Control de un robot móvil tipo uniciclo basado en rechazo activo de perturbaciones para el seguimiento de trayectorias

Francisco Javier Camacho Jaimes, Sebastián Jair Granados Canaria

Trabajo de Grado para obtener el título de Ingeniero Mecánico

Directora

Yennifer Yuliana Ríos Díaz

Doctora en Ciencias, Especialidad Ingeniería Eléctrica

Codirector.

Jose Jorge Carreño Zagarra

PhD. en Ingeniería- Control Automático

Universidad Industrial de Santander
Facultad de Ingenierías Fisicomecánicas
Escuela de Ingeniería Mecánica
Bucaramanga
2025

Dedicatoria

Principalmente a Dios por darme fuerzas y ánimo durante los momentos de duda e incertidumbre.

A mi familia, pilar fundamental en mi vida y en el desarrollo de este reto, a mis padres Carlos y María por su apoyo y comprensión durante este proceso, a mis hermanos Johana y Camilo que con sus palabras me dieron claridad en los momentos de dificultad, a mi tío Nacho y a Silvia gracias por su valioso aporte

Solo tengo gratitud por cada uno de ustedes, por hacer parte de este Proyecto.

-Sebastián Jair Granados Canaria

A mi familia, por el apoyo incondicional,

A mi madre Rosa Nelyda por sus esfuerzos, por acompañarme y darme ánimos de continuar siempre, a mis hermanas Laura, Silvia y Paula por la sabiduría transmitida, los consejos y el apoyo infinito que me dieron siempre, a Paola por haber acompañado mis días en la universidad y motivarme a ser mejor.

A todos aquellos que directa o indirectamente sumaron algo en mí durante este proceso.

-Francisco Javier Camacho Jaimes

Tabla de contenido

Lista de figuras

Lista de Tablas

Lista de Apéndices

Los apéndices están disponibles en el Repositorio Institucional

Apéndice A. COMUNICACIÓN_M_A_PID_v7

Apéndice B. IMPLEMENTACIÓN_ADRC&PID

Apéndice C. MANUAL DE USUARIO ADRC

Apéndice D. SIMULACIÓN_ADRC&PID

Resumen

Título: Control de un robot móvil tipo uniciclo basado en rechazo activo de perturbaciones para el seguimiento de trayectorias [*]

Autores: Francisco Javier Camacho Jaimes, Sebastián Jair Granados Canaria [**]

Palabras Clave: Uniciclo, robot móvil, control, ADRC, seguimiento de trayectorias.

Descripción: En las últimas décadas, diferentes procesos industriales han sufrido una transformación significativa, impulsada por el desarrollo acelerado de la robótica móvil, ganando un papel fundamental en sectores como manufactura, almacenamiento y distribución. En el seguimiento de trayectorias para robots móviles, los controladores tradicionales como el PID presentan limitaciones ante incertidumbres en el modelo dinámico o cambios en las condiciones del entorno, estas limitaciones se traducen en errores de seguimiento que afectan la precisión y la estabilidad del movimiento, por tanto, surge la necesidad de desarrollar estrategias de control como el rechazo activo de perturbaciones, que permite la adaptación eficiente a cambios del entorno o incertidumbres de los modelos.

En el presente proyecto se aborda el problema de seguimiento de trayectorias para un robot uniciclo a través de un modelo de control basado en Rechazo Activo de Perturbaciones (ADRC por sus siglas en ingles Active Disturbance Rejection Control), el controlador consiste en un modelo de control PID, que cuenta con una variable de control adicional encargada de filtrar la información y garantizar un mejor rendimiento. Se implementó un bloque de control PID clásico y se comparó con el bloque basado en ADRC para tres tipos de trayectorias: Lineal, circular y Senoidal, donde se obtuvieron resultados que demostraron la superioridad del bloque de control propuesto en comparación con el PID tradicional, bajo el modelo de control basado en ADRC se logró un movimiento con menor oscilación, y mayor estabilización de errores, además se observó que el tiempo de respuesta es menor en el controlador propuesto.

---

[*] Trabajo de Grado

[**] Facultad de Ingenierías fisicomecánicas Escuela de Ingeniería mecánica. Director: Yennifer Yuliana Ríos Díaz. Doctora en Ciencias, Especialidad Ingeniería Eléctrica. Codirector: Jose Jorge Carreño Zagarra. PhD. en Ingeniería-Control automático

Abstract

Títle: Unicycle mobile robot control based on active disturbance rejection for path tracking [*]

Authors: Francisco Javier Camacho Jaimes, Sebastián Jair Granados Canaria [**]

Key Words: Unicycle, mobile robot, control, ADRC, path tracking.

Descripción: In recent decades, various industrial processes have undergone significant transformation, driven by the accelerated development of mobile robotics, gaining a fundamental role in sectors such as manufacturing, warehousing, and distribution. In path tracking for mobile robots, traditional controllers such as PID controllers present limitations when faced with uncertainties in the dynamic model or changes in environmental conditions. These limitations translate into tracking errors that affect motion accuracy and stability. Therefore, there is a need to develop control strategies such as active disturbance rejection, which enables efficient adaptation to environmental changes or model uncertainties.

This project addresses the problem of path tracking for a unicycle robot through a control model based on Active Disturbance Rejection Control (ADRC). The controller consists of a PID control model with an additional control variable responsible for filtering information and ensuring improved performance. A classic PID control block was implemented and compared with the ADRC-based block for three types of trajectories: Linear, circular and Sinusoidal, where results were obtained that demonstrated the superiority of the proposed control block compared to the traditional PID, under the ADRC-based control model, a movement with less oscillation and greater error stabilization was achieved, in addition, it will be detected that the response time is lower in the proposed controller.

---

[*] Degree Work

[**] Faculty of Physical and Mechanical Engineering, School of Mechanical Engineering. Director: Yennifer Yuliana Ríos Díaz. PhD in Electrical Engineering. Co-director: Jose Jorge Carreño Zagarra. PhD inAutomatic Control Engineering

Introducción

El robot de tipo uniciclo es una estructura que consta de dos ruedas cuyos ejes son colineales, estas ruedas son controladas de manera independiente y además poseen una rueda adicional no controlada con el objetivo de estabilizarlo (Il Bambino, 2008). Dentro de los diferentes tipos de robots, los uniciclos se utilizan con frecuencia para realizar diferentes tareas gracias a su movilidad y configuración, esta estructura se ha utilizado en diferentes aplicaciones como vigilancia, limpieza de suelos y transporte carga industrial (Jeschke, Liu, & Schilberg, 2011). En términos generales se ha visto un crecimiento expansivo en el área de la robótica, impulsado por los avances en la computación y la electrónica, los robots están generando una nueva perspectiva en los procedimientos que se emplean en la agricultura, la minería y la industria en general. (L. García, 2010). Un reto importante en el uso de estas plataformas móviles es el seguimiento de trayectorias, donde se han implementado desde algoritmos simples hasta estrategias de planeación y seguimiento más complejas que permiten a los robots cumplir con esta tarea (Yandún Torres, 2011).

Una estrategia popularizada en los últimos años para atender problemas de ingeniería moderna, incluyendo el área de los robots móviles es el control mediante el rechazo activo de perturbaciones, por ejemplo, se ha utilizado esta estrategia para controlar vehículos autónomos o estabilizar bicicletas sin necesidad de conductor (Ramírez-Neria et al., 2022).

El objetivo principal del presente proyecto es desarrollar e implementar un algoritmo de control basado en ADRC para seguimiento de trayectorias en un robot de tipo uniciclo, lo cual representa un reto de gran magnitud el área de control, y supone un problema que tiene amplias aplicaciones en diversos campos de la ingeniería para equipos móviles no tripulados.

1. Objetivos

1.1 Objetivo general

Desarrollar un algoritmo de control robusto para el seguimiento de trayectorias en un móvil uniciclo mediante el enfoque de rechazo activo de perturbaciones.

1.2 Objetivos específicos

- Diseñar un controlador multivariable basado en rechazo activo de perturbaciones para el seguimiento de trayectorias en el robot.
- Validar el funcionamiento del controlador diseñado en el robot móvil uniciclo.
- Generar un manual de usuario que contenga instrucciones claras para el uso del robot móvil uniciclo.

El documento de trabajo de grado está estructurado de la siguiente manera:

En el capítulo dos se aborda la problemática y el contexto general del seguimiento de trayectorias mediante estrategias de control. En el tercer capítulo del libro, se abordan las generalidades y el diseño del controlador, explorando alternativas como el control PID, antes de presentar el bloque de control basado en rechazo activo de perturbaciones. En el cuarto capítulo, se exponen los procedimientos de simulación del controlador diseñado, y la puesta en marcha de la estrategia diseñada para diferentes trayectorias. En el capítulo cinco, se contextualiza sobre el manual de usuario diseñado para la replicación de la estrategia de control expuesta en el presente proyecto. Finalmente, se incluyen algunas recomendaciones, alternativas para abordar trabajos futuros y conclusiones.

2. Marco referencial

El desarrollo de sistemas de control para robots móviles ha sido un área de creciente interés en la ingeniería moderna, en aplicaciones de manufactura, logística, exploración autónoma y robótica de asistencia (Canudas de Wit, Siciliano & Bastin, 1996). La capacidad de estos robots para operar en entornos dinámicos y adaptarse a cambios inesperados depende en gran medida de la robustez de sus sistemas de control (Oriolo et al., 2002). Entre las estrategias de control más estudiadas, el Control de Rechazo Activo de Perturbaciones (ADRC) ha emergido como una alternativa eficiente y flexible frente a métodos clásicos como el Control Proporcional-Integral-Derivativo (PID) (Gao et al., 2010). El ADRC ofrece una mayor capacidad de adaptación ante perturbaciones externas y modelos inciertos, lo que lo hace especialmente útil en aplicaciones donde los robots deben navegar en entornos no estructurados (Ramírez-Neria et al., 2022).

Este capítulo aborda las principales características de los sistemas de control para robots móviles, enfocándose en sus configuraciones estructurales, los métodos de control convencionales y las ventajas de un sistema de control basado en ADRC frente a un PID.

2.1 Tipos de Robots Móviles

Los robots móviles pueden clasificarse según su configuración estructural y su sistema de tracción. La elección del sistema de ruedas y la disposición de los motores influye directamente en la estrategia de control que debe implementarse para lograr un movimiento preciso y eficiente (Murray, Li & Sastry, 1994).

2.1.1 Robots Omnidireccionales

Los robots omnidireccionales están diseñados para moverse en cualquier dirección sin necesidad de girar su estructura principal. Esto se logra mediante el uso de ruedas Mecanum o ruedas esféricas, permitiendo movimientos laterales, diagonales y circulares con alta precisión (Gutiérrez Martínez, 2021). Son ampliamente utilizados en almacenes inteligentes y plataformas de asistencia en espacios reducidos.

2.1.2 Robots Cuatriciclos

Este tipo de robots se asemejan a los vehículos convencionales de cuatro ruedas, con un sistema de dirección que permite giros controlados. Su estabilidad y facilidad de control los hacen adecuados para aplicaciones industriales y agrícolas (Méndez Canú, 2021).

2.1.3 Robots Triciclos

Los robots triciclos suelen incorporar dos ruedas traseras motorizadas y una rueda delantera giratoria. Este diseño proporciona estabilidad y eficiencia en la conducción, aunque puede presentar limitaciones en cuanto a maniobrabilidad en espacios reducidos (Canudas de Wit et al., 1996).

2.1.4 Robots Uniciclos

Los robots uniciclos se caracterizan por tener dos ruedas fijas que son controladas de manera independiente, junto con una rueda libre que actúa como apoyo. Este diseño permite un alto grado de flexibilidad en el movimiento, aunque requiere algoritmos de control avanzados para mantener la estabilidad dinámica (Gutiérrez Martínez, 2021). Para el desarrollo del presente proyecto se consideró esta configuración mecánica, y cinemática.

2.2 Generalidades de los Sistemas de Control en Robots Móviles

El control de robots móviles implica el diseño de algoritmos capaces de regular su velocidad, orientación y trayectoria, garantizando estabilidad y respuesta eficiente ante perturbaciones externas (Slotine & Li, 1991). El ADRC es una estrategia relativamente nueva que posee potencial para resolver problemas de control en procesos con altos niveles de incertidumbre (Yi, 2014).

2.2.1 Control Proporcional-Integral-Derivativo (PID)

El controlador PID ha sido un método de control ampliamente empleado en la automatización industrial desde que fue introducido por Minorsky en 1922. Su diseño se basa en la combinación de tres términos fundamentales:

- Proporcional (P): Ajusta la salida de control en función de la desviación del sistema respecto al valor deseado.
- Integral (I): Corrige desviaciones persistentes acumulando el error a lo largo del tiempo.
- Derivativo (D): Aplica correcciones anticipadas para evitar sobrepasos y oscilaciones excesivas.

Si bien el PID es efectivo en sistemas con dinámicas bien definidas, presenta limitaciones ante perturbaciones o modelos inciertos, lo que ha motivado la exploración de estrategias robustas como el ADRC (Gao et al., 2010).

2.2.2 Control de Rechazo Activo de Perturbaciones (ADRC)

El ADRC es una técnica moderna de control que mejora las deficiencias del PID al estimar perturbaciones en tiempo real y compensarlas de manera proactiva (Han, 2009). A diferencia del

PID, que ajusta la respuesta del sistema en función del error medido, el control mediante rechazo activo de perturbaciones introduce un observador que permite mejorar la capacidad de respuesta del sistema ante cambios inesperados en la dinámica del entorno (Ramírez-Neria et al., 2022).

En (Gao, 2014) se destacan tres características sobre el control mediante rechazo activo de perturbaciones:

- No es necesario un modelo complejo del proceso que se desea controlar, pues solo es necesario el orden del sistema, y la ganancia crítica, es decir el parámetro que relaciona la entrada y la enésima derivada de la salida.
- Se relacionan perturbaciones e incertidumbres bajo el nombre de perturbación total, para cancelarlas mediante la acción de control y evitar que se reflejen a la salida.
- Induce al sistema real a comportarse como una planta nominal que facilita el diseño del controlador.

Es importante mencionar que es posible obtener modelos de control basados en rechazo activo de perturbaciones, que garanticen una mayor estabilidad que un PID, sin llegar a la precisión de un ADRC completo, para el caso concreto del presente proyecto se emplea un PID filtrado, que garantice una reacción rápida, y un seguimiento adecuado de las referencias.

2.3 Trabajos relacionados

A continuación, se nombran algunos trabajos utilizados como referencia para el desarrollo y construcción del presente proyecto.

A nivel local, se encuentran dos proyectos desarrollados en la Universidad Industrial de Santander el primero de ellos, por (Villamizar y Rangel, 2022) que un referente clave debido a su enfoque en el diseño y construcción de un robot móvil autónomo, lo que proporciona una base

fundamental para comprender aspectos esenciales como la odometría y la navegación en entornos reales. Es Importante mencionar el trabajo de Ramírez-Neria et al. (2022), en el cual se aborda la sincronización robusta en robots móviles de tracción diferencial mediante un esquema de Active Disturbance Rejection Control (ADRC), utilizando un Observador de Estado Extendido (ESO) de alto orden para estimar y compensar perturbaciones externas.

Adicionalmente (Zhao, 2023) presenta un trabajo presenta con enfoque integral para la modelización y control de robots móviles tipo diferencial, abordando tanto su comportamiento cinemático como dinámico para mejorar el seguimiento de trayectorias, lo cual es clave para el desarrollo del presente proyecto pues ofrece una perspectiva muy similar en cuanto a ecuaciones de movimiento para el seguimiento de trayectorias.

2.4 Referencias de robots uniciclos en el mercado

Es fundamental que el robot cumpla una serie de requerimientos para hacer posible la implementación del controlador:

- Placa de control reprogramable.
- Compatibilidad con sistemas de comunicación inalámbrica.
- Sensores de lectura de velocidades en los motores (Encoders).

A continuación, se describen algunas de las opciones de robots móviles disponibles en el mercado:

2.4.1 MBot2

El mBot2 es un robot móvil de 3 ruedas que cuenta con una tarjeta Cyber Pi, Infrarrojos seguidores de línea, acelerómetro, sensor de ultrasonido, motores DC con encoders de cuadratura,

además cuenta con un manual de programación guiada en el uso de cada uno de los sensores y demás componentes.

2.4.2 Pico Robot

Pico Robot es un robot móvil de 3 ruedas basado en la placa de desarrollo PICO de Raspberry Pi. Es un robot de código abierto ampliamente utilizado por su gran versatilidad y desempeño, sus principales compradores son investigadores y estudiantes. Este robot móvil permite realizar funciones como evitar obstáculos, seguidor de línea y el control remoto ya sea por vía infrarroja o sonora. Al contar con una placa de desarrollo de alto rendimiento como la Raspberry Pi Pico permite realizar cambios en la programación de manera fácil, ya que cuenta con una interfaz digital flexible basada en micro python.

2.4.3 AlphaBot

AlphaBot es un robot móvil de 3 ruedas con una placa de desarrollo compatible con Raspberry Pi y Arduino. Está formada por la placa base AlphaBot un chasis móvil, motores dc, sensores infrarrojos y sensores de comunicación vía bluetooth (HC05). Cuenta con una interfaz de código abierto y un programa base de seguimiento de línea, evitación de obstáculos, monitoreo de video, control remoto WiFi/Bluetooth/ZigBee/infrarrojos.

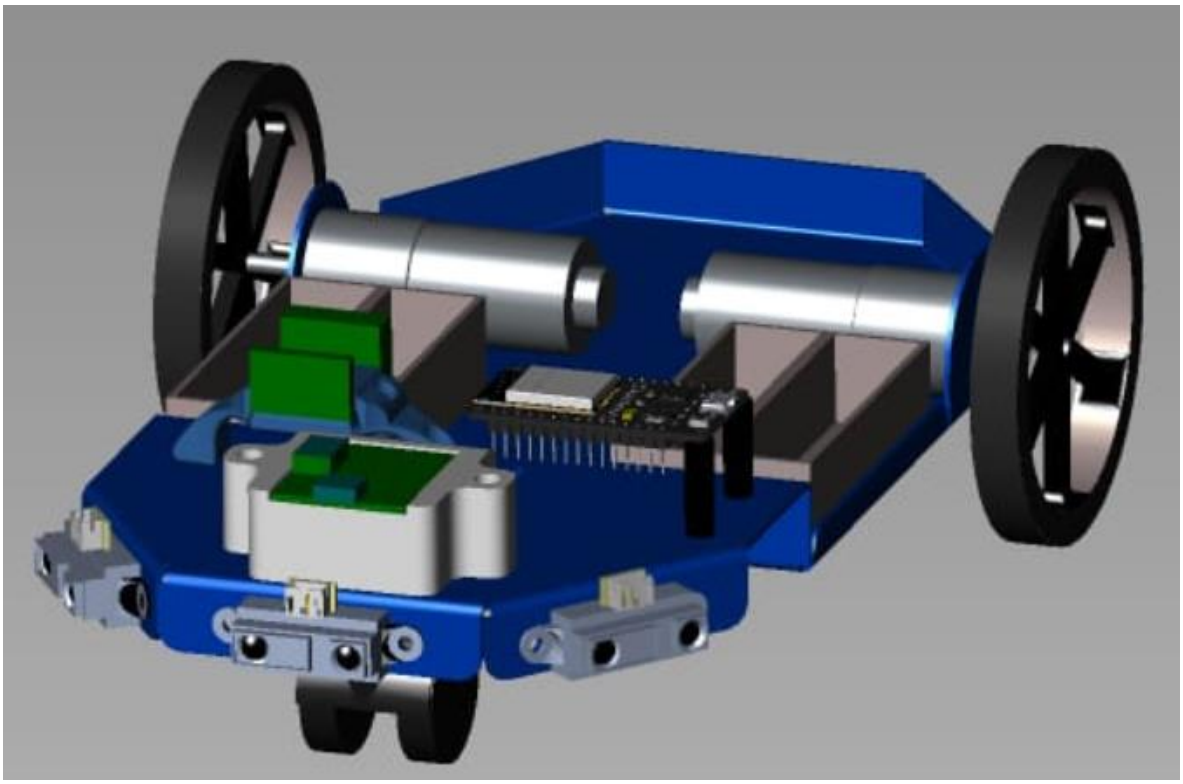
2.4.4 Rover_V2.0

Es un robot móvil de 3 ruedas con un módulo de comunicación ESP32 que permite tener una comunicación vía Bluetooth o wifi en tiempo real, además de contar con motores dc con encoders de cuadratura, sensores infrarrojos en su parte frontal y un sensor de posición(giroscopio) el cual permite saber su posición en el plano XY, cuanta con 2 fuentes de alimentación separadas, una para los motores y otra para los módulos de comunicación y detección de obstáculos, este

robot es de código abierto y es compatible con arduino y Matlab brindando una facilidad en la comunicación y recepción de los datos. En la Figura 1, se muestra el modelo del robot.

Figura 1.

Modelo de robot Rover_V2.0



2.5 Selección del robot

Teniendo en cuenta que el desarrollo del robot se encuentra en un entorno educativo, según (Solano, Vázquez y Vargas, 2024) es importante la utilización de una placa de control de código abierto, y además encoders que garanticen precisión en la captación de datos, para así lograr un rendimiento satisfactorio.

De los modelos presentados el que más se acercó a los parámetros de trabajo buscados y requerimientos fue el robot móvil diferencial ROVER_V2.0, pues como se muestra en la Tabla 1. El ROVER_V2.0 cumple con los requerimientos para el desarrollo del proyecto.

Tabla 1.

Criterios de selección del robot

Robot	Encoders	Placa de Control	Comunicación	Reprogramable
MBot2	Si	Cyber pi	Wifi/bluetooth	No
Pico robot	No	Raspberry pi	Bluetooth	Si
AlphaBot	No	Arduino / Raspberry	Bluetooth	Si
Rober_V2.0	Si	Arduino	Wifi/bluetooth	Si

El robot móvil ROVER_V2.0 se clasifica como robot móvil tipo diferencial y está equipado con los siguientes componentes, incluyendo sensores actuadores y baterías de alimentación:

- 2 moto reductores DC de 6v con encoder magnético de cuadratura
- 2 baterías de litio 18650 de 3.7v de 3500mha
- 1 ESP32 (Modulo de comunicación vía wifi o bluetooth)
- 3 sensores infrarrojos análogos
- 1 giroscopio

3. Diseño del controlador multivariable para control de trayectorias

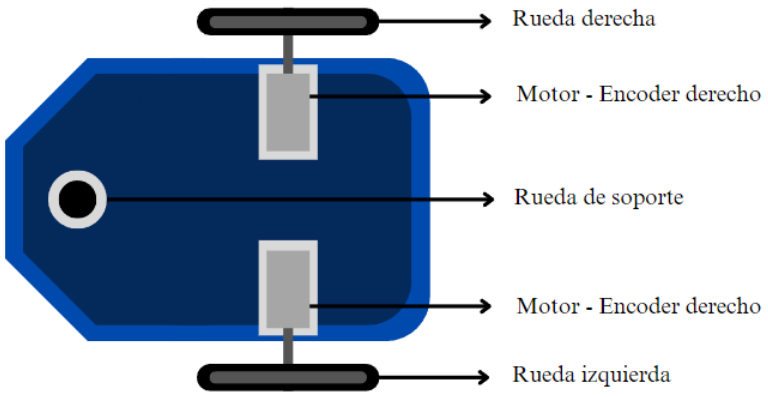
En el presente capitulo se describen los diferentes aspectos que se consideraron para diseñar un controlador multivariable para seguimiento de trayectorias.

3.1 Modelo del robot

Los robots diferenciales se clasifican en diferentes tipos según la disposición y funcionalidad de sus ruedas, se puede describir la naturaleza mecánica de los robots diferenciales de tipo uniciclo como la combinación de 3 ruedas en un mismo chasis rígido, que cumpla con la configuración ilustrada en la Figura 2.

Figura 2.

Configuración mecánica de un robot móvil tipo uniciclo



Las dos ruedas motrices tienen su propia unidad de transmisión de potencia, lo cual permite controlar la velocidad y dirección de giro de cada una de ellas de forma independiente, la rueda de soporte no está conectada a ningún motor, y su única función es brindar un punto de apoyo adicional, para ayudar a mantener la estabilidad del robot.

3.1.1 Características Cinemáticas del robot

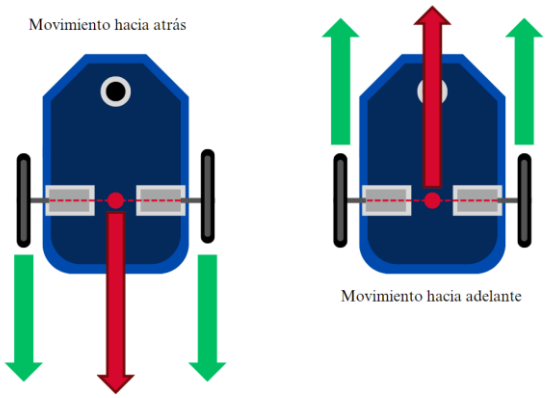
Los robots móviles, de tracción diferencial se caracterizan por utilizar sistemas de transmisión de dos ruedas independientes, es decir que cada una de estar está asociada a un motor diferente.

Cada una de las trayectorias que el robot pueda describir están sujetas a la diferencia de velocidades en sus ruedas. Por otra parte, el sentido de giro de los motores determina la orientación del movimiento, y en conjunto con las diferencias de velocidades puede generar movimientos de rotación, de traslación o una combinación de ellos.

Ambas ruedas girando a la misma velocidad: Si ambas ruedas giran a la misma velocidad, y en la misma dirección se obtienen velocidades lineales, y el robot podrá moverse hacia delante, y hacia atrás, como se observa en la Figura 3.

Figura 3.

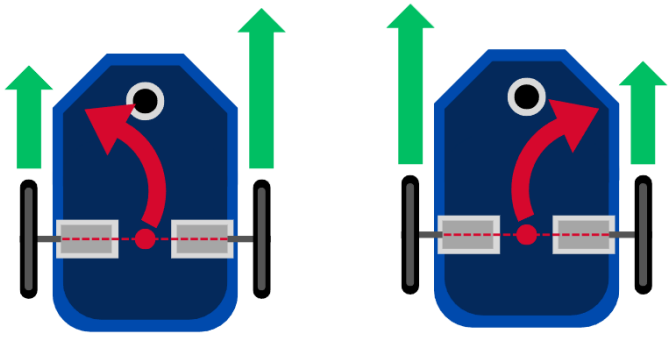
Movimiento Tipo 1



Ruedas girando a diferentes velocidades: Si ambas ruedas giran en la misma dirección, pero con velocidades diferentes se va a producir un impulso de giro que va a internar mover el robot hacia la dirección del motor más lento como se observa en la Figura 4.

Figura 4.

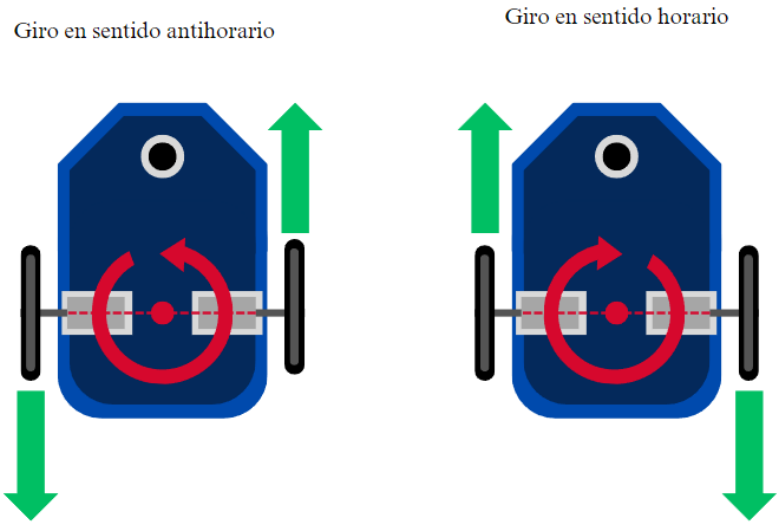
Movimiento Tipo 2



Ruedas girando a velocidades iguales, pero en dirección opuesta: Para este escenario, las velocidades se van a contrarrestar produciendo la rotación del robot justo sobre su eje central. Y se puede producir en sentido horario, y en sentido antihorario, como se observa en la Figura 5.

Figura 5.

Movimiento tipo 3

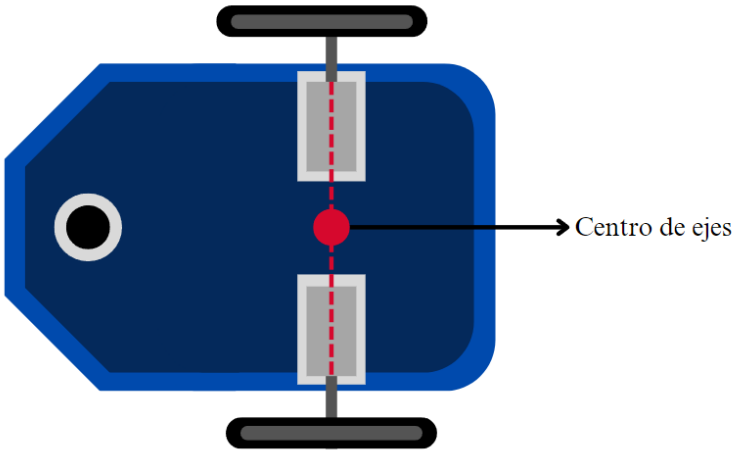


3.1.2 Modelo cinemático del robot

Para el desarrollo del modelo cinemático del robot, es importante considerar el robot como un cuerpo rígido, de masa uniforme, es decir que se desarrolla como un punto en el espacio, para esto se va a trazar una línea que conecte los ejes de los motores, y en el punto medio se va a ubicar el punto de equivalencia de robot, este es justamente el punto que se controla cinemáticamente, como se observa en la Figura 6.

Figura 6.

Centro de control cinemático del robot

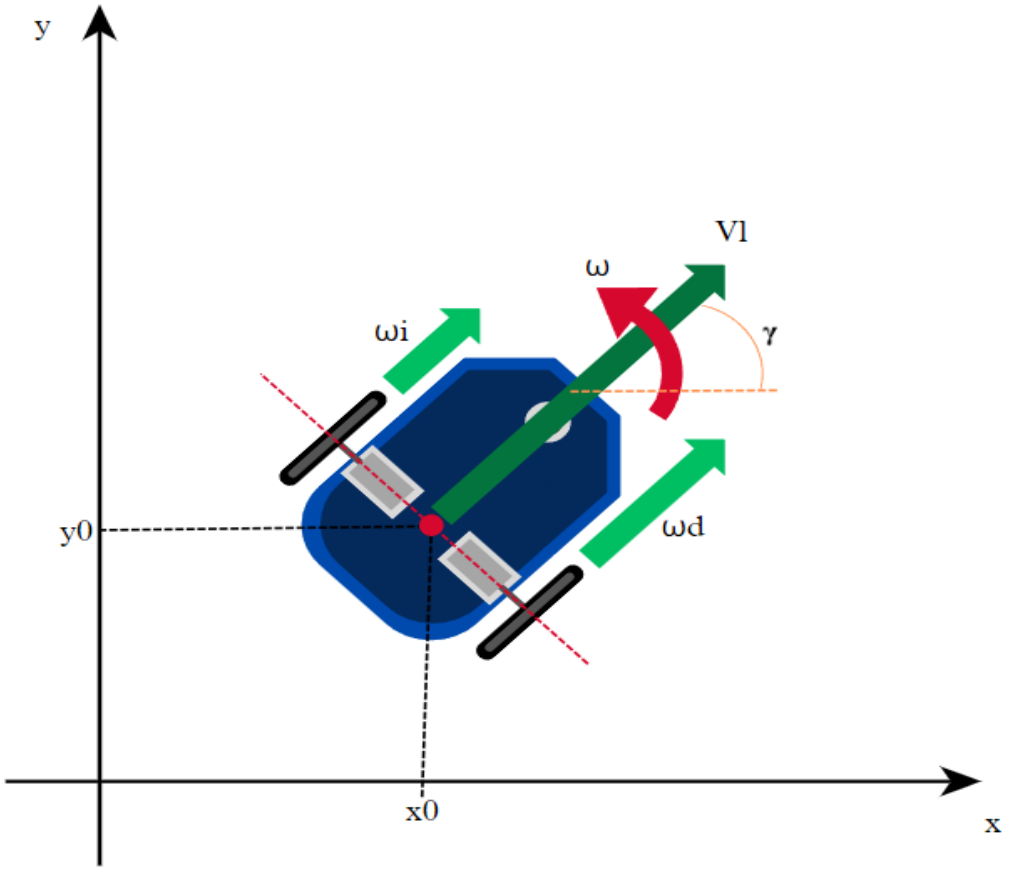


El modelo cinemático del robot es necesario para desarrollar la estrategia de control, conociendo las características mecánicas del robot expuestas en el capítulo anterior, se establece que el sistema tiene dos entradas, una velocidad angular presente en la rueda derecha, y una velocidad angular presente en la rueda izquierda, y como producto de la interacción de estas dos velocidades se generar una velocidad lineal y una Velocidad angular que aportan movimiento al robot de forma combinada.

El robot se encuentra en un plano cartesiano, donde (x0, y0) son las coordenadas de ubicación del punto de control cinemático del robot, el robot cuenta con un radio de rueda de 40.25 milímetros, y una distancia entre ellas de 160 milímetros.

Figura 7.

Modelo cinemático del robot



la relación entre las velocidades individuales de las ruedas con el movimiento general del robot está dada por las ecuaciones 1 y 2, como se describe a continuación:

$$vl = \frac{r(\omega i + \omega d)}{2} \qquad (1)$$

$$\omega = \frac{r(\omega i + \omega d)}{l} \qquad (2)$$

Donde:

- $\boldsymbol{\omega}$ es la velocidad angular del robot
- $\boldsymbol{vl}$ es la velocidad lineal del robot
- $\boldsymbol{r}$ es el radio de las ruedas,
- $\boldsymbol{\omega i}$ es la velocidad angular del motor izquierdo
- $\boldsymbol{\omega d}$ es la velocidad angular del motor derecho
- $\boldsymbol{l}$ es la longitud que separa las dos ruedas
- γ es el ángulo entre el vector de velocidad lineal y el eje x

La representación matricial de las velocidades del robot uniciclo se muestran en la ecuación 3.

$$\begin{bmatrix} vl \\ \omega \end{bmatrix} = \begin{bmatrix} r/2 & r/2 \\ -r/l & r/l \end{bmatrix} \begin{bmatrix} \omega i \\ \omega d \end{bmatrix} = Ar \begin{bmatrix} \omega i \\ \omega d \end{bmatrix} \quad (3)$$

Adicionalmente existe una relación entre el movimiento del robot en el plano, la velocidad lineal y la velocidad angular, como se muestra en las ecuaciones 4, 5 y 6, entendiendo la velocidad como la derivada de la posición respecto al tiempo.

$$\dot{x}0 = v * \cos(\gamma) \quad (4)$$

$$\dot{y}0 = v * sen(\gamma) \quad (5)$$

$$\gamma = \omega \quad (6)$$

3.1.3 Simulación del modelo cinemático

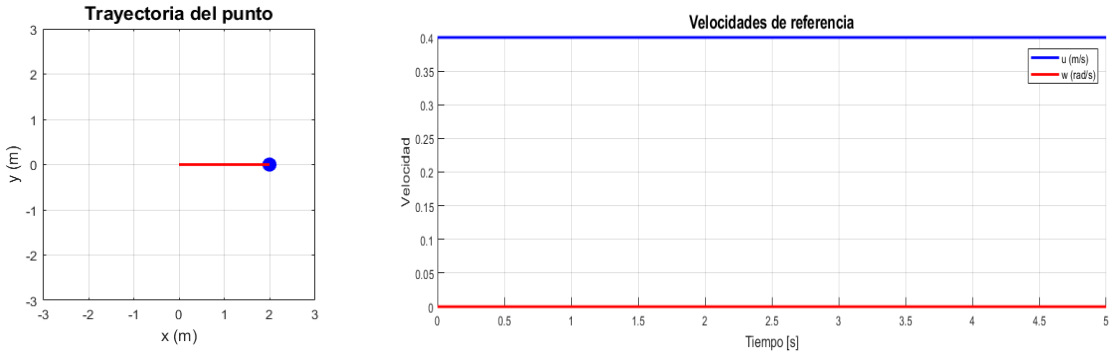
Para la evaluación del modelo cinemático anteriormente expuesto se emplea la herramienta MATLAB, donde realizando una variación de velocidades angulares de los motores, se puede comprobar el comportamiento de un punto en el espacio, en las Figuras 8, 9 y 10 se grafica el

movimiento ideal del robot (Punto azul) y la trayectoria realizada por el mismo (Línea roja), adicionalmente se grafican la velocidad lineal y angular del robot.

- **Caso 1:** Cuando se simula velocidad angular de igual magnitud y mismo sentido en ambos motores, se obtiene un movimiento perfectamente lineal, como muestra la Figura 8.

Figura 8.

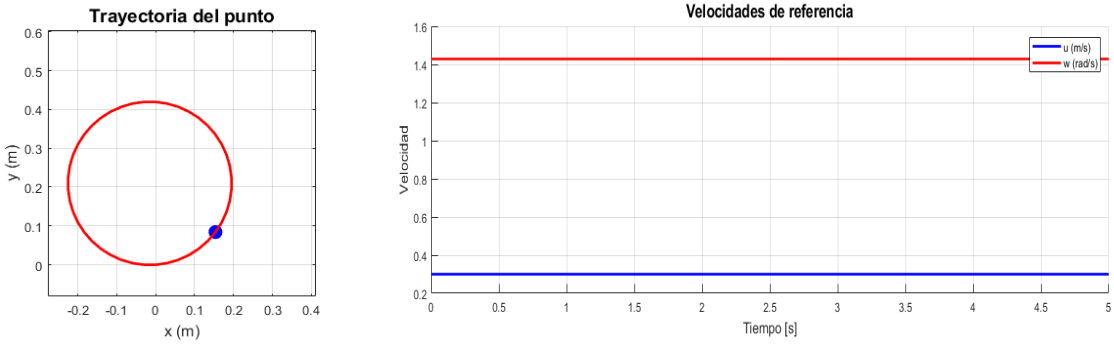
Simulación movimiento tipo 1



- **Caso 2:** Cuando se simulan velocidades angulares en los dos motores en el mismo sentido, pero con diferente magnitud, se obtiene una trayectoria circular como muestra la Figura 9.

Figura 9.

Simulación movimiento tipo 2

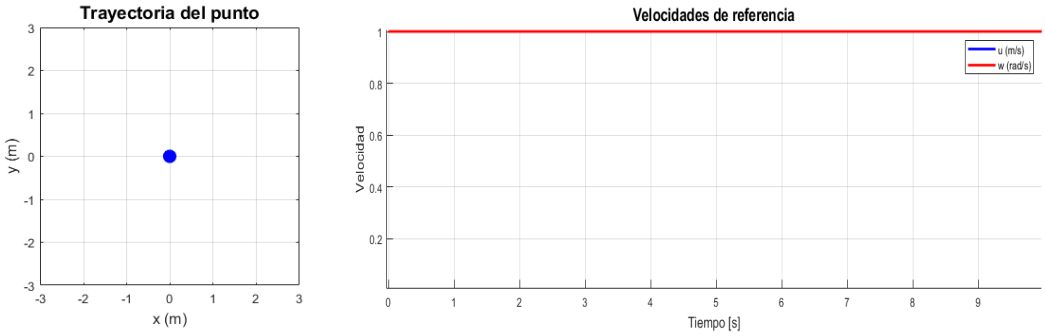


- **Caso 3:** Cuando se simulan velocidades angulares en los motores de igual magnitud, pero en sentidos contrarios, se obtiene el fenómeno de rotación sobre el eje central, el punto de

estudio cinemático no cambia de coordenadas en el campo (x,y), es decir el robot se encuentra girando sobre su propio eje, sin sufrir cambio de coordenadas, como se muestra en la figura 10.

Figura 10.

Simulación movimiento tipo 3



3.2 Detalles técnicos del robot móvil uniciclo

En el presente segmento se describen los componentes que hacen parte del robot móvil uniciclo, se trabaja con una combinación de sistemas mecánicos y electrónicos que permiten el movimiento del robot, y además facilita el desarrollo de algoritmos de control. En este caso específico el robot está compuesto por dos motores que incluyen un sistema de encoders, tres sensores de proximidad infrarrojos, dos fuentes de alimentación, un microcontrolador y un puente H controlador de motores.

Con el objetivo de mejorar la estética del robot y brindar protección al cableado, se diseña una carcasa protectora (Figura 11) que se imprime en 3D y se instala sobre el Rover_V2.0, como se muestra en las Figura 12.

Figura 11.

Modelo virtual del robot

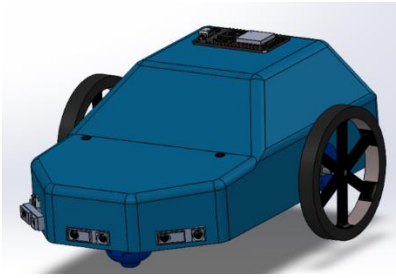


Figura 12.

Modelo físico del robot

3.2.1 Componentes electronicos del Robot

A continuación, detallan los componentes utilizados en el ensamble electrónico del robot móvil, para la integración y control del sistema.

Placa ESP WROOM 32.

Es un módulo microcontrolador que incluye un procesador Dual Core, que permite tener velocidades de hasta 240 MHz, lo que facilita el procesamiento de datos de forma eficiente en tareas complejas, también cuenta con conectividad Wifi, o Bluetooth, lo que es muy útil al momento de recibir o enviar datos de forma inalámbrica. También cuenta con una memoria de 4 MB y 520 KB de SRAM, lo que lo hace uno de los microcontroladores más populares y utilizados en la actualidad.

Sensores de Proximidad Infrarrojos tipo SHARP

Este tipo de sensor es conocido por utilizar un sistema de triangulación óptica para detectar la posición de los objetos, básicamente los sensores emiten un haz de luz infrarroja para que rebote contra un objeto y sea recibida de vuelta por un fotodetector, es el ángulo de llegada del haz de luz lo que permite el cálculo de las distancias. Es ideal su uso en robótica de navegación, porque su rango de detección es más amplio que el de los sensores habitualmente utilizados.

Motorreductor con encoder GM25370

Este tipo de motor cuenta con una caja de engranajes que permite trabajos con mayor toque a menor velocidad, adicionalmente trae implementado un encoder que convierte la rotación del motor en señales digitales que pueden ser interpretadas por un microcontrolador, resulta muy útil su uso cuando se requiere una medición exacta de velocidades angulares en los motores, como es el caso de este proyecto.

Controlador de motor L298N

Es un controlador de motores ampliamente utilizado en la rama del control de vehículos, es un controlador doble puente H, y tiene la capacidad de controlar motores de corriente continua, y motores paso a paso, es útil para el desarrollo del proyecto pues permite controlar la velocidad y el sentido de dos motores de forma independiente.

Sensor MPU6050

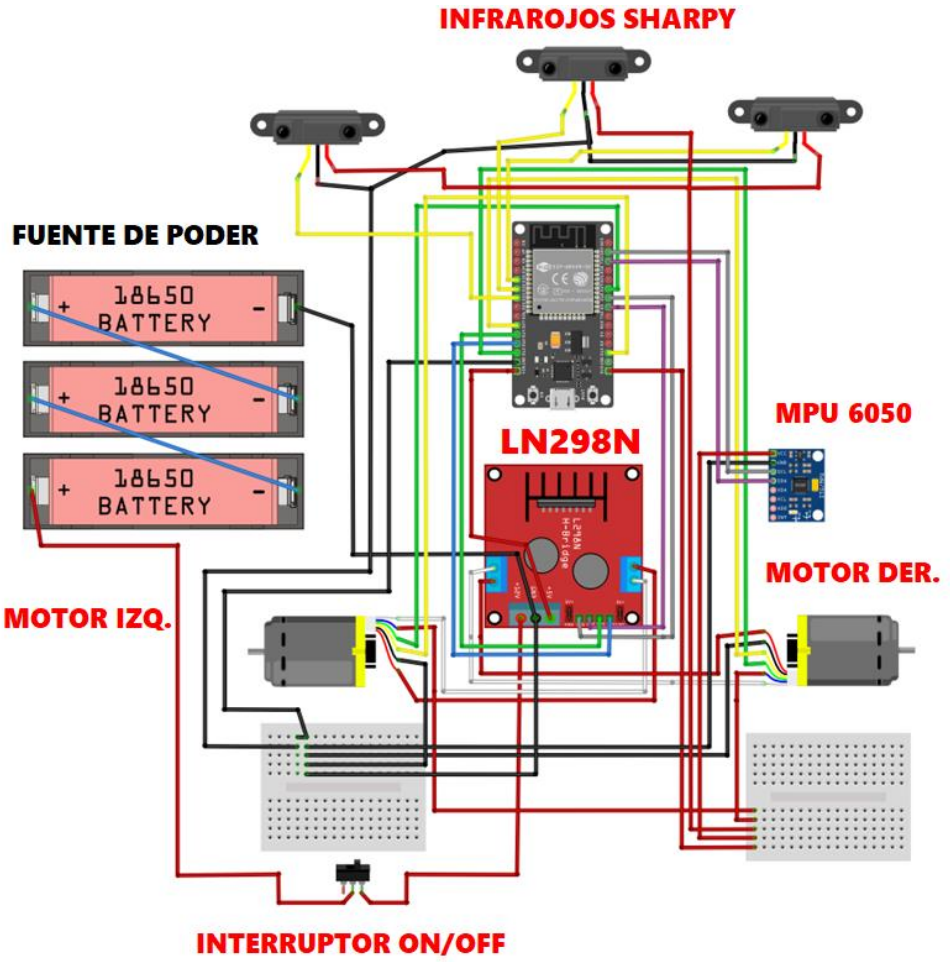
Es un tipo de sensor versátil que se suele utilizar en diferentes sistemas de control de movimiento, cuenta con un acelerómetro y un giroscopio de tres ejes, lo que le permite medir la aceleración lineal y la velocidad angular de forma conjunta.

3.2.2 Montaje electrónico del robot móvil

En la figura 13, se muestra la conexión de los componentes electrónicos expuestos en el capítulo anterior. El montaje se ha hecho en la plataforma FRITZING, que es un programa de uso libre enfocado en diseños electrónicos, y la configuración que se utilizó permite el uso de los componentes de forma simultánea y precisa.

Figura 13.

Montaje electrónico del robot móvil- Fritzing

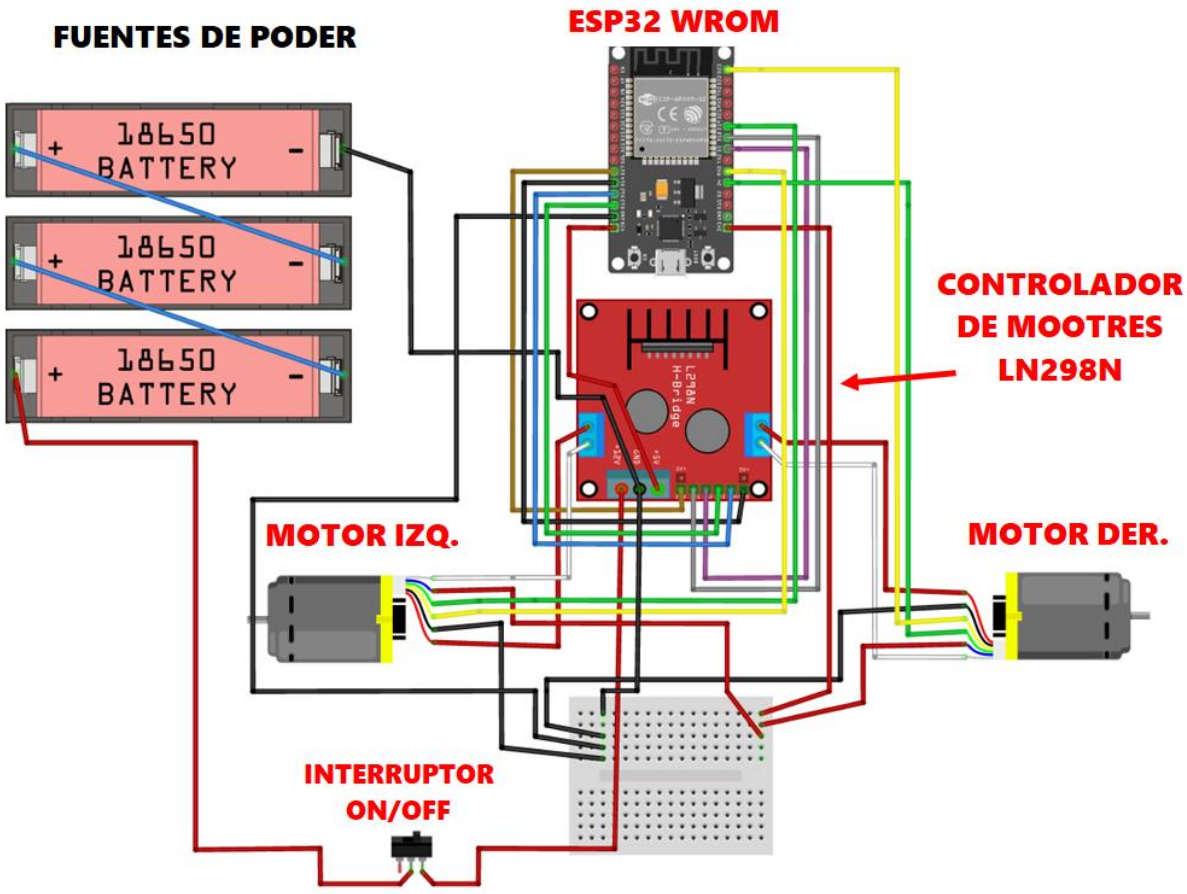


Para los efectos prácticos asociados al desarrollo del presente proyecto se hace la simplificación del montaje electrónico del robot, descartando los sistemas de detección de

obstáculos mediante sensores infrarrojos y el sensor MPU6050, como se muestra en la Figura 14, es importante resaltar que la presencia de estos sensores en el robot son oportunidades para abordar en trabajos futuros, como se describe en el capítulo 6.

Figura 14.

Montaje electrónico simplificado del robot móvil- Fritzing



3.3 Sistemas de Comunicación de la ESP32 WROOM 32

Teniendo en cuenta que el robot debe estar libre para poder moverse, se hace necesario implementar un sistema de comunicación inalámbrica que permita el intercambio de datos que para manipular o controlar los actuadores, y enviar la información que recogen los sensores, el objetivo de la comunicación es crear un puente entre la computadora (**Matlab**) y la unidad

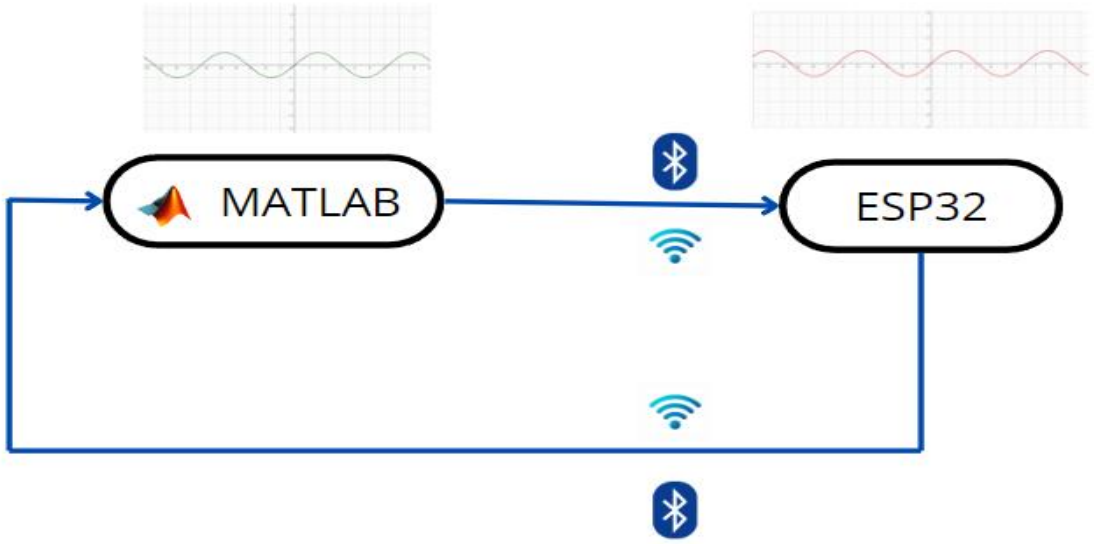
microcontroladora (**ESP32**), debido a la versatilidad de la placa, existió la posibilidad de hacer el intercambio de datos a través de Bluetooth o vía Wifi.

Para validar los canales de comunicación que tiene disponible la placa se hizo el intercambio de una función senoidal entre Matlab y la EPS32, para así visualizar el tiempo de respuesta y el alcance de la señal en cada escenario.

Desde Matlab se genera la función $y = sen(x)$ y se envía de forma inalámbrica a la placa ESP32, Luego se recibe la señal y nuevamente es enviada desde la ESP32 vía Bluetooth o Wifi, de vuelta a Matlab, donde es posible visualizar los datos recibidos, la lógica de envío de datos se describe en la Figura 15.

Figura 15.

Comunicación Matlab- ESP32



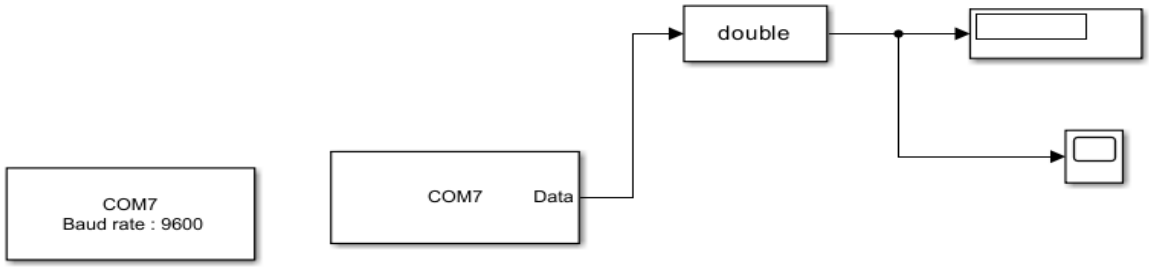
3.3.1 Prueba de Sistema de Comunicación a través de Bluetooth

Para establecer la comunicación a través de bluetooth es necesario preparar la placa ESP32 para el envío de la función Senoidal, la programación de la placa se realiza a través del Arduino

IDE. Y la recepción de datos se realiza desde Simulink, a través de un puerto virtual, que se asigna a los dispositivos Bluetooth, como se muestra en la Figura 16.

Figura 16.

Recepción de datos vía Bluetooth- Simulink



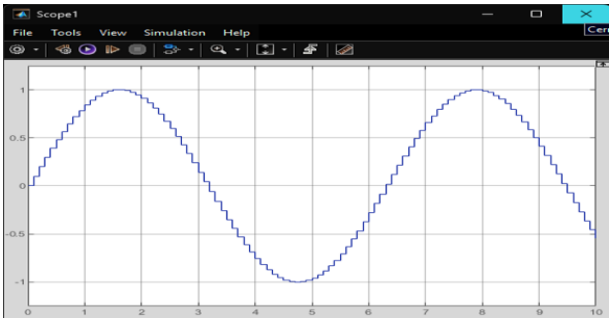
Se establece la comunicación entre el computador y la placa ESP32 vía bluetooth, y se usa el puerto virtual desde Simulink, es importante mencionar que los datos que se envían desde el microcontrolador son del tipo “Single” y los procesados por Matlab son del tipo “double” por lo que es necesario aplicar una conversión a través de un “data type conversion”, en la Figura 17 se muestra la confirmación tras establecer el emparejamiento entre el PC y la ESP32. Así mismo en la figura 18 se muestran los datos que son recibidos por parte de la placa y son graficados en el computador.

Figura 17.

Emparejamiento ESP32 Bluetooth

Figura 18.

Datos recibidos vía bluetooth



Una vez se establece una conexión de comunicación a través de bluetooth, se verifica su radio de alcance, obteniendo una comunicación estable hasta a una distancia de cinco metros, pero el envío de datos presenta alta latencia, lo cual no favorece al desarrollo del proyecto, y motivo por el cual se descarta el uso del bluetooth como método de comunicación.

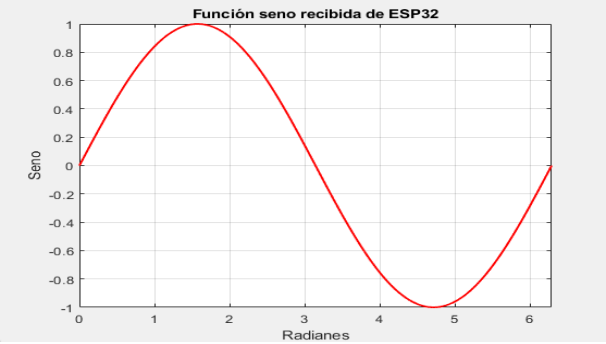
3.3.2 Prueba de Sistema de Comunicación a través de Wifi

Para establecer la comunicación a través de wifi es necesario preparar la placa ESP32 para la recepción y envío de la función Senoidal, la programación de la placa se realiza a través del Arduino IDE. La otra parte de la comunicación se realiza desde el panel principal de Matlab, por lo cual es necesario preparar un script dedicado a envío y recepción de datos.

Lo que se hace en esta etapa es programar el script mencionado anteriormente en la placa y se valida la conexión de la placa a la red de internet, una vez garantizada la conexión se ejecuta el código desde Matlab, en donde se grafica la función que recibe desde la placa ESP WROOM 32, y se muestra en la Figura 19.

Figura 19.

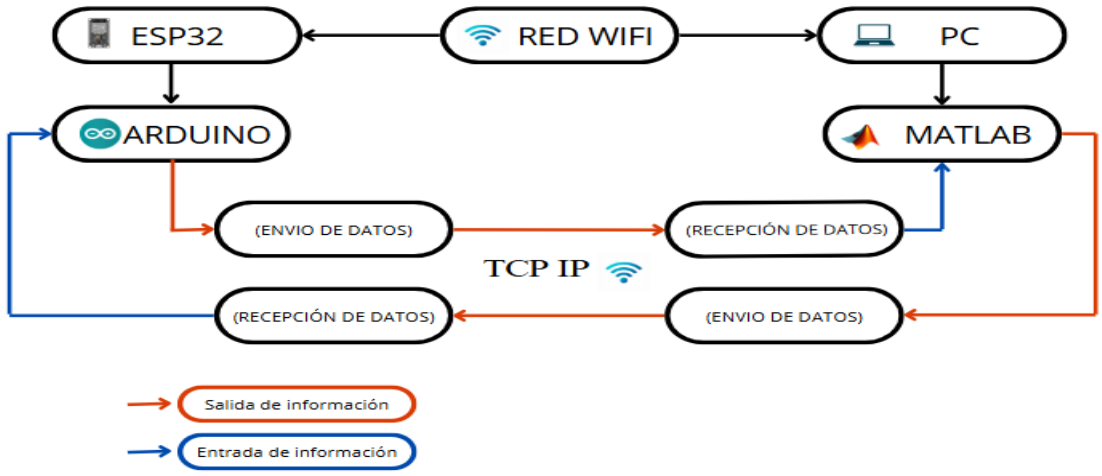
Datos recibidos vía Wifi



3.3.3 Implementación sistema de comunicación con el protocolo TCP_IP

Figura 20.

Lógica de sistema de comunicación- Protocolo TCP_IP



En la Figura 20 se ilustra el diagrama de flujo de la información entre la placa ESP32 y el PC, La ESP32 primero se conecta a una red WiFi utilizando las credenciales configuradas en el código. Una vez establecida la conexión, intenta enlazarse con MATLAB, que actúa como servidor en una IP y puerto específicos. Si la conexión es exitosa, se establece un puente de comunicación

TCP/IP entre ambos. Para el intercambio de información, Arduino (ESP32) y MATLAB se configuran dos vectores de comunicación:

- Vector 1: para recibir datos (desde MATLAB hacia la ESP32, conteniendo los valores de referencia o setpoints).
- Vector 2: para enviar datos (desde la ESP32 hacia MATLAB, con las velocidades reales medidas).

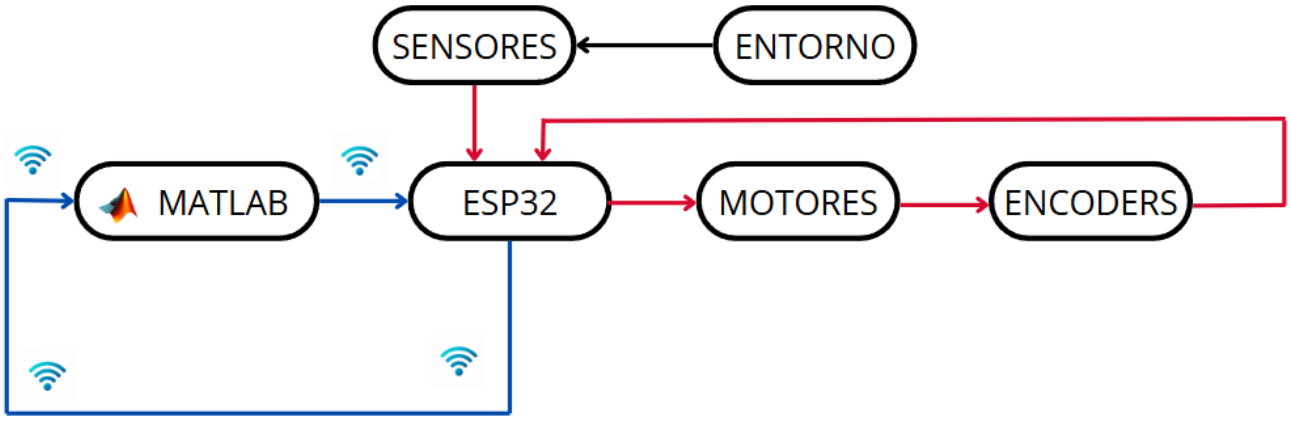
El proceso de comunicación funciona en un bucle continuo: MATLAB envía los valores deseados (velocidades lineal y angular), la ESP32 recibe estos datos, los procesa y ajusta la señal de control de los motores. Luego, la ESP32 calcula las velocidades reales, las envía de vuelta a MATLAB, y este puede analizarlas o corregir los setpoints si es necesario.

3.3.4 Implementación de Sistema de comunicación vía Wifi

Tras realizar las pruebas y descartar el sistema de comunicación por Bluetooth, se procede a crear el protocolo de comunicación vía wifi, para envío y recepción de datos, como se ilustra en la Figura 21.

Figura 21.

Esquema de comunicación entre ESP32 y MATLAB



La propuesta de comunicación que se muestra en la Figura 21, establece los siguientes pasos para los conjuntos de datos:

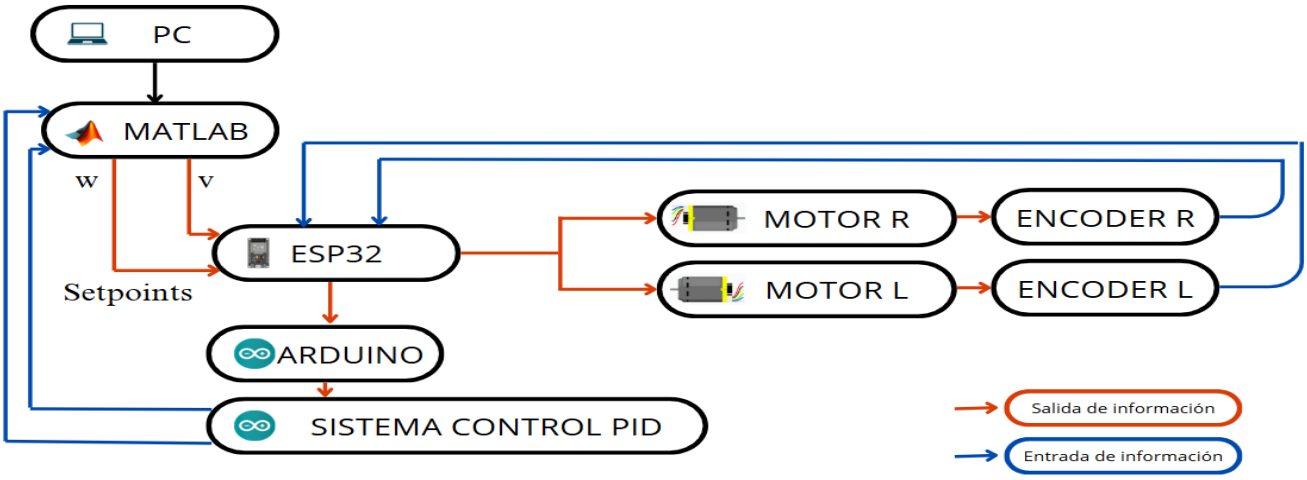
- Paso 1: Matlab envía información a la ESP WROOM 32 vía wifi para controlar la velocidad de los motores.
- Paso 2: Los motores reciben la señal a través del controlador L298N, y la convierten en una velocidad angular.
- Paso 3: La velocidad Angular de los motores es leída por los encoders, y enviada de vuelta a la ESP WROOM 32.
- Paso 4: La placa ESP WROOM 32 recibe de forma simultánea la información de todos los componentes del robot y la envía vía Wifi a MATLAB para realizar la retroalimentación.

3.4 Implementación de sistema PID

A continuación, se describe el procedimiento de la implementación de un controlador PID para el seguimiento de trayectorias en un robot uniciclo, en la Figura 22 se ilustra la lógica de flujo para el controlador.

Figura 22.

Lógica de implementación control PID



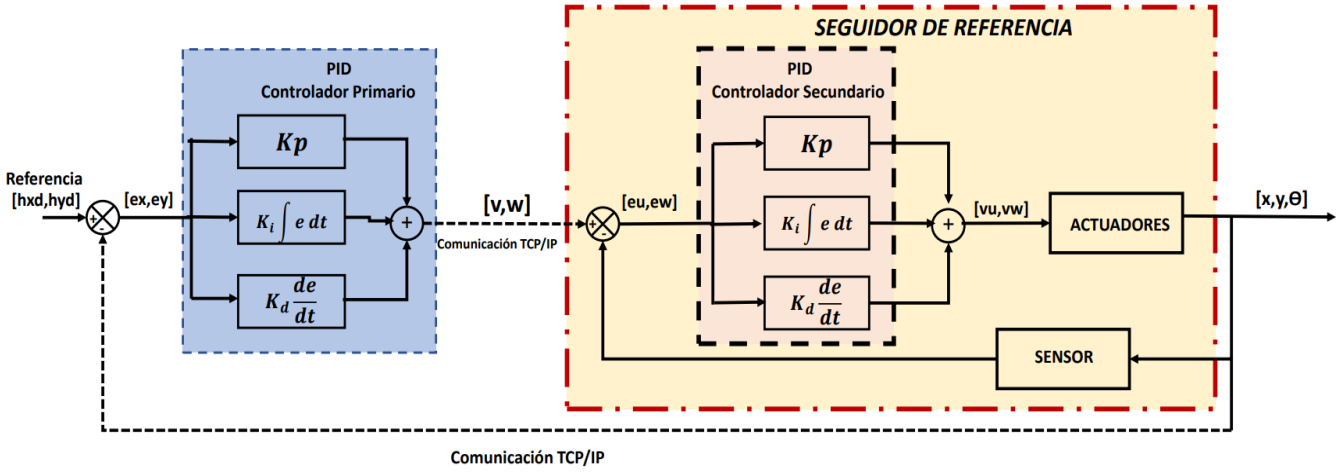
El código implementa un control PID (Proporcional-Integral-Derivativo) para ajustar la velocidad de las ruedas del sistema según los valores de referencia (setpoints) enviados desde MATLAB. Primero, la ESP32 mide la velocidad real de las ruedas utilizando encoders y la compara con la velocidad deseada. A partir de esta diferencia (error), se calcula una señal de control que ajusta la velocidad de los motores. La parte proporcional (P) responde de manera inmediata al error, mientras que la parte integral (I) acumula el error a lo largo del tiempo para corregir desviaciones persistentes. La señal de control resultante se convierte en valores de PWM, que determinan la dirección y potencia de los motores. Para evitar saturaciones, se limita la señal de control dentro de un rango permitido. Finalmente, la ESP32 envía las velocidades reales de vuelta a MATLAB, permitiendo una comunicación en tiempo real para monitoreo y ajuste. Todo este proceso se repite continuamente, garantizando que el sistema siga los valores deseados con precisión

3.4.1 Lógica del bloque de control

En la Figura 23 se expone el bloque de control de forma detallada para la estrategia de control PID.

Figura 23.

Lógica de bloque de control PID



El controlador primario es el encargado de procesar la información que está saliendo de los encoders y es procesada por Matlab para enviar nuevas órdenes a la placa ESP32.

El controlador secundario es el encargado de fidelizar la información que se recibe desde Matlab y entregarla lo más limpia posible a los actuadores.

Es importante mencionar que, para este primer escenario, el controlador primario y el secundario son algoritmos de control PID.

3.5 Implementación de sistema basado en ADRC

A continuación, se describe la implementación de un controlador basado en rechazo activo de perturbaciones para seguimiento de trayectorias en un robot uniciclo, es importante mencionar que el modelo matemático expuesto en el presente proyecto utiliza una adaptación de las ecuaciones (7) y (8) descritas por (Carreño-Zagarra, Moreno, & Guzmán, 2024) en su estudio,

donde se abordó el control de sistemas dinámicos de segundo orden en presencia de incertidumbres y perturbaciones, también mediante un enfoque de ADRC

$$C(s) = Kp(1 + \frac{1}{T_i s} + \frac{T_d s}{N T_d s} \quad (7)$$

Donde Kp, Ti, Td y N son las ganancias proporcionales, el periodo integrativo, el periodo y el filtro derivativos respectivamente.

$$k3 = \frac{1}{NT_D} \; ; \; k2 = k_p\left(1 + \frac{1}{N}\right) \; ; \; k1 = k_p\left(\frac{1}{NT_d} + \frac{1}{T_i}\right) \; ; k0 = k_p\left(\frac{1}{NT_i T_d}\right) \quad (8)$$

3.5.1 Fundamento matemático del controlador.

El código empleado para implementar el controlador define los parámetros de un controlador PID filtrado, que actúa de forma similar a un ADRC (Active Disturbance Rejection Control), y donde se consideran los siguientes parámetros:

- **Kp**: Ganancia proporcional.
- **Ti**: Tiempo integral.
- **Ta**: Tiempo derivativo.
- **N**: Factor de filtrado derivativo.

Este factor **N** se utiliza para suavizar la acción derivativa dentro del controlador PID, evitando amplificación del ruido en la derivada. Su valor no se calcula directamente, sino que se elige según las características del sistema, para el caso de un seguidor de trayectorias se estima realizando una sintonización a partir de pruebas en un ambiente controlado, hay que tener varias consideraciones con este factor, por ejemplo si **N** es pequeño el filtro tiene un efecto más fuerte y la respuesta del controlador será más lenta, por el contrario Si **N** es grande la respuesta es más

rápida, pero puede amplificar el ruido en la medición. A través de diferentes pruebas e iteraciones se encontró que el óptimo de **N** para la implementación del controlador fue de 30.

A partir de los anteriores parámetros y con un PID clásico, se calculan los coeficientes del PID modificado dando los siguientes factores:

$$K_0 = Kp \quad K_1 = \frac{1}{NTa} \quad K_2 = Kp\left(\frac{Ta}{Ti + Ta}\right) \quad K_3 = \left(\frac{Ti}{Ti + Ta}\right) \tag{9}$$

K_0 representa la ganancia proporcional pura del controlador, **k_1** es la que controla la acción integral, que compensa errores sostenidos y actúa como un observador de estado, similar a un Luenberger Observer.

K2 Es el término derivativo ajustado según la relación entre $\boldsymbol{Ti}$ y $\boldsymbol{Td}$ el cual reduce el efecto de los cambios rápidos en el error

K3 Ajusta la rapidez con la que responde el término derivativo filtrando ruido es decir ayuda a estimar perturbaciones, una característica clave de ADRC. Aplicando esto a ley de control para la velocidad lineal y angular es:

$$v = k_0 e_x + k_1 \int e_x dt + k_2 \frac{de_x}{dt} + k_3 \hat{dx} \tag{10}$$

Se ajusta ecuación control de velocidad lineal según (Carreño-Zagarra, Moreno, & Guzmán, 2024)

$$\omega = k_0 e_y + k_1 \int e_y dt + k_2 \frac{de_y}{dt} + k_3 \hat{dy} \tag{11}$$

Donde ***ey*** es el error en la velocidad lineal para la primera ecuación y $e_{y\,es}$ el error angular para la segunda ecuación, además $\hat{dy}, \hat{dx}$ ayudan a evitar fluctuaciones bruscas, y se aplica un suavizado con un filtro de primer orden.

En términos generales este filtro es el encargado de suavizar y evitar fluctuaciones bruscas en los datos.

$$v = \alpha v_{prev} + (1 - \alpha)v \qquad (12)$$

$$\omega = \alpha \omega_{prev} + (1 - \alpha)\omega \qquad (13)$$

En las Ecuaciones 12 y 13, $\boldsymbol{\alpha}$ es el filtro que se aplica a las señales de velocidad angular y lineal obtenidas con el controlador. Este filtro ayuda a evitar cambios bruscos en los datos de velocidad comparándolos con valores previos. Es decir, $\boldsymbol{v}$**prev** y $\boldsymbol{w}$**prev** representan las velocidades en una iteración anterior. El filtro toma valores entre 0 y 1: un valor de 0 significa que no se ha aplicado filtrado, mientras que un valor de 1 implica un filtrado fuerte donde solo se considera la iteración anterior. Se eligen valores intermedios para lograr un equilibrio entre estabilidad y respuesta rápida, permitiendo suavizar la señal sin retrasarla excesivamente.

$$\begin{pmatrix} e_x^{\text{global}} \\ e_y^{global} \end{pmatrix} = \begin{pmatrix} hxd - x \\ hyd - y \end{pmatrix} \qquad (14)$$

A continuación, se presentan las ecuaciones de error corregido para trayectorias circulares.

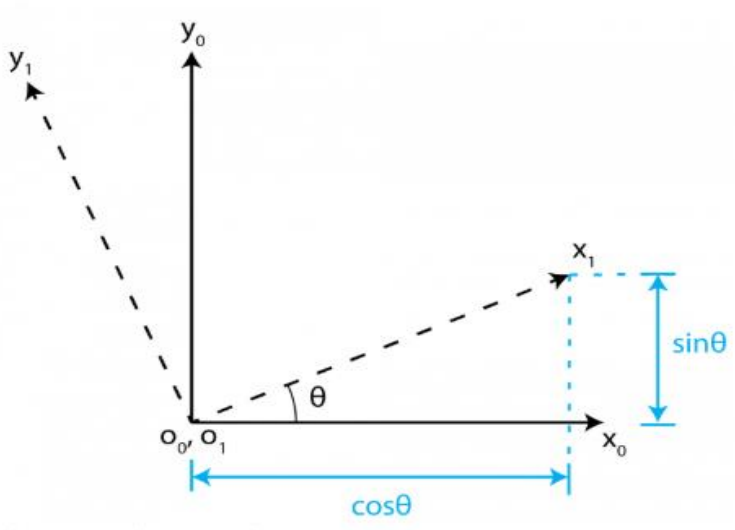
$$R = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \qquad (15)$$

$$\begin{pmatrix} e_x^{\text{local}} \\ e_y^{\text{local}} \end{pmatrix} = R \cdot \begin{pmatrix} hxd - x \\ hyd - y \end{pmatrix} \qquad (16)$$

Las ecuaciones (15) y (16) se utilizan en la cinemática de robots y permiten describir cómo un objeto se desplaza y gira en el espacio sin cambiar su tamaño ni forma. A diferencia de las coordenadas globales ($\boldsymbol{x}$), que solo indican un punto en el espacio sin considerar la orientación del robot, la transformación con la matriz de rotación **R** tiene en cuenta tanto la traslación como la rotación del robot. Este método permite expresar el error de posición en un sistema de referencia local alineado con el robot un ejemplo de esto es la siguiente figura, La imagen muestra la rotación de un sistema de coordenadas en un plano en un ángulo $\boldsymbol{\theta}$. Inicialmente, el sistema de referencia está alineado con los ejes $\boldsymbol{x}$ y $\boldsymbol{y0}$, pero al rotarlo en sentido antihorario en $\boldsymbol{\theta}$ grados, los nuevos ejes $\boldsymbol{x1}$ y $\boldsymbol{y1}$ toman una nueva orientación. La figura destaca cómo un punto en el eje $\boldsymbol{x0}$ se desplaza a una nueva posición en el eje $\boldsymbol{x1}$, con componentes de desplazamiento **cosθ** en la dirección horizontal y **sin($\boldsymbol{\theta}$)** en la dirección vertical. Esta transformación es clave en la cinemática de robots y en la manipulación de coordenadas en espacios rotacionales. En la figura 24 se puede apreciar cómo se interpretan las variaciones de posición (x1, y1) como una variación angular de (x0,y0)

Figura 24.

Rotación plana de 90°

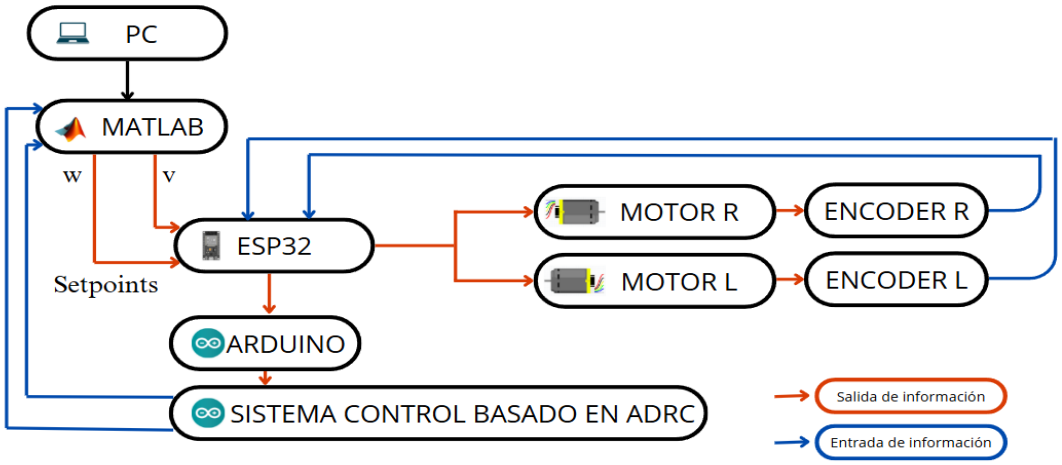


3.5.2 Lógica de implementación

La Figura 25 representa de forma simplificada la estructura de la comunicación entre los diferentes componentes para la puesta en marcha del sistema de control, en el robot móvil.

Figura 25.

Lógica de implementación control basado en ADRC



El movimiento del robot uniciclo está basado en el modelo cinemático que se expone en la sección 3.1.2 y las ecuaciones de la (1) a la (6) para seguimiento de trayectorias, adicionalmente se calculan constantemente los errores de posición y orientación, esto con animo de definir un sistema de control PID, donde el controlador de velocidad lineal ajusta la distancia a la trayectoria, y el controlador de velocidad angular corrige la orientación para garantizar que el robot se alinee de forma apropiada.

Buscando una mayor estabilidad, la velocidad lineal pasa a segundo plano cuando el error angular es grande, priorizando la corrección de la orientación antes de avanzar. Ambos controladores aplican saturaciones para evitar valores extremos que podrían generar movimientos

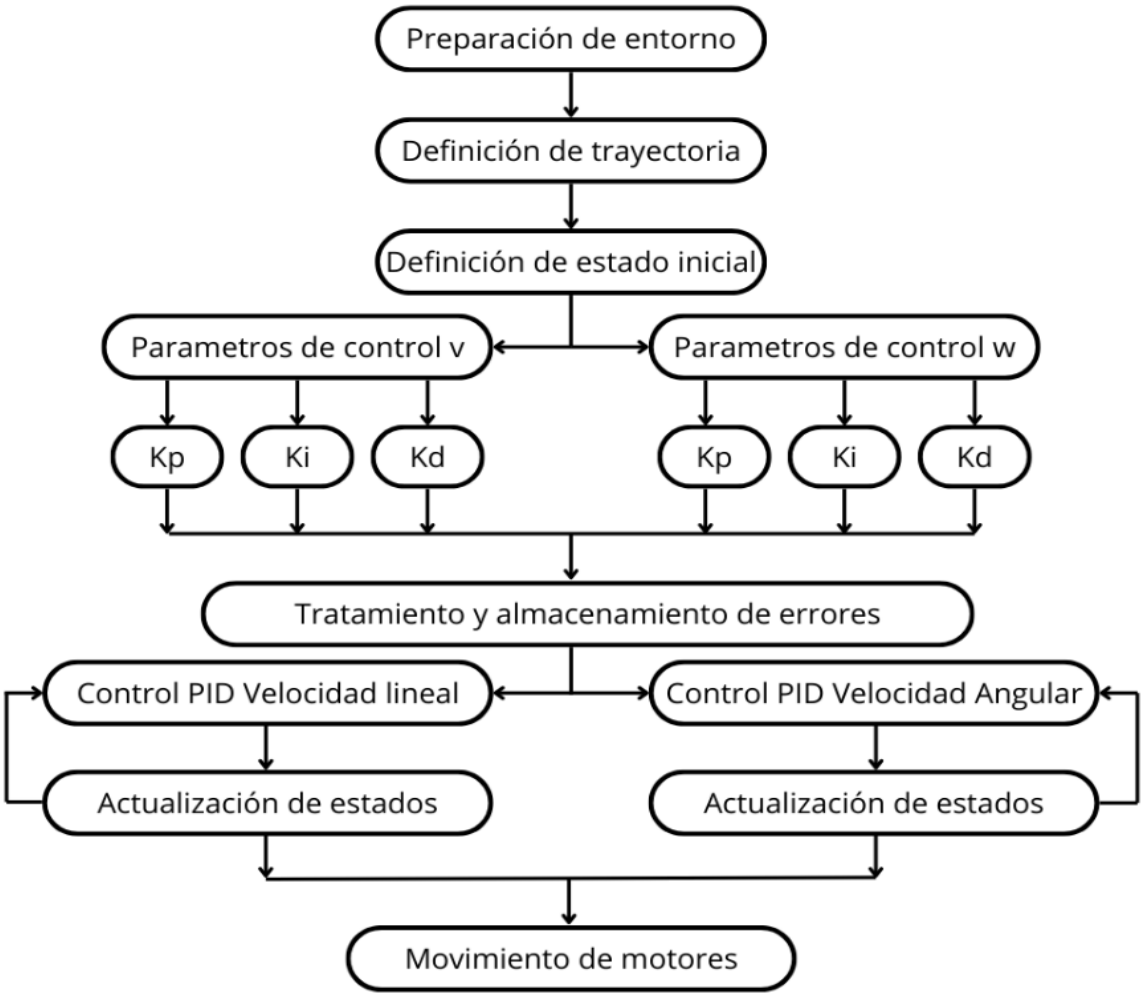
inestables. Finalmente, el estado del robot se actualiza en cada iteración en función de los valores calculados, lo que garantiza un seguimiento preciso de la trayectoria.

3.5.3 Lógica de flujo de información para control basado en ADRC

La Figura 26 Muestra el flujo de los datos cuando se pone en marcha el sistema de control basado en ADRC.

Figura 26.

Flujo de datos para sistema de control basado en ADRC



A continuación, se describen los diferentes pasos que son mencionados en el flujograma de datos.

- Preparación del entorno: Se limpia el entorno de MATLAB eliminando variables, cerrando figuras abiertas y desactivando advertencias. Esto garantiza que la simulación se ejecute sin conflictos de datos previos ni interrupciones por mensajes innecesarios.
- Definición de trayectoria: Se establece la trayectoria que el robot debe seguir. Para efectos del proyecto se manejan las trayectorias en formato parametrizado como se describen en el capitulo 4.
- Estado Inicial del Robot: Se fijan las condiciones iniciales del robot, ubicándolo en un punto específico del espacio y orientado en cierto Angulo con respecto al eje x.
- Parámetros del Control PID: Se definen las ganancias del controlador PID para la velocidad lineal y angular. Estas constantes determinarán la respuesta del sistema al error de posición y orientación, afectando la estabilidad y rapidez con la que el robot corrige su trayectoria.
- Variables Auxiliares para el Control PID: Se inicializan variables para almacenar el error acumulado y el error previo, necesarias para calcular los términos integral y derivativo del control PID. Estas variables se actualizarán en cada iteración de la simulación.
- Almacenamiento de Resultados: Se crean vectores vacíos para registrar la evolución de la posición, orientación, velocidades y errores a lo largo de la simulación. Estos datos serán utilizados posteriormente para análisis y visualización.

- Cálculo del Error en Coordenadas Globales: Se determina la diferencia entre la posición deseada y la posición actual del robot en los ejes (x,y).
- Transformación a Coordenadas Locales: El error se convierte a un sistema de referencia relativo al robot, separando el error en una componente de distancia frontal y una componente angular. La orientación del error se normaliza para evitar discontinuidades.
- Control PID de la Velocidad Lineal: Se aplica el control PID para determinar la velocidad lineal del robot, considerando la magnitud del error, su acumulación a lo largo del tiempo y su variación en el último instante. Se penaliza la velocidad si la orientación del robot es incorrecta y se limita su valor máximo para evitar movimientos bruscos.
- Control PID de la Velocidad Angular: Se aplica otro control PID para ajustar la velocidad angular, asegurando que el robot gire en la dirección correcta para minimizar el error de orientación. También se restringe el valor máximo de esta velocidad para evitar oscilaciones excesivas.

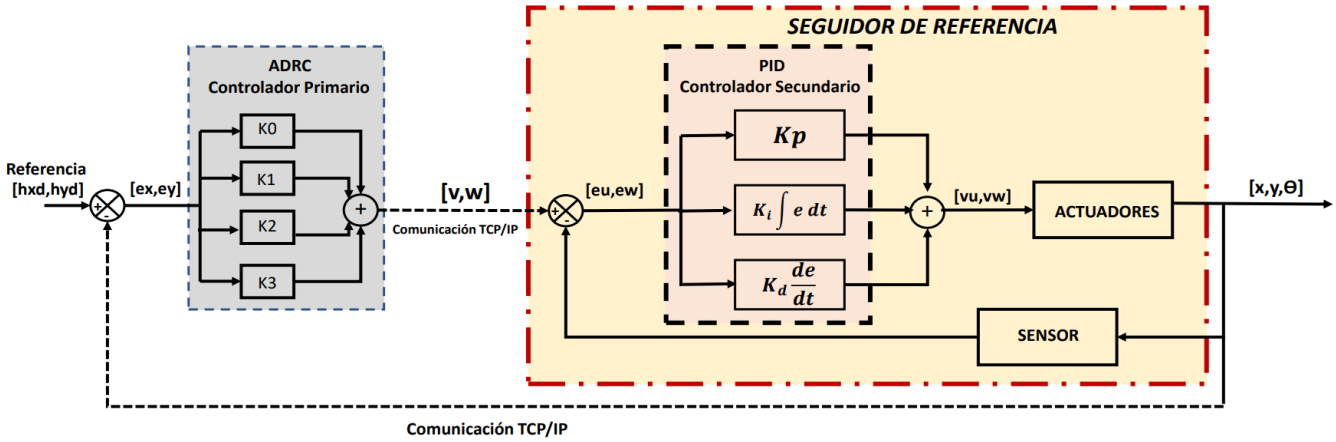
•Actualización del Estado del Robot: Usando el modelo cinemático del robot diferencial, se calcula la nueva posición y orientación a partir de las velocidades lineal y angulares determinadas por los controladores.

3.5.4 Lógica del bloque de control

En la Figura 27 se presenta el bloque de control de forma detallada para la estrategia de control basado en ADRC.

Figura 27.

Lógica de bloque de control basado en ADRC



El controlador primario es el encargado de procesar la información que está saliendo de los encoders y es procesada por Matlab para enviar nuevas órdenes a la placa ESP32. La función de este controlador es principalmente recibir las señales de acción por parte del robot y generar las nuevas referencias de velocidad angular y velocidad lineal del robot y enviar las al controlador secundario.

El controlador secundario es el encargado de fidelizar la información que se recibe desde Matlab y entregarla de forma apropiada a los actuadores. A través del uso de este controlador de tipo seguidor de referencia se garantiza la calidad de los datos y señales que entran a los actuadores y permiten el movimiento del robot.

El controlador primario que es objetivo principal del presente proyecto está basado en una estrategia de control mediante rechazo activo de perturbaciones, a través de un filtro derivativo en un PID, por otra parte, el controlador secundario trabaja con un algoritmo PID tradicional, y en conjunto trabajan para que el robot siga trayectorias de forma óptima.

4. Evaluación del modelo de control

En el presente capitulo se aborda la evaluación del sistema de control basado en rechazo activo de perturbaciones, para el fin de este proyecto, este algoritmo se nombrará por sus siglas como: **SCB-ADRC**.

Para la verificación se seleccionaron tres curvas de referencia, las cuales son el objetivo a seguir por el robot, y se presentan de forma parametrizada bajo la siguiente forma:

- Función senoidal

$$x(t) = A\cos(Bt)\,;\; x(t) = 0.1(t) \qquad (17)$$

$$y(t) = D\sin(Bt)\,;\; y(t) = 0.4\,sin\,(0.3\,t) \qquad (18)$$

- Función lineal

$$x(t) = At + B;\; x(t) = 0.1(t) \qquad (19)$$

$$y(t) = Ct + D;\; y(t) = 0.00001(\,t) + 0.5 \qquad (20)$$

- Función circular

$$x(t) = A\cos(t + \emptyset)\;;\; x(t) = 1\cos(0.15\,t) \qquad (21)$$

$$y(t) = D\sin(t + \emptyset)\;;\; y(t) = 1\sin(0.15\,t) \qquad (22)$$

Para realizar la sintonización de los sistemas se utilizaron las metodologías previamente descritas y las simulaciones en las cuales podemos ver una representación gráfica del comportamiento del robot móvil frente a estas trayectorias, utilizando la metodología de sintonización experimental basada en Ziegler–Nichols, el cual explica el orden que se debe seguir para obtener una mejora en la sintonización de los modelos PID Utilizando el siguiente enfoque:

- Paso 1: Sintonización de la acción proporcional: Se empieza con un valor de cero en las acciones integral (I) y derivativa (D), para aumentar progresivamente la ganancia proporcional hasta que se logra una respuesta aceptable (o se alcanza el umbral de oscilación en algunos métodos, como el de Ziegler–Nichols).
- Paso 2: Incorporación de la acción integral: Una vez ajustado P para conseguir una respuesta rápida, se añade la acción integral para eliminar el error en estado estacionario.
- Paso 3: Ajuste de la acción derivativa: Finalmente se añade (o se ajusta) la acción derivativa para mejorar la respuesta transitoria y reducir el sobre impulso.

Se implementan estos pasos en cada una de las simulaciones de seguimiento de trayectorias, las simulaciones obtenidas para cada una de las funciones de referencia generaron un valor especifico de ganancia en cada uno de los controladores para cada tipo de trayectoria. Posteriormente se muestran las gráficas obtenidas en las pruebas con el robot móvil diferencial, donde se evidencian diferencias en las ganancias simuladas con las obtenidas en la implementación del modelo físico, ya que en el entorno simulado se asume que el seguimiento de la velocidad angular y lineal es ideal, a diferencia de la implementación del modelo real que presenta latencias

de comunicación y perturbaciones externas como lo es el mismo peso de la estructura del robot y el terreno.

El tiempo de reacción del PID del robot móvil, genera perturbaciones dentro del sistema global de Matlab. En la fase de implementación de los modelos para el seguimiento de trayectorias se aplica el siguiente protocolo de pruebas, el cual muestra el paso a paso de las acciones a tener en cuenta antes de realizar una simulación o una aplicación práctica, que facilite el correcto seguimiento de la referencia.

4.1 Simulación de algoritmos de control

A continuación, se describen los resultados de la puesta en marcha de la simulación de los sistemas de control anteriormente mencionados.

Las unidades en las que trabaja el modelo son m, m/s, rad y rad/s.

- **Verificar Estado inicial del robot**: En este punto se deben especificar las coordenadas iniciales del robot y su orientación en el espacio en unidades de radianes
- **Elección de trayectoria a seguir:** Se debe elegir la función que describa el recorrido a seguir, dándole así una referencia al controlador
- **Verificar el espacio de simulación:** Dependiendo de la función a implementar se deben corregir los rangos de trabajo para poder visualizar los datos obtenidos.
- **Sintonización de ganancias (PID – ADRC):** En este punto se deben sintonizar las ganancias como indica la metodología de Ziegler Nicols para el modelo PID, en el caso del modelo ADRC basado en PID se recomienda iniciar las simulaciones con valores Kp bajos, luego se puede aplicar la misma metodología de aumentar la ganancia proporcional hasta que el sistema empiece a oscilar luego se deben incluir

los valores para el tiempo integral como el diferencial los cuales intervienen en las constantes **K1, K2 y K3** , posteriormente se pueden suavizar las salidas y estimar algunas perturbaciones del controlador con las variables alpha y beta respectivamente

- **Revisión de los datos obtenidos:** En el monitor serial de Matlab(ventana de comandos) podemos ver la actualización de los estados de cada una de las variables del sistema, lo cual nos ayudara a la hora de sintonizar los controladores PID-ADRC.

4.1.1 Simulación de control PID vs SCB-ADRC – Trayectoria lineal

Los parámetros que se tienen en cuenta para la simulación de seguimiento de una trayectoria lineal son los siguientes:

Tabla 2.

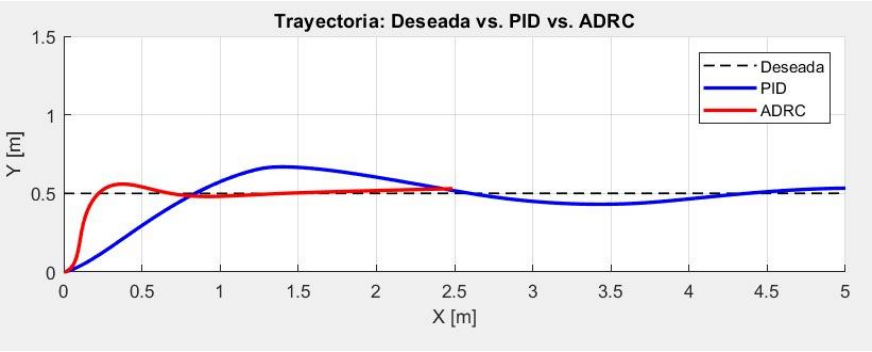
Parámetros Simulación trayectoria lineal PID – SCB ADRC

	Velocidad Lineal			Velocidad Angular		
	Kp	**Ki**	**Kd**	**Kp**	**Ki**	**Kd**
PID	0,08	0,3	0,1	0,32	0	0,9

	Kp	**Ti**	**Ta**	**N**	**α**	**β**
ADRC	0,02	0,01	0,0051	4,5	0,3	0,93

Figura 28.

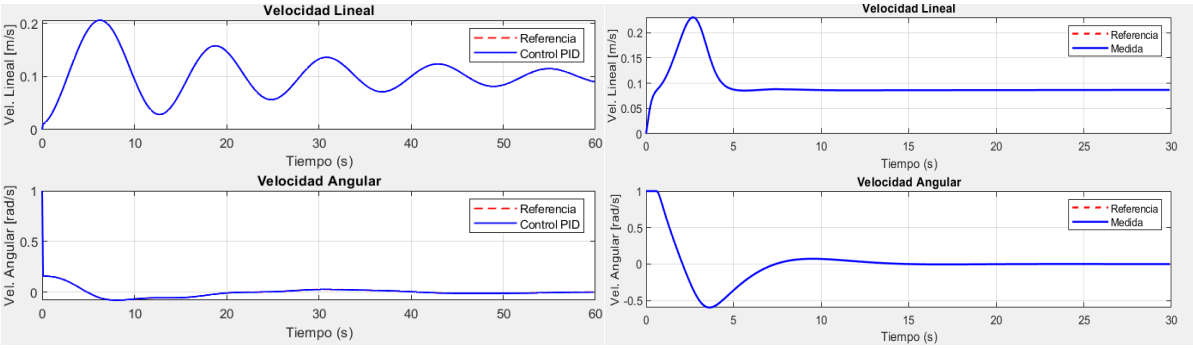
Seguimiento de trayectoria lineal PID vs SCB-ADRC



Cómo se puede evidenciar en la figura 28, hay un comportamiento más eficaz al momento de encontrar la trayectoria cuando se emplea el sistema de control basado en rechazo activo de perturbaciones, la curva de búsqueda encuentra la referencia más rápido, adicionalmente se evidencian picos y valles de magnitud menor en comparación con el PID, para seguimiento de trayectorias lineales se obtienen resultados que confirman que el sistema de control basado en rechazo activo de perturbaciones es más adecuado que el PID.

Figura 29.

Comportamiento de velocidades trayectoria lineal PID vs SCB-ADRC

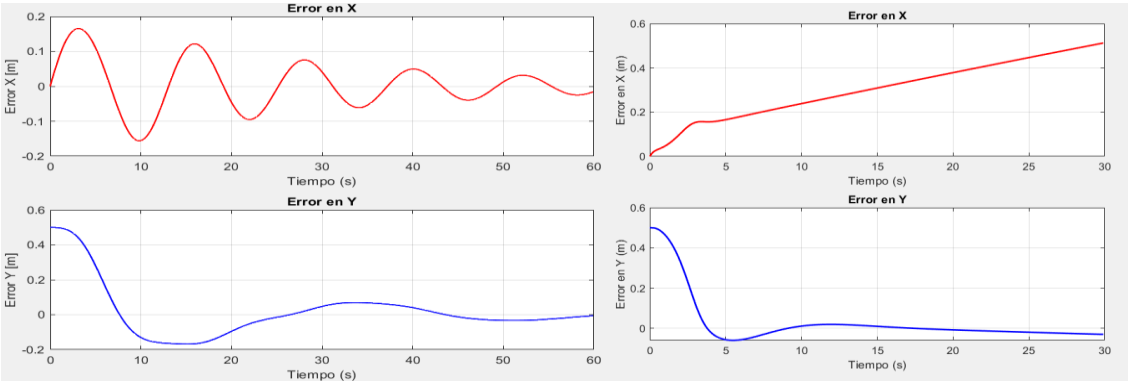


Cómo se observa en la figura 29, existe un comportamiento con un sobre pico al iniciar el movimiento para el sistema de control basado en rechazo activo de perturbaciones, que luego se estabiliza en un valor contante, tanto para velocidad angular, como lineal.

Para el caso del sistema de control PID existe un comportamiento oscilatorio para la velocidad lineal y un comportamiento estable en velocidad angular a bajas revoluciones.

Figura 30.

Comportamiento de errores trayectoria lineal PID vs SCB-ADRC



En cuanto a los errores expuestos en la figura 30, se puede apreciar una caída rápida en y para ambos sistemas, por otra parte, hay una estabilización lineal en x para el modelo de control basado en rechazo activo de perturbaciones.

4.1.2 Simulación de control PID vs SCB-ADRC – Trayectoria circular

Los parámetros que se tienen en cuenta para la simulación de seguimiento de una trayectoria circular son los siguientes:

Tabla 3.

Parámetros Simulación trayectoria circular PID - SCB ADRC

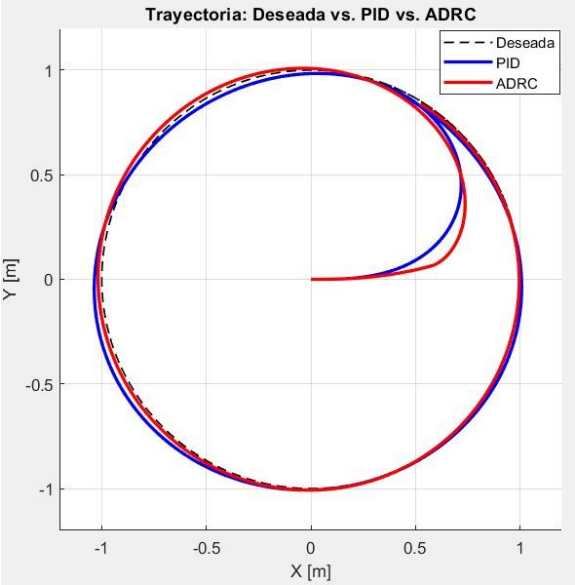
	Velocidad Lineal			Velocidad Angular		
	Kp	**Ki**	**Kd**	**Kp**	**Ki**	**Kd**
PID	0,17	0	0	0,14	0,01	0

	Kp	**Ti**	**Ta**	**N**	**α**	**β**
ADRC	0,155	0,6	0,001	31	0,2	0,95

Es importante mencionar que para la trayectoria circular, se utilizó el error local, descrito en la sección 3.5.1, debido a la naturaleza del desplazamiento en el plano (x,y)

Figura 31.

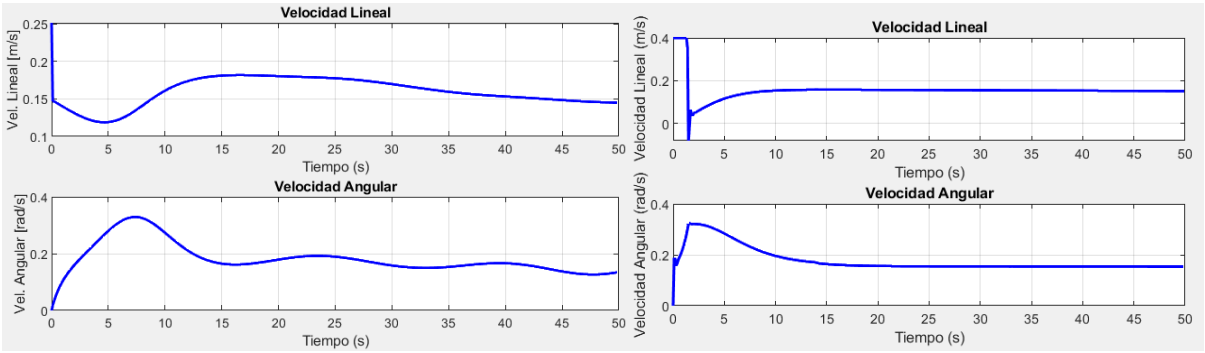
Seguimiento de trayectoria circular PID vs SCB-ADRC



Cómo se puede ver en la figura 31, el seguimiento de trayectoria circular es bastante similar para ambas estrategias de control, presentando en el PID un desfase despreciable, y siendo superior y más exacto en el modelo de control basado en ADRC.

Figura 32.

Comportamiento de velocidades trayectoria circular PID vs SCB-ADRC

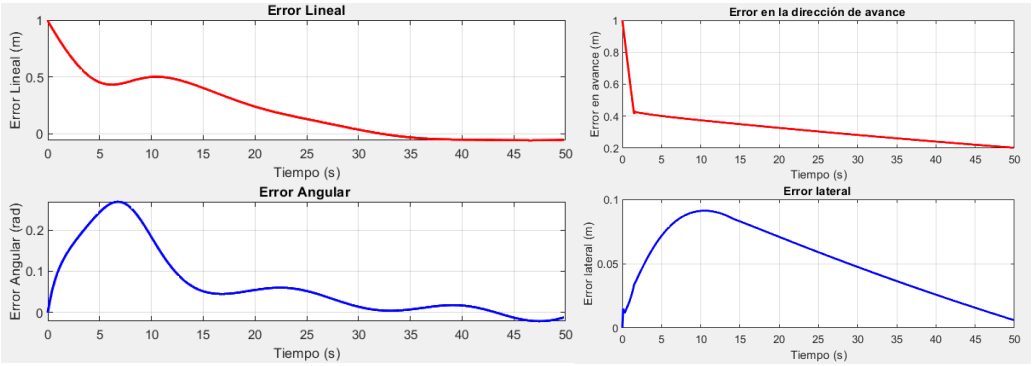


En la gráfica 32 se muestra el comportamiento de las velocidades angulares y lineales para ambas estrategias de control, es posible notar que a pesar del sobre pico inicial, el modelo de

control basado en rechazo activo de perturbaciones tiene un comportamiento más estable en el tiempo.

Figura 33.

Comportamiento de errores trayectoria circular PID vs SCB-ADRC



En la Figura 33 se ilustra el comportamiento de errores para las dos estrategias de control, como se puede observar en la imagen el error lineal para ambos ejes presenta mayor fluctuación cuando se emplea un PID clásico.

4.1.3 Simulación de control PID vs SCB-ADRC – Trayectoria senoidal

Los parámetros que se tienen en cuenta para la simulación de seguimiento de una trayectoria circular son los siguientes:

Tabla 4.

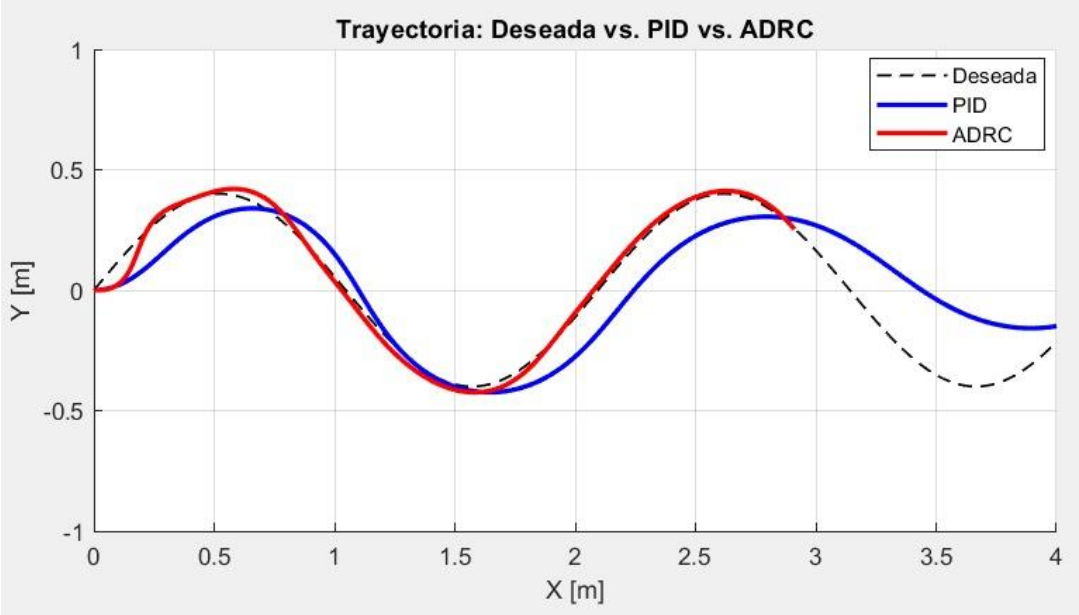
Parámetros Simulación trayectoria senoidal PID – SCB ADRC

	Velocidad Lineal			Velocidad Angular		
	Kp	**Ki**	**Kd**	**Kp**	**Ki**	**Kd**
PID	0,1	0,001	0	0,33	0	0,02

	Kp	**Ti**	**Ta**	**N**	**α**	**β**
ADRC	0,25	0,008	0,01	0,9	0,35	0,86

Figura 34.

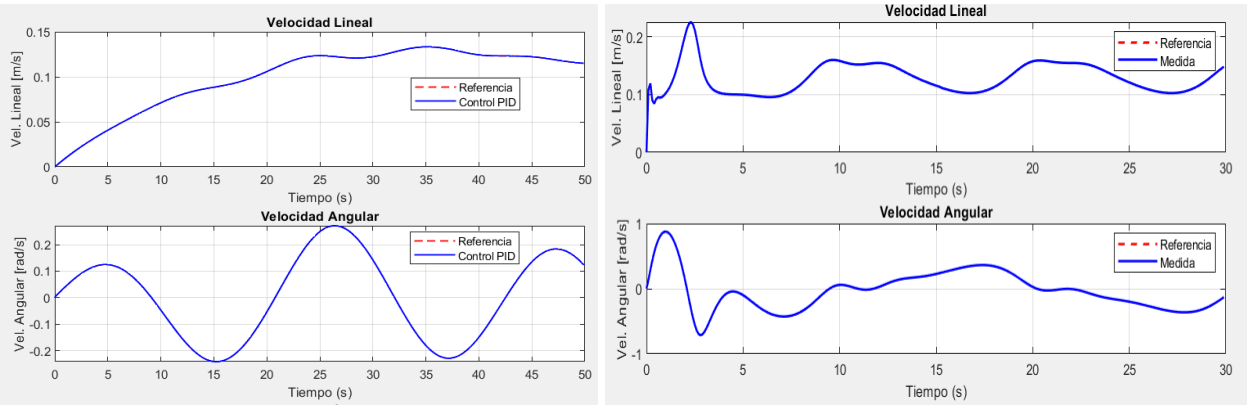
Seguimiento de trayectoria senoidal PID vs SCB-ADRC



En la figura 34 es posible visualizar la comparación entre el seguimiento de trayectorias senoidales, para el caso del PID se aprecia un desfase significativo asociado a los cambios repentinos en las magnitudes de velocidad de los motores, por otra parte, aunque existe un desfase en el modelo de control basado en rechazo activo de perturbaciones, es posible visualizar que el sistema es más tolerante a los cambios y hace un seguimiento más eficiente de la trayectoria.

Figura 35.

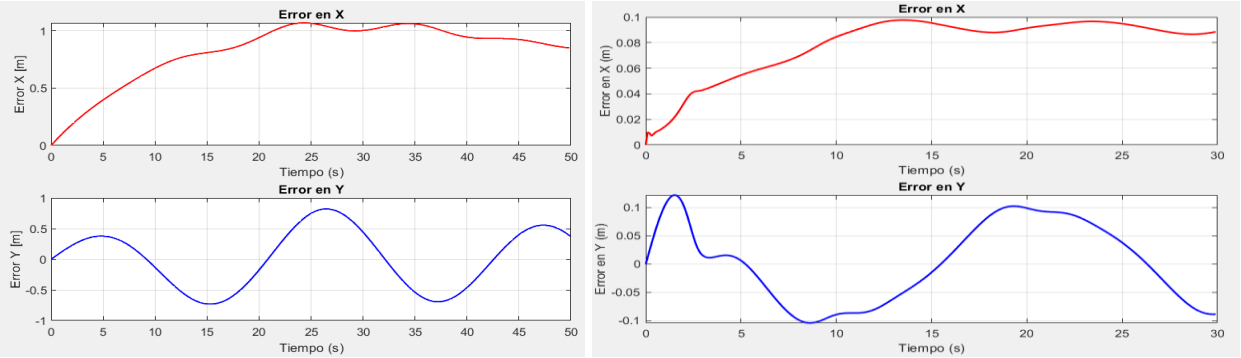
Comportamiento de velocidades trayectoria senoidal PID vs SCB-ADRC



En la figura 35 se refiere el comportamiento de las velocidades angulares y lineales para ambos sistemas de control, donde es posible ver que existen fluctuaciones significativas en ambos escenarios, esto por la naturaleza del movimiento, hay una intento de estabilización y menor fluctuación en el sistema de control basado en ADRC.

Figura 36.

Comportamiento de errores trayectoria senoidal PID vs SCB-ADRC



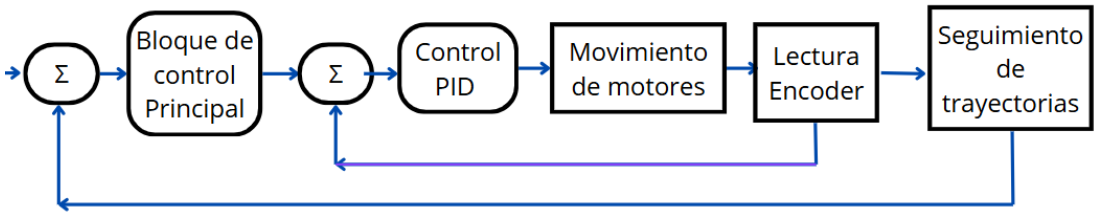
Para el caso de los errores, en ambos sistemas de control hay un comportamiento similar de forma, pero la magnitud de los errores en el modelo de control basado en ADRC, son significativamente menores.

4.2 Implementación de los algoritmos de control en el robot uniciclo.

Para la implementación del sistema de control en el robot se hace necesario el uso de un segundo algoritmo basado en PID para garantizar la estabilidad de las señales que recibe la ESP32 desde Matlab. Así pues, se cuenta con dos controles, uno dedicado a seguimiento de trayectorias basado en ADRC, y otro dedicado a recepción de señales, basado en PID, como muestra en la Figura 37.

Figura 37.

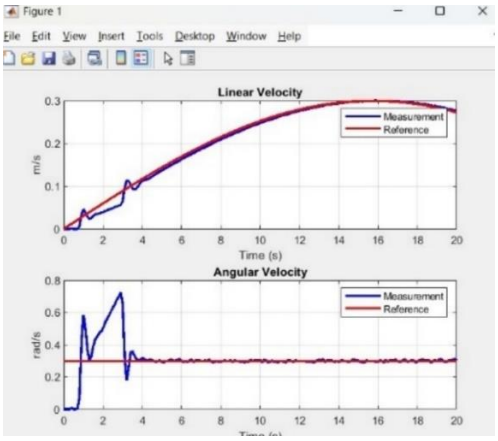
Simplificación de diagrama de flujo de información



A continuación, se muestra la eficiencia del sistema de control propuesto para recepción de datos, en la Figura 38 es posible apreciar que existen sobrepicos, causados por la inercia de los motores, una vez se consigue una velocidad estable, se hace un seguimiento de la referencia bastante preciso.

Figura 38.

Simplificación de diagrama de flujo de información



Para poner en marcha el modelo de control en el robot se muestran los pasos a tener en cuenta y los parámetros de trabajo que mejor optimizan el proceso.

- **Verificar Accesibilidad a la red wifi:** En este paso se debe tener una conexión Wifi que permita conectar el robot diferencial y el computador donde se están realizando las simulaciones.

- **Verificar Estado inicial del robot:** En esta sección se busca revisar el robot y su fuente de alimentación, verificar el nivel de la batería y el estado de las ruedas, así como definir cuál será la orientación que va a tener al comenzar las pruebas. (En todos los casos el robot estará orientado hacia el eje x de referencia)

Verificar la Conexión wifi: En este punto se debe buscar la IP dinámica del computador una vez conectado a la red wifi (*Para este paso se debe acceder al símbolo del sistema luego→escribir el comando "**ipconfig**" el cual desplegara una lista de resultados en la cual se debe buscar el ítem→ "**IPv4**" el cual mostrara la IP dinámica actual del PC*), posteriormente actualizarla en el código de Arduino que se cargara en el robot diferencial, el cual se conectara automáticamente a la misma red wifi y buscara la IP del dispositivo servidor (computador) , una vez conectado se encenderá un led de color azul en el módulo ESP32, al mismo tiempo se mostrara un mensaje en la ventana principal de Matlab verificando la conexión y visualizando los datos que se envían y los que se reciben del robot.

- **Elección de trayectoria a seguir:** Se debe elegir la función que describa el recorrido a seguir, dándole así una referencia al controlador
- **Verificar el espacio de simulación:** Dependiendo de la función a implementar se deben corregir los rangos de trabajo para poder visualizar los datos obtenidos.
- **Sintonización de ganancias (PID – ADRC):** En este punto se deben sintonizar las ganancias como indica la metodología de Ziegler Nicols para los modelos ADRC y PID como se realizó en el protocolo de pruebas de las simulaciones, cabe aclarar que estos valores tienden a variar un poco con respecto a la simulación ya que las mismos actuadores y sensores tienden a tener pequeñas diferencias en su estructura

lo que genera diferentes niveles de alimentación y por ende generan diferentes velocidades a un mismo valor de PWM. Por esta razón se recomienda tomar como referencia la sintonización obtenida en la simulación y ajustar las ganancias con pequeños cambios en los valores.

- **Revisión de los datos obtenidos:** En el monitor serial de Matlab (ventana de comandos) podemos ver la actualización de los estados de cada una de las variables del sistema, lo cual nos ayudara a la hora de sintonizar los controladores PID – ADRC.

4.2.1 Puesta en marcha de control PID vs SCB-ADRC – Trayectoria lineal

Los parámetros que se tienen en cuenta para la implementación de seguimiento de una trayectoria lineal son los siguientes:

Tabla 5.

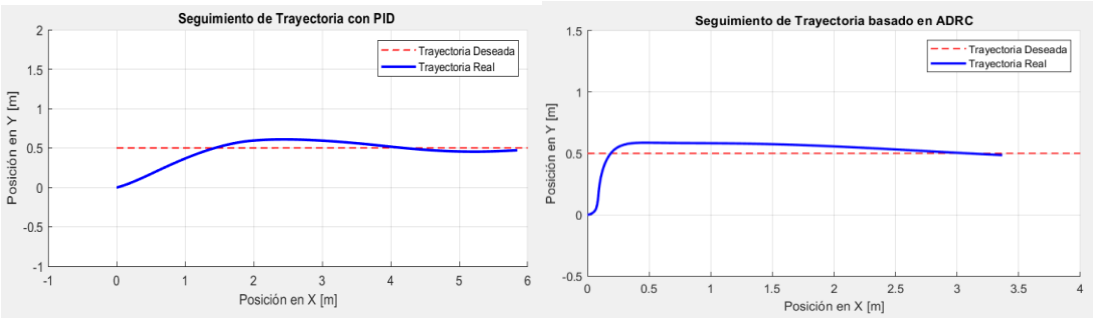
Parámetros Implementación trayectoria lineal PID - ADRC

	Velocidad Lineal			Velocidad Angular		
	Kp	**Ki**	**Kd**	**Kp**	**Ki**	**Kd**
PID	0,17	0,11	0,005	0,14	0,0001	0,85

	Kp	**Ti**	**Ta**	**N**	**α**	**β**
ADRC	0,1	0,0004	0,002	16	0,25	0,87

Figura 39.

Seguimiento de trayectoria lineal PID vs SCB-ADRC

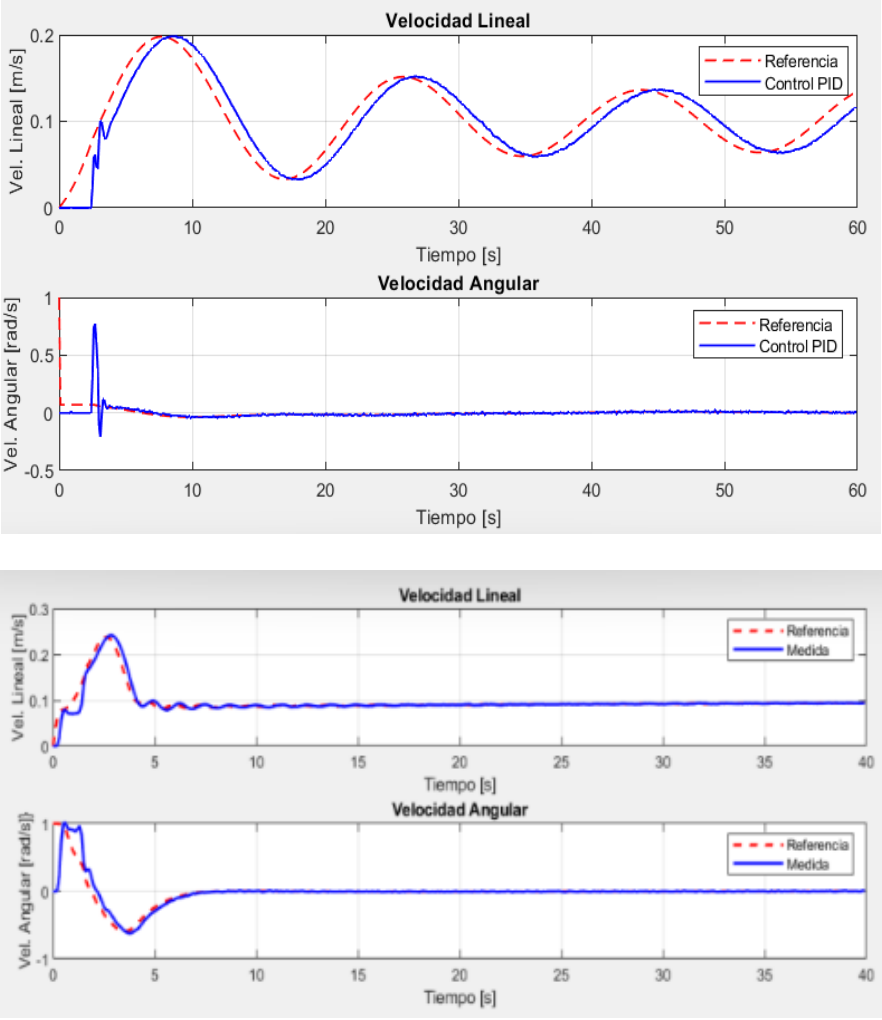


Cómo se aprecia en la figura 39, en el caso de del controlador basado en ADRC, no existe oscilación con respecto a la referencia, sin embargo, hay un desfase en el eje y.

También es posible ver que el acercamiento a la referencia es más rápido en el segundo escenario, dejando ver la superioridad de estabilización del control mediante rechazo activo de perturbaciones.

Figura 40.

Comportamiento de velocidades trayectoria lineal PID vs SCB-ADRC

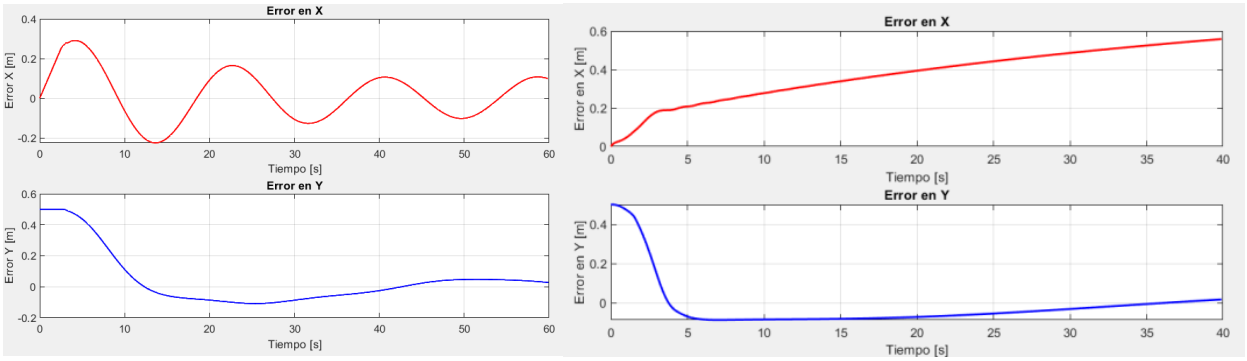


En la figura 40 es posible visualizar el comportamiento de las velocidades lineales y angulares de ambos sistemas de control, es importante resaltar que aunque existen sobre picos al

iniciar el movimiento, sin embargo el sistema de control basado en ADRC muestra una señal más estable.

Figura 41.

Comportamiento de errores trayectoria lineal PID vs SCB-ADRC



En cuanto a los errores mostrados en la figura 41, se ve fluctuación en el modelo de control basado en PID, y una señal más estable y menor para el caso del controlador basado en ADRC.

Una señal de error estable permite realizar correcciones en la trayectoria de forma más eficiente.

4.2.2 Puesta en marcha de control PID vs SCB-ADRC – Trayectoria circular

Los parámetros que se tienen en cuenta para la implementación de seguimiento de una trayectoria lineal son los siguientes:

Tabla 6.

Parámetros Implementación trayectoria circular PID – SCB ADRC

	Velocidad Lineal			Velocidad Angular		
	Kp	Ki	Kd	Kp	Ki	Kd
PID	0,09	0	0,00035	0,36	0,00008	0

	Kp	Ti	Ta	N	α	β
ADRC	0,12	0,0015	0,002	24	0,141	0,8

En la figura 42 es posible visualizar un comportamiento similar en ambos modelos de control, presentando desfase insignificante en ambos escenarios. Mostrando así que el PID normal puede tener comportamiento similar al modelo de control basado en ADRC en algunos escenarios.

Figura 42.

Seguimiento de trayectoria circular PID vs SCB-ADRC

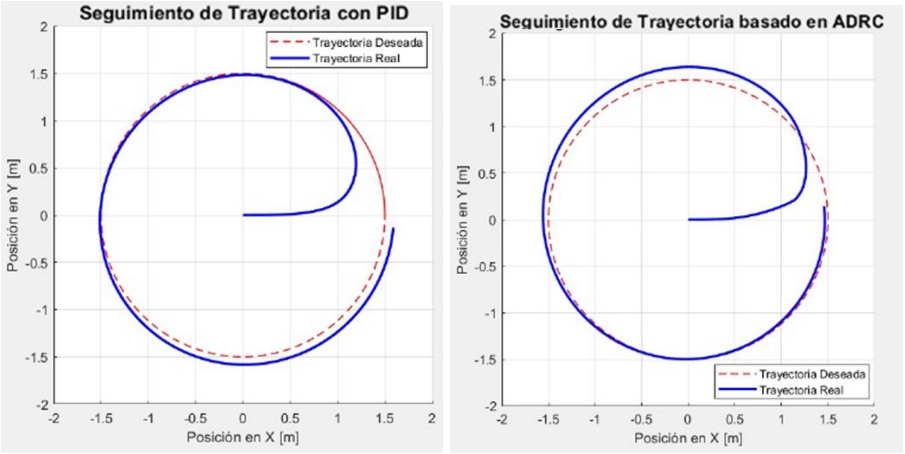
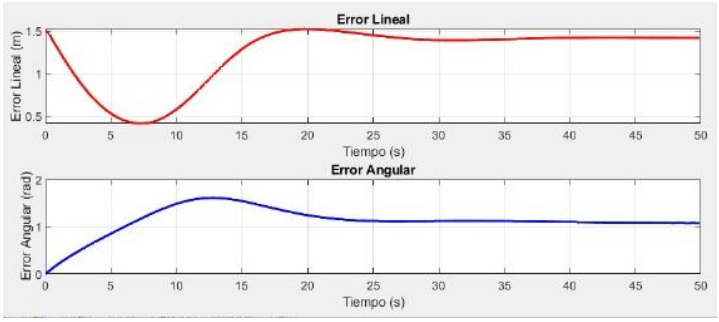
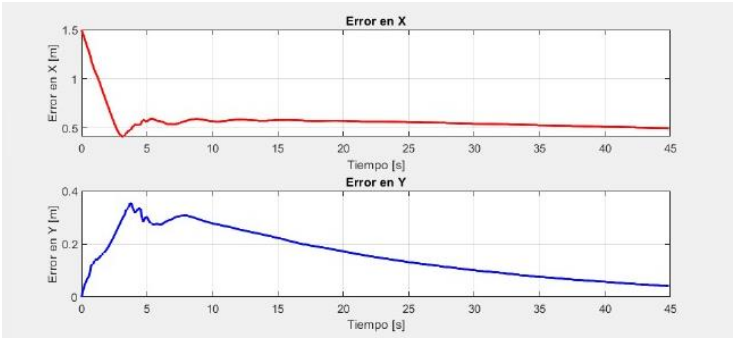


Figura 43.

Comportamiento de velocidades trayectoria circular PID vs SCB-ADRC

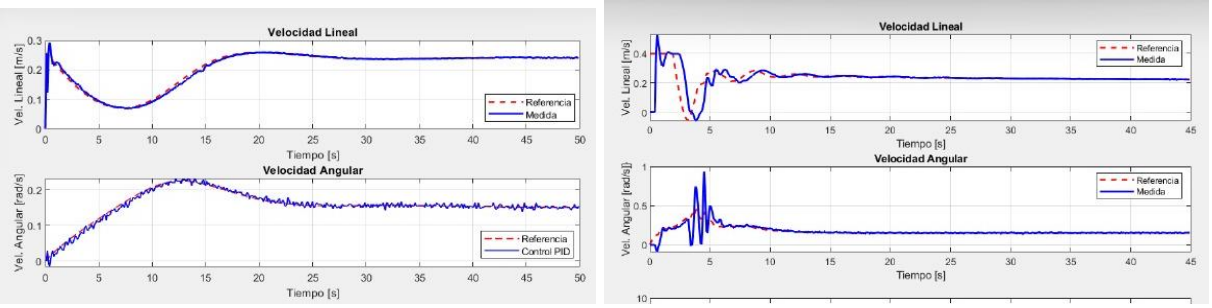




En la gráfica 43 se visualiza el comportamiento de los errores para el modelo de control basado en rechazo activo de perturbaciones, contra un PID normal, es posible visualizar que, aunque ambas señales de error son estables, para el caso del controlador basado en ADRC la magnitud de estos es significativamente menor.

Figura 44.

Comportamiento de errores trayectoria circular PID vs SCB-ADRC



En la figura 44 es posible visualizar el comportamiento de las velocidades angulares y lineales para ambos controladores, en el caso del PID se presenta ruido, pero la señal es estable, por otro lado, en el controlador basado en ADRC se aprecia la señal más suavizada.

4.2.3 Puesta en marcha de control PID vs SCB-ADRC – Trayectoria senoidal

Los parámetros que se tienen en cuenta para la implementación de seguimiento de una trayectoria lineal son los siguientes:

Tabla 7.

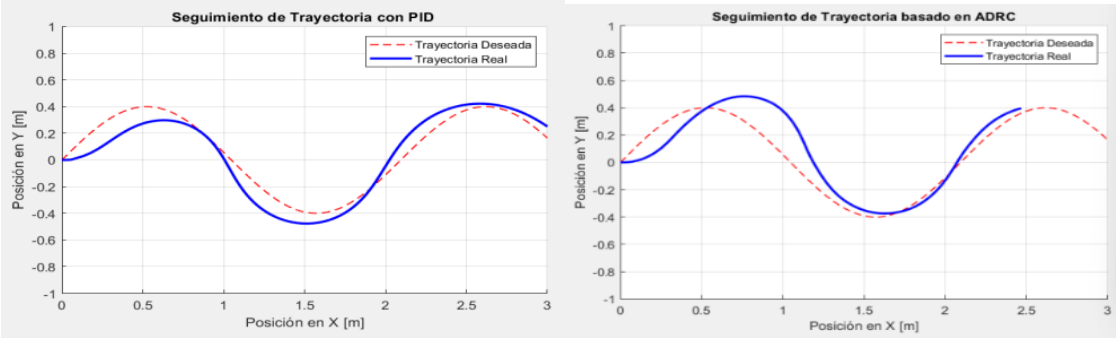
Parámetros Implementación trayectoria senoidal PID – SCB ADRC

	Velocidad Lineal			Velocidad Angular		
	Kp	**Ki**	**Kd**	**Kp**	**Ki**	**Kd**
PID	0,17	0	0	0,14	0,01	0

	Kp	**Ti**	**Ta**	**N**	**α**	**β**
ADRC	0,155	0,6	0,001	31	0,2	0,95

Figura 45.

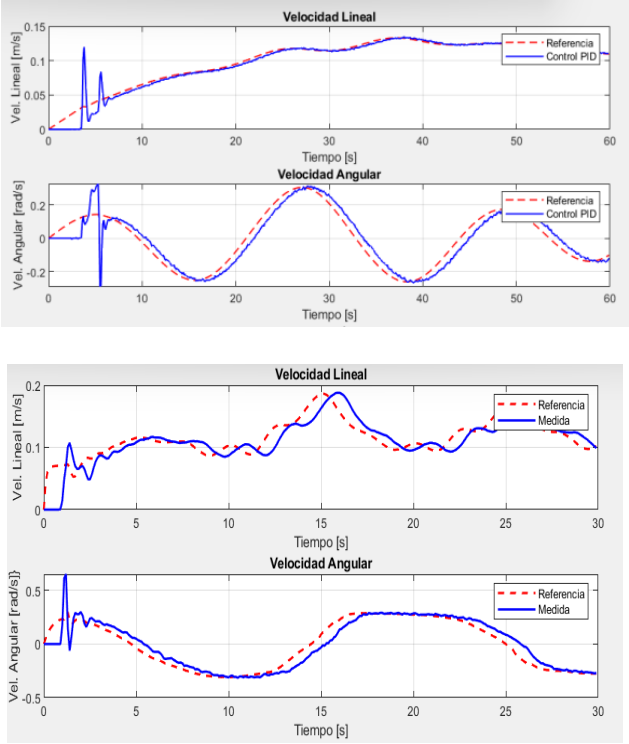
Seguimiento de trayectoria senoidal PID vs SCB-ADRC



Cómo se puede ver en la figura 45, el control basado en rechazo activo de perturbaciones hace un seguimiento más agresivo de la referencia, pero al ser la perturbación tan fuerte, existe un exceso de movimiento que se puede asociar al sobrepico de velocidad al inicio del movimiento, una vez pasado el sobre pico, el robot intenta estabilizarse sobre la referencia, con un desfase minúsculo y sin oscilar como es el caso del controlador PID.

Figura 46.

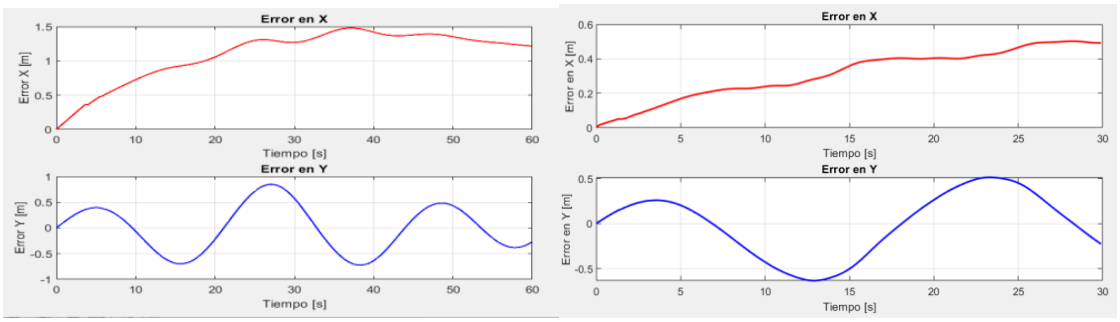
Comportamiento de velocidades trayectoria senoidal PID vs SCB-ADRC



En cuanto al comportamiento de las velocidades es posible afirmar que ambos sistemas presentan un seguimiento de referencias con un leve desfase, es importante resalar que la velocidad lineal tiene una menor magnitud para el caso del PID durante el seguimiento de trayectorias, lo cual apunta a un seguimiento menos exigente en cuando a controlador.

Figura 47.

Comportamiento de errores trayectoria senoidal PID vs SCB-ADRC



En la figura 47 se observa un comportamiento muy similar para ambos sistemas de control, pero para el caso del controlador PID, hay mayor variación en los errores, y estos son de mayor magnitud en comparación al controlador basado en ADRC, por lo que el modelo de control basado en ADRC mostró mejor comportamiento.

4.3 Discusión de Resultados

Se seleccionaron tres curvas de seguimiento, dónde se hacía necesario controlar las velocidades lineales y angulares del robot para conseguir una replicación fiel de trayectorias. De forma adicional se posicionó al robot en un punto externo a esta trayectoria, y se evaluó la capacidad de este para posicionarse y continuar con el seguimiento de la misma.

Para el escenario de la línea recta, se hacía necesario igualar las velocidades de los dos motores, mediante el control PID se obtuvieron ligeras curvaturas en el seguimiento de trayectorias, y en el modelo basado en ADRC se obtuvo una línea recta con un leve desfase, es decir se evidencia que el bloque de control propuesto es más adecuado que un PID tradicional.

En el caso de trayectorias circulares el controlador basado en rechazo activo de perturbaciones tiene un comportamiento muy similar al PID tradicional y no muestra un seguimiento significativamente mejor.

Para el seguimiento de la función senoidal, que es el escenario con mayor variación de velocidades durante el tiempo de implementación se observa una mejor tendencia a estabilizarse y a encontrar la referencia en el controlador basado en ADRC, sin embargo, existen desfases para ambos modelos de control.

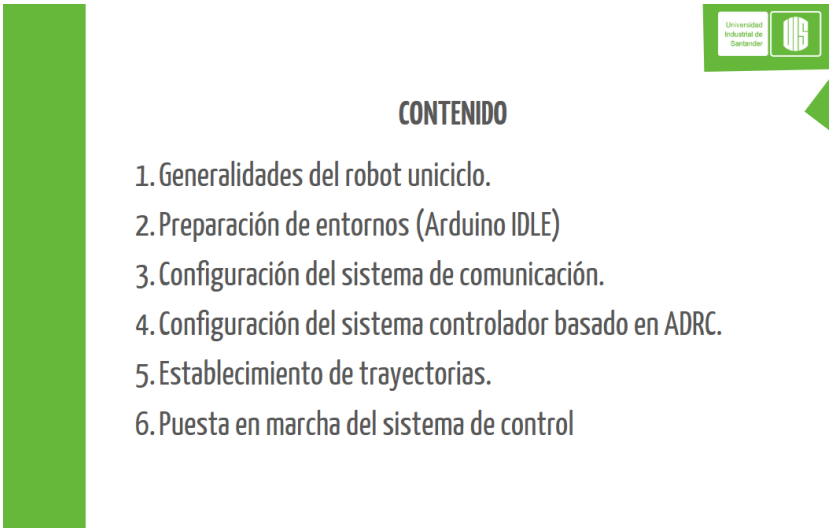
5. Manual de usuario para uso del robot diferencial

En el presente capitulo se describe el contenido del manual de usuario diseñado para ofrecer al usuario una serie de instrucciones de forma clara para la programación y el uso del robot seguidor de trayectorias.

Dentro de los apéndices del trabajo de grado se entrega copia del manual de usuario.

Figura 48.

Contenido del manual de usuario



5.1 Generalidades del robot uniciclo

En este primer capítulo se brinda al usuario una contextualización sobre el robot uniciclo que se va a programar, es de vital importancia conocer las características mecánicas y electrónicas del robot de forma previa para comprender qué se va a controlar.

5.2 Preparación de entornos

En este segmento del instructivo se da al usuario una serie de indicaciones que permitan el uso correcto de la placa ESP32 con el entorno de Arduino, naturalmente este controlador

no es compatible con el IDLE de Arduino, por lo que es necesario instalar de forma manual una serie de controladores que van a permitir la correcta programación del microcontrolador.

5.3 Configuración del sistema de comunicación

En esta parte del instructivo se orienta al usuario en cómo debe programar su placa ESP32, para que esta pueda conectarse a la red Wifi y posteriormente sea posible establecer un protocolo de comunicación entre el ordenador y el robot seguidor de trayectorias.

En este paso es necesario modificar el código que se graba en la placa, y también el código que se ejecuta de forma simultánea en Matlab.

5.4 Establecimiento de trayectorias

Para hacer efectivo el seguimiento de trayectorias es importante tener en cuenta que la trayectoria debe poderse definir como una función en un plano (x,y)

La parametrización de las ecuaciones en función de “t” es la manera como el algoritmo de control las interpreta y hace efectivo su seguimiento.

5.5 Configuración del sistema de control basado en ADRC

En este paso se hace la calibración y sintonización de las constantes de control, los autores sugieren a usuario, sintonizar el controlador siguiendo el método experimental basado en iteraciones descrito en el capítulo 4.

5.6 Puesta en marcha del sistema de control

En la parte final del instructivo se exponen los pasos para ejecutar el controlador previamente programado en el robot uniciclo.

6. Trabajos futuros

Teniendo en cuenta la versatilidad del robot móvil que fue seleccionado, y la cantidad de sensores que pueden incorporarse a la placa de control, se sugieren dos variaciones que permiten explorar el control de robots móviles desde otras perspectivas.

6.1 Seguimiento de trayectorias con un enfoque de rechazo de obstáculos

Aprovechando la presencia de los sensores infrarrojos de tipo SHARP, se propone un esquema de control que incluya detección y evasión de obstáculos, la perturbación que generaría un obstáculo dentro de la trayectoria se podría analizar como un error externo, y diseñar un controlador específico para mitigar el desfase que estos producirían.

6.2 Seguimiento de trayectorias con un sensor de retroalimentación.

Teniendo en cuenta que el robot cuenta con un acelerómetro, se propone la idea de utilizar la información que recoge el acelerómetro como un bloque de retroalimentación que permita añadir información útil al sistema de control y mejorar su precisión.

7.Conclusiones

El presente trabajo de grado permitió el desarrollo de una estrategia de control para el seguimiento de trayectorias en un robot móvil de tipo uniciclo, se utilizó un enfoque de rechazo activo de perturbaciones, a través del cual se consiguió estabilidad en los seguimientos, incluso teniendo en cuenta la incertidumbre del proceso.

Se realizó el diseño de un controlador basado en ADRC a través de un PID filtrado, lo cual permitió un desempeño óptimo para el manejo del modelo dinámico del robot móvil, y el seguimiento de trayectorias. Se realizaron simulaciones que permitieron visualizar el comportamiento del bloque de control propuesto, y cómo este superó al PID tradicional.

La implementación del controlador en el robot móvil uniciclo permitió demostrar la efectividad del seguimiento de trayectorias en el modelo físico, también se implementó un PID tradicional, contra el que se comparó y se evidenció una respuesta más efectiva en el modelo de control basado en ADRC, donde se garantizó más estabilidad y mejores tiempos de respuesta.

Teniendo en cuenta que los resultados obtenidos fueron favorables, es posible continuar trabajando en modelos de control similares, o incluso optimizar el propuesto en el presente proyecto, es importante mencionar que se creó un manual de usuario que permite la replicación de la estrategia de control, y se propusieron diferentes enfoques utilizando el mismo robot.

En resumen, el proyecto logró cumplir con los objetivos propuestos, pues los resultados obtenidos producto de las simulaciones y la implementación demuestran que el PID filtrado como estrategia basada en rechazo activo de perturbaciones es una técnica efectiva y viable para seguimiento de trayectorias.

8. Bibliografía

Bayona Gómez, A., & Forero Gómez, F. Y. (2023). *Diseño y construcción de un prototipo de robot móvil para navegar en terrenos irregulares*.

Canudas de Wit, C., Siciliano, B., & Bastin, G. (1996). *Theory of robot control*. Springer.

Carreño-Zagarra, J. J., Moreno, J. C., & Guzmán, J. L. (2024). *Optimal active disturbance rejection control for second order systems*.

Gao, W., Wang, Y., & Homaifar, A. (2010). Sliding mode control for a class of uncertain nonlinear systems. *Mathematical Problems in Engineering*.

Gracia Calandín, L. I. (2010). *Modelado cinemático y control de robots móviles con ruedas* (Tesis doctoral). Universidad Politécnica de Valencia, Departamento de Ingeniería de Sistemas y Automática.

Gutiérrez Martínez, A. (2021). *Control de robots móviles de tracción diferencial mediante modos deslizantes de segundo orden*.

Han, J. (2009). From PID to active disturbance rejection control. *IEEE Transactions on Industrial Electronics*.

Hernández Germán, Ríos Luis, Bueno Maximiliano. (2016). *Implementation of a position and movement controller for a differential mobile robot*.

Il Bambino. (2008). *Una introducción a los robots móviles*.

Jeschke, S., Liu, H., & Schilberg, D. (Eds.). (2011). *Intelligent Robotics and Applications: 4th International Conference, ICIRA 2011, Proceedings, Part I*. Springer.

Méndez Canú, A. (2021). *Control de trayectoria para robots móviles usando control difuso adaptativo*.

Murray, R. M., Li, Z., & Sastry, S. S. (1994). *A mathematical introduction to robotic manipulation*.

Oriolo, G., Siciliano, B., & Cacace, J. (2002). WMR control via dynamic feedback linearization: Design, implementation, and experimental validation. *IEEE Transactions on Control Systems Technology*.

Ramírez-Neria, L., Aguilar-López, M. A., & Sossa-Azuela, H. (2022). Control por modos deslizantes para un robot móvil diferencial. *Computación y Sistemas*.

Ramírez-Neria, M., Luviano-Juárez, A., Madonski, R., Hernández-Martínez, E. G., Fernández-Anaya, G., & Lozada-Castillo, N. (2022). Trajectory tracking kinematic control of omnidirectional mobile robots via active disturbance rejection control with anti-peaking mechanism. *IEEE 18th International Conference on Automation Science and Engineering (CASE)*, 2295–2300.

Solano, J. C., Vázquez, Ma. E., & Vargas, J. E. (2024). *Convergencia mecatrónica*.

Villamizar García, J. D., Rangel Castro, J., & Meneses Flóres, J. E. (2022). Diseño y construcción de un robot para desinfección de superficies con luz ultravioleta. *Revista Científica Ingeniería Ciencia, Tecnología e Innovación*.

Yandún Torres, A. I. (2011). *Planeación y seguimiento de trayectorias para un robot móvil*. Escuela Politécnica Nacional.

Yi, Huang, Wenchao, Xue, Gao, Zhiqiang, Sira-Ramirez, Hebertt, Dan, Wu, & Mingwei, Sun. (2014). *Active Disturbance Rejection Control: Methodology, Practice and Analysis*.