



UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE INGENIERÍA CAMPUS MEXICALI
PROGRAMA DE POSGRADO MYDCI

DESARROLLO DE UN ALGORITMO DE CONTROL
ROBUSTO PARA REGULAR LA ORIENTACIÓN Y
VELOCIDAD DE MOVIMIENTO DE UN ROBOT MÓVIL
OMNIDIRECCIONAL

T E S I S

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN CIENCIAS

PRESENTA:

JOSUÉ ABRAHAM RIVAS CARRILLO

DIRECTOR:

DR. DAVID ISAÍAS ROSAS ALMEIDA



Mexicali, Baja California, mayo del 2025

Dedicatoria ...

A mi padre, mi madre, mis hermanos, mi hermana y mi novia, que me motivan y apoyan siempre a seguir mis sueños.

Agradecimientos

A mis padres Moisés Rivas López y Guillermina Carrillo Cabrera, que nunca ha dejado de apoyarme, incluso en momentos difíciles, gracias por todo su amor y creer en mí.

A mis hermanos Moisés Emanuel Rivas Carrillo, Aarón David Rivas Carrillo, Consuelo Sarahí Rivas Carrillo y a mi novia Emma Fernanda Encinas Moreno, por tolerar todas mis charlas sobre aprendizajes nuevos de control y experiencias escolares que me emocionan.

Agradezco especialmente al Dr. David Isaías Rosas Almeida, por su apoyo y guía académica de alta calidad en este proyecto de tesis. Por ayudarme a entender una materia tan compleja como lo es la rama de control.

A los integrantes del comité de tesis: Dra. Rosa Citlalli Anguiano Cota, Dr. Ernesto Víctor González Solís, Dr. Iván Olaf Hernández Fuentes y al Dr. Álvaro González Ángeles por la retroalimentación y sugerencias semestre tras semestre.

Gracias a la Universidad Autónoma de Baja California, a la facultad de Ingeniería campus Mexicali y al programa de Maestría y Doctorado en Ciencias e Ingeniería.

Por permitirme crecer profesionalmente y ayudarme a cumplir mis metas, gracias.

Índice general

Agradecimientos	II
1 Introducción	1
1.1 Justificación	2
1.2 Objetivos	3
1.2.1 Objetivo general	3
1.2.2 Objetivos específicos	3
1.3 Metodología	4
1.4 Estructura del documento	5
2 Antecedentes	7
2.1 Manufactura aditiva e impresión en 3D	7
2.2 Sistema	8
2.2.1 Sensores	8
2.2.2 Actuadores	12
2.2.3 NI MyRIO1900	12
2.3 Modelo matemático	13
2.3.1 Identificación de sistemas	15
2.3.2 Sistemas de 1er orden	16
2.3.3 Sistemas de control en lazo abierto y lazo cerrado	19
2.4 Objetivos de control	20
2.5 Controlador regulador y seguidor de trayectorias	21
2.6 Acciones de control	22
2.6.1 Acción Proporcional	22

2.6.2	Acción Integral	24
2.6.3	Acción Proporcional-Integral	25
2.6.4	Controlador por modos deslizantes	26
2.7	Sincronización de sistemas dinámicos	30
2.7.1	Sincronización Maestro-Esclavo	31
2.8	Robots móviles	31
2.9	Modelado cinemático de sistemas mecánicos	32
3	Diseño y construcción de un robot móvil omnidireccional con ruedas	
	Mecanum	34
3.1	Ruedas Mecanum	35
3.1.1	Movimientos típicos de un robot móvil omnidireccional	36
3.1.2	Modelo cinemático directo e inverso del RMO	38
3.2	Descripción general de las características del RMO	39
3.3	Diseño y construcción de los subsistemas de instrumentación y control . . .	40
3.3.1	Motor de 12V Pololu con encoder de 464.64 CPR	40
3.3.2	IMU: 3 Space Sensor de Yost Labs	41
3.3.3	NI MyRIO 1900 dentro del RMO	43
3.4	Diseño y construcción de los subsistemas eléctricos de potencia	45
3.4.1	Batería y cargado de batería	45
3.4.2	Puente H	45
3.5	Diseño y construcción de los subsistemas mecánicos	46
3.5.1	Ruedas Mecanum	46
3.5.2	Impresión en PLA para el diseño del chasis del RMO	48
4	Desarrollo de módulos de control utilizando la plataforma myRIO y	
	LabVIEW	53
4.1	Programa para el guardado de datos eficiente a través de una NI MyRIO	
1900	54
4.1.1	Acceso a los datos guardados dentro de la tarjeta NI MyRIO	56

4.1.2	Tiempo proveído por el lazo de simulación y control, contra tiempo medido por la MyRIO	60
4.2	Programa para generar movimiento en un motor de CC	62
4.3	Programa para generar los movimientos típicos de un RMO	63
4.4	Programa para leer los datos de censado de orientación de un IMU (3 space-sensor by Yost Labs)	66
4.5	Funcionamiento del programa completo de movimiento omnidireccional de un RMO, con censado del angulo de orientación y guardado de datos . . .	68
5	Modelado matemático e identificación de parámetros de los 4 motores de CC con base en experimentos	73
5.1	Experimentación y obtención de la función de transferencia estimada para los 4 motores	74
5.2	Validación de la función de transferencia	76
6	Desarrollo de algoritmos de control para el control de velocidad de las ruedas	81
6.1	Propuesta de un algoritmo de control en base a retroalimentación de estado	81
6.2	Simulaciones del RMO aplicando diferentes ganancias de control.	83
6.3	Experimentación del RMO sobre concreto aplicando diferentes ganancias de control.	90
7	Desarrollo de algoritmos de control robusto, de velocidad y orientación del robot, basados en sincronización	96
7.1	Esquemas de sincronización	97
7.2	Simulación de control robusto con diferentes esquemas de sincronización con las diferentes ganancias de control	98
7.3	Experimentación de control robusto con diferentes esquemas de sincronización con las diferentes ganancias de control y resultados	99
8	Conclusiones y trabajo futuro	110

Índice de figuras

2.1	Funcionamiento típico de un Encoder.	9
2.2	3 space-sensor de Yost Labs.	10
2.3	Diagrama de comunicacion SPI entre un componente maestro y uno esclavo.	11
2.4	Motor típico de DC.	12
2.5	Imagen de una MyRIO real sin conexiones.	13
2.6	Hardware de la tarjeta NI-MyRIO-1900 obtenido de [28].	14
2.7	a) Diagrama de un sistema de control en lazo cerrado. b) Diagrama de un sistema de control en lazo abierto.	20
2.8	Comportamiento típico de un controlador regulador.	23
2.9	Comportamiento típico de un controlador seguidor de trayectorias.	23
2.10	Influencia de un controlador proporcional sobre un sistema ideal con respuesta predominantemente de 1er orden.	24
2.11	Influencia de un controlador integral sobre un sistema ideal con respuesta predominantemente de 1er orden.	26
2.12	Influencia de un controlador proporcional-integral sobre un sistema ideal con respuesta predominantemente de 1er orden.	27
2.13	Influencia de un controlador regulador con modos deslizantes sobre un sistema ideal con respuesta predominantemente de 1er orden.	29
2.14	Influencia de un controlador seguidor de trayectoria con modos deslizantes sobre un sistema ideal con respuesta predominantemente de 1er orden.	29
2.15	Diagrama del esquema de sincronización unidireccional o Maestro-Esclavo.	32
2.16	Diagrama de clasificación de robots móviles	32
2.17	Diagrama general de la relación entre	33

3.1	Diagrama de una rueda Mecanum con sus grados de libertad.	35
3.2	Vista superior de una rueda Mecanum zurda y una diestra.	36
3.3	Vista superior de un RMO, con los respectivos vectores de velocidad generados en cada rueda.	37
3.4	Movimientos típicos de un RMO de 4 ruedas con la condición de dirección de cada una de sus ruedas.	37
3.5	Diagrama de interconexiones del RMO, entre sus respectivos sensores, actuadores y procesador.	40
3.6	Diagrama de conexiones del motor Pololu.	41
3.7	Diagrama de conexiones con dimensiones del sensor Yost Labs 3 Space-Sensor	42
3.8	Diagrama de conexiones de los lados A y B de la MyRIO.	44
3.9	Diagrama de conexiones del lado C de la MyRIO.	44
3.10	Diagrama con medidas de una tarjeta MyRIO-1900.	45
3.11	Diagrama con medidas de la batería de alimentación del RMO.	46
3.12	Diagrama con medidas de un L2988N.	47
3.13	Diagrama de conexiones de los 4 puentes H.	47
3.14	Dimensiones de la rueda Mecanum utilizada en el RMO.	48
3.15	Piezas 3D impresas en PLA para el chasis del RMO.	49
3.16	Diagrama con medidas de las placas impresas en 3D para el soporte de la MyRIO y para el soporte de la pila.	50
3.17	Diagrama con medidas de las placas impresas en 3D para el chasis principal del RMO y su soporte.	51
3.18	Vista explotada 3D del RMO, con sus componentes base.	52
4.1	Código visual de LabVIEW para el guardado de datos en MyRIO.	55
4.2	Diagrama del <i>SubVi</i> encargado del conteo de tiempo en segundos con resolución en micro segundos.	55
4.3	Diagrama del <i>SubVi</i> encargado de convertir datos en un archivo delimitado por comas.	56
4.4	Panel sistemas remotos de NI MAX.	57

4.5	Opciones extras (Click derecho) sobre la MyRIO en NI MAX.	57
4.6	Ingreso de información para acceder a los datos internos de la MyRIO a través de google Chrome.	58
4.7	Directorio interno de la MyRIO para el acceso a los experimentos guardados internamente.	58
4.8	Obtención de datos desde NI MAX y google chrome, de un experimento guardado internamente en una MyRIO-1900	59
4.9	Dirección IP de una MyRIO como enlace en microsoft edge/ internet explorer.	60
4.10	Ingreso de información para acceder a los datos internos de la MyRIO a través de Internet Explorer.	61
4.11	Directorio interno de la MyRIO para el acceso a los experimentos guardados internamente desde Internet Explorer.	61
4.12	Vista de una comparativa de la toma de datos utilizando el tiempo proveído por el lazo de simulación y control y el tiempo medido por la MyRIO. . . .	61
4.13	Comparación de datos con el tiempo proveído por el lazo de simulación y control contra el tiempo medido por la MyRIO.	62
4.14	Diagrama de bloques y conexiones para la conversión de cuentas del encoder a radianes por segundo.	63
4.15	Panel frontal y diagrama de conexiones de el programa para generar movimiento y dirección de un motor de CC.	64
4.16	Diagrama de bloques del programa para generar movimientos omnidireccionales en un RMO.	65
4.17	Panel frontal del programa para generar movimientos omnidireccionales en un RMO.	65
4.18	Diagrama de bloques del <i>SubVi</i> , encargado de convertir los botones booleanos de dirección, a referencias numéricas para cada rueda.	66
4.19	Diagrama de bloques y conexiones para el <i>Vi</i> , encargado de leer el angulo de guiñada del RMO.	67
4.20	Diagrama de bloques y conexiones para el <i>SubVi</i> , encargado de abrir las referencias para utilizar el IMU.	67

4.21	Diagrama de bloques y conexiones para el <i>SubVi</i> , encargado de la lectura de un dato del ángulo de guiñada.	69
4.22	Diagrama de bloques y conexiones para el <i>SubVi</i> , encargado de enviar un comando al IMU.	69
4.23	Diagrama de bloques y conexiones para el <i>SubVi</i> , encargado de convertir los 4 bytes proporcionados por el IMU, en un número flotante representando el ángulo de Euler de guiñada.	70
4.24	Diagrama de bloques y conexiones para el <i>SubVi</i> , encargado de cerrar las referencias de comunicación con la MyRIO.	70
4.25	Panel frontal completo para la lectura de velocidad angular en cada rueda y control de movimiento en lazo abierto de un RMO, con censado del ángulo de guiñada y guardado de datos.	71
4.26	Diagrama de bloques completo para la lectura de velocidad angular en cada rueda y control de movimiento en lazo abierto de un RMO, con censado de ángulo de guiñada y guardado de datos.	72
5.1	4 de los 10 experimentos mostrando la respuesta de los 4 motores ante series de entradas escalón.	75
5.2	Respuestas del sistema real y de la respuesta simulado para la validación de los modelos matemáticos obtenidos.	77
5.3	Programa de Simulink para simular el comportamiento del RMO en lazo abierto.	78
5.4	Velocidad angular de los motores al aplicarle una entrada PWM del 100 % durante 7 segundos.	79
5.5	Posición del RMO, respecto a sus coordenadas móviles al aplicarle una entrada de PWM del 100 % durante 7 segundos.	79
5.6	Posición del RMO respecto a sus coordenadas globales al aplicarle una entrada del PWM del 100 % durante 7 segundos.	80
6.1	Programa de Simulink para simular el comportamiento del RMO con controladores individuales en cada rueda.	84

6.2	Angulo de orientación y trayectoria del RMO de las 8 simulaciones con movimientos hacia adelante.	85
6.3	Angulo de orientación y trayectoria del RMO de las simulaciones con los mejores controladores con movimiento hacia adelante.	86
6.4	Angulo de orientación y trayectoria del RMO de las 8 simulaciones con movimiento transversal hacia la derecha.	87
6.5	Angulo de orientación y trayectoria del RMO de las simulaciones con los mejores controladores con movimiento transversal hacia la derecha.	88
6.6	Angulo de orientación y trayectoria del RMO de las 8 simulaciones con movimiento en diagonal.	88
6.7	Angulo de orientación y trayectoria del RMO de las simulaciones con los mejores controladores con movimiento en diagonal.	89
6.8	Imagen real del RMO fabricado.	90
6.9	Superficie de concreto plano no pulido, con irregularidades controladas, sin hoyos profundos, ni sustancias no reconocidas.	90
6.10	Angulo de orientación y trayectoria del RMO sobre concreto, de los 8 experimentos de las diferentes estrategias de control, realizando un movimiento delantero.	91
6.11	Angulo de orientación y trayectoria del RMO sobre concreto, de las mejores estrategias de control aplicadas, realizando un movimiento delantero.	92
6.12	Angulo de orientación y trayectoria del RMO sobre concreto, de los 8 experimentos de las diferentes estrategias de control, realizando un movimiento transversal hacia la derecha.	93
6.13	Angulo de orientación y trayectoria del RMO sobre concreto, de las mejores estrategias de control aplicadas, realizando un movimiento transversal hacia la derecha.	93
6.14	Angulo de orientación y trayectoria del RMO sobre concreto, de los 8 experimentos de las diferentes estrategias de control, realizando un movimiento diagonal.	94

6.15	Angulo de orientación y trayectoria del RMO sobre concreto, de las mejores estrategias de control aplicadas, realizando un movimiento diagonal.	94
7.1	Propuestas de esquemas de sincronización entre los 4 motores de un RMO.	97
7.2	Velocidades angulares y trayectorias simuladas del RMO, con acción proporcional-integral sin y con sincronización.	99
7.3	Simulación de la acción proporcional integral, sin y con los diferentes esquemas de sincronización.	100
7.4	Simulación de la acción integral con modos deslizantes, sin y con los diferentes esquemas de sincronización.	100
7.5	Simulación de la acción proporcional integral con modos deslizantes, sin y con los diferentes esquemas de sincronización.	101
7.6	Respuesta del RMO sobre concreto con un movimiento frontal, con acción proporcional integral, sin y con los diferentes esquemas de sincronización sobre las ruedas.	102
7.7	Respuesta del RMO sobre concreto con un movimiento frontal, con acción proporcional con modos deslizantes, sin y con los diferentes esquemas de sincronización sobre las ruedas.	102
7.8	Respuesta del RMO sobre concreto con un movimiento frontal, con acción integral con modos deslizantes, sin y con los diferentes esquemas de sincronización sobre las ruedas.	103
7.9	Respuesta del RMO sobre concreto con un movimiento frontal, con acción proporcional integral con modos deslizantes, sin y con los diferentes esquemas de sincronización sobre las ruedas.	103
7.10	Respuesta del RMO sobre concreto con un movimiento transversal hacia la derecha, con acción proporcional integral, sin y con los diferentes esquemas de sincronización sobre las ruedas.	104
7.11	Respuesta del RMO sobre concreto con un movimiento transversal hacia la derecha, con acción proporcional con modos deslizantes, sin y con los diferentes esquemas de sincronización sobre las ruedas.	104

7.12	Respuesta del RMO sobre concreto con un movimiento transversal hacia la derecha, con acción integral con modos deslizantes, sin y con los diferentes esquemas de sincronización sobre las ruedas.	105
7.13	Respuesta del RMO sobre concreto con un movimiento transversal hacia la derecha, con acción proporcional integral con modos deslizantes, sin y con los diferentes esquemas de sincronización sobre las ruedas.	105
7.14	Respuesta del RMO sobre concreto con un movimiento diagonal frontal hacia la derecha, con acción proporcional integral, sin y con los diferentes esquemas de sincronización sobre las ruedas.	107
7.15	Respuesta del RMO sobre concreto con un movimiento diagonal frontal hacia la derecha, con acción proporcional con modos deslizantes, sin y con los diferentes esquemas de sincronización sobre las ruedas.	107
7.16	Respuesta del RMO sobre concreto con un movimiento diagonal frontal hacia la derecha, con acción integral con modos deslizantes, sin y con los diferentes esquemas de sincronización sobre las ruedas.	108
7.17	Respuesta del RMO sobre concreto con un movimiento diagonal frontal hacia la derecha, con acción proporcional integral con modos deslizantes, sin y con los diferentes esquemas de sincronización sobre las ruedas.	108

Capítulo 1

Introducción

Hoy en día se presentan sistemas robóticos que en un momento se creyeron, serían solo fantasía, como los robots humanoides Atlas de Boston Dynamics, drones tanto para el entretenimiento en espectáculos visuales como para el uso personal, incluso los vehículos autónomos de Tesla, que no necesitan de un conductor para llevar un vehículo a un destino pre-programado, así como estos existen una gran cantidad de sistemas mecatrónicos que solucionan muchos tipos de problemas de manera creativa, responsable e inteligente. En el área de la robótica se cuenta con un apartado dedicado a los robots móviles, el cual es de gran interés actualmente, ya que estos se utilizan día a día, tanto para transportar personas a sus casas, trabajos, etc. como para transportar objetos en almacenes y hospitales; incluso se les puede encontrar en el área del entretenimiento y en departamentos de rescate de situaciones extremas como derrumbes, incendios, entre otros. Específicamente se investiga mucho el área de los robots móviles guiados automáticamente y uno de los mayores problemas que se han encontrado, se encuentra en la etapa de diseño, ya que, a pesar de utilizar una programación avanzada, la maniobrabilidad y evasión de obstáculos seguirá estrictamente limitada por el diseño mecánico del robot. Si bien existen categorías bien definidas de los robots móviles, cada una ofrece métodos de desplazamiento muy definidos y/o limitados, es decir existen vehículos marinos para transportarse sobre la superficie del agua, submarinos para transportarse debajo del agua, robots móviles aéreos como los aviones o helicópteros con movimientos muy definidos. Entre los robots móviles aéreos más populares se encuentran los drones que ofrecen un alto nivel de maniobrabilidad, debido a

su capacidad de moverse en un espacio de coordenadas XYZ. Estas características, se han deseado para robots móviles terrestres y para ello se han propuesto diferentes soluciones como los robots móviles con patas, pero que no suelen ofrecer las mismas ventajas que los robots móviles con ruedas, entre una de las soluciones propuestas, que proporciona una atractiva solución, son los robots móviles omnidireccionales con ruedas Mecanum. Estos al igual que los drones ofrecen un alto nivel de maniobrabilidad al prometer manipular un robot móvil sobre una superficie, independientemente de la orientación del vehículo, permitiéndole seguir trayectorias complejas sin cambiar la orientación del vehículo. Existen algunas áreas de mejora actualmente en el uso de Robots móviles omnidireccionales con ruedas Mecanum, varias de ellas se atienden desde el diseño mecánico de las ruedas, pero existe una problemática que actualmente se sigue investigando debido a que causa desviaciones difíciles de ignorar en el comportamiento del vehículo. Este comportamiento viene añadido por la capacidad del vehículo de deslizarse hacia cualquier dirección, ya que al momento de avanzar, de frenar o simplemente cambiar de dirección, genera derrapes no deseados, los cuales también suelen alterar su orientación, por lo cual se siguen investigando diversas estrategias de control que minimicen el error provocado por el derrape en las ruedas omnidireccionales.

1.1. Justificación

La tecnología avanza de manera exponencial cada día y con ello, surgen cada vez más problemáticas para mejorar la eficiencia de esa tecnología, para resolver estas problemáticas existen diferentes ramas, entre una de ellas está la rama de control, la cual se utiliza para manipular sistemas y que produzcan ciertos comportamientos deseados. Una problemática actual es la de manipular diferentes sistemas de manera que logren un objetivo en común de una manera eficiente. Este problema asume que el trabajo colaborativo de sistemas que participan en un mismo proceso puede mejorar la eficacia del proceso, con este argumento se busca implementar una sincronización de subsistemas de un sistema mecánico paralelo para mejorar la eficacia del mismo. Se busca mejorar el desempeño de un robot móvil omnidireccional con ruedas Mecanum, este tipo de vehículos son altamente investigados

en las áreas de navegación e industria automotriz debido a su tentativa propuesta de generar un movimiento en cualquier dirección del plano XY, sin alterar su orientación. Esta propuesta puede ayudar a mejorar la eficiencia de sistemas paralelos como drones, robots móviles, robots con ruedas, robots manipuladores paralelos, entre otros. Una de las pocas problemáticas a causa del uso de ruedas omnidireccionales son los movimientos no deseados a causa del deslizamiento en las ruedas. Se han propuesto diferentes estrategias de control tanto para las ruedas como para el sistema completo, así como observadores para el seguimiento de trayectorias, aun así, se siguen buscando alternativas como el método de sincronización controlada mencionado en [21], para comparar resultados, hasta encontrar diseños óptimos, aptos para su uso en cualquier área que involucre robots móviles.

1.2. Objetivos

1.2.1. Objetivo general

El objetivo general de este trabajo de tesis es el de documentar el proceso de diseño de un robot móvil omnidireccional didáctico fácil de recrear, junto con el análisis de diseño de un algoritmo de control robusto en base a controladores usuales como el proporcional e integral, junto con estrategias de control mas complejas como controlador por modos deslizantes y sincronización de sistemas dinámicos, a la vez diseñar un programa en LabVIEW que permita implementar esa ley de control robusto a un robot móvil omnidireccional.

1.2.2. Objetivos específicos

Los objetivos específicos se clasifican según los procesos realizados para la fabricación del robot móvil omnidireccional, así como para el desarrollo del código para controlar la velocidad de movimiento y orientación de un robot móvil omnidireccional.

1. Diseño y construcción de un robot móvil omnidireccional didáctico.
 - 1.1. Definir el hardware necesario para cubrir las funciones básicas de un robot móvil omnidireccional.

- 1.2. Diseñar un prototipo en CAD en base a las medidas del hardware a utilizar.
- 1.3. Fabricar y ensamblar el robot móvil omnidireccional.
2. Modelado matemático del sistema.
 - 2.1. Diseño de un código que manipule las funciones básicas de un robot móvil omnidireccional.
 - 2.2. Experimentación con el sistema para realizar una identificación en base a experimentos.
 - 2.3. Obtener un modelo matemático y validar su utilidad.
3. Diseño de un algoritmo de control que permita resolver problemas de regulación en cada una de las ruedas del robot.
4. Diseño de un código para implementar las propuestas de control formuladas.
5. Diseño de un algoritmo de control robusto basado en sincronización de sistemas dinámicos.
 - 5.1. Proponer diferentes esquemas de sincronización entre las ruedas del robot.
 - 5.2. Diseñar las funcionales de sincronización para los diferentes esquemas propuestos.
6. Diseño de un código para implementar un controlador con las diferentes propuestas de sincronización.

1.3. Metodología

Para el desarrollo de un algoritmo de control robusto para regular la orientación y velocidad de un robot móvil omnidireccional con sincronización controlada, se tomaron tres enfoques principales: Diseño de sistemas mecatrónicos, modelado matemático y experimental. La primera con el fin de obtener un sistema real con el cual realizar todas las pruebas necesarias a lo largo del proyecto, como analizar las funciones básicas de un

robot móvil omnidireccional, así como el análisis del robot ante la influencia de diferentes configuraciones de control y por último el análisis del robot según diferentes esquemas de sincronización. El enfoque de modelado matemático se tomó con el fin de establecer y aplicar diversas propuestas de controladores que permitan resolver o atenuar la problemática de regulación del robot, así como el diseño de propuestas de sincronización entre los cuatro motores del sistema. Además, se realizaron diversas simulaciones para validar las propuestas de control a través de diversas herramientas de apoyo matemático como MATLAB, Simulink y LabVIEW, este último también se utilizó como software principal para el desarrollo de un código en una tarjeta NI-MyRIO 1900 que permitiera el uso de funciones básicas de motores de CC, a través de PWM, la lectura de las velocidades angulares en cada uno de los motores utilizando encoders incrementales y la obtención de los ángulos de Euler a través de un sensor de unidad inercial “3 space sensor” de Yost Labs. Con la tarjeta NI-MyRIO 1900 también se implementaron las leyes de control propuestas dentro de la dinámica del robot móvil omnidireccional.

1.4. Estructura del documento

La estructura del documento es la siguiente. En el capítulo 2 se presentan todos los conocimientos que fueron necesarios para realizar y entender este trabajo de tesis, desde el diseño de fabricación de un robot móvil omnidireccional, los fundamentos matemáticos en los que fueron basadas las leyes de control y las funcionales de sincronización propuestas, así como los programas y protocolos utilizados para la simulación y programación del sistema a través de MATLAB, Simulink y LabVIEW. En el capítulo 3 se presenta el proceso detallado para el diseño, fabricación y construcción del robot móvil omnidireccional según todos sus subsistemas. Posteriormente, en el capítulo 4 se documenta el funcionamiento de cada uno de los programas principales utilizados para manipular al robot móvil omnidireccional con una tarjeta NI-MyRIO 1900 programada en LabVIEW. En el capítulo 5 se realizó un proceso para modelar los 4 motores de CC en base a experimentos y se validaron a través de la paquetería *System identification toolbox* de MATLAB, además se presentó un programa para simular el comportamiento de un robot móvil omnidireccional en base

a las ecuaciones de cinemática y los modelos de sus motores. En el capítulo 6 se presentan las leyes de control diseñadas para controlar la velocidad angular de los motores del robot y, por lo tanto, la generación de movimientos típicos de un robot móvil omnidireccional; en este capítulo también se realizaron simulaciones y, posteriormente, experimentos en ambientes reales para analizar el desempeño de los diferentes controladores propuestos. El capítulo 7 introduce diferentes esquemas de sincronización y funcionales de sincronización entre las cuatro ruedas del vehículo; se analizó el desempeño de las propuestas en comparación de los controladores sin sincronización tanto en simulación como en experimentos en ambientes reales. Por último, el capítulo 8 presenta el análisis de los resultados obtenidos y conclusiones de la investigación, así como posibles áreas de investigación futuras relacionadas.

Capítulo 2

Antecedentes

2.1. Manufactura aditiva e impresión en 3D

La manufactura aditiva es una de las técnicas modernas más populares para manufacturar piezas. Durante mucho tiempo se trabajó con lo que hoy se conoce como manufactura sustractiva, es decir, se tiene un material generalmente en forma de prisma cuadrangular, rectangular o un cilindro, al cual se le hacían cortes hasta obtener una pieza. Hoy en día se utiliza en muchos lugares la fabricación aditiva, ya que se produce menos material sobrante, debido a que en este método se empieza con nada, posteriormente poco a poco se agrega algún material o materiales, de forma controlada, hasta obtener una pieza [5]. Uno de los beneficios mayores en este método de manufactura es el costo, ya que se pueden elaborar piezas relativamente complejas con materiales plásticos baratos como el PLA, el PET, la resina y otros. El proceso típico para impresión en 3D, hoy en día, consta de realizar un diseño en alguna plataforma de CAD (Computer Aided Design), posteriormente pasar el diseño por un laminado, donde se obtendrán los comandos de coordenadas espaciales específicos para la máquina que vaya a imprimir la pieza. Una de las principales diferencias de la manufactura aditiva es el método de impresión. El método más popular de impresión aditiva hoy en día es la impresión FDM (Fused Deposition Modeling), que consta de extruir el material para administrarlo por una resistencia que lo funde y es empujado a través de una boquilla para obtener un plástico fluido sobre una cama de impresión especial, para que logre adherirse. Estas tiras de plástico fluido son depositadas de manera

controlada sobre la camilla en el espacio XY , hasta formar una de las capas y al finalizar todas sus coordenadas en XY , se forma lo que se conoce como capa. Al terminar una capa, la impresora da un pequeño paso en el eje Z , para agregar altura, es decir, volumen, a la pieza y forma otra capa. Este proceso se repite hasta que todas las capas estén terminadas, formando así una pieza 3D.

2.2. Sistema

Un sistema es el conjunto de elementos o subsistemas, ya sean eléctricos, electrónicos, mecánicos, químicos, ópticos, hidráulicos y/o neumáticos, que trabajan en grupo para obtener un resultado en específico [29]. En algunas ramas de la ingeniería se pueden ignorar los procesos individuales internos de un sistema y tratarse como un bloque conjunto, también conocido como planta, siempre y cuando los únicos datos de interés sean la(s) entrada(s) y la(s) salida(s) del sistema [6]. Un sistema puede denominarse de dos maneras: estático o dinámico. Un sistema estático es aquel cuya salida en un instante de tiempo solo depende del valor de sus entradas en ese instante de tiempo. En cambio, en un sistema dinámico su salida en un instante dado depende no solo de la entrada actual, sino también de todas las entradas aplicadas anteriormente, por lo tanto, su salida varía siempre que no esté en su estado de equilibrio [29].

2.2.1. Sensores

En electrónica, un sensor es un componente sensible que produce un cambio, que suele ser resistivo, reactivo, magnético o incluso puede generar una señal de voltaje, y esto es provocado ante algún fenómeno físico que interactúe directamente sobre el sensor, como el calor, la posición, la fuerza, intensidad de luz, etc., donde su respuesta es una función lineal o no lineal del cambio producido en el fenómeno físico [6, 39].

Encoder incremental

Un encoder o codificador incremental es un sensor que convierte el desplazamiento en una señal eléctrica. Los encoders más populares son los rotativos, aunque también existen

lineales y funcionan a través del mismo principio. Se requiere de un generador de señal que constantemente envía una señal a un sensor primario, y son interrumpidos por dos componentes ranurados, los cuales giran o se mueven linealmente según sea el caso, y al moverse, interrumpen la señal del generador y el sensor, marcando así un movimiento y una dirección [6, 23, 32]. Estos sensores se emplean ampliamente en sistemas electromecánicos hoy en día, por su eficacia, linealidad e incluso bajo costo. Según [2] son de las mejores opciones al medir velocidad angular, en comparación con otros sensores, como sensores de proximidad inductivos. El funcionamiento general de un encoder incremental rotativo se muestra de manera gráfica en la figura 2.1.

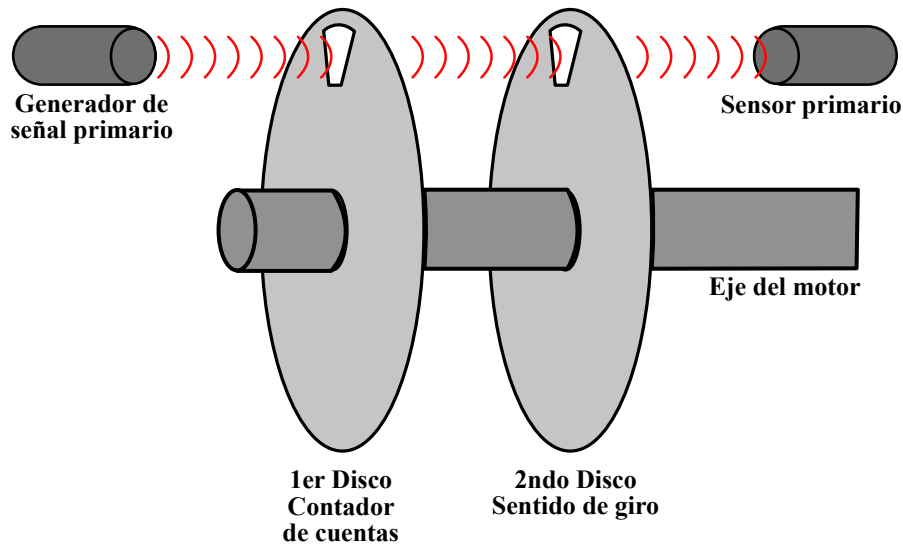


Figura 2.1: Funcionamiento típico de un Encoder.

Unidad de medición inercial (IMU) 3 Space

La unidad de medición inercial 3 Space es un sensor inteligente fabricado por la empresa Yost Labs, el cual puede medir la aceleración en las componentes X, Y, Z, la velocidad angular en los tres ejes, así como los ángulos de Euler a través de un acelerómetro, un giroscopio, un magnetómetro y una serie de algoritmos en tiempo real que permiten tener mediciones precisas y con un mínimo nivel de ruido. El 3 Space Sensor de Yost Labs

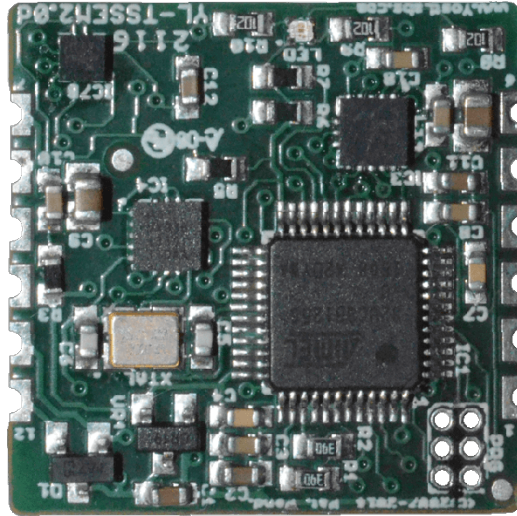


Figura 2.2: 3 space-sensor de Yost Labs.

tiene diferentes versiones comerciales que se diferencian principalmente por sus opciones de comunicación, es decir, existe una versión con comunicación WiFi, USB, microSD, Bluetooth o una versión embebida en la que es posible conectarlo según las necesidades del usuario. Es posible comunicarse con el sensor a través de diferentes protocolos de comunicación como comunicación USB 2.0, serial o SPI. Es importante comprender la forma de comunicarse con el sensor a través del protocolo SPI, ya que cuenta con más de 240 comandos diferentes. Al igual que muchos sensores que trabajan con protocolos de comunicación, la comunicación se envía en paquetes, donde se hacen peticiones al sensor en forma de bytes y posteriormente el sensor responde de la misma manera. Una imagen del sensor embebido puede ser vista en 2.2. Para más información, revisar [43].

Comunicación SPI

La comunicación SPI o Interfaz Periférica Serial es uno de los protocolos de comunicación más utilizados en sensores con periféricos integrados, debido a la alta frecuencia a la que pueden trabajar y su facilidad de uso. La lógica de la comunicación SPI es muy simple de entender, ya que solo utiliza de 3 a 4 cables, comúnmente 4. Se necesita de un sensor, el cual se suele llamar componente esclavo o subnodo, y un maestro o nodo principal, el

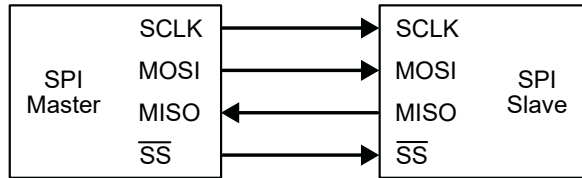


Figura 2.3: Diagrama de comunicacion SPI entre un componente maestro y uno esclavo.

cual suele ser un microcontrolador. Entre estos 2 componentes solo existe comunicación a través de 4 cables, los cuales se detallan en la tabla 2.1 y se muestran de manera gráfica en la figura 2.3. Para más información sobre el protocolo de comunicación SPI, consultar [12].

PIN	Significado
Reloj SPI o SCLK	Este cable envía una serie de pulsos desde el microcontrolador al sensor, para marcar tiempos, es decir, si el sensor ocupa enviar un paquete de un byte, al ser 4 bits, necesitan pasar 4 periodos del reloj, para marcar un byte.
Selección de Chip o \overline{SS}	Este unicamente se encarga de informar al sensor si se va a utilizar o no. Comúnmente mientras se utiliza se mantiene un 0 lógico y si se desea dejar de utilizar es necesario enviar un 1 lógico.
Salida de maestro, entrada esclavo (MOSI)	Transmite información del microcontrolador al sensor, todas las peticiones salen desde este cable.
Entrada de maestro, salida esclavo (MISO)	Recibe la información proporcionada por el sensor. Todas las peticiones son respondidas por el sensor a través de este cable.

Tabla 2.1: Porcentaje de ciclo de trabajo PWM aplicado como entrada escalón en los 10 experimentos para la identificación del sistema.



Figura 2.4: Motor típico de DC.

2.2.2. Actuadores

Un actuador convierte un tipo de energía en una acción de una máquina o dispositivo, comúnmente mecánica, los cuales se dividen en actuadores rotatorios o lineales, provocando un cambio de posición. Un actuador rotatorio es aquel que produce un movimiento angular y el actuador lineal es aquel que puede moverse en una línea recta. Existen actuadores eléctricos como motores, neumáticos e hidráulicos, como pistones, y otros más [6, 29, 33].

Motor de corriente directa

Son unos de los actuadores más utilizados actualmente en sistemas mecatrónicos y sistemas de control, ya sea en su presentación simple, o en versiones acondicionadas como los servomotores, motores a pasos, etc. Los motores de CD típicos constan de dos partes importantes, un estátor, que es la parte estática que produce el magnetismo para mover el rotor, que es la parte móvil del motor que proporciona la fuerza mecánica. De las características más importantes de un motor de CD es la proporcionalidad del voltaje de alimentación del motor, con la velocidad angular producida sobre el eje, al igual que existe una proporcionalidad entre la corriente que pasa por el devanado interno del motor con el par del motor. Para más información sobre actuadores eléctricos, consulte [33, 6]. Una imagen de un motor real típico de DC se muestra en 2.4.

2.2.3. NI MyRIO1900

Es una tarjeta de adquisición de datos con entradas y salidas reconfigurables, fabricada específicamente para que estudiantes e investigadores puedan practicar y/o simular



Figura 2.5: Imagen de una MyRIO real sin conexiones.

resolver problemas de ingeniería real. La tarjeta contiene un procesador principal para operación en “Tiempo real” y un arreglo de compuertas de campo programable, mejor conocido como FPGA. Entre los terminales que se disponen en la tarjeta se encuentran: 40 entradas y salidas digitales reconfigurables, de las cuales se pueden conectar a 4 encoders, 8 puertos PWM, 2 secciones de comunicación SPI, 2 secciones de comunicación I²C, 6 salidas analógicas, 10 entradas analógicas, 1 puerto USB tipo A, 1 puerto USB tipo B, 1 puerto de alimentación, 1 push button, 1 Aux de entrada y uno de salida, 4 LEDs. Además, cuenta con puertos de salida de 15v, 5v y 3.3v, todas estas características y otras más, mencionadas en [23, 28], hacen de esta tarjeta un instrumento apropiado para la investigación y prototipado de robots y sistemas mecatrónicos. En la figura 2.5 se muestra una imagen de una tarjeta MyRIO real y en la figura 2.6 se muestra un diagrama del hardware de la tarjeta, obtenido de [28] .

2.3. Modelo matemático

Un modelo matemático es una representación en forma de ecuaciones, que puede describir el comportamiento de un sistema físico [11]. El modelo matemático de un sistema se puede obtener al analizar las ecuaciones matemáticas que describen las leyes físicas, dinámicas y otros comportamientos que rigen a cada uno de los subsistemas por separado

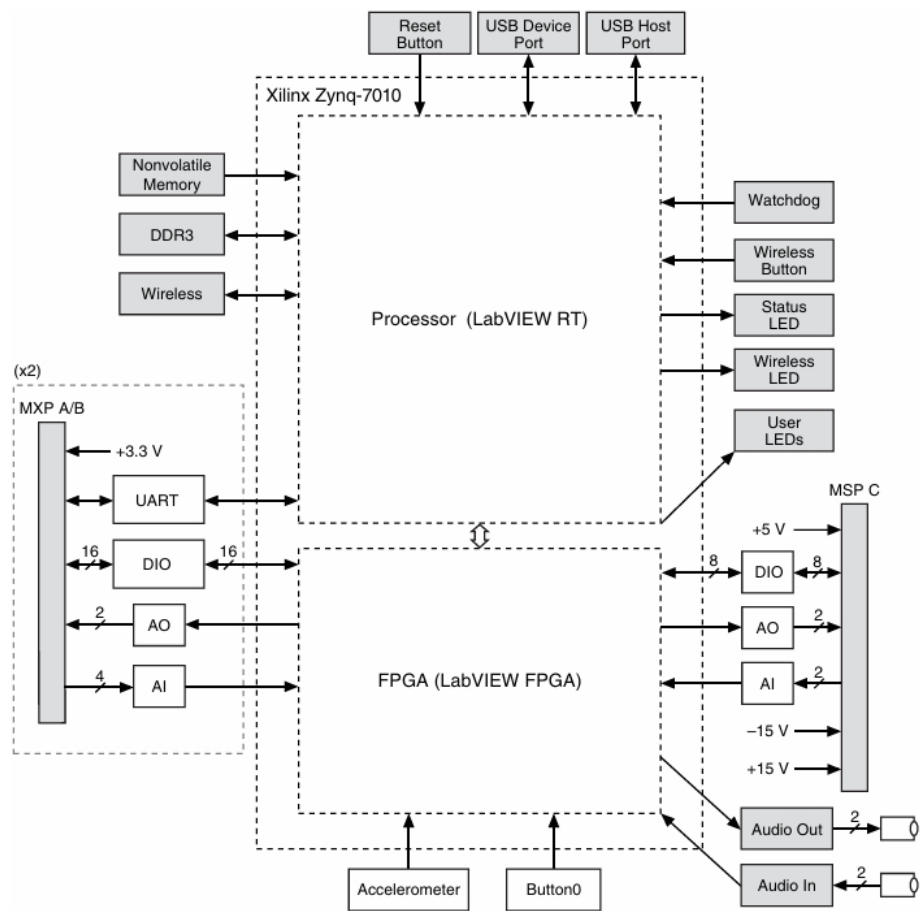


Figura 2.6: Hardware de la tarjeta NI-MyRIO-1900 obtenido de [28].

y sumándolos al final [18]. Es importante mencionar que un modelo matemático es solo una aproximación del comportamiento del sistema, ya que, en estos se asume conocer a la perfección todos los parámetros, no linealidades y limitantes del sistema, así como las perturbaciones desconocidas que afectan al sistema. Se puede obtener un modelo tan simple o complejo, o tan exacto según los requisitos que establezcan una utilidad del modelo.

2.3.1. Identificación de sistemas

En ocasiones el modelado de un sistema con base en ecuaciones físicas no describe de la manera más adecuada a un sistema, debido a no idealidades presentadas en componentes del sistema, como desgastes mecánicos, irregularidades en componentes, entre otros, los cuales pueden provocar que el modelado no sea suficientemente útil para algunos casos prácticos [29]. En estos casos se puede realizar el método de identificación de sistemas con base en experimentos, donde se excita al sistema con señales de entrada conocidas y se analizan las señales de salida producidas ante esas entradas. Al igual que el modelado con base en leyes físicas, mientras más información se tenga, es decir, más experimentos con diferentes amplitudes, frecuencias y formas de onda, se puede obtener un modelo más certero. Un método efectivo y relativamente sencillo es utilizando software de apoyo como Matlab, con ayuda de funciones *iddata*, *merge*, *tfest* y *lsim* [24, 31].

Matlab para la identificación de sistemas

Matlab es un software de apoyo matemático utilizado ampliamente en la comunidad ingeniera, para resolver problemas matemáticos, crear análisis estadísticos, realizar simulaciones de sistemas reales, entre otros. Este software cuenta con una amplia gama de funciones diseñadas específicamente para la identificación y modelado de sistemas reales, llamado *System Identification Toolbox*. Para hacer un modelado basado en experimentos, como el mismo nombre lo dice, necesita de experimentos realizados con el sistema que se desea identificar, los datos necesarios para el modelado son las señales de entrada aplicadas, las señales de salida producidas y el tiempo en el cual se realizó el experimento. Estos experimentos pueden ser guardados en diferentes formatos como *.dat*, *.mat* o *.csv* y

cargados a Matlab con la función *load*, la cual permite cargar archivos en forma de matrices de información, de la cual se pueden extraer los vectores de entrada, salida y tiempo de los experimentos. Otra función necesaria es la de *iddata*, la cual crea un objeto con la información conjunta de entrada, salida y tiempo de un experimento, para poder ser comprendida por otras funciones del *system identification toolbox* [24]. Si bien es posible obtener un modelo con un solo experimento, como se mencionó anteriormente, al realizarse más experimentos con más información se puede conseguir un modelo más certero. Para esto se utiliza la función *merge*, la cual almacena varios objetos *iddata* en uno solo [24]. Estos objetos *iddata* únicamente almacenan la información de varios experimentos, la cual puede ser utilizada para la obtención de un modelo, como una función de transferencia. Esto se puede hacer con el comando *tfest*, el cual proporciona una función de transferencia aproximada del sistema en base al objeto *iddata* ingresado y la cantidad de polos y ceros estimados del sistema [31]. Si bien el proceso de modelado puede terminar al obtener un modelo matemático del sistema, es apropiado validar el modelo, comparándolo con experimentos reales, hasta obtener un modelo suficientemente útil como se requiera, para esto se utiliza la función *lsim*, donde se ingresa la función de transferencia que se desea simular, una señal de entrada u , un vector de tiempo t y si es necesario condiciones iniciales x_0 .

2.3.2. Sistemas de 1er orden

En el análisis de sistemas físicos lineales, los modelos matemáticos se suelen clasificar de acuerdo al orden de su ecuación diferencial representante, en el cual, se toma en consideración dos etapas importantes, que son la respuesta transitoria, que representa la etapa entre el estado inicial del sistema hasta antes de estabilizarse, la segunda etapa es la respuesta en estado estable, en la cual la respuesta transitoria se considera despreciable o que el efecto de la entrada aplicada ha sido absorbido completamente [15]. Si bien al modelar matemáticamente se define la exactitud y simplicidad del modelo matemático según qué tan completo este [29], en el mundo práctico se puede definir a una gran cantidad de sistemas, como sistemas predominantemente de primer orden. Matemáticamente hablando,

los sistemas de primer orden se representan con una ecuación diferencial de primer orden

$$\dot{y}(t) + ay(t) = bu(t), \quad (2.1)$$

$$y(0) = y_0,$$

donde $y(t)$ representa la salida del sistema, a y b son constantes reales, y_0 es la condición inicial para $y(t)$ y $u(t)$ es la entrada conocida del sistema. Para encontrar la función $y(t)$ de la ecuación 2.1, se aplica la transformada de Laplace a una ecuación 2.1 y se obtiene que:

$$Y(s) = \frac{bU(s)}{s+a} + \frac{y_0}{s+a} \quad (2.2)$$

Considerando una señal de entrada tipo escalón para el sistema se puede decir que

$$u(t) = A \quad (2.3)$$

donde A es una constante, por lo tanto, la transformada de Laplace de la entrada definida en la ecuación 2.3 corresponde

$$U(s) = \frac{A}{s} \quad (2.4)$$

Si se sustituye la ecuación 2.4 en 2.2 podemos decir entonces que:

$$Y(s) = \frac{Ab}{s(s+a)} + \frac{y_0}{s+a} \quad (2.5)$$

Debido a la forma que tiene la ecuación 2.5 se puede aplicar el método de fracciones parciales a la primera fracción, la cual se expresaría como:

$$\frac{Ab}{s(s+a)} = \frac{k_1}{s} + \frac{k_2}{s+a} \quad (2.6)$$

Al despejar las constantes k_1 y k_2 se obtiene:

$$k_1 = \frac{Ab}{a}, \quad k_2 = -\frac{Ab}{a} \quad (2.7)$$

y sustituyendo 2.7 en 2.6 y después en 2.5 se dice que:

$$Y(s) = \frac{Ab}{a} \left(\frac{1}{s} - \frac{1}{s+a} \right) + \frac{y_0}{s+a} \quad (2.8)$$

Utilizando la transformada de Laplace inversa, se pueden pasar los términos de la ecuación 2.8 en dominio del tiempo y obteniendo así, la respuesta del sistema ante una entrada escalón y se expresaría de la siguiente forma:

$$y(t) = \frac{Ab}{a} (1 - e^{-at}) + y_0 e^{-at} \quad (2.9)$$

Para más información sobre la solución de la ecuación diferencial de primer orden consulte [18]. Una de las características principales de los sistemas de primer orden es que estos no presentan sobre elongaciones, ni oscilaciones en su comportamiento. Esto se puede observar al determinar el valor final de la ecuación 2.9, suponiendo una $y_0 = 0$

$$\lim_{t \rightarrow \infty} y(t) = \frac{Ab}{a} \quad (2.10)$$

De esta manera se logra observar que la respuesta del sistema tiende a $\frac{Ab}{a}$, siempre y cuando el sistema sea estable (Se considera que un sistema de primer orden es estable si y solo si $a > 0$). Otra forma de representar un sistema es mediante variables de estado. Donde el estado dinámico de un sistema se representa por un conjunto de variables de estado, que caracterizan la dinámica del sistema. El estado de un sistema dinámico representa el conjunto más pequeño de variables de estado $x(t)$, de manera que al conocer esas variables en un instante de tiempo $t = t_0$, junto con la entrada $u(t)$ en todo instante de tiempo $t \geq t_0$ y las ecuaciones que describen la dinámica del sistema como lo son la ecuación de estado $\dot{x}(t) = f(x, t, u)$ y la ecuación de salida $y(t) = g(x, t, u)$. Es posible determinar el comportamiento de los estados $x(t)$ y salidas $y(t)$ en todo instante de tiempo $t \geq t_0$. Considerando la ecuación 2.1 y definiendo $x = y$ como variable de estado, el sistema se representaría de la manera siguiente:

$$\dot{x}(t) = -ax(t) + bu(t) \quad (2.11)$$

donde, para el caso simple de sistemas de primer orden, la salida se define como:

$$y(t) = x(t) \tag{2.12}$$

Por lo tanto, el modelo en forma matricial es:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{2.13}$$

donde $A = [-a]$, $B = [b]$, $C = [1]$, $D = [0]$. Para más información consulte [15]. Este tipo de modelo matemático resulta particularmente útil para el diseño de controladores modernos y para la simulación computacional de sistemas dinámicos.

2.3.3. Sistemas de control en lazo abierto y lazo cerrado

Los sistemas de control o controles automáticos se pueden definir como un arreglo de componentes físicos conectados de tal manera, que el arreglo pueda comandar, dirigir o regular sí mismo o a otro sistema [15], y es usado ampliamente en industrias para controlar procesos, regulando variables dentro de los diferentes sistemas de la misma industria. Para ser llamado sistema de control este debe contar con varios elementos fundamentales como lo son: El proceso a controlar, la variable controlada o de referencia, la variable manipulada o de salida, la perturbación, un elemento de medición, un controlador, elemento final de control, una entrada y una salida (en algunos sistemas complejos se puede tener uno o más de alguno de los elementos mencionados). Los sistemas de control pueden dividirse en dos tipos diferentes: Un sistema de control en lazo abierto, donde la señal de salida del sistema depende únicamente de la señal de entrada aplicada al sistema y los elementos de control aplicados al sistema (condiciones para que el sistema se active o desactive). Por otro lado un sistema de control en lazo cerrado es diferente al lazo abierto, debido a que la señal de salida tiene un efecto directo sobre los elementos de control [9]. Una representación gráfica es mostrada en la gura (1), en donde se aprecia que el usuario especifica una señal de referencia y esta pasa por una etapa de comparación entre la señal de referencia y la

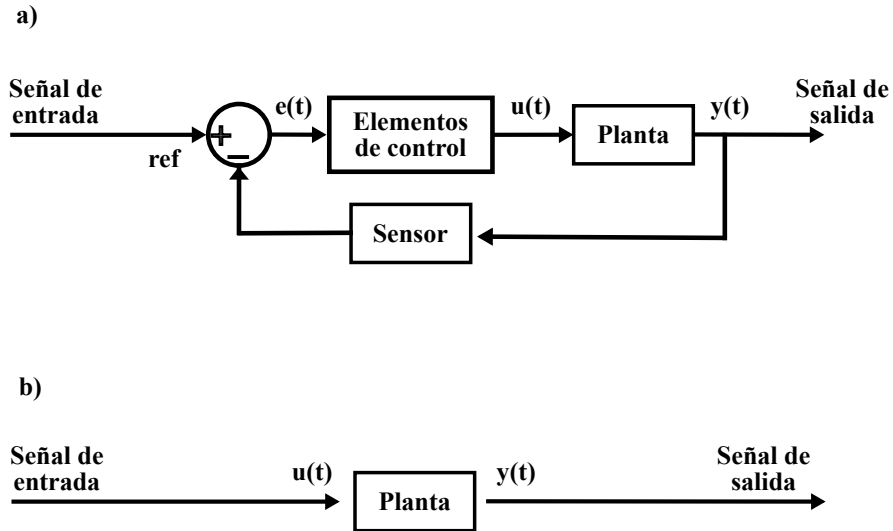


Figura 2.7: a) Diagrama de un sistema de control en lazo cerrado. b) Diagrama de un sistema de control en lazo abierto.

señal de salida medida por un objeto sensor, la señal resultante se conoce como error y esta pasa por una etapa de control (según los objetivos de control se tendrán diferentes elementos de control), donde se adecúa la señal para luego ingresar al sistema y produzca una señal de salida idealmente convergente a la señal de salida. Los sistemas de lazo cerrado suelen ser más precisos debido a que la corrección es realizada por una retroalimentación directa, pero suelen ser más complejos, costosos, difíciles de mantener y debido a la mayor cantidad de componentes, se suelen descomponer con mayor frecuencia [6] Los beneficios de los sistemas en lazo abierto es que suelen ser menos costosos, más sencillos de implementar, pero debido a que no existe una corrección directa para el comportamiento de salida del sistema, suelen ser más imprecisos [9]. Estos 2 suelen ser explicados gráficamente con el diagrama mostrado en la figura 2.7.

2.4. Objetivos de control

Los objetivos de un sistema de control ideal son: Garantizar la estabilidad ante perturbaciones y errores del modelo, ser lo más eficiente posible según el problema establecido y

ser fácilmente manipulable en tiempo real [17]. Para cumplir con esto hay algunos conceptos clave que se deben comprender claramente antes de establecer los objetivos de control, estos son: Exactitud, Velocidad de respuesta, sobreelongación o sobrepaso y estabilidad. El primero es la exactitud, es decir, que puede mantener la señal de salida del sistema controlado en un valor exactamente igual o suficientemente aproximado a la señal de referencia especificada. Un sistema es más exacto mientras menor sea el error producido al comparar la señal de referencia entre la señal de salida real del sistema [15]. Otro objetivo de control es la velocidad de respuesta, es decir, es el tiempo que toma un sistema desde que es detectado un cambio en la entrada del sistema hasta que su salida obtenga una etapa en estado estable. Idealmente, esta etapa en estado estable debe alcanzar el valor de referencia; en caso de que la salida no converja al valor de referencia se dice que se alcanzó un error en estado estable [15]. La sobreelongación, o también llamada como máximo sobreimpulso o sobrepaso máximo, es el momento de valor máximo alcanzado, diferente del valor de referencia inducido [9]. Idealmente, un sistema de control debe tener un sobrepaso máximo nulo, pero debido a que las condiciones ideales son imposibles de conseguir, se suele establecer un punto máximo permitido como límite. Por último, está la estabilidad. La estabilidad en un sistema es la habilidad de mantener la señal de salida de respuesta con un comportamiento similar o idealmente convergente al de la señal de entrada. Un sistema es inestable cuando el comportamiento en su señal de salida es más alejado del comportamiento de la señal de entrada, es decir, no converge en ningún momento del tiempo [15].

2.5. Controlador regulador y seguidor de trayectorias

En teoría de control, un sistema regulador es aquel cuyo propósito es el de mantener una variable controlada en un valor constante [15]. Se utiliza un sistema de control en lazo cerrado en donde se pueda ingresar un valor de referencia, los elementos de control actúan para corregir el comportamiento de salida del sistema hasta que este alcance el valor de referencia especificado. Para que el sistema alcance este valor, pasa por dos etapas, una etapa transitoria, en donde los elementos de control están realizando correcciones mayores

a la entrada del sistema para que este alcance el valor deseado y la etapa en estado estable, donde el sistema asienta la señal de salida, idealmente sobre el valor de referencia. Este comportamiento se puede apreciar gráficamente en la figura 2.8. A diferencia del control regulador, el control seguidor de trayectorias se refiere al control aplicado a un sistema cuyo objetivo o señal de referencia está en constante cambio [15] . De la misma manera que el control regulador, se utiliza la metodología de sistemas de control en lazo cerrado, donde los elementos de control cumplen con el objetivo de disminuir la señal de error, es decir, disminuyen la diferencia entre la señal del sistema real y en este caso ya no es un valor de referencia jo, sino una señal de referencia variante (esta puede tener ya sea una forma típica como rampa, sinusoidal, cuadrada, triangular o una forma atípica). Al igual que en el sistema regulador, existen dos etapas en la señal de control seguidora de trayectorias, la etapa transitoria y la etapa en estado estable. Aunque la señal puede lucir diferente a la de un control regulador, el principio es el mismo. La etapa transitoria es el lapso de tiempo en que la señal de salida del sistema trata de converger a la señal de salida del sistema y la etapa en estado estable es el tiempo en el que a pesar de que la señal de referencia sigue variando, la señal de salida varía de la misma manera. El comportamiento de control seguidor de trayectorias se puede apreciar gráficamente en la figura 2.9.

2.6. Acciones de control

2.6.1. Acción Proporcional

El controlador P o proporcional, es esencialmente un amplificador de ganancia donde su salida es igual al error multiplicado por una ganancia k_p . Es decir:

$$u(t) = k_p e(t) \tag{2.14}$$

Este elemento de control se puede expresar mediante el concepto de banda proporcional el cual se define como el porcentaje del inverso de la ganancia k_p :

$$BP = \frac{100}{k_p} \tag{2.15}$$

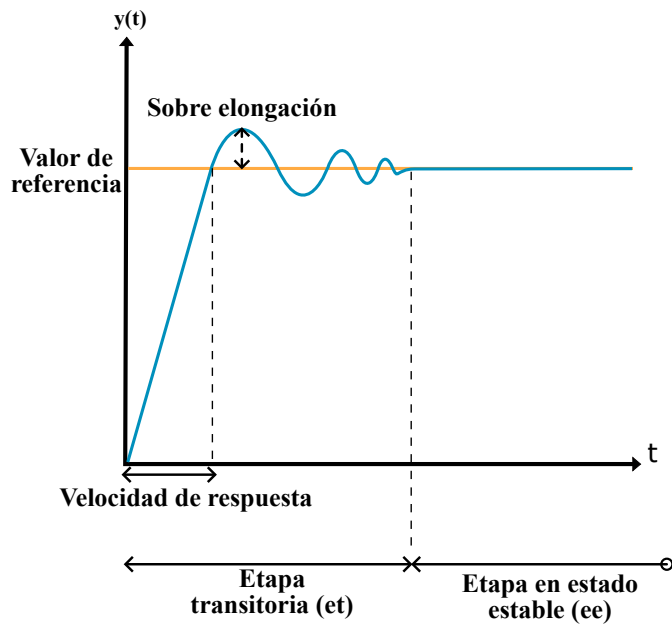


Figura 2.8: Comportamiento típico de un controlador regulador.

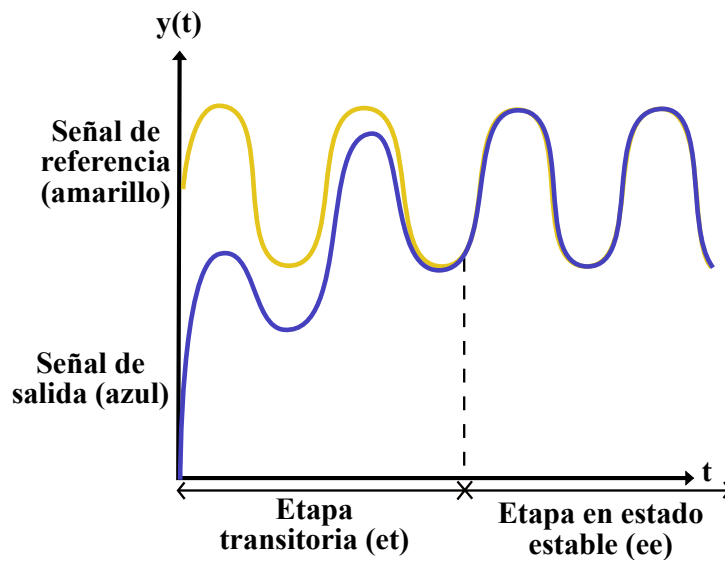


Figura 2.9: Comportamiento típico de un controlador seguidor de trayectorias.

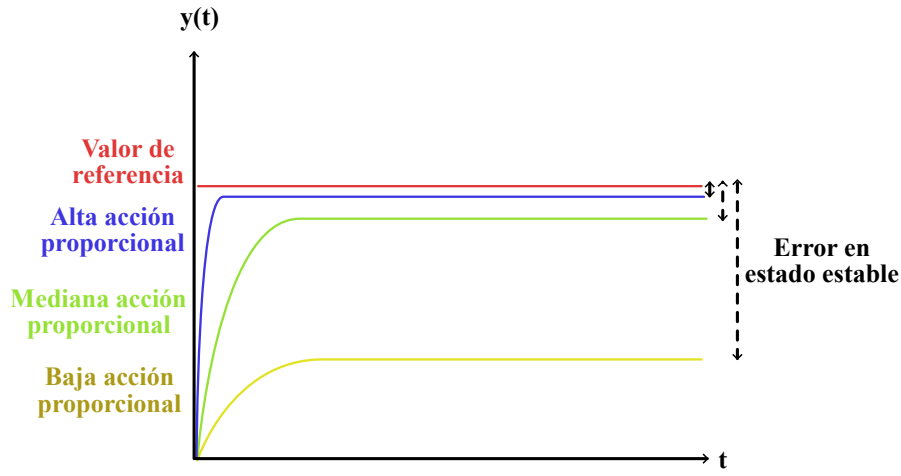


Figura 2.10: Influencia de un controlador proporcional sobre un sistema ideal con respuesta predominantemente de 1er orden.

donde al tener una ganancia muy alta se tendrá una banda proporcional que tienda a cero, por lo que el sistema actuará velozmente para corregir el error entre la salida del sistema; se debe tener cuidado, ya que, en un sistema real, el ruido o perturbaciones no consideradas también podrían mantener al sistema corrigiéndose en todo momento a muy altas velocidades y con entradas muy altas, lo cual podría dañar al sistema. En cambio, una ganancia k_p demasiado baja produciría una banda proporcional muy grande, lo cual produciría un error en estado estable o incluso no generar ninguna acción de corrección manteniendo un error absoluto del sistema [15]. Nótese que utilizar una ganancia adecuada en algunos sistemas no oscilatorios podría conseguir una corrección óptima, obteniendo así una etapa en estado estable con error mínimo, tal como se muestra en la figura 2.10.

2.6.2. Acción Integral

El controlador tipo I o integral genera una salida proporcional a la integral del error, es decir, que la tasa de cambio de la salida de control $u(t)$, es proporcional a la señal de

error:

$$\frac{du(t)}{dt} = k_i e(t) \quad (2.16)$$

Para despejar la variable $u(t)$, se integra la ecuación anterior tal que:

$$u(t) - u(0) = k_i \int_0^t e(t) dt \quad (2.17)$$

Suponiendo condiciones iniciales del sistema en cero se puede decir que:

$$u(t) = k_i \int_0^t e(t) dt \quad (2.18)$$

Para más información, se puede consultar [6]. Ya que en este controlador se toma en cuenta la tasa de cambio de la salida del sistema, este no permite que exista un error en estado estable (siempre y cuando exista una etapa en estado estable). Además, la integral aplicada al sistema aumenta el orden del mismo. Esto pasa debido a que se tiende a sobre corregir el error y a causar oscilaciones al sistema. Nótese que, debido a su naturaleza, la velocidad de cambio producida es muy lenta, por lo cual no se utiliza solo [15], el comportamiento típico de esta acción es mostrado en la figura 2.11.

2.6.3. Acción Proporcional-Integral

Es la suma del controlador Proporcional con el controlador Integral y se expresa matemáticamente de la siguiente forma:

$$u(t) = k_p e(t) + \frac{1}{T_i} \int_0^t e(t) dt \quad (2.19)$$

donde T_i es el tiempo de acción integral y es igual a:

$$T_i = \frac{k_p}{k_i} \quad (2.20)$$

Esta expresión se puede definir como el tiempo necesario para que la respuesta integral sea igual a la proporcional después de un cambio producido en el error [15]. Este controlador

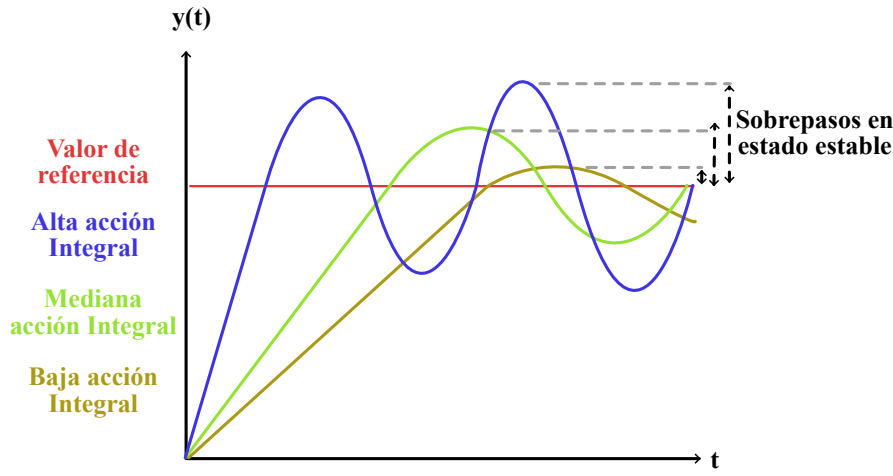


Figura 2.11: Influencia de un controlador integral sobre un sistema ideal con respuesta predominantemente de 1er orden.

utiliza las características de ambas acciones de control, por lo que la ganancia proporcional puede obtener una respuesta rápida y la ganancia integral se encarga de eliminar el error en estado estable. Debido a la naturaleza del controlador integral, al aumentar el orden del sistema, este puede agregar oscilaciones, lo que puede ocasionar que el sistema oscile permanentemente alrededor del valor de referencia.

2.6.4. Controlador por modos deslizantes

Se le conoce como modos deslizantes a un fenómeno observado en sistemas físicos con comportamientos discontinuos, donde un estado del sistema $x(t)$ evoluciona hasta alcanzar una superficie σ sobre la cual se “desliza” y permanece en “deslizamiento”, incluso en presencia de perturbaciones acotadas $f(x_1, x_2, t)$, donde se incluyen perturbaciones externas y dinámicas no consideradas en el modelo del sistema. El control por modos deslizantes es una estrategia de control que recrea este comportamiento de manera artificial, con el objetivo de conseguir una convergencia en tiempo finito y robustez, es decir reducción o incluso inmunidad ante perturbaciones. Para recrear este comportamiento es necesario proponer una variable de deslizamiento $\sigma(x_1, x_2)$ en los espacios de estado del sistema,

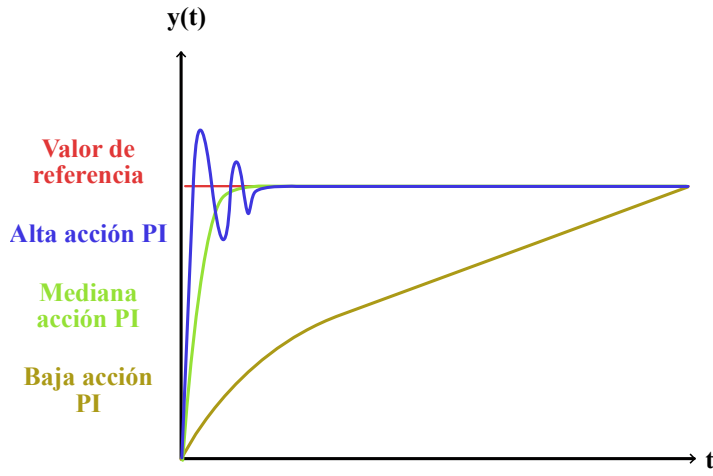


Figura 2.12: Influencia de un controlador proporcional-integral sobre un sistema ideal con respuesta predominantemente de 1er orden.

de tal manera que las variables de estado x_1, x_2 converja a 0 en un tiempo finito t_r y se mantienen en la superficie, incluso en presencia de las perturbaciones acotadas $f(x_1, x_2, t)$, es decir, las incertidumbres paramétricas, dinámicas no modeladas en el sistema y perturbaciones externas, a través de la variable de control u , la cual se puede expresar en término generales como:

$$\text{si } \sigma(x) > 0 \rightarrow u = u^+ \rightarrow \dot{x} = f(x, u^+) = f^+ \quad (2.21)$$

$$\text{si } \sigma(x) < 0 \rightarrow u = u^- \rightarrow \dot{x} = f(x, u^-) = f^- \quad (2.22)$$

La propuesta típica de control por modos deslizantes se define de la siguiente manera:

$$u = k * \text{sign}(\sigma)$$

en donde la función signo establece que:

$$\text{sign}(x) = \begin{cases} 1; & \text{si } x > 0 \\ 0; & \text{si } x = 0 \\ -1; & \text{si } x < 0 \end{cases}$$

Esta estrategia de control se le conoce como controlador por modos deslizantes o SMC por sus siglas en inglés. Un sistema con modos deslizantes, presenta dos etapas importantes conocidos como fase de alcance, donde la trayectoria de los estados evoluciona hasta alcanzar la superficie de deslizamiento σ y el otro comportamiento conocido como fase de deslizamiento, donde la trayectoria de los estados se mantiene en la superficie de deslizamiento hasta convertir las variables de estado x_1, x_2 en 0. Un comportamiento muy importante que aparece con el fenómeno de modos deslizantes es el “chattering”, donde los estados del sistema oscilan a muy altas frecuencias con cierta amplitud alrededor de la superficie deslizante. El comportamiento ideal del modo deslizante es aquel donde las oscilaciones del sistema tienen una frecuencia infinita y la amplitud alrededor de la superficie deslizante tiende a 0. Existen dos métodos utilizados para comprender la dinámica del sistema, que son la solución de Filippov para ecuaciones diferenciales con lado derecho discontinuo y el método de control equivalente, el cual no representa directamente el control aplicado al sistema, sino la función de control necesaria para asegurar que las trayectorias del sistema permanezcan en la superficie de deslizamiento una vez alcanzada, es decir, se encarga de compensar las perturbaciones $f(x_1, x_2, t)$. Los comportamientos típicos de este controlador para resolver problemas de regulación y seguimiento de trayectorias, pueden observarse de manera gráfica en las figuras 2.13 y 2.14, respectivamente. Para más información sobre controladores por modos deslizantes y el método de implementación, consulte [4, 26, 38, 40].

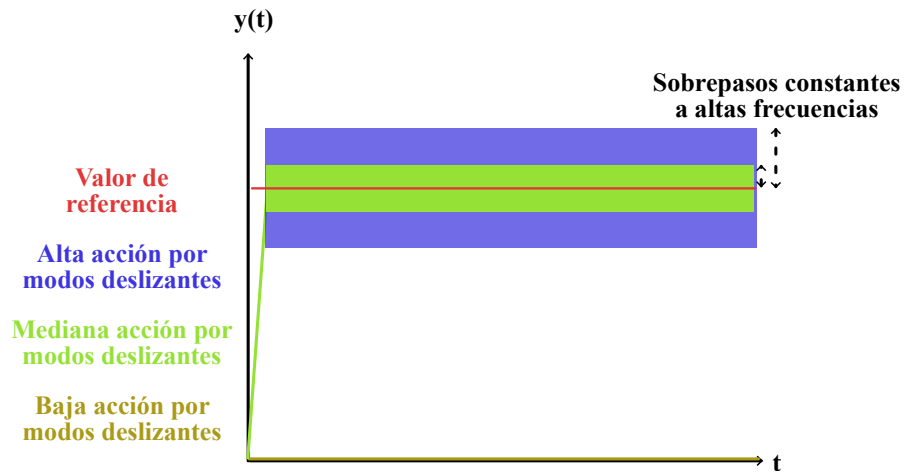


Figura 2.13: Influencia de un controlador regulador con modos deslizantes sobre un sistema ideal con respuesta predominantemente de 1er orden.

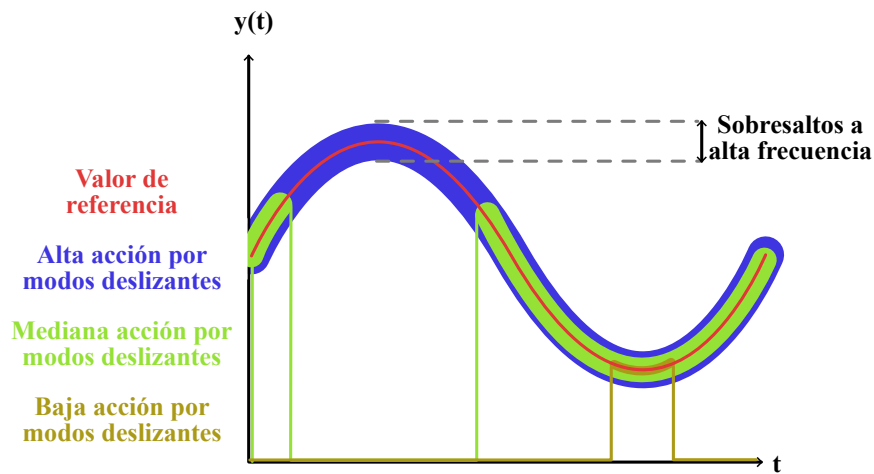


Figura 2.14: Influencia de un controlador seguidor de trayectoria con modos deslizantes sobre un sistema ideal con respuesta predominantemente de 1er orden.

2.7. Sincronización de sistemas dinámicos

Se define como sincronización al comportamiento de dos o más sistemas que concuerdan en el tiempo [21]. Este es un comportamiento que puede ser observado en la naturaleza, como por ejemplo en las llamadas de apareamiento sincronizadas en algunas especies de ranas, que evolucionaron la manera de emitir estos llamados para evitar ser localizados por depredadores [22]. A estos sistemas sincronizados de manera natural se les conoce como sincronización natural; en ocasiones donde la sincronización no ocurre de manera natural, es posible introducir una dinámica de interconexión entre los sistemas. En [20] se hace una definición general para k sistemas dinámicos, $\sum_i = \{T, U_i, X_i, Y_i, \phi_i, h_i\}$, donde T representa todos los posibles instantes de tiempo. De la misma manera, U_i, X_i, Y_i representan el conjunto de entradas, estados y salidas posibles, respectivamente. Por otro lado, $\phi_i : T \times X_i \times U_i \rightarrow X_i$ representan cómo evoluciona el estado X_i dado el tiempo, la entrada y el estado actual; a este se le denomina mapa de transición. De la misma manera, $h_i : T \times X_i \times U_i \rightarrow Y_i$ representa cómo evoluciona la salida Y_i dado el tiempo, la entrada y el estado actual; a este se le denomina mapa de salida. Posteriormente se definen l funcionales $g_j : \mathcal{Y}_1 \times \mathcal{Y}_2 \times \dots \times \mathcal{Y}_k \times T \rightarrow \mathbb{R}^l$, $j = 1, \dots, l$. donde \mathcal{Y} representa todo el conjunto de posibles respuestas dado T en Y_i , es decir, todas las posibles curvas en un lapso de tiempo, ante cualquier entrada. También se establece que para $\tau \in T$ se define un operador de cambio σ_τ , es decir, que $\sigma_\tau : \mathcal{Y} \rightarrow \mathcal{Y}_i$ es dado como $(\sigma_\tau y)(t) = y(t + \tau)$ para toda $y \in \mathcal{Y}$ y para toda $t \in T$. Es posible que existan sistemas sincronizados, aproximadamente sincronizados o sintéticamente sincronizados. Estos se definen de la siguiente manera.

$$g_j(\sigma_{\tau_1} y_1(\cdot), \dots, \sigma_{\tau_k} y_k(\cdot), t) \equiv 0, \quad j = 1, \dots, l \quad (2.23)$$

Es decir, si las condiciones de sincronización se cumplen para todo $t \in T$ los sistemas se denominan como sincronizados. Para el caso donde:

$$|g_j(\sigma_{\tau_1} y_1(\cdot), \dots, \sigma_{\tau_k} y_k(\cdot), t)| \leq \epsilon, \quad j = 1, \dots, l \quad (2.24)$$

siendo $\epsilon > 0$, los sistemas se denominan aproximadamente sincronizados, ya que están “sincronizados” bajo un criterio establecido. Por último el caso donde:

$$\lim_{t \rightarrow \infty} g_j(\sigma_{\tau_1} y_1(\cdot), \dots, \sigma_{\tau_k} y_k(\cdot), t) = 0, \quad j = 1, \dots, l \quad (2.25)$$

para algunos $\tau_1, \dots, \tau_k \in T$, se denominan asintóticamente sincronizados, ya que las funcionales de sincronización tienden a cumplirse con el tiempo. Para más información consulte [20, 21, 35].

2.7.1. Sincronización Maestro-Esclavo

Una de las funcionales de sincronización más populares debido a su fácil implementación y buen desempeño de sincronización es el esquema de maestro-esclavo. Un esquema representativo se muestra en la figura 2.15, donde se logra observar que la señal de control del sistema denominado “esclavo” depende no solo de uno de sus estados, sino también de uno de los estados del sistema denominado “maestro”. Atendiendo la sincronización más simple como un problema de control de regulación, se puede definir un error de sincronización tal que:

$$e_{sincro} = y_s - y_m \quad (2.26)$$

de tal manera que el objetivo de control sea eliminar el error de sincronización. Nótese que es posible proponer diferentes esquemas de sincronización según los estados deseados a sincronizar y el tipo de sincronización deseada.

2.8. Robots móviles

Es un tipo de sistema electromecánico capaz de desplazarse, a través del uso de diferentes actuadores y puede ser autónomo o manipulado. En caso de ser autónomo, este debe ser capaz de conocer su posición relativa, respecto a un punto de origen establecido y el punto destino en el espacio a través del uso de sensores. Al ser un tipo de robot, posee características establecidas como la capacidad de actuar, sentir y pensar, en este caso, desplazarse, conocer el entorno y adaptarse al interactuar con el entorno. Debido a

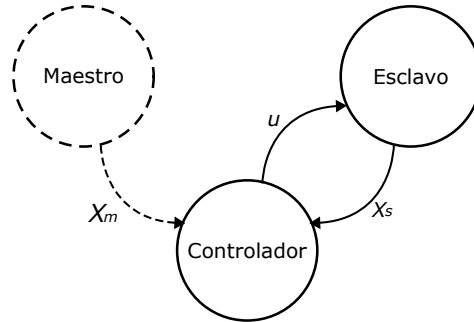


Figura 2.15: Diagrama del esquema de sincronización unidireccional o Maestro-Esclavo.

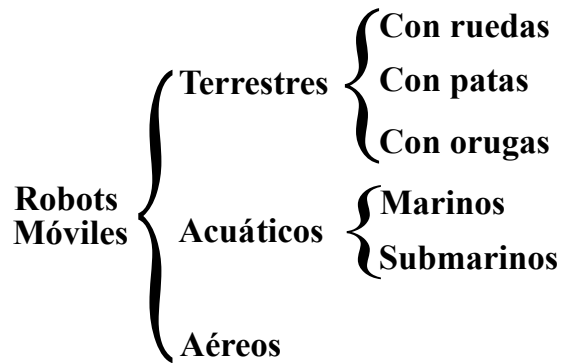


Figura 2.16: Diagrama de clasificación de robots móviles

la alta demanda de robots móviles con múltiples ventajas, a lo largo de los años, se han estructurado clasificaciones de los robots móviles, según el medio y/o el método a través del cual se desplazan[30]. Un diagrama de clasificación de robots móviles se muestra en la figura 2.16.

2.9. Modelado cinemático de sistemas mecánicos

El modelo cinemático de un sistema es aquel que relaciona los movimientos en las articulaciones de un sistema y el movimiento resultante de los cuerpos rígidos del robot o efector final sin considerar las fuerzas o torques que influyen sobre él. La cinemática estudia dos problemáticas principales, la cinemática directa, encargada de determinar el

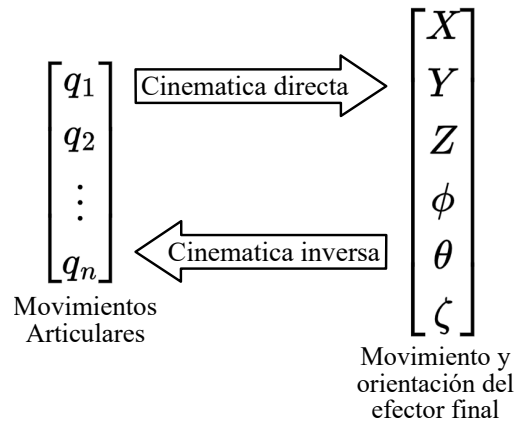


Figura 2.17: Diagrama general de la relación entre

movimiento de un cuerpo rígido del robot o efector final, a través del movimiento de sus articulaciones y la cinemática inversa, que calcula el movimiento de las articulaciones del robot según el cuerpo rígido o efector final del robot [30, 37]. Las ecuaciones de cinemática son modelos que pueden ayudar a describir el movimiento de una gran variedad de sistemas mecánicos, pero es necesario tomar en cuenta que, si bien hay diferencias entre un sistema y otro, el proceso de modelado cinemático de sistemas mecánicos seriales, como en un brazo manipulador y en sistemas mecánicos paralelos como en un sistema de coordenadas X , Y , Z , es completamente diferente. A pesar de eso, estas relaciones pueden ser representadas en un diagrama de la manera más general en la figura 2.17. donde q_i son los movimientos articulares del sistema y $X, Y, Z, \phi, \theta, \zeta$ son los movimientos sobre los ejes X , Y , Z y las orientaciones sobre los ejes Z , Y y X , respectivamente, del efector final.

Capítulo 3

Diseño y construcción de un robot móvil omnidireccional con ruedas Mecanum

El primer vehículo auto-propulsado registrado en la historia fue creado por Nicolas Joseph Cugnot, en 1769. El cual era un pequeño cuerpo de madera con 3 ruedas, que no tenía ni siquiera la capacidad de girar ni frenar. Pasaron más de 100 años, hasta que Karl Benz patentó el primer vehículo impulsado por un motor de combustión interna, para uso cotidiano y con la capacidad de girar [27]. A partir de ese momento, los automóviles ganaron gran popularidad a lo largo de Europa, evolucionaron y avanzaron tecnológicamente hasta ser el vehículo de transporte que conocemos hoy en día, donde ahora los vehículos se pueden controlar a distancia e incluso su operación puede ser autónoma. Por otro lado, existe toda una rama de robots móviles autónomos, donde se atienden dos problemas fundamentales de la programación, refiriéndose a la inteligencia, automatización y control, así como el problema de mejorar su capacidad motriz de superar obstáculos [25], con el fin de crear robots de servicio, para condiciones inaccesibles para humanos o simplemente para facilitar el trabajo, como trabajos de carga en la industria o en sitios de construcción, incluso en el espacio [3]. Una de las soluciones que hasta el día de hoy se sigue investigando, debido a sus propiedades atractivas en la movilidad, son los robots móviles con ruedas omnidireccionales, los cuales se estarán analizando en este documento bajo la abreviación de RMO [13]. Existen dos variantes altamente estudiadas de movimientos omnidireccionales, uno es conocido como tipo Stanford, el cual cuenta con rodillos perpendiculares al eje de

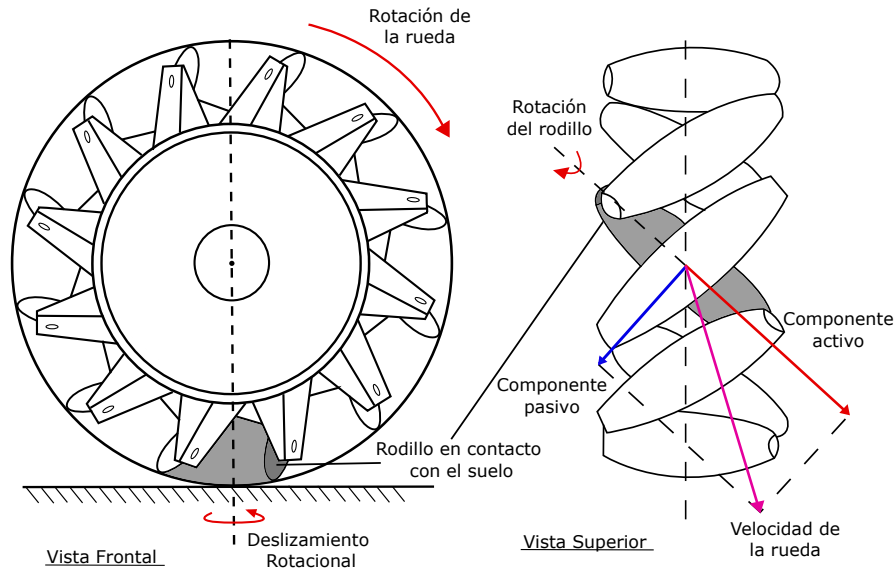


Figura 3.1: Diagrama de una rueda Mecanum con sus grados de libertad.

la rueda, el segundo son las ruedas conocidas como Mecanum, con rodillos en diagonal al eje de la rueda [16]. Ambos tipos de ruedas tienen la capacidad de ofrecer a un robot móvil mayor maniobrabilidad al permitirle generar movimientos no solo en direcciones delanteras y de reversa, sino en cualquier dirección del plano, sin la necesidad de cambiar la orientación inicial del vehículo y viceversa, es decir, cambiar la orientación del vehículo con respecto a su eje central sin la necesidad de alterar su ruta en el plano.

3.1. Ruedas Mecanum

En 1973 se inventaron las ruedas Mecanum por el ingeniero sueco Ilon Bengt, de la compañía Mecanum. Las cuales cuentan con un cuerpo de metal de manera cilíndrica al igual que cualquier otra rueda, la diferencia yace en el perímetro del círculo, el cual está cubierto de rodillos, para otorgar un grado de libertad que permita deslizar al vehículo en cualquier dirección. Estos rodillos están orientados a 45° del eje de la rueda, con el fin de agregar otro grado de libertad, el cual genera un vector de velocidad en el sentido del eje del rodillo como se muestra en la figura 3.1. Si se plantea el diseño de un robot móvil omnidireccional, es necesario el acomodo correcto de ruedas Mecanum, ya que una rueda

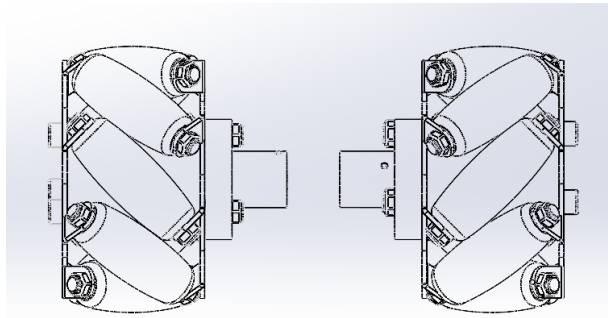


Figura 3.2: Vista superior de una rueda Mecanum zurda y una diestra.

solo otorga movimientos hacia una dirección de 45° con respecto al eje del vehículo, pero al agregar pares de ruedas diestras y zurdas, como las que se muestran en la figura 3.2, las cuales cuentan con los rodillos orientados a 45° y -45° con respecto al eje de la rueda, y de esta manera generan movimientos en todas las direcciones en el vehículo, sin alterar su orientación inicial. Esto se demuestra al hacer una simple sumatoria de vectores de velocidad de las ruedas sobre el eje central del vehículo.

3.1.1. Movimientos típicos de un robot móvil omnidireccional

Para visualizar los efectos de las ruedas Mecanum sobre un RMO, se puede utilizar la figura 3.3, en donde se logra apreciar el vector de velocidad producido en cada rueda. Al hacer una sumatoria de vectores sobre el eje del robot se encuentran los vectores en la componente x , que tienen la misma magnitud pero en sentido opuesto, y por lo tanto se cancelan, dejando únicamente las componentes en y , produciendo un movimiento hacia adelante. Utilizando la misma lógica es posible para un RMO, producir movimientos en direcciones de 0° , 45° , 90° , 135° , 180° , 225° , 270° y 315° , además de giros sobre el propio eje del robot, como se muestra en la figura 3.4. Sin embargo, los movimientos de un RMO no se limitan a esas direcciones, sino que con una combinación adecuada de velocidades angulares de cada rueda, es posible proporcionar un movimiento hacia cualquier dirección del plano (x, y) además de mantener la orientación inicial o incluso manipularla de manera independiente. Para esto se deben tomar en cuenta las ecuaciones de cinemática directa e inversa, mencionadas a continuación.

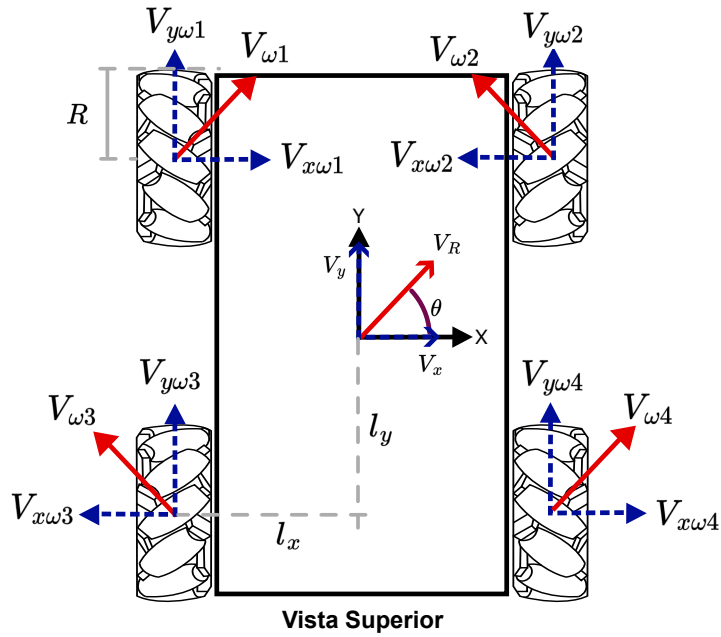


Figura 3.3: Vista superior de un RMO, con los respectivos vectores de velocidad generados en cada rueda.

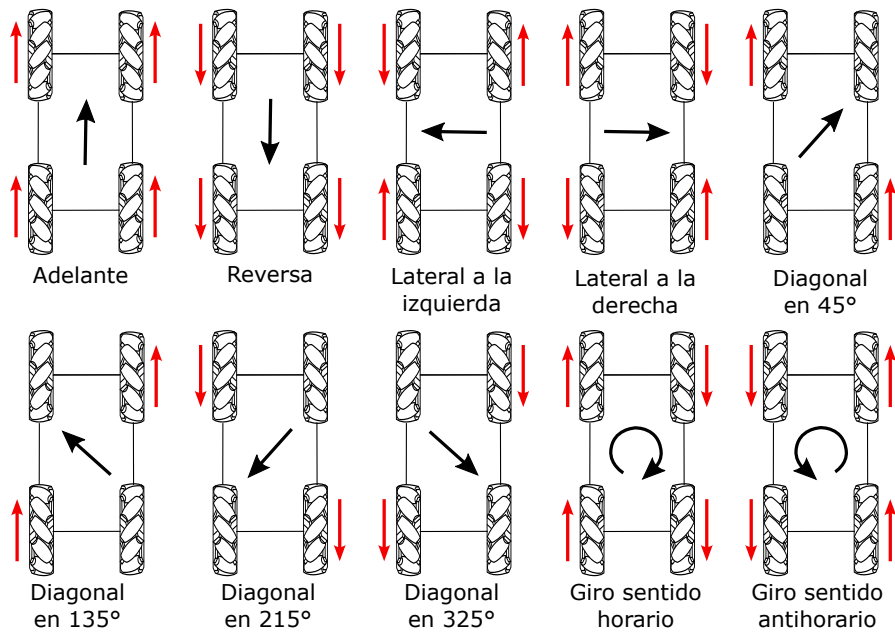


Figura 3.4: Movimientos típicos de un RMO de 4 ruedas con la condición de dirección de cada una de sus ruedas.

3.1.2. Modelo cinemático directo e inverso del RMO

Como se explicó en los antecedentes, el modelo cinemático es la ecuación de un mecanismo que calcula la relación de movimiento entre las articulaciones del mecanismo y los movimientos del efector final, sin tomar en cuenta las fuerzas que interactúan con el sistema. En el caso del RMO, se estudian las velocidades angulares de cada rueda y su relación con las velocidades en los ejes X y Y, así como la velocidad angular del RMO. Es decir, para la cinemática directa se calculan las velocidades en las componentes X, Y y del ángulo de orientación del vehículo completo, según la velocidad angular proporcionada por cada rueda, mientras que en la cinemática inversa se calculan las velocidades angulares en cada rueda según las velocidades en las componentes X, Y y del ángulo de orientación del vehículo. Las variables utilizadas en las ecuaciones 3.1 y 3.2 se describen en la tabla 3.1:

Símbolo	Significado
V_x	Velocidad del robot sobre su componente X.
V_y	Velocidad del robot sobre su componente Y.
V_R	Velocidad resultante del robot.
$\dot{\theta}$	Velocidad angular del robot, sobre su propio eje.
θ	Posición angular del robot, con respecto a su propio eje.
α	Angulo de la velocidad resultante del robot.
R	Radio de la rueda
l_x	Longitud del vehículo en X, desde el eje de la rueda hasta el eje del robot.
l_y	Longitud del vehículo en Y, desde el eje de la rueda hasta el eje del robot.
$\omega_{1,2,3,4}$	Velocidad angular de las ruedas.

Tabla 3.1: Tabla de Variables utilizadas en las ecuaciones de cinemática.

Matemáticamente, la cinemática directa de un RMO con 4 ruedas se puede explicar utilizando la siguiente ecuación:

$$\begin{bmatrix} V_x(t) \\ V_y(t) \\ \dot{\theta}(t) \end{bmatrix} = \frac{R}{4} \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ -\frac{1}{l_x+l_y} & \frac{1}{l_x+l_y} & -\frac{1}{l_x+l_y} & \frac{1}{l_x+l_y} \end{bmatrix} \begin{bmatrix} \omega_1(t) \\ \omega_2(t) \\ \omega_3(t) \\ \omega_4(t) \end{bmatrix} \quad (3.1)$$

A través de esta ecuación es posible hacer una estimación de la posición del robot sobre el plano cartesiano, a pesar de no contar con un acelerómetro y/o giroscopio.

Utilizando como base la cinemática directa explicada en la ecuación (3.1), es posible calcular la cinemática inversa, es decir, que se busque calcular las velocidades necesarias de cada rueda para producir las velocidades dadas del vehículo completo. Se utiliza la ecuación de cinemática inversa, la cual está dada por la siguiente ecuación:

$$\begin{bmatrix} \omega_1(t) \\ \omega_2(t) \\ \omega_3(t) \\ \omega_4(t) \end{bmatrix} = \frac{R}{4} \begin{bmatrix} 1 & 1 & -1(l_x + l_y) \\ -1 & 1 & 1(l_x + l_y) \\ -1 & 1 & -1(l_x + l_y) \\ 1 & 1 & 1(l_x + l_y) \end{bmatrix} \begin{bmatrix} V_x(t) \\ V_y(t) \\ \dot{\theta}(t) \end{bmatrix} \quad (3.2)$$

Tomando en cuenta que la velocidad resultante del RMO

$$V_r = \sqrt{V_x^2 + V_y^2}$$

y que el ángulo de ese vector de velocidad resultante es

$$\alpha = \text{atan} \frac{V_y}{V_x}$$

Ya es posible calcular las velocidades necesarias en cada rueda, para generar un movimiento en cualquier dirección sin alterar la orientación del RMO. Para más información, consultar [7, 8, 13, 14, 41, 42]

3.2. Descripción general de las características del RMO

Una de las actividades de este proyecto de investigación fue el diseño de un RMO para la prueba de algoritmos de control. Este robot fue implementado con componentes de alta calidad y bajo costo, con la finalidad de ser reproducible en cualquier institución de investigación o de educación superior.

Para describir la planta se toman en cuenta los componentes principales del tema, los cuales son la tarjeta MyRIO, encargada del procesamiento del algoritmo de control, el sensor IMU

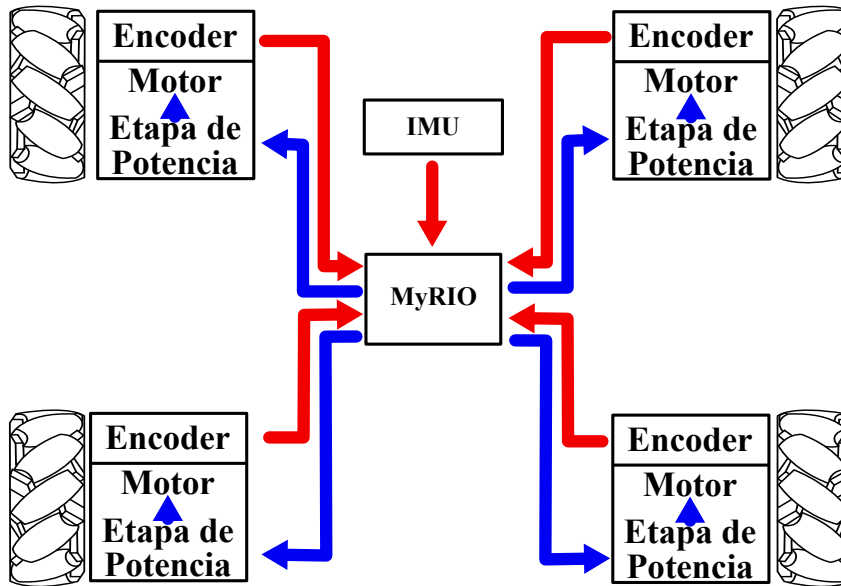


Figura 3.5: Diagrama de interconexiones del RMO, entre sus respectivos sensores, actuadores y procesador.

3 Space Sensor de Yost Labs, encargado de medir la orientación del robot, las etapas de potencia conectadas a los motores, encargados de generar el movimiento del RMO y los encoders en cada motor, encargados de medir la velocidad angular de cada motor; esto se puede visualizar de manera gráfica en la figura 3.5. A continuación se explicarán las especificaciones de cada componente de la planta, necesarias para entender los parámetros del sistema y para fabricar el chasis del robot.

3.3. Diseño y construcción de los subsistemas de instrumentación y control

3.3.1. Motor de 12V Pololu con encoder de 464.64 CPR

El RMO cuenta con 4 motores "Metal Gearmotor 25D × 63L mm MP 12V with 48 CPR Encoder", los cuales son motores Pololu, alimentados con 12V, capaces de dar una velocidad de 800 revoluciones por minuto, que además cuentan con un encoder de cuadratura alimentado con 5V y una resolución de 48 cuentas por revolución, pero debido

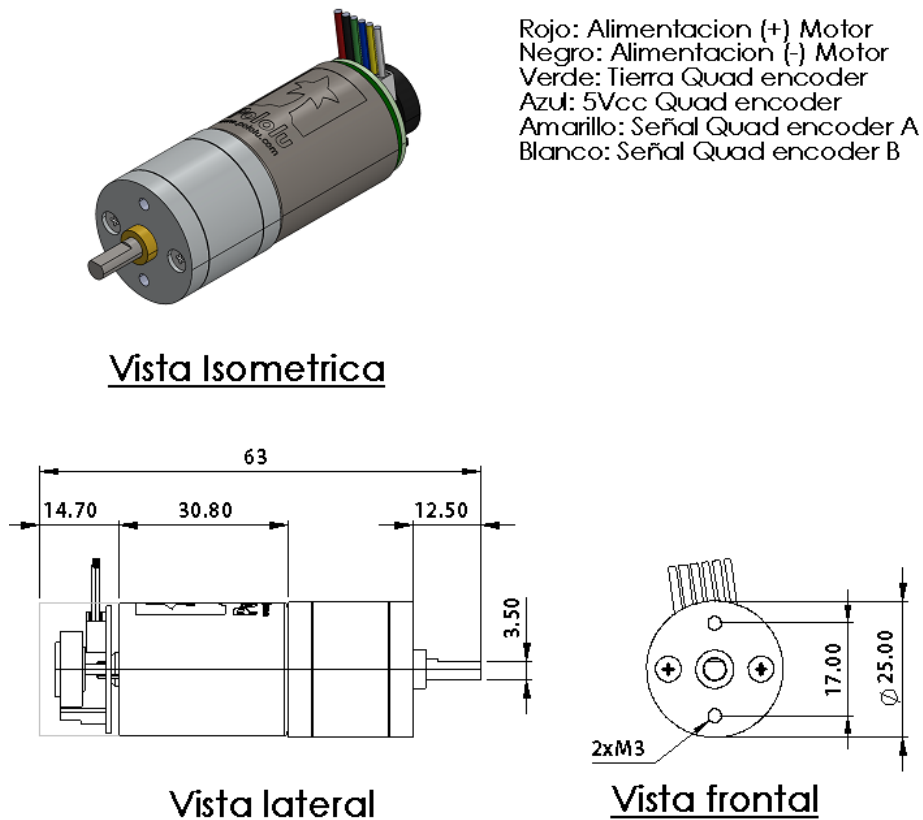


Figura 3.6: Diagrama de conexiones del motor Pololu.

a que el motor cuenta con transmisión, el encoder termina con una resolución de 464.64 cuentas por revolución del eje que se acopla a la rueda. La figura 3.6 muestra las conexiones del motor Pololu, junto con sus dimensiones en milímetros.

3.3.2. IMU: 3 Space Sensor de Yost Labs

Si bien no es completamente necesaria la implementación de un sensor espacial, para el algoritmo de control propuesto en este trabajo, es necesario para la validación de la eficacia del algoritmo de control. En este caso se incorporó un sensor espacial “Yost Labs 3 Space-Sensor”, el cual integra un acelerómetro, capaz de medir las aceleraciones en las componentes X, Y y Z, además de un giroscopio, el cual obtiene la posición angular con respecto a los ejes X, Y y Z, mejor conocidos como cabeceo, guiñada y alabeo. Por último, incorpora un compás, capaz de detectar orientación magnética. Para poder incorporar el IMU al RMO, es necesario tomar en cuenta sus dimensiones y conexiones; estas pueden

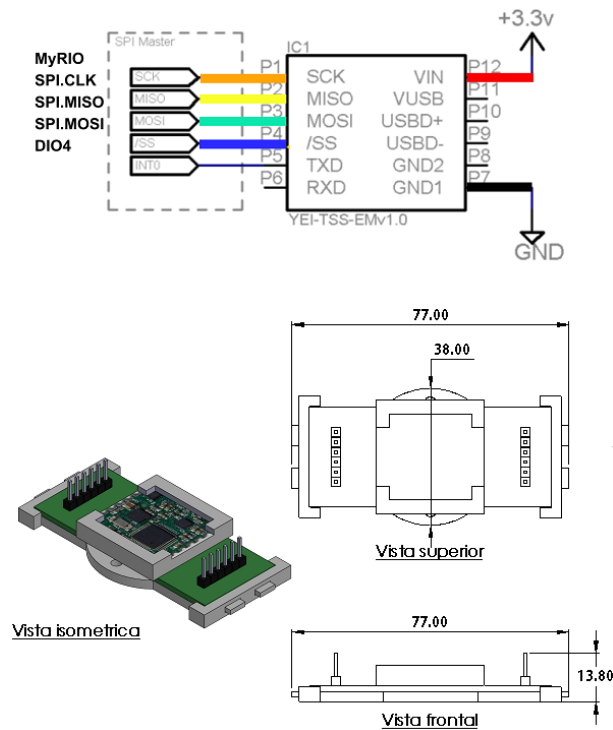


Figura 3.7: Diagrama de conexiones con dimensiones del sensor Yost Labs 3 Space-Sensor

apreciarse en la figura 3.7. Debido a que este es un sensor inteligente con diversas capacidades, es posible comunicarse con él a través de diferentes protocolos de comunicación. Debido al uso de la tarjeta MyRIO y para explotar al máximo la velocidad de la misma, se utilizó el protocolo de comunicación SPI. Si se revisa el manual del “Yost Labs 3 Space-Sensor” [43], se puede observar la gran cantidad de comandos que puede recibir el sensor; entre ellos, el de mayor interés para este trabajo es el comando para obtener la orientación tarada como ángulos de Euler. Debido a que la comunicación entre el IMU y la MyRIO es con el protocolo SPI, se siguen las instrucciones establecidas en el manual [43], donde se explica cómo enviar paquetes de información, los cuales se explicarán a continuación.

1. Se envía al sensor un un byte $0xF6$, el cual indica que se va a enviar un paquete de datos.
2. Se envía el byte del comando que se desea realizar. En este caso se pide la orientación tarada como ángulos de Euler a través del byte $0x07$.
3. Se debe esperar a que el sensor haya terminado de procesar el comando, por lo cual se

debe enviar el byte *0xFF*, el cual espera al sensor, al mismo tiempo se lee la respuesta del sensor hasta que este envíe un 1, indicando que esta listo para enviar la señal pedida.

4. Justo después de indicar que esta listo, el sensor empieza a enviar los datos de vuelta en forma de bytes, de uno en uno, cada que el maestro (MyRIO), envía el comando *0xFF*, hasta conseguir los 3 datos, en forma de números flotantes representando al cabeceo, guiñada y alabeo, donde cada 4 bytes representan un numero flotante. En otras palabras, se debe enviar el comando *0xFF*, 12 veces, para conseguir los 3 ángulos de Euler. Debido a que el sensor siempre envía la información en el orden de cabeceo, guiñada y alabeo, y solo es de interés la guiñada, el paquete es interrumpido justo después de este para no consumir mas tiempo del MyRIO.
5. Después de esto el IMU descansa hasta recibir otro byte *0xF6*.

El programa en LabVIEW con interfaz de MyRIO es detallado a profundidad en el siguiente capítulo.

3.3.3. NI MyRIO 1900 dentro del RMO

En este sistema, el papel de la NI MyRIO es el de comunicar todo el sistema, en esta tarjeta de entradas y salidas reconfigurables se programa la lectura de señales de los encoders y la lectura de la orientación del vehículo a través del IMU, se programan todos los algoritmos de control y las señales PWM que se enviarán a los puentes H para mover los motores. Las conexiones específicas de cada puerto de la MyRIO con los puertos de cada componente del RMO están representadas de manera gráfica en las figuras 3.8 y 3.9. Las medidas más importantes de la tarjeta MyRIO están especificadas en la figura 3.10.

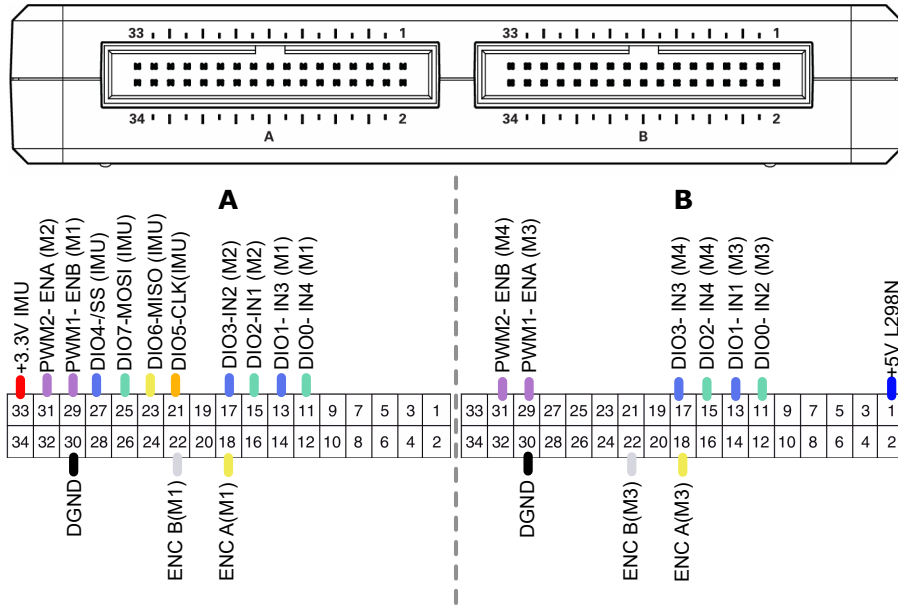


Figura 3.8: Diagrama de conexiones de los lados A y B de la MyRIO.

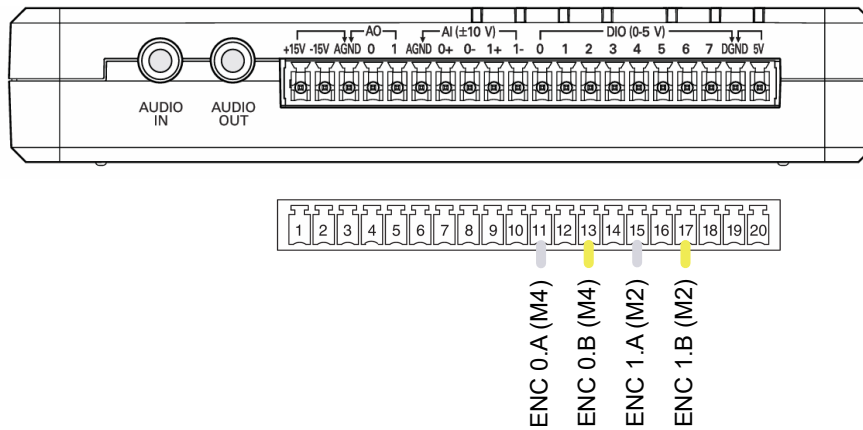


Figura 3.9: Diagrama de conexiones del lado C de la MyRIO.

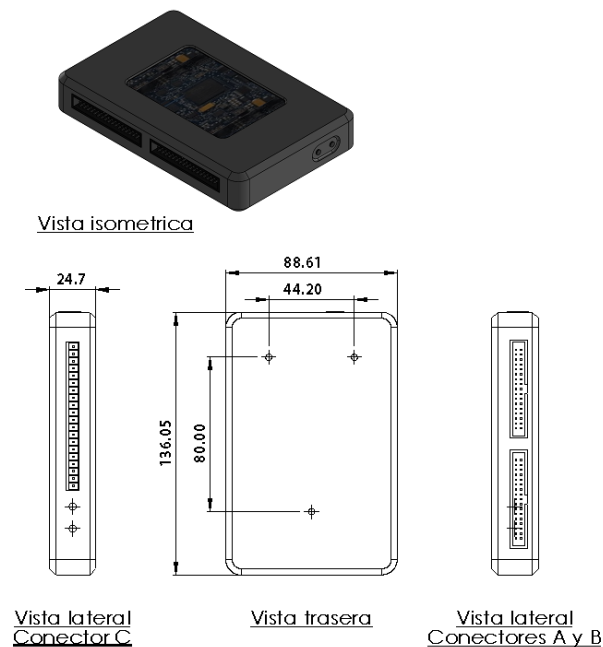


Figura 3.10: Diagrama con medidas de una tarjeta MyRIO-1900.

3.4. Diseño y construcción de los subsistemas eléctricos de potencia

3.4.1. Batería y cargado de batería

Para alimentar al sistema se utiliza como fuente principal una batería arrancadora de carros “Type S 12V 400A 8000mA”, la cual, como dice su nombre, proporciona al sistema hasta 12V, con 400A y 8000mAh. Cuenta con entradas USB tipo C, tipo A y micro USB; una imagen con las medidas y apariencia real se muestra en la figura 3.11. Se conecta directamente a la tarjeta MyRIO-1900 y las terminales de alimentación de los puentes H de los motores. Las instrucciones para cargar la batería son simples, ya que al contar con puerto de entrada de tipo C, se puede conectar a un cargador convencional de celulares, sea que este tenga alguna conexión macho de USB tipo C o micro USB.

3.4.2. Puente H

Debido a que el sistema utiliza como actuadores principales motores a los cuales se desea controlar su velocidad y sentido de giro, pero requieren una alimentación de 12V, no



Figura 3.11: Diagrama con medidas de la batería de alimentación del RMO.

se pueden conectar directamente a la MyRIO, sino que en su lugar se conectan a un “L298N Dual H-Bridge Motor Drive”, los cuales permiten el paso de voltaje de la pila, en forma de PWM, el ancho del PWM está controlado por el MyRIO. El diagrama del componente con sus medidas se encuentra en la figura 3.12 y el diagrama con las conexiones de los 4 puentes H, viene en la figura 3.13.

3.5. Diseño y construcción de los subsistemas mecánicos

3.5.1. Ruedas Mecanum

En este RMO se utilizaron 2 pares de ruedas Mecanum (2 diestras y 2 zurdas), de 60 mm de diámetro externo, con 8 rodillos de hule. Las paredes de la rueda son placas de una aleación de aluminio, lo que ofrece un cuerpo estable y duradero, para proporcionar el movimiento omnidireccional. Esta rueda cuenta con 8 rodillos de hule a lo largo de su circunferencia. Las medidas generales de la rueda se especifican en la figura 3.14.

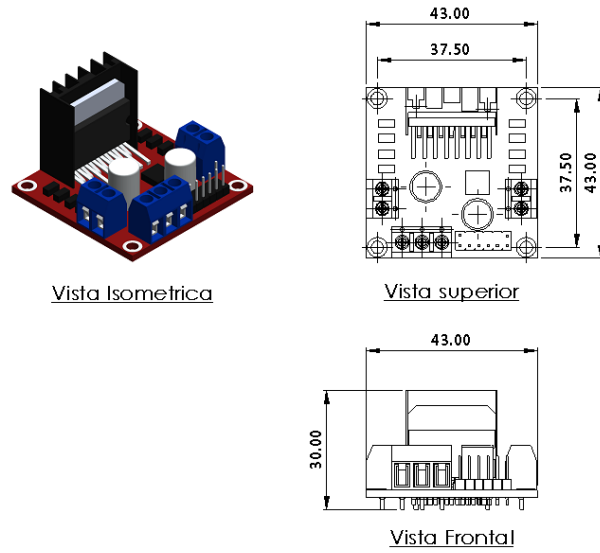


Figura 3.12: Diagrama con medidas de un L298N.

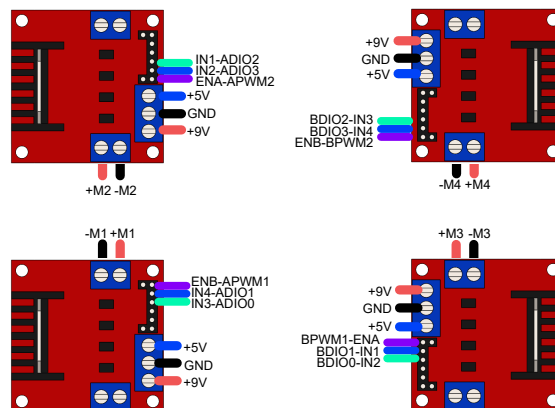


Figura 3.13: Diagrama de conexiones de los 4 puentes H.

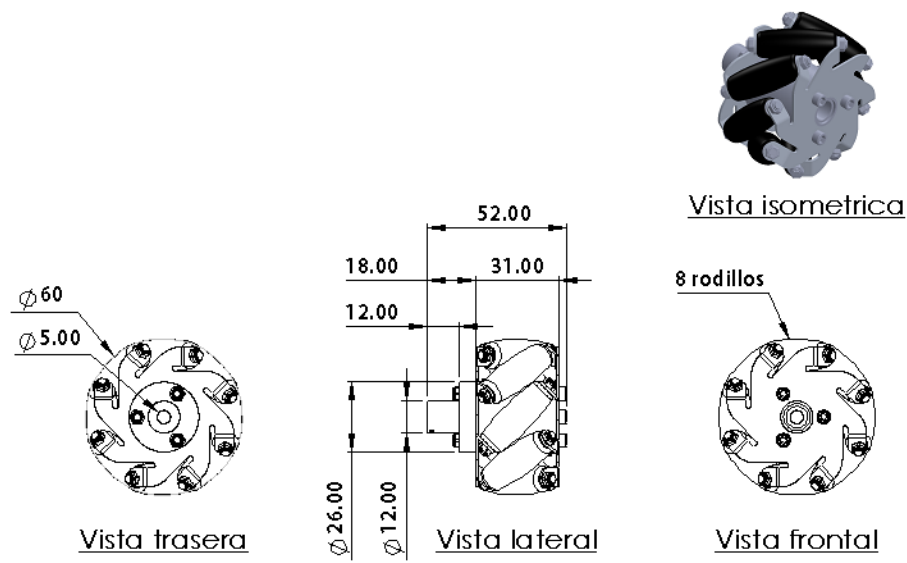


Figura 3.14: Dimensiones de la rueda Mecanum utilizada en el RMO.

3.5.2. Impresión en PLA para el diseño del chasis del RMO

Una vez conociendo todos los componentes necesarios para el ensamble del RMO, se puede generar un diseño en una aplicación de CAD, en donde se tomen en cuenta todas las medidas necesarias para fabricar un chasis. Tomando en cuenta que el RMO a estudiar es un robot didáctico, el cual se desea que sea fácilmente recreable, se planificó un diseño que pueda ser impreso en 3D y, debido a que las características mecánicas no necesitan cubrir altas temperaturas ni altos impactos, se propuso que el chasis fuera impreso con PLA (Ácido Poliláctico), ya que es un material de bajo costo y muy versátil en el área de impresión aditiva.

Las piezas impresas en PLA son: La base para la tarjeta MyRIO, la base para la pila, el chasis principal y el soporte del chasis. Y se pueden apreciar de mejor manera en la figura 3.15 Se diseñó de esta manera para que las piezas fueran apilables y ensamblables, ya que normalmente las impresoras 3D no suelen trabajar con dimensiones demasiado largas. Todas las piezas fueron impresas en una impresora “Creality Ender V3 SE” , con dimensiones máximas de $220 \times 220 \times 250$ mm. Cabe mencionar que la tornillería usada en el vehículo real es, en su mayoría, tornillos de cabeza Allen, con diámetro de rosca de 3.2 mm y que antes de ensamblar las bases de la pila y la MyRIO, se realizó una etapa de soldado de cableado eléctrico, para evitar accidentes, falsos contactos debido a las vibraciones del

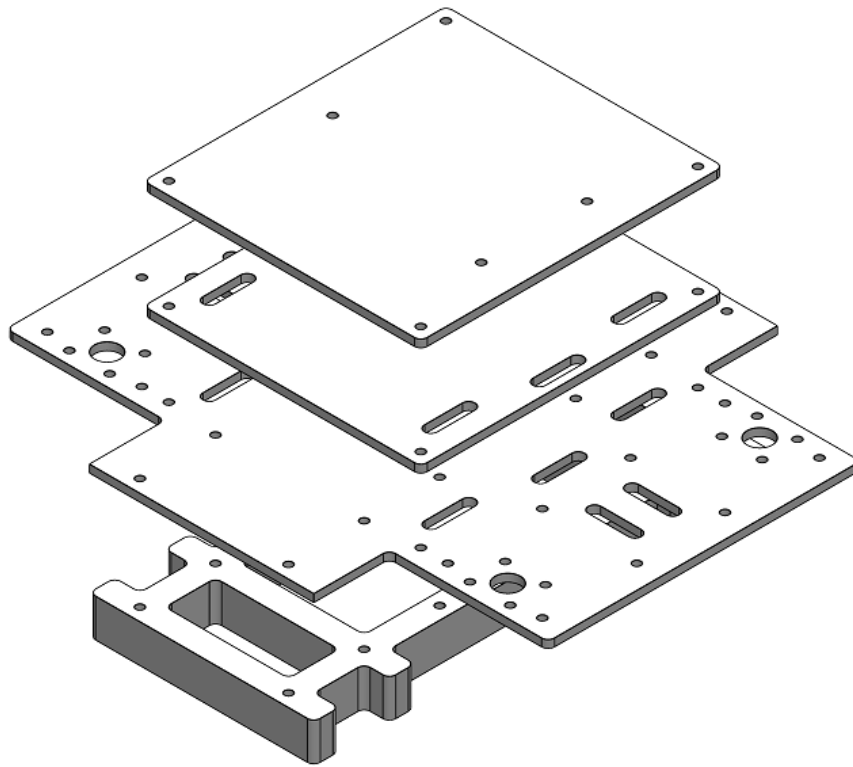


Figura 3.15: Piezas 3D impresas en PLA para el chasis del RMO.

vehículo al andar. Para un mejor entendimiento de las piezas, se presentan los planos de cada una de las piezas, empezando por las bases para la tarjeta MyRIO y para la pila en la figura 3.16 y posteriormente la base principal ó chasis del RMO y el soporte del chasis en la figura 3.17. Para más información sobre el ensamblado del vehículo y/o los archivos para imprimir en 3D, del RMO, consultar el siguiente enlace: <https://drive.google.com/drive/folders/11EX0CxBut18sL0QwxZ1Ck8r2g5DPguv0?usp=sharing>.

Capítulo 3.

Diseño y construcción de un robot móvil omnidireccional con ruedas Mecanum

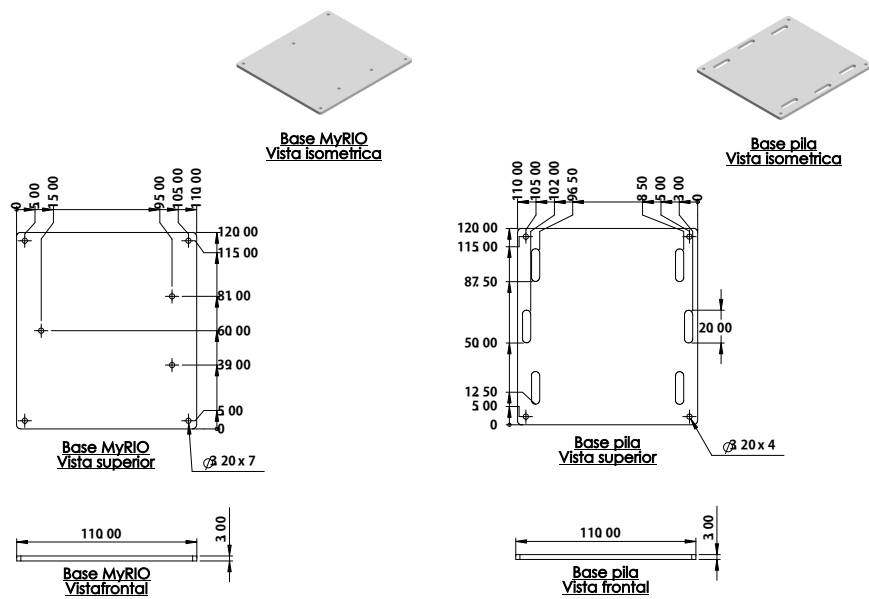


Figura 3.16: Diagrama con medidas de las placas impresas en 3D para el soporte de la MyRIO y para el soporte de la pila.

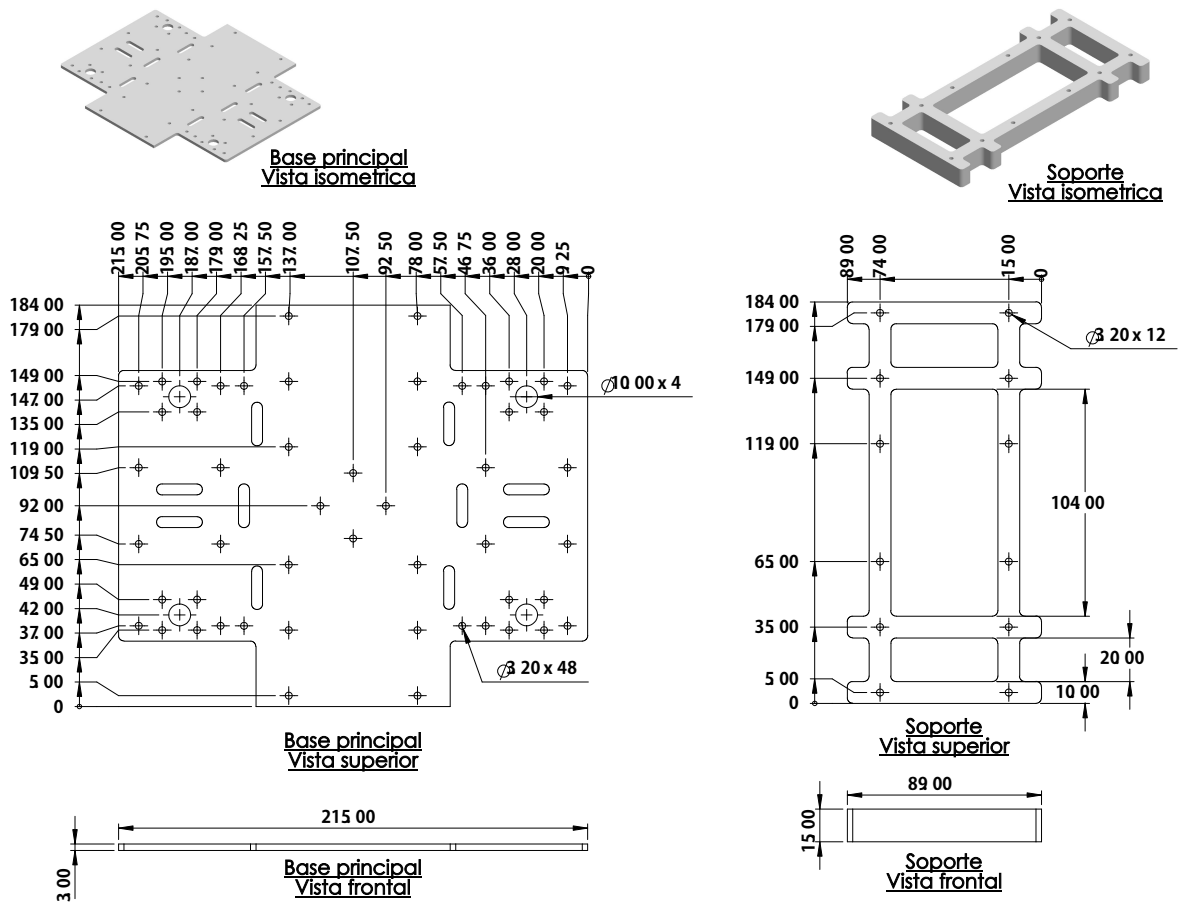


Figura 3.17: Diagrama con medidas de las placas impresas en 3D para el chasis principal del RMO y su soporte.

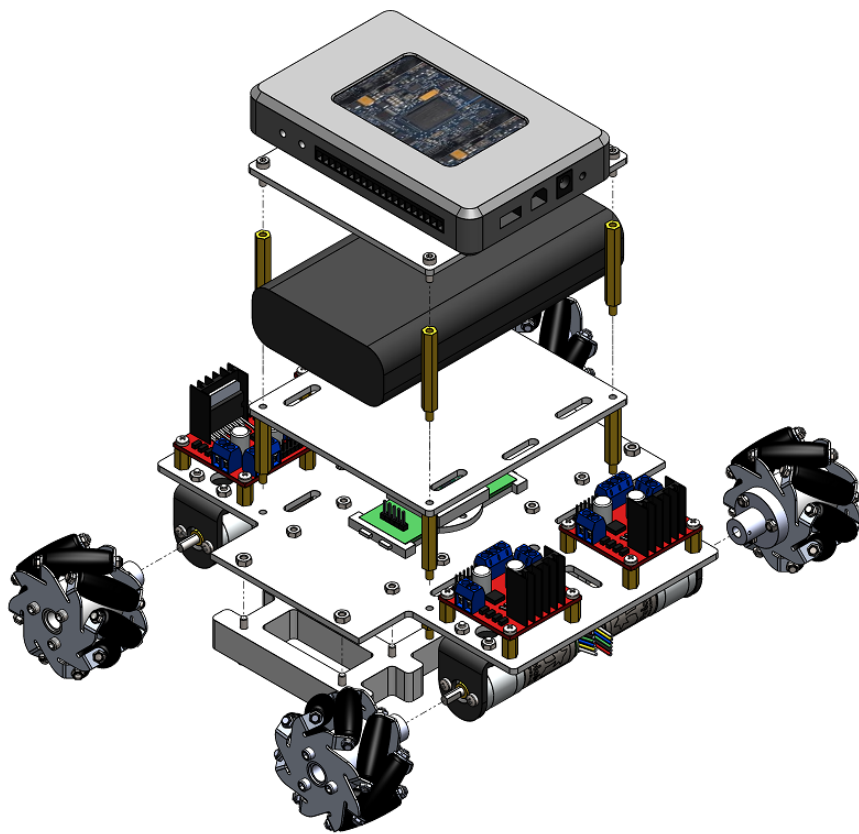


Figura 3.18: Vista explotada 3D del RMO, con sus componentes base.

Capítulo 4

Desarrollo de módulos de control utilizando la plataforma myRIO y LabVIEW

Para desarrollar un algoritmo de control suficientemente robusto y con un buen desempeño computacional, fue necesario hacer diferentes pruebas de los programas principales que forman el proyecto en LabVIEW, como lo son: la adquisición y guardado de datos en la tarjeta NI MyRIO 1900, el programa a cargo de la generación de velocidad y dirección de un motor de CC, el programa para la generación de los movimientos típicos de un RMO, el programa encargado de la comunicación con el IMU para el censado de orientación de guiñada del RMO y, por último, un programa que combina todos los programas principales que se utilizaron a lo largo del proceso de movimiento del vehículo (Generación de movimiento de un RMO, mientras censa la orientación de guiñada y, al finalizar cada experimento, logra guardar los datos de manera eficiente dentro de la tarjeta MyRIO). Todos los programas se diseñaron en base a los fundamentos de LabVIEW propuestos en [36].

4.1. Programa para el guardado de datos eficiente a través de una NI MyRIO 1900

Uno de los programas más relevantes para este proyecto es el que realiza el guardado eficiente de datos en la memoria interna de la tarjeta NI MyRIO 1900. Este código fue utilizado en todo momento, debido a que es necesario para la revisión de datos fuera de línea en cada experimento.

Se tomó la decisión de almacenar los datos experimentales dentro de la misma tarjeta con base en las recomendaciones de [10], que propone no usar variables locales si no es completamente necesario, y las recomendaciones de [19], donde se menciona que el uso de variables compartidas retrasa la velocidad del procesador al ejecutar un programa, y por lo tanto, disminuye la velocidad de adquisición de datos y reduce la eficacia del controlador. El código para el guardado de datos se muestra en la figura 4.1 y funciona de la siguiente manera.

Tomando en cuenta que todos los programas se ejecutarán de manera continua y se necesitan utilizar funciones de la librería de Simulación y Control, se agrega una estructura *control and simulation loop*, la cual no se detendrá hasta que se cumpla, ya sea la condición de tiempo establecido, o hasta que se detenga la ejecución de manera manual. Ya que el lazo de simulación y control proporciona un tiempo determinístico, preestablecido por el usuario, es necesario calcular el tiempo real a través de un método como el que se proporciona en este capítulo, el cual es el tiempo real que toma el procesador en realizar cada iteración, intentando cumplir con las peticiones del usuario. Para el cálculo del tiempo se utilizó la función de *Tick counts (usec)* y un arreglo para convertirlo a segundos, este proceso se muestra en la figura 4.2. Los datos correspondientes al tiempo y el intervalo de tiempo en cada ciclo de ejecución del programa son agregados a un arreglo, para luego ser acumulados con la función *Collector*, la cual guarda los datos de cada iteración hasta que el programa deje de ejecutarse, ya que el *Collector* almacena las señales de entrada junto con el tiempo del lazo de simulación y control, es necesario separarlas de sus respectivos *Clusters* y combinar los datos en un arreglo de datos, para después enviarlos al SubVi de guardado de datos en MyRIO, el cual se muestra en la figura 4.3. En este último *SubVi*,

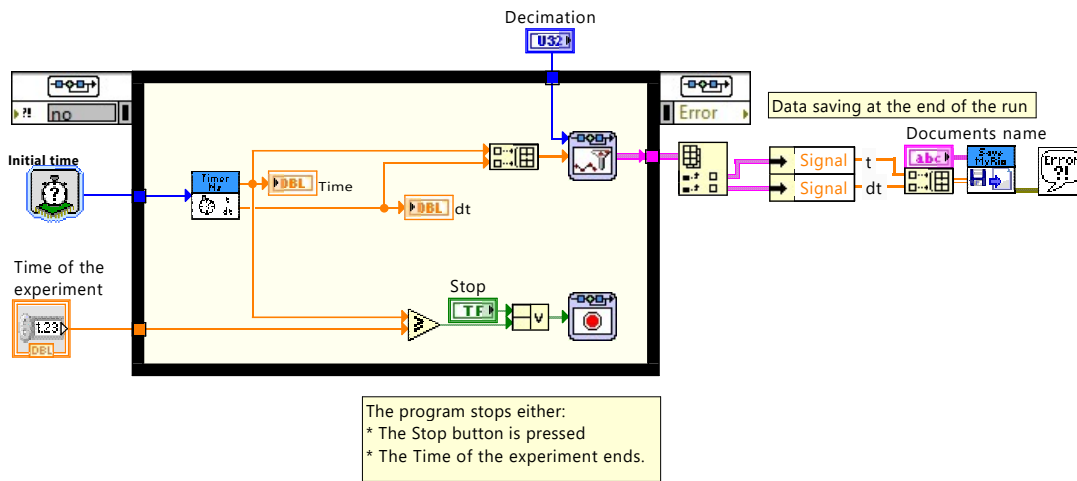


Figura 4.1: Código visual de LabVIEW para el guardado de datos en MyRIO.

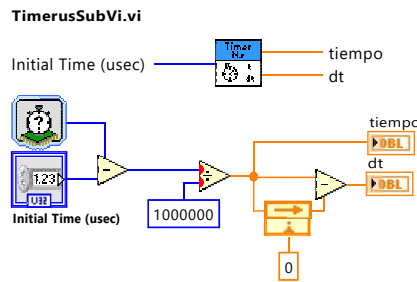


Figura 4.2: Diagrama del *Sub Vi* encargado del conteo de tiempo en segundos con resolución en micro segundos.

se convierten los datos en cadenas de caracteres delimitadas por comas, y se ingresan como información para crear un archivo delimitado por comas y en lugar de ingresar una dirección, se utiliza la función *Application Directory* la cual proporciona una dirección dentro de la misma computadora en la que se crea el código de LabVIEW, que en este caso sería la tarjeta MyRIO.

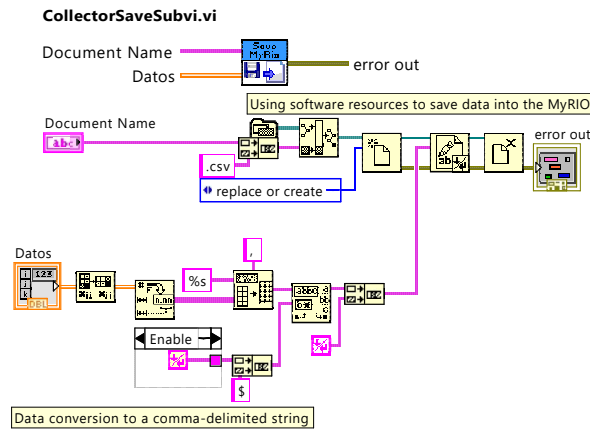


Figura 4.3: Diagrama del *SubVi* encargado de convertir datos en un archivo delimitado por comas.

4.1.1. Acceso a los datos guardados dentro de la tarjeta NI MyRIO

Existen diversas maneras de acceder a los datos guardados en la MyRIO, uno de ellos es a través de la aplicación NI MAX, la cual se instala junto con LabVIEW. En esta aplicación se pueden revisar todos los sistemas conectados a la computadora del usuario, como se muestra en la figura 4.4. Una vez identificado el módulo con el que se está trabajando, es posible acceder a la transferencia de datos, como se muestra en la figura 4.5. Esto abrirá el navegador por defecto definido por el usuario, con la dirección IP de su tarjeta de adquisición de datos como enlace y pedirá un usuario y contraseña como se muestra en la figura 4.6. El usuario predeterminado en MyRIO es “admin” y la contraseña predeterminada está en blanco, pero es necesario establecer una contraseña para poder acceder a los datos dentro de la tarjeta. Una vez ingresados, aparece la librería interna completa de MyRIO. Para acceder a los datos guardados en experimentos con MyRIO, es necesario acceder a :files/C/ni-rt/startup, tal y como se muestra en la figura 4.7. Los documentos guardados suelen aparecer en forma de texto, tal y como se muestra en la figura 4.8, pero al dar clic derecho y "guardar como", se pueden guardar en la computadora en el formato establecido desde LabVIEW; en caso de contar con una aplicación que pueda leer archivos csv como Microsoft Excel, se descargarán automáticamente. Debido a que la MyRIO se lanzó en 2013, otra manera de acceder a su librería interna es desde plataformas como

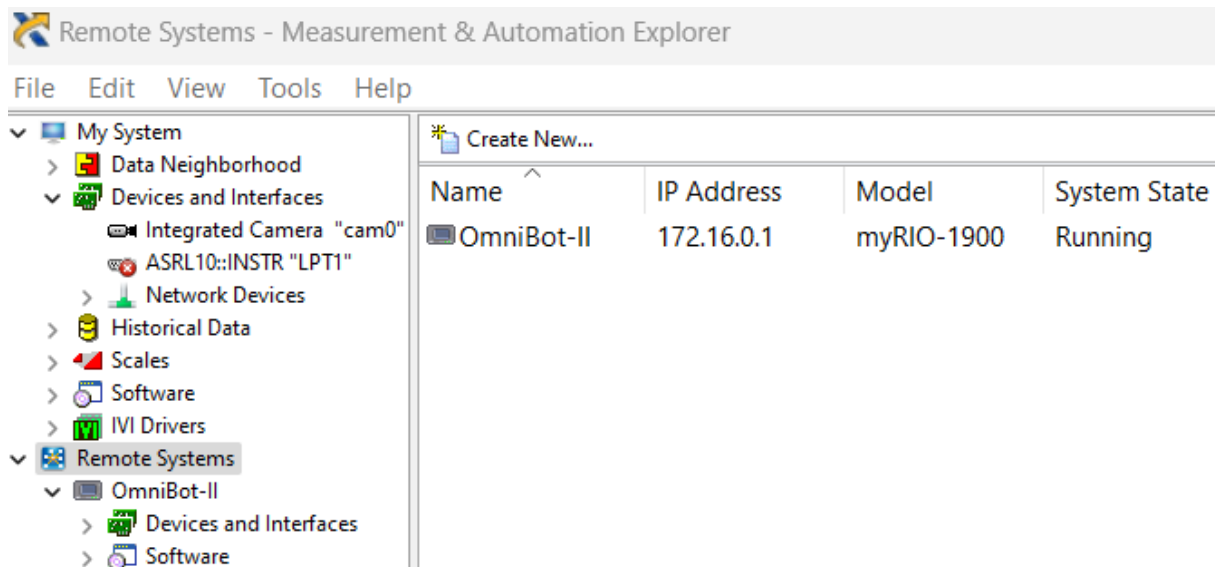


Figura 4.4: Panel sistemas remotos de NI MAX.

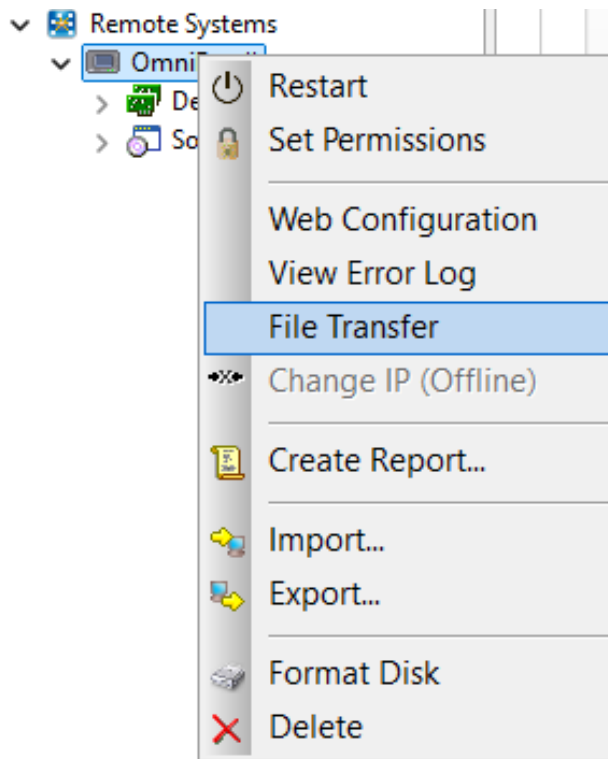


Figura 4.5: Opciones extras (Click derecho) sobre la MyRIO en NI MAX.

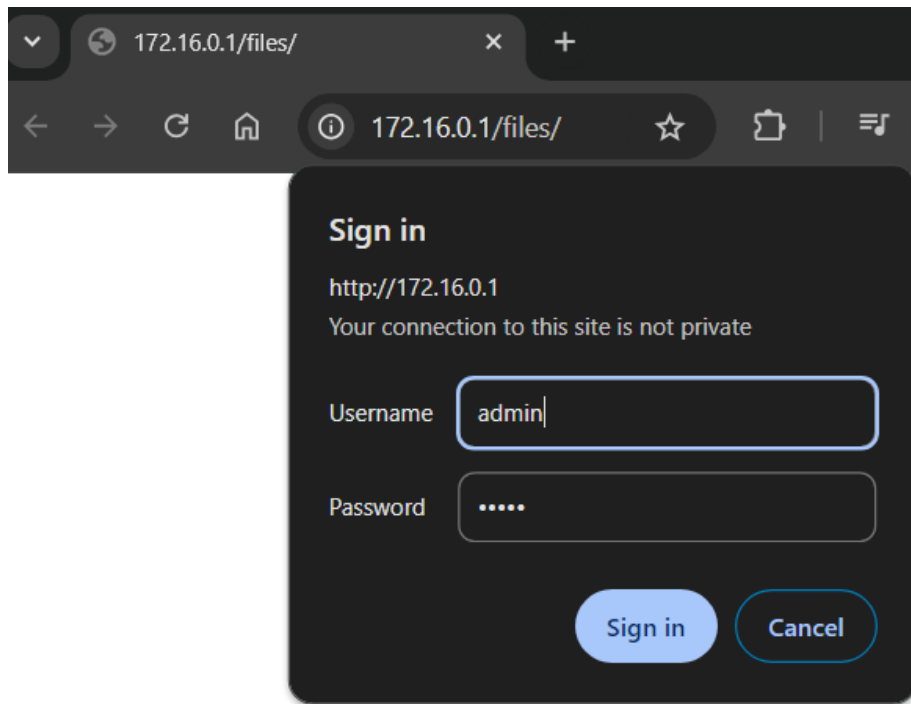


Figura 4.6: Ingreso de información para acceder a los datos internos de la MyRIO a través de google Chrome.



Figura 4.7: Directorio interno de la MyRIO para el acceso a los experimentos guardados internamente.

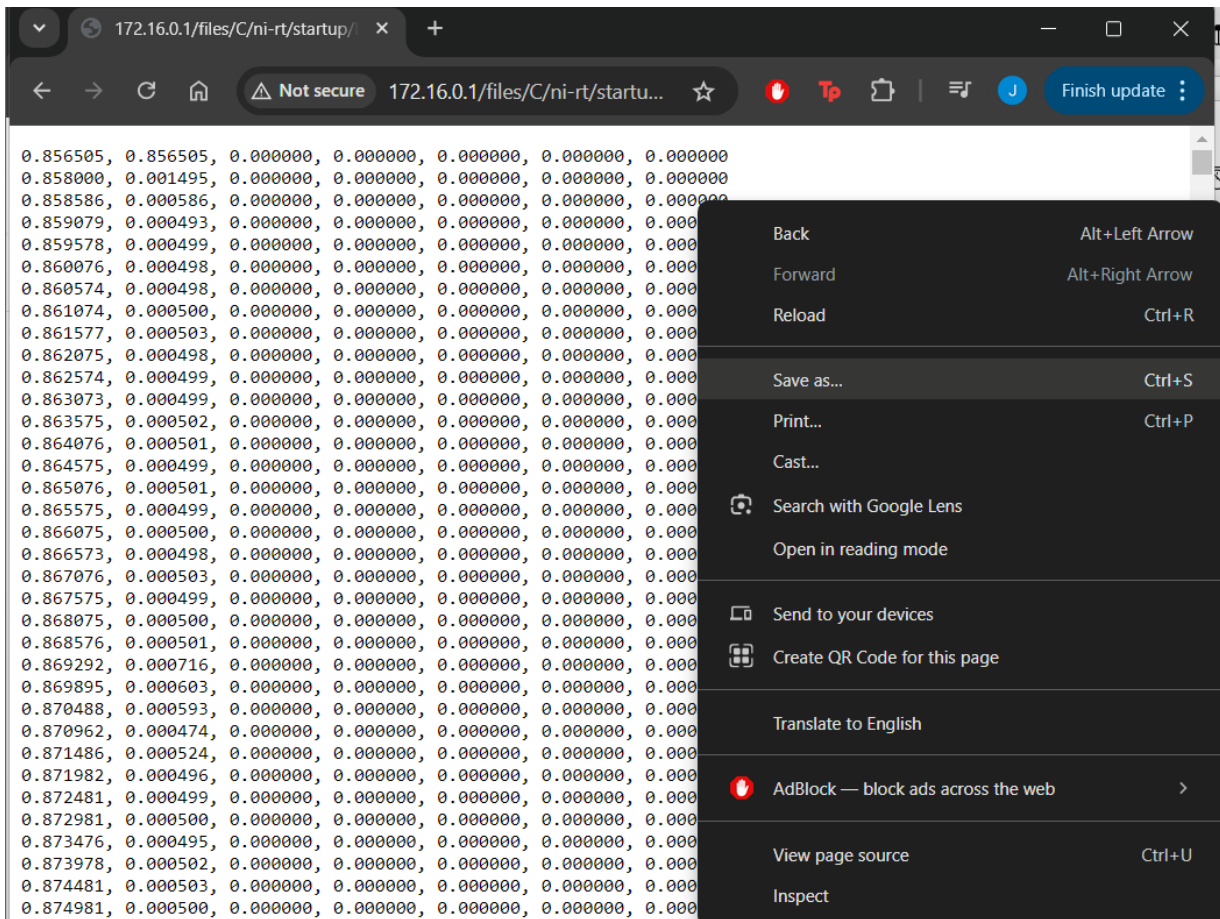


Figura 4.8: Obtención de datos desde NI MAX y google chrome, de un experimento guardado internamente en una MyRIO-1900

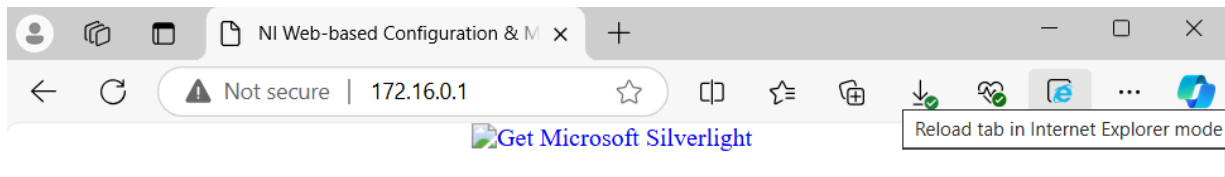


Figura 4.9: Dirección IP de una MyRIO como enlace en microsoft edge/ internet explorer.

Internet Explorer (no Microsoft Edge). Para esto es necesario ingresar la dirección IP, desde Microsoft Edge y refrescar la página como página en modo Internet Explorer, como se muestra en la figura 4.9. Posteriormente se mostrará la verdadera plataforma diseñada para el acceso y configuración de datos de la MyRIO, donde también se necesita ingresar el usuario y la contraseña para poder acceder a su librería, como se muestra en la figura 4.10. Los documentos guardados desde LabVIEW se encuentran en la misma dirección: /C/ni-rt/startup. Desde aquí, todos los archivos pueden ser descargados y se muestra el nombre de los documentos, tamaño en bytes y última fecha de modificación, como en la figura 4.11.

4.1.2. Tiempo proveído por el lazo de simulación y control, contra tiempo medido por la MyRIO

Según LabVIEW, el lazo de simulación y control ofrece un tiempo determinístico; por lo tanto, es un tiempo que no es real. Esto es mencionado en [1] y para comprobarlo se realizó un experimento comparativo entre el tiempo proporcionado por el lazo de simulación y control y el tiempo medido por la MyRIO, con la función *Tick counts (usec)*. Las señales guardadas en el código se pueden apreciar en la figura 4.12 . Este programa es el que se utiliza en el siguiente apartado para generar movimiento y dirección en un motor de CC, mostrado en la figura 4.15, pero analizando únicamente la comparación entre los datos de tiempo guardados por el lazo de simulación y control y el medido por la tarjeta MyRIO. Solo se realizó un experimento corto, de aproximadamente 10 segundos, el cual, según el tiempo proporcionado por el lazo de simulación y control, duró más de 20 segundos; esto se muestra en la figura 4.13. Los datos sobre tiempos que serán utilizados en los análisis en el resto del proyecto serán los obtenidos con la medición de tiempo de la tarjeta MyRIO.

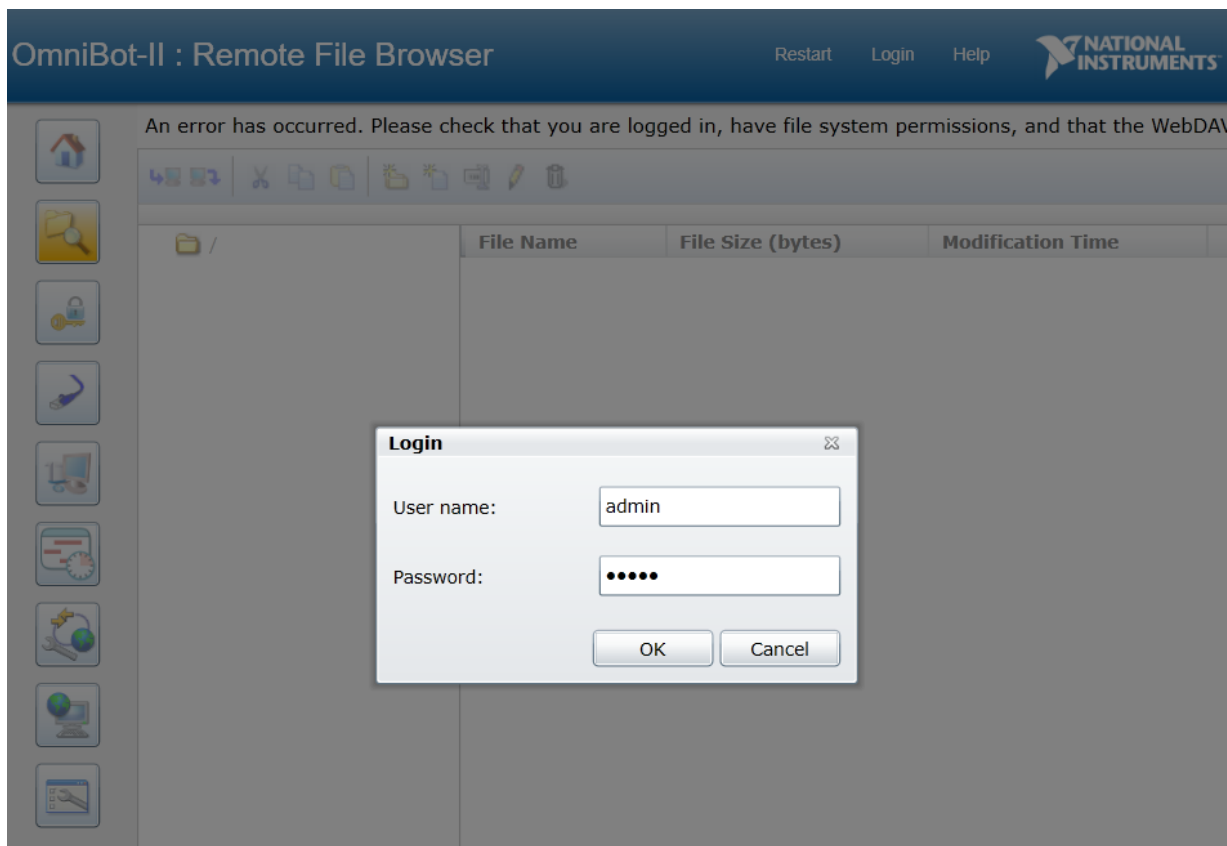


Figura 4.10: Ingreso de información para acceder a los datos internos de la MyRIO a través de Internet Explorer.

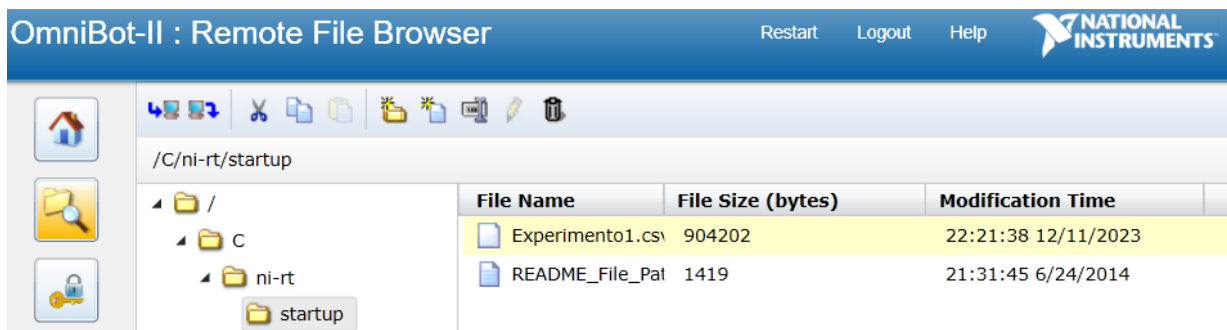


Figura 4.11: Directorio interno de la MyRIO para el acceso a los experimentos guardados internamente desde Internet Explorer.

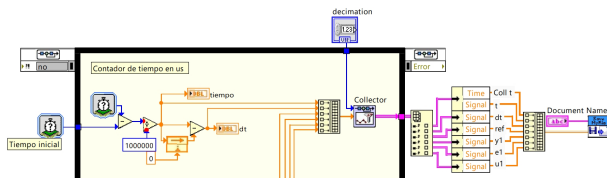


Figura 4.12: Vista de una comparativa de la toma de datos utilizando el tiempo proveído por el lazo de simulación y control y el tiempo medido por la MyRIO.

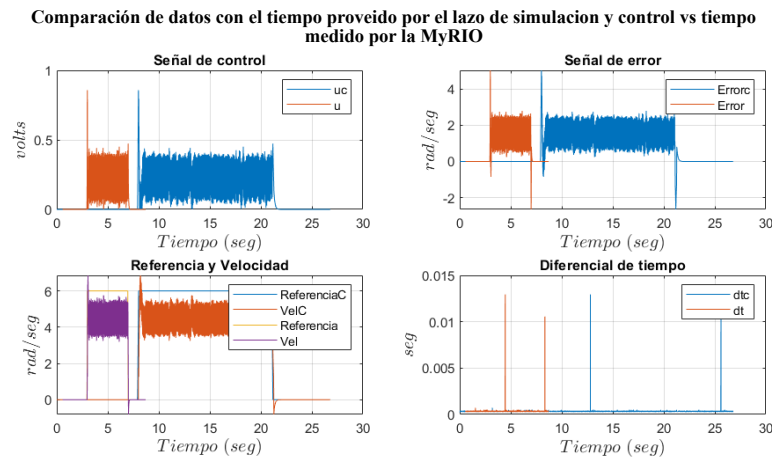


Figura 4.13: Comparación de datos con el tiempo proveído por el lazo de simulación y control contra el tiempo medido por la MyRIO.

4.2. Programa para generar movimiento en un motor de CC

En este apartado se describirá de manera más detallada el código mostrado en la figura 4.15, encargado de generar movimiento y la dirección de un motor de CC. En este programa se inicia eligiendo la rueda a analizar, lo cual abre las referencias de la tarjeta MyRIO para el uso del puerto PWM para el control de velocidad del motor, una salida digital para controlar la dirección de giro del motor y los puertos del encoder, para poder leer las cuentas del encoder sobre el motor a analizar. Las cuentas contadas por el encoder son después ingresadas al *SubVi*, mostrado en la figura 4.14, para convertir las cuentas del encoder en radianes por segundo. Nótese que la velocidad después es ingresada a una función de transferencia que sirve como filtro, para obtener una velocidad más real del sistema. El programa también incluye el código de la figura 4.1, para el guardado de datos, y a este se le añaden la señal PWM de entrada al motor de CC y la señal de velocidad en radianes por segundo del motor calculada a partir del encoder.

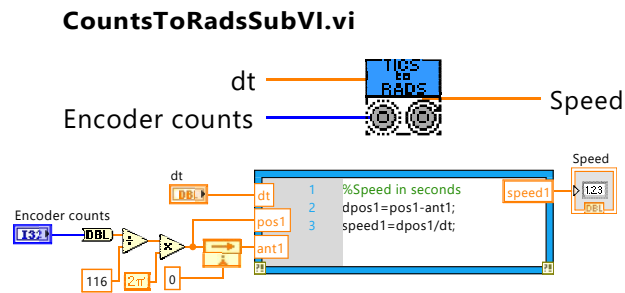


Figura 4.14: Diagrama de bloques y conexiones para la conversión de cuentas del encoder a radianes por segundo.

4.3. Programa para generar los movimientos típicos de un RMO

Una vez teniendo un programa capaz de generar el PWM para controlar la velocidad y el sentido de giro del motor, es posible integrarlo a un código nuevo que se comuniquen con los 4 motores y encoders de un RMO de 4 ruedas.

Tomando en cuenta la información mostrada en 3.1.1, es posible crear las condiciones que permitan el movimiento a cada movimiento típico, según se desee, por lo cual se crea el *SubVi* mostrado en la figura 4.18, el cual a través de botones booleanos, crea las referencias matemáticas para definir la dirección de giro en cada motor o si es necesario mantener al motor en un estado de reposo forzado. Este *SubVi* llamado *OmniRefs*, es agregado a un código en el cual se incorpora la lectura de 4 encoders, y la generación de movimiento y de giro de 4 motores, como el mostrado en la figura 4.16, el cual es manipulado por el panel frontal mostrado en la figura 4.17, en el cual también es posible observar las señales de PWM aplicadas a cada motor y las velocidades producidas. Haciendo uso de este código, es posible mover un RMO generando movimientos típicos omnidireccionales, como los mostrados en la figura 3.4.

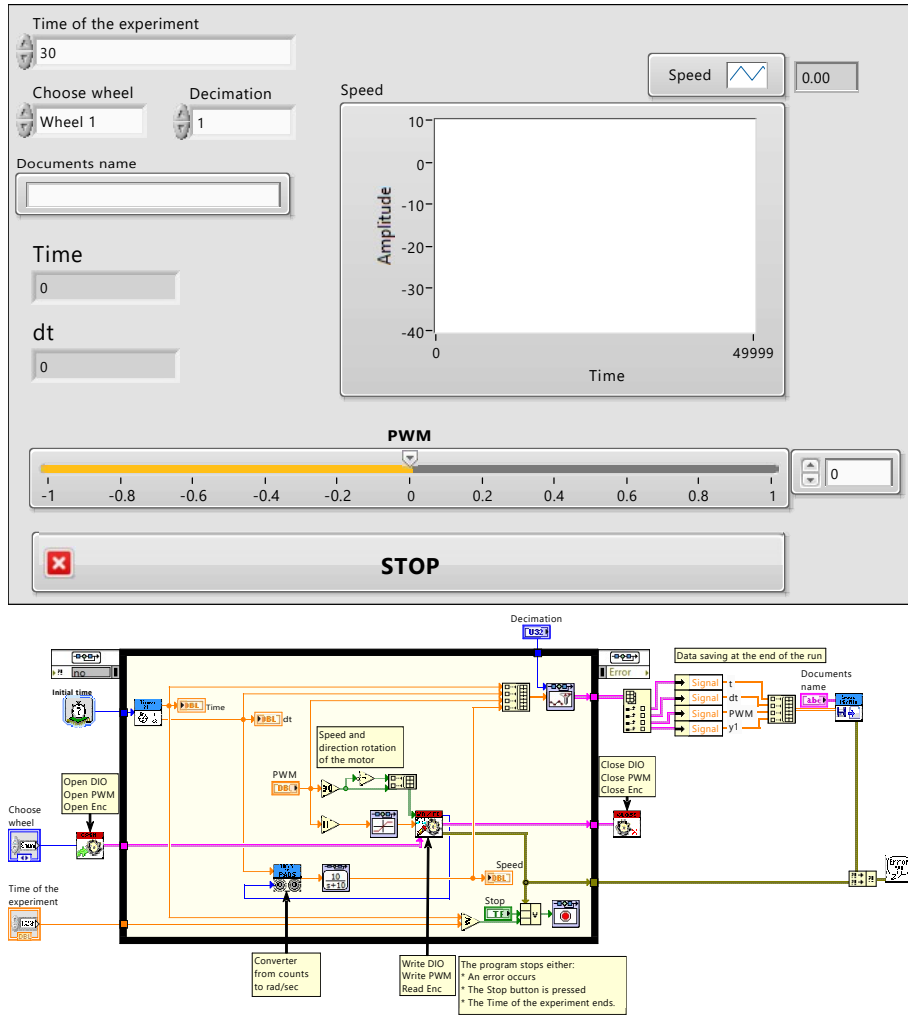


Figura 4.15: Panel frontal y diagrama de conexiones de el programa para generar movimiento y dirección de un motor de CC.

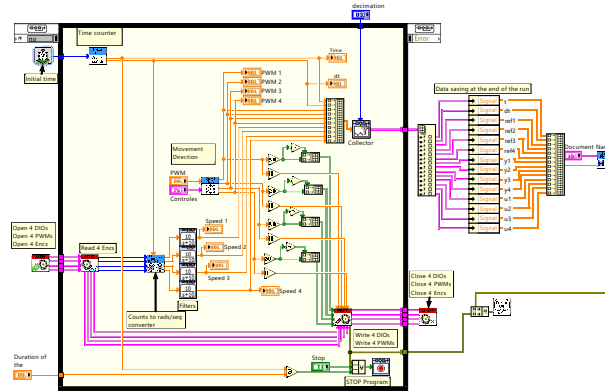


Figura 4.16: Diagrama de bloques del programa para generar movimientos omnidireccionales en un RMO.

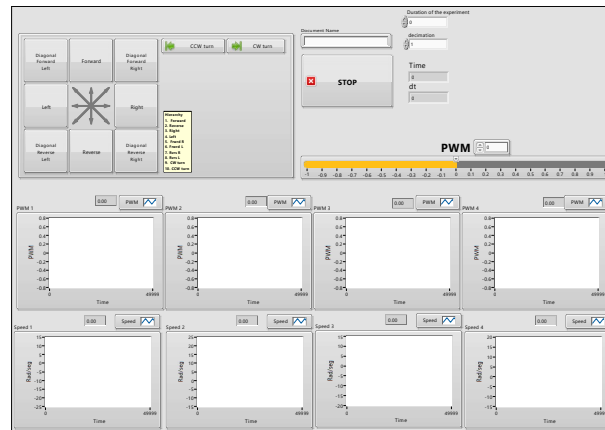


Figura 4.17: Panel frontal del programa para generar movimientos omnidireccionales en un RMO.

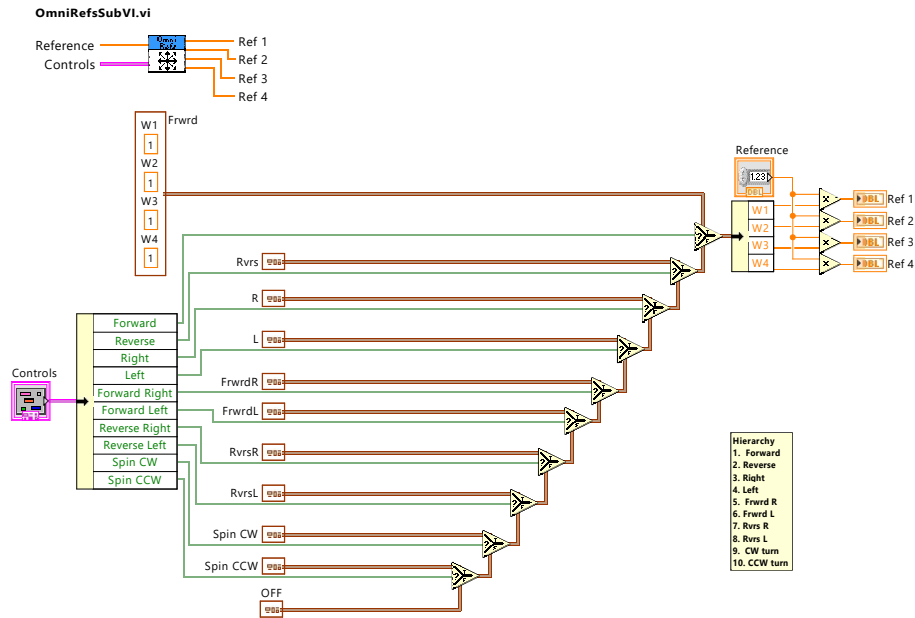


Figura 4.18: Diagrama de bloques del *SubVi*, encargado de convertir los botones booleanos de dirección, a referencias numéricas para cada rueda.

4.4. Programa para leer los datos de censado de orientación de un IMU (3 space-sensor by Yost Labs)

El último segmento de código que será utilizado en todos los códigos del proyecto es la medición del ángulo de guiñada con respecto al eje central del RMO, mostrado en la figura 4.19. Este es de gran importancia para verificar la eficacia de cada uno de los algoritmos de control propuestos. Este código hace uso de 3 subcódigos que se describen a continuación.

- **OpenIMU:** Este código, mostrado en la figura 4.20, se encarga crear las referencias para los canales que utiliza el IMU en la tarjeta MyRIO, los cuales son los puertos de comunicación SPI y un canal de salida digital para crear los pulsos de reloj para la comunicación SPI.
- **IMUYaw:** Este código, mostrado en la figura 4.21, se encarga de enviar al IMU desde la MyRIO, los paquetes de bytes necesarios para leer los ángulos de Euler. Para esto se toma en cuenta lo mencionado en la sección 3.3.2, donde se menciona el orden

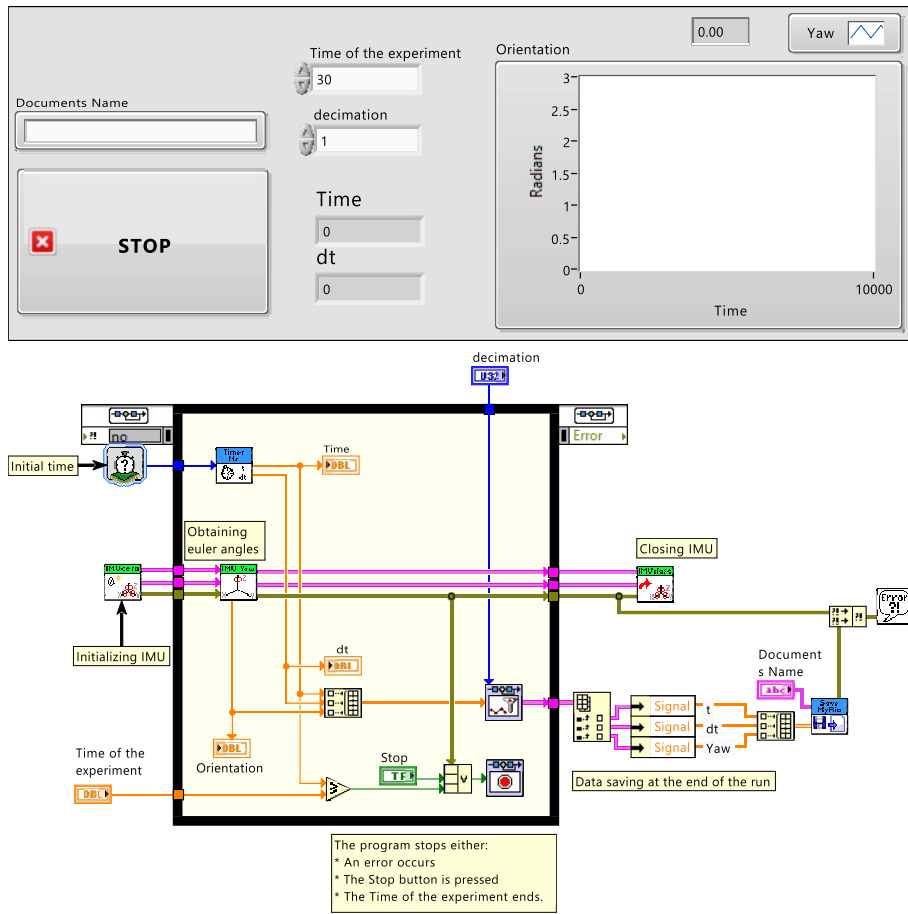


Figura 4.19: Diagrama de bloques y conexiones para el V_i , encargado de leer el ángulo de guiñada del RMO.

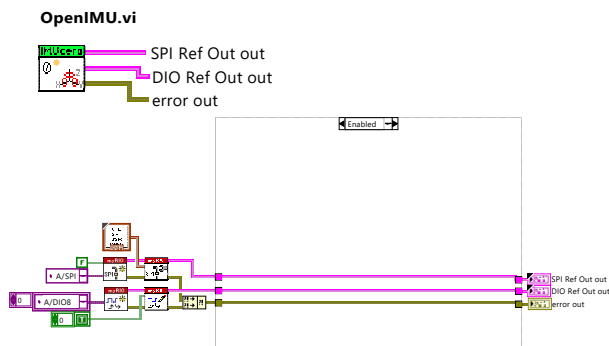


Figura 4.20: Diagrama de bloques y conexiones para el SubVi, encargado de abrir las referencias para utilizar el IMU.

correcto de transferencia de paquetes de bytes. Este *SubVI*, empieza generando el primer byte $0xF6$, para inicializar al IMU, accediendo al *SubVI*, mostrado en la figura 4.22, donde se logra apreciar que unicamente se escriben los comandos y se crea un pulso de reloj, para darle la señal al IMU que se le ha enviado un comando. Lo mismo se hace en el resto de paquetes de bytes, donde el segundo byte en enviarse es el comando $0x1$ que pide los datos de los ángulos de Euler que han sido tarados internamente en el IMU. Posteriormente es necesario esperar al IMU enviando el byte $0xFF$, hasta que el IMU envíe un byte $0x1$, que representa que el IMU esta listo y procederá a enviar los 4 bytes para cada angulo de Euler. Debido a que solo es de interés el angulo de guiñada, solo se tomaran los segundos 4 bytes del IMU con el comando $0xFF$, por lo que se harán 4 iteraciones para el primer angulo de cabeceo, el cual no se guarda, y después de las siguientes 4 iteraciones representantes del angulo de guiñada se hará un arreglo de estos bytes para el *SubVi* mostrado en la figura 4.23, donde se convierten 4 bytes en un numero flotante utilizando el formato IEEE-754, explicado en [34]. Para finalizar, el valor numérico de punto flotante, representando a la guiñada, es enviado como dato al exterior del *SubVi* y el caso del código cambia a “End” donde se detiene el *SubVi*.

- **CloseIMU:** Este es el ultimo *SubVi* importante relacionado al uso del IMU, donde lo único que sucede es el cerrado de las referencias a los canales de uso de comunicación SPI y de salida analógica de la MyRIO.

4.5. Funcionamiento del programa completo de movimiento omnidireccional de un RMO, con censado del angulo de orientación y guardado de datos

En este programa se junta el código de los programas descritos en los apartados anteriores 4.1 ,4.3 y 4.4 para crear un código capaz de generar los movimientos típicos de un

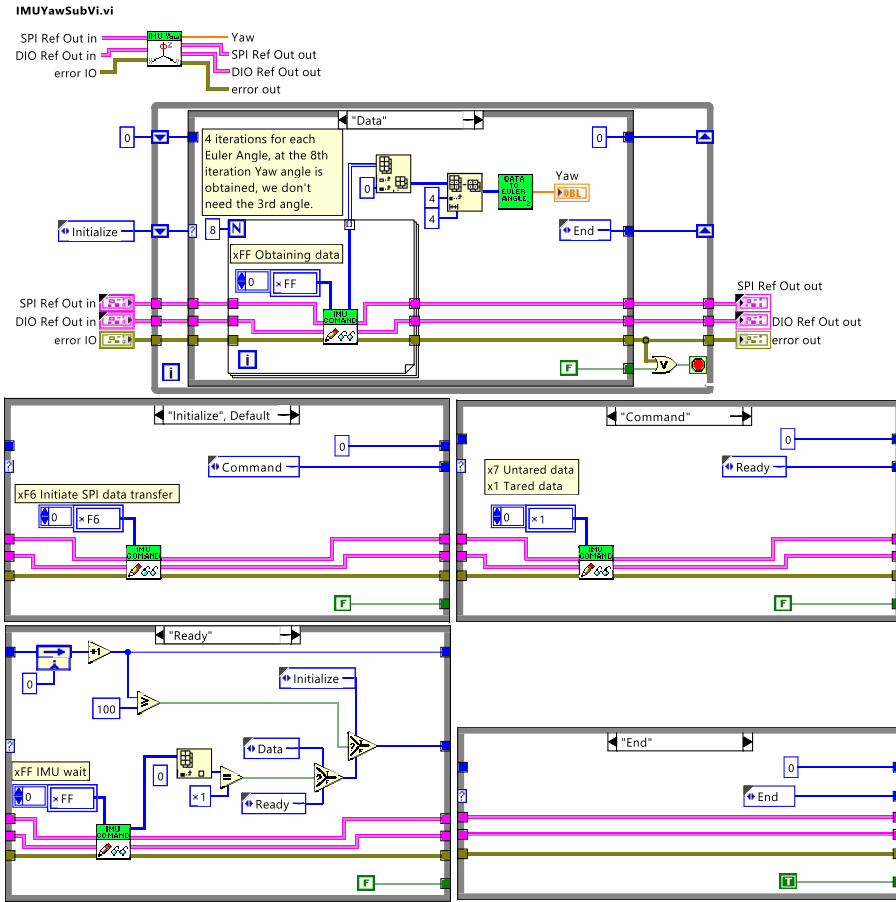


Figura 4.21: Diagrama de bloques y conexiones para el *SubVi*, encargado de la lectura de un dato del ángulo de guiñada.

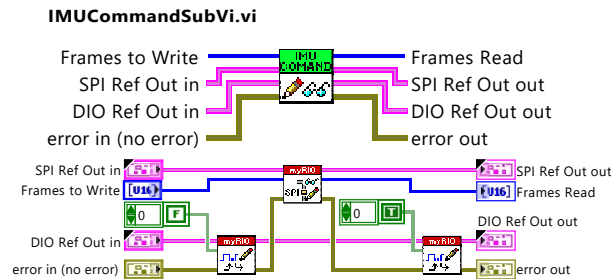


Figura 4.22: Diagrama de bloques y conexiones para el *SubVi*, encargado de enviar un comando al IMU.

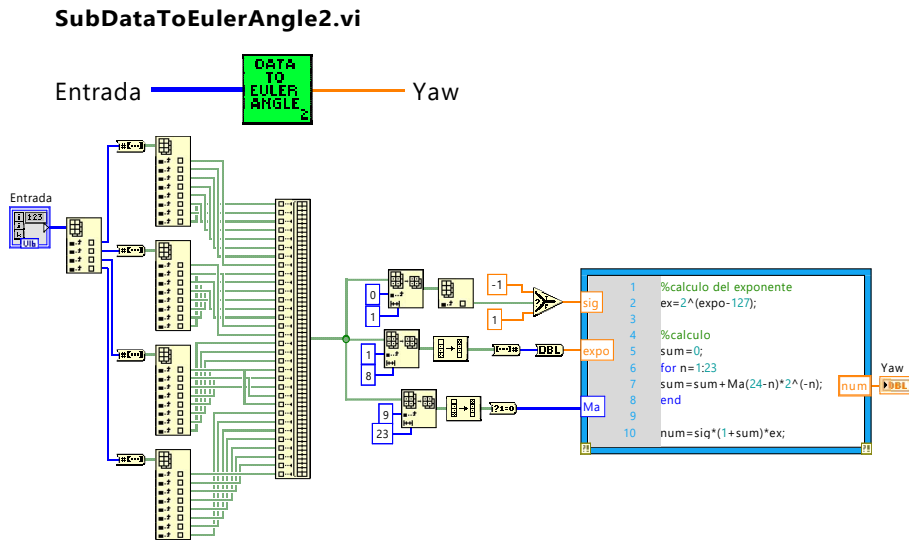


Figura 4.23: Diagrama de bloques y conexiones para el *Sub Vi*, encargado de convertir los 4 bytes proporcionados por el IMU, en un numero flotante representando el angulo de Euler de guiñada.

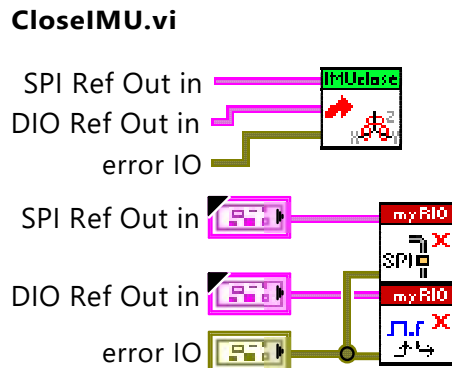


Figura 4.24: Diagrama de bloques y conexiones para el *Sub Vi*, encargado de cerrar las referencias de comunicación con la MyRIO.

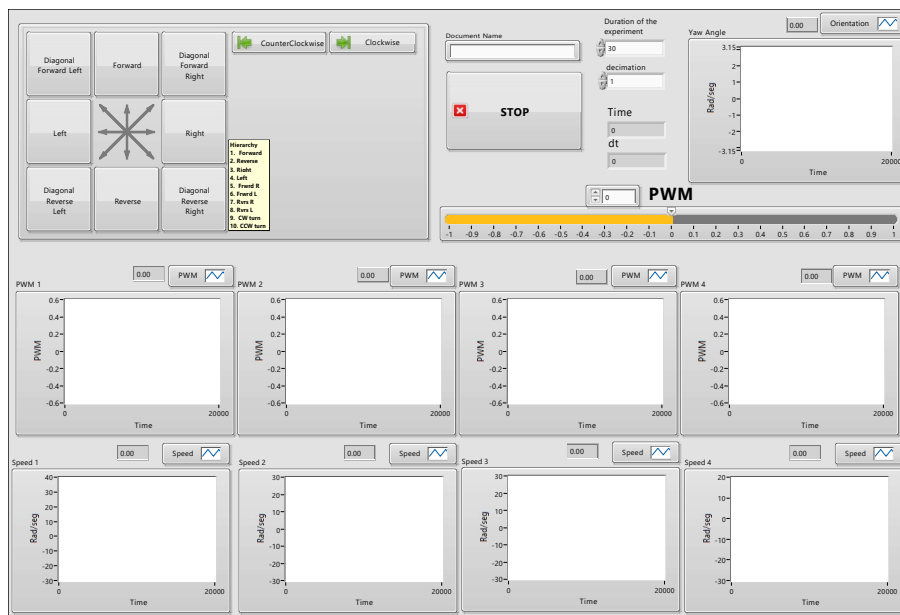


Figura 4.25: Panel frontal completo para la lectura de velocidad angular en cada rueda y control de movimiento en lazo abierto de un RMO, con censado del ángulo de guiñada y guardado de datos.

RMO, mientras se censa el ángulo de guiñada a través del IMU y al finalizar el experimento guarde los datos deseados en la tarjeta de almacenamiento interna de la MyRIO. El código del programa se puede visualizar en 4.26, mientras que el panel frontal se muestra en 4.25. En este programa se define la dirección típica del RMO, así como el voltaje de entrada aplicado en cada rueda, controlado por el porcentaje del ciclo de trabajo del PWM. Además, se visualizan esos voltajes de entrada junto con las velocidades angulares de cada motor, junto con el ángulo de guiñada del RMO.

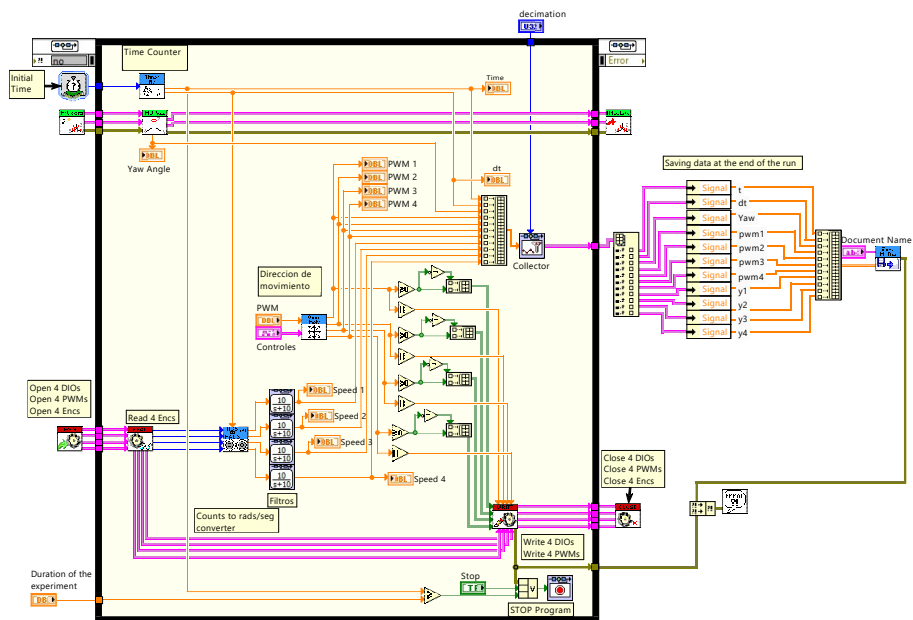


Figura 4.26: Diagrama de bloques completo para la lectura de velocidad angular en cada rueda y control de movimiento en lazo abierto de un RMO, con censado de ángulo de guiñada y guardado de datos.

Capítulo 5

Modelado matemático e identificación de parámetros de los 4 motores de CC con base en experimentos

Como se detalló en la figura 3.5, el sistema puede ser descrito como 4 motores de CC, los cuales pueden ser representados como sistemas predominantemente de 1er orden, que forman parte de las articulaciones del vehículo y deben ser enumerados del 1 al 4, para poder describir el movimiento del vehículo completo a través de la cinemática directa. Es decir, es necesario hacer un modelado matemático de los 4 motores de CC para poder hacer simulaciones del sistema completo. El modelo matemático de los motores también es usado para diseñar un controlador por retroalimentación de estado, el cual se describirá mejor en el siguiente capítulo. En este capítulo se detalla el proceso para la identificación de los 4 motores de CC del RMO con base en experimentos, así como la validación de los modelos matemáticos obtenidos y una simulación del sistema completo, tomando en cuenta los limitantes reales del sistema.

5.1. Experimentación y obtención de la función de transferencia estimada para los 4 motores

Utilizando el programa mostrado en el apartado 4.5, se realizaron los experimentos para la identificación del sistema. Para este proceso se aplicaron una serie de entradas de tipo escalón a cada uno de los motores al mismo tiempo; estas entradas son el porcentaje y dirección del ciclo de trabajo del PWM, las cuales se guardaron al igual que cada una de las señales de las velocidades angulares. Las diferentes entradas escalón programadas en cada experimento se muestran en la tabla 5.1, tomando en cuenta que el sistema inicia con una entrada $u_0 = 0$ hasta el 1er segundo y termina con una entrada $u_{(t \geq 9)} = 0$. En la figura 5.1

No. experimento	$u_{(1 \leq t < 5)}$	$u_{(5 \leq t < 9)}$
Exp. 1	40 %	100 %
Exp. 2	65 %	90 %
Exp. 3	50 %	60 %
Exp. 4	-40 %	-100 %
Exp. 5	-65 %	-90 %
Exp. 6	-50 %	-60 %
Exp. 7	50 %	-50 %
Exp. 8	-60 %	60 %
Exp. 9	80 %	50 %
Exp. 10	-80 %	-50 %

Tabla 5.1: Porcentaje de ciclo de trabajo PWM aplicado como entrada escalón en los 10 experimentos para la identificación del sistema.

se muestran 4 de los experimentos con sus respectivas señales de entrada y las velocidades angulares de los 4 motores. Una vez guardados los datos de la señal de entrada y las 4 señales de salida, se exportan al software de MATLAB, donde con el comando *iddata* se encapsulan los datos de entrada y salida de una rueda en un experimento en específico, en un solo objeto. Los 10 experimentos de un solo motor se mezclan con el comando *merge (iddata)*, el cual puede almacenar varios objetos *iddata* en un solo objeto. Posteriormente, este objeto con los 10 experimentos puede ser utilizado por la función *tfest* para estimar la función de transferencia de un motor. Este proceso se aplica de la misma manera para los 4 motores para poder obtener las 4 funciones de transferencia necesarias para definir

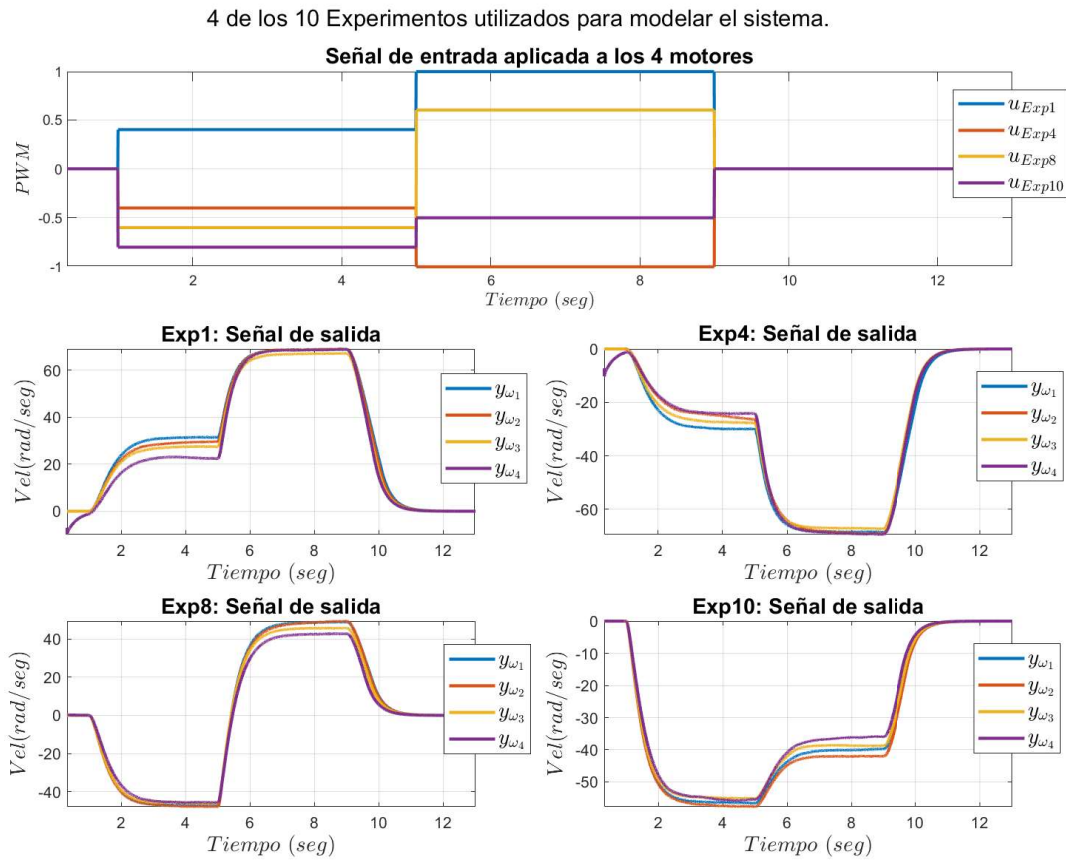


Figura 5.1: 4 de los 10 experimentos mostrando la respuesta de los 4 motores ante series de entradas escalón.

al sistema:

$$\begin{aligned} FT\omega_1(s) &= \frac{378.2}{s + 4.982}, & FT\omega_2(s) &= \frac{373.6}{s + 4.921}, \\ FT\omega_3(s) &= \frac{371.4}{s + 5.166}, & FT\omega_4(s) &= \frac{354.5}{s + 5} \end{aligned} \quad (5.1)$$

5.2. Validación de la función de transferencia

Con un modelo matemático, es posible simular todo tipo de experimentos sin someter al sistema real a dichos experimentos, pero para eso es necesario contar con un modelo matemático suficientemente útil. Esto se puede hacer comparando las respuestas simuladas del sistema con las respuestas reales del sistema ante las mismas entradas en los mismos instantes de tiempo. En este caso se utilizaron las señales del experimento 1 y se introdujeron las mismas señales de entrada a las funciones de transferencia propuestas con el comando *lsim*, para producir una señal de salida en cada motor. Las comparaciones entre las velocidades angulares reales y las velocidades angulares simuladas se pueden observar en la figura 5.2. Con estas gráficas es posible determinar que los modelos obtenidos son útiles, al predecir el comportamiento predominantemente de 1er orden del sistema, además de tener respuestas transitorias y amplitudes similares. Implementando la ecuación de cinemática en Simulink, además de sus respectivas conversiones para coordenadas móviles y coordenadas globales, es posible hacer simulaciones del vehículo completo. Estas simulaciones se hicieron utilizando el programa en Simulink mostrado en la figura. Utilizando este programa se puede apreciar la necesidad de controladores debido a las no igualdades entre los motores. Incluso en las simulaciones donde todavía no se toma en cuenta los parámetros de las ruedas, las etapas de potencia y perturbaciones externas, es posible notar las diferencias entre los motores y cómo afectan en los movimientos del vehículo, en donde según los movimientos típicos de un RMO con ruedas Mecanum, únicamente deben avanzar hacia adelante todas las ruedas, pero claramente por la ecuación de cinemática directa esas diferencias de velocidades angulares provocan desviaciones en su movimiento y en su orientación. Esta diferencia en las ruedas puede verse reflejada en la figura 5.4 y en el movimiento completo del RMO simulado en la figura 5.5 5.6 donde se muestran el ángulo de orientación del RMO, así como su movimiento en las coordenadas móviles X, Y y su

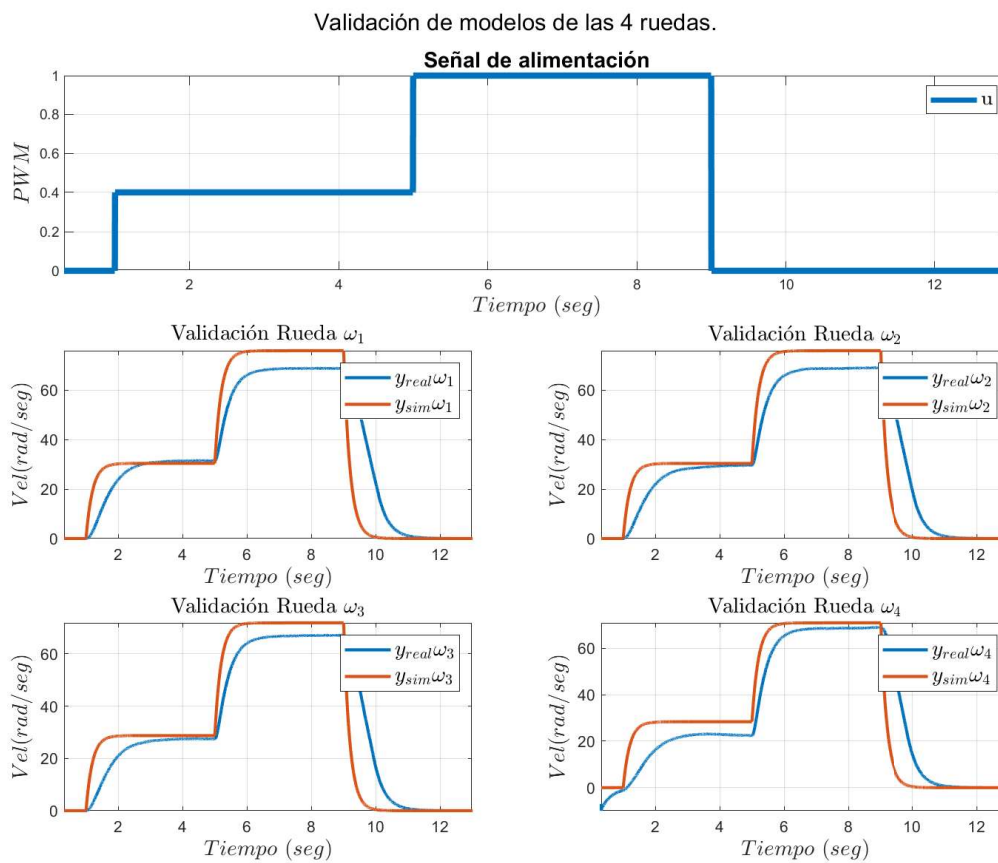


Figura 5.2: Respuestas del sistema real y de la respuesta simulado para la validación de los modelos matemáticos obtenidos.

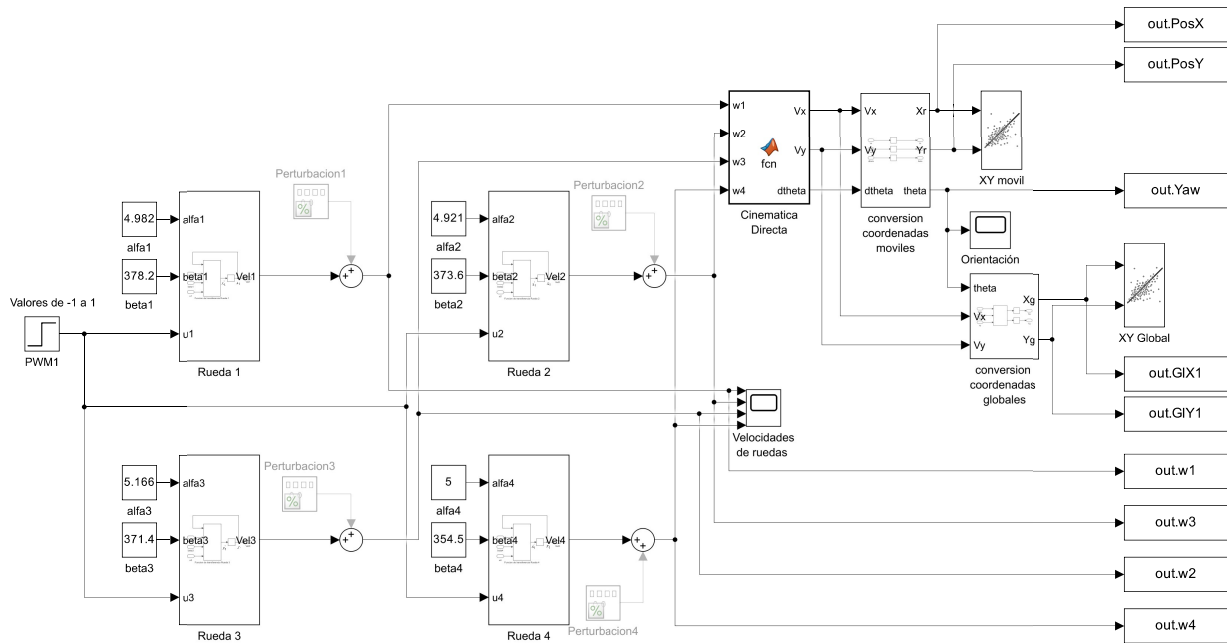


Figura 5.3: Programa de Simulink para simular el comportamiento del RMO en lazo abierto.

movimiento convertido en coordenadas globales X, Y. En la figura 5.6 se logra observar la desviación global del vehículo, es decir, la desviación provocada al desorientar su ángulo θ y seguir avanzando hacia adelante, lo cual es un problema de navegación automatizada que no se abarcará en este trabajo de tesis, pero demuestra la necesidad de implementar controladores en lazo cerrado útiles, para controlar el movimiento del RMO en el marco global, pero que no puede ser atendido correctamente, si no se implementa un controlador útil para controlar el RMO en el marco móvil, del mismo.

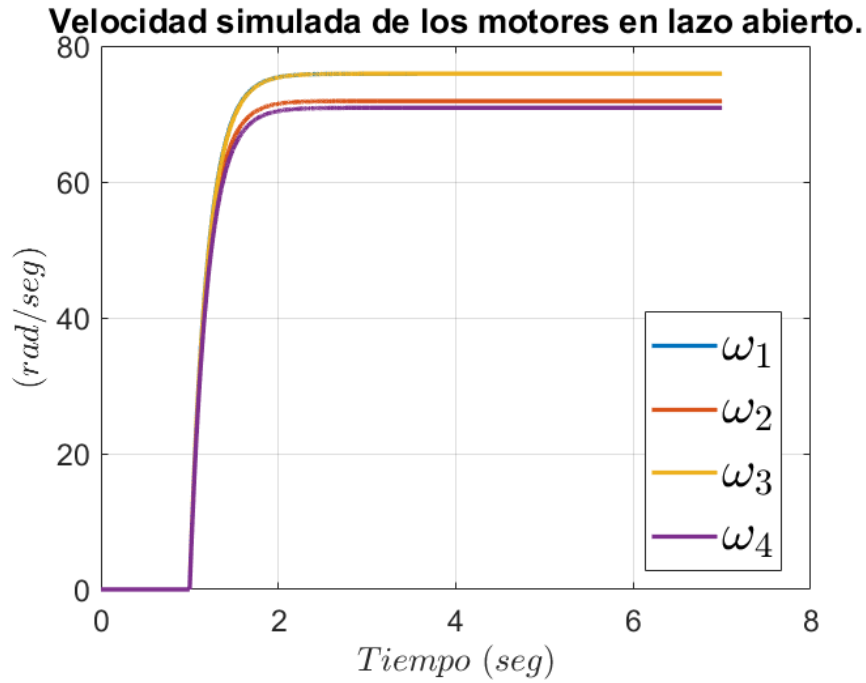


Figura 5.4: Velocidad angular de los motores al aplicarle una entrada PWM del 100 % durante 7 segundos.

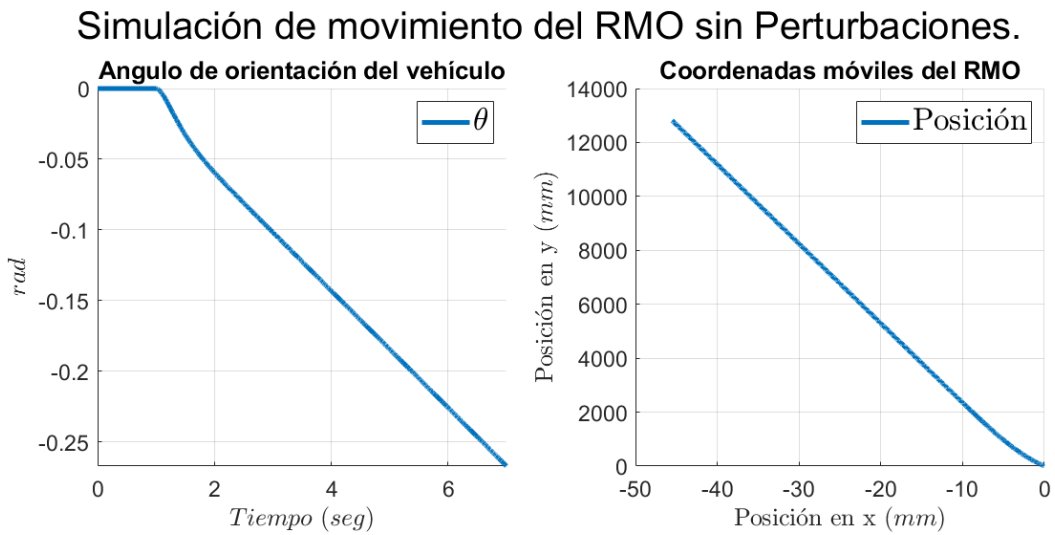


Figura 5.5: Posición del RMO, respecto a sus coordenadas móviles al aplicarle una entrada de PWM del 100 % durante 7 segundos.

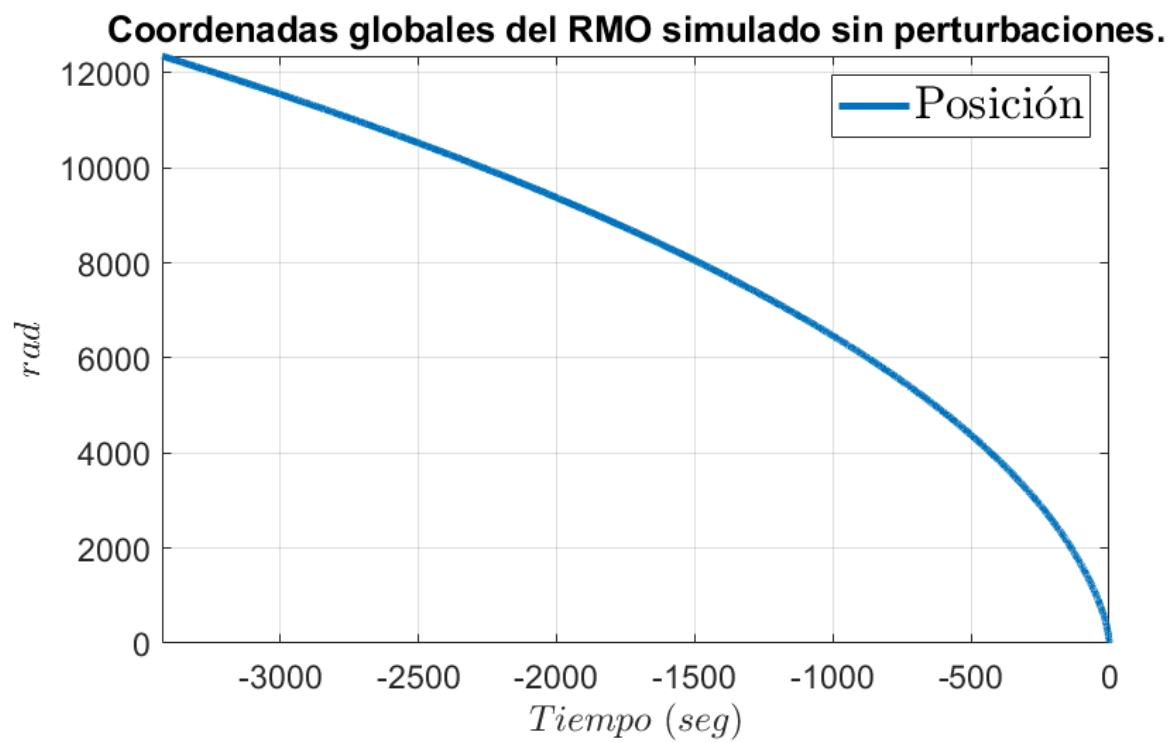


Figura 5.6: Posición del RMO respecto a sus coordenadas globales al aplicarle una entrada del PWM del 100% durante 7 segundos.

Capítulo 6

Desarrollo de algoritmos de control para el control de velocidad de las ruedas

6.1. Propuesta de un algoritmo de control en base a retroalimentación de estado

Cada uno de los motores será controlado por su propia señal de control, ya que únicamente se analizan los movimientos típicos del RMO. Se debe tomar en cuenta que las velocidades tienen diferentes direcciones según el movimiento especificado. Es decir, se puede especificar una única velocidad de referencia y, según la dirección a la cual se desee mover el vehículo, la referencia en cada rueda cambiará, según la siguiente tabla:

Dirección de movimiento	ω_{ref1}	ω_{ref2}	ω_{ref3}	ω_{ref4}
↑	$= \omega_{ref}$	$= \omega_{ref}$	$= \omega_{ref}$	$= \omega_{ref}$
↗	$= \omega_{ref}$	$= 0 * \omega_{ref}$	$= 0 * \omega_{ref}$	$= \omega_{ref}$
→	$= \omega_{ref}$	$= -\omega_{ref}$	$= -\omega_{ref}$	$= \omega_{ref}$
↘	$= 0 * \omega_{ref}$	$= -\omega_{ref}$	$= -\omega_{ref}$	$= 0 * \omega_{ref}$
↓	$= -\omega_{ref}$	$= -\omega_{ref}$	$= -\omega_{ref}$	$= -\omega_{ref}$
↙	$= \omega_{ref}$	$= 0 * \omega_{ref}$	$= 0 * \omega_{ref}$	$= -\omega_{ref}$
←	$= -\omega_{ref}$	$= \omega_{ref}$	$= \omega_{ref}$	$= -\omega_{ref}$
↖	$= 0 * \omega_{ref}$	$= \omega_{ref}$	$= \omega_{ref}$	$= 0 * \omega_{ref}$
⌚	$= \omega_{ref}$	$= -\omega_{ref}$	$= \omega_{ref}$	$= -\omega_{ref}$
⌚	$= -\omega_{ref}$	$= \omega_{ref}$	$= -\omega_{ref}$	$= \omega_{ref}$

Tabla 6.1: Tabla de referencias para cada motor según el movimiento deseado.

Para iniciar con la etapa de los controladores se tomó en cuenta primero un controlador con retroalimentación de estado, tomando en cuenta únicamente una función de transferencia promedio para describir el comportamiento de los 4 motores, con la forma:

$$FT_{general} = \frac{369.425}{s + 5.017} \quad (6.1)$$

Si bien las funciones de transferencia obtenidas en la sección anterior pueden ser utilizadas para hacer simulaciones con las diferencias en la dinámica de cada motor, el controlador solo tomará la ecuación 6.1 para agregar simplicidad a la implementación del mismo, pero, en cambio, se tomará enfoque en la robustez de los controladores para compensar por esas dinámicas no modeladas. Para iniciar a declarar el controlador se especifica el objetivo del control, donde se busca que ω_i sea igual a una referencia ω_{ref} , por lo tanto, la dinámica del error se verá descrita como:

$$e_i = \omega_i - \omega_{ref} \quad (6.2)$$

La primera estrategia de control a tomar en cuenta, utiliza la ecuación 6.1 , para obtener una retroalimentación de estado que cancele la dinámica del sistema. Entonces, tomando en cuenta que la dinámica del sistema es:

$$\dot{\omega}_i = -\alpha\omega_i + \beta u_i \quad (6.3)$$

se propone u_i tal que:

$$u_i = \frac{\alpha\omega_{iref}}{\beta} \quad (6.4)$$

A partir de esta ecuación es posible implementar diversas acciones de control para agregar robustez al controlador. Se propone analizar el sistema tomando en cuenta las acciones de control proporcional, integral, de modos deslizantes y combinaciones entre estas. Para el controlador por modos deslizantes se elige como superficie deslizante

$$s_i = e_i$$

por la cual la acción de control por modos deslizantes controlada por una ganancia, se

describe de la siguiente manera:

$$u_{MD} = k_{MD}(\text{sign}(s_i))$$

Agregando las acciones de control proporcional, integral y por modos deslizantes con retroalimentación de estado, la ecuación de control completa, tomará la siguiente forma:

$$u_i = \frac{\alpha\omega_{iref} - k_p e_i - k_i z_i - k_{MD}(\text{sign}(e_i))}{\beta} \quad (6.5)$$

donde:

$$\dot{z}_i = e_i$$

Una vez obtenida la ecuación de control 6.5 se realizó un programa de simulación del controlador, así como el programa en LabVIEW para realizar los experimentos del controlador con el sistema real.

6.2. Simulaciones del RMO aplicando diferentes ganancias de control.

Una vez establecida la ecuación de control 6.5, se pueden variar las ganancias del controlador, para potenciar los efectos de la acción proporcional, integral o por modos deslizantes, e incluso hacer combinaciones entre ellas. Se realizaron 8 experimentos diferentes para analizar la influencia de las diferentes acciones de control sobre el RMO y encontrar la o las combinaciones de acciones de control más adecuadas para el sistema. Los 8 experimentos muestran los comportamientos del RMO sin ningún controlador, con acción proporcional, con acción integral, con controlador por modos deslizantes, con acción proporcional-integral, con acción proporcional con modos deslizantes, con acción integral con modos deslizantes y con acción proporcional-integral con modos deslizantes. Las ganancias para cada acción se muestran en la tabla 6.2.

Modificando el programa mostrado en la figura 5.3 , se puede crear un programa para simular el comportamiento del RMO con un controlador en cada uno de los motores,

Experimentos	k_p	k_i	k_{MD}
Abierto	0	0	0
P	300	0	0
I	0	20	0
MD	0	0	180
PI	250	50	0
PMD	200	0	50
IMD	0	100	60
PIMD	100	80	50

Tabla 6.2: Tabla de ganancias de acciones de control aplicadas a los motores del RMO.

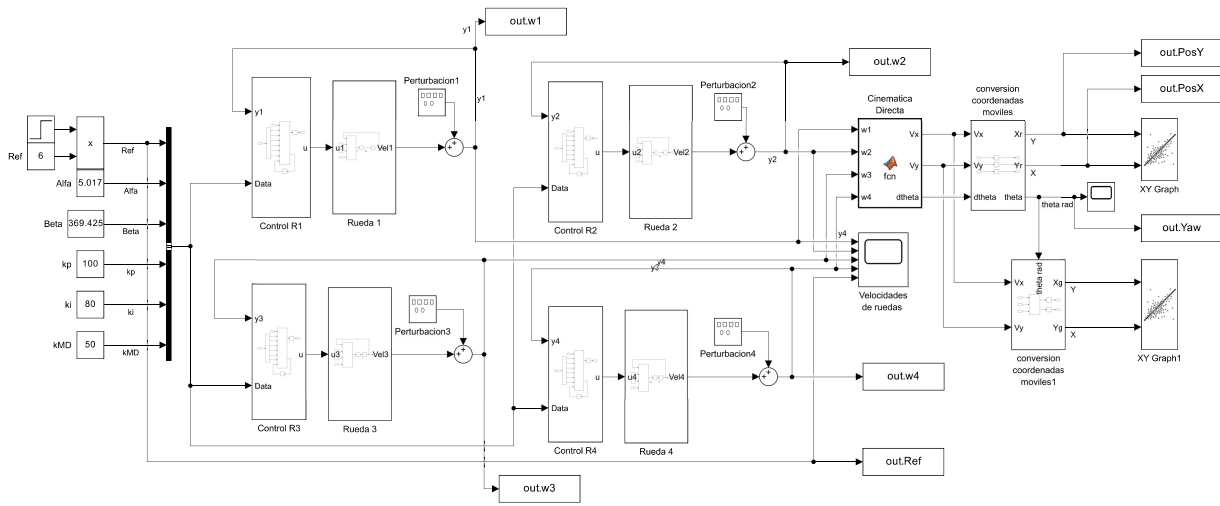


Figura 6.1: Programa de Simulink para simular el comportamiento del RMO con controladores individuales en cada rueda.

tal como se muestra en la figura 6.1. En este programa se realizaron las simulaciones del sistema según las ganancias de control de la tabla 6.2, aplicando una señal de referencia de $6 \frac{rad}{seg}$ y perturbaciones senoidales diferentes en cada uno de los motores durante 200 segundos y se exportaron los datos de entrada y de salida de cada experimento a Matlab para poder comparar el ángulo de orientación θ y la trayectoria simulada según las coordenadas móviles del RMO, como se muestra en la figura 6.2. Tomando en cuenta que los controladores son aplicados a cada motor de manera individual, sin tomar en consideración una retroalimentación de la cinemática del sistema, ni algún tipo de comunicación entre las otras ruedas, es lógico que las estrategias de control que no logren hacer que los 4 motores alcancen el estado estable en el mismo valor, obtendrán por consecuencia una variación mayor, tanto en el ángulo de orientación, como en la trayectoria recorrida del

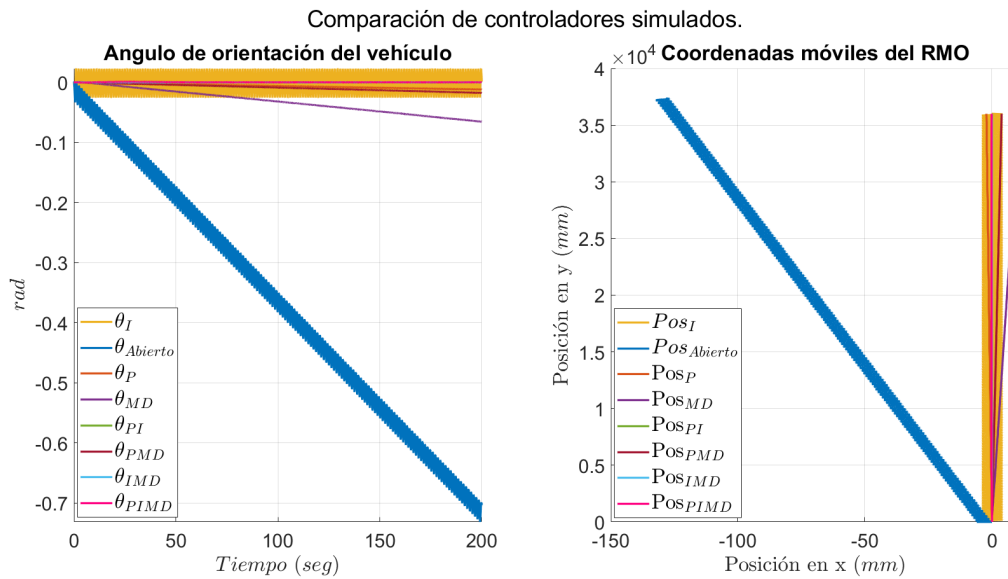


Figura 6.2: Angulo de orientación y trayectoria del RMO de las 8 simulaciones con movimientos hacia adelante.

RMO. El parámetro estadístico de error utilizado para elegir las mejores estrategias de control, es la raíz del error cuadrático medio o RMSE por sus siglas en inglés, acrónimo por el cual se le llamará a lo largo de la tesis. Que se utiliza para medir el error entre 2 conjuntos de datos, dándole mayor peso a los errores mayores. Se calculó el RMSE de las velocidades angulares de los 4 motores $\omega_1, \omega_2, \omega_3, \omega_4$, el ángulo de orientación del vehículo θ y la posición X_R y/o Y_R según sea el caso. Eliminando de la gráfica las señales que tienen una mayor desorientación se obtiene la figura 6.3, en donde se muestra que, mínimo en la simulación, las estrategias de control más aptas para implementar al sistema, son aquellas que incluyen una acción integral, ya que esta tiene como objetivo eliminar el error en estado estable, pero como se puede observar, en esta simulación de 200 segundos, la estrategia con acciones proporcional-integral con modos deslizantes, tardó aproximadamente 50 segundos en estabilizarse, lo cual no es deseable en aplicaciones reales, por otro lado, la estrategia proporcional-integral mantiene oscilaciones alrededor de los valores deseados, desde el inicio de la simulación, pero las sobre elongaciones son bastante amplias, considerando que no se atenúan en ningún momento, mientras la estrategia de acción integral con modos deslizantes parece ser la más útil en la simulación, en la vida real esto no sucede, ya que la acción integral puede generar una desorientación muy grande con cada cambio de

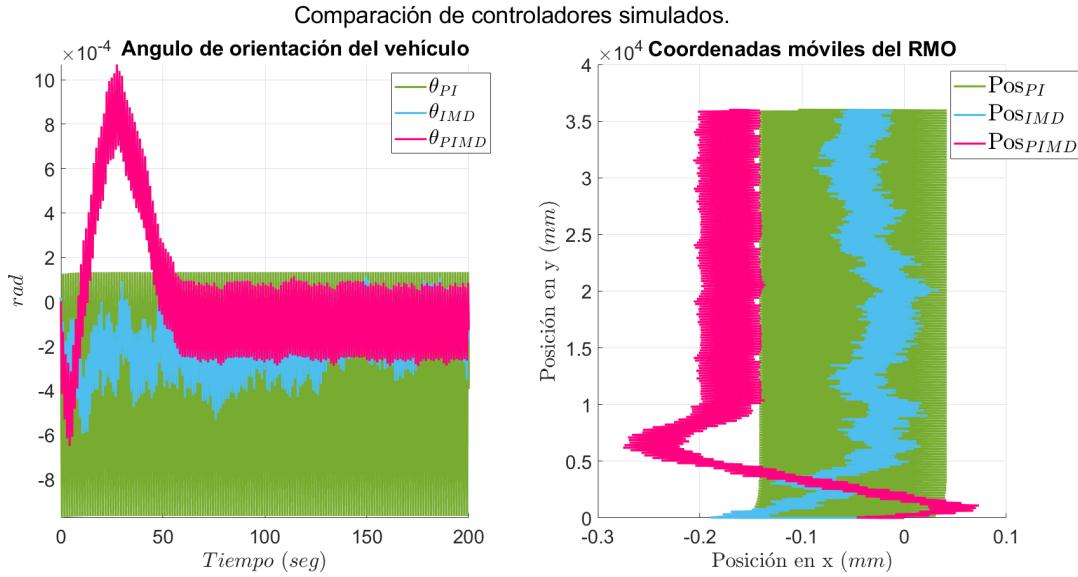


Figura 6.3: Ángulo de orientación y trayectoria del RMO de las simulaciones con los mejores controladores con movimiento hacia adelante.

referencia. Nótese que las simulaciones son de larga duración, lo cual puede mostrar comportamientos y ventajas diferentes de las simulaciones de corta duración, y debe tomarse en cuenta según las necesidades de los experimentos.

	$\omega_{ref1}(\frac{rad}{seg})$	$\omega_{ref2}(\frac{rad}{seg})$	$\omega_{ref3}(\frac{rad}{seg})$	$\omega_{ref4}(\frac{rad}{seg})$	$\theta(rad)$	$X_R(mm)$	$Y_R(mm)$
P	0.0348	0.0412	0.0463	0.0414	0.0071	1.29	1.72
I	0.4391	0.8492	1.9386	0.3922	0.0164	2.93	3.13
MD	0.1139	0.1136	0.1126	0.1078	0.0375	6.17	2.49
PI	0.0357	0.0444	0.0511	0.0444	0.0004	0.08	0.13
PMD	0.0466	0.0464	0.0467	0.0454	0.0103	2.16	0.84
IMD	0.0710	0.0698	0.0702	0.0707	0.0002	0.0518	0.97
PIMD	0.0457	0.0452	0.0467	0.0449	0.0002	0.16	0.36

Tabla 6.3: Tabla de RMSE según las diferentes acciones de control aplicadas a los motores del RMO simulados, con movimiento hacia adelante.

De la misma manera se realizaron simulaciones para analizar los mismos controladores aplicados al RMO, con la misma duración y las mismas ganancias de control, para movimientos transversales hacia la derecha y movimientos en diagonal. El ángulo de orientación y la trayectoria simulada del RMO con los diferentes controladores se visualizan en la figura 6.4. Para poder apreciar los resultados ante las acciones de control más funcionales, observe la figura 6.5. Para apreciar mejor el desempeño de cada una de las acciones de

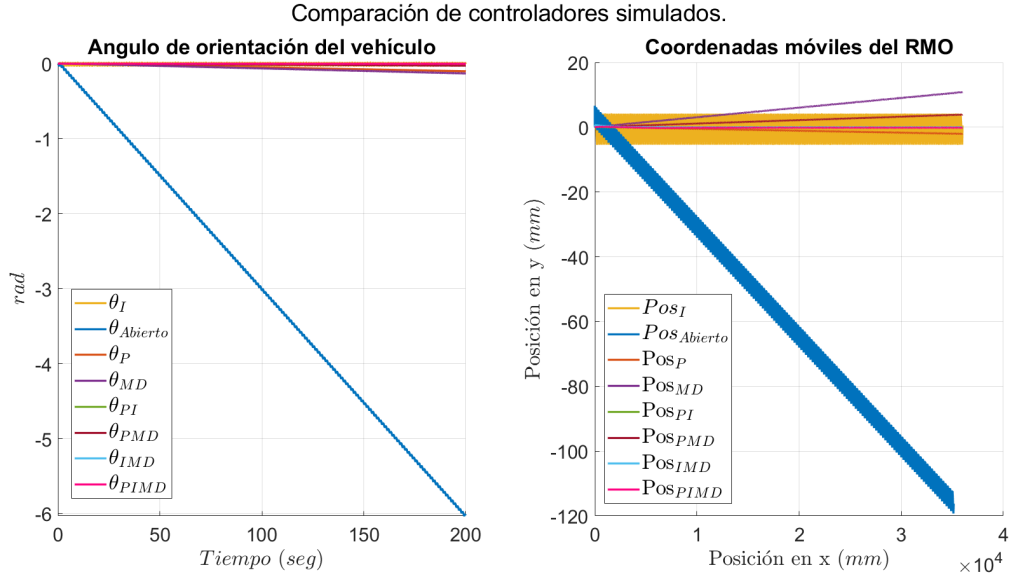


Figura 6.4: Angulo de orientación y trayectoria del RMO de las 8 simulaciones con movimiento transversal hacia la derecha.

control y sus combinaciones, observar la tabla de RMSE del RMO en movimientos transversales hacia la derecha 6.4. Los resultados de la simulación del RMO con movimiento

	$\omega_{ref1}(\frac{rad}{seg})$	$\omega_{ref2}(\frac{rad}{seg})$	$\omega_{ref3}(\frac{rad}{seg})$	$\omega_{ref4}(\frac{rad}{seg})$	$\theta(rad)$	$X_R(mm)$	$Y_R(mm)$
P	0.0348	0.0421	0.0472	0.0414	0.0590	1.88	1.15
I	0.4391	0.8566	1.9455	0.3922	0.0181	3.10	3.06
MD	0.1139	0.1134	0.1136	0.1078	0.0752	2.44	6.28
PI	0.0357	0.0453	0.0519	0.0444	0.0031	0.2057	0.0943
PMD	0.0466	0.0473	0.0477	0.0454	0.0161	0.8627	2.25
IMD	0.0710	0.0776	0.0783	0.0707	0.0015	1.10	0.01061
PIMD	0.0457	0.0463	0.0469	0.0449	0.0018	0.3860	0.1677

Tabla 6.4: Tabla de RMSE según las diferentes acciones de control aplicadas a los motores del RMO simulados con movimiento transversal hacia la derecha.

en diagonal con las diferentes ganancias de control se pueden observar en la figura 6.6. Al igual que en las simulaciones pasadas, el comportamiento del RMO perturbado sin controladores muestra una gran desorientación, por lo cual se hizo la figura 6.7, donde se muestra el comportamiento únicamente con las mejores respuestas de los controladores propuestos. Para apreciar mejor el desempeño de cada una de las acciones de control y sus combinaciones, observar la tabla de RMSE del RMO en movimientos diagonales 6.5. Incluso ante diferentes direcciones se puede observar que los mejores esquemas de control

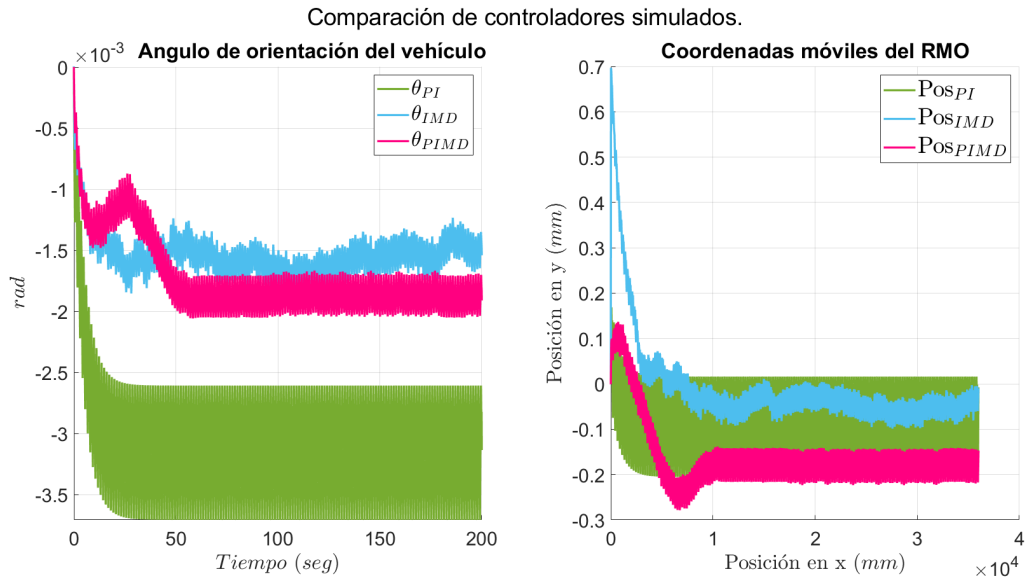


Figura 6.5: Angulo de orientación y trayectoria del RMO de las simulaciones con los mejores controladores con movimiento transversal hacia la derecha.

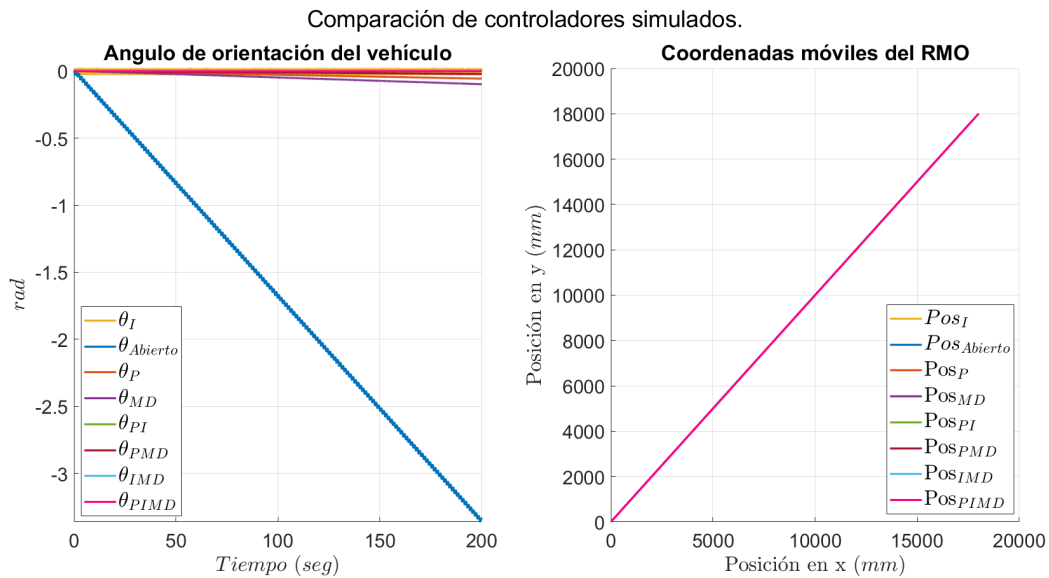


Figura 6.6: Angulo de orientación y trayectoria del RMO de las 8 simulaciones con movimiento en diagonal.

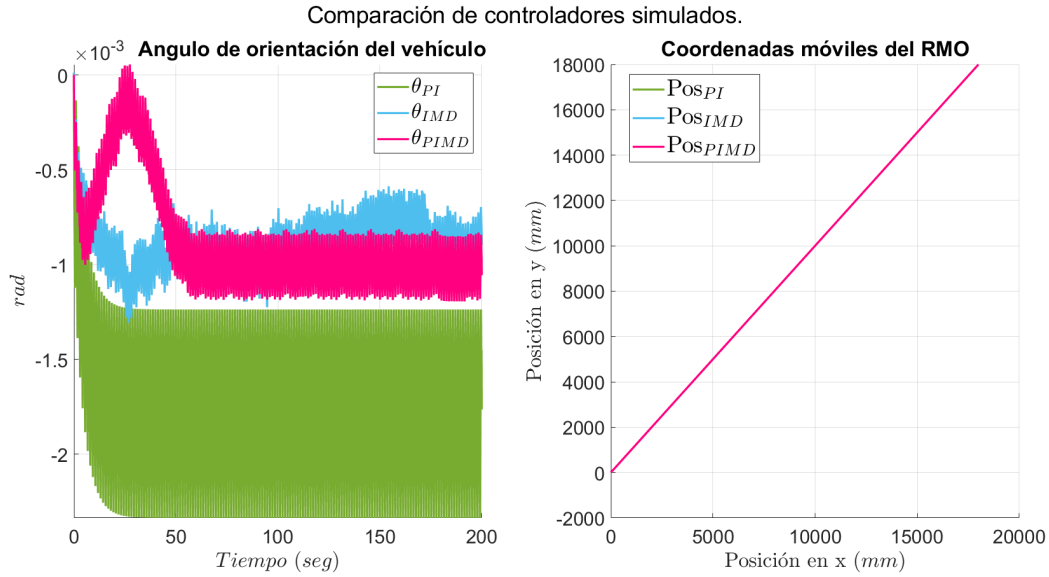


Figura 6.7: Angulo de orientación y trayectoria del RMO de las simulaciones con los mejores controladores con movimiento en diagonal.

	$\omega_{ref1}(\frac{rad}{seg})$	$\omega_{ref2}(\frac{rad}{seg})$	$\omega_{ref3}(\frac{rad}{seg})$	$\omega_{ref4}(\frac{rad}{seg})$	$\theta(rad)$	$X_R(mm)$	$Y_R(mm)$
P	0.0348	0.0255	0.0331	0.0414	0.033	1.53	1.41
I	0.4391	0.8420	1.9375	0.3922	0.0169	2.98	3.07
MD	0.1139	0.1040	0.1045	0.1078	0.0565	4.06	4.32
PI	0.0357	0.0305	0.0396	0.0444	0.0017	0.1269	0.1175
PMD	0.0466	0.0336	0.0341	0.0454	0.0132	1.2982	1.32
IMD	0.0710	0.0302	0.1045	0.0707	0.0088	0.51	0.4998
PIMD	0.0457	0.0314	0.0323	0.0449	0.0092	0.2417	0.2307

Tabla 6.5: Tabla de RMSE según las diferentes acciones de control aplicadas a los motores del RMO simulados con movimiento diagonal.

son aquellos en los que los controladores no utilizan una única acción de control. Tomando en cuenta que las simulaciones no modelan todos los parámetros y perturbaciones externas que influyen sobre el sistema, es necesario hacer una experimentación sobre el sistema real, para comprobar los comportamientos obtenidos en las simulaciones.

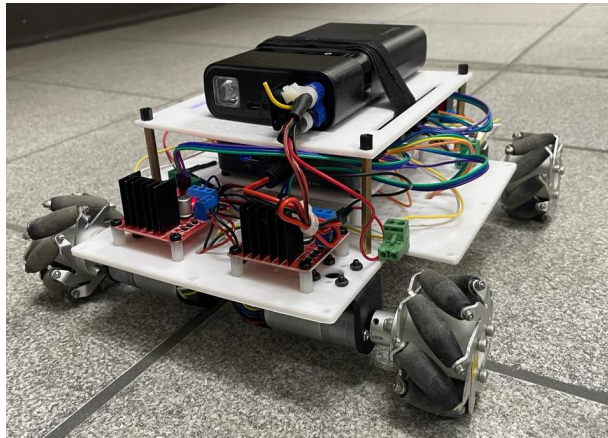


Figura 6.8: Imagen real del RMO fabricado.



Figura 6.9: Superficie de concreto plano no pulido, con irregularidades controladas, sin hoyos profundos, ni sustancias no reconocidas.

6.3. Experimentación del RMO sobre concreto aplicando diferentes ganancias de control.

Para la etapa de experimentación de un RMO en ambientes reales, se utilizó el RMO mostrado en la figura 6.8, sobre una superficie de concreto plano, la cual es mostrada en la figura 6.9, sin hoyos donde se pueda quedar estancada una rueda, sin separaciones entre bloques, ni losetas, ni sustancias líquidas inapropiadas para el uso del RMO. Para esta etapa se realizaron de igual manera 8 experimentos en la dirección adelante, transversal hacia la derecha y diagonal utilizando la misma tabla de ganancias 6.2 , a excepción del controlador por modos deslizantes puro, el cual se le aplicó una $k_{MD} = 260$, para el movimiento

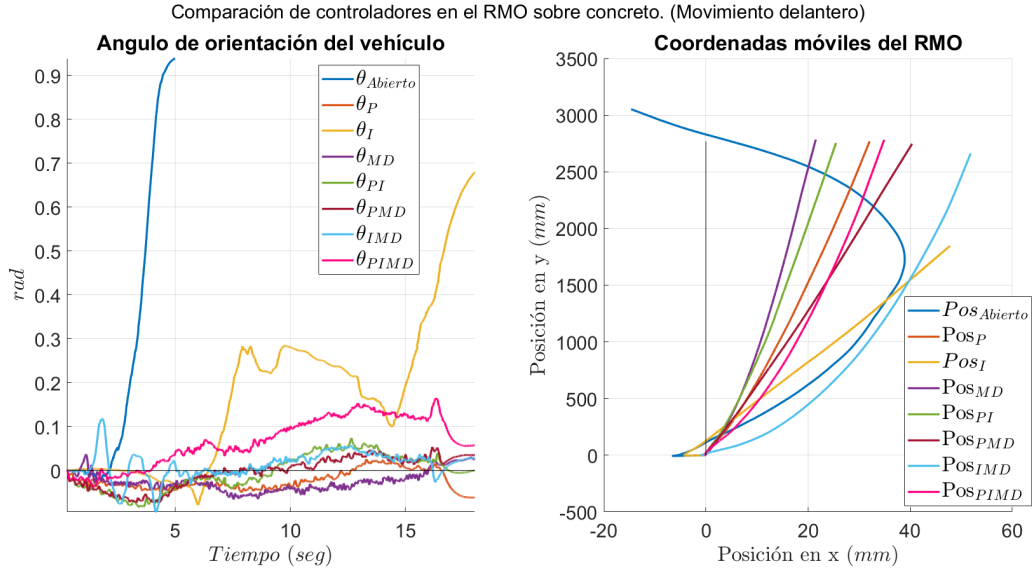


Figura 6.10: Angulo de orientación y trayectoria del RMO sobre concreto, de los 8 experimentos de las diferentes estrategias de control, realizando un movimiento delantero.

transversal hacia la derecha y una $k_{MD} = 240$ para el movimiento en diagonal. Todos los experimentos se realizaron aplicando una referencia de $6 \frac{rad}{seg}$, durante 16 segundos y un estado de reposo al iniciar el experimento y 2 segundos antes de finalizar el experimento. Los resultados del ángulo de orientación y la trayectoria del RMO sobre concreto, aplicando las 8 diferentes configuraciones de control, realizando un movimiento delantero se muestran en la figura 6.10, a pesar de que los resultados son mas fáciles de visualizar que en la simulación, para comparar los resultados de los mejores resultados obtenidos, consulte la figura 6.11. Para apreciar mejor el desempeño de cada una de las acciones de control y sus combinaciones, observar la tabla de RMSE del RMO en movimientos diagonales 6.6.

Los mismos experimentos se realizaron sobre el sistema también provocando un movimiento transversal hacia la derecha, aplicando una referencia de $6 \frac{rad}{seg}$ a la rueda 1 y 4, y una referencia de $-6 \frac{rad}{seg}$, durante 16 segundos y un estado de reposo al iniciar el experimento y 2 segundos antes de finalizar el experimento. Los resultados del ángulo de orientación y la trayectoria del RMO sobre concreto, aplicando las 8 diferentes configuraciones de control, realizando un movimiento transversal hacia la derecha se muestran en la figura 6.12, a pesar de que los resultados son mas fáciles de visualizar que en la simulación, para comparar los resultados de los mejores resultados obtenidos, consulte la figura 6.13. Para

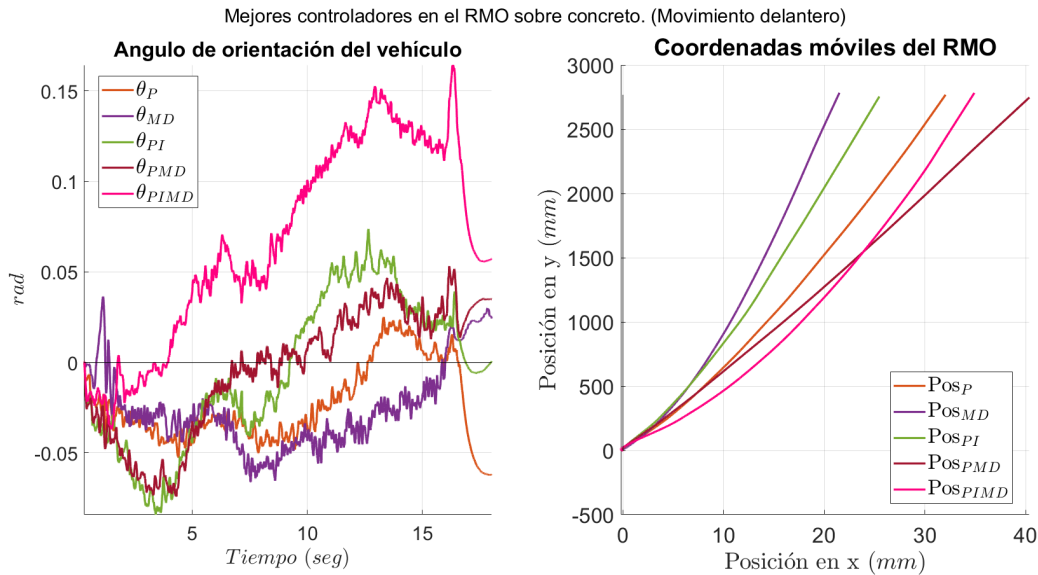


Figura 6.11: Angulo de orientación y trayectoria del RMO sobre concreto, de las mejores estrategias de control aplicadas, realizando un movimiento delantero.

apreciar mejor el desempeño de cada una de las acciones de control y sus combinaciones, observar la tabla de RMSE del RMO en movimientos diagonales 6.7. Además, se realizaron los 8 experimentos sobre el sistema provocando un movimiento diagonal hacia delante a la derecha, aplicando una referencia de $6 \frac{rad}{seg}$ a la rueda 1 y 4, y una referencia de $0 \frac{rad}{seg}$, durante 16 segundos y un estado de reposo al iniciar el experimento y 2 segundos antes de finalizar el experimento. Los resultados del ángulo de orientación y la trayectoria del RMO sobre concreto, aplicando las 8 diferentes configuraciones de control, realizando un movimiento diagonal hacia adelante a la derecha se muestran en la figura 6.12, a pesar de que los resultados son mas fáciles de visualizar que en la simulación, para comparar los resultados de los mejores resultados obtenidos, consulte la figura 6.13. Para apreciar mejor el desempeño de cada una de las acciones de control y sus combinaciones, observar la tabla de RMSE del RMO en movimientos diagonales 6.7.

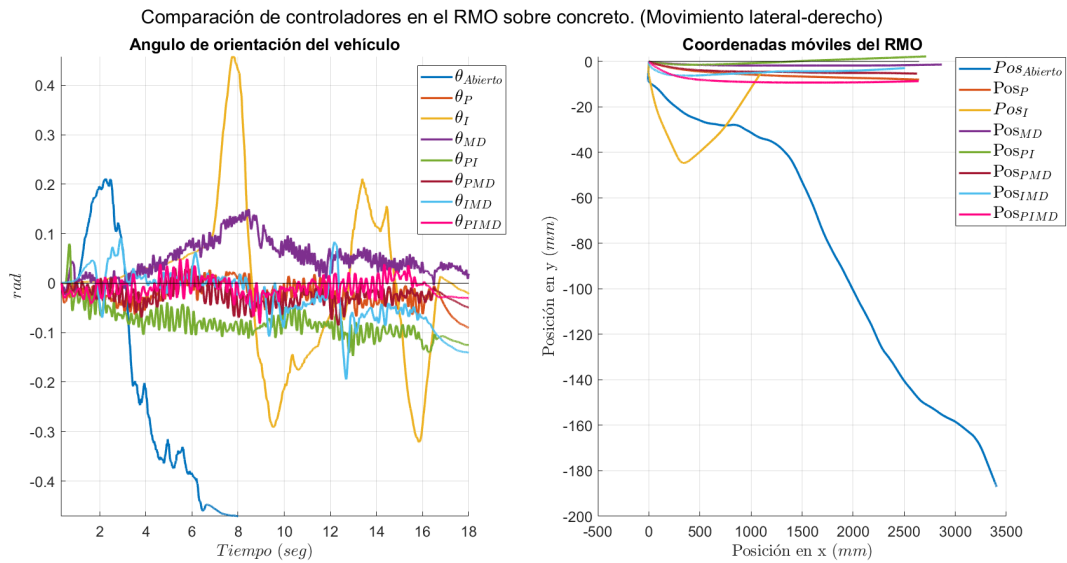


Figura 6.12: Angulo de orientación y trayectoria del RMO sobre concreto, de los 8 experimentos de las diferentes estrategias de control, realizando un movimiento transversal hacia la derecha.

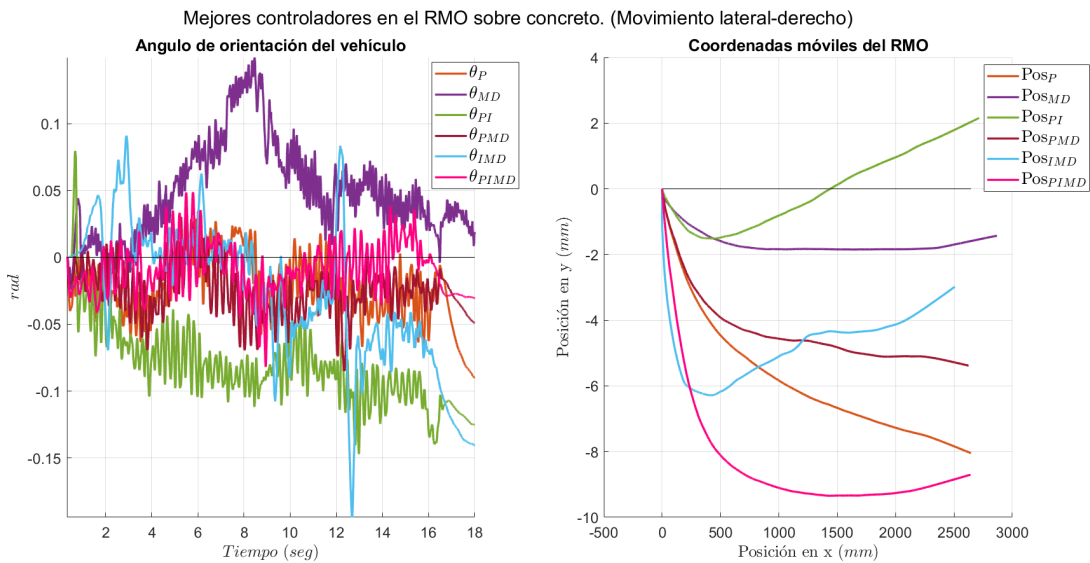


Figura 6.13: Angulo de orientación y trayectoria del RMO sobre concreto, de las mejores estrategias de control aplicadas, realizando un movimiento transversal hacia la derecha.

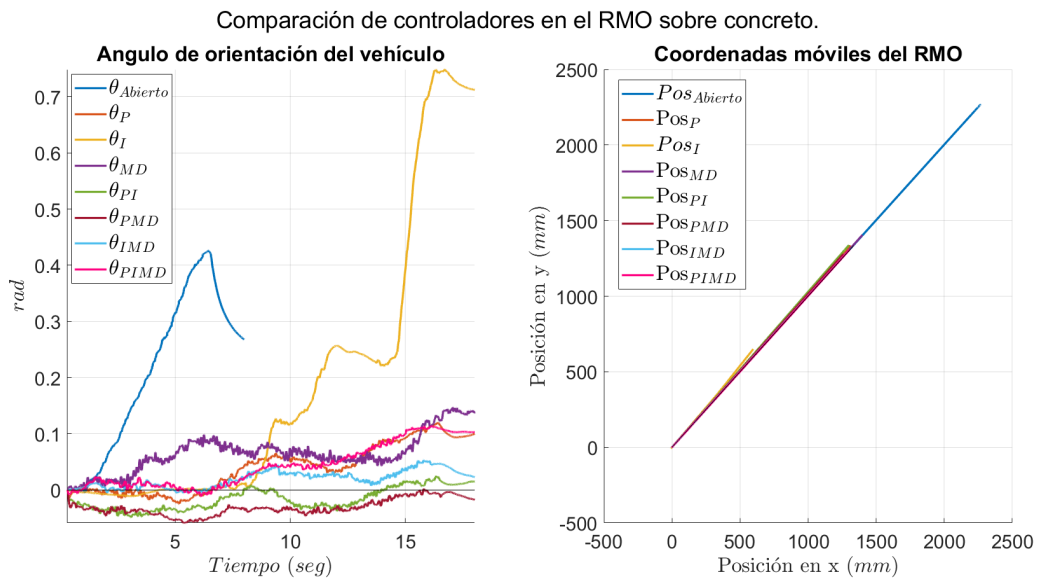


Figura 6.14: Angulo de orientación y trayectoria del RMO sobre concreto, de los 8 experimentos de las diferentes estrategias de control, realizando un movimiento diagonal.

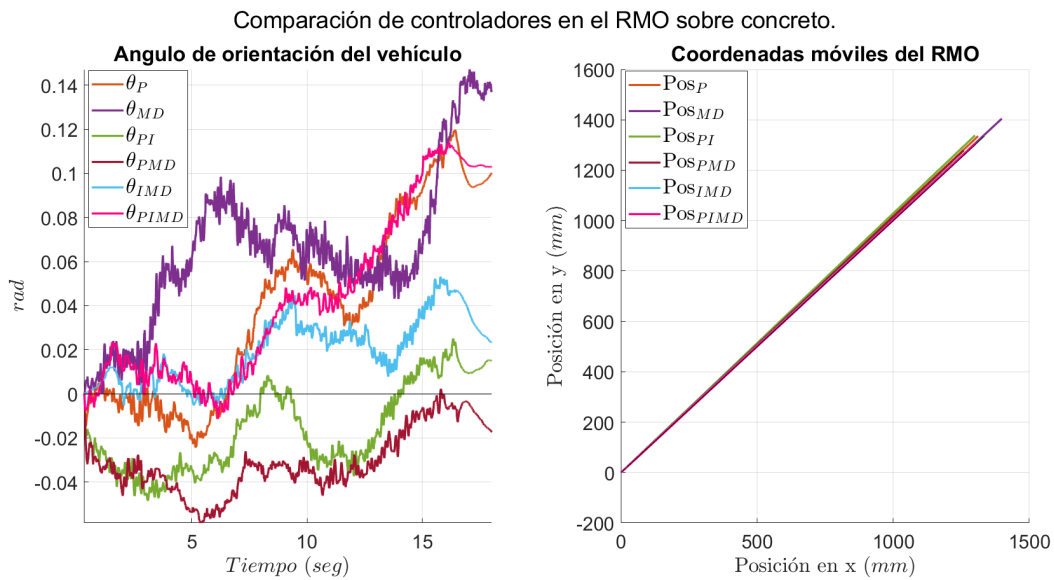


Figura 6.15: Angulo de orientación y trayectoria del RMO sobre concreto, de las mejores estrategias de control aplicadas, realizando un movimiento diagonal.

	$\omega_{ref1}(\frac{rad}{seg})$	$\omega_{ref2}(\frac{rad}{seg})$	$\omega_{ref3}(\frac{rad}{seg})$	$\omega_{ref4}(\frac{rad}{seg})$	$\theta(rad)$	$X_R(mm)$
Abierto	24.19	21.78	24.41	21.97	0.45	20.47
P	0.268	0.44	0.39	0.39	0.0318	15.62
I	7.08	6.26	6.75	6.61	0.2498	17.48
MD	0.51	0.64	0.58	0.65	0.0342	10.89
PI	0.28	0.41	0.37	0.37	0.0408	12.35
PMD	0.23	0.44	0.38	0.37	0.0349	18.6
IMD	1.78	2	1.66	2.21	0.0334	28.85
PIMD	0.42	0.56	0.53	0.51	0.0868	18.02

Tabla 6.6: Tabla de RMSE del RMO en movimiento delantero sobre concreto según las diferentes acciones de control aplicadas a los motores del RMO.

	$\omega_{ref1}(\frac{rad}{seg})$	$\omega_{ref2}(\frac{rad}{seg})$	$\omega_{ref3}(\frac{rad}{seg})$	$\omega_{ref4}(\frac{rad}{seg})$	$\theta(rad)$	$Y_R(mm)$
Abierto	13.87	16.03	17.27	14.10	0.3053	113.16
P	0.5856	0.6142	0.5766	0.6	0.0335	5.07
I	5.3	4.59	4.822	5.18	0.1516	22.71
MD	0.2558	0.2911	1.11	1.12	0.0610	1.46
PI	0.6348	0.6414	0.7538	0.7385	0.0878	1.05
PMD	0.6711	0.6909	0.6595	0.6883	0.0313	3.82
IMD	1.79	1.87	1.7443	1.81	0.0605	4.57
PIMD	0.708	0.708	0.6788	0.7437	0.0211	7.36

Tabla 6.7: Tabla de RMSE del RMO en movimiento lateral sobre concreto según las diferentes acciones de control aplicadas a los motores del RMO.

	$\omega_{ref1}(\frac{rad}{seg})$	$\omega_{ref2}(\frac{rad}{seg})$	$\omega_{ref3}(\frac{rad}{seg})$	$\omega_{ref4}(\frac{rad}{seg})$	$\theta(rad)$	$X_R = Y_R(mm)$
Abierto	20	0.0251	0.0962	19.48	0.2526	2.97
P	0.52	0.0812	0.1093	0.6019	0.0575	10.39
I	5.78	0.0828	1.15	5.22	0.3076	14.70
MD	0.3776	0.0998	0.1057	0.4339	0.0740	2.34
PI	0.5038	0.0936	0.1420	0.5824	0.0252	16.34
PMD	0.6264	0.0867	0.1011	0.7056	0.341	8.45
IMD	1.88	0.0287	0.0740	1.95	0.0246	1.99
PIMD	0.6209	0.0427	0.0698	0.7183	0.0578	0.95

Tabla 6.8: Tabla de RMSE del RMO en movimiento diagonal sobre concreto según las diferentes acciones de control aplicadas a los motores del RMO.

Capítulo 7

Desarrollo de algoritmos de control robusto, de velocidad y orientación del robot, basados en sincronización

Se define una funcional $\Gamma_i(\cdot)$ la cual corresponde a todos los parámetros no modelados en los motores, debido a las no linealidades despreciadas, el modelo no identificado correctamente, desgastes en componentes mecánicos y a las existentes perturbaciones externas no identificadas. Se suma esta $\Gamma_i(\cdot)$ a la dinámica del sistema, la cual asumiendo que no puede ser completamente compensada por un controlador provoca desorientaciones no deseadas en el vehículo, ya que las perturbaciones en cada rueda pueden ser distintas. Por lo tanto se proponen distintas ecuaciones de sincronización, que tengan como objetivo igualar las perturbaciones de una rueda esclavo a las perturbaciones de otra rueda maestro. con el fin de eliminar un error de sincronización propuesto como:

$$e_{i \text{ sincro}} = e_s - e_m \quad (7.1)$$

Para 2 ruedas que avancen hacia la misma dirección. Se propone otra ecuación en donde las ruedas avanzan en dirección opuesta:

$$e_{i \text{ sincro}} = e_s + e_m \quad (7.2)$$

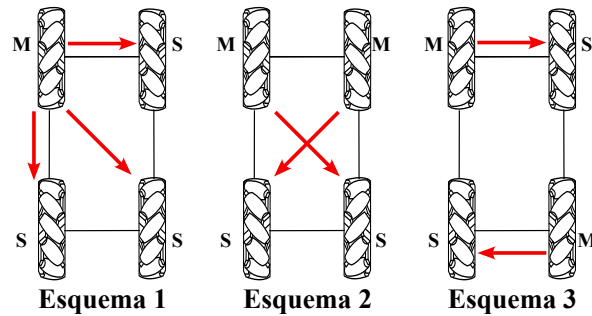


Figura 7.1: Propuestas de esquemas de sincronización entre los 4 motores de un RMO.

Y una última ecuación, donde la rueda esclavo no atiende las perturbaciones de la rueda maestro.

$$e_{i \text{ sincro}} = e_s \quad (7.3)$$

Con estas 3 ecuaciones es posible definir las ecuaciones de sincronización necesarias para todos los movimientos típicos descritos en la tabla 6.1 .

7.1. Esquemas de sincronización

Considerando las ecuaciones (7.1, 7.2,7.3) se proponen distintos esquemas de sincronización, los cuales se analizarán para comparar la eficacia del controlador sobre la velocidad angular de las ruedas y cómo influye esto en la orientación angular del RMO y la trayectoria realizada. Los esquemas analizados en este trabajo de tesis son los mostrados en la figura 7.1. Donde se muestran 3 distintos esquemas. El 1er esquema utiliza una sola rueda maestro y las ruedas faltantes son esclavo. El 2ndo y 3er esquema utilizan 2 ruedas maestro y dos ruedas esclavo, a diferencia que el 2ndo esquema es entre las ruedas pares, y el 3er esquema entre las ruedas impares.

7.2. Simulación de control robusto con diferentes esquemas de sincronización con las diferentes ganancias de control

Se utilizó la misma tabla de ganancias del apartado anterior 6.2, con una ganancia de sincronización $k_{sincro} = 3$ para todas las simulaciones. Se realizaron las mismas simulaciones que en el apartado anterior, tomando únicamente en cuenta las combinaciones de controladores, ya que las acciones de control por sí solas no son suficientemente robustas. Se aplicó una señal de referencia de $6\frac{rad}{seg}$ y perturbaciones senoidales diferentes en cada uno de los motores durante 200 segundos y se compararon las respuestas simuladas entre sus mismas acciones de control, es decir, las acciones Proporcional-Integral sin y con los 3 esquemas de sincronización. Lo mismo se aplicó para las acciones Proporcional con modos deslizantes, Integral con modos deslizantes y Proporcional-Integral con modos deslizantes. En la figura 7.2 se muestra una simulación de 1 segundo con acción proporcional-integral, sin y con el 1er esquema de sincronización, además se hace un acercamiento de las velocidades angulares de los 4 motores del RMO con sus respectivas perturbaciones. En esta imagen se logra apreciar la diferencia entre las velocidades al actuar de manera independiente y al estar influenciadas por el 1er esquema de sincronización donde la rueda 1 es la maestra de las ruedas 2, 3 y 4. Además se logra apreciar cómo influye esto en la trayectoria recorrida por el RMO. Una vez aclarada esta idea, se presentan los controladores simulados con mejor resultados mostrados en la figura 6.3, es decir, las acciones proporcional-integral, integral con modos deslizantes y proporcional-integral con modos deslizantes. Estas simulaciones se volvieron a realizar, esta vez implementando los diferentes esquemas de sincronización mostrados en la figura 7.1 y se compararon los resultados con las mismas acciones de control con el fin de analizar las influencias de los esquemas de sincronización sobre el RMO. Al igual que en las simulaciones anteriores, la duración de la simulación es de 200 segundos, con una referencia de $6\frac{rad}{seg}$. Las simulaciones de sin y con sincronización de las acciones proporcional-integral, integral con modos deslizantes y proporcional-integral con modos deslizantes se muestran en las figuras 7.3, 7.4, 7.5, res-

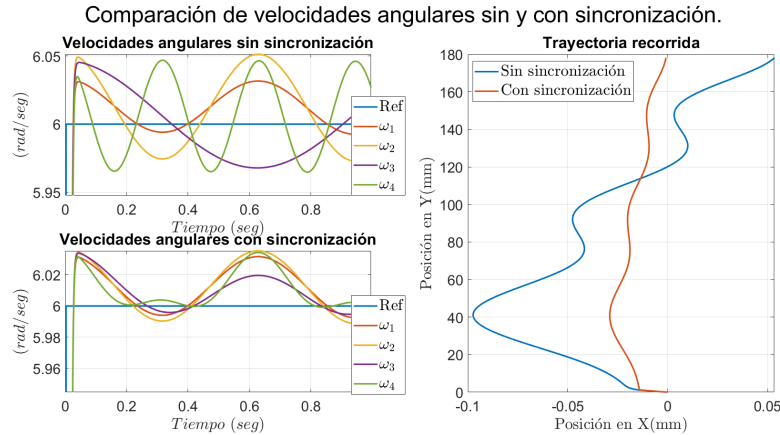


Figura 7.2: Velocidades angulares y trayectorias simuladas del RMO, con acción proporcional-integral sin y con sincronización.

pectivamente. En estas simulaciones se logra observar cómo en algunos casos, como con las acciones proporcional-integral, el uso de sincronización es sin duda una mejor alternativa, reduciendo considerablemente las oscilaciones tanto en la orientación del vehículo como en la trayectoria recorrida. Sin embargo, hay casos donde agregar otra señal de control puede causar que los sistemas tengan más diferencias, provocando que el sistema esté más lejano de sincronizarse que en la versión sin funcional de sincronización.

7.3. Experimentación de control robusto con diferentes esquemas de sincronización con las diferentes ganancias de control y resultados

Para poder analizar a profundidad la influencia de los diferentes esquemas de sincronización se realizaron experimentos del RMO sobre una superficie de concreto plana como la que se muestra en la figura 6.9, se le aplicó una referencia de $6 \frac{rad}{seg}$ a cada una de las ruedas para generar un movimiento frontal durante 16 segundos con un reposo inicial hasta los 0.5 segundos y al final, después del segundo 16, durante 2 segundos. Los 3 esquemas de sincronización mostrados en 7.1 fueron comparados con el sistema sin sincronización ante las acciones de control proporcional-integral, proporcional con modos deslizantes, integral con modos deslizantes y proporcional con modos deslizantes. Estas pueden observarse en

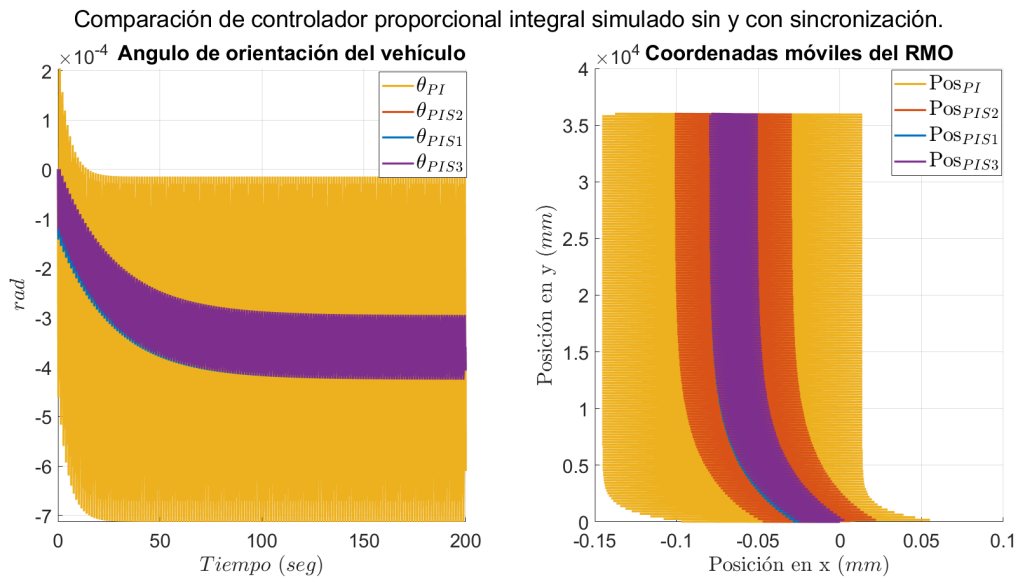


Figura 7.3: Simulación de la acción proporcional integral, sin y con los diferentes esquemas de sincronización.

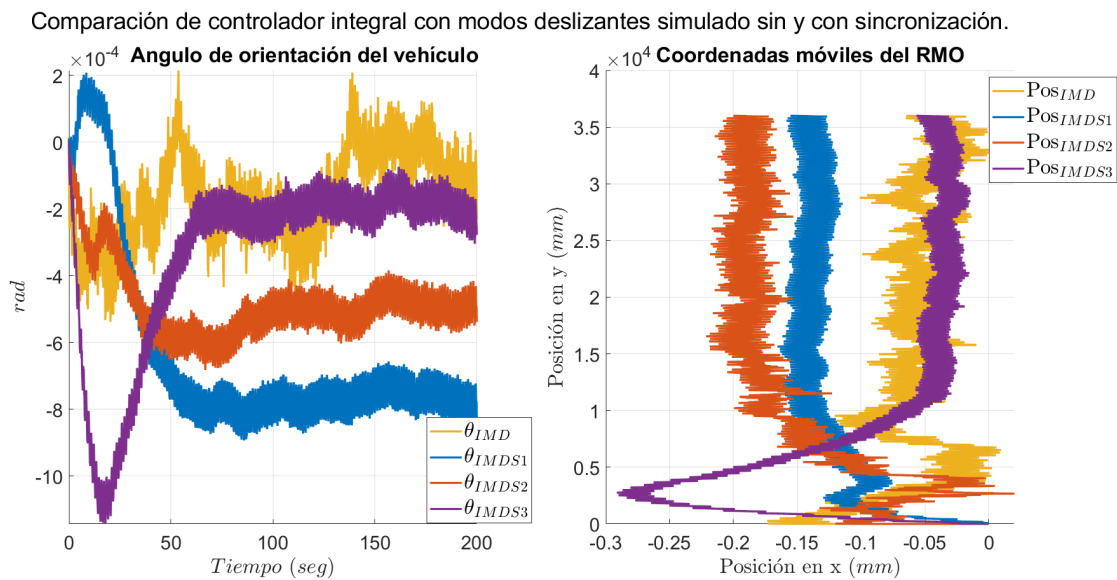


Figura 7.4: Simulación de la acción integral con modos deslizantes, sin y con los diferentes esquemas de sincronización.

Comparación de controlador proporcional integral con modos deslizantes simulado sin y con sincronización.

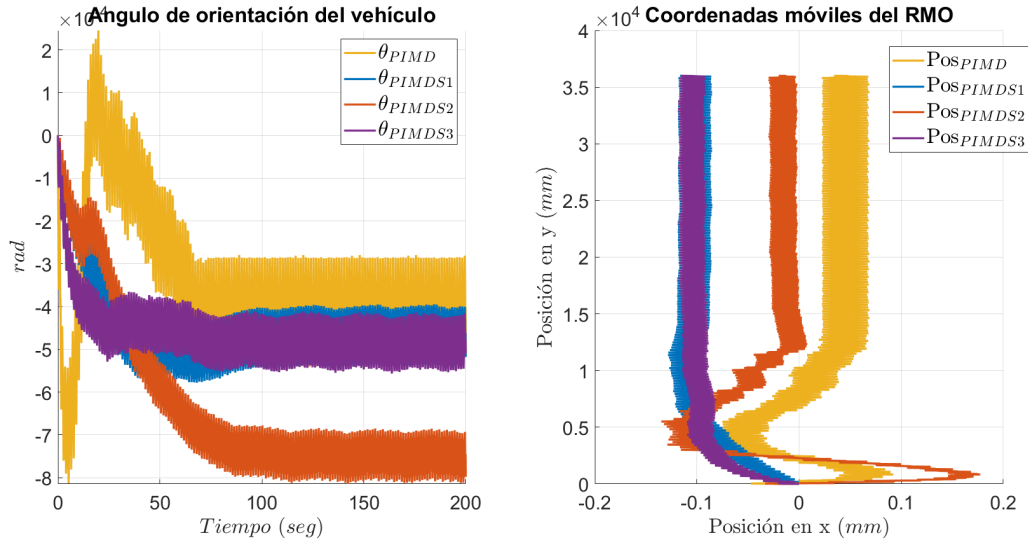


Figura 7.5: Simulación de la acción proporcional integral con modos deslizantes, sin y con los diferentes esquemas de sincronización.

las figuras 7.6,7.7,7.8,7.9, respectivamente. Tras graficar la respuesta del sistema ante las diferentes acciones de control sin y con sincronización, es posible observar cómo en todos los casos para movimiento frontal, la sincronización demuestra ser una herramienta que reduce indudablemente las desorientaciones y los desplazamientos no deseados del vehículo. Esto se demuestra en la tabla 7.1, donde para la mayoría de los casos, los esquemas de sincronización presentaron menor error de orientación y en la trayectoria.

Posteriormente se realizaron experimentos similares con desplazamiento transversal hacia la derecha, es decir, con una referencia de $6\frac{rad}{seg}$ para las ruedas 1 y 4, y una referencia de $-6\frac{rad}{seg}$ para las ruedas 2 y 3. Los experimentos del sistema sin sincronización ante las acciones de control proporcional-integral, proporcional con modos deslizantes, integral con modos deslizantes y proporcional con modos deslizantes, para generar un movimiento transversal hacia la derecha pueden observarse en las figuras 7.10,7.11,7.12,7.13, respectivamente. En este movimiento se obtuvieron varias configuraciones en las cuales si mejoró el resultado al agregar una sincronización, pero no todos los casos fueron iguales, a diferencia del movimiento frontal, hubo algunos experimentos como el esquema de sincronización 1 y 2 con acción proporcional con modos deslizantes. Evidenciando que existen mejores esquemas de sincronización según el movimiento realizado, en este caso para movimiento

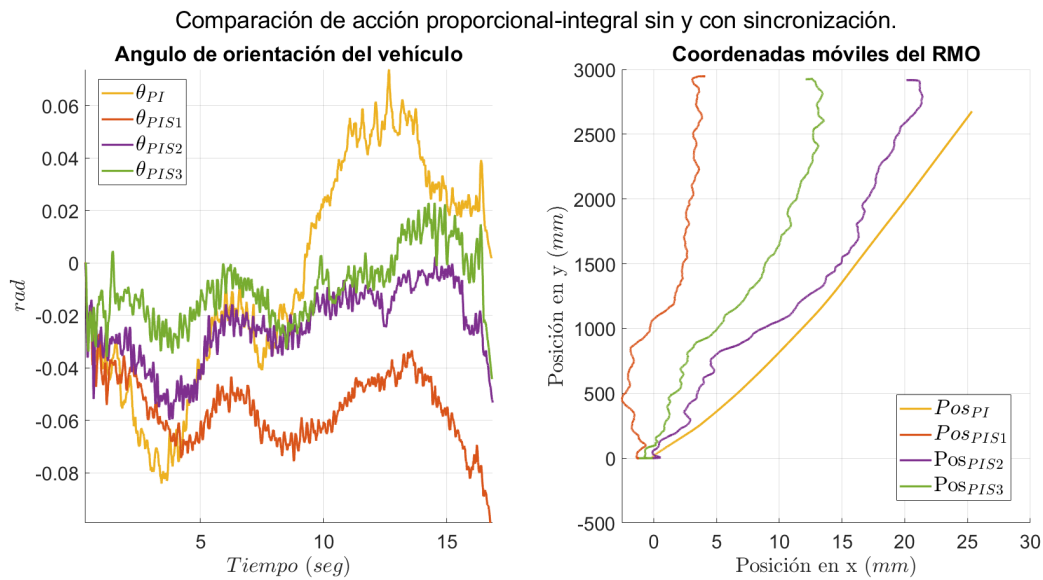


Figura 7.6: Respuesta del RMO sobre concreto con un movimiento frontal, con acción proporcional integral, sin y con los diferentes esquemas de sincronización sobre las ruedas.

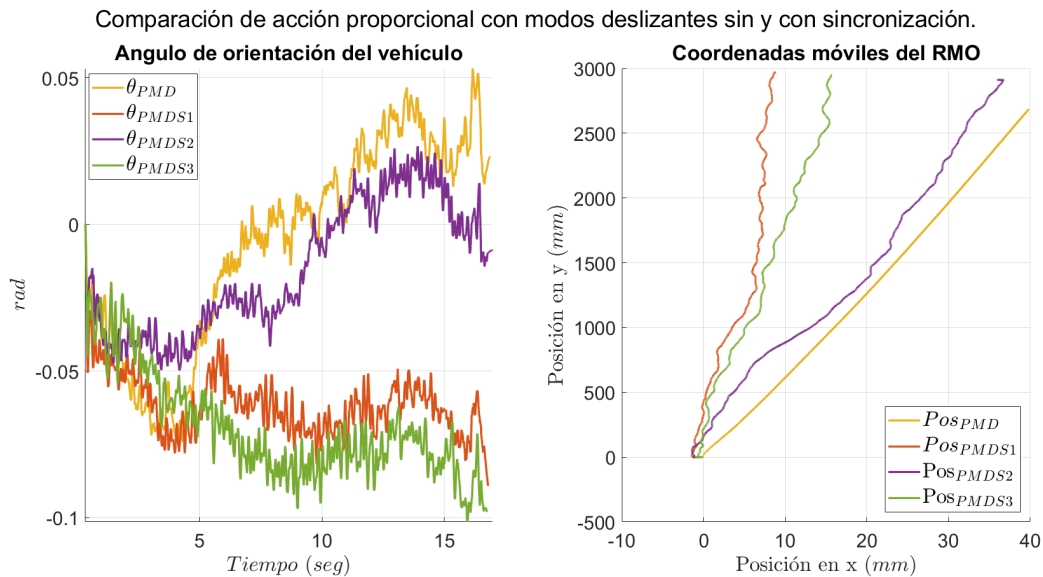


Figura 7.7: Respuesta del RMO sobre concreto con un movimiento frontal, con acción proporcional con modos deslizantes, sin y con los diferentes esquemas de sincronización sobre las ruedas.

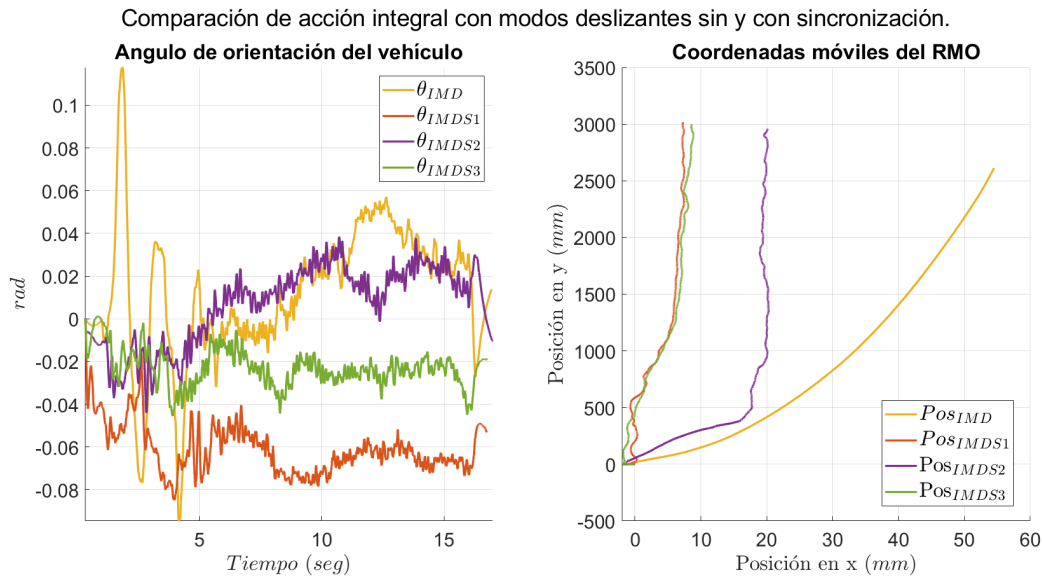


Figura 7.8: Respuesta del RMO sobre concreto con un movimiento frontal, con acción integral con modos deslizantes, sin y con los diferentes esquemas de sincronización sobre las ruedas.

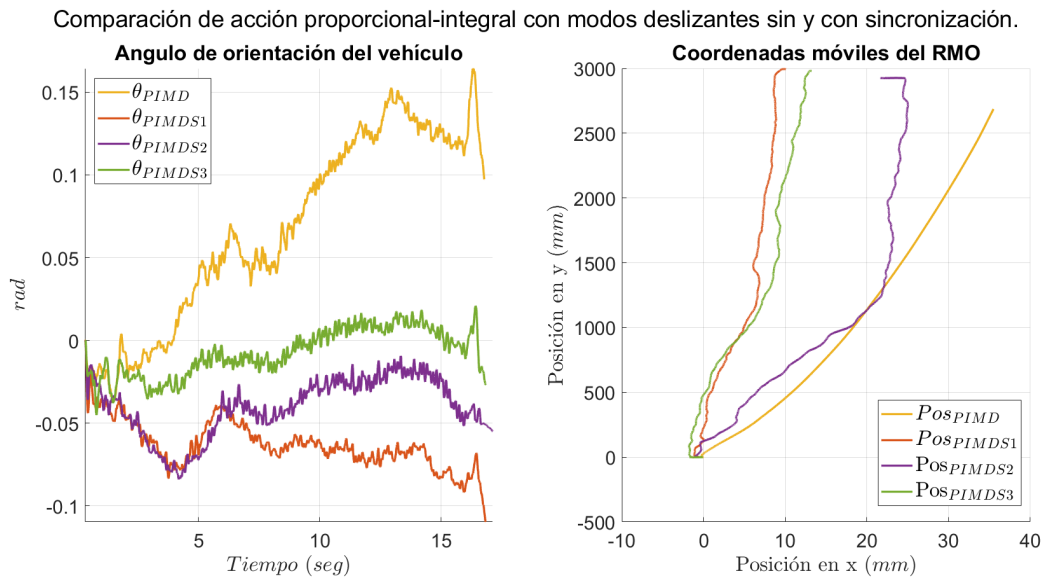


Figura 7.9: Respuesta del RMO sobre concreto con un movimiento frontal, con acción proporcional integral con modos deslizantes, sin y con los diferentes esquemas de sincronización sobre las ruedas.

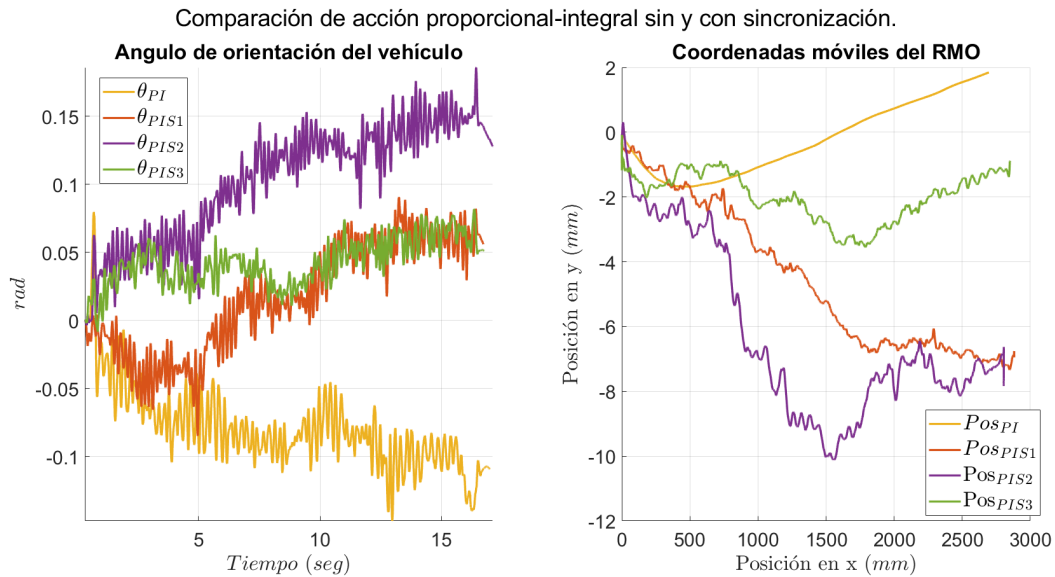


Figura 7.10: Respuesta del RMO sobre concreto con un movimiento transversal hacia la derecha, con acción proporcional integral, sin y con los diferentes esquemas de sincronización sobre las ruedas.

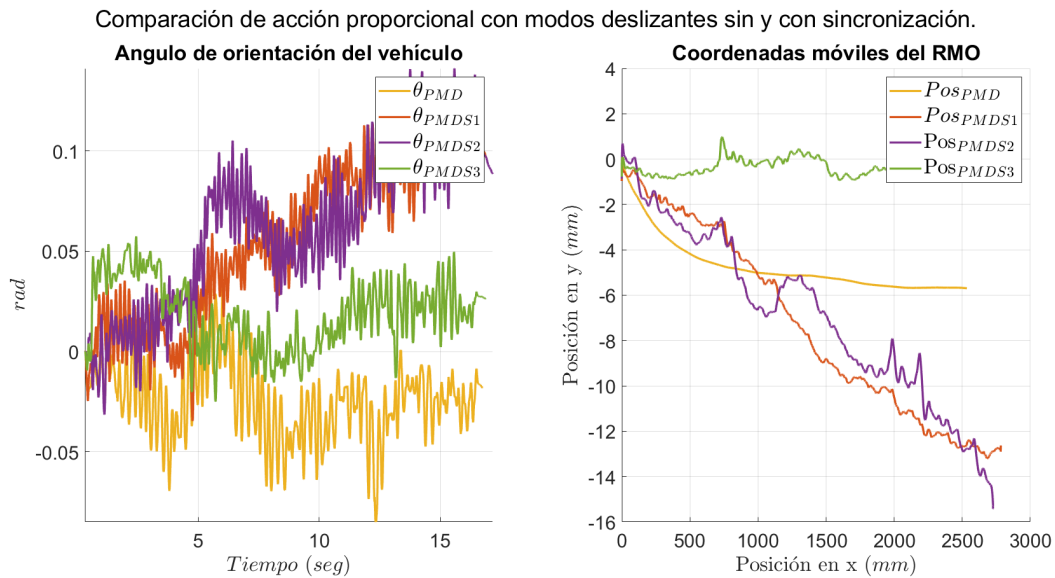


Figura 7.11: Respuesta del RMO sobre concreto con un movimiento transversal hacia la derecha, con acción proporcional con modos deslizantes, sin y con los diferentes esquemas de sincronización sobre las ruedas.

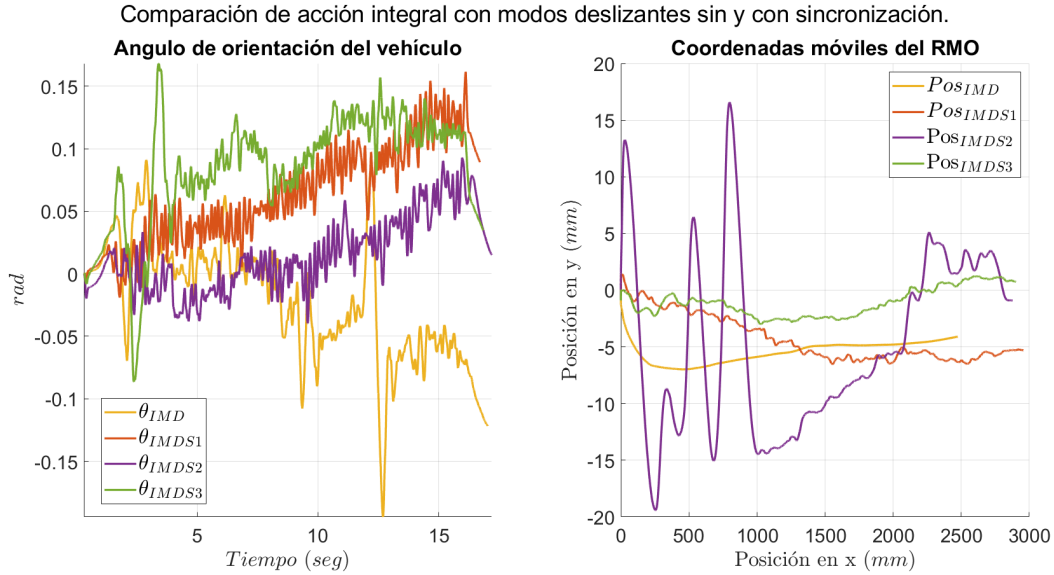


Figura 7.12: Respuesta del RMO sobre concreto con un movimiento transversal hacia la derecha, con acción integral con modos deslizantes, sin y con los diferentes esquemas de sincronización sobre las ruedas.

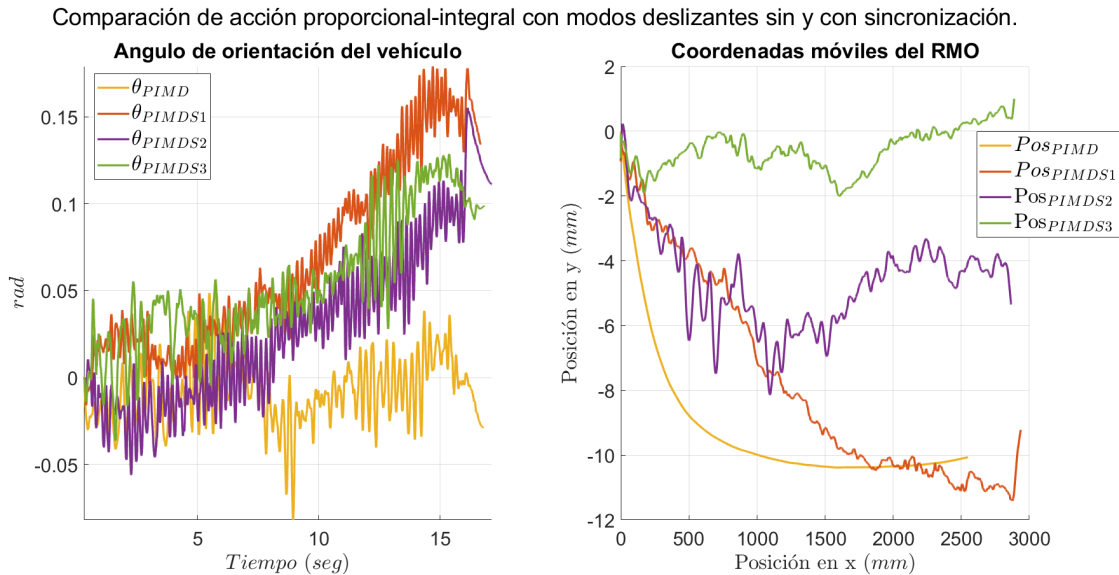


Figura 7.13: Respuesta del RMO sobre concreto con un movimiento transversal hacia la derecha, con acción proporcional integral con modos deslizantes, sin y con los diferentes esquemas de sincronización sobre las ruedas.

	$\omega_{ref1}(\frac{rad}{seg})$	$\omega_{ref2}(\frac{rad}{seg})$	$\omega_{ref3}(\frac{rad}{seg})$	$\omega_{ref4}(\frac{rad}{seg})$	$\theta(rad)$	$X_R(mm)$
PI	0.28	0.41	0.37	0.37	0.0408	12.35
PI _{Sincro 1}	0.2827	0.2825	0.2906	0.3358	0.0568	2.52
PI _{Sincro 2}	0.2756	0.3792	0.3466	0.3270	0.0282	14.06
PI _{Sincro 3}	0.3002	0.3174	0.3786	0.3337	0.0164	8.96
PMD	0.23	0.44	0.38	0.37	0.0349	18.6
PMD _{Sincro 1}	0.2417	0.2534	0.2947	0.2855	0.0624	5.81
PMD _{Sincro 2}	0.1976	0.3552	0.3142	0.2803	0.0266	22.4
PMD _{Sincro 3}	0.2977	0.2890	0.3689	0.3159	0.0716	9.23
IMD	1.78	2	1.66	2.21	0.0334	28.85
IMD _{Sincro 1}	1.774	1.69	1.77	1.72	0.0622	5.35
IMD _{Sincro 2}	1.447	1.40	1.41	1.43	0.0193	17.83
IMD _{Sincro 3}	1.494	1.45	1.49	1.46	0.0246	5.94
PIMD	0.42	0.56	0.53	0.51	0.0868	18.02
PIMD _{Sincro 1}	0.3690	0.3384	0.3585	0.3794	0.0646	6.39
PIMD _{Sincro 2}	0.3990	0.4671	0.4351	0.4138	0.0437	19.31
PIMD _{Sincro 3}	0.476	0.4181	0.5085	0.4560	0.0159	8.24

Tabla 7.1: Tabla de RMSE del RMO en movimiento diagonal sobre concreto según las diferentes acciones de control aplicadas a los motores del RMO.

transversal el 3er esquema de sincronización obtuvo mejores resultados.

Para finalizar se realizaron experimentos con un movimiento diagonal frontal hacia la derecha con las mismas acciones de control y los mismos esquemas de sincronización. Para generar este movimiento se introdujo una referencia de $6\frac{rad}{seg}$ a las ruedas 1 y 4, y una referencia de $0\frac{rad}{seg}$ para las ruedas 2 y 3. Los experimentos del sistema sin sincronización ante las acciones de control proporcional-integral, proporcional con modos deslizantes, integral con modos deslizantes y proporcional con modos deslizantes, para generar un movimiento diagonal frontal hacia la derecha pueden observarse en las figuras 7.14,7.15,7.16,7.17, respectivamente. Para el movimiento diagonal frontal hacia la derecha, en la mayoría de los casos, funcionó mejor la configuración sin sincronización, a excepción de el segundo esquema de sincronización en los casos de acción integral con modos deslizantes y en la acción proporcional-integral con modos deslizantes, lo cual es lógico, debido a que en un movimiento diagonal, se estableció la ecuación 7.3, la cual describe que no hace falta una sincronización entre las ruedas impares para este movimiento. Para información sobre el desempeño en base a la raíz del error promedio cuadrado consulte la tabla 7.3.

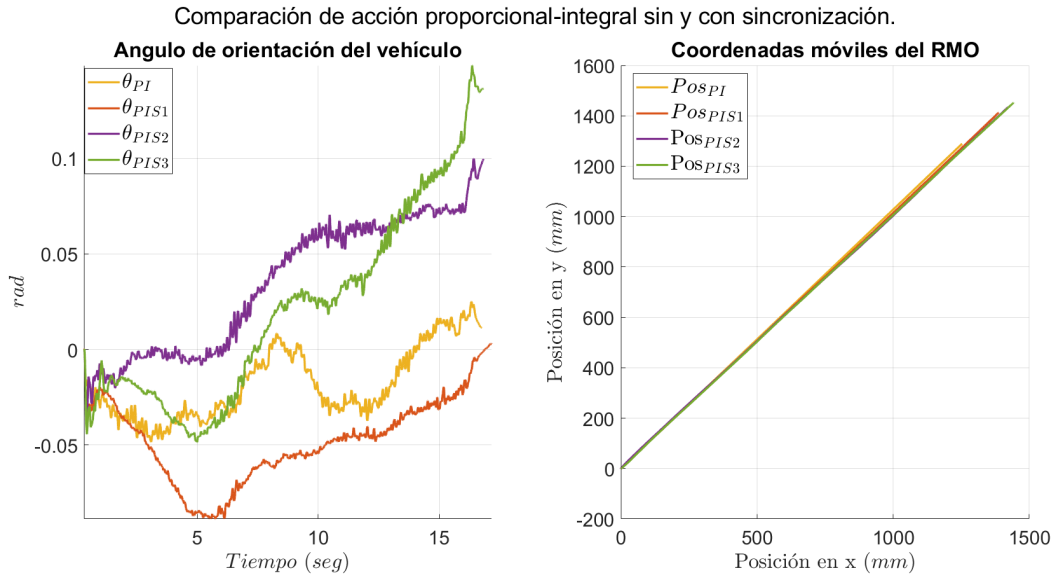


Figura 7.14: Respuesta del RMO sobre concreto con un movimiento diagonal frontal hacia la derecha, con acción proporcional integral, sin y con los diferentes esquemas de sincronización sobre las ruedas.

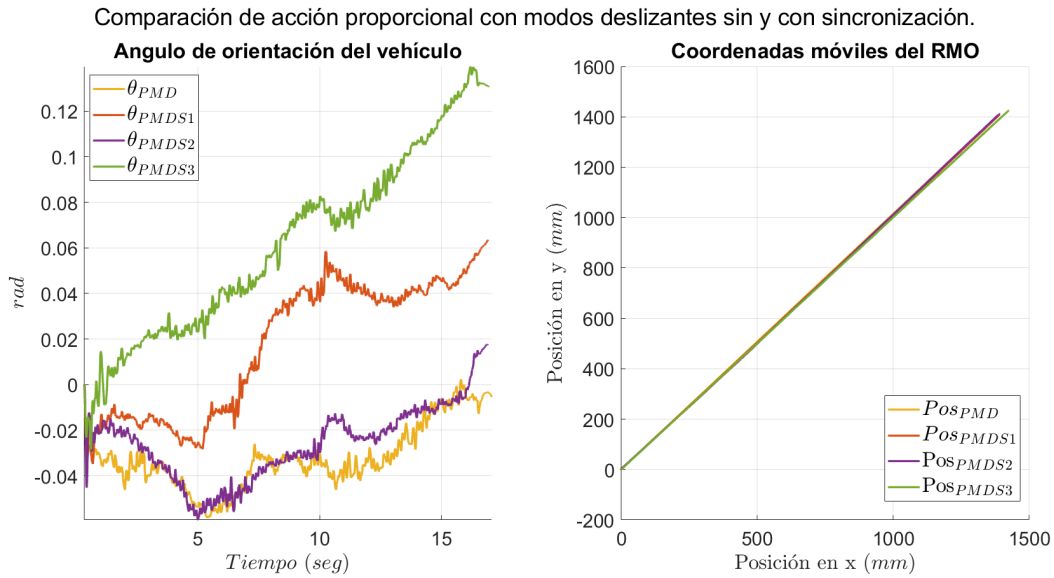


Figura 7.15: Respuesta del RMO sobre concreto con un movimiento diagonal frontal hacia la derecha, con acción proporcional con modos deslizantes, sin y con los diferentes esquemas de sincronización sobre las ruedas.

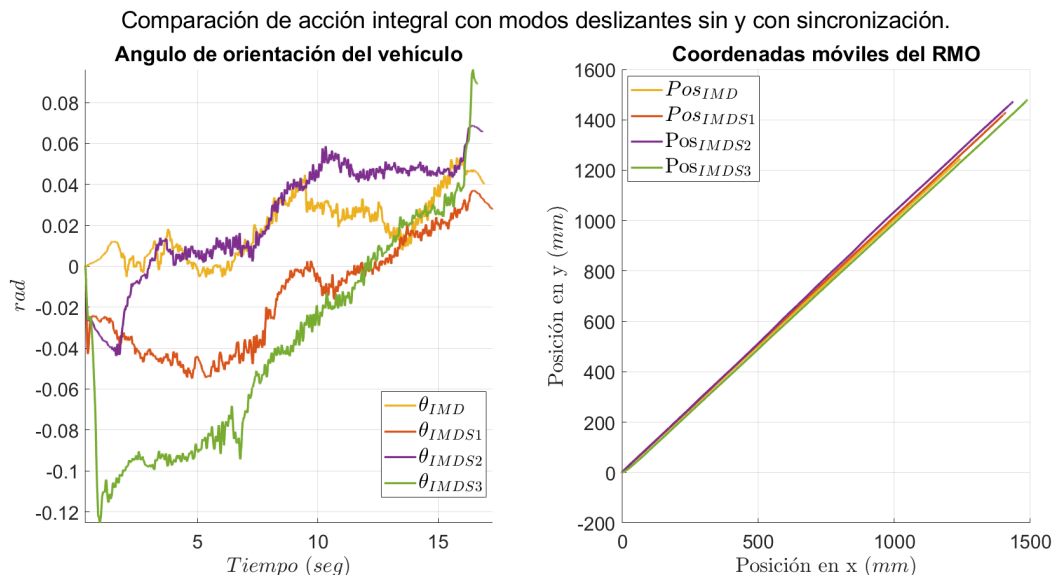


Figura 7.16: Respuesta del RMO sobre concreto con un movimiento diagonal frontal hacia la derecha, con acción integral con modos deslizantes, sin y con los diferentes esquemas de sincronización sobre las ruedas.

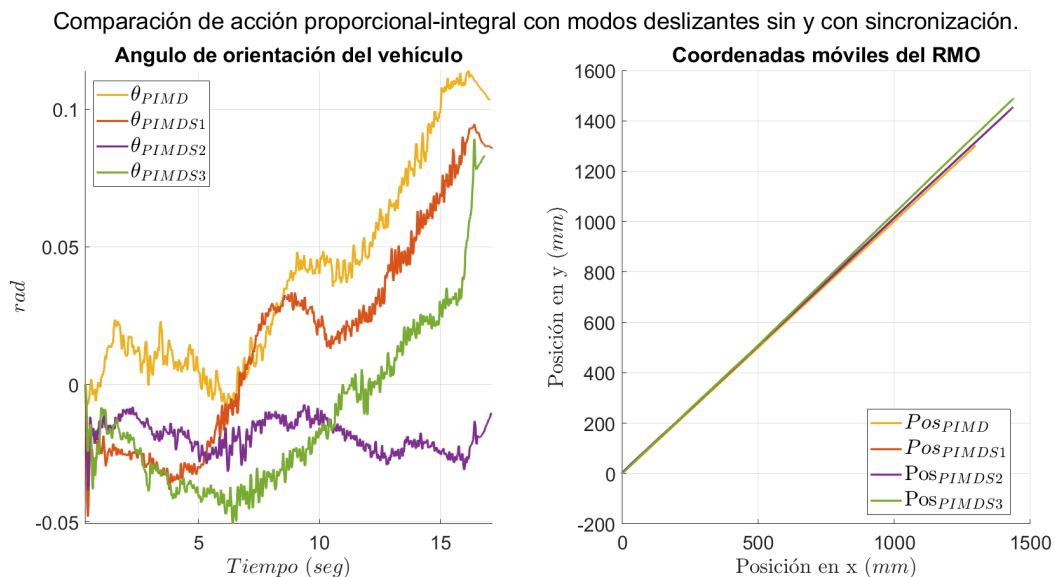


Figura 7.17: Respuesta del RMO sobre concreto con un movimiento diagonal frontal hacia la derecha, con acción proporcional integral con modos deslizantes, sin y con los diferentes esquemas de sincronización sobre las ruedas.

	$\omega_{ref1}(\frac{rad}{seg})$	$\omega_{ref2}(\frac{rad}{seg})$	$\omega_{ref3}(\frac{rad}{seg})$	$\omega_{ref4}(\frac{rad}{seg})$	$\theta(rad)$	$Y_R(mm)$
PI	0.6348	0.6414	0.7538	0.7385	0.0878	1.05
PI _{Sincro 1}	0.5361	0.4642	0.425	0.452	0.0434	5.02
PI _{Sincro 2}	0.5622	0.5520	0.4673	0.4515	0.1113	6.742
PI _{Sincro 3}	0.5294	0.4554	0.5266	0.4748	0.0457	2.067
PMD	0.6711	0.6909	0.6595	0.6883	0.0313	3.82
PMD _{Sincro 1}	0.6474	0.5293	0.495	0.498	0.0680	8.53
PMD _{Sincro 2}	0.6959	0.6644	0.5328	0.5215	0.0731	8.31
PMD _{Sincro 3}	0.6456	0.570	0.6641	0.572	0.0252	0.93
IMD	1.79	1.87	1.7443	1.81	0.0605	4.57
IMD _{Sincro 1}	1.50	1.469	1.49	1.48	0.0743	4.54
IMD _{Sincro 2}	1.70	2.29	2.19	1.659	0.0328	8.58
IMD _{Sincro 3}	1.56	1.54	1.78	1.75	0.0949	1.56
PIMD	0.708	0.708	0.6788	0.7437	0.0211	7.36
PIMD _{Sincro 1}	0.720	0.617	0.582	0.620	0.0857	8.19
PIMD _{Sincro 2}	0.761	0.7964	0.613	0.601	0.0555	4.78
PIMD _{Sincro 3}	0.671	0.6113	0.660	0.606	0.0656	0.84

Tabla 7.2: Tabla de RMSE del RMO en movimiento diagonal sobre concreto según las diferentes acciones de control aplicadas a los motores del RMO.

	$\omega_{ref1}(\frac{rad}{seg})$	$\omega_{ref2}(\frac{rad}{seg})$	$\omega_{ref3}(\frac{rad}{seg})$	$\omega_{ref4}(\frac{rad}{seg})$	$\theta(rad)$	$X_R(mm)$
PI	0.5038	0.0936	0.1420	0.5824	0.0252	16.34
PI _{Sincro 1}	0.4383	0.0431	0.116	0.4232	0.0523	66.89
PI _{Sincro 2}	0.4426	0.1109	0.12	0.4564	0.0493	47.49
PI _{Sincro 3}	0.4002	0.0735	0.115	0.3281	0.0548	32.08
PMD	0.6264	0.0867	0.1011	0.7056	0.341	8.45
PMD _{Sincro 1}	0.5114	0.0624	0.1033	0.4872	0.0339	9.201
PMD _{Sincro 2}	0.5035	0.1062	0.1239	0.490	0.0307	10.41
PMD _{Sincro 3}	0.4108	0.0731	0.073	0.338	0.0735	0.8788
IMD	1.88	0.0287	0.0740	1.95	0.0246	1.99
IMD _{Sincro 1}	1.6288	0.0786	0.099	1.646	0.0307	12.25
IMD _{Sincro 2}	1.4642	0.12	0.1225	1.4736	0.0369	24.44
IMD _{Sincro 3}	0.758	0.075	0.253	0.400	0.0646	10.55
PIMD	0.6209	0.0427	0.0698	0.7183	0.0578	0.95
PIMD _{Sincro 1}	0.572	0.069	0.099	0.561	0.0420	10.52
PIMD _{Sincro 2}	0.548	0.098	0.084	0.527	0.0196	12.08
PIMD _{Sincro 3}	0.459	0.093	0.214	0.3126	0.0317	27.7

Tabla 7.3: Tabla de RMSE del RMO en movimiento diagonal sobre concreto según las diferentes acciones de control aplicadas a los motores del RMO.

Capítulo 8

Conclusiones y trabajo futuro

Se diseñó y construyó un robot móvil omnidireccional, fácil de recrear y útil, para el ámbito académico y de investigación. Se hizo un análisis útil sobre el uso apropiado y bien aprovechado de una tarjeta NI-MyRIO-1900 y el guardado de datos interno y eficiente, para la realización de experimentos científicos de mediana duración con una gran cantidad de variables a guardar. Con estas herramientas se logró hacer una identificación de los 4 motores de CC con base en experimentos. Además, se presentaron diversos programas de LabVIEW, tanto para el guardado eficiente de datos de la MyRIO, como para el control de un solo motor de CC e incluso un programa de control robusto para regular la orientación y la velocidad de un robot móvil omnidireccional con ruedas Mecanum, basado en retroalimentación de estados, controladores de acción proporcional, integral y por modos deslizantes, suficientemente robusto para compensar los parámetros no identificados del sistema, linealidades no consideradas en el sistema y perturbaciones externas. Además, se propusieron distintos esquemas de sincronización para compensar por las perturbaciones externas provocadas en algunas de las ruedas del RMO, de manera que cuando las ruedas estén sincronizadas y obtengan una velocidad resultante, esta no provoque una desorientación no deseada en el vehículo. Se analizó el desempeño de 8 diferentes esquemas de control aplicados de manera individual sobre los 4 motores del RMO para movimientos frontales, laterales y diagonales de manera simulada, como de manera experimental sobre una superficie de concreto plana, donde en ambos casos se logra apreciar que utilizar una sola acción de control puede no ser suficiente para alcanzar los objetivos de control como se

desea, por lo cual se tomaron únicamente las opciones de controladores con varias acciones de control, es decir, las acciones proporcional-integral, proporcional con modos deslizantes, integral con modos deslizantes y proporcional integral con modos deslizantes, donde cada uno de estos mostró tener un buen desempeño. Sin embargo, existen sistemas, por ejemplo, sistemas paralelos, donde no basta con tener controladores en cada una de sus subsistemas, sino que es necesario crear una retroalimentación ya sea del sistema completo o del resto de los subsistemas, para compensar por las diferencias entre cada subsistema. Para esto se propusieron 3 esquemas de sincronización y se comparó el desempeño contra las versiones sin sincronización, donde desde la simulación es posible observar que una apropiada sincronización entre las ruedas de un RMO puede reducir los cambios de orientación no deseados y las desviaciones en su trayectoria. Los mismos esquemas de sincronización se pusieron a prueba en el RMO fabricado para este trabajo de tesis al desplazarse sobre una superficie de concreto, es decir, una superficie irregular similar a una superficie de trabajo real, donde se demostró que en la mayoría de los casos, un esquema de sincronización acoplado apropiadamente puede mejorar el desempeño del vehículo. Con estos resultados es posible concluir que la sincronización de subsistemas en sistemas paralelos puede mejorar la eficiencia de un sistema, aumentando la robustez del sistema y eliminando comportamientos no deseados provocados por las diferencias paramétricas y perturbaciones externas distintas sobre cada subsistema. El presente trabajo tiene área de mejora todavía, en el sentido de que se puede establecer una funcional de sincronización única útil para cualquier movimiento del RMO, no únicamente para movimientos típicos. Además, parece adecuado investigar más sobre los esquemas de sincronización de subsistemas para sistemas en paralelo. Este tipo de análisis puede resultar útil, por ejemplo, en distintos robots móviles como drones, robots móviles con patas e incluso en robots manipuladores paralelos.

Bibliografía

- [1] M. Ahmed, H. Taha, A. Saaid, and M. Tarek. A real time control system simulation model based on labview graphical programming language. In *The International Conference on Electrical Engineering*, volume 10, pages 1–11. Military Technical College, 2016.
- [2] D. D. Aryanto and S. Sukir. Comparison of dc motor speed calculation efficiency between inductive proximity sensor and rotary encoder. *Elinvo (Electronics, Informatics, and Vocational Education)*, 8(1):19–25, 2023.
- [3] A. Barrientos. Nuevas aplicaciones de la robótica. robots de servicio. *Avances en robótica y visión por computador. Cuenca, Ediciones Castilla-La Mancha*, 288, 2002.
- [4] R. D. Betancourt Manjarres and S. Morales Rodriguez. Diseño e implementación de un controlador por modos deslizantes para el péndulo invertido. Fundación Universitaria Los Libertadores. Sede Bogotá.
- [5] G. M. Boizas. *Introducción a la Fabricación Aditiva (Cómo y porque de la impresión en 3D)*. Germán Martín Boizas, 2018.
- [6] W. Bolton. *Mecatrónica*. Alpha Editorial, 2017.
- [7] C. Cáceres, J. Rosário, and D. Amaya. Design, simulation, and control of an omni-directional mobile robot. *International Review of Mechanical Engineering*, 12(4):382, 2018.

- [8] Z.-Y. Chen, P.-R. Liaw, V. L. Nguyen, and P. T. Lin. Design of a high-payload mecanum-wheel ground vehicle (mngv). *Robotic Systems and Applications*, 1(1):24–34, 2021.
- [9] D. Chuk. Los sistemas de primer orden y los controladores pid, 2012.
- [10] N. I. CORP. Product documentation labview programming reference manual ... local variable. https://www.ni.com/docs/en-US/bundle/labview-api-ref/page/functions/local-variable.html?srsltid=AfmB0orjLuJiUbdtdCFJHoN9LqThBH4B-3Jimy_NfjQ0wzZzkBGw7Cf-, 2025. Accedido: (28 de Enero de 2025).
- [11] D. G. Z. . M. R. Cullen. *Ecuaciones Diferenciales*. McGraw-Hill Interamericana, 2006.
- [12] P. Dhaker. Introduction to spi interface. *Analog Dialogue*, 52(3):49–53, 2018.
- [13] I. Doroftei. Conceptual design of an omni-directional mobile robot. In *SYROM 2009: Proceedings of the 10th IFToMM International Symposium on Science of Mechanisms and Machines, held in Brasov, Romania, october 12-15, 2009*, pages 115–127. Springer, 2009.
- [14] I. Doroftei and B. Stirbu. Design, modeling and control of an omni-directional mobile robot. *Solid State Phenomena*, 166:173–178, 2010.
- [15] J. F. Dulhoste. Teoria de control. *Mérida-Venezuela: Universidad de Los Andes*, 2006.
- [16] M. GALA, V. KRYS, and T. KOT. The mechanical design and realization of the omnidirectional mobile robot odin. *MM Science Journal*, 2016.
- [17] M. d. V. Gutiérrez Hinestroza and S. Iturralde Kure. Fundamentos básicos de instrumentación y control, 2017.
- [18] V. M. Hernández-Guzmán, R. Silva-Ortigoza, and R. V. Carrillo-Serrano. *Control Automático: Teoría de diseño, construcción de prototipos, modelado, identificación y pruebas experimentales*. COLECCIÓN CIDETEC, 2013.

- [19] S. Humayun, M. Mehmood, F. Mahmood, and Q. UIIslam. A study and comparison of data transfer methods in labview. *World applied sciences journal*, 28(11):1772–1775, 2013.
- [20] e. a. I.I. Blekhman. On self-synchronization and controlled synchronization. *Elsevier*, 1997.
- [21] e. a. J. Pena Ramirez A. Denasi. Controlled synchronization: a huygens inspired approach. *The International Federation of Automatic Control*, 2014.
- [22] B. X. Legett HD, Page RA. Synchronized mating signals in a communication network: the challenge of avoiding predators while attracting mates. *The Royal Society*, 2019.
- [23] N. K. Liyanage. *MyRIO*. PhD thesis, Murdoch University, 2016.
- [24] L. Ljung and R. Singh. Version 8 of the matlab system identification toolbox. *IFAC Proceedings Volumes*, 45(16):1826–1831, 2012.
- [25] L. M. Londoño García. Vehículos autónomos retos de su implementación. *Universidad Externado de Colombia*, 2023.
- [26] R. J. Mantz. *Control por Modos Deslizantes Tomo 1. Modos deslizantes de primer orden: Fundamentos y aplicaciones*. National University of La Plata, 11 2020.
- [27] C. Miguez Santos et al. Análisis de la evolución del diseño del automóvil desde sus inicios hasta la actualidad. *Universidad de Valladolid*, 2018.
- [28] *User Guide and Specifications, NI MyRIO-1900*.
- [29] K. Ogata. *DINAMICA DE SISTEMAS*. PRENTICE-HALL HISPANOAMERICANA, S.A., 1987.
- [30] W. R. F. Olmedo and M. Sampedro. Cinemática y dinámica de robots móviles con ruedas. In *CILADI*, 2023.
- [31] A. A. Ozdemir and S. Gumussoy. Transfer function estimation in system identification toolbox via vector fitting. *IFAC-PapersOnLine*, 50(1):6232–6237, 2017.

- [32] D. Purcaru and E. Niculescu. An alternative method for resolution improvement to rotary incremental encoder. In *Proceedings of the 2nd WSEAS/IASME International Conference on Educational Technologies*, pages 33–38. Citeseer, 2006.
- [33] L. G. C. Ramírez, G. S. A. Jiménez, and J. M. Carreño. *Sensores y actuadores*. Grupo Editorial Patria, 2014.
- [34] J. J. Raygoza, S. Ortega, M. A. Carrasco, and A. Pedroza. Implementación en hardware de un sumador de punto flotante basado en el estándar ieee 754-2008. *e-Gnosis*, 7, 2009.
- [35] A. Rodriguez-Angeles and H. Nijmeijer. Mutual synchronization of robots via estimated state feedback: a cooperative approach. *IEEE Transactions on control systems technology*, 12(4):542–554, 2004.
- [36] J. C. Rodriguez-Quinonez and O. Real-Moreno. *Graphical Programming Using LabVIEWTM: Fundamentals and Advanced Techniques*, volume 46. IET, 2022.
- [37] M. P. Romero and M. Aarón. Análisis cinemático e implementación de una mano robótica servo-articulada aplicable como prótesis. *Maestro en Ciencias en Ingeniería Mecánica con especialidad en Diseño Mecánico, Instituto Politécnico Nacional*, 2011.
- [38] Y. Shtessel, C. Edwards, L. Fridman, A. Levant, et al. *Sliding mode control and observation*, volume 10. Springer, 2014.
- [39] A. C. Solé. *Instrumentación Industrial*. Alfaomega Grupo Editor, S.A. de C.V., de México, 2010.
- [40] S. Spurgeon. Sliding mode control: a tutorial. In *2014 European Control Conference (ECC)*, pages 2272–2277. IEEE, 2014.
- [41] H. Taheri, B. Qiao, and N. Ghaeminezhad. Kinematic model of a four mecanum wheeled mobile robot. *International journal of computer applications*, 113(3):6–9, 2015.

- [42] M. O. Tătar, C. Cirebea, and D. Mândru. Structures of the omnidirectional robots with swedish wheels. *Solid State Phenomena*, 198:132–137, 2013.
- [43] *3 Space sensor. User's Manual.*