

UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN MECATRÓNICA

TRABAJO DE GRADO

TEMA:

"SISTEMA DE ODOMETRÍA MODULAR PARA PLATAFORMA DIFERENCIAL AUTÓNOMA USANDO ROS"

Trabajo de titulación previo a la obtención del título de Ingeniero en Mecatrónica

Línea de investigación: Producción industrial y tecnología sostenible

AUTOR:

Mauricio Alexander López Villalba

DIRECTOR:

Dr. Carlos Xavier Rosero Chandi

Ibarra – Ecuador 2026

UNIVERSIDAD TÉCNICA DEL NORTE

DIRECCIÓN DE BIBLIOTECA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO	
APELLIDOS Y NOMBRES:	López Villalba Mauricio Alexander

DATOS DE LA OBRA	
TÍTULO:	"Sistema de odometría modular para plataforma diferencial autónoma usando ROS"
AUTOR:	López Villalba Mauricio Alexander
FECHA:	27/07/2026
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	■ PREGRADO ☐ POSGRADO
TÍTULO POR EL QUE OPTA:	Ingeniero Mecatrónico
ASESOR / DIRECTOR:	PhD. Carlos Xavier Rosero Chandi MSc. Luz María Tobar Subía Contento

2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 27 días del mes de Julio de 2026

EL AUTOR: Validar únicamente en FirmaEC. Firmado electrónicamente por: MAURICIO ALEXANDER LOPEZ VILLALBA

Nombre: López Villalba Mauricio Alexander

C.I: 1755746599

CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ibarra, 27 de julio de 2026

PhD. Carlos Xavier Rosero Chandi

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.



PhD. Carlos Xavier Rosero Chandi

C.I:1002515821

Director de Tesis

APROBACIÓN DEL COMITÉ CALIFICADOR

El Comité Calificado del trabajo de Integración Curricular “Sistema de odometría modular para plataforma diferencial autónoma usando ROS” elaborado por Mauricio Alexander López Villalba, previo a la obtención del título de Ingeniero Mecatrónico, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:



Dr. Carlos Xavier Rosero Chandi

C.I:1002515821

Director de Tesis



MSc. Luz María Tobar Subía Contento

C.I:1002444204

Asesor de Tesis

Dedicatorias

Dedico este trabajo, con todo mi amor y gratitud, a mi abuelita, por todo su apoyo y muestras de amor, por ser ejemplo de fortaleza y de cómo salir adelante. Gracias por tus consejos, por acompañarme en cada paso y, sobre todo, por creer siempre en mí.

A mi mamá, con todo mi amor y gratitud, por ser el pilar de mi vida y el ejemplo de perseverancia y entrega. Gracias por tus sacrificios, tu apoyo incondicional y por enseñarme a nunca rendirme. Este logro también es tuyo.

A mis hermanos, Sebas y Martín, por su confianza, su apoyo y por acompañarme, cada uno a su manera, a lo largo de este camino. Gracias por motivarme a seguir adelante y por formar parte de este sueño.

A mi familia, por su compañía constante, su fe en mí y por acompañarme en cada etapa de este largo camino.

En especial a la memoria de mi padre que, aunque ya no está presente entre nosotros, vive en mi recuerdo, en mis decisiones y en el esfuerzo diario por convertirme en alguien del que estaría orgulloso, por su ejemplo y apoyo en vida, encontré la motivación para continuar mi carrera.

Y a todas aquellas personas que, con su amistad, apoyo y compañía, hicieron posible que hoy me encuentre aquí, cumpliendo esta meta. Este logro no es solo mío también es de ustedes.

Agradecimientos

Al culminar esta etapa tan significativa de mi vida, quiero expresar mi más sincero agradecimiento primeramente a Dios y a todas las personas que hicieron posible este logro.

A mi mamá, por su amor incondicional, sus sacrificios, sus consejos y por ser mi mayor apoyo en cada etapa de mi vida. Gracias por creer siempre en mí y por enseñarme que la perseverancia es el camino para alcanzar los sueños.

A mi hermano Sebas, por su apoyo constante y por creer en mí incluso cuando yo mismo dudaba, siempre has sido una fuente de inspiración para poder seguir adelante y te sientas orgulloso de mi.

A mi hermano Martín, quien, a pesar de la distancia, siempre me ha demostrado su cariño. Espero que este logro también sea motivo de orgullo para ti.

A mis amigos, por acompañarme en los momentos más difíciles y en los de mayor alegría. Gracias por su apoyo, su compañía y por hacer este camino mucho más llevadero.

Agradezco al Dr. Xavier Rosero, director de esta tesis, por su guía, dedicación y valiosas enseñanzas. Su apoyo y experiencia fueron fundamentales para el desarrollo de este trabajo.

A mis profesores, por compartir sus conocimientos, valores y enseñanzas, contribuyendo a mi crecimiento personal y profesional.

A Brisa, gracias por haber compartido una parte importante de este camino, por tu apoyo, tu compañía y los aprendizajes que me dejaste. Siempre valoraré el tiempo y el impulso que brindaste durante esta etapa.

Finalmente, agradezco a todas las personas que, con una palabra, un consejo o un gesto de apoyo, dejaron una huella en este recorrido. A todos ustedes, muchas gracias.

Índice general

Certificación del director del trabajo de integración curricular II

Aprobación del comité calificador III

Dedicatorias IV

Agradecimientos V

Índice general VI

Índice de figuras IX

Índice de tablas X

Resumen XII

Abstract 1

I. Introducción 2

1.1. Planteamiento del problema 2

1.2. Objetivos 3

1.2.1. General 3

1.2.2. Específicos 3

1.3. Alcance y delimitación 3

1.4. Justificación 4

II. Marco teórico **5**

2.1. Antecedentes 5

2.2. Fundamentos de robots móviles 7

2.2.1. Robots móviles diferenciales 7

2.2.2. Cinemática de robots móviles 8

2.2.3. Odometría diferencial 8

2.2.4. Fuentes de error en la odometría 9

2.3. Modelo matemático de la odometría 10

2.3.1. Modelo cinemático diferencial 10

2.3.2. Modelo cinemático tipo bicicleta 11

2.4. Sistema ROS 2 12

2.4.1. Arquitectura ROS 2 12

2.4.2. Nodos ROS 14

2.4.3. Herramientas y paquetes de ROS 2 para el desarrollo del sistema de odometría 14

2.4.4. Encoders y sensores en odometría 16

III. Marco metodológico **18**

3.1. Enfoque de la investigación 18

3.2. Diseño de la investigación 19

3.2.1. Fase 1: Análisis y modelado del sistema de odometría diferencial . . . 19

3.2.2. Fase 2: Implementación del sistema en ROS 2 19

3.2.3. Fase 3: Validación experimental y análisis de resultados 20

3.3. Descripción de la plataforma robótica utilizada 20

3.3.1. Antecedentes de la plataforma robótica 20

3.3.2. Características mecánicas del robot 21

3.3.3. Sistema de tracción diferencial 21

3.3.4. Sistema de adquisición de datos 21

3.3.5. Unidad de procesamiento Jetson Orin Nano 22

3.3.6. Integración hardware-software 23

3.4. Arquitectura del sistema de odometría desarrollado 24
3.4.1. Arquitectura general del sistema 24
3.4.2. Sistema operativo Ubuntu 22.04.5 LTS 26
3.4.3. Entorno de desarrollo ROS 2 26
3.4.4. Nodo de encoders simulados 27
3.4.5. Nodo de encoders reales 27
3.4.6. Nodo de cálculo de odometría 28
3.4.7. Publicación de mensajes Odometry 28
3.4.8. Visualización mediante RViz 29

IV. Resultados y análisis **30**

4.1. Configuración experimental 30
4.1.1. Entorno de simulación 30
4.1.2. Entorno real 31
4.1.3. Parámetros de prueba 31
4.2. Pruebas con encoders simulados 32
4.2.1. Movimiento rectilíneo 32
4.2.2. Movimiento circular 33
4.2.3. Movimiento aleatorio 34
4.3. Pruebas con encoders reales 36
4.3.1. Movimiento rectilíneo 36
4.3.2. Movimiento circular 38
4.3.3. Trayectoria libre bajo operación manual 39
4.4. Comparación entre simulación y entorno real 40
4.4.1. Comparación de trayectorias 40
4.4.2. Análisis de desviaciones observadas 41
4.4.3. Limitaciones del sistema 42
4.5. Análisis de resultados 42

V. Conclusiones y recomendaciones **44**

5.1. Conclusiones 44

5.2. Recomendaciones 45

5.3. Trabajo a futuro 45

Anexos **51**

Anexo A 51

Índice de figuras

2.1. Elementos principales de la arquitectura de ROS 13
2.2. Módulo FC-03 17

3.1. Diagrama del sistema de adquisición de datos 22
3.2. Jetson Orin Nano 8GB developer kit [1] 23
3.3. Diagrama de bloques del sistema de odometría planteado 24
3.4. Arquitectura funcional del sistema de odometría diferencial desarrollado 25

4.1. Trayectoria estimada en movimiento rectilíneo (encoders simulados) 33
4.2. Trayectoria estimada en movimiento circular (encoders simulados) 34
4.3. Trayectoria estimada en movimiento aleatorio (encoders simulados) 35
4.4. Trayectoria estimada en movimiento rectilíneo (encoders reales) 37
4.5. Trayectoria estimada en movimiento circular (encoders reales) 38
4.6. Trayectoria estimada durante operación manual (encoders reales) 39

Índice de tablas

2.1. Clasificación de errores en sistemas de odometría diferencial. 9
2.2. Mecanismos de comunicación en la arquitectura ROS 2. 15
2.3. Componentes de la mensajería de odometría en ROS 2. 16
2.4. Comparación entre encoder incremental y absoluto. 17

4.1. Comparación entre simulación y entorno real. 31
4.2. Resumen de las pruebas realizadas con encoders simulados. 36
4.3. Resumen de las pruebas realizadas con encoders reales. 40
4.4. Comparación entre las trayectorias obtenidas en simulación y en el entorno real. 41

Resumen

El presente trabajo de titulación tiene como objetivo desarrollar e implementar un sistema de odometría diferencial para un robot móvil de cuatro ruedas con tracción diferencial utilizando ROS 2. La odometría diferencial constituye una técnica ampliamente empleada en robótica móvil para estimar la posición y orientación de un robot a partir de la información proporcionada por los encoders de las ruedas; sin embargo, presenta una acumulación inherente de error que hace necesaria su evaluación tanto en entornos simulados como reales.

Para alcanzar este objetivo, se realizó un análisis de los fundamentos teóricos de la odometría diferencial y de su implementación en ROS 2, considerando trabajos previos y plataformas robóticas comparables. Posteriormente, se diseña e implementa una arquitectura modular para la adquisición y procesamiento de datos provenientes de encoders simulados en Gazebo y encoders físicos instalados en la plataforma robótica. Finalmente, el sistema es validado mediante pruebas experimentales realizadas tanto en el entorno de simulación como en el entorno real, considerando movimientos rectilíneos, circulares y trayectorias variables.

Los resultados obtenidos demuestra que el algoritmo desarrollado reproduce de manera consistente el comportamiento esperado del modelo matemático en simulación, obteniendo trayectorias acordes con los comandos de movimiento aplicados. En el entorno real se observa un comportamiento similar, aunque con pequeñas desviaciones atribuibles a las limitaciones propias de la plataforma física y de la odometría basada exclusivamente en encoders.

El sistema desarrollado se constituye como una plataforma modular para el estudio de la odometría diferencial en robots móviles y una base para futuras investigaciones relacionadas con técnicas de fusión sensorial, localización y navegación autónoma. El presente trabajo contribuye al estudio práctico de la odometría diferencial en entornos educativos y de investigación.

Palabras clave: odometría diferencial, ROS 2, robot móvil, encoders incrementales, robótica móvil.

Abstract

The aim of this degree project is to develop and implement a differential odometry system for a four-wheeled mobile robot with differential drive, using ROS 2. Differential odometry is a technique widely used in mobile robotics to estimate a robot's position and orientation based on information provided by the wheel encoders; however, it suffers from an inherent accumulation of error, making it necessary to evaluate it in both simulated and real-world environments.

To achieve this objective, an analysis was carried out of the theoretical foundations of differential odometry and its implementation in ROS 2, taking into account previous work and comparable robotic platforms. Subsequently, a modular architecture was designed and implemented for the acquisition and processing of data from encoders simulated in Gazebo and physical encoders installed on the robotic platform. Finally, the system is validated through experimental tests carried out in both the simulation and real-world environments, considering rectilinear and circular movements as well as variable trajectories.

The results obtained demonstrate that the algorithm developed consistently reproduces the expected behaviour of the mathematical model in simulation, producing trajectories consistent with the applied motion commands. In the real-world environment, similar behaviour is observed, albeit with minor deviations attributable to the inherent limitations of the physical platform and of odometry based exclusively on encoders.

The system developed serves as a modular platform for the study of differential odometry in mobile robots and provides a basis for future research into sensor fusion techniques, localisation and autonomous navigation. This work contributes to the practical study of differential odometry in educational and research environments.

Keywords: differential odometry, ROS 2, mobile robot, incremental encoders, mobile robotics.

Capítulo I

Introducción

1.1. Planteamiento del problema

La estimación confiable de la posición y orientación en robots móviles es un requisito fundamental para tareas de navegación autónoma, planificación de trayectorias y control de movimiento. En robots con tracción diferencial, la odometría basada en encoders de rueda se implementa por su simplicidad y bajo costo. No obstante, este método presenta limitaciones considerables debido a errores sistemáticos y acumulativos generados por el deslizamiento, desgaste mecánico o irregularidades en el terreno, lo cual afecta directamente la precisión del sistema [2].

Si bien el entorno ROS (Robot Operating System) proporciona herramientas para el desarrollo modular y visualización de datos espaciales, la implementación de un sistema de odometría robusto y de bajo costo computacional continúa siendo un desafío, especialmente en plataformas sin acceso a sensores externos como GPS o LIDAR. Esta situación limita la capacidad de los robots móviles para operar de forma autónoma en entornos reales, comprometiendo su rendimiento y fiabilidad.

El robot móvil diferencial de cuatro ruedas, desarrollado en un trabajo de titulación anterior en la Universidad Técnica del Norte, actualmente en fase de desarrollo, requiere de un sistema de odometría funcional que permita avanzar hacia la integración de algoritmos de navegación autónoma. Sin esta capacidad, el sistema carece de una base de localización interna que soporte comportamientos autónomos coherentes y reproducibles.

Ante esta necesidad, se plantea el diseño e implementación de un sistema de odometría basado en ROS, que permita estimar en tiempo real la posición y orientación del robot, utilizando únicamente sensores de rueda y herramientas de código abierto. Esta solución busca ser repli-

cable, eficiente y adaptable, constituyendo un paso técnico clave en la evolución del robot hacia operaciones autónomas en entornos reales.

1.2. Objetivos

1.2.1. General

Desarrollar un sistema de odometría en un robot móvil diferencial utilizando ROS, que permita la estimación en tiempo real de su posición y orientación.

1.2.2. Específicos

- Analizar los fundamentos teóricos de la odometría diferencial y su implementación en ROS, considerando trabajos previos y plataformas comparables.
- Diseñar un sistema de adquisición y procesamiento de datos a partir de sensores de rueda, integrando los nodos necesarios en ROS.
- Validar el sistema de odometría mediante pruebas experimentales que comparen las trayectorias estimadas frente a referencias reales.

1.3. Alcance y delimitación

El presente trabajo de grado se centra en el diseño e implementación de un sistema de odometría para un robot móvil diferencial actualmente en fase de desarrollo. El objetivo es estimar la posición y orientación del robot sobre un plano bidimensional, utilizando únicamente la información obtenida a partir de sensores de movimiento instalados en las ruedas. Esta estimación permitirá establecer una base técnica para futuras tareas de navegación autónoma.

El desarrollo del sistema contempla tres etapas principales. En la primera, se diseña el subsistema de adquisición de datos, considerando las señales generadas por los sensores de desplazamiento. En la segunda, se implementa el modelo matemático de odometría en un entorno de programación modular, el cual incluye la visualización de trayectorias y la representación del desplazamiento del robot en tiempo real. Finalmente, se realiza una fase de pruebas en condiciones controladas, complementada con una validación en simulación digital.

El proyecto no contempla el uso de técnicas de localización global, ni la incorporación de sensores externos de alta complejidad. Se enfoca en establecer una solución funcional, extensi-

ble y compatible con el sistema general del robot móvil, aportando una base sólida para futuras mejoras en localización, navegación autónoma y control inteligente. El sistema será evaluado en instalaciones institucionales y quedará disponible para posteriores desarrollos en el marco de proyectos de investigación y aplicación tecnológica.

1.4. Justificación

La navegación autónoma es un componente esencial en la robótica móvil, con aplicaciones directas en sectores como la logística, la agricultura, la exploración y la automatización industrial. En este contexto, la odometría se establece como una técnica fundamental para estimar la posición y orientación de un robot mediante el análisis del movimiento de sus ruedas, siendo ampliamente utilizada por su simplicidad. No obstante, sus limitaciones por acumulación de errores exigen una implementación cuidadosa y bien comprendida.

Este proyecto permite al estudiante de Ingeniería en Mecatrónica aplicar conocimientos de programación, electrónica y control en un sistema real, fortaleciendo competencias clave para su formación profesional. Desde el punto de vista de la comunidad académica, contribuye al desarrollo de soluciones tecnológicas accesibles y replicables, útiles en entornos de aprendizaje y formación técnica.

Para la ingeniería, la propuesta aporta un enfoque práctico a la integración de sistemas embebidos, modelado cinemático y procesamiento de datos, promoviendo una base tecnológica que puede escalar hacia aplicaciones más complejas como la localización simultánea y el mapeo autónomo. Además, ofrece una plataforma base para futuras investigaciones en robótica inteligente, cerrando brechas entre teoría y práctica y fomentando la vinculación entre academia, industria y sociedad.

Capítulo II

Marco teórico

2.1. Antecedentes

La odometría es una técnica fundamental en la robótica móvil, especialmente en robots con tracción diferencial que utilizan dos ruedas motrices independientes para desplazarse. Esta técnica permite estimar la posición y orientación del robot en un entorno determinado, basándose en los desplazamientos relativos de sus ruedas. No obstante, la acumulación de errores provocada por la imprecisión de los sensores y por simplificaciones en los modelos matemáticos puede afectar significativamente su precisión con el tiempo. En este contexto, el uso de plataformas de desarrollo como ROS facilita integrar, simular y ajustar algoritmos de localización de forma accesible y eficiente.

La implementación de la odometría en robots móviles diferenciales se estudia ampliamente y se aplica en diversos contextos. Por ejemplo, en [2] se analiza el control de velocidad y la odometría de un robot móvil 4WD integrado con ROS 2, lo que destaca la importancia de combinar sistemas de control y localización en plataformas robóticas educativas. Asimismo, se proponen técnicas de fusión sensorial, como la integración de unidades de medición inercial con sensores de rueda, para mejorar la precisión de localización. En [3] se evalúa dicha fusión utilizando el marco AMCL en un robot de cuatro ruedas motrices, lo que demuestra mejoras en la estimación posicional dentro de entornos simulados.

El uso de ROS facilita el desarrollo e implementación de sistemas de odometría, al ofrecer herramientas y bibliotecas para la simulación y control de robots en entornos virtuales como reales. En [4], se implementa navegación autónoma en la plataforma Parallax Eddie con ROS, integrando un sistema de navegación destinado a tareas específicas como la toma de fotografías. En [5], se realiza una simulación de un robot de dos ruedas con tracción diferencial, comparando trayectorias deseadas y reales para verificar la implementación de modelos dinámicos y

de control, utilizando algoritmos de optimización como Levenberg-Marquardt. El codiseño de hardware/software también se propone como solución eficiente. En [6], se desarrolla una arquitectura donde módulos de hardware decodifican señales de encoders y calculan velocidades, mientras un procesador embebido realiza cálculos de posición, lo que libera carga computacional del sistema principal; en entornos donde el GPS no es viable, se emplean alternativas como la odometría visual. En [7], se proponen técnicas de visión por computadora para estimar el movimiento a partir de imágenes, integrando modelos de aprendizaje profundo para robustecer la estimación bajo condiciones adversas. Otros trabajos como [8] evalúan la exactitud de la odometría mediante pruebas con distintas configuraciones de robots móviles, identificando errores sistemáticos y aplicando calibración geométrica para su corrección. En [9], se explora el uso de redes neuronales para predecir la posición y orientación de robots diferenciales, lo que evidencia una tendencia hacia la aplicación de inteligencia artificial para mitigar errores acumulativos.

Por su parte, en [10], se implementa un sistema de localización basado en odometría visual y sensores de rueda, alcanzando una mejora significativa en la estimación de trayectoria. En [11] se plantea un enfoque híbrido que combina odometría clásica, visual e inercial, lo que demuestra una reducción sustancial en la deriva acumulativa durante pruebas en condiciones reales. Como es sabido, ningún sensor es completamente infalible por sí solo; en [12] se analiza cómo, tras una década de avances, la navegación integrada evoluciona desde arquitecturas basadas en filtros de Kalman hasta enfoques de optimización gráfica para entornos interiores complejos. En esta línea, en [13] se describe un control cinemático implementado en TwinCAT 3, con errores inferiores al 0,03 tras procesos de calibración, lo que demuestra que es posible mantener alta precisión en la odometría incluso con hardware de bajo costo. Ahora bien, la precisión física necesita de un middleware capaz de garantizar fluidez de datos entre nodos; en [14], se subraya precisamente la evolución del modelo monolítico de ROS 1 pasando a una nueva arquitectura que prioriza la calidad y seguridad, lo que permite a los nodos de odometría coexistir con percepción, planificación y control en tiempo real.

La modularidad de ROS 2 cobra aún más sentido cuando se puede analizar la comparativa de implementaciones DDS realizada por [15], ya que sus pruebas revelan diferencias notables en lo que respecta a latencia y fiabilidad en factores que llegan a ser críticos cuando la odometría se publica a alta frecuencia para varios robots colaborativos. En [16] se demuestra que una pila de navegación basada en ROS 2 puede recorrer decenas de kilómetros manteniendo la estimación de posición gracias a la mezcla de odometría visual e inercial por ruedas, y la realineación. Por su parte, [17] detallan cómo integrar esa cadena de sensores en un software unificado; los nodos de estimación angular, control dinámico y lógica de evasión de obstáculos se orquestan para

que la odometría ruede sin interrupciones incluso en maniobras bruscas.

Por otro lado en, [18] y [19] se presentan las bibliotecas de ROS más relevantes para el cálculo y adquisición de datos odométricos. Finalmente en [20], se demuestra cómo la modularidad en los sistemas de odometría diferencial, al incorporar controladores autónomos junto con sensores básicos, permite una implementación eficiente y didáctica, adecuada para aplicaciones en contextos académicos y prototipos de investigación.

La Universidad Técnica del Norte identifica la necesidad de contar con plataformas robóticas que sirvan como herramientas de enseñanza. La implementación de un sistema de odometría funcional en un robot móvil diferencial facilita la comprensión de conceptos fundamentales de localización, control de movimiento y navegación autónoma, y permite fortalecer la formación práctica en robótica.

2.2. Fundamentos de robots móviles

2.2.1. Robots móviles diferenciales

Los robots móviles de tracción diferencial representan una de las arquitecturas más comunes en el ámbito de la investigación y el desarrollo de sistemas autónomos, destacándose por su simplicidad mecánica, bajo costo y facilidad de control [21]. El principio operativo de estas plataformas se fundamenta en la diferencia de velocidades entre las ruedas situadas en los extremos izquierdo y derecho, lo que les otorga la capacidad de realizar desplazamientos rectilíneos, giros sobre su propio eje y maniobras complejas con alta eficiencia [22].

Aunque este esquema se asocia frecuentemente con robots de dos ruedas motrices, la configuración de cuatro ruedas también emplea el mismo principio de tracción diferencial. En estos casos, las ruedas de cada lateral operan de manera coordinada para gobernar el movimiento del vehículo, pueden proporcionar una mayor estabilidad mecánica y capacidad de carga respecto a configuraciones de dos ruedas motrices, aunque a costa de una mayor demanda energética y complejidad en la distribución de torque [21].

Debido a estas características, los robots diferenciales son ampliamente utilizados en tareas de localización, seguimiento de trayectorias y navegación móvil. No obstante, la estimación precisa de la posición depende de la correcta implementación de los sistemas de odometría y de la reducción de los errores asociados al movimiento de la plataforma.

2.2.2. Cinemática de robots móviles

La cinemática de los robots móviles estudia el movimiento de estas plataformas en función de sus variables mecánicas y de control, sin considerar las fuerzas o torques que lo producen. Su objetivo principal consiste en establecer relaciones matemáticas entre las velocidades de entrada y los cambios de posición y orientación del robot en un entorno determinado [23, 24].

En aplicaciones de robótica móvil terrestre, el movimiento suele analizarse en un plano bidimensional. La ubicación del robot se describe mediante la pose, compuesta por su posición y orientación respecto a un sistema de referencia global. La pose puede representarse mediante un vector de estado conformado por las coordenadas cartesianas, así como por el ángulo de orientación

$$q = [x, y, \theta], \tag{2.1}$$

donde, x e y y representan la posición del robot en el plano cartesiano, mientras que θ corresponde a su orientación angular [24].

La posición define la ubicación del robot dentro del entorno de trabajo, mientras que la orientación indica la dirección hacia la cual se encuentra desplazando. La combinación de estos parámetros permite representar completamente el estado del robot en un espacio bidimensional.

La cinemática constituye la base para el desarrollo de modelos matemáticos capaces de describir el comportamiento de robots móviles durante su desplazamiento. A partir de estos modelos es posible estimar trayectorias, analizar movimientos y desarrollar sistemas de localización basados en la información proporcionada por sensores de movimiento.

2.2.3. Odometría diferencial

La odometría constituye una de las técnicas más utilizadas para estimar la posición y orientación de un robot móvil a partir de información proporcionada por sensores de movimiento. Esta técnica permite determinar la pose del robot, definida por las coordenadas de posición (x,y) y su orientación (θ), mediante el cálculo de los desplazamientos realizados desde una posición inicial conocida.

En robots con tracción diferencial, la odometría se basa en la medición del desplazamiento de las ruedas motrices mediante sensores denominados encoders. De acuerdo con [12], la odometría diferencial permite calcular la posición y orientación del robot a partir de la diferencia de velocidades o desplazamientos registrados en las ruedas izquierda y derecha. Para ello, se emplean modelos matemáticos que relacionan el movimiento diferencial de las ruedas con los cambios de posición y orientación de la plataforma móvil.

De igual manera, [13] establece que el movimiento de un robot diferencial puede describirse mediante un modelo cinemático que relaciona las velocidades angulares de las ruedas con la velocidad lineal y angular del robot. Este tipo de configuración es ampliamente utilizada en aplicaciones de automatización industrial, logística y sistemas móviles autónomos debido a su simplicidad mecánica y facilidad de control.

La estimación de la pose se realiza a partir de la lectura de los encoders acoplados a las ruedas motrices. Mediante el conteo de pulsos generados durante la rotación de las ruedas, es posible calcular la distancia recorrida por cada una de ellas y posteriormente determinar la posición y orientación del robot a lo largo de su trayectoria.

2.2.4. Fuentes de error en la odometría

Aunque la odometría diferencial constituye un método sencillo y de bajo costo computacional, presenta limitaciones importantes derivadas de la acumulación progresiva de errores que crecen proporcionalmente a la distancia recorrida [24, 25]. Los errores presentes en los sistemas de odometría diferencial pueden clasificarse en errores sistemáticos y errores no sistemáticos. La Tabla 2.1 resume las principales características, causas y posibles métodos de mitigación asociados a cada categoría.

Tabla 2.1: Clasificación de errores en sistemas de odometría diferencial.

Tipos de error	**Descripción**
Errores sistemáticos	Se originan en imperfecciones mecánicas y de calibración que permanecen constantes en el tiempo. Sus principales causas incluyen diferencias en el radio o diámetro de las ruedas, desalineación de ejes, incertidumbre en el ancho de vía y configuraciones incorrectas de los sensores. Estos errores pueden reducirse mediante procedimientos de calibración sistemática, como el método UMBmark [25].
Errores no sistemáticos	Son errores aleatorios e impredecibles producidos por la interacción del robot con el entorno. Entre sus principales causas se encuentran el deslizamiento de las ruedas sobre superficies irregulares, derrapes ocasionados por aceleraciones o frenados bruscos, vibraciones durante el desplazamiento y la presencia de obstáculos menores en el terreno [26].

La acumulación de ambos tipos de errores afecta significativamente la precisión de la localización a lo largo del tiempo, lo que limita la exactitud de la trayectoria estimada. Por esta razón,

diversos autores proponen técnicas de calibración y compensación de errores con el objetivo de mejorar la precisión de la estimación odométrica en aplicaciones robóticas.

2.3. Modelo matemático de la odometría

2.3.1. Modelo cinemático diferencial

La configuración de tracción diferencial es uno de los esquemas más utilizados en la robótica móvil debido a su simplicidad mecánica y versatilidad en entornos interiores [13]. Este modelo describe el movimiento de una plataforma móvil a partir de la velocidad de sus ruedas motrices, sin considerar las fuerzas dinámicas ni las inercias involucradas en el sistema. Por esta razón, constituye una de las aproximaciones más empleadas para el análisis cinemático y la estimación de posición en robots móviles.

En un robot de tracción diferencial, el estado del sistema se controla mediante la variación independiente de las velocidades de las ruedas derecha (V_R) e izquierda (V_L). La relación entre estas velocidades y el movimiento global del robot en términos de su velocidad lineal (v) y angular (ω) se define mediante el modelo cinemático directo [27],

$$v = \frac{V_R + V_L}{2}, \tag{2.2}$$

$$\omega = \frac{V_R - V_L}{B}, \tag{2.3}$$

donde B representa la distancia entre las ruedas motrices del robot.

La ecuación (2.2) indica que la velocidad lineal del robot corresponde al promedio de las velocidades de ambas ruedas, mientras que la ecuación (2.3) establece que la velocidad angular depende de la diferencia entre dichas velocidades. Estas relaciones permiten describir el comportamiento del robot a partir del movimiento individual de cada rueda.

De manera inversa, es posible determinar las velocidades requeridas en cada rueda a partir de una velocidad lineal y angular deseadas. Este procedimiento se conoce como cinemática inversa [28],

$$V_R = v + \frac{B}{2}\omega, \tag{2.4}$$

$$V_L = v - \frac{B}{2}\omega, \tag{2.5}$$

Estas expresiones permiten convertir los comandos de movimiento en velocidades individuales para cada rueda motriz, siendo ampliamente utilizadas en sistemas de control y navegación de robots diferenciales.

Cuando las velocidades de ambas ruedas son distintas, el robot describe una trayectoria curva alrededor de un punto denominado Centro Instantáneo de Curvatura (ICC, por sus siglas en inglés). Por otro lado, cuando ambas ruedas giran a la misma velocidad, el robot genera un movimiento rectilíneo [13]. Este concepto permite representar matemáticamente los movimientos de giro y analizar el comportamiento cinemático del robot durante los cambios de dirección.

En el presente trabajo, el modelo cinemático diferencial constituye la base matemática para la estimación de la odometría del robot móvil. A partir de las velocidades y desplazamientos obtenidos mediante los encoders instalados en las ruedas, es posible calcular la trayectoria recorrida y estimar la posición y orientación durante su desplazamiento.

2.3.2. Modelo cinemático tipo bicicleta

El modelo cinemático tipo bicicleta es una representación simplificada ampliamente utilizada para describir el movimiento planar de vehículos con dirección frontal y de robots móviles cuyo comportamiento puede aproximarse a un sistema equivalente de dos ruedas. Este modelo reduce la geometría del vehículo a una rueda delantera responsable de la dirección y una rueda trasera encargada del desplazamiento, permitiendo analizar su movimiento mediante relaciones geométricas simplificadas [24, 29].

El modelo asume que el movimiento ocurre sobre un plano bidimensional y que no existe deslizamiento lateral significativo en las ruedas. Bajo estas consideraciones, la configuración del vehículo puede describirse mediante el un vector de estado,

$$q = [x, y, \theta], \tag{2.6}$$

donde (x,y) representan la posición del vehículo en el plano cartesiano, mientras que θ corresponde a la orientación del robot respecto a un sistema de referencia global.

La cinemática del modelo tipo bicicleta se expresa mediante un conjunto de ecuaciones diferenciales que describen la evolución de la posición y orientación del vehículo a lo largo del tiempo,

$$\dot{x} = v\cos(\theta), \tag{2.7}$$

$$\dot{y} = v\sin(\theta), \tag{2.8}$$

$$\dot{\theta} = \frac{v}{L}\tan(\delta), \tag{2.9}$$

donde v representa la velocidad lineal del vehículo, L corresponde a la distancia entre ejes y δ es el ángulo de dirección asociado a la rueda delantera.

La ecuación (2.7) describe la variación de la posición del vehículo sobre el eje horizontal x, considerando la velocidad de avance y la orientación instantánea del sistema. De manera similar, la ecuación (2.8) determina la evolución de la posición sobre el eje vertical y, permitiendo representar el desplazamiento del vehículo en el plano de trabajo.

Por otro lado, la ecuación (2.9) establece la velocidad angular del vehículo, indicando cómo cambia su orientación en función de la velocidad lineal, la distancia entre ejes y el ángulo de dirección. Esta relación permite modelar trayectorias curvas y analizar el comportamiento del sistema durante maniobras de giro.

A partir de estas ecuaciones es posible determinar la evolución de la pose del robot a lo largo del tiempo. La integración de dichas expresiones permite estimar la posición y orientación de la plataforma durante su desplazamiento, constituyendo la base para diversos algoritmos de localización y odometría empleados en robótica móvil.

Debido a su simplicidad matemática y bajo costo computacional, el modelo tipo bicicleta ha sido ampliamente utilizado en aplicaciones de navegación autónoma, control de trayectoria y simulación de vehículos móviles [30]. Su capacidad para representar movimientos curvos mediante un conjunto reducido de variables lo convierte en una herramienta adecuada para el análisis cinemático de plataformas robóticas.

En el caso de robots móviles diferenciales, este modelo puede emplearse como una aproximación matemática para representar el comportamiento global del sistema. Aunque la plataforma utilizada en esta investigación corresponde a un robot diferencial de cuatro ruedas, su movimiento puede aproximarse mediante un modelo equivalente de tipo bicicleta, permitiendo simplificar el análisis de la trayectoria y el cálculo de la pose del robot, esto contribuye a la base matemática utilizada para la implementación de la odometría, facilitando la representación de la trayectoria recorrida por el robot dentro del entorno de visualización RViz.

2.4. Sistema ROS 2

2.4.1. Arquitectura ROS 2

Robot Operating System (ROS) es una plataforma de desarrollo de software orientada a la creación de aplicaciones robóticas modulares. Proporciona herramientas y bibliotecas que faci-

litan la integración de sensores, actuadores y algoritmos de procesamiento. Entre sus principales funcionalidades se encuentran el intercambio de mensajes entre procesos, el registro de datos, la visualización de información sensorial, la simulación física y el soporte para lenguajes de programación como Python y C++ [31]. La organización general de los elementos que conforman ROS se presenta en la Fig. 2.1.



Figura 2.1: Elementos principales de la arquitectura de ROS

Una versión ROS 2 consiste en una plataforma mejorada de ROS, el cual es conocido como un kit de desarrollo de software para la robótica. Este, al ser más avanzado, se ha convertido en una herramienta necesaria para la investigación y desarrollo robótico debido a su flexibilidad y escalabilidad [14].

Según [17] una de las ventajas principales de ROS es su arquitectura modular basada en nodos. Cada nodo representa un proceso que realiza una función específica dentro del sistema, el cual se puede encargar de leer datos de los sensores, calcular trayectorias o controlar actuadores. Esta modularidad permite dividir el desarrollo del sistema en diferentes tareas independientes que pueden ser desarrolladas y probar su efectividad. En [14] se muestra que en ROS 2, la arquitectura fue rediseñada desde cero para que sea más adecuada para aplicaciones industriales, incluyendo mejoras en la seguridad, comunicación en tiempo real, tolerancia a fallos de red y diferentes medidas de soporte para sistemas embebidos. La diferencia más notable entre ROS y ROS 2 es que este utiliza DDS (Data Distribution Service) lo que permite una comunicación robusta y confiable entre nodos.

2.4.2. Nodos ROS

Según [15] los nodos en ROS son las unidades funcionales fundamentales; cada uno se ejecuta como un proceso independiente el cual se comunica con otros nodos mediante un sistema de paso de mensajes. Estos son capaces de publicar topics, ofrecer servicios síncronos o realizar tareas de larga duración con retro alimentación. En [14] se explica que el nodo es una unidad organizativa importante que permite al usuario razonar sobre un sistema complejo. Por ejemplo, un nodo puede encargarse de leer la odometría desde los encoders de las ruedas, mientras que otro nodo se encarga del mapeo y otro de la planificación de rutas.

2.4.3. Herramientas y paquetes de ROS 2 para el desarrollo del sistema de odometría

El Robot Operating System 2 constituye un ecosistema de desarrollo modular que proporciona un conjunto estandarizado de herramientas, bibliotecas y protocolos para la construcción de sistemas robóticos complejos. A diferencia de su predecesor, ROS 2 se fundamenta en el estándar Data Distribution Service lo que garantiza una comunicación distribuida, robusta y con calidad de servicio adecuada para aplicaciones en tiempo real [32].

2.4.3.1 Topics de ROS

En ROS 2, la comunicación entre procesos (nodos) se realiza mediante tres mecanismos principales: topics, services y actions. El sistema de odometría se basa primordialmente en la comunicación mediante topics, los cuales operan bajo un modelo de publicación-suscripción asíncrono [32]. Este esquema permite que el nodo de lectura de sensores publique datos de alta frecuencia (como los pulsos de los encoders) sin esperar una respuesta inmediata del nodo de procesamiento [33].

La Tabla 2.2 detalla las características de estos mecanismos de comunicación en el marco de una arquitectura distribuida.

Tabla 2.2: Mecanismos de comunicación en la arquitectura ROS 2.

Mecanismo	Definición técnica
Topics (Temas):	Comunicación asíncrona unidireccional. Es el estándar para flujos continuos de datos como lecturas de sensores, estado de odometría y señales de control.
Services (Servicios):	Interacción síncrona basada en solicitud-respuesta. Se emplea para configuraciones o comandos puntuales que requieren confirmación, como el reinicio de la posición odométrica.
Actions (Acciones):	Comunicación orientada a tareas de larga duración con retroalimentación continua. Permiten monitorear el progreso y cancelar procesos, como el desplazamiento a una coordenada específica.

En el sistema desarrollado, la comunicación mediante topics permite transmitir la información obtenida desde los encoders hacia el nodo encargado del cálculo de la odometría. Posteriormente, los resultados de posición, orientación y velocidad se publican para su visualización y análisis dentro del entorno ROS 2 [14].

2.4.3.2 Mensajería y transformaciones en odometría diferencial

La representación de la información de movimiento en sistemas robóticos móviles requiere tanto un estándar de mensajería como un mecanismo de transformación entre marcos de referencia. En ROS 2, el mensaje `nav/msgs/msg/Odometry` constituye el formato oficial para transmitir la estimación de la pose y la velocidad del robot [33], mientras que la librería TF2 asegura la coherencia espacial entre los diferentes sistemas de coordenadas [34]. La integración de ambos elementos es esencial para garantizar la interoperabilidad, la fusión sensorial y la correcta interpretación de datos.

En la Tabla 2.3 se presentan los principales componentes de la mensajería de odometría en ROS 2, junto con la librería TF2.

Tabla 2.3: Componentes de la mensajería de odometría en ROS 2.

Elemento	Definición técnica
Pose con covarianza	Representa la posición (x, y, z) y la orientación del robot mediante cuaterniones, evitando problemas como el gimbal lock. Incluye una matriz de covarianza 6×6 que cuantifica la incertidumbre de la estimación, utilizada en algoritmos de fusión sensorial.
Twist con covarianza	Describe las velocidades lineales y angulares del robot (velocidades de cuerpo). En un robot diferencial, estas se derivan directamente de la diferencia de velocidad entre las ruedas izquierda y derecha.
Librería TF2	Gestiona las transformaciones entre sistemas de referencia. Publica la relación entre el frame `odom` (referencia global) y el frame `base_link` (centro del robot), permitiendo que sensores posicionen sus lecturas correctamente en el espacio permitiendo su visualización en RViz.

2.4.3.3 Herramienta RViz

RViz2 es una herramienta de visualización gráfica de ROS 2 que permite representar información publicada por los nodos de un sistema robótico. Su uso facilita el análisis del comportamiento de algoritmos de localización, navegación y control mediante la representación de datos dentro de un entorno visual [34].

En el presente trabajo, RViz2 se utiliza para visualizar la información odométrica publicada en el *topic* `/odom`. Esta información corresponde a la estimación de posición, orientación y velocidad generada por el nodo de odometría a partir de los datos obtenidos desde los encoders.

La visualización en RViz2 permite observar la trayectoria estimada del robot durante las pruebas simuladas y reales. De esta manera, se puede verificar el comportamiento del modelo cinemático e identificar desviaciones acumulativas entre la trayectoria estimada y el desplazamiento esperado del robot. Esta herramienta constituye un apoyo para el análisis de los resultados obtenidos durante la evaluación del sistema de odometría.

2.4.4. Encoders y sensores en odometría

Los encoders son transductores que convierten el desplazamiento angular de una rueda o eje en señales eléctricas. Estas señales permiten registrar el número de vueltas o fracciones de vuelta realizadas por las ruedas, a partir de las cuales se puede estimar la distancia recorrida y

el cambio de orientación de un robot móvil [35].

Existen distintos tipos de encoders, entre ellos los encoders incrementales y los absolutos cada uno tiene diferentes características como se muestra en la Tabla 2.4 .

Tabla 2.4: Comparación entre encoder incremental y absoluto.

Característica	**Encoder incremental**	**Encoder absoluto**
Tipo de salida	Por pulsos eléctricos por cada incremento angular.	Código digital único para cada posición angular.
Información de posición	Relativa (es necesaria una referencia inicial).	Absoluta (cada posición tiene un valor único).
Reinicio	Pierde la posición si se apaga o se reinicia.	Mantiene la posición incluso después de apagarlo.
Resolución	Determinada por el número de pulsos por vuelta (PPR).	Determinada por el número de bits del código de salida.
Complejidad del sistema	Es simple y más fácil de implementar.	Más complejo y requiere interpretación del código.

En los sistemas de odometría, los datos generados por los encoders son adquiridos por un controlador o nodo de software y utilizados para calcular el desplazamiento de las ruedas. A partir de esta información, el modelo cinemático estima la posición, orientación y velocidad del robot en cada instante de tiempo.

En la Fig. 2.2 se presenta el encoder, empleado como sensor para la detección de pulsos asociados al giro de la rueda. La imagen fue obtenida de [36].



Figura 2.2: Módulo FC-03

Capítulo III

Marco metodológico

3.1. Enfoque de la investigación

En el presente trabajo de titulación se adopta un enfoque metodológico de tipo experimental y aplicado, orientado al diseño, implementación y validación de un sistema de odometría diferencial para un robot móvil mediante el uso de ROS 2. Este enfoque permite plantear una solución práctica al problema planteado, relacionado con la estimación del movimiento del robot en tiempo real, además de facilitar el aprendizaje de los estudiantes de Mecatrónica.

Para el desarrollo de la investigación se emplea, como un apoyo el enfoque documental, el cual consistió en la recopilación y análisis de información proveniente de fuentes bibliográficas y técnicas. Esta revisión permitió fundamentar la formulación del modelo matemático y cinemático, tomando como referencia el modelo de tipo bicicleta, así como orientar su posterior implementación computacional mediante nodos desarrollados en ROS 2, los cuales cumplen funciones específicas dentro del sistema.

El desarrollo metodológico del trabajo se estructuró en tres fases principales:

1) Modelado matemático y definición del sistema de odometría diferencial.

2) Implementación del sistema en ROS 2 mediante nodos programados principalmente en el lenguaje Python.

3) Validación experimental del sistema a través de pruebas de movimiento controladas y el análisis gráfico básico de los resultados obtenidos.

3.2. Diseño de la investigación

El diseño de la investigación se estructuró bajo un enfoque secuencial como antes se nombro, organizado en fases y actividades claramente definidas, orientadas al desarrollo progresivo del sistema de odometría diferencial. Cada fase fue planteada con el objetivo de garantizar el cumplimiento de la meta propuesta y a su vez garantizar una transición lógica del trabajo de titulación desde el análisis teórico hasta la validación experimental del sistema de odometría.

3.2.1. Fase 1: Análisis y modelado del sistema de odometría diferencial

Esta fase estuvo orientada al estudio y recopilación de datos de los fundamentos teóricos y matemáticos necesarios para el desarrollo del sistema de odometría. Se realizó una revisión bibliográfica enfocada en la odometría diferencial, la cinemática de robots móviles y los errores asociados a este tipo de estimación.

Actividad 1.1: Revisión bibliográfica sobre odometría diferencial

Se recopiló información proveniente de artículos científicos, libros y documentación técnica general relacionada con modelos cinemáticos de robots móviles, principios de la odometría, funcionamiento, importancia y sus principales fuentes de error.

Actividad 1.2: Formulación del modelo matemático y cinemático

A partir de la información analizada, se formuló el modelo matemático de la odometría diferencial, considerando un robot móvil con cuatro ruedas motrices independientes y se definieron las ecuaciones necesarias para la estimación de la posición y orientación del robot.

Actividad 1.3: Definición de la arquitectura general del sistema

Se estableció la arquitectura del sistema bajo el paradigma modular de ROS 2, definiendo los nodos, tópicos y tipos de mensajes necesarios para la adquisición de datos, el procesamiento de la odometría y la visualización de resultados

3.2.2. Fase 2: Implementación del sistema en ROS 2

Esta fase se llevo a cabo la implementación computacional del sistema de Odometría diferencial en la micro cpu, utilizando ROS 2 y como lenguaje de programación a Python.

Actividad 2.1: Desarrollo del nodo de lectura de encoders

Se implemento dos nodos encargados de la lectura de datos de los encoders, ya sean en modalidad real o simuladas, permitiendo la obtención de los desplazamientos o velocidades de cada rueda.

Actividad 2.2: Implementación del nodo de cálculo de odometría

Se desarrollo el nodo responsable de aplicar las ecuaciones cinemáticas del modelo diferencial para calcular la pose del robot a partir de los datos de los encoders.

Actividad 2.3: Publicación y visualización de la información de odometría

Se configuro la publicación de los datos de posición y orientación mediante un mensaje Odometry, facilitando su integración con la herramienta de visualización RViz.

3.2.3. Fase 3: Validación experimental y análisis de resultados

La fase final planteada estuvo orientada a la validación del sistema de odometría mediante pruebas experimentales con datos simulados y reales en entornos controlados.

Actividad 3.1: Definición de trayectorias y condiciones de prueba

Se establecieron diferentes formas de trayectorias de movimiento como son los desplazamientos lineales, circulares y aleatorios, con el fin de evaluar el comportamiento del sistema ante distintos escenarios.

Actividad 3.2: Ejecución de pruebas experimentales

Se realizaron pruebas de movimiento del robot diferencial aplicando movimientos mediante el uso del APK, registrando los datos de posición y orientación estimados por el sistema de odometría.

Actividad 3.3: Análisis gráfico de resultados

Los datos obtenidos fueron presentados gráficamente para analizar las trayectorias estimadas y evaluar el desempeño del sistema, identificando posibles errores y limitación inherentes al método de odometría.

3.3. Descripción de la plataforma robótica utilizada

3.3.1. Antecedentes de la plataforma robótica

El presente trabajo de grado emplea una plataforma robótica móvil desarrollada en trabajos de titulación previos en la carrera de Ingeniería Mecatrónica de la Universidad Técnica del Norte [37, 38, 39]. Dichos proyectos tuvieron como objetivo principal el diseño mecánico del chasis, la construcción de la estructura y la integración inicial de los componentes electrónicos. En este capítulo se describen únicamente las características de la plataforma necesarias para comprender la implementación del sistema de odometría diferencial desarrollado en esta investigación.

3.3.2. Características mecánicas del robot

El robot cuenta con un chasis metálico de configuración rectangular, diseñado para soportar cargas de hasta 100 kg. Sus dimensiones aproximadas son de 60 cm de largo, 45 cm de ancho y 25 cm de altura. La estructura está fabricada en aluminio y acero, lo que garantiza rigidez y resistencia. El sistema de ruedas está compuesto por cuatro neumáticos de 20 cm de diámetro, montados sobre ejes paralelos con una separación de 40 cm entre los laterales [38]. El radio de las ruedas y la distancia entre los laterales constituyen parámetros fundamentales para el cálculo del modelo de odometría diferencial.

3.3.3. Sistema de tracción diferencial

La plataforma emplea una configuración skid-steer de cuatro ruedas motrices. La dirección se logra mediante la variación de velocidad entre las ruedas de cada lado, lo que permite giros sobre su propio eje y desplazamientos con alta maniobrabilidad. Este esquema de tracción es el fundamento sobre el cual se aplican los modelos matemáticos de odometría diferencial, la configuración skid-steer se modela mediante una aproximación cinemática equivalente a un robot diferencial, considerando las velocidades promedio del lado izquierdo y derecho del vehículo. [37].

3.3.4. Sistema de adquisición de datos

Para el cálculo de la odometría se utilizan encoders ópticos acoplados a los motores de tracción. Estos sensores generan pulsos digitales proporcionales al giro de cada rueda, los cuales son contabilizados por un microcontrolador Arduino Mega 2560 [37]. La información obtenida se transmite a la unidad de procesamiento principal, donde se ejecutan los algoritmos de odometría diferencial. Este sistema constituye la base experimental sobre la cual se validan los modelos matemáticos planteados en la investigación.

En la Figura 3.1 se muestra el diagrama del sistema de adquisición de datos, donde se observa la conexión de los encoders al Arduino Mega 2560 y la transmisión de la información hacia la Jetson Orin Nano 8GB.

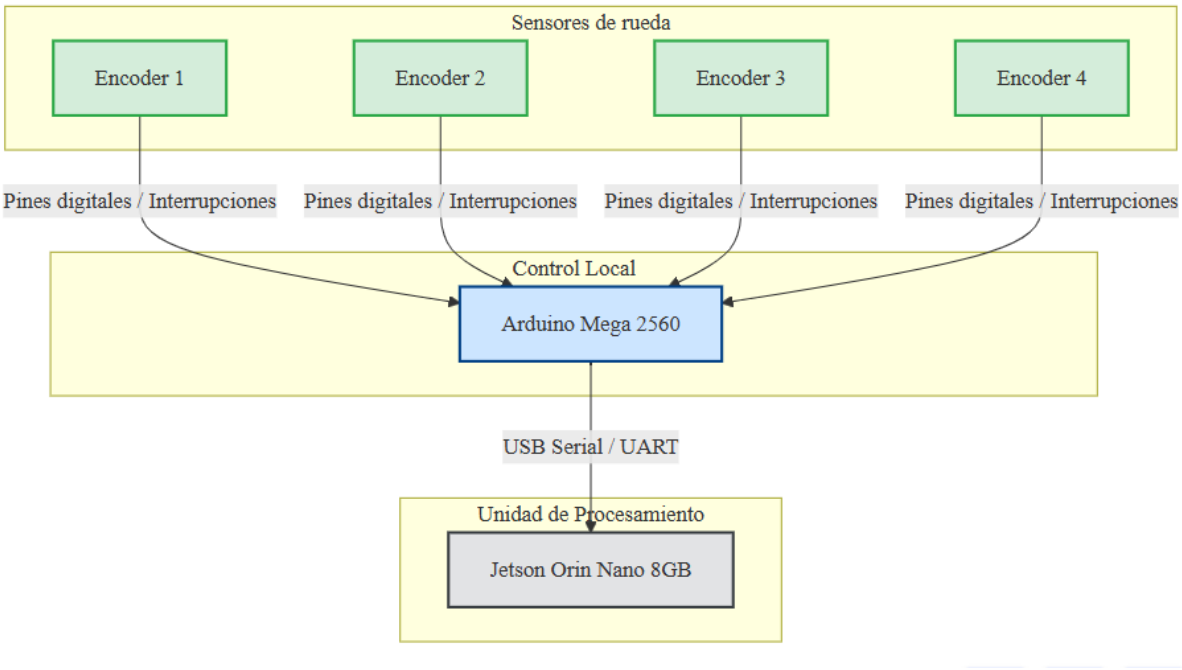


Figura 3.1: Diagrama del sistema de adquisición de datos

3.3.5. Unidad de procesamiento Jetson Orin Nano

La unidad de procesamiento seleccionada corresponde a la **Jetson Orin Nano 8GB**, mostrada en la Fig. 3.2. Este dispositivo actúa como el núcleo del sistema, ejecutando el sistema operativo Ubuntu junto con el entorno de desarrollo **ROS 2**. En esta plataforma se implementan los nodos encargados del cálculo de la odometría diferencial, la gestión de los datos provenientes de los encoders y la coordinación con el microcontrolador **Arduino Mega 2560**, responsable de la adquisición de pulsos generados por las ruedas del robot.

La capacidad de cómputo de la Jetson Orin Nano permite integrar algoritmos de control, percepción y localización en tiempo real, asegurando un flujo de información continuo y confiable para la validación experimental del modelo de odometría diferencial. Además, su arquitectura flexible y escalable facilita la incorporación de nuevas aplicaciones y módulos sin necesidad de modificar la estructura principal del sistema, lo que la convierte en una plataforma adecuada para proyectos de investigación y futuras extensiones orientadas a la mejora del robot diferencial.

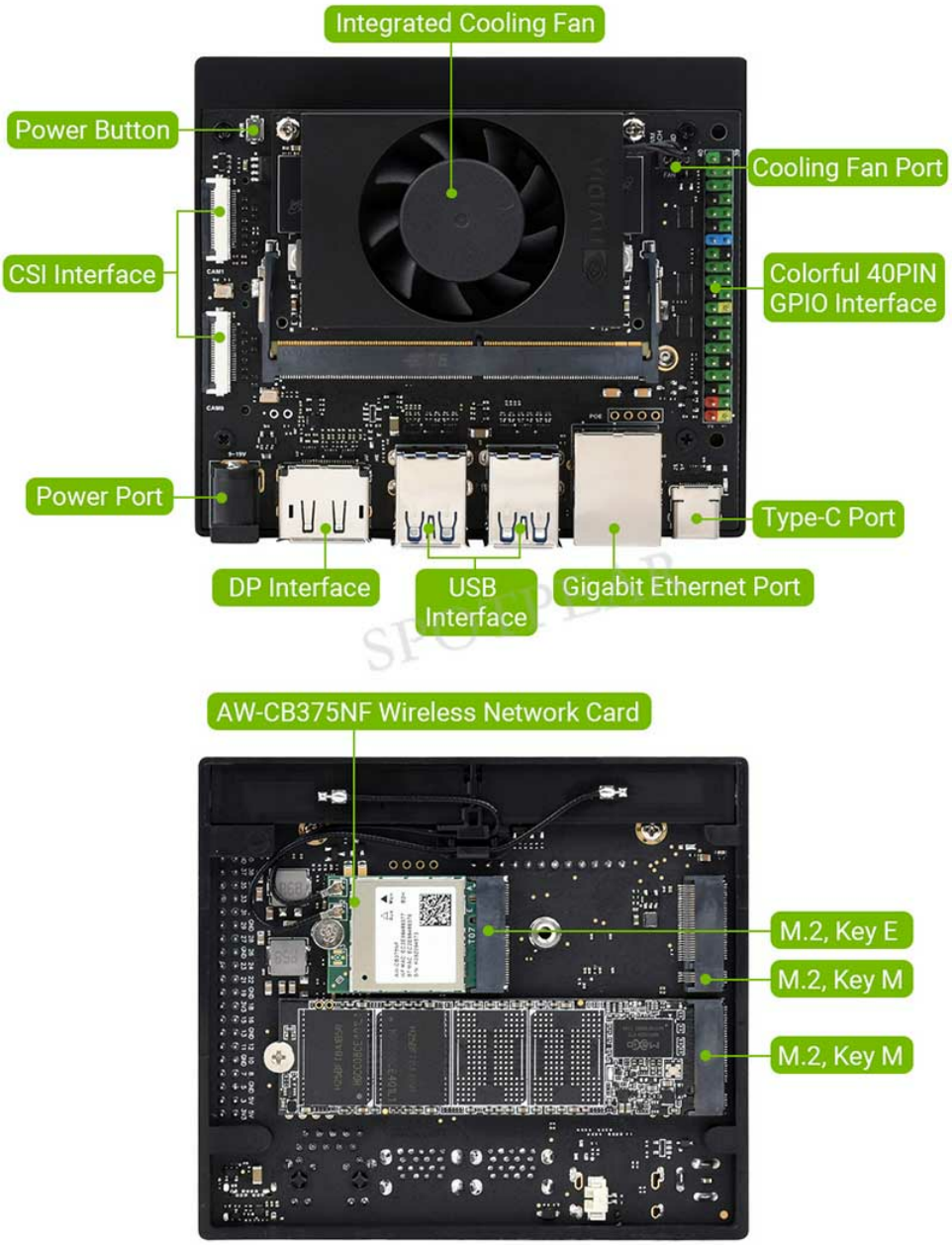


Figura 3.2: Jetson Orin Nano 8GB developer kit [1]

3.3.6. Integración hardware-software

La arquitectura del robot combina el microcontrolador Arduino para la adquisición de datos de los encoders y la Jetson Orin Nano como unidad central de procesamiento. Esta integración garantiza un sistema robusto para la estimación de posición y orientación, constituyendo la base

experimental sobre la cual se validan los modelos matemáticos de odometría diferencial. Además, la compatibilidad entre los componentes electrónicos asegura una comunicación estable y un correcto funcionamiento del robot en su conjunto.

3.4. Arquitectura del sistema de odometría desarrollado

El sistema diseñado está compuesto por un conjunto de nodos ROS 2 desarrollados en Python, los cuales interactúan mediante tópicos estandarizados. A continuación, se describen sus principales componentes para su funcionamiento.

3.4.1. Arquitectura general del sistema

El sistema de odometría es desarrollado a partir del modelo cinemático equivalente de un robot diferencial e implementado mediante ROS 2 utilizando una arquitectura modular basada en nodos. La estimación de la posición y orientación del robot se realiza a partir de las velocidades angulares de las ruedas, las cuales son obtenidas mediante encoders simulados o físicos y posteriormente procesadas por el nodo de cálculo de odometría.

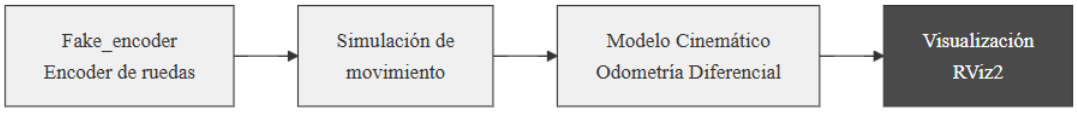


Figura 3.3: Diagrama de bloques del sistema de odometría planteado

El sistema opera bajo un esquema modular como se presenta en la Fig. 3.3 , donde cada nodo cumple una función bien definida: generación de comandos de velocidad, simulación de encoders, cálculo de odometría y publicación de transformaciones espaciales (gráfica). Esta separación funcional permite analizar de forma individual el comportamiento de cada componente y facilita la validación del modelo matemático propuesto.

Las principales variables estimadas corresponden a la posición del robot en el plano bidimensional (x, y) y su orientación angular θ. Esta información se publica en tiempo real mediante mensajes del tipo `nav_msgs/Odometry`, y posteriormente se visualiza en el entorno RViz.

Con el propósito de facilitar la comprensión del funcionamiento del sistema desarrollado, en la Figura 3.4 se presenta la arquitectura general implementada en ROS 2, donde se muestran los principales nodos, los tópicos de comunicación y el flujo de información desde la adquisición de datos hasta la visualización de la odometría.

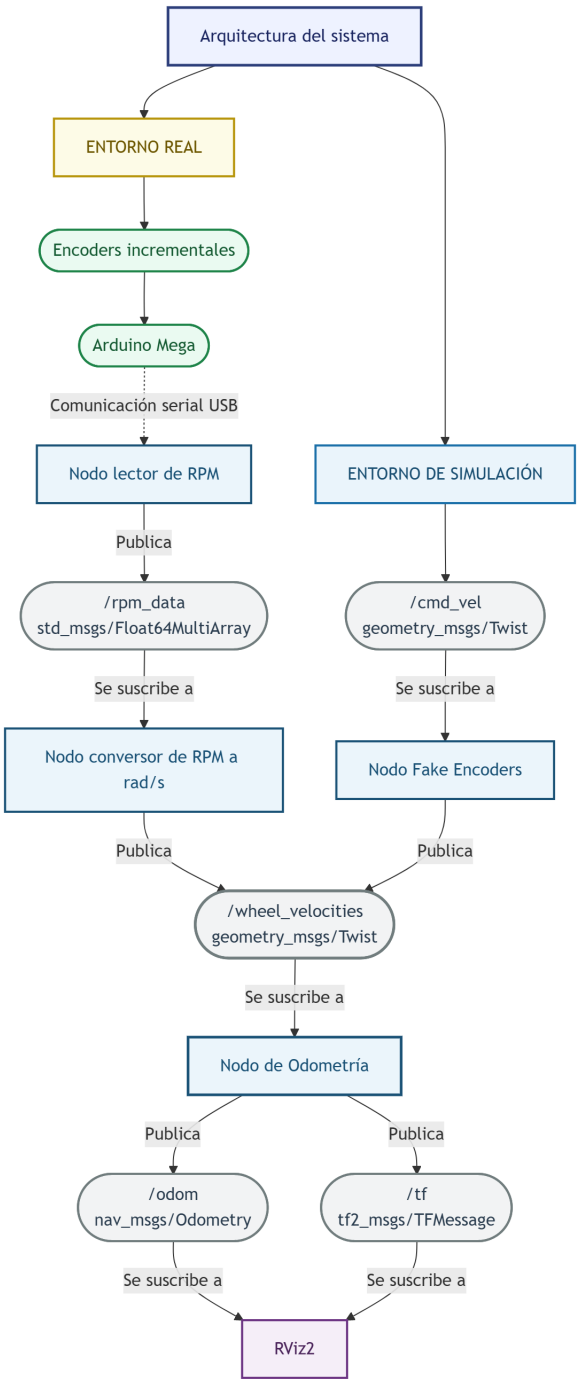


Figura 3.4: Arquitectura funcional del sistema de odometría diferencial desarrollado

Como se observa, el sistema admite dos fuentes de información para el cálculo de la odometría. Durante la etapa de simulación, las velocidades angulares de las ruedas son generadas mediante un nodo de encoders simulados a partir de los comandos de velocidad. En las pruebas

experimentales, dichas velocidades son obtenidas a partir de los encoders físicos instalados en el robot, cuyos datos son adquiridos por el Arduino Mega y convertidos a velocidad angular antes de ser enviados al nodo de cálculo de odometría. En ambos casos, la información procesada se publica mediante mensajes del tipo `nav_msgs/Odometry`, permitiendo su visualización en RViz2 mediante la publicación de la transformación correspondiente.

3.4.2. Sistema operativo Ubuntu 22.04.5 LTS

El desarrollo y la ejecución del sistema de odometría se realiza sobre el sistema operativo Ubuntu 22.04.5 LTS, el cual es ampliamente utilizado en aplicaciones de robótica debido a su estabilidad, soporte a largo plazo y amplia compatibilidad con el ecosistema de ROS 2. Esta versión LTS ofrece actualizaciones de seguridad y mantenimiento extendidas, lo que garantiza un entorno confiable para el desarrollo de sistemas robóticos.

Ubuntu proporciona un entorno adecuado para la compilación, ejecución y depuración de nodos ROS 2, facilitando la gestión de paquetes, dependencias y bibliotecas necesarias para el correcto funcionamiento del sistema. Del mismo modo, permite la integración directa con herramientas de visualización y análisis gráfico como RVIz y utilidades de monitoreo, las cuales fueron empleadas para evaluar el comportamiento el robot y analizar los resultados obtenidos.

3.4.3. Entorno de desarrollo ROS 2

Se emplea **ROS 2 (Robot Operating System 2)** como middleware principal, debido a que proporciona una arquitectura distribuida basada en nodos que facilita la comunicación entre los diferentes componentes del sistema. Esta arquitectura permite desarrollar aplicaciones modulares, donde cada nodo ejecuta una función específica y se comunica mediante tópicos, favoreciendo la escalabilidad, el mantenimiento y la integración con otros subsistemas robóticos [14].

En el sistema desarrollado, ROS 2 permite implementar de manera independiente los nodos encargados de la adquisición de datos provenientes de los encoders, el cálculo de la odometría diferencial y la publicación de la posición y orientación estimadas del robot. Esta organización modular facilita el desarrollo, las pruebas individuales de cada componente y la integración del sistema completo.

La comunicación entre los nodos se realiza mediante *topics* bajo el modelo publicador-suscriptor (*publish/subscribe*), permitiendo el intercambio de información sin acoplamiento directo entre los procesos. Los datos obtenidos de los encoders son procesados por el nodo de

odometría, el cual calcula la pose del robot utilizando el modelo cinemático y publica los resultados mediante mensajes del tipo `nav_msgs/Odometry`.

Por otra parte, ROS 2 proporciona herramientas que simplifican la supervisión y visualización del sistema. En este trabajo se utilizó **RViz2** para representar gráficamente la trayectoria estimada del robot y verificar el comportamiento del algoritmo de odometría durante las pruebas experimentales.

3.4.4. Nodo de encoders simulados

Además, un nodo de encoders simulados es desarrollado con el propósito de validar el modelo matemático de la odometría diferencial en un entorno controlado, eliminando la influencia de errores asociados al hardware. Este nodo recibe como entrada los comandos de velocidad lineal y angular publicados en el tópico `/cmd_vel`, a partir de los cuales calcula las velocidades angulares correspondientes a las ruedas izquierda y derecha mediante las ecuaciones cinemáticas del robot diferencial.

Para el cálculo se consideran como parámetros geométricos el radio de las ruedas y la distancia entre ambos laterales del robot. Las velocidades angulares obtenidas se expresan en rad/s y se publican mediante el tópico `/wheel_velocities` utilizando mensajes del tipo `Float32MultiArray`. Esta información constituye la entrada del nodo de cálculo de odometría.

El empleo de este nodo hizo posible verificar el correcto funcionamiento del algoritmo de odometría, facilitando la validación del modelo cinemático antes de realizar las pruebas con los encoders físicos del robot.

3.4.5. Nodo de encoders reales

La adquisición de datos provenientes de los encoders físicos se implementa mediante dos nodos independientes. El primero se encarga de establecer la comunicación serial entre el microcontrolador Arduino Mega 2560 y la Jetson Orin Nano, recibiendo continuamente las revoluciones por minuto (RPM) correspondientes a las ruedas izquierda y derecha del robot. Los valores recibidos son validados y publicados en el tópico `/rpm_data` para su posterior procesamiento.

Debido a que el modelo matemático de la odometría diferencial utiliza velocidades angulares expresadas en rad/s, se desarrolla un segundo nodo encargado de convertir automáticamente las RPM recibidas a velocidad angular mediante la relación de

$$\omega = \frac{2\pi \cdot RPM}{60}. \tag{3.1}$$

Una vez realizada la conversión, las velocidades angulares de ambas ruedas se publican en el tópico `/wheel_velocities`, constituyendo la entrada utilizada por el nodo de cálculo de odometría.

La separación de estas funciones en dos nodos independientes facilita desacoplar el proceso de adquisición de datos del proceso de conversión de unidades, facilitando las tareas de depuración, mantenimiento y reutilización del sistema.

3.4.6. Nodo de cálculo de odometría

El nodo de cálculo de odometría constituye el componente principal del sistema desarrollado. Su función consiste en estimar la posición y orientación del robot a partir de las velocidades angulares de las ruedas publicadas en el tópico `/wheel_velocities`.

Inicialmente, el nodo calcula la velocidad lineal y la velocidad angular del robot mediante las ecuaciones del modelo cinemático diferencial empleando el radio de las ruedas y la distancia entre los laterales del vehículo. Posteriormente, utilizando el intervalo de tiempo transcurrido entre dos mediciones consecutivas, se realiza la integración numérica de las ecuaciones cinemáticas para actualizar las coordenadas (x), (y) y el ángulo de orientación (θ).

Con la información obtenida se construye un mensaje del tipo `nav_msgs/Odometry`, el cual contiene tanto la pose estimada del robot como su velocidad lineal y angular. Finalmente, este mensaje es publicado en el tópico `/odom`, permitiendo que otros nodos del entorno ROS 2 puedan utilizar la información generada para tareas de visualización, navegación o registro de datos.

3.4.7. Publicación de mensajes Odometry

La información estimada por el nodo de odometría se publica mediante mensajes del tipo `nav_msgs/Odometry` a través del tópico `/odom`. Este mensaje contiene la posición del robot, su orientación representada mediante las velocidades lineal y angular estimadas, siguiendo el estándar establecido por ROS 2.

Este tipo de mensaje garantiza la compatibilidad del sistema con herramientas del ecosistema ROS 2, permitiendo que la información de odometría pueda ser utilizada por aplicaciones de visualización, localización y planificación de trayectorias sin requerir adaptaciones adicionales.

3.4.8. Visualización mediante RViz

La visualización de la trayectoria estimada se realiza mediante RViz 2, utilizando un nodo encargado de publicar la transformación espacial entre los sistemas de referencia `odom` y `base_link`. Para ello, el nodo se suscribe al tópico `/odom`, extrae la posición y orientación calculadas por el algoritmo de odometría y genera continuamente la transformación correspondiente mediante la librería `tf2`.

La publicación de esta transformación permite que RViz 2 represente en tiempo real la posición y orientación del robot dentro del sistema de coordenadas global, facilitando la supervisión del comportamiento del algoritmo durante las pruebas de simulación y experimentación real.

Capítulo IV

Resultados y análisis

4.1. Configuración experimental

4.1.1. Entorno de simulación

Con el propósito de validar el funcionamiento del algoritmo de odometría en un entorno controlado, se realizaron pruebas experimentales empleando un conjunto de nodos de simulación desarrollados en ROS 2. Este entorno permite evaluar el comportamiento del sistema sin la influencia de factores físicos propios de la plataforma robótica, tales como vibraciones, deslizamiento de las ruedas, ruido en los sensores o retardos de comunicación.

Debido a que durante una etapa del desarrollo la plataforma robótica fue utilizada simultáneamente en otro trabajo de titulación, se implementa un nodo de encoders simulados capaz de generar las velocidades angulares de las ruedas a partir de los comandos de movimiento publicados en el tópico `/cmd_vel`. Esta estrategia facilita continuar con el desarrollo y la validación del algoritmo de odometría de manera independiente del hardware, garantizando la continuidad del proceso de investigación.

En este entorno, el nodo de encoders simulados proporciona datos ideales y libres de perturbaciones externas, los cuales son procesados por el mismo nodo de cálculo de odometría desarrollado para la plataforma física. De esta manera, es posible verificar que la estimación de la posición y orientación correspondiera con el comportamiento esperado del modelo cinemático antes de realizar las pruebas experimentales sobre el robot real.

4.1.2. Entorno real

Una vez validado el algoritmo bajo condiciones ideales, se realiza pruebas experimentales utilizando la plataforma robótica diferencial. En este escenario, la estimación de la odometría se efectúa a partir de las lecturas de los encoders incrementales instalados en las ruedas del robot.

La adquisición de los datos es realizada mediante un Arduino Mega 2560, encargado de registrar los pulsos generados por los encoders durante el desplazamiento del robot. Posteriormente, esta información es enviada mediante comunicación serial hacia la Jetson Orin Nano, donde se ejecutaron dos nodos desarrollados en ROS 2: el primero recibe los datos provenientes del microcontrolador y el segundo realiza la conversión de las velocidades medidas en revoluciones por minuto hacia velocidades angulares expresadas en rad/s, necesarias para el cálculo de la odometría.

Finalmente, las velocidades angulares obtenidas son procesadas por el nodo de cálculo de odometría desarrollado en esta investigación, generando la estimación de la posición y orientación del robot, la cual es visualizada en tiempo real mediante RViz2.

4.1.3. Parámetros de prueba

Tabla 4.1: Comparación entre simulación y entorno real.

Parámetro	**Simulación**	**Entorno real**
Sistema operativo	Ubuntu 22.04.5 LTS	Ubuntu 22.04.5 LTS
Middleware	ROS 2 Humble	ROS 2 Humble
Fuente de datos	Nodo de encoders simulados	Encoders incrementales
Procesamiento	Nodo de odometría	Nodo de odometría
Visualización	RViz2	RViz2
Tipo de pruebas	Rectilínea, circular y aleatoria	Rectilínea, circular y libre
Variables evaluadas	x, y, θ	x, y, θ

Como se observa en la Tabla 4.1 en ambos escenarios se mantiene la misma arquitectura de procesamiento desarrollada en el Capítulo III, modificándose únicamente el origen de los datos de entrada. En el entorno de simulación las velocidades son generadas mediante el nodo de encoders simulados, mientras que en el entorno real proviene de los encoders físicos instalados en el robot. Esta configuración permite realizar una comparación directa entre el comportamiento ideal del modelo y su desempeño bajo condiciones reales de operación.

4.2. Pruebas con encoders simulados

Una vez configurado el entorno de simulación, se realiza diferentes pruebas experimentales utilizando los encoders simulados desarrollados para esta investigación. El propósito de estas pruebas es validar el correcto funcionamiento del algoritmo de odometría diferencial bajo condiciones ideales, donde las velocidades de las ruedas no se ven afectadas por factores propios del hardware.

Se diseña las pruebas considerando tres escenarios de movimiento representativos: desplazamiento rectilíneo, trayectoria circular y movimiento con trayectorias variables. Esta metodología permite evaluar el comportamiento del algoritmo frente a diferentes condiciones de operación antes de su implementación sobre la plataforma robótica real.

4.2.1. Movimiento rectilíneo

La primera prueba consiste en evaluar el comportamiento del sistema de odometría durante un desplazamiento rectilíneo. Para ello se aplica una velocidad lineal constante manteniendo la velocidad angular igual a cero, condición que provoca que las ruedas motrices giren a la misma velocidad y que el robot se desplace en línea recta.

Bajo estas condiciones, el modelo cinemático establece que la orientación del robot debe permanecer prácticamente constante, mientras que la posición debe incrementarse únicamente sobre el eje correspondiente al sentido de avance. Esta prueba permite verificar el correcto cálculo de la velocidad lineal y la integración de la posición cuando no existen cambios en la orientación del robot.

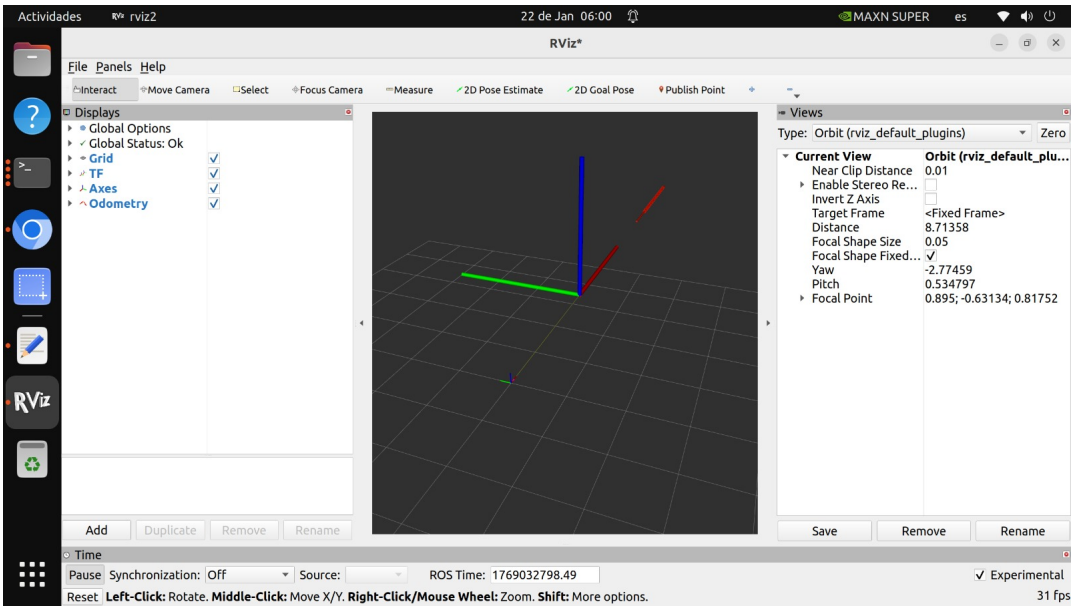


Figura 4.1: Trayectoria estimada en movimiento rectilíneo (encoders simulados)

Como se observa en la Figura 4.1, la trayectoria estimada corresponde a un desplazamiento rectilíneo, manteniendo una orientación prácticamente constante durante todo el recorrido. Este comportamiento coincide con la respuesta esperada del modelo cinemático empleado para el cálculo de la odometría.

Al utilizar encoders simulados, las velocidades angulares de las ruedas permanecen constantes y libres de perturbaciones externas, por lo que la trayectoria obtenida no presenta desviaciones laterales ni cambios inesperados de orientación. Estos resultados validan el correcto funcionamiento del algoritmo de integración para movimientos rectilíneos y constituyen una referencia ideal para la comparación posterior con las pruebas realizadas en el entorno real.

4.2.2. Movimiento circular

La segunda prueba tiene como finalidad validar la estimación de la trayectoria cuando el robot realiza desplazamientos curvos. Para ello se aplica simultáneamente una velocidad lineal constante y una velocidad angular distinta de cero, generando un movimiento circular continuo.

En este tipo de trayectoria intervienen tanto el cálculo de la velocidad lineal como la actualización continua de la orientación del robot, por lo que esta prueba permite verificar el comportamiento conjunto de ambas variables durante el proceso de integración de la odometría.

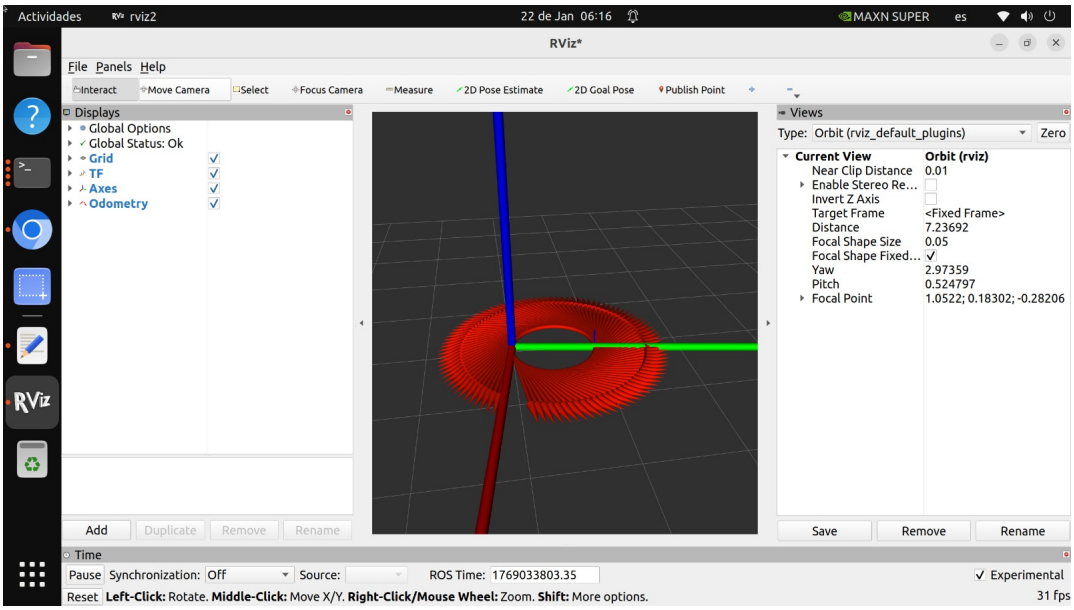


Figura 4.2: Trayectoria estimada en movimiento circular (encoders simulados)

La Figura 4.2 muestra que la trayectoria obtenida mantiene un comportamiento circular estable, evidenciando que la orientación del robot evoluciona de manera continua y consistente con los comandos de velocidad aplicados. La forma de la trayectoria confirma que el algoritmo calcula adecuadamente la variación angular y la utiliza para actualizar la posición del robot durante el desplazamiento.

Al igual que en la prueba anterior, el empleo de encoders simulados elimina la influencia de perturbaciones físicas, permitiendo comprobar que el comportamiento observado corresponde exclusivamente al modelo matemático propuesto.

4.2.3. Movimiento aleatorio

Con el propósito de evaluar el comportamiento del algoritmo bajo condiciones de operación más cercanas a un entorno real, se realiza las pruebas empleando comandos de velocidad lineal y angular variables. Para ello se utilizaron nodos generadores de velocidad desarrollados en esta investigación, los cuales publican valores dentro de rangos previamente establecidos, produciendo cambios continuos tanto en la velocidad de avance como en la dirección del movimiento.

Este escenario representa una condición de mayor complejidad respecto a las pruebas anteriores, debido a que el algoritmo debe integrar continuamente variaciones simultáneas de posición y orientación durante todo el recorrido.

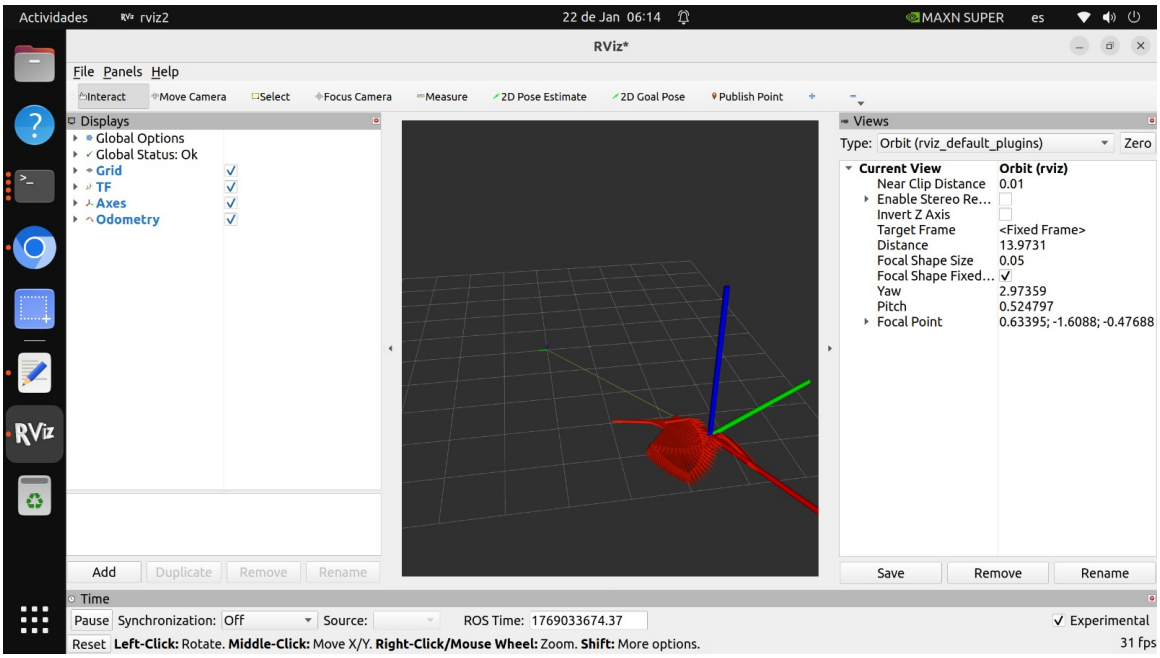


Figura 4.3: Trayectoria estimada en movimiento aleatorio (encoders simulados)

Como se observa en la Figura 4.3, la trayectoria obtenida presenta cambios continuos de dirección acordes con las velocidades aplicadas durante la simulación. A pesar de la variabilidad del movimiento, la estimación de la posición mantiene una evolución continua y coherente con el modelo cinemático implementado.

Los resultados obtenidos evidencian que el algoritmo de odometría es capaz de integrar correctamente movimientos combinados de traslación y rotación, incluso cuando las velocidades cambian de manera dinámica. Debido a que los datos provienen de encoders simulados, la trayectoria generada refleja el comportamiento ideal esperado del modelo matemático, constituyendo una referencia para comparar posteriormente el desempeño observado con los encoders físicos de la plataforma robótica.

En conjunto, las pruebas realizadas con encoders simulados permite verificar el comportamiento del algoritmo de odometría bajo diferentes condiciones de movimiento. Los resultados obtenidos constituyen una validación inicial del modelo desarrollado, al demostrar que la estimación de la posición y orientación del robot es consistente con las trayectorias esperadas en un entorno libre de perturbaciones físicas. La Tabla 4.2 resume los principales resultados obtenidos en cada una de las pruebas realizadas.

Tabla 4.2: Resumen de las pruebas realizadas con encoders simulados.

Prueba	Objetivo de la prueba	Resultado obtenido	Validación
Movimiento rectilíneo	Verificar la estimación de la trayectoria durante un desplazamiento lineal	Trayectoria recta con orientación constante	Correcta
Movimiento circular	Validar la actualización de la orientación durante un giro continuo	Trayectoria circular estable	Correcta
Movimiento aleatorio	Evaluar el comportamiento del algoritmo ante cambios dinámicos de velocidad	Trayectoria coherente con los comandos aplicados	Correcta

4.3. Pruebas con encoders reales

Una vez verificado el correcto funcionamiento del algoritmo de odometría mediante el uso de encoders simulados, se realiza pruebas experimentales sobre la plataforma robótica real con el propósito de validar el comportamiento del sistema bajo condiciones reales de operación.

A diferencia del entorno de simulación, en estas pruebas las velocidades angulares de las ruedas fueron obtenidas a partir de los encoders incrementales instalados en la plataforma robótica. Las mediciones son adquiridas mediante un Arduino Mega 2560 y posteriormente enviadas a la Jetson Orin Nano, donde son procesadas por los nodos desarrollados para la adquisición de datos y conversión de velocidades antes de ser utilizadas por el nodo de cálculo de odometría.

Este escenario incorpora las condiciones propias del sistema físico, incluyendo pequeñas variaciones en las mediciones de los encoders, retardos asociados a la comunicación serial y características mecánicas de la plataforma, permitiendo evaluar el desempeño del algoritmo fuera del entorno ideal de simulación.

4.3.1. Movimiento rectilíneo

La primera prueba en el entorno real consiste en evaluar el comportamiento de la odometría durante un desplazamiento rectilíneo. Para ello se aplica una velocidad lineal constante manteniendo la velocidad angular igual a cero, replicando las mismas condiciones utilizadas durante la validación en simulación.

El objetivo de esta prueba es verificar si la trayectoria estimada por el algoritmo mantiene un comportamiento similar al observado con los encoders simulados, permitiendo además identificar posibles diferencias ocasionadas por las características propias de la plataforma física.

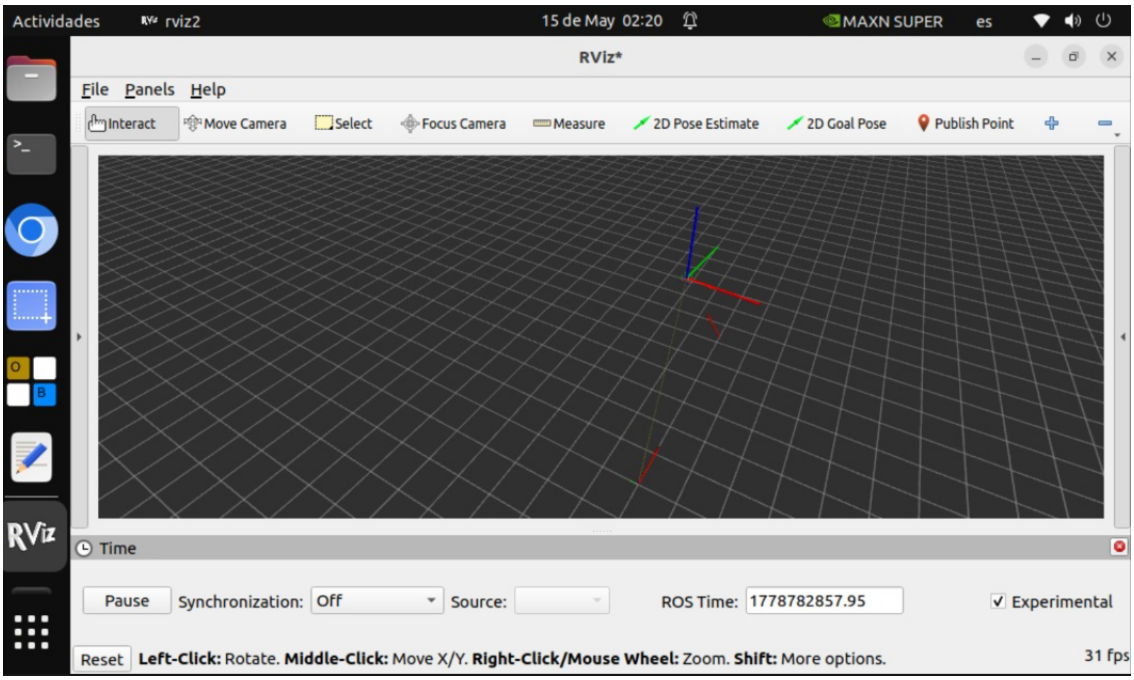


Figura 4.4: Trayectoria estimada en movimiento rectilíneo (encoders reales)

Como se observa en la Figura 4.4, la trayectoria obtenida mantiene un comportamiento predominantemente rectilíneo; sin embargo, se aprecia una ligera tendencia hacia el lado derecho conforme aumenta la distancia recorrida. A diferencia de las pruebas realizadas con encoders simulados, esta desviación evidencia la influencia de factores físicos que no se encuentran representados en el modelo ideal.

Entre las posibles causas de este comportamiento se encuentran pequeñas diferencias en la velocidad efectiva de los motores, desgaste mecánico, irregularidades en la superficie de desplazamiento, distribución del peso de la plataforma y variaciones en el nivel de carga de la batería. De igual manera, las mediciones provenientes de los encoders pueden presentar pequeñas fluctuaciones debido al proceso de adquisición y transmisión de los datos desde el Arduino hacia la Jetson Orin Nano.

A pesar de estas diferencias, la trayectoria estimada conserva el comportamiento general esperado para un desplazamiento rectilíneo, lo que confirma que el algoritmo de odometría propuesto continúa proporcionando una estimación consistente de la posición del robot bajo condiciones reales de operación.

4.3.2. Movimiento circular

La segunda prueba consiste en evaluar el comportamiento del sistema durante un desplazamiento circular utilizando velocidades lineales y angulares constantes. Esta configuración permite verificar la capacidad del algoritmo para actualizar simultáneamente la posición y la orientación del robot durante un movimiento curvo.

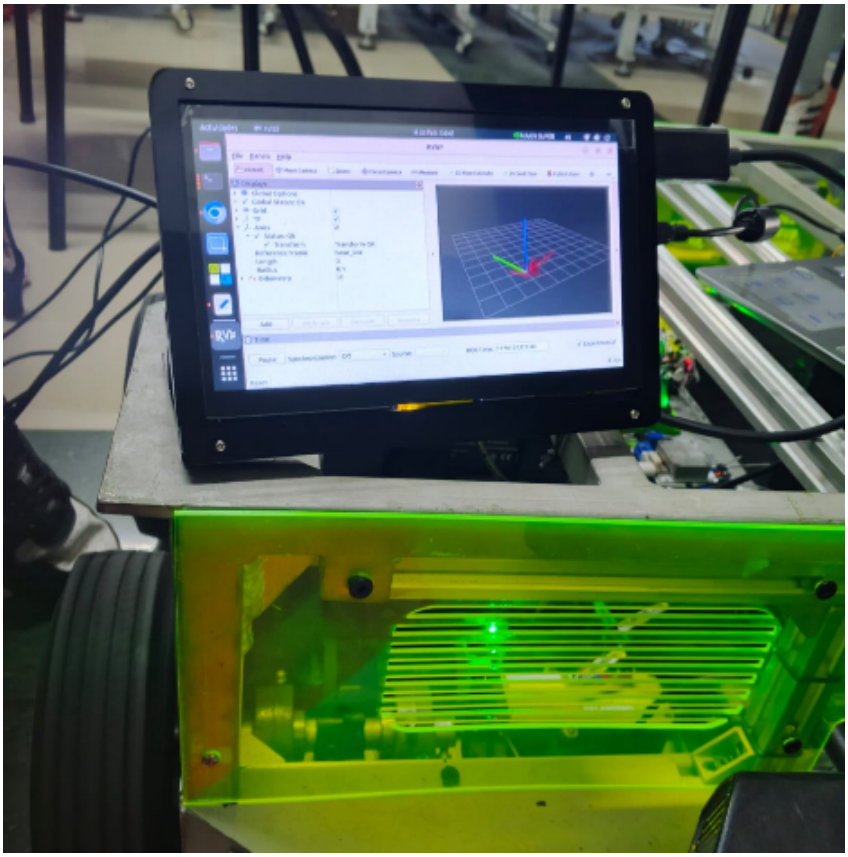


Figura 4.5: Trayectoria estimada en movimiento circular (encoders reales)

Como se observa en la Figura 4.5, la trayectoria obtenida presenta un comportamiento circular similar al observado durante las pruebas de simulación. No obstante, se identifican pequeñas irregularidades mínimas en el recorrido que no se encontraban presentes en el entorno ideal. Estas variaciones como antes se menciono pueden atribuirse a las características propias de la plataforma física.

A pesar de estas perturbaciones, el algoritmo mantiene una estimación continua de la trayectoria y de la orientación del robot, demostrando que el sistema es capaz de operar adecuadamente utilizando información proveniente de sensores reales.

4.3.3. Trayectoria libre bajo operación manual

Como prueba final de validación se realiza un recorrido libre mediante operación manual del robot, permitiendo variar continuamente tanto la velocidad lineal como la velocidad angular durante el desplazamiento.

A diferencia de las pruebas anteriores, donde los movimientos son previamente definidos, este escenario reproduce de manera más cercana las condiciones reales de operación de la plataforma, ya que el algoritmo debe estimar continuamente la posición del robot frente a cambios constantes de velocidad y dirección.



Figura 4.6: Trayectoria estimada durante operación manual (encoders reales)

Como se observa en la Figura 4.6, la trayectoria estimada presenta múltiples cambios de dirección y velocidad, manteniendo un comportamiento continuo durante todo el recorrido. La forma irregular de la trayectoria corresponde al movimiento realizado manualmente, evidenciando que el algoritmo responde de manera adecuada ante variaciones simultáneas de traslación y rotación.

Durante las pruebas se ha identificado pequeñas desviaciones respecto al comportamiento ideal observado en simulación, propias del funcionamiento de la plataforma robótica.

En conjunto, las pruebas realizadas sobre la plataforma robótica permiten validar el funcionamiento del algoritmo bajo condiciones reales de operación. Aunque se observa diferencias respecto al entorno de simulación, estas corresponden a las limitaciones propias del sistema físico y no comprometen el funcionamiento general de la arquitectura.

Si bien se observa pequeñas diferencias respecto a las trayectorias obtenidas en simulación, estas corresponden a las limitaciones propias del sistema físico y no comprometen el funcionamiento general de la arquitectura. La Tabla 4.3 presenta un resumen de los resultados obtenidos en cada uno de los escenarios evaluados.

Tabla 4.3: Resumen de las pruebas realizadas con encoders reales.

Prueba	**Objetivo**	**Resultado observado**	**Validación**
Movimiento rectilíneo	Validar el desplazamiento lineal utilizando encoders físicos	Ligera desviación hacia la derecha, manteniendo el comportamiento esperado	Correcta
Movimiento circular	Verificar la actualización de posición y orientación durante trayectorias curvas	Trayectoria circular con pequeñas variaciones	Correcta
Trayectoria libre	Evaluar el algoritmo bajo condiciones dinámicas de operación	Estimación continua de la trayectoria	Correcta

4.4. Comparación entre simulación y entorno real

Una vez realizadas las pruebas experimentales utilizando tanto encoders simulados como encoders reales, se efectúa una comparación entre ambos escenarios con el propósito de identificar las diferencias existentes entre el comportamiento ideal del modelo matemático y su implementación sobre la plataforma robótica. Este análisis permite evaluar el desempeño del sistema de odometría bajo condiciones controladas y determinar la influencia que ejercen las características propias del hardware sobre la estimación de la posición y orientación del robot.

4.4.1. Comparación de trayectorias

Con el propósito de facilitar el análisis comparativo, la Tabla 4.4 resume las principales diferencias observadas entre las trayectorias obtenidas mediante encoders simulados y aquellas registradas durante las pruebas realizadas sobre la plataforma robótica.

Los resultados obtenidos muestran que, en ambos escenarios, el algoritmo de odometría genera trayectorias coherentes con los comandos de velocidad aplicados. Sin embargo, las tra-

Tabla 4.4: Comparación entre las trayectorias obtenidas en simulación y en el entorno real.

Aspecto evaluado	Simulación	Entorno real
Trayectoria rectilínea	Recta y estable	Ligera desviación hacia la derecha
Trayectoria circular	Circular uniforme	Pequeñas irregularidades
Movimiento dinámico	Cambios suaves	Cambios influenciados por el hardware
Ruido en mediciones	No presente	Presente
Repetibilidad	Alta	Moderada

yectorias obtenidas mediante encoders simulados presentan un comportamiento prácticamente ideal, caracterizado por desplazamientos suaves y una elevada repetibilidad, debido a que los datos utilizados durante la simulación se encuentran libres de perturbaciones externas.

Por el contrario, las trayectorias obtenidas durante las pruebas con encoders reales presentan pequeñas variaciones respecto al comportamiento ideal. Estas diferencias se manifiestan principalmente en ligeras desviaciones durante los movimientos rectilíneos y pequeñas irregularidades en las trayectorias curvas, evidenciando la influencia de las condiciones físicas de la plataforma sobre la estimación de la odometría.

A pesar de estas diferencias, el comportamiento general observado en ambos escenarios mantiene una alta correspondencia, lo que permite concluir que el algoritmo propuesto reproduce adecuadamente el modelo cinemático tanto en simulación como en condiciones reales de operación.

4.4.2. Análisis de desviaciones observadas

Las diferencias observadas entre las trayectorias obtenidas en simulación y aquellas registradas durante las pruebas reales corresponden a las limitaciones de cualquier sistema de odometría basado exclusivamente en encoders incrementales. Mientras que el entorno de simulación representa un modelo ideal, la plataforma física incorpora diferentes fuentes de incertidumbre que afectan la estimación de la posición.

Durante las pruebas de movimiento rectilíneo se identifica una ligera tendencia del robot a desviarse hacia el lado derecho conforme aumentaba la distancia recorrida. Esta característica ha sido observada durante el desarrollo de las pruebas experimentales y puede asociarse principalmente a las condiciones mecánicas propias de la plataforma, más que al algoritmo de odometría.

Entre los factores que pueden contribuir a este comportamiento se encuentran posibles pe-

queñas diferencias en la velocidad efectiva de los motores, distribución del peso sobre la plataforma, irregularidades presentes en la superficie de desplazamiento, variaciones en el nivel de carga de la batería y pequeñas fluctuaciones en las mediciones provenientes de los encoders durante el proceso de adquisición y transmisión de los datos.

En consecuencia, las desviaciones observadas forman parte del comportamiento esperado de una plataforma robótica real y no comprometen el funcionamiento general del algoritmo desarrollado, el cual mantiene una estimación continua y consistente de la trayectoria del robot durante todas las pruebas realizadas.

4.4.3. Limitaciones del sistema

Si bien el sistema de odometría desarrollado ha hecho posible estimar satisfactoriamente la posición y orientación del robot durante las diferentes pruebas experimentales, es importante reconocer las limitaciones a este tipo de solución.

La implementación desarrollada utiliza como única fuente de información las mediciones provenientes de los encoders incrementales instalados en las ruedas motrices. En consecuencia, cualquier error acumulado durante la estimación de la posición se propaga a lo largo del recorrido, fenómeno característico de los sistemas de odometría diferencial.

Del mismo modo, el modelo cinemático empleado asume condiciones ideales de desplazamiento, por lo que no considera fenómenos presentes en la plataforma física.

Otra limitación identificada corresponde a la frecuencia de actualización de las mediciones reales, la cual depende del proceso de adquisición realizado por el Arduino Mega 2560, de la comunicación serial con la Jetson Orin Nano y del procesamiento ejecutado por los nodos desarrollados en ROS 2. Estas condiciones introducen pequeñas variaciones temporales que no se presentan durante la simulación.

4.5. Análisis de resultados

Una vez finalizadas las pruebas experimentales y el análisis, se evalúa el cumplimiento de los objetivos específicos planteados. Para ello, los resultados obtenidos se analizan en función de las evidencias experimentales presentadas en las secciones anteriores, permitiendo determinar el alcance y la validez del sistema de odometría diferencial propuesto.

El primer objetivo específico consistió en desarrollar el modelo matemático de odometría diferencial para un robot móvil de cuatro ruedas utilizando la aproximación del modelo cinemático de bicicleta.

Los resultados obtenidos durante las pruebas con encoders simulados evidencian que la implementación realizada reproduce correctamente el comportamiento esperado del modelo matemático. Las trayectorias rectilíneas, circulares y con velocidades variables presentaron una evolución consistente con los comandos de movimiento aplicados, verificando que las ecuaciones cinemáticas fueron implementadas correctamente.

Se considera que el primer objetivo específico fue cumplido satisfactoriamente, al demostrarse que el algoritmo desarrollado estima de forma adecuada la posición y orientación del robot bajo condiciones ideales de operación.

El segundo objetivo específico estuvo orientado a integrar el sistema de adquisición de datos provenientes de los encoders físicos con el algoritmo de odometría propuesto.

Las pruebas realizadas sobre la plataforma robótica demostraron que la arquitectura desarrollada demostró adquirir, procesar y convertir correctamente la información generada por los encoders incrementales para su utilización dentro del nodo de odometría. Durante los diferentes escenarios de prueba, el sistema mantuvo una estimación continua de la posición y orientación del robot, validando el funcionamiento conjunto de los nodos desarrollados para la comunicación entre el Arduino Mega 2560 y la Jetson Orin Nano.

Aunque se observaron pequeñas desviaciones respecto al comportamiento ideal obtenido en simulación, estas corresponden a las limitaciones propias de la plataforma física y no afectan el funcionamiento general del algoritmo desarrollado. Por lo tanto, se considera que el segundo objetivo específico fue alcanzado.

El tercer objetivo específico consistió en validar experimentalmente el sistema de odometría diferencial mediante pruebas realizadas tanto en simulación como sobre la plataforma robótica.

Los resultados obtenidos permitieron comprobar que el algoritmo propuesto mantiene un comportamiento consistente en ambos escenarios. Las pruebas con encoders simulados sirvieron para verificar el funcionamiento del modelo matemático bajo condiciones ideales, mientras que las pruebas con encoders reales permitieron identificar el efecto de las condiciones físicas propias del robot sobre la estimación de la trayectoria.

La comparación entre ambos escenarios evidenció que las diferencias observadas corresponden principalmente a factores mecánicos, variaciones en las mediciones de los encoders y limitaciones de la odometría basada únicamente en sensores de rueda. No obstante, el sistema fue capaz de estimar de manera continua la posición y orientación del robot durante todos los recorridos realizados.

En función de los resultados experimentales obtenidos, se concluye que el tercer objetivo específico fue cumplido, validando el desempeño del sistema desarrollado tanto en simulación como en condiciones reales de operación.

Capítulo V

Conclusiones y recomendaciones

5.1. Conclusiones

El desarrollo del presente trabajo de grado facilitó implementar y validar un sistema de odometría diferencial para un robot móvil de cuatro ruedas utilizando el modelo cinemático equivalente de bicicleta y una arquitectura basada en ROS 2. La implementación realizada demostró que es posible estimar la posición y orientación del robot a partir de la información proporcionada por encoders incrementales, obteniendo resultados consistentes tanto en un entorno de simulación como sobre la plataforma robótica real.

Los resultados obtenidos durante las pruebas con encoders simulados evidenciaron que el algoritmo desarrollado reproduce correctamente el comportamiento esperado del modelo matemático. Las trayectorias rectilíneas, circulares y con velocidades variables presentaron un comportamiento coherente con los comandos de movimiento aplicados, validando la correcta implementación de las ecuaciones cinemáticas y del proceso de integración de la odometría.

La validación experimental realizada sobre la plataforma robótica hizo posible comprobar que el algoritmo mantiene un funcionamiento consistente al utilizar información proveniente de encoders reales. Aunque se observaron pequeñas desviaciones respecto a las trayectorias ideales obtenidas en simulación, estas fueron atribuidas a factores del sistema físico, como diferencias en la respuesta de los motores, variaciones mecánicas de la plataforma, condiciones de la superficie de desplazamiento y limitaciones propias de la odometría basada exclusivamente en encoders.

La comparación entre el entorno de simulación y el entorno real permitió comprobar que las diferencias observadas corresponden principalmente a las características del hardware y no a deficiencias en la implementación del algoritmo. En consecuencia, los resultados experimentales confirman la validez del sistema desarrollado y demuestran que la arquitectura propuesta

constituye una base adecuada para futuras aplicaciones de navegación y localización en robots móviles.

Finalmente, se concluye que los objetivos específicos planteados al inicio de la investigación fueron alcanzados satisfactoriamente. La implementación desarrollada no solo demostró validar el funcionamiento del algoritmo de odometría diferencial, sino que también proporcionó una arquitectura modular basada en ROS 2 que puede extenderse mediante la incorporación de nuevos sensores, algoritmos de fusión sensorial o sistemas de navegación autónoma.

5.2. Recomendaciones

1. Realizar una calibración periódica de los parámetros geométricos de la plataforma robótica, tales como el diámetro efectivo de las ruedas, la distancia entre ruedas y la resolución de los encoders, ya que pequeñas variaciones en estos parámetros afectan directamente la precisión de la estimación de la odometría.

2. Durante las pruebas experimentales se recomienda emplear sistemas de control que permitan mantener velocidades más uniformes, reduciendo la influencia del operador sobre la trayectoria recorrida y facilitando la repetibilidad de los experimentos.

3. Se recomienda realizar las pruebas sobre superficies planas y con condiciones controladas, minimizando el efecto del deslizamiento de las ruedas y otras perturbaciones mecánicas que puedan introducir errores adicionales en la estimación de la posición.

5.3. Trabajo a futuro

El sistema desarrollado abre múltiples líneas de investigación y mejora que pueden ser abordadas en estudios posteriores. Una de las principales extensiones propuestas consiste en la incorporación de técnicas de fusión sensorial, mediante la implementación de algoritmos como el Filtro de Kalman Extendido (EKF), el cual permitirá combinar la odometría diferencial con datos provenientes de sensores adicionales como unidades de medición inercial (IMU). Sensores LiDAR o cámaras. Esta integración contribuiría a reducir la deriva acumulativa característica de los sistemas de odometría durante trayectorias prolongadas.

Otra línea de investigación consiste en integrar el sistema desarrollado con algoritmos de localización y mapeo simultáneo (SLAM), utilizando paquetes ampliamente empleados dentro del ecosistema ROS 2. Esta integración permitiría generar mapas del entorno mientras se mejora

la estimación de la posición mediante información obtenida de sensores externos y características del ambiente.

De igual manera, la arquitectura modular desarrollada constituye una base para la implementación de sistemas de navegación autónoma mediante el uso del paquete Nav2 de ROS 2. Esto permitiría que la plataforma planifique rutas, evite obstáculos y alcance objetivos definidos de forma completamente autónoma.

Finalmente, se propone el desarrollo de una interfaz gráfica que permita visualizar en tiempo real la trayectoria estimada del robot, las variables asociadas al sistema de odometría y diferentes indicadores de desempeño. Esta herramienta facilitaría el monitoreo del sistema y ampliaría las posibilidades de utilización de la plataforma en actividades de investigación, docencia y desarrollo de nuevos proyectos.

Bibliografía

[1] RaspberryMe, "Nvidia jetson orin nx : Kit de développement avec 16 go de ram et 128 go de ssd nvme," 2026. [En línea].

[2] L. J. S. M. Silva, F. Rodrigues, "Control de velocidad y odometría de un robot móvil 4wd usando ros 2," *Revista de Robótica Educativa*, vol. 5, pp. 123–134, 2020.

[3] S. E. Silva, R. A. Oliveira, V. R. Motta, C. T. Valadao, D. F. Gamarra, and M. A. Cuadros, "Fusión de imu y odometría en robots móviles: Implementación en ros con amcl," *International Journal of Robotics and Automation*, vol. 13, pp. 45–59, 11 2020.

[4] C. O. H. Anas, "Navegación autónoma de robots con ros en parallax eddie: Implementación y pruebas," in *Proceedings of the 2021 IEEE Robotics and Automation Conference* (H. Anas, ed.), pp. 200–210, 8 2021.

[5] M. SIWEK, J. PANASIUK, L. BARANOWSKI, W. KACZMAREK, and S. BORYS, "Trajectory tracking control of a mobile robot with the ros system," *Problems of Mechatronics Armament Aviation Safety Engineering*, vol. 13, pp. 67–84, 12 2022.

[6] A.-D. Suárez-Gómez and W.-J. Pérez-Holguín, "Sistema de odometría basado en codiseño h/s para un robot móvil diferencial," in *IV Congreso Internacional de Ciencias Básicas e Ingeniería CICI2022*, pp. 1–4, 2022.

[7] L. Herrera, "Odometría visual con ia: Innovando las tecnologías de localización," *KOMPUTER SAPIENS*, vol. 1, pp. 25–29, 1 2025.

[8] A. Kelly, "Mobile robot localization from large-scale appearance mosaics," *The International Journal of Robotics Research*, vol. 19, pp. 1104–1125, 11 2000.

[9] K. Puentes and L. Morales, "Trajectory tracking of a mobile robot using a pid controller combined with neural networks," in *2023 IEEE Seventh Ecuador Technical Chapters Meeting (ECTM)*, pp. 1–6, IEEE, 10 2023.

[10] C. Golban, P. Cobarzan, and S. Nedevschi, "A comparison study on replacing stereo disparity with lidar in visual odometry methods," in *2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pp. 277–282, IEEE, 9 2020.

[11] X. Zou, J. Gao, H. Li, Y. Zhang, F. Liu, H. Wu, and R. Yao, "Novel visual odometry method for water autonomous navigation," in *2022 2nd International Conference on Computation, Communication and Engineering (ICCCE)*, pp. 13–17, IEEE, 11 2022.

[12] Y. Zhuang, X. Sun, Y. Li, J. Huai, L. Hua, X. Yang, X. Cao, P. Zhang, Y. Cao, L. Qi, J. Yang, N. El-Bendary, N. El-Sheimy, J. Thompson, and R. Chen, "Multi-sensor integrated navigation/positioning systems using data fusion: From analytics-based to learning-based approaches," *Information Fusion*, vol. 95, pp. 62–90, 7 2023.

[13] M. Ferreira, L. Moreira, and A. Lopes, "Differential drive kinematics and odometry for a mobile robot using twincat," *Electronic Research Archive*, vol. 31, pp. 1789–1803, 2023.

[14] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, 5 2022.

[15] V. F. Raña, "Sistema de comunicaciones dds entrerobots en ros 2," 7 2023.

[16] S. Macenski, F. Martín, R. White, and J. G. Clavero, "The marathon 2: A navigation system," 7 2020.

[17] A. O. Prasad, P. Mishra, U. Jain, A. Pandey, A. Sinha, A. S. Yadav, R. Kumar, A. Sharma, G. Kumar, K. H. Salem, A. Sharma, and A. K. Dixit, "Design and development of software stack of an autonomous vehicle using robot operating system," *Robotics and Autonomous Systems*, vol. 161, p. 104340, 3 2023.

[18] O. S. R. Foundation, "Ros wiki," 2 2025.

[19] J. Newans, "Ros overview," 2025.

[20] A. J. J. Muñoz and D. A. R. Flore, *Simulación y control de un robot cuadrúpedo multiservicio en entornos regulares utilizando Robot Operating System*. PhD thesis, Escuela Superior Politécnica Del Litoral, 2025.

[21] L. K. J, H. S, G. K.B, D. K. K, and A. VM, "Two and four wheels differential racing bots," in *2025 International Conference on Computing Technologies (ICOCT)*, pp. 1–6, IEEE, 6 2025.

[22] L. Tagliavini, G. Colucci, A. Botta, P. Cavallone, L. Baglieri, and G. Quaglia, "Wheeled mobile robots: State of the art overview and kinematic comparison among three omnidirectional locomotion strategies," *Journal of Intelligent & Robotic Systems*, vol. 106, p. 57, 11 2022.

[23] T. T. Mac, C.-Y. Lin, N. G. Huan, L. D. Nhat, P. C. Hoang, and H. hong Hai, "Hybrid slam-based exploration of a mobile robot for 3d scenario reconstruction and autonomous navigation," *Acta Polytechnica Hungarica*, vol. 18, pp. 197–212, 2021.

[24] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*. MIT Press, 2nd ed., 2011.

[25] A. Georgiev and P. Allen, "Localization methods for a mobile robot in urban environments," *IEEE Transactions on Robotics*, vol. 20, pp. 851–864, 10 2004.

[26] L. Herrera, "Odometría visual con ia: Innovando las tecnologías de localización," *Del Cerebro a la Máquina: Sinapsis Digitales con Re-des Neuronales de Impulso*, 2025.

[27] H. A. Poonawala and M. W. Spong, "Time-optimal velocity tracking control for differential drive robots," *Automatica*, vol. 85, pp. 153–157, 2017.

[28] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey," *Robotica*, vol. 33, no. 3, pp. 463–497, 2015.

[29] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, 2009.

[30] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[31] A. Innovation, "Sistema ros para el movimiento de robots," 3 2023.

[32] L. Hanenko and V. Zhebka, "Development of the navigation system of an autonomous mobile robot using ros 2," *Cybersecurity Education Science Technique*, p. 498, 6 2025.

[33] S. Macenski, T. Moore, D. Lu, A. Merzlyakov, and M. Ferguson, "From the desks of ros maintainers: A survey of modern & capable mobile robotics algorithms in the robot operating system 2," 7 2023.

[34] P. Gyanani, M. Agarwal, R. Osari, and G. Chandore, "Autonomous mobile vehicle using ros2 and 2d-lidar and slam navigation," 5 2024.

[35] A. . Rafael and T. Rosaleny, “Encoders Ópticos,” 2021.

[36] E. S. M. Borja, *ROBOT MÓVIL DE ALTO TORQUE: DESARROLLO ELECTRÓNICO*. PhD thesis, UNIVERSIDAD TÉCNICA DEL NORTE, 2 2025.

[37] E. S. M. Borja, “Robot móvil de alto torque: desarrollo electrónico,” 2025. [Online]. Disponible en: https://repositorio.utn.edu.ec/handle/123456789/17015.

[38] J. A. G. Cabezas, “Robot móvil de alto torque: desarrollo mecánico,” 2025. [Online]. Disponible en: https://repositorio.utn.edu.ec/handle/123456789/16984.

[39] P. S. H. Rojas, “Robot móvil de alto torque: visión artificial,” 2025. [Online]. Disponible en: https://repositorio.utn.edu.ec/handle/123456789/17002.

Anexos

Anexo A

Códigos y scripts disponibles en: https://github.com/malexandervillalba27-web/Tesis-Odometria