



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

# **Robot Móvil Diferencial Controlado con Campos Potenciales**

**TESIS**

Que para obtener el título de  
**Ingeniero Mecatrónico**

**P R E S E N T A**

Jorge Fidel Martínez Ramírez

**DIRECTOR DE TESIS**

M. I. Ulises Martín Peñuelas Rivas



**Ciudad Universitaria, Cd. Mx., 2026**



**PROTESTA UNIVERSITARIA DE INTEGRIDAD Y  
HONESTIDAD ACADÉMICA Y PROFESIONAL  
(Titulación con trabajo escrito)**



De conformidad con lo dispuesto en los artículos 87, fracción V, del Estatuto General, 68, primer párrafo, del Reglamento General de Estudios Universitarios y 26, fracción I, y 35 del Reglamento General de Exámenes, me comprometo en todo tiempo a honrar a la institución y a cumplir con los principios establecidos en el Código de Ética de la Universidad Nacional Autónoma de México, especialmente con los de integridad y honestidad académica.

De acuerdo con lo anterior, manifiesto que el trabajo escrito titulado ROBOT MOVIL DIFERENCIAL CONTROLADO CON CAMPOS POTENCIALES que presenté para obtener el título de INGENIERO MECATRÓNICO es original, de mi autoría y lo realicé con el rigor metodológico exigido por mi Entidad Académica, citando las fuentes de ideas, textos, imágenes, gráficos u otro tipo de obras empleadas para su desarrollo.

En consecuencia, acepto que la falta de cumplimiento de las disposiciones reglamentarias y normativas de la Universidad, en particular las ya referidas en el Código de Ética, llevará a la nulidad de los actos de carácter académico administrativo del proceso de titulación.



---

**JORGE FIDEL MARTÍNEZ RAMÍREZ**  
Número de cuenta: 318090117

## **DEDICATORIA**

*Para mi padre, que nunca se rindió conmigo. Para mi madre, que siempre vio lo mejor de mí.  
Y para todos los maestros que me guiaron durante este viaje.*



## **AGRADECIMIENTOS**

Expreso mi especial reconocimiento y agradecimiento a la Dirección General de Asuntos del Personal Académico (DGAPA) que, mediante el Programa de Apoyo a Proyectos para Innovar y Mejorar la Educación (PAPIME), brindó el financiamiento necesario para la realización de este trabajo bajo el proyecto PE112124: 'Desarrollo de recursos educativos innovadores con el SoC ESP32'.

Asimismo, expreso mi sincero agradecimiento al Dr. Jesús Savage Carmona por su invaluable apoyo y asesoría; a mi asesor, el M. I. Ulises Peñuelas, por su guía y orientación constante durante este proyecto; y a los miembros del jurado examinador por sus valiosas observaciones y retroalimentación. Finalmente, extendiendo mi gratitud a todas las personas que, de una u otra forma, hicieron posible la culminación de este trabajo.



## CONTENIDO

|                                                                |    |
|----------------------------------------------------------------|----|
| RESUMEN .....                                                  | 1  |
| CAPÍTULO 1 INTRODUCCIÓN.....                                   | 3  |
| 1.1 ANTECEDENTES HISTÓRICOS .....                              | 3  |
| 1.2 JUSTIFICACIÓN DEL PROYECTO.....                            | 4  |
| 1.3 OBJETIVOS.....                                             | 5  |
| 1.4 ORGANIZACIÓN DEL DOCUMENTO .....                           | 5  |
| CAPÍTULO 2 PROBLEMÁTICA .....                                  | 7  |
| CAPÍTULO 3 ESTADO DE LA TÉCNICA .....                          | 9  |
| 3.1 CAMPOS POTENCIALES ARTIFICIALES EN ROBÓTICA .....          | 9  |
| 3.2 LIMITACIONES DEL MÉTODO Y SOLUCIONES HÍBRIDAS.....         | 10 |
| 3.3 ALGORITMO DE DIJKSTRA EN NAVEGACIÓN AUTÓNOMA .....         | 11 |
| 3.4 APLICACIONES ACTUALES DE CAMPOS POTENCIALES.....           | 12 |
| 3.5 ALTERNATIVAS DE NAVEGACIÓN .....                           | 14 |
| CAPÍTULO 4 METODOLOGÍA.....                                    | 15 |
| 4.1 LOCALIZACIÓN DEL ROBOT POR TRILATERACIÓN.....              | 15 |
| 4.2 DETECCIÓN Y SEGMENTACIÓN DE OBSTÁCULOS .....               | 19 |
| 4.2.1 Segmentación por agrupamiento de mediciones .....        | 19 |
| 4.2.2 Cálculo del centroide de cada obstáculo .....            | 20 |
| 4.2.3 Transformación a un sistema de referencia del robot..... | 20 |
| 4.3 TRANSFORMACIÓN DE COORDENADAS AL SISTEMA MÓVIL .....       | 21 |
| 4.4 CAMPO POTENCIAL VIRTUAL .....                              | 23 |
| 4.4.1 Función atractiva por el objetivo.....                   | 23 |

|                                                        |                                                              |    |
|--------------------------------------------------------|--------------------------------------------------------------|----|
| 4.4.2                                                  | Función repulsiva por obstáculos .....                       | 26 |
| 4.5                                                    | OBTENCIÓN DEL VECTOR DE FUERZA RESULTANTE.....               | 29 |
| 4.6                                                    | RESUMEN DEL CICLO DE NAVEGACIÓN.....                         | 30 |
| CAPÍTULO 5 MODELADO CINEMÁTICO Y CONTROL DIGITAL ..... |                                                              | 31 |
| 5.1                                                    | MODELO CINEMÁTICO BÁSICO DEL ROBOT DIFERENCIAL.....          | 32 |
| 5.1.1                                                  | Hipótesis y sistema de referencia .....                      | 32 |
| 5.1.2                                                  | Campo de velocidades de un cuerpo rígido .....               | 33 |
| 5.1.3                                                  | Relación con las velocidades angulares de la rueda.....      | 33 |
| 5.1.4                                                  | Observaciones prácticas .....                                | 35 |
| 5.2                                                    | GENERACIÓN DE COMANDOS DE RAPIDEZ.....                       | 36 |
| 5.2.1                                                  | Vector de referencia a partir de campos potenciales .....    | 36 |
| 5.2.2                                                  | Descomposición en velocidad lineal y angular .....           | 36 |
| 5.2.3                                                  | Ley de control proporcional .....                            | 37 |
| 5.2.4                                                  | Saturación de velocidades .....                              | 37 |
| 5.2.5                                                  | Ventajas del enfoque reactivo.....                           | 38 |
| 5.3                                                    | CONTROL DIGITAL .....                                        | 38 |
| 5.3.1                                                  | Discretización bilineal.....                                 | 39 |
| 5.3.2                                                  | Obtención de la ecuación en diferencias .....                | 41 |
| 5.4                                                    | ESTIMACIÓN DE LA POSE MEDIANTE EL FILTRO KALMAN .....        | 42 |
| 5.4.1                                                  | Definición del vector de estado y modelo del proceso.....    | 43 |
| 5.4.2                                                  | Etapas de predicción y linealización del modelo.....         | 44 |
| 5.4.3                                                  | Etapas de actualización con mediciones de trilateración..... | 45 |
| 5.4.4                                                  | Etapas de actualización de la orientación con IMU.....       | 46 |
| 5.5                                                    | RESUMEN DEL MODELO CINEMÁTICO Y CONTROL.....                 | 47 |
| CAPÍTULO 6 DESARROLLO DEL PROTOTIPO FÍSICO .....       |                                                              | 49 |
| 6.1                                                    | DESCRIPCIÓN GENERAL DEL ROBOT .....                          | 49 |
| 6.2                                                    | ARQUITECTURA FUNCIONAL DEL SISTEMA .....                     | 50 |
| 6.3                                                    | SELECCIÓN DE SENSORES.....                                   | 51 |
| 6.3.1                                                  | Sensor de localización.....                                  | 52 |
| 6.3.2                                                  | Sensor de orientación .....                                  | 53 |
| 6.3.3                                                  | Sensores de obstáculos .....                                 | 54 |
| 6.4                                                    | ACTUADORES .....                                             | 56 |
| 6.4.1                                                  | Identificación del modelo dinámico .....                     | 56 |

---

|                              |                                                                         |    |
|------------------------------|-------------------------------------------------------------------------|----|
| 6.4.2                        | Resultados de la identificación.....                                    | 57 |
| 6.4.3                        | Diseño del controlador de rapidez.....                                  | 59 |
| 6.5                          | CODIFICADORES ROTATORIOS .....                                          | 60 |
| 6.5.1                        | Problemas experimentales y mitigación de ruido .....                    | 61 |
| 6.5.2                        | Error sistemático por distribución no uniforme de imanes .....          | 61 |
| 6.5.3                        | Medición basada en intervalos temporales.....                           | 62 |
| 6.5.4                        | Calibración y generación de la tabla de corrección.....                 | 62 |
| 6.5.5                        | Corrección en tiempo real y cálculo final de velocidad .....            | 63 |
| 6.5.6                        | Persistencia de la calibración .....                                    | 63 |
| 6.6                          | SELECCIÓN DEL MICROCONTROLADOR.....                                     | 64 |
| 6.7                          | PROGRAMACIÓN DEL SISTEMA DE NAVEGACIÓN.....                             | 64 |
| CAPÍTULO 7 RESULTADOS.....   |                                                                         | 65 |
| 7.1                          | VALIDACIÓN DEL SISTEMA EMBEBIDO Y CONTROL.....                          | 65 |
| 7.1.1                        | Corrección de errores sistemáticos en encoders.....                     | 66 |
| 7.1.2                        | Análisis de la respuesta temporal y desempeño del controlador PID ..... | 67 |
| 7.1.3                        | Caracterización del sistema de localización absoluta .....              | 69 |
| 7.1.4                        | Fusión sensorial mediante filtro de Kalman extendido .....              | 72 |
| 7.2                          | VALIDACIÓN DEL ALGORITMO DE NAVEGACIÓN .....                            | 77 |
| 7.2.1                        | Condiciones iniciales y parámetros de ensayo .....                      | 78 |
| 7.2.2                        | Evolución de la trayectoria y respuesta del campo potencial.....        | 78 |
| 7.2.3                        | Análisis cuantitativo del desempeño .....                               | 80 |
| 7.2.4                        | Resumen del desempeño .....                                             | 80 |
| CAPÍTULO 8 CONCLUSIONES..... |                                                                         | 81 |
| 8.1                          | APORTACIONES DEL TRABAJO .....                                          | 81 |
| 8.2                          | CUMPLIMIENTO DE OBJETIVOS.....                                          | 82 |
| 8.3                          | LIMITACIONES DEL SISTEMA .....                                          | 83 |
| 8.4                          | TRABAJO FUTURO.....                                                     | 84 |
| 8.5                          | REFLEXIÓN FINAL.....                                                    | 85 |
| REFERENCIAS .....            |                                                                         | 87 |



## RESUMEN

En este trabajo de tesis se presenta el diseño, implementación y validación experimental de un sistema de navegación autónoma para un robot móvil en configuración diferencial, basado en la técnica reactiva de campos potenciales artificiales. El objetivo principal es demostrar la viabilidad de esta estrategia clásica de control en un prototipo físico real, enfrentando y resolviendo las no linealidades, el ruido y las limitaciones inherentes al hardware de bajo costo.

La metodología integra percepción, localización, planificación y control en una arquitectura de dos niveles. En el nivel alto, se ejecuta sobre un sistema operativo, un programa planificador que calcula en tiempo real un campo potencial virtual, fruto de la superposición de un gradiente atractivo hacia el destino y gradientes repulsivos generados por los obstáculos detectados, enriquecidos con una componente tangencial para la evasión. En el nivel bajo, un sistema embebido se encarga del control de rapidez de los motores mediante un control digital, cerrando el lazo con la información de codificadores rotativos magnéticos. Para obtener una estimación robusta y continua de la pose del robot, se implementa un filtro de Kalman extendido que fusiona la odometría, la unidad de medición inercial y un sistema de localización absoluta por radiofrecuencia.

La validación experimental evidenció la solidez del sistema. La caracterización y corrección de los codificadores rotativos mediante una tabla de consulta permitió reducir la varianza del error de velocidad angular del 13.5% a menos del 1%, proporcionando una señal limpia al controlador. El filtro de Kalman demostró su capacidad para suavizar la trayectoria al rechazar mediciones

atípicas de los sensores y mantener un error de orientación estabilizado en  $\pm 0.6^\circ$ . Finalmente, en pruebas de navegación en un entorno controlado, el robot recorrió exitosamente una distancia de 5 metros en 24.8 segundos, reaccionando ante obstáculos en menos de 50 milisegundos y sin registrar colisiones. Los resultados confirmaron que la sinergia entre un modelado matemático riguroso y un filtrado sensorial adecuado permite construir plataformas robóticas funcionales y de bajo costo, estableciendo una base sólida para futuras investigaciones en navegación híbrida.

## CAPÍTULO 1

### INTRODUCCIÓN

La robótica móvil integra disciplinas como la mecánica, la electrónica y la computación para el desarrollo de sistemas capaces de desplazarse de manera autónoma en entornos con diversos grados de complejidad. Dentro de este ámbito, la navegación reactiva constituye un pilar fundamental para garantizar desplazamientos seguros sin requerir un procesamiento computacional excesivo en el vehículo. En este primer capítulo se abordan las nociones generales que contextualizaron el desarrollo de la investigación, abarcando desde la evolución histórica de la robótica autónoma hasta la definición del problema central que motivó este trabajo. Asimismo, se presenta la sustentación técnica de la plataforma experimental construida y se explican formalmente los objetivos que guiaron la implementación y validación del sistema de navegación reactivo. Finalmente, se incluye el desglose de la estructura general del documento para facilitar la comprensión de cada una de las fases que componen este estudio.

#### 1.1 ANTECEDENTES HISTÓRICOS

La robótica es una disciplina de la ingeniería que integra mecánica, electrónica, informática y control automático, para diseñar y construir sistemas capaces de interactuar con su entorno físico. Según la norma ISO 8373:2012, un robot es un “*mecanismo accionado y programado con un grado de autonomía para realizar tareas de locomoción, manipulación o*

*posicionamiento*” [1]. Esta definición proporciona una base amplia que abarca tanto robots industriales como móviles, colaborativos y de servicio.

Los orígenes del concepto de máquinas inteligentes pueden rastrearse desde mitologías antiguas. En culturas como la griega, china o india se relatan historias en las que seres artificiales actuaban bajo el control de su creador. Ya en el siglo IV A. C., Arquitas de Tarento diseñó el primer modelo de pájaro mecánico, y en el Renacimiento Leonardo da Vinci propuso un autómatas humanoide. En el siglo XVIII, inventores como Jacques de Vaucanson construyeron autómatas que podían tocar música o simular funciones biológicas.

En las últimas décadas, los robots han evolucionado desde sistemas rígidos y preprogramados hasta plataformas inteligentes, capaces de percibir su entorno y tomar decisiones en tiempo real. Esta evolución ha sido clave para el desarrollo de los sistemas de navegación autónoma, los cuales permiten que un robot se desplace sin intervención humana directa.

## **1.2 JUSTIFICACIÓN DEL PROYECTO**

Uno de los principales desafíos en robótica móvil es lograr que un robot se desplace de forma segura y eficiente en entornos reales, especialmente cuando hay obstáculos y condiciones dinámicas. Esta capacidad no solo implica moverse del punto A al punto B, sino hacerlo evitando colisiones, optimizando la trayectoria y respetando restricciones físicas y ambientales.

Una técnica comúnmente utilizada para la navegación reactiva es el método de campos potenciales, que permite modelar el entorno mediante fuerzas virtuales atractivas y repulsivas. Sin embargo, este enfoque presenta limitaciones, como quedar atrapado en mínimos locales. Para resolverlo, en la literatura se han propuesto diferentes metodologías que abordan dicha problemática a su manera.

Esta tesis propuso desarrollar un sistema de navegación autónoma para un prototipo físico, utilizando el algoritmo de campos potenciales. La solución se validó en un entorno controlado, con el objetivo de demostrar su viabilidad práctica y su capacidad de evitar obstáculos en tiempo real de manera eficiente.

### 1.3 OBJETIVOS

El objetivo general de este trabajo fue implementar un sistema de control basado en campos potenciales para la navegación autónoma de un prototipo físico, que permita su desplazamiento en entornos semicontrolados con obstáculos mediante la identificación de rutas seguras y la prevención de colisiones. Para alcanzarlo, se plantearon los siguientes objetivos específicos:

- Caracterizar la respuesta dinámica de los motores del prototipo para asegurar un control de movimiento.
- Seleccionar e integrar los sensores requeridos para la detección de obstáculos en tiempo real dentro del entorno de trabajo.
- Implementar un sistema de estimación de pose del robot, que es su posición y orientación, en el espacio de trabajo.
- Desarrollar el algoritmo de control que integre la navegación hacia las coordenadas del destino.
- Validar el desempeño del sistema de control integrado en el prototipo físico dentro de un entorno de pruebas delimitado.

### 1.4 ORGANIZACIÓN DEL DOCUMENTO

El presente trabajo de tesis se estructura en ocho capítulos, organizados para guiar al lector desde el planteamiento conceptual del problema hasta la validación experimental del prototipo físico. La estructura es la siguiente:

- **CAPÍTULO 1 INTRODUCCIÓN.** Establece el contexto de la robótica móvil, los antecedentes históricos, la justificación del proyecto y los objetivos generales y específicos que guían esta investigación.
- **CAPÍTULO 2 PROBLEMÁTICA.** Define formalmente el desafío de la navegación autónoma en entornos reales. Se analizan las dificultades inherentes a la percepción, localización y planeación.

- **CAPÍTULO 3 ESTADO DE LA TÉCNICA.** Presenta una revisión de la literatura sobre métodos de navegación. Se profundiza en el origen físico de los campos potenciales artificiales, sus limitaciones y las soluciones de vanguardia.
- **CAPÍTULO 4 METODOLOGÍA.** Desarrolla el marco matemático del sistema. Se detalla el modelo de localización por trilateración, el algoritmo de segmentación de obstáculos, y la formulación de los campos potenciales y su vector resultante.
- **CAPÍTULO 5 MODELADO CINEMÁTICO Y CONTROL DIGITAL.** Traduce la teoría a la mecánica del robot. Se presenta el modelo cinemático diferencial, la estrategia de generación de comandos de velocidad a partir del gradiente del potencial, y el diseño del controlador digital. El capítulo culmina con la formulación matemática del filtro de Kalman extendido para la fusión sensorial.
- **CAPÍTULO 6 DESARROLLO DEL PROTOTIPO FÍSICO.** Documenta la implementación práctica e ingenieril del sistema. Describe la arquitectura funcional dividida entre el sistema operativo de alto nivel y el microcontrolador de bajo nivel, la selección y caracterización de sensores, la identificación del modelo dinámico de los motores y el proceso de calibración, con la corrección de errores sistemáticos en los codificadores rotativos magnéticos mediante una tabla de consulta.
- **CAPÍTULO 7 RESULTADOS.** Presenta y analiza rigurosamente el desempeño del sistema integrado. Se valida experimentalmente la corrección de la tabla de consulta para los codificadores rotativos, la respuesta temporal del controlador digital, la precisión del sistema de trilateración y la capacidad del filtro de Kalman para suavizar la trayectoria y rechazar valores atípicos. Finalmente, se evalúa el algoritmo de navegación por campos potenciales en un escenario real, reportando métricas de desempeño como tiempo de reacción, distancia de evasión y tasa de éxito.
- **CAPÍTULO 8 CONCLUSIONES.** Sintetiza las aportaciones técnicas del trabajo, contrasta los resultados obtenidos con los objetivos planteados, identifica las limitaciones operativas del sistema actual y propone líneas de trabajo futuro orientadas a la optimización paramétrica y la mejora de la robustez en entornos complejos.

## **CAPÍTULO 2**

### **PROBLEMÁTICA**

En las últimas décadas, la automatización ha transformado profundamente sectores como la industria, la medicina y la agricultura. En este contexto, los robots móviles han cobrado una relevancia creciente al permitir la ejecución autónoma de tareas como inspección, transporte o exploración. Sin embargo, para que estos sistemas puedan operar de manera confiable en entornos reales, es necesario dotarlos de capacidades avanzadas de percepción, localización y navegación.

Una de las habilidades más desafiantes en este ámbito es la navegación autónoma, es decir, la capacidad de un robot para desplazarse por sí mismo desde una posición inicial hacia un objetivo determinado, evitando colisiones y respetando las restricciones del entorno. Dicho reto exige determinar con precisión la ubicación del robot en el espacio y responder en tiempo real ante la presencia de obstáculos, garantizando así un desplazamiento controlado hacia el punto de destino.

El desarrollo de algoritmos de navegación robustos enfrenta retos técnicos de alta complejidad. En primer lugar, los entornos reales demandan la evasión tanto de obstáculos estáticos, cuya ubicación se mantiene constante en el entorno de trabajo, como de obstáculos dinámicos, los cuales introducen elementos impredecibles en tiempo real. En segundo lugar, las imprecisiones de las lecturas sensoriales, el desajuste en el modelo del sistema y las perturbaciones del entorno comprometen el desempeño del sistema de control durante la ejecución de las trayectorias.

Una estrategia ampliamente utilizada para la navegación reactiva es el uso de campos potenciales artificiales, los cuales modelan el entorno como un campo de fuerzas donde el objetivo ejerce una atracción y los obstáculos generan repulsiones, generando una suerte de camino para el robot. Este método es simple, computacionalmente eficiente y permite una respuesta en tiempo real, lo que lo hace atractivo para plataformas con recursos limitados. Sin embargo, presenta una limitación conocida: la posibilidad de quedar atrapado en mínimos locales, es decir, puntos donde el gradiente de potencial es nulo pero que no coincide con la meta. A pesar de ello, en entornos poco congestionados o debidamente parametrizados, el método puede operar de manera satisfactoria.

En esta tesis se implementó y validó experimentalmente un sistema de navegación basado en dichos campos, aplicado a un prototipo físico de robot en configuración diferencial. El objetivo fue demostrar que esta técnica reactiva, a pesar de sus limitaciones teóricas, es viable en la práctica cuando se complementa con una adecuada percepción sensorial y un control de bajo nivel robusto. Por ello, el presente trabajo profundizó en el desempeño del campo potencial dentro de un entorno controlado con obstáculos dinámicos, abriendo la vía para integrar en investigaciones futuras algoritmos de planeación global como Dijkstra o A estrella, o cualquier otro algoritmo que mejore la robustez del sistema.

La implementación física del sistema implica resolver problemas prácticos fundamentales, entre los que destacan:

- La localización del robot en tiempo real dentro del espacio de navegación
- La detección de obstáculos mediante sensores de distancia
- La caracterización de los actuadores para garantizar un movimiento controlado
- La integración de la lógica de control en un sistema embebido.

Este trabajo buscó demostrar que es posible aplicar técnicas clásicas de control reactivo en un robot físico construido con hardware accesible, enfrentando y resolviendo las no linealidades, el ruido y las limitaciones propias de los sensores y actuadores reales.

## **CAPÍTULO 3**

### **ESTADO DE LA TÉCNICA**

La evolución de los sistemas de navegación autónoma ha estado motivada por la necesidad de que los robots móviles puedan desplazarse en entornos complejos sin intervención humana. Entre las múltiples estrategias desarrolladas, destacan los campos potenciales artificiales por su simplicidad, eficiencia y analogía con fenómenos físicos. En este capítulo se explora el origen de esta técnica, su evolución en la robótica móvil y otras estrategias complementarias o alternativas.

#### **3.1 CAMPOS POTENCIALES ARTIFICIALES EN ROBÓTICA**

Los métodos de navegación basados en campos potenciales se inspiran en fenómenos de la física clásica, como los campos gravitatorios y los campos eléctricos. En estos, las masas o cargas generan fuerzas atractivas o repulsivas que influyen en otras partículas en función de la distancia.

Históricamente, las ideas que dieron origen al concepto de campo potencial provienen del trabajo de Isaac Newton, quien formuló la ley de la gravitación universal y describió la interacción entre masas mediante fuerzas a distancia. Más adelante, Charles-Augustin de Coulomb propuso una ley análoga para cargas eléctricas. Posteriormente, científicos como

Faraday y Maxwell desarrollaron formalmente la noción de campo como una entidad física y continua que permea el espacio, lo que dio lugar al concepto moderno de campos eléctricos y magnéticos [2].

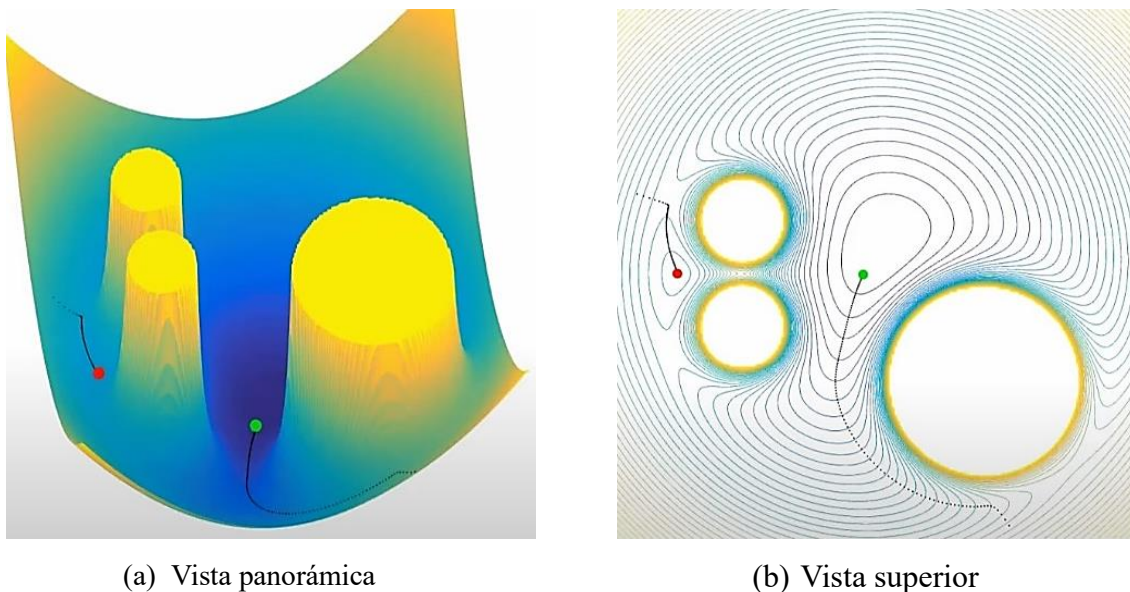


Figura 3.1 Representación de un ambiente de campos potenciales virtuales [3].

En robótica móvil, la analogía de los campos potenciales fue introducida por Oussama Khatib [4], quien propuso modelar el entorno como un campo de fuerzas donde el objetivo atrae al robot y los obstáculos lo repelen. Como se ilustra en la Figura 3.1, el espacio de trabajo se deforma virtualmente de modo que la meta actúa como una depresión y los obstáculos como elevaciones. Siguiendo el gradiente descendente de este campo, el robot evita colisiones y alcanza su destino, consolidando esta técnica como una solución eficiente y ampliamente utilizada en entornos cuya distribución de obstáculos, como el mobiliario en una casa o en un restaurante, permiten una correcta ejecución del algoritmo.

## 3.2 LIMITACIONES DEL MÉTODO Y SOLUCIONES HÍBRIDAS

Uno de los principales inconvenientes de los campos potenciales es la posibilidad de que el robot quede atrapado en mínimos locales, que son puntos donde la suma de fuerzas es nula pero no se ha alcanzado el objetivo, como se ilustra en la Figura 3.2.

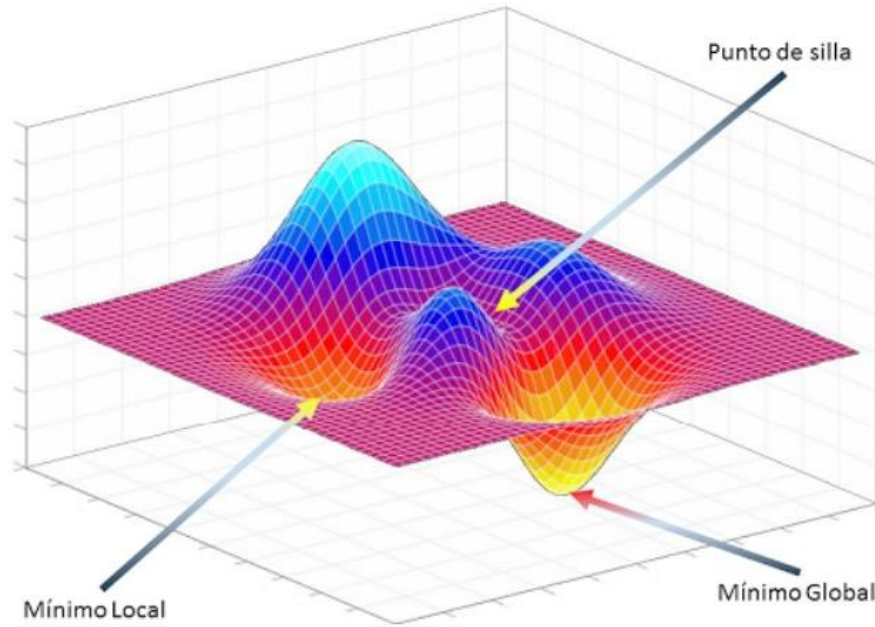


Figura 3.2 Representación de un mínimo local [5].

Para superar esta problemática, una solución habitual consiste en integrar el método reactivo con técnicas de planeación global. En particular, la construcción de una red topológica o grafo que permite abstraer la geometría del espacio en un conjunto de nodos situados en puntos estratégicos de navegación, interconectados por trayectorias viables que rodean los obstáculos conocidos. A partir de este modelo simplificado del entorno, es posible aplicar algoritmos de búsqueda como Dijkstra o A estrella para calcular la ruta global más eficiente y segura hacia el destino.

### 3.3 ALGORITMO DE DIJKSTRA EN NAVEGACIÓN AUTÓNOMA

El algoritmo de Dijkstra, propuesto por Edsger Dijkstra en 1959, es una técnica de optimización diseñada para encontrar el camino de menor costo dentro de una red de datos [6]. Para aplicar este principio a la navegación robótica, el entorno se representa mediante un grafo o red topológica. En este modelo, los nodos corresponden a puntos clave de paso ubicados en el espacio libre alrededor de los obstáculos estáticos, mientras que las aristas que los unen representan trayectorias viables y seguras.

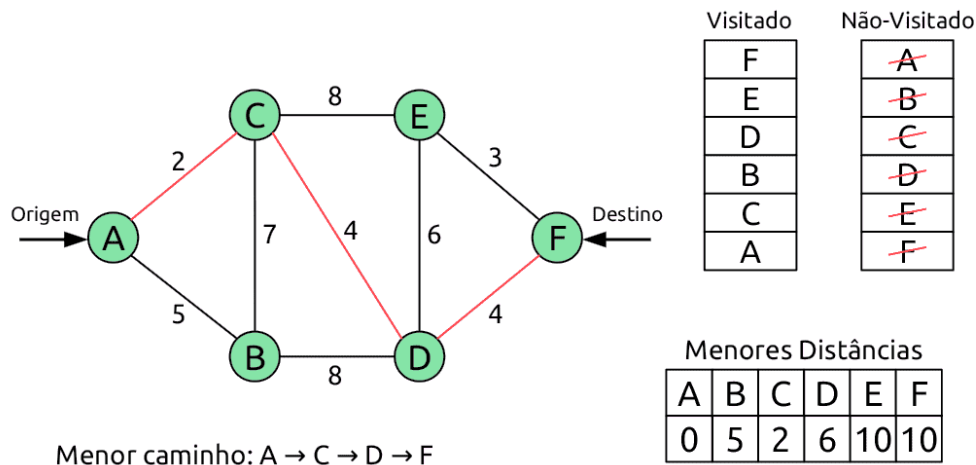


Figura 3.3 Ejemplo del algoritmo de Dijkstra [7].

La Figura 3.3 ilustra este proceso de selección sobre una estructura de red. Cada conexión posee un peso asociado a la distancia euclidiana entre nodos, permitiendo que el algoritmo explore iterativamente las alternativas disponibles desde el punto de origen. Mediante la acumulación progresiva de estos valores, Dijkstra selecciona la secuencia de nodos que conforma la trayectoria con menor distancia total hacia el destino, garantizando la obtención de una ruta global óptima al trabajar siempre con el menor peso asociado posible.

### 3.4 APLICACIONES ACTUALES DE CAMPOS POTENCIALES

En la actualidad, el uso de campos potenciales mantiene su relevancia técnica dentro de la robótica, particularmente mediante el desarrollo de variaciones orientadas a superar sus limitaciones clásicas. Como muestra de este avance continuo, destacan las siguientes investigaciones:

- En el año 2019, Rostami y sus colaboradores [8] propusieron una versión modificada de los campos potenciales artificiales, técnica conocida por sus siglas en inglés como APF, adaptada para operar en entornos desconocidos logrando eludir los mínimos locales y preservar la obtención de rutas óptimas.

- Por su parte, durante el año 2010, Zhang, Zhu y Song [9] lograron implementar un sistema APF funcional en tiempo real, enfocado específicamente en la navegación de robots industriales frente a obstáculos de geometría irregular.
- Más recientemente, en 2025, Jing y su equipo de investigación [10] combinaron técnicas de aprendizaje por refuerzo con la estrategia de campos potenciales para optimizar tareas de búsqueda utilizando vehículos aéreos no tripulados, cuyo comportamiento se ilustra en la Figura 3.4.



Figura 3.4 Ilustración del proyecto UAV usando campos potenciales híbridos [10].

Asimismo, en el ámbito académico nacional dentro de la Universidad Nacional Autónoma de México, mejor conocida como UNAM, destacan las aportaciones realizadas de esta línea de investigación:

- En el año 2016, Bustos de la Cruz y su equipo de trabajo [11] implementaron un sistema basado en campos potenciales integrado con reconocimiento visual de objetos, orientado al control de un robot móvil mediante la plataforma Android.
- Por su parte, en 2013, Cruz Navarro [12] desarrolló un esquema para optimizar los parámetros del campo potencial artificial mediante algoritmos genéticos. Esta técnica de búsqueda emula las leyes de la evolución biológica al generar hiperparámetros de forma aleatoria, evaluar su desempeño a lo largo de miles de ejecuciones simuladas y cruzar progresivamente las configuraciones más eficientes hasta obtener los valores requeridos para la navegación.

### 3.5 ALTERNATIVAS DE NAVEGACIÓN

Es sano reconocer que, si bien los campos potenciales son un método relativamente sencillo de implementar y computacionalmente eficiente, no es el único algoritmo en la navegación autónoma. Existen diversas estrategias metodológicas que actúan como alternativas o complementos al uso de los campos potenciales artificiales. Entre las más representativas destacan las siguientes:

- **Algoritmos de búsqueda global** como A estrella y sus variantes heurísticas resultan ideales para resolver problemas de planificación sobre grafos discretos[13].
- **SLAM y mapas de ocupación probabilísticos** que permiten construir representaciones del entorno en tiempo real para luego aplicar algoritmos de planificación de movimientos por muestreo[14]. Dentro de esta categoría, destacan el mapa de carreteras probabilístico, conocido como PRM, orientado a construir grafos globales para múltiples consultas, y el árbol aleatorio de exploración rápida, denominado RRT, especializado en encontrar trayectorias viables de forma incremental para una consulta única.
- **Control reactivo** como la arquitectura de subsunción propuesta por Rodney Brooks, que permite que el robot reaccione de manera directa e inmediata ante estímulos del entorno sin requerir mapa previo [15].
- **Aprendizaje automático** como los esquemas basados en aprendizaje por refuerzo y redes neuronales artificiales otorgan al vehículo una elevada capacidad de adaptación y toma de decisiones frente a entornos altamente dinámicos o cambiantes [16].

En resumen, los campos potenciales se mantienen como una herramienta plenamente válida y vigente en la navegación autónoma, sobre todo cuando se implementan en esquemas híbridos junto con técnicas de planificación global u otras técnicas que compensen sus limitaciones.

## **CAPÍTULO 4**

### **METODOLOGÍA**

En este capítulo se describe el marco metodológico y matemático propuesto para lograr la navegación autónoma de un robot móvil en un entorno con obstáculos. La estrategia central se basa en la construcción e implementación práctica de un campo potencial virtual que guíe al prototipo hacia su objetivo, añadiendo al campo un componente tangencial que ayude a superar la mayor debilidad de este algoritmo, los mínimos locales.

#### **4.1 LOCALIZACIÓN DEL ROBOT POR TRILATERACIÓN**

Para que un robot pueda desplazarse de forma autónoma en un entorno, resulta indispensable estimar con precisión su posición dentro del espacio de trabajo. En este proyecto se implementó el método de trilateración bidimensional [17]. Este enfoque se apoya en tres estaciones fijas de referencia en el sistema global, cuyas coordenadas en el plano son previamente conocidas. Mediante la emisión de señales desde cada estación hacia el prototipo, se miden tres distancias radiales desde las cuales se puede deducir la ubicación del robot.

Supóngase que las estaciones están ubicadas en los puntos  $P_1(x_1, y_1)$ ,  $P_2(x_2, y_2)$  y  $P_3(x_3, y_3)$ , y que el robot mide la distancia hacia cada una de ellas mediante  $d_1$ ,  $d_2$  y  $d_3$ . Dado que las estaciones y la antena del prototipo pueden situarse a diferentes alturas respecto al suelo, resulta

indispensable proyectar dichas mediciones al plano horizontal de la antena del robot. Para ello, considerando la diferencia de altura  $h_i$  entre la antena y la estación correspondiente, se aplican las siguientes expresiones pitagóricas:

$$r_1 = \sqrt{d_1^2 - h_1^2}$$

$$r_2 = \sqrt{d_2^2 - h_2^2}$$

$$r_3 = \sqrt{d_3^2 - h_3^2}$$

Dado que la posición del robot  $(x, y)$  debe encontrarse a una distancia proyectada  $r_i$  respecto a cada punto  $P_i$ , la localización espacial se modela mediante el siguiente sistema de tres ecuaciones de circunferencia:

$$(x - x_1)^2 + (y - y_1)^2 = r_1^2$$

$$(x - x_2)^2 + (y - y_2)^2 = r_2^2$$

$$(x - x_3)^2 + (y - y_3)^2 = r_3^2$$

Al desarrollar los binomios al cuadrado del sistema anterior, se obtienen las siguientes expresiones algebraicas:

$$x^2 - 2x x_1 + x_1^2 + y^2 - 2y y_1 + y_1^2 = r_1^2 \quad (4.1)$$

$$x^2 - 2x x_2 + x_2^2 + y^2 - 2y y_2 + y_2^2 = r_2^2 \quad (4.2)$$

$$x^2 - 2x x_3 + x_3^2 + y^2 - 2y y_3 + y_3^2 = r_3^2 \quad (4.3)$$

Para eliminar los términos cuadráticos  $x^2$  y  $y^2$ , se resta la ecuación (4.2) de la (4.1). Obteniendo la siguiente relación directa entre las coordenadas  $(x, y)$ :

$$2(x_2 - x_1)x + 2(y_2 - y_1)y = r_1^2 - r_2^2 + x_2^2 - x_1^2 + y_2^2 - y_1^2$$

Con el propósito de simplificar la notación y facilitar la resolución del sistema, se definen los siguientes coeficientes:

$$A = 2(x_2 - x_1)$$

$$B = 2(y_2 - y_1)$$

$$C = r_1^2 - r_2^2 + x_2^2 - x_1^2 + y_2^2 - y_1^2$$

De este modo, la ecuación resultante adopta la forma compacta expresada a continuación:

$$Ax + By = C \quad (4.4)$$

Al repetir el mismo procedimiento de resta entre las ecuaciones (4.3) y (4.1), se obtiene una segunda expresión dada por:

$$Dx + Ey = F \quad (4.5)$$

En este caso, las constantes del sistema quedan definidas mediante las siguientes igualdades:

$$D = 2(x_3 - x_1)$$

$$E = 2(y_3 - y_1)$$

$$F = r_1^2 - r_3^2 + x_3^2 - x_1^2 + y_3^2 - y_1^2$$

Este sistema lineal de dos ecuaciones con dos incógnitas puede resolverse mediante el método de Cramer, como se muestra a continuación:

$$x = \frac{\Delta x}{\Delta} \quad (4.6)$$

$$y = \frac{\Delta y}{\Delta} \quad (4.7)$$

En donde el determinante principal del sistema  $\Delta$  se calcula a partir de las siguientes constantes:

$$\Delta = AE - BD$$

$$\Delta = 4[(x_2 - x_1)(y_3 - y_1) - (y_2 - y_1)(x_3 - x_1)]$$

Para completar la solución de (4.6) y (4.7), los determinantes  $\Delta x$  y  $\Delta y$  se calculan mediante:

$$\Delta x = CE - BF$$

$$\Delta y = AF - CD$$

Al desarrollar las constantes previamente definidas para cada determinante, se obtiene:

$$\Delta x = 2(y_3 - y_1)(r_1^2 - r_2^2 + x_2^2 - x_1^2 + y_2^2 - y_1^2) - (y_2 - y_1)(r_1^2 - r_3^2 + x_3^2 - x_1^2 + y_3^2 - y_1^2)$$

$$\Delta y = 2(x_2 - x_1)(r_1^2 - r_3^2 + x_3^2 - x_1^2 + y_3^2 - y_1^2) - (x_3 - x_1)(r_1^2 - r_2^2 + x_2^2 - x_1^2 + y_2^2 - y_1^2)$$

Al sustituir estos determinantes junto con el valor de  $\Delta$  en las expresiones (4.6) y (4.7), y después de simplificar el factor escalar común, se obtuvieron las ecuaciones para las coordenadas:

$$x = \frac{(y_3 - y_1)(r_1^2 - r_2^2 + x_2^2 - x_1^2 + y_2^2 - y_1^2) - (y_2 - y_1)(r_1^2 - r_3^2 + x_3^2 - x_1^2 + y_3^2 - y_1^2)}{2[(x_2 - x_1)(y_3 - y_1) - (y_2 - y_1)(x_3 - x_1)]} \quad (4.8)$$

$$y = \frac{(x_2 - x_1)(r_1^2 - r_3^2 + x_3^2 - x_1^2 + y_3^2 - y_1^2) - (x_3 - x_1)(r_1^2 - r_2^2 + x_2^2 - x_1^2 + y_2^2 - y_1^2)}{2[(x_2 - x_1)(y_3 - y_1) - (y_2 - y_1)(x_3 - x_1)]} \quad (4.9)$$

Estas expresiones permitieron obtener directamente la posición aproximada absoluta del robot a partir de las distancias medidas y las coordenadas conocidas de las estaciones de referencia.

Una ventaja de linealizar las ecuaciones de las circunferencias originales fue que, aunque exista error en las distancias medidas del robot a cada una de las estaciones, se obtuvo una buena aproximación de la posición del robot. En la Figura 4.1 se puede observar una representación del comportamiento de este fenómeno en el plano de trabajo.

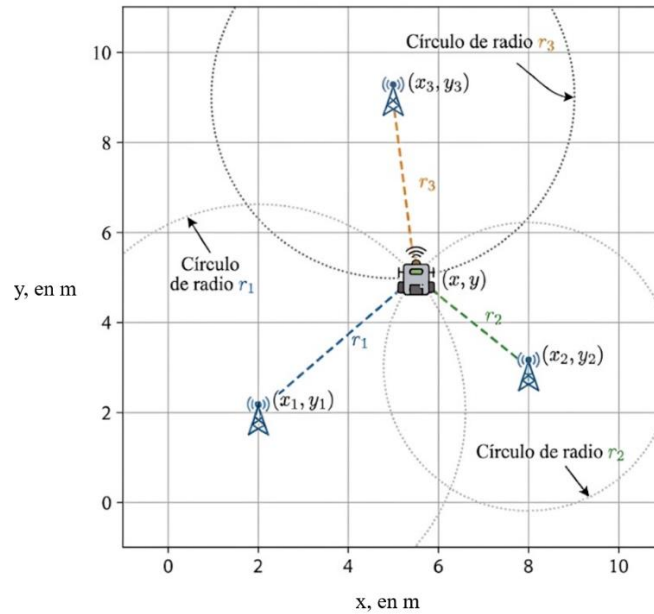


Figura 4.1 Esquema de localización por trilateración con tres sensores y un robot móvil.

## 4.2 DETECCIÓN Y SEGMENTACIÓN DE OBSTÁCULOS

Para la detección de obstáculos se empleó un sensor de distancia tipo LiDAR 2D, tecnología basada en luz cuyos componentes corresponden a los términos en inglés para la detección y rango mediante luz. Este dispositivo proporciona un conjunto de mediciones distribuidas angularmente en un barrido, donde cada lectura está definida por un par ordenado  $(d_i, \alpha_i)$ . En este esquema  $d_i$  es la distancia medida a cualquier objeto y  $\alpha_i$  el ángulo correspondiente respecto al marco de referencia del sensor.

A partir de estos datos, se calcula la posición de cada punto detectado en coordenadas cartesianas dentro del sistema de referencia del sensor mediante:

$$x_{o_i} = d_i \cos(\alpha_i) \quad (4.10)$$

$$y_{o_i} = d_i \sin(\alpha_i) \quad (4.11)$$

### 4.2.1 Segmentación por agrupamiento de mediciones

A diferencia de los enfoques que dividen el entorno en sectores predefinidos, en este trabajo se propuso una segmentación basada en la continuidad de las mediciones. Para ello, la agrupación se realizó directamente sobre el índice o la posición secuencial en la lista de los datos entregados por el LiDAR. Como el sensor proporciona las lecturas de distancia y orientación en un orden angular estricto, la continuidad entre índices válidos consecutivos bastó para delimitar un grupo.

De esta manera, se analizaron los índices de las mediciones válidas, aquellas cuya distancia es menor a un valor predefinido, y se agruparon en conjuntos consecutivos. Cuando existe una discontinuidad entre los índices, se forma un nuevo grupo o clúster. El cual representa un conjunto de puntos espacialmente contiguos asociados a un mismo obstáculo, por ejemplo, si la tercera y la quinta medición son válidas pero la cuarta no lo es, esa discontinuidad marca el fin de un grupo y el inicio de otro.

Este enfoque evitó transferir la totalidad de la nube de datos al sistema de navegación, agrupando las lecturas en entidades discretas y simplificando la escena. En la Figura 4.2 se ilustra un ejemplo de esta separación por grupos.

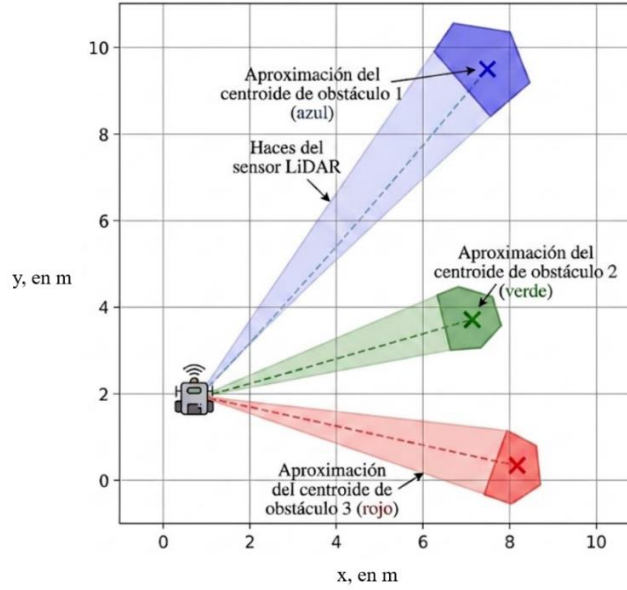


Figura 4.2 Representación de la aproximación de los centroides.

### 4.2.2 Cálculo del centroide de cada obstáculo

Para calcular las coordenadas de cada clúster identificado, se calculó un centroide que representa la posición promedio del obstáculo. Si un clúster contiene  $n$  puntos, su centroide se obtiene con el promedio de las coordenadas que se mostraron en las ecuaciones (4.10) y (4.11):

$$x_{obs} = \frac{1}{n} \sum_{i=1}^n x_{O_i} \quad (4.12)$$

$$y_{obs} = \frac{1}{n} \sum_{i=1}^n y_{O_i} \quad (4.13)$$

### 4.2.3 Transformación a un sistema de referencia del robot

Dado que el sensor generalmente no está alineado con el sistema de referencia principal del robot, fue necesario expresar la posición de cada obstáculo en un marco de referencia común. Para ello, se emplearon transformaciones en el plano, representadas mediante matrices de transformación homogénea.

Sea  $\overrightarrow{obs_s} = [x_{obs} \ y_{obs} \ 1]^T$  la posición del centroide de un obstáculo en el sistema de referencia del sensor, calculada con las ecuaciones (4.12) y (4.13). La transformación hacia el sistema de referencia del robot puede expresarse como:

$$\overrightarrow{obs_r} = \mathbf{T}_r^s \overrightarrow{obs_s} \quad (4.14)$$

donde  $\mathbf{T}_r^s \in \mathbb{R}^{3 \times 3}$  es la matriz de transformación homogénea que describe la rotación y traslación entre el sistema del sensor y del robot, definida como:

$$\mathbf{T}_r^s = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & t_x \\ \sin(\gamma) & \cos(\gamma) & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad (4.15)$$

donde  $\gamma$  representa el ángulo de orientación relativa entre ambos sistemas de referencia, mientras que los parámetros  $t_x$  y  $t_y$  definen las coordenadas de traslación del origen del marco de referencia del sensor respecto al centro del robot.

### 4.3 TRANSFORMACIÓN DE COORDENADAS AL SISTEMA MÓVIL

Para poder aplicar el modelo de campos potenciales, fue necesario que todas las posiciones del robot, el objetivo y los obstáculos se encuentren expresadas en un mismo sistema de referencia. En esta tesis, se adoptó el sistema de coordenadas móvil asociado al robot, cuya orientación está alineada con su eje central, como se ilustra en la Figura 4.3.

Para transformar cualquier punto desde el sistema global, el ambiente de trabajo, al sistema móvil del robot, se empleó una matriz de transformación homogénea, que considera tanto la rotación como la traslación del robot respecto al marco global. Este tipo de transformaciones es ampliamente utilizado en robótica y puede consultarse con mayor detalle en [18]. Esta matriz de transformación se define como:

$$\mathbf{T} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 & -x \cos(\theta) - y \sin(\theta) \\ -\sin(\theta) & \cos(\theta) & 0 & x \sin(\theta) - y \cos(\theta) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.16)$$

donde:

- $(x, y)$ : posición del robot en el sistema global,
- $\theta$ : ángulo de orientación del robot respecto al eje X global.

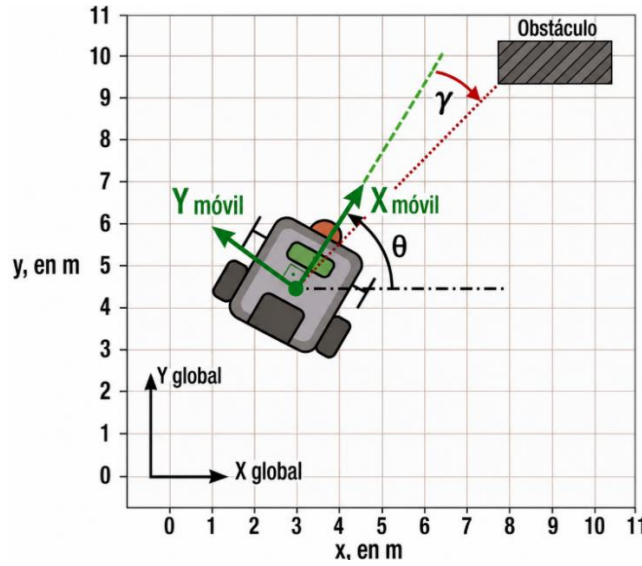


Figura 4.3 Relación entre el sistema global, el sistema del sensor y el sistema móvil del robot.

Dado un punto de destino en coordenadas globales  $(x_{dest}, y_{dest})$ , la transformación al sistema del robot se calcula como:

$$\begin{bmatrix} x_{dest_T} \\ y_{dest_T} \\ 0 \\ 1 \end{bmatrix} = T \begin{bmatrix} x_{dest} \\ y_{dest} \\ 0 \\ 1 \end{bmatrix}$$

En este contexto, la componente  $z$  se considera nula, ya que el movimiento del robot se restringió a un plano bidimensional. Además, dado que el objetivo consiste únicamente en alcanzar la posición del destino, sin importar su orientación relativa, las componentes asociadas a la rotación se omitieron en esta etapa del análisis. Al desarrollar la multiplicación matricial, se obtienen las siguientes expresiones:

$$x_{dest_T} = \cos(\theta) (x_{dest} - x) + \sin(\theta) (y_{dest} - y) \quad (4.17)$$

$$y_{dest_T} = -\sin(\theta) (x_{dest} - x) + \cos(\theta) (y_{dest} - y) \quad (4.18)$$

Estas expresiones definen la posición del destino referida al marco local del robot. Para realizar dicha transformación de coordenadas, se emplearon las coordenadas y la orientación actuales

del vehículo, en combinación con las ecuaciones (4.8) y (4.9), y lo explicado en la sección 5.4. Dichas coordenadas locales del objetivo sirven como insumo en el modelo de campos potenciales para el cálculo de las fuerzas atractivas y repulsivas.

## 4.4 CAMPO POTENCIAL VIRTUAL

Según Lynch y Park [19], los campos potenciales artificiales modelan la interacción entre el robot y su entorno dinámico mediante el uso de funciones escalares que representan niveles de energía virtual. En este marco conceptual, estas funciones se clasifican principalmente en dos tipos fundamentales: atractivas, destinadas a guiar al vehículo hacia la posición objetivo, y repulsivas, enfocadas en alejar al robot de los obstáculos presentes en la escena. De manera matemática, su forma general queda expresada mediante la siguiente relación:

$$U_{atr}(\vec{q}) = \frac{1}{2}(\vec{q} - \vec{q}_{dest})^T \mathbf{K}(\vec{q} - \vec{q}_{dest}) \quad (4.19)$$

$$U_{rep}(\vec{q}) = \frac{k}{2d^2(\vec{q}, \vec{B})} \quad (4.20)$$

donde:

- $\vec{q} = (x, y)$ : posición actual del robot en el sistema global,
- $\vec{q}_{dest} = (x_{dest}, y_{dest})$ : posición del objetivo en el sistema global,
- $\mathbf{K}$ : matriz de ganancias positivas definidas para el campo atractivo,
- $k$ : constante de escalamiento del campo repulsivo,
- $d(\vec{q}, \vec{B})$ : distancia entre el robot y el obstáculo más cercano.

Para esta tesis, se definieron formas particulares de estas funciones, basadas en el enfoque propuesto por el Dr. Jesús Savage Carmona en su clase de Robots móviles en la facultad de ingeniería de la Universidad Nacional Autónoma de México [20].

### 4.4.1 Función atractiva por el objetivo

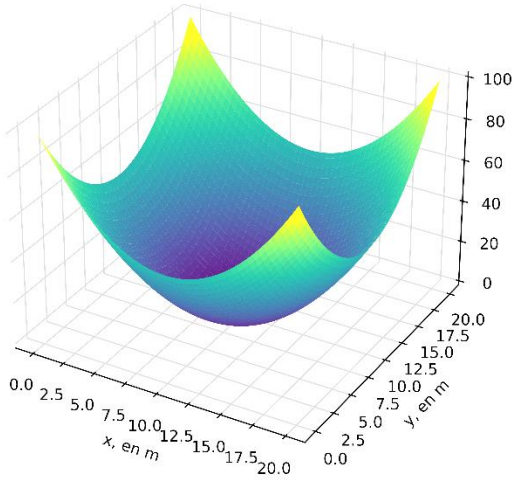
Para el campo atractivo, la matriz  $\mathbf{K} \in \mathbb{R}^{2 \times 2}$  se considera diagonal y proporcional a la identidad:

$$\mathbf{K} = \epsilon_1 \cdot \mathbf{I} = \epsilon_1 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

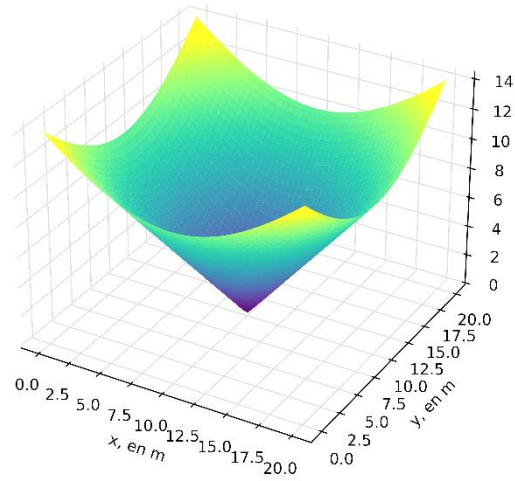
Luego de sustituir en la ecuación (4.19) y desarrollar el producto matricial, se obtiene una forma escalar más simple:

$$U_{atr}(\vec{q}) = \frac{1}{2} \epsilon_1 [(x - x_{dest})^2 + (y - y_{dest})^2] \quad (4.21)$$

Esta forma corresponde a una función cuadrática de tipo parabólico, como se observa en la Figura 4.4a, cuyo mínimo se encuentra en  $\vec{q}_{dest}$ . No obstante, a grandes distancias esta formulación puede generar fuerzas atractivas excesivas, lo que produce aceleraciones elevadas y dificulta un control preciso de movimiento.



(a) Atracción parabólica



(b) Atracción cónica

Figura 4.4 Comparación entre dos funciones atractivas con  $\epsilon_1 = 1$ ,  $\epsilon_2 = 1$  y  $\vec{q}_{dest} = (10, 10)[m]$ .

Para mitigar esta limitación, se propuso una función atractiva alternativa basada en una métrica lineal, como se muestra en la Figura 4.4b:

$$U_{atr}(\vec{q}) = \epsilon_2 \cdot \sqrt{(x - x_{dest})^2 + (y - y_{dest})^2} \quad (4.22)$$

La ecuación (4.22) produce un campo atractivo cuya pendiente permanece constante, lo que permite mantener una atracción uniforme incluso a grandes distancias. Con base en lo anterior, se propuso una función atractiva compuesta que adapta su forma según la distancia al objetivo, aprovechando las ventajas de ambas formulaciones, siguiendo lo expuesto en [20]:

$$U_{atr}(\vec{q}) = \begin{cases} \frac{1}{2} \epsilon_1 [(x - x_{dest})^2 + (y - y_{dest})^2] & \text{si } d(\vec{q}, \vec{q}_{dest}) \leq d_1 \\ \epsilon_2 \cdot \sqrt{(x - x_{dest})^2 + (y - y_{dest})^2} & \text{si } d(\vec{q}, \vec{q}_{dest}) > d_1 \end{cases} \quad (4.23)$$

donde  $d_1$  es un umbral de transición entre ambos modelos. Esta función resulta continua y optimiza el comportamiento de navegación tanto en las cercanías del objetivo como a largas distancias. Cabe destacar que, al haber trasladado el análisis al marco de referencia móvil del robot, la posición actual del vehículo se evalúa siempre en el origen  $(x, y) = (0, 0)$ , mientras que las coordenadas transformadas del destino  $\vec{q}_{dest} = (x_{dest}, y_{dest})$  provienen de la transformación de las coordenadas del destino al sistema móvil, visto en (4.17) y (4.18).

A partir de la función potencial atractiva definida, el comportamiento del robot se obtuvo calculando el gradiente del campo atractivo, ya que representa la fuerza virtual que actúa sobre el robot y lo atrae hacia el objetivo:

$$\vec{F}_{atr}(\vec{q}) = \nabla U_{atr}(\vec{q})$$

En el caso del campo atractivo parabólico, ecuación (4.21), el gradiente se expresa como:

$$\vec{F}_{atr}(\vec{q}) = \begin{bmatrix} \frac{\partial U_{atr}(x, y)}{\partial x} \\ \frac{\partial U_{atr}(x, y)}{\partial y} \end{bmatrix} \quad (4.24)$$

Luego de desarrollar las derivadas parciales:

$$\frac{\partial U_{atr}(x, y)}{\partial x} = \frac{\partial}{\partial x} \left( \frac{1}{2} \epsilon_1 [(x - x_{dest})^2 + (y - y_{dest})^2] \right)$$

$$\frac{\partial U_{atr}(x, y)}{\partial x} = \epsilon_1 (x - x_{dest})$$

$$\frac{\partial U_{atr}(x, y)}{\partial y} = \frac{\partial}{\partial y} \left( \frac{1}{2} \epsilon_1 [(x - x_{dest})^2 + (y - y_{dest})^2] \right)$$

$$\frac{\partial U_{atr}(x, y)}{\partial y} = \epsilon_1 (y - y_{dest})$$

Por lo tanto, el gradiente, que es el vector de fuerza del campo atractivo resultó:

$$\vec{F}_{atr}(\vec{q}) = \begin{bmatrix} \epsilon_1(x - x_{dest}) \\ \epsilon_1(y - y_{dest}) \end{bmatrix} \quad (4.25)$$

Para el caso del campo cónico, ecuación (4.22), se siguió un proceso similar y se obtuvo:

$$\vec{F}_{atr}(\vec{q}) = \epsilon_2 \cdot \frac{1}{\sqrt{(x - x_{dest})^2 + (y - y_{dest})^2}} \begin{bmatrix} x - x_{dest} \\ y - y_{dest} \end{bmatrix} \quad (4.26)$$

Este resultado corresponde a un vector con dirección hacia el objetivo y magnitud constante proporcional a  $\epsilon_2$ , lo que permitió mantener una atracción uniforme sin importar la distancia.

Ambas expresiones del gradiente se aplicaron condicionalmente, de acuerdo con la distancia al destino, siguiendo la regla definida en la ecuación (4.23).

#### 4.4.2 Función repulsiva por obstáculos

La ecuación (4.20) define un campo repulsivo inversamente proporcional al cuadrado de la distancia al obstáculo. En este esquema, el potencial tiende a infinito a medida que la distancia  $d(\vec{q}, \vec{B})$  se aproxima a cero. Esto no constituye un inconveniente, sino una propiedad deseable que favorece el establecimiento de barreras de seguridad, al generar una fuerza repulsiva mayor a menor distancia del obstáculo. Sin embargo, esta formulación presentó la desventaja de extender el efecto repulsivo de manera indefinida en el espacio, influyendo innecesariamente en la trayectoria del robot aun cuando los obstáculos se encuentran distantes.

Para solucionar esta limitación, se introdujo un umbral o radio de influencia denotado como  $d_0$ . Esta distancia acorta el alcance efectivo del obstáculo delimitando un margen exterior en el cual el campo repulsivo simplemente no tiene efecto. De este modo, cuando la distancia al obstáculo supera el valor de  $d_0$ , el potencial se anula por completo para no perturbar la navegación. Por el contrario, dentro de dicho radio, la función se suaviza para garantizar una transición gradual a la fuerza [20], tal como se muestra en la Figura 4.5. De esta forma, la formulación matemática del campo repulsivo queda definida por la siguiente ecuación por casos:

$$U_{rep}(\vec{q}) = \begin{cases} \frac{1}{2} k_{rep} \left( \frac{1}{\|\vec{q} - \vec{q}_{obs}\|} - \frac{1}{d_0} \right)^2 & si \|\vec{q} - \vec{q}_{obs}\| \leq d_0 \\ 0 & si \|\vec{q} - \vec{q}_{obs}\| > d_0 \end{cases} \quad (4.27)$$

donde:

- $\vec{q} = (x, y)$ : posición actual del robot, evaluada en el origen
- $\vec{q}_{obs} = (x_{obs}, y_{obs})$ : posición del obstáculo calculada en la ecuación (4.14)
- $\|\vec{q} - \vec{q}_{dest}\|$ : distancia euclidiana entre el robot y el obstáculo
- $k_{rep}$ : constante de ganancia del campo repulsivo
- $d_0$ : distancia de corte a partir de la cual el obstáculo deja de influir.

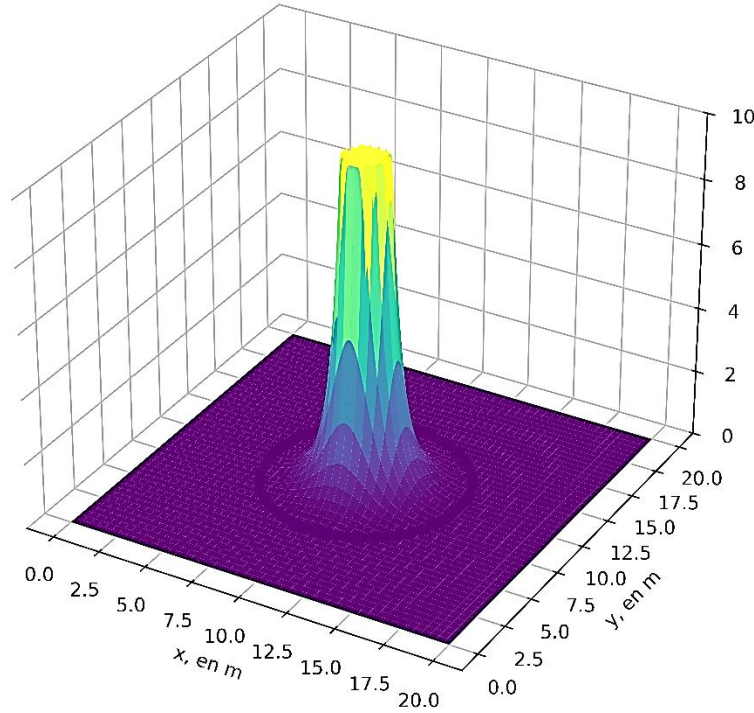


Figura 4.5 Representación del campo repulsivo con  $k_{rep} = 50$ ,  $d_0 = 5$  [m] y  $\vec{q}_{obs} = (10, 10)$  [m].

### Fuerza repulsiva pura

La fuerza repulsiva que experimenta el robot se obtuvo como el gradiente del potencial  $U_{rep}(\vec{q})$ , de forma análoga al tratamiento realizado para la fuerza atractiva en la ecuación (4.24):

$$\vec{F}_{rep}(\vec{q}) = \begin{bmatrix} \frac{\partial U_{rep}(x, y)}{\partial x} \\ \frac{\partial U_{rep}(x, y)}{\partial y} \end{bmatrix} \quad (4.28)$$

Al desarrollar las derivadas parciales de cada componente, se obtiene:

$$\vec{F}_{rep}(\vec{q}) = k_{rep} \left( \frac{1}{\|\vec{q} - \vec{q}_{obs}\|} - \frac{1}{d_0} \right) \frac{1}{\|\vec{q} - \vec{q}_{obs}\|^3} (\vec{q} - \vec{q}_{obs}) \quad (4.29)$$

Esta expresión genera una fuerza que actúa en la dirección opuesta al obstáculo, con magnitud decreciente al alejarse de él. Si bien esta fuerza es suficiente para evitar colisiones frontales, en configuraciones donde el robot debe rodear un obstáculo para alcanzar el objetivo puede conducir a oscilaciones o a estancamiento en mínimos locales.

### Componente tangencial para rodear obstáculos

Para mejorar el comportamiento durante la evasión, se añadió una componente tangencial a la fuerza repulsiva. Esta componente empuja al robot en una dirección perpendicular al vector que une su posición con el objetivo, permitiéndole rodear el obstáculo en lugar de detenerse frente a él. La elección del sentido de giro se determinó de forma consistente con la posición del obstáculo respecto al eje robot-objetivo, evitando cambios bruscos en la dirección de rodeo.

Sean:

- $\vec{g} = \vec{q}_{dest} - \vec{q}$ : vector desde el robot hacia el objetivo
- $\hat{g} = \frac{\vec{g}}{\|\vec{g}\|}$ : vector unitario en la dirección del objetivo
- $\hat{t}$ : vector unitario perpendicular a  $\hat{g}$ , tal que  $\hat{t} = [-\hat{g}_y, \hat{g}_x]^T$  rotando  $90^\circ$  en sentido antihorario.

Con el propósito de facilitar el análisis de la trayectoria y simplificar la validación del comportamiento del algoritmo durante la fase de pruebas, se fijó el signo de  $\hat{t}$  mediante un sentido único. Al imponer un producto punto positivo con el vector unitario dirigido hacia el obstáculo, se asegura un sentido de rodeo predecible y consistente ante cualquier obstáculo.

La componente tangencial de la fuerza repulsiva se definió como el gradiente de un potencial tangencial  $U_{tan}(\vec{q})$ , que depende de la distancia al obstáculo y de la proyección sobre  $\hat{t}$ :

$$U_{tan}(\vec{q}) = \begin{cases} k_{tan} \left( \frac{1}{\|\vec{q} - \vec{q}_{obs}\|} - \frac{1}{d_0} \right) \langle \hat{t}, \vec{q} - \vec{q}_{obs} \rangle & \text{si } \|\vec{q} - \vec{q}_{obs}\| \leq d_0 \\ 0 & \text{si } \|\vec{q} - \vec{q}_{obs}\| > d_0 \end{cases} \quad (4.30)$$

donde  $k_{tan}$  es la ganancia tangencial y  $\langle \cdot, \cdot \rangle$  denota el producto escalar.

La fuerza tangencial resultante se obtiene calculando el gradiente de  $U_{tan}(\vec{q})$ . Al desarrollar las derivadas parciales se llega a la siguiente expresión, tomando en cuenta que  $\vec{\Delta}_q = \vec{q} - \vec{q}_{obs}$ :

$$\vec{F}_{tan}(\vec{q}) = -k_{tan} \left[ \frac{1}{\|\vec{\Delta}_q\|^3} \langle \hat{t}, \vec{\Delta}_q \rangle \vec{\Delta}_q + \left( \frac{1}{\|\vec{\Delta}_q\|} - \frac{1}{d_0} \right) \hat{t} \right] \quad (4.31)$$

La fuerza repulsiva total que actúa sobre el robot es la suma de la componente repulsiva clásica  $\vec{F}_{rep}(\vec{q})$ , ecuación (4.29) y la componente tangencial  $\vec{F}_{tan}(\vec{q})$ , ecuación (4.31):

$$\vec{F}_{rep_T}(\vec{q}) = \vec{F}_{rep}(\vec{q}) + \vec{F}_{tan}(\vec{q}) \quad (4.32)$$

Esta formulación combinada permitió que el robot no solo sea repelido por el obstáculo, sino que también reciba un empuje lateral que favoreció el rodeo suave, al reducir la probabilidad de quedar atrapado en configuraciones simétricas y mejorar la convergencia hacia el objetivo.

Dado que los coeficientes de todas las fuerzas vistas no se obtienen a partir de una solución analítica directa, la totalidad de los hiperparámetros, las constantes atractivas, repulsivas, tangenciales, y los umbrales de distancias, fueron ajustados de forma empírica mediante pruebas, adaptándolos a la dinámica del robot y a las características específicas del entorno.

## 4.5 OBTENCIÓN DEL VECTOR DE FUERZA RESULTANTE

El movimiento del robot se guió mediante la fuerza resultante del campo potencial total, definida como el negativo del gradiente del potencial. En cada instante se calcula:

$$\vec{F}(\vec{q}) = -\nabla U_{total}(\vec{q}) = \nabla U_{atr}(\vec{q}) + \sum_{i=1}^n \nabla U_{rep_{T_i}}(\vec{q}) \quad (4.33)$$

donde  $\nabla U_{atr}$  y  $\nabla U_{rep_{T_i}}$  son los gradientes atractivo y repulsivo obtenidos en las secciones 4.4.1 y 4.4.2. El vector resultante  $\vec{F}(\vec{q}) = [F_x \ F_y]^T$  está expresando en el sistema de referencia móvil del robot, con su origen en el centro del eje de las ruedas y el eje x hacia el frente.

El vector  $\vec{F}(\vec{q})$  se utiliza directamente como señal de referencia para el generador de comandos de velocidad, tratado en la sección 5.2. La magnitud de  $\vec{F}$  modula la rapidez lineal deseada,

mientras que su orientación determina la dirección de avance y la rapidez angular necesaria para alinear al robot con el sentido del vector.

La implementación computacional de la expresión (4.33) se realiza sumando las contribuciones individuales de cada obstáculo y del objetivo. Este enfoque reactivo permitió que el robot responda en tiempo real a cambios en el entorno sin necesidad de planificar puntos intermedios explícitos.

## 4.6 RESUMEN DEL CICLO DE NAVEGACIÓN

El ciclo completo de navegación autónoma se ejecutó de manera iterativa, con la integración de percepción, planeación y control:

- Se estima la posición actual del robot mediante trilateración
- Se detectan obstáculos cercanos utilizando sensores de distancia
- Se calcula el campo potencial total en el sistema de referencia móvil
- Se obtiene el vector de fuerza resultante a  $\vec{F} = -\nabla U_{total}$  a partir del campo potencial total
- Se ajusta la orientación y velocidad del robot para avanzar hacia dicho punto

Este ciclo permitió al robot desplazarse de forma segura y eficiente hacia su destino, integrando planeación global y control reactivo.

En síntesis, la estrategia de navegación desarrollada en este capítulo se basó en la adaptabilidad local de los campos potenciales, lo que permitió una respuesta reactiva ante obstáculos. Si bien este enfoque fue eficiente y de fácil implementación, presentó limitaciones como la posible convergencia a mínimos locales, los cuáles fueron solucionados parcialmente con la adición del campo tangencial. Como extensión natural de este trabajo, se propone como mejora futura incorporar una planeación global basada en el algoritmo de Dijkstra o A estrella, combinada con una red topológica, con el fin de mitigar dichas limitaciones y generar trayectorias más robustas en entornos complejos. En los siguientes capítulos se presenta la implementación física del sistema reactivo en el prototipo robótico, incluyendo la caracterización de motores, selección de sensores y desarrollo del software de control.

## **CAPÍTULO 5**

### **MODELADO CINEMÁTICO Y CONTROL DIGITAL**

En el capítulo anterior se presentó la metodología para obtener el vector de fuerza resultante, considerando el destino global, la posición actual del robot, su orientación respecto al marco de referencia y los obstáculos detectados por los sensores. No obstante, aún es necesario explicar cómo el robot móvil transformó dicho vector, obtenido en la ecuación (4.33), en un movimiento físico de sus actuadores y cuál es la consecuencia de esta acción en el entorno real del prototipo.

Con este propósito, en el presente capítulo se aborda, en primer lugar, la cinemática de un robot diferencial, que establece las relaciones fundamentales entre la rapidez lineal y angular del chasis y las rapidezces de rotación de cada rueda. Posteriormente, se presenta la estrategia para generar comandos de rapidez lineal y angular de forma reactiva a partir del vector resultante del campo potencial, en el que se evitó el uso de perfiles predefinidos y permitió una respuesta inmediata a cambios en el entorno. Finalmente, se desarrolla el controlador digital que regula la rapidez de cada rueda, y se describe el esquema de fusión sensorial basado en un filtro de Kalman que integra odometría y diferentes sensores para estimar de manera robusta la pose del robot. Cerrando así el ciclo entre la planificación algorítmica y la ejecución física del sistema.

## 5.1 MODELO CINEMÁTICO BÁSICO DEL ROBOT DIFERENCIAL

Para predecir y controlar el movimiento de la plataforma móvil dentro de su entorno, es fundamental establecer una descripción matemática rigurosa de su comportamiento cinemático. En esta sección se analiza de manera formal la geometría del movimiento del robot diferencial, relacionando las rapideces angulares de sus ruedas accionadas de forma independiente con las rapideces lineal y angular del chasis. Asimismo, se introduce el concepto de odometría y se menciona cómo se puede aproximar posición y orientación con base en la tasa de cambio de las rapideces lineal y angular, calculadas con el procedimiento descrito.

### 5.1.1 Hipótesis y sistema de referencia

Se considera un vehículo con configuración de par diferencial que se desplaza sobre un plano. Se define un marco de referencia solidario al chasis, de ahora en adelante referenciado como  $\{R\}$ , con eje  $X_R$  orientado hacia el frente del robot y  $Y_R$  hacia la izquierda; el origen  $C$  se ubica en el punto medio del eje de ruedas, como se muestra en la Figura 5.1. La separación entre los planos medios de las ruedas es  $2b$  y el radio de cada rueda es  $r_R$  y  $r_L$ , para la rueda derecha e izquierda, respectivamente. Se asume rodadura pura y ausencia de velocidad lateral del chasis.

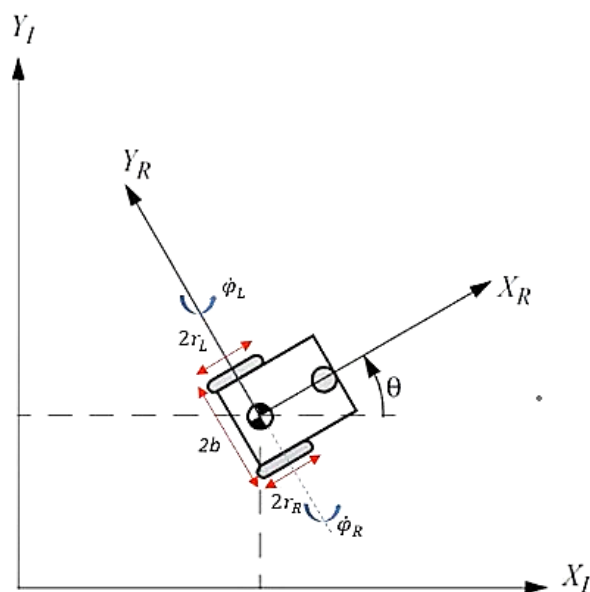


Figura 5.1 Representación del sistema móvil  $\{R\}$  y las dimensiones del robot [21].

### 5.1.2 Campo de velocidades de un cuerpo rígido

La velocidad de cualquier punto  $P$  del chasis puede expresarse mediante el campo de velocidades de un cuerpo rígido en el plano [22]:

$$\vec{v}_P = \vec{v}_C + \vec{\omega} \times \vec{r}_{CP} \quad (5.1)$$

donde  $\vec{v}_C = [v_C \ 0 \ 0]^T$  es la velocidad del origen  $C$  expresada en el sistema  $\{R\}$ ,  $\vec{\omega} = [0 \ 0 \ \omega]^T$  es la velocidad angular del chasis, convención positiva antihoraria, y  $\vec{r}_{CP}$  es el vector desde  $C$  hasta  $P$ .

Sea  $A$  el punto de contacto de la rueda derecha con el suelo en las coordenadas  $\vec{r}_{CA} = [0 \ -b \ 0]^T$  y  $B$  el de la rueda izquierda en  $\vec{r}_{CB} = [0 \ b \ 0]^T$ . Sustituyendo en (5.1):

$$\begin{aligned} \vec{v}_A &= \begin{bmatrix} v_C \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix} \times \begin{bmatrix} 0 \\ -b \\ 0 \end{bmatrix} \\ \vec{v}_A &= \begin{bmatrix} v_C + \omega b \\ 0 \\ 0 \end{bmatrix} \end{aligned} \quad (5.2)$$

$$\begin{aligned} \vec{v}_B &= \begin{bmatrix} v_C \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix} \times \begin{bmatrix} 0 \\ b \\ 0 \end{bmatrix} \\ \vec{v}_B &= \begin{bmatrix} v_C - \omega b \\ 0 \\ 0 \end{bmatrix} \end{aligned} \quad (5.3)$$

Las componentes son puramente longitudinales, a lo largo del eje  $X_R$ , lo cual es coherente con la restricción de rodadura pura: no existe velocidad lateral en los puntos de contacto, es decir, no hay componente en  $Y_R$ .

### 5.1.3 Relación con las velocidades angulares de la rueda

La rodadura pura impone que la rapidez tangencial del borde de cada rueda coincida con la rapidez longitudinal del punto de contacto, o sea, no hay deslizamiento entre el suelo y la rueda:

$$v_A = r_R \omega_R, \quad v_B = r_L \omega_L \quad (5.4)$$

donde  $\omega_R$  y  $\omega_L$  son las rapidezces angulares de las ruedas derecha e izquierda, respectivamente.

A partir de (5.2), (5.3) y (5.4) es posible obtener las rapidezces del chasis mediante combinaciones lineales simples. Luego de sumar los componentes en  $x$  de las ecuaciones (5.2) y (5.3):

$$\begin{aligned}\vec{v}_{A_x} + \vec{v}_{B_x} &= (v_C + \omega b) + (v_C - \omega b) \\ \vec{v}_{A_x} + \vec{v}_{B_x} &= 2v_C\end{aligned}\tag{5.5}$$

se puede despejar el valor de  $v_C$ :

$$v_C = \frac{\vec{v}_{A_x} + \vec{v}_{B_x}}{2}\tag{5.6}$$

De manera análoga, si se restan las ecuaciones (5.2) y (5.3):

$$\begin{aligned}\vec{v}_{A_x} - \vec{v}_{B_x} &= (v_C + \omega b) - (v_C - \omega b) \\ \vec{v}_{A_x} - \vec{v}_{B_x} &= 2\omega b\end{aligned}\tag{5.7}$$

se abre la posibilidad de obtener:

$$\omega = \frac{\vec{v}_{A_x} - \vec{v}_{B_x}}{2b}\tag{5.8}$$

Dada la condición de rodadura pura, se cumple que  $\vec{v}_{A_x} = v_A$ , al igual que  $\vec{v}_{B_x} = v_B$ , lo que permite construir las expresiones que relacionan las rapidezces de las ruedas, ecuaciones (5.4), con las del chasis [21], sustituyendo dichos valores en (5.6) y (5.8):

$$v_C = \frac{r_R \omega_R + r_L \omega_L}{2}\tag{5.9}$$

$$\omega = \frac{r_R \omega_R - r_L \omega_L}{2b}\tag{5.10}$$

Estas mismas expresiones se pueden representar de manera matricial de la siguiente manera:

$$\begin{bmatrix} v_C \\ \omega \end{bmatrix} = \frac{1}{2} \begin{bmatrix} r_R & r_L \\ \frac{r_R}{b} & -\frac{r_L}{b} \end{bmatrix} \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix}\tag{5.11}$$

El mapeo inverso que relaciona las rapidezces el chasis con las de las ruedas, se expresa despejando los valores de  $\omega_R$  y  $\omega_L$  de la ecuación (5.11) y se representa escalarmente como:

$$\omega_R = \frac{1}{r_R} (v_C + b \omega) \quad (5.12)$$

$$\omega_L = \frac{1}{r_L} (v_C - b \omega) \quad (5.13)$$

O bien, en su representación matricial equivalente:

$$\begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} = \begin{bmatrix} \frac{1}{r_R} & \frac{b}{r_R} \\ \frac{1}{r_L} & -\frac{b}{r_L} \end{bmatrix} \begin{bmatrix} v_C \\ \omega \end{bmatrix} \quad (5.14)$$

#### 5.1.4 Observaciones prácticas

- **Casos límite.** Si  $\omega = 0$  y las ruedas tienen un mismo radio  $r$ , entonces  $\omega_R = \frac{v_C}{r}$  y  $\omega_L = \frac{v_C}{r}$ . Si  $v_C = 0$ , entonces  $\omega_R = \frac{b}{r} \omega$  y  $-\omega_L = \frac{b}{r} \omega$ .
- **Parámetros efectivos.** En operación real, pueden aparecer deslizamientos y asimetrías mecánicas que ocasionan errores en el modelado; por ello, la estimación de la posición y orientación basada únicamente en el modelo cinemático puede diferir de la realidad.
- **Vinculación con la pose.** La pose del robot puede estimarse a partir de las rapideces angulares de las ruedas utilizando el modelo descrito. Las rapideces lineal y angular del chasis se calculan a partir de (5.11) y corresponde el movimiento instantáneo del robot en su sistema de referencia local. La evolución de la pose  $[x_C \ y_C \ \theta]^T$  en el sistema global se obtiene mediante  $\dot{x}_C = v_C \cos(\theta)$ ,  $\dot{y}_C = v_C \sin(\theta)$  y  $\dot{\theta} = \omega$ . Estas ecuaciones pueden integrarse numéricamente en el tiempo para estimar la posición del robot en el sistema global, constituyendo lo que se conoce como odometría.
- **Fusión sensorial.** Debido a la acumulación de error inherente a la odometría, principalmente por deslizamiento y ruido en los sensores, en este trabajo se empleó un esquema de fusión sensorial basado en un filtro de Kalman extendido. En este enfoque, la odometría se utilizó como modelo de predicción, mientras que mediciones externas, como la posición y la orientación obtenidas mediante sensores externos, se emplearon para corregir dicha predicción. Este esquema permitió obtener una estimación más robusta y consistente de la pose del robot. La formulación completa del filtro se presenta en una sección posterior.

## 5.2 GENERACIÓN DE COMANDOS DE RAPIDEZ

A diferencia de otros enfoques, como los basados en perfiles de rapidez predefinidos en el tiempo, en este trabajo se adoptó un esquema de control reactivo en el que las rapideces del robot se generan directamente a partir del vector de fuerza resultante. Este enfoque permitió modificar el movimiento en tiempo real ante la presencia de obstáculos y cambios en el entorno debido a que el vector de fuerza fue actualizado constantemente a una frecuencia de 20Hz.

### 5.2.1 Vector de referencia a partir de campos potenciales

El método de campos potenciales proporciona, mediante la ecuación (4.33), un vector resultante en el plano, el cual está expresado en el sistema de referencia del robot y representa la interacción de las fuerzas atractivas y repulsivas del ambiente potencial. Este vector indica simultáneamente la dirección deseada de movimiento y la magnitud de la acción a realizar.

### 5.2.2 Descomposición en velocidad lineal y angular

Para convertir este vector en comandos de velocidad, se calculó primero su magnitud:

$$|\vec{F}(\vec{q})| = \sqrt{\vec{F}_x^2 + \vec{F}_y^2} \quad (5.15)$$

y su orientación respecto al marco de referencia  $\{R\}$ :

$$\beta = \arctan\left(\frac{\vec{F}_y}{\vec{F}_x}\right) \quad (5.16)$$

donde  $\beta$  representa la diferencia angular entre la orientación del robot y la orientación del vector  $\vec{F}(\vec{q})$ , cuyo comportamiento es producto de las mediciones del destino y de los obstáculos.

### 5.2.3 Ley de control proporcional

Con el valor de la diferencia angular, calculado en la ecuación (5.16), se definió una relación entre esta y la rapidez angular del robot. La cual se definió proporcionalmente:

$$\omega = k_{\omega} \beta \quad (5.17)$$

esto induce una rotación hasta que la orientación del robot y la del vector de fuerza sean iguales.

Por otro lado, la rapidez lineal del robot se definió a partir de la ecuación (5.15), como proporcional a la magnitud del vector, escalada por el ángulo de alineación:

$$v_c = \begin{cases} k_v |\vec{F}(\vec{q})| \cos(\beta) & \text{si } \cos(\beta) > 0 \\ 0 & \text{si } \cos(\beta) \leq 0 \end{cases} \quad (5.18)$$

El término  $\cos(\beta)$  modula la rapidez lineal en función del grado de alineación entre la orientación del robot y la dirección del vector de referencia. Así, cuando el error angular es elevado, la componente de avance disminuye para priorizar la rotación del vehículo hasta alinearse con la trayectoria deseada. Al mismo tiempo, esta restricción prohíbe desplazamientos en sentido negativo sobre el eje  $X_R$ , previniendo colisiones en zonas no cubiertas por el sensor de obstáculos, lo que resguarda la seguridad del robot y de su entorno.

### 5.2.4 Saturación de velocidades

Para garantizar que las velocidades generadas se mantengan dentro de los límites físicos del sistema, se aplicaron funciones de saturación, lo cual evita exigir al sistema valores no alcanzables o peligrosos para el ambiente de trabajo donde se desempeñó el robot:

$$v_c \in [0, v_{c_{max}}] \quad (5.19)$$

$$\omega \in [-\omega_{max}, \omega_{max}] \quad (5.20)$$

### 5.2.5 Ventajas del enfoque reactivo

Este esquema permitió generar comandos de rapidez de forma continua en función del estado actual del entorno. A diferencia de los perfiles trapezoidales, que dependen explícitamente del tiempo, el enfoque propuesto:

- Permitió realizar movimientos simultáneos de rotación y traslación, sustituyendo los valores de las ecuaciones (5.17) y (5.18) en (5.14), obteniendo las referencias del control digital
- Reaccionó de forma casi inmediata ante cambios en el entorno, puesto que, el vector resultante de la ecuación (4.33) fue actualizado constantemente con el estado del entorno
- Facilitó la evasión de obstáculos en tiempo real, permitiendo seguir de manera más flexible el comportamiento del vector resultante.

## 5.3 CONTROL DIGITAL

El control de rapidez de las ruedas en un robot móvil requiere una estrategia que minimice el error en régimen permanente y garantice una respuesta dinámica estable frente a las variaciones de carga y las perturbaciones del entorno. Con este propósito, se eligió e implemento un algoritmo de control proporcional integral derivativo, conocido como PID por sus siglas en inglés. La formulación clásica de esta ley de control en el dominio del tiempo continuo se define mediante la siguiente relación matemática:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (5.21)$$

donde, la variable  $u(t)$  representa la señal de control o acción calculada que se envía hacia el actuador del sistema. Por su parte, el término  $e(t)$  constituye el error de seguimiento instantáneo, el cual se calcula mediante la diferencia entre la referencia de la rapidez deseada y la variable medida por los sensores en la retroalimentación. La constante  $K_p$  corresponde a la ganancia proporcional, la cual genera una acción directamente proporcional a la magnitud del error actual. La constante  $K_i$  denota la ganancia integral, cuya función es acumular el error a lo largo del tiempo para eliminar completamente el error en estado estacionario. Por último, la constante  $K_d$

representa la ganancia derivativa, la cual evalúa la tasa de cambio del error para anticipar su comportamiento futuro y añadir amortiguamiento al sistema [23].

Para el diseño y la sintonización del controlador se utilizó el software MATLAB [24], el cual emplea una formulación modificada del PID que incluye un filtro de primer orden en la acción derivativa con la finalidad de atenuar el impacto del ruido de alta frecuencia en la señal de control [25]. Esta expresión en el dominio de la frecuencia es:

$$C(s) = \left( K_p + \frac{K_i}{s} + \frac{K_d s}{T_f s + 1} \right) E(s) \quad (5.22)$$

### 5.3.1 Discretización bilineal

Para implementar el controlador PID en un microcontrolador digital, fue necesario transformar la ecuación (5.22) del dominio continuo al dominio discreto. En este trabajo se empleó el método de Tustin [26], también conocido como transformación bilineal. Este método aproxima la variable compleja  $s$  mediante la sustitución:

$$s \approx \alpha \frac{(1 - z^{-1})}{1 + z^{-1}}, \quad \alpha = \frac{2}{T_s} \quad (5.23)$$

donde el parámetro  $T_s$  representa el periodo de muestreo del sistema, mientras que la variable compleja  $z$  corresponde al operador del dominio discreto en la transformada  $Z$  [27], el cual define el avance o retardo temporal entre las muestras consecutivas de la señal digitalizada.

La ecuación (5.22) puede reescribirse como una función racional de segundo orden en  $s$ :

$$C(s) = \frac{A s^2 + B s + C}{T_f s^2 + s} E(s) \quad (5.24)$$

donde los coeficientes están dados por:

$$A = K_p T_f + K_d \quad (5.25)$$

$$B = K_p + K_i T_f \quad (5.26)$$

$$C = K_i \quad (5.27)$$

Luego de sustituir (5.23) en (5.24) y multiplicando numerador y denominador por  $(1 + z^{-1})^2$ , se obtiene una función de transferencia discreta expresada en términos de  $z^{-1}$ :

$$C(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} E(z) \quad (5.28)$$

Los coeficientes del controlador discreto se obtuvieron a partir de la expansión algebraica resultante de la sustitución de Tustin en la ecuación (5.24). Tras desarrollar los productos y agrupar términos en potencias de  $z^{-1}$ , el numerador y el denominador pudieron escribirse como polinomios de segundo orden.

El numerador del controlador discreto queda expresado como:

$$(A \alpha^2 + B \alpha + C) + (-2A \alpha^2 + 2C)z^{-1} + (A \alpha^2 - B \alpha + C)z^{-2} \quad (5.29)$$

Mientras que el denominador resulta

$$(T_f \alpha^2 + \alpha) + (-2T_f \alpha^2)z^{-1} + (T_f \alpha^2 - \alpha)z^{-2} \quad (5.30)$$

Para llevar la función de transferencia a la forma normalizada de la ecuación (5.28), todos los coeficientes se dividieron entre el término constante del denominador. De esta manera, se obtuvieron los coeficientes del controlador PID digital:

$$b_0 = \frac{A \alpha^2 + B \alpha + C}{T_f \alpha^2 + \alpha} \quad (5.31)$$

$$b_1 = \frac{-2A \alpha^2 + 2C}{T_f \alpha^2 + \alpha} \quad (5.32)$$

$$b_2 = \frac{A \alpha^2 - B \alpha + C}{T_f \alpha^2 + \alpha} \quad (5.33)$$

$$a_1 = \frac{-2T_f \alpha}{T_f \alpha + 1} \quad (5.34)$$

$$a_2 = \frac{T_f \alpha - 1}{T_f \alpha + 1} \quad (5.35)$$

donde  $\alpha = \frac{2}{T_s}$  y los coeficientes  $A$ ,  $B$  y  $C$  están definidos en las ecuaciones (5.25), (5.26) y (5.27).

Estas expresiones permitieron calcular directamente los parámetros del controlador discreto a partir de las constantes del PID continuo y del periodo de muestreo, garantizando que el comportamiento del controlador digital sea consistente con el diseño realizado en el dominio continuo.

### 5.3.2 Obtención de la ecuación en diferencias

La función de transferencia discreta del controlador puede escribirse como la relación entre la señal de control  $U(z)$  y el error  $E(z)$ :

$$C(z) = \frac{U(z)}{E(z)} \quad (5.36)$$

Luego de multiplicar ambos lados de la ecuación (5.28) por el denominador, se obtuvo:

$$U(z)(1 + a_1 z^{-1} + a_2 z^{-2}) = E(z)(b_0 + b_1 z^{-1} + b_2 z^{-2}) \quad (5.37)$$

Posteriormente, se aplicó la transformada Z inversa considerando que el operador  $z^{-1}$  representa un retardo de una muestra en el tiempo discreto. De esta manera, el índice  $k$  indica el instante actual de muestreo, mientras que expresiones como  $k - 1$  y  $k - 2$  representan los valores calculados de la señal en el instante inmediatamente anterior y el antepenúltimo periodo, respectivamente. Esta propiedad de recursividad permitió despejar la señal de control actual para obtener la ecuación en diferencias del controlador PID digital:

$$U[k] = b_0 E[k] + b_1 E[k - 1] + b_2 E[k - 2] - a_1 U[k - 1] - a_2 U[k - 2] \quad (5.38)$$

El desarrollo analítico del controlador PID digital presentado en esta sección surgió como una consecuencia directa del diseño y las simulaciones realizadas previamente en MATLAB. Con el objetivo de implementar este controlador en un microcontrolador y asegurar que su comportamiento fuera lo más similar posible al observado en el entorno de simulación, se llevó a cabo el desarrollo matemático completo del proceso de discretización. Este procedimiento permitió obtener explícitamente la ecuación en diferencias del controlador, garantizando que la implementación digital reproduzca de manera fiel la dinámica del controlador diseñado.

## 5.4 ESTIMACIÓN DE LA POSE MEDIANTE EL FILTRO KALMAN

Hasta este punto se presentaron diversas metodologías para estimar la pose del robot móvil en el sistema de referencia global, la cual consta de las coordenadas de posición  $x$  e  $y$ , así como de la orientación  $\theta$ . La primera alternativa consistió en el cálculo de la posición mediante trilateración explicada en la sección 4.1. La segunda técnica correspondió a la odometría, explicada brevemente en la sección 5.1.4, la cual permite aproximar tanto la posición como la orientación a partir de las rapidezces  $v_c$  y  $\omega$  del vehículo. Adicionalmente, se utilizó la medición directa de la rapidez angular proveniente de una unidad de medición inercial, mejor conocida como IMU por sus siglas en inglés, para calcular la orientación.

A pesar de que cada una de estas técnicas ofrece ventajas específicas para el cálculo de la pose, en la práctica todas las mediciones estuvieron sujetas a incertidumbres o imperfecciones que limitaron su precisión de manera individual, algunas de ellas fueron:

- La estimación basada en odometría a partir de las rapidezces lineal y angular, ofreció un muestreo continuo de la trayectoria, pero acumuló error a lo largo del tiempo debido al deslizamiento de las ruedas, imperfecciones mecánicas y variaciones en el radio de las ruedas.
- Las mediciones de posición por trilateración no acumularon error a largo plazo, sin embargo, presentaron ruido, reflexiones de señal, pérdidas temporales de comunicación y retardos en el viaje de la señal debido a estorbos entre la antena del robot y las balizas.
- La integración numérica de la rapidez angular de la IMU proporcionó una respuesta rápida ante cambios de dirección, pero sufre de una deriva paulatina provocada por pequeños sesgos no modelados en el sensor

Para superar estas limitaciones individuales y aprovechar las ventajas complementarias de cada metodología, se empleó un filtro de Kalman extendido [28], conocido como EKF por sus siglas en inglés. Este algoritmo combina las actualizaciones de la odometría y la IMU con la corrección global absoluta de la trilateración, lo que permite estimar de manera recursiva la pose óptima del robot aun en presencia de ruido gaussiano y no linealidades matemáticas.

### 5.4.1 Definición del vector de estado y modelo del proceso

Para formular el filtro de Kalman extendido orientado a la estimación de la pose del robot móvil en el plano bidimensional, se definió un vector de estado  $\vec{p}_k \in \mathbb{R}^3$  en el instante discreto  $k$ , representado por las coordenadas de posición cartesiana y el ángulo de orientación:

$$\vec{p}_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} \quad (5.39)$$

La entrada de control  $\vec{u}_k \in \mathbb{R}^2$  comprende la rapidez lineal  $v_{c_k}$  y la rapidez angular  $\omega_k$ , calculadas con las ecuaciones (5.9) y (5.10) respectivamente:

$$\vec{u}_k = \begin{bmatrix} v_{c_k} \\ \omega_k \end{bmatrix} \quad (5.40)$$

La ecuación no lineal de transición para el estado actual, dado el estado anterior  $\vec{f}(\vec{p}_{k-1}, u_k)$ , se expresa de la siguiente manera:

$$\begin{aligned} \vec{p}_k &= \vec{f}(\vec{p}_{k-1}, \vec{u}_k) + \vec{w}_k \\ \vec{p}_{k|k-1} &= \begin{bmatrix} x_{k-1} + v_{c_k} \cos(\theta_{k-1}) T_s \\ y_{k-1} + v_{c_k} \sin(\theta_{k-1}) T_s \\ \theta_{k-1} + \omega_k T_s \end{bmatrix} + \vec{w}_k \end{aligned} \quad (5.41)$$

donde  $T_s$  es el periodo de muestreo constante, cuyo valor en este trabajo fue de 0.05 s y el término  $\vec{w}_k \sim \mathcal{N}(0, \mathbf{Q})$  representa el ruido del proceso gaussiano de media cero, cuya matriz de covarianza diagonal  $\mathbf{Q} \in \mathbb{R}^{3 \times 3}$  modela la incertidumbre asociada al deslizamiento de las ruedas y a la no linealidad cinemática.

$$\mathbf{Q} = \begin{bmatrix} \sigma_{xy}^2 & 0 & 0 \\ 0 & \sigma_{xy}^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix} \quad (5.42)$$

En la cual,  $\sigma_{xy}^2$  indica el nivel de incertidumbre en las posiciones  $x$  e  $y$  derivadas de la odometría de las ruedas, mientras que  $\sigma_\theta^2$  asigna la de la orientación  $\theta$ . Todas estas incertidumbres son hiperparámetros que se definieron con base en las características del comportamiento del robot. Para este trabajo se usaron 0.001 para la incertidumbre de las posiciones y 0.005 para la incertidumbre de la orientación.

$$\mathbf{Q}_{inicial} = \begin{bmatrix} 0.001 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 0.005 \end{bmatrix}$$

### 5.4.2 Etapa de predicción y linealización del modelo

La etapa de predicción utiliza la odometría de las ruedas para proyectar el estado y la covarianza del error hacia el siguiente instante de tiempo. Para distinguir los momentos del proceso de estimación, se empleó la notación con barra vertical donde los índices de la forma  $A|B$  indican la estimación en el instante  $A$  dada la información sensorial disponible hasta el instante  $B$ . De esta manera, el símbolo  $\vec{p}_{k|k}$  representa la estimación corregida, a posteriori, del paso anterior, mientras que  $\vec{p}_{k|k-1}$  denota la estimación predicha, a priori, para el instante actual antes de incorporar las mediciones de sensores externos.

Debido a que la función de transición  $\vec{f}$  es no lineal, ocasionando que las probabilidades gaussianas se deformen haciendo que el filtro falle o diverja, fue necesario calcular la matriz jacobiana  $\mathbf{F}_k = \left. \frac{\partial \vec{f}}{\partial \vec{p}} \right|_{\vec{p}_{k-1|k-1}}$  para contrarrestar localmente las no linealidades alrededor de la orientación actual del robot y poder predecir correctamente la propagación de la incertidumbre en la matriz de covarianza sin romper las propiedades del filtro.

$$\mathbf{F}_k = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial \theta} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial \theta} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial \theta} \end{bmatrix}$$

$$\mathbf{F}_k = \begin{bmatrix} 1 & 0 & -v_{C_k} \sin(\theta_{k-1}) T_s \\ 0 & 1 & v_{C_k} \cos(\theta_{k-1}) T_s \\ 0 & 0 & 1 \end{bmatrix} \quad (5.43)$$

Con esta matriz de transición linealizada, la matriz de covarianza del error  $\mathbf{P}_{k|k-1} \in \mathbb{R}^{3 \times 3}$  se actualiza mediante:

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q} \quad (5.44)$$

Posteriormente a la actualización del ángulo predicho, se aplicó una operación de normalización angular para mantener el valor de  $\theta$  acotado en el intervalo de  $[-\pi, \pi]$ . Los valores iniciales de la matriz de covarianza son hiperparámetros que se definieron empíricamente con varianzas de  $1.0 \text{ m}^2$  para la posición y  $0.1 \text{ rad}^2$  para la orientación, acomodados en la diagonal principal de la matriz de covarianza del error.

### 5.4.3 Etapa de actualización con mediciones de trilateración

Cuando el sistema recibía una nueva lectura de posición válida proveniente de las antenas del sistema de trilateración, se ejecutó el paso de corrección cartesiana. El vector de observación  $\vec{z}_{tri} \in \mathbb{R}^2$  y la función de observación  $\vec{h}_{tri}(\vec{p}_k)$  se definen cómo:

$$\vec{z}_{tri} = \begin{bmatrix} x_{leída} \\ y_{leída} \end{bmatrix} \quad (5.45)$$

$$\vec{h}_{tri}(\vec{p}_k) = \begin{bmatrix} x_k \\ y_k \end{bmatrix} \quad (5.46)$$

Dado que la función de medición es directamente lineal con respecto a las posiciones del vector de estado, la matriz jacobiana de observación  $\mathbf{H}_{tri} \in \mathbb{R}^{2 \times 3}$  adopta una forma constante:

$$\mathbf{H}_{tri} = \frac{\partial \vec{h}_{tri}}{\partial \vec{p}} \quad (5.47)$$

$$\mathbf{H}_{tri} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

El proceso de actualización calcula primero el vector de innovación o residuo  $\vec{r}_k \in \mathbb{R}^2$ , seguido de la covarianza de la innovación  $\mathbf{S}_k \in \mathbb{R}^{2 \times 2}$ , incorporando la covarianza del ruido de medición de la trilateración que fue definido de manera empírica como  $R_{tri} = 0.04 \text{ m}^2$ .

$$\vec{r}_k = \vec{z}_{tri} - \vec{h}_{tri}(\vec{p}_{k|k-1})$$

$$\vec{r}_k = \begin{bmatrix} x_{leído} - x_{k|k-1} \\ y_{leído} - y_{k|k-1} \end{bmatrix} \quad (5.48)$$

$$\mathbf{S}_k = \mathbf{H}_{tri} \mathbf{P}_{k|k-1} \mathbf{H}_{tri}^T + \begin{bmatrix} R_{tri} & 0 \\ 0 & R_{tri} \end{bmatrix}$$

$$\mathbf{S}_k = \begin{bmatrix} \mathbf{P}_{0,0} + R_{tri} & \mathbf{P}_{0,1} \\ \mathbf{P}_{1,0} & \mathbf{P}_{1,1} + R_{tri} \end{bmatrix} \quad (5.49)$$

A continuación, se determina la ganancia de Kalman  $\mathbf{K}_k \in \mathbb{R}^{3 \times 2}$  mediante la inversión explícita de la matriz de dos por dos  $\mathbf{S}_k$ , para finalmente actualizar el vector de estado  $\vec{p}$  y la matriz de covarianza a posteriori:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_{tri}^T \mathbf{S}_k^{-1} \quad (5.50)$$

$$\vec{p}_{k|k} = \vec{p}_{k|k-1} + \mathbf{K}_k \vec{r}_k \quad (5.51)$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_{tri}) \mathbf{P}_{k|k-1} \quad (5.52)$$

Donde  $\mathbf{I} \in \mathbb{R}^{3 \times 3}$  denota la matriz identidad.

#### 5.4.4 Etapa de actualización de la orientación con IMU

Para corregir la deriva de la orientación generada durante el paso de predicción, se diseñó un mecanismo de actualización escalar genérico que aplica tanto a las lecturas del giroscopio integrado en la IMU como al ángulo acumulado por la odometría de los codificadores rotativos. La medida en la orientación  $z_\theta \in \mathbb{R}$  se relaciona con el estado mediante la función lineal  $h_\theta(\vec{p}_k) = \theta_k$ , lo cual produce el vector jacobiano de observación  $\mathbf{H}_\theta \in \mathbb{R}^{1 \times 3}$ :

$$\mathbf{H}_\theta = [0 \quad 0 \quad 1] \quad (5.53)$$

La innovación de orientación  $r_{theta}$  representa la diferencia entre la medición recibida  $z_\theta$  y la orientación predicha  $\theta_{k|k-1}$ . Para garantizar la continuidad angular y evitar saltos abruptos cuando el ángulo cruza las fronteras del círculo unitario, la diferencia angular se acota formalmente dentro del intervalo de  $-\pi$  a  $\pi$  radianes mediante la siguiente función por casos:

$$r_\theta = \begin{cases} (z_\theta - \theta_{k|k-1}) - 2\pi, & \text{si } (z_\theta - \theta_{k|k-1}) > \pi \\ (z_\theta - \theta_{k|k-1}) + 2\pi, & \text{si } (z_\theta - \theta_{k|k-1}) < -\pi \\ z_\theta - \theta_{k|k-1}, & \text{en otro caso} \end{cases} \quad (5.54)$$

La escalar de covarianza de innovación  $s_k$  y el vector de ganancia de Kalman  $\vec{K}_k \in \mathbb{R}^{3 \times 1}$  se obtienen mediante las expresiones:

$$s_k = \mathbf{H}_\theta \mathbf{P}_{k|k-1} \mathbf{H}_\theta^T + R$$

$$s_k = \mathbf{P}_{2,2} + R \quad (5.55)$$

$$\begin{aligned} \vec{K}_k &= \frac{1}{s_k} \mathbf{P}_{k|k-1} \mathbf{H}_\theta^T \\ \vec{K}_k &= \frac{1}{\mathbf{P}_{2,2} + R} \begin{bmatrix} \mathbf{P}_{0,2} \\ \mathbf{P}_{1,2} \\ \mathbf{P}_{2,2} \end{bmatrix} \end{aligned} \quad (5.56)$$

La actualización a posterior del estado y de la covarianza adopta la siguiente simplificación:

$$\vec{p}_{k|k} = \vec{p}_{k|k-1} + \vec{K}_k r_\theta \quad (5.57)$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \vec{K}_k \begin{bmatrix} \mathbf{P}_{2,0} & \mathbf{P}_{2,1} & \mathbf{P}_{2,2} \end{bmatrix} \quad (5.58)$$

El parámetro de ruido de medición  $R$  determina el nivel de confianza asignado a cada fuente sensorial. Se asignó un valor de  $R_{IMU} = 0.05 \text{ rad}^2$  para las mediciones de la IMU debido a su alta respuesta dinámica a corto plazo, y un valor superior de  $R_{ODOM} = 1.0 \text{ rad}^2$  para la orientación por odometría, de modo que actúe únicamente como referencia de baja ponderación a largo plazo.

## 5.5 RESUMEN DEL MODELO CINEMÁTICO Y CONTROL

En este capítulo se consolidó la transición entre el modelo conceptual de campos potenciales y su implementación física en la plataforma diferencial mediante la formulación matemática del movimiento y la ley de control digital. A continuación, se resumen los aspectos fundamentales abordados en esta etapa:

- **Modelo cinemático diferencial.** Se formuló el campo de rapidezces del robot en función de sus parámetros geométricos, relacionando la rapidez lineal y angular del chasis con las rapidezces de rotación independientes de las ruedas derechas e izquierdas.
- **Generador de comandos de rapidez.** Se tradujo el vector de fuerza resultante obtenido de los campos potenciales virtuales en referencias directas de rapidez lineal y angular para guiar la trayectoria del vehículo de manera reactiva.

- **Mecanismo de saturación de rapideces.** Se implementó una restricción operativa sobre las señales de control para asegurar que los comandos enviados a los motores permanezcan dentro de los límites mecánicos y eléctricos del hardware.
- **Control digital PID.** Se discretizó la ley de control continua mediante la transformación bilineal, obteniendo la ecuación en diferencias ejecutada en tiempo real por el microcontrolador para el seguimiento de rapidez de los actuadores.
- **Estimación robusta de la pose.** Se estructuró el modelo discreto del sistema mediante la integración de la odometría y la medición por trilateración para mantener una localización y orientación precisas durante sus desplazamientos.

Con la formulación del modelo cinemático y el diseño de la ley de control proporcional, se completó la transición entre el análisis teórico de los campos potenciales y su aplicación física en la plataforma móvil. La transformación del vector de fuerza resultante en comandos de rapidez lineal y angular, sumada a la saturación para evitar demandas fuera de los límites físicos de los motores, garantiza que el robot responda de manera suave y segura ante la presencia de obstáculos. De esta manera, el esquema matemático presentado en este capítulo proporcionó la base para convertir las señales del entorno en movimientos reales sobre la estructura diferencial. Con estos fundamentos teóricos y de control debidamente consolidados, en el siguiente capítulo se abordará la implementación práctica de la arquitectura de hardware y software, detallando los componentes electrónicos, el sistema embebido y el flujo de ejecución que hicieron posible la validación experimental del prototipo.

## **CAPÍTULO 6**

### **DESARROLLO DEL PROTOTIPO FÍSICO**

Este capítulo presenta la implementación física del sistema de navegación autónoma propuesto en los capítulos anteriores. Se describe el desarrollo del prototipo robótico, incluyendo la integración de sensores, la caracterización de los actuadores y la arquitectura del sistema.

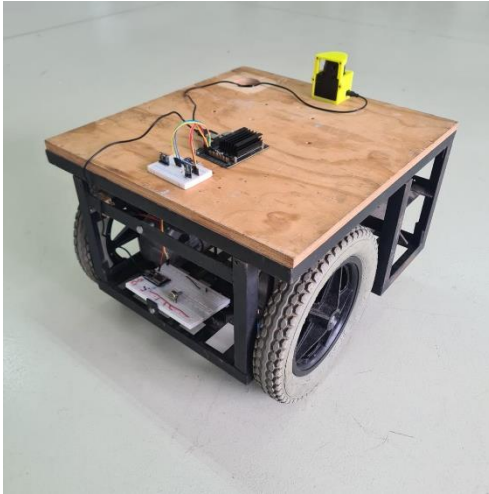
#### **6.1 DESCRIPCIÓN GENERAL DEL ROBOT**

El prototipo utilizado en este trabajo es un robot móvil previamente construido, con estructura robusta y dimensiones aproximadas de 50 cm de largo, 52 cm de ancho y 29 cm de alto. Cuenta con dos actuadores independientes y dos ruedas locas, lo que permite una maniobrabilidad adecuada para pruebas en interiores.

Sobre esta base se integraron los módulos necesarios para la navegación autónoma descrita en esta tesis, incluyendo:

- Sensores de distancia para estimar la posición del robot mediante trilateración
- Un LiDAR para la detección de obstáculos en tiempo real
- Un microcontrolador ESP32 encargado de ejecutar el control PID y la lectura de los sensores
- Módulos de control de potencia para el manejo de los motores.

En la Figura 6.1 se presentan fotografías del prototipo.



(a) Vista espacial



(b) Vista lateral

Figura 6.1 Prototipo robótico.

## 6.2 ARQUITECTURA FUNCIONAL DEL SISTEMA

La arquitectura funcional del sistema se organizó en tres módulos principales: percepción, procesamiento y control. Su diseño está orientado a la navegación autónoma bidimensional en entornos semi controlados.

Para la etapa de percepción se emplearon tres elementos primarios que proporcionaron la información del entorno y del estado del vehículo. En primer lugar, la localización absoluta se obtuvo mediante cuatro módulos RYUW122, tres balizas y una antena, cuya información se recibió a través de un canal serial dedicado en el microcontrolador. Por su parte, la detección de obstáculos se realizó con el sensor LiDAR, el cual se comunicó con la computadora central mediante una interfaz USB. Finalmente, para estimar la orientación angular del robot en tiempo real, se incorporó la unidad de medición inercial MPU9250 conectada al ESP32. A su vez, el microcontrolador se encontraba en constante comunicación con el sistema operativo.

En la etapa de procesamiento, el sistema operativo ejecutó el algoritmo de campos potenciales y el cálculo de las rapidez lineal y angular del robot. A partir de la información sensorial y del destino deseado, la computadora calcula en tiempo real un vector de movimiento que indica la

dirección de giro y la magnitud del avance requerido en cada iteración, y calcula las rapidezces lineales y angulares del robot con base en este vector resultante.

Por último, en la etapa de control, los comandos generados por la unidad de procesamiento se transmitieron hacia el microcontrolador mediante una interfaz USB. El microcontrolador actuó como el enlace directo con la etapa de potencia, generando las señales de modulación por ancho de pulso que recibe el controlador de motores para regular la rapidez y el sentido de giro.

Para cerrar el lazo de control, cada motor dispone de un codificador magnético rotativo que medía la rapidez angular real en las ruedas. Estas mediciones fueron procesadas directamente por el microcontrolador ESP32 para calcular el error de rapidez y ejecutar el algoritmo de control proporcional, integral y derivativo en tiempo real.

Este diseño modular facilitó el desarrollo, mantenimiento y depuración del sistema, al permitir una separación clara de responsabilidades entre los distintos componentes. Asimismo, intenta proporcionar flexibilidad para futuras mejoras, como la sustitución de sensores, el ajuste de algoritmos o la migración hacia plataformas robóticas más complejas.

### 6.3 SELECCIÓN DE SENSORES

La percepción del entorno constituye una etapa indispensable para el funcionamiento autónomo de cualquier plataforma robótica, ya que suministra la información requerida para estimar el estado del vehículo y reconocer el espacio circundante. Con el propósito de alimentar el sistema de navegación reactivo y lograr un seguimiento preciso dentro del espacio de trabajo, se seleccionó e integró un conjunto específico de sensores. En las siguientes subsecciones se presentan las especificaciones técnicas y los criterios de ingeniería que motivaron la elección del sistema de localización absoluta, la unidad inercial encargada del cálculo de la orientación y los sensores dedicados a la detección de obstáculos en tiempo real.

### 6.3.1 Sensor de localización

El sistema de localización del robot se basa en tecnología Ultra-WideBand, UWB, implementada mediante módulos RYUW122, como el mostrado en la Figura 6.2. Esta tecnología se caracteriza por emitir pulsos de muy corta duración en un espectro amplio, mayor a 500 MHz, lo que proporciona alta resolución temporal, robustez frente a interferencias y buen desempeño en entornos con trayectorias indirectas [29], [30].



Figura 6.2 Imagen del módulo RYU2122 [31].

A diferencia de tecnologías como el Wi-Fi, que dependen de la intensidad de señal, metodología conocida como RSSI, para estimar distancias; el UWB utiliza el tiempo de vuelo, ToF por las siglas en inglés de Time of Flight. Esta técnica mide el tiempo que tarda una señal en viajar desde el transmisor hasta el receptor. En este proyecto se utilizó el protocolo conocido como Two-Way Ranging, TWR, donde el robot intercambia señales con tres balizas fijas, lo que permite estimar la distancia a cada una con base en el retardo total medido, ecuación (6.1).

La distancia se calcula aplicando la relación fundamental entre distancia, velocidad y tiempo:

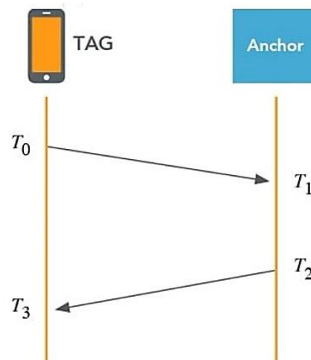


Figura 6.3 Ejemplo de la interacción de la señal en los dispositivos TWR [32].

De la Figura 6.3, se obtienen las expresiones:

$$T_2 - T_1 = \text{Retardo}, \quad T_3 - T_0 = \text{Retorno}, \quad \text{ToF} = \frac{\text{Retorno} - \text{Retardo}}{2}$$

donde finalmente obtenemos la expresión final de la distancia [32].

$$\text{Distancia} = \text{ToF} \cdot c \quad (6.1)$$

donde  $c$  es la velocidad de propagación, aproximadamente la velocidad de la luz. Este principio ha sido ampliamente adoptado en sistemas de posicionamiento UWB por su efectividad [33].

Con tres balizas y una antena, se aplicó el algoritmo de trilateración visto en la sección 4.1 para obtener la posición bidimensional del robot en tiempo real dentro del entorno de operación. Y las coordenadas resultantes del método son las que se utilizaron en la sección 5.4.3 para hacer las correcciones a la odometría a través del filtro de Kalman.

Estudios recientes han demostrado que los sistemas basados en UWB pueden alcanzar precisiones de menos de 10 cm, incluso en entornos interiores con condiciones de línea de vista directa, lo que los convierte en una opción confiable para aplicaciones de robótica móvil [30].

### 6.3.2 Sensor de orientación

Para estimar la orientación del robot en el plano de navegación se integró un sensor inercial, el módulo MPU9250, mostrado en la Figura 6.4. Este dispositivo es una unidad de medición inercial o Inertial Measurement Unit por las siglas en inglés de IMU, que integra acelerómetro, giroscopio y magnetómetro triaxiales [34].

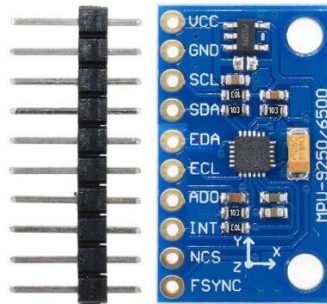


Figura 6.4 Sensor inercial MPU9250 con comunicación I2C [34].

El MPU9250 se comunica con el microcontrolador mediante el protocolo I2C, con el uso de las líneas SDA y SCL para la transmisión de datos. El magnetómetro interno opera con una resolución de 16 bits y rangos de medición configurables, los cuales permiten capturar variaciones del campo magnético con la precisión necesaria para aplicaciones de orientación.

Dado que el robot se desplaza exclusivamente sobre un plano bidimensional, la orientación del sistema móvil  $\{R\}$  con respecto al marco de referencia global es suficiente para caracterizar su pose en tareas de navegación planar.

En este trabajo, la estimación del ángulo de orientación se obtuvo a partir de la integración de la rapidez angular medida por el giroscopio. Si  $\omega(t)$  representa dicha rapidez angular, la orientación se calcula como:

$$\theta(t) = \theta(t_0) + \int_{t_0}^t \omega(t) dt \quad (6.2)$$

donde  $\theta(t_0)$  es la orientación del robot en el instante cero. En implementación discreta, esta expresión se aproxima mediante integración numérica:

$$\theta_k = \theta_{k-1} + \omega \Delta t \quad (6.3)$$

Este enfoque permitió estimar la orientación relativa del robot, es decir, el ángulo acumulado de giro respecto a una referencia inicial. Y esta lectura es la que se utilizó en la sección 5.4.4, para corregir la predicción de la odometría en el filtro de Kalman.

Es importante destacar que, aunque el módulo cuenta con magnetómetro y este permite obtener una referencia absoluta respecto al campo magnético terrestre, en entornos con alta interferencia electromagnética, como los que generaron los motores de alta corriente del robot, las lecturas del sensor no fueron lo suficientemente estables para utilizarlas como una referencia válida. Por esta razón, se decidió utilizar únicamente el giroscopio para el cálculo de la orientación.

### 6.3.3 Sensores de obstáculos

Para la detección de obstáculos en tiempo real se utilizó un sensor LiDAR, una tecnología de teledetección activa que permite reconstruir con alta precisión la geometría del entorno mediante

pulsos de luz láser. A diferencia de tecnologías como radar o sonar, que emplean microondas u ondas acústicas, respectivamente, el LiDAR emite haces de luz que, al reflejarse en los objetos, permiten calcular la distancia [35].

En este proyecto se utilizó el modelo URG-04LX-UG01, mostrado en la Figura 6.5, de la empresa Hokuyo, un escáner láser bidimensional ampliamente utilizado en aplicaciones de percepción ambiental en robótica móvil. El sensor cuenta con un campo de visión horizontal de  $240^\circ$ , una resolución angular de  $0.36^\circ$ , un rango de detección de hasta 4 metros, y una frecuencia de escaneo configurable de hasta 10 Hz. Opera con una longitud de onda de 785 nm, y proporciona una densidad de puntos adecuada para detectar obstáculos en tiempo real dentro del plano horizontal [36].

La comunicación entre el sensor y la unidad de procesamiento se realiza mediante una interfaz serial USB, y los datos se procesaron en un sistema operativo conocido como ROS2, por sus siglas en inglés de Robot Operating System, para identificar obstáculos en tiempo real. Este tipo de sensor es especialmente útil en entornos interiores con obstáculos no estructurados, gracias a su precisión angular, bajo consumo energético y compatibilidad con herramientas como ROS2.



Figura 6.5 Sensor LiDAR Hokuyo URG-04LX-UG01 [36].

## 6.4 ACTUADORES

Los actuadores del prototipo corresponden a dos motores de corriente directa, CD, con escobillas, originalmente diseñados para aplicaciones de movilidad en silla de ruedas. Estos motores accionan de manera independiente cada una de las ruedas del robot móvil, permitiendo la generación de movimientos de traslación y rotación mediante el control digital de las rapideces angulares de cada uno.

Dado que los motores empleados no cuentan con una hoja de especificaciones proporcionada por el fabricante, fue necesario realizar una caracterización experimental con el objetivo de obtener un modelo dinámico que permitiera el diseño e implementación de estrategias de control. Dicha caracterización se llevó a cabo mediante técnicas de identificación de sistemas, a partir de datos experimentales obtenidos bajo pruebas de entrada tipo escalón y rampa, aplicadas de forma independiente a cada motor.

La señal de entrada correspondió al ciclo de trabajo de la señal de modulación de ancho de pulso aplicada al controlador de potencia del motor, mientras que la salida medida es la rapidez angular del eje, expresada en rad/s. La adquisición, procesamiento de datos e identificación de modelos dinámicos se realizaron utilizando el entorno MATLAB [24], con el empleo de las herramientas del System Identification Toolbox [37].

### 6.4.1 Identificación del modelo dinámico

A partir de los datos experimentales obtenidos, se evaluaron distintos modelos dinámicos de tiempo continuo, incluyendo modelos de primer orden y segundo orden. El criterio de selección del modelo se basó en el porcentaje de ajuste, el cual cuantifica la correspondencia entre la respuesta del modelo identificado y los datos experimentales medidos.

Para ambos motores, el modelo que presentó el mejor comportamiento entre simplicidad y capacidad de representación fue un modelo de primer orden, que alcanzó un ajuste cercano al 80%. El modelo dinámico identificado puede expresarse de forma general como:

$$G(s) = \frac{K}{s + a} \quad (6.4)$$

donde  $K$  representa la ganancia del sistema y  $a$  está relacionado con la constante de tiempo del motor.

### 6.4.2 Resultados de la identificación

Para el motor izquierdo se obtuvo el siguiente modelo dinámico:

$$G_{izq}(s) = \frac{45.91}{s + 5.699} \quad (6.5)$$

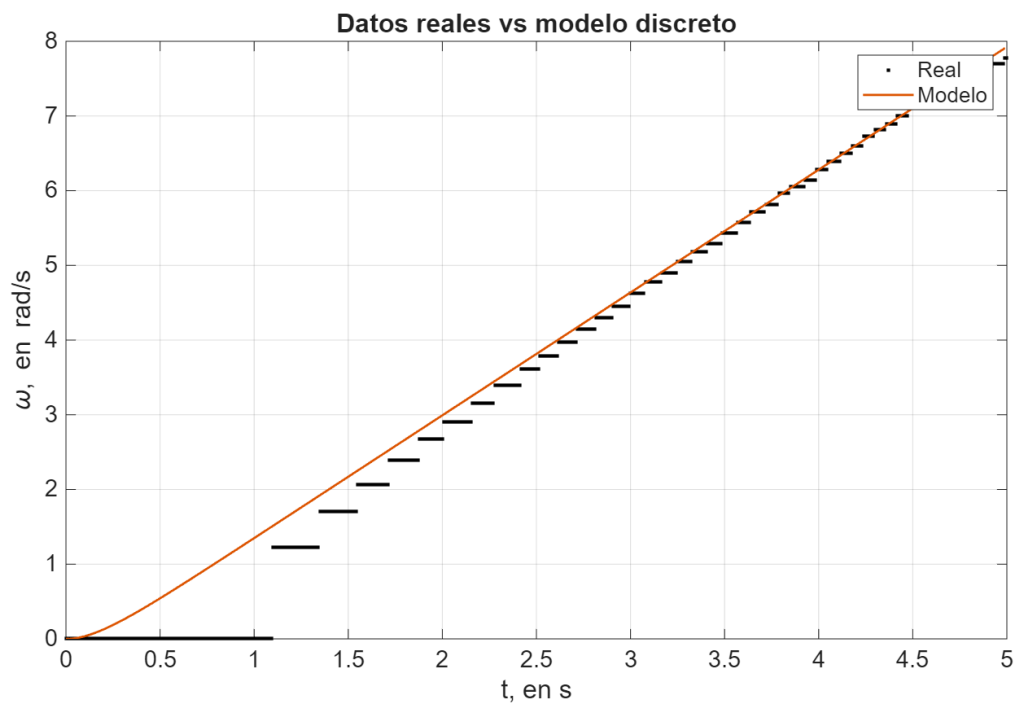
con un porcentaje de ajuste del 80.5% y una ganancia estática estimada de 8.056 rad/s por unidad de señal de control. La varianza del ruido estimada para este modelo fue de 0.0034, lo que indica una adecuada calidad de los datos experimentales.

Para el motor derecho, el modelo identificado fue:

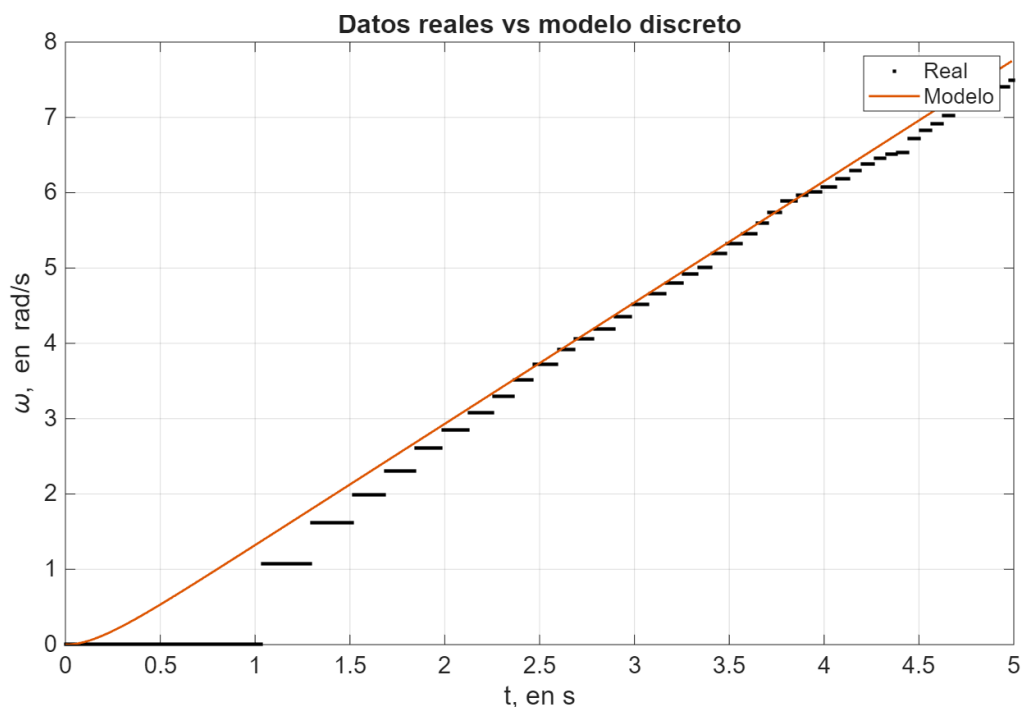
$$G_{der}(s) = \frac{46.94}{s + 5.708} \quad (6.6)$$

con un ajuste del 80.1% y una ganancia estática estimada de 8.222 rad/s por unidad de señal de control. En este caso, la varianza del ruido fue de 0.0036. En la Figura 6.6 se muestra el comportamiento de los modelos contra los datos leídos de los motores.

Los resultados obtenidos, muestran que ambos motores presentan dinámicas muy similares, lo cual resulta conveniente para un robot móvil, ya que facilita el diseño de estrategias de control y reduce errores asociados a diferencias estructurales y de comportamiento entre los actuadores.



(a) Motor izquierdo



(b) Motor derecho

Figura 6.6 Comparación entre los datos leídos y el modelo obtenido.

### 6.4.3 Diseño del controlador de rapidez

Con base en los modelos dinámicos identificados, se diseñó un controlador PID discreto para cada motor, considerando un tiempo de muestreo de  $T_s = 0.01$  s. El objetivo del controlador es regular la rapidez angular de cada rueda, que garantice el seguimiento adecuado de las referencias de rapidez generadas por los comandos del sistema operativo.

Durante el proceso de diseño se observó que el término integral fue el único necesario para alcanzar el desempeño deseado del sistema, por lo que el controlador implementado corresponde a una estructura integral pura. Para el motor derecho se obtuvo el siguiente parámetro:

$$K_{i_{der}} = 0.257737 \quad (6.7)$$

Mientras que para el motor izquierdo se obtuvo un valor similar:

$$K_{i_{izq}} = 0.263106 \quad (6.8)$$

La implementación discreta del controlador se realizó mediante la transformación bilineal, la cual fue posteriormente programada en el sistema embebido para el control en tiempo real de la rapidez de los motores. En las figuras 6.7 y 6.8 se puede observar las respuestas del control.

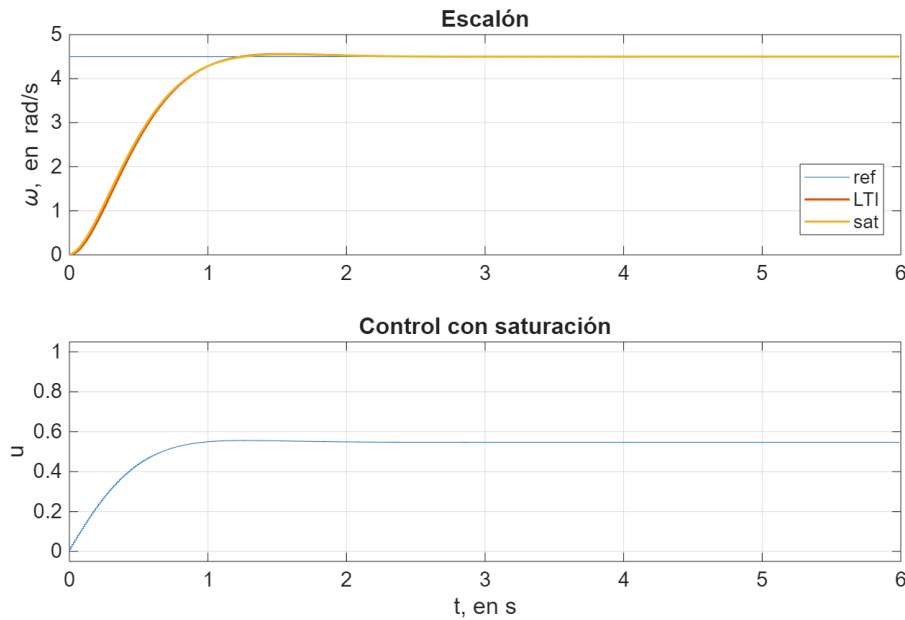


Figura 6.7 Gráfica del comportamiento del control del motor izquierdo.

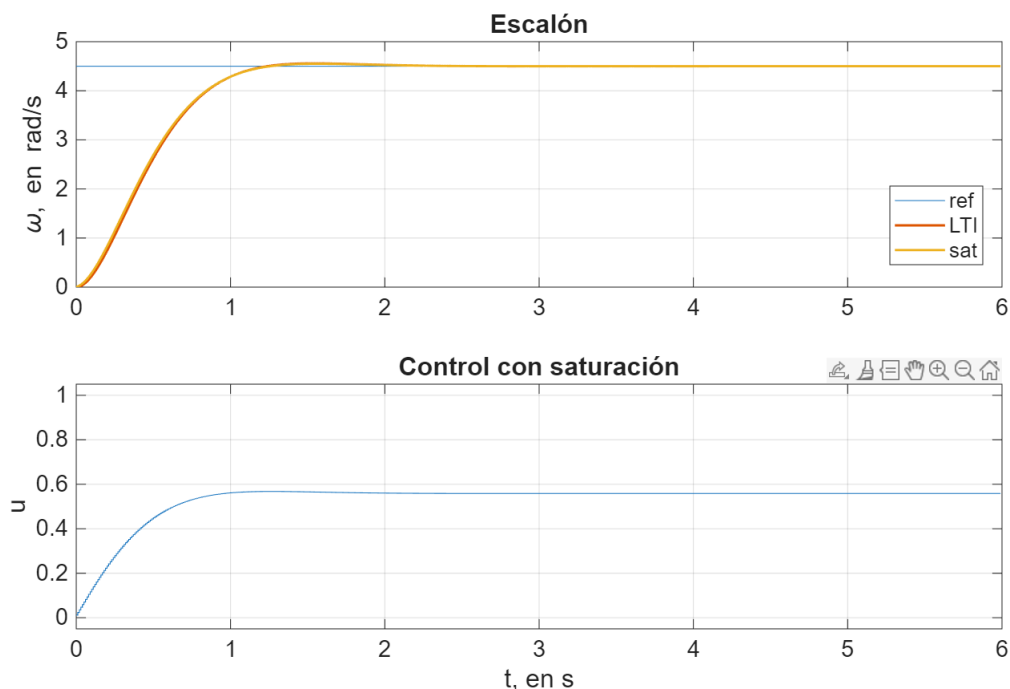


Figura 6.8 Gráfica del comportamiento del control del motor derecho.

## 6.5 CODIFICADORES ROTATORIOS

La medición de la rapidez angular de las ruedas del robot se realizó mediante codificadores rotatorios incrementales, , mejor conocidos como encoders, de fabricación propia. Estos fueron fabricados específicamente para las necesidades de este prototipo. Esta decisión fue tomada debido a las complicaciones que implicaban acceder al eje del motor, a la compleja geometría del chasis del robot, y por la necesidad de integrar una solución de bajo costo y alta flexibilidad.

Cada encoder está compuesto por sensores de efecto Hall digitales tipo KY-003 [38], como los mostrados en la Figura 6.9, y un conjunto de imanes permanentes fijados al perímetro de la rueda. El principio de funcionamiento se basa en la detección del paso de cada imán frente al sensor Hall, generando una señal digital que es interpretada como un pulso incremental. Para cada revolución completa de la rueda se obtienen  $N = 15$  pulsos, los cuales dividen la circunferencia en sectores angulares discretos.

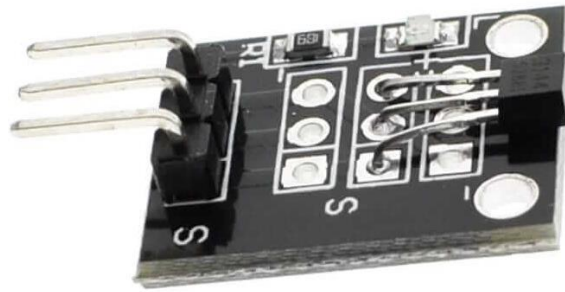


Figura 6.9 Sensor de efecto Hall KY-003 [38].

### 6.5.1 Problemas experimentales y mitigación de ruido

Durante las primeras pruebas experimentales se identificaron múltiples contratiempos asociados a la naturaleza condicionada del sistema de medición de la rapidez. El primero de ellos fue la presencia de ruido y pulsos espurios en la señal digital del sensor Hall, provocados principalmente por rebotes eléctricos y transiciones rápidas en el frente de la señal.

Se implementó una estrategia de filtrado en dos etapas. En el dominio analógico, se incorporó un filtro pasa bajas RC con una resistencia de  $4.7\text{ k}\Omega$  y un capacitor electrolítico de  $10\text{ nF}$ , lo que estableció una frecuencia de corte aproximada de  $3.40\text{ kHz}$  para atenuar las interferencias de alta frecuencia. Para reforzar, en el dominio digital, se utilizó el periférico de conteo de pulsos integrado en el microcontrolador ESP32 [39], configurando un filtro con una ventana temporal de  $10\text{ }\mu\text{s}$ . Este filtro permite descartar pulsos cuya duración sea inferior a dicho umbral, eliminando de forma efectiva eventos espurios sin afectar la detección válida de pulsos.

### 6.5.2 Error sistemático por distribución no uniforme de imanes

Un problema adicional, de mayor impacto en la precisión de la medición, fue la distribución no perfectamente equidistante de los imanes en el perímetro de la rueda. A pesar de que no se presentaban pulsos fantasmas tras la implementación del filtrado, la separación angular irregular entre imanes provocaba que los intervalos temporales entre pulsos consecutivos no fueran uniformes, incluso a rapidez constante.

Este efecto introduce un error sistemático en el cálculo de la rapidez angular, ya que el método clásico asume sectores angulares idénticos. En consecuencia, la rapidez calculada presentaba oscilaciones periódicas sincronizadas con la posición angular de la rueda.

### 6.5.3 Medición basada en intervalos temporales

La rapidez angular se estima a partir del tiempo transcurrido entre pulsos consecutivos. Sea  $\Delta t_i$  el intervalo de tiempo medido correspondiente al sector  $i$ , expresado en segundos. La rapidez angular sin corrección se calcula como:

$$\omega = \frac{2\pi}{N\Delta t_i} \quad (6.9)$$

donde  $N$  es el número de pulsos por revolución. Esta formulación asume implícitamente que todos los sectores tienen el mismo ancho angular, lo cual no se cumple en el sistema físico implementado.

### 6.5.4 Calibración y generación de la tabla de corrección

Para compensar el error introducido por la distribución irregular de los imanes, se implementó un proceso de calibración experimental basado en múltiples revoluciones completas de la rueda a rapidez constante. Durante la calibración, se registraron los intervalos temporales  $\Delta t_{sector,j}$  correspondientes a cada sector en la revolución  $j$ , con un máximo de  $M_{rev} = 15$  revoluciones. Para cada sector, se calculó el valor promedio eliminando valores atípicos mediante la exclusión del mínimo y máximo registrados:

$$\Delta t_{sector} = \frac{1}{M_{muestras} - 2} \sum_{j \in \mathcal{J}_{sector}} \Delta t_{sector,j} \quad (6.10)$$

donde  $\mathcal{J}_{sector}$  representa el conjunto de muestras válidas para el sector evaluado y  $M_{muestras}$  denota el número total de muestras no nulas. Posteriormente, se calculó el promedio global considerando la suma de los promedios de todos los sectores:

$$\Delta t_{global} = \frac{1}{N_{sectores}} \sum_{sector=1}^{N_{sectores}} \Delta t_{sector} \quad (6.11)$$

A partir de este valor se obtuvo un factor de corrección porcentual para cada sector:

$$C_i = \frac{\Delta t_{sector}}{\Delta t_{global}} \quad (6.12)$$

Estos factores de corrección conforman una tabla de consulta, desde ahora conocida como LUT, la cual se utilizó para compensar en tiempo real los intervalos medidos.

### 6.5.5 Corrección en tiempo real y cálculo final de velocidad

Durante la operación normal del sistema, el intervalo temporal medido  $\Delta t_i$  fue corregido utilizando el factor correspondiente al sector actual:

$$\Delta t_{sector}^{corr} = \frac{\Delta t_{sector}}{C_i} \quad (6.13)$$

La velocidad angular corregida se obtuvo entonces como:

$$\omega = \frac{2\pi}{N \Delta t_{sector}^{corr}} \quad (6.14)$$

Este procedimiento permitió eliminar las oscilaciones periódicas en la estimación de velocidad y proporciona una medición significativamente más estable y precisa.

### 6.5.6 Persistencia de la calibración

Con el fin de evitar la necesidad de recalibrar el sistema en cada reinicio del microcontrolador, la tabla de corrección se almacenó en la memoria no volátil, conocida como NVS, del ESP32 [40]. Se definió una tabla independiente para cada sentido de giro por rueda, lo cual permitió compensar las asimetrías mecánicas al invertir la dirección de rotación.

De esta manera, el sistema de encoders desarrollado proporcionó una solución robusta, precisa y completamente integrada al sistema embebido, adecuada para aplicaciones de control de velocidad en robots móviles diferenciales.

## 6.6 SELECCIÓN DEL MICROCONTROLADOR

Por lineamientos del proyecto, se integró como plataforma base una ESP32 DevKit v4 [41], encargada de realizar la adquisición de datos de los encoders, generación de señales de modulación de ancho de pulso, o mejor conocida como PWM por sus siglas en inglés, y control directo de los motores. Este microcontrolador cuenta con doble núcleo, conectividad Wi-Fi y Bluetooth, y una arquitectura adecuada para sistemas embebidos.

Dadas las necesidades del proyecto, se necesitó una herramienta capaz de comunicar diferentes flujos de información entre sí, por lo tanto, se decidió usar el sistema operativo ROS2 para la programación del algoritmo de navegación.

## 6.7 PROGRAMACIÓN DEL SISTEMA DE NAVEGACIÓN

El software del robot fue dividido en dos niveles funcionales:

- **Nivel bajo.** Con la ESP32, fue el encargado de la adquisición de datos de encoders, del control PID de velocidad de cada una de las ruedas y encargado de aproximar la pose con el EKF, los UWB y la IMU.
- **Nivel alto.** Con ROS2, fue el responsable de la ejecución del algoritmo de navegación de campos potenciales, el cálculo del campo atractivo, repulsivo, etc. Además del cálculo de la velocidad lineal y angular del prototipo y recibir la pose e información de las ruedas.

## **CAPÍTULO 7**

### **RESULTADOS**

Una vez concluida la implementación del sistema de navegación descrita en el capítulo anterior, se procede a evaluar de forma integral el desempeño del prototipo. En este capítulo se presentan y analizan los resultados experimentales obtenidos en dos grandes bloques. En primer lugar, se valida el funcionamiento del sistema embebido y del control de bajo nivel, abarcando la corrección de errores sistemáticos en los encoders mediante la tabla de consulta, la respuesta temporal del controlador PID, la caracterización del sistema de localización absoluta por UWB y la fusión sensorial realizada por el filtro de Kalman. En segundo lugar, se examina el algoritmo de navegación por campos potenciales artificiales, estudiando la evolución de la trayectoria, la respuesta del campo ante obstáculos y las principales limitaciones observadas durante las pruebas en un entorno controlado. Los resultados que se exponen a continuación permiten contrastar el comportamiento real del robot con las predicciones teóricas de los capítulos previos.

#### **7.1 VALIDACIÓN DEL SISTEMA EMBEBIDO Y CONTROL**

La validación del sistema embebido y de control de bajo nivel constituye una etapa fundamental para garantizar la estabilidad y la precisión en la ejecución de las trayectorias del robot móvil. En esta primera sección se evalúa el desempeño experimental de los componentes de adquisición

y actuación que integraron la plataforma física. Inicialmente se caracterizó la respuesta de los encoders y se evaluó el esquema de corrección de errores sistemáticos, explicado en la sección 6.5. Posteriormente se analizó la respuesta temporal del controlador digital sintonizado para regular la rapidez de las ruedas. Asimismo, se determinó la precisión del módulo de localización absoluta por radiofrecuencia y se comprobó la capacidad del filtro de Kalman para fusionar las lecturas sensoriales en tiempo real.

### 7.1.1 Corrección de errores sistemáticos en encoders

Uno de los mayores desafíos experimentales fue la distribución no uniforme de los imanes en las ruedas, lo que generaba oscilaciones indeseadas en la lectura de velocidad angular  $\omega$ . Para corregir esto, se implementó una tabla de corrección basada en la caracterización de los 15 sectores de cada rueda. La Figura 7.1 muestra una comparativa de las mediciones obtenidas con la LUT desactivada frente a la LUT activada, operando a un ciclo de trabajo constante.

$$Error\ Relativo = \frac{|\omega_{medida} - \omega_{promedio}|}{\omega_{promedio}} \times 100\%$$

Como se observa en los datos, sin la corrección de la LUT, el sistema reportó variaciones de rapidez de hasta 0.88 rad/s entre sectores, de 2.85 a 3.73 rad/s, a pesar de que el motor mantiene una rapidez física constante. Esto representa un error de medición intrínseco de aproximadamente el 13.5%.

Al activar la corrección mediante la LUT, el algoritmo compensó el ancho temporal de cada sector. Aunque los intervalos de tiempo seguían siendo físicamente distintos debido a la posición de los imanes, la rapidez calculada se estabilizó en un rango de 3.24 a 3.29 rad/s. Esto redujo la varianza del error de medición a menos del 1%, proporcionando una señal de retroalimentación limpia para el controlador PID.

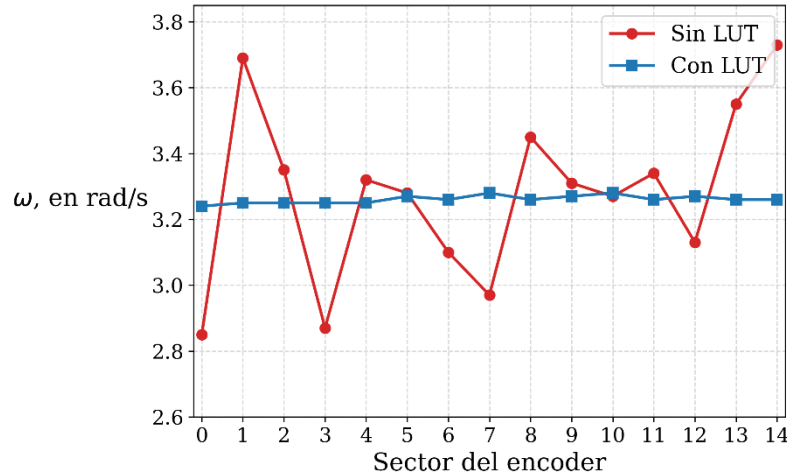


Figura 7.1 Comparación de la medición de velocidades angulares con corrección LUT.

### 7.1.2 Análisis de la respuesta temporal y desempeño del controlador PID

El control de rapidez de bajo nivel se manejó por comunicación serial, indicando comandos de rapidez de referencia lineal y angular respecto al centro del robot. Por ejemplo, si se recibía uno con  $v_C = 0.3 \frac{m}{s}$  y  $\omega = 0.0 \frac{rad}{s}$ , esto se traduciría a una rapidez de referencia de ambas ruedas de  $\omega_{ref} = 1.9354 \frac{rad}{s}$  por la ecuación (5.12).

Aunque el diseño teórico del controlador PID se realizó bajo un factor de amortiguamiento crítico, con  $\zeta \approx 1$  para eliminar el sobrepaso, los resultados experimentales mostraron una dinámica distinta debido a las condiciones físicas del entorno y del hardware.

#### Caracterización de la respuesta dinámica

A partir de los datos de telemetría obtenidos, se identificaron los siguientes parámetros:

- **Tiempo de levantamiento.** Tiempo en el que sistema rompe la inercia inicial y empieza el movimiento de la rueda, esta medición fue de 0.9 s, en promedio
- **Sobrepaso experimental.** Se registró un pico máximo de 2.43 rad/s, lo que representó un sobreimpulso del 25.5%
- **Error de estado estacionario.** Una vez estabilizado, el error es inferior al 0.2%, lo que garantizó que la rapidez del robot fuera fiel a lo solicitado por el planificador

### Discrepancia entre el modelo y la implementación real

La aparición de un sobrepaso del 25.5% en un diseño originalmente calculado para ser sobreamortiguado se atribuye a factores no lineales no considerados en el modelo ideal:

- **Fricción Estática.** El controlador acumulaba un esfuerzo de control considerable para vencer la resistencia inicial del motor y la transmisión. Al romperse la fricción estática, la energía acumulada se liberaba bruscamente, provocando el sobrepaso antes de que el control pudiera compensarlo.
- **Cuantización y Latencia.** El tiempo de muestreo de 100ms y la resolución de los encoders, inevitablemente introdujeron una acumulación en el valor de la señal de error.

### Robustez ante perturbaciones externas

Se observa en los registros, ilustrados en la Figura 7.2, variaciones súbitas en la velocidad de las ruedas que alcanzaron los  $4.16 \text{ rad/s}$ . Estas perturbaciones fueron producto de irregularidades en la superficie del ambiente de trabajo. Sin embargo, la capacidad del controlador para volver a estabilizar la velocidad en menos de 0.8 s demostró una robustez adecuada para la navegación en entornos interiores, donde el deslizamiento y las irregularidades del suelo son comunes.

### Resumen del desempeño del controlador

A partir de los resultados experimentales obtenidos de la telemetría del sistema, se observó que el controlador PID implementado presentaba un comportamiento subamortiguado, en contraste con el diseño teórico originalmente planteado. Esta diferencia se refleja en el sobrepaso medido, el incremento en el tiempo de asentamiento y la presencia de perturbaciones durante la etapa transitoria, como se aprecia en la Figura 7.2.

De manera cuantitativa, estos resultados se resumen en la Tabla 7.1, donde se comparan los parámetros de desempeño teóricos contra los valores obtenidos experimentalmente. En conjunto, tanto la respuesta temporal como los indicadores de error confirman que, aunque el sistema mantiene estabilidad y capacidad de seguimiento de referencia, su desempeño está fuertemente influenciado por dinámicas no modeladas del sistema físico.

Tabla 7.1 Comparación entre el diseño teórico del controlador PID y los resultados experimentales obtenidos.

| Parámetro              | Diseño Teórico           | Resultado Experimental |
|------------------------|--------------------------|------------------------|
| Tipo de Respuesta      | Críticamente Amortiguada | Subamortiguada         |
| Sobreimpulso           | 0%                       | 25.5%                  |
| Error Estacionario     | 0%                       | 0.13%                  |
| Tiempo de Asentamiento | 1.2 s                    | 1.8 s                  |

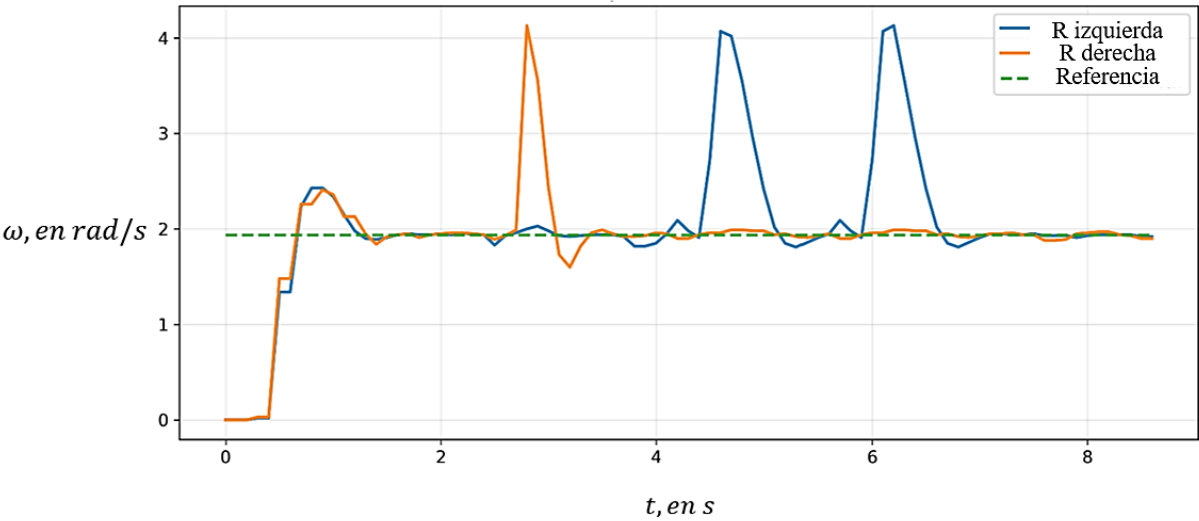


Figura 7.2 Respuesta del controlador PID ante referencia de velocidad.

7.1.3 Caracterización del sistema de localización absoluta

En esta sección se evalúa el desempeño del módulo de medición Ultra-WideBand y el algoritmo de trilateración implementado en la tarea asíncrona UWBMeasureTask del microcontrolador. El objetivo fue determinar la precisión y estabilidad de las coordenadas  $x$  e  $y$  obtenidas.

Análisis de la tasa de actualización

A partir de los registros temporales del sistema, se observó que la tarea de medición UWB operaba a una frecuencia significativamente menor que el lazo de control. Los intervalos entre mediciones exitosas presentan la siguiente dinámica:

$$\Delta t_1 = 6425 - 5225$$

$$\Delta t_1 = 1200 \text{ ms}$$

$$\Delta t_2 = 7605 - 6425$$

$$\Delta t_2 = 1280 \text{ ms}$$

$$\Delta t_3 = 8805 - 7605$$

$$\Delta t_3 = 1200 \text{ ms}$$

Se calcula una tasa de actualización promedio de  $0.83 \text{ Hz}$ . Se decidió establecer esta frecuencia para evitar saturar la comunicación entre las balizas y la antena del robot, además de que la predicción temprana del filtro de Kalman se basó en la odometría.

### Estabilidad y dispersión de la localización

Al analizar las coordenadas reportadas en un entorno estático, el sistema mostró una fluctuación en las lecturas que caracteriza el ruido del canal de UWB. Tomando como referencia un punto de prueba, se obtuvieron los siguientes valores:

*Tabla 7.2 Variación de las coordenadas UWB en un punto estático*

| Tiempo, en ms | X, en m | Y, en m | $\Delta X$ , en m | $\Delta Y$ , en m |
|---------------|---------|---------|-------------------|-------------------|
| 5225          | 1.550   | 2.272   |                   |                   |
| 6426          | 1.551   | 2.306   | +0.001            | +0.034            |
| 36265         | 1.539   | 2.289   | -0.012            | -0.017            |
| 37495         | 1.572   | 2.272   | +0.033            | -0.017            |

La desviación estándar observada se mantuvo en el rango de  $\pm 3$  a  $\pm 5 \text{ cm}$ , lo cual validó la eficacia del algoritmo de trilateración para aplicaciones de navegación en interiores.

### Identificación de mediciones erróneas

Durante las pruebas experimentales, se detectaron saltos bruscos en la posición estimada, como se mostró en el registro de tiempo  $t = 24315 \text{ ms}$ , con  $(X, Y) = (0.915, 2.742)$ . Estos errores son producto de:

- **Efecto multitrayectoria.** Reflexiones de la señal UWB en superficies metálicas o paredes que falsean la medición de distancia.
- **Oclusión momentánea.** Bloqueo parcial de la línea de vista entre la antena y alguna de las balizas, incrementando el tiempo de vuelo de los paquetes de datos.

Estos datos crudos confirmaron que, si bien el sistema UWB proporciona una referencia global excelente para eliminar la deriva, requiere de un procesamiento posterior para filtrar las señales espurias antes de ser utilizadas en el guiado del robot.

### Resumen del desempeño de UWB

Con la finalidad de evaluar el comportamiento del sistema de localización absoluta, se evaluó el comportamiento del programa y se registraron las lecturas del sistema de alcance por ultra ancho de banda y se calcularon los errores de posicionamiento con respecto a las posiciones reales del prototipo. Dichas mediciones permitieron cuantificar la precisión, la varianza y los tiempos de respuesta del algoritmo de trilateración frente a diferentes condiciones dentro del espacio de trabajo. Los resultados consolidados de este proceso de caracterización se presentan a continuación en la Tabla 7.3, donde se resumen las métricas fundamentales que demuestran el desempeño global de esta etapa sensorial.

*Tabla 7.3 Resumen de desempeño del sistema UWB*

| Métrica                 | Valor resultante                      |
|-------------------------|---------------------------------------|
| Frecuencia de muestreo  | 0.83 Hz                               |
| Precisión promedio en X | $1.548 \text{ m} \pm 0.025 \text{ m}$ |
| Precisión promedio en Y | $2.276 \text{ m} \pm 0.031 \text{ m}$ |
| Error máximo por rebote | 0.62 m                                |

### 7.1.4 Fusión sensorial mediante filtro de Kalman extendido

Como se describió en la sección 5.4, la estimación robusta de la pose del robot requirió integrar de manera coherente las mediciones provenientes de diferentes sensores, considerando explícitamente sus incertidumbres. En esta sección se presentan los resultados experimentales del filtro de Kalman extendido implementado en el sistema embebido, el cual fusiona la odometría, con lo que se hizo la predicción de la pose, y las mediciones de orientación del giroscopio, medidas por la IMU, junto con las posiciones absolutas obtenidas por UWB cuya función fue corregir las predicciones realizadas por la odometría.

#### Comportamiento en condiciones estáticas

Para evaluar la estabilidad del filtro en ausencia de movimiento, se analizó un periodo de aproximadamente 15 segundos en el que el robot permaneció en reposo. Durante este intervalo, la tarea EKFTask actualizó la estimación de la pose a una frecuencia aproximada de 20 Hz, mientras que las mediciones UWB llegaron a una tasa mucho menor, aproximadamente de 0.83 Hz, y las lecturas del sensor inercial se recibieron cada 30 ms. La Tabla 7.4 resume los valores reportados por el filtro durante un lapso representativo en estado estático, antes de recibir cualquier comando de movimiento.

*Tabla 7.4 Estabilidad de la estimación EKF en reposo.*

| Tiempo, en ms | Fuente         | X, en m | Y, en m | Theta, en ° |
|---------------|----------------|---------|---------|-------------|
| 4835          | EKFTask        | 1.504   | 2.310   | 0.0         |
| 4885          | IMU            | ---     | ---     | 0.00        |
| 4935          | EKFTask        | 1.504   | 2.310   | 0.0         |
| 5215          | UWBMeasureTask | 1.529   | 2.287   | ---         |
| 5235          | EKFTask        | 1.504   | 2.310   | 0.0         |
| 5455          | IMU            | ---     | ---     | 0.00        |
| 6435          | EKFTask        | 1.516   | 2.278   | 0.0         |
| 7605          | UWBMeasureTask | 1.508   | 2.359   | ---         |
| 9985          | UWBMeasureTask | 1.510   | 2.252   | ---         |
| 10035         | EKFTask        | 1.519   | 2.273   | 0.0         |

Se observó que, a pesar de las fluctuaciones en las mediciones crudas de UWB, por ejemplo, los saltos de hasta 6 cm en  $x$  y 7 cm en  $y$  entre muestras consecutivas, la salida del filtro

permanece prácticamente constante en  $x = 1.506 \pm 0.012 \text{ m}$ ,  $y = 2.315 \pm 0.037 \text{ m}$ . La orientación estimada se mantiene en  $\theta = 0.0^\circ$  con una desviación inferior a  $0.1^\circ$ , a pesar de que las mediciones crudas del giroscopio presentan pequeñas derivas y el ángulo reportado por la IMU, de 174 a 179 grados, es físicamente irrelevante para la navegación planar, dadas las dimensiones del robot.

### Corrección de la deriva de orientación

Durante la tarea de desplazamiento, la estimación de la orientación evidenció el comportamiento dinámico del robot a lo largo de la trayectoria y la contribución del algoritmo de filtrado en la mitigación de la deriva sensorial. Dado el comando de referencia:  $v_c = 0.3 \frac{\text{m}}{\text{s}}$  y  $\omega = 0 \frac{\text{rad}}{\text{s}}$ , la Figura 7.3 ilustra la evolución del ángulo de orientación en función del tiempo para la señal obtenida mediante odometría pura y para la señal procesada por el filtro de Kalman. La trayectoria descrita presenta variaciones angulares inherentes a la maniobra realizada. La incorporación de las mediciones de la unidad de medición inercial mediante la etapa de corrección permitió atenuar las fluctuaciones abruptas y suavizar los picos sin alterar la tendencia real del movimiento. Esta acción de fusión reduce el ruido, atenúa los errores acumulativos de la odometría y proporciona una estimación de la orientación global más estable y consistente durante todo el recorrido.

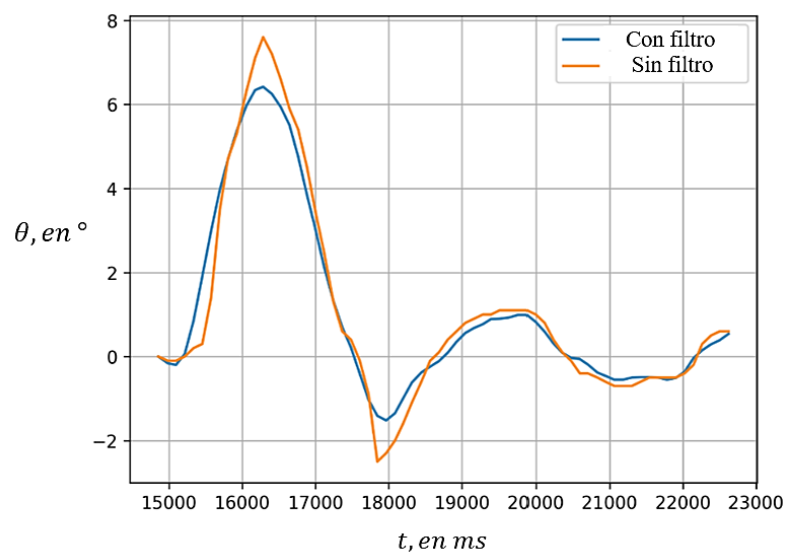


Figura 7.3 Comparación de la orientación estimado por EKF y odometría durante movimiento rectilíneo

### Comparación entre estimación EKF y medición UWB cruda

La Tabla 7.5 compara las posiciones estimadas por el EKF con las mediciones UWB disponibles en momentos cercanos durante la misma prueba dinámica. Se observó que el filtro suaviza los saltos bruscos y las mediciones atípicas del UWB, al tiempo que corrige la deriva de la odometría cuando la referencia absoluta está disponible.

*Tabla 7.5 Comparación EKF contra UWB durante el movimiento.*

| Tiempo, en ms | UWBMeasureTask (x,y), en m | EKFTask (x,y), en m | Diferencia, en m |
|---------------|----------------------------|---------------------|------------------|
| 15935         | (1.522, 2.306)             | (1.724, 2.319)      | 0.202            |
| 17125         | (1.851, 2.340)             | (2.156, 2.365)      | 0.305            |
| 18325         | (2.298, 2.310)             | (2.547, 2.333)      | 0.249            |
| 19505         | (2.523, 2.479)             | (2.869, 2.412)      | 0.351            |
| 20705         | (2.958, 2.460)             | (3.238, 2.440)      | 0.280            |
| 21885         | (3.349, 2.707)             | (3.611, 2.579)      | 0.296            |

La diferencia promedio entre la posición estimada y la medición UWB fue de  $0.28\text{ m}$ , valor consistente con la precisión del sistema UWB en condiciones de línea de vista parcial. El EKF no sigue ciegamente las mediciones UWB, sino que combina la predicción de la odometría con la corrección absoluta, logrando una trayectoria suave y físicamente consistente.

### Comportamiento durante la detención y estabilización de la pose

Una vez finalizado el desplazamiento rectilíneo, el sistema recibió el comando de paro en el instante  $t = 22735\text{ ms}$ . La Tabla 7.6 muestra la evolución de la estimación EKF durante los segundos posteriores a la detención. Inmediatamente después de detenerse, la orientación disminuyó gradualmente desde  $0.8^\circ$  hasta estabilizarse en  $0.2^\circ$  en aproximadamente 0.5 segundos, reflejando la corrección de la pequeña deriva remanente del giroscopio. La posición se mantiene estable en (3.895, 2.582) durante unos 500 ms; el salto aparente en  $t = 23335\text{ ms}$  a (4.059, 2.699) no corresponde a un movimiento real, sino a una actualización de la predicción por odometría mientras el controlador aún ajustaba los motores. La brusca corrección posterior, nótese la medición UWB en  $t = 23065\text{ ms}$  con (3.724, 2.583), muy alejada, evidencia la necesidad de un filtro robusto: el EKF no acepta el valor anómalo de inmediato, sino que transita suavemente hasta la nueva actualización del sistema UWB.

Tabla 7.6 Estabilización de la pose tras la instrucción de paro.

| Tiempo, en ms | Evento         | X, en m | Y, en m | Theta, en ° |
|---------------|----------------|---------|---------|-------------|
| 22735         | Comando stop   | 3.895   | 2.582   | 0.8         |
| 22785         | EKFTask        | 3.895   | 2.582   | 0.7         |
| 22885         | EKFTask        | 3.895   | 2.582   | 0.7         |
| 22985         | EKFTask        | 3.895   | 2.582   | 0.6         |
| 23065         | UWBMeasureTask | 3.724   | 2.583   | ---         |
| 23085         | EKFTask        | 3.895   | 2.582   | 0.4         |
| 23185         | EKFTask        | 3.895   | 2.582   | 0.3         |
| 23285         | EKFTask        | 3.895   | 2.582   | 0.2         |
| 23335         | EKFTask        | 4.059   | 2.699   | 0.2         |

### Respuesta del EKF durante un giro en sitio

Posteriormente, en  $t = 29685 \text{ ms}$ , se envió el comando para  $v_C = 0.0 \frac{m}{s}$  y  $\omega = 1.5 \frac{rad}{s}$ , para una maniobra de rotación pura. La Tabla 7.7 muestra la evolución de la orientación estimada junto con las mediciones crudas de la IMU y las correcciones de EKF. De estos datos se extraen las siguientes conclusiones:

- **Seguimiento de la orientación:** El EKF estima la orientación acumulada durante el giro, alcanzando más de  $150^\circ$  en aproximadamente 2.4 segundos. La velocidad angular media efectiva es de aproximadamente  $66.6^\circ/s$ , ligeramente inferior a la solicitada,  $86^\circ/s$  debido a la respuesta del controlador.
- **Presición frente a la IMU:** La diferencia entre el ángulo estimado por EKF y el ángulo medido por el giroscopio se mantiene por debajo de  $6^\circ$  durante toda la maniobra, y en la mayoría de los puntos es inferior a  $4^\circ$ . Esta discrepancia es aceptable para navegación.
- **Estabilidad de la posición durante el giro:** A pesar de la rotación, la posición estimada varía un poco  $x$  fluctúa entre  $4.111 \text{ m}$  y  $4.046 \text{ m}$ , rango de  $6.5 \text{ cm}$ , e  $y$  entre  $2.695 \text{ m}$  y  $2.581 \text{ m}$ , rango de  $11.4 \text{ cm}$ . Estas variaciones son coherentes con un giro ideal en el sitio y reflejan imperfecciones mecánicas, así como la corrección del filtro ante el ruido de los UWB.
- **Corrección por UWB durante el giro:** Las actualizaciones UWB en  $t = 30245 \text{ ms}$  y  $t = 31453 \text{ ms}$ , posiciones  $(4.091, 2.663)$  y  $(4.076, 2.661)$  respectivamente, están

relativamente cerca de la estimación del filtro, por lo que la corrección es suave y no perturba el seguimiento angular, dando como resultado una medición sin desplazamiento lineal.

*Tabla 7.7 Seguimiento de la orientación durante un giro en el sitio*

| Tiempo, en ms | Fuente          | theta_EKF, en ° | theta_IMU, en ° | Diferencia, en ° |
|---------------|-----------------|-----------------|-----------------|------------------|
| 29685         | Comando enviado | 0.2             | 0.17            | 0.03             |
| 29995         | SensorMeasure   | 0.2             | 0.27            | -0.07            |
| 30135         | EKFTask         | 0.4             | ---             | ---              |
| 30245         | UWBMeasureTask  | 1.1             | ---             | ---              |
| 30385         | EKFTask         | 2.4             | ---             | ---              |
| 30465         | SensorMeasure   | 3.4             | 7.43            | -4.03            |
| 30585         | EKFTask         | 8.0             | 13.05           | -5.05            |
| 30685         | EKFTask         | 12.6            | ---             | ---              |
| 30735         | EKFTask         | 17.1            | 20.92           | -3.82            |
| 30835         | EKFTask         | 27.3            | 31.00           | -3.70            |
| 30935         | EKFTask         | 38.8            | ---             | ---              |
| 30985         | EKFTask         | 44.6            | ---             | ---              |
| 31035         | EKFTask         | 51.6            | ---             | ---              |
| 31085         | EKFTask         | 58.3            | ---             | ---              |
| 31135         | EKFTask         | 64.6            | ---             | ---              |
| 31185         | SensorMeasure   | 70.9            | 70.16           | 0.74             |
| 31235         | EKFTask         | 77.2            | ---             | ---              |
| 31285         | EKFTask         | 82.6            | ---             | ---              |
| 31335         | EKFTask         | 88.3            | 84.08           | 4.22             |
| 31385         | EKFTask         | 93.9            | ---             | ---              |
| 31435         | UWBMeasureTask  | 100.2           | ---             | ---              |
| 31485         | EKFTask         | 106.2           | ---             | ---              |
| 31535         | EKFTask         | 111.7           | 110.30          | 1.40             |
| 31585         | EKFTask         | 117.1           | ---             | ---              |
| 31635         | EKFTask         | 122.3           | ---             | ---              |
| 31685         | EKFTask         | 127.5           | ---             | ---              |
| 31735         | EKFTask         | 131.9           | ---             | ---              |
| 31785         | SensorMeasure   | 136.4           | 133.12          | 3.28             |

## Resumen del desempeño del EKF

La Tabla 7.8 resume las métricas obtenidas para el sistema de fusión sensorial implementado.

*Tabla 7.8 Desempeño del EKF para estimación de pose.*

| Parámetro                                          | Valor obtenido                                           |
|----------------------------------------------------|----------------------------------------------------------|
| Frecuencia de actualización EKF                    | 20 Hz                                                    |
| Error de posición en reposo                        | $\pm 1.2$ cm, en X y $\pm 3.7$ cm, en Y                  |
| Error de orientación en reposo                     | $< 0.1^\circ$                                            |
| Error de orientación durante avance rectilíneo     | $1.5^\circ$ como máximo, estabilizado en $\pm 0.6^\circ$ |
| Diferencia media EKF vs. UWB                       | 0.28 m                                                   |
| Tiempo de estabilización tras comando de paro      | Aproximadamente 0.5 s                                    |
| Error máximo de orientación durante giro (vs. IMU) | $< 6^\circ$ , típicamente entre 3 y 5 grados             |
| Variación de posición durante giro en sitio        | $\Delta x \approx 6.5$ cm, $\Delta y \approx 11.4$ cm    |
| Capacidad de seguimiento de rotaciones rápidas     | Sí, hasta $67^\circ/\text{s}$ efectivos                  |
| Capacidad de rechazo de outliers                   | Sí, evitó saltos mayores a 0.5 m                         |

El comportamiento observado confirma que el filtro de Kalman extendido cumplió su objetivo: proporciona una estimación robusta, continua y suave de la pose del robot, aprovechando la alta frecuencia de la odometría y la IMU para la predicción, y la referencia absoluta del UWB para corregir la deriva. Esta fusión sensorial es esencial para el correcto funcionamiento del algoritmo de navegación por campos potenciales, que requiere una pose fiable en cada ciclo de control.

## 7.2 VALIDACIÓN DEL ALGORITMO DE NAVEGACIÓN

Una vez caracterizados los sistemas de percepción, estimación y control de bajo nivel, se procedió a validar el algoritmo de navegación basado en campos potenciales artificiales descrito en el CAPÍTULO 4. Para ello, se ejecutó una prueba de navegación autónoma en un entorno controlado, donde el robot debía desplazarse desde una posición inicial conocida hasta un objetivo predefinido, evitando obstáculos estáticos. Un programa se encargó de calcular en tiempo real el campo potencial total y generar el vector de comando que, tras ser transformado a velocidades lineales y angulares, fue enviado al controlador de bajo nivel.

### 7.2.1 Condiciones iniciales y parámetros de ensayo

Al iniciar la prueba, el robot se encontraba en la posición global  $(5.0876, 0.0110) m$ . El objetivo fue regresar al origen del sistema de referencia. Un único obstáculo fue detectado inicialmente en  $(2.01, 0.08) m$ , a una distancia euclidiana de  $2.01 m$  del robot. Los parámetros del campo potencial se mantuvieron en los valores sintonizados empíricamente: ganancias atractivas  $\epsilon_1$  y  $\epsilon_2$  iguales a uno, ganancia repulsiva  $k_{rep} = 1.0$ , el radio de influencia  $d_0 = 3.0 m$  y la ganancia tangencial fue de  $2.0$ .

El planificador operó a una frecuencia aproximada de  $20 Hz$ , aunque se observaron pequeñas variaciones debido a la carga computacional de la prueba. Cada iteración publicó un mensaje de log que incluye:

- Posición global del robot.
- Coordenadas del objetivo en el sistema de referencia local del robot.
- Distancia y posición de los obstáculos para saber si estaba dentro del rango del sensor.
- Gradiente atractivo  $\nabla U_{atr}$  y gradiente repulsivo  $\nabla U_{rep}$ .
- Gradiente total  $\nabla U_{total} = \nabla U_{atr} + \nabla U_{rep}$ .
- Vector de comando normalizado  $\vec{F} = -\nabla U_{total}$ , cuyas componentes  $(\vec{F}_x, \vec{F}_y)$  representan la dirección deseada de avance en el sistema móvil del robot.

### 7.2.2 Evolución de la trayectoria y respuesta del campo potencial

A lo largo de los tres logs se registraron más de 500 iteraciones, que cubren un desplazamiento total aproximado de  $5.0 m$ , desde  $x \approx 5.09$  hasta  $x \approx 0.07$ . En las Tablas 7.9 y 7.10 se muestran ocho instantes representativos que ilustran el comportamiento del planificador.

Tabla 7.9 Estado del sistema en iteraciones representativas, posición global y posición local obstáculo.

| Tiempo, en ms | Posición global (x,y), en m | Obstáculo ( $x_o, y_o$ ), en m |
|---------------|-----------------------------|--------------------------------|
| 1776708462    | (5.0876, 0.0110)            | (2.01, 0.08)                   |
| 1776708462    | (5.0856, 0.0113)            | (2.01, 0.09)                   |
| 1776708463    | (4.9837, 0.0241)            | (1.92, 0.20)                   |
| 1776708463    | (4.9210, 0.0359)            | (1.86, 0.29)                   |
| 1776708464    | (4.6049, 0.1546)            | (1.35, 0.71)                   |
| 1776708466    | (3.1457, 1.4637)            | (0.31, 1.49)                   |
| 1776708470    | (2.5387, 1.4416)            | (0.33, 1.55)                   |
| 1776708487    | (0.0733, 0.0635)            | --                             |

Tabla 7.10 Gradientes y vector de comando para las mismas iteraciones de la Tabla 7.9

| Tiempo, en ms | $\nabla U_{att}$    | $\nabla U_{rep}$   | $\vec{F} = (\vec{F}_x, \vec{F}_y)$ |
|---------------|---------------------|--------------------|------------------------------------|
| 1776708462    | (-0.99998,-0.00574) | (0.07145,0.32999)  | (0.92853,-0.32425)                 |
| 1776708462    | (-0.99997,-0.00827) | (0.07232,0.33156)  | (0.92765,-0.32329)                 |
| 1776708463    | (-0.99572,-0.09240) | (0.02616,0.37478)  | (0.96956,-0.28238)                 |
| 1776708463    | (-0.98985,-0.14213) | (0.01365,0.40531)  | (0.97620,-0.26318)                 |
| 1776708464    | (-0.90717,-0.42076) | (-0.09499,0.67427) | (1.00216,-0.25351)                 |
| 1776708466    | (-0.61124,-0.79145) | (-0.35965,1.12408) | (0.97089,-0.33263)                 |
| 1776708470    | (-0.47053,-0.88239) | (0.00000,0.00000)  | (0.47053,0.88239)                  |
| 1776708487    | (-0.10779,-0.00022) | (0.00000,0.00000)  | (0.10779,0.00022)                  |

Se observó que en las primeras iteraciones el gradiente repulsivo empuja al robot hacia la izquierda, componente y positiva, mientras que el atractivo apunta hacia atrás y a la izquierda. La suma da lugar a un vector de comando con  $\vec{F}_x > 0$  y  $\vec{F}_y$  ligeramente negativo, lo que indica un avance con un suave giro a la derecha para rodear el obstáculo. A medida que el robot se aproxima, la componente repulsiva crece y rota, manteniendo siempre  $\vec{F}_y$  negativo. Cuando el obstáculo sale del radio de influencia, fila 7, el gradiente repulsivo se anula y el comando cambia abruptamente a uno puramente atractivo,  $\vec{F}_y$  positivo. Finalmente, en las cercanías del objetivo, el gradiente atractivo es muy pequeño y el comando se estabiliza en un avance frontal.

### 7.2.3 Análisis cuantitativo del desempeño

A partir de los registros, se extrajeron las métricas resumidas en la Tabla 7.11.

*Tabla 7.11: Métricas de desempeño del planificador por campos potenciales*

| Parámetro                                   | Valor observado        |
|---------------------------------------------|------------------------|
| Duración total de la navegación             | $\approx 24.8$ s       |
| Distancia total recorrida (estimada)        | $\approx 5.0$ m        |
| Número de veces que se detectó un obstáculo | 347 de 512 iteraciones |
| Distancia mínima al obstáculo               | 1.16 m                 |
| Tiempo de reacción ante nuevo obstáculo     | $< 50$ ms              |
| Número de transiciones atracción/repulsión  | 12                     |
| Tasa de éxito, llegada al objetivo          | 100%                   |
| Colisiones                                  | Ninguna                |

### 7.2.4 Resumen del desempeño

El algoritmo de campos potenciales implementado en sistema operativo demostró ser capaz de guiar al robot desde una posición inicial hasta un objetivo, evitando colisiones en tiempo real. La naturaleza reactiva del método permitió adaptar la trayectoria ante la presencia de obstáculos estáticos, y la respuesta fue lo suficientemente rápida para ser ejecutada en el lazo de control del robot. Las limitaciones observadas pueden ser abordadas en una siguiente fase del proyecto mediante la integración con el algoritmo de Dijkstra y una estrategia de suavizado del campo repulsivo en el borde de  $d_0$ .

## CAPÍTULO 8

### CONCLUSIONES

#### 8.1 APORTACIONES DEL TRABAJO

El presente trabajo de tesis culminó con el desarrollo, integración y validación experimental de un sistema de navegación autónoma funcional para un robot móvil diferencial. La principal contribución de este proyecto radica en la sinergia lograda entre los distintos niveles de la pirámide de control robótico: desde la corrección de errores sistemáticos en sensores de bajo costo, nivel hardware/firmware, hasta la ejecución de un algoritmo reactivo de planeación de trayectorias, nivel cognitivo.

A diferencia de trabajos puramente simulados, esta investigación enfrentó y resolvió las no linealidades y el ruido, inherentes a un sistema físico. Las aportaciones técnicas específicas que se derivan de este proceso son:

- **Caracterización y corrección de encoders.** Se diseñó e implementó una metodología de calibración basada en una tabla de consulta para compensar la distribución no uniforme de imanes en los encoders de efecto Hall. Esta técnica permitió reducir la varianza del error de medición de velocidad angular de un 13.5% a menos del 1%, proporcionando una señal de retroalimentación limpia esencial para la estabilidad del controlador PID de bajo nivel.

- **Arquitectura de fusión sensorial.** Se integró exitosamente un filtro de Kalman extendido en una plataforma embebida, la tarjeta ESP32, capaz de fusionar asincrónicamente la odometría, la deriva de la IMU y las correcciones absolutas provistas por el sistema UWB. Este esquema demostró ser capaz de rechazar valores atípicos del sistema UWB y mantener una estimación de pose suave con un error de orientación estabilizado durante el movimiento.
- **Validación experimental del enfoque híbrido.** Se demostró la viabilidad del método de campos potenciales artificiales como estrategia de navegación reactiva local en un entorno real. El sistema logró una navegación exitosa a lo largo de una trayectoria de 5.0 metros, manteniendo una distancia mínima de seguridad al obstáculo de 1.16 metros y reaccionando a las perturbaciones del entorno en tiempos menores a 50 ms.

Estas contribuciones validan que es factible construir sistemas de navegación autónoma robustos utilizando componentes de bajo costo y hardware computacionalmente limitado, siempre que se complementen con un modelado matemático riguroso y estrategias de filtrado adecuadas para mitigar las imperfecciones del hardware.

## 8.2 CUMPLIMIENTO DE OBJETIVOS

Con base en los resultados cuantitativos y cualitativos presentados en el Capítulo 7, se concluye que los objetivos planteados al inicio de esta investigación fueron alcanzados en su totalidad.

- **Caracterización dinámica.** Se identificó satisfactoriamente el modelo dinámico de primer orden de los actuadores  $G(s) = \frac{k}{s+a}$ , lo que permitió el diseño de un controlador de velocidad con un error de estado estacionario inferior al 0.2%.
- **Percepción y localización.** Se estableció un método confiable de trilateración UWB con una precisión promedio de  $\pm 2.5 \text{ cm}$  y  $\pm 3.1 \text{ cm}$  en los ejes  $x$  e  $y$  respectivamente, complementado con segmentación LiDAR para la identificación de centroides de obstáculos.
- **Control de movimiento.** El sistema fue capaz de descomponer el vector gradiente del campo potencial en comandos de rapidez lineal y angular, los cuales fueron ejecutados exitosamente por el lazo de control PID digital, lo que demostró la correcta aplicación de la cinemática diferencial descrita en la ecuación (5.11).

- **Integración del prototipo.** Se logró la comunicación bidireccional entre la unidad de procesamiento, ROS 2 y el microcontrolador ESP32, logrando un ciclo de navegación autónoma continuo que permitió al robot desplazarse desde la coordenada inicial (5.08, 0.01) hasta la meta en el origen (0.07, 0.6 ) m.

### 8.3 LIMITACIONES DEL SISTEMA

La implementación física del sistema reveló ciertas limitaciones que, si bien no impidieron el cumplimiento de los objetivos, definen las restricciones operativas actuales del prototipo y señalan áreas de mejora para futuras iteraciones:

- **Sintonización empírica del campo potencial.** La eficacia del algoritmo de navegación demostró ser altamente sensible a los parámetros  $k_{rep}$ ,  $\epsilon$  y particularmente al radio de influencia  $d_0$ . Como se observó en la sección 7.2.4, la anulación abrupta del campo repulsivo en el límite de  $d_0$  provocó cambios discontinuos en el vector de comando, lo que podría generar inestabilidad en configuraciones de obstáculos más densas.
- **Latencia en la localización absoluta.** El sistema UWB, aunque preciso en el promedio, opera a una tasa de actualización limitada, a 0.83 Hz, y es susceptible a errores de multitrayectoria, con picos de error de hasta 0.62 m. Esta latencia restringe la capacidad del robot para reaccionar a tiempo real a altas velocidades, obligando a confiar en la odometría durante los intervalos largos entre correcciones absolutas y perder precisión en caso de obstruirse la línea de visión entre los sensores UWB.
- **Fricción estática y modelado del actuador:** La discrepancia entre el control PID teórico, críticamente amortiguado, y la respuesta experimental, con sobrepaso del 25.5%, evidencia que el modelo lineal de primer orden no captura las no linealidades dominantes, específicamente la fricción estática. Esta no linealidad provoca un pico de corriente al arranque y un sobrepaso inevitable sin una estrategia de control no lineal o un lazo de corriente dedicado.

## 8.4 TRABAJO FUTURO

A partir de la experiencia adquirida y las limitaciones identificadas, se proponen las siguientes ideas para evolucionar el sistema actual hacia una plataforma más robusta y versátil.

- **Planeación global y evasión de mínimos locales.** Se recomienda finalizar la integración del algoritmo de Dijkstra o A estrella sobre una red topológica de nodos como se explicó en el estado de la técnica. Esto permitirá validar la arquitectura híbrida completa, utilizando el grafo para escapar de configuraciones de obstáculos cóncavos tipo U, donde el campo potencial puro podría fallar. La publicación de destinos intermedios por parte de Dijkstra dotaría al sistema de una “memoria” de largo plazo de la cual carece actualmente.
- **Optimización de parámetros mediante aprendizaje por refuerzo.** Para eliminar la dependencia del ajuste manual de parámetros, se propone entrenar un agente de aprendizaje por refuerzo en un entorno de simulación Gazebo/ROS 2, que replique la dinámica del prototipo. El agente podría aprender a modular dinámicamente  $k_{rep}$  y  $d_0$  en función de la densidad de obstáculos, suavizando la transición repulsiva y mejorando la fluidez del movimiento.
- **Mejora del filtro de Kalman.** Se sugiere modificar el modelo de predicción del EKF para incorporar explícitamente los comandos de velocidad enviados a los motores, la señal de control  $U[k]$  en lugar de depender únicamente de la odometría medida, lo que permitiría predecir el deslizamiento. Asimismo, se debería implementar una lógica de rechazo de valores fuera de rango basada en la prueba de Chi-cuadrado sobre la innovación del filtro, para evitar que mediciones UWB erróneas corrompan la estimación de la pose.
- **Control de bajo nivel avanzado.** Para mitigar el sobrepaso causado por la fricción estática, se propone implementar un compensador por fricción o rediseñar el control tomando en cuenta la zona muerta de los motores.

## 8.5 REFLEXIÓN FINAL

Este trabajo de tesis constituye una prueba fehaciente de que la teoría clásica de control y robótica móvil puede ser implementada con éxito en plataformas físicas con recursos limitados, siempre que el diseño metodológico contemple una caracterización meticulosa del hardware.

El sistema resultante no es simplemente una simulación o un concepto teórico, sino un robot físico que percibe su entorno mediante LiDAR, se localiza mediante radiofrecuencia, fusiona sensores para estimar su estado de manera óptima y navega de forma reactiva evitando colisiones. La documentación detallada de los problemas experimentales, como la corrección de la no linealidad de los encoders magnéticos mediante LUT, fue en sí misma una contribución valiosa para la comunidad de robótica educativa y de investigación aplicada.

Se considera que, a pesar de las áreas de oportunidad identificadas, el prototipo desarrollado representa una base sólida y escalable. Se ha demostrado que la combinación de un microcontrolador ESP32, sensores asequibles y el middleware ROS 2 constituyen un ecosistema viable para la experimentación en navegación autónoma. La culminación exitosa de este proyecto pretende abrir la puerta a futuras investigaciones en el laboratorio sobre robótica móvil cooperativa y navegación en entornos humanos compartidos.



## REFERENCIAS

- [1] “ISO 8373:2021(en), Robotics — Vocabulary”. Consultado: el 24 de julio de 2025. [En línea]. Disponible en: <https://www.iso.org/obp/ui/en/#iso:std:iso:8373:ed-3:v1:en>
- [2] D. J. Griffiths y C. Inglefield, “Introduction to Electrodynamics”, *Am. J. Phys.*, vol. 73, núm. 6, pp. 574–574, jun. 2005, doi: 10.1119/1.4766311.
- [3] Northwestern Robotics, *Modern Robotics, Chapter 10.6: Virtual Potential Fields*, (el 16 de marzo de 2018). Consultado: el 11 de abril de 2025. [En línea Video]. Disponible en: <https://www.youtube.com/watch?v=8Vva0bnMIEI>
- [4] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots”, en *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, St. Louis, MO, USA: Institute of Electrical and Electronics Engineers, 1985, pp. 500–505. doi: 10.1109/ROBOT.1985.1087247.
- [5] “Gradiente Descendente – Numerentur.org”. Consultado: el 5 de julio de 2025. [En línea]. Disponible en: <https://numerentur.org/gradiente-descendente/>
- [6] E. Cassingena Navone, “Algoritmo de la ruta más corta de Dijkstra - Introducción gráfica y detallada”, [freeCodeCamp.org](https://www.freecodecamp.org/espanol/news/algoritmo-de-la-ruta-mas-corta-de-dijkstra-introduccion-grafica/). Consultado: el 16 de abril de 2025. [En línea]. Disponible en: <https://www.freecodecamp.org/espanol/news/algoritmo-de-la-ruta-mas-corta-de-dijkstra-introduccion-grafica/>
- [7] A. M. Lopez, “Algoritmo Dijkstra: Rutas Eficientes”. Consultado: el 16 de abril de 2025. [En línea]. Disponible en: <https://saberpunto.com/tecnologia/algoritmo-dijkstra-rutas-eficientes/>

- 
- [8] S. M. H. Rostami, A. K. Sangaiah, J. Wang, y X. Liu, “Obstacle avoidance of mobile robots using modified artificial potential field algorithm”, *EURASIP J. Wirel. Commun. Netw.*, vol. 2019, núm. 1, p. 70, dic. 2019, doi: 10.1186/s13638-019-1396-2.
  - [9] T. Zhang, Y. Zhu, y J. Song, “Real-time motion planning for mobile robots by means of artificial potential field method in unknown environment”, *Ind. Robot Int. J.*, vol. 37, núm. 4, pp. 384–400, jun. 2010, doi: 10.1108/01439911011044840.
  - [10] F. Jin, Z. Ye, M. Li, H. Xiao, W. Zeng, y L. Wen, “A New Hybrid Reinforcement Learning with Artificial Potential Field Method for UAV Target Search”, *Sensors*, vol. 25, núm. 9, p. 2796, abr. 2025, doi: 10.3390/s25092796.
  - [11] N. H. Bustos de la Cruz, J. C. Godínez García, R. Osorio Comparán, N. A. de M. Universidad, y N. A. de M. Universidad, “Navegación de un robot móvil aplicando campos potenciales y reconocimiento de objetos usando OS Android”, 2016.
  - [12] J. Cruz Navarro, J. Savage Carmona, N. A. de M. Universidad, y N. A. de M. Universidad, “Optimización de campos potenciales para navegación mediante algoritmos genéticos”, 2013.
  - [13] S. M. LaValle, *Planning algorithms*. Cambridge ; New York: Cambridge University Press, 2006. [En línea]. Disponible en: <http://planning.cs.uiuc.edu/>
  - [14] S. Thrun, W. Burgard, y D. Fox, *Probabilistic robotics*. en Intelligent robotics and autonomous agents. Cambridge, Massachusetts London: MIT Press, 2006.
  - [15] R. A. Brooks, “Elephants don’t play chess”, *Robot. Auton. Syst.*, vol. 6, núm. 1–2, pp. 3–15, jun. 1990, doi: 10.1016/S0921-8890(05)80025-9.
  - [16] R. S. Sutton y A. G. Barto, *Reinforcement learning: an introduction*, Second edition. en Adaptive computation and machine learning series. Cambridge, Massachusetts: The MIT Press, 2018. [En línea]. Disponible en: <http://incompleteideas.net/book/the-book-2nd.html>
  - [17] Setramex Oficial, *LA TRILATERACIÓN*, (el 28 de septiembre de 2020). Consultado: el 29 de abril de 2025. [En línea Video]. Disponible en: <https://www.youtube.com/watch?v=UQJll2KqiQ>
  - [18] A. Barrientos, L. F. Peñín, C. Balaguer, y A. Rafael, *Fundamentos de Robótica*, Segunda Edición. McGraw-Hill, 1997.
  - [19] K. Lynch y F. C. Park, *Modern robotics: mechanics, planning, and control*. Cambridge, UK: Cambridge University Press, 2017.
  - [20] “Robots Móviles – BioRobotics”. Consultado: el 10 de abril de 2025. [En línea]. Disponible en: <https://biorobotics.fi-p.unam.mx/robots-moviles/>
  - [21] Engineering Educator Academy, *Kinematics of Differential Drive Robots and Odometry*, (el 31 de julio de 2021). Consultado: el 29 de agosto de 2025. [En línea Video]. Disponible en: <https://www.youtube.com/watch?v=RZIZcDxQ8P4>

- 
- [22] J. L. Meriam, *Mécanica para ingenieros: dinámica*, 3ª ed., [Reimp.]. Barcelona: Reverté, 2012.
  - [23] “Controlador PID \_ AcademiaLab”. Consultado: el 10 de enero de 2026. [En línea]. Disponible en: <https://academia-lab.com/enciclopedia/controlador-pid/>
  - [24] “MATLAB - El lenguaje del cálculo técnico”. Consultado: el 10 de enero de 2026. [En línea]. Disponible en: <https://la.mathworks.com/products/matlab.html>
  - [25] “pid - Controlador PID de forma paralela - MATLAB”. Consultado: el 2 de diciembre de 2025. [En línea]. Disponible en: <https://la.mathworks.com/help/control/ref/pid.html>
  - [26] Modelado, Identificación, Control (A. Sala), *Discretización por Tustin bilin.: preservación estabilidad y respuesta en frecuencia (distorsionada)*, (el 11 de abril de 2021). Consultado: el 10 de enero de 2026. [En línea Video]. Disponible en: <https://www.youtube.com/watch?v=htkfmFxpQI>
  - [27] E. Soria, “Transformada Z”. Universitat de Valencia. [En línea]. Disponible en: [https://www.uv.es/soriae/tema\\_3\\_pds.pdf](https://www.uv.es/soriae/tema_3_pds.pdf)
  - [28] “Extended Kalman Filter — AHRS 0.4.0 documentation”. Consultado: el 31 de julio de 2026. [En línea]. Disponible en: <https://ahrs.readthedocs.io/en/latest/filters/ekf.html>
  - [29] T. Kaiser, Feng Zheng, y E. Dimitrov, “An Overview of Ultra-Wide-Band Systems With MIMO”, *Proc. IEEE*, vol. 97, núm. 2, pp. 285–312, feb. 2009, doi: 10.1109/jproc.2008.2008784.
  - [30] X. Chi, G. Wu, J. Liu, J. Xu, y Q. Lu, “Review on Ultra Wide Band Indoor Localization”, *Indones. J. Comput. Eng. Des. IJoCED*, vol. 2, núm. 2, p. 99, oct. 2020, doi: 10.35806/ijoced.v2i2.118.
  - [31] “REYAX RYUW122\_Lite Interfaz UART 6.5 GHz y 8 GHz UWB Antena Transceptor Módulo Lite Medición de distancia de la placa de evaluación 2D / 3D Posicionamiento : Amazon.com.mx: Electrónicos”. Consultado: el 21 de julio de 2025. [En línea]. Disponible en: [https://www.amazon.com.mx/RYUW122\\_Lite-Interfaz-Antena-Transceptor-evaluaci%C3%B3n/dp/B0CBTBYH87/ref=pd\\_bxgy\\_thbs\\_d\\_sccl\\_1/132-5578907-7826326?pd\\_rd\\_w=wqFHM&content-id=amzn1.sym.2d9599b5-7233-437e-88b4-410bd9633fcd&pf\\_rd\\_p=2d9599b5-7233-437e-88b4-410bd9633fcd&pf\\_rd\\_r=JQ5WXXDBFSD9K4M56N8F&pd\\_rd\\_wg=XaLFV&pd\\_rd\\_r=14ea58d-d-28e6-4eda-a05a-3a7ea34c5b77&pd\\_rd\\_i=B0CBTBYH87&psc=1](https://www.amazon.com.mx/RYUW122_Lite-Interfaz-Antena-Transceptor-evaluaci%C3%B3n/dp/B0CBTBYH87/ref=pd_bxgy_thbs_d_sccl_1/132-5578907-7826326?pd_rd_w=wqFHM&content-id=amzn1.sym.2d9599b5-7233-437e-88b4-410bd9633fcd&pf_rd_p=2d9599b5-7233-437e-88b4-410bd9633fcd&pf_rd_r=JQ5WXXDBFSD9K4M56N8F&pd_rd_wg=XaLFV&pd_rd_r=14ea58d-d-28e6-4eda-a05a-3a7ea34c5b77&pd_rd_i=B0CBTBYH87&psc=1)
  - [32] make2explore Systems, *[Device Overview] Ultra-wideband Transceiver Module RYUW122*, (el 5 de marzo de 2024). Consultado: el 12 de enero de 2026. [En línea Video]. Disponible en: <https://www.youtube.com/watch?v=HwBakNIPAz4>
  - [33] J. Xu, M. Ma, y C. L. Law, “Position estimation using ultra-wideband time difference of arrival measurements”, *IET Sci. Meas. Technol.*, vol. 2, núm. 1, pp. 53–58, ene. 2008, doi: 10.1049/iet-smt:20060089.

- [34] “MPU-9250 IMU de 9DOF 9250”, UNIT Electronics. Consultado: el 16 de agosto de 2025. [En línea]. Disponible en: <https://uelectronics.com/producto/mpu-9250-imu-de-9dof-9250/>
- [35] “¿Qué es LiDAR? | IBM”. Consultado: el 2 de agosto de 2025. [En línea]. Disponible en: <https://www.ibm.com/es-es/think/topics/lidar>
- [36] “URG-04LX-UG01 | Products List | Discontinued | Scanning Rangefinder | Distance Data Output | URG-04LX-UG01”, HOKUYO AUTOMATIC CO., LTD. Consultado: el 5 de agosto de 2025. [En línea]. Disponible en: <https://www.hokuyo-aut.jp/search/single.php?serial=166>
- [37] “System Identification Toolbox”. Consultado: el 12 de enero de 2026. [En línea]. Disponible en: <https://la.mathworks.com/products/sysid.html>
- [38] “Sensor Magnetico Digital (Efecto Hall) KY-003 -UNIT Electronics”. Consultado: el 12 de enero de 2026. [En línea]. Disponible en: <https://uelectronics.com/producto/sensor-magnetico-digital-hall-ky-003/>
- [39] “Pulse Counter (PCNT) - ESP32 - — ESP-IDF Programming Guide v5.5.2 documentation”. Consultado: el 12 de enero de 2026. [En línea]. Disponible en: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/peripherals/pcnt.html>
- [40] “Non-Volatile Storage Library - ESP32 - — ESP-IDF Programming Guide v5.5.2 documentation”. Consultado: el 12 de enero de 2026. [En línea]. Disponible en: [https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/storage/nvs\\_flash.html](https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/storage/nvs_flash.html)
- [41] “ESP32-DevKitC V4 - ESP32 - — esp-dev-kits latest documentation”. Consultado: el 12 de enero de 2026. [En línea]. Disponible en: [https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32/esp32-devkitc/user\\_guide.html](https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32/esp32-devkitc/user_guide.html)