



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES

TRABAJO FINAL DE MÁSTER

LOCALIZACIÓN PRECISA DE ACTIVOS Y
HUMANOS EN ESPACIOS CERRADOS PARA
SEGURIDAD EN ENTORNOS INDUSTRIALES

MÁSTER EN AUTOMATIZACIÓN E INFORMÁTICA INDUSTRIAL

Alumno: Bárbara María Cabrera Castro

Director: Juan Francisco Blanes Noguera

Valencia, diciembre 2020



*A todos los que formaron parte:
de mi tiempo en Valencia,
del desarrollo de este proyecto,
del camino para llegar aquí.
A los que me esperaban en casa.*

Gracias.

ÍNDICE

I.	RESUMEN	8
II.	ABSTRACT	9
III.	RESUM	10
IV.	MEMORIA	11
1.	Introducción.....	12
1.1	Situación y Contexto.....	12
1.2	Motivación	12
1.3	Objetivos	12
1.3.1	Objetivo Académico.....	13
1.3.2	Objetivo Técnico	13
1.4	Estructura del proyecto	14
2.	Estado del arte	14
2.1	Localización mediante Bluetooth.....	14
2.2	Localización mediante Wifi	16
2.3	Localización mediante GPS	18
2.4	Localización mediante UWB	19
2.5	Comparativa.....	21
3.	Emplazamiento	21
4.	Sistemas integrados en la planta.....	22
4.1	Hardware	22
4.1.1	Hardware inicial	22
4.1.2	Hardware actual.....	25
4.2	Software	28
5.	Desarrollo del proyecto	28
5.1	Fase I.....	28
5.2	Fase II.....	40
5.3	Fase III.....	48
6.	Conclusiones y líneas abiertas	57
7.	Conclusions and open lines	58
8.	Normas y Referencias.....	60
8.1	Disposiciones Legales y Normas Aplicables	60
8.2	Softwares empleados.....	60
8.3	Referencias bibliográficas y documentales	61
9.	Definiciones y abreviaturas	62

9.1 Definiciones	62
9.2 Abreviaturas	62
V. ANEXO I.....	64
1. Código desarrollado en Java y tratamiento de los datos obtenidos	65
VI. ANEXO II	73
1. Planos originales y actualizaciones	74

ÍNDICE DE FIGURAS

Figura 1: Método de direccionamiento de señal Bluetooth mediante el ángulo de llegada.....	15
Figura 2: Método de direccionamiento de señal Bluetooth mediante el ángulo de salida.....	16
Figura 3:Sistema Wifi.....	16
Figura 4: Sistema de localización mediante Wifi.....	17
Figura 5:Sistema de localización mediante Wifi (Tiempos).....	17
Figura 6: Interferencias por Multipath.....	18
Figura 7:Triangulación GPS.....	18
Figura 8: Método TWR.....	20
Figura 9: Emplazamiento Universitat Politècnica de València (Dirección: Camino de Vera S/n, Edificio 8G, 46022 Valencia).....	22
Figura 10: Plano inicial del laboratorio.....	22
Figura 11: Antenas (versión original).....	23
Figura 12: Tags móviles.....	24
Figura 13: Servidor.....	24
Figura 14: Robotino.....	25
Figura 15: Antenas (versión actual).....	25
Figura 16:Plano nueva ubicación antenas.....	26
Figura 17: Cambio a método TWR.....	27
Figura 18: Tags de desarrollo.....	27
Figura 19: Esquema Software.....	28
Figura 20: Coordenadas de origen plano actual de laboratorio.....	29
Figura 21: Pantalla de configuración de tags móviles.....	29
Figura 22: Configuración entorno TDOA.....	30
Figura 23:: Plano antiguo con tags en RT.....	33
Figura 24: Plano antiguo con tags en RT (2).....	34
Figura 25: Planta piloto FESTO.....	34
Figura 26: Trayectoria tag 5128 – Kalman full (antenas originales fase I).....	37
Figura 27: Trayectoria tag 5128 – Kalman Outlier (antenas originales fase I).....	37
Figura 28: Trayectoria tag 5128 – UWB full (antenas originales fase I).....	38
Figura 29: Trayectoria tag 5128 – UWB Outlier (antenas originales fase I).....	38
Figura 30: Trayectoria tag 5128 – Kalman crudo con filtro de mediana matlab (antenas originales fase I).....	39
Figura 31: Trayectoria tag 5128 - UWB crudo con filtro de mediana matlab (antenas originales fase I).....	39
Figura 32: Coordenadas de origen plano actual de laboratorio.....	40
Figura 33: Trayectoria tag 5128 – Kalman full (antenas originales fase II).....	45
Figura 34: Trayectoria tag 5128 – Kalman Outlier (antenas originales fase II).....	45
Figura 35: Trayectoria tag 5128 – UWB full (antenas originales fase II).....	46
Figura 36: Trayectoria tag 5128 – UWB Outlier (antenas originales fase II).....	46
Figura 37: Trayectoria tag 5128 – Kalman crudo con filtro de mediana matlab (antenas originales fase II).....	47
Figura 38: Trayectoria tag 5128 – UWB crudo con filtro de mediana matlab (antenas originales fase II).....	47

Figura 39: Pantalla de configuración de tags de desarrollo.....	48
Figura 40: Configuración entorno TWR.....	49
Figura 41: Configuración control tags.....	49
Figura 42: Trayectoria tag 27191 – Kalman Moving Median.....	53
Figura 43: Trayectoria tag 27191 – Kalman FIR.....	54
Figura 44: Trayectoria tag 27191 – Kalman Moving Average.....	54
Figura 45: Trayectoria tag 27191 – Kalman Moving Median.....	55
Figura 46: Trayectoria tag 27191 – Kalman Moving FIR.....	55
Figura 47: Trayectoria tag 27191 – Kalman Moving Average.....	56
Figura 48: Trayectoria tag 27191 – Kalman crudo con filtro de mediana móvil Matlab.....	56
Figura 49: Trayectoria tag 27191 – Kalman crudo con filtro de mediana móvil Matlab.....	57
Figura 50: Eclipse java – Estructura del proyecto.....	65
Figura 51: Eclipse java – Estructura clase Coordinates.....	66
Figura 52: Eclipse java – Estructura clase Data.....	67
Figura 53: Eclipse java – Estructura clase Metrics.....	67
Figura 54: Eclipse java – Estructura clase Orientation.....	67
Figura 55: Eclipse java – Estructura clase Rates.....	68
Figura 56: Eclipse java – Estructura clase TagData.....	68
Figura 57: Eclipse java – Estructura clase TagPosition.....	68
Figura 58: Eclipse java – Librerías.....	69
Figura 59: Eclipse java –clase MQTT.....	69
Figura 60: Eclipse java –método messageArrived (1).....	69
Figura 61: Eclipse java –método messageArrived (2).....	70
Figura 62: Eclipse java –método messageArrived (3).....	70
Figura 63: Eclipse java –método messageArrived (4).....	70
Figura 64: Eclipse java –métodos conecctionLost y deliveryComplete.....	71
Figura 65: Eclipse java –métodos main y comprobarArchivo.....	71
Figura 66: Eclipse java –método run.....	72
Figura 67: Plano original.....	74
Figura 68: Boceto plano original.....	74
Figura 69: Boceto plano actual.....	75

ÍNDICE DE TABLAS

Tabla 1: Tabla comparativa de los métodos TWR y TDOA.	21
Tabla 2: Tabla comparativa de los sistemas de localización.	21
Tabla 3: Identificadores tags utilizados.	23
Tabla 4: Identificadores tags utilizados fase I.	30
Tabla 5: Filtros disponibles en el método TDOA.	31
Tabla 6: Media módulo. UWB Outlier fase I.	31
Tabla 7: Media módulo. UWB Full fase I.	31
Tabla 8: Media módulo. Kalman Outlier fase I.	32
Tabla 9: Media módulo. Kalman Full fase I.	32
Tabla 10: Media módulo. UWB crudo fase I.	32
Tabla 11: Media módulo. Kalman crudo fase I.	32
Tabla 12: Media módulo. UWB crudo y filtro de mediana fase I.	33
Tabla 13: Media módulo. Kalman crudo y filtro de mediana fase I.	33
Tabla 14: Identificadores tags utilizados fase II.	40
Tabla 15: Media módulo. UWB Outlier fase II.	41
Tabla 16: Media módulo. UWB Full fase II.	41
Tabla 17: Media módulo. Kalman Outlier fase II.	41
Tabla 18: Media módulo. Kalman Full fase II.	41
Tabla 19: Media módulo. UWB crudo fase II.	42
Tabla 20: Media módulo. Kalman crudo fase II.	42
Tabla 21: Media módulo. UWB crudo y filtro de mediana fase II.	42
Tabla 22: Media módulo. Kalman crudo y filtro de mediana fase II.	42
Tabla 23: Identificadores tags utilizados fase III.	50
Tabla 24: Filtros disponibles en el método TWR.	50
Tabla 25: Media módulo. UWB FIR.	50
Tabla 26: Media módulo. UWB Moving Average.	50
Tabla 27: Media módulo. UWB Moving Median.	50
Tabla 28: Media módulo. UWB crudo Fase III.	51
Tabla 29: Media módulo. UWB crudo con filtro de mediana móvil.	51
Tabla 30: Media módulo. Kalman FIR.	51
Tabla 31: Media módulo. Kalman Moving Average.	51
Tabla 32: Media módulo. Kalman Moving Median.	51
Tabla 33: Media módulo. Kalman crudo Fase III.	51
Tabla 34: Media módulo. Kalman crudo con filtro de mediana móvil.	52
Tabla 35: Resumen fases del proyecto.	57
Tabla 36: Summary of project phases.	59
Tabla 37: Eclipse java – Estructura de las clases y sus métodos.	66
Tabla 38: Eclipse java – métodos clase MQTT.	69

ÍNDICE DE GRÁFICOS

Gráfico 1: Coordenadas x e y - tag 5125 (antenas originales fase I).....	35
Gráfico 2: Coordenadas x e y - tag 5139 (antenas originales fase I).....	35
Gráfico 3: Coordenadas x e y - tag 5147 (antenas originales fase I).....	35
Gráfico 4: Coordenadas x e y - tag 5171 (antenas originales fase I).....	36
Gráfico 5: Coordenadas x e y - tag 5173 (antenas originales fase I).....	36
Gráfico 6: Coordenadas x e y - tag 5125 (antenas originales fase II).	43
Gráfico 7: Coordenadas x e y - tag 5139 (antenas originales fase II).	43
Gráfico 8: Coordenadas x e y - tag 5147 (antenas originales fase II).	43
Gráfico 9: Coordenadas x e y - tag 5171 (antenas originales fase II).	44
Gráfico 10: Coordenadas x e y - tag 5173 (antenas originales fase II).	44
Gráfico 11: Coordenadas x e y - tag 27188.....	52
Gráfico 12: Coordenadas x e y - tag 27252.....	52

ÍNDICE DE ECUACIONES

Ecuación 1: Tiempo entre tag y antena (TWR).	20
Ecuación 2: Distancia entre tag y antena (TWR).	20

I. RESUMEN

Los sistemas RTL (Real Time Location) han permitido hasta el momento el desarrollo de aplicaciones de localización basadas en señales de radiofrecuencia con precisión suficientemente aceptable para la localización, aunque siempre por encima de las decenas de centímetros. Con la aparición de los sistemas de UWB (Ultra Wide Band), se están alcanzando precisiones teóricas por debajo de los 10 centímetros teóricos. Este trabajo aborda la evaluación de un sistema UWB y su aplicación a la determinación de la posición de humanos y robots en entornos industriales, para su integración en sistemas de seguridad de los entornos de trabajo.

PALABRAS CLAVE: UWB, RTL, localización, radiofrecuencia, posicionamiento, industria, sistema de seguridad.

II. ABSTRACT

RTL (Real Time Location) systems have allowed the development of localisation applications based on radio frequency signals with an acceptable accuracy for positioning but always above dozens of centimeters. With the occurrence of UWB (Ultra Wide Band) systems, theoretical accuracies below 10 centimeters are being achieved. This paper deals with the evaluation of a UWB system and its application to humans and robots position determination in industrial environments, for its integration into safety systems in working environments.

KEY WORDS: UWB, RTL, localisation, radio frequency, positioning, industry, safety systems.

III. RESUM

Els sistemes RTL (Real Time Location) han permés fins al moment el desenrotllament d'aplicacions de localització basades en senyals de radiofreqüència amb precisió prou acceptable per a la localització, encara que sempre per damunt de les desenes de centímetres. Amb l'aparició dels sistemes d'UWB (Ultra Wide Band), s'estan aconseguint precisions teòriques per davall dels 10 centímetres teòrics. Este treballat aborda l'avaluació d'un sistema UWB i la seua aplicació a la determinació de la posició d'humans i robots en entorns industrials, per a la seua integració en sistemes de seguretat dels entorns de treball.

PARAULES CLAU: UWB, RTL, localització, radiofreqüència, posicionament, indústria, sistema de seguretat.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES

TRABAJO FINAL DE MÁSTER

MEMORIA

**MÁSTER EN AUTOMATIZACIÓN E INFORMÁTICA
INDUSTRIAL**

Título del proyecto:

**LOCALIZACIÓN PRECISA DE ACTIVOS Y HUMANOS EN
ESPACIOS CERRADOS PARA SEGURIDAD EN ENTORNOS
INDUSTRIALES**



1. Introducción

1.1 Situación y contexto

Este proyecto se basa en las tecnologías de localización mediante UWB, por ello, se muestra una pequeña introducción a los sistemas de localización existentes

Actualmente para la localización de objetos y persona se utilizan sistemas Bluetooth, Wifi o GPS. Los sistemas Bluetooth se basan en una tecnología de corto alcance mediante radiofrecuencias en la banda ISM de los 2.4 GHz. Así mismo, los sistemas Wifi permiten la conexión inalámbrica mediante un punto de acceso inalámbrico que les permite conectarse entre sí o a internet. Finalmente, el uso de sistemas GPS es muy común, ya que se trata de un sistema de posicionamiento a nivel global mediante el uso de satélites y la trilateración, es decir, la determinación de la posición mediante cálculos matemáticos entre las posiciones de referencia y la posición del objeto.

Se espera que este trabajo ayude en la investigación para la implementación de entornos seguros en las industrias actuales ya que, el sector de la automatización está en auge.

1.2 Motivación

Este TFM surge de la imperante necesidad de generar entornos seguros en la industria para los trabajadores, debido a que el avance tecnológico implementa en ella medios automatizados para la realización de tareas. Por ello, es necesario investigar las posibilidades en el campo del posicionamiento mediante banda ultra ancha (UWB) frente a métodos como Wifi, Bluetooth o GPS.

Los entornos robotizados cuentan con maquinaria que ya sea por nivel de peligrosidad propia de la máquina, o por la posibilidad de errores humanos, suponen un peligro para la integridad física de los operarios que trabajan en la planta. Con este trabajo, se espera sentar las bases de un proyecto que permita controlar de forma precisa dónde está situado cada operario y sus movimientos (en concreto los movimientos de sus manos) para que el robot posea dicha información y en caso de colisión o entrada en contacto, éste cese su actividad, o lo evite en la medida de lo posible, para que el trabajador no sufra daños.

Mejorar la productividad eliminando los riesgos propios del trabajo interactivo entre humanos y robots, es el siguiente paso para avanzar en la industria automatizada.

1.3 Objetivos

En este apartado se plantean los objetivos a lograr con el desarrollo del proyecto, para ello se definen los objetivos generales y específicos del proyecto.

1.3.1 Objetivo académico

Este proyecto tiene, como fin fundamental, superar la asignatura Trabajo Final de Máster (TFM) para el máster en Automatización e Informática Industrial. Éste tiene como propósito, el uso de los conocimientos adquiridos a lo largo del mismo, además de aportar nuevas aptitudes y conocimientos propios del proyecto elegido. Entre ellos, el protocolo MQTT, la adquisición de competencias en sistemas de localización mediante UWB y la creación de entornos seguros para los trabajadores industriales en caso de compartir espacios físicos con robots móviles.

1.3.2 Objetivo técnico

Por otro lado, el objetivo técnico primordial de este Trabajo de Final de Máster es la evaluación y la adaptación de un sistema de localización basado en UWB para una planta compuesta de robots de trabajo. Se busca crear un entorno de trabajo seguro en el que los sistemas automatizados y las personas puedan trabajar de forma conjunta eliminando o minimizando el riesgo de colisión entre ellos.

Así pues, se hace uso de un sistema llamado POZYX que se comercializa desde 2015 y que permite el posicionamiento en los ejes X, Y y Z mediante el uso de antenas y banda ultra ancha (UWB), lo cual permite un posicionamiento bastante preciso en entornos interiores. Lo interesante de este sistema es que permite realizar un seguimiento de objetos en movimiento, de tal forma que se puede conocer donde está un sistema en tiempo real.

1.4 Estructura del proyecto

Este primer capítulo denominado introducción, se exponen las motivaciones y objetos del proyecto que lo han hecho posible. Con ello se pretende que el lector conozca la tecnología actual en la industria y las ideas que han llevado a desarrollarlo con un Trabajo Final de Máster (de aquí en adelante denominado TFM).

Posteriormente, veremos en el capítulo dos los sistemas de localización existentes que servirían para el concepto que se plantea en el apartado anterior, descartando desde un principio los sistemas inviables o no adecuados para nuestro objetivo como, por ejemplo, los sistemas de infrarrojos o ultrasonido. Así, al final de este se realizará una comparativa y se explicará qué sistema de los presentados, cumple mejor con las necesidades de nuestro caso.

En el capítulo tres de la memoria, se presenta el emplazamiento donde tendrá lugar el desarrollo del proyecto de manera física.

Posteriormente, en el capítulo cuatro, veremos las normas y las referencias. Esto es, que se mostrarán disposiciones legales y normas aplicables, así como el software empleado y las referencias bibliográficas y documentales utilizadas para el desarrollo del mismo.

El quinto capítulo engloba las abreviaturas y definiciones necesarias para que el lector del documento pueda comprender el texto de una forma más práctica y rápida.

Una vez se muestra el glosario de vocabulario necesario para entender el contexto del proyecto, se muestra el capítulo seis, que contiene los sistemas integrados en la planta de trabajo. Éste consta de dos partes: hardware y software. En el hardware encontraremos los equipos físicos que conforman el sistema de localización en el laboratorio (ya sean los originales o los actuales) y en el área de software, encontraremos las funcionalidades de la aplicación predeterminada del sistema y sus características.

Finalmente, encontramos el capítulo siete, el último capítulo de la memoria. En el se redacta el desarrollo del proyecto y por tanto se divide en tres fases: fase I, fase II y fase III. La primera fase abarca el sistema inicial que se encontraba en el laboratorio al comenzar el proyecto, las pruebas que se hicieron en el y los resultados obtenidos con unos comentarios sobre ellos. La segunda y última fase consta de la actualización del sistema del laboratorio, la repetición de las pruebas y los resultados extraídos de ellas. Por último, la tercera fase es la comparación de los resultados de las fases anteriores, teniendo de esta manera una visión más clara del proceso, los resultados y las posibles conclusiones que se presentarán más adelante en los Anexos VI y VII.

2. Estado del arte

Actualmente hay varios sistemas de posicionamiento, como ya hemos visto, que se recogen en el IEEE 802, es decir, en el proyecto 802 del Instituto de Ingenieros Eléctricos y Electrónicos. Este comité, desarrolla las normas y prácticas más recomendadas para redes locales y metropolitanas entre otras. Para cada área de trabajo (wifi, Bluetooth, Ethernet...) existe un grupo de trabajo.^[4]

Así pues, veamos de forma más detallada, cómo funcionan, en qué entornos es más recomendable su uso y pros y contras para cada uno de ellos, de tal manera que desarrollemos una visión general de los mismos.

2.1 Localización por Bluetooth

El Bluetooth (IEEE 802.15) un sistema para redes inalámbricas de área personal (WPAN) que permite la transmisión de datos sin conexión mediante radiofrecuencias banda ISM sin licencia entre los 2,402 GHz y los 2,480 GHz. Este sistema se estructura mediante maestros (que proporciona la referencia de sincronización a partir de su reloj interno) y esclavos (el resto de los dispositivos), lo que conforma una red denominada “piconet”. Los componentes de una piconet deben estar siempre en proximidad directa, es decir, a un máximo de 10 metros o 100 si se utilizan repetidores, además de tener activado el sistema bluetooth.^[2]

Existen distintas versiones de este sistema Bluetooth, desde la primera creada en 1999, hasta la más reciente de 2019. Esta última ha sido denominada Bluetooth 5.1.

En esta versión, mantiene la velocidad de transmisión de datos de 2 Mbps y casi cuatro veces más distancia de alcance (hasta 240 metros) que ya se presentaba en su predecesor, el 5.0. Todo esto es aplicable solamente al tipo de bluetooth de baja energía, es decir, el

que ha sido diseñado para dispositivos que están conectados 24h, como aparatos domóticos o elementos del IoT.

Las nuevas funciones del Bluetooth 5.1, permiten la localización de direcciones, es decir, permite determinar la dirección desde la que proviene la señal. Para ello, se utiliza el ángulo de llegada (AoA) y el ángulo de salida (AoD).

- » Ángulo de llegada: en este método, el dispositivo que desea conocer la dirección envía una señal de búsqueda utilizando una sola antena. El receptor, en cambio, posee varias antenas de forma matricial que perciben la señal y conforme la señal atraviesa la matriz hasta la antena transmisora, es capaz de distinguir una diferencia de fase de señal debido a la diferencia de distancia. El método AoA está diseñado para usarse en sistemas de localización de tiempo real.

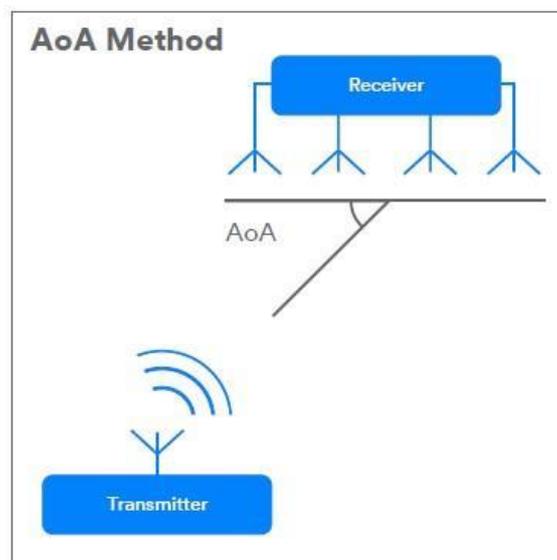


Figura 1: Método de direccionamiento de señal Bluetooth mediante el ángulo de llegada.

- » Ángulo de salida: en este método, el dispositivo del que queremos conocer la dirección actúa como una baliza de localización transmitiendo una señal con sus antenas dispuestas en forma de matriz. El receptor, con su única antena, toma muestras de IQ para calcular la dirección de la señal relativa. El método de búsqueda de dirección de AoD está destinado a soluciones IPS.

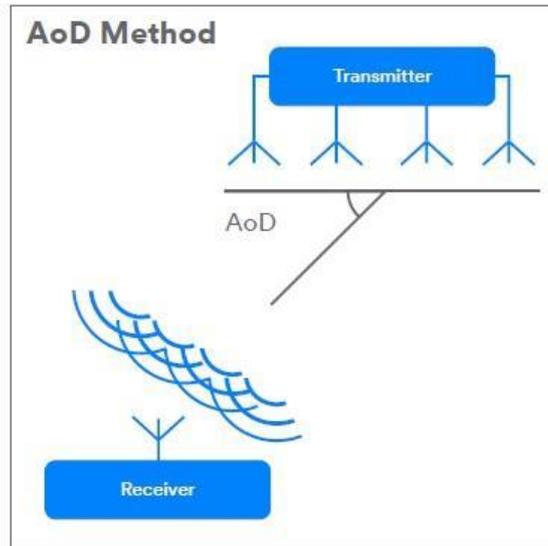


Figura 2: Método de direccionamiento de señal Bluetooth mediante el ángulo de salida.

Mientras que las primeras versiones del Bluetooth no son demasiado exactas y, por tanto, pueden suscitar un error de entre 1 y 5 metros, la versión 5.1 permite el posicionamiento con un error de centímetros, aunque sigue siendo una tecnología muy reciente y con pocos sistemas comerciales disponibles. ^[3]

2.2 Localización por Wifi

El Wifi (IEEE 802.11) es un sistema inalámbrico de conexión entre dispositivos que, además, permite la conexión a internet mediante radiofrecuencias o infrarojos. Para este sistema necesitamos de un comunicador (generalmente un adaptador o tarjeta inalámbrica) que se encarga de enviar los datos como señal de radiofrecuencia y un router (que incluye un módulo modem) que envía el paquete de datos a Internet a través de una conexión cableada de Ethernet (IEEE 802.3). Este sistema es capaz de transmitir a frecuencias de entre 2.4 y 5 GHz y cuanto mayor es la frecuencia, más información puede transmitir la señal. ^[4]

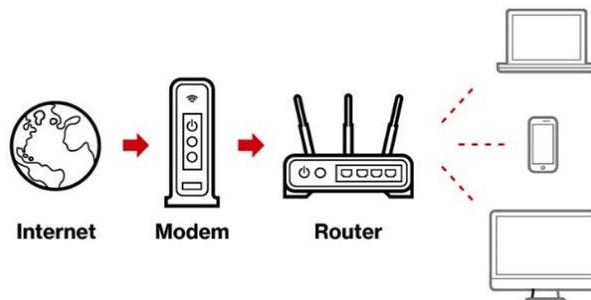


Figura 3: Sistema Wifi.

La banda de 2.4GHz se encuentra definida en los estándares del IEEE 802.11b, IEEE 802.11g e IEEE 802.11n, lo que permite velocidades de 11Mbps, 54Mbps y 300Mbps respectivamente. Debemos saber que esta banda se ve afectada por interferencias debido

a que ese rango de frecuencias es utilizado, como hemos visto anteriormente por bluetooth u otros sistemas inalámbricos.

La banda de 5GHz, en cambio, es la más actual y se le aplica el estándar IEEE 802.11ac, donde encontramos que no comparte el canal de frecuencias con otros sistemas inalámbricos y, por tanto, es más fiable. Esta banda, ofrece, además, una velocidad de transmisión de datos cercana a 1Gbps.

Para utilizar este método como localizador, se usa el tiempo que la señal está en transmisión, es decir, el tiempo que tarda la señal en llegar al receptor. Para tener una localización exacta, se necesita de al menos 3 puntos de acceso para generar una posición en 3 dimensiones.



Figura 4: Sistema de localización mediante Wifi.

Una localización exacta necesita de una sincronización de dispositivos a nivel de nanosegundos, lo cual ocurre mediante un envío de señal de ida y vuelta. Una vez ya se ha realizado la sincronización, se pueden enviar paquetes de datos (ver figura x). En esos paquetes, el punto de acceso indica la hora de salida (T1) y lo envía al receptor, que calcula cuanto tiempo ha tardado en llegar y lo envía de vuelta al emisor con su hora de llegada (T2) y la nueva hora de salida (T3). Finalmente, llega de nuevo al emisor y anota la hora de llegada (T4).

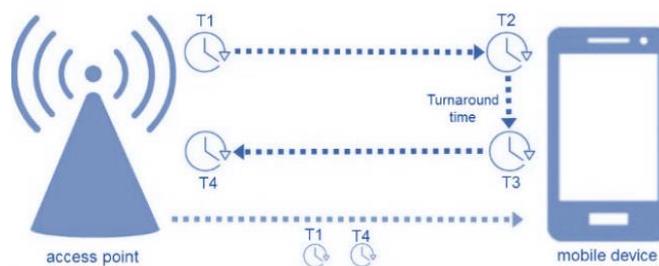


Figura 5: Sistema de localización mediante Wifi (Tiempos).

En este caso, la precisión de la localización mediante un sistema Wifi depende de varios factores, entre ellos el ancho de banda del canal, el tráfico de datos y el Multipath, es decir si la señal que se recibe proviene de forma directa o si es recibida mediante rebotes de la señal en otros objetos. ^[5]

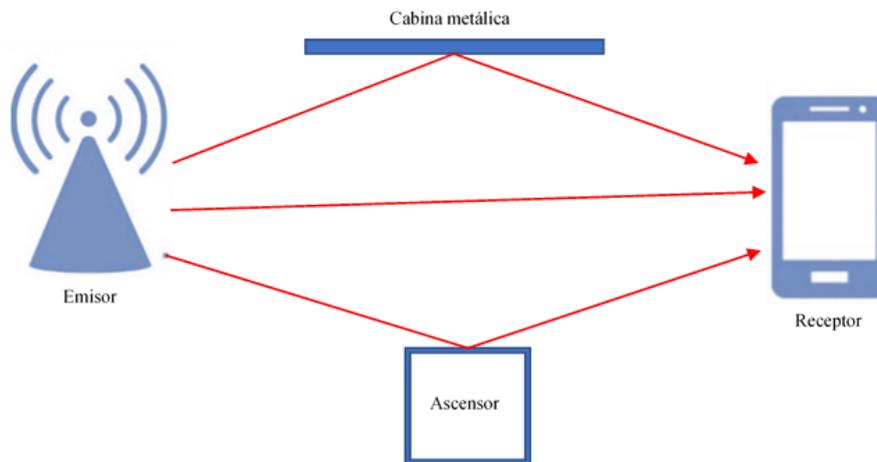


Figura 6: Interferencias por Multipath.

2.3 Localización por GPS

Un sistema GPS es un sistema de posicionamiento global que permite obtener la localización de un dispositivo mediante el uso de satélites. Actualmente los sistemas GPS utilizan una constelación de 24 a 32 satélites y necesitan de un canal de frecuencia de 1GHz para comunicarse con ellos. [6]

No se requiere de antenas para su uso, pero no son capaces de ubicar sujetos dentro de edificios o estructuras que no permitan las conexiones. Así pues, aunque en el exterior es un instrumento de localización bastante preciso, ya que muestra una media de error de entre 6 y 10 metros, en interiores no es efectivo.

Para realizar una localización con este sistema, se precisa el uso de al menos 3 satélites con los que poder realizar una triangulación de señales como la que se muestra en la figura 7:

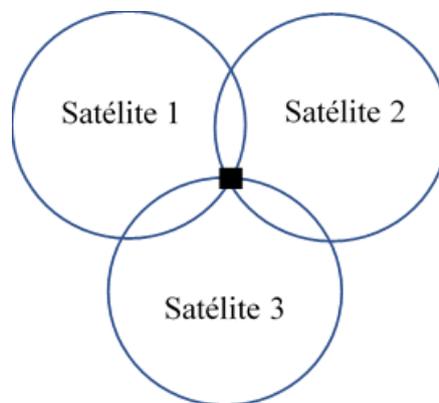


Figura 7: Triangulación GPS.

Podemos ver, que de esta manera la zona de intersección de los receptores indica la posición del emisor.

Para mejorar este sistema de localización, existe el sistema diferencial del GPS, el DGPS. En este, se sitúan un número de radioemisores o radiobalizas que permiten una localización prácticamente exacta. En caso de tener un sistema GPS bastaría con acoplarle un receptor Rasant que capte las señales de las balizas.

Sin embargo, el uso de GPS de alta sensibilidad es capaz de proporcionar ubicaciones en algunos entornos interiores, aunque las señales llegan muy atenuadas debido a los materiales del entorno. Se ha probado que este tipo de posicionamiento es capaz de ubicar un sujeto tras unas tres paredes de ladrillo, pero se trata de un sistema muy poco preciso.
[7]

2.4 Localización mediante UWB

Los sistemas de localización por Ultra Wide Band (UWB) utilizan un rango de frecuencias mayor que el habitual ya que permite transmitir los datos a mayor velocidad. Estos sistemas se basan en el envío de pulsos muy breves con los que se puede medir la distancia. Para conocer la distancia de nuestros objetos (lo cuales estarían identificados con un tag) se precisa conocer las posiciones exactas de las antenas que recibirán las señales y aplicar alguno de los siguientes métodos:

- » AOA: en este método, el dispositivo que desea conocer la dirección envía una señal de búsqueda utilizando una sola antena y el receptor, compuesto de varias antenas en una forma matricial, es capaz de distinguir una diferencia de fase de señal debido a la diferencia de distancia.
- » RSSI: El método de indicador de la fuerza de la señal recibida se basa en la medida de la intensidad de la señal recibida. Cuanto más cerca está el emisor (tag) del receptor (antena) más intensa será la señal que se recibe y por tanto, de esta manera, se puede estimar mediante la atenuación de la señal la distancia a la que se encuentra el emisor. Hay que tener en cuenta, que el RSSI indica la intensidad recibida, pero no la calidad de la señal, por lo que esta última se determina comparando la intensidad de la señal con la relación entre la señal y el ruido.
- » TDOA: Este método se basa en la medida de la diferencia temporal de la señal, es decir, la diferencia entre el tiempo de llegada y el tiempo de salida. Para utilizar este método, se necesita que las funciones de tiempo estén sincronizadas de forma muy precisa, si no las medidas serían erróneas. Este tipo de método funciona bien en plantas relativamente vacías, ya que no encuentra obstáculos y con una única medida se obtienen valores bastante buenos.
- » TOA: funciona mediante el tiempo de llegada. Esto significa que se utiliza la intersección entre las señales generadas por varias antenas distribuidas por la zona a controlar. la distancia entre la antena y el tag se obtiene calculando el tiempo de

propagación de la señal de forma unidireccional, ya que el tag es el que emite y la antena la que recibe el pulso.

- » **TWR:** el “two way ranging” es un método de localización por el cual la se intercambian tres mensajes entre la antena y el tag. Inicialmente, el tag envía un mensaje a la antena y guarda el tiempo de envío. La antena guarda el tiempo de llegada de dicho mensaje y luego responde. Cuando el mensaje llega al tag, éste guarda un segundo tiempo de llegada y compone un mensaje con todos los datos obtenidos. Se puede comprender mejor en la figura 8:

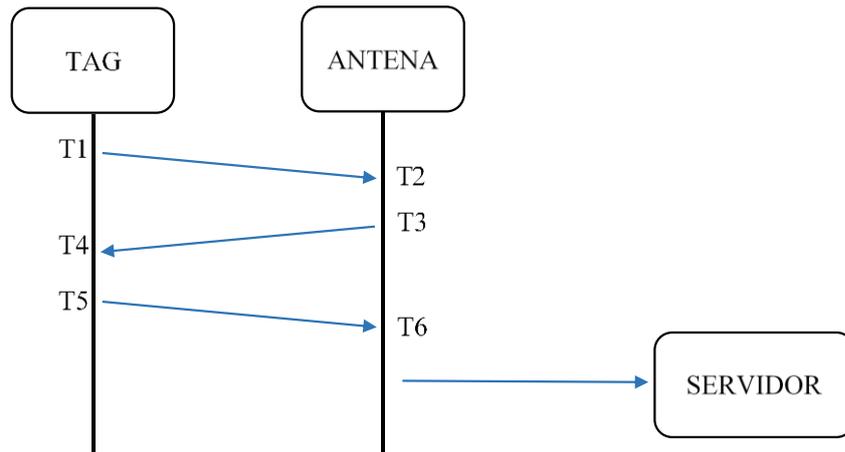


Figura 8: Método TWR.

De esta manera, podemos calcular el tiempo entre ambos como:

$$Tiempo = \frac{[(T4 - T1) - (T3 - T2) + (T6 - T3) - (T5 - T4)]}{4}$$

Ecuación 1: Tiempo entre tag y antena (TWR).

Por lo que la distancia entre el tag y la antena viene dada por:

$$Distancia = Tiempo \cdot Velocidad\ de\ la\ luz$$

Ecuación 2: Distancia entre tag y antena (TWR).

Este método, es bastante útil en casos industriales, ya que el envío de varios mensajes es capaz de corregir los errores en las mediciones en casos donde el entorno es más complicado y contiene mayor cantidad de obstáculos.

En nuestro caso, sin embargo, solo podremos aplicar los métodos TDOA y TWR debido a que son los únicos que vienen integrados en nuestro sistema para su uso. Veamos una comparativa teórica sobre estos dos métodos.

	Sincronización antenas	Energía necesaria	Disponibilidad de la información	Otros
TDOA	Sí	Muy baja	Servidor	Poca precisión de posicionamiento fuera del área de las antenas y deficiente en eje z
TWR	No	Alta	Tags y servidor	Corrección de la precisión mediante envío de varios mensajes.

Tabla 1: Tabla comparativa de los métodos TWR y TDOA.

Por lo que, en un principio, a excepción del uso de la batería, se estima que el método más útil en nuestro caso si utilizáramos UWB es el TWR.

2.5 Comparativa

Una vez hemos visto los sistemas de localización más utilizados y fiables, veamos cuál de ellos es el más adecuado para nuestro caso mediante una tabla comparativa que recoge los datos generales de cada uno.

	Error teórico	Banda de frecuencia	Transmisión de datos	Uso en interiores
Bluetooth	1-5 m	2.4 GHz	50 Mbps	Sí
WIFI	1-5 m	2.4 GHz 5 GHz	10 Gbps	Sí
GPS	6-10 m	1.6 GHz	-	No
UWB	10-30 cm	3.1-10.6 GHz	100-500 Mbps	Sí

Tabla 2: Tabla comparativa de los sistemas de localización.

Así pues, observando la tabla anterior y comprendiendo el funcionamiento básico de los sistemas descritos en los puntos anteriores, podemos concluir que, la decisión más correcta en cuanto a precisión, uso en interiores y banda de frecuencia es la UWB.

Esto nos permitirá trabajar en entornos de interior con errores de localización mucho más pequeños que otros sistemas y trabajar en una banda de frecuencia mayor que nos permite enviar más información en menos tiempo.

3. Emplazamiento

La realización de este proyecto tiene lugar en el Laboratorio de Industria 4.0 del ai2 dentro de la propia Universitat Politècnica de València, en el edificio 8G, situado en la Provincia de Valencia. Se muestra el emplazamiento en la figura 9 a continuación.



Figura 9: Emplazamiento Universitat Politècnica de València
(Dirección: Camino de Vera S/n, Edificio 8G, 46022 Valencia).

4. Sistemas integrados en la planta

4.1 Hardware

4.1.1 Hardware inicial

En el laboratorio se presenta un sistema de posicionamiento de la marca POZYX conformado por:

- Antenas: Seis antenas dispuestas de forma perimetral al área del laboratorio, es decir enfrentadas por parejas a lo largo del eje y del laboratorio como se muestra en la figura 10, representadas en rojo.

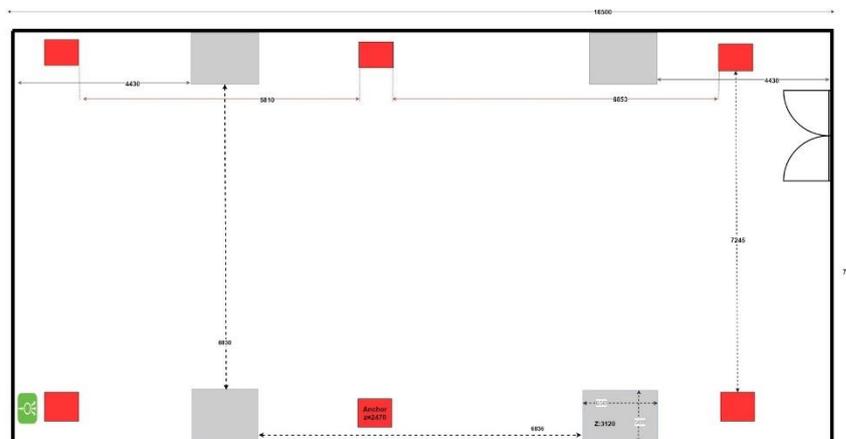


Figura 10: Plano inicial del laboratorio.

Inicialmente, las antenas (figura 11) poseían una versión de firmware 1.2 y trabajaban únicamente con el método TDOA, por el que los tags generan de una forma periódica un parpadeo UWB que es recibido por al menos dos antenas que miden la diferencia temporal y dan un valor aproximado de posicionamiento.



Figura 11: Antenas (versión original).

- Tags móviles: en este campo, se hace referencia a los 4 dispositivos móviles que se utilizan para obtener la posición de objetos móviles (robots o personas) o fijos (elementos del entorno que desean ser monitorizados junto a objetos móviles).

Nº TAG	MODELO
5125	MÓVIL
5139	MÓVIL
5147	MÓVIL
5171	MÓVIL

Tabla 3: Identificadores tags utilizados.

Una muestra de los tags utilizados se ve en la imagen 12, donde podemos ver los tags móviles como los dispositivos de color blanco. Para una mejor localización a simple vista, se les ha colocado unas pegatinas con sus números de identificación. Además, es importante saber que este tipo de tags solo funciona con el método TDOA, ya que no están diseñados para recibir señales, solo para enviar.



Figura 12: Tags móviles.

- Servidor: el último elemento de hardware del cual se dispone para el sistema es un servidor (o Gateway) que permite la conexión a la app del sistema. Para conectarnos a él, simplemente se deberá implementar un código que mediante un puente MQTT se suscriba a los “topics” del servidor que se envían a través de la dirección IP correspondiente.



Figura 13: Servidor.

- Robotino: adicionalmente, se utilizará un robot móvil llamado Robotino de la marca FESTO, con el que se probará el uso del sistema en el seguimiento de una trayectoria. Este robot, controlado mediante Visual (utilizando lenguaje C++) nos permitirá introducir unas coordenadas para definir la trayectoria. Para ello, solo debemos colocar el Robotino en unas coordenadas de origen (las que nosotros decidamos) y a partir de ahí indicar el recorrido en metros que debe realizar para cada eje.



Figura 14: Robotino.

4.1.2 Hardware actual

Para la mejora de la precisión en el sistema de localización, se propone la sustitución de las antenas antiguas por unas en su versión más actual. Estas antenas, no solo son diferentes a nivel visual, sino que incluyen una funcionalidad nueva, la localización por TWR.



Figura 15: Antenas (versión actual).

Se realizó un pedido de 8 antenas nuevas, 2 más que en el sistema inicial, y se propuso una nueva disposición para ellas en el laboratorio. Como en este caso, no se precisa de una localización en el eje Z, es decir, solo se utilizan las medidas en el eje x e y del plano independientemente de su altura, no se modifican las posiciones en dicho eje y se modifican levemente las coordenadas de las seis antenas anteriores para 6 de las nuevas



Figura 17: Cambio a método TWR.

Una vez nos hemos asegurado que, efectivamente, nuestras antenas están en el método de localización correcto, debemos asegurarnos que nuestros tags de desarrollo también están configurados de la misma manera.

Denominamos tags de desarrollo a los “developer tags”. Éstos son placas compatibles con Arduino (Uno R3, Nano y Mega mediante el I2C) y con Raspberry Pi (a través de micro USB). En este caso, la diferencia fundamental con los tags móviles radica en que pueden ser configurados con diferentes parámetros y pueden ser instalados en robots o estaciones industriales de una forma más práctica. Aunque ya se disponía de estos tags en la versión original del sistema, no se comienzan a utilizar hasta la fase III del proyecto, por lo que pueden considerarse parte del software actual.

En el caso de estos tags, la actualización del firmware a la versión 2.2 nos permite el uso de TWR, a diferencia de los tags móviles que no pueden actualizarse y por tanto solo disponen de localización mediante TDOA.



Figura 18: Tags de desarrollo.

Para configurar estos tags necesitamos la herramienta DfuSeDemo mediante el cual instalaremos la versión 2.2 comentada anteriormente y el software Pozyx Device Configurator para comprobar que la instalación del firmware ha sido un éxito y editarlo para que funcione con el método especificado. Todo esto está explicado en la

documentación de Pozyx de la web oficial. Sin embargo, es importante la utilización de un cable microUSB que permita la carga y la transferencia de información al mismo tiempo, ya que, si escogemos, por ejemplo, un cable muy fino, es probable que no podamos configurarlo correctamente.

4.2 Software

Para el desarrollo del proyecto, no solo se ha tenido en cuenta la parte física compuesta por el hardware, sino que el software ha supuesto una parte imprescindible, sobre todo para la recopilación de los datos del sistema.

En el siguiente esquema, podemos ver el uso de los sistemas de software para la obtención de dichos datos y en los anexos, se puede consultar el código utilizado para ello.

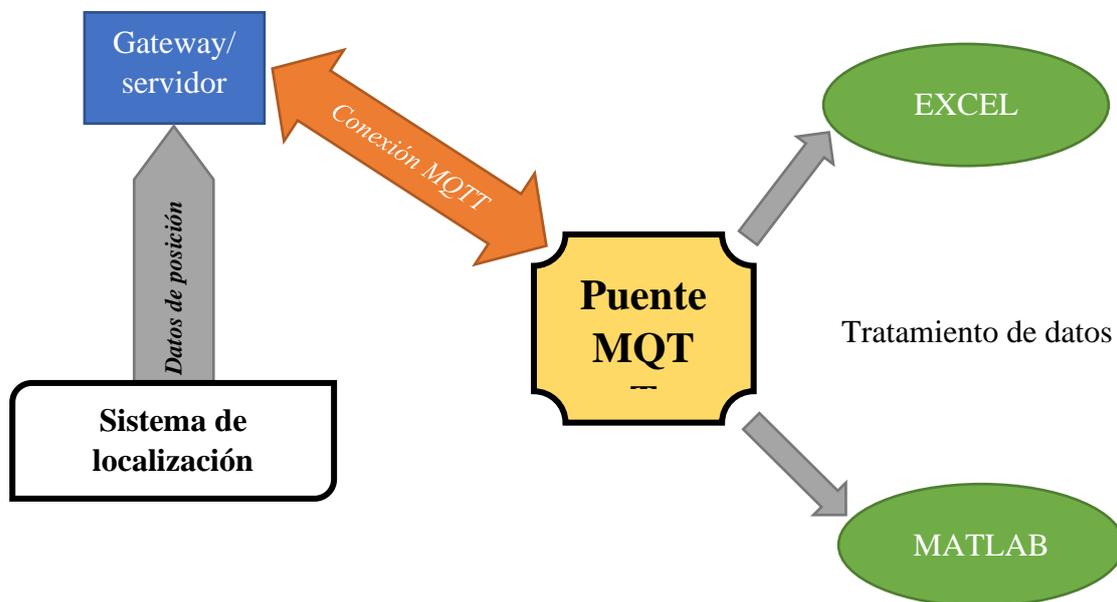


Figura 19: Esquema Software

Se puede ver, por tanto, que el proceso para la obtención de los datos comienza por el envío de las coordenadas al servidor desde el sistema de localización. Estas coordenadas se envían al puente mediante una conexión MQTT y desde allí se almacenarán en documentos de texto que serán tratados mediante las herramientas Excel y Matlab. Para el puente MQTT se utilizará el entorno Eclipse y el lenguaje Java (ver Anexo I).

5. Desarrollo del proyecto

5.1 Fase I

Para la primera fase del proyecto, se planificó la puesta en marcha y calibración del sistema instalado previamente en el laboratorio, para la comprobación del correcto funcionamiento de este utilizando las antenas más antiguas.

La primera acción, fue poner el sistema a punto mediante el uso de un plano del laboratorio actualizado y calibrando que las posiciones de las antenas en el plano y el laboratorio fueran las mismas. Además, se estableció el origen de coordenadas como veremos en la siguiente figura:

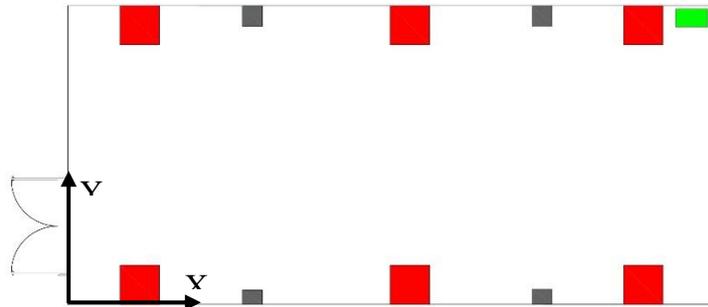


Figura 20: Coordenadas de origen plano actual de laboratorio.

Una vez el sistema era fiel a la realidad, se procedió a la calibración de los tags.

La calibración de los tags precisa de un lector NFC que permite, mediante el uso del software “Pozyx Device Configurator” disponible para Linux y Windows, la modificación de las características de nuestros tags.

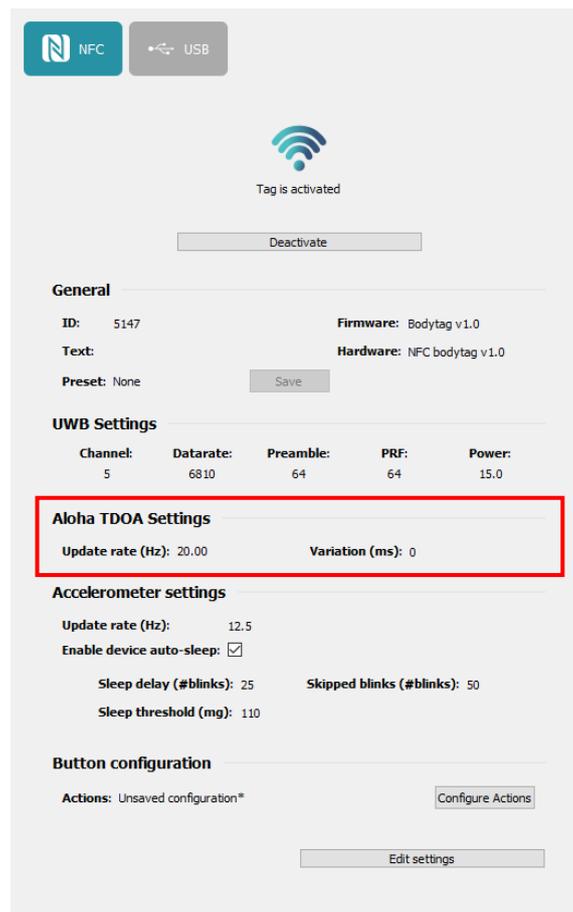


Figura 21: Pantalla de configuración de tags móviles.

Para configurar el tag, nos fijamos en el recuadro rojo y se modificó el “update rate (Hz)” a un valor adecuado, en nuestro caso, unas 20 pulsaciones por segundo son suficientes, siendo el máximo posible unas 23/24 pulsaciones.

Una vez se tienen los tags configurados como queremos, se debe configurar el entorno en el que trabajaremos. De esta manera, deberemos indicarle al sistema que trabajaremos con el método TDOA, ya que es el único disponible en este momento, como se ve en la figura 22.

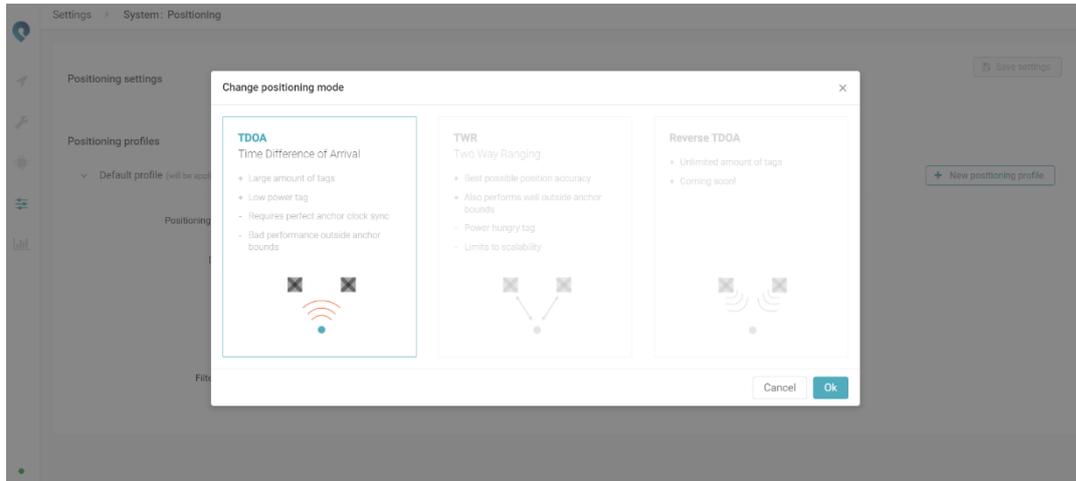


Figura 22: Configuración entorno TDOA.

Tras configurar los tags y el entorno de la forma que precisábamos, se procede a comprobar la precisión y fiabilidad del sistema.

Por ello, se colocan varios tags móviles por el laboratorio y se miden sus coordenadas con un medidor láser respecto al eje, para tener las coordenadas físicas exactas obteniendo las siguientes posiciones:

Nº TAG	COORDENADA X	COORDENADA Y
5125	6.952 m	1.287 m
5139	3.404 m	12.564 m
5147	6.840 m	11.229 m
5171	6.226 m	4.964 m
5173	7.030 m	13.457 m

Tabla 4: Identificadores tags utilizados fase I.

Estos tags representan a un operario del laboratorio que mantenga una posición fija, excepto en el caso del tag 5128 (el cual no aparece en la tabla), que se ha colocado en un robot móvil con una trayectoria conocida.

Antes de comenzar con la recopilación de datos del sistema de localización, se valoró la opción de filtrarlos utilizando un filtro de media, sin embargo, al revisar la aplicación del sistema se observó que existían varias opciones de pre filtrado, de tal forma que podríamos obtener los datos ya trabajados y evitando así añadir filtros de forma innecesaria.

En este caso, para el método TDOA, se observa que se disponen de los algoritmos UWB, el cual trabaja con la señal UWB sin tener en cuenta ningún tipo de predicción de

movimiento y el algoritmo de Kalman, el cual agrega un filtro de predicción de Kalman a las mediciones de UWB. Además, se puede elegir entre los siguientes filtros:

FILTROS	DESCRIPCIÓN
Outlier	Filtro de mediana
Full	Filtro que agrega un filtro exponencial doble al filtro de mediana.

Tabla 5: Filtros disponibles en el método TDOA.

Se realizaron varios estudios para probar la eficacia de estos filtros y elegir, así, el más adecuado para nuestro sistema y entorno. Conociendo las posiciones reales, se fueron recopilando grandes cantidades de datos de los tags fijos, obteniendo de esta manera, los valores que mide el sistema con cada filtro. Se calculó el módulo de cada medida respecto a las coordenadas reales y se hizo la media de dichos módulos. Se muestran, a continuación, los valores de las medias de los módulos para cada caso.

En el caso de elegir una localización solo de UWB con filtro de mediana (UWB Outlier), se recopilaron los resultados y se calcularon los siguientes módulos.

UWB OUTLIER	
Nº TAG	MEDIA MÓDULO
5125	0.26 m
5139	0.57 m
5147	0.36 m
5171	0.98 m
5173	0.47 m

Tabla 6: Media módulo. UWB Outlier fase I.

Utilizando una localización solo de UWB con filtro doble exponencial aplicado a un filtro de mediana (UWB Full), se recopilaron los resultados y se calcularon los siguientes módulos.

UWB FULL	
Nº TAG	MEDIA MÓDULO
5125	0.26 m
5139	0.61 m
5147	0.41 m
5171	1.07 m
5173	0.32 m

Tabla 7: Media módulo. UWB Full fase I.

En el caso de una localización de UWB con filtro de Kalman y filtro de mediana (Kalman Outlier), se recopilaron los resultados y se calcularon los siguientes módulos.

KALMAN OUTLIER

Nº TAG	MEDIA MÓDULO
5125	3.97 m
5139	0.45 m
5147	0.50 m
5171	0.48 m
5173	0.56 m

Tabla 8: Media módulo. Kalman Outlier fase I.

Si eligiésemos una localización de UWB con filtro de Kalman y filtro de doble exponencial aplicado a un filtro de mediana (Kalman Full), se recopilaron los resultados y se calcularon los siguientes módulos.

KALMAN FULL	
Nº TAG	MEDIA MÓDULO
5125	3.39 m
5139	0.44 m
5147	0.36 m
5171	0.54 m
5173	0.52 m

Tabla 9: Media módulo. Kalman Full fase I.

Sin embargo, todas estas medidas se han tomado aplicando el filtro en su máxima potencia (asumiendo que es la forma ideal de utilizarlo). Además, se ha querido comprobar la eficacia del filtro de mediana que viene por defecto, recopilando los datos en crudo (sin ningún filtro) y aplicándole de forma externa el filtro de mediana. De esta forma se han obtenido los siguientes resultados.

UWB CRUDO	
Nº TAG	MEDIA MÓDULO
5125	0.20 m
5139	0.61 m
5147	0.31 m
5171	0.94 m
5173	0.36 m

Tabla 10: Media módulo. UWB crudo fase I.

KALMAN CRUDO	
Nº TAG	MEDIA MÓDULO
5125	3.30 m
5139	0.42 m
5147	0.30 m
5171	0.27 m
5173	0.41 m

Tabla 11: Media módulo. Kalman crudo fase I.

Y una vez filtrados, obtenemos los valores de:

UWB CRUDO + FILTRO DE MEDIANA

Nº TAG	MEDIA MÓDULO
5125	0.11 m
5139	0.40 m
5147	0.27 m
5171	0.39 m
5173	0.22 m

Tabla 12: Media módulo. UWB crudo y filtro de mediana fase I.

KALMAN CRUDO + FILTRO DE MEDIANA	
Nº TAG	MEDIA MÓDULO
5125	3.36 m
5139	0.48 m
5147	0.39 m
5171	0.41 m
5173	0.37 m

Tabla 13: Media módulo. Kalman crudo y filtro de mediana fase I.

Observando los resultados y las posiciones de los tags en el plano (figura 23) vemos como los tags que presentan más problemas a la hora de localizarlos con mayor exactitud son los tags 5171 y 5125.

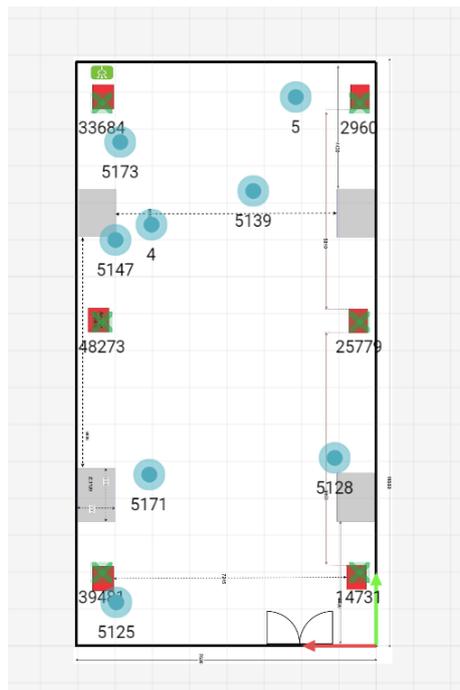


Figura 23:: Plano antiguo con tags en RT.

En el caso del 5125 podemos ver que es un tag que se encuentra fuera del rango de media de las antenas (las más cercanas son las antenas 14731 y 39481) y se sobreentiende que los errores de posición tan grandes entre la posición real y la obtenida mediante el sistema de localización se debe a eso. De hecho, aunque no se aprecia en la figura anterior, al observar el tag en la aplicación del servidor el punto perteneciente al tag siempre parpadeaba y cambiaba mucho de posición o incluso en ocasiones desaparecía como se ve en la siguiente figura:

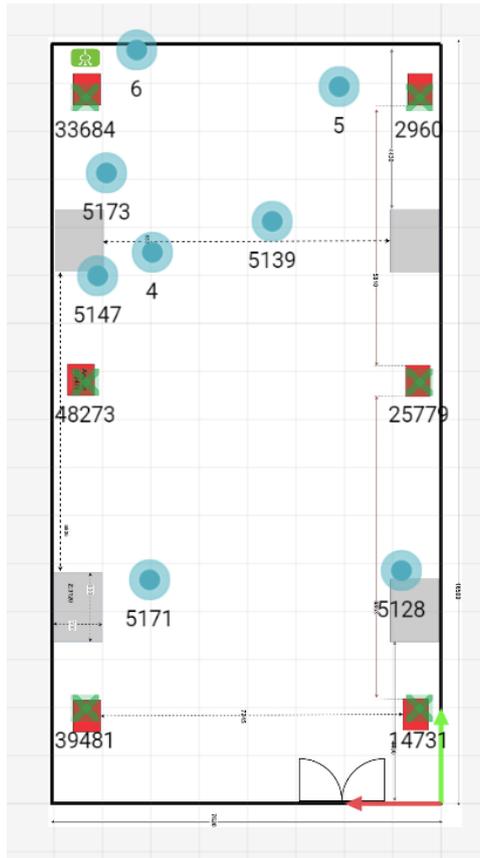


Figura 24: Plano antiguo con tags en RT (2).

Sin embargo, en el caso del tag 5171, aunque esta no se ve en el mapa, está colocado cerca de una planta piloto FESTO que, como podemos ver en la figura 25, está hecha de metal y, por tanto, impide el paso de la señal, crea interferencias con el sistema de localización y elimina precisión en las medidas.



Figura 25: Planta piloto FESTO.

Analizando el resto de los datos, vemos que lo óptimo en este caso es obtener los datos en UWB en crudo, es decir, sin ningún filtro aplicado y aplicarle nosotros de forma manual un filtro de mediana ya que, aunque el sistema seguiría sin ser lo suficientemente

preciso, se rebajaría levemente el error en comparación con los datos filtrados por el sistema de forma automática.

De esta forma se obtienen las siguientes gráficas para las posiciones de cada tag, en rojo la señal sin filtrar y en azul la filtrada.

TAG 5125

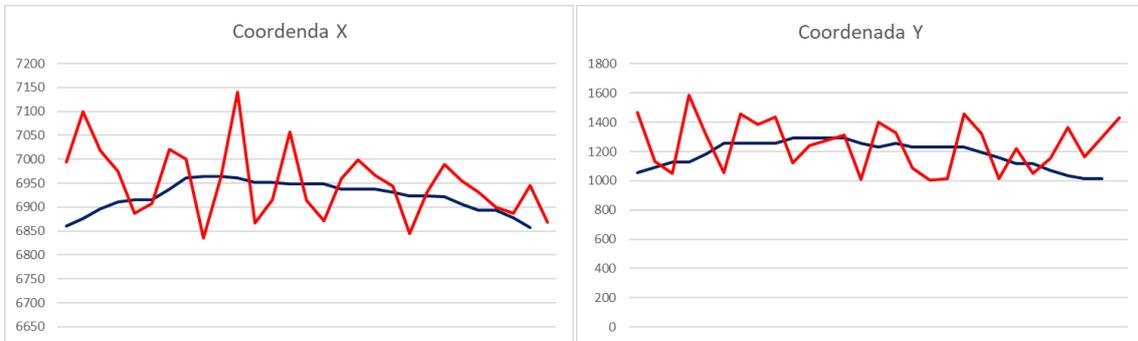


Gráfico 1: Coordenadas x e y - tag 5125 (antenas originales fase I).

TAG 5139

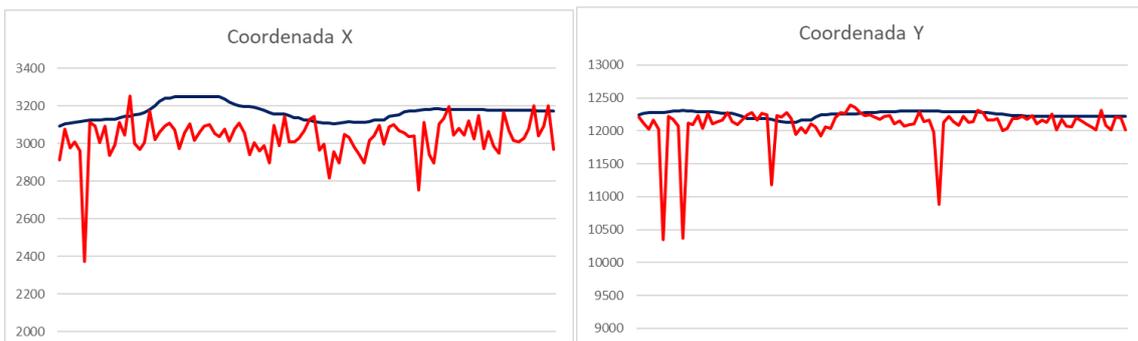


Gráfico 2: Coordenadas x e y - tag 5139 (antenas originales fase I).

TAG 5147

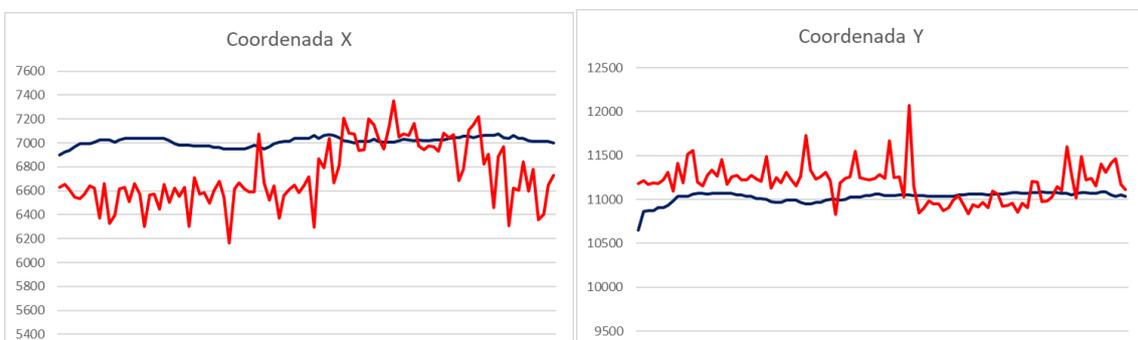


Gráfico 3: Coordenadas x e y - tag 5147 (antenas originales fase I).

TAG 5171

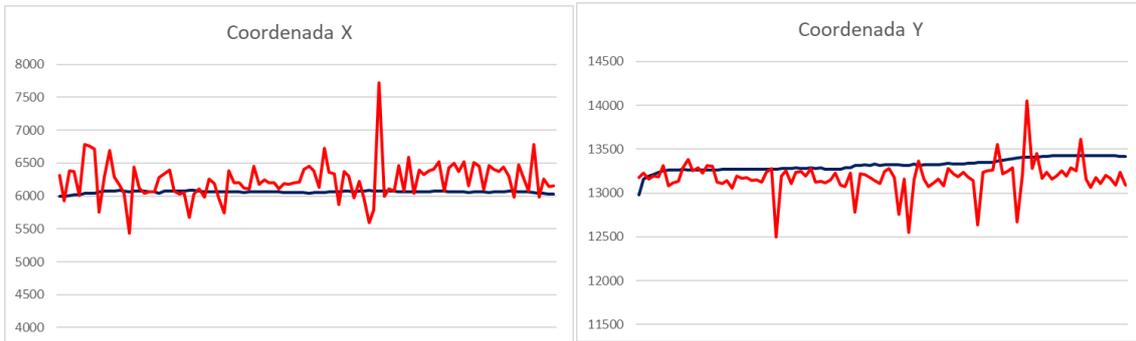


Gráfico 4: Coordenadas x e y - tag 5171 (antenas originales fase I).

TAG 5173

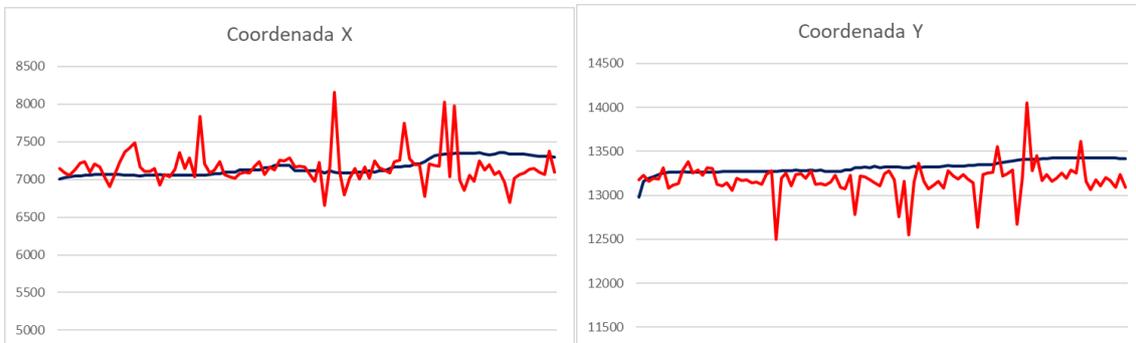


Gráfico 5: Coordenadas x e y - tag 5173 (antenas originales fase I).

Vemos, además, que en la representación del tag 5125, al estar fuera del rango, se recogen menos medidas que para el resto de los tags en la misma cantidad de tiempo.

Anteriormente, se comentaba que el tag 5128 se utilizaría para conocer la trayectoria del Robotino. Teniendo en cuenta que es un tag en movimiento, en vez de medir las posiciones fijas, tomaremos los datos de la trayectoria que sigue el robot móvil y la compararemos con la trayectoria que debe realizar. Para ello, se usarán todos los filtros obteniendo los siguientes recorridos sobre el plano. En rojo veremos la trayectoria a realizar, en verde el recorrido real que realiza el Robotino y en azul el recorrido que realiza según el sistema de localización.

TRAYECTORIA EN EL PLANO, KALMAN FULL

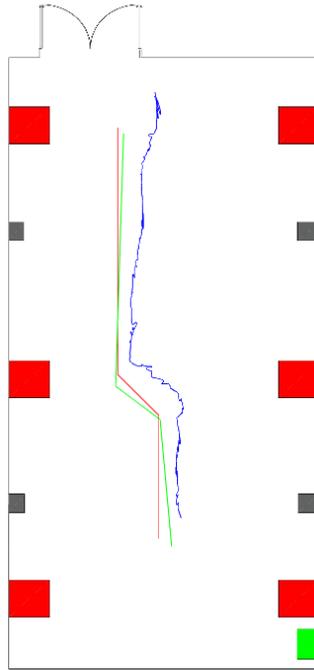


Figura 26: Trayectoria tag 5128 – Kalman full (antenas originales fase I).

TRAYECTORIA EN EL PLANO, KALMAN OUTLIER

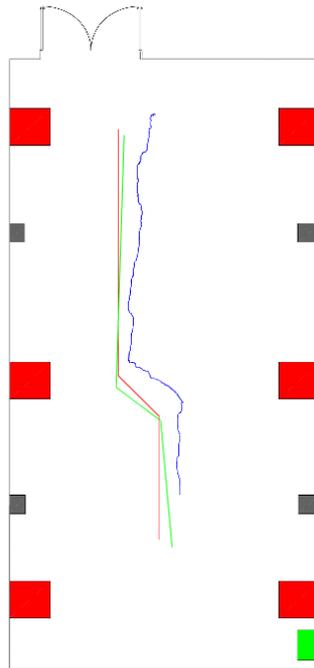


Figura 27: Trayectoria tag 5128 – Kalman Outlier (antenas originales fase I).

TRAYECTORIA EN EL PLANO, UWB FULL

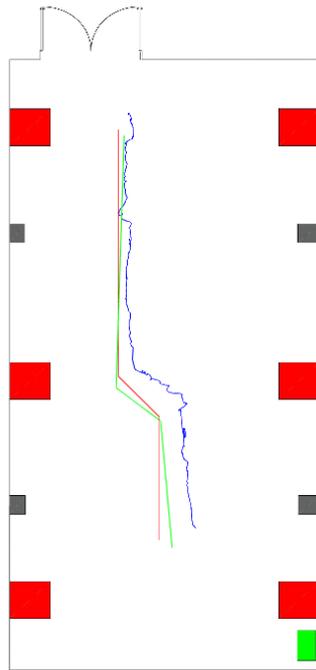


Figura 28: Trayectoria tag 5128 – UWB full (antenas originales fase I).

TRAYECTORIA EN EL PLANO, UWB OUTLIER

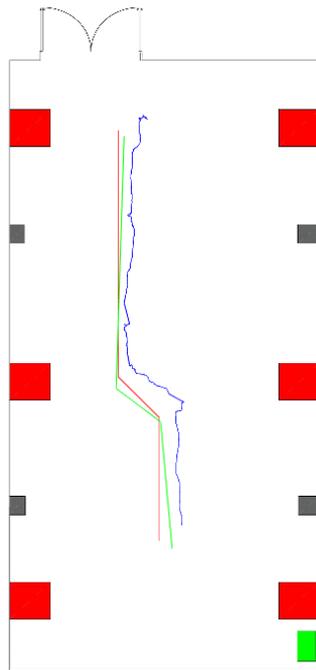


Figura 29: Trayectoria tag 5128 – UWB Outlier (antenas originales fase I).

También, se han obtenido los valores en crudo y se les ha aplicado un filtro de mediana, para comprobar en el caso de los tags en movimiento, qué tipo de medida se ajusta más.

TRAYECTORIA EN EL PLANO, KALMAN CRUDO

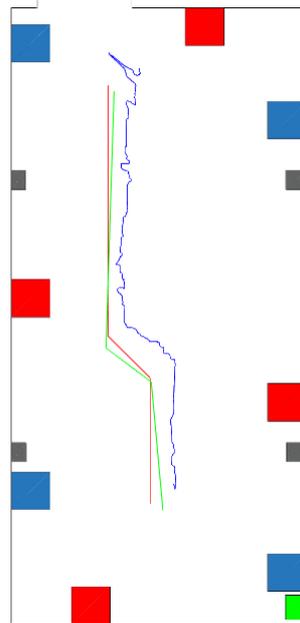


Figura 30: Trayectoria tag 5128 – Kalman crudo con filtro de mediana matlab (antenas originales fase I).

TRAYECTORIA EN EL PLANO, UWB CRUDO

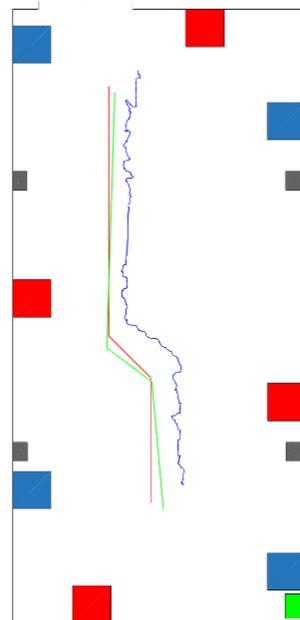


Figura 31: Trayectoria tag 5128 - UWB crudo con filtro de mediana matlab (antenas originales fase I).

Si nos fijamos, veremos que todas las representaciones tienen algo en común, y es que en la parte inferior la trayectoria que indica el sistema y la real no coinciden o se acercan en ningún caso. Al igual que pasaba con el tag 5171, se debe a la planta piloto FESTO situada cerca de esa zona. Por tanto, la mejor aproximación se realiza con el filtro UWB Outlier,

asumiendo que, en casos de movimiento, a diferencia de los casos estáticos, es mejor obtener los valores con un filtro de mediana previo.

5.2 Fase II

Para la segunda fase del proyecto, se realiza un cambio de antenas, sustituyendo las originales por su versión más actualizada (véase la descripción del hardware). Así pues, utilizando una disposición de 8 antenas frente a las 6 anteriores, se realiza un procedimiento igual al de la fase I para el método TDOA.

Para este caso, se repitieron los ensayos de la fase I y se comprueban los resultados obtenidos. Para este caso, el sistema de coordenadas ha cambiado, al cambiar la disposición de las antenas y de los tags, se ha planteado que el sistema de coordenadas se coloque en la esquina del plano donde se encuentra en servidor:

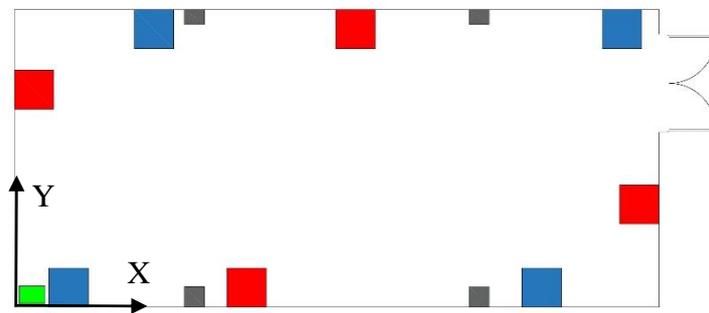


Figura 32: Coordenadas de origen plano actual de laboratorio.

Por ello, se colocan varios tags móviles por el laboratorio y se miden sus coordenadas con un medidor láser respecto al eje, para tener las coordenadas físicas exactas obteniendo las siguientes posiciones:

Nº TAG	COORDENADA X	COORDENADA Y
5125	5.324 m	1.915 m
5139	15.132 m	1.527 m
5147	11.435 m	1.610 m
5171	3.833 m	4.760 m
5173	2.678 m	0.766 m

Tabla 14: Identificadores tags utilizados fase II.

Estos tags representan, de nuevo, a un operario del laboratorio que mantenga una posición fija excepto, de nuevo, el tag 5128, que se ha colocado en el Robotino para evaluar su trayectoria.

Se realizaron los mismos estudios que en la fase anterior para probar la eficacia de estos filtros y elegir, así, el más adecuado para nuestro sistema y entorno. Conociendo las posiciones reales, se fueron recopilando datos de los tags fijos, obteniendo de esta manera, los valores que mide el sistema con cada filtro. Se calculó el módulo de cada medida respecto a las coordenadas reales y se hizo la media de dichos módulos. Se muestran, a continuación, los valores de las medias de los módulos para cada caso.

En el caso de elegir una localización solo de UWB con filtro de media (UWB Outlier), se recopilaron los resultados y se calcularon los siguientes módulos.

UWB OUTLIER	
Nº TAG	MEDIA MÓDULO
5125	0.39 m
5139	0.25 m
5147	0.07 m
5171	0.11 m
5173	0.18 m

Tabla 15: Media módulo. UWB Outlier fase II.

Utilizando una localización solo de UWB con filtro doble exponencial aplicado a un filtro de mediana (UWB Full), se recopilaron los resultados y se calcularon los siguientes módulos.

UWB FULL	
Nº TAG	MEDIA MÓDULO
5125	0.50 m
5139	0.23 m
5147	0.08 m
5171	0.11 m
5173	0.21 m

Tabla 16: Media módulo. UWB Full fase II.

En el caso de una localización de UWB con filtro de Kalman y filtro de mediana (Kalman Outlier), se recopilaron los resultados y se calcularon los siguientes módulos.

KALMAN OUTLIER	
Nº TAG	MEDIA MÓDULO
5125	0.37 m
5139	0.23 m
5147	0.09 m
5171	0.12 m
5173	0.25 m

Tabla 17: Media módulo. Kalman Outlier fase II.

Si eligiésemos una localización de UWB con filtro de Kalman y filtro de doble exponencial aplicado a un filtro de mediana (Kalman Full), se recopilaron los resultados y se calcularon los siguientes módulos.

KALMAN FULL	
Nº TAG	MEDIA MÓDULO
5125	0.38 m
5139	0.24 m
5147	0.11 m
5171	0.13 m
5173	0.26 m

Tabla 18: Media módulo. Kalman Full fase II.

Sin embargo, todas estas medidas se han tomado aplicando el filtro en su máximo potencia (asumiendo que es la forma ideal de utilizarlo). Además, se ha querido probar los datos en crudo (sin ningún filtro) y aplicándole de forma externa el filtro de mediana. De esta forma se han obtenido los siguientes resultados.

UWB CRUDO	
Nº TAG	MEDIA MÓDULO
5125	0.40 m
5139	0.27 m
5147	0.11 m
5171	0.10 m
5173	0.17 m

Tabla 19: Media módulo. UWB crudo fase II.

KALMAN CRUDO	
Nº TAG	MEDIA MÓDULO
5125	0.38 m
5139	0.26 m
5147	0.11 m
5171	0.13 m
5173	0.31 m

Tabla 20: Media módulo. Kalman crudo fase II.

Y una vez filtrados, obtenemos los valores de:

UWB CRUDO + FILTRO DE MEDIANA	
Nº TAG	MEDIA MÓDULO
5125	0.39 m
5139	0.25 m
5147	0.07 m
5171	0.16 m
5173	0.15 m

Tabla 21: Media módulo. UWB crudo y filtro de mediana fase II.

KALMAN CRUDO + FILTRO DE MEDIANA	
Nº TAG	MEDIA MÓDULO
5125	0.37 m
5139	0.26 m
5147	0.10 m
5171	0.19 m
5173	0.28 m

Tabla 22: Media módulo. Kalman crudo y filtro de mediana fase II.

Vemos que efectivamente, en este caso, aunque el sistema de coordenadas ha cambiado respecto a la fase anterior y la distribución de los tags es diferente, volvemos a tener dos tags que en el mejor de los casos poseen un error mayor a los 20 centímetros. Si comparamos todos los resultados vemos que el uso de un filtro de media aplicado previamente a la recopilación de datos es más útil que el resto de los filtros o el filtrado manual.

Si viéramos una representación de los tags en el plano veríamos que, de nuevo, los tags con mayor error en la medida son los que se encuentran cerca de las zonas de la planta piloto FESTO (en este caso son el 5125 y el 5139) que, aunque se hayan eliminado las

zonas muertas con la implementación de dos antenas más, siguen siendo zonas problemáticas cerca de mucho metal.

Así pues, en el caso de los tags fijos, nos decantamos en esta fase por el uso de un filtro UWB Outlier, ya que en general los valores en las medidas no son tan grandes como en otros casos.

De esta forma se obtienen las siguientes gráficas para las coordenadas obtenidas de cada tag, en rojo la señal sin filtrar y en azul la posición real.

TAG 5125

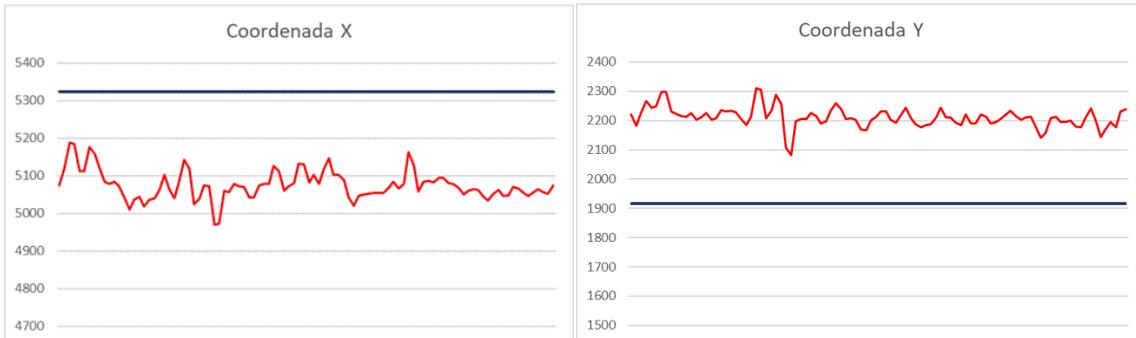


Gráfico 6: Coordenadas x e y - tag 5125 (antenas originales fase II).

TAG 5139

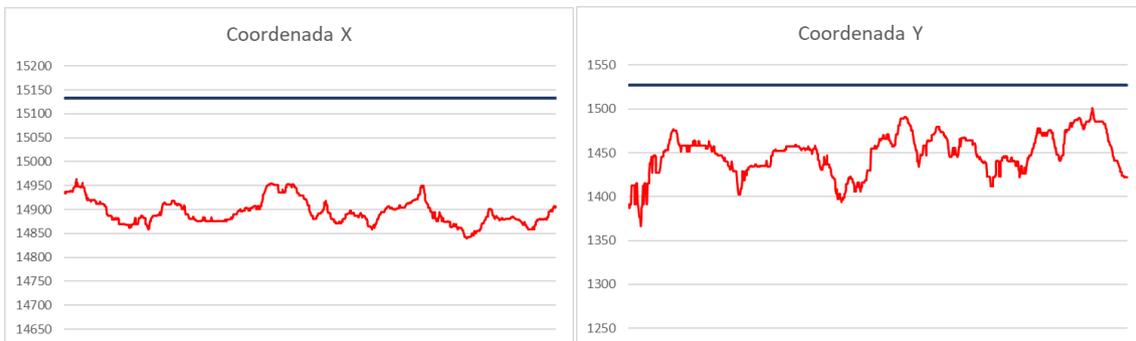


Gráfico 7: Coordenadas x e y - tag 5139 (antenas originales fase II).

TAG 5147

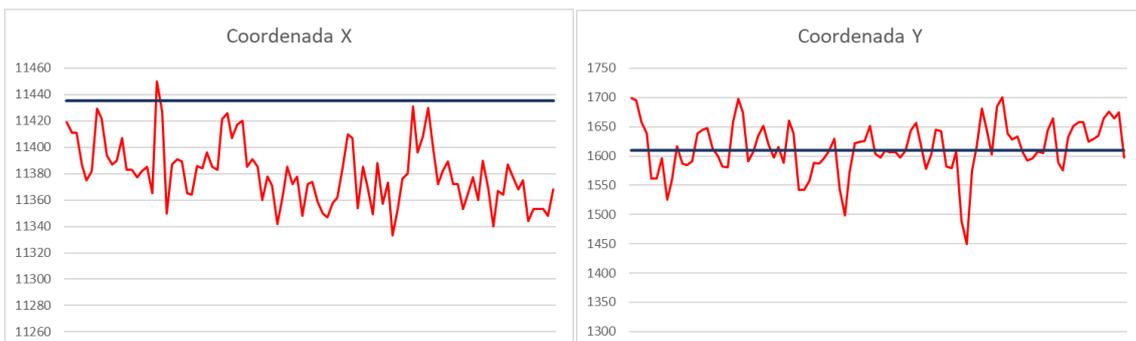


Gráfico 8: Coordenadas x e y - tag 5147 (antenas originales fase II).

TAG 5171

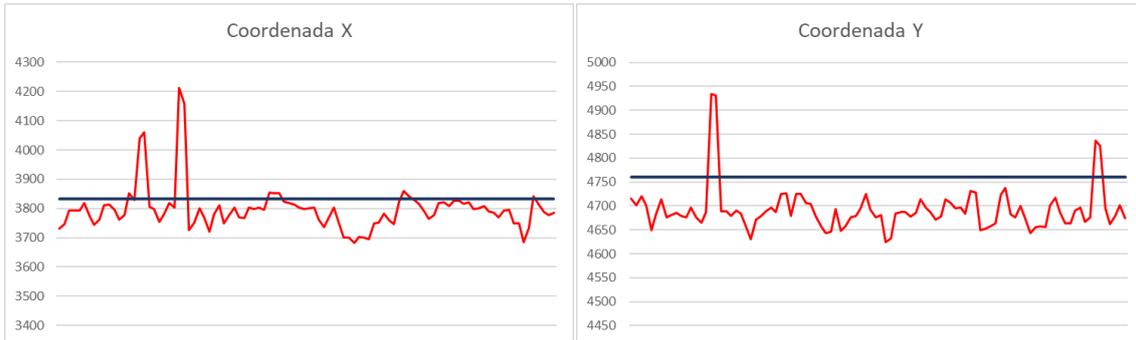


Gráfico 9: Coordenadas x e y - tag 5171 (antenas originales fase II).

TAG 5173

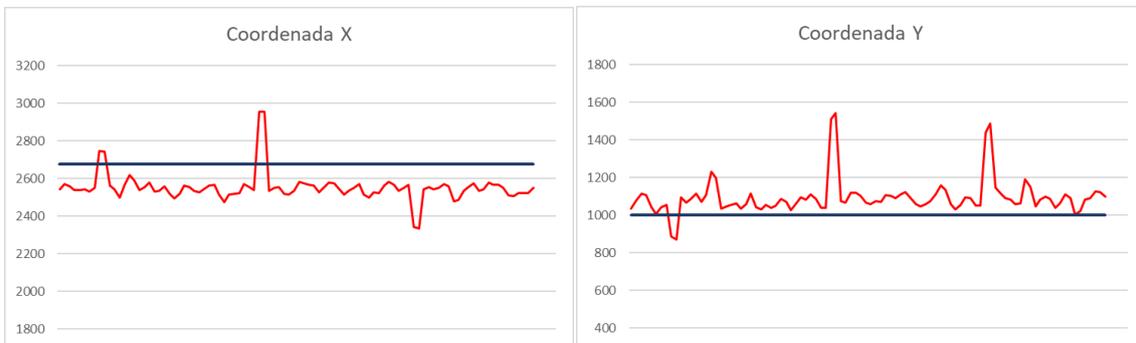


Gráfico 10: Coordenadas x e y - tag 5173 (antenas originales fase II).

Al ver estas gráficas, podemos apreciar que los tags en los que, como ya hemos comentado, se obtienen mayor error, las gráficas indican unos datos son mucho más irregulares.

Veamos, sin embargo, que sucede cuando el tag está en movimiento. Utilizando el mismo trayecto que en la fase I, veamos gráficamente los resultados que se obtienen con el nuevo hardware.

Teniendo en cuenta que el tag 5128 es un tag en movimiento, en vez de medir las posiciones fijas, tomaremos los datos de la trayectoria que sigue el robot móvil y la compararemos con la trayectoria que debe realizar. Para ello, se usarán todos los filtros obteniendo los siguientes recorridos sobre el plano. En rojo veremos la trayectoria a realizar, en verde el recorrido real que realiza el Robotino y en azul el recorrido que realiza según el sistema de localización.

TRAYECTORIA EN EL PLANO, KALMAN FULL

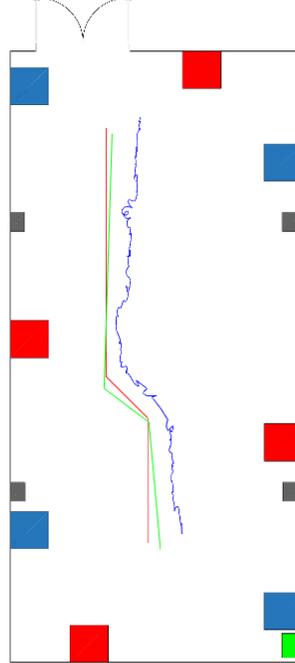


Figura 33: Trayectoria tag 5128 – Kalman full (antenas originales fase II).

TRAYECTORIA EN EL PLANO, KALMAN OUTLIER

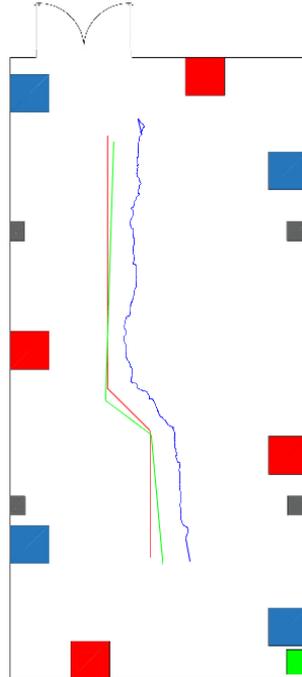


Figura 34: Trayectoria tag 5128 – Kalman Outlier (antenas originales fase II).

TRAYECTORIA EN EL PLANO, UWB FULL

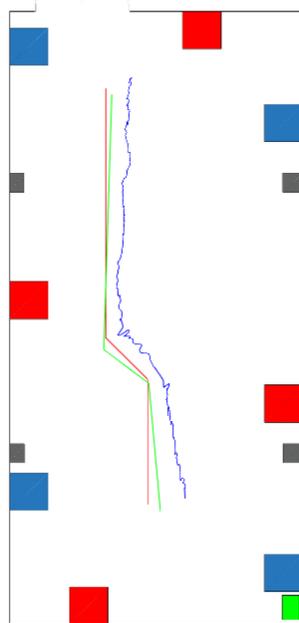


Figura 35: Trayectoria tag 5128 – UWB full (antenas originales fase II).

TRAYECTORIA EN EL PLANO, UWB OUTLIER

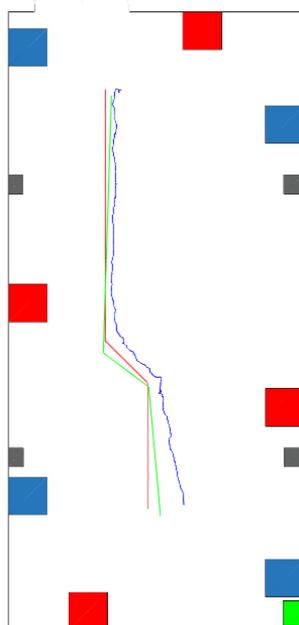


Figura 36: Trayectoria tag 5128 – UWB Outlier (antenas originales fase II).

También, se han obtenido los valores en crudo y se les ha aplicado un filtro de mediana, para comprobar en el caso de los tags en movimiento, qué tipo de medida se ajusta más.

TRAYECTORIA EN EL PLANO, KALMAN CRUDO

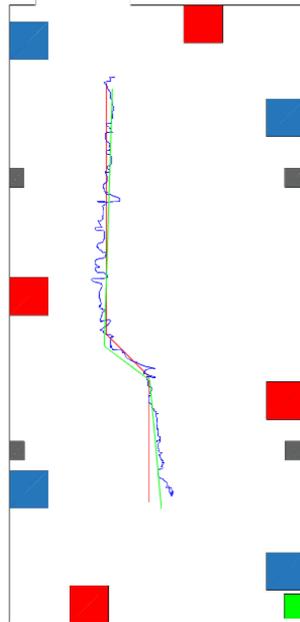


Figura 37: Trayectoria tag 5128 – Kalman crudo con filtro de mediana matlab (antenas originales fase II).

TRAYECTORIA EN EL PLANO, UWB CRUDO

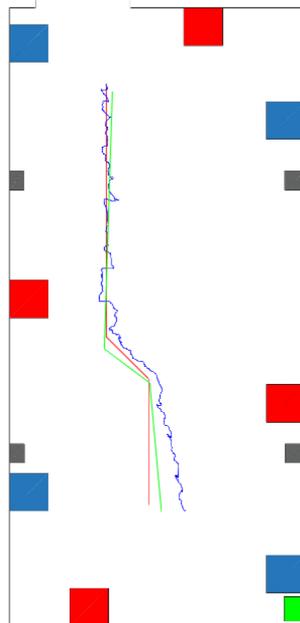


Figura 38: Trayectoria tag 5128 – UWB crudo con filtro de mediana matlab (antenas originales fase II).

En este caso, y contra todo pronóstico, viendo los resultados podemos decir que el mejor método cuando tenemos un tag en movimiento es el uso de un filtro de Kalman al que se le añade de forma externa al sistema un filtro de mediana.

Este resultado difiere bastante de los resultados de la fase I donde los resultados indicaban que el UWB con filtro de media era el más adecuado y que el método de filtrado de Kalman no era muy fiable, ya que era el que mayor error demostraba.

5.3 Fase III

En esta última fase del proyecto, se tratará de realizar el estudio cambiando el método TDOA por el método TWR, que según la teoría vista en el segundo punto de la memoria debería ser bastante más fiable en entornos industriales con mucho metal u obstáculos.

Para ello, utilizaremos el mismo sistema de referencia que en el apartado anterior (figura x) y cambiaremos los tags por los tags de desarrollos (todos ellos descritos en el séptimo punto de la memoria).

Al cambiar los tags por otros diferentes, se deben calibrar de forma correcta. La calibración de los tags precisa de un cable microUSB que permite, mediante el uso del software “Pozyx Device Configurator” disponible para Linux y Windows, la modificación de las características de nuestros tags. En la figura 39 podemos ver una imagen de ejemplo:

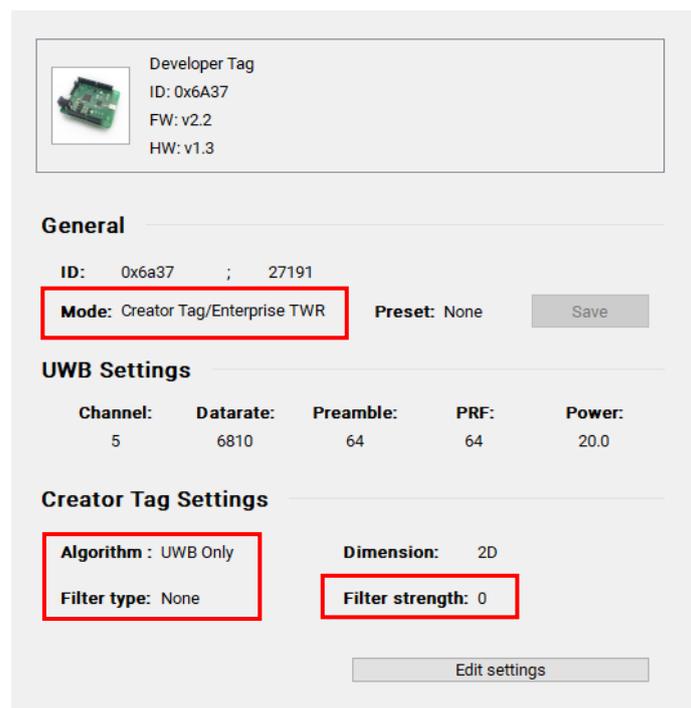


Figura 39: Pantalla de configuración de tags de desarrollo.

Lo primero que se debe modificar es el apartado “General”, concretamente, colocar el modo del tag en TWR, ya que es el método que queremos probar. Posteriormente, habría que seleccionar un algoritmo y un filtro en el apartado de “Creator Tag Settings”, teniendo en cuenta que, en estos ensayos al igual que en las fases anteriores, se probarán los filtros al máximo, ya que se espera obtener mejores resultados de esta manera.

Una vez se tienen los tags configurados como queremos, se debe configurar el entorno en el que trabajaremos. De esta manera, deberemos indicarle al sistema que trabajaremos con el método TWR como se ve en la figura 40.

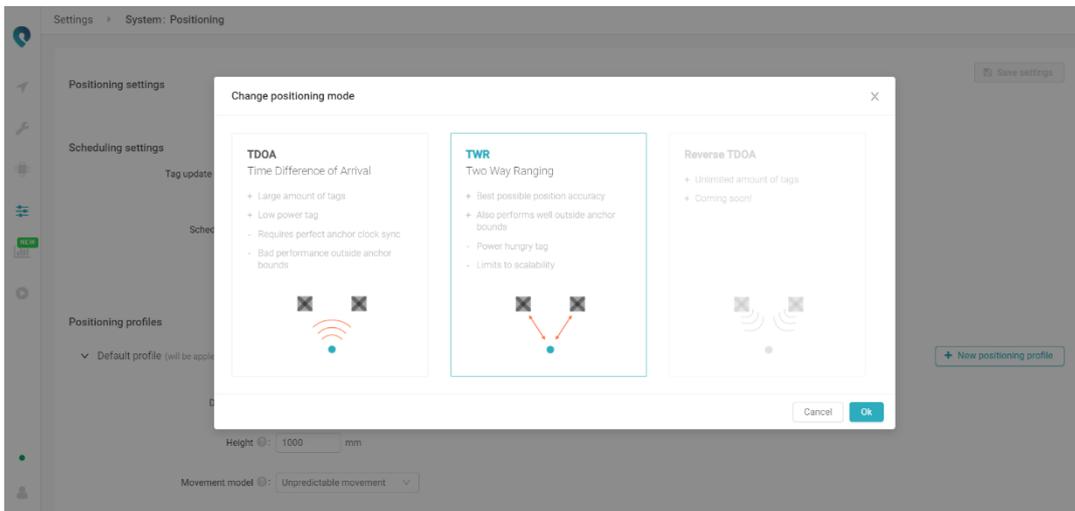


Figura 40: Configuración entorno TWR.

Además, se precisa de la incorporación manual de los tags de desarrollo que se quieren controlar en el visualizador, de tal forma que hay que añadirlos en la lista como se aprecia en la siguiente figura.

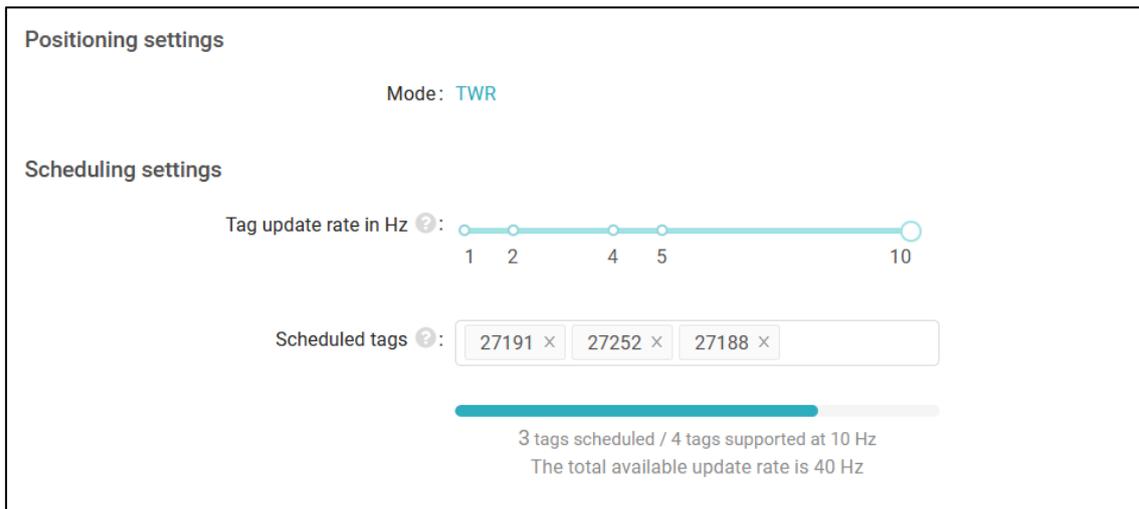


Figura 41: Configuración control tags.

Tras configurar los tags y el entorno de la forma que precisábamos, se procede a comprobar la precisión y fiabilidad del sistema.

Por ello, se colocan dos tags de desarrollo por el laboratorio y se miden sus coordenadas con un medidor láser respecto al eje de coordenadas, para tener las posiciones físicas exactas. En este caso, para comprobar su eficacia, se han colocado los tags en zonas que anteriormente resultaban conflictivas.

El tag 27188 se ha colocado en un extremo del laboratorio casi al límite de la zona útil y el tag 27525, cerca de la planta piloto FESTO para comprobar si con este método se

solventa el problema del metal. Se obtienen, de esta forma, las siguientes posiciones para los tags:

Nº TAG	COORDENADA X	COORDENADA Y
27188	1.364 m	2.231 m
27252	11.517 m	1.601 m

Tabla 23: Identificadores tags utilizados fase III.

Además de estos tags que se dejaron en posiciones fijas, se utilizará otro, el 27191 para analizar el uso de los tags de desarrollo usando el método TWR en el caso de seguir la trayectoria de objetos móviles, concretamente, el Robotino.

Para este método (TWR), se observa que se disponen al igual que antes, de los algoritmos UWB, el cual trabaja con la señal UWB sin tener en cuenta ningún tipo de predicción de movimiento y el algoritmo de Kalman, el cual agrega un filtro de predicción de Kalman a las mediciones de UWB. Además, se puede elegir entre los siguientes filtros:

FILTROS	DESCRIPCIÓN
Moving Average	Filtro de media móvil, basado en promedios.
Moving Median	Filtro de mediana.
FIR	Filtro de Respuesta finita al impulso.

Tabla 24: Filtros disponibles en el método TWR.

Utilizando estos filtros y realizando todas las combinaciones posibles, se hicieron varios estudios para probar la eficacia de estos y elegir, así, el más adecuado para nuestro sistema y entorno. Conociendo las posiciones reales, se fueron recopilando datos de los tags fijos, obteniendo de esta manera, los valores que mide el sistema con cada filtro. Se calculó el módulo de cada medida respecto a las coordenadas reales y se hizo la media de dichos módulos. Se muestran, a continuación, los valores de las medias de los módulos para cada caso.

UWB FIR	
Nº TAG	MEDIA MÓDULO
27188	0.09 m
27252	0.10 m

Tabla 25: Media módulo. UWB FIR.

UWB MOVING AVERAGE	
Nº TAG	MEDIA MÓDULO
27188	0.10 m
27252	0.10 m

Tabla 26: Media módulo. UWB Moving Average.

UWB MOVING MEDIAN	
Nº TAG	MEDIA MÓDULO
27188	0.10 m
27252	0.13 m

Tabla 27: Media módulo. UWB Moving Median.

Veamos ahora qué valores se obtienen cuando recopilamos los datos sin ningún tipo de filtro aplicado y comprobemos si añadir el filtro de Moving Median de forma manual, mejora los datos o no con respecto a los filtros ofrecidos en el sistema.

UWB CRUDO	
Nº TAG	MEDIA MÓDULO
27188	0.10 m
27252	0.11 m

Tabla 28: Media módulo. UWB crudo Fase III.

UWB CRUDO + FILTRO MOVING MEDIAN	
Nº TAG	MEDIA MÓDULO
27188	0.10 m
27252	0.10 m

Tabla 29: Media módulo. UWB crudo con filtro de mediana móvil.

Sin embargo, se presentan a continuación, los datos para el caso en que se añade el algoritmo de Kalman a la señal UWB:

KALMAN FIR	
Nº TAG	MEDIA MÓDULO
27188	0.09 m
27252	0.10 m

Tabla 30: Media módulo. Kalman FIR.

KALMAN MOVING AVERAGE	
Nº TAG	MEDIA MÓDULO
27188	0.12 m
27252	0.12 m

Tabla 31: Media módulo. Kalman Moving Average.

KALMAN MOVING MEDIAN	
Nº TAG	MEDIA MÓDULO
27188	0.13 m
27252	0.12 m

Tabla 32: Media módulo. Kalman Moving Median.

Veamos ahora qué valores se obtienen cuando recopilamos los datos sin ningún tipo de filtro aplicado y comprobemos si añadir el filtro de Moving Median de forma manual, mejora los datos o no con respecto a los filtros ofrecidos en el sistema.

KALMAN CRUDO	
Nº TAG	MEDIA MÓDULO
27188	0.11 m
27252	0.06 m

Tabla 33: Media módulo. Kalman crudo Fase III.

KALMAN CRUDO + FILTRO MOVING MEDIAN	
Nº TAG	MEDIA MÓDULO
27188	0.10
27252	0.02

Tabla 34: Media módulo. Kalman crudo con filtro de mediana móvil.

Viendo estos resultados, podemos decir que para el tag de desarrollo 27188, los valores son aproximadamente los mismos para cualquiera de las opciones que escojamos. Sin embargo, para el tag 27252 si escogemos los datos en crudo y le añadimos de una forma externa al sistema un filtro de mediana móvil (Moving Median) podemos ver como obtenemos unos resultados casi perfectos que nos devuelven valores de error de 2cm.

Si nos fijamos en el plano, vemos como el tag que está en el extremo de la zona comprendida por las antenas, sigue perdiendo eficacia con respecto a otras áreas. Esto ya ocurría en las fases anteriores, donde se apreciaba que los tags más cerca de los límites perdían eficacia y notificaban un error mayor.

Por otro lado, obtenemos muy buenos resultados con el tag 27252, que como se comentaba al principio, se encontraba en una buena posición, pero cerca de la planta piloto FESTO, lo cual anteriormente generaba problemas y aumentaba los errores de manera grave. En este caso, los resultados implican un posicionamiento casi perfecto, dejando mayores errores en la periferia del área comprendida por las antenas.

De esta forma se obtienen las siguientes gráficas para las posiciones de cada tag, en rojo la señal sin filtrar y en azul la posición real.

TAG 27188

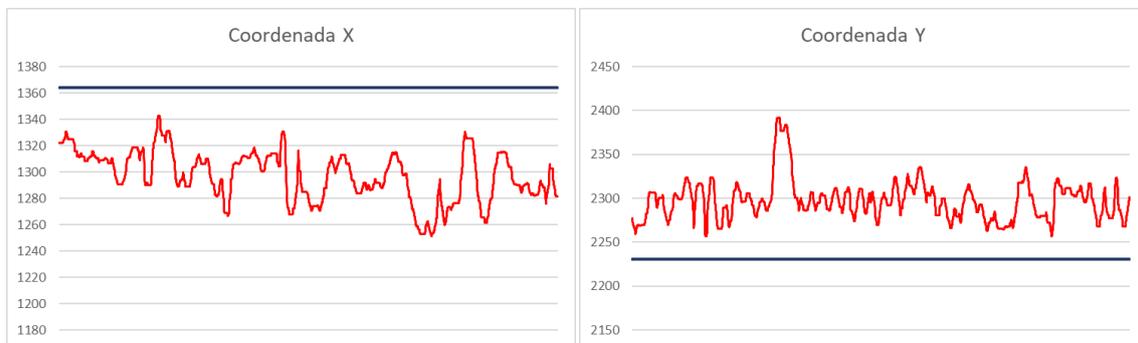


Gráfico 11: Coordenadas x e y - tag 27188.

TAG 27252

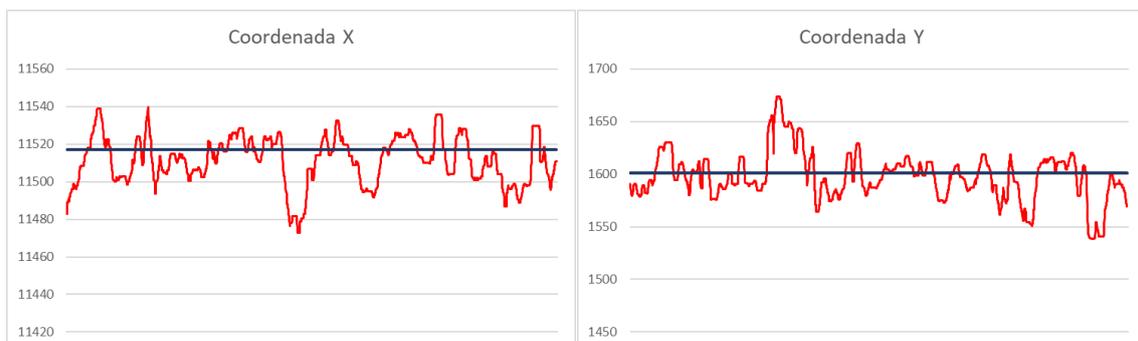


Gráfico 12: Coordenadas x e y - tag 27252.

Veamos entonces qué sucede si utilizamos el tag de desarrollo con el Robotino y le marcamos la misma trayectoria utilizada anteriormente.

Teniendo en cuenta que el tag 27191 es un tag en movimiento, en vez de medir las posiciones fijas, tomaremos los datos de la trayectoria que sigue el robot móvil y la compararemos con la trayectoria que debe realizar. Para ello, se usarán todos los filtros obteniendo los siguientes recorridos sobre el plano. En rojo veremos la trayectoria a realizar, en verde el recorrido real que realiza el Robotino y en azul el recorrido que realiza según el sistema de localización.

TRAYECTORIA EN EL PLANO, KALMAN MOVING MEDIAN

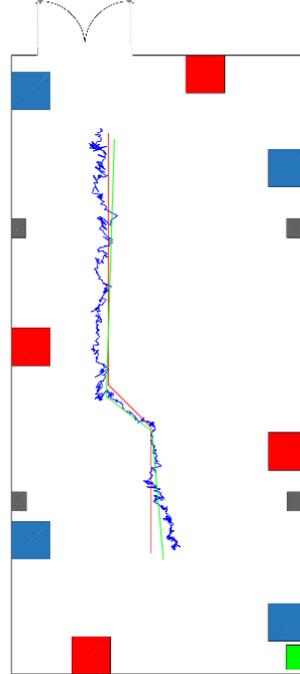


Figura 42: Trayectoria tag 27191 – Kalman Moving Median.

TRAYECTORIA EN EL PLANO, KALMAN FIR

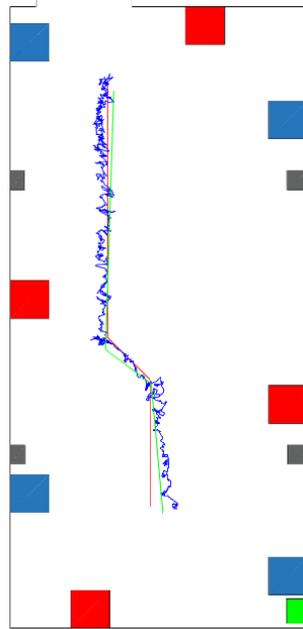


Figura 43: Trayectoria tag 27191 – Kalman FIR.

TRAYECTORIA EN EL PLANO, KALMAN MOVING AVERAGE

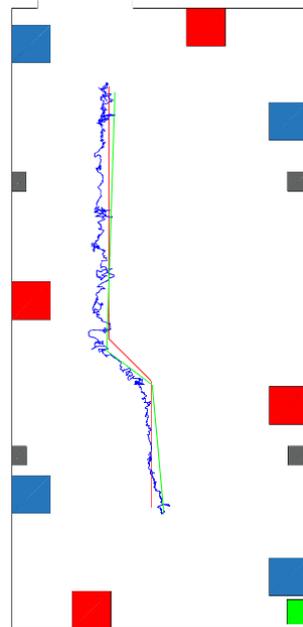


Figura 44: Trayectoria tag 27191 – Kalman Moving Average.

TRAYECTORIA EN EL PLANO, UWB MOVING MEDIAN

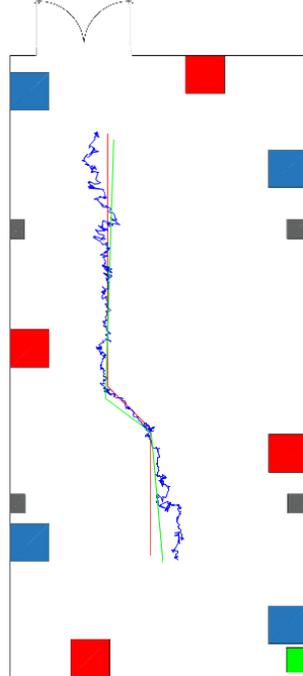


Figura 45: Trayectoria tag 27191 – Kalman Moving Median.

TRAYECTORIA EN EL PLANO, UWB FIR

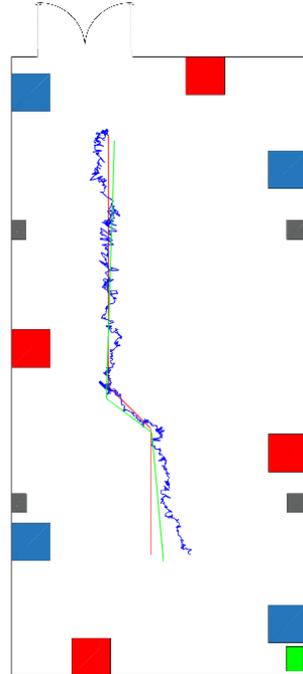


Figura 46: Trayectoria tag 27191 – Kalman Moving FIR.

TRAYECTORIA EN EL PLANO, UWB MOVING AVERAGE

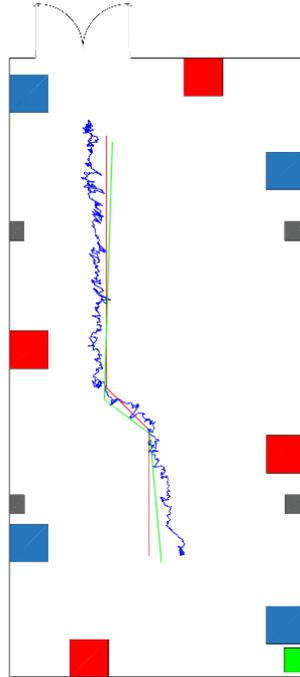


Figura 47: Trayectoria tag 27191 – Kalman Moving Average.

También, se han obtenido los valores en crudo y se les ha aplicado un filtro de mediana, para comprobar en el caso de los tags en movimiento, qué tipo de medida se ajusta más.

TRAYECTORIA EN EL PLANO, KALMAN CRUDO

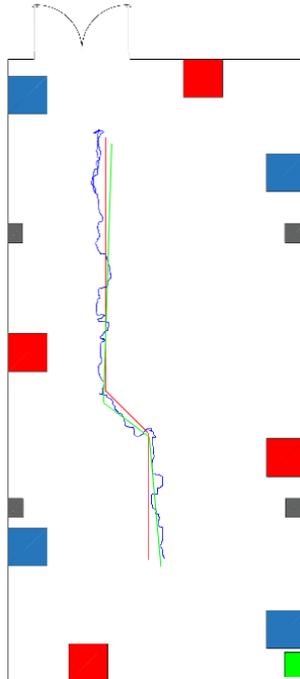


Figura 48: Trayectoria tag 27191 – Kalman crudo con filtro de mediana móvil Matlab.

TRAYECTORIA EN EL PLANO, UWB CRUDO

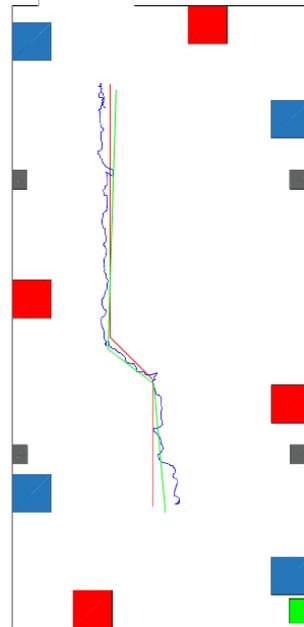


Figura 49: Trayectoria tag 27191 – Kalman crudo con filtro de mediana móvil Matlab.

Así pues, vemos en las gráficas que al igual que en el estudio de los tags fijos, el mejor método para ver la posición con el método TWR es la recopilación de los datos mediante el algoritmo de Kalman y aplicándole posteriormente el filtro de mediana móvil (Moving median).

6. Conclusiones y líneas abiertas

En este apartado se puede ver una comparativa de los datos recopilados en las fases anteriores y un desglose los resultados para obtener una visión más global de ellos.

FASE	Método	Filtro tags fijos	Filtro tags móviles	Rango de error
I	TDOA	UWB + filtro mediana	UWB Outlier	10-40 cm
II	TDOA	UWB Outlier	Kalman + filtro mediana	7-40 cm
III	TWR	Kalman + filtro mediana móvil	Kalman + filtro mediana móvil	2-10 cm

Tabla 35: Resumen fases del proyecto.

Si se observa la tabla anterior podemos hablar de ventajas y desventajas del sistema dependiendo de la fase en la que nos encontremos.

La fase inicial, teniendo en cuenta los resultados obtenidos, no presenta una mayor ventaja que la de tener un sistema de localización instalado con el que empezar a trabajar. Sus resultados no son fiables ni precisos y no es óptimo para seguimiento de trayectorias ni localización de activos ya que, aunque en el mejor caso nos ubica algunos con un error de entre 10 y 40 centímetros, existe la posibilidad de que si no se elige el método adecuado haya errores de hasta 3 metros en tags fijos. Sin embargo, cuando se trata de tags móviles es un poco más eficaz.

En cuanto a la segunda fase, el cambio de antenas y la nueva distribución mejora levemente los resultados en los tags móviles. En los tags fijos no se aprecia demasiada mejoría, ya que se siguen teniendo, en el mejor de los casos, un error de entre 7 y 10 centímetros. En este caso, obtenemos valores más homogéneos en lo que respecta a las zonas de medida. Se sigue observando que los peores valores se obtienen en condiciones muy específicas: las zonas más extremas del área comprendida por las antenas y las zonas cerca de grandes estructuras de metal o columnas.

Por último, la tercera fase mejora considerablemente todos los resultados obtenidos hasta ahora. En este caso, los tags fijos consiguen un error mínimo hasta el momento de entre 2 y 10 centímetros. Aunque no se han podido probar más tags debido a la cantidad disponible en el laboratorio, se han colocado en lugares que hasta entonces eran conflictivos (cerca de metal y en los extremos del área) y se ha demostrado que el método TWR funciona mucho mejor en entornos industriales que el método TDOA utilizado hasta el momento.

Así pues, se concluye que para entornos industriales y, concretamente, nuestro objeto de estudio (el Laboratorio de Industria 4.0 del ai2) el mejor método de localización es el de Two Way Ranging (TWR), ya que reduce enormemente los errores valorados en las distintas fases del proyecto, mediante la recopilación de datos obtenidos con el algoritmo de Kalman y aplicándole de forma externa un filtro de mediana móvil (Moving Median) utilizando la herramienta Matlab.

En el futuro, sería una buena opción continuar validando la precisión del sistema mediante el uso de más tags de desarrollo en el modo TWR y probando diferentes filtros aplicados de forma manual y no solo el Moving Median. También, sería interesante llevar el sistema de localización más lejos, aplicándolo al manejo de robot autónomos en entornos industriales permitiendo a los robots evitar la colisión con activos humanos o fijos que estén distribuidos por el área de trabajo. De esta manera, podría obtenerse un entorno de trabajo en el cual tanto robots como activos realicen sus actividades sin poner en peligro a ningún trabajador o usuario del entorno.

Además, este proyecto da pie al desarrollo en mayor profundidad del modo TWR que, pese al trabajo presencial realizado en estos meses de incertidumbre debido a la pandemia, no hemos podido llevar a cabo por falta de tiempo y la complicación para actualizar ciertos elementos del sistema por parte de la empresa.

7. Conclusions and open lines

In this section you can see a comparison of the data collected in the previous phases and a breakdown of the results to obtain a more global vision of them.

PHASE	Method	Fixed tag filter	Mobile tags filter	Error range
I	TDOA	UWB + median filter	UWB Outlier	10-40 cm
II	TDOA	UWB Outlier	Kalman + median filter	7-40 cm
III	TWR	Kalman + moving median filter	Kalman + moving median filter	2-10 cm

Tabla 36: Summary of project phases.

If you look at the table above, we can talk about advantages and disadvantages of the system depending on the phase in which it is.

The initial phase, considering the results obtained, does not present a greater advantage than having a localization system installed to start working. Its results are not reliable or accurate and is not optimal for tracking trajectories or location of assets because, although in the best case we locate some with an error of between 10 and 40 centimeters, there is the possibility that if you do not choose the right method there are errors of up to 3 meters in fixed tags. However, when it comes to mobile tags it is a little more effective.

As for the second phase, the change of anchors and the new distribution slightly improves the results in mobile tags. For fixed tags, there is not much improvement, since there is still, in the best case, an error of between 7 and 10 centimeters. In this case, we obtain more homogeneous values about the measurement zones. It is still observed that the worst values are obtained under very specific conditions: the most extreme zones of the area covered by the anchors and the zones near large metal structures.

Finally, the third phase considerably improves all the results obtained so far. In this case, the fixed tags achieve a minimum error between 2 and 10 centimeters. Although no more tags could be tested due to the amount available in the laboratory, they were placed in zones that were previously conflicting (near metal and at the ends of the area) and it has been shown that the TWR method works much better in industrial environments than the TDOA method used so far.

Therefore, it is concluded that for industrial environments and, specifically, our object of study (the ai2 Industry 4.0 Laboratory) the best location method is the Two Way Ranging (TWR), since it greatly reduces the errors assessed in the various phases of the project, by collecting data obtained with the Kalman algorithm and externally applying a moving median filter using the Matlab tool.

In the future, it would be a good option to continue validating the accuracy of the system by using more development tags in TWR mode and trying different filters applied manually and not only Moving Median. Also, it would be interesting to take the location

system further, applying it to the handling of autonomous robots in industrial environments allowing the robots to avoid collision with human or fixed assets that are distributed by the work area. Of this way, it could be obtained an environment of work in which robots and assets realize their activities without putting in danger to any worker or user of the environment.

In addition, this project gives rise to the further development of the TWR mode which, despite the face-to-face work done in these months of uncertainty due to the pandemic, we have not been able to carry out due to lack of time and the complication of updating certain elements of the system by the company.

8. Normas y referencias

8.1 Disposiciones legales y normas aplicables

- » UNE 157001: Criterios generales para la elaboración formal de los documentos que constituyen un proyecto técnico.
- » Norma ISO 690. Documentación - Referencias bibliográficas: Contenido, forma y estructura.
- » UNE 50-104-94. Referencias bibliográficas: contenido, forma y estructura.
- » Norma ISO 690-2. Información y documentación. Referencias bibliográficas. Parte 2: Documentos electrónicos y sus partes
- » UNE-EN ISO 5455:1996. Dibujos Técnicos. Escalas.

8.2 Softwares empleados

- » Autocad: programa de diseño para la creación y modificación de planos.
- » Eclipse: entorno de desarrollo utilizado para la programación en lenguaje Java (obtención datos del sistema).
- » Matlab R2020a: sistema de cómputo numérico utilizado para el filtrado y representación de datos y trayectorias del sistema de localización.

- » Microsoft Office Excel 2019: programa de hojas de cálculo utilizado para la recopilación y tratamiento de los datos obtenidos.
- » Microsoft Office Word 2019: procesador de texto utilizado para la redacción de la memoria.
- » Visual Studio: entorno de desarrollo integrado para Windows utilizado para la programación en C++ (programación trayectoria Robotino).

8.3 Referencias bibliográficas y documentales

- [1] Documentation | Pozyx. (s. f.). Pozyx NV. <https://www.pozyx.io/documentation>
- [2] 1&1 IONOS España S.L.U. (2020, 20 julio). Bluetooth. IONOS Digitalguide. <https://www.ionos.es/digitalguide/servidores/known-how/que-es-bluetooth/>
- [3] Bluetooth 5.1 AoA y AoD. (2019, 16 agosto). Tecnología Co., Ltd de Shenzhen Feasycom. <http://www.feasywifi.com/info/bluetooth-5-1-aoa-and-aod-38658597.html>
- [4] LMSC, LAN/MAN Standards Committee (Project 802). (s. f.). IEEE802. <https://www.ieee802.org/>
- [5] Marshall Brain, Tracy V. Wilson & Bernadette Johnson "How WiFi Works" 30 April 2001. HowStuffWorks.com. <https://computer.howstuffworks.com/wireless-network.htm>
- [6] GPS.gov: El Sistema de Posicionamiento Global. (s. f.). Gobierno de los Estados Unidos. <https://www.gps.gov/systems/gps/spanish.php>
- [7] Cómo funciona el sistema de posicionamiento GPS. (2014, 24 noviembre). AristaSur. <https://www.aristasur.com/contenido/como-funciona-el-sistema-de-posicionamiento-gps>
- [8] Koyuncu, H. and Hua Yang, S., 2010. A Survey of Indoor Positioning and Object Locating Systems. *IJCSNS International Journal of Computer Science and Network Security*, 10(5), pp.121-128.
- [9] Olav Hegerland, S. (2018, junio). An Experimental Study of a Redundant Position System for the Autonomous Ferry Milliampere Utilizing Ultra Wide-Band Radio Frequencies (TFM). NTNU. <http://hdl.handle.net/11250/2565295>

[10] Feng, T., Yu, Y., Wu, L., Bai, Y., Xiao, Z. and Lu, Z., 2018. A Human-Tracking Robot Using Ultra Wideband Technology. *IEEE Access*, 6, pp.42541-42550.

[11] M. Kolakowski and V. Djaja-Josko, "TDOA-TWR based positioning algorithm for UWB localization system," 2016 21st International Conference on Microwave, Radar and Wireless Communications (MIKON), Krakow, 2016, pp. 1-4, doi: 10.1109/MIKON.2016.7491981.

9. Definiciones y abreviaturas

9.1 Definiciones

- » Autómata: máquina programable capaz de realizar ciertas operaciones de forma autónoma.
- » Bandas ISM: se denomina con este nombre a las bandas de frecuencia sin licencia. Una banda de frecuencia señala un área del espectro electromagnético que se usa para la comunicación técnica.
- » Bluetooth: sistema de localización mediante radiofrecuencia de corto alcance.
- » Hardware: grupo de componentes físicos que conforman un sistema informático. Generalmente, contiene elementos eléctricos y mecánicos.
- » IQ: plano de fase y cuadratura de la señal.
- » Software: grupo de componentes lógicos que conforman los programas y rutinas que permite al sistema informático realizar diversas tareas.
- » Topics: se define como un tema del bróker al que nos suscribimos en una conexión MQTT para obtener los datos que se están publicando.
- » Wifi: sistema de conexión inalámbrico mediante punto de acceso de red.

9.2 Abreviaturas

- » AoA: Angle of Arrival.
- » AoD: Angle os Departure.

- » IEEE: Institute of Electrical and Electronic Engineers.
- » IoT: Internet of Things.
- » IPS: Indoor Positioning System.
- » Gbps: Gigabits por segundo. Se trata de una medida de velocidad para la transmisión de datos, es decir, el caudal de datos.
- » GPS: sistema de posicionamiento global (Global Positioning System en inglés) mediante satélites y trilateración.
- » Mbps: megabits por segundo. Se trata de una medida de velocidad para la transmisión de datos, es decir, el caudal de datos.
- » NFC: Near Field Communication. Sistema de comunicación inalámbrica entre dispositivos cercanos.
- » PoI: Point of Interest.
- » RSSI: Received Signal Strength Indicator.
- » RTLS: Real Time Location System.
- » TDOA: Time Difference Of Arrival. Sistema de posicionamiento mediante la diferencia temporal de las señales enviadas a dos antenas, donde la posición de las antenas es conocida.
- » TOA: Time Of Arrival. Sistema de posicionamiento basado en la intersección de los receptores.
- » TWR: Two Way Ranging. Sistema de posicionamiento por el que antenas y tags se comunican de forma bidireccional y se determina la distancia entre ellos conociendo el tiempo de transmisión y la velocidad de la luz.
- » UWB: sistema de banda ultra ancha (Ultra Wide Band en inglés) que usa un ancho de banda mayor de 500 MHz o del 25% de la frecuencia central.
- » WPAN: Wireless Personal Area Network. Se trata de una red inalámbrica de uso personal



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES

TRABAJO FINAL DE MÁSTER

ANEXO I

**MÁSTER EN AUTOMATIZACIÓN E INFORMÁTICA
INDUSTRIAL**

Título del proyecto:

**LOCALIZACIÓN PRECISA DE ACTIVOS Y HUMANOS EN
ESPACIOS CERRADOS PARA SEGURIDAD EN ENTORNOS
INDUSTRIALES**



1. Código desarrollado en Java y tratamiento de los datos obtenidos

En este primer punto del Anexo I, se procede a describir el código desarrollado en Java para el sistema de localización y el tratamiento de los datos obtenidos del mismo.

Para la obtención de datos del sistema, se crea un paquete llamado `pozyx`, dentro de un proyecto denominado `MQTT_POZYX`, como se ve en la figura 50.

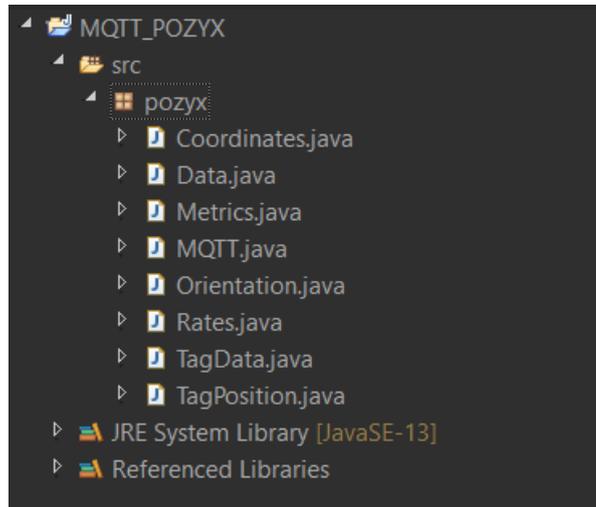


Figura 50: Eclipse java – Estructura del proyecto.

Vemos, además que este paquete contiene 8 ficheros diferentes, de las cuales, la más relevante será “MQTT.java” aunque esta no funcionaría sin “TagPosition.java” ya que la finalidad es obtener las posiciones de los tags distribuidos en el laboratorio.

Cada fichero contiene una clase con el mismo nombre y estas están vacías ya que se componen solo por variables y un constructor vacío, a excepción de la clase MQTT. Veamos un pequeño resumen de la estructura:

Clase	Descripción	Métodos
Coordinates	Devuelve las coordenadas: x, y, z.	Coordinates()
Data	Se le pasan las Clases TagData, Coordinates, Orientation y Metrics. Además, devuelve un objeto: anchorData.	Data()
Metrics	Se le pasa la clase Rates y devuelve la latencia.	Metrics()
MQTT	Crea la conexión con el servidor, obtiene los datos y los envía a un archivo txt.	Void messageArrived(String, MqttMessage) Void connectionLost(Throwable) Void deliveryComplete(IMqttDeliveryToken) Void main(String[]) Void comprobarArchivo(String, String) run()

Orientation	Devuelve las tres orientaciones posibles: yaw, roll, pitch.	Orientation()
Rates	Devuelve la velocidad de actualización (update) y de éxito (success).	Rates()
TagData	Clase vacía.	TagData()
TagPosition	Se le pasa la clase Data y devuelve la versión (version), el identificador del tag (tagId), la fecha (timestamp) y nos indica mediante booleanos si el tag está activo (alive) y si funciona correctamente (success).	TagPosition()

Tabla 37: Eclipse java – Estructura de las clases y sus métodos.

De esta manera, las clases más pequeñas, compuestas por variables y su constructor, se pueden ver en las figuras de la 51 a la 57:

```

1 package pozyx;
2
3 public class Coordinates {
4
5     public int x;
6     public int y;
7     public int z;
8
9     public Coordinates() {
10
11     }
12 }
13 }
14

```

Figura 51: Eclipse java – Estructura clase Coordinates.

```

1 package pozyx;
2
3 public class Data {
4     public TagData tagData;
5     public Coordinates coordinates;
6     public Object[] anchorData;
7     public Orientation orientation;
8     public Metrics metrics;
9
10    public Data() {
11    }
12 }
13
14

```

Figura 52: Eclipse java – Estructura clase Data.

```

1 package pozyx;
2
3 public class Metrics {
4     public double latency;
5     public Rates rates;
6
7
8    public Metrics() {
9    }
10 }
11
12
13

```

Figura 53: Eclipse java – Estructura clase Metrics.

```

1 package pozyx;
2
3 public class Orientation {
4     public double yaw;
5     public double roll;
6     public double pitch;
7
8    Orientation() {
9    }
10 }
11
12 }
13

```

Figura 54: Eclipse java – Estructura clase Orientation.

```

1 package pozyx;
2
3 public class Rates {
4
5     public double update;
6     public double success;
7
8     Rates() {
9
10    }
11 }
12

```

Figura 55: Eclipse java – Estructura clase Rates.

```

1 package pozyx;
2
3 public class TagData {
4     TagData() {
5
6     }
7
8 }
9

```

Figura 56: Eclipse java – Estructura clase TagData.

```

1 package pozyx;
2
3 public class TagPosition {
4
5     public String version;
6     public boolean alive;
7     public String tagId;
8     public boolean success;
9     public double timestamp;
10    public Data data;
11
12    public TagPosition() {
13
14    }
15
16 }
17

```

Figura 57: Eclipse java – Estructura clase TagPosition.

Sin embargo, entraremos a ver un poco más en profundidad la clase restante, la “MQTT”.

Lo primero que podemos ver en el fichero “MQTT.java” (figura x), es la importación de las librerías necesarias para su funcionamiento, entre ellas vemos librerías conocidas como “java.io” o “java.util”. Además, se importa la librería “Eclipse.paho” que permite los protocolos de mensajería Machie-to-machine (M2M) y el API de Google “Gson” que permite transformar objetos JSON a objetos de Java, es decir, nos permite la deserialización de los objetos.

```

1 package pozyx;
2
3 import java.io.BufferedWriter;
4 import java.io.File;
5 import java.io.FileWriter;
6 import java.io.IOException;
7 import java.io.PrintWriter;
8
9 import org.eclipse.paho.client.mqttv3.IMqttDeliveryToken;
10 import org.eclipse.paho.client.mqttv3.MqttCallback;
11 import org.eclipse.paho.client.mqttv3.MqttClient;
12 import org.eclipse.paho.client.mqttv3.MqttConnectOptions;
13 import org.eclipse.paho.client.mqttv3.MqttException;
14 import org.eclipse.paho.client.mqttv3.MqttMessage;
15 import org.eclipse.paho.client.mqttv3.persist.MemoryPersistence;
16
17 import com.google.gson.Gson;

```

Figura 58: Eclipse java – Librerías.

Tras importar las librerías, necesitamos crear la clase que vamos a utilizar. En este caso se llama MQTT, ya que contiene la conexión al servidor mediante un puente MQTT.

```

19 public class MQTT implements MqttCallback {
20
21     String address = "C:/Barcabca/UPV/TFM/";
22     public String ruta;
23

```

Figura 59: Eclipse java –clase MQTT.

Vemos que, además, se declaran varios términos, un multimap que más adelante veremos que se utiliza para el trabajo con los datos del servidor, la dirección que usaremos para guardar los archivos y la variable ruta, que se utilizará en conjunto con la dirección.

Dentro de esta clase, recordemos que se encuentran los siguientes métodos:

MQTT	void messageArrived(String topic, MqttMessage message)
	void comprobarArchivo(String texto, String ruta)
	void run()
	void main(String[] args)
	void deliveryComplete(IMqttDeliveryToken arg0)
	void connectionLost(Throwable arg0)

Tabla 38: Eclipse java –métodos clase MQTT.

Dentro del primer método, se reciben los mensajes de la conexión MQTT. Este método incluye inicialmente la deserialización del JSON que envía el servidor.

```

24 @Override
25 public void messageArrived(String topic, MqttMessage message) throws Exception {
26
27     PrintWriter stdout = new PrintWriter(System.out);
28     Gson gson = new Gson();
29     TagPosition[] tagPositions = gson.fromJson(message.toString(), TagPosition[].class);
30
31     for (TagPosition tagPosition : tagPositions) {
32
33         if (tagPosition.success) {

```

Figura 60: Eclipse java –método messageArrived (1).

Para la deserialización, basta con crearnos un objeto Gson e invocar el método fromJson, al cual se le pasa el mensaje con un cast para convertirlo en String y la clase que queremos deserializar, en nuestro caso “TagPosition”.

Posteriormente a eso, con un for al tagPosition y el if si funciona correctamente, se procede a obtener los datos y enviarlos a un fichero .txt en una ruta especificada.

Veamos un ejemplo para cuando se utilizan tags fijos y un ejemplo para tags de desarrollo.

```
35         if (tagPosition.tagId.equals("5125")) {
36
37             ruta = adress + "pos5125.txt";
38             String texto = "x: " + tagPosition.data.coordinates.x + ", y: " + tagPosition.data.coordinates.y + "\n";
39             comprobarArchivo(texto, ruta);
40             System.out.println("Successfully wrote the 5125 to the file.");
41
42         }
43
```

Figura 61: Eclipse java –método messageArrived (2).

Para obtener los datos correspondientes al tag 5125, se compara el tagId del mensaje entrante con un string con el número del identificador y en caso de que la coincidencia devuelva True, completaremos la ruta de guardado, utilizando la dirección declarada anteriormente y añadiéndole el nombre del archivo.

Definimos como queremos guardar los valores de las posiciones, utilizando un String llamado texto y que posee la siguiente estructura:

x: “posición x”, y: “posición y”

Luego, se llama al método “comprobarArchivo”, se le pasa la ruta y el texto como variable y se imprime por pantalla un mensaje que indica que la posición del tag ha sido escrita de forma correcta al archivo.

En el caso de los tags de desarrollo, el procedimiento es exactamente igual, ya que estos tags también poseen un número de identificación (figura x).

```
94         if (tagPosition.tagId.equals("27252")) {
95
96             ruta = adress + "pos27252.txt";
97             String texto = "x: " + tagPosition.data.coordinates.x + ", y: " + tagPosition.data.coordinates.y + "\n";
98             comprobarArchivo(texto, ruta);
99             System.out.println("Successfully wrote the 27252 to the file.");
100
101         }
102
```

Figura 62: Eclipse java –método messageArrived (3).

Finalmente, para acabar y cerrar el método “messageArrived” imprimimos por pantalla los datos que están entrando al método y nos aseguramos de enviar los mensajes en la cola.

```
125         System.out.println("*****");
126         System.out.println("Tag ID: " + tagPosition.tagId);
127         System.out.println("Coordenadas tag (x,y,z):" + tagPosition.data.coordinates.x + "," + tagPosition.data.coordinates.y
128             + "," + tagPosition.data.coordinates.z);
129         System.out.println("*****");
130
131     }
132 }
133
134     stdout.flush();
135 }
136 }
137
```

Figura 63: Eclipse java –método messageArrived (4).

Tras este, encontramos varios métodos vacíos que se generan de forma automática cuando se realizan las conexiones MQTT.

```

138● @Override
139 public void connectionLost(Throwable arg0) {
140     // TODO Auto-generated method stub
141 }
142
143
144● @Override
145 public void deliveryComplete(IMqttDeliveryToken arg0) {
146     // TODO Auto-generated method stub
147 }
148
149

```

Figura 64: Eclipse java –métodos `connectionLost` y `deliveryComplete`.

Lo siguiente son los métodos “main” que es el método principal y el que se ejecutará siempre y “comprobarArchivo”:

```

150● public static void main(String[] args) throws IOException {
151     new MQTT().run();
152 }
153
154● public static void comprobarArchivo(String texto, String ruta) throws IOException {
155
156     File file = new File(ruta);
157     FileWriter fw = new FileWriter(file.getAbsolutePath(), true);
158     BufferedWriter bw = new BufferedWriter(fw);
159     bw.write(texto);
160     bw.close();
161 }
162

```

Figura 65: Eclipse java –métodos `main` y `comprobarArchivo`.

Como se puede ver en la figura 65, el “main” se realiza una llamada al método `run` y “comprobarArchivo” crea un archivo en la ruta especificada si no existe y escribe en la línea de texto que queremos. Una vez escrito, cierra el archivo. En el caso de que el archivo este creado, simplemente añadirá las líneas de texto al final del mismo.

```

public void run() throws IOException {

    String topic = "tags";
    int qos = 0;
    String broker = "tcp://158.42.126.70:1883";
    String clientId = "JavaSample_sub";
    MemoryPersistence persistence = new MemoryPersistence();

    try {
        MqttClient sampleClient = new MqttClient(broker, clientId, persistence);
        MqttConnectOptions connOpts = new MqttConnectOptions();
        connOpts.setCleanSession(true);

        System.out.println("Connecting to broker: " + broker);
        sampleClient.connect(connOpts);
        System.out.println("Connected");

        //Topic subscription
        sampleClient.subscribe(topic, qos);
        System.out.println("Suscrito al topic: " + topic);

        //Sets the callback listener to use for events that happen asynchronously.
        while(true) {
            sampleClient.setCallback(this);
        }
    } catch(MqttException me){
        System.out.println("reason " + me.getReasonCode());
        System.out.println("msg " + me.getMessage());
        System.out.println("loc " + me.getLocalizedMessage());
        System.out.println("cause " + me.getCause());
        System.out.println("except " + me);
        me.printStackTrace();
    }
}
}

```

Figura 66: Eclipse java –método run.

Como comentábamos antes, el “main” hace una llamada al método run(), el cual podemos ver en la figura 66. En él se programó la conexión al servidor.

Para la conexión se utiliza un topic al que suscribirnos (concretamente el topic “tags”), una dirección IP (la de nuestro servidor con el puerto de conexión) y el identificador de cliente. Nos conectamos utilizando el “connect”, nos suscribimos con el “suscribe” y añadimos las excepciones.

Con esto, ya somos capaces de recibir los datos del servidor y almacenarlos en documentos.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES

TRABAJO FINAL DE MÁSTER

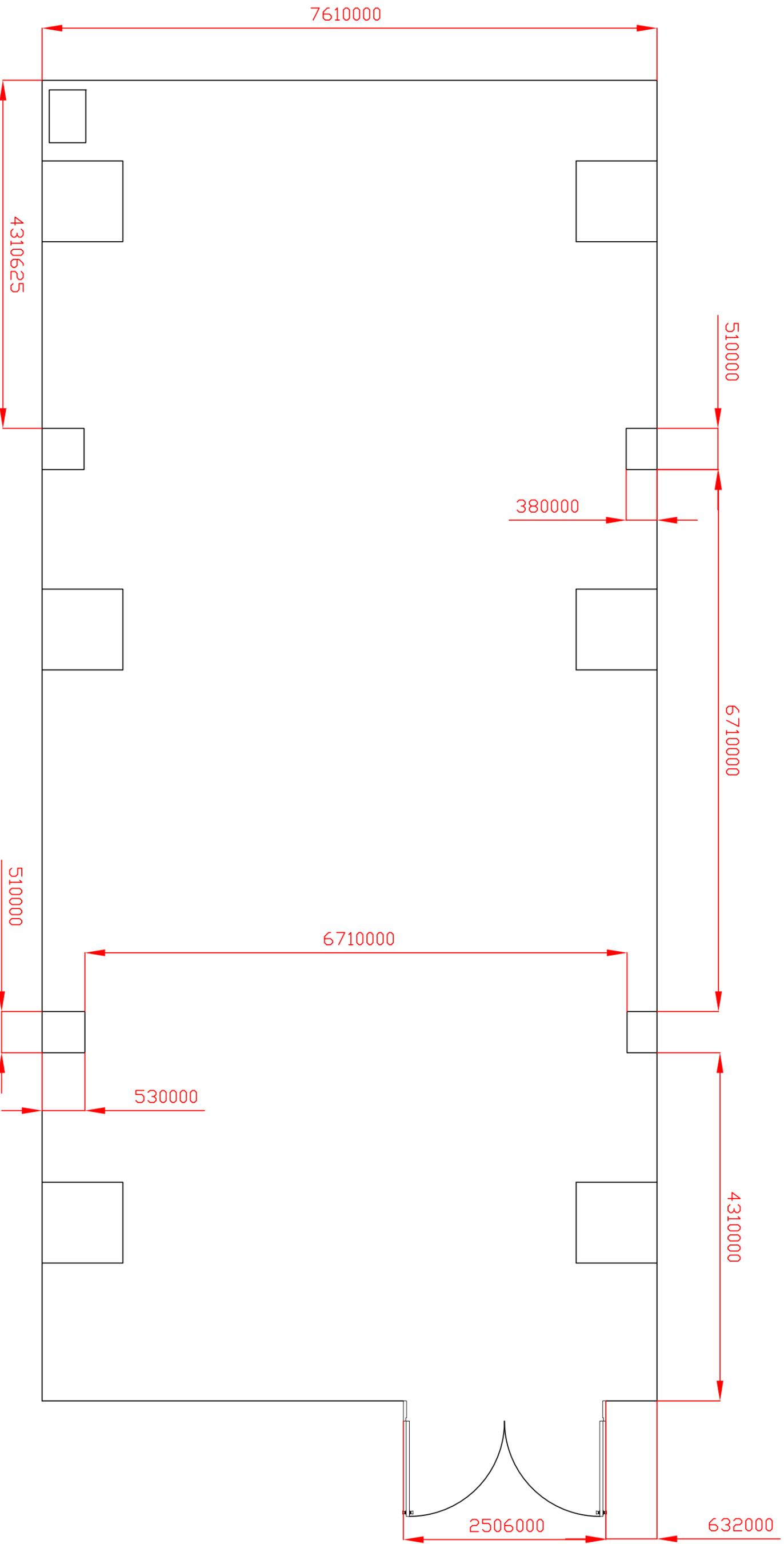
ANEXO II

**MÁSTER EN AUTOMATIZACIÓN E INFORMÁTICA
INDUSTRIAL**

Título del proyecto:

**LOCALIZACIÓN PRECISA DE ACTIVOS Y HUMANOS EN
ESPACIOS CERRADOS PARA SEGURIDAD EN ENTORNOS
INDUSTRIALES**





LOCALIZACIÓN PRECISA DE ACTIVOS Y HUMANOS EN ESPACIOS CERRADOS PARA
SEGURIDAD EN ENTORNOS INDUSTRIALES

Fecha

SEPTIEMBRE-2020

Autor

Bárbara María
Cabrera Castro



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

DPTO. DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

Máster Universitario en Automática
e Informática Industrial

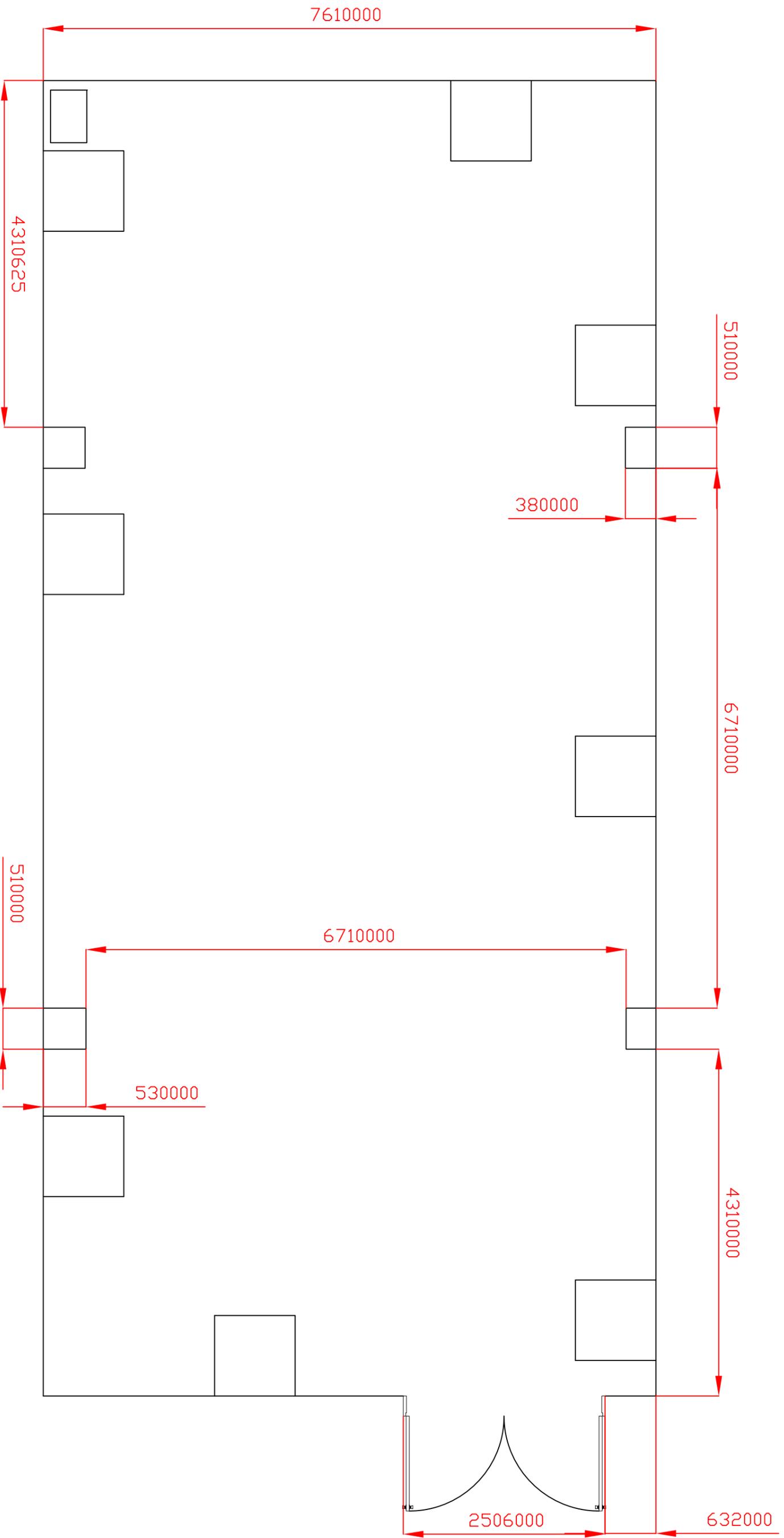
Id. s. normas

UNE-EN-DIN

PLANO Y DISTRIBUCIÓN ORIGINAL DE ANTENAS
DEL LABORATORIO PEDRO ALBERTOS

PLANO 1

Escala 1/50



LOCALIZACIÓN PRECISA DE ACTIVOS Y HUMANOS EN ESPACIOS CERRADOS PARA
SEGURIDAD EN ENTORNOS INDUSTRIALES

Fecha

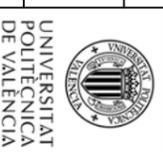
Autor

OCTUBRE-2020

Bárbara María
Cabrera Castro

Id. s. normas

UNE-EN-DIN



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

DPTO. DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

Master Universitario en Automática
e Informática Industrial

PLANO Y DISTRIBUCIÓN ACTUAL DE ANTENAS
DEL LABORATORIO PEDRO ALBERTOS

PLANO 2

Escala 1/50