

Instituto Tecnológico de Costa Rica
Carrera de Ingeniería Mecatrónica



Diseño de un Sistema de Control de Velocidad para un Rover de Aplicaciones Lunares

**Informe de Proyecto de Graduación para optar por el título de Ingeniero en
Mecatrónica con el grado académico de Licenciatura**

María Fernanda del Barco León

Cartago, junio de 2025

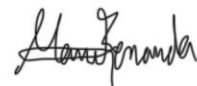


Esta obra está bajo una Licencia Creative Commons Atribución- NoComercial-CompartirIgual 4.0 Internacional.

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.



Cartago, 25 de junio de 2025

María Fernanda Del Barco León
Ced: 1-1758-0496

**INSTITUTO TECNOLÓGICO DE COSTA RICA
PROGRAMA DE LICENCIATURA EN INGENIERÍA MECATRÓNICA
PROYECTO FINAL DE GRADUACIÓN
ACTA DE APROBACIÓN**

El profesor asesor del presente trabajo final de graduación, indica que el documento presentado por el estudiante cumple con las normas establecidas por el programa de Licenciatura en Ingeniería Mecatrónica del Instituto Tecnológico de Costa Rica para ser defendido ante el jurado evaluador, como requisito final para aprobar el curso Proyecto Final de Graduación y optar así por el título de Ingeniero(a) en Mecatrónica, con el grado académico de Licenciatura.

Estudiante: María Fernanda Del Barco León

Proyecto: Diseño de un Sistema de Control de Velocidad para un Rover de Aplicaciones Lunares



TEC | Tecnológico
de Costa Rica
Ingeniería Mecatrónica

Dr. Ing. Juan Luis Crespo Mariño
Asesor

Cartago, 25 de junio del 2025


INSTITUTO TECNOLÓGICO DE COSTA RICA
PROGRAMA DE LICENCIATURA EN INGENIERÍA MECATRÓNICA
PROYECTO FINAL DE GRADUACIÓN
ACTA DE APROBACIÓN

Proyecto final de graduación defendido ante el presente jurado evaluador como requisito para optar por el título de Ingeniero(a) en Mecatrónica con el grado académico de Licenciatura, según lo establecido por el programa de Licenciatura en Ingeniería Mecatrónica, del Instituto Tecnológico de Costa Rica.

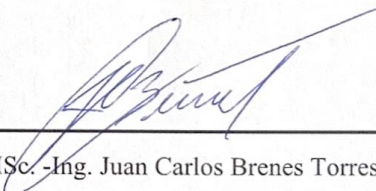
Estudiante: María Fernanda Del Barco León

Proyecto: Diseño de un Sistema de Control de Velocidad para un Rover de Aplicaciones Lunares

Miembros del jurado evaluador



Ing. Eduardo José Asenjo Castillo
Jurado



MSc. -Ing. Juan Carlos Brenes Torres
Jurado

Los miembros de este jurado dan fe de que el presente proyecto final de graduación ha sido aprobado y cumple con las normas establecidas por el programa de Licenciatura en Ingeniería Mecatrónica.

Cartago, 25 de junio del 2025

Resumen

Este proyecto aborda el diseño de un sistema de control de velocidad para un Rover orientado a aplicaciones lunares, considerando las particularidades del entorno y su implementación práctica. Se identificó que el prototipo original no era adecuado para validación, por lo que se seleccionó uno nuevo y se establecieron especificaciones de desempeño acordes con los casos de uso definidos. Se aplicó la metodología de diseño de Ulrich y Eppinger para estructurar el proceso, incluyendo la selección de conceptos, el desarrollo del modelo y la validación. El modelado del sistema se realizó a partir de datos experimentales, obteniendo una función de transferencia discreta que sirvió como base para el diseño del controlador. Se desarrolló un controlador PID utilizando herramientas de simulación. La implementación se realizó en un Arduino Uno, integrando sensado, control y registro de datos. Las pruebas se llevaron a cabo sobre superficies con y sin obstáculos, incluyendo arena y piedras, para evaluar el desempeño del sistema en condiciones similares a las reales. Los resultados muestran un cumplimiento parcial de las especificaciones, con oportunidades de mejora identificadas para futuras iteraciones del diseño.

Palabras clave: Rover, control PID, velocidad, superficie lunar, sistemas multiagente.

Abstract

This project addresses the design of a velocity control system for a Rover intended for lunar applications, taking into account both the specific environmental requirements and practical implementation constraints. It was determined that the original prototype was unsuitable for validation, leading to the selection of a new one and the establishment of performance specifications aligned with the defined use cases. The Ulrich and Eppinger design methodology was applied to structure the development process, including concept selection, model development, and validation. The system modeling was based on experimental data, resulting in a discrete transfer function used as the foundation for controller design. A PID controller was developed using simulation tools. The implementation was carried out on an Arduino Uno, integrating sensing, control, and data logging. Tests were conducted on surfaces with and without obstacles, including sand and rocks, to assess system performance under conditions similar to those expected in the final application. The results indicate partial compliance with the established specifications, along with opportunities for improvement identified for future design iterations.

Keywords: Rover, PID control, velocity, lunar surface, multi-agent systems.

A mis papás, que me permitieron llegar hasta aquí con su esfuerzo y apoyo incondicional.

Agradecimientos

Quiero agradecer al Laboratorio de Sistemas Espaciales y al Dr. Johan Carvajal por confiar en mí y apoyarme a lo largo de mis años como estudiante.

A mis amistades y compañeros que me ayudaron a sobrellevar los momentos más difíciles y me hicieron disfrutar el camino.

A Carlos, por acompañarme y enseñarme a creer en mí.

Tabla de contenidos

<i>Lista de figuras</i>	<i>v</i>
<i>Lista de tablas</i>	<i>viii</i>
<i>Lista de abreviaciones</i>	<i>ix</i>
Capítulo 1. Introducción	1
1.1 Antecedentes	1
1.2 Descripción del problema a resolver	3
1.3 Objetivos del proyecto	4
1.3.1 Objetivo General.....	4
1.3.2 Objetivos Específicos	5
1.4 Estructura del documento	5
Capítulo 2. Marco Teórico	6
2.1 Rovers para exploración espacial	6
2.1.1 Arquitectura Rocker-Bogie	7
2.2 Paradigmas multiagente	9
2.3 Estado del arte de sistemas de control de velocidad en rovers de exploración planetaria basados en sistemas con seis ruedas	10
Capítulo 3. Metodología	17
3.1 Identificación de necesidades	17
3.2 Establecimiento de especificaciones	18
3.3 Generación de conceptos	19
3.4 Selección y evaluación de conceptos	19
3.5 Pruebas y validación	20
3.6 Establecer especificaciones finales	20
Capítulo 4. Especificaciones técnicas y parámetros operativos	21

4.1	Diagnóstico del Rover	21
4.1.1	Descripción general del Rover y sus subsistemas	23
4.2	Identificación de necesidades.....	31
4.3	Establecimiento de especificaciones	33
4.3.1	Selección de valores objetivo y marginales para la métrica de mapeo del sistema a agentes.....	34
4.3.2	Selección de valores objetivo y marginales para la métrica sobre el error de medición de velocidad.....	35
4.3.3	Selección de valores objetivo y marginales para la métrica del tiempo de estabilización.....	36
4.3.4	Selección de valores objetivo y marginales para la métrica del tiempo de respuesta	37
4.3.5	Selección de valores objetivo y marginales para la métrica de sobreimpulso	37
4.3.6	Selección de valores objetivo y marginales para la métrica sobre el desempeño en al menos dos superficies distintas.....	38
4.4	Concepto de operaciones (ConOps).....	40
Capítulo 5. Diseño del sistema de control de velocidad.....		42
5.1	Descomposición en subfunciones.....	42
5.2	Generación y selección de concepto para sistema de realimentación de velocidad	44
5.2.1	Acelerómetro	44
5.2.2	Combinación de sensores con filtro de Kalman	45
5.2.3	Motores con encoders.....	45
5.2.4	Sensores tipo Hall	46
5.2.5	Selección de concepto de realimentación de velocidad	47
5.3	Generación y selección de concepto para controlador	49
5.3.1	PI.....	50
5.3.2	PID	51
5.3.3	I-PD.....	52
5.3.4	Realimentación de estado integral.....	53
5.3.5	Selección de concepto de controlador	54
5.4	Almacenamiento de datos experimentales	58
5.5	Propuesta de diseño del sistema de realimentación de velocidad	61
5.5.1	Principio de funcionamiento	61
5.5.2	Implementación física del sistema de sensado	63
5.5.3	Validación del sistema de sensado.....	69
5.6	Propuesta de diseño del controlador	71

5.6.1	Modelado del sistema.....	72
5.7	Diseño del controlador PID	76
5.8	Implementación en el Rover	81
5.9	Validación del mapeo a agentes del sistema	82
Capítulo 6. Resultados y análisis		87
6.1	Diseño de banco de pruebas.....	87
6.1.1	Diseño de rampa.....	87
6.1.2	Diseño de superficie sin obstáculos.....	92
6.1.3	Diseño de superficie con obstáculos.....	94
6.2	Pruebas de validación	95
6.2.1	Casos de prueba	95
6.2.2	Prueba de validación de error de velocidad	100
6.2.3	Prueba de validación de tiempo de estabilización	102
6.2.4	Prueba de validación de tiempo de respuesta.....	109
6.2.5	Prueba de validación de sobreimpulso.....	112
6.2.6	Prueba de validación de desempeño en distintas superficies	114
6.2.7	Resumen de validación.....	116
6.8	Análisis económico	116
Capítulo 7. Conclusiones y recomendaciones.....		122
7.1	Conclusiones.....	122
7.2	Recomendaciones	124
Bibliografía.....		125
Apéndices.....		129
Apéndice A. Concepto de Operaciones		129
Apéndice B.....		137
Apéndice C		139
Apéndice D		144
Apéndice E.....		144
Anexos.....		145

Anexo A. Lista completa de partes del Rover. 145

Lista de Figuras

Figura 1. Frentes de trabajo del proyecto de investigación Elanav.	3
Figura 2. Diagrama de líneas de la arquitectura Rocker-Bogie. Fuente: [6].	8
Figura 3. Modelo computacional del rover Spirit y su plataforma de aterrizaje, combinados con datos reales del terreno marciano en 3D obtenidos por sus cámaras Fuente: [10].	11
Figura 4. Modelo 3D del rover Curiosity. Fuente: [12].	13
Figura 5. Rover Perseverance y sus componentes. Fuente: [14].	14
Figura 6. Rover Yutu sobre la superficie de la Luna. [16].	15
Figura 7Rover Pragyán. Fuente: [17].	16
Figura 8. Etapas de la metodología de diseño utilizada. Fuente: [18].	17
Figura 9. Primer prototipo considerado.	22
Figura 10. SunFounder GalaxyRVR Rover Kit. [19].	23
Figura 11. Suspensión Rocker-Bogie del Rover. [19].	23
Figura 12. Rueda del Rover. [20].	24
Figura 13. Uno de los seis motores del Rover. [20].	25
Figura 14. Placa de desarrollo SunFounder R3, basada en Arduino. [20].	26
Figura 15. Placa de expansión GalaxyRVR Shield. [20].	27
Figura 16. Ubicación de los controladores de motor en el GalaxyRVR Shield. [19].	28
Figura 17. Flujo de señal hacia los motores. [19].	28
Figura 18. Cámara ESP32 [19].	29
Figura 19. Servo motor. [19].	29
Figura 20. Sensor infrarrojo. [19].	30
Figura 21. Sensor ultrasónico. [19].	30
Figura 22. Matriz de necesidades y métricas.	33
Figura 23. Diagrama de caja negra del sistema.	43
Figura 24. Diagrama de descomposición funcional.	43
Figura 25. Árbol de clasificación de conceptos para el sensado de velocidad.	44
Figura 26. Árbol de clasificación de conceptos para el controlador.	50
Figura 27. Diagrama de bloques de un controlador proporcional-integral. Fuente: [30].	51

Figura 28. Diagrama de bloques de un controlador proporcional-integral-derivativo. Fuente: [30].....	52
Figura 29. Diagrama de bloques de un controlador I-PD. Fuente: [32].....	53
Figura 30. Diagrama de bloques de un controlador de realimentación de estado integral. Fuente: [34].....	54
Figura 31. Vista frontal y posterior del módulo OpenLog de SparkFun.	59
Figura 32. Ubicación y pines del conector del sensor ultrasónico utilizado para la conexión del OpenLog.	60
Figura 33. Tarjeta Maxell microSD de 16 GB Clase 10 utilizada para el almacenamiento de datos.....	60
Figura 34. Módulo de sensor Hall modelo KY-024.	64
Figura 35. Montaje del sensor Hall en el Rover.	65
Figura 36. Ubicación y pines del conector del sensor infrarrojo izquierdo utilizado para la conexión del sensor Hall.	66
Figura 37. Imán de neodimio en forma de disco 8x2 mm, clasificación N35 (fuerza magnética estándar).....	67
Figura 38. Posicionamiento de los cuatro imanes para sensado de velocidad angular.....	68
Figura 39. Datos de estimación y validación del modelo para los motores.....	73
Figura 40. Señal de salida filtrada.	74
Figura 41. Ajuste del modelo estimado a los datos experimentales de validación.	75
Figura 42. Diagrama de bloques del lazo de control propuesto para la velocidad lineal del Rover.....	76
Figura 43. Respuesta simulada del sistema ante una entrada escalón de velocidad lineal.	78
Figura 44. Lugar de las raíces del sistema.	79
Figura 45. Respuesta del controlador implementado en el Rover.	80
Figura 46. Metodología para mapeo a agentes. Fuente: [1].....	84
Figura 47. Arquitectura multiagente.....	85
Figura 48. Geometría de la rampa de pruebas.....	89
Figura 49. Base de la rampa.....	91
Figura 50. Vista lateral de la rampa utilizada para la validación.	92
Figura 51. Alfombra de grama artificial utilizada.	93

Figura 52. Rover sobre superficie con arena.....	94
Figura 53. Rover sobre superficie con obstáculos distribuidos sobre césped artificial.	95
Figura 54. Respuesta caso 1.....	97
Figura 55. Respuesta caso 2.....	98
Figura 56. Respuesta caso 3.....	99
Figura 57. Respuesta caso 4.....	100

Lista de tablas

Tabla 1. Jerarquía de necesidades.....	32
Tabla 2. Importancia de necesidades.	32
Tabla 3. Especificaciones objetivo y marginales.....	39
Tabla 4. Matriz de selección de conceptos de realimentación de velocidad.	48
Tabla 5. Matriz de selección de conceptos de controlador.....	56
Tabla 6. Resumen y estadísticas de los resultados de la validación del sistema de sensado de velocidad angular.....	71
Tabla 7. Relación entre velocidad lineal, velocidad angular y señal PWM aplicada.	72
Tabla 8. Parámetros del controlador PID.....	80
Tabla 9. Características de los casos de prueba utilizados.....	96
Tabla 10. Resultados del error de velocidad del sistema en distintos casos de uso.	101
Tabla 11. Resultados del tiempo de estabilización del sistema en distintos casos de uso.	103
Tabla 12. Comparación de tiempos de estabilización a distintos márgenes de tolerancia.....	106
Tabla 13. Resultados del tiempo de respuesta del sistema en distintos casos de uso.....	110
Tabla 14. Resultados del sobreimpulso del sistema en distintos casos de uso.....	113
Tabla 15. Variación porcentual entre distintas superficies.....	115
Tabla 16. Nivel de cumplimiento de métricas.....	116
Tabla 17. Desglose de costos del proyecto.	117
Tabla 18. Estimación de recursos generados por el proyecto.	120

Lista de abreviaciones

LRV: Apollo Lunar Roving Vehicle, Vehículo explorador lunar

CNSA: Administración Espacial Nacional China

CADRE: Exploración robótica distribuida autónoma y cooperativa

CONOPS: Concepto de operaciones

ELANav: Biblioteca embebida para la navegación de sistemas espaciales de misión crítica

FDIR: Detección, aislamiento y recuperación de fallos

FPGA: Matriz de compuertas programables en campo

ICSP: Conector de programación en circuito

IR: Sensores infrarrojos

ISRO: Organización de Investigación Espacial de la India

I-PD: Controlador proporcional-derivativo con acción integral separada

JAXA: Agencia espacial japonesa

Kd: Constante derivativa del controlador

Ki: Constante integral del controlador

Kp: Constante proporcional del controlador

MAES: Marco de trabajo multiagente para sistemas embebidos

MAPE: Error porcentual absoluto medio

MAS: Sistemas multiagente

MER: Mars Exploration Rover

MSL: Mars Science Laboratory

NASA: Administración nacional de aeronáutica y el espacio

N: Constante del filtro derivativo

PID: Controlador proporcional-integral-derivativo

PI: Controlador proporcional-integral

PWM: Modulación por ancho de pulso

REI: Requerimiento de entrada

RPM: Revoluciones por minuto

SETEC Lab: Laboratorio de Sistemas Espaciales del Instituto Tecnológico de Costa Rica

Capítulo 1. Introducción

Este capítulo presenta el contexto y la motivación del proyecto, enmarcado en la investigación sobre software embebido para sistemas espaciales de misión crítica.

1.1 Antecedentes

El presente proyecto se desarrolla en el marco del Laboratorio de Sistemas Espaciales del Instituto Tecnológico de Costa Rica (SETEC Lab). Este laboratorio se dedica a la investigación y desarrollo de tecnologías aeroespaciales, con un enfoque particular en la navegación y control de sistemas espaciales de misión crítica, es decir, aquellos cuya falla puede comprometer el éxito de toda la misión, la integridad del equipo o la seguridad de la operación. Como parte de esta línea de investigación, el Dr. Johan Carvajal ha desarrollado una metodología basada en arquitecturas de software orientadas a agentes, cuyo objetivo es mejorar la tolerancia a fallos en satélites y otros sistemas espaciales.

La base conceptual de esta metodología se encuentra en la tesis doctoral del Dr. Johan Carvajal, titulada *Agent-based Architectures Supporting Fault-Tolerance in Small Satellites* (arquitecturas basadas en agentes para la tolerancia a fallos en pequeños satélites) [1]. En este trabajo, se plantea la adopción de sistemas multiagente (MAS) para la implementación de software embebido en pequeños satélites, con el fin de reducir la complejidad del software y mejorar su capacidad de detección, aislamiento y recuperación de fallos (FDIR). Entre sus contribuciones, se destacan un modelo de comunicación de buses distribuido, un algoritmo de detección y corrección de errores en sensores, y una metodología para el desarrollo de software basado en agentes.

Actualmente, esta línea de investigación ha evolucionado hacia el proyecto *Biblioteca embebida para la navegación de sistemas espaciales de misión crítica*, o ELANav. Su objetivo

principal es crear una biblioteca de software embebido, con capacidades de redundancia y adaptabilidad, utilizando el paradigma de múltiples agentes de software [2].

En este contexto, se utilizará *MAES (Multi-Agent Framework for Embedded Systems)*, o marco multiagente para sistemas embebidos, un marco de software diseñado para la implementación de sistemas multiagente en plataformas embebidas con requisitos en tiempo real. *MAES* surgió como una respuesta a la creciente complejidad del software en misiones espaciales, donde los sistemas tradicionales de control centralizado están siendo reemplazados por arquitecturas distribuidas [3].

Su evolución ha pasado por diversas adaptaciones en términos de lenguaje de programación y sistema operativo:

- **VxMAES:** Creada en 2017, originalmente implementada en C++ para una arquitectura basada en Texas Instruments (*TI-RTOS*) y utilizando el sistema operativo VxWorks, el cual es ampliamente utilizado en misiones espaciales debido a su certificación para vuelo [3].
- **Migración a FreeRTOS:** En 2021, se adaptó MAES a un sistema operativo de código abierto, FreeRTOS, manteniendo su implementación en C++. [4]
- **Conversión a C:** Posteriormente en 2022, se realizó la transición de MAES de C++ a C, debido a que el lenguaje C es el predominante en sistemas espaciales por su facilidad de certificación. [4]
- **Regreso a VxWorks:** En 2024, debido a las limitaciones de certificación de FreeRTOS, el sistema fue migrado nuevamente a VxWorks, asegurando su compatibilidad con aplicaciones aeroespaciales certificadas. [4]

Actualmente, los esfuerzos de investigación del SETEC Lab se centran en utilizar la metodología desarrollada por el Dr. Johan Carvajal para portar la biblioteca MAES a nuevas plataformas y desarrollar aplicaciones para su implementación.

En paralelo, se trabajará en el desarrollo e implementación de una plataforma de hardware para validar casos de uso reales del sistema. Se planea construir una plataforma espacial tipo Rover, que represente una arquitectura característica de un sistema espacial de misión crítica, para la cual se desarrollarán modelos de estimación y control de navegación utilizando la herramienta Matlab Simulink. Estos modelos deberán representar escenarios reales del sistema y estar

alineados con la arquitectura de hardware seleccionada para la implementación del Rover. Todo esto con el fin de desarrollar una metodología para la traducción de los modelos de navegación implementados en Simulink a la biblioteca embebida de agentes de software para sistemas de misión crítica [2].

Debido al reciente inicio del proyecto ELANav, aún no se cuenta con un prototipo oficial del Rover que será desarrollado en el marco del proyecto. Sin embargo, se dispone de un prototipo previo, el cual, si bien no representa un diseño final ni completamente alineado con los objetivos de ELANav, servirá como punto de partida para la validación inicial de modelos y pruebas de integración. Este prototipo permitirá evaluar aspectos clave del sistema mientras se avanza en el desarrollo del diseño definitivo del Rover.

1.2 Descripción del problema a resolver

El equipo del proyecto ELANav se encuentra estructurado en diferentes áreas de trabajo que se relacionan, como se muestra en la Figura 1.

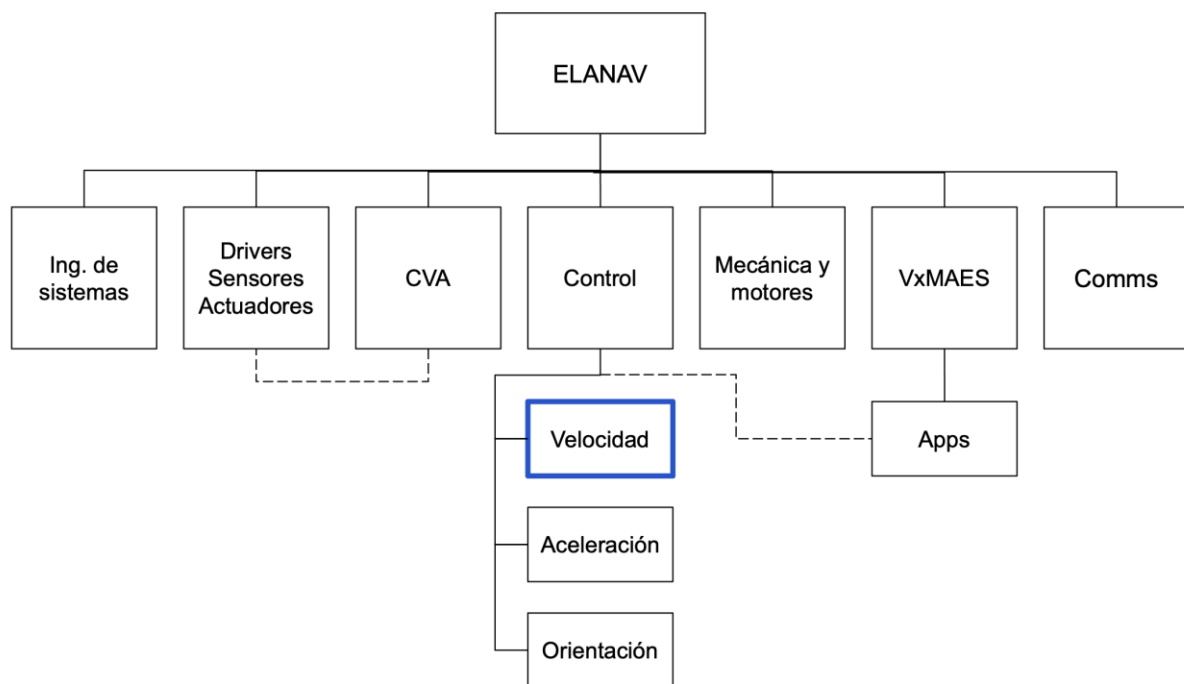


Figura 1. Frentes de trabajo del proyecto de investigación Elanav.

El subproyecto de control de velocidad, que es el foco de este trabajo, se encuentra dentro del área de control y aborda el diseño de un sistema de control de velocidad para el Rover de

aplicación lunar. La implementación del control de orientación y aceleración se desarrollará posteriormente en el proyecto ELANav, utilizando como base este trabajo.

Una de las consideraciones clave es que los algoritmos desarrollados deben ser compatibles con la arquitectura de software basada en agentes de MAES. Para ello, se requiere definir los bloques funcionales del sistema de control de manera que puedan mapearse a agentes, siguiendo un enfoque modular. Un ejemplo de este paradigma es el diagrama de bloques del filtro de Kalman, en el cual los procesos de predicción y medición se combinan para optimizar la estimación de estados. De manera similar, los componentes del sistema de control del Rover deben ser analizados y agrupados en agentes para su integración en MAES.

En síntesis, el problema central del proyecto radica en la ausencia de modelos de estimación y control que permitan la navegación autónoma del Rover. Actualmente, el prototipo disponible para pruebas depende de un sistema manual de control, lo que limita su capacidad operativa. Esta falta de un sistema de control automático es incompatible con casos de usos reales, donde se requiere navegación independiente, y con los objetivos de validación del proyecto ELANav [2].

El presente proyecto se centra exclusivamente en la solución de esta problemática mediante el diseño e implementación de un sistema de control de velocidad para el Rover. Tomando en consideración que dicho sistema deberá integrarse con la arquitectura de software embebido MAES para cumplir con los requisitos definidos en la estructura del proyecto ELANav.

1.3 Objetivos del proyecto

1.3.1 Objetivo General

1. Diseñar un sistema de control de velocidad para un Rover destinado a aplicaciones lunares, considerando los requisitos operativos específicos del entorno y su implementación práctica.

1.3.2 Objetivos Específicos

1. Establecer los valores de especificaciones y parámetros que caracterizan el problema a solucionar y la operatividad de la solución.
2. Diseñar un conjunto de algoritmos de control que cumplan con los requerimientos operativos definidos para el Rover en ambiente lunar.
3. Evaluar el desempeño de los algoritmos de control propuestos mediante pruebas en condiciones de laboratorio que simulen escenarios reales.

1.4 Estructura del documento

El presente informe se organiza en capítulos que permiten una comprensión estructurada del desarrollo del proyecto. En primer lugar, esta introducción contextualiza la problemática abordada, presenta los objetivos del estudio y describe la organización del documento. Luego, el marco teórico recopila los conceptos y fundamentos necesarios para entender los principios del control de velocidad en sistemas espaciales.

La metodología describe el enfoque seguido en la investigación, incluyendo las herramientas utilizadas y el proceso de diseño e implementación. Posteriormente, el documento se estructura en tres capítulos que corresponden a los objetivos específicos del proyecto. El primer capítulo se enfoca en el establecimiento de las especificaciones y parámetros que caracterizan el problema y la operatividad de la solución. El segundo capítulo aborda el diseño y desarrollo de los algoritmos de control que cumplen con los requisitos operativos del Rover en un entorno lunar. Finalmente, el tercer capítulo presenta la evaluación del desempeño del sistema de control mediante pruebas experimentales en condiciones de laboratorio, junto con el análisis de los resultados obtenidos.

El documento concluye con el capítulo de conclusiones y recomendaciones, donde se presentan los hallazgos más relevantes y se sugieren mejoras futuras. Adicionalmente, se incluyen las referencias bibliográficas junto con los apéndices y anexos, que proporcionan información complementaria relevante para el estudio.

Capítulo 2. Marco Teórico

En esta sección se enmarcan los conceptos relevantes al desarrollo de este proyecto final de graduación y los desarrolla de forma tal que alguien con conocimiento de ingeniería pueda entender el documento en su totalidad. Se definirá el estado del arte del tema, la librería MAES para software embebido, rovers de exploración espacial y algoritmos de control para navegación espacial.

2.1 Rovers para exploración espacial

El desarrollo de sistemas espaciales no tripulados está presente desde el inicio de la carrera espacial con el lanzamiento de Sputnik 1, lanzado por la Unión Soviética el 4 de octubre de 1957. Esta primera sonda operaba como un transmisor de señales de radio [5]. Desde entonces, el desarrollo de sondas y rovers ha evolucionado significativamente, permitiendo la exploración de cuerpos celestes como la Luna, Marte y asteroides cercanos. Ejemplos notables incluyen el programa *Apollo Lunar Roving Vehicle* (vehículo explorador lunar o LRV por sus siglas en inglés), los rovers de la NASA en Marte (como Spirit, Opportunity, Curiosity y Perseverance) y las misiones chinas Chang'e con sus rovers Yutu.

Este tipo de exploración conlleva una serie de retos específicos por el ambiente en el que se desarrolla, entre ellos, ambientes hostiles, difícil acceso a soporte, comunicación con retrasos, ambientes desconocidos para los que no existen modelos completos, condiciones gravitacionales distintas a las terrestres, superficies con composiciones distintas a las presentes en la Tierra entre otros. Estas condiciones hacen que, actualmente, la exploración humana en estos ambientes sea costosa y riesgosa, por lo que se han desarrollado sistemas robóticos para cumplir esta función como las sondas espaciales y los rovers.

Un rover es un vehículo explorador diseñado para desplazarse sobre la superficie de cuerpos celestes como la Luna o Marte. A diferencia de las sondas estáticas, los rovers pueden moverse y realizar estudios en distintos puntos del terreno, lo que les permite recolectar mayor cantidad de datos sobre la geología, la atmósfera y otras condiciones ambientales. Están equipados con

sensores, cámaras, herramientas científicas y, en algunos casos, brazos robóticos para analizar muestras del suelo y realizar experimentos *in situ*. Su diseño y movilidad son variados porque deben adaptarse al entorno en el que operarán.

Los avances en movilidad, autonomía y eficiencia energética han sido clave para mejorar las capacidades de estos sistemas. Actualmente, se desarrollan rovers con mayor grado de inteligencia artificial para permitir una toma de decisiones autónoma en tiempo real, optimizando la exploración y reduciendo la dependencia de comandos remotos desde la Tierra.

2.1.1 Arquitectura Rocker-Bogie

En el diseño de rovers se consideran diversas configuraciones, como robots con ruedas, con extremidades articuladas o combinados también conocidos como *wheel-step*. Los robots con ruedas son más eficientes en términos de consumo de energía que los de extremidades, aunque presentan limitaciones de movilidad. Los sistemas híbridos *wheel-step* combinan ambas capacidades, permitiendo un desplazamiento eficiente en superficies planas e irregulares [6].

Dentro de la categoría de sistemas *wheel-step*, una de las configuraciones mecánicas más utilizadas en el diseño de rovers es la arquitectura *Rocker-Bogie*. Esta se caracteriza por su capacidad para mantener la estabilidad del vehículo en superficies irregulares, distribuyendo de manera eficiente las cargas entre las ruedas y beneficiando la tracción. El sistema fue originalmente desarrollado para las misiones de la NASA, siendo implementado en Rovers como Sojourner, Spirit, Opportunity, Curiosity y Perseverance, debido a su alta adaptabilidad frente a obstáculos [6].

El mecanismo se compone de dos elementos principales por cada lado del chasis: el *rocker* y el *bogie*. El *rocker* es un eslabón largo que se conecta al chasis mediante un diferencial pasivo, permitiendo que cuando uno de los lados se eleva al superar un obstáculo, el otro descienda, manteniendo el centro de masa relativamente estable. Este eslabón está equipado con una rueda en su extremo frontal y se articula con el *bogie* en el extremo opuesto. El *bogie* es una estructura

más corta que contiene dos ruedas y se encuentra unida al *rocker* mediante una articulación pivotante como se muestra en la Figura 2 [6].

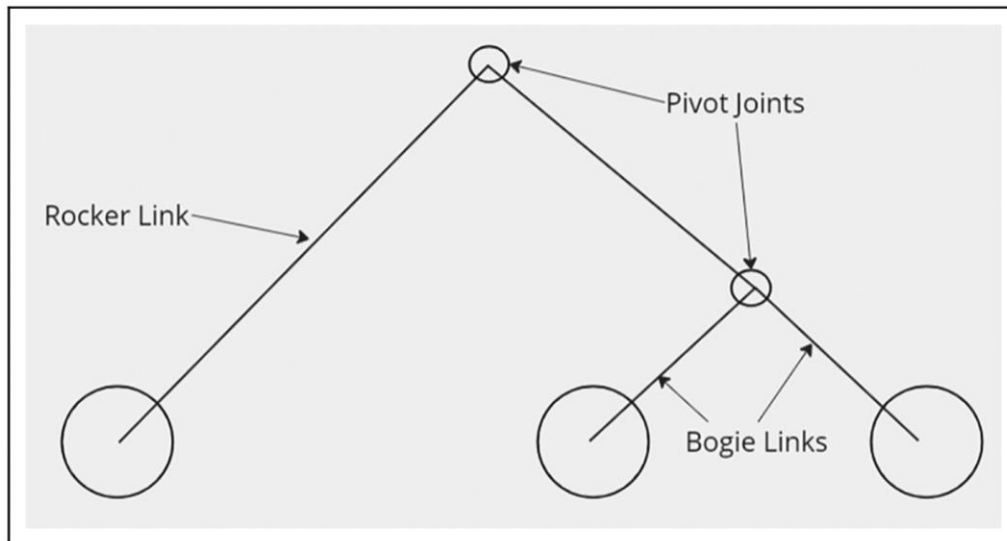


Figura 2. Diagrama de líneas de la arquitectura Rocker-Bogie. Fuente: [6].

En conjunto, cada lateral del mecanismo dispone de tres ruedas, totalizando seis en la arquitectura completa. Este diseño simétrico permite superar obstáculos de hasta una altura equivalente al diámetro de las ruedas, sin comprometer la estabilidad del chasis ni transmitir vibraciones excesivas de las ruedas a los sistemas a bordo. Además, las juntas pivotantes libres entre los enlaces permiten una rotación independiente de cada componente, lo cual actúa como sistema de amortiguación al reducir el impacto directo entre las diferentes partes [6].

Más allá de su aplicación en misiones espaciales, la arquitectura *Rocker-Bogie* ha demostrado ser eficaz en entornos terrestres exigentes, como minas, zonas forestales o regiones de difícil acceso, consolidándose como una solución robusta y versátil para plataformas móviles autónomas [6].

2.2 Paradigmas multiagente

Los sistemas multiagente están compuestos por múltiples agentes inteligentes que interactúan entre sí [7].

Un agente puede entenderse como una unidad computacional autónoma que opera dentro de un entorno específico y actúa con el propósito de cumplir sus metas de diseño. A diferencia de sistemas convencionales que simplemente transmiten información, los agentes son capaces de colaborar y competir, utilizar lenguajes con contenido semántico y tomar decisiones mediante mecanismos complejos como negociaciones, votaciones, subastas o coaliciones [7].

Desde una perspectiva funcional, un agente es capaz de combinar acciones reactivas e intencionales, adaptándose a distintas condiciones del entorno mediante procesos como planificación, aprendizaje y manejo de restricciones. Esta autonomía le permite decidir y actuar de forma independiente en situaciones variables, incluso ante eventos no previstos o fuera de control humano [7].

Al estar inmersos en un entorno, los agentes perciben parcialmente su contexto y ejecutan acciones sobre él. A pesar de que no tienen control sobre el entorno y en el mejor de los casos pueden tenerlo solo parcialmente [7].

La implementación de agentes autónomos se ve favorecida por el uso de *frameworks* (entornos de desarrollo) especializados, distribuidos en forma de bibliotecas embebidas que ofrecen interfaces de programación (APIs) para su integración y desarrollo [2].

Una aplicación concreta de este enfoque puede observarse en [3], donde se presenta una prueba de concepto de una biblioteca embebida orientada al control de satélites miniaturizados, basada en el paradigma de sistemas multiagente. En esta propuesta, el software a bordo del satélite es modelado como un sistema multiagente, donde cada agente se asigna a uno de los computadores embarcados tipo *PocketQube*, los cuales funcionan como plataformas de ejecución independientes para los agentes, actuando como contenedores lógicos. Esta arquitectura permite distribuir la lógica de control, aumentando la flexibilidad y resiliencia de la misión ante fallos o condiciones imprevistas, lo cual resulta especialmente valioso en contextos con recursos limitados como los *CubeSats*.

Este tipo de arquitectura distribuida basada en agentes puede adaptarse al diseño de sistemas de control para plataformas móviles terrestres, como un rover. En este contexto, cada módulo funcional como navegación, percepción, gestión de energía o control de velocidad, puede ser representado como un agente autónomo que interactúa con los demás en función del estado del entorno y los objetivos del sistema. Particularmente, el control de velocidad puede delegarse a un agente especializado que, además de regular la potencia entregada a los actuadores, sea capaz de coordinarse con el agente de navegación para ajustar su comportamiento en tiempo real, considerando obstáculos, condiciones del terreno o requerimientos energéticos. Esta distribución de responsabilidades facilita el diseño modular, promueve la escalabilidad del sistema y mejora la capacidad de adaptación ante escenarios dinámicos e inciertos [7].

El proyecto *Cooperative Autonomous Distributed Robotic Exploration* (Exploración robótica distribuida autónoma y cooperativa o CADRE por sus siglas en inglés) de la NASA es una demostración tecnológica orientada a validar la exploración cooperativa mediante múltiples rovers autónomos sobre la superficie lunar. Esta iniciativa propone una arquitectura distribuida en la que cada rover actúa como un agente autónomo, capaz de percibir su entorno, tomar decisiones locales y coordinar sus acciones con los demás mediante comunicación inalámbrica. El sistema está diseñado para funcionar sin supervisión continua desde la Tierra, lo que permite una respuesta adaptativa ante condiciones dinámicas e impredecibles. CADRE representa una aplicación directa del paradigma de sistemas multiagente, al asignar comportamientos individuales e interacciones coordinadas a cada Rover [8]. Esta tecnología se enmarca en los objetivos del programa Artemis, como un componente clave para misiones lunares avanzadas en las que múltiples robots colaboren para tareas de exploración científica y logística [9].

2.3 Estado del arte de sistemas de control de velocidad en rovers de exploración planetaria basados en sistemas con seis ruedas

En esta sección se presenta una revisión del estado del arte en sistemas de control de velocidad implementados en rovers de exploración planetaria con configuraciones de seis ruedas. Para

delimitar el alcance del análisis, se han considerado únicamente misiones exitosas desarrolladas y operadas a partir del año 2000, abarcando tanto vehículos enviados a Marte como a la Luna.

Entre los rovers de exploración desarrollados por la NASA y enviados a Marte en el siglo XXI, se destacan los vehículos gemelos Spirit y Opportunity, así como Curiosity y Perseverance. Todos ellos comparten una arquitectura de locomoción basada en el mecanismo rocker-bogie.

Spirit y Opportunity, lanzados por la NASA en 2003 como parte de la misión Mars Exploration Rover (MER), operaban a partir de comandos secuenciales transmitidos desde la Tierra que especificaban la información que debían recolectar sobre la superficie marciana. Una vez recibidos, los vehículos ejecutaban estos comandos de forma autónoma mediante su sistema de locomoción de seis ruedas, todas motrices, pero con capacidad de dirección únicamente en las cuatro ruedas de las esquinas. La ejecución del movimiento estaba respaldada por sensores propioceptivos y estimaciones de posición basadas en odometría inercial. En escenarios de baja tracción, o cuando la odometría de rueda se volvía poco confiable, los rovers utilizaban odometría visual a partir de imágenes estereoscópicas para estimar desplazamientos con mayor precisión. Esta capacidad permitía mantener una estimación de posición con errores acumulados inferiores al 10 % de la distancia recorrida en trayectos menores a 100 m, incluso sobre terrenos irregulares. La navegación autónoma se basaba en la generación local de mapas de transitabilidad utilizando el sistema GESTALT, selección incremental de rutas seguras y actualización continua de la posición estimada. En modo autónomo, la velocidad de desplazamiento estaba diseñada para alcanzar hasta 35 m/hora en terrenos de complejidad moderada. No obstante, estos rovers podían alcanzar una velocidad traslacional de 5 cm/s [10].

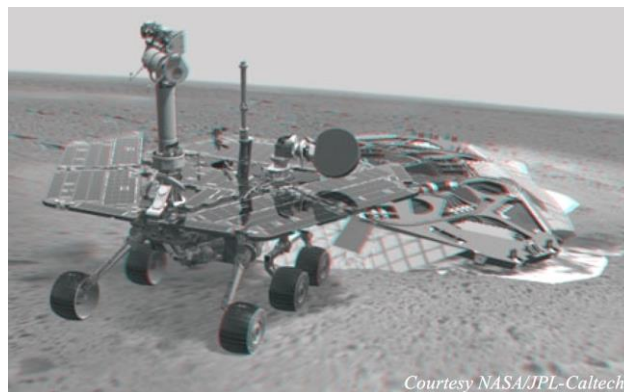


Figura 3. Modelo computacional del rover Spirit y su plataforma de aterrizaje, combinados con datos reales del terreno marciano en 3D obtenidos por sus cámaras Fuente: [10].

Curiosity, desarrollado por la NASA como parte de la misión *Mars Science Laboratory* (MSL) y aterrizado en Marte en agosto de 2012, incorporó un sistema de control de velocidad avanzado diseñado para adaptarse a la topografía del terreno en tiempo real. En respuesta al desgaste prematuro de las ruedas detectado a partir de octubre de 2013, el *Jet Propulsion Laboratory* (JPL) desarrolló e implementó una actualización del software de movilidad que permitiera reducir las cargas sobre los actuadores. Este algoritmo adapta dinámicamente la velocidad angular de cada rueda utilizando mediciones del sistema de suspensión *rocker-bogie* (ángulos y velocidades relativas) y tasas de actitud del chasis, con el fin de estimar los ángulos de contacto rueda-terreno en tiempo real. Para ello, se emplean sensores propioceptivos como codificadores de posición en las articulaciones del sistema *rocker-Bogie* y giroscopios que conforman la unidad de medición inercial (IMU). Estos datos permiten calcular no solo la orientación y el movimiento del chasis, sino también las tasas relativas de las articulaciones, fundamentales para deducir el perfil local del terreno y el comportamiento de las ruedas en contacto con él [11].

A partir de esta estimación, el sistema comanda velocidades angulares ideales bajo la suposición de no deslizamiento, ajustando cada rueda para que siga su trayectoria cinemática teórica de forma consistente con la geometría del terreno. Este control no se basa en esquemas tradicionales como PID, sino en un modelo de cinemática rígida que genera comandos de velocidad individualizados por rueda, con el objetivo de minimizar las fuerzas de interacción no deseadas. Además, el algoritmo es capaz de detectar y corregir automáticamente condiciones de pérdida de contacto, conocidas como *wheelies*, que pueden comprometer la estabilidad del vehículo [11].

Las pruebas en tierra mostraron reducciones de fuerza de hasta un 19 % en las ruedas delanteras, y de 11 % en las centrales, mientras que los datos de vuelo confirmaron una disminución del consumo de corriente asociada al par motor de un 18,7 % sobre una distancia acumulada de 3,6 km. Estos resultados evidencian una mejora sustancial en la eficiencia del control de velocidad [11].

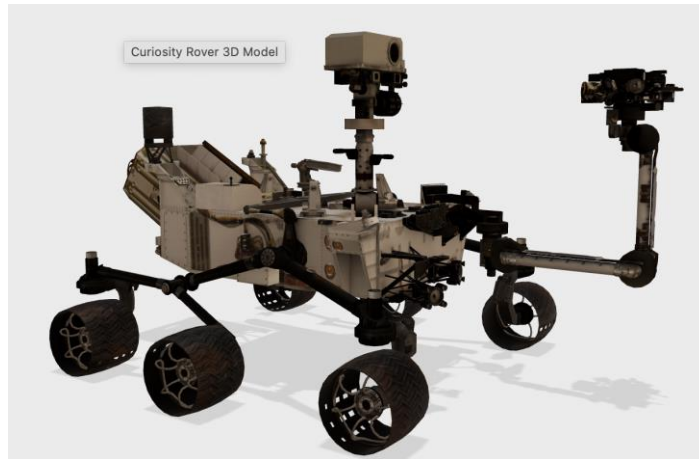


Figura 4. Modelo 3D del rover Curiosity. Fuente: [12]

Perseverance, lanzado por la NASA en 2020 como parte de la misión Mars 2020 y aterrizado en el cráter Jezero en febrero de 2021, representa un avance significativo en cuanto a autonomía en rovers de exploración planetaria. Aunque comparte varios componentes con su predecesor Curiosity, como el sistema de locomoción de seis ruedas motrices con dirección en las ruedas de las esquinas, un conjunto de controladores de motor similar y una velocidad mecánica máxima de aproximadamente 4.4 cm/s, Perseverance integra un conjunto de capacidades autónomas mucho más robustas. Su sistema AutoNav permite al vehículo conducir de forma continua y adaptativa sin detenerse para calcular nuevas trayectorias, gracias a un coprocesador dedicado conocido como *Vision Compute Element* (elemento de procesamiento de visión o VCE por sus siglas en inglés) y un FPGA integrado que acelera el procesamiento de imágenes estéreo y la odometría visual. Esta arquitectura habilita la planificación de trayectoria en tiempo real mientras el rover se desplaza, una capacidad inédita en misiones anteriores. Como resultado, AutoNav fue responsable del 88.7 % de los 17.7 km recorridos durante el primer año marciano de operación, estableciendo récords como la mayor distancia recorrida en un solo día (347.7 m) y la mayor distancia sin revisión humana (699.9 m) [13].

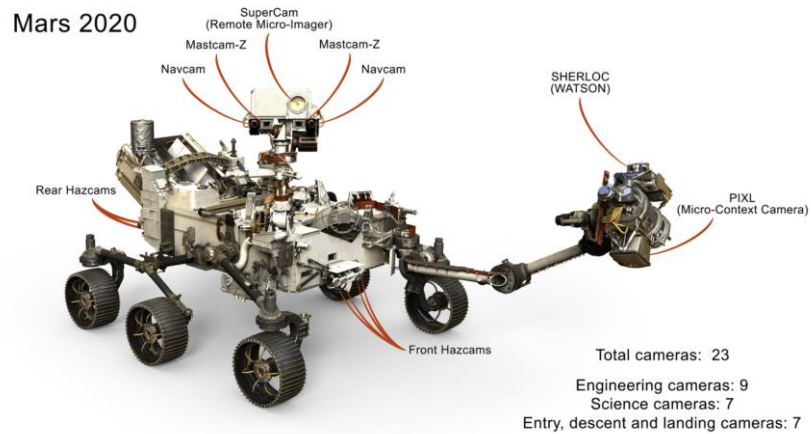


Figura 5. Rover Perseverance y sus componentes. Fuente: [14]

En cuanto al control de velocidad, Perseverance es el primer rover en conducir de manera autónoma casi a su velocidad electromecánica máxima, incluso con todas las funciones de AutoNav activadas. Esta capacidad se logra mediante una estrategia de planificación de trayectoria predictiva que calcula cada paso no en función de la posición actual, sino de la posición estimada al final del paso. A diferencia de rovers anteriores, que requerían detenerse para recalcular la trayectoria, Perseverance puede planificar y ejecutar su siguiente movimiento mientras aún está en desplazamiento. Esta capacidad le permite mantener una conducción fluida y continua, reduciendo las pausas entre comandos y evitando variaciones innecesarias en la velocidad [13].

Estas mejoras posicionan a Perseverance como el rover más avanzado hasta la fecha en cuanto a control de velocidad y navegación autónoma.

A continuación, se analizan dos casos exitosos de rovers desplegados en la superficie lunar. Se trata del rover Pragyan, desarrollado por la Organización de Investigación Espacial de la India (ISRO, por sus siglas en inglés), y del rover Yutu, diseñado por la Administración Espacial Nacional China (CNSA, por sus siglas en inglés). Ambos demuestran que, de forma similar a los modelos marcianos previamente descritos, el mecanismo de suspensión *Rocker-Bogie* es efectivo en entornos lunares.

El rover Yutu, desarrollado como parte de la misión Chang'e 3, incorpora un sistema de locomoción con seis ruedas de tracción independiente y control de dirección en cuatro de ellas.

Este diseño le permite ejecutar maniobras como giros sobre su propio eje, desplazamiento lateral y navegación sobre terreno irregular. El control de velocidad se adapta a las condiciones del entorno lunar mediante la combinación de guiado autónomo y control remoto desde Tierra, con capacidades para identificar obstáculos y planificar rutas de patrullaje en tiempo real. Para mejorar la precisión de navegación y mitigar los errores acumulados por deslizamientos o deriva del sistema inercial, Yutu combina navegación inercial con posicionamiento visual, utilizando técnicas de odometría visual basadas en la selección de puntos de correspondencia. Estos avances permiten que el sistema mantenga una trayectoria segura y estable incluso al enfrentar pendientes superiores a 20° , rocas mayores a 20 cm o cráteres de más de 2 m de diámetro [15].

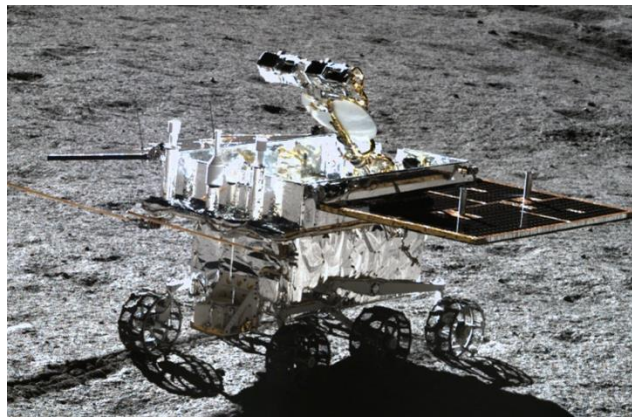


Figura 6. Rover Yutu sobre la superficie de la Luna. [16]

El rover Pragyan, desarrollado por la Organización de Investigación Espacial de la India como parte de la misión Chandrayaan-3, fue diseñado para operar durante un día lunar (aproximadamente 14 días terrestres) en la superficie de la Luna. Su sistema de locomoción se basa en una arquitectura de seis ruedas montadas sobre una suspensión pasiva tipo *Rocker-Bogie*. Cada rueda está impulsada por un motor eléctrico independiente de tipo *brushless* DC, lo que permite el control de movimiento mediante dirección diferencial, ajustando la velocidad de cada rueda para maniobrar. La velocidad de avance fue deliberadamente baja, limitada a 1 cm/s, con un alcance operativo planificado de hasta 500 metros, aunque logró recorrer 101.4 metros durante su misión [17].

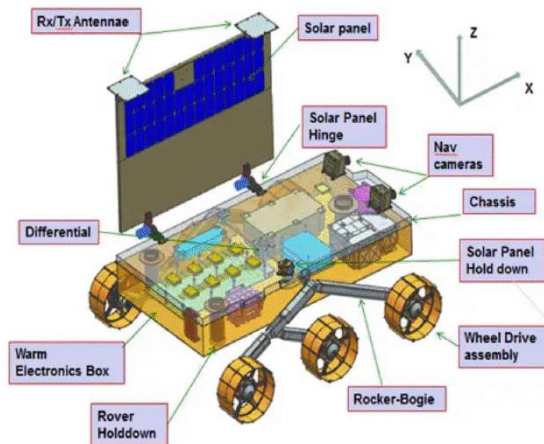


Figura 7Rover Pragyan. Fuente: [17]

La navegación fue semi-autónoma, utilizando un par de cámaras estereoscópicas frontales (NAVCAMs) para generar modelos digitales de elevación y facilitar la planificación de rutas seguras desde el centro de control terrestre. Durante su operación, el rover detectó y evitó de forma segura obstáculos como un cráter de cuatro metros de diámetro. El control de velocidad no empleó algoritmos avanzados como control adaptativo o predictivo, sino una lógica determinista supervisada, orientada a garantizar la seguridad y eficiencia energética en una misión de demostración tecnológica. La comunicación se realizaba exclusivamente a través del módulo de alunizaje Vikram, que retransmitía los datos hacia la Tierra [17].

Capítulo 3. Metodología

La metodología utilizada en este trabajo se basa en el enfoque propuesto por Ulrich y Eppinger en su libro “Diseño y desarrollo de productos” [18]. Esta metodología permite estructurar el desarrollo de soluciones mediante etapas concretas que consisten en identificar las necesidades del cliente, establecer criterios medibles, explorar alternativas de diseño y validar el desempeño del diseño propuesto, como se muestra en la Figura 8.

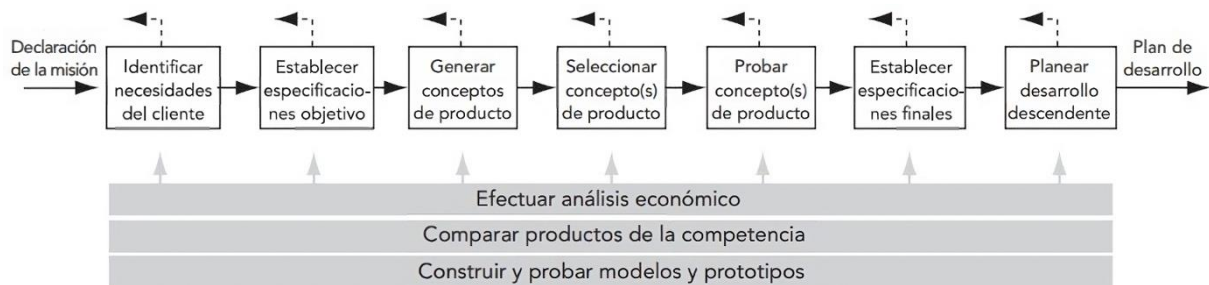


Figura 8. Etapas de la metodología de diseño utilizada. Fuente: [18].

A continuación, se describen cada una de estas etapas y la forma en que fueron adaptadas y aplicadas específicamente al desarrollo del sistema de control de velocidad para el Rover.

3.1 Identificación de necesidades

La primera etapa de la metodología consistió en identificar las necesidades específicas del sistema de control de velocidad a desarrollar. Para ello, se llevaron a cabo dos entrevistas presenciales con el coordinador del SETEC Lab, quien además funge como cliente del proyecto. La primera entrevista tuvo como objetivo comprender el contexto general del proyecto ELANav y el papel que cumple el sistema propuesto dentro de esta iniciativa. La segunda entrevista se centró en identificar los requerimientos funcionales y operativos del sistema, así como las expectativas de desempeño del cliente.

A partir de la información recopilada, se elaboró una lista preliminar de necesidades, la cual fue posteriormente organizada de forma jerárquica para distinguir entre necesidades primarias y secundarias. Con el fin de conocer el nivel de prioridad de cada una, se aplicó una encuesta de valoración al mismo entrevistado, utilizando una escala del 1 al 5. Esta escala permitió clasificar cada necesidad según su importancia relativa dentro del sistema, y así orientar adecuadamente las decisiones de diseño.

Estas necesidades se utilizaron como base para definir las métricas y especificaciones técnicas que guiaron el desarrollo del proyecto, las cuales se presentan en la siguiente sección.

3.2 Establecimiento de especificaciones

Para proceder a las siguientes etapas de desarrollo, antes es necesario definir especificaciones que describen lo que el producto tiene que hacer de manera cuantificable. Esta etapa incluye la selección de métricas relevantes y la asignación de valores objetivo y marginales, con el objetivo de poder medir el grado de cumplimiento de cada necesidad [18].

Un desafío importante en esta etapa fue la limitada disponibilidad de literatura técnica que detalle, con precisión numérica, los criterios de desempeño específicos aplicables al control de velocidad en Rovers de misiones exitosas de la última década. Además, los sistemas utilizados en misiones espaciales tienden a combinar múltiples estrategias avanzadas como odometría visual, fusión de sensores inerciales y algoritmos de visión por computadora, lo que los vuelve difícilmente comparables con la solución simplificada planteada en este trabajo.

Ante esta dificultad, se recurrió a fuentes complementarias, incluyendo documentación sobre robótica educativa de baja velocidad, así como el criterio técnico del cliente del proyecto, quien cuenta con experiencia en la investigación de sistemas espaciales. De este modo, fue posible definir un conjunto de métricas representativas para el sistema a implementar, y asignarles valores objetivo y marginales. Estos valores permiten establecer rangos de desempeño aceptable, asegurando tanto la funcionalidad del sistema como su viabilidad para futuras etapas del proyecto ELANav.

3.3 Generación de conceptos

Con base en las necesidades y especificaciones definidas, se procedió al análisis funcional del sistema mediante un modelo de bloques, lo que permitió identificar las subfunciones clave necesarias para implementar el control de velocidad. A partir de este análisis, se abordó la generación de conceptos utilizando tanto búsqueda externa como interna.

La búsqueda externa incluyó la revisión de literatura relacionada con sistemas de control en Rovers de exploración espacial, así como consultas con expertos del SETEC Lab. Esta información sirvió como referencia para identificar soluciones comunes o recomendadas para cada subfunción. Por su parte, la búsqueda interna se enfocó en proponer ideas viables con base en el conocimiento adquirido durante el desarrollo del proyecto y las restricciones prácticas del prototipo disponible. Para cada subproblema identificado, se consideraron al menos cuatro alternativas de solución, que luego fueron organizadas en un árbol de conceptos para facilitar su comparación.

Este proceso permitió construir una base sólida de alternativas, adaptadas al contexto específico del sistema desarrollado, con miras a seleccionar la solución más adecuada en la siguiente etapa.

3.4 Selección y evaluación de conceptos

La selección del concepto más adecuado para el sistema se realizó mediante un proceso de filtrado cualitativo, tomando como base los conceptos generados en la etapa anterior. Esta evaluación consistió en comparar cada alternativa frente a un concepto de referencia utilizando una matriz de filtrado, donde se calificaron las ventajas (+), similitudes (0) o desventajas (-) de cada opción.

Dado el alcance definido para este trabajo, las restricciones presupuestarias y las características del hardware disponible en el laboratorio, no fue necesario implementar una segunda etapa de evaluación ponderada. El filtrado permitió identificar con claridad un concepto superior al resto, cuya viabilidad técnica y operativa lo posicionó como la mejor opción para ser desarrollado. En consecuencia, se procedió con dicho concepto hacia las etapas de diseño detallado y validación.

3.5 Pruebas y validación

La etapa de validación fue clave para comprobar si el sistema desarrollado cumplía con los requisitos funcionales establecidos. Como parte de esta fase, se construyó un banco de pruebas específico que permitió simular condiciones de operación relevantes y observar el comportamiento del sistema en escenarios controlados.

Estas pruebas fueron diseñadas no solo para comprobar el desempeño del sistema, sino principalmente para verificar si las especificaciones definidas eran alcanzables con la solución propuesta. A partir de los resultados obtenidos, fue posible identificar limitaciones técnicas, validar la viabilidad de ciertos objetivos de desempeño y, cuando fue necesario, realizar ajustes en el diseño y en la estrategia de control. Esto permitió asegurar que el sistema no solo funcionara en condiciones ideales, sino también bajo condiciones similares a las esperadas en su aplicación final.

El diseño detallado de estas pruebas, así como los resultados obtenidos y su análisis, se presentan en el Capítulo 6.

3.6 Establecer especificaciones finales

Con el concepto seleccionado y el diseño del sistema implementado, se procedió a refinar las especificaciones iniciales para obtener una caracterización más precisa de la solución propuesta. Mientras que las primeras especificaciones fueron definidas como rangos amplios, en esta etapa se determinaron los valores reales alcanzados mediante las pruebas de validación.

Esta comparación entre los objetivos planteados al inicio y los resultados finales permitió evaluar el grado de cumplimiento de cada especificación y detectar posibles oportunidades de mejora o ajustes en el diseño. Los resultados obtenidos se presentan en el Capítulo 6, junto con el análisis correspondiente.

Capítulo 4. Especificaciones técnicas y parámetros operativos

Esta sección presenta las condiciones técnicas y operativas que guían el desarrollo del sistema. Se parte de un diagnóstico inicial del problema, seguido por la descripción del Rover utilizado, la identificación de necesidades, la definición de especificaciones técnicas y, finalmente, el concepto de operaciones que resume el funcionamiento esperado del sistema en su entorno de validación.

4.1 Diagnóstico del Rover

Inicialmente, se consideró utilizar un prototipo ya disponible en el laboratorio que se muestra en la Figura 9, sin embargo, tras un diagnóstico se determinó que este presentaba múltiples limitaciones técnicas que ponían en riesgo la viabilidad del proyecto. Entre los principales problemas identificados se encuentran:

- Inestabilidad estructural: El sistema de suspensión presentaba deformaciones (pandeo) e inestabilidad mecánica, afectando la trayectoria y desempeño del Rover durante el movimiento.
- Baja velocidad angular en los motores: Las revoluciones por minuto eran demasiado reducidas, lo que limitaba la capacidad de observar con claridad el efecto del sistema de control de velocidad, comprometiendo la validación.
- Deficiencias en el sistema de alimentación: Las baterías disponibles estaban en mal estado o se descargaban rápidamente, lo que implicaba costos adicionales y afectaba la continuidad de las pruebas. Aun cuando se empleaban baterías nuevas, su autonomía era limitada.
- Corrientes de arranque elevadas: Durante el encendido del sistema, se registraban picos de corriente significativos debido a un torque no regulado, lo cual podría comprometer la seguridad del microcontrolador y otros componentes electrónicos.

- Baja calidad general de los componentes: En términos de electrónica, mecánica y acabados, el prototipo original no cumplía con los estándares requeridos para garantizar resultados reproducibles ni confiables.



Figura 9. Primer prototipo considerado.

Dado el cronograma establecido para el desarrollo de este proyecto y la precisión que exige la implementación y validación del sistema de control propuesto, se decidió en conjunto con la coordinación del laboratorio buscar una alternativa comercial más adecuada. Tras evaluar distintas opciones disponibles en el mercado, se seleccionó un nuevo prototipo comercial de tipo educativo, adquirido a través de la empresa SunFounder, especializada en *kits* de robótica y electrónica.

La elección de este nuevo prototipo se fundamentó en varios criterios: su arquitectura mecánica tipo *Rocker-bogie*, preferida por el cliente; la buena relación entre costo y calidad de sus componentes; y su capacidad de integración con sensores y ejecución de algoritmos de control gracias al microcontrolador incorporado. Esta decisión no solo permitió garantizar el cumplimiento de los objetivos planteados en este proyecto, sino que también proporcionó al laboratorio una plataforma reutilizable para futuras investigaciones y validaciones relacionadas con el desarrollo del Rover.

En la siguiente sección se detallan las características del Rover seleccionado.

4.1.1 Descripción general del Rover y sus subsistemas

El Rover utilizado en este proyecto es el GalaxyRVR Rover de la empresa SunFounder.



Figura 10. SunFounder GalaxyRVR Rover Kit. [19]

Este modelo cuenta con una arquitectura de suspensión tipo *Rocker-Bogie*, como se muestra en la Figura 11, la cual ha sido ampliamente adoptada en vehículos de exploración planetaria debido a su capacidad para mantener la estabilidad en terrenos irregulares.

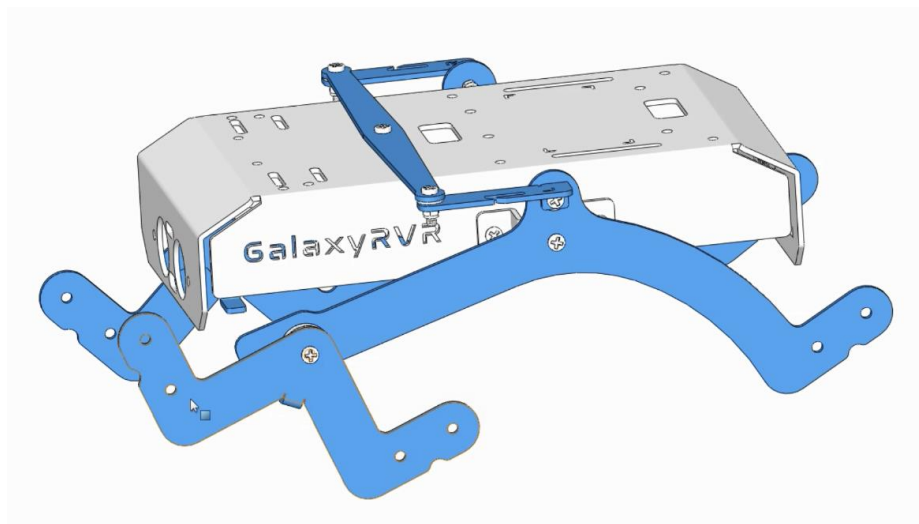


Figura 11. Suspensión Rocker-Bogie del Rover. [19]

La estructura del sistema *Rocker-Bogie* se compone principalmente de dos elementos articulados por lado: el *rocker* y el *bogie*. El *rocker* funciona como un brazo largo en cada lado del chasis del Rover. Ambos *rockers* están conectados entre sí y al cuerpo principal mediante un mecanismo diferencial, lo que permite que giren en direcciones opuestas. Esta característica asegura que la mayoría de las ruedas se mantengan en contacto con el terreno, distribuyendo de forma más uniforme el peso y aumentando la estabilidad del vehículo. El cuerpo del Rover se ajusta dinámicamente a una inclinación media entre los dos *rockers*. En el extremo de cada *rocker* se conecta un *bogie*, que es una estructura más pequeña compuesta por dos ruedas y una articulación central que se une al *rocker*.

El Rover cuenta con un total de 6 ruedas fabricadas en caucho sintético como la que se muestra en la Figura 12. Cada una tiene un diámetro de 75 mm y un ancho de 37 mm, esta banda de rodamiento ancha y el diseño de labrado mejoran la tracción sobre diversas superficies. Estas ruedas están integradas al sistema *rocker-bogie*, facilitan una distribución efectiva del peso y el desplazamiento continuo incluso en superficies irregulares.



Figura 12. Rueda del Rover. [20]

El Rover utiliza motores tipo TT con caja reductora, los cuales integran un motor de corriente continua acoplado a un sistema de engranajes dentro de una carcasa plástica compacta como se observa en la Figura 13. La incorporación de la reducción mecánica permite aumentar el par de salida, esto es esencial para el desplazamiento del Rover porque le permite mover cargas

mayores y superar irregularidades en el terreno. El control de dirección del motor se logra mediante la inversión de la polaridad de la corriente eléctrica, lo que permite cambiar el sentido de giro del eje y, por ende, el sentido de movimiento del vehículo. Hay un motor para cada rueda.



Figura 13. Uno de los seis motores del Rover. [20]

La placa de desarrollo SunFounder R3, compatible con Arduino, es el componente principal encargado de la programación y control del Rover. Su diseño permite una integración sencilla con diversos actuadores y sensores, este se muestra en la Figura 14.

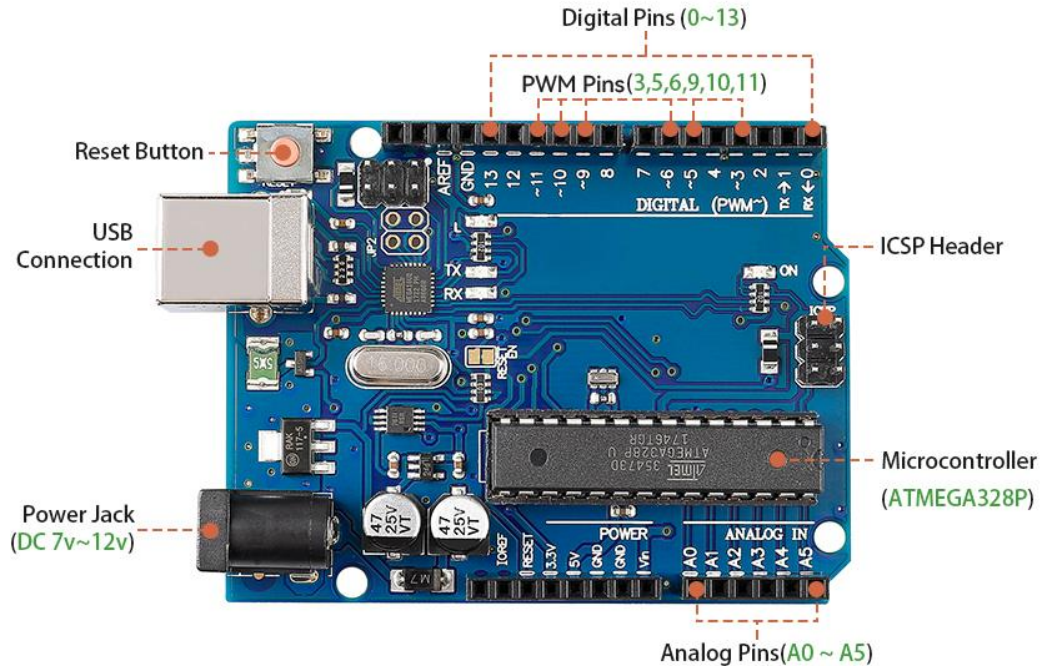


Figura 14. Placa de desarrollo SunFounder R3, basada en Arduino. [20]

Esta placa cuenta con 14 pines digitales configurables como entrada o salida, utilizados para el envío o recepción de señales binarias. Seis de estos pines admiten modulación por ancho de pulso (PWM), útil para el control de velocidad en motores, ajuste de brillo en luces LED u otros dispositivos que requieren regulación analógica. Además, dispone de 6 entradas analógicas, empleadas para la lectura de señales provenientes de sensores que entregan datos en forma de voltaje variable. Estas señales se digitalizan para ser procesadas por el microcontrolador.

La placa se programa a través de una conexión USB, que también permite la comunicación con una computadora durante las pruebas. Para su alimentación eléctrica, incorpora un conector de energía externa, compatible con baterías o adaptadores. También incluye un conector ICSP para reprogramación directa del microcontrolador y un botón de reinicio, que reinicia la ejecución del programa cargado.

Como complemento a la placa SunFounder R3, se utiliza un *shield* (placa de expansión) diseñada específicamente para facilitar la conexión de los distintos componentes del sistema, incluyendo motores, sensores y la fuente de alimentación. Esta placa se puede observar en la Figura 15 y actúa como centro de control y distribución de señales. Esta se monta directamente sobre la placa SunFounder R3.

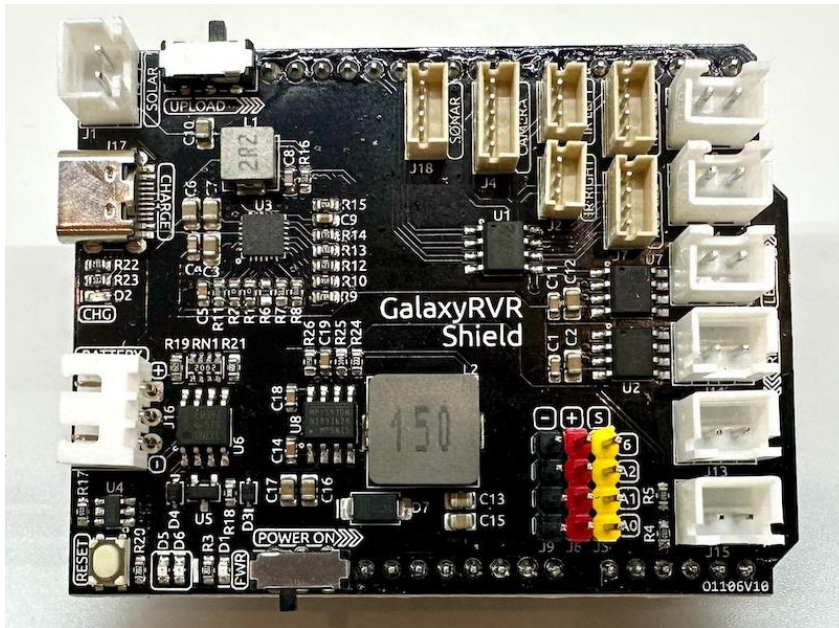


Figura 15. Placa de expansión GalaxyRVR Shield. [20]

Un aspecto clave a considerar es que los pines digitales de una placa Arduino típica solo pueden entregar corrientes del orden de 20 mA, lo cual es insuficiente para accionar directamente un motor de corriente continua. Por esta razón, el sistema incorpora circuitos integrados de potencia (*drivers*) que funcionan como intermediarios entre la placa SunFounder R3 y los motores. Estos *drivers* amplifican la señal de control de baja corriente proveniente del microcontrolador, entregando la corriente necesaria para accionar los motores.

En el caso específico de este Rover, el *shield* cuenta con seis puertos de conexión para motores, distribuidos en dos grupos: tres para el lado izquierdo y tres para el lado derecho del Rover. Cada grupo de motores está controlado por un *driver* independiente, como se muestra en la Figura 16, donde se puede ver la ubicación de estos componentes en la placa.

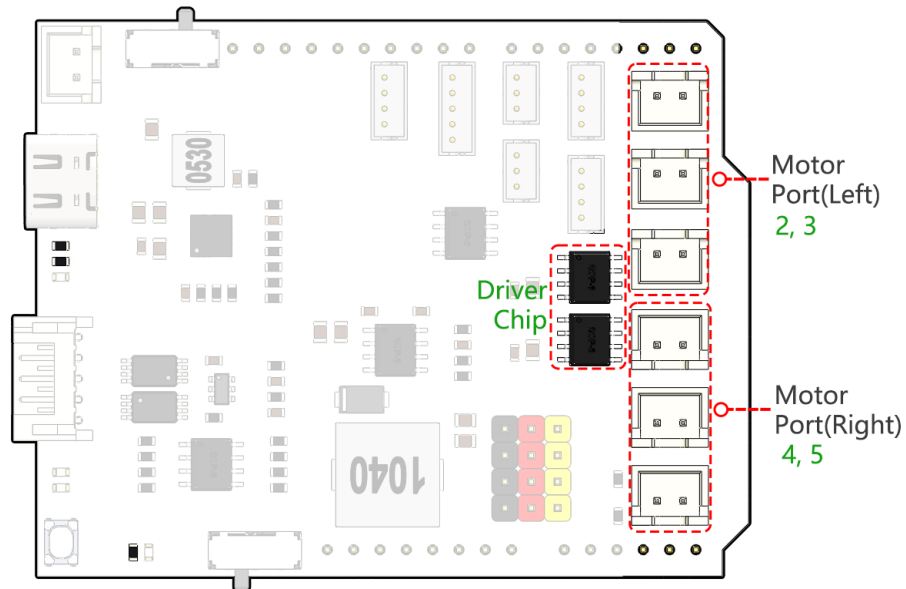


Figura 16. Ubicación de los controladores de motor en el GalaxyRVR Shield. [19]

El funcionamiento se basa en un principio sencillo: los pines 2 y 3 del Arduino están conectados a uno de los *drivers*, el cual a su vez controla los tres motores del lado izquierdo conectados en paralelo. Del mismo modo, los pines 4 y 5 están conectados al segundo *driver*, que controla los tres motores del lado derecho, también conectados en paralelo. Esta arquitectura se ilustra en la Figura 17, donde se representa el flujo de señal desde el microcontrolador hacia los motores a través de los *drivers*.

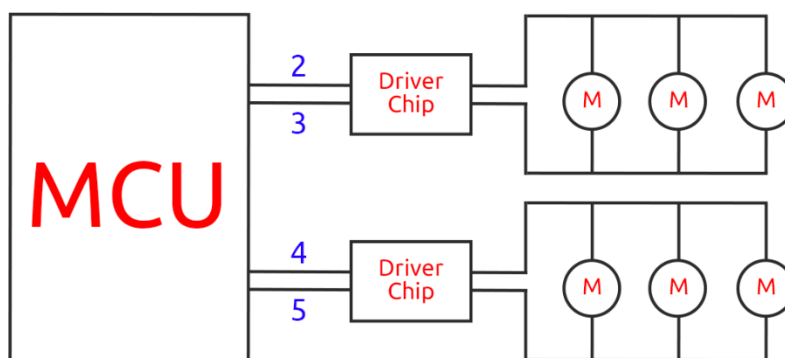


Figura 17. Flujo de señal hacia los motores. [19]

Adicionalmente, el Rover cuenta con un conjunto de sensores y actuadores que complementan sus capacidades de percepción y control. Entre ellos se incluye el módulo ESP32-CAM de la

Figura 18, una cámara compacta con sensor OV2560, conectividad Wi-Fi y ranura para tarjeta microSD que permite almacenar imágenes de forma local. Este módulo, de dimensiones reducidas (27x40.5x4.5 mm) y bajo consumo (6 mA), se conecta al *shield* del Rover mediante una placa adaptadora a través de un conector ZH de 5 pines.

La cámara está montada sobre un servo motor de rotación angular, que permite ajustar con precisión la orientación de la cámara en un rango de hasta 180°. Este servo que se muestra en la Figura 19, con engranajes metálicos y alimentación entre 4.8 V y 7.4 V, se conecta al *shield* mediante un conector de 3 pines, permitiendo la captura de imágenes desde distintos ángulos durante la navegación del Rover.

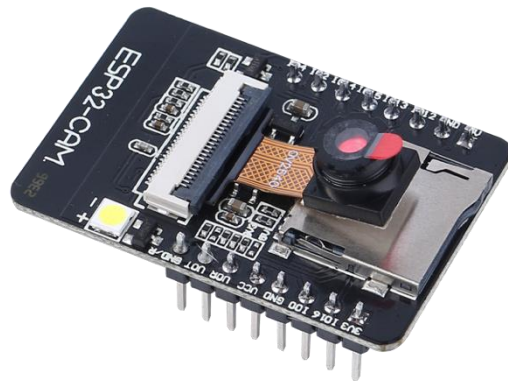


Figura 18. Cámara ESP32. [19]



Figura 19. Servo motor. [19]

El sistema también integra dos módulos de sensores infrarrojos (IR) de evasión de obstáculos, ubicados en el lateral izquierdo y derecho del Rover. Estos sensores pueden detectar objetos a una distancia de entre 2 cm y 40 cm, dependiendo del nivel de reflectividad de la superficie detectada (menor para objetos oscuros). Como se muestra en la Figura 20, cada módulo incluye un emisor y un receptor infrarrojo; cuando la señal emitida se refleja en un objeto y es captada por el receptor, se genera una salida digital de nivel bajo que puede ser procesada por el sistema.

Además, el Rover incorpora un sensor ultrasónico HC-SR04 de uso común, con un rango de medición de 2 cm a 400 cm y una precisión de hasta ± 3 mm. Este sensor permite estimar distancias con mayor exactitud que los módulos IR y se conecta al *shield* mediante un conector de 5 pines.

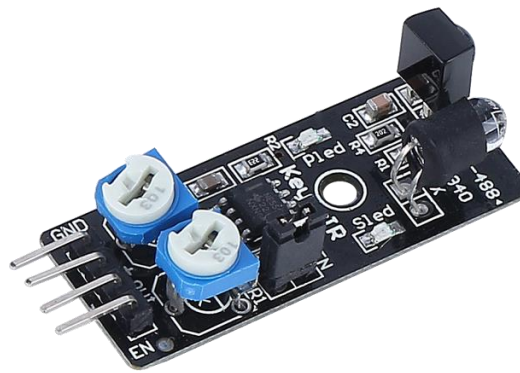


Figura 20. Sensor infrarrojo. [19]



Figura 21. Sensor ultrasónico. [19]

Finalmente, con respecto al sistema de energía del Rover, este se compone de un paquete de baterías recargables y un panel solar, los cuales trabajan de forma complementaria para garantizar la operación continua del vehículo tanto durante el día como en la noche.

El paquete de baterías, desarrollado por SunFounder, integra dos celdas 18650 de 3.7 V y 2000 mAh cada una, alcanzando una capacidad combinada que permite una autonomía de aproximadamente 90 minutos de operación continua. El sistema admite una corriente de carga de 5 V a 2 A, con un tiempo estimado de carga de 130 minutos, y una salida de energía de hasta 5 V a 5 A, suficiente para alimentar todos los subsistemas del Rover. La batería puede recargarse mediante una fuente de alimentación externa conectada al *shield*, o a través del panel solar, dicha conexión eléctrica se realiza mediante un conector XH2.54 de 3 pines.

El panel solar proporciona una fuente de energía renovable con una salida nominal de 6 V y 660 mA, lo que teóricamente permite recargar completamente el paquete de baterías en aproximadamente 7.2 horas bajo condiciones óptimas de iluminación solar. Este módulo cuenta con dimensiones de 170 × 170 mm, utiliza cableado de tipo 24AWG (200 mm de longitud), y se conecta mediante un conector XH2.54 de 2 pines.

La lista completa de partes se encuentra en el Anexo A.

4.2 Identificación de necesidades

A partir del diagnóstico realizado y su análisis, se plantearon una serie de entrevistas con el coordinador del laboratorio. Las necesidades identificadas fueron las siguientes:

- 1) Cada bloque funcional del sistema diseñado (SD) es mapeable a agentes.
- 2) El SD ejecuta el control de velocidad de manera autónoma.
- 3) El SD es estable.
- 4) El SD responde con rapidez ante cambios en las condiciones del terreno.
- 5) El SD evita variaciones abruptas o excesos al alcanzar la velocidad deseada.
- 6) El SD funciona en varios terrenos distintos.

Estas se organizaron en necesidades primarias y secundarias, donde se definen cuáles son las necesidades principales y el conjunto de secundarias que componen cada una. El resultado se muestra en la Tabla 1.

Tabla 1. Jerarquía de necesidades.

Necesidades primarias	Necesidades secundarias
El SD puede ser mapeado a una arquitectura multiagente.	Cada bloque funcional del SD es mapeable a agentes.
El SD debe ejecutar el control de velocidad de manera autónoma.	El SD es estable.
	El SD responde con rapidez ante cambios en las condiciones del terreno.
	El SD evita variaciones abruptas o excesos al alcanzar la velocidad deseada.
El SD funciona en varios terrenos distintos.	

Como se mencionó en el Capítulo 3, se consultó con el cliente la importancia de cada necesidad identificada en una escala de 1 a 5, como se muestra en la Tabla 2.

Tabla 2. Importancia de necesidades.

	Necesidad	Importancia
1	El SD puede ser mapeado a una arquitectura multiagente.	5
2	El SD ejecuta el control de velocidad de manera autónoma.	5
3	El SD es estable.	4
4	El SD responde con rapidez ante cambios en las condiciones del terreno.	3
5	El SD evita variaciones abruptas o excesos al alcanzar la velocidad deseada.	3
6	El SD funciona en varios terrenos distintos.	4

Se puede observar que ninguna de las necesidades establecidas esta clasificada como no importante o indeseables, lo que indica que estas son relevantes para el diseño y deseadas por el cliente.

4.3 Establecimiento de especificaciones

La definición de especificaciones es un paso clave en el proceso de diseño, ya que establece los requisitos técnicos que debe cumplir el sistema para satisfacer las necesidades del proyecto. Estas especificaciones se derivan del análisis de los objetivos funcionales, las condiciones operativas previstas, las limitaciones del entorno y los recursos disponibles. En esta sección se presentan las especificaciones que guiarán el desarrollo del sistema. En la Figura 20 se muestra la matriz de necesidades y métricas obtenida.

		Métrica					
		1	2	3	4	5	6
		Los bloques funcionales del sistema pueden ser mapeados a agentes	Error de medición de velocidad	Tiempo de estabilización	Tiempo de respuesta	Sobreimpulso	Desempeño en al menos dos superficies distintas
Necesidad							
1	El SD puede ser mapeado a una arquitectura multiagente.	•					
2	El SD ejecuta el control de velocidad de manera autónoma.		•				
3	El SD es estable.			•			
4	El SD responde con rapidez ante cambios en las condiciones del terreno.				•		
5	El SD evita variaciones abruptas o excesos al alcanzar la velocidad deseada.					•	
6	El SD funciona en varios terrenos distintos.						•

Figura 22. Matriz de necesidades y métricas.

Se puede observar que cada necesidad cuenta con una métrica y cada métrica responde a una necesidad. Esta correspondencia asegura que las especificaciones técnicas del sistema están alineadas con los requerimientos funcionales del proyecto, permitiendo una evaluación objetiva de su cumplimiento durante las etapas de desarrollo y validación.

4.3.1 Selección de valores objetivo y marginales para la métrica de mapeo del sistema a agentes

La primera métrica establecida corresponde a “Los bloques funcionales del sistema pueden ser mapeados a agentes”. Esta métrica evalúa si el sistema propuesto ha sido diseñado de manera que sus componentes funcionales puedan, en una etapa posterior del proyecto, ser mapeados a una arquitectura multiagente. Dado que esta verificación no implica implementar directamente un sistema multiagente, sino únicamente asegurar su viabilidad estructural, la métrica se define como binaria, lo que significa que únicamente puede tomar dos valores posibles:

- Pasa: Todos los bloques funcionales del sistema pueden ser mapeados de forma clara y lógica a agentes individuales dentro de una arquitectura multiagente.
- No pasa: Uno o más bloques funcionales del sistema no pueden ser mapeados adecuadamente a agentes, ya sea por limitaciones estructurales, conceptuales o por falta de independencia funcional.

El carácter binario de esta métrica se justifica por la naturaleza cualitativa de la necesidad. El sistema, desde el punto de vista de diseño, o es compatible con una arquitectura multiagente, o no lo es. La evaluación de la mapeabilidad a agentes no admite escalas intermedias, ya que el objetivo no es medir qué tan avanzado es el mapeo ni qué tan óptima es la futura arquitectura multiagente, sino si el sistema permite o no permite ser reestructurado en términos de agentes.

Considerando la importancia asignada por el cliente a esta necesidad, se establece que tanto el valor objetivo como el valor marginal para esta métrica es ‘Pasa’. Es decir, no se contempla la aceptación de un diseño que no cumpla con esta capacidad.

4.3.2 Selección de valores objetivo y marginales para la métrica sobre el error de medición de velocidad

Esta métrica evalúa la precisión del sistema de control en alcanzar y mantener la velocidad deseada del Rover, considerando la diferencia entre la referencia de velocidad establecida y la velocidad real medida durante el desplazamiento. Se expresa como un porcentaje de error respecto al valor de referencia.

En aplicaciones de exploración planetaria los rovers operan a velocidades bajas, usualmente en el orden de centímetros por segundo. Algunos ejemplos representativos son:

- Perseverance: aproximadamente 4.47 cm/s [14].
- Curiosity: aproximadamente 4 cm/s [21].
- Spirit: aproximadamente 3.75 cm/s [22].

Este orden de magnitud en la velocidad implica que los errores de velocidad se traducen en desviaciones de posicionamiento relativamente pequeñas, lo cual resulta aceptable en trayectos cortos, segmentados y planificados como los que ejecutan este tipo de vehículos en escenarios reales [23].

Dado este contexto, y considerando que el presente proyecto se enmarca en el desarrollo de un prototipo inicial para el SETEC LAB y el estudio de caso del proyecto ELANAV [2], no se exige una precisión estricta en esta métrica. Se establece como valor objetivo un error igual o inferior al 10%, lo cual garantiza una velocidad suficientemente próxima a la deseada para propósitos de validación. Asimismo, se considera aceptable un valor marginal de hasta el 15%, que permite tolerar las variaciones asociadas al uso de sensores simples y las diferencias físicas del entorno sin comprometer el funcionamiento general del sistema.

Esta flexibilidad es coherente con la etapa actual de desarrollo del sistema, en la cual se prioriza la estabilidad y el diseño modular sobre la precisión exacta en el seguimiento de velocidad.

4.3.3 Selección de valores objetivo y marginales para la métrica del tiempo de estabilización

El tiempo de estabilización se define como el tiempo que tarda la salida del sistema en alcanzar y mantenerse dentro de un margen de $\pm 5\%$ respecto al valor de referencia, sin volver a salir de ese rango posteriormente. Esta métrica es fundamental para evaluar el desempeño dinámico de sistemas de control, ya que refleja la rapidez con la que el sistema puede alcanzar un estado estable luego de un cambio en la referencia o ante una perturbación.

En el contexto del presente proyecto, el tiempo de estabilización se interpreta como el intervalo requerido para que la velocidad del motor converja a la velocidad deseada después de un cambio en la señal de entrada.

Dado que el sistema está basado en un control implementado en un microcontrolador de recursos limitados (Arduino UNO) y que opera a bajas velocidades, se define como valor objetivo un tiempo de estabilización entre 1 y 2 segundos, intervalo considerado adecuado para aplicaciones móviles de baja velocidad y carga reducida. [24]

Como valor marginal, se establece un tiempo de estabilización de hasta 4 segundos. Este umbral se considera aún aceptable en la fase de desarrollo del prototipo, ya que permite tolerar efectos de fricción variable, pequeñas imprecisiones en la medición de velocidad y limitaciones en la sintonización del controlador, sin comprometer la funcionalidad del sistema. No obstante, valores superiores indicarían una respuesta lenta o deficiente que podría requerir ajustes adicionales en el diseño del control.

4.3.4 Selección de valores objetivo y marginales para la métrica del tiempo de respuesta

El tiempo de respuesta se define como el intervalo que transcurre desde que se aplica un cambio en la entrada del sistema hasta que la salida alcanza por primera vez un valor cercano a la referencia. En sistemas de control clásicos, se calcula generalmente como el tiempo requerido para que la señal de salida pase del 10 % al 90 % del valor final, aunque esta definición puede variar levemente según el tipo de sistema o aplicación [25].

En este caso, el tiempo de respuesta representa qué tan rápido reacciona el sistema ante una nueva consigna de velocidad. Esta característica es importante para garantizar que el Rover pueda ajustarse con agilidad a cambios de referencia o condiciones del terreno, especialmente en maniobras como arranques, giros o cambios de dirección.

Dado que el sistema está destinado a operar a velocidades bajas y no requiere una reacción inmediata como en sistemas de control crítico o de alta velocidad, se establece como valor objetivo un tiempo de respuesta menor o igual a 1.5 segundos.

Como valor marginal, se acepta un tiempo de respuesta de hasta 3 segundos, lo cual sigue siendo razonable para aplicaciones de baja velocidad y prototipado, donde se busca un equilibrio entre rapidez y estabilidad. Además, este rango tolera las limitaciones asociadas al uso de sensores simples y al procesamiento del microcontrolador.

4.3.5 Selección de valores objetivo y marginales para la métrica de sobreimpulso

El sobreimpulso se define como el valor máximo que alcanza la salida de un sistema por encima del valor final deseado, expresado como un porcentaje de dicho valor de referencia [26]. Esta métrica es particularmente importante cuando se requiere evitar respuestas bruscas, ya que un sobreimpulso elevado puede comprometer tanto la precisión como la seguridad de operación.

En este caso particular, el sistema está diseñado para operar a velocidades significativamente inferiores a la capacidad máxima del hardware, y las superficies de prueba previstas presentan

únicamente obstáculos de tamaño moderado. Esto significa que incluso si se produce una subida de velocidad algo abrupta, no se espera que tenga efectos mecánicos o estructurales relevantes sobre el sistema, ya que ni la energía involucrada ni la aceleración resultante serían lo suficientemente elevadas como para comprometer su integridad o funcionalidad. Por tanto, se admite cierto margen de sobreimpulso sin que ello represente un riesgo para el prototipo.

Dado este contexto, se establece como valor objetivo un sobreimpulso menor al 10%, lo cual es coherente con buenas prácticas en el diseño de sistemas de control, y asegura una transición estable sin respuestas excesivas. Como valor marginal, se acepta un sobreimpulso de hasta 15 %, tolerable en el prototipo desarrollado, especialmente ya que no supone un peligro sobre el comportamiento físico del Rover. Superar este umbral implicaría la necesidad de ajustar el controlador o de revisar los parámetros del sistema.

4.3.6 Selección de valores objetivo y marginales para la métrica sobre el desempeño en al menos dos superficies distintas

Esta métrica tiene como objetivo evaluar si el sistema de control es capaz de mantener un comportamiento aceptable en al menos dos tipos de superficie con características físicas diferentes. Estas superficies serán definidas durante el proceso de pruebas, pero deben representar condiciones reales o relevantes para la operación del vehículo.

Con este objetivo, la métrica se definió más específicamente como la variación porcentual promedio en indicadores de desempeño del sistema entre dos superficies distintas. La evaluación se realiza utilizando tres de los indicadores de desempeño mencionados anteriormente (tiempo de estabilización, tiempo de respuesta y porcentaje de sobreimpulso). Estos tres indicadores se obtienen experimentalmente al aplicar la misma señal de referencia en dos superficies distintas. A partir de estos datos, se calcula la variación porcentual relativa de cada indicador mediante la siguiente fórmula:

$$\Delta X_i = \left| \frac{X_{i,superficie2} - X_{i,superficie1}}{X_{i,superficie1}} \right| \cdot 100. \quad (4.1)$$

Donde X_i representa cada uno de los indicadores mencionados. Posteriormente, se calcula el promedio aritmético de las tres variaciones obtenidas. El resultado es un único valor que representa la variación promedio en el desempeño del sistema al cambiar de superficie. Este enfoque permite mitigar la influencia de desviaciones puntuales en uno de los indicadores, ofreciendo una evaluación más representativa del comportamiento general del sistema.

Considerando la naturaleza de esta métrica y el propósito del proyecto, se establece que el valor objetivo corresponde a una variación promedio igual o inferior al 20%. Este límite se considera aceptable, ya que puede reflejar una degradación moderada causada por factores físicos inevitables, como cambios en la tracción, hundimiento o deslizamiento [27]. Estas variaciones son inherentes a la interacción entre las ruedas del Rover y el terreno, cuyo modelado o predicción precisa representa un problema de alta complejidad, especialmente en superficies no rígidas como lo es la mayor parte del terreno lunar [28]. Debido a esta complejidad, se considera que dicho modelado queda fuera del alcance del presente trabajo. En consecuencia, se acepta que ciertas variaciones en el desempeño pueden surgir incluso con un controlador bien ajustado. Por ello, se define el valor marginal como una variación promedio igual o inferior al 40%.

Un resumen de las especificaciones y sus correspondientes valores se muestra en la Tabla 3.

Tabla 3. Especificaciones objetivo y marginales.

Número de métrica	Número de necesidad	Métrica	Imp	Unidades	Valor marginal	Valor ideal
1	1	Los bloques del sistema pueden ser mapeados a agentes	5	Binaria	Pasa	Pasa
2	2	Error de medición de velocidad	5	%	≤ 15	≤ 10
3	3	Tiempo de estabilización	4	s	≤ 4	≤ 2
4	4	Tiempo de respuesta	3	s	≤ 3	≤ 1.5
5	5	Sobreimpulso	3	%	≤ 15	≤ 10
6	6	Variación porcentual promedio en indicadores de desempeño	4	%	≤ 40	≤ 15

Cabe destacar que la definición de los valores objetivo y marginales para las métricas del sistema representó un desafío significativo, debido a la escasa disponibilidad de literatura técnica que detalle parámetros de desempeño específicos en sistemas de exploración planetaria. Aunque existe abundante información sobre los logros científicos y operativos de misiones

espaciales, las agencias responsables suelen limitar la divulgación de especificaciones detalladas a nivel de subsistemas. Además, los sistemas actuales representan soluciones de alta complejidad que integran múltiples funciones simultáneamente, como el control de orientación, velocidad, aceleración, percepción mediante visión por computadora y estrategias para mitigar el desgaste mecánico, lo cual dificulta establecer comparaciones directas con un sistema simplificado como el desarrollado en este proyecto.

Por esta razón, se optó por tomar como referencia valores de desempeño propios de plataformas de robótica móvil educativa de baja velocidad, ajustando los márgenes según la naturaleza del entorno y los objetivos del proyecto. Algunas métricas presentan rangos amplios, debido a la alta sensibilidad de su cálculo ante pequeñas variaciones, como ocurre en el caso de la medición de diferencias de desempeño entre superficies. Estos valores fueron definidos con el fin de establecer una base realista para el diseño, implementación y evaluación del sistema.

4.4 Concepto de operaciones (ConOps)

En el contexto de sistemas espaciales, el Concepto de Operaciones (CONOPS) es un documento técnico que describe de manera estructurada cómo se espera que funcione un sistema dentro de su entorno operativo, considerando tanto sus componentes internos como las interacciones externas. Su propósito es establecer una visión clara del uso previsto del sistema, especificando sus funcionalidades principales, los escenarios de operación, las condiciones asumidas, las interfaces entre subsistemas y los criterios que definen el éxito de la misión o prueba. En el caso de sistemas como rovers o plataformas de prueba en laboratorio, el CONOPS permite al equipo de desarrollo anticipar las condiciones reales de uso, definir procedimientos de validación y asegurar que todos los elementos del sistema, incluyendo hardware, software y entorno, estén alineados con los objetivos del proyecto. Además, sirve como punto de referencia para todas las fases del desarrollo, desde el diseño hasta la integración y validación final.

El Concepto de Operaciones completo que describe el funcionamiento del sistema de control de velocidad, sus componentes, interfaces, escenarios de prueba y criterios de éxito, se presenta en el Apéndice A de este documento.

Capítulo 5. Diseño del sistema de control de velocidad

Este capítulo presenta el diseño del sistema de control de velocidad implementado en el Rover. Se abordan de manera secuencial las etapas necesarias para su diseño, partiendo de la descomposición funcional del sistema y la generación de conceptos para los módulos de control y realimentación, hasta su modelado, ajuste e implementación. La sección incluye además el análisis cinemático del sistema, el planteamiento del modelo dinámico y la implementación en el prototipo físico.

5.1 Descomposición en subfunciones

La descomposición funcional constituye el primer paso en el proceso de desarrollo de soluciones técnicas, ya que permite dividir el sistema general en funciones parciales que orientan el diseño de los distintos subsistemas. En esta etapa se parte de un enfoque de caja negra, en el cual se representa el sistema como una unidad que recibe entradas y entrega salidas, sin especificar su estructura interna.

La Figura 23 presenta el diagrama de caja negra del sistema de control de velocidad, donde se identifican las entradas principales (energía de alimentación y señal de referencia) y la salida deseada (movimiento controlado del Rover).

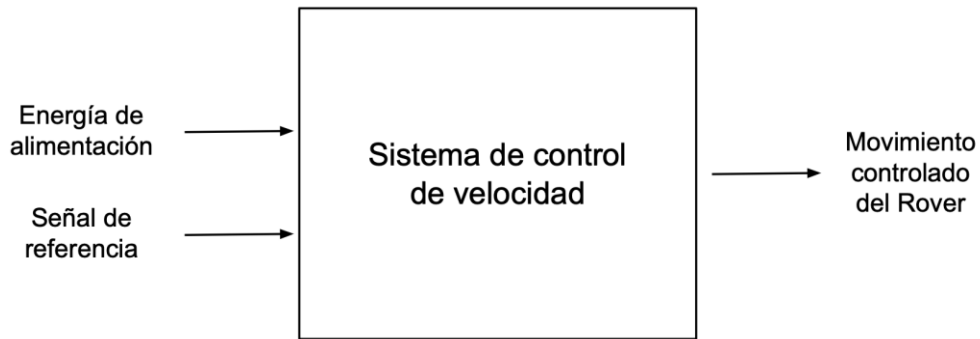


Figura 23. Diagrama de caja negra del sistema.

Posteriormente, este esquema es descompuesto en subfunciones que permiten identificar los procesos internos necesarios para cumplir con los requerimientos del sistema. En la Figura 24 se muestra el diagrama funcional del sistema, donde se evidencian funciones clave como el sensado de velocidad, el procesamiento de la señal de control, la generación de la señal PWM y la ejecución del movimiento a través de los actuadores.

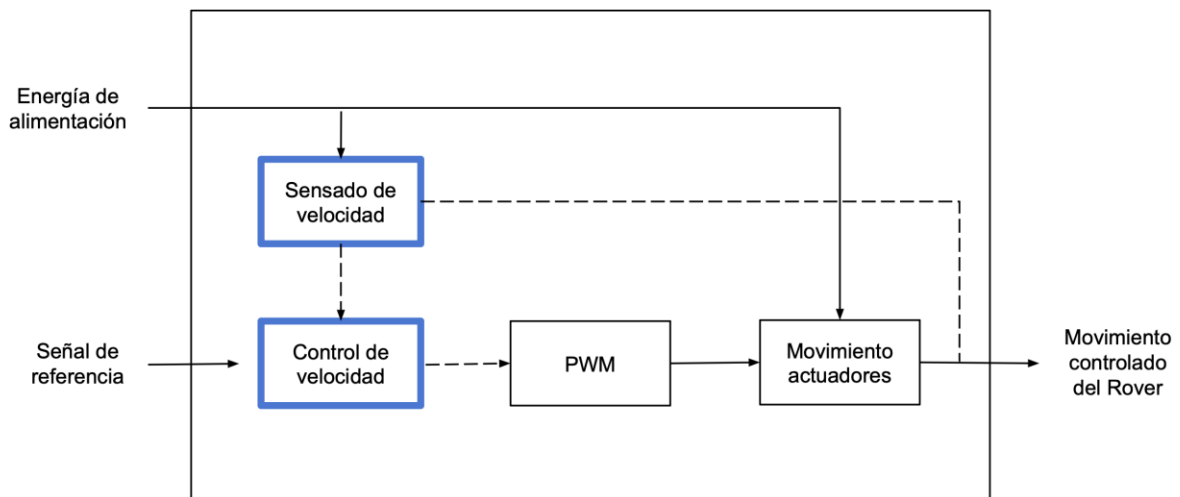


Figura 24. Diagrama de descomposición funcional.

Esta descomposición proporciona la base para la generación y evaluación de conceptos en las siguientes secciones.

5.2 Generación y selección de concepto para sistema de realimentación de velocidad

En este trabajo, es indispensable conocer la velocidad del Rover en tiempo real para corregir desviaciones respecto a una referencia establecida. Actualmente, el Rover no cuenta con el sistema de realimentación de velocidad necesario para un lazo de control cerrado. Para solventar esta necesidad, en esta sección se estudiarán cuatro potenciales soluciones. En la Figura 25 se presenta el árbol de clasificación de conceptos para este subproblema.

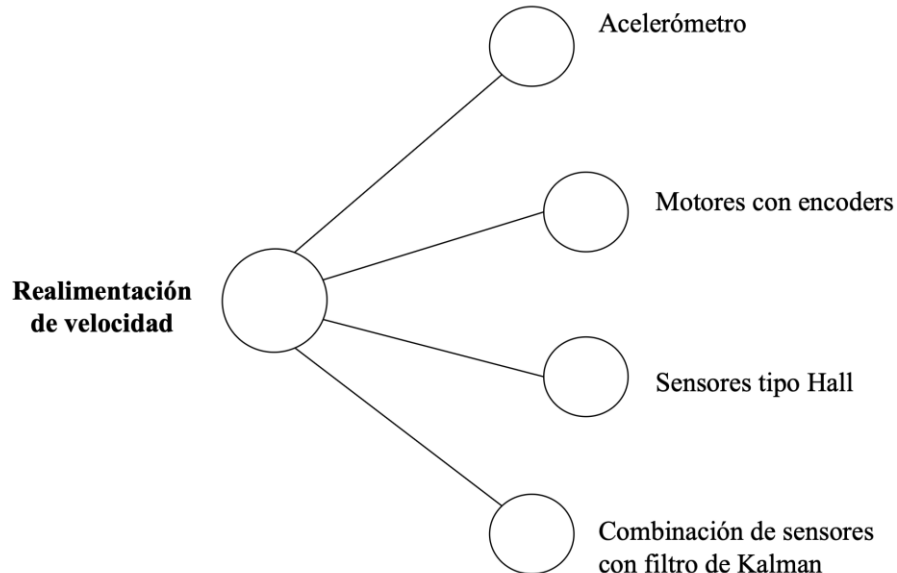


Figura 25. Árbol de clasificación de conceptos para el sentido de velocidad.

5.2.1 Acelerómetro

Uno de los primeros conceptos considerados fue el uso de un acelerómetro para estimar la velocidad del chasis del Rover. Este enfoque consiste en integrar los datos de aceleración en el tiempo para obtener una estimación de la velocidad lineal. Su principal ventaja radica en la simplicidad de implementación, ya que se requiere únicamente un sensor y su procesamiento

asociado. Sin embargo, la desventaja principal de este método es significativa y radica en la acumulación de errores de integración, conocida como *drift*. Este fenómeno produce una desviación progresiva en la estimación de la velocidad, incluso en ausencia de aceleración real, lo que podría comprometer la precisión del sistema de control.

5.2.2 Combinación de sensores con filtro de Kalman

El segundo concepto estudiado fue elegido debido a que, en aplicaciones de navegación espacial, es común emplear la combinación de múltiples sensores para mejorar la estimación de variables de estado. Esta estrategia, conocida como fusión de sensores, permite reducir el error asociado a mediciones individuales mediante el uso de filtros estadísticos, como el filtro de Kalman. Al integrar información proveniente de diversas fuentes, como acelerómetros, giroscopios y *encoders*, se obtiene una estimación más precisa y robusta de la velocidad. No obstante, la principal desventaja de este enfoque radica en la complejidad de su implementación. El diseño e implementación de un filtro de Kalman requiere un conocimiento profundo del modelo dinámico del sistema, así como de las características estadísticas de los sensores involucrados, lo cual puede extenderse más allá del alcance de proyectos de corto plazo o con recursos limitados, como el presente.

5.2.3 Motores con encoders

Una tercera alternativa evaluada fue la sustitución de los motores actuales por unidades que incorporen *encoders*. Estos sensores generan señales digitales fáciles de procesar y ofrecen una precisión adecuada para sistemas de realimentación de velocidad, lo que los convierte en una opción confiable para la implementación de un lazo de control cerrado. Sin embargo, esta solución presenta limitaciones importantes a nivel de hardware. El Rover adquirido incluye un microcontrolador Arduino y una placa de expansión con un número limitado de entradas y salidas, restringidas a las conexiones preestablecidas del sistema original. Mientras que los motores actuales solo requieren dos pines (alimentación y tierra), los motores con *encoder* demandan al menos cuatro pines adicionales: alimentación del *encoder* y dos salidas de señal. Además, sería necesario sustituir los seis motores del Rover, ya que, como se explicó en el capítulo anterior, los motores del lado derecho son controlados por una señal común, al igual que los del lado izquierdo. No se encontraron motores con *encoders* que coincidieran con las

especificaciones de alimentación y relación de engranajes de los motores originales, por lo que reemplazar solo un motor por lado implicaría introducir diferencias en el consumo eléctrico y el comportamiento dinámico. Esto podría generar inconsistencias o interferencias en el funcionamiento del sistema de movilidad del Rover.

5.2.4 Sensores tipo Hall

La cuarta alternativa propuesta consiste en la adición de sensores externos tipo Hall para estimar la velocidad del Rover. Esta solución plantea colocar un imán en las ruedas de cada motor y ubicar un sensor Hall fijo sobre la estructura, de modo que cada vez que el imán pase frente al sensor, se detecte una revolución completa de la rueda. Al medir el tiempo entre eventos sucesivos, es posible calcular la velocidad angular de la rueda, y a partir de esta, la velocidad lineal del Rover. Esta estrategia presenta varias ventajas: no requiere sustituir los motores existentes, su implementación es relativamente sencilla, y el procesamiento de las señales digitales provenientes del sensor Hall puede realizarse fácilmente con el microcontrolador disponible. Además, permite aprovechar los pines aún disponibles del sistema sin superar la capacidad de la placa de expansión. La desventaja de esta opción radica en las variables que pueden afectar la precisión de la información obtenida, tales como la sensibilidad del sensor, la fuerza del campo magnético de los imanes, la alineación del montaje, entre otros factores. Estas condiciones deben ser cuidadosamente consideradas para garantizar mediciones confiables durante la operación del sistema. Durante la etapa de análisis también se evaluó la posibilidad de emplear sensores ópticos como alternativa, sin embargo, la naturaleza móvil del Rover introduce un nivel de complejidad adicional, ya que los sensores ópticos requieren una alineación precisa entre el emisor y el receptor. Esta condición es difícil de mantener en un entorno dinámico, lo que puede comprometer la confiabilidad del sensado. En contraste, los sensores magnéticos ofrecen una mayor tolerancia al desalineamiento, lo que los hace más adecuados para este tipo de aplicaciones móviles.

5.2.5 Selección de concepto de realimentación de velocidad

Siguiendo la metodología establecida, se realizó una comparación entre las cuatro alternativas propuestas para la implementación del sistema de realimentación de velocidad. Para ello, se definieron cuatro criterios de selección: precisión, complejidad, facilidad de implementación y costo.

- **Precisión** evalúa qué tan confiable y precisa es la información de velocidad que proporciona cada solución. Este criterio es el más importante y se relaciona con la necesidad 2, que establece que el sistema debe ejecutar el control de velocidad de manera autónoma, lo cual requiere que la señal de realimentación represente adecuadamente la velocidad real del Rover. Además, una medición precisa es esencial para poder medir y satisfacer las métricas de las necesidades 3, 4 y 5, que exigen estabilidad, rapidez de respuesta ante cambios y la evitación de variaciones abruptas en el control de velocidad.
- **Complejidad** considera la dificultad técnica asociada a la adquisición y procesamiento de los datos. Si bien no está directamente vinculada a una necesidad específica del cliente, su inclusión responde al principio de diseño eficiente: en ingeniería se busca alcanzar los objetivos funcionales con el menor nivel de complejidad posible, reduciendo riesgos de fallo o errores de implementación. Sin este criterio, la selección podría favorecer soluciones innecesariamente sofisticadas que no aporten mejoras proporcionales al esfuerzo requerido.
- **Facilidad de implementación** contempla especialmente las restricciones impuestas por el hardware del Rover, como la disponibilidad de pines y la compatibilidad con componentes ya existentes. Este criterio está alineado con las condiciones prácticas del proyecto, que exige trabajar con el Rover actualmente disponible y con los recursos del laboratorio. No es viable sustituir sistemas completos del Rover por motivos de presupuesto, tiempos de adquisición de componentes y la etapa temprana de desarrollo del prototipo.

- **Costo** analiza el impacto económico de cada opción en términos de componentes adicionales o modificaciones requeridas. Dado que este trabajo se desarrolla como parte del proyecto de investigación ELANAV, financiado con un presupuesto específico, es necesario considerar este criterio para evitar impactos negativos sobre la planificación y ejecución financiera del mismo.

A cada alternativa se le asignaron valores cualitativos positivos (+), neutros (0) o negativos (-) según su desempeño frente a cada criterio. La evaluación neta se calculó como la diferencia entre la cantidad de aspectos positivos y negativos. La Tabla 4 muestra el resultado del análisis comparativo.

Tabla 4. Matriz de selección de conceptos de realimentación de velocidad.

Criterios de Selección	Conceptos			
	A	B	C	D
	Estimación de velocidad mediante acelerómetro	Fusión de sensores mediante filtro de Kalman	Sustitución de motores por unidades con encoders	Incorporación de sensores Hall externos
Precisión	-	+	+	0
Complejidad	0	-	+	+
Facilidad de implementación	+	-	-	+
Costo	+	0	-	+
Suma +	2	1	2	3
Suma 0	1	1	0	1
Suma -	1	2	2	0
Evaluación neta	1	-1	0	3
Lugar	2	4	3	1

Del análisis comparativo se observa que la incorporación de sensores Hall externos (concepto D) obtuvo la evaluación neta más alta, con un total de +3, posicionándose como la solución más equilibrada entre los criterios establecidos. Destaca particularmente por su baja complejidad, facilidad de implementación y bajo costo, aunque su precisión fue evaluada como neutra, esta se considera aceptable dentro del contexto del proyecto.

La segunda mejor opción fue la estimación mediante acelerómetro (concepto A), con una evaluación neta de +1. Si bien presenta una implementación sencilla y bajo costo, su baja precisión, debida al *drift*, la hace menos recomendable para aplicaciones que requieren control en lazo cerrado.

Las opciones de sustitución de motores por unidades con *encoders* (concepto C) y fusión de sensores con filtro de Kalman (concepto B) obtuvieron evaluaciones netas de 0 y -1 respectivamente, ubicándose en los últimos lugares. En ambos casos, si bien ofrecen una alta precisión, se ven penalizadas por su alta complejidad y dificultad de implementación, además de implicar mayores costos.

En conjunto, los resultados justifican la selección del concepto D como la solución más viable para el sistema de realimentación de velocidad del Rover.

5.3 Generación y selección de concepto para controlador

El diseño del controlador constituye el núcleo funcional del presente proyecto, ya que es el componente encargado de regular la velocidad angular del sistema de acuerdo con los requerimientos definidos en las especificaciones. A través del controlador se materializa la capacidad del sistema para responder ante cambios en la referencia, adaptarse a variaciones del entorno y cumplir con los criterios de desempeño establecidos. Por tanto, esta sección aborda la generación y selección del concepto de control más adecuado para el sistema, considerando la naturaleza de la planta, las restricciones de implementación y la necesidad de un comportamiento dinámico predecible y estable. En la Figura 26, se presenta el árbol de clasificación de conceptos para este subproblema.

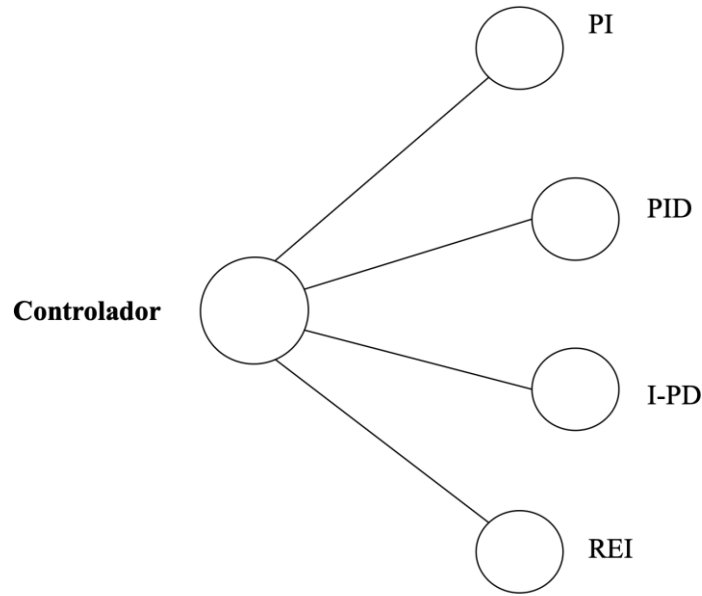


Figura 26. Árbol de clasificación de conceptos para el controlador.

5.3.1 PI

El controlador proporcional-integral (PI) es una estrategia de control ampliamente utilizada en sistemas dinámicos, especialmente cuando se requiere un buen equilibrio entre desempeño y simplicidad de implementación. Este tipo de controlador actúa sobre el error entre la señal de referencia y la salida del sistema, combinando dos componentes: el término proporcional y el término integral. El término proporcional (P) responde de forma instantánea al error actual, generando una acción correctiva directamente proporcional a su magnitud. Esto permite al sistema reaccionar rápidamente ante desviaciones. Por su parte, el término integral (I) acumula el error a lo largo del tiempo, y su contribución crece mientras exista una diferencia sostenida entre la salida y la referencia. Gracias a esta acción acumulativa, el PI puede eliminar el error en estado estacionario que no puede ser corregido únicamente con la acción proporcional [29]. En la Figura 27 se muestra el diagrama de bloques de este controlador.

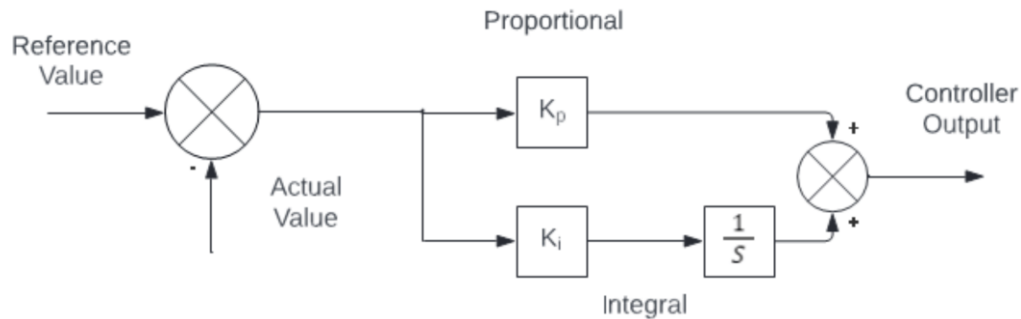


Figura 27. Diagrama de bloques de un controlador proporcional-integral. Fuente: [30]

Esta configuración es particularmente útil en sistemas donde se busca mantener una señal cercana a un valor constante o suavemente variable, como la velocidad angular en este proyecto. También donde no se requiere una corrección anticipada basada en la pendiente del error, como lo haría un término derivativo. Entre sus principales ventajas se encuentran su capacidad para reducir el error en estado estacionario y su relativa robustez ante perturbaciones [29]. Sin embargo, el controlador PI también presenta algunas limitaciones como la ausencia de un término derivativo que puede hacer que su desempeño frente a cambios rápidos o transitorios no sea tan fino. Desde el punto de vista computacional, el PI es eficiente y poco exigente, lo que lo hace especialmente adecuado para su implementación en microcontroladores con recursos limitados.

5.3.2 PID

El controlador proporcional-integral-derivativo, conocido como PID, extiende la estructura del PI al incluir un tercer componente: la acción derivativa. Esta se basa en la tasa de cambio del error y permite anticipar su comportamiento futuro, mejorando la respuesta del sistema ante variaciones rápidas. Mientras el término proporcional responde al error presente y el integral corrige errores acumulados en el tiempo, el derivativo introduce una acción correctiva basada en la pendiente del error, actuando como un amortiguador que reduce la posibilidad de sobreimpulso y oscilaciones [29]. En la Figura 28 se muestra el diagrama de bloques de este controlador.

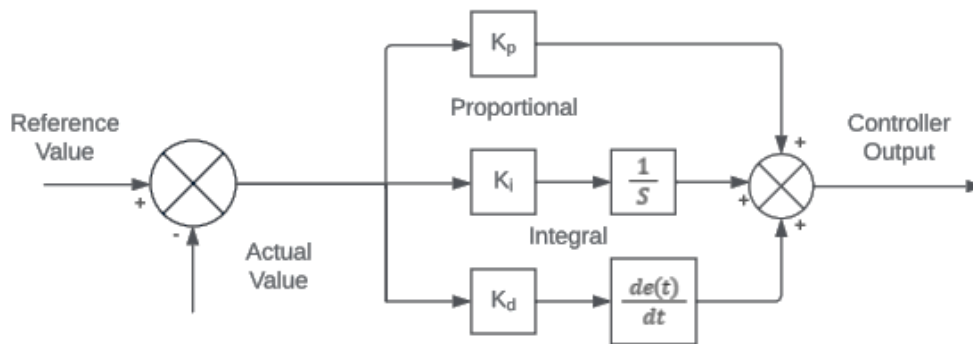


Figura 28. Diagrama de bloques de un controlador proporcional-integral-derivativo. Fuente: [30]

Gracias a esta combinación, el PID ofrece un control más fino y estable en sistemas con dinámicas complejas o que requieren respuestas más precisas ante perturbaciones repentinas. Es especialmente útil en procesos donde se desea una rápida corrección sin sacrificar estabilidad, como en sistemas mecánicos con inercia significativa o en control de posición. Sin embargo, su sensibilidad al ruido debido a la acción derivativa y su mayor complejidad de ajuste de ganancias pueden representar desventajas, sobre todo en sistemas donde las mediciones son discretas o imprecisas. Además, aunque su carga computacional sigue siendo relativamente baja, es superior a la del PI, lo cual puede ser un factor a considerar en plataformas con recursos limitados.

5.3.3 I-PD

El controlador I-PD es una variante estructural del PID tradicional que distribuye sus componentes de manera diferente: la acción integral se aplica directamente a la señal de control, mientras que las acciones proporcional y derivativa se aplican únicamente a la señal de retroalimentación. En otras palabras, la referencia solo afecta al integrador, y no directamente a los términos proporcional ni derivativo. Esta configuración permite que el controlador mantenga las propiedades de rechazo de perturbaciones propias del PID, pero reduce la influencia directa de la referencia sobre el comportamiento dinámico del sistema, lo cual puede ser útil en aplicaciones donde se desea minimizar el sobreimpulso o evitar una respuesta agresiva ante cambios en la referencia [31]. En la Figura 29 se muestra el diagrama de bloques de este controlador.

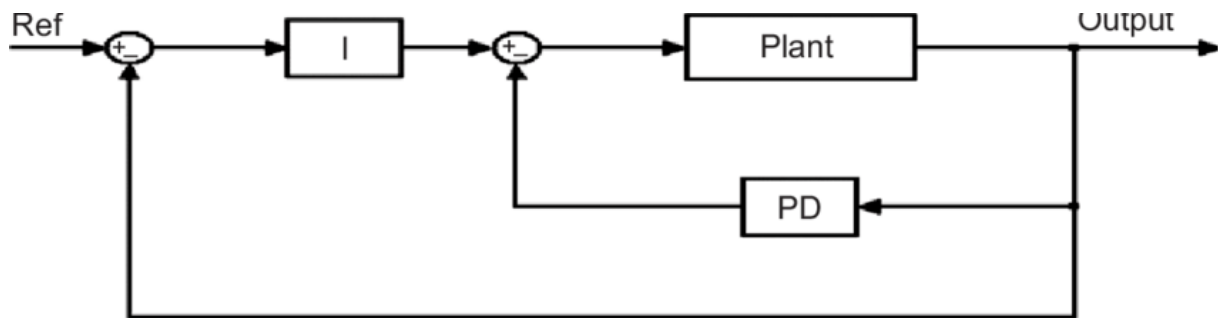


Figura 29. Diagrama de bloques de un controlador I-PD. Fuente: [32]

Una de las principales ventajas del I-PD es su mejor comportamiento frente a variaciones abruptas de referencia ya que, al no afectar directamente al término proporcional, se atenúan las oscilaciones y se facilita el cumplimiento de requisitos de suavidad en la transición. Además, permite una implementación más estable en sistemas donde la referencia cambia en escalones o donde hay limitaciones físicas que podrían ser exigidas innecesariamente por un controlador PID convencional [31].

No obstante, esta configuración también tiene limitaciones. La separación entre la referencia y los términos proporcional y derivativo puede dificultar el ajuste de la respuesta dinámica, y su implementación requiere un análisis cuidadoso de la arquitectura del sistema de control. Desde el punto de vista computacional, el I-PD no representa una carga significativamente mayor que el PID clásico.

5.3.4 Realimentación de estado integral

La realimentación de estado con acción integral (REI) es una técnica de control que combina el uso de todas las variables de estado del sistema con una acción integradora sobre el error entre la salida y la referencia. Su objetivo principal es garantizar que, ante entradas constantes o perturbaciones sostenidas, el error en estado estacionario tienda a cero. Para lograrlo, se introduce una nueva variable de estado que acumula el error en el tiempo, y se la incluye en la acción de control junto con las demás variables del sistema [33]. En la Figura 30 se muestra el diagrama de bloques de este controlador.

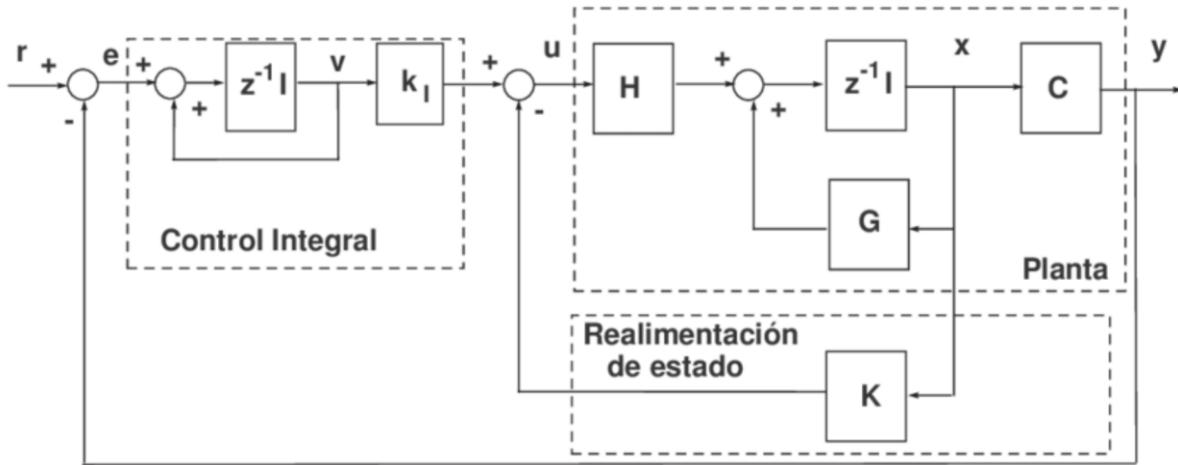


Figura 30. Diagrama de bloques de un controlador de realimentación de estado integral.

Fuente: [34]

Entre sus principales ventajas se encuentra su capacidad para cumplir con especificaciones dinámicas exigentes, su adaptabilidad frente a perturbaciones constantes y su aplicabilidad en sistemas donde es posible medir o estimar adecuadamente todas las variables de estado. No obstante, una de sus principales limitaciones radica en la necesidad de contar con un modelo matemático preciso del sistema y con sensores suficientes para acceder a las variables de estado, o en su defecto, emplear observadores de estado que incrementan la complejidad computacional. En consecuencia, aunque el enfoque REI ofrece ventajas teóricas importantes, su alto consumo de recursos puede dificultar su implementación práctica [33].

5.3.5 Selección de concepto de controlador

Siguiendo la metodología establecida, se realizó una comparación entre las alternativas consideradas para el diseño del controlador encargado de regular la velocidad angular del Rover. Las opciones analizadas incluyen las estrategias PI, PID, I-PD y realimentación de estado con acción integral (REI). Para la evaluación se definieron cuatro criterios principales: desempeño dinámico, adaptabilidad a variaciones del entorno, compatibilidad con el hardware disponible y complejidad de implementación.

- **Desempeño dinámico** considera la capacidad del controlador para cumplir con los indicadores de respuesta transitoria establecidos en las especificaciones, como tiempo de estabilización, sobreimpulso y tiempo de respuesta. Este criterio se vincula directamente con las necesidades identificadas por el cliente relacionadas con la estabilidad del sistema (necesidad 3), su capacidad de responder rápidamente ante cambios en las condiciones del terreno (necesidad 4), y la evitación de variaciones abruptas al alcanzar la velocidad deseada (necesidad 5). También está relacionado con la necesidad 2, que establece que el sistema debe ejecutar el control de velocidad de forma autónoma e implica que el comportamiento del sistema debe ser confiable y estable sin intervención externa. Estas necesidades reflejan la importancia de que el sistema tenga un comportamiento predecible y controlado ante distintas condiciones operativas.
- **Adaptabilidad a variaciones del entorno** evalúa la capacidad del controlador para mantener un desempeño aceptable ante posibles cambios en la carga, la superficie de rodamiento o perturbaciones externas durante la operación. Este criterio responde directamente a la necesidad 6, que establece que el sistema debe funcionar en varios terrenos distintos, lo cual implica enfrentar condiciones que afectan la tracción y el comportamiento dinámico del sistema. Aunque estas condiciones no pueden modelarse con precisión en esta etapa, el diseño del controlador debe mostrar suficiente tolerancia a estas variaciones.
- **Compatibilidad con el hardware** analiza si la solución propuesta puede implementarse de forma efectiva en el microcontrolador Arduino UNO R3, considerando sus limitaciones de procesamiento y capacidad de almacenamiento, así como su viabilidad con los sensores disponibles en este proyecto. Esta restricción responde a la necesidad práctica de desarrollar el sistema utilizando el Rover actualmente disponible, que se trata de un prototipo en fase inicial. Por motivos presupuestarios y debido a los plazos que implica la adquisición de componentes mediante fondos del TEC, no es factible destinar recursos significativos para reemplazar partes fundamentales del hardware del Rover en esta etapa. Si bien es posible sustituir algunos componentes menores, el controlador seleccionado debe ser compatible con la base de hardware existente.

- **Complejidad de implementación** contempla la dificultad asociada al diseño, ajuste de parámetros y puesta en marcha del controlador, así como la necesidad de conocimientos avanzados o herramientas adicionales. Aunque no está vinculada directamente con una necesidad específica del cliente, en ingeniería de control y modelado se reconoce que la simplicidad en el diseño es un valor en sí mismo: se busca lograr la funcionalidad deseada con la menor complejidad posible, lo que favorece la implementación del sistema. Sin este criterio, el proceso de selección tendería inevitablemente hacia soluciones más complejas, incluso si las mejoras en desempeño que ofrecen son mínimas en relación con el aumento en esfuerzo, costo o riesgo de implementación que implican.

Cada alternativa fue evaluada con valores cualitativos positivos (+), neutros (0) o negativos (-), según su desempeño frente a cada criterio. La evaluación neta se calculó como la diferencia entre la cantidad de aspectos positivos y negativos. La Tabla 5 presenta el resultado del análisis comparativo.

Tabla 5. Matriz de selección de conceptos de controlador.

Criterios de Selección	Conceptos			
	A	B	C	D
	PI	PID	I-PD	REI
Desempeño dinámico	0	+	+	0
Adaptabilidad al entorno	0	+	+	0
Compatibilidad con hardware	+	+	0	-
Complejidad de implementación	+	0	-	-
Suma +	2	3	2	0
Suma 0	2	1	1	2
Suma -	0	0	1	2
Evaluación neta	2	3	1	-2
Lugar	2	1	3	4

El análisis comparativo se observa que el controlador PID obtuvo la evaluación neta más alta, con un total de +3, destacándose por su equilibrio entre desempeño dinámico, adaptabilidad al entorno y viabilidad de implementación. Aunque implica un leve aumento en el consumo computacional respecto al PI, este incremento es manejable dentro de los márgenes disponibles del microcontrolador utilizado. Además, si bien su componente derivativo puede introducir sensibilidad al ruido, su ajuste adecuado permite obtener una mejor respuesta ante perturbaciones externas. Esto lo convierte en una opción favorable para entornos variables como los contemplados en este proyecto.

La segunda mejor opción fue el controlador PI, con una evaluación neta de +2. Su simplicidad y bajo consumo de recursos lo hacen muy atractivo para implementaciones en plataformas con restricciones. No obstante, su principal limitación frente al PID es la ausencia del componente derivativo, lo que reduce su capacidad para anticiparse a variaciones del sistema. En consecuencia, si bien cumple adecuadamente con los requisitos básicos del control de velocidad, su respuesta puede no ser tan eficaz en escenarios donde las condiciones del terreno cambian y presentan obstáculos.

Los controladores I-PD y REI obtuvieron evaluaciones netas de +1 y -2 respectivamente, siendo descartados por su mayor complejidad y menor compatibilidad con el hardware del prototipo. El I-PD mostró desventajas en los criterios de compatibilidad y complejidad, se evaluó como menos compatible con el hardware que un PI o PID debido a que requiere ligeramente más recursos computacionales que un PID. Por otro lado, también es más compleja su implementación cuando se compara con un control PI o PID que son muy comunes y estandarizados.

En el caso del controlador REI, este posee un desempeño teórico prometedor, ya que la realimentación de estado con acción integral permite un control preciso, con rechazo de perturbaciones constantes y buen seguimiento de referencia. No obstante, en el contexto específico de este proyecto, no fue posible explotar ese potencial debido a limitaciones clave en la implementación: no se cuenta con un modelo matemático detallado de la planta, ni con sensores suficientes para acceder a todas las variables de estado, ni con un observador de estados que permita estimarlas. Por ello, tanto en el criterio de desempeño dinámico como en el de adaptabilidad al entorno, se le asignó una evaluación neutra (0), ya que aunque su

desempeño podría ser sobresaliente en teoría, no puede garantizarse en la práctica con la infraestructura disponible. De haber contado con los medios necesarios para su implementación completa, habría sido razonable asignarle una evaluación más alta; sin embargo, en condiciones reales de operación con el Rover y el hardware disponible, su comportamiento sería limitado.

En conjunto, los resultados justifican la selección del controlador PID (concepto B) como la solución más apropiada para el sistema de control de velocidad del Rover.

5.4 Almacenamiento de datos experimentales

Durante las pruebas del sistema de control, resulta indispensable poder almacenar los datos generados, particularmente las mediciones de velocidad obtenidas. Esta información es crucial para evaluar el desempeño del controlador y analizar el comportamiento del Rover en distintos escenarios. Sin embargo, el microcontrolador Arduino UNO R3 no dispone de un sistema de archivos ni de la capacidad de memoria suficiente para crear y guardar archivos en formato .csv o .txt de forma interna. Por esta razón, se requiere el uso de un módulo externo que permita registrar y conservar los datos experimentales de manera estructurada y accesible para su posterior análisis.

Con este fin, se seleccionó el módulo OpenLog de SparkFun, un registrador de datos de código abierto que opera mediante una conexión serial simple y permite almacenar información en tarjetas microSD de hasta 32 GB. Este dispositivo actúa como una “caja negra”, registrando en tiempo real todos los datos que recibe por el puerto serial, lo que lo hace ideal para aplicaciones científicas o de validación de sistemas.

El OpenLog está basado en un microcontrolador ATmega328 funcionando a 16 MHz, y destaca por su bajo consumo energético: aproximadamente 2–3 mA en estado inactivo y entre 10 y 20 mA durante la escritura activa, dependiendo de la tarjeta microSD utilizada. Es compatible con tarjetas microSD desde 512 MB hasta 32 GB formateadas en FAT16 o FAT32. Todos los datos son almacenados directamente en la tarjeta, lo que permite su posterior análisis en cualquier sistema compatible. Además, cuenta con dos indicadores LED que permiten visualizar el estado de escritura en tiempo real.

Debido a sus dimensiones compactas (4x15x19 mm), su integración en el sistema es sencilla y no representa una carga significativa en términos de espacio o consumo. En la Figura 31 se muestra el módulo OpenLog utilizado, con vistas de ambos lados para ilustrar sus características físicas y sus pines de conexión.

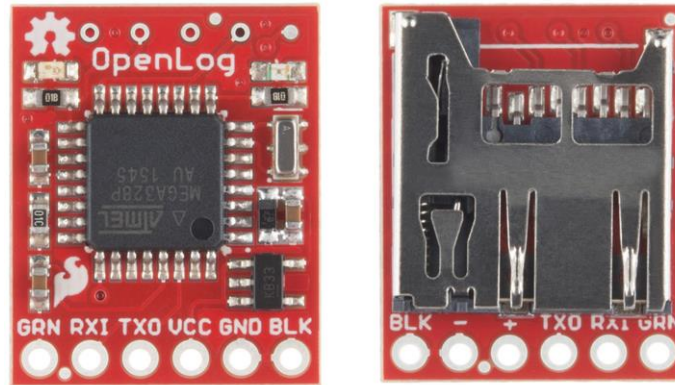


Figura 31. Vista frontal y posterior del módulo OpenLog de SparkFun.

Para la conexión del módulo OpenLog se utilizaron únicamente los pines RX, VCC y GND, ya que en este proyecto solo se requiere que el módulo reciba datos provenientes del Arduino; no es necesario transmitir información de vuelta, por lo que el pin TX permanece sin uso.

En la placa de expansión del Rover, se aprovecharon los pines originalmente destinados al sensor ultrasónico, ya que este conector proporciona una línea de alimentación de 5 V, GND y acceso al pin digital D10, el cual se configuró como salida serial mediante la librería *SoftwareSerial* para enviar los datos al OpenLog. La Figura 32 muestra los pines utilizados en la placa de expansión.

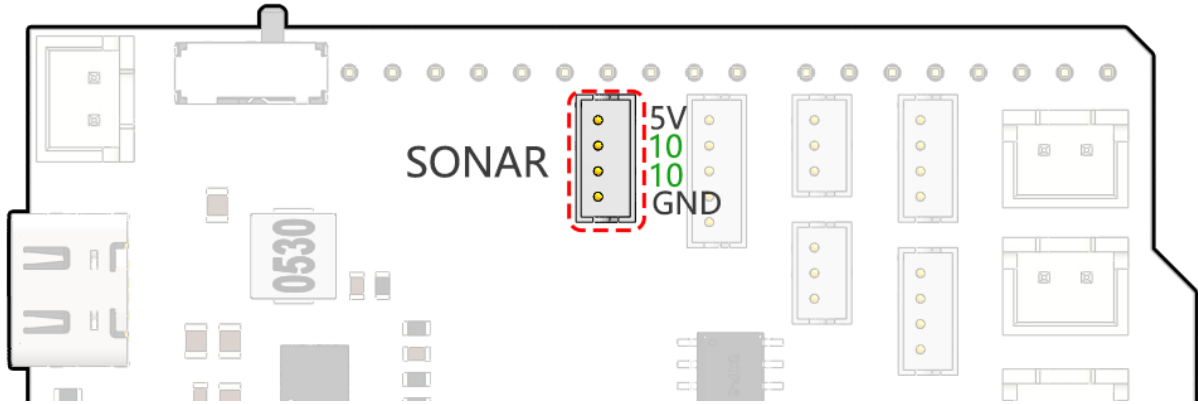


Figura 32. Ubicación y pines del conector del sensor ultrasónico utilizado para la conexión del OpenLog.

Para el almacenamiento de los datos, se seleccionó una tarjeta Maxell microSD de 16 GB modelo MCSD16CL10, la cual cumple con las especificaciones necesarias para operar correctamente con el módulo OpenLog. Esta tarjeta es compatible con el sistema de archivos FAT32 y ofrece una capacidad ampliamente suficiente para almacenar los datos generados durante los experimentos. Además, su clasificación Clase 10 garantiza una velocidad de escritura adecuada para registrar datos en tiempo real sin pérdidas. La Figura 33 muestra la tarjeta microSD utilizada en este proyecto.



Figura 33. Tarjeta Maxell microSD de 16 GB Clase 10 utilizada para el almacenamiento de datos.

Para asegurar una comunicación adecuada entre el Arduino y el módulo OpenLog, fue necesario configurar sus parámetros de operación mediante un archivo de configuración almacenado en

la tarjeta microSD. Este archivo, denominado config.txt, define los valores de funcionamiento que el OpenLog carga automáticamente al iniciar. En este proyecto, se utilizó la configuración "115200,26,3,0,1,0,0". Esta línea indica que el OpenLog se comunica a una velocidad de 115200 baudios, utiliza el carácter ASCII 26 como comando de escape, permite hasta tres repeticiones del carácter de escape, opera en modo estándar, activa mensajes para diagnóstico, desactiva el eco de los comandos enviados y no ignora los datos recibidos por la línea RX. Esta configuración es compatible con la librería *SoftwareSerial* utilizada en el Arduino UNO R3, y permite un registro confiable de los datos enviados desde el sistema de sensado.

Para asegurar una escritura eficiente de los datos, se configuró el módulo OpenLog con una velocidad de transmisión de 115200 baudios, que corresponde al valor máximo permitido por este dispositivo. Esta configuración permite reducir el tiempo necesario para almacenar cada muestra de datos. Según las pruebas realizadas, el tiempo de escritura por muestra con este baudrate es de aproximadamente 6 ms.

5.5 Propuesta de diseño del sistema de realimentación de velocidad

De acuerdo con el proceso de selección descrito en la sección anterior, se optó por desarrollar un sistema de sensado basado en sensores Hall. Esta sección detalla el diseño del sistema seleccionado, incluyendo su principio de funcionamiento, ubicación, procesamiento de señales y limitaciones.

5.5.1 Principio de funcionamiento

La magnitud física de interés en este proyecto es la velocidad lineal del chasis del Rover, ya que constituye la variable que se desea controlar. No obstante, dicha velocidad no se mide de forma directa, sino que se estima indirectamente a partir de la velocidad angular de las ruedas motrices. Para ello, se utiliza un sensor Hall que detecta el paso periódico de imanes acoplados a una de las ruedas, lo que permite calcular su velocidad angular con base en la frecuencia de los pulsos generados.

Este sistema de medición opera, en esencia, como un *encoder* magnético. En esta configuración, uno o más imanes se fijan al eje de una rueda, mientras que el sensor Hall se ubica en una posición fija, próxima a la trayectoria de rotación. Cada vez que un imán pasa frente al sensor, este genera un pulso eléctrico que es enviado al microcontrolador. Al contar la cantidad de pulsos detectados en un intervalo de tiempo definido, es posible calcular la velocidad angular de la rueda mediante la Ecuación 1.

$$\omega = \frac{2\pi \cdot 1000}{\Delta t \cdot N} \quad (5.1)$$

Donde:

- ω : Velocidad angular de la rueda (rad/s)
- Δt : Tiempo entre pulsos consecutivos (ms)
- N : Número de pulsos generados por cada revolución completa de la rueda (pulsos/rev)
- 2π : Radianes en una vuelta completa
- 1000 : Factor de conversión de milisegundos a segundos

Este sistema utiliza el tiempo entre pulsos para calcular la velocidad angular, en lugar de contar la cantidad de pulsos dentro de un intervalo fijo. Esta decisión se fundamenta en que, a bajas velocidades como las previstas en este proyecto, el método basado en frecuencia pierde precisión, ya que pueden ocurrir intervalos de muestreo en los que no se detecta ningún pulso, generando lecturas erróneas o nulas. En cambio, al medir el tiempo transcurrido entre dos pulsos consecutivos, se obtiene una estimación más precisa y continua de la velocidad angular.

Una vez determinada la velocidad angular, la velocidad lineal del chasis se obtiene mediante una relación geométrica directa, como se muestra en la Ecuación 2, que incorpora el radio de la rueda que es 37,5 mm en este caso.

$$v = \omega \cdot r \quad (5.2)$$

Donde:

- v : Velocidad lineal del chasis (m/s)
- ω : Velocidad angular de la rueda (rad/s)
- r : Radio de la rueda (m)

Adicionalmente, se calcula la velocidad en revoluciones por minuto (RPM) utilizando la Ecuación 3, lo cual resulta útil en las pruebas iniciales del sensor.

$$\omega_{RPM} = \frac{60000}{\Delta t \cdot N} \quad (5.3)$$

Donde:

- ω_{RPM} : Velocidad angular expresada en revoluciones por minuto (RPM)
- Δt : Tiempo entre pulsos consecutivos (ms)
- N : Número de pulsos generados por cada revolución completa (pulsos/rev)
- 60000 : Factor de conversión de milisegundos a minutos (1000 ms/s \times 60 s/min)

5.5.2 Implementación física del sistema de sensado

Para la implementación física del sistema, se seleccionó el sensor KY-024 debido a sus características adecuadas para el diseño del Rover. Este sensor Hall permite la detección de campos magnéticos a través de una salida digital y una analógica, lo que facilita su integración con microcontroladores como el Arduino UNO R3.

Entre sus ventajas destacan su tamaño compacto (32 mm x 14 mm x 6 mm), bajo peso y consumo eléctrico reducido (corriente típica menor a 10 mA). Estas propiedades lo hacen adecuado para el Rover de este trabajo.

El sensor cuenta con un potenciómetro que permite ajustar su sensibilidad, lo cual facilita su calibración en función del campo magnético generado por los imanes utilizados. Además, incorpora un LED indicador que se activa al detectar un campo magnético, lo que permite verificar visualmente su funcionamiento durante las pruebas. La Figura 34 muestra la ubicación del sensor KY-024.

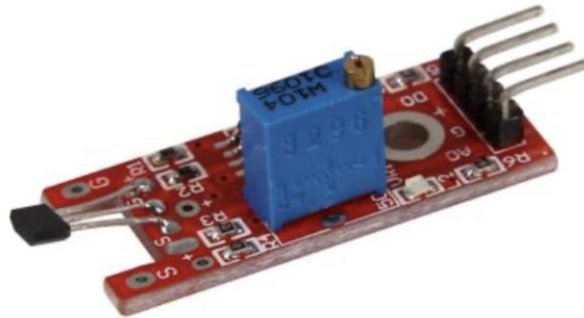


Figura 34. Módulo de sensor Hall modelo KY-024.

Dado que el Rover dispone únicamente de dos drivers, el sistema cuenta con una señal de control para los tres motores del lado izquierdo y otra para los tres del lado derecho. En el presente proyecto se contempla únicamente el desplazamiento en línea recta, sin control de orientación, por lo que ambos lados del Rover deben moverse a la misma velocidad para lograr un movimiento rectilíneo.

Bajo estas condiciones, resulta innecesario sensor la velocidad angular de más de un motor, ya que todos deben operar de forma sincronizada. Por esta razón, se decidió colocar un único sensor Hall en uno de los motores, lo que permite reducir el consumo eléctrico, el peso del sistema y la complejidad de las conexiones al microcontrolador.

El motor seleccionado corresponde a la rueda central del lado izquierdo. Esta elección se debe a dos motivos: en primer lugar, la rueda central está ubicada más cerca del centro de masa del Rover, por lo que su velocidad angular es más representativa de la velocidad lineal del chasis. En segundo lugar, la proximidad de este motor a la placa de expansión permite realizar una conexión más directa al sensor, utilizando cables más cortos y favoreciendo un cableado más ordenado. En la Figura 35 se muestra la ubicación del sensor sobre el motor.

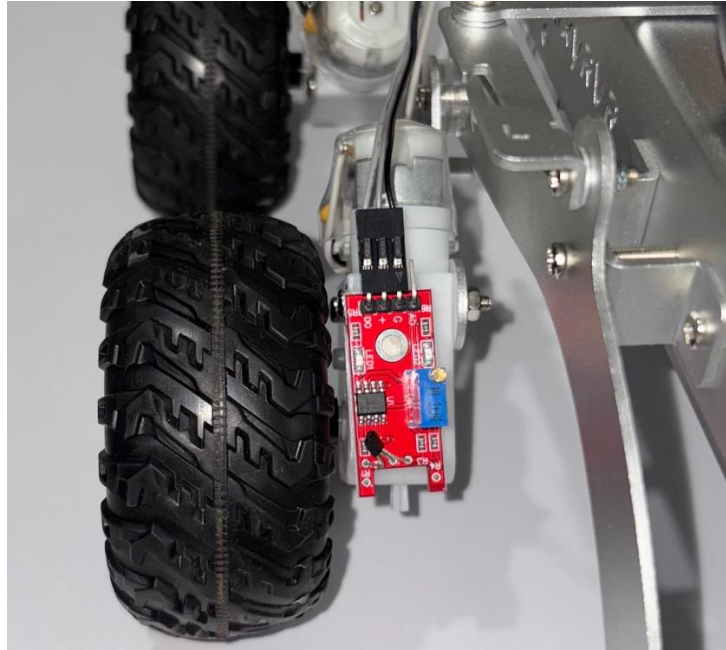


Figura 35. Montaje del sensor Hall en el Rover.

Para optimizar la detección de los imanes acoplados a la rueda, se ajustó la sensibilidad del sensor Hall mediante el potenciómetro integrado en el módulo. Este componente permite modificar el umbral de activación, facilitando la calibración del sensor en función de la intensidad del campo magnético percibido. En este proyecto, el potenciómetro se configuró al nivel máximo de sensibilidad.

Según la documentación del fabricante, el sensor no proporciona valores absolutos de magnitud física (como temperatura en °C o intensidad de campo en mT), sino que opera bajo un principio de umbral relativo. Es decir, el usuario define manualmente un valor límite, y el sensor emite una señal digital únicamente cuando el campo magnético supera dicho umbral. Esta característica, combinada con el LED indicador incluido en el módulo, permite una verificación visual rápida del funcionamiento durante las pruebas.

La placa de expansión del Rover, que contiene los drivers para los motores, ya utiliza todos los pines analógicos del Arduino Uno R3 para otras funciones internas del sistema. Además, esta placa está diseñada para ser compatible exclusivamente con los sensores incluidos de fábrica, lo que limita las opciones de conexión para sensores externos. Estas condiciones se describen

con mayor detalle en la sección correspondiente a la descripción general del Rover y sus subsistemas.

Por esta razón, para la conexión del sensor Hall se optó por utilizar uno de los puertos de sensor predefinidos en la placa, específicamente el correspondiente al sensor infrarrojo izquierdo. Este conector dispone exactamente de los pines necesarios para operar el módulo Hall: alimentación de 5V, conexión a tierra (GND) y un pin digital configurable para lectura. La Figura 36 muestra la disposición de estos pines en el conector seleccionado, el cual resulta adecuado tanto en funcionalidad como en cercanía al motor donde se instaló el sensor.

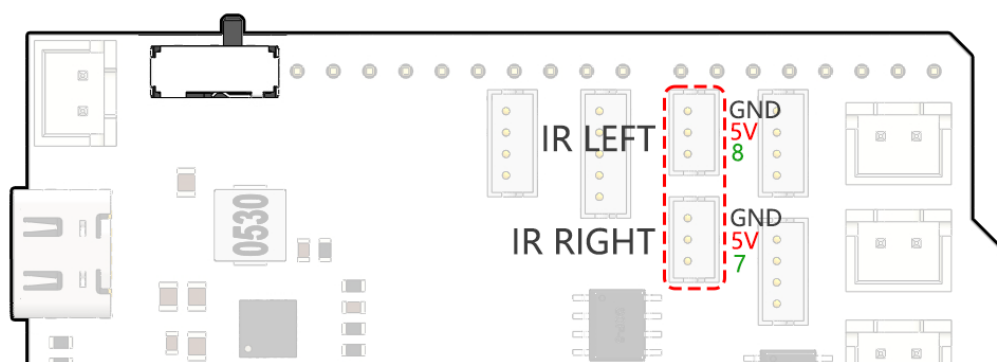


Figura 36. Ubicación y pines del conector del sensor infrarrojo izquierdo utilizado para la conexión del sensor Hall.

Dado que el sensor Hall se conecta al pin digital D8, no es posible utilizar la función *attachInterrupt()* de Arduino, ya que esta solo está disponible para un subconjunto específico de pines con soporte de interrupciones externas (en el Arduino Uno, únicamente los pines D2 y D3). Como consecuencia, la lectura del sensor debe realizarse mediante sondeo periódico (*polling*) del estado del pin digital dentro del ciclo principal del programa. Esta técnica implica consultar de forma continua el pin en cada iteración del *loop()*, lo cual introduce una carga computacional adicional. No obstante, esta estrategia resulta viable para el presente proyecto.

Por otra parte, se realizaron múltiples pruebas para definir la mejor configuración de imanes que permitiera una detección confiable mediante el sensor Hall. El objetivo era maximizar la resolución del sistema, entendida como la cantidad de pulsos por revolución, sin comprometer la precisión ni generar lecturas erróneas.

Se comprobó que el sensor Hall requiere una distancia mínima de aproximadamente 5 mm entre el imán y el sensor para activar correctamente su salida digital. Asimismo, se determinó que una mayor cercanía del imán al sensor reduce la probabilidad de errores. Inicialmente se evaluaron imanes individuales de 2 mm y 3 mm de espesor (8x2 mm y 5x3 mm), pero se concluyó que estos, por sí solos, no generaban un campo magnético lo suficientemente intenso. Ante esto, se exploraron combinaciones de varios imanes. Se descartó mezclar tamaños distintos, ya que los imanes del mismo tipo se adhieren mejor entre sí. También se definió una longitud máxima admisible de 6 mm para evitar interferencias mecánicas con la carcasa del motor o con el propio sensor. Por ejemplo, una pila de tres imanes de 3 mm (9 mm en total) resultó inviable. De las configuraciones evaluadas, se comprobó que dos imanes de 5x3 mm (6 mm) no proporcionaban suficiente intensidad magnética. En cambio, una pila de tres imanes de 8x2 mm alcanzaba el límite físico permitido (6 mm) y ofrecía el mejor desempeño, en parte gracias a su mayor diámetro (8 mm frente a 5 mm), lo que incrementa la superficie efectiva y la fuerza del campo. En la Figura 37 se muestran los imanes utilizados.



Figura 37. Imán de neodimio en forma de disco 8x2 mm, clasificación N35 (fuerza magnética estándar).

Una vez seleccionada la configuración magnética óptima, se procedió a colocar cuatro conjuntos de imanes equidistantes sobre una de las ruedas motrices del Rover, como se muestra en la Figura 38, generando así cuatro pulsos por cada revolución completa. Esta cantidad fue determinada como el máximo número de imanes que puede colocarse sin comprometer la confiabilidad de detección. Aunque un mayor número de pulsos incrementaría la resolución del

sistema, se observó que al colocar cinco o más imanes, la proximidad entre ellos impedía que el campo magnético decayera lo suficiente entre eventos consecutivos. Como resultado, el sensor Hall no retornaba a su estado bajo antes de activarse el siguiente imán, lo que dificultaba la detección individual de los pulsos. Por tanto, el uso de cuatro imanes representa el mejor equilibrio entre resolución y error dentro de las condiciones del proyecto.



Figura 38. Posicionamiento de los cuatro imanes para sensado de velocidad angular.

Es aspecto importante a destacar en la instalación de los imanes es la orientación de su polaridad. El sensor Hall utilizado en este proyecto responde únicamente al campo magnético generado por el polo sur del imán. Por lo tanto, es fundamental que, durante el montaje, los imanes se coloquen con el polo sur orientado hacia el sensor. En caso de que se instalen con la polaridad invertida, el campo magnético no superará el umbral de activación del sensor, impidiendo la generación de pulsos y, por ende, afectando el funcionamiento del sistema de sensado. Este detalle fue verificado experimentalmente durante las pruebas preliminares y debe ser cuidadosamente considerado al posicionar los imanes sobre la rueda.

Bajo esta configuración, la resolución angular es de 90 grados por pulso, o $\frac{\pi}{2}$ radianes por pulso. Este valor se obtiene dividiendo el total de la rotación circular, 360° o 2π radianes, entre la cantidad de pulsos por revolución. Por otro lado, la resolución temporal del sistema está definida por la función *millis()* de Arduino, utilizada para medir el tiempo entre pulsos. Esta función tiene una resolución mínima de un milisegundo, lo que representa el menor cambio

detectable en el intervalo de tiempo entre pulsos. En consecuencia, la velocidad angular calculada solo puede variar en incrementos determinados por ese intervalo mínimo.

Como el cálculo de las revoluciones por minuto se basa en el tiempo entre pulsos, este proceso no se realiza a intervalos regulares, sino únicamente cuando se detecta un nuevo pulso. Es decir, la frecuencia de cálculo es variable y depende directamente de la velocidad de rotación. Sin embargo, para el modelado del sistema y el análisis del comportamiento dinámico, es indispensable contar con datos de velocidad registrados a un intervalo de muestreo constante. Para no comprometer la precisión de la medición, se optó por mantener el último valor de velocidad calculado hasta que se detecte un nuevo pulso, y registrar ese valor de forma periódica.

Este registro periódico se implementó con una frecuencia de muestreo constante de 50 ms. Esta decisión se basa en que, a la mayor velocidad considerada para el sistema, el tiempo entre pulsos ronda los 200 ms. Dado que cada pulso corresponde a un cuarto de revolución, por la presencia de cuatro imanes en la rueda, esto implica una revolución completa cada 800 ms. Por lo tanto, se obtienen aproximadamente 16 registros por cada vuelta de la rueda, lo cual permite una representación suficientemente detallada del comportamiento del sistema. Además, se consideró el tiempo requerido para escribir datos en la memoria microSD mediante el módulo OpenLog, asegurando así la estabilidad del proceso sin pérdidas ni retrasos. Un muestreo más frecuente no aportaría mejoras significativas en precisión y solo incrementaría innecesariamente la carga de procesamiento del microcontrolador.

En la siguiente sección se documentan las pruebas realizadas al sistema de sensado, con el fin de verificar su funcionamiento.

5.5.3 Validación del sistema de sensado

Para la validación del sistema de sensado de velocidad, se utilizó como referencia externa la aplicación móvil RPM Meter. En particular, se empleó el módulo Camera RPM, el cual permite estimar la velocidad angular de un objeto en rotación utilizando la cámara del teléfono móvil.

El entorno de prueba se configuró de la siguiente manera: el sensor Hall fue conectado al Arduino UNO R3 y acoplado a un motor con rueda, replicando la disposición del sistema en el Rover. El Arduino fue alimentado y comunicado con una computadora portátil mediante un cable USB, permitiendo cargar el programa y visualizar las mediciones en el monitor serial de la plataforma Arduino IDE. Paralelamente, se utilizó un teléfono móvil con la aplicación RPM Meter, la cual, al utilizar la cámara del dispositivo, permite ajustar manualmente un valor de RPM en pantalla. Este valor se incrementa o disminuye hasta que la imagen capturada por la cámara coincide visualmente con el movimiento real de la rueda, haciendo que esta parezca inmóvil. Ese punto indica que la velocidad angular ingresada es igual a la de la rueda, y se toma como valor de referencia.

Para mantener estable la posición del teléfono móvil durante la prueba, este se fijó a la superficie mediante cinta adhesiva de doble cara, colocándolo a una distancia aproximada de 15 cm de la rueda. Detrás del teléfono se colocó un soporte a modo de cuña (un recipiente cerámico con peso en su interior) para mantener la inclinación del dispositivo sin obstruir la cámara.

Una vez activado el motor, se marcó visualmente un punto de referencia en la rueda para facilitar la identificación del movimiento. A continuación, se ajustó el valor de RPM en la aplicación hasta lograr una coincidencia visual con el movimiento de la rueda, registrando este valor como referencia para comparar con la medición obtenida por el sistema basado en el sensor Hall.

Con el fin de validar el funcionamiento del sistema de sensado, se realizaron tres pruebas experimentales utilizando distintos niveles de señal de control PWM: 70, 120 y 200. Para cada nivel de PWM se llevaron a cabo 10 mediciones independientes, sumando un total de 30 mediciones en todo el experimento. Cada medición consistió en activar el motor durante un minuto y ajustar manualmente en la aplicación RPM Meter el valor de RPM hasta que coincidiera visualmente con la rotación observada, aprovechando este tiempo para probar distintos valores y encontrar el más preciso. Dado que el sistema registra datos a una frecuencia constante de 50 ms, cada repetición generó aproximadamente 1200 muestras. Para reducir esta gran cantidad de datos a un único valor representativo, se calculó la media aritmética de las 1200 muestras obtenidas en cada repetición, y ese valor promedio se utilizó como velocidad medida para efectos de comparación. Posteriormente, para cada conjunto de 10 valores

promedio correspondientes a un mismo nivel de PWM, se calcularon estadísticas descriptivas como la media, el error respecto al valor de referencia y la desviación estándar. Estos resultados se presentan en la Tabla 6. Las mediciones detalladas están disponibles en el Apéndice B.

Tabla 6. Resumen y estadísticas de los resultados de la validación del sistema de sensado de velocidad angular.

Resultados	Entrada PWM		
	70	120	200
Media	27,05	37,96	48,94
Desv. estándar	0,51	0,44	0,55
Error (%)	0,74	0,93	0,69

Los resultados obtenidos muestran un desempeño consistente y preciso del sistema de sensado en los tres niveles de entrada PWM evaluados. Las medias de RPM registradas muestran una progresión coherente con el aumento de la señal de control, lo que indica una adecuada sensibilidad del sistema. Además, los valores de desviación estándar se mantuvieron bajos en todos los casos con un máximo de 0.55, reflejando una alta repetibilidad en las mediciones. El error porcentual respecto al valor de referencia fue inferior al 1 % en los tres niveles, con un máximo de 0,93 %, lo cual confirma que el sistema presenta una excelente precisión para las condiciones de prueba planteadas. Estos resultados validan la confiabilidad del sistema de medición implementado y lo califican como adecuado para ser utilizado en el lazo de control del prototipo.

5.6 Propuesta de diseño del controlador

Como resultado del proceso de selección de conceptos descrito en secciones anteriores, se determinó que la estrategia de control más adecuada para este sistema es el controlador proporcional-integral-derivativo (PID). En esta sección se presenta su desarrollo.

5.6.1 Modelado del sistema

Para orientar el diseño y validar el desempeño del sistema de control, se definieron tres velocidades objetivo que representan los casos de uso del Rover durante las pruebas experimentales. La Tabla 7 muestra la relación entre la señal PWM aplicada, la velocidad lineal esperada del Rover y su correspondiente velocidad angular, expresada tanto en radianes por segundo como en revoluciones por minuto. Estos valores permiten comprender el rango operativo del sistema y sirvieron como base para la toma de datos del modelo. Se seleccionó la velocidad de 15 cm/s como punto de operación para la identificación del modelo de la planta, al tratarse del valor intermedio entre los casos definidos y representar adecuadamente el comportamiento del sistema.

Tabla 7. Relación entre velocidad lineal, velocidad angular y señal PWM aplicada.

	Velocidad lineal (cm/s)	Velocidad angular (rad/s)	Velocidad angular (rpm)	PWM asociado
1	10	2,67	25,46	32
2	15	4,00	38,20	41
3	20	5,33	50,93	52

Para obtener un modelo del comportamiento de los motores del Rover, se empleó la herramienta de identificación de sistemas de MATLAB, utilizando datos experimentales de entrada y salida. Se aplicó una señal PWM constante con un valor de 41, correspondiente a una velocidad lineal de 15 cm/s, tal como se detalló anteriormente. La salida del sistema fue la velocidad angular en radianes por segundo, medida mediante el sistema de sensado implementado. Los datos se registraron durante 15 segundos con un tiempo de muestreo fijo de 50 ms. Este procedimiento se repitió en seis ocasiones independientes con el objetivo de obtener una base de datos para la identificación del modelo.

En la Figura 39 se presentan las señales de entrada (PWM) y salida (velocidad angular en rad/s) utilizadas en el proceso de identificación del modelo dinámico. La señal de entrada corresponde a una serie de escalones de valor constante, mientras que la salida muestra la respuesta dinámica medida por el sistema de sensado. Los datos han sido divididos en dos conjuntos: los primeros tres ciclos corresponden al conjunto de estimación, mientras que los tres restantes fueron

reservados para la validación del modelo. Esta separación se representa mediante el cambio de color en las curvas, permitiendo visualizar claramente el segmento de datos destinado a cada etapa del proceso.

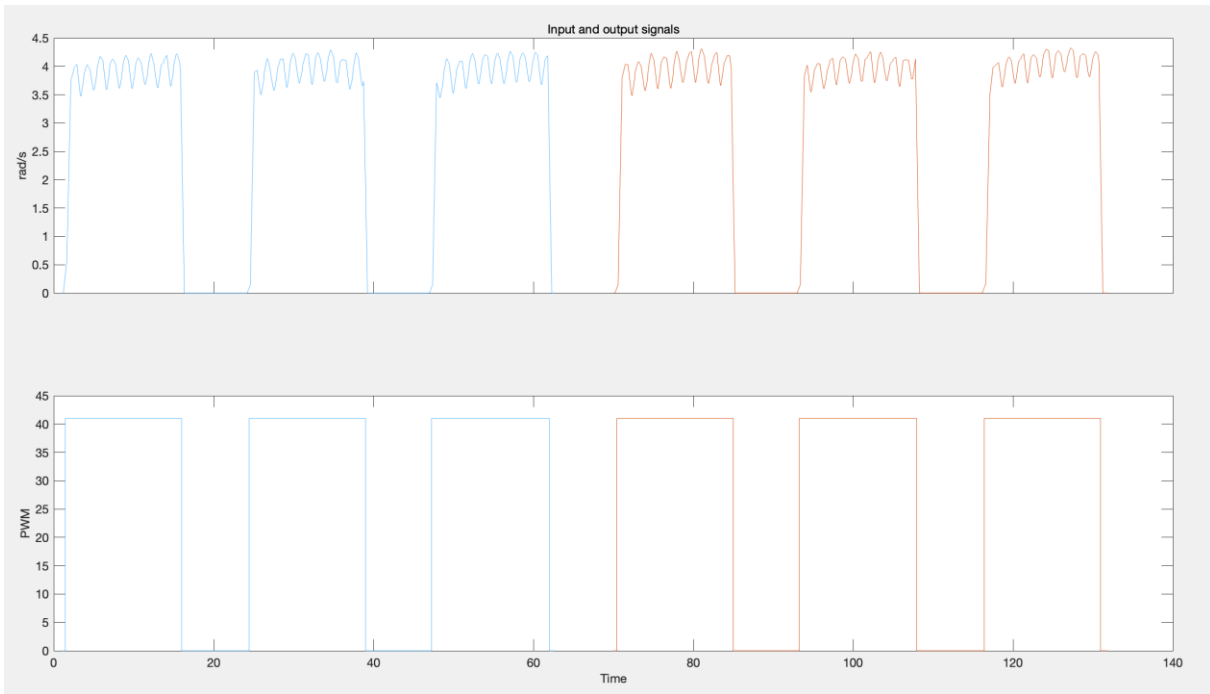


Figura 39. Datos de estimación y validación del modelo para los motores.

Antes de llevar a cabo el proceso de identificación, se aplicó un filtro de media móvil a los datos de salida con una ventana de tamaño 10 muestras, con el fin de suavizar las variaciones abruptas y facilitar la estimación del modelo. Como se muestra en la Figura 40, la señal original presenta tramos con valores constantes que no reflejan el comportamiento dinámico real del motor. Esta característica es consecuencia del método de sensado utilizado, en el cual se guarda el último valor de velocidad medido hasta la detección de un nuevo pulso. Dado que la frecuencia de muestreo del sistema es mayor que la frecuencia de pulsos a velocidades bajas, es esperable que varios registros consecutivos contengan el mismo valor. El filtrado atenúa este efecto, reduciendo las repeticiones y resaltando la tendencia dinámica de la respuesta, lo cual permite obtener un modelo más representativo del sistema real.

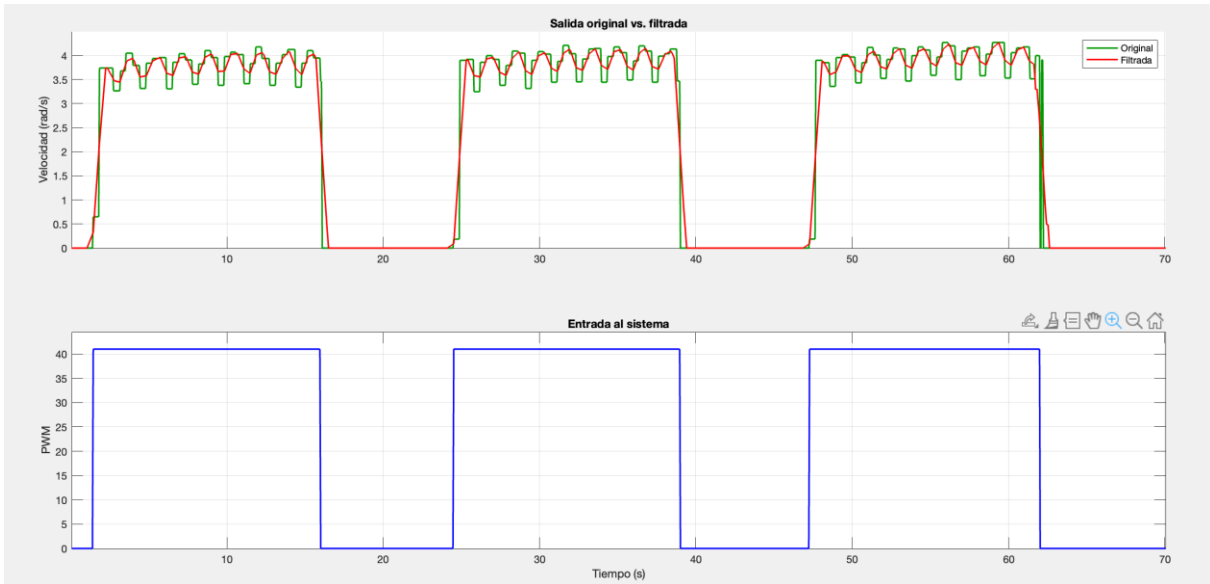


Figura 40. Señal de salida filtrada.

A partir de los datos experimentales previamente descritos, se obtuvo una función de transferencia discreta utilizando el apoyo de Matlab. Se decidió obtener una función de primer orden debido a que al aumentar los polos y ceros no se encontraron mejoras significativas de ajuste y en algunos casos este incluso disminuía. Además, de acuerdo a la búsqueda externa realizada una función de transferencia de primer orden puede describir un motor de corriente de continua y brindar excelentes resultados [35]. Durante el proceso de ajuste del modelo, también se consideró la inclusión de un retardo en la respuesta del sistema, utilizando distintos valores de tiempo. Sin embargo, se observó que esta modificación no tuvo ningún efecto significativo en la mejora del ajuste del modelo a los datos experimentales.

La Figura 41 muestra la comparación entre la salida medida y la respuesta simulada del modelo ajustado. La mejor función identificada alcanzó un nivel de ajuste del 81,68 % respecto a los datos de validación.

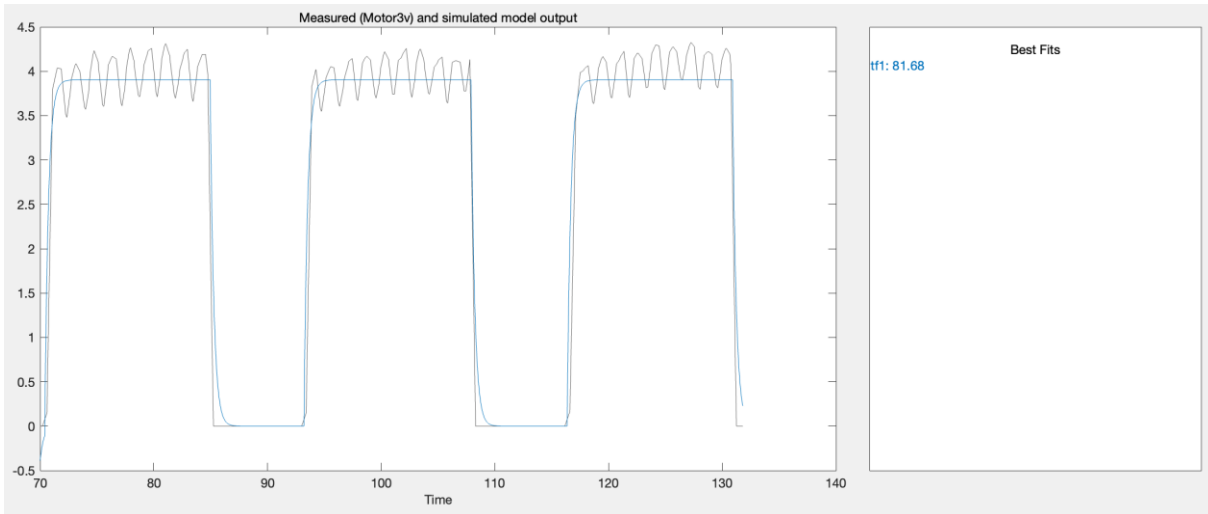


Figura 41. Ajuste del modelo estimado a los datos experimentales de validación.

Aunque en general se considera deseable un ajuste superior al 90 %, en este caso particular la limitación se debe a las restricciones físicas del sistema de sensado implementado. Como se explicó anteriormente, el sensor Hall detecta únicamente cuatro pulsos por revolución, lo que establece un límite en la resolución de las mediciones. Esta baja resolución dificulta la captura precisa de pequeñas variaciones en la respuesta dinámica del sistema. No obstante, como se evidenció en el análisis de la señal antes de ser filtrada, el hecho de que aparezcan valores constantes en tramos breves indica que la frecuencia de muestreo utilizada es superior al intervalo entre pulsos detectados. Esto confirma que no se están perdiendo eventos relevantes y que el sistema de adquisición está capturando todos los cambios que el sensor puede registrar. En otras palabras, se está extrayendo el máximo nivel de detalle posible sobre el comportamiento de los motores, dentro de las limitaciones físicas impuestas por la resolución del sistema de sensado. Bajo estas condiciones, un ajuste del 81,68 % se considera aceptable y representa un punto de partida adecuado para el diseño e implementación del controlador.

La función de transferencia discreta obtenida para el sistema es la siguiente:

$$G(z) = \frac{0.01388}{1 - 0.8542 z^{-1}} \quad (5.4)$$

Esta representación en términos de z^{-1} es común en sistemas discretos, ya que este término indica un retardo de un paso de muestreo. Sin embargo, para ciertas aplicaciones de análisis y

diseño en el dominio z , es útil expresar la función de transferencia en términos de potencias positivas. Para ello, se multiplica el numerador y el denominador por z , obteniéndose:

$$G(z) = \frac{0.01388 z}{z - 0.8542} \quad (5.5)$$

Este resultado proporciona la base para el desarrollo del controlador, el cual se detalla en la siguiente sección.

5.7 Diseño del controlador PID

Para el diseño del controlador, se implementó un lazo de control cerrado en el dominio discreto utilizando la función de transferencia obtenida previamente y las herramientas de simulación de MATLAB. La Figura 42 muestra el diagrama de bloques empleado para simular el comportamiento del sistema. Es importante señalar que dicha función de transferencia representa tanto la planta como el sistema de medición, ya que modela la relación entre la señal PWM enviada a los motores y la velocidad angular registrada por el sensor.

Además, dado que el Rover utiliza tres motores por lado controlados por una única señal, y que el objetivo es generar un movimiento rectilíneo, se asume que todos los motores se comportan de forma similar y responden de manera equivalente al control. Por esta razón, la simulación se realiza considerando un solo motor como representación del sistema, lo cual constituye una simplificación razonable para este tipo de análisis.

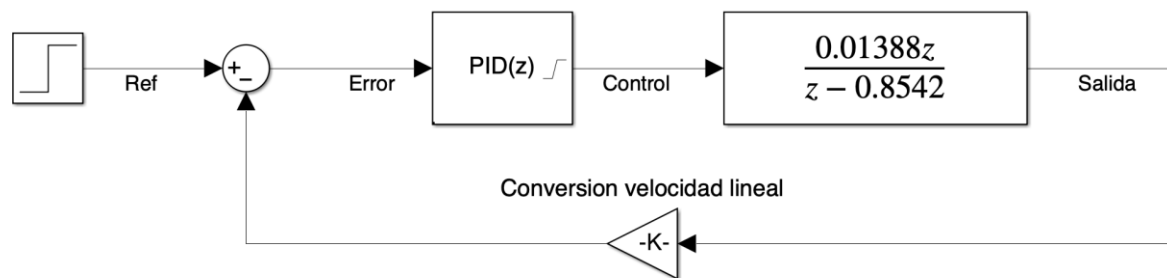


Figura 42. Diagrama de bloques del lazo de control propuesto para la velocidad lineal del Rover.

Aunque la función de transferencia identificada representa la dinámica entre la señal PWM de entrada y la velocidad angular de salida, el objetivo del sistema de control es regular la velocidad lineal del chasis del Rover. Por esta razón, la señal de referencia se expresa en unidades de cm/s, y la salida del sistema se multiplica por el radio de la rueda para convertirla también a cm/s antes de realizar el cálculo del error. De esta forma, tanto la señal de referencia como la de retroalimentación están expresadas en las mismas unidades físicas, lo cual es indispensable para un control coherente.

Cabe destacar que, alternativamente, también habría sido posible diseñar el controlador utilizando velocidades angulares, realizando la conversión de la referencia a rad/s en lugar de convertir la salida. Esta elección no afecta el comportamiento dinámico del sistema ni la calidad del control obtenido. La única diferencia entre ambas opciones es que los valores de las constantes del PID varían proporcionalmente al cambio de unidades, sin que esto represente ventajas funcionales.

Las constantes del controlador PID se determinaron de manera empírica mediante un proceso iterativo de ajuste sobre la respuesta simulada del sistema, buscando un compromiso adecuado entre las métricas de desempeño a evaluar. Los parámetros obtenidos son los siguientes:

$$Kp = 5.17 \quad (5.6)$$

$$Ki = 19.71 \quad (5.7)$$

$$Kd = -0.49 \quad (5.8)$$

$$N = 6.05 \quad (5.9)$$

La configuración final fue seleccionada tras múltiples pruebas, siendo la que ofreció la mejor respuesta. Esta se muestra en la Figura 43 donde se evidencia una rápida convergencia hacia el valor de referencia, con un sobreimpulso bajo y sin oscilaciones sostenidas.

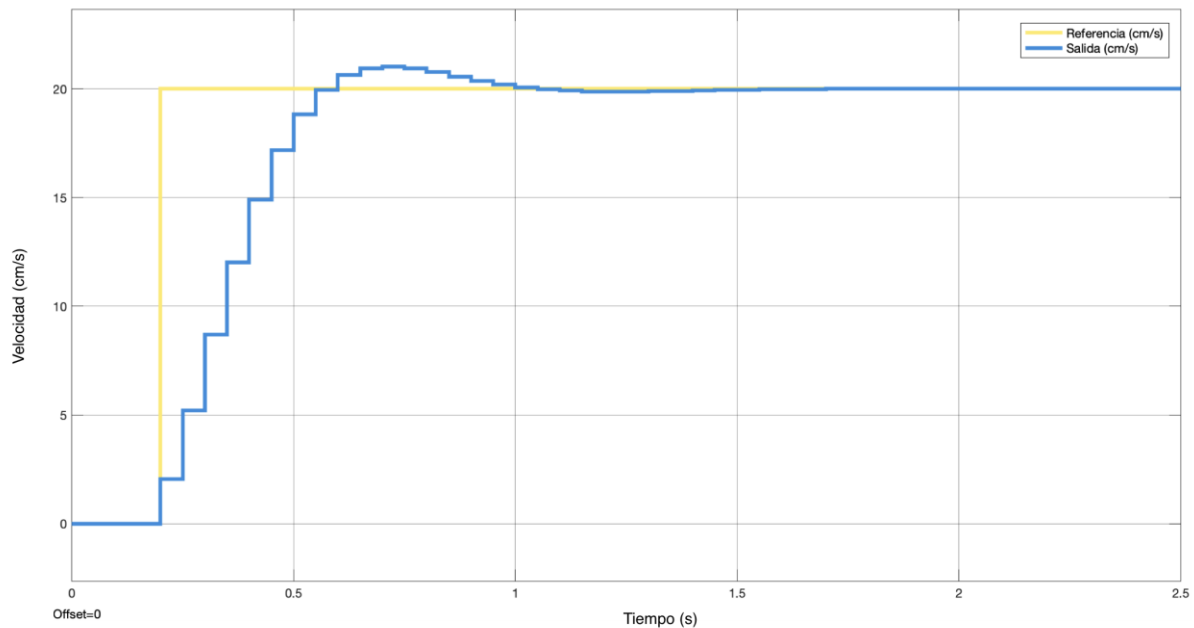


Figura 43. Respuesta simulada del sistema ante una entrada escalón de velocidad lineal.

Adicionalmente, se configuró el controlador con saturación de salida, estableciendo límites mínimos y máximos de operación para la señal PWM, con el fin de proteger los motores y evitar esfuerzos innecesarios. Para prevenir la acumulación excesiva del término integral durante periodos de saturación, se aplicó la técnica de *clamping* (*anti-windup*). Asimismo, se incorporó un filtro en el término derivativo para atenuar el impacto del ruido en la señal de velocidad medida, característica especialmente importante considerando la resolución limitada del sistema de sensado, para este filtro se utilizó un valor $N = 6.05$.

La Figura 44 presenta el diagrama de lugar de las raíces del sistema con el controlador PID implementado. Este análisis permite evaluar la estabilidad y el desempeño dinámico del sistema al variar la ganancia. Como se observa, con la ganancia seleccionada las trayectorias de los polos en lazo cerrado permanecen dentro del círculo unitario, lo que indica la estabilidad del sistema. Además, los polos migran hacia posiciones más cercanas al origen conforme aumenta la ganancia, lo cual se traduce en una mayor rapidez de respuesta. Si bien las trayectorias presentan una ligera componente imaginaria, esta es limitada, lo que indica un comportamiento ligeramente subamortiguado y, por tanto, con oscilaciones controladas alrededor del punto de operación.

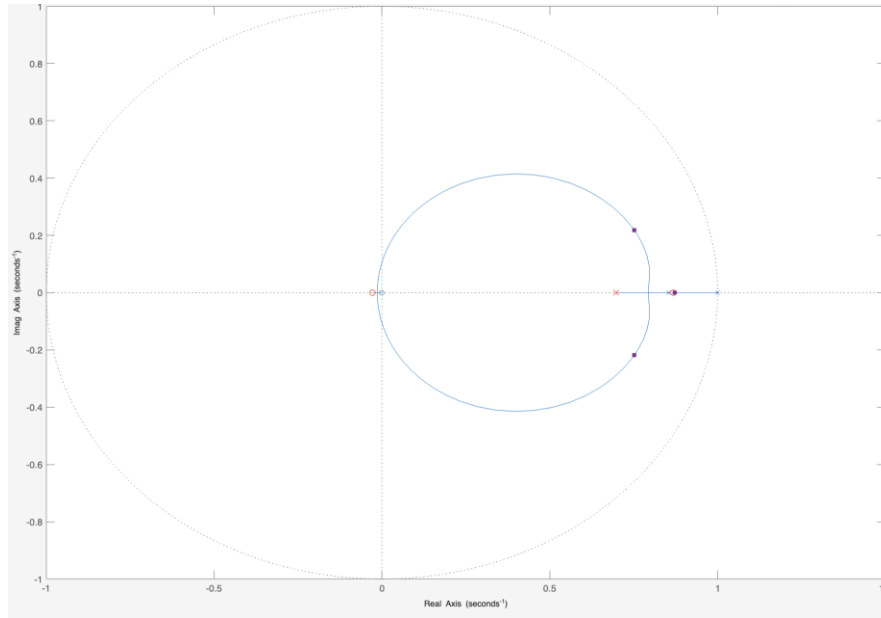


Figura 44. Lugar de las raíces del sistema.

Aunque la simulación permitió definir una configuración inicial para el controlador PID, fue necesario realizar pruebas experimentales con el Rover para ajustar su desempeño en condiciones reales. El modelo utilizado solo representa una aproximación del sistema, con un ajuste del 81.68 %, lo cual es aceptable considerando las limitaciones del sistema de sensado, pero está por debajo de lo ideal. Durante las pruebas, se probaron distintas combinaciones de parámetros, como aumentar el término derivativo o disminuir el integral, buscando una mejor respuesta. Sin embargo, los mejores resultados se obtuvieron con los mismos valores calculados en la simulación. Si bien la respuesta del sistema real no fue idéntica a la simulada, esta configuración ofreció el mejor equilibrio entre rapidez, estabilidad y comportamiento general.

Es importante señalar que el ajuste del controlador en condiciones reales se vio limitado por las capacidades del microcontrolador empleado. A diferencia de otros desarrollos donde es posible aumentar la resolución del PWM para facilitar la sintonización del controlador, en este proyecto no se pudo superar la resolución nativa de 8 bits. Además, por diseño del fabricante, los pines con funcionalidad PWM por hardware ya están reservados para otras funciones en el Rover y la señal de control se aplica mediante la librería SoftPWM a los pines 2, 3, 4 y 5. Esta librería solo permite valores enteros de PWM, lo que restringe aún más la precisión con la que puede aplicarse la acción de control y complica su ajuste fino en pruebas experimentales.

La Figura 45 muestra la respuesta obtenida experimentalmente, en la que se evidencia un sobreimpulso mayor al esperado y una oscilación persistente alrededor del punto de referencia. Sin embargo, esta oscilación se mantiene dentro de un margen aproximado de $\pm 10\%$ del valor objetivo, por lo que el sistema puede considerarse estable. Este comportamiento puede atribuirse al retardo en las lecturas del sensor, que, al no ser lo suficientemente rápidas, provoca que el controlador actúe con información desfasada, generando una respuesta oscilatoria alrededor del punto de referencia sin llegar a converger exactamente a un valor fijo.

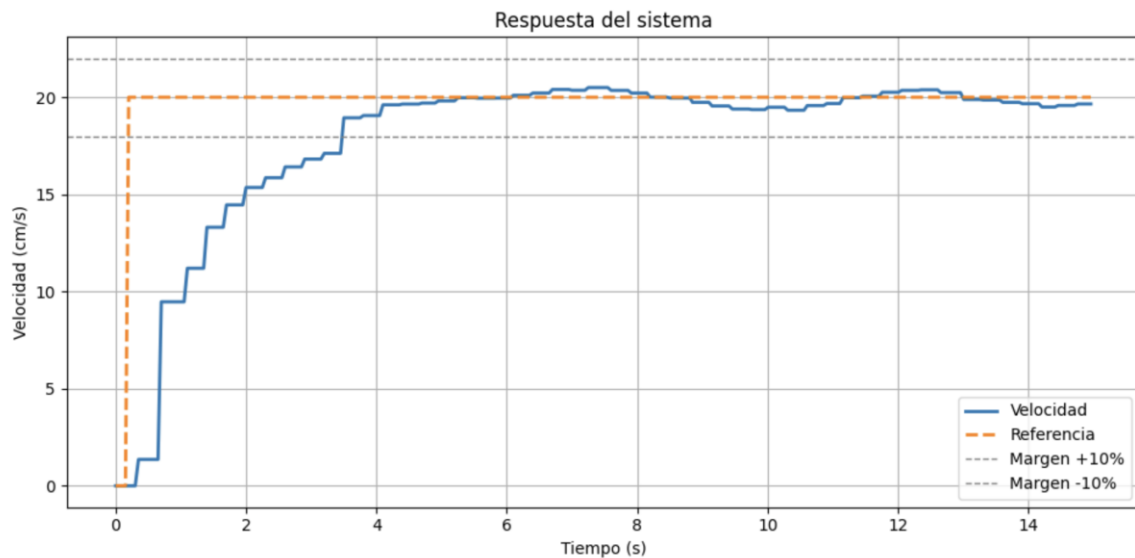


Figura 45. Respuesta del controlador implementado en el Rover.

En la Tabla 8 se muestran los parámetros obtenidos, si bien se acercan mucho a los límites marginales establecidos, se encuentra dentro del rango permitido.

Tabla 8. Parámetros del controlador PID.

Parámetro	Resultado
Tiempo de respuesta	2.80 s
Tiempo de estabilización	3.50 s
Sobreimpulso	2.49 %

En resumen, la implementación y ajuste del controlador PID permitió obtener un desempeño funcional básico, suficiente para validar el enfoque propuesto. Sin embargo, la respuesta del

sistema dista de ser óptima, con oscilaciones persistentes y una precisión limitada, lo que refleja tanto las limitaciones del modelo como del sistema de sensado utilizado. Estos resultados ofrecen una base inicial que deberá ser mejorada en futuras iteraciones.

5.8 Implementación en el Rover

Para implementar el controlador PID en el Rover, se desarrolló un programa en Arduino que integra las funciones de control de velocidad, adquisición de datos y generación de señales PWM. El código está estructurado para ejecutar en tiempo real la lógica del controlador, leer el sensor Hall, registrar datos y aplicar la señal de control a los motores.

El sistema utiliza los pines digitales 2, 3, 4 y 5 del Arduino UNO R3 para controlar los motores, mediante la librería SoftPWM. Esta fue necesaria ya que los pines con funcionalidad PWM por hardware están ocupados por otras funciones predefinidas en la placa SunFounder Galaxy RVR. SoftPWM permite generar señales PWM en pines digitales, aunque solo admite valores enteros, lo cual limita la resolución de control. Además, la resolución máxima del PWM en este microcontrolador es de 8 bits, lo cual restringe aún más la fineza con la que puede ajustarse la señal de control.

El controlador PID se ejecuta con un periodo de muestreo de 50 ms. En cada ciclo de control, se calcula el error entre la velocidad de referencia, establecida en cm/s, y la velocidad medida. Esta última se obtiene a partir del sensor Hall, que detecta pulsos generados por un imán adherido a la rueda. El código calcula la velocidad angular a partir del tiempo entre pulsos y luego la convierte en velocidad lineal usando el radio de la rueda. Para mejorar la calidad de la señal medida, se aplica un filtro de media móvil sobre la velocidad angular antes de calcular la velocidad lineal final.

La salida del controlador es una señal PWM saturada entre un valor mínimo y máximo de 30 y 60, que se aplica directamente a los motores. Estos límites se definieron por razones prácticas y de seguridad: con valores por debajo del mínimo establecido, el Rover no logra iniciar el movimiento; mientras que con valores por encima del máximo, se observó un hundimiento del sistema al iniciar la marcha sobre superficies arenosas. Este comportamiento no es contemplado por el modelo utilizado, ya que no incorpora fenómenos complejos de interacción entre las ruedas y el terreno. Por esta razón, se impuso un límite superior a la señal de control, con el fin

de evitar este tipo de efectos no deseados. Cabe destacar que los casos de uso definidos para el Rover implican velocidades muy bajas, propias de su aplicación, por lo que esta limitación no compromete el cumplimiento de los objetivos funcionales del sistema.

Se comprobó utilizando el control remoto del fabricante que, ante la misma entrada, el lado izquierdo se mueve ligeramente más rápido. Esto puede ser causado por diferencias en los motores o incluso en el ensamblaje del Rover. Para mitigar esta leve diferencia entre lados del Rover, se introdujo una corrección del 10 % en el motor derecho.

Como se mencionó anteriormente, los rovers de exploración planetaria en aplicaciones reales operan mediante secuencias predefinidas, lo que permite una ejecución controlada de cada fase de la misión. En coherencia con este enfoque, el sistema desarrollado se implementó como una máquina de estados compuesta por tres fases: una inicial de espera, una fase activa durante la cual se aplica una entrada tipo escalón correspondiente al valor de referencia, y una fase final de reposo. Durante la fase activa, se ejecutan las rutinas principales del sistema, que incluyen el control de velocidad mediante PID, la lectura del sensor Hall para el cálculo de la velocidad, y el registro de los datos obtenidos en una tarjeta microSD mediante el módulo OpenLog, conectado al microcontrolador por medio de la librería SoftwareSerial.

Las funciones clave del código implementado son las siguientes:

- SetMotors(): aplica la señal PWM a los motores.
- SpeedControl(): ejecuta el controlador PID para obtener la señal de control.
- GetAngVel(): calcula RPM, rad/s y cm/s a partir de pulsos, utiliza filtro de media móvil.
- MuestreoDataLog(): registra datos en OpenLog a intervalo fijo.
- DetectarFlancoSubida(): identifica eventos de pulso del sensor Hall.
- CheckZeroVelocity(): detecta si la velocidad de salida vuelve a cero.

5.9 Validación del mapeo a agentes del sistema

Para validar teóricamente que los bloques funcionales del sistema de control PID discreto, incluyendo la función de transferencia discreta de primer orden que representa tanto la planta o

motor como el sensado, pueden ser mapeados a una arquitectura multiagente, se consideran criterios definidos en trabajos reconocidos como [36] y [37].

En primer lugar, cada bloque del modelo cumple con el criterio de encapsulamiento funcional, pues tiene una tarea clara y específica: generación de referencia, cálculo del error, control PID, y transformación entre unidades (conversión de velocidad lineal). Esta separación funcional asegura la independencia lógica necesaria para definir un agente.

En segundo lugar, estos bloques poseen autonomía estructural, dado que cada bloque ejecuta sus tareas sin necesidad de acceder directamente a estados internos de otros. Por ejemplo, el controlador PID calcula la acción de control basada únicamente en la señal de error recibida, sin depender de detalles internos específicos de la planta.

Además, cada bloque muestra clara capacidad de interacción, ya que la comunicación entre ellos ocurre mediante interfaces claramente definidas como entradas y salidas, equivalentes conceptualmente a mensajes intercambiados en un sistema multiagente. Por ejemplo, el bloque de planta, representado por la función de transferencia discreta, intercambia información con el controlador a través de señales de control y retroalimentación de velocidad medida.

En cuanto a los estados internos identificables, el controlador PID posee estados internos evidentes, como el acumulador de error integral y las diferencias de error para la acción derivativa. Asimismo, la planta presenta estados internos asociados a su dinámica discreta representada por la función de transferencia, como la velocidad actual medida y los efectos retardados del PWM sobre el motor.

También, todos estos bloques individualmente cumplen con la condición de posibilidad de desacoplamiento. Cada bloque puede abstraerse y migrarse conceptualmente hacia otros entornos físicos o computacionales, como por ejemplo distintos microcontroladores o plataformas embebidas, sin perder funcionalidad esencial. Este desacoplamiento funcional es crucial para facilitar una eventual implementación en una arquitectura distribuida.

Además de considerar estos criterios teóricos, se plantea cómo podría implementarse el sistema en una arquitectura multiagente haciendo uso de la metodología desarrollada en [1]. De acuerdo a la hipótesis, con la metodología mostrada en la Figura 46 es posible mapear cualquier sistema a una arquitectura de agentes.

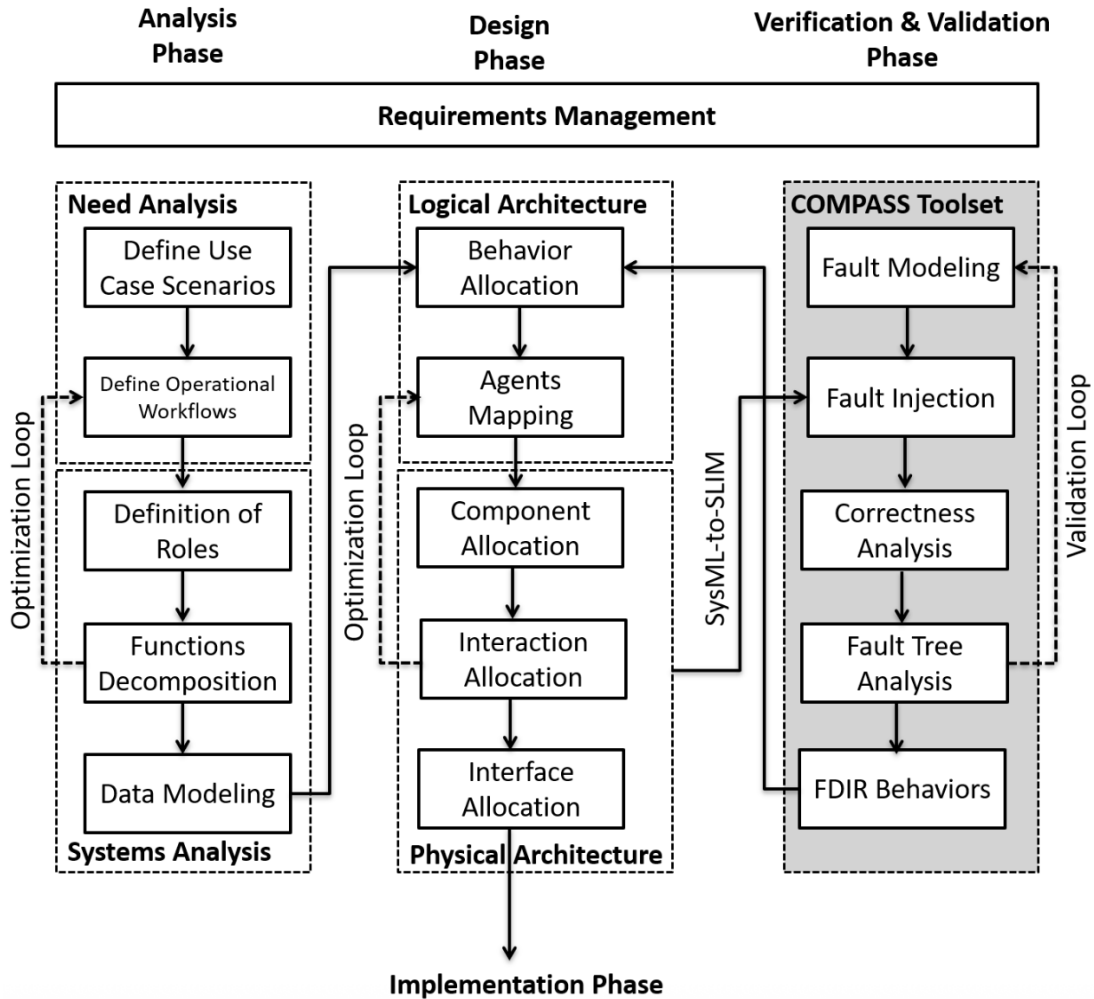


Figura 46. Metodología para mapeo a agentes. Fuente: [1]

La primera fase de análisis consiste en definir el flujo operacional del Rover, para esto se utilizará el código mencionado en la sección anterior de este documento. Según la metodología, normalmente se definen los roles y luego se descomponen en funciones. Sin embargo, en este caso se realiza el proceso inverso debido que ya se cuenta con las funciones del código y estas se pueden agrupar y encapsularse en distintos agentes. Para este sistema se propone definir tres roles: medición, control y actuación. A continuación, se muestran las actividades o funciones de cada uno en el código implementado:

- Medición
 - DetectarFlancoSubida()
 - GetAngVel()

- CheckZeroVelocity()
- MuestreoDataLog
- Control
 - SpeedControl():
- Actuación
 - SetMotors

A partir de este análisis de definición de roles y descomposición de funciones, estos conjuntos se encapsulan en agentes. En la Figura 47 se definen estos agentes y las señales de comunicación entre ellos. Cabe destacar que este es un sistema simple sin muchos componentes, conforme aumenta la complejidad del sistema también lo hacen las diferentes señales de comunicación entre agentes.

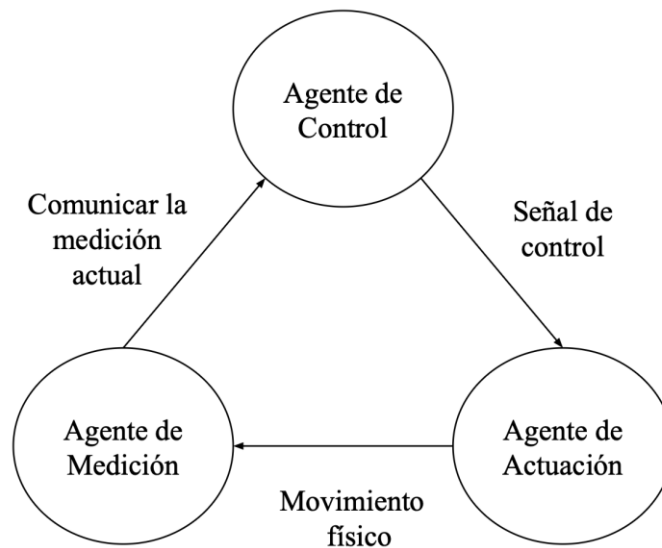


Figura 47. Arquitectura multiagente.

En resumen, el sistema de control de velocidad de este proyecto se compone de bloques funcionales clásicos: generación de referencia, medición de velocidad, cálculo de error, controlador PI o PID, y actuador. Cada uno de estos bloques cumple funciones específicas y puede ser abstraído como un agente software o físico, siguiendo las definiciones antes mencionadas [36] [37]. Adicionalmente, se plantea una posible arquitectura multiagente a partir de las funciones desarrolladas para el código de validación haciendo uso de la metodología utilizada en el proyecto ELANav [1] [2]. Por tanto, el sistema ha sido diseñado desde su etapa

inicial con modularidad suficiente para asegurar su futura mapeabilidad a una arquitectura multiagente, aunque dicha arquitectura no se implementará en el presente trabajo.

Capítulo 6. Resultados y análisis

6.1 Diseño de banco de pruebas

En esta sección se describe el diseño y la elaboración del banco de pruebas utilizado para la validación del sistema de control de velocidad propuesto. El diseño del entorno de pruebas se definió en función de tres casos de uso representativos: descenso por rampa, desplazamiento sobre terreno plano sin obstáculos y desplazamiento sobre terreno con obstáculos.

6.1.1 Diseño de rampa

El caso del descenso por rampa se eligió debido a que el despliegue exitoso del vehículo en la superficie extraterrestre es un elemento crítico en las misiones de exploración planetaria. Existen métodos de despliegue aéreos para llevar a cabo este objetivo, sin embargo, estos utilizan paracaídas que requieren una atmósfera para desacelerar la velocidad de descenso. Estos métodos no pueden emplearse de la misma forma en la Luna donde existe un vacío virtual, por lo que el uso de rampas ha sido una constante a través de la historia [38].

Los rovers lunares Lunokhod de la Unión Soviética, Yutu de la Administración Espacial Nacional China y Pragyan de la Organización de Investigación Espacial de la India utilizaron sistemas de rampas integradas en sus módulos de aterrizaje para descender de forma autónoma o semiautónoma. Estos mecanismos permitieron un despliegue controlado desde la plataforma del módulo hasta la superficie lunar. En contraste, el *Lunar Roving Vehicle*, desarrollado por la NASA y utilizado en las misiones Apolo 15, 16 y 17, fue el único Rover que no requirió una rampa, ya que fue ensamblado y desplegado manualmente por los astronautas directamente sobre la superficie de la Luna, después del alunizaje [39].

Para el diseño de la rampa se consideraron dos factores principales: la inclinación y la capacidad estructural para soportar el peso del Rover. En cuanto a la inclinación, se tomaron como referencia el diseño exitoso del sistema de despliegue del Chandrayaan-3 y un estudio desarrollado por la Agencia Espacial Japonesa JAXA.

En el caso del Chandrayaan-3, uno de los requisitos funcionales establecidos fue “facilitar el despliegue controlado del rover sobre un terreno lunar con una pendiente de hasta $\pm 15^\circ$ ” [39]. El Pragyan Rover, montado sobre una rampa, estaba equipado con un sistema de suspensión tipo rocker-bogie, con un ángulo de estabilidad estática de 50° . A partir de esto, se determinó que el ángulo θ de la rampa debía ser inferior a 35° para asegurar la estabilidad del rover en condiciones de pendiente. Finalmente, se seleccionó un ángulo de 30° como una opción adecuada [39].

Por otro lado, el estudio de JAXA analizó diversos tipos de rampas y concluyó que, considerando la travesabilidad de su rover, una inclinación de 20° era apropiada para permitir el descenso del vehículo [38].

Con base en estas referencias, se consideró en este trabajo la posibilidad de encontrar pendientes de hasta $\pm 15^\circ$ en la superficie lunar. Para evaluar la capacidad del Rover ante estas condiciones, se realizaron pruebas experimentales utilizando una superficie plana de plástico, la cual fue colocada progresivamente a diferentes ángulos de inclinación. El objetivo de estas pruebas fue identificar el punto a partir del cual el Rover comenzaba a presentar deslizamientos o riesgo de vuelco. Se observó que para inclinaciones menores a 40° no se manifestaban fenómenos de desestabilización. Considerando un margen de $\pm 15^\circ$ por las condiciones del terreno lunar, se definió una inclinación de rampa de 20° , garantizando que no se superen los 40° totales que comprometerían la estabilidad del sistema. Además, este valor se encuentra dentro del rango de 20° a 30° , utilizado en los casos de estudio mencionados, cuyas soluciones han sido validadas y demostradas en aplicaciones reales por agencias espaciales de reconocimiento internacional.

Debido al tamaño del espacio físico con el que se cuenta en el laboratorio, se conoce que la extensión de la rampa no debe exceder 1.5 metros. Tomando esta información y los ángulos definidos para la inclinación de la rampa, se obtuvieron los valores de los lados c y b .

$$c = \frac{a}{\sin(A)} = \frac{150}{\sin(70^\circ)} \approx 159.6 \text{ cm} \quad (6.1)$$

$$b = c \cdot \sin(B) = 159.6 \cdot \sin(20^\circ) \approx 54.6 \text{ cm} \quad (6.2)$$

El resultado se observa en la Figura 48, estas son las dimensiones para la construcción de la rampa.

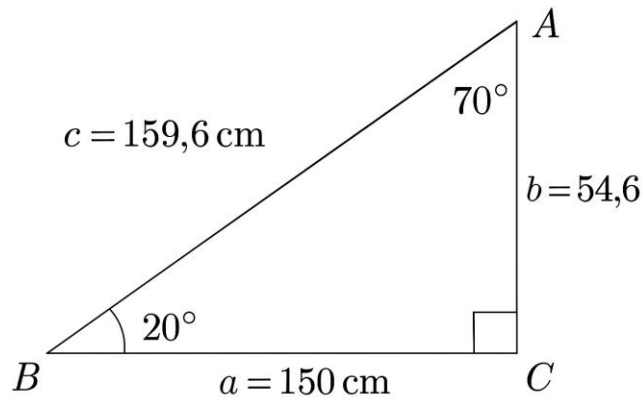


Figura 48. Geometría de la rampa de pruebas.

El segundo factor a considerar en el diseño de la rampa es la capacidad estructural para soportar el peso del Rover que es aproximadamente 3.1 libras o 1.41 kilogramos.

Para dimensionar adecuadamente la sección transversal de la rampa, en primer lugar, se calculó el momento flector máximo generado por una carga puntual en el centro de una viga simplemente apoyada, considerando el peso del objeto que la rampa debe soportar.

$$F = m \cdot g = 1.41 \cdot 9.81 \approx 13.8\text{N} \quad (6.3)$$

$$M_{\text{máx}} = \frac{F \cdot L}{4} = \frac{13.8 \cdot 1.596}{4} \approx 5.5 \text{ N} \cdot \text{m} \quad (6.4)$$

Luego, se determinó el módulo de sección necesario a partir del momento flector y el esfuerzo admisible del material (madera en este caso, para la que se asume un esfuerzo admisible de 8 MPa).

$$S = \frac{M_{\text{máx}}}{\sigma_{\text{adm}}} = \frac{5.5}{8 \times 10^6} = 6.875 \times 10^{-7} \text{ m}^3 \quad (6.5)$$

Finalmente, se empleó la fórmula correspondiente a una sección rectangular para despejar la dimensión requerida de la altura, asumiendo un valor para la base, con el fin de garantizar la

resistencia estructural bajo las condiciones establecidas. El valor para la base se define como 45 centímetros que corresponde a las dimensiones del Rover más 9 cm adicionales a cada lado.

$$S = \frac{bh^2}{6} = \frac{0.45 h^2}{6} \quad (6.6)$$

$$h^2 = \frac{6 \cdot 6.875 \times 10^{-7}}{0.45} = 9.167 \times 10^{-6} \quad (6.7)$$

$$h = \sqrt{9.167 \times 10^{-6}} \approx 0.00303 \text{ m} = 3.03 \text{ mm} \quad (6.8)$$

Estos cálculos indican que una altura de 3.03 mm sería teóricamente suficiente para soportar la carga de 1.41 kg sin exceder el esfuerzo admisible del material. Sin embargo, se recomienda aumentar el grosor de la rampa a varios centímetros por razones prácticas, como la facilidad de manufactura, la disponibilidad comercial de materiales, y proporcionar una mayor rigidez estructural para evitar vibraciones o deformaciones excesivas durante el uso. Además, un espesor mayor ofrece un margen de seguridad adicional frente a posibles cargas dinámicas o impactos accidentales.

Adicionalmente, la rampa cuenta con una base de la que parte el Rover. Las dimensiones de esta base se muestran en la Figura 49, se conforma por un área cuadrada que tiene una longitud y un ancho de 45 cm y una altura de 54.6 cm, lo que le da forma de prisma rectangular. Esta estructura sirve como plataforma de despliegue para el Rover, proporcionando una superficie estable desde la cual puede iniciar su desplazamiento.

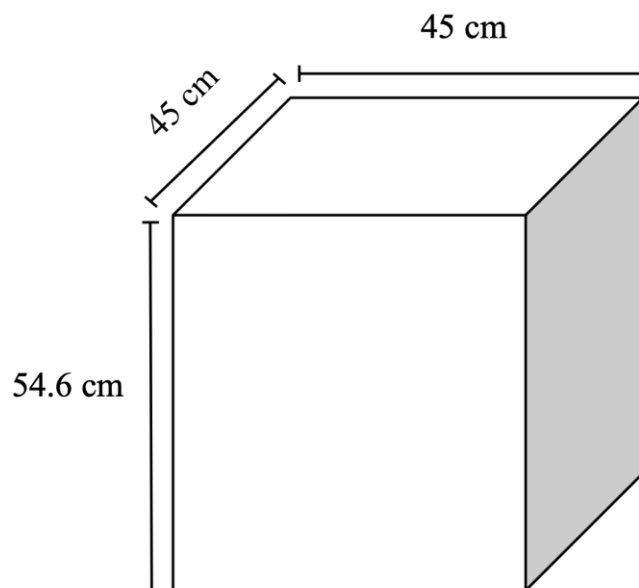


Figura 49. Base de la rampa.

Debido a limitaciones logísticas surgidas durante la coordinación entre el proveedor de la rampa y el laboratorio, no fue posible contar con la estructura diseñada en el plazo previsto. Como alternativa, se implementó una rampa provisional construida con materiales disponibles, a fin de no afectar la ejecución del caso de prueba correspondiente. La estructura utilizada consistió en una tabla de madera de 149 cm de largo y 30 cm de ancho, elevada a una altura de 36 cm en uno de sus extremos. Esta configuración permitió simular una pendiente con un ángulo de inclinación aproximado de 16.1° , suficiente para representar una condición de descenso moderado y evaluar el desempeño del sistema de control bajo dicha perturbación. En la Figura 50 se muestra la rampa utilizada.

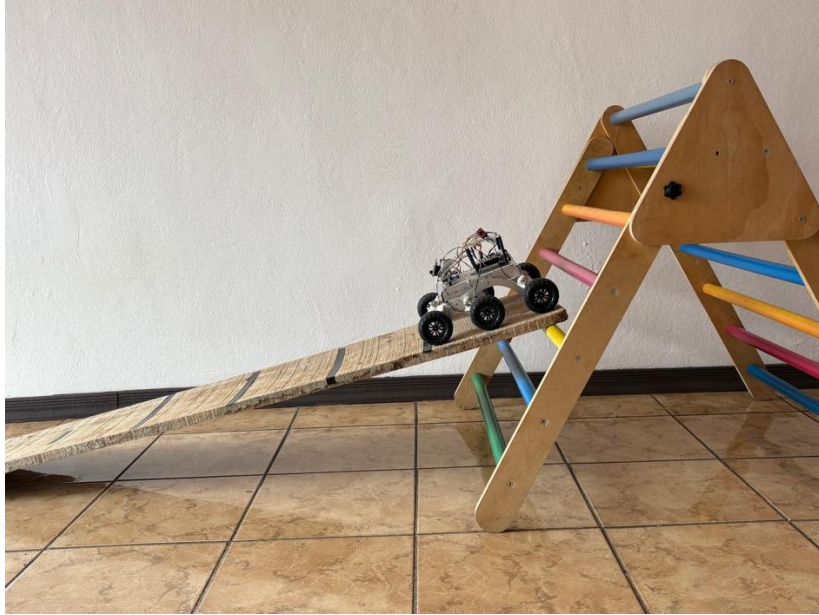


Figura 50. Vista lateral de la rampa utilizada para la validación.

Es importante señalar que, aunque la tabla utilizada para las pruebas tiene una longitud total de 149 cm, el Rover empleado mide aproximadamente 30 cm de largo. Esto implica que el espacio útil disponible para el desplazamiento controlado es de 119 cm. Además, dado que el borde final de la tabla no cuenta con una salida progresiva ni una estructura de transición (es decir, termina abruptamente como una grada), no se considera seguro ni viable utilizar la totalidad del tramo para mediciones. Por esta razón, se definieron 100 cm efectivos para las pruebas, los cuales fueron divididos en cinco trayectos de 20 cm cada uno, marcados con cintas adhesivas colocadas a intervalos regulares a lo largo de la superficie.

6.1.2 Diseño de superficie sin obstáculos

Para esta superficie se utilizó como base alfombra de césped artificial, a la que se le colocaron líneas de cinta negra equidistantes a 20 cm de inicio a fin para posteriormente poder medir la velocidad del Rover de forma externa mediante vídeo. En total, se definieron 13 trayectos de 20 cm cada uno, abarcando una sección de 260 cm dentro de la superficie disponible para las

pruebas. Esto permite un tiempo de prueba máximo de aproximadamente 17,3 segundos y 13 segundos, a velocidades de 15 cm/s y 20 cm/s respectivamente.



Figura 51. Alfombra de grama artificial utilizada.

Se utilizaron dos alfombras como la mostrada en la Figura 51 para un total de 4 metros de distancia disponible. Además de la superficie descrita, se empleó también un terreno de arena fina con el objetivo de ampliar la validez de las pruebas en un segundo tipo de superficie sin obstáculos. Esta superficie de prueba se elaboró sobre la misma alfombra de césped artificial utilizada previamente, la cual funcionó como base y contenedor para sostener el nuevo material. Sobre esta se colocó una capa uniforme de arena gruesa del tipo comúnmente utilizada en áreas de juego infantil, conocida comercialmente como *play sand*, como se muestra en la Figura 52. Este tipo de arena, de textura fina y relativamente suelta, fue seleccionada por sus características físicas similares a las de terrenos que podrían encontrarse en escenarios lunares o planetarios. La franja de arena colocada tuvo un ancho aproximado de 40 cm, suficiente para cubrir el paso del Rover, cuyo ancho total es de 30 cm, asegurando así que todas las ruedas estuvieran siempre en contacto con esta superficie durante la prueba.

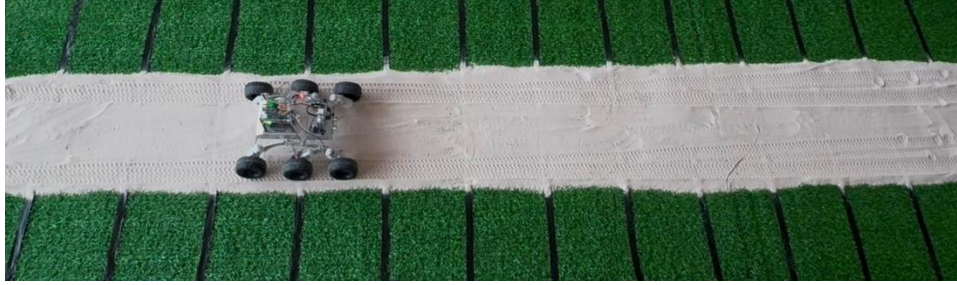


Figura 52. Rover sobre superficie con arena.

Al incorporar ambos escenarios, una superficie regular y otra de mayor complejidad mecánica, se buscó obtener un conjunto de resultados experimentales más representativo del comportamiento del sistema bajo diferentes condiciones operativas.

6.1.3 Diseño de superficie con obstáculos

Aunque la superficie de la Luna está compuesta en gran medida por regolito, una mezcla fina de polvo y fragmentos minerales que se asemeja a la arena, también existen regiones caracterizadas por terrenos rocosos e irregulares. Dado que estos escenarios representan condiciones reales de desplazamiento para rovers en este tipo de aplicación, se diseñó una superficie de prueba que incorporara este tipo de dificultad. Para ello, se utilizaron piedras de distintos tamaños distribuidas de forma aleatoria sobre la misma base de césped artificial empleada en las demás superficies de validación, como se muestra en la Figura 53. En este caso, se definieron 9 trayectos de 20 cm cada uno, cubriendo un tramo total de 180 cm utilizados para el análisis del desempeño del sistema.



Figura 53. Rover sobre superficie con obstáculos distribuidos sobre césped artificial.

El objetivo de este terreno fue introducir perturbaciones físicas que afectaran el desplazamiento del Rover y, con ello, poner a prueba el desempeño del sistema de control de velocidad en condiciones de mayor complejidad. Esta configuración permitió evaluar la capacidad del controlador para mantener la referencia deseada a pesar de las interferencias provocadas por las irregularidades del terreno.

6.2 Pruebas de validación

Con el objetivo de validar el desempeño del sistema de control de velocidad propuesto, este se implementó sobre el prototipo físico del Rover. La validación experimental se estructuró en torno a los tres casos de uso: desplazamiento en rampa, desplazamiento sobre superficie sin obstáculos y desplazamiento sobre superficie con obstáculos. Para cada uno de estos casos se definieron velocidades de referencia distintas: 10 cm/s en rampa, 20 cm/s en superficie libre y 15 cm/s en presencia de obstáculos. A fin de verificar el cumplimiento de las especificaciones establecidas, se midieron las métricas de desempeño: error de medición de velocidad, tiempo de respuesta, sobreimpulso y tiempo de estabilización. En esta sección se presenta un resumen de los resultados obtenidos para cada métrica.

6.2.1 Casos de prueba

Las pruebas realizadas se dividen en cuatro casos de uso, con las características que se detallan en la Tabla 9.

Tabla 9. Características de los casos de prueba utilizados.

Casos de prueba				
Características	Caso 1	Caso 2	Caso 3	Caso 4
Superficie	Césped	Césped con obstáculos	Arena	Rampa
Velocidad (cm/s)	20	15	20	10
Duración (s)	15	15	15	11

Para cada caso se realizaron 30 repeticiones y se utilizó una frecuencia de muestreo y de control de 50 ms. En las siguientes secciones se analiza el desempeño para cada métrica.

A continuación, se presentan las curvas de respuesta del sistema correspondientes a cada uno de los cuatro casos de prueba definidos, con el fin de visualizar su comportamiento ante la aplicación de la señal de referencia.

Para el caso 1, la respuesta del sistema de la Figura 54 muestra un comportamiento caracterizado por un ascenso rápido al inicio, alcanzando el margen de estabilización de $\pm 10\%$ respecto al valor de referencia en menos de 4 segundos. A partir de ese momento, la señal de velocidad oscila de forma constante alrededor del punto de referencia, lo cual es consistente con una respuesta subamortiguada. El sobreimpulso observado es bajo, lo que indica un buen control del exceso de corrección tras el cambio de referencia. Aunque persisten oscilaciones alrededor del valor deseado, estas se mantienen contenidas dentro del margen definido, lo cual sugiere que el controlador está actuando eficazmente bajo las condiciones físicas del caso de prueba. Este tipo de respuesta es deseable en sistemas donde se prioriza un tiempo de respuesta corto sin sacrificar la estabilidad, y valida la capacidad del sistema para seguir adecuadamente la referencia en una superficie como el césped.

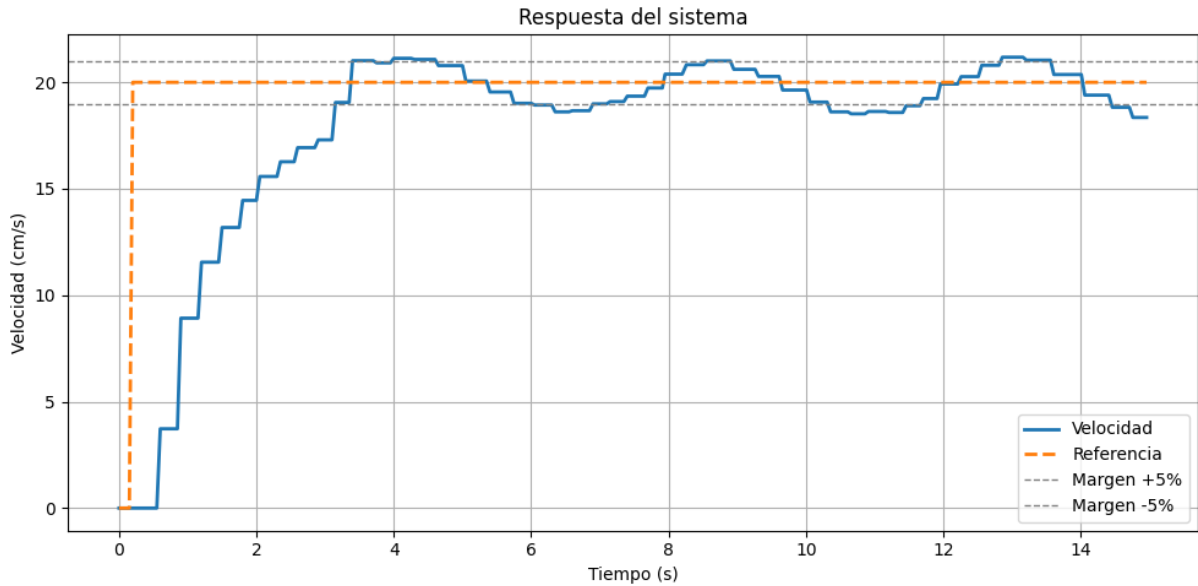


Figura 54. Respuesta caso 1.

La Figura 55 muestra la respuesta del sistema para el caso 2, donde se introducen obstáculos sobre la superficie de césped. En este escenario, se observa un incremento en la magnitud de las oscilaciones respecto al caso anterior, lo cual es coherente con la perturbación adicional que generan los obstáculos sobre el desplazamiento del Rover. Esta condición afecta la capacidad del sistema para estabilizarse dentro del margen objetivo de $\pm 10\%$ en torno al valor de referencia. Tras el análisis de las 30 repeticiones, se determinó que, aunque el sistema logra estabilizarse, lo hace dentro de un margen ligeramente inferior a $\pm 15\%$, por lo que no cumple con los criterios establecidos inicialmente. A pesar de estas oscilaciones más pronunciadas, tanto el tiempo de respuesta como el tiempo de estabilización son comparables a los obtenidos en el caso 1, lo que sugiere que el controlador mantiene una respuesta rápida, aunque con menor precisión bajo condiciones más exigentes.

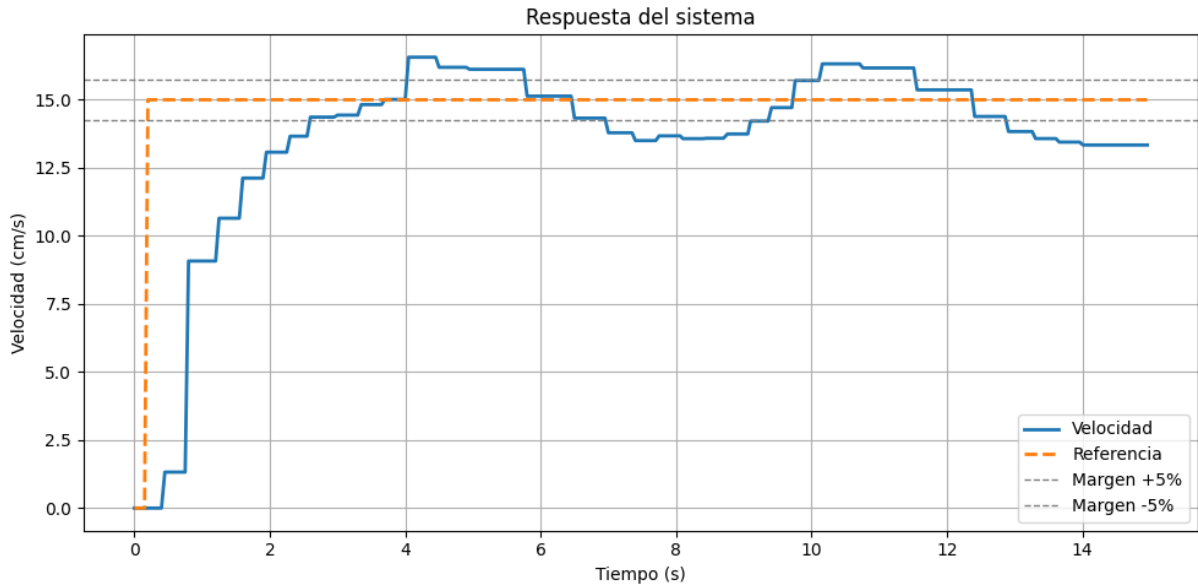


Figura 55. Respuesta caso 2.

La Figura 56 muestra la respuesta del sistema sobre una superficie de arena. En este caso, se observa que la velocidad alcanza el valor de referencia de forma más lenta en comparación con los casos anteriores, evidenciando un mayor tiempo de respuesta. Sin embargo, una vez alcanzada la velocidad objetivo, la señal se mantiene con una oscilación mínima y un error en estado estacionario bajo. La curva presenta el comportamiento más constante y estable entre los cuatro casos de prueba, con oscilaciones muy contenidas.

Este resultado sugiere que las características físicas de la arena, como su capacidad de absorción de vibraciones y mayor resistencia al desplazamiento, actúan como un amortiguador natural, favoreciendo la estabilidad del sistema y reduciendo la sensibilidad a pequeñas perturbaciones. En conjunto, este comportamiento evidencia una interacción positiva entre el entorno físico y el controlador, permitiendo un desempeño más predecible y uniforme.

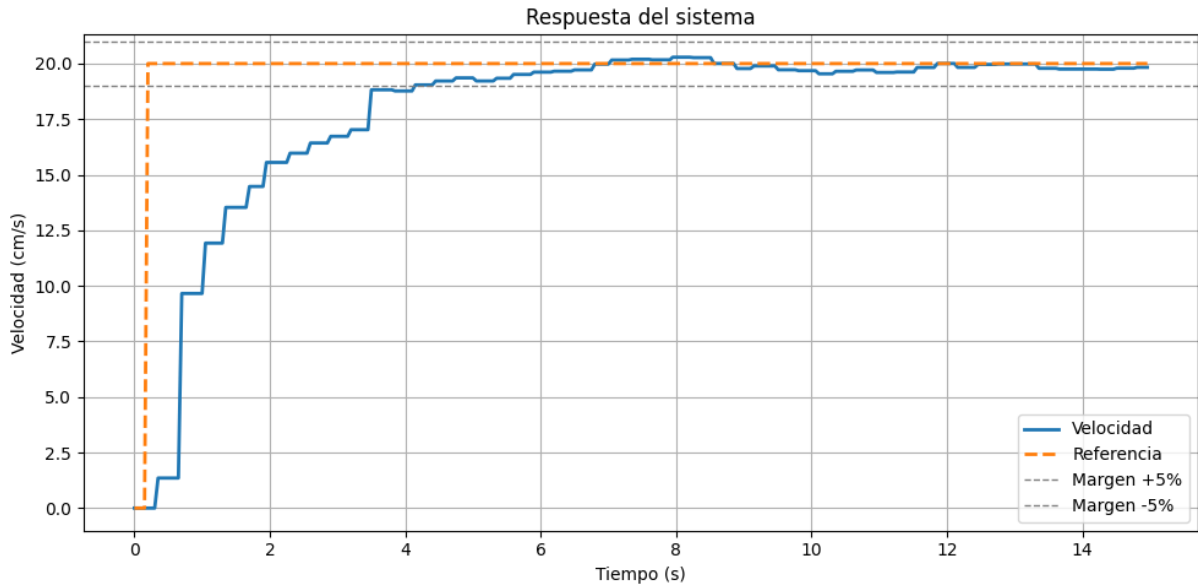


Figura 56. Respuesta caso 3.

La Figura 57 ilustra la respuesta del sistema al iniciar el movimiento desde la parte superior de una rampa en descenso, con una referencia de velocidad de 10 cm/s. Este caso presenta el mayor sobreimpulso de los cuatro escenarios evaluados, lo cual puede atribuirse a la contribución de la fuerza gravitacional y la energía potencial del Rover al inicio del descenso. Como resultado, la velocidad supera de forma considerable el valor de referencia poco después del arranque.

A diferencia de los casos anteriores, el sistema no logra estabilizarse en torno al valor de referencia, y aunque se intentó extender la duración de la prueba para observar el comportamiento en estado estacionario, la longitud de la rampa disponible limitó esta posibilidad. No obstante, como se aprecia en la Figura 57, la rampa es lo suficientemente extensa como para constatar que no se alcanza el objetivo de estabilización ni el tiempo de respuesta requerido.

Adicionalmente, este fue el caso que presentó mayor dispersión en los resultados, con una desviación estándar notablemente superior a la de los demás escenarios. Esto evidencia una sensibilidad elevada del sistema ante las perturbaciones inducidas por el cambio de pendiente y la aceleración adicional provocada por la gravedad, lo que afecta significativamente el desempeño del controlador bajo esta condición.

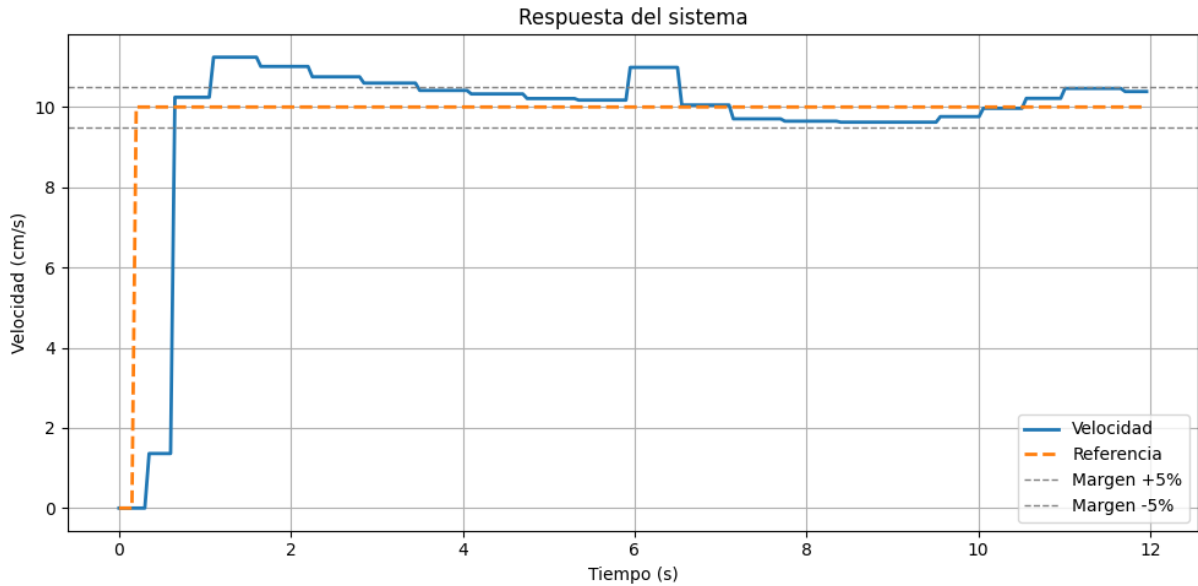


Figura 57. Resposta caso 4.

En conclusión, el análisis comparativo de los cuatro casos de prueba permitió evidenciar cómo las características físicas del entorno influyen significativamente en el desempeño del sistema de control. Mientras que superficies con mayor amortiguación natural, como la arena, favorecen una respuesta más estable y precisa, escenarios con perturbaciones externas, como obstáculos o pendientes descendentes, introducen oscilaciones, sobreimpulso y mayor variabilidad en los resultados. Si bien el controlador logra mantener un comportamiento aceptable en la mayoría de los casos, se identifican limitaciones claras en su capacidad de adaptación frente a condiciones dinámicas más exigentes, especialmente en presencia de aceleraciones adicionales no compensadas por el modelo de control.

6.2.2 Prueba de validación de error de velocidad

Para evaluar el desempeño del sistema de control de velocidad se optó por utilizar el error porcentual absoluto medio (MAPE, por sus siglas en inglés). Esta elección responde a la necesidad de medir no solo la precisión global del sistema, sino también su capacidad para mantener consistentemente la velocidad deseada a lo largo del trayecto. A diferencia del error calculado a partir de la velocidad promedio, el MAPE permite identificar desviaciones

puntuales, ya que cuantifica, en promedio, cuánto se aleja la velocidad real del valor de referencia en cada tramo individual del recorrido.

Para calcular esta métrica, se realizaron registros temporales en el momento en que el Rover alcanzaba una serie de marcas físicas dispuestas a intervalos regulares de veinte centímetros. A partir de estos datos, se determinó la velocidad en cada tramo individual dividiendo la distancia fija entre marcas entre el tiempo transcurrido entre cada par consecutivo de mediciones. Luego, cada una de estas velocidades fue comparada con el valor de referencia establecido, obteniendo así su error porcentual individual. Finalmente, el error medio se calculó como el promedio de todos estos errores porcentuales a lo largo de los tramos. De esta forma se identifican desviaciones puntuales e inconsistencias en el seguimiento de la velocidad, incluso en aquellos casos en que el promedio global de velocidad pueda parecer adecuado. El detalle de estos valores para cada tramo de cada prueba se encuentra en el Apéndice C.

Tabla 10. Resultados del error de velocidad del sistema en distintos casos de uso.

Resultados	Caso 1	Caso 2	Caso 3	Caso 4
Velocidad objetivo (cm/s)	20.00	15.00	20.00	10.00
Velocidad Promedio (cm/s)	18.71	13.92	18.39	10.09
Moda	18.69	13.38	18.60	9.65
Máximo	19.26	14.43	19.07	10.81
Mínimo	17.52	13.35	17.60	9.51
Desviación estándar	0.37	0.32	0.36	0.37
Error Porcentual Absoluto Medio (%)	9.15	19.36	8.35	7.55

Los resultados muestran que, en tres de los cuatro casos de prueba, el sistema logró mantener el error porcentual absoluto medio dentro del rango objetivo establecido, igual o inferior al 10%. En particular, los casos 1, 3 y 4 presentaron valores de 9,15 %, 8,35 % y 7,55 % respectivamente, lo que indica un seguimiento adecuado de la velocidad de referencia bajo condiciones controladas o con amortiguación natural. En contraste, el caso 2, correspondiente a la superficie de césped con obstáculos, superó incluso el umbral marginal de 15 %, con un error de 19,36 %, reflejando la dificultad del sistema para mantener una velocidad constante ante perturbaciones físicas externas.

Además del error medio, los resultados muestran una baja desviación estándar en todos los casos, con valores iguales o inferiores a 0,37 cm/s. Esto indica que el sistema tiene un comportamiento estable y repetible entre pruebas. Incluso en el caso 2, donde el error medio fue alto, la baja dispersión sugiere que la causa no está en fallos aleatorios del sistema, sino en una perturbación externa constante. Esto demuestra que, aunque existen oscilaciones y tiempos de estabilización largos en algunos escenarios, el sistema logra mantener velocidades cercanas al valor objetivo de forma sostenida. En aplicaciones móviles, este comportamiento es clave, ya que la estabilidad en el seguimiento de la consigna es tan importante como la precisión instantánea.

En resumen, los resultados respaldan que el sistema ofrece un desempeño adecuado y consistente para trayectos rectos en distintas condiciones. Si bien hay espacio para mejorar la estabilidad ante perturbaciones, el desempeño actual ya cumple con uno de los objetivos principales del proyecto: controlar la velocidad con errores bajos y consistentes.

6.2.3 Prueba de validación de tiempo de estabilización

El tiempo de estabilización es una métrica clave para evaluar la capacidad del sistema de alcanzar un estado estable luego de un cambio en la señal de entrada. Los resultados obtenidos durante las pruebas se muestran en la Tabla 11. En este contexto, los valores máximos registrados (14,96 s para los casos 1 a 3 y 11,96 s para el caso 4) indican que el sistema no logró estabilizarse dentro del margen definido antes de que finalizara la prueba. Por lo tanto, estos valores deben interpretarse como indicativos de fallo en la estabilización.

Tabla 11. Resultados del tiempo de estabilización del sistema en distintos casos de uso.

Experimento	Tiempo de estabilización			
	Caso 1	Caso 2	Caso 3	Caso 4
1	4.55	14.96	7.10	11.96
2	3.80	14.96	4.55	6.40
3	3.60	14.96	4.20	11.96
4	9.56	14.96	4.00	11.96
5	3.30	14.96	13.46	6.55
6	14.96	14.96	12.71	11.96
7	14.96	14.96	7.20	11.96
8	14.96	14.96	7.20	11.11
9	14.96	14.96	11.86	6.80
10	14.96	14.96	5.50	7.86
11	14.96	14.96	4.15	6.70
12	14.96	14.96	5.10	6.70
13	14.96	14.96	4.25	11.96
14	14.96	14.96	4.15	11.56
15	14.96	14.96	3.45	11.96
16	14.96	14.96	3.60	11.96
17	14.96	14.91	3.45	11.96
18	14.96	14.96	3.50	9.41
19	14.96	14.56	3.40	11.96
20	14.96	14.96	3.50	10.96
21	14.96	14.66	3.55	11.96
22	14.96	14.96	3.60	11.96
23	14.96	14.96	12.61	11.96
24	14.96	14.96	4.70	11.96
25	11.91	14.96	14.86	11.96
26	14.96	14.96	12.26	11.96
27	14.96	14.86	14.96	10.56
28	14.96	14.96	4.30	6.65
29	14.96	14.81	14.96	11.96
30	14.96	14.71	4.60	11.96
Media	13.19	14.92	6.89	10.55
Moda	14.96	14.96	7.20	11.96
Máximo	14.96	14.96	14.96	11.96
Mínimo	3.30	14.56	3.40	6.40
Desv. estándar	3.91	0.10	4.21	2.18

El caso 1 presenta una media de 13,19 s, con 19 de las 30 repeticiones alcanzando el valor máximo de 14,96 s, lo que indica que en la mayoría de los experimentos el sistema no se estabilizó en el tiempo disponible. Aunque algunas repeticiones lograron tiempos inferiores

como el mínimo de 3,30 s, la alta desviación estándar de 3,91 muestra una gran variabilidad en el desempeño, lo cual refleja la dificultad del sistema para mantener la velocidad dentro del valor deseado en este caso.

El caso 2 fue el más crítico donde 29 de las 30 pruebas finalizaron en 14,96 s sin alcanzar la estabilización, y solo una muestra registró un tiempo apenas menor. Esto se refleja en una media de 14,92 s y una desviación estándar muy baja, que simplemente indica una respuesta consistentemente deficiente en esta métrica. Este resultado confirma que la presencia de obstáculos genera perturbaciones que el controlador no logra compensar de forma efectiva.

Aunque se esperaría que la arena, por su amortiguamiento natural, ofreciera mejores condiciones de estabilización, los resultados muestran que el sistema tampoco logró cumplir el criterio en la mayoría de las repeticiones. La media fue de 6,89 s, la más baja entre todos los casos, pero 12 de las 30 pruebas terminaron en 14,96 s, es decir, sin lograr estabilizarse. La desviación estándar es la más alta de todos los casos, indicando un comportamiento muy variable. Algunas repeticiones muestran tiempos notablemente bajos como el mínimo de 3,40 s, lo que sugiere que el entorno es más favorable que en los casos anteriores, pero la inconsistencia sigue siendo un problema.

Para el caso 4, el sistema presenta una media de 10,55 s, con muchas repeticiones alcanzando ese límite, lo que indica nuevamente que no se logra la estabilización en la mayoría de los intentos. La desviación estándar también evidencia una dispersión considerable.

En resumen, bajo este criterio, ninguno de los cuatro escenarios logra cumplir de manera consistente con el objetivo de estabilización. El sistema de control, en su configuración actual, falla sistemáticamente en mantener la salida dentro del margen de referencia durante el tiempo requerido. Estas observaciones refuerzan la necesidad de ajustes tanto en hardware como en el diseño del control, especialmente si se busca operar en entornos con perturbaciones relevantes.

En esta métrica, se definió formalmente este parámetro como el tiempo requerido para que la salida permanezca dentro de un margen de $\pm 5\%$ respecto al valor de referencia, siguiendo la convención más común en el análisis de sistemas de control. Los resultados obtenidos bajo este criterio son los que determinan si el sistema cumple o no con los objetivos de desempeño

planteados. No obstante, para complementar el análisis y comprender mejor el comportamiento estacionario del sistema, se realizó una evaluación adicional considerando márgenes ampliados de $\pm 10\%$ y $\pm 15\%$. La Tabla 12 presenta estos resultados exploratorios que permiten identificar patrones relevantes en la dinámica de cada caso.

Tabla 12. Comparación de tiempos de estabilización a distintos márgenes de tolerancia.

Experimento	Comparación de tiempos de estabilización bajo diferentes márgenes de tolerancia			
	Caso 1 ($\pm 10\%$)	Caso 2 ($\pm 15\%$)	Caso 3 ($\pm 10\%$)	Caso 4 ($\pm 10\%$)
1	3.65	2.00	3.70	7.10
2	3.50	1.95	3.60	6.40
3	3.60	3.65	3.60	0.80
4	3.60	1.90	3.40	1.70
5	3.30	1.95	3.80	2.25
6	3.05	2.85	3.65	0.65
7	3.30	2.75	3.75	2.45
8	2.80	3.30	3.75	6.70
9	3.15	2.80	3.80	6.25
10	2.70	2.05	3.60	7.30
11	3.10	1.95	3.50	6.15
12	2.95	1.95	3.55	6.10
13	2.85	1.95	3.60	6.10
14	3.30	1.95	3.50	6.45
15	2.85	1.85	3.45	11.81
16	2.50	1.90	3.60	10.06
17	3.15	1.80	3.45	6.35
18	2.80	1.55	3.50	6.55
19	2.80	1.55	3.40	6.35
20	2.75	1.55	3.50	6.40
21	3.30	1.55	3.55	7.40
22	3.05	1.45	3.30	7.35
23	3.05	1.45	3.65	11.06
24	3.25	1.50	3.75	6.45
25	3.15	1.45	3.60	11.81
26	3.15	1.50	3.60	6.40
27	2.45	1.70	3.70	6.60
28	2.85	1.45	3.60	6.05
29	2.75	1.55	3.70	11.91
30	3.10	1.50	3.65	7.40
Media	3.06	1.94	3.59	6.55
Moda	3.30	1.95	3.60	6.40
Máximo	3.65	3.65	3.80	11.91
Mínimo	2.45	1.45	3.30	0.65
Desv. estándar	0.31	0.57	0.12	2.92

Por ejemplo, se observa que en los casos 1 y 3, el sistema no logra estabilizarse dentro del margen aceptado de $\pm 5\%$, pero sí muestra una tendencia clara de convergencia dentro de un margen de $\pm 10\%$, con tiempos promedio de estabilización de 3,06 s y 3,59 s, respectivamente.

Esto indica un control parcialmente efectivo y sugiere que la respuesta, aunque no cumple con el criterio establecido, no es errática ni inestable, sino que presenta una oscilación significativa que el controlador no logra compensar de forma adecuada dadas las condiciones físicas del entorno. En el caso 2 se observa un comportamiento similar, donde se alcanza una aparente estabilización dentro de un margen de $\pm 15\%$ tras un tiempo promedio de 1,94 s.

Analizando los casos de uso bajo estos criterios, se puede observar que en el caso 1 la respuesta del sistema fue relativamente uniforme, con una desviación estándar moderada de 0.31, lo que sugiere un comportamiento predecible y suficientemente estable bajo condiciones planas, incluso con la resolución y frecuencia de muestreo disponibles.

El caso 3, asociado a operación sobre arena, presentó una media de 3.59 s y una desviación estándar baja de 0.12 s, lo cual indica una respuesta lenta pero consistente. Este comportamiento puede explicarse por el efecto amortiguador del terreno arenoso, que, aunque suaviza la dinámica del sistema, también puede extender el tiempo requerido para alcanzar la estabilidad.

Por otro lado, el caso 2, sobre superficie con piedras, tampoco cumple con el criterio de estabilidad bajo el margen del $\pm 10\%$, ya que la mayoría de las oscilaciones se mantienen alrededor del $\pm 15\%$. Este comportamiento puede atribuirse a la naturaleza impredecible del contacto con los obstáculos, que varía en cada repetición y genera la mayor dispersión observada entre los cuatro escenarios con una desviación estándar de 0.57 s.

Finalmente, el caso 4, correspondiente al descenso por rampa, representa el desempeño menos favorable, con una media de 6.55 s y una desviación estándar significativamente alta de 2.92 s. El sistema no logra estabilizarse adecuadamente en la mayoría de las repeticiones, probablemente debido al efecto sostenido de la gravedad sobre la pendiente, lo que impide que la velocidad angular converja de forma sostenida dentro del margen aceptable. En varios casos, los tiempos de estabilización reportados superan los 10 segundos, y considerando que la duración total de la prueba es de 11 segundos, es razonable asumir que el sistema no llegó a estabilizarse realmente. No fue posible extender este tiempo de prueba debido a las limitaciones físicas de la rampa disponible.

Durante este análisis del comportamiento de estabilización del sistema, se identificó que las oscilaciones presentes en varias de las pruebas no se deben exclusivamente a la respuesta del

entorno físico, sino que están fuertemente influenciadas por las limitaciones del sistema de realimentación de velocidad angular. Específicamente, se observó que el método de sensado utilizado, basado en la medición de pulsos de un sensor cada 50 ms, introduce un desfase temporal relevante entre la aplicación de la acción de control y la observación de sus efectos.

Dado que los cambios en los pulsos del sensor se manifiestan de forma perceptible solo cada 150 a 200 ms, especialmente en condiciones de baja velocidad como las empleadas en la prueba de rampa a 10 cm/s, el sistema registra periodos en los que no detecta variación alguna. Durante estos intervalos, el controlador PID interpreta erróneamente que la salida no ha respondido, y por tanto, incrementa progresivamente la señal de control. Una vez que finalmente se registran los pulsos, el sistema percibe un exceso de corrección y responde aplicando una subcompensación, generando así un ciclo de oscilaciones sostenidas. Este comportamiento no refleja inestabilidad clásica, pero sí una limitación fundamental de resolución y latencia en la realimentación que afecta directamente el desempeño del controlador.

Adicionalmente, el diseño del PID depende críticamente de la medición precisa del tiempo entre cambios, debido a la naturaleza de sus términos derivativo e integral, ambos sensibles a la relación entre error y tiempo transcurrido. Cuando los intervalos de muestreo no reflejan con precisión el comportamiento real del sistema, aunque el PID esté bien sintonizado teóricamente, no puede evaluar correctamente la dinámica del error, lo que degrada sustancialmente la calidad de control.

Como medida de validación para el sistema de sensado, se utilizó una aplicación móvil como referencia externa, en la cual se ajustaban manualmente los valores de RPM hasta lograr una visualización estable, como se detalla en la sección correspondiente. Las pruebas de sensado duraban aproximadamente un minuto para asegurar ese ajuste fino manual, y se promediaban los datos obtenidos cada 50 ms. Bajo estas condiciones, se logró un porcentaje de error menor al 1% y, con ayuda de filtrado de la señal, un modelo matemático del sistema con un 81 % de ajuste, lo cual fue considerado adecuado dadas las limitaciones de resolución del sensor. Sin embargo, este enfoque resultó ser insuficiente para un control PID efectivo, ya que el filtro de media móvil, aunque útil para modelado, no resuelve el desfase inherente entre acción y respuesta en tiempo real.

En consecuencia, se concluye que para mejorar el desempeño dinámico del sistema en futuras implementaciones, es fundamental utilizar un sensor o *encoder* con mayor resolución temporal, capaz de proporcionar una realimentación precisa a las frecuencias de muestreo deseadas. Esto permitiría aprovechar plenamente las capacidades del controlador PID y evitar el comportamiento oscilante observado.

Los resultados experimentales también evidenciaron que el sistema presenta un mejor desempeño en condiciones donde existe una mayor amortiguación natural, como en el caso de la superficie de arena. En estas pruebas, la respuesta fue menos inestable, lo que facilitó el control. Por el contrario, superficies como la rampa, que reducen el amortiguamiento y favorecen la aceleración debido a la acción de la gravedad, tienden a acentuar las oscilaciones del sistema. Como parte del proceso de ajuste del controlador, se evaluó el incremento del término derivativo del PID, tanto en magnitud como en signo, con el objetivo de aumentar el nivel de amortiguamiento de esta forma. No obstante, esta modificación generó un incremento significativo en el tiempo de respuesta y en el tiempo de estabilización. Considerando que estas métricas fueron clasificadas con niveles de importancia 3 y 4 respectivamente, y que el sobreimpulso posee también un nivel de importancia 3, se concluyó que la estrategia más adecuada consistía en mantener una configuración que ofreciera un equilibrio entre estas tres métricas. De este modo, la respuesta del controlador seleccionado es una que, si bien no elimina por completo las oscilaciones, fue la que ofreció el mejor desempeño global del sistema durante los experimentos realizados.

6.2.4 Prueba de validación de tiempo de respuesta

Respecto al criterio de desempeño relacionado con el tiempo de respuesta, los resultados mostrados en la Tabla 13 evidencian un cumplimiento desigual dependiendo de las condiciones del entorno.

Tabla 13. Resultados del tiempo de respuesta del sistema en distintos casos de uso.

Experimento	Tiempo de respuesta			
	Caso 1	Caso 2	Caso 3	Caso 4
1	2.90	1.50	2.90	0.75
2	2.80	1.50	2.85	0.85
3	2.75	1.50	2.85	0.35
4	3.00	1.45	2.80	0.35
5	2.70	1.10	3.15	0.30
6	2.35	1.50	2.85	0.30
7	2.70	2.50	2.90	0.30
8	2.10	3.35	2.90	0.30
9	2.60	2.50	2.90	0.35
10	2.05	1.50	2.85	0.70
11	2.50	1.45	2.80	0.35
12	2.35	1.45	2.85	0.35
13	2.05	1.40	2.80	0.30
14	2.65	1.45	2.80	0.30
15	2.05	1.45	2.80	0.30
16	1.75	1.10	2.80	0.30
17	2.35	1.40	2.75	0.30
18	2.05	1.10	2.75	0.30
19	2.10	1.10	2.70	0.35
20	2.05	1.10	2.75	0.30
21	2.65	1.05	2.70	0.30
22	2.35	1.10	2.70	0.30
23	2.30	0.70	2.85	0.30
24	2.65	1.10	2.85	0.75
25	2.60	0.70	2.90	0.30
26	2.55	0.70	2.85	0.30
27	1.70	1.05	2.85	0.30
28	2.05	0.70	2.75	0.35
29	2.05	1.05	2.90	0.35
30	2.50	0.75	2.85	0.30
Media	2.38	1.34	2.83	0.38
Moda	2.05	1.50	2.90	0.30
Máximo	3.00	3.35	3.15	0.85
Mínimo	1.70	0.70	2.70	0.30
Desv. estándar	0.34	0.57	0.09	0.16

El caso 1, sobre césped artificial plano, presenta un tiempo de respuesta promedio de 2.38 s, el cual, si bien no cumple con el valor ideal, se mantiene dentro del rango marginalmente aceptable. Este resultado indica que en condiciones planas, pero sin asistencia gravitacional ni

amortiguación significativa, el sistema responde con cierta lentitud, posiblemente por la falta de aceleración pasiva que sí está presente en otros escenarios.

El caso 2, que representa una superficie irregular con piedras, presenta una media de 1.34 s, logrando cumplir con el criterio objetivo. Sin embargo, también registra la desviación estándar más alta en 0.57 s, lo que sugiere una alta variabilidad en la respuesta, posiblemente causada por el carácter impredecible de las perturbaciones introducidas por el terreno. Es importante mencionar que alrededor de cada tres pruebas, las piedras eran movidas por el Rover durante su desplazamiento por lo que había que volver a distribuir las en su camino, entonces la disposición de los obstáculos era ligeramente distinta en cada iteración. Aun así, todos los valores registrados se encuentran dentro del rango aceptable a excepción de una medición de 3.35 s que se considera como un caso atípico.

El caso 4, correspondiente al descenso por rampa, se destaca como el escenario con mejor desempeño, con una media de tan solo 0.38 s, muy por debajo del valor objetivo de 1.5 s. Esta respuesta acelerada puede atribuirse al efecto de la gravedad, que favorece la dinámica del sistema al reducir el esfuerzo necesario para alcanzar el punto de consigna, permitiendo al controlador ejecutar una corrección más rápida. Sin embargo, esta ventaja viene acompañada de un sobreimpulso elevado como se discute en la siguiente sección.

En el extremo opuesto, el caso 3, correspondiente a operación sobre arena, muestra el desempeño menos favorable, con una media de 2.83 s, lo cual lo posiciona por encima del umbral objetivo y dentro del rango marginalmente aceptable. Este comportamiento sugiere que la resistencia pasiva del terreno arenoso dificulta la generación de tracción efectiva, reduciendo la capacidad del sistema para alcanzar rápidamente su objetivo. No obstante, la baja desviación estándar de 0.09 s indica una respuesta consistente, aunque lenta, a lo largo de las repeticiones.

En resumen, tres de los cuatro casos se ubican dentro del rango marginalmente aceptable, y uno cumple estrictamente con el tiempo de respuesta objetivo. Esto resalta la importancia de considerar no solo el valor medio, sino también la consistencia de la respuesta ante perturbaciones del entorno, lo cual puede ser crítico para garantizar un desempeño confiable en condiciones operativas reales.

6.2.5 Prueba de validación de sobreimpulso

En cuanto al criterio de desempeño relacionado con el sobreimpulso, los resultados obtenidos muestran un cumplimiento parcial de los rangos establecidos como se puede observar en la Tabla 14. En los casos 1 y 3, la media del sobreimpulso se mantiene por debajo del 10 %, lo cual se encuentra dentro del rango objetivo definido. Por su parte, el caso 2 presenta una media de 12,56 %, situándose dentro del rango marginal aceptable, establecido por debajo del 15 %. Finalmente, el caso 4 es el único que no cumple con los criterios definidos, al registrar una media de sobreimpulso de 19,53 %, superando el límite máximo permitido.

Tabla 14. Resultados del sobreimpulso del sistema en distintos casos de uso.

Experimento	Sobreimpulso			
	Caso 1	Caso 2	Caso 3	Caso 4
1	1.68	10.95	0.00	17.94
2	2.49	10.43	0.00	10.53
3	2.20	12.75	0.95	9.30
4	3.50	11.40	1.66	10.98
5	2.81	13.61	0.00	12.41
6	4.64	13.52	0.00	9.13
7	5.38	10.37	0.00	14.44
8	5.56	6.91	0.00	15.13
9	5.50	9.84	0.00	15.58
10	5.94	10.36	0.00	21.69
11	6.18	12.71	0.00	13.89
12	5.56	12.80	0.00	17.78
13	4.86	13.50	0.77	20.50
14	3.66	13.41	1.43	24.16
15	6.11	13.89	2.47	23.89
16	6.45	13.83	1.88	29.81
17	6.09	14.82	1.69	27.03
18	6.09	12.47	1.16	23.07
19	5.69	13.08	1.65	22.16
20	5.40	13.27	2.35	23.21
21	4.86	13.24	1.74	21.31
22	5.28	14.96	3.93	24.70
23	4.90	13.01	0.00	26.14
24	4.75	13.52	0.00	16.68
25	4.97	13.10	0.00	23.56
26	5.90	12.59	0.00	24.46
27	6.72	13.85	0.00	22.37
28	5.58	12.99	0.66	14.34
29	5.72	13.15	0.00	22.65
30	6.16	12.35	0.00	26.99
Media	5.02	12.56	0.74	19.53
Máximo	6.72	14.96	3.93	29.81
Mínimo	1.68	6.91	0.00	9.13
Desv. estándar	1.31	1.66	1.03	5.81

El análisis de los resultados obtenidos revela una relación lógica entre las condiciones del terreno y el comportamiento dinámico del sistema, particularmente en lo referente al sobreimpulso. El caso 2, que consiste en la operación del Rover sobre una superficie irregular compuesta por piedras de distintos tamaños distribuidas aleatoriamente, presenta un aumento

evidente del sobreimpulso en comparación con el caso 1, que se desarrolla sobre césped artificial plano. Este incremento, de una media de 5.02 % en el caso 1 a 12.56 % en el caso 2, puede atribuirse a las perturbaciones mecánicas no lineales introducidas por las irregularidades del terreno. Las piedras provocan variaciones abruptas en la tracción y cambios imprevistos en la dinámica del vehículo, lo que reduce la capacidad del controlador PID para mantener un perfil de respuesta suave, y por ende genera un sobreimpulso mayor.

Por otro lado, el caso 3, correspondiente a la operación sobre arena, muestra el desempeño más favorable con una media de sobreimpulso de apenas 0.74 %. Este resultado puede explicarse por el efecto de amortiguamiento natural que ofrece el terreno arenoso, el cual tiende a suavizar las aceleraciones repentinas y oscilaciones mecánicas del sistema, facilitando así el trabajo del controlador para mantener una respuesta más estable. No obstante, se debe considerar que este mismo efecto puede tener implicaciones sobre otros criterios de desempeño como el tiempo de estabilización y la velocidad alcanzada, como se analizó en las secciones anteriores.

En contraste, el caso 4, correspondiente al descenso por una rampa, reporta un mal desempeño con una media de 19.53 %. En este escenario, la acción de la gravedad incrementa significativamente la aceleración del sistema y dificulta el control fino durante la transición inicial del movimiento, especialmente si el controlador no está adecuadamente sintonizado para compensar variaciones fuertes de pendiente. Esto se traduce en un sobreimpulso considerable que supera el umbral definido como aceptable. Adicionalmente, el alto valor de la desviación estándar de 5.81 % sugiere una alta variabilidad entre repeticiones, lo cual podría indicar que el comportamiento del sistema en este tipo de superficie es menos predecible o que existen factores externos influyendo de forma más agresiva.

6.2.6 Prueba de validación de desempeño en distintas superficies

Para evaluar el comportamiento del sistema de control en distintas condiciones físicas del terreno, se aplicó la métrica de variación porcentual definida en la Sección 4.3.6, utilizando como superficies de prueba el césped artificial y la arena. La Tabla 15 muestra los resultados experimentales obtenidos para tres indicadores clave: tiempo de respuesta, tiempo de

estabilización y sobreimpulso. A partir de estos datos, se calculó la variación porcentual relativa entre ambas superficies, así como el promedio de dichas variaciones. El valor obtenido fue de 50.72 %, por encima del límite marginal establecido de 40 %, lo que indica que el sistema no logra cumplir con el criterio definido por un margen de 10.72 %. Al analizar los indicadores por separado, se observa que el tiempo de respuesta presenta variación inferior al valor marginal. Sin embargo, el tiempo de estabilización y el porcentaje de sobreimpulso mostraron una variación significativamente mayor, elevando el promedio general.

Tabla 15. Variación porcentual entre distintas superficies.

Superficie	Respuesta (s)	Estabilización (s)	Sobreimpulso (%)
1	2.38	13.19	5.02
2	2.83	6.89	0.74
Variación %	19.23	47.76	85.16
Variación Promedio %	50.72		

Este resultado evidencia que la mayor sensibilidad del sistema ante cambios de superficie se manifiesta en la magnitud del sobreimpulso, lo cual puede atribuirse a la interacción compleja entre las ruedas del Rover y la superficie arenosa, especialmente durante las fases iniciales de aceleración. La superficie arenosa impone exigencias adicionales de tracción ya que, a diferencia del césped artificial, la arena ofrece una resistencia variable y menos uniforme al avance, lo que obliga al sistema motriz a generar un mayor esfuerzo para mantener el movimiento. Esta condición introduce una carga mecánica adicional sobre el motor, lo que se traduce en una respuesta más lenta y amortiguada del sistema de control. En consecuencia, se observa un incremento en el tiempo de respuesta, dado que el motor requiere más tiempo para alcanzar la velocidad objetivo bajo estas condiciones. Sin embargo, el tiempo de estabilización tiende a disminuir, ya que la mayor amortiguación presente en el entorno contribuye a que el sistema alcance un estado estable más rápidamente.

Además, esta respuesta sobreamortiguada reduce significativamente la tendencia natural del sistema a sobrepasar el valor de referencia, lo que explica la marcada disminución del sobreimpulso observada en la superficie arenosa respecto al césped artificial. Así, el cambio drástico en este indicador no obedece a un comportamiento inestable, sino a un efecto directo de la dinámica más lenta y contenida impuesta por las condiciones del terreno.

Este resultado también sugiere que el modelo actual del sistema no representa con suficiente precisión las variaciones de carga generadas por superficies de mayor resistencia, por lo que se requiere un refinamiento del modelado para mejorar la capacidad de adaptación del controlador frente a este tipo de circunstancias.

6.2.7 Resumen de validación

Para facilidad de análisis, la Tabla 16 muestra el nivel de cumplimiento alcanzado para las distintas métricas por cada caso de uso.

Tabla 16. Nivel de cumplimiento de métricas.

Métrica	Casos	Nivel de cumplimiento		
		Ideal	Aceptable	Insuficiente
Los bloques del sistema pueden ser mapeados a agentes	Todos	X		
Error de velocidad	1	X		
	2			X
	3	X		
	4	X		
Tiempo de estabilización	Todos			X
Tiempo de respuesta	1		X	
	2	X		
	3		X	
	4	X		
Sobreimpulso	1	X		
	2		X	
	3	X		
	4			X
Variación de desempeño entre superficies	1 y 3			X

6.8 Análisis económico

En la Tabla 17 se presenta el desglose detallado de los costos asociados a la realización del presente proyecto. El componente de mayor valor corresponde al kit SunFounder GalaxyRVR, que incluye el Rover utilizado para la validación y todos sus componentes base. A este se suman los elementos adicionales requeridos para la implementación del sistema de sensado de velocidad y el almacenamiento de datos.

Asimismo, se consideran los materiales utilizados en la construcción del banco de pruebas, tales como arena, rocas, alfombra base y la rampa. En cuanto a los activos utilizados, se estimó el costo de uso de una computadora portátil valorada en 700 USD, aplicando el método de amortización lineal con una vida útil de 48 meses y un periodo de uso de cuatro meses. Bajo estos parámetros, el costo correspondiente es de 58,33 USD. Adicionalmente, se consideró el uso del software Matlab para el desarrollo y simulación del sistema. La universidad mantiene una licencia con un costo de 275 USD por usuario al año, por lo que el valor proporcional correspondiente al periodo de cuatro meses es de 91,67 USD.

Para el cálculo del costo asociado al recurso humano, se consideró un régimen de medio tiempo (20 horas semanales) durante las 16 semanas de duración del proyecto. Con una tarifa estimada de ₡2250 por hora, el costo total por concepto de trabajo del investigador asciende a ₡720 000, equivalentes a aproximadamente 1440 USD al tipo de cambio de referencia. El costo total estimado del proyecto es de 1801,91 USD.

Tabla 17. Desglose de costos del proyecto.

Categoría	Elemento	Unidades	Costo en \$USD
Componentes	SunFounder GalaxyRVR Kit	1	130
	Sensor Hall	1	2.49
	Imanes	12	10
	Cables	2	2.49
	OpenLog	1	19.95
	MicroSD	1	8
Subtotal Componentes			\$172.93
Banco de pruebas	Rampa	1	50
	Alfombra	2	25.5
	Arena	1	3.5
	Rocas	1	11.65
	Cinta adhesiva	1	3.2
Subtotal Banco de Pruebas			\$93.85
Activos utilizados	Licencia Matlab	4 meses	91,67

	Computadora portátil	4 meses	58.33
Subtotal Activos			\$150
Salario del investigador	Salario por hora		4.4
	Horas trabajadas por semana		20
	Semanas del proyecto		16
Subtotal Salario			\$1440
Total General			\$1856.78

Adquirir un Rover con capacidades avanzadas para investigación representa una inversión considerable que actualmente resulta prohibitiva para el laboratorio, ya que usualmente oscila entre los 5000 y 7000 dólares. Para contextualizar este dato, se consultó con expertos del *Interdisciplinary Centre for Security, Reliability and Trust* (Centro Interdisciplinario para la Seguridad, Confiabilidad y Confianza) de la Universidad de Luxemburgo, donde operan un laboratorio especializado en robótica espacial y simulación de ambiente lunar para validación de sistemas. Según esta fuente, sus prototipos base son adquiridos a través de la empresa LeoRover, donde el modelo más básico tiene un costo inicial de 4799 euros, al cual se le agregan los sensores y actuadores específicos requeridos por cada cliente, aumentando así el costo final según la personalización [40]. A partir de esta referencia, es posible dimensionar la diferencia significativa en términos de inversión. El prototipo funcional desarrollado en este proyecto, que incluye tanto el Rover base como los componentes adicionales necesarios para el sistema de sentido, control y almacenamiento de datos, tuvo un costo total de tan solo 172,93 USD. Esto representa una fracción mínima del costo de un Rover comercial como el LeoRover, incluso antes de considerar la personalización con sensores adicionales. Esta comparación revela que el proyecto logró una reducción de costo de más del \$4627 USD, sin dejar de cumplir con los requisitos técnicos para validación de sistemas de velocidad.

Este proyecto deja como resultado un prototipo funcional de Rover y un banco de pruebas físico, ambos disponibles como recursos permanentes dentro del SETEC Lab. Estos activos no solo permiten la validación experimental del sistema desarrollado, sino que también constituyen la base para futuras investigaciones. En el marco del proyecto ELANav, se proyecta el desarrollo de al menos 10 tesis de grado durante un periodo de dos años, de las cuales una por

semestre estará dedicada a continuar el diseño y evolución del Rover del laboratorio. Esta tesis representa la primera de esa línea de trabajo, por lo que se espera que al menos tres proyectos de graduación hagan uso directo de los componentes, materiales y banco de pruebas implementados en este trabajo. Considerando que la adquisición de un banco de pruebas y un prototipo comercial similar para cada uno de esos proyectos implicaría un costo cercano a los \$270 USD, se estima que el valor del aporte en infraestructura reutilizable asciende a aproximadamente \$810 USD, en términos de costos evitados por el laboratorio a futuro.

Cabe destacar que tanto los activos adquiridos como el salario del investigador asignado a este proyecto representan gastos no recurrentes, realizados una sola vez. Sin embargo, la base de conocimiento generada, que incluye el diagnóstico técnico, las recomendaciones de diseño y la caracterización detallada de los sistemas del Rover, tendrá un impacto sostenido en el tiempo. Este conocimiento facilitará y acelerará el desarrollo de futuras mejoras al prototipo, reduciendo la necesidad de repetir tareas ya resueltas. Se estima que las próximas tres tesis de grado que continuarán esta línea de trabajo podrán ahorrar al menos 45 horas de trabajo cada una, gracias a la documentación y los resultados obtenidos en esta investigación. Considerando una tarifa de \$4.40 USD por hora de trabajo, este ahorro representa un beneficio económico adicional de aproximadamente \$594 USD, únicamente en tiempo de investigación.

Desde la perspectiva institucional, este proyecto incrementa significativamente la capacidad del SETEC Lab para participar en convocatorias de financiamiento nacionales, como el programa ConectadODS de la Promotora Costarricense de Innovación e Investigación (PCII). Esta iniciativa destina ₡100 millones de colones a la financiación de al menos cuatro proyectos, asignando aproximadamente ₡25 millones de colones por propuesta seleccionada, y exige como requisito contar con un prototipo funcional. Gracias a los resultados de este trabajo, el laboratorio cumple con ese requerimiento técnico, lo que habilita su postulación. Si se considera una probabilidad conservadora del 10 % de adjudicación, el valor económico esperado derivado de esta capacidad de participación asciende a ₡2.5 millones de colones, equivalentes a aproximadamente \$5000 USD. Este monto representa un beneficio proyectado tangible, atribuible directamente al desarrollo del prototipo implementado en el presente proyecto. Esto se alinea con los objetivos estratégicos del proyecto ELANav, que incluyen el establecimiento

de al menos una alianza empresarial y una colaboración con un centro de investigación internacional.

Actualmente, el proyecto cuenta con el respaldo económico de la Vicerrectoría de Investigación y Extensión (VIE) del Tecnológico de Costa Rica, la cual asigna montos del orden de ₡2.5 millones, aproximadamente \$5000 USD, para iniciativas de investigación. A partir de los resultados alcanzados durante este trabajo, existe la posibilidad de ampliar dicho financiamiento o formular nuevas propuestas de investigación derivadas, lo que contribuiría a incrementar el retorno de inversión institucional asociado a este proyecto.

También se contempla la presentación de estos avances en congresos internacionales, como el Congreso Internacional de Astronáutica, así como la publicación de al menos un artículo científico indexado por año, en concordancia con los objetivos del proyecto ELANav. Estas acciones contribuirán directamente al incremento de la producción científica del laboratorio y fortalece el posicionamiento de la reputación técnica del laboratorio.

Todos los aportes mencionados resultan fundamentales para el objetivo del laboratorio de consolidar un conjunto de herramientas abiertas, orientadas a comunidades que desarrollan sistemas espaciales de misión crítica, incluyendo universidades, agencias espaciales y empresas del sector. En la Tabla 18 se presenta un resumen comparativo entre los recursos invertidos y los recursos generados a partir de la ejecución del proyecto.

Tabla 18. Estimación de recursos generados por el proyecto.

Concepto	Descripción	Valor estimado (USD)
Reducción de costo frente a Rover comercial	Diferencia entre el costo de un Rover comercial para investigación y el prototipo desarrollado	\$4627
Infraestructura reutilizable	Uso estimado por al menos 3 tesis futuras. Costo evitado de \$270 USD por tesis (prototipo y banco de pruebas)	\$810
Ahorro en horas de investigación futuras	45 horas ahorradas por 3 tesis a \$4.40USD/hora gracias al conocimiento y diagnóstico generado	\$594

Valor potencial por participación en convocatoria ConectadODS	Probabilidad del 10% de obtener equivalente a \$5000 USD al cumplir requisito de prototipo funcional	\$5000
Proyección de financiamiento institucional adicional (VIE)	Posibilidad de ampliar financiamiento o generar nuevos proyectos derivados de este trabajo	\$5000
Total estimado de recursos generados		\$16031

En resumen, el análisis económico del proyecto evidencia una gestión eficiente de los recursos disponibles, logrando desarrollar un prototipo inicial de bajo costo, con una inversión total de 1856.78 USD, y con un alto potencial de impacto académico, científico e institucional. La inversión realizada no solo permitió la validación experimental del sistema y la creación de infraestructura reutilizable, sino que también habilita la proyección de resultados tangibles en términos de publicaciones, alianzas estratégicas y formación de capital humano. El valor estimado de los recursos generados asciende a 16031 USD, lo que refleja una relación entre beneficios y costos altamente favorable. Estos aportes contribuyen directamente al cumplimiento de los objetivos del proyecto ELANav y a la visión del Laboratorio de Sistemas Espaciales como referente técnico en el desarrollo de herramientas abiertas para el control de sistemas espaciales.

Capítulo 7. Conclusiones y recomendaciones

7.1 Conclusiones

Al finalizar este proyecto, se logró diseñar e implementar un sistema de control de velocidad para un Rover enfocado en aplicaciones lunares, tomando en cuenta tanto los requisitos operativos del entorno como las limitaciones prácticas del hardware disponible. El sistema fue evaluado mediante pruebas experimentales sobre distintas superficies, simulando condiciones relevantes para su uso final. Los resultados obtenidos permiten concluir que el proyecto cumplió con su objetivo general, sentando una base sólida para futuras mejoras y para el avance en el desarrollo del Rover del Laboratorio de Sistemas Espaciales. Las conclusiones alcanzadas se detallan a continuación:

- Se realizó un diagnóstico inicial del prototipo disponible para la validación, el cual permitió identificar limitaciones críticas en su estructura y operatividad. Como resultado, se descartó su uso y se seleccionó un nuevo prototipo que ofreciera condiciones más apropiadas para el desarrollo y prueba del sistema de control. A partir de esta selección, se definieron las especificaciones técnicas pertinentes a la fase actual de desarrollo, considerando los casos de uso definidos por el cliente y el propósito de la solución dentro del marco del Laboratorio de Sistemas Espaciales.
- Se logró diseñar un sistema de control PID que responde a los requerimientos operativos definidos, manteniendo un desempeño funcional en la mayoría de los escenarios evaluados. El controlador fue implementado con una arquitectura modular, mapeando los bloques del sistema de control a agentes funcionales, lo cual permite una estructura escalable y adaptable a futuras mejoras. Los resultados del error entre la velocidad de referencia y la velocidad real del Rover se encuentran por debajo del 10%, dentro del rango ideal definido en las especificaciones, excepto por el caso donde la superficie tiene obstáculos distribuidos aleatoriamente. Aunque el desempeño no fue óptimo en todos los casos, por ejemplo, sobre terreno con obstáculos, pendientes pronunciadas o

en relación con el tiempo de estabilización, las limitaciones observadas se atribuyen principalmente a las restricciones del hardware disponible, en particular a la resolución del PWM y la capacidad de sensado. A pesar de estas limitaciones, el diseño del algoritmo cumplió con su propósito general, mostrando un equilibrio razonable entre desempeño, simplicidad y viabilidad de implementación.

- Para evaluar el desempeño de la solución propuesta, se diseñó y construyó un banco de pruebas con cuatro tipos de superficies: césped artificial, terreno con piedras, arena fina y rampa con inclinación, representando así terrenos planos y con obstáculos, de acuerdo con los desafíos operativos de los rovers en entornos lunares descritos en la literatura. Las pruebas se realizaron siguiendo casos de uso definidos en conjunto con el cliente, bajo condiciones que simulan situaciones reales de exploración planetaria. Los resultados experimentales mostraron que el sistema de control posee un error por debajo del 10% ideal; un tiempo de respuesta menor a 3 segundos en superficies planas y menor a 1,5 segundos en superficies irregulares; un comportamiento oscilante que no le permite alcanzar la estabilización a $\pm 5\%$ del valor de referencia; un sobreimpulso desigual dependiendo de las condiciones del entorno presentando un valor objetivo de menos de 10% en superficies planas, un valor marginal menor al 15% en superficies irregulares e incumpliendo el rango aceptable cuando la superficies es inclinada; y una variación de desempeño de hasta 50.72% entre distintas superficies. Estos hallazgos evidencian que, aunque el sistema cumple parcialmente con los criterios establecidos, se requiere una mejora en las capacidades de sensado y procesamiento para lograr un control más preciso en condiciones más exigentes.

7.2 Recomendaciones

A continuación, se presentan una serie de recomendaciones orientadas a facilitar el desarrollo y la mejora del Rover y sus sistemas de control en investigaciones futuras:

- Sustituir el sistema actual de medición de velocidad por uno con mayor resolución, con el fin de permitir un control más fino y estable que incremente la precisión del movimiento del Rover.
- Reemplazar los motores utilizados por otros que incorporen *encoders* integrados, lo cual facilitaría tanto el cableado como el montaje del sistema de sensado.
- Optar por un microcontrolador con mayor resolución en la generación de señales PWM, lo que simplificaría la sintonización del controlador y mejoraría la calidad de la señal de control.
- Rediseñar la placa de expansión para incluir más puertos y asegurar que sean compatibles con componentes comerciales disponibles localmente.
- Implementar control independiente para cada lado del Rover. Esta mejora permitiría un mejor comportamiento del vehículo en presencia de perturbaciones y aumentaría su capacidad de orientación.
- Considerar el uso de una estrategia basada en modos de operación, en la que distintos algoritmos de control se activen según el escenario. Con sensores adicionales, la selección del modo puede realizarse de forma automática. Este enfoque es común en rovers de exploración planetaria.
- Debido a que los motores tienden a calentarse con facilidad, se recomienda definir trayectos en secuencias cortas, intercalando pausas entre ellas para evitar sobrecalentamientos.

Bibliografía

- [1] J. Carvajal-Godínez, Agent-based architectures supporting fault-tolerance in small satellites, Vols. %1 de %2doi: 10.4233/uuid:b528d7be-e82d-4205-abdf-3fb3fa7f1011., Delft: Delft University of Technology, 2021.
- [2] J. Carvajal-Godínez, *Proyecto de investigación: Biblioteca embebida para la navegación de sistemas espaciales de misión crítica (ELANav)*, Cartago: Instituto Tecnológico de Costa Rica, 2024.
- [3] C. Chan-Zheng, MAES: A Multi-Agent Systems Framework for Embedded Systems, M.Sc. thesis, Vols. %1 de %2Available: <http://resolver.tudelft.nl/uuid:e1c076dc-ab64-4e77-8453-bce618033907> , Delft: Delft University of Technology, 2017.
- [4] D. Rojas, *Caravela: Sistemas Multiagente para Aplicaciones Críticas en el Lenguaje Rust*, Cartago: Tecnológico de Costa Rica, 2025.
- [5] A. Cracknell, C. Varotsos, «Editorial and cover: Fifty years after the first artificial satellite: from Sputnik 1 to ENVISAT,» *International Journal of Remote Sensing*, vol. 28(10), p. 2071–2072. <https://doi.org/10.1080/01431160701347147>, 2007.
- [6] K. Pradhan, «Rovers with Rocker-Bogie Mechanisms: A review of progress and innovations in design, manufacturing, and control,» de *Proceedings of the Institution of Mechanical Engineers*, 2024.
- [7] G. Weiss, *Multiagent Systems*, 2nd ed.; Intelligent Robotics and Autonomous Agents Series., Cambridge, MA: MIT Press, 2013.
- [8] NASA, «Cooperative Autonomous Distributed Robotic Exploration (CADRE),» URL: https://www.nasa.gov/directorates/spacetech/game_changing_development/projects/CADRE., 2021.
- [9] J. Guidi, D. Elburn, S. Creech, «Artemis: an overview of NASA's activities to return humans to the Moon,» 2022.
- [10] M. Maimone, E. Tunstel, «Surface navigation and mobility intelligence on the Mars Exploration Rovers,» *Intelligence for Space Robotics*, p. pp. 45–69., 2006.

- [11] O. Toupet, A. Rankin, «Terrain-adaptive wheel speed control on the Curiosity Mars rover: Algorithm and flight results,» *Journal of Field Robotics*, vol. vol. 37, n° no. 5, p. pp. 699–728, 2020.
- [12] NASA, «Mars Science Laboratory: Curiosity Rover,» Abril 2025. [En línea]. Available: <https://science.nasa.gov/mission/msl-curiosity/>. [Último acceso: 31 Mayo 2025].
- [13] V. Verma, M. Maimone, D. Gaines, «Autonomous robotics is driving Perseverance rover's progress on Mars,» *Science robotics*, vol. 8, p. doi: 10.1126/scirobotics.adi3099., 2023.
- [14] NASA, «Perseverance Rover Components,» 2020. En línea]. Available: <https://science.nasa.gov/mission/mars-2020-perseverance/rover-components/> [Último acceso: Mayo 2025].
- [15] N. Teng, X. He, «A Review of Control Techniques for Lunar Rovers,» de *Proceedings of the 2024 2nd International Conference on Frontiers of Intelligent Manufacturing and Automation*, 2024.
- [16] New York Times, «China's Rover Finds Layers of Surprise Under Moon's Far Side,» 26 Febrero 2020. [En línea]. Available: <https://www.nytimes.com/2020/02/26/science/china-moon-far-side.html>. [Último acceso: 31 Mayo 2025].
- [17] «Chandrayaan-3: India's Third Lunar Exploration Mission,» StudyHub, [Online]. Available: <https://studyhub.net.in/space-science/chandrayaan-3/>, [Accessed: 12-May-2025].
- [18] K. Ulrich, S. Eppinger, *Diseño y desarrollo de productos*, vol. 1, 2013.
- [19] SunFounder, «SunFounder Mars Rover Kit Documentation,» 2025. [En línea]. Available: <https://docs.sunfounder.com/projects/galaxy-rvr/en/latest/>. [Último acceso: Mayo 2025].
- [20] J. Aufranc, «SunFounder GalaxyRVR review – An Arduino programmable Mars Rover-like robot for education,» CNX Software, 2024.
- [21] NASA, «Spacecraft: Mars Science Laboratory (MSL) – Curiosity,» NASA Space Science Data Coordinated Archive, Disponible en: <https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=2011-070A>. [Accedido: 1-abr-2025].

- [22] NASA, «Spacecraft: Mars Exploration Rover - B (MER-B) – Opportunity,» NASA Space Science Data Coordinated Archive, Disponible en: <https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=2003-027A>. [Accedido: 1-abr-2025]..
- [23] J. K. Olivier Lamarre, «Overcoming the Challenges of Solar Rover Autonomy: Enabling Long-Duration Planetary Navigation,» de *In Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS'18)*, Madrid, Spain, Jun. 4-6, 2018.
- [24] A. Su, T. Naing, «Comparison of Response Time for Target-Following Mobile Robot between ON/OFF and PID Control Algorithms,» *International Journal of Latest Technology in Engineering, Management & Applied Science (IJLTEMAS)*, vol. VIII, nº VI, 2019.
- [25] W. S. Levine, *The Control Handbook*, Boca Raton, Florida: CRC Press, 1996, p. xvi+1548.
- [26] Benjamin Kuo, *Automatic control systems*, New York: John Wiley & Sons, 2003.
- [27] G. Ishigami, «Terramechanics-based model for steering maneuver of planetary exploration rovers on loose soil.,» *Journal of Field robotics*, vol. 24, nº 3, pp. 233-250, 2007.
- [28] S. Noble, «The Lunar Regolith,» de *Lunor Regolith Simulant Workshop*, Alabama, United States, 2009.
- [29] R. Dorf, R. Bishop, *Sistemas de control moderno*, vol. 10, Madrid: Pearson Educación, 2005.
- [30] Jayesh Tripathi, Krishna Sharma, «Speed Control Analysis of Brushless DC Motor Using PI, PID and Fuzzy-PI Controllers,» *International Journal of Electrical and Electronics Research*, vol. 10, pp. 470-474. doi:10.37391/ijeer.100311, 09 2022.
- [31] D. Puangdownreong, A. Nawikavatan, C. Thammarat, «Optimal Design of I-PD Controller for DC Motor Speed Control System by Cuckoo Search,» *Procedia Computer Science*, vol. 86, pp. 83-86. doi: <https://doi.org/10.1016/j.procs.2016.05.021>, 2016.
- [32] D. Sain, «PID, I-PD and PD-PI controller design for the ball and beam system: A comparative study,» vol. 9, pp. 9-14, 2017.

- [33] S. Fadali, A. Visioli, «State feedback control,» de *Digital Control Engineering*, 2020, pp. 387-439.
- [34] L. Moreno, S. Garrido, *Apuntes de Control de Sistemas II*, 2015.
- [35] V. Shreya, «A Comparative Study of PI Controller Performance in Simulated and Real-World Interplanetary Rover Actuator Systems,» Ramaiah University of Applied Sciences, Bangalore, 2024.
- [36] P. Stone, M. Veloso, «Multiagent Systems: A Survey from a Machine Learning Perspective,» *Autonomous Robots*, vol. 8, n° 10.1023/A:1008942012299, 2000.
- [37] S. Russell y P. Norvig, *Artificial Intelligence: A Modern Approach*, Egglewood Cliffs: Prentice Hall, 1995.
- [38] M. Sutoh, T. Hoshino y S. Wakabayashi, «Rover deployment system for lunar landing mission,» *Acta Astronautica*, , n° 138(), p. 454–461. doi: <https://doi.org/10.1016/j.actaastro.2017.06.019>, 2017.
- [39] A. Shetty, H. Kumar, «Rover ramp deployment system for Chandrayaan-3: Concept, design, development and operations.,» *Acta Astronautica*, vol. 229, pp. 1-12, ISSN 0094-5765. , 2025,.
- [40] LeoRover, «Leo Rover Kits,» [En línea]. Available: <https://www.leorover.tech/shop>. [Último acceso: 2 Junio 2025].
- [41] K. Ogata, *Ingeniería de Control Moderna*, 5ta Edición, Mexico: Pearson Educación, 2010.

Apéndices

Apéndice A. Concepto de Operaciones

Instituto Tecnológico de Costa Rica

Concepto de Operaciones (CONOPS)

Sistema de control de velocidad y batería de pruebas para Rover con
arquitectura *Rocker-Bogie*

Laboratorio de Sistemas Espaciales

Autora:

María Fernanda del Barco León

Junio 2025

1. Introducción

1.1 Propósito del Documento

Este documento describe el Concepto de Operaciones (CONOPS) del sistema de control de velocidad desarrollado para el Laboratorio de Sistemas Espaciales del Instituto Tecnológico de Costa Rica. El objetivo es validar el desempeño del sistema bajo tres escenarios representativos: desplazamiento en terreno plano, pendiente descendiente y terreno irregular. El CONOPS proporciona una visión de las condiciones de operación, objetivos de las pruebas y criterios de éxito asociados, además de las interfaces existentes entre los subsistemas, el entorno y el usuario.

1.2 Alcance del Sistema

El sistema objeto de estudio es un controlador de velocidad para un modelo de Rover estilo *Rocker-Bogie* equipado con seis motores de corriente continua. Su función principal es mantener una velocidad deseada mediante ajustes a la potencia eléctrica entregada a los motores, adaptándose a diferentes condiciones de terreno. Este documento establece los escenarios de prueba, los procedimientos, los criterios de éxito y las interfaces clave del sistema.

2. Descripción general del sistema

2.1 Componentes

- Chasis de Rover con arquitectura *Rocker-Bogie*.
- Seis motores de corriente continua con rueda acoplada.
- Unidad de procesamiento principal Arduino Uno.
- Placa de expansión con drivers para seis motores TT.
- Sensor Hall acoplado sobre el motor, alineado con la circunferencia de la rueda que cuenta con 4 imanes equidistantes.
- Interfaz de comunicación con el usuario: IDE de Arduino.
- Módulo OpenLog y tarjeta microSD para almacenamiento de datos.

2.2 Arquitectura funcional

El sistema utiliza control PID para regular la velocidad. Los sensores proporcionan retroalimentación continua que es procesada por el controlador para ajustar la señal de control PWM. El control es diferencial entre los dos lados del robot, pero sincronizado dentro de cada lado.

Se implementa un controlador PID digital que se ejecuta en un microcontrolador Arduino Uno. El algoritmo se encarga de calcular, en intervalos de tiempo regulares, la diferencia entre la

velocidad de referencia y la velocidad actual medida por el sensor Hall. A partir de este error, el controlador genera una señal de salida que ajusta el ciclo de trabajo de la señal PWM, que a su vez controla la potencia entregada a los motores.

El control se aplica de forma independiente a cada lado del Rover (izquierdo y derecho), pero los tres motores de cada lado comparten una señal de control común. Esto significa que, aunque no hay control individual por rueda, sí se mantiene un control diferencial básico entre ambos lados del Rover, suficiente para garantizar un desplazamiento rectilíneo bajo condiciones normales.

El algoritmo PID, así como todas las rutinas necesarias para la lectura de sensores, el cálculo de velocidad, el control de motores y el registro de datos, se programan en el entorno Arduino IDE y se cargan al microcontrolador mediante conexión USB. Es importante mencionar que la placa de expansión del Rover cuenta con un switch que se debe activar para cargar el código de forma correcta y evitar conflictos con otros componentes electrónicos. Una vez cargado, el sistema puede operar de forma autónoma, ejecutando una secuencia predefinida de pruebas. Los datos recolectados se almacenan en una tarjeta microSD a través de un módulo OpenLog para su posterior análisis.

2.3 Interacción entre subsistemas

El sistema de control opera mediante una arquitectura cíclica de realimentación. El subsistema de sensado, compuesto por un sensor Hall montado en el eje del motor, detecta el paso de imanes adheridos a la rueda y genera pulsos proporcionales a la velocidad de rotación. Esta información se utiliza para calcular la velocidad angular y lineal del Rover en tiempo real.

Estos datos son enviados al subsistema de procesamiento, implementado en el microcontrolador Arduino Uno, donde se ejecuta el algoritmo PID. El controlador compara la velocidad actual con la velocidad de referencia predefinida y, a partir del error, calcula una señal de control.

Dicha señal se convierte en un valor PWM que es enviado al subsistema de actuación, conformado por los drivers de los motores. Los motores reciben esta señal modulada y ajustan su velocidad en consecuencia. El resultado del ajuste se refleja en el comportamiento del Rover, el cual es nuevamente registrado por el sensor Hall, cerrando así el lazo de control.

Este flujo de información entre sensado, procesamiento y actuación ocurre de forma continua durante el experimento, permitiendo la adaptación dinámica del sistema ante perturbaciones del entorno o cambios en las condiciones del terreno.

3. Interfaces del sistema

- **Interfaz de usuario:** El usuario interactúa con el sistema a través del entorno de desarrollo Arduino IDE. Esta interfaz se utiliza para cargar el código al microcontrolador, modificar parámetros del controlador y realizar ajustes durante la

etapa de desarrollo. No se utiliza para visualizar resultados en tiempo real, ya que los datos se almacenan externamente.

- **Interfaz con sensores:** El sensor Hall está conectado directamente al microcontrolador a través de un pin digital. La lectura de los pulsos generados por el sensor se realiza mediante muestreo periódico, permitiendo el cálculo de la velocidad del Rover. Esta interfaz es unidireccional, del sensor al controlador, y de tipo digital.
- **Interfaz con actuadores:** Los actuadores son los seis motores de corriente continua distribuidos simétricamente a ambos lados del Rover. Están controlados por señales PWM generadas desde el Arduino, utilizando la librería SoftPWM debido a la ocupación de los pines PWM por otras funciones. Esta interfaz es de tipo digital y bidireccional lógica (activación y apagado).
- **Interfaz de control:** La lógica de control está embebida en el código cargado al Arduino Uno. Internamente, el controlador PID regula el comportamiento del sistema basándose en los datos de sensado y las referencias definidas. Esta interfaz es completamente autónoma y no requiere intervención externa una vez cargado y ejecutado el programa.
- **Exportación de datos:** El sistema cuenta con una interfaz de exportación a través de un módulo OpenLog conectado por SoftwareSerial, que registra los datos en una tarjeta microSD en formato CSV. Esta información incluye variables como tiempo, velocidad medida, señal de control, error, entre otras, y es utilizada para análisis posteriores y validación del desempeño.

4. Supuestos operacionales

- Los sensores están calibrados previamente.
- Las pruebas se realizan bajo condiciones controladas.
- El hardware está en condiciones óptimas durante las pruebas.

5. Escenarios Operacionales

5.1 Prueba 1: Terreno Plano

Objetivo: Validar el control de velocidad sobre una superficie plana.

Métricas: Error de velocidad con respecto a referencia, tiempo de estabilización, tiempo de respuesta, sobreimpulso.

Procedimiento:

1. Activar el switch de “UPLOAD” en la placa de expansión.
2. Cargar el código del Arduino IDE al Arduino Uno mediante USB. En este se indica la velocidad de referencia y duración deseadas para la prueba.
3. Desactivar el switch de “UPLOAD” en la placa de expansión.
4. Desconectar cable USB.
5. Asegurar que la memoria microSD se encuentre en su lugar.
6. Colocar el Rover en la posición de inicio.

7. Encender el Rover.
8. Grabar la prueba para posteriormente calcular la velocidad real del Rover.
9. Apagar el Rover una vez que deja de moverse.
10. Realizar las repeticiones necesarias.
11. Extraer la microSD para obtener los datos experimentales.



Figura 1. Rover sobre superficie plana con arena fina.

5.2 Prueba 2: Descenso en pendiente

Objetivo: Validar el control de velocidad sobre una rampa.

Métricas: Error de velocidad con respecto a referencia, tiempo de estabilización, tiempo de respuesta, sobreimpulso.

Procedimiento:

1. Activar el switch de “UPLOAD” en la placa de expansión.
2. Cargar el código del Arduino IDE al Arduino Uno mediante USB. En este se indica la velocidad de referencia y duración deseadas para la prueba.
3. Desactivar el switch de “UPLOAD” en la placa de expansión.
4. Desconectar cable USB.
5. Asegurar que la memoria microSD se encuentre en su lugar.
6. Colocar el Rover en la posición de inicio.
7. Encender el Rover.
8. Grabar la prueba para posteriormente calcular la velocidad real del Rover.
9. Apagar el Rover una vez que deja de moverse.
10. Realizar las repeticiones necesarias.
11. Extraer la microSD para obtener los datos experimentales.

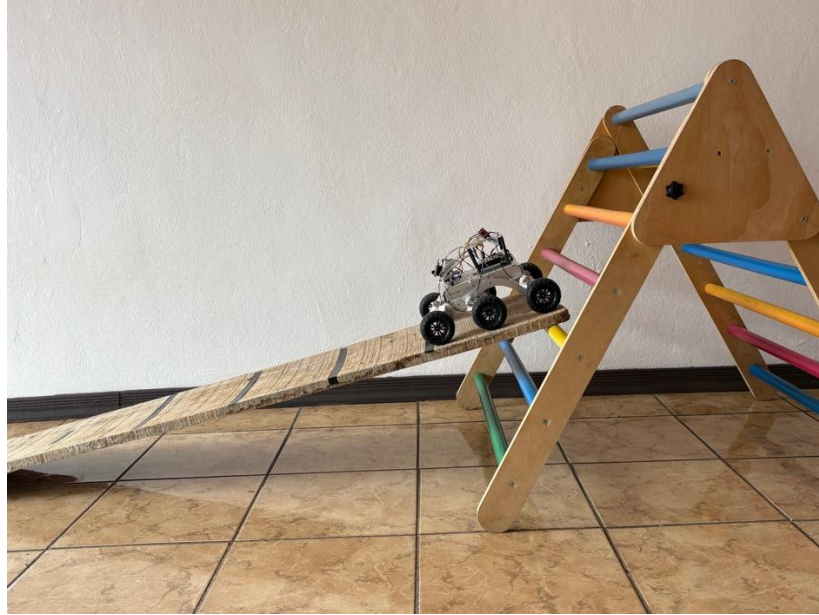


Figura 2. Rover sobre rampa.

5.3 Prueba 3: Terreno irregular

Objetivo: Validar el control de velocidad sobre una superficie con obstáculos o irregularidades distribuidos de forma aleatoria.

Métricas: Error de velocidad con respecto a referencia, tiempo de estabilización, tiempo de respuesta, sobreimpulso.

Procedimiento:

1. Activar el switch de “UPLOAD” en la placa de expansión.
2. Cargar el código del Arduino IDE al Arduino Uno mediante USB. En este se indica la velocidad de referencia y duración deseadas para la prueba.
3. Desactivar el switch de “UPLOAD” en la placa de expansión.
4. Desconectar cable USB.
5. Asegurar que la memoria microSD se encuentre en su lugar.
6. Colocar el Rover en la posición de inicio.
7. Encender el Rover.
8. Grabar la prueba para posteriormente calcular la velocidad real del Rover.
9. Apagar el Rover una vez que deja de moverse.
10. Realizar las repeticiones necesarias.
11. Extraer la microSD para obtener los datos experimentales.



Figura 3. Rover sobre superficie plana con obstáculos distribuidos aleatoriamente.

6. Criterios de éxito del sistema

El sistema será considerado exitoso si cumple con los criterios de desempeño definidos a partir de las necesidades funcionales del Rover, descritas en la siguiente tabla. Estos criterios se establecieron con valores marginales (mínimos aceptables) e ideales (deseables) que guían tanto el diseño como la evaluación del sistema de control de velocidad.

Número de métrica	Número de necesidad	Métrica	Imp	Unidades	Valor marginal	Valor ideal
1	1	Los bloques del sistema pueden ser mapeados a agentes	5	Binaria	Pasa	Pasa
2	2	Error de medición de velocidad	5	%	≤ 15	≤ 10
3	3	Tiempo de estabilización	4	s	≤ 4	≤ 2
4	4	Tiempo de respuesta	3	s	≤ 3	≤ 1.5
5	5	Sobreimpulso	3	%	≤ 15	≤ 10
6	6	Variación porcentual promedio en indicadores de desempeño	4	%	≤ 40	≤ 15

Estas métricas permiten una evaluación cuantitativa del sistema bajo los distintos escenarios definidos, asegurando que el desempeño sea funcional, estable y replicable ante variaciones del terreno.

7. Referencias

[1] SunFounder, Galaxy RVR - Documentation, SunFounder, 2024. [En línea]. Disponible en: <https://docs.sunfounder.com/projects/galaxy-rvr/en/latest/index.html> [Accedido: 13-jun-2025].

Apéndice B

Tabla B.1. Mediciones obtenidas durante la validación del sensado de velocidad angular para el experimento 1.

Experimento	PWM = 70		
	RPM Medido	RPM Referencia	Error (%)
1	27,93	28	0,25
2	27,99	28	0,04
3	27,15	27	0,56
4	26,59	27	1,52
5	26,78	27	0,81
6	26,88	27	0,44
7	26,71	27	1,07
8	26,73	27	1,00
9	26,66	27	1,26
10	27,11	27	0,41
Media	27,05	27,20	0,74
Desv. estándar	0,51	0,42	0,47

Tabla B.2. Mediciones obtenidas durante la validación del sensado de velocidad angular para el experimento 2.

Experimento	PWM = 120		
	RPM Medido	RPM Referencia	Error (%)
1	37,64	38	0,95
2	37,79	38	0,55
3	38,20	38	0,53
4	38,55	39	1,15
5	38,53	39	1,21
6	38,36	38	0,95
7	38,05	38	0,13
8	37,63	38	0,97
9	37,46	37	1,24
10	37,38	38	1,63
Media	37,96	38,10	0,93
Desv. estándar	0,44	0,57	0,43

Tabla B.3. Mediciones obtenidas durante la validación del sensado de velocidad angular para el experimento 3.

Experimento	PWM = 200		
	RPM Medido	RPM Referencia	Error (%)
1	48,09	49	1,86
2	48,14	48	0,29
3	48,47	49	1,08
4	48,80	49	0,41
5	49,12	49	0,24
6	49,28	49	0,57
7	49,15	49	0,31
8	49,21	49	0,43
9	49,51	50	0,98
10	49,65	50	0,70
Media	48,94	49,10	0,69
Desv. estándar	0,55	0,57	0,50

Apéndice C

Tabla C.1. Cálculo de velocidad y error por cada tramo del caso 1.

Prueba	Cálculo	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	Promedio	MAPE (%)
1	Velocidad	22.22	18.18	18.18	16.67	20.00	20.00	18.18	20.00	18.18	18.18	20.00	20.00	20.00	19.07	6.33
	% Error	11.11	9.09	9.09	16.67	0.00	0.00	9.09	0.00	9.09	9.09	0.00	0.00	0.00	19.07	6.33
2	Velocidad	20.00	20.00	18.18	20.00	20.00	20.00	18.18	18.18	18.18	20.00	20.00	18.18	18.18	19.02	4.90
	% Error	0.00	0.00	9.09	0.00	0.00	0.00	9.09	9.09	9.09	0.00	0.00	9.09	9.09	19.02	4.90
3	Velocidad	22.22	18.18	20.00	18.18	20.00	18.18	18.18	18.18	20.00	20.00	18.18	18.18	20.00	19.19	5.75
	% Error	11.11	9.09	0.00	9.09	0.00	9.09	9.09	9.09	0.00	0.00	9.09	9.09	0.00	19.19	5.75
4	Velocidad	25.00	18.18	18.18	20.00	20.00	18.18	16.67	18.18	20.00	18.18	18.18	18.18	20.00	19.15	8.10
	% Error	25.00	9.09	9.09	0.00	0.00	9.09	16.67	9.09	0.00	9.09	9.09	9.09	0.00	19.15	8.10
5	Velocidad	25.00	18.18	20.00	18.18	18.18	18.18	18.18	20.00	20.00	18.18	18.18	20.00	18.18	19.26	7.52
	% Error	25.00	9.09	0.00	9.09	0.00	9.09	9.09	0.00	0.00	9.09	9.09	0.00	19.26	7.52	
6	Velocidad	25.00	20.00	18.18	18.18	18.18	18.18	20.00	16.67	18.18	18.18	18.18	20.00	15.38	18.79	9.88
	% Error	25.00	0.00	9.09	9.09	9.09	9.09	0.00	16.67	9.09	9.09	9.09	0.00	23.08	18.79	9.88
7	Velocidad	16.67	20.00	20.00	16.67	14.29	18.18	20.00	16.67	14.29	20.00	20.00	14.29	16.67	17.52	12.42
	% Error	16.67	0.00	0.00	16.67	28.57	9.09	0.00	16.67	28.57	0.00	0.00	28.57	16.67	17.52	12.42
8	Velocidad	20.00	20.00	20.00	18.18	20.00	20.00	20.00	16.67	15.38	20.00	20.00	18.18	15.38	18.40	8.01
	% Error	0.00	0.00	0.00	23.08	0.00	0.00	0.00	16.67	23.08	0.00	0.00	9.09	23.08	18.40	8.01
9	Velocidad	22.22	22.22	20.00	18.18	18.18	18.18	18.18	20.00	15.38	18.18	20.00	18.18	16.67	18.77	9.55
	% Error	11.11	11.11	0.00	16.67	16.67	9.09	0.00	0.00	23.08	9.09	0.00	9.09	16.67	18.77	9.55
10	Velocidad	22.22	20.00	20.00	18.18	15.38	20.00	20.00	16.67	15.38	20.00	20.00	16.67	16.67	18.55	8.95
	% Error	11.11	0.00	0.00	9.09	23.08	0.00	0.00	16.67	23.08	0.00	0.00	16.67	16.67	18.55	8.95
11	Velocidad	22.22	20.00	20.00	20.00	16.67	16.67	20.00	20.00	13.33	18.18	22.22	16.67	14.29	18.48	11.02
	% Error	11.11	0.00	0.00	0.00	16.67	16.67	0.00	0.00	33.33	9.09	11.11	16.67	28.57	18.48	11.02
12	Velocidad	22.22	18.18	20.00	20.00	15.38	18.18	20.00	18.18	15.38	18.18	22.22	18.18	15.38	18.58	10.53
	% Error	11.11	9.09	0.00	0.00	23.08	9.09	0.00	9.09	23.08	9.09	11.11	9.09	23.08	18.58	10.53
13	Velocidad	22.22	18.18	20.00	18.18	18.18	20.00	20.00	20.00	14.29	18.18	22.22	18.18	15.38	18.71	9.88
	% Error	11.11	9.09	0.00	9.09	9.09	0.00	0.00	0.00	28.57	9.09	11.11	9.09	23.08	18.71	9.88
14	Velocidad	20.00	20.00	20.00	18.18	18.18	18.18	18.18	20.00	16.67	20.00	20.00	18.18	15.38	18.69	6.55
	% Error	0.00	0.00	0.00	9.09	9.09	9.09	9.09	0.00	16.67	0.00	0.00	9.09	23.08	18.69	6.55
15	Velocidad	25.00	18.18	20.00	20.00	16.67	18.18	20.00	18.18	14.29	18.18	22.22	18.18	14.29	18.72	11.95
	% Error	25.00	9.09	0.00	0.00	16.67	9.09	0.00	9.09	28.57	9.09	11.11	9.09	28.57	18.72	11.95
16	Velocidad	22.22	20.00	20.00	18.18	15.38	20.00	20.00	18.18	13.33	20.00	22.22	15.38	14.29	18.40	11.42
	% Error	11.11	0.00	0.00	9.09	23.08	0.00	0.00	9.09	33.33	0.00	11.11	23.08	28.57	18.40	11.42
17	Velocidad	22.22	18.18	20.00	20.00	16.67	16.67	20.00	20.00	13.33	20.00	22.22	18.18	14.29	18.60	10.43
	% Error	11.11	9.09	0.00	0.00	16.67	16.67	0.00	0.00	33.33	0.00	11.11	9.09	28.57	18.60	10.43
18	Velocidad	25.00	18.18	20.00	20.00	16.67	18.18	20.00	18.18	14.29	18.18	22.22	18.18	15.38	18.80	11.53
	% Error	25.00	9.09	0.00	0.00	16.67	9.09	0.00	9.09	28.57	9.09	11.11	9.09	23.08	18.80	11.53
19	Velocidad	25.00	18.18	20.00	20.00	16.67	18.18	20.00	18.18	14.29	20.00	20.00	18.18	14.29	18.69	10.40
	% Error	25.00	9.09	0.00	0.00	16.67	9.09	0.00	9.09	28.57	0.00	0.00	9.09	28.57	18.69	10.40
20	Velocidad	22.22	20.00	20.00	18.18	16.67	18.18	20.00	18.18	14.29	20.00	22.22	18.18	14.29	18.65	10.18
	% Error	11.11	0.00	0.00	9.09	16.67	9.09	0.00	9.09	28.57	0.00	11.11	9.09	28.57	18.65	10.18
21	Velocidad	25.00	18.18	20.00	20.00	16.67	18.18	20.00	20.00	15.38	18.18	22.22	18.18	16.67	19.13	9.91
	% Error	25.00	9.09	0.00	0.00	16.67	9.09	0.00	0.00	23.08	9.09	11.11	9.09	16.67	19.13	9.91
22	Velocidad	25.00	20.00	20.00	18.18	18.18	18.18	20.00	20.00	15.38	18.18	22.22	18.18	15.38	19.14	9.82
	% Error	25.00	0.00	0.00	9.09	9.09	9.09	0.00	0.00	23.08	9.09	11.11	9.09	23.08	19.14	9.82
23	Velocidad	18.18	20.00	20.00	20.00	16.67	18.18	20.00	20.00	14.29	20.00	22.22	18.18	14.29	18.62	8.63
	% Error	9.09	0.00	0.00	0.00	16.67	9.09	0.00	0.00	28.57	0.00	11.11	9.09	28.57	18.62	8.63
24	Velocidad	20.00	20.00	20.00	20.00	18.18	18.18	18.18	20.00	18.18	18.18	20.00	20.00	18.18	19.16	4.20
	% Error	0.00	0.00	0.00	0.00	9.09	9.09	9.09	0.00	0.00	0.00	0.00	0.00	9.09	19.16	4.20
25	Velocidad	20.00	20.00	20.00	20.00	18.18	16.67	20.00	20.00	16.67	16.67	20.00	22.22	16.67	19.01	6.68
	% Error	0.00	0.00	0.00	0.00	9.09	16.67	0.00	0.00	16.67	16.67	0.00	11.11	16.67	19.01	6.68
26	Velocidad	18.18	20.00	20.00	20.00	15.38	18.18	20.00	20.00	15.38	20.00	20.00	18.18	15.38	18.51	7.42
	% Error	9.09	0.00	0.00	0.00	23.08	9.09	0.00	0.00	23.08	0.00	0.00	9.09	23.08	18.51	7.42
27	Velocidad	18.18	22.22	20.00	15.38	16.67	20.00	20.00	16.67	14.29	20.00	22.22	14.29	16.67	18.20	12.43
	% Error	9.09	11.11	0.00	23.08	16.67	0.00	0.00	16.67	28.57	0.00	11.11	28.57	16.67	18.20	12.43
28	Velocidad	20.00	18.18	20.00	18.18	18.18	18.18	20.00	20.00	15.38	18.18	22.22	18.18	14.29	18.32	10.10
	% Error	0.00	9.09	0.00	9.09	9.09	9.09	0.00	0.00	23.08	9.09	11.11	9.09	28.57	18.32	10.10
29	Velocidad	22.22	18.18	20.00	20.00	15.38	18.18	22.22	18.18	14.29	20.00	20.00	20.00	14.29	18.69	9.98
	% Error	11.11	9.09	0.00	0.00	23.08	9.09	11.11	9.09	28.57	0.00	0.00	0.00	28.57	18.69	9.98
30	Velocidad	20.00	20.00	20.00	18.18	15.38	20.00	20.00	18.18	13.33	20.00	22.22	16.67	14.29	18.33	10.07
	% Error	0.00	0.00	0.00	9.09	23.08	0.00	0.00	9.09	33.33	0.00	11.11	16.67	28.57	18.33	10.07

Tabla C.2. Cálculo de velocidad y error por cada tramo del caso 2.

Prueba	Cálculo	V1	V2	V3	V4	V5	V6	V7	V8	V9	Promedio	MAPE (%)
1	Velocidad	15.38	16.67	15.38	12.50	10.00	16.67	15.38	7.14	15.38	13.83	14.98
	% Error	2.56	11.11	2.56	16.67	33.33	11.11	2.56	52.38	2.56		
2	Velocidad	15.38	15.38	15.38	8.70	14.29	16.67	12.50	8.33	15.38	13.56	14.36
	% Error	2.56	2.56	2.56	42.03	4.76	11.11	16.67	44.44	2.56		
3	Velocidad	18.18	16.67	14.29	16.67	12.50	9.09	16.67	15.38	9.52	14.33	17.16
	% Error	21.21	11.11	4.76	11.11	16.67	39.39	11.11	2.56	36.51		
4	Velocidad	16.67	15.38	16.67	8.00	13.33	16.67	13.33	8.00	16.67	13.86	18.06
	% Error	11.11	2.56	11.11	46.67	11.11	11.11	11.11	46.67	11.11		
5	Velocidad	16.67	15.38	15.38	8.00	14.29	16.67	11.76	7.41	16.67	13.58	18.01
	% Error	11.11	2.56	2.56	46.67	4.76	11.11	21.57	50.62	11.11		
6	Velocidad	16.67	15.38	15.38	14.29	8.33	15.38	16.67	10.00	10.00	13.57	16.20
	% Error	11.11	2.56	2.56	4.76	44.44	2.56	11.11	33.33	33.33		
7	Velocidad	15.38	15.38	16.67	10.00	11.76	15.38	14.29	8.33	14.29	13.50	14.19
	% Error	2.56	2.56	11.11	33.33	21.57	2.56	4.76	44.44	4.76		
8	Velocidad	14.29	14.29	16.67	12.50	11.11	14.29	15.38	9.09	12.50	13.35	14.07
	% Error	4.76	4.76	11.11	16.67	25.93	4.76	2.56	39.39	16.67		
9	Velocidad	16.67	15.38	16.67	8.70	13.33	15.38	13.33	7.69	16.67	13.76	16.83
	% Error	11.11	2.56	11.11	42.03	11.11	2.56	11.11	48.72	11.11		
10	Velocidad	14.29	16.67	15.38	9.52	11.76	16.67	13.33	7.41	15.38	13.38	16.88
	% Error	4.76	11.11	2.56	36.51	21.57	11.11	11.11	50.62	2.56		
11	Velocidad	15.38	15.38	16.67	7.69	13.33	16.67	12.50	7.41	15.38	13.38	17.45
	% Error	2.56	2.56	11.11	48.72	11.11	11.11	16.67	50.62	2.56		
12	Velocidad	15.38	16.67	15.38	8.00	15.38	16.67	11.11	8.00	16.67	13.70	17.81
	% Error	2.56	11.11	2.56	46.67	2.56	11.11	25.93	46.67	11.11		
13	Velocidad	14.29	16.67	16.67	8.00	13.33	18.18	10.53	7.69	18.18	13.73	22.86
	% Error	4.76	11.11	11.11	46.67	11.11	21.21	29.82	48.72	21.21		
14	Velocidad	14.29	16.67	16.67	8.00	13.33	16.67	14.29	7.14	16.67	13.75	18.24
	% Error	4.76	11.11	11.11	46.67	11.11	11.11	4.76	52.38	11.11		
15	Velocidad	20.00	16.67	15.38	7.41	15.38	16.67	10.00	8.70	18.18	14.27	23.10
	% Error	33.33	11.11	2.56	50.62	2.56	11.11	33.33	42.03	21.21		
16	Velocidad	15.38	16.67	15.38	8.00	15.38	18.18	9.52	9.09	18.18	13.98	20.42
	% Error	2.56	11.11	2.56	46.67	2.56	21.21	36.51	39.39	21.21		
17	Velocidad	16.67	16.67	13.33	8.00	16.67	18.18	9.09	10.00	18.18	14.09	22.92
	% Error	11.11	11.11	11.11	46.67	11.11	21.21	39.39	33.33	21.21		
18	Velocidad	15.38	16.67	14.29	8.00	16.67	18.18	11.11	8.33	18.18	14.09	21.00
	% Error	2.56	11.11	4.76	46.67	11.11	21.21	25.93	44.44	21.21		
19	Velocidad	15.38	16.67	14.29	8.70	15.38	18.18	9.52	9.52	18.18	13.98	19.83
	% Error	2.56	11.11	4.76	42.03	2.56	21.21	36.51	36.51	21.21		
20	Velocidad	15.38	18.18	14.29	8.33	15.38	18.18	11.76	8.00	18.18	14.19	20.69
	% Error	2.56	21.21	4.76	44.44	2.56	21.21	21.57	46.67	21.21		
21	Velocidad	16.67	16.67	14.29	8.33	15.38	18.18	11.11	8.33	18.18	14.13	20.75
	% Error	11.11	11.11	4.76	44.44	2.56	21.21	25.93	44.44	21.21		
22	Velocidad	16.67	16.67	15.38	8.33	15.38	18.18	9.52	9.09	18.18	14.16	21.12
	% Error	11.11	11.11	2.56	44.44	2.56	21.21	36.51	39.39	21.21		
23	Velocidad	16.67	14.29	8.70	15.38	18.18	9.52	9.09	18.18	18.18	14.24	22.22
	% Error	11.11	4.76	42.03	2.56	21.21	36.51	39.39	21.21	21.21		
24	Velocidad	16.67	13.33	8.70	16.67	16.67	8.00	11.76	18.18	18.18	14.24	21.90
	% Error	11.11	11.11	42.03	11.11	11.11	46.67	21.57	21.21	21.21		
25	Velocidad	16.67	15.38	8.70	14.29	20.00	10.00	10.00	16.67	18.18	14.43	21.42
	% Error	11.11	2.56	42.03	4.76	33.33	33.33	33.33	11.11	21.21		
26	Velocidad	16.67	14.29	8.33	16.67	18.18	8.00	11.76	16.67	18.18	14.31	21.47
	% Error	11.11	4.76	44.44	11.11	21.21	46.67	21.57	11.11	21.21		
27	Velocidad	18.18	11.76	9.09	16.67	16.67	8.00	11.76	18.18	16.67	14.11	22.77
	% Error	21.21	21.57	39.39	11.11	11.11	46.67	21.57	21.21	11.11		
28	Velocidad	16.67	15.38	8.00	14.29	18.18	11.76	8.70	15.38	18.18	14.06	19.30
	% Error	11.11	2.56	46.67	4.76	21.21	21.57	42.03	2.56	21.21		
29	Velocidad	18.18	15.38	7.69	14.29	18.18	10.53	8.70	16.67	18.18	14.20	22.52
	% Error	21.21	2.56	48.72	4.76	21.21	29.82	42.03	11.11	21.21		
30	Velocidad	16.67	15.38	8.33	12.50	18.18	10.00	9.09	16.67	18.18	13.89	22.34
	% Error	11.11	2.56	44.44	16.67	21.21	33.33	39.39	11.11	21.21		

Tabla C.3. Cálculo de velocidad y error por cada tramo del caso 3.

Prueba	Calculo	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	Promedio	MAPE (%)	
		% Error	9.09	0.00	16.67	16.67	16.67	16.67	16.67	16.67	16.67	16.67	16.67	16.67			16.67
1	Velocidad	18.18	20.00	16.67	16.67	16.67	20.00	16.67	16.67	20.00	18.18	18.18	16.67	16.67	11.11	18.37	8.16
	% Error	16.67	9.09	18.18	18.18	18.18	20.00	16.67	0.00	16.67	18.18	18.18	16.67	0.00			
2	Velocidad	16.67	18.18	18.18	18.18	18.18	18.18	18.18	18.18	18.18	18.18	18.18	18.18	18.18	18.60	6.99	
	% Error	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	0.00			
3	Velocidad	18.18	18.18	18.18	18.18	18.18	18.18	18.18	18.18	18.18	18.18	18.18	18.18	18.18	18.60	6.99	
	% Error	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	0.00			
4	Velocidad	16.67	18.18	18.18	18.18	18.18	18.18	18.18	18.18	18.18	18.18	18.18	18.18	18.18	18.48	7.58	
	% Error	16.67	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	0.00			
5	Velocidad	18.18	18.18	16.67	16.67	18.18	18.18	18.18	18.18	16.67	16.67	18.18	16.67	16.67	17.74	11.31	
	% Error	9.09	9.09	16.67	16.67	9.09	9.09	9.09	9.09	16.67	16.67	9.09	16.67	0.00			
6	Velocidad	18.18	18.18	18.18	18.18	18.18	16.67	18.18	18.18	18.18	16.67	18.18	16.67	16.67	17.97	10.14	
	% Error	9.09	9.09	9.09	9.09	9.09	16.67	9.09	9.09	16.67	9.09	16.67	16.67	0.00			
7	Velocidad	20.00	18.18	16.67	16.67	18.18	18.18	18.18	16.67	18.18	18.18	18.18	16.67	16.67	18.00	10.02	
	% Error	0.00	9.09	16.67	16.67	9.09	9.09	9.09	16.67	9.09	9.09	9.09	16.67	0.00			
8	Velocidad	22.22	16.67	18.18	16.67	18.18	18.18	16.67	20.00	18.18	20.00	16.67	18.18	20.00	18.45	9.48	
	% Error	11.11	16.67	9.09	16.67	9.09	9.09	16.67	0.00	9.09	0.00	16.67	9.09	0.00			
9	Velocidad	18.18	18.18	16.67	18.18	18.18	16.67	18.18	20.00	16.67	20.00	18.18	16.67	20.00	18.14	9.32	
	% Error	9.09	9.09	16.67	9.09	9.09	16.67	9.09	0.00	16.67	0.00	9.09	16.67	0.00			
10	Velocidad	16.67	20.00	16.67	18.18	16.67	18.18	20.00	18.18	16.67	20.00	18.18	18.18	20.00	18.28	8.62	
	% Error	16.67	0.00	16.67	9.09	16.67	9.09	9.09	9.09	16.67	0.00	9.09	9.09	0.00			
11	Velocidad	20.00	18.18	18.18	16.67	20.00	18.18	18.18	18.18	16.67	20.00	18.18	18.18	20.00	18.62	6.88	
	% Error	0.00	9.09	9.09	16.67	0.00	9.09	9.09	9.09	16.67	0.00	9.09	9.09	0.00			
12	Velocidad	20.00	18.18	18.18	18.18	18.18	18.18	18.18	20.00	18.18	18.18	18.18	18.18	18.18	18.60	6.99	
	% Error	0.00	9.09	9.09	9.09	9.09	9.09	9.09	0.00	9.09	9.09	9.09	9.09	0.00			
13	Velocidad	18.18	18.18	18.18	18.18	18.18	20.00	18.18	18.18	18.18	18.18	18.18	16.67	20.00	18.48	7.58	
	% Error	9.09	9.09	9.09	9.09	9.09	0.00	9.09	9.09	9.09	9.09	9.09	16.67	0.00			
14	Velocidad	20.00	18.18	18.18	16.67	18.18	20.00	18.18	18.18	18.18	20.00	18.18	20.00	20.00	18.76	6.18	
	% Error	0.00	9.09	9.09	16.67	9.09	9.09	9.09	9.09	9.09	0.00	9.09	0.00	0.00			
15	Velocidad	16.67	18.18	18.18	18.18	20.00	18.18	18.18	18.18	16.67	22.22	18.18	16.67	20.00	18.56	8.90	
	% Error	16.67	9.09	9.09	9.09	0.00	9.09	9.09	9.09	16.67	11.11	9.09	16.67	0.00			
16	Velocidad	20.00	16.67	20.00	16.67	20.00	18.18	20.00	16.67	18.18	20.00	20.00	18.18	20.00	18.81	5.94	
	% Error	0.00	16.67	0.00	16.67	0.00	9.09	0.00	16.67	9.09	0.00	0.00	9.09	0.00			
17	Velocidad	18.18	18.18	18.18	18.18	18.18	20.00	18.18	18.18	18.18	20.00	18.18	20.00	18.18	18.60	6.99	
	% Error	9.09	9.09	9.09	9.09	9.09	0.00	9.09	9.09	9.09	0.00	9.09	0.00	9.09			
18	Velocidad	20.00	18.18	20.00	16.67	18.18	18.18	20.00	18.18	18.18	20.00	16.67	20.00	20.00	18.79	6.06	
	% Error	0.00	9.09	0.00	16.67	9.09	9.09	9.09	9.09	9.09	0.00	16.67	0.00	0.00			
19	Velocidad	20.00	18.18	18.18	18.18	20.00	18.18	18.18	20.00	18.18	20.00	18.18	18.18	20.00	18.88	5.59	
	% Error	0.00	9.09	9.09	9.09	0.00	9.09	9.09	0.00	9.09	0.00	9.09	9.09	0.00			
20	Velocidad	18.18	18.18	18.18	18.18	20.00	18.18	18.18	18.18	18.18	20.00	20.00	16.67	20.00	18.76	6.18	
	% Error	9.09	9.09	9.09	9.09	0.00	9.09	9.09	9.09	9.09	0.00	0.00	16.67	0.00			
21	Velocidad	18.18	18.18	20.00	18.18	20.00	18.18	18.18	18.18	18.18	20.00	18.18	18.18	20.00	18.74	6.29	
	% Error	9.09	9.09	0.00	9.09	0.00	9.09	9.09	9.09	9.09	0.00	9.09	9.09	0.00			
22	Velocidad	20.00	20.00	18.18	18.18	20.00	18.18	18.18	18.18	18.18	20.00	18.18	16.67	22.22	19.07	6.33	
	% Error	0.00	0.00	9.09	9.09	0.00	0.00	0.00	0.00	0.00	0.00	9.09	16.67	11.11			
23	Velocidad	18.18	18.18	18.18	16.67	18.18	16.67	16.67	16.67	18.18	18.18	20.00	18.18	20.00	18.00	10.02	
	% Error	9.09	9.09	9.09	16.67	9.09	16.67	16.67	16.67	9.09	9.09	0.00	9.09	0.00			
24	Velocidad	22.22	16.67	20.00	16.67	18.18	16.67	18.18	18.18	16.67	18.18	20.00	18.18	20.00	18.45	9.48	
	% Error	11.11	16.67	0.00	16.67	9.09	16.67	9.09	9.09	16.67	9.09	0.00	9.09	0.00			
25	Velocidad	16.67	18.18	18.18	16.67	18.18	16.67	18.18	18.18	16.67	18.18	18.18	18.18	20.00	17.86	10.72	
	% Error	16.67	9.09	9.09	16.67	9.09	16.67	9.09	9.09	16.67	9.09	9.09	9.09	0.00			
26	Velocidad	20.00	18.18	16.67	16.67	18.18	18.18	18.18	18.18	16.67	18.18	16.67	20.00	18.18	18.11	9.44	
	% Error	0.00	9.09	16.67	16.67	9.09	9.09	9.09	9.09	16.67	9.09	16.67	0.00	9.09			
27	Velocidad	18.18	20.00	16.67	18.18	18.18	16.67	18.18	18.18	20.00	16.67	18.18	18.18	18.18	18.11	9.44	
	% Error	9.09	0.00	16.67	9.09	9.09	16.67	9.09	9.09	0.00	16.67	9.09	9.09	9.09			
28	Velocidad	18.18	18.18	18.18	18.18	16.67	20.00	16.67	18.18	18.18	20.00	16.67	18.18	20.00	18.39	8.04	
	% Error	9.09	9.09	9.09	9.09	16.67	0.00	16.67	0.00	9.09	0.00	16.67	9.09	0.00			
29	Velocidad	18.18	18.18	16.67	16.67	18.18	18.18	18.18	18.18	16.67	18.18	16.67	18.18	16.67	17.60	12.00	
	% Error	9.09	9.09	16.67	16.67	9.09	9.09	9.09	9.09	16.67	9.09	16.67	9.09	16.67			
30	Velocidad	18.18	16.67	18.18	18.18	18.18	18.18	18.18	18.18	18.18	18.18	18.18	18.18	20.00	18.20	8.97	
	% Error	9.09	16.67	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	9.09	0.00			

Tabla C.4. Cálculo de velocidad y error por cada tramo del caso 4.

Prueba	Cálculo	V1	V2	V3	V4	V5	Promedio	MAPE (%)
1	Velocidad	15.38	10.53	8.33	8.70	11.11	10.81	19.99
	% Error	53.85	5.26	16.67	13.04	11.11		
2	Velocidad	11.11	9.52	8.70	8.70	9.52	9.51	9.34
	% Error	11.11	4.76	13.04	13.04	4.76		
3	Velocidad	11.76	8.70	8.33	10.53	9.52	9.77	11.48
	% Error	17.65	13.04	16.67	5.26	4.76		
4	Velocidad	11.11	8.70	8.33	10.53	10.00	9.73	9.22
	% Error	11.11	13.04	16.67	5.26	0.00		
5	Velocidad	11.76	8.70	8.70	9.09	10.00	9.65	10.56
	% Error	17.65	13.04	13.04	9.09	0.00		
6	Velocidad	10.53	8.70	8.70	9.52	11.11	9.71	9.44
	% Error	5.26	13.04	13.04	4.76	11.11		
7	Velocidad	11.76	8.70	8.70	9.52	10.53	9.84	10.75
	% Error	17.65	13.04	13.04	4.76	5.26		
8	Velocidad	11.76	8.70	8.33	9.52	11.11	9.88	12.65
	% Error	17.65	13.04	16.67	4.76	11.11		
9	Velocidad	10.53	9.52	9.09	9.09	9.52	9.55	6.59
	% Error	5.26	4.76	9.09	9.09	4.76		
10	Velocidad	13.33	10.00	9.09	9.52	9.52	10.29	10.39
	% Error	33.33	0.00	9.09	4.76	4.76		
11	Velocidad	11.11	9.52	9.52	8.70	9.09	9.59	8.55
	% Error	11.11	4.76	4.76	13.04	9.09		
12	Velocidad	10.53	10.00	9.09	9.52	9.09	9.65	5.64
	% Error	5.26	0.00	9.09	4.76	9.09		
13	Velocidad	11.11	9.52	9.52	9.52	10.00	9.93	5.08
	% Error	11.11	4.76	4.76	4.76	0.00		
14	Velocidad	11.76	10.53	9.09	10.00	10.00	10.28	6.40
	% Error	17.65	5.26	9.09	0.00	0.00		
15	Velocidad	11.76	10.53	10.00	10.00	10.00	10.46	4.58
	% Error	17.65	5.26	0.00	0.00	0.00		
16	Velocidad	11.76	10.53	9.52	10.53	10.00	10.47	6.59
	% Error	17.65	5.26	4.76	5.26	0.00		
17	Velocidad	11.76	9.52	10.53	10.00	9.52	10.27	6.49
	% Error	17.65	4.76	5.26	0.00	4.76		
18	Velocidad	11.76	10.53	9.52	9.09	10.00	10.18	7.35
	% Error	17.65	5.26	4.76	9.09	0.00		
19	Velocidad	11.11	10.00	10.00	10.00	10.00	10.22	2.22
	% Error	11.11	0.00	0.00	0.00	0.00		
20	Velocidad	11.11	10.53	10.00	9.09	10.00	10.15	5.09
	% Error	11.11	5.26	0.00	9.09	0.00		
21	Velocidad	12.50	10.00	10.00	10.00	10.00	10.50	5.00
	% Error	25.00	0.00	0.00	0.00	0.00		
22	Velocidad	11.76	10.00	10.53	10.00	10.00	10.46	4.58
	% Error	17.65	0.00	5.26	0.00	0.00		
23	Velocidad	11.76	10.53	10.00	9.52	11.11	10.58	7.76
	% Error	17.65	5.26	0.00	4.76	11.11		
24	Velocidad	11.11	10.53	9.52	9.52	9.52	10.04	6.13
	% Error	11.11	5.26	4.76	4.76	4.76		
25	Velocidad	11.76	11.11	9.52	9.52	10.53	10.49	8.71
	% Error	17.65	11.11	4.76	4.76	5.26		
26	Velocidad	11.76	9.52	10.00	10.00	10.00	10.26	4.48
	% Error	17.65	4.76	0.00	0.00	0.00		
27	Velocidad	11.11	9.52	10.00	9.09	10.00	9.94	4.99
	% Error	11.11	4.76	0.00	9.09	0.00		
28	Velocidad	10.53	9.52	9.52	8.70	9.52	9.56	6.52
	% Error	5.26	4.76	4.76	13.04	4.76		
29	Velocidad	11.11	10.00	10.53	10.00	10.53	10.43	4.33
	% Error	11.11	0.00	5.26	0.00	5.26		
30	Velocidad	11.76	10.53	10.00	9.52	10.00	10.36	5.53
	% Error	17.65	5.26	0.00	4.76	0.00		

Apéndice D

Código: https://github.com/ferdelbarco/PFG_ROVER_SETEC/tree/main

Apéndice E

Pruebas de validación: [https://drive.google.com/drive/folders/1JaTn-joxpukmnxN95za8BNVh6mU_Esus?usp=share link](https://drive.google.com/drive/folders/1JaTn-joxpukmnxN95za8BNVh6mU_Esus?usp=share_link)

Anexos

Anexo A. Lista completa de partes del Rover.

