

UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

**CONTROL DE TRAYECTORIA PARA LA NAVEGACIÓN DE UN ROBOT
OMNIDIRECCIONAL MEDIANTE EL USO DE OBSERVADORES DE ESTADO**

Tesis para obtener el grado de
Maestro en Robótica

Presenta:

Ing. Alfredo Enrique Feria de la Rosa

Director de Tesis:

Dr. Jorge Luis Barahona Avalos

Co-director de Tesis:

Dr. Oscar David Ramírez Cárdenas

H. Cd. de Huajuapán de León, Oaxaca, México; 8 de enero de 2026.

Dedicatoria

Con profundo cariño a mi familia y a mi novia, Perla, por su amor, paciencia y apoyo incondicional a lo largo de este camino.

Agradecimientos

A mi pilar y soporte incondicional: Mi madre, por ser el faro que me guía. Tu apoyo incondicional, tus ánimos constantes y tu inquebrantable fe en mí han sido la principal motivación para seguir adelante y alcanzar esta meta.

A mi amada novia Perla, mi más grande soporte a lo largo de todos estos años. Este logro es tan tuyo como mío. Tu comprensión, tu infinita paciencia y el cariño que me has brindado en cada momento han sido esenciales para culminar este camino.

A mis queridos hermanos, Karla y Ernesto. A pesar de las dificultades y los retos que enfrentamos en estos años, nunca se apartaron de mi lado. Que sepan que mi apoyo y afecto hacia ustedes serán siempre recíprocos e incondicionales.

Al Dr. Jorge Luis Barahona Avalos, mi director de tesis, por su invaluable guía durante la realización de este trabajo, su paciencia y por compartir generosamente su vasto conocimiento.

Al Dr. Óscar David Ramírez Cárdenas, mi co-director, por su dedicación, su acompañamiento constante en este proceso y por convertirse en una persona a quien tengo en la más alta estima y amistad.

A mis sinodales, el Mtro. Felipe Santiago Espinosa, el Dr. Edgardo Yescas Mendoza, el Dr. Juan Javier Montesinos García y el Dr. Jesús Linares Flores. Agradezco profundamente la huella que han dejado sus fructíferas enseñanzas, tanto dentro del aula como a lo largo del desarrollo de esta investigación.

A la Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI), por el fundamental apoyo económico brindado (Becario No. 1314419) durante mis estudios de maestría, sin el cual la realización de este trabajo de tesis no hubiera sido factible.

Y por último, a mí mismo. Por la convicción de creer en mi potencial, por la resiliencia de no ceder ante las adversidades y por la audacia de atreverme a ir más allá para la consecución de este objetivo.

Suéñalo, desealo, consíguelo...

Resumen

Se presenta el desarrollo y la validación de un robot móvil omnidireccional de tres ruedas orientado a aplicaciones en entornos con espacio restringido. El trabajo aborda dos limitaciones frecuentes en robótica móvil: la maniobrabilidad reducida de plataformas diferencial/Ackermann y la degradación del seguimiento de trayectorias ante perturbaciones y errores de estimación de posición. Como solución, se implementa un esquema de control basado en Control por Rechazo Activo de Perturbaciones (ADRC) para la regulación de velocidad de las ruedas, acoplado a un controlador cinemático de posición para seguimiento de trayectorias en el plano.

La contribución técnica incluye el desarrollo de la plataforma experimental, el diseño e implementación del controlador ADRC en hardware embebido, el procedimiento de sintonía y las consideraciones prácticas de discretización y muestreo, y la comparación experimental frente a un controlador PID representativo. Los resultados muestran que ADRC mejora la precisión de seguimiento y la robustez respecto de PID, reduciendo la sensibilidad ante perturbaciones y errores de modelado. Estos hallazgos respaldan el uso de ADRC como reemplazo viable del PID en plataformas omnidireccionales de bajo costo, y sientan bases para su extensión a esquemas de estimación más avanzados y a tareas de navegación de mayor complejidad.

Índice

Índice de figuras	XIII
Índice de tablas	XVII
1. Introducción	1
1.1. Antecedentes	2
1.2. Planteamiento del problema	5
1.3. Justificación	7
1.3.1. Pertinencia	8
1.3.2. Relevancia	9
1.4. Hipótesis	9
1.5. Objetivos	10
1.5.1. Objetivo general	10
1.5.2. Objetivos específicos	10
1.6. Alcances y delimitaciones	11

1.6.1.	Alcances	11
1.6.2.	Delimitaciones	11
1.7.	Metodología	12
2.	Marco teórico	15
2.1.	Robots móviles	15
2.1.1.	Robots terrestres	16
2.1.2.	Robots con ruedas	17
2.1.3.	Robot omnidireccional con ruedas universales	18
2.1.4.	Ventajas de los robots omnidireccionales de tres ruedas	19
2.1.5.	Ruedas especiales u omnidireccionales	19
2.1.6.	Odometría del robot omnidireccional de tres ruedas	22
2.2.	Control por Rechazo Activo de Perturbaciones	26
2.3.	Observadores de estado	27
2.3.1.	Observador Proporcional Integral Generalizado (GPIO)	27
3.	Desarrollo del robot omnidireccional	31
3.1.	Diseño mecánico	31
3.1.1.	Objetivos de diseño	31
3.1.2.	Diseño CAD	32
3.1.3.	Fabricación del chasis	34
3.2.	Diseño electrónico	35

ÍNDICE

3.2.1. Componentes electrónicos	35
3.3. Construcción	39
3.3.1. Ensamblaje del robot	39
4. Control cinemático	43
4.1. Modelado y control cinemático del robot	43
4.2. Diseño del control cinemático	45
4.3. Control de trayectoria	47
4.4. Exportación del modelo a CoppeliaSim	48
4.5. Resultados de simulación	49
4.5.1. Control de posición	50
4.5.2. Control de trayectoria	52
4.6. Resultados de la implementación	53
5. Diseño de controladores	71
5.1. Diseño del controlador ADRC y observador GPI	71
5.1.1. Análisis de estabilidad del observador	77
5.2. Simulación del control ADRC y el observador GPI	80
5.3. Diseño y simulación del control PID	85
5.4. Comparación de resultados de la simulación	88
6. Implementación del control ADRC y el observador GPI en el robot	89

6.1. Discretización e implementación en el microcontrolador	90
6.1.1. Filtro pasa bajas de primer orden	92
6.2. Resultados de la implementación	93
7. Discusión y conclusiones	109
7.1. Discusión	109
7.2. Conclusiones	111
7.3. Trabajo futuro	112
Referencias	116
Anexos	116
A.1. Diagrama de conexiones	119
B.2. Caracterización del motor de CD	120
B.2.1. Obtención de R_a	120
B.2.2. Obtención de L_a	121
B.2.3. Obtención de K_e	121
B.2.4. Obtención de J	123
B.2.5. Obtención de B	124
C.3. Simulación del motor de CD	126
D.4. Sintonización del controlador PID	129
E.5. Herramientas de software	131

Índice de figuras

1.1. Modelo en V	13
2.1. Ejemplo de un robot omnidireccional con tres ruedas universales	18
2.2. Cuatro diseños de ruedas universales	20
2.3. Rueda mecanum o sueca	20
2.4. Rueda esférica activa	21
2.5. Robot omnidireccional de tres ruedas universales	22
3.1. Vistas del robot móvil	33
3.2. Simulación del mecanizado de las piezas en el software VISI®	34
3.3. Microcontrolador ESP32 WROOM	35
3.4. Motor de CD con encoder	36
3.5. Módulo controlador de motores VNH2SP30	37
3.6. Batería LiPo	38
3.7. Módulo regulador de voltaje LM2596	38
3.8. Diagrama de conexiones de los componentes electrónicos	40

3.9. Placa PCB diseñada.	41
3.10. Robot ensamblado	41
4.1. Representación del robot omnidireccional en el plano cartesiano	44
4.2. Vistas del robot en el simulador	49
4.3. Control de posición en el punto deseado en CoppeliaSim [®]	50
4.4. Posición del robot en los ejes X, Y	51
4.5. Orientación del robot respecto al eje Z	51
4.6. Trayectoria seguida por el robot	52
4.7. Diagrama del funcionamiento del sistema	54
4.8. Control de posición en el punto deseado	57
4.9. Trayectoria realizada por el robot	58
4.10. Comparativa de las velocidades frontal y lateral	59
4.11. Posición en x e y durante el desplazamiento	60
4.12. Orientación del robot durante el desplazamiento.	60
4.13. Comparativa de las trayectorias	63
4.14. Comparativa de las velocidades frontal y lateral	64
4.15. Posición del robot durante el seguimiento	65
4.16. Orientación del robot durante el seguimiento	66
4.17. Error de posición a lo largo de la trayectoria	67
4.18. Error euclidiano de posición a lo largo de la trayectoria	67

ÍNDICE DE FIGURAS

5.1. Diagrama de bloques del sistema en lazo cerrado	81
5.2. Velocidades vs referencia con controlador ADRC y GPIO	82
5.3. ISE del Motor 1 con el controlador ADRC	83
5.4. Velocidades vs referencia suavizada	84
5.5. Diagrama de bloques del controlador PID	86
5.6. Velocidades vs referencia con controlador PID	86
5.7. ISE del Motor 1 con el controlador PID	87
5.8. Comparación del ISE de ambos controladores	88
6.1. Trayectoria realizada por el robot	94
6.2. Comparativa de las velocidades frontal y lateral	95
6.3. Posición en x e y durante el desplazamiento	96
6.4. Orientación del robot durante el desplazamiento	96
6.5. Comparativa de las velocidades medidas y estimadas con el GPIO	97
6.6. Comparativa de las trayectorias	100
6.7. Comparativa de las velocidades frontal y lateral	101
6.8. Posición del robot durante el seguimiento	102
6.9. Orientación del robot durante el seguimiento	102
6.10. Error de posición a lo largo de la trayectoria	103
6.11. Error euclidiano de posición a lo largo de la trayectoria	104
6.12. Comparativa de las velocidades medidas y estimadas con el GPIO	105

1.	Diagrama de bloques del sistema dinámico en lazo abierto.	127
2.	Corrientes de los motores en lazo abierto.	128
3.	Velocidades de los motores en lazo abierto.	128

Índice de tablas

3.1. Características del motor JGB37-520.	36
3.2. Características del módulo VNH2SP30-E.	37
3.3. Piezas utilizadas en el ensamble.	39
4.1. Entorno de desarrollo.	56
4.2. Desempeño del controlador PID en control cinemático de posición.	61
4.3. Desempeño del controlador PID en seguimiento de trayectorias.	68
6.1. Desempeño del controlador ADRC en control cinemático de posición.	98
6.2. Desempeño del controlador ADRC en seguimiento de trayectorias.	106
1. Valores de inductancia en cada motor.	121
2. Parámetros de cada motor de CD	122
3. Valores de B para cada motor.	125

Introducción

La robótica móvil es un campo en constante evolución que desempeña un papel fundamental en la automatización de diversas aplicaciones industriales y de servicio. Los robots móviles omnidireccionales u holonómicos, con sus capacidades de movimiento altamente versátiles, se han convertido en un objeto de estudio importante tanto a nivel de investigación como de desarrollo de sistemas robóticos avanzados. Estos robots, equipados con ruedas omnidireccionales, tienen la capacidad de moverse en cualquier dirección sin necesidad de cambiar de orientación, lo que los hace ideales para tareas que requieren maniobras precisas y rápidas [1].

El control de trayectoria en robots móviles omnidireccionales es un tema de investigación crítico, ya que afecta directamente la eficiencia y la precisión de sus movimientos. La habilidad de planificar y seguir trayectorias de manera precisa es esencial para una amplia gama de aplicaciones, como en el transporte de mercancías y la logística de almacenes automatizados [2, 3]. Además, este control de trayectoria desempeña un papel fundamental en la interacción segura y efectiva entre humanos y robots en entornos colaborativos.

1.1. Antecedentes

Los avances en la investigación de robots móviles con capacidad para desplazarse eficientemente en espacios reducidos han sido una línea de estudio que se remonta a varias décadas atrás. En particular, un hito significativo tuvo lugar en 1972 con la introducción de las ruedas *mecanum*. Estas innovadoras ruedas otorgan a los robots la habilidad de movilizarse en cualquier dirección sin necesidad de girar, marcando un quiebre con la convencionalidad de los vehículos terrestres [1].

Con la incorporación de las ruedas omnidireccionales en el panorama de la robótica móvil, los investigadores se vieron impulsados a desarrollar modelos cinemáticos que describieran de manera precisa el movimiento de estos robots. Simultáneamente, se inició un esfuerzo constante por perfeccionar el diseño de estas ruedas, reconociendo su papel fundamental en la movilidad mejorada de los robots [4].

Simultáneamente, se produjo un avance significativo en las técnicas de control, buscando no solo la estabilidad del sistema sino también la estimación precisa de parámetros desconocidos. Este progreso fue esencial para aplicaciones que demandan una precisión excepcional, como el transporte de mercancías, donde la robustez del sistema se convierte en un requisito fundamental.

El seguimiento de trayectorias en robots móviles ha sido abordado mediante diversas técnicas de control a lo largo del tiempo, abarcando desde métodos tradicionales como el control cinemático y odometría hasta enfoques más avanzados, como el uso de cámaras y sistemas de inspección visual para una posición precisa.

La implementación del control Proporcional Integral Derivativo (PID) ha sido una práctica común en el campo, ofreciendo un equilibrio entre simplicidad y eficacia para el seguimiento de tra-

1. Introducción

yectorias. Aunque el controlador PID ha sido ampliamente empleado para regular la velocidad de motores en robots móviles, su eficacia puede estar comprometida en escenarios que presentan perturbaciones o variaciones en los parámetros del sistema. En estudios como el presentado por Nanda *et al.* [5] han mostrado resultados significativos mediante la implementación de controladores PI para la regulación de la velocidad en motores de corriente directa, aplicados a un robot limpiador de pisos. Dicho estudio demuestra que el controlador estabiliza eficazmente la rotación del motor, optimizando la eficiencia en la operación, y destaca la adaptabilidad del método a robots de mayor escala y a distintos sistemas de motores, tanto de corriente continua como alterna. Asimismo, Mérida *et al* [6] se describe un sistema avanzado de control de procesos. Este sistema combina un controlador PID con un prefiltro, un predictor de Smith, un esquema anti-saturación y un compensador para la fricción de Coulomb. La contribución principal radica en el diseño del esquema de control de motores y en el método de ajuste de los parámetros de los controladores. Su implementación en un robot móvil compacto y de bajo costo ha demostrado, mediante experimentos, mejoras sustanciales en el desempeño del seguimiento, evidenciando el potencial del sistema para abordar las limitaciones típicas en robots económicos. Jiang y Song [3] proponen un esquema de control y planificación de trayectorias punto a punto basado en planitud diferencial para alcanzar la posición y orientación finales deseadas y también hacer frente al problema del movimiento suave en un espacio de movimiento limitado. Además, técnicas más complejas, como la fusión sensorial, han sido empleadas para mejorar la precisión en el seguimiento de trayectorias. Romadon *et al.* [7], proponen una combinación de odometría, sensores inerciales y procesamiento de imágenes, destacando la importancia de integrar información de múltiples fuentes para una navegación precisa.

Otro de los métodos de control que han sido propuestos es el método de control backstepping adaptativo como el presentado por Huang y Tsai [8] para el seguimiento y estabilización de la trayectoria de un robot móvil omnidireccional, esto con la finalidad de estimar las variables desconocidas

del sistema que generan el deslizamiento de las ruedas.

Asimismo, otras investigaciones, como la documentada por Byoung-Suk *et al.* [9], exploran soluciones innovadoras, como el uso de sistemas de Identificación por Radio Frecuencia (RFID), para determinar la posición del robot. Este enfoque se muestra particularmente útil en entornos estructurados, como pasillos provistos de identificadores RFID, ofreciendo una alternativa efectiva para el seguimiento de trayectorias en condiciones específicas.

Santos *et al.* [10] implementan un enfoque de control predictivo con filtro de Smith, con el objetivo específico de minimizar tanto el error de posición del robot como el efecto del ruido en las mediciones.

Además, en la investigación de Mehrez *et al.* [11], se utiliza una técnica de Control Predictivo por Modelo (MPC) con el objetivo de garantizar la estabilidad asintótica de un punto de referencia deseado para un robot móvil holonómico. Rodríguez *et al.* [12] proponen un control de trayectoria sin medición de velocidad de un robot móvil omnidireccional mediante un controlador basado en pasividad y un observador diseñado mediante la técnica de inmersión e invariancia.

Otra de las alternativas utilizadas para conocer la posición del robot es a través de sistemas Optitrack, como ilustra Garrido [13]. En este estudio, se implementa un algoritmo SLAM (Simultaneous Localization And Mapping) y se utiliza un sistema Optitrack para adquirir información sobre la posición del robot, así como para su control.

Szayer *et al.*[14] plantean un método genérico, aplicable a cualquiera de las configuraciones mecánicas, de bajo coste computacional para utilizar la velocidad y aceleración máximas de un robot móvil omnidireccional basado en los modelos cinemático y dinámico inversos.

1. Introducción

Sin embargo, la precisión de la estimación de la posición puede verse comprometida en terrenos irregulares [7, 9, 15]. Los sensores acoplados a las ruedas (encoders) utilizados para conocer la posición del robot pueden enfrentar dificultades, como lo es el deslizamiento de las ruedas omnidireccionales y, consecuentemente, errores en la estimación de la posición. Este desafío resalta la necesidad de abordar cuestiones específicas relacionadas con la navegación y la precisión en entornos dinámicos y variados.

Para abordar estas dificultades, se han desarrollado técnicas de control avanzadas, como el control por rechazo activo de perturbaciones (ADRC), que permite mejorar la robustez y el desempeño del sistema en condiciones no ideales. En estudios como el de Leal-Ramos *et al.* [16], se presenta una plataforma experimental orientada al desarrollo de algoritmos en robótica, control y aprendizaje. Esta plataforma es validada mediante la implementación de un controlador ADRC asistido por un observador de estado extendido (ESO), el cual permite estimar perturbaciones internas y externas del robot, todo ello en una arquitectura abierta basada en ROS. Los resultados de dicho estudio muestran que los algoritmos de control propuestos son livianos y adecuados para su implementación en sistemas embebidos con recursos computacionales limitados. Por su parte, Cortez *et al.* [17] implementaron un controlador ADRC para regular la velocidad de un servomecanismo. Este controlador fue evaluado experimentalmente y comparado con un controlador proporcional-integral (PI), evidenciando que no requiere un modelo matemático preciso del sistema, lo cual simplifica su aplicación y mejora la eficiencia en términos de consumo de corriente respecto al controlador PI.

1.2. Planteamiento del problema

En la actualidad, la investigación en automática y robótica ha puesto un énfasis significativo en la robótica móvil debido a sus amplias aplicaciones en la industria. Este enfoque ha generado

la necesidad de emplear técnicas de control modernas para evaluar el rendimiento de los robots móviles en sus tareas asignadas. Las técnicas tradicionales se han vuelto menos confiables debido a la naturaleza compleja y dinámica de estos sistemas.

La mayoría de los estudios en el campo de la robótica móvil se centran en la configuración del tipo diferencial y Ackerman. Sin embargo, estas configuraciones presentan una desventaja principal, que es su baja maniobrabilidad en espacios reducidos. Para cambiar de dirección o sentido, este tipo de configuración requiere una reorientación respecto a su eje central, lo cual limita considerablemente la cantidad de aplicaciones y tareas que se pueden desarrollar tanto en el ámbito industrial como en el de investigación.

La implementación de robots móviles suele enfrentar el desafío del error en la estimación de la posición, originado por la naturaleza no lineal de estos sistemas, así como por factores adicionales como la fricción, las vibraciones y el deslizamiento entre las ruedas y el terreno [18]. Usualmente, se utiliza la odometría basada en la información extraída de los encoders acoplados a las ruedas para abordar este problema. Sin embargo, la confiabilidad de estos datos está directamente relacionada con la rugosidad del terreno en el que se desplaza el robot móvil. Cualquier perturbación o anomalía en el terreno puede provocar la pérdida de pulsos durante la lectura del encoder [7, 9, 15].

Incluso al trabajar en una superficie completamente plana, el problema persiste debido a las limitaciones de los sensores de cuadratura que no son capaces de detectar completamente el desplazamiento que pueden sufrir las ruedas omnidireccionales [19].

1.3. Justificación

La precisión en la estimación de la posición de plataformas móviles es crucial para su desempeño eficiente y confiable. Tradicionalmente, la dependencia exclusiva de encoders ha sido una práctica común, sin embargo, esta metodología presenta limitaciones significativas. La variabilidad en el terreno, ya sea rugoso o liso, puede inducir lecturas erróneas, comprometiendo la exactitud del sistema de posicionamiento.

Es esencial reconocer que la confianza plena en encoders puede ser insuficiente, especialmente en entornos irregulares. Aquí es donde se destaca la relevancia de implementar técnicas de Control por Rechazo Activo de Perturbaciones (ADRC, por sus siglas en inglés: Active Disturbance Rejection Control) y la integración de observadores de estados.

La exploración de otras alternativas para conocer la posición del robot también revela desafíos significativos. Métodos como sistemas optitrack, sensores láser o cámaras de tres dimensiones (RGB-D) pueden ser efectivos, pero su aplicabilidad se ve restringida por el costo y limitaciones espaciales, especialmente en entornos reducidos. Otra opción que se presenta es el sistema de posicionamiento global (GPS), pero su utilidad disminuye drásticamente en interiores y sufre de imprecisiones notables en exteriores.

Por lo tanto, la justificación para esta investigación se fundamenta en la necesidad imperante de superar las limitaciones de las técnicas convencionales de posicionamiento en plataformas móviles. La implementación de técnicas de rechazo activo de perturbaciones y observadores de estados no solo aborda las deficiencias asociadas a la dependencia exclusiva de encoders, sino que también responde a la ineficacia de otras alternativas en entornos específicos. Este trabajo de tesis busca proporcionar una solución robusta y eficiente para mejorar la precisión de la estimación de posi-

ción en plataformas móviles omnidireccionales, abriendo así nuevas posibilidades de aplicación en diversos contextos.

Adicionalmente, la incorporación de observadores en una plataforma robótica ofrece una mayor robustez en la determinación de la ubicación del robot, abordando así uno de los desafíos constantes en el ámbito de la robótica. Esta estrategia busca mejorar la precisión en la estimación de la ubicación, así como también permitir la implementación de controles específicos destinados a optimizar la posición del robot, proporcionando una solución integral para los desafíos inherentes a la navegación autónoma.

1.3.1. Pertinencia

La presente tesis aborda la necesidad de reducir el error de estimación de la posición en un robot omnidireccional de tres ruedas. La precisión en la determinación de la posición es fundamental ya que la movilidad eficiente del robot es esencial para su rendimiento óptimo.

En este contexto, se propone emplear la técnica de control ADRC. Simple y eficaz, el ADRC ofrece una alternativa a la dependencia de encoders acoplados a los motores del robot, presentando además una solución económicamente viable en comparación con otros métodos existentes. Esta elección estratégica no solo busca mejorar la precisión de la estimación de posición, sino también optimizar los recursos necesarios para implementar esta solución.

Al considerar el ADRC, no solo estamos abordando un desafío técnico significativo, sino también adoptando una perspectiva económica que puede resultar crucial en aplicaciones prácticas. En las siguientes secciones, se explorará en detalle la aplicación práctica y los beneficios específicos

1. Introducción

que esta técnica puede ofrecer en el contexto para el robot omnidireccional de tres ruedas.

1.3.2. Relevancia

La investigación propuesta se sitúa en el núcleo de la vanguardia tecnológica en robótica y control automático. En un contexto donde la aplicación de robots móviles se ha expandido significativamente, desde la exploración espacial hasta entornos industriales y domésticos, la implementación eficiente de estrategias de control por rechazo activo de perturbaciones se presenta como un factor crítico. La relevancia de este estudio se evidencia en su capacidad para mejorar la robustez y adaptabilidad de los robots en escenarios del mundo real, contribuyendo así al avance general de la ingeniería robótica y a la solución de problemas prácticos en diversos campos de aplicación.

1.4. Hipótesis

Al comparar la respuesta en el seguimiento de trayectoria de un robot móvil omnidireccional con el uso de control por rechazo activo de perturbaciones, se observará una mejora significativa en la capacidad para corregir y compensar los efectos no deseados ante perturbaciones externas e internas que usando solamente la información de los encoders.

1.5. Objetivos

1.5.1. Objetivo general

Desarrollar un robot móvil omnidireccional de tres ruedas para posteriormente implementar en él una técnica de control ADRC y observadores de estado para la estimación precisa de la posición que permita la navegación y seguimiento de trayectorias.

1.5.2. Objetivos específicos

- Desarrollar una plataforma omnidireccional de tres ruedas que sirva como base para los experimentos y pruebas de control, estableciendo así el entorno físico necesario para la implementación de la técnica de control ADRC.
- Establecer el modelo matemático que describa el comportamiento cinemático del robot, permitiendo su simulación en un entorno virtual para evaluar y ajustar la técnica de control antes de la implementación en la plataforma física.
- Implementar un control cinemático en la plataforma construida para validar su funcionamiento.
- Desarrollar el control ADRC y un observador de estados en Matlab Simulink®, adaptándolos a las particularidades del robot omnidireccional.
- Implementar el control ADRC en la plataforma física, ajustando y optimizando el sistema en caso necesario.

1. Introducción

- Realizar pruebas para evaluar la capacidad del robot para seguir trayectorias predefinidas.
- Analizar los resultados obtenidos durante las pruebas para validar la eficacia del control ADRC y del observador de estados.

1.6. Alcances y delimitaciones

1.6.1. Alcances

Este estudio se enfoca en la implementación y evaluación de la técnica ADRC y observadores de estados para la mejora de la precisión en la estimación de posición de un robot móvil omnidireccional de tres ruedas.

1.6.2. Delimitaciones

- Este estudio se centra en un robot móvil omnidireccional de tres ruedas.
- Se opta por un modelo de tres ruedas debido a su diseño mecánico más simple, mayor eficiencia energética y menor costo computacional para realizar el control de tres motores.
- La investigación se limita a un robot con dimensiones máximas de 50 cm de largo x 50 cm de ancho x 30 cm de alto.
- Los observadores de estados se implementan para obtener las velocidades de cada rueda acoplada a cada motor de CD.

-
- La investigación se basa en el modelo y control cinemático de un robot omnidireccional de tres ruedas.
 - El control ADRC sirve para controlar la velocidad de cada motor de CD.
 - La trayectoria del robot móvil omnidireccional es programada y definida internamente en la placa de control.
 - La implementación a nivel experimental se lleva a cabo en un entorno controlado, es decir, no realiza evasión de obstáculos.

1.7. Metodología

Para la realización del presente trabajo se hace uso de la metodología flexible para el desarrollo de sistemas mecatrónicos propuesta por Gausemeier y Moehringeren [20], la cual se describe a continuación.

- **Requerimientos:** El punto de partida es la identificación y clarificación de los objetivos generales y específicos de la investigación. Los requisitos son los parámetros y condiciones que guiaron la evaluación del éxito de la presente investigación.
- **Diseño del sistema:** En esta etapa se desarrolla la planificación y diseño de la plataforma omnidireccional de tres ruedas y la obtención del modelo matemático para describir el comportamiento cinemático del robot. Además se lleva a cabo la construcción de la plataforma.
- **Diseño específico del dominio:** Se lleva a cabo el desarrollo del control ADRC y observadores de estados, adaptándolos a las características específicas de la plataforma construida.

1. Introducción

- **Integración del sistema:** En esta etapa se implementa y ajusta el control ADRC en la plataforma física, asegurando de que todos los componentes trabajen de forma adecuada.
- **Verificación/validación:** Se realizaron pruebas y evaluaciones para asegurar que el robot se comporte según lo esperado y cumpla con los objetivos establecidos.
- **Modelado y análisis del modelo:** Se realizaron simulaciones en entornos virtuales para evaluar y ajustar estrategias de control antes de la implementación en la plataforma física.
- **Producto:** El producto final fue la capacidad del robot para seguir trayectorias bajo la influencia de perturbaciones provenientes del suelo, validando así la eficacia del control ADRC y del observador de estados.

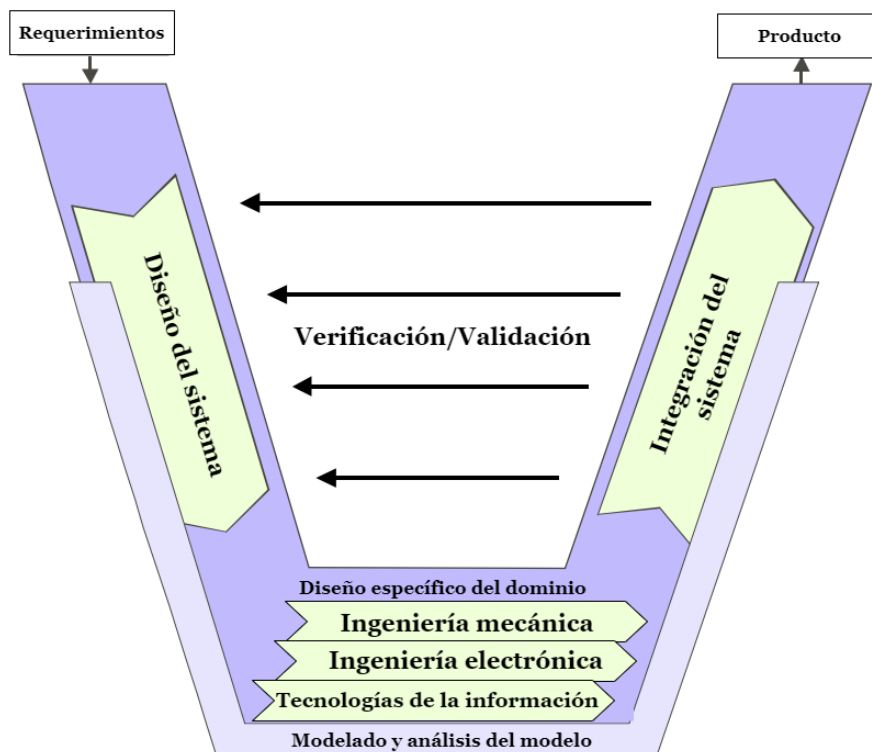


Figura 1.1: Modelo en V. Adaptado de [20].

Tal como ilustra la Figura 1.1, en la práctica, este procedimiento representó una interacción continua de verificación y validación. Su finalidad es asegurar el cumplimiento de los requisitos inicialmente establecidos, así como identificar aspectos adicionales que pudieron no haber sido contemplados al inicio.

Debido a la naturaleza altamente interrelacionada entre los diversos elementos que componen un sistema mecatrónico, como un robot, se optó por esta metodología para orientar el progreso del proyecto.

Marco teórico

En este capítulo se formalizan las ideas centrales relacionadas con los robots móviles omnidireccionales y el control por rechazo activo de perturbaciones. Se introducen y se desarrollan los aspectos teóricos más relevantes para este trabajo de tesis, incluyendo la clasificación de los robots móviles con ruedas, las características específicas de los robots omnidireccionales, y los principios del control por rechazo activo de perturbaciones.

2.1. Robots móviles

El desarrollo de robots móviles responde a la necesidad de extender el campo de aplicación de la robótica, restringido inicialmente al alcance de una estructura mecánica anclada en uno de sus extremos. Se trata también de incrementar la autonomía limitando todo lo posible la intervención humana [21].

De acuerdo con Barrientos *et al.* [22], un robot móvil se define como *un sistema electromecánico capaz de desplazarse de manera autónoma sin estar sujeto físicamente a un solo punto. Posee sensores que permiten monitorear a cada momento su posición relativa a su punto de origen y a su punto de destino. Normalmente su control es en lazo cerrado. Su desplazamiento es proporcio-*

nado mediante dispositivos de locomoción, tales como ruedas, patas, orugas, etc. En ocasiones se hace referencia a los robots móviles como *AGV*, siglas que proceden del término *Autonomous Guided Vehicle* o vehículo guiado autónomamente. Nuevamente, la definición resulta insuficiente para abarcar a todos, pues dentro de los robots móviles se pueden considerar los robots móviles terrestres (denominados *UGV Unmanned Ground Vehicles*), los robots móviles aéreos (conocidos como *UAV Unmanned Aerial Vehicles*) y los robots móviles submarinos (identificados como *AUV Autonomous Underwater Vehicles*), siendo todos ellos en ocasiones, a pesar del nombre, semi-autónomos [23].

2.1.1. Robots terrestres

Los robots móviles terrestres incluyen generalmente los siguientes componentes: plataforma, sensores, sistemas de control, interfaz de guiado, enlaces de comunicación y funcionalidades de integración de sistemas. La plataforma puede basarse en un diseño de vehículo todo terreno e incluye el sistema de locomoción, los sensores y la fuente de energía.

Dentro de las formas de locomoción más comunes se tienen las orugas, las ruedas y las patas. Debido a estos mecanismos de desplazamiento se los suele clasificar en robots móviles con ruedas (*WMR, Wheeled Mobile Robots*), robots móviles con patas (*LMR, Legged Mobile Robots*), robots móviles con orugas (*TMR, Tracked Mobile Robots*) y robots móviles híbridos [23].

En lo referente a las partes de las que se compone un robot móvil con ruedas, se tiene un arreglo cinemático y un sistema de actuadores, que dotan de movimiento a la estructura cinemática. Ambos sistemas están íntimamente ligados y son dignos de estudiarse en conjunto [22].

2. Marco teórico

2.1.2. Robots con ruedas

Los vehículos con ruedas son la solución más simple y eficiente para conseguir la movilidad en terrenos suficientemente duros y libres de obstáculos, permitiendo conseguir velocidades relativamente altas, además de que mediante diferentes tipos de configuración se les confieren características y propiedades diferentes respecto a la eficiencia energética, dimensiones, cargas útiles y maniobrabilidad.

A continuación se comentan brevemente las características más significativas de las configuraciones más comunes en robots móviles con ruedas [21]:

Ackerman: Esta configuración es comúnmente empleada en vehículos convencionales de cuatro ruedas. De hecho, muchos robots móviles diseñados para operar en exteriores se desarrollan a partir de la adaptación de automóviles u otros vehículos de mayor tamaño. No obstante, la principal desventaja del sistema de locomoción tipo Ackerman radica en su limitada capacidad de maniobra.

Triciclo: En esta configuración, la rueda delantera cumple una doble función: proporciona la tracción y permite el direccionamiento del vehículo. El eje trasero, compuesto por dos ruedas laterales, actúa de manera pasiva, permitiendo que sus ruedas giren libremente. Aunque esta disposición ofrece una maniobrabilidad superior en comparación con la configuración Ackerman, puede presentar dificultades de estabilidad al operar sobre terrenos irregulares. Gracias a su simplicidad constructiva, esta configuración es comúnmente utilizada en vehículos robóticos destinados tanto a interiores como a exteriores con superficies pavimentadas.

Diferencial: El direccionamiento viene dado por la diferencia de velocidades de las ruedas laterales, las cuales también proporcionan la fuerza de tracción. Además, se incorporan una o más ruedas auxiliares que cumplen funciones de soporte estructural. Adicionalmente, existen una

o más ruedas para soporte. Esta configuración es la más frecuente en robots para interiores.

Omnidireccional: Los vehículos omnidireccionales ofrecen la máxima capacidad de maniobra. Este tipo de robot móvil puede desplazarse de forma simultánea e independiente a lo largo de los ejes del plano cartesiano, además de girar alrededor del eje perpendicular al plano, lo que le permite ejecutar movimientos complejos sin necesidad de reorientar su estructura.

2.1.3. Robot omnidireccional con ruedas universales

Un robot omnidireccional, también conocido como robot holonómico, es un robot móvil cuya configuración le permite moverse en cualquier dirección sin la necesidad de modificar previamente la orientación del chasis (máxima maniobrabilidad), es decir, que tiene la capacidad de generar movimientos en cualquiera de las componentes del plano, ya sean traslaciones (hacia adelante, laterales, en reversa) o rotaciones, sin necesidad de reorientarse, claro está que esto conlleva un mayor grado de complejidad en el proceso de control.

Un robot móvil omnidireccional debe contar con al menos tres ruedas activas para ser considerado un vehículo omnidireccional holonómico (ver Figura 2.1). Por lo común, exhiben una configuración mecánica de tres o cuatro ruedas.

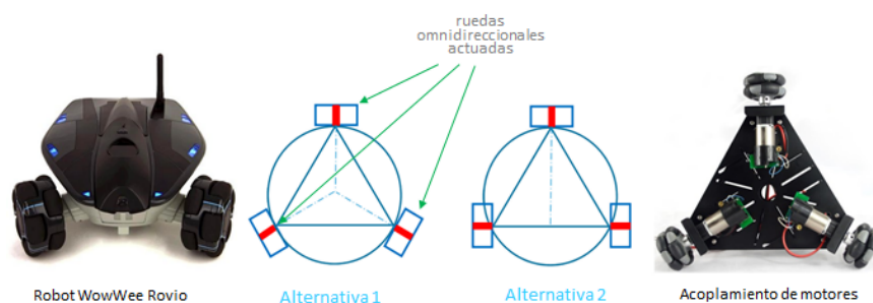


Figura 2.1: Ejemplo de un robot omnidireccional con tres ruedas universales. Recuperado de [23].

2. Marco teórico

Un inconveniente de la rueda universal es que hay una vibración vertical debido a los contactos discontinuos durante el movimiento. Para resolver este problema, se han propuesto una variedad de diseños mecánicos. Otro inconveniente es su durabilidad relativamente baja en comparación con los neumáticos convencionales [23].

2.1.4. Ventajas de los robots omnidireccionales de tres ruedas

Una de las principales ventajas de utilizar ruedas universales es que se pueden construir fácilmente robots móviles omnidireccionales con lo que se logra máxima maniobrabilidad en el plano, esto significa que el robot puede moverse en cualquier dirección sin necesidad de reorientarse, en contraste, con los otros tipos de robots vistos hasta el momento que tienen una maniobrabilidad restringida.

Dado que se pueden construir robots omnidireccionales sin utilizar módulos de ruedas de dirección activa, las estructuras mecánicas de las piezas de accionamiento pueden tener estructuras más simples en comparación a los robots con 4 ruedas. Sin embargo, el diseño mecánico de una rueda se vuelve un poco complicado.

2.1.5. Ruedas especiales u omnidireccionales

Este tipo de ruedas poseen estructuras mecánicas únicas que incluyen rodillos o esferas, que permiten además del giro convencional también el giro lateral, es decir que no restringen el movimiento del robot en comparación con las ruedas convencionales. Estas ruedas están diseñadas de manera que tengan tracción activa en una dirección y movimiento pasivo en otra, permitiendo así una mayor maniobrabilidad en entornos congestionados. En el sistema de locomoción de un robot móvil se emplean como simples ruedas libres (con mejor comportamiento que las ruedas locas con-

vencionales) o también como ruedas motrices. Tenemos tres tipos principales de ruedas especiales: universal, mecanum y esférica [23].

Rueda Universal: Su diseño combina movimientos restringidos y libres durante el giro. Está compuesta por pequeños rodillos pasivos en torno a su diámetro exterior que están acoplados perpendicularmente al eje de rotación de la rueda ($\gamma = 90^\circ$) como se muestra en la Figura 2.2. Los rodillos pasivos pueden girar libremente alrededor del eje de rotación, lo que da como resultado un movimiento lateral de la rueda. Por lo cual, la rueda puede desplazarse en la dirección paralela al eje de la rueda además de rotar sobre el mismo eje.



Figura 2.2: Cuatro diseños de ruedas universales. Recuperado de [23].

Rueda Mecanum o Sueca: Su diseño es parecido a la rueda universal, con la diferencia que los rodillos están montados en un ángulo γ distinto de 90° , usualmente $\gamma = \pm 45^\circ$, según se muestra en la Figura 2.3. Este tipo de rueda permite un giro (actuado) alrededor del eje de la rueda (con velocidad $\dot{\varphi}$) y otro giro (no actuado) de los pequeños rodillos. Es capaz de girar en un plano vertical, así como también de desplazarse en un plano horizontal [23].



Figura 2.3: Rueda mecanum o sueca¹.

2. Marco teórico

Rueda de Bola o Esférica: No impone restricciones directas al movimiento, es decir, es omnidireccional como las ruedas universal y mecanum. En este caso, el eje de rotación de la rueda puede ubicarse en cualquier dirección arbitraria. Una alternativa para conseguir esta libertad de movimiento es mediante el empleo de un anillo activo impulsado por un motor y una caja de cambios que permiten transferir potencia a la bola a través de los rodillos y la fricción, con lo cual esta puede girar en cualquier dirección de forma instantánea. En la Figura 2.4 se muestran diseños de ruedas esféricas activas. Debido a su difícil construcción, la rueda de bola rara vez se utiliza en la práctica como rueda motriz. Otro inconveniente es que la carga útil que pueden soportar estas ruedas debe ser bastante baja debido al punto de contacto, además la superficie de la esfera puede contaminarse al viajar sobre suelo sucio y es difícil superar las condiciones irregulares del suelo [23].

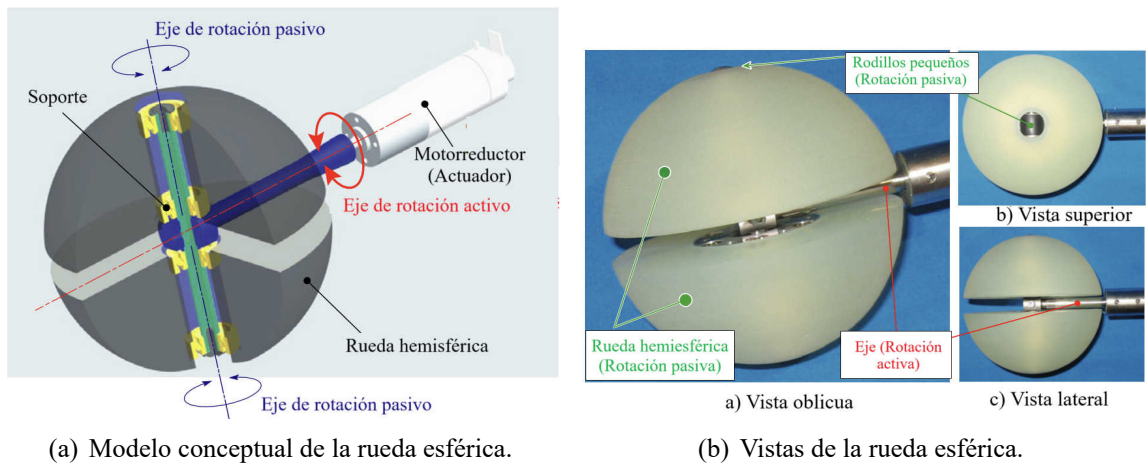


Figura 2.4: Rueda esférica activa desarrollada en el laboratorio Kaneko Higashimori (Universidad de Osaka). Adaptado de [24]

¹Recuperado de <https://www.luisllamas.es/robot-con-mecanum-wheel-controlado-por-arduino/>

2.1.6. Odometría del robot omnidireccional de tres ruedas

La determinación computacional de la posición y orientación de un robot móvil en un momento específico se logra integrando su modelo cinemático en relación con un sistema de referencia global. Este procedimiento se conoce como odometría (dead reckoning), una técnica de navegación empleada en embarcaciones, aeronaves, vehículos terrestres y, más recientemente, en robots móviles. La configuración geométrica de un robot omnidireccional provisto de tres ruedas universales se presenta en la Figura 2.5. Este sistema de locomoción, caracterizado por ruedas universales con un ángulo de los rodillos $\gamma = 0^\circ$, dispone las ruedas de manera radial y simétrica, asegurando que los ejes de rotación de los rodillos sean perpendiculares a los ejes de rotación de cada rueda principal. El sistema de referencia local del robot se ha establecido en el centro del chasis, permitiendo su alineación de modo que el eje X_R del robot sea colineal con el eje Y_W correspondiente a la rueda 1. Se asume que la distancia entre cada rueda y el origen del sistema de referencia del robot es L , y que todas las ruedas poseen un radio uniforme de magnitud r .

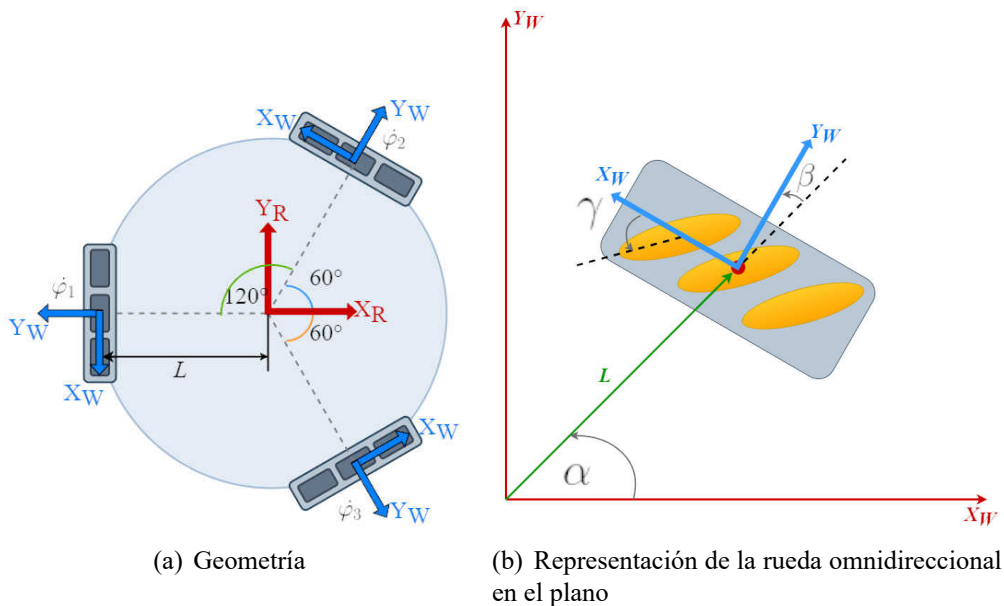


Figura 2.5: Robot omnidireccional de tres ruedas universales.

2. Marco teórico

El cálculo del vector de posición y orientación del robot respecto al sistema de referencia global, denotado como ${}^I\dot{\xi}$, se obtiene a partir de la combinación de las restricciones impuestas por las tres ruedas omnidireccionales del sistema [23], conforme a (2.1):

$$[\sin(\alpha_i + \beta_i + \gamma_i) - \cos(\alpha_i + \beta_i + \gamma_i) - l_i \cos(\beta_i + \gamma_i)] {}^R\mathbf{R}_I {}^I\dot{\xi} = r_w \dot{\varphi}_i \cos \gamma_i \quad (2.1)$$

Antes de aplicar la ecuación de restricción de rodadura, es fundamental definir los valores de los parámetros α , β y γ para cada rueda. Al analizar la Figura 2.5, se observa que el ángulo γ adopta un valor de 0° para la rueda universal, es decir, $\gamma_1 = \gamma_2 = \gamma_3 = 0^\circ$. Esta condición permite una simplificación inmediata de (2.1), la cual describe las restricciones de rodadura de una rueda estándar fija. Debido a la disposición específica del sistema de referencia local del robot, el cálculo de los valores del parámetro α para cada rueda se realiza de manera directa: $\alpha_1 = 60^\circ$, $\alpha_2 = 180^\circ$ y $\alpha_3 = -60^\circ$. Asimismo, dado que las ruedas son tangentes al cuerpo circular del robot, el valor de β es uniforme para todas ellas, adoptando un valor de 0° . Empleando estos valores se pueden obtener las ecuaciones de restricción para las tres ruedas.

Restricción combinada para la rueda 1:

$$[\sin(180^\circ) - \cos(180^\circ) - L \cos(60^\circ)] {}^R\dot{\xi} = r\dot{\varphi}_1 \cos(0^\circ)$$

$$[0 \ 1 \ -L] {}^R\dot{\xi} = r\dot{\varphi}_1$$

Restricción combinada para la rueda 2:

$$[\sin(60^\circ) - \cos(60^\circ) - L \cos(0^\circ)] {}^R\dot{\xi} = r\dot{\varphi}_2 \cos(0^\circ)$$

$$\left[\frac{\sqrt{3}}{2} \quad -\frac{1}{2} \quad -L \right] {}^R \dot{\xi} = r \dot{\varphi}_2$$

Restricción combinada para la rueda 3:

$$[\sin(-60^\circ) - \cos(-60^\circ) - L \cos(0^\circ)] {}^R \dot{\xi} = r \dot{\varphi}_3 \cos(0^\circ)$$

$$\left[-\frac{\sqrt{3}}{2} \quad -\frac{1}{2} \quad -L \right] {}^R \dot{\xi} = r \dot{\varphi}_3$$

La fusión de las tres ecuaciones matriciales de restricción da como resultado la siguiente expresión:

$$\begin{bmatrix} 0 & 1 & -L \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} & -L \\ -\frac{\sqrt{3}}{2} & -\frac{1}{2} & -L \end{bmatrix} {}^R \dot{\xi} = \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r \end{bmatrix} \begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ \dot{\varphi}_3 \end{bmatrix}$$

A partir de la expresión anterior se determina la matriz Jacobiana, la cual permite obtener la expresión cinemática directa de este robot:

$$\mathbf{J} = \mathbf{B}^+ \mathbf{D} = \mathbf{B}^{-1} \mathbf{D}$$

$$\mathbf{J} = \begin{bmatrix} 0 & \frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} \\ \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3L} & -\frac{1}{3L} & -\frac{1}{3L} \end{bmatrix} \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r \end{bmatrix} = \frac{r}{3} \begin{bmatrix} 0 & \sqrt{3} & -\sqrt{3} \\ 2 & -1 & -1 \\ -\frac{1}{L} & -\frac{1}{L} & -\frac{1}{L} \end{bmatrix}$$

$${}^R \dot{\xi} = \mathbf{J} \dot{\delta}$$

2. Marco teórico

$$\begin{aligned}
 {}^R\dot{\xi} &= \frac{r}{3} \begin{bmatrix} 0 & \sqrt{3} & -\sqrt{3} \\ 2 & -1 & -1 \\ -\frac{1}{L} & -\frac{1}{L} & -\frac{1}{L} \end{bmatrix} \dot{\delta} \\
 \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} &= \frac{r}{3} \begin{bmatrix} 0 & \sqrt{3} & -\sqrt{3} \\ 2 & -1 & -1 \\ -\frac{1}{L} & -\frac{1}{L} & -\frac{1}{L} \end{bmatrix} \begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ \dot{\varphi}_3 \end{bmatrix}
 \end{aligned} \tag{2.2}$$

Por lo tanto, las expresiones para las velocidades lineal y angular del robot en función de la velocidad de las ruedas con respecto a su sistema de referencia local serían:

$$\begin{aligned}
 \dot{x} = V_f &= \frac{r\sqrt{3}}{3}(\dot{\varphi}_2 - \dot{\varphi}_3) \\
 \dot{y} = V_l &= \frac{r}{3}(2\dot{\varphi}_1 - \dot{\varphi}_2 - \dot{\varphi}_3) \\
 \dot{\theta} = \omega &= \frac{r}{3L}(\dot{\varphi}_1 + \dot{\varphi}_2 + \dot{\varphi}_3)
 \end{aligned} \tag{2.3}$$

Donde V_f y V_L son las componentes de las velocidades lineales del robot en el marco de referencia local de este en dirección al eje x e y respectivamente, y ω es la velocidad angular del robot.

Para calcular la cinemática inversa respecto al sistema de coordenadas global se debe hallar la inversa de la ecuación 2.2, dando como resultado

$$\begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ \dot{\varphi}_3 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 0 & 1 & -\frac{L}{3} \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} & -\frac{L}{3} \\ -\frac{\sqrt{3}}{2} & -\frac{1}{2} & -\frac{L}{3} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \tag{2.4}$$

2.2. Control por Rechazo Activo de Perturbaciones

En la mayoría de las industrias de control, es difícil establecer modelos matemáticos precisos para describir un sistema. Además, hay algunos términos que no se conocen explícitamente en las ecuaciones matemáticas y, por otro lado, existen algunas perturbaciones externas desconocidas en el entorno del sistema. La incertidumbre, que incluye incertidumbre interna y perturbaciones externas, es omnipresente en los sistemas de control prácticos.

Entre los años 1980 y 1990, Jingqing Han, de la Academia China de Ciencias, propuso un poderoso enfoque de control no convencional para abordar la enorme incertidumbre en los sistemas no lineales. Esta nueva tecnología de control se denominó más tarde control ADRC [25].

La idea principal del ADRC es simplificar la descripción de la planta de manera que agrupe todas las perturbaciones, tanto externas como internas, y todas las cantidades y expresiones desconocidas o ignoradas en un solo término de perturbación. Luego, se procede a estimar los efectos de esta perturbación de manera precisa y a idear los medios para cancelar sus efectos, utilizando la estimación recopilada como parte de la acción de control de retroalimentación.

En general, la perturbación es no estructurada, es decir, puede depender de variables internas del sistema de manera desconocida o ignorada, o también puede depender de fenómenos externos que afectan al sistema.

El control por rechazo activo de perturbaciones se ha posicionado como un enfoque innovador, ya que reformula el problema del control tradicional. En lugar de depender de una representación matemática precisa del sistema para el diseño del controlador, este método se basa en un estimador que, a partir de las señales de entrada y salida, determina la perturbación total, incluyendo incertidumbres y perturbaciones externas. Con esta información, se modifica el comportamiento de la planta para que se asemeje a su versión nominal, permitiendo así el diseño de un controlador que

2. Marco teórico

opere sobre un sistema libre de perturbaciones [26].

El ADRC se basa fundamentalmente en la posibilidad de estimar en línea las perturbaciones de entrada desconocidas que afectan el comportamiento de la planta mediante observadores adecuados y proceder a cancelarlas mediante una ley de control de retroalimentación apropiada utilizando la estimación recopilada de perturbación. La ley de control de retroalimentación generalmente requiere un conocimiento completo del estado, pero la falta de verificación de esta suposición no se elimina de la realidad. El desafío entonces es diseñar un observador de un solo estado que simultáneamente estime las variables de estado no medidas del sistema y las perturbaciones desconocidas [27].

2.3. Observadores de estado

Hasta ahora, se ha asumido que el estado completo x está disponible para su medición, es decir, de alguna manera podemos medir todas las variables incluidas en el vector de estado x (por ejemplo, añadiendo sensores). En la práctica, no todas las variables están disponibles. Existen magnitudes cuya medición directa resulta compleja, mientras que en otros casos los sensores requeridos para obtener dichas mediciones son excesivamente costosos. Afortunadamente, en muchos casos es posible estimar el estado x basándose únicamente en la entrada u y la salida y . El mecanismo, o algoritmo, para tal estimación se llama observador [28].

2.3.1. Observador Proporcional Integral Generalizado (GPIO)

En esta sección se presentan algunas generalidades relacionadas con los así llamados observadores GPIO (por sus siglas en inglés, *Generalized Proportional Integral Observer*). Tal tipo de estimador será empleado en la Sección 5.1 en combinación con un controlador ADRC con la finalidad de regular la velocidad angular del eje de los motores de CD del robot omnidireccional.

Considérese el siguiente sistema escalar de segundo orden [29]:

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= ku + \xi \\
 y &= x_1
 \end{aligned} \tag{2.5}$$

donde en (2.5), ξ es una perturbación completamente desconocida pero acotada, que puede ser de naturaleza no lineal y dependiente de los estados del sistema y de otras perturbaciones externas; y k es la ganancia del sistema. El diseño clásico de un observador GPIO de orden completo se establece como:

$$\begin{aligned}
 \dot{\hat{x}}_1 &= \hat{x}_2 + \lambda_{n+1}(y - \hat{x}_1) \\
 \dot{\hat{x}}_2 &= z_1 + ku + \lambda_n(y - \hat{x}_1) \\
 \dot{z}_1 &= z_2 + \lambda_{n-1}(y - \hat{x}_1) \\
 \dot{z}_2 &= z_3 + \lambda_{n-2}(y - \hat{x}_1) \\
 &\vdots \\
 \dot{z}_{n-1} &= z_n + \lambda_1(y - \hat{x}_1) \\
 \dot{z}_n &= \lambda_0(y - \hat{x}_1)
 \end{aligned} \tag{2.6}$$

Definiendo el error de estimación como $\tilde{e} = y - \hat{x}_1$, la dinámica del error de estimación está dada por:

$$\tilde{e}^{(n+2)} + \lambda_{n+1}\tilde{e}^{(n+1)} + \dots + \lambda_1\tilde{e}^{(1)} + \lambda_0\tilde{e} = \xi^{(n)} \tag{2.7}$$

donde se ha empleado la notación $f^{(n)}$ como la n -ésima derivada temporal de la función f . Si $\xi^{(n)}$ es absolutamente y uniformemente acotada por una constante $\epsilon > 0$, es posible elegir los coeficientes λ_i en una forma tal que el error de observación converja asintóticamente a un disco de radio arbitrariamente pequeño, pre-especificado en el $n + 2$ espacio de fase del sistema. Obsérvese

2. Marco teórico

que la parte predominantemente lineal del sistema perturbado (2.7), obedece el siguiente polinomio característico para \tilde{e} :

$$\lambda(s) = s^{n+2} + \lambda_{n+1}s^{n+1} + \dots + \lambda_1s + \lambda_0 \quad (2.8)$$

y las ganancias λ_i del observador pueden ser sintonizadas utilizando un polinomio deseado $\lambda_d(s)$ como sigue:

$$f(x) = \begin{cases} (s^2 + 2\zeta\omega_n s + \omega_n^2)^{(n+2)/2} & \text{si } n \text{ es par} \\ (s + \omega_n)(s^2 + 2\zeta\omega_n s + \omega_n^2)^{(n+1)/2} & \text{si } n \text{ es impar} \end{cases} \quad (2.9)$$

Desarrollo del robot omnidireccional

3.1. Diseño mecánico

En esta sección se detalla el diseño mecánico del robot móvil omnidireccional de tres ruedas, abarcando desde la concepción inicial hasta la implementación final. Se analizan los materiales seleccionados y las consideraciones estructurales esenciales para asegurar la funcionalidad y eficiencia del robot. Además, se describen las etapas de diseño y construcción, incluyendo la integración de los componentes mecánicos y su interacción con los sistemas de control.

3.1.1. Objetivos de diseño

Tal como se mencionó en el objetivo general de este trabajo de tesis, se evaluó el desempeño de un robot móvil. Para este propósito, se desarrolló un robot omnidireccional de tres ruedas, sirviendo este como plataforma base para la realización de experimentos y pruebas de control, con el objetivo de implementar la estrategia de control ADRC.

El material principal seleccionado para las piezas del chasis fue una lámina de acrílico de 3 mm,

elegida por su relativa resistencia, durabilidad y ligereza. Se optó por un diseño circular para la placa inferior del chasis, la cual incluye aberturas específicas para las ruedas del robot, así como orificios estratégicamente ubicados para facilitar el ensamblaje mediante tornillos, espaciadores y tuercas. El diseño completo de las placas que conforman el chasis es simétrico, lo que contribuye a un equilibrio óptimo de las fuerzas generadas por los motores, asegurando así una distribución uniforme y estable de las cargas. Las dimensiones del chasis se determinaron en función del espacio necesario para acomodar correctamente los motores y demás componentes del robot, resultando en un diámetro de 350 mm.

3.1.2. Diseño CAD

Para el diseño del chasis se utilizó el software SolidWorks® tomando en consideración los puntos antes mencionados: material, geometría y dimensiones. Como se muestra en la Figura 3.1, el chasis consta de tres placas de acrílico de 3 mm de grosor, diseñadas para proporcionar una estructura estable y ligera. Las placas fueron modeladas cuidadosamente, teniendo en cuenta las dimensiones necesarias para alojar todos los componentes del robot.

Además de las placas del chasis, se diseñó una pieza que cumple con la función de alojar las dos baterías LiPo que el prototipo utiliza. Dicha pieza también fue modelada utilizando SolidWorks® .

3. Desarrollo del robot omnidireccional

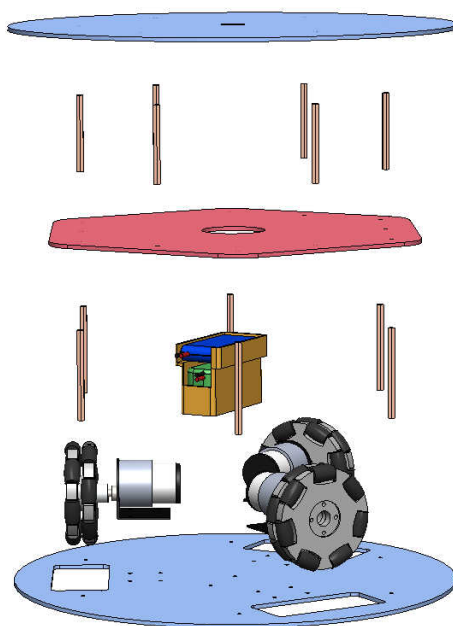
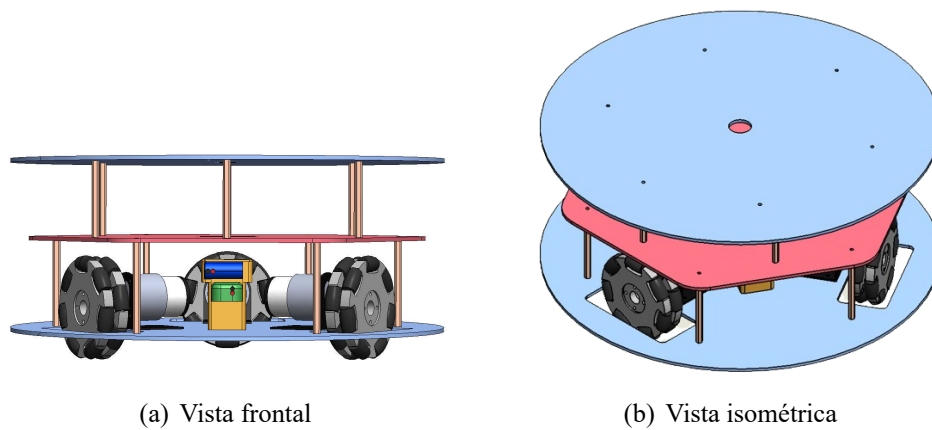
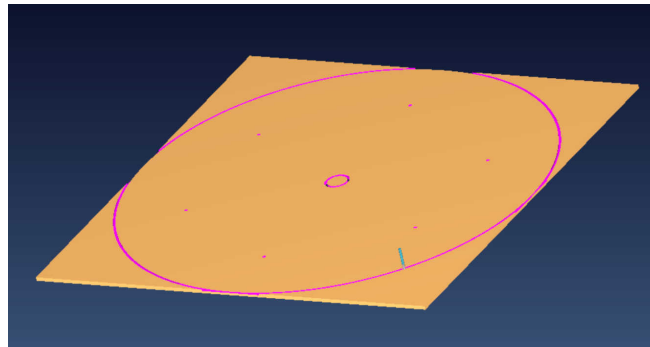


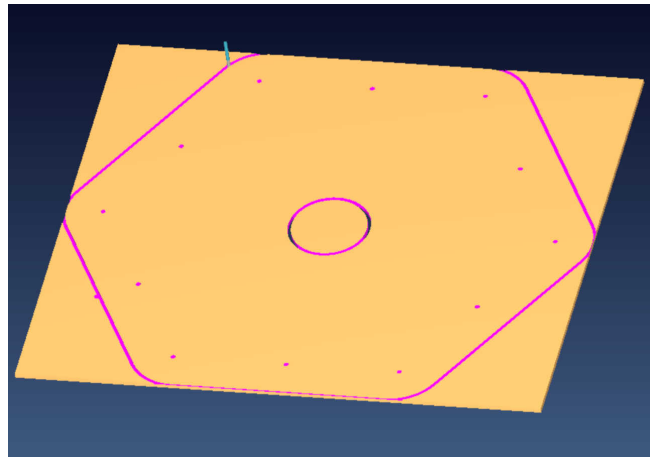
Figura 3.1: Vistas del robot móvil.

3.1.3. Fabricación del chasis

El diseño CAD de las piezas fue exportado a archivos compatibles con el router CNC disponible en el taller de plásticos de la Universidad Tecnológica de la Mixteca. Antes de proceder con el mecanizado, se realizaron simulaciones de las rutas del cortador para asegurar el éxito del proceso como se muestra en la Figura 3.2. El maquinado involucró el corte preciso de las placas de acrílico, garantizando que todas las piezas encajaran perfectamente según las tolerancias especificadas.



(a) Pieza superior de robot



(b) Soporte medio del robot

Figura 3.2: Simulación del maquinado de las piezas en el software VISI® .

3. Desarrollo del robot omnidireccional

3.2. Diseño electrónico

En esta sección se detalla el diseño electrónico del robot móvil omnidireccional. La electrónica representa un componente esencial en la arquitectura funcional del robot, al posibilitar un control cinemático preciso y facilitar la integración operativa de los distintos subsistemas que lo conforman. A continuación, se describen los principales elementos electrónicos utilizados; además, se aborda el diseño del circuito y el montaje de los componentes.

3.2.1. Componentes electrónicos

ESP32 WROOM: Es un microcontrolador potente con conectividad Wi-Fi y Bluetooth, que se dirige a una amplia variedad de aplicaciones, que van desde las redes de sensores de baja potencia hasta las tareas más exigentes, como la codificación de voz, la transmisión de música y la decodificación de MP3. Dispone de dos núcleos de CPU que pueden controlar individualmente, y la frecuencia de reloj de la CPU es ajustable de 80 MHz a 240 MHz [30]. Estas características lo hacen ideal para manejar la comunicación y el control de los componentes del robot. Su capacidad de procesamiento permite la implementación de algoritmos de control.

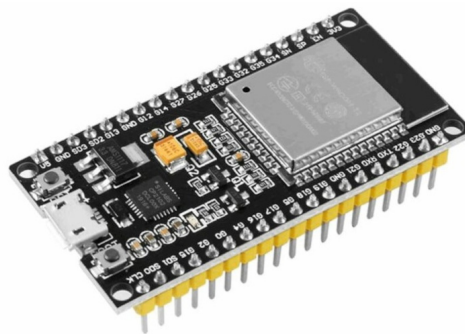


Figura 3.3: Microcontrolador ESP32 WROOM¹.

Motor de CD con encoder: Los motores de CD modelo JGB37-520 están equipados con encoders de efecto Hall que ofrecen retroalimentación sobre la velocidad y la posición del eje. Estos motores operan a 12 V y tienen una velocidad de 110 RPM. En la Tabla 3.1 se muestran las principales características de este motor.

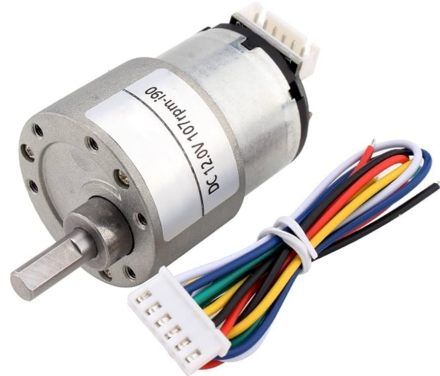


Figura 3.4: Motor de CD con encoder².

Tabla 3.1: Características del motor JGB37-520.

Modelo del motor	JGB37-520-12100-EN
Voltaje nominal	12 V
Tasa de reducción de la caja	90:1
Velocidad sin carga	110 rpm
Corriente sin carga	0.15 A
Velocidad con carga	85 rpm
Corriente con carga	0.65 A
Torque	3 kg · cm
Potencia	7.8 W
Resolución del encoder	3960 ppr

Módulo Monster Motor Shield (Puente H): El módulo controlador de motores de puente completo VNH2SP30-E está destinado a una amplia gama de aplicaciones automotrices. Además, incorpora un sensor de corriente que proporciona 0.13 V aproximadamente por amperio de

¹Recuperado de <https://www.tostatronic.com>

²Recuperado de <https://www.amazon.com>

3. Desarrollo del robot omnidireccional

corriente de salida [31]. La Tabla 3.2 muestra las principales características de este controlador de motor de puente completo.

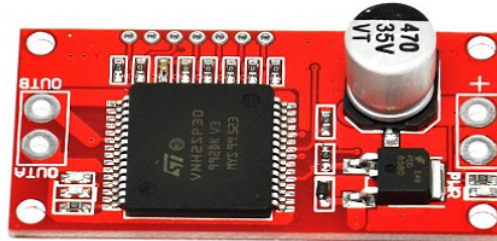


Figura 3.5: Módulo controlador de motores VNH2SP30³.

Tabla 3.2: Características del módulo VNH2SP30-E.

Modelo del módulo	VNH2SP30-E
Voltaje de operación	5.5-16 V
Corriente máxima de operación	30 A
Frecuencia máxima de PWM	20 kHz
Número de canales de entrada	1
Número de canales de salida	1

Baterías LiPo: La batería de polímero de litio también conocida como LiPo, se diferencia del resto de las baterías por el electrolito usado. Dentro de las especificaciones eléctricas de este tipo de baterías se encuentran su alta densidad de energía, su bajo mantenimiento, no presenta efectos de memoria y bajo porcentaje de auto-descarga [32]. Este tipo de baterías son ampliamente utilizadas en proyectos de robótica móvil debido a sus adecuadas características.

En este caso, se emplean dos baterías LiPo: la primera, de dos celdas (7.4 V) y 1000 mA_H, proporciona el voltaje lógico para el microcontrolador y los encoders de los motores. La segunda, de tres celdas (11.1 V) y 2200 mA_H, suministra el voltaje necesario para el funcionamiento de los motores.

³Recuperado de <https://robocraze.com>



Figura 3.6: Batería LiPo⁴.

Módulo regulador de voltaje: El módulo LM2596 es una fuente de alimentación Step-Down (reductora) DC a DC que permite ajustar el voltaje de entrada para obtener un voltaje de salida menor. En este caso, se empleó para reducir el voltaje de 7.4 V a 5 V, el cual era requerido por los otros componentes.

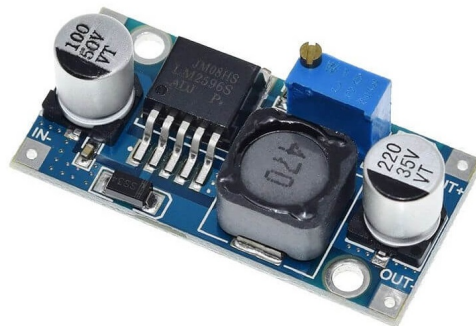


Figura 3.7: Módulo regulador de voltaje LM2596⁵.

⁴Recuperado de <https://www.flydonkeydrone.com>

⁵Recuperado de <https://uelectronics.com>

3. Desarrollo del robot omnidireccional

3.3. Construcción

Con todas las piezas necesarias listas, se procedió con el ensamblaje del chasis y la instalación de los componentes restantes. A continuación, se detalla el proceso de armado y montaje de cada pieza, asegurando una integración correcta y eficiente del robot.

3.3.1. Ensamblaje del robot

En la Tabla 3.3 se muestran algunas de las piezas utilizadas para el ensamblaje del robot. Además, se indica la procedencia de estas piezas, ya que no todas fueron manufacturadas internamente, sino que algunas fueron adquiridas de manera comercial.

Tabla 3.3: Piezas utilizadas en el ensamble.

Nombre de la pieza	Cantidad	Manufacturada	Material
Base de chasis	1	Si	Acrílico 3 mm
Soporte medio del robot	1	Si	Acrílico 3 mm
Tapa el robot	1	Si	Acrílico 3 mm
Portabaterías	1	Si	PLA
Soporte tipo L para motor	3	No	Aluminio
Motor de CD	3	No	-
Rueda universal	3	No	Aluminio/Silicón
Espaciador de latón de 7 mm	6	No	Latón
Espaciador de latón de 6 mm	6	No	Latón
Tornillo M3×5	14	No	Latón
Tuerca M3	14	No	Latón

La base del chasis, junto con los espaciadores de 7 mm, conformó el espacio donde se colocaron los motores y el porta-baterías. Los motores, previamente montados en sus respectivos soportes tipo L y equipados con ruedas universales, se ajustaron en su lugar con ayuda de tornillos y tuercas. Sobre

los espaciadores se instaló la segunda pieza de acrílico, la cual sirvió como soporte medio del robot. A esta segunda pieza se atornillaron espaciadores de 6 mm, creando así el espacio necesario para alojar los demás componentes electrónicos, como el microcontrolador, los puentes H y el regulador de voltaje. Una vez realizadas las conexiones entre los componentes (ver Figura 3.8) en la placa PCB y atornillada esta a la base de acrílico (ver Figura 3.9), se colocó la tapa del robot. De esta manera, el ensamblaje del robot se completó de forma estructurada y organizada. Para una descripción más detallada de las conexiones entre los componentes, se recomienda consultar el Anexo A.1, en el cual se presenta el diagrama esquemático correspondiente de la PCB realizada.

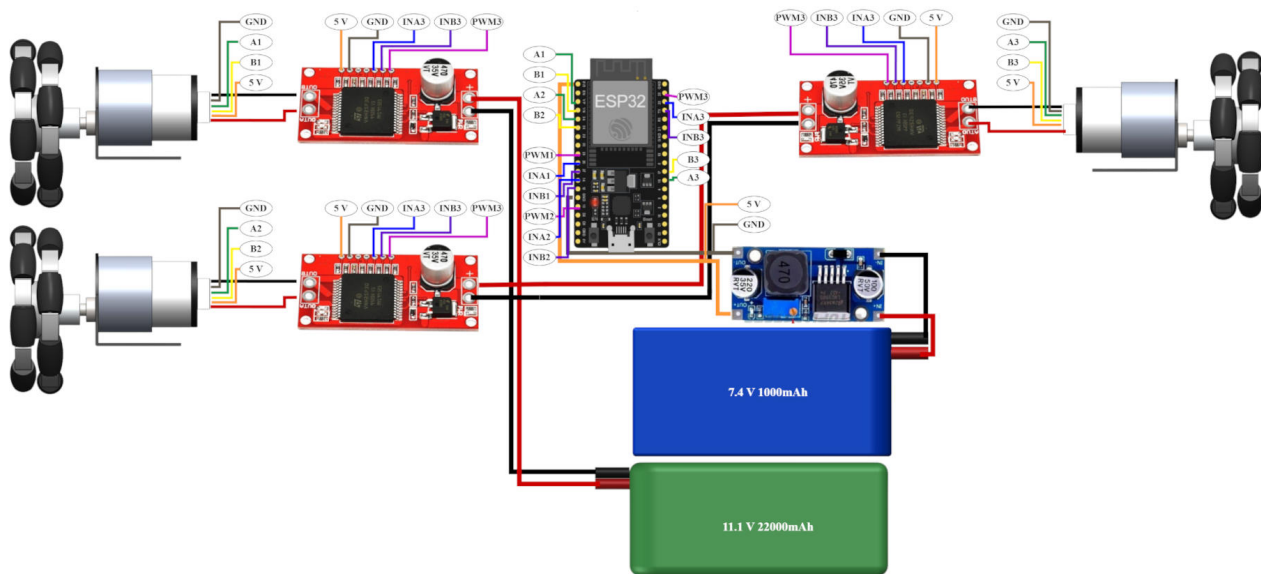


Figura 3.8: Diagrama de conexiones de los componentes electrónicos.

Nota: En el Anexo E.5 se puede consultar las herramientas de software utilizadas durante la elaboración de este trabajo de tesis.

3. Desarrollo del robot omnidireccional

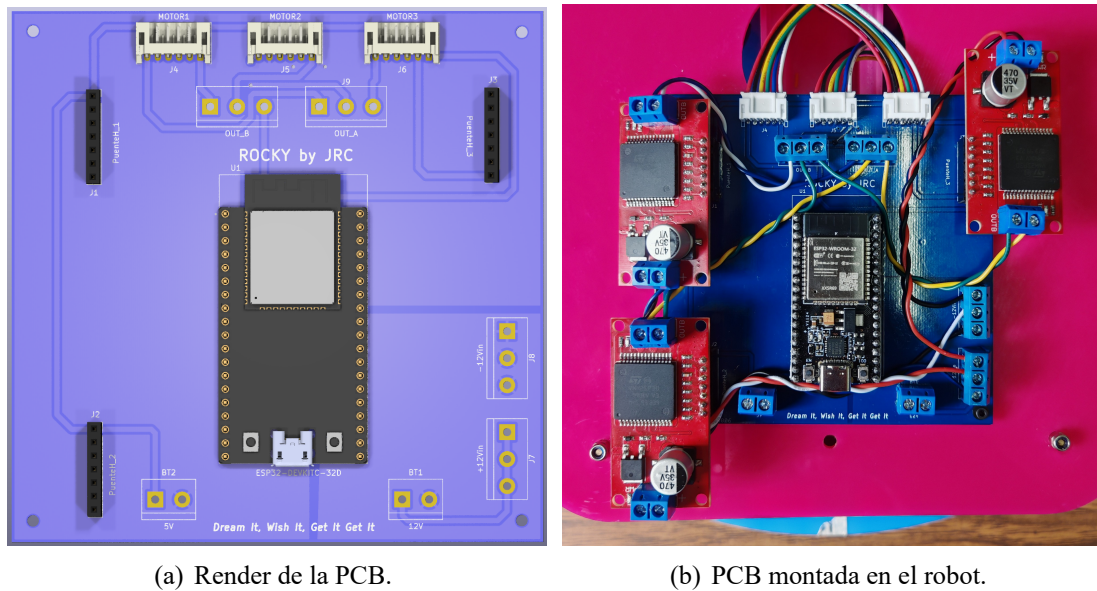


Figura 3.9: Placa PCB diseñada.

Como se puede observar en los pasos descritos anteriormente, el robot cuenta con una estructura modular de fácil ensamblaje. Este diseño modular permite un acceso rápido a los componentes y facilita la realización de modificaciones. En la Figura 3.10 se presenta el robot completamente ensamblado según el diseño presentado en la sección 3.1.2. A partir de este punto, dicho prototipo será referido como ROCKY.

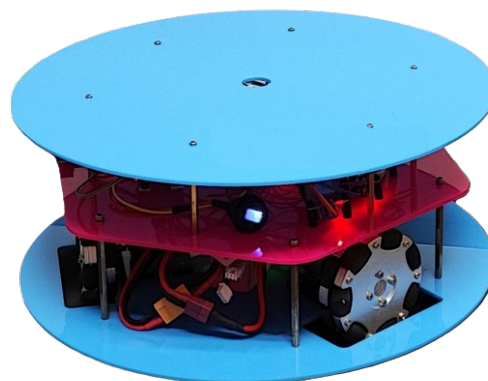


Figura 3.10: Robot ensamblado.

Control cinemático

En este capítulo se aborda el diseño del control cinemático del robot omnidireccional de tres ruedas. Se detalla el proceso de simulación utilizando el simulador CoppeliaSim[®] y se presentan los resultados obtenidos. Estos resultados ayudan a validar la cinemática directa e inversa del robot, asegurando su correcto funcionamiento antes de la implementación física.

4.1. Modelado y control cinemático del robot

El modelo cinemático de un robot móvil representa las relaciones matemáticas que describen cómo se mueve el robot en función de sus velocidades de entrada, como las velocidades de las ruedas o las velocidades en los actuadores. Este modelo es esencial para entender y controlar el movimiento del robot, ya que permite predecir su posición y orientación futuras basadas en las entradas de control actuales.

Considere el robot omnidireccional de tres ruedas que se muestra en la Figura 4.1 en un marco de referencia cartesiano. Supongamos que el robot parte del reposo y en algún instante de tiempo alcanza la posición $[x_c, y_c]$. Las ecuaciones que describen el modelo cinemático diferencial del robot en el punto central son:

$$\begin{aligned}
 \dot{x}_c &= V_f \cos(\varphi) - V_L \sin(\varphi) \\
 \dot{y}_c &= V_f \sin(\varphi) + V_L \cos(\varphi) \\
 \dot{\varphi} &= \omega
 \end{aligned}
 \tag{4.1}$$

En estas ecuaciones \dot{x}_c y \dot{y}_c representan las velocidades lineales en las direcciones x e y respectivamente, y φ es la orientación del robot respecto al marco de referencia global. V_f y V_L son las componentes de las velocidades lineales del robot en el marco de referencia local de este, y ω es la velocidad angular del robot.

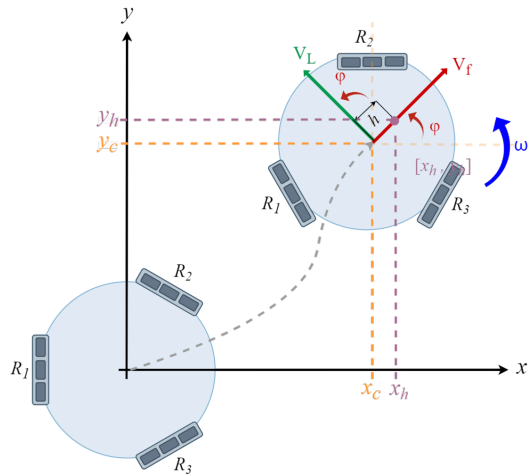


Figura 4.1: Representación del robot omnidireccional en el plano cartesiano.

En ciertos casos, es necesario controlar un punto específico del robot que no sea su centro. Esto es especialmente relevante en aplicaciones donde la precisión del movimiento de una parte específica del robot es crucial, como en la manipulación de objetos o ensamblaje. Controlar un punto distinto del centro del robot puede implicar cálculos adicionales de cinemática para garantizar que el movimiento del punto deseado sea preciso y seguro. Si ahora el punto de interés se encuentra

4. Control cinemático

desplazado una distancia h frente al punto c del robot, el modelo se reescribe como:

$$\begin{aligned}\dot{x}_h &= \dot{x}_c - h \sin(\varphi) \dot{\varphi} \\ \dot{y}_h &= \dot{y}_c + h \cos(\varphi) \dot{\varphi} \\ \dot{\varphi} &= \omega\end{aligned}\tag{4.2}$$

sustituyendo (4.1) en (4.2) se obtiene:

$$\begin{aligned}\dot{x}_h &= V_f \cos(\varphi) - V_L \sin(\varphi) - h \sin(\varphi) \dot{\varphi} \\ \dot{y}_h &= V_f \sin(\varphi) + V_L \cos(\varphi) + h \cos(\varphi) \dot{\varphi} \\ \dot{\varphi} &= \omega\end{aligned}\tag{4.3}$$

Al considerar que el robot parte desde el reposo, inicialmente $\dot{x}_c = 0$, $\dot{y}_c = 0$ y $\dot{\varphi} = 0$. Esto implica que V_f y V_L también son inicialmente cero. A medida que el robot comienza a moverse, las velocidades V_f , V_L y ω se incrementan, lo que provoca cambios en la posición $[x_c, y_c]$ y la orientación φ .

4.2. Diseño del control cinemático

Para obtener un control más preciso sobre el movimiento del robot, se reescribe (4.3) en forma matricial, lo que facilita su manipulación matemática y la aplicación de técnicas de control:

$$\begin{bmatrix} \dot{x}_h \\ \dot{y}_h \\ \dot{\varphi} \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\varphi) & -\sin(\varphi) & -h \sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) & h \cos(\varphi) \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{J}} \begin{bmatrix} V_f \\ V_L \\ \omega \end{bmatrix}\tag{4.4}$$

De esta forma, obtenemos la matriz Jacobiana \mathbf{J} , que nos permite expresar la cinemática directa del robot.

Al aplicar la cinemática inversa, se obtienen las velocidades V_f , V_L y ω . La cinemática inversa se puede obtener calculando la inversa de la matriz Jacobiana, como se muestra a continuación:

$$\begin{bmatrix} V_f \\ V_L \\ \omega \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \dot{x}_h \\ \dot{y}_h \\ \dot{\varphi} \end{bmatrix} \quad (4.5)$$

donde \mathbf{J}^{-1} es la inversa de la matriz Jacobiana, que se calcula como:

$$\mathbf{J}^{-1} = \begin{bmatrix} \cos(\varphi) & \sin(\varphi) & h \sin(\varphi) \\ -\sin(\varphi) & \cos(\varphi) & -h \cos(\varphi) \\ 0 & 0 & 1 \end{bmatrix}$$

Para controlar la posición del robot, definimos las velocidades deseadas u_x , u_y y u_φ utilizando una ley de control Proporcional-Integral (PI). La inclusión del término integral tiene como objetivo eliminar el error de estado estacionario causado por dinámicas no modeladas como fricción o deslizamientos leves. Las ecuaciones de control quedan definidas como:

$$\begin{aligned} \dot{x}_h = u_x &= k_p(x_d - x) + k_i \int_0^t (x_d - x) d\tau \\ \dot{y}_h = u_y &= k_p(y_d - y) + k_i \int_0^t (y_d - y) d\tau \\ \dot{\varphi} = u_\varphi &= k_p(\varphi_d - \varphi) + k_i \int_0^t (\varphi_d - \varphi) d\tau \end{aligned} \quad (4.6)$$

donde $k_p \in \mathbb{R}^+$ y $k_i \in \mathbb{R}^+$ son las ganancias proporcional e integral respectivamente. Los términos $(x_d - x)$, $(y_d - y)$ y $(\varphi_d - \varphi)$ representan los errores de posición y orientación respecto a las

4. Control cinemático

referencias deseadas x_d , y_d y φ_d . Finalmente, se sustituyen las señales de control calculadas en (4.6) dentro del modelo cinemático inverso (4.5), obteniendo las velocidades lineal y angular requeridas por el robot:

$$\begin{aligned}V_f &= u_x \cos(\varphi) + u_y \sin(\varphi) + u_\varphi h \sin(\varphi) \\V_L &= -u_x \sin(\varphi) + u_y \cos(\varphi) - u_\varphi h \cos(\varphi) \\ \omega &= u_\varphi\end{aligned}\tag{4.7}$$

A partir de estas expresiones se calculan las entradas de control necesarias para mover al robot hacia su pose deseada, asegurando un seguimiento de trayectoria preciso y robusto ante perturbaciones.

4.3. Control de trayectoria

El control de trayectoria extiende el control de posición previamente descrito al considerar referencias que varían en el tiempo. En lugar de fijar una posición deseada estática (x_d, y_d, φ_d) , se define una trayectoria deseada $(x_d(t), y_d(t), \varphi_d(t))$ que el robot debe seguir en cada instante. Para garantizar que el robot siga la trayectoria de manera precisa, se definen los errores de seguimiento como:

$$\begin{aligned}e_x &= x_d(t) - x \\e_y &= y_d(t) - y \\e_\varphi &= \varphi_d(t) - \varphi\end{aligned}\tag{4.8}$$

Dichos errores deben ser regulados a cero conforme el tiempo avanza. Para ello, se establece una ley de control que combina la derivada de la trayectoria deseada (prealimentación) con una acción

de corrección Proporcional-Integral:

$$\begin{aligned}\dot{x}_h = u_x &= \dot{x}_d(t) + k_p e_x + k_i \int_0^t (e_x) d\tau \\ \dot{y}_h = u_y &= \dot{y}_d(t) + k_p e_y + k_i \int_0^t (e_y) d\tau \\ \dot{\varphi} = u_\varphi &= \dot{\varphi}_d(t) + k_p e_\varphi + k_i \int_0^t (e_\varphi) d\tau\end{aligned}\tag{4.9}$$

donde k_p y k_i son las ganancias proporcional e integral, respectivamente. La inclusión del término integral permite compensar perturbaciones sostenidas y minimizar el error de arrastre durante el movimiento. Sustituyendo las señales de control calculadas en (4.9) dentro de la cinemática inversa (4.5), se obtienen las expresiones finales para las velocidades del robot:

$$\begin{aligned}V_f &= u_x \cos(\varphi) + u_y \sin(\varphi) + u_\varphi h \sin(\varphi) \\ V_L &= -u_x \sin(\varphi) + u_y \cos(\varphi) - u_\varphi h \cos(\varphi) \\ \omega &= u_\varphi\end{aligned}\tag{4.10}$$

De esta forma, el robot es capaz de seguir la trayectoria deseada compensando dinámicamente los errores de seguimiento y garantizando una convergencia robusta hacia la trayectoria de referencia.

4.4. Exportación del modelo a CoppeliaSim

CoppeliaSim[®] es una herramienta de software utilizada para el desarrollo rápido de algoritmos, creación rápida de prototipos y verificación, educación en robótica, monitoreo remoto y más [33].

El modelo CAD previamente diseñado se exportó al simulador CoppeliaSim[®]. Este paso resultó crucial, ya que permitió verificar que el modelo del robot funcionara correctamente en un entorno virtual antes de su implementación física. Utilizando CoppeliaSim[®], se pudieron realizar diversas pruebas en el modelo, asegurando que el robot se comportara adecuadamente y operara conforme

4. Control cinemático

a las especificaciones de diseño.

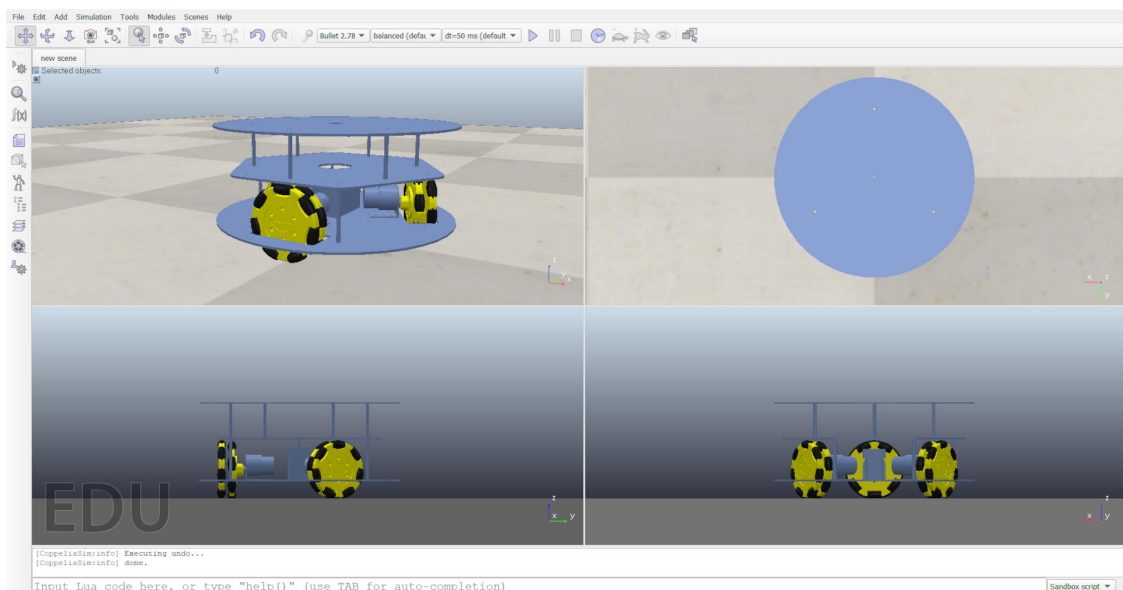


Figura 4.2: Vistas del robot en el simulador.

En la Figura 4.2 se presentan diversas vistas del modelo del robot en el simulador CoppeliaSim[®]. Con el modelo exportado, fue posible proceder con las pruebas del control cinemático, asegurando así su correcto funcionamiento en un entorno virtual.

4.5. Resultados de simulación

En esta sección se presentan los resultados obtenidos de las simulaciones del control cinemático tanto de posición como de seguimiento de trayectorias llevadas a cabo en CoppeliaSim[®]. Se describen los escenarios de validación y se analiza el desempeño del robot bajo dichas condiciones.

4.5.1. Control de posición

La primera validación corresponde al control de posición, donde se estableció como objetivo que el robot alcanzara un punto específico en el plano. Además de las coordenadas cartesianas en los ejes x e y , se consideró también una orientación deseada φ_d .

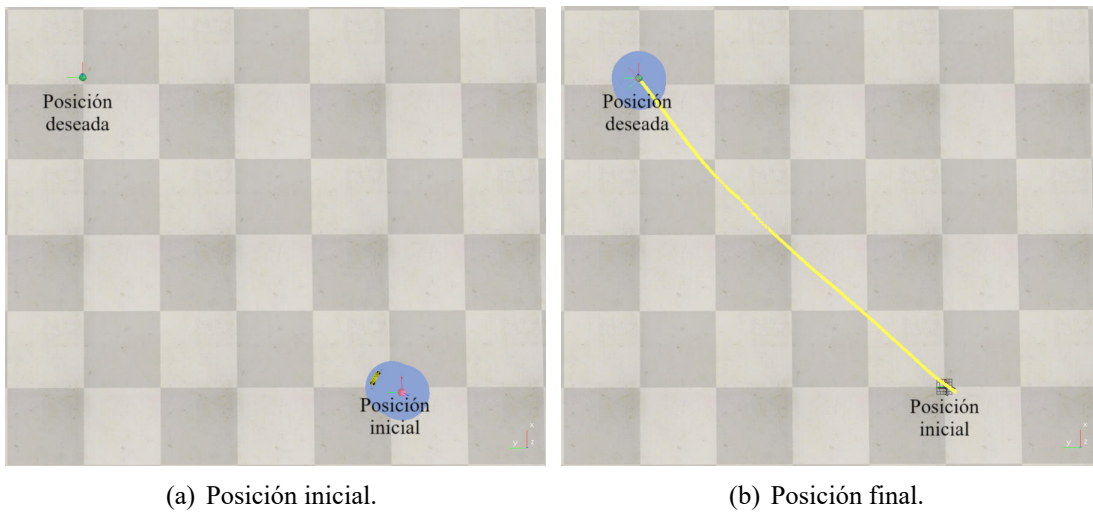


Figura 4.3: Control de posición en el punto $[2, 2]$ m.

En la Figura 4.3(b) se ilustra la trayectoria seguida por el robot para alcanzar el objetivo definido en $[2, 2]$ m con una orientación final de 45° .

Con el fin de analizar de manera más detallada la respuesta del sistema, se estableció una comunicación entre CoppeliaSim y Matlab[®], lo que permitió representar gráficamente la evolución de la posición y orientación.

Los resultados mostrados en la Figura 4.4 evidencian que el robot converge hacia el objetivo en aproximadamente 2 segundos. Para esta simulación se fijó un valor de $k = 0.5$ en las ganancias asociadas a u_x , u_y y u_φ , logrando que la velocidad frontal, lateral y la orientación se regularan de

4. Control cinemático

forma simultánea. La evolución de la orientación puede observarse en la Figura 4.5, alcanzando la referencia en un tiempo cercano al de la posición.

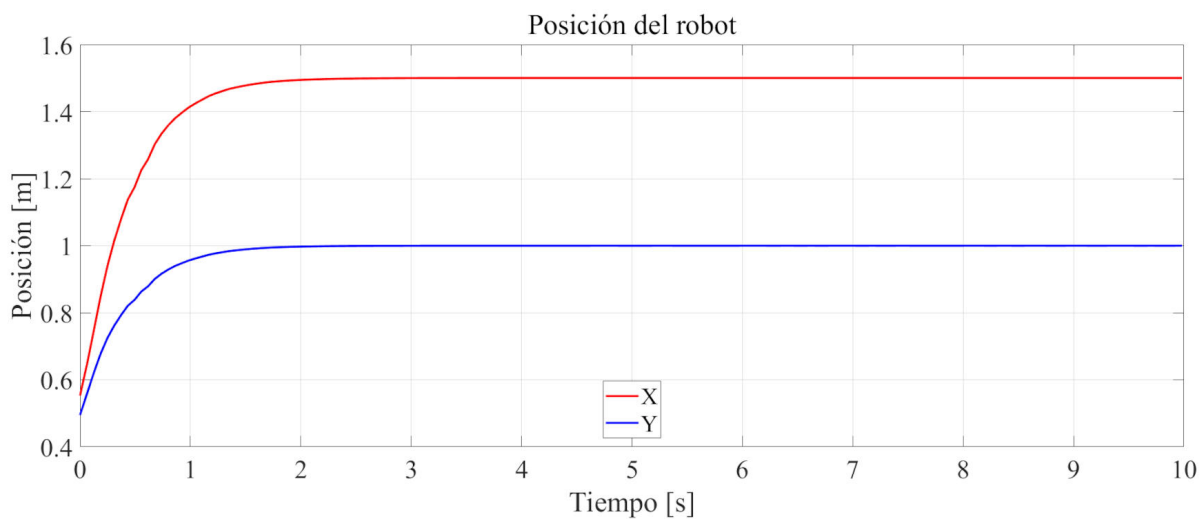


Figura 4.4: Posición del robot en los ejes X, Y .

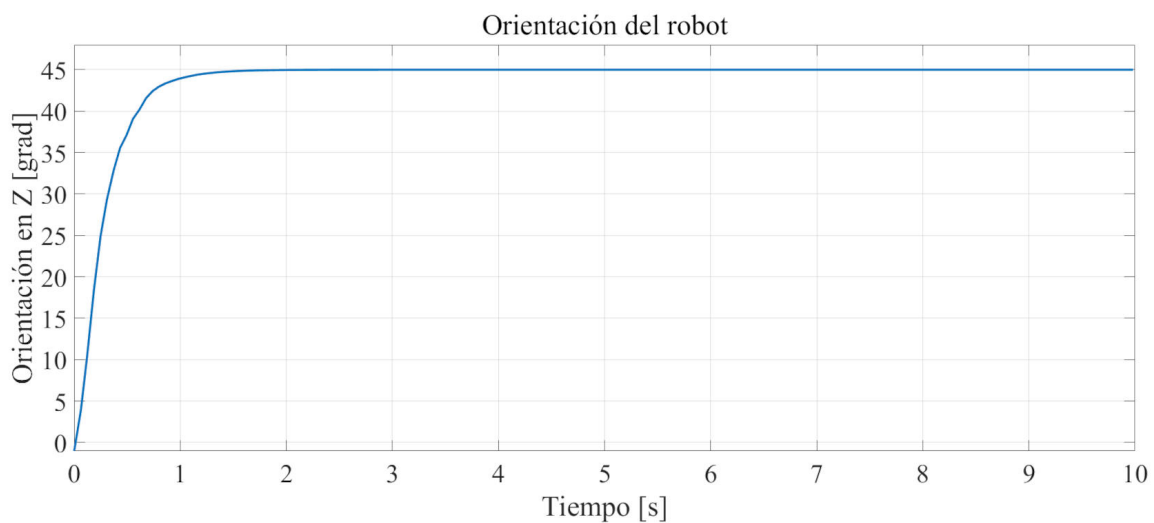


Figura 4.5: Orientación del robot respecto al eje Z .

4.5.2. Control de trayectoria

Además del control de regulación de posición, se evaluó el desempeño del robot en el seguimiento de trayectorias. Para ello, se definió un recorrido de referencia en el plano mediante una curva paramétrica en función del tiempo, la cual debía ser seguida por el robot a través del control cinemático. Las ecuaciones que describen dicha trayectoria se muestran en (4.11):

$$\begin{aligned}x_d(t) &= a \cdot \cos(0.5 \cdot t) \\y_d(t) &= a \cdot \sin(t)\end{aligned}\tag{4.11}$$

Donde a corresponde a la amplitud de la curva y t representa la variable temporal.

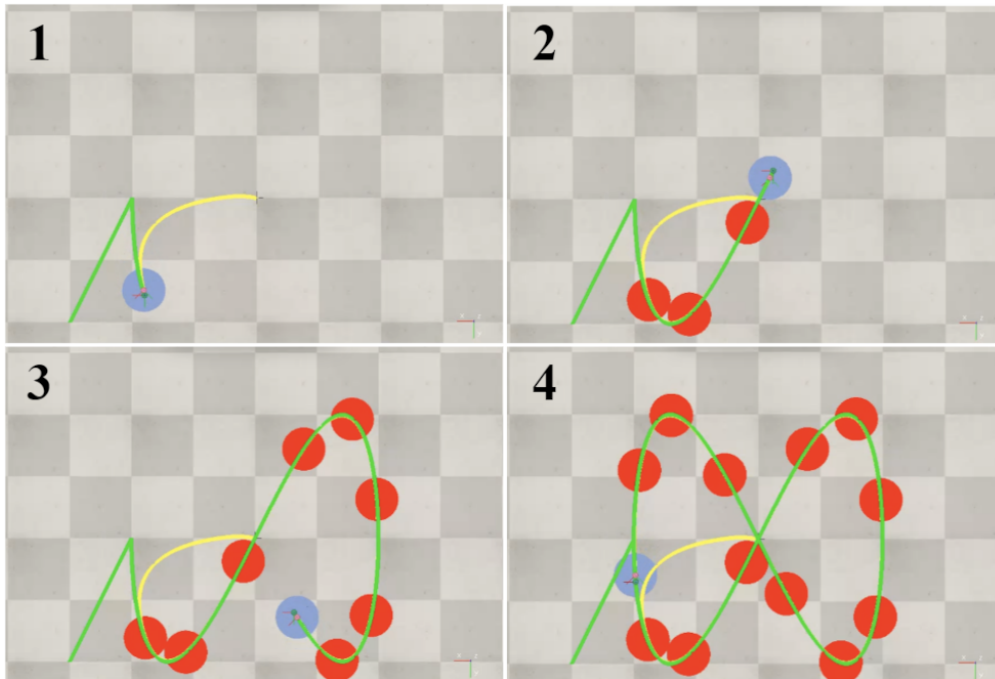


Figura 4.6: Trayectoria seguida por el robot.

En la Figura 4.6 se ilustran los marcadores generados por el robot a lo largo de su recorrido,

4. Control cinemático

los cuales permiten verificar la correspondencia entre la trayectoria seguida y la de referencia. La trayectoria deseada se representa en color verde, mientras que la trayectoria real aparece en color amarillo, lo que facilita la comparación visual y confirma el adecuado desempeño del control cinemático.

4.6. Resultados de la implementación

En esta sección se presentan los resultados experimentales obtenidos a partir de la implementación del control cinemático en el robot desarrollado. Se realizaron pruebas tanto de regulación de posición como de seguimiento de trayectorias, con el objetivo de evaluar el desempeño del sistema bajo condiciones reales de operación.

La arquitectura general del sistema se muestra en la Figura 4.7. En ella se aprecia cómo un ordenador externo ejecuta el control cinemático, mientras que el microcontrolador ESP32 se encarga del control de velocidad de los motores, la estimación de odometría y la lectura de los encoders. Para este fin, se empleó un controlador PID encargado de regular la velocidad angular de cada motor, de modo que las referencias generadas por el control cinemático pudieran ser seguidas con precisión. El análisis de la sintonización de este controlador se muestra en el Anexo D.4.

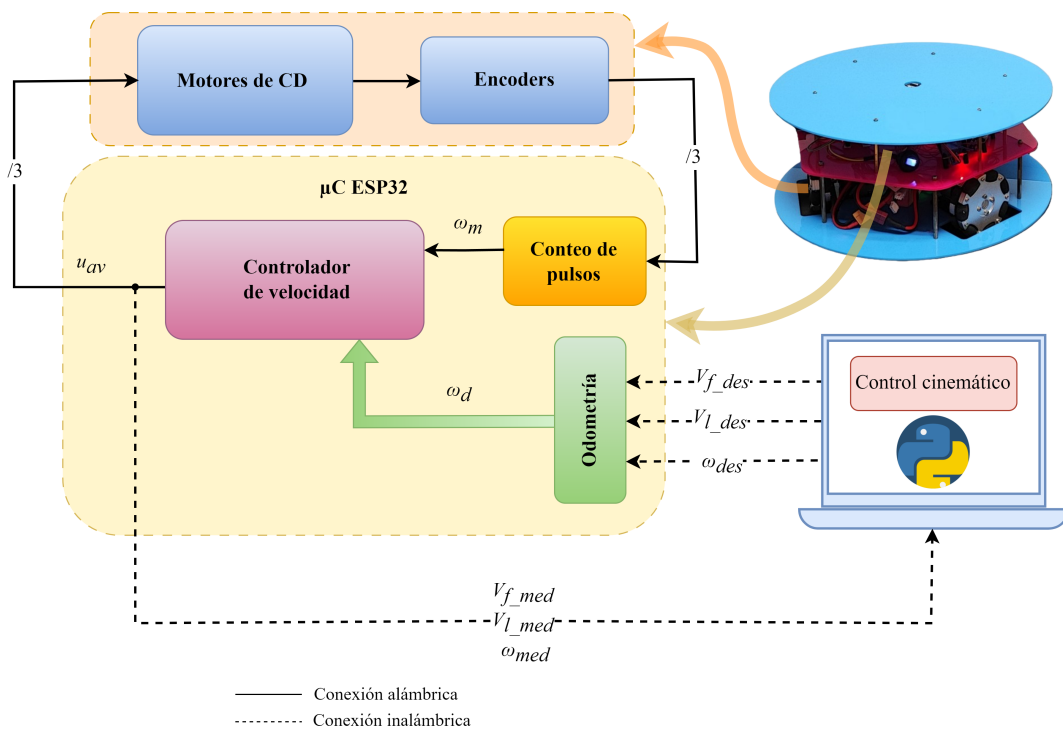


Figura 4.7: Diagrama del funcionamiento del sistema.

Para llevar a cabo la implementación del control cinemático en ROCKY, se desarrollaron dos programas principales. El primero corresponde al código ejecutado en el microcontrolador ESP32, programado en lenguaje C++. Este módulo establece la comunicación inalámbrica con la computadora que ejecuta el controlador cinemático y recibe de él las velocidades frontal, lateral y angular (V_f , V_L y ω). Dichas velocidades se transforman en velocidades angulares de referencia mediante cálculos de odometría implementados directamente en el ESP32. Posteriormente, se aplica una ley de control encargada de regular la velocidad de cada motor, garantizando el seguimiento de las referencias. Finalmente, el ESP32 transmite a la computadora la velocidad actual del robot, cerrando así el lazo de realimentación. El pseudocódigo correspondiente a esta programación se presenta en el Algoritmo 1.

4. Control cinemático

Algoritmo 1 Ejecución en el microcontrolador ESP32

```
1: función INICIALIZACIÓN(Puerto)
2:   Comunicación ← EstablecerComunicación(Puerto)
3:   ConfigurarEncoders()
4:   ConfigurarMotores()
5: fin función
6: función BUCLEDECONTROL
7:   mientras verdadero hacer
8:     VelocidadesDeseadas ← RecibirVelocidades()
9:     VelocidadesMedidas ← LeerEncoders()
10:    VelocidadesRobot ← CalcularOdometría(VelocidadesMedidas)
11:    SeñalControl ← AplicarLeyDeControl(VelocidadesDeseadas, VelocidadesMedidas)
12:    AjustarMotores(SeñalControl)
13:    EnviarDatos(VelocidadesRobot)
14:   fin mientras
15: fin función
```

El segundo programa fue desarrollado en Python y se ejecuta en una computadora externa. Su función principal es implementar el control cinemático del robot de forma inalámbrica. La Tabla 4.1 resume la versión de Python y las principales librerías utilizadas durante el desarrollo. En primer lugar, se establece la comunicación serial con el ESP32, a través de la cual se intercambian datos de manera bidireccional. Una vez activa la conexión, la computadora recibe periódicamente las velocidades actuales del robot —frontal, lateral y angular (V_f , V_L y ω)— medidas por los encoders y transmitidas desde el microcontrolador. Con base en estas velocidades y en la posición deseada, el programa calcula las nuevas referencias mediante el control cinemático y las envía de regreso al ESP32.

Este ciclo se ejecuta de manera continua hasta que el robot alcanza la posición objetivo, en el caso del control de regulación de posición, o hasta que se emite un comando de detención, en el caso del seguimiento de trayectorias. El pseudocódigo correspondiente a esta implementación se presenta en el Algoritmo 2.

Algoritmo 2 Control cinemático y comunicación serial en ordenador (Python)

```
función INICIALIZACIÓN(Puerto)
2:   Comunicación ← EstablecerComunicación(Puerto)
     InicializarEstado()
4:   ConfigurarReferencia()
fin función
6: función BUCLEDECONTROL
     mientras verdadero hacer
8:     ReferenciaActual ← LeerReferencia()
     EstadoActual ← LeerEstado()
10:    Error ← CalcularError(ReferenciaActual, EstadoActual)
     VelocidadesDeseadas ← CalcularVelocidades(Error)
12:    EnviarVelocidades(VelocidadesDeseadas)
     VelocidadesMedidas ← RecibirVelocidades()
14:    ActualizarEstado(VelocidadesMedidas)
     GuardarDatos()
16:    MostrarDatosConsola()
     EsperarTiempoMuestreo()
18:  fin mientras
fin función
20: función FINALIZAR
     EnviarVelocidades([0,0,0])
22:  CerrarComunicación()
     GenerarGráficas()
24: fin función
```

Tabla 4.1: Entorno de desarrollo.

Librería	Versión
python	3.8.6
numpy	1.24.4
matplotlib	3.7.5
pyserial	3.5

En conjunto, los dos programas conforman el lazo de control en tiempo real del robot. El ESP32 actúa como unidad de bajo nivel, encargado del control de velocidad, la odometría y la gestión de

4. Control cinemático

los encoders, mientras que la computadora externa ejecuta el control cinemático, generando las referencias de movimiento a partir de la posición deseada. La comunicación inalámbrica entre ambos asegura el intercambio continuo de datos, permitiendo que el robot pueda regular su posición o seguir trayectorias en condiciones experimentales de manera estable y precisa. Las pruebas experimentales se realizaron en el Laboratorio de Aprendizaje Automático e Interacción Hombre Robot (LAAIHR), cuyo suelo presenta ligeros desniveles debido a la separación entre las lozas que lo conforman. Durante los ensayos con el control cinemático, se implementó un controlador PID para regular la velocidad de cada motor, partiendo de los parámetros calculados analíticamente en el Anexo D.4. No obstante, debido a las perturbaciones mecánicas introducidas por la irregularidad de la superficie, dichas ganancias fueron sometidas a un ajuste experimental fino. Esta sintonización final priorizó la suavidad de la respuesta, buscando mitigar las vibraciones inducidas por las uniones de las lozas y evitar oscilaciones bruscas en la corriente de los motores.

La primera prueba consistió en llevar al robot hacia el punto $[1.5, 1]$ m, con una orientación final deseada de 45° . Los resultados de este experimento se presentan a continuación.

La Figura 4.8 muestra la posición inicial y final del robot durante la ejecución de la prueba, destacando además las direcciones positivas de los ejes x y y .

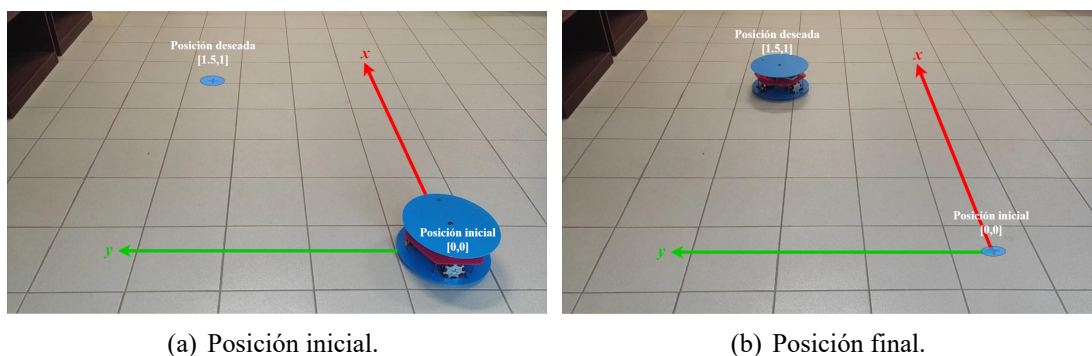


Figura 4.8: Control de posición en el punto $[1.5, 1]$ m y orientación 45° .

En la Figura 4.9 se presenta la trayectoria realizada por el robot desde su posición inicial hasta

el objetivo establecido. La curva en color azul corresponde al recorrido ejecutado, partiendo del marcador verde (posición inicial) y concluyendo en el marcador rojo (posición deseada). Se observa que el robot logra alcanzar la posición objetivo; no obstante, presenta un sobrepaso que requiere una corrección mediante una ligera curva de retorno. El tiempo total empleado para alcanzar el destino fue de aproximadamente 22 segundos.

Para esta prueba se utilizaron los valores $k_p = 0.15$ y $k_i = 0.05$ en las componentes u_x , u_y y u_φ del control cinemático. Dicha sintonización tuvo como objetivo obtener una respuesta transitoria críticamente amortiguada (o ligeramente subamortiguada), evitando oscilaciones bruscas en el inicio del movimiento. El valor de k_p fue determinado para maximizar la velocidad de reacción ante cambios en la referencia, mientras que k_i se incrementó gradualmente hasta anular el error residual de posición, manteniendo el sistema dentro de los márgenes de estabilidad operativa.

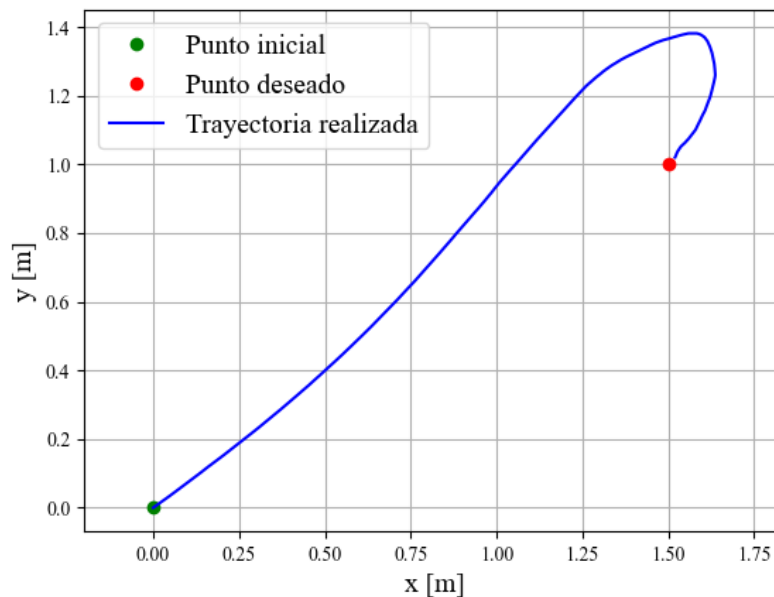


Figura 4.9: Trayectoria realizada por el robot.

No obstante, para evaluar con mayor detalle el desempeño del sistema, se analizó el comporta-

4. Control cinemático

miento de las velocidades de referencia frente a las velocidades reales. En la Figura 4.10 se comparan las velocidades de referencia generadas por el controlador cinemático con las velocidades reales medidas mediante los encoders. Puede apreciarse que las componentes frontal y lateral siguen adecuadamente las referencias, aunque con un leve desfase temporal. Este comportamiento evidencia que el sistema es capaz de reproducir de manera fiel las señales deseadas, manteniendo errores transitorios limitados y una respuesta estable durante la regulación de la posición.

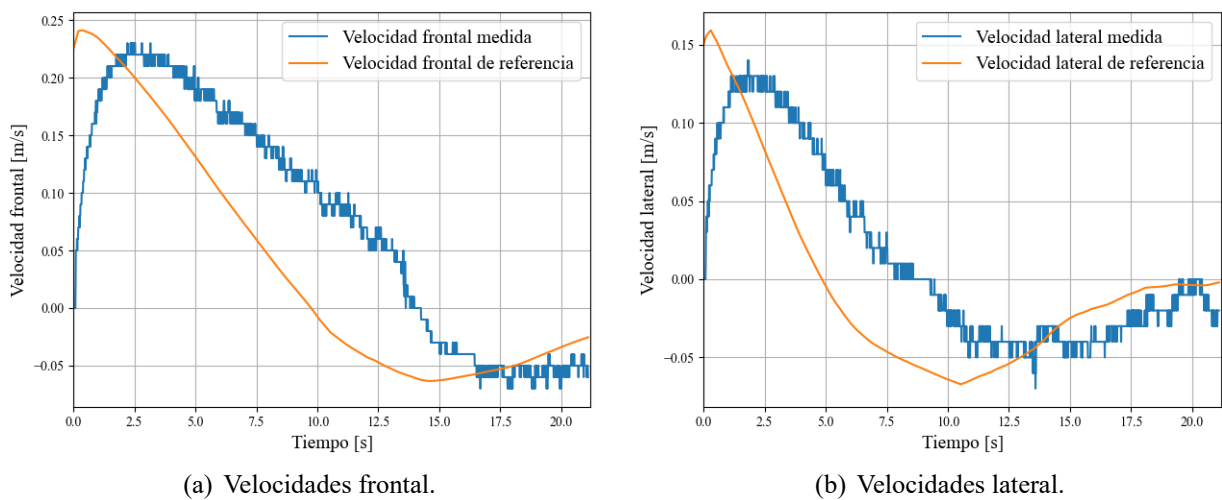


Figura 4.10: Comparativa de las velocidades frontal y lateral.

En la Figura 4.11 se muestra la evolución de la posición en los ejes x e y a lo largo del tiempo, mientras que la Figura 4.12 presenta el comportamiento de la orientación a lo largo del desplazamiento. En la gráfica de la posición se observa una ligera desviación respecto a la posición final deseada. Esta diferencia puede estar relacionada con pequeñas inexactitudes en la ejecución del algoritmo o con las limitaciones propias de la resolución de los sensores empleados. No obstante, dicha desviación permanece dentro de un margen reducido de ± 2 cm.

Por su parte, la gráfica de orientación refleja cómo evolucionó la orientación del robot durante su desplazamiento. Al alcanzar la posición final, la orientación es de aproximadamente 41° , lo cual

demuestra una considerable precisión del sistema para llegar a la orientación deseada.

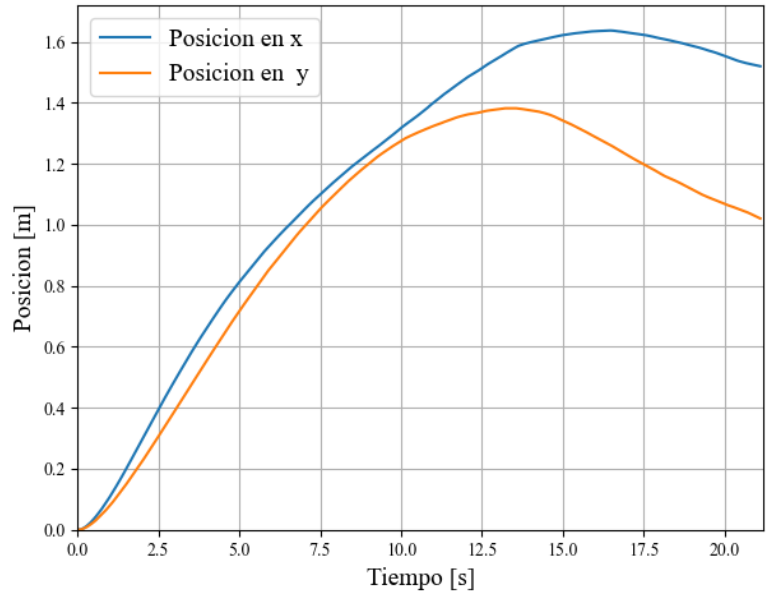


Figura 4.11: Posición en x e y durante el desplazamiento.

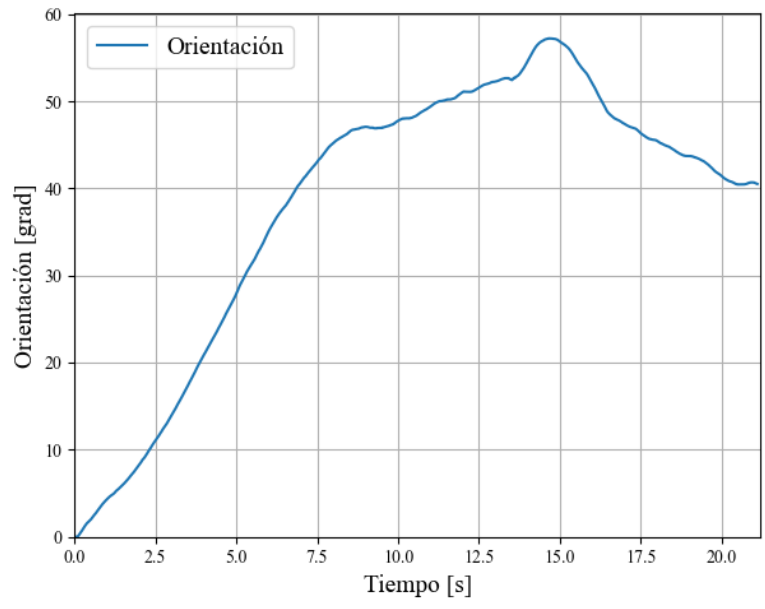


Figura 4.12: Orientación del robot durante el desplazamiento.

4. Control cinemático

Las gráficas anteriores corresponden a los resultados analizados en una sola prueba; sin embargo, para cuantificar la eficacia del control cinemático de posición con PID como regulador de velocidad por motor se realizaron experimentos exhaustivos. Las ganancias del control cinemático fueron las ya descritas; para el PID de los motores se utilizaron $k_p = 15.0$, $k_i = 38.0$ y $k_d = 0.02$.

Los experimentos consistieron en solicitar al robot alcanzar posiciones y orientaciones deseadas seleccionadas para cubrir los cuatro cuadrantes del plano. Se definieron escenarios con las configuraciones deseadas (x_d, y_d, ϕ_d) distribuidas en los cuatro cuadrantes del plano. Cada escenario se ejecutó diez veces, definiendo un tiempo máximo de 35 s para alcanzar la configuración deseada. Un ensayo se consideró exitoso y se detuvo cuando los errores de posición en los ejes x e y fueron ≤ 0.02 m.

Los resultados se resumen en la Tabla 4.2. Conforme al criterio adoptado, todas las métricas reportadas en dicha tabla se calcularon únicamente sobre los ensayos exitosos (también se reporta el porcentaje de éxito por escenario para contextualizar las medias). La Tabla 4.2 resume el desem-

Tabla 4.2: Desempeño del controlador PID en control cinemático de posición.

Métrica	(1.0, 1.0, 30.0°)	(1.0, -1.0, -30.0°)	(-1.0, 1.0, -30.0°)	(-1.0, -1.0, 30.0°)
Éxitos [%]	100.0	60.0	80.0	80.0
\bar{t}_{alcance} [s]	20.846	21.325	20.018	20.203
RMSE_x [m]	0.392	0.396	0.397	0.40
RMSE_y [m]	0.391	0.383	0.386	0.397
RMSE ϕ [grad]	11.48	10.68	11.53	11.99
$e_{\text{final},x}$ [m]	-0.036	-0.036	0.033	0.037
$e_{\text{final},y}$ [m]	-0.036	0.015	-0.022	0.025
$e_{\text{final},\phi}$ [grad]	6.19	-8.78	0.42	-0.45

Nota: \bar{t}_{alcance} : Tiempo promedio de alcance. RMSE: Raíz Cuadrada del Error Cuadrático Medio de la trayectoria. e_{final} : Error final promedio.

peño del controlador, donde las columnas indican la posición y orientación deseadas en el formato $(x[\text{m}], y[\text{m}], \phi[\text{grad}])$. Las métricas de desempeño se definen como:

-
- **Éxitos [%]:** Porcentaje de repeticiones que cumplieron con el criterio de error de posición (≤ 0.02 m).
 - $\bar{t}_{\text{alcance}}[\text{s}]$: El tiempo promedio de alcance del objetivo.
 - **RMSE** (x, y, ϕ): La raíz cuadrada del error cuadrático medio de la trayectoria para cada coordenada.
 - e_{final} (x, y, ϕ): El error final promedio que indica la precisión residual al momento de detener la prueba.

La fila de % de éxitos indica con qué frecuencia se alcanza el objetivo; \bar{t}_{alcance} resume la rapidez. Los RMSE en x, y y ϕ miden el error típico a lo largo de todo el recorrido (incluye el transitorio): valores menores implican seguimiento más estable. Los e_{final} cuantifican la precisión al asentarse (promedio del último segundo). Las divergencias entre RMSE y e_{final} señalan que antes de asentarse, el robot se mueve con oscilaciones grandes o correcciones rápidas debido a sobrepasos de la posición deseada.

Por otro lado, la validación experimental del control cinemático de seguimiento se realizó mediante una prueba con ROCKY. En este experimento se empleó una trayectoria parametrizada en función del tiempo, análoga a la utilizada en la simulación descrita en la Sección 4.5.2. Con el fin de obtener una respuesta suave, se definió una amplitud de $a = 0.25$ m en el eje x y de $a = 0.1$ m en el eje y . Adicionalmente, se aplicó un factor de escala de 0.05 al tiempo, con el propósito de ralentizar la trayectoria y asegurar que el robot pudiera seguirla de manera adecuada. De esta forma, la trayectoria de referencia quedó establecida de acuerdo con (4.12)–(4.13).

$$x_d(t) = 0.25 \cdot \cos(0.05 \cdot t) \quad (4.12)$$

$$y_d(t) = 0.1 \cdot \sin(0.1 \cdot t) \quad (4.13)$$

4. Control cinemático

En la Figura 4.13 se presenta la comparación entre la trayectoria deseada y la trayectoria seguida por el robot. La trayectoria de referencia corresponde a una curva comúnmente conocida como *Lemniscata de Bernoulli*, mientras que la trayectoria real muestra una aproximación a esta con desviaciones visibles, especialmente en las zonas de cambio de curvatura. Por su forma simétrica y continua, este tipo de trayectoria resultó útil para evaluar la capacidad del sistema de control en el seguimiento de trayectorias con cambios suaves y repetitivos de curvatura. A pesar de estas diferencias, el robot logró recorrer una trayectoria cercana a la deseada, manteniendo un error de seguimiento de al rededor de 5 cm.

Para esta prueba se fijó experimentalmente $k_p = 0.45$ en u_x , u_y y u_ϕ . Se prescindió de la acción integral dado que el lazo interno de velocidad compensó eficazmente las perturbaciones, manteniendo el error estacionario en niveles despreciables con control puramente proporcional. El robot inició su movimiento desde el origen $[0, 0]$ m y comenzó el seguimiento de la trayectoria conforme a la referencia establecida.

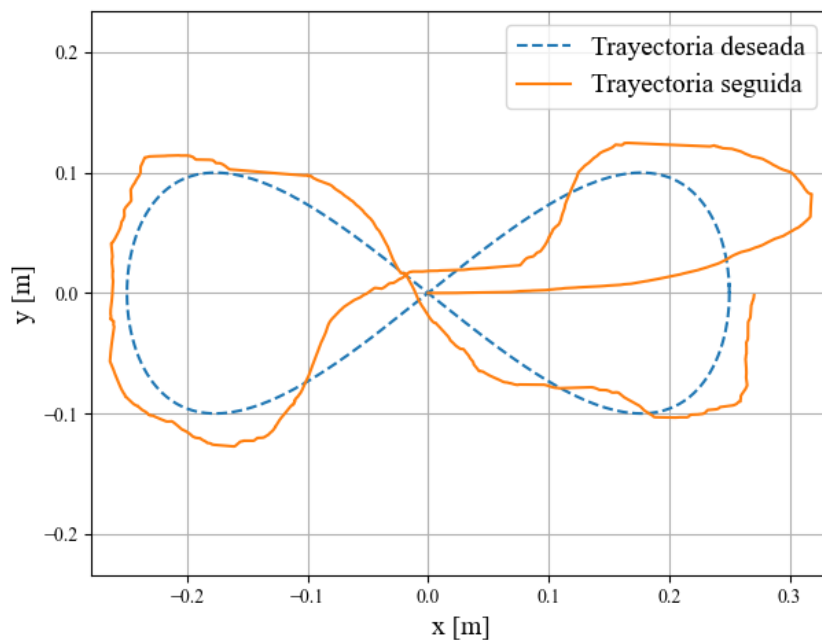


Figura 4.13: Comparativa de las trayectorias.

Asimismo, se analizaron las velocidades de referencia generadas por el control cinemático y las velocidades reales medidas durante la prueba. En la Figura 4.14 se ilustran las comparaciones de las componentes frontal y lateral. En estas gráficas se observa la presencia de desfases entre las señales deseadas y las velocidades medidas, lo que podría explicar que la trayectoria ejecutada no coincida con mayor exactitud con la trayectoria de referencia.

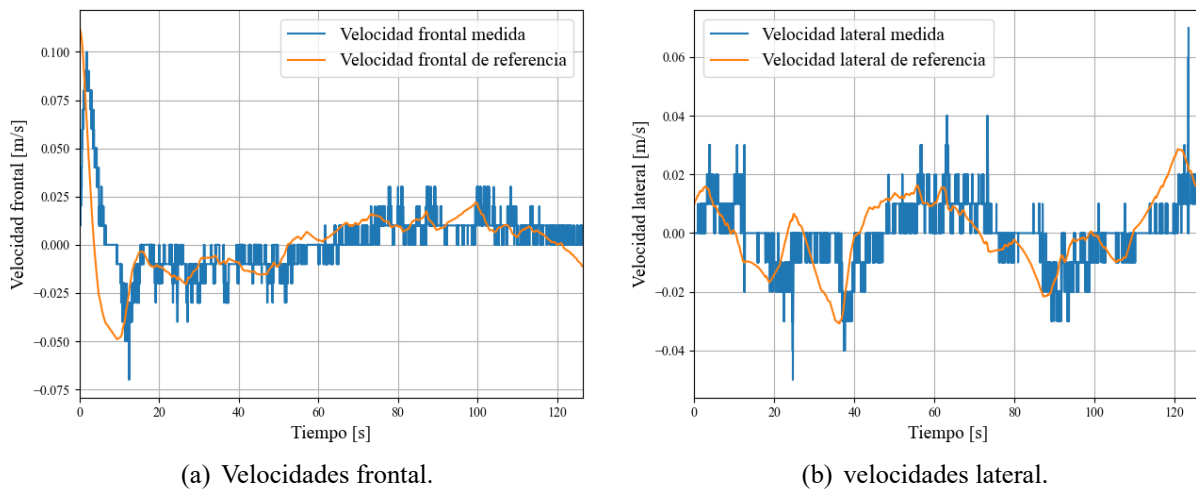


Figura 4.14: Comparativa de las velocidades frontal y lateral en el seguimiento.

En la Figura 4.15 se comparan las posiciones deseadas y medidas en los ejes x e y a lo largo de la trayectoria. Se observa que el robot reprodujo de forma cercana las referencias establecidas y mantuvo una evolución continua en ambos ejes. No obstante, aparecen ligeras discrepancias, principalmente en forma de desfases y pequeñas desviaciones de amplitud, especialmente en los instantes de cambio de dirección. Estos resultados evidencian que el controlador PID logra un seguimiento adecuado de la trayectoria, aunque con limitaciones en la compensación de perturbaciones y en la exactitud al reproducir la referencia, lo cual se refleja en los errores observados.

Por otro lado, la Figura 4.16 ilustra el comportamiento de la orientación del robot a lo largo de la trayectoria. En la gráfica se evidencia que la orientación no se mantuvo constante en el valor de referencia de 0° , presentando fluctuaciones con una amplitud máxima cercana a $\pm 10^\circ$. Aunque

4. Control cinemático

parte de este comportamiento puede atribuirse a perturbaciones externas o retardos en la respuesta de los actuadores, es fundamental considerar las limitaciones del modelo cinemático empleado. Dicho modelo asume condiciones ideales de rodadura pura sin deslizamiento; sin embargo, la interacción física entre las ruedas y la superficie suele inducir pérdidas de tracción. Estos deslizamientos, al no ser contabilizados por la cinemática ideal, generan discrepancias entre el movimiento calculado y el real, lo que contribuye directamente a la deriva observada en la orientación y señala una oportunidad para refinar el esquema de control.

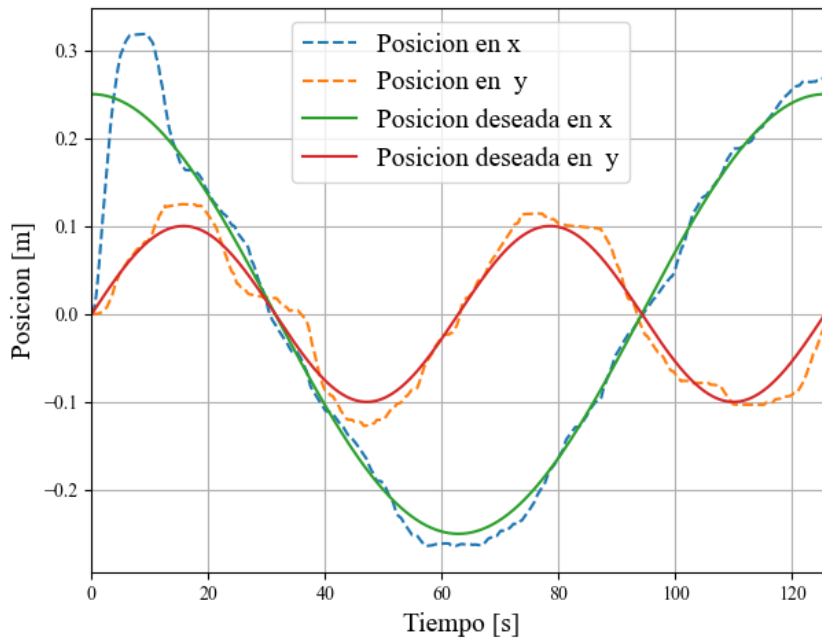


Figura 4.15: Posición del robot durante el seguimiento

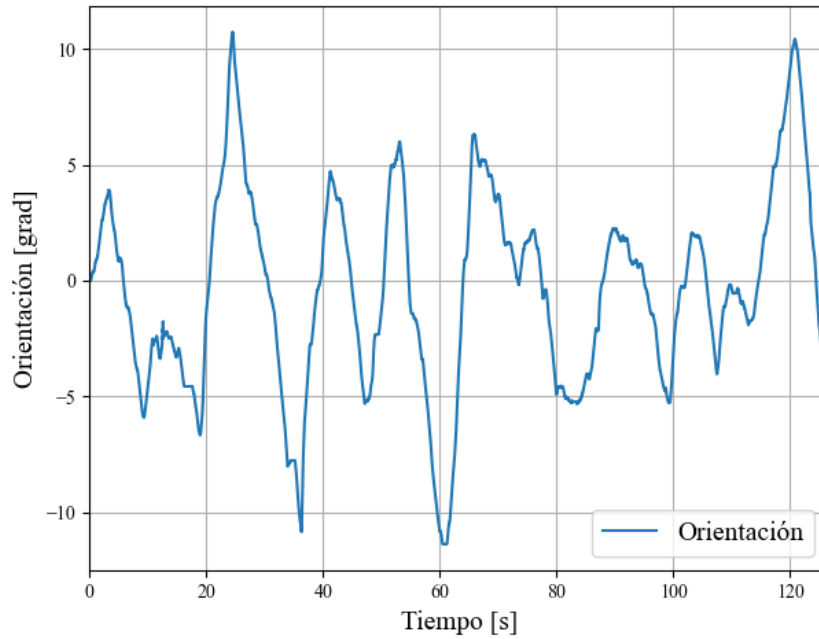


Figura 4.16: Orientación del robot durante el seguimiento.

Uno de los indicadores clave para evaluar el desempeño del controlador es el error de posición. En la Figura 4.17 se presenta su evolución durante el recorrido del robot. Se aprecia que el error osciló en torno a cero, con desviaciones máximas del orden de ± 5 cm, lo que indicó un seguimiento adecuado de la trayectoria con fluctuaciones limitadas. A partir de los errores en los ejes x e y , la Figura 4.18 presenta el error euclidiano de posición, calculado como la distancia entre la posición deseada y la posición medida en cada instante de tiempo. Las gráficas indican que el error de seguimiento de trayectoria se mantuvo acotado dentro de un margen reducido, lo cual respalda la validez del controlador propuesto para las condiciones evaluadas.

4. Control cinemático

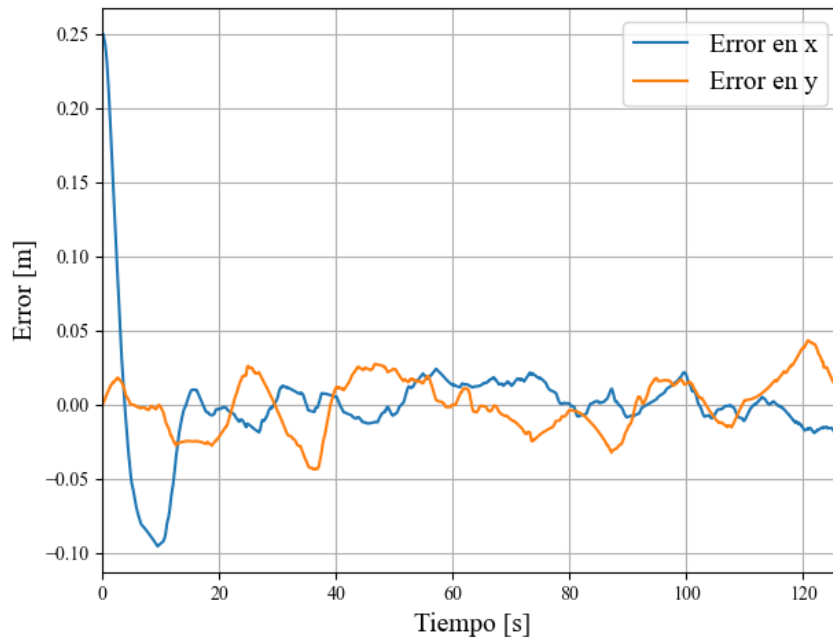


Figura 4.17: Error de posición a lo largo de la trayectoria.

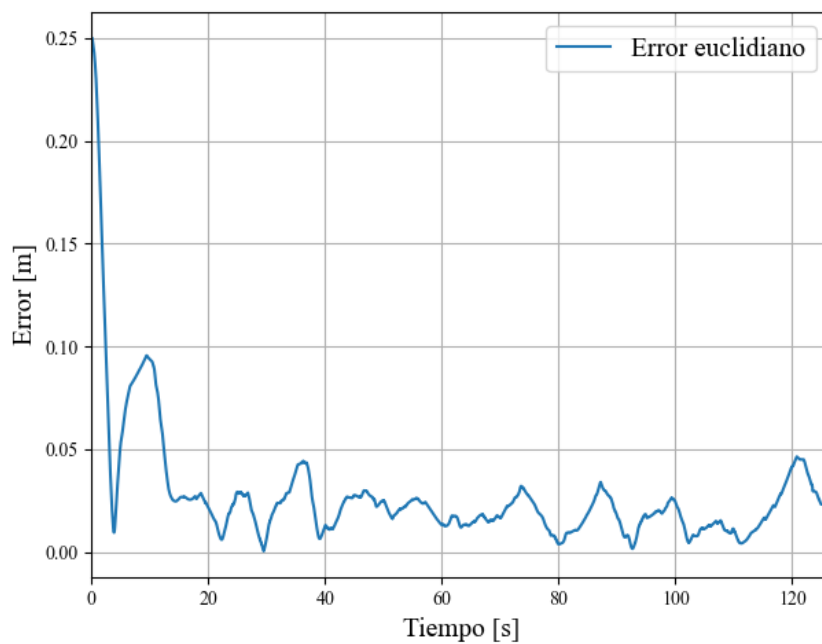


Figura 4.18: Error euclidiano de posición a lo largo de la trayectoria.

Las gráficas anteriores corresponden a los resultados analizados en una sola prueba; sin embargo, para cuantificar la eficacia del control cinemático de seguimiento de trayectorias con PID como regulador de velocidad por motor se realizaron experimentos exhaustivos. Las ganancias del control cinemático y del controlador PID fueron las ya descritas.

En esta ocasión se consideraron dos trayectorias definidas mediante curvas parametrizadas en el tiempo: la lemniscata de Bernoulli y un círculo. Se eligieron por ser trayectorias continuas que presentan zonas críticas asociadas a cambios de cuadrante en el plano. Cada trayectoria se ejecutó diez veces. A diferencia de la prueba individual mostrada anteriormente, aquí el robot no partió del origen del plano: se posicionó directamente sobre la propia trayectoria en un punto correspondiente al instante inicial ($t = 0$). Es decir, el arranque ocurrió ya en el contorno de la curva (sobre el círculo o sobre la lemniscata), evitando la fase de alcance desde el origen. Con ello se minimizaron los errores debidos a la convergencia inicial y se aisló mejor el desempeño de seguimiento. Los resultados se resumen en la Tabla 4.3.

Tabla 4.3: Desempeño del controlador PID en seguimiento de trayectorias.

Métrica	Lemniscata	Círculo
RMSE_x [m]	0.01	0.015
RMSE_y [m]	0.016	0.017
RMSE_pos [m]	0.019	0.022
RMSE ϕ [grad]	4.01	3.57
P95($ e_x $) [m]	0.021	0.027
P95($ e_y $) [m]	0.031	0.031
Tiempo en tolerancia [%]	69.8	60.7
$\ u\ _{L2}$	0.111	0.132

Nota: $RMSE_{pos} = \sqrt{RMSE_x^2 + RMSE_y^2}$. P95 es el percentil 95 del error absoluto por eje. El esfuerzo $L2$ se calcula sobre los comandos normalizados y es comparable entre pruebas con la misma ventana temporal.

En ambas trayectorias, los errores medios por eje se mantienen en el orden de centímetros. La lemniscata muestra, en promedio, menor error y mayor tiempo en tolerancia que el círculo: 69.8 % del tiempo dentro de ± 2 cm en ambos ejes, frente a 60.7 % en el círculo. Los percentiles 95 se

4. Control cinemático

ubican entre 2 cm y 3 cm; es decir, durante el 95 % del tiempo el error absoluto por eje no supera esos valores. El esfuerzo $\|u\|_{L_2}$ también es menor en la lemniscata, lo que sugiere un seguimiento más suave y eficiente con los parámetros actuales.

- **P95 (percentil 95):** Es el valor tal que el 95 % de las muestras queda por debajo o igual a él. Por ejemplo, $P95(|e_x|) = 0.021$ m significa que durante el 95 % del tiempo se cumple $|e_x| \leq 2.1$ cm.
- **Norma $\|u\|_{L_2}$:** Representa la energía de control integrada del vector de comandos u normalizados a $V_{\text{máx}}$ y $\Omega_{\text{máx}}$. Valores menores indican movimientos más suaves en la ventana temporal considerada. En estas pruebas se usaron ventanas de 126 segundos (lemniscata) y 92 segundos (círculo), con $V_{\text{máx}} = 0.60$ m/s y $\Omega_{\text{máx}} = 2.0$ rev/s.

Diseño de controladores

5.1. Diseño del controlador ADRC y observador GPI

Con el objetivo de diseñar un controlador ADRC para el motor de CD, resultó crucial contar con una caracterización precisa de sus parámetros dinámicos y eléctricos. La metodología y los resultados de dicha caracterización se presentan en el Anexo B.2. Además, en el Anexo C.3 se presentan las ecuaciones que describen el comportamiento dinámico del motor de CD, mismas que se utilizan a continuación.

Partiendo de (6)-(8) que describen el modelo dinámico del motor de CD, donde τ_L es el par de perturbación en el eje del motor, el cual se supone variable en el tiempo, pero acotado, aunque de naturaleza desconocida. Eligiendo $i = x_1$ y $\omega_m = x_2$, entonces el modelo queda como:

$$\dot{x}_1 = \frac{1}{L}u - \frac{R_a}{L}x_1 - \frac{K}{L}x_2 \quad (5.1)$$

$$\dot{x}_2 = \frac{K}{J}x_1 - \frac{B}{J}x_2 - \frac{1}{J}\tau_L \quad (5.2)$$

$$y = x_2 \quad (5.3)$$

Considerando el sistema sin perturbar, es decir $\tau_L = 0$, de (5.2) se tiene:

$$\begin{aligned}\frac{K}{J}x_1 &= \dot{x}_2 + \frac{B}{J}x_2 \\ x_1 &= \frac{J}{K}\dot{x}_2 + \frac{B}{K}x_2\end{aligned}\quad (5.4)$$

Sustituyendo (5.4) en (5.1):

$$\begin{aligned}\frac{J}{K}\ddot{x}_2 + \frac{B}{J}\dot{x}_2 &= \frac{1}{L}u - \frac{R_a}{L}\left(\frac{J}{K}\dot{x}_2 + \frac{B}{K}x_2\right) - \frac{K}{L}x_2 \\ \frac{J}{K}\ddot{x}_2 + \frac{B}{J}\dot{x}_2 &= \frac{1}{L}u - \frac{R_a J}{LK}\dot{x}_2 - \frac{R_a B}{LK}x_2 - \frac{K}{L}x_2 \\ u &= \frac{JL}{K}\ddot{x}_2 + \left(\frac{BL}{K} + \frac{R_a J}{K}\right)\dot{x}_2 + \left(\frac{R_a B}{K} + K\right)x_2\end{aligned}\quad (5.5)$$

No es difícil apreciar de (5.4) y (5.5) que las variables de estado y la entrada de control son funciones diferenciables de x_2 , por lo que dicha variable es una salida plana, y así el sistema es diferencialmente plano. Ahora, considerando nuevamente el modelo perturbado (5.1)-(5.3), derivando sucesivamente (5.3) nos queda que:

$$\dot{y} = \dot{x}_2$$

Es decir:

$$\dot{y} = \frac{K}{J}x_1 - \frac{B}{J}x_2 - \frac{1}{J}\tau_L$$

Derivando nuevamente:

$$\ddot{y} = \frac{K}{J}\dot{x}_1 - \frac{B}{J}\dot{x}_2 - \frac{1}{J}\dot{\tau}_L\quad (5.6)$$

Usando (5.1) en (5.6)

$$\begin{aligned}\ddot{y} &= \frac{K}{J}\left(\frac{1}{L}u - \frac{R_a}{L}x_1 - \frac{K}{L}x_2\right) - \frac{B}{J}\dot{x}_2 - \frac{1}{J}\dot{\tau}_L \\ \ddot{y} &= \frac{K}{JL}u - \frac{KR_a}{JL}x_1 - \frac{K^2}{JL}x_2 - \frac{B}{J}\dot{x}_2 - \frac{1}{J}\dot{\tau}_L\end{aligned}\quad (5.7)$$

5. Diseño de controladores

Despejando para x_1 en (5.2) se tiene:

$$\begin{aligned}\frac{K}{J}x_1 &= \dot{x}_2 + \frac{B}{J}x_2 + \frac{1}{J}\tau_L \\ x_1 &= \frac{J}{K}\dot{x}_2 + \frac{B}{K}x_2 + \frac{1}{K}\tau_L\end{aligned}\quad (5.8)$$

Sustituyendo (5.8) en (5.7) se tiene:

$$\begin{aligned}\ddot{y} &= \frac{K}{JL}u - \frac{KR_a}{JL}\left(\frac{B}{K}x_2 + \frac{J}{L}\dot{x}_2 + \frac{1}{K}\tau_L\right) - \frac{K^2}{JL}x_2 - \frac{B}{J}\dot{x}_2 - \frac{1}{J}\dot{\tau}_L \\ \ddot{y} &= \frac{K}{JL}u - \frac{R_aB}{JL}x_2 - \frac{R_a}{L}\dot{x}_2 - \frac{R_a}{JL}\tau_L - \frac{K^2}{JL}x_2 - \frac{B}{J}\dot{x}_2 - \frac{1}{J}\dot{\tau}_L \\ \ddot{y} &= \frac{K}{JL}u - \left(\frac{R_aB}{JL} + \frac{K^2}{JL}\right)x_2 - \left(\frac{R_a}{L} + \frac{B}{J}\right)\dot{x}_2 - \frac{R_a}{JL}\tau_L - \frac{1}{J}\dot{\tau}_L \\ \ddot{y} &= \frac{K}{JL}u - \left(\frac{R_aB}{JL} + \frac{K^2}{JL}\right)y - \left(\frac{R_a}{L} + \frac{B}{J}\right)\dot{y} - \frac{R_a}{JL}\tau_L - \frac{1}{J}\dot{\tau}_L\end{aligned}\quad (5.9)$$

Seleccionando como variables de fase a $y_1 = y = \omega_m$ y $y_2 = \dot{y} = \dot{\omega}_m$, entonces (5.9) se puede escribir como:

$$\dot{y}_1 = y_2 \quad (5.10)$$

$$\dot{y}_2 = \frac{K}{JL}u + \varphi \quad (5.11)$$

Donde en (5.11):

$$\varphi(y_1, y_2, \tau_L, \dot{\tau}_L) = \left(\frac{R_aB}{JL} + \frac{K^2}{JL}\right)y - \left(\frac{R_a}{L} + \frac{B}{J}\right)\dot{y} - \frac{R_a}{JL}\tau_L - \frac{1}{J}\dot{\tau}_L \quad (5.12)$$

representa la perturbación del sistema. Por lo tanto, (5.11) puede escribirse como:

$$\dot{y}_2 = bu + \varphi \quad (5.13)$$

Entonces, de acuerdo con Sira-Ramírez *et al.* [34], la expresión que representa la ley del control es:

$$u = \frac{1}{b} \left\{ [y^*(t)]'' - \sum_{j=0}^1 [k_j(y_{j+1} - y^*(t))^{(j)} - \hat{\varphi}(t)] \right\} \quad (5.14)$$

Para el caso de un sistema de segundo orden, (5.14) queda como:

$$u = \frac{1}{b} [\ddot{y}^* - k_0(y_1 - y^*) - k_1(y_2 - \dot{y}^*) - \hat{\varphi}(t)] \quad (5.15)$$

donde y^* es la velocidad de referencia ω_d .

El diseño del controlador propuesto se fundamenta bajo las siguientes suposiciones:

- La única variable que puede medirse directamente es la posición angular, a partir de la cual se calcula luego la velocidad angular ω_m .
- Los valores nominales de los parámetros del modelo dinámico del motor de CD (L, R_a, K, J, B) son conocidos.
- La función de perturbación total $\varphi(y_1, y_2, \tau_L, \dot{\tau}_L)$ y sus derivadas, son uniformemente absolutamente acotadas, i.e. $\sup |\varphi^{(j)}(t)| = \|\varphi^{(j)}(t)\|_\infty \leq \kappa_j$ para toda $0 \leq j \leq m$. En otras palabras, la perturbación total es acotada.

En la literatura del ADRC, el enfoque clásico se apoya en el observador de estado extendido (ESO, por sus siglas en inglés) para la estimación de dinámicas no modeladas y perturbaciones externas [35, 36]. Si bien esta herramienta permite controlar sistemas sin depender de un modelo matemático preciso, su estructura no lineal suele añadir complejidad a la implementación práctica.

Como alternativa, Sira-Ramírez *et al.* [34, 37] proponen el uso del observador proporcional integral generalizado (GPIO). A diferencia del ESO, el GPIO opera bajo una estructura lineal, lo que simplifica significativamente su diseño y puesta en marcha. La principal ventaja de este enfoque

5. Diseño de controladores

radica en su capacidad para aproximar perturbaciones, tanto exógenas como dependientes del estado, mediante un modelo polinomial en el tiempo auto-actualizable, garantizando que el error de estimación converja asintóticamente a una vecindad arbitrariamente pequeña de cero.

Bajo esta premisa, y con la finalidad de estimar la perturbación definida en (5.11) y (5.12), se propone la implementación de un observador GPI, cuya estructura general se detalla en la Sección 2.3.1. Al particularizar la ecuación (2.6) para el sistema de segundo orden bajo estudio, el observador queda definido como:

$$\dot{\hat{y}}_1 = \hat{y}_2 + \lambda_3 (y_1 - \hat{y}_1) \quad (5.16)$$

$$\dot{\hat{y}}_2 = \frac{K}{JL}u + z_1 + \lambda_2 (y_1 - \hat{y}_1) \quad (5.17)$$

$$\dot{z}_1 = z_2 + \lambda_1 (y_1 - \hat{y}_1) \quad (5.18)$$

$$\dot{z}_2 = \lambda_0 (y_1 - \hat{y}_1) \quad (5.19)$$

Donde \hat{y}_1 es el estimado de la velocidad angular, \hat{y}_2 es el estimado de la aceleración angular. El estado z_1 produce un estimado próximo a la señal incierta φ . Usando (5.10), (5.11) y (5.16)-(5.19) puede verse que el error de observación de salida, $e_1 = y_1 - \hat{y}_1$, evoluciona de acuerdo a:

$$\dot{e}_1 = e_2 - \lambda_3 e_1 \quad (5.20)$$

$$\dot{e}_2 = \varphi - z_1 - \lambda_2 e_1 \quad (5.21)$$

$$\dot{z}_1 = z_2 + \lambda_1 e_1 \quad (5.22)$$

$$\dot{z}_2 = \lambda_0 e_1 \quad (5.23)$$

Derivando (5.20) respecto al tiempo:

$$\ddot{e}_1 = \dot{e}_2 - \lambda_3 \dot{e}_1 \quad (5.24)$$

Sustituyendo (5.21) en (5.24)

$$\ddot{e}_1 = \varphi - z_1 - \lambda_2 e_1 - \lambda_3 \dot{e}_1 \quad (5.25)$$

Derivando (5.25)

$$\overset{(3)}{e}_1 = \dot{\varphi} - \dot{z}_1 - \lambda_2 \dot{e}_1 - \lambda_3 \ddot{e}_1 \quad (5.26)$$

Sustituyendo (5.22) en (5.26)

$$\overset{(3)}{e}_1 = \dot{\varphi} - z_2 - \lambda_1 e_1 - \lambda_2 \dot{e}_1 - \lambda_3 \ddot{e}_1 \quad (5.27)$$

Derivando (5.27)

$$\overset{(4)}{e}_1 = \ddot{\varphi} - \dot{z}_2 - \lambda_1 \dot{e}_1 - \lambda_2 \ddot{e}_1 - \lambda_3 \overset{(3)}{e}_1 \quad (5.28)$$

Sustituyendo (5.23) en (5.28)

$$\overset{(4)}{e}_1 = \ddot{\varphi} - \lambda_0 e_1 - \lambda_1 \dot{e}_1 - \lambda_2 \ddot{e}_1 - \lambda_3 \overset{(3)}{e}_1$$

Reacomodando

$$\overset{(4)}{e}_1 + \lambda_3 \overset{(3)}{e}_1 + \lambda_2 \ddot{e}_1 + \lambda_1 \dot{e}_1 + \lambda_0 e_1 = \ddot{\varphi} \quad (5.29)$$

Usando la transformada de Laplace puede determinarse que el polinomio característico de la parte predominantemente lineal de (5.29) está dado por:

$$p(s) = s^4 + \lambda_3 s^3 + \lambda_2 s^2 + \lambda_1 s + \lambda_0 \quad (5.30)$$

Proponiendo como polinomio deseado a

$$\begin{aligned} p_d(s) &= \left(s^2 + 2\zeta\omega_{n_0}s + \omega_{n_0}^2 \right)^2 \\ p_d(s) &= s^4 + 4\zeta\omega_{n_0}s^3 + \left(4\zeta^2\omega_{n_0}^2 + 2\omega_{n_0}^2 \right) s^2 + 4\zeta\omega_{n_0}^3 s + \omega_{n_0}^4 \end{aligned} \quad (5.31)$$

5. Diseño de controladores

Comparando (5.30) con (5.31)

$$\lambda_3 = 4\zeta\omega_{n0} \quad (5.32)$$

$$\lambda_2 = 4\zeta^2\omega_{n0}^2 + 2\omega_{n0}^2$$

$$\lambda_1 = 4\zeta^2\omega_{n0}^3$$

$$\lambda_0 = \omega_{n0}^4$$

Con ζ siendo el factor de amortiguamiento y ω_{n0} la frecuencia natural del sistema.

5.1.1. Análisis de estabilidad del observador

Haciendo la siguiente selección de variables:

$$x_1 = e_1$$

$$x_2 = \dot{e}_1$$

$$x_3 = \ddot{e}_1$$

$$x_4 = e_1^{(3)}$$

$$x_5 = e_1^{(4)}$$

Se tiene que:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = x_3$$

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = x_5$$

$$\dot{x}_5 = -\lambda_4 x_5 - \lambda_3 x_4 - \lambda_2 x_3 - \lambda_1 x_2 - \lambda_0 x_1 + \varphi^{(3)}$$

Por lo que (5.29) se puede escribir en la forma:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \quad (5.33)$$

$$e_1 = \mathbf{C}\mathbf{x} \quad (5.34)$$

donde:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -\lambda_0 & -\lambda_1 & -\lambda_2 & -\lambda_3 & -\lambda_4 \end{pmatrix}; \quad \mathbf{B} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}; \quad u = \overset{(3)}{\varphi}$$

$$\mathbf{C} = (1 \ 0 \ 0 \ 0 \ 0); \quad \mathbf{x} = (x_1 \ x_2 \ x_3 \ x_4 \ x_5)^T$$

De acuerdo con Sontag y Wang [38], y Marquez [39] se tiene la siguiente definición:

Definición 1 *El sistema*

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \quad (5.35)$$

$$y = \mathbf{C}\mathbf{x} \quad (5.36)$$

se dice localmente estable de entrada al estado (ISS) si existe una función β de clase \mathcal{KL} , y una función γ de clase \mathcal{K} y constantes $K_1, K_2 \in \mathbb{R}_+$, tal que:

$$e_1(t) \leq \beta(\|\mathbf{x}(0)\|, t) + \gamma(\|u\|_\infty), \quad \forall t$$

para todo $\mathbf{x}(0) \in D$ y $u \in D_u$ que satisfacen: $\|\mathbf{x}(0)\| < K_1$, y $\sup_{t>0} \|u\| < K_2$.

5. Diseño de controladores

La conocida fórmula de variación de parámetros da la siguiente solución a la ecuación de estado dada en (5.33)-(5.34):

$$\mathbf{x}(t) = \exp(\mathbf{A}t) \mathbf{x}(0) + \int_0^t \exp(\mathbf{A}(t - \tau)) \mathbf{B}u d\tau \quad (5.37)$$

Usando (5.37) en (5.34) se tiene que:

$$e_1(t) = \mathbf{C} \exp(\mathbf{A}t) \mathbf{x}(0) + \mathbf{C} \int_0^t \exp(\mathbf{A}(t - \tau)) \mathbf{B}u d\tau \quad (5.38)$$

Si el conjunto de coeficientes $\{\lambda_4, \lambda_3, \lambda_2, \lambda_1, \lambda_0\}$ se elige de manera tal que la matriz \mathbf{A} es Hurwitz, entonces es posible establecer la siguiente desigualdad:

$$\begin{aligned} e_1(t) &\leq \|\mathbf{C} \exp(\mathbf{A}t) \mathbf{x}(0)\| + \left\| \mathbf{C} \int_0^t \exp(\mathbf{A}(t - \tau)) \mathbf{B}u d\tau \right\| \\ &\leq \underbrace{\|\exp(\mathbf{A}t)\| \|\mathbf{x}(0)\|}_{\beta(\|\mathbf{x}(0)\|, t) \in \mathcal{KL}} + \underbrace{\|\mathbf{B}\| \|u\| \int_0^t \|\exp(\mathbf{A}(t - \tau))\| d\tau}_{\gamma(\|u\|_\infty) \in \mathcal{K}_\infty} \end{aligned} \quad (5.39)$$

Ya que existe $\beta \in \mathcal{KL}$ y $\gamma \in \mathcal{K}_\infty$, entonces, de acuerdo con la Definición 1, la dinámica del error de observación es ISS, es decir, las soluciones para la dinámica del error de observación están acotadas por $\gamma(\|u\|_\infty)$ y el sistema es asintóticamente estable a $e_1(t) = 0$ para $u = 0$. Además, si el conjunto de coeficientes $\{\lambda_4, \lambda_3, \lambda_2, \lambda_1, \lambda_0\}$ se eligen de modo tal que la matriz \mathbf{A} es Hurwitz, con valores propios en orden creciente $l_1(\mathbf{A}) < l_2(\mathbf{A}) < l_3(\mathbf{A}) < l_4(\mathbf{A}) < l_5(\mathbf{A})$, se sabe que, $\|\exp(\mathbf{A}t)\| \leq \alpha \exp(-l_1 t)$, con $\alpha = \|\mathbf{T}\| \|\mathbf{T}^{-1}\|$, donde \mathbf{T} es una matriz tal que $\mathbf{T}^{-1} \mathbf{A} \mathbf{T}$ es diagonal. Usando el hecho anterior en (5.39), se obtiene:

$$\begin{aligned} e_1(t) &\leq \exp(-l_1 t) \|\mathbf{x}(0)\| + \alpha \kappa_3 \int_0^t \exp(-l_1(t - \tau)) d\tau \\ &\leq \exp(-l_1 t) \|\mathbf{x}(0)\| + \frac{\alpha \kappa_3}{l_1} (1 - \exp(-l_1 t)) \end{aligned} \quad (5.40)$$

Consecuentemente $\|e_1(t)\|$ converge exponencialmente a una esfera de radio $\rho = \frac{\alpha \kappa_3}{l_1}$ conforme

$t \rightarrow \infty$. Donde κ_3 es la cota superior para la tercera derivada de la perturbación total φ , es decir, se cumple que $\|u\| = \left\| \begin{matrix} (3) \\ \varphi \end{matrix} \right\|_{\infty} \leq \kappa_3$

Observación 1 *El hecho de que la dinámica del error de observación tenga la propiedad de estabilidad de entrada al estado (ISS) permite establecer una especie de principio de separación que asegura que las ganancias del observador $\{\lambda_4, \lambda_3, \lambda_2, \lambda_1, \lambda_0\}$ se pueden elegir en forma independiente sin que sus valores afecten la selección de las ganancias del controlador.*

Observación 2 *El primer término en (5.40) puede dominar para t pequeño, y esto sirve para cuantificar la magnitud del comportamiento transitorio del observador como una función del tamaño inicial del estado $\mathbf{x}(0)$ y del valor l_1 . Esta propiedad permite elegir de manera juiciosa al conjunto de coeficientes $\{\lambda_4, \lambda_3, \lambda_2, \lambda_1, \lambda_0\}$ a fin de tener un término transitorio aceptable.*

Observación 3 *El primer término en (5.40) se desvanece para un t suficientemente grande. Así, la dinámica del error de observación satisface la propiedad de ganancia asintótica. Es decir, para todo t suficientemente grande, la trayectoria existe y se acerca arbitrariamente a una esfera cuyo radio es proporcional a la cota de u e inversamente proporcional a l_1 . En el lenguaje del control robusto, la estimación podría ser denominada una condición de límite final.*

5.2. Simulación del control ADRC y el observador GPI

Una vez que se comprobó que los parámetros obtenidos en el Anexo B.2 eran correctos, lo siguiente fue simular el sistema en lazo cerrado. Para esto se implementó la ecuación (5.15) del control ADRC y las ecuaciones del observador GPI.

5. Diseño de controladores

En principio y^* es constante en este caso, por lo que (5.15) queda como:

$$u = \frac{1}{b} [-k_0 (y_1 - y^*) - k_1 (y_2) - \hat{\varphi}] \quad (5.41)$$

En la Figura 5.1 se presenta el diagrama de bloques utilizado para la simulación de los motores en lazo cerrado con el control ADRC y el observador GPIO, implementados en Simulink®. En la simulación, se buscó analizar el comportamiento de los motores cuando fueron sometidos a una perturbación de naturaleza constante y acotada.

La Figura 5.2 muestra el comportamiento de la velocidad de cada motor en comparación con la velocidad de referencia, la cual es de 10 rad/s. Adicionalmente, se introdujo una perturbación constante de 0.25 Nm, aplicada de forma súbita en el instante 0.02 segundos para analizar su efecto en el sistema.

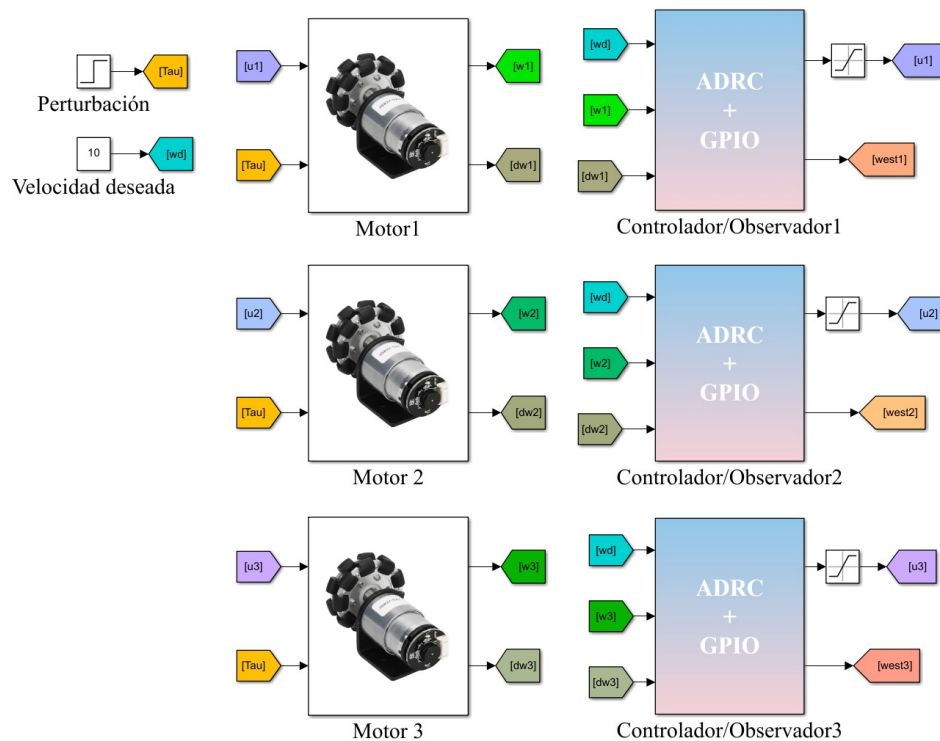


Figura 5.1: Diagrama de bloques del sistema en lazo cerrado.

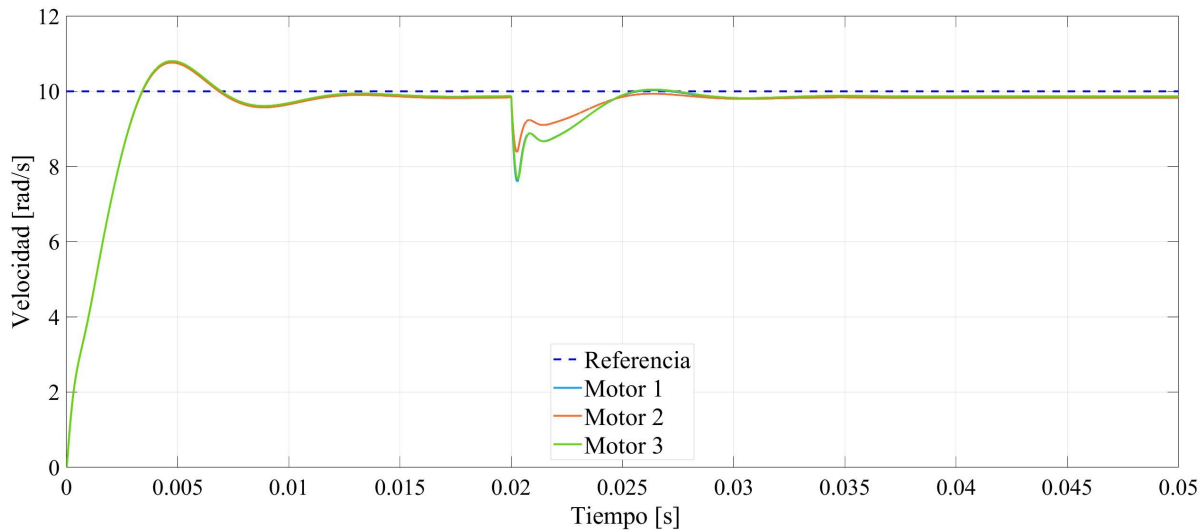


Figura 5.2: Velocidades vs referencia con controlador ADRC y GPIO.

Como se observa en la Figura 5.2, el control ADRC demostró una robustez suficiente para retornar el sistema a su referencia. Esto evidencia que el observador GPIO respondió con la rapidez necesaria para estimar la perturbación y asistir al controlador en la estabilización del sistema, incluso frente a la perturbación introducida.

Para evaluar el desempeño del controlador, se analizó la gráfica del Error Cuadrático Integral (ISE), que muestra la evolución del error acumulado entre la velocidad de referencia y la velocidad medida. En la Figura 5.3 se presenta el ISE correspondiente al motor 1, ya que, al tratarse de controladores idénticos para los tres motores, el análisis de uno de ellos fue representativo del comportamiento general.

5. Diseño de controladores

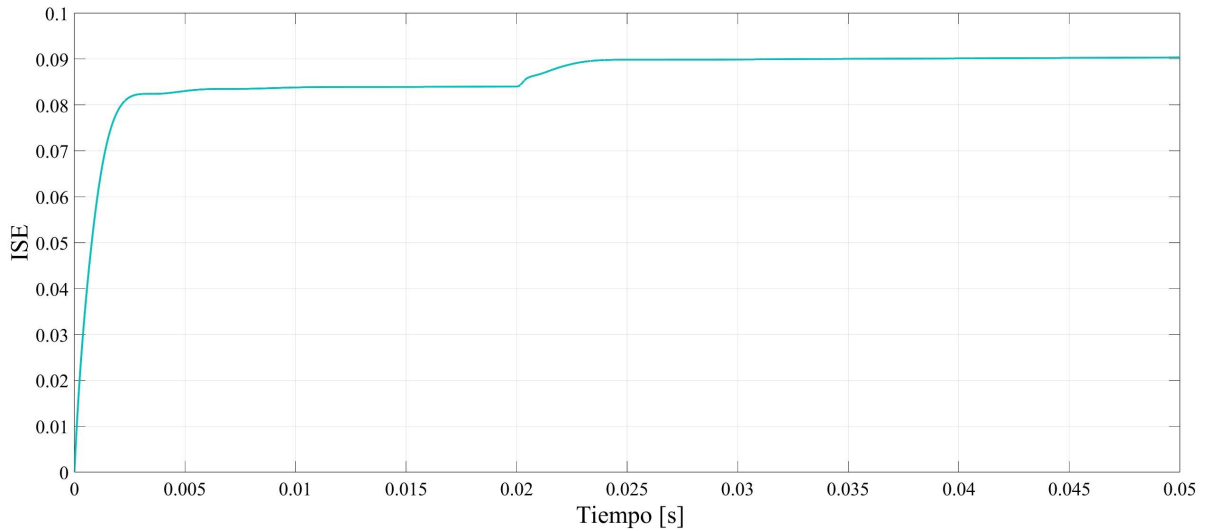


Figura 5.3: ISE del Motor 1 con el controlador ADRC.

A continuación, se analizó el caso en el que la velocidad deseada se definió a partir de una trayectoria suavizada mediante un polinomio de Bézier. Para ello, se incorporó en la simulación un bloque encargado de generar dicha curva, la cual estableció la referencia de velocidad que deben seguir los motores. La elección de este enfoque radicó en la capacidad del polinomio de Bézier para producir trayectorias continuas y controlables. A diferencia de funciones lineales o cuadráticas, esta formulación permitió ajustar los puntos inicial, final y de control, garantizando transiciones suaves y evitando discontinuidades en la velocidad o la aceleración.

Los parámetros definidos fueron los siguientes: tiempo inicial ($t_i = 0.05$ seg), el tiempo final ($t_f = 1$ seg) —instante en el que se espera alcanzar la velocidad deseada—, velocidad inicial ($\omega_i = 0$ rad/seg), y la velocidad final deseada ($\omega_d = 10$ rad/seg). En la Figura 5.4 se muestra la respuesta de los motores frente a la velocidad de referencia generada mediante el polinomio de Bézier. El gráfico incluye, además, el efecto de una perturbación súbita aplicada en el tiempo $t = 1$ s con una magnitud de 0.25 Nm.

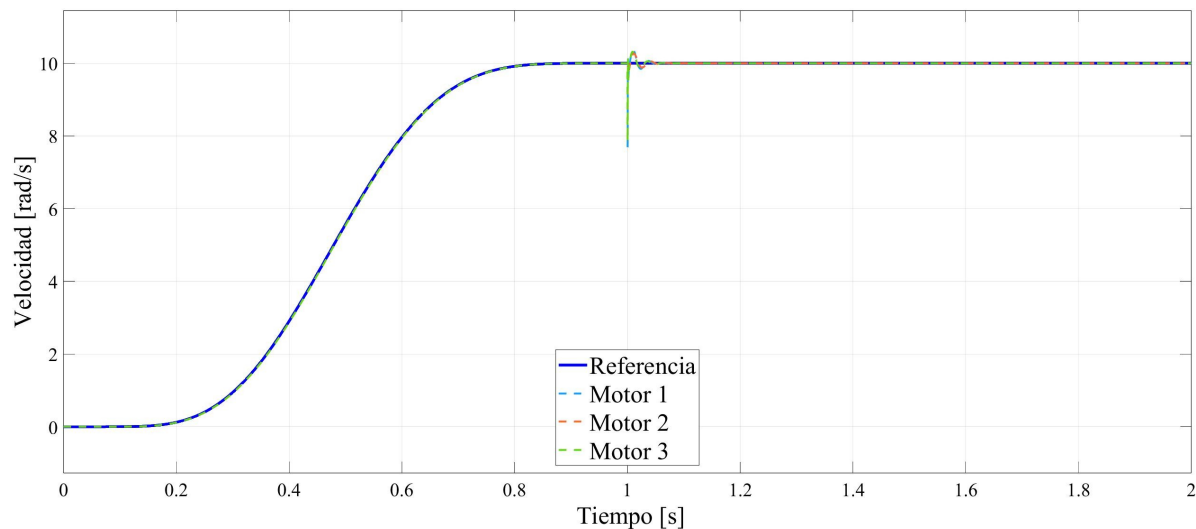


Figura 5.4: Velocidades vs referencia suavizada.

Como se aprecia en la Figura 5.4, el controlador logró que los motores alcanzaran la señal de referencia con alta precisión, aun bajo la acción de la perturbación. Esta conservó la misma magnitud que la considerada previamente en la Sección 5.2, cuando la velocidad de referencia se mantuvo constante. Con este resultado se buscó validar el diseño del controlador ADRC en un entorno simulado, previo a su implementación en el microcontrolador a bordo del robot.

5.3. Diseño y simulación del control PID

En la primera etapa de este trabajo de tesis, el control cinemático se implementó utilizando un controlador PID para regular la velocidad de los motores. La integración de este controlador en el sistema ROCKY se llevó a cabo en el marco del control cinemático descrito en la Sección 4.6. Inicialmente, se realizaron simulaciones en Matlab/Simulink[®] empleando el modelo dinámico caracterizado, donde se calcularon los parámetros del PID siguiendo la metodología de asignación de polos detallada en el Anexo D.4. Posteriormente, dichas ganancias fueron refinadas experimentalmente durante las pruebas de control de posición (Sección 4.5.1), con el propósito de asegurar la regulación precisa de la velocidad ante las referencias generadas por el control cinemático.

En la Figura 5.5 se muestra el diagrama de bloques utilizado para la simulación del control PID implementado en Simulink[®]. En la simulación, se buscó analizar el comportamiento de los motores cuando fueron sometidos a una perturbación de naturaleza constante y acotada.

En la Figura 5.6 se muestra el comportamiento de la velocidad de cada motor en comparación con la velocidad de referencia, la cual es de 10 rad/s. Adicionalmente, se introdujo una perturbación constante de $0.25 \text{ N} \cdot \text{m}$, aplicada de forma súbita en un instante específico del tiempo para analizar su efecto en el sistema.

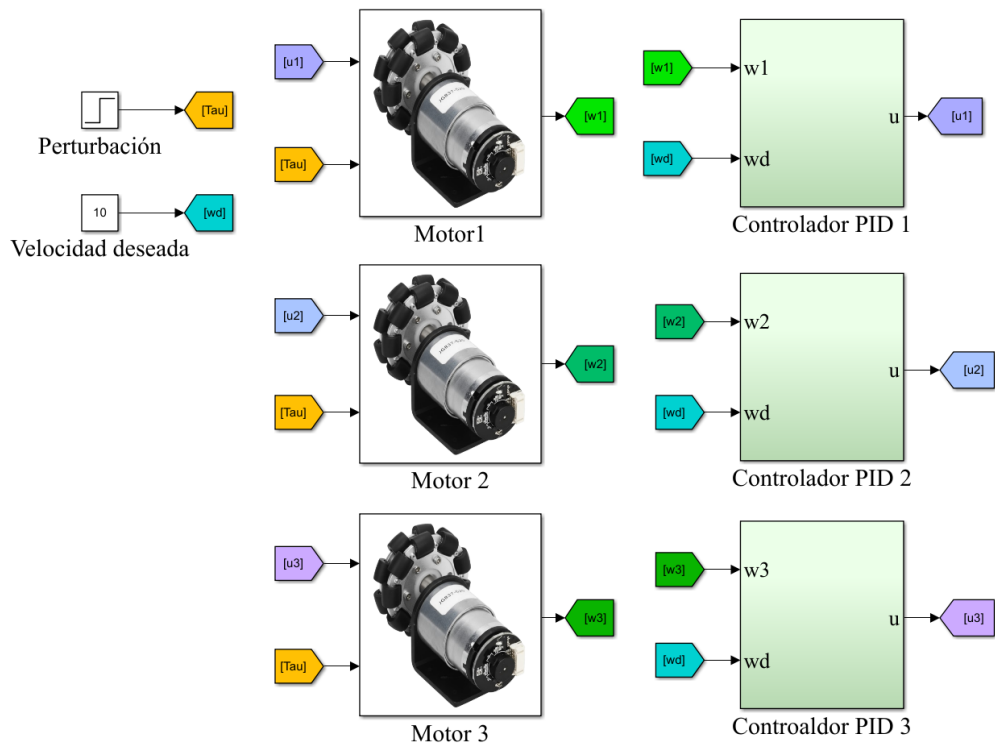


Figura 5.5: Diagrama de bloques del controlador PID.

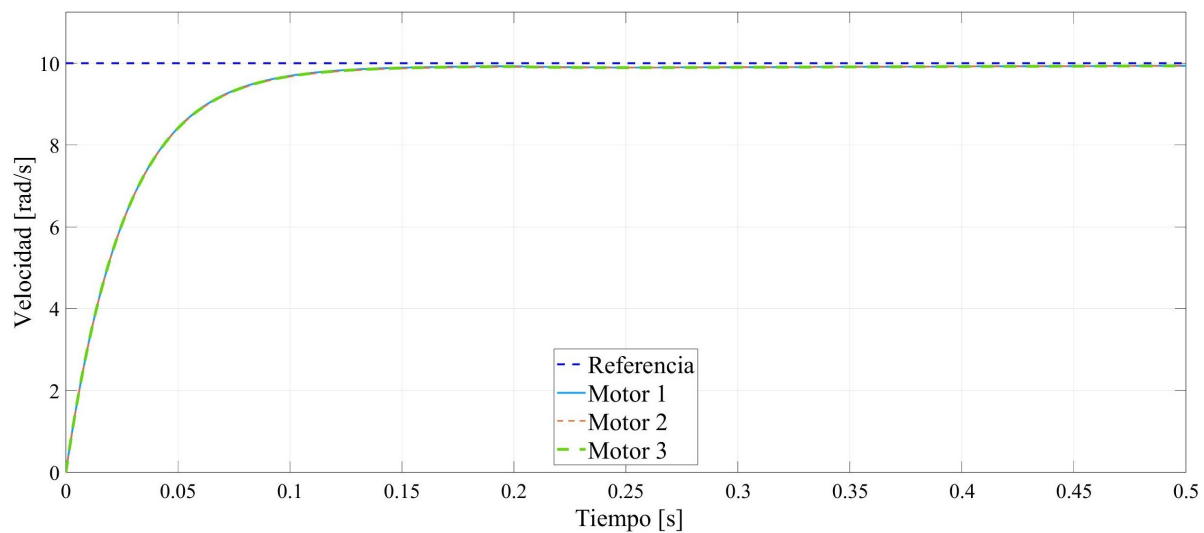


Figura 5.6: Velocidades vs referencia con controlador PID.

5. Diseño de controladores

Como se muestra en la Figura 5.6, el controlador PID logró acercarse al sistema a la referencia, aunque no consiguió alcanzarla completamente. Se observa un error persistente de aproximadamente 0.06 rad/s, con un tiempo de asentamiento cercano a 0.15 segundos. Es importante destacar que el sistema no presenta sobreimpulso durante su respuesta. Además, se aplica la perturbación antes descrita en la sección 5.2 de forma súbita en el instante 0.2 segundos, la cual desaparece en el instante 0.35 s.

De igual manera, el desempeño del controlador se evaluó mediante la gráfica del ISE, que reflejó la acumulación del error entre la velocidad de referencia y la velocidad medida. En la Figura se muestra el ISE correspondiente al motor 1, ya que, al tratarse de controladores idénticos para los tres motores, fue suficiente analizar el comportamiento de uno de ellos.

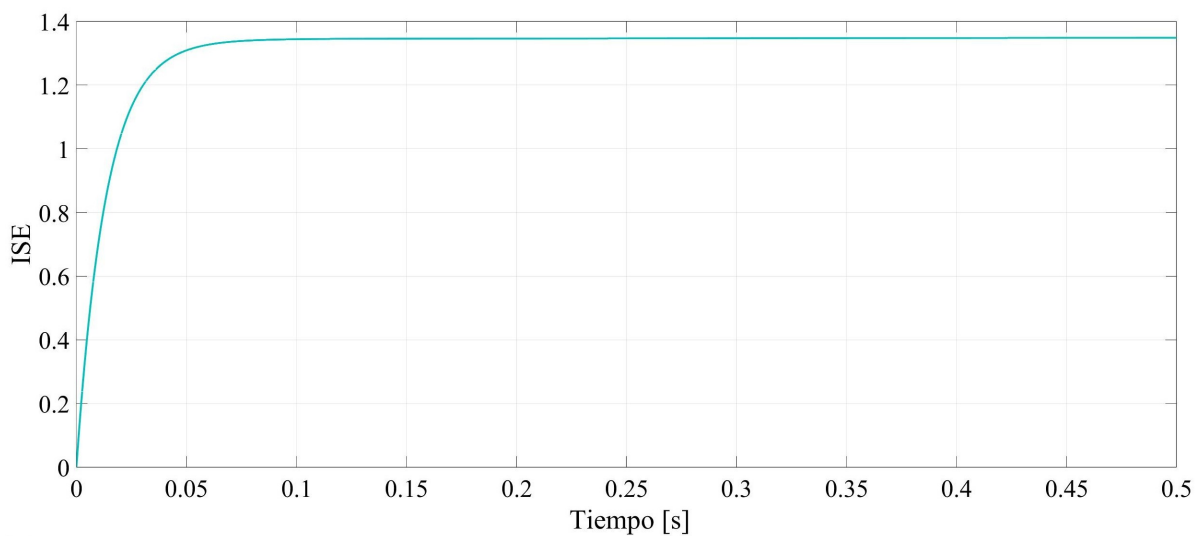


Figura 5.7: ISE del Motor 1 con el controlador PID.

5.4. Comparación de resultados de la simulación

Para comparar el desempeño de ambos controladores, se analizó la gráfica del ISE en cada caso. Los resultados muestran que el controlador ADRC generó un ISE significativamente menor en comparación con el controlador PID, lo que indicó una mejor capacidad para minimizar el error entre la velocidad de referencia y la velocidad medida, lo que se traduce en un seguimiento más preciso de la referencia.

Esta comparación resaltó la efectividad del controlador ADRC en condiciones similares, confirmando su superioridad en términos de precisión y estabilidad en el entorno de simulación.

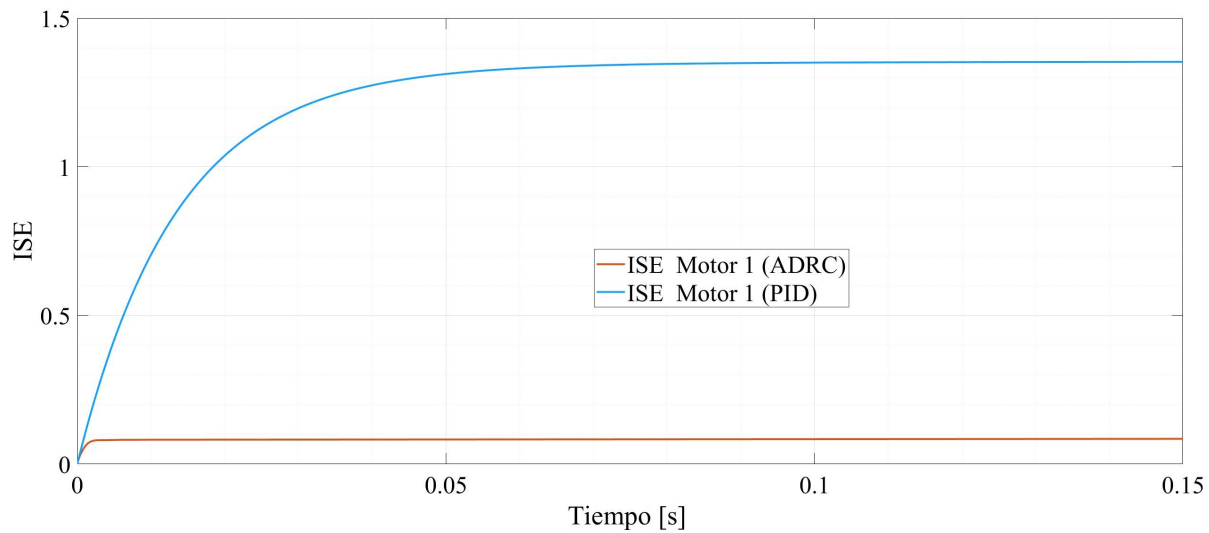


Figura 5.8: Comparación del ISE de ambos controladores.

Implementación del control ADRC y el observador GPI en el robot

Hasta este punto, se presentaron el diseño y la simulación del control cinemático del robot, así como su implementación práctica utilizando un controlador PID para la regulación de la velocidad de los motores. Sin embargo, con el propósito de incrementar la robustez y el desempeño del sistema frente a perturbaciones e incertidumbres del modelo, se desarrolló un controlador ADRC complementado con un observador GPI. Este esquema sustituye al controlador PID en el microcontrolador, con el objetivo de lograr una regulación más eficiente y precisa de la velocidad de los motores.

En este capítulo se describe la implementación del controlador ADRC en el sistema ROCKY, abordando tanto su integración en el microcontrolador como los resultados obtenidos en pruebas de control cinemático de regulación de posición y seguimiento de trayectorias. Finalmente, se analiza el comportamiento del sistema bajo este nuevo esquema y se compara su desempeño con el obtenido previamente mediante el controlador PID.

6.1. Discretización e implementación en el microcontrolador

Para implementar el controlador ADRC diseñado en la Sección 5.1, fue necesario llevar a cabo su discretización, ya que el microcontrolador opera en un dominio de tiempo discreto. Tanto el controlador ADRC como el observador GPI fueron formulados inicialmente en forma continua, por lo que fue indispensable adaptar sus ecuaciones mediante un método de integración adecuado, en este caso el método de Euler hacia adelante, con un tiempo de muestreo fijo.

El proceso de discretización consideró las condiciones operativas del microcontrolador, incluyendo la frecuencia de muestreo, la capacidad de cómputo y la necesidad de mantener una respuesta en tiempo real. Durante esta etapa, fue necesario ajustar las ganancias originales definidas en el diseño continuo, con el fin de evitar la saturación de la señal de control durante la ejecución del código en el microcontrolador. Estos ajustes permitieron mantener la estabilidad del sistema y asegurar un desempeño adecuado bajo las limitaciones propias del entorno embebido.

Una vez discretizado el esquema de control, se procedió a su codificación en el entorno del microcontrolador. El Algoritmo 3 presenta el pseudocódigo que describe la lógica de implementación del controlador ADRC junto con el observador GPI para la regulación de velocidad de los motores.

6. Implementación del control ADRC y el observador GPI en el robot

Algoritmo 3 Control ADRC con Observador GPI en microcontrolador ESP32

```
función INICIALIZACIÓN(Puerto)
    Comunicación ← EstablecerComunicación(Puerto)
3:   ConfigurarEncoders()
    ConfigurarMotores()
    InicializarVariablesEstado()
6: fin función
función BUCLEDECONTROL
    mientras verdadero hacer
9:     Referencias ← RecibirVelocidades()
        Velocidades ← LeerEncoders()
        Velocidades filtradas ← FiltroPasabajas(Velocidades)
12:    Derivadas ← CalcularDerivadas(Referencias)
        Derivadas filtradas ← FiltroPasabajas(Derivadas)
        EstadoEstimado ← ObservadorGPI(Velocidades filtradas)
15:    SeñalControl ← ControlADRC(Referencias, Derivadas filtradas, EstadoEstimado)
        AjustarMotores(SeñalControl)
        EnviarDatos(Velocidades)
18:    fin mientras
fin función
función OBSERVADORGPI(Velocidades)
21:     $e$  ← ErrorObservador(Velocidades, EstadoEstimado)
        ActualizarEstadosGPI( $e$ )
        devolver EstadoEstimado
24: fin función
función CONTROLADRC(Referencias, Derivadas, EstadoEstimado)
     $(y_1, y_2, \phi)$  ← EstadoEstimado
27:     $u$  ← LeyDeControlADRC(Referencias, Derivadas,  $y_1, y_2, \phi$ )
        devolver Saturar( $u$ )
fin función
```

6.1.1. Filtro pasa bajas de primer orden

Con el fin de reducir el ruido presente en las señales medidas y mejorar la estabilidad del controlador, se implementó un filtro pasa bajas de primer orden aplicado tanto a las velocidades obtenidas por los encoders como a las derivadas de las señales de referencia.

La ecuación discreta utilizada corresponde a una actualización recursiva del tipo:

$$y_{\text{filtrado}}(k) = y_{\text{filtrado}}(k - 1) + \alpha_{LP} \left(y_{\text{medido}}(k) - y_{\text{filtrado}}(k - 1) \right), \quad (6.1)$$

donde $y_{\text{medido}}(k)$ es la señal original, $y_{\text{filtrado}}(k)$ es la señal resultante y α_{LP} es el coeficiente de filtrado que determina el ancho de banda del filtro. Este esquema, equivalente a un filtro exponencial de primer orden, presenta la ventaja de su baja complejidad computacional, lo que permite ejecutarlo en tiempo real dentro del microcontrolador ESP32 sin comprometer los tiempos de muestreo del lazo de control.

El valor de α_{LP} se calculó a partir de la constante de tiempo τ del filtro y del periodo de muestreo T_s , de acuerdo con la expresión:

$$\alpha_{LP} = \frac{T_s}{\tau + T_s}. \quad (6.2)$$

Para esta implementación se empleó un periodo de muestreo $T_s = 1$ ms y una constante de tiempo $\tau = 60$ ms, obteniéndose así un coeficiente α_{LP} adecuado para atenuar el ruido sin introducir un retardo significativo en la señal filtrada.

6. Implementación del control ADRC y el observador GPI en el robot

Selección del parámetro α_{LP}

El parámetro α_{LP} se ajustó de forma empírica con el fin de alcanzar un compromiso adecuado entre suavizado y retardo. Valores pequeños de α_{LP} proporcionan una mayor atenuación del ruido, aunque introducen un retardo más pronunciado en la señal filtrada. En contraste, valores grandes reducen el retardo pero permiten el paso de mayor contenido en alta frecuencia.

Para la implementación en el microcontrolador ESP32 se seleccionó un valor intermedio que permitió atenuar las oscilaciones rápidas no deseadas en las mediciones de velocidad y en las derivadas de las referencias, sin afectar de manera significativa la dinámica del control.

6.2. Resultados de la implementación

Una vez discretizado, el esquema de control se integró con el control cinemático previamente validado en la Sección 4.6. Para la implementación experimental se consideraron los valores de las ganancias correspondientes al control ADRC y al observador GPI, los cuales se presentan a continuación.

$$\zeta_{control} = 0.9, \quad \omega_{n_{control}} = 15.0 \text{ rad/s}, \quad \zeta_{observador} = 1.0, \quad \omega_{n_{observador}} = 20.0 \text{ rad/s}$$

De manera análoga a las pruebas presentadas en la Sección 4.6, se inició la validación experimental del ADRC con un escenario de regulación de posición. En este caso, el objetivo fue llevar al robot hasta la posición $[1.5, 1]$ m con una orientación final deseada de 45° . Los resultados obtenidos para esta configuración se muestran a continuación.

En la Figura 6.1 se presenta la trayectoria realizada por el robot desde su posición inicial

hasta el objetivo establecido. La curva azul corresponde al recorrido ejecutado, comenzando en el marcador verde (posición inicial) y finalizando en el marcador rojo (posición deseada). Se observa que el robot alcanza correctamente la posición deseada, describiendo un movimiento continuo, sin sobrepasos ni desviaciones apreciables. El tiempo total empleado para la alcanzar la posición deseada fue de aproximadamente 16 segundos.

Para esta prueba se emplearon las ganancias $k_p = 0.18$ y $k_i = 0.25$ en las componentes u_x , u_y y u_φ . Dichos parámetros fueron ajustados de forma empírica durante las pruebas de validación del control cinemático de regulación de posición (Sección 4.6).

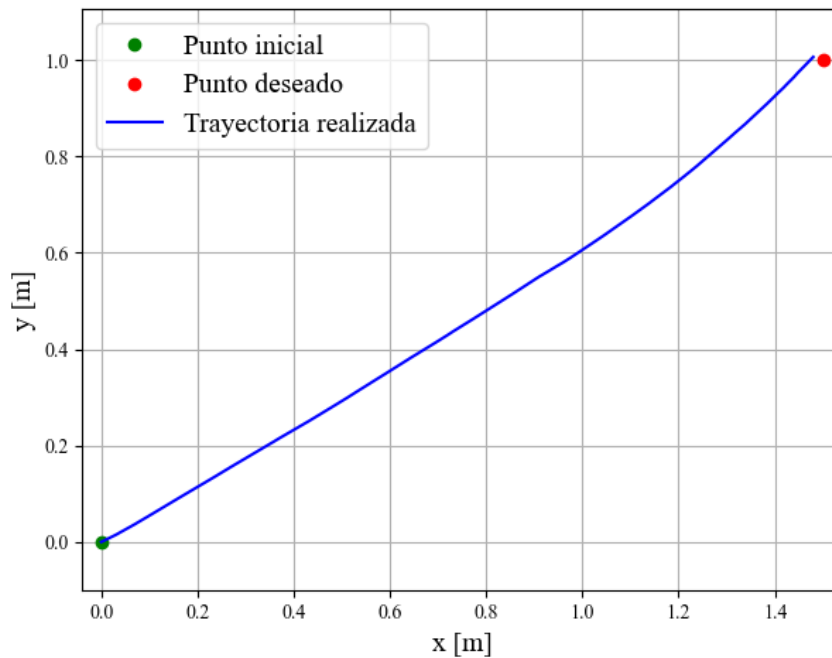


Figura 6.1: Trayectoria realizada por el robot.

De manera similar a la evaluación realizada con el controlador PID, se analizó también el comportamiento de las velocidades de referencia frente a las velocidades medidas bajo el esquema ADRC. En la Figura 6.2 se ilustran las componentes frontal y lateral de la velocidad de referencia generadas por el controlador cinemático, junto con las señales medidas a partir de los encoders

6. Implementación del control ADRC y el observador GPI en el robot

acoplados a los motores. Se observa que las velocidades medidas reprodujeron adecuadamente la tendencia de las referencias, aunque con un desfase en amplitud. Este desfase no comprometió de manera significativa el resultado global, ya que el robot mantuvo la estabilidad y logró alcanzar la posición objetivo.

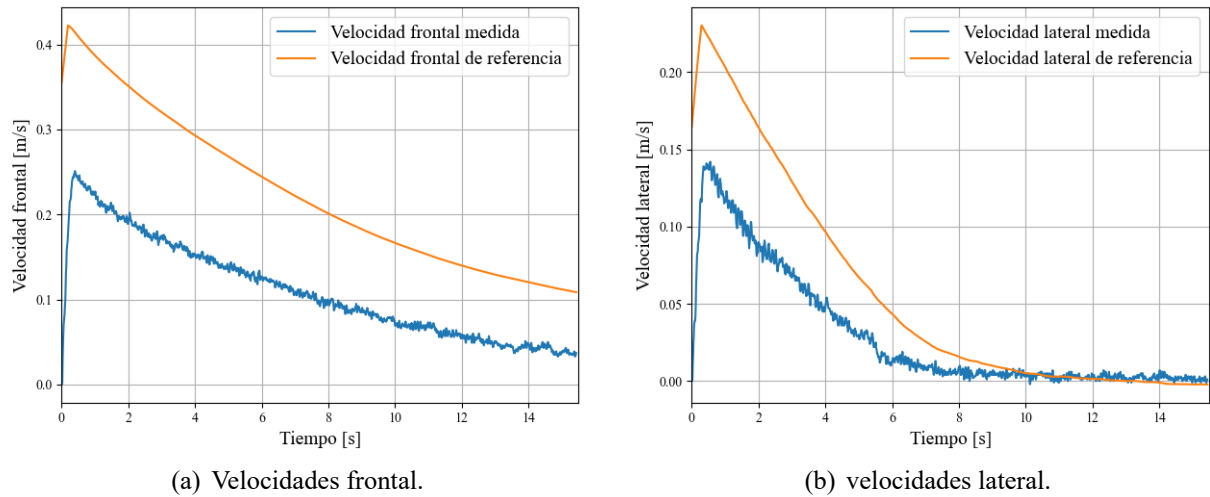


Figura 6.2: Comparativa de las velocidades frontal y lateral.

En la Figura 6.3 se presenta la evolución de la posición en los ejes x e y a lo largo del tiempo, mientras que la Figura 6.4 muestra el comportamiento de la orientación durante el desplazamiento del robot. La primera gráfica evidencia que el robot alcanzó el objetivo sin sobrepasarlo, manteniendo un desplazamiento continuo en ambos ejes. Este resultado resalta la capacidad del controlador ADRC para evitar sobreimpulsos en la señal de control y garantizar una respuesta suave y estable durante todo el proceso de regulación, lo que confirmó su eficacia en términos de estabilidad y precisión posicional.

Por su parte, la gráfica de orientación reflejó que durante el recorrido el robot ajustó de manera continua su ángulo para alinearse con la referencia. Al finalizar el desplazamiento, en la posición objetivo, la orientación registrada es de aproximadamente 43.5° , valor muy cercano al esperado, lo

que demuestra una elevada precisión en el control de la orientación.

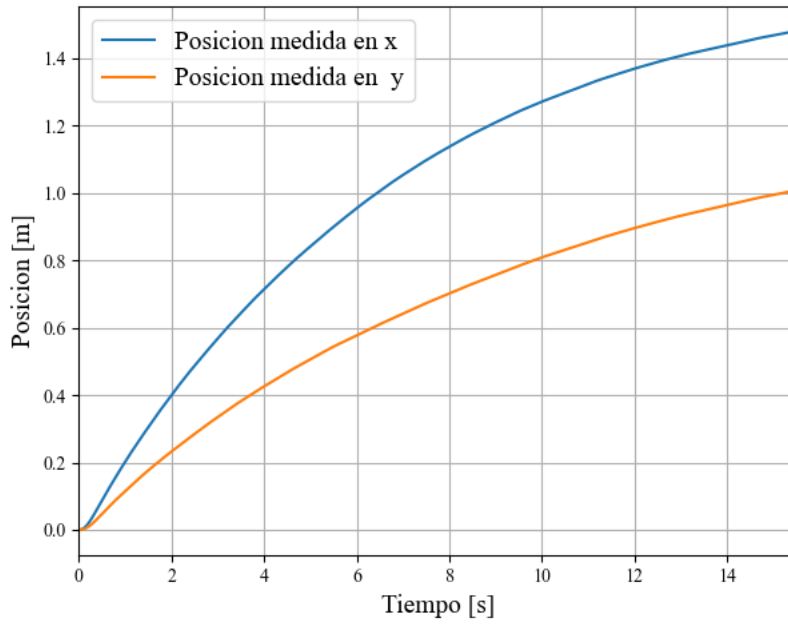


Figura 6.3: Posición en x e y durante el desplazamiento.

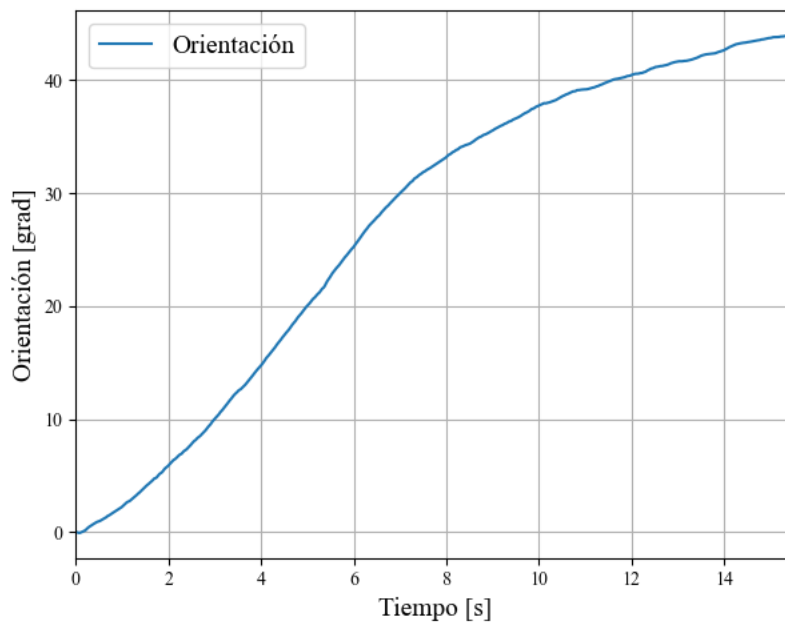
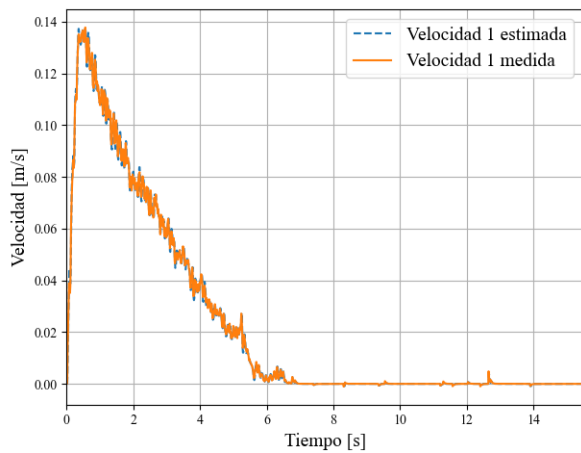


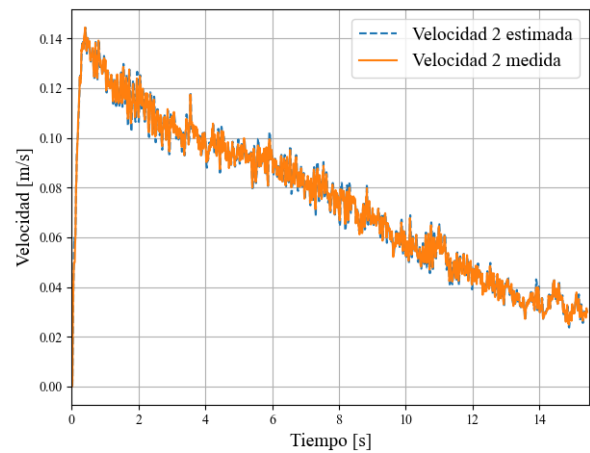
Figura 6.4: Orientación del robot durante el desplazamiento.

6. Implementación del control ADRC y el observador GPI en el robot

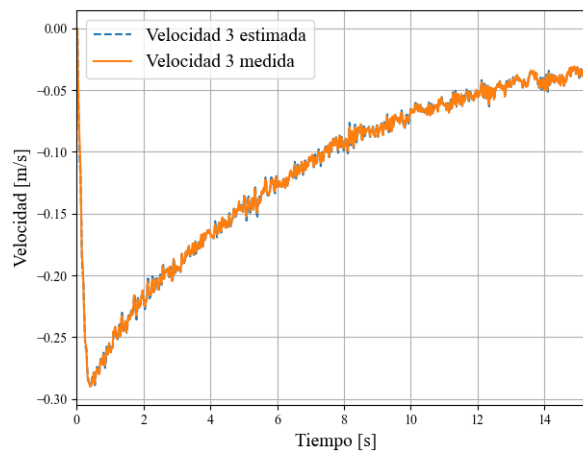
Dado que en este esquema de control se empleó un observador GPI para estimar las velocidades de cada motor, resultó fundamental evaluar su desempeño. En las Figuras 6.5 se presenta la comparación entre las velocidades medidas mediante los encoders y las estimadas por el observador durante la prueba. Se aprecia una correspondencia estrecha entre ambas señales, lo que confirmó que el observador es capaz de reproducir con alta fidelidad la dinámica de los motores. Este resultado validó su funcionamiento y garantizó que las estimaciones obtenidas pudieran emplearse de manera confiable dentro del esquema ADRC.



(a) Comparativa motor 1.



(b) Comparativa motor 2.



(c) Comparativa motor 3.

Figura 6.5: Comparativa de las velocidades medidas y estimadas con el GPI.

Para cuantificar la eficacia del control cinemático de posición utilizando el ADRC como regulador de velocidad de los motores, se replicó la metodología experimental detallada en la Sección 4.6.

Los parámetros de las pruebas, incluyendo las posiciones/orientaciones deseadas, el número de repeticiones y el criterio de éxito (≤ 0.02 m de error en posición), se mantuvieron idénticos a los utilizados en la evaluación con el controlador PID.

Los resultados de esta serie de pruebas se encuentran resumidos en la Tabla 6.1. Al igual que en el análisis previo, esta tabla presenta las métricas obtenidas únicamente de las repeticiones consideradas como exitosas.

Tabla 6.1: Desempeño del controlador ADRC en control cinemático de posición.

Métrica	(1.0, 1.0, 30.0°)	(1.0, -1.0, -30.0°)	(-1.0, 1.0, -30.0°)	(-1.0, -1.0, 30.0°)
Éxitos [%]	100.0	100.0	90.0	100.0
\bar{t}_{alcance} [s]	17.191	16.346	17.017	16.289
RMSE_x [m]	0.436	0.441	0.435	0.442
RMSE_y [m]	0.436	0.435	0.424	0.442
RMSE ϕ [grad]	12.79	11.36	11.71	10.23
$e_{\text{final},x}$ [m]	0.005	0.028	-0.027	-0.021
$e_{\text{final},y}$ [m]	0.028	-0.023	-0.001	-0.029
$e_{\text{final},\phi}$ [grad]	-4.29	6.2	-2.13	-0.99

Nota: \bar{t}_{alcance} : Tiempo promedio de alcance. RMSE: Raíz Cuadrada del Error Cuadrático Medio de la trayectoria. e_{final} : Error final promedio.

Las pruebas realizadas demostraron que el controlador ADRC alcanzó tasas de éxito muy altas: 100 % en tres de cuatro escenarios, y únicamente una prueba fallida en el escenario (-1.0,1.0,-30.0°). El tiempo de alcance promedio rondó entre 16 y 17 segundos. Los RMSE_x y RMSE_y reportaron valores en torno a 0.43 m, mientras que los errores finales fueron más pequeños (± 2.8 cm) y el error angular final quedó dentro de $\pm 6.2^\circ$, lo que indicó buen asentamiento una vez alcanzada la vecindad del objetivo. Los RMSE relativamente altos estuvieron dominados por el tramo transitorio (desde distancias iniciales del orden de 1 m), más que por el régimen permanente. Los RMSE se

6. Implementación del control ADRC y el observador GPI en el robot

calcularon sobre todo el desplazamiento; por ello reflejan el proceso de aproximación además del estado estacionario.

La validación experimental del control cinemático de seguimiento bajo el esquema ADRC se llevó a cabo mediante una prueba con ROCKY. En este caso se utilizó una trayectoria parametrizada en función del tiempo, de forma similar a la empleada en la Sección 4.5.2. Para obtener un desplazamiento suave, se definió una amplitud de $a = 0.25$ m en el eje x y de $a = 0.1$ m en el eje y . A diferencia de la prueba con PID, aquí se aplicó un factor de escala distinto en el tiempo, con el propósito de adecuar la dinámica de la trayectoria a las condiciones de validación del ADRC. De este modo, la referencia quedó establecida según la expresión (6.3).

$$\begin{aligned}x_d(t) &= 0.25 \cdot \cos(0.12 \cdot t) \\y_d(t) &= 0.1 \cdot \sin(0.24 \cdot t)\end{aligned}\tag{6.3}$$

En la Figura 6.6 se muestra la comparación entre la trayectoria de referencia y la ejecutada por el robot bajo el esquema de control ADRC. La referencia corresponde a una *Lemniscata de Bernoulli*, mientras que el recorrido real reprodujo con buena precisión su forma, presentando desviaciones menores en comparación con lo observado con el controlador PID, particularmente en las zonas con mayor curvatura. A diferencia del comportamiento anterior, donde se observaban picos u oscilaciones más marcadas, en este caso el seguimiento se aproximó a un desplazamiento más lineal, semejante a un offset constante respecto a la trayectoria ideal. El error de seguimiento se mantuvo en torno a 4 cm, lo que confirmó un desempeño más estable y preciso frente al esquema anterior.

Para esta prueba se emplearon las ganancias cinemáticas $k_p = 0.45$ y $k_i = 0.5$ en los términos u_x , u_y y u_φ , ajustadas empíricamente a partir de ensayos experimentales. El robot inició su movimiento desde el origen $[0, 0]$ m y completó la trayectoria establecida de manera satisfactoria. Cabe destacar que en este caso la referencia avanzó a una velocidad mayor que en las pruebas con PID,

lo cual fue posible gracias a la robustez del ADRC. No obstante, durante la ejecución se presentaron limitaciones en la comunicación inalámbrica con el ordenador externo, posiblemente asociadas al mayor volumen de información transmitida en estas condiciones. Debido a esto, las pruebas se realizaron con el robot conectado de manera alámbrica al ordenador externo.

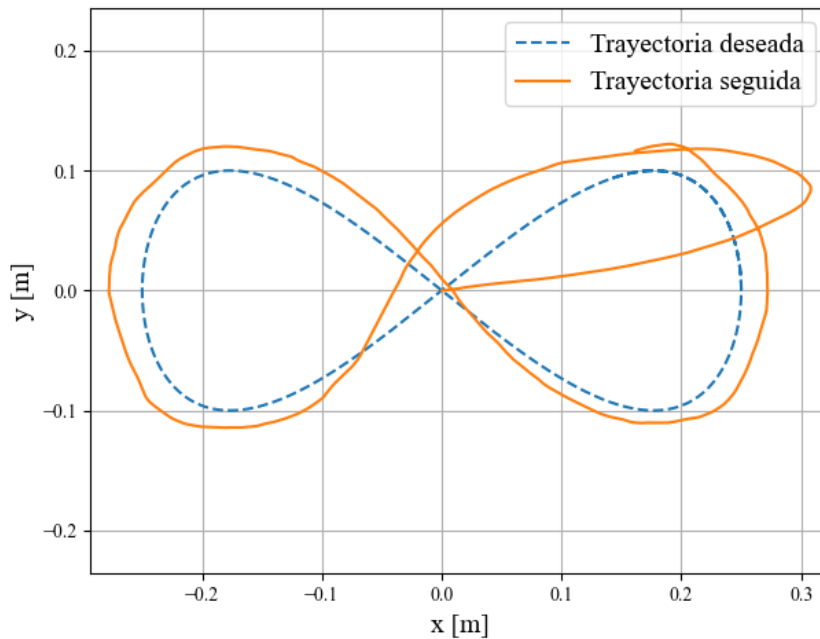


Figura 6.6: Comparativa de las trayectorias.

Asimismo, se analizaron las velocidades de referencia generadas por el control cinemático y las velocidades reales medidas durante la prueba. En la Figura 6.7 se ilustran las comparaciones de las componentes frontal y lateral. Se observa que las velocidades medidas presentaron un ligero desfase en amplitud respecto a las referencias, pero con una reducción notable del ruido gracias al filtro pasa bajas aplicado a las señales antes de su incorporación al controlador ADRC.

Además, la correspondencia en la forma general de las señales confirmó que el sistema es capaz de reproducir adecuadamente las variaciones dinámicas de la referencia. Este comportamiento contribuyó a un seguimiento de trayectoria más suave y estable, evitando las oscilaciones pronunciadas

6. Implementación del control ADRC y el observador GPI en el robot

que se observaron en las pruebas con PID. En conjunto, estos resultados evidencian la mayor robustez del ADRC frente a perturbaciones y mediciones ruidosas, así como la efectividad del observador en la estimación de estados confiables para el control.

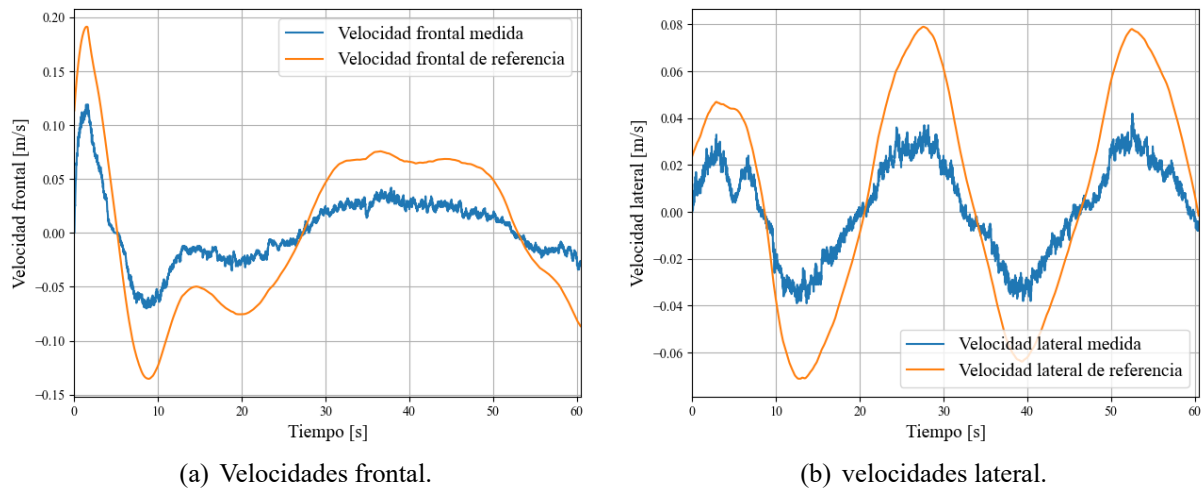


Figura 6.7: Comparativa de las velocidades frontal y lateral.

En la Figura 6.8 se comparan las posiciones deseadas y medidas en los ejes x e y a lo largo de la trayectoria. El robot reprodujo de manera cercana las referencias establecidas, manteniendo una evolución continua en ambos ejes y con discrepancias menores respecto al caso con PID. En particular, el seguimiento resultó más preciso y con menor nivel de ruido, lo que evidenció una mejora significativa en la calidad de la trayectoria.

Por otra parte, la Figura 6.9 muestra la evolución de la orientación del robot durante el desplazamiento. Aunque no se mantuvo constante en torno a la referencia de 0° , las variaciones se limitaron a valores de aproximadamente 6° en los picos más altos y a 4.5° en los cambios de dirección. Estas desviaciones fueron considerablemente menores a las registradas con el esquema PID, lo que confirmó la mayor estabilidad y precisión del ADRC en el control de la orientación.

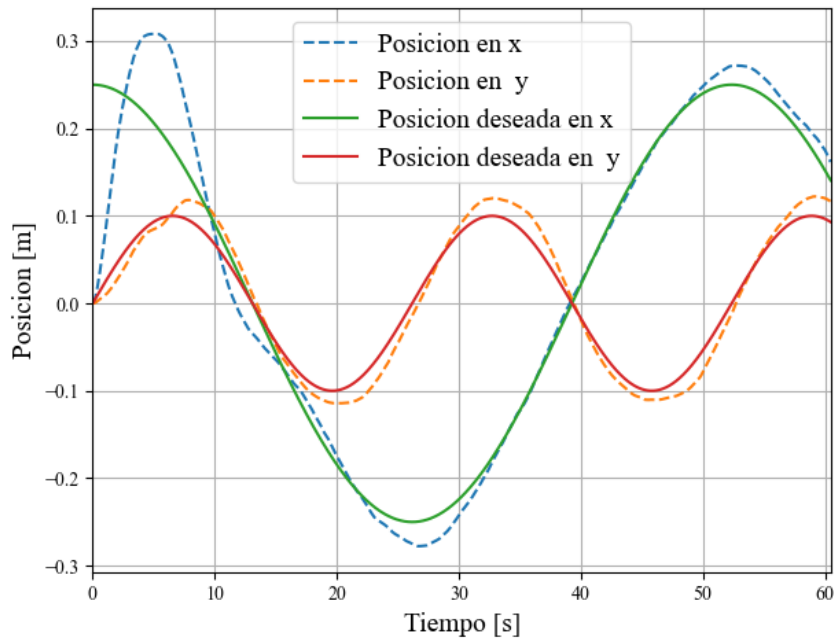


Figura 6.8: Posición del robot durante el seguimiento

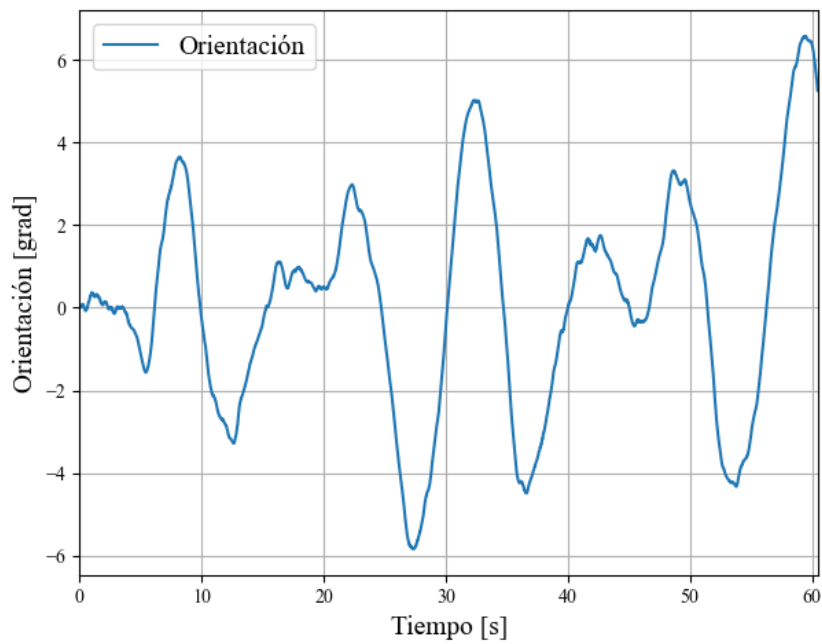


Figura 6.9: Orientación del robot durante el seguimiento.

6. Implementación del control ADRC y el observador GPI en el robot

El análisis del desempeño del controlador se complementó con la evaluación del error de posición. En la Figura 6.10 puede verse que la componente en el eje x exhibió las variaciones más notorias al inicio del recorrido, lo cual obedece a la mayor separación existente entre la posición de partida del robot y la trayectoria de referencia en dicho eje. Con el avance de la prueba, estas variaciones se atenuaron progresivamente y solo volvieron a manifestarse de forma puntual en los segmentos donde la trayectoria exigió un cambio de dirección. Aun en estos casos, el error se mantuvo confinado dentro de un margen cercano a ± 4 cm.

A partir de las componentes en x e y , se calculó el error euclidiano de posición, mostrado en la Figura 6.11. Este indicador refleja de manera global la distancia entre la posición real y la referencia en cada instante. Los resultados evidencian que dicho error permaneció reducido y estable durante toda la trayectoria, lo cual confirmó la capacidad del esquema ADRC para sostener un seguimiento preciso y confiable bajo las condiciones experimentales establecidas.

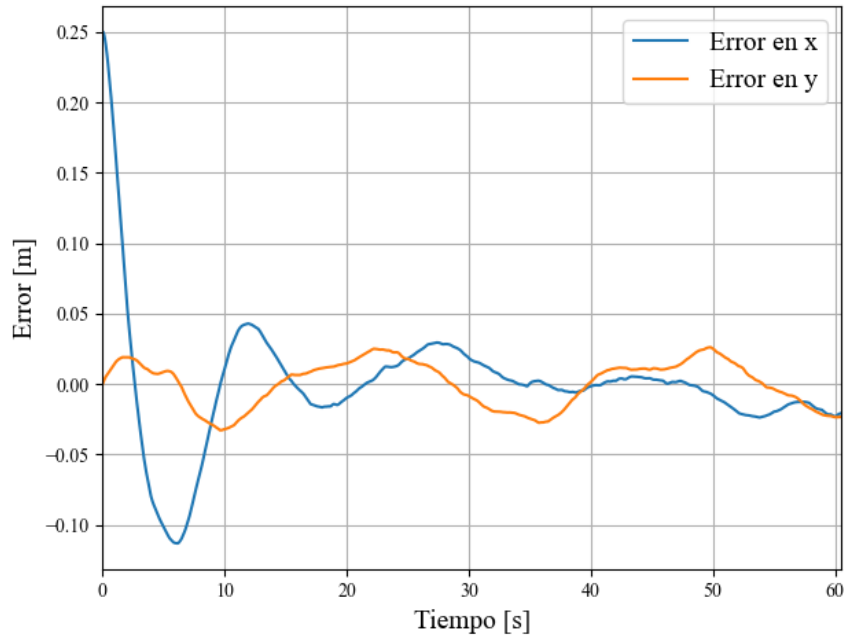


Figura 6.10: Error de posición a lo largo de la trayectoria.

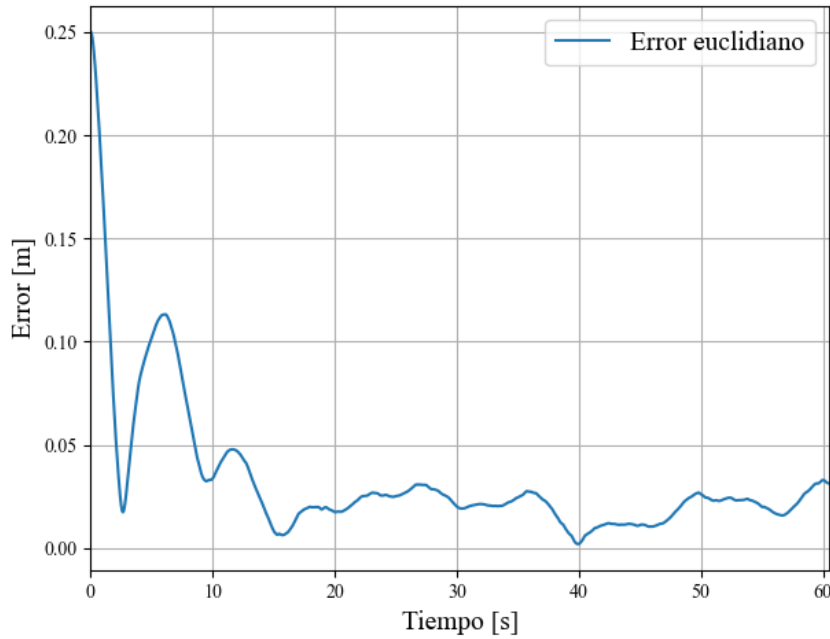


Figura 6.11: Error euclidiano de posición a lo largo de la trayectoria.

También se analizó el comportamiento del observador GPI encargado de estimar las velocidades de los motores. En las Figuras 6.12 se muestran las señales registradas por los encoders en contraste con las generadas por el observador durante la ejecución de la prueba. Las curvas presentaron una coincidencia consistente a lo largo del tiempo, lo que puso de manifiesto la capacidad del estimador para capturar adecuadamente la dinámica de los actuadores. Estos resultados demostraron que las estimaciones obtenidas son suficientemente precisas para ser utilizadas de forma segura dentro de la estructura del ADRC, asegurando así la confiabilidad del esquema de control.

6. Implementación del control ADRC y el observador GPI en el robot

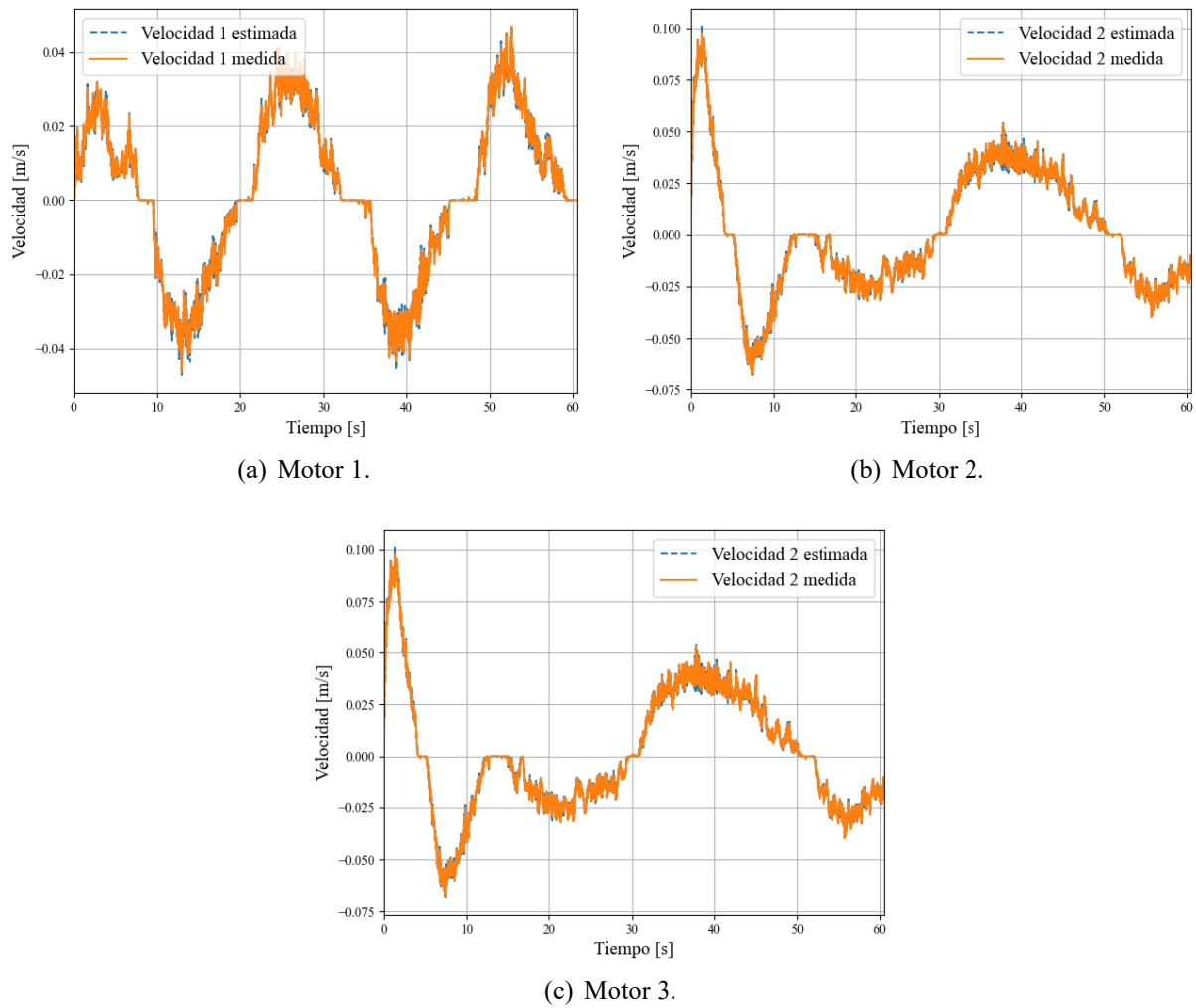


Figura 6.12: Comparativa de las velocidades medidas y estimadas con el GPIO.

Las gráficas anteriores corresponden a los resultados analizados en una sola prueba; sin embargo, para cuantificar la eficacia del control cinemático de seguimiento de trayectorias utilizando el ADRC como regulador de velocidad de los motores, se replicó la metodología experimental detallada en la Sección 4.6. Los parámetros de las pruebas, incluyendo las trayectorias deseadas, el número de repeticiones y demás criterios se mantuvieron idénticos a los utilizados en la evaluación con el controlador PID.

Los resultados de esta serie de pruebas se encuentran resumidos en la Tabla 6.2. Al igual que en el análisis previo, esta tabla presenta las métricas obtenidas de las diez repeticiones de cada trayectoria.

Tabla 6.2: Desempeño del controlador ADRC en seguimiento de trayectorias.

Métrica	Lemniscata	Círculo
RMSE_x [m]	0.01	0.016
RMSE_y [m]	0.015	0.019
RMSE_pos [m]	0.018	0.025
RMSE ϕ [grad]	2.13	2.54
P95($ e_x $) [m]	0.022	0.024
P95($ e_y $) [m]	0.028	0.035
Tiempo en tolerancia [%]	78.5	46.7
$\ u\ _{L2}$	0.699	0.952

Nota: $RMSE_{pos} = \sqrt{RMSE_x^2 + RMSE_y^2}$. P95 es el percentil 95 del error absoluto por eje. El esfuerzo $L2$ se calcula sobre los comandos normalizados y es comparable entre pruebas con la misma ventana temporal.

Los resultados mostraron que, en ambas trayectorias, el $RMSE_{pos}$ se mantuvo en el orden de los centímetros: 1.8 cm para la lemniscata y 2.5 cm para el círculo. La orientación también quedó contenida en un rango reducido ($\approx 2.1\text{--}2.5^\circ$), lo que indicó un seguimiento adecuado y una buena alineación del robot.

En cuanto a robustez, los percentiles $P95(|e_x|)$ y $P95(|e_y|)$ se situaron entre 2.2 cm y 3.5 cm; es decir, durante el 95 % del tiempo el error absoluto por eje no excede esos valores. La lemniscata presentó, en promedio, menor error y mayor tiempo en tolerancia que el círculo: 78.5 % del tiempo dentro de ± 2 cm en ambos ejes, frente a 46.7 % en el círculo. Esta diferencia sugirió un pequeño *offset* persistente en el círculo (véase la Figura 6.6), más que errores de gran magnitud. Finalmente, el esfuerzo $\|u\|_{L2}$ también es menor en la lemniscata, lo que apuntó a un seguimiento más suave con los parámetros actuales.

Para estas pruebas se consideraron ventanas de 53 segundos para la lemniscata y 32 segundos para el círculo. Con estos resultados, y en comparación con la Sección 4.6, puede concluirse que

6. Implementación del control ADRC y el observador GPI en el robot

la implementación del ADRC como regulador de velocidad por motor fue satisfactoria: permitió operar a mayor velocidad manteniendo errores acotados.

Discusión y conclusiones

7.1. Discusión

Los resultados finales indican que el controlador ADRC permite un seguimiento de trayectorias más rápido que el controlador PID sin penalizar la precisión. En la lemniscata, el tiempo por vuelta se reduce de 126 segundos (PID) a 53 segundos (ADRC), y en el círculo de 92 segundos a 32 segundos, manteniendo errores de posición del orden de 2–3 cm y errores angulares de pocos grados ($\approx 2.1\text{--}2.5^\circ$). En otras palabras, con una calidad de seguimiento comparable, el ADRC logra menor tiempo de ejecución de las trayectorias. Aunque el PID destacó por su simplicidad de implementación, el ADRC redujo el desfase y el ruido en las señales, alcanzando mayor fidelidad respecto a las referencias. No obstante, esta mejora implicó un mayor esfuerzo computacional y requirió el uso de comunicación cableada con la computadora externa. Este cambio aunque limitó la autonomía física, fue determinante para la viabilidad del control, pues el enlace físico eliminó la incertidumbre en la transmisión de datos (jitter) y las pérdidas de paquetes inherentes a la red inalámbrica, garantizando el determinismo temporal requerido para la correcta convergencia del observador de estados.

En términos comparativos, el PID sigue siendo atractivo en escenarios con recursos limitados,

mientras que el ADRC se perfila como una alternativa más precisa y robusta para entornos donde la exactitud del seguimiento es prioritaria. En simulación, el ADRC logró un error en estado estacionario menor que el PID, y en la implementación práctica mantuvo esta superioridad, obteniendo menores tiempos de respuesta en la ejecución de trayectorias completas.

Los resultados obtenidos son consistentes con los reportados en [16], donde se demuestra la viabilidad del ADRC para la tarea de seguimiento de trayectorias en robots omnidireccionales de tres ruedas. Sin embargo, a diferencia de dicho estudio, que empleó un sistema OptiTrack para estimar posición y orientación, en este trabajo se recurrió únicamente a la odometría basada en encoders. A pesar de utilizar hardware más económico y accesible, se obtuvieron resultados comparables en el seguimiento de trayectorias, lo que refuerza la pertinencia del enfoque propuesto.

Es importante señalar que las pruebas se realizaron en un entorno controlado y estático, lo cual podría diferir de condiciones reales más dinámicas. Asimismo, la caracterización del motor se basó en un modelo lineal que no captura la totalidad de sus dinámicas no lineales. Otro factor que influyó en los resultados fue la comunicación inalámbrica entre el microcontrolador y la computadora externa, que no pudo mantenerse durante las pruebas de trayectoria, limitando el alcance experimental. Estas limitaciones deben ser consideradas al extrapolar los hallazgos a escenarios más complejos.

Además del análisis de los controladores, el desarrollo del robot móvil constituye en sí mismo una contribución significativa. A diferencia de trabajos que se restringen al ámbito de la simulación, este proyecto avanzó hacia la implementación experimental, superando los retos propios de un sistema físico real. El diseño del robot valida la factibilidad práctica de los controladores y proporciona una plataforma replicable que puede aprovecharse en futuras investigaciones, tanto en esta universidad como en otras instituciones interesadas en el estudio de robots omnidireccionales.

En cuanto a la proyección futura, se recomienda la integración de una unidad de medición inercial (IMU) que complemente el control cinemático mediante técnicas de fusión sensorial, con el

7. Discusión y conclusiones

fin de mejorar la precisión en el seguimiento de trayectorias. Asimismo, la incorporación de una microcomputadora a bordo permitiría eliminar la dependencia de un ordenador externo y, con ello, los problemas de comunicación inalámbrica, dando lugar a un sistema más autónomo y robusto.

7.2. Conclusiones

Esta sección sintetiza los hallazgos más relevantes obtenidos durante el desarrollo de esta investigación, respondiendo a los objetivos planteados:

1. Validación del objetivo principal: Se cumplió exitosamente el objetivo principal de desarrollar un robot móvil omnidireccional de tres ruedas y de implementar técnicas avanzadas de control para la navegación. Esto incluyó el diseño del sistema físico, la caracterización de los actuadores, y la validación del controlador ADRC y los observadores de estado para la estimación precisa de la posición.
2. La técnica de Control Activo de Rechazo a Perturbaciones (ADRC) demostró ser superior a las técnicas tradicionales como el controlador PID para la tarea de regulación de la velocidad de los motores en el robot móvil omnidireccional. Esta superioridad se manifestó en una notable mejora en el seguimiento de trayectorias, minimizando el impacto de perturbaciones comunes como la vibración, la fricción y el deslizamiento de las ruedas.
3. Mejora en el desempeño dinámico: La robustez inherente al controlador ADRC permitió que el robot omnidireccional siguiera referencias de trayectoria a mayor velocidad, incrementando el desempeño dinámico del sistema en comparación con el control PID.
4. Aporte metodológico: Este trabajo sienta las bases de una metodología de implementación de técnicas de control ADRC en sistemas físicos, particularmente en la configuración de robot móvil omnidireccional de tres ruedas. Esta configuración es un área menos explorada

en comparación con los robots diferenciales o de Ackermann, lo que constituye un aporte significativo al campo de la robótica móvil.

7.3. Trabajo futuro

Los hallazgos de esta investigación tienen implicaciones directas en la mejora de la precisión y velocidad de los sistemas de navegación de robots móviles. El uso del ADRC permite diseñar plataformas más confiables para tareas que requieren alta exactitud en entornos dinámicos. A partir de estas conclusiones, se proponen las siguientes líneas de investigación futuras:

1. Integración sensorial avanzada: Continuar la investigación incorporando sensores complementarios, como Unidades de Medición Inercial (IMU), sistemas de visión por computadora o LiDAR, para robustecer la estimación de la posición, fusionando datos con los observadores de estado implementados.
2. Validación en entornos dinámicos: La validación del sistema de control ADRC en entornos con mayores incertidumbres y perturbaciones es una línea inmediata. Esto puede complementarse con la implementación de técnicas de aprendizaje automático para la sintonización adaptativa de los controladores.
3. Aplicaciones específicas: Proponer y validar aplicaciones reales y específicas para el robot omnidireccional, tales como el transporte de mercancías en almacenes o su rol como robot de servicio en entornos de interacción directa con humanos.

Referencias

- [1] I. Doroftei, V. Grosu, and V. Spinu, *Omnidirectional Mobile Robot - Design and Implementation*. IntechOpen, 09 2007.
- [2] M. Fuente, D. Cepeda, E. Garza, I. Saca, D. Garza, and F. Aleman, “Design and control of a four wheeled omnidirectional robot with image processing,” 10 2019, pp. 514–519.
- [3] S.-Y. Jiang and K.-T. Song, “Differential flatness-based motion control of a steer-and-drive omnidirectional mobile robot,” in *2013 IEEE International Conference on Mechatronics and Automation*, 2013, pp. 1167–1172.
- [4] S. Mishra, M. Sharma, and S. Mohan, “Behavioural fault tolerant control of an omnidirectional mobile robot with four mecanum wheels,” *Defence Science Journal*, 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:202100520>
- [5] R. Nanda, K. Karyadi, A. Suhara, T. Tukino, and M. Guntur, “Implementation of a pid controller in a dc motor system: a case study on a floor cleaning robot,” *Jurnal Polimesin*, vol. 23, p. 24, 02 2025.
- [6] L. Mérida-Calvo, A. San-Millan, F. Ramos, and V. Feliu, “Advanced motor control for improving the trajectory tracking accuracy of a low-cost mobile robot,” *Machines*, vol. 11, p. 14, 12 2022.
- [7] Z. T. Romadon, H. Oktavianto, I. K. Wibowo, B. Sena Bayu Dewantara, E. A. Nurrohmah, and R. Adryantoro Priambudi, “Pose estimation on soccer robot using data fusion from encoders, inertial sensor, and image data,” in *2019 International Electronics Symposium (IES)*, 2019, pp. 454–459.
- [8] H.-C. Huang and C.-C. Tsai, “Adaptive trajectory tracking and stabilization for omnidirectional mobile robot with dynamic effect and uncertainties,” *IFAC Proceedings Volumes*, vol. 17, 07 2008.

-
- [9] B.-S. Choi, J.-W. Lee, J.-J. Lee, and K.-T. Park, “A hierarchical algorithm for indoor mobile robot localization using rfid sensor fusion,” *IEEE Transactions on Industrial Electronics*, vol. 58, no. 6, pp. 2226–2235, 2011.
- [10] J. Santos, A. G. Conceição, and T. L. Santos, “Trajectory tracking of omni-directional mobile robots via predictive control plus a filtered smith predictor,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10 250–10 255, 2017, 20th IFAC World Congress. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896317318013>
- [11] M. W. Mehrez, K. Worthmann, J. P. Cenerini, M. Osman, W. W. Melek, and S. Jeon, “Model predictive control without terminal constraints or costs for holonomic mobile robots,” *Robotics and Autonomous Systems*, p. 103468, 02 2020.
- [12] H. Rodríguez-Cortés, M. Velasco-Villa, and E. Aranda-Bricaire, “Observer based trajectory-tracking control of an omnidirectional mobile robot,” in *Congreso Anual de la Asociación de México de Control Automático*. Asociación de México de Control Automático, 2007.
- [13] M. L. Garrido, “Sistema de localización y mapeo simultáneo para interiores basado en imágenes de profundidad (rgb-d),” Master’s thesis, Benemérita Universidad Autónoma de Puebla, jan 2021.
- [14] G. Szayer, B. Kovács, F. Tajti, and P. Korondi, “Feasible utilization of the inherent characteristics of holonomic mobile robots,” *Robotics and Autonomous Systems*, vol. 94, pp. 12–24, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889016306327>
- [15] M. A. Mahmud, M. S. Aman, H. Jiang, A. Abdelgawad, and K. Yelamarthi, “Kalman filter based indoor mobile robot navigation,” in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, 2016, pp. 1949–1953.
- [16] L. Leal-Ramos, L. Jonguitud-Indalecio, M. Ortiz-Michimani, J. Diaz-Téllez, J. Sánchez-Santana, and J. F. Guerrero-Castellanos, “Plataforma experimental para el modelado y control de robots omnidireccionales,” *Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI*, vol. 12, pp. 93–99, 04 2024.
- [17] E. Cortez-Dena, H. Contreras-Mendoza, M. Mastranzo-Pérez, E. Ranírez-Muñoz, and J. F. Guerrero-Castellanos, “Control de velocidad para servomecanismos libre de modelo,” *Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI*, vol. 10, pp. 91–96, 11 2022.
- [18] W. Jianbin and C. Jianping, “An adaptive sliding mode controller for four-wheeled omnidirectional mobile robot with input constraints,” in *2019 Chinese Control And Decision Conference (CCDC)*, 2019, pp. 5591–5596.

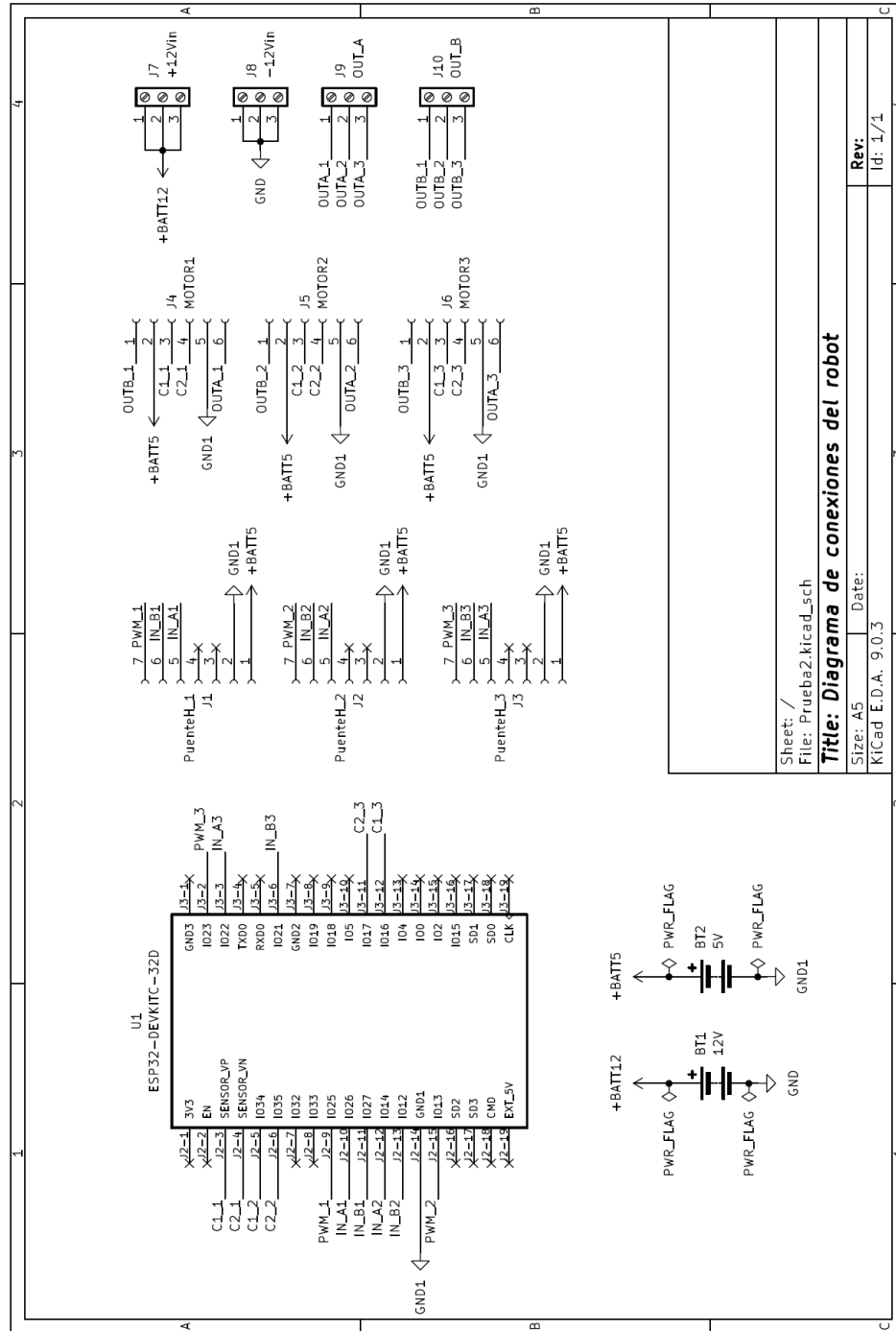
REFERENCIAS

- [19] V. Barbulescu, I. Marica, V. Gheorghe, M. Nistor, and M. Patrascu, “Encoder-based path tracking with adaptive cascaded control for a three omni-wheel robot,” in *2017 16th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, 2017, pp. 1–6.
- [20] J. Gausemeier and S. Moehring, “Vdi 2206- a new guideline for the design of mechatronic systems,” *IFAC Proceedings Volumes*, vol. 35, no. 2, pp. 785–790, 2002, 2nd IFAC Conference on Mechatronic Systems, Berkeley, CA, USA, 9-11 December. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474667017340351>
- [21] I. Maza and A. Ollero, *Robótica. Manipuladores y Robots Móviles*. Marcombo, 01 2001.
- [22] V. Barrientos, J. Sánchez, and R. Ortigoza, “Robots móviles: Evolución y estado del arte,” *Polibits*, vol. 35, pp. 12–17, 01 2007.
- [23] J. Figueroa-Olmedo, W. M. Montalvo-López, and M. M. Bayas-Sampedro, *Cinemática y dinámica de robots móviles con ruedas*, 1st ed. CILADI S.A.S., apr 2023, vol. 1.
- [24] K. Tadakuma, R. Tadakuma, and J. Berengeres, “Development of holonomic omnidirectional vehicle with “omni-ball”: spherical wheels,” 12 2007, pp. 33 – 39.
- [25] B.-Z. Guo and Z. Zhao, *Active Disturbance Rejection Control for Nonlinear Systems: An Introduction*. Wiley, 10 2016.
- [26] B. Martínez, J. Sanchis, S. García-Nieto, and M. Martínez, “Control por rechazo activo de perturbaciones: guía de diseño y aplicación,” *Revista Iberoamericana de Automática e Informática industrial*, vol. 18, no. 3, p. 201–217, jul. 2021. [Online]. Available: <https://polipapers.upv.es/index.php/RIAI/article/view/14058>
- [27] H. Sira-Ramirez, A. Luviano-Juárez, M. Ramirez-Neria, and E. Zurita-Bustamante, *Active Disturbance Rejection Control of Dynamic Systems: A flatness based approach*, 10 2018.
- [28] A. Nakhmani, *Modern Control State-Space Analysis and Design Methods*. Mc Graw Hill, 2020.
- [29] D. L. Martinez-Vazquez, A. Rodriguez-Angeles, and H. Sira-Ramirez, “Robust gpi observer under noisy measurements,” in *2009 6th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, 2009, pp. 1–5.
- [30] E. Systems, “Esp32wroom32 datasheet,” online, 2023.
- [31] STMicroelectronics, “Vnh2sp30-e automotive fully integrated h-bridge motor driver,” online, 2013.
- [32] C. P. Ordóñez, “Estudio de baterías para vehículos eléctricos,” Master’s thesis, Universidad Carlos III de Madrid, may 2011.

-
- [33] C. Robotics, “Coppeliassim: Coppelia robotics,” <http://www.coppeliarobotics.com>, 2023, accessed: 2024-06-11.
- [34] H. Sira-Ramirez, A. Luviano-Juárez, and J. Cortés-Romero, “Control lineal robusto de sistemas no lineales diferencialmente planos,” *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 8, no. 1, pp. 14–28, 2011. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1697791211700048>
- [35] L. Guo and S. Cao, “Anti-disturbance control theory for systems with multiple disturbances: A survey,” *ISA Transactions*, vol. 53, no. 4, pp. 846–849, 2014, disturbance Estimation and Mitigation. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0019057813001717>
- [36] S. Li, J. Yang, W.-H. Chen, and X. Chen, “Generalized extended state observer based control for systems with mismatched uncertainties,” *IEEE Transactions on Industrial Electronics*, vol. 59, no. 12, pp. 4792–4802, 2012.
- [37] H. Sira-Ramirez and M. A. Oliver-Salazar, “On the robust control of buck-converter dc-motor combinations,” *IEEE Transactions on Power Electronics*, vol. 28, no. 8, pp. 3912–3922, 2013.
- [38] E. Sontag and Y. Wang, “On characterizations of the input-to-state stability property,” *Systems And Control Letters*, vol. 24, pp. 351–359, 07 1996.
- [39] H. J. Marquez, *Nonlinear control systems*. John Wiley & Sons., 2003.
- [40] M. G. Quijano Ortega and C. G. Hernández Capacho, “Obtención experimental de los parámetros del motor que se utilizará en el sistema de locomoción de una esfera rodante,” Master’s thesis, Universidad Pontificia Bolivariana, may 2009.
- [41] B. C. Kuo, *Sistemas de control automático*. Prentice Hall Hispanoamericana, 1996. [Online]. Available: <https://books.google.com.mx/books?id=GyWr6cT8SEsC>

Anexos

A.1. Diagrama de conexiones



B.2. Caracterización del motor de CD

Es fundamental contar con una estimación aproximada de los parámetros de los motores para aplicar de manera efectiva la técnica de control, ya sea un control PID (Proporcional-Integral- Derivativo) o ADRC. Los parámetros clave incluyen la resistencia de la armadura (R_a), la inductancia de la armadura (L_a), el momento de inercia (J) y la constante de fricción de Coulomb (B). Además, se requiere la estimación de parámetros auxiliares como lo son la constante contraelectromotriz (K_e), la constante de torque (K_t) y la constante de tiempo mecánica (t_m). Para obtener estos valores, es necesario realizar una serie de mediciones y cálculos que se describen a continuación.

B.2.1. Obtención de R_a

Para determinar el valor de R_a , se midió la resistencia de armadura de cada motor empleando un multímetro conectado a las terminales + y - correspondientes. Las mediciones se realizaron con los motores completamente desconectados y sin alimentación de ninguna fuente de voltaje. Como resultado, se obtuvieron las siguientes resistencias de armadura promedio para cada motor:

$$R_{a_1} = 3.3 \Omega, \quad R_{a_2} = 2.6 \Omega, \quad R_{a_3} = 3.1 \Omega$$

B.2.2. Obtención de L_a

Para determinar el valor de L_a , se utilizó un medidor de inductancia que opera a diferentes frecuencias. Para esto, se realizaron múltiples mediciones para cada motor, y los valores obtenidos se promediaron. Los resultados finales se presentan en la Tabla 1.

Tabla 1: Valores de inductancia en cada motor.

Frecuencia	Motor 1 L_{a_1}	Motor 2 L_{a_2}	Motor 3 L_{a_3}
100 Hz	4.53 mH	3.62 mH	4.62 mH
120 Hz	4.52 mH	3.63 mH	4.63 mH
1 kHz	4.31 mH	3.44 mH	4.40 mH
10 kHz	4.63 mH	450 μ H	4.74 mH
100 kHz	26.70 μ H	26.57 μ H	26.97 μ H

Los valores de la inductancia de armadura promedio para cada motor son:

$$L_{a_{1prom}} = 3.60 \text{ mH}, \quad L_{a_{2prom}} = 2.23 \text{ mH}, \quad L_{a_{3prom}} = 3.68 \text{ mH}$$

B.2.3. Obtención de K_e

Para poder determinar el valor del momento de inercia J es necesario conocer la constante contraelectromotriz (K_e). Siguiendo el método propuesto por Quijano y Hernández [40], se empleó (1), en la cual se sustituyeron los obtenidos de las mediciones, variando el voltaje desde 1 V hasta alcanzar el voltaje máximo de 12 V.

$$K_e = \frac{V_a - R_a \cdot I_a}{\omega} \quad (1)$$

Los valores obtenidos de las mediciones se muestran en la Tabla 2, donde I_a se expresa en mA , ω en rad/s y K_e en $\frac{V \cdot s}{rad}$

Tabla 2: Valores de corriente, velocidad angular y constante contraelectromotriz para cada motor.

Voltaje [V]	Motor 1			Motor 2			Motor 3		
	I_{a1}	ω_1	K_{e1}	I_{a2}	ω_2	K_{e2}	I_{a3}	ω_3	K_{e3}
1	60	0.83	0.9663	50	0.79	1.1013	50	0.87	0.9713
2	70	1.79	0.9883	60	1.71	1.0784	60	1.75	1.0366
3	80	2.76	0.9913	80	2.63	1.0616	70	2.68	1.0384
4	90	3.76	0.9848	80	3.49	1.0865	80	3.62	1.0365
5	100	4.73	0.9873	90	4.52	1.0544	90	4.60	1.0263
6	110	5.71	0.9872	110	5.54	1.0314	90	5.51	1.0383
7	110	6.70	0.9906	110	6.49	1.0345	100	6.51	1.0276
8	120	7.66	0.9927	110	7.43	1.0382	110	7.44	1.0294
9	120	8.57	1.0040	120	8.35	1.0405	110	8.38	1.0333
10	130	9.50	1.0075	130	9.31	1.0378	110	9.36	1.0319
11	130	10.36	1.0204	130	10.15	1.0504	120	10.38	1.0239
12	140	11.36	1.0157	140	11.25	1.0343	120	11.33	1.0263

Los valores de K_e promediados para cada motor son:

$$K_{e1prom} = 0.9947 \frac{V \cdot s}{rad}$$

$$K_{e2prom} = 1.0541 \frac{V \cdot s}{rad}$$

$$K_{e3prom} = 1.0267 \frac{V \cdot s}{rad}$$

Además, como se asegura en [40], utilizando la técnica paramétrica dimensional se puede de-

terminar que la constante de torque es equivalente a la constante contraelectromotriz ($K_t = K_e$), únicamente cambiando las unidades de la constante de torque K_t . De esta forma se tiene que:

$$\begin{aligned}K_{t_{1prom}} &= 0.9947 \frac{\text{N} \cdot \text{m}}{\text{A}} \\K_{t_{2prom}} &= 1.0541 \frac{\text{N} \cdot \text{m}}{\text{A}} \\K_{t_{3prom}} &= 1.0267 \frac{\text{N} \cdot \text{m}}{\text{A}}\end{aligned}$$

B.2.4. Obtención de J

Ahora se determina el momento de inercia J de cada motor. Para ello se despeja el momento de inercia de la ecuación del tiempo de subida, quedando la expresión mostrada en (2).

$$J = \frac{K_t \cdot K_e \cdot t_m}{R_a} \quad (2)$$

En esta expresión, los parámetros K_t , K_m y R_a , son conocidos, mientras que la constante de tiempo mecánica t_m se desconoce. Este último parámetro se determinó siguiendo el procedimiento descrito en [40], que consiste en alimentar el motor con una entrada escalón hasta su valor nominal y medir el tiempo requerido para que la señal de salida alcance el 63.2% de su valor final. Dicho tiempo corresponde a la constante de tiempo mecánica t_m .

Los valores de t_m encontrados para cada motor son los siguientes:

$$t_{m_1} = 50 \mu\text{s}, \quad t_{m_2} = 52 \mu\text{s}, \quad t_{m_3} = 50 \mu\text{s}$$

Con los parámetros ya determinados, se sustituyen en (2) para calcular el momento de inercia de cada motor. Los valores obtenidos son los siguientes:

$$J_1 = 15.0 \times 10e^{-6} \text{ kg} \cdot \text{m}^2$$

$$J_2 = 22.2 \times 10e^{-6} \text{ kg} \cdot \text{m}^2$$

$$J_3 = 17.0 \times 10e^{-6} \text{ kg} \cdot \text{m}^2$$

B.2.5. Obtención de B

Finalmente, se determina el valor de la constante de fricción de Coulomb B . Utilizando (3) que relaciona el momento de inercia J , la velocidad angular ω , la constante de torque K_t y la fricción de Coulomb B utilizada en [40].

$$J \frac{d\omega}{dt} = K_t - B \cdot \omega \quad (3)$$

Considerando el estado estacionario, la velocidad angular ω es constante, por lo tanto su derivada es 0. De este modo la ecuación 3 queda como:

$$K_t - B \cdot \omega = 0 \quad (4)$$

B.2. Caracterización del motor de CD

Despejando B de (4) y manipulando se obtiene:

$$B = \frac{K_e \cdot I_a}{\omega} \quad (5)$$

Utilizando los valores de la Tabla 2 y sustituyéndolos en (5) se obtuvieron los datos que se muestran en la Tabla 3.

Tabla 3: Valores de B para cada motor.

Voltaje [V]	Motor 1	Motor 2	Motor 3
1	0.0699	0.0697	0.0558
2	0.0386	0.0378	0.0355
3	0.0287	0.0323	0.0271
4	0.0236	0.0249	0.0229
5	0.0209	0.0210	0.0201
6	0.0190	0.0205	0.0170
7	0.0163	0.0175	0.0158
8	0.0156	0.0154	0.0152
9	0.0141	0.0150	0.0136
10	0.0138	0.0145	0.0121
11	0.0128	0.0135	0.0118
12	0.0125	0.0129	0.0109

Posteriormente se promediaron los valores, obteniendo así el valor de B para cada motor, como se muestra a continuación:

$$B_{1prom} = 0.0238 \text{ N} \cdot \text{m} \cdot \text{s}, \quad B_{2prom} = 0.0246 \text{ N} \cdot \text{m} \cdot \text{s}, \quad B_{3prom} = 0.0215 \text{ N} \cdot \text{m} \cdot \text{s}$$

C.3. Simulación del motor de CD

Una vez que los motores de corriente directa (CD) fueron caracterizados, se procedió a simular su comportamiento dinámico utilizando Matlab/Simulink®. El objetivo de esta simulación fue comprobar que el sistema simulado se asemeja al comportamiento del sistema real.

Para modelar el motor, se utilizaron las ecuaciones (6)-(8), las cuales describen su dinámica:

$$L \frac{di_a}{dt} = V_a - R_a \cdot i_a - K \cdot \omega_m \quad (6)$$

$$J \frac{d\omega_m}{dt} = K \cdot i_a - B \cdot \omega_m - \tau_L \quad (7)$$

$$y = \omega_m \quad (8)$$

En la Figura 1 se presenta el diagrama de bloques utilizado para la simulación de los motores, implementados en Simulink®. En la simulación, se buscó analizar el comportamiento de los motores cuando fueron excitados con un voltaje de 12 V.

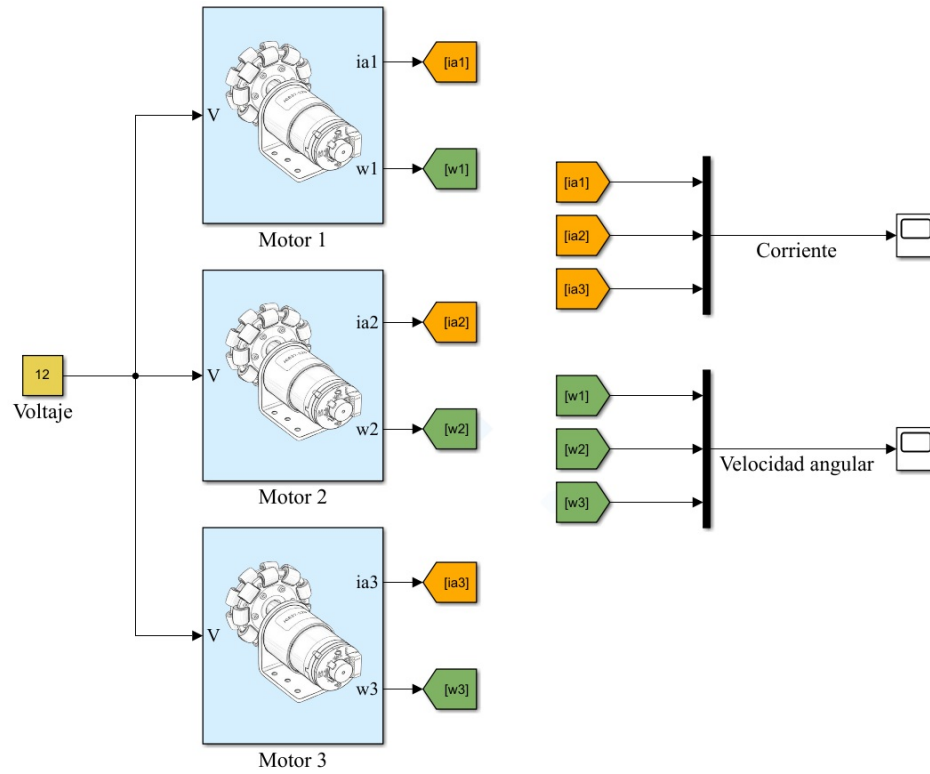


Figura 1: Diagrama de bloques del sistema dinámico en lazo abierto.

En las Figuras 2 y 3 se observa el comportamiento de las corrientes de armadura y la velocidad angular de los motores a lo largo del tiempo. Se aprecia que todas convergieron y se estabilizaron en valores cercanos a los obtenidos durante las pruebas experimentales, los cuales se observan en la Tabla 2 correspondientes a la corriente y velocidad de los motores alimentados a 12 V.

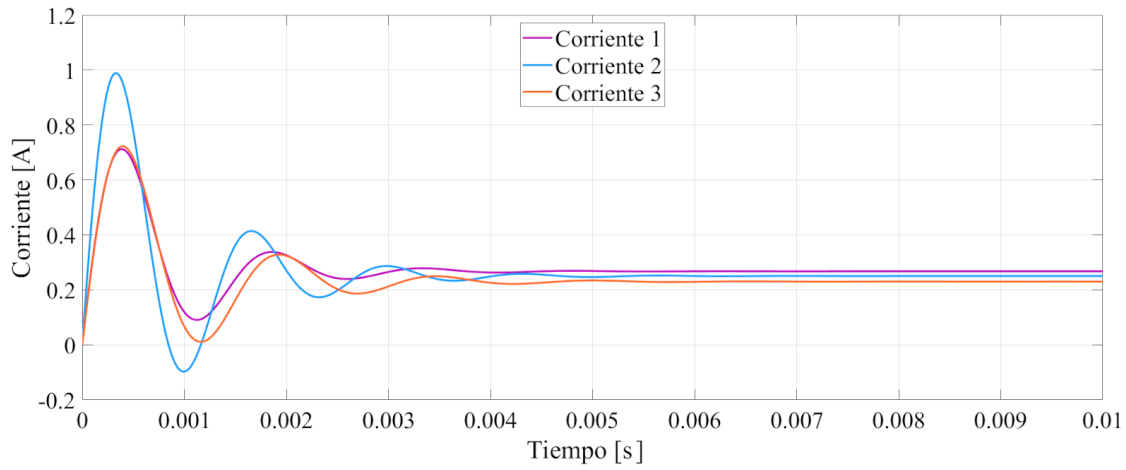


Figura 2: Corrientes de los motores en lazo abierto.

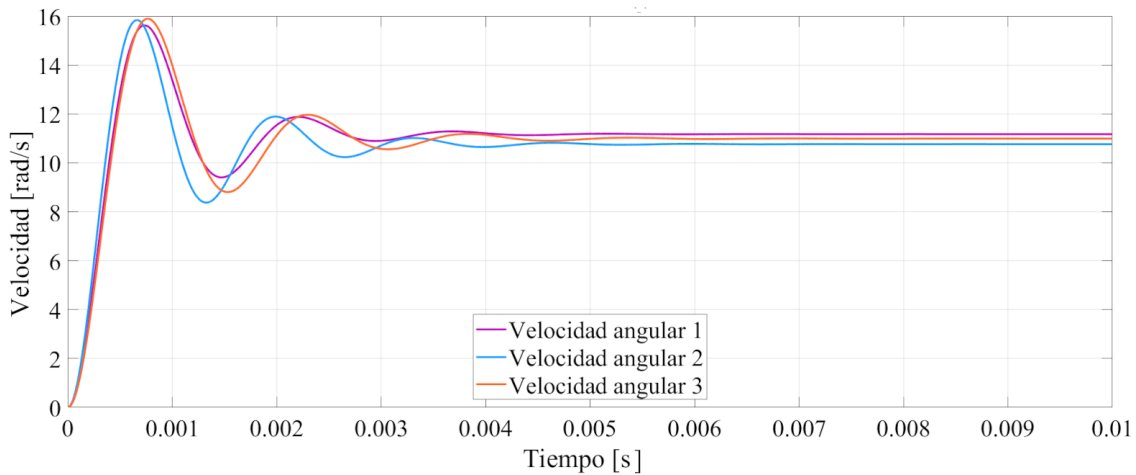


Figura 3: Velocidades de los motores en lazo abierto.

La simulación permitió verificar la correcta parametrización del motor, lo cual es fundamental para posteriormente diseñar el control ADRC.

D.4. Sintonización del controlador PID

Para una sintonización analítica precisa, se partió del modelo dinámico del motor de CD con el objetivo de encontrar las ganancias K_p , K_i y K_d que ubiquen los polos del sistema en lazo cerrado en posiciones que satisfagan los requerimientos de desempeño.

El comportamiento del motor está regido por el sistema de ecuaciones diferenciales acopladas mostrado en (6)-(8). Combinando (6) y (7) (despejando la corriente $i_a(t)$ de la ecuación mecánica y sustituyéndola en la eléctrica), se obtiene la ecuación diferencial de segundo orden que relaciona el voltaje con la velocidad:

$$L \cdot J \frac{d^2 \omega_m}{dt^2} + (L \cdot B + R_a \cdot J) \frac{d\omega}{dt} + (R_a \cdot B + K^2) \omega = K \cdot V_a(t) \quad (9)$$

Para simplificar la notación en el diseño del control, la ecuación (9) se reescribe como:

$$\gamma \ddot{\omega}_m + \beta \dot{\omega}_m + \alpha \omega = K \cdot V_a(t) \quad (10)$$

donde $\gamma = L \cdot J$, $\beta = L \cdot B + R_a \cdot J$ y $\alpha = R_a \cdot B + K^2$.

La ley de control PID en el dominio del tiempo se define como:

$$V_a(t) = K_p \cdot e(t) + K_i \int e(t) dt + K_d \cdot \frac{de(t)}{dt} \quad (11)$$

Con $e(t) = \omega_m(t) - \omega_{ref}$. Al sustituir la ley de control en la ecuación dinámica del motor y aplicar la Transformada de Laplace para analizar los polos, se obtiene el polinomio característico en lazo

cerrado del sistema real:

$$P(s) = s^3 + \frac{\beta + K \cdot K_d}{\gamma} s^2 + \frac{\alpha + K \cdot K_p}{\gamma} s + \frac{K \cdot K_i}{\gamma} = 0 \quad (12)$$

Para determinar las ganancias del controlador, se empleó la técnica de asignación de polos. Este método analítico, descrito formalmente por Kuo [41], permite imponer una dinámica de lazo cerrado específica mediante la selección de las raíces de la ecuación característica. Se define un comportamiento deseado mediante un polinomio de Routh-Hurwitz de tercer orden, compuesto por un par de polos complejos conjugados (dominantes) y un polo real rápido:

$$P_d(s) = (s^2 + 2\zeta\omega_n s + \omega_n^2)(s + p) = s^3 + (2\zeta\omega_n + p)s^2 + (\omega_n^2 + 2\zeta\omega_n p)s + p\omega_n^2 \quad (13)$$

Igualando coeficiente a coeficiente (12) con (13), se obtienen las fórmulas directas para las ganancias en función de los parámetros físicos del robot:

Ganancia Derivativa (K_d):

$$K_d = \frac{\gamma(2\zeta\omega_n + p) - \beta}{K}$$

Ganancia Proporcional (K_p):

$$K_p = \frac{\gamma(\omega_n^2 + 2\zeta\omega_n p) - \alpha}{K}$$

Ganancia Integral (K_i):

$$K_i = \frac{\gamma(p\omega_n^2)}{K}$$

Este procedimiento asegura que la dinámica del motor responda según los parámetros de diseño ζ , ω_n y p , compensando la inercia J y la inductancia L propias del sistema.

E.5. Herramientas de software

Se enumeran las herramientas de software empleadas durante las etapas de diseño, simulación, manufactura y control de ROCKY. A continuación, se describe brevemente la naturaleza de cada software y su función específica dentro del desarrollo de este proyecto, respetando el orden cronológico de su implementación.

- **SolidWorks®** : Software de diseño asistido por computadora (CAD) para modelado mecánico en 3D, utilizado ampliamente en ingeniería para la creación de piezas y ensamblajes paramétricos. Se empleó la versión SolidWorks® Premium SP4.1 para el diseño y modelado dimensional de todos los componentes mecánicos, incluyendo las placas del chasis del robot y el alojamiento de las baterías.
- **VISI CAD/CAM®**: Solución integral de software para el diseño de moldes y mecanizado, especializada en la generación de trayectorias de corte para manufactura asistida por computadora (CAM). Se utilizó la versión 15.0 para procesar los diseños geométricos provenientes de SolidWorks® y generar los códigos G necesarios. Estos archivos permitieron la manufactura automatizada de las piezas de acrílico del chasis mediante un router CNC.
- **KiCad EDA®**: Suite de automatización de diseño electrónico (EDA) de código abierto, diseñada para la creación de esquemas eléctricos y diseño de circuitos impresos. Mediante la versión 9.0.3, se realizó el diseño esquemático y el trazado de pistas (layout) de la placa de circuito impreso (PCB) dedicada a la interconexión de los drivers y el microcontrolador del robot.
- **CoppeliaSim®**: Entorno de simulación robótica (anteriormente conocido como V-REP) que

permite modelar interacciones físicas realistas mediante diversos motores dinámicos integrados. Se utilizó la versión CoppeliaSim® Edu 4.9.0 (rev. 6) para validar virtualmente la geometría del robot, así como para poner a prueba las ecuaciones del modelo cinemático desarrolladas antes de su implementación física.

- **Matlab® / Simulink®**: Plataforma de programación y cálculo numérico que, a través de su entorno gráfico de diagramas de bloques (Simulink), facilita el modelado, simulación y análisis de sistemas dinámicos multidominio. Se empleó la versión R2021a para establecer una co-simulación con CoppeliaSim®, permitiendo la adquisición y graficado de datos. Asimismo, se utilizó Simulink para diseñar y validar la arquitectura de los controladores propuestos mediante diagramas de bloques.
- **Arduino IDE**: Entorno de Desarrollo Integrado (IDE) de código abierto que facilita la escritura, compilación y carga de código en microcontroladores compatibles. Se utilizó la versión 1.8.19 para desarrollar el firmware del microcontrolador ESP32. Este software gestiona la regulación de velocidad de bajo nivel de los motores y administra la comunicación serial con la unidad de procesamiento externa.
- **Python**: Lenguaje de programación de alto nivel, interpretado y multipropósito, ampliamente utilizado en robótica por su versatilidad en el manejo de datos y algoritmos de control. Se empleó como el entorno de ejecución principal para el control cinemático de alto nivel y la generación de trayectorias. Python gestiona la lógica de navegación y envía las consignas de velocidad al microcontrolador a través de comunicación inalámbrica/serial (ver librerías en Tabla 4.1).