

Trabajo Fin de Grado

Ingeniería de las Tecnologías de Telecomunicación

Localización en interiores con tecnología UWB y ponderación de medidas de distancia

Autor: Emilio De Santa-Ana Pérez

Tutora: Eva María Arias de Reyna Domínguez

Cotutor: José Antonio Pérez Carrasco

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

Sevilla, 2022



Trabajo Fin de Grado
Ingeniería de las Tecnologías de Telecomunicación

Localización en interiores con tecnología UWB y ponderación de medidas de distancia

Autor:

Emilio De Santa-Ana Pérez

Tutora:

Eva María Arias de Reyna Domínguez

Profesora titular

Cotutor:

José Antonio Pérez Carrasco

Profesor titular

Dpto. de Teoría de la Señal y Comunicaciones

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2022

Trabajo Fin de Grado: Localización en interiores con tecnología UWB y ponderación de medidas de distancia

Autor: Emilio De Santa-Ana Pérez

Tutora: Eva María Arias de Reyna Domínguez

Cotutor: José Antonio Pérez Carrasco

El tribunal nombrado para juzgar el trabajo fin de grado arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Fecha:

El secretario del Tribunal

A mi familia
A todas las personas que confiaron en mí

AGRADECIMIENTOS

Agradecer a mi tutora, Eva, por ser mi guía en este trabajo y por su paciencia infinita. Agradecer también al profesor José Antonio, por su ayuda ofrecida en este último camino.

Agradecer a mi familia, por todos los esfuerzos que han hecho para que yo pueda estar escribiendo estas palabras, por su apoyo, dedicación y comprensión.

A todas aquellas personas que han estado conmigo y me han transmitido a lo largo de estos duros años de carrera su orgullo y confianza en mí.

Emilio De Santa-Ana Pérez

Sevilla, 2022

RESUMEN

Un sistema de localización en interiores consiste en localizar por radiofrecuencia cualquier dispositivo, como un objeto o persona, en un plano bidimensional de un recinto o espacio delimitado del interior de cualquier infraestructura como pueden ser supermercados, hospitales, centros comerciales, oficinas, almacenes etc., en tiempo real.

Este trabajo fin de grado se basa en un sistema de localización en interiores con la tecnología inalámbrica de banda ultra ancha, *UWB (Ultra Wide Band)* donde se han desarrollado algoritmos de localización para la estimación de la posición de un nodo móvil, y posteriormente se ha estudiado cómo repercute la línea de visión directa e indirecta entre transmisor y receptor en la localización y cómo el sistema de localización identifica, interpreta y evalúa cada condición de visión. Es por ello que la localización en interiores de edificios y la tecnología inalámbrica de banda ultra ancha tendrán un enorme impacto en el campo de las telecomunicaciones en un futuro no muy lejano, ya que la tecnología *UWB* se presenta como un firme candidato en la localización en interiores donde aporta una gran precisión, seguridad y fiabilidad haciendo que cobre un enorme protagonismo en el internet de las cosas.

ABSTRACT

An indoor location system consists in locating by radio frequency any device, such as an object or a person, on a two-dimensional plane of an enclosure or delimited space inside any infrastructure such as supermarkets, hospitals, shopping centers, offices, warehouses etc., in real time.

This final degree project is based on an indoor location system with ultra wide band wireless technology, *UWB*, where localization algorithms have been developed for estimating the position of a mobile node, and subsequently studied how the direct and indirect line of sight between transmitter and receiver impacts the location and how the location system identifies, interprets and evaluates each vision condition. That is why indoor location of buildings and ultra wide band wireless technology will have a huge impact on the telecommunications field in the not too distant future, because *UWB* technology is presented as a strong candidate in indoor location where it provides great precision, security and reliability making it gain a huge prominence in internet of things.

ÍNDICE

Agradecimientos	ix
Resumen	xi
Abstract.....	xiii
Índice	xv
Índice de Figuras.....	xix
Índice de Tablas	xxi
Índice de Fórmulas	xxiii
1 Introducción.....	1
1.1 Objetivo del trabajo.....	1
1.2 Estructura	1
2 Fundamentos Teóricos.....	3
2.1 Sistema de localización en interiores	3
2.1.1 Tecnologías empleadas para la localización en interiores	3
Wi-Fi.....	4
BLE.....	4
RFID	4
VLC	4
UWB.....	4
2.1.2 Cálculo de la localización y posicionamiento de un nodo móvil	5
Medidas de Distancia	5
Tiempo de Llegada (ToA).....	6
Diferencia de Tiempo de Llegada (DToA)	6
Ángulo de Llegada (AoA).....	6
Fuerza de la Señal Recibida (RSS)	6
Estimación de la Posición	6
Triangulación.....	6
Trilateración.....	7
Mapeo.....	7
2.2 Tecnología UWB	10
2.2.1 Transmisión de pulsos	11
2.2.2 Ancho de banda y densidad espectral de potencia	11
3 Implementación de Algoritmos de Localización	13
3.1 Algoritmo sin identificación.....	15

3.2	Algoritmo con conocimiento perfecto de la condición de visión LOS o NLOS	16
3.3	Algoritmo con identificación de la condición de visión. $P[\text{NLOS} \text{LOS}]$ y $P[\text{LOS} \text{NLOS}]$	16
4	Simulaciones y Análisis de Resultados	19
4.1	RMSE.....	20
4.2	Simulaciones.....	21
4.2.1	RMSE para parejas de probabilidades de identificación errónea iguales	22
	Casos LOS/NLOS	24
	Caso 1: 4 anclas LOS y 1 ancla NLOS.....	24
	Caso 2: 3 anclas LOS y 2 anclas NLOS	26
	Caso 3: 2 anclas LOS y 3 anclas NLOS	28
4.2.2	RMSE para parejas de probabilidades de identificación errónea distintas	29
	Casos LOS/NLOS	31
	Caso 1: 4 anclas LOS y 1 ancla NLOS.....	31
	Caso 2: 3 anclas LOS y 2 anclas NLOS	32
	Caso 3: 2 anclas LOS y 3 anclas NLOS	33
4.2.3	RMSE fijando una probabilidad de identificación errónea igual a cero	34
	$P[\text{NLOS} \text{LOS}]=0$	34
	$P[\text{LOS} \text{NLOS}]=0$	36
4.2.4	RMSE en función de todas las probabilidades de identificación errónea.....	38
5	Conclusiones y Líneas Futuras.....	41
5.1	Conclusiones.....	41
5.2	Líneas Futuras	42
	Bibliografía y Referencias.....	43
	Anexo A: Códigos Matlab Algoritmos y Gráficas.....	45
	A.1 Algoritmo sin identificación.....	45
	A.2 Algoritmo con conocimiento perfecto de la condición de visión LOS o NLOS	45
	A.3 Algoritmo con identificación de la condición de visión. $P[\text{NLOS} \text{LOS}]$ y $P[\text{LOS} \text{NLOS}]$	46
	A.4 Representación gráfica del RMSE de cada algoritmo	47
	A.5 Representación gráfica del RMSE en función de las probabilidades de identificación errónea	48
	A.6 Archivo principal: Ejecución de los algoritmos y gráficas.....	49
	A.7 Casos Anclas LOS/NLOS	52
	Anexo B: Base de Datos de las Medidas.....	61
	B.1 Posiciones reales del nodo móvil.....	61
	B.2 Posiciones de los nodos anclas	62
	B.3 Medidas de distancia	63
	B.4 Condición de visión.....	64
	B.5 Caso 1: 4 anclas LOS y 1 anclas NLOS	66

Posición real del nodo móvil	66
Medidas de distancia.....	66
Condición de visión	66
B.6 Caso 2: 3 anclas LOS y 2 anclas NLOS	66
Posiciones reales del nodo móvil	66
Medidas de distancia.....	66
Condición de visión	66
B.7 Caso 3: 2 anclas LOS y 3 anclas NLOS	67
Posiciones reales del nodo móvil	67
Medidas de distancia.....	67
Condición de visión	68

ÍNDICE DE FIGURAS

Figura 1: Sistema real de localización en interiores	3
Figura 2: Etapas en la localización del nodo móvil.....	5
Figura 3: Triangulación	6
Figura 4: Trilateración	7
Figura 5: Mapeo (grid)	7
Figura 6: Diagrama general del sistema de localización en interiores	8
Figura 7: Diagrama del sistema de localización en interiores.....	9
Figura 8: Espectro UWB	10
Figura 9: Impulso radio	11
Figura 10: Densidad espectral UWB.....	11
Figura 11: Grid del recinto donde se aplican los algoritmos de localización	14
Figura 12: Grid del recinto con las ubicaciones de los nodos anclas y del nodo móvil a localizar.....	14
Figura 13: Grid del recinto con los nodos anclas, nodo móvil a localizar y la posición estimada del nodo móvil.....	15
Figura 14: Dpto. Teoría de la Señal y Comunicaciones. TFG: Mitigación de los errores por obstrucción de la línea de visión directa en un sistema de localización. Carmelo García.....	19
Figura 15: Representación gráfica del RMSE. Parejas de probabilidades de identificación errónea iguales.....	23
Figura 16: Representación gráfica del RMSE en función de las probabilidades de identificación errónea iguales.	23
Figura 17: Representación gráfica del RMSE. Parejas de probabilidades de identificación errónea iguales. Caso 1: 4 anclas LOS y 1 ancla NLOS para $k=500$	25
Figura 18: Representación gráfica del RMSE. Parejas de probabilidades de identificación errónea iguales. Caso 1: 4 anclas LOS y 1 ancla NLOS para $k=1000$	25
Figura 19: Representación gráfica del RMSE. Parejas de probabilidades de identificación errónea iguales. Caso 2: 3 anclas LOS y 2 anclas NLOS para $k=500$	27
Figura 20: Representación gráfica del RMSE. Parejas de probabilidades de identificación errónea iguales. Caso 2: 3 anclas LOS y 2 anclas NLOS para $k=1000$	27
Figura 21: Representación gráfica del RMSE. Parejas de probabilidades de identificación errónea iguales. Caso 3: 2 anclas LOS y 3 anclas NLOS	28
Figura 22: Representación gráfica del RMSE. Parejas de probabilidades de identificación errónea distintas	30
Figura 23: Representación gráfica del RMSE en función de las parejas de probabilidades de identificación errónea distintas	30
Figura 24: Representación gráfica del RMSE. Parejas de probabilidades de identificación errónea distintas. Caso 1: 4 anclas LOS y 1 ancla NLOS.....	31
Figura 25: Representación gráfica del RMSE. Parejas de probabilidades de identificación errónea distintas. Caso 2: 3 anclas LOS y 2 anclas NLOS.....	32

Figura 26: Representación gráfica del RMSE. Parejas de probabilidades de identificación errónea distintas. Caso 3: 2 anclas LOS y 3 anclas NLOS	33
Figura 27: Representación gráfica del RMSE fijando una probabilidad de identificación errónea igual a cero. $P[NLOS LOS]=0$	35
Figura 28: Representación gráfica del RMSE en función de una probabilidad de identificación errónea igual a cero. $P[NLOS LOS]=0$	35
Figura 29: Representación gráfica del RMSE fijando una probabilidad de identificación errónea igual a cero. $P[LOS NLOS]=0$	37
Figura 30: Representación gráfica del RMSE en función de una probabilidad de identificación errónea igual a cero. $P[LOS NLOS]=0$	37
Figura 31: Representación gráfica del RMSE en función de todas las probabilidades de identificación errónea	40

ÍNDICE DE TABLAS

Tabla 1: Límites de potencia UWB. CNAF	12
Tabla 2: RMSE parejas de probabilidades de identificación errónea iguales.....	22
Tabla 3: RMSE parejas de probabilidades de identificación errónea iguales. Caso 1: 4 anclas LOS y 1 ancla NLOS. $k=500$ y $k=1000$	24
Tabla 4: RMSE parejas de probabilidades de identificación errónea iguales. Caso 2: 3 anclas LOS y 2 anclas NLOS. $k=500$ y $k=1000$	26
Tabla 5: RMSE parejas de probabilidades de identificación errónea iguales. Caso 3: 2 anclas LOS y 3 anclas NLOS	28
Tabla 6: RMSE parejas de probabilidades de identificación errónea distintas.....	29
Tabla 7: RMSE parejas de probabilidades de identificación errónea distintas. Caso 1: 4 anclas LOS y 1 ancla NLOS.....	31
Tabla 8: RMSE parejas de probabilidades de identificación errónea distintas. Caso 2: 3 anclas LOS y 2 anclas NLOS	32
Tabla 9: RMSE parejas de probabilidades de identificación errónea distintas. Caso 3: 2 anclas LOS y 3 anclas NLOS	33
Tabla 10: RMSE fijando una probabilidad de identificación errónea igual a cero. $P[NLOS LOS]=0$	34
Tabla 11: RMSE fijando una probabilidad de identificación errónea igual a cero. $P[LOS NLOS]=0$	36
Tabla 12: RMSE todas las parejas de probabilidades de identificación errónea	39

ÍNDICE DE FÓRMULAS

Fórmula 1: Método de mínimos cuadrados	13
Fórmula 2: RMSE.....	20

1 INTRODUCCIÓN

Hoy en día, la geolocalización por satélite es nuestro mayor fiel amigo para ubicarnos en un mapa en tiempo real y que nos facilite el desplazamiento, que nos ayude a encontrar lugares o posicionar a otros dispositivos como objetos o personas. La limitación de este sistema viene cuando necesitamos mayor precisión, es decir, cuando necesitamos ubicar un lugar o un dispositivo muy concreto en el interior de un edificio o movernos por el interior del mismo.

Surge entonces, la necesidad de desarrollar un sistema de localización y posicionamiento que nos permita ubicarnos o ubicar en tiempo real y con una enorme precisión en interiores de edificios, ya que la geolocalización por satélite pierde mucha precisión cuando penetra en los edificios y no es eficiente para conseguir dicho objetivo. Es por ello que, en lugar de emplear una red de satélites como lo hacen las tecnologías *GPS*, *Galileo*, *GLONASS* y *BeiDou*, se emplea una red de dispositivos ubicados dentro de los edificios que a través de su tecnología inalámbrica permiten localizar y posicionar a objetos y personas con exactitud y en tiempo real en interiores.

1.1 Objetivo del trabajo

Este trabajo fin de grado es la continuación del proyecto de Carmelo García García, “Mitigación de los errores por obstrucción de la línea de visión directa en un sistema de localización”, donde él estudió cómo actúa la obstrucción de la línea de visión directa (*Non Line Of Sight*, *NLOS*) en un sistema de localización en interiores. Por lo tanto, con las conclusiones que obtuvo Carmelo, este trabajo fin de grado tiene como objetivo el diseño, desarrollo y simulación de distintos algoritmos matemáticos de localización para la estimación de la posición de un nodo móvil en un recinto delimitado de interior basado en la tecnología inalámbrica *UWB*, *Ultra Wide Band*.

Con las diferentes posiciones del nodo móvil que se estimarán se calculará la raíz del error cuadrático medio, *RMSE*, y en base a dicho valor se evaluará las precisiones de los algoritmos en la localización, se estudiará cómo afecta la línea de visión, *LOS* (*Line Of Sight*) o *NLOS* (*Non Line Of Sight*) en la localización y cómo actúa una posible probabilidad de error en la identificación de la condición de visión, *LOS* o *NLOS*, de cada medida de distancia, para la estimación de la posición.

1.2 Estructura

Este trabajo fin de grado se estructura en cuatro capítulos:

Fundamentos teóricos:

Se detalla la base teórica sobre la localización en interiores y la tecnología de banda ultra ancha, *UWB*.

Implementación de Algoritmos de Localización:

En este capítulo se describen los diferentes algoritmos que se han desarrollado para la localización en interiores.

Simulación y Análisis de Resultados:

En este capítulo se simulan los distintos algoritmos de localización para todas las posiciones reales del móvil a localizar. Se estudiará la precisión y eficacia de los distintos algoritmos analizando los resultados obtenidos con cada uno de ellos y se dará una respuesta a los objetivos marcados en este trabajo.

Conclusiones y Líneas Futuras:

Capítulo donde se dará una conclusión general de los resultados obtenidos en este proyecto, y se hablará del futuro de la localización en interiores y de la tecnología *UWB*.

2 FUNDAMENTOS TEÓRICOS

En este capítulo vamos a explicar qué es un sistema de localización y posicionamiento en interiores de edificios, así como las tecnologías que lo implementan, y en cuál nos hemos basado para el desarrollo de este proyecto.

2.1 Sistema de localización en interiores

Para el desarrollo de este apartado se han empleado las siguientes referencias: [1], [2], [3], [4], [5], [6], [7], [8], [9], [10] y [11].

Un sistema de localización en interiores de edificios (en inglés *Indoor Positioning System, IPS*) o también conocido como un sistema de localización en tiempo real (en inglés *Real Time Location System, RTLS*) es una red de dispositivos electrónicos que a través de una comunicación inalámbrica por radiofrecuencia y algoritmos matemáticos permiten localizar y posicionar con exactitud y en tiempo real a objetos y personas dentro de edificios.

El sistema de localización en interiores de este proyecto consta de unos dispositivos electrónicos, los cuales llamamos nodos anclas, con una ubicación conocida, que estimarán las medidas de distancia que hay al objeto o persona a localizar, llamado nodo móvil, mediante la tecnología inalámbrica *UWB (Ultra Wide Band)*. Con esta información, el sistema estimará en tiempo real con la mayor exactitud y precisión posible la posición del nodo móvil en un sistema bidimensional de coordenadas x e y , dentro del espacio o área cerrada donde se emplea la localización *indoor*. Para ello, el sistema de localización tendrá que hacer frente a todas las complicadas situaciones que se darán en la localización, como el multitrayecto de las señales y las zonas de sombra, dando lugar a medidas de distancia *NLOS (Non Line Of Sight)*, donde la línea de visión entre un nodo ancla y el nodo móvil a localizar no es directa, es decir, hay un obstáculo obstruyendo la línea de visión. Siendo considerada en el caso de sí haber línea de visión directa entre el nodo ancla y el nodo móvil como *LOS (Line Of Sight)*.



Figura 1: Sistema real de localización en interiores [1]

2.1.1 Tecnologías empleadas para la localización en interiores

Actualmente, no hay ningún estándar en cuanto a las tecnologías empleadas para la localización y el posicionamiento en interiores de edificios ya que son muchas de ellas potencialmente válidas y efectivas.

Entre las más importantes tecnologías se destacan, Wi-Fi (*Wireless Fidelity*), BLE (*Bluetooth Low Energy*), RFID (*Radio Frequency Identification*), VLC (*Visible Light Communication*) y UWB (*Ultra Wide Band*).

Wi-Fi

El *Wi-Fi* (*Wireless Fidelity*) es una buena y óptima tecnología para la localización en interiores ya que la mayoría de los edificios cuentan con una infraestructura de red *Wi-Fi*, por lo que esto lo haría un candidato de rápida y económica implantación. Permite una conexión de gran alcance con una transmisión rápida de grandes cantidades de datos. Además, todos los dispositivos electrónicos cuentan con *Wi-Fi*, presentando una fácil configuración ya conocida por los usuarios.

Los inconvenientes o desventajas que puede presentar esta tecnología para la localización en interiores es que las redes *Wi-Fi* están sujetas a posibles interferencias y, por consiguiente, una mayor latencia que puede hacer que su precisión y eficacia disminuya. Además, hay que tener en cuenta que, al ser una tecnología tan conocida y utilizada por todos los usuarios desde hace muchos años, se puede convertir en una tecnología más vulnerable ya que tiene más posibilidades de sufrir algún tipo de ataque de seguridad.

BLE

BLE (*Bluetooth Low Energy*) es una tecnología muy económica, y la mayoría de dispositivos electrónicos tienen bluetooth por lo que no habría que emplear ningún hardware externo al igual que el *Wi-Fi*. La ventaja principal de esta tecnología inalámbrica es su bajo consumo, ya que genera una señal de baja energía, pero esto hace que la comunicación bluetooth presente un alcance de conexión pequeño y una menor velocidad de transferencia de datos que el *Wi-Fi*. Además, el bluetooth, al encontrarse en la misma banda que el *Wi-Fi*, la hace también vulnerable a posibles interferencias.

RFID

Esta tecnología inalámbrica de identificación por radiofrecuencia se basa en la lectura de etiquetas o tags instaladas en los dispositivos a localizar. Las etiquetas pueden ser pasivas (no tienen batería, por lo tanto, la señal que les llega se refleja con la información proporcionada por la etiqueta, ya que dicha señal alimenta su circuito, el cual modula la señal que se refleja para establecer la comunicación) o etiquetas activas (sí tienen baterías, por lo tanto, las etiquetas de tipo activas tienen un mayor alcance de comunicación que las pasivas).

La tecnología *NFC* (*Near Field Communication*) es una *RFID* conocida actualmente por sus aplicaciones en los teléfonos móviles.

La desventaja de esta tecnología es que no todos los dispositivos electrónicos cuentan con *RFID* como, por ejemplo, *NFC*, por lo que su implantación se encarecería respecto a otras tecnologías. Además, esta tecnología inalámbrica presenta generalmente un corto alcance de comunicación, dependiendo también del tipo de etiqueta que se emplee, pero hace que esté más limitada para la localización y posicionamiento en interiores de edificios.

VLC

La comunicación por luz visible (*Visible Light Communication*), se basa en la transmisión de ondas de luz en lugar de ondas de radio. Con esta tecnología se puede aprovechar la propia iluminación del edificio para la transmisión de datos, haciendo que estas fuentes lumínicas modulen la señal siendo imperceptible para el ojo humano.

Esta tecnología presenta un corto alcance de comunicación, ya que el nodo móvil a localizar tendría que estar cerca de una fuente lumínica y que el haz de luz no se vea afectado por ningún obstáculo. Es por ello que actualmente, se encuentra muy limitada para localización *indoor* en comparación con otras tecnologías, pero se presenta como un candidato muy interesante para muchas aplicaciones.

UWB

La banda ultra ancha, *Ultra Wide Band*, se define como una tecnología de comunicación inalámbrica que emplea una gran porción del espectro electromagnético, convirtiéndose en una tecnología muy óptima para la localización en interiores de edificios, ya que la señal se transmite en periodos muy cortos de tiempo. Esto permite que las señales *UWB* puedan coexistir con otros sistemas de comunicaciones inalámbricas sin verse perjudicadas por las interferencias.

Este proyecto se basa en la tecnología *UWB* ya que es una tecnología muy robusta para la localización en interiores. Hablaremos de las características de esta tecnología con más detalle en el apartado 2.2 de este capítulo.

2.1.2 Cálculo de la localización y posicionamiento de un nodo móvil

La estimación de la localización y posicionamiento de un nodo móvil de este proyecto se divide en dos etapas o pasos. En primer lugar, se estiman las medidas de distancia que hay entre los nodos anclas y el nodo móvil a localizar mediante *UWB*, etapa que fue desarrollada por Carmelo García en su trabajo fin de grado, y, en segundo lugar, se estima la posición del nodo móvil.

Con las medidas de distancia se ubica al nodo móvil en una zona del espacio o área cerrada donde se aplica la localización *indoor*, y a continuación se procede a posicionarlo con exactitud en coordenadas x e y dentro de esa zona.



Figura 2: Etapas en la localización del nodo móvil
Figura realizada en PowerPoint

Medidas de Distancia

Las medidas de distancia para la localización de un nodo móvil pueden estimarse mediante diferentes técnicas, denominadas técnicas de localización. El sistema de localización, mediante dichas técnicas, empleará medidas de tiempos, medidas de ángulos o medidas de la potencia de la señal recibida, según la técnica de localización que se aplique, para estimar la distancia de cada nodo ancla al nodo móvil a localizar.

Las técnicas de localización más conocidas que se emplean para estimar la distancia entre un nodo ancla y el nodo móvil son las siguientes:

Tiempo de Llegada (ToA)

La técnica *ToA* (*Time of Arrival*) se basa en calcular el tiempo que una señal emitida por un nodo ancla, de velocidad conocida, tarda en llegar al nodo móvil a localizar. Para ello, ambos nodos deben estar sincronizados para poder determinar el tiempo de llegada de la señal.

Hay una relación lineal entre tiempo de propagación y distancia recorrida, ya que las señales viajan a una velocidad conocida para el sistema, por lo tanto, se podrán estimar las medidas de distancia entre los nodos ancla y el nodo móvil.

Diferencia de Tiempo de Llegada (DToA)

La técnica *DToA* (*Differential Time of Arrival*) se emplea cuando no hay una sincronización entre el nodo ancla y el nodo móvil, pero sí debe haber sincronización entre dos nodos anclas ya que estas anclas estimarán las medidas de distancia con los tiempos de llegada de cada ancla al móvil.

Ángulo de Llegada (AoA)

La técnica *AoA* (*Angle of Arrival*) calcula la medida de distancia mediante el ángulo formado por la dirección de propagación de la señal de llegada y una determinada dirección de referencia.

Fuerza de la Señal Recibida (RSS)

La técnica *RSS* (*Received Signal Strength*) estima la distancia del nodo ancla al nodo móvil mediante el nivel de potencia de la señal recibida por el nodo ancla. En base a la atenuación recibida de la señal se puede calcular la medida de distancia con la relación entre la distancia y el nivel de señal recibido.

Estimación de la Posición

Una vez obtenidas estas medidas de distancias se procede a estimar la posición del nodo móvil en un plano bidimensional, en un sistema de coordenadas x e y . Para ello, existen varias técnicas de posicionamiento que emplean algoritmos matemáticos para la estimación de la posición del nodo móvil. Las técnicas de posicionamiento más conocidas son las siguientes:

Triangulación

Esta técnica, mediante el uso de triángulos, determina la posición de un nodo móvil. Para ello, la técnica de triangulación emplea medidas de ángulos y al menos, una medida de distancia entre un nodo ancla-nodo móvil conocida.

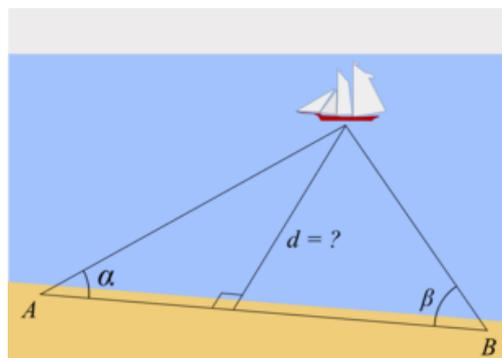


Figura 3: Triangulación [7]

Trilateración

La técnica de trilateración determina la posición de un nodo móvil mediante las medidas de distancia estimadas de cada nodo ancla al nodo móvil a localizar. Con estas distancias se aplica un sistema de ecuaciones que calcula la intersección de tres circunferencias.

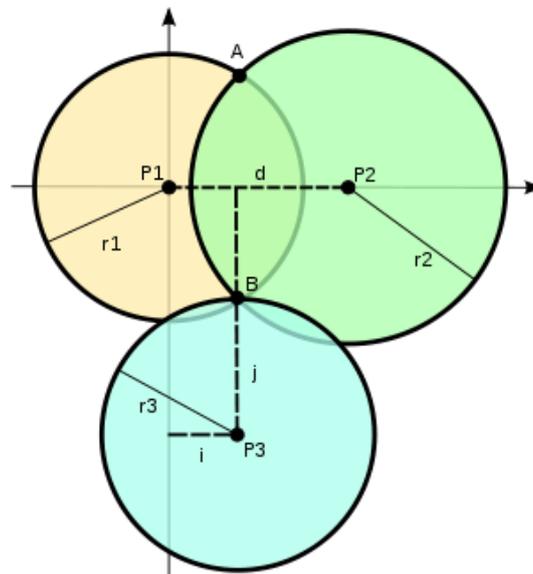


Figura 4: Trilateración [8]

Mapeo

La técnica de mapeo, también llamada *fingerprinting*, emplea una base de datos donde se almacenan los posibles valores de lo que se quiere identificar. Para este proyecto, esta técnica emplearía una base de datos donde se almacenen todas las posibles ubicaciones del nodo móvil en el recinto donde se aplica la localización en interiores, la cual denominamos *grid*. La identificación de la posición del nodo móvil se lleva a cabo mediante un método matemático de mínimos cuadrados.

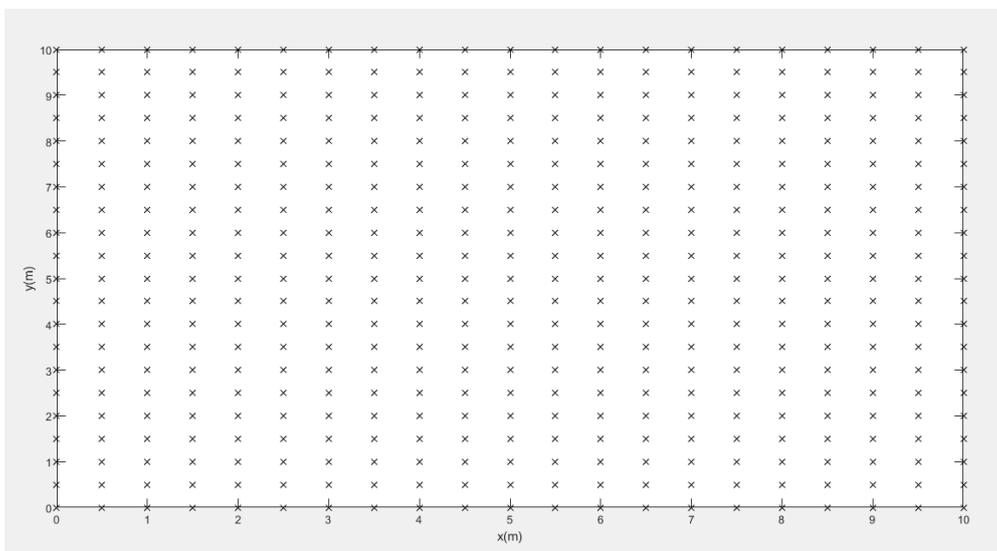


Figura 5: Mapeo (grid)
Figura realizada en MATLAB

El diagrama general del sistema de localización en interiores de este trabajo es el siguiente:

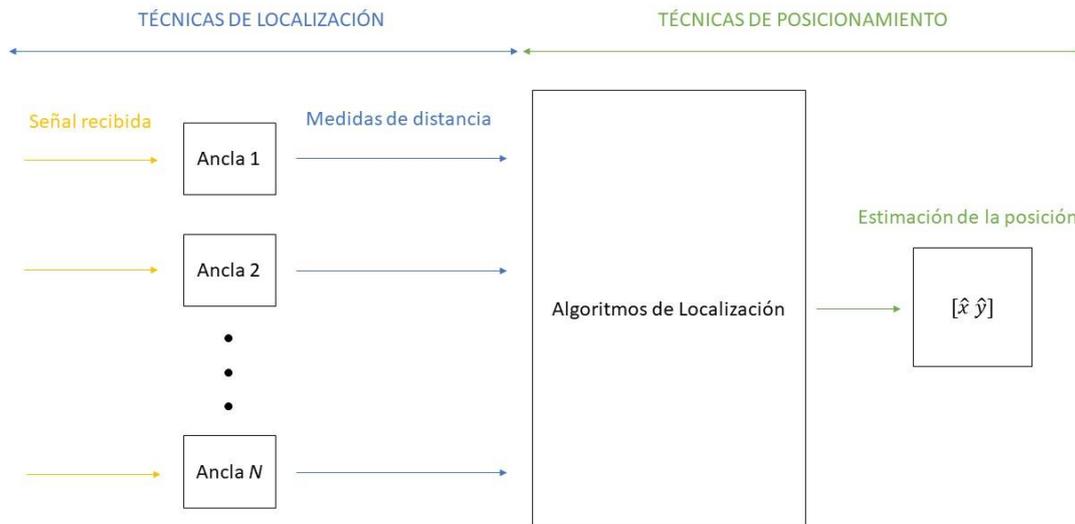


Figura 6: Diagrama general del sistema de localización en interiores

Figura realizada en PowerPoint

Los nodos anclas se comunican por radiofrecuencia con el nodo móvil a localizar, por lo que las señales inalámbricas que viajan por el espacio o área cerrada para establecer la comunicación pueden verse afectadas por diversos factores a tener en cuenta como los siguientes:

- Atenuación:
 - Por distancia: A mayor distancia tendremos una mayor pérdida de propagación, por lo que el nivel de potencia de la señal transmitida decrecerá. Hay que tener en cuenta que las tecnologías inalámbricas que se emplean en la localización de interiores son de baja potencia, por lo que son más sensibles a la distancia.
 - Por obstáculos: Cuando la señal de radio se encuentra con un obstáculo, este absorbe parte de la energía de la señal. Es lo que se conoce como el fenómeno de absorción. En un espacio cerrado podemos encontrarnos multitud de obstáculos, así como objetos, mobiliario, paredes y el constante movimiento de personas. En este caso se produce un retraso en el tiempo de llegada de la señal, pues la velocidad de propagación es menor dentro del obstáculo, dando lugar a una medida de distancia *NLOS (Non Line Of Sight)*. En algunos casos, el obstáculo puede absorber la totalidad de la energía de la señal dando lugar a que no haya comunicación entre el nodo ancla y el nodo móvil.
- Propagación multitrayecto: También conocida como propagación multicamino. Este fenómeno se da cuando la propagación de una onda electromagnética viaja por varios caminos diferentes y llega al receptor en diferentes tiempos debido a todos los posibles obstáculos que se puede ir encontrando la señal en su camino. Este fenómeno afectará notablemente en el cálculo de las medidas de distancia, dándonos una medida de distancia *NLOS* cuando la línea de visión directa entre un nodo ancla y el nodo móvil esté obstruida y la señal llegue por otro camino dando un tiempo de propagación mayor.
- Interferencias: A todo lo anterior hay que añadirle el efecto negativo de las transmisiones en el mismo rango de frecuencia de otros dispositivos electrónicos, lo que se conoce como ruido electromagnético.

El sistema de localización en interiores tendrá que contar con una red de dispositivos y una tecnología inalámbrica capaz de solventar estos efectos de la radiopropagación, ya que el canal de radio en interiores es muy sensible a la propagación multitrayecto y a las zonas de sombra por lo que es imposible predecir el comportamiento de las señales.

En este trabajo fin de grado se ha procedido de la siguiente manera.

La primera etapa para la localización en interiores es emplear una red de nodos anclas, que, a través de una comunicación inalámbrica *UWB*, apliquen la técnica de localización de tiempo de llegada, *ToA*, para calcular las medidas de distancia que hay entre los nodos anclas y el nodo móvil a localizar, etapa que fue desarrollado por Carmelo García.

En este proyecto, se ha desarrollado la segunda etapa, la estimación de la posición del nodo móvil. Para ello, se han implementado algoritmos de localización que recibirán como entradas las medidas de distancia calculadas en la primera etapa y emplearán una técnica de posicionamiento basada en mapeo, donde el sistema de localización tiene una base de datos con todas las posibles ubicaciones del nodo móvil en el recinto de interior, denominada *grid*. A continuación, los algoritmos de localización estimarán la posición del nodo móvil, en coordenadas x e y , mediante un método de mínimos cuadrados, el cual se detalla en el siguiente capítulo.

El diagrama del sistema de localización en interiores de este trabajo es el siguiente:

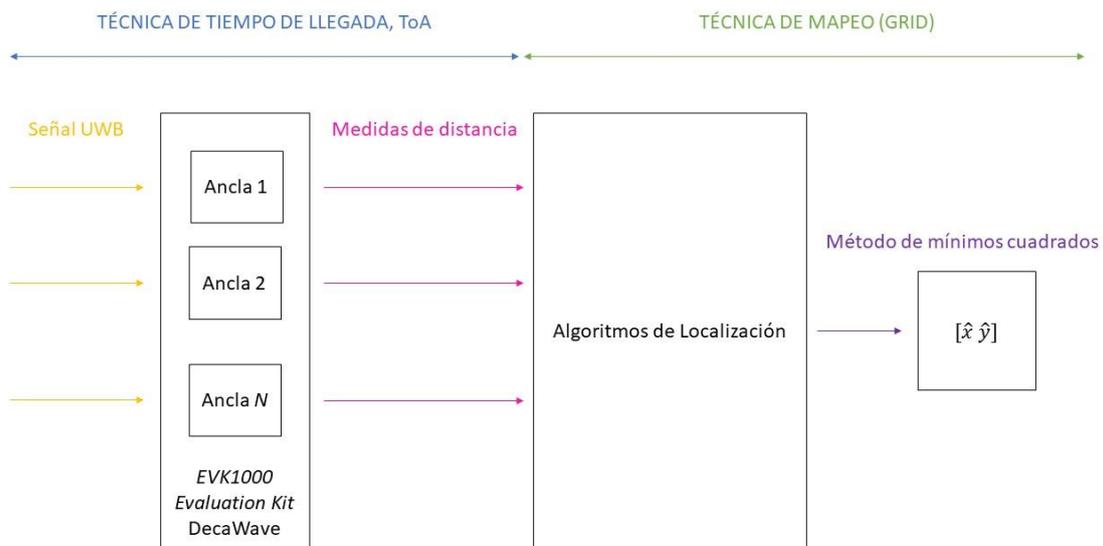


Figura 7: Diagrama del sistema de localización en interiores
Figura realizada en PowerPoint

2.2 Tecnología UWB

Para el desarrollo de este apartado se han empleado las siguientes referencias: [12], [13], [14], [15] y [16].

La tecnología *UWB*, *Ultra Wide Band*, es un sistema de comunicación inalámbrica basado en pulsos, consiste en la transmisión de pulsos electromagnéticos modulados de manera repetitiva de muy corta duración, del orden de 1ns, duración mucho menor que el intervalo de tiempo de un bit. Como resultado de la extremadamente corta duración de los pulsos de energía transmitidos, esta tecnología inalámbrica emplea un espectro de frecuencia de un gran ancho de banda, por lo tanto, su densidad espectral es muy baja ya que debe distribuir su energía por todo su gran ancho de banda.

Por todo ello, esta tecnología es muy robusta para las comunicaciones inalámbricas ya que puede coexistir con otras comunicaciones de radiofrecuencia en el mismo espectro sin verse afectada por interferencias y sin generar interferencias a otras señales de radio.

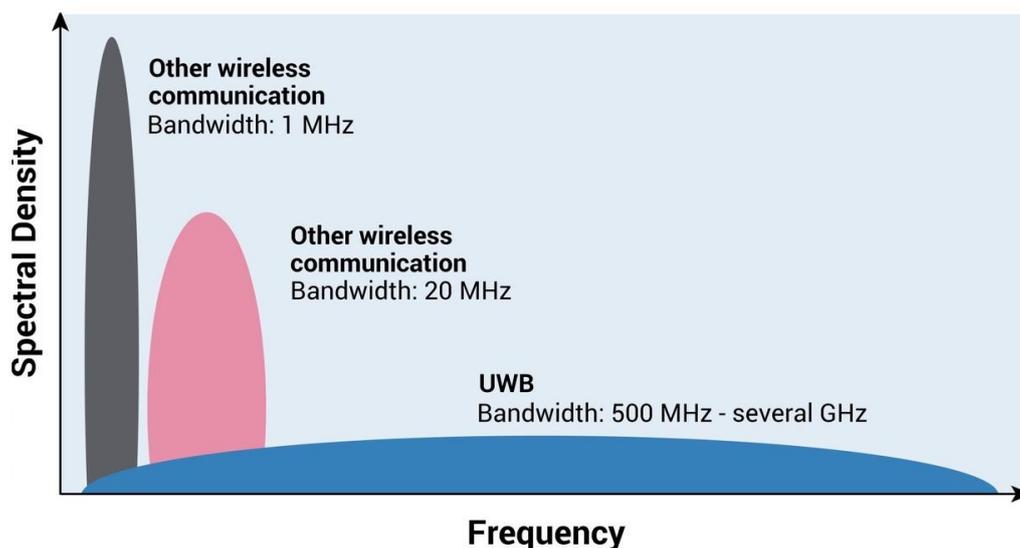


Figura 8: *Espectro UWB* [16]

Además, *UWB*, destaca por ser una tecnología muy eficiente al ofrecer un bajo consumo energético y por su bajo coste de construcción y desarrollo.

Las señales transmitidas de banda ultra ancha presentan una alta inmunidad al multirayecto, en comparación con otras tecnologías, ya que se caracterizan por tener una gran capacidad de penetración en distintos materiales debido a su amplio rango de frecuencias. Por lo tanto, esta tecnología inalámbrica hace posible que se pueda obtener una estimación precisa, con una precisión de centímetros, de la posición del nodo móvil en interiores de edificios.

UWB permite alcanzar velocidades de transmisión de datos de centenares de megabits por segundo, que van desde los 100Mbps en 10 metros, en torno a los 500 Mbps en 1 metro, y hasta más de 1Gbps en menos de un metro.

2.2.1 Transmisión de pulsos

Las señales inalámbricas *UWB* que se transmiten, las cuales se basan en impulsos radio (*IR, Impulse Radio*) presentan un flujo de pulsos con un bajo ciclo de trabajo, permaneciendo apagados durante la mayor parte del tiempo, en la que cada bit se representa y transmite mediante uno o más pulsos.

Cada bit tiene una duración T , el cual se transmite en tramas de duración T_f . Cada trama se compone de un pulso de duración T_p con un bajo ciclo de trabajo. Por lo tanto, cada bit se compone de N pulsos radio.

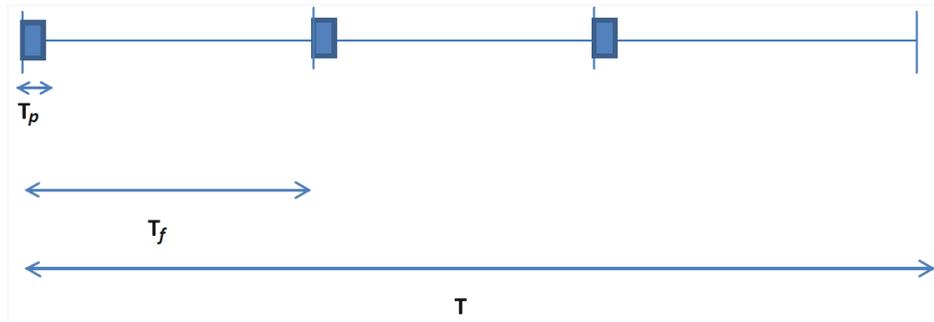


Figura 9: Impulso radio [12]

2.2.2 Ancho de banda y densidad espectral de potencia

La *ITU-R (International Telecommunication Union Radiocommunication Sector)*, define la tecnología *UWB* como la transmisión inalámbrica cuyo ancho de banda, medido a un nivel de -10 dB respecto del valor máximo de la densidad espectral de potencia (*PSD*) supera el menor valor entre los 500 MHz de ancho de banda o el 20% de la frecuencia central. Es decir, el ancho de banda de una señal de banda ultra ancha tiene que ser mayor que 500 MHz a -10 dB del valor máximo de *PSD* o mayor que el 20% de la frecuencia central de la transmisión.

- Ancho de banda, $B = f_H(\text{High Frequency}) - f_L(\text{Low Frequency}) > 500$ MHz.
- Ancho de banda, $B > 20\%$ de $f_c(\text{Center Frequency}) = (f_H + f_L) / 2$

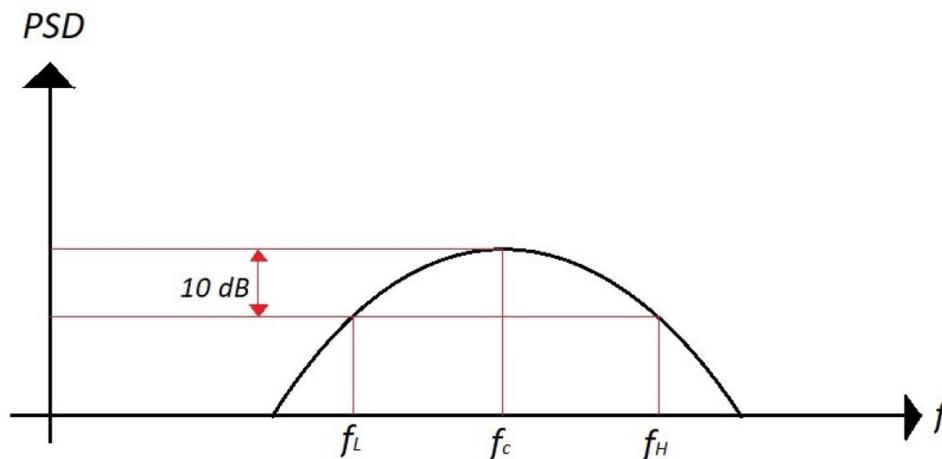


Figura 10: Densidad espectral *UWB* [10]

Las frecuencias que emplea *UWB* con sus respectivas restricciones de potencia según la banda, establecidas y reguladas por el *CNAF*, *Cuadro Nacional de Atribución de Frecuencias*, en la nota *UN-137*, se muestra en la siguiente tabla:

Banda de frecuencias (GHz)	Máxima densidad media de PIRE (dBm/MHz)
< 1.6	-90
1.6 - 2.7	-85
2.7 - 3.4	-70
3.4 - 3.8	-80
3.8 - 6	-70
6 - 8.5	-41.3
8.5 - 10.6	-65
>10.6	-85
57 – 66 (según nota UN-164)	-13

Tabla 1: *Límites de potencia UWB. CNAF [12]*

3 IMPLEMENTACIÓN DE ALGORITMOS DE LOCALIZACIÓN

En este capítulo se definen y explican los algoritmos de localización que se han desarrollado en este proyecto para estimar la posición del nodo móvil.

Los algoritmos se han implementado en MATLAB y la estimación de la posición del nodo móvil se ha calculado mediante el método matemático de mínimos cuadrados siguiente:

$$\hat{z} = \arg \min_z \left\{ \sum_{i=1}^N \beta_i (\hat{d}_i - \|z - z_i\|)^2 \right\}$$

Fórmula 1: Método de mínimos cuadrados [17]

Donde:

- N es el número total de nodos anclas del sistema de localización en interior.
- \hat{d}_i son las medidas de distancia estimadas que hay entre los nodos ancla y el nodo móvil a localizar.
- z_i son las coordenadas de cada nodo ancla, $z_i = [x_i \ y_i]$, conocidas para el sistema.
- β_i es el peso que se le da a cada medida de distancia ancla-móvil, según sea la medida *LOS* (*Line Of Sight*) o *NLOS* (*Non Line Of Sight*). $\beta = \{\beta_{LOS} = 1, \beta_{NLOS} = 0.001\}$
- z son las coordenadas de cada posible ubicación del nodo móvil del *grid* del recinto donde se aplica la localización y posicionamiento *indoor*, $z = [x \ y]$. Por lo tanto, z es la base de datos de las posibles ubicaciones del nodo móvil.
- Y finalmente, \hat{z} es la posición estimada del móvil, $\hat{z} = [\hat{x} \ \hat{y}]$. La salida del sistema.

El método de mínimos cuadrados se utiliza para calcular la recta de regresión lineal que minimiza los residuos, es decir, las diferencias entre los valores reales y los estimados por la recta. En nuestro caso, el residuo es la diferencia entre la distancia estimada, \hat{d}_i , y la distancia real entre el nodo ancla y el nodo móvil, $\|z - z_i\|$, multiplicado por un peso β .

Carmelo García, en su trabajo, concluyó que para las medidas de distancia *NLOS*, β tome el valor de 0.001, siendo este valor el óptimo en su exhaustivo estudio de su proyecto. Las medidas de distancia *NLOS* siempre van a identificar una posible ubicación del nodo móvil errónea, pero si a estas medidas se le da ese valor tan pequeño en lugar de descartarlas, lo que se intenta hacer es “disfrazar” esa medida *NLOS* como una medida de distancia *LOS*. Por lo tanto, nos van a ayudar a calcular una posición del nodo móvil más exacta, ya que cuantas más medidas de distancia *LOS* tengamos, mayor precisión vamos a tener en la estimación.

Y dicha fórmula se ha implementado de la siguiente manera:

Tomando de ejemplo un sistema de localización en interiores que consta de:

- Dimensiones del recinto donde se aplica la localización: 10x10 *metros*.
- $N=4$ nodos anclas, cuyas posiciones conocidas son: [1, 1], [1, 9], [9, 1] y [9, 9].
- Posición real del móvil a localizar: [5.5, 6].
- Caso ideal: los nodos anclas tienen visión directa, *LOS*, con el móvil, $\beta = \beta_{LOS} = 1$.

Definimos el recinto donde se va a localizar y posicionar el nodo móvil mediante un *grid* constituido por cruces, siendo estas cruces las posibles ubicaciones del móvil, por lo tanto, cada una

de estas cruces están definidas en coordenadas x e y .

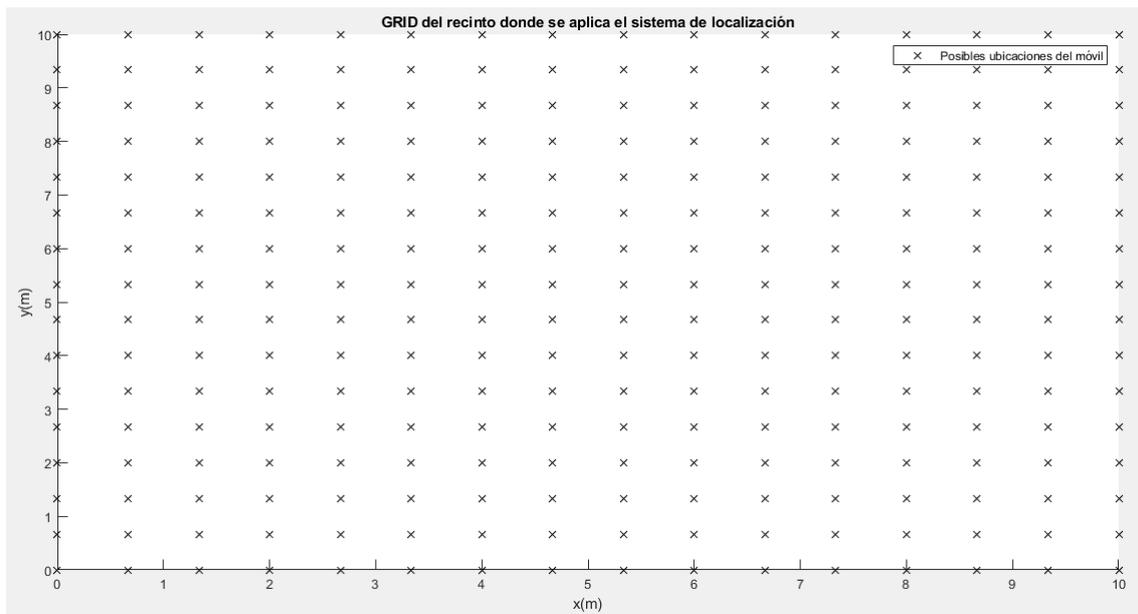


Figura 11: Grid del recinto donde se aplican los algoritmos de localización
Figura realizada en MATLAB

A mayor número de cruces del *grid*, mayor precisión se obtendrá en la estimación de la posición del nodo móvil, pero, por consiguiente, se empleará más tiempo para ello al haber un mayor cálculo computacional.

Los nodos anclas estarán ubicados por el recinto cuyas coordenadas, x e y , serán conocidas y estimarán la distancia que hay de cada uno de ellos al nodo móvil.

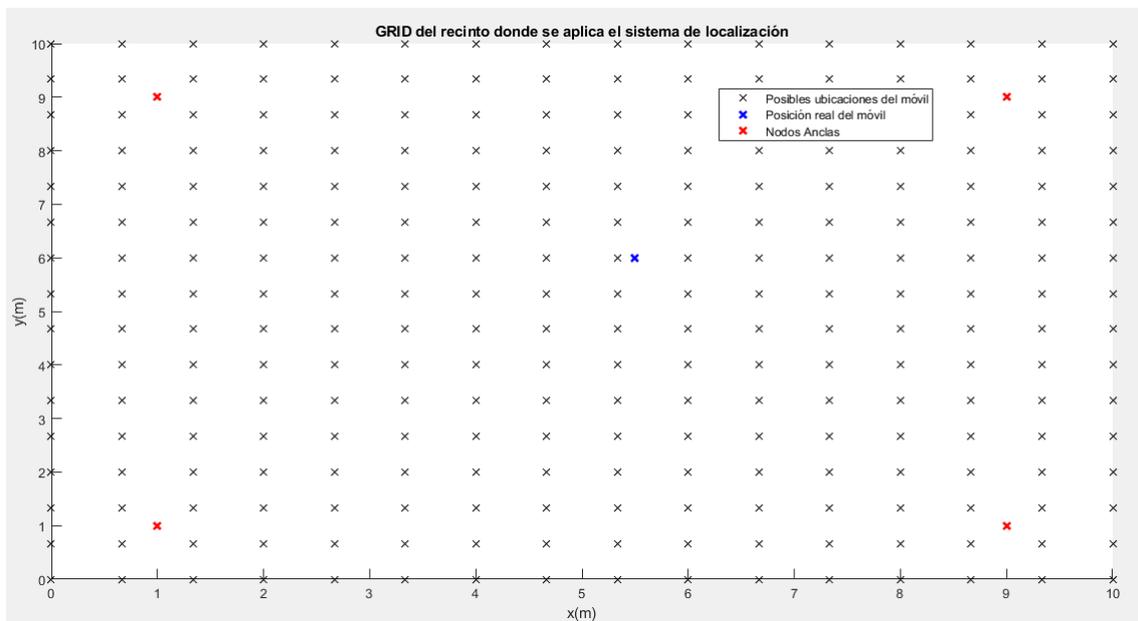


Figura 12: Grid del recinto con las ubicaciones de los nodos anclas y del nodo móvil a localizar
Figura realizada en MATLAB

Los algoritmos de localización implementados en este proyecto tomarán como entradas las medidas de distancia estimadas, iremos recorriendo el *grid* de tal manera que, para cada cruz del mismo, calculamos la distancia que hay de cada nodo ancla a cada cruz y se le resta la medida de distancia de ese nodo ancla al nodo móvil. El resultado se eleva al cuadrado y se le da un peso β , según esa medida de distancia estimada ancla-móvil sea *LOS* o *NLOS*. Por lo tanto, para cada cruz del *grid*, tendremos N cálculos, uno por cada nodo ancla. Dichos cálculos se suman y se almacenan en una variable de dimensión (*número de cruces del grid*)². El mínimo valor de dicha variable, es decir, el mínimo sumando, será quien presente un menor error de la estimación. Por lo tanto, dicho valor se corresponderá con una cruz del *grid* cuyas coordenadas x e y serán la posición estimada del nodo móvil.

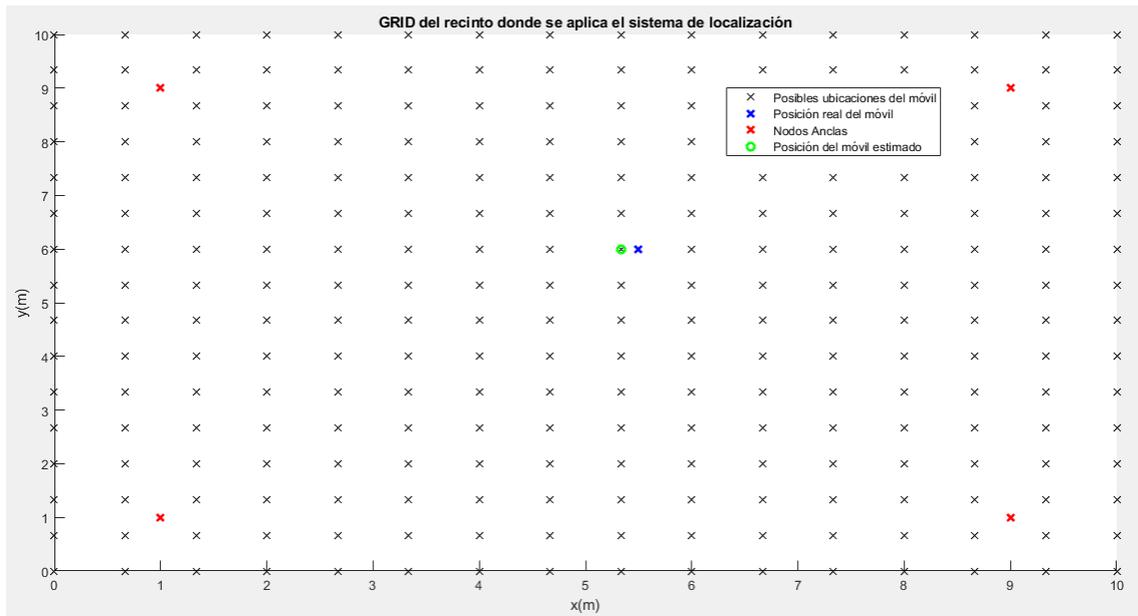


Figura 13: Grid del recinto con los nodos anclas, nodo móvil a localizar y la posición estimada del nodo móvil
Figura realizada en MATLAB

El algoritmo de localización da como salida una posición estimada del nodo móvil = [5.333, 6], dando un error de estimación de 0.167 metros.

Los distintos algoritmos de localización que se han desarrollado en este proyecto son los siguientes:

- Algoritmo sin identificación.
- Algoritmo con conocimiento perfecto de la condición de visión *LOS* o *NLOS*.
- Algoritmo con identificación de la condición de visión. $P[NLOS|LOS]$ y $P[LOS|NLOS]$.

3.1 Algoritmo sin identificación

Este algoritmo de localización emplea todas las medidas de distancia sin mitigar, es decir, sin hacer distinción entre medidas de distancia *LOS* o *NLOS*. El algoritmo no conoce si hay línea de visión directa, *LOS*, o no hay línea de visión directa, *NLOS*, entre cada nodo ancla al nodo móvil a localizar.

Este método calcula la posición estimada del nodo móvil sin tener en cuenta la condición de visión, es por ello que en este método $\beta = 1$. Por lo tanto, las medidas *NLOS*, no se suavizan, pero sí se descartan aquellas medidas cuando no hay comunicación entre algún nodo ancla al nodo móvil o cuando ambos coinciden en posición. Este algoritmo se empleará de referencia para compararlos con los algoritmos en los que sí se evalúa la condición de visión *LOS* o *NLOS*.

3.2 Algoritmo con conocimiento perfecto de la condición de visión *LOS* o *NLOS*

Este algoritmo tiene un conocimiento perfecto de la condición de visión, sabe si hay visión directa, *LOS*, o si no hay visión directa, *NLOS*, entra cada nodo ancla al nodo móvil. Por lo tanto, β , toma protagonismo en este algoritmo, $\beta = \{\beta_{LOS}, \beta_{NLOS}\}$ según sea la medida de distancia *LOS* (β_{LOS}) o *NLOS* (β_{NLOS}).

- Si la medida de distancia es *LOS*, entonces, $\beta = \beta_{LOS} = 1$.
- Si la medida de distancia es *NLOS*, entonces, $\beta = \beta_{NLOS} = 0.001$.
- Y si no hay comunicación entre alguna ancla-móvil o si ambos coinciden en posición, entonces, $\beta = 0$. La medida se descarta.

Este algoritmo de localización se puede definir como un algoritmo ideal, ya que en todos los casos se conoce la condición de visión, *LOS* o *NLOS*, de cada medida de distancia, pudiendo actuar en consecuencia consiguiendo resultados más precisos en la localización. En la realidad esto no ocurre, en un caso práctico de localización el sistema no sabe con exactitud si la medida de distancia que estima tiene visión directa o no, por lo tanto, en la identificación de la medida de distancia como *LOS* o *NLOS* entra en juego una probabilidad de identificación errónea de la medida, la cual da lugar al desarrollo del siguiente algoritmo de localización. Por todo ello, este algoritmo ideal se empleará como referencia para compararlos con los algoritmos realistas.

3.3 Algoritmo con identificación de la condición de visión. $P[NLOS|LOS]$ y $P[LOS|NLOS]$

Este algoritmo de localización contemplará la posibilidad de que haya un error en identificar las medidas de distancia *LOS* cuando son *LOS* o *NLOS* cuando son *NLOS*. En un caso real de localización, el sistema puede interpretar una medida de distancia que tiene visión directa como una medida de distancia en la cual no hay visión directa entre el nodo ancla y el nodo móvil, o viceversa. Por lo tanto, habrá una probabilidad de identificación errónea que condicionará si la medida se interpreta como *NLOS* cuando es *LOS* o si se interpreta como *LOS* cuando es *NLOS*:

- ❖ $P[NLOS/LOS]$ = Probabilidad de identificar la medida de distancia *LOS* como *NLOS*.
- ❖ $P[LOS/NLOS]$ = Probabilidad de identificar la medida de distancia *NLOS* como *LOS*.

En este método se procede de la siguiente manera:

- Si la medida de distancia es *LOS*:
 - Habrá una probabilidad de error, $P[NLOS/LOS]$, de identificar la medida *LOS* como *NLOS*, entonces, $\beta = \beta_{NLOS} = 0.001$. Identificación errónea.
 - Y, por el contrario, tendremos una probabilidad de acierto, $1-P[NLOS/LOS]$, de identificar la medida *LOS* como *LOS*, entonces, $\beta = \beta_{LOS} = 1$. Identificación correcta.
- Si la medida de distancia es *NLOS*:
 - Habrá una probabilidad de error, $P[LOS/NLOS]$, de identificar la medida *NLOS* como *LOS*, entonces, $\beta = \beta_{LOS} = 1$. Identificación errónea.
 - Y, por el contrario, tendremos una probabilidad de acierto, $1-P[LOS/NLOS]$, de identificar la medida *NLOS* como *NLOS*, entonces, $\beta = \beta_{NLOS} = 0.001$. Identificación correcta.
- Y si no hay comunicación entre algún nodo ancla-nodo móvil o si ambos coinciden en posición, entonces, $\beta = 0$. La medida se descarta.

Este algoritmo de localización se va a emplear con distintos valores de probabilidades de identificación errónea para desarrollar un estudio más completo del funcionamiento del mismo. La idea y el objetivo de este algoritmo es que, al simularlo, averiguar qué probabilidad o qué probabilidades de identificación errónea se tiene que dar para tener una buena localización del nodo móvil.

Se van a emplear 5 parejas distintas de probabilidades de identificación errónea, por lo tanto, este algoritmo de localización tendrá 5 modalidades, una para cada pareja de probabilidades.

- ❖ $P[NLOS/LOS] = \{P1[NLOS/LOS], P2[NLOS/LOS], P3[NLOS/LOS], P4[NLOS/LOS], P5[NLOS/LOS]\}$.
- ❖ $P[LOS/NLOS] = \{P1[LOS/NLOS], P2[LOS/NLOS], P3[LOS/NLOS], P4[LOS/NLOS], P5[LOS/NLOS]\}$.

Estas parejas de probabilidades no tienen por qué ser iguales entre sí, por lo que, para un estudio más completo del comportamiento de este algoritmo de localización, iremos cambiando los valores de las parejas de probabilidad de error.

También hay que tener en cuenta que, al actuar una probabilidad en este algoritmo, cada modalidad de este algoritmo se va a ejecutar k veces para cada posición real del móvil a localizar, para que así, obtengamos resultados más significativos. En el desarrollo práctico se hablará del valor de k óptimo elegido para los distintos casos.

Cabe destacar que existen distintos métodos para que un sistema de localización identifique una medida de distancia como *LOS* o *NLOS* cuando la calcula, pero dichos métodos quedan fuera de los objetivos de este proyecto. El algoritmo *con identificación de la condición de visión* se caracteriza por actuar unas probabilidades de error en la identificación realizada por dichos métodos. Por lo tanto, el algoritmo *con identificación de la condición de visión* para este proyecto se ve como una caja negra y el objetivo es averiguar cómo influyen estas probabilidades de identificación errónea en la localización.

4 SIMULACIONES Y ANÁLISIS DE RESULTADOS

En este proyecto, al ser la continuación del trabajo fin de grado de Carmelo García García, se ha empleado la misma base de datos (posiciones de los nodos anclas, posiciones reales del móvil, medidas de distancia, y la condición de visión, *LOS* o *NLOS*, de cada medida de distancia) que Carmelo generó para su desarrollo práctico con el equipo *EVK1000 Evaluation Kit* de la compañía DecaWave [11], en un espacio de interior de las siguientes características:

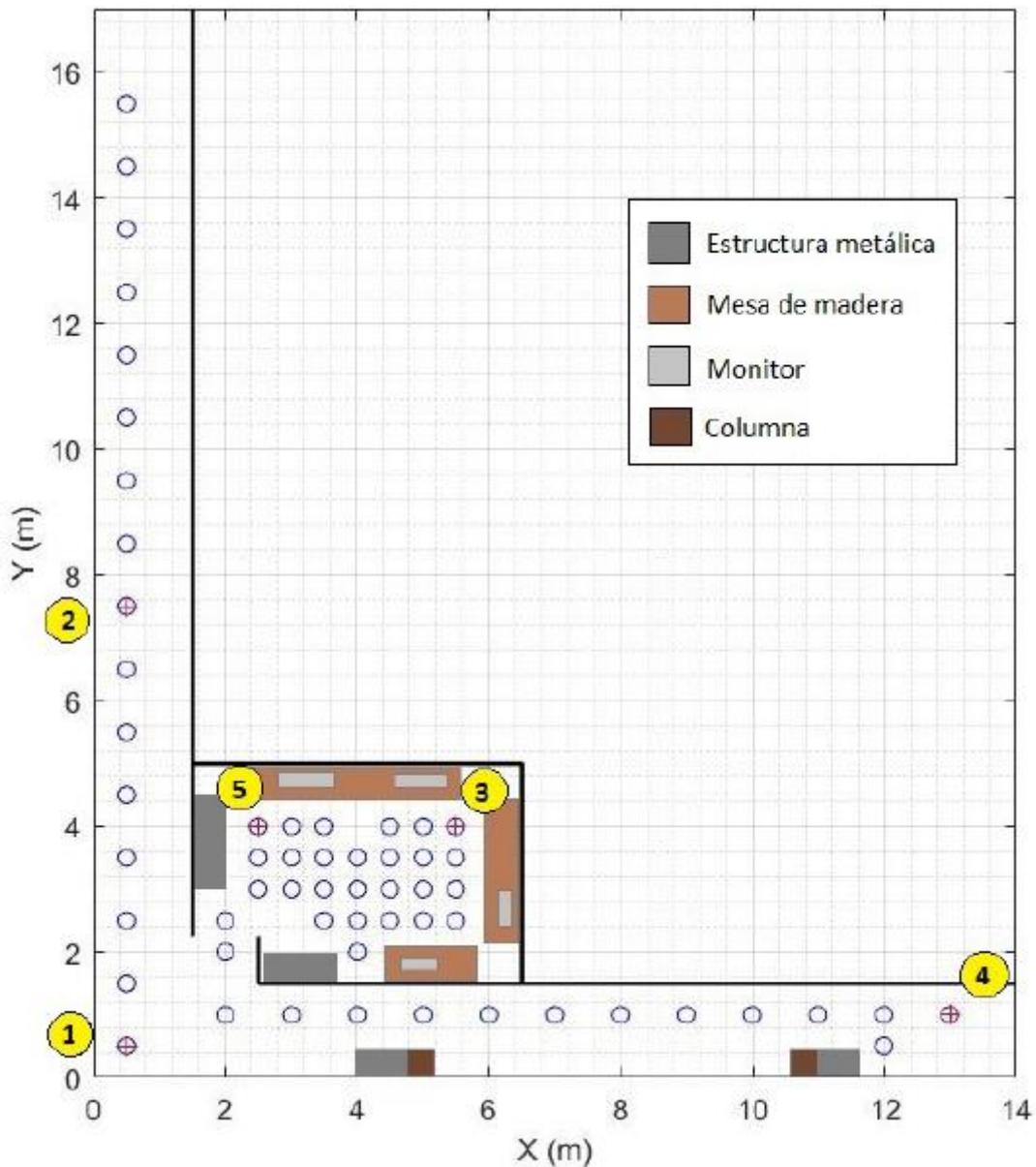


Figura 14: Dpto. Teoría de la Señal y Comunicaciones. TFG: Mitigación de los errores por obstrucción de la línea de visión directa en un sistema de localización. Carmelo García [10]

El recinto delimitado para la localización se ubica en el Dpto. Teoría de la Señal y Comunicaciones de la Escuela Técnica Superior de Ingeniería de la Universidad de Sevilla. Consta de una sala y un pasillo en donde podemos encontrar diferentes obstáculos como paredes, mesas, sillas, estanterías... así como el constante movimiento de personas por el interior del departamento.

Los algoritmos empleados para localización que se desarrollan en este proyecto se van a aplicar en este espacio delimitado. Se trata de un sistema de localización en interiores que se compone de 5 nodos anclas (*cruces rojas dentro de puntos blancos*) y 57 posiciones reales del móvil (*puntos blancos*) que son los nodos móviles que se van a localizar en las simulaciones. La distribución de los nodos anclas por el recinto de interior queda de la siguiente manera:

Los nodos ancla 1, 2 y 4 se encuentran ubicados en el exterior de la sala. El nodo ancla 1 se encuentra en la intersección de ambas partes del pasillo, teniendo línea de visión directa con los nodos móviles de ambas partes del pasillo y con algunos nodos móviles del interior de la sala ya que la puerta de la sala se dejó abierta durante la toma de las medidas. El nodo ancla 2 está situado en la parte vertical del pasillo, teniendo línea de visión directa con los nodos móviles de esa misma parte del pasillo. El nodo ancla 4 se encuentra ubicado en la parte horizontal del pasillo, teniendo línea de visión directa con los nodos móviles de esa zona del pasillo. Por último, los nodos ancla 3 y 5 están situados en el interior de la sala, teniendo línea de visión directa con todos los nodos móviles del interior de la misma y con aquellos nodos móviles del pasillo que se encuentran cerca de la entrada de la sala.

Disponemos de 57 posiciones reales del móvil a localizar para tener un estudio más exhaustivo y completo, ya que, los nodos anclas se encuentran en *LOS*, *NLOS* y sin comunicación o coincidiendo con la posición real de cada móvil a localizar. De esta manera, el estudio es más realista, ya que la idea del proyecto es simular estas situaciones para estudiar y verificar la efectividad de la localización de los distintos algoritmos empleados en una situación real, así como darle respuesta a los objetivos marcados en este trabajo.

El objetivo del desarrollo práctico de este proyecto es simular los algoritmos de localización con cada posición real del móvil que se quiere localizar. Los algoritmos tomarán como entradas las medidas de distancia de cada nodo ancla al nodo móvil a localizar. Con estas medidas de distancia, se calcula y se estima una posición del nodo móvil, y a continuación, se obtiene una representación gráfica del error cuadrático medio, *RMSE*, de cada algoritmo.

4.1 RMSE

El *RMSE* (*Root Mean Square Error, raíz del error cuadrático medio*) es la media de los errores de la estimación al cuadrado. Por lo tanto, cuanto más pequeño sea el *RMSE*, más cercanas estarán la posición real y la posición estimada del nodo móvil.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n e_i^2}$$

Fórmula 2: *RMSE* [10]

Donde:

- n es el número total de nodos móvil a localizar.
- e es el error de la estimación, distancia entre la posición real y la posición estimada.
- Para el algoritmo *con identificación de la condición de visión*. $P[NLOS/LOS]$ y $P[LOS/NLOS]$, al ejecutarse k veces para cada posición real del móvil, el sumatorio llegará, por tanto, a $n \cdot k$.

4.2 Simulaciones

Se procede a la simulación de los algoritmos de localización que hemos implementado en este proyecto, para ello, se aplican dichos algoritmos para los distintos nodos móviles a localizar, se calcula el *RMSE* y se representa gráficamente. Para ello, disponemos de la base de datos que generó Carmelo García en el desarrollo de su trabajo fin de grado con todas las medidas necesarias para que los algoritmos desarrollados funcionen correctamente y puedan estimar la posición del nodo móvil, por lo tanto, tendremos una base de datos con 4 ficheros detallados a continuación:

- ❖ Las 57 posiciones reales de cada nodo móvil a localizar en coordenadas x e y .
- ❖ Las 5 posiciones de los nodos anclas en coordenadas x e y .
- ❖ Las medidas de distancia de cada ancla-móvil.
- ❖ La condición de visión de cada ancla-móvil (*LOS*, *NLOS*, sin comunicación o que coinciden en posición).

También, para un estudio más exhaustivo y completo de este proyecto, se ha realizado un desglose en diferentes casos de *LOS* y *NLOS*, que consiste en quedarnos únicamente con los 5 nodos anclas operativos, es decir, aquellos nodos anclas que tienen visión directa con el nodo móvil o que no tienen visión directa con el nodo móvil, teniendo, para cada nodo móvil a localizar, los 5 nodos anclas en *LOS* o en *NLOS*, descartando, por lo tanto, aquellos casos en los que algún nodo ancla no tiene comunicación con el nodo móvil o ambos coinciden en posición.

Este estudio nos sirve para sacar conclusiones acerca de la relevancia que tienen los nodos anclas según estén en *LOS* o *NLOS* con el móvil a localizar. Por lo tanto, teniendo en todo momento los 5 nodos anclas operativos, tenemos 3 casos de anclas en *LOS* o *NLOS*, definidos de la siguiente manera:

- CASO 1: 4 ANCLAS *LOS* y 1 ANCLA *NLOS*. Contiene 1 posición real del nodo móvil.
- CASO 2: 3 ANCLAS *LOS* y 2 ANCLAS *NLOS*. Contiene 7 posiciones reales del nodo móvil.
- CASO 3: 2 ANCLAS *LOS* y 3 ANCLAS *NLOS*. Contiene 28 posiciones reales del nodo móvil.

En total, nos quedamos con 36 posiciones reales del móvil a localizar, por lo tanto, se descartan, debido a lo comentado, 21 posiciones reales del móvil.

Para desarrollar este estudio se ha creado una nueva base de datos en este proyecto partiendo de la que generó Carmelo García en la que se ha ajustado al nuevo número de medidas para cada caso de desglose. Entonces, por ejemplo, para el primer caso, "4 ANCLAS *LOS* y 1 *NLOS*", tenemos 3 ficheros (posiciones reales del móvil, medidas de distancia y condición de visión). Y lo mismo ocurre para los otros dos casos. El fichero de las posiciones de los nodos anclas seguiría siendo el mismo para los tres casos.

Por todo ello, en las simulaciones se van a emplear los algoritmos de localización para diferentes números de posiciones reales del móvil:

- Para el total de las posiciones reales del móvil a localizar, 57 posiciones.
- Y para los casos de desglose en nodos anclas *LOS/NLOS*. 36 posiciones.
 - Caso 1: 1 posición real del móvil a localizar.
 - Caso 2: 7 posiciones reales del móvil a localizar.
 - Caso 3: 28 posiciones reales del móvil a localizar.

En el apartado 3.3: *Algoritmo con identificación de la condición de visión*. $P[NLOS|LOS]$ y $P[LOS|NLOS]$ del anterior capítulo, se comentó que dicho algoritmo se ejecuta k veces para cada posición real del móvil a localizar. Carmelo García concluyó que para $k=500$ se obtenían buenos resultados, pero él empleaba el total de posiciones del nodo móvil a localizar, las 57 posiciones. En el desarrollo de este proyecto se ha observado que el valor de k está relacionado con el número total de nodos móviles a localizar, a menor número de posiciones reales del móvil, mayor número de veces debe ejecutarse el algoritmo para cada posición del nodo móvil. Por lo tanto, el valor de k será diferente

según el número de posiciones reales del móvil a localizar tengamos.

4.2.1 RMSE para parejas de probabilidades de identificación errónea iguales

Para las siguientes parejas de probabilidades de identificación errónea,

- ❖ $P1[NLOS/LOS]=0.01$ y $P1[LOS/NLOS]=0.01$
- ❖ $P2[NLOS/LOS]=0.02$ y $P2[LOS/NLOS]=0.02$
- ❖ $P3[NLOS/LOS]=0.05$ y $P3[LOS/NLOS]=0.05$
- ❖ $P4[NLOS/LOS]=0.1$ y $P4[LOS/NLOS]=0.1$
- ❖ $P5[NLOS/LOS]=0.2$ y $P5[LOS/NLOS]=0.2$

el RMSE de cada algoritmo para las 57 posiciones reales del móvil es el siguiente:

<i>ALGORITMOS</i>	<i>RMSE (metros)</i>
<i>Algoritmo 1:</i> Sin identificación	1.245
<i>Algoritmo 2:</i> Con conocimiento perfecto de la condición de visión <i>LOS/NLOS</i>	0.884
<i>Algoritmo 3:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0.01$ y $P[LOS NLOS]=0.01$	0.915
<i>Algoritmo 4:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0.02$ y $P[LOS NLOS]=0.02$	0.919
<i>Algoritmo 5:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0.05$ y $P[LOS NLOS]=0.05$	0.968
<i>Algoritmo 6:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0.1$ y $P[LOS NLOS]=0.1$	1.060
<i>Algoritmo 7:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0.2$ y $P[LOS NLOS]=0.2$	1.280

Tabla 2: RMSE parejas de probabilidades de identificación errónea iguales

Cuya representación gráfica queda tal que así:

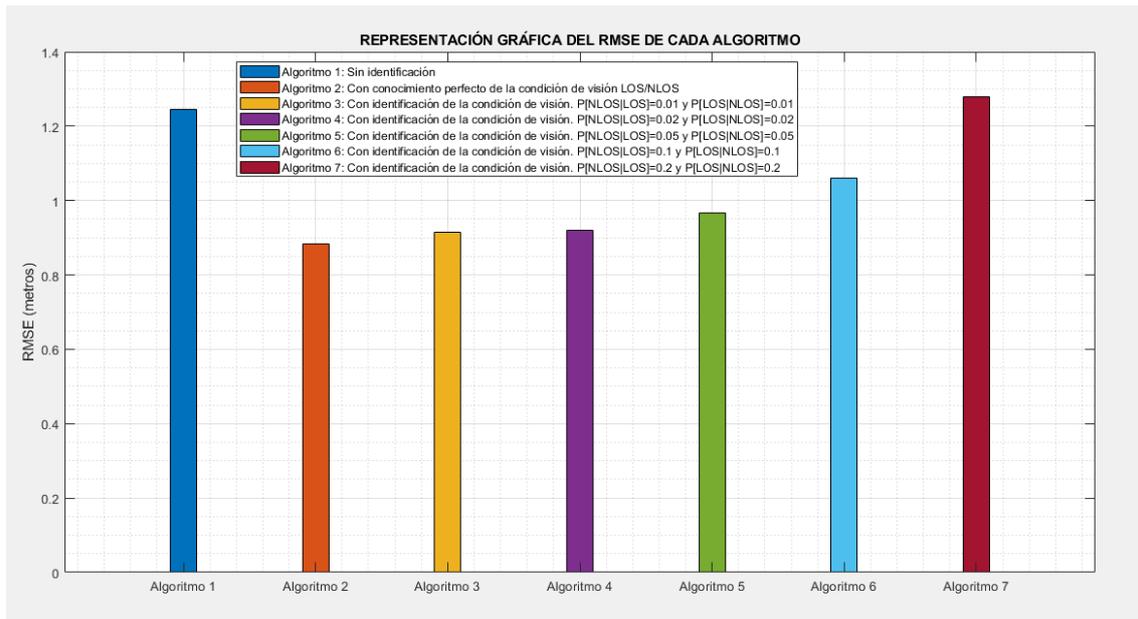


Figura 15: Representación gráfica del RMSE. Parejas de probabilidades de identificación errónea iguales
Figura realizada en MATLAB

En este caso observamos que para una probabilidad de identificación errónea de un 20% conseguiríamos una localización del móvil peor que la localización del móvil que daría el algoritmo *sin identificación*, por lo que tendríamos, para dicha probabilidad de error, una localización desfavorable del nodo móvil.

Y la representación gráfica en 3D del RMSE en función de las probabilidades de identificación errónea es la siguiente:

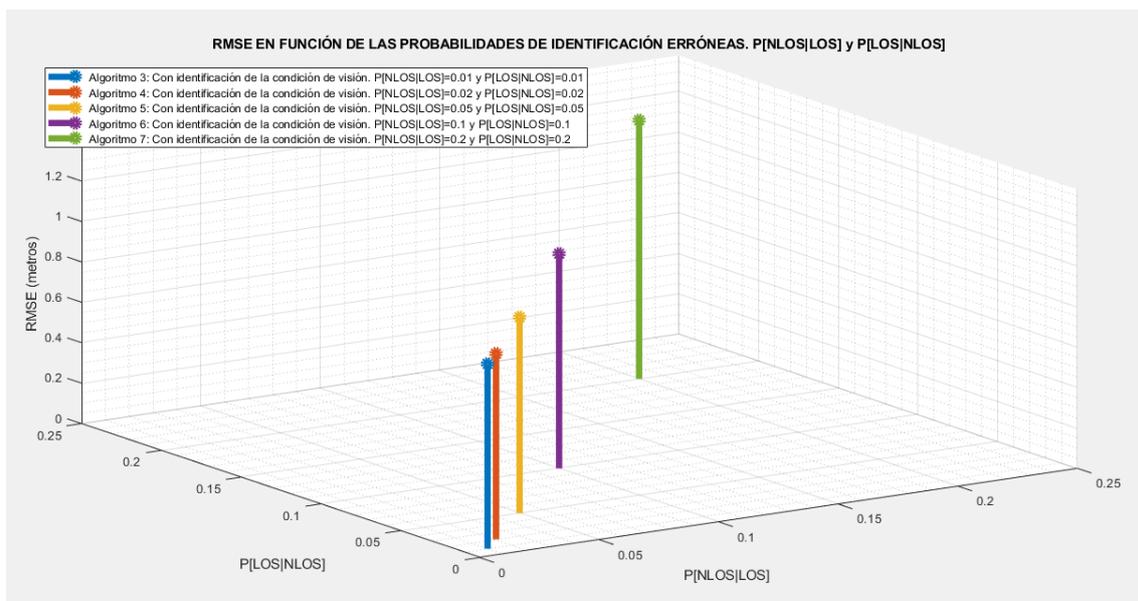


Figura 16: Representación gráfica del RMSE en función de las probabilidades de identificación errónea iguales.
Figura realizada en MATLAB

Casos LOS/NLOS

Caso 1: 4 anclas LOS y 1 ancla NLOS

El RMSE de cada algoritmo para 1 posición real del móvil es el siguiente:

$k=500$		$k=1000$	
ALGORITMOS	RMSE (metros)	ALGORITMOS	RMSE (metros)
Algoritmo 1: Sin identificación	0.177	Algoritmo 1: Sin identificación	0.177
Algoritmo 2: Con conocimiento perfecto de la condición de visión LOS/NLOS	0.125	Algoritmo 2: Con conocimiento perfecto de la condición de visión LOS/NLOS	0.125
Algoritmo 3: Con identificación de la condición de visión. $P[NLOS LOS]=0.01$ y $P[LOS NLOS]=0.01$	0.132	Algoritmo 3: Con identificación de la condición de visión. $P[NLOS LOS]=0.01$ y $P[LOS NLOS]=0.01$	0.135
Algoritmo 4: Con identificación de la condición de visión. $P[NLOS LOS]=0.02$ y $P[LOS NLOS]=0.02$	0.139	Algoritmo 4: Con identificación de la condición de visión. $P[NLOS LOS]=0.02$ y $P[LOS NLOS]=0.02$	0.141
Algoritmo 5: Con identificación de la condición de visión. $P[NLOS LOS]=0.05$ y $P[LOS NLOS]=0.05$	0.159	Algoritmo 5: Con identificación de la condición de visión. $P[NLOS LOS]=0.05$ y $P[LOS NLOS]=0.05$	0.165
Algoritmo 6: Con identificación de la condición de visión. $P[NLOS LOS]=0.1$ y $P[LOS NLOS]=0.1$	0.232	Algoritmo 6: Con identificación de la condición de visión. $P[NLOS LOS]=0.1$ y $P[LOS NLOS]=0.1$	0.222
Algoritmo 7: Con identificación de la condición de visión. $P[NLOS LOS]=0.2$ y $P[LOS NLOS]=0.2$	0.683	Algoritmo 7: Con identificación de la condición de visión. $P[NLOS LOS]=0.2$ y $P[LOS NLOS]=0.2$	0.464

Tabla 3: RMSE parejas de probabilidades de identificación errónea iguales. Caso 1: 4 anclas LOS y 1 ancla NLOS. $k=500$ y $k=1000$

Y su representaciones gráficas son las siguientes:

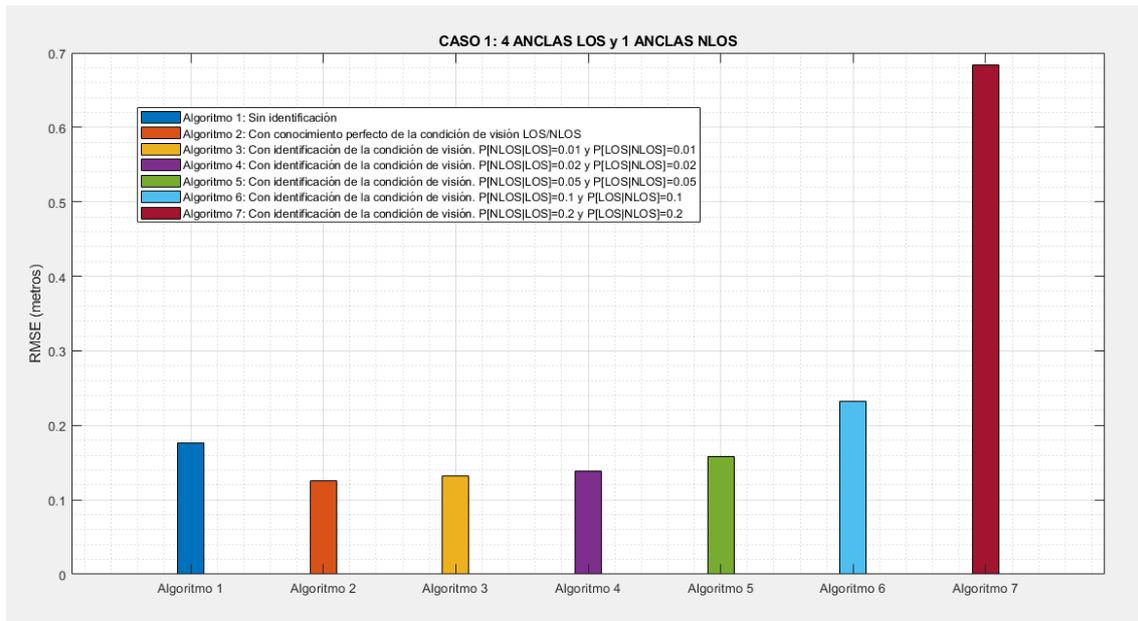


Figura 17: Representación gráfica del RMSE. Parejas de probabilidades de identificación errónea iguales. Caso 1: 4 anclas LOS y 1 ancla NLOS para $k=500$
Figura realizada en MATLAB

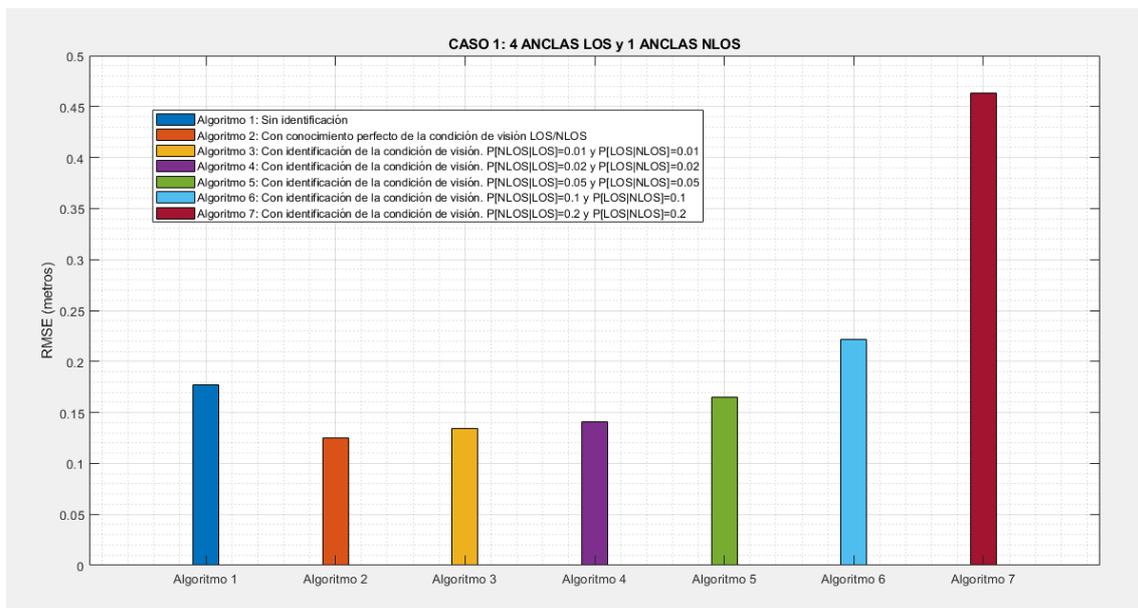


Figura 18: Representación gráfica del RMSE. Parejas de probabilidades de identificación errónea iguales. Caso 1: 4 anclas LOS y 1 ancla NLOS para $k=1000$
Figura realizada en MATLAB

En estas representaciones del RMSE, donde el algoritmo con identificación de la condición de visión, se ejecuta $k=500$ y $k=1000$ veces para cada localización del nodo móvil, se observa que el RMSE para $k=1000$ mejora para las probabilidades de error más altas. Se obtienen resultados más significativos. Para valores superiores a $k=1000$ se ha comprobado que no hay diferencias relevantes.

Caso 2: 3 anclas LOS y 2 anclas NLOS

El RMSE de cada algoritmo para las 7 posiciones reales del móvil es el siguiente:

$k=500$		$k=1000$	
<i>ALGORITMOS</i>	<i>RMSE (metros)</i>	<i>ALGORITMOS</i>	<i>RMSE (metros)</i>
<i>Algoritmo 1:</i> Sin identificación	0.189	<i>Algoritmo 1:</i> Sin identificación	0.189
<i>Algoritmo 2:</i> Con conocimiento perfecto de la condición de visión LOS/NLOS	0.149	<i>Algoritmo 2:</i> Con conocimiento perfecto de la condición de visión LOS/NLOS	0.149
<i>Algoritmo 3:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0.01$ y $P[LOS NLOS]=0.01$	0.183	<i>Algoritmo 3:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0.01$ y $P[LOS NLOS]=0.01$	0.173
<i>Algoritmo 4:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0.02$ y $P[LOS NLOS]=0.02$	0.202	<i>Algoritmo 4:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0.02$ y $P[LOS NLOS]=0.02$	0.199
<i>Algoritmo 5:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0.05$ y $P[LOS NLOS]=0.05$	0.266	<i>Algoritmo 5:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0.05$ y $P[LOS NLOS]=0.05$	0.253
<i>Algoritmo 6:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0.1$ y $P[LOS NLOS]=0.1$	0.329	<i>Algoritmo 6:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0.1$ y $P[LOS NLOS]=0.1$	0.317
<i>Algoritmo 7:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0.2$ y $P[LOS NLOS]=0.2$	0.443	<i>Algoritmo 7:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0.2$ y $P[LOS NLOS]=0.2$	0.441

Tabla 4: RMSE parejas de probabilidades de identificación errónea iguales. Caso 2: 3 anclas LOS y 2 anclas NLOS. $k=500$ y $k=1000$

Y su representaciones gráficas son las siguientes:

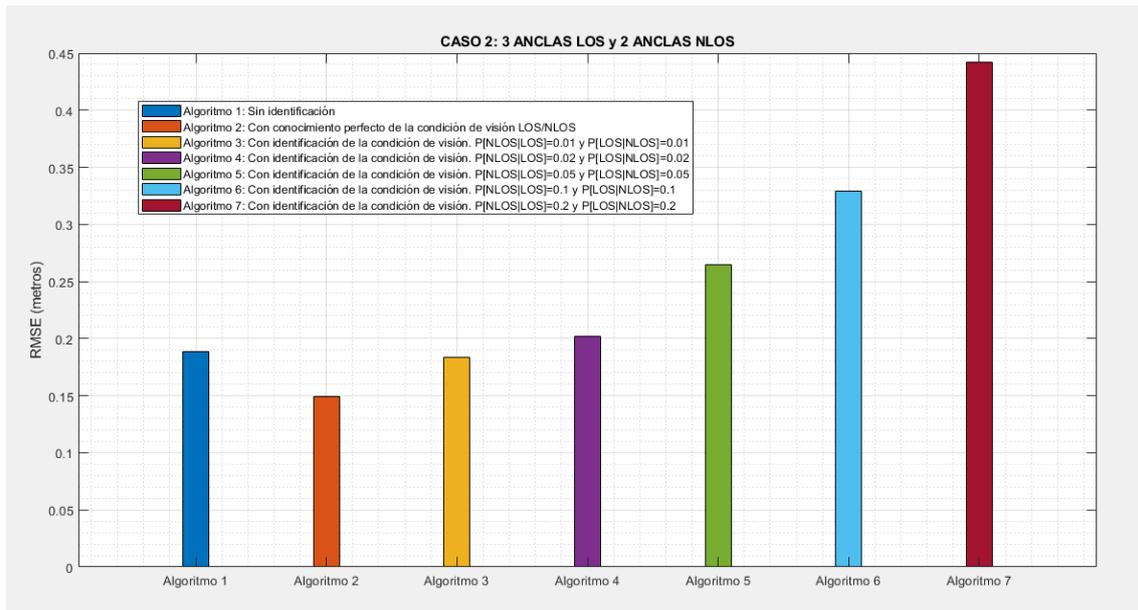


Figura 19: Representación gráfica del RMSE. Parejas de probabilidades de identificación errónea iguales. Caso 2: 3 anclas LOS y 2 anclas NLOS para $k=500$
Figura realizada en MATLAB

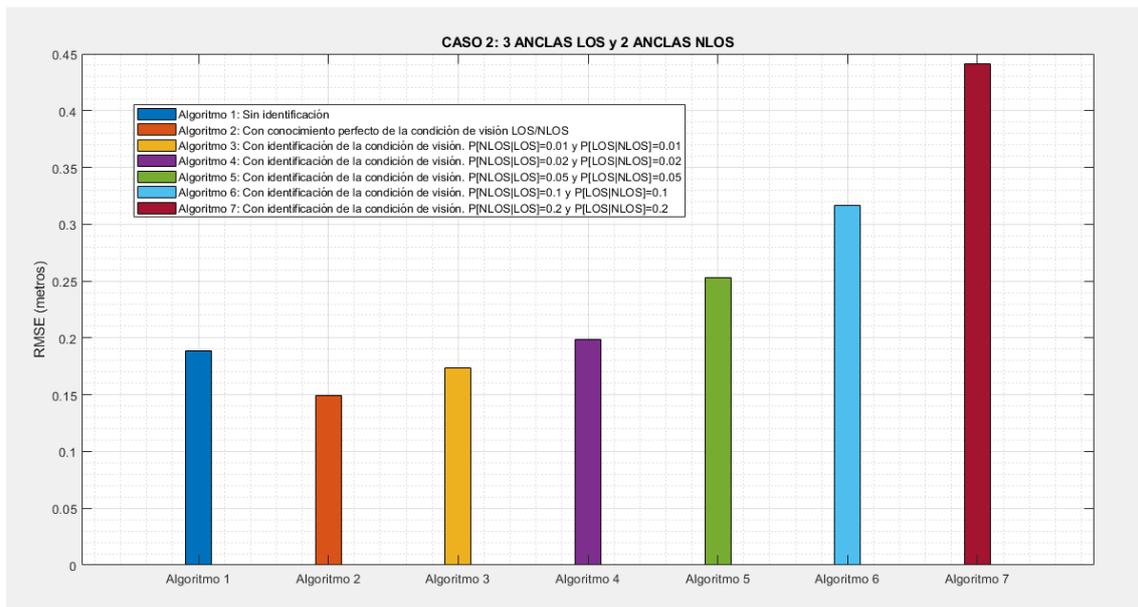


Figura 20: Representación gráfica del RMSE. Parejas de probabilidades de identificación errónea iguales. Caso 2: 3 anclas LOS y 2 anclas NLOS para $k=1000$
Figura realizada en MATLAB

En este caso también conseguimos un mejor $RMSE$ para $k=1000$, los resultados son más significativos. Es por ello que en estos casos de desglose, donde tenemos un menor número de posiciones reales del móvil a localizar, nos quedaremos con $k=1000$ para el algoritmo *con identificación de la condición de visión*, en el cual actúa una probabilidad de error.

Caso 3: 2 anclas LOS y 3 anclas NLOS

El RMSE de cada algoritmo para las 28 posiciones reales del móvil es el siguiente:

<i>ALGORITMOS</i>	<i>RMSE (metros)</i>
<i>Algoritmo 1: Sin identificación</i>	0.374
<i>Algoritmo 2: Con conocimiento perfecto de la condición de visión LOS/NLOS</i>	0.319
<i>Algoritmo 3: Con identificación de la condición de visión. $P[NLOS LOS]=0.01$ y $P[LOS NLOS]=0.01$</i>	0.355
<i>Algoritmo 4: Con identificación de la condición de visión. $P[NLOS LOS]=0.02$ y $P[LOS NLOS]=0.02$</i>	0.388
<i>Algoritmo 5: Con identificación de la condición de visión. $P[NLOS LOS]=0.05$ y $P[LOS NLOS]=0.05$</i>	0.456
<i>Algoritmo 6: Con identificación de la condición de visión. $P[NLOS LOS]=0.1$ y $P[LOS NLOS]=0.1$</i>	0.550
<i>Algoritmo 7: Con identificación de la condición de visión. $P[NLOS LOS]=0.2$ y $P[LOS NLOS]=0.2$</i>	0.663

Tabla 5: RMSE parejas de probabilidades de identificación errónea iguales. Caso 3: 2 anclas LOS y 3 anclas NLOS

Y su representación gráfica queda tal que así:

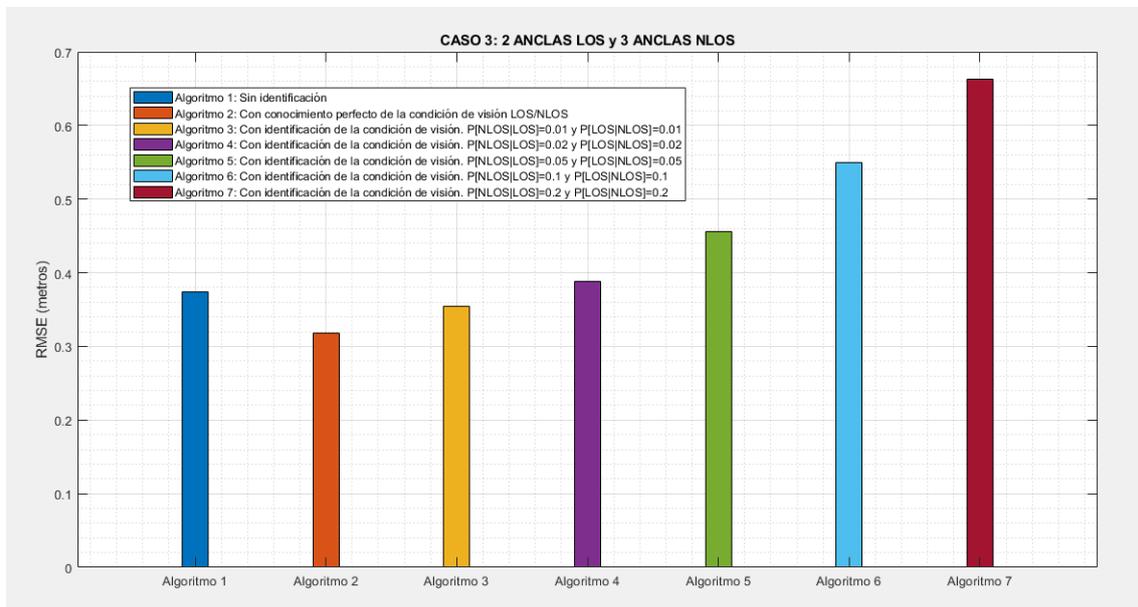


Figura 21: Representación gráfica del RMSE. Parejas de probabilidades de identificación errónea iguales. Caso 3: 2 anclas LOS y 3 anclas NLOS

Figura realizada en MATLAB

Con estos tres casos de desglose en nodos anclas LOS y NLOS, se demuestra la relevancia que tiene la línea de visión directa entre los nodos anclas y el nodo móvil en la localización. En estos casos LOS y NLOS, tenemos un mayor número de nodos anclas LOS por cada posición real del móvil, debido

a los casos que se han descartado, cobrando así las medidas de distancia *LOS* más protagonismo en cada estimación de la posición del nodo móvil.

4.2.2 RMSE para parejas de probabilidades de identificación errónea distintas

Para las siguientes parejas de probabilidades de identificación errónea,

- ❖ $P1[NLOS/LOS]=0.01$ y $P1[LOS/NLOS]=0.02$
- ❖ $P2[NLOS/LOS]=0.05$ y $P2[LOS/NLOS]=0.02$
- ❖ $P3[NLOS/LOS]=0.05$ y $P3[LOS/NLOS]=0.1$
- ❖ $P4[NLOS/LOS]=0.15$ y $P4[LOS/NLOS]=0.1$
- ❖ $P5[NLOS/LOS]=0.15$ y $P5[LOS/NLOS]=0.2$

el *RMSE* de cada algoritmo para las 57 posiciones reales del móvil es el siguiente:

<i>ALGORITMOS</i>	<i>RMSE (metros)</i>
<i>Algoritmo 1:</i> Sin identificación	1.245
<i>Algoritmo 2:</i> Con conocimiento perfecto de la condición de visión <i>LOS/NLOS</i>	0.884
<i>Algoritmo 3:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0.01$ y $P[LOS NLOS]=0.02$	0.925
<i>Algoritmo 4:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0.05$ y $P[LOS NLOS]=0.02$	0.939
<i>Algoritmo 5:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0.05$ y $P[LOS NLOS]=0.1$	1.020
<i>Algoritmo 6:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0.15$ y $P[LOS NLOS]=0.1$	1.085
<i>Algoritmo 7:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0.15$ y $P[LOS NLOS]=0.2$	1.248

Tabla 6: *RMSE* parejas de probabilidades de identificación errónea distintas

Cuya representación gráfica queda tal que así:

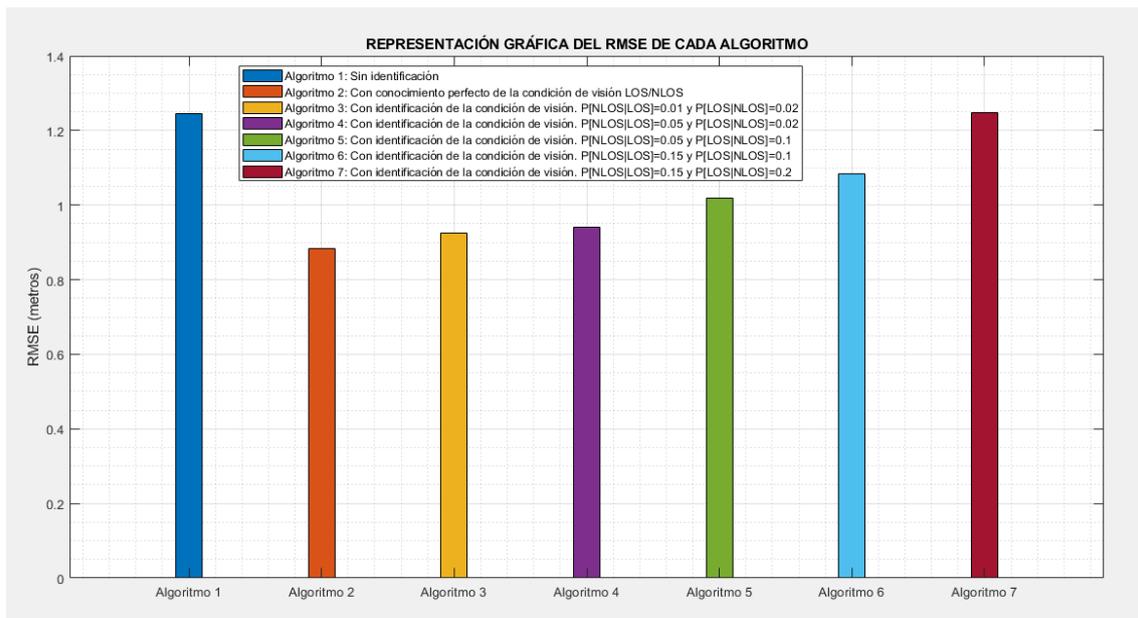


Figura 22: Representación gráfica del RMSE. Parejas de probabilidades de identificación errónea distintas
Figura realizada en MATLAB

Y la representación gráfica en 3D del RMSE en función de las probabilidades de identificación errónea es la siguiente:

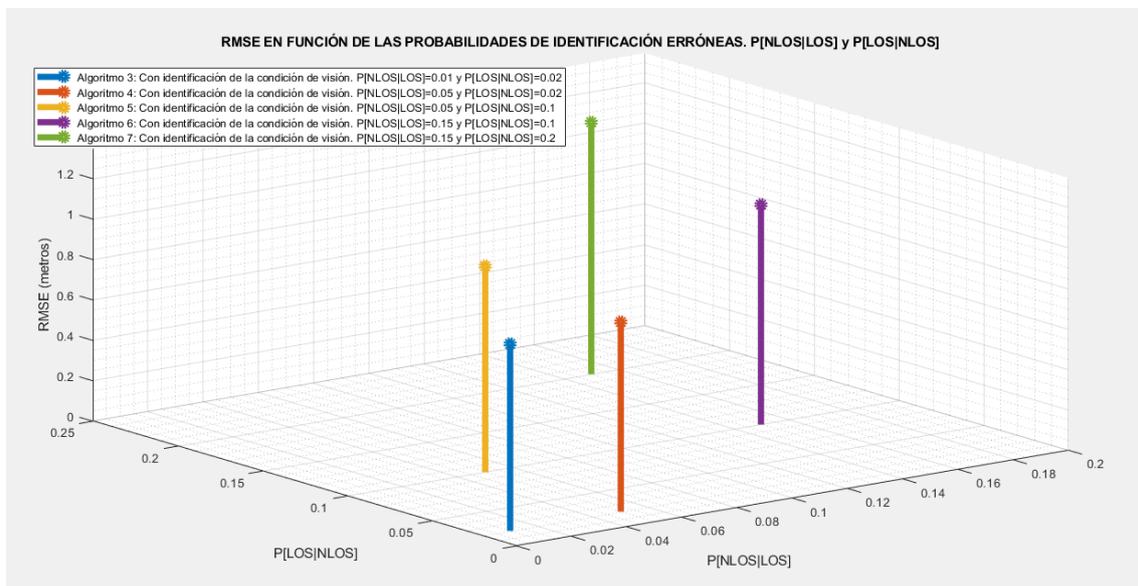


Figura 23: Representación gráfica del RMSE en función de las parejas de probabilidades de identificación errónea distintas
Figura realizada en MATLAB

Casos LOS/NLOS

Caso 1: 4 anclas LOS y 1 ancla NLOS

El RMSE de cada algoritmo para 1 posición real del móvil es el siguiente:

ALGORITMOS	RMSE (metros)
Algoritmo 1: Sin identificación	0.177
Algoritmo 2: Con conocimiento perfecto de la condición de visión LOS/NLOS	0.125
Algoritmo 3: Con identificación de la condición de visión. $P[NLOS LOS]=0.01$ y $P[LOS NLOS]=0.02$	0.133
Algoritmo 4: Con identificación de la condición de visión. $P[NLOS LOS]=0.05$ y $P[LOS NLOS]=0.02$	0.158
Algoritmo 5: Con identificación de la condición de visión. $P[NLOS LOS]=0.05$ y $P[LOS NLOS]=0.1$	0.161
Algoritmo 6: Con identificación de la condición de visión. $P[NLOS LOS]=0.15$ y $P[LOS NLOS]=0.1$	0.256
Algoritmo 7: Con identificación de la condición de visión. $P[NLOS LOS]=0.15$ y $P[LOS NLOS]=0.2$	0.633

Tabla 7: RMSE parejas de probabilidades de identificación errónea distintas. Caso 1: 4 anclas LOS y 1 ancla NLOS

Y su representación gráfica queda tal que así:

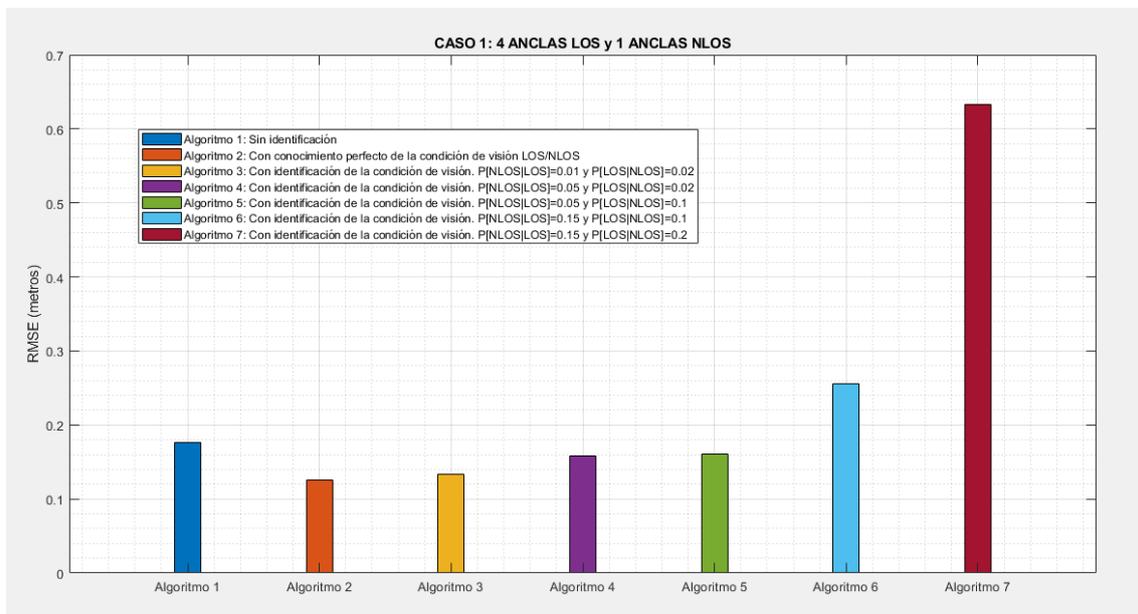


Figura 24: Representación gráfica del RMSE. Parejas de probabilidades de identificación errónea distintas. Caso 1: 4 anclas LOS y 1 ancla NLOS

Figura realizada en MATLAB

Caso 2: 3 anclas LOS y 2 anclas NLOS

El RMSE de cada algoritmo para las 7 posiciones reales del móvil es el siguiente:

<i>ALGORITMOS</i>	<i>RMSE (metros)</i>
<i>Algoritmo 1: Sin identificación</i>	0.189
<i>Algoritmo 2: Con conocimiento perfecto de la condición de visión LOS/NLOS</i>	0.149
<i>Algoritmo 3: Con identificación de la condición de visión. $P[NLOS LOS]=0.01$ y $P[LOS NLOS]=0.02$</i>	0.182
<i>Algoritmo 4: Con identificación de la condición de visión. $P[NLOS LOS]=0.05$ y $P[LOS NLOS]=0.02$</i>	0.248
<i>Algoritmo 5: Con identificación de la condición de visión. $P[NLOS LOS]=0.05$ y $P[LOS NLOS]=0.1$</i>	0.264
<i>Algoritmo 6: Con identificación de la condición de visión. $P[NLOS LOS]=0.15$ y $P[LOS NLOS]=0.1$</i>	0.375
<i>Algoritmo 7: Con identificación de la condición de visión. $P[NLOS LOS]=0.15$ y $P[LOS NLOS]=0.2$</i>	0.401

Tabla 8: RMSE parejas de probabilidades de identificación errónea distintas. Caso 2: 3 anclas LOS y 2 anclas NLOS

Y su representación gráfica queda tal que así:

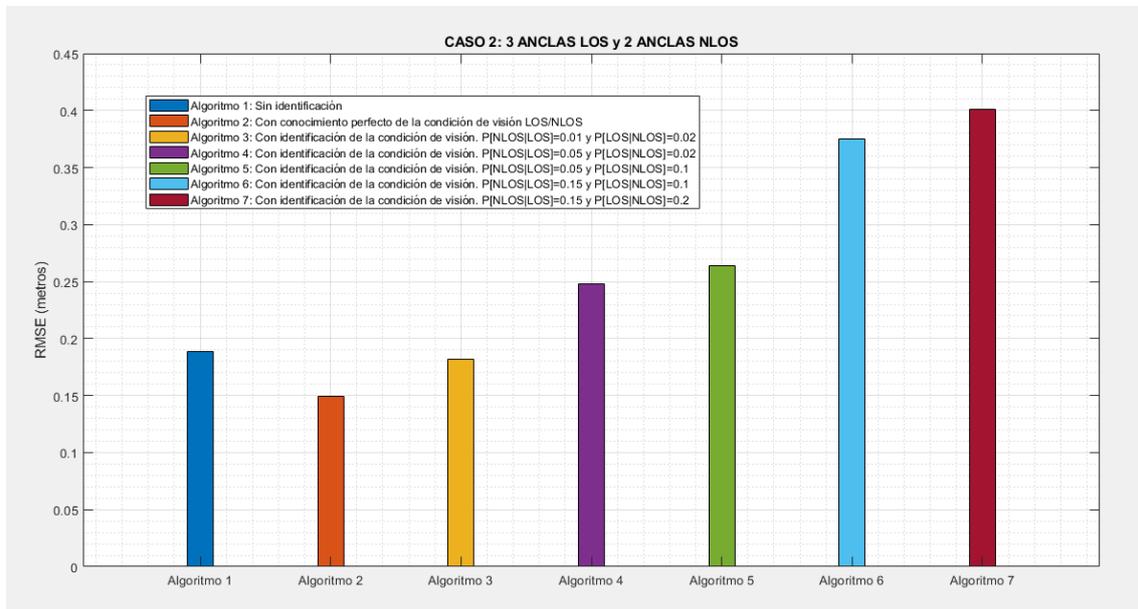


Figura 25: Representación gráfica del RMSE. Parejas de probabilidades de identificación errónea distintas. Caso 2: 3 anclas LOS y 2 anclas NLOS

Figura realizada en MATLAB

Caso 3: 2 anclas LOS y 3 anclas NLOS

El RMSE de cada algoritmo para las 28 posiciones reales del móvil es el siguiente:

<i>ALGORITMOS</i>	<i>RMSE (metros)</i>
<i>Algoritmo 1: Sin identificación</i>	0.374
<i>Algoritmo 2: Con conocimiento perfecto de la condición de visión LOS/NLOS</i>	0.319
<i>Algoritmo 3: Con identificación de la condición de visión. $P[NLOS LOS]=0.01$ y $P[LOS NLOS]=0.02$</i>	0.379
<i>Algoritmo 4: Con identificación de la condición de visión. $P[NLOS LOS]=0.05$ y $P[LOS NLOS]=0.02$</i>	0.402
<i>Algoritmo 5: Con identificación de la condición de visión. $P[NLOS LOS]=0.05$ y $P[LOS NLOS]=0.1$</i>	0.538
<i>Algoritmo 6: Con identificación de la condición de visión. $P[NLOS LOS]=0.15$ y $P[LOS NLOS]=0.1$</i>	0.554
<i>Algoritmo 7: Con identificación de la condición de visión. $P[NLOS LOS]=0.15$ y $P[LOS NLOS]=0.2$</i>	0.652

Tabla 9: RMSE parejas de probabilidades de identificación errónea distintas. Caso 3: 2 anclas LOS y 3 anclas NLOS

Y su representación gráfica queda tal que así:

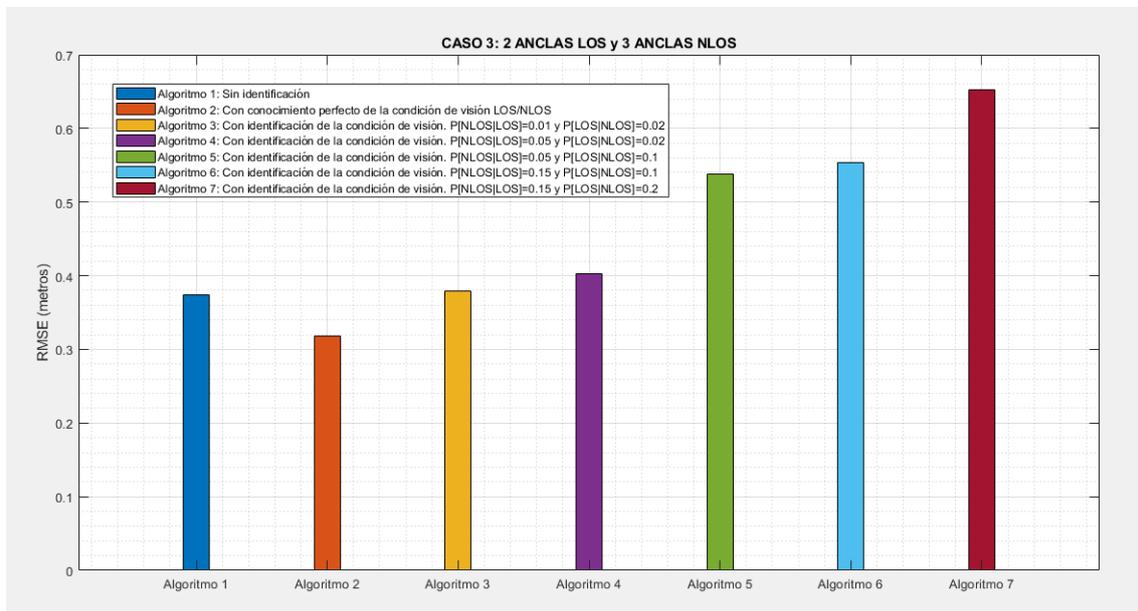


Figura 26: Representación gráfica del RMSE. Parejas de probabilidades de identificación errónea distintas. Caso 3: 2 anclas LOS y 3 anclas NLOS

Figura realizada en MATLAB

4.2.3 RMSE fijando una probabilidad de identificación errónea igual a cero

$$P[NLOS|LOS]=0.$$

Para las siguientes parejas de probabilidades de identificación errónea,

- ❖ $P1[NLOS/LOS]=0$ y $P1[LOS/NLOS]=0.01$
- ❖ $P2[NLOS/LOS]=0$ y $P2[LOS/NLOS]=0.02$
- ❖ $P3[NLOS/LOS]=0$ y $P3[LOS/NLOS]=0.05$
- ❖ $P4[NLOS/LOS]=0$ y $P4[LOS/NLOS]=0.1$
- ❖ $P5[NLOS/LOS]=0$ y $P5[LOS/NLOS]=0.2$

el RMSE de cada algoritmo para las 57 posiciones reales del móvil es el siguiente:

ALGORITMOS	RMSE (metros)
<i>Algoritmo 1:</i> Sin identificación	1.245
<i>Algoritmo 2:</i> Con conocimiento perfecto de la condición de visión LOS/NLOS	0.884
<i>Algoritmo 3:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0$ y $P[LOS NLOS]=0.01$	0.895
<i>Algoritmo 4:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0$ y $P[LOS NLOS]=0.02$	0.904
<i>Algoritmo 5:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0$ y $P[LOS NLOS]=0.05$	0.934
<i>Algoritmo 6:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0$ y $P[LOS NLOS]=0.1$	0.987
<i>Algoritmo 7:</i> Con identificación de la condición de visión. $P[NLOS LOS]=0$ y $P[LOS NLOS]=0.2$	1.049

Tabla 10: RMSE fijando una probabilidad de identificación errónea igual a cero. $P[NLOS|LOS]=0$

Cuya representación gráfica queda tal que así:

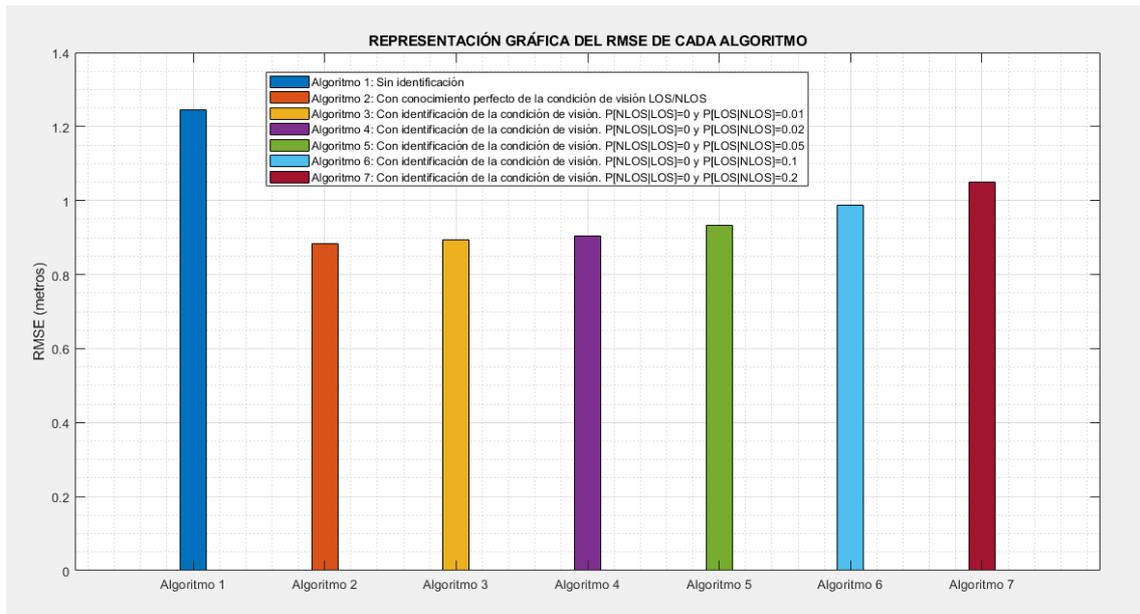


Figura 27: Representación gráfica del RMSE fijando una probabilidad de identificación errónea igual a cero. $P[NLOS|LOS]=0$

Figura realizada en MATLAB

Y la representación gráfica en 3D del RMSE en función de las probabilidades de identificación errónea es la siguiente:

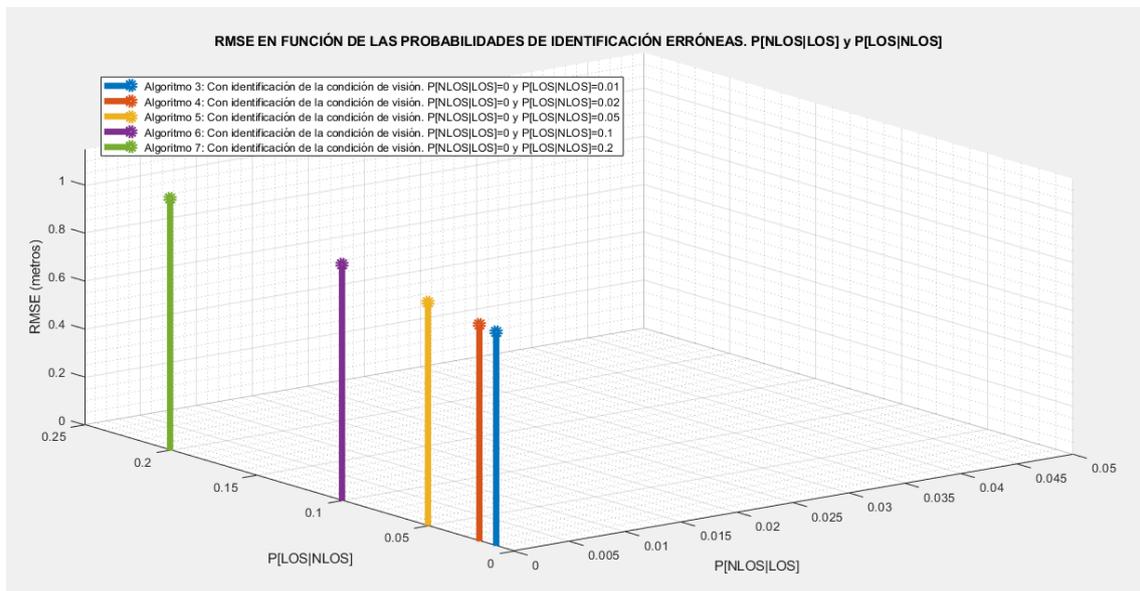


Figura 28: Representación gráfica del RMSE en función de una probabilidad de identificación errónea igual a cero. $P[NLOS|LOS]=0$

Figura realizada en MATLAB

P[LOS|NLOS]=0

Para las siguientes parejas de probabilidades de identificación errónea,

- ❖ P1[NLOS/LOS]=0.01 y P1[LOS/NLOS]=0
- ❖ P2[NLOS/LOS]=0.02 y P2[LOS/NLOS]=0
- ❖ P3[NLOS/LOS]=0.05 y P3[LOS/NLOS]=0
- ❖ P4[NLOS/LOS]=0.1 y P4[LOS/NLOS]=0
- ❖ P5[NLOS/LOS]=0.2 y P5[LOS/NLOS]=0

el RMSE de cada algoritmo para las 57 posiciones reales del móvil es el siguiente:

ALGORITMOS	RMSE (metros)
Algoritmo 1: Sin identificación	1.245
Algoritmo 2: Con conocimiento perfecto de la condición de visión LOS/NLOS	0.884
Algoritmo 3: Con identificación de la condición de visión. P[NLOS/LOS]=0.01 y P[LOS NLOS]=0	0.892
Algoritmo 4: Con identificación de la condición de visión. P[NLOS/LOS]=0.02 y P[LOS NLOS]=0	0.899
Algoritmo 5: Con identificación de la condición de visión. P[NLOS/LOS]=0.05 y P[LOS NLOS]=0	0.919
Algoritmo 6: Con identificación de la condición de visión. P[NLOS/LOS]=0.1 y P[LOS NLOS]=0	0.956
Algoritmo 7: Con identificación de la condición de visión. P[NLOS/LOS]=0.2 y P[LOS NLOS]=0	1.014

Tabla 11: RMSE fijando una probabilidad de identificación errónea igual a cero. P[LOS|NLOS]=0

Cuya representación gráfica queda tal que así:

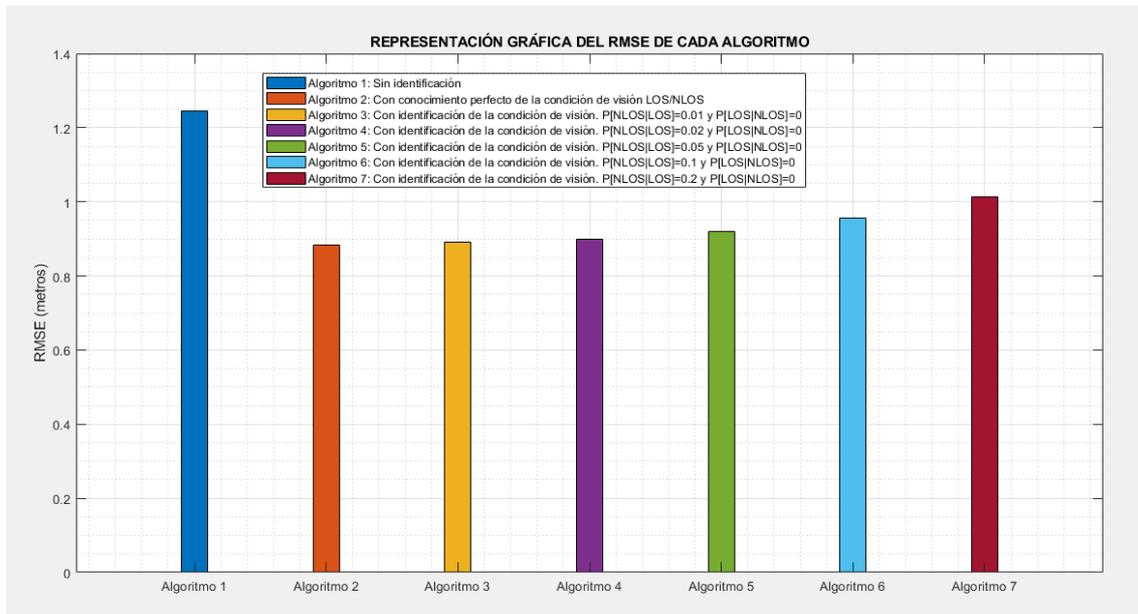


Figura 29: Representación gráfica del RMSE fijando una probabilidad de identificación errónea igual a cero. $P[LOS|NLOS]=0$

Figura realizada en MATLAB

Y la representación gráfica en 3D del RMSE en función de las probabilidades de identificación errónea es la siguiente:

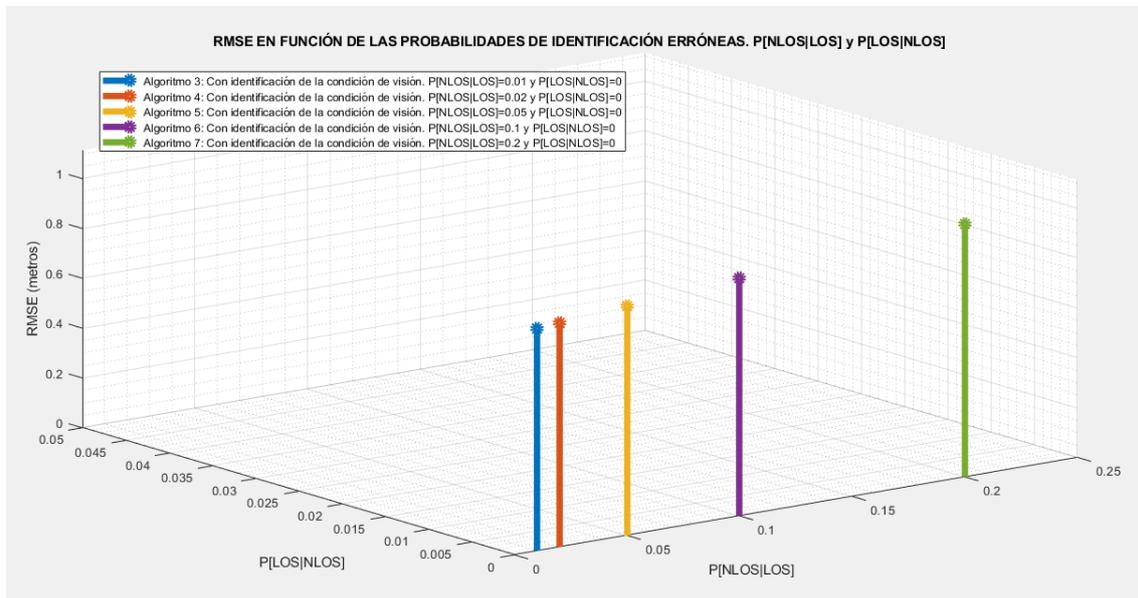


Figura 30: Representación gráfica del RMSE en función de una probabilidad de identificación errónea igual a cero. $P[LOS|NLOS]=0$

Figura realizada en MATLAB

Los resultados que obtenemos en estos dos apartados donde fijamos una probabilidad de identificación errónea igual a cero son coherentes, ya que, si una probabilidad de error es cero, se obtiene un menor RMSE que en los casos donde todas las parejas de probabilidades son distintas de cero.

También se observa que, $P[LOS|NLOS]$, es decir, la probabilidad de identificar una medida de

distancia *NLOS* como *LOS*, es más relevante, ya que afecta más a la localización al dar ligeramente un mayor *RMSE* cuando es distinta de cero. Entonces, cuando $P[LOS|NLOS]=0$ se obtiene menor *RMSE* que cuando $P[NLOS|LOS]=0$.

4.2.4 RMSE en función de todas las probabilidades de identificación errónea

A continuación, se representa gráficamente en 3D el *RMSE* en función de todas las probabilidades de identificación errónea definidas en los puntos anteriores, para todas las posiciones reales del móvil a localizar. Por lo tanto, para todas las parejas de probabilidad que hemos definido en este proyecto,

- ❖ $P1[NLOS|LOS]=0.01$ y $P1[LOS|NLOS]=0.01$
- ❖ $P2[NLOS|LOS]=0.02$ y $P2[LOS|NLOS]=0.02$
- ❖ $P3[NLOS|LOS]=0.05$ y $P3[LOS|NLOS]=0.05$
- ❖ $P4[NLOS|LOS]=0.1$ y $P4[LOS|NLOS]=0.1$
- ❖ $P5[NLOS|LOS]=0.2$ y $P5[LOS|NLOS]=0.2$
- ❖ $P6[NLOS|LOS]=0.01$ y $P6[LOS|NLOS]=0.02$
- ❖ $P7[NLOS|LOS]=0.05$ y $P7[LOS|NLOS]=0.02$
- ❖ $P8[NLOS|LOS]=0.05$ y $P8[LOS|NLOS]=0.1$
- ❖ $P9[NLOS|LOS]=0.15$ y $P9[LOS|NLOS]=0.1$
- ❖ $P10[NLOS|LOS]=0.15$ y $P10[LOS|NLOS]=0.2$
- ❖ $P11[NLOS|LOS]=0$ y $P11[LOS|NLOS]=0.01$
- ❖ $P12[NLOS|LOS]=0$ y $P12[LOS|NLOS]=0.02$
- ❖ $P13[NLOS|LOS]=0$ y $P13[LOS|NLOS]=0.05$
- ❖ $P14[NLOS|LOS]=0$ y $P14[LOS|NLOS]=0.1$
- ❖ $P15[NLOS|LOS]=0$ y $P15[LOS|NLOS]=0.2$
- ❖ $P16[NLOS|LOS]=0.01$ y $P16[LOS|NLOS]=0$
- ❖ $P17[NLOS|LOS]=0.02$ y $P17[LOS|NLOS]=0$
- ❖ $P18[NLOS|LOS]=0.05$ y $P18[LOS|NLOS]=0$
- ❖ $P19[NLOS|LOS]=0.1$ y $P19[LOS|NLOS]=0$
- ❖ $P20[NLOS|LOS]=0.2$ y $P20[LOS|NLOS]=0$

tendremos una tabla de valores de *RMSE* como la siguiente:

<i>ALGORITMOS</i>	<i>RMSE (metros)</i>
Con identificación de la condición de visión. $P1[NLOS LOS]=0.01$ y $P1[LOS NLOS]=0.01$	0.915
Con identificación de la condición de visión. $P2[NLOS LOS]=0.02$ y $P2[LOS NLOS]=0.02$	0.919
Con identificación de la condición de visión. $P3[NLOS LOS]=0.05$ y $P3[LOS NLOS]=0.05$	0.968
Con identificación de la condición de visión. $P4[NLOS LOS]=0.1$ y $P4[LOS NLOS]=0.1$	1.060
Con identificación de la condición de visión. $P5[NLOS LOS]=0.2$ y $P5[LOS NLOS]=0.2$	1.280
Con identificación de la condición de visión. $P6[NLOS LOS]=0.01$ y $P6[LOS NLOS]=0.02$	0.925

Con identificación de la condición de visión. P7[NLOS LOS]=0.05 y P7[LOS NLOS]=0.02	0.939
Con identificación de la condición de visión. P8[NLOS LOS]=0.05 y P8[LOS NLOS]=0.1	1.020
Con identificación de la condición de visión. P9[NLOS LOS]=0.15 y P9[LOS NLOS]=0.1	1.085
Con identificación de la condición de visión. P10[NLOS LOS]=0.15 y P10[LOS NLOS]=0.2	1.248
Con identificación de la condición de visión. P11[NLOS LOS]=0 y P11[LOS NLOS]=0.01	0.895
Con identificación de la condición de visión. P12[NLOS LOS]=0 y P12[LOS NLOS]=0.02	0.904
Con identificación de la condición de visión. P13[NLOS LOS]=0 y P13[LOS NLOS]=0.05	0.934
Con identificación de la condición de visión. P14[NLOS LOS]=0 y P14[LOS NLOS]=0.1	0.987
Con identificación de la condición de visión. P15[NLOS LOS]=0 y P15[LOS NLOS]=0.2	1.049
Con identificación de la condición de visión. P16[NLOS LOS]=0.01 y P16[LOS NLOS]=0	0.892
Con identificación de la condición de visión. P17[NLOS LOS]=0.02 y P17[LOS NLOS]=0	0.899
Con identificación de la condición de visión. P18[NLOS LOS]=0.05 y P18[LOS NLOS]=0	0.919
Con identificación de la condición de visión. P19[NLOS LOS]=0.1 y P19[LOS NLOS]=0	0.956
Con identificación de la condición de visión. P20[NLOS LOS]=0.2 y P20[LOS NLOS]=0	1.014

Tabla 12: RMSE todas las parejas de probabilidades de identificación errónea

Y su representación gráfica es la siguiente:

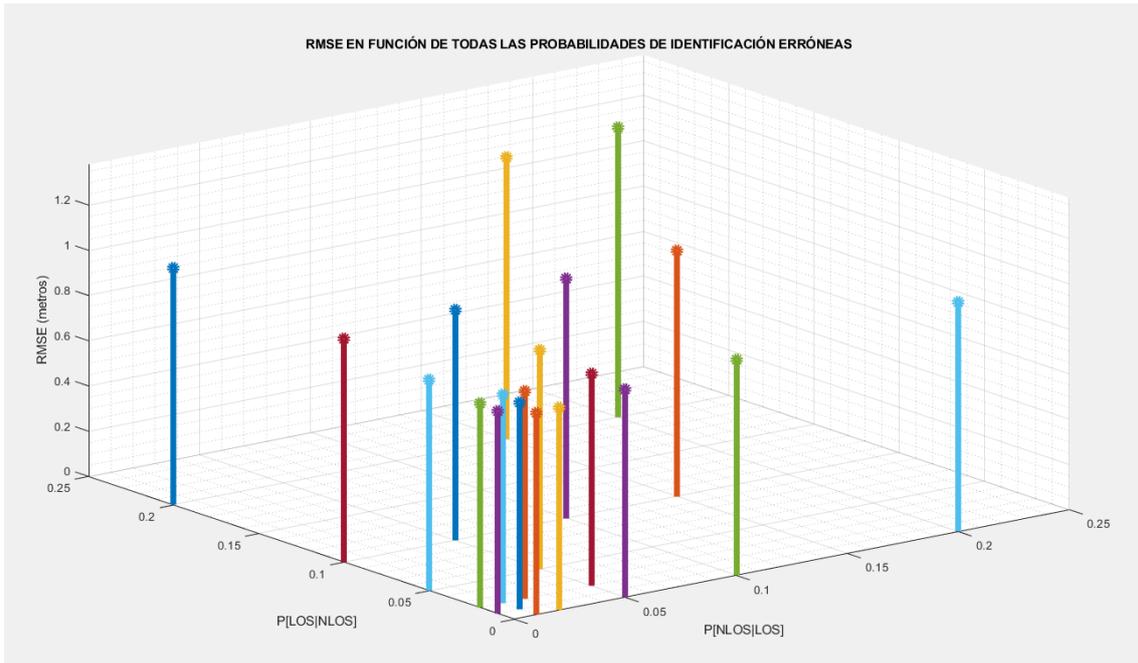


Figura 31: Representación gráfica del RMSE en función de todas las probabilidades de identificación errónea
Figura realizada en MATLAB

5 CONCLUSIONES Y LÍNEAS FUTURAS

5.1 Conclusiones

Con las simulaciones realizadas y los resultados obtenidos durante el desarrollo de este proyecto se obtienen las siguientes conclusiones para el conjunto de medidas reales que se han empleado:

- El *algoritmo sin identificación* no hace distinción entre medidas de distancia con línea de visión directa, *LOS*, o medidas de distancia en las que no hay línea de visión directa, *NLOS*, con el nodo móvil. Por lo tanto, si el *RMSE* que se obtiene con el algoritmo *con identificación de la condición de visión* no queremos que sea mayor que el *RMSE* obtenido con el *algoritmo sin identificación*, las probabilidades de identificación errónea no deben ser iguales o superiores al 20%, ya que obtendremos una estimación de la posición del nodo móvil desfavorable.
- En los casos de desglose en nodos anclas *LOS* y *NLOS*, casos donde tenemos un menor número de posiciones reales del nodo móvil a localizar, si empleamos el algoritmo *con identificación de la condición de visión* en el cual, actúa una probabilidad de identificación errónea de la condición de visión de la medida de distancia, tendremos que ejecutarlos más veces para cada posición real del móvil. Es por ello que, en este trabajo, concluimos que con $k=1000$ se obtienen resultados más significativos que para $k=500$, pero para valores superiores a $k=1000$ ya no se aprecian diferencias relevantes.
- Cuanto mayor sea el número de nodos anclas *LOS* que tengamos en la localización del nodo móvil, menor será el *RMSE* obtenido. Esto es coherente en un sistema de localización en interiores ya que los nodos anclas *LOS*, aquellos nodos anclas que tienen línea de visión directa con el nodo móvil a localizar, son los que van a estimar una medida de distancia más próxima a la real y, por consiguiente, van a aportar una mayor precisión en el cálculo de la posición. Por lo tanto, cuanto mayor número de anclas *LOS* tengamos, mayor precisión va a tener el sistema en la localización.
- $P[LOS|NLOS]$, la probabilidad de identificar una medida de distancia *NLOS* como *LOS*, tiene ligeramente más peso en la localización, ya que da un valor de *RMSE* mayor que $P[NLOS|LOS]$, la probabilidad de identificar una medida *LOS* como *NLOS*. Esto ocurre porque cuando $P[LOS|NLOS]$, es distinta de cero, habrá casos en los cuales el algoritmo de localización interprete una medida de distancia *NLOS* como *LOS*. Entonces, dicha medida de distancia no se suaviza con $\beta = 0.001$, por lo tanto, las medidas de distancia que sí son *LOS* no cobran tanto protagonismo en la localización, cobrando dicho protagonismo las medidas de distancia *NLOS* y, como resultado, se estimará una posición del nodo móvil desfavorable.

5.2 Líneas Futuras

En este apartado se han empleado las siguientes referencias: [14], [15] y [16].

La tecnología de banda ultra ancha, *UWB*, es una muy buena tecnología para la localización en interiores de edificios, aportando un gran potencial. Es por ello que, múltiples de empresas y entidades apuestan por ella para la localización con diferentes finalidades.

Empresas de la telefonía móvil, como *Apple* y *Samsung*, la incorporan en algunos de sus modelos más nuevos para la localización y comunicación entre sus dispositivos como alternativa al bluetooth.

También, la industria automovilística está apostando por la tecnología *Ultra Wide Band*, empresas como *Volkswagen* la están empleando en sus vehículos, permitiendo que, a través de la comunicación entre las llaves del coche con el propio coche, se pueda localizar el vehículo.

La *NFL*, la *Liga Nacional de Fútbol Americano*, la emplea para la localización de sus jugadores durante los partidos. También se apuesta por la tecnología *UWB* en empresas para la localización de la mercancía en sus grandes almacenes.

La tecnología de banda ultra ancha ha llegado para quedarse, aunque todavía falta mucho por avanzar en el enorme potencial que aporta, pero lo que está claro que tendrá un impacto muy positivo en el sector de las telecomunicaciones en un futuro no muy lejano, consiguiendo mejorar nuestra calidad de vida al darnos más facilidades en nuestro día a día.

BIBLIOGRAFÍA Y REFERENCIAS

- [1] *Sistemas de posicionamiento en interiores para APPs con navegación en interiores y localización de empleados.* (2020, 11 agosto). SITUM. Último acceso: 09/05/2022. <https://situm.com/es/blog/posicionamiento-en-interiores/posicionamiento-en-interiores-para-apps-con-navegacion-en-interiores-y-localizacion-de-empleados/?cn-reloaded=1>
- [2] Innovation, A. (2021, 20 mayo). *Sistemas de localización en interiores.* ATRIA Innovation. Último acceso: 09/05/2022. <https://www.atriainnovation.com/sistemas-localizacion-interiores/>
- [3] Compte, M. (2019, 11 julio). *Posicionamiento Indoor.* UNIGIS. Último acceso: 09/05/2022. <https://www.unigis.es/posicionamiento-indoor/>
- [4] *Posicionamiento de interiores – Las 4 tecnologías punteras.* (s. f.). Philips. Último acceso: 09/05/2022. <https://www.lighting.philips.es/soporte/contacto/tendencias-en-iluminacion/para-los-profesionales/indoor-positioning-posicionamiento-interiores>
- [5] Fernández, C. (2022, 18 febrero). *¿Qué son los sistemas de localización en interiores?* ABAMOBILE. Último acceso: 09/05/2022. <https://abamobile.com/web/que-son-sistemas-localizacion-en-interiores/>
- [6] G. (2018, 29 mayo). *¿Cómo funcionan los dispositivos GPS? Trilateración vs Triangulación.* El blog de franz. Último acceso: 09/05/2022. <https://acolita.com/como-funcionan-los-dispositivos-gps-trilateracion-vs-triangulacion/>
- [7] *Triangulación.* (2021, 11 abril). Wikipedia, la enciclopedia libre. Último acceso: 09/05/2022. <https://es.wikipedia.org/wiki/Triangulaci%C3%B3n>
- [8] *Trilateración.* (2022, 14 febrero). Wikipedia, la enciclopedia libre. Último acceso: 09/05/2022. <https://es.wikipedia.org/wiki/Trilateraci%C3%B3n>
- [9] L. Díaz-Ambrona, *Sistema de localización en interiores.* Universidad Politécnica de Madrid, 2008.
- [10] C. García, *Mitigación de los errores por obstrucción de la línea de visión directa en un sistema de localización.* Universidad de Sevilla, 2019.
- [11] EVK1000 Evaluation Kit. (2019, 2 mayo). Decawave. Último acceso: 09/05/2022. <https://www.decawave.com/product/evk1000-evaluation-kit/>
- [12] M. J. Madero, *Apuntes de Sistemas Emergentes de Comunicaciones.* Universidad de Sevilla, 2019-2020.
- [13] *Qué es... UWB (Ultra Wide-Band).* (s. f.). Ramón Millán. Último acceso: 09/05/2022. <https://www.ramonmillan.com/tutoriales/ultrawideband.php>
- [14] Romero, M. S. (2020, 24 octubre). *¿Qué es la UWB y cómo podríamos utilizarla?* Computer Hoy. Último acceso: 09/05/2022. <https://computerhoy.com/reportajes/tecnologia/ubw-que-es-742355>
- [15] Tu.com. (2021, 23 marzo). *Ultra Wide Band: La nueva tecnología por la que apuestan Apple y Samsung.* Blogthinkbig.com. Último acceso: 09/05/2022. <https://blogthinkbig.com/ultra-wideband>
- [16] Pérez, E. (2019, 12 octubre). *¿Qué ventajas tiene la tecnología de banda ultra ancha (UWB): una vieja alternativa al Bluetooth.* Xataka. Último acceso: 09/05/2022. <https://www.xataka.com/servicios/que-ventajas-tiene-tecnologia-banda-ultrancha-uwv-vieja-alternativa-al-bluetooth-que-apple-samsung-estan-revitalizando>
- [17] I. Guvenc y C. C. Chong, *A Survey on TOA Based Wireless Localization and NLOS Mitigation Techniques.* IEEE Communications Survey and Tutorials, 2009.

ANEXO A: CÓDIGOS MATLAB ALGORITMOS Y GRÁFICAS

En esta sección se muestran los códigos MATLAB de los algoritmos de localización y de las gráficas que se han implementado para el desarrollo de este trabajo.

A.1 Algoritmo sin identificación

```
% ALGORITMO SIN IDENTIFICACIÓN

function [x_estimado, y_estimado] = sinIdentificacion (n_cruces_x, n_cruces_y, x_cruz,
y_cruz, n_anclas, m, n, d, LOS_NLOS)

z = @(x,y) 0;
Guvenc = @(x,y) 0;
for i=1:n_anclas
    if d(i) ~= 0 && LOS_NLOS(i) ~= -1 % Nos quedamos con las medidas LOS y NLOS
        z = @(x,y) (d(i)-sqrt((x-m(i))^2 + (y-n(i))^2))^2;
        Guvenc = @(x,y) Guvenc(x,y) + z(x,y);
    end
end

posibles_min = zeros(1,n_cruces_x*n_cruces_y);
for i=1:n_cruces_x*n_cruces_y
    posibles_min(i) = Guvenc(x_cruz(i),y_cruz(i));
end

% Elegimos la x_cruz e y_cruz que da la mínima suma de las Cruces-Anclas distancias:
[valor_min, posicion_min] = min(posibles_min);
x_estimado = x_cruz(posicion_min);
y_estimado = y_cruz(posicion_min);

end
```

A.2 Algoritmo con conocimiento perfecto de la condición de visión LOS o NLOS

```
% ALGORITMO CON CONOCIMIENTO PERFECTO DE LA CONDICIÓN DE VISIÓN LOS/NLOS

function [x_estimado, y_estimado] = conocimientoPerfectoLOS_NLOS (n_cruces_x,
n_cruces_y, x_cruz, y_cruz, n_anclas, m, n, d, LOS_NLOS, beta_LOS, beta_NLOS)

z = @(x,y) 0;
Guvenc = @(x,y) 0;
for i=1:n_anclas
    if LOS_NLOS(i) == 1 % La medida Ancla(i)-Móvil es LOS
        beta_i = beta_LOS;
    end
    if LOS_NLOS(i) == 0 % La medida Ancla(i)-Móvil es NLOS
        beta_i = beta_NLOS;
    end
    if LOS_NLOS(i) == -1 || d(i) == 0 % No hay comunicación entre Ancla(i)-Móvil o
    coinciden en posición
        beta_i = 0; % Se descarta la medida
    end
    z = @(x,y) beta_i*( (d(i)-sqrt((x-m(i))^2 + (y-n(i))^2))^2 );
    Guvenc = @(x,y) Guvenc(x,y) + z(x,y);
end

end
```

```

posibles_min = zeros(1,n_cruces_x*n_cruces_y);
for i=1:n_cruces_x*n_cruces_y
    posibles_min(i) = Guvenc(x_cruz(i),y_cruz(i));
end

% Elegimos la x_cruz e y_cruz que da la mínima suma de las Cruces-Anclas distancias:
[valor_min, posicion_min] = min(posibles_min);
x_estimado = x_cruz(posicion_min);
y_estimado = y_cruz(posicion_min);

end

```

A.3 Algoritmo con identificación de la condición de visión. $P[NLOS|LOS]$ y $P[LOS|NLOS]$

```

% ALGORITMO CON IDENTIFICACIÓN DE LA CONDICIÓN DE VISIÓN.  $P[NLOS|LOS]$  y  $P[LOS|NLOS]$ 

function [x_estimado, y_estimado] = identificacionCondicionVision (n_cruces_x,
n_cruces_y, x_cruz, y_cruz, n_anclas, m, n, d, LOS_NLOS, beta_LOS, beta_NLOS,
P_NLOS_LOS, P_LOS_NLOS)

z = @(x,y) 0;
Guvenc = @(x,y) 0;
for i=1:n_anclas
    numeroAleatorio = rand;
    if LOS_NLOS(i) == 1 % La medida Ancla(i)-Móvil es LOS
        if numeroAleatorio <= P_NLOS_LOS % Identificación de la medida LOS como NLOS
            beta_i = beta_NLOS; % Identificación incorrecta
        else % numeroAleatorio > P[NLOS|LOS]: Identificación de la medida LOS como LOS
            beta_i = beta_LOS; % Identificación correcta
        end
    end
    if LOS_NLOS(i) == 0 % La medida Ancla(i)-Móvil es NLOS
        if numeroAleatorio <= P_LOS_NLOS % Identificación de la medida NLOS como LOS
            beta_i = beta_LOS; % Identificación incorrecta
        else % numeroAleatorio > P[LOS|NLOS]: Identificación de la medida NLOS como
            NLOS
            beta_i = beta_NLOS; % Identificación correcta
        end
    end
    if LOS_NLOS(i) == -1 || d(i) == 0 % No hay comunicación entre Ancla(i)-Móvil o
    coinciden en posición
        beta_i = 0; % Se descarta la medida
    end
    z = @(x,y) beta_i*( (d(i)-sqrt((x-m(i))^2 + (y-n(i))^2))^2 );
    Guvenc = @(x,y) Guvenc(x,y) + z(x,y);
end

posibles_min = zeros(1,n_cruces_x*n_cruces_y);
for i=1:n_cruces_x*n_cruces_y
    posibles_min(i) = Guvenc(x_cruz(i),y_cruz(i));
end

% Elegimos la x_cruz e y_cruz que da la mínima suma de las Cruces-Anclas distancias:
[valor_min, posicion_min] = min(posibles_min);
x_estimado = x_cruz(posicion_min);
y_estimado = y_cruz(posicion_min);

end

```

A.4 Representación gráfica del RMSE de cada algoritmo

```

% REPRESENTACIÓN GRÁFICA DEL RMSE DE CADA ALGORITMO

function [] = graficaRMSE (RMSE, P, titulo)

    figure('name', titulo)

    ancho_barras = 0.2;

    for i = 1:length(RMSE)
        bar(i, RMSE(i), ancho_barras);
        hold on
    end

    grid on
    grid minor
    xticks([1 2 3 4 5 6 7])
    xticklabels({'Algoritmo 1','Algoritmo 2','Algoritmo 3','Algoritmo 4','Algoritmo
5','Algoritmo 6','Algoritmo 7'})
    ylabel('RMSE (metros)')
    title(titulo)

    % Creación de la leyenda:

    Leyenda_Alg_1 = 'Algoritmo 1: Sin identificación';

    Leyenda_Alg_2 = 'Algoritmo 2: Con conocimiento perfecto de la condición de visión
LOS/NLOS';

    Texto_1 = 'Algoritmo 3: Con identificación de la condición de visión.
P[NLOS|LOS]=';
    P1_NLOS_LOS = num2str(P(1));
    Texto = ' y P[LOS|NLOS]=';
    P1_LOS_NLOS = num2str(P(2));
    Leyenda_Alg_3 = strcat(Texto_1,P1_NLOS_LOS,Texto,P1_LOS_NLOS);

    Texto_2 = 'Algoritmo 4: Con identificación de la condición de visión.
P[NLOS|LOS]=';
    P2_NLOS_LOS = num2str(P(3));
    P2_LOS_NLOS = num2str(P(4));
    Leyenda_Alg_4 = strcat(Texto_2,P2_NLOS_LOS,Texto,P2_LOS_NLOS);

    Texto_3 = 'Algoritmo 5: Con identificación de la condición de visión.
P[NLOS|LOS]=';
    P3_NLOS_LOS = num2str(P(5));
    P3_LOS_NLOS = num2str(P(6));
    Leyenda_Alg_5 = strcat(Texto_3,P3_NLOS_LOS,Texto,P3_LOS_NLOS);

    Texto_4 = 'Algoritmo 6: Con identificación de la condición de visión.
P[NLOS|LOS]=';
    P4_NLOS_LOS = num2str(P(7));
    P4_LOS_NLOS = num2str(P(8));
    Leyenda_Alg_6 = strcat(Texto_4,P4_NLOS_LOS,Texto,P4_LOS_NLOS);

    Texto_5 = 'Algoritmo 7: Con identificación de la condición de visión.
P[NLOS|LOS]=';
    P5_NLOS_LOS = num2str(P(9));
    P5_LOS_NLOS = num2str(P(10));
    Leyenda_Alg_7 = strcat(Texto_5,P5_NLOS_LOS,Texto,P5_LOS_NLOS);

    legend(Leyenda_Alg_1,Leyenda_Alg_2,Leyenda_Alg_3,Leyenda_Alg_4,Leyenda_Alg_5,Leyenda_A
lg_6,Leyenda_Alg_7)

end

```

A.5 Representación gráfica del RMSE en función de las probabilidades de identificación errónea

```

% REPRESENTACIÓN GRÁFICA DEL RMSE EN FUNCIÓN DE P[NLOS|LOS] Y P[LOS|NLOS]

function [] = graficaRMSE_P (RMSE_P, P, titulo)

    P_NLOS_LOS = [P(1) P(3) P(5) P(7) P(9)];
    P_LOS_NLOS = [P(2) P(4) P(6) P(8) P(10)];

    figure('name', titulo)

    for i=1:length(RMSE_P)
        stem3(P_NLOS_LOS(i), P_LOS_NLOS(i), RMSE_P(i), 'linewidth', 5)
        hold on
    end

    grid on
    grid minor
    xlim([0, max(P_NLOS_LOS)+0.05])
    ylim([0, max(P_LOS_NLOS)+0.05])
    zlim([0, max(RMSE_P)+0.1])
    xlabel('P[NLOS|LOS]')
    ylabel('P[LOS|NLOS]')
    zlabel('RMSE (metros)')
    title(titulo)

    % Creación de la leyenda:

    Texto_1 = 'Algoritmo 3: Con identificación de la condición de visión.
    P[NLOS|LOS]=';
    P1_NLOS_LOS = num2str(P(1));
    Texto = ' y P[LOS|NLOS]=';
    P1_LOS_NLOS = num2str(P(2));
    Leyenda_Alg_3 = strcat(Texto_1,P1_NLOS_LOS,Texto,P1_LOS_NLOS);

    Texto_2 = 'Algoritmo 4: Con identificación de la condición de visión.
    P[NLOS|LOS]=';
    P2_NLOS_LOS = num2str(P(3));
    P2_LOS_NLOS = num2str(P(4));
    Leyenda_Alg_4 = strcat(Texto_2,P2_NLOS_LOS,Texto,P2_LOS_NLOS);

    Texto_3 = 'Algoritmo 5: Con identificación de la condición de visión.
    P[NLOS|LOS]=';
    P3_NLOS_LOS = num2str(P(5));
    P3_LOS_NLOS = num2str(P(6));
    Leyenda_Alg_5 = strcat(Texto_3,P3_NLOS_LOS,Texto,P3_LOS_NLOS);

    Texto_4 = 'Algoritmo 6: Con identificación de la condición de visión.
    P[NLOS|LOS]=';
    P4_NLOS_LOS = num2str(P(7));
    P4_LOS_NLOS = num2str(P(8));
    Leyenda_Alg_6 = strcat(Texto_4,P4_NLOS_LOS,Texto,P4_LOS_NLOS);

    Texto_5 = 'Algoritmo 7: Con identificación de la condición de visión.
    P[NLOS|LOS]=';
    P5_NLOS_LOS = num2str(P(9));
    P5_LOS_NLOS = num2str(P(10));
    Leyenda_Alg_7 = strcat(Texto_5,P5_NLOS_LOS,Texto,P5_LOS_NLOS);

    legend(Leyenda_Alg_3,Leyenda_Alg_4,Leyenda_Alg_5,Leyenda_Alg_6,Leyenda_Alg_7)

end

```

A.6 Archivo principal: Ejecución de los algoritmos y gráficas

```

% ESTIMACIÓN POSICIÓN DEL MÓVIL Y ERROR DE LA ESTIMACIÓN

% CREACIÓN DEL GRID:

% Posición inicial y final de las cruces eje X e Y, y el espacio entre las cruces:
x_cruz_inicial = -5;
x_cruz_final = 20;

y_cruz_inicial = -5;
y_cruz_final = 20;

espacioEntreCruces = 0.125;

% Número de cruces del eje X del grid:
n_cruces_x = (x_cruz_final-x_cruz_inicial)/espacioEntreCruces +1; %201
% Número de cruces del eje Y del grid:
n_cruces_y = (y_cruz_final-y_cruz_inicial)/espacioEntreCruces +1; %201

% Definimos las cruces del grid:
x_cruz = zeros(1,n_cruces_x*n_cruces_y);
y_cruz = zeros(1,n_cruces_x*n_cruces_y);

k = 1;
for i=x_cruz_inicial:espacioEntreCruces:x_cruz_final
    for j=y_cruz_inicial:espacioEntreCruces:y_cruz_final
        x_cruz(k) = i;
        y_cruz(k) = j;
        k = k + 1;
    end
end

% Importamos las bases de datos:
posiciones_RealMovil = importdata('realPositions.txt')*0.5;
posiciones_Anclas = importdata('fixedNodesPosition.txt')*0.5;
medidas_Distancia = importdata('measuredDistance.txt');
condición_Vision = importdata('sightCondition.txt');

% Número de posiciones reales del Móvil:
n_movil = length(posiciones_RealMovil(:,1)); % 57 posiciones

% Número de Anclas:
n_anclas = length(posiciones_Anclas(:,1)); % 5 anclas

% Definimos las coordenadas (x,y) de las Anclas:
x = ones(n_anclas,1);
y = ones(n_anclas,1);
for k=1:n_anclas
    x(k) = posiciones_Anclas(k,1);
    y(k) = posiciones_Anclas(k,2);
end

% Definimos una matriz para cada algoritmo donde vamos a almacenar
% la posición estimada del móvil(coordenadas x e y) y el error de la estimación:
% Algoritmo 1: Sin identificación
sin_Identificacion = zeros(3,n_movil);
% Algoritmo 2: Con conocimiento perfecto de la condición de visión LOS/NLOS
conocimientoPerfecto_LOS_NLOS = zeros(3,n_movil);
% Algoritmo 3, 4, 5, 6 y 7: Con identificación de la condición de visión.
P[NLOS|LOS] y P[LOS|NLOS]
k = 500; % Se ejecutarán k veces para cada posición real del móvil para tener
resultados significativos ya que actúa una probabilidad
m = 0; % variable auxiliar para rellenar la matriz
identificacion_CondicionVisionP1 = zeros(3,n_movil*k);
identificacion_CondicionVisionP2 = zeros(3,n_movil*k);
identificacion_CondicionVisionP3 = zeros(3,n_movil*k);
identificacion_CondicionVisionP4 = zeros(3,n_movil*k);
identificacion_CondicionVisionP5 = zeros(3,n_movil*k);

% Definimos los pesos beta de LOS y NLOS:
beta_LOS=1; % Line Of Sight
beta_NLOS=0.001; % Non Line Of Sight

```

```

% Definimos las probabilidades de identificación errónea para los algoritmos 3, 4, 5,
6 y 7:
% P[NLOS|LOS] = Probabilidad de identificar la medida LOS como NLOS.
% P[LOS|NLOS] = Probabilidad de identificar la medida NLOS como LOS.
% Para el Algoritmo 3:
P1_NLOS_LOS = 0.01;
P1_LOS_NLOS = 0.01;
% Para el Algoritmo 4:
P2_NLOS_LOS = 0.02;
P2_LOS_NLOS = 0.02;
% Para el Algoritmo 5:
P3_NLOS_LOS = 0.05;
P3_LOS_NLOS = 0.05;
% Para el Algoritmo 6:
P4_NLOS_LOS = 0.1;
P4_LOS_NLOS = 0.1;
% Para el Algoritmo 7:
P5_NLOS_LOS = 0.2;
P5_LOS_NLOS = 0.2;

% APLICAMOS LOS ALGORITMOS:

% Para cada una de las posiciones reales del móvil aplicamos los algoritmos de
localización
for i=1:n_movil

    xm = posiciones_RealMovil(i,1); % Coordenada x del Móvil(i)
    ym = posiciones_RealMovil(i,2); % Coordenada y del Móvil(i)
    d = medidas_Distancia(i,:); % Medidas de distancia de cada Ancla al Móvil(i)
    LOS_NLOS = condicion_Vision(i,:); % Condición de visión de cada Ancla al Móvil(i)

    % Algoritmo 1: Sin identificación
    [x_estimado, y_estimado] = sinIdentificacion (n_cruces_x, n_cruces_y, x_cruz,
y_cruz, n_anclas, x, y, d, LOS_NLOS);
    % Almacenamos x e y estimadas y el error de la estimación:
    sin_Identificacion(1,i) = x_estimado;
    sin_Identificacion(2,i) = y_estimado;
    % Cálculo del error de la estimación:
    sin_Identificacion(3,i) = sqrt((xm-x_estimado)^2+(ym-y_estimado)^2);

    % Algoritmo 2: Con conocimiento perfecto de la condición de visión LOS/NLOS
    [x_estimado, y_estimado] = conocimientoPerfectoLOS_NLOS (n_cruces_x, n_cruces_y,
x_cruz, y_cruz, n_anclas, x, y, d, LOS_NLOS, beta_LOS, beta_NLOS);
    % Almacenamos x e y estimadas y el error de la estimación:
    conocimientoPerfecto_LOS_NLOS(1,i) = x_estimado;
    conocimientoPerfecto_LOS_NLOS(2,i) = y_estimado;
    % Cálculo del error de la estimación:
    conocimientoPerfecto_LOS_NLOS(3,i) = sqrt((xm-x_estimado)^2+(ym-
y_estimado)^2);

    for j=1:k
    % Algoritmo 3: Con identificación de la condición de visión. P1[NLOS|LOS] y
P1[LOS|NLOS]
    [x_estimado, y_estimado] = identificacionCondicionVision (n_cruces_x, n_cruces_y,
x_cruz, y_cruz, n_anclas, x, y, d, LOS_NLOS, beta_LOS, beta_NLOS, P1_NLOS_LOS,
P1_LOS_NLOS);
    % Almacenamos x e y estimadas y el error de la estimación:
    identificacion_CondicionVisionP1(1,j+m) = x_estimado;
    identificacion_CondicionVisionP1(2,j+m) = y_estimado;
    % Cálculo del error de la estimación:
    identificacion_CondicionVisionP1(3,j+m) = sqrt((xm-x_estimado)^2+(ym-
y_estimado)^2);

    % Algoritmo 4: Con identificación de la condición de visión. P2[NLOS|LOS] y
P2[LOS|NLOS]
    [x_estimado, y_estimado] = identificacionCondicionVision (n_cruces_x, n_cruces_y,
x_cruz, y_cruz, n_anclas, x, y, d, LOS_NLOS, beta_LOS, beta_NLOS, P2_NLOS_LOS,
P2_LOS_NLOS);
    % Almacenamos x e y estimadas y el error de la estimación:
    identificacion_CondicionVisionP2(1,j+m) = x_estimado;
    identificacion_CondicionVisionP2(2,j+m) = y_estimado;
    % Cálculo del error de la estimación:
    identificacion_CondicionVisionP2(3,j+m) = sqrt((xm-x_estimado)^2+(ym-
y_estimado)^2);
    end
end

```

```

% Algoritmo 5: Con identificación de la condición de visión. P3[NLOS|LOS] y
P3[LOS|NLOS]
[x_estimado, y_estimado] = identificacionCondicionVision (n_cruces_x, n_cruces_y,
x_cruz, y_cruz, n_anclas, x, y, d, LOS_NLOS, beta_LOS, beta_NLOS, P3_NLOS_LOS,
P3_LOS_NLOS);
% Almacenamos x e y estimadas y el error de la estimación:
identificacion_CondicionVisionP3(1,j+m) = x_estimado;
identificacion_CondicionVisionP3(2,j+m) = y_estimado;
% Cálculo del error de la estimación:
identificacion_CondicionVisionP3(3,j+m) = sqrt((xm-x_estimado)^2+(ym-
y_estimado)^2);

% Algoritmo 6: Con identificación de la condición de visión. P4[NLOS|LOS] y
P4[LOS|NLOS]
[x_estimado, y_estimado] = identificacionCondicionVision (n_cruces_x, n_cruces_y,
x_cruz, y_cruz, n_anclas, x, y, d, LOS_NLOS, beta_LOS, beta_NLOS, P4_NLOS_LOS,
P4_LOS_NLOS);
% Almacenamos x e y estimadas y el error de la estimación:
identificacion_CondicionVisionP4(1,j+m) = x_estimado;
identificacion_CondicionVisionP4(2,j+m) = y_estimado;
% Cálculo del error de la estimación:
identificacion_CondicionVisionP4(3,j+m) = sqrt((xm-x_estimado)^2+(ym-
y_estimado)^2);

% Algoritmo 7: Con identificación de la condición de visión. P5[NLOS|LOS] y
P5[LOS|NLOS]
[x_estimado, y_estimado] = identificacionCondicionVision (n_cruces_x, n_cruces_y,
x_cruz, y_cruz, n_anclas, x, y, d, LOS_NLOS, beta_LOS, beta_NLOS, P5_NLOS_LOS,
P5_LOS_NLOS);
% Almacenamos x e y estimadas y el error de la estimación:
identificacion_CondicionVisionP5(1,j+m) = x_estimado;
identificacion_CondicionVisionP5(2,j+m) = y_estimado;
% Cálculo del error de la estimación:
identificacion_CondicionVisionP5(3,j+m) = sqrt((xm-x_estimado)^2+(ym-
y_estimado)^2);

end
m = m + k;
end

% CÁLCULO DEL RMSE (Raíz del Error Cuadrático Medio) DE CADA ALGORITMO:

rmse_1=0;
rmse_2=0;
rmse_3=0;
rmse_4=0;
rmse_5=0;
rmse_6=0;
rmse_7=0;
for i=1:n_movil
    rmse_1 = rmse_1 + sin_Identificacion(3,i)^2;
    rmse_2 = rmse_2 + conocimientoPerfecto_LOS_NLOS(3,i)^2;
end
for i=1:n_movil*k
    rmse_3 = rmse_3 + identificacion_CondicionVisionP1(3,i)^2;
    rmse_4 = rmse_4 + identificacion_CondicionVisionP2(3,i)^2;
    rmse_5 = rmse_5 + identificacion_CondicionVisionP3(3,i)^2;
    rmse_6 = rmse_6 + identificacion_CondicionVisionP4(3,i)^2;
    rmse_7 = rmse_7 + identificacion_CondicionVisionP5(3,i)^2;
end

% RMSE Algoritmo 1: Sin identificación
RMSE_1 = sqrt((1/n_movil)*rmse_1);
% RMSE Algoritmo 2: Con conocimiento perfecto de la condición de visión LOS/NLOS
RMSE_2 = sqrt((1/n_movil)*rmse_2);
% RMSE Algoritmo 3: Con identificación de la condición de visión. P1[NLOS|LOS] y
P1[LOS|NLOS]
RMSE_3 = sqrt((1/(n_movil*k))*rmse_3);
% RMSE Algoritmo 4: Con identificación de la condición de visión. P2[NLOS|LOS] y
P2[LOS|NLOS]
RMSE_4 = sqrt((1/(n_movil*k))*rmse_4);
% RMSE Algoritmo 5: Con identificación de la condición de visión. P3[NLOS|LOS] y
P3[LOS|NLOS]
RMSE_5 = sqrt((1/(n_movil*k))*rmse_5);

```

```

% RMSE Algoritmo 6: Con identificación de la condición de visión. P4[NLOS|LOS] y
P4[LOS|NLOS]
RMSE_6 = sqrt((1/(n_movil*k))*rmse_6);
% RMSE Algoritmo 7: Con identificación de la condición de visión. P5[NLOS|LOS] y
P5[LOS|NLOS]
RMSE_7 = sqrt((1/(n_movil*k))*rmse_7);

RMSE = [RMSE_1 RMSE_2 RMSE_3 RMSE_4 RMSE_5 RMSE_6 RMSE_7];
P = [P1_NLOS_LOS P1_LOS_NLOS P2_NLOS_LOS P2_LOS_NLOS P3_NLOS_LOS P3_LOS_NLOS
P4_NLOS_LOS P4_LOS_NLOS P5_NLOS_LOS P5_LOS_NLOS];

% REPRESENTACIÓN GRÁFICA DEL RMSE DE CADA ALGORITMO:

titulo_RMSE = 'REPRESENTACIÓN GRÁFICA DEL RMSE DE CADA ALGORITMO';
graficaRMSE (RMSE, P, titulo_RMSE);

% REPRESENTACIÓN GRÁFICA DEL RMSE EN FUNCIÓN DE LAS PROBABILIDADES DE IDENTIFICACIÓN
ERRÓNEA:

RMSE_P = [RMSE_3 RMSE_4 RMSE_5 RMSE_6 RMSE_7];
titulo_RMSE_P = 'RMSE EN FUNCIÓN DE LAS PROBABILIDADES DE IDENTIFICACIÓN ERRÓNEA.
P[NLOS|LOS] y P[LOS|NLOS]';
graficaRMSE_P (RMSE_P, P, titulo_RMSE_P);

```

A.7 Casos Anclas LOS/NLOS

```

% CASOS ANCLAS LOS/NLOS:

% CREACIÓN DEL GRID:

% Posición inicial y final de las cruces eje X e Y, y el espacio entre las cruces:
x_cruz_inicial = -5;
x_cruz_final = 20;

y_cruz_inicial = -5;
y_cruz_final = 20;

espacioEntreCruces = 0.125;

% Número de cruces del eje X del grid:
n_cruces_x = (x_cruz_final-x_cruz_inicial)/espacioEntreCruces +1; %201
% Número de cruces del eje Y del grid:
n_cruces_y = (y_cruz_final-y_cruz_inicial)/espacioEntreCruces +1; %201

% Definimos las cruces del grid:
x_cruz = zeros(1,n_cruces_x*n_cruces_y);
y_cruz = zeros(1,n_cruces_x*n_cruces_y);

k = 1;
for i=x_cruz_inicial:espacioEntreCruces:x_cruz_final
    for j=y_cruz_inicial:espacioEntreCruces:y_cruz_final
        x_cruz(k) = i;
        y_cruz(k) = j;
        k = k + 1;
    end
end

% CASO 1: 4 ANCLAS LOS y 1 ANCLA NLOS
% CASO 2: 3 ANCLAS LOS y 2 ANCLAS NLOS
% CASO 3: 2 ANCLAS LOS y 3 ANCLAS NLOS

% Definimos los siguientes parámetros comunes para todos los casos:

% Importamos la base de datos de las posiciones de las anclas:
posiciones_Anclas = importdata('fixedNodesPosition.txt')*0.5;

```

```

% Número de Anclas:
n_anclas = length(posiciones_Anclas(:,1)); % 5 anclas

% Definimos las coordenadas (x,y) de las Anclas:
x = ones(n_anclas,1);
y = ones(n_anclas,1);
for k=1:n_anclas
    x(k) = posiciones_Anclas(k,1);
    y(k) = posiciones_Anclas(k,2);
end

% Definimos los pesos beta de LOS y NLOS:
beta_LOS=1; % Line Of Sight
beta_NLOS=0.001; % Non Line Of Sight

% Definimos las probabilidades de identificación errónea:
% P[NLOS|LOS] = Probabilidad de identificar la medida LOS como NLOS.
% P[LOS|NLOS] = Probabilidad de identificar la medida NLOS como LOS.
% Para el Algoritmo 3:
P1_NLOS_LOS = 0.01;
P1_LOS_NLOS = 0.01;
% Para el Algoritmo 4:
P2_NLOS_LOS = 0.02;
P2_LOS_NLOS = 0.02;
% Para el Algoritmo 5:
P3_NLOS_LOS = 0.05;
P3_LOS_NLOS = 0.05;
% Para el Algoritmo 6:
P4_NLOS_LOS = 0.1;
P4_LOS_NLOS = 0.1;
% Para el Algoritmo 7:
P5_NLOS_LOS = 0.2;
P5_LOS_NLOS = 0.2;

% CASO 1: 4 ANCLAS LOS y 1 ANCLA NLOS

% Importamos las bases de datos para este caso:
posiciones_RealMovil = importdata('4_1_posicionRealMovil.txt')*0.5;
medidas_Distancia = importdata('4_1_medidasDistancia.txt');
condición_Vision = importdata('4LOS_1NLOS.txt');

% Número de posiciones reales del Móvil para este caso:
n_movil = length(posiciones_RealMovil(:,1)); % 1 posición

% Definimos una matriz para cada algoritmo donde vamos a almacenar
% la posición estimada del móvil(coordenadas x e y) y el error de la estimación para
este caso:
% Algoritmo 1: Sin identificación
sin_Identificacion = zeros(3,n_movil);
% Algoritmo 2: Con conocimiento perfecto de la condición de visión LOS/NLOS
conocimientoPerfecto_LOS_NLOS = zeros(3,n_movil);
% Algoritmo 3, 4, 5, 6 y 7: Con identificación de la condición de visión.
P[NLOS|LOS] y P[LOS|NLOS]
k = 1000; % Se ejecutarán k veces para cada posición real del móvil para tener
resultados significativos ya que actúa una probabilidad
m = 0; % variable auxiliar para rellenar la matriz
identificacion_CondicionVisionP1 = zeros(3,n_movil*k);
identificacion_CondicionVisionP2 = zeros(3,n_movil*k);
identificacion_CondicionVisionP3 = zeros(3,n_movil*k);
identificacion_CondicionVisionP4 = zeros(3,n_movil*k);
identificacion_CondicionVisionP5 = zeros(3,n_movil*k);

% APLICAMOS LOS ALGORITMOS:

% Para cada una de las posiciones reales del móvil aplicamos los algoritmos de
localización
for i=1:n_movil

    xm = posiciones_RealMovil(i,1); % Coordenada x del Móvil(i)
    ym = posiciones_RealMovil(i,2); % Coordenada y del Móvil(i)
    d = medidas_Distancia(i,:); % Medidas de distancia de cada Ancla al Móvil(i)
    LOS_NLOS = condición_Vision(i,:); % Condición de visión de cada Ancla al Móvil(i)

```

```

% Algoritmo 1: Sin identificación
[x_estimado, y_estimado] = sinIdentificacion (n_cruces_x, n_cruces_y, x_cruz,
y_cruz, n_anclas, x, y, d, LOS_NLOS);
% Almacenamos x e y estimadas y el error de la estimación:
sin_Identificacion(1,i) = x_estimado;
sin_Identificacion(2,i) = y_estimado;
% Cálculo del error de la estimación:
sin_Identificacion(3,i) = sqrt((xm-x_estimado)^2+(ym-y_estimado)^2);

% Algoritmo 2: Con conocimiento perfecto de la condición de visión LOS/NLOS
[x_estimado, y_estimado] = conocimientoPerfectoLOS_NLOS (n_cruces_x, n_cruces_y,
x_cruz, y_cruz, n_anclas, x, y, d, LOS_NLOS, beta_LOS, beta_NLOS);
% Almacenamos x e y estimadas y el error de la estimación:
conocimientoPerfecto_LOS_NLOS(1,i) = x_estimado;
conocimientoPerfecto_LOS_NLOS(2,i) = y_estimado;
% Cálculo del error de la estimación:
conocimientoPerfecto_LOS_NLOS(3,i) = sqrt((xm-x_estimado)^2+(ym-
y_estimado)^2);

for j=1:k
% Algoritmo 3: Con identificación de la condición de visión. P1[NLOS|LOS] y
P1[LOS|NLOS]
[x_estimado, y_estimado] = identificacionCondicionVision (n_cruces_x, n_cruces_y,
x_cruz, y_cruz, n_anclas, x, y, d, LOS_NLOS, beta_LOS, beta_NLOS, P1_NLOS_LOS,
P1_LOS_NLOS);
% Almacenamos x e y estimadas y el error de la estimación:
identificacion_CondicionVisionP1(1,j+m) = x_estimado;
identificacion_CondicionVisionP1(2,j+m) = y_estimado;
% Cálculo del error de la estimación:
identificacion_CondicionVisionP1(3,j+m) = sqrt((xm-x_estimado)^2+(ym-
y_estimado)^2);

% Algoritmo 4: Con identificación de la condición de visión. P2[NLOS|LOS] y
P2[LOS|NLOS]
[x_estimado, y_estimado] = identificacionCondicionVision (n_cruces_x, n_cruces_y,
x_cruz, y_cruz, n_anclas, x, y, d, LOS_NLOS, beta_LOS, beta_NLOS, P2_NLOS_LOS,
P2_LOS_NLOS);
% Almacenamos x e y estimadas y el error de la estimación:
identificacion_CondicionVisionP2(1,j+m) = x_estimado;
identificacion_CondicionVisionP2(2,j+m) = y_estimado;
% Cálculo del error de la estimación:
identificacion_CondicionVisionP2(3,j+m) = sqrt((xm-x_estimado)^2+(ym-
y_estimado)^2);

% Algoritmo 5: Con identificación de la condición de visión. P3[NLOS|LOS] y
P3[LOS|NLOS]
[x_estimado, y_estimado] = identificacionCondicionVision (n_cruces_x, n_cruces_y,
x_cruz, y_cruz, n_anclas, x, y, d, LOS_NLOS, beta_LOS, beta_NLOS, P3_NLOS_LOS,
P3_LOS_NLOS);
% Almacenamos x e y estimadas y el error de la estimación:
identificacion_CondicionVisionP3(1,j+m) = x_estimado;
identificacion_CondicionVisionP3(2,j+m) = y_estimado;
% Cálculo del error de la estimación:
identificacion_CondicionVisionP3(3,j+m) = sqrt((xm-x_estimado)^2+(ym-
y_estimado)^2);

% Algoritmo 6: Con identificación de la condición de visión. P4[NLOS|LOS] y
P4[LOS|NLOS]
[x_estimado, y_estimado] = identificacionCondicionVision (n_cruces_x, n_cruces_y,
x_cruz, y_cruz, n_anclas, x, y, d, LOS_NLOS, beta_LOS, beta_NLOS, P4_NLOS_LOS,
P4_LOS_NLOS);
% Almacenamos x e y estimadas y el error de la estimación:
identificacion_CondicionVisionP4(1,j+m) = x_estimado;
identificacion_CondicionVisionP4(2,j+m) = y_estimado;
% Cálculo del error de la estimación:
identificacion_CondicionVisionP4(3,j+m) = sqrt((xm-x_estimado)^2+(ym-
y_estimado)^2);

% Algoritmo 7: Con identificación de la condición de visión. P5[NLOS|LOS] y
P5[LOS|NLOS]
[x_estimado, y_estimado] = identificacionCondicionVision (n_cruces_x, n_cruces_y,
x_cruz, y_cruz, n_anclas, x, y, d, LOS_NLOS, beta_LOS, beta_NLOS, P5_NLOS_LOS,
P5_LOS_NLOS);
% Almacenamos x e y estimadas y el error de la estimación:

```

```

        identificacion_CondicionVisionP5(1,j+m) = x_estimado;
        identificacion_CondicionVisionP5(2,j+m) = y_estimado;
        % Cálculo del error de la estimación:
        identificacion_CondicionVisionP5(3,j+m) = sqrt((xm-x_estimado)^2+(ym-
        y_estimado)^2);
    end
    m = m + k;

end

% Cálculo del RMSE CASO 1: 4 ANCLAS LOS y 1 ANCLAS NLOS:

rmse_1=0;
rmse_2=0;
rmse_3=0;
rmse_4=0;
rmse_5=0;
rmse_6=0;
rmse_7=0;
for i=1:n_movil
    rmse_1 = rmse_1 + sin_Identificacion(3,i)^2;
    rmse_2 = rmse_2 + conocimientoPerfecto_LOS_NLOS(3,i)^2;
end
for i=1:n_movil*k
    rmse_3 = rmse_3 + identificacion_CondicionVisionP1(3,i)^2;
    rmse_4 = rmse_4 + identificacion_CondicionVisionP2(3,i)^2;
    rmse_5 = rmse_5 + identificacion_CondicionVisionP3(3,i)^2;
    rmse_6 = rmse_6 + identificacion_CondicionVisionP4(3,i)^2;
    rmse_7 = rmse_7 + identificacion_CondicionVisionP5(3,i)^2;
end

% RMSE Algoritmo 1: Sin identificación
RMSE_1 = sqrt((1/n_movil)*rmse_1);
% RMSE Algoritmo 2: Con conocimiento perfecto de la condición de visión LOS/NLOS
RMSE_2 = sqrt((1/n_movil)*rmse_2);
% RMSE Algoritmo 3: Con identificación de la condición de visión. P1[NLOS|LOS] y
P1[LOS|NLOS]
RMSE_3 = sqrt((1/(n_movil*k))*rmse_3);
% RMSE Algoritmo 4: Con identificación de la condición de visión. P2[NLOS|LOS] y
P2[LOS|NLOS]
RMSE_4 = sqrt((1/(n_movil*k))*rmse_4);
% RMSE Algoritmo 5: Con identificación de la condición de visión. P3[NLOS|LOS] y
P3[LOS|NLOS]
RMSE_5 = sqrt((1/(n_movil*k))*rmse_5);
% RMSE Algoritmo 6: Con identificación de la condición de visión. P4[NLOS|LOS] y
P4[LOS|NLOS]
RMSE_6 = sqrt((1/(n_movil*k))*rmse_6);
% RMSE Algoritmo 7: Con identificación de la condición de visión. P5[NLOS|LOS] y
P5[LOS|NLOS]
RMSE_7 = sqrt((1/(n_movil*k))*rmse_7);

RMSE_4LOS_1NLOS = [RMSE_1 RMSE_2 RMSE_3 RMSE_4 RMSE_5 RMSE_6 RMSE_7];

% CASO 2: 3 ANCLAS LOS y 2 ANCLAS NLOS

% Importamos las bases de datos para este caso:
posiciones_RealMovil = importdata('3_2_posicionRealMovil.txt')*0.5;
medidas_Distancia = importdata('3_2_medidasDistancia.txt');
condición_Vision = importdata('3LOS_2NLOS.txt');

% Número de posiciones reales del Móvil para este caso:
n_movil = length(posiciones_RealMovil(:,1)); % 7 posiciones

% Definimos una matriz para cada algoritmo donde vamos a almacenar
% la posición estimada del móvil(coordenadas x e y) y el error de la estimación para
este caso:
% Algoritmo 1: Sin identificación
sin_Identificacion = zeros(3,n_movil);
% Algoritmo 2: Con conocimiento perfecto de la condición de visión LOS/NLOS
conocimientoPerfecto_LOS_NLOS = zeros(3,n_movil);
% Algoritmo 3, 4, 5, 6 y 7: Con identificación de la condición de visión.
P[NLOS|LOS] y P[LOS|NLOS]

```

```

k = 1000; % Se ejecutarán k veces para cada posición real del móvil para tener
resultados significativos ya que actúa una probabilidad
m = 0; % variable auxiliar para rellenar la matriz
identificacion_CondicionVisionP1 = zeros(3,n_movil*k);
identificacion_CondicionVisionP2 = zeros(3,n_movil*k);
identificacion_CondicionVisionP3 = zeros(3,n_movil*k);
identificacion_CondicionVisionP4 = zeros(3,n_movil*k);
identificacion_CondicionVisionP5 = zeros(3,n_movil*k);

% APLICAMOS LOS ALGORITMOS:

% Para cada una de las posiciones reales del móvil aplicamos los algoritmos de
localización
for i=1:n_movil

    xm = posiciones_RealMovil(i,1); % Coordenada x del Móvil(i)
    ym = posiciones_RealMovil(i,2); % Coordenada y del Móvil(i)
    d = medidas_Distancia(i,:); % Medidas de distancia de cada Ancla al Móvil(i)
    LOS_NLOS = condicion_Vision(i,:); % Condición de visión de cada Ancla al Móvil(i)

    % Algoritmo 1: Sin identificación
    [x_estimado, y_estimado] = sinIdentificacion (n_cruces_x, n_cruces_y, x_cruz,
y_cruz, n_anclas, x, y, d, LOS_NLOS);
    % Almacenamos x e y estimadas y el error de la estimación:
    sin_Identificacion(1,i) = x_estimado;
    sin_Identificacion(2,i) = y_estimado;
    % Cálculo del error de la estimación:
    sin_Identificacion(3,i) = sqrt((xm-x_estimado)^2+(ym-y_estimado)^2);

    % Algoritmo 2: Con conocimiento perfecto de la condición de visión LOS/NLOS
    [x_estimado, y_estimado] = conocimientoPerfectoLOS_NLOS (n_cruces_x, n_cruces_y,
x_cruz, y_cruz, n_anclas, x, y, d, LOS_NLOS, beta_LOS, beta_NLOS);
    % Almacenamos x e y estimadas y el error de la estimación:
    conocimientoPerfecto_LOS_NLOS(1,i) = x_estimado;
    conocimientoPerfecto_LOS_NLOS(2,i) = y_estimado;
    % Cálculo del error de la estimación:
    conocimientoPerfecto_LOS_NLOS(3,i) = sqrt((xm-x_estimado)^2+(ym-
y_estimado)^2);

    for j=1:k
    % Algoritmo 3: Con identificación de la condición de visión. P1[NLOS|LOS] y
P1[LOS|NLOS]
    [x_estimado, y_estimado] = identificacionCondicionVision (n_cruces_x, n_cruces_y,
x_cruz, y_cruz, n_anclas, x, y, d, LOS_NLOS, beta_LOS, beta_NLOS, P1_NLOS_LOS,
P1_LOS_NLOS);
    % Almacenamos x e y estimadas y el error de la estimación:
    identificacion_CondicionVisionP1(1,j+m) = x_estimado;
    identificacion_CondicionVisionP1(2,j+m) = y_estimado;
    % Cálculo del error de la estimación:
    identificacion_CondicionVisionP1(3,j+m) = sqrt((xm-x_estimado)^2+(ym-
y_estimado)^2);

    % Algoritmo 4: Con identificación de la condición de visión. P2[NLOS|LOS] y
P2[LOS|NLOS]
    [x_estimado, y_estimado] = identificacionCondicionVision (n_cruces_x, n_cruces_y,
x_cruz, y_cruz, n_anclas, x, y, d, LOS_NLOS, beta_LOS, beta_NLOS, P2_NLOS_LOS,
P2_LOS_NLOS);
    % Almacenamos x e y estimadas y el error de la estimación:
    identificacion_CondicionVisionP2(1,j+m) = x_estimado;
    identificacion_CondicionVisionP2(2,j+m) = y_estimado;
    % Cálculo del error de la estimación:
    identificacion_CondicionVisionP2(3,j+m) = sqrt((xm-x_estimado)^2+(ym-
y_estimado)^2);

    % Algoritmo 5: Con identificación de la condición de visión. P3[NLOS|LOS] y
P3[LOS|NLOS]
    [x_estimado, y_estimado] = identificacionCondicionVision (n_cruces_x, n_cruces_y,
x_cruz, y_cruz, n_anclas, x, y, d, LOS_NLOS, beta_LOS, beta_NLOS, P3_NLOS_LOS,
P3_LOS_NLOS);
    % Almacenamos x e y estimadas y el error de la estimación:
    identificacion_CondicionVisionP3(1,j+m) = x_estimado;
    identificacion_CondicionVisionP3(2,j+m) = y_estimado;
    % Cálculo del error de la estimación:
    identificacion_CondicionVisionP3(3,j+m) = sqrt((xm-x_estimado)^2+(ym-

```

```

        y_estimado)^2);

% Algoritmo 6: Con identificación de la condición de visión. P4[NLOS|LOS] y
P4[LOS|NLOS]
[x_estimado, y_estimado] = identificacionCondicionVision (n_cruces_x, n_cruces_y,
x_cruz, y_cruz, n_anclas, x, y, d, LOS_NLOS, beta_LOS, beta_NLOS, P4_NLOS_LOS,
P4_LOS_NLOS);
    % Almacenamos x e y estimadas y el error de la estimación:
        identificacion_CondicionVisionP4(1,j+m) = x_estimado;
        identificacion_CondicionVisionP4(2,j+m) = y_estimado;
    % Cálculo del error de la estimación:
        identificacion_CondicionVisionP4(3,j+m) = sqrt((xm-x_estimado)^2+(ym-
y_estimado)^2);

% Algoritmo 7: Con identificación de la condición de visión. P5[NLOS|LOS] y
P5[LOS|NLOS]
[x_estimado, y_estimado] = identificacionCondicionVision (n_cruces_x, n_cruces_y,
x_cruz, y_cruz, n_anclas, x, y, d, LOS_NLOS, beta_LOS, beta_NLOS, P5_NLOS_LOS,
P5_LOS_NLOS);
    % Almacenamos x e y estimadas y el error de la estimación:
        identificacion_CondicionVisionP5(1,j+m) = x_estimado;
        identificacion_CondicionVisionP5(2,j+m) = y_estimado;
    % Cálculo del error de la estimación:
        identificacion_CondicionVisionP5(3,j+m) = sqrt((xm-x_estimado)^2+(ym-
y_estimado)^2);

end
m = m + k;

end

% Cálculo del RMSE CASO 2: 3 ANCLAS LOS y 2 ANCLAS NLOS:

rmse_1=0;
rmse_2=0;
rmse_3=0;
rmse_4=0;
rmse_5=0;
rmse_6=0;
rmse_7=0;
for i=1:n_movil
    rmse_1 = rmse_1 + sin_Identificacion(3,i)^2;
    rmse_2 = rmse_2 + conocimientoPerfecto_LOS_NLOS(3,i)^2;
end
for i=1:n_movil*k
    rmse_3 = rmse_3 + identificacion_CondicionVisionP1(3,i)^2;
    rmse_4 = rmse_4 + identificacion_CondicionVisionP2(3,i)^2;
    rmse_5 = rmse_5 + identificacion_CondicionVisionP3(3,i)^2;
    rmse_6 = rmse_6 + identificacion_CondicionVisionP4(3,i)^2;
    rmse_7 = rmse_7 + identificacion_CondicionVisionP5(3,i)^2;
end

% RMSE Algoritmo 1: Sin identificación
RMSE_1 = sqrt((1/n_movil)*rmse_1);
% RMSE Algoritmo 2: Con conocimiento perfecto de la condición de visión LOS/NLOS
RMSE_2 = sqrt((1/n_movil)*rmse_2);
% RMSE Algoritmo 3: Con identificación de la condición de visión. P1[NLOS|LOS] y
P1[LOS|NLOS]
RMSE_3 = sqrt((1/(n_movil*k))*rmse_3);
% RMSE Algoritmo 4: Con identificación de la condición de visión. P2[NLOS|LOS] y
P2[LOS|NLOS]
RMSE_4 = sqrt((1/(n_movil*k))*rmse_4);
% RMSE Algoritmo 5: Con identificación de la condición de visión. P3[NLOS|LOS] y
P3[LOS|NLOS]
RMSE_5 = sqrt((1/(n_movil*k))*rmse_5);
% RMSE Algoritmo 6: Con identificación de la condición de visión. P4[NLOS|LOS] y
P4[LOS|NLOS]
RMSE_6 = sqrt((1/(n_movil*k))*rmse_6);
% RMSE Algoritmo 7: Con identificación de la condición de visión. P5[NLOS|LOS] y
P5[LOS|NLOS]
RMSE_7 = sqrt((1/(n_movil*k))*rmse_7);

RMSE_3LOS_2NLOS = [RMSE_1 RMSE_2 RMSE_3 RMSE_4 RMSE_5 RMSE_6 RMSE_7];

```

```

% CASO 3: 2 ANCLAS LOS y 3 ANCLAS NLOS

% Importamos las bases de datos para este caso:
posiciones_RealMovil = importdata('2_3_posicionRealMovil.txt')*0.5;
medidas_Distancia = importdata('2_3_medidasDistancia.txt');
condicion_Vision = importdata('2LOS_3NLOS.txt');

% Número de posiciones reales del Móvil para este caso:
n_movil = length(posiciones_RealMovil(:,1)); % 28 posiciones

% Definimos una matriz para cada algoritmo donde vamos a almacenar
% la posición estimada del móvil(coordenadas x e y) y el error de la estimación para
este caso:
% Algoritmo 1: Sin identificación
sin_Identificacion = zeros(3,n_movil);
% Algoritmo 2: Con conocimiento perfecto de la condición de visión LOS/NLOS
conocimientoPerfecto_LOS_NLOS = zeros(3,n_movil);
% Algoritmo 3, 4, 5, 6 y 7: Con identificación de la condición de visión.
P[NLOS|LOS] y P[LOS|NLOS]
k = 1000; % Se ejecutarán k veces para cada posición real del móvil para tener
resultados significativos ya que actúa una probabilidad
m = 0; % variable auxiliar para rellenar la matriz
identificacion_CondicionVisionP1 = zeros(3,n_movil*k);
identificacion_CondicionVisionP2 = zeros(3,n_movil*k);
identificacion_CondicionVisionP3 = zeros(3,n_movil*k);
identificacion_CondicionVisionP4 = zeros(3,n_movil*k);
identificacion_CondicionVisionP5 = zeros(3,n_movil*k);

% APLICAMOS LOS ALGORITMOS:

% Para cada una de las posiciones reales del móvil aplicamos los algoritmos de
localización
for i=1:n_movil

    xm = posiciones_RealMovil(i,1); % Coordenada x del Móvil(i)
    ym = posiciones_RealMovil(i,2); % Coordenada y del Móvil(i)
    d = medidas_Distancia(i,:); % Medidas de distancia de cada Ancla al Móvil(i)
    LOS_NLOS = condicion_Vision(i,:); % Condición de visión de cada Ancla al Móvil(i)

    % Algoritmo 1: Sin identificación
    [x_estimado, y_estimado] = sinIdentificacion (n_cruces_x, n_cruces_y, x_cruz,
y_cruz, n_anclas, x, y, d, LOS_NLOS);
    % Almacenamos x e y estimadas y el error de la estimación:
    sin_Identificacion(1,i) = x_estimado;
    sin_Identificacion(2,i) = y_estimado;
    % Cálculo del error de la estimación:
    sin_Identificacion(3,i) = sqrt((xm-x_estimado)^2+(ym-y_estimado)^2);

    % Algoritmo 2: Con conocimiento perfecto de la condición de visión LOS/NLOS
    [x_estimado, y_estimado] = conocimientoPerfectoLOS_NLOS (n_cruces_x, n_cruces_y,
x_cruz, y_cruz, n_anclas, x, y, d, LOS_NLOS, beta_LOS, beta_NLOS);
    % Almacenamos x e y estimadas y el error de la estimación:
    conocimientoPerfecto_LOS_NLOS(1,i) = x_estimado;
    conocimientoPerfecto_LOS_NLOS(2,i) = y_estimado;
    % Cálculo del error de la estimación:
    conocimientoPerfecto_LOS_NLOS(3,i) = sqrt((xm-x_estimado)^2+(ym-
y_estimado)^2);

    for j=1:k
        % Algoritmo 3: Con identificación de la condición de visión. P1[NLOS|LOS] y
P1[LOS|NLOS]
        [x_estimado, y_estimado] = identificacionCondicionVision (n_cruces_x, n_cruces_y,
x_cruz, y_cruz, n_anclas, x, y, d, LOS_NLOS, beta_LOS, beta_NLOS, P1_NLOS_LOS,
P1_LOS_NLOS);
        % Almacenamos x e y estimadas y el error de la estimación:
        identificacion_CondicionVisionP1(1,j+m) = x_estimado;
        identificacion_CondicionVisionP1(2,j+m) = y_estimado;
        % Cálculo del error de la estimación:
        identificacion_CondicionVisionP1(3,j+m) = sqrt((xm-x_estimado)^2+(ym-
y_estimado)^2);

        % Algoritmo 4: Con identificación de la condición de visión. P2[NLOS|LOS] y
P2[LOS|NLOS]
        [x_estimado, y_estimado] = identificacionCondicionVision (n_cruces_x, n_cruces_y,

```

```

x_cruz, y_cruz, n_anclas, x, y, d, LOS_NLOS, beta_LOS, beta_NLOS, P2_NLOS_LOS,
P2_LOS_NLOS);
    % Almacenamos x e y estimadas y el error de la estimación:
    identificacion_CondicionVisionP2(1,j+m) = x_estimado;
    identificacion_CondicionVisionP2(2,j+m) = y_estimado;
    % Cálculo del error de la estimación:
    identificacion_CondicionVisionP2(3,j+m) = sqrt((xm-x_estimado)^2+(ym-
y_estimado)^2);

% Algoritmo 5: Con identificación de la condición de visión. P3[NLOS|LOS] y
P3[LOS|NLOS]
[x_estimado, y_estimado] = identificacionCondicionVision (n_cruces_x, n_cruces_y,
x_cruz, y_cruz, n_anclas, x, y, d, LOS_NLOS, beta_LOS, beta_NLOS, P3_NLOS_LOS,
P3_LOS_NLOS);
    % Almacenamos x e y estimadas y el error de la estimación:
    identificacion_CondicionVisionP3(1,j+m) = x_estimado;
    identificacion_CondicionVisionP3(2,j+m) = y_estimado;
    % Cálculo del error de la estimación:
    identificacion_CondicionVisionP3(3,j+m) = sqrt((xm-x_estimado)^2+(ym-
y_estimado)^2);

% Algoritmo 6: Con identificación de la condición de visión. P4[NLOS|LOS] y
P4[LOS|NLOS]
[x_estimado, y_estimado] = identificacionCondicionVision (n_cruces_x, n_cruces_y,
x_cruz, y_cruz, n_anclas, x, y, d, LOS_NLOS, beta_LOS, beta_NLOS, P4_NLOS_LOS,
P4_LOS_NLOS);
    % Almacenamos x e y estimadas y el error de la estimación:
    identificacion_CondicionVisionP4(1,j+m) = x_estimado;
    identificacion_CondicionVisionP4(2,j+m) = y_estimado;
    % Cálculo del error de la estimación:
    identificacion_CondicionVisionP4(3,j+m) = sqrt((xm-x_estimado)^2+(ym-
y_estimado)^2);

% Algoritmo 7: Con identificación de la condición de visión. P5[NLOS|LOS] y
P5[LOS|NLOS]
[x_estimado, y_estimado] = identificacionCondicionVision (n_cruces_x, n_cruces_y,
x_cruz, y_cruz, n_anclas, x, y, d, LOS_NLOS, beta_LOS, beta_NLOS, P5_NLOS_LOS,
P5_LOS_NLOS);
    % Almacenamos x e y estimadas y el error de la estimación:
    identificacion_CondicionVisionP5(1,j+m) = x_estimado;
    identificacion_CondicionVisionP5(2,j+m) = y_estimado;
    % Cálculo del error de la estimación:
    identificacion_CondicionVisionP5(3,j+m) = sqrt((xm-x_estimado)^2+(ym-
y_estimado)^2);

end
m = m + k;

end

% Cálculo del RMSE CASO 3: 2 ANCLAS LOS y 3 ANCLAS NLOS:

rmse_1=0;
rmse_2=0;
rmse_3=0;
rmse_4=0;
rmse_5=0;
rmse_6=0;
rmse_7=0;
for i=1:n_movil
    rmse_1 = rmse_1 + sin_Identificacion(3,i)^2;
    rmse_2 = rmse_2 + conocimientoPerfecto_LOS_NLOS(3,i)^2;
end
for i=1:n_movil*k
    rmse_3 = rmse_3 + identificacion_CondicionVisionP1(3,i)^2;
    rmse_4 = rmse_4 + identificacion_CondicionVisionP2(3,i)^2;
    rmse_5 = rmse_5 + identificacion_CondicionVisionP3(3,i)^2;
    rmse_6 = rmse_6 + identificacion_CondicionVisionP4(3,i)^2;
    rmse_7 = rmse_7 + identificacion_CondicionVisionP5(3,i)^2;
end

% RMSE Algoritmo 1: Sin identificación
RMSE_1 = sqrt((1/n_movil)*rmse_1);
% RMSE Algoritmo 2: Con conocimiento perfecto de la condición de visión LOS/NLOS
RMSE_2 = sqrt((1/n_movil)*rmse_2);

```

```

% RMSE Algoritmo 3: Con identificación de la condición de visión. P1[NLOS|LOS] y
P1[LOS|NLOS]
RMSE_3 = sqrt((1/(n_movil*k))*rmse_3);
% RMSE Algoritmo 4: Con identificación de la condición de visión. P2[NLOS|LOS] y
P2[LOS|NLOS]
RMSE_4 = sqrt((1/(n_movil*k))*rmse_4);
% RMSE Algoritmo 5: Con identificación de la condición de visión. P3[NLOS|LOS] y
P3[LOS|NLOS]
RMSE_5 = sqrt((1/(n_movil*k))*rmse_5);
% RMSE Algoritmo 6: Con identificación de la condición de visión. P4[NLOS|LOS] y
P4[LOS|NLOS]
RMSE_6 = sqrt((1/(n_movil*k))*rmse_6);
% RMSE Algoritmo 7: Con identificación de la condición de visión. P5[NLOS|LOS] y
P5[LOS|NLOS]
RMSE_7 = sqrt((1/(n_movil*k))*rmse_7);

RMSE_2LOS_3NLOS = [RMSE_1 RMSE_2 RMSE_3 RMSE_4 RMSE_5 RMSE_6 RMSE_7];

% REPRESENTACIÓN GRÁFICA DEL RMSE PARA CADA CASO:

P = [P1_NLOS_LOS P1_LOS_NLOS P2_NLOS_LOS P2_LOS_NLOS P3_NLOS_LOS P3_LOS_NLOS
P4_NLOS_LOS P4_LOS_NLOS P5_NLOS_LOS P5_LOS_NLOS];

% Representación gráfica del RMSE CASO 1: 4 ANCLAS LOS y 1 ANCLAS NLOS:
titulo_4LOS_1NLOS = 'CASO 1: 4 ANCLAS LOS y 1 ANCLAS NLOS';
graficaRMSE (RMSE_4LOS_1NLOS, P, titulo_4LOS_1NLOS);

% Representación gráfica del RMSE CASO 2: 3 ANCLAS LOS y 2 ANCLAS NLOS:
titulo_3LOS_2NLOS = 'CASO 2: 3 ANCLAS LOS y 2 ANCLAS NLOS';
graficaRMSE (RMSE_3LOS_2NLOS, P, titulo_3LOS_2NLOS);

% Representación gráfica del RMSE CASO 3: 2 ANCLAS LOS y 3 ANCLAS NLOS:
titulo_2LOS_3NLOS = 'CASO 3: 2 ANCLAS LOS y 3 ANCLAS NLOS';
graficaRMSE (RMSE_2LOS_3NLOS, P, titulo_2LOS_3NLOS);

```

ANEXO B: BASE DE DATOS DE LAS MEDIDAS

En esta sección se muestra la base de datos de las medidas que han empleado los algoritmos de localización para la estimación de la posición del nodo móvil. Está constituida por las posiciones reales del nodo móvil, las posiciones de los nodos anclas, las medidas de distancia, y la condición de visión. Medidas que Carmelo García generó para el desarrollo práctico de su trabajo fin de grado con el equipo *EVK1000 Evaluation Kit* de la compañía DecaWave [10] y [11].

Base de datos para el total de posiciones reales del nodo móvil a localizar, 57 posiciones:

B.1 Posiciones reales del nodo móvil: `realPositions.txt`

Este fichero contiene todas las posiciones reales del nodo móvil en coordenadas x e y . 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. Carmelo lo hizo de esta manera para tener una mayor comodidad a la hora de tomar los datos, pero en los códigos de MATLAB, estos datos son transformados a la escala real.

```
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

B.2 Posiciones de los nodos anclas: fixedNodesPosition.txt
Este fichero contiene las posiciones de los nodos anclas en coordenadas x e y Estos valores están escalados y son interpretados en los códigos MATLAB de igual manera que el fichero anterior.
1,1 1,15 11,8 26,2 5,8

B.3 Medidas de distancia: `measuredDistance.txt`

Este fichero contiene las medidas de distancia de cada nodo ancla al nodo móvil, definida en metros. Si la medida de distancia es igual a 0, significa que no hay comunicación entre el nodo ancla y el nodo móvil o ambos coinciden en posición

```

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

B.4 Condición de visión: sightCondition.txt
Este fichero contiene la condición de visión de cada nodo ancla al nodo móvil:
<i>LOS</i> : 1
<i>NLOS</i> : 0
No hay comunicación entre el nodo ancla y el nodo móvil o ambos coinciden en posición: -1
<pre> -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 1, 1, 0, -1, 0 1, -1, 0, -1, 0 1, 1, -1, -1, 0 1, 1, -1, -1, 0 1, 1, -1, -1, 0 1, 1, -1, -1, 0 1, 1, -1, -1, 0 1, 1, -1, -1, 0 1, 1, -1, -1, 0 1, 1, -1, -1, 0 1, 1, -1, -1, 0 1, 0, 0, 1, 1 1, 0, 0, 1, 0 1, 0, 0, 1, 0 </pre>

A continuación, se muestra la base de datos que se han empleado en los casos de desglose en nodos anclas *LOS* o *NLOS*. Esta base de datos se ha generado para el desarrollo de este proyecto.

B.5 Caso 1: 4 anclas *LOS* y 1 anclas *NLOS*. 1 posición real del móvil a localizar.

Posición real del nodo móvil: 4_1_posicionRealMovil.txt
1, 3

Medidas de distancia: 4_1_medidasDistancia.txt
0.720, 5.653, 5.412, 12.599, 4.019

Condición de visión: 4LOS_1NLOS.txt
1, 1, 1, 1, 0

B.6 Caso 2: 3 anclas *LOS* y 2 anclas *NLOS*. 7 posiciones reales del móvil a localizar.

Posiciones reales del nodo móvil: 3_2_posicionRealMovil.txt
4, 2
4, 5
5, 6
6, 6
6, 7
7, 7
7, 8

Medidas de distancia: 3_2_medidasDistancia.txt
1.448, 6.890, 4.940, 10.764, 2.750
2.298, 6.096, 3.640, 12.077, 1.315
2.873, 5.147, 2.877, 11.036, 0.780
3.305, 5.436, 2.465, 10.527, 0.956
3.641, 4.882, 2.316, 10.612, 0.474
4.015, 5.303, 1.944, 10.253, 0.957
4.383, 5.056, 1.843, 10.513, 0.895

Condición de visión: 3LOS_2NLOS.txt
1, 0, 0, 1, 1

```
1, 0, 1, 0, 1  
1, 0, 1, 0, 1  
1, 0, 1, 0, 1  
1, 0, 1, 0, 1  
1, 0, 1, 0, 1  
1, 0, 1, 0, 1
```

B.7 Caso 3: 2 anclas LOS y 3 anclas NLOS. 28 posiciones reales del nodo móvil.

Posiciones reales del nodo móvil: [2_3_posicionRealMovil.txt](#)

```
1,5  
6,2  
8,2  
10,2  
12,2  
14,2  
4,4  
5,7  
5,8  
6,8  
7,5  
7,6  
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
```

Medidas de distancia: [2_3_medidasDistancia.txt](#)

```
1.810, 4.697, 5.488, 13.008, 4.105
```

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
1.882, 5.938, 4.012, 12.783, 1.805
3.505, 4.906, 2.835, 11.666, 0.238
4.156, 4.074, 2.856, 11.193, 0
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
4.125, 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

Condición de visión: 2LOS_3NLOS.txt
1, 1, 0, 0, 0
1, 0, 0, 1, 0
1, 0, 0, 1, 0
1, 0, 0, 1, 0
1, 0, 0, 1, 0
1, 0, 0, 1, 0
1, 0, 0, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1

0, 0, 1, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1
0, 0, 1, 0, 1