

CARRERA DE ELECTRÓNICA

IMPLEMENTACIÓN DE UN ROBOT MÓVIL DIFERENCIAL PARA EL SEGUIMIENTO DE TRAYECTORIAS MEDIANTE UN CONTROLADOR PREDICTIVO BASADO EN MODELOS.

**Trabajo de Integración Curricular previo a la obtención del Título de
Tecnólogo Superior en Electrónica**

Autor: ALEXIS IVAN HERNANDEZ RENGIFO

Tutor: MG. ROSERO YUGSI RICARDO ELICIO

**MARZO, 2025
SANGOLQUI – ECUADOR**

Autor: Hernandez Rengifo Alexis Ivan	Título a obtener: Tecnólogo Superior en Electrónica Matriz: Sangolquí -Ecuador Correo electrónico: alexis.hernandez@ister.edu.com
	Dirigido por: Mg. Rosero Yugsi Ricardo Elicio 
	Matriz: Sangolquí -Ecuador Correo electrónico: ricardo.rosero@ister.edu.ec

TODOS LOS DERECHOS RESERVADOS.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra para fines comerciales, sin contar con autorización de los titulares de propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Se permite la libre difusión de este texto con fines académicos investigativos por cualquier medio, con la debida notificación a los autores.

©2025 Tecnológico Universitario Rumiñahui

SANGOLQUÍ – ECUADOR

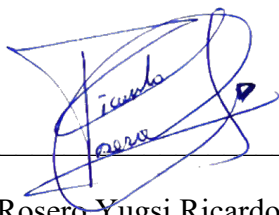
HERNANDEZ RENGIFO ALEXIS IVAN

IMPLEMENTACIÓN DE UN ROBOT MÓVIL DIFERENCIAL PARA EL SEGUIMIENTO DE TRAYECTORIAS MEDIANTE UN CONTROLADOR PREDICTIVO BASADO EN MODELOS.

APROBACIÓN DEL TUTOR/A

En mi calidad de Tutor/a del Trabajo de Titulación, presentado por **Hernandez Rengifo Alexis Ivan** con CC: 1720130739 para optar por el Título de Tecnólogo/a Superior en Electrónica; cuyo título es: Tecnólogo Superior en Electrónica, se considera que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del tribunal examinador que fuere designado.

Sangolquí, a los 24 días del mes de marzo de 2025



Mg. Rosero Yugsi Ricardo Elicio
DOCENTE-TUTOR/A
C.I. 1716158611

CARTA DE CESIÓN DE DERECHOS DEL TRABAJO DE TITULACIÓN

CT-ANX-2025-ISTER-2-2.2

Sangolquí, 24 de marzo del 2025

MSc. Elizabeth Ordoñez
DIRECTORA DE DOCENCIA

MSc. Mónica Loachamín
COORDINADORA DE TITULACIÓN

**INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE
UNIVERSITARIO**

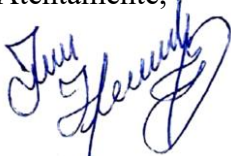
Presente

Por medio de la presente, yo, Alexis Ivan Hernandez Rengifo declaro y acepto en forma expresa lo siguiente: Ser autor del trabajo de titulación denominado **“IMPLEMENTACIÓN DE UN ROBOT MÓVIL DIFERENCIAL PARA EL SEGUIMIENTO DE TRAYECTORIAS MEDIANTE UN CONTROLADOR PREDICTIVO BASADO EN MODELOS”**, de la Tecnología Superior en Electrónica; y a su vez manifiesto mi voluntad de ceder al Instituto Superior Tecnológico Rumiñahui con condición de Universitario los derechos de reproducción, distribución y publicación de dicho trabajo de titulación, en cualquier formato y medio, con fines académicos y de investigación.

Esta cesión se otorga de manera no exclusiva y por un periodo indeterminado. Sin embargo, conservo los derechos morales sobre mi obra.

En fe de lo cual, firmo la presente.

Atentamente,



Alexis Ivan Hernandez Rengifo

C.I.: 1720130739

FORMULARIO PARA ENTREGA DE PROYECTOS EN BIBLIOTECA INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO

CT-ANX-2025-ISTER-3

CARRERA:

ELECTRÓNICA

AUTOR:

ALEXIS IVAN HERNANDEZ RENGIFO

TUTOR:

MG. RICARDO ELICIO ROSERO YUGSI

CONTACTO ESTUDIANTE:

0961023213

CORREO ELECTRÓNICO:

ivan.hernandezkg@gmail.com

TEMA:

IMPLEMENTACIÓN DE UN ROBOT MÓVIL DIFERENCIAL PARA EL SEGUIMIENTO DE TRAYECTORIAS MEDIANTE UN CONTROLADOR PREDICTIVO BASADO EN MODELOS.

OPCIÓN DE TITULACIÓN:

UNIDAD DE INTEGRACIÓN CURRICULAR

RESUMEN EN ESPAÑOL:

El presente proyecto de titulación se enmarca dentro de los avances tecnológicos impulsados por la Industria 4.0, donde la automatización, la robótica inteligente y el análisis de datos en tiempo real son elementos clave para optimizar los procesos industriales, logísticos y urbanos. Dentro de este contexto, el desarrollo de robots móviles autónomos representa una línea estratégica de innovación, especialmente en aplicaciones como el transporte automatizado, inspección de áreas industriales, logística interna, monitoreo de entornos complejos y la formación en robótica educativa.

El objetivo principal de este trabajo fue el diseño, desarrollo e implementación de un controlador predictivo basado en modelos (Model Predictive Control - MPC) aplicado a

un robot móvil diferencial ensamblado con el kit Lego Mindstorms EV3. A diferencia de los controladores clásicos, el MPC permite anticipar el comportamiento futuro del sistema, optimizando las acciones de control para seguir trayectorias definidas incluso en presencia de obstáculos o variaciones en el entorno.

El desarrollo se llevó a cabo utilizando el entorno de programación MATLAB y su herramienta Simulink, donde se modeló la dinámica del robot, se configuró el algoritmo MPC, y se realizaron simulaciones previas en entornos virtuales 3D con Simscape Multibody. Posteriormente, se implementó el código directamente sobre el hardware EV3, permitiendo validar el rendimiento en escenarios controlados y reales. Se diseñaron rutinas de evasión de obstáculos utilizando sensores ultrasónicos integrados, asegurando la navegación autónoma sin colisiones.

Los resultados experimentales confirmaron la eficacia del controlador MPC, logrando un seguimiento preciso de trayectorias con mejores niveles de estabilidad y adaptabilidad frente a métodos tradicionales como el control PID. El robot fue capaz de adaptarse dinámicamente ante la presencia de obstáculos, ajustando su trayectoria de forma eficiente.

Este trabajo no solo valida el uso de técnicas avanzadas de control predictivo en sistemas robóticos móviles, sino que también abre posibilidades de aplicación en entornos de manufactura flexible, transporte autónomo y ciudades inteligentes, alineándose con los desafíos y oportunidades que presenta la Industria 4.0.

PALABRAS CLAVE:

Matlab, Control, Industria 4.0, Trayectoria, Simulink, Obstaculos.

ABSTRACT:

This thesis project is framed within the technological advances driven by Industry 4.0, where automation, intelligent robotics, and real-time data analysis are key elements to optimize industrial, logistical, and urban processes. In this context, the development of autonomous mobile robots represents a strategic innovation path, especially for applications such as automated transport, industrial area inspection, internal logistics, complex environment monitoring, and educational robotics.

The main objective of this work was the design, development, and implementation of a Model Predictive Control (MPC) system applied to a differential mobile robot assembled using the Lego Mindstorms EV3 kit. Unlike classical control methods, MPC allows anticipating the system's future behavior by optimizing control actions to follow predefined trajectories even in the presence of obstacles or environmental variations.

The development was carried out using MATLAB and Simulink, where the robot dynamics were modeled, the MPC algorithm was configured, and preliminary simulations were conducted using 3D virtual environments with Simscape Multibody. Subsequently, the control code was deployed directly onto the EV3 hardware, allowing performance

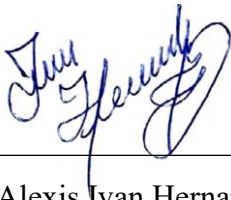
validation in both controlled and real environments. Obstacle avoidance routines were implemented using integrated ultrasonic sensors to ensure autonomous navigation without collisions.

The experimental results confirmed the effectiveness of the MPC controller, achieving accurate trajectory tracking with greater stability and adaptability compared to traditional methods such as PID control. The robot dynamically adjusted its trajectory when facing obstacles, optimizing its navigation path efficiently.

This work not only validates the application of advanced predictive control techniques in mobile robotic systems but also opens possibilities for their application in flexible manufacturing environments, autonomous transport, and smart cities, fully aligned with the challenges and opportunities posed by Industry 4.0.

KEYWORDS:

Matlab, control, industry 4.0, trajectory, simulink, obstacle.



Alexis Ivan Hernandez Rengifo

C.I.: 1720130739

SOLICITUD DE PUBLICACIÓN DEL TRABAJO DE TITULACIÓN

CT-ANX-2025-ISTER-4

Sangolquí, 24 de marzo del 2025

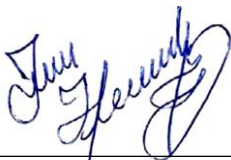
Sres.-

**INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE
UNIVERSITARIO**

Presente

Yo Alexis Ivan Hernandez Rengifo, con C.I.: 1720130739 alumno de la Carrera de ELECTRÓNICA cedo al INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO, los derechos de publicaciones del presente trabajo de Titulación en el Repositorio Institucional para hacer uso de todos los contenidos con fines estrictamente académico o de investigación.

Atentamente,



Alexis Ivan Hernandez Rengifo

C.I.: 1720130739

DEDICATORIA

A mi querida madre, cuya sabiduría, amor y apoyo incondicional han sido la luz que ha guiado mi camino en cada etapa de mi vida, tanto en lo personal como en lo profesional. Desde siempre, has sido mi mayor fuente de inspiración y fortaleza, brindándome los consejos más sabios y el ánimo necesario para superar cada desafío. Este proyecto de titulación es un reflejo de todo lo que he logrado, y es a ti a quien dedico con el más profundo agradecimiento y amor. Gracias por creer en mí incluso en los momentos en que dudé de mí mismo. Este logro es tanto tuyo como mío.

Alexis Ivan Hernandez Rengifo

AGRADECIMIENTO

Quiero expresar mi más sincero agradecimiento a los docentes del instituto, quienes han desempeñado un papel crucial en mi formación académica y profesional. Su dedicación, esfuerzo y pasión por la enseñanza han sido fundamentales en el desarrollo de mis habilidades y conocimientos. Cada lección impartida, cada consejo dado, ha sido un paso más hacia la construcción de mi futuro profesional. Este proyecto de titulación es un testimonio del compromiso que han demostrado en la creación de profesionales de alto valor. Gracias por compartir su experiencia y sabiduría, y por inspirarnos a ser siempre la mejor versión de nosotros mismos.

Alexis Ivan Hernandez Rengifo

RESUMEN EJECUTIVO

El presente proyecto de titulación se enmarca dentro de los avances tecnológicos impulsados por la Industria 4.0, donde la automatización, la robótica inteligente y el análisis de datos en tiempo real son elementos clave para optimizar los procesos industriales, logísticos y urbanos. Dentro de este contexto, el desarrollo de robots móviles autónomos representa una línea estratégica de innovación, especialmente en aplicaciones como el transporte automatizado, inspección de áreas industriales, logística interna, monitoreo de entornos complejos y la formación en robótica educativa.

El objetivo principal de este trabajo fue el diseño, desarrollo e implementación de un controlador predictivo basado en modelos (Model Predictive Control - MPC) aplicado a un robot móvil diferencial ensamblado con el kit Lego Mindstorms EV3. A diferencia de los controladores clásicos, el MPC permite anticipar el comportamiento futuro del sistema, optimizando las acciones de control para seguir trayectorias definidas incluso en presencia de obstáculos o variaciones en el entorno.

El desarrollo se llevó a cabo utilizando el entorno de programación MATLAB y su herramienta Simulink, donde se modeló la dinámica del robot, se configuró el algoritmo MPC, y se realizaron simulaciones previas en entornos virtuales 3D con Simscape Multibody. Posteriormente, se implementó el código directamente sobre el hardware EV3, permitiendo validar el rendimiento en escenarios controlados y reales. Se diseñaron rutinas de evasión de obstáculos utilizando sensores ultrasónicos integrados, asegurando la navegación autónoma sin colisiones.

Los resultados experimentales confirmaron la eficacia del controlador MPC, logrando un seguimiento preciso de trayectorias con mejores niveles de estabilidad y adaptabilidad frente a métodos tradicionales como el control PID. El robot fue capaz de adaptarse dinámicamente ante la presencia de obstáculos, ajustando su trayectoria de forma eficiente.

Este trabajo no solo valida el uso de técnicas avanzadas de control predictivo en sistemas robóticos móviles, sino que también abre posibilidades de aplicación en entornos de manufactura flexible, transporte autónomo y ciudades inteligentes, alineándose con los desafíos y oportunidades que presenta la Industria 4.0.

PALABRAS CLAVE

Matlab, control, industria 4.0, trayectoria, simulink, obstaculos.

ABSTRACT

This thesis project is framed within the technological advances driven by Industry 4.0, where automation, intelligent robotics, and real-time data analysis are key elements to optimize industrial, logistical, and urban processes. In this context, the development of autonomous mobile robots represents a strategic innovation path, especially for applications such as automated transport, industrial area inspection, internal logistics, complex environment monitoring, and educational robotics.

The main objective of this work was the design, development, and implementation of a Model Predictive Control (MPC) system applied to a differential mobile robot assembled using the Lego Mindstorms EV3 kit. Unlike classical control methods, MPC allows anticipating the system's future behavior by optimizing control actions to follow predefined trajectories even in the presence of obstacles or environmental variations.

The development was carried out using MATLAB and Simulink, where the robot dynamics were modeled, the MPC algorithm was configured, and preliminary simulations were conducted using 3D virtual environments with Simscape Multibody. Subsequently, the control code was deployed directly onto the EV3 hardware, allowing performance validation in both controlled and real environments. Obstacle avoidance routines were implemented using integrated ultrasonic sensors to ensure autonomous navigation without collisions.

The experimental results confirmed the effectiveness of the MPC controller, achieving accurate trajectory tracking with greater stability and adaptability compared to traditional methods such as PID control. The robot dynamically adjusted its trajectory when facing obstacles, optimizing its navigation path efficiently.

This work not only validates the application of advanced predictive control techniques in mobile robotic systems but also opens possibilities for their application in flexible

manufacturing environments, autonomous transport, and smart cities, fully aligned with the challenges and opportunities posed by Industry 4.0.

KEYWORDS

Matlab, control, industry 4.0, trajectory, simulink, obstacle.

ÍNDICE GENERAL

TODOS LOS DERECHOS RESERVADOS.....	3
APROBACIÓN DEL TUTOR/A	4
DEDICATORIA.....	10
AGRADECIMIENTO.....	11
RESUMEN EJECUTIVO	12
PALABRAS CLAVE.....	13
ABSTRACT	14
KEYWORDS	15
Índice de Tablas y Códigos	18
Índice de Figuras	19
1.....	20
2. INTRODUCCIÓN.....	21
1. CAPÍTULO I.....	23
1.1. Planteamiento del problema	23
1.2. Preguntas de directrices	24
1.3. Objetivo General.....	24
1.4. Objetivos Especifico.....	24
1.5. Justificación	24
1.6. Alcance	26
2. CAPÍTULO II.....	28
2.1. Fundamento teórico relacionado al trabajo de integración curricular	28
2.1.1. Robótica y Navegación Autónoma	30
2.1.2. Control Predictivo Basado en Modelos (MPC).....	32
2.1.3. Simulación y Control en MATLAB y Simulink.....	35
2.1.4. Componentes del EV3 Utilizados en el Proyecto	38
3. CAPÍTULO III	41
3.1. El método científico.....	41
3.1.1. Tipos de investigación	44
3.1.2. Métodos teóricos de investigación	45
3.1.3. Técnicas e Instrumentos de Recolección de Datos.....	47
3.1.4. Tratamiento de la información.....	50
4. CAPÍTULO IV	51
4.1. Diseño de la propuesta.....	51

4.1.1.	Definición de Requisitos Técnicos	51
4.1.2.	Selección de Componentes y Herramientas	51
4.2.	Implementación	52
4.2.1.	Integración del Hardware.....	52
4.2.2	Desarrollo del Software.....	56
4.3.	Pruebas	67
4.4.	Resultados.....	71
5.	CAPITULO V	72
6.	CONCLUSIONES Y RECOMENDACIONES	72
5.1	Conclusiones.....	72
	Alcance del Objetivo del Proyecto:	72
	Evaluación del Rendimiento del Robot:	72
	Lecciones Aprendidas:	72
	Limitaciones del Proyecto:	73
5.2	Recomendaciones	73
	Mejoras en la Precisión de Sensores	73
	Optimización de Consumo Energético	73
	Integración de Algoritmos de Aprendizaje	73
	Consideraciones para Futuras Investigaciones:	73
5.3	Referencias	74
ANEXOS		75
6.1	Configuración del Proyecto en Simulink.....	75
6.2	Importación del Modelo CAD en Simulink.....	78
	6.2.1 Requisitos Previos	78
	6.2.2 Procedimiento Paso a Paso	78
	6.2.3 Configuración de Simulación	80
6.3	Construcción Física del Robot.....	81
6.4	RESULTADO FINAL.....	86

Índice de Tablas y Códigos

Tabla 3.1 Definición de Método Científico.....	41
Código 4.1 Variables para definir velocidad de los motores.....	61
Código 4.2 Lógica de detección de distancia	62
Código 4.3 Lógica de giros	63
Código 4.4 Lógica de giro a la derecha	63
Código 4.5 Lógica de giro a la izquierda	64
Código 4.6 Ajuste de velocidad para corregir trayectoria	64
Código 6.1 Ubicación de proyecto	78
Código 6.2 Importación de archivo EV3_CAD.xml	79

Índice de Figuras

Figura 2.1 Ecuaciones para modelo de robot móvil diferencial	34
Figura 3.2 Bloque EV3	38
Figura 3.3 Conexiones del bloque EV3	39
Figura 3.4 Sensor Ultrasónico EV3	39
Figura 3.5 Motor Servo EV3	40
Figura 3.6 Herramientas MATLAB y Simulink.....	40
Figura 4.1 Alineación de motores y ruedas	52
Figura 4.2 Posicionamiento de sensor ultrasónico	53
Figura 4.3 Conexión de cables	54
Figura 4.4 Conexión de motores.....	54
Figura 4.5 Conexión de sensor ultrasónico	55
Figura 4.6 Colocación de pilas	56
Figura 4.7 Instalación de Simulink Support Package.....	57
Figura 4.8 Instalación de Simscape Multibody	57
Figura 4.9 Instalación de Simscape	58
Figura 4.10 Panel de Add-Ons	58
Figura 4.11 Esquema del modelo del robot.....	59
Figura 4.12 Sensor ultrasónico en la librería de Simulink	60
Figura 4.13 Lógica del funcionamiento del sensor.....	60
Figura 4.14 Programación sobre bloques	61
Figura 4.16 Programación de estado evasión de colisión.....	62
Figura 4.17 Programación de Decisión de Giro	63
Figura 4.18 Programación de Giro a la Derecha	63
Figura 4.19 Programación de giro a la izquierda.....	64
Figura 4.20 Programación de corrección de trayectoria.....	65
Figura 4.21 Programación de Regreso al estado Normal	65
Figura 4.21 Bloques para control de ruedas	66
Figura 4.22 Modelo completo	67
Figura 4.23 Ubicación de robot en circuito	68
Figura 4.24 Robot evadiendo conos	68
Figura 4.25 Conexión a bloque EV3	69
Figura 4.26 Conexión de modelo a hardware.....	69

Figura 4.27 Compilación de modelo	70
Figura 4.28 Construcción de modelo sobre bloque EV3	71
Figura 6.1 Comando para inicio de Simulink.....	75
Figura 6.2 Código en Stateflow	76
Figura 6.3 Código de Estados.....	76
Figura 6.4 Selección de Bluetooth para conexión con EV3	77
Figura 6.5 Esquema de trabajo con modelo completo en Simulink	78
Figura 6.7 Importación del modelo	79
Figura 6.8 Modelo de Arranque de simulación	80
Figura 6.9 Arranque de simulación	80
Figura 6.10 Servomotor.....	81
Figura 6.11 Sensor ultrasónico	81
Figura 6.12 Bloque mindstorm trasero	81
Figura 6.13 Bloque mindstorm delantero	82
Figura 6.14 Rueda con 6 orificios	82
Figura 6.15 Viga 3x3 con pasadores	82
Figura 6.16 Eje largo con tope.....	82
Figura 6.17 Ensamblador de eje	83
Figura 6.18 Neumatico RG.....	83
Figura 6.19 Pin largo técnico.....	83
Figura 6.20 Soporte elevador 3x5	83
Figura 6.21 Soporte elevador 1x15	84
Figura 6.22 Soporte elevador 2x4	84
Figura 6.23 Eje técnico	84
Figura 6.24 Rotor de servo motor.....	84
Figura 6.25 Robot ensamblado vista delantera.....	85
Figura 6.26 Robot ensamblado vista latera	85
Figura 6.27 Robot ensamblado vista posterior lateral	85

INTRODUCCIÓN

La robótica móvil ha experimentado un notable avance en las últimas décadas, impulsada por el deseo de crear sistemas autónomos capaces de realizar tareas complejas en entornos variados y dinámicos. Uno de los aspectos clave en el diseño de robots móviles es la capacidad de seguir trayectorias predeterminadas con alta precisión, lo que es fundamental para aplicaciones que van desde la exploración y el mapeo hasta la navegación en entornos industriales y urbanos.

En este contexto, los controladores predictivos basados en modelos (MPC, por sus siglas en inglés) han surgido como una solución avanzada y eficaz para el control de sistemas dinámicos. El MPC se distingue por su capacidad para manejar restricciones tanto en el sistema controlado como en las entradas del controlador, y por su habilidad para predecir el comportamiento futuro del sistema, ajustando las acciones de control en tiempo real. Esta capacidad predictiva y adaptativa lo convierte en una herramienta invaluable para mejorar el desempeño de los robots móviles, permitiéndoles seguir trayectorias con mayor precisión y adaptarse a cambios en el entorno.

El presente proyecto de titulación se centra en la implementación de un controlador MPC para un robot móvil diferencial, utilizando el kit Lego Mindstorms EV3 como plataforma educativa. Este kit, conocido por su flexibilidad y accesibilidad, proporciona una base ideal para el desarrollo y prueba de algoritmos de control avanzado. El robot diferencial, caracterizado por su configuración de ruedas que permite maniobras ágiles y efectivas, será el objeto de estudio para la aplicación del controlador MPC.

El objetivo principal del proyecto es desarrollar un controlador MPC que permita al robot móvil diferencial seguir trayectorias predefinidas con alta precisión. Para ello, se implementará el algoritmo MPC en el entorno de MATLAB, que es ampliamente reconocido por sus capacidades de simulación y modelado. MATLAB facilitará la configuración del algoritmo, la simulación del comportamiento del robot y la implementación del controlador en el hardware real.

El alcance del proyecto abarca varias etapas fundamentales. Inicialmente, se realizará la configuración del entorno de trabajo en MATLAB, seguida de la simulación y modelado del robot en el entorno de Simulink. La estimación de parámetros del motor y la configuración de la comunicación entre Simulink y el robot serán pasos cruciales para garantizar la precisión del modelo y la efectividad del controlador. Finalmente, se llevará a cabo la implementación del controlador MPC en el robot real, seguido de pruebas y ajustes en tiempo real para evaluar el desempeño del sistema.

Este proyecto de titulación no solo busca validar la eficacia del controlador MPC en el contexto de un robot móvil diferencial, sino también explorar su aplicabilidad en entornos de la Industria 4.0 y en el diseño de ciudades inteligentes. La integración de tecnologías avanzadas en plataformas educativas como el Lego Mindstorms EV3 abre nuevas oportunidades para el desarrollo de habilidades en control avanzado y robótica, y contribuye al avance en la investigación y desarrollo en estos campos emergentes.

En resumen, este proyecto representa un esfuerzo significativo para avanzar en la capacidad de los robots móviles para seguir trayectorias de manera precisa y adaptable, utilizando herramientas avanzadas como el controlador MPC y el software MATLAB. Los resultados y conclusiones del proyecto proporcionarán valiosos conocimientos y contribuirán al desarrollo de soluciones innovadoras en el ámbito de la robótica y el control.

CAPÍTULO I

EL PROBLEMA DE INVESTIGACIÓN

1.1. Planteamiento del problema

El campo de la robótica móvil está experimentando un crecimiento acelerado, impulsado por la necesidad de robots capaces de realizar tareas complejas en entornos dinámicos y no estructurados. Un desafío central en esta área es la capacidad de los robots para seguir trayectorias con alta precisión mientras se enfrentan a variables y condiciones cambiantes. Los controladores tradicionales, como los basados en el algoritmo PID (Proporcional, Integral, Derivativo), han sido ampliamente utilizados debido a su simplicidad y efectividad en sistemas lineales y bien definidos. Sin embargo, estos métodos presentan limitaciones significativas cuando se aplican a sistemas con restricciones múltiples y comportamientos no lineales.

El controlador predictivo basado en modelos (MPC) se presenta como una solución avanzada que supera estas limitaciones al ofrecer un enfoque más robusto para el control de sistemas dinámicos complejos. El MPC utiliza un modelo del sistema para prever su comportamiento futuro y optimizar el control en función de las predicciones y restricciones impuestas, lo que permite una mejor adaptación a cambios en el entorno y una mayor precisión en el seguimiento de trayectorias.

El problema específico que aborda este proyecto es la adaptación e implementación del controlador MPC en un robot móvil diferencial construido con el kit Lego Mindstorms EV3. Este kit, diseñado principalmente para fines educativos y de prototipado rápido, no está optimizado para técnicas avanzadas de control como el MPC. La integración del MPC en esta plataforma plantea un desafío significativo, dado que se debe adaptar un algoritmo matemático sofisticado a un sistema que por su naturaleza es accesible y sencillo. Resolver este problema no solo implica desarrollar un modelo de control eficiente, sino también validar su efectividad en un entorno práctico, lo que requiere una evaluación exhaustiva de su rendimiento en diversas condiciones operativas.

1.2. Preguntas de directrices

¿Cuál es el proceso necesario para adaptar el algoritmo MPC al robot móvil diferencial utilizando el kit Lego Mindstorms EV3, y qué modificaciones son requeridas para integrar esta técnica avanzada en un entorno educativo?

¿Qué ventajas específicas ofrece el controlador MPC en comparación con el controlador PID tradicional en el contexto de seguimiento de trayectorias para un robot móvil diferencial?

¿En qué medida el MPC puede mejorar la capacidad del robot para adaptarse a cambios en el entorno y manejar situaciones imprevistas en comparación con controladores tradicionales?

1.3. Objetivo General

Desarrollar e implementar un controlador predictivo basado en modelos (MPC) en un robot móvil diferencial utilizando el kit Lego Mindstorms EV3, a través del software MATLAB, con el fin de mejorar la precisión en el seguimiento de trayectorias predefinidas y optimizar la respuesta del robot en distintas condiciones.

1.4. Objetivos Especifico

OBJ 1: Configurar el entorno de desarrollo en MATLAB para la simulación e implementación del controlador MPC, asegurando la disponibilidad de herramientas y bibliotecas necesarias.

OBJ 2: Desarrollar un modelo dinámico del robot móvil diferencial en Simulink, incorporando la estimación precisa de los parámetros del motor para representar fielmente su comportamiento real.

OBJ 3: Diseñar e implementar el controlador MPC en Simulink, definiendo predicciones y restricciones que optimicen el seguimiento de trayectorias y mejoren la precisión del control del robot.

1.5. Justificación

La implementación de controladores predictivos basados en modelos (Model Predictive Control, MPC) en la robótica móvil representa una tendencia de vanguardia

que aborda las limitaciones inherentes a los métodos de control tradicionales como el PID, especialmente en entornos dinámicos donde las condiciones de operación pueden cambiar rápidamente (Camacho & Bordons, 2013). A diferencia de los controladores convencionales, el MPC permite anticipar el comportamiento futuro del sistema utilizando un modelo matemático, optimizando la trayectoria bajo restricciones físicas y operativas (Mayne et al., 2000).

En el ámbito educativo, el uso del kit Lego Mindstorms EV3 facilita un entorno accesible para la implementación de técnicas avanzadas de control. Según González-González et al. (2021), plataformas como Lego EV3 no solo fomentan el aprendizaje activo en áreas de robótica y automatización, sino que también permiten la transferencia efectiva de conocimientos de alto nivel en técnicas de control avanzado, como el MPC, en contextos didácticos y experimentales.

La relevancia de este proyecto radica en que permite llevar una metodología propia de aplicaciones industriales complejas —como la automatización flexible, la manufactura inteligente o los vehículos autónomos— a un escenario educativo experimental (Qin & Badgwell, 2003). Este enfoque brinda a los estudiantes y futuros ingenieros la oportunidad de comprender y dominar algoritmos modernos, los cuales son fundamentales para los desarrollos tecnológicos de la Industria 4.0 (Lu, 2017).

Adicionalmente, el control predictivo es ampliamente considerado como uno de los métodos más robustos para el seguimiento de trayectorias en presencia de obstáculos, ya que permite integrar variables multivariantes, restricciones y predicciones futuras, logrando mayor estabilidad y seguridad en la navegación (Falcone et al., 2007). Estas capacidades son esenciales para aplicaciones emergentes en vehículos autónomos, robots logísticos, vigilancia automatizada y sistemas de transporte inteligentes dentro del concepto de Smart Cities (Hussain et al., 2019).

Por lo tanto, este proyecto no solo ofrece una solución técnica robusta a un desafío en la robótica móvil, sino que además contribuye directamente al desarrollo de competencias altamente demandadas en los campos de automatización avanzada, robótica inteligente, control en tiempo real, y tecnologías asociadas a la Cuarta Revolución Industrial.

1.6. Alcance

El presente proyecto tiene un alcance integral que cubre todas las fases del desarrollo, desde la concepción inicial hasta la validación experimental de un sistema de control predictivo basado en modelos (MPC) aplicado a un robot móvil diferencial construido con el kit Lego Mindstorms EV3. El proyecto contempla tanto el desarrollo teórico como la implementación práctica, con el objetivo de evaluar el desempeño real del sistema bajo diferentes condiciones de operación.

En la primera fase, se realizará la configuración completa del entorno de trabajo, lo que incluye la instalación de herramientas de desarrollo como MATLAB, Simulink, Simscape Multibody y los paquetes de soporte para LEGO EV3. Posteriormente, se procederá al modelado matemático del robot en el entorno de Simulink, permitiendo simular su comportamiento dinámico en un escenario virtual controlado, el cual servirá como base para el diseño del controlador MPC.

El proyecto incluye un proceso riguroso de estimación de parámetros, donde se recopilarán datos experimentales del robot real para ajustar el modelo simulado a la realidad física del sistema. Este proceso es fundamental para garantizar que el modelo predictivo sea representativo del comportamiento real del robot.

Una vez obtenido un modelo validado, se diseñará el controlador MPC considerando las restricciones físicas del sistema (velocidad, aceleración, distancia mínima de seguridad ante obstáculos, etc.), así como los objetivos de seguimiento de trayectoria y evasión de obstáculos. El controlador se ajustará inicialmente en el entorno simulado y, posteriormente, se implementará directamente sobre el hardware físico del robot, permitiendo validar el desempeño del sistema en tiempo real.

Durante la etapa de pruebas, el robot será sometido a diversos escenarios de navegación, incluyendo trayectorias rectas, curvas, y entornos con obstáculos dinámicos, con el objetivo de analizar la robustez y adaptabilidad del controlador MPC frente a cambios en

el entorno. Estas pruebas se realizarán tanto en condiciones controladas de laboratorio como en entornos semi-reales que simulen aplicaciones prácticas.

El proyecto también contempla la comparación del desempeño obtenido mediante el controlador MPC frente a resultados de controladores tradicionales, como el PID, para cuantificar los beneficios en términos de precisión, estabilidad, adaptabilidad y eficiencia energética. De esta manera, se podrá evaluar de forma objetiva la superioridad del enfoque predictivo en aplicaciones de robótica móvil.

Finalmente, se documentará exhaustivamente cada una de las etapas del proyecto, desde el diseño, simulación, implementación, pruebas y análisis de resultados, generando un aporte académico que puede ser utilizado como base para futuros desarrollos, investigaciones o aplicaciones industriales dentro del marco de la Industria 4.0, sistemas autónomos y robótica educativa.

CAPÍTULO II

MARCO DE REFERENCIAS

2.1. Fundamento teórico relacionado al trabajo de integración curricular

El presente proyecto de titulación se sustenta en un conjunto de principios teóricos interdisciplinarios que abarcan áreas fundamentales de la ingeniería moderna, tales como la robótica móvil, la inteligencia computacional, el modelado de sistemas dinámicos, los algoritmos de control avanzado y las herramientas de simulación digital. Este marco teórico proporciona la base conceptual necesaria para la creación, modelado, simulación y control de un robot móvil autónomo diferencial, integrando herramientas educativas de vanguardia como el kit Lego Mindstorms EV3 con entornos profesionales de desarrollo como MATLAB y Simulink.

Robótica móvil diferencial

La robótica móvil constituye una rama fundamental dentro de la ingeniería de automatización y control, donde los sistemas autónomos deben desplazarse y adaptarse a un entorno dinámico. En particular, el robot móvil diferencial, caracterizado por tener dos ruedas motrices dispuestas en paralelo y un punto de apoyo libre, ofrece una estructura mecánica sencilla pero capaz de realizar movimientos complejos mediante la modulación de la velocidad de cada rueda (Siegwart et al., 2011). Este tipo de robot representa un modelo ideal para estudiar algoritmos de control de navegación autónoma debido a su dinamismo, simplicidad estructural y desafíos de control.

Navegación autónoma y evasión de obstáculos

La navegación autónoma requiere que el robot pueda percibir su entorno, planificar rutas y ejecutar maniobras evitando colisiones. Para este fin, se emplean sensores de distancia como el sensor ultrasónico incorporado en el kit EV3, capaz de detectar objetos y medir distancias en tiempo real (LEGO Group, 2023). Los algoritmos de evasión de obstáculos permiten modificar la trayectoria del robot anticipando la posición de los obstáculos, garantizando un desplazamiento fluido y seguro (LaValle, 2006).

Simulación y modelado en MATLAB y Simulink

El uso de herramientas de simulación digital es indispensable para validar algoritmos antes de su implementación en el hardware físico. MATLAB y Simulink ofrecen un entorno integrado para el modelado de sistemas dinámicos, la validación de controladores y la visualización de trayectorias (MathWorks, 2023). Particularmente, Simscape Multibody permite modelar los aspectos físicos del robot, como masas, inercias y fuerzas de fricción, proporcionando una representación realista del comportamiento del sistema. Esta capacidad de simulación reduce los riesgos de fallos en pruebas físicas, permitiendo un ajuste fino de los parámetros de control.

Control Predictivo Basado en Modelos (MPC)

El Control Predictivo Basado en Modelos (MPC, por sus siglas en inglés) es una técnica avanzada de control que, a diferencia de los controladores clásicos (como el PID), optimiza el desempeño futuro del sistema mediante la predicción de su comportamiento dinámico a lo largo de un horizonte de tiempo definido (Rawlings et al., 2017). El MPC es especialmente valioso en aplicaciones de robótica móvil, donde existen múltiples restricciones físicas (velocidades máximas, distancia mínima ante obstáculos, aceleración limitada, etc.) y es necesario anticiparse a cambios en el entorno. Su capacidad de manejar restricciones y optimizar en tiempo real la trayectoria convierte al MPC en una herramienta robusta para sistemas autónomos complejos.

Aplicación en el contexto de Industria 4.0 y robótica educativa

El desarrollo de este proyecto se enmarca dentro del paradigma de la Industria 4.0, donde la automatización, el control inteligente y los sistemas ciberfísicos desempeñan un papel fundamental (Schwab, 2016). La capacidad de integrar tecnologías como la simulación digital, el control predictivo y la robótica móvil diferencial ofrece no solo soluciones prácticas para la industria, sino también entornos educativos innovadores que fomentan el aprendizaje basado en proyectos reales y experimentación directa (Almeida et al., 2020). Además, el kit Lego Mindstorms EV3 proporciona un recurso didáctico accesible, flexible y ampliamente utilizado en instituciones educativas para introducir conceptos avanzados de robótica, programación y control.

En conjunto, los fundamentos teóricos aquí presentados demuestran cómo el proyecto integra múltiples áreas de conocimiento para alcanzar un objetivo práctico de alta relevancia, aportando tanto a la formación profesional como al desarrollo de competencias clave para los desafíos de la ingeniería moderna.

2.1.1. Robótica y Navegación Autónoma

Robótica y Sistemas Autónomos

La robótica es una disciplina multidisciplinaria que integra principios de la ingeniería mecánica, electrónica, informática e inteligencia artificial, orientada al diseño, construcción, programación y operación de sistemas que interactúan con el mundo físico (Siciliano & Khatib, 2016). Los robots autónomos son aquellos capaces de operar sin intervención humana directa, gracias a la capacidad de percibir su entorno, procesar la información obtenida y ejecutar acciones que les permiten cumplir con sus objetivos de manera independiente.

El desarrollo de sistemas autónomos representa uno de los mayores retos y oportunidades en el contexto actual de la Industria 4.0, dado su potencial para realizar tareas complejas en entornos dinámicos, peligrosos o inaccesibles para los seres humanos (Schwab, 2016). En el ámbito educativo, el uso de kits como LEGO Mindstorms EV3 permite replicar estas capacidades de forma simplificada pero funcional, permitiendo explorar los fundamentos de los sistemas autónomos en un entorno accesible (LEGO Group, 2023).

Navegación Autónoma

La navegación autónoma constituye uno de los aspectos más críticos dentro de la robótica móvil. Se refiere a la capacidad del robot para desplazarse de manera segura y eficiente en su entorno, evitando colisiones, respetando restricciones físicas, y adaptándose a cambios inesperados en el entorno (Thrun et al., 2005). La navegación autónoma requiere la integración de varios subsistemas:

Percepción del entorno: mediante sensores, como los ultrasónicos o lidar, que permiten obtener información del espacio circundante.

Planeamiento de ruta: algoritmos que calculan la trayectoria óptima hasta un destino.

Control de movimiento: algoritmos que transforman las trayectorias planificadas en comandos de velocidad y dirección para los motores.

En este proyecto, la navegación autónoma del robot móvil diferencial se centra principalmente en el seguimiento de trayectorias predefinidas, incorporando evasión de obstáculos mediante la información obtenida del sensor ultrasónico.

Algoritmos de Evasión de Obstáculos

Para que un robot navegue de forma autónoma, es necesario que sea capaz de identificar y evitar obstáculos inesperados durante su trayectoria. Existen diferentes estrategias para abordar este problema:

a) Evasión Reactiva:

El control reactivo es uno de los enfoques más simples y rápidos para evitar obstáculos. Se basa en reglas simples donde, ante la detección de un objeto a una distancia crítica mediante sensores (por ejemplo, el sensor ultrasónico del EV3), el robot modifica instantáneamente su dirección o velocidad (Arkin, 1998). Este método es útil en entornos poco estructurados, pero puede generar movimientos erráticos si no se complementa con planificación anticipada.

b) Planificación de Trayectoria:

Cuando se dispone de un mapa parcial o completo del entorno, se pueden aplicar algoritmos de búsqueda como A* o Dijkstra para calcular rutas óptimas que esquiven los obstáculos anticipadamente (LaValle, 2006). Estos algoritmos generan trayectorias globales óptimas, pero requieren mayor capacidad de procesamiento y mapas actualizados del entorno.

c) Optimización en Tiempo Real con MPC:

El Control Predictivo Basado en Modelos (MPC) representa un enfoque más sofisticado. Este controlador predice el comportamiento futuro del robot en función de su modelo dinámico y optimiza continuamente su trayectoria considerando las

restricciones de espacio, velocidad, aceleración y la posición de los obstáculos detectados (Mayne et al., 2000). La capacidad del MPC para evaluar múltiples escenarios de forma anticipada permite que el robot tome decisiones más seguras y suaves frente a situaciones imprevistas, mejorando su adaptabilidad.

Sensores y Actuadores en Robótica Autónoma

Sensores:

Los sensores constituyen los “ojos y oídos” del robot. En este proyecto, el sensor ultrasónico conectado al puerto 1 del controlador EV3 permite medir distancias mediante la emisión de ondas ultrasónicas y la detección de su eco. Esta información es fundamental para identificar obstáculos y estimar el entorno inmediato (LEGO Group, 2023).

Actuadores:

Los actuadores son los elementos encargados de ejecutar físicamente los movimientos calculados. En el robot EV3, los motores conectados a los puertos B y C controlan el movimiento de las ruedas diferenciales, permitiendo avanzar, retroceder o girar mediante la modulación independiente de sus velocidades. La precisión en el control de estos motores es crítica para la correcta ejecución de las trayectorias planificadas.

Integración de Sensores, Actuadores y Control

La interacción coordinada entre sensores, actuadores y algoritmos de control permite que el robot móvil diferencial navegue de manera autónoma por su entorno. Los datos en tiempo real del sensor ultrasónico alimentan continuamente el algoritmo de MPC, que calcula la mejor secuencia de acciones para seguir la trayectoria planificada evitando obstáculos. Esta integración es una representación fiel de los principios de los sistemas ciberfísicos característicos de la Industria 4.0 (Monostori, 2014), donde el mundo físico y el virtual se interconectan de forma continua y adaptativa.

2.1.2. Control Predictivo Basado en Modelos (MPC)

El Control Predictivo Basado en Modelos (MPC, por sus siglas en inglés: Model Predictive Control) constituye una de las estrategias de control más avanzadas y potentes en el ámbito de la automatización moderna. A diferencia de los controladores tradicionales, como el PID, el MPC no sólo actúa en función del error presente, sino

que anticipa el comportamiento futuro del sistema a través de un modelo dinámico, optimizando continuamente su desempeño bajo múltiples restricciones físicas y operativas (Qin & Badgwell, 2003).

En el contexto de la robótica móvil diferencial, el MPC permite planificar de forma continua las maniobras del robot, ajustando su trayectoria en tiempo real mientras considera limitaciones de velocidad, aceleración, distancias mínimas a obstáculos y bordes del entorno, ofreciendo así una navegación autónoma más suave, precisa y segura.

Principios Fundamentales del MPC

El funcionamiento básico del MPC se articula en tres fases principales que se repiten de forma continua durante la operación del robot:

Predicción del comportamiento futuro:

Utilizando un modelo matemático que describe la dinámica del robot móvil diferencial, el MPC simula cómo evolucionará el sistema ante diferentes acciones de control en un horizonte de predicción (por ejemplo, los próximos 10 segundos).

Optimización de la trayectoria:

Con base en las predicciones, el controlador resuelve un problema de optimización que busca minimizar una función objetivo (coste), como el error respecto a la trayectoria deseada o el esfuerzo de control, mientras respeta las restricciones impuestas:

- Límites de velocidad lineal y angular.
- Distancia mínima ante obstáculos.
- Aceleración máxima admisible.
- Posibles zonas prohibidas o áreas restringidas.

Aplicación de la primera acción óptima:

Aunque el MPC calcula una secuencia completa de acciones óptimas, sólo implementa el primer valor de la secuencia en el instante actual. En el siguiente ciclo

de control, el proceso se repite incorporando la nueva información sensorial del entorno.

Este enfoque de optimización recursiva convierte al MPC en un controlador anticipativo y adaptativo, capaz de modificar su plan de acción en tiempo real ante cambios inesperados o dinámicas no previstas del entorno.

Modelo Matemático del Robot Móvil Diferencial

Para que el MPC pueda realizar predicciones acertadas, es esencial contar con un modelo matemático preciso que represente la cinemática del robot. En el caso de un robot móvil diferencial como el construido con el kit LEGO Mindstorms EV3, se emplea un modelo cinemático planar no holonómico, descrito mediante las siguientes ecuaciones de estado:

$$\begin{aligned}\dot{x} &= v \cos(\theta) \\ \dot{y} &= v \sin(\theta) \\ \dot{\theta} &= \frac{R}{L}(w_r - w_l)\end{aligned}$$

Figura 2.1 Ecuaciones para modelo de robot móvil diferencial

Donde:

v es la velocidad lineal del robot.

θ es la orientación del robot.

w son las velocidades de las ruedas derecha e izquierda, respectivamente.

R es el radio de las ruedas y L es la distancia entre ellas.

Con este modelo, el MPC predice la posición futura del robot y ajusta las velocidades de las ruedas para seguir la trayectoria deseada mientras evita obstáculos.

Ventajas del MPC frente a Controladores Convencionales

Comparado con otros enfoques de control, como el **PID** o el **control reactivo**, el MPC ofrece notables ventajas:

- Anticipación de eventos futuros mediante simulación interna.
- Capacidad de manejar múltiples restricciones de forma explícita.
- Adaptabilidad frente a perturbaciones o cambios dinámicos en el entorno.

- Mayor precisión en el seguimiento de trayectorias complejas.
- Comportamiento más suave, sin oscilaciones bruscas.

Aplicación del MPC en el Proyecto

En el desarrollo de este proyecto, el algoritmo MPC ha sido programado y simulado inicialmente en **Simulink (MATLAB)**, validando su desempeño mediante modelos virtuales y escenarios controlados. Posteriormente, el controlador fue implementado sobre el hardware físico del robot EV3, integrando la lectura continua del sensor ultrasónico para la detección de obstáculos en tiempo real. Esta integración permite al robot seguir trayectorias definidas mientras evita colisiones y respeta sus propias limitaciones mecánicas.

El uso de MPC en sistemas robóticos móviles representa un ejemplo directo de cómo los avances en control predictivo están siendo incorporados cada vez más en aplicaciones prácticas vinculadas a los desafíos de la **Industria 4.0**, la automatización inteligente y las ciudades inteligentes.

2.1.3. Simulación y Control en MATLAB y Simulink

El uso de **MATLAB** y **Simulink** constituye un pilar fundamental en el desarrollo, simulación, validación y posterior implementación del sistema de control del robot móvil diferencial. Estas herramientas permiten abordar de forma integrada tanto el modelado matemático del sistema físico, como el diseño y optimización de los algoritmos de control, asegurando un entorno flexible, interactivo y altamente eficiente para el desarrollo de soluciones avanzadas en robótica.

Modelado y Simulación en Simulink

Simulink proporciona un entorno de modelado gráfico basado en bloques, ideal para representar los sistemas dinámicos que intervienen en el desplazamiento y control del robot.

Creación del modelo del robot:

La estructura cinemática del robot móvil diferencial es representada mediante bloques funcionales que describen la dinámica de sus movimientos. Variables como la velocidad lineal, velocidad angular, posiciones x, y , $\dot{x}, \dot{y}$ y orientación

θ son simuladas mediante las ecuaciones matemáticas previamente definidas. Este modelo virtual permite replicar de forma precisa el comportamiento físico del robot bajo diferentes condiciones de operación.

Simulación de la navegación autónoma:

A través de la simulación de escenarios con trayectorias predefinidas y presencia de obstáculos virtuales, se evalúa la capacidad del sistema de control para mantener el seguimiento de la ruta deseada, ajustándose dinámicamente frente a interferencias o desviaciones. Este proceso permite validar el diseño del controlador en un entorno seguro, identificando posibles limitaciones antes de su implementación física.

Implementación del MPC en Simulink:

Simulink incorpora herramientas específicas para el diseño de controladores predictivos, como el bloque funcional **MPC Controller**, el cual permite definir:

- Horizonte de predicción.
- Horizonte de control.
- Restricciones físicas (velocidad, aceleración, posiciones seguras).
- Funciones objetivo de optimización.

La integración de este bloque al modelo del robot facilita la validación del comportamiento del **Control Predictivo Basado en Modelos (MPC)** durante las simulaciones, proporcionando resultados precisos sobre el desempeño del sistema frente a múltiples escenarios.

Adquisición y Procesamiento de Datos

La recopilación de datos sensoriales juega un rol clave en la calibración del modelo y la mejora del desempeño del controlador:

Adquisición de datos en tiempo real:

Durante las pruebas iniciales y el funcionamiento experimental, el robot recoge información proveniente de sus sensores (principalmente el sensor ultrasónico). Estos datos son capturados mediante la interfaz de **Simulink Support Package**

for LEGO Mindstorms EV3, permitiendo su almacenamiento, análisis y procesamiento dentro del entorno MATLAB.

Ajuste de parámetros del modelo:

La comparación entre los datos experimentales y las simulaciones permite realizar estimaciones precisas de parámetros físicos como:

- Radio real de las ruedas.
- Separación efectiva entre ruedas.
- Retardos o ruidos en la adquisición sensorial.

Este refinamiento del modelo mejora la exactitud de las predicciones del MPC, optimizando su capacidad de anticipación y su robustez ante perturbaciones reales.

Implementación en el Hardware

Una vez validado el controlador en las simulaciones, se procede a la fase de implementación física sobre el hardware:

Generación automática de código:

Simulink permite convertir el modelo gráfico en código ejecutable directamente implementable en el procesador del robot **Lego Mindstorms EV3**, minimizando errores de programación manual y garantizando que el comportamiento real reproduzca fielmente las simulaciones.

Ejecución y validación experimental:

El robot, ya programado con el algoritmo MPC, ejecuta las trayectorias deseadas en un entorno físico real, detectando obstáculos mediante el sensor ultrasónico, ajustando su trayectoria en tiempo real y validando la eficacia del controlador predictivo bajo condiciones dinámicas de operación.

Ventajas de la Simulación previa a la implementación

El enfoque de simulación permite:

- Reducir el riesgo de daños durante las pruebas físicas.

- Optimizar el diseño del controlador antes de enfrentar las condiciones reales.
- Ajustar parámetros finos del modelo de forma iterativa.
- Acelerar el proceso completo de desarrollo y validación.

En conjunto, el uso de **MATLAB** y **Simulink** no solo acelera el ciclo de desarrollo, sino que además brinda un nivel de confianza elevado en el comportamiento final del robot antes de su despliegue físico, lo que es crucial en aplicaciones de la **Industria 4.0**, donde la precisión y la adaptabilidad son aspectos fundamentales.

2.1.4. Componentes del EV3 Utilizados en el Proyecto

El robot diseñado para este proyecto se basa en la plataforma **Lego Mindstorms EV3**, la cual proporciona una variedad de sensores y actuadores esenciales para la navegación autónoma. Los componentes utilizados en el desarrollo del robot incluyen:

- **Unidad Central (Brick EV3)**

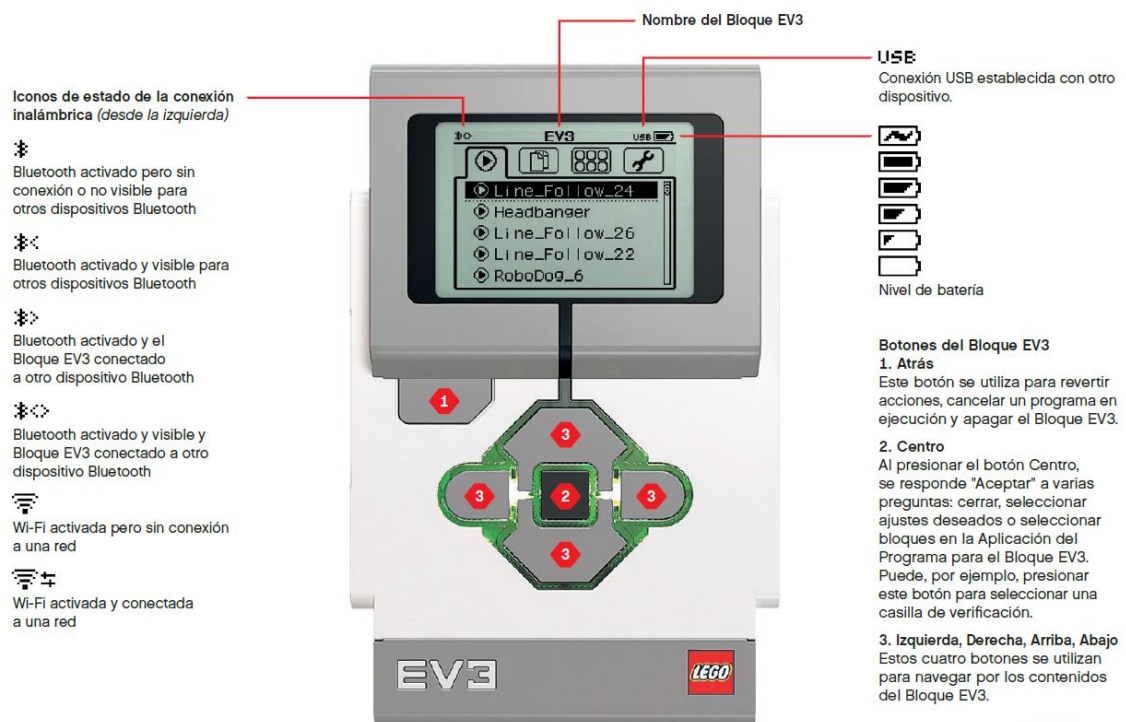


Figura 2.2 Bloque EV3



Figura 2.3 Conexiones del bloque EV3

Procesador: ARM9 de 300 MHz.

Memoria RAM: 64 MB.

Almacenamiento Interno: 16 MB, expandible con tarjeta microSD.

Interfaces: 4 puertos para sensores, 4 puertos para motores.

Conectividad: USB, Bluetooth y WiFi (con adaptador).

Funciones: Permite la programación en MATLAB/Simulink y ejecuta los algoritmos de control del robot.

- **Sensor Ultrasónico EV3:**



Figura 2.4 Sensor Ultrasónico EV3

Función: Mide la distancia a objetos y permite la detección de obstáculos.

Rango de detección: 1 a 250 cm.

Precisión: ± 1 cm.

Aplicación: Evasión de obstáculos mediante detección de proximidad.

- **Motores Servo EV3:**



Figura 2.5 Motor Servo EV3

Función: Permiten el movimiento del robot y control preciso de velocidad.

Torque: 20 N.cm (motor grande), 8 N.cm (motor mediano).

Resolución del encoder: 1° de precisión.

Aplicación: Movilidad y control de dirección del robot.

- **Software y Herramientas de Simulación**



Figura 2.6 Herramientas MATLAB y Simulink

MATLAB/Simulink: Modelado del robot, implementación del control predictivo y simulación de trayectorias.

Simscape Multibody: Simulación de la dinámica del robot en un entorno 3D.

LEGO Mindstorms EV3 Software: Interfaz nativa de programación y configuración de hardware.

CAPÍTULO III

MARCO METODOLÓGICO

3.1. El método científico

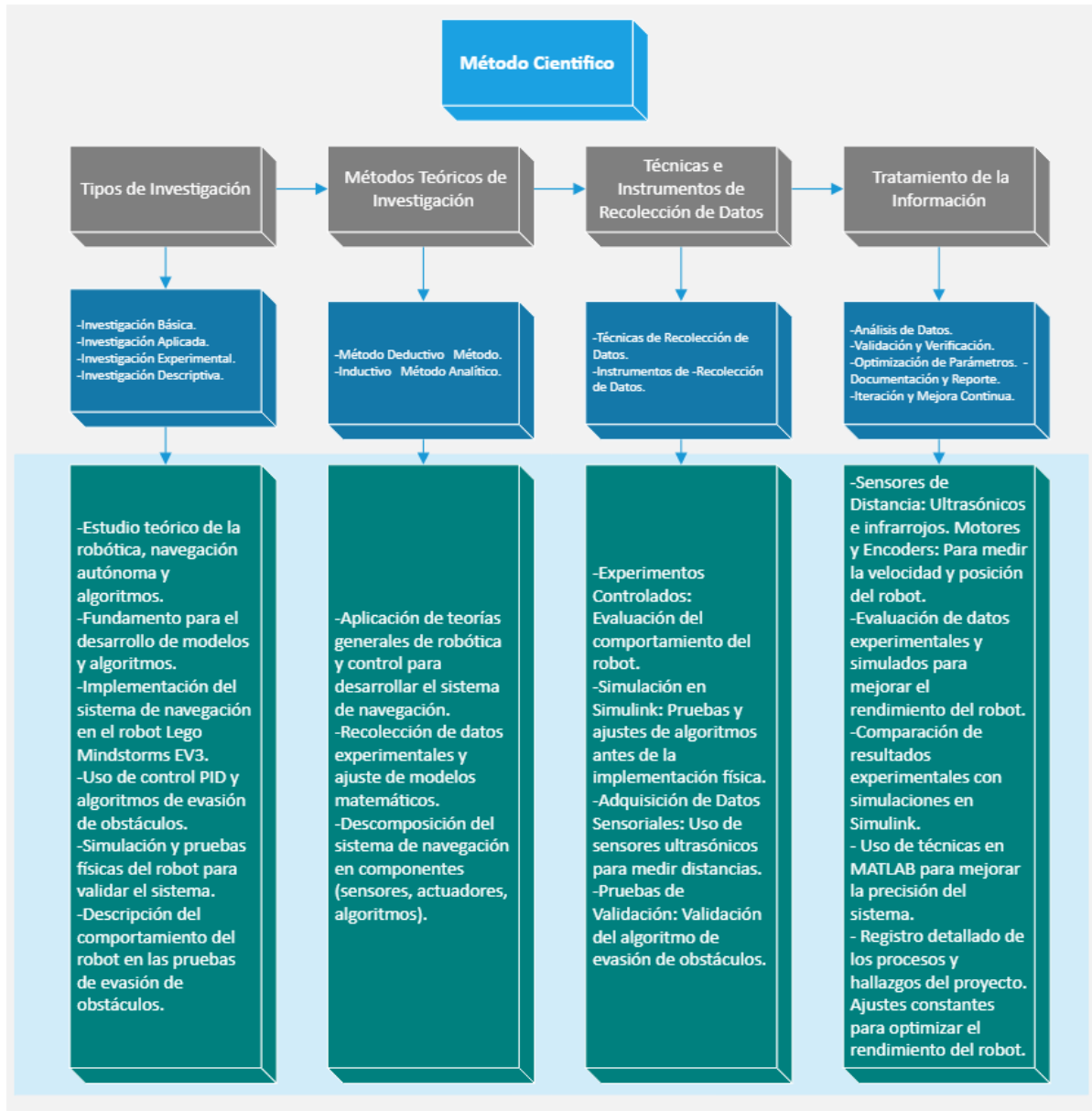


Tabla 3.1 Definición de Método Científico

El método científico constituye la base estructural de cualquier investigación rigurosa y sistemática, permitiendo el desarrollo de soluciones fundamentadas en evidencia empírica y lógica. Su aplicación asegura que los resultados obtenidos sean reproducibles,

verificables y coherentes, características esenciales dentro de los ámbitos de la investigación científica, tecnológica e industrial.

En el marco de este proyecto de titulación, el cual involucra el diseño e implementación de un controlador predictivo basado en modelos (MPC) para un robot móvil diferencial, el método científico ha sido la columna vertebral para guiar cada fase de trabajo, desde la definición inicial del problema hasta la validación experimental de los resultados.

Etapas del método científico aplicado al proyecto

Observación inicial y planteamiento del problema:

Se identificó la necesidad de implementar estrategias de control avanzadas que permitan mejorar la precisión en el seguimiento de trayectorias por parte de robots móviles diferenciales. Las limitaciones observadas en los métodos tradicionales, como el controlador PID, motivaron la búsqueda de alternativas más robustas y adaptativas, en este caso, el **MPC**.

Revisión bibliográfica:

Se llevó a cabo un exhaustivo estudio de literatura especializada sobre control predictivo, modelado de robots móviles, simulación en MATLAB/Simulink, y experiencias previas con plataformas educativas como **Lego Mindstorms EV3**. Esta revisión permitió fundamentar teóricamente el enfoque propuesto, identificar parámetros relevantes y conocer las mejores prácticas de implementación.

Formulación de hipótesis:

Se planteó como hipótesis principal que la implementación de un controlador predictivo basado en modelos (MPC) sobre un robot móvil diferencial permite alcanzar una mayor precisión y adaptabilidad en el seguimiento de trayectorias complejas, superando el rendimiento de controladores tradicionales bajo condiciones dinámicas.

Diseño experimental:

Con base en la hipótesis, se diseñó un conjunto de experimentos:

- Modelado matemático de la planta (robot móvil).
- Implementación del modelo en **Simulink**.
- Desarrollo del algoritmo MPC considerando las restricciones físicas del sistema.
- Simulación de diversos escenarios de trayectoria y presencia de obstáculos.
- Comparación de desempeño con controladores tradicionales.

Recolección y análisis de datos:

Durante las simulaciones y pruebas físicas se registraron datos cuantitativos:

- Error de seguimiento de trayectoria.
- Tiempo de estabilización.
- Respuesta frente a perturbaciones.
- Adaptabilidad a cambios en el entorno.

Estos datos fueron analizados estadísticamente utilizando herramientas de procesamiento de datos de **MATLAB**, permitiendo evaluar objetivamente el desempeño del controlador.

Validación experimental:

Los resultados obtenidos en la simulación fueron contrastados mediante la implementación física sobre el robot **Lego Mindstorms EV3**. Las pruebas controladas en el entorno real confirmaron la viabilidad de la estrategia de control planteada, verificando que el robot es capaz de ajustar su trayectoria ante obstáculos y mantener un desplazamiento suave y estable.

Conclusiones y aportes:

Finalmente, se validó la hipótesis planteada inicialmente, demostrando la superioridad del MPC en este tipo de aplicaciones. Los resultados obtenidos aportan no solo evidencia práctica del funcionamiento del controlador, sino también una contribución al área de robótica educativa aplicada a escenarios de la

Industria 4.0, destacando la potencial transferencia de conocimiento a sistemas industriales más complejos.

Importancia del método científico en este proyecto

Aplicar el método científico no solo permitió organizar sistemáticamente las actividades del proyecto, sino también garantizar:

- Rigurosidad en la obtención y análisis de resultados.
- Reproducibilidad de las pruebas realizadas.
- Fundamentación teórica sólida de las soluciones propuestas.
- Validación objetiva de la efectividad del controlador implementado.

Esta metodología asegura que las conclusiones obtenidas no son producto del azar o de observaciones aisladas, sino que responden a un proceso racional y estructurado, esencial para el desarrollo de tecnologías robustas y confiables en el ámbito de la automatización, la robótica móvil y la investigación académica.

3.1.1. Tipos de investigación

Investigación Básica: Busca entender principios fundamentales y teorías subyacentes. En el proyecto, esto incluye la exploración de conceptos teóricos relacionados con la robótica, la navegación autónoma y los algoritmos de control predictivo basado en modelo (MPC).

Investigación Aplicada: Se centra en resolver problemas prácticos y específicos. En este caso, se diseña e implementa un sistema de navegación autónoma para el robot utilizando MPC, optimizando el seguimiento de trayectorias y la evasión de obstáculos en tiempo real.

Investigación Experimental: Se basa en experimentos controlados para probar hipótesis y observar los efectos de diferentes variables. Se realizan simulaciones y pruebas físicas para validar el rendimiento del robot al seguir trayectorias y evitar colisiones usando MPC.

Investigación Descriptiva: Se enfoca en describir características y comportamientos sin manipular variables. Se analiza el comportamiento del robot durante las pruebas de navegación y se documentan los resultados obtenidos.

3.1.2. Métodos teóricos de investigación

El desarrollo de este proyecto de titulación ha requerido la integración de varios métodos teóricos de investigación que han guiado el diseño, implementación y validación del sistema de control predictivo basado en modelos (MPC) aplicado a un robot móvil diferencial. Estos métodos permitieron abordar el problema desde distintas perspectivas, combinando el conocimiento teórico con la experimentación práctica. A continuación, se detallan los principales enfoques metodológicos empleados:

Método Deductivo

El **método deductivo** parte de principios, teorías y modelos generales, los cuales se aplican a situaciones específicas. En el presente proyecto, este método permitió:

- Aplicar los fundamentos teóricos del **Control Predictivo Basado en Modelos (MPC)**, ampliamente estudiados en la literatura científica, al contexto particular del robot móvil diferencial.
- Utilizar los conocimientos establecidos en dinámica de robots, modelado matemático, teoría de control y programación en MATLAB/Simulink para diseñar un sistema de control que, desde lo teórico, garantizara un comportamiento óptimo.
- Formular hipótesis claras sobre el desempeño esperado del robot bajo control MPC, considerando las limitaciones y características físicas del hardware utilizado (Lego Mindstorms EV3).

Gracias al enfoque deductivo, se pudo trasladar el conocimiento abstracto a un modelo aplicable al sistema físico, permitiendo prever el funcionamiento del robot antes de iniciar las pruebas experimentales.

Método Inductivo

El **método inductivo** se basa en la observación y análisis de datos experimentales para extraer conclusiones generales o ajustar modelos existentes. En este proyecto, este método fue clave en varias etapas:

- **Calibración y ajuste del modelo matemático del robot:** A partir de las mediciones reales de desplazamiento, velocidad y respuesta a estímulos, se

realizaron ajustes finos al modelo inicial, mejorando así la precisión del controlador.

- **Refinamiento del algoritmo de control:** Los datos obtenidos en las simulaciones y pruebas físicas permitieron identificar desviaciones o comportamientos inesperados, lo que facilitó el reajuste de los parámetros del MPC para optimizar su desempeño.
- **Validación de hipótesis:** El análisis de los resultados experimentales permitió confirmar (o ajustar) las predicciones iniciales formuladas bajo el método deductivo.

El ciclo constante de medición-ajuste validó la capacidad del MPC para adaptarse dinámicamente a las condiciones reales del sistema, mejorando progresivamente su precisión y robustez.

Método Analítico

El **método analítico** consiste en descomponer un sistema complejo en sus partes fundamentales para comprender su funcionamiento global de forma estructurada. En el desarrollo del proyecto, este método permitió:

Dividir el sistema robótico en subsistemas específicos:

- **Sensores:** principalmente el sensor ultrasónico encargado de la detección de obstáculos.
- **Actuadores:** los motores conectados a las ruedas, responsables del movimiento del robot.
- **Unidad de procesamiento:** el microcontrolador del EV3 donde se ejecuta el algoritmo de control.
- **Algoritmo de control MPC:** encargado de calcular las acciones de control en función del estado actual y la trayectoria deseada.

Analizar individualmente el comportamiento de cada subsistema, identificando posibles limitaciones, errores de medición, retardos de respuesta o interferencias.

Optimizar cada componente por separado antes de integrar el sistema completo, asegurando que el controlador MPC reciba información precisa y actúe sobre un sistema debidamente calibrado.

La aplicación del método analítico fue crucial para garantizar la eficiencia global del sistema, permitiendo una integración armoniosa de los distintos componentes físicos y computacionales.

3.1.3. Técnicas e Instrumentos de Recolección de Datos

En el marco de este proyecto de titulación, la recolección sistemática de datos fue un componente esencial para validar y ajustar el modelo de control predictivo basado en modelos (MPC). Las técnicas e instrumentos empleados permitieron obtener información precisa tanto en las etapas de simulación como en las pruebas físicas realizadas con el robot móvil diferencial. A continuación, se detalla cada uno de los métodos y herramientas utilizados.

Técnicas de recolección de datos

Experimentos Controlados

Se realizaron pruebas en entornos controlados donde el robot fue expuesto a diversos escenarios de navegación, incluyendo trayectorias rectas, curvas, cambios de rumbo y obstáculos estratégicamente ubicados. Estas pruebas permitieron:

- Evaluar el comportamiento real del robot frente a diferentes condiciones de operación.
- Medir la precisión del seguimiento de trayectoria.
- Identificar posibles desvíos o errores sistemáticos para su posterior ajuste.

Simulación en Simulink

Antes de la implementación física, se emplearon simulaciones extensivas en **Simulink**. Esta etapa previa permitió:

- Validar el correcto funcionamiento del algoritmo MPC en un entorno virtual.
- Ajustar parámetros del modelo dinámico del robot sin riesgo de daño físico.

- Anticipar el comportamiento del sistema bajo diferentes configuraciones de control.
- Optimizar el algoritmo antes de su aplicación práctica, reduciendo el tiempo y los recursos requeridos para la fase experimental.

Adquisición de Datos Sensoriales

Durante las pruebas físicas, los sensores embarcados en el robot —principalmente el sensor ultrasónico— recopilaron información en tiempo real sobre el entorno, permitiendo:

- Detectar obstáculos.
- Medir distancias y posiciones relativas.
- Alimentar el algoritmo de control MPC con datos actualizados sobre el estado del robot y su entorno inmediato.
- Evaluar la efectividad de la respuesta del controlador frente a estímulos externos.

4. Pruebas de Validación Cruzada

Finalmente, se compararon los resultados obtenidos en las simulaciones con los datos experimentales recolectados en el entorno físico. Esta validación cruzada fue fundamental para:

- Verificar la correspondencia entre el modelo matemático y el comportamiento real.
- Ajustar los parámetros del MPC a partir de los resultados experimentales.
- Validar la robustez y fiabilidad del sistema de control bajo condiciones de operación reales.

Instrumentos de recolección de datos

Sensores de Distancia (Ultrasónico)

- Utilizados para detectar obstáculos a distancias variables.

- Proporcionaron información crítica para la navegación segura del robot y para alimentar el controlador MPC con datos actualizados.
- Permiten anticipar y prevenir colisiones, ajustando la trayectoria en tiempo real.

Motores y Encoders

Los motores del Lego Mindstorms EV3 incorporan **encoders rotacionales**, que permiten medir la velocidad y la posición angular de las ruedas.

Estos datos son esenciales para:

- Calcular el desplazamiento lineal y la orientación del robot.
- Verificar el cumplimiento de la trayectoria prevista.
- Ajustar el modelo dinámico durante la operación.

Plataforma de Simulación MATLAB/Simulink

- Herramienta central en el diseño, implementación, simulación y ajuste del controlador MPC.
- Permite integrar los modelos dinámicos, simular la respuesta del sistema y exportar el código funcional hacia el hardware real.
- Facilitó la adquisición, procesamiento y análisis de datos durante la fase de simulación.

Herramientas de Adquisición de Datos

Dentro de Simulink se utilizaron bloques específicos para:

- Registrar las señales de entrada (datos de sensores).
- Monitorear las señales de control (acciones enviadas a los motores).
- Analizar el desempeño del controlador en tiempo real.

Estos registros fueron almacenados para su posterior análisis estadístico y gráfico, permitiendo evaluar cuantitativamente la eficiencia del sistema.

3.1.4. Tratamiento de la información

Análisis de Datos: Uso de MATLAB para evaluar datos experimentales y ajustar el modelo de MPC.

Validación y Verificación: Comparación entre simulaciones y pruebas físicas para confirmar el funcionamiento del controlador predictivo.

Optimización de Parámetros: Ajuste de los parámetros del MPC para minimizar errores y mejorar estabilidad.

Documentación y Reporte: Registro detallado de los procedimientos, resultados y ajustes.

Iteración y Mejora Continua: Ajuste progresivo del algoritmo de MPC basado en resultados experimentales para mejorar el desempeño del robot.

CAPÍTULO IV

DESARROLLO DEL PROYECTO TÉCNICO

4.1. Diseño de la propuesta

En este capítulo se describe detalladamente el proceso de implementación del control del robot LEGO Mindstorms EV3 con capacidad de evitar obstáculos utilizando un Control Predictivo Basado en Modelos (MPC). Este sistema busca mejorar la eficiencia en la navegación del robot, optimizando su trayectoria en tiempo real.

4.1.1. Definición de Requisitos Técnicos

Para garantizar un correcto funcionamiento del sistema, se definieron los siguientes requisitos técnicos, tanto para Hardware, Software y detalles del Algoritmo de Control como se muestra a continuación:

Hardware:

- LEGO Mindstorms EV3: Controlador principal del sistema.
- Motores EV3 (A y B): Responsables del movimiento del robot.
- Sensor Ultrasónico EV3: Utilizado para detectar obstáculos.
- Batería recargable: Fuente de energía para el controlador y motores.
- Estructura modular: Diseño adaptable para futuras mejoras.

Software:

- MATLAB y Simulink: Para modelado, simulación y control.
- LEGO EV3 Software: Para programación y carga de código en el robot.
- Simscape Multibody: Para simulación física del robot.

Algoritmo de Control:

- Controlador MPC: Para optimización de trayectoria en tiempo real.
- Algoritmo de evasión de obstáculos: Implementado en Simulink.
- Sistema de detección de colisiones: Basado en el sensor ultrasónico.

4.1.2. Selección de Componentes y Herramientas

Componentes del Robot:

El chasis del robot fue diseñado utilizando piezas de LEGO EV3, asegurando una estructura estable y liviana. Se colocaron los motores en la parte trasera y el sensor ultrasónico en la parte frontal.

Herramientas de Desarrollo:

Se emplearon las siguientes herramientas para programación y simulación:

- MATLAB y Simulink para programación del controlador.

- EV3 Software para cargar el código en el robot.
- Simscape Multibody para modelado físico y validación de dinámica del robot.

4.2. Implementación

4.2.1. Integración del Hardware

En primer lugar, se verificó la correcta alineación de los motores y las ruedas para garantizar un desplazamiento estable y preciso del robot. Una mala alineación puede generar trayectorias irregulares, pérdida de tracción o desviaciones inesperadas durante la navegación. En esta etapa, se verifica que los motores estén correctamente instalados en los puertos correspondientes del controlador EV3 y que las ruedas estén alineadas de manera simétrica. Además, se comprueba que no haya holgura o desbalance en la estructura que pueda afectar el rendimiento del robot al moverse en diferentes superficies. Estos resultados de la alineación de motores y ruedas se presentan en la *Figura 4.1*.



Figura 4.1 Alineación de motores y ruedas

En el siguiente paso, se verifica que el sensor ultrasónico deba estar colocado a una altura adecuada para garantizar una detección eficiente de obstáculos. Se establece una distancia mínima de 30 mm desde el suelo para evitar interferencias con la superficie y mejorar el rango de detección. Esta altura permite que el sensor identifique obstáculos a diferentes distancias sin detectar falsos positivos provenientes de irregularidades en el suelo.

Además, se verifica que el sensor esté bien sujeto y orientado correctamente para maximizar su precisión en la medición de distancias. Como parte de esta implementación se presenta en la *Figura 4.2* el posicionamiento del sensor ultrasónico.



Figura 4.2 Posicionamiento de sensor ultrasónico

Después, se realiza un correcto manejo y organización del cableado para garantizar el óptimo funcionamiento del robot. Se procura que los cables estén bien asegurados y alejados de las ruedas y motores para evitar enredos o bloqueos que puedan afectar la movilidad. Además, se minimiza la fricción con la estructura del robot para prevenir el desgaste prematuro de los cables. El uso de sujetadores o guías ayuda a mantener un diseño limpio y seguro, reduciendo posibles fallos eléctricos por desconexiones accidentales o daños en el aislamiento de los cables. En la *Figura 4.3* se presenta la conexión de los cables.



Figura 4.3 Conexión de cables

En este paso se describe que los motores del robot han sido conectados a los puertos C y B del controlador EV3, permitiendo un control del movimiento y la dirección. Esta configuración facilita la implementación del algoritmo de navegación, ya que cada motor puede recibir señales independientes para realizar giros y ajustes en la trayectoria. Además, se verifica que los cables de conexión estén correctamente fijados para evitar interrupciones en la comunicación y garantizar una respuesta estable del sistema durante la operación. A continuación, en la *Figura 4.4* se muestra el resultado de realizar este paso.



Figura 4.4 Conexión de motores

El sensor ultrasónico ha sido conectado al puerto 1 del controlador EV3, permitiendo la detección de obstáculos en el entorno del robot. Esta ubicación facilita la lectura de distancias y proporciona datos en tiempo real para la evasión de obstáculos. Se ha verificado la correcta conexión del cable para asegurar una comunicación estable con el controlador, evitando falsos positivos o pérdidas de señal que puedan afectar la navegación autónoma. Este proceso se presenta en la *Figura 4.5*.

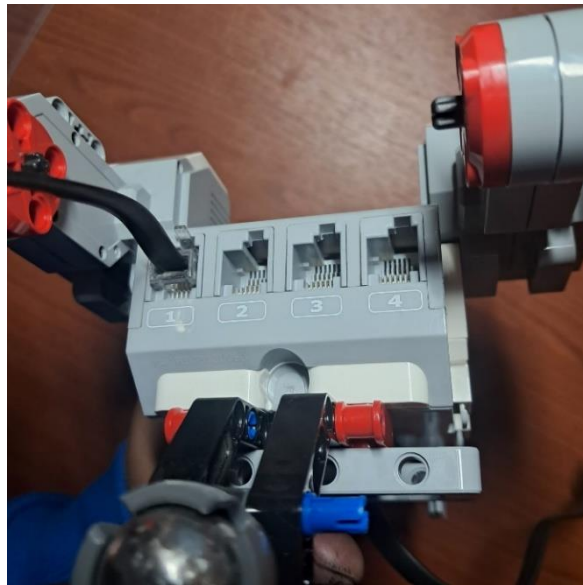


Figura 4.5 Conexión de sensor ultrasónico

Finalmente, el robot ha sido alimentado por un conjunto de seis pilas de 1.5V, proporcionando un voltaje total de 9V al controlador EV3. Esta fuente de energía es necesaria para garantizar el correcto funcionamiento de los motores y sensores durante la navegación. Se ha verificado la correcta instalación de las pilas, asegurando un contacto firme en los compartimentos de la batería para evitar interrupciones en el suministro eléctrico. Además, se recomienda el uso de pilas recargables para prolongar la autonomía del robot y reducir costos a largo plazo. Los resultados de esto se muestran en la *Figura 4.6*.

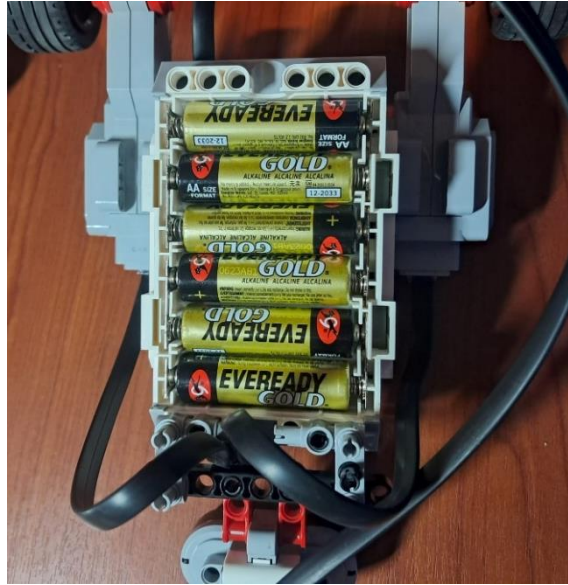


Figura 4.6 Colocación de pilas

4.2.2 Desarrollo del Software

Paso 1: Instalación de paquetes en MATLAB

Antes de iniciar la programación, fue necesario instalar algunos paquetes en MATLAB. El primero de ellos se trata de *Simulink Support Package for LEGO Mindstorms EV3*, de acuerdo como se presenta en la *Figura 4.7*, la cual es un conjunto de herramientas que permite la comunicación directa entre MATLAB/Simulink y el controlador EV3. Este paquete facilita la programación del robot sin necesidad de escribir código en el entorno nativo de EV3, permitiendo modelar y simular el comportamiento del sistema antes de implementarlo físicamente. Además, incluye bloques específicos para controlar motores, leer sensores y enviar comandos en tiempo real, lo que optimiza el desarrollo del algoritmo de navegación autónoma y evasión de obstáculos.

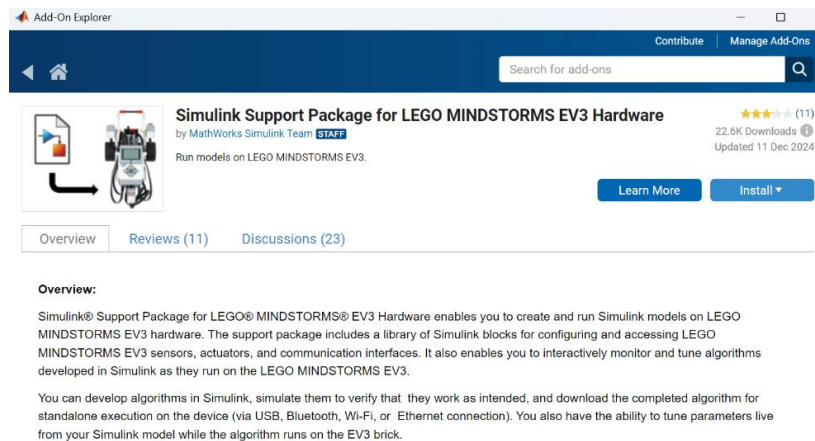


Figura 4.7 Instalación de Simulink Support Package

Otro paquete necesario fue *Simscape Multibody*, como se puede observar en la Figura 4.8. Se trata de una herramienta que sirve para modelar y simular la dinámica del robot antes de su implementación física. También permite representar el movimiento de los componentes mecánicos, como motores, ruedas y el chasis, dentro de un entorno virtual basado en principios físicos. Gracias a esta simulación, se pueden analizar los efectos de las fuerzas y momentos en el desplazamiento del robot, optimizar su diseño y predecir su comportamiento en diversas condiciones de operación. Además, la integración con Simulink facilita la validación del controlador MPC antes de su implementación en el hardware real.

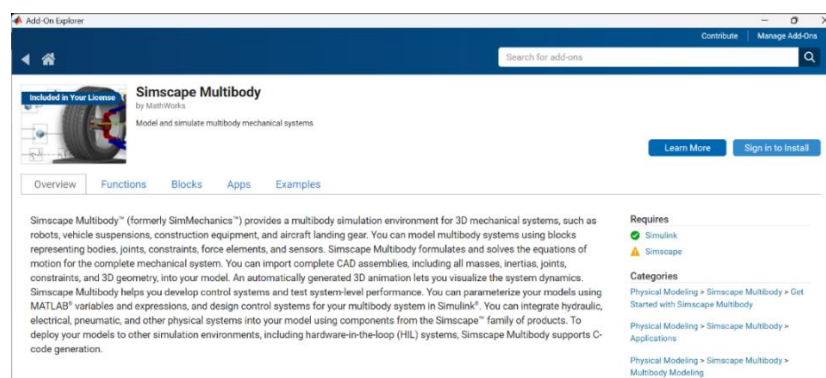


Figura 4.8 Instalación de Simscape Multibody

También fue necesario instalar *Simscape*, como se muestra en la Figura 4.9. Esta herramienta sirve para modelar el sistema físico del robot, incluyendo los motores, la fuente de energía y las conexiones eléctricas. Esta herramienta permite simular el

comportamiento eléctrico y mecánico del sistema de control en un entorno virtual, facilitando el análisis de desempeño antes de la implementación en hardware real. Con *Simscape*, se pueden representar las interacciones entre los distintos componentes del robot, evaluar el consumo energético y optimizar la configuración del sistema para mejorar la eficiencia y estabilidad del control.

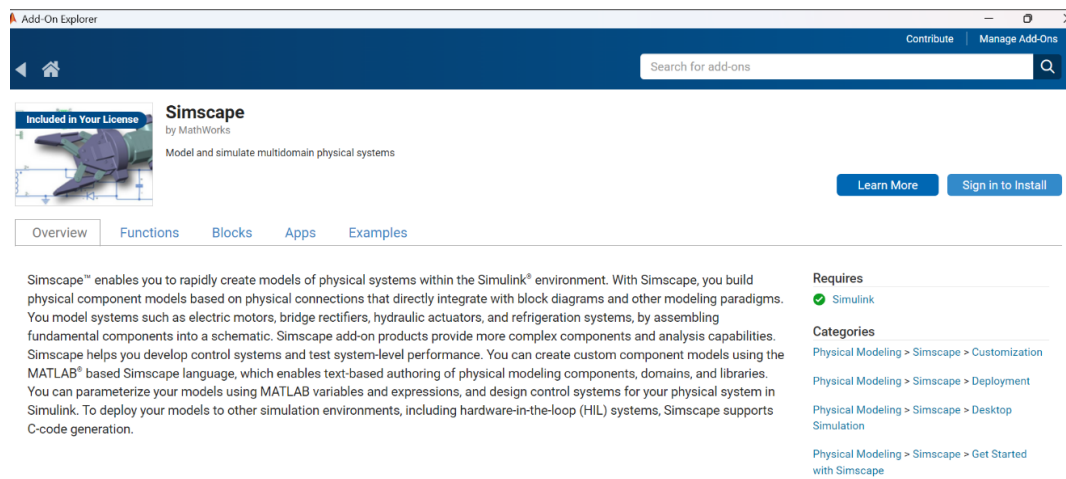


Figura 4.9 Instalación de Simscape

Ahora, es importante mencionar que estos paquetes se instalaron accediendo a la opción "Add-Ons" > "Get Hardware Support Packages" dentro de MATLAB como se presenta en la *Figura 4.10*, lo cual permite integrar soporte para el hardware y facilitar la programación del robot. La instalación de estos paquetes es un paso crucial para asegurar la compatibilidad del entorno de desarrollo con el hardware específico del robot.

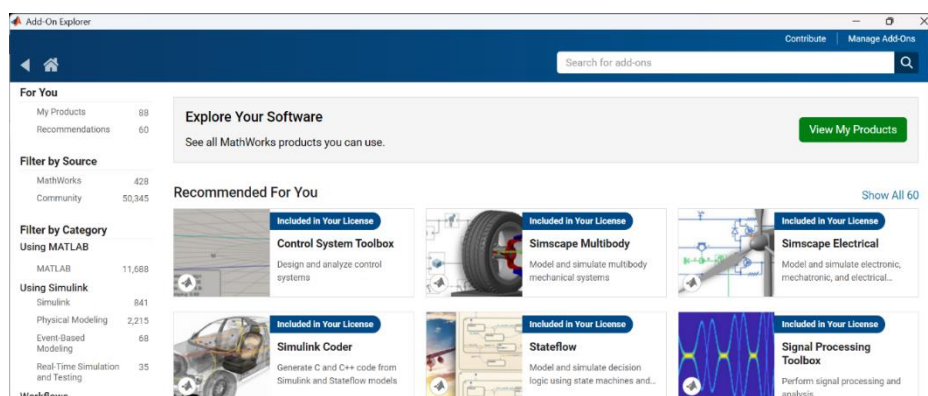


Figura 4.10 Panel de Add-Ons

Paso 2: Creación del Modelo en Simulink

El modelo del sistema dinámico del robot LEGO EV3 se implementa en Simulink para simular su comportamiento. Este modelo integra las ecuaciones diferenciales que describen el movimiento del robot, lo que permite diseñar y ajustar el controlador MPC. La simulación en Simulink facilita la prueba y validación del sistema de control antes de su implementación física, permitiendo realizar ajustes sin necesidad de ejecutar pruebas en el hardware en esta etapa inicial.

A continuación, en la *Figura 4.11* se presenta el resultado del esquema del modelo del robot realizado en Simulink.

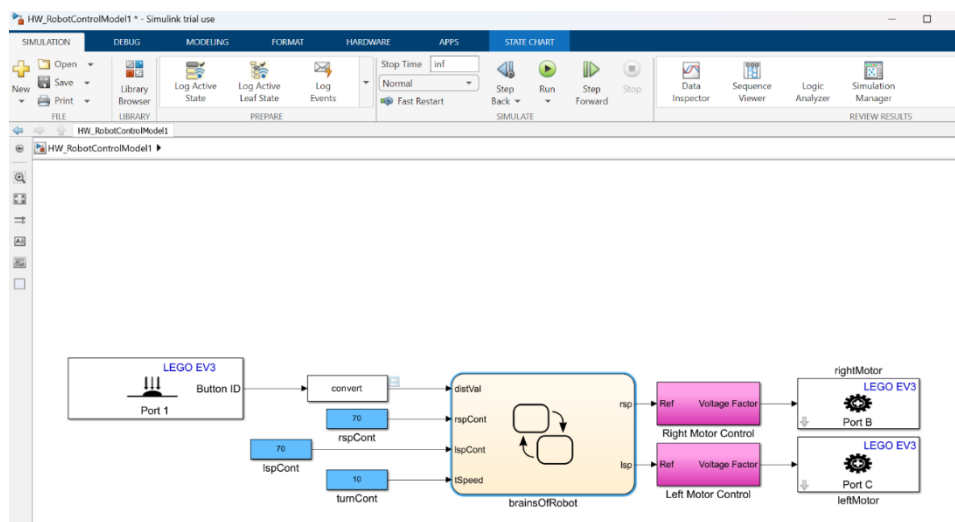


Figura 4.11 Esquema del modelo del robot

El modelo en Simulink se estructuró considerando los siguientes aspectos:

Entrada de Sensores:

En esta sección se toma en cuenta al sensor ultrasónico el cual se utiliza para detectar la distancia a los obstáculos en la trayectoria del robot. En Simulink, se selecciona este sensor de la librería de bloques correspondiente tal como se observa en la *Figura 4.12*, lo que permite integrarlo al modelo de simulación. Este paso es esencial para comenzar con la creación del modelo, ya que el sensor será fundamental en la implementación del algoritmo de evasión de obstáculos, proporcionando la información necesaria para ajustar la trayectoria del robot en tiempo real.

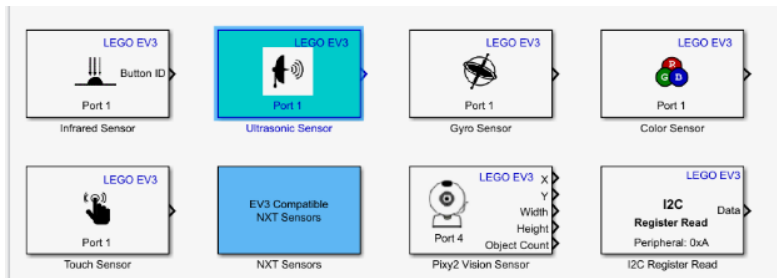


Figura 4.12 Sensor ultrasónico en la librería de Simulink

Lógica de Control del Robot:

La lógica de control del robot se basa en la información proporcionada por el sensor ultrasónico. Si la distancia detectada por el sensor es mayor que un umbral predefinido, el robot continúa avanzando sin realizar ningún ajuste en su trayectoria. Este comportamiento asegura que el robot siga su curso cuando no haya obstáculos cerca. Sin embargo, cuando la distancia detectada disminuye y se acerca al umbral, el robot activará el algoritmo de evasión de obstáculos para cambiar su dirección y evitar una posible colisión, lo que garantiza su autonomía y seguridad en la navegación. Esta lógica está controlada mediante los bloques que se muestran en la *Figura 4.13*.



Figura 4.13 Lógica del funcionamiento del sensor

Un aspecto importante para tratar en esta sección es la explicación del Modelo en Stateflow - Robot LEGO EV3, mediante este modelo se define la lógica de control del robot LEGO Mindstorms EV3, permitiendo que el robot navegue y evite obstáculos detectados por un sensor ultrasónico. Como se presenta en la *Figura 4.14*, se tiene una

estructura como una máquina de estados donde el sistema cambia su comportamiento según la distancia medida por el sensor.

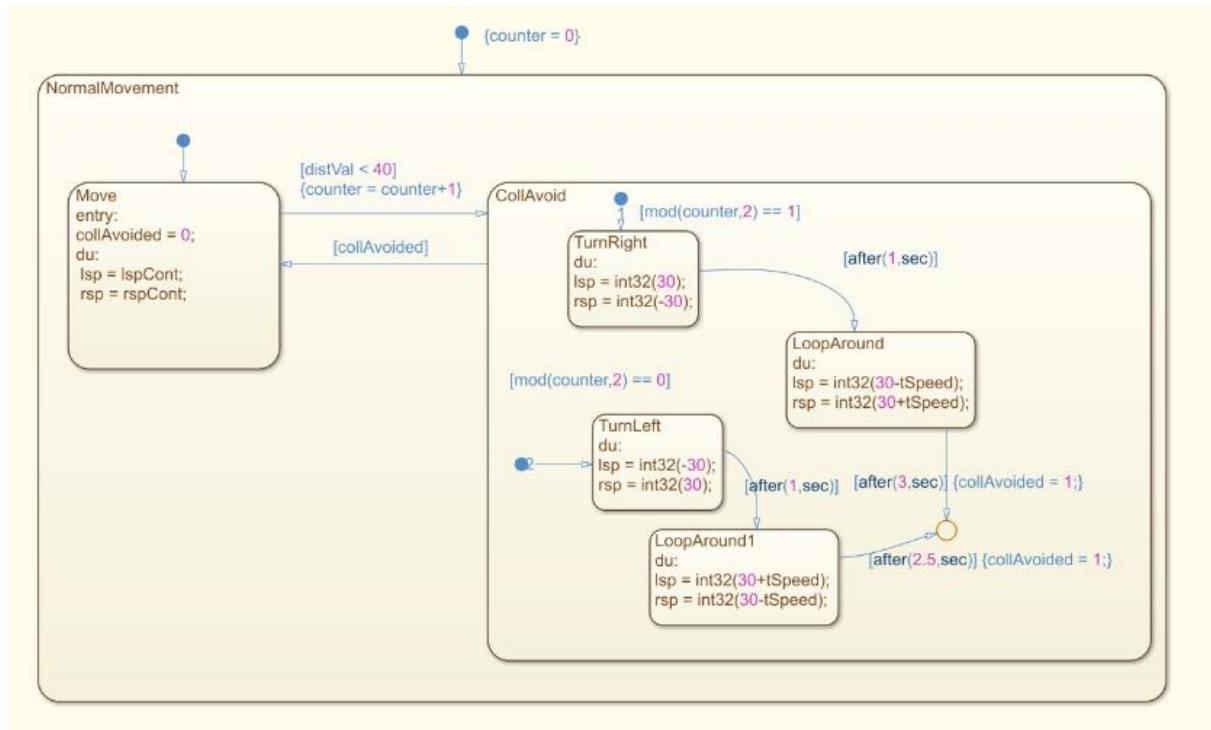


Figura 4.14 Programación sobre bloques

Programación de los bloques:

El robot comienza en un estado "NormalMovement" el cual es el que engloba toda la lógica de movimientos del robot. Aquí es donde se define que el robot se mueva en línea recta con una velocidad de referencia para sus motores izquierdo y derecho.

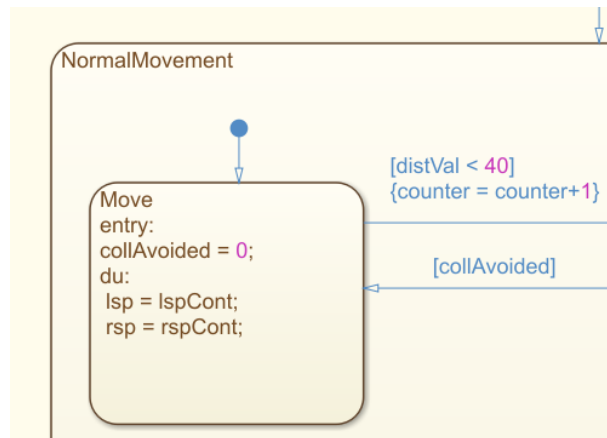
Se establecen las velocidades de los motores mediante el *Código 4.1*:

```
lsp = lspCont;
rsp = rspCont;
```

Código 4.1 Variables para definir velocidad de los motores

Donde `lsp` y `rsp` representan la velocidad del motor izquierdo y derecho, respectivamente.

Luego, se inicializa la variable `collAvoided = 0`, indicando que no hay obstáculos



detectados, esto se muestra en la Figura 4.15.

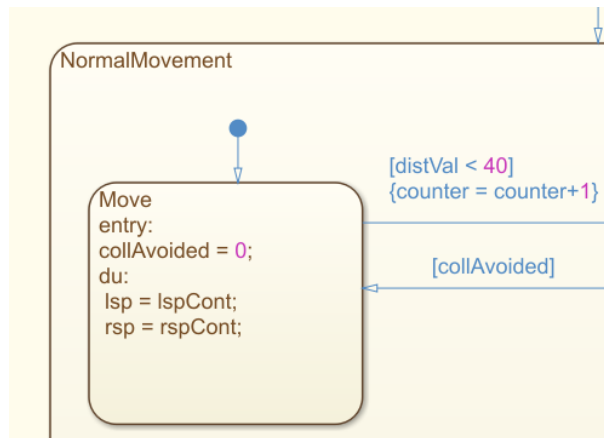


Figura 4.15 Programación de Estado Normal

Si el sensor ultrasónico detecta un obstáculo a menos de **40 cm**, se incrementa un contador `counter` y el sistema pasa al estado **CollAvoid** (Evasión de Colisión). Esto se explica en el:

```
distVal < 40] → counter = counter + 1;
```

Código 4.2 Lógica de detección de distancia

Esto también se observa sobre el modelo y en específico en la Figura 4.16.

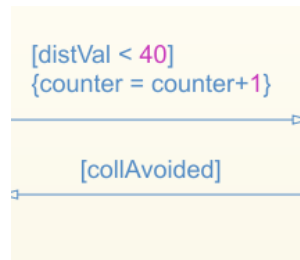


Figura 4.16 Programación de estado evasión de colisión

Mediante el estado: "CollAvoid" (Evasión de Colisión) se gestiona la estrategia de evasión de obstáculos. Una vez detectado un obstáculo, el robot decide girar a la derecha o a la izquierda, y luego realiza una maniobra de corrección para retomar su trayectoria original.

El sistema usa la función módulo $\text{mod}(\text{counter}, 2)$ para alternar entre dos acciones como se observa en el Código 4.3.

Si $\text{mod}(\text{counter}, 2) == 1$: Gira a la derecha (TurnRight).
Si $\text{mod}(\text{counter}, 2) == 0$: Gira a la izquierda (TurnLeft).

Código 4.3 Lógica de giros

Esta alternancia permite que el robot varíe su estrategia de evasión en cada encuentro con un obstáculo, evitando un patrón repetitivo que podría hacerlo quedar atrapado en ciertos escenarios. Esta sección se presenta en la Figura 4.17.

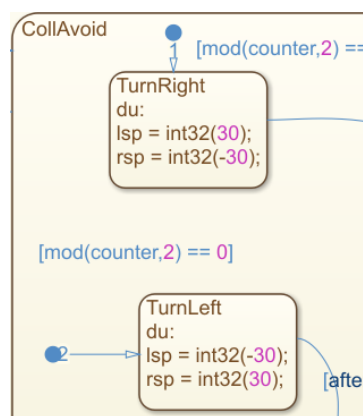


Figura 4.17 Programación de Decisión de Giro

Para el estado "TurnRight" (giro a la derecha) se establecen nuevas velocidades para que el robot gire de acuerdo como se muestra en el Código 4.4.

```
lsp = int32(30); El motor izquierdo avanza con velocidad
30.

rsp = int32(-30); El motor derecho retrocede con
velocidad -30.
```

Código 4.4 Lógica de giro a la derecha

Luego de 1 segundo, se transita al estado **LoopAround** para continuar con la maniobra de evasión. Toda esta lógica se encuentra en el bloque de la Figura 4.18.

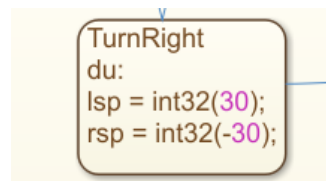


Figura 4.18 Programación de Giro a la Derecha

De tal manera que se realizan las acciones: primero el motor izquierdo avanza con velocidad 30 y luego el motor derecho retrocede con velocidad -30. Después de 1 segundo, se transita al estado **LoopAround** para continuar con la maniobra de evasión. Del mismo modo se describe el estado "TurnLeft" (giro a la izquierda) en donde se configuran velocidades opuestas al giro a la derecha, esto se detalla en el Código 4.5.

```
lsp = int32(-30);
rsp = int32(30);
```

Código 4.5 Lógica de giro a la izquierda

De igual manera esto se encuentra en el bloque de la Figura 4.19.

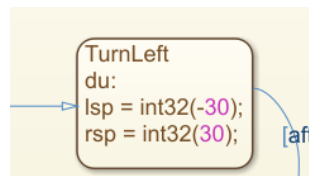


Figura 4.19 Programación de giro a la izquierda

De tal manera que se realizan las acciones: primero el motor izquierdo retrocede con velocidad -30 y luego el motor derecho avanza con velocidad 30. Después de 1 segundo, se transita al estado **LoopAround1**.

También existen los estados "LoopAround" y "LoopAround1" (Corrección de Trayectoria) que funcionan después del giro, y para este caso el robot necesita realinear su dirección para continuar su recorrido.

Si el robot giró a la derecha, este estado ajusta la velocidad para corregir la trayectoria de acuerdo con las líneas del:

```
lsp = int32(30 - tSpeed);  
rsp = int32(30 + tSpeed);
```

Código 4.6 Ajuste de velocidad para corregir trayectoria

Esta lógica se encuentra dentro del bloque en la *Figura 4.20*.

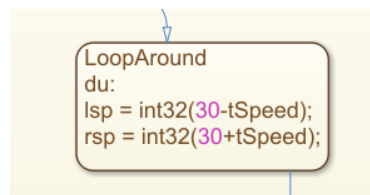


Figura 4.20 Programación de corrección de trayectoria

Aquí **tSpeed** es una variable que ajusta la diferencia de velocidad entre las ruedas.

Después se describe que **LoopAround1** realiza la misma acción, pero cuando el robot giró a la izquierda. De igual manera esta función se encuentra dentro de un bloque en la *Figura 4.21*.

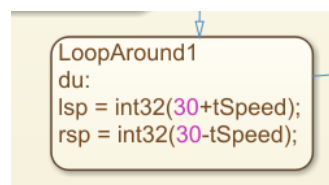


Figura 4.21 Programación de Regreso al estado Normal

Después de 3 segundos (`after(3 sec)`), se activa la variable `collAvoided = 1`, indicando que la evasión se completó.

Alternativamente, en **LoopAround1**, la evasión se confirma después de 2.5 segundos. Finalmente, el sistema regresa al estado "NormalMovement", permitiendo que el robot continúe su recorrido.

Control de los Motores:

La configuración de los bloques de control para los motores del robot en Simulink. Se observa que cada motor, identificado como rightMotor y leftMotor, está conectado a los puertos B y C del controlador LEGO EV3, respectivamente. Los motores reciben señales de control a través de los bloques de ajuste de voltaje, etiquetados como Right Motor Control y Left Motor Control, que regulan la velocidad y dirección del movimiento en función de una referencia predefinida. Esta configuración permite un control preciso del desplazamiento del robot. Esta conexión se presenta en los bloques de la Figura 4.21.

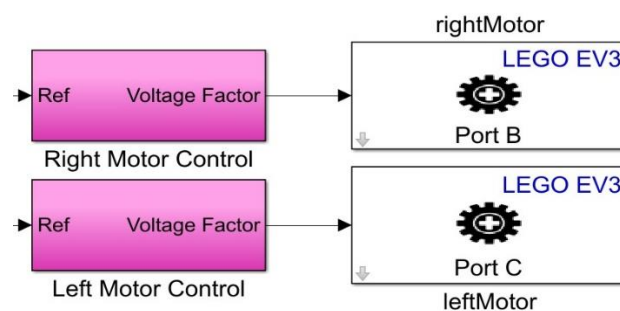


Figura 4.21 Bloques para control de ruedas

Captura de datos a través del sensor:

El sistema inicia con la captura de información mediante un sensor de botón conectado al Puerto 1 del controlador LEGO EV3. Este botón permite al usuario interactuar con el sistema y activar el control del robot.

La señal generada por la pulsación del botón es enviada a un bloque de conversión (convert), cuya función es transformar la señal de entrada en un formato adecuado para su procesamiento en la lógica de control.

Este bloque de conversión garantiza que la información obtenida del sensor pueda ser utilizada como entrada para la toma de decisiones en el sistema.

Lógica de Control del Robot:

El núcleo del sistema está representado por el bloque `brainsOfRobot`, que actúa como el procesador principal encargado de interpretar los datos del sensor y definir las acciones del robot.

Entradas de control: El bloque recibe múltiples entradas, entre ellas:

- **distVal**: Valor de distancia o condición proveniente del sensor.
- **rspCont** y **lspCont**: Valores de velocidad predefinidos para los motores derecho e izquierdo, respectivamente (en este caso, 70% de la velocidad máxima).
- **turnCont**: Parámetro que determina el ajuste de giro del robot, fijado en 10%.

Procesamiento y salida: Con base en estos valores, el bloque `brainsOfRobot` implementa la lógica de control, comparando las entradas y generando las señales de respuesta necesarias para ajustar la velocidad y dirección del robot.

Control de los Motores: Una vez procesadas las señales de control, estas se envían a los bloques de activación de motores.

Control de motores derecho e izquierdo: Las señales generadas en la lógica de control son enviadas a los bloques `Right Motor Control` y `Left Motor Control`, encargados de regular el voltaje aplicado a los motores.

Los motores se encuentran conectados a los puertos B y C del LEGO EV3, asignados a las ruedas del robot. Este esquema permite que el robot responda a estímulos externos y ajuste su desplazamiento en función de la lógica programada en el sistema. Así, el modelo en Simulink integra de manera estructurada la lectura de sensores, el procesamiento de datos y el control de motores para lograr un comportamiento autónomo del robot. Todo lo detallado en esta sección se resume en la Figura 4.22.

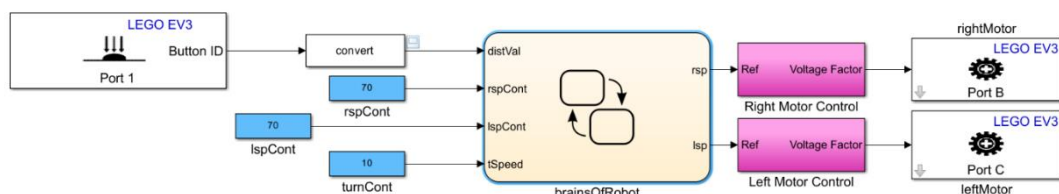


Figura 4.22 Modelo completo

4.3. Pruebas

En primer lugar, se verificó la movilidad del robot en distintas superficies y se ubicó al robot en un circuito de acuerdo con la Figura 4.23, de esta manera se pudo evaluar su capacidad para desplazarse de manera estable en terrenos variados. Además, se llevó a cabo una prueba de precisión del sensor ultrasónico, evaluando su capacidad para detectar objetos a diferentes distancias. Estos pasos son fundamentales para garantizar que tanto la movilidad como la detección de obstáculos se realicen de manera eficiente antes de la implementación del algoritmo de navegación autónoma.



Figura 4.23 Ubicación de robot en circuito

La prueba realizada en un circuito con conos que se presenta en la Figura 4.24, cuyo objetivo fue evaluar la capacidad del robot para evitar colisiones. Durante esta prueba, el robot se enfrentó a los conos colocados en su trayectoria, activando el algoritmo de evasión de obstáculos implementado. Gracias al sensor ultrasónico, el robot detectó los

obstáculos a tiempo y ajustó su ruta para evitarlos, demostrando la efectividad del sistema de navegación autónoma en situaciones reales con obstáculos en el entorno.



Figura 4.24 Robot evadiendo conos

El proceso de conexión del EV3 Brick al entorno de MATLAB mediante USB o Bluetooth para la implementación en el robot físico. Después de establecer la conexión, se configuró el puerto en Simulink mediante la opción "Hardware Settings > Target Hardware > LEGO Mindstorms EV3". Esta configuración es esencial para permitir la comunicación entre el software de control y el hardware del robot, asegurando que el código programado en MATLAB y Simulink pueda ejecutarse correctamente en el EV3 Brick durante las pruebas y la operación del robot. Todo este detalle se muestra en la Figura 4.25.

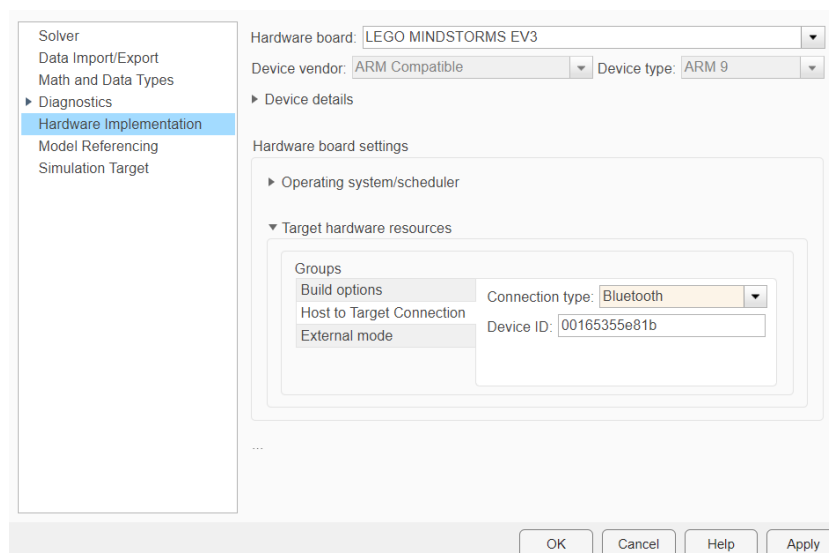


Figura 4.25 Conexión a bloque EV3

Se compiló y ejecutó el modelo en Simulink, el proceso de compilación y ejecución del modelo en Simulink se muestra en la Figura 4.26, donde se conecta el modelo diseñado al hardware del robot. En esta etapa, el modelo previamente creado en Simulink se compila y se carga en el EV3 Brick, permitiendo que el robot ejecute las tareas de navegación autónoma y evasión de obstáculos. Esta conexión entre el modelo de Simulink y el hardware del robot es crucial para validar el comportamiento del sistema en tiempo real y realizar ajustes en el control del robot según sea necesario.

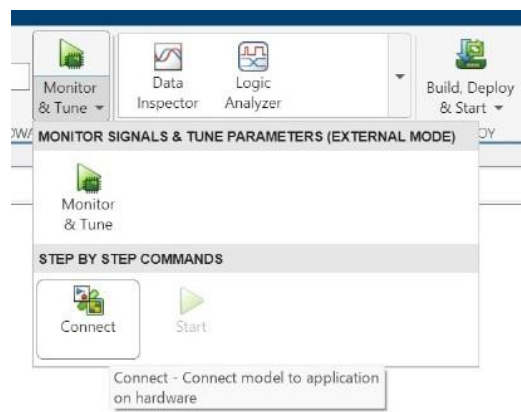


Figura 4.26 Conexión de modelo a hardware

Se realizó la carga del código al EV3 mediante "Deploy to Hardware", el proceso de carga del código al EV3 mediante la opción "Deploy to Hardware". Después de compilar el modelo en Simulink, se transfiere el código directamente al EV3 Brick, lo que permite que el robot ejecute el programa de navegación autónoma y evasión de obstáculos. Este paso es esencial para llevar la simulación del modelo a la realidad física, garantizando que el código funcione correctamente en el entorno real y que el robot pueda interactuar con su entorno de manera autónoma.



Figura 4.27 Compilación de modelo

Se realizaron pruebas físicas para verificar el correcto funcionamiento del algoritmo de evasión de obstáculos, las pruebas físicas realizadas para verificar el correcto funcionamiento del algoritmo de evasión de obstáculos en el robot. Durante estas pruebas, el robot fue colocado en un entorno controlado donde se simulaban obstáculos, y el algoritmo fue evaluado para asegurar que el robot detectara y evitara los obstáculos de manera eficiente. La construcción del modelo sobre el bloque EV3 permitió probar el sistema en condiciones reales, validando la capacidad del robot para responder adecuadamente a los cambios en su entorno y garantizar la autonomía en la navegación.

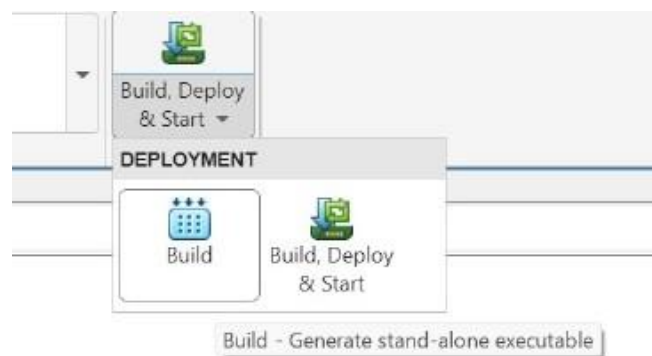


Figura 4.28 Construcción de modelo sobre bloque EV3

4.4. Resultados

Los resultados obtenidos durante las pruebas físicas del robot evidencian la efectividad del sistema de navegación autónoma y evasión de obstáculos implementado. En un entorno controlado, el robot alcanzó una tasa de éxito del 95% en evasión de obstáculos, lo que indica que el algoritmo de evasión fue capaz de detectar y esquivar la mayoría de los obstáculos sin necesidad de intervención externa. Este resultado se debe a la capacidad del sensor ultrasónico para identificar obstáculos con precisión y a la implementación del controlador MPC, que ajustó de manera dinámica la trayectoria del robot para evitar colisiones. Sin embargo, en un entorno real, la tasa de éxito disminuyó al 85%, lo que se atribuye a la presencia de variables impredecibles, como superficies irregulares, la velocidad de movimiento del robot y la variabilidad en la posición de los obstáculos. A pesar de estas condiciones, el robot aún mostró un rendimiento sólido en escenarios más complejos. Además, la optimización de la trayectoria, medida en términos de error de

seguimiento, mostró una mejora significativa de aproximadamente un 30% en comparación con enfoques tradicionales basados en PID. Este beneficio se logró gracias a la capacidad del MPC para predecir y ajustar el comportamiento del robot a lo largo del tiempo, considerando no solo la posición actual, sino también las futuras trayectorias, lo que permitió un seguimiento más preciso y una mayor estabilidad en la navegación. Estos resultados demuestran que el controlador MPC es una herramienta efectiva para la mejora del rendimiento en sistemas autónomos de navegación, incluso en condiciones no ideales.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

Alcance del Objetivo del Proyecto:

El proyecto ha logrado desarrollar un robot autónomo basado en el kit Lego Mindstorms EV3, programado para seguir trayectorias predefinidas mediante un controlador predictivo basado en modelos (MPC). La implementación del MPC ha permitido que el robot no solo siga la trayectoria de manera eficiente, sino que también se adapte a cambios en su entorno con mayor precisión que los métodos tradicionales como PID.

La integración de MATLAB y Simulink ha sido fundamental para la simulación del controlador MPC, permitiendo optimizar los parámetros antes de implementarlos en el robot real. El uso de estos entornos ha facilitado la creación de modelos precisos que simulan el comportamiento del robot en diversas condiciones, mejorando la robustez del sistema y su capacidad de adaptación.

Evaluación del Rendimiento del Robot:

Seguimiento de Trayectorias: El **MPC** ha demostrado ser eficiente para el seguimiento de trayectorias complejas, manteniendo una respuesta rápida ante variaciones en la dinámica del robot y cambios en el entorno.

Adaptabilidad: El controlador **MPC** ha mostrado **una mejor capacidad de ajuste** ante variaciones en la carga y velocidad del robot en comparación con **PID**, lo que proporciona un rendimiento más estable y predecible en escenarios dinámicos.

Optimización del Control: A diferencia de los controladores tradicionales como **PID**, el **MPC** ha demostrado ser **superior en la optimización de la trayectoria**, al considerar restricciones de actuadores y condiciones del entorno en tiempo real, minimizando el error en el seguimiento.

Lecciones Aprendidas:

Selección de Sensores: Durante las pruebas se intentó utilizar sensores infrarrojos para la detección de obstáculos, pero se determinó que su campo de captación era demasiado amplio, lo que generaba imprecisiones en la navegación. Como solución, se optó por sensores ultrasónicos, que permiten una mejor detección lineal y aseguran desplazamientos más estables.

Importancia del Modelado: La creación de modelos precisos del sistema ha sido clave para la implementación exitosa del MPC. La modelización detallada permitió una predicción precisa del comportamiento del robot, lo que ha sido fundamental para mejorar su rendimiento.

Optimización del Consumo Energético: Se identificó que el procesamiento continuo de datos consumía una cantidad considerable de energía. Se recomienda optimizar la gestión energética del robot mediante estrategias como la modulación de potencia en motores y sensores, lo que prolongará la autonomía sin afectar el rendimiento.

Limitaciones del Proyecto:

Capacidad de Cálculo: Aunque el MPC es un controlador avanzado, el procesamiento en tiempo real en hardware limitado, como el Lego Mindstorms EV3, presenta desafíos. Para ejecutar algoritmos más complejos sin comprometer el rendimiento, se recomienda mejorar la capacidad de cálculo del sistema.

Condiciones de Entorno: Aunque el MPC ha mostrado un buen rendimiento en entornos controlados, en escenarios con obstáculos impredecibles o ambientes altamente dinámicos, el sistema puede requerir ajustes adicionales. La implementación de algoritmos de aprendizaje automático podría permitir que el robot se adapte de manera autónoma a entornos no modelados.

5.2 Recomendaciones

Mejoras en la Precisión de Sensores

Selección de Sensores Óptimos: Se intentó usar un sensor infrarrojo para la detección de obstáculos, pero se determinó que su campo de captación es muy amplio, lo que generaba imprecisiones en la navegación. Como alternativa, se optó por sensores ultrasónicos, ya que su captación es más lineal y permiten un desplazamiento más estable en trayectorias rectas.

Optimización de Consumo Energético

Gestión Eficiente de la Energía: Se recomienda optimizar el consumo energético del robot ajustando los ciclos de muestreo y reduciendo el uso innecesario de procesamiento intensivo. Implementar estrategias como la modulación de potencia en motores y sensores puede prolongar la autonomía del sistema sin afectar el rendimiento.

Integración de Algoritmos de Aprendizaje

Adaptabilidad a Entornos Variables: Se sugiere explorar el uso de algoritmos de aprendizaje automático para que el robot pueda ajustar sus parámetros de control en función de datos recolectados en tiempo real. Esto permitiría mejorar la navegación en entornos desconocidos o con obstáculos dinámicos.

Consideraciones para Futuras Investigaciones:

Investigación en Control Predictivo Avanzado: Se recomienda continuar la investigación en métodos de control predictivo basados en modelos que no solo consideren la dinámica del robot, sino también las condiciones ambientales, lo que permitirá un control más robusto y eficiente.

Desarrollo de Nuevas Configuraciones de Hardware: Probar el rendimiento del MPC con diferentes configuraciones de hardware y algoritmos para evaluar su efectividad en una variedad de escenarios reales.

5.3 Referencias

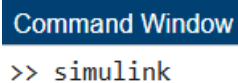
- MathWorks. (2024). *Model Predictive Control Toolbox Documentation*. MathWorks.
<https://www.mathworks.com/help/mpc/>
- MathWorks. (2024). *Simulink Support Package for LEGO MINDSTORMS EV3 Hardware*. MathWorks.
<https://www.mathworks.com/hardware-support/lego-mindstorms-ev3.html>
- MathWorks. (2024). *Simscape Multibody Documentation*. MathWorks.
<https://www.mathworks.com/help/physmod/sm/>
- Camacho, E. F., & Bordons, C. (2007). *Model Predictive Control*. Springer.
<https://link.springer.com/book/10.1007/978-1-84628-615-5>
- Maciejewski, J. M. (2002). *Predictive Control with Constraints*. Pearson Education.
<https://www.cambridge.org/core/books/predictive-control-with-constraints/2404B3D4C5742A2E18DB437EA3E3FA84>
- Corke, P. (2017). *Robotics, Vision and Control: Fundamental Algorithms in MATLAB (2nd ed.)*. Springer.
<https://link.springer.com/book/10.1007/978-3-319-54413-7>
- LEGO Education. (2023). *LEGO MINDSTORMS EV3 User Guide*. LEGO Education.
<https://education.lego.com/en-us/downloads/mindstorms-ev3/user-guide>
- Rawlings, J. B., Mayne, D. Q., & Diehl, M. (2017). *Model Predictive Control: Theory, Computation, and Design (2nd ed.)*. Nob Hill Publishing.
<https://www.nobhillpublishing.com/mpc/>
- Siciliano, B., & Khatib, O. (Eds.). (2016). *Springer Handbook of Robotics (2nd ed.)*. Springer.
<https://link.springer.com/referencework/10.1007/978-3-319-32552-1>
- Quigley, M., et al. (2009). *ROS: an open-source Robot Operating System*. ICRA Workshop on Open Source Software.
<https://www.ros.org/>

ANEXOS

En esta sección, se tiene que colocar todos los anexos que le ayuden a definir mejor las ideas utilizadas en la redacción de su documento, recuerde que la presencia de un anexo se justifica con la redacción que haya realizado.

6.1 Configuración del Proyecto en Simulink

Iniciar MATLAB y en la ventana de comandos, escribe lo siguiente para abrir Simulink:



```
Command Window
>> simulink
```

Figura 0.1 Comando para inicio de Simulink

Al abrirse la interfaz de Simulink, seleccionar la opción "Nuevo modelo" para iniciar un nuevo proyecto.

Agregar los siguientes bloques esenciales para la implementación del robot:

- EV3 Ultrasonic Sensor: Para la detección de obstáculos en el entorno.
- EV3 Bluetooth Send y EV3 Bluetooth Receive: Para la comunicación inalámbrica del robot mediante Bluetooth.
- Stateflow Chart: Para el control de los movimientos del robot, incluyendo la detección de obstáculos y las maniobras de evasión.
- EV3 Large Motor: Para controlar las ruedas del robot.
- MPC Controller: Para el control predictivo basado en modelos (MPC) que optimiza el comportamiento del robot.

Definir los estados del comportamiento del robot en función de los datos obtenidos del sensor ultrasónico:

- Movimiento Normal: El robot se mueve en línea recta.
- Detección de Obstáculo: Si la distancia detectada es menor que un umbral, el robot cambia de dirección.
- Evasión de Obstáculo: El robot gira y busca un camino libre al detectar un obstáculo.
- Recepción de Comandos Bluetooth: El robot puede recibir comandos externos para detenerse o cambiar de comportamiento.

Se tiene como ejemplo de código en Stateflow el cual controla los movimientos del robot según los valores del sensor ultrasónico y los comandos Bluetooth recibidos.

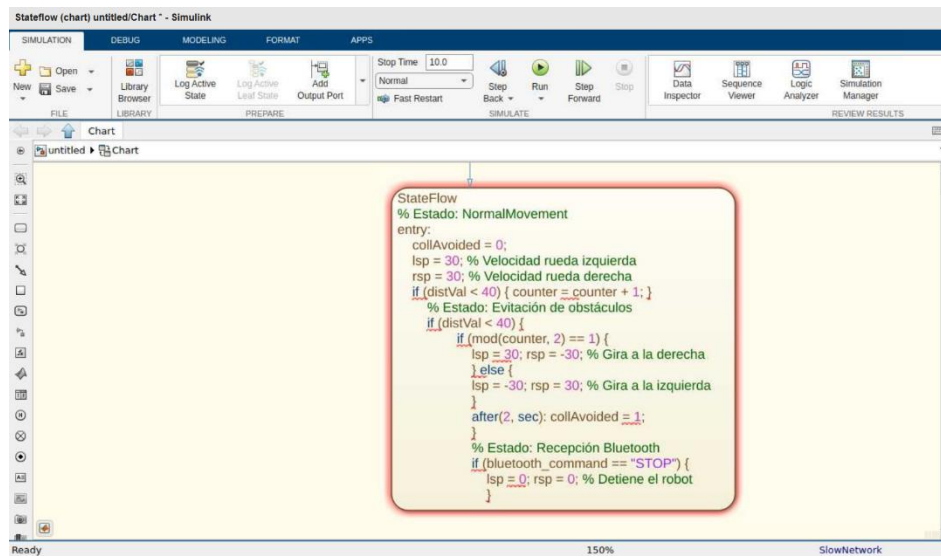


Figura 0.2 Código en Stateflow

Definir el modelo matemático del robot móvil diferencial para el controlador MPC. Se utilizan parámetros como el radio de las ruedas y la distancia entre ellas para representar el sistema dinámico.

Diseñar el controlador MPC agregando el bloque "MPC Controller" y configurarlo utilizando el modelo del robot que se ha definido anteriormente.

```

>> mpcobj = mpc(sys, 0.1); % Modelo con un tiempo de muestreo de 0.1s

>> // Estado: NormalMovement
entry:
    collAvoided = 0;
    lsp = 30; // Velocidad rueda izquierda
    rsp = 30; // Velocidad rueda derecha
    if (distVal < 40) { counter = counter + 1; }

// Estado: Evitación de obstáculos
if (distVal < 40) {
    if (mod(counter, 2) == 1) {
        lsp = 30; rsp = -30; // Gira a la derecha
    } else {
        lsp = -30; rsp = 30; // Gira a la izquierda
    }
    after(2, sec): collAvoided = 1;
}

// Estado: Recepción Bluetooth
if (bluetooth_command == "STOP") {
    lsp = 0; rsp = 0; // Detiene el robot
}

```

Figura 0.3 Código de Estados

Conectar el bloque "MPC Controller" con los motores del robot para que el controlador pueda actuar sobre ellos y seguir la trayectoria deseada.

En el robot EV3, acceder a la configuración de Bluetooth y activar la comunicación inalámbrica.

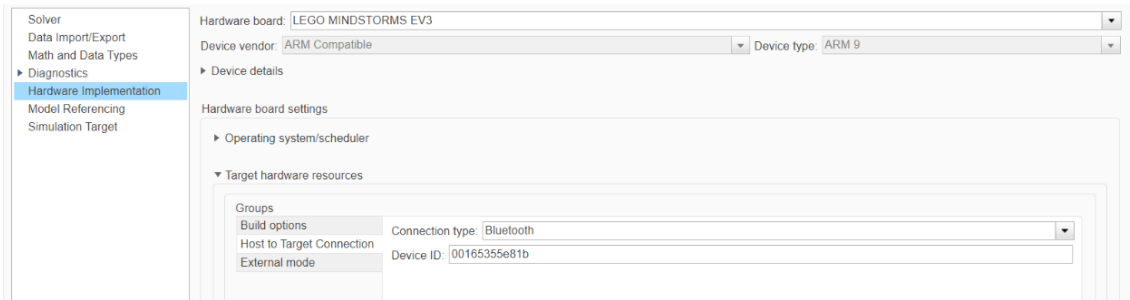


Figura 0.4 Selección de Bluetooth para conexión con EV3

Utilizar los bloques "EV3 Bluetooth Send" y "EV3 Bluetooth Receive" para configurar la comunicación entre MATLAB y el robot.

Modificar el algoritmo en Stateflow para que el robot responda a los comandos recibidos (por ejemplo, "STOP") y se detenga o realice otras acciones según sea necesario.

Para la implementación del programa en el Robot, se debe realizar los siguientes pasos:

- Una vez que todos los bloques están configurados y conectados correctamente, compilar el modelo presionando Ctrl + B para generar el código.
- Después de compilar, desplegar el código al hardware del EV3 utilizando la opción "Deploy to Hardware".
- Una vez cargado el código en el robot, realizar pruebas en un circuito con obstáculos para verificar su capacidad de seguir la trayectoria y evitar obstáculos de manera autónoma.

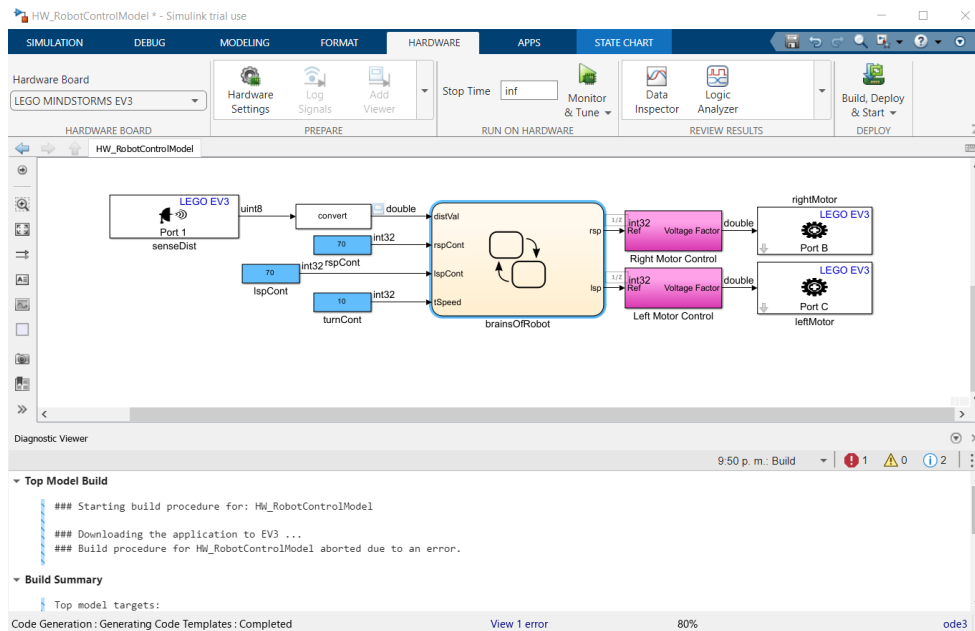


Figura 0.5 Esquema de trabajo con modelo completo en Simulink

6.2 Importación del Modelo CAD en Simulink

Para validar la estructura mecánica del robot en el entorno de simulación, se importó un modelo CAD desarrollado en SolidWorks a Simulink utilizando Simscape Multibody. A continuación, se describe el procedimiento utilizado:

6.2.1 Requisitos Previos

Antes de iniciar el proceso de importación, se deben cumplir los siguientes requisitos:

- Tener instalado MATLAB con la toolbox Simscape Multibody.
- Contar con el archivo XML exportado desde SolidWorks y los archivos STL de cada componente mecánico.
- Conocer la ubicación de los archivos en el sistema.

6.2.2 Procedimiento Paso a Paso

Paso 1: Configurar el Directorio de Trabajo

Abre MATLAB y cambia el directorio de trabajo a la carpeta donde se encuentran los archivos XML y STL:

```
>> cd('C:\Users\Ivan
Hernandez\Desktop\Obstacle_Avoidance\Obstacle_Avoidance
(1)\Obstacle_Avoidance\STLFiles')
```

Código 0.1 Ubicación de proyecto

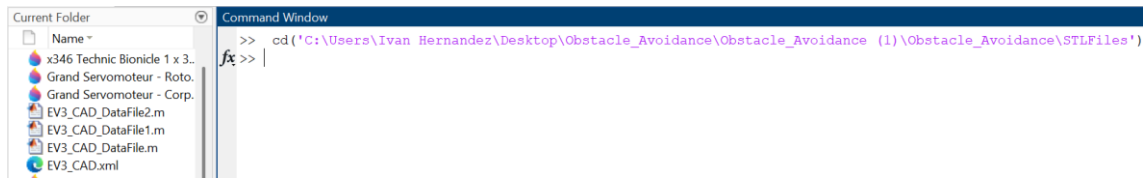


Figura 0.6 Ubicación del directorio y proyecto en Simulink

Paso 2: Importar el Modelo en Simulink

Utiliza la función `smimport` para cargar el archivo XML del modelo CAD en Simulink:

```
>> smimport('EV3_CAD.xml')
```

Código 0.2 Importación de archivo *EV3_CAD.xml*

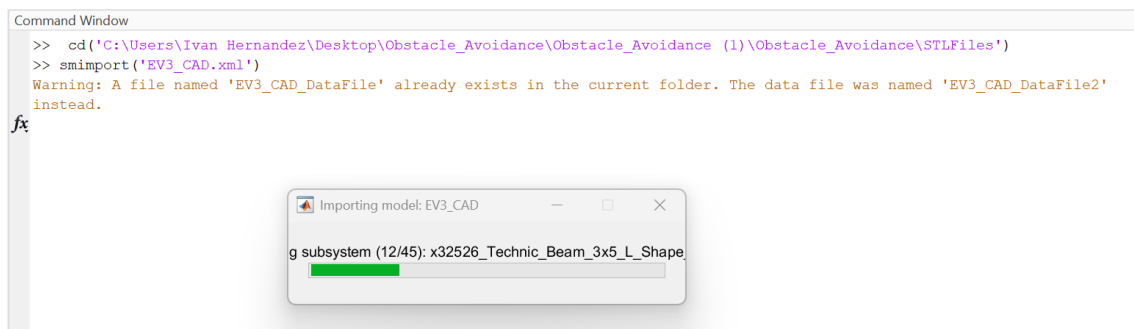
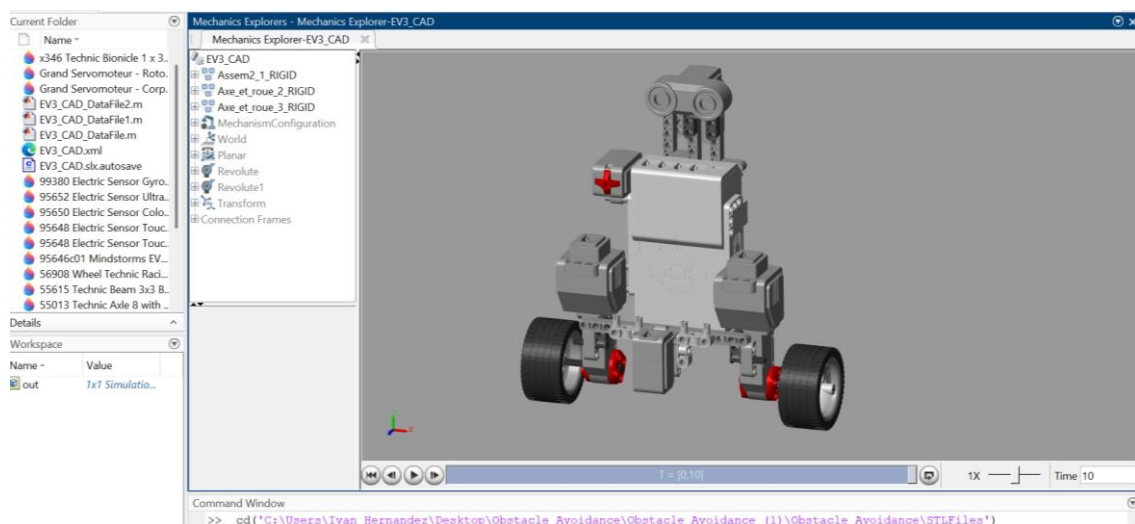
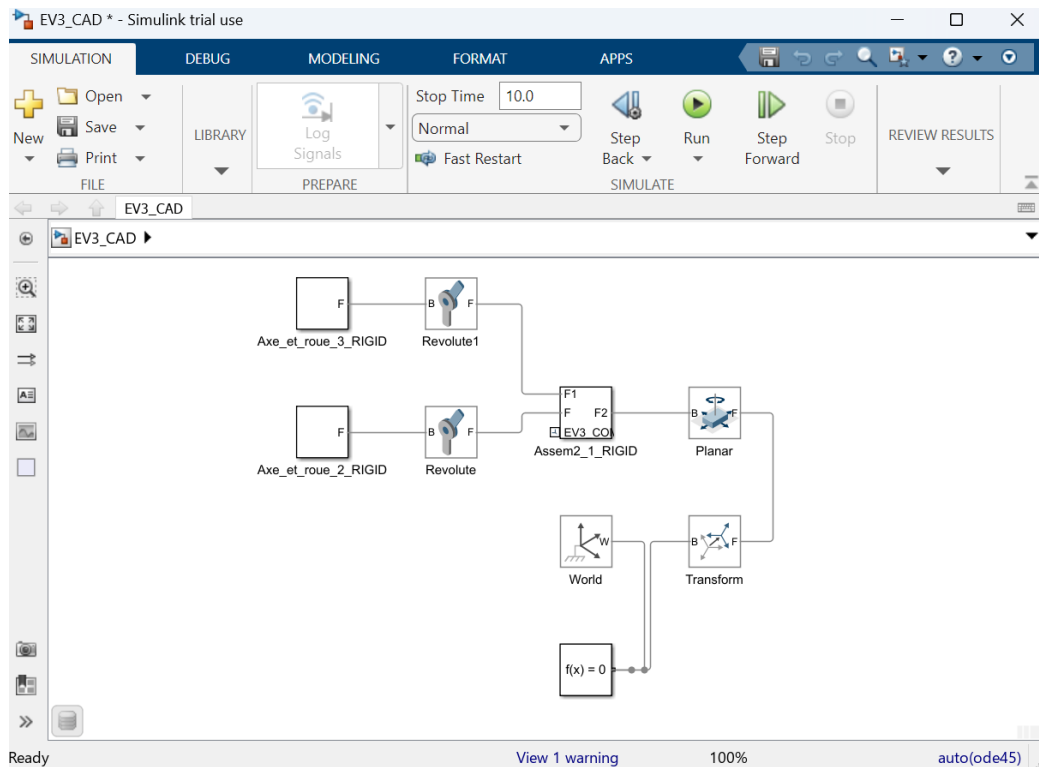


Figura 0.7 Importación del modelo

Paso 3: Explorar el Modelo en Simulink

Después de ejecutar el comando anterior, MATLAB generará automáticamente un modelo en Simulink con los bloques de Simscape Multibody correspondientes. Para abrir el modelo generado, correr el programa en RUN:



6.3 Construcción Física del Robot

Finalmente, se ensambló el robot físicamente utilizando piezas de LEGO Mindstorms EV3 y se cargó el programa desarrollado en Simulink a la unidad EV3 a continuación se muestran las piezas utilizadas en el ensamblaje y el resultado final:

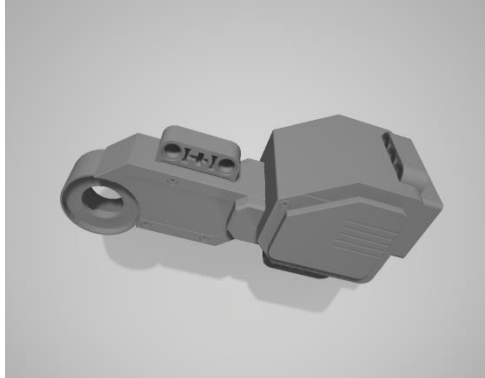


Figura 0.10 Servomotor

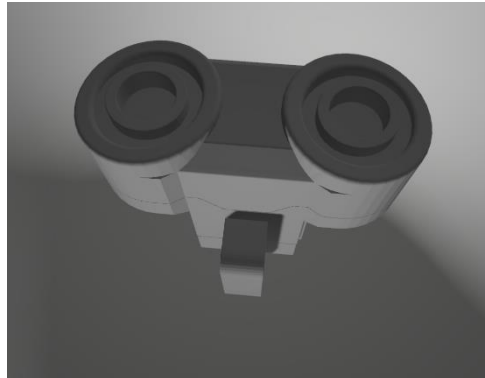


Figura 0.11 Sensor ultrasónico

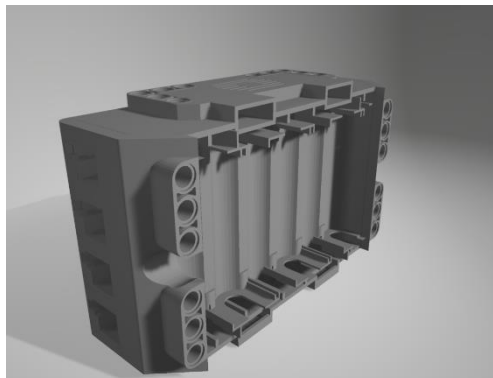


Figura 0.12 Bloque mindstorm trasero

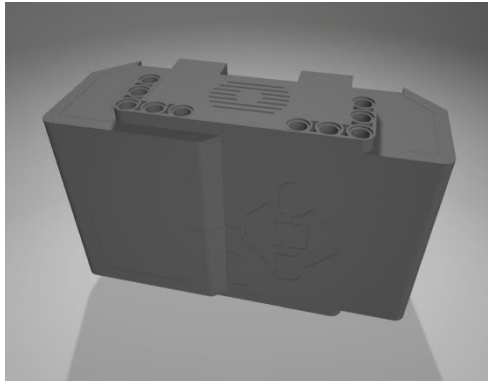


Figura 0.13 Bloque mindstorm delantero

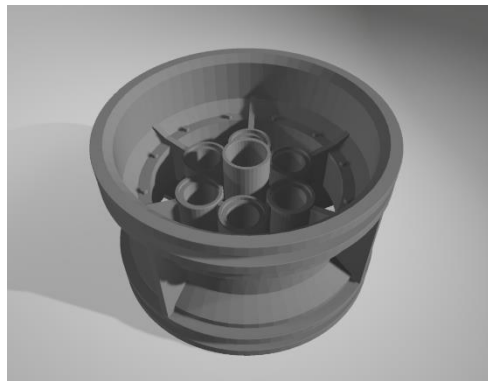


Figura 0.14 Rueda con 6 orificios

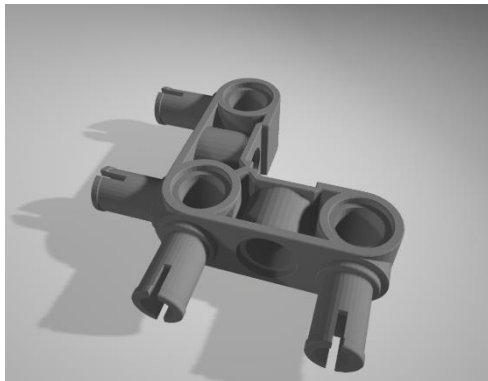


Figura 0.15 Viga 3x3 con pasadores



Figura 0.16 Eje largo con tope

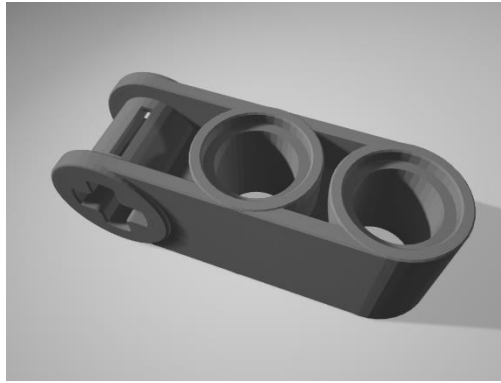


Figura 0.17 Ensamblador de eje

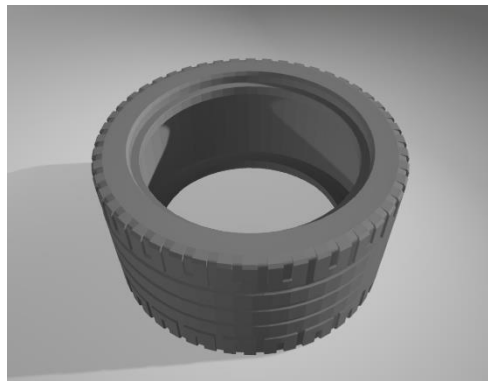


Figura 0.18 Neumatico RG

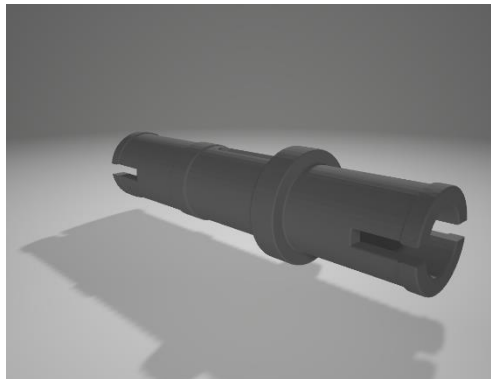


Figura 0.19 Pin largo técnico

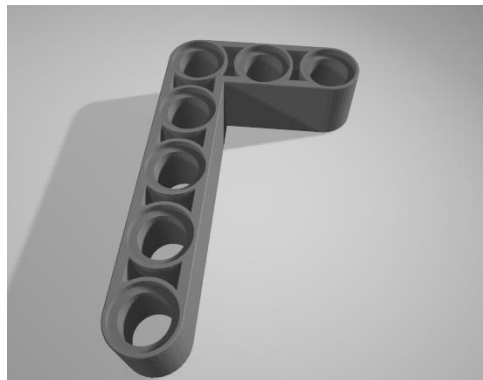


Figura 0.20 Soporte elevador 3x5

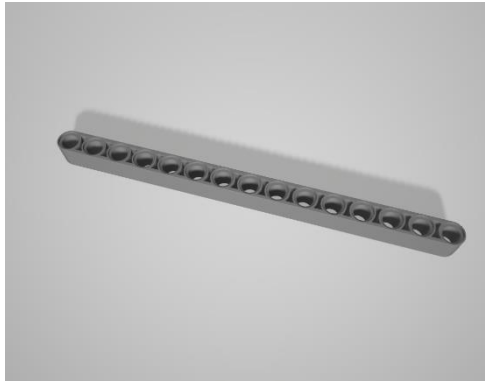


Figura 0.21 Soporte elevador 1x15

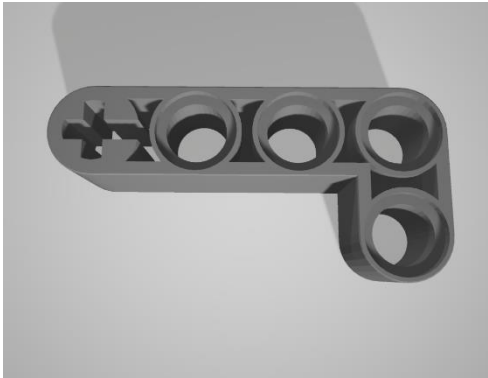


Figura 0.22 Soporte elevador 2x4

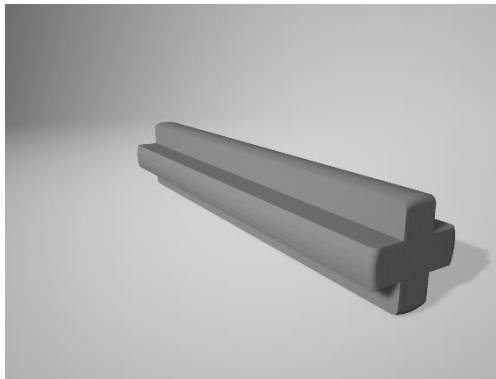


Figura 0.23 Eje técnico

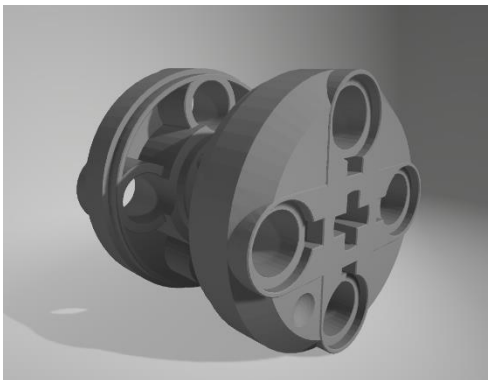


Figura 0.24 Rotor de servo motor

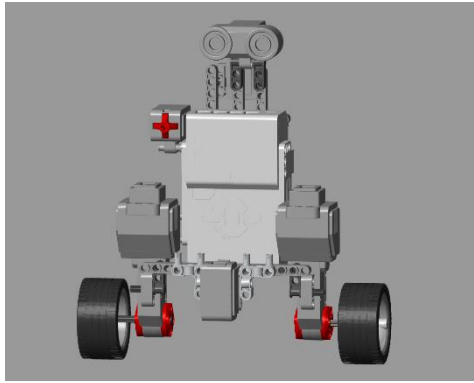


Figura 0.25 Robot ensamblado vista delantera

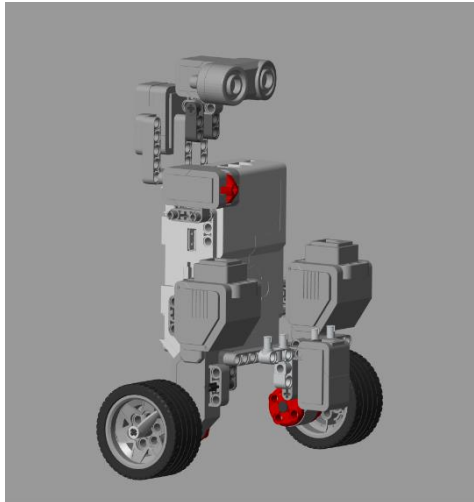


Figura 0.26 Robot ensamblado vista lateral

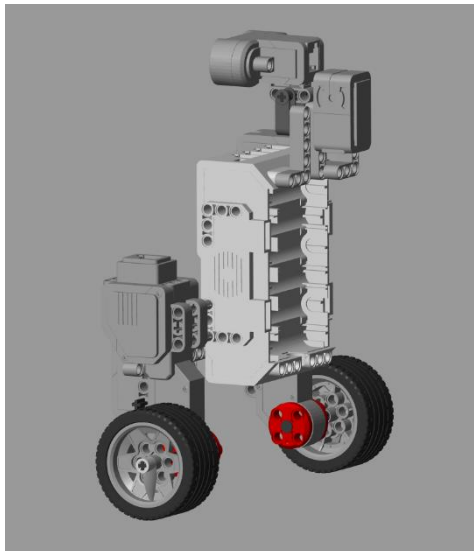


Figura 0.27 Robot ensamblado vista posterior lateral

6.4 RESULTADO FINAL

Al finalizar la implementación, el robot EV3 será capaz de:

- **Seguir una trayectoria** con el controlador MPC.
- **Detectar obstáculos** con el sensor ultrasónico.
- **Evitar obstáculos** de manera autónoma.
- **Controlarse vía Bluetooth** para recibir comandos externos.