Evaluación de módulos ultra wide band (UWB) para sistemas de navegación local

Alfredo Andres Meléndez Mendoza

UVG UNIVERSIDAD DEL VALLE DE GUATEMALA

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Evaluación de módulos ultra wide band (UWB) para sistemas de navegación local

Trabajo de graduación presentado por Alfredo Andres Meléndez Mendoza para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2024

Vo.Bo.:

(f) _____

M. Sc. Carlos Esquit

Tribunal Examinador:

(f) _____

M.Sc. Carlos Esquit

(f) _____

M. Sc. Miguel Enrique Zea Arenales

(f) _____

Ing. Kurt Emmanuel Kellner

Fecha de aprobación: Guatemala, 13 de febrero de 2025.

Prefacio

El desarrollo de este trabajo surgió de mi interés por fusionar sensores y algoritmos para mejorar sistemas de navegación. La oportunidad de profundizar en este tema me permitió comprender los retos de la ingeniería en el día a día y las formas de solucionar los problemas. También, me permitió aplicar los conocimientos adquiridos durante la carrera y con ello contribuir a la ciencia.

Quiero expresar mis agradecimientos primordialmente a Dios, a mi familia, especialmente a mis padres, quienes me han apoyado desde niño, han creído en mí y han sido mi ejemplo a seguir de nunca rendirse. Quiero agradecer también a mi familia en Texas y Florida por brindarme oportunidades a lo largo de la carrera. Agradezco al Ing. Kurt Kellner, Ing. Pedro Castillo, M.Sc. Miguel Zea, a Ph.D. Luis Rivera, quienes me brindaron orientación y apoyo invaluables a lo largo de todo el proceso y la educación brindada en la carrera. Sus comentarios, sugerencias y enseñanzas fueron fundamentales para el desarrollo de este trabajo. Finalmente, agradezco a mis amistades por su constante apoyo emocional y comprensión durante los momentos más desafiantes de este proceso.

Este trabajo representa no solo un logro académico, sino también un viaje de conocimiento personal y profesional, por el cual estaré agradecido. Quiero dedicar este trabajo a mi abuelo Chepe, quien me enseñó el mundo de la electrónica.

Índice

Prefacio III

Lista de figuras X

Lista de cuadros XI

Resumen XIII

Abstract XV

1. Introducción 1

2. Antecedentes 3

2.1. Implementación de un sistema de localización en tiempo real en espacios cerrados para el robot Rover UVG 3

2.2. Uso de tecnología de banda ultra-ancha para control de ubicación de un carro 4

2.3. Mejora de la precisión de posicionamiento del módulo MDEK1001 al tener múltiples rangos y en interiores; implementación de EKF 5

3. Justificación 8

4. Objetivos 10

4.1. Objetivo general 10

4.2. Objetivos específicos 10

5. Alcance 11

6. Marco teórico 13

6.1. Tecnología ultra wide band 13

6.2. Sistema de captura de movimiento con cámaras Optitrack 15
6.3. Sistema de navegación local 17
6.4. Aplicaciones de posicionamiento en interiores 19
6.5. Filtro de Kalman 19
6.6. Filtro de Savitzky-Golay 22
6.7. Homografía 23
6.8. Filtro complementario 25

7. Módulo MDEK1001 27
7.1. Montaje en Robotat 28
7.2. Conexión a MDEK1001 31
7.3. Decodificación formato TLV 33
7.3.1. Formato little-endian 33
7.3.2. Ejemplo de decodificación 34

8. Selección de microcontrolador 35
8.1. Métodos de comunicación 36
8.1.1. Comunicación wifi y bluetooth 36
8.1.2. Comunicación UART 37
8.1.3. Comunicación I2C 38

9. Unidad de medición inercial MPU9250 40

10. Implementación física 42
10.1. Carro de pruebas 42
10.2. Integración de módulos en PCB 43

11. Calibración por homografía y pruebas estáticas 49
11.1. Calibración por homografía 51
11.1.1. Aplicación de la homografía 51
11.1.2. Sistema de coordenadas UWB y Optitrack 51
11.1.3. Metodología de calibración por homografía 52
11.1.4. Resultados de calibración 53

12. Filtros 57
12.1. Filtro complementario 58
12.1.1. Filtro pasa altas (HPF) aplicado sobre aceleraciones IMU 58
12.1.2. Combinación de los filtros LPF + HPF 59
12.1.3. Resultados estáticos de filtro complementario normal 60
12.2. Filtro complementario butterworth 62
12.2.1. Filtro pasa bajas (LPF) 63

12.2.2. Filtro pasa altas (HPF) Butterworth aplicado sobre aceleraciones IMU 63
12.2.3. Combinación de los filtros LPF + HPF Butterworth 64
12.2.4. Resultados estáticos de filtro complementario Butterworth 64

13. Pruebas dinámicas 68
13.1. Filtro complementario normal 68
13.2. Filtro complementario Butterworth 71
13.3. Implementación en tiempo real 74

14. Conclusiones 77

15. Recomendaciones 78

16. Referencias 80

17. Anexos 83
17.1. Pruebas dinámicas adicionales 83
17.2. Esquemáticos e implementación PCB 89
17.3. Carro de pruebas e iteraciones 92
17.4. Pruebas previas 94

18. Glosario 98

Lista de figuras

Figura 1. *Módulo MDEK1001.* 3

Figura 2. *Entorno de conexión de anclas y etiquetas.* 5

Figura 3. *Montaje de experimento con ground truth.* 6

Figura 4. *Precisión en distintos sistemas de localización.* 13

Figura 5. *Tiempo de vuelo (ToF) con rango bidireccional (TWR).* 14

Figura 6. *Topologías de comunicación UWB.* 15

Figura 7. *Montaje de 6 cámaras PrimeX 41.* 16

Figura 8. *Esquema de conexión cámaras PrimeX de Optitrack.* 17

Figura 9. *Aplicaciones de sistemas de navegación.* 17

Figura 10. *Posicionamiento por tres rangos.* 18

Figura 11. *Algoritmo de cálculo EKF.* 21

Figura 12. *Algoritmo de cálculo UKF.* 21

Figura 13. *Reconstrucción de señal con filtro SG.* 23

Figura 14. *Tipos de homografía.* 24

Figura 15. *Arquitectura de filtro complementario normal.* 26

Figura 16. *Pines de Raspberry Pi model A para montar MDEK1001.* 28

Figura 17. *Diagrama de comunicación de módulos MDEK1001-Dev.* 28

Figura 18. *Sujetador MDEK1001.* 29

Figura 19. *MDEK1001 montado en trípodes del laboratorio Robotat UVG.* 30

Figura 20. *Disposición de 6 módulos MDEK1001 en laboratorio Robotat.* 31

Figura 21. *Esquema de comunicación UWB para modo Shell o TLV.* 32
Figura 22. *Formato TLV para obtener posición.* 33
Figura 23. *Pinout DOIT DevKit V1 ESP32.* 36
Figura 24. *Lectura en osciloscopio modo decodificación UART.* 38
Figura 25. *Conexión de dispositivos con comunicación I2C.* 39
Figura 26. *Pinout de MPU9250-GY91.* 41
Figura 27. *Módulos montados en carro de pruebas conectados mediante jumpers.* 44
Figura 28. *Arquitectura de PCB.* 45
Figura 29. *Bottom layer y vista 3D del PCB.* 46
Figura 30. *PCB fabricada con montaje de headers.* 47
Figura 31. *Módulo de navegación local con carcasa.* 47
Figura 32. *Iteración final de carro con montaje de módulo de navegación.* 48
Figura 33. *Arquitectura de obtención y procesamiento de datos.* 50
Figura 34. *Sistema de coordenadas Optitrack vs. UWB (sin transformación) y posición de postes de cámaras con etiquetas en plataforma del laboratorio Robotat UVG.* 52
Figura 35. *Transformación de coordenadas UWB a Optitrack mediante matriz de homografía 11.1.4.* 54
Figura 36. *Arquitectura de filtro complementario con constante de tiempo τ para posición combinando sistema UWB con aceleraciones de IMU.* 58
Figura 37. *Aplicación de filtro complementario normal en puntos de datasets de calibración transformados como pruebas estáticas.* 60
Figura 38. *Arquitectura de filtro complementario usando Butterworth para posición combinando sistema UWB con aceleraciones de la IMU.* 63
Figura 39. *Aplicación de filtro complementario Butterworth en puntos de datasets de calibración transformados como pruebas estáticas.* 65
Figura 40. *Trayectoria 1 con filtro complementario normal con* $\mathbb{R}^2 = 0.9469$. 69
Figura 41. *Trayectoria 2 con filtro complementario normal con* $\mathbb{R}^2 = 0.9672$. 69
Figura 42. *Trayectoria 3 con filtro complementario normal con* $\mathbb{R}^2 = 0.9322$. 70
Figura 43. *Trayectoria 4 con filtro complementario normal con* $\mathbb{R}^2 = 0.8338$. 70
Figura 44. *Trayectoria 5 con filtro complementario normal con* $\mathbb{R}^2 = 0.9582$. 71
Figura 45. *Trayectoria 1 con filtro complementario Butterworth con* $\mathbb{R}^2 = 0.9793$. 72

Figura 46. *Trayectoria 2 con filtro complementario Butterworth con* $\mathbb{R}^2 = 0.9886$. 72
Figura 47. *Trayectoria 3 con filtro complementario Butterworth con* $\mathbb{R}^2 = 0.9717$. 73
Figura 48. *Trayectoria 4 con filtro complementario Butterworth con* $\mathbb{R}^2 = 0.8958$. 73
Figura 49. *Trayectoria 5 con filtro complementario Butterworth con* $\mathbb{R}^2 = 0.9818$. 74
Figura 50. *Entorno de PyGame para visualización de módulo de navegación en el centro.* 75
Figura 51. *Entorno de PyGame para visualización de módulo de navegación fuera de la plataforma.* 76
Figura 52. *Trayectoria cuadrada 1 con filtro complementario normal con* $\mathbb{R}^2 = 0.9695$ *y media Qf = 69.24 %.* 83
Figura 53. *Trayectoria cuadrada 2 con filtro complementario normal con* $\mathbb{R}^2 = 0.9729$ *y media Qf = 68.27 %.* 84
Figura 54. *Trayectoria cuadrada 3 con filtro complementario normal con* $\mathbb{R}^2 = 0.9209$ *y media Qf = 69.89 %.* 84
Figura 55. *Trayectoria cuadrada 4 con filtro complementario normal con* $\mathbb{R}^2 = 0.9591$ *y media Qf = 70.69 %.* 85
Figura 56. *Trayectoria cuadrada 5 con filtro complementario normal con* $\mathbb{R}^2 = 0.9784$ *y media Qf = 70.76 %.* 85
Figura 57. *Trayectoria cuadrada 1 con filtro complementario Butterworth con* $\mathbb{R}^2 = 0.9895$ *y media Qf = 69.24 %.* 86
Figura 58. *Trayectoria cuadrada 2 con filtro complementario Butterworth con* $\mathbb{R}^2 = 0.9902$ *y media Qf = 68.27 %.* 86
Figura 59. *Trayectoria cuadrada 3 con filtro complementario Butterworth con* $\mathbb{R}^2 = 0.9437$ *y media Qf = 69.89 %.* 87
Figura 60. *Trayectoria cuadrada 4 con filtro complementario Butterworth con* $\mathbb{R}^2 = 0.9895$ *y media Qf = 70.69 %.* 87
Figura 61. *Trayectoria cuadrada 5 con filtro complementario Butterworth con* $\mathbb{R}^2 = 0.9922$ *y media Qf = 70.76 %.* 88
Figura 62. *Esquemático de placa PCB para integración de sensores y control de carro.* 89
Figura 63. *Diagrama de bloques PCB.* 90
Figura 64. *Primera iteración de PCB fabricada.* 91

Figura 65. *PCB fabricada con módulos montados.* 91

Figura 66. *Iteración 2 del carro antes del montaje completo.* 92

Figura 67. *Iteración 2 del carro a control remoto con el montaje completo tomando muestras de Optitrack con marcador y el módulo de navegación en la plataforma del laboratorio Robotat en UVG.* 93

Figura 68. *Iteración 2 del carro con la primera interfaz de visualización de trayectorias y orientación únicamente de la etiqueta MDEK1001.* 93

Figura 69. *Dos etiquetas con 8 anclas funcionando en paralelo y un DOIT DevKit V1.* 94

Figura 70. *Nube de puntos de 2 etiquetas estáticas.* 94

Figura 71. *Test con IMU MPU9250-GY91 y cubo de calibración para filtros complementarios en roll, pitch y yaw guardados en memoria SD para análisis.* 95

Figura 72. *Implementación de carro a control remoto usando DualShock4 de Playstation conectado al ESP32.* 95

Figura 73. *Primera implementación de envío de datos mediante wifi TCP en una página del ESP32 configurado como Soft-AP.* 96

Figura 74. *Primer plot de datos extraídos de la terminal de la aplicación DRTLS como archivo .log y decodificados mediante parsing con un script de Python y graficado en Matlab desde el archivo .csv generado.* 96

Figura 75. *Primera obtención de datos de ubicación sin decodificar en la terminal del IDE de Arduino.* 97

Lista de cuadros

Cuadro 1. *Comparación de medias y desviaciones estándar de sistema UWB crudo y transformado contra Optitrack para el eje x* 55

Cuadro 2. *Comparación de medias y desviación estándar de sistema UWB crudo y transformado contra Optitrack para el eje y* 56

Cuadro 3. *Medición de efectividad del filtro complementario normal en los distintos puntos para eje x* 61

Cuadro 4. *Medición de efectividad del filtro complementario normal en los distintos puntos para eje y.* 62

Cuadro 5. *Medición de efectividad de filtro complementario Butterworth en los distintos puntos para eje x* 66

Cuadro 6. *Medición de efectividad de filtro complementario Butterworth en los distintos puntos para eje y* 67

Resumen

El proyecto propuesto crea un sistema de navegación local en dos dimensiones por medio de la tecnología de banda ultra ancha con los módulos MDEK1001 de la empresa Qorvo en conjunto con una unidad de medición inercial. Esta es una alternativa para localización en una escala menor, donde la precisión mejora respecto de otros sistemas de navegación, como el GNSS a mayor escala.

El objetivo principal consiste en la evaluación de los módulos para mejorar exactitud y precisión de la posición e implementación de orientación en un entorno de dos dimensiones y compararlos con un *ground truth* de un sistema de cámaras Optitrack en la plataforma del laboratorio *Robotat* en la Universidad del Valle de Guatemala (UVG). Los datos obtenidos de los sensores de banda ultra ancha y la unidad de medición inercial utilizando el acelerómetro son fusionados por medio de filtros complementarios. Se utiliza homografía proyectiva para representar el marco de referencia de navegación del sistema de sensores de banda ultra ancha en el marco de referencia de navegación del sistema de cámaras Optitrack. Con los datos transformados y fusionados por medio de filtros, se analizaron las pruebas para implementarlo en tiempo real haciendo uso de comunicación wifi TCP para recepción y procesamiento de datos.

Dentro de las pruebas, se encuentran las estáticas y dinámicas, cada una con su respectiva configuración de montaje en condiciones de línea de vista. Las pruebas dinámicas se realizan con un carro controlado de manera remota por bluetooth con un PCB que integra los módulos de banda ultra ancha, MPU9250, ESP32 y un marker de Optitrack. Por último, se analizan

cualitativa y cuantitativamente los datos utilizando MATLAB de modo que se pueda validar la mejora de exactitud y precisión para la creación de un sistema de navegación local 2D.

Palabras clave: banda ultra ancha, unidad de medición inercial, sistema de navegación local, fusión de sensores, homografía.

Abstract

This study evaluates the performance of Qorvo's Ultra-Wideband (UWB) MDEK1001 modules through sensor fusion with an Inertial Measurement Unit (IMU) MPU9250 to develop a two-dimensional Local Navigation System (LNS). The proposed system serves as an alternative to large-scale positioning systems such as GPS and GNSS or small-scale camera-based systems like OptiTrack. This research focuses on improving the accuracy and precision of UWB in controlled environments under Line-of-Sight (LOS) conditions.

The primary objective was to design a motion capture system capable of enhancing positional accuracy by using sensor fusion techniques, including complementary filters, and projective homography calibration for aligning the UWB system with the OptiTrack Motion Capture (MoCap) coordinate system, used as ground truth due to its precision and accuracy of up to 0.02 mm. The study involved static and dynamic testing in LOS scenarios using an ESP32 microcontroller, MDEK1001 modules, and an MPU9250 integrated into a PCB referred to as the "Navigation Module", which can operate independently or be mounted on a vehicle for remote data capture and accomplish LOS conditions.

Experimental results demonstrate significant improvements in accuracy and precision, with sensor fusion and homography reducing positional deviations by approximately 50 % and achieving deviations below 25 mm. For dynamic trajectories, the coefficient of determination ($\mathbb{R}^2$) were above 0.9 in most cases with Quality Factor on a 69 % mean for trajectories. These findings validate the potential of UWB and IMU sensor fusion for local navigation applications, offering a cost-effective and precise alternative for various scenarios and requirements.

Keywords: ultra-wideband, inertial measurement unit, local navigation system, sensor fusion, homography.

CAPÍTULO 1

Introducción

Este trabajo tiene como propósito evaluar los módulos MDEK1001 de la empresa Qorvo que utilizan tecnología de ancho de banda ultra ancha o UWB. Para evaluar estos módulos se implementa una unidad de medición inercial IMU de 9 grados de libertad, que permite observar mediciones de acelerómetro, giroscopio y magnetómetro, cuya información se puede fusionar con la tecnología UWB, de modo que se puedan tener mediciones con mejor exactitud y precisión mientras se implementa orientación. La forma en la que se delimitaron los objetivos fue realizar los experimentos en condiciones controladas en dos dimensiones con línea de vista para evitar errores. Las condiciones controladas son dadas por el sistema de cámaras Optitrack con precisión de hasta 0.01 mm y sirven como referencia o *ground truth* (GT) para comparar el sistema UWB.

Los módulos UWB MDEK1001 carecen de orientación. Utilizando el giroscopio y acelerómetro de la IMU, se puede obtener orientación en los tres ejes con fusión de sensores. A lo largo del trabajo se integraron los módulos UWB y la IMU por medio de una PCB para tener un módulo de navegación que puede ser montado en un carro de pruebas para las mediciones en condiciones de línea de vista LOS. Las condiciones LOS involucran no tener elementos que puedan obstaculizar o causar interferencias en la plataforma del laboratorio Robotat, donde se hicieron mediciones en conjunto con el sistema Optitrack. También se ejecutó una homografía que fue parte fundamental para mapear el sistema de referencia UWB

hacia el sistema de referencia Optitrack y así evaluar la exactitud. En el caso de la precisión, se mejora o bien la desviación estándar disminuye al implementar filtros complementarios.

Se obtuvieron resultados de mejora en la exactitud y precisión. Al pasar la información por la matriz de homografía y aplicar filtros complementarios se obtuvieron mejoras alrededor de 50 % en la precisión con errores de exactitud por debajo de 40 %. aproximadamente. Los resultados ayudan a validar que sí se puede mejorar la exactitud y precisión fusionando sensores con el fin de crear un sistema de navegación local (LNS). Esto puede ser útil para distintas aplicaciones a pequeña y gran escala donde la posición y orientación son cruciales para implementación de control o bien el registro de movimientos. Para ello, también se ejecutaron pruebas dinámicas comparando trayectorias con GT, crudas transformadas y transformadas filtradas para evaluar comportamientos de filtros en aplicaciones de movimiento.

CAPÍTULO 2

Antecedentes

2.1. Implementación de un sistema de localización en tiempo real en espacios cerrados para el robot Rover UVG

En el proyecto realizado por Natalia de León [1] se detalla el uso e implementación de los sensores de posicionamiento DWM1001 de la empresa Qorvo [2], los cuales están integrados en los módulos MDEK1001 como se muestra en la Figura 1.

Figura 1.
Módulo MDEK1001.

Nota. Adaptada de [3].

En el proyecto se llevó a cabo lo que es el estudio RTLS en donde se obtuvieron resultados relacionados a exactitud, precisión, montaje en Robot-Rover UVG utilizando una tasa de transferencia de datos de 10 Hz. A su vez se exploró la aplicación móvil Decawave DRTLS Manager para Android, obteniendo posicionamiento mediante el protocolo TLV. Se realizó la comparación con instrumentos de medición como un metro y el sistema de cámaras Optitrack en el laboratorio Robotat ubicado en la Universidad del Valle de Guatemala. La evolución del proyecto comenzó desde pruebas con cuatro sensores en configuración de anclas y una etiqueta o *tag* para poder medir posición en una figura rectangular formada por las anclas (que son los módulos MDEK1001 configurado en un punto fijo).

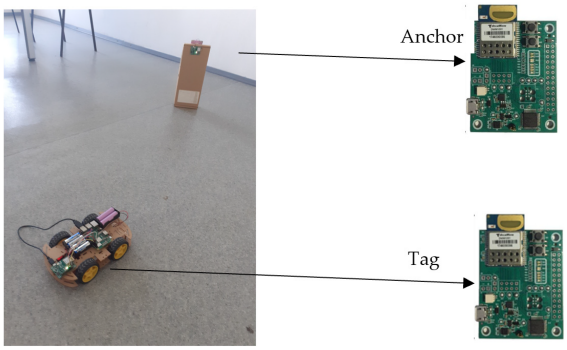
En cuanto a geometrías irregulares se realizaron pruebas en ambientes externos haciendo uso de herramientas de la aplicación para posicionamiento automático de los sensores anclados y en lugares cerrados con geometría definida siendo un rectángulo y aumentando el número de sensores a seis con una etiqueta. Este sistema de posicionamiento fue integrado a la plataforma ROS para poder trabajar en conjunto con la captura de movimiento de las seis cámaras Optitrack y realizar las comparaciones pertinentes de posición obteniendo porcentajes de error en las coordenadas (x,y) de 2.23 % y 3.78 % respectivamente para un área de 4000 x 5000 mm.

2.2. Uso de tecnología de banda ultra-ancha para control de ubicación de un carro

El proyecto elaborado por Mischie Septimu y Alina Munteanu [4] consistió en buscar una solución de posicionamiento de un vehículo que hiciese uso de GPS o sensores inerciales debido a las fallas presentadas en cuanto a la precisión por lo cual se optó por hacer uso de la tecnología de ancho de banda ultra-ancha, utilizando los módulos MDEK1001 en la Figura 1. En este proyecto se propuso el uso de cuatro anclas y una etiqueta en donde la información recibida del *tag* se procesa por medio del protocolo UART en la microcomputadora Raspberry Pi que a su vez transmite información a través de internet a la computadora. Por otra parte la propia etiqueta envía información por medio de BLE hacia un teléfono inteligente. En la Figura 2 se muestra un despliegue simple del uso de estos módulos. Dentro del estudio, se realizaron mediciones estáticas con desviaciones estándar de hasta $\pm$ 5.9 cm en el eje x

y en el eje y hasta $\pm$ 2.4 cm donde se comprobó que se encuentra dentro lo establecido por el fabricante con precisión de $\pm$ 10 cm. Además se obtuvo un factor de calidad qf del 75 % que devuelve los módulos UWB por cada medición.

Figura 2.
Entorno de conexión de anclas y etiquetas.



Nota. Adaptada de [4].

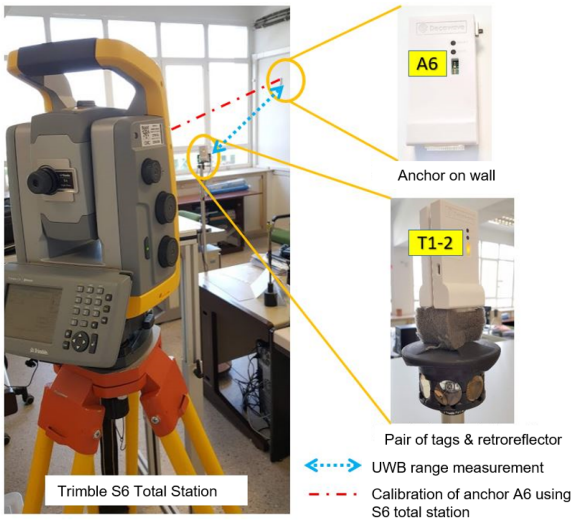
Luego en pruebas dinámicas se colocó una coordenada inicial y una final para saber la exactitud de la ubicación al alcanzar dicha coordenada. En algunos casos los valores llegaban a exceder la coordenada establecida o estar por debajo del destino, esto para las pruebas del carro en avance y retroceso en donde el eje x tuvo resultados dispersos mientras que el eje y tuvo una dispersión uniforme. También se discute que en algunos casos se excede la precisión de $\pm$ 10 cm pero no afectó a la llegada del carro a su destino o cercano a este. Este estudio recomienda la implementación de sensores giroscópicos para determinar la orientación del carro y con ello distintas formas de poder controlar el carro.

2.3. Mejora de la precisión de posicionamiento del módulo MDEK1001 al tener múltiples rangos y en interiores; implementación de EKF

En el estudio realizado por Antonio R. Jimenez y Fernando Seco [5] se presentó una propuesta para poder calibrar los sensores en un ambiente LOS en una oficina. Se hizo uso de un instrumento de medición con $\pm$ 1 mm de precisión siendo la estación Trimble S6 en conjunto con un retroreflector con dos módulos MDEK1001 montados encima de este como

se muestra en la Figura 3, con la finalidad de tener un *ground truth* (GT) de la información de los dos módulos configurados en comunicación BLE dado que fue considerado como el método de comunicación más eficiente. Se hicieron mediciones en paralelo del Trimble S6 en modo de seguimiento y los módulos UWB en donde uno tenía 4 anclas y el otro igual, formando dos redes a fin de tener múltiples rangos y con ello elaborar un filtro de Kalman Extendido (EKF por sus siglas en inglés).

Figura 3.
Montaje de experimento con ground truth.



Nota. Adaptada de [5].

En la elaboración del filtro EKF se plantearon seis variables de estado siendo posición en ejes x y z, adicionalmente las velocidades en un vector columna, para ello se planteó el vector como una función sumado con una variable $g(a)$ que contiene el ruido. Partiendo de dicha ecuación se planteó una matriz de transición y con ello una matriz Q. Posteriormente también se hizo uso de una matriz H la cual funcionó para medición de las posiciones de los tres ejes. Con ello en mano se realizaron gráficos en conjunto con la estación Trimble S6 y para ambas redes correspondientes a cada etiqueta y con los muestreos se pudo implementar el EKF con desviación estándar menor a 20 cm. Visualmente los resultados obtenidos muestran una trayectoria "limpia"que se apega al GT en el ambiente LOS.

En otro experimento realizado dentro de un apartamento se definió como un entorno NLOS en el cual hay obstáculos como paredes, muebles o bien elementos metálicos, entre otros objetos que dificultan la comunicación del sistema UWB. Para este experimento se

consideró el uso de unidades de medición inercial colocadas en los pies como GT que, de acuerdo con el autor, no es fiable pero es una opción útil al no poder usar la estación Trimble S6 en dicho ambiente. Los módulos UWB colocados en la cabeza se usaron para medir trayectorias dentro del apartamento en conjunto con las IMU y se encontró una desviación estándar de $\pm$ 0.6 m y que los rangos mayores a 2 m empiezan a enviar valores atípicos por el hecho de tener que atravesar paredes, objetos o incluso personas. Se aplicó el filtro EKF, mostrando resultados con ruido bajo o valores atípicos con mejor precisión. El estudio concluye que los módulos MDEK1001 son más exactos y precisos en ambientes de línea de vista haciendo uso de un filtro EKF para dispositivos con poca cobertura satelital o sistema de posicionamiento, esto lo hace escalable para soluciones corporativas.

CAPÍTULO 3

Justificación

En distintos estudios de sensores de banda ultra ancha se ha demostrado que son una opción de útil para localización en ambientes interiores en condiciones LOS o NLOS para estudiar movimiento en un espacio. Existen otras opciones de posicionamiento en interiores como sistemas basados en sensores inerciales, visión por computadora, ondas de sonido, entre otras.

En el caso de este estudio se usará un sistema basado en radiofrecuencia de banda ultra ancha. El sistema tiene la capacidad de funcionar en tiempo real con un mínimo de cuatro anclas que generan rangos o distancias respecto de una etiqueta a una frecuencia de 10 Hz como mínimo. El sistema provee la solución de posicionamiento, sin embargo no tiene orientación alguna que es útil para poder realizar trabajos relacionados con localización en el área de robótica, control, y navegación de partículas.

En trabajos de graduación previamente hechos en la Universidad del Valle de Guatemala y estudios con los módulos MDEK1001, la obtención de datos es poco flexible dado que los datos de posicionamiento se pueden obtener de distintos métodos como en celular o por medio de la computadora. La desventaja es que no posee un formato de fácil obtención o que requiere otros métodos para extraer datos en tiempo real. Este problema se da de modo que para obtener información se extrae de la aplicación móvil DRTLS de Qorvo y una forma descentralizada de realizarlo es por medio de sistemas embebidos que puedan

solicitar la información de localización en este caso. De dicha forma se pueden implementar aplicaciones para distintos campos de estudio descritos previamente.

El presente trabajo de graduación tiene como propósito facilitar obtención de datos mediante wifi con el protocolo TCP, procesamiento de los datos al aplicar filtros complementarios para mejorar precisión y uso de homografía proyectiva para mejorar exactitud. Dentro de este trabajo también se considera la implementación de orientación para poder tener un sistema de navegación local y funcional.

CAPÍTULO 4

Objetivos

4.1. Objetivo general

- Implementar un sistema de navegación local en un entorno de dos dimensiones con módulos UWB.

4.2. Objetivos específicos

- Integrar una unidad de medición inercial en el Sistema de Navegación Local.
- Implementar filtros y homografía para mejorar la precisión y exactitud del posicionamiento en dos dimensiones.
- Realizar experimentos con el Sistema de Navegación Local en condiciones controladas de Línea de Vista LOS en dos dimensiones.

CAPÍTULO 5

Alcance

Este trabajo se enfoca en desarrollar un sistema de navegación mejorando las capacidades de exactitud y precisión mediante fusión de sensores en ambiente controlado de LOS, que es un ambiente libre de obstáculos que causen interferencias. Se implementa la combinación de la tecnología UWB y sensores inerciales como una IMU usando el MPU9250, con el propósito comparar y validar la mejora de precisión y exactitud. Esta comparación se realiza respecto de otro sistema que nos da información de referencia o *Ground Truth* que es Optitrack. La plataforma principal para obtener datos es el microcontrolador ESP32, mientras que por medio wifi TCP y Python se recibe la información. Dicha información se procesa utilizando MATLAB para evaluar y visualizar los resultados.

Este trabajo logra la integración, obtención y procesamiento de datos mediante algoritmos de fusión de sensores, como filtro complementario normal y filtro complementario Butterworth. Un componente importante de este trabajo es la aplicación de transformación mediante homografía proyectiva que corrige los datos crudos de UWB para verlos representados en el marco de referencia Optitrack como GT. Este proceso ha permitido comparar ambos sistemas y validar que la información obtenida del sistema UWB puede ser mejorada al complementarla con otro sensor.

De manera adicional, se ha logrado la implementación en tiempo real de obtención de datos mediante protocolo wifi TCP, lo que permite que el sistema pueda funcionar en dis-

tintas aplicaciones dinámicas que deban ser monitoreadas en tiempo real o la información pueda ser procesada posteriormente.

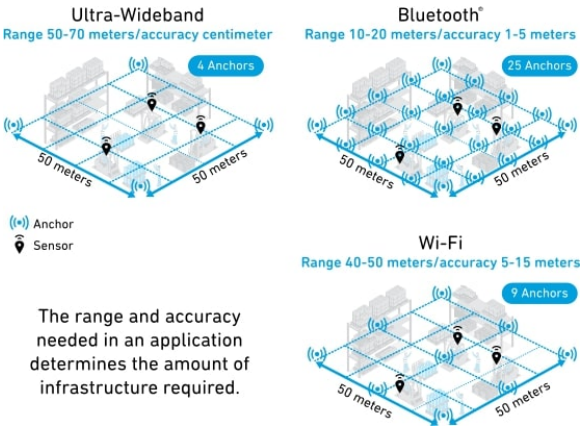
CAPÍTULO 6

Marco teórico

6.1. Tecnología ultra wide band

La tecnología UWB de acuerdo con la Comisión Federal de Comunicaciones (FCC) es cualquier dispositivo que se encuentre por encima de los 1.5 GHz del espectro de transmisión por radiofrecuencia [6]. El estándar IEEE 802.15.4a/z utiliza en esta tecnología para micro-localización como propuesta de solución a sistemas de posicionamiento como BLE, GPS o wifi con precisión en rangos de 1 hasta 15 metros, mientras que la tecnología UWB posee un rango de 5 a 10 cm como se observa en la Figura 4.

Figura 4.
Precisión en distintos sistemas de localización.

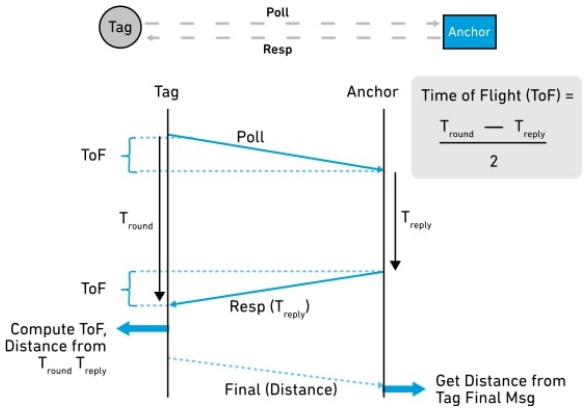


Nota. Adaptada de [7].

Una vez observada la comparación, se tiene que este sistema es más preciso y exacto respecto de los sistemas mencionados anteriormente. La importancia de estos módulos es que pueden ser útiles para sistemas de localización en interiores como la seguridad de hogares, industria automotriz, robótica, deportes y aplicaciones industriales para detección de objetos o personas en un entorno donde la precisión y exactitud sea un factor importante para obtención y análisis de datos. Existen varias ventajas que podemos encontrar es la confiabilidad ante interferencias en entornos cerrados, la baja latencia para objetos en movimiento siendo de 1000 actualizaciones por segundo o bien 1 mili-segundo para entregar datos de localización en tiempo real[7] . Cada sistema de localización posee ventajas y desventajas de acuerdo con la aplicación que se implemente ya sea lo escalable, precio, duración, entre otras. Los sistemas embebidos de UWB se pueden capturar hasta 27 Mbps y a futuro se pretende mejorar los estándares para aumentar la velocidad de transmisión.

La manera en la que se emplea la tecnología UWB es por medio de anclas y etiquetas, siendo que las anclas son fijadas en una posición determinada y las etiquetas son agentes móviles que recopilan y envían información a un canal deseado. En el caso del funcionamiento de estos dispositivos, consiste en que el ancla envía una señal a la etiqueta y el tiempo que se tarda en viajar y regresar como una respuesta se denomina "Tiempo de Vuelo"ToF y esta dinámica se denomina rango bidireccional TWR que, como mencionado previamente se toma un tiempo de respuesta y se multiplica por la velocidad de la luz como se observa en la Figura 5. Este sistema se puede implementar tanto para entornos bidimensionales a tridimensionales.

Figura 5.
Tiempo de vuelo (ToF) con rango bidireccional (TWR).



Nota. Adaptada de [7].

Existen otras topologías de diseño como TDoA en el cual las anclas envían información constante del tiempo y localización a una computadora para calcular diferencias de tiempo de vuelo al tener una etiqueta enviando señal de parpadeo constantemente para determinar su localización en 2D y 3D y a su vez envían señales por medio de radio-frecuencia para sincronización de las anclas. Por otro lado tenemos el TDoA Inverso en el cual las anclas emiten señales de parpadeo y la etiqueta es la que envía la información a una computadora por distintos medios de comunicación. Finalmente está el PDoA que utilizando dos dispositivos UWB puede transmitir posicionamiento y una fase o angulo de uno respecto al otro con método de transmisión TWR. En la Figura 6 podemos observar la clasificación y ventajas de las distintas topologías de comunicación, así como sus aplicaciones e información útil.

Figura 6.
Topologías de comunicación UWB.

Topology Design	TWR (for Ranging & Secure Bubble)	TWR (w/Tag Data Backhaul or UWB Listeners)	TDoA	Reverse TDoA	PDoA (w/TWR)
Main applications	**Secure bubble & access control**	**General purpose RTLS***	**Asset & people RTLS***	**Navigation**	**General purpose & peer-to-peer tracking**
Tag density	Hundreds	Hundreds	Thousands	Unlimited	Hundreds
Battery lifetime - tag side (**)	1-12 months	1-12 months	5 years +	Days/week or more	1-12 months
Location engine (software) runs...	In the tag or anchor	In the tag or anchor	In a master anchor, or in a server of cloud	In the tag	In the tag or anchor
RTLS* scalability	N/A	Low/medium	High	High	Low/medium
Ideal for...	Security (payment, identification, access control) & safety	2D & 3D RTLS* with seamless installation & scalability. Access control & secure transactions.	2D & 3D asset & people RTLS* in large-scale deployments	Drones, robotics, sport analytics, RTLS* with battery powered infrastructure	Access control, P2P, tracking and follow-me drones & robots
Benefits	• Defeats relay attacks • Compliance with CCC specifications	• Seamless implementation • No need for anchor syncing/backhaul channel)	• Very low power tag • High amount of tags can be used	• 'GPS-like' • Infinite amount of tags can be used • Low-power anchors (can be battery-powered)	• Peer-to-peer localization • Polar coordinates • Spherical coordinates possible with 2 instances of IC's/module
Applications	• Distance measurements • Door locks physical access control • Car keyless entry • Car immobilizers	• 2D/3D asset & people tracking • Industry 4.0 • Warehouse logistics • Healthcare • Retail	• 2D/3D asset & people tracking • Industry 4.0 • Warehouse & logistics • Healthcare • Agriculture	• 2D/3D navigation • Robotics • Valet parking • Lone worker protection	• 'Follow me' robots, drones, etc. • Platooning (trucks) • Door locks & access control

*RTLS Real Time Location Systems
** Indicative - actual bettery lifetime figures depend on exact RF & RTLS configuration and on battery type & capacity.

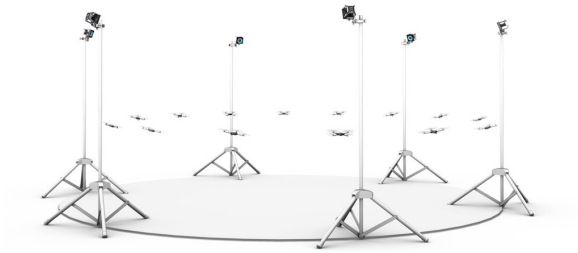
Nota. Adaptada de [7].

6.2. Sistema de captura de movimiento con cámaras Optitrack

El sistema MoCap de Optitrack [8] consiste principalmente en cámaras de captura de movimiento con tecnología infrarroja integrada de 850 nm para detectar marcadores o bien

esferas con pintura reflectiva y así determinar posición y orientación de dicho marcador. Para un entorno más avanzado y mejor precisión se utilizan de 6 a 8 cámaras con la finalidad de tener mayor cobertura en cuanto a la captura de movimiento. En general sistema cuenta con una precisión de $\pm$ 0.2 mm con una latencia menor a 10 ms y rotación con $\pm$ 0.1 grados para el caso de la serie Prime. En la Figura 8, se aprecia un montaje de 6 cámaras PrimeX 41.

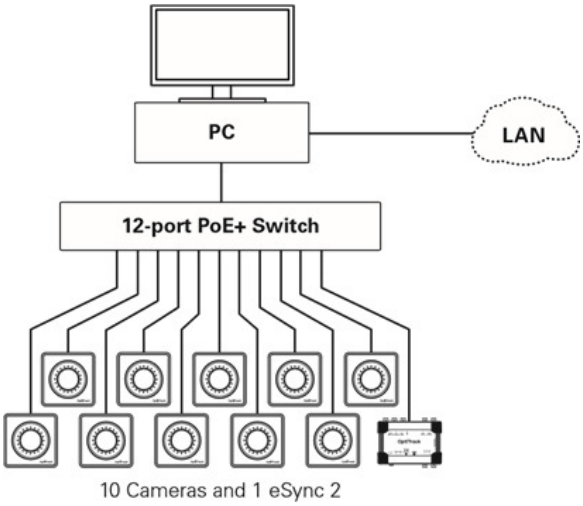
Figura 7.
Montaje de 6 cámaras PrimeX 41.



Nota. Adaptada de [9].

Estos sistemas se caracterizan por la facilidad de uso en cuanto a su interfaz y conectividad con aplicaciones como Matlab, Motion Builder, NatNet, Unity, ROS, entre otras; además, se pueden colocar en distintas disposiciones y sincronizarlas de modo que lo hace flexible a distintas aplicaciones. El método de comunicación es por medio de cable Cat6 ethernet; la información se transmite de las cámaras a una computadora que bien puede ser configurada para publicar datos a una red IP o localmente. Las aplicaciones del sistema de captura Optitrack son variadas, tal como la producción virtual, el análisis biomecánico, la realidad virtual, las animaciones y la robótica. En la Figura 8, se encuentra la forma en la que se conecta un sistema de cámaras PrimeX de Optitrack para menos de 11 cámaras.

Figura 8.
Esquema de conexión cámaras PrimeX de Optitrack.

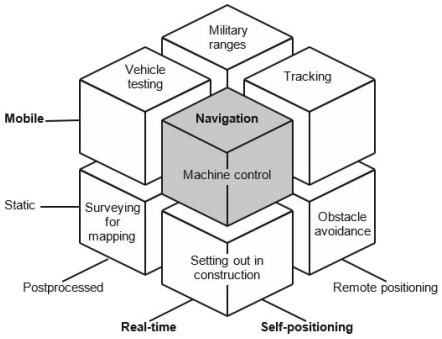


Nota. Adaptada de [8].

6.3. Sistema de navegación local

De acuerdo con Grooves [10], la navegación se define como la planificación de movimiento de un cuerpo rígido en un espacio tomando en cuenta la velocidad y aceleración respecto de un marco inercial que se denomina también, la ciencia de navegación. Grooves plantea que la planificación de movimiento consiste en mover algún objeto de un punto a otro manteniendo una ruta y que pueda ser capaz de evadir obstáculos, siendo que, para determinar una posición se puede realizar de manera manual o automática en el dominio del tiempo. En la Figura 8 podemos observar la distribución de las aplicaciones de posicionamiento.

Figura 9.
Aplicaciones de sistemas de navegación.



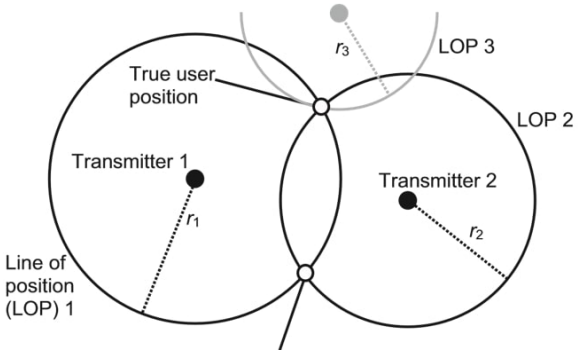
Nota. Adaptada de [10].

Para un sistema de navegación se debe tener un contexto, siendo este el ambiente en el cual dicho sistema opera y que puede alterar o no la información de posicionamiento de un objeto o vehículo. Con dicha información, se sabe que para tener un sistema eficiente, se debe acoplar a un contexto, dado que pueden haber obstáculos que impidan que ciertas señales como radiofrecuencia por ejemplo, no puedan penetrar dichos obstáculos, o bien en otro contexto como ambientes marítimos es útil el sonar para determinar la altura y objetos que puedan ser un potencial peligro o bien una forma de ubicarse. En la aviación se tiene la instrumentación para saber posicionamiento, para ello debe haber alternativas para saber posición en caso algún instrumento falle, así se puede tener un estimado o incluso la misma información para tener noción de la ubicación en el espacio.

Si bien se sabe, existen múltiples métodos de localización, ya sea por GPS, IMU, UWB, wifi, sistemas Infrarrojos, RFID o por eco que pueden dar un estimado de posición respecto a un punto y estos se pueden integrar y mejorar su precisión con algoritmos de estimación para corregir y proveer una solución a un sistema de navegación que dependa de múltiples sensores. Se debe tomar en cuenta que para la elaboración de una solución de navegación se debe detectar errores de falla, de manera que deba ser lo más seguro posible ya que si se desea implementar en situaciones críticas, es importante considerar que pueden existir fallos.

En el área de posicionamiento por radiofrecuencia existen los rangos que son utilizados para dichos sistemas con al menos 3 rangos para arriba. En la Figura 10 se aprecia la existencia de 3 transmisores que a lo largo de su radio detectan un posicionamiento con dos alternativas pero gracias al tercer dispositivo se puede obtener una posición real o cercana.

Figura 10.
Posicionamiento por tres rangos.



Nota. Adaptada de [10].

6.4. Aplicaciones de posicionamiento en interiores

Según Grooves [10], existen categorías de posicionamiento exterior e interior, como navegación peatonal, vehicular para pruebas, vallas de seguridad virtual, tours virtuales en función de posición, etc. También propone que en el futuro habrán tendencias sobre estas tecnologías de navegación como mejoras en sensores siendo la integración de estos, disminución de tamaño y coste; otra mejora es en la comunicación dado que está en constante revisión y mejora, mapeos 3D de topología de navegación, integración de sensores y procesamiento de datos.

6.5. Filtro de Kalman

De acuerdo con Friedman [11], Rudolf E. Kalman fue el responsable de desarrollar un algoritmo capaz de encontrar estados más precisos de un sistema dinámico. El filtro elaborado por Kalman se limitó para sistemas lineales como producto de las aplicaciones siendo funciones de transferencia o respuestas impulsionales pero dado que evolucionaron los sistemas dinámicos, se tomó en cuenta que no todos son lineales por lo que el filtro eventualmente se expresó en ecuaciones diferenciales para aplicarlo a sistemas no lineales y ampliar la capacidad de estudio utilizando este filtro dando datos más cercanos a la realidad. Estas variaciones del filtro se conocen como Filtro de Kalman Extendido, Filtro de Kalman Unscented. A continuación se describe cómo se forma el Filtro de Kalman.

$$\dot{x} = Ax + Bu + Fv \tag{1}$$

$$y = Cx + w \tag{2}$$

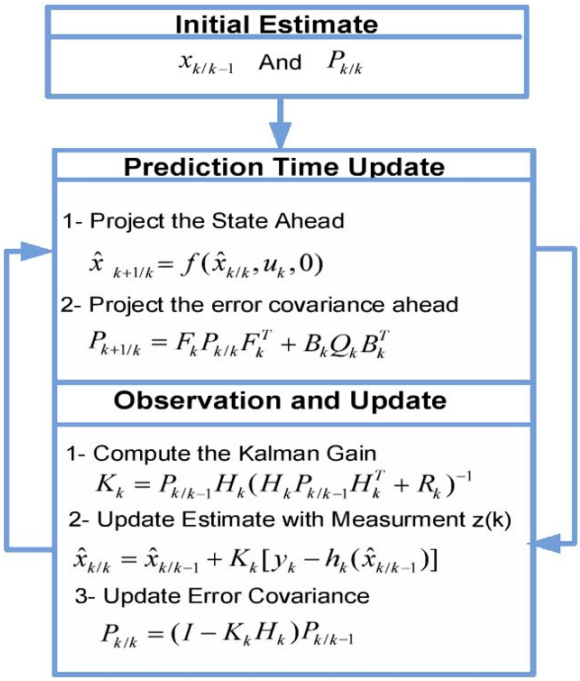
Para las cuales u representa el control v y w representan el ruido blanco o bien varianzas. Para resolver el problema se plantea una ecuación de observador de estado como se muestra a continuación.

$$\dot{\hat{x}} = A\hat{x} + Bu + K(y - C\hat{x}) \tag{3}$$

- $\dot{\hat{x}}$ representa la tasa de cambio del estado estimado
- $\hat{x}$ es el estado estimado
- A y B son las matrices del sistema dinámico
- u es el vector de control
- K es la ganancia de Kalman que disminuye el error estimado en la covarianza
- y es el vector de referencia.
- C es la matriz que mapea el espacio de estados dentro del espacio observado
- $y - C\hat{x}$ es la diferencia que mide la discrepancia de los datos en predicción con las mediciones actuales

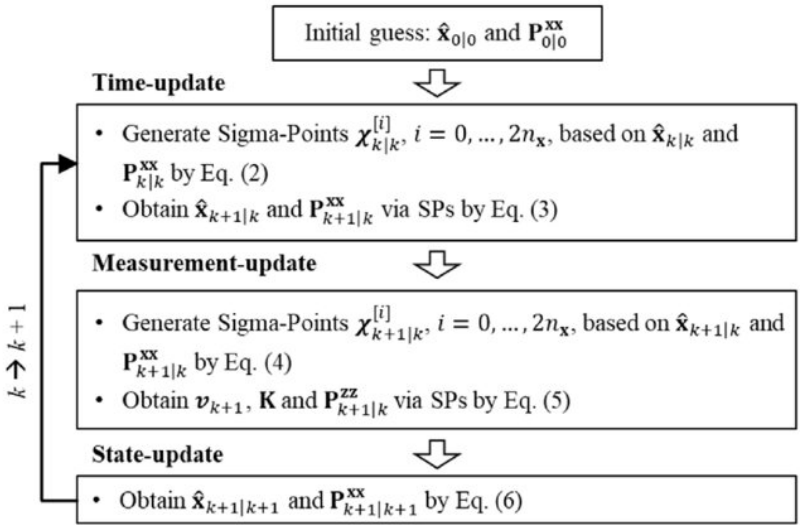
Como se mencionó previamente, el filtro está diseñado para casos lineales y por ello se han desarrollado variantes como el EKF o Filtro de Kalman Extendido y el UKF o Filtro de Kalman Unscented. Ambos se utilizan para sistemas dinámicos no lineales. Para el EKF se tiene la ventaja que al ser no lineal se pueden realizar linealizaciones para obtener varianzas y covarianzas, además es variante en el tiempo y el uso de observador es útil para tener datos estimados muy cerca de un dato real, sin embargo si el estimado inicial diverge del estado original o real, puede llegar a causar problemas en la optimización. El filtro UKF por otro lado es más utilizado para manejo de no linealidades, propagación por puntos sigma y predicción de estados. En la Figura 11 se puede observar el algoritmo para el cálculo del EKF y en la siguiente Figura 12 de igual manera con el UKF.

Figura 11.
Algoritmo de cálculo EKF.



Nota. Adaptada de [12].

Figura 12.
Algoritmo de cálculo UKF.



Nota. Adaptada de [13].

6.6. Filtro de Savitzky-Golay

El filtro de Savitzky-Golay de acuerdo con Schafer [14] se da cuando Abraham Savitzky y Marcel J.E. Golay realizaron un método para obtener información suavizada utilizando aproximaciones polinomiales con la finalidad de reducir ruido en cualquier arreglo de datos. A continuación se detalla la ecuación.

$$y'_i = \sum_{k=-m}^{m} c_k y_{i+k}$$

Donde y'_i es el valor resultante de la derivada en la posición i, y y_{i+k} siendo los datos reales de un arreglo en el rango $i - m$ a $i + m$.

Los coeficientes c_k se obtienen con la finalidad de poder reducir el error cuadrático obtenidos de los polinomios en ajuste tomando en cuenta los datos reales con lo que se conoce una ventana de datos o puntos a graficar entre valor previo y posterior. Para lograr esto, se resuelve un SEL utilizando el método de mínimos cuadrados. A continuación se muestran los pasos para el cálculo de coeficientes:

1. **Obtención de matriz de Vandermonde**: la matriz A es una matriz de Vandermonde modificada, para la cual cada fila representa un punto en un rango o ventana de datos y las potencias aumentan conforme las columnas se extienden $n + 1$ empezando en $n = 0$ hasta n.

$$A = \begin{bmatrix} 1 & -m & (-m)^2 & \cdots & (-m)^n \\ 1 & -m+1 & (-m+1)^2 & \cdots & (-m+1)^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 0 & 0^2 & \cdots & 0^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & m & m^2 & \cdots & m^n \end{bmatrix}$$

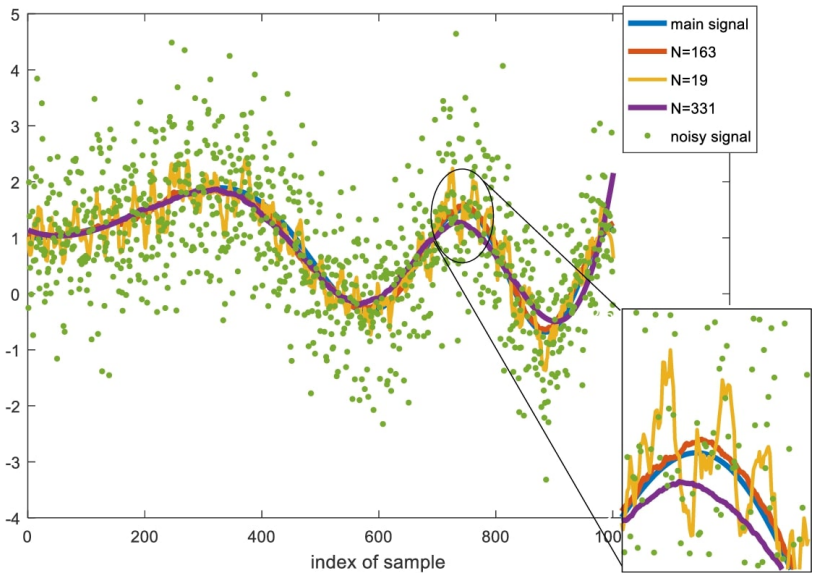
2. **Obtención de la pseudoinversa de la matriz** A: luego de obtener A, se debe utilizar para obtener los coeficientes c_k que son calculados al multiplicar **A** con un vector **b**

que son los valores deseados a partir de la derivación.

$$\mathbf{c} = (A^T A)^{-1} A^T \mathbf{b}$$

Finalmente, haciendo uso de este filtro se pueden obtener gráficos como se muestra en la Figura 13 que consiste en la selección de ventana para reconstruir una señal con ruido y acoplarla a la señal original.

Figura 13.
Reconstrucción de señal con filtro SG.



Nota. Adaptada de [15].

6.7. Homografía

La homografía consiste en realizar una transformación proyectiva de modo que se pueden relacionar planos distintos por medio de correspondencia. De dicho modo se pueden mapear puntos de un plano a otro, de modo que se colocan puntos fijos y móviles (que son los que se desean mapear). La transformación por homografía es comúnmente usada en visión por computadora para calibrar parámetros de cámara que serían los puntos fijos con los del entorno o lo que observa la cámara y lo hace en un espacio tridimensional ya que se puede calcular desplazamientos en los tres ejes teniendo una imagen plana o 2D. La homografía se

representa utilizando una matriz 3×3 y se da por la siguiente ecuación:

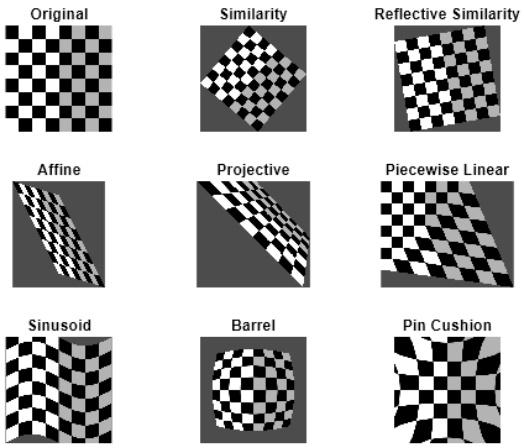
$$x' = Hx \tag{4}$$

Donde:

- x y x' representan los vectores de coordenadas en los planos que se corresponden.
- H representa la matriz de homografía con dimensiones 3×3, esta matriz es la encargada de mapear cómo los puntos se representan de un plano a otro

En términos matemáticos, la matriz de homografía puede obtener transformaciones de pose, es decir, puede transformar las traslaciones, rotaciones y a su vez puede escalar o ajustar al otro plano. Para poder obtener la matriz, se debe calcular como mínimo 4 pares de puntos correspondientes, es decir puntos en común entre dos planos de coordenadas. Ahora bien, como los puntos se relacionan mediante transformación proyectiva, se tiene que la matriz H se calcula al resolver un SEL. Aún así, también es necesario utilizar métodos que eviten la influencia de puntos atípicos que causen transformaciones erróneas por correspondencias de dichos puntos [16]. En la Figura 14 se muestran los distintos tipos de homografía que existen y cómo se proyectan como en la que tiene la etiqueta *Original*.

Figura 14.
Tipos de homografía.



Nota. Adaptada de [17].

La transformación por homografía descrita previamente se detalla a continuación:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{5}$$

donde $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ representa las coordenadas homogéneas del punto en el plano original, y $\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix}$ son las coordenadas transformadas en el plano destino. Para obtener las coordenadas cartesianas finales (x', y'), se realiza una normalización dividiendo por w' como factor de escalamiento:

$$x'_{\text{cart}} = \frac{x'}{w'}, \quad y'_{\text{cart}} = \frac{y'}{w'}.$$

6.8. Filtro complementario

El filtro complementario es una herramienta en el dominio de frecuencia que se utiliza para fusión de sensores o específicamente fusión de datos de sensores. Existen ejemplos comunes como acelerómetros fusionados con giroscopios en donde la finalidad es obtener una estimación para medir variables de orientación o incluso posición mitigando los defectos del ruido estocástico y obteniendo una mejor estimación de la información obtenida. La ventaja de estos filtros es la fácil implementación en aplicaciones relacionadas a robótica, navegación, control, etc. [18]

Como se mencionó previamente el filtro tiene la capacidad de combinar las características dinámicas de dos o más sensores por ejemplo:

- Se aplica un filtro pasa bajas al acelerómetro, que es un sensor exteroceptivo, confiando en sus datos a bajas frecuencias, para este caso en la inclinación ocasionada por la gravedad.

- Se aplica un filtro pasa altas al giroscopio, que es un sensor propioceptivo, aprovechando su precisión a altas frecuencias para medir las tasas angulares sin tomar en cuenta perturbaciones externas.

La fusión de sensores se realiza mediante la siguiente ecuación:

$$\text{Estimación} = \alpha(\text{Ángulo Giroscopio}) + (1 - \alpha)(\text{Ángulo Acelerómetro}) \tag{6}$$

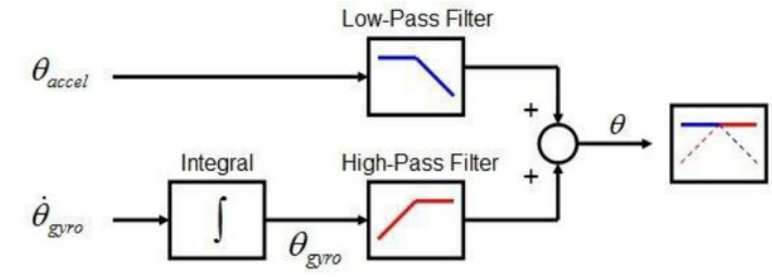
donde α es un parámetro o peso ajustable ($0 < \alpha < 1$) que puede distribuir las contribuciones de cada sensor, es decir en qué sensor se puede confiar más de acuerdo con los resultados. [19]. Los filtros complementarios son ideales para sistemas embebidos con recursos limitados o donde se requiere una implementación rápida. [20]. En la Figura 15 podemos observar la arquitectura de un filtro complementario normal, de modo que gráficamente muestra cómo funciona el ejemplo mostrado previamente.

Aplicaciones

El filtro complementario se puede utilizar en los siguientes ámbitos:

- Estabilización de drones
- Estimación de la orientación en dispositivos móviles como carros, drones, etc.
- Navegación y plataformas autónomas.

Figura 15.
Arquitectura de filtro complementario normal.



Nota. Adaptada de [21].

CAPÍTULO 7

Módulo MDEK1001

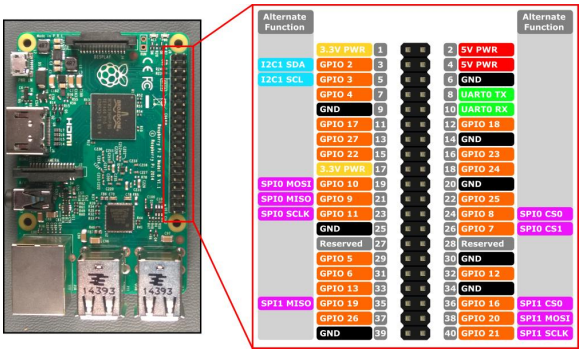
El módulo MDEK1001 es una placa de desarrollo elaborada originalmente por la empresa Decawave (ahora Qorvo) que consiste principalmente en el circuito integrado DWM1001 como transceptor y cuya tecnología se basa en la implementación de la norma IEEE 802.15.4-2011 sobre ancho de banda ultra ancha. Utiliza el chip nRF52832 que integra la información del DWM1001, la antena BLE y el acelerómetro STM LIS2DH12TR. La finalidad del DWM1001 es funcionar como un transceptor y comunicarse con otras antenas iguales de modo que se pueda saber la localización de un módulo específico.

Para comprender como funciona, se conectan como mínimo 4 módulos en configuración de ancla y con posiciones determinadas. Para poder detectar ubicación se debe tener un módulo configurado como etiqueta o *tag*, esta información se puede obtener mediante un ancla configurada como *Gateway* o ancla iniciadora y conectada a una computadora. La segunda forma de obtener datos es por medio de BLE conectando el celular utilizando la aplicación *DRTLS*, de modo que se puede observar un mapa con las anclas y la etiqueta. La aplicación cuenta con una terminal en la que se encuentran datos como coordenadas x,y,z y factor de calidad de medición que se actualizan a tasas desde 10 Hz (0.1s) hasta 0.017 Hz (1

min), la aplicación tiene una consola que registra dichos valores y exporta un archivo *.log*.

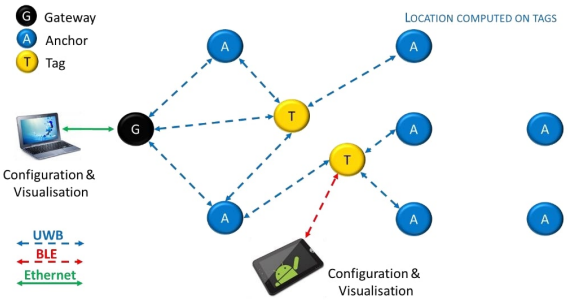
Dentro de la aplicación se puede colocar las configuraciones de las anclas y etiquetas, nombrarlos y ubicarlos. El módulo MDEK1001 tiene formas alternativas de comunicación por medio de los pines UART o SPI de la placa como se muestra en la Figura 16, los cuales tienen dos filas de 13 headers cada una para el montaje en la placa Raspberry Pi Model A. En la Figura 17 se muestra cómo funciona la comunicación entre módulos.

Figura 16.
Pines de Raspberry Pi model A para montar MDEK1001.



Nota. Adaptada de [2].

Figura 17.
Diagrama de comunicación de módulos MDEK1001-Dev.



Nota. Adaptada de [2].

7.1. Montaje en Robotat

En la Figura 1, se muestra la carcasa que resguarda los sensores, de modo que si no se usan los pines, se pueden usar los módulos MDEK1001 conectados mediante micro-usb dentro o fuera de la carcasa. Para montar estos módulos en los trípodes donde están montadas las

cámaras PrimeX 41, se realizaron distintas iteraciones de modo que fuese fácil de montar, sujetar y conectar; se instalaron 6 módulos en el entorno. Estos sujetadores se elaboraron utilizando impresión 3D con filamento TPU a modo de hacer una sola pieza.

En la Figura 18, se muestra el modelo CAD de la pieza con el espacio para montaje del módulo, sujeción al poste y redireccionamiento del cable micro-usb a un costado. En la Figura 19, se muestra el montaje y el funcionamiento del módulo en el laboratorio Robotat. Parte del montaje también consistió en conectar los módulos a una regleta USB que a su vez está conectada a un enchufe inteligente que se activa mediante wifi para encender los módulos al activar un script del servidor del laboratorio Robotat que habilita el enchufe.

Figura 18.
Sujetador MDEK1001.

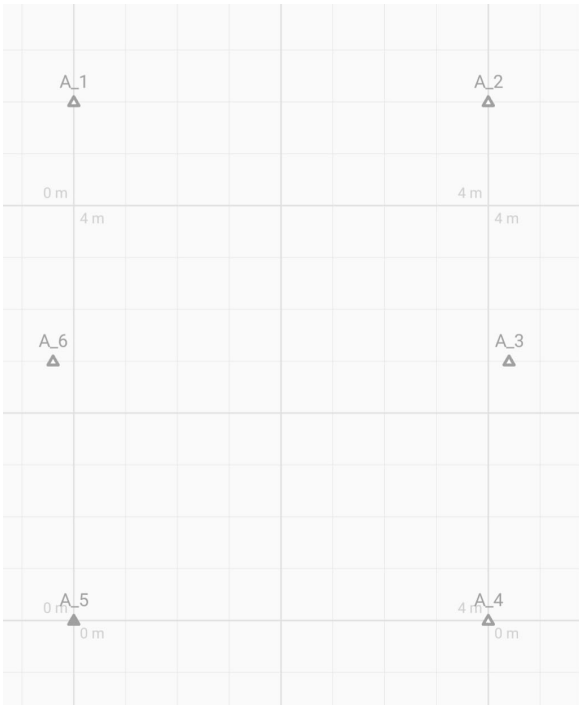
Nota. Elaboración propia.

Figura 19.
MDEK1001 montado en trípodes del laboratorio Robotat UVG.

Nota. Elaboración propia.

Una vez realizado el montaje, se colocó en la aplicación *DRTLS* la información de la ubicación de los sensores, puestos en la disposición que muestra la Figura 20, de modo que todos representan un ancla y *A5* es el ancla iniciadora o *Gateway* que es el único triángulo relleno en la figura. La plataforma del Robotat cuenta con dimensiones 4x5 metros, de modo que en el ancla *A1* hacia *A2* hay 4 metros y de *A5* hacia *A1* hay 5 metros. De dicha forma se colocaron los sensores, también se consideró el espacio que las anclas *A6* y *A3* con una longitud de separación desde la orilla de la plataforma hasta el trípode, siendo de 20.0 cm. Tomar en cuenta esto ayuda a definir donde se encuentran las anclas de manera adecuada, sin embargo no es algo exacto y para corregir las mediciones se emplea la calibración por homografía más adelante.

Figura 20.
Disposición de 6 módulos MDEK1001 en laboratorio Robotat.



Nota. Elaboración propia.

Un detalle importante a considerar en el sistema de coordenadas es que, A1 representa el origen del sistema de navegación UWB y el origen del sistema de captura Optitrack es en el centro de la plataforma. Esto se abarcará más adelante para igualar los sistemas de coordenadas.

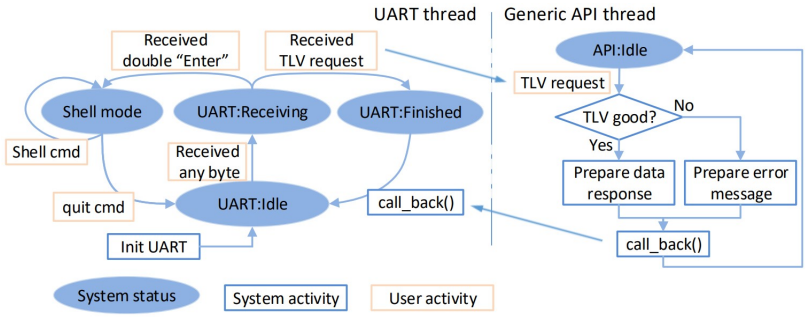
7.2. Conexión a MDEK1001

Como se detalló al inicio de este capítulo, el módulo MDEK1001 tiene distintos métodos de comunicación como se indica en la guía API del DWM1001. En este caso se optó el método de comunicación por UART que, de acuerdo con el API [2] se puede conectar el módulo a la computadora y con una terminal enviar doble *enter* o el código ASCII siendo *0x0D* para acceder al modo *Shell*. Este modo proporciona opciones para configurar el MDEK1001 utilizando comandos ASCII.

Por otra parte se puede realizar solicitudes mediante el formato TLV, en la cual no hay

que enviar doble *enter* para acceder sino los valores HEX para solicitar información específica al DWM1001. Los caracteres ASCII que envían más bytes de datos comparado con el formato TLV que se solicita con HEX y se recibe información en formato HEX directamente. En la Figura 21 se muestra el esquema de comunicación descrito anteriormente para solicitar información al DWM1001.

Figura 21.
Esquema de comunicación UWB para modo Shell o TLV.



Nota. Adaptada de [2].

Con dicho formato se realizaron pruebas utilizando la terminal serial Hercules. Para ello se conectó el MDEK1001 a la computadora por medio de micro-usb y en la terminal se configuró modo ASCII, el puerto al cual se conectó, el formato de línea para este caso CR o *Carriage Return* y la tasa de baudios siendo 115200 para el DWM1001 de acuerdo con el API. Para realizar pruebas se escribió y enviaron dos *enter* para acceder al modo Shell para explorar las opciones, luego se solicitó las coordenadas *x,y,z* enviando el comando *apg* y devuelve el mensaje *x:0 y:0 z:0*. Una aclaración importante es que al momento de realizar la solicitud no había ninguna red montada y entregaba ceros como localización. Posterior a esta prueba se modificaron las configuraciones de la terminal, de tal manera que, se pudiese leer valores hexadecimales para emplear la comunicación TLV; también se retiró el CR y se enviaron los valores 0x02 0x00 obteniendo como respuesta 0x40 0x01 0x00 0x41 0x0D seguido de 13 bytes para valores *x,y,z* (4 bytes por coordenada) y el último byte corresponde al factor de calidad de la medición.

Para comprender las respuestas, en la Figura 22 se muestra cómo se descompone la información recibida para posicionamiento. La ventaja de este método es que se envía una solicitud y se chequea si esta bien y se recibe de vuelta sin usar caracteres adicionales como es con ASCII, esto reduce la cantidad de mensajes enviados y recibidos por solicitud.

Figura 22.
Formato TLV para obtener posición.

Declaration:

TLV request	
Type	**Length**
0x02	0x00

Example:

TLV request	
Type	**Length**
0x02	0x00

TLV response								
Type	**Length**	**Value**	**Type**	**Length**	**Value**			
		err_code			**Position**			
					32 bit value in little endian is x coordinate in millimeters	32 bit value in little endian is y coordinate in millimeters	32 bit value in little endian is z coordinate in millimeters	8 bit value is quality factor in percents (0-100)
0x40	0x01	0x00	0x41	0x0D	0x79 0x00 0x00 0x00 0x32 0x00 0x00 0x00 0xfb 0x00 0x00 0x00 0x64			

Nota. Adaptada de [2].

7.3. Decodificación formato TLV

El módulo DWM1001 utiliza el formato TLV para la transmisión de información, donde cada elemento de datos contiene un campo que define el tipo, la longitud y el valor de los datos transmitidos. Para el caso de la obtención de la posición, el campo de valor contiene las coordenadas en milímetros representadas en un formato de 32 bits codificado en little-endian.

7.3.1. Formato little-endian

El formato little-endian es una forma de almacenar y transmitir datos, de modo que los bytes menos significativos LSB son los que se transmiten al principio, seguido de los bytes más significativos MSB. Dicho formato es utilizado en el envío y recepción de datos de los microcontroladores, en este caso el ESP32 quien solicita y decodifica y el DWM1001 quien envía el mensaje. También existe la parte contraria, siendo el formato Big-Endian donde se envían los MSB primero y al final los LSB.

A continuación se muestra un ejemplo, de un valor de 32 bits (4 bytes) en formato little-endian, siendo este 0x65 0x02 0x00 0x01 se decodifica como:

- 0x65: LSB

- 0x02: Segundo byte
- 0x00: Tercer byte
- 0x01: MSB

Para poder leer y decodificar luego este valor, se debe reorganizar al sentido contrario, es decir 0x01 0x00 0x02 0x65, leyéndolo de derecha a izquierda. Para calcular el valor o la conversión se debe hacer un desplazamiento de bytes, es decir, el LSB al estar al inicio no se desplaza, el segundo byte debe desplazarse 1 byte hacia atrás, el tercer byte se desplaza 2 bytes hacia atrás, de modo que se calcula cada valor independiente llevado a una posición de LSB para luego sumarlo y obtener el valor en entero o decimal. Para el caso del ejemplo mostrado arriba tenemos que 0X01 0x00 0x02 0x65 representa 104 milímetros al convertir esos valores HEX a decimal. A continuación se detalla de mejor manera cómo decodificar la información.

7.3.2. Ejemplo de decodificación

El DWM1001 responde al comando HEX 0x02 0x00 con las coordenadas de posición *x,y,z* en milímetros, y al final posee un factor de calidad *Quality Factor* o *QF* en el formato TLV La decodificación del valor de estas coordenadas se realiza al leer los datos como enteros de 32 bits (4 bytes) en little-endian. A continuación se muestra un ejemplo típico de respuesta TLV:

- Tipo: 0x40 indica que el tipo de dato es una respuesta de posición
- Longitud: 0x0D representa 13 bytes de datos
- Valores:
  - Coordenada x: 0x79 0x00 0x00 0x00 equivalente a 121 mm
  - Coordenada y: 0x32 0x00 0x00 0x00 equivalente a 50 mm
  - Coordenada z: 0xfb 0x00 0x00 0x00 equivalente a 251 mm
  - qf: 0x64 equivalente a 100 %

Esta información se observa mejor en la Figura 22

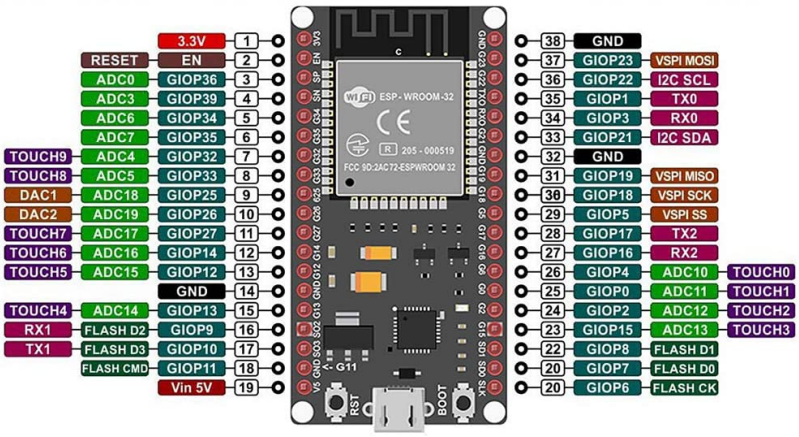
CAPÍTULO 8

Selección de microcontrolador

La selección de un microcontrolador es importante para cumplir las necesidades del proyecto en base a los sensores a montar. Para la implementación del sistema de navegación local se consideró tener posición usando UWB y orientación utilizando la unidad de medición inercial IMU; estos dos utilizan distintos métodos de comunicación, siendo UART para UWB e I2C para la IMU. Adicionalmente a lo largo del proyecto surgió la necesidad de tener un dispositivo móvil como un carro de pruebas para obtener mediciones en el tiempo en distintas posiciones. Para dichas necesidades se agregaron otros dispositivos como el HC05 y un puente H L298N para controlar los motores.

Dado que todos los dispositivos seleccionados funcionan a un voltaje lógico de 3.3V y otras necesidades del proyecto involucraron el uso de wifi para obtención de datos, se seleccionó el DOIT DevKit V1 ESP32-Wroom-32 dado que cumple con los requerimientos, especialmente con los canales UART siendo 3 para pinout, uno dedicado a comunicación mediante micro-usb con el ESP32 y otro libre; además se tiene la comunicación I2C para conexión con la IMU En la Figura 23 se muestra el microcontrolador elegido para obtención, envío y manejo de información.

Figura 23.
Pinout DOIT DevKit V1 ESP32.



Nota. Adaptada de [22].

8.1. Métodos de comunicación

8.1.1. Comunicación wifi y bluetooth

El ESP32 utiliza el protocolo IEEE 802.11 b/g/n/e/i, operando en la banda 2.4 GHz, cuyo ancho de banda oscila entre 20 MHz a 40 MHz. La modulación tiene OFDM teniendo tasa de datos hasta 150 Mbps estando en modo 802.11n y DSSS con tasas de 11 Mbps en el modo 802.11b. El chip puede soportar distintos modos de operación como Acces Point (AP), Station (STA), y AP+STA, de modo que hace que ESP32 pueda funcionar como un punto de acceso para poder conectarse a otros puntos de acceso de manera simultánea. Tiene características de seguridad WPA/WPA2, WAPT y tiene protocolos de red como IPv4, IPv6, TCP/UDP, HTTP, FTP, y SSL/TLS.

Dentro del ESP32 se incluye un módulo Bluetooth que funciona con Bluetooth Classic o Bluetooth Low Energy BLE El Bluetooth Classic es compatible con estándares como SPP, A2DP y AVRCP, esto permite que se pueda transmitir audio o comunicación de datos en serie. Para el BLE, el ESP32 puede conectarse hasta 9 dispositivos con tasa de datos de 2 Mbps con la codificación 2M PHY. Para que pueda funcionar en paralelo con wifi, existe un sistema de arbitraje el cual disminuye la interferencia en ambos radios cuando trabajan simultáneamente. El ESP32 tiene un rango de transmisión de 10 dBm y tiene una distancia

de hasta 100 metros en condiciones LOS [23].

8.1.2. Comunicación UART

De acuerdo con [24], el término UART se refiere a un método de comunicación utilizado en la periferia I/O de un microcontrolador. Este método, a diferencia de otros, no utiliza un reloj de sincronización; en su lugar, conecta dos dispositivos que se comunican a una velocidad previamente configurada en ambos, como por ejemplo, 115200 baudios en el caso del ESP32. Además, es bidireccional, lo que significa que ambos dispositivos pueden tanto enviar como recibir datos simultánea.

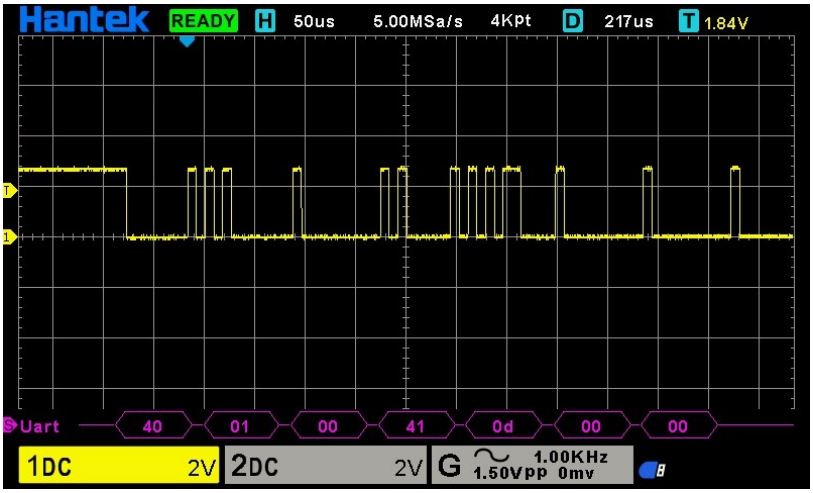
Para establecer la conexión, se debe conectar el pin TX (transmisor) de un dispositivo, en este caso el ESP32, con el pin RX (receptor) del otro dispositivo. Del mismo modo, el pin TX del segundo dispositivo se conecta al RX del ESP32, completando así la comunicación bidireccional.

La comunicación UART soporta diversas tasas de transmisión, que pueden variar desde 300 hasta 460400 baudios, siendo las más comunes las que oscilan entre 9600 y 115200 baudios.

El ESP32 dispone de tres canales UART, de los cuales están UART0, este es el que se conecta al puerto micro-USB y se utiliza para cargar el código en el dispositivo. Además, sus pines GPIO 1 y GPIO 3 corresponden a TX0 y RX0, respectivamente, permitiendo también la comunicación serial una vez cargado el programa en el ESP32. Para fines prácticos, se utilizó el canal UART2 del ESP32 para establecer la comunicación con el módulo MDEK1001 a una tasa de 115200 baudios. Para lograr esto, se escribió un script en el IDE de Arduino, configurando el Serial2 en el ESP32 para habilitar la comunicación con el módulo MDEK1001.

En este proceso, se enviaron dos bytes de acuerdo con el formato mostrado en la Figura 24. El módulo MDEK1001 respondió a esta solicitud, y los datos recibidos fueron procesados y mostrados en la terminal serial del ESP32. Para comprender los datos recibidos, se siguió el esquema de decodificación descrito previamente en la sección correspondiente a la interpretación del formato TLV.

Figura 24.
Lectura en osciloscopio modo decodificación UART.



Nota. Elaboración propia.

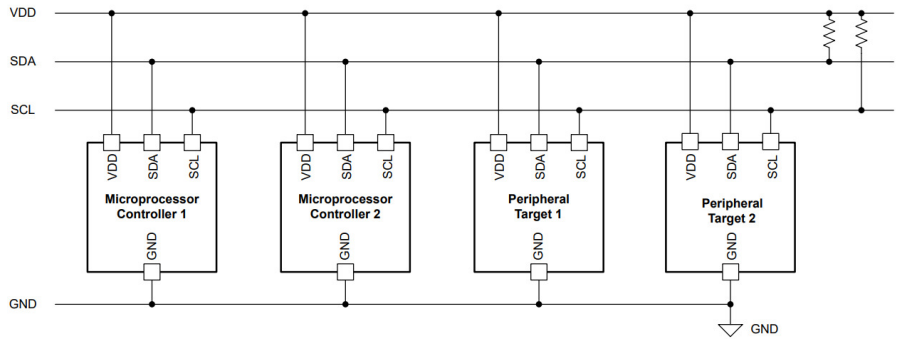
8.1.3. Comunicación I2C

I2C es un protocolo de comunicación que utiliza dos líneas o cables con resistencias en pullup cada una, siendo SDA que es la encargada de transmitir información y SCL que sincroniza uno o varios dispositivos ya que funciona como un reloj.

El término I2C quiere decir *Inter-Integrated Circuit*, el cual fue desarrollado en 1982 por Philips Semiconductor con la finalidad de conectar múltiples dispositivos periféricos de una manera sencilla. En 2006 ya no fue necesario una licencia para este método de comunicación y con ello múltiples compañías implementaron la comunicación en los dispositivos debido a la facilidad de conexión y como alternativa a los otros métodos. Este método tiene distintas velocidades desde 100 Kbps hasta 5 Mbps en modo ultra rápido.

La comunicación I2C es half-duplex, esto quiere decir que un controlador esta enviando información a uno o varios dispositivos pero no se puede enviar información de otro dispositivo media vez la línea SDA esté ocupada. Este tipo de comunicación puede soportar hasta 128 dispositivos conectados con distintas direcciones de escritura siendo 7 bits de para dirección y el octavo bit para leer o escribir, de modo que si se quiere escribir algo de un dispositivo a otro el último bit es el que determina ello al enviar un mensaje. En la Figura 25 se puede observar la conexión de dispositivos a las líneas de datos y comunicación.

Figura 25.
Conexión de dispositivos con comunicación I2C.



Nota. Adaptada de [25].

La elección de esta comunicación fue importante ya que el sensor de medición inercial elegido utiliza la comunicación SPI e I2C, sin embargo, para fines de conexión en una PCB se eligió la comunicación I2C ya que puede soportar desde 100 Kbps hasta 400 Kbps. Al tener un solo dispositivo conectado a un tipo de comunicación, facilita la obtención de datos con otros dispositivos en paralelo como MDEK1001 o HC-05 con otro canal UART.

CAPÍTULO 9

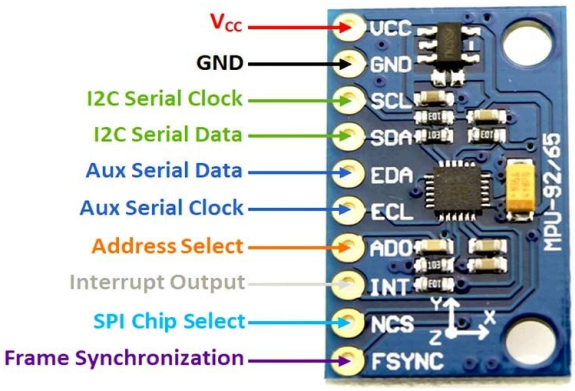
Unidad de medición inercial MPU9250

Un dispositivo importante para la fusión de sensores es una unidad de medición inercial IMU. La IMU seleccionada fue el MPU9250, de modo que tiene un sensor de 9 grados de libertad, actuando como acelerómetro, giroscopio y magnetómetro. Este módulo es útil para aplicaciones de navegación que requieran orientación en los 3 ejes, control de movimientos ya sea para un humano (en relojes inteligentes) o algo que se desee estudiar, celulares, realidad virtual, drones, entre otras aplicaciones.

El MPU9250 tiene dimensiones de 25.9 x 15.5 x 1 mm con los tres sensores en un chip integrado. Para el giroscopio, se tienen rangos de sensibilidad desde ± 250 deg/s hasta ± 2000 deg/s, útiles para medir rotaciones en los tres ejes o las velocidades, incluso aceleraciones angulares. Este puede tomar muestras desde 4 Hz hasta 8000 Hz. El acelerómetro tiene rangos de sensibilidad de ± 2 g hasta ± 16 g y se pueden ajustar de acuerdo con las necesidades del proyecto a realizar; además, la tasa de muestreo es de 4 Hz hasta 4000 Hz. El magnetómetro tiene rangos de sensibilidad de 0.6 μT hasta ± 4800 μT, con tasas de muestreo desde 8 Hz hasta 4000 Hz.

Una de las capacidades importantes de este módulo es que cuenta con *digital motion processor* (DMP), el cual permite procesar datos y enviarlos en tiempo. Para comunicarse con el MPU9250, se utiliza la dirección 0x68 para establecer conexión con el módulo [26]. En la Figura 26, se muestra el Pinout del módulo inercial utilizado.

Figura 26.
Pinout de MPU9250-GY91.



Nota. Adaptada de [27].

CAPÍTULO 10

Implementación física

La implementación física consiste en la arquitectura del proyecto, de modo que se tiene un dispositivo en donde se puede montar un módulo de navegación que a su vez es la integración de sensores en un PCB. La idea principal es tener acceso a un módulo que tenga la dinámica *plug and play* que sea fácil de programar, montar y tomar datos de interés sin tener que hacer un procedimiento extenso cada vez que se desee obtener información. Para esta implementación se realizaron arquitecturas de implementación física y la forma en la que se toman y procesan los datos. De dicho modo, se pueden generar datasets para calibración que pueden ser a su vez pruebas estáticas y pruebas dinámicas.

10.1. Carro de pruebas

En el objetivo específico 2 es de interés tener un ambiente LOS de modo que en la plataforma del laboratorio Robotat no hayan objetos o personas que puedan interferir en la señal de las anclas hacia la etiqueta o *tag*. Para ello se tomó la decisión de utilizar un robot móvil de dos ruedas de control manual, en el cual se montan los sensores. La elaboración del

carro tuvo tres iteraciones, siendo la primera con 4 motorreductores conectados a un módulo L298N, la segunda con solo 2 motorreductores.

La última iteración con una estructura impresa en 3D con los módulos montados como se muestra en la Figura 27. Para controlar el carro, se utilizó el módulo Bluetooth HC05, que utilizando la aplicación *Arduino Bluetooth Control* se puede conectar al módulo y en una interfaz de control remoto mandar caracteres. Dentro de la programación del ESP32 se activaron pines seriales para comunicarse con el módulo de modo que al recibir los caracteres se definieron rutinas para controlar los motores con otros 4 pines GPIO digitales del ESP32 al L298N. Los controles son izquierda, derecha, adelante y atrás.

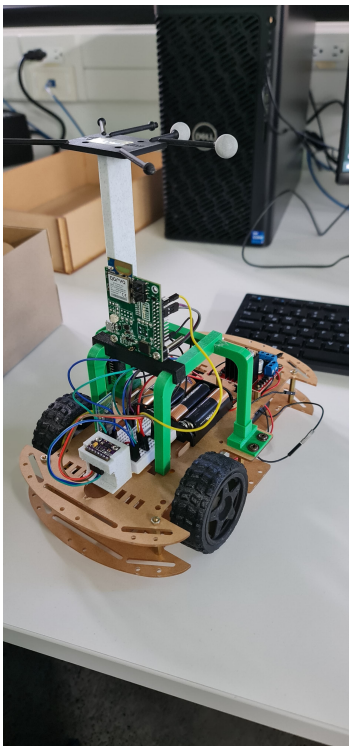
Al momento de realizar las mediciones con el prototipo montado sobre el carro fue de gran utilidad desplazarlo ya que no había necesidad de utilizar un gripper para moverlo de lugar. Otro detalle importante fue imprimir las ruedas del carro ya que las originales realizaban movimientos de subida y bajada, luego al agregar estas ruedas funcionó bien, sin embargo tenían mucho deslizamiento en las superficies y al avanzar se podía apreciar que provocaba vibraciones. Para reducir estos efectos se agregó foam a las ruedas para incrementar la tracción y reducir las vibraciones que pueden provocar mediciones inestables de la IMU.

10.2. Integración de módulos en PCB

El primer objetivo específico, trata de la integración de sensores adicionales al sistema de navegación local, de tal forma que para lograrlo se agregó la IMU para agregar 9 grados de libertad adicionales a este sistema. Esto es importante ya que no solo se tienen posiciones x,y,z sino orientación en roll, pitch, yaw. Incialmente se realizaron prototipos conectando los módulos MDEK1001, MPU9250-GY91, HC-05, L298N (para el carro de pruebas) y se obtuvo el comportamiento esperado en la obtención de datos. Los datos obtenidos son posiciones en coordenadas xy, aceleración, giroscopio, magnetómetro en los tres ejes cada una, esta información es la que nos interesa para analizar la data. Por otro lado se tiene información para control del carro que es la que recibe el HC-05 y manda al ESP32 para activar los pines del L298N.

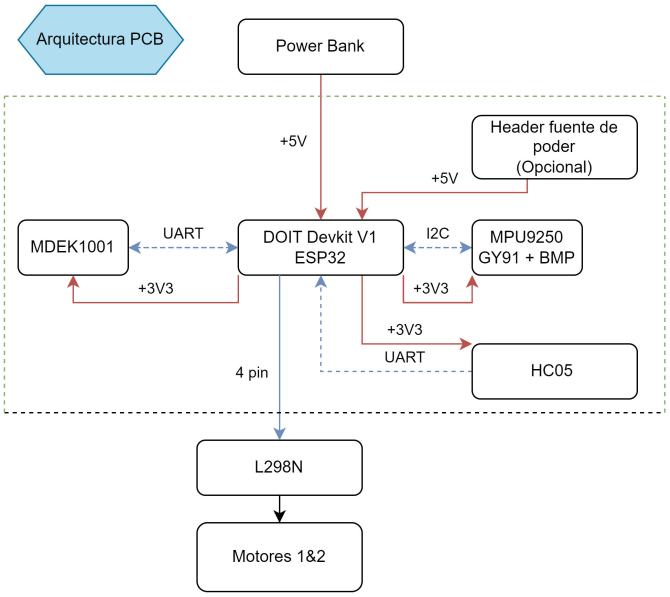
En la Figura 27 podemos observar el carro de pruebas con los módulos conectados mediante cables jumper para realizar obtención de datos preliminares. Dado el buen funcionamiento, se delimitó la arquitectura que se muestra en la Figura 28 que muestra lo que está dentro de la placa y los componentes que se encuentran afuera o deben ser conectados por jumpers.

Figura 27.
Módulos montados en carro de pruebas conectados mediante jumpers.



Nota. Elaboración propia.

Figura 28.
Arquitectura de PCB.



Nota. Elaboración propia.

Una vez funcional el prototipo se eligió la plataforma EasyEda para realizar el esquemático, y el layout del PCB Una vez realizado el esquemático con los componentes descritos, se trasladó a layout de PCB para realizar las conexiones manualmente, también para fines de fabricación, se utilizó casi todas las conexiones en la capa inferior a modo de utilizar una sola capa.

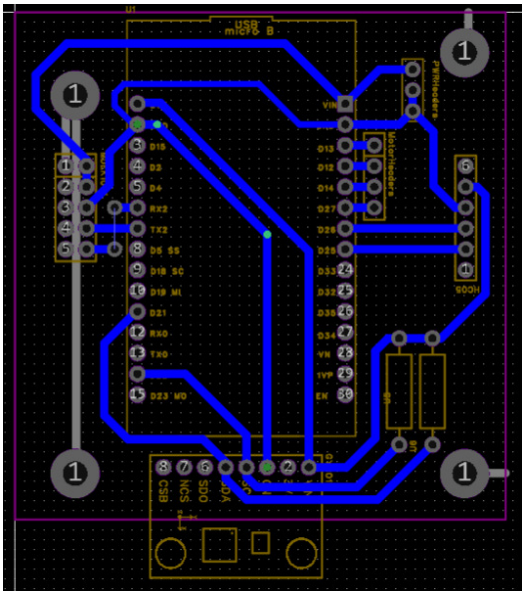
Para la fabricación del PCB se consideraron las especificaciones del fabricante, en este caso el laboratorio Makerlab de la UVG donde hay una CNC de PCBs que es la LPKF S103. Las placas utilizan 1 oz/ft^2 y ofrece anchos mínimos de pista de hasta 10 mil así como el espacio entre las mismas. Otra información importante a considerar es la corriente que consume, que para el ESP32 consiste en máximo 1200 mA y cada pin oscila entre los 20 a 40 mA. Se utilizaron 10 pines periféricos, resultando en 400 mA (usando el máximo) si se utilizan todos los módulos al mismo tiempo y en la máxima corriente, de modo que se considera el peor de los casos para cada pin. Con esta información se utilizó la calculadora *ANSI IPC-2221A PCB Trace Width Calculator* e introduciendo los datos previamente descritos, con subida de temperatura de 10 °C, y largo de conductor máximo de 4 pulgadas, se obtuvo un ancho mínimo de pista de 8.91 mil. Para fines de seguridad, se utilizó el peor de los casos

para el ESP32, colocando 1200 mA resultando en 40.57 mil, de este modo se eligió el valor 40 mil para cada pista.

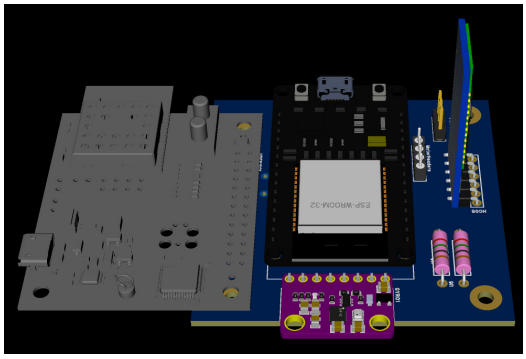
Una vez calculado el ancho de la pista, agregaron las reglas de diseño en una opción que ofrece EasyEDA acoplándose a las especificaciones del fabricante. También se configuraron todos los agujeros de 1.0 mm de diámetro con la finalidad de colocar headers en la placa.

En la Figura 29a se muestra la conexión final de los módulos resultando en una placa de 60x62 mm y con los agujeros de montaje del lado izquierdo alineados a los agujeros del módulo MDEK1001. A la derecha en 29b se muestra la Figura 3D que sirvió para poder modelar la carcasa del PCB con los módulos montados.

Figura 29.
Bottom layer y vista 3D del PCB.



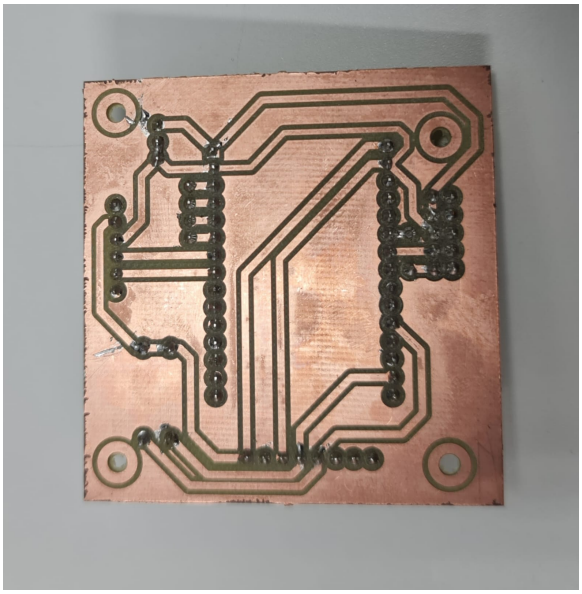
(a) *Bottom layer* de PCB y componentes conectados.



(b) *Vista 3D* del PCB con componentes montados.

Nota. Elaboración propia.

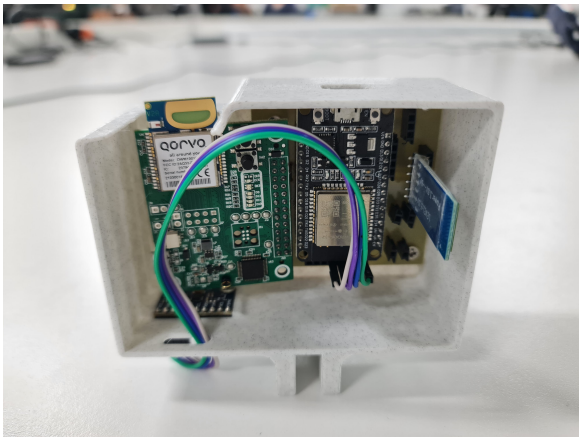
Figura 30.
PCB fabricada con montaje de headers.



Nota. Elaboración propia.

Al tener estos layouts se enviaron los archivos al fabricante y se realizó de manera exitosa en una segunda iteración que es la que se observa en la Figura 30. Luego de tener los componentes montados en los headers de la placa, se realizó lo que se muestra en la Figura 31 que es la integración de módulos junto con la carcasa.

Figura 31.
Módulo de navegación local con carcasa.



Nota. Elaboración propia.

En la figura anterior se muestra que la carcasa tiene la IMU por debajo del módulo

UWB de modo que pueda estar cerca o alineado a su centro para tener mediciones que giren alrededor de un eje al montarlo en el carro de pruebas. Este modelo se logró en la segunda iteración al agregar la IMU debajo. A continuación en la Figura 32 se aprecia el carro en la iteración final. Algunas características de esta iteración son que si el módulo de navegación debe ser guardado o utilizado en otro dispositivo móvil se puede hacer sin problema alguno, solamente se desconectan los cables que van al L298N.

Figura 32.
Iteración final de carro con montaje de módulo de navegación.

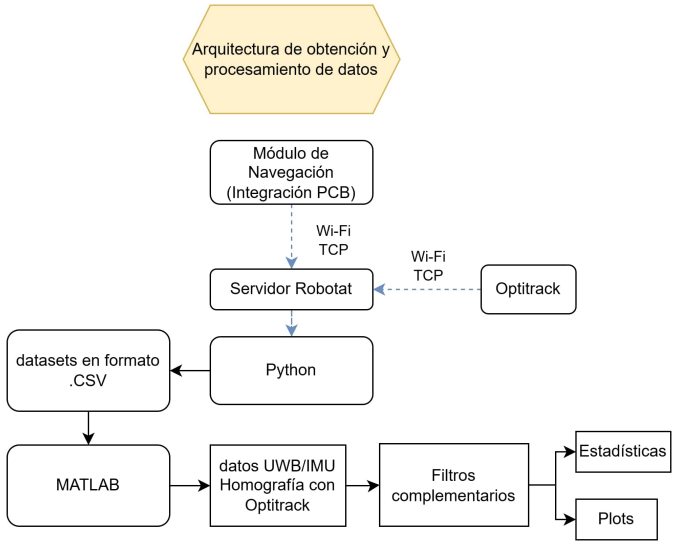
Nota. Elaboración propia.

CAPÍTULO 11

Calibración por homografía y pruebas estáticas

Al tener los módulos MDEK1001 montados en los postes de las cámaras PrimeX41, el módulo de navegación local en PCB con su carcasa y montado en un carro de pruebas se tiene así el sistema de navegación local. Para trabajar el tercer objetivo, se tiene la implementación de filtros con la finalidad de mejorar la exactitud y precisión del sistema de navegación UWB en 2D. Para cumplir esto se definió una forma de trabajo donde se describe cómo se obtiene la información y cómo se procesa. En la Figura 33 se aprecia la arquitectura de obtención de datos y el procesamiento de los mismos.

Figura 33.
Arquitectura de obtención y procesamiento de datos.



Nota. Elaboración propia.

Esta arquitectura fue cambiando con el tiempo ya que originalmente todo se estaba trabajando con MATLAB, sin embargo, para obtener datos con el protocolo TCP, llegaba a tiempos de hasta dos minutos para obtener al menos cien datos a una tasa de 10 Hz. Para fines de prueba se decidió utilizar el lenguaje Python y crear funciones que permitiesen conectarse al puerto del ESP32 en el servidor Robotat; también se implementó una función de desconexión para liberar recursos. Al tener la función de conexión únicamente del ESP32 y obtener las mismas cien filas de datos, se obtuvieron tiempos de 10.1 s hasta 10.5 si la la red del Robotat no estaba siendo usada por nadie más.

Para poder realizar comparaciones con un sistema de captura como lo es Optitrack que funciona como GT, se definió los últimos tres pasos de la Figura 33, donde se debe calibrar los sensores UWB por medio de una homografía proyectiva y con ello obtener una transformación de coordenadas para comparar la exactitud de ambos sistemas y luego la precisión por medio de filtros. De este modo, cualquier coordenada que tengamos del sistema UWB, lo podemos observar en coordenadas del Optitrack y así minimizar los errores de posición.

11.1. Calibración por homografía

11.1.1. Aplicación de la homografía

Para obtener una estimación de la homografía entre puntos del sistema de coordenadas UWB y Optitrack, se utilizó la función de MATLAB `fitgeotform2d`, de modo que se puede ajustar la transformación proyectiva con correspondencia de puntos. A continuación se muestra un ejemplo para poder calcular dicha transformación:

```
% Puntos móviles (UWB) y Puntos Fijos (OptiTrack)
pts_moviles = [x1U, y1U; x2U, y2U; x3U, y3U; x4U, y4U];  % Puntos en UWB
pts_fijos = [x1O, y1O; x2O, y2O; x3O, y3O; x4O, y4O];    % Puntos en OptiTrack

% Transformación proyectiva
tform = fitgeotform2d(pts_moviles, pts_fijos, 'projective');

% Aplicar la transformación a los puntos móviles
outputPoints = transformPointsForward(tform, pts_moviles);
```

La teoría de la homografía y la forma en la que se realizan las estimaciones robustas se cubren en [16], donde se amplía cómo aplicar estas transformaciones orientado a visión por computadora y lo que es reconstrucción en 3D por medio de imágenes 2D.

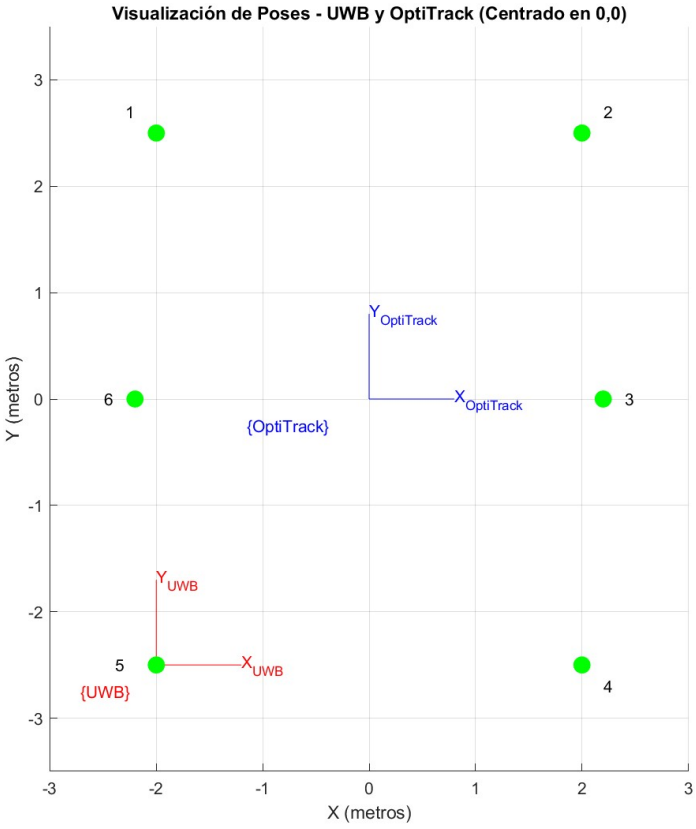
11.1.2. Sistema de coordenadas UWB y Optitrack

Como se abordó en la parte anterior, se realiza lo que es el mapeo de puntos de un sistema de coordenadas a otro. Para comprender esto en la Figura 34 se observa que las coordenadas (0,0) se encuentran en el centro del rectángulo de 4×5 m, este origen es del sistema Optitrack, mientras que el origen de UWB se encuentra en el punto (-2,-2.5) respecto de Optitrack. Al mapear las coordenadas por medio de la homografía, se obtiene el desplazamiento y el escalamiento por parte de la matriz de transformación H.

En pruebas anteriores el origen del sistema UWB estaba colocado en el poste 2 que se

muestra en la Figura 34. Esto representaba una transformación en la posición y rotación ya que los ejes x,y se encontraban alternados. Al colocar el origen en el poste 5 solamente habría que hacer desplazamiento de coordenadas; al presentar resultados, sería comprensible lo que ocurre con la transformación. Cabe destacar que sin importar el origen del sistema UWB, la transformación se realiza correctamente si se calibra bien con los puntos correspondientes.

Figura 34.
Sistema de coordenadas Optitrack vs. UWB (sin transformación) y posición de postes de cámaras con etiquetas en plataforma del laboratorio Robotat UVG.



Nota. Elaboración propia.

11.1.3. Metodología de calibración por homografía

En este trabajo, la homografía se utiliza para mapear los puntos obtenidos del sistema de localización UWB, representando puntos móviles y fijos, dados por el sistema *OptiTrack*. De esta manera se puede obtener el mapeo entre los dos sistemas y mejorar la estimación de posiciones.

La metodología de calibración fue un paso crucial, ya que es importante tomar puntos

en paralelo de dos sistemas diferentes. Para ello en un script de Python se implementaron funciones de conexión mediante TCP para obtener de un puerto las coordenadas de un marker Optitrack específico y en otro puerto obtener las coordenadas del módulo UWB. En el script se colocaron inputs para poder especificar ruta de archivo y nombre del mismo.

Los nombres de los archivos se construyen de la siguiente manera El archivo generado fue nombrado de la siguiente manera, `CALIB_PCB1_combined_data_STATIC_R2.csv`. Se colocó `CALIB` ya que pertenece a un dataset de calibración, combined data es la combinación entre Optitrack y UWB, `STATIC` es toma de datos con marker y UWB estáticos, `R2` quiere decir el marco de referencia utilizado, en este caso fue del poste No. 5 ya que R1 sería el poste No.2 con el cual se obtuvieron los primeros resultados del primer marco de referencia pero no fueron los resultados finales.

La idea de tener un *dataset* separado por comas es tener información cruda para poderla procesar. Una vez definida esta metodología se realizó la toma de datos, siendo 200 filas de datos por punto y se tomaron 24 puntos, resultando en 4800 filas de datos para calibrar el sistema UWB. Para aclarar, la plataforma del Laboratorio Robotat en la UVG tiene 6 tarimas, tomando 4 puntos por tarima para realizar un total de 24 puntos de calibración ya mencionados.

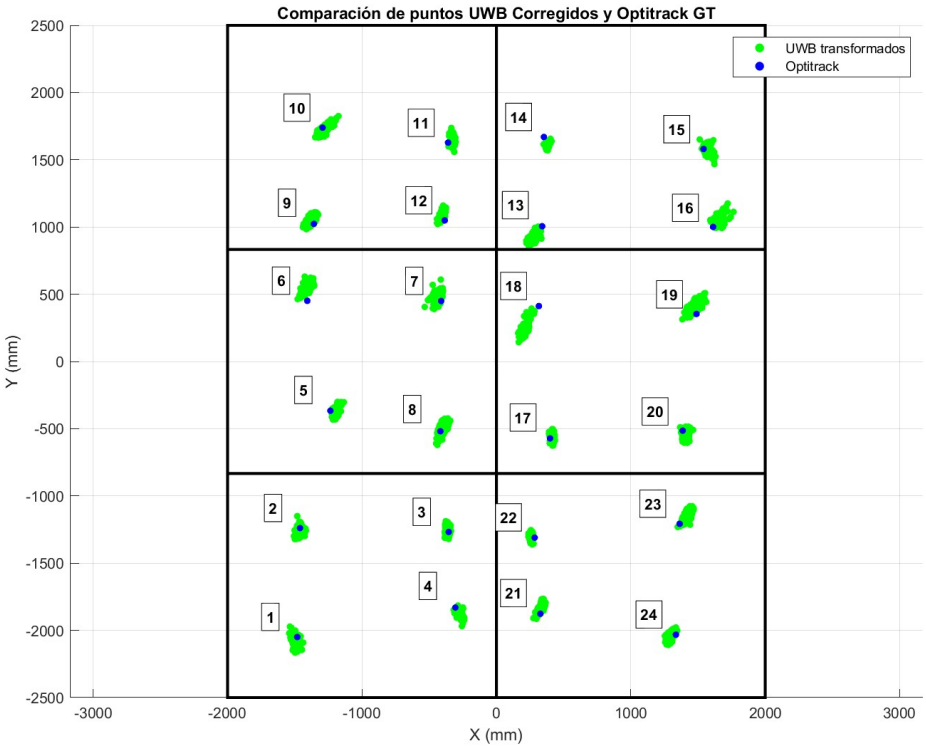
11.1.4. Resultados de calibración

Utilizando la metodología de calibración, se realizó un script en MATLAB para obtener la matriz de homografía para los 24 puntos descritos previamente. Una vez realizado esto se realizó un plot con las coordenadas transformadas, de modo que el marco de referencia UWB esté alineado con el del Optitrack. Esto se puede observar en la Figura 35, a su vez también se colocó cada tarima en negro para tener una referencia física de dónde se realizó cada punto. A continuación se muestra la matriz de Homografía resultante.

$$H = \begin{bmatrix} 0.9806 & 0.0487 & -2036.3 \\ -0.0347 & 1.0527 & -2511.9 \\ -1.8418 \times 10^{-6} & 1.0074 \times 10^{-5} & 1 \end{bmatrix}$$

Figura 35.

Transformación de coordenadas UWB a Optitrack mediante matriz de homografía 11.1.4.



Nota. Elaboración propia.

Cuadro 1.

Comparación de medias y desviaciones estándar de sistema UWB crudo y transformado contra Optitrack para el eje x

Pt.	$\bar{x}$ UWB	$\bar{x}$ Optitrack	Diff $\bar{x}$	e(%)	σ UWB	σ Optitrack
1	-1487.35	-1481.65	5.7	0.385	18.772	0.02
2	-1450.06	-1461.26	11.2	0.766	17.002	0.011
3	-362.08	-355.11	6.97	1.964	9.718	0.009
4	-275.07	-305.23	30.16	9.881	14.992	0.016
5	-1188.38	-1234.95	46.57	3.771	15.756	0.013
6	-1401.27	-1407.15	5.88	0.418	23.48	0.023
7	-431.33	-410.92	20.41	4.967	19.726	0.006
8	-399.76	-416.67	16.91	4.058	20.508	0.009
9	-1381.67	-1358.08	23.59	1.737	22.703	0.013
10	-1272.35	-1293.19	20.84	1.612	33.968	0.025
11	-329.45	-359.32	29.88	8.314	9.613	0.084
12	-405.3	-384.68	20.62	5.36	12.315	0.04
13	270.5	340.54	70.03	20.566	22.8	0.014
14	375.11	353.17	21.94	6.212	8.34	0.011
15	1571.31	1541.33	29.98	1.945	19.188	0.018
16	1655.48	1612.21	43.27	2.684	26.604	0.015
17	412.91	399.02	13.88	3.48	9.578	0.006
18	211.16	315.81	104.65	33.138	23.642	0.009
19	1468.06	1488.18	20.12	1.352	32.472	0.01
20	1404.32	1385.9	18.42	1.329	14.876	0.009
21	331.02	327.38	3.64	1.112	17.799	0.019
22	255.65	284.6	28.95	10.173	7.648	0.008
23	1411.87	1363.13	48.74	3.575	23.559	0.017
24	1294.56	1335.2	40.64	3.044	18.13	0.016

Nota. Elaboración propia. Todas las columnas a partir de la 2 están en milímetros.

Cuadro 2.
Comparación de medias y desviación estándar de sistema UWB crudo y transformado contra Optitrack para el eje y

Pt.	$\bar{x}$ UWB	$\bar{x}$ Optitrack	Diff $\bar{x}$	e(%)	σ UWB	σ Optitrack
1	-2088.17	-2049.66	38.51	1.879	34.99	0.021
2	-1266.56	-1239.35	27.21	2.196	24.764	0.01
3	-1260.61	-1268.11	7.5	0.592	24.35	0.005
4	-1876.68	-1830.33	46.35	2.532	23.61	0.021
5	-369.4	-366.42	2.98	0.813	29.361	0.008
6	554.32	451.23	103.09	22.846	30.55	0.039
7	477.28	449.83	27.45	6.102	33.914	0.008
8	-496.88	-519.01	22.12	4.262	36.994	0.013
9	1050.46	1023.24	27.23	2.661	27.33	0.011
10	1734.1	1739.25	5.15	0.296	32.887	0.019
11	1646.35	1627.9	18.44	1.133	26.859	0.053
12	1088.98	1049.38	39.6	3.774	25.124	0.026
13	911.22	1005.94	94.72	9.416	30.214	0.008
14	1610.15	1669.85	59.7	3.575	14.847	0.012
15	1574.5	1580.5	6	0.379	27.285	0.012
16	1061.71	1000.67	61.04	6.1	30.249	0.015
17	-557.07	-572.21	15.14	2.646	25.849	0.008
18	246.89	412.03	165.14	40.079	43.156	0.009
19	407.61	353.15	54.45	15.419	34.73	0.017
20	-540.1	-514.5	25.6	4.976	27.392	0.012
21	-1828.28	-1876.42	48.14	2.566	31.675	0.011
22	-1306.3	-1310.17	3.87	0.295	22.27	0.009
23	-1155.03	-1207.62	52.59	4.355	33.257	0.015
24	-2043.37	-2032.29	11.09	0.546	26.392	0.019

Nota. Elaboración propia. Todas las columnas a partir de la 2 están en milímetros.

Podemos observar que en ambos cuadros se tienen errores desde 0.3 % hasta 40 %, que indica buen rango de exactitud al tener una refencia GT como lo es Optitrack. Esta información también nos puede servir como mapa para detectar por qué algunos puntos tienen desviaciones estándar elevadas. Por ejemplo, en casos anteriores, por valores atípicos, que eran errores de obtención de datos, la desviación estándar llegaba hasta 500 mm, luego al corregir estos errores, disminuyó significativamente los porcentajes de error y las desviaciones estándar. También se puede observar de manera comparativa que la desviación estándar en Optitrack es significativamente baja, de hasta 0.01 mm en promedio.

CAPÍTULO 12

Filtros

En el capítulo anterior se abarcó lo que es el segundo objetivo en cuanto a la mejora de exactitud del sistema UWB utilizando homografía por medio del sistema Optitrack como Ground Truth. En este capítulo se tratará el tema de la precisión. Para comprender la precisión, se trata sobre la consistencia de datos, siendo que en este caso, si tenemos información estática como la Figura 35, Optitrack se puede observar que al ser 200 puntos por cada dataset, es muy consistente en la medición en dicha posición. Por otra parte UWB que está en verde, puede observarse que tiene un error más grande comparado con Optitrack. La información que directamente nos habla de la precisión es la desviación estándar como se muestra en los cuadros 1 y 2 la desviación estándar no pasa de 35 mm. De acuerdo con la empresa Qorvo, los módulos MDEK1001 tienen desviaciones de $\pm$ 100 mm o $\pm$ 10 cm. Esto puede mejorarse aún más al aplicarse filtrado en las posiciones.

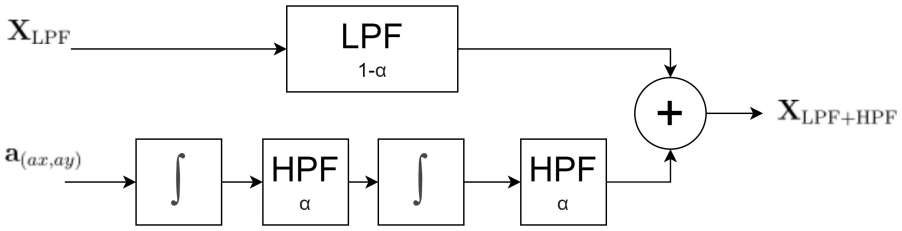
Para saber qué tanto mejora la consistencia en varias mediciones, se utilizaron filtros. El primer filtro es el complementario normal, el segundo es filtro complementario usando Butterworth. En cuanto a los filtros complementarios, estos se trabajan en dominio de frecuencia, de modo que, en una arquitectura simple tiene un valor ingresando a un filtro

pasa bajas y otro valor a un pasa altas, luego estos valores se suman para tener el total de información.

12.1. Filtro complementario

Para esta parte se utilizó la arquitectura de la Figura 36, que sirvió para implementar las contribuciones de cada sensor y mejorar la precisión del posicionamiento.

Figura 36.
Arquitectura de filtro complementario con constante de tiempo τ para posición combinando sistema UWB con aceleraciones de IMU.



Nota. Elaboración propia.

Para comprender la arquitectura comenzamos con la posición que nos da el sistema UWB, este se filtra con un pasa bajas. Para ello la ecuación es la siguiente:

$$\mathbf{X}_{\text{LPF}} = (1 - \alpha) \cdot \mathbf{X}_{\text{UWB}} \tag{7}$$

- $\mathbf{X}_{\text{UWB}}$ es la posición del el sistema UWB.
- $\alpha = \frac{\tau}{\tau + \Delta t}$ representa el coeficiente de filtro pasa altas.
- τ es la constante de tiempo o el peso que se le da que es 1 para este caso.
- $\Delta t = \frac{1}{f_s}$ es la frecuencia de muestreo.

12.1.1. Filtro pasa altas (HPF) aplicado sobre aceleraciones IMU

El filtro pasa altas es aplicado a las aceleraciones que mide la IMU (en unidades de 1g luego convertidas). En este proceso, para eliminar el *drift* por falta de las constantes en la

integración, se hacen las dos integraciones, seguida cada una de un filtro pasa altas para obtener posición a partir de las aceleraciones. Esto se divide en los siguientes pasos:

Paso 1: Integrar aceleración para obtener velocidad

Para obtener la velocidad en los ejes x e y se integra la aceleración.

$$\mathbf{V} = \int \mathbf{a}\, dt \tag{8}$$

- $\mathbf{a}_{(ax,ay)}$ corresponde a las aceleraciones de la IMU.

Luego de integrarlo aplica un filtro pasa altas a dicha velocidad:

$$\mathbf{V}_{\text{HPF}} = \alpha \cdot \mathbf{V} \tag{9}$$

Paso 2: Integrar velocidad para obtener posición

De igual manera con el paso anterior, se integra y ahora se obtiene la posición como se muestra a continuación:

$$\mathbf{X}_{\text{HPF}} = \alpha \cdot \int \mathbf{V}_{\text{HPF}}\, dt \tag{10}$$

12.1.2. Combinación de los filtros LPF + HPF

Por último, se combinan los filtros LPF + HPF para tener una estimación de la posición como se muestra a continuación:

$$\mathbf{X}_{\text{Filtrado}} = \mathbf{X}_{\text{LPF}} + \mathbf{X}_{\text{HPF}} \tag{11}$$

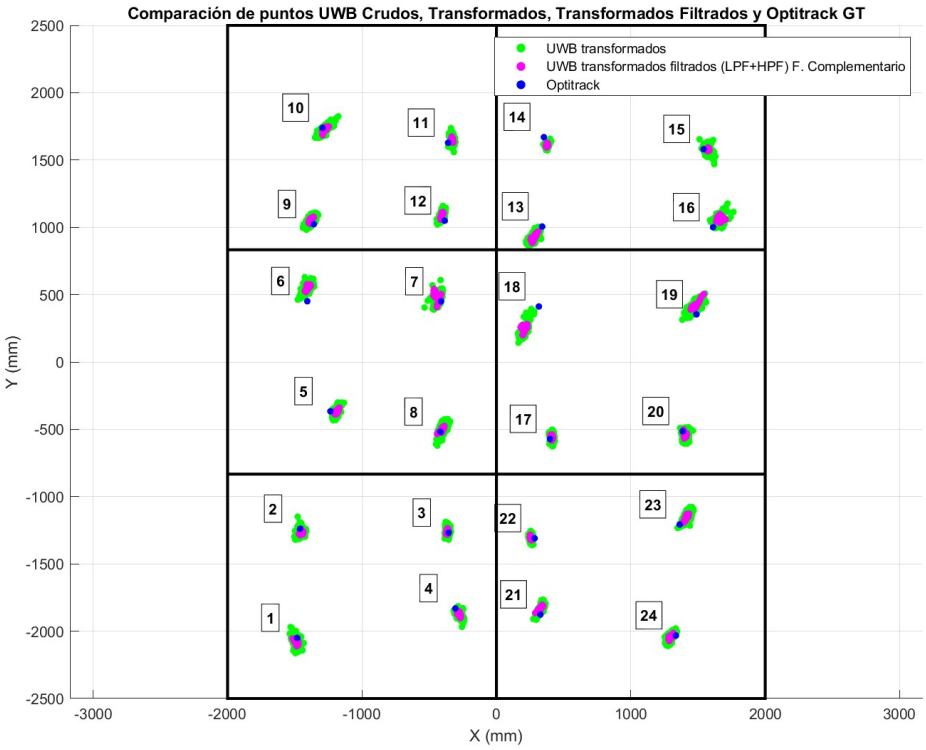
- $\mathbf{X}_{\text{LPF}}$ representa la posición filtrada con LPF de posiciones del sistema UWB).

- $\mathbf{X}_{\mathrm{HPF}}$ representa la posición filtrada con HPF de aceleraciones de la IMU.

12.1.3. Resultados estáticos de filtro complementario normal

Se aplicó el filtro complementario descrito previamente y se observó una mejora en la precisión de los puntos UWB. Esto se puede apreciar en la Figura 37 al ver cómo reduce la desviación estándar significativamente.

Figura 37.
Aplicación de filtro complementario normal en puntos de datasets de calibración transformados como pruebas estáticas.



Nota. Elaboración propia.

A continuación se muestran las tablas para comparar medias de UWB, determinar si se pierde información al aplicar filtrado y cómo mejora la precisión

Cuadro 3.
Medición de efectividad del filtro complementario normal en los distintos puntos para eje x

Pt.	$\bar{x}$ UWB	$\bar{x}$ UWB Filt	Diff $\bar{x}$ UWB	σ UWB	σ UWB Filt	% Mejora σ
1	-1487.3463	-1488.5515	1.2052	18.7724	7.7022	58.9708
2	-1450.0619	-1451.2949	1.233	17.0022	7.0137	58.7486
3	-362.081	-362.7067	0.62568	9.7183	4.3744	54.9874
4	-275.0726	-274.8211	0.25146	14.9915	8.1369	45.7233
5	-1188.3792	-1188.5663	0.18714	15.756	8.1139	48.5025
6	-1401.2684	-1402.0086	0.74028	23.4803	10.0185	57.3324
7	-431.3332	-433.8295	2.4963	19.726	15.7569	20.1211
8	-399.7623	-402.1931	2.4308	20.5081	9.1032	55.6118
9	-1381.6683	-1380.333	1.3353	22.7025	8.366	63.1494
10	-1272.3521	-1271.8936	0.45843	33.968	12.6215	62.843
11	-329.4475	-328.7831	0.66444	9.613	4.0103	58.2831
12	-405.2957	-405.5104	0.21477	12.3149	6.0935	50.5196
13	270.5022	273.4538	2.9516	22.8	10.9166	52.1202
14	375.1128	375.2004	0.087565	8.3401	3.5821	57.0494
15	1571.3068	1571.6105	0.30366	19.1875	4.8867	74.5321
16	1655.4849	1659.4643	3.9795	26.6037	13.9933	47.4008
17	412.9076	412.935	0.027328	9.5782	4.3312	54.7809
18	211.157	210.8234	0.33353	23.642	10.0096	57.6619
19	1468.0568	1472.0638	4.007	32.4717	15.7001	51.65
20	1404.3208	1404.3432	0.02242	14.876	6.6324	55.4155
21	331.0173	328.6762	2.3411	17.7989	11.4187	35.846
22	255.6461	256.658	1.0119	7.6482	3.5684	53.3435
23	1411.8718	1412.875	1.0032	23.5591	10.7658	54.303
24	1294.5598	1295.4299	0.87004	18.1305	8.9901	50.4145

Nota. Elaboración propia. Todas las columnas a partir de la 2 están en milímetros, excepto la última.

Cuadro 4.
Medición de efectividad del filtro complementario normal en los distintos puntos para eje y.

Pt.	$\bar{x}$ UWB	$\bar{x}$ UWB Filt	Diff $\bar{x}$ UWB	σ UWB	σ UWB Filt	% Mejora σ
1	-2085.1086	-2083.8983	1.2103	34.9129	13.3513	61.7584
2	-1265.4212	-1266.3489	0.92772	24.7126	9.5655	61.2931
3	-1260.4593	-1261.7803	1.321	24.3223	10.7825	55.6681
4	-1875.8351	-1876.9977	1.1626	23.5862	13.2928	43.6415
5	-369.7506	-370.7644	1.0138	29.3281	10.96	62.6297
6	553.2079	553.2368	0.028927	30.5409	13.3993	56.1268
7	476.6499	475.3251	1.3248	33.9242	29.6142	12.7049
8	-497.3395	-499.2764	1.937	36.9701	11.5264	68.8225
9	1049.2142	1050.8541	1.6399	27.341	11.9123	56.4306
10	1733.112	1730.9021	2.2099	32.9443	12.5998	61.7541
11	1646.6075	1647.6121	1.0045	26.8901	9.7427	63.7685
12	1088.6377	1088.8899	0.25227	25.1481	14.7839	41.2125
13	911.3043	915.3869	4.0826	30.2597	18.6306	38.431
14	1611.311	1610.241	1.0701	14.8753	9.5234	35.9787
15	1576.6962	1577.2967	0.60052	26.4251	6.3866	75.8314
16	1063.2294	1062.5784	0.65106	30.3302	13.0675	56.9159
17	-557.7775	-558.6825	0.90508	25.8509	10.8132	58.1709
18	246.4178	247.1267	0.70882	43.1847	17.579	59.2934
19	407.7263	412.593	4.8667	34.7963	16.2243	53.3735
20	-541.1516	-539.1226	2.029	27.4148	12.7692	53.4222
21	-1828.3175	-1830.4877	2.1702	31.6241	15.2956	51.633
22	-1306.686	-1307.0074	0.32145	22.2581	10.307	53.6935
23	-1156.4551	-1155.1341	1.321	33.2537	17.1345	48.4734
24	-2044.6425	-2043.5396	1.1029	26.3651	10.2811	61.005

Nota. Elaboración propia. Todas las columnas a partir de la 2 están en milímetros, excepto la última.

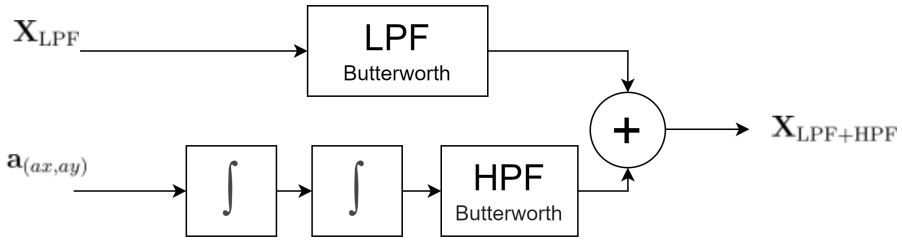
Como se observó en los cuadros 3 y 4, se tiene una mejora significativa de hasta 75 % en ambos ejes. Esto nos indica que funciona bien para datos estáticos. También se observa que cambia la media hasta 5 mm para el eje y; con el eje x cambia hasta 4 mm. Esto podría indicar cierta pérdida de información, sin embargo, al ser datos estáticos, no dice mucho si esto fuese aplicado a una trayectoria.

12.2. Filtro complementario butterworth

En esta sección, se tiene el filtro complementario Butterworth (Figura 38). Es importante observar que el acelerómetro tiene un filtro pasa altas, debido a que se puede variar la frecuencia de corte que, a diferencia del filtro anterior, es fijo o, en algunas ocasiones puede

modificarse alpha para distribuir la confianza de contribuciones de cada sensor con el fin de obtener los mejores resultados sin perder información de ambos.

Figura 38.
Arquitectura de filtro complementario usando Butterworth para posición combinando sistema UWB con aceleraciones de la IMU.



Nota. Elaboración propia.

12.2.1. Filtro pasa bajas (LPF)

La posición que nos da el sistema UWB, se le aplica un filtro pasa bajas Butterworth como se muestra a continuación:

$$\mathbf{X}_{\mathrm{LPF}} = \mathrm{ButterworthLPF}(\mathbf{X}_{\mathrm{UWB}}) \tag{12}$$

Donde:

- $\mathbf{X}_{\mathrm{UWB}}$ corresponde posición obtenida del sistema UWB.
- ButterworthLPF() corresponde al filtro pasa bajas Butterworth.

12.2.2. Filtro pasa altas (HPF) Butterworth aplicado sobre aceleraciones IMU

Las aceleraciones de la IMU se integran doblemente a modo de obtener posición. Como se mencionó anteriormente puede tener *drift* y el filtro pasa altas ayuda a eliminar efectos de frecuencias bajas y que tenga tendencia al infinito. Este proceso es el mismo que en la sección pasada por lo cual se omite.

12.2.3. Combinación de los filtros LPF + HPF Butterworth

Para obtener la posición final y filtrada, se hace combinación de posiciones, de igual manera que la sección anterior y con ello obtenemos el filtro complementario Butterworth para posiciones como se muestra a continuación:

$$\mathbf{X}_{\text{final}} = \mathbf{X}_{\text{LPF}} + \mathbf{X}_{\text{HPF}} \tag{13}$$

Donde:

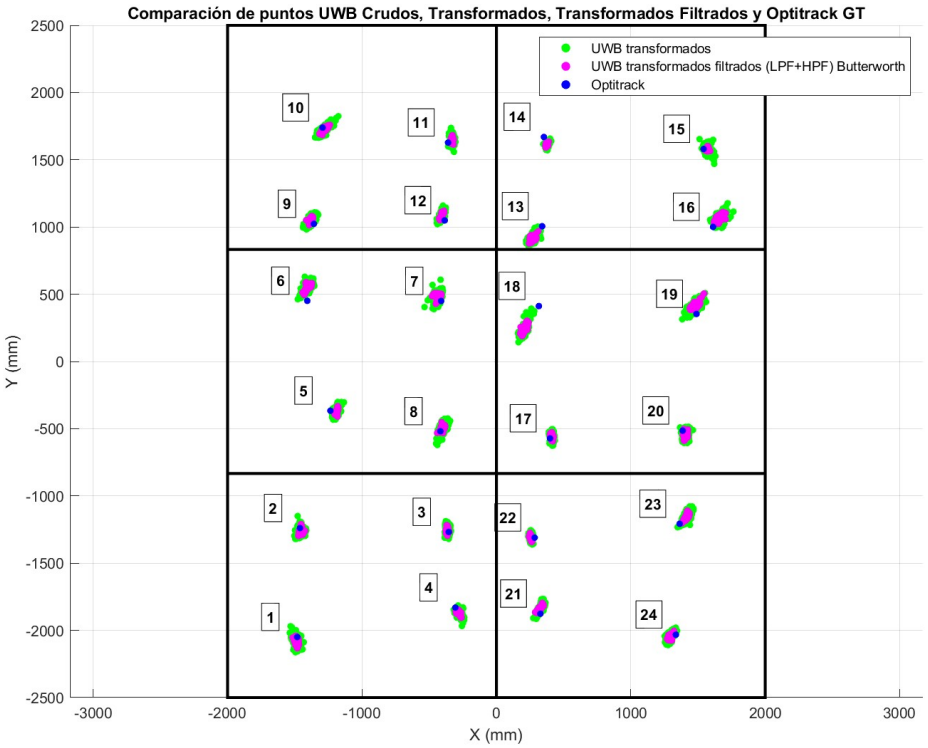
- $\mathbf{X}_{\text{LPF}}$ representa la posición LPF del sistema UWB.
- $\mathbf{X}_{\text{HPF}}$ representa la posición de HPF de las aceleraciones de la IMU.

12.2.4. Resultados estáticos de filtro complementario Butterworth

Al aplicar el Filtro complementario Butterworth se obtuvieron resultados satisfactorios como se observar en la Figura 39. En verde tenemos las posiciones UWB transformadas y crudas, en morado tenemos las posiciones transformadas y filtradas y se observa que se reduce la desviación estándar respecto de la información cruda. Esto representa una mejoría en la precisión del sistema UWB al aplicar este filtro complementario considerando las aceleraciones y posiciones estáticas.

Figura 39.

Aplicación de filtro complementario Butterworth en puntos de datasets de calibración transformados como pruebas estáticas.



Nota. Elaboración propia.

Cuadro 5.
Medición de efectividad de filtro complementario Butterworth en los distintos puntos para eje x

Pt.	$\bar{x}$ UWB	$\bar{x}$ UWB Filt	Diff $\bar{x}$ UWB	σ UWB	σ UWB Filt	% Mejora σ
1	-1487.3463	-1487.3728	0.026541	18.7724	9.3234	50.3345
2	-1450.0619	-1450.3617	0.29987	17.0022	10.8697	36.0692
3	-362.081	-362.147	0.065995	9.7183	5.6872	41.4793
4	-275.0726	-275.2809	0.20839	14.9915	10.569	29.5004
5	-1188.3792	-1188.7424	0.36323	15.756	7.8604	50.1119
6	-1401.2684	-1401.5224	0.25405	23.4803	15.6614	33.2998
7	-431.3332	-431.4619	0.12867	19.726	14.3776	27.1132
8	-399.7623	-400.1199	0.35761	20.5081	9.6496	52.9476
9	-1381.6683	-1381.8229	0.15465	22.7025	10.2676	54.7732
10	-1272.3521	-1272.0387	0.31336	33.968	17.882	47.3563
11	-329.4475	-329.4131	0.034379	9.613	4.7485	50.6039
12	-405.2957	-405.2771	0.018526	12.3149	8.1216	34.0504
13	270.5022	270.8957	0.39354	22.8	12.738	44.1315
14	375.1128	375.1708	0.058002	8.3401	4.9601	40.5268
15	1571.3068	1571.1936	0.1132	19.1875	6.5979	65.6134
16	1655.4849	1655.901	0.41613	26.6037	18.2366	31.4509
17	412.9076	412.92	0.012402	9.5782	6.4248	32.9228
18	211.157	210.9583	0.19864	23.642	14.6772	37.9191
19	1468.0568	1469.458	1.4012	32.4717	18.6223	42.6506
20	1404.3208	1404.3908	0.069978	14.876	9.8238	33.9623
21	331.0173	330.6079	0.40943	17.7989	12.9423	27.286
22	255.6461	255.7375	0.091381	7.6482	5.4352	28.9351
23	1411.8718	1412.1524	0.28066	23.5591	15.1081	35.8714
24	1294.5598	1294.7053	0.14547	18.1305	11.8824	34.4618

Nota. Elaboración propia. Todas las columnas a partir de la 2 están en milímetros, excepto la última.

Cuadro 6.
Medición de efectividad de filtro complementario Butterworth en los distintos puntos para eje y

Pt.	$\bar{x}$ UWB	$\bar{x}$ UWB Filt	Diff $\bar{x}$ UWB	σ UWB	σ UWB Filt	% Mejora σ
1	-2085.1086	-2085.209	0.10042	34.9129	19.1719	45.0866
2	-1265.4212	-1265.4619	0.040716	24.7126	15.8792	35.7446
3	-1260.4593	-1260.3371	0.1222	24.3223	15.3193	37.0152
4	-1875.8351	-1875.6387	0.19649	23.5862	15.4304	34.5784
5	-369.7506	-370.6168	0.86615	29.3281	15.8611	45.9183
6	553.2079	553.0484	0.15941	30.5409	19.851	35.0021
7	476.6499	476.3121	0.33783	33.9242	20.5231	39.5031
8	-497.3395	-497.2156	0.12386	36.9701	15.5334	57.9839
9	1049.2142	1049.0886	0.1256	27.341	14.0526	48.6025
10	1733.112	1732.8633	0.24865	32.9443	15.8578	51.8647
11	1646.6075	1646.6622	0.054656	26.8901	14.2555	46.986
12	1088.6377	1089.0894	0.45176	25.1481	16.4192	34.7097
13	911.3043	911.4352	0.13083	30.2597	18.9645	37.3275
14	1611.311	1611.1775	0.13354	14.8753	8.8549	40.4727
15	1576.6962	1576.4476	0.24861	26.4251	8.2714	68.6986
16	1063.2294	1063.0567	0.17272	30.3302	20.5987	32.0852
17	-557.7775	-557.8157	0.038241	25.8509	16.4472	36.3768
18	246.4178	245.815	0.6028	43.1847	24.7147	42.7699
19	407.7263	409.3796	1.6533	34.7963	19.0092	45.3701
20	-541.1516	-540.8803	0.27128	27.4148	17.9208	34.6307
21	-1828.3175	-1828.9078	0.59028	31.6241	19.4897	38.3709
22	-1306.686	-1306.4267	0.25934	22.2581	16.4273	26.1963
23	-1156.4551	-1156.3001	0.15503	33.2537	21.6754	34.818
24	-2044.6425	-2044.5517	0.090798	26.3651	15.4955	41.2272

Nota. Elaboración propia. Todas las columnas a partir de la 2 están en milímetros, excepto la última.

En los cuadros 5 y 6, se tiene porcentajes de mejora de hasta 68 %. Además la diferencia de medias entre los puntos transformados y los puntos transformados-filtrados es de hasta 2 mm. De igual manera que el filtro complementario normal, no nos da mucha información de la efectividad en trayectorias, sin embargo en datos estáticos transformados y filtrados nos muestra una mejoría significativa en la desviación estándar con algunos datos por debajo de 10 mm y otros hasta 20 mm.

CAPÍTULO 13

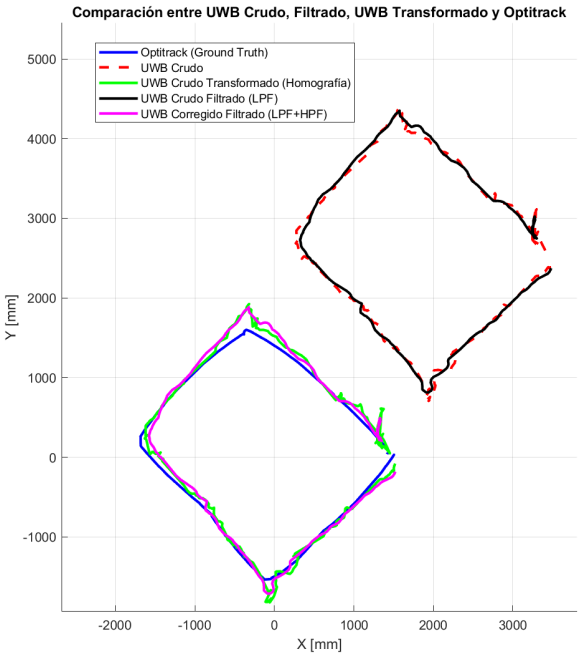
Pruebas dinámicas

Ya que se evaluaron las pruebas estáticas que corresponden a los datasets de calibración, se realizó la implementación para pruebas dinámicas. Se realizaron distintas trayectorias para ver qué tanto se apega a la figura original que son los puntos dados por Optitrack. Esto se realizó con la finalidad de observar que en distintas trayectorias funcionan los filtros y evaluar cuestiones como condiciones LOS. Esto forma parte del tercer objetivo que son experimentos con módulos UWB en lo que respecta las trayectorias en 2D. Además en el capítulo anterior también se involucra este objetivo con las pruebas estáticas como experimento ya que se observaron datos sobre desviación estándar y porcentajes de error en medias y mejora de las desviaciones.

13.1. Filtro complementario normal

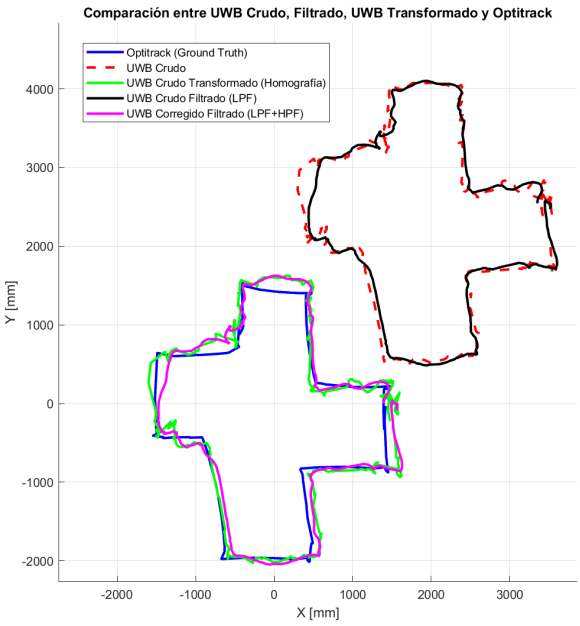
A continuación se muestra una serie de Figuras con trayectorias con información cruda almacenada en distintos *datasets* para utilizar el filtro complementario normal, a modo de determinar cualitativamente el comportamiento de este filtro y compararlo con otro filtro.

Figura 40.
Trayectoria 1 con filtro complementario normal con $\mathbb{R}^2 = 0.9469$.



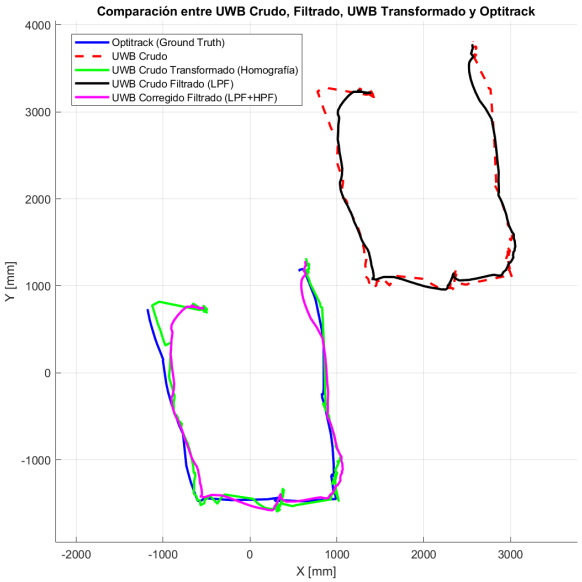
Nota. Elaboración propia.

Figura 41.
Trayectoria 2 con filtro complementario normal con $\mathbb{R}^2 = 0.9672$.



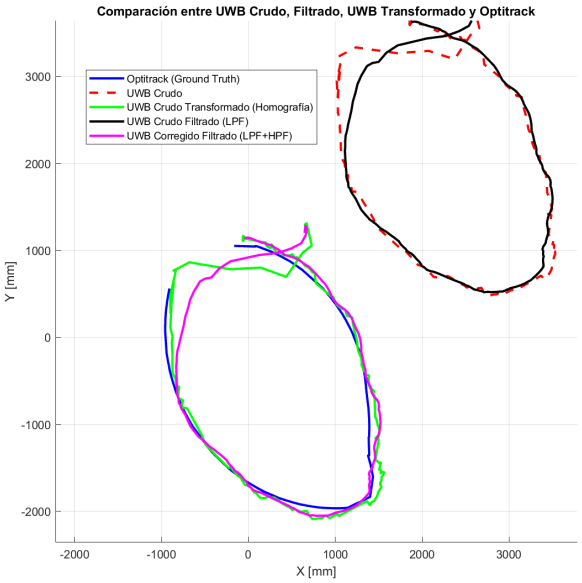
Nota. Elaboración propia.

Figura 42.
Trayectoria 3 con filtro complementario normal con $\mathbb{R}^2 = 0.9322$.



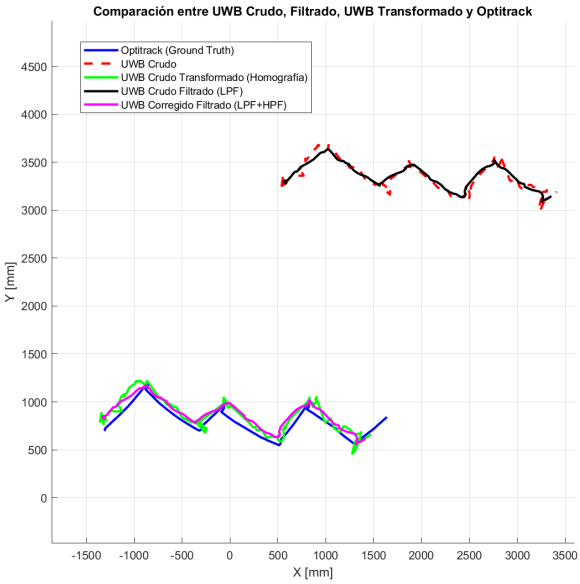
Nota. Elaboración propia.

Figura 43.
Trayectoria 4 con filtro complementario normal con $\mathbb{R}^2 = 0.8338$.



Nota. Elaboración propia.

Figura 44.
Trayectoria 5 con filtro complementario normal con $\mathbb{R}^2 = 0.9582$.

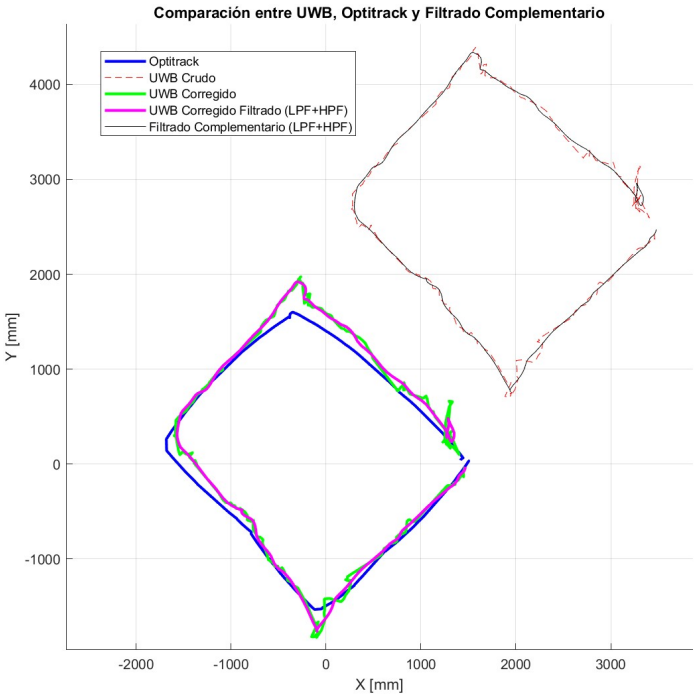


Nota. Elaboración propia.

13.2. Filtro complementario Butterworth

A continuación se muestra una serie de figuras con filtro complementario Butterworth. A comparación de la sección anterior, la información que está transformada y filtrada se apega a la trayectoria generada por Optitrack y no se pierde tanta información. Esto se observa al ver qué tan pegado va a la trayectoria GT y cómo elimina el ruido o picos de la trayectoria cruda transformada. El Filtro complementario Butterworth ha mostrado ser mejor respecto al complementario normal.

Figura 45.
Trayectoria 1 con filtro complementario Butterworth con $\mathbb{R}^2 = 0.9793$.

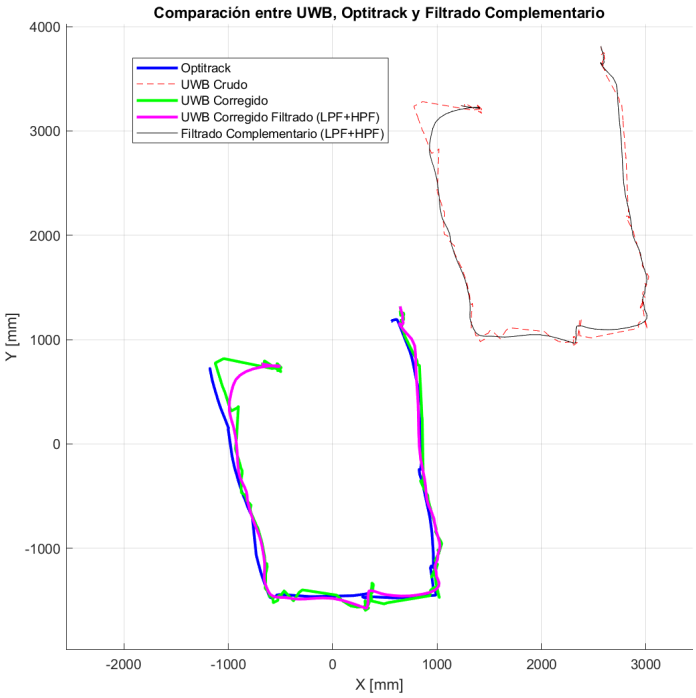


Nota. Elaboración propia.

Figura 46.
Trayectoria 2 con filtro complementario Butterworth con $\mathbb{R}^2 = 0.9886$.

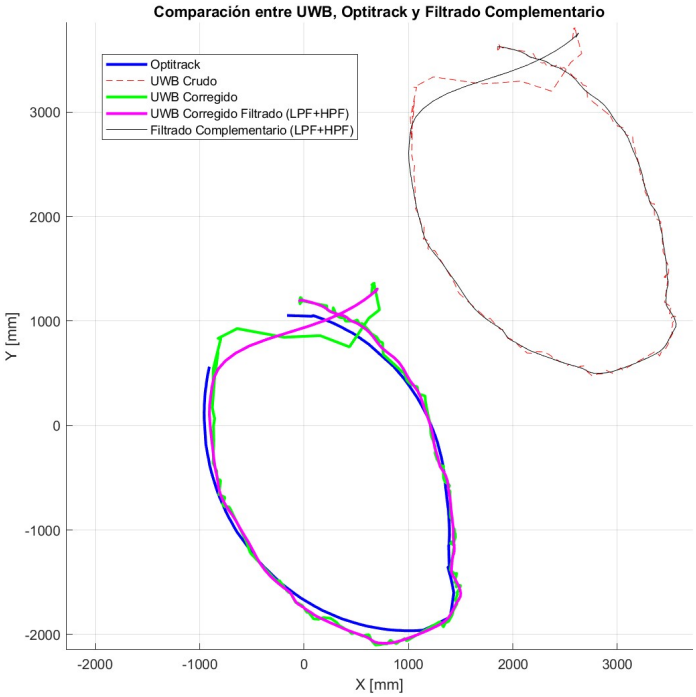
Nota. Elaboración propia.

Figura 47.
Trayectoria 3 con filtro complementario Butterworth con $\mathbb{R}^2 = 0.9717$.



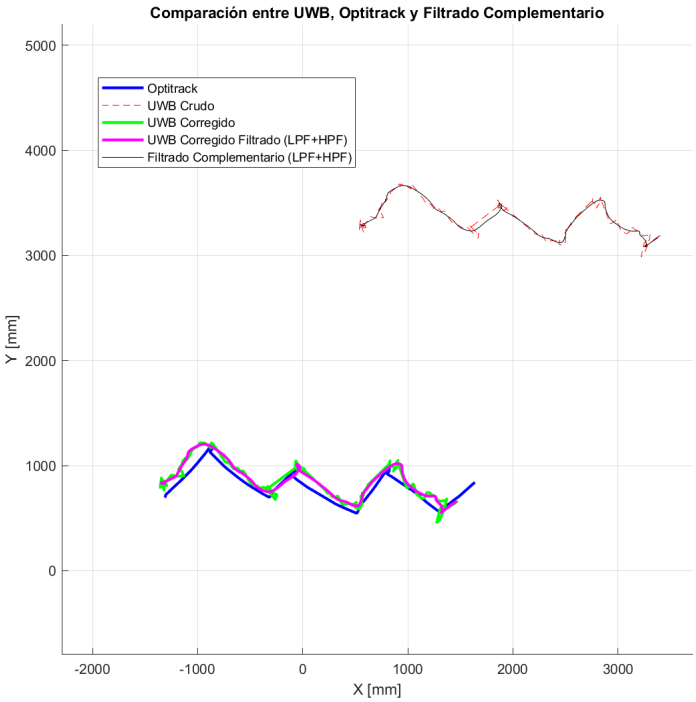
Nota. Elaboración propia.

Figura 48.
Trayectoria 4 con filtro complementario Butterworth con $\mathbb{R}^2 = 0.8958$.



Nota. Elaboración propia.

Figura 49.
Trayectoria 5 con filtro complementario Butterworth con $\mathbb{R}^2 = 0.9818$.

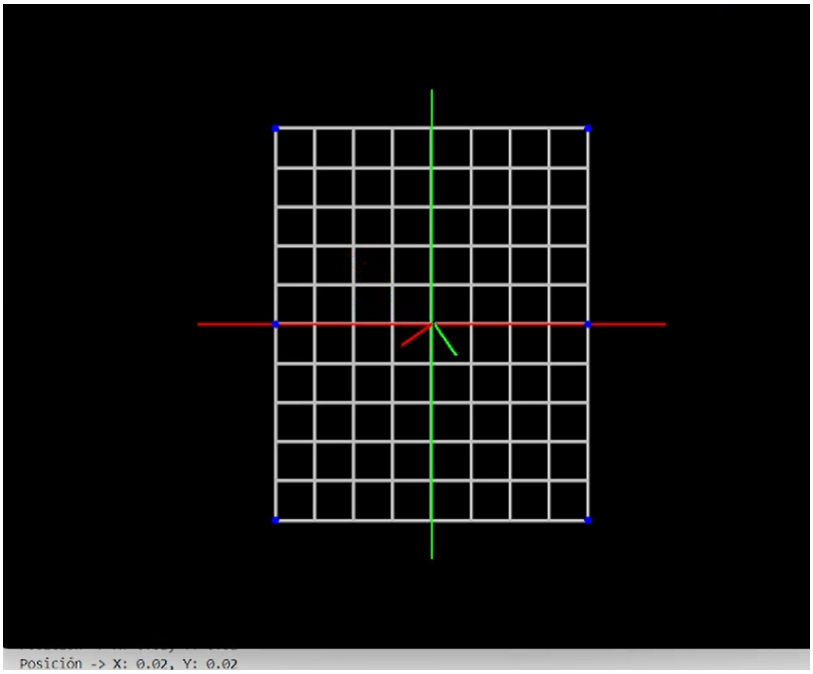


Nota. Elaboración propia.

13.3. Implementación en tiempo real

Como se pudo observar en las últimas secciones se logró implementar el filtro complementario normal y Butterworth con similitudes a las trayectorias de Optitrack altas. Una vez logrado esto se procedió a implementar un entorno de visualización del módulo de navegación de modo que, al momento de estar conectado a la red de la computadora, se pueda ver su posición en tiempo real. Se emplea homografía proyectiva así como filtros complementarios para posición y orientación, siempre en dos dimensiones. Esta implementación es el resultado de la calibración y uso de filtros que permiten tener una estimación de la posición dentro y fuera del entorno de pruebas. En la Figura 50, se presenta el módulo de navegación en el entorno de PyGame.

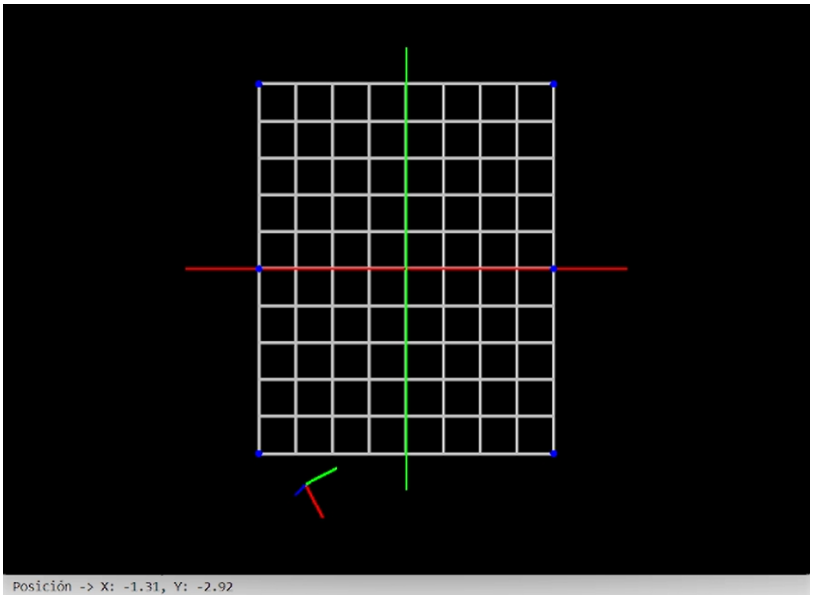
Figura 50.
Entorno de PyGame para visualización de módulo de navegación en el centro.



Nota. Elaboración propia.

Este entorno tiene la capacidad de poder mover la cámara que apunta al centro de la grilla. También desplazamiento en ejes x,y para poder buscar el Módulo de Navegación en caso esté fuera de rango. También se puede observar que tiene puntos azules que se colocaron manualmente para describir la posición de los trípodes o módulos MDEK1001; no es importante sino es para fines de visualización. Otro detalle importante son los ejes, siendo el eje x de color rojo y el eje y de color verde. Esta figura es una representación del plano cartesiano donde en el primer cuadrante (esquina superior izquierda) los dos ejes son positivos y así con el resto de los cuadrantes con sus respectivos signos.

Figura 51.
Entorno de PyGame para visualización de módulo de navegación fuera de la plataforma.



Nota. Elaboración propia.

En la Figura 51 se muestra que es posible poder utilizar el módulo fuera del entorno. Esto es ideal para aplicaciones que requieran mayor espacio y monitoreo o bien estudio.

CAPÍTULO 14

Conclusiones

- Se cumplió el primer objetivo (integración de sensores adicionales al sistema de navegación local) por medio de una PCB que contiene una IMU para correcciones por medio de filtros complementarios; también, una etiqueta UWB y cámaras Optitrack para calibración por medio de homografía. Este objetivo también se logró al elaborar el carro de pruebas, donde se monta el módulo de navegación y el marker de Optitrack.
- Se cumplió el segundo objetivo al implementar filtros que mejoran la precisión. Con Optitrack, se realizó una homografía proyectiva para calibración mejorando la exactitud en dos dimensiones.
- Se cumplió con el tercer objetivo en cuanto a la realización de experimentos con módulos UWB y acelerómetro en condiciones LOS para un sistema de coordenadas de dos dimensiones.
- Se cumple el objetivo general al implementar un sistema de navegación local en dos dimensiones al tener el montaje de módulos UWB en un ambiente de línea de vista.
- El cumplimiento de los objetivos permitió la implementación en tiempo real para localización del módulo de navegación por medio de wifi TCP.

CAPÍTULO 15

Recomendaciones

- **Para el filtro de Kalman en navegación local, se recomienda:**
  - Programar el ESP32 en el IDE de Arduino utilizando el filtro de Madgwick para obtener la matriz de rotación R.
  - Incorporar el eje z mediante un barómetro (*módulo MPU9250 GY-91*) para estimar la altura.
  - Configurar módulos UWB a diferentes alturas para capturar el eje z de mejor manera.
- **Para el modelado de ruido con el filtro de Kalman, se recomienda:**
  - Realizar pruebas experimentales para capturar el ruido de los sensores y obtener modelo de ruido. Esto se puede realizar de la siguiente forma:
    - Mantener el módulo de navegación estático y someterlo a perturbaciones controladas (golpes leves).
    - Colocar el módulo de navegación a una altura fija sobre el suelo para caracterizar el ruido del barómetro.
- **Para la visualización en tiempo real 3D, se recomienda:**
  - Adaptar el código de visualización en PyGame u otra plataforma para mostrar la trayectoria al incluir el eje z en un entorno 3D.

- **Para la comunicación bluetooth, se recomienda:**
  - Implementar comunicación bluetooth como alternativa al wifi TCP para aplicaciones de exterior o donde no haya red wifi.
- **Para la calibración de módulos UWB sin Optitrack:**
  - Diseñar un método de calibración para los módulos UWB sin depender del sistema Optitrack, permitiendo su uso en espacios abiertos o fuera de entornos controlados.
- **Para las pruebas dinámicas, se recomienda:**
  - Realizar pruebas dinámicas en condiciones NLOS
  - Implementar un algoritmo de SLAM con fusión de sensores para la generación de un mapa mientras se realiza la localización en tiempo real.
  - Realizar pruebas dinámicas con robots móviles como Pololu 3pi+, drones o rovers para capturar su dinámica y generar trayectorias en 2D y 3D.

CAPÍTULO 16

Referencias

[1] C. N. de León Bercián, "*Implementación de un sistema de localización en tiempo real en espacios cerrados para el robot Rover UVG*," 2022, Tésis de Licenciatura.

[2] Qorvo. "*MDEK1001 Ultra-Wideband (UWB) Transceiver Development Kit.*" Accessed: [2024-03-26].

[3] M. Electronics. "MDEK1001," visitado 26 de mar. de 2024. dirección: https://www.mouser.mx/ProductDetail/Qorvo/MDEK1001?qs=TiOZkKH1s2TlRZBi6MtMNg%3D%3D.

[4] S. Mischie e I. A. Munteanu, "Using Ultrawideband Technology to Control a Car to Reach Its Destination," Engineering Proceedings, vol. 56, n.º 1, 2023, ISSN: 2673-4591. DOI: 10.3390/ASEC2023-16331. dirección: https://www.mdpi.com/2673-4591/56/1/156.

[5] A. R. Jiménez y F. Seco, "Improving the Accuracy of Decawave's UWB MDEK1001 Location System by Gaining Access to Multiple Ranges," Sensors, vol. 21, n.º 5, 2021, ISSN: 1424-8220. DOI: 10.3390/s21051787. dirección: https://www.mdpi.com/1424-8220/21/5/1787.

[6] Federal Communications Commission. "*Revision of Part 15 of the Commission's Rules Regarding Ultra-Wideband Transmission Systems*," visitado 9 de mayo de 2024. dirección: https://docs.fcc.gov/public/attachments/FCC-00-163A1.pdf.

[7] M. Viot et al., UltraWideband for Dummies, E. Kuball, ed. Wiley, 2021, ISBN: 978-1-119-80960-9. visitado 9 de mayo de 2024. dirección: https://www.qorvo.com/design-hub/ebooks/ultra-wideband-for-dummies.

[8] OptiTrack. "*PrimeX 41 Camera Manual.*" Accedido el 2024-05-09. dirección: https://docs.optitrack.com/hardware/cameras/ethernet-cameras/primex-41.

[9] OptiTrack. "*Optitrack PrimeX Camera Setup.*" Accedido el 2024-05-09. dirección: https://optitrack.com/systems/#robotics/primex-41/6.

[10] P. D. Groves, Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Second Edition. Artech House, abr. de 2013.

[11] B. Friedland, Control System design: An Introduction to State-Space Methods. Courier Corporation, mar. de 2005.

[12] F. Outamazirt, L. Yan, F. Li y A. Nemra, "Solving the UAV localization problem using a Smooth Variable Structure Filtering," en 2015 IEEE Aerospace Conference, 2015, págs. 1-12. DOI: 10.1109/AERO.2015.7119259.

[13] S.-H. Lee y J. Song, "Regularization-Based Dual Adaptive Kalman Filter for Identification of Sudden Structural Damage Using Sparse Measurements," Applied Sciences, vol. 10, n.º 3, 2020, ISSN: 2076-3417. DOI: 10.3390/app10030850. dirección: https://www.mdpi.com/2076-3417/10/3/850.

[14] R. W. Schafer, "What Is a Savitzky-Golay Filter? [Lecture Notes]," IEEE Signal Processing Magazine, vol. 28, n.º 4, págs. 111-117, 2011. DOI: 10.1109/MSP.2011.941097.

[15] M. Sadeghi y F. Behnia, "Optimum window length of Savitzky-Golay filters with arbitrary order," arXiv: Signal Processing, 2018. dirección: https://api.semanticscholar.org/CorpusID:126330871.

[16] R. Hartley y A. Zisserman, Multiple View Geometry in Computer Vision, 2.ª ed. Cambridge, UK: Cambridge University Press, 2003, ISBN: 978-0521540513.

[17] MathWorks, Create a Gallery of Transformed Images, Accessed: 2024-11-19, 2024. dirección: https://www.mathworks.com/help/images/creating-a-gallery-of-transformed-images.html.

[18] R. Mahony, T. Hamel y J.-M. Pflimlin, "Nonlinear Complementary Filters on the Special Orthogonal Group," IEEE Transactions on Automatic Control, vol. 53, n.º 5, págs. 1203-1218, 2008. DOI: 10.1109/TAC.2008.923738.

[19] Y. Bar-Shalom, X.-R. Li y T. Kirubarajan, Estimation with Applications to Tracking and Navigation. ene. de 2002. DOI: 10.1002/0471221279. dirección: https://doi.org/10.1002/0471221279.

[20] T. Kailath, A. H. Sayed y B. Hassibi, Linear estimation. Pearson, ene. de 2000.

[21] D. Kumar y V. Divakar, "Spy Robot Using Self-Balancing Algorithm with Pan and Tilt Control of Camera," International Journal of Innovative Research in Computer and Communication Engineering, vol. 3, n.º 9, págs. 8395-8402, 2015.

[22] Grobotronics. "*ESP32 Development Board - DEVKIT V1.*" Accedido el 2024-03-09. dirección: https://grobotronics.com/esp32-development-board-devkit-v1.html?sl=en.

[23] Espressif Systems, ESP32 Series Datasheet, Accedido el 2024-09-03, 2024. dirección: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.

[24] D. M. Harris y S. L. Harris, Digital design and computer architecture. Elsevier, ene. de 2013.

[25] J. Wu, A Basic Guide to I2C, Application Report SBAA565, Texas Instruments, nov. de 2022. dirección: https://www.ti.com/lit/an/sbaa565/sbaa565.pdf.

[26] T. InvenSense, MPU-9250 Product Specification Revision 1.1, Accessed: 2024-09-09, 2014. dirección: https://invensense.tdk.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf.

[27] Components101. "*MPU9250 9-DOF MEMS Sensor Module.*" Accessed: 2024-11-02. dirección: https://components101.com/sensors/MPU9250-9-dof-mems-sensor-module-datasheet-pinout-features-working.

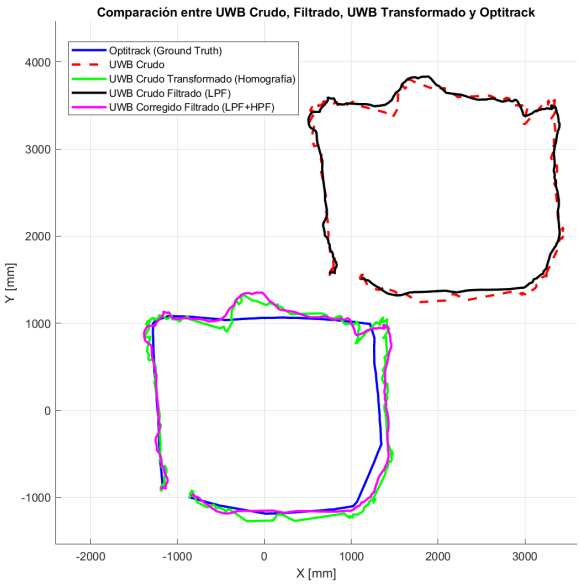
CAPÍTULO 17

Anexos

17.1. Pruebas dinámicas adicionales

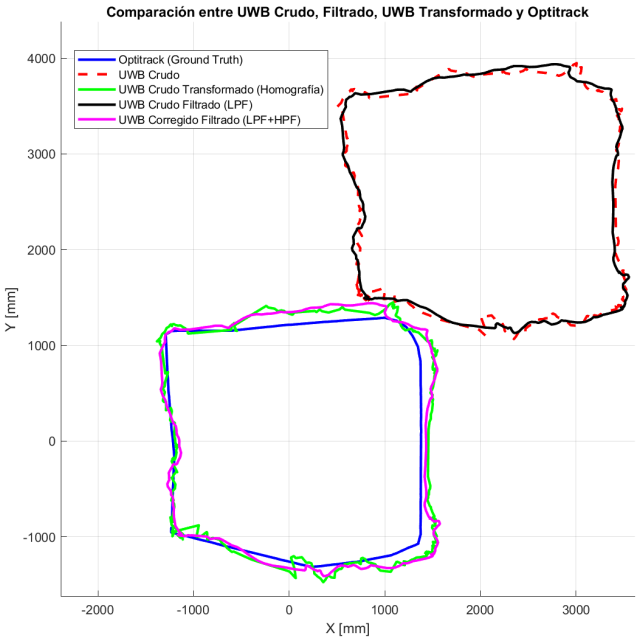
Nota: El nombre "Trayectoria cuadrada 1" por ejemplo, pertenece a un dataset, en el cual se pueden comparar ambas arquitecturas de filtrado. Primero, se muestran 5 trayectorias con filtro complementario normal y luego con Butterworth.

Figura 52.
Trayectoria cuadrada 1 con filtro complementario normal con $\mathbb{R}^2 = 0.9695$ *y media Qf* $= 69.24\,\%$.



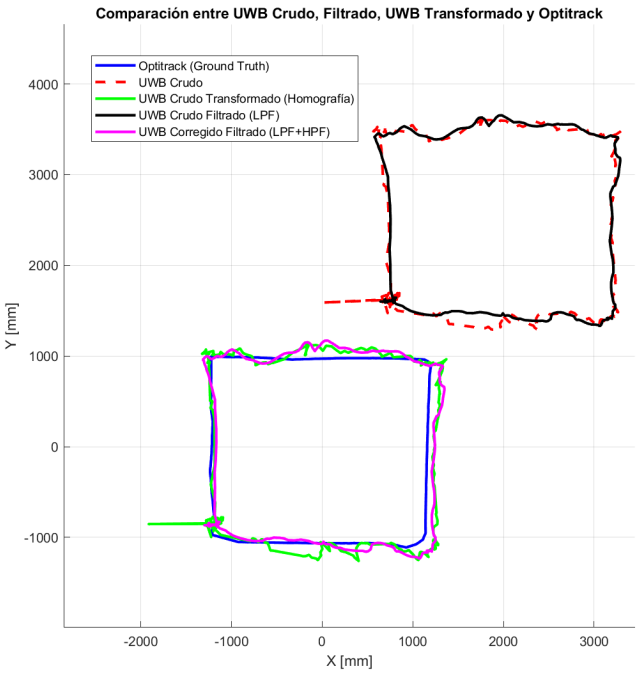
Nota. Elaboración propia.

Figura 53.
Trayectoria cuadrada 2 con filtro complementario normal con $\mathbb{R}^2 = 0.9729$ *y media Qf = 68.27 %.*



Nota. Elaboración propia.

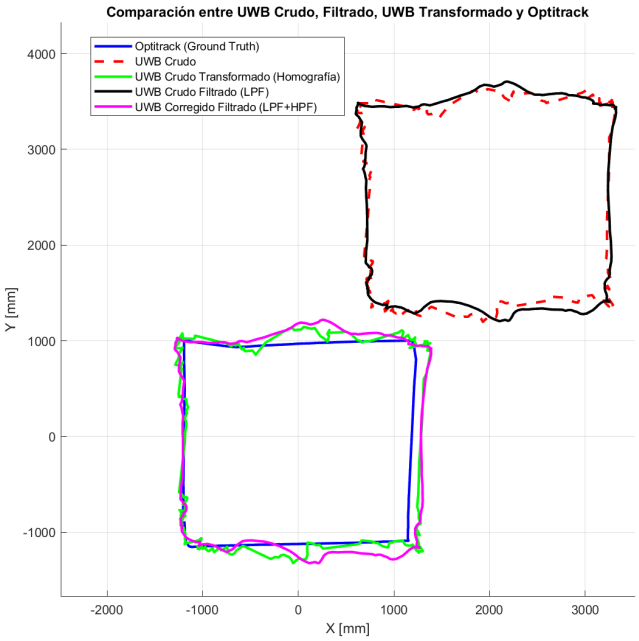
Figura 54.
Trayectoria cuadrada 3 con filtro complementario normal con $\mathbb{R}^2 = 0.9209$ *y media Qf = 69.89 %.*



Nota. Elaboración propia.

Figura 55.

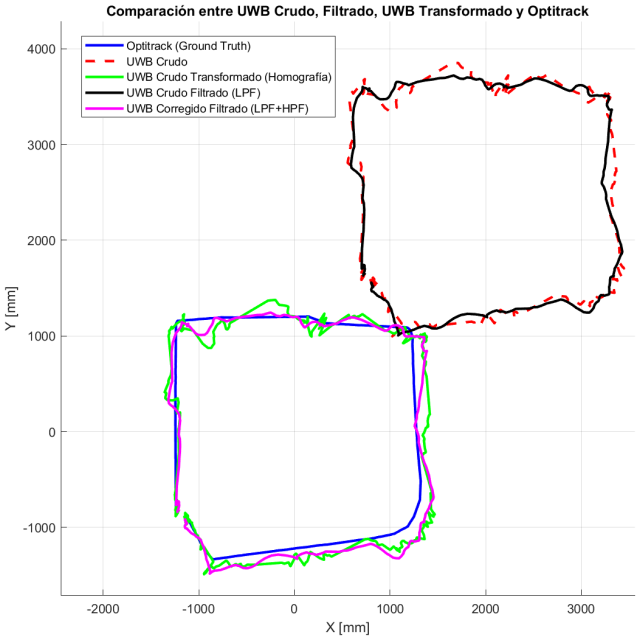
Trayectoria cuadrada 4 con filtro complementario normal con $\mathbb{R}^2 = 0.9591$ *y media* $Qf = 70.69\,\%$.



Nota. Elaboración propia.

Figura 56.

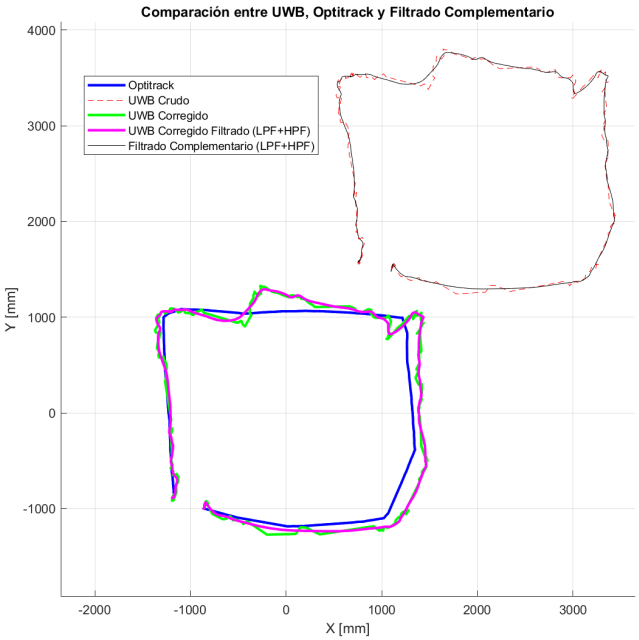
Trayectoria cuadrada 5 con filtro complementario normal con $\mathbb{R}^2 = 0.9784$ *y media* $Qf = 70.76\,\%$.



Nota. Elaboración propia.

Figura 57.

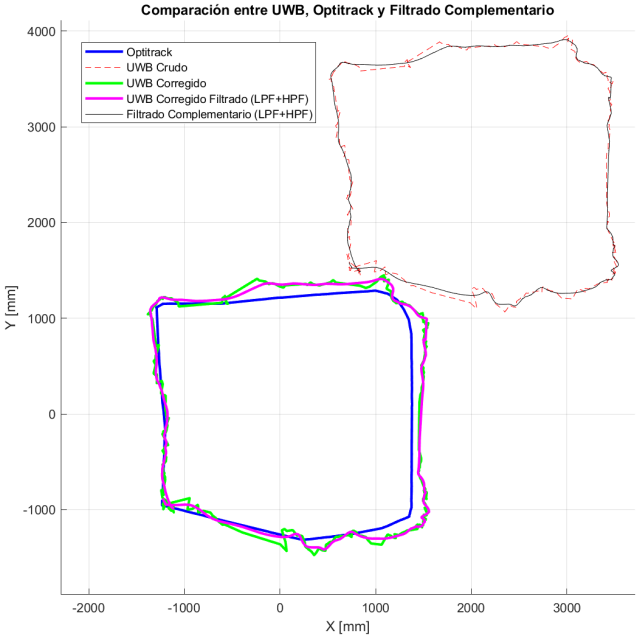
Trayectoria cuadrada 1 con filtro complementario Butterworth con $\mathbb{R}^2 = 0.9895$ *y media Qf = 69.24 %.*



Nota. Elaboración propia.

Figura 58.

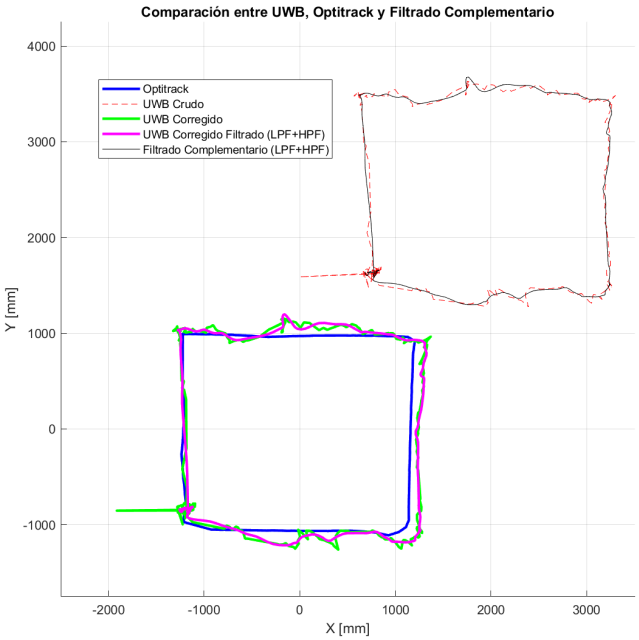
Trayectoria cuadrada 2 con filtro complementario Butterworth con $\mathbb{R}^2 = 0.9902$ *y media Qf = 68.27 %.*



Nota. Elaboración propia.

Figura 59.

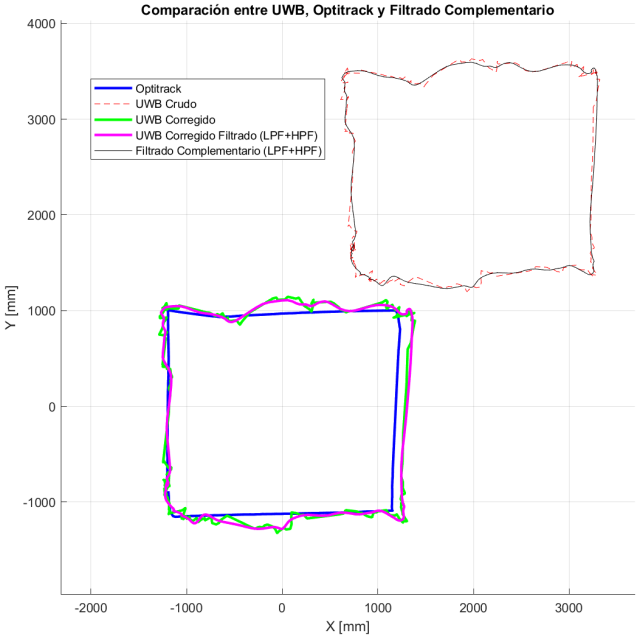
Trayectoria cuadrada 3 con filtro complementario Butterworth con $\mathbb{R}^2 = 0.9437$ *y media Qf = 69.89 %.*



Nota. Elaboración propia.

Figura 60.

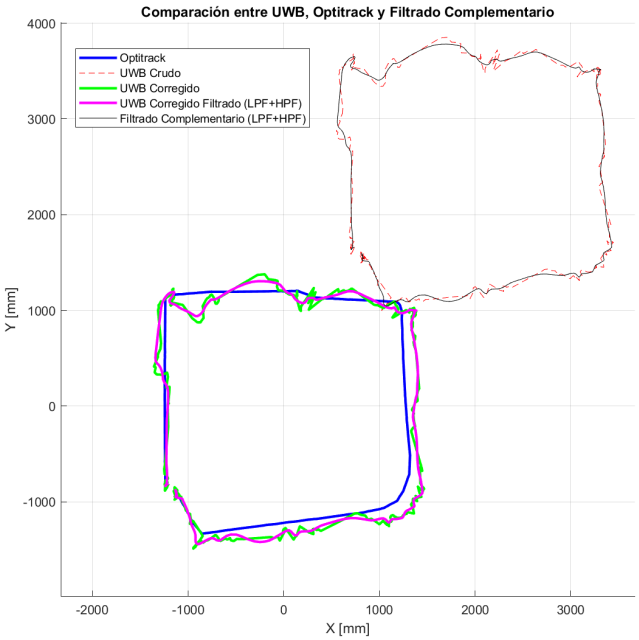
Trayectoria cuadrada 4 con filtro complementario Butterworth con $\mathbb{R}^2 = 0.9895$ *y media Qf = 70.69 %.*



Nota. Elaboración propia.

Figura 61.

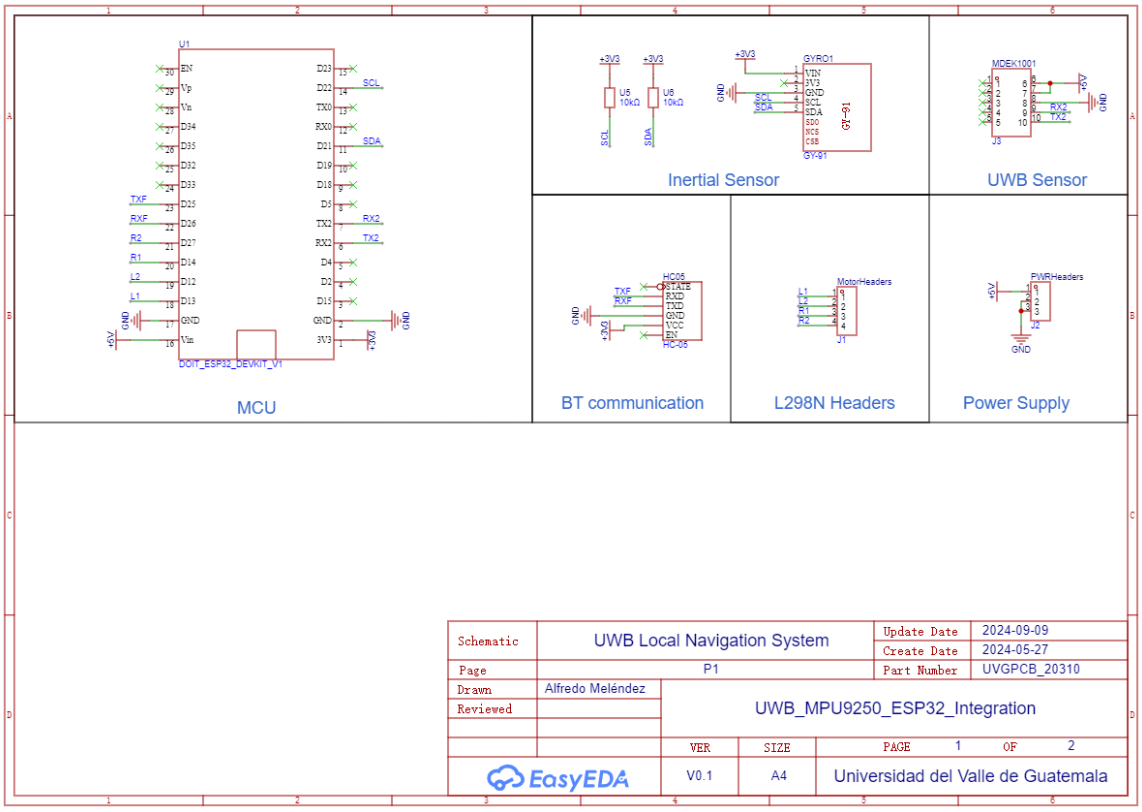
Trayectoria cuadrada 5 con filtro complementario Butterworth con $\mathbb{R}^2 = 0.9922$ *y media Qf* $=$ *70.76 %.*



Nota. Elaboración propia.

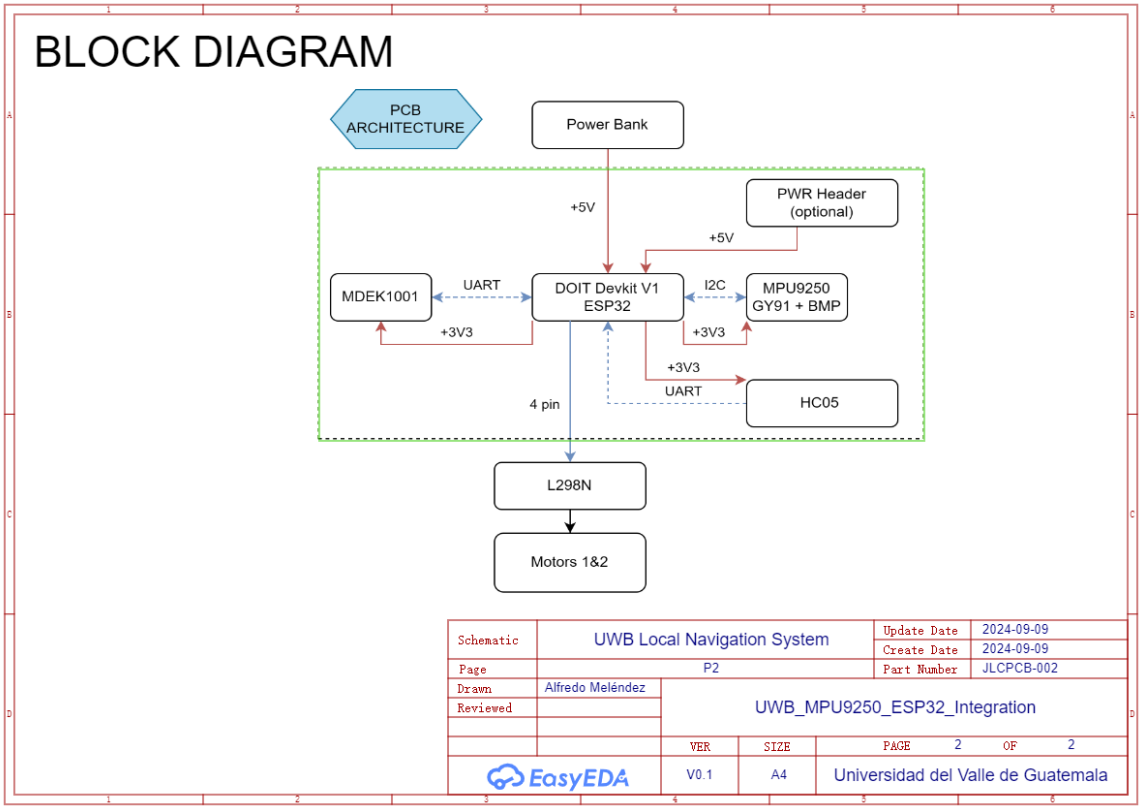
17.2. Esquemáticos e implementación PCB

Figura 62.
Esquemático de placa PCB para integración de sensores y control de carro.



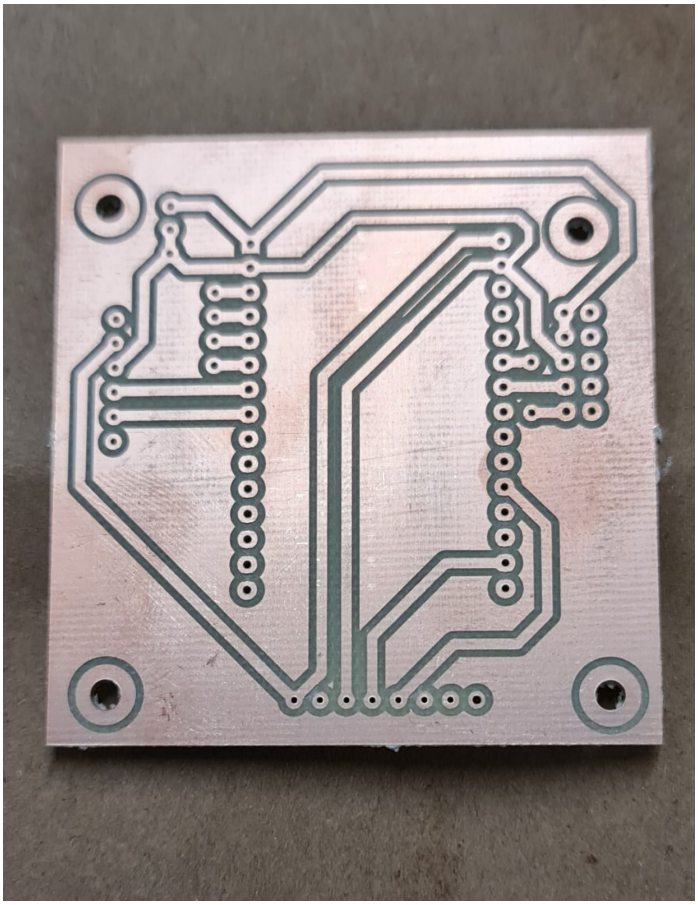
Nota. Elaboración propia.

Figura 63.
Diagrama de bloques PCB.



Nota. Elaboración propia.

Figura 64.
Primera iteración de PCB fabricada.



Nota. Elaboración propia.

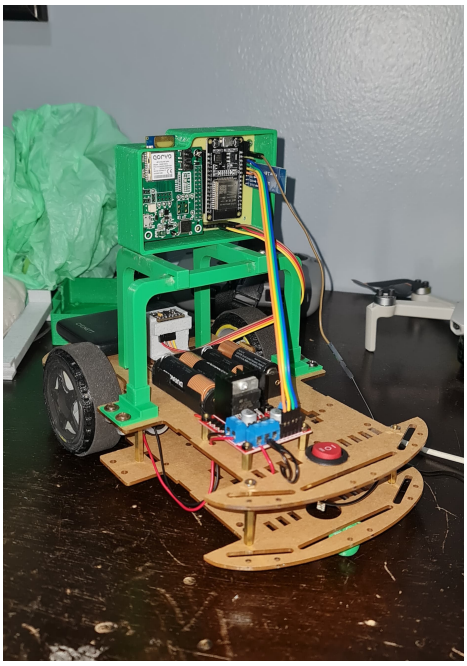
Figura 65.
PCB fabricada con módulos montados.

Nota. Elaboración propia.

17.3. Carro de pruebas e iteraciones

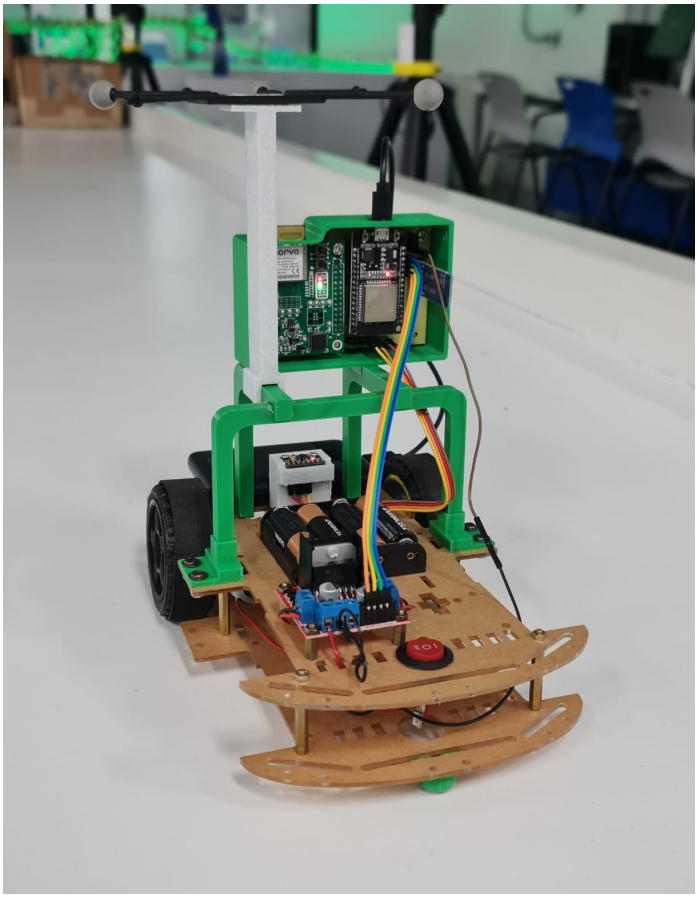
La primera iteración del carro se encuentra en la Figura 27 y la iteración final está en la Figura 32. Para la Figura 66, se tiene la segunda iteración con un módulo de navegación montado, pero con la IMU fuera del mismo.

Figura 66.
Iteración 2 del carro antes del montaje completo.



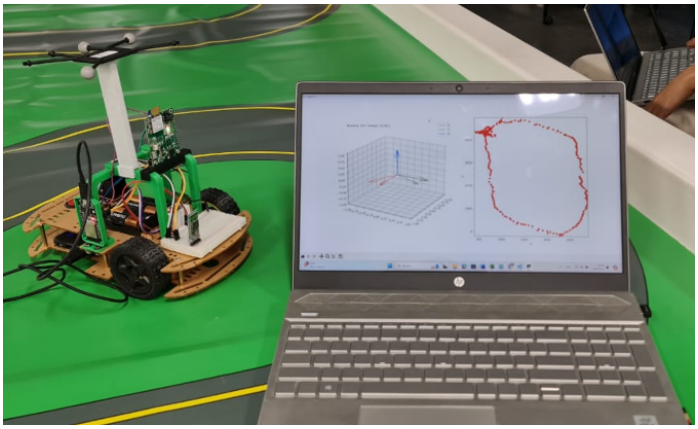
Nota. Elaboración propia.

Figura 67.
Iteración 2 del carro a control remoto con el montaje completo tomando muestras de Optitrack con marcador y el módulo de navegación en la plataforma del laboratorio Robotat en UVG.



Nota. Elaboración propia.

Figura 68.
Iteración 2 del carro con la primera interfaz de visualización de trayectorias y orientación únicamente de la etiqueta MDEK1001.

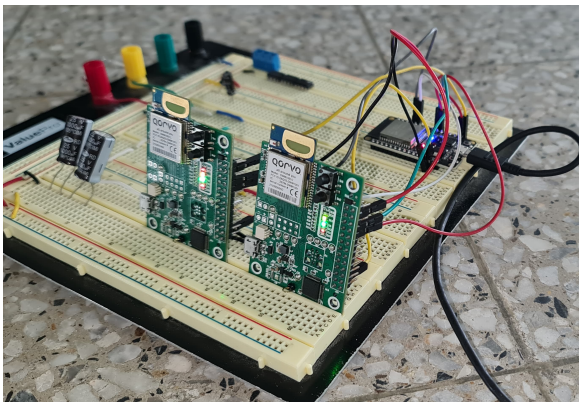


Nota. Elaboración propia.

17.4. Pruebas previas

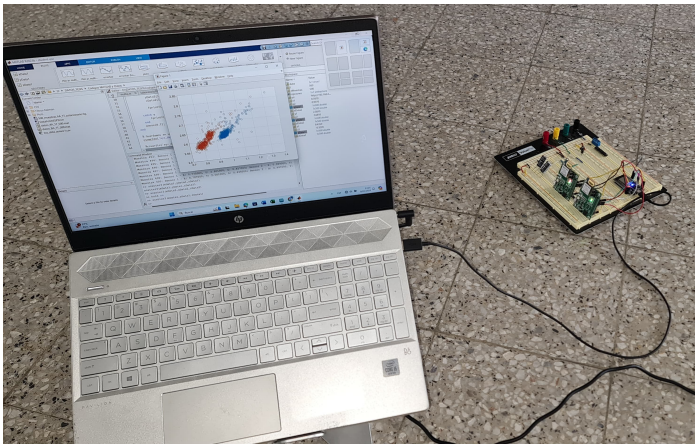
En esta sección se muestran las pruebas hechas previo a la implementación final. Esto consiste en distintas formas de implementar los módulos UWB y otros arreglos durante el proceso.

Figura 69.
Dos etiquetas con 8 anclas funcionando en paralelo y un DOIT DevKit V1.



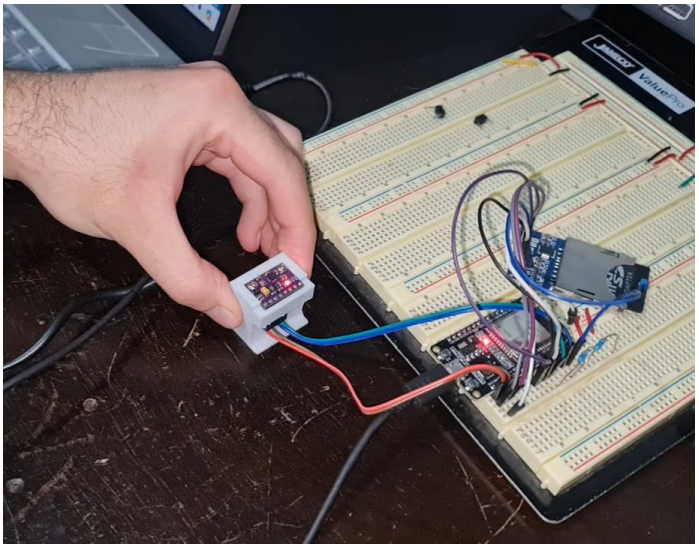
Nota. Elaboración propia.

Figura 70.
Nube de puntos de 2 etiquetas estáticas.



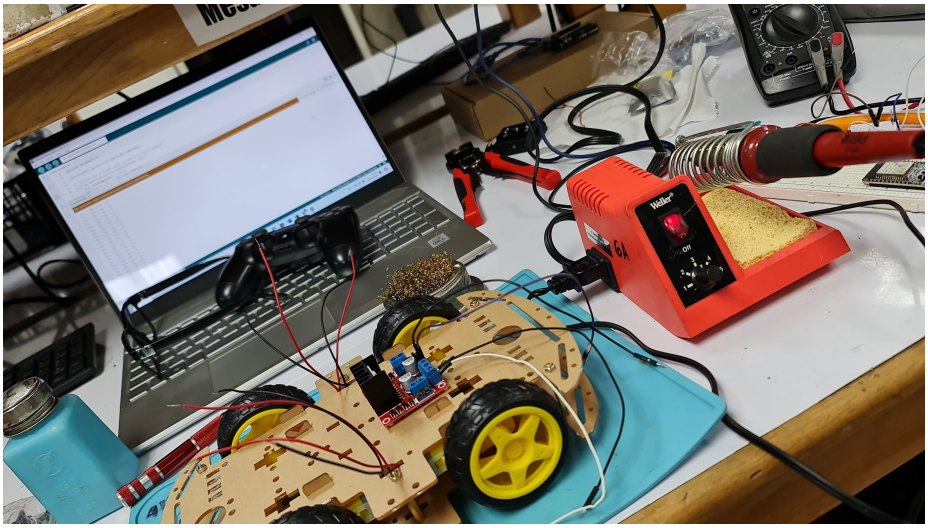
Nota. Elaboración propia.

Figura 71.
Test con IMU MPU9250-GY91 y cubo de calibración para filtros complementarios en roll, pitch y yaw guardados en memoria SD para análisis.



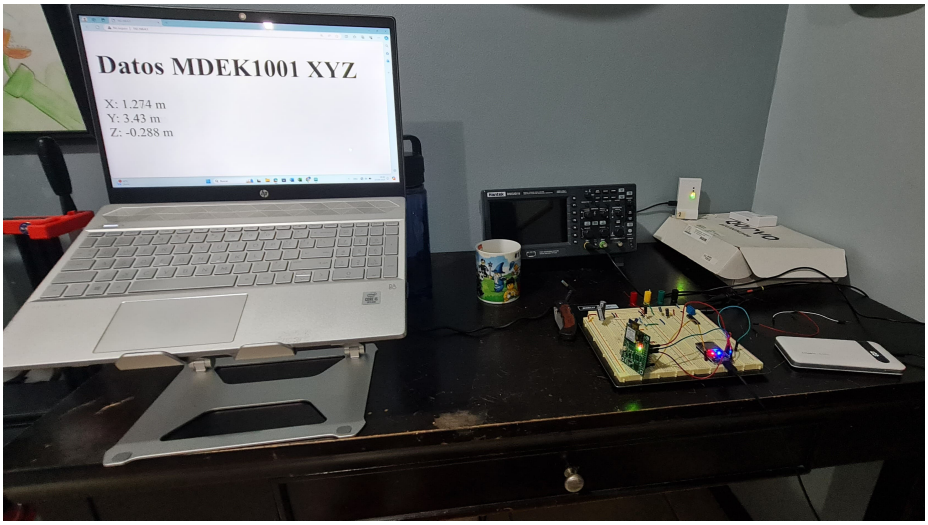
Nota. Elaboración propia.

Figura 72.
Implementación de carro a control remoto usando DualShock4 de Playstation conectado al ESP32.



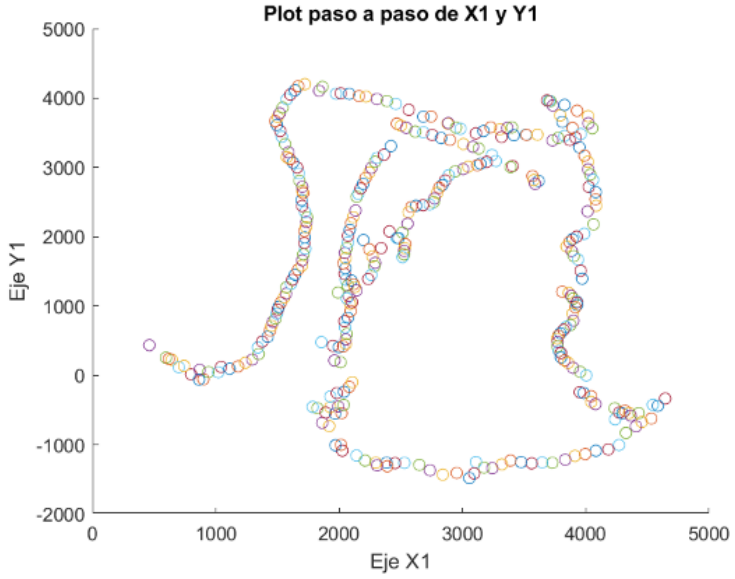
Nota. Elaboración propia.

Figura 73.
Primera implementación de envío de datos mediante wifi TCP en una página del ESP32 configurado como Soft-AP.



Nota. Elaboración propia.

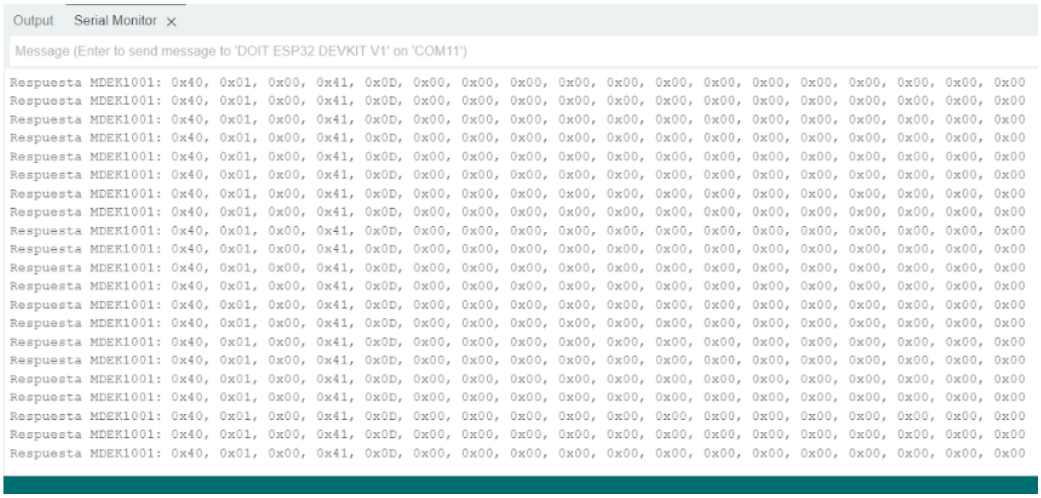
Figura 74.
Primer plot de datos extraídos de la terminal de la aplicación DRTLS como archivo .log y decodificados mediante parsing con un script de Python y graficado en Matlab desde el archivo .csv generado.



Nota. Elaboración propia.

Figura 75.

Primera obtención de datos de ubicación sin decodificar en la terminal del IDE de Arduino.



Nota. Elaboración propia.

CAPÍTULO 18

Glosario

BLE: bluetooth de baja energía.

DSSS: espectro ensanchado por secuencia directa.

GPIO: entrada o salida de propósito general.

GPS: sistema de posicionamiento global.

GT: referencia real.

I2C: interfaz de circuito integrado.

IEEE: Instituto de Ingenieros Eléctricos y Electrónicos.

IMU: unidad de medición inercial.

LNS: sistema de navegación local.

LOS: línea de vista o visión.

LSB: byte menos significativo.

MSB: byte más significativo.

NLOS: sin línea de vista o visión.

OFDM: multiplexación por división de frecuencia ortogonal.

PCB: placa de circuito impreso.

PDoA: diferencia de fase de llegada.

RFID: identificación por radiofrecuencia.

RTLS: sistema de localización en tiempo real.

SEL: sistema de ecuaciones lineales.

SPI: interfaz periférica serial.

TCP: transmisión confiable de datos.

TDoA: diferencia de tiempo de llegada.

TLV: formato correspondiente a tipo, longitud y valor de una cadena de datos.

ToF: tiempo de vuelo.

TWR: rango de dos vías.

UART: receptor-transmisor asíncrono universal.

UWB: ancho de banda ultra ancha.