

UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA



Profesor Patrocinante:
Dr. Lautaro D. Salazar S.

Informe de Memoria de Título
para optar al título de:
Ingeniero Civil Eléctrico

Asistente inteligente para movilidad, gestión
energética y efectividad de servicio de un
vehículo eléctrico autónomo.

UNIVERSIDAD DE CONCEPCIÓN
Facultad de Ingeniería
Departamento de Ingeniería Eléctrica

Profesor Patrocinante
Dr. Lautaro D. Salazar S.

Asistente inteligente para movilidad, gestión energética y efectividad de servicio de un vehículo eléctrico autónomo.

Daniel Alexis Ruminot Moscoso

Informe de Memoria de Título
para optar al Título de

Ingeniero Civil Eléctrico

Diciembre 2024

Resumen

Este informe presenta el desarrollo de un asistente inteligente para un vehículo eléctrico autónomo, centrado en mejorar la movilidad, gestión energética y efectividad del servicio en entornos urbanos y confinados como centros comerciales y parques. La creciente necesidad de soluciones de movilidad sostenible hace que este proyecto sea relevante y prometedor.

La solución propuesta consiste en un sistema de control inalámbrico basado en Internet que mejora la movilidad del vehículo, habilita la gestión energética y aumenta la efectividad del servicio. Para el desarrollo del proyecto se presenta una metodología que incluye la revisión del estado del arte, el análisis y simulación del modelo matemático del vehículo, la selección e integración de sensores y actuadores, la implementación del sistema de control inalámbrico al cual se le incluye una interfaz de usuario intuitiva. Los resultados se validan en un prototipo de vehículo autónomo disponible.

El sistema de control inalámbrico utiliza un módem 4G para la conectividad a Internet, permitiendo acceso remoto y transmisión en tiempo real de datos. También se desarrolló un dispositivo auxiliar de control de corta distancia con un microcontrolador ESP8266 conectado a un acelerómetro y alimentado de una batería de litio teniendo como objetivo ejecutar maniobras precisas, la carcasa fue diseñada con softwares de modelado 3D y fue construido con filamento de material PLA con una impresora 3D.

Las pruebas de campo han demostrado la eficiencia y seguridad del sistema, cumpliendo con los objetivos del proyecto. La recopilación y análisis de datos en tiempo real permiten ajustes dinámicos que mejoran el rendimiento y prolongan la vida útil del vehículo.

El proyecto representa una contribución en la movilidad autónoma y la gestión energética, sentando las bases para futuras investigaciones y desarrollos en vehículos eléctricos autónomos. La implementación de este sistema mejora la eficiencia operativa aumentando la autonomía lo que contribuye a potenciar el uso de vehículos eléctricos.



A los alumnos del pasado, presente y futuro de la UDEC.

Agradecimientos

Quiero expresar mi más profundo agradecimiento a todas las personas que me han apoyado y acompañado durante el desarrollo de este proyecto.

En primer lugar, agradezco sinceramente a mi profesor guía, Lautaro Salazar, por su invaluable orientación, apoyo y paciencia a lo largo de este trabajo. Su experiencia y conocimientos han sido fundamentales para la realización de esta tesis.

A mi madre, María Elena Moscoso, y a mi tía, Marisol Moscoso, les agradezco de todo corazón por su amor incondicional y su constante apoyo. Sin su confianza y ánimo, no habría podido llegar hasta aquí.

A mi pareja, Francisca Roa, gracias por estar siempre a mi lado, por tu comprensión y por ser mi fuente de inspiración. Tu apoyo ha sido esencial para superar cada desafío en este camino.

También quiero agradecer de manera especial a mis compañeros de laboratorio, quienes han sido una fuente constante de motivación y camaradería. Su ayuda y colaboración han sido invaluableles en la culminación de este proyecto.

A mis familiares y amigos, les agradezco por su continuo apoyo y por creer en mí. Su presencia y aliento han sido un pilar fundamental en mi vida.

Finalmente, agradezco a toda la institución y organización de la Universidad de Concepción que han contribuido de alguna manera a la realización de este trabajo.

Tabla de Contenidos

NOMENCLATURA	IX
ABREVIACIONES	IX
1. INTRODUCCIÓN.....	1
1.1. INTRODUCCIÓN GENERAL	1
1.2. JUSTIFICACIÓN DEL PROYECTO	1
1.3. IDENTIFICACIÓN DEL PROBLEMA	2
1.4. OBJETIVOS DEL PROYECTO	2
1.5. IMPORTANCIA DEL PROBLEMA	3
1.6. JUSTIFICACIÓN DEL ESTUDIO	3
1.7. ESTADO DEL ARTE	4
1.8. METODOLOGÍA.....	7
1.9. ALCANCES DEL PROYECTO Y LIMITACIONES.....	7
<i>Alcances comprometidos.....</i>	<i>8</i>
2. PROTOTIPO DE VEHÍCULO ELÉCTRICO AUTÓNOMO.....	9
2.1. INTRODUCCIÓN.....	9
3. MODELO MATEMÁTICO DEL VEHÍCULO AUTÓNOMO	17
3.1. INTRODUCCIÓN.....	17
3.2. DINÁMICA DEL VEHÍCULO AUTÓNOMO	18
3.3. MODELADO DEL MOTOR DE CORRIENTE CONTINUA.....	22
3.4. VALIDACIÓN DEL MODELO EN LAZO ABIERTO MEDIANTE SIMULACIÓN.....	¡ERROR! MARCADOR NO DEFINIDO.
3.5. INTEGRACIÓN DE LOS COMPONENTES DEL SISTEMA EN UNA SIMULACIÓN	25
3.6. VALIDACIÓN EXPERIMENTAL.....	33
3.7. CONCLUSIONES MODELADO MATEMÁTICO	36
4. INGENIERÍA CONCEPTUAL DE OPERACIÓN Y CONTROL REMOTO DEL VEHÍCULO.....	38
4.1. INTRODUCCIÓN.....	38
4.2. MONITOREO Y SELECCIÓN DE VARIABLES.....	38
4.3. IDENTIFICACIÓN DE VARIABLES CRÍTICAS.....	39
4.4. SENSORES PARA EL MONITOREO	40
4.5. ESTRUCTURA HARDWARE Y FUNCIONES.....	41
4.6. HERRAMIENTAS Y SOFTWARE UTILIZADOS PARA EL PROCESAMIENTO DE DATOS	51
4.7. SISTEMA DE MONITOREO	53
4.8. SISTEMA DE ENERGÍA ELÉCTRICA Y DISTRIBUCIÓN.....	54
4.9. CONCLUSIONES	57
5. CONTROL INALÁMBRICO.....	58
5.1. INTRODUCCIÓN.....	58
5.2. ESTRUCTURA DE CONTROL.....	59
5.3. IMPLEMENTACIÓN	59
5.4. SEGURIDAD Y VALIDACIÓN DEL SISTEMA.....	61
5.5. IMPLEMENTACIÓN DE UN DISPOSITIVO AUXILIAR DE CONTROL DE CORTA DISTANCIA	61
5.6. MÉTODO DE CONTROL INALÁMBRICO Y PROTOCOLOS DE COMUNICACIÓN.....	65
6. INTERFAZ DE USUARIO Y VISUALIZACIÓN DE DATOS	66
<i>Conclusiones</i>	<i>71</i>
7. DISCUSIÓN Y CONCLUSIONES.....	72
7.1 <i>Resultado Principales</i>	<i>72</i>
7.2 <i>Conclusiones Generales y Cierre del Proyecto</i>	<i>75</i>
REFERENCIAS.....	77

Lista de Ilustraciones

Ilustración 1 Modelo 3d de la Estructura del VEA	9
Ilustración 2 Placa DIE Puente H.....	11
Ilustración 3 LMD18200	11
Ilustración 4 Placa DIE Puente H.....	12
Ilustración 5 Placa DIE Puente H.....	12
Ilustración 6 Placa DIE Puente H.....	13
Ilustración 7 Modelo Cinemático Diferencial del Automóvil.....	19
Ilustración 8 Diagrama de Bloque Matlab/Simulink Movimiento Cinemático Diferencial	20
Ilustración 9 Movimiento hacia adelante Distancia vs Tiempo	20
Ilustración 10 Movimiento hacia atrás Distancia vs Tiempo	21
Ilustración 11 Movimiento con giro a la derecha Distancia vs Tiempo.....	21
Ilustración 12 Movimiento con giro a la izquierda Distancia vs Tiempo	21
Ilustración 13 Modelo del Motor	22
Ilustración 14 Diagrama de Bloque del Motor.....	¡Error! Marcador no definido.
Ilustración 15 Representación del automóvil en Simscape Matlab.....	25
Ilustración 16 Representación del automóvil en Simscape Matlab.....	26
Ilustración 17 Voltaje Motor Vs Tiempo	26
Ilustración 18 Corriente Motor Vs Tiempo.....	27
Ilustración 19 Potencia Motor Vs Tiempo	27
Ilustración 20 Velocidad Automóvil Vs Tiempo	28
Ilustración 21 Aceleración Automóvil Vs Tiempo	28
Ilustración 22 Distancia recorrida por Automóvil Vs Tiempo.....	29
Ilustración 23 Modelo para inclinación.....	29
Ilustración 24 Voltaje Motor Vs Tiempo	30
Ilustración 25 Corriente Motor Vs Tiempo.....	30
Ilustración 26 Potencia Motor Vs Tiempo	31
Ilustración 27 Angulo de Inclinación Automóvil Vs Tiempo.....	31
Ilustración 28 Aceleración Automóvil Vs Tiempo	32
Ilustración 29 Velocidad Automóvil Vs Tiempo	32
Ilustración 30 Prueba de Campo del VEA	35
Ilustración 31 Arquitectura del Hardware	41
Ilustración 32 Conexión de los equipo que interactúan con la Computadora Raspberry pi 4	42
Ilustración 33 Conexión Micro Controlador Atmega328 con sensores de corriente ASC712 y voltaje FZ0430	43
Ilustración 34 Código Controlador Atmega328 con sensores de corriente ASC712 y voltaje FZ0430	44
Ilustración 35 Conexión Micro Controlador Atmega328 con sensores de proximidad y Acelerómetro	45
Ilustración 36 Conexión Micro Controlador Atmega328 con sensores de proximidad y Acelerómetro	46
Ilustración 37 Conexión Micro Controlador Atmega328 con Encoder	47
Ilustración 38 Conexión Micro Controlador Atmega328 con Encoder	48
Ilustración 39 Código Python que recibe los datos en Raspberry pi.....	49

Ilustración 40 Rutina configuración Cámaras en Raspbian	50
Ilustración 41 Dispositivos y tecnologías empleadas . Referenciado como interactúan entre ellas ..	53
Ilustración 42 Unilineal Distribución Eléctrica del VEA.....	55
Ilustración 43 Estructura de la implementación.....	60
Ilustración 44 Flujo de Trabajo del Dispositivo Auxiliar	61
Ilustración 45 Circuito Control Inalámbrico	62
Ilustración 46 Modelo 3D del Control Inalámbrico Diseñado Para Impresión 3D.....	63
Ilustración 47 Interfaz de usuario.....	71

Lista de Tablas

Tabla 1 Parámetros del Chasis	10
Tabla 2 Parámetros del Puente H LMD18200	14
Tabla 3 Parámetros del Motor Maxon DC motor RE 35	24
Tabla 4 Resultados pruebas de laboratorio a Motor Maxon DC motor RE 35	34



Nomenclatura

Escalares

[V]	: Volt
[A]	: Ampere
[rpm]	: revoluciones por minuto
[mts]	: Metros
[Kg]	: Kilogramos
[Ω]	: Ohm
[H]	: Henrios
[N*m]	: Fuerza
[Watt]	: Potencia
[Seg]	: Segundos
[Va]	: Volt Armadura
[Ia]	: Corriente Armadura
[Ra]	: Resistencia Armadura
[La]	: Inductancia armadura
[Tnom]	: Torque Nominal
[Nnom]	: Velocidad nominal



Abreviaciones

Mayúsculas

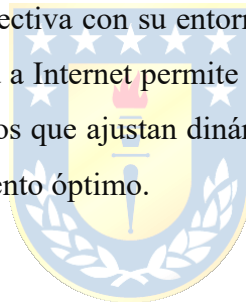
F. de T.	: función de transferencia.
M. de T.	: matriz de transferencia.
B.W.	: ancho de banda.
VEA.	: vehículo eléctrico Autónomo.
P.I.D.	: controlador proporcional integral derivativo.
T.L.	: Transformada de Laplace.
T.F.	: Transformada de Fourier.
T.F.F.D.	: Transformada de Fourier de Frecuencia Discreta.
T.Z.	: Transformada \mathcal{Z} .
T.F.T.D.	: Transformada de Fourier de Tiempo Discreta.
T.F.D.	: Transformada de Fourier Discreta.

1. Introducción

1.1. Introducción General

El desarrollo de la tecnología en el ámbito de la movilidad eléctrica ha cobrado una relevancia significativa en los últimos años, especialmente en lo que respecta a la eficiencia energética y la sostenibilidad. Este informe se centra en la implementación de un asistente inteligente para un vehículo eléctrico autónomo, con el objetivo de mejorar su movilidad, gestión energética y efectividad del servicio. Este proyecto es parte del campo de la Ingeniería Eléctrica y, a su vez, contribuye a los avances en la ingeniería en general.

En el contexto actual[2] [3], la optimización de los vehículos eléctricos autónomos no solo se enfoca en la autonomía de la batería u el control de los motores, sino también en la capacidad de estos vehículos para interactuar de manera efectiva con su entorno. La integración de sistemas de control inalámbrico que utilicen la conectividad a Internet permite no solo el monitoreo en tiempo real, sino también la implementación de algoritmos que ajustan dinámicamente los parámetros de conducción del vehículo para garantizar un rendimiento óptimo.



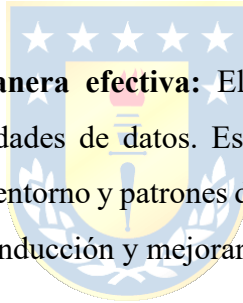
1.2. Justificación del Proyecto

La movilidad sostenible es crucial en nuestras ciudades, y los vehículos eléctricos autónomos ofrecen una solución prometedora. Para ser efectivos, estos vehículos necesitan sistemas de manejo que optimicen el uso de energía, mejoren la experiencia del usuario y aseguren un servicio fiable. Para lograr esto, se requiere una gran cantidad de datos para la creación de modelos matemáticos que sean capaces de predecir y responder a diversas condiciones de conducción. Actualmente, la integración de estos datos en sistemas de control autónomos aún no está completamente desarrollada, lo que presenta un área de oportunidad significativa para la investigación y el desarrollo.

1.3. Identificación del Problema

El problema central de este proyecto radica en la necesidad de integrar un sistema de control inalámbrico basado en Internet para un vehículo eléctrico autónomo . La autonomía en este contexto se refiere a la capacidad de controlar el vehículo de manera remota y no de la capacidad energética del vehículo, garantizando el manejo de las variables básicas necesarias para su correcta operación. Este sistema debe ser capaz de:

- **Optimizar la movilidad del vehículo:** Asegurando que pueda navegar de manera autónoma y eficiente en diversos entornos, tanto interiores como exteriores.
- **Mejorar la gestión energética:** Proporcionando un uso eficiente de la energía disponible, lo cual es crucial para maximizar la capacidad energética del vehículo y su operatividad.
- **Aumentar la efectividad del servicio proporcionado:** Garantizando que el vehículo pueda cumplir con sus tareas de manera confiable y segura, interactuando adecuadamente con su entorno y los usuarios.
- **Gestionar y utilizar datos de manera efectiva:** El vehículo debe ser capaz de recopilar, transmitir y procesar grandes cantidades de datos. Estos datos incluyen información sobre el estado del vehículo, condiciones del entorno y patrones de uso, que son fundamentales para ajustar dinámicamente los parámetros de conducción y mejorar la eficiencia y la seguridad del vehículo.



1.4. Objetivos del Proyecto

El Objetivo principal es desarrollar e implementar un sistema de control inalámbrico basado en Internet para un vehículo eléctrico autónomo, optimizando su movilidad, mejorando la gestión energética y aumentando la efectividad del servicio proporcionado. El sistema implementado busca sentar las bases para definir la infraestructura necesaria en la obtención de datos, que ayuden a futuros estudios de Machine Learning e Inteligencia Artificial en el campo de vehículo eléctrico autónomo. Objetivos Especificos comprometidos los conceptos y caracterissticas de movilidad propuestas se verificsn utilizando prototipo de auto autonomo disponible :

- Describir el auto eléctrico disponible analizando el modelo matemático que caracteriza el accionamiento eléctrico del vehículo y validarlo con un programa de simulación.

- Analizar y especificar las variables a monitorear e investigar y proponer los sensores y actuadores más adecuados para la medición precisa de parámetros.
- Desarrollar e implementar un método de control inalámbrico basado en Internet que asegure la seguridad y eficiencia en la transmisión de datos entre el vehículo autónomo y la red.
- Diseñar e implementar una interfaz que permita al usuario el monitoreo y control del sistema mediante un computador portátil, facilitando una interacción eficaz y una visualización del estado del vehículo para su control.

1.5. Importancia del Problema

La resolución de los problemas identificados en el desarrollo de un sistema de control inalámbrico para vehículos eléctricos autónomos tiene una gran importancia tecnológica, ambiental y social. Tecnológicamente, representa un avance en la movilidad autónoma y la inteligencia vehicular, mejorando la eficiencia operativa y sentando las bases para futuras investigaciones en Inteligencia Artificial y Machine Learning aplicados a la movilidad.

Ambientalmente, el uso de vehículos eléctricos autónomos reduce significativamente las emisiones de gases de efecto invernadero y mejora la calidad del aire en entornos urbanos, disminuyendo la dependencia de combustibles fósiles. Socialmente, estos vehículos mejoran la movilidad urbana, ofreciendo soluciones de transporte más seguras y eficientes, especialmente en áreas con movilidad limitada, lo que impacta positivamente en la calidad de vida y reduce la congestión vial.

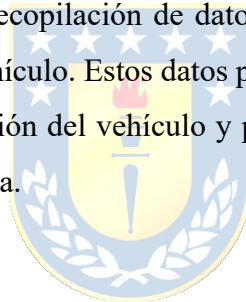
La gestión efectiva de los datos recopilados por estos vehículos mejorará su desempeño en tiempo real y proporcionará una base valiosa para futuros estudios. Estos datos permitirán desarrollar algoritmos de aprendizaje automático que optimicen la movilidad y eficiencia energética de los vehículos eléctricos autónomos, promoviendo el progreso continuo en esta área tecnológica.

1.6. Justificación del Estudio

La movilidad sostenible es un desafío crucial en el contexto urbano moderno, y los vehículos eléctricos autónomos ofrecen una solución prometedora. Este estudio se justifica por varias razones:

- **Impacto Ambiental:** La reducción de las emisiones de gases de efecto invernadero es esencial para mitigar el cambio climático. Los vehículos eléctricos autónomos, al ser más eficientes y depender de fuentes de energía limpia, contribuyen significativamente a este objetivo.

- **Eficiencia Energética:** La gestión óptima de la energía en vehículos eléctricos no solo mejora su autonomía, sino que también reduce los costos operativos. Un sistema de control avanzado puede maximizar el uso eficiente de la energía, extendiendo la vida útil de las baterías y mejorando la sostenibilidad del vehículo.
- **Avance Tecnológico:** El desarrollo e implementación de sistemas de control inalámbrico basados en Internet para vehículos eléctricos autónomos representa un avance significativo en la tecnología de la movilidad. Este proyecto no solo mejora la funcionalidad del vehículo, sino que también proporciona una base para futuras investigaciones y desarrollos en inteligencia artificial y machine learning.
- **Seguridad y Confiabilidad:** La capacidad de un vehículo autónomo para interactuar de manera segura y eficiente con su entorno es fundamental. Este estudio aborda la necesidad de sistemas de control que garanticen una respuesta rápida y precisa a las condiciones cambiantes del entorno, mejorando la seguridad y la confiabilidad del servicio.
- **Recolección y Uso de Datos:** La recopilación de datos en tiempo real es crucial para mejorar continuamente el desempeño del vehículo. Estos datos pueden utilizarse para desarrollar modelos predictivos que optimicen la operación del vehículo y proporcionen información valiosa para la investigación en movilidad autónoma.



1.7. Estado del Arte

La evolución tecnológica en el ámbito de los vehículos autónomos ha experimentado un crecimiento significativo en la última década, impulsado por avances en inteligencia artificial, sensores, control y conectividad. Este documento explora las tendencias actuales, desafíos y soluciones propuestas en el desarrollo de estos sistemas, destacando tanto los aspectos técnicos como las implicaciones sociales y económicas.

Tecnologías de Vehículos Autónomos

- **Sistemas Avanzados de Asistencia al Conductor (ADAS)**

Los sistemas ADAS son esenciales en la transición hacia vehículos autónomos, proporcionando funcionalidades como el control de crucero adaptativo, mantenimiento de carril y frenado automático. Según Anrigo, Gouget y Maignan (2015), estos sistemas han demostrado su eficacia en la validación y prueba de entornos dinámicos. [2]

- **Control Cooperativo y Toma de Decisiones en Tiempo Real**

La implementación de vehículos autónomos requiere la integración de algoritmos cooperativos capaces de tomar decisiones en tiempo real. Vlacic, Thomas y Pérez (2011) subrayan la importancia de sistemas que permitan la interacción efectiva entre múltiples vehículos y el entorno, optimizando la seguridad y la eficiencia. [3]

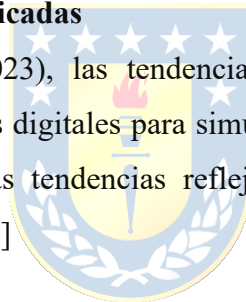
- **Comentario y Conclusión:**

Los avances en ADAS y en los sistemas de toma de decisiones cooperativas son esenciales para garantizar el funcionamiento seguro y eficiente de los vehículos autónomos. Estas tecnologías establecen la base para lograr mayores niveles de automatización.

Tendencias Actuales en Vehículos Autónomos

- **Principales Tendencias Identificadas**

De acuerdo con StartUs Insights (2023), las tendencias recientes incluyen la integración de inteligencia artificial, el uso de gemelos digitales para simulación, y la implementación de modelos sostenibles de movilidad urbana. Estas tendencias reflejan una evolución hacia un ecosistema tecnológico más robusto y sostenible. [7]



- **Perspectivas desde la Industria**

Chiao (2024) presenta una visión optimista basada en la colaboración entre empresas tecnológicas y fabricantes automotrices. Este enfoque está diseñado para acelerar la adopción de vehículos autónomos mediante la combinación de recursos y conocimiento especializado. [5]

- **Comentario y Conclusión:**

Las tendencias actuales demuestran un enfoque dual en sostenibilidad e innovación tecnológica. La colaboración entre industrias es un componente clave para superar los retos asociados con la adopción masiva de vehículos autónomos.

Infraestructura Tecnológica

- **Sistemas de Control y Modelado Matemático**

Ogata (2010) establece los principios fundamentales para el diseño de sistemas de control, proporcionando una base teórica sólida para la implementación de vehículos autónomos. [6]

- **Plataformas de Desarrollo y Simulación**

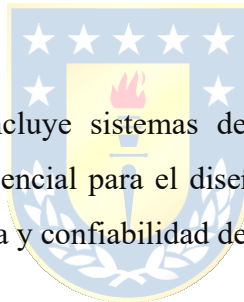
Las plataformas como Arduino y Raspberry Pi permiten un desarrollo rápido y flexible de prototipos. Según Raspberry Pi Foundation (n.d.), estas herramientas son ideales para la integración de sensores y sistemas embebidos en proyectos de vehículos autónomos. [17] [10]

- **Análisis de Datos y Machine Learning**

Firestore y BigQuery se destacan como herramientas efectivas para manejar grandes volúmenes de datos, cruciales para el entrenamiento y la predicción en tiempo real de modelos de machine learning (Google, n.d.). [8] [9]

- **Comentario y Conclusión:**

La infraestructura tecnológica, que incluye sistemas de control, plataformas de simulación y herramientas de análisis de datos, es esencial para el diseño y desarrollo de vehículos autónomos. Estas tecnologías garantizan la eficiencia y confiabilidad de los sistemas.



Sostenibilidad y Futuro

- **Impacto de los Vehículos Eléctricos Autónomos**

Stricker et al. (2020) enfatizan que los vehículos eléctricos autónomos tienen el potencial de transformar el transporte, ofreciendo soluciones más sostenibles y eficientes. [4]

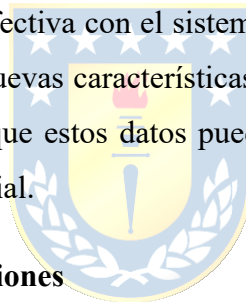
- **Comentario y Conclusión:**

Aunque los avances son significativos, persisten retos en términos de regulación, aceptación social y seguridad cibernética (Chiao, 2024). Estos aspectos requieren un enfoque colaborativo entre los diferentes actores involucrados.

El impacto de los vehículos eléctricos autónomos es innegable en términos de sostenibilidad, pero su adopción masiva depende de superar barreras técnicas, sociales y regulatorias.

1.8. Metodología

Para alcanzar estos objetivos, se emplea una metodología que incluye varias etapas, como ya vimos la revisión del estado del arte hasta la implementación y pruebas del sistema. Se comienza con el análisis del modelo matemático general del vehículo y su validación mediante simulaciones. Posteriormente, se realizará una descripción técnica del estado actual del prototipo del vehículo, dado que no es construido desde cero, y se planificará la implementación del sistema propuesto. Luego, se seleccionarán los sensores y actuadores necesarios para el monitoreo de las variables críticas. Con esta base, se desarrollará el sistema de control inalámbrico, enfocado en la seguridad y eficiencia de la transmisión de datos entre el vehículo autónomo y la red. Finalmente, se diseñará una interfaz de usuario intuitiva para una interacción efectiva con el sistema y se llevarán a cabo pruebas de campo para validar el funcionamiento de las nuevas características. Además, se incluirán mecanismos para la recolección continua de datos para que estos datos puedan ser utilizados en futuros estudios de Machine Learning e Inteligencia Artificial.



1.9. Alcances del proyecto y Limitaciones

El proyecto "Asistente inteligente para movilidad efectiva, gestión energética y efectividad de servicio para un vehículo eléctrico autónomo" presenta las siguientes limitaciones:

- **Chasis y Motores:** El chasis y los motores de corriente continua de imanes permanentes ya están contruidos y no pueden ser modificados.
- **Control y Comunicación:** La comunicación basada en Raspberry Pi 4 (Computador de bajo consumo energético y tamaño reducido) tiene limitaciones de latencia, afectando la respuesta en tiempo real para tareas complejas.
- **Enfoque del Estudio:** No se realizará un estudio profundo de los motores y baterías porque no es el enfoque principal solo se abordarán para la implementación y operación básica.
- **Sensores y Actuadores:** Se utilizarán sensores y actuadores de bajo costo, lo que puede limitar la precisión y fiabilidad de los datos.

- **Asistente Inteligente:** El término "asistente inteligente" no implica el uso de inteligencia artificial , sino que hace referencia a un sistema capaz de ofrecer soporte y/o realizar tareas automatizadas a través de comandos predefinidos o respuestas programadas .
- **Protocolos y Servicios:** Se emplearán soluciones gratuitas ,de licencia libre o de bajo costo para protocolos de comunicación ,bases de datos y almacenamiento en la nube, lo que puede limitar la robustez y escalabilidad del sistema en algunas características .

Alcances comprometidos

- **Modelo Matemático y Simulación de la dinámica del vehículo autónomo (eléctrica y mecánica):** Basado en literatura estándar y validado en Matlab.
- **Evaluación de Protocolos:** Se evaluarán protocolos de comunicación inalámbrica para determinar la opción más eficiente y segura.
- **Prototipo de Interfaz de Usuario:** Con el objeto de monitoreo y control remoto en “tiempo real” se desarrolla una interfaz gráfica de usuario que permita verificar la funcionalidad del sistema propuesto.
- **Pruebas de Campo:** Validación del sistema en la infraestructura existente , haciendo pruebas de manejo de manera remota y comprobando el control de dirección , velocidad y giro del vehículo . A la vez monitorear la velocidad , corriente de los motores , voltaje de las baterías ,distancia con los objetos cercanos , aceleración en diferentes ejes del vehículo y la transmisión de las imágenes que muestra la cámara .

Estas limitaciones establecen un marco claro a las capacidades del proyecto, evitando expectativas no alineadas con los objetivos definidos.

También dada las limitaciones de la longitud del este documento solo se agregara intrusiones de código necesarias para expresar objetos claves y el resto ira a Anexos

2. Prototipo de vehículo eléctrico autónomo

2.1. Introducción

El análisis inicial del vehículo autónomo comienza con una evaluación detallada del chasis y los motores, componentes fundamentales para su funcionamiento. El chasis está construido principalmente de aluminio, con elementos adicionales hechos de policarbonato y acero para mayor robustez y funcionalidad. El vehículo usa 4 motores con accionamiento independiente los motores son cruciales para la movilidad del vehículo estos son de 90W Maxon MOTOR DC 199216.

El chasis del vehículo está compuesto por varios perfiles y componentes de aluminio que proporcionan una estructura ligera pero resistente. Los componentes principales incluyen:

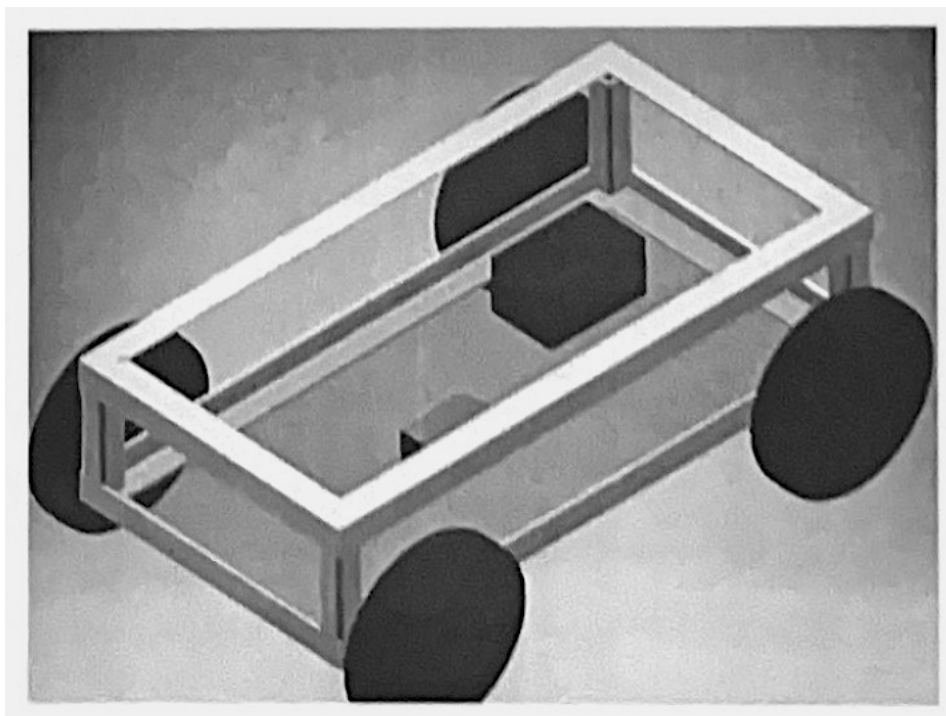


Ilustración 1 Modelo 3d de la Estructura del VEA

Componentes y Masas

Componente	Material	Masa (kg)
Perfiles Chasis T	Aluminio 2.54 mm	1.08
Piso Chasis	Aluminio 1 mm	0.354
Bandejas	Aluminio 3 mm	0.365
Barras	Aluminio 10x10 mm	1.232
Tapas	Policarbonato 1 mm	0.878
Varillas eje	Acero 5 mm	0.093
Bloques de Sujeción		0.857
Ruedas y Llantas		3.00

Tabla 1 Parámetros del Chasis

El motor empleado en este proyecto es el Maxon MOTOR DC 199216, compuesto por tres partes: un motor tipo Graphite Brushes de 90 Watts con un diámetro de 35 mm, una caja reductora Planetary Gearhead GP 32 de 32 mm de diámetro con una reducción de 28:1, y un encoder óptico HEDS 5540. Este conjunto permite obtener velocidades rotacionales más bajas y una mayor precisión en el control de la velocidad y la posición del vehículo.

2.2. Evaluación de la Electronica y Sistemas de Control Existentes

La evaluación del sistema de control basado en microcontroladores reveló una infraestructura compuesta por varios componentes claves. La placa DIEBot1, un circuito electrónico que controla los motores DC, está conformada por 4 puentes H (LMD18200) que permiten el control independiente de 4 motores DC, y 8 optoacopladores duales (HCPL-2630) que aíslan la sección de potencia de la sección de control. Las señales de control provinieron de un computador Raspberry Pi 2 B, ahora reemplazada debido a que no tenía la capacidad para procesar toda la información por un Raspberry Pi 4B, son procesadas por estos puentes H para manejar la velocidad y dirección de los motores. El sistema cuenta con un microcontrolador ATmega 2560, que ayuda en la gestión y procesamiento de señales. El sistema incluye una fuente de 5V para alimentar la Raspberry Pi y otros componentes electrónicos.

La comunicación entre la Raspberry Pi y los microcontroladores se realiza mediante UART, lo que asegura una baja latencia y alta fiabilidad en la transmisión de datos. Sin embargo, la comunicación implementada presenta limitaciones debido a la falta de sensores operativos, excepto los encoders de los motores. Estos encoders proporcionan información crucial sobre la posición y velocidad de las

ruedas, pero el sistema carece de otros sensores funcionales necesarios para una operación autónoma completa.

La ilustración 2 explica el funcionamiento a grandes rasgos del Puente H en donde los sw y diodos representan al Driver LMD18200 , la fuente VDC representa a las baterías y el MDC al motor de corriente continua que trae el Auto.

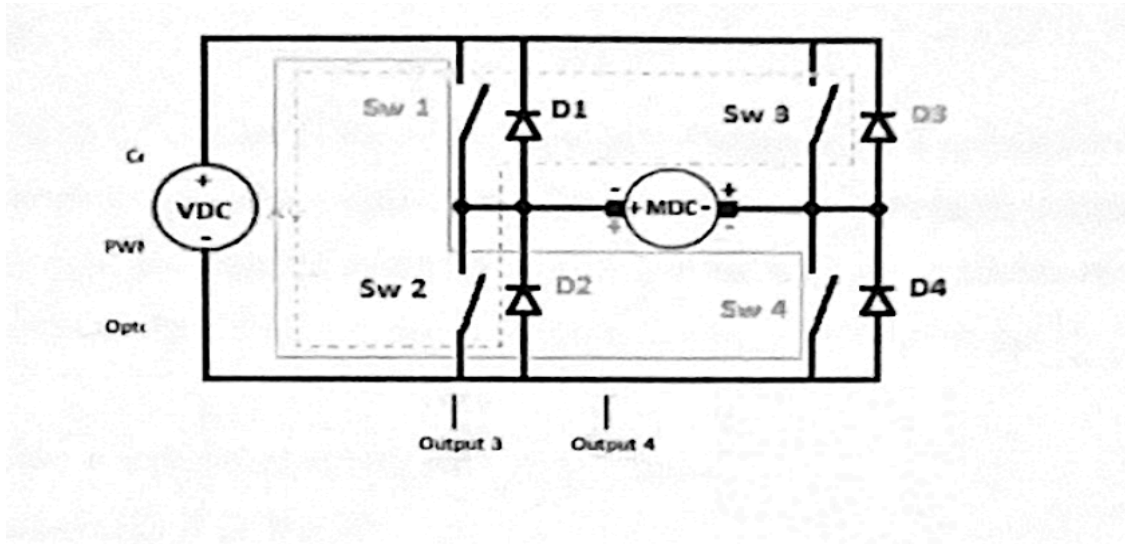


Ilustración 2 Placa DIE Puente H

La ilustración 3 nos presenta la forma física del Driver LMD18200 con cada uno de sus pines

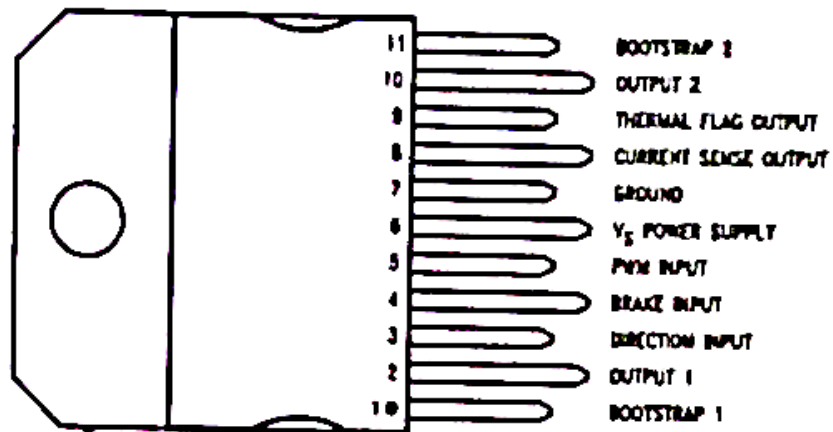


Ilustración 3 LMD18200

La ilustración 4 nos presenta el diagrama electrónico de Driver LMD18200 con cada uno de sus pines en el se puede apreciar la composición de este que tiene módulos lógicos que controlan 4 MOSFET que son los encargados de controlar el voltaje de los motores

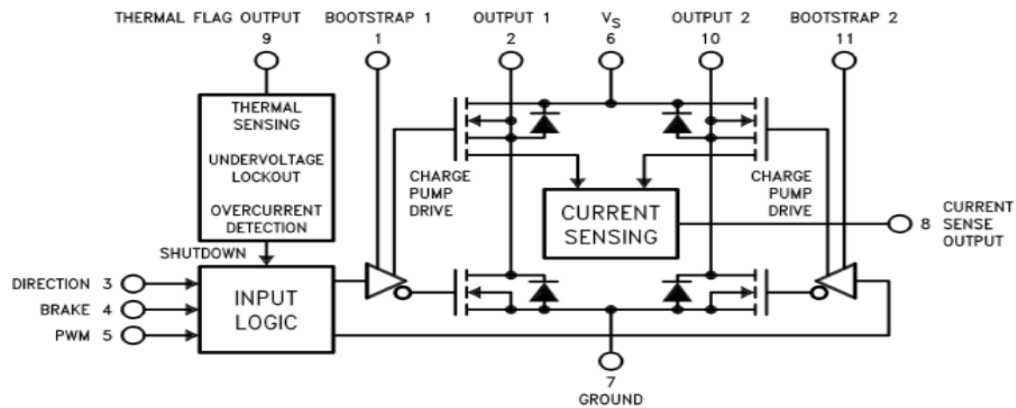


Ilustración 4 Placa DIE Puente H

La ilustración 5 nos presenta la placa DIEbot existente que es la encargada de controlar el voltaje que le llega a los motores mediante una PWM enviada desde un microcontrolador

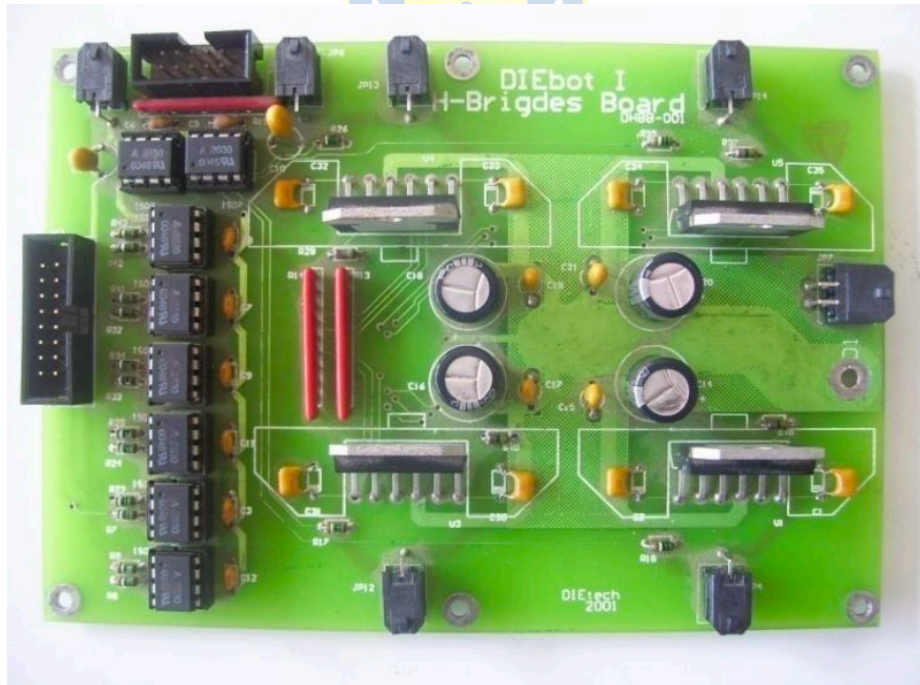


Ilustración 5 Placa DIE Puente H

La ilustración 6 muestra el circuito completo de la placa DIEbot, que consta principalmente de optoacopladores y un puente H necesario para controlar el motor de corriente continua (DC). En esta imagen se representa solo el circuito de un driver; la placa cuenta con cuatro drivers de este tipo.

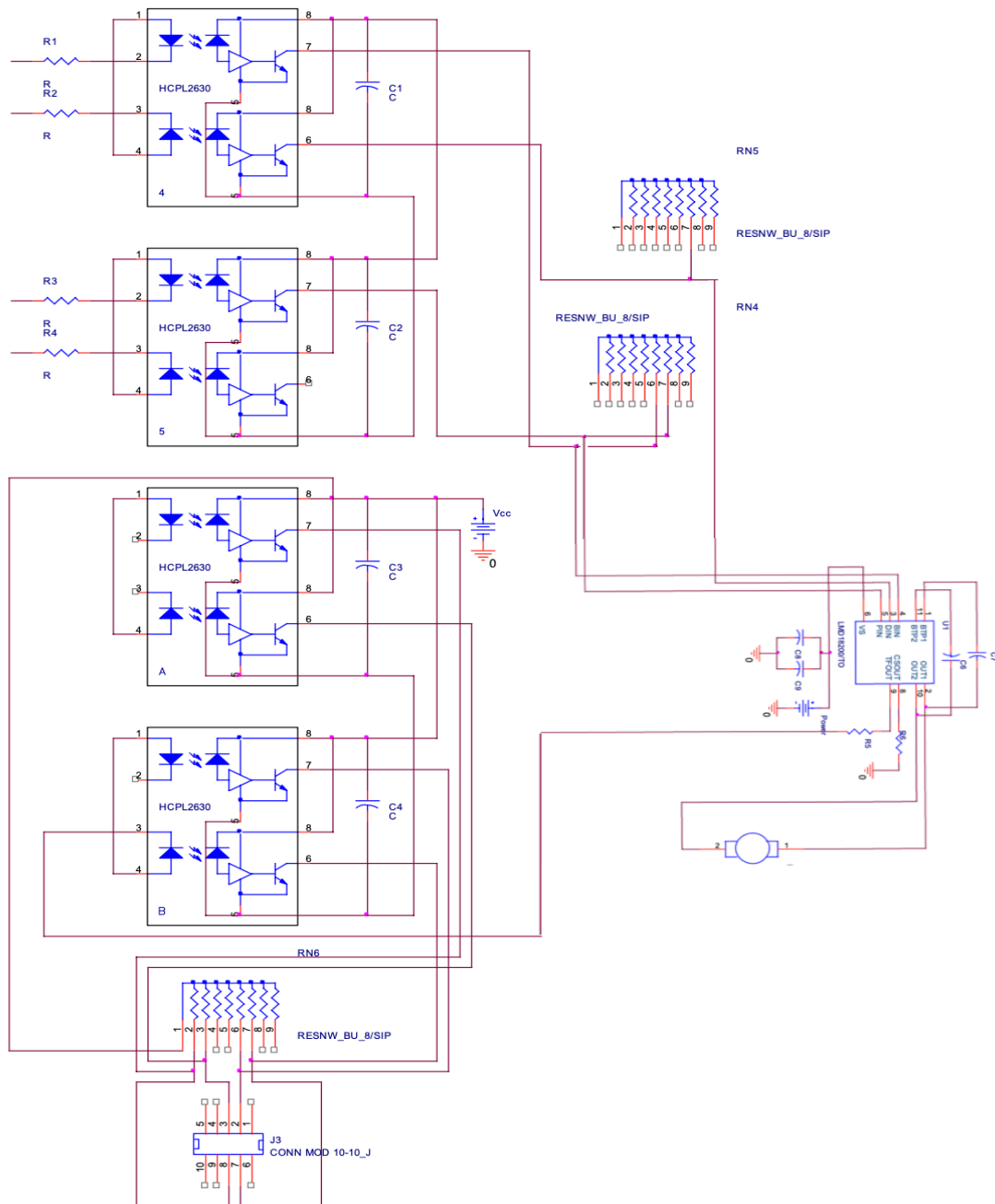


Ilustración 6 Placa DIE Puentes H

Para hacer funcionar este circuito, se conectan los 4 pines de entrada a un microcontrolador ATmega 2560, que se encarga de generar los pulsos necesarios para su operación. Gracias a la arquitectura de este microcontrolador, es posible utilizar funciones nativas que, mediante código, permiten controlar con precisión los pulsos generados. La configuración y control del sistema están centralizados en una Raspberry Pi, que envía comandos predefinidos a través de comunicación serial. Estos mensajes tienen una estructura específica en el formato “XXXXXX”, donde las primeras cuatro “X” indican la dirección deseada[0 o 1], y las dos últimas representan el ancho del pulso PWM requerido para el circuito[0 a 255]. Esto se repite para los 4 motores por lo que tenemos 16 pines conectados al microcontrolador.

<i>Rangos Máximos</i>	
<i>Voltaje de Alimentación</i>	60[V]
<i>Voltaje en Pines 3,4,5</i>	12[V]
<i>Corriente de Salida Peak (200 ms)</i>	6[A]
<i>Corriente de salida continua</i>	3[A]
<i>Potencia disipada</i>	25[W]
<i>Rangos de Operación</i>	
<i>Voltaje de Alimentación</i>	+12[V] min. +55[V] máx.
<i>Características eléctricas</i>	
<i>Periodo mínimo de la señal de control</i>	1 [μs]
<i>Nivel de voltaje bajo de la señal de control</i>	-0.1 [V] min. 0.8 [V] máx.
<i>Nivel de voltaje alto de la señal de control</i>	2[V] min. 12[V] máx.

Tabla 2 Parámetros del Puente H LMD18200

Análisis de los Sensores y Actuadores Disponibles

Actualmente, el vehículo solo cuenta con encoders ópticos HEDS 5540 en los motores para la medición de la velocidad y posición. No hay otros sensores funcionales instalados en el sistema. Los encoders han demostrado ser precisos y fiables en la obtención de datos sobre la velocidad rotacional y la posición del motor. Sin embargo, la falta de otros sensores limita la capacidad del vehículo para interactuar con su entorno y tomar decisiones autónomas.

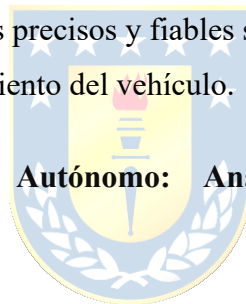
Listado y funcionalidad de los sensores y actuadores instalados:

- **Encoders Ópticos HEDS 5540:** Utilizados para medir la velocidad y la posición de los motores.

Precisión y fiabilidad de los datos obtenidos con los sensores actuales:

- Los encoders proporcionan datos precisos y fiables sobre la velocidad y la posición, lo que es crucial para el control de movimiento del vehículo.

Evaluación Inicial del Vehículo Autónomo: Análisis de Diseño, Componentes y Recomendaciones para Mejoras



La evaluación inicial del vehículo autónomo revela un diseño sólido y una implementación técnica robusta, aunque con algunas áreas que requieren atención y mejoras. El chasis, fabricado principalmente en aluminio con componentes de policarbonato y acero, proporciona una estructura ligera pero resistente, adecuada para soportar las condiciones de operación del vehículo. Sin embargo, se han identificado signos de desgaste y oxidación en algunos componentes, lo cual podría comprometer la durabilidad a largo plazo.

Los motores de corriente continua con imanes permanentes, específicamente el modelo Maxon MOTOR DC 199216, son adecuados para las necesidades del proyecto, proporcionando un buen equilibrio entre potencia y control de precisión gracias a la combinación con la caja reductora y los encoders ópticos.

El sistema de control electrónico, basado en la placa DIEBot1 y la comunicación UART entre la Raspberry Pi y los microcontroladores, demuestra una infraestructura adecuada para el control de los motores y la gestión de señales. La sustitución de la Raspberry Pi 2 por una nueva fue una medida necesaria debido a que no tenía la capacidad para procesar toda la información, al igual que la reparación del multiplexor de cámaras para restaurar la capacidad de visión del vehículo.

La limitación más significativa es la falta de sensores adicionales más allá de los encoders ópticos, lo que restringe la capacidad del vehículo para interactuar de manera autónoma con su entorno. Para mejorar la funcionalidad y la autonomía del vehículo, será crucial incorporar sensores adicionales que proporcionen datos sobre el entorno y permitan una toma de decisiones más informada.

Aunque el diseño y los componentes actuales del vehículo autónomo son adecuados para las necesidades básicas del proyecto, se requiere atención en áreas clave como la actualización y reparación de componentes electrónicos y la incorporación de sensores adicionales para lograr una operación autónoma más completa y eficiente.



3. Modelo Matemático del Vehículo Autónomo

3.1. Introducción

El sistema de accionamiento eléctrico de un vehículo autónomo consiste en varios componentes fundamentales que trabajan juntos para garantizar un rendimiento eficiente y fiable. Este vehículo está diseñado con motores de corriente continua de imanes permanentes (DC-PM) en cada una de sus cuatro ruedas, proporcionando una tracción individual y controlada.

El modelo matemático del vehículo autónomo se construye sobre la base de la dinámica del vehículo, la caracterización de los motores eléctricos y la integración de los componentes del sistema, incluyendo los componentes mecánicos que el auto dispone (Ruedas, cajas reductoras, Masa de Vehículo). Este modelo es esencial para predecir y analizar el comportamiento del vehículo bajo diversas condiciones operativas, y se valida mediante simulaciones precisas en software especializado como MATLAB/Simulink/SimScape.

Objetivos

Los objetivos del modelado matemático se centran en el desempeño dinámico del vehículo autónomo y considera lo siguiente:

- **Comportamiento dinámico del vehículo:** Esto incluye la modelación de la cinemática diferencial y la dinámica del vehículo, considerando las fuerzas actuantes como la fricción, la resistencia al aire y las fuerzas de tracción de los motores.
- **Motores de corriente continua:** Describir ecuaciones básicas que describen el comportamiento eléctrico y mecánico de los motores DC-PM, asegurando que se tengan los parámetros necesarios para que la simulación funcione correctamente.
- **Integrar los componentes del sistema:** Desarrollar un modelo que combine los motores con la parte mecánica que debe movilizar los motores y el sistema de control. Esto con el objetivo de permitir la simulación del sistema completo y la evaluación de su rendimiento bajo diferentes escenarios.



3.2. Dinámica del Vehículo Autónomo

El modelo cinemático diferencial del vehículo autónomo a estudiar en este caso considera la tracción en las cuatro ruedas, permitiendo un control individualizado de cada rueda para mejorar la maniobrabilidad y la estabilidad en los giros. Las velocidades de las ruedas se utilizan para definir el movimiento y la dirección del vehículo.

Para que el vehículo se desplace en línea recta, las velocidades de las cuatro ruedas deben ser iguales. La velocidad media (V) del vehículo se puede calcular como:

$$V = \frac{R}{4} \sum_{i=1}^4 V_i \quad (1)$$

Donde:

- R es el radio de las ruedas.
- V_i son las velocidades de cada una de las cuatro ruedas.

Para el desplazamiento en su propio eje, cuando hay una diferencia entre las velocidades de las ruedas de la derecha e izquierda, la velocidad angular (ω) de las ruedas del vehículo se puede expresar como:

$$\omega = R(V_{r_superior} + V_{r_inferior} - V_{l_superior} - V_{l_inferior}) \quad (2)$$

Donde:

- $V_{r_superior}$, $V_{r_inferior}$, $V_{l_superior}$, $V_{l_inferior}$ son las velocidades de las ruedas

$$\dot{x} = v * \cos \varphi + d * \sin \varphi \quad (3)$$

$$\dot{y} = v * \sin \varphi - d * \cos \varphi \quad (4)$$

$$\dot{\varphi} = \omega \quad (5)$$

Estas ecuaciones se pueden reescribir en forma matricial como:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos \varphi & d * \sin \varphi \\ \sin \varphi & -d * \cos \varphi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (6)$$

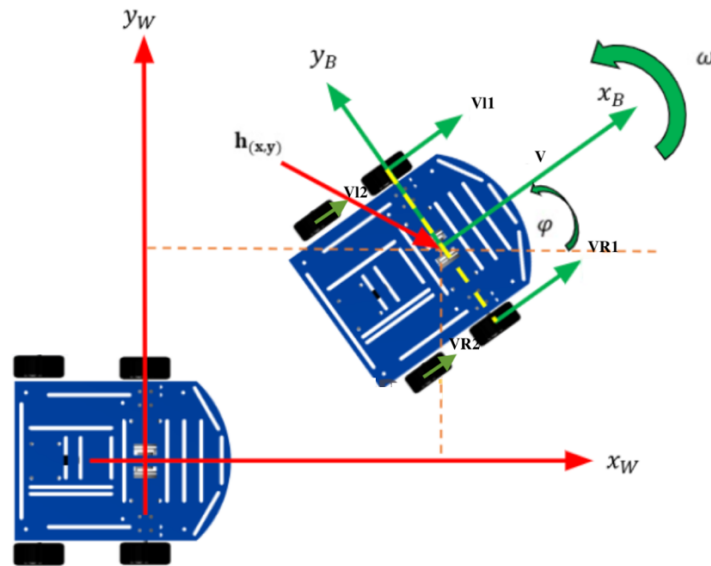


Ilustración 7 Modelo Cinemático Diferencial del Automóvil

Donde la matriz de velocidad lineal y angular se define como:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{R}{4} & \frac{R}{4} & \frac{R}{4} & \frac{R}{4} \\ R \cdot \cos \varphi & R \cdot \cos \varphi & -R \cdot \cos \varphi & -R \cdot \cos \varphi \\ \frac{L}{L} & \frac{L}{L} & \frac{L}{L} & \frac{L}{L} \end{bmatrix} \quad (7)$$

Donde: R es el Radio de la rueda (15 cm), L es la longitud entre ruedas (35 cm) , d es la distancia al centro (30 cm) y φ es la dirección de la rueda

Haciendo pruebas para los movimientos más comunes. El **movimiento hacia adelante** se logra cuando todas las ruedas giran en la misma dirección y con la misma velocidad angular , lo que resulta en un desplazamiento uniforme al verificar su combinación en la matriz correspondiente como podemos ver en la ilustración 9. El **movimiento hacia atrás** es similar en la ilustración 10, pero con las ruedas girando en sentido opuesto, manteniendo la misma velocidad angular en valor absoluto, pero con signo negativo. Para un **giro a la derecha** podemos ver la ilustración 11, las ruedas de un lado giran hacia adelante y las del otro hacia atrás, generando un momento angular hacia la derecha debido a la contribución opuesta de los términos relacionados con $\frac{R \cdot \cos \theta}{L}$. Invertir este patrón de giro produce un **giro a la izquierda** como vemos en la ilustración 12, donde las ruedas generan un momento angular hacia el lado contrario.

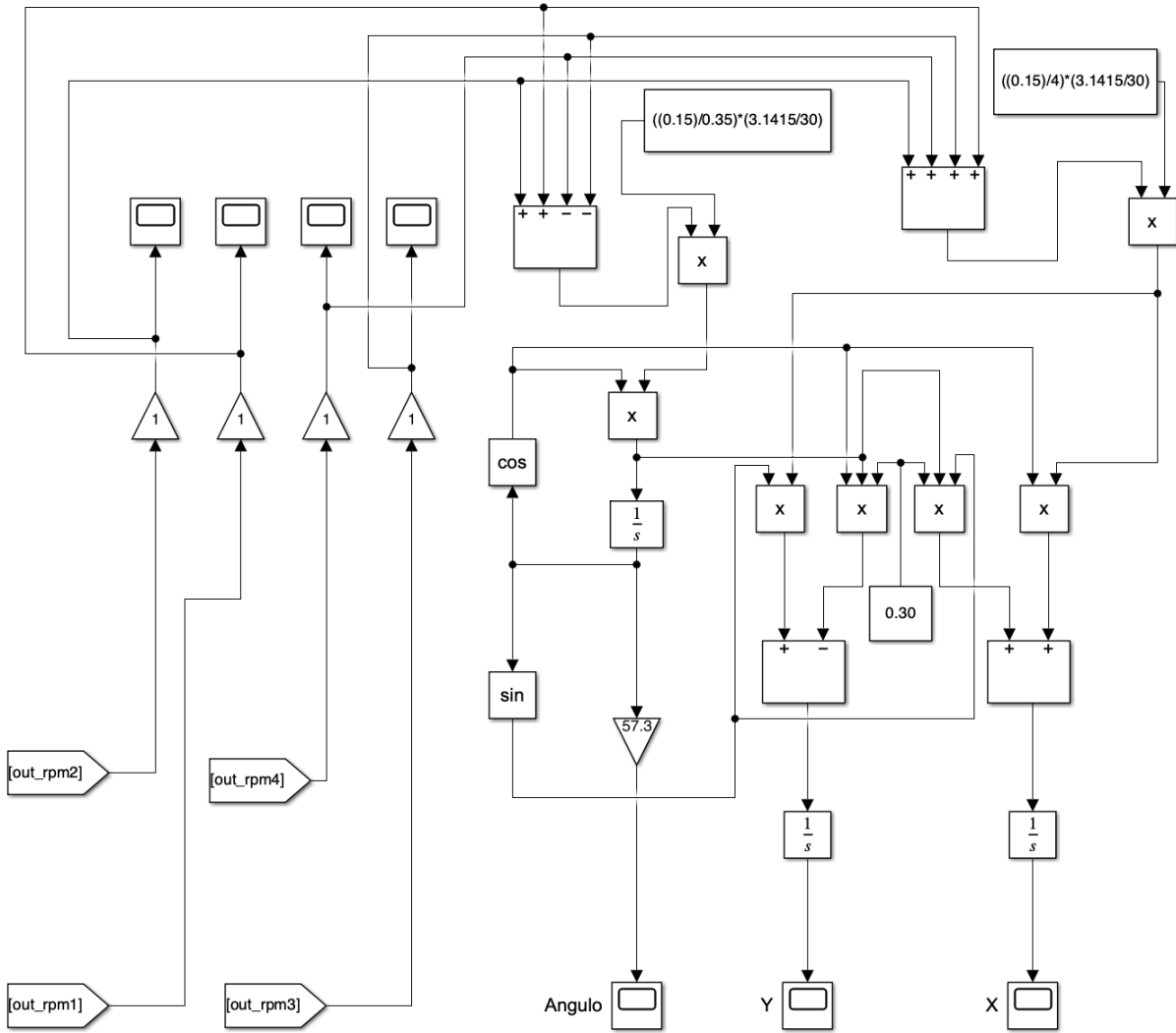


Ilustración 8 Diagrama de Bloque Matlab/Simulink Movimiento Cinemático Diferencial

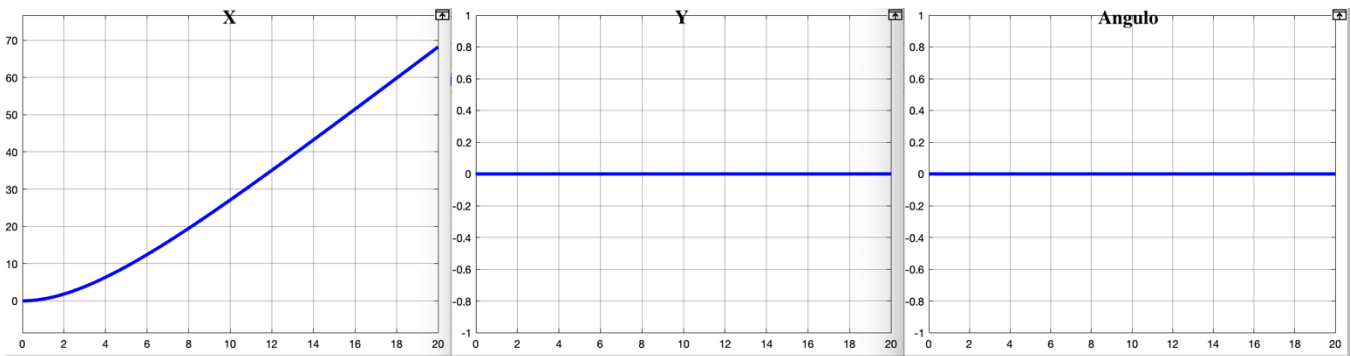


Ilustración 9 Movimiento hacia adelante Distancia vs Tiempo

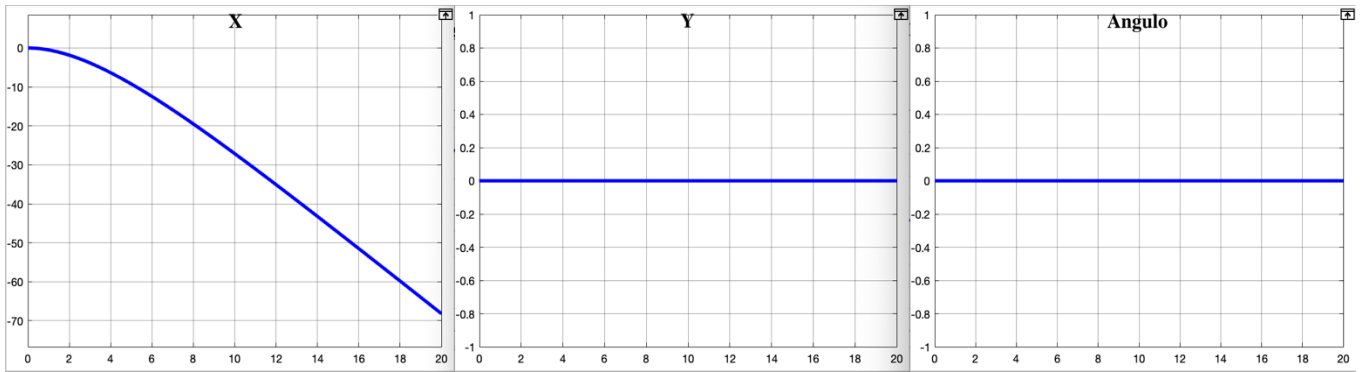


Ilustración 10 Movimiento hacia atrás Distancia vs Tiempo

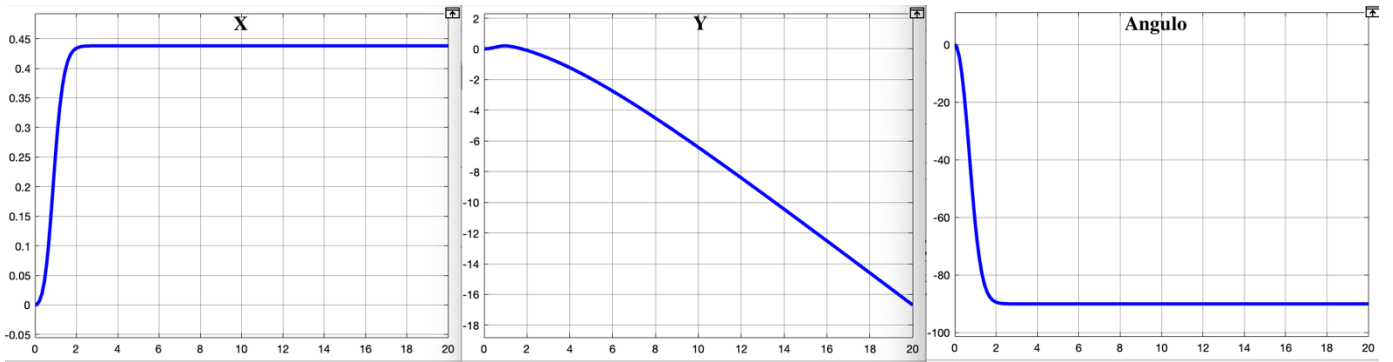


Ilustración 11 Movimiento con giro a la derecha Distancia vs Tiempo

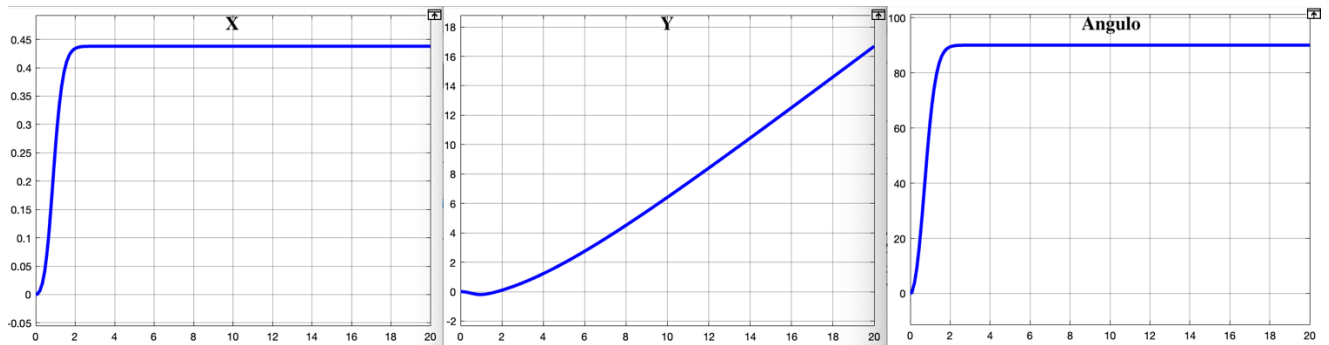


Ilustración 12 Movimiento con giro a la izquierda Distancia vs Tiempo

3.3. Modelado del Motor de Corriente Continua

El motor de corriente continua (DC) de imanes permanentes utilizado en el vehículo autónomo puede ser descrito por un conjunto de ecuaciones que representan tanto el circuito eléctrico como el comportamiento mecánico del motor. Como podemos apreciar en la Ilustración 7. Este es necesario para calcular algunos parámetros que no aparecen en la hoja de datos .

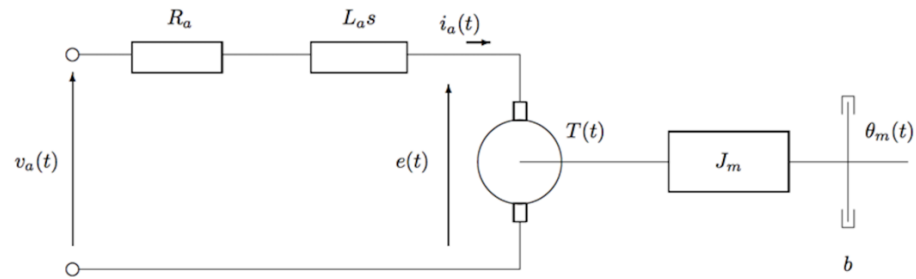


Ilustración 13 Modelo del Motor

Ecuación de la tensión en el circuito eléctrico del motor

$$V_a = L_a \frac{di_a}{dt} + R_a i_a + e_b \quad (14)$$

donde:

- V_a es la tensión aplicada al armadura del motor (voltios).
- L_a es la inductancia de la armadura (henrios).
- R_a es la resistencia de la armadura (ohmios).
- i_a es la corriente de la armadura (amperios).
- e_b es la fuerza contraelectromotriz (fem) generada por el motor (voltios).

Ecuación de la fuerza contraelectromotriz:

$$e_b = K\Phi\omega_m \quad (15)$$

donde:

- $K\Phi$ es la constante de la fuerza contraelectromotriz (voltios por radian por segundo).
- ω_m es la velocidad angular del motor (radianes por segundo).

Ecuación del par motor:

$$T_m = K_t i_a \quad (16)$$

donde:

- T_m es el par generado por el motor (newton-metros).
- K_t es la constante de torque (newton-metros por amperio).

Ecuación de la dinámica rotacional:

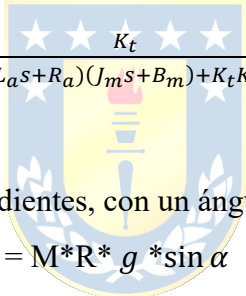
$$T_m - T_L = J_m \frac{d\omega_m}{dt} + B_m \omega_m \quad (17)$$

donde:

- T_L es el torque de carga (newton-metros).
- J_m es el momento de inercia del rotor (kilogramos por metro cuadrado).
- B_m es el coeficiente de fricción viscosa del motor (newton-metros por radian por segundo).

Función de Transferencia del Motor :

La función de transferencia del motor que relaciona la tensión aplicada V_a con la velocidad angular ω_m es:


$$\frac{\omega_m}{V_a} = \frac{K_t}{(L_a s + R_a)(J_m s + B_m) + K_t K \phi} \quad (18)$$

Torque potencial:

Es el torque asociado a subir o bajar pendientes, con un ángulo de inclinación α .

$$T_{pot} = M * R * g * \sin \alpha \quad (19)$$

donde:

- T_{pot} es el torque de potencial(newton-metros).
- M es la Masa del cuerpo (kilogramos).
- R es el radio de la rueda (metros).
- g es la aceleración de la gravedad (metros/segundos al cuadrado)

Torque de fricción:

Es el torque producido por la fricción de las ruedas, esta está asociada a la fuerza de roce

$$T_{fr} = C_{rr} * M * R * g * \cos \alpha \quad (20)$$

donde:

- T_{fr} es el torque de fricción(newton-metros).
- M es la Masa del cuerpo (kilogramos).
- R es el radio de la rueda (metros).
- g es la aceleración de la gravedad (metros/segundos al cuadrado)
- C_{rr} es el coeficiente generado por el terreno

Torque de aceleración:

Es el torque asociado a acelerar una masa M

$$T_{dyn} = \frac{d\omega_{motor(t)}}{dt} * (J + R * M) \quad (21)$$

Donde:

- $\omega_{motor(t)}$ es la velocidad angular del motor (radianes / segundos)

Torque de carga total:

El torque total para el motor es

$$T_l = T_{pot} + T_{fr} + T_{dyn} \quad (22)$$

Ecuaciones de armadura y mecánica:

$$V_a(s) = (sL_a + R_a)i_a(s) + K\Phi * \omega_m(s) \quad (23)$$

$$sJ_m * \omega_m(s) = K\Phi * i_a(s) - B_m * \omega_m(s) - T_l(s) \quad (25)$$

Mediante la combinación de estas ecuaciones y técnicas de análisis, se puede desarrollar un modelo preciso del motor de corriente continua que permita su integración en el sistema de control del vehículo autónomo, optimizando su rendimiento y eficiencia. En la tabla 3 se puede ver datos calculados necesarios para la simulación del modelo .

Parámetros	Valores[Unidad]
V_a	24[V]
I_a	3.62[A]
K_T	0.0292[Nm/A]
R_a	0.583[Ω]
L_a	0.191[mH]
T_{nom}	1.200[Nm]
N_{nom}	7000[rpm]
J	7.92[μKgm ²]

Tabla 3 Parámetros calculado y utilizador para el accionante del Motor Maxon DC motor RE 35

3.4. Integración de los componentes del sistema en una simulación

El VEA tiene importantes desafíos uno de ellos es la integración de varios de sus componentes , este tiene componentes eléctricos y mecánicos por lo que para una simulación integral hay que acoplar varios componentes . En la ilustración 15 y 16 tenemos la integración de esto en una simulación gracias a las herramientas que ofrece Simscape de Matlab . Este nos permite agregar el comportamiento de varios componentes que mediante formulación matemática podría ser muy extenso como lo podría ser la relación que tiene con los neumáticos , el cuerpo del vehículo , la caja reductora y los motores DC entre otros .

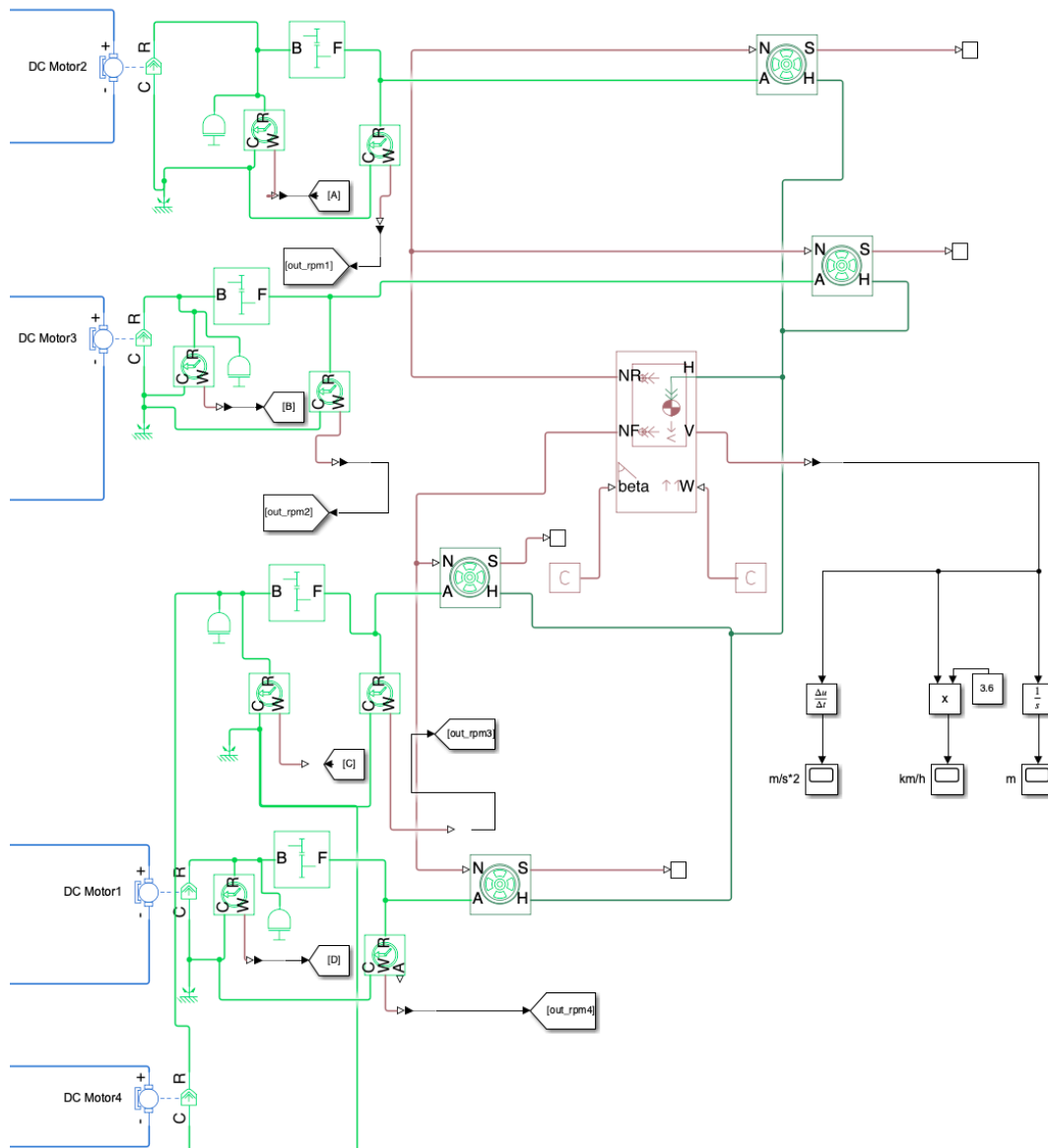


Ilustración 14 Representación del automóvil en SimScape de Matlab

La ilustración 16 representa las señales de alimentación del el Motor DC , esta estructura se repite 4 veces una por cada motor y es posible operarla de forma independiente .

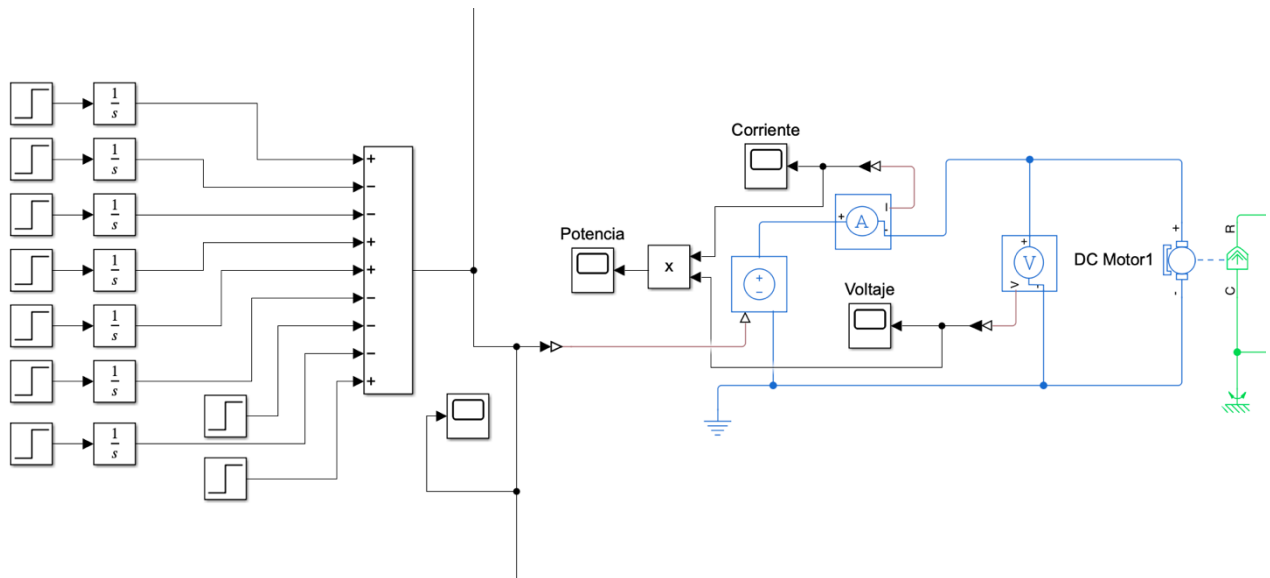


Ilustración 15 Representación del automóvil en Simscape Matlab

En la ilustración 17 podemos ver la señal generada de voltaje para evaluar su comportamiento en diversas situaciones típicas y casos bordes que no queremos que sucedan respecto a distintas señales de voltaje. Aquí tenemos Rampas , valores fijos y escalones .

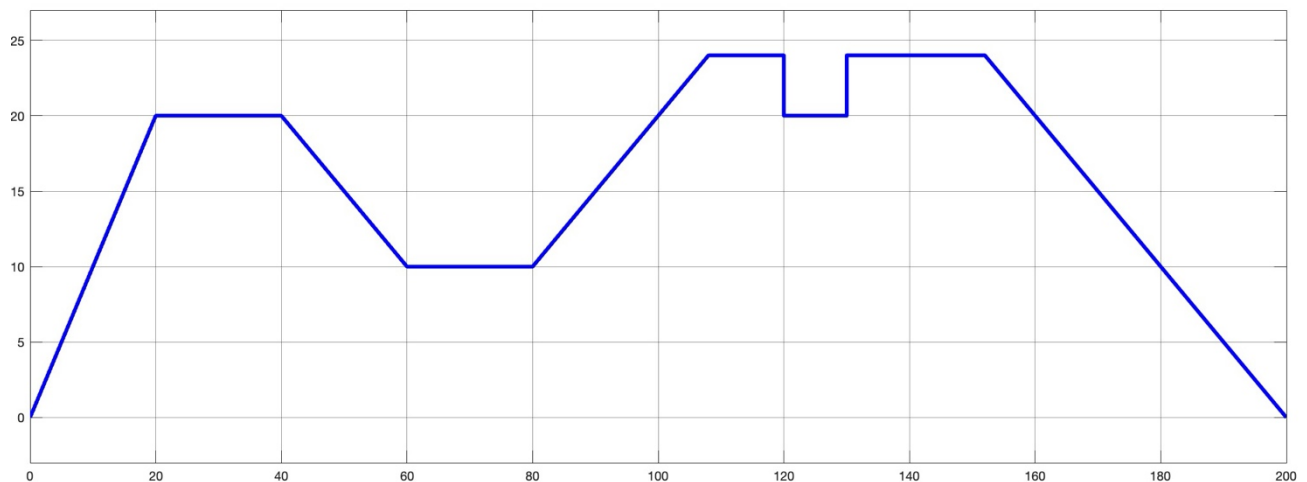


Ilustración 16 Voltaje Motor Vs Tiempo

En la ilustración 18, se observa el comportamiento de la corriente del motor en respuesta a las señales de voltaje aplicadas. Estas señales presentan un perfil característico de rampas y escalones, lo que refleja las variaciones en la magnitud del voltaje aplicado y su impacto en la dinámica del motor.

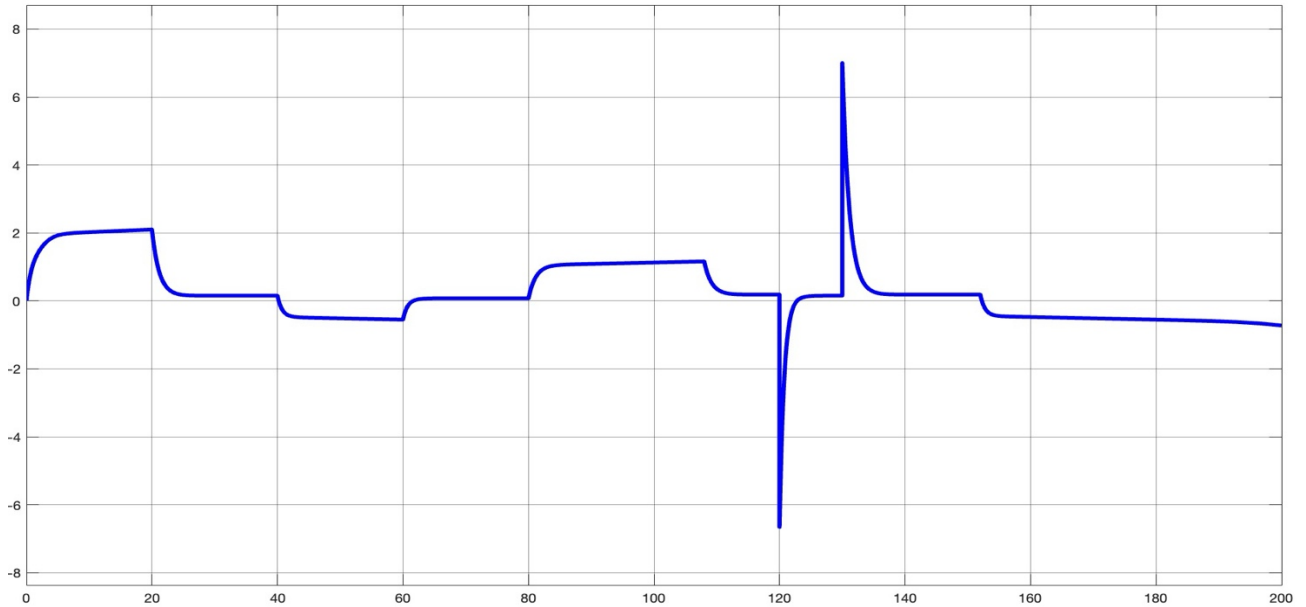


Ilustración 17 Corriente Motor Vs Tiempo

En la ilustración 19 apreciamos cómo se comporta la Potencia del motor en respuesta a las señales de voltaje aplicadas.

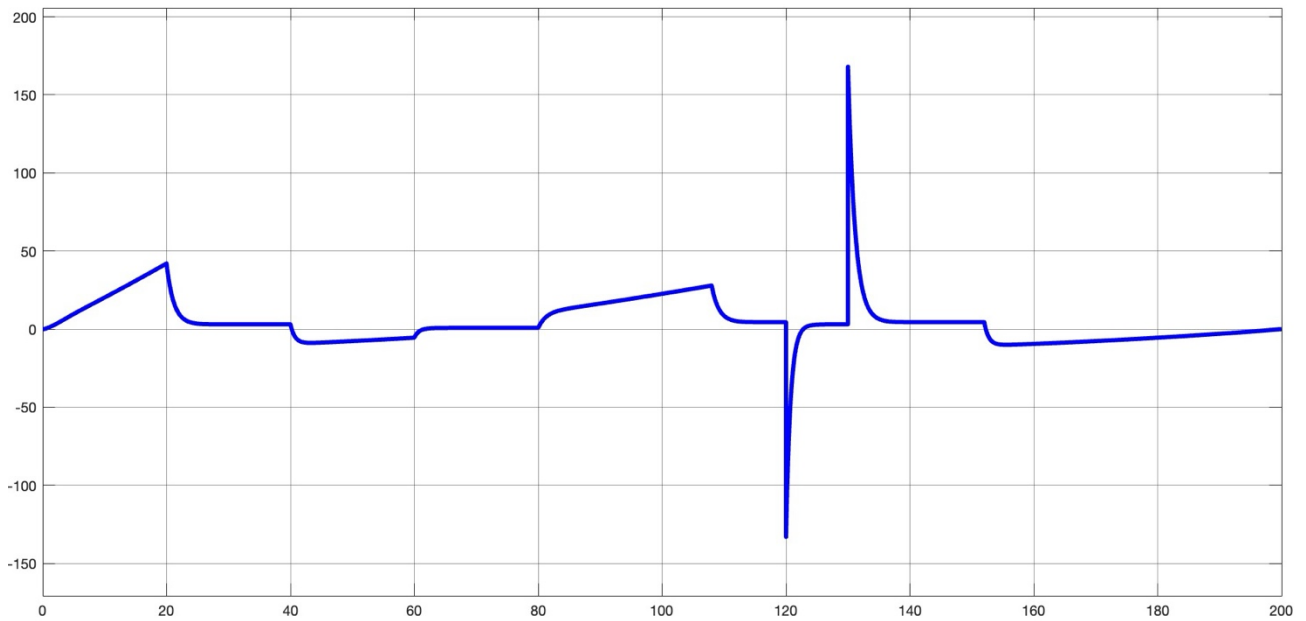


Ilustración 18 Potencia Motor Vs Tiempo

En la ilustración 20 apreciamos la velocidad del Automóvil mediante en respuesta a las señales de voltaje aplicadas.

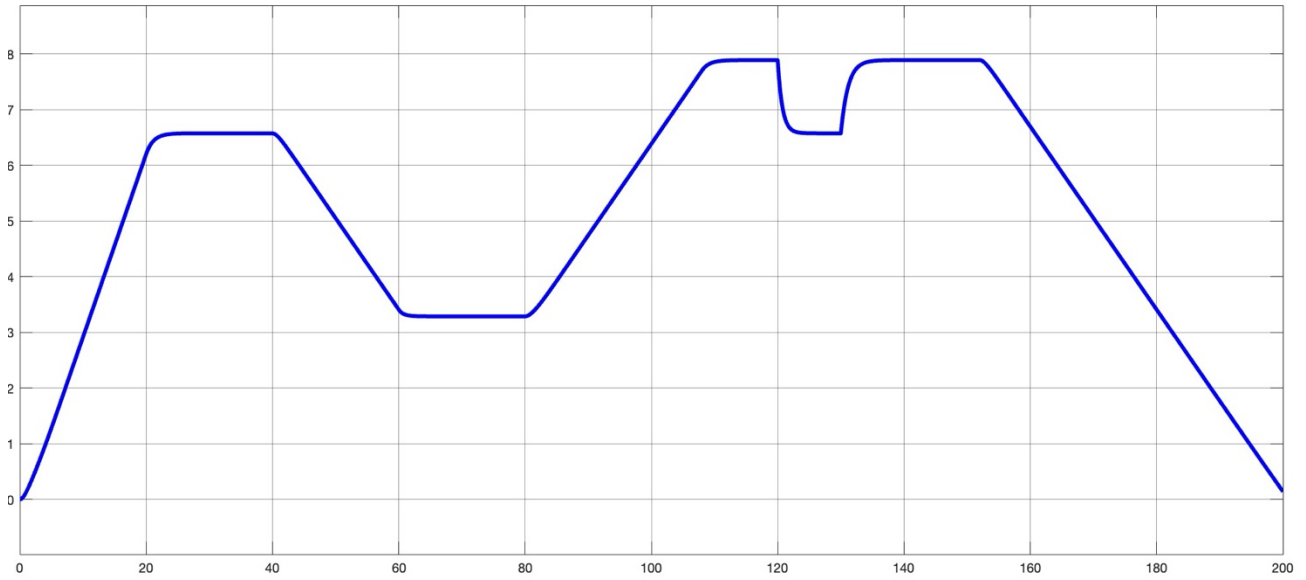


Ilustración 19 Velocidad Automóvil Vs Tiempo

En la ilustración 21 apreciamos la Aceleracion del Automóvil mediante en respuesta a las señales de voltaje aplicadas.

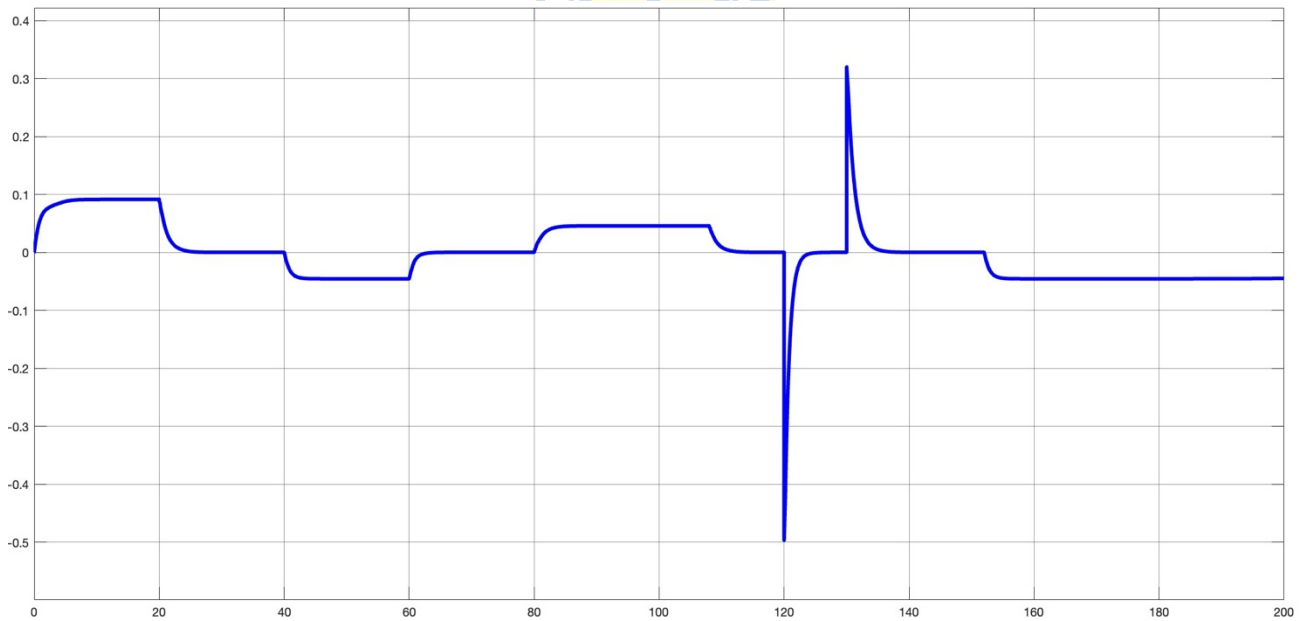


Ilustración 20 Aceleración Automóvil Vs Tiempo

En la ilustración 22 apreciamos la Distancia recorrida del Automóvil mediante en respuesta a las señales de voltaje aplicadas.

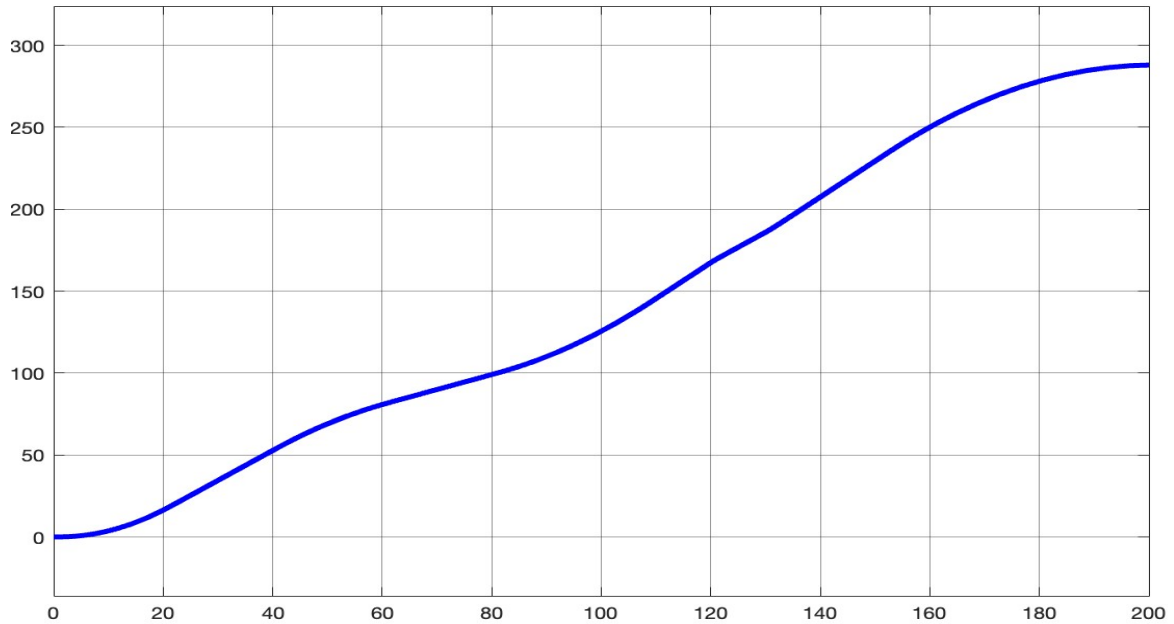


Ilustración 21 Distancia recorrida por Automóvil Vs Tiempo

En la ilustración 23 apreciamos la Modificaciones realizadas al Modelo para hacer la prueba de esfuerzo en escenarios donde la pendiente cambia

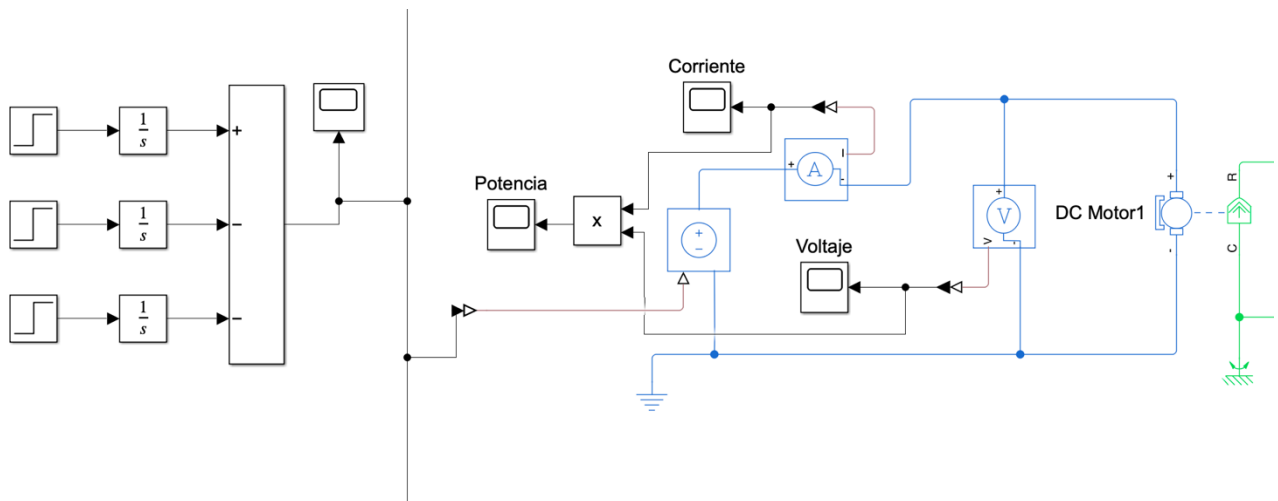


Ilustración 22 Modelo para inclinación

En la ilustración 24 podemos ver la señal generada de voltaje para evaluar su comportamiento en diversas situaciones donde el Angulo de la pendiente cambia .

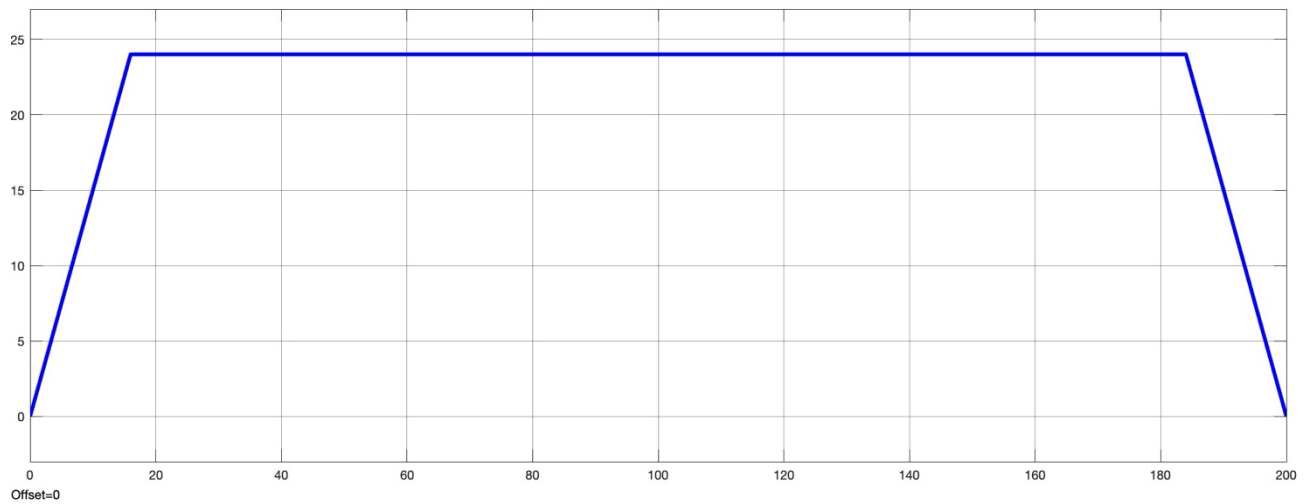


Ilustración 23 Voltaje Motor Vs Tiempo

En la ilustración 25, se observa el comportamiento de la corriente del motor en respuesta distintas pendientes aplicas y su impacto en la dinámica del motor.

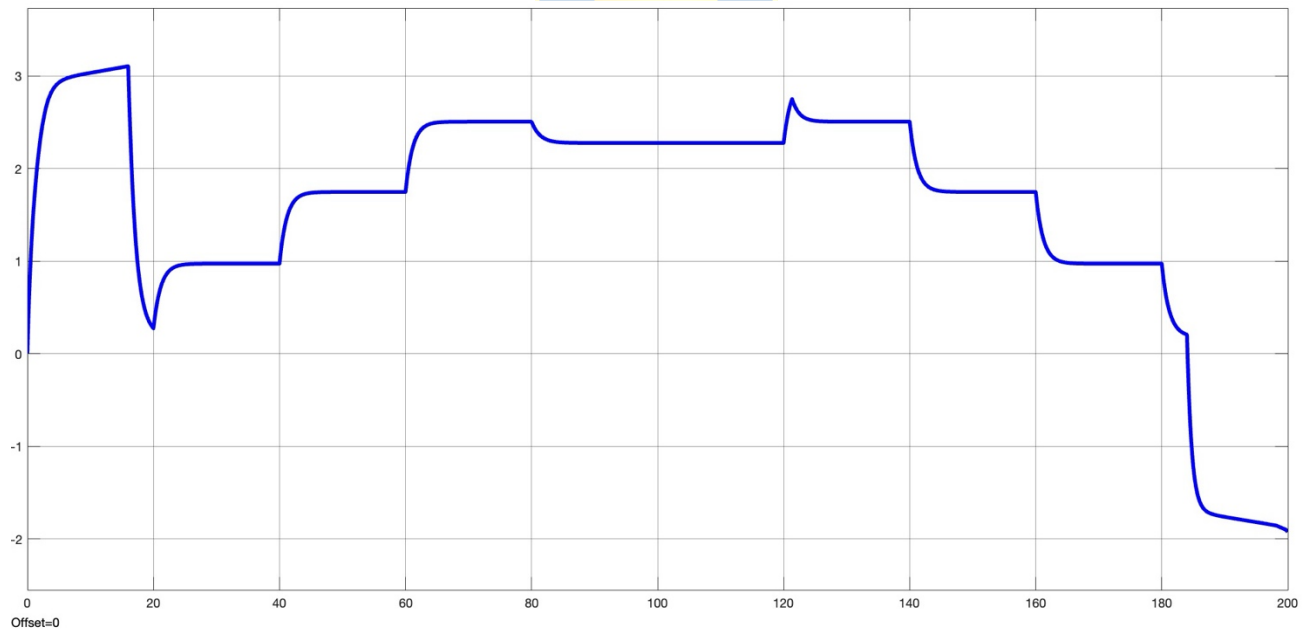


Ilustración 24 Corriente Motor Vs Tiempo

En la ilustración 26, se observa el comportamiento de la potencia del motor en respuesta distintas pendientes aplicas y su impacto en la dinámica del motor.

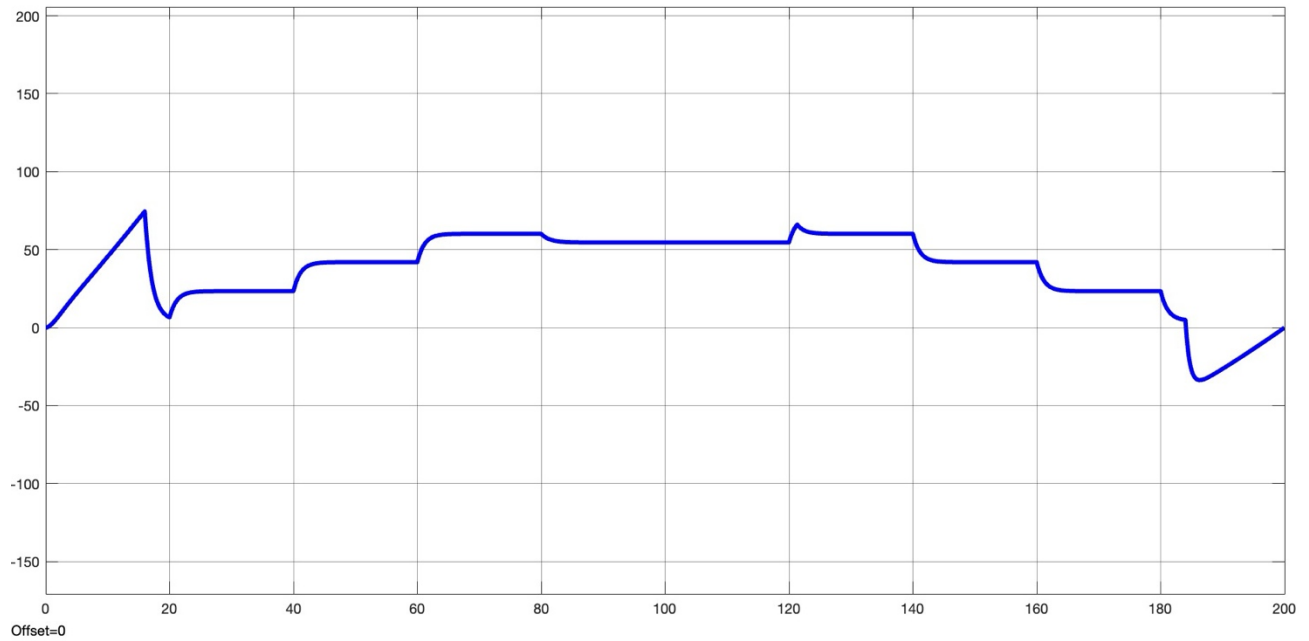


Ilustración 25 Potencia Motor Vs Tiempo

En la ilustración 27, se observa las los distintos ángulos al que la simulación fue espuesta para el cuerpo del automóvil .

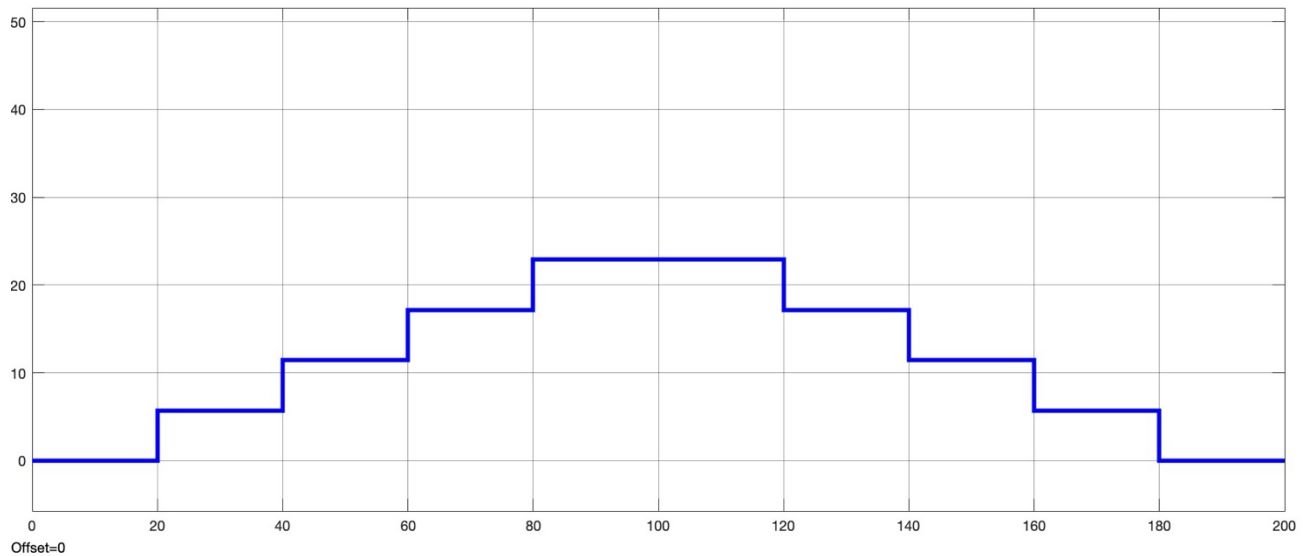


Ilustración 26 Angulo de Inclinación Automóvil Vs Tiempo

En la ilustración 28, se observa el comportamiento de la aceleración del automóvil en respuesta distintas pendientes aplicas

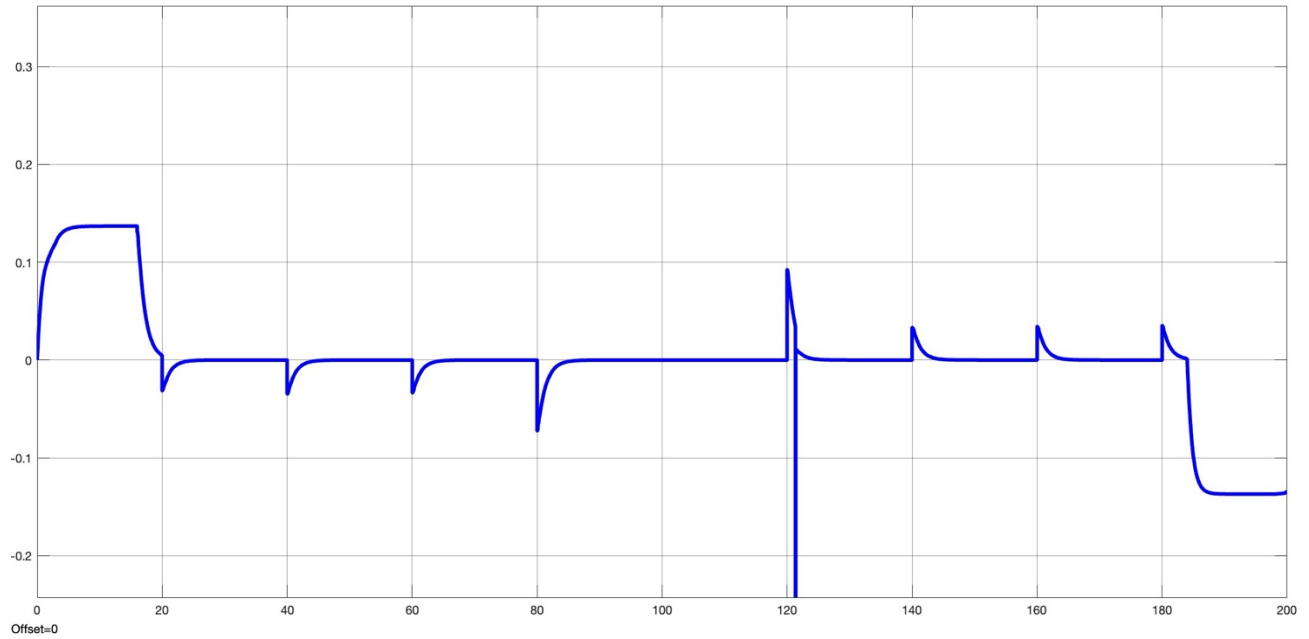


Ilustración 27 Aceleración Automóvil Vs Tiempo

En la ilustración 29, se observa el comportamiento de la velocidad del automóvil en respuesta distintas pendientes aplicas

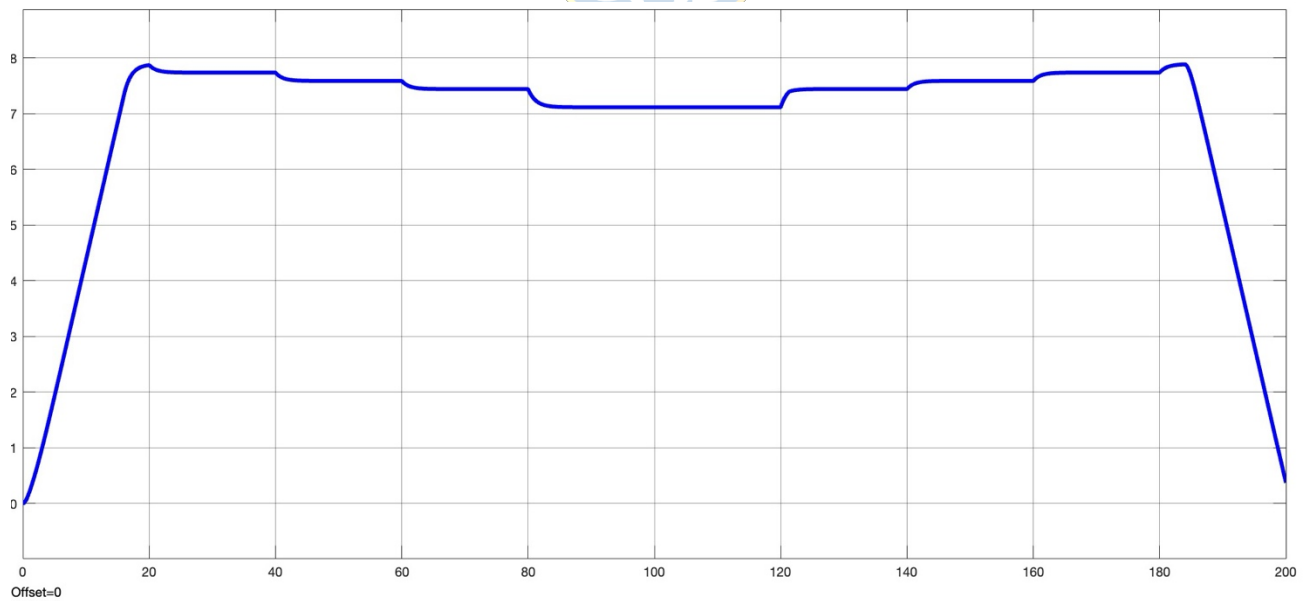


Ilustración 28 Velocidad Automóvil Vs Tiempo

3.5. Validación Experimental

Motor

La validación experimental del modelo matemático del motor de corriente continua implica la realización de pruebas prácticas para comparar los resultados obtenidos en las simulaciones con datos reales.

• Metodología:

Motor y Banco de Baterías:

Se configuró el motor de corriente continua y el banco de baterías de acuerdo con las especificaciones utilizadas en el modelo. El motor se conectó a un banco de pruebas controlado que permite variar la tensión aplicada y la carga del motor. Esto se logró gracias a una fuente DC-DC de alta eficiencia regulable de 7 a 45V , por lo que se puede despreciar las pérdidas de esta.

Sensores y Medidores: Se instalaron equipos especializados para registrar la velocidad angular (ω_m), corriente de armadura (i_a), tensión aplicada (V_a) y par motor (T_a). Los equipos especializados incluyen tacómetros, amperímetros de precisión y sensores de par.

• Condiciones de Prueba:

Pruebas de Arranque: Se realizaron pruebas de arranque del motor aplicando diferentes tensiones de entrada (15V, 24V y 48V) y registrando la respuesta del motor en términos de velocidad angular, corriente y par.

Pruebas de Carga: Se aplicaron diferentes niveles de carga al motor utilizando un freno dinámico, y se registraron los valores de velocidad angular, corriente y par bajo cada condición de carga.

• Procedimiento de Prueba:

Iniciar Pruebas: Se iniciaron las pruebas aplicando la tensión nominal y registrando todos los parámetros relevantes.

Variar Condiciones: Se modificaron la tensión aplicada y las condiciones de carga en pasos controlados, registrando los resultados para cada condición.

Registrar Datos: Se aseguraron de registrar todos los datos necesarios para una comparación exhaustiva con los resultados de la simulación.

Una vez realizadas las pruebas experimentales, se procedió a comparar los resultados con los obtenidos en las simulaciones. De esta comparación en laboratorio se pudo apreciar gran similitud entre las simulaciones y las pruebas . solo habiendo mínimas diferencias que se pueden atribuir a la precisión de los equipos y la perdidas de algunos componente electrónicos .

En la tabla siguiente se pueden apreciar algunos de los datos medidos y luego comparados con el datasheet de fabricante.

Prueba	Tensión nominal (V)	Velocidad (rpm)	Corriente (mA)	Máx. rendimiento (%)
Condiciones de vacío	24	7580	124	86
Condiciones de carga nominal	24	6800	3520	85
Condiciones de vacío	15	7000	240	83
Condiciones de carga nominal	15	6320	3900	82
Condiciones de vacío	48	7000	90	87
Condiciones de carga nominal	48	6400	1700	88

Tabla 4 Resultados pruebas de laboratorio de Motor Maxon DC motor RE 35

Estos resultados validaron el modelo matemático del motor de corriente continua con una alta precisión, reflejando adecuadamente el comportamiento real del sistema.

Prueba de Campo

La metodología de **prueba de arranque**, se evalúa la capacidad del vehículo a diferentes niveles de tensión (15, 20 y 24 V), tanto sin carga extra, registrando el consumo de corriente y la velocidad angular inicial (rpm). La **prueba de velocidad máxima y maniobrabilidad** mide la velocidad promedio y máxima alcanzada en línea recta sobre una distancia de 100 metros en baldosas antideslizantes, y la precisión de giros de 90° y 180° evaluando el grado de desviación. Finalmente, la **prueba de gestión energética** evalúa el consumo de corriente, en pendientes de 5°, 10°, 15°, 20°, lo que permite analizar la eficiencia del sistema en condiciones diversas.

En su mayoría estas pruebas fueron efectuadas en el pasillo del Tecnólogo Mecánico en el piso 2 como se puede apreciar en la ilustración 30 y 31, en ciertas ocasiones se sacó al exterior para ver cómo se comportaba con otro tipo de piso viendo que en cemento igual se comportaba de manera adecuada pero el enfoque principal es en baldosas antideslizantes



Ilustración 29 Prueba de Campo del VEA

El vehículo supero todas las pruebas excepto la de inclinación a 30° lo que hace sentido ya que en las simulaciones sucedió lo mismo ya que sobrepasaba la capacidad del VEA . de igual manera se comporto como fueron las simulaciones de cinemática diferencial y dinámica del vehículo. Se aprecia las dificultad para acelerar y el aumento de la corriente para compensar , luego de estabilizarse la corriente nominal era muy cercana a lo que nos aparecía en la simulación .

3.6. Conclusiones Modelado Matemático

Interpretación de los Resultados Obtenidos

Analizando los resultados obtenidos de las pruebas y simulaciones realizadas sobre el modelo matemático del vehículo eléctrico autónomo. Los resultados son cruciales para evaluar la precisión y la eficacia del modelo desarrollado.

Resultados de la Simulación

Los resultados de la simulación del modelo matemático se compararon con los datos experimentales obtenidos de las pruebas de laboratorio del motor Dc por separado y algunas pruebas de campo con los motores montados en el VEA mas la baterías. La simulación se llevó a cabo utilizando Matlab/Simulink/SimScape, la eficiencia de los motores de corriente continua de imanes permanentes, y la capacidad de respuesta del sistema a cargas en la carrocería y desniveles del camino .

- **Consumo Energético:** El análisis del consumo energético mostró una correlación estrecha entre los datos simulados y los datos experimentales. Bajo condiciones de carga constante, el modelo predijo el consumo energético con un margen de error del bajo.
- **Eficiencia de los Motores:** La eficiencia de los motores, tanto en simulación como en pruebas experimentales, mostró una eficiencia promedio del 85% en el punto de operación que ocuparemos (24V), con variaciones mínimas bajo diferentes condiciones de carga que tiene concordancia con los valores de la hoja de datos . Esto demuestra que los motores seleccionados son adecuados para el propósito del vehículo.
- **Capacidad de Respuesta del Sistema:** La capacidad de respuesta del sistema a cargas en la carrocería y desniveles del camino fue evaluada tanto en simulaciones como en pruebas de campo. La respuesta del sistema mostró una estabilidad adecuada, manteniendo una velocidad angular estable con mínimas fluctuaciones, incluso en terrenos algo irregulares. Si Crr es en Baldosas antideslizantes el ángulo máximo es de 23.938° .

- **Respuesta a Variaciones de Carga y Terreno:** La respuesta del sistema mostró una estabilidad adecuada, manteniendo una velocidad angular estable con mínimas fluctuaciones.
- **Concordancia entre Simulación y Experimentos:** La comparación entre los resultados de la simulación y los datos experimentales permite verificar la precisión del modelo utilizado. Una alta concordancia indica que el modelo simulado es adecuado para predecir el comportamiento real del sistema, lo que valida su uso en análisis futuros y ajustes de parámetros
- **Optimización Energética:** La precisión en la predicción del consumo energético permite una mejor planificación y gestión del uso de la batería del VEA, lo cual es crucial para maximizar la autonomía y eficiencia del vehículo. Los ajustes en el modelo para condiciones de alta demanda ayudarán a evitar subestimaciones del consumo energético y a planificar de manera más efectiva las necesidades de recarga.



4. Ingeniería Conceptual de Operación y Control remoto del Vehículo

4.1. Introducción

El monitoreo de variables es crucial para los vehículos eléctricos autónomos (VEA), asegurando su eficiencia y seguridad. Permite detectar y corregir problemas en tiempo real, optimizando la operación y extendiendo la vida útil de los componentes.

En un entorno autónomo, la supervisión constante permite ajustar parámetros operativos y prevenir fallos, optimizando el uso de la batería y maximizando la autonomía del vehículo. Mediante algoritmos predictivos, se anticipan problemas potenciales, permitiendo implementar medidas preventivas.

Futuras integraciones de inteligencia artificial (IA) mejorarán el análisis de datos, identificando patrones y tendencias para una respuesta más rápida y efectiva ante anomalías. La integración de IA está prevista para trabajos futuros y no para la implementación actual.

La selección de las variables a monitorear se basa en su relevancia para los objetivos del monitoreo y en la capacidad de los sensores para medirlas con precisión y fiabilidad. Cada variable proporciona información crítica que, cuando se analiza en conjunto, ofrece una visión completa del estado y desempeño del vehículo autónomo eléctrico.

4.2. Monitoreo y Selección de Variables

Las Variables a monitorear son las siguientes

- **Velocidad del Vehículo:** Crucial para la gestión del movimiento y la estabilidad del vehículo.
- **Estado de Carga de la Batería:** Fundamental para la gestión energética y la planificación de la autonomía.
- **Corriente de los Motores:** Relacionada con la demanda de potencia y el consumo energético.
- **Posición y Orientación del Vehículo:** Esencial para la navegación y el control del movimiento.
- **Condiciones del Entorno:** Incluyendo factores como la inclinación del terreno y la presencia de obstáculos.

4.3. Identificación de Variables Críticas

Listado de Variables a Monitorear

Para garantizar un funcionamiento óptimo y seguro del vehículo eléctrico autónomo, se deben monitorear las siguientes variables críticas:

- **Imágenes del Entorno**

Justificación: Las cámaras delantera y trasera proporcionan información visual del entorno del vehículo, lo que es crucial para la navegación autónoma y la detección de obstáculos. Monitorear y procesar estas imágenes ayuda a mejorar la seguridad y la eficiencia en la navegación

- **Estado de Carga de la Batería**

Justificación: El estado de carga de la batería es crucial para la gestión energética del vehículo. Conocer el nivel de carga permite optimizar el uso de la energía, planificar la autonomía del vehículo y evitar situaciones de descarga profunda que podrían dañar la batería.

- **Corriente de los Motores**

Justificación: La corriente consumida por los motores es un indicador directo de la demanda de potencia y el esfuerzo mecánico que están realizando. Monitorear la corriente permite detectar condiciones de sobrecarga y ajustar el control del motor para prevenir daños.

- **Velocidad del Vehículo**

- **Justificación:** La velocidad es una variable fundamental para el control del movimiento y la estabilidad del vehículo. Monitorear la velocidad permite ajustar dinámicamente la operación del vehículo, asegurando una navegación precisa y eficiente.

- **Distancia a Obstáculos**

- **Justificación:** Detectar obstáculos es fundamental para evitar colisiones y asegurar una operación segura. Monitorear la distancia a obstáculos permite al vehículo tomar decisiones de navegación en tiempo real.

4.4. Sensores para el Monitoreo

Listan de los sensores elegidos

- **Sensor de Velocidad:** Encoder HEDM-55xx
- **Sensor de Voltaje:** FZ0430
- **Sensor de Corriente:** ASC712
- **Acelerómetro:** ADXL345
- **Sensor de Distancia:** VL53L0
- **Cámaras:** Cámara USB (trasera) y cámara Raspberry Pi (delantera)

Especificaciones Técnicas y Criterios de Selección de Sensores.

- **Sensor de Velocidad (Encoder HEDM-55xx)**
 - **Especificaciones:** Alta precisión, compatibilidad con microcontroladores Mega.
 - **Criterios de Selección:** Precisión, fiabilidad, facilidad de integración.
- **Sensor de Voltaje (FZ0430)**
 - **Especificaciones:** Rango adecuado para las baterías del vehículo, precisión en la medición de voltaje.
 - **Criterios de Selección:** Precisión, compatibilidad, costo.
- **Sensor de Temperatura (Integrados en los Motores)**
 - **Especificaciones:** Medición directa de la temperatura del motor.
 - **Criterios de Selección:** Integración directa, precisión, costo reducido.
- **Sensor de Corriente (ASC712)**
 - **Especificaciones:** Alta precisión en la medición de corriente, rápida respuesta.
 - **Criterios de Selección:** Precisión, compatibilidad, facilidad de uso.
- **Acelerómetro (ADXL345)**
 - **Especificaciones:** Medición de aceleración en tres ejes, alta precisión.
 - **Criterios de Selección:** Multifuncionalidad, precisión, compatibilidad.
- **Sensor de Distancia (VL53L0)**
 - **Especificaciones:** Tecnología láser para medición precisa de distancias, rápida respuesta.
 - **Criterios de Selección:** Precisión, rapidez, tamaño compacto.

- **Cámaras (USB y Raspberry Pi)**

- **Especificaciones:**

1. **Cámara USB:** Alta resolución, capacidad de transmisión en tiempo real, fácil integración con sistemas USB.
2. **Cámara Raspberry Pi:** Alta resolución, compatibilidad directa con la Raspberry Pi, adecuada para procesamiento en tiempo real.

- **Criterios de Selección:** Calidad de imagen, compatibilidad con sistemas de procesamiento de imágenes, costo.

La integración de estos sensores en el sistema del vehículo autónomo asegura un monitoreo y control precisos, optimizando su desempeño y garantizando su seguridad y eficiencia en diversas condiciones operativas.

4.5. Estructura hardware y funciones

La arquitectura del sistema de control del vehículo está diseñada para garantizar una comunicación eficiente y fiable entre los sensores, actuadores y los microcontroladores. La comunicación se realiza principalmente a través de la Raspberry Pi 4, que actúa como el cerebro del sistema, gestionando la recopilación de datos y el control de los Motores como se describió en el capítulo 2 y 3. Este Computador tiene sistema operativo Raspbian que es una distribución de Linux basada en Debian.

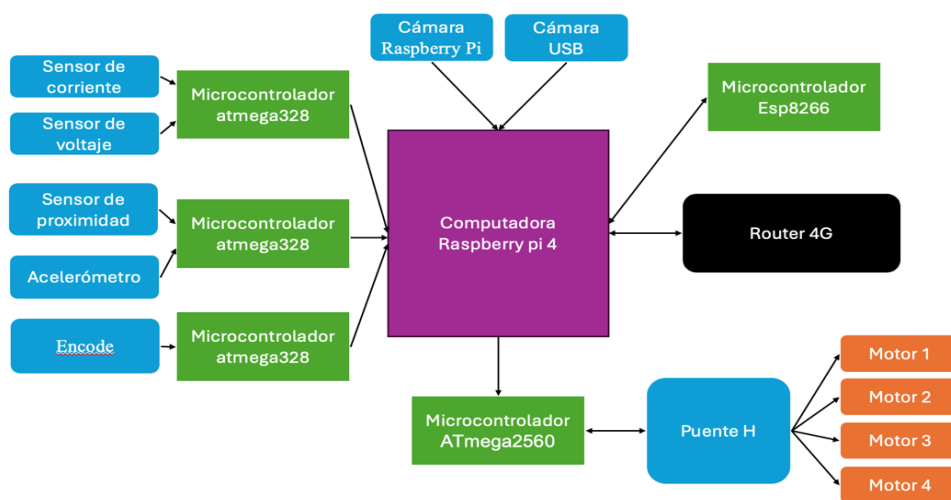


Ilustración 30 Arquitectura del Hardware

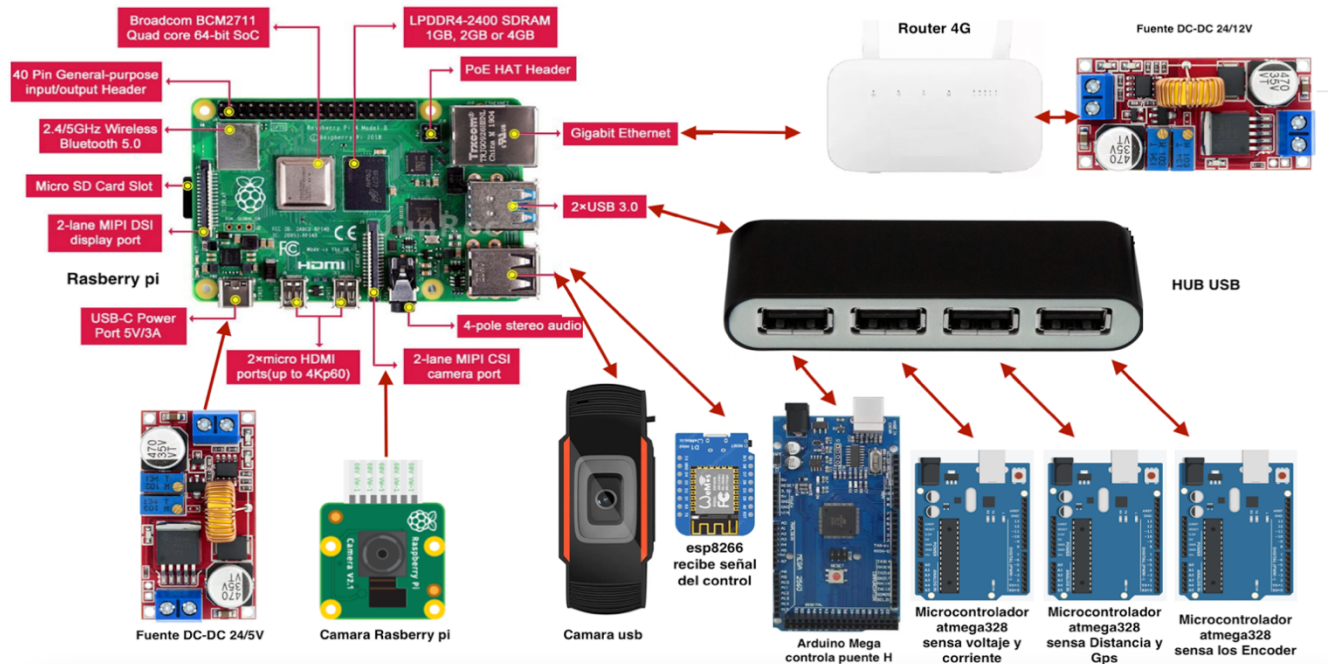


Ilustración 31 Conexión de los equipo que interactúan con la Computadora Raspberry pi 4

- **Red de sensores y microcontroladores:**
 - **I2C y SPI:** Protocolos de comunicación utilizados entre algunos sensores y los microcontroladores, permitiendo la conexión de múltiples sensores con un mínimo de cables.
 - **Señales analógicas:** Algunos sensores reciben datos en formato analógico a los microcontroladores, que luego convierten estos datos a digital antes de enviarlos a la Raspberry Pi.
- **Red de actuadores:**
 - **PWM (Pulse Width Modulation):** Utilizado para controlar la velocidad de los motores y los servomotores. Los microcontroladores generan señales PWM que son enviadas a los controladores de los motores.
- **Red de comunicación interna:**
 - **UART:** Utilizado para la comunicación entre los microcontroladores y la Raspberry Pi. Los microcontroladores se comunican mediante UART con velocidades de transmisión de 4800, 9600, 19200 y 115200 baudios. Esta red permite la transmisión de manera eficiente y con baja latencia.

- **Red de comunicación externa:**

- **LAN (Local Area Network):** La Raspberry Pi se conecta a la red local mediante una conexión LAN, asegurando una comunicación estable y de alta velocidad con otros dispositivos en la red.
- **Wi-Fi:** Utilizado para la comunicación entre la Raspberry Pi y control inalámbrico. Esta el control remoto del vehículo.

Medición Corriente y voltaje

La ilustración 33 muestra un esquema de conexión de un Atmega328 sensores de corriente y un sensor de voltaje. Los sensores de corriente están diseñados para medir la corriente de cuatro motores independientes, y el sensor de voltaje está conectado al banco de baterías para monitorear su nivel. Los sensores están conectados a los pines analógicos del Microcontrolador (A0 a A4) y a los pines de alimentación (5V y GND). La comunicación entre el Arduino y una Raspberry Pi se realiza a través del puerto USB, lo que permite el monitoreo de las lecturas en tiempo real. Mientras que en la ilustración 14 tenemos la rutina que se ejecuta en el microcontrolador . Los pines analógicos del Arduino (A0 a A3) están configurados para medir la corriente de los motores utilizando sensores ACS712, mientras que el pin A4 mide el voltaje del banco de baterías. Los valores analógicos obtenidos de los sensores se convierten a unidades físicas, donde la corriente se calcula en función de la sensibilidad del sensor ACS712 (66 mV/A) y el voltaje se ajusta según el divisor resistivo del circuito. En la ilustración 34 tenemos la rutina del código que se ejecuta en el microcontrolador.

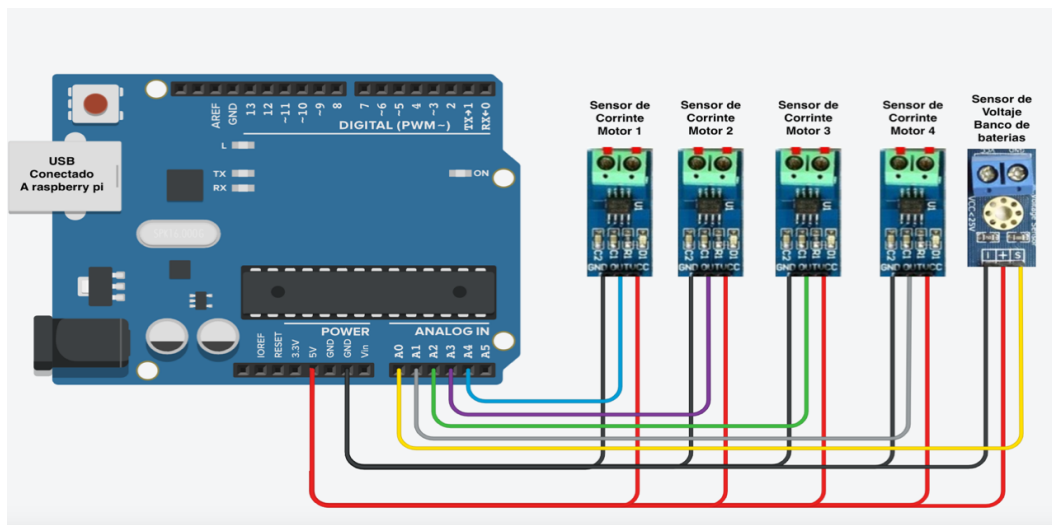


Ilustración 32 Conexión Micro Controlador Atmega328 con sensores de corriente ASC712 y voltaje FZ430

```

1 // Pines analógicos para sensores de corriente y voltaje
2 const int sensorCorriente1 = A0;
3 const int sensorCorriente2 = A1;
4 const int sensorCorriente3 = A2;
5 const int sensorCorriente4 = A3;
6 const int sensorVoltaje = A4;
7
8 // Variables para almacenar las lecturas
9 float corrienteMotor1, corrienteMotor2, corrienteMotor3, corrienteMotor4, voltajeBanco;
10
11 // Función para calcular corriente a partir del sensor ACS712
12 float calcularCorriente(int sensorPin) {
13     int valorAnalogico = analogRead(sensorPin); // Lectura del ADC
14     float voltaje = (valorAnalogico * 5.0) / 1023.0; // Convertir a voltaje (5V referencia)
15     float corriente = (voltaje - 2.5) / 0.066; // Escala del ACS712 (66mV/A, centrado en 2.5V)
16     return corriente; // Retorna la corriente en Amperios
17 }
18
19 // Función para calcular voltaje del banco de baterías
20 float calcularVoltaje(int sensorPin) {
21     int valorAnalogico = analogRead(sensorPin); // Lectura del ADC
22     float voltaje = (valorAnalogico * 5.0) / 1023.0; // Convertir a voltaje
23     return voltaje * 5.0; // Ajustar el divisor de tensión (1:5 en este ejemplo)
24 }
25
26 void setup() {
27     Serial.begin(9600); // Inicializar la comunicación serial
28     // Configurar los pines como entrada (opcional)
29     pinMode(sensorCorriente1, INPUT);
30     pinMode(sensorCorriente2, INPUT);
31     pinMode(sensorCorriente3, INPUT);
32     pinMode(sensorCorriente4, INPUT);
33     pinMode(sensorVoltaje, INPUT);
34
35     Serial.println("Sistema iniciado. Enviando datos...");
36 }
37
38 void loop() {
39     // Leer los valores de los sensores
40     corrienteMotor1 = calcularCorriente(sensorCorriente1);
41     corrienteMotor2 = calcularCorriente(sensorCorriente2);
42     corrienteMotor3 = calcularCorriente(sensorCorriente3);
43     corrienteMotor4 = calcularCorriente(sensorCorriente4);
44     voltajeBanco = calcularVoltaje(sensorVoltaje);
45
46     // Enviar los datos por Serial a la Raspberry Pi en formato CSV
47     Serial.print(corrienteMotor1); Serial.print(",");
48     Serial.print(corrienteMotor2); Serial.print(",");
49     Serial.print(corrienteMotor3); Serial.print(",");
50     Serial.print(corrienteMotor4); Serial.print(",");
51     Serial.println(voltajeBanco);
52
53     delay(100); // Esperar 0.1 segundo antes de la siguiente lectura
54 }

```

Ilustración 33 Código Controlador Atmega328 con sensores de corriente ASC712 y voltaje FZ0430

Medición de Distancia y Aceleración

La ilustración 35 muestra un esquema de conexión de un Atmega328 a sensores de distancia **VL53L0X** y un acelerómetro **ADXL345**. Los sensores de distancia están diseñados para medir la distancia de hasta seis puntos independientes, mientras que el acelerómetro mide las aceleraciones en los tres ejes (X, Y, Z). Los sensores **VL53L0X** están conectados al bus I2C (SDA y SCL) del Arduino y controlados mediante los pines digitales **XSHUT** (pines 7 a 12) para asignarles direcciones únicas en el bus I2C. El acelerómetro **ADXL345** también utiliza el bus I2C para la comunicación. La alimentación de todos los sensores se realiza a través de los pines de 5V y GND del Arduino. La comunicación con una Raspberry Pi se lleva a cabo mediante un puerto USB, lo que permite el monitoreo en tiempo real de los datos.

Los pines digitales **XSHUT** del Arduino (7 a 12) están configurados para activar individualmente cada sensor **VL53L0X** y asignarles direcciones únicas en el bus I2C. Los valores obtenidos de los sensores se convierten en medidas de distancia en milímetros, verificando la validez de cada medición. Paralelamente, el acelerómetro **ADXL345** mide la aceleración en los ejes X, Y y Z, con un rango configurado de $\pm 16g$. Todos los datos procesados, tanto de los sensores de distancia como del acelerómetro, se envían a la Raspberry Pi a través de comunicación serial, permitiendo un monitoreo continuo de las lecturas con intervalos de 100 milisegundos.

En la ilustración 36 tenemos la rutina del código que se ejecuta en el microcontrolador.

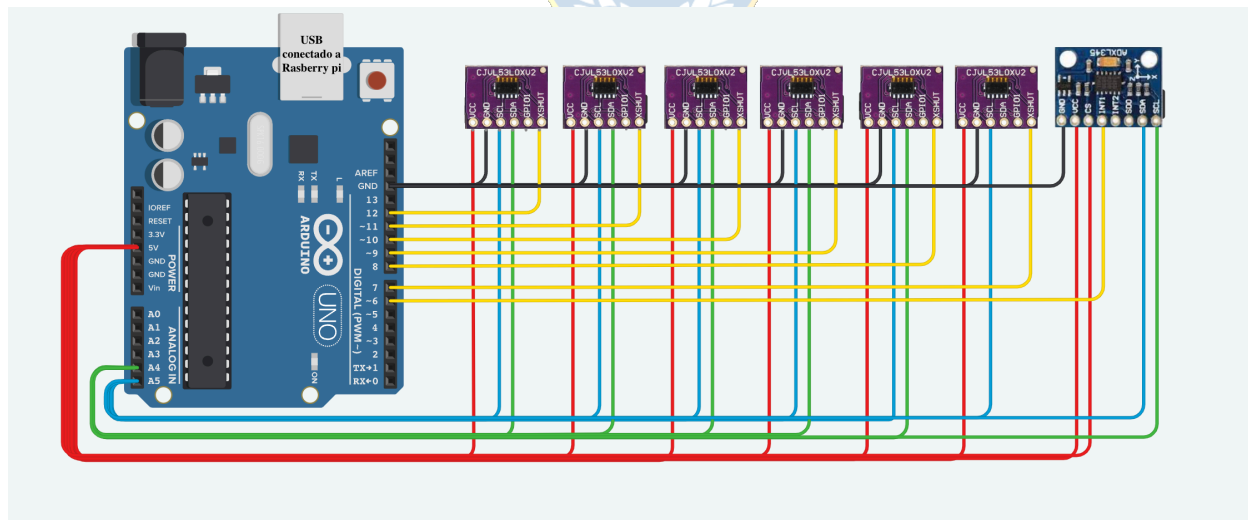


Ilustración 34 Conexión Micro Controlador Atmega328 con sensores de proximidad y Acelerómetro

```

1  #include <Wire.h> // Librería para comunicación I2C
2  #include <Adafruit_VL53L0X.h> // Librería para sensores VL53L0X
3  #include <Adafruit_Sensor.h> // Librería base para sensores
4  #include <Adafruit_ADXL345_U.h> // Librería para el acelerómetro ADXL345
5  // Definición de pines XSHUT para cada sensor VL53L0X
6  #define XSHUT1 7
7  #define XSHUT2 8
8  #define XSHUT3 9
9  #define XSHUT4 10
10 #define XSHUT5 11
11 #define XSHUT6 12
12 // Creación de objetos para cada sensor VL53L0X
13 Adafruit_VL53L0X sensor1 = Adafruit_VL53L0X();Adafruit_VL53L0X sensor2 = Adafruit_VL53L0X();
14 Adafruit_VL53L0X sensor3 = Adafruit_VL53L0X();Adafruit_VL53L0X sensor4 = Adafruit_VL53L0X();
15 Adafruit_VL53L0X sensor5 = Adafruit_VL53L0X();Adafruit_VL53L0X sensor6 = Adafruit_VL53L0X();
16 // Creación del objeto para el acelerómetro ADXL345
17 Adafruit_ADXL345_Unified adxl = Adafruit_ADXL345_Unified(12345);
18 void setup() {Serial.begin(115200); // Inicializar comunicación serial
19   Wire.begin(); // Inicializar el bus I2C
20   pinMode(XSHUT1, OUTPUT); pinMode(XSHUT2, OUTPUT); pinMode(XSHUT3, OUTPUT);
21   pinMode(XSHUT4, OUTPUT); pinMode(XSHUT5, OUTPUT); pinMode(XSHUT6, OUTPUT);
22   digitalWrite(XSHUT1, LOW); digitalWrite(XSHUT2, LOW); digitalWrite(XSHUT3, LOW);
23   digitalWrite(XSHUT4, LOW); digitalWrite(XSHUT5, LOW); digitalWrite(XSHUT6, LOW);delay(10);
24   inicializarSensor(sensor1, XSHUT1, 0x30);inicializarSensor(sensor2, XSHUT2, 0x31);
25   inicializarSensor(sensor3, XSHUT3, 0x32);inicializarSensor(sensor4, XSHUT4, 0x33);
26   inicializarSensor(sensor5, XSHUT5, 0x34);inicializarSensor(sensor6, XSHUT6, 0x35);
27   if (!adxl.begin()) { Serial.println("No se pudo encontrar un acelerómetro ADXL345."); while (1);}
28   adxl.setRange(ADXL345_RANGE_16_G);Serial.println("Sistema inicializado.");}
29 void loop() { // Leer distancias de los sensores VL53L0X
30   leerDistancia(sensor1, "Sensor 1");
31   leerDistancia(sensor2, "Sensor 2");leerDistancia(sensor3, "Sensor 3");
32   leerDistancia(sensor4, "Sensor 4");leerDistancia(sensor5, "Sensor 5");
33   leerDistancia(sensor6, "Sensor 6");
34   // Leer datos del acelerómetro ADXL345
35   leerAcelerometro();delay(100);}
36 void inicializarSensor(Adafruit_VL53L0X &sensor, int xshutPin, uint8_t nuevaDireccion) {
37   digitalWrite(xshutPin, HIGH); delay(10);
38   if (!sensor.begin(nuevaDireccion)) {Serial.print("Error al inicializar VL53L0X en dirección 0x");
39     Serial.println(nuevaDireccion, HEX); while (1);}
40 void leerDistancia(Adafruit_VL53L0X &sensor, const char* nombre) {
41   VL53L0X_RangingMeasurementData_t medida;
42   sensor.rangingTest(&medida, false);
43   if (medida.RangeStatus != 4) {
44     Serial.print(nombre); Serial.print(": ");
45     Serial.print(medida.RangeMilliMeter); Serial.println(" mm");
46   } else {Serial.print(nombre); Serial.println(": Error de medición");}
47 void leerAcelerometro() {sensors_event_t event; adxl.getEvent(&event);
48   Serial.print(""); Serial.print(event.acceleration.x); Serial.println("X");
49   Serial.print(""); Serial.print(event.acceleration.y); Serial.println("Y");
50   Serial.print(""); Serial.print(event.acceleration.z); Serial.println("Z");}

```

Ilustración 35 Conexión Micro Controlador Atmega328 con sensores de proximidad y Acelerómetro

Medición de Posición y Dirección

La ilustración 37 muestra un esquema de conexión de un Atmega328 a cuatro encoders incrementales HEDS-55xx. Estos encoders están diseñados para medir la posición angular y la dirección de rotación de un eje o motor. Cada encoder genera señales en cuadratura (A y B) que permiten determinar la dirección y cantidad de pulsos, además de una señal de índice (Z) que marca una posición de referencia única por cada revolución del eje. Los canales A y B de cada encoder están conectados a los pines digitales del Microcontrolador para la detección de pulsos mediante interrupciones, mientras que los canales Z están configurados como entradas con resistencia pull-up para detectar el índice. La alimentación de los encoders se realiza a través de los pines de 5V y GND del Microcontrolador. La comunicación con una Raspberry Pi se realiza mediante un puerto USB, lo que permite el monitoreo en tiempo real de los datos.

Los pines digitales del Microcontrolador están configurados para capturar los pulsos generados por los canales A y B de los encoders mediante interrupciones, permitiendo contar los pulsos y calcular la dirección de rotación en función de la relación entre ambas señales. El canal Z se utiliza para detectar una posición de referencia única y, opcionalmente, reiniciar el contador de posición del encoder. Los valores obtenidos de los encoders representan la posición angular acumulada de cada eje, aumentando o disminuyendo según la dirección de rotación. Todos los datos procesados se envían a la Raspberry Pi a través de comunicación serial, permitiendo un monitoreo continuo de las lecturas con intervalos de 50 milisegundos.

En la ilustración 38 tenemos la rutina del código que se ejecuta en el microcontrolador.

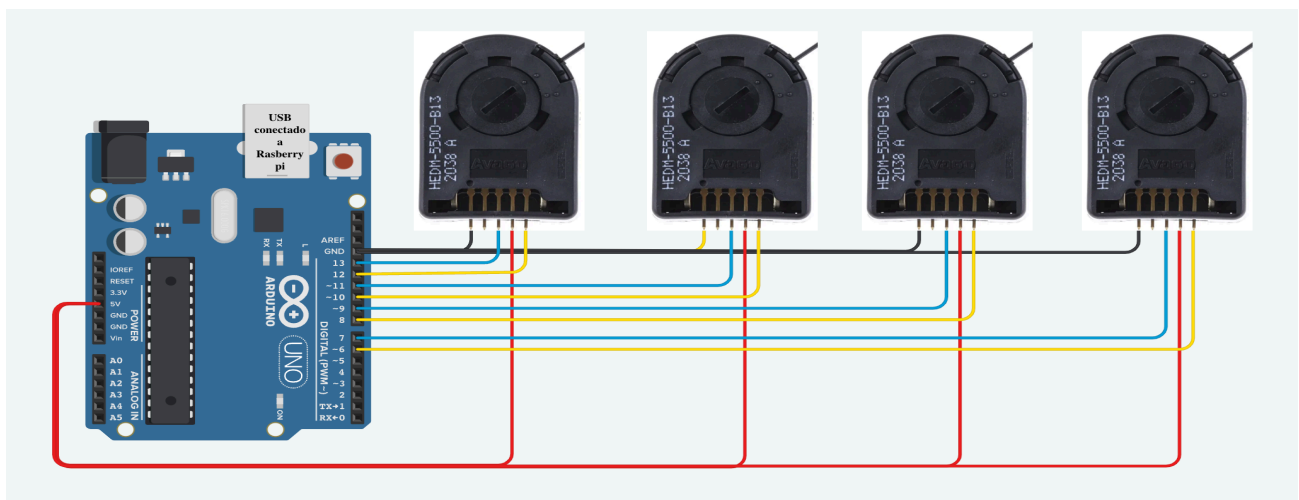


Ilustración 36 Conexión Micro Controlador Atmega328 con Encoder

```

1  #define ENCODER1_A 2 // Pines asignados para los encoders
2  #define ENCODER1_B 3
3  #define ENCODER1_Z 4
4  #define ENCODER2_A 5
5  #define ENCODER2_B 6
6  #define ENCODER2_Z 7
7  #define ENCODER3_A 8
8  #define ENCODER3_B 9
9  #define ENCODER3_Z 10
10 #define ENCODER4_A 11
11 #define ENCODER4_B 12
12 #define ENCODER4_Z 13 // Variables para almacenar las posiciones de los encoders
13 volatile long posicionEncoder1 = 0, posicionEncoder2 = 0, posicionEncoder3 = 0, posicionEncoder4 = 0;
14 // Variables para detectar índice Z (opcional)
15 volatile bool indexDetectado1 = false, indexDetectado2 = false, indexDetectado3 = false, indexDetectado4 = false;
16 void manejarEncoder1A() { // Funciones para manejar interrupciones del canal A
17     if (digitalRead(ENCODER1_A) == digitalRead(ENCODER1_B)) posicionEncoder1++; else posicionEncoder1--;
18 }
19 void manejarEncoder2A() {
20     if (digitalRead(ENCODER2_A) == digitalRead(ENCODER2_B)) posicionEncoder2++; else posicionEncoder2--;
21 }
22 void manejarEncoder3A() {
23     if (digitalRead(ENCODER3_A) == digitalRead(ENCODER3_B)) posicionEncoder3++; else posicionEncoder3--;
24 }
25 void manejarEncoder4A() {
26     if (digitalRead(ENCODER4_A) == digitalRead(ENCODER4_B)) posicionEncoder4++; else posicionEncoder4--;
27 }
28 // Funciones para manejar interrupciones del canal Z
29 void manejarEncoder1Z() { indexDetectado1 = true; posicionEncoder1 = 0; }
30 void manejarEncoder2Z() { indexDetectado2 = true; posicionEncoder2 = 0; }
31 void manejarEncoder3Z() { indexDetectado3 = true; posicionEncoder3 = 0; }
32 void manejarEncoder4Z() { indexDetectado4 = true; posicionEncoder4 = 0; }
33 void setup() { Serial.begin(57600); // Inicializar comunicación serial a 57600 bps
34     // Configurar pines como entradas para cada encoder
35     pinMode(ENCODER1_A, INPUT); pinMode(ENCODER1_B, INPUT); pinMode(ENCODER1_Z, INPUT_PULLUP);
36     pinMode(ENCODER2_A, INPUT); pinMode(ENCODER2_B, INPUT); pinMode(ENCODER2_Z, INPUT_PULLUP);
37     pinMode(ENCODER3_A, INPUT); pinMode(ENCODER3_B, INPUT); pinMode(ENCODER3_Z, INPUT_PULLUP);
38     pinMode(ENCODER4_A, INPUT); pinMode(ENCODER4_B, INPUT); pinMode(ENCODER4_Z, INPUT_PULLUP);
39     // Configurar interrupciones para los encoders
40     attachInterrupt(digitalPinToInterrupt(ENCODER1_A), manejarEncoder1A, CHANGE);
41     attachInterrupt(digitalPinToInterrupt(ENCODER2_A), manejarEncoder2A, CHANGE);
42     attachInterrupt(digitalPinToInterrupt(ENCODER3_A), manejarEncoder3A, CHANGE);
43     attachInterrupt(digitalPinToInterrupt(ENCODER4_A), manejarEncoder4A, CHANGE);
44     attachInterrupt(digitalPinToInterrupt(ENCODER1_Z), manejarEncoder1Z, FALLING);
45     attachInterrupt(digitalPinToInterrupt(ENCODER2_Z), manejarEncoder2Z, FALLING);
46     attachInterrupt(digitalPinToInterrupt(ENCODER3_Z), manejarEncoder3Z, FALLING);
47     attachInterrupt(digitalPinToInterrupt(ENCODER4_Z), manejarEncoder4Z, FALLING);
48     Serial.println("Encoders HEDS-55xx inicializados.");
49 }
50 void loop() { // Mostrar posiciones de los encoders
51     Serial.print("Encoder 1: "); Serial.print(posicionEncoder1);
52     if (indexDetectado1) { Serial.print(" (Índice detectado)"); indexDetectado1 = false; } Serial.println();
53     Serial.print("Encoder 2: "); Serial.print(posicionEncoder2);
54     if (indexDetectado2) { Serial.print(" (Índice detectado)"); indexDetectado2 = false; } Serial.println();
55     Serial.print("Encoder 3: "); Serial.print(posicionEncoder3);
56     if (indexDetectado3) { Serial.print(" (Índice detectado)"); indexDetectado3 = false; } Serial.println();
57     Serial.print("Encoder 4: "); Serial.print(posicionEncoder4);
58     if (indexDetectado4) { Serial.print(" (Índice detectado)"); indexDetectado4 = false; } Serial.println();
59     delay(50); // Pausa entre actualizaciones
60 }

```

Ilustración 37 Conexión Micro Controlador Atmega328 con Encoder

En la ilustración 39 tenemos la rutina del código que se ejecuta en el Raspberry pi en Python para la el rescate de los datos

```
1 import serial
2 import threading
3 import time
4 # Configuración de los puertos seriales y velocidades de comunicación
5 configuracion_serial = [
6     {'puerto': '/dev/ttyUSB0', 'baudrate': 9600}, # Configuración para el primer Arduino
7     {'puerto': '/dev/ttyUSB1', 'baudrate': 115200}, # Configuración para el segundo Arduino
8     {'puerto': '/dev/ttyUSB2', 'baudrate': 57600} # Configuración para el tercer Arduino
9 ]# Lista para almacenar las conexiones seriales
10 conexiones = []
11 # Función para procesar datos de cada Arduino
12 def procesar_datos(nombre, conexion):
13     """
14     Leer datos del puerto serial y procesarlos según el formato esperado.
15     """
16     try:
17         while True:
18             if conexion.in_waiting > 0:
19                 # Leer línea desde el puerto serial
20                 linea = conexion.readline().decode('utf-8').strip()
21                 datos = linea.split(',')
22                 # Identificar datos según el origen
23                 if len(datos) == 6: # Sensores VL53L0X y acelerómetro
24                     print(f"[{nombre}] Distancias sensores: {datos[:5]}, Acelerómetro: {datos[5]}")
25                 elif len(datos) == 4: # Encoders
26                     print(f"[{nombre}] Posiciones encoders: {datos}")
27                 elif len(datos) == 5: # Sensores de corriente y voltaje
28                     print(f"[{nombre}] Corrientes: {datos[:4]}, Voltaje: {datos[4]}")
29                 else:
30                     print(f"[{nombre}] Datos no reconocidos: {datos}")
31             except Exception as e:
32                 print(f"Error en {nombre}: {e}")
33         finally:
34             conexion.close()
35             print(f"Conexión con {nombre} cerrada.")
36
37 # Iniciar las conexiones seriales
38 try:
39     for config in configuracion_serial:
40         conexion = serial.Serial(config['puerto'], config['baudrate'], timeout=1)
41         conexiones.append(conexion)
42         print(f"Conectado a {config['puerto']} a {config['baudrate']} baudios.")
43 except serial.SerialException as e:
44     print(f"No se pudo abrir uno de los puertos: {e}")
45     exit()
46
47 # Crear un hilo para cada Arduino
48 hilos = []
49 for i, conexion in enumerate(conexiones):
50     hilo = threading.Thread(target=procesar_datos, args=(f"Arduino {i + 1}", conexion))
51     hilos.append(hilo)
52     hilo.start()
```

Ilustración 38 Código Python que recibe los datos en Raspberry pi

En la ilustración 40 tenemos la rutina de comando necesarias para configurar las cámaras en el Raspberry pi mediante la consola del terminal . Estas se iniciaran con el sistema y se puede acceder a ellas mediante el puerto que se configure en este caso <http://localhost:9000> y <http://localhost:9001> . Esta rutina es para configurar 1 cámara por lo que se repitió 2 veces para configurar la segunda cámara hay que cambiar algunos detalles como el nombre de la cámara y el puerto en que se configura

```
1  ### Paso 1: Instalar ffmpeg y otros paquetes necesarios
2  Primero, asegúrate de tener `ffmpeg` instalado en tu Raspberry Pi. Si no lo tienes, puedes instalarlo con:
3  ```bash
4  sudo apt-get update
5  sudo apt-get install ffmpeg
6  ```
7  ### Paso 2: Instalar MJPG-Streamer
8  Para transmitir el video en formato MJPEG, puedes utilizar `mjpg-streamer`. Instálalo siguiendo estos pasos:
9  ```bash
10 sudo apt-get install cmake
11 cd ~
12 git clone https://github.com/jacksonliam/mjpg-streamer.git
13 cd mjpg-streamer/mjpg-streamer-experimental
14 make
15 sudo make install
16 ```
17 ### Paso 3: Ejecutar MJPG-Streamer
18 Ahora, ejecuta `mjpg-streamer` con la cámara USB. Usa el siguiente comando:
19 ```bash
20 mjpg_streamer -i "input_uvc.so -d /dev/video0 -r 640x480 -f 30" -o "output_http.so -p 9000 -w ./www"
21 ```
22 Este comando hace lo siguiente:
23 - `-i "input_uvc.so -d /dev/video0 -r 640x480 -f 30"`:
24 Configura la entrada de video desde la cámara USB (/dev/video0) a una resolución de 640x480 y 30 fps.
25 - `-o "output_http.so -p 9000 -w ./www"`:
26 | Configura la salida del video a través de un servidor HTTP en el puerto 9000, utilizando los archivos web en el directorio `./www`.
27 ### Paso 4: Acceder al Stream
28 Abre un navegador web y accede a `http://localhost:9000`. Deberías ver el video transmitido en tiempo real desde tu cámara USB.
29 ### Paso 5 (Opcional): Configurar Autostart
30 Si deseas que `mjpg-streamer` se ejecute automáticamente al iniciar la Raspberry Pi, puedes añadir el comando anterior a tu archivo `rc.local`:
31 ```bash
32 sudo nano /etc/rc.local
33 ```
34 Añade la línea antes de `exit 0`:
35 ```bash
36 mjpg_streamer -i "input_uvc.so -d /dev/video0 -r 640x480 -f 30" -o "output_http.so -p 9000 -w /home/druminot/mjpg-streamer/mjpg-streamer-experimental/www" &
37 ```
38 Guarda y cierra el archivo. Ahora, `mjpg-streamer` se iniciará automáticamente cada vez que enciendas tu Raspberry Pi.
39 ### Paso 6: Instalar dependencias necesarias para compilar
40 Si encuentras un error relacionado con `jpeglib.h`, instala las dependencias necesarias:
41 ```bash
42 sudo apt-get install libjpeg-dev
43 ```
44 ### Paso 7: Volver a compilar `mjpg-streamer`
45 Después de instalar `libjpeg-dev`, vuelve a compilar `mjpg-streamer` siguiendo estos pasos:
46 ```bash
47 cd ~/mjpg-streamer/mjpg-streamer-experimental
48 make clean
49 make
50 sudo make install
51 ```
52 ### Paso 8: Ejecutar `mjpg-streamer`
53 Una vez que la compilación y la instalación se hayan completado sin errores, ejecuta `mjpg-streamer` con el siguiente comando:
54 ```bash
```

Ilustración 39 Rutina configuración Cámaras en Raspbian

4.6. Herramientas y Software Utilizados para el Procesamiento de Datos

El procesamiento de datos y el filtrado de señales son esenciales para asegurar que la información utilizada por el sistema de control sea precisa y fiable. Dada que son varias rutinas distribuidas en distintos códigos se decidió agregarlas en el anexo técnico y solo explicar métodos empleados:

- **Filtrado de Señales:**

- **Filtro de Kalman:** Utilizado para la estimación de variables de estado en sistemas lineales y no lineales. Ayuda a reducir el ruido en las mediciones y a mejorar la precisión de los datos recogidos.
- **Filtro de Media Móvil:** Ayuda a suavizar los datos eliminando fluctuaciones rápidas y mejorando la estabilidad de las mediciones.

- **Detección y Eliminación de Datos Falsos:**

- **Análisis de Desviación Estándar:** Identifica datos anómalos comparando las mediciones actuales con un conjunto de datos históricos para determinar si están fuera del rango esperado.

- **Técnicas de Ingeniería de Datos:**

- **Normalización y Escalado:** Para asegurar que los datos estén en un rango adecuado para el procesamiento.
- **Imputación de Datos Faltantes:** Utilizando métodos estadísticos para completar datos faltantes y asegurar la integridad del conjunto de datos.



Para el procesamiento de datos, se utilizan diversas herramientas y plataformas que facilitan el análisis, almacenamiento y visualización de los datos. Estas incluyen:

- **Python y Bibliotecas Asociadas:**

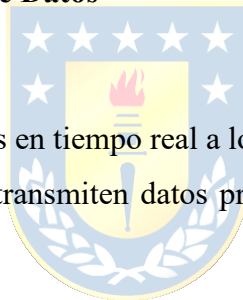
- **Pandas:** Para la manipulación y análisis de datos.
- **NumPy:** Para operaciones matemáticas y de álgebra lineal.
- **SciPy:** Para cálculos científicos y técnicos avanzados.
- **Matplotlib y Seaborn:** Para la visualización de datos.

- **Bases de Datos:**
 - **SQLite:** Utilizado en la Raspberry Pi para el almacenamiento local de datos debido a su ligereza y facilidad de integración con Python.
 - **Firebase:** Para el respaldo de datos en tiempo real y la sincronización de datos entre dispositivos.
 - **Google BigQuery:** Para el análisis de grandes conjuntos de datos y la ejecución de consultas SQL a gran escala.
- **Plataformas de Procesamiento y Almacenamiento en la Nube:**
 - **Firebase:** Proporciona una base de datos en tiempo real y servicios de autenticación, permitiendo la sincronización y el almacenamiento seguro de datos.
 - **Google Cloud Platform (GCP):** Utilizado para BigQuery y otras herramientas de procesamiento y análisis de datos en la nube.

Flujo de Trabajo del Procesamiento de Datos

1. Adquisición de Datos:

- Los sensores envían datos en tiempo real a los microcontroladores.
- Los microcontroladores transmiten datos preprocesados a la Raspberry Pi mediante UART.



2. Procesamiento Inicial:

- La Raspberry Pi realiza un procesamiento inicial de los datos utilizando Python y sus bibliotecas.
- Se aplican filtros y técnicas de eliminación de datos falsos para asegurar la calidad de los datos.

3. Almacenamiento y Sincronización:

- Los datos procesados se almacenan localmente en SQLite y se respaldan en Firebase.
- La sincronización con Google BigQuery permite un análisis avanzado y consultas a gran escala.

4. Análisis y Visualización:

- Los datos almacenados y sincronizados son analizados utilizando herramientas en la nube y visualizados mediante bibliotecas de Python.
- Los resultados del análisis son utilizados para ajustar dinámicamente los parámetros del vehículo y mejorar su rendimiento.



Ilustración 17 Flujo del Procesamiento de los Datos

Este enfoque integral asegura que los datos recopilados sean precisos, fiables y útiles para el control y la optimización del vehículo eléctrico autónomo.

4.7. Sistema de Monitoreo

La arquitectura del sistema de monitoreo está diseñada para proporcionar una visión integral del estado y el rendimiento del vehículo eléctrico autónomo en tiempo real. El sistema de monitoreo se compone de varios componentes clave que interactúan entre sí para recopilar, procesar y visualizar datos de manera eficiente.

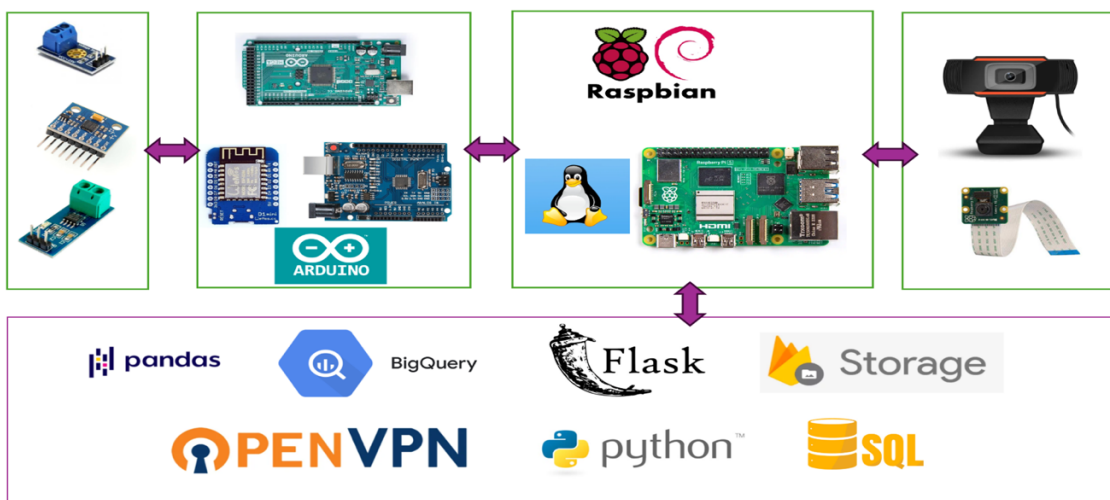


Ilustración 40 Dispositivos y tecnologías empleadas . Referenciado como interactúan entre ellas

Componentes Principales:

- **Sensores y Microcontroladores:** Recogen y transmiten datos sobre diversas variables del vehículo.
- **Raspberry Pi:** Actúa como el núcleo central del sistema, gestionando la recopilación de datos, el procesamiento inicial y la transmisión de datos a la nube. Este tiene sistema operativo Raspbian es una distribución de Linux basada en Debian.
- **Base de Datos Local (SQLite):** Almacena temporalmente los datos antes de su respaldo en la nube.
- **Firestore y Google BigQuery:** Proporcionan almacenamiento y análisis de datos en la nube.
- **Interfaz de Usuario (UI):** Permite la visualización y el control del sistema a través de una aplicación basada en web.

4.8. Sistema de Energía Eléctrica y Distribución

La fuente de alimentación principal del sistema incluye una configuración serie-paralelo de las baterías para proporcionar una salida de 24V. Esta energía se regula mediante fuentes de 12V y 5V para alimentar los diferentes componentes del sistema. La fuente de 12V alimenta el módem 4G, asegurando una conectividad estable y fiable para la transmisión de datos. La fuente de 5V se utiliza para alimentar la Raspberry Pi, microcontroladores y otros componentes electrónicos sensibles, garantizando una operación estable y sin interferencias.

El diseño eléctrico propuesto para el vehículo autónomo se centra en la distribución eficiente y segura de la energía y las señales de control entre los diversos componentes del sistema. El esquema unilineal desarrollado que se aprecia en la Ilustración 42 proporciona una vista clara de cómo se interconectan las fuentes de alimentación, microcontroladores, sensores, actuadores y módulos de comunicación. Este enfoque garantiza que cada componente reciba la energía y las señales necesarias para su funcionamiento óptimo

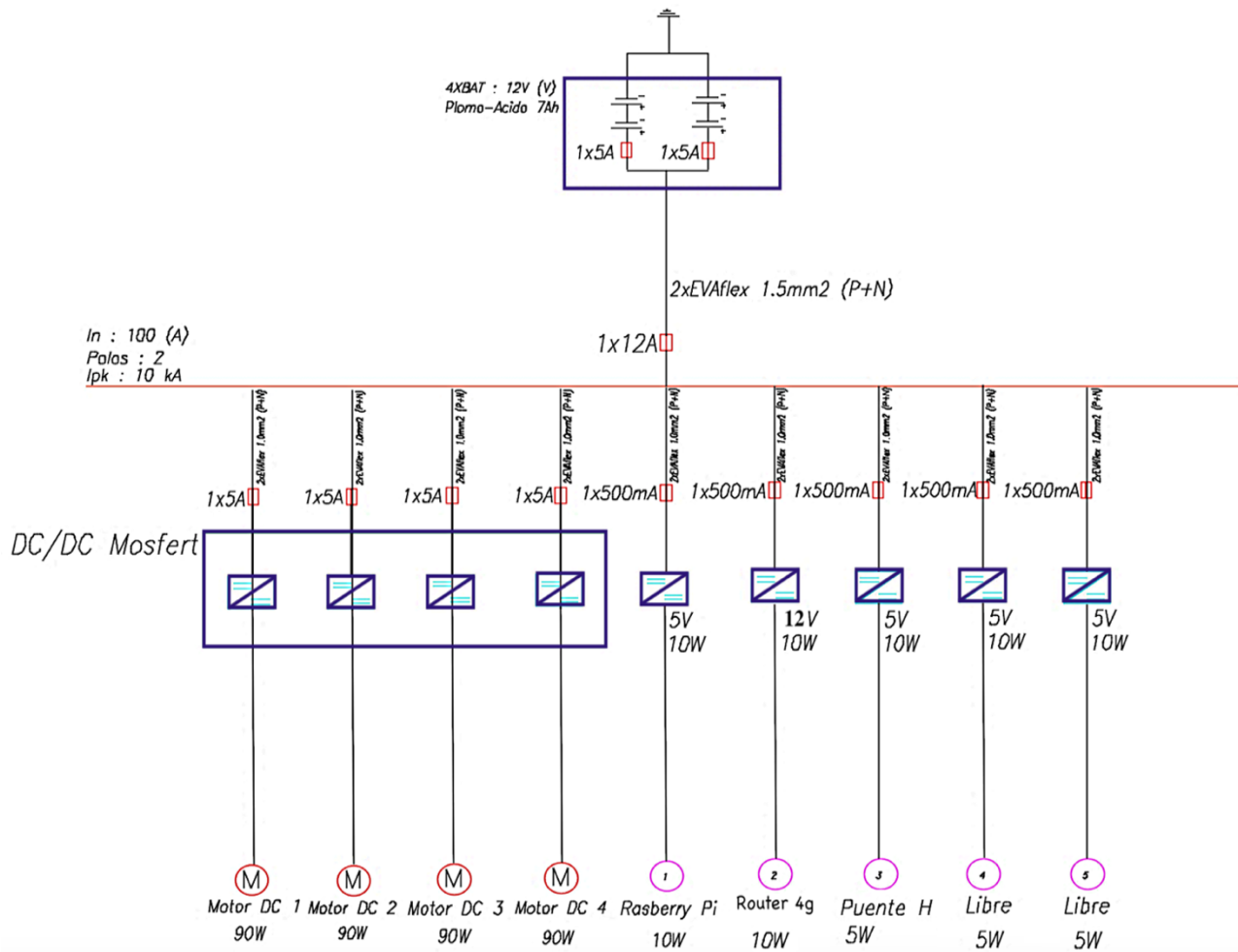


Ilustración 41 Unilineal Distribución Eléctrica del VEA

El esquema unilineal parte de las baterías del vehículo, que son la fuente primaria de energía. Estas baterías están configuradas en paralelo y en serie, con dos baterías en serie para obtener un total de 24V, y luego las parejas de baterías en paralelo para aumentar la capacidad total. Este diseño proporciona una fuente robusta de 24V que se distribuye a varios componentes del sistema mediante reguladores de voltaje y fusibles para protección.

- **Baterías:**
 - **Configuración:** 4 baterías conectadas en un arreglo serie-paralelo para obtener 24V en total , la distribución es la siguiente 2 baterías en serie de 12V 7 Ampere-Hora de Plomo-Acido conectades en paralela junto a otras 2 de las misma características anteriores .
 - **Protección:** Fusibles instalados para proteger los circuitos de sobrecargas y cortocircuitos.
- **Fuentes Regulables:**
 - **Fuente de 12V:** Alimenta el módem 4G que proporciona conectividad de datos para el vehículo.
 - **Fuente de 5V:** Alimenta la Raspberry Pi, microcontroladores y otros componentes electrónicos sensibles.
- **Distribución de Energía:**
 - La energía de 24V se distribuye directamente a los puentes H y los motores a través de fusibles de protección.
 - Las fuentes de 12V y 5V regulan el voltaje para suministrar energía adecuada a los diferentes componentes del sistema.
- **Componentes Principales y Conexiones:**
 - **Raspberry Pi:** Alimentada por la fuente de 5V, gestiona la comunicación y el procesamiento de datos.
 - **Microcontroladores:** También alimentados por la fuente de 5V, procesan las señales de los sensores y controlan los actuadores.
 - **Módem 4G:** Alimentado por la fuente de 12V, proporciona conectividad de datos.
 - **Motores y Puente H (LMD18200):** Alimentados directamente por la línea de 24V, controlan la tracción y dirección del vehículo.

4.9. Conclusiones

El sistema de monitoreo implementado para el vehículo eléctrico autónomo ha permitido la recopilación y análisis detallado de datos en tiempo real, proporcionando información valiosa sobre el rendimiento y la eficiencia del vehículo. Los hallazgos principales incluyen la precisión de los datos recolectados, el rendimiento de los sensores y actuadores, y la eficiencia de la comunicación entre los componentes del sistema. Los filtros aplicados, como el filtro de Kalman y la media móvil, han demostrado ser efectivos en la reducción del ruido y la mejora de la precisión de los datos. Los sensores utilizados han proporcionado datos fiables sobre la posición, velocidad, estado de la batería y el entorno del vehículo. Además, los actuadores controlados por los microcontroladores y la Raspberry Pi han respondido adecuadamente a las señales de control, garantizando un movimiento preciso y eficiente del vehículo. La comunicación UART entre los microcontroladores y la Raspberry Pi ha mostrado una baja latencia y alta fiabilidad, mientras que la sincronización de datos con Firebase y Google BigQuery ha permitido un almacenamiento seguro y un análisis avanzado de los datos.

El diseño del sistema de monitoreo garantiza una recopilación y procesamiento eficiente de datos en tiempo real, permitiendo un control preciso y una optimización continua del vehículo eléctrico autónomo. La integración de tecnologías de almacenamiento en la nube y análisis de datos asegura que el sistema sea robusto, escalable y capaz de proporcionar información valiosa para mejorar el rendimiento del vehículo. Las pruebas y la validación exhaustivas aseguran que el sistema cumple con los requisitos y expectativas establecidos.

5. Control Inalámbrico

5.1. Introducción

La implementación de control inalámbrico en vehículos autónomos eléctricos es fundamental para mejorar su operatividad y eficiencia. Este método busca no solo facilitar la comunicación entre el vehículo y los sistemas de monitoreo, sino también optimizar la gestión energética y la interacción con el entorno. En el contexto del presente proyecto, se utiliza un módem 4G con un chip prepago para proporcionar conectividad a Internet al vehículo. Esta conexión es esencial para permitir el acceso remoto al vehículo y la transmisión en tiempo real de datos de monitoreo y control

Ventajas:

- **Mayor Flexibilidad:** Permite el control del vehículo desde cualquier ubicación con acceso a Internet.
- **Reducción de Costos:** Elimina la necesidad de infraestructura física extensa para la transmisión de datos (como la implementación de antenas).
- **Seguridad:** La implementación de una VPN en este caso de la mano de OpenVPN proporciona una conexión segura, protegiendo contra posibles intrusiones.
- **Eficiencia:** Los dispositivos 4G pueden cambiar dinámicamente entre diferentes frecuencias y configuraciones para adaptarse a la intensidad de la señal o la congestión de la red. Esto garantiza una experiencia más consistente

Desafíos:

- **Latencia:** A pesar de las mejoras, la latencia siempre será un desafío en los sistemas de control inalámbrico.
- **Configuración y Mantenimiento:** Requiere una configuración inicial y un mantenimiento continuo para asegurar el correcto funcionamiento de la VPN y la conectividad 4G.

5.2. Estructura de control

Los sistemas de control inalámbrico utilizan tecnologías de comunicación sin cables para transmitir datos entre el vehículo y un sistema de control remoto. Los beneficios incluyen una mayor flexibilidad y la reducción de costos de infraestructura. Sin embargo, estos sistemas también enfrentan desafíos como la latencia y la seguridad en la transmisión de datos.

Para asegurar una conexión estable y segura, se evaluaron diferentes métodos de acceso remoto. Inicialmente, se consideró el uso de VNC (Virtual Network Computing) para acceder al vehículo de manera remota. Aunque VNC permite controlar el vehículo a través de una interfaz gráfica, no es ideal para un usuario común debido a su complejidad y potenciales problemas de latencia, también se evaluó una conexión directa tipo abriendo el raspberry pi al internet pero se descarto por los riesgo que eso conlleva (ataques de fuerza , hackeos ,etc).

Por ello, se propuso el uso de una VPN en este caso OpenVPN por su prestigio y opciones gratuitas para crear una red privada virtual .Esta solución permite a los usuarios conectarse al vehículo a través de una red privada, asegurando la seguridad y mejorando la latencia. Esta crea un túnel cifrado entre los dispositivos de los usuarios y el vehículo, asegurando la integridad y confidencialidad de los datos transmitidos, además de mitigar riesgos asociados con accesos no autorizados.

5.3. Implementación

1. Conectividad 4G:

- **Módem 4G con Chip Prepago:** Proporciona conectividad a Internet al vehículo, permitiendo la transmisión de datos en tiempo real y el acceso remoto. Este módem es esencial para la operatividad del control inalámbrico.

2. Acceso Remoto:

- **VNC (Virtual Network Computing):** Inicialmente utilizado para el acceso remoto, permite el control del vehículo a través de una interfaz gráfica. Sin embargo, presenta limitaciones en términos de facilidad de uso y latencia.

- **OpenVPN:** Implementado como la solución principal para el acceso remoto. OpenVPN crea una red privada virtual que proporciona una conexión segura y eficiente. Los usuarios pueden conectarse al vehículo a través de esta VPN y acceder a su control mediante un navegador web.

3. Control y Monitoreo a través del Navegador:

- **Configuración como Local Host:** Una vez conectado a la VPN, el vehículo se configura como un host local. Esto permite que los usuarios accedan al sistema de control del vehículo directamente desde su navegador web, proporcionando una interfaz amigable y fácil de usar.
- **Seguridad y Eficiencia:** OpenVPN asegura que la transmisión de datos sea segura, protegiendo la información del vehículo y del usuario. Además, esta configuración mejora la latencia, proporcionando una respuesta rápida y fiable en el control del vehículo.

En la ilustración 43 se muestra cómo funciona una VPN, que actúa como un túnel cifrado entre el automóvil y el usuario, asegurando que toda la información transmitida esté protegida incluso al viajar por internet público. Este túnel cifra los datos antes de enviarlos, haciéndolos ilegibles para terceros, y solo el usuario y el automóvil pueden descifrarlos. La VPN verifica la identidad de ambas partes para garantizar una conexión segura y proteger la privacidad del usuario al ocultar su ubicación e identidad reales. Al mantener los datos protegidos, la VPN evita ataques como la interceptación o el acceso no autorizado, creando un canal seguro y confiable para el control remoto del automóvil.

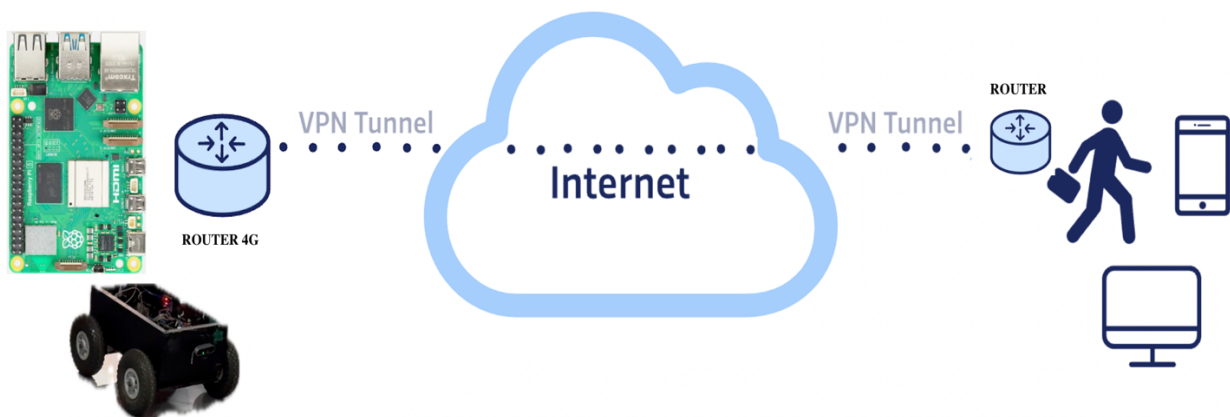


Ilustración 42 Estructura de la implementación

5.4. Seguridad y Validación del Sistema

La seguridad y validación del sistema de control inalámbrico son componentes esenciales para garantizar la integridad y la funcionalidad del vehículo autónomo. A continuación, se describen los métodos y estrategias implementadas para asegurar y validar el sistema.

Seguridad del Sistema:

- **Encriptación de Datos:** OpenVPN utiliza encriptación avanzada para asegurar que todas las comunicaciones entre el vehículo y el sistema de control remoto sean seguras. Esto previene accesos no autorizados y protege la integridad de los datos transmitidos.
- **Autenticación de Usuarios:** La configuración de OpenVPN incluye mecanismos de autenticación robustos, asegurando que solo usuarios autorizados puedan acceder al sistema de control del vehículo.
- **Monitoreo de Seguridad:** Se implementan herramientas para monitorear constantemente la red y detectar cualquier actividad sospechosa o intentos de acceso no autorizados. Esto permite una respuesta rápida a posibles amenazas de seguridad.

5.5. Implementación de un Dispositivo Auxiliar de Control de Corta Distancia

Para situaciones que requieren maniobras precisas o para el control auxiliar del vehículo a corta distancia, se ha diseñado y construido un dispositivo de control remoto basado en el ESP8266. Este dispositivo, que se comunica con el vehículo a una distancia de 10 a 20 metros, ofrece una solución práctica y eficaz para el control del vehículo en espacios confinados o durante maniobras complicadas.

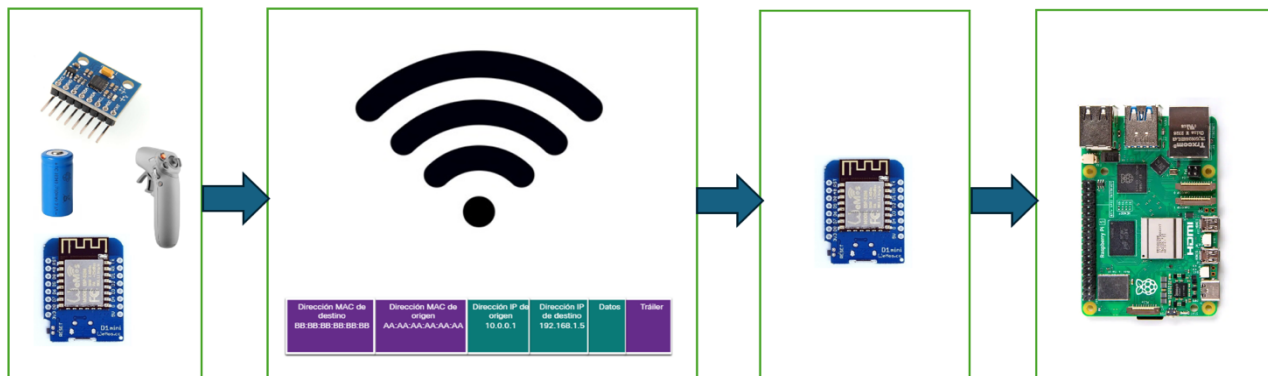


Ilustración 43 Flujo de Trabajo del Dispositivo Auxiliar

Descripción del Dispositivo Auxiliar

El dispositivo auxiliar está compuesto por los siguientes elementos:

- **ESP8266:** Este microcontrolador se encarga de la comunicación inalámbrica entre el mando y el vehículo.
- **Batería:** Proporciona la energía necesaria para el funcionamiento del dispositivo.
- **Acelerómetro:** Utilizado para detectar la orientación del mando.
- **Botón de Aceleración:** Permite al usuario controlar la aceleración del vehículo.

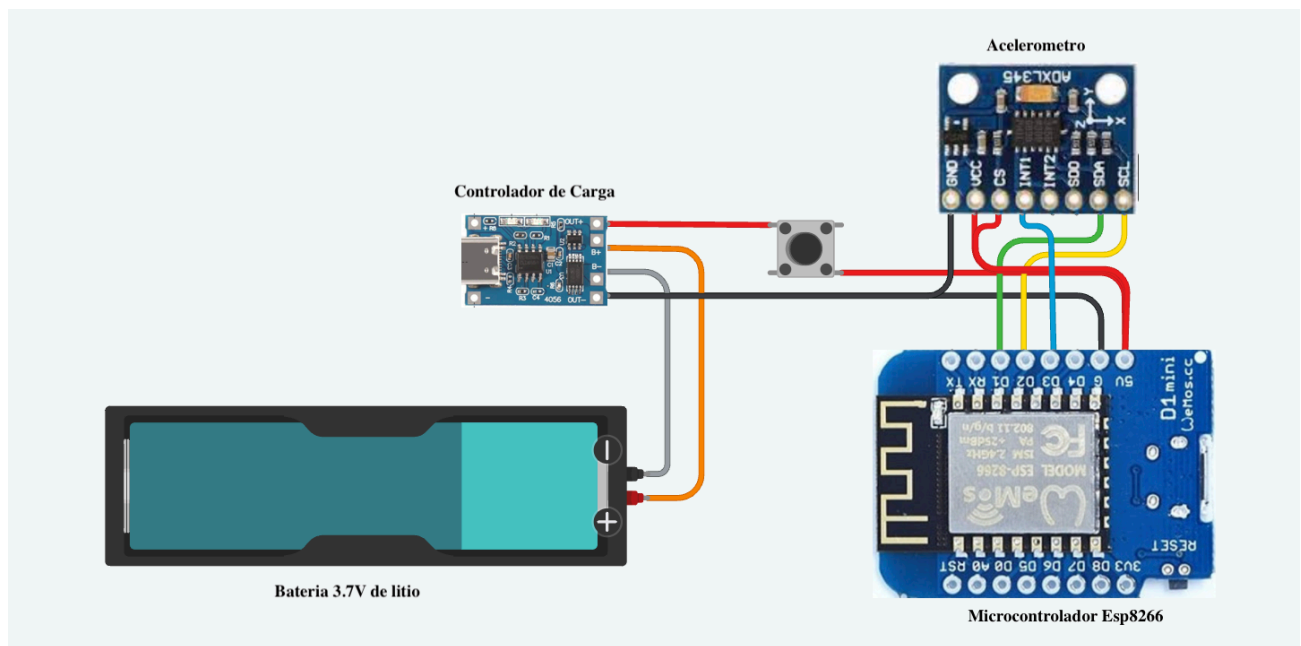


Ilustración 44 Circuito Control Inalámbrico

Diseño y Construcción

El mando fue diseñado con una carcasa en 3D, la cual fue impresa utilizando una impresora 3D para proporcionar una ergonomía adecuada y facilitar su uso. La carcasa contiene todos los componentes electrónicos y tiene un diseño compacto y robusto. El acelerómetro integrado en el mando detecta cuando el dispositivo se apunta hacia adelante, y al presionar el botón, el vehículo acelera en esa dirección. En la Ilustración 46 podemos apreciar el modelo 3D creado

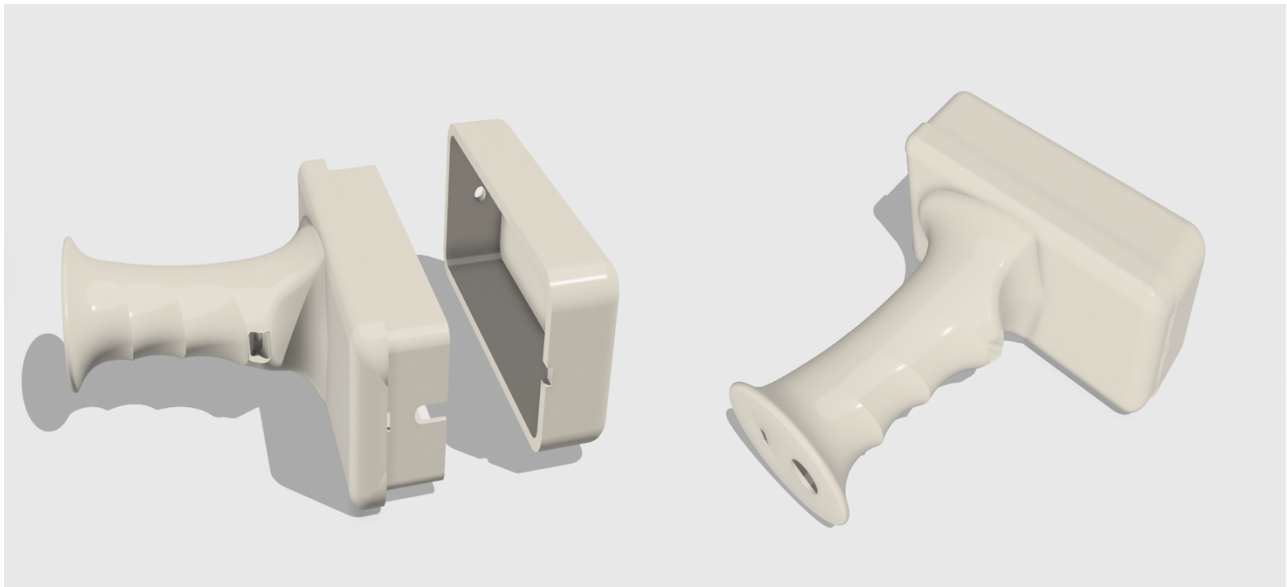


Ilustración 45 Modelo 3D del Control Inalámbrico Diseñado Para Impresión 3D

Comunicación entre Dispositivos

La comunicación entre el mando y el vehículo se realiza mediante dos módulos ESP8266 que operan en la capa 2 del modelo OSI, donde se encuentran las subcapas MAC (Media Access Control) y LLC (Logical Link Control). La implementación se basa en la modificación de los paquetes a nivel de enlace de datos para incluir la información necesaria para el control del vehículo. La comunicación incluye una encriptación sencilla propia, que asegura que los datos transmitidos estén protegidos contra posibles interferencias o accesos no autorizados.

- **Capa 2 del Modelo OSI:**

- Los módulos ESP8266 intercambian paquetes a nivel de la subcapa MAC, que gestiona el acceso al medio y controla la comunicación de los dispositivos en la red.
- Los paquetes incluyen datos básicos como la orientación del mando y el estado del botón de aceleración, gestionados por la subcapa LLC que controla la lógica del enlace de datos.

- **Encriptación Propia:**

- Para asegurar la integridad y privacidad de los datos transmitidos, se utiliza una encriptación sencilla propia. Esta encriptación protege los datos contra accesos no autorizados y asegura que solo el vehículo interprete las señales enviadas desde el mando.

Trabajos futuros

El dispositivo auxiliar es especialmente útil para maniobras precisas y como una herramienta auxiliar en situaciones complejas. Sin embargo, hay oportunidades para mejorar este sistema en el futuro:

- **Tecnología LoRa:** Puede ofrecer un rango de comunicación más amplio y una mayor fiabilidad.
- **Integración con 4G/5G:** Para una conectividad más robusta y de mayor alcance, permitiendo un control más flexible y remoto.

El dispositivo auxiliar de control de corta distancia proporciona una solución eficaz para el manejo preciso del vehículo en situaciones específicas. Su diseño compacto y la comunicación eficiente mediante ESP8266 aseguran su funcionalidad y fiabilidad, mientras que las posibles mejoras tecnológicas ofrecen un camino claro para futuras optimizaciones. La implementación basada en la capa 2 del modelo OSI, donde se manejan las subcapas MAC y LLC, junto con la encriptación propia, garantiza una comunicación segura y efectiva.

5.6. Método de Control Inalámbrico y Protocolos de Comunicación

La implementación del control inalámbrico en el vehículo autónomo eléctrico ha sido fundamental para mejorar la operatividad y eficiencia del sistema. El uso de tecnologías de comunicación inalámbrica ha proporcionado mayor flexibilidad y ha reducido los costos de infraestructura. Sin embargo, se han enfrentado desafíos relacionados con la latencia y la seguridad en la transmisión de datos.

Tecnología de Control Inalámbrico

El control inalámbrico se implementó utilizando un módem 4G con un chip prepago para proporcionar conectividad a Internet al vehículo. Inicialmente se consideró VNC para el acceso remoto, pero se descartó por su complejidad y posibles problemas de latencia para usuarios comunes. Por ello, se optó por implementar OpenVPN, que ofrece una VPN segura y eficiente. Esta configuración permite a los usuarios conectarse al vehículo a través de una red privada y controlarlo directamente desde su navegador web, mejorando la seguridad y reduciendo la latencia.

Dispositivo Auxiliar de Control de Corta Distancia

Además del control vía OpenVPN, se diseñó y construyó un dispositivo auxiliar de control de corta distancia utilizando módulos ESP8266 para la comunicación. Este dispositivo, que incorpora un acelerómetro y un botón de aceleración, se comunica con el vehículo a través de paquetes de capa 2 del modelo OSI, asegurando una comunicación eficiente y fiable en un rango de 10 a 20 metros.

Conclusión

La implementación del control inalámbrico y los protocolos de comunicación ha resultado en una operatividad más eficiente y segura del vehículo autónomo eléctrico. La integración de OpenVPN para el control remoto y la solución auxiliar con ESP8266 para distancias cortas proporciona una base sólida para el control y monitoreo del vehículo, optimizando su rendimiento en diversas condiciones operativas. Esta implementación mejora la experiencia del usuario y establece un camino para futuras optimizaciones tecnológicas, asegurando la flexibilidad, seguridad y eficiencia del sistema de control del vehículo.

6. Interfaz de Usuario y Visualización de Datos

6.1. Introducción

La interfaz de usuario (UI) es crucial en los sistemas de vehículos autónomos eléctricos, ya que facilita la interacción entre el operador y el vehículo, mejorando la eficiencia y la seguridad operativa. El objetivo del diseño de la UI es proporcionar una herramienta intuitiva y accesible que permita a los usuarios monitorear y controlar el vehículo de manera efectiva.

En el desarrollo del sistema de control y monitoreo para el vehículo eléctrico autónomo, se ha implementado una interfaz de usuario basada en tecnologías web que permite el acceso y la gestión de diferentes componentes del sistema. Esta interfaz está diseñada para ser accesible desde cualquier dispositivo con conexión a Internet, utilizando navegadores web estándar. La interfaz principal incluye la visualización en tiempo real de datos críticos, control manual del vehículo y acceso a cámaras de monitoreo.



6.2 Especificaciones de la Interfaz

Para desarrollar una UI efectiva, se deben identificar claramente los requisitos del usuario y las especificaciones técnicas y funcionales. Estos incluyen:

Requisitos del usuario:

- **Facilidad de uso:** La interfaz debe ser intuitiva, permitiendo a los usuarios interactuar con el sistema sin necesidad de capacitación especializada.
- **Acceso rápido a información crítica:** Los datos más relevantes, como la velocidad del vehículo, estado de la batería y alertas de sensores, deben estar disponibles de manera inmediata.
- **Personalización de vistas:** Los usuarios deben poder configurar la interfaz para mostrar solo la información que consideran relevante.

Especificaciones técnicas:

- **Compatibilidad con diferentes dispositivos:** La interfaz debe ser accesible desde computadoras portátiles, tablets y smartphones.
- **Baja latencia en la actualización de datos:** Los datos deben actualizarse en tiempo real para reflejar el estado actual del vehículo.
- **Capacidad de manejo de múltiples conexiones:** El sistema debe soportar múltiples usuarios accediendo simultáneamente sin degradar el rendimiento.

Consideraciones de usabilidad:

- **Diseño intuitivo:** La disposición de los elementos debe seguir principios de diseño centrado en el usuario.
- **Accesibilidad para usuarios con discapacidades:** La interfaz debe cumplir con estándares de accesibilidad para asegurar que todos los usuarios puedan interactuar con el sistema.
- **Respuesta rápida a las interacciones del usuario:** Las acciones realizadas por los usuarios deben reflejarse de manera inmediata en el sistema.

6.3 Diseño y Desarrollo de la Interfaz de Usuario

El diseño de la UI se basa en principios de usabilidad, simplicidad y eficiencia. Se utilizan herramientas modernas de diseño y desarrollo, como HTML5, CSS3 y JavaScript, para crear prototipos y maquetas interactivas. La arquitectura del software de la interfaz incluye elementos gráficos y controles interactivos, integrados con el sistema de control y monitoreo del vehículo.

Descripción de la arquitectura y funcionalidades:

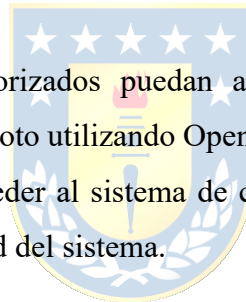
a) **Página Principal (localhost:8000):**

- **Visualización de Cámaras:** A la derecha de la página principal, se integran las vistas de las cámaras ubicadas en `localhost:9001` (Cámara 1) y `localhost:9000` (Cámara 2).
- **Controles de Movimiento:** A la izquierda, se disponen los controles de palanca táctil para el manejo del vehículo, permitiendo ajustes en tiempo real.

- **Monitoreo de Variables:** En la parte inferior de la página, se muestran datos esenciales como velocidad, aceleración, corriente de motores y estado de sensores de distancia y voltaje de la batería.
- b) **Página de Análisis de Datos:**
 - **Gráficos y Reportes:** Se proporcionan gráficos detallados para el análisis intensivo de datos históricos y en tiempo real, obtenidos de Firebase. Esta página está diseñada para permitir una inspección más profunda de las variables críticas y el rendimiento del vehículo.
- c) **Integración con Firebase:**
 - **Arquitectura de Datos:** Los datos se almacenan y recuperan desde Firebase, permitiendo una gestión eficiente y segura de la información. Esto incluye tanto datos en tiempo real como históricos, facilitando el análisis y la toma de decisiones.

Seguridad y Acceso:

Para asegurar que solo usuarios autorizados puedan acceder y controlar el vehículo, se ha implementado un sistema de acceso remoto utilizando OpenVPN. Esto permite que cualquier usuario en la misma red de Internet pueda acceder al sistema de control de manera segura, protegiendo la información y garantizando la integridad del sistema.



Herramientas y Tecnologías Utilizadas:

- **HTML5 y CSS3:** Para la estructura y estilo de la interfaz.
- **JavaScript:** Para la funcionalidad interactiva y la comunicación en tiempo real con el backend.
- **Flask (Python):** Para el desarrollo del backend y la gestión de las rutas y APIs del sistema.
- **Firebase:** Para la gestión de datos en tiempo real y almacenamiento en la nube.
- **WebSockets:** Para la transmisión de datos en tiempo real entre el servidor y la interfaz de usuario.

6.4 Visualización de Datos en Tiempo Real

La visualización de datos en tiempo real es esencial para el monitoreo efectivo del vehículo. Se implementan técnicas avanzadas para representar gráficamente variables críticas como la velocidad, el estado de la batería y la temperatura. La interfaz está diseñada para actualizar y refrescar los datos en tiempo real, asegurando que los usuarios tengan acceso a información precisa y actualizada. Estas visualizaciones incluyen gráficos interactivos y paneles de control que permiten a los usuarios comprender rápidamente el estado actual del vehículo y tomar decisiones informadas.

Funcionamiento de WebSockets:

Los WebSockets son una tecnología clave para lograr la visualización en tiempo real de los datos. A diferencia de las conexiones HTTP tradicionales, que son unidireccionales y requieren una solicitud del cliente para recibir datos del servidor, los WebSockets permiten una comunicación bidireccional y persistente entre el cliente y el servidor. Esto se logra mediante los siguientes pasos:

a) Establecimiento de Conexión:

- El cliente (navegador) envía una solicitud de conexión WebSocket al servidor.
- El servidor acepta la solicitud y establece una conexión persistente.

b) Comunicación Bidireccional:

- Una vez establecida, la conexión WebSocket permite que el servidor envíe datos al cliente sin que este los solicite explícitamente.
- De igual manera, el cliente puede enviar datos al servidor en cualquier momento a través de la misma conexión.

c) Actualización en Tiempo Real:

- Los datos críticos, como la velocidad y el estado de la batería, se transmiten continuamente desde el servidor al cliente.
- En el lado del cliente, se utilizan bibliotecas de JavaScript para actualizar dinámicamente el DOM (Document Object Model) con los datos recibidos, asegurando que la interfaz refleje el estado actual del vehículo sin necesidad de recargar la página.

d) **Manejo de Eventos:**

- WebSockets permiten la gestión de eventos en tiempo real, como cambios en el estado de los sensores o comandos de control del usuario. Esto garantiza una respuesta inmediata y sincronizada entre el vehículo y la interfaz de usuario.

Integración con Flask y Firebase:

- **Flask:** Actúa como el servidor backend, manejando las conexiones WebSocket y sirviendo los datos en tiempo real a la interfaz de usuario. Flask-SocketIO es una extensión que facilita la integración de WebSockets en aplicaciones Flask.
- **Firebase:** Almacena y sincroniza los datos del vehículo en tiempo real, permitiendo que el backend de Flask recupere y transmita estos datos a los clientes conectados mediante WebSockets.

6.5 Pruebas de Usabilidad y Evaluación del Rendimiento

Las pruebas de usabilidad son fundamentales para garantizar que la UI cumpla con los requisitos de los usuarios. Se utiliza una metodología de pruebas que incluye la retroalimentación de los usuarios y la validación en condiciones operativas reales. Los resultados de las pruebas se analizan para identificar áreas de mejora y realizar ajustes necesarios. La evaluación del rendimiento de la interfaz también considera la satisfacción del usuario y la eficiencia del sistema en su conjunto.

Metodología de pruebas:

- **Recopilación de retroalimentación:** Se obtiene retroalimentación de usuarios reales mediante encuestas y entrevistas.
- **Validación en condiciones reales:** Se prueba la interfaz en escenarios operativos reales para asegurar su funcionalidad y eficacia.
- **Análisis de resultados:** Se analizan los datos obtenidos de las pruebas para identificar mejoras y realizar los ajustes necesarios.

Evaluación del rendimiento:

- **Satisfacción del usuario:** Se mide la satisfacción de los usuarios con la interfaz y su facilidad de uso.
- **Eficiencia del sistema:** Se evalúa la eficiencia del sistema en términos de tiempo de respuesta y fiabilidad.

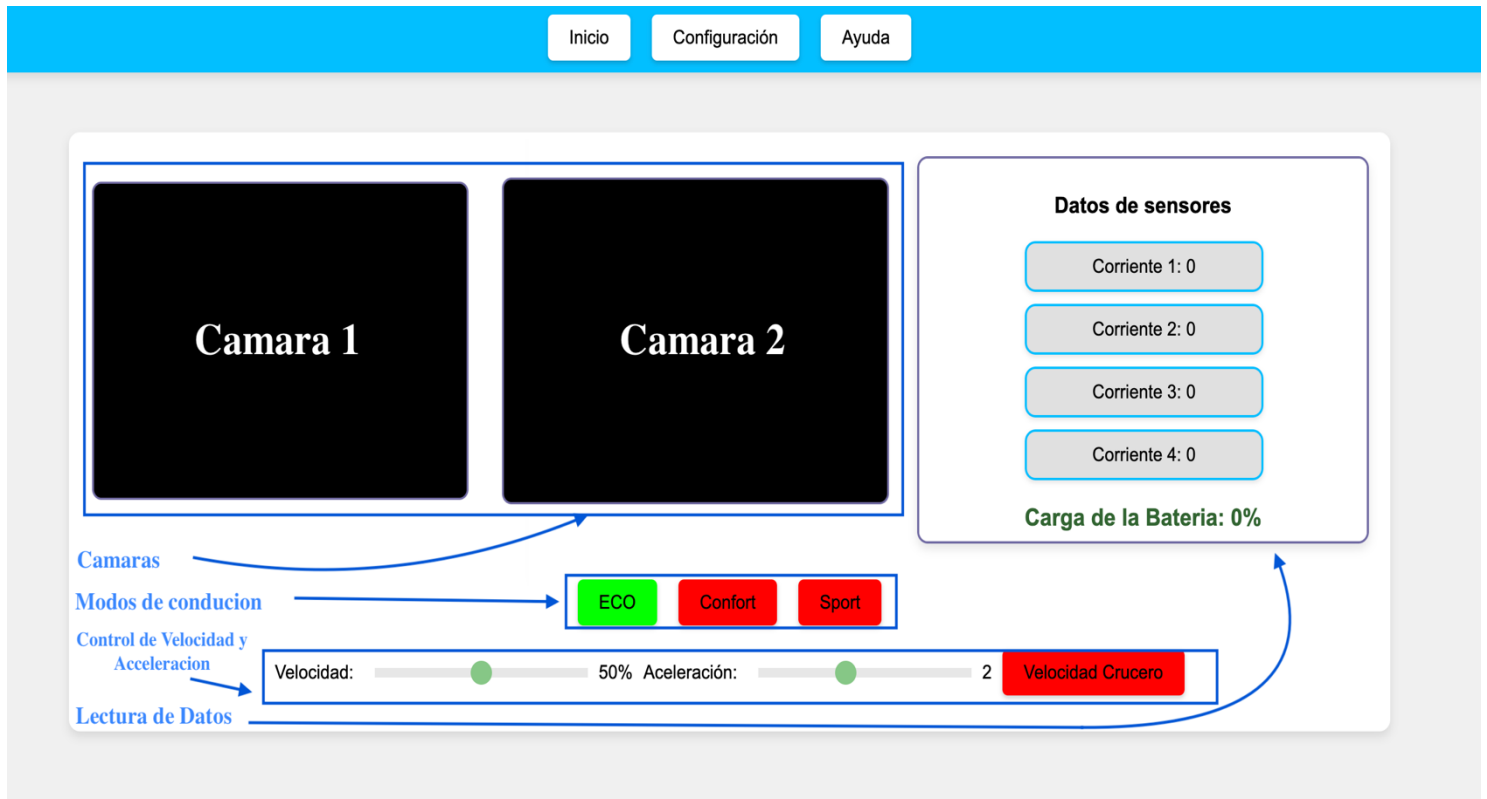


Ilustración 46 Interfaz de usuario

Conclusiones

La implementación de una interfaz de usuario eficiente y accesible es fundamental para la operatividad y el control del vehículo eléctrico autónomo. A través del uso de tecnologías web modernas, un diseño centrado en el usuario y herramientas como Flask y Firebase, se ha desarrollado una plataforma que permite la supervisión y control del vehículo de manera intuitiva y efectiva, asegurando una gestión óptima de sus funcionalidades y garantizando la seguridad operativa. La visualización de datos en tiempo real y las pruebas de usabilidad han sido claves para garantizar que la interfaz cumpla con los estándares de calidad y expectativas de los usuarios, proporcionando una herramienta robusta y confiable para el monitoreo y control del vehículo autónomo.

7. Discusión y Conclusiones

7.1 Resultado Principales

El desarrollo del proyecto “Asistente inteligente para movilidad, gestión energética y efectividad de servicio de un vehículo eléctrico autónomo” cumplió los objetivos generales y específicos planteados en el capítulo inicial, demostrando avances significativos en la integración de tecnologías para la movilidad autónoma y sostenible. Sin embargo, el camino hacia estos logros implicó enfrentar y superar diversas dificultades técnicas y operativas, lo que enriqueció la experiencia y los aprendizajes derivados del proyecto.

Cumplimiento de Objetivos



- **Desarrollo e implementación del sistema de control inalámbrico basado en Internet:**

Se logró integrar un sistema que combina control remoto mediante OpenVPN y un dispositivo auxiliar de corto alcance con ESP8266, brindando flexibilidad y seguridad. Este sistema cumple con el objetivo de optimizar la movilidad del vehículo y garantizar una comunicación efectiva entre sus componentes. En donde se implementó un sistema de control inalámbrico utilizando un módem 4G el cual permitía la conexión con otros equipo mediante la VPN “OpenVPN” la cual proporcionar una conexión segura y eficiente. Esto permitió el control remoto del vehículo a través de una interfaz web intuitiva. Sin embargo, la latencia sigue siendo un desafío, especialmente en condiciones de alta demanda de datos, lo que puede afectar la respuesta en tiempo real del sistema.

Por otra parte se desarrolló un dispositivo auxiliar de control de corta distancia utilizando módulos ESP8266, asegurando una comunicación fiable en un rango de 10 a 20 metros.

- **Modelo matemático y validación experimental:**

Se desarrolló un modelo matemático que describe con precisión el comportamiento dinámico del vehículo, validado mediante simulaciones y pruebas experimentales. Incluyendo la caracterización de los motores de corriente continua de imanes permanentes (DC-PM), la dinámica del vehículo y la integración de componentes del sistema como la batería y los controladores. Esto permitió predecir el rendimiento del sistema bajo distintas condiciones, sentando una base sólida para futuras optimizaciones y análisis.

El modelo del motor DC mostró una precisión del 97% en la predicción del consumo energético bajo condiciones constantes. Este resultado indica que el modelo es altamente confiable en escenarios estables, lo cual es crucial para la planificación y gestión energética del vehículo. Sin embargo, en situaciones de alta demanda, se observó una subestimación del 5% en el consumo energético. Esta discrepancia sugiere la necesidad de ajustar el modelo para mejorar su precisión en condiciones extremas.

- **Selección e integración de sensores y actuadores:**

Se implementaron sensores económicos para monitorear variables críticas, como velocidad, corriente y distancia, cumpliendo con los objetivos iniciales. Aunque se identificaron limitaciones en la precisión de los componentes, estos fueron suficientes para demostrar la viabilidad del sistema. .

- **Interfaz de usuario intuitiva para monitoreo y control:**

Se diseñó una interfaz basada en tecnologías web modernas, que permite a los usuarios controlar y monitorear el vehículo en tiempo real de forma sencilla y eficiente. La integración con Firebase y Google Cloud asegura un acceso continuo y seguro a los datos. Para esto se utilizaron tecnologías web modernas como HTML5, CSS3 y JavaScript, permitiendo el monitoreo y control efectivo del sistema en tiempo real. La interfaz ofrece visualización en tiempo real de datos críticos, control manual del vehículo y acceso a cámaras de monitoreo.

Impacto del Proyecto:

Este proyecto establece una base sólida para futuras investigaciones en el campo de los vehículos eléctricos autónomos, especialmente en la integración de sistemas de control inalámbrico y monitoreo en tiempo real. La validación experimental exhaustiva del modelo matemático y los sistemas de control añade un valor significativo y proporciona una plataforma para desarrollos futuros.

La optimización de la gestión energética y la reducción del consumo de energía contribuyen a la sostenibilidad ambiental. La implementación de vehículos eléctricos autónomos representa un paso importante hacia la movilidad sostenible en nuestras ciudades gracias a la facilidad de poder definir modos de conducción que optimizan el consumo energético.

Las tecnologías implementadas pueden ser aplicadas en diversos entornos, mejorando la movilidad urbana y ofreciendo soluciones eficientes para la logística y transporte en áreas confinadas como centros comerciales, universidades y parques industriales. La implementación de sistemas de control inalámbrico y la interfaz de usuario intuitiva hacen que estas tecnologías sean accesibles y prácticas para una amplia gama de aplicaciones.



Limitaciones del Hardware:

La utilización de sensores y actuadores de bajo costo puede afectar la precisión y fiabilidad de los datos recolectados. Estos componentes económicos, aunque accesibles, presentan limitaciones en términos de exactitud y durabilidad, lo que puede influir en el rendimiento global del sistema.

Latencia en la Comunicación:

La comunicación basada en Raspberry Pi 4 presenta limitaciones de latencia, afectando la respuesta en tiempo real para tareas complejas. La capacidad de procesamiento y la eficiencia de transmisión de datos de la Raspberry Pi 4, aunque adecuada para muchas aplicaciones, puede no ser suficiente para las exigencias de un sistema de control autónomo en tiempo real.

7.2 Conclusiones Generales y Cierre del Proyecto

El proyecto ha demostrado la viabilidad de sistemas de control inalámbrico y una interfaz de usuario para vehículos autónomos, proporcionando una base sólida para futuras investigaciones. La implementación exitosa de estas tecnologías muestra el potencial de mejorar significativamente la movilidad urbana y la gestión energética de vehículos eléctricos.

Las aplicaciones y beneficios del proyecto en la industria son vastos, mejorando la movilidad urbana y optimizando el transporte en diversos entornos. Las tecnologías desarrolladas pueden aplicarse en múltiples escenarios, desde la logística hasta el transporte público, ofreciendo soluciones innovadoras y eficientes.

Este proyecto ha avanzado significativamente en la implementación de tecnologías para vehículos autónomos eléctricos. Agradezco a todos los colaboradores y financiadores. Espero que los resultados obtenidos inspiren futuras investigaciones en este campo apasionante.

Este análisis proporciona una conclusión detallada y crítica del proyecto, evaluando sus resultados, impacto y sugerencias para futuras investigaciones, asegurando una comprensión profunda y profesional de los hallazgos y su relevancia en el campo de los vehículos autónomos.

Recomendaciones para Trabajos Futuros:

- **Mejorar la Precisión de los Sensores:**

Invertir en sensores de mayor precisión para mejorar la fiabilidad de los datos recogidos. Utilizar sensores con mejores especificaciones técnicas y mayor resolución puede aumentar significativamente la exactitud del monitoreo del vehículo autónomo.

- **Optimización de la Latencia:**

Investigar métodos avanzados para reducir la latencia en la comunicación inalámbrica, posiblemente integrando tecnologías de 5G. La adopción de redes más rápidas y eficientes puede mejorar la capacidad de respuesta en tiempo real del sistema de control.

- **Ampliar el Alcance del Sistema:**

Explorar la integración de inteligencia artificial para mejorar la toma de decisiones autónomas del vehículo. Implementar algoritmos de machine learning puede optimizar el rendimiento y adaptabilidad del vehículo en diversos entornos operativos.

- **Integración de Algoritmos de Machine Learning:**

Para la optimización continua del rendimiento del vehículo. Estos algoritmos pueden analizar datos históricos y en tiempo real para ajustar dinámicamente los parámetros operativos y mejorar la eficiencia.

- **Ampliación del Sistema de Monitoreo:**

Incluir más variables críticas y sensores avanzados para una supervisión más completa y detallada. La incorporación de nuevos tipos de sensores, como LiDAR y cámaras térmicas, puede proporcionar una visión más integral del entorno y el estado del vehículo.



Referencias

- [1] OpenVPN Technologies, Inc. (n.d.). *OpenVPN Documentation*. Recuperado de <https://openvpn.net/community-resources/how-to/>.
- [2] Anrigo, Ph; Gouget,N; Maignan, D. (2015). ADAS & Autonomous Vehicle. Test & Validations Approach.
- [3] Vlacic, L.; Thomas, D.; Pérez, J. (2011). Real time co-operative decision making & control systems
- [4] Klaus Stricker, Tom Wendt, Wilko Stark, Mark Gottfredson, Raymond Tsang (2020),” Electric and autonomus Vehicles:The Future is Now”, dirección <https://www.bain.com/insights/electric-and-autonomous-vehicles-the-future-is-now/>.
- [5] Derek Chiao. (2024). “Autonomous vehicles moving forward: Perspectives from industry leaders”, dirección <https://www.mckinsey.com/features/mckinsey-center-for-future-mobility/our-insights/autonomous-vehicles-moving-forward-perspectives-from-industry-leaders>
- [6] Ogata, K. (2010). *Modern Control Engineering* (5th ed.). Prentice Hall
- [7] StartUs Insights. (2023). 8 autonomous vehicle trends in 2023. Recuperado de <https://www.startus-insights.com/innovators-guide/autonomous-vehicle-trends/>.
- [8] Google. (n.d.). *Firestore Documentation*. Recuperado de <https://firebase.google.com/docs>.
- [9] Google. (n.d.). *BigQuery Documentation*. Recuperado de <https://cloud.google.com/bigquery/docs>.
- [10] Raspberry Pi Foundation. (n.d.). *Raspberry Pi Documentation*. Recuperado de <https://www.raspberrypi.org/documentation/>.
- [11] Raspberry Pi Foundation. (n.d.). *Raspbian Documentation*. Recuperado de <https://www.raspberrypi.org/documentation/raspbian/>.
- [12] Pandas development team. (n.d.). *pandas: Python Data Analysis Library*. Recuperado de <https://pandas.pydata.org/docs/>.
- [13] Python Software Foundation. (n.d.). *Python Documentation*. Recuperado de <https://docs.python.org/3/>.
- [14] World Wide Web Consortium (W3C). (n.d.). *HTML5 Documentation*. Recuperado de <https://www.w3.org/TR/html5/>.
- [15] Pallets Projects. (n.d.). *Flask Documentation*. Recuperado de <https://flask.palletsprojects.com/en/2.0.x/>.
- [16] StartUs Insights. (2023). 8 autonomous vehicle trends in 2023. Recuperado de <https://www.startus->

insights.com/innovators-guide/autonomous-vehicle-trends/.

- [17] Arduino. (n.d). *Arduino Documentation*. Recuperado de <https://www.arduino.cc/en/Reference/HomePage>.
- [18] <https://roboticoss.com/la-cinematica-en-la-robotica/>
- [19] Joukhadar, A., Kass Hanna, D., & Zemouche, A. (2018). UKF-based enhancement and ROS implementation of 4-WDDMR localization with advanced control system. *Cogent Engineering*, 5(1). <https://doi.org/10.1080/23311916.2018.1481561>
- [20] Matlab (n.d) <https://la.mathworks.com/help/simscape/ug/cascaded-rc-circuit-in-simulink-and-simscape.html>
- [21] Matlab (n.d) Vehicle body with two axles in longitudinal motion <https://la.mathworks.com/help/sdl/ref/vehiclebody.html>



