

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



Evaluación de Módulos *Ultra Wide Band (UWB)* para
Sistemas de Navegación Local

Protocolo de trabajo de graduación presentado por Alfredo Andres
Meléndez Mendoza, estudiante de Ingeniería Mecatrónica

Guatemala,

2024

Resumen

El proyecto propuesto, trata de crear un sistema de navegación local 2D por medio de la tecnología de banda ultra ancha (UWB) con los módulos MDEK1001 de la empresa Qorvo en conjunto con un acelerómetro MPU9250. Esto es dado que hay dispositivos presentan una alternativa para localización en una escala menor y en donde la precisión mejora respecto de otros sistemas de navegación como por ejemplo el GNSS a mayor escala.

El objetivo principal es la evaluación de los módulos con la finalidad de obtener posición y orientación en un entorno 2D y compararlos con un *Ground Truth* de un sistema de cámaras Optitrack en el *Robotat*. Los datos obtenidos de los sensores UWB y el acelerómetro son fusionados para ser pasados por un Filtro Kalman de modo que se pueda mejorar la exactitud y precisión, y con ello analizar las pruebas para implementarlo en tiempo real haciendo uso de IoT para recepción de datos.

Dentro de las pruebas se encuentran las estáticas y dinámicas, cada una con su respectiva configuración de montaje y en condiciones de línea de visión libre (LOS). Las pruebas dinámicas se realizan con un carro controlado de manera remota por bluetooth con un PCB que integra los módulos y un marker. Finalmente se analiza 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.

Antecedentes

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 lo que es 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. [3]

En el proyecto se llevó a cabo lo que es el estudio RTLS (Real Time Location System por sus siglas en inglés) en donde se obtuvieron resultados de tasa de transferencia de datos siendo 100 ms/Hz en una configuración de la aplicación Decawave DRTLS Manager y la precisión del posicionamiento con datos obtenidos mediante el protocolo TLV (Type-Length-Value por sus siglas en inglés). 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 uno en etiqueta para poder medir posición en una figura rectangular.

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.

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 no usara GPS o sensores iniciales debido a las fallas presentadas en cuanto a la precisión por lo cual se optó por hacer uso de la tecnología 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 en la etiqueta 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 (Bluetooth Low Energy por sus siglas en inglés) hacia un teléfono inteligente. Una vez la comunicación se estableció y colocadas las anclas en una posición cuadrada. Como se observa en la Figura 2 mediciones estáticas con desviaciones estándar de hasta 59mm en el eje x y en el eje y hasta 24mm donde se comprobó que se encuentra dentro lo establecido por el fabricante con precisión de ± 10 mm. Además se utilizó un factor de calidad *qf* del 75 %.

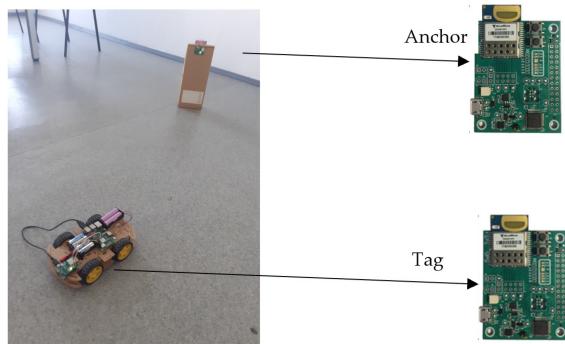


Figura 2: Entorno de conexión de anclas y etiquetas. [4]

Luego en pruebas dinámicas se colocó una coordenada inicial y una final para saber la precisión de la ubicación al alcanzar dicha coordenada. En algunos casos los valores llegaban a exceder la coordenada 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 lineal. También se discute que en algunos casos se excede la precisión de ± 10 mm 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.

Mejora de la precisión de posicionamiento del Módulo MDEK1001 al tener múltiples rangos y en interiores e 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 (Line of Sight por sus siglas en inglés) en una oficina. Se hizo uso de un instrumento de medición con ± 1 mm de precisión siendo el 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).

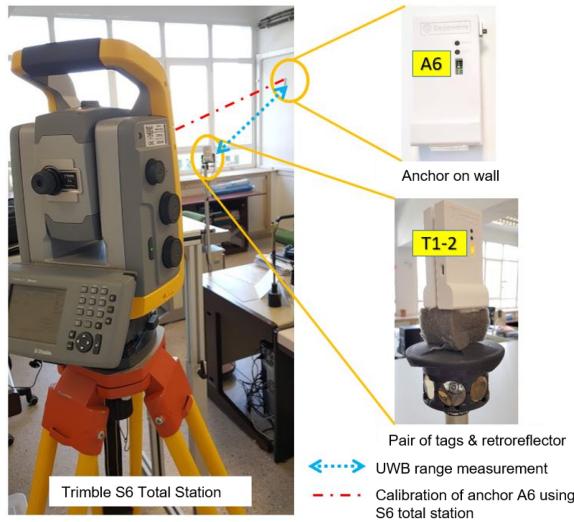


Figura 3: Montaje de experimento con ground truth. [5]

En la elaboración del filtro EKF se plantearon seis variables de estado siendo posición en ejes xyz, y 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 por tanto 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 el Trimble S6 y para ambas redes correspondientes a cada etiqueta y con los muestreros se pudo implementar el EKF con desviación estándar menor a 0.2 m. 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 pretendió tener un entorno NLOS (None-Line-of-Sight o linea de visión nula por sus siglas en inglés) en el cual hay obstáculos como paredes, muebles o bien elementos metálicos, entre otros objetos que dificulten la comunicación del UWB. Para este experimento se consideró el uso de IMUs colocadas en los pies como GT que, de acuerdo con el autor, no es fiable pero es una opción útil al no poder usar el Trimble S6 en dicho ambiente. Los módulos UWB colocados en la cabeza se usaron para medir trayectorias dentro del apartamento en conjunto con las IMUs y se encontró una desviación estándar de 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 sin ruido alguno o valores atípicos con mayor precisión. El estudio concluye que los módulos MDEK1001 son más precisos en ambientes de LOS haciendo uso de un filtro EKF para dispositivos con poca cobertura satelital o sistema de posicionamiento lo cual lo hace escalable para soluciones corporativas.

Justificación

En distintos estudios de sensores de posición 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 iniciales, visión por computadora, híbridos y sonido.

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 rangos los cuales se dan por cuatro anclas respecto de una etiqueta a una frecuencia mínima de 100 Hz. 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, logística de almacenes y navegación de partículas.

En trabajos de graduación previamente hechos en la Universidad del Valle de Guatemala y estudios con estos 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 o simplemente los datos luego de ser recolectados. Este problema se da a raíz que la información se puede obtener pero no es lo suficientemente práctico para trabajar.

El presente trabajo de graduación tiene el propósito de facilitar la obtención de datos en tiempo real sin necesidad de estar utilizando una computadora portátil conectada a un módulo para extraer datos manualmente sino automática utilizando microcontroladores que se comuniquen por UART. Se pretende implementar orientación para que tenga mayor utilidad para navegación. Finalmente se desea filtrar los datos obtenidos para mejorar la precisión de posicionamiento.

Objetivos

Objetivo General

- Implementar un sistema de navegación local en un entorno de 2 dimensiones con módulos UWB.

Objetivos Específicos

- Integrar sensores adicionales en el Sistema de Navegación Local.
- Realizar experimentos con módulos UWB y acelerómetro de 9DOF en condiciones controladas de LOS en 2D.
- Implementar filtros para mejorar la precisión y exactitud del posicionamiento en dos dimensiones.

A continuación se va a detallar un listado de subtareas para poder alcanzar los objetivos propuestos.

1. Entender la comunicación por medio de UART usando computadora y decodificador de oscilloscopio.
2. Emplear el formato TLV para obtención de datos.
3. Añadir comunicación usando el formato TLV por medio de una placa de desarrollo para automatizar la obtención de datos.
4. Crear un punto de acceso y subir los datos a internet.
5. Fabricación de carro controlado remotamente mediante bluetooth y placa PCB para integración de sensores
6. Realizar pruebas estáticas y dinámicas en condiciones LOS en un entorno controlado como el *Robotat* de la Universidad del Valle de Guatemala. Este debe hacerse con cuatro anclas hasta once de ellas con una etiqueta o dos de acuerdo con el montaje deseado
7. Con los datos obtenidos realizar un filtrado del posicionamiento.
8. Organizar información y pruebas hechas para discutir observaciones en función de los resultados.

Marco teórico

Tecnología UWB

La tecnología UWB (banda ultra-ancha por sus siglas en inglés) 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 Bluetooth de baja energía (BLE), GPS o Wi-Fi 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.

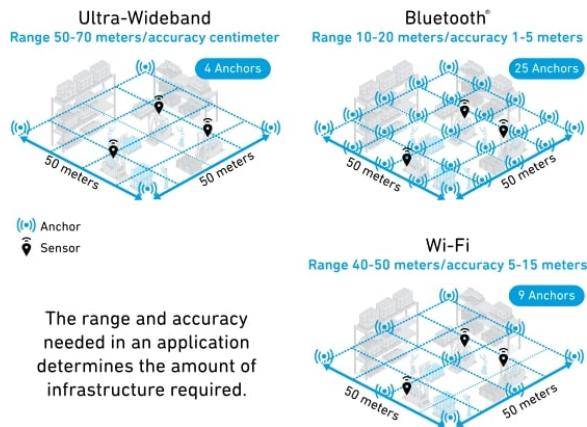


Figura 4: Precisión en distintos sistemas de localización. [7]

Una vez observada la comparación se tiene que este sistema es más preciso que 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 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 milisegundo 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 móviles y un dispositivo que recopila y envía 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 mencionando 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.

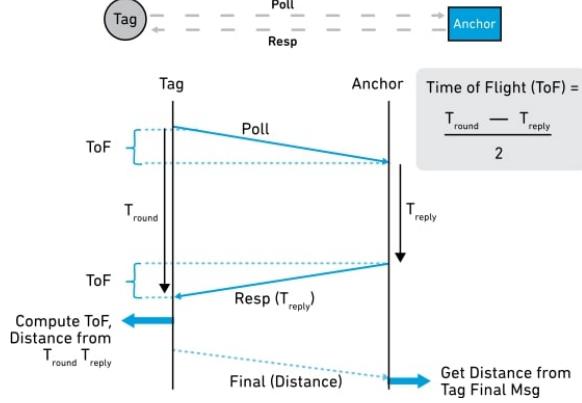


Figura 5: Representación de tiempo de vuelo (ToF) con rango bidireccional (TWR). [7]

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 cables o por radiofrecuencia para sincronización de anclas, siendo en total 3 señales. Por otro lado tenemos el TDoA inverso en el cual las anclas emiten señales de parpadeo y la etiqueta es la que manda 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.

Sistema de captura de movimiento con cámaras Optitrack

El sistema MoCap de Optitrack [8] consiste principalmente en cámaras de captura de movimiento con iluminación infrarroja integrada de 850nm para poder detectar marcadores o bien esferas con pintura reflectiva para poder 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 $\pm 0.2\text{mm}$ con una latencia menor a 10 ms y rotación con ± 0.1 grados para el caso de la serie Prime. En la Figura 8 se aprecia un montaje de 6 cámaras PrimeX 41.

Estos sistemas se caracterizan por la facilidad de uso en cuanto a su interfaz, conectividad con otras 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 y 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 de los sistemas de captura de Optitrack son variadas tal como la producción virtual, biomecánica, realidad virtual, animaciones y robótica. En la Figura 8. se encuentra la forma en la que se conecta un sistema Optitrack de cámaras PrimeX para una cantidad menor a 11 cámaras.

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	<ul style="list-style-type: none"> • Defeats relay attacks • Compliance with CCC specifications 	<ul style="list-style-type: none"> • Seamless implementation • No need for anchor syncing/backhaul channel 	<ul style="list-style-type: none"> • Very low power tag • High amount of tags can be used 	<ul style="list-style-type: none"> • 'GPS-like' • Infinite amount of tags can be used • Low-power anchors (can be battery-powered) 	<ul style="list-style-type: none"> • Peer-to-peer localization • Polar coordinates • Spherical coordinates possible with 2 instances of ICs/module
Applications	<ul style="list-style-type: none"> • Distance measurements • Door locks physical access control • Car keyless entry • Car immobilizers 	<ul style="list-style-type: none"> • 2D/3D asset & people tracking • Industry 4.0 • Warehouse logistics • Healthcare • Retail 	<ul style="list-style-type: none"> • 2D/3D asset & people tracking • Industry 4.0 • Warehouse & logistics • Healthcare • Agriculture 	<ul style="list-style-type: none"> • 2D/3D navigation • Robotics • Valet parking • Lone worker protection 	<ul style="list-style-type: none"> • 'Follow me' robots, drones, etc. • Platooning (trucks) • Door locks & access control

*RTLS Real Time Location Systems

** Indicative - actual battery lifetime figures depend on exact RF & RTLS configuration and on battery type & capacity.

Figura 6: Topologías de comunicación UWB. [7]



Figura 7: Montaje de 6 cámaras PrimeX 41 [9]

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 conoce también como la ciencia de la 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.

Para un sistema de navegación se debe tener un contexto, siendo este el ambiente en el cual dicho sistema opera que puede alterar o no la información de posicionamiento de un objeto o vehículo. Teniendo esto en mente, se sabe que para tener un sistema eficiente, se debe acoplar al contexto dado que pueden haber obstáculos que impidan que señales de radiofrecuencia por ejemplo, puedan penetrar, o bien en ambientes marítimos es útil usar el sonar para saber altura y objetos que puedan ser un potencial peligro, también para la

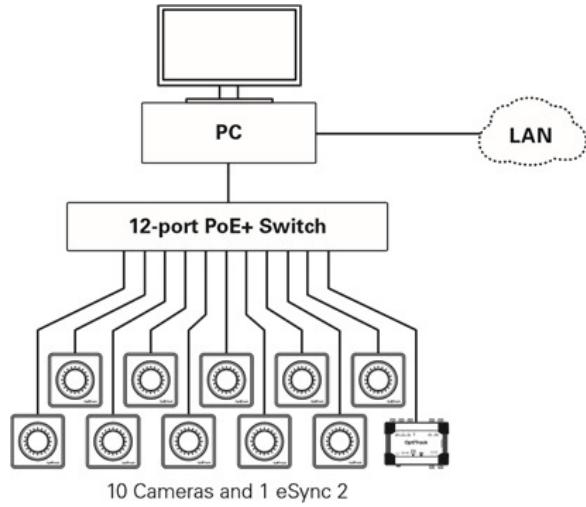


Figura 8: Esquema de conexión cámaras PrimeX de Optitrack [8]

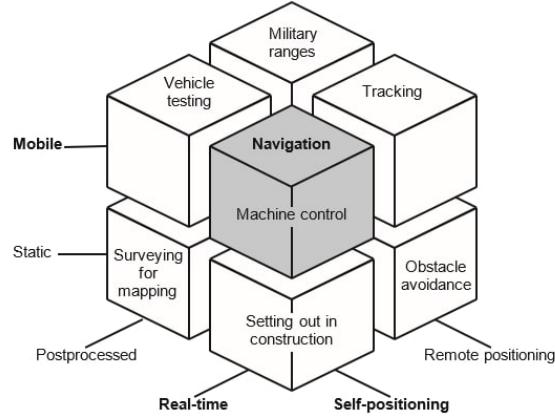


Figura 9: Aplicaciones de sistemas de navegación. [10]

aviación la instrumentación para saber posicionamiento debe ser algo considerado en dado caso un instrumento falle, pueda haber otro que detecte un estimado o lo mismo con la finalidad de tener una noción de la ubicación en el espacio.

Si bien se sabe, existen múltiples métodos de localización, ya sea por GPS, IMUs, UWB, Wi-Fi, sistemas IR, 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.

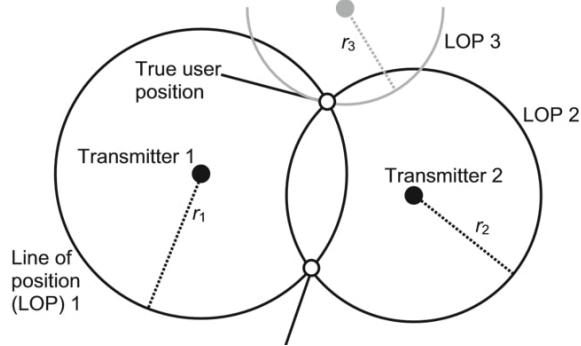


Figura 10: Posicionamiento por tres rangos. [10]

Aplicaciones de posicionamiento en interiores

Según Groves [10], existen categorías de posicionamiento exterior e interior, siendo las de interior la 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.

Filtros 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.

$$\dot{x} = Ax + Bu + Fv \quad (1)$$

$$y = Cx + w \quad (2)$$

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

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

donde:

- $\dot{\hat{x}}$ 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 del 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.

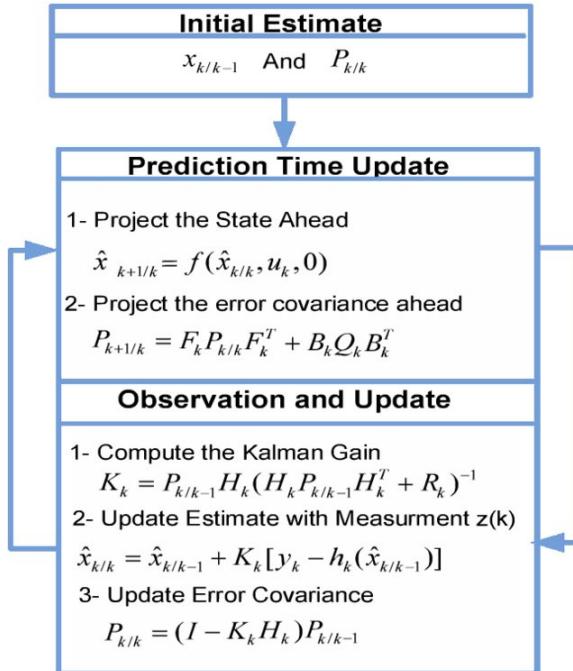


Figura 11: Algoritmo de cálculo EKF. [12]

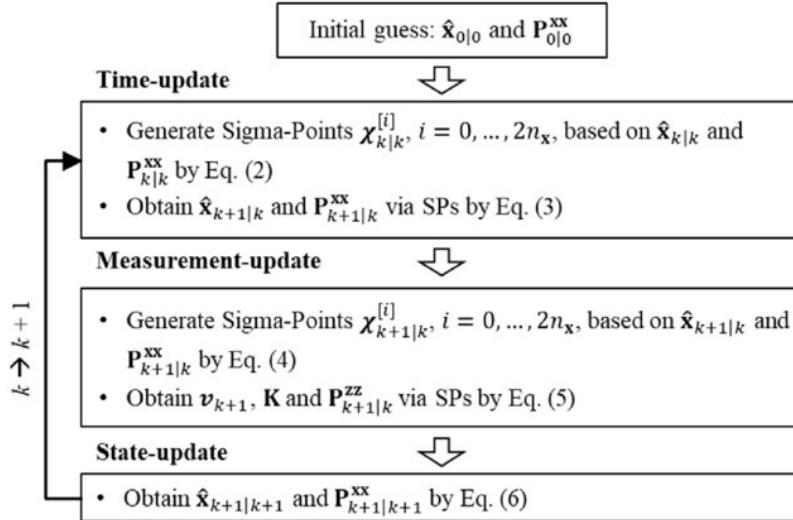


Figura 12: Algoritmo de cálculo UKF. [13]

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 \mathbf{A} con un vector \mathbf{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.

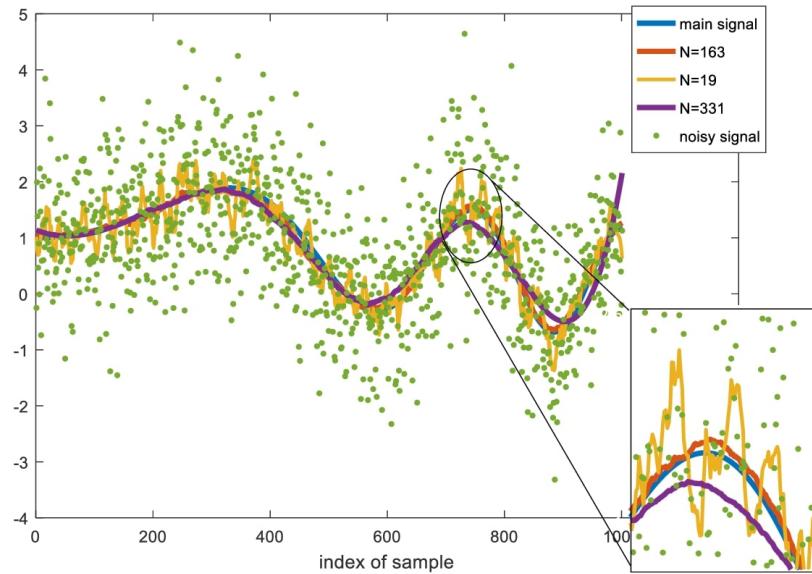


Figura 13: Reconstrucción de señal con filtro SG. [15]

Metodología

Comunicación con celular mediante BLE

Los módulos UWB MDEK1001 cuentan con comunicación BLE para obtener posicionamiento en tiempo real. Dentro de la aplicación DRTLS se puede configurar los módulos para activar o desactivar dicha opción y con ello obtener información en el celular por lo que se conoce como "gateway". Dentro de la aplicación se tiene un registro de mensajes donde se obtienen rangos, calidad de medición de cada rango y otro para la posición en general. Lo que se realiza para obtener estos mensajes es descargar el archivo ".log" que la aplicación proporciona, luego realizar un código de python de parsing para extraer únicamente las coordenadas X, Y. Para llegar a ese punto se debe montar al menos cuatro anclas en un ambiente despejado (sin obstáculos que alteren la línea de visión de cada uno) y una etiqueta que será la que envía los datos al celular.

Una vez montado el escenario se debe obtener los datos en el archivo previamente mencionado, el cual será enviado a la computadora y con el código de python se debe filtrar la información deseada. Con los datos obtenidos en formato ".csv" se puede graficar ya sea en el mismo entorno o en Matlab con los distintos gráficos interactivos para observar el comportamiento de un movimiento para un tiempo determinado.

Comunicación mediante formato TLV

Los módulos MDEK1001 ofrecen la flexibilidad de distintos métodos de comunicación como lo es por medio BLE, UART usando USB o de manera digital mediante pines que se pretenden conectar a un módulo Raspberry Pi 3 que tiene pines GPIO y otros reservados para cada acción. En el caso de la placa del módulo en cuestión, posee pines de interés siendo el módulo UART con Tx y Rx. Para corroborar que si se puede obtener posicionamiento, se debe conectar un módulo a la computadora para probar la interfaz ascii la cual se puede acceder al presionar dos veces enter. Con ello en presente se debe utilizar una aplicación interactiva para comunicación con dispositivos siendo esta "Hercules Setup Utility" con ello configurar el puerto, velocidad en baudios y sin paridad. Ya habilitado el puerto se debe enviar el comando previamente descrito (dos enter teniendo el entorno configurado con CR) y con ello se accede a una interfaz para comunicarse usando comandos ascii. Si se obtienen valores esperados, en el caso del comando para obtener posicionamiento, se denomina *apg* seguido de un enter y se obtienen los mismos valores que proporciona el BLE.

Como bien se sabe el formato ascii envía caracteres mediante UART y al describir cadenas de texto y números el mensaje se extiende, por lo cual existe el formato TLV "Type, Length, Value" que funciona usando comandos en hexadecimal. La ventaja de este formato es lo corto que es y envía los datos en menor tiempo y caracteres comparado contra ascii; sabiendo esto en la guía del firmware API [16] muestra los comandos en hex que se puede enviar y la respuesta que se espera, siendo el tipo, longitud y los valores solicitados. Los datos deben ser convertidos de hex a decimal y modificar el formato little endian para poder realizar la conversión de manera correcta.

Al obtener información TLV mediante USB se debe luego migrar a una comunicación que

no sea directamente con la computadora, sino mas bien con una placa de desarrollo siendo un DOIT Devkit V1 ESP32 cuya ventaja es que posee dos canales UART para el uso de Tx2 y Rx2 con la finalidad de obtener datos mediante dicho canal y mostrarlos en la terminal serial de la IDE de Arduino a fin de corroborar si se reciben los datos esperados. Si los datos se reciben de acuerdo a lo establecido, se debe hacer un shifteo de datos y conversión a decimal para tener X, Y.

Implementación de Punto de Acceso para carga de información de posicionamiento

Dado que la placa de desarrollo del ESP32 seleccionada posee una antena Wi-Fi, se debe agregar al script existente, la creación de un AP de la misma placa para conectar computadoras, celulares mediante una dirección IP dada por la misma. Una vez realizado el setup y el handle de datos en el punto de acceso se debe crear un script HTML que permita visualizar los datos en tiempo real. Se debe implementar un tiempo de captura de datos, siendo como mínimo 100ms para no saturar el servidor y luego al ejecutar el código se obtienen datos tanto en la terminal serial como en el IP del AP creado.

Al obtener los datos de una etiqueta se debe proceder a agregar otro canal, para ello se debe considerar que los pines Tx0 y Rx0 están reservados para comunicación de la computadora con el ESP32, sin embargo se puede solucionar con una librería de Arduino denominada "*SoftwareSerial*" la cual permite la creación de comunicación UART adicional. Con ello se debe repetir los pasos anteriores de conversión y subida de datos así como la modificación del script HTML para visualizar ambos datos de posicionamiento en tiempo real.

Para agregar un sensor de 9DOF siendo el MPU9250 para la aceleración en X, Y, los valores del magnetómetro y yaw para saber orientación. Una vez obtenidos los datos se debe tener los valores X, Y, del magnetómetro, calibrarlos para obtener Yaw a partir de esos valores y el Yaw que proporciona el módulo. Para fines de comparación se igualará la posición inicial a la obtenida con los módulos UWB (como condiciones iniciales) luego al integrar doblemente las aceleraciones ya se podrá obtener una posición que puede compararse con las obtenidas con los módulos UWB para que, posteriormente se implemente el filtro Kalman. Luego estos datos se deben subir al AP para poderlos obtener en plataformas como Matlab.

Integración en PCB

Para poder centralizar los módulos MDEK1001 con el MPU9250 y el ESP32 se debe diseñar un esquemático con las conexiones correspondientes a cada uno, esto luego de comprobar que los datos se obtienen de manera exitosa y así poderlo pasar a placa. Con los esquemáticos hechos se elaboran los tracks y se fabrica con una CNC para PCBs. Ya con la PCB se debe tomar en cuenta generar una cubierta que permita proteger los módulos o bien que pueda ser montada en cualquier parte deseada.

Pruebas estáticas

Las pruebas estáticas consisten en tomar datos de los módulos de manera estacionaria, con la finalidad de tomar la mayor cantidad de muestras y determinar la razón por la cual pueden existir datos atípicos y posible mejora de la precisión de los módulos en dichas pruebas haciendo uso de filtros. Dentro de las pruebas estáticas se debe realizar con una etiqueta e implementando orientación e igual con dos etiquetas. Ambas pruebas deben ser realizadas en comparación con datos obtenidos de las cámaras Optitrack, esto usando Matlab y guardar los datos en un archivo ".csv".

Carro a control remoto para obtención de datos

Un dispositivo útil para poder obtener datos es un robot móvil de ruedas, el cual debe ser controlado por medio de bluetooth (cada motor de forma independiente con un joystick) y con ello montar la placa con los módulos integrados junto con un marker para poder obtener múltiples datos en el entorno *Robotat*.

Pruebas Dinámicas

El propósito de las pruebas dinámicas es poder observar el comportamiento de datos en el tiempo. Así mismo se debe obtener la información, comparada con las cámaras Optitrack que corresponden al GT (Ground Truth). De igual manera que las pruebas estáticas se debe tener posición 2D y orientación pero en tiempo real utilizando Matlab para la obtención de datos. Una forma de tener datos congruentes es elaborar un carro controlado por medio de bluetooth en una aplicación con 2 joysticks, de dicho modo se puede leer datos de posición y orientación y luego graficar estos valores. Para el montaje se necesita diseñar e imprimir sujetadores de los módulos, ya sea para trípode o para los postes de las cámaras PrimeX 41, también para el carro de modo que se pueda montar la PCB con módulos integrados junto con un marker.

Se deben realizar dichas pruebas y una que pueda graficar en tiempo real. Las propuestas de configuración son, utilizando 1 etiqueta, con 6 anclas, luego 2 etiquetas con 8 o 10 anclas y observar los comportamientos en las configuraciones deseadas y posteriormente comparar; esto aplica de igual manera para las pruebas estáticas.

Filtrado de Posicionamiento

Una vez se obtengan los datos de las pruebas dinámicas y estáticas se debe implementar un Filtro Kalman en 2D utilizando como guía los algoritmos de la Figura 11 y la Figura 12 o bien el enfoque clásico de Kalman dado que es en 2D, esto se debe realizar en el entorno Matlab. Luego se debe comparar los resultados del filtro con los GT y finalmente implementarlo en tiempo real para obtener tres gráficos que corresponden a la señal cruda, la filtrada y GT de Optitrack.

Análisis de resultados

Una vez finalizadas las pruebas, se debe organizar los resultados obtenidos por prueba descrito por la configuración y ejecución de la misma, con ello se analiza para las pruebas estáticas y dinámicas. Para las estáticas se debe analizar datos estadísticos como media, desviación estándar, etc. de igual manera con las dinámicas. Los gráficos de interés pueden ser de caja y bigotes, dispersión, continuos y en el tiempo basado en marcas de tiempo.

Cronograma de actividades

No.	Tarea	Fecha inicio	Fecha finalización	Junio	Julio	Agosto	Septiembre				
				03/06/2024 10/06/2024	17/06/2024 24/06/2024	01/07/2024 08/07/2024	15/07/2024 22/07/2024	29/07/2024 05/08/2024	12/08/2024 19/08/2024	26/08/2024 02/09/2024	09/09/2024 16/09/2024
1 Comunicación con celular mediante BLE											
1.1	Comprender el uso de la aplicación DRTLS y extraer datos de ubicación en archivos .log y parsing de datos en python	03/06/2024	04/06/2024								
2 Emplear formato TLV en HEX para obtener datos de localización											
2.1	conectar a computadora para tener datos en tiempo real	04/06/2024	05/06/2024								
2.2	Realizar programa en IDE arduino para comunicar con UART usando el ESP32 y obtener respuestas de posición en HEX	04/06/2024	05/06/2024								
2.3	Transformar datos HEX a decimal	04/06/2024	05/06/2024								
3 Implementar punto de acceso Wi-Fi para subir datos de posicionamiento 2D y orientación en yaw											
3.1	Crear un script HTML para fetch de datos	06/06/2024	06/06/2024								
3.2	Crear IP y subir datos al punto de acceso del ESP32	07/06/2024	07/06/2024								
3.3	Agregar un canal de comunicación UART adicional para 2T	08/06/2024	08/06/2024								
3.4	Agregar comunicación UART por Softwareserial para comunicación ininterrumpida	08/06/2024	08/06/2024								
3.5	Modificar HTML y subir datos al punto de acceso de dos posiciones	08/06/2024	08/06/2024								
3.6	Obtener datos de MPU9250 siendo aceleraciones xy, orientación en yaw y valores del magnetómetro, subirlos al AP e implementarlos al HTML	09/06/2024	10/06/2024								
4 Pruebas estáticas											
4.1	Medición e impresión 3D de sujetadores a postes de Robotat	11/06/2024	13/06/2024								
4.2	Crear un script de MATLAB para obtener datos subidos al AP del ESP32 y datos de Optitrack	14/06/2024	16/06/2024								
4.3	Realizar pruebas estáticas en ambiente LOS (Robotat) con 4A y 1T en una network. Obtener datos con MATLAB en un .csv comparándolo con Optitrack como GT en posición y orientación	17/06/2024	20/06/2024				No hay labs				
4.4	Realizar pruebas estáticas en ambiente LOS (Robotat) con 8A y 2T en una network. Obtener datos con MATLAB en un .csv comparándolo con Optitrack como GT en posición y orientación	21/06/2024	23/06/2024								
4.5	Realizar pruebas estáticas con 8A y 2T con 2 networks separadas comparándolo con Optitrack	01/07/2024	02/07/2024								
5 Pruebas dinámicas											
5.1	Elaborar un carro con control bluetooth para montar sensores y marker	03/07/2024	06/07/2024								
5.2	Montar protoboard en el carro con 2 sensores UWB y un marker de Optitrack para extracción de datos	07/07/2024	08/07/2024								
5.3	Elaborar placa PCB para Integrar ESP32, DWM1001 y MPU9250	09/07/2024	14/07/2024								
5.4	Probar módulo Integrado en PCB en carro para obtener datos de Optitrack con 1T y obtener datos	15/07/2024	16/07/2024								
6 Filtro de posicionamiento											
6.1	Fusionar sensores de magnetómetro, orientación, aceleración para posición y la del UWB	22/07/2024	29/07/2024								
6.2	Implementación de Filtro Kalman no lineal	22/07/2024	29/07/2024								
6.3	Comparación de datos filtrados contra datos reales (Optitrack, AP del ESP32 y filtrados)	30/07/2024	12/08/2024								
6.4	Filtrado de posicionamiento en tiempo real	30/07/2024	12/08/2024								
7 Resultados											
7.1	Ánalisis estadístico de pruebas estáticas y discusión de resultados	19/08/2024	02/09/2024								
7.2	Ánalisis estadístico de pruebas dinámicas y discusión de resultados	19/08/2024	02/09/2024								
7.3	Ánalisis de los filtros utilizados y discusión de resultados	19/08/2024	02/09/2024								
8 Redacción documento de tesis											

Figura 14: Cronograma de actividades

Índice preliminar

- 1. Resumen
- 2. Abstract
- 3. Introducción
- 4. Antecedentes
 - 4.1 Implementación de un sistema de localización en tiempo real en espacios cerrados para el robot Rover UVG
 - 4.2 Uso de tecnología de banda ultra-ancha para control de ubicación de un carro
 - 4.3 Mejora de la precisión de posicionamiento del Módulo MDEK1001 al tener múltiples rangos y en interiores e implementación de EKF
- 5. Justificación
- 6. Objetivos
 - 6.1 Objetivo General
 - 6.2 Objetivos Específicos
- 7. Alcance
- 8. Marco teórico
 - 8.1 Tecnología UWB
 - 8.2 Sistema de captura de movimiento con cámaras Optitrack
 - 8.3 Sistema de Navegación Local
 - 8.4 Aplicaciones de posicionamiento en interiores
 - 8.5 Filtros Kalman
 - 8.6 Filtro de Savitzky-Golay
- 9. Métodos de Comunicación
 - 9.1 DOIT DevKit V1 ESP32
 - 9.2 Comunicación con celular mediante BLE
 - 9.3 Comunicación mediante formato TLV
 - 9.4 Implementación de Punto de Acceso para carga de información de posicionamiento
- 10. Integración en PCB
- 11. Pruebas estáticas
- 12. Carro a control remoto para obtención de datos
- 13. Pruebas Dinámicas

- 14. Análisis de resultados
- 15. Conclusiones
- 16. Recomendaciones
- 17. Referencias
- 18. Anexos
- 19. Glosario

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, Tesis de Licenciatura.
- [2] Qorvo. «MDEK1001 Development Kit.» Accessed: [2024-03-26]. () .
- [3] M. Electronics. «MDEK1001.» (2024), dirección: <https://www.mouser.mx/ProductDetail/Qorvo/MDEK1001?qs=Ti0ZkKH1s2T1RZBi6MtMNg%3D%3D> (visitado 26-03-2024).
- [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.» (2000), dirección: <https://docs.fcc.gov/public/attachments/FCC-00-163A1.pdf> .
- [7] M. Viot, J. Seegars, K. Dwyer et al., *UltraWideband for Dummies*, E. Kuball, ed. Wiley, 2021, ISBN: 978-1-119-80960-9. dirección: <https://www.qorvo.com/design-hub/ebooks/ultra-wideband-for-dummies> (visitado 09-05-2024).
- [8] OptiTrack. «PrimeX 41 Camera Manual.» Accedido el 2024-05-09. (2024), dirección: <https://docs.optitrack.com/hardware/cameras/ethernet-cameras/primex-41>.
- [9] OptiTrack. «Optitrack PrimeX Camera Setup.» Accedido el 2024-05-09. (2024), 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] Qorvo, *DWM1001 FIRMWARE APPLICATION PROGRAMMING INTERFACE (API) GUIDE*, en, 2019. dirección: <https://forum.qorvo.com/uploads/default/original/1X/dcac22d1c0feaf8238f68d11515ad55dbef1b963.pdf>.