



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

ESCUELA DE INGENIERÍA EN MECATRÓNICA

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO EN MECATRÓNICA

TEMA:

“CONTROL DE VELOCIDAD DE UN ROBOT MÓVIL CON
DIRECCIÓN DIFERENCIAL”

AUTOR: GABRIEL ANDRÉS FERNÁNDEZ CRUZ

DIRECTOR: CARLOS XAVIER ROSERO CHANDI

IBARRA-ECUADOR

MARZO 2019



UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA
AUTORIZACIÓN DE USO Y PUBLICACIÓN
A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DEL AUTOR			
CÉDULA DE IDENTIDAD:	1004267892		
APELLIDOS Y NOMBRES:	FERNÁNDEZ CRUZ GABRIEL ANDRÉS		
DIRECCIÓN:	URCUQUI - SAN IGNACIO		
EMAIL:	gafernandezc@utn.edu.ec		
TELÉFONO FIJO:	062939542	TELÉFONO MÓVIL:	0990928675

DATOS DE LA OBRA	
TÍTULO:	"CONTROL DE VELOCIDAD DE UN ROBOT MÓVIL CON DIRECCIÓN DIFERENCIAL"
AUTOR:	FERNÁNDEZ CRUZ GABRIEL ANDRÉS
FECHA :	FEBRERO 2019
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	PREGRADO
TÍTULO POR EL QUE OPTA:	INGENIERO EN MECATRÓNICA
ASESOR/DIRECTOR:	CARLOS XAVIER ROSERO C.

2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, febrero de 2019

A handwritten signature in blue ink, consisting of several overlapping loops and lines, positioned above the printed name and ID number.

Gabriel Andrés Fernández Cruz
C.I.: 1004267892



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CERTIFICACIÓN

En calidad de director del trabajo de grado “CONTROL DE VELOCIDAD DE UN ROBOT MÓVIL CON DIRECCIÓN DIFERENCIAL”, presentado por el egresado FERNÁNDEZ CRUZ GABRIEL ANDRÉS, para optar por el título de Ingeniero en Mecatrónica, certifico que el mencionado proyecto fue realizado bajo mi dirección.

Ibarra, febrero de 2019



Carlos Xavier Rosero C.
DIRECTOR DE TESIS



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
DECLARACIÓN

Yo, Gabriel Andrés Fernández Cruz con cédula de identidad Nro. 100426789 – 2, declaro bajo juramento que el trabajo aquí escrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

Ibarra, febrero de 2019

Gabriel Andrés Fernández Cruz
C.I.: 1004267892

Agradecimiento

En primer lugar a Dios, quien con su bendición llena siempre mi vida y a toda mi familia.

A la Universidad Técnica del Norte, por ser la institución que me permitió cumplir una de las metas mas importantes en mi vida académica y profesional.

A mis padres y hermanos por ser principal fuente de apoyo en cada momento de mi formación como profesional y mas importante aún como persona. A mis tíos y primos por estar de una u otra forma siempre pendientes de mi avance en el proyecto.

A Lizeth Flores, por su comprensión, por estar conmigo incluso en los momentos más difíciles, por su motivación y su ayuda a lo largo de toda mi etapa universitaria

A mi director Carlos Xavier Rosero por su conocimiento y tiempo, quien además confió en mi desde el primer momento, permitiéndome realizar el presente trabajo.

Al grupo de docentes que conforma la Carrera de Ingeniería en Mecatrónica, por ser mas que educadores, un grupo de amigos, cuyo conocimiento compartido es fundamental para la culminación de este trabajo.

Gabriel Andrés F.

Dedicatoria

Principalmente dedico este trabajo a mis padres, quienes con su sacrificio y esfuerzo han permitido que pueda obtener un logro mas en mi vida.

A mi hija Betsabe Fernández, por enseñarme a vivir una nueva etapa en mi vida y por ser mi motivación mas grande para concluir con éxito este proyecto de tesis.

Gabriel Andrés F.

Resumen

Los robots móviles son sistemas electromecánicos capaces de desplazarse de forma autónoma. En los últimos años su estudio ha ido en aumento, proporcionando avances tecnológicos de hardware y software, muchos de ellos usados dentro del proceso de enseñanza-aprendizaje. Particularmente en la Universidad Técnica del Norte existe un robot móvil, el cual está construido con hardware y software existente en la Universidad. Este posee motores acoplados a llantas, colocados en configuración diferencial para mejorar el desplazamiento y rotación. Además, se cuenta con un sistema de medición basado en codificadores incrementales utilizados para la odometría. Para el control y adquisición de datos en tiempo real se usa un sistema embebido sbRIO 9631. El control de velocidad implementado está en lazo abierto, lo que no proporciona un desempeño eficiente. El presente trabajo propone implementar un control de velocidad en lazo cerrado. Para desarrollar el controlador se realiza un modelado matemático siguiendo el método de Cecil-Smith, además se realiza una identificación con el software MatLab. El control se diseña considerando: mayor estabilidad del sistema, reducción de sobre impulso, facilidad de implementar, para lo cual se realiza una comparación entre distintos métodos de diseño de controladores. Además, se realizan simulaciones del controlador y la planta en lazo cerrado mediante el software MatLab, para comprobar su desempeño. Una interfaz gráfica se desarrolla en el software LabView que permite manipular variables del controlador y observar las señales de los sensores a través de indicadores fáciles de interpretar. Las pruebas del controlador se realizan en distintas condiciones, con el fin de comprobar su correcto funcionamiento. Como resultado el controlador cumple con las especificaciones de desempeño consideradas en el diseño.

Abstract

Mobile robots are electromechanical systems capable of moving autonomously. In recent years, its study has been increasing, providing technological advances in hardware and software, many of them are used in the teaching-learning process. Particularly in the Universidad Técnica del Norte there is a mobile robot, which is built with existing hardware and software at the University. This has motors coupled to tires, placed in differential configuration to improve the displacement and rotation. It is also equipped with a measurement system based on incremental encoders used for the odometry for the control and acquisition of data in real time an embedded system is used sbRIO 9631. The speed control implemented is in open loop, which does not provide effective performance. This work proposes to implement a speed control in closed loop. To develop the controller, a mathematical modeling is carried out following the Cecil-Smith method, in addition an identification is made with the MatLab software. The control is designed considering: greater stability of the system, reduction of over impulse, ease to implement, for which a comparison is made between different design methods of controllers In addition, simulations of the controller and the closed loop plant are carried out using the MatLab software, to check its performance. A graphical interface is developed in LabView software that allows you to manipulate controller variables and observe the signals of the sensors through easy-to-interpret indicators. The controller tests are performed in different conditions, in order to check its correct operation. As a result, the controller complies with the performance specifications considered in the design.

Índice general

Índice general	X
Índice de figuras	XIV
Índice de tablas	XVII
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	2
1.2.1. Objetivo general	2
1.2.2. Objetivos específicos	2
1.3. Antecedentes	2
1.4. Problema	3
1.5. Justificación	3
1.6. Alcance	4
2. Revisión Literaria	5
2.1. Robótica móvil	5
2.2. Configuración diferencial	5
2.3. Control robot móvil	7
2.4. Cinemática del sistema	7
2.4.1. Representación en el plano	8
2.5. Dinámica del sistema	11
3. Identificación del sistema motriz	15
3.1. Descripción general del robot móvil	15

3.2.	Aproximación en bloques	16
3.3.	Métodos de aproximación a un modelo matemático	17
3.3.1.	Modelado teórico	17
3.3.2.	Identificación de un sistema	17
3.4.	Proceso de identificación de los motores	18
3.4.1.	Obtención de datos entrada - salida	18
3.4.2.	Tratamiento de los datos registrados	18
3.4.3.	Elección de la estructura del modelo	19
3.4.4.	Validación del modelo matemático	21
3.5.	Discretización de la función de transferencia	22
4.	Diseño del controlador de velocidad en lazo cerrado	24
4.1.	Control sobre el modelo cinemático	24
4.2.	Sistema de control digital de velocidad	27
4.3.	Métodos para diseñar un controlador	28
4.3.1.	Lugar geométrico de las raíces	28
4.3.1.1.	Controlador PI	31
4.3.2.	Método analítico	33
4.3.2.1.	Error de seguimiento nulo a entrada escalón	34
4.3.2.2.	Estabilidad	34
4.3.2.3.	Respuesta Dead-beat	35
4.3.3.	Ziegler Nichols - MatLab	36
4.3.3.1.	Control PI	38
4.3.4.	Controlador escogido	40
5.	Implementación del controlador sobre el robot móvil	42
5.1.	LabView	42
5.2.	Sistema en LabView	43
5.3.	Interfaz gráfica	43

5.3.1.	Visualización del estado de las baterías	43
5.3.2.	Visualización de la velocidad de los motores	44
5.3.3.	Maniobrabilidad del robot	44
5.3.4.	Controlador PI	44
5.3.5.	Visualización de sensor IMU	44
5.3.6.	Lectura de codificadores	45
5.3.7.	Análisis odométrico	45
5.3.8.	Sensor ping	45
5.4.	Implementación del controlador en el software LabView	46
5.4.1.	Implementación del programa en el computador	48
6.	Simulaciones y Pruebas de funcionamiento	50
6.1.	Simulaciones	50
6.1.1.	Modelo cinemático	50
6.1.2.	Función de transferencia	53
6.2.	Prueba en los sensores	54
6.2.1.	Lectura de codificadores	54
6.2.2.	Velocidad de los motores	55
6.2.2.1.	Controlador en lazo abierto	56
6.2.2.2.	Controlador en lazo cerrado	56
6.2.3.	Voltaje de baterías	57
6.2.4.	Distancia	58
6.3.	Pruebas del controlador de velocidad	60
6.3.1.	Controlador en vacío	60
6.3.2.	Controlador en superficies planas	62
6.3.2.1.	Trayectoria rectilínea control en lazo abierto	63
6.3.2.2.	Trayectoria rectilínea control en lazo cerrado	64
6.3.2.3.	Trayectoria circular control en lazo abierto	66
6.3.2.4.	Trayectoria circular control lazo cerrado	68

6.3.3.	Controlador en superficies rugosas	69
6.3.3.1.	Trayectoria circular control en lazo abierto	69
6.3.3.2.	Trayectoria circular control en lazo cerrado	71
7.	Conclusiones y trabajo futuro	73
7.1.	Conclusiones	73
7.2.	Trabajo futuro	74
	Bibliografía	75
	Apéndice	79
.A.	Software	79
.A.1.	Modelo cinemático	79
.A.2.	Función de transferencia	81

Índice de figuras

2.1. Robot móvil starter kit LabView Robotics	6
2.2. Robot en configuración diferencial	6
2.3. Sistema de coordenadas cartesianas	8
2.4. Ejes del robot en coordenadas universales	9
2.5. Posición del robot móvil	10
3.1. Diagrama de bloques de control en lazo abierto	16
3.2. Diagrama de bloques de control en lazo cerrado	16
3.3. Programación en LabView para recolección de datos	18
3.4. Respuesta de la planta, aplicando media móvil para disminuir ruido	19
3.5. respuesta de primer orden Cecil-Smith	20
3.6. Identificación MatLab. a) barra de herramientas MatLab, b) ventana principal Ident, c) Importar datos a Ident, d) Tipo de estimación para el modelo, e) datos de K, τ, θ , f) estimaciones de la función de transferencia principal	22
4.1. Diagrama de bloques de control en lazo cerrado	28
4.2. Diagrama de bloques de control en lazo cerrado en dominio de z	28
4.3. Lugar geométrico de las raíces	29
4.4. Diagrama de bloques de control PI en lazo cerrado	31
4.5. Respuesta al escalón unitario del control y planta	33
4.6. LGR del controlador por método analítico	36
4.7. Respuesta al escalón unitario	36
4.8. Respuestas al escalón unitario, a) respuesta al escalón, b) comando Step	37
4.9. a) lugar geométrico del sistema en lazo cerrado, b) polos y ceros del sistema	38
4.10. Respuesta de primer orden al escalón unitario	39

4.11. Respuesta del sistema al escalón	40
5.1. Interfaz gráfica	46
5.2. Diagrama de flujo control de velocidad	47
5.3. Diagrama de bloques del controlador	48
5.4. SubVi control PI	48
6.1. Respuesta a velocidades iguales	51
6.2. Respuesta a velocidades iguales, a) desplazamiento general, b) desplazamiento en los vectores x, y, ϕ	51
6.3. Respuesta a velocidades distintas	52
6.4. Respuesta a velocidades distintas, a) desplazamiento general, b) desplazamiento en los vectores x, y, ϕ	52
6.5. Diagrama de bloques en Simulink, a) tiempo discreto, b) tiempo continuo	53
6.6. Respuesta de diagrama de bloques	54
6.7. Prueba de lectura de codificadores, a) programa implementado en LabView, b) robot vista inferior	55
6.8. Lectura de velocidad, a) lazo abierto, b) lazo cerrado	57
6.9. Lectura de voltaje en las baterías, a) programa implementado LabView, b) vol- tímetro digital	58
6.10. Medición de distancias, a) programa LabView, b) sensor ping	59
6.11. Robot móvil vista inferior	60
6.12. Prueba en vacío de motores	61
6.13. Prueba en vacío de motores con perturbación exterior	61
6.14. Prueba en vacío de motores, a) valor de referencia, tiempo de estabilización, sobre impulso, b) valor de referencia y respuesta a perturbaciones	62
6.15. Interfaz gráfica con resultados del control en lazo abierto. Pág 49	63
6.16. Prueba real trayectoria rectilínea superficie lisa a) punto de partida, b) punto final	64
6.17. Interfaz gráfica con resultados del controlador en lazo cerrado. Pág 49	65

6.18. Prueba real trayectoria rectilínea superficie lisa, a) punto de partida, b) punto final	66
6.19. Interfaz gráfica con resultados del controlador en lazo abierto. Pág 49	67
6.20. Prueba real trayectoria circular superficie lisa	67
6.21. Interfaz gráfica con resultados del controlador en lazo cerrado. Pág. 49	68
6.22. Prueba real trayectoria circular superficie lisa	69
6.23. Interfaz gráfica con resultados del controlador en lazo abierto. Pág. 49	70
6.24. Prueba de trayectoria circular control lazo abierto, a) punto inicial de la prueba, b) recorrido del robot	71
6.25. Interfaz gráfica con resultados control lazo cerrado. Pág 49	71
6.26. Prueba de trayectoria circular lazo cerrado, a) punto inicial, b) recorrido del robot móvil	72
1. Diagrama de bloques del controlador implementado en el RT	84

Índice de tablas

4.1. Parámetros para calculo de controladores P, PI, PID	39
6.1. Comparación revoluciones y pulsos	55
6.2. Velocidades de las ruedas sin control	56
6.3. Velocidades de las ruedas con el control implementado	57
6.4. Voltaje de baterías	58
6.5. Comparación de distancias	59

Capítulo 1

Introducción

Este trabajo de grado ha sido realizado con el *Grupo de Investigación en Sistemas Inteligentes de la Universidad Técnica del Norte (GISI-UTN)*.

1.1. Motivación

Para que un robot móvil pueda seguir órdenes definidas y completar la trayectoria sin errores, se debe diseñar un control de trayectorias, esto implica tener varios algoritmos de control (posición, dirección, velocidad, etc), en niveles jerárquicos, de manera que el robot móvil pueda tener la mayor autonomía posible.

La Universidad Técnica del Norte posee un robot móvil, el cual sirve para investigación. El robot no cuenta con un sistema de control ya que en trabajos anteriores se trabajó en la construcción y estudio en la odometría. Este trabajo propone implementar un control de velocidad para cada uno de los motores del robot.

1.2. Objetivos

1.2.1. Objetivo general

Desarrollar un controlador de velocidad en lazo cerrado para cada uno de los motores del robot móvil con dirección diferencial.

1.2.2. Objetivos específicos

- Realizar la identificación de los motores con el fin de obtener un modelo que describa su funcionamiento.
- Diseñar la estrategia de control considerando el modelo de los motores.
- Implementar el control sobre los motores del robot móvil.

1.3. Antecedentes

En la Universidad Técnica del Norte no se ha encontrado temas similares al propuesto, por lo tanto se toma de referencia trabajos propuestos por otras Universidades dentro y fuera del país. Por ejemplo, La Universidad Politécnica Salesiana sede Guayaquil, realizó el Diseño e Implementación de un robot autónomo móvil usando tecnología FPGA de National Instrument, construye un robot de 2 ruedas monitoreado en tiempo real por una PC a través de una interfaz gráfica realizada en LabView [1].

La Universidad Nacional de San Juan, propone estrategias de control para seguimiento de trayectoria de robots móviles, presenta por métodos numéricos y por interpolación lineal, analiza las condiciones bajo las cuales el sistema tiene solución exacta así entonces obtener la trayectoria deseada de algunos estados. El controlador obtenido es de fácil ajuste y la influencia de parámetros en el ajuste del controlador es muy intuitiva además se obtuvo diferentes

controladores para una misma aproximación [2, 3].

1.4. Problema

El control de dirección del robot móvil y los procesos subyacentes, por lo general son un problema muy complejo debido a restricciones de varios factores entre los que se encuentran: tiempo de retardo, perturbaciones, tiempo de muestreo y respuesta del actuador. Debido a esto, la mayoría de robots móviles necesitan una re-sintonización de los parámetros de control o incluso cambiar el programa de control por completo [4, 5].

En lo concerniente a la Universidad Técnica del Norte y particularmente en la Facultad de Ingeniería en Ciencias Aplicadas, existe un robot móvil que se utiliza como herramienta de investigación, para el planeamiento de rutas. Éste se encuentra aún en desarrollo y se puede conocer más detalle en el documento [6].

El mencionado robot móvil no presenta ningún tipo de controlador (sensores, actuadores, SCADA) ya que anteriormente se realizó su implementación y no se abordó el tema de control [6]. Debido a esto, el robot no es capaz de seguir trayectorias, compuestas de líneas rectas y curvas, como paso previo para su movimiento autónomo con planeación de rutas. Con la finalidad de resolver el problema mencionado, se propone implementar un control de velocidad de este robot móvil con dirección diferencial [7].

1.5. Justificación

La robótica móvil posee un amplio campo de investigación. Los robots móviles son una parte importante en la creación de sistemas con autonomía, capaces de procesar la información de su entorno y establecer un comportamiento similar al de una persona; tienen gran variedad de aplicaciones [8].

Un robot móvil es un sistema de control en lazo cerrado, que se desplaza sobre ruedas y se controla de forma autónoma utilizando la información captada por sus sensores y transmitida a sus motores de forma independiente, para que así pueda evitar obstáculos [9].

Para que un robot móvil pueda tener autonomía de desplazamiento debe tener la capacidad de reconocer su posición dentro de su ambiente. Si no es capaz de lograr esto, corre el riesgo de sufrir choques contra obstáculos que se presenten en su recorrido además de desviarse de su ruta establecida [7, 9].

La implementación de este sistema de control se basa en nociones, habilidades, capacidades, destrezas, aptitudes, vinculadas a la competencia profesional y los conocimientos adquiridos por el Ingeniero Mecatrónico [6].

1.6. Alcance

Se diseñará el sistema de control de velocidad para un robot móvil proporcionado por el Grupo de Investigación en Sistemas Inteligentes – UTN, el cual está constituido por 4 motores DC (corriente continua). Se identificará los motores del robot móvil a través de técnicas convencionales y se controlará la velocidad individual de las ruedas en lazo cerrado, con la finalidad de satisfacer los objetivos propuestos.

Capítulo 2

Revisión Literaria

2.1. Robótica móvil

La robótica es una ciencia que ha venido evolucionando desde hace mucho tiempo, siempre con el fin de lograr una mayor autonomía y hacer frente a tareas complejas. La robótica posee un gran número de ramas en las que se destaca la ROBÓTICA MÓVIL [10].

Un robot móvil se define como un sistema electromecánico capaz de desplazarse de forma autónoma desde un punto A a un punto B. La norma ISO 8373 define robot móvil como, “Robot capaz de viajar bajo su propio control moviéndose dentro de su entorno, con la capacidad para realizar las tareas previstas basadas en el estado actual y la detección, sin intervención humana” [11, 12].

La Fig. 2.2 muestra un Robot móvil creado por National Instrument para el uso de la tarjeta embebida NI sbRIO-9631, con un entorno de programación gráfica del módulo LabView Robotics [13].

2.2. Configuración diferencial

La configuración diferencial se presenta como la más sencilla de todas, consta de dos ruedas situadas diametralmente opuestas en un eje perpendicular a la dirección del robot [14]. Con dos ruedas es imposible mantener la horizontal del robot. Se producen cabeceos al cambiar la dirección. Para solventar este problema, se colocan 2 ruedas extra, dándole estabilidad al robot.

Cada una de ellas esta dotada de un motor, de forma que los giros se realizan dándoles diferentes velocidades [15].

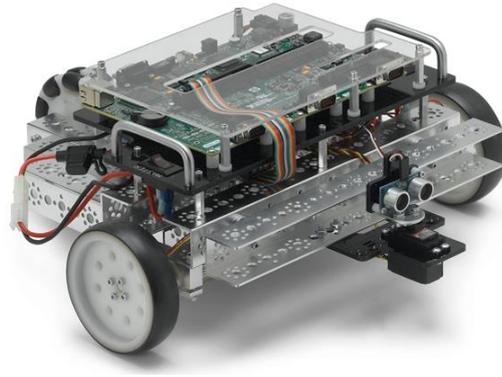


Figura 2.1: Robot móvil starter kit LabView Robotics

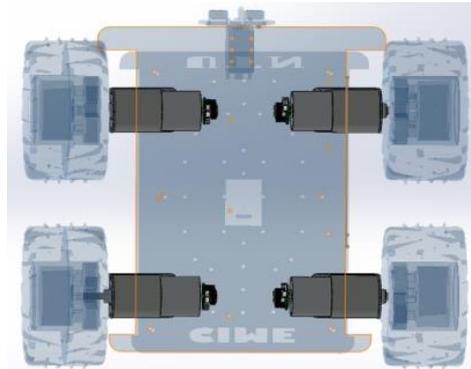


Figura 2.2: Robot en configuración diferencial

En la Fig. 2.2 se muestra al robot móvil con estructura diferencial, utilizando 4 ruedas para la tracción las cuales están conectados a motores y a su vez estos están conectados a encoders para la lectura de su posición y velocidad [14]. Este tipo de robots no son aptos para terrenos irregulares, por lo que su campo de acción de encuentra limitado a superficies planas.

2.3. Control robot móvil

El objetivo de un sistema de control es, dada una referencia determinada obtener una salida del sistema que se aproxime a la señal de referencia considerando perturbaciones, dinámica del sistema y a factores externos que puedan intervenir en el desplazamiento del robot.

En la Universidad de la República en Montevideo Uruguay se realiza la programación en java en el software Eclipse, donde se desarrolla una propuesta que se basa en la representación del espacio de configuraciones libres a través de la construcción de roadmaps, muestreo aleatorio de configuraciones en torno a las aristas del roadmap y búsqueda de caminos óptimos con un algoritmo genético; no especifica si construyó o no el robot [2].

La Universidad Técnica de Ambato Ecuador presentó un trabajo de modelación cinemática y dinámica de un robot móvil tipo unicycle no holonómico, el modelo propuesto recibe como entradas la velocidad lineal y a la velocidad angular. Para resolver el problema de movimiento de los robots móviles se propone tres algoritmos de control: control de posicionamiento, control de seguimiento de trayectoria y el control de seguimiento de caminos. El diseño de los algoritmos de control se basa en dos subsistemas, un controlador cinemático, el mismo que cumple con los objetivos de la tarea; y un controlador dinámico que compensa la dinámica del robot móvil, en cada uno de estos controladores se llevó a cabo el análisis de estabilidad mediante la aplicación de la teoría de Lyapunov [16].

2.4. Cinemática del sistema

Para realizar el modelado del robot se debe tener en cuenta ciertas hipótesis que generalizan el comportamiento del robot [17]

- Se asume que el robot se desplaza en una superficie plana idealmente sin rozamiento.
- Los ejes de las ruedas son perpendiculares al suelo por donde se desplaza.

- El robot se debe mover únicamente por las fuerzas ejercidas por el movimiento rotacional de las ruedas.

Para obtener un control preciso de robots móviles es indispensable crear modelos matemáticos que permitan simular la dinámica de movimiento.

2.4.1. Representación en el plano

Para describir la posición y orientación de robot móvil en el plano, se utiliza los movimientos de rotación y traslación, realizando operaciones básicas se determina la posición y orientación del robot. Para reconocer el movimiento del robot móvil se debe conocer la posición y orientación inicial y final, esto nos introduce en el análisis de la cinemática de robots móviles [6].

Determinar la posición de un robot en el plano es indispensable para el estudio de robots móviles, debido a que la mayoría de éstos se mueven en áreas conocidas sin irregularidades. Es necesario conocer la posición en coordenadas x, y , el ángulo de orientación γ , para tener definido al robot [14].

En la Fig. 2.3 se determina un sistema de coordenadas, A representa las coordenadas del sistema de referencia.

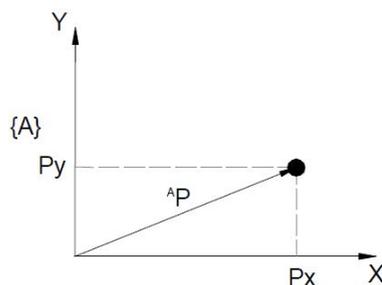


Figura 2.3: Sistema de coordenadas cartesianas

Se puede definir al punto P respecto al eje de coordenadas A como un vector de posición de

la siguiente forma

$${}^A P = \begin{bmatrix} P_x \\ P_y \end{bmatrix} \quad (2.1)$$

En la Fig. 2.4 se observa un eje de coordenadas B en el centro de masa del móvil de forma que los vectores unitarios de este segundo sistema de coordenadas son X_B y Y_B . La dirección del vector Y_B forma un ángulo con el vector P_x .

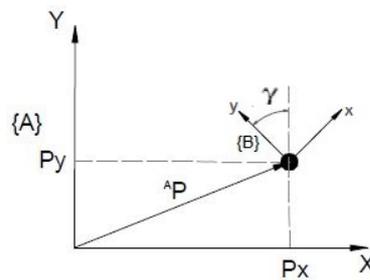


Figura 2.4: Ejes del robot en coordenadas universales

Expresando los vectores unitarios del sistema B en el A , se obtiene ${}^A X_B$ y ${}^A Y_B$

$${}^A X_B = \begin{bmatrix} \text{Cos}(\gamma) \\ \text{Sin}(\gamma) \end{bmatrix} \quad (2.2)$$

$${}^A Y_B = \begin{bmatrix} -\text{Sin}(\gamma) \\ \text{Cos}(\gamma) \end{bmatrix} \quad (2.3)$$

Estos vectores se disponen según las columnas de una de una matriz denominada matriz de rotación. Estas matrices son muy importantes en los modelos empleados en robótica.

$${}^A R = \begin{bmatrix} {}^A X_B & {}^A Y_B \end{bmatrix} = \begin{bmatrix} \text{Cos}(\gamma) & \text{Sin}(\gamma) \\ \text{Sin}(\gamma) & \text{Cos}(\gamma) \end{bmatrix} \quad (2.4)$$

El proceso básico para la localización de un robot móvil se basa en el modelo cinemático del sistema de tracción. Este sistema es el que permite al robot moverse alrededor de un determinado entorno, el tipo de sistema de este robot móvil es de tracción diferencial. La posición de

un robot de este tipo puede ser estimada a partir de las ecuaciones geométricas que surgen de la relación entre los componentes del sistema de tracción y de la información de los codificadores rotativos que llevan acoplados a sus ruedas [6].

En la Fig. 2.5 se muestra la localización del robot dentro de un sistema de referencia (x, y, γ) , donde la V es la velocidad lineal instantánea del robot móvil y V_L, V_R es la velocidad lineal tangencial de cada rueda, con las cuales se puede determinar las ecuaciones de posición del robot móvil [17].

$$V_R = R_R * \omega_R \quad (2.5)$$

$$V_L = R_L * \omega_L \quad (2.6)$$

$$V = (V_R + V_L)/2 \quad (2.7)$$

donde ω_R y ω_L son las velocidades angulares de cada rueda [17].

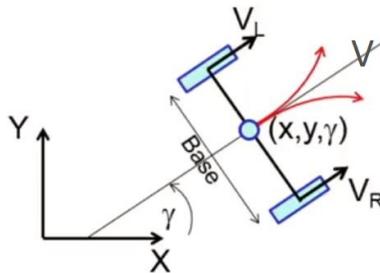


Figura 2.5: Posición del robot móvil

la velocidad angular del robot móvil se obtiene con

$$\omega = (V_R - V_L)/Base \quad (2.8)$$

para la representación de la posición del robot, con un incremento de tiempo considerablemente leve se obtiene

$$x' = x + V * \cos(\gamma) * t \quad (2.9)$$

$$y' = y + V * \sin(\gamma) * t \quad (2.10)$$

$$\gamma' = \gamma + \omega * t \quad (2.11)$$

Conociendo que el robot cumple con hipótesis antes mencionadas se puede demostrar que, q es el vector de coordenadas y orientación del punto guía del robot y q' es el vector de velocidad lineal y angular del robot móvil y se expresa como

$$q = [x(t), y(t), \gamma(t)]^T \quad (2.12)$$

$$q' = [v(t), \omega(t)]^T \quad (2.13)$$

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\gamma}(t) \end{bmatrix} = \begin{bmatrix} \cos(\gamma)(t) & 0 \\ \sin(\gamma)(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} \quad (2.14)$$

2.5. Dinámica del sistema

El diseño de un controlador continuo o discreto, ya sea mediante técnicas convencionales o variables de estado, necesita de un modelo de planta para controlar, que simule su comportamiento dinámico [16]. Dicho modelo permite realizar simulaciones para ajustes de parámetros del controlador que permiten obtener una respuesta que satisfaga el diseño [17, 18].

Considerar el modelo dinámico en el diseño del controlador es de gran importancia para que el sistema de tracción no pierda precisión en la tarea requerida. Para desarrollar el sistema dinámico se consideran efectos producidos por la masa de la estructura en el comportamiento del robot, además del momento de inercia del robot, los rozamientos y todas las perturbaciones [17, 18].

A continuación se presenta un modelo dinámico Euler-Lagrange para un robot móvil con tracción diferencial. Se parte de la ecuación de Lagrange dada por

$$\frac{d}{dt} \left(\frac{\partial L(q, \dot{q})}{\partial \dot{q}_i} \right) - \frac{\partial L(q, \dot{q})}{\partial q_i} = \tau_i, \quad i = 1, \dots, n \quad (2.15)$$

Donde:

- $q \in \mathbb{R}^{1 \times n}$ es el vector de coordenadas Generalizadas.
- $\tau \in \mathbb{R}^{1 \times n}$ es el vector de pares o fuerzas externas.
- $L(q, \dot{q}) \in \mathbb{R}$ corresponde a la función de Lagrange, la cual esta definida por la diferencia entre la energía cinética y la energía potencial, tal como se representa en

$$L(q, \dot{q}) = K(q(t), \dot{q}(t)) - U(q(t)) \quad (2.16)$$

La estructura general del modelo dinámico se define por

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (2.17)$$

Donde:

- $G(q)$ denota al vector de pares gravitacionales, el cual es de dimensión $n * 1$.
- La matriz de pares inerciales $M(q, \dot{q})$, es simétrica y definida positiva de dimensiones $n * n$.
- $C(q, \dot{q})$, es la matriz de fuerzas centrífugas y de Coriolis, que es una matriz de dimensiones $n * n$. Las cuales satisfacen la siguiente propiedad.

$$\dot{q}^T \left[\frac{1}{2} \dot{M}(q) - C(q, \dot{q}) \right] \dot{q} = 0, \quad \forall q, \dot{q} \in \mathbb{R}^n \quad (2.18)$$

Una vez descrita la formulación lagrangiana, el modelo dinámico no lineal de un robot con tracción diferencial, donde el centro de masa se encuentra ubicado en el centro geométrico, está

descrito de la siguiente manera [18].

$$\begin{bmatrix} m & 0 \\ 0 & I_z \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (2.19)$$

Donde:

- m e I_z corresponden a la masa y el momento de inercia del robot móvil, respectivamente.
- \dot{v} es la aceleración lineal del robot.
- $\dot{\omega}$ es la aceleración angular.
- $\tau = [\tau_1 \quad \tau_2]^T$ corresponde al vector de fuerzas externas.

Teniendo en cuenta que 2.12 describe la posición del robot móvil y se encuentra a una distancia α del centro geométrico del mismo, se define la velocidad del robot móvil por

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos(\gamma) & -\alpha \sin(\gamma) \\ \sin(\gamma) & \alpha \cos(\gamma) \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2.20)$$

aplicando derivadas a \dot{x}, \dot{y} con respecto a γ , se puede concluir que

$$\begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) \\ -\frac{1}{\alpha} \sin(\gamma) & \frac{1}{\alpha} \cos(\gamma) \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} -\omega \sin(\gamma) & \omega \cos(\gamma) \\ -\frac{\omega}{\alpha} \cos(\gamma) & -\frac{\omega}{\alpha} \sin(\gamma) \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \quad (2.21)$$

de lo cual, sustituyendo 2.21 en 2.19, se obtiene el modelo dinámico que puede describirse en función de la posición 2.12 en el plano coordenado como

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} m \cos(\gamma) & m \sin(\gamma) \\ -\frac{I_z}{\alpha} & \frac{I_z}{\alpha} \cos(\gamma) \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} -m \omega \sin(\gamma) & m \omega \cos(\gamma) \\ -\frac{I_z}{\alpha} \omega \cos(\gamma) & -\frac{I_z}{\alpha} \omega \sin(\gamma) \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \quad (2.22)$$

Donde:

- 2.22 es válido únicamente en los intervalos de $\gamma \in [0 \quad \frac{\pi}{2})$, de modo que el $\cos(\gamma) > 0$

- $\dot{q} = [\dot{x} \ \dot{y}]^T$ corresponde a la velocidad del robot móvil en el plano
- $\ddot{q} = [\ddot{x} \ \ddot{y}]^T$ representa a la aceleración del robot móvil en el plano

Tomando en cuenta que el robot móvil cumple con ciertas hipótesis establecidas (el robot móvil se desplaza en un plano bidimensional horizontal), el término correspondiente a los pares de fuerzas gravitacionales son nulos, por concerniente el gradiente de la energía potencial es nulo [17, 18].

Capítulo 3

Identificación del sistema motriz

En este capítulo se realiza la identificación de los motores a través de técnicas de ingeniería de control, con el fin de obtener un modelo que describa su funcionamiento.

3.1. Descripción general del robot móvil

Las características básicas del robot móvil se especifican teniendo en cuenta los factores que van a inferir sobre el robot [6]. Para obtener un modelo matemático que satisfaga los objetivos propuestos el robot debe cumplir:

- La estructura del robot móvil debe ser liviano y simétrico entre todos los elementos que lo conforman.
- Los motores deben poseer el torque suficiente para poder mover toda su estructura.
- Las ruedas del robot deben tener un buen agarre para mejorar la fricción y además mejorar la estabilidad del mismo.
- El funcionamiento se realiza en espacios abiertos planos y no debe haber irregularidades en el suelo, con el fin de evitar errores de lectura de los sensores.
- El robot debe tener un nivel de autonomía aceptable.
- La tarjeta de procesamiento es un sistema embebido en tiempo real, la cual ayuda a la identificación de los motores.
- El robot posee un sensor de distancia para la evasión de obstáculos.

- Control de velocidad aplicado a cada uno de los motores del robot móvil.

3.2. Aproximación en bloques

La concepción básica de control de velocidad del robot móvil se comprende a través de 2 métodos principales:

En la Fig. 3.1 se observa un sistema de control en lazo abierto, lo que significa que la señal de salida no afecta al funcionamiento del sistema por completo, la información de las variables que controlan el proceso circulan en un solo sentido, desde la consigna del sistema de control hasta el proceso realizado, sin la verificación que se haya realizado dicha acción de control [17].

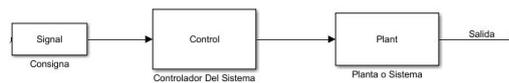


Figura 3.1: Diagrama de bloques de control en lazo abierto

En la Fig. 3.2 se observa un control en lazo cerrado, lo que significa que existe una retroalimentación de la señal controlada para una posterior verificación de la acción de control. La retroalimentación analiza la información de la salida (se realiza por medio de sensores), de un sistema y de ésta manera modificar si es conveniente la señal de entrada.

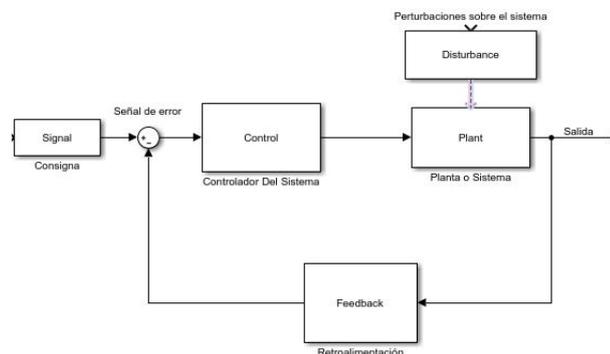


Figura 3.2: Diagrama de bloques de control en lazo cerrado

3.3. Métodos de aproximación a un modelo matemático

Existen dos métodos principales para obtener el modelo de un sistema [18].

3.3.1. Modelado teórico

Es un método analítico, en el que se recurre a leyes básicas de la física para describir el comportamiento dinámico de un fenómeno o proceso.

El modelado teórico tiene un campo de aplicación restringido a procesos muy sencillos de modelar, o aplicaciones en que no se requiera gran exactitud en el modelo obtenido. En muchos casos, además, la estructura del modelo obtenido a partir del conocimiento físico de la planta posee un conjunto de parámetros desconocidos y que sólo se pueden determinar experimentando sobre el sistema real. De ahí la necesidad de recurrir a los métodos de identificación de sistemas [17].

3.3.2. Identificación de un sistema

Es un método experimental que permite obtener el modelo de un sistema a partir de datos reales recogidos de la planta bajo estudio.

El método de identificación tiene algunas desventajas como son:

- El rango de validez puede ser limitado, trabajan solo con un tipo de entrada o un proceso concreto.
- Es difícil dar significado físico al modelo obtenido, debido a que los parámetros identificados no tienen relación.

3.4. Proceso de identificación de los motores

Se realiza una serie de pasos para obtener un modelo matemático que describa el funcionamiento de los motores del robot móvil.

3.4.1. Obtención de datos entrada - salida

Se necesita excitar la planta mediante una señal de entrada y registrar la evolución de sus datos tanto de entrada y salida durante un intervalo de tiempo.

Para la obtención de los datos se realiza mediante la tarjeta sbRIO, utilizando software de LabView, en el cual se varia PWM (modulación por ancho de pulsos), teniendo una respuesta de RPM (revoluciones por minuto) durante un intervalo de tiempo.

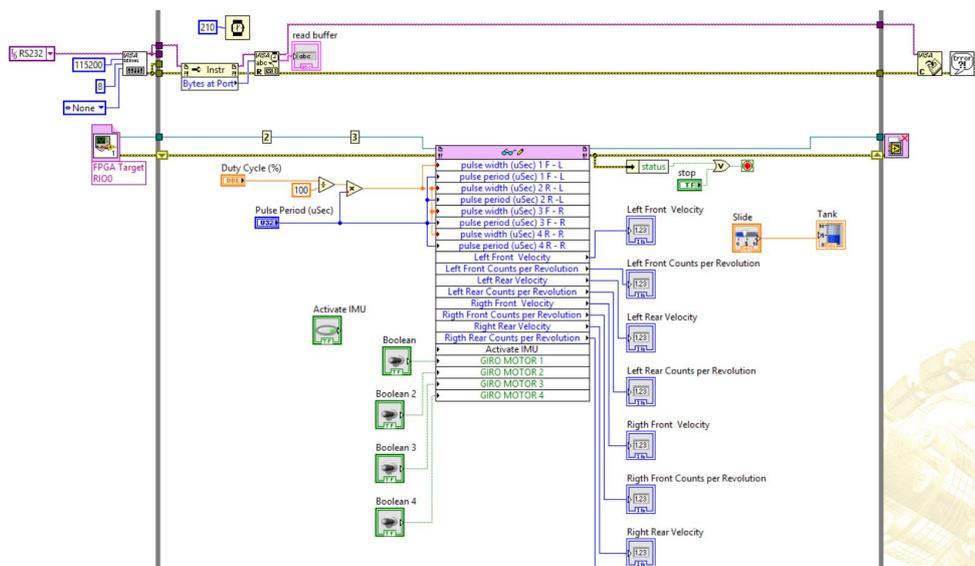


Figura 3.3: Programación en LabView para recolección de datos

3.4.2. Tratamiento de los datos registrados

Los datos registrados tienden a tener ruido debido a interferencias como los cables u otros factores, por lo cual se trata la señal por el método de la media móvil. Se trata con este método

ya que el ruido es considerablemente bajo [19].

Con los datos obtenidos se crea la Fig. 3.4, la cual representa la respuesta de la planta, en RPM vs Tiempo de muestreo.

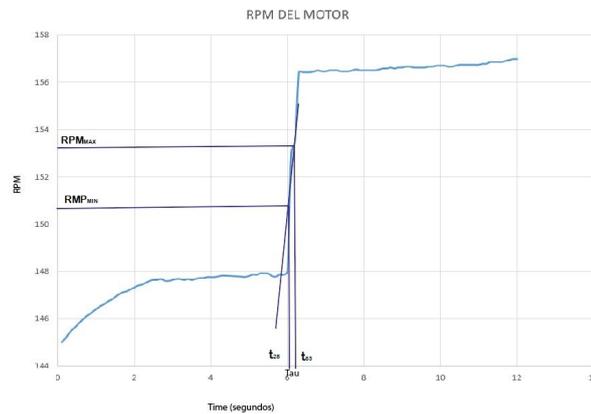


Figura 3.4: Respuesta de la planta, aplicando media móvil para disminuir ruido

En la Fig. 3.4 se observa que no existe un sobre-impulso, lo que quiere decir que se trata de un sistema de primer orden.

3.4.3. Elección de la estructura del modelo

Anteriormente se dedujo que el sistema es de primer orden, lo que significa que se necesita un método para determinar la función de transferencia de primer orden.

Cecil-Smith es un método gráfico en el que se obtiene el tiempo muerto del sistema, τ, θ , mediante la selección de dos puntos del transitorio: t_{28} y t_{63} que corresponden a los tiempos para alcanzar una respuesta fraccional de 28,3 y 63,2% respectivamente [19].

Respuesta de primer orden:

$$G(s) = \frac{K}{\tau s + 1} e^{-\theta s} \quad (3.1)$$

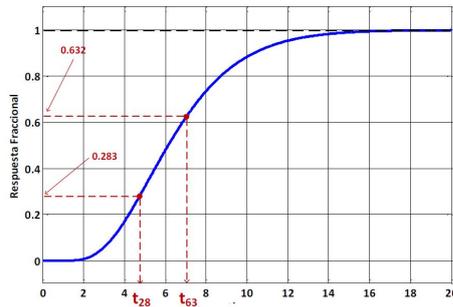


Figura 3.5: respuesta de primer orden Cecil-Smith

Donde:

- K (ganancia), es el cambio en Y que representa la salida del sistema en RPM, con respecto a la entrada a la entrada U que representa el cambio de PWM. Para el cálculo de K se lo realiza con la Fig. 3.4, donde se observa la variación de la salida.

$$K = \frac{\Delta Y}{\Delta U} = \frac{RPM_{Max} - RPM_{Min}}{PWM_{50\%} - PWM_{0\%}} = 2,76$$

- θ (tiempo muerto del sistema), tiempo que se demora en responder a un cambio de la variable de entrada, en este caso es igual al tiempo de muestreo.

$$\theta = 0,05$$

- τ (constante de tiempo del sistema). Cecil-Smith determina los puntos donde corresponde a t_{28} y t_{63} .

$$(\Delta Y) \times t_{63} + offset = P_1 = 153,37$$

$$(\Delta Y) \times t_{28} + offset = P_2 = 150,41$$

Con los valores P_1 y P_2 se realiza una proyección para calcular τ , dando un resultado

$$\tau = 0,1276$$

Reemplazando los valores de K , θ y τ en 3.1, se obtiene

$$G(s) = \frac{2,76}{0,1276s + 1} e^{-0,05s} \quad (3.2)$$

3.4.4. Validación del modelo matemático

Con la finalidad de validar el modelo matemático antes descrito, se realiza la identificación a través del software MATLAB [20].

1. Los datos obtenidos de velocidad vs duty cycle, se importa a MatLab, ver Fig. 3.6, a).
2. Se ingresa el comando *Ident* en *command Windows* de MatLab, debe aparecer la Fig. 3.6, b), se selecciona *import data*, luego se escoje la opción *Time domain data*.
3. En la siguiente Fig. 3.6, c), se necesita ingresar los nombres de los vectores de entrada y salida de la planta, así como el tiempo de muestreo según las configuraciones a la hora de tomar los datos, luego se selecciona la opción *Import*.
4. Se regresa a la ventana principal de *Ident* donde se muestra los datos importados, se debe seleccionar *Estimate* → *process models*, ver Fig. 3.6, d).
5. En *process models* aparece una ventana de configuración, en este caso la respuesta del motor no tiene un sobre impulso, se considera como un modelo de primer orden. Seleccionar la opción *estimate*. Como se puede observar en la Fig. 3.6, e), los valores de las constantes antes calculadas no son las mismas pero tienden a una aproximación de alrededor del 93 %.
6. Por último en la Fig. 3.6, f), se tienen las aproximaciones que se realizaron y de ver cual es la mejor respuesta.

Comparación de función de transferencia entre método de Cecil-Smith e identificación por MatLab

$$G(s) = \frac{2,76}{0,1276s + 1} e^{-0,05s}(a) \quad G(s) = \frac{2,6878}{0,12422s + 1} e^{-0,04925s}(b) \quad (3.3)$$

Donde *a* es igual a la función de transferencia calculado por el método de Cecil-Smith y *b* es igual a la función de transferencia obtenida por la opción *Ident* de MatLab. Se observa la

gran similitud entre las 2 funciones de transferencia que sobrepasa el 90% [20, 21].

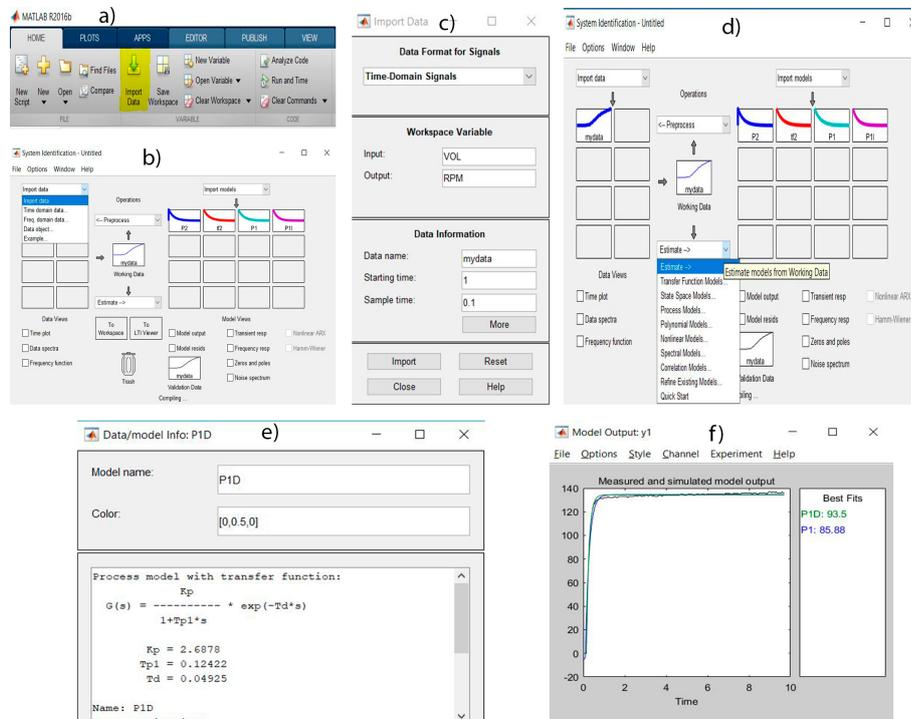


Figura 3.6: Identificación MatLab. a) barra de herramientas MatLab, b) ventana principal Ident, c) Importar datos a Ident, d) Tipo de estimación para el modelo, e) datos de K , τ , θ , f) estimaciones de la función de transferencia principal

Se realiza un modelo y una identificación por cada uno de los motores, para diseñar el controlador individual y en todas se obtuvo mas del 90% de similitud.

3.5. Discretización de la función de transferencia

El controlador que se necesita implementar es de tipo digital, se debe discretizar la función de transferencia que se obtuvo anteriormente, pasándola del dominio de s al dominio de z , para lo cual se utiliza la transformada z con un retenedor de orden cero ROC [22].

Se establece un periodo de muestreo de $T = 0,005$, para obtener

$$HG(z) = (1 - Z^{-1}) Z^N Z \left\{ \frac{G(s)}{s} \right\} \quad (3.4)$$

donde:

$$N \text{ es igual a } \frac{\theta}{T} \Rightarrow N = 100 \quad (3.5)$$

reemplazando 3.2 y 3.5 en 3.4 se obtiene

$$HG(z) = \left(\frac{z-1}{z} \right) z^{-100} Z \left\{ \frac{2,69}{s(0,125s+1)} \right\} \quad (3.6)$$

despejando la ganancia de 3.2 y operando se tiene

$$HG(z) = 2,69 \left(\frac{z-1}{z} \right) z^{-100} Z \left\{ \frac{8}{s(s+8)} \right\} \quad (3.7)$$

aplicando tablas de transformada z se puede realizar la transformación sin ningún problema

$$HG(z) = 2,69 \left(\frac{z-1}{z} \right) z^{-100} \left\{ \frac{(1 - e^{-0,008})z}{(z-1)(z - e^{0,008})} \right\} \quad (3.8)$$

operando y eliminando términos semejantes se determina

$$HG(z) = 2,69 z^{-100} \left[\frac{0,00797}{z - 0,992} \right] \quad (3.9)$$

por último se realiza la multiplicación para obtener la función en tiempo discreto.

$$HG(z) = \frac{0,0214}{z^{101} - 0,992^{100}} \quad (3.10)$$

Esta planta en tiempo discreto se utiliza para diseñar la estrategia de control de velocidad de cada una de las ruedas, además se realizan simulaciones de la estrategia de control [22].

Capítulo 4

Diseño del controlador de velocidad en lazo cerrado

Para implementar un control en lazo cerrado se debe plantear una serie de incógnitas como:

- ¿ Qué tipo de controlador se debe elegir ?
- ¿ Qué método utilizar para determinar los valores de los parámetros de control ?
- ¿ Cual es el mejor criterio de control para la planta ?

Entre otras incertidumbres que se deben tomar en cuenta, para lo cual se realiza un estudio de control sobre el modelo cinemático-dinámico y un control digital sobre la función de transferencia [17].

4.1. Control sobre el modelo cinemático

La mayoría de robots móviles poseen un controlador basado en el modelo cinemático, por lo general basta con este tipo de control para conocer la posición del robot en el plano, si el robot esta expuesto a fuerzas que intervengan en el desplazamiento del mismo o se desea controlar la velocidad en lugar de la posición, es necesario diseñar un controlador con el modelo dinámico y a su vez con el modelo cinemático [18].

Un método eficaz para solucionar el problema de control de velocidad en robots móviles, que incluya tanto al modelo cinemático como dinámico, es utilizar el método de BACKSTEP-

PING [23].

Backstepping es una teoría de control desarrollada por Petar V. Kokotovic para el diseño de controladores estabilizadores para sistemas dinámicos, el cual esta basado en 2 pasos [23].

- Encontrar la velocidades deseadas tales que estabilicen el modelo cinemático.
- Encontrar una ley de control que verifique que las velocidades reales se asemejen a las velocidades requeridas

En 4.1 se define el modelo cinemático que gobierna al robot móvil. Para resolver el primer paso del método de backstepping, se debe seguir el procedimiento propuesto por Fierro y Lewis [23], donde q satisface la trayectoria de referencia.

$$q = \begin{bmatrix} \cos(\gamma(t)) & 0 \\ \sin(\gamma(t)) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} \quad (4.1)$$

aplicando el marco local de referencia del robot, se define el error sobre las coordenadas del robot y se representa como

$$e = \begin{bmatrix} e1 \\ e2 \\ e3 \end{bmatrix} = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(t) - x \\ y(t) - y \\ \gamma(t) - \gamma \end{bmatrix} \quad (4.2)$$

donde $e1$ y $e2$ son los errores de posición del marco de referencia P_o, X_m, Y_m , $e3$ es el error de la orientación.

De 4.2 se representa la dinámica del error, para lo cual se debe derivar $e1$, $e2$, $e3$ con respecto a γ y se obtiene

$$\dot{e}1 = \dot{\gamma}e2 + \cos(\gamma)(\cos(\gamma(t))v(t) - \cos(\gamma)v(t)) + \sin(\gamma)(\sin(\gamma(t))v(t)) \quad (4.3)$$

aplicando la propiedad $\sin(\gamma)\sin(\gamma(t) + \cos(\gamma)\cos(\gamma(t) = \cos(\gamma(t) - (\gamma))$ se obtiene

$$\dot{e}_1 = \dot{\gamma}e_2 + v + v(t)\cos(e_3) \quad (4.4)$$

$$\dot{e}_2 = -\dot{\gamma}e_1 - \sin(\gamma)(\cos(\gamma(t)v(t) - \cos(\gamma)v) + \cos(\gamma)(\sin(\gamma(t)v(t) - \sin(\gamma)v) \quad (4.5)$$

aplicando la propiedad $\cos\gamma\sin\gamma(t) - \sin\gamma\cos\gamma(t) = \sin(\gamma(t) - (\gamma))$ se obtiene

$$\dot{e}_2 = v(t)(\cos\gamma\sin\gamma(t) - \sin\gamma\cos\gamma(t)) - \dot{\gamma}e_1 \quad (4.6)$$

$$\dot{e}_3 = \dot{\gamma}(t) - \omega \quad (4.7)$$

sustituyendo 4.7 en 4.4 y 4.6, se obtiene la dinámica del error de seguimiento.

$$\begin{aligned} \dot{e}_1 &= v(t)\cos(e_3) - v + \omega e_2 \\ \dot{e}_2 &= -\omega e_1 + v(t)\sin(e_3) \\ \dot{e}_3 &= \dot{\gamma}(t) - \gamma \end{aligned} \quad (4.8)$$

Aplicando el teorema de Fierro y Lewis [23] sobre la dinámica del error de seguimiento se obtiene como resultado

$$v_r = v(t)\cos(e_3) + K_1 e_1 + \omega e_2 \quad (4.9)$$

$$\omega_r = \omega(t) + K_2 v(t)e_2 + K_3 v(t)\sin(e_3) \quad (4.10)$$

donde, K_1, K_2, K_3 , son constantes bidimensionales positivas

Para tomar en cuenta la dinámica del robot móvil se determina un $\tau \in \mathbb{R}$

$$\lim_{t \rightarrow t_s} \|v_r(t) - v(t)\| = 0 \quad (4.11)$$

donde, $t_s < \infty$ es el tiempo que se demora el sistema en volverse estable

Se define una ley de control que obliga a las velocidades reales ser aquellas que se necesitan

en 4.9 y 4.10, aplicado la ley de control al modelo dinámico se obtiene

$$\begin{aligned}\tau &= M(q) [\dot{v}_r + K_s + \sin(\sigma) + K_p \sigma] + F(v) \\ \sigma &= v_r - v\end{aligned}\tag{4.12}$$

Donde:

- K_s y k_p son matrices diagonales positivas definidas de dimensiones 2×2
- σ es el vector de error de velocidad traslacional y rotacional

4.2. Sistema de control digital de velocidad

Es importante que un sistema móvil como en el que se está trabajando pueda tratar con las perturbaciones exteriores, además de adecuar la respuesta a las necesidades dinámicas que se presenten.

El principal objetivo es diseñar y obtener un controlador digital de velocidad que corrija las perturbaciones que tenga el sistema y que el error en estado estacionario tienda a 0.

De entre todos los tipos de controladores, se ha escogido el controlador PI en lazo cerrado para el control de velocidad del robot móvil, ya que el sistema es una planta de primer orden. Las ventajas de utilizar este tipo de controlador son las siguientes [17].

- Mayor estabilidad del sistema
- Reducción del error en estado estacionario
- Métodos de cálculo muy desarrollados
- Fácil de implementar

El sistema de control digital que se realiza se basa en el diagrama de la Fig. 4.1.

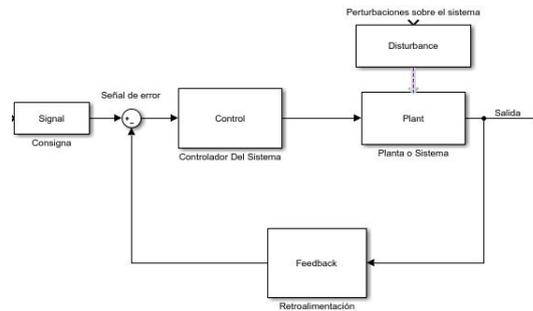


Figura 4.1: Diagrama de bloques de control en lazo cerrado

Una vez calculada la función de transferencia discretizada, se describe el sistema como en la Fig. 4.2, donde $G(z)$, es el controlador a implementar y $H(z)$ es la función de transferencia de la planta en el dominio de z

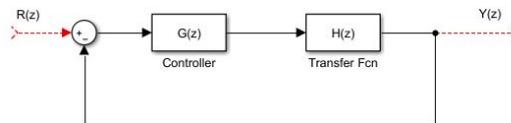


Figura 4.2: Diagrama de bloques de control en lazo cerrado en dominio de z

4.3. Métodos para diseñar un controlador

Se diseña el controlador por los métodos: Lugar geométrico de las raíces, método analítico, Ziegler Nichols - MatLab, con el fin de comparar el controlador de velocidad.

4.3.1. Lugar geométrico de las raíces

La estabilidad y respuesta transitoria de un sistema de control en lazo cerrado, esta relacionado con la ubicación de los polos de la función de transferencia en el plano complejo. Para lo cual se analiza el comportamiento del lugar geométrico de las raíces (LGR) del sistema de

control en lazo cerrado modificando sus parámetros (polos y ceros) [14].

El lugar geométrico de las raíces (LGR) es un método gráfico para identificar la posición de los ceros y polos del sistema de control en el plano complejo y se puede modificar con la compensación de los parámetros. La información que proporciona este método es utilizada para el análisis de estabilidad y funcionamiento del sistema [14].

En la Fig. 4.3 se puede deducir que el sistema se hace estable para valores de ganancia que estén entre $0 \leq k \leq 2,7$, para valores de ganancia mayores los polos se ubican por fuera del círculo.

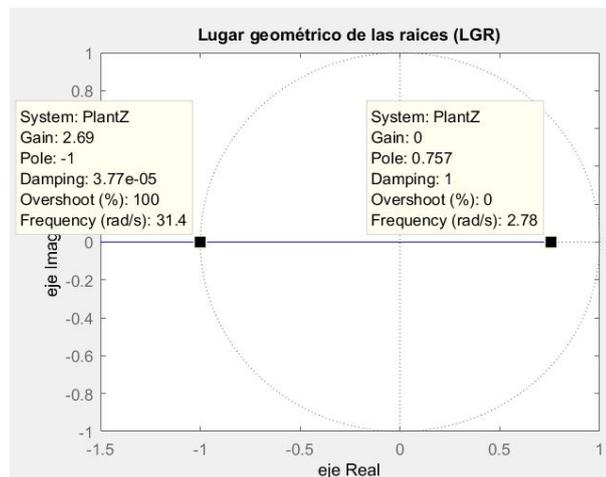


Figura 4.3: Lugar geométrico de las raíces

La compensación es la modificación de la dinámica del sistema con la finalidad de obtener los resultados planteados [18]. En este caso se utiliza el método de LGR, para obtener la respuesta deseada, una vez que se introduzca los ceros y polos que modifiquen la respuesta del sistema. La función de transferencia del compensador es

$$G(z) = K * \frac{z + a}{z + b} \quad (4.13)$$

donde, K es la ganancia de compensación (ajuste de compensación). El valor del cero de com-

pensación se representa por a y el valor del polo de compensación se representa por b . se coloca el polo discreto en $z = 1$ haciendo $b = -1$. Ésta posición permite una respuesta al escalón con error nulo en $t=10s$. El error al escalón puede suprimirse totalmente sin tener en cuenta la posición del cero ni la función de transferencia, como se indica en

$$\lim_{z \rightarrow 1} (z-1) \cdot \frac{z}{(z-1)} \cdot \left(1 - \frac{k \cdot (z+a) \cdot 0,6539}{(z+b)(z-0,757) + k \cdot (z+a) \cdot 0,6539} \right) \quad (4.14)$$

el valor de a afecta el comportamiento del sistema, para conseguir un LGR determinado primero se debe tener condiciones que debe cumplir el sistema como:

- Tiempo de estabilización $t_s = 1s$.
- Tiempo de muestreo $T = 0,1s$.
- Sobre impulso $M_p = < 5\%$.

Con las condiciones se puede calcular el resto de parámetros para obtener los polos del sistema.

- Coeficiente de amortiguamiento ξ .

$$M_p \geq e^{\frac{-\xi\pi}{\sqrt{(1-\xi^2)}}} \Rightarrow \xi \geq -\frac{\ln M_p}{\sqrt{(\ln M_p)^2 + \pi^2}} \quad (4.15)$$

$$\xi \geq 0,6901$$

- Frecuencia natural.

$$t_s \geq \frac{4}{\xi W_n} \Rightarrow \xi W_n \geq 4 \quad (4.16)$$

$$W_n \leq 5,7962$$

- Tiempo de subida.

$$t_r = \frac{1,8}{W_n} \quad (4.17)$$

$$t_r = 0,31$$

Una vez determinado los valores con las especificaciones deseadas se calcula la posición del polo

$$P_{1,2} = \xi W_n \pm W_n \sqrt{1 - \xi^2} i \quad (4.18)$$

$$P_1 = -4 + 4,1948P_2 = -4 - 4,1948$$

4.3.1.1. Controlador PI

Al insertar un controlador PI en el sistema, el LGR de la función de transferencia toma la forma de un círculo centrado en el cero del controlador, el diseño del controlador termina cuando se ajusta la ganancia del controlador para que los polos de lazo cerrado se encuentren en la posición deseada [18].

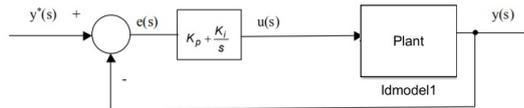


Figura 4.4: Diagrama de bloques de control PI en lazo cerrado

Donde:

- $C(s) = K_p + \frac{k_i}{s}$ es el controlador PI del sistema.
- K_p Parte proporcional que multiplica al error por una ganancia constante.
- $\frac{k_i}{s}$ Parte integral que integra el error y multiplica por una constante.
- Plant es el modelo del sistema $G(s) = \frac{k}{\tau s + 1}$.

La función de transferencia en lazo cerrado se muestra en

$$H(s) = \frac{y(s)}{e(s)} = \frac{C(s)G(s)}{1 + C(s)G(s)} \quad (4.19)$$

reemplazando valores de $C(s)$ y $G(s)$ en 4.19 se obtiene

$$H(s) = \frac{\left(\frac{k_p s + k_i}{s}\right) \left(\frac{k}{\tau s + 1}\right)}{1 + \left(\frac{k_p + k_i}{s}\right) \left(\frac{k}{\tau s + 1}\right)} \quad (4.20)$$

resolviendo y simplificando 4.20, se obtiene la función de transferencia en lazo cerrado del modelo de la planta y el controlador.

$$H(s) = \frac{\left(\frac{k}{\tau}\right) (k_p s + k_i)}{s^2 + \left(\frac{1 + k_p k}{\tau}\right) s + \left(\frac{k_i k}{\tau}\right)} \quad (4.21)$$

La ecuación característica de segundo orden es 4.22 y la ecuación característica de la planta es el denominador de 4.21, se realiza una igualdad de coeficientes de las 2 ecuaciones características [20]

$$s^2 + 2\xi \omega_n s + \omega_n^2 \quad (4.22)$$

$$s^2 + 2\xi \omega_n s + \omega_n^2 = s^2 + \left(\frac{1 + k_p k}{\tau}\right) s + \left(\frac{k_i k}{\tau}\right)$$

igualando el primer coeficiente y reemplazando los valores calculados anteriormente se obtiene k_p .

$$2\xi \omega_n = \left(\frac{1 + k_p k}{\tau}\right)$$

$$k_p = 0,42389$$

igualando el primer coeficiente y reemplazando los valores calculados anteriormente se obtiene k_i .

$$\omega_n^2 = \left(\frac{k_i k}{\tau}\right)$$

$$k_i = 1,6687$$

con estos valores se construye la función de transferencia del controlador PI

$$C(z) = 0,734 + 2,16 \cdot \frac{0,1}{z - 1} \quad (4.23)$$

Para obtener la ecuación definitiva del controlador y la planta en lazo cerrado se reemplaza

los datos en 4.19 dando como resultado

$$H(z) = \frac{0,48009 (z - 0,7052) (z - 0,7571) (z - 1)}{(z - 1) (z - 0,7571) (z^2 - 1,277z + 0,4186)} \quad (4.24)$$

La respuesta al escalón unitario de la función de transferencia 4.24, es la que se muestra en la Fig. 4.5, en la que se demuestra que el sobre pico es menor del 5% y no existe error en estado estacionario.

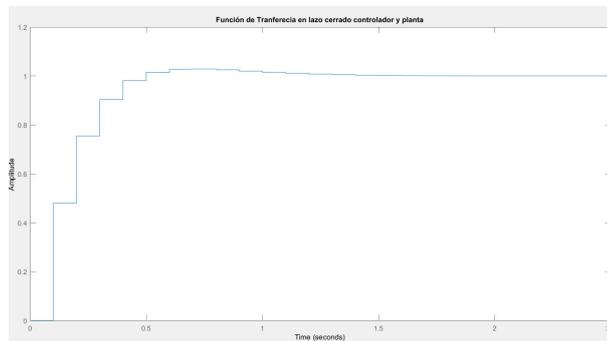


Figura 4.5: Respuesta al escalón unitario del control y planta

4.3.2. Método analítico

Un sistema de control en lazo cerrado ante una entrada de escalón unitario muestra el tiempo de estabilización mínimo posible, esto se conoce como respuesta "Dead-beat" (respuesta con oscilación muerta), esto se logra con gran facilidad con un controlador digital ya que muchos esquemas que han sido complicados controlar con controles analógicos, se ha resultado mediante control digital [17]

El objetivo en este apartado es encontrar una ecuación $C(z)$ para que la función de transferencia en lazo cerrado tenga el comportamiento deseado. Se parte de

$$H(z) = \frac{C(z)G(z)}{1 + C(z)G(z)} \quad (4.25)$$

despejando $C(z)$

$$C(z) = \frac{H(z)}{G(z) + G(z)H(z)} \quad (4.26)$$

se debe establecer parámetros para el diseño de $C(z)$

- $C(z)$ debe de corresponder a un sistema físicamente realizable, el orden del denominador tiene que ser igual o mayor al numerador
- $C(z)$ corresponde a un sistema estable dado como en 4.25
- Error de seguimiento nulo a entrada de escalón unitario

A partir de las especificaciones propuestas anteriormente se realiza el diseño analítico del controlador $C(z)$ [24].

4.3.2.1. Error de seguimiento nulo a entrada escalón

Sabiendo que el escalón unitario posee la forma de 4.27 [18], y para una respuesta de n muestras se obtiene

$$R(z) = \frac{1}{1 - z^{-2}} \quad (4.27)$$

$$F_1(z) = a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n} \quad (4.28)$$

se calcula el límite en el infinito para que tienda a ser igual a cero

$$\lim_{z \rightarrow 1} (1 - z^{-1}) \cdot F_1(z) = \lim_{z \rightarrow 1} (1 - K(Z)) \quad (4.29)$$

resolviendo se obtiene la primera condición del diseño

$$(1 - K(Z)) = (1 - z^{-1}) \cdot F_1(Z) \quad (4.30)$$

4.3.2.2. Estabilidad

Se parte de 4.25 y teniendo en cuenta que se conoce los valores de $G(z)$, se puede decir que $H(z)$ contiene como polos los ceros de $G(z)$ y como ceros los polos de $G(z)$, aplicando la

primera condición $C(z)$ anula los polos inestables de $G(z)$ y así se obtiene la segunda condición

$$K(Z) = F_2(Z) \quad (4.31)$$

4.3.2.3. Respuesta Dead-beat

F_1 y F_2 son polinomios de z^{-1} , el grado del exponente z es el número mínimo de muestras que se necesitan para llegar a un régimen estacionario, de 4.25 y las condiciones de diseño se llega a plantear

$$\begin{aligned} K(z) &= a_1 z^{-1} \\ F_1(z) &= 1 + b_0 \\ F_2(z) &= c_1 z^{-1} \end{aligned} \quad (4.32)$$

reemplazando 4.32 y las condiciones de diseño en 4.25, se obtiene en función de F_1 y F_2 , para lo cual se debe de encontrar los valores para c_1 y b_0 con la ayuda de las condiciones

$$C(z) = \frac{H(z)}{G(z)(1-K(z))} = \frac{1 - 0,7571z^{-1}}{0,6538z^{-1}(1-z^{-1})} \cdot \frac{c_1 z^{-1}}{1+b_0} \quad (4.33)$$

aplicando las condiciones de diseño se obtiene el valor de c_1 y b_0 , dando como resultado

$$c_1 = 1 \quad b_0 = 0$$

por lo tanto el controlador queda de la forma

$$C(z) = \frac{1 - 07571z^{-1}}{0,6538z^{-1}(1-z^{-1})} \quad (4.34)$$

De esta manera se grafica el LGR del sistema en lazo cerrado como se muestra en la Fig. 4.6, donde se observa que el controlador coloca un polo en $z = 1$ con la finalidad de eliminar el error en estado estacionario y además del cero que elimina los efectos del polo.

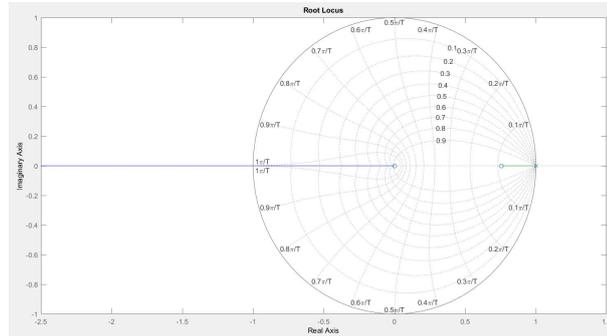


Figura 4.6: LGR del controlador por método analítico

Una vez encontrada la función de transferencia del controlador se encuentra analíticamente la función de lazo cerrado que incluye al controlador y planta dando como resultado

$$H(z) = \frac{0,48z - 0,4611}{z^2 - 1,2751z + 0,418} \quad (4.35)$$

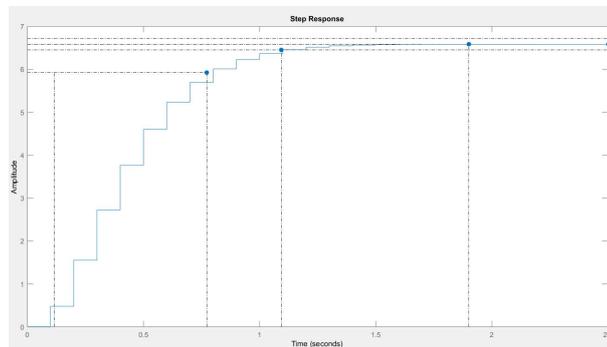


Figura 4.7: Respuesta al escalón unitario

En la Fig. 4.7 se observa que el controlador disminuye el tiempo de estabilización a 1s y elimina el sobre impulso.

4.3.3. Ziegler Nichols - MatLab

Una vez obtenida la función de transferencia en el dominio de z , se analiza la estabilidad mediante la utilización del programa MatLab.

En la Fig. 4.8 se observa el comportamiento de la planta, teniendo un periodo transitorio logra estabilizarse en 2,5 aprox. de ahí la necesidad de crear un controlador que logre la estabilización en 1 [14]

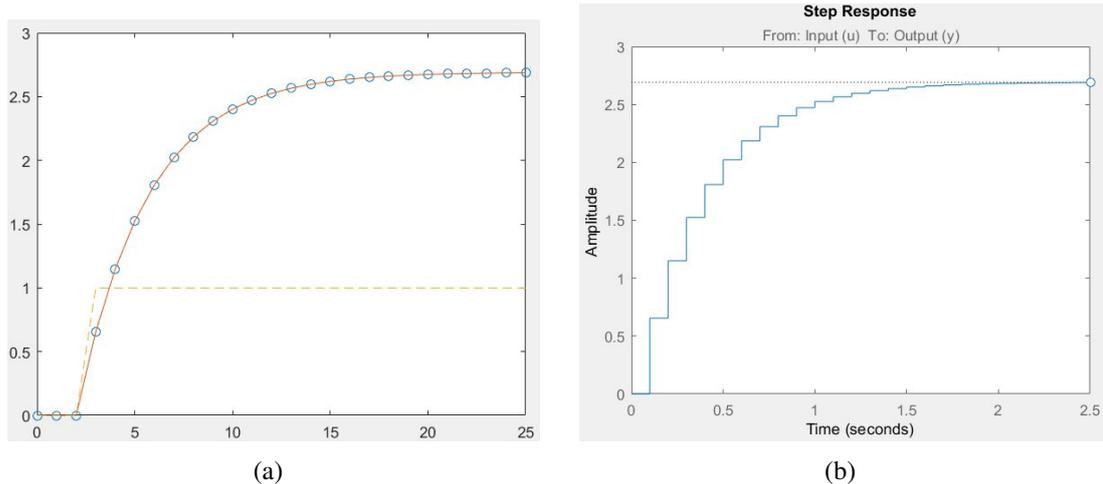


Figura 4.8: Respuestas al escalón unitario, a) respuesta al escalón, b) comando Step

Para aplicar este método se determinan: las raíces, polos, y respuesta en lazo abierto y lazo cerrado.

■ Lugar geométrico de las raíces

En la Fig. 4.9 a), se observa que el sistema se estabiliza con valores de ganancia que varían entre 0 y 2,7, si se aplica una ganancia fuera de este rango el sistema se vuelve inestable.

■ Ubicación de ceros y polos

De 4.34 se puede deducir la existencia de un solo polo, el mismo está dentro del círculo unitario lo que quiere decir que el sistema es estable, tal y como se muestra en la Fig. 4.9 b).

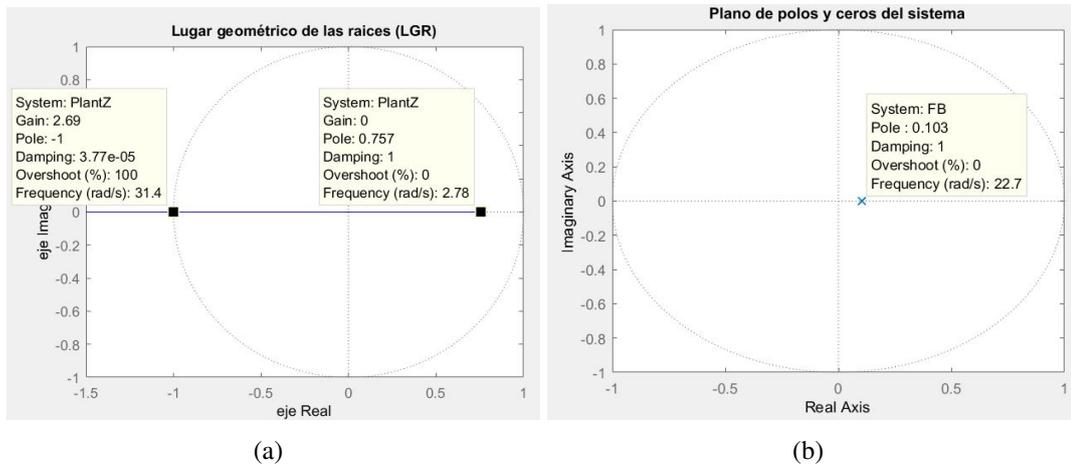


Figura 4.9: a) lugar geométrico del sistema en lazo cerrado, b) polos y ceros del sistema

4.3.3.1. Control PI

Para determinar el controlador PI, se analiza la respuesta del sistema en lazo abierto para una entrada escalón, la cual se obtuvo anteriormente y se estableció que la respuesta es estable, ya que el método de Ziegler-Nichols solo es válido para sistemas estables a lazo abierto. Para determinar el controlador PI por el método de Ziegler-Nichols se sigue el siguiente procedimiento[18].

- Aplicar a la planta sólo control proporcional con ganancia K_p pequeña.
- En un instante dado aplicar un cambio escalón a la entrada.
- Registrar la respuesta de la salida hasta que se estabilice en el nuevo punto de operación del sistema.
- Ajustar los parámetros del controlador PI de acuerdo a la tabla 4.1.

Tabla 4.1: Parámetros para calculo de controladores P, PI, PID

Tipo de controlador	K_p	T_r	T_d
P	$\frac{T}{L}$	ínf	0
PI	$0,9\frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2\frac{T}{L}$	$2L$	$\frac{1}{2}L$

Los valores de la Tabla 4.1 corresponden a una respuesta como se observa en la Fig. 4.10 donde:

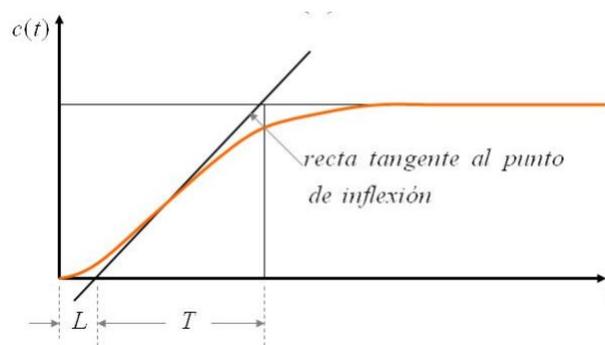


Figura 4.10: Respuesta de primer orden al escalón unitario

- L es el tiempo donde se realiza una intersección entre la línea tangente con respecto al eje horizontal que corresponde a la variable independiente del tiempo.
- T es la diferencia entre L y el tiempo donde la señal se vuelve estable pero con respecto a la línea tangente

Del capítulo 2 se puede obtener el valor de L y T con la gráfica de referencia para una respuesta de primer orden, dando como resultado $L = 0,579$

$$T = 0,2895$$

reemplazando los valores de L y T en las formulas del controlador PI se obtiene las ganancias proporcional e integral $K_p = 0,45$

$$T_r = 1,93$$

una vez determinado las ganancias, se sustituye en 4.36 que es la equivalente de la función de transferencia del controlador y se obtienen

$$C(s) = \left(K_p + T_r \frac{1}{s} \right) \quad (4.36)$$

$$C(s) = \left(0,45 + \frac{1,93}{s} \right) \quad (4.37)$$

Se realiza la simulación del diagrama dando como resultado la Fig. 4.11, el controlador cumple con las especificaciones [24].

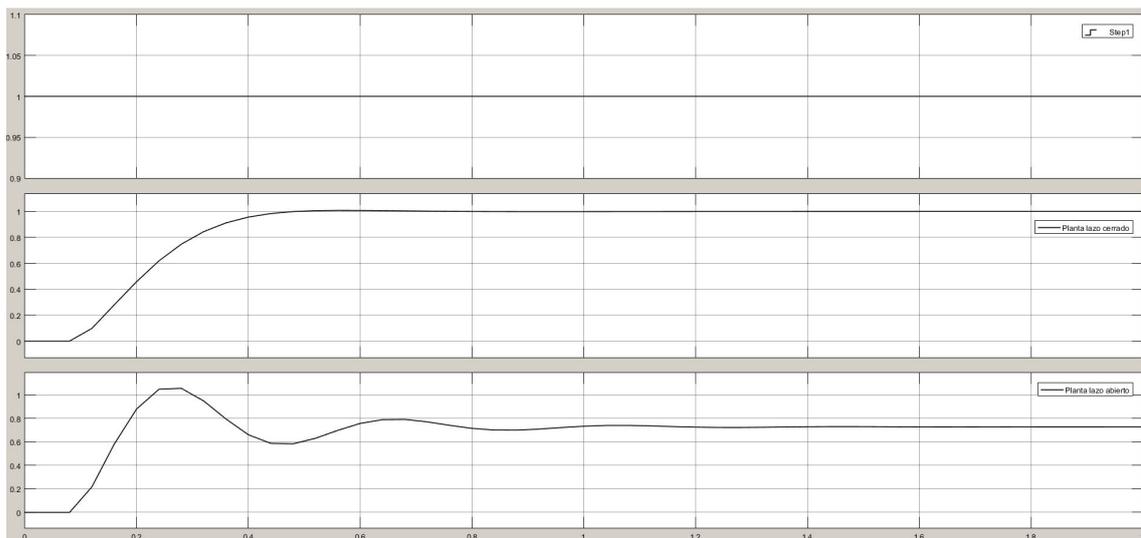


Figura 4.11: Respuesta del sistema al escalón

4.3.4. Controlador escogido

Una vez determinado los métodos de obtención del controlador, se necesita escoger uno que cumpla con las especificaciones mínimas requeridas para mejorar la dinámica del sistema, para lo cual se realiza una comparación de la respuesta al escalón unitario del controlador y planta.

Dando como resultado, que todos los métodos cumplen con las especificaciones dadas, esto quiere decir que no importa que controlador se escoja siempre se obtiene respuestas similares, ahora bien para determinar cual controlador se implementará en el sistema se escoge el que resulte ser mas fácil de implementar, ya que esto representa un ahorro de tiempo al momento de programar el controlador.

El método escogido es el de Ziegler Nichlos - MatLab, con el controlador obtenido de este método se implementa sobre los motores del robot móvil.

Capítulo 5

Implementación del controlador sobre el robot móvil

En la etapa de implementación del controlador se realiza en el software LabView de National Instrument, ya que en trabajos anteriores se trabajó en esta misma plataforma.

5.1. LabView

Según José Lajara LabView “es el acrónimo de Laboratory Virtual Instrument Engineering Workbench que en español quiere decir Laboratorio de Ingeniería de Instrumentos Virtuales de Laboratorio. Es un lenguaje y a la vez un entorno de programación gráfica en la que se puede crear aplicaciones de una forma rápida y sencilla” [13].

A continuación se presenta una lista de algunas aplicaciones que se puede realizar por medio de esta plataforma [13].

- Diseños de sistemas embebidos de control y monitoreo.
- Automatización de sistemas de pruebas y validación.
- Campo aeroespacial es utilizado para la instrumentación.
- Monitorear y dar movimiento a robots.

5.2. Sistema en LabView

El programa diseñado en LabVIEW solo se ejecuta si el ordenador al cual se va a conectar el robot móvil, posee instalado el software y módulos que se presentan en la siguiente lista [11].

- Sistema Operativo Windows 10.
- Software LabVIEW de National Instrument versión 2015.
- Módulos de:
 - Robótica.
 - Tiempo real .
 - Control y supervisión.
 - FPGA.

Tanto el Software como los módulos deben ser instalados en la version 2015 sp1.

5.3. Interfaz gráfica

La interfaz humano-robot con la cual el usuario puede manipular al robot de una forma fácil, la cual consta de 8 secciones que se detallan a continuación.

5.3.1. Visualización del estado de las baterías

Sección donde se encuentra una gráfica del estado actual del voltaje de las baterías y unos indicadores teniendo como lectura máxima 10v cada una que significa que está a carga máxima. Es importante mencionar que si el estado de carga de las baterías es menor a 6v el hmi genera un mensaje de advertencia que indica que el estado de la batería es baja y para el funcionamiento del sistema. Fig 5.1, 1.

5.3.2. Visualización de la velocidad de los motores

En esta sección se muestra una gráfica de las velocidades de cada uno de los motores además existen indicadores numéricos que muestran el valor de la velocidad en rpm. Los motores están divididos en *RF* (motor frontal derecho), *LF* (motor frontal izquierdo), *RR* (motor posterior derecho), *LR* (motor posterior izquierdo), ver Fig 5.1, 2.

5.3.3. Maniobrabilidad del robot

Una sección en donde se encuentra los controles para poder tener maniobrabilidad sobre el robot en donde a través de cuatro botones damos movilidad al robot, cabe mencionar que el control de navegación también se lo puede tener por teclado a través de las teclas F2, END, F1, F3 y un puntero de deslizamiento horizontal el cual sirve para controlar la velocidad de los motores. Fig 5.1, 3.

5.3.4. Controlador PI

En esta sección es donde se establece el controlador para cada uno de los motores, se puede modificar el valor velocidad de referencia en los cuales se desea que el robot se establezca, además de indicar la velocidad en $\frac{m}{s}$ y el error del valor de referencia y el valor leído por los encoders. Fig 5.1, 4.

5.3.5. Visualización de sensor IMU

En esta sección se puede apreciar la lectura del sensor de medición inercial IMU en donde a través de una gráfica podemos observar las señales de todos sus ejes tanto del acelerómetro, giroscopio y magnetómetro seguido de unos indicadores numéricos en el que se visualiza los valores de los ángulos. Fig 5.1, 5.

5.3.6. Lectura de codificadores

Unos indicadores gráficos visualizan el estado lógico de cada uno de los codificadores de sus ruedas y además un indicador numérico muestra el conteo incremental de los pulsos de cada rueda del robot. Fig 5.1, 6.

5.3.7. Análisis odométrico

Se estima la localización del robot móvil en una gráfica “x” e “y” con unos indicadores, además se muestra la distancia recorrida total, ángulo de posición y distancia recorrida por cada rueda derecha e izquierda igualmente se tiene un indicador del compás en donde indica la orientación del robot móvil. Fig 5.1, 7.

5.3.8. Sensor ping

Sección de visualización del sensor ping: en esta sección se puede observar mediante un indicador la posición del servomotor con un rango de 90° y -90° y la lectura del sensor ping interpretando la distancia en una gráfica. Fig 5.1, 8.

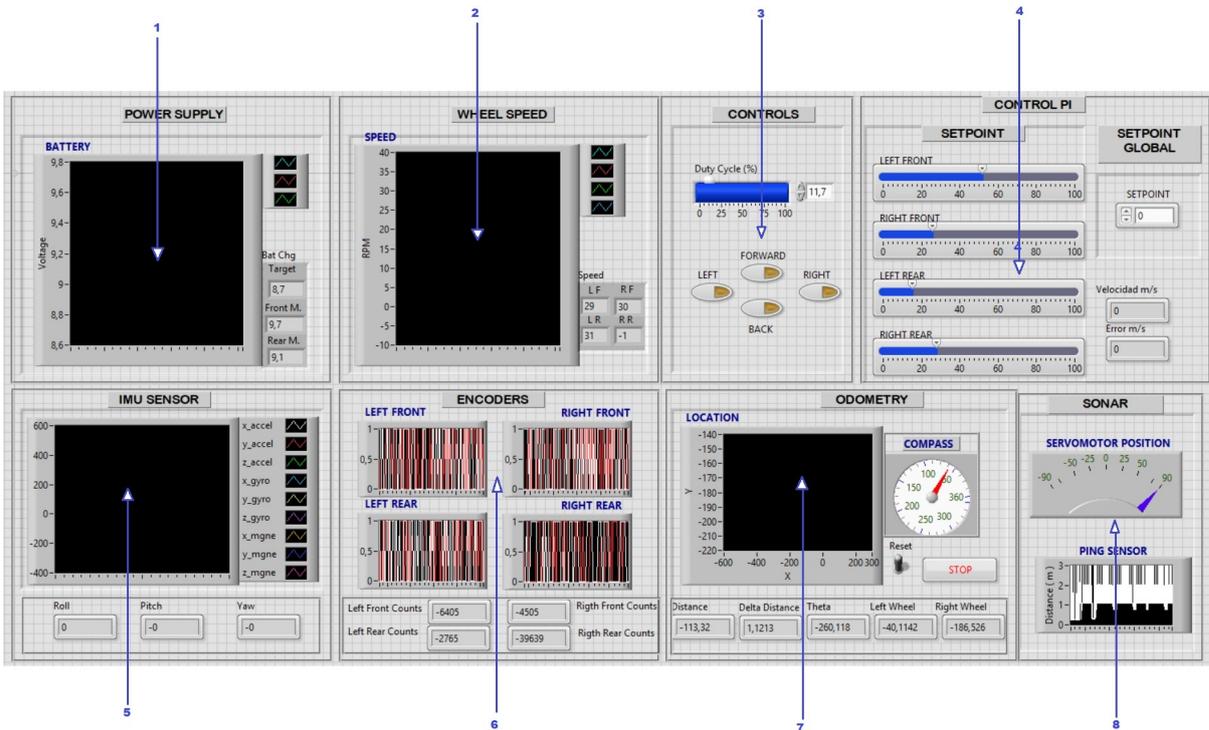


Figura 5.1: Interfaz gráfica

5.4. Implementación del controlador en el software LabVIEW

Para el desarrollo e implementación del controlador en la plataforma LabVIEW, se debe de tener en cuenta todo el diseño que se ha tratado en el capítulo anterior.

La Fig. 5.2 muestra un diagrama de flujo básico del control de velocidad del robot móvil donde:

- k_p, k_i constantes del controlador *PI* calculadas con anterioridad.
- **Set Point** es el valor de la velocidad deseada que se puede modificar manualmente, velocidad a la que se desea que se establezca el robot.
- **Velocidad** $\frac{m}{s}$ variable donde se almacena el valor de la velocidad real de las ruedas.

- **Control PI** SubVi creado en LabVIEW donde se produce el control de velocidad.
- **Señal de control** señal de control que proviene del SubVI del controlador PI.

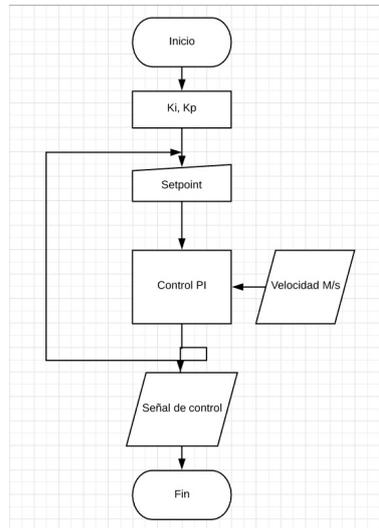


Figura 5.2: Diagrama de flujo control de velocidad

El controlador implementado en el robot móvil está programado en un subVi, de las herramientas del software LabView destinadas para este tipo de controlador, dentro del RT de la tarjeta de código embebido donde se realiza el cálculo en tiempo real para el controlador, ver Fig. 5.3.

En la Fig. 5.3 se puede observar el diagrama del controlador al cual se debe aplicar las constantes k_p , k_i , setpoint de referencia (velocidad deseada) y el valor de velocidad de las ruedas.

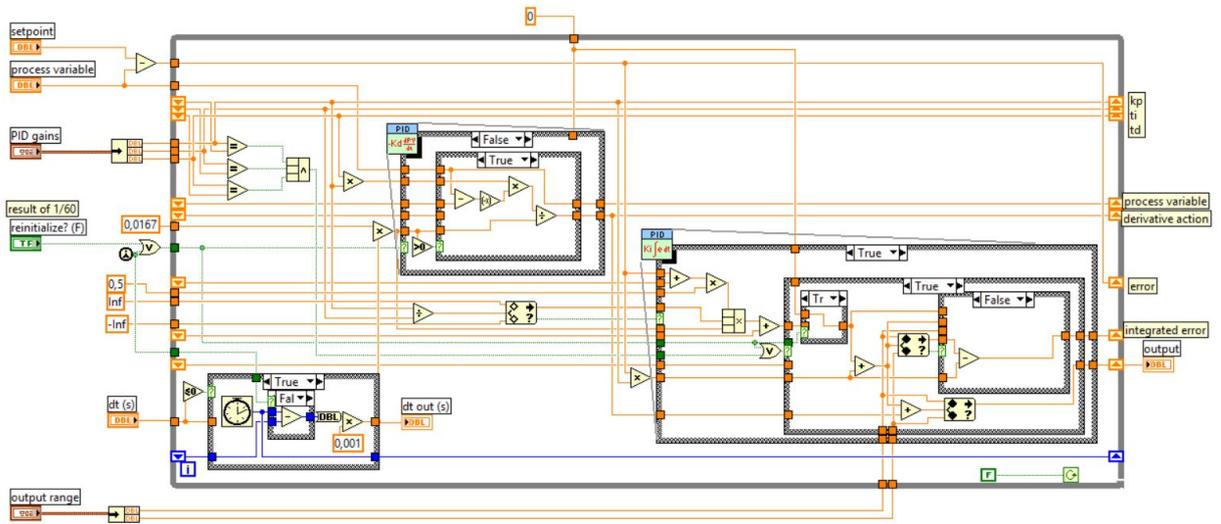


Figura 5.3: Diagrama de bloques del controlador

5.4.1. Implementación del programa en el computador

El subVi del controlador que se implementa dentro del RT se presenta en la Fig. 5.4 y consta de:

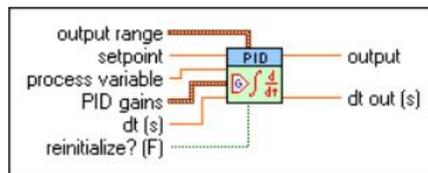


Figura 5.4: SubVi control PI

- **Output range:** el rango de salida del controlador, que en este caso es de 0 a $120 \frac{m}{s}$.
- **Setpoint:** especifica el valor de referencia.
- **Process variable:** especifica el valor medido de la variable de proceso que se controla.
- **PID gains:** Constantes del controlador *PI*
- **dt(s):** especifica el tiempo del ciclo del bucle interno del VI.

- **reinitialize:** si es verdadero el Vi se reinicia.
- **Output:** devuelve la salida de control del algoritmo *PI* que luego se aplica al proceso controlado.
- **dt out(s)** devuelve el intervalo de tiempo que esta corriendo el VI.

En la Fig. 1 se observa el SubVi del controlador que se realizó dentro del RT de la tarjeta haciendo uso de un elemento para abrir la referencia a un *FPGA.vi*, del cual se puede leer los datos provenientes de la tarjeta *SbRIO-9631*, tanto entradas como salidas que actúan sobre el robot móvil. Además dentro del RT se encuentran la lectura del sensor *IMU*, control manual de direccionamiento del robot, análisis de la odometría con las señales de los codificadores para obtener una posición en *x, y*. Esta información se puede reforzar en [6].

Capítulo 6

Simulaciones y Pruebas de funcionamiento

En este capítulo se presenta las simulaciones correspondientes a los modelos matemáticos descritos anteriormente, además de los resultados obtenidos de las pruebas de funcionamiento de los motores, los sensores y actuadores que están implementados en el robot.

Todo esto se realiza bajo ciertos parámetros y condiciones los cuales tienen la finalidad de determinar y comprobar el correcto funcionamiento de la plataforma.

6.1. Simulaciones

Las simulaciones están divididas en 2 grupos y se realizan con la ayuda del software MatLab.

6.1.1. Modelo cinemático

Para simular el modelo cinemático del robot se programó el script 1 en el software MatLab, que permite verificar el comportamiento del modelo, del cual se obtuvo los siguientes resultados:

Si se aplica la misma velocidad a cada una de las ruedas se obtiene como resultado una trayectoria rectilínea a lo largo del eje X, como se observa en la Fig. 6.1.

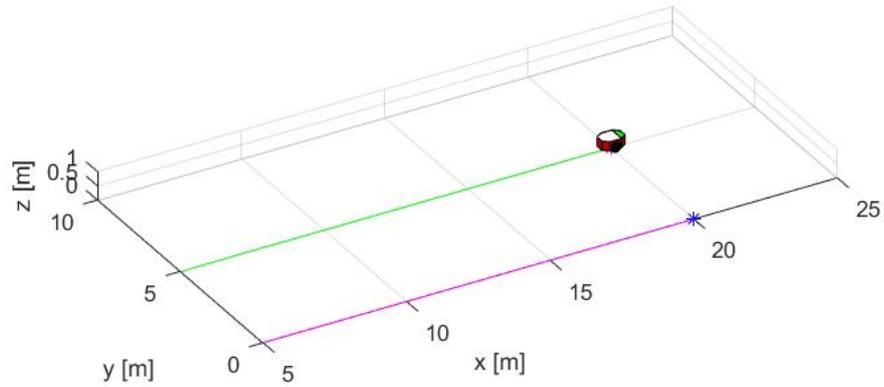


Figura 6.1: Respuesta a velocidades iguales

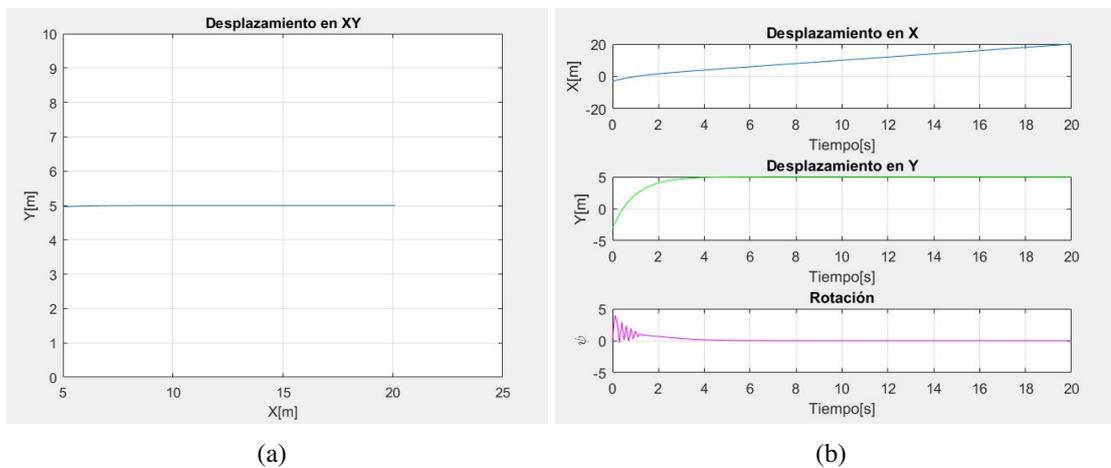


Figura 6.2: Respuesta a velocidades iguales, a) desplazamiento general, b) desplazamiento en los vectores x, y, ϕ

En la Fig. 6.2 a), se muestra el desplazamiento total del robot, dando una línea horizontal en el eje x y la Fig. 6.2 b), muestra cada uno de los vectores de desplazamiento y rotación.

Si se aplica una velocidad mayor a una de las ruedas, esto permite obtener una trayectoria

rotacional con desplazamiento circular, como se observa en la Fig. 6.3

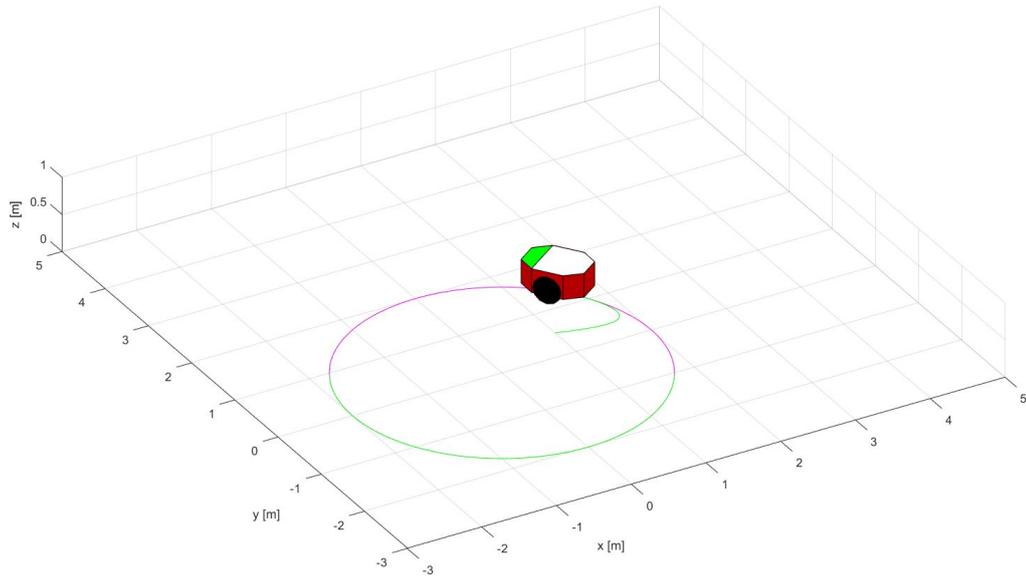


Figura 6.3: Respuesta a velocidades distintas

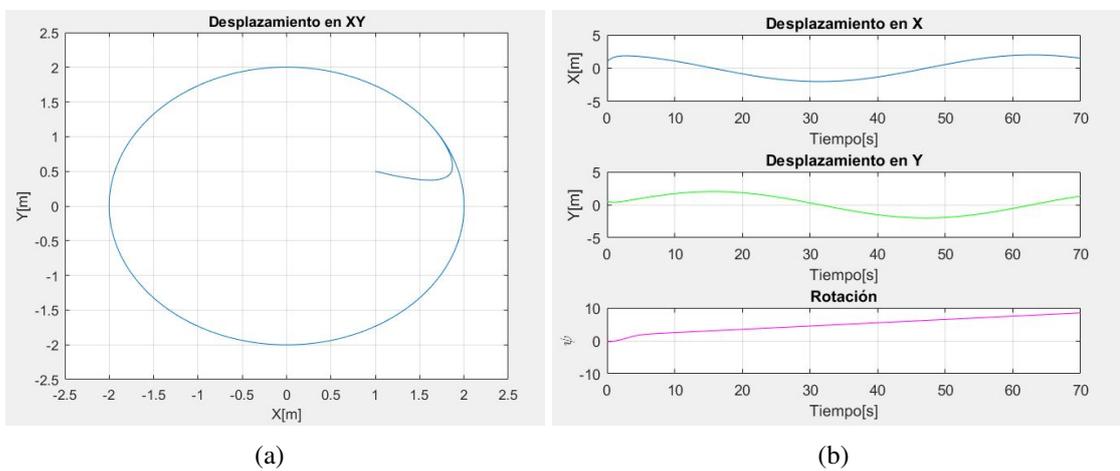


Figura 6.4: Respuesta a velocidades distintas, a) desplazamiento general, b) desplazamiento en los vectores x, y, ϕ

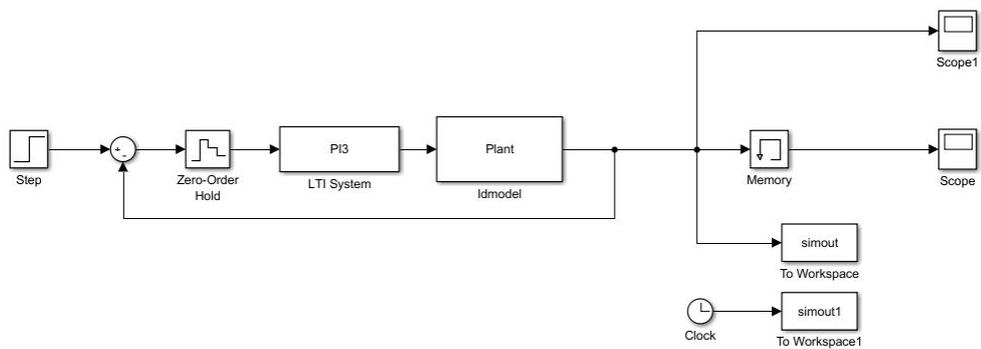
El robot móvil puede seguir cualquier trayectoria que se desee, para lo que es necesario saber que velocidad aplicar a cada una de las ruedas para ello se necesita encontrar las ecuaciones que

determinan la trayectoria.

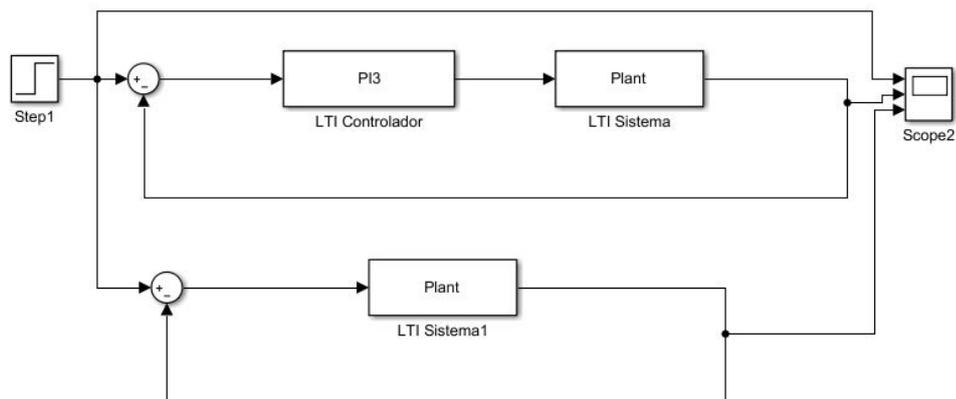
6.1.2. Función de transferencia

Para simular el modelo matemático en el dominio de s y z se utiliza el software MatLab y Simulink de la misma plataforma, esto con la finalidad de verificar el comportamiento del modelo.

En MatLab se programa un script 2 donde se realizan las operaciones, además se programa un diagrama de bloques en Simulink del cual se obtienen los datos de salida del modelo con el controlador ya implementado. esto se lo puede observar en la Fig. 6.5.



(a)



(b)

Figura 6.5: Diagrama de bloques en Simulink, a) tiempo discreto, b) tiempo continuo

La Fig. 6.5 a), muestra el control aplicado a la planta en tiempo discreto, la Fig. 6.5 b), muestra el diagrama en tiempo continuo.

Los resultados se observan en la Fig. 6.6, el tiempo de estabilización disminuye al implementar el controlador, así como el sobre impulso.

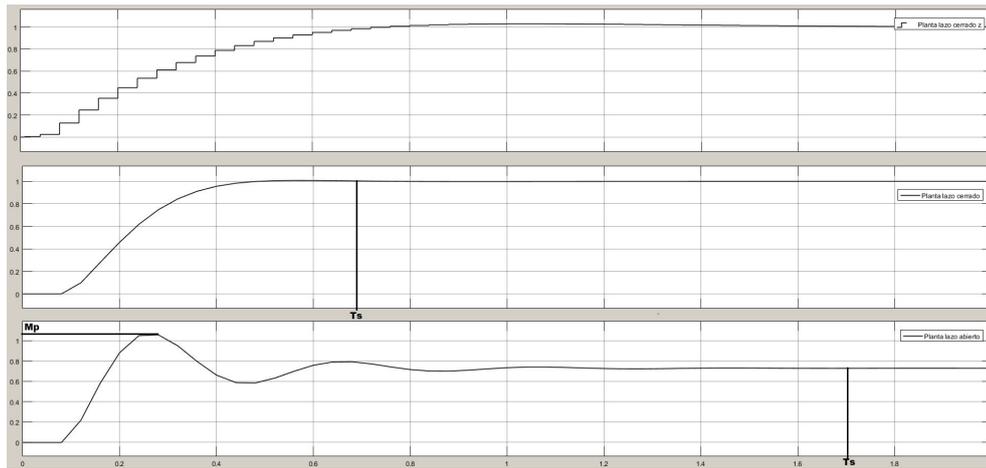


Figura 6.6: Respuesta de diagrama de bloques

6.2. Prueba en los sensores

Con la finalidad de comprobar el correcto funcionamiento de los sensores implementados sobre el robot móvil se realiza una prueba a cada uno de los sensores.

6.2.1. Lectura de codificadores

Esta prueba consta de hacer girar una vuelta completa la rueda para comprobar el funcionamiento del algoritmo implementado en la FPGA a través de conteos de pulsos. En la Fig. 6.7 se observa el acople del motor a la rueda y el contador de pulsos programado en LabView.

Los resultados obtenidos se presentan en la Tabla 6.1, donde se indica la cantidad de pulsos contados tanto en sentido horario y anti-horario, las pruebas se realizan a cada una de las ruedas.

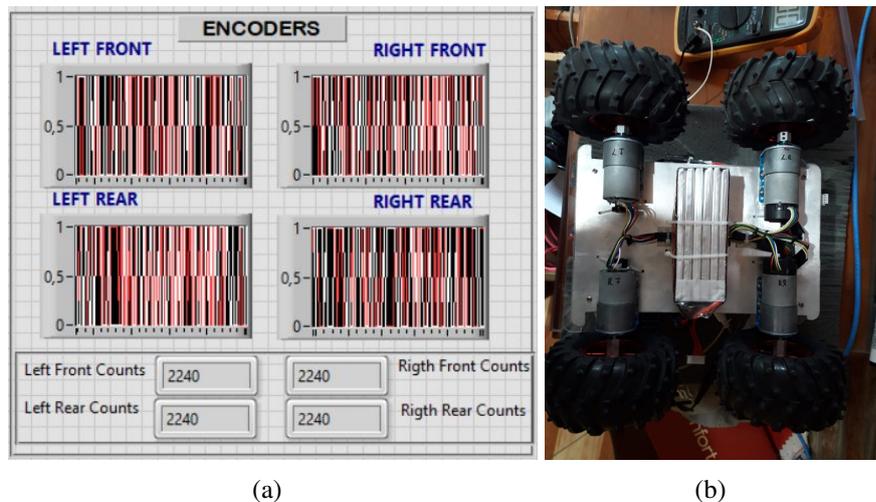


Figura 6.7: Prueba de lectura de codificadores, a) programa implementado en LabView, b) robot vista inferior

Tabla 6.1: Comparación revoluciones y pulsos

Número de vueltas	Pulsos default	Pulsos Decodificador LF	Pulsos Decodificador RF	Pulsos Decodificador LR	Pulsos Decodificador RR
1 Rev Sentido horario	2240	2240	2240	2240	2240
10 Rev Sentido horario	22400	22400	22400	22400	22400
1 Rev Sentido anti-horario	-2240	-2240	-2240	-2240	-2240
10 Rev Sentido anti-horario	-22400	-22400	-22400	-22400	-22400

Como se puede observar los pulsos que están de referencia (Pulsos default) y los pulsos leídos por los codificadores, se deduce que no existe un error en la lectura de este sensor.

6.2.2. Velocidad de los motores

Esta prueba consta de hacer girar los motores acoplados a las ruedas para comprobar el funcionamiento del algoritmo que mide su velocidad implementado en la FPGA.

6.2.2.1. Controlador en lazo abierto

Se realiza la medición de velocidad del controlador en lazo abierto, esta prueba consiste en ajustar varias velocidades por medio de variación del duty cycle en el panel frontal de la interfaz gráfica y comparar las lecturas con las de un tacómetro digital.

En la Fig. 6.8 a), se observa la toma de datos a través del panel frontal además del tacómetro digital proporcionando las velocidades en RPM de los motores girando a diferentes duty cycle.

Los resultados obtenidos se presentan en la Tabla 6.2, donde se indica la velocidad de cada una de las ruedas y el valor leído por el tacómetro digital.

Tabla 6.2: Velocidades de las ruedas sin control

Ciclo de trabajo	RPM Tacómetro	RPM LF	RPM RF	RPM LR	RPM RR
10 %	23	20	19	21	18
40 %	129	130	128	127	125
60 %	147	148	146	147	145
80 %	160	162	157	159	156
100 %	172	169	165	166	160

Se observan las velocidades leídas tanto por el tacómetro y por el programa implementado en LabView, se deduce que existe mas del 10% de error global entre el tacómetro y las velocidades que lee los codificadores.

6.2.2.2. Controlador en lazo cerrado

Se realiza la medición de velocidad de cada una de las ruedas con el controlador en lazo cerrado, al igual que en la sección anterior esta prueba consiste en ajustar varias velocidades por medio de variación del duty cycle en el panel frontal y comparar con las lecturas del tacómetro digital.

En la Fig. 6.8 b), se observa la toma de datos a través del panel frontal además del tacómetro digital proporcionando las velocidades en RPM de los motores girando a diferentes duty cycle.

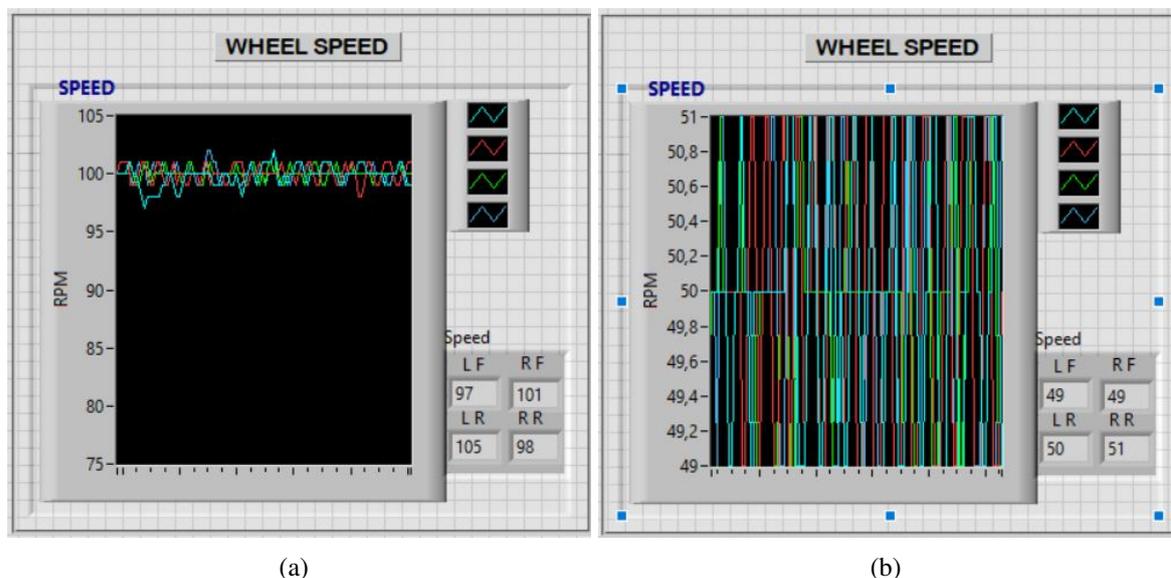


Figura 6.8: Lectura de velocidad, a) lazo abierto, b) lazo cerrado

Los resultados obtenidos se presentan en la Tabla 6.3, donde se indica la velocidad de cada una de las ruedas y el valor leído por el tacómetro digital.

Tabla 6.3: Velocidades de las ruedas con el control implementado

Ciclo de trabajo	RPM Tacómetro	RPM LF	RPM RF	RPM LR	RPM RR
10 %	23	22	23	22	23
40 %	129	130	128	129	130
60 %	147	147	147	147	146
80 %	160	160	161	160	160
100 %	172	172	171	170	173

Como se puede observar en la anterior tabla las velocidades leídas tanto del tacómetro y del programa en LabView, se ve que el error es menor al 3%, llegando a la conclusión que el controlador esta haciendo su trabajo.

6.2.3. Voltaje de baterías

En la Fig. 6.9, se realiza la medición de voltaje de baterías con un voltímetro digital y se compara con los datos obtenidos del programa implementado en LabView. Los cuales se detallan en la Tabla 6.4.

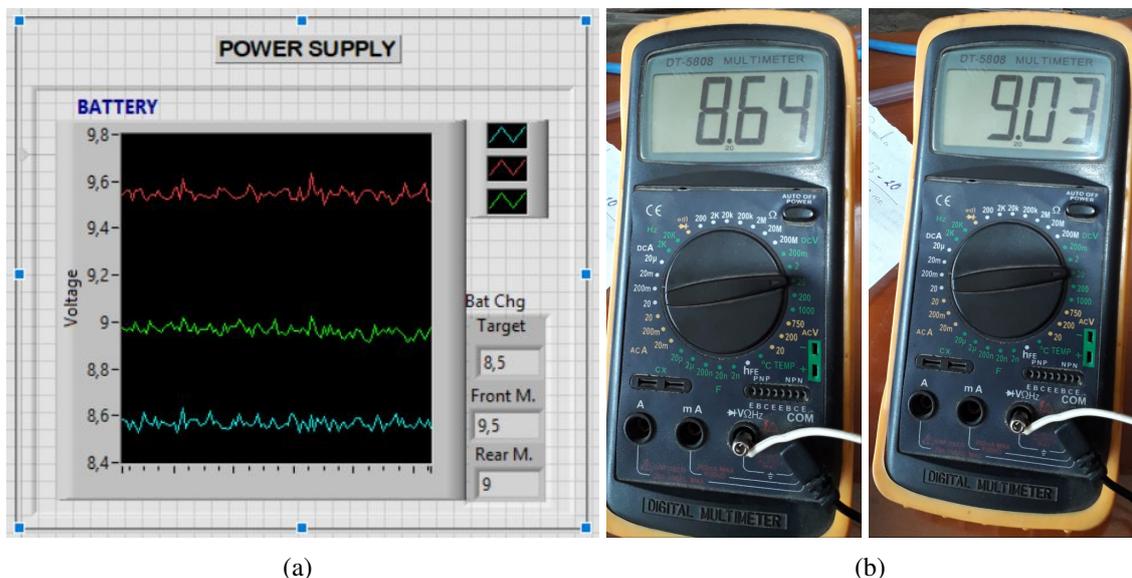


Figura 6.9: Lectura de voltaje en las baterías, a) programa implementado LabView, b) voltímetro digital

Tabla 6.4: Voltaje de baterías

Batería	Voltaje voltímetro	Voltaje HMI
Fuente de alimentación	8,5	8,5
Motores delanteros	9,5	9,5
Motores posteriores	9	9

La Tabla 6.4 muestra los valores leídos por el voltímetro y el programa y la Tabla 6.5 muestra una comparación entre las distancias medidas.

6.2.4. Distancia

Se realiza una comparación entre las medidas obtenidas por el sensor PING y medidas reales, tal y como se puede apreciar en la Fig. 6.10, a esto se realiza una tabla comparativa, Tabla 6.5.

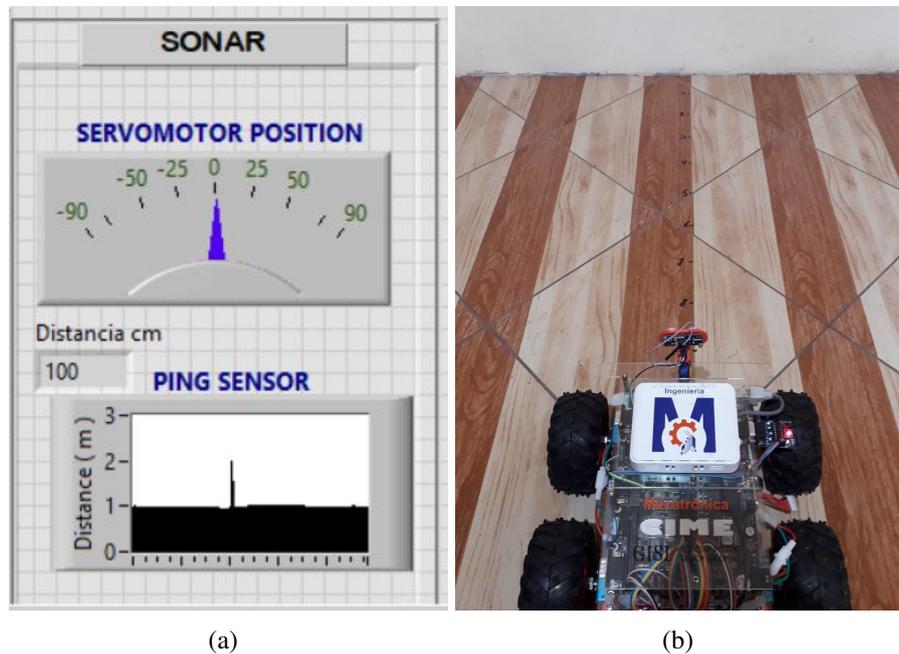


Figura 6.10: Medición de distancias, a) programa LabView, b) sensor ping

Se escribe la distancia sobre la superficie de desplazamiento del robot, para su posterior comparación con las medidas del programa.

Tabla 6.5: Comparación de distancias

Distancia real cm	Distancia medida cm
20	20
50	50
80	81
100	99
150	148

La Tabla 6.5 muestra la comparación de distancias leídas, dando como resultado que el sensor a distancias cortas funciona sin error, pero a medidas de $1,50m$ o más tiene un error de $3cm$.

6.3. Pruebas del controlador de velocidad

Para comprobar el correcto funcionamiento del controlador de velocidad se realizan pruebas sobre el robot móvil en lazo abierto y lazo cerrado. Las pruebas de funcionamiento son

- Control de velocidad en vacío (sin carga).
- Controlador sobre superficies lisas.
- Controlador sobre superficies rugosas.

6.3.1. Controlador en vacío

En esta sección se comprueba el funcionamiento del controlador en lazo cerrado. Los motores solo tienen la carga de las ruedas en esta prueba. Se coloca una referencia a cada uno de los motores, LF, LR, RF, RR (Cap.5), para identificar a que lectura pertenece en la RT de LabView, ver Fig. 6.11.

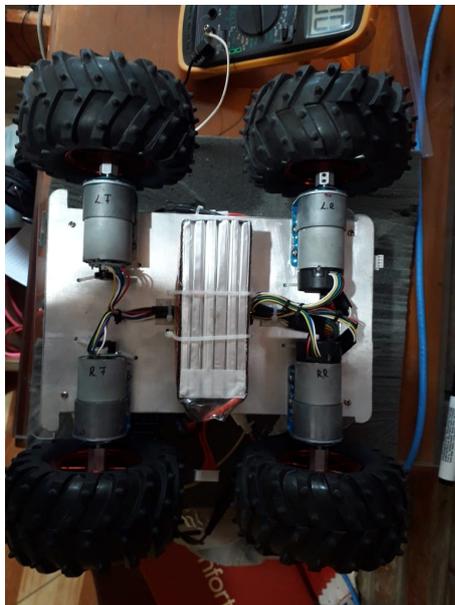


Figura 6.11: Robot móvil vista inferior

Se observan que las velocidades de los motores tienen un mínimo error y el controlador trata de igualar al valor de referencia 50RPM, ver la Fig. 6.12.



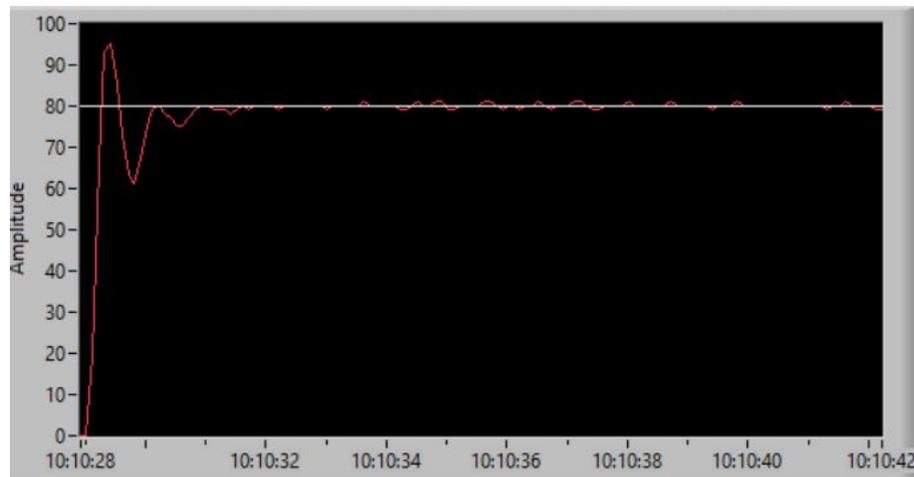
Figura 6.12: Prueba en vacío de motores

Se comprueba que el robot mantenga las velocidades de referencia aunque se aplique una perturbación exterior como en la Fig. 6.13.



Figura 6.13: Prueba en vacío de motores con perturbación exterior

En la Fig. 6.14 se observa la respuesta del controlador ante una perturbación exterior, esta prueba se realiza a cada uno de los motores para interpretar mejor los datos.



(a)



(b)

Figura 6.14: Prueba en vacío de motores, a) valor de referencia, tiempo de estabilización, sobre impulso, b) valor de referencia y respuesta a perturbaciones

6.3.2. Controlador en superficies planas

Con la finalidad de analizar el correcto funcionamiento del controlador de velocidad, se realiza una comparación de funcionamiento del robot con el controlador en lazo abierto y con-

trolador en lazo cerrado. En esta sección se realiza la prueba en una superficie plana libre de obstáculos.

La prueba consiste en hacer mover el robot una distancia en línea recta y hacerle seguir una trayectoria circular, con la finalidad de observar el comportamiento del robot y hacer una comparación entre las pruebas.

6.3.2.1. Trayectoria rectilínea control en lazo abierto

El robot parte de una posición inicial $p_o = (0,0)$. La Fig. 6.15, muestra los resultados de los datos obtenidos de los sensores que durante la prueba fueron leídos desde el robot móvil.

En esta prueba el duty cycle aplicado a cada uno de los motores es el mismo, las velocidades de los motores no son las mismas dando como resultado no una trayectoria rectilínea si no una trayectoria tipo parabólica como se observa en la Fig. 6.15.

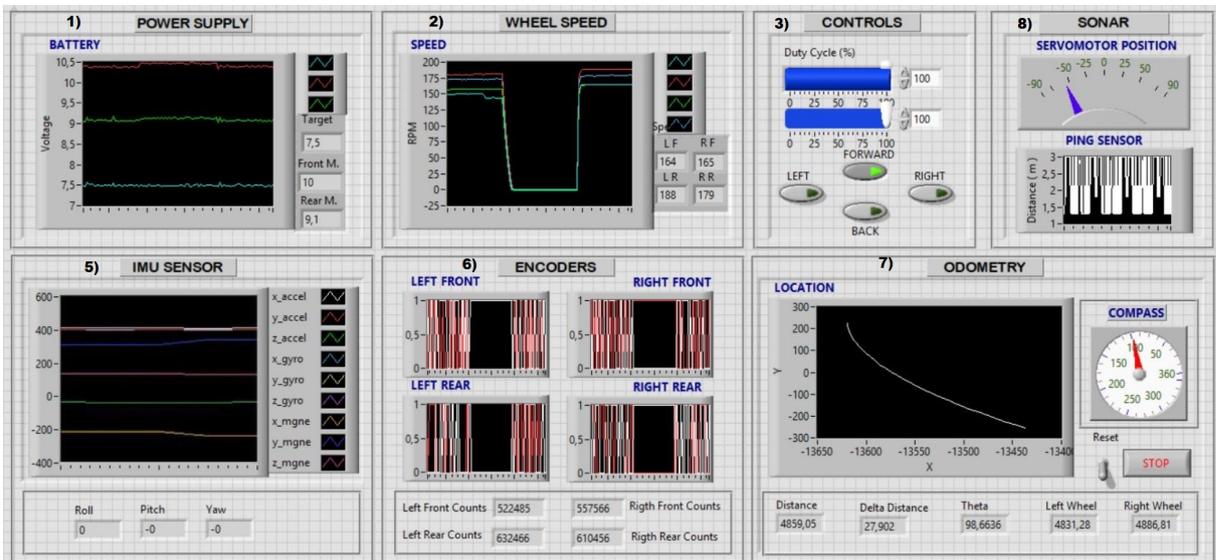


Figura 6.15: Interfaz gráfica con resultados del control en lazo abierto. Pág 49

La distancia total recorrida del robot móvil es calcula por el programa integrado en el robot, se observa que la distancia es 408.83 cm con un ángulo de desviación de 24° desde la posición

inicial y en la Fig. 6.16 se puede verificar la distancia recorrida por el robot mediante mediciones escritas en la superficie de trayectoria.

En la Fig. 6.16 se observa la trayectoria del robot donde se supone que debería seguir en línea recta y no haber error en el eje Y y en la orientación, esto sucede ya que no existe un control en lazo cerrado o un análisis dinámico que compense las perturbaciones exteriores.

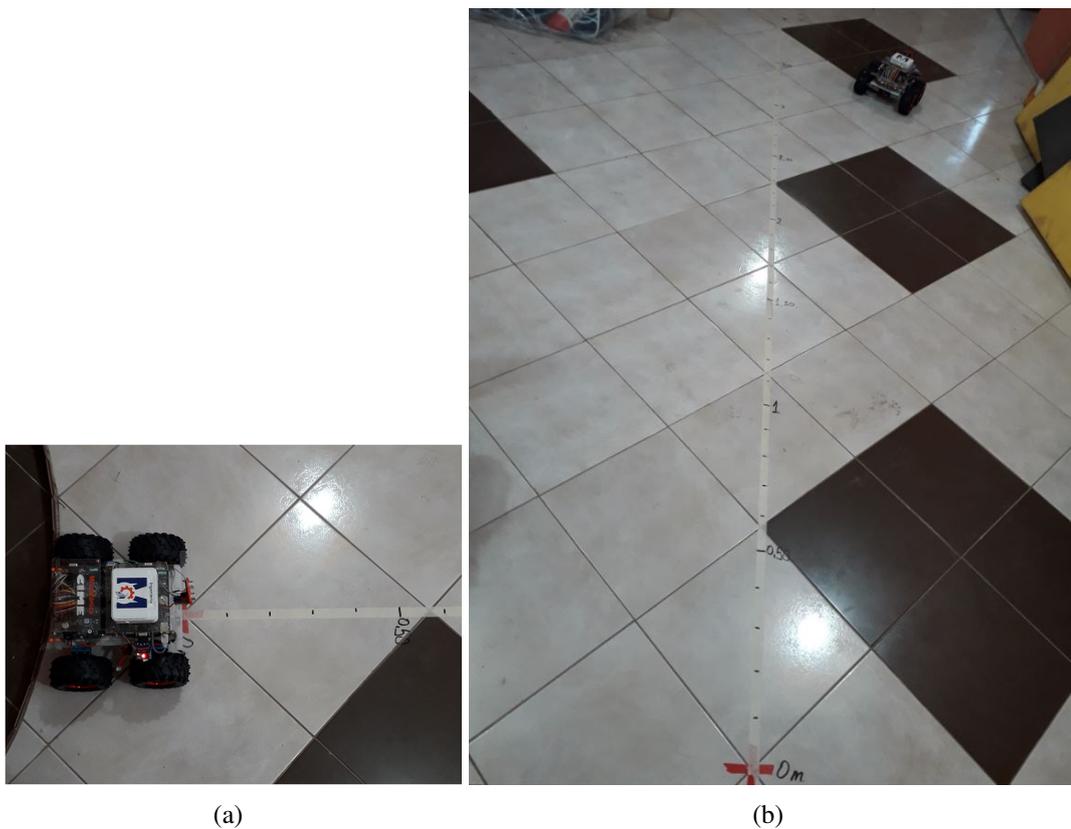


Figura 6.16: Prueba real trayectoria rectilínea superficie lisa a) punto de partida, b) punto final

6.3.2.2. Trayectoria rectilínea control en lazo cerrado

Se realiza la prueba de forma similar al apartado anterior, en esta sección el controlador ya esta implementado en el robot móvil.

El robot parte de una posición inicial $p_o = (0,0)$. La Fig. 6.17, muestra los resultados de los datos obtenidos de los sensores que durante la prueba fueron leídos desde el robot móvil.

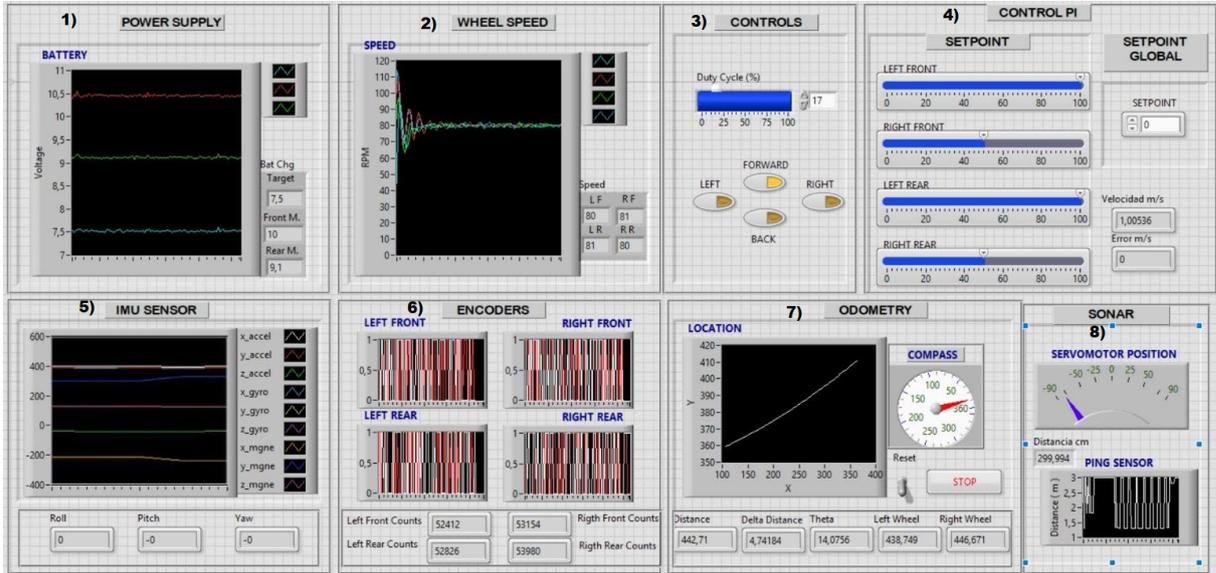


Figura 6.17: Interfaz gráfica con resultados del controlador en lazo cerrado. Pág 49

En esta prueba se debe asignar una velocidad de referencia a la que se desee que se estilice el robot, esta velocidad se aplica a cada uno de los motores, las velocidades de los motores tiene un error mínimo, dando como resultado una trayectoria casi rectilínea como se observa en la Fig. 6.18.

La distancia total recorrida del robot móvil es calcula por el programa integrado en el robot, se observa que la distancia es 442.71 cm con un ángulo de desviación de 5° desde la posición inicial y en la Fig. 6.17 se puede verificar la distancia recorrida por el robot mediante mediciones escritas en la superficie de trayectoria.

En la Fig. 6.18 se observa la trayectoria del robot donde tiende a ser una trayectoria rectilínea reduciendo considerablemente el error.



Figura 6.18: Prueba real trayectoria rectilínea superficie lisa, a) punto de partida, b) punto final

6.3.2.3. Trayectoria circular control en lazo abierto

En esta prueba se trata de hacer mover al robot desde una posición inicial $P_o = (0,0)$ siguiendo una trayectoria circular volver al mismo punto de partida. En la Fig. 6.19 se puede observar los resultados de los datos obtenidos de los sensores que durante la prueba fueron leídos desde el robot móvil.

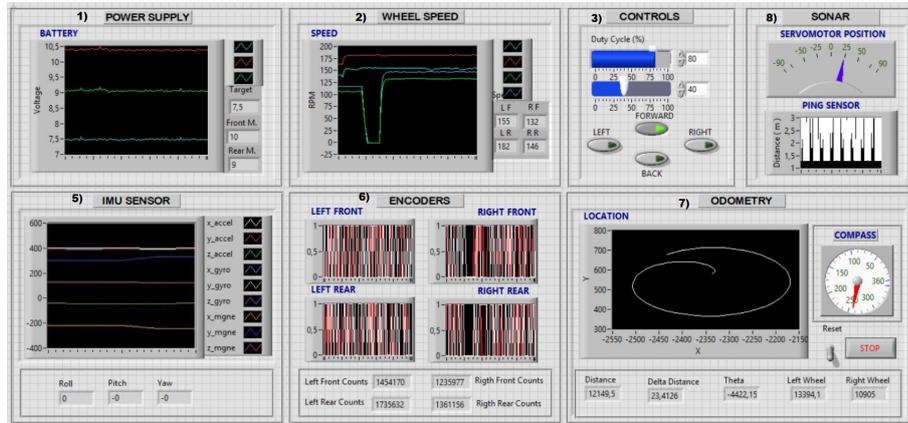


Figura 6.19: Interfaz gráfica con resultados del controlador en lazo abierto. Pág 49

Para esta prueba el duty cycle es diferente para los motores teniendo los motores de la izquierda la mitad del duty cycle que los de la derecha, ya que para poder realizar la trayectoria circular las velocidades deben ser diferentes, al igual que en la trayectoria rectilínea aquí existe error en las velocidades del robot dando como resultado no una trayectoria circular si no una trayectoria tipo elipse sin llegar a cerrar en el mismo punto de partida, tal y como se observa en la Fig. 6.19.



Figura 6.20: Prueba real trayectoria circular superficie lisa

En la Fig. 6.20 se observa la trayectoria del robot donde se supone que debería seguir en

línea circular, esto sucede ya que no existe un control en lazo cerrado o un análisis dinámico que compense las perturbaciones exteriores.

6.3.2.4. Trayectoria circular control lazo cerrado

Se realiza la prueba de forma similar al apartado anterior, en esta sección el controlador ya está implementado en el robot móvil.

En esta prueba se trata de hacer mover al robot desde una posición inicial $P_o = (0,0)$ siguiendo una trayectoria circular volver al mismo punto de partida. En la Fig. 6.21 se puede observar los resultados de los datos obtenidos de los sensores que durante la prueba fueron leídos desde el robot móvil.

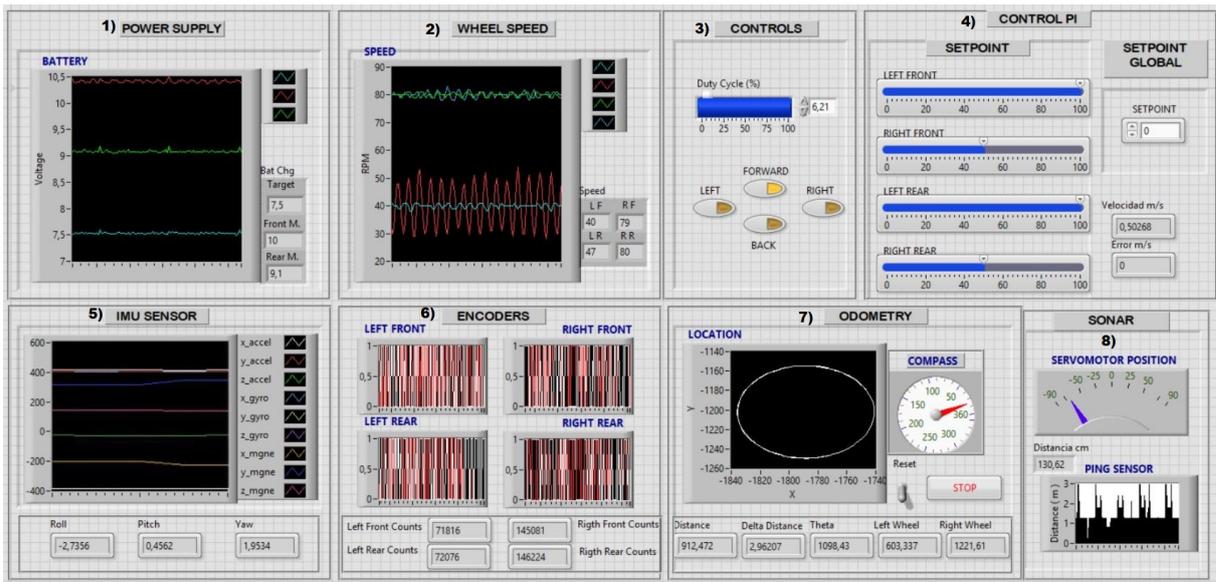


Figura 6.21: Interfaz gráfica con resultados del controlador en lazo cerrado. Pág. 49

En esta prueba se debe asignar una velocidad de referencia a los motores de la derecha y la mitad de la velocidad de referencia a los motores de la izquierda para poder realizar la trayectoria circular, las velocidades de los motores tienen un error mínimo, dando como resultado una trayectoria casi perfecta como se observa en la Fig. 6.21.



Figura 6.22: Prueba real trayectoria circular superficie lisa

En la Fig. 6.22 se observa la trayectoria circular que sigue el robot donde el error disminuye.

6.3.3. Controlador en superficies rugosas

En esta sección se realizan las pruebas en una superficie rugosa libre de obstáculos.

La prueba consiste en hacer mover el robot una distancia en línea recta y hacerle seguir una trayectoria circular, con la finalidad de observar el comportamiento del robot y hacer una comparativa entre las pruebas.

6.3.3.1. Trayectoria circular control en lazo abierto

Siguiendo las mismas condiciones de funcionamiento de la sección anterior se pasa al robot a una superficie rugosa, dando como resultados la Fig. 6.23.

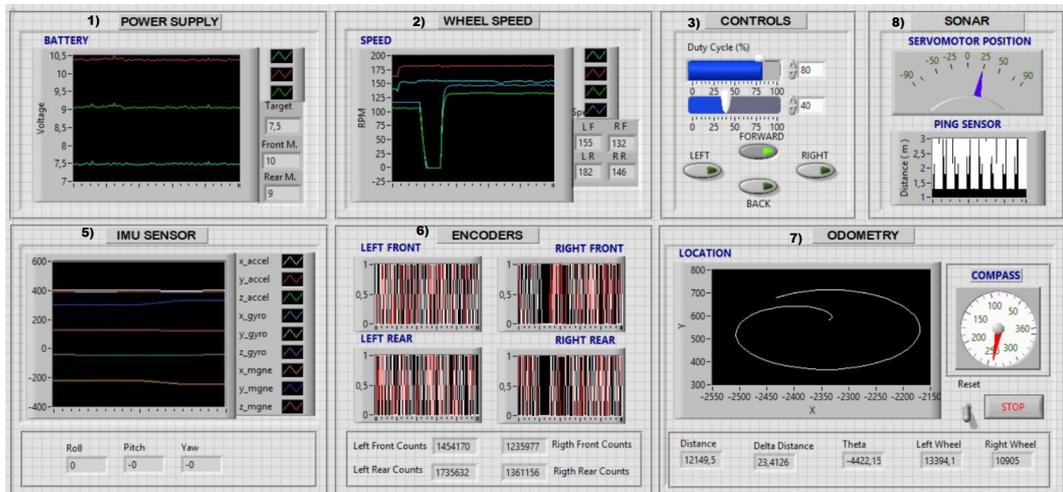


Figura 6.23: Interfaz gráfica con resultados del controlador en lazo abierto. Pág. 49

Al igual que en la prueba anterior existe una variación de velocidades debido a los factores mencionados anteriormente pero en este caso por ser la superficie rugosa existe más agarre entre las ruedas y el piso, durante la trayectoria del robot el ángulo de desviación es menor en comparación a la prueba en superficies lisas.

En la Fig. 6.24 se observa la distancia recorrida del robot mediante mediciones escritas en la superficie de trayectoria y se pueden comparar con las mediciones leídas por los sensores del robot en la Fig. 6.24.

Aunque la superficie rugosa permite mas agarre de las ruedas con el piso el error sigue siendo elevado por tal motivo es indispensable implementar un controlador para corregir estas desviaciones.

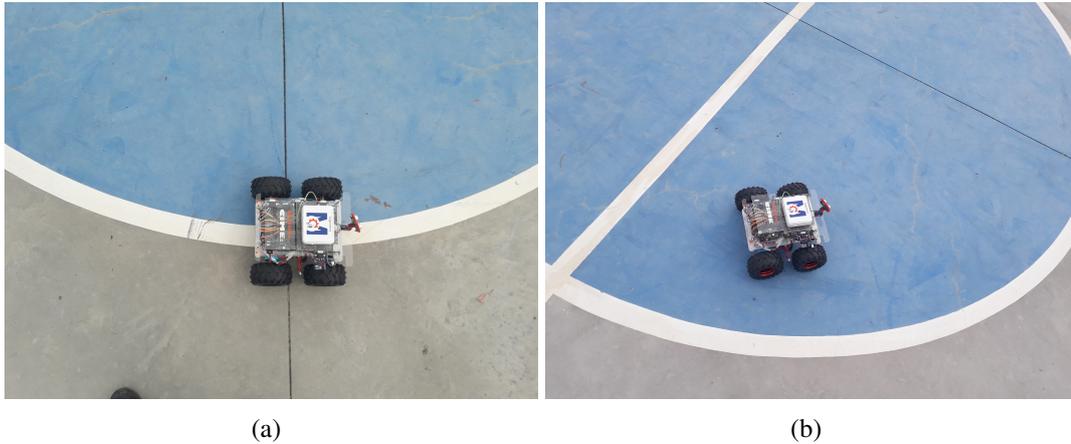


Figura 6.24: Prueba de trayectoria circular control lazo abierto, a) punto inicial de la prueba, b) recorrido del robot

6.3.3.2. Trayectoria circular control en lazo cerrado

En esta prueba se trata de hacer mover al robot desde una posición inicial $P_o = (0,0)$ siguiendo una trayectoria circular volver al mismo punto de partida. En la Fig. 6.25 se puede observar los resultados de los datos obtenidos de los sensores que durante la prueba fueron leídos desde el robot móvil.

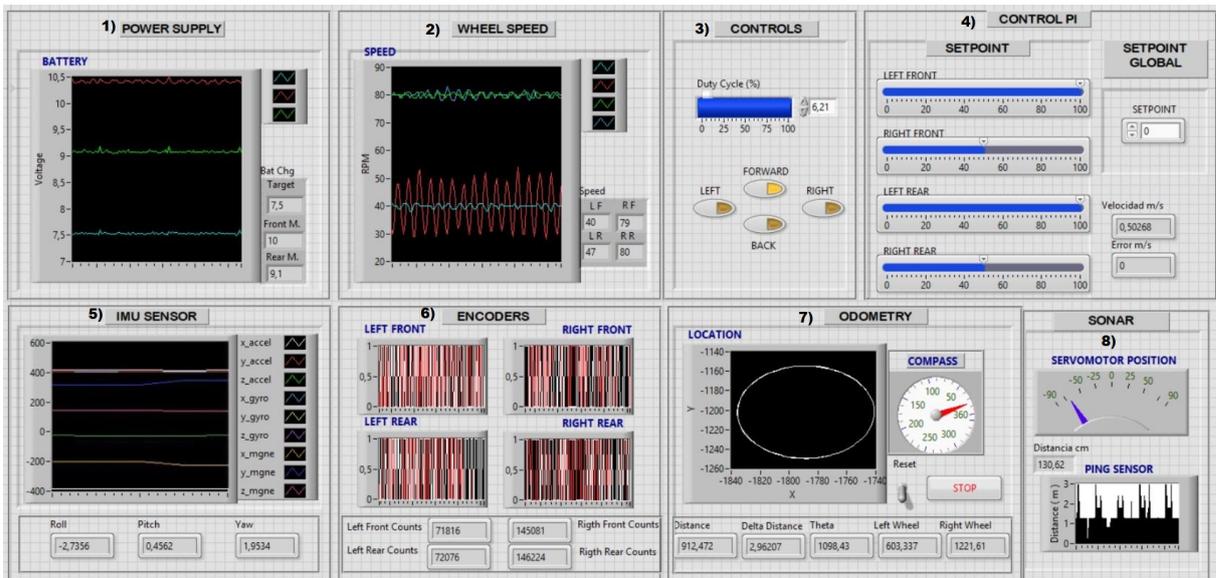


Figura 6.25: Interfaz gráfica con resultados control lazo cerrado. Pág 49

En esta prueba se debe asignar una velocidad de referencia a los motores de la derecha y la mitad de la velocidad de referencia a los motores de la izquierda para poder realizar la trayectoria circular, las velocidades de los motores tienen un error mínimo, dando como resultado una trayectoria casi perfecta como se observa en la Fig. 6.26.

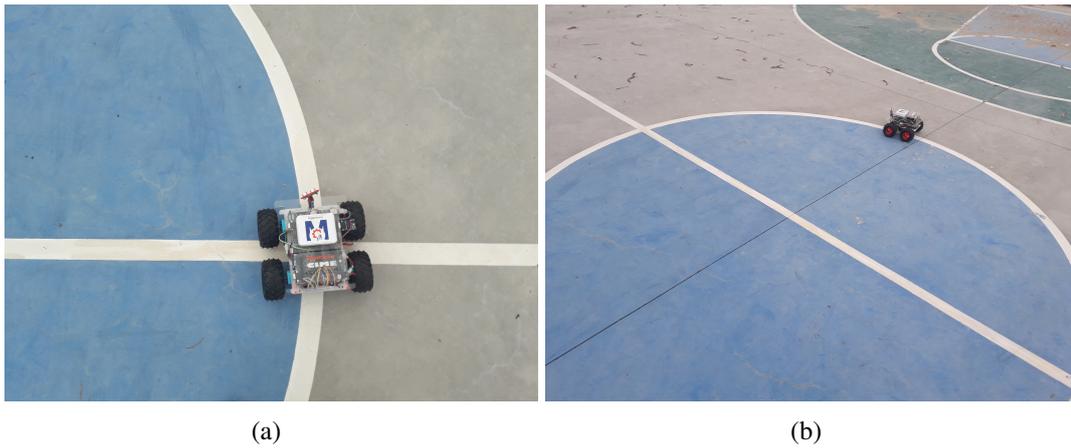


Figura 6.26: Prueba de trayectoria circular lazo cerrado, a) punto inicial, b) recorrido del robot móvil

Capítulo 7

Conclusiones y trabajo futuro

7.1. Conclusiones

- Utilizando técnicas convencionales se logró determinar un modelo matemático que describe la dinámica del robot, comprobando su efectividad y dando como resultado un error mínimo a comparación del funcionamiento real del robot.
- Se modeló una función de transferencia que describe el funcionamiento de cada uno de los motores, los cuales se consideraron solo con las cargas de las ruedas, además se validó con otros métodos.
- La interfaz gráfica desarrollada en el software LabView permite una mejor adquisición de datos en tiempo real directamente de la tarjeta FPGA y una mejor interacción entre humano-robot, facilitando la interpretación mediante el panel frontal del programa.
- El modelo final con el que se trabajó es un sistema de primer orden, esto ayudó a simplificar los cálculos para el controlador, se discretizó el modelo matemático ya que la implementación debe ser de un controlador digital y debe estar en el dominio de z .
- Para el diseño del controlador de velocidad, se analizó distintas técnicas de control, se realizó una comparación entre los métodos de diseño, se optó por la de Ziegler Nichols - MatLab ya que cumplía con las especificaciones de diseño.
- Como se pudo comprobar en el anterior capítulo el controlador cumple con su función, de esta manera se prueba la efectividad para la realización de un sistema de control de velocidad para un robot móvil con dirección diferencial.

7.2. Trabajo futuro

A partir de este trabajo se plantean diferentes trabajos para desarrollar en futuras investigaciones:

- Se plantea desarrollar estrategias de navegación con algoritmos capaces de dotar al robot de autonomía en diferentes entornos de trabajo.

Bibliografía

- [1] R. Ramírez Rivera and R. Reyez Pérez, “Diseño e implementación de un robot autónomo móvil usando tecnología fpga”, 1st ed. Guayaquil, 2015, p. 174.
- [2] G. Scaglia, “Estrategias para seguimiento de trayectorias”, 1st ed. San Juan Argentina, 2014, p. 58.
- [3] A. Calderon, J. Rosero and N. Sotomayor, “Plataforma autónoma móvil para seguimiento de trayectorias y evasión de obstáculos”, VISUS, p. 12, 2017.
- [4] G. Hernández, H. Ríos and M. Bueno López, “Implementation of a position and movement controller for a differential mobile robot”, SCIELO, p. 18, 2016. Available: <http://dx.doi.org/10.14483/udistrital.jour.tecnura.2016.2.a09>.
- [5] X. Wu, M. Xu and L. Wang, “Differential Speed Steering Control for Four-Wheel Independent Driving Electric Vehicle”, International Journal of Materials, Mechanics and Manufacturing.
- [6] A. Vaca, “Robot móvil para aplicaciones de planificación de movimiento: sistema de odometría”. Ibarra, 2017.
- [7] Mohd Saifizi Saidonr, H. Desa and M. Rudzuan, “A differential steering control with proportional controller for an autonomous mobile robot”, in *IEEE 7th International Colloquium on Signal Processing and its Applications*, 2011.
- [8] B. Siciliano, “Springer handbook of robotics”. Berlin: Springer, 2008.
- [9] Y. Yi, F. Mengyin, S. Changsheng, W. Meiling and Z. Cheng, “Control Law Design of Mobile Robot Trajectory Tracking and Development of Simulation Platform”, in *Chinese Control Conference*, 2006. Available: 10.1109/chicc.2006.4347355.

- [10] R. Silva Ortigoza, J. García Sánchez, V. Barrientos Sotelo, M. Molina Vilchis, V. Hernández Guzmán and G. Silva Ortigoza, "Una panorámica de los robots móviles", DIALNET, Revista Electrónica de Estudios Telemáticos, vol. 6, no. 3, p. 14, 2007.
- [11] B. Sotelo, V. Ricardo, G. Sánchez, J. Rafael, S. Ortigoza, "Robots Móviles: Evolución. Polibits", 2007, Available: <<http://www.redalyc.org/articulo.oa?id=402640448003>>ISSN 1870-9044.
- [12] I. 8373:2012, "ISO 8373:2012", ISO, 2019. [Online]. Available: <https://www.iso.org/standard/55890.html>.
- [13] N.Instrument "LabVIEW Robotics Starter Kit (Robotics Module) - LabVIEW 2014 Robotics Module Help - National Instruments", Zone.ni.com, 2019. [Online]. Available : http://zone.ni.com/reference/en-XX/help/372983F-01/lvrobogsm/robo_skit_o_verview/.
- [14] A. Baño Azcon, "Análisis y diseño del control de posición de un robot móvil con tracción diferencial", 2003.
- [15] R. Hernández Alemán, "Control descentralizado de formación y movimiento coordinado de robots móviles tipo unicycle". Madrid España, 2013, p. 132.
- [16] G. Andaluz, V. Andaluz and A. Rosales, "Modelación, identificación y control de robots móviles", Universidad Técnica de Ambato, p. 9, 2017.
- [17] M. Babar and M. Baglietto, "Optimal feedback control for the identification of two-wheeled mobile robot", in *21st International Conference on System Theory, Control and Computing (ICSTCC)*, 2017.
- [18] L. Gracia Calandín, "Modelado cinemático y control de robots móviles con ruedas", Doctorado, Universidad Politécnica de Valencia, 2015.
- [19] C. Smith and D. Starks, "A dynamic model of a water treatment unit", in *1977 IEEE Conference on Decision and Control including the 16th Symposium on Adaptive Proces-*

- ses and A Special Symposium on Fuzzy Set Theory and Applications*, 1977. Available: 10.1109/cdc.1977.271625.
- [20] K. Lutvica, N. Osmic and Z. Juric, "Identification of an Ethane-ethylene distillation column system ; A procedure for MIMO system identification using the IDENT toolbox", in *21st Telecommunications Forum Telfor (TELFOR)*, 2013. Available: 10.1109/telfor.2013.6716281.
- [21] N. Ramli and M. Chandra Mohan, "Linear Model Predictive Control of a Debutanizer column", in *4th International Conference on Software Engineering and Computer Systems (ICSECS)*, 2015. Available: 10.1109/icsecs.2015.7333090.
- [22] Xin He, B. Gallas and E. Frey, "Three-Class ROC Analysis—Toward a General Decision Theoretic Solution", in *IEEE Transactions on Medical Imaging*, vol. 29, no. 1, pp. 206-215, 2010. Available: 10.1109/tmi.2009.2034516.
- [23] Fierro, R. y Lewis, F. (1995). "Control of a nonholonomic mobile robot: Backstepping kinematics into dynamics", in *Proc. 34th Conference on Decision and Control*, 3805 – 3810.
- [24] J.M. Guerrero, J.C. Vasquez, J. Matas, L.G. de Vicuña, and M. Castilla, "Hierarchical control of droop-controlled AC and DC microgrids: a general approach toward standardization", in *IEEE Transactions on Industrial Electronics*, vol. 58, no. 1, pp. 158-172, Jan. 2011

Apéndice

Este apéndice incluye el software desarrollado en el proyecto. Sólo se han anexado los archivos más importantes. Para obtener el código completo, consulte los medios adjuntos.

.A. Software

.A.1. Modelo cinemático

Programa 1: Script para simular el modelo cinemático

```

clc; clear all; close all; %elimina datos y variables
tfin=70; %tiempo de simulacion%
To=0.1; %tiempo de muestreo
t= [0:To:tfin]; %tiempo
a=0.2;

%l)Condiciones iniciales
x(1) = 1; %posicion inicial en el eje x
y(1) = 0.5; %posicion inicial en el eje y
psi(1) = 0; %Orientacion inicial respecto a z

%referencia deseada
Pxd=2*cos(0.1*t); % Posicion deseada en el eje X
Pyd=2*sin(0.1*t); % Posicion deseada en el eje Y
Pxd_p=-2*0.1*sin(0.1*t); % Velocidad deseada en el eje X
Pyd_p=2*0.1*cos(0.1*t); % Velocidad deseada en el eje y

% Bucle de control
for k=1: length(t)
    xerr(k) = Pxd(k) - x(k);
    yerr(k) = Pyd(k) - y(k);
    err = [xerr(k); yerr(k)];
    J = [cos(psi(k)) -a*sin(psi(k));...
         sin(psi(k)) a*cos(psi(k))];
    K = [1 0;...
         0 1];
    vref = inv(J)*([Pxd_p(k); Pyd_p(k)]+K*(err));
    u_ref(k) = vref(1); %velocidad lineal de control[m/s]
    w_ref(k) = vref(2); %velocidad angular[rad/seg]

```

```

%robot movil
%l1 Modelo cinematico
xp(k) =u_ref(k)*cos(psi(k))-a*w_ref(k)*sin(psi(k));
yp(k) =u_ref(k)*sin(psi(k))+a*w_ref(k)*cos(psi(k));
psi(k+1) = To*w_ref(k)+psi(k);
x(k+1) =To*xp(k)+x(k);
y(k+1) =To*yp(k)+y(k);
end
%%Animacion
scrsz=get(0,'ScreenSize');
off1=0;
off2=100;
figpos=[off1 off2 scrsz(3)-off1 scrsz(4)-off2];

f1=figure(1);
set(f1,'Color','w','Position',figpos);

paso=5; axis 'equal'

Robot_Dimension(2);
Ho=Robot_Plot_3D(x(1),y(1),psi(1),'g'); hold on

H1 = plot(x(1),y(1),'*m'); hold on;
H2 = plot(x(1),y(1),'*m'); hold on;

H4 = plot(Pxd(1),Pyd(1),'b');
%Bucle de animacion
for i=1:paso:length(t)

    delete(Ho)
    delete(H1)
    delete(H2)
    delete(H4)
    axis([-3 5 -3 5 0 1]);
    view([-30.0,30.0]);
    Ho=Robot_Plot_3D(x(i),y(i),psi(i),'g');hold on
    H1 = plot(x(i),y(i),'*m');hold on
    H2 = plot(x(1:i),y(1:i),'g');hold on

    H3 = plot(Pxd,Pyd,'m');hold on
    H4 = plot(Pxd(i),Pyd(i),'*b');
    grid on;
    pause(To)

```

```

end
%Figuras de desplazamiento en X,Y y rotacion ,

figure
subplot(3,1,1)
    plot(t,x(1:length(t))); grid;
    title('Desplazamiento en X');
    xlabel('Tiempo[s]'); ylabel('X[m]');

    subplot(3,1,2)
        plot(t,y(1:length(t)),'g'); grid;
        title('Desplazamiento en Y');
        xlabel('Tiempo[s]'); ylabel('Y[m]');

    subplot(3,1,3)
        plot(t,psi(1:length(t)),'m'); grid;
        title('Rotacion');
        xlabel('Tiempo[s]'); ylabel('\psi')
figure

    plot(x,y); grid;
    title('Desplazamiento en XY');
    xlabel('X[m]'); ylabel('Y[m]');

```

.A.2. Función de transferencia

Programa 2: Script para simular la función de trasferencia

```

clc; clear all; close all; warning off;
load('Planta.mat')
load('Control_planta.mat')
load('controlz.mat')
load('Control_PI.mat')
TFs=zpk(Plant)
PlantZ=c2d(TFs,0.1,'zoh')

% Lazo Abierto
step(TFs)

```

```

hold on
step(PlantZ)
title('Funcion de Tranferencia dominio de S y Z en lazo abierto')

% Lazo cerrado
figure
FB=(feedback(PlantZ,1))
step(FB)
title('Funcion de Tranferencia en lazo cerrado')
figure
rlocus(PlantZ)
title('Lugar geometrico de las raices')

% polos
polos=pole(FB)
abs(polos)
figure
pzmap(FB)
title('plano de ceros y polos del sistema')

% funcion Lazo cerrado con controlador
figure
C=(control_z*PlantZ)/(1+control_z*PlantZ)
step(C)
title('Funcion de tranferencia en lazo cerrado controlador y planta')
figure
rlocus(C)
title('Lugar geometrico de las raices del controlador y planta')

% polos 2
polos=pole(C)
abs(polos)
figure
pzmap(C)
title('plano de ceros y polos del sistema')

% Matlab
zpk(PlantZ)
n=[0.65388]
d=[1 -0.7571]
k=0:25;
u=[0,0,0 ones(1,23)];
y=filter(n,d,u);

```

```
plot(k,y,'o',k,y,'-',k,u,'--')
figure
step(ans)

% Lugar geometrico de las raices
figure
rlocus(PlantZ)
xlabel ('eje Real')
ylabel ('eje Imaginario')
title ('Lugar geometrico de las raices (LGR)')

% polos y ceros del sistema
FB=feedback(PlantZ,1)
figure
pzmap(FB)
title ('Plano de polos y ceros del sistema')

% Respuesta a escalon lazo cerrado
figure
step (FB)
title ('Respuesta a escalon unitario en lazo cerrado')
```

