



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA**

CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN

**IMPLEMENTACIÓN DE UN SISTEMA DE NAVEGACIÓN AUTÓNOMA
PARA UN ROBOT MÓVIL DE TRACCIÓN DIFERENCIAL**

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

AUTOR: ANTHONY FERNANDO BRAVO CARRIÓN

TUTOR: ING. EDUARDO PINOS VELEZ, PhD

COTUTOR: DR. ING. JULIO CESAR MONTESDEOCA CONTRERAS

Cuenca – Ecuador

2023


**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE
TITULACIÓN**

Yo, Anthony Fernando Bravo Carrión con documento de identificación N° 1104894454; manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Cuenca, 28 de julio del 2023

Atentamente,



Anthony Fernando Bravo Carrión

1104894454


**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO
DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Yo, Anthony Fernando Bravo Carrión con documento de identificación N° 1104894454, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del Proyecto Técnico: “Implementación De Un Sistema De Navegación Autónoma Para Un Robot Móvil De Tracción Diferencial”, el cual ha sido desarrollado para optar por el título de: Ingeniero en Electrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 28 de julio del 2023

Atentamente,



Anthony Fernando Bravo Carrión

1104894454

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Eduardo Pinos Velez con documento de identificación N° 0102942190, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: IMPLEMENTACIÓN DE UN SISTEMA DE NAVEGACIÓN AUTÓNOMA PARA UN ROBOT MÓVIL DE TRACCIÓN DIFERENCIAL, realizado por Anthony Fernando Bravo Carrión con documento de identificación N° 1104894454, obteniendo como resultado final el trabajo de titulación bajo la opción proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 28 de Julio del 2023

Atentamente,

Ing. Eduardo Pinos Velez, PhD

0102942190

AGRADECIMIENTOS

Agradezco principalmente a Dios por estar presente en mi vida y por darme la
fortaleza para continuar adelante.

A mis padres, por su apoyo, amor y comprensión que me han sabido brindar
día tras día, por creer en mí y ser esa voz de aliento que solo un padre puede
dar.

A mi hermana, por contar con ella en todo momento.

A mi novia, por apoyarme incondicionalmente y darme su comprensión.

A mis asesores de tesis, Julio Cesar Montesdeoca Contreras y Eduardo Pinos
Vélez, por el tiempo dedicado para la realización de esta tesis. Muchas gracias
por su amabilidad y confianza.

DEDICATORIAS

DEDICATORIA

A mi familia en especial a Fernando y Mirian.

Con mucho amor, respeto y admiración.

Por todo el apoyo brindado, el esfuerzo y la dedicación en ser los mejores padres y un pilar fundamental para mi desarrollo.

Índice general

Agradecimientos	I
Dedicatorias	II
Índice General	III
Índice de figuras	V
Índice de tablas	VI
Resumen	VII
Abstract	VIII
Antecedentes	1
Justificación	2
Objetivos	3
Introducción	4
1. MODELADO MATEMÁTICO	6
1.1. MODELO CINEMÁTICO	6
1.2. CONTROL BASADO EN CINEMÁTICA INVERSA	8
1.3. CONTROL DE BAJO NIVEL	8
1.4. TAREAS DE CONTROL	10
1.4.1. Estrategias de seguimiento de trayectorias	10
1.4.2. Estrategias de seguimiento de caminos:	12

<i>ÍNDICE GENERAL</i>	IV
2. Mapeo y navegación autónoma	13
2.1. Mapeo	14
2.1.1. TRANSFORMADAS	14
2.1.2. Localización y mapeo simultáneos (SLAM):	15
2.1.3. Configuración RTAB-Map	18
2.2. Navegación	20
2.2.1. Planificación de Ruta	20
2.2.2. Seguimiento de Ruta	21
2.2.3. Evitación de Obstáculos	21
2.3. Implementación de la electrónica para el sistema de navegación autónomo	22
3. Experimentos y Resultados	24
4. Conclusiones y Trabajos Futuros	31
Glosario	32
Referencias	35

Índice de figuras

1.1. Modelo de Robot de Tracción Diferencial	7
1.2. Placa para el control de bajo nivel con Esp32	10
2.1. Transformadas Move base	14
2.2. marcos estáticos	15
2.3. Diagrama de bloques del sistema de navegación Autónomo . . .	22
3.1. mapa de transformadas	25
3.2. mapa de transformadas con la integración de una cámara	26
3.3. Levantamiento del mapa en 2D del pasillo de la universidad . . .	26
3.4. Punto de partida	27
3.5. Recorrido del robot hasta el final del pasillo	27
3.6. Mapeo en 2D del pasillo aplicando detección de obstáculos	28
3.7. Capa local costmap	28
3.8. Capa Global costmap	29
3.9. capas: global y local costmap	29
3.10. Navegación usando la herramienta 2D nav Goal	30

Índice de tablas

Resumen

El presente trabajo de tesis aborda la implementación de un sistema de navegación autónomo para robots 4wd. Tema que en la actualidad constituye un área de investigación activa en el desarrollo de nuevos robots, estos estudios permiten estar en constante evolución y obtener notables avances en temas como la fusión de sensores, planificación de rutas y la detección de obstáculos.

Este sistema permitirá que el robot se mueva de forma autónoma en diferentes entornos y situaciones, lo que le dará la capacidad de identificar y evadir obstáculos presentes en el medio. Para llevar a cabo el proyecto, se ha adaptado el hardware del robot 4wd con un procesador Jetson Nano, un microcontrolador ESP32, un sensor YDLIDAR y una cámara RealSense D435.

Estos dispositivos conforman nuestro sistema de navegación autónoma integrados en ROS. Al montar el sistema de navegación en el robot, nos permitirá mapear el área de navegación y recopilar información para trazar la mejor ruta, mejorando así la seguridad ya que el robot puede detectar obstáculos y evitar colisiones, lo que se traduce para el robot en ser más versátil y útil para diversas aplicaciones como la exploración de entornos naturales o la inspección de rutas.

Palabras clave: Robot Móvil; Navegación Autónoma; ROS; SLAM

Abstract

This thesis work deals with the implementation of an autonomous navigation system for 4wd robots. autonomous navigation system for 4wd robots. This topic is currently an active area of research in the development of new robots. an area of active research in the development of new robots, these studies allow to be constantly These studies allow to be in constant evolution and to obtain notable advances in topics such as sensor fusion such as sensor fusion, route planning and obstacle detection. This system will allow the robot to move autonomously in different environments and situations, which will enable it to different environments and situations, which will give it the ability to identify and evade obstacles in the and evade obstacles present in the environment. In order to carry out the project, the the 4wd robot hardware has been adapted with a Jetson Nano processor, an ESP32 microcontroller, an ESP32 ESP32 microcontroller, a YDLIDAR sensor and a RealSense D435 camera. These devices make up our autonomous navigation system, integrated in ROS. integrated into ROS. When the navigation system is mounted on the robot, it will allow us to mapping the navigation area and gathering information to plot the best route, thus improving safety route, thus improving safety as the robot can detect obstacles and avoid collisions, which is and avoid collisions, which translates to the robot being more versatile and useful for a variety of applications applications such as the exploration of natural environments or the inspection of routes. route inspection.

Keywords: Mobile Robot; SLAM, Autonomus Navigation

Antecedentes

Dentro del campo de la robótica móvil uno de los mayores retos es la navegación autónoma, en donde la barrera a vencer es la localización y posicionamiento de robots móviles. El campo de la navegación autónoma de robots brinda un área de amplia investigación en la robótica actual. Dentro de la Carrera de Electrónica Y Automatización se han desarrollado varios robots móviles, los cuales aún no tienen integrada la función de navegación autónoma. Por ello, se propone implementar un sistema para la navegación autónoma de un robot móvil mediante el levantamiento de mapas del entorno de navegación.

Justificación

La complejidad de la navegación de un robot móvil depende principalmente del control. Partiendo del interés de contribuir con un sistema de navegación simple y eficiente; se propone la implementación de un algoritmo capaz de hacer que una arquitectura móvil cumpla con el propósito de recorrer la ruta más corta libre de colisiones, desde un inicio hasta una meta de esta manera se consigue mejorar la eficiencia, seguridad y capacidad de investigación en el campo de la robótica. Lo cual puede resultar en costos más bajos y una mayor productividad. Así mismo, dentro de la Carrera de Electrónica y Automatización existen cátedras relacionadas a la robótica móvil dentro de las cuales se han desarrollado trabajos relativos al tema. Como estudiantes de la carrera de consideramos que contamos con los conocimientos y capacidades para desarrollar e implementar este sistema autónomo de navegación, lo que impulsara el desarrollo de nuevas aplicaciones.

Objetivos

Objetivo General

- Implementar un sistema de navegación autónoma para robots móviles de tracción diferencial..

Objetivos específicos:

- Adecuar tecnológicamente un robot móvil de tracción diferencial con el fin de ejecutar los algoritmos de control de movimiento.
- Implementar un algoritmo para el levantamiento del escenario de navegación usando un sistema basado en LIDAR y en una cámara RGB-D.
- Evaluar el desempeño del sistema de navegación autónomo implementado.

Introducción

Uno de los principales retos al hablar de robótica móvil es la navegación autónoma ya sea en escenarios externos como internos. Consecuencia a esto el área de estudio es muy amplio, con diferentes perspectivas a tratar. Para el desarrollo de esta tesis se propone un sistema de navegación autónomo basado en un sensor YDlidar y una cámara Real Sense que serán ensamblados o montados en un robot de tracción diferencial 4wd. De igual forma a través de los encoders de cada motor se podrá calcular la odometría del robot para hacer un sistema mas robusto. El desarrollo del algoritmo esta basado en el método de mapeo RTAB-map (Real-Time Appearance-Based Mapping). Esto nos permitirá reconstruir mediante la información recopilada por los sensores el área donde navegará el robot. Los datos recopilados serán la odometría, datos del sensor YDlidar, y datos de la cámara RealSense d435.

La tecnología utilizada en la técnica RTAB-map es conocida como SLAM (Simultaneous localization and mapping), tecnología que permite implementar los datos recopilados para generar controles de navegación cada vez mas autónomos en ambientes dinámicos. Esta técnica se basa en adquirir la representación del entorno, mismo que servirá para realizar la localización del robot móvil.

El modelo inicial del robot diferencial, tanto en su estructura ,hardware como en software sobre el cual se ah trabajado, en primera instancia se hizo actualizaciones en su estructura como renovaciones en su hardware y de esta manera poder acondicionar el software. La estructura es actualizada para un nuevo prototipo en este caso se parte desde un primer material que es madera, y se le da un aspecto diferente. El software es actualizado y cambiamos la raspberry pi 4b por una Jetson Nano, para el control en bajo nivel se utilizaba

una tarjeta arduino mega, se mejora con una tarjeta ESP32, reduciendo de esta manera el espacio utilizado y mejorando el rendimiento, ya que tiene mejores prestaciones que el arduino mega. Ya actualizado el hardware, se trabajo en la adecuación del código para el control de bajo nivel. Con el acondicionamiento del robot se dio paso para trabajar en el sistema de navegación autónomo.

Capítulo 1

MODELADO MATEMÁTICO

En el campo de la robótica móvil hay diferentes tipos de robots, el robot de tracción diferencial es uno de ellos. Este robot consta de dos ruedas que al ser controladas de forma independiente le dan movimiento al robot en diferentes direcciones. En el caso de un robot de 4 ruedas independientemente controladas que tiene sus ruedas en una configuración de simetría, las ruedas delanteras y traseras se agrupan en cada lado y se controlan como una sola unidad, lo que simplifica el control. Esto reduce la complejidad del sistema de control y permite el uso de técnicas de control de robots diferenciales 2WD bien establecidas y ampliamente utilizadas, lo que facilita el diseño e implementación del sistema de control. Además, se puede utilizar la geometría de un vehículo tipo diferencial para obtener el modelo cinemático de un robot diferencial.

1.1. MODELO CINEMÁTICO

La estructura del robot móvil de tracción diferencial utilizado para crear el modelo cinematográfico extendido propuesto en [1] se muestra en la Figura 1.1. De tal manera que, (x^w, y^w) indican las coordenadas cartesianas globales en dos dimensiones; (x, y) indica el punto de interés del robot, que se encuentra entre las ruedas del robot; y ϕ indica la orientación del robot, que se mide positiva en sentido contrario a las manecillas del reloj respecto a (x^w) . En consecuencia, $q = [x, y, \phi]^T$ proporciona las coordenadas generalizadas del

robot. La velocidad de rotación del robot es ω , mientras que la velocidad lineal es u y la velocidad lateral es v , respectivamente.

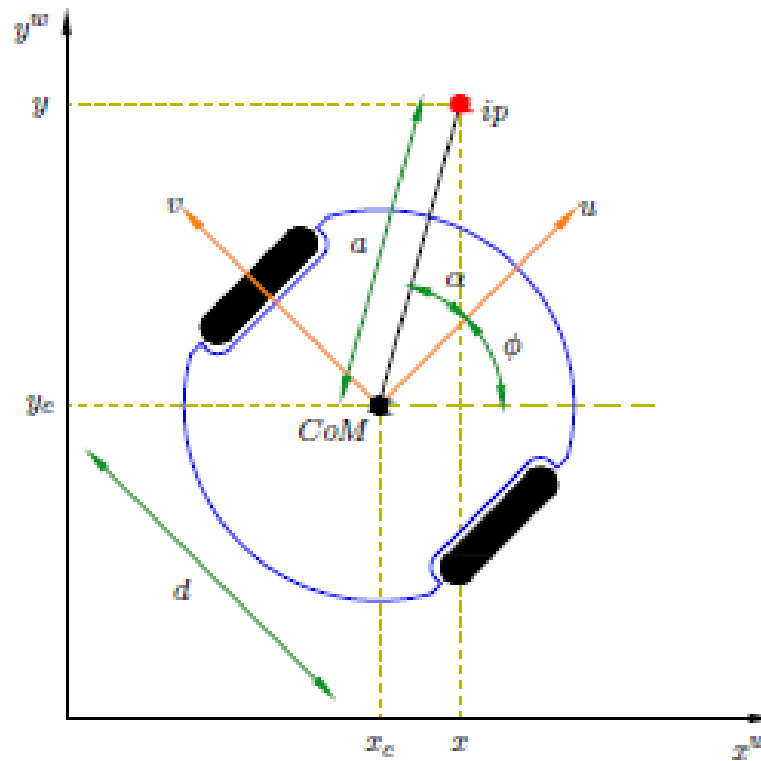


Figura 1.1: Modelo de Robot de Tracción Diferencial

El modelo cinemático extendido del robot de tracción diferencial se muestra en la ecuación 1.1, en donde el modelo presentado se puede representar de forma matricial mediante $\dot{q} = A(q)u$. El modelo propuesto al ser invertible en todo momento, se puede utilizar la cinemática inversa como estrategia de control. Lo que da como resultado el siguiente modelo matricial $u = A^{-1}(q)\dot{q}$. En donde \dot{q} es el vector de estados generalizados, u es el vector de entradas de control, $A^{-1}(q)$ representa la matriz de cinemática inversa.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ \omega \end{bmatrix} \quad (1.1)$$

1.2. CONTROL BASADO EN CINEMÁTICA INVERSA

A partir de lo ya expuesto, se implementa un controlador basado en el modelo cinemático extendido representado en 1.2, su representación compacta esta dada por $u_c = A^{-1}(q) n$. En donde, u_c representa al vector de entradas de control, n es el vector que representa la ley de control.

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} \quad (1.2)$$

1.3. CONTROL DE BAJO NIVEL

El control de bajo nivel hace referencia a controlar de forma directa los actuadores del sistema, en este caso los motores. El objetivo principal es garantizar un movimiento preciso y estable del robot. En un robot de tracción diferencial se pueden controlar de forma independiente las ruedas, lo que permite una variedad de movimientos y comportamientos. Finalmente, los sensores se utilizan para medir variables relevantes como la velocidad de cada rueda, además de poder estimar la posición y la dirección del robot. Estos sensores proporcionan información en tiempo real al controlador, lo que facilita que el controlador ajuste y calibre continuamente los movimientos del robot.

El control en bajo nivel se basa en varios conceptos fundamentales, como la retroalimentación, la modelización del sistema y el diseño de controladores. La retroalimentación es esencial para conocer el estado actual del sistema y compararlo con el estado deseado. Utilizando esta información, se pueden tomar decisiones para ajustar los actuadores y corregir cualquier error entre el estado actual y el estado deseado.

El diseño del controlador involucra el desarrollo de algoritmos y estrategias que determinan cómo ajustar los actuadores para lograr el comportamiento deseado del robot. Los controladores pueden ser de varios

tipos, como controladores PID (proporcional-integral-derivativo), controladores de estado, controladores adaptativos y otros. Cada tipo de controlador tiene sus ventajas y desafíos, y elegir el controlador adecuado depende de las especificaciones y los requisitos de su sistema.

En nuestro robot se propuso un controlador PID para cada motor, este fue programado en una tarjeta ESP32, este se comunica con ROS a través del puerto serial y realiza la recepción de las acciones de control. Los sensores usados para la lectura de velocidad son de tipo efecto hall que vienen incorporados en cada motor. Cuya lectura se hace a través de una interrupción externa de la tarjeta, para esto se propone un tiempo de muestreo de $t_s = 100ms$, debido a la baja resolución del sensor de efecto hall usado.

El calculo de la velocidad angular esta dado por la siguiente ecuación:

$$w = 2\pi \frac{\#pulsos}{t_s Resol} [rad/s] \quad (1.3)$$

■ Conversión de velocidades

Las acciones de control al tener magnitudes como velocidad lineal m/s y velocidad angular rad/s ; es preciso realizar la conversión para que el controlador de bajo nivel se ajusten a las magnitudes de velocidad angular . La velocidad lineal y la velocidad angular están descritas en la siguiente matriz:

$$\begin{bmatrix} u_c \\ w_c \end{bmatrix} = \begin{bmatrix} r/2 & r/2 \\ r/d & -r/d \end{bmatrix} \begin{bmatrix} w_r \\ w_l \end{bmatrix} \quad (1.4)$$

Aplicamos la inversa e la ecuación 1.4, en el resultado obtenemos las velocidades angulares para cada una de las ruedas en donde la magnitud esta expresada en rad/s

$$\begin{bmatrix} w_r \\ w_l \end{bmatrix} = \begin{bmatrix} 1/r & d/2r \\ 1/r & -d/2r \end{bmatrix} \begin{bmatrix} u_c \\ w_c \end{bmatrix} \quad (1.5)$$

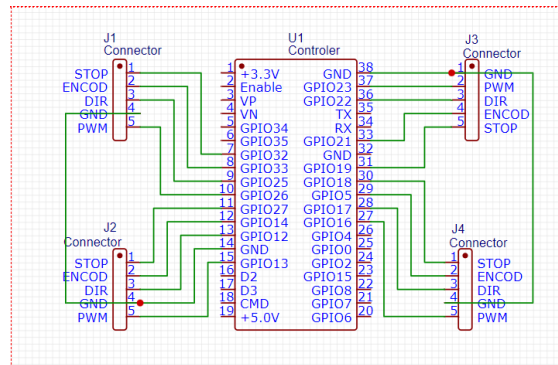


Figura 1.2: Placa para el control de bajo nivel con Esp32

1.4. TAREAS DE CONTROL

Dentro del campo de la robótica móvil, el seguimiento de trayectorias y de caminos, son enfoques esenciales para guiar un robot hacia su destino. Cada enfoque ocupa diferentes métodos o estrategias para cumplir con su objetivo. El seguimiento de trayectorias hace referencia a la capacidad que tiene el robot para seguir una trayectoria predefinida dentro del medio de navegación. En cuanto al seguimiento de caminos es la capacidad de seguir un camino predefinido en el espacio de navegación. Al desarrollar un sistema de control para robots, lo más común es utilizar modelos matemáticos y simulaciones con el fin de evaluar el desempeño del sistema. Por ejemplo, en un estudio se creó un sistema de control para un robot móvil que utiliza retroalimentación visual, lo que le permite rastrear objetivos mediante un sistema de visión monocular y un diagrama de control basado en la modelación cinemática del sistema completo. [2]. Otro estudio propuso la creación de un sistema lógico difuso para controlar la locomoción bípeda de un humanoide. Este sistema utilizaría las posiciones de los ángulos de las piernas como entradas y la posición cercana a cero del ángulo del torso como salida. [3]. Entre las estrategias de seguimiento más destacadas tenemos:

1.4.1. Estrategias de seguimiento de trayectorias

- Control de realimentación** El control de retroalimentación es una estrategia la cual hace uso de sensores para medir la posición y velocidad del robot, mediante esta recepción de datos se puede ajustar estos

parámetros en tiempo real para seguir una trayectoria deseada . Esta estrategia es utilizada en la fisioterapia asistida con robots, gracias a su precisión y adaptabilidad en las fuerzas de interacción humana-robot, de igual manera en la modulación de amplitud y frecuencia en las trayectorias de referencia [4].

- **Modelado dinámico:** Es una técnica que se apoya en modelos matemáticos para predecir el comportamiento de robots y ajustar su trayectoria en función de ello. Esta valiosa técnica busca mejorar la precisión del movimiento del robot y asegurar su desplazamiento de forma segura y eficiente. Al desarrollar un modelo matemático para un robot móvil, es esencial considerar aspectos cinemáticos, dinámicos y eléctricos del motor. Para validar el funcionamiento adecuado del modelo, también puedo llevar a cabo simulaciones utilizando Simulink [5].
- **Algoritmos de optimización:** Los algoritmos de optimización buscan encontrar la ruta más eficiente para el robot considerando factores como distancia, velocidad y aceleración. Su aplicación puede abarcar diversas áreas, como optimizar el espacio de trabajo del robot [6], lograr un control lineal óptimo para estabilizar trayectorias circulares en un robot móvil [7], o calcular las dimensiones más eficientes para un robot paralelo tipo delta con el menor consumo de energía [8]. Para llevar a cabo estos algoritmos, se puede utilizar herramientas como algoritmos genéticos [6], programación semi-infinita o algoritmos estocásticos de aproximaciones externas [9]. Además, realizaré análisis de la cinemática directa e inversa del robot para lograr una comprensión más profunda de su funcionamiento y comportamiento [6].
- **Técnica de adición de integrador:** Como estudiante de electrónica y automatización, mi tesis se enfoca en la técnica de adición de integrador, la cual es una estrategia empleada en el control de sistemas para mejorar la precisión en el seguimiento de trayectorias. Esta técnica consiste en incluir un término integrador en la ley de control, lo que posibilita que el sistema responda de manera más rápida y precisa

ante modificaciones en la trayectoria. En mi investigación, me enfoco especialmente en su aplicación en el control de robots móviles con ruedas. Los enfoques clásicos de modelamiento y control aplicados a robots móviles de locomoción diferencial utilizan ecuaciones matemáticas que aproximadamente representan la dinámica del sistema, y funcionan adecuadamente dentro de un rango específico de trabajo cuando el sistema es lineal. Sin embargo, estos métodos pueden presentar limitaciones en términos de precisión cuando se enfrentan a diversas variaciones dinámicas en el tiempo o ante la presencia de perturbaciones. Por ello, la incorporación de un término integrador en la ley de control se muestra como una solución efectiva para lograr una respuesta más veloz y precisa ante los cambios en la trayectoria del robot [10].

1.4.2. Estrategias de seguimiento de caminos:

- **Algoritmo de Bug:** Es una categoría de algoritmos diseñados para ayudar a los robots a planificar sus movimientos en entornos con obstáculos. Este algoritmo se basa en seguir el contorno de los obstáculos hasta encontrar una posición más cercana al objetivo deseado. Dentro de las principales versiones del algoritmo de Bug, encontramos Bug0, Bug1 y Bug2, cada una con sus propias características distintivas. La distinción principal entre los algoritmos de la familia Bug radica en su enfoque para determinar el lado por el cual rodear el obstáculo. [11].
- **Campos de potencial** Esta técnica permite que el robot se desplace dentro de un entorno virtual de fuerzas combinadas. Este campo virtual está compuesto por zonas atractivas que dirigen al robot hacia su destino, y zonas repulsivas que evitan colisiones con obstáculos en su ruta [12].

El objetivo de este algoritmo es guiar al robot desde su posición actual hasta una posición predefinida. El robot es dirigido mediante el uso del gradiente de este campo de fuerza virtual, lo cual presenta la ventaja de generar simultáneamente la trayectoria que debe seguir y ejecutar el movimiento del robot en tiempo real [12].

Capítulo 2

Mapeo y navegación autónoma

En los últimos años, ha habido un notable incremento en la investigación en el campo de la robótica, motivado por los avances en inteligencia artificial y el crecimiento del poder de procesamiento de los sistemas informáticos, a tal punto de tener avances muy significativos en cuanto a métodos para crear mapas y permitir la navegación autónoma en robots diferenciales. Estos robots son ampliamente utilizados en diversas aplicaciones robóticas gracias a su capacidad para desplazarse de forma eficiente y adaptable en distintos entornos.

En el ámbito de la robótica móvil, la cartografía y la navegación son dos elementos fundamentales. La cartografía consiste en desarrollar un modelo del entorno circundante al robot, mientras que la navegación implica la planificación y ejecución de movimientos necesarios para que el robot alcance una ubicación o destino específico dentro de dicho entorno. Estas herramientas son esenciales para que el robot pueda comprender su entorno y moverse con éxito en diferentes situaciones. No obstante, lograr una cartografía precisa representa un desafío complejo debido a la incertidumbre presente en los datos sensoriales y en la propia naturaleza del entorno.

En este capítulo, abordo aspectos esenciales relacionados con el mapeo y la navegación autónoma en robots diferenciales como parte de mi tesis en electrónica y automatización. Comienzo examinando el estado actual de este campo, analizando la técnica de mapeo SLAM y los algoritmos más comúnmente utilizados para construir mapas y planificar rutas en entornos desconocidos. Luego, selecciono el algoritmo que mejor se adapta a

nuestras necesidades. Describo detalladamente la implementación del sistema propuesto, que combina varios métodos para lograr un rendimiento óptimo en términos de precisión cartográfica y eficiencia de navegación.

Algunas de los algoritmos revisados incluyen Occupancy Grid Mapping, Visual SLAM y RTAB-Map. Cada uno de ellos ha sido cuidadosamente evaluada para determinar cuál será la más adecuada para nuestros objetivos de investigación.

2.1. Mapeo

2.1.1. TRANSFORMADAS

En ROS, los marcos de referencia y las transformaciones de coordenadas son conceptos esenciales que se utilizan para representar las relaciones espaciales entre diferentes partes de un sistema de robot. Un marco de referencia es un sistema de coordenadas utilizado para describir la posición (es decir, la ubicación y orientación) de un objeto en el espacio. En ROS, los marcos de referencia se representan mediante un "frame ID" que es una cadena que identifica el marco de referencia. Las transformaciones de coordenadas se utilizan para convertir coordenadas de un marco de referencia a otro. El árbol de transformaciones consiste en dos marcos dinámicos [2.1](#) y seis marcos estáticos [2.2](#).

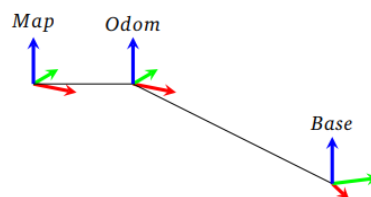


Figura 2.1: Transformadas Move base

Es muy importante obtener la transformación correcta Map-Odom-Base. Para resolver la primera transformación, se utiliza la localización y mapeo simultáneos (SLAM), La segunda se resuelve mediante el filtro de Kalman extendido (EKF), se aconseja tener esa transformación adicional a .odom.^{en} lugar

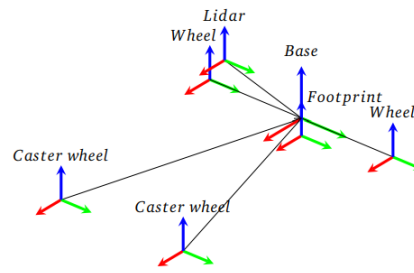


Figura 2.2: marcos estáticos

de ir directamente de Map Base. La razón es que los algoritmos de optimización de posición del lidar son relativamente lentos y no son suaves, mientras que el proceso odom-base es rápido y suave, pero puede desviarse con el tiempo. Así que para obtener la transformación correcta durante todo el recorrido, se calcula la transformación Map-Odom utilizando la información de Odom-Base y luego se corrige con su transformación. El resultado es una transformación suave que no se desvía con el tiempo.

2.1.2. Localización y mapeo simultáneos (SLAM):

Localización y mapeo simultáneos (SLAM, por sus siglas en inglés). Esta técnica es ampliamente utilizada en robótica para permitir que un robot pueda moverse o llevar a cabo tareas de manera autónoma en un ambiente desconocido. El objetivo principal es dotar al robot de la capacidad de conocer su posición y orientación, además de crear una representación detallada del entorno en el que se encuentra [13]. El proceso de SLAM se divide en dos tareas esenciales: la localización, que se enfoca en la habilidad del robot para estimar su posición y orientación en el entorno, y el mapeo, que se concentra en la creación de un mapa preciso del entorno que lo rodea [14].

Occupancy Grid Mapping:

La Mapeo de Rejilla de Ocupación es una técnica utilizada en robótica para crear mapas de un entorno utilizando datos de sensores. La técnica consiste en dividir el entorno en una rejilla de celdas y asignar a cada celda un valor de probabilidad que representa la posibilidad de que esa celda esté ocupada por un obstáculo [15]. Los valores de probabilidad se actualizan a

medida que el robot se mueve por el entorno y recopila datos de sensores, lo que permite refinar el mapa con el tiempo [14] [15].

La Mapeo de Rejilla de Ocupación se utiliza a menudo en conjunto con algoritmos de Localización y Mapeo Simultáneos (SLAM), que permiten a un robot mapear simultáneamente un entorno y determinar su propia ubicación dentro de ese entorno [16]. Los algoritmos SLAM utilizan datos de sensores para estimar la posición y orientación del robot, y luego actualizan el mapa en consecuencia. Esto permite que el robot navegue por el entorno evitando obstáculos y siguiendo una ruta planificada [13] [15].

Visual SLAM:

Usando cámaras y algoritmos de visión por computadora, la técnica visual SLAM crea mapas y calcula la posición y orientación del robot. Este método se basa en identificar y seguir pistas visuales en las imágenes que toman las cámaras del robot, como puntos de interés o patrones visuales. Visual SLAM es particularmente útil en entornos con características visuales distintivas, como interiores de edificios o áreas urbanas.

RTAB-Map (Real-Time Appearance-Based Mapping):

RTAB-Map (Mapeo basado en apariencia en tiempo real) es un enfoque de SLAM (Localización y Mapeo Simultáneo) basado en grafos que utiliza datos recolectados de sensores de visión para localizar el robot y mapear el entorno. Es un enfoque de SLAM RGB-D de código abierto basado en un detector global de cierre de bucles Bayesianos. El detector de cierre de bucles utiliza un enfoque de bolsa de palabras para determinar la probabilidad de que una nueva imagen provenga de una ubicación previa o de una ubicación nueva. Cuando se acepta una hipótesis de cierre de bucle, se agrega una nueva restricción al grafo del mapa y luego un optimizador de grafo minimiza los errores en el mapa en tiempo real.

RTAB-Map está optimizado para SLAM a gran escala y a largo plazo mediante el uso de múltiples estrategias que permiten que el cierre de bucles se realice en tiempo real. El cierre de bucles ocurre lo suficientemente rápido como

para que el resultado pueda obtenerse antes de que se adquirieran las siguientes imágenes de la cámara. Se utiliza un enfoque de gestión de memoria para limitar el número de ubicaciones utilizadas para la detección de cierre de bucles y la optimización del grafo, de modo que siempre se respeten las restricciones en tiempo real en entornos a gran escala.

RTAB-Map se puede utilizar de forma independiente con un Kinect portátil, una cámara estéreo o un LiDAR 3D para el mapeo 6DoF, o en un robot equipado con un telémetro láser para el mapeo 3DoF. Se utiliza en diversas aplicaciones, como robots móviles autónomos, realidad aumentada y escaneo 3D. RTAB-Map se considera una de las mejores técnicas de mapeo en robots debido a varias razones:

- **Fusión de datos de múlti-sensor:** RTAB-Map puede combinar datos de varios sensores, lo que permite la creación de mapas más detallados y completos. El sistema puede recopilar datos de cámaras RGB-D, sensores láser y odómetros para capturar varios aspectos del entorno y mejorar la calidad del mapa resultante.
- **Reconocimiento visual y localización:** para identificar ubicaciones y características ambientales, RTAB-Map emplea un enfoque de apariencia visual. Al combinar datos visuales con datos de sensores, esto le permite crear mapas que son más sólidos y precisos. Además, el robot puede ubicarse con mayor precisión en un mapa gracias a su capacidad de reconocimiento visual.
- **Adaptabilidad al cambio:** RTAB-Map puede adaptarse a los cambios en el entorno. Esto indica que puede adaptarse a cambios como la aparición o desaparición de objetos, ajustes en la iluminación o ajustes en la estructura del entorno. Es adecuado para aplicaciones en entornos dinámicos y no estructurados debido a su flexibilidad en tales circunstancias.
- **Exploración autónoma eficiente:** el robot RTAB-Map está equipado con algoritmos de exploración autónomos que le permiten elegir dónde y cómo explorar eficientemente su entorno. Cuando se realizan tareas como

el mapeo completo de una región inexplorada o la búsqueda y el rescate en entornos peligrosos, esto es especialmente útil.

- Código abierto y comunidad activa RTAB-Map es un proyecto de código abierto, que hace que su código fuente sea accesible a la comunidad de desarrolladores para su uso y mejora. Como resultado, se ha desarrollado una comunidad vibrante, contribuyendo activamente con nuevas funciones, mejoras y correcciones de errores para garantizar un desarrollo continuo y una solución sólida.
- Cuando se combinan, estas características hacen de RTAB-Map una técnica de mapeo de robots muy fuerte y adaptable, capaz de abordar una variedad de aplicaciones y entornos. Los expertos en mapeo robótico lo consideran una opción superior debido a su capacidad para combinar datos de varios sensores, énfasis en el reconocimiento visual y adaptabilidad al cambio.

Por las ventajas que ofrece este algoritmo de mapeo y la integración de los diferentes sensores para la construcción del mapa, se a escogido trabajar con el mismo.

2.1.3. Configuración RTAB-Map

Una de las características mas destacadas de RTAB-Map es su adaptabilidad a varias configuraciones y requisitos de aplicaciones particulares. En esta introducción se analizarán las configuraciones principales de RTAB-Map, junto con la forma en que pueden afectar la efectividad y los resultados del mapeo robótico.

- Para usar RTAB-Map de manera efectiva, la configuración del sensor es esencial. Las especificaciones de los sensores utilizados, como cámaras RGB-D, sensores láser y odómetros, deben definirse con precisión. Esto implica ajustar la resolución particular de cada sensor, la frecuencia de muestreo, los rangos de detección y otras configuraciones. La configuración adecuada de un sensor garantiza una recopilación de datos precisa y confiable, lo que mejora el rendimiento del mapeo.

- Configuración del algoritmo: el algoritmo de mapeo se puede ajustar de varias maneras con RTAB-Map. Estos parámetros incluyen la distancia mínima de detección de características, el umbral de coincidencia de características, la resolución del mapa, el umbral de bucle y muchos otros. Para adaptarse a las necesidades de la aplicación y las características del entorno, la configuración del algoritmo debe ajustarse cuidadosamente. La precisión y la durabilidad del mapeo se pueden aumentar con el uso de una configuración ideal.
- Configuración de la memoria: para almacenar y mantener datos sobre el entorno mapeado, RTAB-Map hace uso de la memoria a largo plazo. El tamaño máximo de la memoria, la velocidad a la que se retienen los datos en la memoria y la duración del almacenamiento de datos se incluyen en la configuración de la memoria. La capacidad del sistema para manejar entornos grandes, la efectividad computacional y la persistencia de datos se ven afectados por estos parámetros.
- Opciones de configuración para salida y visualización: Para la salida y presentación de resultados de mapeo, RTAB-Map proporciona una serie de opciones de configuración. Esto incluye la creación de mapas 2D y 3D, la visualización en vivo de la ubicación y orientación del robot, así como la exportación de datos en varios formatos. Para cumplir con los requisitos específicos de análisis, presentación o integración del sistema, la configuración de salida y visualización se puede personalizar.
- Para lograr los mejores resultados en el mapeo robótico, RTAB-Map debe configurarse correctamente. La precisión, la eficacia y el calibre del mapeo se pueden aumentar tomando las decisiones correctas con respecto a la configuración del sensor, los algoritmos, la memoria y la salida. Para utilizar completamente las capacidades de mapeo en tiempo real de RTAB-Map, es fundamental comprender las opciones de configuración disponibles y adaptarlas a los requisitos particulares de la aplicación.

2.2. Navegación

Una vez que se ha creado el mapa, el robot puede emplearlo para desplazarse por los pasillos de la universidad. Para lograr una navegación autónoma, es posible utilizar el algoritmo D* con el propósito de planificar la ruta del robot en tiempo real. A diferencia del algoritmo A*, el algoritmo D* cuenta con la ventaja de adaptarse a los cambios en el entorno y ajustar la trayectoria al instante, lo cual lo hace muy apropiado para entornos dinámicos donde los obstáculos pueden cambiar de posición o aparecer repentinamente [17].

Debido a la limitación no holonómica que presentan los robots diferenciales, la trayectoria obtenida mediante el algoritmo D* se somete a un proceso de post-procesamiento para que posteriormente pueda ser ejecutada por el robot mediante un controlador no lineal [18]. Además del algoritmo D*, existen otros métodos como el algoritmo RRT y el algoritmo VFH+ que se pueden aplicar para lograr la navegación autónoma de un carro diferencial [19]. La elección del algoritmo dependerá de las características específicas del entorno, las restricciones del robot y los objetivos particulares del proyecto. Es crucial evaluar y comparar diferentes algoritmos para determinar cuál de ellos es el más adecuado en cada caso.

La tesis tiene como objetivo implementar una solución que permita la navegación autónoma y la generación de mapas para visualizar el área del levantamiento.

2.2.1. Planificación de Ruta

La planificación de ruta es un aspecto esencial de la navegación autónoma. La tesis propone un algoritmo de planificación de ruta dinámico que puede generar una ruta libre de colisiones para el robot. El algoritmo utiliza una combinación de mapas de costos globales y locales para planificar la ruta. Los mapas de costos globales son regiones alrededor de obstáculos que utiliza el planificador global para calcular rutas para navegar de un punto del mapa a otro sin acercarse demasiado a los obstáculos estáticos registrados durante el mapeo.

Los mapas de costos locales son similares a los globales, pero son más pequeños y se mueven con el robot. El planificador local los utiliza para evitar obstáculos, tanto estáticos como dinámicos, mientras intenta seguir la ruta global calculada por el planificador global [20].

2.2.2. Seguimiento de Ruta

La tesis también propone un algoritmo de seguimiento de ruta que puede seguir con precisión la ruta generada. El algoritmo utiliza un método de seguimiento puro regulado que ajusta las velocidades lineales con un enfoque particular en la curvatura de la ruta [20].

2.2.3. Evitación de Obstáculos

La evitación de obstáculos es otro aspecto crucial de la navegación autónoma. La tesis propone un método de evitación de obstáculos que utiliza una combinación de mapeo y el algoritmo de "Dynamic Window Approach"(DWA). El método puede reconocer obstáculos y generar nuevas trayectorias para alcanzar el destino [20].

Global costmap

El mapa de costos global es un mapa de cuadrícula que representa la vista global del entorno del robot y se creó como una capa encima del mapa base. Se utilizó una configuración en la que las celdas del mapa de costos se interpretaron con solo tres valores: libre, ocupado e desconocido. Los valores van desde valores típicos de píxeles (0-255), donde 255 representa un obstáculo y 0 está libre. Si se utiliza en un modo con más de tres valores, se pueden agregar lugares donde el robot deba evitar o disminuir la velocidad al tener un valor intermedio. Luego, se calculan los costos de atravesar ese lugar y se utilizan en la planificación de rutas y el comportamiento del robot se ajusta para evitar esas áreas. Para evitar colisionar con objetos y paredes, se tuvieron en cuenta las restricciones cinemáticas y la forma del robot en el mapa de costos [20].

Local costmap

El mapa de costos local se utiliza junto con el mapa de costos global y representa una vista más local del entorno del robot. El mapa de costos local se implementa como un complemento en la pila de navegación (Navigation2) y es utilizado por varios otros nodos mencionados aquí. Juntos, crean un campo potencial alrededor de todos los obstáculos en el mapa y son detectados por el LIDAR. Utilicé un cuadrado de 10x10m como dimensión del mapa de costos local, lo cual fue suficiente debido a la baja velocidad del robot, pero si se espera que el robot se desplace más rápido o los obstáculos se acerquen rápidamente, podría ser necesario un mapa de costos más grande. Al igual que en el mapa de costos global, se puede especificar el tamaño del campo potencial alrededor de los obstáculos para mantener una distancia segura. Un problema con el que me encontré fue que idealmente el robot debería poder diferenciar entre obstáculos para mantener una distancia más larga de aquellos que se mueven, como por ejemplo, las personas [20].

2.3. Implementación de la electrónica para el sistema de navegación autónomo

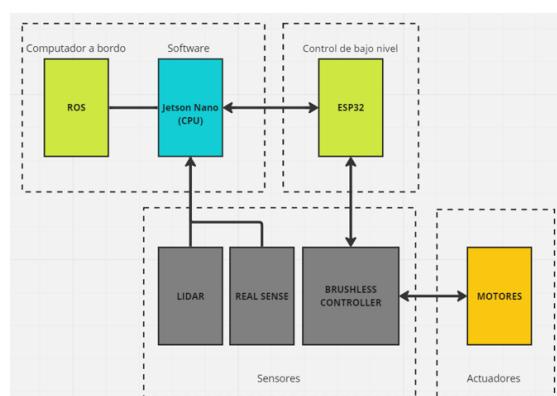


Figura 2.3: Diagrama de bloques del sistema de navegación Autónomo

En el diagrama de bloques mostrado en la fig 2.3 se observa el sistema de navegación autónomo propuesto, el cual está integrado por un sensor láser YDLidar, un procesador Jetson Nano, una cámara RealSense D435 y un

microcontrolador ESP32 para control de bajo nivel.

Una potente plataforma de IA creada específicamente para aplicaciones de inferencia y aprendizaje automático se conoce como Jetson Nano. Sirve como el cerebro del sistema, procesando la entrada de datos y emitiendo juicios basados en algoritmos de control.

La cámara RealSense D435 captura imágenes en 3D mediante tecnología estereoscópica. Permite que el sistema perciba el entorno en tres dimensiones mediante la recopilación de información espacial mediante dos sensores de profundidad.

Con la ayuda de la tecnología de escaneo láser, el sensor YDLidar puede detectar objetos y calcular distancias. Para navegar de forma autónoma y evitar obstáculos, proporciona información sobre la proximidad y ubicación de objetos cercanos.

El control de bajo nivel y la comunicación con los componentes del sistema son manejados por el microcontrolador ESP32. Es un microcontrolador altamente adaptable con bajos requisitos de energía que puede manejar tareas de control en tiempo real y controlar la comunicación con otros dispositivos.

Este sistema de navegación autónomo combina datos de la cámara RealSense D435 y el sensor YDLidar para crear un mapa del área circundante y tomar decisiones basadas en la información. Para elegir el mejor camino, evitar peligros y navegar de forma independiente, Jetson Nano procesa y analiza estos datos.

El microcontrolador ESP32 es el encargado de realizar los algoritmos de control en tiempo real, ajustando los actuadores y motores del sistema para mantener la estabilidad y viajar por la ruta predeterminada. Adicionalmente, se encarga de establecer conexiones con otros sistemas o dispositivos externos para permitir la integración con sistemas de control superiores.

Capítulo 3

Experimentos y Resultados

En este trabajo se ha implementado un sistema de navegación autónoma para un robot diferencial de 4 ruedas, utilizando una cámara RGB-D y un sensor YDLIDAR. Este sistema permite que el robot se mueva en un entorno estructurado, concretamente, por pasillos y habitaciones accesibles desde dichos pasillos. Para implementar el sistema de navegación, se utiliza el algoritmo RTAB-Map, el cual es un algoritmo de mapeo y localización simultánea (SLAM) que crea un mapa 3D o 2D del entorno, además proporciona la ubicación del robot en tiempo real. En este caso, debido a las prestaciones del procesador usado se trabaja con el mapeo en 2D.

El proceso de navegación comienza con la construcción del mapa en 2D. luego el robot se posiciona en el centro del pasillo, con el comando 2D Nav Goal de la interfaz RVIZ podemos enviar la orden al robot de avanzar hasta cierto punto indicado, con una determinada orientación. Luego, avanza hacia el objetivo, y en caso de detectar un obstáculo, busca la forma más rápida de evadirlo hasta lograr llegar a su objetivo.

Los resultados obtenidos en tiempo real fueron los esperados al navegar por el recinto. El sistema de navegación autónoma del robot diferencial logró construir un mapa 2D del entorno y rastrear la ubicación del robot en tiempo real, lo que facilita su movilidad en un entorno desconocido. A continuación se presenta los resultados obtenidos en la experimentación. El primer paso es montar el árbol de transformadas como se muestra en la imagen [3.1](#)

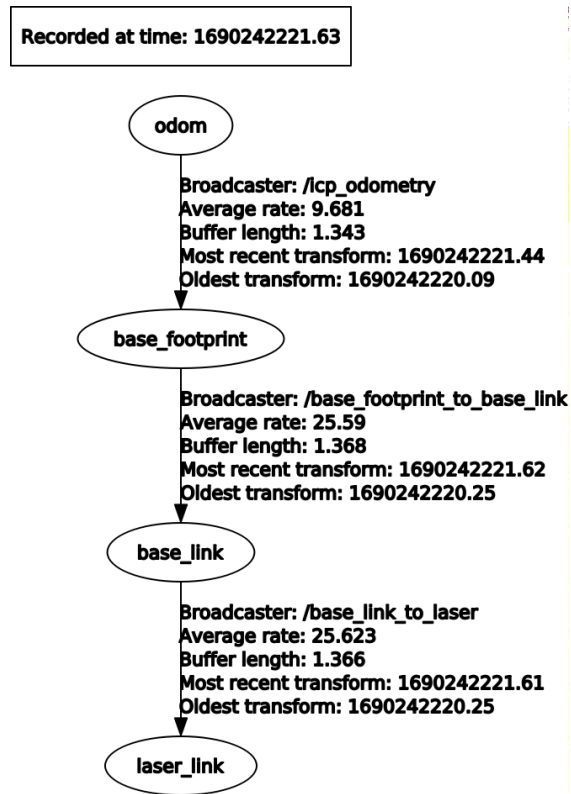


Figura 3.1: mapa de transformadas

si las transformadas no están montadas de forma correcta, la construcción del mapa no se realizara y surgirán errores al momento de correr los programas. cabe mencionar que una forma de conseguir resultados mas robustos en la construcción del mapa es implementar una cámara para la construcción del mapa, el árbol de transformadas se muestra en la imagen 3.2.

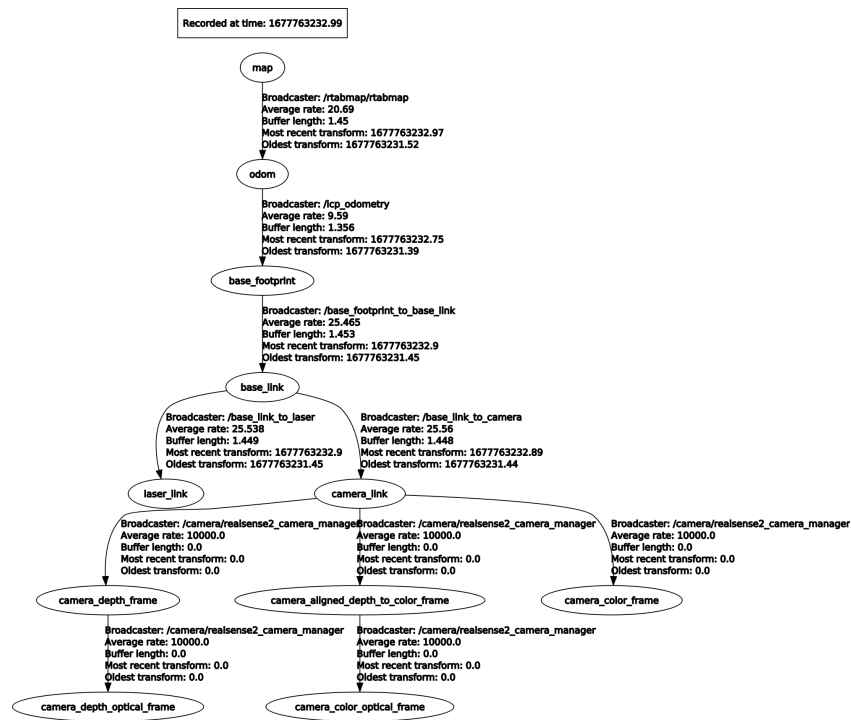


Figura 3.2: mapa de transformadas con la integración de una cámara

Con las transformadas echas procedemos a experimentar con el levantamiento de mapas en 2D.

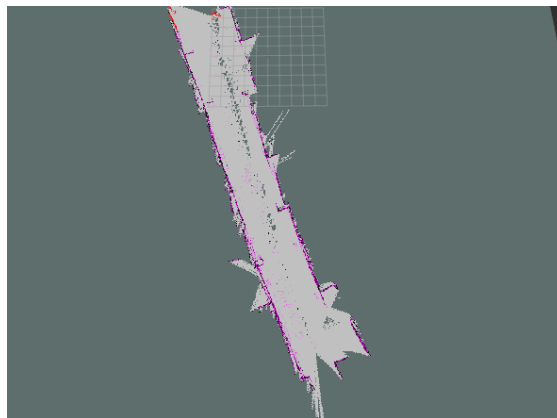


Figura 3.3: Levantamiento del mapa en 2D del pasillo de la universidad

En la 3.3 se puede observar el mapa en 2D construido el sistema de navegación implementado en el robot diferencial 4wd, el mapa pertenece al pasillo de la universidad ubicado en el tercer piso, junto a la oficina de dirección de carrera de electrónica y automatización.

Para llevar a cabo las pruebas de mapeo, se seleccionó como punto de inicio la sección del pasillo donde se encuentran las escaleras que suben desde

la secretaría hacia la mano derecha, tal como se representa en la imagen 3.4. El robot partió desde el centro del pasillo y realizó un giro completo de 180 grados en sentido horario, asegurándose de cubrir todos los puntos relevantes en el mapa 2D

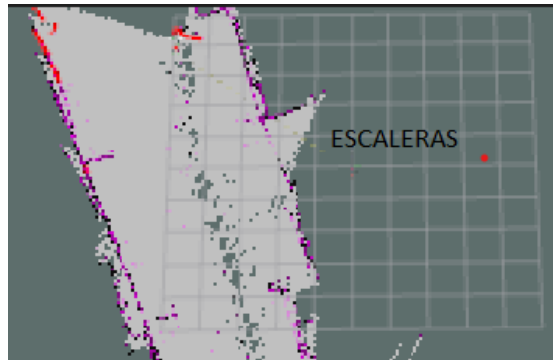


Figura 3.4: Punto de partida

De manera teleoperada, el robot avanza hacia adelante en dirección a las oficinas de audiovisuales que se encuentran al final del pasillo. Los detalles del recorrido del robot se encuentran representados en la figura 3.5, donde se pueden apreciar diversos aspectos capturados en el mapa 2D, como por ejemplo, las ubicaciones de las puertas y la entrada a un pasillo.

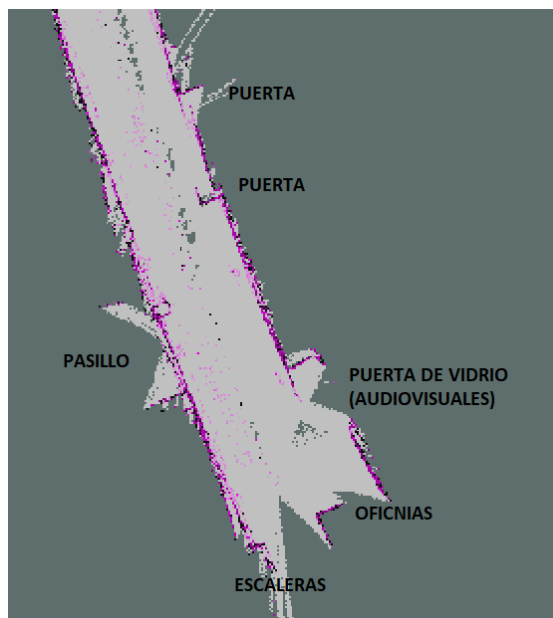


Figura 3.5: Recorrido del robot hasta el final del pasillo

De este modo, hemos confirmado que el mapeo en 2D se está llevando a cabo de manera adecuada, capturando la información ambiental necesaria

para la navegación del robot. A continuación, procedimos a realizar pruebas adicionales para incorporar la detección de obstáculos, lo cual nos dio paso para habilitar la navegación autónoma una vez que estas pruebas fueron completadas

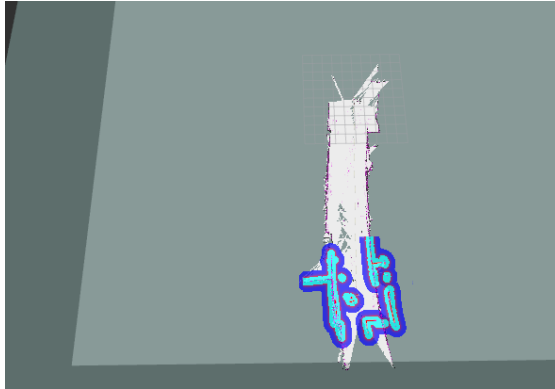


Figura 3.6: Mapeo en 2D del pasillo aplicando detección de obstáculos

En la figura 3.6, podemos apreciar la aplicación del mismo concepto utilizado en el experimento previo. En este nuevo experimento, seguimos una trayectoria mediante teleoperación, desde un punto inicial hasta un punto final. Al implementar la detección de obstáculos, notamos que alrededor de nuestro robot aparecen marcas de color azul, lo que corresponde a la capa de local costmap. Esta capa nos permite identificar los obstáculos que rodean al robot, permitiéndonos establecer un margen seguro para evitar colisiones y garantizar una navegación segura en espacios libres de estas marcas. La figura 3.7 muestra una vista más amplia de la capa de local costmap. Es relevante destacar que la distinción de colores se refiere a las áreas libres, ocupadas y desconocidas.



Figura 3.7: Capa local costmap

Como se explica en el capítulo 3. En esta sección, se menciona la existencia de una capa denominada "global costmap", que, al igual que la capa "local costmap", nos permite obtener información sobre los obstáculos presentes, pero de una manera global, como su nombre lo sugiere. Esta capa se puede observar en la imagen 3.8, las marcas de color negro son los obstáculos que se han ido detectando durante el recorrido.

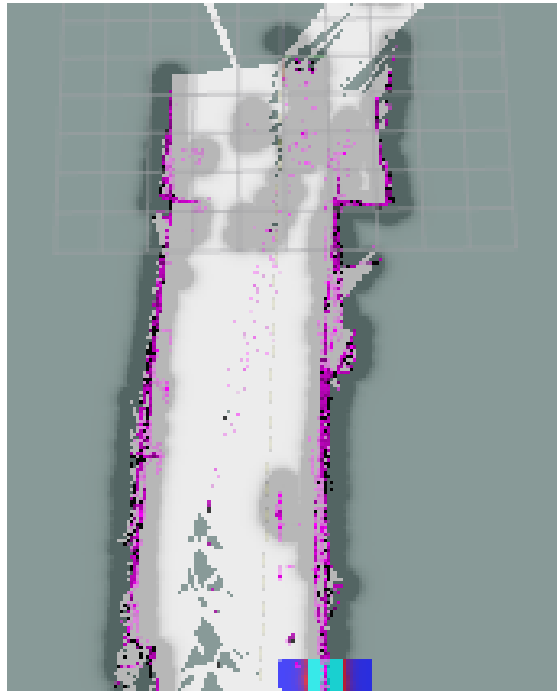


Figura 3.8: Capa Global costmap

Como se muestra en la imagen 3.9, se puede observar simultáneamente ambas capas. Se ha optado por diferenciarlas para facilitar la percepción. La capa local costmap está representada con tonos de azul degradados, mientras que la capa global costmap se muestra con degradados de color negro.

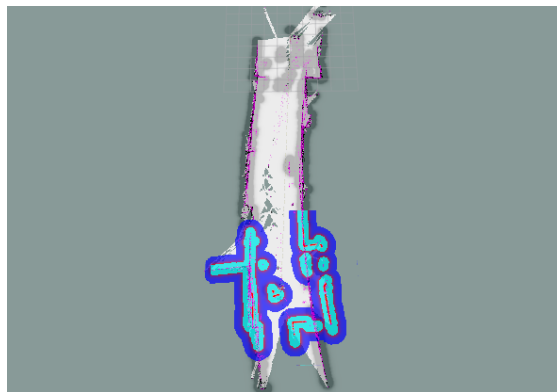


Figura 3.9: capas: global y local costmap

Para las pruebas de navegación se hizo uso de la herramienta 2D Nav Goal en RViz para enviar la posición objetivo al nodo move base en el stack de navegación ROS. Esta herramienta permite establecer la posición deseada a la que el robot debe dirigirse. Se encuentra ubicada en el panel superior de RViz y puede seleccionarse haciendo clic en ella. Una vez seleccionada, puedo indicar la posición objetivo dentro del mapa haciendo clic izquierdo en el mapa con el mouse. La posición objetivo será enviada al nodo move base para que el robot se desplace hacia esa ubicación, como se muestra en la imagen 3.10.

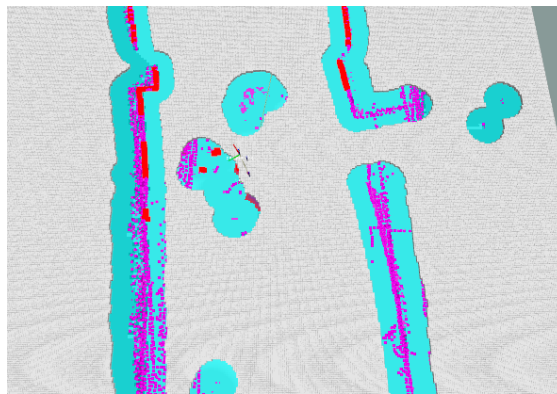


Figura 3.10: Navegación usando la herramienta 2D nav Goal

Es importante tener en cuenta que el robot no se moverá si no está correctamente localizado en el entorno. El sistema de localización debe inicializarse antes de enviar una posición objetivo al robot. Además, el robot debe contar con un mapa del entorno para poder navegar adecuadamente.

Capítulo 4

Conclusiones y Trabajos Futuros

El trabajo realizado ha sido abordado de manera eficiente en cada paso desarrollado, obteniendo así diferentes experiencias en el mismo. Uno de los aspectos más importantes al momento de trabajar con sistemas de navegación para robots, son los marcos de referencia y las transformadas de coordenadas.

Debido a que estos parámetros son los que nos permiten representar la relación entre el robot y el ambiente de navegación. Si las transformadas no están bien definidas, será imposible que el robot pueda garantizar la precisión y la localización como en el mapeo.

En cuanto a la técnica de mapeo Rtabmap para este proyecto, fue muy útil ya que cumple con las prestaciones que se necesita para la navegación en los pasillos de la Universidad Politécnica Salesiana. De igual manera los parámetros del algoritmo de mapeo deben ser configurados en base a las necesidades de mapeo, de esta manera se logra conseguir resultados más robustos y eficientes.

En el futuro, este trabajo podría servir como base para continuar explorando nuevos enfoques de navegación autónoma y proponer algoritmos más avanzados para mejorar aún más la precisión y eficiencia de los sistemas de localización y mapeo en los robots. Además, se podrían investigar aplicaciones adicionales para la optimización de sensores y la integración de sistemas más complejos.

Glosario

PID Control Proporcional Integral Derivativo.

ROS Robotic Operating System.

RRT Rapidly-Exploring Random Trees.

RTAB-Map Real-Time-Based Aparentia Mapping.

SLAM Simultaneous Localization and Mapping.

VFH Vector Field Histograma.

Referencias

- [1] J. C. Montesdeoca, M. C. P. Santos, M. Monllor y D. Herrera, «Trajectory tracking controller for differential-drive mobile robots,» en *2017 XVII Workshop on Information Processing and Control (RPIC)*, 2017, págs. 1-4. DOI: 10.23919/RPIC.2017.8211621.
- [2] S. Villacis y M. Antonio, «Control de un robot móvil para el seguimiento de objetos mediante realimentación visual,» 2018.
- [3] K. L. F. Rodríguez y F. Trujillo-Romero, «Desarrollo de un sistema lógico difuso para el control de la locomoción bípeda de un robot humanoide NAO,» *Res. Comput. Sci.*, vol. 113, págs. 181-194, 2016.
- [4] J. García-Blancas, O. A. Domínguez-Ramírez, E. E. Rodríguez-Torres y L. E. Ramos-Velasco, «Adaptabilidad en la tarea de control de exoesqueleto bípedo para fisioterapia asistida,» *Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI*, 2022.
- [5] L. H. R. y M. ., «MODELO MATEMÁTICO PARA UN ROBOT MÓVIL,» Español, *Scientia Et Technica*, vol. XIV, págs. 13-18, 2008, ISSN: 0122-1701. dirección: <https://www.redalyc.org/articulo.oa?id=84903803>.
- [6] A. Gutiérrez-Preciado, M. A. González-Palacios, L. A. Aguilera-Cortés y F. J. Mendoza-Patiño, «Optimización del Espacio de Trabajo de un Robot Deltoide,» *Revista Iberoamericana de Automática e Informática industrial*, 2019.
- [7] J. M. Gutierrez Arias, L. Hernández Angulo, M. M. Morín Castillo y J. E. Flores Mena, «Control Óptimo para Trayectorias Circulares en un Robot Móvil,» *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 8, n.º 3, págs. 229-240, 2011, ISSN: 1697-7912. DOI: <https://doi.org/10.1016/j.riai.2011.06.010>. dirección: <https://www.sciencedirect.com/science/article/pii/S1697791211000112>.

- [8] C. A. A. Peñalort, E. Martínez Oviedo y P. F. A. Cárdenas Herrera, «OPTIMIZACIÓN DIMENSIONAL DE UN ROBOT PARALELO TIPO DELTA BASADO EN EL MENOR CONSUMO DE ENERGÍA», en *Ciencia e Ingeniería Neogranadina*, vol. 21, págs. 73 -88, jun. de 2011, ISSN: 0124-8170. dirección: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0124-81702011000100005&nrm=iso.
- [9] C. P. N. Pilar y Q. C. Carolina, *Algoritmo estocástico para la optimización de la trayectoria de un robot*, 2005. dirección: <http://hdl.handle.net/20.500.12749/1324>.
- [10] G. Suarez-Rivera, N.-D. Muñoz-Ceballos y H.-M. Vázquez-Carvajal, «Development of an Adaptive Trajectory Tracking Control of Wheeled Mobile Robot,» en *Revista Facultad de Ingeniería*, vol. 30, mar. de 2021, ISSN: 0121-1129. dirección: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0121-11292021000100106&nrm=iso.
- [11] M. Serna, A. S. López, M. E. Analco, M. B. B. Loranca y E. E. P. G., «Navegación de robots móviles utilizando algoritmos Bugs extendidos,» *Res. Comput. Sci.*, vol. 148, n.º 8, págs. 159-171, 2019.
- [12] T. Osuna Altamirano, «Navegación y control de robot móvil,» Tesis doct., 2010.
- [13] T. Pire, «Localización y mapeo simultáneos mediante el uso de un sistema de visión estéreo,» 2017.
- [14] V. M. B. Narvaez, F. Yandún, D. Pozo, L. Morales, J. Rosero, A. Rosales y F. Auat, «Diseño e Implementación de un Sistema de Localización y Mapeo Simultáneos (SLAM) para la Plataforma Robótica ROBOTINO,» 2014.
- [15] M. Realpe y B. V. Burgos, «HACIA LA NAVIGACIÓN AUTÓNOMA DE ROBOTS A PARTIR DE LA IMPLEMENTACIÓN DE UN MÉTODO DE LOCALIZACIÓN Y MAPEO SIMULTÁNEOS (SLAM) MEDIANTE EL USO DE UN SISTEMA DE VISIÓN 3D,» 2006.
- [16] L. K. G. Obregón, L. A. F. Rincón y O. M. D. Suárez, «Diseño e implementación de un sistema de visión artificial usando una técnica de mapeo y localización simultánea (SLAM) sobre una plataforma robótica móvil,» *Mundo Fesc*, vol. 8, págs. 8-17, 2018.
- [17] P. López Torres, «Análisis de Algoritmos para Localización y Mapeado simultáneo de Objetos,» 2016.

- [18] D. D. Tenorio, «Diseño e implementación de un sistema para el mapeo y navegación de un robot móvil,» *Proyecto de Grado, Dto. Automática y electrónica, Univ. Autónoma de Occidente, Santiago de Cali, Valle, 2017.*
- [19] J. E. Jara Jimbo y J. P. Uruchima Calle, «Levantamiento de mapas de ambientes para la navegación autónoma de robots móviles,» B.S. thesis, 2021.
- [20] K. Hjartarson, *Dynamic Path Planning, Mapping, and Navigation for Autonomous GPR Survey Robots*, 2023.