

Diseño y validación experimental de un robot omnidireccional de tres ruedas con control PID sobre ESP32.

Design and Experimental Validation of a Three-Wheeled Omnidirectional Robot with ESP32-Based PID Control.

Alexander Armando Maldonado Peña ^a, Eduardo Daniel de la Cruz Bautista ^b, Kristian Freyri Maya Gress ^c, Omar Samperio Vázquez ^d, Juan Carlos González Islas ^e

Abstract:

This study presents the creation, programming, and management of a three-wheeled omnidirectional mobile robot with a hexagonal base, developed as an experimental platform for research in kinematics and trajectory control. An ESP32 unit manages the system, receiving velocity references generated in a Python programming environment and transmitted through wireless communication. The speed regulation of each motor is performed using encoder feedback and a PID algorithm, while route monitoring is verified through tests with predefined paths. Experimental results demonstrate stable and precise performance in executing trajectories, confirming the effectiveness of the proposed methodology. The proposed system provides a low-cost and reproducible platform for research and education in mobile robotics.

Keywords:

Omnidirectional mobile robot, ESP32, kinematics, PID control, trajectory tracking, wireless communication.

Resumen:

Este estudio expone la creación, programación y gestión de un robot móvil omnidireccional de tres ruedas con una base hexagonal, creado como un soporte experimental para la investigación en cinemática y control de trayectoria. Una unidad ESP32 gestiona el sistema, recibiendo referencias de velocidad producidas en un ambiente de programación en Python y transmitiéndolas a través de comunicación sin cables. La regulación de la velocidad de cada motor se lleva a cabo a través de la retroalimentación de los encoders y un algoritmo PID, mientras que el monitoreo de rutas se verifica a través de pruebas con rutas preestablecidas. Los resultados experimentales demuestran un desempeño estable y preciso en la ejecución de trayectorias, confirmando la efectividad de la metodología propuesta. El sistema propuesto ofrece una plataforma de bajo costo y reproducible para la investigación y la enseñanza en robótica móvil.

Palabras Clave:

Robot móvil omnidireccional, ESP32, cinemática, control PID, seguimiento de trayectoria, comunicación inalámbrica.

1. Introducción

Los robots omnidireccionales son plataformas móviles capaces de moverse en todas direcciones sin la necesidad de girar antes, lo cual les proporciona un alto

^b Autor de Correspondencia, Universidad Autónoma del Estado de Hidalgo | Centro de Investigación en Tecnologías de Información y Sistemas | Mineral de la Reforma | Hidalgo | México, <https://orcid.org/0009-0007-9581-4961> ^b, Email: de453444@uaeh.edu.mx ^b

^{a,c,d,e} Universidad Autónoma del Estado de Hidalgo | Centro de Investigación en Tecnologías de Información y Sistemas | Mineral de la Reforma | Hidalgo | México, <https://orcid.org/0009-0006-2122-3117> ^a, <https://orcid.org/0000-0001-8425-0960> ^c, <https://orcid.org/0009-0008-3417-0100> ^d, <https://orcid.org/0000-0002-2190-0660> ^e, Email: ma441044@uaeh.edu.mx ^a, kristian_maya10493@uaeh.edu.mx ^c, omarsamvaz@uaeh.edu.mx ^d, juan_gonzalez7024@uaeh.edu.mx ^e

nivel de destreza y eficacia en espacios pequeños o con obstáculos complicados. Se ha decidido utilizar este tipo de robots en varios campos en años recientes. La implementación de sistemas de control y el perfeccionamiento del diseño de las ruedas han contribuido significativamente a que sean más populares (Taheri & Zhao, 2020; Lu et al., 2025).

Se logra la movilidad en todas las direcciones mediante varios mecanismos. Las ruedas Mecanum y otros diseños de este tipo, posibilitan rotaciones y traslados delicados y exactos (Sato et al., 2023; Ye et al., 2011). La capacidad de carga de los sistemas ha aumentado, su tamaño ha disminuido y su adaptación a diferentes tipos de terreno ha mejorado gracias a las innovaciones actuales (Tanaka et al., 2023). Además, hay propuestas que utilizan patas articuladas o ruedas de tornillo, lo que aumenta la capacidad de operar en terrenos irregulares e incluso bajo el agua (Zou et al., 2018; Xu et al., 2023).

El control es un componente esencial para alcanzar un movimiento preciso en esta clase de robots. Para asegurar la estabilidad, eludir impedimentos ante perturbaciones externas y optimizar el seguimiento de trayectorias, se han diseñado estrategias que incluyen controladores adaptativos y embebidos, entre los cuales se encuentran las redes neuronales, LQR y la lógica difusa (Qian et al., 2023; Wu et al., 2021). Asimismo, el uso combinado de sistemas de visión, control predictivo y métodos de mapeo ha hecho posible que estas plataformas sean más autónomas en contextos dinámicos. Esto posibilita una detección más exacta y una mejor capacidad para prevenir choques en áreas concurridas (Yao et al., 2024; Mamun et al., 2018).

Los robots omnidireccionales tienen un rango extenso de aplicaciones. En el sector industrial, se utilizan en la logística interna para optimizar el traslado de materiales (Masmoudi et al., 2016; Azizi et al., 2021). Se han propuesto, además, versiones híbridas y de tamaño miniatura; por ejemplo, los robots que poseen las dimensiones de una mano y cuya creación está pensada para operar en espacios peligrosos o reducidos (Wu & Huang, 2021; Sprunk et al., 2016; Watanabe, 1998).

Las investigaciones recientes se han enfocado en el desarrollo de robots más autónomos y colaborativos, capaces de aprender itinerarios, identificar patrones en su entorno y adaptarse a circunstancias inesperadas mediante la utilización de inteligencia artificial (Menegatti et al., 2006). Estas características los transforman en herramientas valiosas para asistir, transportar provisiones en hospitales y llevar a cabo operaciones de

rescate en áreas afectadas por desastres, donde su habilidad para maniobrar es esencial (Micolini et al., 2022).

Otro aspecto clave de investigación es la eficiencia energética, con esfuerzos dirigidos hacia el uso de materiales ligeros y diseños compactos que reduzcan el consumo y extiendan el tiempo de operación, lo cual resulta indispensable en aplicaciones de larga duración (Zhang et al., 2023; Liu et al., 2022). No obstante, un reto persistente es el elevado costo de producción. Dependiendo del nivel de precisión y tiempo de respuesta requerido, los costos pueden incrementarse significativamente (Liu et al., 2022).

En este marco, esta investigación se distingue de otros trabajos al sugerir la creación y validación de un robot omnidireccional de tres ruedas como plataforma experimental que esté orientada específicamente a la evaluación y estudio de métodos para controlar trayectorias en tiempo real. Esta investigación tiene como objetivo proporcionar un entorno de experimentación accesible para el estudio de tácticas de control, en especial el control PID, y su efecto en la estabilidad y precisión del desplazamiento. Esto contrasta con estudios recientes que se enfocan en aplicaciones industriales o en la miniaturización de plataformas. Esta perspectiva representa un aporte práctico al área, porque brinda una base de pruebas que es reutilizable y adaptable para futuras investigaciones en robótica móvil.

Modelo Matemático

El análisis cinemático de un sistema mecánico se refiere a la determinación de la posición y la velocidad de los distintos elementos mecánicos que conforman el mecanismo en estudio. La combinación de la posición y la velocidad de un elemento en un momento dado se denomina estado de dicho elemento. El cálculo del estado cinemático de elementos mecánicos, como la posición, la orientación y la tasa de cambio, a partir de movimientos axiales dados representa el problema de cinemática directa (Al-Ammri & Ahmed, 2010).

Para entender cómo se mueve un robot omnidireccional de tres ruedas, primero hay que revisar la cinemática, es decir, las reglas matemáticas que explican su movimiento. La Figura 1 muestra los vectores de movimiento que puede tener cada conjunto de ruedas: unos indican hacia dónde se puede mover libremente el robot y otros señalan las limitaciones que aparecen según la posición de las ruedas. En pocas palabras, el robot se desplaza porque cada motor hace girar su rueda, y la

combinación de esas tres ruedas trabajando al mismo tiempo permite que el robot avance, gire o se mueva en cualquier dirección sin necesidad de cambiar su orientación (Al-Ammri & Ahmed, 2010).

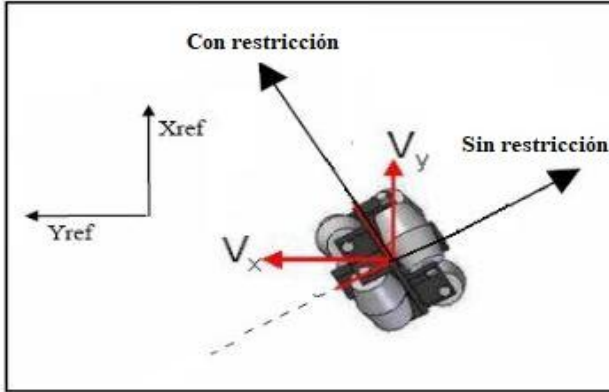


Figura 1. Cinemática de la rueda Omnidireccional.

Mediante el uso de la Figura 2, es posible obtener las ecuaciones cinemáticas del sistema de propulsión.

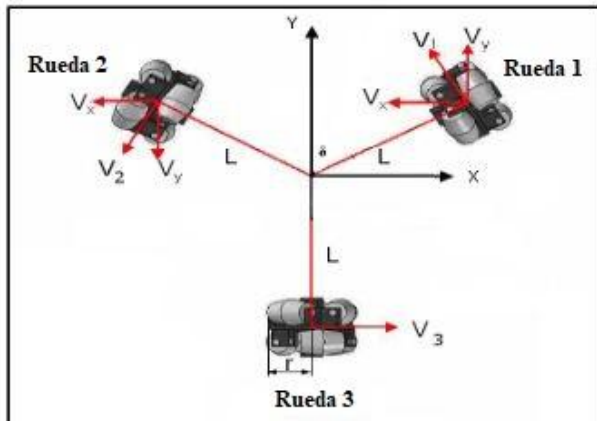


Figura 2. Representación cinemática del robot omnidireccional de tres ruedas.

Las ecuaciones empleadas en el sistema de control del robot son las siguientes:

$$V_x = V_3 - V_1 \cos(\theta) - V_2 \cos(\theta) \quad (1)$$

$$V_y = V_1 \sin(\theta) - V_2 \sin(\theta) \quad (2)$$

$$V_\phi = \frac{V_1}{L} + \frac{V_2}{L} + \frac{V_3}{L} \quad (3)$$

$$V_{i(1,2,3)} = w \cdot r \quad (4)$$

Donde:

r = Radio de la rueda (cm).

w = Velocidad angular de la rueda (rad/s)

V_x = Velocidad del robot en el eje de x (rad/s)

V_y = Velocidad del robot en el eje de y (rad/s)

V_i = Velocidad traslacional de cada rueda (rad/s)

Entonces, en el análisis teórico, la velocidad de cada una de las tres ruedas será igual a la velocidad del servomotor multiplicada por el radio de la rueda.

Las ruedas están separadas por 120° , entonces damos por hecho que nuestra $\theta = 60^\circ$, por lo tanto, podemos reinterpretar las anteriores ecuaciones en una matriz jacobiana (Al-Ammri & Ahmed, 2010).

La matriz jacobiana relaciona los movimientos del robot en el plano (adelante, a los lados, y girar sobre su eje) con las velocidades que deben tener las ruedas. En otras palabras, transforma las órdenes de movimiento en coordenadas V_x, V_y, V_ϕ en las velocidades angulares de cada motor. Gracias a esta relación matemática, se calcula cómo deben girar los motores para que el robot siga una trayectoria deseada o mantenga un control estable durante su desplazamiento.

$$\begin{bmatrix} V_x \\ V_y \\ V_\phi \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} & -\frac{1}{2} & 1 \\ \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & 0 \\ \frac{1}{L} & \frac{1}{L} & \frac{1}{L} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} \quad (5)$$

En la Figura 3 se muestra el funcionamiento del sistema en un diagrama de bloques.

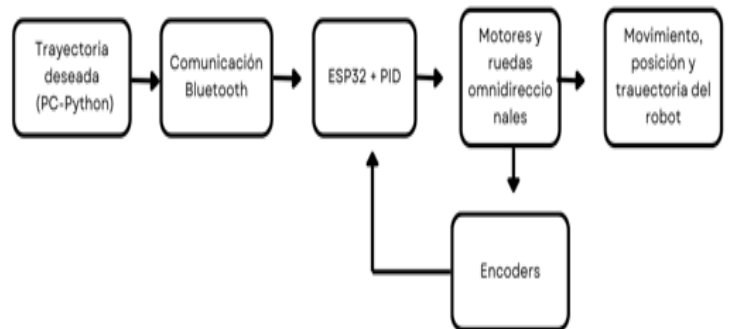


Figura 3. Funcionamiento del sistema representado en un diagrama de bloques.

2. Construcción

El primer paso para la creación del robot fue el diseño de la base en 3D. Por ende, se optó por una base hexagonal para colocar los tres motorreductores de manera balanceada. Esto garantiza la estabilidad del robot y que cada motorreductor esté separado por 120° . El prototipo, posteriormente, fue diseñado en Tinkercad y fabricado a través de un servicio de impresión 3D con resina dura. Se eligió este material debido a que es ligero, duradero y tiene un coste de fabricación reducido. Sin poner en peligro la estabilidad del sistema, la base conserva la

rigidez estructural que se requiere. La instalación de los motores se llevó a cabo con tornillos M3, los cuales están equipados con tuercas de fijación y acoplamientos, lo que asegura un buen nivel de sujeción y minimiza las vibraciones.

El diseño final de la base se muestra en la Figura 4.

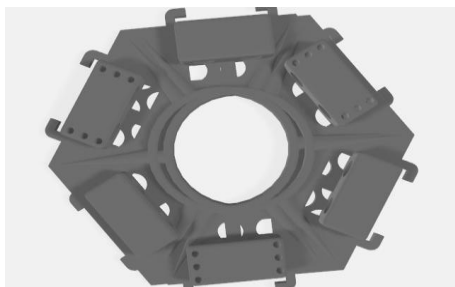


Figura 4. Perspectiva 3D de la base del robot.

Componentes físicos y electrónicos del robot

Se seleccionaron los motoredutores **Pololu 30:1 Metal Gearmotor 37Dx68L** mm 12V que tienen un encoder de 64 CPR debido a su gran resolución y óptima relación coste-calidad. Estos motores tienen la capacidad de producir un torque máximo de 14 kg.cm y una velocidad sin carga de 330 RPM, lo cual les posibilita generar la fuerza y precisión necesarias para mover el robot con estabilidad, aun en terrenos dispares, lo que simplifica el control del movimiento en contextos complejos.

En la parte de potencia se optó por un driver L298N debido a que es fácil de integrar con la ESP32 y puede gestionar corriente en aplicaciones prototípicas. Para el control se utilizó una ESP32 Dev Kit, la cual brinda conectividad inalámbrica y un bajo consumo energético; además, se añadió un módulo Bluetooth HC-06 para poder comunicarse con la PC. La plataforma se alimenta con una batería de 12 V y se usa un regulador que ajusta el voltaje para la ESP32.

La selección de componentes se justificó principalmente en términos de costo accesible, compatibilidad y facilidad de integración, asegurando así un sistema estable y eficiente para la validación del control de trayectoria.

Diseño de la PCB

Sobre la base se colocó una placa electrónica (PCB). A ella se fijaron los componentes principales - una tarjeta de desarrollo ESP32, que procesa y controla - un módulo de comunicación Bluetooth HC-06, para el enlace con la computadora y las conexiones a los tres motoredutores. En la Figura 5 se muestra el esquema de las conexiones usado para diseñar la placa.

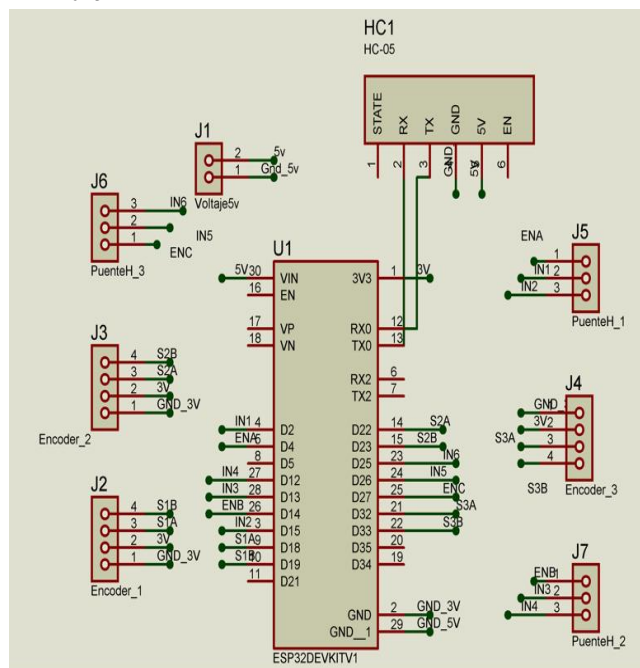


Figura 5. Esquema de la PCB.

Con esta disposición, el cableado se mantuvo ordenado y el montaje fue compacto. La placa es doble cara como se puede ver en la Figura 6.

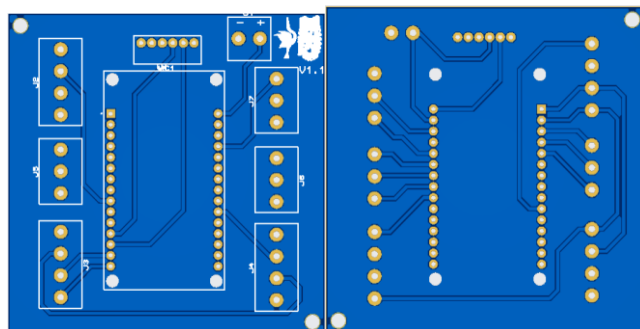


Figura 6. Capa superior e inferior de la PCB.

Comunicación y coordinación entre la PC y el robot

La computadora y el robot se comunican por Bluetooth por el módulo HC-06. Este opera como un puerto serial sin cables. Por medio de esta conexión, se envían y reciben datos sin interrupción.

En esta conexión, el programa de Python de la computadora envía al robot las trayectorias y velocidades deseadas a la ESP32.

La ESP32, que está programada en Arduino, recibe los datos. Calcula las acciones que se necesitan y usa el control PID directo en los motoredutores, además,

devuelve información al programa de Python, como las velocidades medidas. Esto permite que el programa las compare con las velocidades de referencia y siga la trayectoria de forma correcta.

Una vez terminada la parte de programación se procedió a ensamblar el robot dando como resultado la Figura 7.

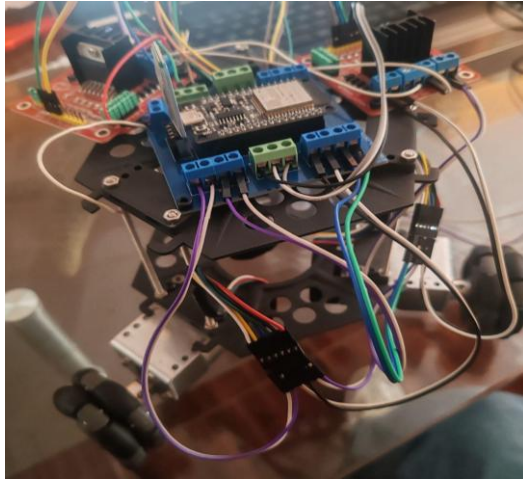


Figura 7. Vista general de robot omnidireccional ensamblado.

Proceso de sintonización del controlador PID

Se llevó a cabo un ajuste gradual del controlador PID, iniciando con todas las ganancias igual a cero. En primer lugar, K_p se aumentó hasta que la velocidad del motor respondiera de forma aceptable; esto es, una salida constante y con error bajo en el régimen permanente. Tras establecer esta ganancia, se procedió a modificar T_i con el fin de suprimir el error en estado estacionario y optimizar la precisión del sistema. Finalmente, se añadió T_d con el propósito de disminuir las oscilaciones y optimizar la estabilidad total de la respuesta.

Este procedimiento hizo posible encontrar una serie de parámetros apropiados para el sistema, a través de un método práctico de ajuste progresivo. El rendimiento fue equilibrado, en cuanto a estabilidad, exactitud y velocidad de respuesta, sin que fuera necesario utilizar procedimientos complicados para la sintonización.

3. Sistema de control

En la robótica móvil, particularmente en sistemas de locomoción no convencionales como vehículos de tres llantas, el diseño de una ley de control adecuada resulta fundamental para garantizar un comportamiento dinámico conforme a los objetivos de navegación, estabilidad y seguimiento de trayectoria. Considerando las características físicas del sistema y las restricciones cinemáticas impuestas por su configuración, se busca

desarrollar una estrategia de control realimentado que permita al vehículo responder de manera precisa y robusta ante perturbaciones externas y variaciones en el entorno. El propósito central es abordar los desafíos inherentes al control de movimiento, tales como la compensación de errores de seguimiento, la adaptación a terrenos irregulares y la coordinación de actuadores en sistemas con movilidad diferencial, por lo que se deben atacar tres aspectos relevantes:

- **Posición:** el propósito es situar al robot en un punto de referencia establecido, con una orientación deseada.
- **Seguimiento de trayectoria:** la meta es lograr que el robot logre y permanezca con un error cero en una ruta diseñada en el tiempo.
- **Monitoreo de ruta:** el propósito es asegurar que el robot siga una ruta deseada, sin ninguna restricción temporal.

Se creó un algoritmo de control que facilita la determinación de las velocidades de las tres ruedas mediante la inversión de la matriz Jacobiana, incluyendo el error entre la trayectoria deseada y la real, con el objetivo de rectificar desviaciones y garantizar un monitoreo exacto del movimiento del robot.

$$\underline{q} = J^{-1}(h_d - h_e) \quad (13)$$

Donde:

$h_d =$ Posición deseada

$h_e =$ Error entre la trayectoria deseada y la real

$\underline{q} =$ Variables del movimiento del robot

Control de posición

El control de posición consiste en ubicar al robot en un punto de referencia deseado, con o sin una orientación específica, en la Figuras 8 y Figura 9 observamos el control de posición del robot simulado.

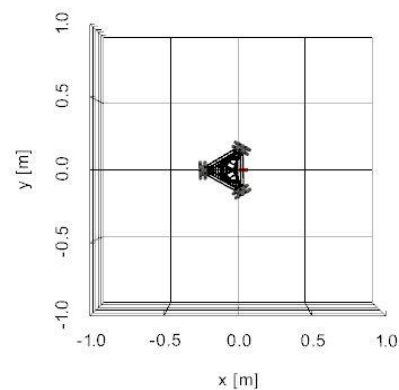


Figura 8. Orientación inicial del robot en 0°.

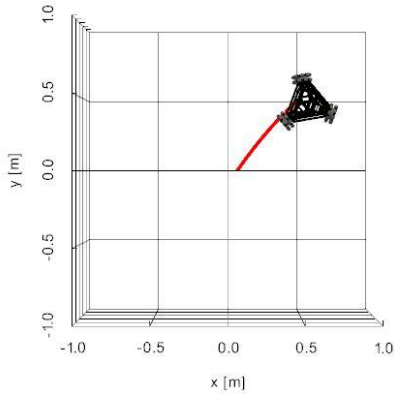


Figura 9. Control con una orientación de 160° .

Tomando en cuenta la ecuación (13), en este caso h_d es constante, por lo tanto $\dot{h}_d = 0$. Por lo tanto, obtenemos la siguiente ecuación:

$$\underline{q} = J^{-1}(-h_e) \quad (14)$$

$$\underline{q}_{ref} = J^{-1}(Kh_e) \quad (19)$$

Método Lyapunov

La teoría de estabilidad de Lyapunov constituye una herramienta fundamental en el análisis de sistemas dinámicos no lineales, especialmente en aplicaciones de robótica donde los modelos suelen presentar comportamientos complejos y dependientes del tiempo. Esta teoría permite evaluar rigurosamente la estabilidad de sistemas de control, incluso en presencia de incertidumbres o perturbaciones externas, y proporciona una base sólida para el diseño de controladores robustos. El enfoque se centra en la construcción de una función candidata de Lyapunov que satisfaga condiciones específicas de positividad y derivada negativa, lo cual garantiza la estabilidad asintótica del sistema en torno a sus estados de equilibrio. En robótica móvil, este método es particularmente útil para asegurar el comportamiento deseado en tareas como el seguimiento de trayectorias, la regulación de postura y la coordinación de múltiples grados de libertad.

Ahora, Se propone una función candidata de Lyapunov para evaluar la estabilidad del sistema de control, definida como:

$$V(h_e) = \frac{h_e^t h_e}{2} \quad (15)$$

donde $h_e \in \mathbb{R}^n$ representa el vector de error del sistema. Esta función es positiva definida y refleja la energía cuadrática del error.

La derivada temporal de $V(h_e)$ se obtiene aplicando la regla de la cadena:

$$\underline{V} = h_e^t \underline{\dot{h}_e} \quad (16)$$

Para garantizar la estabilidad del sistema, se propone una ley de control basada en realimentación lineal negativa:

$$\underline{\dot{h}_e} = -Kh_e \quad (17)$$

donde $K \in \mathbb{R}^{n \times n}$ es una matriz definida positiva. Sustituyendo la ecuación (17) en (16), se obtiene:

$$\underline{V} = -h_e^t K h_e < 0 \quad (18)$$

Dado que K es simétrica y definida positiva, la derivada de $V(h_e)$ es negativa definida. Por lo tanto, el sistema es **asintóticamente estable** en el sentido de Lyapunov.

4. Control de seguimiento de trayectoria

La gestión del seguimiento de trayectoria en robots implica garantizar que el robot siga como referencia a una curva parametrizada en el tiempo.

El propósito de este tipo de controlador es lograr, a través de una ley de control, que el robot logre y mantenga con error cero los estados deseados variados a lo largo del tiempo. Estos estados definen la ruta a seguir. Se observa en la Figura 10 la simulación del seguimiento de trayectoria.

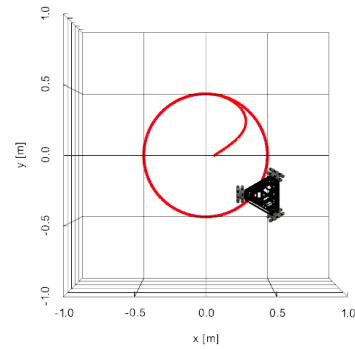


Figura 10. Simulación ideal del seguimiento de trayectoria.

Si se considera que:

$$\underline{q} = J^{-1}(h_d - h_e) \quad (20)$$

En este caso $\dot{h}_d \neq 0$, por lo tanto, la ley de control queda de la siguiente forma.

$$\underline{q}_{ref} = J^{-1}(\dot{h}_d + Kh_e) \quad (21)$$

Para el análisis de estabilidad se asume el seguimiento perfecto de velocidad.

$$\underline{q}_{ref} = \underline{q} \quad (22)$$

Después se establece la función candidata de Lyapunov.

$$V(h_e) = \frac{h_e^t h_e}{2} \quad (23)$$

$$\underline{V} = h_e^t \underline{h_e} \quad (24)$$

Por último, verificamos las condiciones de la derivada.

$$\underline{h} = Jq \quad (25)$$

$$\underline{q}_{ref} = J^{-1}(\underline{h}_d + K h_e) \quad (26)$$

$$\underline{h}_e = \underline{h}_d - \underline{h} \quad (27)$$

$$\underline{h} = JJ^{-1}(\underline{h}_d + K h_e) \quad (28)$$

$$\underline{h}_d - \underline{h}_e = \underline{h}_d + K h_e \quad (29)$$

$$\underline{h}_e = -K h_e \quad (30)$$

$$\underline{V} = -h_e^t K h_e < 0 \quad (31)$$

Obtenemos una función definida negativa, por lo que el sistema para el seguimiento de trayectoria es asintóticamente estable.

El **control de seguimiento de rutas** supone que el robot siga la ruta sin ninguna determinación temporal, en contraposición al seguimiento de trayectorias, donde los puntos representan una función temporal.

Un camino es un conjunto de líneas que conectan puntos situados en el plano o espacio.

Un camino también puede provenir de la discretización de una curva continua que interpola ciertos puntos de la misma

Consideramos la ecuación anteriormente vista para el seguimiento de trayectoria (21). Para el siguiente caso necesitamos considerar una velocidad distinta:

$$\underline{P}_d \neq \underline{h}_d \quad (32)$$

Reemplazando los términos obtenemos la siguiente ley de control.

$$\underline{q}_{ref} = J^{-1}(\underline{P}_d + K h_e) \quad (32)$$

5. Pruebas experimentales

En esta sección se hacen tres experimentos que corresponden al posicionamiento, seguimiento de trayectoria y seguimiento de rutas.

Control de posición

A continuación, se presenta el resultado de una prueba de control de posición donde se requiere que el robot se posicione en la coordenada (5,5), cómo se puede ver en la Figura 11.

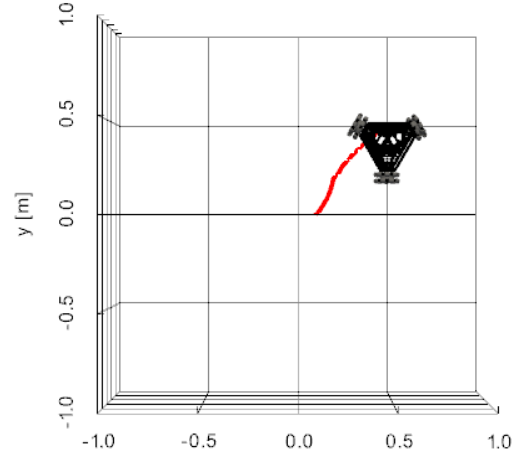


Figura 11. Movimiento real del robot trazando una línea recta.

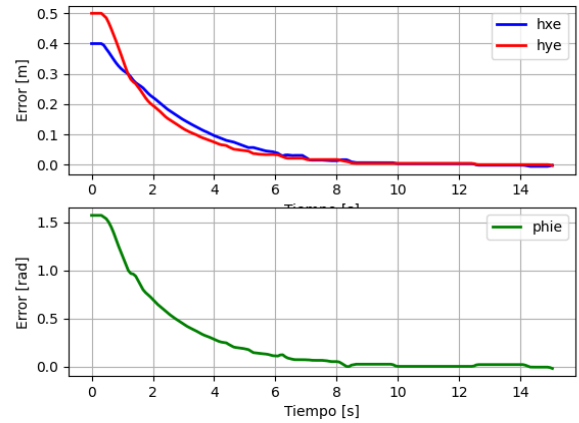


Figura 12. Evolución de los errores de posición y orientación del robot - Control de posición.

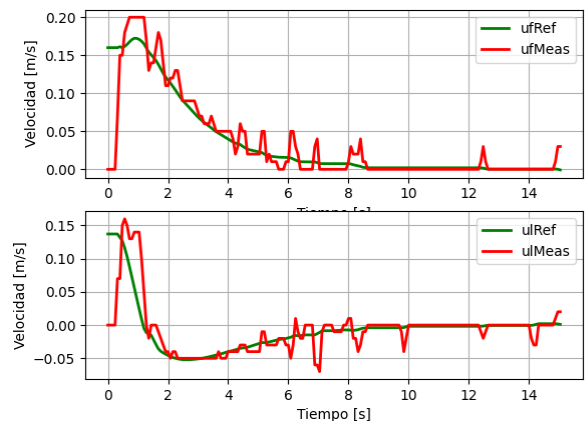


Figura 13. Seguimiento de velocidades en la dirección frontal y lateral comparada con las velocidades ideales - Control posición.

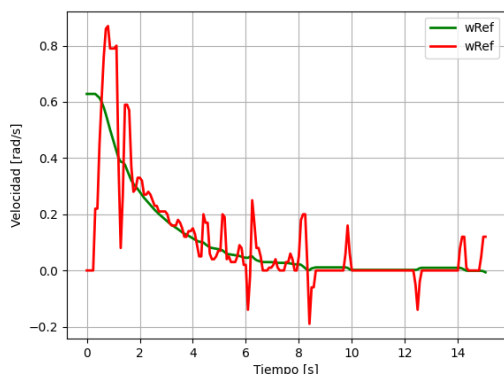


Figura 14. Comparación de la velocidad angular deseada y medida del robot- Control de posición.

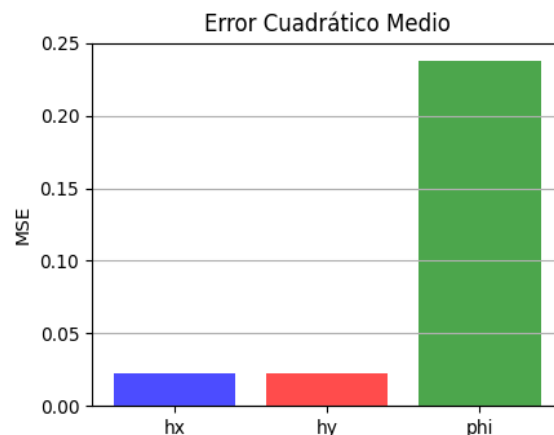


Figura 17. Error cuadrático medio (ECM) – Control de posición.

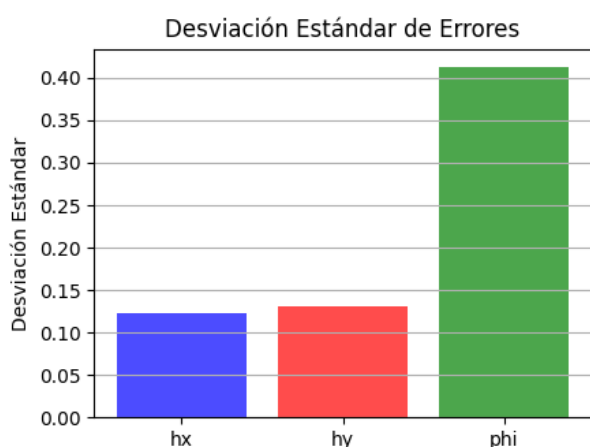


Figura 15. Desviación estándar del sistema – Control de posición.

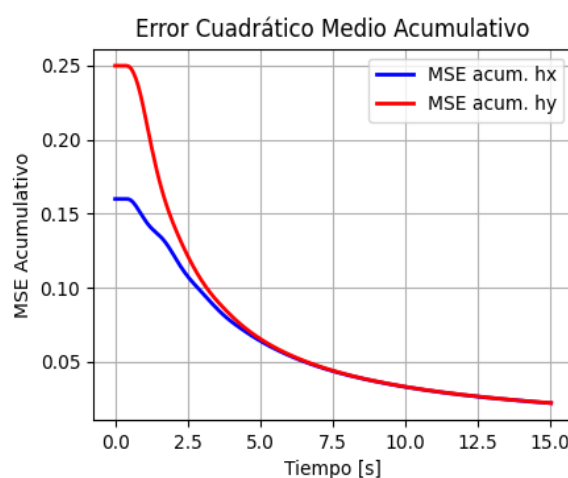


Figura 18. Evolución del error cuadrático medio acumulativo (ECMA) – Control de posición.



Figura 16. Evolución temporal de la desviación estándar – Control de posición.

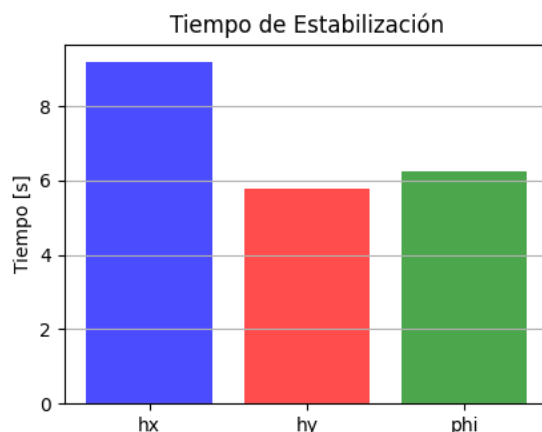


Figura 19. Tiempo de estabilización errores de posición y orientación del robot – Control de posición.

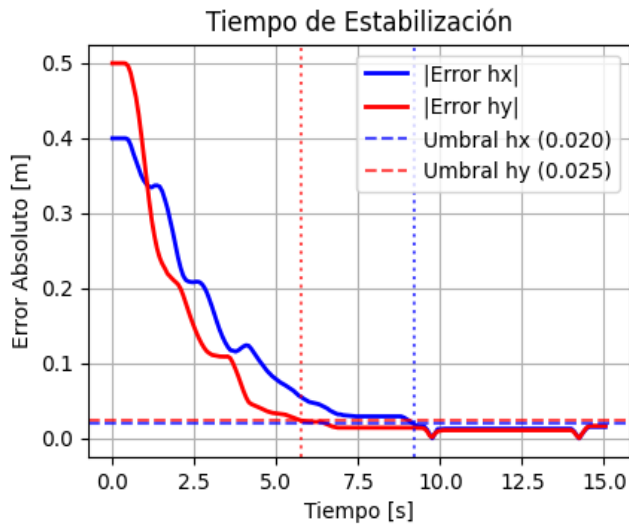


Figura 20. Tiempo de estabilización con umbrales y líneas de referencia – Control de posición.

En el primer experimento, el robot recorre una línea recta, 0.5m en el eje X y 0.5m en el eje Y, con una posición deseada de 90°. El robot llegó al punto final, pero el comportamiento tuvo pequeñas oscilaciones en el camino. Esto muestra que el controlador PID hace su trabajo, así que podemos ajustar los números para que el error sea menor y el resultado sea más exacto.

Los resultados cuantitativos nos revelan que el tiempo de estabilización fue óptimo y confirmamos que el desempeño del sistema es aceptable.

Pese a esas variaciones, el robot acomodó correctamente el camino trazado, así como la posición deseada, lo que corrobora que la táctica de control empleada es eficaz.

Se observa en la Figura 12 que los errores tienden a 0. Concluimos que el algoritmo de control diseñado para esta aplicación es correcto y el sistema es asintóticamente estable.

Control de seguimiento de trayectoria

Ahora se procede a hacer una prueba para seguimiento de trayectoria en el que se le pide al robot seguir el perímetro de una circunferencia de 0.5 m de radio como se aprecia en la Figura 21.

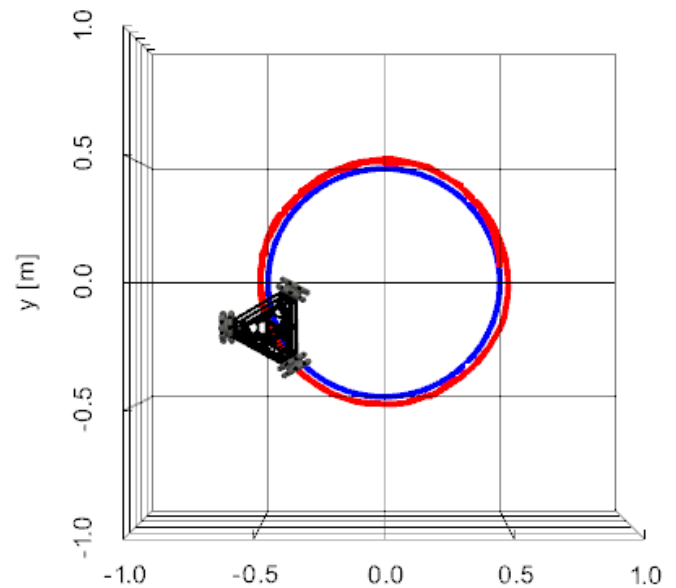


Figura 21. Movimiento real del robot siguiendo una trayectoria circular.

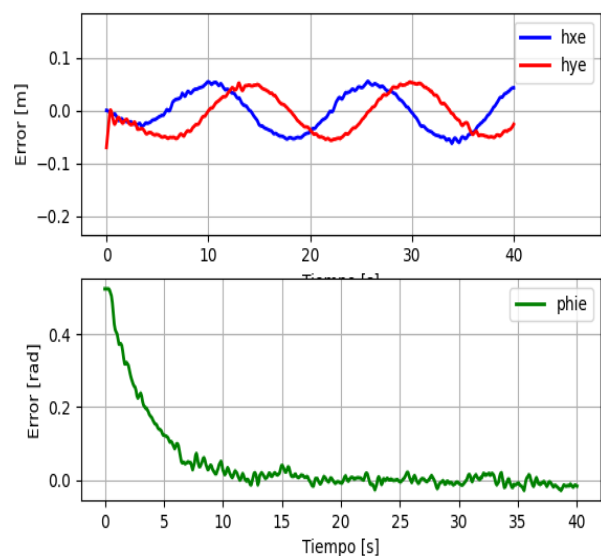


Figura 22. Evolución de los errores de posición y orientación del robot- Seguimiento de trayectoria.

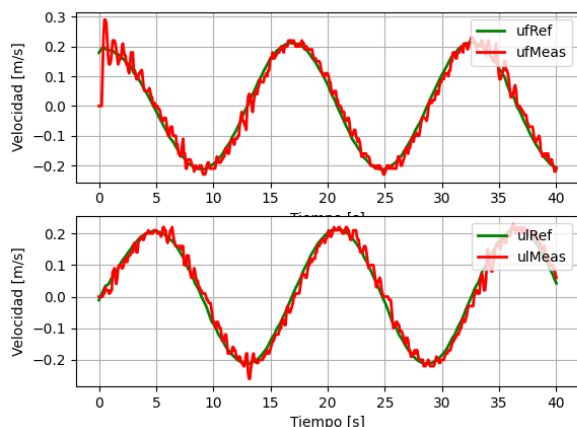


Figura 23. Seguimiento de velocidades en la dirección frontal y lateral comparada con las velocidades ideales - Seguimiento de trayectoria.

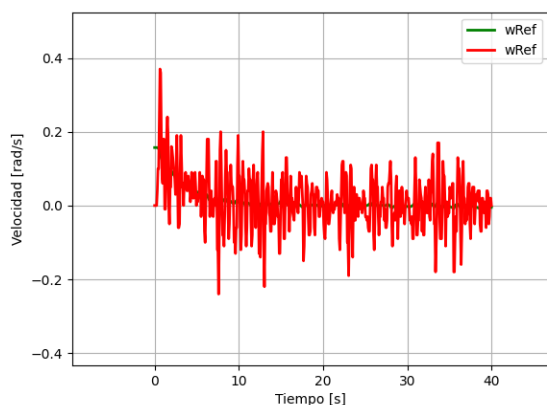


Figura 24. Comparación de la velocidad angular deseada y medida del robot - Seguimiento de trayectoria.

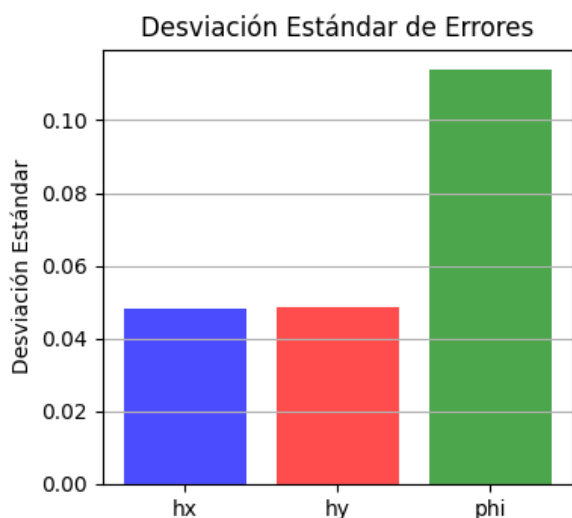


Figura 25. Desviación estándar del sistema – Seguimiento de trayectoria.

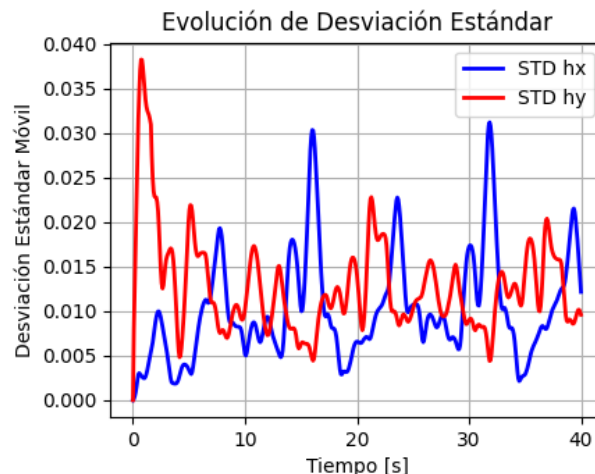


Figura 26. Evolución temporal de la desviación estándar – Seguimiento de trayectoria.

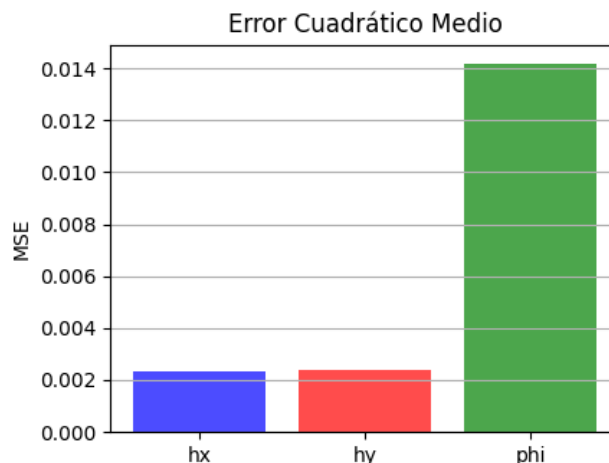


Figura 27. Error cuadrático medio (ECM) – Seguimiento de trayectoria.

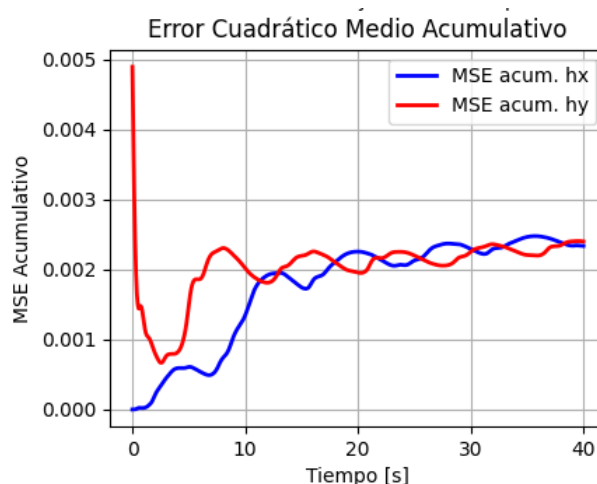


Figura 28. Evolución del error cuadrático medio acumulativo (ECMA) – Seguimiento de trayectoria.

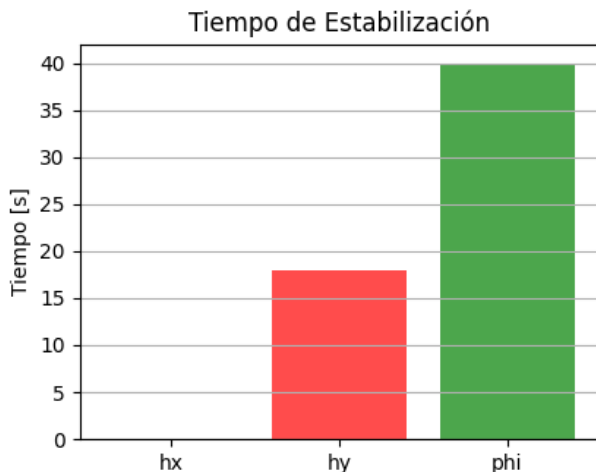


Figura 29. Tiempo de estabilización errores de posición y orientación del robot – Seguimiento de trayectoria.

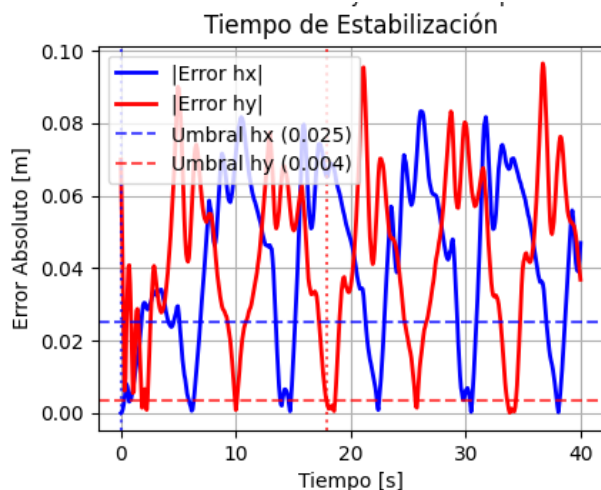


Figura 30. Tiempo de estabilización con umbrales y líneas de referencia – Seguimiento de trayectoria.

En esta prueba experimental podemos notar que el robot sigue la ruta circular dada por el programa de Python, observamos que el robot se desempeña de manera precisa y que las oscilaciones son mínimas. Se observa en la Figura 22 que el error tiende a 0, por lo que demostramos que el algoritmo de control es correcto y el sistema es asintóticamente estable. Concluimos que el resultado para este algoritmo de control es satisfactorio. Además, se obtuvieron medidas cuantitativas cuyos resultados fueron óptimos y confirman el buen desempeño del sistema.

Las limitaciones físicas del prototipo, como el uso de llantas de bajo costo y componentes de hardware básicos, así como la necesidad de un ajuste más preciso en los parámetros PID, son las que principalmente explican las oscilaciones actuales. Estos elementos inducen ligeras imprecisiones en la

respuesta, pero no tienen un impacto importante en la estabilidad o la validez de los resultados obtenidos. Se propone, como posibles soluciones, hacer un ajuste sistemático del controlador PID, usar llantas de mejor calidad y optimizar la rigidez mecánica del chasis. Esto haría posible disminuir las vibraciones y aumentar aún más la precisión del sistema.

Control de seguimiento de ruta

Ahora se plantea el seguimiento de polígono como se muestra en la Figura 31.

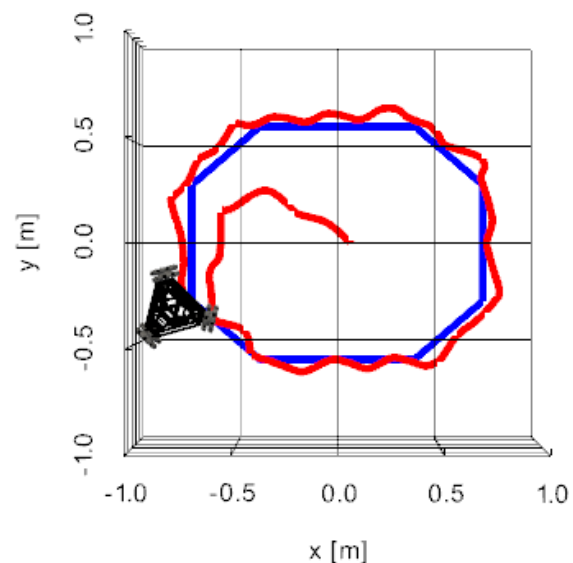


Figura 31. Movimiento real del robot siguiendo un camino trazado.

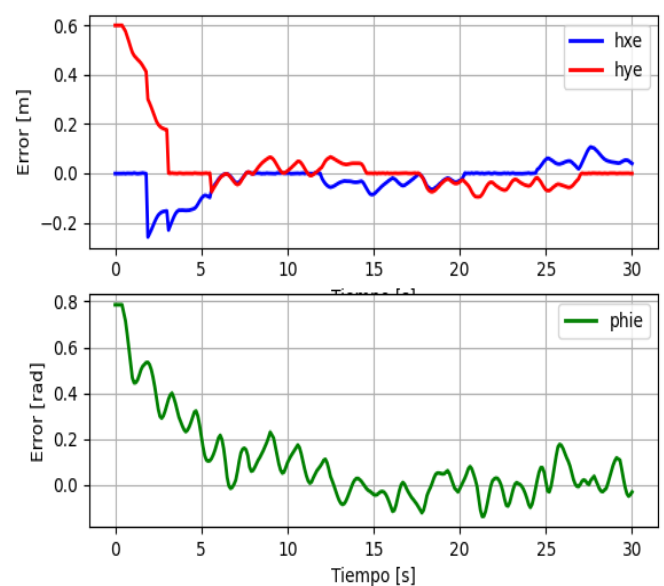


Figura 32. Evolución de los errores de posición y orientación del robot - Seguimiento de ruta.

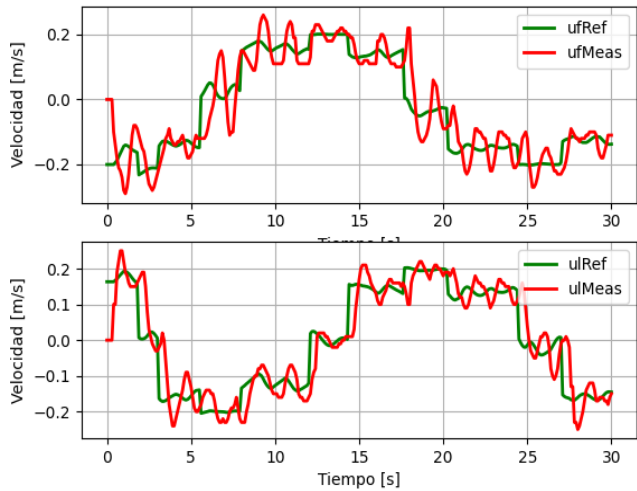


Figura 33. Seguimiento de velocidades en la dirección frontal y lateral comparada con las velocidades ideales - Seguimiento de ruta.

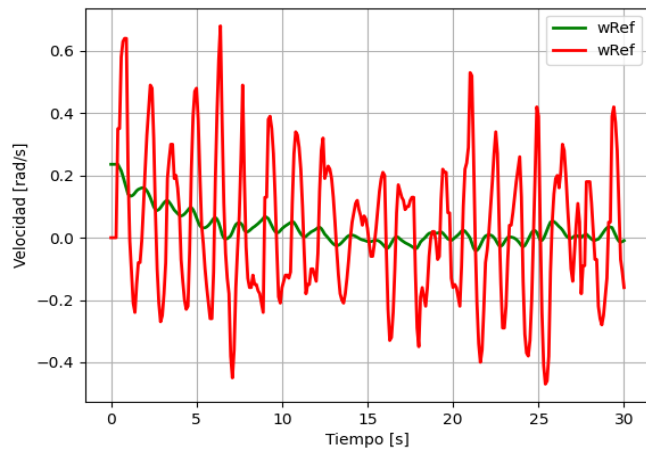


Figura 33. Comparación de la velocidad angular deseada y medida del robot - seguimiento de ruta.

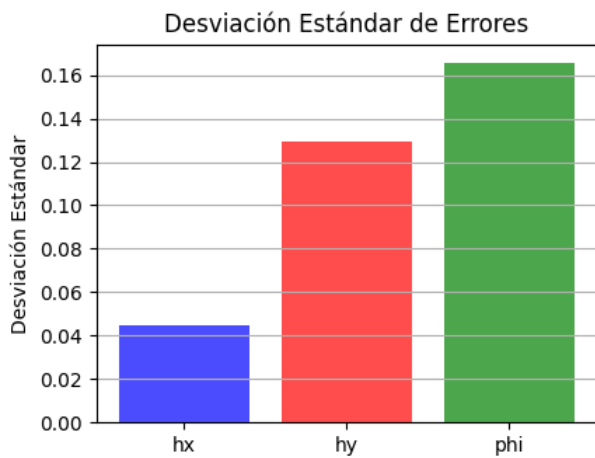


Figura 35. Desviación estándar del sistema – Seguimiento de ruta.



Figura 36. Evolución temporal de la desviación estándar – Seguimiento de ruta.

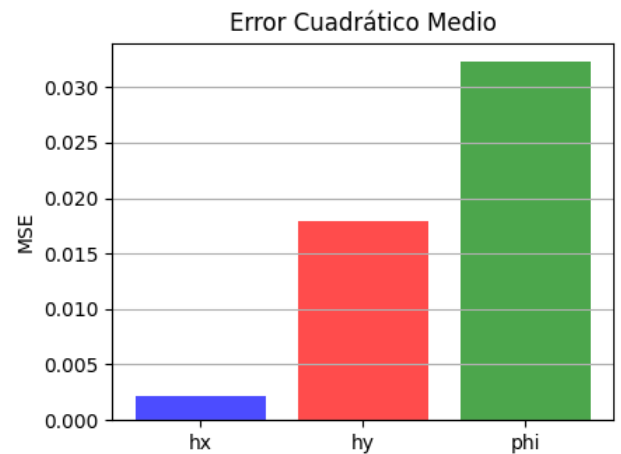


Figura 37. Error cuadrático medio (ECM) – Seguimiento de ruta.

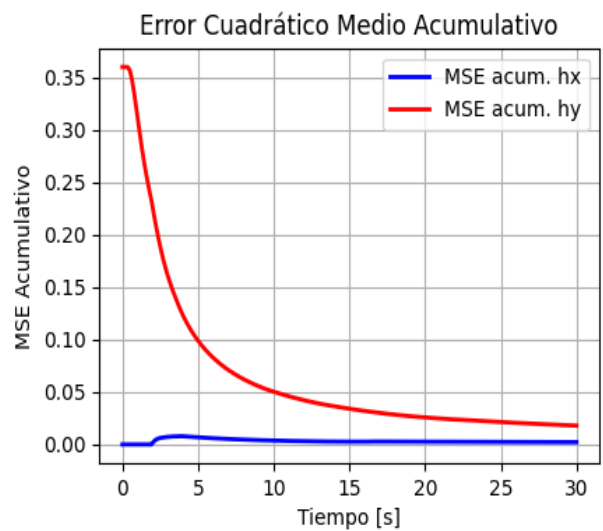


Figura 38. Evolución del error cuadrático medio acumulativo (ECMA) – Seguimiento de ruta.

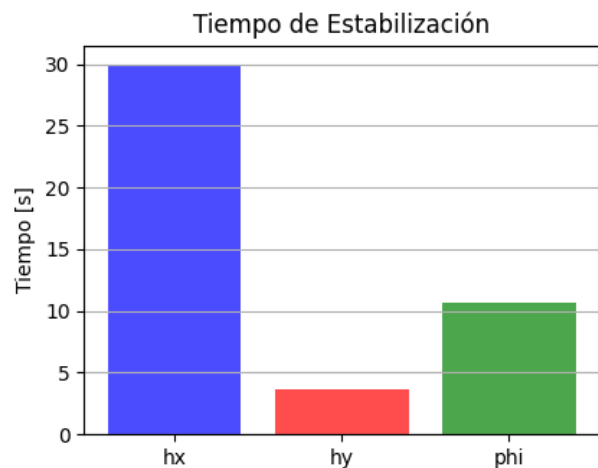


Figura 39. Tiempo de estabilización errores de posición y orientación del robot – Seguimiento de ruta.

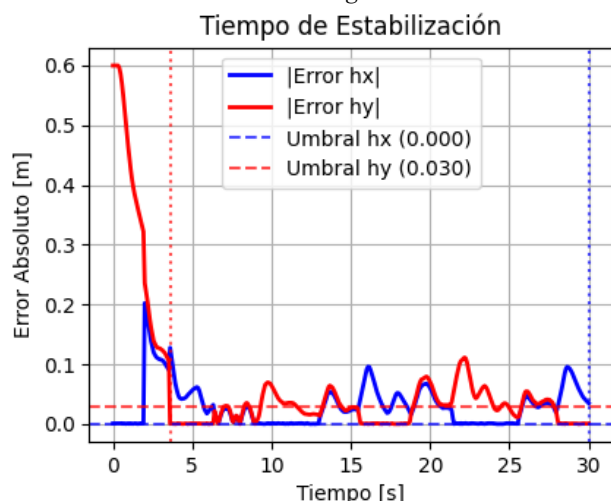


Figura 40. Tiempo de estabilización con umbrales y líneas de referencia – Seguimiento de ruta.

Observamos que el robot consigue completar la ruta trazada, al igual que las otras pruebas experimentales el robot presenta pequeñas oscilaciones las cuales pueden disminuir ajustando el PID y los parámetros de control, observamos en la Figura 32 que el error tiende a 0, nuevamente confirmamos que el algoritmo de control empleado para el seguimiento de camino es correcto y que el sistema es asintóticamente estable. Al final conseguimos obtener resultados satisfactorios al aplicar este algoritmo de control.

Una vez obtenidos los resultados de las pruebas experimentales, comparamos los resultados de las tres pruebas realizadas tomando como referencia el error cuadrático medio.

Tabla 1. Tabla comparativa del error cuadrático medio.

	Control de posición	Seguimiento de trayectoria	Seguimiento de ruta
h_x	0.025	0.002	0.002
h_y	0.025	0.002	0.002
ϕ	0.25	0.014	0.030

En el **control de posición** se observó un desempeño aceptable, pero el error en la orientación revela que el robot presenta dificultad para mantener el ángulo fijo durante el desplazamiento.

En el **seguimiento de trayectoria** el robot tuvo el mejor desempeño global, presentando errores muy bajos en traslación y orientación, lo cual demuestra que se obtienen resultados estables en trayectorias suaves y continuas.

Por último, en el **seguimiento de ruta** el robot presentó una traslación precisa, como en el círculo, pero el error angular se incrementó debido a los cambios bruscos que el robot realiza en cada vértice de la ruta.

Analizando los comportamientos del robot, concluimos que este se adapta mejor a trayectorias continuas que a movimientos puntuales.

Conclusiones

El desarrollo del robot omnidireccional de tres ruedas permitió comprobar en la práctica los algoritmos de control diseñados para control de posición, seguimiento de trayectoria y seguimiento de camino. El robot respondió de manera correcta a las referencias planteadas, logrando los objetivos principales de desplazamiento y control.

Durante las pruebas se observaron algunas oscilaciones en la respuesta, sobre todo en momentos de cambio brusco de dirección o ajustes de velocidad. Estos problemas se presentan por las restricciones físicas del prototipo, las cuales pueden ser resueltas mejorando la calidad de las ruedas, sin sacrificar el bajo costo de producción.

Estas variaciones no impidieron el correcto funcionamiento del sistema y sirvieron para validar la firmeza de los controladores aplicados. Las gráficas obtenidas muestran cómo el robot siguió la referencia y compensó errores a lo largo del tiempo.

En general, los resultados confirman que los algoritmos implementados cumplen con lo esperado en cada aplicación. Además, la plataforma desarrollada se perfila como una herramienta de bajo costo y reproducible para ensayar estrategias de control más avanzadas, lo que abre la posibilidad de trabajar en aspectos como la eficiencia energética y la robustez en entornos no estructurados en investigaciones futuras, así como la implementación de sensores o un control libre, usando LiDAR para graficar un entorno en 2D, o implementar un robot autónomo móvil capaz de esquivar obstáculos.

Referencias

- [1] Lu, Z., Ren, Y., Liu, C., Chen, S., & Zhao, Y. (2025). Design, analysis, and flexibility evaluation of a double-mode underactuated coupled-drive omnidirectional mobile robot. *Journal of Field Robotics*. <https://doi.org/10.1002/rob.22574>
- [2] Taheri, H., & Zhao, C. (2020). Omnidirectional mobile robots, mechanisms and navigation approaches. *Mechanism and Machine Theory*, 153, 103958. <https://doi.org/10.1016/j.mechmachtheory.2020.103958>
- [3] Sato, Y., Kanada, A., & Mashimo, T. (2023). A palm-sized omnidirectional mobile robot driven by 2-DOF torus wheels. *IEEE Robotics and Automation Letters*, 8(1), 105–112. <https://doi.org/10.1109/LRA.2022.3223296>
- [4] Ye, C., Ma, S., & Hui, L. (2011). An omnidirectional mobile robot. *Science China Information Sciences*, 54(12), 2631–2638. <https://doi.org/10.1007/s11432-011-4275-y>
- [5] Tanaka, T., Ozaki, K., & Miyagusuku, R. (2023). Development of an omnidirectional mobile robot. *Proceedings of the JSME Annual Conference on Robotics and Mechatronics (Robomec)*. <https://doi.org/10.1299/jsmermd.2023.1a1-b10>
- [6] Zou, J., Lin, Y., Ji, C., & Yang, H. (2018). A reconfigurable omnidirectional soft robot based on caterpillar locomotion. *Soft Robotics*, 5(2), 164–174. <https://doi.org/10.1089/soro.2017.0008>
- [7] Qian, N., Bisoyi, H., Wang, M., Huang, S., Liu, Z., Chen, X.-M., Hu, J., Yang, H., & Li, Q. (2023). A visible and near-infrared light-fueled omnidirectional twist-bend crawling robot. *Advanced Functional Materials*, 33(15), 2214205. <https://doi.org/10.1002/adfm.202214205>
- [8] Wu, S., Ze, Q., Dai, J., Udipi, N., Paulino, G., & Zhao, R. (2021). Stretchable origami robotic arm with omnidirectional bending and twisting. *Proceedings of the National Academy of Sciences*, 118(44), e2110023118. <https://doi.org/10.1073/pnas.2110023118>
- [9] Yao, S., Cheng, X., Liu, Z., & Kan, J. (2024). An omnidirectional screw-driven forestry robot. *Journal of Field Robotics*, 42(2), 257–271. <https://doi.org/10.1002/rob.22408>
- [10] Xu, Y., Li, H., Yu, F., Zuo, Q., Xie, F., Xie, X., & He, K. (2023). A novel omnidirectional swimming robot with articulated-compliant legs. *IEEE Robotics and Automation Letters*, 8(10), 5664–5671. <https://doi.org/10.1109/LRA.2023.3292077>
- [11] Mamun, M. A. A., Nasir, M., & Khayyat, A. (2018). Embedded system for motion control of an omnidirectional mobile robot. *IEEE Access*, 6, 6722–6739. <https://doi.org/10.1109/ACCESS.2018.2794441>
- [12] Masmoudi, M., Masmoudi, N. K., Masmoudi, M., & Derbel, N. (2016). Fuzzy logic controllers design for omnidirectional mobile robot navigation. *Applied Soft Computing*, 49, 901–919. <https://doi.org/10.1016/j.asoc.2016.08.057>
- [13] Azizi, M. R., Rastegarpanah, A., & Stolkin, R. (2021). Motion planning and control of an omnidirectional mobile robot in dynamic environments. *Robotics*, 10(1), 48. <https://doi.org/10.3390/robotics10010048>
- [14] Wu, X., & Huang, Y. (2021). Adaptive fractional-order non-singular terminal sliding mode control based on fuzzy wavelet neural networks for omnidirectional mobile robot manipulator. *ISA Transactions*, 120, 96–108. <https://doi.org/10.1016/j.isatra.2021.03.035>
- [15] Sprunk, C., Lau, B., Pfaff, P., & Burgard, W. (2016). An accurate and efficient navigation system for omnidirectional robots in industrial environments. *Autonomous Robots*, 41(2), 473–493. <https://doi.org/10.1007/s10514-016-9557-1>
- [16] Watanabe, K. (1998). Control of an omnidirectional mobile robot. *Proceedings of the 1998 Second International Conference on Knowledge-Based Intelligent Electronic Systems (KES'98)*, 1, 51–60. <https://doi.org/10.1109/KES.1998.725827>
- [17] Menegatti, E., Pretto, A., Scarpa, A., & Pagello, E. (2006). Omnidirectional vision scan matching for robot localization in dynamic environments. *IEEE Transactions on Robotics*, 22(3), 523–535. <https://doi.org/10.1109/TRO.2006.875495>
- [18] Zhang, Z., Wang, Z., Zhou, Z., Li, H., Zhang, Q., Zhou, Y., Li, X., & Liu, W. (2023). Omnidirectional continuous movement method of dual-arm robot in a space station. *Sensors*, 23(11), 5025. <https://doi.org/10.3390/s23115025>
- [19] Liu, P., Ma, S., Liu, S., Li, Y., & Li, B. (2022). Omnidirectional jump control of a locust-computer hybrid robot. *Soft Robotics*, 10(1), 48–57. <https://doi.org/10.1089/soro.2021.0137>
- [20] Micolini, O., Ventre, L. O., Coutinho, A., Malatini, H., & Ayme, R. E. (2022). Hermes III: Robot omnidireccional con capacidad de SLAM. *Revista de la Facultad de Ciencias Exactas, Físicas y Naturales*, 9(1), 31–38.
- [21] Al-Ammri, A. S., & Ahmed, I. (2010). Control of omnidirectional mobile robot motion. *Al-Khwarizmi Engineering Journal*, 6(4), 1–9.