ancla',
    'Posición estimada del nodo móvil', 'Posición real del nodo móvil',
    'Distancia de error'})
xlabel('X (m)');
ylabel('Y (m)');
title(['Conocimiento perfecto con \beta NLOS = ' num2str(betaNLOS) ' (
    fminsearch)'])
end

```

Código B.4 Algoritmo con conocimiento perfecto y *grid*.

```

function errors = WLSLocalizationConConocimientoPerfectoyGrid(path,
    betaNLOS)
limx = [-3 20];
limy = [-3 20];
separator = 0.25;
realPositions=(load(strcat(path,'/realPositions.txt'))).*0.5;
fixedNodesPosition=(load(strcat(path,'/fixedNodesPosition.txt'))).*0.5;
measuredDistance=load(strcat(path,'/measuredDistance.txt'));
sightCondition=load(strcat(path,'/sightCondition.txt'));
N=length(fixedNodesPosition(:,1)); %Número de nodos ancla
M=length(measuredDistance(:,1)); %Número de posiciones medidas
beta = ones(M,N);
estimPositions = ones(M,2);
errors=ones(1,M);
betaLOS = 1;
for k=1:1:M %Bucle para para cada medida
    for j=1:1:N
        if sightCondition(k,j) == 1 %Si la medida es LOS (Line of Sight)
            beta(k,j) = 1;
        elseif sightCondition(k,j) == 0 %Si la medida es NLOS (Non-Line
            of Sight)
            beta(k,j) = betaNLOS;
        else
            beta(k,j) = 0;
        end
    end
    funIndiv = @(x,y) 0;
    funTotal = @(x,y) 0;
    for i=1:1:N %Bucle para cada nodo ancla
        funIndiv = @(x,y) beta(k,i)*(measuredDistance(k,i)-norm([x,y]-
            fixedNodesPosition(i,:)))^2;
        funTotal = @(x,y) funTotal(x,y) + funIndiv(x,y);
    end
    fmin = inf;
    for xx=limx(1):separator:limx(2)
        for yy=limy(1):separator:limy(2)
            fval = funTotal(xx,yy); %Evaluación de la función
            if fval < fmin
                fmin = fval;
                estimPositions(k,:) = [xx,yy];
            end
        end
    end
    errors(k)=norm(realPositions(k,:)-estimPositions(k,:));
    errorPlot = plot([realPositions(k,1) estimPositions(k,1)], [
        realPositions(k,2) estimPositions(k,2)], 'k');
    hold on
end

```

```

fixedPlot = plot(fixedNodesPosition(:,1),fixedNodesPosition(:,2),'+', '
    MarkerEdgeColor','red');
hold on
estimPlot = plot(estimPositions(:,1), estimPositions(:,2), 'x', '
    MarkerEdgeColor','blue');
hold on
realPlot = plot(realPositions(:,1), realPositions(:,2), 'o', '
    MarkerEdgeColor','green');
hold on
ylim([-3,17]);
xlim([-3,14]);
grid on
grid minor
legend([fixedPlot, estimPlot, realPlot, errorPlot],{'Nodo ancla', '
    Posición estimada del nodo móvil', 'Posición real del nodo móvil', '
    Distancia de error'})
xlabel('X (m)');
ylabel('Y (m)');
title(['Conocimiento perfecto con \beta NLOS = ' num2str(betaNLOS) ' (
    grid)'])

end

```

B.3 Algoritmo basado en la varianza de las medidas

Código B.5 Algoritmo basado en la varianza de las medidas y *fminsearch*.

```

function errors = WLSLocalizationConVarianza(path)
realPositions=(load(strcat(path,'/realPositions.txt'))).*0.5;
fixedNodesPosition=(load(strcat(path,'/fixedNodesPosition.txt'))).*0.5;
measuredDistance=load(strcat(path,'/measuredDistance.txt'));
stdDev=load(strcat(path,'/stdDev.txt'));
N=length(fixedNodesPosition(:,1)); %Número de nodos ancla
M=length(measuredDistance(:,1)); %Número de posiciones medidas
beta=ones(1,N);
estimPositions = ones(M,2);
errors=ones(1,M);
for k=1:1:M %Bucle para para cada medida
    funIndiv = @(x,y) 0;
    funTotal = @(x,y) 0;
    for j=1:1:N
        if stdDev(k,j) == 0 %Si la desviación típica es 0, significa
            que no había comunicación entre nodo y punto.
            beta(j) = 0;
        else
            beta(j) = 1/((stdDev(k,j)/100)^2);
        end
    end
end
end

```

```

for i=1:1:N %Bucle para cada nodo ancla
    funIndiv = @(x,y) beta(i)*(measuredDistance(k,i)-norm([x,y]-
        fixedNodesPosition(i,:)))^2;
    funTotal = @(x,y) funTotal(x,y) + funIndiv(x,y);
end
x0=[0,0];
[xmin, fval] = fminsearch(@(x)funTotal(x(1),x(2)), x0); %Evaluación
    de la función
estimPositions(k,:) = xmin;
errors(k)=norm(realPositions(k,:)-estimPositions(k,:));
errorPlot = plot([realPositions(k,1) estimPositions(k,1)], [
    realPositions(k,2) estimPositions(k,2)], 'k');
hold on
end
fixedPlot = plot(fixedNodesPosition(:,1),fixedNodesPosition(:,2),'+', '
    MarkerEdgeColor','red');
hold on
estimPlot = plot(estimPositions(:,1), estimPositions(:,2), 'x', '
    MarkerEdgeColor','blue');
hold on
realPlot = plot(realPositions(:,1), realPositions(:,2), 'o', '
    MarkerEdgeColor','green');
hold on
legend([fixedPlot, estimPlot, realPlot, errorPlot],{'Nodo ancla', '
    Posición estimada del nodo móvil', 'Posición real del nodo móvil', '
    Distancia de error'})
ylim([-3,17]);
xlim([-3,14]);
xlabel('X (m)');
ylabel('Y (m)');
title('Algoritmo basado en la varianza de las medidas (fminsearch)');
grid on
grid minor
end

```

Código B.6 Algoritmo basado en la varianza de las medidas y *grid*.

```

function errors = WLSLocalizationConVarianzayGrid(path)
limx = [-3 20];
limy = [-3 20];
separator = 0.25;
realPositions=(load(strcat(path,'/realPositions.txt'))).*0.5;
fixedNodesPosition=(load(strcat(path,'/fixedNodesPosition.txt'))).*0.5;
measuredDistance=load(strcat(path,'/measuredDistance.txt'));
stdDev=load(strcat(path,'/stdDev.txt'));
N=length(fixedNodesPosition(:,1)); %Número de nodos ancla
M=length(measuredDistance(:,1)); %Número de posiciones medidas
beta=ones(1,N);
estimPositions = ones(M,2);

```

```

errors=ones(1,M);
for k=1:1:M %Bucle para para cada medida
    funIndiv = @(x,y) 0;
    funTotal = @(x,y) 0;
    for j=1:1:N
        if stdDev(k,j) == 0 %Si la desviación típica es 0, significa que
            no había comunicación entre nodo y punto.
            beta(j) = 0;
        else
            beta(j) = 1/((stdDev(k,j)/100)^2);
        end
    end
    for i=1:1:N %Bucle para cada nodo ancla
        funIndiv = @(x,y) beta(i)*(measuredDistance(k,i)-norm([x,y]-
            fixedNodesPosition(i,:)))^2;
        funTotal = @(x,y) funTotal(x,y) + funIndiv(x,y);
    end
    fmin = inf;
    for xx=limx(1):separator:limx(2)
        for yy=limy(1):separator:limy(2)
            fval = funTotal(xx,yy); %Evaluación de la función
            if fval < fmin
                fmin = fval;
                estimPositions(k,:) = [xx,yy];
            end
        end
    end
    errors(k)=norm(realPositions(k,:)-estimPositions(k,:));
    errorPlot = plot([realPositions(k,1) estimPositions(k,1)], [
        realPositions(k,2) estimPositions(k,2)],'k');
    hold on
end
fixedPlot = plot(fixedNodesPosition(:,1),fixedNodesPosition(:,2),'+', '
    MarkerEdgeColor','red');
hold on
estimPlot = plot(estimPositions(:,1), estimPositions(:,2), 'x', '
    MarkerEdgeColor','blue');
hold on
realPlot = plot(realPositions(:,1), realPositions(:,2), 'o', '
    MarkerEdgeColor','green');
hold on
legend([fixedPlot, estimPlot, realPlot, errorPlot],{'Nodo ancla', '
    Posición estimada del nodo móvil', 'Posición real del nodo móvil', '
    Distancia de error'})
title('Algoritmo basado en la varianza de las medidas (grid)');
ylim([-3,17]);
xlim([-3,14]);
xlabel('X (m)');
ylabel('Y (m)');
grid on

```

```
grid minor
end
```

B.4 Algoritmo basado en estadísticos de la señal recibida

Código B.7 Algoritmo basado en estadísticos de la señal recibida y *fminsearch*.

```
function errors = WLSLocalizationConProbabilidades(path, betaNLOS)
realPositions=(load(strcat(path,'/realPositions.txt'))).*0.5;
fixedNodesPosition=(load(strcat(path,'/fixedNodesPosition.txt'))).*0.5;
measuredDistance=load(strcat(path,'/measuredDistance.txt'));
sightCondition=load(strcat(path,'/sightCondition.txt'));
N=length(fixedNodesPosition(:,1)); %Número de nodos ancla
M=length(measuredDistance(:,1)); %Número de posiciones medidas
beta = ones(M,N);
estimPositions = ones(M,2);
errors=ones(1,M);
betaLOS = 1;
for k=1:1:M %Bucle para para cada medida
    for j=1:1:N
        num = rand;
        if sightCondition(k,j) == 1 %Si la medida es LOS (Line of Sight)
            if num <= 0.979
                beta(k,j) = betaLOS; %Identificación correcta
            else
                beta(k,j) = betaNLOS; %Identificación incorrecta
            end
        elseif sightCondition(k,j) == 0 %Si la medida es NLOS (Non-Line
            of Sight)
            if num <= 0.959
                beta(k,j) = betaNLOS; %Identificación correcta
            else
                beta(k,j) = betaLOS; %Identificación incorrecta
            end
        else
            beta(k,j) = 0;
        end
    end
end
funIndiv = @(x,y) 0;
funTotal = @(x,y) 0;
for i=1:1:N %Bucle para cada nodo ancla
    funIndiv = @(x,y) beta(k,i)*(measuredDistance(k,i)-norm([x,y]-
        fixedNodesPosition(i,:)))^2;
    funTotal = @(x,y) funTotal(x,y) + funIndiv(x,y);
end
x0=[0,0];
[xmin, fval] = fminsearch(@(x)funTotal(x(1),x(2)), x0); %Evaluación
de la función
```

```

estimPositions(k,:) = xmin;
errors(k)=norm(realPositions(k,:)-estimPositions(k,:));
errorPlot = plot([realPositions(k,1) estimPositions(k,1)], [
    realPositions(k,2) estimPositions(k,2)], 'k');
hold on
end
fixedPlot = plot(fixedNodesPosition(:,1),fixedNodesPosition(:,2),'+', '
    MarkerEdgeColor','red');
hold on
estimPlot = plot(estimPositions(:,1), estimPositions(:,2), 'x', '
    MarkerEdgeColor','blue');
hold on
realPlot = plot(realPositions(:,1), realPositions(:,2), 'o', '
    MarkerEdgeColor','green');
hold on
ylim([-3,17]);
xlim([-3,14]);
grid on
grid minor
legend([fixedPlot, estimPlot, realPlot, errorPlot],{'Nodo ancla', '
    Posición estimada del nodo móvil', 'Posición real del nodo móvil', '
    Distancia de error'})
xlabel('X (m)');
ylabel('Y (m)');
title(['Algoritmo basado en estadísticos de la señal recibida con \beta
    NLOS = ' num2str(betaNLOS) ' (fminsearch)'])

end

```

Código B.8 Algoritmo basado en estadísticos de la señal recibida y *grid*.

```

function errors = WLSLocalizationConProbabilidadesyGrid(path, betaNLOS)
limx = [-3 20];
limy = [-3 20];
separator = 0.25;
realPositions=(load(strcat(path,'/realPositions.txt'))).*0.5;
fixedNodesPosition=(load(strcat(path,'/fixedNodesPosition.txt'))).*0.5;
measuredDistance=load(strcat(path,'/measuredDistance.txt'));
sightCondition=load(strcat(path,'/sightCondition.txt'));
N=length(fixedNodesPosition(:,1)); %Número de nodos ancla
M=length(measuredDistance(:,1)); %Número de posiciones medidas
beta = ones(M,N);
estimPositions = ones(M,2);
errors=ones(1,M);
betaLOS = 1;
for k=1:1:M %Bucle para para cada medida
    for j=1:1:N
        num = rand;
        if sightCondition(k,j) == 1 %Si la medida es LOS (Line of Sight)

```



```

    if num <= 0.979
        beta(k,j) = betaLOS; %Identificación correcta
    else
        beta(k,j) = betaNLOS; %Identificación incorrecta
    end
elseif sightCondition(k,j) == 0 %Si la medida es NLOS (Non-Line
of Sight)
    if num <= 0.959
        beta(k,j) = betaNLOS; %Identificación correcta
    else
        beta(k,j) = betaLOS; %Identificación incorrecta
    end
else
    beta(k,j) = 0;
end
end
funIndiv = @(x,y) 0;
funTotal = @(x,y) 0;
for i=1:1:N %Bucle para cada nodo ancla
    funIndiv = @(x,y) beta(k,i)*(measuredDistance(k,i)-norm([x,y]-
        fixedNodesPosition(i,:)))^2;
    funTotal = @(x,y) funTotal(x,y) + funIndiv(x,y);
end
fmin = inf;
for xx=limx(1):separator:limx(2)
    for yy=limy(1):separator:limy(2)
        fval = funTotal(xx,yy); %Evaluación de la función
        if fval < fmin
            fmin = fval;
            estimPositions(k,:) = [xx,yy];
        end
    end
end
end
errors(k)=norm(realPositions(k,:)-estimPositions(k,:));
errorPlot = plot([realPositions(k,1) estimPositions(k,1)], [
    realPositions(k,2) estimPositions(k,2)], 'k');
hold on
end
fixedPlot = plot(fixedNodesPosition(:,1),fixedNodesPosition(:,2),'+', '
    MarkerEdgeColor','red');
hold on
estimPlot = plot(estimPositions(:,1), estimPositions(:,2), 'x', '
    MarkerEdgeColor','blue');
hold on
realPlot = plot(realPositions(:,1), realPositions(:,2), 'o', '
    MarkerEdgeColor','green');
hold on
ylim([-3,17]);
xlim([-3,14]);
grid on

```

```

grid minor
legend([fixedPlot, estimPlot, realPlot, errorPlot],{'Nodo ancla', '
    Posición estimada del nodo móvil', 'Posición real del nodo móvil','
    Distancia de error'})
xlabel('X (m)');
ylabel('Y (m)');
title(['Algoritmo basado en estadísticos de la señal recibida con \beta
    NLOS = ' num2str(betaNLOS) ' (grid)'])
end

```

B.5 Ejecución de algoritmos

Código B.9 Ejecución de algoritmos.

```

%Algoritmos de referencia
figure(1)
subplot(1,2,1);
errorsRef1 = WLSLocalizationConTodasLasMedidas('Datos Reales');
rmseRef1=sqrt((sum(errorsRef1.^2))/length(errorsRef1));
subplot(1,2,2);
errorsRef2 = WLSLocalizationConTodasLasMedidasyGrid('Datos Reales');
rmseRef2=sqrt((sum(errorsRef2.^2))/length(errorsRef2));
%%
figure(2)
subplot(1,2,1);
errorsRef3 = WLSLocalizationConConocimientoPerfecto('Datos Reales', 0);
rmseRef3=sqrt((sum(errorsRef3.^2))/length(errorsRef3));
subplot(1,2,2);
errorsRef4 = WLSLocalizationConConocimientoPerfectoyGrid('Datos Reales',
    0);
rmseRef4=sqrt((sum(errorsRef4.^2))/length(errorsRef4));
%%
figure(3)
subplot(1,2,1);
errorsRef5 = WLSLocalizationConConocimientoPerfecto('Datos Reales',
    0.001);
rmseRef5=sqrt((sum(errorsRef5.^2))/length(errorsRef5));
subplot(1,2,2);
errorsRef6 = WLSLocalizationConConocimientoPerfectoyGrid('Datos Reales',
    0.001);
rmseRef6=sqrt((sum(errorsRef6.^2))/length(errorsRef6));
%%
%Algoritmo basado en la varianza de las medidas
figure(4)
subplot(2,2,1)
errorsVar1 = WLSLocalizationConVarianza('Datos Reales');
rmseVar1=sqrt((sum(errorsVar1.^2))/length(errorsVar1));
subplot(2,2,2)

```

```

errorsVar2 = WLSLocalizationConVarianzayGrid('Datos Reales');
rmseVar2=sqrt((sum(errorsVar2.^2))/length(errorsVar2));
subplot(2,2,3)
errorsVar3 = WLSLocalizationConTodasLasMedidasGrid('Datos Reales');
rmseVar3=sqrt((sum(errorsVar3.^2))/length(errorsVar3));
subplot(2,2,4)
errorsVar4 = WLSLocalizationConConocimientoPerfectoyGrid('Datos Reales',
    0.001);
rmseVar4=sqrt((sum(errorsVar4.^2))/length(errorsVar4));
%%
%%Algoritmo basado en estadísticos de la señal recibida con beta NLOS =
    0.8
figure(5)
subplot(2,2,1)
errorsProb81 = WLSLocalizationConProbabilidades('Datos Reales',0.8);
rmseProb81=sqrt((sum(errorsProb81.^2))/length(errorsProb81));
subplot(2,2,2)
errorsProb82 = WLSLocalizationConProbabilidadesyGrid('Datos Reales',0.8)
    ;
rmseProb82=sqrt((sum(errorsProb82.^2))/length(errorsProb82));
subplot(2,2,3)
errorsProb83 = WLSLocalizationConTodasLasMedidasGrid('Datos Reales');
rmseProb83=sqrt((sum(errorsProb83.^2))/length(errorsProb83));
subplot(2,2,4)
errorsProb84 = WLSLocalizationConConocimientoPerfectoyGrid('Datos Reales
    ', 0.001);
rmseProb84=sqrt((sum(errorsProb84.^2))/length(errorsProb84));
%%
%%Algoritmo basado en estadísticos de la señal recibida con beta NLOS =
    0.4
figure(6)
subplot(2,2,1)
errorsProb41 = WLSLocalizationConProbabilidades('Datos Reales',0.4);
rmseProb41=sqrt((sum(errorsProb41.^2))/length(errorsProb41));
subplot(2,2,2)
errorsProb42 = WLSLocalizationConProbabilidadesyGrid('Datos Reales',0.4)
    ;
rmseProb42=sqrt((sum(errorsProb42.^2))/length(errorsProb42));
subplot(2,2,3)
errorsProb43 = WLSLocalizationConTodasLasMedidasGrid('Datos Reales');
rmseProb43=sqrt((sum(errorsProb43.^2))/length(errorsProb43));
subplot(2,2,4)
errorsProb44 = WLSLocalizationConConocimientoPerfectoyGrid('Datos Reales
    ', 0.001);
rmseProb44=sqrt((sum(errorsProb44.^2))/length(errorsProb44));
%%
%%Algoritmo basado en estadísticos de la señal recibida con beta NLOS
    =0.001
figure(7)
subplot(2,2,1)

```

```

errorsProb11 = WLSLocalizationConProbabilidades('Datos Reales',0.001);
rmseProb11=sqrt((sum(errorsProb11.^2))/length(errorsProb11));
subplot(2,2,2)
errorsProb12 = WLSLocalizationConProbabilidadesyGrid('Datos Reales'
,0.001);
rmseProb12=sqrt((sum(errorsProb12.^2))/length(errorsProb12));
subplot(2,2,3)
errorsProb13 = WLSLocalizationConTodasLasMedidasyGrid('Datos Reales');
rmseProb13=sqrt((sum(errorsProb13.^2))/length(errorsProb13));
subplot(2,2,4)
errorsProb14 = WLSLocalizationConConocimientoPerfectoyGrid('Datos Reales
', 0.001);
rmseProb14=sqrt((sum(errorsProb14.^2))/length(errorsProb14));

```

Código B.10 Bucle de ejecuciones del algoritmo basado en estadísticos de la señal recibida.

```

%% Bucle de ejecuciones para el algoritmo basado en estadísticos de la
    señal recibida
numEjec = 500;
%% Caso beta NLOS = 0.8 y fminsearch
errors81 = ones(57, numEjec);
for i=1:1:numEjec
errors81(:,i) = WLSLocalizationConProbabilidades('Datos Reales',0.8);
end
rmse81=sqrt(sum(sum(errors81.^2))/(57*numEjec));
%% Caso beta NLOS = 0.8 y grid
errors82 = ones(57, numEjec);
for i=1:1:numEjec
errors82(:,i) = WLSLocalizationConProbabilidadesyGrid('Datos Reales'
,0.8);
end
rmse82=sqrt(sum(sum(errors82.^2))/(57*numEjec));
%% Caso beta NLOS = 0.4 y fminsearch
errors41 = ones(57, numEjec);
for i=1:1:numEjec
errors41(:,i) = WLSLocalizationConProbabilidades('Datos Reales',0.4);
end
rmse41=sqrt(sum(sum(errors41.^2))/(57*numEjec));
%% Caso beta NLOS = 0.4 y grid
errors42 = ones(57, numEjec);
for i=1:1:numEjec
errors42(:,i) = WLSLocalizationConProbabilidadesyGrid('Datos Reales'
,0.4);
end
rmse42=sqrt(sum(sum(errors42.^2))/(57*numEjec));
%% Caso beta NLOS = 0.001 y fminsearch
errors11 = ones(57, numEjec);
for i=1:1:numEjec
errors11(:,i) = WLSLocalizationConProbabilidades('Datos Reales',0.001);

```

```

end
rmse11=sqrt(sum(sum(errors11.^2))/(57*numEjec));
%% Caso beta NLOS = 0.001 y grid
errors12 = ones(57, numEjec);
for i=1:1:numEjec
errors12(:,i) = WLSLocalizationConProbabilidadesyGrid('Datos Reales'
,0.001);
end
rmse12=sqrt(sum(sum(errors12.^2))/(57*numEjec));

```

B.6 Histograma de comparación

Código B.11 Creación del histograma de comparación de RMSE según el algoritmo utilizado.

```

rmse = [1.5088,2.9848,0.8411,2.6414,1.4449,1.3277,1.0428;
        1.1002,1.6978,0.7716,0.9424,1.0844,1.0447,0.8408];
figure(1)
h = bar(rmse);
grid on
grid minor
l = cell(1,7);
l{1}='Método de no identificación';
l{2}='Conocimiento perfecto con \beta NLOS = 0';
l{3}='Conocimiento perfecto con \beta NLOS = 0.001';
l{4}='Algoritmo basado en la varianza de las medidas';
l{5}='Algoritmo basado en estadísticos de la señal recibida con \beta
NLOS = 0.8';
l{6}='Algoritmo basado en estadísticos de la señal recibida con \beta
NLOS = 0.4';
l{7}='Algoritmo basado en estadísticos de la señal recibida con \beta
NLOS = 0.001';
legend(h,l,'Location', 'northeast');
set(gca,'XTickLabel',{ 'Minimización con fminsearch', 'Minimización con
grid'});
ylabel('RMSE (Raíz del error cuadrático medio) (metros)');
title('Comparación de RMSE según el algoritmo utilizado');

```


Apéndice C

Ficheros de datos

En este apéndice se muestran los datos obtenidos durante la fase experimental del proyecto, posteriormente utilizados para la ejecución de los códigos del apéndice anterior.

C.1 Posiciones reales del nodo móvil

El primer fichero de texto introducido en Matlab, *realPositions.txt* está formado por los valores (x,y) de las posiciones reales del nodo móvil, donde la distancia ha sido medida hasta él desde los cinco nodos ancla. Estos valores están escalados por las losas del suelo, es decir, $x = 1$ corresponde a $x = 0.5$ en la escala real, pues la dimensión de la losa es de 0.5 m de lado. Se hizo de esta forma para tener una mayor comodidad a la hora de tomar datos en el Departamento, sin embargo como se observa en los códigos de Matlab, estos datos son transformados a la escala real. El contenido del fichero es el siguiente:

1,1
1,3
1,5
1,7
1,9
1,11
1,13
1,15
1,17
1,19
1,21
1,23
1,25
1,27
1,29
1,31
4,2
6,2
8,2
10,2
12,2
14,2

16,2
18,2
20,2
22,2
24,2
24,1
26,2
4,4
4,5
5,6
5,7
5,8
6,6
6,7
6,8
7,5
7,6
7,7
7,8
8,4
8,5
8,6
8,7
9,5
9,6
9,7
9,8
10,5
10,6
10,7
10,8
11,5
11,6
11,7
11,8

C.2 Posiciones de los nodos ancla

Este fichero *fixedNodesPosition.txt* incluye la posición (x,y) de cada uno de los nodos ancla, de forma que en los siguientes ficheros donde se incluyen las medidas se sigue el orden de nodos ancla introducidos a continuación, habiendo tantos nodos como filas tenga este fichero. Estos valores están escalados de la misma forma que el fichero anterior (*realPositions.txt*).

1,1
1,15
11,8
26,2
5,8

C.3 Distancias medidas por cada nodo ancla

Pasamos a los ficheros con medidas, siendo el contenido de *measuredDistance.txt* la distancia medida en metros por cada nodo ancla, habiendo tantas filas como posiciones reales del nodo móvil haya, y tantos valores (separados por coma) por fila como nodos ancla usemos, correspondiéndose el orden de los valores de cada fila con el orden de las filas del fichero anterior (*fixedNodesPosition.txt*). Cuando una distancia es igual a 0, significa que el nodo ancla no llegaba a tener comunicación con el nodo móvil o que la posición del nodo móvil coincidía con la del nodo ancla; en definitiva, cuando el valor es 0, la medida es descartada. Esta observación se tiene en cuenta en el código de los algoritmos de localización (excepto en casos donde no afecta de todas formas).

0, 6.765, 6.363, 12.267, 4.150
 0.720, 5.653, 5.412, 12.599, 4.019
 1.810, 4.697, 5.488, 13.008, 4.105
 2.768, 3.812, 6.005, 0, 3.950
 3.702, 2.742, 6.535, 0, 2.867
 4.742, 1.775, 5.776, 0, 3.209
 5.749, 0.725, 5.892, 0, 3.368
 6.847, 0, 6.540, 0, 4.321
 7.982, 0.778, 0, 0, 5.490
 8.888, 1.802, 0, 0, 6.381
 9.671, 2.733, 0, 0, 7.514
 10.714, 3.845, 0, 0, 8.298
 11.726, 4.682, 0, 0, 9.104
 12.703, 5.796, 0, 0, 10.656
 13.684, 6.986, 0, 0, 11.866
 14.710, 8.090, 0, 0, 17.253
 1.448, 6.890, 4.940, 10.764, 2.750
 2.340, 7.765, 4.337, 9.779, 3.134
 3.380, 8.305, 3.479, 8.895, 3.842
 4.396, 9.484, 3.272, 7.849, 4.304
 5.335, 9.484, 3.261, 6.863, 4.866
 6.323, 9.941, 3.678, 5.803, 5.752
 7.307, 13.319, 4.326, 4.861, 6.524
 8.308, 0, 4.764, 3.876, 7.332
 9.477, 0, 6.003, 2.756, 8.027
 10.526, 0, 6.718, 1.833, 9.710
 11.302, 0, 8.360, 0.800, 10.512
 11.479, 0, 8.655, 1.016, 10.158
 12.223, 0, 8.891, 0, 12.007
 1.882, 5.938, 4.012, 12.783, 1.805
 2.298, 6.096, 3.640, 12.077, 1.315
 2.873, 5.147, 2.877, 11.036, 0.780
 3.505, 4.906, 2.835, 11.666, 0.238
 4.156, 4.074, 2.856, 11.193, 0
 3.305, 5.436, 2.465, 10.527, 0.956
 3.641, 4.882, 2.316, 10.612, 0.474
 4.084, 4.548, 2.280, 10.984, 0.278
 3.741, 6.398, 2.317, 10.079, 1.693
 3.960, 5.602, 2.058, 10.163, 1.306

4.015, 5.303, 1.944, 10.253, 0.957
4.383, 5.056, 1.843, 10.513, 0.895
4.125, 6.783, 2.603, 9.562, 2.330
4.166, 6.431, 1.936, 9.887, 2.014
4.363, 6.090, 1.640, 9.677, 1.712
4.607, 5.613, 1.398, 10.068, 1.400
4.710, 6.733, 1.634, 9.522, 2.380
4.817, 6.560, 1.239, 8.930, 2.121
5.046, 6.014, 0.945, 9.407, 1.909
5.301, 5.772, 0.783, 9.865, 1.958
5.182, 7.069, 1.452, 8.307, 2.790
5.272, 6.721, 0.974, 8.522, 2.472
5.425, 6.514, 0.491, 8.722, 2.383
5.779, 6.301, 0.239, 9.522, 2.411
6.312, 7.454, 1.251, 8.188, 3.256
5.861, 7.282, 0.786, 8.137, 2.980
6.517, 6.910, 0.222, 8.912, 2.900
6.301, 6.793, 0, 8.942, 2.893

C.4 Desviación típica

En el fichero *stdDev.txt* encontraremos la desviación típica calculada por cada nodo ancla. Cuando el valor es 0, significa que el nodo ancla no llegaba a tener comunicación con el nodo móvil o que la posición del nodo móvil coincidía con la del nodo ancla. Esta observación es considerada en el código de los algoritmos de localización, descartando la medida cuando el valor de la desviación típica es 0.

0, 2.78, 1.91, 1.83, 1.94
2.06, 2.26, 2.22, 4.94, 2.28
2.95, 2.44, 1.97, 2.20, 22.38
2.55, 1.86, 26.36, 0, 5.10
2.19, 1.87, 10.25, 0, 1.57
2.19, 1.67, 3.05, 0, 2.92
1.88, 2.85, 3.65, 0, 2.11
1.93, 0, 3.58, 0, 2.42
18.44, 3.06, 0, 0, 2.36
2.48, 2.65, 0, 0, 2.82
1.46, 2.93, 0, 0, 3.37
2.94, 1.48, 0, 0, 9.99
2.15, 2.68, 0, 0, 39.73
1.91, 2.10, 0, 0, 45.77
1.72, 2.90, 0, 0, 7.17
1.82, 2.10, 0, 0, 29.08
2.34, 3.24, 2.51, 1.96, 2.12
1.94, 16.40, 1.45, 2.14, 1.51
1.45, 7.28, 1.86, 2.74, 1.24
2.01, 14.34, 1.73, 1.58, 2.49
1.86, 25.08, 1.94, 2.52, 2.20
2.00, 18.18, 1.37, 3.51, 3.47

1.89, 0, 2.07, 2.33, 2.98
 2.75, 0, 3.50, 1.96, 2.65
 2.69, 0, 8.91, 2.27, 13.77
 2.15, 0, 9.83, 3.00, 2.70
 1.73, 0, 4.45, 3.37, 10.98
 1.66, 0, 18.83, 2.81, 3.92
 1.84, 0, 3.85, 0, 27.21
 1.90, 2.37, 1.55, 2.83, 2.12
 2.93, 10.78, 2.94, 11.10, 2.34
 2.09, 3.48, 3.02, 3.31, 3.71
 1.95, 3.05, 2.32, 17.57, 4.47
 2.37, 3.49, 2.42, 2.80, 0
 2.53, 2.41, 2.70, 21.82, 2.63
 2.55, 2.99, 2.37, 3.59, 2.77
 1.77, 2.43, 2.31, 27.54, 2.77
 11.40, 2.46, 3.82, 5.24, 2.45
 2.94, 2.15, 3.47, 18.06, 2.83
 3.02, 1.38, 2.47, 4.92, 3.34
 1.69, 2.79, 2.63, 3.73, 3.61
 5.02, 2.26, 1.32, 5.15, 1.99
 2.50, 1.71, 3.08, 4.24, 2.92
 1.38, 2.66, 3.37, 36.34, 2.86
 2.82, 2.03, 2.10, 4.10, 2.50
 2.16, 2.33, 2.93, 2.50, 3.24
 1.13, 18.47, 2.11, 5.34, 2.40
 1.25, 1.97, 1.99, 15.96, 1.53
 1.80, 12.59, 1.90, 8.57, 1.85
 10.92, 2.29, 1.89, 3.25, 2.86
 1.63, 2.24, 3.01, 11.76, 2.73
 1.99, 1.76, 3.18, 4.02, 2.39
 2.40, 29.34, 4.23, 2.28, 1.84
 2.66, 2.15, 4.40, 34.57, 2.02
 1.89, 33.79, 2.10, 3.13, 2.62
 1.56, 2.17, 5.74, 12.10, 2.73
 2.13, 7.43, 0, 2.32, 1.70

C.5 Condición de línea de visión

En este último fichero de texto, *sightCondition.txt*, se encuentran las condiciones de línea de visión (LOS/NLOS) de cada medida tomada, siendo 1 para LOS, 0 para NLOS y -1 cuando no existía comunicación entre el nodo ancla y el nodo móvil o cuando la posición de ambos nodos coincidía, descartando la medida en este caso.

-1, 1, 0, 1, 0
 1, 1, 1, 1, 0
 1, 1, 0, 0, 0
 1, 1, 0, -1, 0
 1, 1, 0, -1, 0
 1, 1, 0, -1, 0

0, 0, 1, 0, 1

Índice de Figuras

2.1	Señal Ultra Wide-Band en el dominio de la frecuencia	4
2.2	Máscara espectral del CNAF para el uso de UWB en España	5
2.3	Ejemplo de un sistema de localización con tres nodos ancla y un nodo móvil	7
4.1	Distribución de nodos ancla y puntos en la sala y pasillo	16
4.2	Distribución de posiciones estimadas sin mitigación	18
4.3	Distribución de posiciones estimadas con conocimiento perfecto de la condición LOS/N-LOS y $\beta_{NLOS} = 0$	19
4.4	Distribución de posiciones estimadas con conocimiento perfecto de la condición LOS/N-LOS y $\beta_{NLOS} = 0.001$	19
4.5	Representación en 3D del valor de la función en el punto (3,4)	20
4.6	Representación en 2D del valor de la función en el punto (3,4)	20
4.7	Distribución de posiciones estimadas con el algoritmo basado en la varianza de las medidas	21
4.8	Distribución de posiciones estimadas con algoritmo basado en estadísticos de la señal recibida y $\beta_{NLOS} = 0.8$	22
4.9	Distribución de posiciones estimadas con algoritmo basado en estadísticos de la señal recibida y $\beta_{NLOS} = 0.4$	23
4.10	Distribución de posiciones estimadas con algoritmo basado en estadísticos de la señal recibida y $\beta_{NLOS} = 0.001$	23
4.11	Comparación de RMSE según el algoritmo utilizado	26
A.1	Configuración del nodo ancla	30
A.2	Proceso de configuración del equipo	31
A.3	Proceso de medición de distancia en el pasillo del departamento	32

Índice de Tablas

2.1	Máscara espectral del CNAF para el uso de UWB en España	5
4.1	Tabla de valores RMSE (metros) según algoritmo utilizado	25

Índice de Códigos

B.1	Algoritmo sin identificación y <i>fminsearch</i>	33
B.2	Algoritmo sin identificación y <i>grid</i>	34
B.3	Algoritmo con conocimiento perfecto y <i>fminsearch</i>	35
B.4	Algoritmo con conocimiento perfecto y <i>grid</i>	37
B.5	Algoritmo basado en la varianza de las medidas y <i>fminsearch</i>	38
B.6	Algoritmo basado en la varianza de las medidas y <i>grid</i>	39
B.7	Algoritmo basado en estadísticos de la señal recibida y <i>fminsearch</i>	41
B.8	Algoritmo basado en estadísticos de la señal recibida y <i>grid</i>	42
B.9	Ejecución de algoritmos	44
B.10	Bucle de ejecuciones del algoritmo basado en estadísticos de la señal recibida	46
B.11	Creación del histograma de comparación de RMSE según el algoritmo utilizado	47

Bibliografía

- [AGI] AGILENT, *Ultra-Wideband Communication RF Measurements*. dirección: <https://web.archive.org/web/20090117090424/http://cp.literature.agilent.com/litweb/pdf/5989-0506EN.pdf>.
- [CC01] Á. G. Caffaro y M. G. Caffaro, «Comunicaciones inalámbricas de banda ultra ancha», *Network World*, 2001.
- [CS98] J. Caffery y G. Stuber, «Overview of radiolocation in CDMA cellular systems», *IEEE Communications Magazine*, 1998.
- [dlCal17] L. de la Calle, *Evaluación de dispositivos UWB para localización en interiores con medidas de distancia*. Universidad de Sevilla, 2017.
- [dEco] M. de Economía y Empresa, *Cuadro Nacional de Atribución de Frecuencias*. dirección: <https://avancedigital.gob.es/espectro/Paginas/cnaf.aspx>.
- [dRoj18] P. M. de Rojas, *Evaluación de la resistencia de un método de localización para un sistema UWB frente a las medidas sin línea de visión directa*. Universidad de Sevilla, 2018.
- [Dec] DecaWave, *EVK1000 Evaluation Kit*. dirección: <https://www.decawave.com/product/evk1000-evaluation-kit/>.
- [Ema13] S. Emami, *UWB Communication Systems: Conventional and 60 GHz Principles, Design and Standards*. Springer, 2013.
- [FF17] J. Figueiras y S. Frattasi, *Mobile positioning and tracking : from conventional to cooperative techniques*. Wiley, 2017.
- [Gar08] E. M. García, «Técnicas de Localización en Redes Inalámbricas de Sensores», 2008.
- [GS04] S. Gezici y Z. Sahinoglu, *UWB geolocation techniques for IEEE 802.15.4a personal area networks*. Tech. Rep., 2004.
- [GC09] I. Güvenç y C. C. Chong, «A survey on TOA based wireless localization and NLOS mitigation techniques», *IEEE Communications Surveys and Tutorials*, jun. de 2009.
- [GCWI07] İ. Güvenç, C. C. Chong, F. Watanabe y H. Inamura, «NLOS Identification and Weighted Least-Squares Localization for UWB Systems Using Multipath Channel Statistics», *EURASIP Journal on Advances in Signal Processing*, jul. de 2007.
- [INT] INTEL, *Ultra-Wideband (UWB Technology)*. dirección: <https://web.archive.org/web/20110403113038/http://www.intel.com/technology/comms/uwb/download/Ultra-Wideband.pdf>.

- [LRWW98] J. C. Lagarias, J. A. Reeds, M. H. Wright y P. E. Wright, «CONVERGENCE PROPERTIES OF THE NELDER–MEAD SIMPLEX METHOD IN LOW DIMENSIONS», *SIAM Journal of Optimization*, 1998.
- [Mad19] M. J. Madero, *Apuntes de Sistemas Emergentes de Comunicaciones*. Universidad de Sevilla, 2018-2019.
- [Mata] Matlab, *fminsearch*. dirección: <https://uk.mathworks.com/help/optim/ug/fminsearch.html>.
- [Matb] Matlab, *fminsearch Algorithm*. dirección: <https://uk.mathworks.com/help/optim/ug/fminsearch-algorithm.html>.
- [MCAW18] S. Mazuelas, A. Conti, J. C. Allen y M. Z. Win, «Soft Range Information for Network Localization», *IEEE Transactions on Signal Processing*, jun. de 2018.
- [MGM+10] J. M. Molina, C. García, M. V. Moreno, J. V. Rodríguez, J. Pascual y L. J. Llácer, «Sistemas de comunicaciones UWB», *ETSIT-UPCT*, 2010.
- [PKG16] K. Pahlavan, P. Krishnamurthy e Y. Geng, «Localization challenges for the emergence of the smart world», *IEEE Access*, ene. de 2016.
- [Per10] S. Perez, *Comparativa de software comercial para el cálculo y ajuste de una red gps en una obra de carácter lineal*. Universitat Politècnica de Catalunya, 2010.
- [Sac13] J. Sachs, *Handbook of Ultra-Wideband Short-Range Sensing: Theory, Sensors, Applications*. John Wiley y Sons, 2013.
- [Sew] Sewio, *Time Difference of Arrival*. dirección: <https://www.sewio.net/uwb-technology/time-difference-of-arrival/>.
- [Sta] StatisticsHowTo, *RMSE: Root Mean Square Error*. dirección: <https://www.statisticshowto.datasciencecentral.com/rmse/>.
- [TPHH19] M. F. Tsai, T. N. Pham, B. C. Hu y F. R. Hsu, «Improvement in UWB Indoor Positioning by Using Multiple Tags to Filter Positioning Errors», *Journal of Internet Technology*, 2019.
- [Vic] V. M. Vicente, *Tecnologías Localización Tiempo Real (RTLS): RFID VS BLE VS UWB*. dirección: <https://avancedigital.gob.es/espectro/Paginas/cnaf.aspx>.
- [ZM19] X. Zhang y Q. Ma, «Indoor Positioning System», *Purdue e-Pubs*, 2019.