
Evaluación de sensores de distancia para aplicaciones de mapeo de entornos con agentes robóticos móviles

Jorge Ricardo Cerón Cheley



UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Evaluación de sensores de distancia para aplicaciones de
mapeo de entornos con agentes robóticos móviles**

Trabajo de graduación presentado por Jorge Ricardo Cerón Cheley para
optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2024

Vo.Bo.:

(f) _____
Ing. Luis Rivera

Tribunal Examinador:

(f) _____
Ing. Luis Rivera

(f) _____
MSc. Carlos Esquit

(f) _____
Ing. Luis Pedro Montenegro

Fecha de aprobación: Guatemala, 5 de diciembre de 2018.

Prefacio

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras vitae eleifend ipsum, ut mattis nunc. Pellentesque ac hendrerit lacus. Cras sollicitudin eget sem nec luctus. Vivamus aliquet lorem id elit venenatis pellentesque. Nam id orci iaculis, rutrum ipsum vel, porttitor magna. Etiam molestie vel elit sed suscipit. Proin dui risus, scelerisque porttitor cursus ac, tempor eget turpis. Aliquam ultricies congue ligula ac ornare. Duis id purus eu ex pharetra feugiat. Vivamus ac orci arcu. Nulla id diam quis erat rhoncus hendrerit. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Sed vulputate, metus vel efficitur fringilla, orci ex ultricies augue, sit amet rhoncus ex purus ut massa. Nam pharetra ipsum consequat est blandit, sed commodo nunc scelerisque. Maecenas ut suscipit libero. Sed vel euismod tellus.

Proin elit tellus, finibus et metus et, vestibulum ullamcorper est. Nulla viverra nisl id libero sodales, a porttitor est congue. Maecenas semper, felis ut rhoncus cursus, leo magna convallis ligula, at vehicula neque quam at ipsum. Integer commodo mattis eros sit amet tristique. Cras eu maximus arcu. Morbi condimentum dignissim enim non hendrerit. Sed molestie erat sit amet porttitor sagittis. Maecenas porttitor tincidunt erat, ac lacinia lacus sodales faucibus. Integer nec laoreet massa. Proin a arcu lorem. Donec at tincidunt arcu, et sodales neque. Morbi rhoncus, ligula porta lobortis faucibus, magna diam aliquet felis, nec ultrices metus turpis et libero. Integer efficitur erat dolor, quis iaculis metus dignissim eu.

Prefacio	III
Lista de figuras	VII
Lista de cuadros	VIII
Resumen	IX
Abstract	X
1. Introducción	1
2. Antecedentes	2
2.1. Generación de trayectorias y mapeo de entornos	2
2.2. <i>Ant Colony Optimization</i> (ACO)	3
2.3. <i>Particle Swarm Optimization</i> (PSO)	4
2.4. Implementación de algoritmos de inteligencia de enjambre en UVG	4
3. Justificación	10
4. Objetivos	11
4.1. Objetivo general	11
4.2. Objetivos específicos	11
5. Alcance	12
6. Marco teórico	13
6.1. Interacción de un robot móvil con su entorno	13
6.2. Modelos probabilístico de movimiento para robots móviles en entornos planos	14
6.3. Robot móvil con tracción diferencial	15
6.4. Localización de robots móviles	16
6.5. Localización y Mapeo Simultáneo (SLAM)	17
6.6. Filtro de Kalman extendido (EKF)	18
6.7. <i>Light Detection and Ranging</i> (LiDAR)	19

6.8. Sensores de distancia por tiempo de vuelo	20
6.9. Vehículo diferencial <i>Pololu 3pi+ 32U4</i>	20
7. Validación de sensor de distancia tipo láser	22
7.1. Sensores de distancia tipo láser considerados	22
7.1.1. Sensor VL53L0X	22
7.1.2. LiDAR FHL-LD20	23
7.1.3. YDLIDAR Tmini Pro	23
7.2. Selección del sensor de distancia tipo láser	24
7.2.1. Criterios para el <i>Trade Study</i> de los sensores de distancia tipo láser	25
7.2.2. Resultados del <i>trade study</i>	30
8. Lectura e interpretación de datos extraídos del LiDAR FHL-LD20	32
8.1. Funcionamiento del sensor de distancia FHL-LD20	32
8.1.1. Características generales	32
8.1.2. Protocolo de comunicación	33
8.2. Lectura de datos	35
8.2.1. Lectura e interpretación de datos mediante ESP32 y Matlab	35
8.2.2. Lectura e interpretación de datos mediante ESP32 y Python	41
9. Caracterización del LiDAR FHL-LD20	44
9.1. Compensación en las mediciones del LiDAR FHL-LD20	44
9.2. Depuración de mediciones nulas en el LiDAR FHL-LD20	46
9.3. Rango de medición efectivo del LiDAR FHL-LD20	48
9.3.1. Rango mínimo de medición	49
9.3.2. Rango máximo de medición	51
9.4. Resolución del LiDAR FHL-LD20	51
9.5. Evaluación angular del LiDAR FHL-LD20	52
9.6. Precisión del LiDAR FHL-LD20	52
9.7. Detección de bordes y discontinuidades para el LiDAR FHL-LD20	55
10. Calibración del LiDAR FHL-LD20	57
10.1. Desarrollo de plataforma para calibrar el LiDAR FHL-LD20	57
10.2. Pruebas para calibrar el LiDAR FHL-LD20	58
11. Conclusiones	60
12. Recomendaciones	61
13. Bibliografía	62
14. Anexos	64
14.1. Pruebas de rango de medición mínimo	65

Lista de figuras

1.	Mapa generado a partir de cuadrícula de probabilidad de colisión con ruta óptima [1].	3
2.	Ruta generada a partir del algoritmo ACO mejorado [3].	4
3.	Desplazamiento del grupo de robots hacia la meta [5].	5
4.	Formación triangular del grupo de robots en la meta [5].	5
5.	Mapa y trayectoria generada para la validación en Webots [7].	6
6.	Secuencia de movimientos del agente móvil para la evasión de obstáculos en Webots [7].	7
7.	Escenario simulado en Webots [8].	8
8.	Puntos de visibilidad de los sensores integrados en el agente robótico móvil en Webots. [8].	8
9.	Espacio de trabajo estimado en exploración de 10 minutos, con planificación de trayectoria hacia su posición inicial de simulación [8].	9
10.	Interacción de un robot con su entorno [9].	14
11.	Sistema de coordenadas de referencia de un robot móvil con tracción diferencial. XOY muestra el marco de referencia mundial. (x_b, y_b) es la coordenada del centro de masa del robot a lo largo de los ejes X_R y Y_R , los cuales representan el sistema local de coordenadas del robot [12].	16
12.	Visualización del proceso de SLAM [9].	18
13.	Funcionamiento LiDAR [15].	19
14.	Pololu 3pi+ 32U4 OLED Robot [17].	21
15.	Resultados del <i>trade study</i>	30
16.	Leyenda del gráfico de barras segmentadas.	31
17.	Orientación del sensor (FALTA REF)	33
18.	Esquema de conexión del conector ZH1.5T-4P del LiDAR FHL-LD20 (FALTA REF)	34
19.	Diagrama de flujo general para captura de datos provenientes del sensor.	36
20.	Diagrama de flujo general para procesamiento de datos provenientes del sensor.	38
21.	Ejemplificación del proceso de reconstrucción de mediciones: ángulo de inicio.	39
22.	Visualización de la primera corrida del sensor: 840 puntos de medición.	40

23.	Vista de usuario, interfaz gráfica LiDAR MapViewer.	42
24.	Comparación de la reconstrucción con y sin ángulo de corrección: tres revoluciones completas capturadas.	46
25.	Diagrama de flujo general para procesamiento de datos en formato decimal.	47
26.	Comparación de la reconstrucción con y sin filtro de mediciones nulas: tres revoluciones completas capturadas.	48
27.	Ejemplificación de las lecturas nulas durante la reconstrucción de mediciones: entorno circular de 6 cm de radio.	49
28.	Reconstrucciones vacías de tres revoluciones completas capturadas en entorno circular: 6 cm de radio	50
29.	Reconstrucciones vacías de tres revoluciones completas capturadas en entorno circular: 10 cm de radio	51
30.	Ánálisis de la reconstrucción realizada para las aristas horizontales de una caja de 400×400 mm: tres revoluciones completas capturadas.	54
31.	Ánálisis de la reconstrucción realizada para las aristas verticales de una caja de 400×400 mm: tres revoluciones completas capturadas.	55
32.	Diseño de los codos para transiciones graduales	56
33.	Vista superior de la plataforma para calibración.	58
34.	Vista isométrica del soporte para el LiDAR FHL-LD20.	58
35.	Reconstrucción realizada para verificar la calibración del sensor.	59
36.	Reconstrucciones vacías de tres revoluciones completas de lectura en entorno circular: 7 cm de radio	65
37.	Reconstrucciones vacías de tres revoluciones completas de lectura en entorno circular: 8 cm de radio	66
38.	Reconstrucciones vacías de tres revoluciones completas de lectura en entorno circular: 9 cm de radio	67

Lista de cuadros

1.	Criterios de comparación.	24
2.	Matriz de comparación por pares.	24
3.	Escala fundamental de comparación por pares.	25
4.	Valores de interés para la comparación.	25
5.	Ponderación de costo.	26
6.	Ponderación de disponibilidad.	26
7.	Ponderación de precisión.	26
8.	Ponderación de rango de medición.	27
9.	Ponderación de parámetros eléctricos.	27
10.	Ponderación de velocidad de medición.	28
11.	Ponderación de tamaño.	28
12.	Ponderación de peso.	28
13.	Ponderación de operatividad en aplicaciones deseadas.	29
14.	Ponderación de adaptabilidad a robots móviles.	30
15.	Características de los sensores analizados.	30
16.	Ponderación individual de los sensores evaluados según criterios establecidos.	31
17.	Parámetros de rendimiento.	33
18.	Parámetros de conexión del Lidar FHL-LD20.	34
19.	Parámetros de transmisión.	34
20.	Estructura de un paquete de medición.	34

Resumen

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras vitae eleifend ipsum, ut mattis nunc. Pellentesque ac hendrerit lacus. Cras sollicitudin eget sem nec luctus. Vivamus aliquet lorem id elit venenatis pellentesque. Nam id orci iaculis, rutrum ipsum vel, porttitor magna. Etiam molestie vel elit sed suscipit. Proin dui risus, scelerisque porttitor cursus ac, tempor eget turpis. Aliquam ultricies congue ligula ac ornare. Duis id purus eu ex pharetra feugiat. Vivamus ac orci arcu. Nulla id diam quis erat rhoncus hendrerit. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Sed vulputate, metus vel efficitur fringilla, orci ex ultricies augue, sit amet rhoncus ex purus ut massa. Nam pharetra ipsum consequat est blandit, sed commodo nunc scelerisque. Maecenas ut suscipit libero. Sed vel euismod tellus.

Proin elit tellus, finibus et metus et, vestibulum ullamcorper est. Nulla viverra nisl id libero sodales, a porttitor est congue. Maecenas semper, felis ut rhoncus cursus, leo magna convallis ligula, at vehicula neque quam at ipsum. Integer commodo mattis eros sit amet tristique. Cras eu maximus arcu. Morbi condimentum dignissim enim non hendrerit. Sed molestie erat sit amet porttitor sagittis. Maecenas porttitor tincidunt erat, ac lacinia lacus sodales faucibus. Integer nec laoreet massa. Proin a arcu lorem. Donec at tincidunt arcu, et sodales neque. Morbi rhoncus, ligula porta lobortis faucibus, magna diam aliquet felis, nec ultrices metus turpis et libero. Integer efficitur erat dolor, quis iaculis metus dignissim eu.

Abstract

This is an abstract of the study developed under the

CAPÍTULO 1

Introducción

El objetivo principal de este trabajo fue evaluar tecnologías y opciones de sensores de distancia que puedan usarse para mapeo de entornos. Para ello, se llevó a cabo un análisis de las opciones disponibles en el mercado, considerando distintos criterios, y se seleccionó la mejor opción que se consideraba adecuada para su futura implementación con agentes robóticos móviles. Asimismo, se desarrolló una herramienta de software que permitió la comunicación con el sensor seleccionado, permitiendo adquirir, interpretar y visualizar los datos capturados de manera intuitiva y eficiente.

Además, con esta herramienta, se caracterizó y calibró el LiDAR FHL-LD20, con el objetivo de optimizar su rendimiento y validar su precisión. Se llevaron a cabo pruebas en distintos entornos controlados para evaluar su funcionamiento y la fidelidad de sus mediciones bajo diferentes condiciones. Durante este proceso, se establecieron criterios de comparación y evaluación para analizar el desempeño del mismo. De esta manera, se identificaron posibles limitaciones y áreas de mejora, con el fin de optimizar su uso y garantizar fiabilidad de los datos obtenidos para su futura implementación en aplicaciones de mapeo y navegación en el ecosistema Robotat de la Universidad del Valle de Guatemala.

CAPÍTULO 2

Antecedentes

En el campo de la robótica móvil, la navegación autónoma desempeña una tarea fundamental para que los sistemas robóticos se desplacen de manera segura y eficiente en entornos dinámicos y desconocidos. Este proceso implica la percepción del entorno mediante sensores, la planificación y ejecución de trayectorias y la toma de decisiones en tiempo real, sin la necesidad de intervención humana directa. Por lo tanto, la exitosa integración de este proceso resulta en una mayor capacidad de adaptación y desempeño de los robots móviles con su entorno, promoviendo la autonomía e independencia de estos sistemas.

2.1. Generación de trayectorias y mapeo de entornos

En [1] se describió un método de construcción para mapeo de entornos basado en un mapa de cuadrícula de probabilidad de colisión y la mejora de un algoritmo A*. Se utilizaron técnicas de probabilidad de colisión que consideraron tanto el tamaño del robot como la posición relativa de este y los obstáculos presentes en la cuadrícula para construir un mapa del entorno. El método permitió encontrar rutas seguras y eficientes al integrar información de probabilidad de colisión en el proceso de construcción de mapas y planificación de rutas desde el nodo de inicio hasta el nodo objetivo. El mapa generado ofrece una cuadrícula en escala de grises indicando la probabilidad de colisión según la cercanía de los obstáculos; cuanto más oscuro el color, más peligroso y cercano se encuentra el obstáculo. En cambio, la cuadrícula blanca representa la zona libre y menos propensa a la colisión.

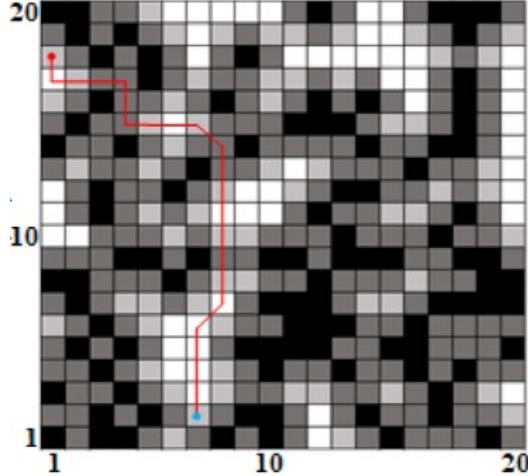


Figura 1: Mapa generado a partir de cuadrícula de probabilidad de colisión con ruta óptima [1].

Además, considera la función de costo del algoritmo A*, minimizando el tiempo de búsqueda de nodos peligrosos en entornos complejos, generando un mapa más completo y reduciendo el tiempo de búsqueda y número de giros en las rutas propuestas. Los resultados de las simulaciones realizadas en Matlab demuestran mejoras en la seguridad ante colisiones y eficiencia de localización de objetos en entornos complejos sobre algoritmos tradicionales. Sin embargo, los autores identifican ciertas limitaciones ante obstáculos dinámicos al momento de mapear el entorno.

En [2] se desarrolló e implementó un algoritmo para el mapeo bidimensional en robots cuadrúpedos utilizando estimaciones de mínimos cuadrados. El algoritmo generado tuvo como finalidad planificar y producir trayectorias de seguimiento que permitieron la evasión de obstáculos en entornos cambiantes. Esto se logró por medio de un radar láser 3D y un sistema de posicionamiento de banda ultraancha (UWB) los cuales proporcionaban datos espaciales y direccionales del entorno en tiempo real. Dicha información fue transformada en un mapa de rejilla 2D donde se estableció la ubicación actual del cuadrúpedo, la presencia de obstáculos y los espacios libres de movimiento. El artículo presentó resultados experimentales que demostraron la precisión del algoritmo en la construcción del entorno y la autonomía del cuadrúpedo al ajustar sus movimientos y trayectorias en tiempo real según diferentes situaciones. Además, se destacó la futura implementación de algoritmos de trayectorias globales y locales para mejorar la reconstrucción del entorno y la estabilidad del sistema de planificación de rutas y navegación espacial.

2.2. *Ant Colony Optimization (ACO)*

En [3] se compararon algoritmos basados en ACO para abordar la problemática de planificación de rutas para robots móviles en entornos complejos. En el artículo se analizaron diferentes parámetros para determinar tanto la convergencia de las trayectorias como la optimización y suavidad de las rutas en diferentes entornos de simulación para cada uno de

los algoritmos propuestos. Además, se destacó la implementación de mecanismos de reacción y supresión de curvas para mejorar el rendimiento y adaptabilidad de estos, resultando en un algoritmo que superó en eficiencia y convergencia las versiones tradicionales de ACO. Los resultados de simulación en MATLAB demostraron que el algoritmo mejorado garantiza que los robots puedan encontrar una trayectoria satisfactoria incluso en situaciones desafiantes.

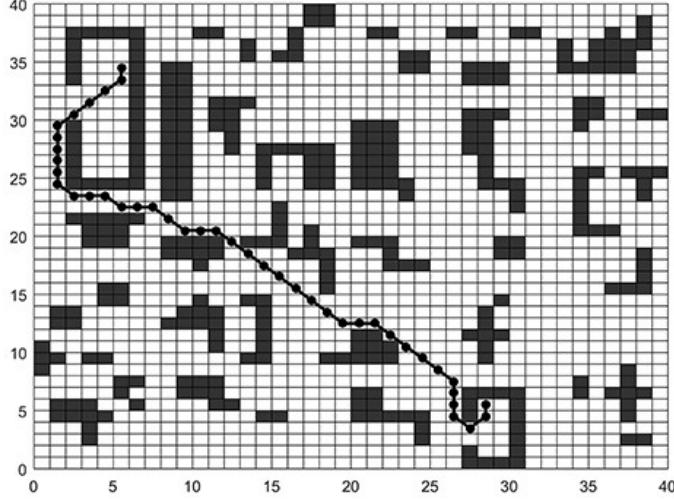


Figura 2: Ruta generada a partir del algoritmo ACO mejorado [3].

2.3. *Particle Swarm Optimization (PSO)*

En [4] se presentó la implementación de PSO para la planificación de rutas de robots móviles en coordenadas polares. Se empleó el algoritmo de optimización por enjambre de partículas como un proceso de búsqueda donde cada agente se movió a través de un espacio bidimensional buscando la mejor ruta desde el punto de inicio hasta el punto objetivo. Durante la ejecución del algoritmo, los agentes se movieron a través del espacio de búsqueda, ajustando su posición y velocidad en función de la información obtenida de su propio desempeño pasado y de la mejor solución global encontrada por otros agentes en el enjambre. Este proceso de búsqueda colectiva permitió que los agentes exploraran el espacio y convergieran hacia soluciones óptimas. Los resultados de simulación demostraron que el algoritmo PSO eliminó puntos de ruta redundantes, optimizando la eficiencia de trayectorias planificadas. Sin embargo, los autores identificaron ciertas limitaciones de escalabilidad en entornos con obstáculos dinámicos.

2.4. Implementación de algoritmos de inteligencia de enjambre en UVG

En el trabajo de investigación [5] se desarrolló e implementó un algoritmo de control para sistemas de robots multi-agente orientado a misiones de búsqueda, basado en métodos teóricos de grafos y control moderno. Se destacó que la integración de una sola función

racional que combina el control de formación y colisiones permitía que los agentes alcanzaran formaciones con un mínimo error cuadrático medio al utilizar grafos totalmente rígidos. Además, los resultados de la simulación en el entorno de Webots demostraron que los agentes lograron alcanzar la meta en el 80 % de los escenarios considerados, con un 11 % de éxito en control de formación. Esto indicó que el algoritmo propuesto fue efectivo en la coordinación de los agentes, más no fue eficiente en la cantidad de formaciones finales exitosas.

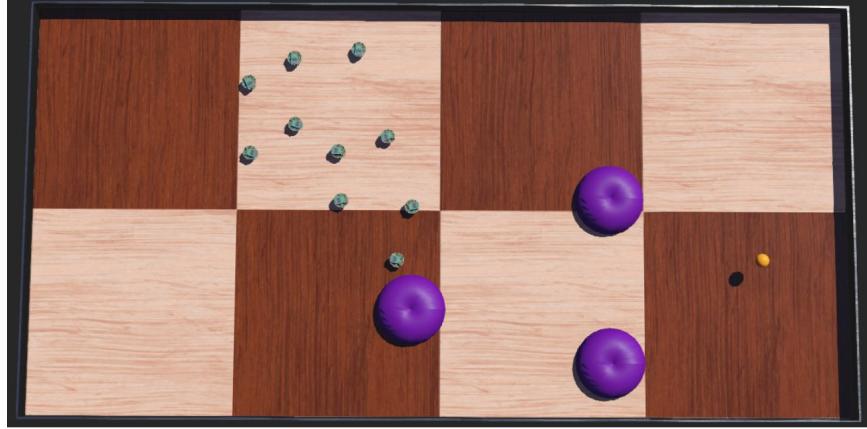


Figura 3: Desplazamiento del grupo de robots hacia la meta [5].

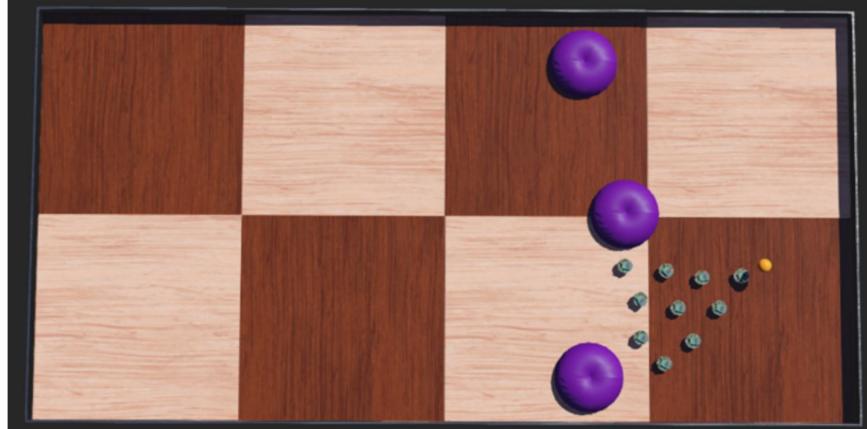


Figura 4: Formación triangular del grupo de robots en la meta [5].

En [6] se desarrolló un algoritmo que determinaba los parámetros más adecuados para mejorar y potenciar el desempeño del algoritmo PSO estándar. Mediante el uso de redes neuronales recurrentes, la herramienta nombrada Deep PSO Tuner, permitió la selección dinámica y automática de los hiper-parámetros que debería emplear el algoritmo, mejorando la precisión y tiempo de convergencia del mismo. Los resultados de las simulaciones realizadas demostraron que la red neuronal BiLSTM fue la más efectiva, logrando superar obstáculos locales y permitiendo una mayor precisión y rendimiento del algoritmo en diferentes situaciones. No obstante, se mencionaron ciertas limitaciones respecto a la escalabilidad del método y la baja reacción del algoritmo en situaciones con obstáculos dinámicos.

En el trabajo de investigación [7] se propusieron dos algoritmos basados en ACO para

resolver problemas de exploración de terrenos, planificación de trayectorias y evasión de obstáculos. Los algoritmos fueron implementados en Matlab y se realizaron validaciones a nivel de simulación en el entorno de Webots de Cybertbotics. Los resultados de este trabajo evidenciaron que las trayectorias generadas por los algoritmos condujeron a que los agentes móviles evadieran exitosamente los obstáculos presentes en el entorno. Asimismo, las modificaciones realizadas al algoritmo, resultaron en mejoras significativas en la autonomía y rendimiento de los agentes y en la fidelidad de las replicas de los entornos explorados.

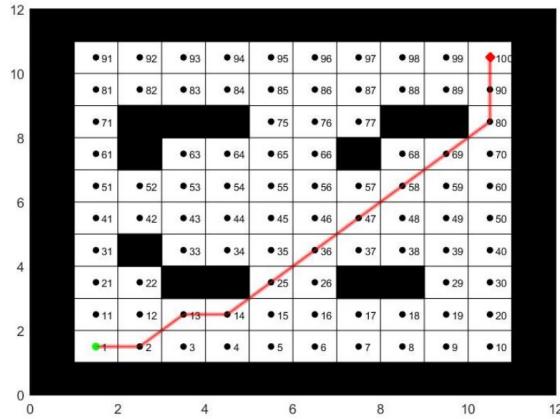


Figura 5: Mapa y trayectoria generada para la validación en Webots [7].

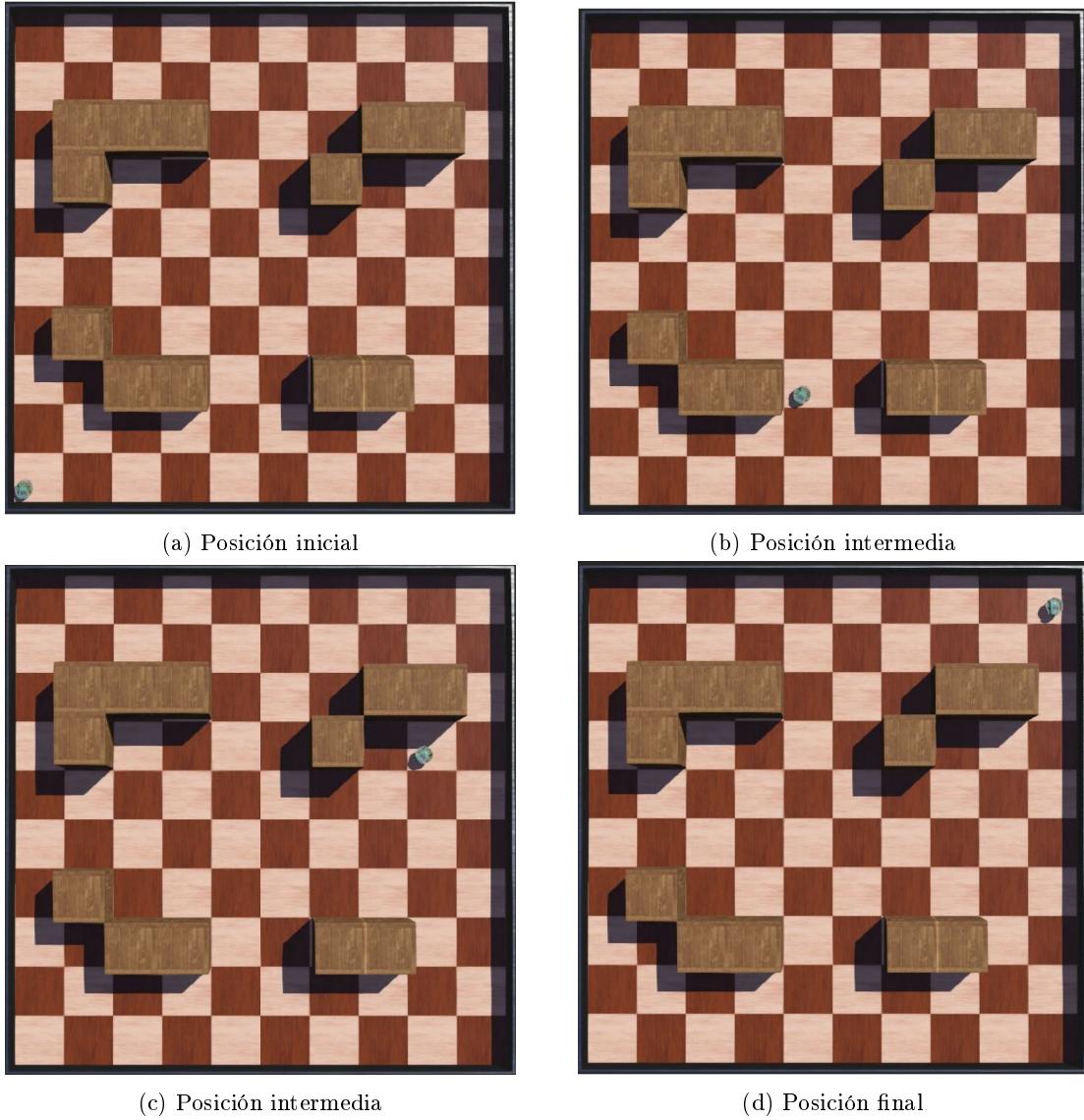


Figura 6: Secuencia de movimientos del agente móvil para la evasión de obstáculos en Webots [7].

En [8] se desarrollaron y evaluaron algoritmos para el mapeo de entornos y generación de trayectorias utilizando sistemas robóticos multi-agente. Se enfocó en la implementación simulada de la combinación de tres algoritmos relacionados con la navegación autónoma en robots con tracción diferencial. Estos algoritmos abarcaron la exploración de entornos, el mapeo bidimensional de los mismos y la generación óptima de trayectorias. Para llevar a cabo la validación, se desarrollaron y simularon en Webots distintos escenarios de trabajo que incluían pasillos, espacios abiertos y obstáculos tridimensionales. Estos entornos fueron reconstruidos, de manera individual y colectiva, por agentes robóticos móviles con sensores de distancia integrados: cinco sensores frontales separados por un ángulo de 45 grados entre sí y un único sensor colocado en la parte trasera del vehículo.

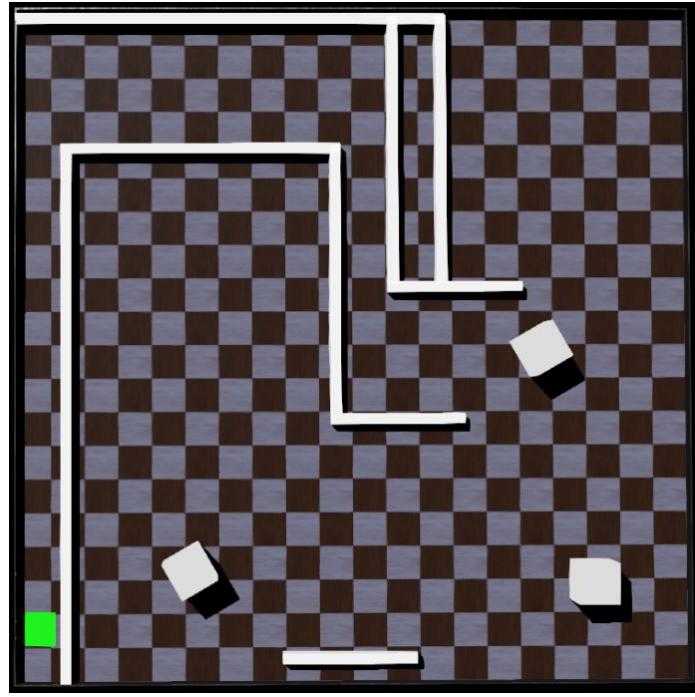


Figura 7: Escenario simulado en Webots [8].

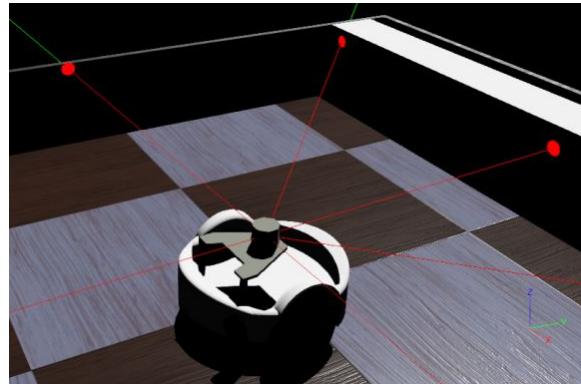


Figura 8: Puntos de visibilidad de los sensores integrados en el agente robótico móvil en Webots. [8].

En las simulaciones computarizadas, se observó que la integración de sensores a distancia tipo láser en lugar de los tipo sonar en los agentes resultaba más precisa y eficaz en la detección de obstáculos, asegurando trayectorias de exploración confiables y no redundantes. Además, se destacó que la disposición e interacción entre estos sensores influía en la capacidad del sistema para obtener una representación precisa del entorno. Se decidió combinar los datos provenientes de todos los sensores delanteros para compensar las limitaciones individuales de cada sensor, obteniendo una buena estimación y reconstrucción del entorno. La precisión en la distribución y disposición de los obstáculos en las estimaciones del espacio de trabajo permitieron el desarrollo y validación del algoritmo de planificación de trayectorias en mapas previamente explorados. Basándose en las validaciones simuladas de los algoritmos desarrollados, el autor recomienda dos sensores de distancia comerciales

para la implementación física de estos algoritmos en futuros estudios. No obstante, se mencionaron ciertas limitaciones respecto a la representación exacta del entorno debido a la falta de información en ciertas áreas por el tiempo de simulación y colisiones recurrentes por obstáculos imprevistos, como la esquina de un obstáculo.

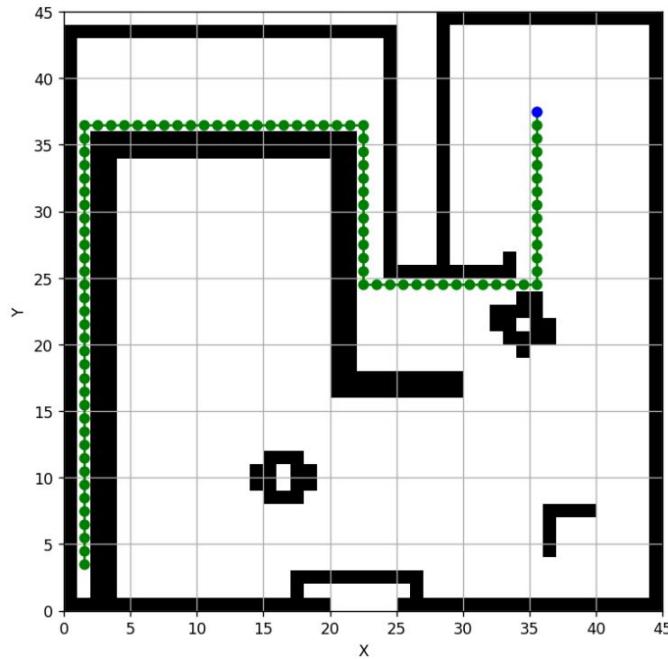


Figura 9: Espacio de trabajo estimado en exploración de 10 minutos, con planificación de trayectoria hacia su posición inicial de simulación [8].

CAPÍTULO 3

Justificación

En entornos desconocidos, los robots móviles deben ser capaces de desplazarse de un punto a otro de manera eficiente y segura tomando en cuenta los obstáculos y elementos de interés dentro del área de operación. En los trabajos de fases anteriores se han desarrollado e implementado, a nivel de simulación, algoritmos de exploración para el mapeo de entornos utilizando modelos virtuales de robots con tracción diferencial. El uso de estos algoritmos junto con las estimaciones del entorno han permitido que se evalúe el rendimiento de algoritmos para la planificación de trayectorias en espacios donde se deseé llevar a cabo una navegación autónoma. Sin embargo, todos los avances que se han realizado han sido a nivel de simulación en software como Matlab y Webots, por lo que no se ha tenido una validación física de los algoritmos en robots móviles funcionales.

Por esta razón, el enfoque principal de este trabajo es abordar, evaluar y seleccionar opciones de sensores de distancia que permitan la futura validación física de los algoritmos de exploración y mapeo de entornos en agentes robóticos móviles. Se busca verificar que, con los datos capturados del sensor, las estimaciones del espacio de trabajo sean consistentes con las mediciones del espacio en sí. Además, se proponen buscar posibles formas de implementación para integrar los sensores de distancia a los robots diferenciales Pololu 3Pi+, lo cual dotaría a los agentes con capacidades adicionales. Esto establecería un precedente con buena base para futuras implementaciones en esta línea de investigación, como lo puede ser el mapeo de entornos físicos dinámicos.

El mapeo de entornos utilizando robots móviles resulta fundamental en una amplia gama de aplicaciones, desde la logística en el transporte de suministros en ambientes comerciales hasta la exploración de terrenos desconocidos en operaciones de búsqueda y rescate. La capacidad de generar mapas detallados que representen con precisión obstáculos y elementos relevantes, no solo mejora la compresión del entorno sino que también establece una base sólida para la toma de decisiones y planificación de acciones en situaciones críticas, como desastres naturales. Estos mapas permiten identificar rutas seguras y evaluar riesgos en las operaciones, lo que mitiga los impactos negativos en situaciones de emergencia para los humanos.

CAPÍTULO 4

Objetivos

4.1. Objetivo general

Evaluar sensores de distancia tipo láser para aplicaciones de mapeo de entornos con agentes robóticos móviles.

4.2. Objetivos específicos

- Seleccionar un sensor de distancia tipo láser que sea factible para investigaciones con robots móviles en el ecosistema Robotat.
- Desarrollar una herramienta de software que permita la obtención e interpretación de los datos provenientes del sensor seleccionado.
- Realizar una caracterización detallada del sensor empleado.
- Desarrollar métodos de prueba y definir criterios de evaluación para la calibración del sensor.
- Evaluar la futura integración del sensor en agentes robóticos móviles disponibles en la Universidad.

CAPÍTULO 5

Alcance

El alcance de este proyecto incluyó el desarrollo una aplicación capaz de capturar, procesar y visualizar las mediciones proporcionadas por el sensor LiDAR FHL-LD20, así como la realización de pruebas físicas para su caracterización y calibración. Estas tareas se llevaron a cabo utilizando herramientas de software como Python y MATLAB. Además, se generaron reconstrucciones precisas basadas en los datos obtenidos del sensor, las cuales permitieron evaluar su rendimiento y ajustar sus parámetros de medición.

De cara a proyectos futuros o trabajos relacionados con el mapeo de entornos y la navegación autónoma, la información recolectada por el sensor podrá integrarse a los algoritmos de navegación de robots móviles. Esto abrirá la posibilidad de implementar funcionalidades avanzadas, como la planificación de trayectorias para navegación punto a punto, optimizando así el desempeño de robots en aplicaciones como la exploración, la detección de obstáculos y el control autónomo en diversos entornos.

CAPÍTULO 6

Marco teórico

6.1. Interacción de un robot móvil con su entorno

El entorno o mundo en el que opera un robot se define por su estado, que engloba las condiciones y características tanto del robot mismo como del entorno circundante. Sin embargo, en la práctica, especificar un estado completo es prácticamente imposible para cualquier sistema robótico realista. Esto se debe a que un estado completo abarcaría no solo todos los aspectos del entorno que podrían tener un impacto en el futuro, sino también las propias características y estados internos del propio robot. Por lo tanto, en las implementaciones prácticas se selecciona un pequeño subconjunto de todas las variables de estado para formar un estado representativo y manejable. Esta selección incluye variables como [9]:

- La pose del robot: Comprende su ubicación y orientación con respecto a un marco de coordenadas global. Para un robot móvil en un entorno plano, la pose suele estar constituida por sus dos coordenadas de ubicación en el plano y su orientación de giro (*yaw*) [9].
- La velocidad del robot: En un entorno plano, las velocidades críticas para un robot móvil de dos ruedas son la velocidad lineal y la angular. La velocidad lineal indica cuán rápido avanza el robot en línea recta a lo largo de su trayectoria, mientras que la velocidad angular determina la rapidez con la que el robot gira alrededor de su centro de giro. Esta última se regula ajustando las velocidades angulares de las ruedas izquierda y derecha [9].
- La ubicación y características de los objetos circundantes en el entorno: Este aspecto abarca características como la apariencia visual de objetos, su forma, tamaño, disposición espacial y cualquier otra propiedad relevante para la interacción del robot con su entorno. Estas variables son cruciales para interpretar y comprender la situación del robot en distintos escenarios [9].

Además de lo anterior, se distinguen dos tipos fundamentales de interacciones entre un

robot y su entorno: la influencia del robot en el estado mediante acciones de control de sus actuadores y la recopilación de información del estado a través de los sensores del propio robot. Las acciones de control, como el movimiento del robot y la manipulación de objetos, provocan cambios instantáneos en el estado del entorno lo que demanda una continua adaptación y respuesta por parte del sistema robótico para actualizar las condiciones del mismo. Por otro lado, se utilizan los sensores para capturar información sobre el estado del entorno. Sin embargo, es común que las mediciones de los sensores tengan cierto retraso, por lo que proporcionan una percepción del estado en momentos relativamente anteriores [9].

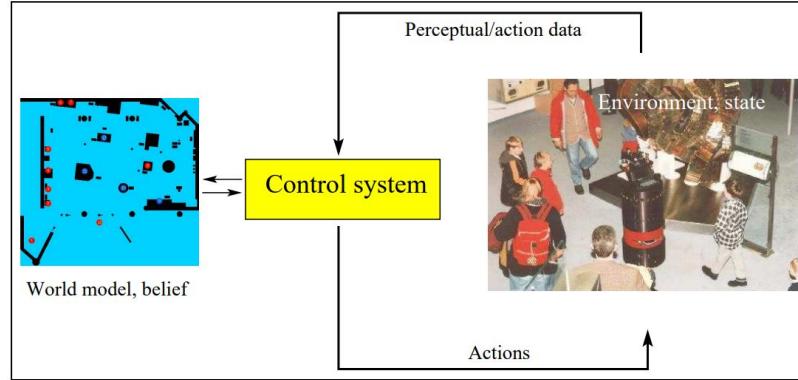


Figura 10: Interacción de un robot con su entorno [9].

6.2. Modelos probabilístico de movimiento para robots móviles en entornos planos

Los modelos probabilísticos de movimiento son empleados en robótica móvil para estimar la posición futura de un robot en función de su movimiento y de las incertidumbres asociadas que puedan influir en su trayectoria como la imprecisión de los sensores, la presencia de fuerzas externas, el ruido, entre otros. Un enfoque comúnmente utilizado es el modelo de movimiento de velocidad, el cual se basa en la relación entre las velocidades lineales y angulares del robot y su ubicación actual en el espacio para predecir su posición futura. Esta metodología en particular ofrece ventajas significativas como su facilidad de implementación y simplicidad para predecir, de manera directa, el movimiento futuro del robot en entornos simples y controlados [9].

Por otro lado, se pueden emplear las mediciones de odometría como base para calcular el movimiento del robot a lo largo del tiempo. La odometría es una técnica utilizada en navegación para estimar la posición y desplazamiento de un vehículo móvil por medio de la medición de las distancias recorridas por sus ruedas o sistemas de locomoción. Este enfoque, conocido como modelo de odometría, integra la información de los encoders en las ruedas para medir la rotación de las mismas y calcular la distancia recorrida. Estas mediciones combinadas con otros datos de orientación son empleados para estimar la posición actual del robot. Sin embargo, este método sufre de errores acumulativos a medida que el robot se desplaza debido a pequeñas desviaciones en la rotación de las ruedas [9].

Alternativamente, existe un modelo más complejo, que considera la cinemática del robot

con la geometría del mismo. El modelo de movimiento cinemático considera la velocidad lineal y angular del robot, junto con la forma y tamaño del mismo para predecir con precisión el desplazamiento y la orientación futura del robot. Este enfoque destaca por su capacidad para capturar el comportamiento del robot en el espacio, ofreciendo una representación más completa del movimiento. Además, su consideración en la geometría proporciona un análisis más profundo en cómo el diseño del robot influye en su movimiento [10].

6.3. Robot móvil con tracción diferencial

Los robots móviles con ruedas (WMRs, por sus siglas en inglés) están compuestos por uno o más cuerpos rígidos equipados con un sistema de locomoción. En términos generales, constan de un cuerpo rígido que funciona como base o chasis, junto con un conjunto de ruedas que les permite moverse sobre el suelo. En un robot móvil de tracción diferencial existen dos ruedas fijas con un eje de rotación común y orientación constante, las cuales se controlan de manera independiente para imponer diferentes valores de velocidad angular y controlar el movimiento lineal del propio robot. Además, estos robots pueden contar con una o más ruedas giratorias de apoyo no motorizadas para proporcionar estabilidad adicional. Es importante destacar que un robot de este tipo puede realizar giros en su propio eje sin necesidad de desplazarse, siempre que las velocidades angulares de las dos ruedas sean iguales y opuestas [11].

En robótica móvil con sistemas de ruedas, se emplea la cinemática diferencial para establecer la relación entre las velocidades de las ruedas del robot y su movimiento en el espacio cartesiano. De esta manera, así como cada rueda contribuye al movimiento general del robot, estas imponen restricciones específicas a la velocidad del mismo. En el caso particular de un robot móvil con tracción diferencial y ruedas fijas, operando en un entorno plano, cada rueda se desplaza únicamente en el plano que la contiene, lo que implica la ausencia de deslizamiento relativo entre la rueda y la superficie, y garantiza que no se produzcan deslizamientos laterales. Estas restricciones, conocidas como restricción de rodadura y restricción de deslizamiento, son fundamentales para definir el modelo cinemático del robot como [12]:

$$x_b \cos \theta + y_b \sin \theta = \frac{D_L \dot{\theta}_L + D_R \dot{\theta}_R}{4} \quad (1)$$

$$x_b \sin \theta + y_b \cos \theta = 0 \quad (2)$$

$$2l\dot{\theta} = D_R \dot{\theta}_R - D_L \dot{\theta}_L \quad (3)$$

donde (x_b, y_b) es la coordenada del centro de masa del robot, \dot{x}_b e \dot{y}_b son las velocidades del centro de masa a lo largo de los ejes X_R y Y_R , correspondiente al sistema local de referencia del robot, donde el eje longitudinal X_R apunta hacia adelante en la dirección del movimiento del robot y Y_R es perpendicular al eje longitudinal apuntando hacia la izquierda del robot. θ representa la orientación del robot respecto de su posición inicial en el punto de inicio del movimiento y $\dot{\theta}$ corresponde a la velocidad angular en el sistema del robot. D_R y D_L son los diámetros nominales de las ruedas derecha e izquierda. Las velocidades angulares de cada rueda se denotan por $\dot{\theta}_R$ y $\dot{\theta}_L$. La distancia nominal entre las dos ruedas se representa por l [12].

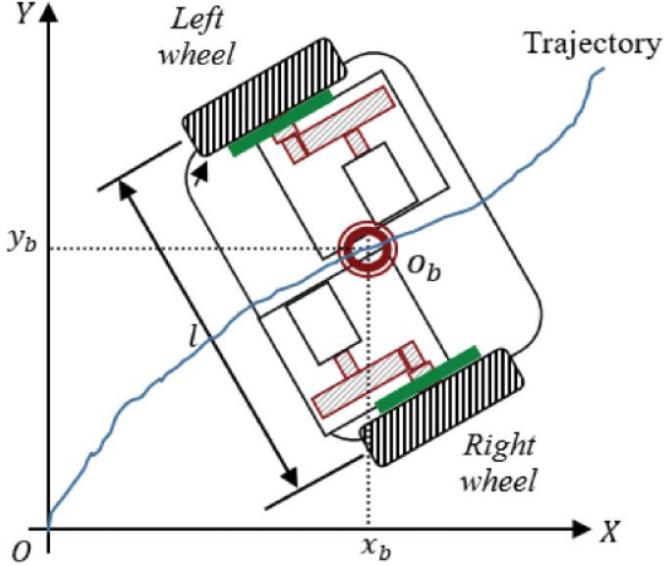


Figura 11: Sistema de coordenadas de referencia de un robot móvil con tracción diferencial. XOY muestra el marco de referencia mundial. (x_b, y_b) es la coordenada del centro de masa del robot a lo largo de los ejes X_R y Y_R , los cuales representan el sistema local de coordenadas del robot [12].

6.4. Localización de robots móviles

En términos de percepción del entorno, la localización de robots móviles constituye uno de los desafíos fundamentales en la robótica. Este desafío se enfoca en determinar la pose de un robot con respecto a un mapa predefinido del entorno en el que opera. Esta información es crucial para permitir que los robots móviles ejecuten tareas autónomas de manera efectiva, tales como la navegación, manipulación de objetos o inspección de entornos [13].

En sí, la localización se refiere al proceso de establecer una correspondencia entre el sistema de coordenadas del mapa mundial con las coordenadas locales del sistema del robot. Conocer esta transformación permite que el robot pueda ubicar objetos de interés en su propio marco de referencia, facilitando su navegación. Sin embargo, la pose del robot no se puede detectar directamente, por lo que se emplea una combinación de diferentes sensores para llevar a cabo la localización de robot como cámaras, LiDARs, sistemas de posicionamiento global, entre otros [13].

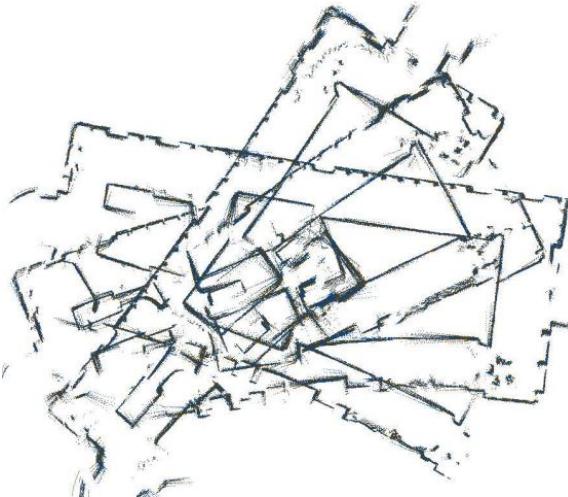
Los problemas de localización varían en dificultad según la naturaleza del entorno y el conocimiento inicial sobre las características del robot. Se distinguen tres tipos de problemas de localización según el conocimiento de la pose del robot: localización local, cuando se conoce la pose inicial; localización global, cuando la pose inicial es desconocida; y el desafío de los “robots secuestrados”, donde el robot es teletransportado durante su operación. Otro factor que tiene un impacto en la dificultad de localización es si el entorno es dinámico o estático, así como si el algoritmo de localización controla o no el movimiento del robot. Además, la cantidad de robots involucrados también puede representar un desafío adicional, ya que un mayor número de robots aumenta la complejidad del proceso de localización [9].

6.5. Localización y Mapeo Simultáneo (SLAM)

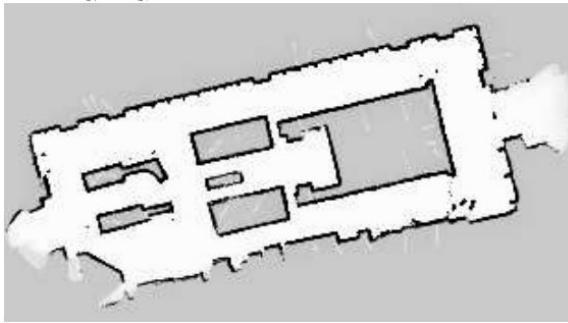
Un desafío central en la robótica móvil es la construcción de un mapa del entorno desconocido mientras simultáneamente se localiza la propia posición del robot dentro de ese mapa. A esta problemática se le denomina localización y mapeo simultáneo o SLAM, por sus siglas en inglés, y a menudo se considera un problema de la clase “huevo y la gallina” debido a su estrecha interrelación. Esto pues, la precisión de la localización del robot depende del mapa que esté construyendo, y la precisión del mapa depende de la posición precisa del robot en el entorno [14].

Para resolver esta problemática, los sistemas de robots móviles emplean una variedad de sensores para recopilar información del entorno y estimar tanto la posición del robot como la estructura del mapa. Una vez se recopilan los datos crudos obtenidos de los sensores, estos son procesados y fusionados para crear mediciones más confiables. Se utiliza algún algoritmo de localización para estimar la posición y orientación del robot en el entorno. Este algoritmo puede basarse en técnicas como el filtro de Kalman, el filtro de partículas, el algoritmo de gráficos de pose u otros [14].

Simultáneamente, se construye el mapa, empleando las mediciones de los sensores y las estimaciones de la posición del robot. Para esto, se requiere la integración de las observaciones sensoriales en un marco de referencia común. A medida que el robot se mueve, se recopilan más datos por lo que el algoritmo escogido actualiza el mapa y la estimación de la posición del robot constantemente. En algunos casos, se pueden emplear algoritmos de optimización para mejorar la distribución de las características del mapa del entorno y la estimación de la posición del robot. El proceso de SLAM es iterativo y complejo. Como se observa en la Figura 12, la información odométrica cruda recolectada por el sistema de sensores de un robot móvil no proporciona una representación precisa del mapa del entorno. Sin embargo, mediante la aplicación de algoritmos de mapeo y localización, esta información odométrica se fusiona entre sí para generar un mapa representativo del entorno circundante [9].



(a) Mapeo de entorno utilizando información odometrífica cruda.



(b) Mapeo de entorno después de fusionar la información odometrífica con algoritmo

Figura 12: Visualización del proceso de SLAM [9].

6.6. Filtro de Kalman extendido (EKF)

El Filtro de Kalman Extendido (EKF, por sus siglas en inglés) es una extensión del Filtro de Kalman clásico que se utiliza para estimar el estado de un sistema dinámico en presencia de ruido y no linealidades. Este algoritmo funciona iterativamente, actualizando continuamente las estimaciones del estado del sistema a medida que se reciben nuevas mediciones, utilizando una versión linealizada del modelo del sistema. Opera bajo dos etapas principales: la etapa de predicción y la etapa de actualización [14].

En la primera etapa, se predice el estado actual del sistema utilizando el modelo dinámico del mismo y el estado anterior estimado. Junto con la predicción del estado, se predice su covarianza (la incertidumbre en la estimación). En la segunda etapa, se incorporan las nuevas mediciones al estado predicho para mejorar la precisión de la estimación. Se calcula la ganancia de Kalman, que determina cuánta confianza se tiene en la estimación del estado del sistema en función de la diferencia entre las mediciones reales y las predichas por el algoritmo. Finalmente, se utiliza esta ganancia para corregir la estimación del estado del

sistema utilizando las mediciones disponibles de tal manera que se minimice el error de la estimación del estado [14].

Esta técnica de estimación es especialmente útil cuando el modelo del sistema y las mediciones no son lineales, lo que lo hace adecuado para sistemas como los robots móviles. Por esta razón, el EKF se utiliza en el contexto de SLAM para estimar la posición del robot en un entorno desconocido mientras construye un mapa del mismo. Siguiendo la metodología anteriormente expuesta, el EKF estima la posición y la orientación del robot en función de las mediciones odométricas y las lecturas de los sensores. A medida que el robot se mueve, se recopilan los datos de sus sensores para crear el mapa del entorno y corregir las estimaciones anteriores de la posición y orientación del mismo. Por medio de la ganancia de Kalman se compensa el error debido a pequeñas desviaciones en la rotación de las ruedas o deslizamientos en las mismas, mejorando así la precisión de la localización del robot y el mapa generado [9].

6.7. Light Detection and Ranging (LiDAR)

Los LiDAR son sistemas de teledetección que emplean pulsos de luz láser para medir distancias. Estos dispositivos emiten, de manera individual o continua, pulsos láser dirigidos hacia el objetivo que se va a medir. Al emitir un pulso láser, un circuito interno de sincronización se activa instantáneamente. Se mide el tiempo entre la emisión del pulso láser y su retorno al receptor. Los sensores receptores, al detectar la luz láser reflejada por el objetivo detienen el contador. El número de pulsos de reloj registrados durante la fase de emisión y detección es empleado para calcular la distancia hacia el objetivo. Esta tecnología es ampliamente utilizada en mediciones topográficas para generar mapas de alta resolución del terreno o la superficie. Además, son empleados en vehículos autónomos para la detección y percepción del entorno [15].

Esta tecnología puede clasificarse en dos tipos principales: el escáner LiDAR de una sola línea y el escáner LiDAR multilínea. El primero se utiliza para medir distancias con precisión en una dimensión específica, mientras que el segundo combina múltiples planos de escaneo para capturar mediciones en varias direcciones simultáneamente. Este último, emite pulsos láser en varias direcciones mientras gira y es utilizado en vehículos autónomos para resolver la problemática de SLAM y ofrecer un modelado del entorno más preciso [15].

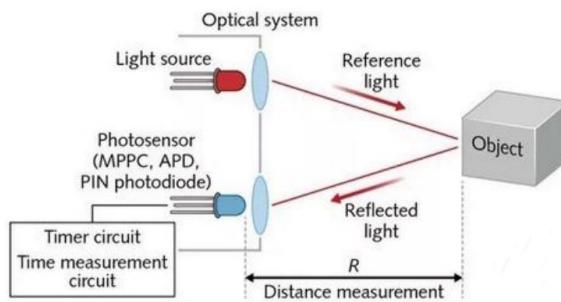


Figura 13: Funcionamiento LiDAR [15].

6.8. Sensores de distancia por tiempo de vuelo

Como su nombre lo indica, estos sensores emplean la tecnología de *Time-of-Flight* (ToF) para medir la distancia entre el sensor y un objeto. El principio de funcionamiento se basa en la velocidad de la luz. En sí, el sensor posee un emisor láser que cada cierto tiempo envía fotones que son reflejados por un objeto y detectados por el receptor. La diferencia de tiempo entre la emisión y la recepción proporciona la distancia real del objetivo en milímetros con una alta precisión. Los sensores ToF ofrecen una rápida respuesta independientemente del tamaño y color del objeto. Son empleados en una amplia gama de aplicaciones, como sistemas de detección de obstáculos, medición de niveles de líquido, reconocimiento de gestos en cámaras y sistemas de seguridad automotriz [16].

6.9. Vehículo diferencial *Pololu 3pi+ 32U4*

El vehículo diferencial Pololu 3pi+ 32U4 es un pequeño robot móvil, con un diámetro nominal de 9.7 cm, manufacturado por Pololu Corporation. Está diseñado para ser un robot de tipo diferencial, por lo que emplea dos ruedas motorizadas independientes para controlar su movimiento. Esto le permite moverse con facilidad y realizar maniobras precisas al variar la velocidad de cada rueda. Una característica notable del robot es que en su núcleo se encuentra un microcontrolador AT-mega32U4 AVR de Microchip, el cual proporciona capacidad de procesamiento y control. Se puede programar utilizando el entorno de desarrollo integrado IDE de Arduino, lo que lo hace accesible para implementar distintos algoritmos de control [17].

Cuenta con una variedad de sensores, que incluyen una unidad de medición inercial completa (acelerómetro, giroscopio y magnetómetro de 3 ejes), cinco sensores de reflectancia orientados hacia abajo para el seguimiento de bordes y sensores de impacto a lo largo de su cara frontal. Dispone de conexión USB para la comunicación con un ordenador, facilitando la programación del mismo en lenguajes como C o C++. Además, ofrece una serie de periféricos de expansión que permite a los usuarios personalizar y ampliar sus capacidades [17].



Figura 14: Pololu 3pi+ 32U4 OLED Robot [17].

CAPÍTULO 7

Validación de sensor de distancia tipo láser

En este capítulo se presentan las alternativas de sensores de distancia tipo láser consideradas y la validación de la selección del sensor más adecuado para la futura implementación de algoritmos de exploración y mapeo de entornos en agentes robóticos móviles. Cada uno de los sensores analizados presenta ventajas específicas, por lo que, con el fin de identificar aquel que mejor se adaptara a las necesidades de esta aplicación, se realizó un *trade study* en el cual se evaluaron diversos parámetros y especificaciones de cada uno de estos. Este análisis comparativo asegura que la elección final ofrezca el mejor rendimiento posible según criterios relevantes para la aplicación.

7.1. Sensores de distancia tipo láser considerados

Las opciones de sensores de distancia tipo láser consideradas para este trabajo fueron: el módulo VL53L0X, el LiDAR FHL-LD20 y el YDLIDAR Tmini Pro. Estos sensores fueron seleccionados estratégicamente para analizar dos tecnologías de medición diferentes: triangulación y tiempo de vuelo (ToF, por sus siglas en inglés). Ambas tecnologías han demostrado ser altamente efectivas en la obtención de mediciones de distancia para aplicaciones de navegación autónoma, haciendo que los sensores que las utilizan sean candidatos adecuados para la aplicación prevista.

7.1.1. Sensor VL53L0X

El sensor de distancia láser VL53L0X de STMicroelectronics es un módulo compacto con microcontrolador integrado que utiliza la tecnología de medición por tiempo de vuelo para obtener mediciones precisas, independientemente de la reflectancia del objetivo. Es capaz de

medir distancias con precisión en un rango de 50 mm a 2 m. Una característica destacada de este sensor es su emisor láser de cavidad vertical de 940 nm, el cual es completamente invisible para el ojo humano durante su funcionamiento.

Además, con un campo de visión (FoV) típico de 25°, el sensor incorpora una avanzada matriz SPAD, capaz de detectar fotones individuales incluso en condiciones de luz extremadamente débil. Esta capacidad, combinada con sus filtros anti-infrarrojos, mejora significativamente la sensibilidad y precisión de las mediciones en entornos con niveles altos de luz infrarroja. Adicionalmente, su interfaz I2C estándar facilita el control y transferencia de datos, lo que simplifica su integración en proyectos electrónicos. Por estas razones, es ampliamente utilizado en aplicaciones como prevención de colisiones, detección de bordes, o en sistemas de auto-enfoque en cámaras digitales.

7.1.2. LiDAR FHL-LD20

El LiDAR FHL-LD20, desarrollado por Youyeetoo, es un sensor omnidireccional de 360° que emplea tecnología de triangulación para escanear su entorno. Una vez recopilados los datos de distancia, el sensor combina estos valores con los ángulos medidos por la unidad de medición angular para generar una nube de puntos, que puede utilizarse para crear un mapa detallado del contorno circundante. Además, cuenta con un CCD fotosensible de alto rendimiento, el cual le permite medir distancias con precisión en un rango de 100 mm a 8 m.

El núcleo de medición de distancia del FHL-LD20 permite capturar 4000 puntos por segundos, lo que facilita la actualización del entorno capturado. Esta capacidad, junto con la modulación de la velocidad del motor, mejora la cobertura del sensor y reduce los puntos ciegos en distintos entornos. Adicionalmente, su comunicación serial UART simplifica la integración del LiDAR en una amplia variedad de aplicaciones, como la navegación autónoma en robots móviles, la reconstrucción de mapas 2D y la elusión de obstáculos.

7.1.3. YDLIDAR Tmini Pro

El YDLIDAR Tmini Pro es un LiDAR 2D de 360° desarrollado por EAI Technology. Emplea el principio de tiempo de vuelo para lograr mediciones de distancia con alta precisión. Durante la medición, su estructura mecánica gira 360 grados, lo que le permite obtener información continua de ángulos y realizar un escaneo completo del entorno. Entre sus características destacadas se encuentra su resistencia al polvo y al agua, lo que lo hace adecuado para operar en condiciones ambientales adversas.

Junto a esto, su tamaño compacto facilita su integración a otros sistemas, optimizando la estructura espacial en una amplia variedad de aplicaciones. Esto lo hace adecuado para pruebas en diversos escenarios. Utiliza comunicación serial UART para transmitir los datos de medición, lo que simplifica su conexión con sistemas de procesamiento. Con un rango de medición de 20 mm a 12 m, el YDLIDAR Tmini Pro es empleado en navegación y elusión de obstáculos en robots de servicio doméstico y aspiradoras robot, mejorando su eficiencia y capacidad de respuesta en entornos complejos.

7.2. Selección del sensor de distancia tipo láser

Las opciones de sensores de distancia tipo láser evaluadas fueron: VL53L0X, LiDAR LD14P y YDLIDAR Tmini Pro. Aunque los tres sensores comparten el mismo propósito, difieren en la metodología de medición y la cantidad de datos que generan, lo que presenta ventajas y desventajas específicas en función del objetivo y contexto de cada proyecto. Por esta razón, para realizar una comparación efectiva entre estos, se establecieron distintos criterios que se consideraron fundamentales para el desarrollo de este trabajo.

En el Cuadro 1 se presentan los criterios empleados en el *trade study* junto con el peso asignado a cada parámetro. Estos pesos fueron determinados mediante el proceso de ponderación por pares, un método utilizado para comparar y sopesar criterios cualitativos y cuantitativos entre sí. Para este estudio, se consideraron 10 criterios para asegurar una decisión fundamentada al escoger el sensor.

Criterios	Peso
Costo	10.19 %
Disponibilidad	14.58 %
Precisión	5.94 %
Rango de medición	6.91 %
Parámetros eléctricos	5.22 %
Velocidad de medición	6.09 %
Tamaño	6.17 %
Masa	7.46 %
Operatividad en aplicaciones deseadas	22.44 %
Adaptabilidad a robots móviles	15.00 %

Cuadro 1: Criterios de comparación.

El método de ponderación por pares emplea una matriz de comparaciones, Cuadro 2, para evaluar la importancia relativa entre cada par de criterios. Primero, se calcula la media geométrica de los valores en cada fila de la matriz. Luego, estos valores se normalizan para obtener los factores de ponderación, garantizando que la suma total sea igual a 1. Los valores para las comparaciones se seleccionan dentro de un intervalo 1 a 9, como se describe en el Cuadro 3.

	Costo	Disponibilidad	Precisión	Rango de medición	Parámetros eléctricos	Velocidad de medición	Tamaño	Peso	Operatividad	Adaptabilidad	Media Geométrica	Peso
Costo	1	1/4	1/2	3	2	5	4	4	1/5	1/3	1.1487	0.1019
Disponibilidad	4	1	3	2	2	4	3	3	1/3	1/4	1.6438	0.1458
Precisión	2	1/3	1	7	3	1	1/4	1/4	1/8	1/6	0.6700	0.0594
Rango de medición	1/3	1/2	1/7	1	5	1/3	5	5	1/3	1/4	0.7794	0.0691
Parámetros eléctricos	1/2	1/2	1/3	1/5	1	1/5	2	1/2	1/2	3	0.5887	0.0522
Velocidad de medición	1/5	1/4	1	3	5	1	1/4	1/4	1	1/2	0.6871	0.0609
Tamaño	1/4	1/3	4	1/5	1/2	4	1	1	1/5	1	0.6960	0.0617
Masa	1/4	1/3	4	1/5	2	4	1	1	1/3	1	0.8414	0.0746
Operatividad	5	3	8	3	2	1	5	3	1	1	2.5313	0.2244
Adaptabilidad	3	4	6	4	1/3	2	1	1	1	1	1.6917	0.1500
										Suma	11.2779	1.0000

Cuadro 2: Matriz de comparación por pares.

Valor	Descripción
1	Igual importancia
3	Moderadamente más importante
5	Fuertemente más importante
7	Muy fuertemente más importante
9	Extremadamente más importante
2,4,6,8	Valores intermedios entre los anteriores

Cuadro 3: Escala fundamental de comparación por pares.

7.2.1. Criterios para el *Trade Study* de los sensores de distancia tipo láser

El *trade study* es un método sistemático de toma de decisiones diseñado para identificar la solución técnica más óptima entre un conjunto de alternativas viables. Este enfoque consiste en evaluar y comparar diferentes opciones en función de una serie de criterios predefinidos, los cuales son ponderados numéricamente según su importancia relativa en el proyecto. A cada criterio se le asigna un peso específico, de manera que las alternativas que satisfacen más plenamente con dicho criterio obtengan una puntuación superior a aquellas que lo satisfacen en menor medida. El Cuadro 4 muestra las calificaciones posibles al desempeño de cada uno de los criterios evaluados.

Adjetivos(s)	Calificación
No aceptable / Nulo	0
Muy malo / Muy bajo	1
Malo / Bajo	2
Moderado / Normal	3
Bueno / Alto	4
Muy bueno / Muy Alto	5

Cuadro 4: Valores de interés para la comparación.

Costo

Dado que no se contaban con los dispositivos, fue necesario evaluar la compra de estos. En este criterio, se asignó la ponderación más alta al módulo VL53L0X, ya que, con un precio de aproximadamente \$6.00, es considerablemente más económico que los otros sensores de distancia. Este sensor es más de diez veces más barato que los LiDARs evaluados. Cabe señalar que las ponderaciones asignadas a cada sensor reflejan las limitaciones económicas del proyecto, donde se priorizaron las opciones que se ajustaran al presupuesto disponibles. En el Cuadro 15 se detallan los precios de los LiDARs, junto con características destacables que serán mencionadas en los criterios de evaluación.

Alternativa	Costo
VL53L0X	5
FHL-LD20	3
YDLIDAR Tmini Pro	1

Cuadro 5: Ponderación de costo.

Disponibilidad

Este criterio es considerado uno de los más importantes, ya que la disponibilidad de un sensor impacta directamente en distintos aspectos. Por un lado, un sensor que está ampliamente disponible permite realizar cambios rápidos y oportunos en caso de que sea necesario sustituirlo. Junto a esto, los sensores con amplia disponibilidad generalmente cuentan con un mejor soporte técnico, lo que facilita su mantenimiento y desarrollo. En este sentido, el módulo VL53L0X recibió la ponderación más alta, ya que es accesible en el mercado local de Guatemala. Eso sí, aunque los otros dos sensores son desarrollados en otros países y generan un cargo adicional por envío, ofrecen un excelente soporte técnico.

Alternativas	Disponibilidad
VL53L0X	5
FHL-LD20	3
YDLIDAR Tmini Pro	1

Cuadro 6: Ponderación de disponibilidad.

Precisión

Como se observa en el Cuadro 15, la precisión de cada sensor está especificada dentro de un rango, donde el primer valor representa la precisión en condiciones óptimas y el segundo valor corresponde a escenarios menos favorables. Aunque estos valores pueden variar según el entorno donde se utilicen, los rangos proporcionan una idea general de la precisión de cada sensor. En este contexto, se observa que ambos sensores de distancia tipo láser LiDAR tienen una precisión inferior en comparación con el módulo VL53L0X. En particular, el YDLIDAR muestra la menor precisión entre los tres sensores evaluados.

Alternativas	Precisión
VL53L0X	5
FHL-LD20	4
YDLIDAR Tmini Pro	3

Cuadro 7: Ponderación de precisión.

Rango de medición

El rango de medición del sensor debe ser adecuado para cubrir las dimensiones del área de trabajo. Puesto que se planea operar en el ecosistema Robotat, que cuenta con una plataforma de 4.8×3.8 m, es importante que el sensor tenga un rango suficiente para medir distancias dentro de este espacio sin restricciones. Además, al considerar su futura integración con robots móviles para el mapeo de entornos, un rango de medición limitado podría restringir significativamente su utilidad. En este contexto, el FHL-LD20 obtiene la ponderación más alta, ya que tiene la capacidad de cubrir áreas de trabajo moderadamente grandes como la plataforma del Robotat.

Alternativas	Rango de medición
VL53L0X	2
FHL-LD20	5
YDLIDAR Tmini Pro	4

Cuadro 8: Ponderación de rango de medición.

Parámetros eléctricos

El sensor debe ser compatible con el voltaje de operación del sistema en el que se integrará, ya que esto influye directamente en el diseño general del mismo. También, es importante conocer la corriente que consume para evitar sobrecargas en el alimentación. En aplicaciones móviles, un sensor con bajo consumo de energía es preferible, ya que prologa el tiempo de funcionamiento y reduce la frecuencia de recargas. Como se observa en el Cuadro 15, los tres sensores operan con voltajes de alimentación similares. Sin embargo, los sensores LiDAR muestran un mayor consumo de corriente durante su funcionamiento. Aunque este incremento es manejable al integrarlos con otros subsistemas, el módulo VL53L0X destaca por su menor consumo energético.

Alternativas	Parámetros eléctricos
VL53L0X	5
FHL-LD20	3
YDLIDAR Tmini Pro	3

Cuadro 9: Ponderación de parámetros eléctricos.

Velocidad de medición

Una mayor velocidad de medición permite capturar rápidamente los cambios en el entorno, lo que mejora la capacidad de respuesta en tiempo real. En este aspecto, tanto el sensor LiDAR FHL-LD20 como el YDLIDAR Tmini Pro pueden obtener mediciones a una velocidad de 4000 puntos por segundo, lo que les otorga una alta ponderación en este criterio. Por otro lado, la velocidad de medición del módulo VL53L0X depende del modo de

operación, con tiempos de respuesta típicos que oscilan entre 30 ms y 200 ms, lo que lo hace más lento en comparación con los otros sensores.

Alternativas	Velocidad de medición
VL53L0X	2
FHL-LD20	5
YDLIDAR Tmini Pro	5

Cuadro 10: Ponderación de velocidad de medición.

Tamaño

Un sensor compacto se integra más fácilmente en sistemas con espacio limitado como lo podría ser un robot móvil. Un tamaño reducido permite mayor flexibilidad en el diseño estructural del sistema. En este contexto, el módulo VL53L0X recibió la ponderación más alta. No obstante, los tamaños de los dos sensores también son manejables y adecuados para la aplicación en cuestión.

Alternativas	Tamaño
VL53L0X	5
FHL-LD20	3
YDLIDAR Tmini Pro	4

Cuadro 11: Ponderación de tamaño.

Masa

Cada gramo adicional puede impactar la eficiencia energética y la estabilidad de sistema. Un sensor liviano reduce la carga que el sistema debe soportar. Por esta razón, el módulo VL53L0X obtuvo la ponderación más alta en términos de peso. Sin embargo, los pesos de los otros dos sensores también son manejables y adecuados para la aplicación en cuestión.

Alternativas	Peso
VL53L0X	5
FHL-LD20	3
YDLIDAR Tmini Pro	4

Cuadro 12: Ponderación de peso.

Operatividad en aplicaciones deseadas

Este fue el criterio que se consideró más importante, pues determina cómo el sensor se ajusta a las necesidades específicas de la aplicación. Puesto que se requiere para futuras

aplicaciones de mapeo de entornos con robots móviles, la naturaleza unidireccional del módulo VL53L0X afecta su rendimiento en comparación con los otros dos sensores, que son omnidireccionales. El VL53L0X mide distancias en una sola dirección a la vez, lo que implica que para capturar un mapa completo del entorno, el sensor se debe girar o el robot cambiar su orientación. Esto puede complicar la eficiencia en la obtención de datos y aumentar el tiempo necesario para crear un mapa exhaustivo.

Por otro lado, los sensores omnidireccionales tienen la ventaja de capturar el entorno completo en una sola rotación. Eso reduce significativamente el tiempo necesario para obtener un mapa completo y facilita la interpretación de bordes y esquinas al realizar SLAM. Esta capacidad mejora notablemente la eficiencia en la programación de algoritmos de mapeo y planificación de trayectorias, haciéndolos ideales para aplicaciones de navegación autónoma, evasión de obstáculos y generación de mapas 2D. En contraste, el sensor unidireccional enfrenta desafíos adicionales en términos de localización y mapeo, dado que requiere múltiples capturas para obtener una visión completa del entorno.

Alternativas	Operatividad en aplicaciones deseadas
VL53L0X	1
FHL-LD20	5
YDLIDAR Tmini Pro	5

Cuadro 13: Ponderación de operatividad en aplicaciones deseadas.

Adaptabilidad a robots móviles

Este criterio, considerado como el segundo más importante, influye significativamente en la integración y el desempeño del sensor. Este criterio hace énfasis en la facilidad de montaje y mantenimiento al integrar el sensor al diseño estructural de distintos sistemas robóticos, asegurando que no interfieran con otros componentes. Otro aspecto considerado es el campo de visión, ya que determina la cobertura del entorno; un campo de visión más amplio proporciona una visión más completa. Además, se evalúa la robustez y resistencia del sensor frente a condiciones operativas adversas.

En este contexto, dado que se planea utilizar los sensores en aplicaciones futuras con robots móviles, la naturaleza unidireccional del módulo VL53L0X requiere la instalación de al menos cinco sensores para cubrir múltiples direcciones y recabar suficientes puntos. Este sensor también implicaría la necesidad de un sistema de rotación para obtener una cobertura completa sin la necesidad de instalar más de uno, lo que añadiría complejidad al diseño. En contraste, los sensores LiDAR están diseñados para ser montados en diversas posiciones y orientaciones, facilitando su ajuste en robots móviles. Además, su capacidad de operar en variedad de condiciones ambientales y su versatilidad en aplicaciones tanto interiores como exteriores los hace una opción más adaptable.

Alternativas	Adaptabilidad a robots móviles
VL53L0X	2
FHL-LD20	4
YDLIDAR Tmini Pro	5

Cuadro 14: Ponderación de adaptabilidad a robots móviles.

	VL53L0X	FHL-LD20	YDLIDAR Tmini Pro
Precio	\$6.00	\$60.00	\$100.00
Disponibilidad	Guatemala	Estados Unidos	Estados Unidos
Precisión	± 3 mm a ± 10 mm	± 10 mm a ± 20 mm	± 25 mm a ± 50 mm
Rango de medición	50 mm a 2 m	100 mm a 8 m	20 mm a 12 m
Parámetros eléctricos	DC 3.3 V @ 20 mA	DC 5 V @ 300 mA $i_{inrush} \leq 1.0$ A	DC 5 V @ 340 mA $i_{inrush} \leq 1.0$ A
Velocidad de medición	30 ms a 200 ms	4000 puntos/s	4000 puntos/s
Tamaño	$4.4 \times 2.4 \times 1.0$ mm	$96.3 \times 59.8 \times 38.8$ mm	$38.6 \times 38.6 \times 33.4$ mm
Peso	2.0 g	100.4 g	45.0 g
Operatividad en aplicaciones deseadas	Unidireccional	Omnidireccional	Omnidireccional
Adaptabilidad a robots móviles	I2C	UART	UART

Cuadro 15: Características de los sensores analizados.

7.2.2. Resultados del *trade study*

Una vez definidos los pesos y ponderaciones de los criterios presentados previamente, se procedió a realizar un esquema de ponderación para obtener los resultados del estudio comparativo. Como se muestra en la Figura 15, el gráfico de barras apiladas refleja el valor total de cada alternativa a través de la longitud de la barra segmentada. Cada segmento de la barra representa un criterio evaluado, y su longitud indica la magnitud de la ponderación obtenida. De acuerdo con la puntuación final, se determinó que el LiDAR FHL-LD20 era la opción más adecuada como sensor de distancia tipo láser. Por lo tanto, se seleccionó para utilizarlo en investigaciones con robots móviles dentro del ecosistema Robotat.

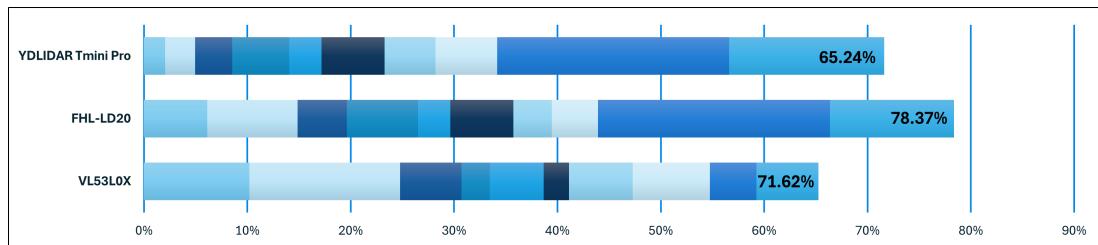


Figura 15: Resultados del *trade study*.

- Costo
- Disponibilidad
- Precisión
- Rango de medición
- Parámetros eléctricos
- Velocidad de medición
- Tamaño
- Peso
- Operatividad
- Adaptabilidad

Figura 16: Leyenda del gráfico de barras segmentadas.

Criterios	VL53L0X	FHL-LD20	YDLIDAR Tmini Pro
	Pesos	Pesos	Pesos
Costo	10.19 %	6.11 %	2.04 %
Disponibilidad	14.58 %	8.74 %	2.91 %
Precisión	5.94 %	4.75 %	3.56 %
Rango de medición	2.76 %	6.91 %	5.53 %
Parámetros eléctricos	5.22 %	3.13 %	3.13 %
Velocidad de medición	2.44 %	6.09 %	6.09 %
Tamaño	6.17 %	3.70 %	4.94 %
Masa	7.46 %	4.48 %	5.97 %
Operatividad en aplicaciones deseadas	4.49 %	22.44 %	22.44 %
Adaptabilidad a robots móviles	6.00 %	12.00 %	15.00 %
Total	65.24 %	78.37 %	71.61 %

Cuadro 16: Ponderación individual de los sensores evaluados según criterios establecidos.

CAPÍTULO 8

Lectura e interpretación de datos extraídos del LiDAR FHL-LD20

Este capítulo aborda el funcionamiento y la interpretación de datos del sensor LiDAR FHL-LD20. Se detalla el formato de los paquetes de datos enviados por el sensor, y se describe cómo se captura y procesa esta información utilizando software como Matlab y Python. Además, se presenta la aplicación desarrollada para que el usuario comprenda cómo operar el sensor y utilizar de manera efectiva los datos que proporciona.

8.1. Funcionamiento del sensor de distancia FHL-LD20

Esta sección se centra en el funcionamiento del sensor FHL-LD20, detallando sus características técnicas, protocolo de comunicación y la estructura de los datos que transmite. Se exploran los principios de operación que permiten al sensor capturar y transmitir información de manera eficiente. La comprensión de estos aspectos es fundamental para integrar y utilizar el sensor de manera efectiva en aplicaciones que requieran mediciones precisas.

8.1.1. Características generales

El sensor de distancia tipo láser FHL-LD20, fabricado por Youyeetoo, está diseñado para capturar datos en un plano bidimensional, generalmente en orientación horizontal. Utiliza un láser infrarrojo con una longitud de onda que varía entre 775 nm y 800 nm, siendo 793 nm el valor típico. Su campo de visión (FOV, por sus siglas en inglés) abarca una sección circular de 360° con un radio que varía entre 100 mm y 8 m. Además, cuenta con un motor eléctrico integrado que hace girar la ventana óptica de emisión y recepción del láser, permitiendo cubrir un ángulo de medición en todo el plano horizontal. A continuación, se presentan algunos parámetros relevantes del sensor:

Parámetros de rendimiento	
Resolución angular	0.54°
<i>Pitch</i>	0°~1.0°
<i>Yaw</i>	-0.5°~0.5°
Tiempo de arranque	≤ 3.4 s Típico de 3 s
Frecuencia de escaneo	6 Hz ± 0.2 Hz
Temperatura de funcionamiento	-10°C a 50°C
Humedad relativa	< 95 % RH
Nivel de seguridad del láser	IEC 60825 Clase 1

Cuadro 17: Parámetros de rendimiento.

Orientación del sensor

El LiDAR FHL-LD20 utiliza un sistema de coordenadas zurdo, donde el centro de rotación actúa como el origen de coordenadas (ver Figura 17). La dirección que conecta el centro de rotación con el centro de la rueda motriz se establece como la dirección de cero grados, y el ángulo de rotación aumenta en sentido horario. Ademas, el sensor tiene un margen limitado de inclinación y rotación en los ejes de *pitch* y *yaw*, por lo que si se superan los límites descritos en el Cuadro 17, su capacidad para medir distancias con precisión podría verse comprometida.

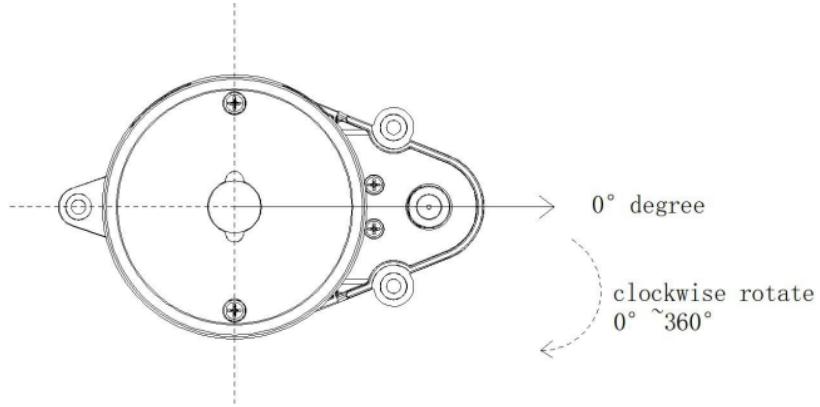


Figura 17: Orientación del sensor (FALTA REF)

8.1.2. Protocolo de comunicación

El FHL-LD20 utiliza un puerto serial asíncrono estándar (UART) para la transmisión unidireccional de datos. La conexión se realiza a través de un conector ZH1.5T-4P de 1.25mm, que se utiliza tanto para la alimentación como para la transmisión de datos hacia un dispositivo *Host*. En la Figura 18 se muestra el orden de los pines de entrada y salida del sensor, y en el Cuadro 19 se especifican los parámetros de transmisión.

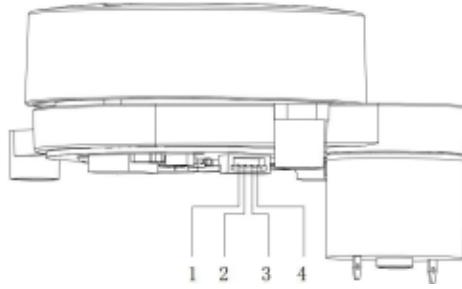


Figura 18: Esquema de conexión del conector ZH1.5T-4P del LiDAR FHL-LD20 (FALTA REF)

No. Pin	Señal	Tipo	Descripción	Voltajes de operación		
				Mínimo	Típico	Máximo
1	RX/PWM	Entrada	Control de velocidad externo	3 V	3.3 V	3.6 V
2	GND	Alimentación	Tierra	-	0 V	-
3	TZ	Salida	Salida de datos del LiDAR	3 V	3.3 V	3.6 V
4	VCC	Alimentación	Alimentación positiva	4.5 V	5 V	5.5 V

Cuadro 18: Parámetros de conexión del Lidar FHL-LD20.

Velocidad en baudios	Longitud de los datos	Bits de parada	Bit de paridad	Control de flujo
230400	8 bits	1	N/A	N/A

Cuadro 19: Parámetros de transmisión.

Formato de paquete

Cuando el sensor alcanza un estado de operación estable, comienza a enviar automáticamente paquetes de datos de medición sin requerir ninguna instrucción. A continuación, se detalla la estructura general de estos paquetes de medición y el significado de cada campo.

Carácter de inicio	VerLen	Velocidad del radar		Ángulo de inicio		Data	Ángulo final		Marca de tiempo		Verificación CRC
54H	2CH	LSB	MSB	LSB	MSB	...	LSB	MSB	LSB	MSB	1 Byte
↓											
Punto de medición 1			Punto de medición 2			...	Punto de medición 12				
Valor distancia 1	Intensidad		Valor distancia 2	Intensidad		...	Valor distancia 12	Intensidad			
LSB	MSB	1 Byte	LSB	MSB	1 Byte	...	LSB	MSB	1 Byte		

Cuadro 20: Estructura de un paquete de medición.

- Carácter de inicio (1 byte): Valor fijo de 0x54, señala el inicio de un paquete.
- VerLen (1 byte): Valor fijo de 0x2C, especifica que cada paquete contiene 12 puntos de medición.
- Velocidad del radar (2 bytes): Representa la velocidad del radar en grados por segundo.
- Ángulo de inicio (2 bytes): Indica el ángulo de inicio en unidades de 0.01 grados.

- Data (36 bytes): Contiene la información de 12 puntos de medición, cada uno de 3 bytes. Los primeros 2 bytes de cada punto indican el valor de distancia en milímetros, y el tercer byte refleja la intensidad de la señal de reflexión; los valores más altos indican una mayor intensidad.
- Ángulo final (2 bytes): Señala el ángulo de finalización de la medición, en unidades de 0.01 grados.
- Marca de tiempo (2 bytes): Representa el tiempo en milisegundos en el que se capturaron los datos.
- Verificación CRC (1 byte): Byte utilizado como *checksum* para asegurar la integridad de todos los datos anteriores que forman un paquete.

8.2. Lectura de datos

La integración de microcontroladores y software de análisis es fundamental para el desarrollo de sistemas de captura y procesamiento de datos en aplicaciones de sensores. En esta sección, se describe el proceso de lectura y decodificación de datos del sensor FHL-LD20, utilizando un microcontrolador ESP32 y plataformas de software como Matlab y Python. También, se presenta la aplicación desarrollada para la recolección e interpretación de datos del sensor, proporcionando una herramienta útil para evaluar su precisión y confiabilidad en diversos escenarios.

8.2.1. Lectura e interpretación de datos mediante ESP32 y Matlab

Para desarrollar el prototipo inicial de un sistema de captura y análisis de datos del sensor FHL-LD20, se utilizó un microcontrolador ESP32, programado mediante la plataforma de Arduino, en combinación con el software Matlab. El microcontrolador ESP32 se encargó de la recolección de información a través de la conexión UART proporcionada por el sensor. Este primer modelo permitió realizar una evaluación preliminar del rendimiento del sensor. En la Figura 19 se muestra el proceso general de captura de datos en términos generales.

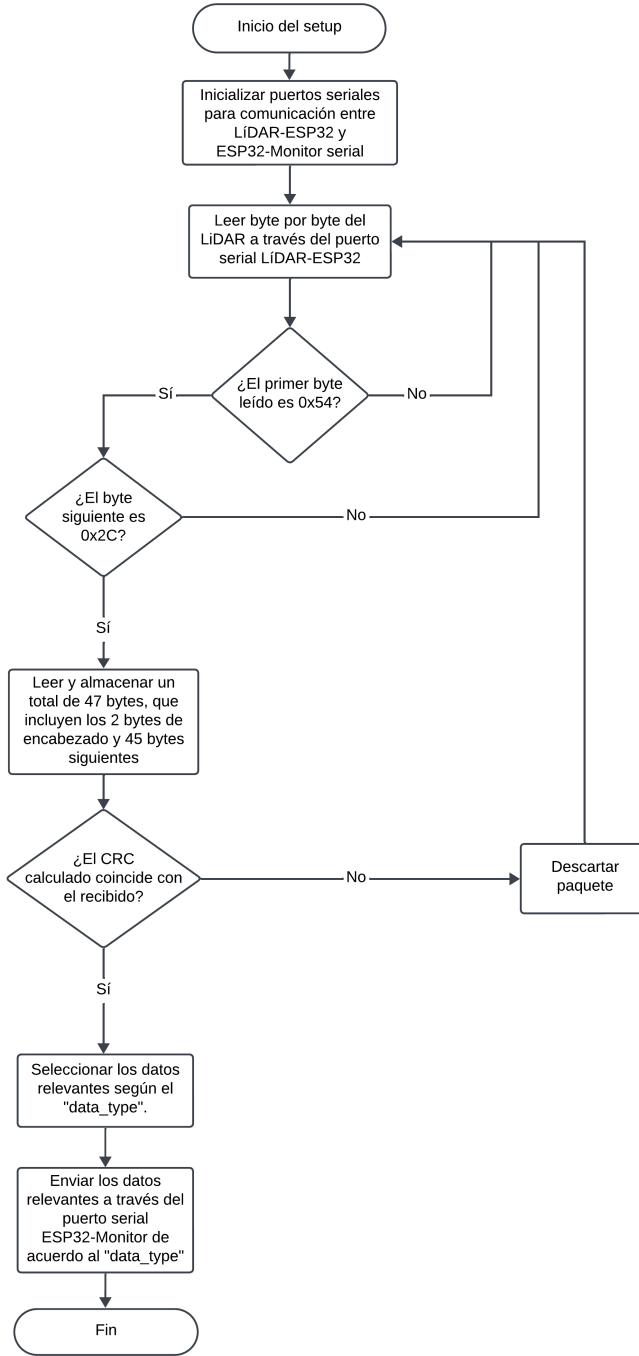


Figura 19: Diagrama de flujo general para captura de datos provenientes del sensor.

El proceso de captura de datos sigue una estructura de validación en la que se espera recibir un paquete de 47 bytes. Cada paquete se valida mediante los dos bytes de encabezados específicos (0x54 y 0x2C), una vez recibido el paquete completo, se realiza una verificación de integridad utilizando un cálculo de verificación por redundancia cíclica (CRC) de 8 bits. El código opera de manera cíclica en el bucle principal, lo que permite un flujo continuo de lectura de datos del LiDAR.

En el diagrama de flujo presentado se destaca la importancia de la variable “data_type”. Esta variable define la cantidad de datos que el microcontrolador enviará a Matlab para su procesamiento. Dicha variable puede asumir uno de tres valores predefinidos: “FULL_DATA”, “RELEVANT_DATA” o “RELEVANT_DATA2”, cada uno especificando una longitud distinta para los datos a transmitir.

- “FULL_DATA”: Corresponde a un paquete completo de 47 bytes (94 caracteres).
- “RELEVANT_DATA”: Paquete de 30 bytes que excluye la velocidad del radar, los bytes de intensidad de la señal, los de la marca de tiempo y el byte de verificación CRC (60 caracteres).
- “RELEVANT_DATA2”: Paquete de 42 bytes que, a diferencia de “RELEVANT_DATA”, incluye los bytes de intensidad de la señal (84 caracteres).

La flexibilidad en el tamaño de los paquetes fue implementada para optimizar el tiempo de procesamiento en Matlab, lo cual es particularmente relevante en aplicaciones en tiempo real o cuando se manejan múltiples paquetes de datos en rápida sucesión. Al reducir la cantidad de información procesada, se logra una respuesta más rápida del sistema sin necesidad de realizar cambios significativos en el software. Cabe destacar que los datos capturados se envían como una cadena de caracteres hexadecimales a través del puerto serial.

Una vez capturada la información del sensor, se utilizó Matlab para desarrollar un programa capaz de recibir los datos transmitidos desde el microcontrolador. El software permitió obtener y decodificar múltiples lecturas consecutivas, así como graficar los resultados. Este proceso se utilizó para verificar la concordancia de las lecturas del sensor con los valores esperados, además de ofrecer una herramienta sencilla para manipular los datos recibidos. La Figura 20 muestra el proceso de recepción, decodificación y visualización de los datos obtenidos.

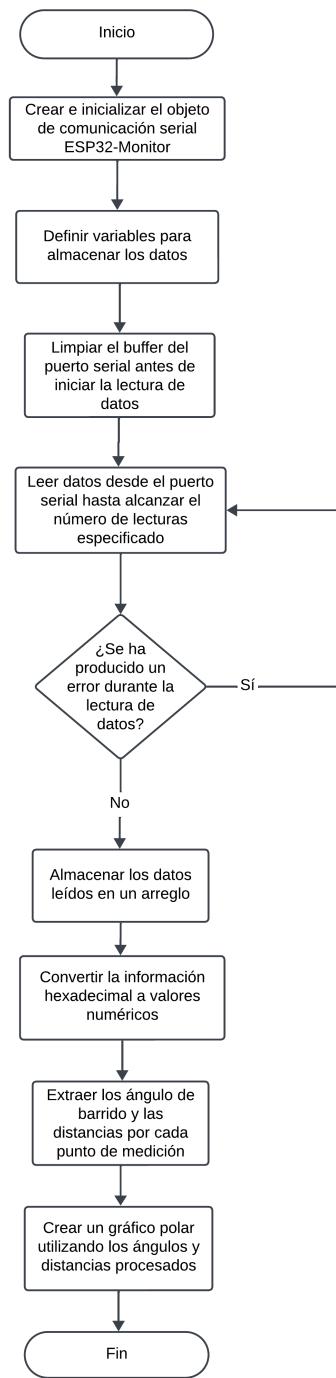


Figura 20: Diagrama de flujo general para procesamiento de datos provenientes del sensor.

Como se mencionó anteriormente, el paquete de datos recibido consiste en un arreglo de caracteres que contiene información estructurada en formato hexadecimal. La conversión de datos se realiza dividiendo el paquete en segmentos y extrayendo pares de caracteres hexadecimales. Para reconstruir los valores completos de ángulo y distancia, se combinan dos partes: un byte de menor orden y un byte de mayor orden.

Por ejemplo, para el ángulo de inicio codificado en dos bytes, se extraen las partes de menor y mayor orden para obtener los valores hexadecimales correspondientes. Luego, se desplaza el byte de mayor orden a la posición correcta dentro del valor de 16 bits y se suma con el byte de menor orden para obtener el valor completo. De esta manera, se reconstruye el valor decimal correspondiente al ángulo de inicio medido, expresado en unidades de 0.01 grados, en el paquete de datos evaluado.

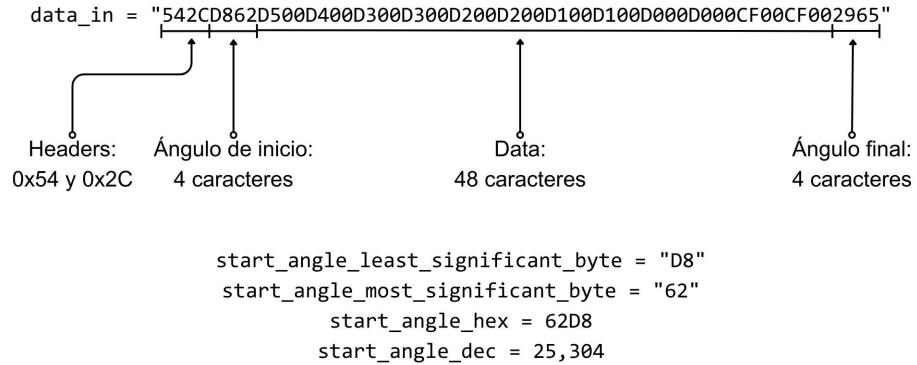
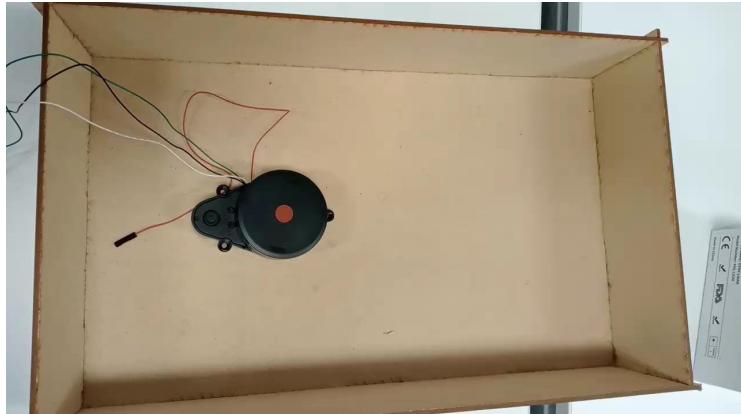


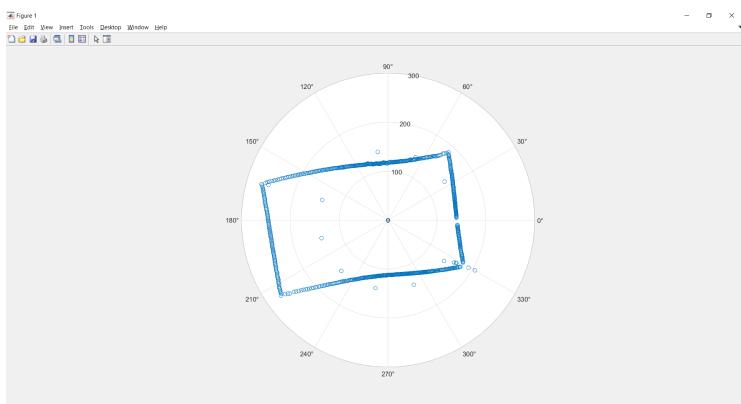
Figura 21: Ejemplificación del proceso de reconstrucción de mediciones: ángulo de inicio.

Una vez que las lecturas fueron decodificadas y segmentadas en sus partes correspondientes, se calculó el ángulo de barrido para cada punto de distancia. Este cálculo se realizó determinando el paso angular, el cual fue obtenido al dividir la diferencia entre los ángulos de inicio y fin por el número de intervalos entre los puntos de medición. Este incremento angular fue sumado acumulativamente al angulo de inicio. De esta manera, a cada punto de distancia medido se le asignaría un ángulo de barrido específico. Cabe destacar que la diferencia entre los ángulos de inicio y fin fue de aproximadamente 6° , lo que implica que cada paquete de datos cubre un barrido de esta magnitud.

Los valores de distancias y ángulos resultantes del procesamiento de datos se organizaron en una matriz de datos la cual se representó en una gráfica de coordenadas polares en Matlab para recrear la visualización generada por el LiDAR. Para verificar su funcionamiento y realizar un reconocimiento de área, se colocó el sensor FHL-LD20 dentro de una caja de 38.5×23 cm. A continuación, se presenta la disposición del sensor dentro de la caja y la reconstrucción visual realizada a partir de 60 paquetes del puerto serial, es decir, 720 puntos de medición procesados o bien, una revolución completa.



(a) Disposición del sensor FHL-LD20 dentro de la caja



(b) Reconstrucción de las 70 lecturas del sensor

Figura 22: Visualización de la primera corrida del sensor: 840 puntos de medición.

A partir de la Figura 22b, se identificaron distintas áreas de mejora al momento de realizar la reconstrucción. En primer lugar, aunque la forma y las proporciones generales de la caja se representaron correctamente, la reconstrucción se presenta de manera “invertida” en comparación con la disposición física observada (Figura 22a). La causa de esta discrepancia radica en que el LiDAR captura los datos en dirección horaria, mientras que los sistemas de visualización convencionales suelen interpretar los datos en sentido antihorario. Como resultado, la orientación visual de la reconstrucción difiere con lo que se observa a simple vista. Este hecho resaltó la necesidad de implementar una opción para ajustar la orientación de los puntos capturados, de modo que la reconstrucción se alinee con la disposición física y sea más intuitiva.

Otro aspecto destacable es la ligera curvatura observada en las aristas de la reconstrucción, que difiere con los bordes rectos de la caja física. Esta distorsión surgió debido a que los datos se procesaron sin considerar el ajuste para alinear los puntos obtenidos con el sistema de referencia deseado (eje central de rotación del sensor como origen). Es decir, no se aplicó una compensación adecuada para corregir los desfases entre el sistema de coordenadas en el que se capturan las mediciones (propio del LiDAR) y el sistema de coordenadas en el que se deberían de visualizar los datos (eje central). Como resultado, la reconstrucción presenta una leve deformación en comparación con la realidad física observada.

Además, se identificó un conjunto de aproximadamente 10 puntos distribuidos de forma circular que no deberían de estar presentes en la reconstrucción. Al revisar el procesamiento de datos, se determinó que estos puntos atípicos surgieron debido a un error en el cálculo del barrido angular. Cuando el ángulo de inicio era mayor que el ángulo final, se obtenía una diferencia negativa, lo que provocaba que las mediciones cercanas al cruce del punto de 0 grados se distorsionaran. Para solventar esto, se implementó una corrección en el cálculo del ángulo, añadiendo 360 grados cuando el ángulo de inicio superaba al ángulo final. Esta corrección garantizaba una transición continua entre mediciones al momento en que el sensor completaba un ciclo y cruzaba por el ángulo de 0 grados. Estos detalles no solo resaltaron la importancia de revisar cuidadosamente el procesamiento de datos, sino también la necesidad de diseñar un soporte para estabilizar y alinear adecuadamente el sensor LiDAR.

8.2.2. Lectura e interpretación de datos mediante ESP32 y Python

En lugar de desarrollar únicamente un software para obtener y procesar los datos, se decidió crear una aplicación que mejorara la experiencia del usuario al utilizar el sensor FHL-LD20. Para ello, se optó por utilizar Python 3.11 como lenguaje de programación, considerando la necesidad de un entorno más flexible y eficiente para el manejo de datos. Las principales razones por las cuales se seleccionó Python sobre Matlab, que se había usado anteriormente, fueron las siguientes:

- A diferencia de Matlab, Python es un lenguaje de código abierto y gratuito.
- Python cuenta con un ecosistema amplio, con una gran variedad de bibliotecas y *frameworks* que facilitan la integración de diferentes funcionalidades en una misma aplicación.
- Para el desarrollo de aplicaciones de escritorio, Python ofrece múltiples frameworks como Tkinter, PyQt y Kivy, mientras que Matlab solo dispone de App Designer para la creación de interfaces gráficas.
- Python cuenta con una comunidad de desarrolladores muy grande y activa, lo que facilita la resolución de problemas y la implementación de nuevas funcionalidades.

Cabe señalar que el código desarrollado para el ESP32, se mantuvo sin cambios, ya que mostró ser eficiente para la recolección de datos.

Ahora bien, para el desarrollo de la aplicación, se empleó CustomTkinter, una extensión moderna y estilizada de Tkinter. Esta elección no solo permitió mejorar la estética de la interfaz, sino que también ofreció una experiencia más funcional en comparación con herramientas tradicionales, como Tkinter. En la Figura 23, se presenta la interfaz gráfica nombrada “LiDAR MapViewer”, junto con las distintas opciones y paneles disponibles dentro de la aplicación. Es importante destacar que el procesamiento de datos siguió la misma estructura general que el diagrama de flujo presentado en la Figura 20. Las modificaciones específicas realizadas se detallan en el siguiente capítulo.

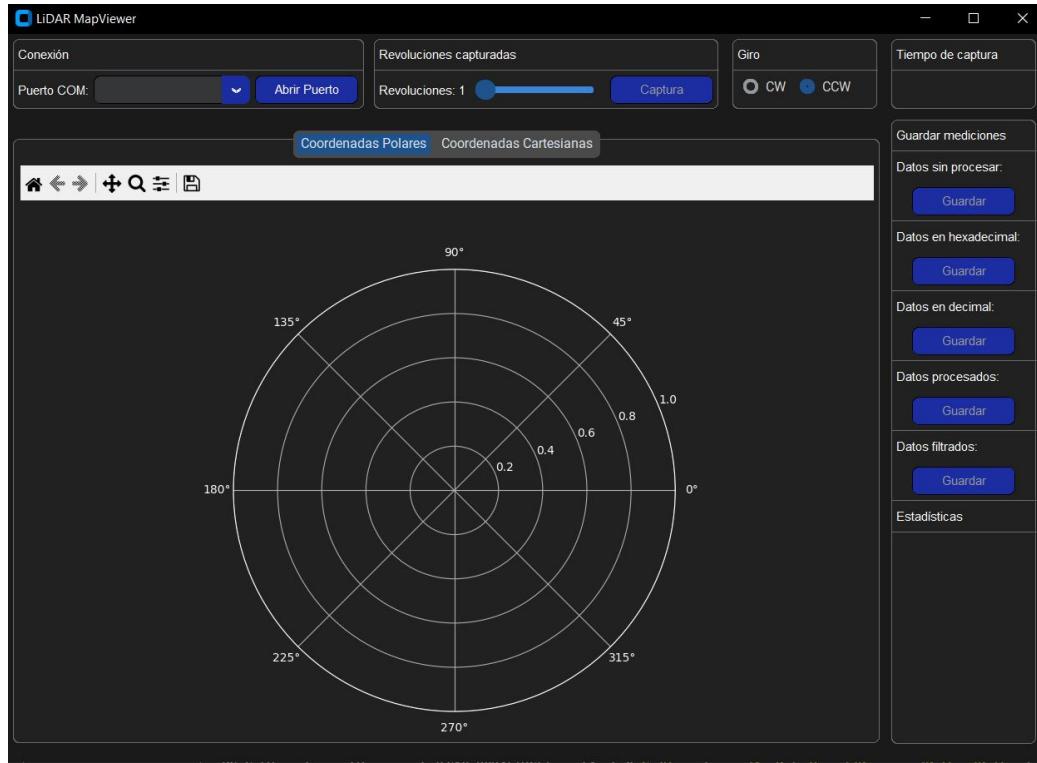


Figura 23: Vista de usuario, interfaz gráfica LiDAR MapViewer.

- Panel de conexión: Este panel gestiona la conexión entre el ESP32 y el monitor a través de un puerto serial, que puede seleccionarse mediante un menú desplegable que muestra los puertos COM disponibles. La conexión se habilita mediante un botón, siempre que el puerto COM se encuentra activo y disponible.
- Panel de revoluciones capturadas: Este marco contiene los controles para seleccionar y visualizar el número de revoluciones capturadas por el LiDAR. Incluye un control deslizante que permite elegir entre 1 y 5 revoluciones, y un botón para iniciar la captura de datos.
- Panel de giro: Este panel cuenta con dos botones que permiten al usuario seleccionar la orientación del giro de los datos capturados por el LiDAR. Esta selección ajusta el cálculo de los ángulos para su visualización.
- Panel de tiempo de captura: Muestra el tiempo total empleado por la interfaz para capturar, procesar y graficar todas las muestras seleccionadas por el usuario. El panel se actualiza al finalizar el proceso de captura.
- Panel de gráficos: Este marco contiene los gráficos de la interfaz, donde se visualizan los resultados de los datos capturados por el LiDAR, tanto en formato polar como cartesiano.
- Panel de guardado de mediciones: Este panel ofrece cinco opciones para almacenar distintos conjuntos de datos en archivos CSV:
 - Datos sin procesar.

- Datos segmentados en hexadecimal.
- Datos segmentados en decimal.
- Datos procesados de ángulos y distancias.
- Datos filtrados de ángulos y distancias.

Cada botón permite al usuario elegir la ubicación donde se guardará la información, simplificando así la organización y el análisis de los datos recopilados.

- Panel de pruebas estadísticas: Sección en la cual se muestran los análisis estadísticos de las aristas capturadas por el sensor. Esta sección solamente aplica para reconstrucciones de entornos cuadriláteros.

CAPÍTULO 9

Caracterización del LiDAR FHL-LD20

En este capítulo se abarca la caracterización del sensor LiDAR FHL-LD20, lo cual implica evaluar su rendimiento en diferentes condiciones operativas. Este análisis tiene como objetivo estudiar su comportamiento ante factores como las propiedades de la superficie del objeto, la distancia y el ángulo de incidencia. Esto permite determinar parámetros clave, como la precisión y exactitud de las mediciones, el rango de detección efectivo, la resolución angular y la sensibilidad a factores externos. Este proceso es fundamental para validar las especificaciones proporcionadas por el fabricante, que pueden no reflejar el rendimiento en el entorno real de uso. Además, la caracterización permite adaptar el sensor a las condiciones específicas de la aplicación, identificar limitaciones, y optimizar su configuración para maximizar su rendimiento en la implementación final.

9.1. Compensación en las mediciones del LiDAR FHL-LD20

En la primera reconstrucción realizada (Figura 22), se observaron deformaciones en las aristas de la caja, que no coincidían con los bordes rectos de la caja física. Estas deformaciones fueron causadas por la falta de corrección angular debida al desplazamiento físico del emisor/receptor del láser dentro de la carcasa. Este desplazamiento implica que el punto exacto desde donde se emite el láser y donde el sensor recoge las reflexiones está ligeramente descentrado con respecto al eje de rotación del sensor.

Cuando el láser no se emite desde el centro del círculo ideal que describe el barrido del LiDAR, se produce un desfase que afecta la precisión angular. Sin embargo, este desplazamiento interno no afecta la distancia medida, ya que sigue siendo una medición directa entre el sensor y el objeto. La luz viaja y vuelve directamente, similar a cómo se usaría una regla para medir la distancia en línea recta. El problema radica en que, un desfase angular interno provoca que el punto aparezca en una ubicación incorrecta en el plano, es decir, más a la

izquierda o derecha de su verdadera posición.

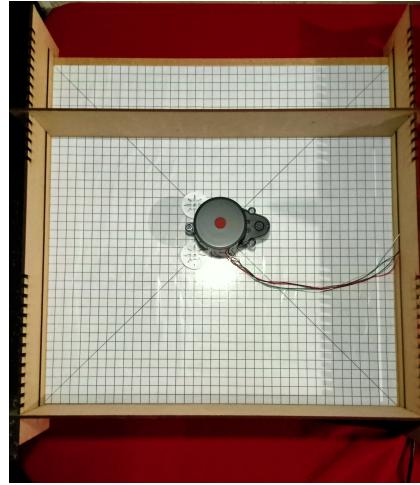
El hecho que el sensor gira sobre un eje para medir diferentes ángulos, si el sensor se encuentra desplazado de ese centro de rotación, los ángulos medidos no representarán correctamente la posición de los objetos en el entorno. Para corregir este error, en la documentación proporcionada por el fabricante, se sugería consultar el proceso de corrección realizado a partir del código fuente (FALTA REF). En este se utilizan dos valores ajustados: x_val y y_val , que incorporan los desplazamientos en los ejes X y Y. Estos valores no son las coordenadas cartesianas de cada punto evaluado, sino ajustes que ayudan a calcular el error angular causado por el desfase físico del sensor.

$$x_val = distancia + x_offset \quad (4)$$

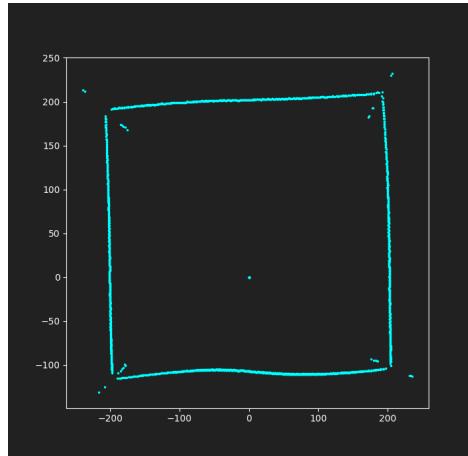
$$y_val = distancia \times y_factor + y_offset \quad (5)$$

En la Ecuación 5, el valor ajustado y_val utiliza una constante “ y_factor ” de 0.11923. Aunque la documentación no especifica la razón exacta de este valor, se puede inferir que su propósito era corregir la proyección angular de las mediciones. Probablemente compensando una inclinación natural del sensor o representando un ajuste empírico destinado a mejorar la precisión de las mediciones. A partir de los valores ajustados, se calcula un ángulo de corrección usando la función arcotangente de la relación y_val sobre x_val . Este ángulo de corrección refleja cuánto debe sumarse o restarse al ángulo original para compensar el desplazamiento interno del sensor.

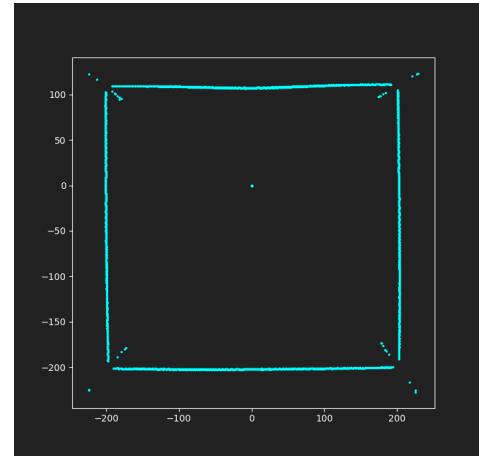
Si el sistema utiliza coordenadas en sentido antihorario, el ángulo se invierte (restándole 360 grados) y luego se suma el ángulo de corrección para obtener el valor final. En caso contrario, el ángulo de corrección se resta directamente del ángulo original. La Figura 24, muestra la disposición del sensor dentro de la caja, junto con una comparación de la reconstrucción visual obtenida a partir de tres revoluciones completas capturadas, utilizando el ángulo sin y con corrección.



(a) Disposición del sensor FHL-LD20 dentro de la caja



(b) Reconstrucción de tres revoluciones completas capturadas sin ángulo de corrección



(c) Reconstrucción de tres revoluciones completas capturadas con ángulo de corrección

Figura 24: Comparación de la reconstrucción con y sin ángulo de corrección: tres revoluciones completas capturadas.

9.2. Depuración de mediciones nulas en el LiDAR FHL-LD20

En el procesamiento de datos, tras haber obtenido las distancias en formato decimal, se emplean estas mediciones para calcular los valores de ajustes indicados anteriormente. En el diagrama de flujo (Figura 25), se puede apreciar que el procesamiento depende de si la distancia es mayor a cero. Cuando es mayor se calcula tanto los valores de ajuste como el ángulo de desplazamiento real. Una vez obtenido el ángulo real, se actualiza el valor del ángulo de corrección antiguo al más reciente.

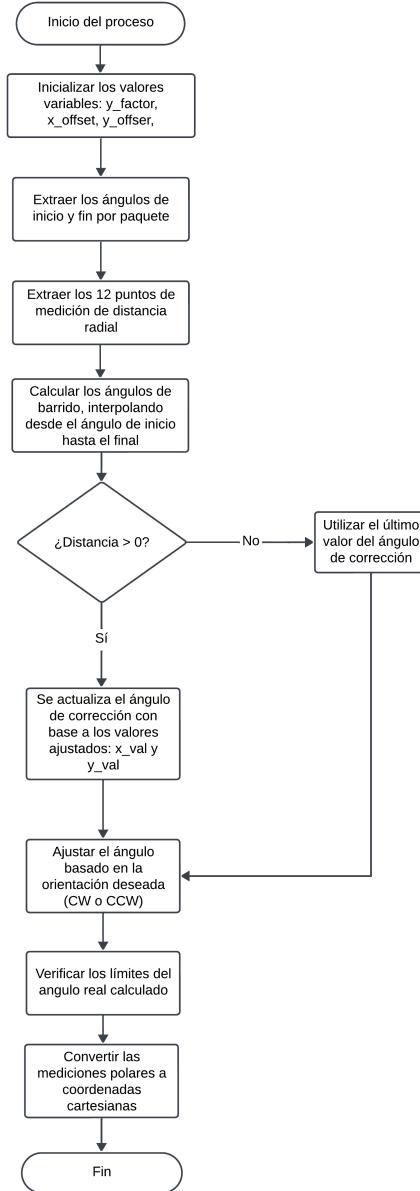
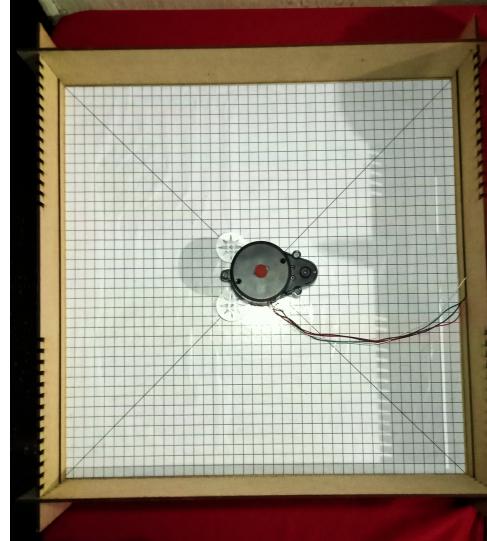


Figura 25: Diagrama de flujo general para procesamiento de datos en formato decimal.

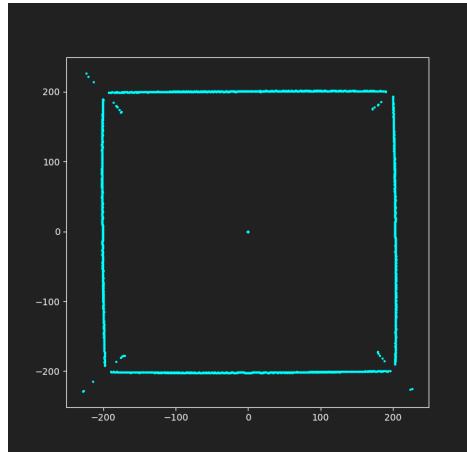
Ahora bien, si la distancia medida es igual a cero, no se recalculan los valores de ajuste, ya que la medición carece de información para ese punto. En su lugar, se reutiliza el último ángulo de corrección calculado, para calcular el ángulo real de esa medición nula. Este proceso asegura que incluso las mediciones nulas mantengan una continuidad angular coherente en relación con las mediciones anteriores, evitando distorsiones en la reconstrucción del entorno.

Sin embargo, incluir las mediciones nulas en los gráficos de coordenadas polares y cartesianas resultaría innecesario, ya que, físicamente, el sensor no es capaz de registrar mediciones justo en su eje de rotación. Además, estos puntos añadirían confusión visual al interpretar el entorno capturado. Por estas razones, se optó por filtrar las mediciones cuya distancia radial

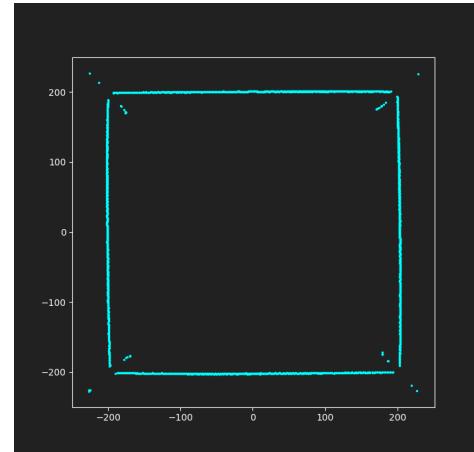
fuese cero. Al eliminar estos puntos en la generación de los gráficos se preservan solo las distancias que aportan información relevante sobre el entorno. Este filtrado mejora la claridad y precisión de la representación de los objetos detectados, lo cual mejora la interpretación de los datos capturados. En la Figura 26 se muestra la comparación antes y después de depurar las mediciones nulas capturadas.



(a) Disposición del sensor FHL-LD20 dentro de la caja



(b) Reconstrucción de tres revoluciones completas capturadas sin filtro de mediciones nulas



(c) Reconstrucción de tres revoluciones completas capturadas con filtro de mediciones nulas

Figura 26: Comparación de la reconstrucción con y sin filtro de mediciones nulas: tres revoluciones completas capturadas.

9.3. Rango de medición efectivo del LiDAR FHL-LD20

Determinar el rango de medición efectivo implica evaluar las distancias mínima y máxima a las cuales el sensor puede operar de con precisión y confiabilidad. El rango mínimo de medición es fundamental para identificar las limitaciones del sensor en distancias cercanas,

mientras que el rango máximo define la extensión de cobertura que el sensor puede alcanzar. Conocer estos rangos permite ajustar el sensor para satisfacer las necesidades específicas de la aplicación en la que se utilizará.

9.3.1. Rango mínimo de medición

Para evaluar la distancia mínima que el sensor es capaz de medir, se optó por diseñar entornos circulares con diámetros cercanos a la distancia mínima especificada por el fabricante. El objetivo era identificar el rango mínimo que permitiera un funcionamiento adecuado del sensor. Optar por un entorno circular en lugar de uno cuadrado resultó ventajoso, ya que la naturaleza del sensor favorece una cobertura uniforme en todas las direcciones.

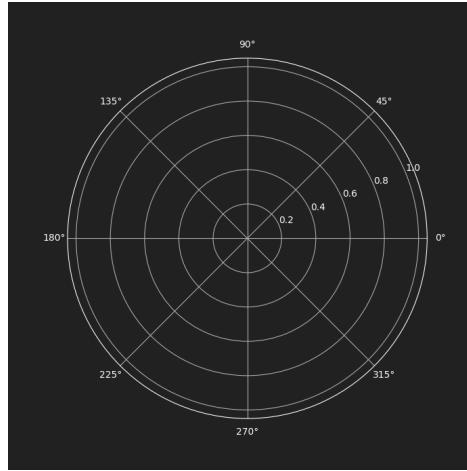
En la Figura 28 se muestra la primera prueba realizada, utilizando un contorno circular de aproximadamente 6 cm de radio. Como se aprecia en las Figuras 28b y 28c, el contorno no pudo ser reconstruido, y la gráfica resultó vacía. Al analizar los datos sin procesar del sensor durante este experimento, se descubrió que el sensor no era capaz de registrar distancias válidas, por lo que se registraban como cero. Y, debido a que las mediciones se filtraban para eliminar las distancias nulas, la reconstrucción de los datos resultó vacía.

Es importante señalar que, a pesar de la falla de mediciones de distancia, el sensor sí capturaba los ángulos de inicio y final, como se muestra en la Figura 27. Esto sugiere que la cobertura del campo de visión del sensor es independiente de su capacidad para registrar distancias radiales válidas.

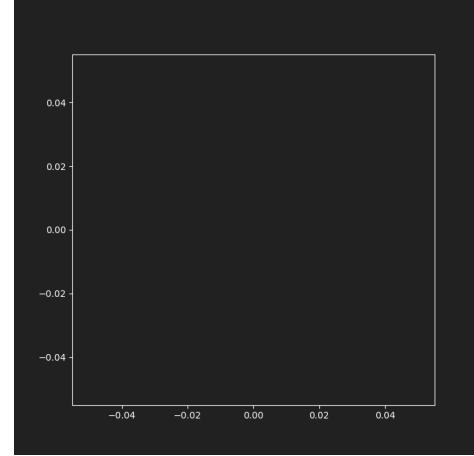
Figura 27: Ejemplificación de las lecturas nulas durante la reconstrucción de mediciones: entorno circular de 6 cm de radio.



(a) Disposición del sensor FHL-LD20 dentro del entorno circular: 6 cm de radio



(b) Reconstrucción en coordenadas polares de tres revoluciones completas capturadas en entorno circular: 6 cm de radio



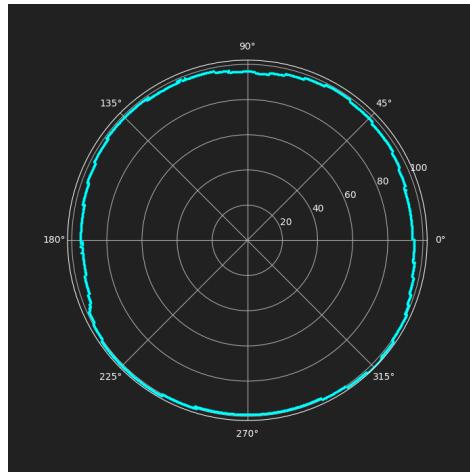
(c) Reconstrucción en coordenadas cartesianas de tres revoluciones completas capturadas en entorno circular: 6 cm de radio

Figura 28: Reconstrucciones vacías de tres revoluciones completas capturadas en entorno circular: 6 cm de radio

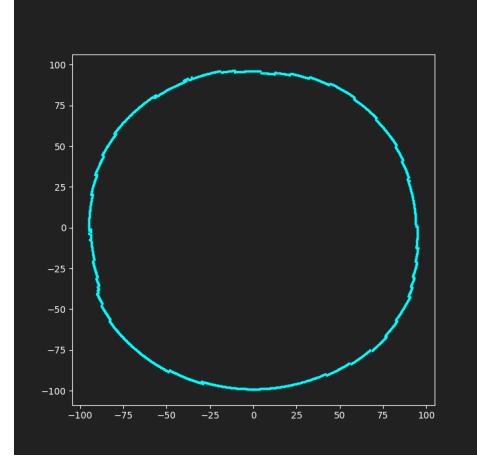
Se realizaron pruebas adicionales con radios circulares de 7, 8, 9 y 10 cm (ver Anexos 36 y 37). El entorno circular cuyo radio era de 9 cm (Anexo 38) fue el primero en mostrar indicios de recolección de datos. Sin embargo, la reconstrucción de la geometría no fue completamente precisa. Se llega a visualizar una geometría circular irregular. Hay variaciones en la distribución de los puntos, y en algunos tramos, los puntos parecen más dispersos. En contraste, la reconstrucción del entorno circular de 10 cm (Figura 29) refleja una geometría más continua. La forma general fue más precisa, reflejando un mejor ajuste a la geometría esperada. Este comportamiento era de esperarse, pues el sensor tiene un rango mínimo de medición de 10 cm, por lo que la geometría del entorno de 9 cm se encuentra muy cerca de este umbral. Esto provoca una mayor inexactitud en las mediciones, pues el sensor no se encuentra optimizado para medir distancias tan cortas con precisión.



(a) Disposición del sensor FHL-LD20 dentro del entorno circular:
10 cm de radio



(b) Reconstrucción en coordenadas polares de
tres revoluciones completas capturadas en
entorno circular: 10 cm de radio



(c) Reconstrucción en coordenadas cartesianas
de tres revoluciones completas capturadas en
entorno circular: 10 cm de radio

Figura 29: Reconstrucciones vacías de tres revoluciones completas capturadas en entorno circular:
10 cm de radio

9.3.2. Rango máximo de medición

FALTAN HACER PRUEBAS EN UNA HABITACIÓN.

9.4. Resolución del LiDAR FHL-LD20

FALTAN HACER PRUEBAS PARA COMPRENDER LA CAPACIDAD DEL SENSOR
PARA DISTINGUIR ENTRE DOS OBJETOS CERCANOS.

9.5. Evaluación angular del LiDAR FHL-LD20

FALTAN HACER PRUEBAS PARA COMPARAR LAS MEDICIONES ANGULARES DEL LIDAR CON LOS VALORES REALES PARA ANALIZAR SU EXACTITUD ANGULAR.

9.6. Precisión del LiDAR FHL-LD20

Para evaluar la precisión del sensor en las mediciones de distancias y ángulos, se llevó a cabo un análisis basado en la reconstrucción de cajas de diferentes dimensiones. Este proceso incluyó la determinación de regresiones lineales para las aristas detectadas y el cálculo de diversas métricas que permiten evaluar la fidelidad de las mediciones del sensor. A continuación se detalla el proceso para evaluar la precisión del sensor para una caja de 400×400 mm.

En primer lugar, se aplicó un filtro para seleccionar los puntos más cercanos a las aristas esperadas, utilizando un intervalo de tolerancia de ± 5 mm. Se excluyó una franja cercana a los bordes de la caja para evitar distorsiones derivadas al efecto de las esquinas, que suelen generar mediciones menos precisas debido a la complejidad de la geometría de estas áreas. Posteriormente, se realizó un ajuste de regresión lineal para cada conjunto de puntos que conformaba una arista de la caja reconstruida. En el caso de las aristas horizontales, el valor de los valores en el eje Y se ajustó en función de los valores del eje X. Mientras que, para las aristas verticales, se ajustaron los valores del eje X en función de los valores del eje Y.

A partir de los puntos filtrados y las regresiones obtenidas, se calcularon las siguientes métricas para cada arista:

- Ecuación de la recta: Pendiente e intercepto obtenidos a partir del ajuste lineal.
- Error cuadrático medio entre los puntos reales y la línea ajustada.
- Intervalos de confianza (IC): Para la pendiente e intercepto, se calcularon intervalos de confianza del 95 %, proporcionando una medida de la incertidumbre en la estimación de los parámetros de la recta.
- Diferencia promedio. Media de las diferencias entre los puntos detectados y el valor real esperado, proporcionando una medida del error sistemático del sensor.
- Puntos máximo y mínimo: Los valores extremos detectados en cada conjunto de puntos que indican la variabilidad en las mediciones.
- Rango: Diferencia entre el valor máximo y mínimo., lo que refleja la dispersión de las mediciones.

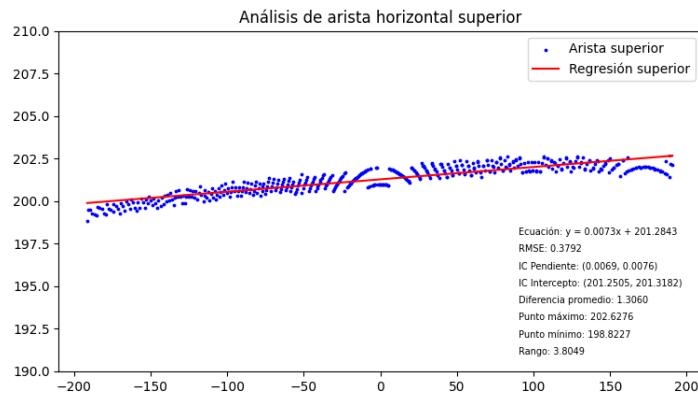
Para evaluar la rectitud de las aristas, se analizaron las pendientes obtenidas a partir del ajuste lineal. Estas pendientes, muestran pequeñas desviaciones respecto a la horizontalidad o verticalidad ideal, lo que sugiere un ligero error sistemático en la orientación del sensor.

Los intervalos de confianza calculados para estas presentan un rango estrecho, lo que indica una alta consistencia en las mediciones y un bajo nivel de variabilidad entre los diferentes puntos medidos. Adicionalmente, las magnitudes de las pendientes están cercanas a cero, lo que sugiere que las aristas reconstruidas se aproximan de manera precisa a la geometría esperada (líneas rectas definidas).

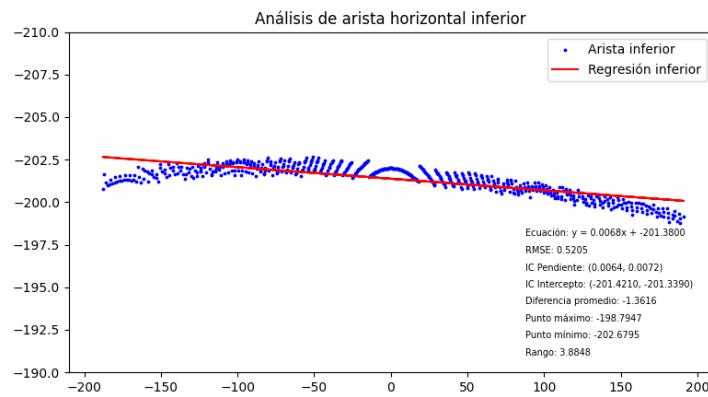
El intercepto calculado resulta útil para evaluar el alineamiento general del sensor con respecto a las posiciones esperadas. En todos los casos, se observan ligeras desviaciones en el intercepto que oscilan entre 0.3959 mm y 2.4565 mm con respecto a las dimensiones esperadas de ± 200 mm, lo cual es relativamente bajo. Sin embargo, estas pequeñas variaciones evidencian la presencia de imprecisiones en las mediciones del sensor.

El análisis de los intervalos de confianza de la pendiente y el intercepto proporciona una medida de la fiabilidad de estos parámetros. Al mostrar rangos estrechos, se sugiere una fuerte consistencia en las mediciones, lo cual refuerza la confiabilidad de las regresiones lineales obtenidas. Junto a esto, el rango de las mediciones, que varía entre 3.8049 mm y 4.4322 mm para las diferentes aristas, refleja una dispersión controlada. La diferencia promedio entre los puntos detectados y el valor real esperado, inferior a 3 mm, confirma que la dispersión de las mediciones se encuentra dentro de los límites de precisión indicados por el fabricante.

En resumen, las métricas analizadas muestran que el sensor presenta una buena precisión general. A pesar que se observen pequeñas desviaciones, los errores están por debajo de los 5 mm, lo cual es aceptable para la mayoría de las aplicaciones, en especial para navegación autónoma.



(a) Análisis de arista horizontal superior



(b) Análisis de arista horizontal inferior

Figura 30: Análisis de la reconstrucción realizada para las aristas horizontales de una caja de 400×400 mm: tres revoluciones completas capturadas.

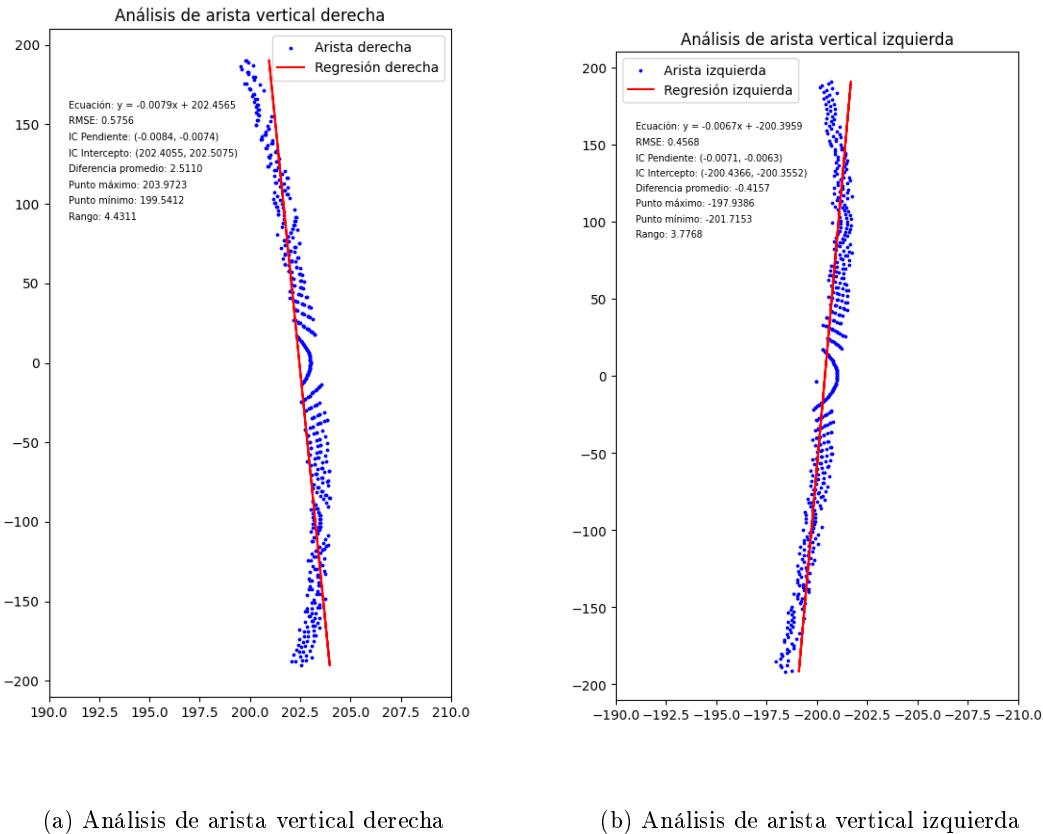


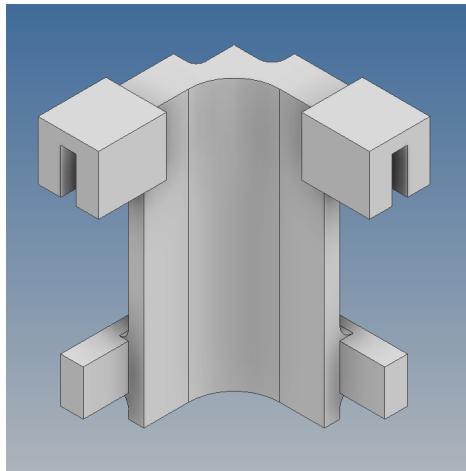
Figura 31: Análisis de la reconstrucción realizada para las aristas verticales de una caja de 400×400 mm: tres revoluciones completas capturadas.

9.7. Detección de bordes y discontinuidades para el LiDAR FHL-LD20

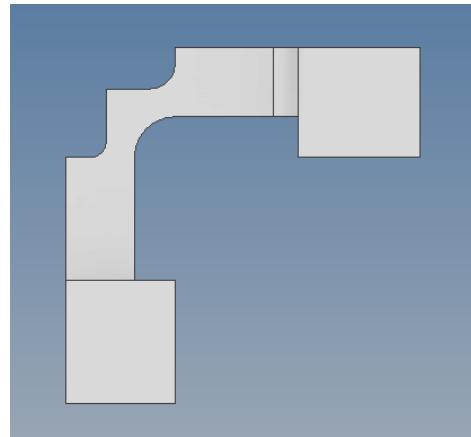
A lo largo de los experimentos realizados se observó que, al probar con entornos cuadriláteros, las esquinas tienden a perderse en las mediciones. Esto sugiere que el sensor podría enfrentar dificultades para detectar con precisión bordes y discontinuidades en estas áreas específicas. Esto se debe a que las esquinas suele generar geometrías complejas y abruptas, lo que genera retornos dispersos e inconsistentes al interactuar con el haz infrarrojo.

Con el fin de evaluar cómo afectan las esquinas la precisión en las mediciones de los bordes, se diseñaron codos con radios de 5, 10 y 15 mm (Figura 32). Estos codos permiten estudiar el comportamiento del sensor cuando frente a transiciones suaves entre superficies, simulando una curvatura en lugar de un ángulo agudo. Al utilizar codos en lugar de esquinas afiladas, es posible analizar si el sensor puede detectar transiciones graduales en los bordes con mayor precisión, lo que reduce las distorsiones capturadas. Además, los codos proporcionan un medio de comparación para evaluar el rendimiento del sensor al medir bordes con radios de curvatura específicos, lo que permite establecer un rango de tolerancia en el cual las

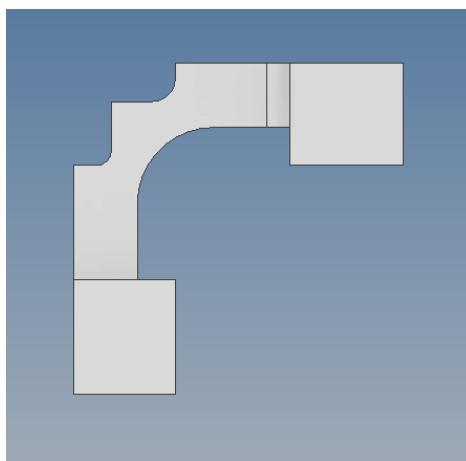
mediciones del sensor sean confiables.



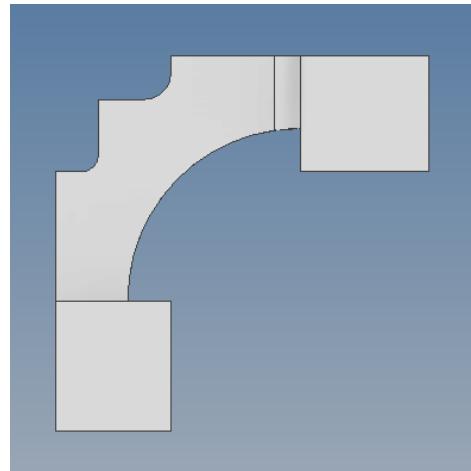
(a) Vista isométrica del diseño del codo para transiciones graduales.



(b) Vista superior del diseño del codo de 5 mm de radio.



(c) Vista superior del diseño del codo de 10 mm de radio.



(d) Vista superior del diseño del codo de 15 mm de radio.

Figura 32: Diseño de los codos para transiciones graduales

CAPÍTULO 10

Calibración del LiDAR FHL-LD20

10.1. Desarrollo de plataforma para calibrar el LiDAR FHL-LD20

Para realizar la calibración del sensor, se desarrolló una plataforma que garantiza la precisión en las mediciones del sensor. La estructura principal consiste en una caja de 400×400 mm, equipada con una alfombra cuadriculada en milímetros y cuatro líneas diagonales a 45 grados, las cuales funcionan de referencia visual para la orientación. Además, se diseñó un soporte para el sensor, que incluye dos geometrías dispuestas radialmente a 45 grados y a una distancia de 3 mm del centro del sensor, simulando una brújula. Estas guías se alinean con la cuadrícula de la alfombra, lo que asegura una correcta orientación del sensor dentro de la caja y facilita el proceso de calibración.

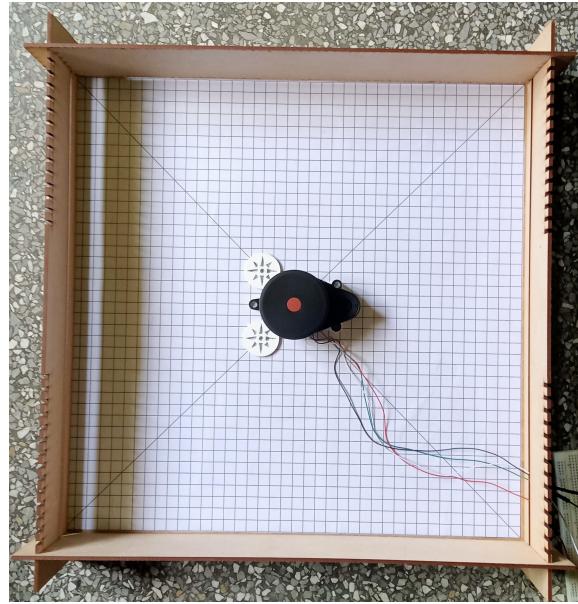


Figura 33: Vista superior de la plataforma para calibración.

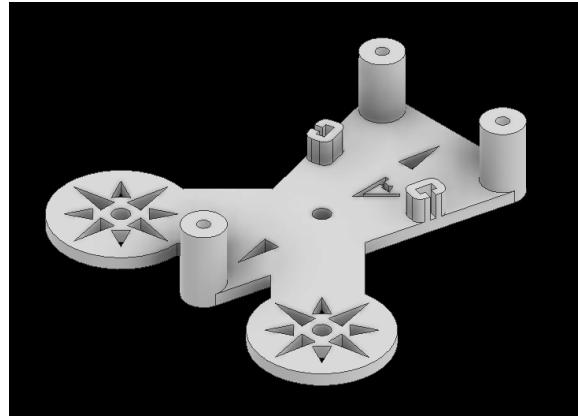


Figura 34: Vista isométrica del soporte para el LiDAR FHL-LD20.

10.2. Pruebas para calibrar el LiDAR FHL-LD20

Para verificar la calibración del sensor, se colocó en el centro de la plataforma utilizando las brújulas incluidas en el soporte para asegurar su alineación. Una vez posicionado, se realizó una captura de tres revoluciones completas y se compararon los valores medidos de las aristas con los valores nominales esperados. Si los resultados obtenidos se encuentran dentro de una desviación igual o menor a la precisión especificada por el radio de medición, se considera que el sensor está calibrado. En este experimento, se obtuvieron los siguientes mediciones de las aristas: 201.28 mm para la arista horizontal superior, -201.38 mm para la inferior, 202.45 mm para la vertical derecha, y -200.39 mm para la izquierda. Estas desviaciones, aunque mínimas, se encuentran dentro del rango de precisión del sensor, lo que confirma que el dispositivo está correctamente calibrado.

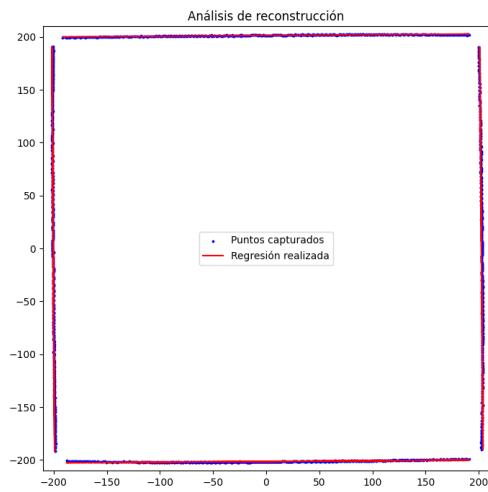


Figura 35: Reconstrucción realizada para verificar la calibración del sensor.

CAPÍTULO 11

Conclusiones

CAPÍTULO 12

Recomendaciones

CAPÍTULO 13

Bibliografía

- [1] J. Li, W. Tang, D. Zhang, D. Fan, J. Jiang e Y. Lu, “Map Construction and Path Planning Method for Mobile Robots Based on Collision Probability Model,” *Symmetry*, vol. 15, n.º 10, pág. 1891, oct. de 2023, ISSN: 2073-8994. DOI: 10.3390/sym15101891. dirección: <https://www.mdpi.com/2073-8994/15/10/1891>.
- [2] Z. Li, B. Li, Q. Liang, W. Liu, L. Hou y X. Rong, “A quadruped robot obstacle avoidance and personnel following strategy based on ultra-wideband and three-dimensional laser radar,” *International Journal of Advanced Robotic Systems*, vol. 19, n.º 4, jul. de 2022, ISSN: 1729-8806. DOI: 10.1177/17298806221114705. dirección: <https://doi.org/10.1177/17298806221114705>.
- [3] X. Dai, S. Long, Z. Zhang y D. Gong, “Mobile Robot Path Planning Based on Ant Colony Algorithm With A* Heuristic Method,” *Frontiers in Neurorobotics*, vol. 13, abr. de 2019, ISSN: 1662-5218. DOI: 10.3389/fnbot.2019.00015. dirección: <https://www.frontiersin.org/articles/10.3389/fnbot.2019.00015>.
- [4] Y. Han, L. Zhang, H. Tan y X. Xue, “Mobile Robot Path Planning Based on Improved Particle Swarm optimization,” en *2019 Chinese Control Conference (CCC)*, ISSN: 1934-1768, jul. de 2019, págs. 4354-4358. DOI: 10.23919/ChiCC.2019.8866634. dirección: <https://ieeexplore.ieee.org/document/8866634>.
- [5] A. Peña, “Algoritmo de sincronización y control de sistemas de robots multiagente para misiones de búsqueda,” Tesis de licenciatura, Universidad del Valle de Guatemala, 2019.
- [6] E. Santizo, “Aprendizaje reforzado y aprendizaje profundo en aplicaciones de robótica de enjambre,” Tesis de licenciatura, Universidad del Valle de Guatemala, 2021.
- [7] D. Baldizón, “Aplicaciones prácticas para algoritmos de inteligencia y robótica de enjambre,” Tesis de licenciatura, Universidad del Valle de Guatemala, 2022.
- [8] F. Godoy, “Desarrollo y evaluación de algoritmos para el mapeo de entornos y generación de trayectorias utilizando sistemas robóticos multi-agente,” Tesis de licenciatura, Universidad del Valle de Guatemala, 2023.

- [9] S. Thrun, W. Burgard y D. Fox, *Probabilistic robotics* (Intelligent robotics and autonomous agents). Cambridge, Massachusetts: MIT Press, 2005, ISBN: 978-0-262-20162-9.
- [10] F. Fahimi, *Autonomous Robots: Modeling, Path Planning, and Control*. Edmonton, Alberta: Springer US, 2009. DOI: 10.1007/978-0-387-09538-7.
- [11] B. Siciliano, L. Sciavicco, L. Villani y G. Oriolo, *Robotics Modelling, Planning and Control* (Advanced Textbooks in Control and Signal Processing). London, England: Springer London, 2009, ISBN: 978-1-84628-641-4 978-1-84628-642-1. DOI: 10.1007/978-1-84628-642-1.
- [12] H. Lu, S. Yang, M. Zhao y S. Cheng, *Multi-Robot Indoor Environment Map Building Based on Multi-Stage Optimization Method*. 2021, vol. 1, págs. 145-161. DOI: 10.23919/CSMS.2021.0011.
- [13] N. Correll, *Introduction to Autonomous Robots* (Kinematics, Perception, Localization and Planning). Independent, 2016, ISBN: 978-1-493-77307-7.
- [14] P. Corke, *Robotics, Vision and Control* (Springer Tracts in Advanced Robotics), Second Edition. Cham, Switzerland: Springer International Publishing, 2017, vol. 118, ISBN: 978-3-319-54412-0 978-3-319-54413-7. DOI: 10.1007/978-3-319-54413-7.
- [15] J. Liu, Q. Sun, Z. Fan e Y. Jia, “TOF Lidar Development in Autonomous Vehicle,” 2018. DOI: 10.1109/OGC.2018.8529992.
- [16] STMicroelectronics *Time of Flight (ToF) Sensors*, 2022. dirección: <https://www.st.com/en/imaging-and-photonics-solutions/time-of-flight-sensors.html>.
- [17] Pololu - 3pi+ 32U4 Robot - Standard Edition, 2024. dirección: <https://www.pololu.com/product/3737>.

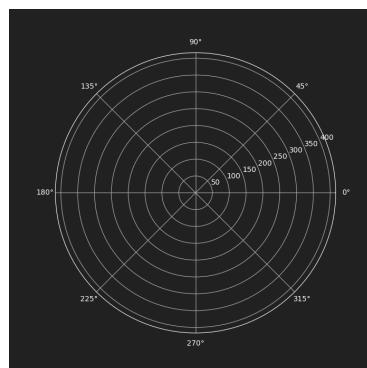
CAPÍTULO 14

Anexos

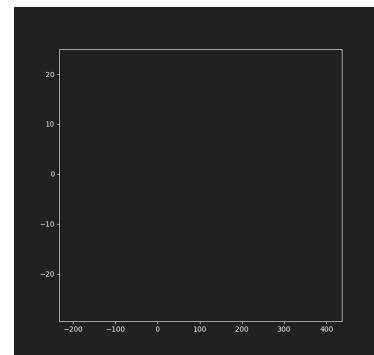
14.1. Pruebas de rango de medición mínimo



(a) Disposición del sensor FHL-LD20 dentro del entorno circular:
7 cm de radio



(b) Reconstrucción en coordenadas polares de
tres revoluciones completas de lectura en entorno
circular: 7 cm de radio

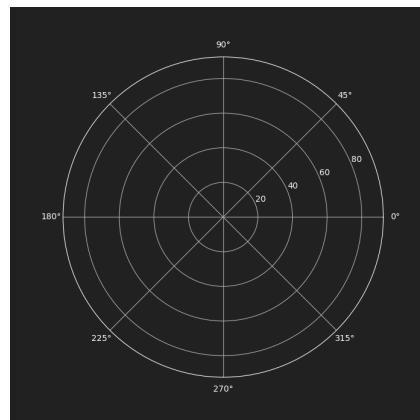


(c) Reconstrucción en coordenadas cartesianas de tres revoluciones completas de lectura en entorno circular: 7 cm de radio

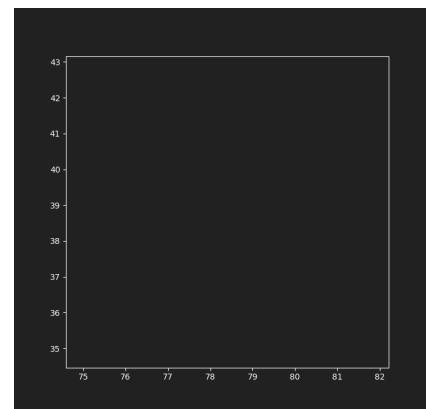
Figura 36: Reconstrucciones vacías de tres revoluciones completas de lectura en entorno circular: 7 cm de radio



(a) Disposición del sensor F HL-LD20 dentro del entorno circular:
8 cm de radio



(b) Reconstrucción en coordenadas polares de
tres revoluciones completas de lectura en entorno
circular: 8 cm de radio

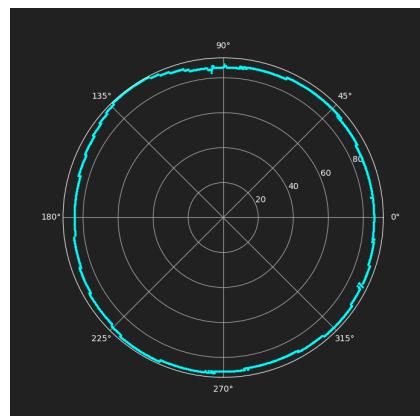


(c) Reconstrucción en coordenadas cartesianas de tres revoluciones completas de lectura en entorno circular: 8 cm de radio

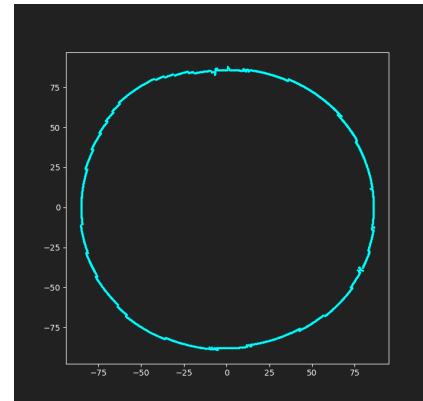
Figura 37: Reconstrucciones vacías de tres revoluciones completas de lectura en entorno circular: 8 cm de radio



(a) Disposición del sensor FHL-LD20 dentro del entorno circular:
9 cm de radio



(b) Reconstrucción en coordenadas polares de
tres revoluciones completas de lectura en entorno
circular: 9 cm de radio



(c) Reconstrucción en coordenadas cartesianas
de tres revoluciones completas de lectura en en-
torno circular: 9 cm de radio

Figura 38: Reconstrucciones vacías de tres revoluciones completas de lectura en entorno circular: 9 cm de radio

