



Universidad Tecnológica de La Habana  
José Antonio Echeverría

# Diseño de una plataforma robótica móvil con algunas características de un explorador espacial

**Autor: Carlos Raúl Morales Hernández**

Tutores:

**Dr. Valery Moreno Vega**

**Dra. Ivón O. Benítez González**

Trabajo de Diploma para optar por el título de Ingeniero en  
Automática

Facultad de Ingeniería Automática y Biomédica

La Habana, junio de 2018

## Declaración de autoría

Por medio de la presente se informa a los interesados que, Carlos Raúl Morales Hernández es el único autor del presente Trabajo de Diploma y autoriza al Departamento de Automática y Computación de La Universidad Tecnológica de la Habana “José Antonio Echeverría” a realizar con el mismo lo que estime más conveniente. Para que así conste, se firma la presente en La Habana, Cuba, a los 15 días del mes junio de 2018.

X 

---

Carlos Raúl Morales Hernández  
Autor

X 

---

Dr. Valery Moreno Vega  
Tutor

X 

---

Dra. Ivón Oristela Benítez  
Tutor

Tu tiempo es limitado, así que no lo gastes  
viviendo la vida de otros. No te dejes atrapar por  
el dogma, que es vivir según los resultados del  
pensamiento de los demás. No dejes que el ruido  
de las opiniones de los demás ahogue tu voz  
interior y más importante, ten el coraje de seguir  
tu corazón e intuición, de algún modo ellos saben  
lo que realmente quieres ser.

Todo lo demás es secundario.

**Stay hungry, stay foolish.**

**Steve Jobs**

# Resumen

El trabajo presenta el diseño de una plataforma robótica móvil con ruedas, basado en Raspberry Pi, con características de un robot para exploraciones espaciales. Se realiza un estudio de los principales componentes de los robots móviles. Se realiza el estudio de la configuración cinemática de los robots móviles espaciales. Se utiliza un sistema de suspensión para superar obstáculos. Se introducen los conceptos básicos sobre navegación en terrenos irregulares. Se realiza un estudio de las etapas de navegación autónoma en un *rover*. Se seleccionan los componentes del hardware de la plataforma y se exponen sus detalles y características físicas. Se explica el programa del controlador principal y del controlador de los motores. Se utiliza el algoritmo de Bellman-Ford para planificar una trayectoria libre de obstáculos y se muestra la ruta planificada en un mapa, representado mediante una descomposición en celdas. Se implementa un algoritmo de generación de trayectorias mediante curvas clotoideas para obtener un camino más suave, mostrando los resultados de las variaciones de orientación en la trayectoria y las curvas generadas entre los puntos planificados. Se implementa un controlador de seguimiento de trayectoria para el robot mediante el método de Orientación del Vector de Campo, aplicado a un robot con configuración diferencial, mostrándose la trayectoria real obtenida y el error con la de referencia. Finalmente se hace un análisis técnico-económico de la propuesta de diseño realizada para comprobar cuan factible es el trabajo de diploma.

# Abstract

The work presents the design of a mobile robot platform, based on Raspberry Pi, with characteristics of a robot for space exploration. A study of the main components of mobile robots is carried out. The kinematic configuration of planetary mobile robots is carried out. A suspension system to overcome obstacles is presented. Basic concepts of navigation in rough terrains are introduced. A study of the autonomous navigation stages of a planetary robot is carried out. The hardware components of the platform are selected and their details and physical characteristics are displayed. The main controller and motor driver programs are explained. The Bellman-Ford algorithm is used as path planning stage to provide a path free of obstacles. A trajectory generation algorithm is implemented using clothoid curves to obtain a smoother path, showing the results of the variations of orientation in the trajectory and the curves generated between the planned points. A trajectory tracking control is implemented for the robot using the Vector Field Orientation algorithm, showing the actual trajectory obtained and the error with the reference. Finally, an economic technical analysis is made of the design proposal made to see how feasible the work of diploma is.

# Agradecimientos

- A mis padres, Dilys y Raúl, por todo su apoyo, por su paciencia y su guía, sin las cuales no hubiera podido ingresar la Universidad al terminar el Tecnológico de Informática, por todos los sacrificios que han hecho para que pueda terminar la carrera y hacer de mí una persona de bien.
- A mis tíos Niurka y Orlando por sus cuidados, por ser tan buenos y divertidos, por ayudarme a relajarme en los momentos que lo necesitaba y por cuidar a mi mamá el tiempo en que no estuve.
- A mis amigos, el Flaco y Fito, por tantos momentos buenos, por creer en mí y por tantas conversaciones tratando de resolver el mundo.
- A mis primos Kamila, Roberto, Ernestico y Laura.
- A Víctor y Pepe, por todos los momentos felices, las risas, las burlas y el “chucho”, por todos los momentos difíciles que compartimos, por haberme graduado de bombero con ellos, las interminables horas de estudio y porque la carrera hubiera sido mucho más difícil (¿o fácil?) sin ellos.
- A mi genial novia Amanda, por creer en mí, ayudarme a esforzarme cada día más y por vivir conmigo los momentos más duros de la carrera, siempre dándome esperanza, ayudándome a resolver los problemas y haciéndome reír.
- A Susset y Ale por su tutoría y ayuda durante la tesis, tenerme de huésped por tres meses, ayudándome en todo lo que necesitaba y dándome los medios para realizar la investigación y permitiéndome abstraerme de muchas responsabilidades.
- A todos los profesores de que he tenido, en especial, a Mirtha mi profesora de secundaria, Ivón, Garcés, Carmen y Gilda.
- A Garcés, Ivón, Gilda, Carmen, Susset y Ale por ayudarme durante el problemón de 3er año.
- A mis tutores, Valery e Ivón por su guía durante la tesis.
- A David, por mostrarme la carrera de Automática y a todos los compañeros del aula.
- A todos los que faltan que han sido parte importante de mi vida.

# Índice General

ÍNDICE DE FIGURAS .....	IX
ÍNDICE DE TABLAS .....	X
GLOSARIO DE TÉRMINOS .....	XI
INTRODUCCIÓN .....	XIV
<b>CAPÍTULO 1. REVISIÓN BIBLIOGRÁFICA SOBRE ROBÓTICA MÓVIL Y ESPACIAL Y LOS ESTÁNDARES DE LA ECSS .....</b>	<b>1</b>
1.1. INTRODUCCIÓN .....	1
1.2. ANTECEDENTES HISTÓRICOS Y MOTIVACIÓN .....	1
1.3. APLICACIONES ACTUALES DE LA ROBÓTICA .....	2
1.4. TIPOS DE ROBOTS .....	3
1.5. SISTEMA DE LOCOMOCIÓN .....	3
1.5.1. Configuraciones de las ruedas .....	4
1.5.2. Configuraciones cinemáticas .....	5
1.5.3. Sistema de suspensión Rocker Bogie .....	7
1.6. CONTROLADOR, MOTORES Y SENSORES .....	8
1.6.1. Motores .....	8
1.6.2. Sensores .....	11
1.6.3. Controladores .....	14
1.7. SISTEMA DE GENERACIÓN Y ALMACENAMIENTO DE ENERGÍA .....	16
1.7.1. Panel Solar .....	17
1.7.2. Batería .....	17
1.7.3. Generador termoeléctrico de radioisótopos .....	19
1.8. SISTEMA DE NAVEGACIÓN AUTÓNOMA Y CONTROL DEL VEHÍCULO .....	19
1.8.1. Localización y mapeo .....	20
1.8.2. Planificación de trayectoria .....	20
1.8.3. Generación de trayectoria .....	24
1.8.4. Control de seguimiento de trayectoria .....	26
1.8.5. Control en lazo cerrado .....	28
1.9. ALGUNAS PLATAFORMAS ROBÓTICAS UTILIZADAS PARA LA EXPLORACIÓN ESPACIAL ....	29
1.10. AUTONOMÍA Y ESTÁNDARES DEL ECSS .....	31
1.11. CONCLUSIONES .....	33
<b>CAPÍTULO 2. DISEÑO DE LA PLATAFORMA .....</b>	<b>34</b>

2.1.	INTRODUCCIÓN.....	34
2.2.	LOCOMOCIÓN .....	34
2.2.1.	<i>Dimensiones</i> .....	35
2.3.	MOTORES .....	35
2.3.1.	<i>Controlador de motores</i> .....	36
2.4.	SENSORES .....	37
2.4.1.	<i>Encoder</i> .....	37
2.4.2.	<i>Sense-Hat</i> .....	38
2.5.	CONTROLADOR PRINCIPAL Y COMUNICACIÓN .....	39
2.6.	SISTEMA DE GENERACIÓN Y ALMACENAMIENTO DE ENERGÍA .....	40
2.6.1.	<i>Consumo de energía</i> .....	41
2.6.2.	<i>Almacenamiento de energía</i> .....	42
2.6.3.	<i>Generación de energía</i> .....	42
2.6.4.	<i>Esquema general</i> .....	43
2.7.	PROGRAMA .....	45
2.8.	CONCLUSIONES .....	49
<b>CAPÍTULO 3.</b>	<b>ALGORITMOS DE NAVEGACIÓN .....</b>	<b>50</b>
3.1.	INTRODUCCIÓN.....	50
3.2.	CONSIDERACIONES PREVIAS .....	50
3.3.	PLANIFICACIÓN DE TRAYECTORIA.....	51
3.3.1.	<i>Algoritmo de Bellman-Ford</i> .....	52
3.4.	GENERACIÓN DE TRAYECTORIA .....	56
3.4.1.	<i>Curvas clotoideas</i> .....	56
3.4.2.	<i>Algoritmo de generación</i> .....	61
3.5.	SEGUIMIENTO DE TRAYECTORIA.....	65
3.6.	CONCLUSIONES .....	71
<b>CAPÍTULO 4.</b>	<b>ANÁLISIS TÉCNICO-ECONÓMICO.....</b>	<b>73</b>
4.1.	INTRODUCCIÓN.....	73
4.2.	COSTO DE LA ELABORACIÓN DEL PROYECTO .....	73
4.2.1.	<i>Personas involucradas en realización del proyecto</i> .....	74
4.2.2.	<i>Otros gastos</i> .....	75
4.2.3.	<i>Gastos por medios o materiales directos</i> .....	75
4.2.4.	<i>Costo indirecto, directo y total</i> .....	76
4.3.	PRECIO.....	76
4.4.	ANÁLISIS CUANTITATIVO Y CUALITATIVO DEL EFECTO ECONÓMICO COMO RESULTADO DE LA IMPLEMENTACIÓN DEL TRABAJO DE DIPLOMA .....	77



<b>CONCLUSIONES .....</b>	<b>78</b>
<b>RECOMENDACIONES .....</b>	<b>80</b>
<b>BIBLIOGRAFÍA .....</b>	<b>81</b>

# Índice de figuras

FIGURA 1.1: TIPOS DE RUEDAS. (A) RUEDA FIJA, (B) RUEDA ESTÁNDAR, (C) RUEDA LOCA, (D) RUEDA SUECA.	5
FIGURA 1.2: CONFIGURACIÓN DOBLE DE ACKERMAN EN UN ROBOT DE SEIS RUEDAS. (A) GIRO EN EL LUGAR Y (B) MOVIMIENTO HACIA LA DERECHA.	6
FIGURA 1.3: CONFIGURACIÓN DIFERENCIAL Y SUS MOVIMIENTOS.	6
FIGURA 1.4: SUSPENSIÓN ROCKER BOGIE EN MOVIMIENTO.	7
FIGURA 1.5: FORMA DE ONDA Y POSICIÓN DEL EJE DE UN SERVOMOTOR.	9
FIGURA 1.6: MODULACIÓN POR ANCHO DE PULSO.	10
FIGURA 1.7: CONFIGURACIONES POSIBLES DE UN PUENTE H.	10
FIGURA 1.8: FUNCIONAMIENTO DE UN <i>ENCODER</i> DE DOS CANALES. (A) GIRO A LA DERECHA. (B) GIRO A LA IZQUIERDA.	11
FIGURA 1.9: UNIDAD DE MEDICIÓN INERCIAL.	12
FIGURA 1.10: DISTRIBUCIÓN DE COMPONENTES DEL RASPBERRY PI 3.	15
FIGURA 1.11: ARDUINO MEGA2560.	16
FIGURA 1.12: <i>ROVER</i> EN SU POSICIÓN NORMAL Y EN UN TERRENO CON IRREGULARIDADES.	21
FIGURA 1.13: MAPA DEL ENTORNO LUEGO DE REALIZAR LA EVALUACIÓN DEL ENTORNO.	22
FIGURA 1.14: CAMINO PLANIFICADO EN UN ENTORNO DESCOMPUESTO EN CELDAS MEDIANTE EL A*.	23
FIGURA 1.15: MODELO ACKERMAN SIMPLIFICADO AL MODELO BICICLETA.	24
FIGURA 1.16: COMPARACIÓN ENTRE UN CAMINO PLANIFICADO Y UNO GENERADO MEDIANTE CURVAS CLOTOIDES.	25
FIGURA 1.17: MODELO CINEMÁTICO DEL ROBOT. (A) DIFERENCIAL. (B) UNICICLO.	27
FIGURA 1.18: <i>MARS EXPLORATION ROVERS</i> .	31
FIGURA 2.1: ESTRUCTURA DEL ROBOT.	34
FIGURA 2.2: DIMENSIONES DE LA PLATAFORMA.	35
FIGURA 2.3: MOTOR DE CC UTILIZADO PARA EL MOVIMIENTO DE LAS RUEDAS.	35
FIGURA 2.4: CONTROLADOR DE MOTORES TB6612.	36
FIGURA 2.5: <i>ENCODER</i> DE CUADRATURA.	37
FIGURA 2.6: SENSE-HAT.	38
FIGURA 2.7: BATERÍA DEL ROBOT.	42
FIGURA 2.8: PANEL SOLAR CON REGULADOR INTERNO.	43
FIGURA 2.9: DIAGRAMA GENERAL DEL SISTEMA.	44
FIGURA 2.10: ESQUEMA DE CONTROL DE VELOCIDAD.	46
FIGURA 2.11: DIAGRAMA DE FLUJO DEL PROGRAMA EN EL ARDUINO.	47
FIGURA 2.12: DIAGRAMA DE FLUJO DEL PROGRAMA EN LA RASPBERRY PI.	48
FIGURA 3.1: ETAPAS DEL PROCESO DE NAVEGACIÓN.	50
FIGURA 3.2: MAPA DE ESTADOS.	51
FIGURA 3.3: MOVIMIENTOS POSIBLES EN UN ENTORNO DESCOMPUESTO EN CELDAS CUADRADAS.	54
FIGURA 3.4: TRAYECTORIA PLANIFICADA.	55
FIGURA 3.5: COMPARACIÓN DE LA CURVATURA DE UNA CLOTOIDE Y UN ARCO CIRCULAR.	57
FIGURA 3.6: CONFIGURACIONES SIMÉTRICAS.	57
FIGURA 3.7: CAMINOS BÁSICOS EN UNA DESCOMPOSICIÓN EN CELDAS CUADRADAS.	60

FIGURA 3.8: DIAGRAMA DE FLUJO DEL ALGORITMO DE GENERACIÓN.....	62
FIGURA 3.9: TRAYECTORIA PLANIFICADA Y GENERADA.....	63
FIGURA 3.10: VARIACIÓN DE ORIENTACIONES DE REFERENCIA DE LA TRAYECTORIA GENERADA. ....	64
FIGURA 3.11: SEGMENTO DE VARIACIONES CON LAS CURVAS. ....	64
FIGURA 3.12: TRAYECTORIA GENERADA. ....	65
FIGURA 3.13: SEGMENTO DE INICIO DE LA TRAYECTORIA REAL Y LA DE REFERENCIA. ....	67
FIGURA 3.14: COMPORTAMIENTO DEL ERROR DEL SEGUIMIENTO DE TRAYECTORIA EN EL EJE X. ....	68
FIGURA 3.15: COMPORTAMIENTO DEL ERROR DEL SEGUIMIENTO DE TRAYECTORIA EN EL EJE Y. ....	68
FIGURA 3.16: ACERCAMIENTO DEL CAMINO EN LA CURVA C2. ....	69
FIGURA 3.17: SEGUIMIENTO DE LA VARIACIÓN DE LA ORIENTACIÓN DE REFERENCIA. ....	70
FIGURA 3.18: ACERCAMIENTO DEL SEGUIMIENTO DE LA ORIENTACIÓN DE REFERENCIA.....	70
FIGURA 3.19: TRAYECTORIA REAL A PARTIR DE LA DE REFERENCIA. ....	71

## Índice de Tablas

TABLA 2.1: ESPECIFICACIONES DEL MOTOR UTILIZADO. ....	36
TABLA 2.2: ESPECIFICACIONES DEL <i>ENCODER</i> .....	38
TABLA 2.3: CARACTERÍSTICAS DEL RASPBERRY PI 3 MODELO B. ....	39
TABLA 2.4: CONSUMO DE POTENCIA DE LOS COMPONENTES ....	41
TABLA 2.5: CARACTERÍSTICAS DEL PANEL SOLAR .....	43
TABLA 3.1: PARÁMETROS PLANTILLA DE LAS CURVAS BÁSICAS.....	61
TABLA 4.1: GASTOS POR MATERIALES DIRECTOS .....	76

# Glosario de términos

- A

**ARM** (*Advanced RISC Machine*) es una arquitectura RISC de 32 bits, actualmente también con 64 bits desarrollada por ARM Holdings.

- C

**CCD** (*Charged-coupled Device*) dispositivo de carga acoplada. Es conocido popularmente como la designación de uno de los elementos principales de las cámaras fotográficas y de video digitales. El CCD es el sensor con diminutas células fotoeléctricas que registran la imagen.

- I

**IDE** (*Integrated Development Environment*) Entorno de desarrollo integrado. Es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

**ISO** (*International Organization for Standardization*) Organización Internacional para la Estandarización. Es una organización para la creación de estándares internacionales compuesta por diversas organizaciones nacionales de estandarización.

**I2C** (*Inter-Integrated Circuit*) Circuito inter-integrado. Es un bus de dos líneas que fue creado en 1980 por Philips para interconectar los microprocesadores o microcontroladores a periféricos más lentos. Un dispositivo máster, como la Raspberry Pi, controla el bus y otros dispositivos esclavos pueden ser direccionados.

- M

**MER** (*Mars Exploration Rover*) Rover de exploración marciana. Fue una misión de la NASA con el objetivo científico de analizar el suelo y buscar pruebas de presencia de agua utilizando dos robots llamados Spirit y Opportunity.

**MPPT** (*Max Power Point Tracking*) Seguimiento del punto de máxima potencia. Es un algoritmo que se utiliza para lograr el punto de máxima corriente en un panel solar bajo las condiciones en que se encuentra.

- P

**PWM** (*Pulse Width Modulation*) Modulación por ancho de pulsos. Es una técnica en la que se modifica el ciclo de trabajo de una señal periódica para controlar la cantidad de energía que se envía a una carga.

**PID** (*Proportional Integrative Derivative*). Es un mecanismo de control por realimentación que calcula el error entre un valor medido y un valor deseado. Consiste en un parámetro proporcional, uno integral y uno derivativo.

- R

**Rover**. También conocido como astromóvil, es un vehículo de exploración espacial diseñado para moverse sobre la superficie de un planeta u otro objeto astronómico.

**RPM** Revoluciones por minuto. Indica el número de rotaciones completadas cada minuto por un cuerpo que gira alrededor de un eje.

**RTG** (*Radioisotope thermoelectric generator*) Generador termoeléctrico de radioisótopos. Es un generador eléctrico simple que obtiene su energía de la liberada por la desintegración de determinados elementos radiactivos.

- S

**SBC** (*Single Board Computer*) es una computadora completa en un sólo circuito. El diseño se centra en un sólo microprocesador con la RAM, E/S y todas las demás características de un computador funcional en una sola tarjeta que suele ser de tamaño reducido, y que tiene todo lo que necesita en la placa base.

**SLAM** (*Self-Localization and Mapping*) localización y mapeo simultáneos. Es una técnica usada por vehículos autónomos para construir un mapa del entorno en que se encuentra, a la vez que estima su posición al desplazarse dentro de este entorno.

**SSH** (*Secure Shell*). Es el nombre de un protocolo y del programa que lo implementa, sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos.

- U

**UART** (*Universal Asynchronous Receiver-Transmitter*) Transmisor-Receptor Asíncrono Universal. Es el dispositivo que controla los puertos y dispositivos serie. Se encuentra integrado en la placa base o en la tarjeta adaptadora del dispositivo.

**USB** (*Universal Serial Bus*) Bus Universal Serie. Es un bus estándar industrial que define los cables, conectores y protocolos usados en un bus para

conectar, comunicar y proveer de alimentación eléctrica entre computadoras, periféricos y dispositivos electrónicos.

- V

**VFO** (*Vector Field Orientation*) Orientación del vector de campo. Permite realizar el control de seguimiento de trayectoria, para un robot tipo diferencial, proporcionando un vector de control que garantiza la convergencia del error.

- W

**WiFi**. Es un mecanismo de conexión de dispositivos electrónicos de forma inalámbrica. Los dispositivos habilitados con wifi pueden conectarse a internet a través de un punto de acceso de red inalámbrica.

# Introducción

La idea de crear máquinas inteligentes ha estado en la humanidad desde hace miles de años. Desde las primeras civilizaciones, una de las mayores ambiciones del hombre ha sido crear artefactos a su imagen y semejanza. La leyenda del titán Prometeo, quien moldeó a la humanidad de la arcilla; o la del gigante Talos, el esclavo de bronce forjado por Hefesto (3500 AC); las estatuas de los oráculos de los egipcios, que ocultaban a los sacerdotes dentro (2500 AC); la Clepsidra, un reloj de agua introducido por los babilonios (1400), el cual fue uno de los primeros artefactos mecánicos automatizados [1]. En los siglos siguientes la creatividad humana dio lugar a una gran cantidad de dispositivos. El concepto de robot fue claramente establecido por aquellas muchas realizaciones históricas creativas. No obstante, la aparición de los robots físicos no fue hasta que llegaron los avances científicos del siglo XX.

El término robot, derivado de la palabra “robota”, que significa “trabajo forzado” u “obligatorio” en las lenguas eslavas, fue utilizado por primera vez en 1921 por el checo Karel Čapek, en su obra *Robots Universales Rossum (Rossum's Universal Robots)* [2]. Los primeros robots se realizaron a mediados del siglo veinte, beneficiándose de los avances en mecánica, control y electrónica del momento. Nuevos diseños dieron paso a nuevos problemas y a nuevas soluciones, aumentando el conocimiento y dando lugar al campo de la robótica, referido en aquel entonces como la ciencia y tecnología de los robots.

En 1945, Isaac Asimov introdujo las tres leyes de interacción entre robots y seres humanos. Inspirado por los libros de este escritor, Joseph H. Engelberger intentó realizar un robot trabajador en la década de los cincuenta, que no fue instalado hasta 1961 en la *General Motors*. Este robot era un brazo robótico con cinco grados de libertad (*Degrees of freedom: DOF*) [2]. Desde entonces, los robots industriales han evolucionado asombrosamente, tomando posiciones en un sinnúmero de aplicaciones industriales, desde su uso en soldadura, pintura, ensamblaje y en muchos procesos hostiles o que requieren de gran precisión, velocidad y repetitividad.

En 1980, la robótica fue definida como la ciencia que estudia la conexión inteligente entre percepción y acción [1]. En esta definición, acción se refiere al movimiento del robot y a los sistemas de manipulación, percepción se refiere a la información extraída de los sensores acerca del estado del robot y del entorno, y la conexión inteligente es la programación, planeo y control, basadas en la percepción y modelos del robot.

A medida que las tareas en las que serían utilizados los robots fueron complejizándose, también se volvió más complejo su diseño y fueron surgiendo los robots con cierta autonomía, estos no solo realizan tareas repetitivas, como los robots industriales, sino que pueden seleccionar un curso de acción en función de lo que perciben de su entorno y su situación actual, de ahí la inteligencia. La autonomía en robótica significa que un robot puede adaptarse a cambios en el entorno o en sí mismo y continuar hacia su objetivo [3]. Entre estos robots no dedicados a aplicaciones industriales, destacan los robots espaciales, robots para aplicaciones submarinas y subterráneas, robots militares, robots con aplicaciones médicas, entre otros.

La robótica autónoma juega un papel vital en las exploraciones espaciales. Los robots se emplean para tareas en el espacio que los seres humanos son incapaces de hacer debido a las condiciones ambientales a las que tienen que enfrentarse. Actualmente, se encuentran realizando misiones de mantenimiento, construcción de hardware y exploración y recolección de muestras para investigaciones científicas, en diferentes cuerpos celestes. Estos agentes mecánicos necesitan tener cierta libertad o capacidad de decisión propia para cumplir las misiones y la habilidad de navegar sin intervención del ser humano.

## **Problema a resolver**

Se desea contar con una plataforma robótica móvil con características similares a las de un robot para exploraciones espaciales.

## **Hipótesis**

Si se diseña un *rover*, basado en Raspberry Pi, con cierto grado de autonomía en la navegación, se mejorarán las plataformas robóticas desarrolladas en la universidad y se podrán aplicar en diversos contextos de una manera más eficiente.

## **Objeto de estudio**

Robots móviles con ruedas dedicados a la exploración espacial.

## **Campo de acción**

Plataforma robótica con sistema de suspensión, movimiento diferencial, subsistema de potencia basado en energía solar y una o más ramas y niveles de autonomía.

## **Objetivo General**

Diseñar una plataforma robótica móvil para la exploración espacial con cierto grado de autonomía en la navegación.



## Objetivos Específicos

- Analizar las características de las plataformas robóticas para exploraciones espaciales.
- Proponer un diseño del hardware y el software de un prototipo de un *rover* para exploración espacial.
- Incluir en el diseño cierto grado de autonomía al desarrollar algoritmos de navegación autónoma.
- Diseñar un sistema de energía que permita la autonomía del robot según lo establecido por la ECSS.

## Estructuración del contenido

- **Capítulo 1:** Se expone el estado actual de la robótica espacial, se realiza un estudio sobre los principales componentes de un *rover* y un robot móvil, se describen varios modelos y se explican las etapas de la navegación autónoma. Además, se explicarán los estándares de la ECSS.
- **Capítulo 2:** Se realiza el diseño de la plataforma, la selección y descripción de los componentes de hardware utilizados, así como el programa de cada uno de los controladores.
- **Capítulo 3:** Se implementa un algoritmo de planificación de trayectoria, generación de curvas y control de seguimiento de trayectoria y se exponen los resultados mediante simulaciones.
- **Capítulo 4:** Análisis técnico-económico.

## Aportes Prácticos esperados

Contar con una plataforma robótica móvil con algunas características de un agente robótico espacial.

# **Capítulo 1. Revisión Bibliográfica sobre robótica móvil y espacial y los estándares de la ECSS**

## **1.1. Introducción**

Este capítulo aborda los conceptos fundamentales sobre robótica móvil para aplicaciones espaciales, las diferentes configuraciones cinemáticas de los robots y sus principales componentes hardware. Se describen tres modelos cinemáticos de robots móviles y los conceptos básicos sobre navegación autónoma. Se introducen los diferentes tipos de robots espaciales, clasificados según los estándares de la ECSS en cuanto a autonomía; además, se realiza una revisión histórica de las principales plataformas robóticas espaciales que han existido y sus características.

## **1.2. Antecedentes históricos y motivación**

En las décadas de 1950 y 1960 se realizaron una serie de exploraciones que se convirtieron en la base de la robótica móvil actual. En aquel entonces, todos los esfuerzos se concentraban en enviar al hombre al espacio exterior, sin embargo, la utilización de robots fue de suma importancia para comprender el entorno al que se iban a enfrentar los astronautas. En todas las misiones exitosas se encuentran, jugando un rol significativo, los vehículos de exploración espacial conocidos como *rovers* (astromóvil), brazos manipuladores, recogedores de muestras de superficies y taladros robóticos [4]. Los objetivos de las misiones son cada vez más complicados y, por tanto, se necesita de plataformas más avanzadas y con más independencia. Actualmente, las agencias espaciales de varios países, realizan misiones en cuerpos celestes con el objetivo de encontrar rastros de vida o condiciones para esta.

En el Departamento de Automática y Computación de la Facultad de Automática y Biomédica de la CUJAE se han realizado varias investigaciones sobre robótica móvil, teniendo resultados en la detección y evasión de obstáculos [5], movimiento con accionamiento diferencial [6] y planificación y generación de trayectorias [7-9].

El plantearse un escenario como el descrito anteriormente, no solo puede ayudar en los contextos típicos de misiones espaciales, sino que muchas de las técnicas y conocimientos que se obtienen de este tipo de diseños son posteriormente aplicables a otras esferas y sectores de la vida, por lo que desarrollar un trabajo de este tipo

puede contribuir a generar líneas de desarrollo aplicables a diferentes situaciones (autonomía de movimientos, fuentes alternativas de energía, movimiento por terrenos irregulares, etc.) en el contexto cubano.

### 1.3. Aplicaciones actuales de la robótica

El término robot es definido de manera formal por la Organización Internacional para la Estandarización (*International Organization for Standardization: ISO*), en la norma ISO 8373:2012, como un mecanismo programable accionado, en dos o más ejes, con un grado de autonomía, moviéndose dentro de su entorno, para realizar las tareas previstas. Existen otras definiciones dadas por asociaciones, por ejemplo, el Instituto de Robótica de América (*Robotics Institute of America: RIA*) y la Asociación Británica de Robots (*British Robot Association: BRA*), pero todas concuerdan en dos aspectos, la capacidad de programación y la multifuncionalidad.

Los robots se pueden usar para casi cualquier aplicación que se pueda imaginar, más específicamente, pudiera decirse que son adecuados para aplicaciones donde:

- 1) **La naturaleza económica o servil de la aplicación resulta en el uso ineficiente de trabajadores humanos** (industria de servicios, agricultura). El uso más obvio de los robots es en la fabricación, donde las actividades repetitivas en entornos desagradables hacen a los trabajadores humanos ineficientes o caros de retener. Una razón por la cual la soldadura ahora es en gran medida robótica, es debido a que es un trabajo desagradable para un humano (caliente, sudoroso y tedioso) con una baja tolerancia a la inexactitud.
- 2) **Para usos humanitarios donde existe un gran riesgo** (desminado de un área de minas terrestres, búsqueda y rescate urbano). Este es uno de los usos más convincentes. Recientemente, se han propuesto robots para ayudar a detectar minas terrestres sin explotar y para realizar búsqueda y el rescate urbano (búsqueda de sobrevivientes después de un bombardeo terrorista a un edificio o un terremoto).
- 3) **Tareas en las que un humano tiene un riesgo significativo** (nuclear, espacial, militar). Una tarea que plantea un riesgo significativo para un ser humano es la exploración espacial. Las personas pueden protegerse en el espacio del vacío, la radiación solar, etc., pero solo a un gran costo económico [3].

## 1.4. Tipos de robots

En la actualidad existe una gran cantidad de tipos de robots, debido al gran desarrollo tecnológico, entre los que se pueden encontrar: aviones autónomos, vehículos terrestres autónomos, humanoides, colaborativos, tipo *rover*, bípedos, con forma de animales, entre otros. Existen varias formas de clasificarlos:

- **Androides:** son robots que parecen y actúan como los humanos. Tratan de imitar el comportamiento del hombre. Se utilizan solo para la experimentación, ya que no cumplen ninguna labor.
- **Zoomórficos:** su sistema de locomoción imita al de los animales. Son utilizados en aplicaciones de terrenos accidentados como volcanes y exploración de planetas, ya que son hábiles para sortear obstáculos. Pueden ser caminadores y no caminadores.
- **Médicos:** son prótesis para discapacitados médicos que se adaptan al cuerpo y están dotados de potentes sistemas de mando. Con ellos se logra imitar de alguna forma los movimientos y funciones de los órganos o extremidades que sustituyen.
- **Industriales:** generalmente están dirigidos a las actividades “3D” sucias (Dirty), aburridas (Dull) y peligrosas (Dangerous). Esta idea se originó del concepto japonés “3K” (kitanai, kiken, kitsui), el cual describe las áreas donde los robots deben operar para aliviar el esfuerzo humano [4], por lo tanto, este tipo está diseñado para realizar operaciones precisas y repetitivas, durante largos intervalos de tiempo, en ambientes donde se espera un bajo desempeño humano. Comprenden más de dos mil millones de dólares de la industria mundial [3]. Actualmente, la mayor cantidad de estos está en Corea del Sur.
- **Móviles:** poseen una gran capacidad de desplazamiento y un alto grado de inteligencia. Son provistos de patas, ruedas u orugas que les permitan el desplazamiento de acuerdo a su programación. Elaboran la información que reciben a través de su propio sistema de sensores. Se utilizan para la exploración en lugares complejos, de difícil acceso o muy distantes, como es el caso de la exploración espacial.

## 1.5. Sistema de locomoción

Un robot móvil necesita de un sistema de locomoción que le habilite el movimiento a través de su entorno. Existen robots capaces de saltar, caminar, correr, deslizarse, nadar, volar y rodar, casi todos los mecanismos están inspirados por su contraparte

biológica, con excepción de los basados en ruedas o esteras [10]. La mayoría de los exploradores espaciales son vehículos con ruedas debido a la simplicidad, fiabilidad y fácil control, aunque han existido todo tipo de diseños.

Se puede definir un robot móvil con ruedas (RMR) como un sistema electromecánico controlado, que utiliza como locomoción ruedas de algún tipo y que es capaz de trasladarse de forma autónoma a una meta preestablecida en un determinado espacio de trabajo [11]. Poseen diferentes configuraciones cinemáticas y en el caso de los robots espaciales un sistema de suspensión que les permita movimiento por terrenos difíciles.

### 1.5.1. Configuraciones de las ruedas

Existen varios tipos de ruedas, como se muestra en la Figura 1.1. Difieren grandemente en la cinemática, por lo que la selección de un tipo definirá la configuración cinemática del robot.

**Rueda fija:** esta rueda puede rotar sobre un eje que pasa por su centro. Este eje es ortogonal al plano de la rueda. La orientación de la rueda respecto al chasis es constante [6], como se muestra en la Figura 1.1 a.

**Rueda estándar:** es aquella en la que el movimiento del plano de la rueda con respecto a la estructura es una rotación alrededor de un eje vertical que pasa a través del centro de la rueda, Figura 1.1 b. Suele cumplir funciones como rueda de dirección o como rueda de tracción-dirección [5].

**Rueda loca o pivotante:** también conocida como rueda castor, es una rueda orientable con respecto a la estructura, tal que la rotación del plano de la rueda es alrededor de un eje vertical que no pasa a través del centro de la rueda, ver Figura 1.1 c, Suele cumplir funciones como rueda de dirección o como rueda de tracción-dirección [5].

**Rueda sueca:** está menos restringida en cuanto a dirección que la rueda estándar convencional, la principal ventaja es que, aunque la rotación de la rueda solo funciona a lo largo del eje principal, la rueda puede moverse cinemáticamente con muy poca fricción a lo largo de muchas trayectorias posibles, no solo hacia adelante y hacia atrás [12].

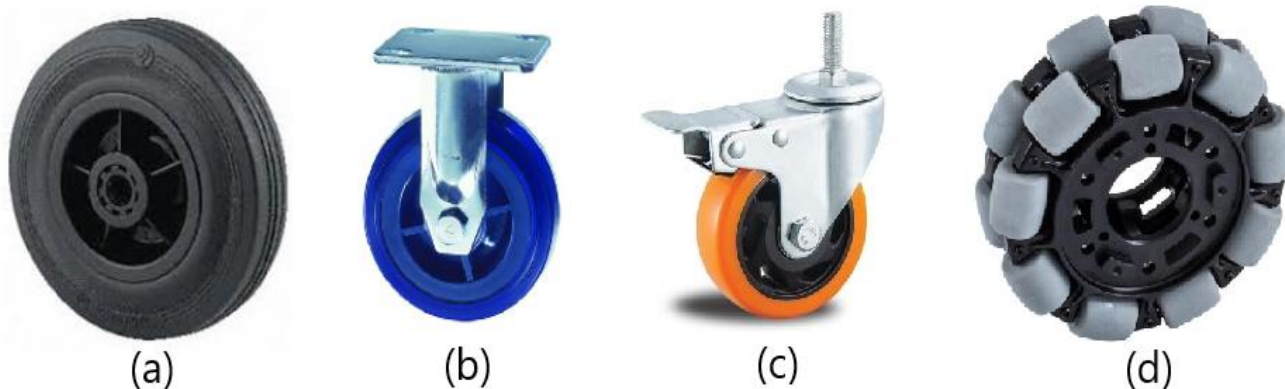


Figura 1.1: Tipos de ruedas. (a) Rueda fija, (b) Rueda estándar, (c) Rueda loca, (d) Rueda sueca.

### 1.5.2. Configuraciones cinemáticas

Los requerimientos básicos del sistema de locomoción para un robot de exploración espacial incluyen la capacidad de girar en el lugar, movimiento tanto hacia adelante como en reversa y la capacidad de negociar obstáculos y pendientes sin perder la estabilidad, esta última no es muy común en los robots terrestres que, por lo general, se diseñan para moverse en superficies planas. En función de estos requerimientos muchos modelos han sido propuestos, sin embargo, se ha convertido en un estándar el uso de un sistema de locomoción de seis ruedas.

Seis ruedas dan una ventaja sobre cuatro ruedas al requerir pares de motor más bajos y proporcionar una capacidad superior para rebasar obstáculos. Ofrecen el mejor compromiso entre el número de ruedas y el rendimiento, más ruedas no ofrecen aumentos significativos de la movilidad para la complejidad adicional [13]. Las configuraciones más utilizadas son:

**Ackerman o tipo carro:** este tipo de diseño es el utilizado por los carros, siendo la configuración más utilizada la de dos ruedas directrices delanteras y dos motrices traseras. Existe un ángulo de giro máximo de las ruedas de dirección. Esta estructura brinda gran estabilidad y evita deslizamiento en las ruedas. Los *rovers* utilizan una configuración doble de Ackerman con seis ruedas, ver Figura 1.2, esa configuración implica dirección en las ruedas delanteras y las traseras, lo que permite realizar giros en el lugar.

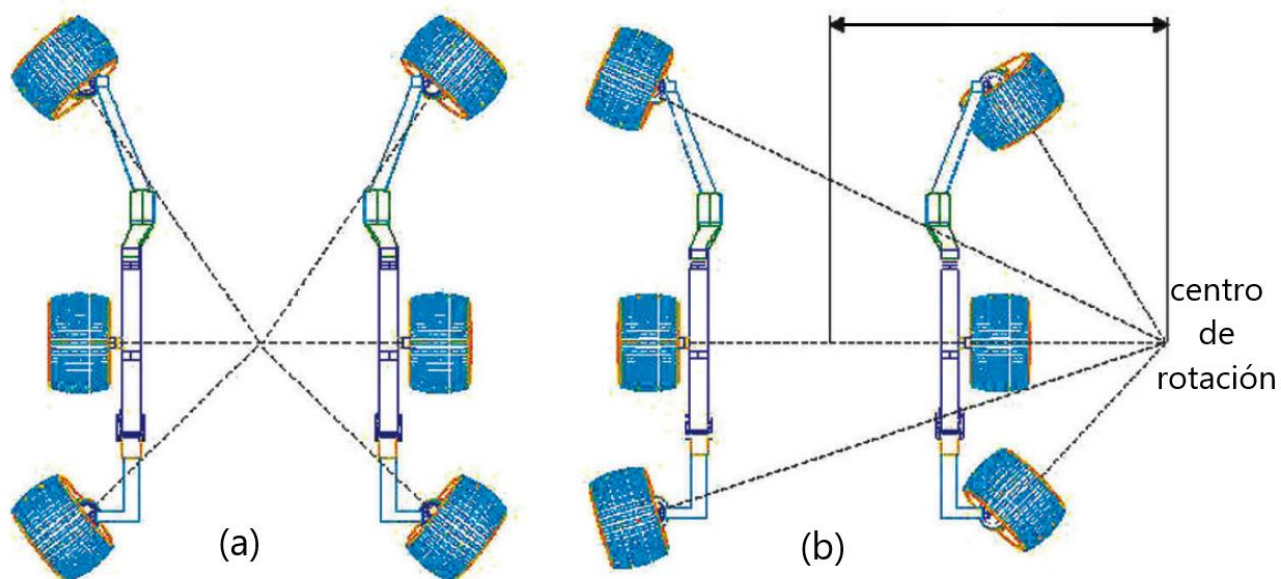


Figura 1.2: Configuración doble de Ackerman en un robot de seis ruedas. (a) giro en el lugar y (b) movimiento hacia la derecha.

**Tracción diferencial:** no hay ruedas directrices. El cambio de dirección se realiza modificando la velocidad relativa de las ruedas a izquierda y derecha, ver Figura 1.3. Es un diseño simple, barato y fácil de implementar, pero requiere de un control preciso para trayectorias rectas. Presenta la restricción de desplazamiento lateral y permite realizar giros en el lugar, como se muestra en la Figura 1.3. Presenta ruedas fijas al cuerpo como se puede observar a continuación.

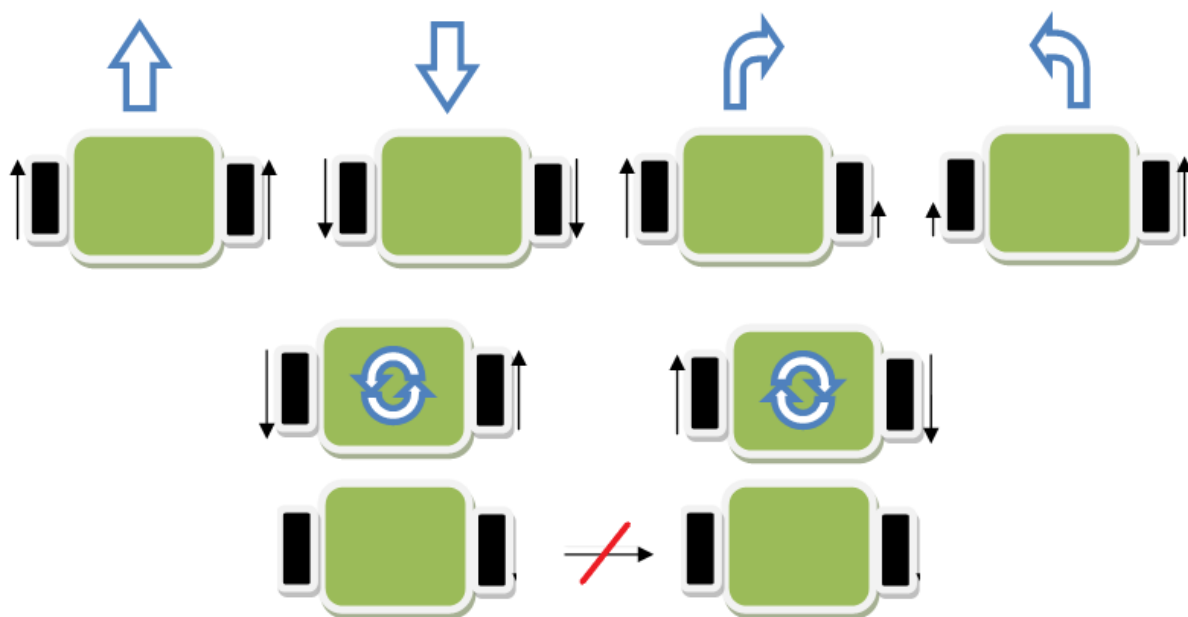


Figura 1.3: Configuración diferencial y sus movimientos.

### 1.5.3. Sistema de suspensión Rocker Bogie

Este diseño se ha convertido en un estándar en los sistemas de exploración espacial debido a su estabilidad y su gran capacidad para superar obstáculos. Fue probado en el Sojourner, los *Rovers* de Exploración Marciana (*Mars Exploration Rovers: MERs*) y el Curiosity. Consiste en un sistema de suspensión articulada sin muelles, con la capacidad de mantener la misma presión en las seis ruedas que lo conforman mientras rebasa obstáculos. La suspensión articulada mejora la movilidad en terrenos peligrosos, permitiéndole al robot cambiar la configuración de su suspensión para ajustar su centro de masa [14]. Esta configuración ofrece el mejor compromiso entre complejidad mecánica y superación de obstáculos [15].

Consta de dos brazos, uno en cada lado del vehículo. El *rocker* (balancín) tiene una rueda trasera conectada a un extremo y un balancín secundario conectado al otro extremo, llamado *bogie* (junta), ver Figura 1.4. Cada *rocker* pivota en su punto medio, donde está conectado al cuerpo del robot usando una barra de torsión diferencial, esta barra es la que permite que el *rover* se mantenga en equilibrio. Cuando un *rocker* sube el otro baja, manteniendo todas las ruedas en contacto con la superficie, como se muestra en la Figura 1.4.

Las seis ruedas son manejadas de forma independiente (motores independientes) y las cuatro de las esquinas son direccionales para proporcionar la capacidad de realizar giros en el lugar. Esta configuración puede rebasar obstáculos de 1.5 veces el diámetro de las ruedas y atravesar grietas del 40% del largo del vehículo, siempre que se mueva a una velocidad de unidades de cm/s, si la velocidad de movimiento aumenta, el vehículo absorbe el impacto con el objeto en lugar de rebasarlo.

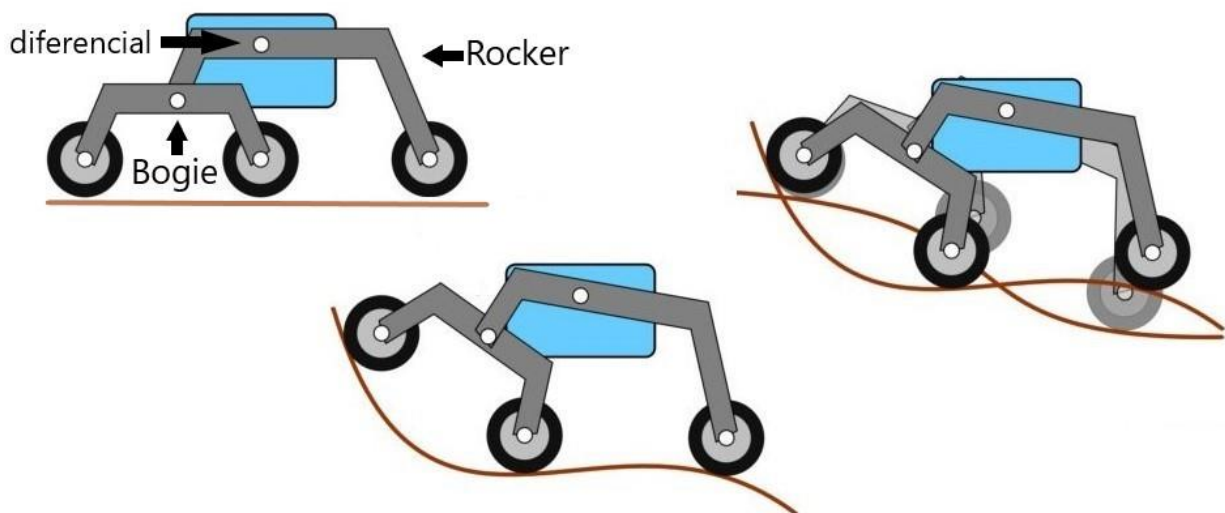


Figura 1.4: Suspensión Rocker Bogie en movimiento.



## 1.6. Controlador, motores y sensores

Un *rover* tiene muchas similitudes con un robot terrestre en cuanto a componentes hardware, los fundamentales para ambos son: el controlador principal, actuadores y sus controladores y los sensores. En uno espacial se incluyen, además, herramientas científicas y algún sistema de generación de energía.

### 1.6.1. Motores

Los motores en robótica se utilizan para generar el movimiento de los diferentes mecanismos o elementos que conforman el robot. Los más utilizados en los RMR son los motores de corriente continua estándar (CC), para mover las ruedas, y los servomotores, para realizar movimientos adicionales, como mover un sensor o una cámara.

#### ***Motores de corriente continua estándar***

Un motor de CC es un dispositivo de dos terminales controlado eléctricamente. Contiene un eje giratorio sobre el que se pueden montar ruedas, engranajes, hélices, etc. Un motor de CC genera una cantidad considerable de revoluciones por minuto (rpm) para su tamaño y puede hacerse girar en sentido horario o antihorario, dependiendo de la polaridad aplicada a los terminales [16]. Estos motores son muy utilizados en la robótica móvil debido a la facilidad de control, su potencia y las revoluciones que ofrecen. Pueden encontrarse entre estos los motores de CC con escobillas y sin escobillas.

Los motores ideales para las aplicaciones de robótica espacial son los motores de CC sin escobillas. Esto se debe a su largo tiempo de vida, alto torque, poco peso, alta eficiencia y poca disipación de calor. Los motores de CC con escobillas de carbón se desgastan excesivamente debido a la fricción con el conmutador y necesitan eventualmente de mantenimiento, por tanto, no deben ser utilizados en misiones de larga duración [17].

#### ***Servomotores***

Un servomotor es un dispositivo similar a un motor de corriente continua, que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición. Está conformado por un motor, una caja reductora y un circuito de control. Los servomotores se utilizan frecuentemente en aplicaciones en las que se requiera alta precisión en el control de posición del eje del rotor [5]. La Figura 1.5 muestra el control de posición de un servomotor mediante

modulación de ancho de pulsos y la posición del eje del motor según el ancho del pulso. La señal modulada entra como referencia al controlador de posición del servomotor.

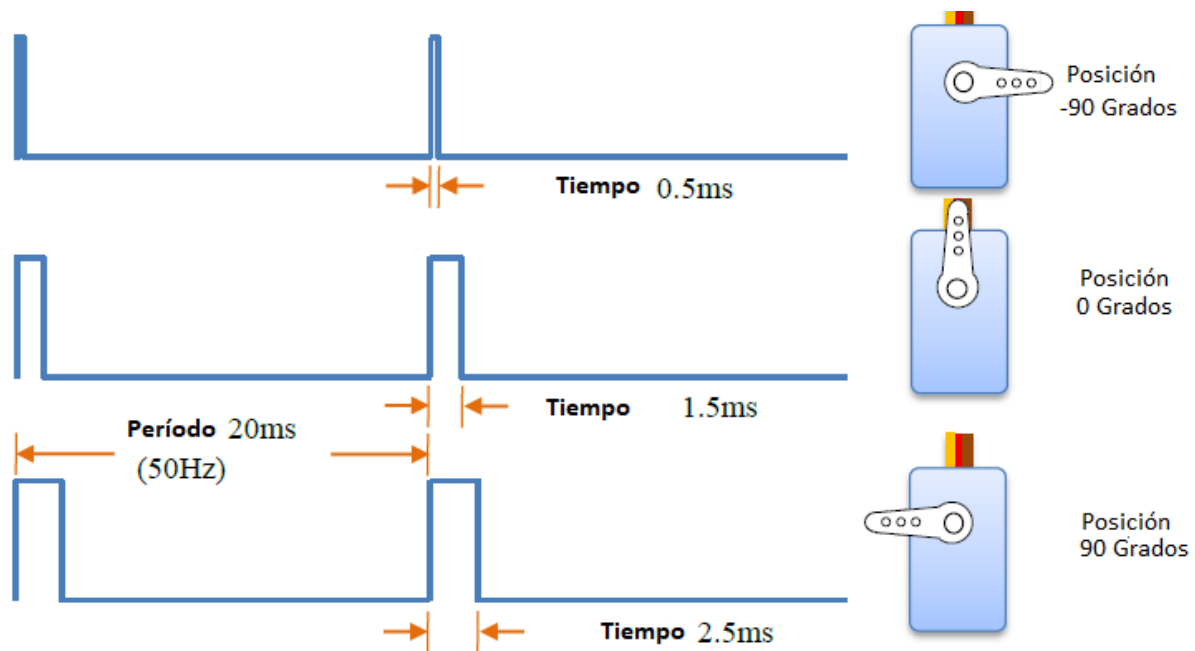


Figura 1.5: Forma de onda y posición del eje de un servomotor.

### ***Modulación por ancho de pulsos (PWM)***

La modulación por ancho de pulsos (*pulse width modulation*), *PWM* por sus siglas en inglés, es un método que evita utilizar circuitos de potencia analógicos para controlar un motor, basándose en que los sistemas mecánicos tienen cierta latencia. Consiste en generar pulsos digitales con el nivel de tensión del sistema, a una frecuencia determinada, por ejemplo, 20 KHz, en vez de crear una señal analógica con una tensión proporcional a la velocidad que se desea [18]. El control de los motores mediante PWM también garantiza un arrancado suave del motor.

El cociente entre tiempo en alto de la señal y el período ( $t_{on}/T$ ) es el ciclo útil. Variando el ciclo útil se modifica el nivel de potencia promedio en el tiempo entregado al motor y por tanto la velocidad de giro. La Figura 1.6 muestra las formas de ondas que se obtienen con PWM.

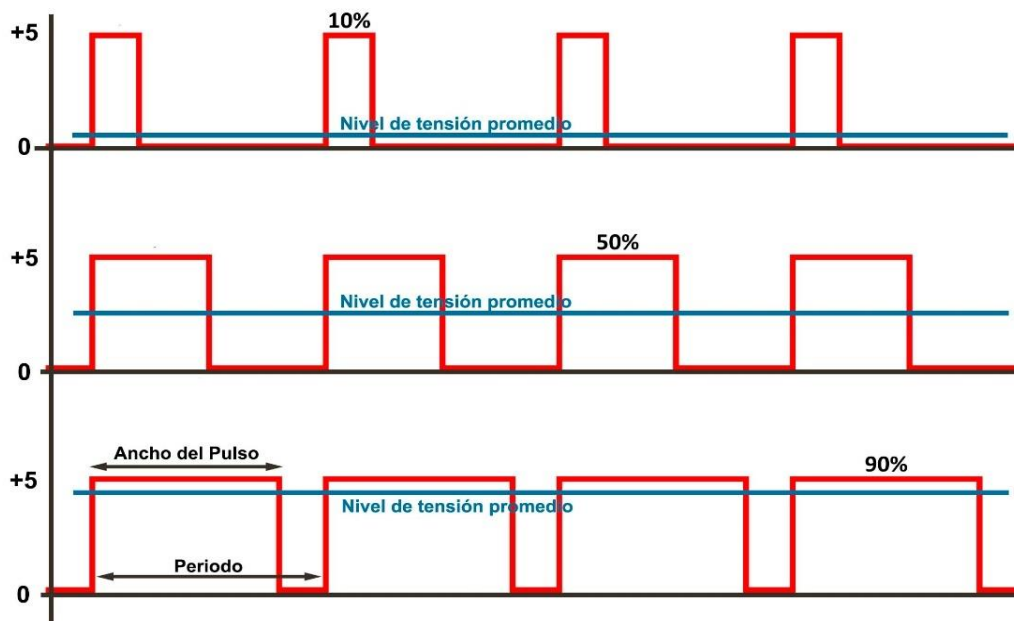


Figura 1.6: Modulación por Ancho de Pulso.

## Puente en H

Un Puente en H es un circuito electrónico que generalmente se usa para permitir a un motor eléctrico de CC girar en ambos sentidos (avance y retroceso). Son ampliamente usados en robótica móvil para controlar el sentido de giro de los motores de las ruedas y para frenar.

El término "puente H" proviene de la típica representación gráfica del circuito. Un puente H se construye con 4 interruptores (mecánicos o eléctricos). Cuando los interruptores S1 y S4 están cerrados y S2 y S3 abiertos se aplica una tensión positiva en el motor, haciéndolo girar en un sentido. Abriendo los interruptores S1 y S4 y cerrando S2 y S3, el voltaje se invierte, permitiendo el giro en sentido inverso del motor [19], además se puede realizar el frenado del motor cerrando S1 y S2 y abriendo los restantes, ver Figura 1.7.

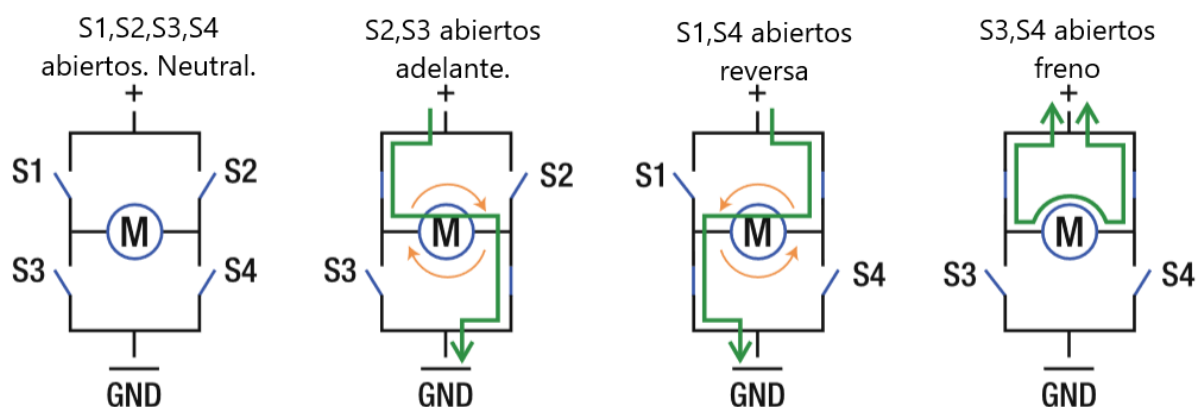


Figura 1.7: Configuraciones posibles de un puente H.

## 1.6.2. Sensores

Los sensores son requeridos para la evasión de obstáculos, la navegación y localización del robot, el manejo de potencia y la detección de fallos. Se pueden dividir en dos ramas [10]:

- Sensores propioceptivos: proveen información sobre estado interno del robot y su dinámica, por ejemplo, velocidad de las ruedas, estado de la batería, etc. (*encoders*, acelerómetros y giróscopos).
- Sensores exteroceptivos: proveen información sobre el medio, por ejemplo, mediciones de distancias, intensidad de luz, etc. (sensor de sol, sensores para detectar obstáculos como ultrasónicos y láser, sensores basados en visión).

El sistema de localización de un *rover* normalmente incluye sensores para medir la distancia recorrida y la orientación relativa y absoluta. Normalmente incluye *encoders*, acelerómetros, giróscopos, sensor de sol y sensores para mapear.

### ***Encoder***

Este es uno de los sensores más populares para medir velocidad angular en el eje de una rueda. Su naturaleza garantiza que no sea necesario el uso de conversores A/D ya que su salida es totalmente digital. Existen diferentes principios de funcionamiento, pero los ópticos y los de efecto Hall son los más utilizados en aplicaciones de robótica. Permiten medir la distancia recorrida por las ruedas mediante el conteo de pulsos, ya sean provocados por un haz de luz o por variaciones electromagnéticas, esto es lo que se conoce como odometría de la rueda. El sentido de movimiento también se puede obtener si se utiliza un *encoder* de cuadratura o de dos canales. En la Figura 1.8 se muestran las formas de onda obtenidas por un *encoder* de cuadratura. En (a) se muestra que cuando ocurre el frente de subida del canal A, el canal B está en nivel bajo, lo que indica giro a la derecha, en (b) se muestra el caso contrario.

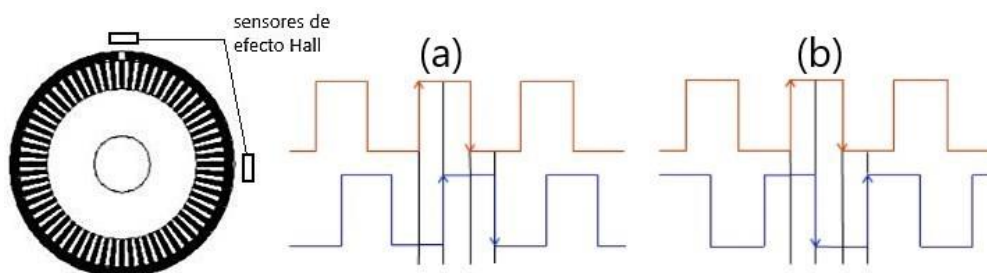


Figura 1.8: Funcionamiento de un *encoder* de dos canales. (a) Giro a la derecha. (b) Giro a la izquierda.

Para determinar el ángulo de giro de una rueda, se cuentan la cantidad de pulsos recibidos y se multiplican por el ángulo de separación entre los pasos, si el sentido de rotación se mantuvo invariable [5].

### ***Unidad de medición inercial***

La unidad de medición inercial (*Inertial Measurement Unit: IMU*) es un dispositivo electrónico compuesto por giróscopos para detectar la dirección, acelerómetros para detectar movimiento (aceleración en ejes) y magnetómetros para detección de campos magnéticos; también pueden contener algún sensor de temperatura [13]. En aplicaciones espaciales generalmente se utilizan con acelerómetros, giróscopos y magnetómetros de tres grados de libertad, alineados con cada eje. Las lecturas de estos sensores se combinan para obtener una orientación precisa con respecto al cabeceo, alabeo y desvío (*pitch, roll, yaw*) [20], véase Figura 1.9.

La ventaja que poseen estos sistemas respecto a los sistemas odométricos es que no se ven afectados por los problemas del deslizamiento o de las irregularidades del suelo en que se desplaza el robot. En la práctica, estos sistemas son más fiables y precisos que los sistemas odométricos, pero son más caros y más frágiles [5].

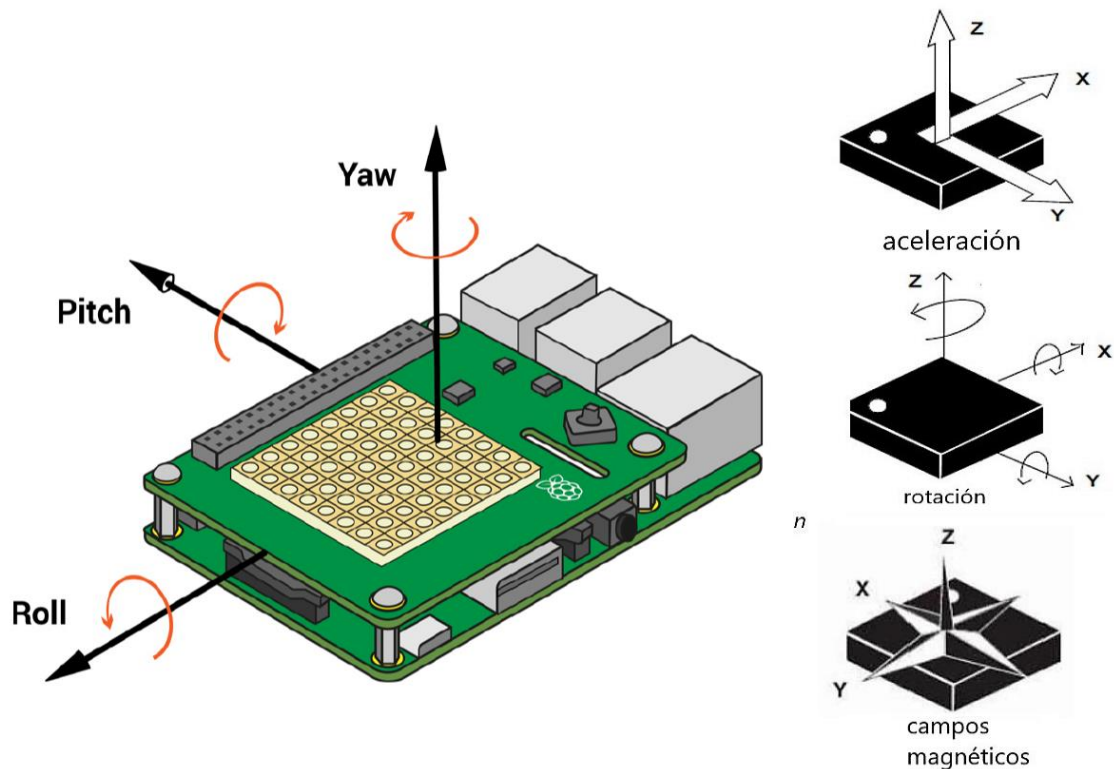


Figura 1.9: Unidad de medición inercial.

Los acelerómetros tienden a sufrir de una relación señal-ruido pobre en aceleraciones bajas, especialmente mientras atraviesan terreno áspero. Suelen tener un error acumulativo o deriva ya que está sumando continuamente los cambios detectados en la posición, cualquier error en esta medida es acumulado. Esto da lugar a un error creciente entre la posición hallada por la IMU y la posición real de ésta.

### ***Sensor de sol***

Un sensor solar se considera esencial para la auto-localización de los robots exploradores en las superficies planetarias. Proporciona una referencia externa con respecto al Sol para superar los problemas incurridos en las imprecisiones de la odometría móvil. El sensor solar es una cámara CCD (*Charge Couple Device*, por sus siglas en inglés) monocromática con una lente angular y filtros de luz de baja intensidad.

### ***Sensores para detección de obstáculos y mapeo***

Para la generación de mapas, la localización y la planificación, los robots requieren sistemas que les permitan detectar elementos del ambiente y determinar cuáles representan obstáculos para su movimiento. Con este propósito se han desarrollado sistemas basados en distintos tipos de sensores, que varían sus características y efectividad dependiendo de la aplicación y del medio donde se emplean [5].

Se han utilizado sensores ultrasónicos e infrarrojos, los primeros sufren de dependencia de la temperatura, reflexión especular y diafonía debido a la reflexión en muchos objetos y los segundos tienen un rango limitado [17]. Actualmente, se utilizan para este propósito los sistemas LIDAR (*Light detection and ranging*) y los sensores basados en visión.

**LIDAR:** emplean haces colimados y con longitud de onda menor que las dimensiones de ondulación de la superficie del objeto, por lo tanto, no se produce el rebote del rayo desde inclinaciones falsas y la reflexión especular. Los escaneos láser proporcionan información sobre los ángulos y las posiciones de los objetos dentro de su campo de visión generando perfiles de profundidad y altura de los objetos en la escena [17]. Estos sistemas son los utilizados en los *rovers* espaciales para detectar obstáculos en conjunto con las cámaras y para formar un mapa del terreno.

**Cámaras:** la visión por cámaras es la mejor forma de reconocimiento del entorno, en los *rovers* sirven para dar soporte a la navegación (generación de mapas y localización de objetos) y para realizar reconocimiento científico [17]. Los *rovers*

utilizan cámaras panorámicas para obtener mayores ángulos de visión. A pesar de que las cámaras son de alta definición, las imágenes son de baja calidad, esto se debe a que pasan por un proceso de filtrado y de edición en el mismo robot para poder transmitir las a la tierra y realizar la navegación.

### **1.6.3. Controladores**

El controlador se encarga de las tareas de percepción (leer el estado de los sensores), controlar la locomoción (manejar los actuadores), procesar la información y manejar la comunicación. Es común encontrarse más de un controlador en un sistema robótico ya que uno solo puede no ser suficiente para todas las tareas, por lo general uno controla los motores y otro realiza las tareas de mayor poder de procesamiento, como el procesamiento de imágenes, la comunicación y la planificación. Últimamente han surgido muchas plataformas de código libre (*open source*) basadas en microprocesadores y microcontroladores y con muchos periféricos que permiten la utilización de estas en muchos campos, la robótica es uno de estos. Un ejemplo es la Raspberry Pi, RPi y los Arduino.

#### ***Raspberry Pi***

Una Raspberry Pi, es una computadora en una placa (*Single Board Computer: SBC*), desarrollada por la Fundación Raspberry Pi en el Reino Unido, para promover el aprendizaje de programación en las escuelas, ver Figura 1.10. Sin embargo, hoy en día se usa ampliamente en muchas áreas, incluso en el proceso de fabricación de supercomputadoras y operaciones avanzadas de robótica [21].

La ventaja de su utilización para aplicaciones robóticas es la facilidad de integración con componentes necesarios en un robot. Para las RPi se han fabricado cámaras con buena resolución, controladores de motores, unidades de medición inercial y sensores específicos para estas.

El procesador del Raspberry Pi es un sistema en chip (*System on Chip: SoC*) Broadcom BCM283x-serie que contiene CPU, GPU, DSP, RAM y controladores USB. Este procesador está basado en la arquitectura de conjunto de instrucciones (*Instruction Set Architecture: ISA*) ARM (ARM11, ARMv6, ARMv7, ARMv8), que provee un bajo consumo de potencia, de ahí que pueda ser alimentada solo con 5V provenientes del micro-USB [22]. Las placas más actuales son la Raspberry Pi 3 modelo B, Raspberry Pi Zero y Zero W.

**Sistema Operativo:** la Raspberry Pi está diseñada para correr sistema operativo Linux, o una de sus distribuciones, aunque puede utilizarse sin uno, en lo que se conoce como modo *bare-metal*, utilizando lenguaje ensamblador. La ventaja de utilizar una de las versiones de Linux es que son compatibles entre ellas y son de código abierto. El sistema operativo recomendado por muchos, incluso por el sitio oficial, es Raspbian [22].

Linux no es un sistema operativo de tiempo real (*Real Time Operation System: RTOS*), esto haría problemático, por ejemplo, muestrear un sensor con precisión de microsegundos ya que puede ocurrir que en el momento de muestreo el Kernel esté ocupado y no se puede interrumpir fácilmente. Por lo tanto, en su estado predeterminado, la RPi no es una plataforma ideal para aplicaciones en tiempo real. Una solución a este problema es incorporar un reloj de tiempo real (*Real Time Clock: RTC*) o combinar la RPi con un microcontrolador como Arduino [21].

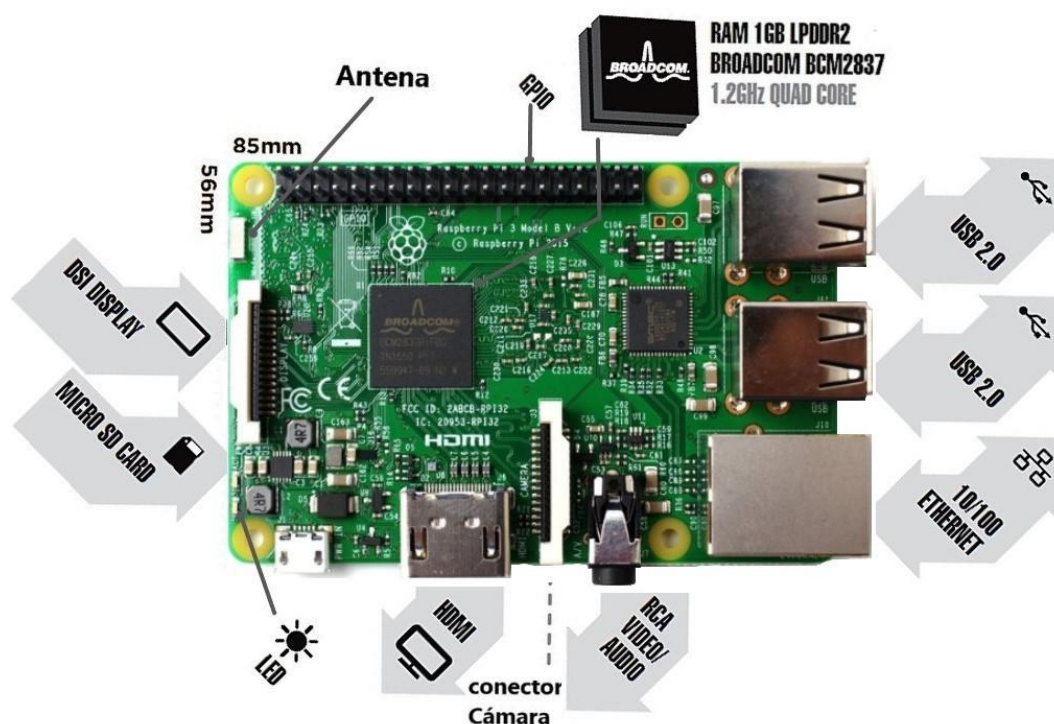


Figura 1.10: Distribución de componentes del Raspberry Pi 3.

## Arduino

Arduino es una plataforma de hardware libre basada en una sencilla placa de entradas y salidas y un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring. Al ser de hardware libre, los esquemáticos están disponibles bajo licencia abierta y pueden ser modificados sin restricciones, cuenta



con una gran comunidad de usuarios y muchos proyectos disponibles para todos, lo que hace que utilizarlo sea muy sencillo y rápido. La programación del controlador interno se realiza mediante USB. Uno de los modelos más utilizados en aplicaciones de robótica es el Arduino Mega [23], ver Figura 1.11.



Figura 1.11: Arduino Mega2560.

## 1.7. Sistema de Generación y almacenamiento de energía

Un sistema robótico planetario es funcionalmente similar a un sistema robótico terrestre, con diferentes características de rendimiento para hacer frente a los requisitos de la misión, por ejemplo, deben ser capaces de sobrevivir sin intervención humana durante meses.

El subsistema de potencia de un *rover* debe ser capaz de generar suficiente energía durante el día y la noche para garantizar el movimiento de los motores, ayudar a la sobrevivencia de los componentes eléctricos, dar soporte a la actividad de exploración científica y despertar al *rover* del modo de hibernación.

Se han utilizado muchas variantes para el sistema de potencia de los *rover*. Se puede hacer lo que se conoce como explorador de día, el cual solo funcionará de día mediante los paneles solares, no incluyen baterías para reducir peso y costo, existen también los que funcionan todo el tiempo, estos están potenciados por Generadores

termoeléctricos de Radioisótopos (*Radioisotope thermoelectric generator: RTG*), que, por su principio de funcionamiento, no se apagan y pueden trabajar durante la noche, pero los más exitosos hasta el momento consisten en paneles solares como fuente de generación de energía, baterías de ion de litio para el almacenamiento y *RTG* en conjunto como fuente de energía.

### **1.7.1. Panel Solar**

Los paneles solares son utilizados para generar electricidad a partir de la incidencia del sol mediante el efecto fotoeléctrico. Esta tecnología es utilizada en varias aplicaciones terrestres y es la fuente principal de energía de las naves espaciales. La utilización de los paneles solares dependerá de la incidencia del Sol en el lugar. En la Tierra la constante de radiación es de  $1366.1 \text{ W m}^{-2}$ , esta sería la incidencia en condiciones excepcionales, pero a medida que aumenta la distancia al Sol la irradiación disminuye, por ejemplo, la constante de radiación en Marte es  $588.6 \text{ W m}^{-2}$  [4].

En la superficie, los efectos atmosféricos pueden afectar también la generación de electricidad, por tanto, para maximizar la eficiencia, las células solares deben ser un poco diferentes a las de la Tierra. Deben estar preparadas para resistir el ambiente severo, caracterizado por tormentas de polvo, bajas temperaturas, intensidad solar baja y un espectro solar que cambia diariamente. La célula solar de la triple-ensambladura de arseniuro de galio (GaAs/GE) es la mejor opción entre las células solares existentes [24]. El arseniuro de galio es el material que ofrece mayor eficiencia actualmente entre los disponibles para paneles solares. Esto significa que produce más potencia para un área de superficie determinada que cualquier otra tecnología solar. Esto es muy importante cuando el área de superficie es limitada. Por ejemplo, la unión triple GaAs solar tiene una eficiencia de 37%, mientras que la de silicio (en el mejor de los casos) alrededor del 21%. Entonces, por unidad de área, GaAs produce casi el doble de potencia. La ventaja de su utilización está en su poco peso y economía, en comparación con las fuentes de radioisótopos (la otra opción de generación de electricidad en el espacio) pero presentan la desventaja de la no generación en espacios sombríos [25].

### **1.7.2. Batería**

Los paneles solares no son una fuente constante de energía ya que dependen de las condiciones atmosféricas, por esto se utiliza un sistema de almacenamiento de

energía. Existen varias formas de almacenar energía, pero la más común es con baterías. Existen diversos tipos de baterías, las cuales se diferencian en su estructura física, modo de funcionamiento o composición química. Algunas son las siguientes:

**Baterías de plomo-ácido:** es el tipo de batería más utilizada hoy en día, ya que todos los vehículos llevan al menos una, y al mismo tiempo es la más antigua de todas, sin haber sufrido grandes cambios desde su invención durante el siglo XIX. Su bajo costo las convierte en ideales para los vehículos eléctricos, siendo utilizadas como acumuladores en vehículos de pequeño tamaño. Sin embargo, cuenta con desventajas como su excesivo peso, la toxicidad del plomo y su lenta recarga. En cambio, es la tecnología que utiliza la batería de 12V de cualquier vehículo con motor diésel o gasolina, muy probada, fiable y robusta, cuya función principal es la del arranque del motor de combustión.

**Baterías de Níquel-Cadmio (NiCd):** estas baterías requieren mantenimiento con menor frecuencia, pueden entregar más corriente en un menor tiempo y su duración es mayor con respecto a las baterías de Pb-ácido. Pueden mantener estados de descarga por períodos largos, siendo más robustas que otras tecnologías al mantener una tensión estable durante ciclos de carga/descarga profundos. Poseen una resistencia interna muy baja, por lo que son ideales para disponer en conexiones en serie. La principal desventaja es su alto costo de inversión, pudiendo ser 10 veces el de un sistema de Pb-ácido, debido al sofisticado proceso de fabricación necesario y a la disponibilidad de los materiales involucrados. Otra de las desventajas de las baterías de Ni-Cd es la alta toxicidad del cadmio.

**Baterías de ion litio (Li-ion):** estas baterías son las más avanzadas que se encuentran en el mercado de vehículos eléctricos. Son muy nuevas y están formadas por un electrolito de sal de litio, por cobalto y por óxido. Los nuevos materiales como el litio han permitido beneficios como la consecución de altas energías específicas, alta eficiencia, eliminación del efecto memoria, ausencia de mantenimiento y facilidad a la hora de reciclar sus desechos. También cuentan con ciertas desventajas, tales como su alto costo de producción, su fragilidad y el especial cuidado a la hora de almacenarlas, ya que necesitan un ambiente frío, además, de estar parcialmente cargadas. Aun así, las ventajas son mayores, pues son realmente capaces de almacenar mayor cantidad de energía ocupando un espacio menor y siendo más ligeras que el resto de las tecnologías que se encuentran actualmente en el mercado. Uno de los aspectos importantes a mencionar sobre las baterías de litio es su alta

densidad de energía; 80-150 Wh/kg. Otra característica es la rápida capacidad de carga/descarga; pueden alcanzar el 90% de su potencia nominal en 200 ms.

Las baterías de ion de litio, representan la opción más prometedora debido a que poseen una densidad de energía, energía específica y rangos de temperatura de operación mayores que los otros tipos utilizados [13]. Son las actualmente utilizadas en las misiones espaciales, sobre todo por su relación peso/energía y a que no es necesario darles mantenimiento. Fueron utilizadas en el Curiosity y los *Mars Exploration Rovers*.

### **1.7.3. Generador termoeléctrico de radioisótopos**

El sistema de energía eléctrica de los MER debe funcionar bien bajo toda la gama de temperaturas del ambiente marciano, que puede variar de  $-140^{\circ}C$  a un alto de  $+20^{\circ}C$  durante la noche y el día marcianos, para garantizar esto, el sistema eléctrico también utiliza un RTG para generar energía auxiliar para las necesidades del *rover*.

El RTG tiene una vida útil larga y emite suficiente calor para mantener la temperatura apropiada para la electrónica dentro del cuerpo del *rover* para sobrevivir las temperaturas frías extremas de Marte. El proceso se basa en la desintegración de algún material radioactivo. Después de que la energía térmica es producida por el RTG, se convierte a energía eléctrica a través de un termopar para ser entregada a unos calentadores eléctricos [24].

## **1.8. Sistema de navegación autónoma y control del vehículo**

La autonomía puede ser definida como la capacidad de un sistema inteligente para establecer y seleccionar diferentes cursos de acción para alcanzar un objetivo, basándose en su entendimiento del entorno y de su situación. Implica cualquier función que ayude a construir una percepción, aprendizaje y toma de decisiones. La navegación es el proceso de atravesar un terreno para alcanzar un objetivo (el objetivo puede ser una coordenada o un punto visual).

La navegación autónoma se basa en sentir el entorno y planear una ruta segura para llegar a una meta, reaccionando ante situaciones inesperadas e involucra cuatro procesos principales: (i) mapeo; (ii) localización; (iii) planificación de una trayectoria; (iv) seguimiento de la trayectoria [17].

Este es uno de los subsistemas presentes tanto en los robots espaciales como en los terrestres, solo que los primeros adicionan ciertas restricciones impuestas por el terreno.

### **1.8.1. Localización y mapeo**

El mapeo implica la interpretación e integración de las mediciones de varios sensores, como cámaras y láser, para construir una representación del entorno. La localización o auto-localización es el proceso mediante el cual el robot determina su configuración (posición y orientación) en el mapa. Existen dos formas fundamentales de localización:

- Localización relativa o *dead reckoning* en inglés, que consiste en calcular la posición y la orientación del robot móvil a partir de una posición inicial conocida, utilizando sensores propioceptivos.
- localización absoluta, se basan en obtener la posición y orientación del sistema de acuerdo a una referencia externa al mismo, aprovechando elementos conocidos del entorno [5].

Los datos obtenidos por estas formas de localización se combinan aplicando filtros Gaussianos, para lograr una estimación de la posición del vehículo [4], así como los valores de *pitch*, *roll* y *yaw* en el caso de un *rover*.

Las tareas de mapeo y localización no pueden ser resueltas de forma independiente. Antes de que un robot pueda responder a la pregunta de cómo se ve su alrededor dado un conjunto de observaciones, necesita saber desde qué ubicación se han hecho estas observaciones. Al mismo tiempo, es difícil estimar la posición actual sin un mapa. No se pueden separar ambas tareas y resolverlas independientemente, para esto se ha propuesto la localización y mapeo simultáneos (*Self-Localization and Mapping: SLAM*). El SLAM es el problema de la construcción de un mapa al mismo tiempo que se localiza el robot dentro de ese mapa [26]. El problema del SLAM aún continúa abierto a mejores soluciones y constituye uno de los objetos de estudio más desarrollados por los investigadores en el campo de la robótica en las últimas décadas [5].

### **1.8.2. Planificación de trayectoria**

Una vez el robot ha creado un mapa del entorno y se ha ubicado en este se procede a la etapa de planificación de trayectoria [17]. Esta etapa consiste en determinar la

secuencia de configuraciones (posición y orientación), que debe determinar el robot para ir de una configuración inicial a una final de forma segura y óptima.

En la exploración espacial es necesario que los algoritmos de planificación permitan la operación sobre diferentes superficies, por lo que algunas características del terreno deben ser consideradas en el modelo del robot, a diferencia de la robótica móvil tradicional. Esto se conoce como el problema de la cinemática inversa y análisis de fuerza y es de suma importancia en la planificación de trayectorias sobre terrenos irregulares para determinar qué localizaciones serán viables [14].

### ***Evaluación del terreno. Cinemática inversa y análisis de fuerza***

El objetivo del análisis cinemático, es determinar la configuración cinemática del robot en un punto del terreno para saber si no se violarán los límites de las articulaciones. Puede ser expresado de forma siguiente para un robot móvil con la configuración de la Figura 1.12: dado un mapa del terreno y la posición de un punto, el robot deberá calcular la configuración espacial que tendría  $(\theta, \phi, \varphi)$  y la configuración  $(\theta_1 \dots \theta_n)$  de su suspensión. Donde  $(\theta, \phi, \varphi)$  son el *roll*, *pitch*, *yaw* del cuerpo del robot respectivamente y  $(\theta_1 \dots \theta_n)$  los ángulos que forman sus articulaciones. En la Figura 1.12 se muestra un ejemplo de este problema. En [27, 28] se ofrece una solución para un robot con esta configuración, se proponen ciertas simplificaciones que reducen el tiempo de ejecución del método, permitiendo su implementación durante la marcha.

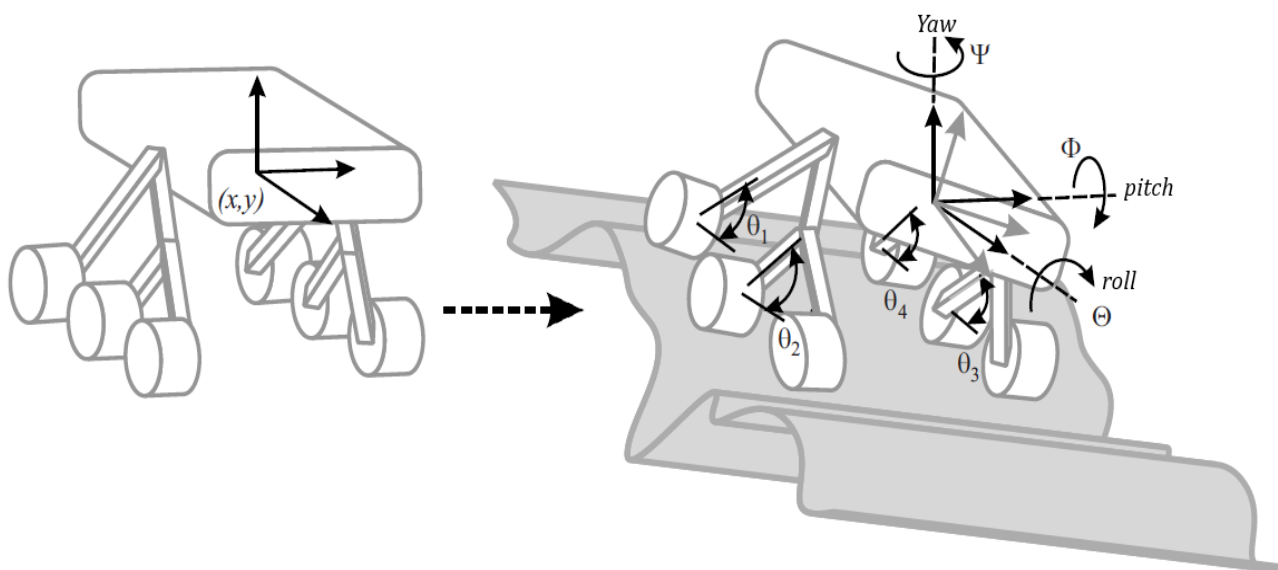


Figura 1.12: Rover en su posición normal y en un terreno con irregularidades.

Otro factor a tener en cuenta en la planificación para este tipo de entornos es si el robot, en un punto del terreno, puede ejercer suficiente empuje en una dirección determinada sin sobrepasar el torque de saturación de los motores, esto se conoce como el análisis de fuerza, y puede ser definido de la siguiente forma: dada la configuración del robot y los ángulos de contacto de las ruedas con el suelo, determinar si existe un conjunto de fuerzas terreno-ruedas que permitan el movimiento del cuerpo en la dirección deseada [14]. Para esto es necesario hacer una caracterización e identificación de los parámetros claves del suelo. En [14, 27, 28] se describe más detalladamente el análisis de fuerza y la estimación de los parámetros del suelo y se proponen soluciones on-line para este problema.

El robot debe realizar los análisis anteriores para obtener lo que se conoce como mapa de transitabilidad [4]. En la Figura 1.13 se muestra un mapa obtenido luego de realizar la evaluación del terreno, el color de las celdas está en función de las propiedades del terreno. Las celdas en rojo indican que es un punto peligroso y las marcadas en azul uno seguro. El camino A es el más corto, pero más peligroso, por lo tanto, se considera el camino B. Esta representación luego es transformada, al igual que la representación del robot para aplicar un algoritmo de planificación.

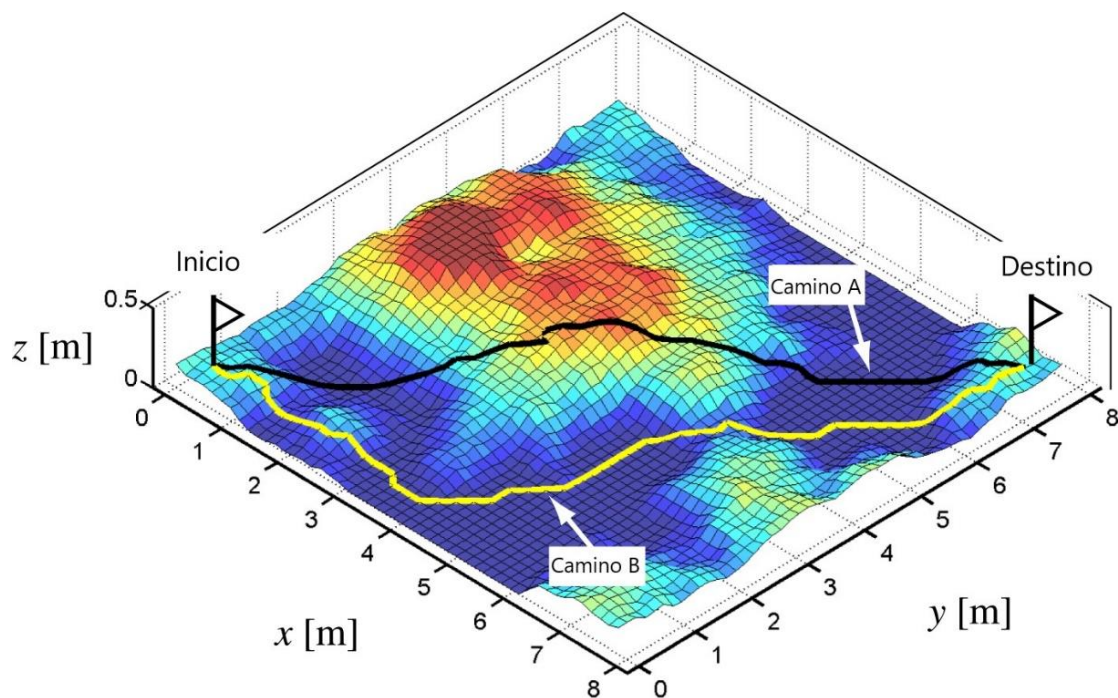


Figura 1.13: Mapa del entorno luego de realizar la evaluación del entorno.



## Algoritmos para planificación

La mayoría de los métodos de planificación se basan en los algoritmos de búsqueda en grafos, para esto es necesario una representación del entorno en la que puedan ser aplicados, esta transformación se conoce como espacio de configuración (C-espacio).

Cada punto del C-espacio representa los valores que corresponden a los grados de libertad del robot, para un robot móvil son posición y orientación,  $q = (p, \theta) = (x, y, \theta)$ , el C-espacio queda definido de forma siguiente:

$$C_{\text{espacio}} = \{q \in C \mid R(q) \cap (\bigcup_{i=1}^q O_i(q)) \neq \emptyset\} \quad (1.1)$$

Donde  $q$  es la configuración del robot,  $R(q)$  es el subconjunto de C ocupado por el robot,  $O_i(q)$  es el conjunto de configuraciones ocupadas por un obstáculo y  $C_{\text{espacio}}$  es el espacio libre de obstáculos [7].

Esta representación permite mantener el robot alejado de los obstáculos modelándolos de diferentes formas. Existen tres métodos principales de transformar el entorno en el C-espacio: (i) descomposición en celdas; (ii) campos potenciales; (iii) mapa de caminos (*roadmaps*). Una descomposición en celdas se muestra en la Figura 1.14, las celdas negras representan localizaciones inválidas, en [7, 8, 29-35] se explica esta forma de representación del entorno.

Una vez realizada la representación del entorno se procede a buscar el camino óptimo. Los algoritmos más populares son el A estrella ( $A^*$ ), el cual es una versión extendida del algoritmo Dijkstra de búsqueda en grafos y el  $D^*$ , que es la versión de  $A^*$  para entornos dinámicos y desconocidos. En [13] se describen el  $A^*$  y el  $D^*$  y se utilizan para generar una trayectoria para un *rover*, en [36, 37] se comparan el  $A^*$  y el de Bellman-Ford mejorado y en [38] se aplica el  $A^*$  a un robot tipo carro.

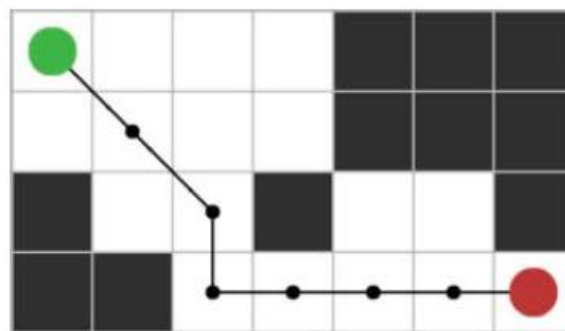


Figura 1.14: Camino planificado en un entorno descompuesto en celdas mediante el  $A^*$ .



### 1.8.3. Generación de trayectoria

La secuencia de poses que se obtiene con el planificador, por lo general, no tiene en cuenta las características cinemáticas del robot, por lo que se incluye la etapa de generación de trayectoria. Esta es la encargada de transformar el camino obtenido en una trayectoria de referencia factible para el robot.

Un robot para exploración espacial, generalmente, tendrá una configuración Ackerman, con seis ruedas impulsadas independientemente. Esta configuración le permite realizar giros en el lugar, lo que significa que podrá seguir cualquier trayectoria planificada, solo tendría que girar sobre su propio eje hasta lograr la dirección deseada y luego moverse en línea recta. Sin embargo, para girar sobre su propio eje, el robot deberá detenerse y alinear sus ruedas, lo que se traduce en más tiempo y potencia, por lo que es preferible utilizar giros Ackerman, estos son más eficientes a la hora de seguir una trayectoria ya que el *rover* podrá realizar cambios continuos de orientación [39].

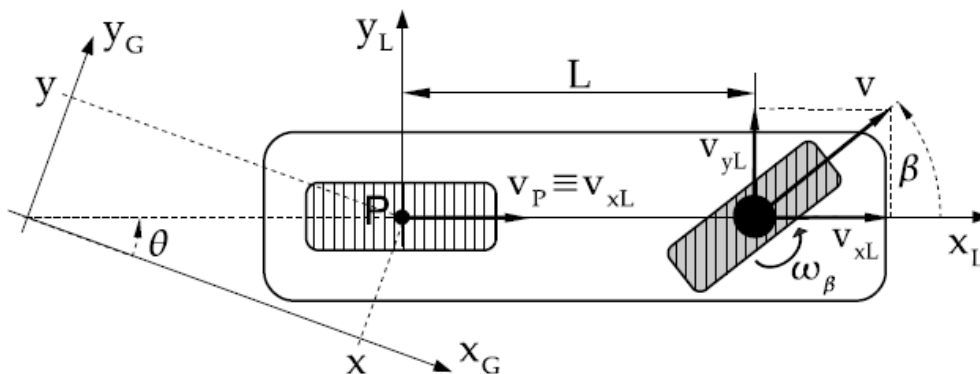


Figura 1.15: Modelo Ackerman simplificado al modelo bicicleta.

La configuración mostrada en la Figura 1.15 es la de un robot tipo carro de cuatro ruedas, donde cada par de ruedas se sustituyó por una sola en el punto medio que las separa. Esta simplificación se conoce como modelo bicicleta y es también válida para un robot con doble dirección y seis ruedas. Las seis ruedas se convierten en tres, la dirección de las ruedas delanteras y las traseras  $\beta$  es simétrica, por tanto, se simplifican las ruedas traseras, quedando solamente el modelo bicicleta [39].

Un robot con esta configuración debe seguir una trayectoria con las siguientes características [7]:

- **Continuidad en posición, orientación y curvatura:** discontinuidades en la orientación del camino conllevan a la necesidad de imprimir un cambio brusco en la orientación del vehículo, el cual no puede ser efectuado debido a la restricción de no holonomicidad. Por otra parte, una discontinuidad en la curvatura requeriría una aceleración infinita de la rueda de dirección.
- **Acotación de los valores que puede tomar la curvatura:** debido a la definición de la curvatura como la inversa del radio de curvatura, existe un radio mínimo que puede realizar el vehículo según el ángulo máximo de direccionamiento.
- **Variación lineal de la curvatura:** una variación suave y lineal de la curvatura minimiza el esfuerzo de control que se debe ejercer sobre los actuadores del vehículo.

Una de las formas más comunes de obtener estos caminos es realizando una interpolación en la secuencia de poses por las que debe pasar el robot. Sin embargo, existe un conjunto de curvas que son utilizadas para obtener caminos suaves, algunas de estas son las espirales cúbicas, las curvas de Bézier, las  $\beta$ -spline y las clotoides [40]. En la Figura 1.16 se muestra la diferencia entre un camino planificado y una trayectoria suave.

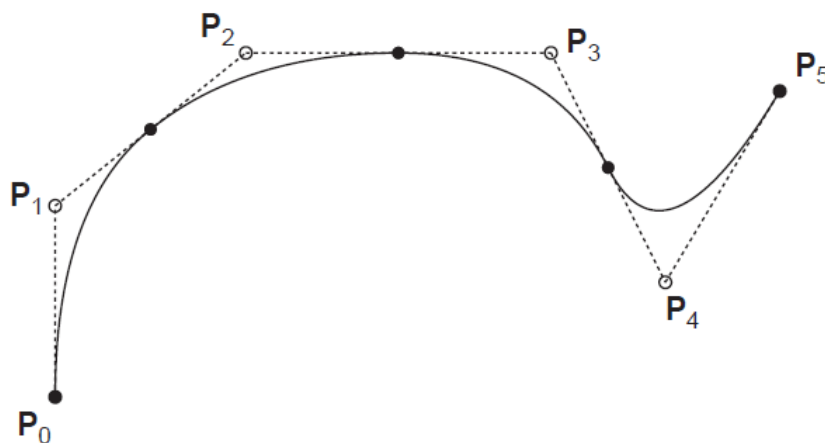


Figura 1.16: Comparación entre un camino planificado y uno generado mediante curvas clotoides.

En [41, 42] se hace uso de las espirales cúbicas y los arcos circulares para generar una trayectoria suave, en [43] las curvas de Bézier y en [44] las  $\beta$ -spline para realizar maniobras de parqueo paralelo.

### 1.8.4. Control de seguimiento de trayectoria

Esta capa es la encargada de llevar a cabo el seguimiento de la trayectoria de referencia generada en la etapa anterior, el resultado final de esta etapa, por lo general, es un conjunto de velocidades de referencia que deberá seguir el robot. La aplicación de un control en lazo abierto es posible debido a que esta etapa recibirá una trayectoria suave, libre de obstáculos y posible de realizar de las anteriores, una trayectoria que considera las restricciones no-holónomas del vehículo.

Sin embargo, eventos no modelados en tiempo real, como deslizamientos de las ruedas y errores en la localización inicial pueden hacer que no se logre la trayectoria deseada, como solución a esto se agrega una retroalimentación del error actual, mediante la odometría, haciendo un control en tiempo real de lazo cerrado [45].

El controlador de seguimiento de trayectoria actúa directamente sobre el modelo cinemático del robot, por lo que varía en función de este.

#### ***Modelo cinemático del robot***

Uno de los modelos más utilizados es el del vehículo de accionamiento diferencial VAD, ver Figura 1.20 a. Las ecuaciones cinemáticas que lo describen son las siguientes:

$$\dot{x} = \frac{r}{2}(\omega_r + \omega_L)\cos\varphi \quad (1.2)$$

$$\dot{y} = \frac{r}{2}(\omega_r + \omega_L)\sin\varphi \quad (1.3)$$

$$\dot{\varphi} = \frac{r}{b}(\omega_r - \omega_L) \quad (1.4)$$

Donde,  $\varphi$  es la orientación del cuerpo del vehículo respecto al eje x,  $r$  es el radio de las ruedas,  $b$  la separación entre las ruedas y  $\omega_r, \omega_L$  las velocidades angulares de la rueda derecha e izquierda respectivamente.

Este modelo se puede aproximar al del unicycle, Figura 1.17 b. El modelo del unicycle es el de un robot cuyo cuerpo es una rueda, solo interesan su posición y dirección. Al realizar esta aproximación se puede trabajar con las velocidades angular y longitudinal del cuerpo del robot, quedando el modelo de forma siguiente:

$$\begin{bmatrix} \dot{\phi} \\ \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ \cos\phi \\ \sin\phi \end{bmatrix} u_2 \quad (1.5)$$

Donde  $u_1, u_2$  son la velocidad angular y lineal de la plataforma respectivamente. Sin embargo, en un robot diferencial, las variables de interés son las velocidades angulares de las ruedas, por lo que, luego de obtener un conjunto de velocidades del cuerpo del robot, es necesario realizar la siguiente transformación para obtener las velocidades angulares de las ruedas:

$$\omega_r = \frac{2u_2 + u_1 b}{2r} \quad (1.6) \quad \omega_l = \frac{2u_2 - u_1 b}{2r} \quad (1.7)$$

La aproximación al unicycle es realizada en [46, 47] para aplicar el método llamado Orientación del Vector de Campo (*Vector Field Orientation: VFO*) para un VAD. Esta forma de control se ha utilizado también en [48-50] para un robot con configuración Ackerman y dirección frontal, trasera y un triciclo respectivamente.

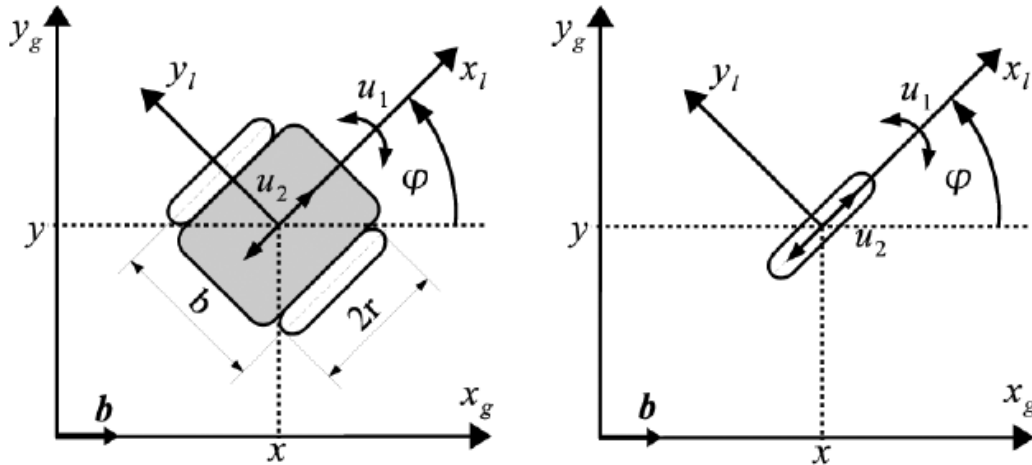


Figura 1.17: Modelo cinemático del robot. (a) diferencial. (b) unicycle.

Un *rover* con configuración doble de Ackerman también puede ser aproximado al modelo del unicycle realizando las siguientes consideraciones: el *rover* puede ser aproximado al modelo de la bicicleta como se explicó anteriormente en la sección 1.8.3, ver Figura 1.15. Una vez en esa forma se puede obviar la dinámica de la rueda delantera asumiendo que la respuesta de giro de la rueda es muy rápida en comparación a la variación del ángulo de giro, esta suposición es válida para estos robots debido a la velocidad de movimiento tan baja (unidades de cm/s).

### 1.8.5. Control en lazo cerrado

El movimiento debe ser garantizado bajo cualquier circunstancia. El vector de velocidades es proporcionado por el controlador principal luego de realizar el control en lazo abierto. Sin embargo, la aplicación de un nivel de tensión determinado al motor no garantiza que este se mueva a la misma velocidad en todas las circunstancias, ya que la velocidad dependerá de la carga.

En un robot de accionamiento diferencial es necesario realizar un control preciso de velocidad en los motores, para el movimiento en línea recta, la velocidad de las ruedas debe ser exactamente la misma, esto debe ser garantizado mediante un control en lazo cerrado ya que factores como la subida de pendientes o la fricción con el terreno puede ocasionar variaciones en la velocidad de las ruedas, lo que es necesario corregir para garantizar el seguimiento de la trayectoria correctamente.

Para realizar el control de velocidad de los motores se utilizan los *encoders* de cuadratura. La medición de la velocidad angular real se realiza contando la cantidad de pasos de los *encoders* y utilizando las ecuaciones 1.8 y 1.9.

$$\theta_{d,i} = 2\pi \frac{CNT_{d,i}}{CeR_e} \quad (1.8)$$

Donde  $\theta_d$  es el desplazamiento angular,  $CNT$  es el conteo de pulsos,  $R_e$  la resolución de los *encoders* y  $Ce$  es la relación de la caja de engranajes del motor.

La velocidad instantánea está dada por el cambio de posición en un intervalo de tiempo expresado como:

$$\omega_{d,i} = \frac{\theta_{d,i}(t) - \theta_{d,i}(t - 1)}{Tm} \quad (1.9)$$

Donde  $\omega_{d,i}$  es la velocidad angular de las ruedas derecha e izquierda y  $Tm$  es el periodo de muestreo.

Con la velocidad angular real se calcula el error con la deseada mediante 1.10, luego este error es el que se utiliza para el controlador.

$$e\omega_{d,i} = \omega d_{d,i} - \omega_{d,i} \quad (1.10)$$

Donde  $e\omega_{d,i}$  es el error con la velocidad deseada,  $\omega d_{d,i}$  es la velocidad angular deseada de las ruedas derecha e izquierda. Se utiliza normalmente un controlador PID para este tipo de aplicaciones debido a la precisión requerida en el movimiento en línea recta del robot diferencial. Para implementar un controlador PID en un

sistema digital es necesario discretizar su forma continua. En las ecuaciones 1.11 y 1.12 se muestra el controlador en sus formas continua y discreta respectivamente:

$$u(t) = K_c \left( e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right) \quad (1.11)$$

$$U(z) = \left[ K_p + \frac{K_i}{1 - z^{-1}} + K_d(1 - z^{-1}) \right] E(z) \quad (1.12)$$

Los parámetros  $K_p$ ,  $K_d$ ,  $K_i$  deben ser ajustados para el correcto funcionamiento del controlador, esto puede hacerse de forma experimental hasta obtener una respuesta favorable del controlador o puede utilizarse el método de Ziegler y Nichols. En [51] se realiza el control de velocidad de los motores de un robot diferencial mediante la implementación de un controlador PID en su forma de velocidad y se realiza el ajuste de los parámetros mediante el método propuesto por Ziegler y Nichols.

## 1.9. Algunas plataformas robóticas utilizadas para la exploración espacial

**Rovers Lunokhod 1 y 2:** el Lunokhod 1 fue desplegado en la Luna en 1970, fue diseñado para operar durante 3 días lunares (alrededor de 3 meses terrestres) pero funcionó durante 11 días lunares (302 días). Durante su tiempo de vida recorrió 10.45 Km. Dos años después fue desplegado el Lunokhod 2. Ambos rovers eran teleoperados desde la Tierra durante el día lunar, y puestos en hibernación durante la noche. Contaban con 4 cámaras de televisión, espectrómetros, telescopio, detectores de rayos cósmicos e instrumentos para recoger muestras del suelo lunar. Para garantizar la supervivencia de los componentes durante la noche se utilizaba una fuente de calor, generado por el polonio-210. Para aprovechar el calor generado, existía un panel solar que se cerraba tapando el radiador. Durante el día, el panel generaba suficiente potencia como para permitir que el rover fuera teleoperado [4].

Podían viajar a casi 60 cm/s, mucho más rápido que las actuales, esto se debe a que fueron desplegadas en la Luna, que está más cerca de la tierra que Marte (donde fueron desplegadas las otras), por lo que la comunicación es mucho mejor, además no tenían suspensión Rocker Bogie, lo cual hace que pudieran viajar a más velocidad

**Sojourner rover:** tocó la superficie de marte por primera vez en 1997. Comparado regularmente con un horno de microondas, el rover Sojourner era de 63 cm de largo, 48 cm de ancho, 28 cm de alto y una masa de 11.2 Kg. La plataforma estaba formada por una caja cerrada que contenía toda la electrónica para mantener el calor,

permitiéndole así sobrevivir a las noches frías de Marte (-110 grados Celsius), esto es lo que se conoce como Caja de electrónica caliente, WEB en inglés (*Warm Electronics Box*).

El subsistema de locomoción consistía en 6 ruedas con suspensión rocker-bogie, lo cual se convirtió en casi un estándar en la robótica de la NASA. Se movía a una velocidad de 0.6 cm/s (0.4 cm/s nominal) [13]. Utilizaba motores de CC con escobillas Maxon RE016 para el movimiento de las ruedas, debido al período de duración para el que fue pensado.

El subsistema de potencia estaba basado en paneles solares, la primera demostración de estos en Marte, y complementado por una batería no recargable que proveía 150 Wh de energía, destinando 4 W para el movimiento, 1 W para el microcontrolador y 1 W para la navegación. La decisión de una batería no recargable fue debido a la duración de la misión. Utilizaba una antena UHF para la comunicación con la lanzadera [52]. Se muestra en la Figura 1.18.

**Rovers Spirit y Opportunity:** los MERs, después nombrados Spirit y Opportunity, fueron enviados a Marte en 2003 y desplegados en 2004, con el objetivo científico de encontrar rastros de agua. Equipados con cámaras panorámicas (PanCam) y una serie de instrumentos para recoger muestras rocosas. Similar al Sojourner, estaba equipado con un sistema de locomoción de 6 ruedas en configuración rocker bogie con 6 motores Maxon RE20 y otros 2 para el sistema de suspensión del *chasis*. Los *rovers* eran mayores que el Sojourner, pesaban 180 Kg, con 1.5 m de alto, 2.3 m de ancho y 1.6 m de largo. Estos se pensaron para una velocidad de 5 cm/s, aunque en la práctica su promedio fue de 1 cm/s [53].

El subsistema de potencia consistía en un arreglo de paneles solares, ocupando un área de 1.3 m<sup>2</sup> y generando hasta 900 Wh, desplegados a partir de 5 motores Maxon RE20. Dos baterías de Litio-ion, de 7.15 Kg cada una, que generaban hasta 600 Wh y 30 V. Ambos sufrieron por el polvo que se alojó en sus paneles afectando la generación de electricidad. El tiempo estimado de duración de ambos era de 3 años y ambos superaron ese tiempo con creces, el Spirit sobrevivió hasta que cayó en un banco de arena suave, en 2009, quedando inmovilizado, hasta que dejó de funcionar en 2010. El Opportunity siguió funcionando, sorprendentemente, hasta el 2015 [4].

**Curiosity rover:** aterrizó en Marte el 6 de agosto de 2012. El *rover* lleva el juego de instrumentos más grande y avanzado para estudios científicos enviados a la superficie marciana, es conocido como el laboratorio científico de Marte [54].

Con una masa de 899 Kg, es la plataforma robótica más pesada que haya aterrizado en otro planeta, superando los 840 Kg del Lunokhod 2. Con un largo de 3.0 metros, un ancho de 2.1 metros y un alto de 2.1 metros, el Curiosity es comparable a un carro pequeño. El sistema de locomoción es semejante al de los MER, contando con sistema de suspensión rocker-bogie y 6 ruedas. Las ruedas son de 50 cm y están hechas de aluminio y titanio. Utilizaba un *RTG* para proveer alimentación y para generar calor en la noche [4]. Este se muestra en la Figura 1.18 junto con el Sojourner y los MER.

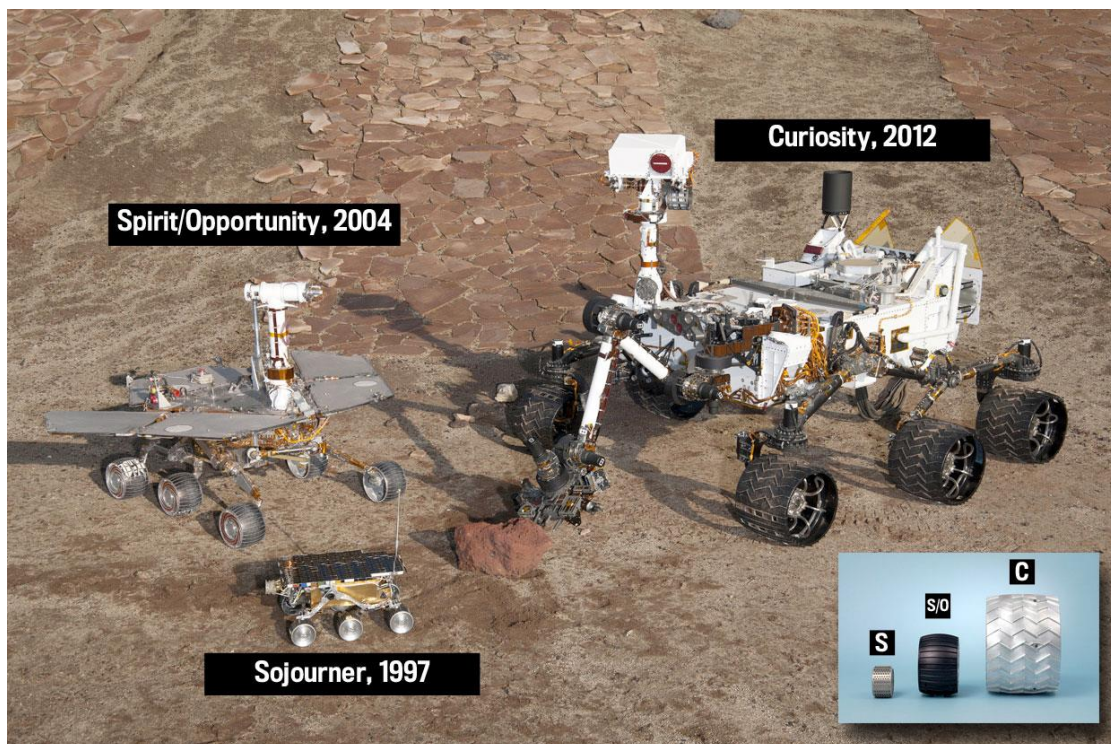


Figura 1.18: *Mars Exploration Rovers*.

## 1.10. Autonomía y estándares del ECSS

Los sistemas robóticos planetarios no precisan de un nivel fijo de autonomía. Pueden utilizar una amplia gama de modos de control desde teleoperación, semiautónomo hasta totalmente autónomo, según corresponda con el objetivo de su misión, ubicación y restricciones operacionales.

Existen tres factores fundamentales que ilustran la necesidad de la autonomía en las misiones espaciales. (i) **Comunicaciones:** debido a la dificultad en las comunicaciones altos niveles de autonomía (*Levels of Autonomy: LoAs*) son requeridos para las misiones espaciales, particularmente cuando la distancia es muy



larga. La distancia entre la Tierra y Marte es de 54.6-401 millones de kilómetros, lo cual resulta en un retardo de tiempo de aproximadamente 3-22 min [4]. (ii) **Incertidumbres en el Entorno:** la representación del entorno que tiene la estación en la Tierra es solo una aproximación de la realidad, y el modelo no es adquirido en tiempo real debido a las dificultades en la comunicación. (iii) **Costo de la operación:** agregando autonomía a la nave espacial o robot, el hombre puede enviar comandos de alto nivel que ocasionen que el sistema reaccione al entorno y se recupere. Se evita el envío de más recursos humanos al espacio, lo cual encarece las misiones.

Para asegurar compatibilidad, interoperabilidad, seguridad, repetibilidad y calidad, esta organización estableció ciertos estándares en cuanto a autonomía y operaciones espaciales [4].

### **1. Operación del segmento espacial (*on-board*).**

Este estándar define una serie de requerimientos para el diseño de funciones *on-board* en naves espaciales no tripuladas. Presenta también el concepto de autonomía de la ECSS.

**“Capacidad del segmento espacial para gestionar operaciones nominales o de contingencia sin intervención del segmento de tierra durante un período de tiempo determinado.”**

Establece además tres ramas para la autonomía *on-board*:

- Autonomía en la ejecución de la misión ('E')
- Autonomía en el manejo de datos ('D')
- Autonomía en el manejo de fallos ('F')

Cada rama tiene un conjunto LoAs que especifican cuánta autonomía se necesita para una misión determinada [55]. Es de interés la autonomía en la ejecución de la misión. Los niveles son los siguientes.

**E1. Teleoperados:** ejecución de la misión bajo control de tierra, capacidad limitada para problemas de seguridad. Se realiza el control en tiempo real y la ejecución de comandos marcados en el tiempo. Velocidad variable. Este modo no es viable para operaciones en Marte.

**E2. Blind Drive:** Ejecución de operaciones de la misión, definidas previamente en tierra. Comandos directos que solo realizan movimientos reactivos y verificaciones de seguridad. Velocidad aproximada de 35 cm/s.

**E3. Semiautónomo:** Ejecución de misiones adaptativas. Evasión de obstáculos basados en señales visuales siguiendo odometría visual en cada paso. Velocidad aproximada de 10 cm/s.

**E4. Totalmente autónomo:** Ejecución de operaciones de misión orientadas a objetivos. Re planeación de misiones orientadas a objetivos. Velocidad aproximada de 2 cm/s.

## 2. Otros modos de operación

**Modo seguro:** Cuando se detectan fallos, la plataforma puede tomar este modo como precaución y suspender todas sus tareas para enfocarse en la recuperación. La recuperación implica primeramente reestablecer la comunicación con el control en tierra y restaurar la potencia en los subsistemas para resumir la misión. El tiempo de recuperación puede ser de minutos, horas, días o semanas, dependiendo de cuan complejo sea reestablecer la comunicación con la tierra y de la complejidad del fallo.

**Modo de hibernación:** Justo antes de una tormenta de arena o durante la noche, el MER entra en bajo consumo. En este modo el reloj de tiempo real (RTC) del robot sigue funcionando, pero todos los servicios son cancelados para ahorrar energía y cargar las baterías. Cuando la batería está en un nivel aceptable, el robot despierta y trata de comunicarse.

Los robots planetarios también pueden ser clasificados en tres grupos, dependiendo de su capacidad de alcanzar diferentes niveles de autonomía de la ECSS:

- **Agentes robóticos:** Actúan como representantes humanos (human proxis) en el espacio para realizar tareas de exploración, ensamblaje, mantenimiento y otras, en los niveles E1-E3.
- **Asistentes robóticos:** Ayudan a los astronautas a realizar tareas de forma rápida, segura y con mayor calidad, operan en los niveles E3 o E4.
- **Exploradores robóticos:** Exploran objetivos extraterrestres usando el nivel E4 de control.

## 1.11. Conclusiones

El estudio de las plataformas para la exploración espacial permitió adquirir conocimiento sobre nuevas formas de movimiento, superación de obstáculos, los principales componentes que conforman sus subsistemas y similitudes con los de un robot terrestre, los modos de operación y niveles de autonomía que pueden tener y los análisis que deben ser realizados para el movimiento en terrenos complejos.

## Capítulo 2. Diseño de la Plataforma

### 2.1. Introducción

En este capítulo se realiza una propuesta de diseño de un robot móvil con algunas características de un *rover*. Se realiza la selección y descripción de los componentes que forman parte del diseño propuesto.

### 2.2. Locomoción

Como se explicó anteriormente, los *rovers* necesitan de tres habilidades principales en cuanto al movimiento, estas son: girar en el lugar, movimiento en ambos sentidos y superar obstáculos.

Se seleccionó una configuración diferencial con suspensión Rocker Bogie (ver Figura 2.1). Esta estructura mecánica permite girar en el lugar, movimiento adelante y hacia atrás y rebasar obstáculos de aproximadamente 1.5 veces el diámetro de las ruedas. Las ruedas son de goma para mayor tracción y disminución del deslizamiento, cada una es manejada de forma independiente y son fijas al cuerpo del robot (Figura 2.1). La configuración está pensada para mantener una velocidad baja, de unos  $3 \text{ cm s}^{-1}$ , ya que si se aumenta mucho la velocidad el cuerpo del robot absorbe todo el impacto y es más difícil superar obstáculos. El cuerpo completo pesa alrededor de 1.5 lb. La plataforma se conoce con el nombre de *Bogie-Runt Rover*.

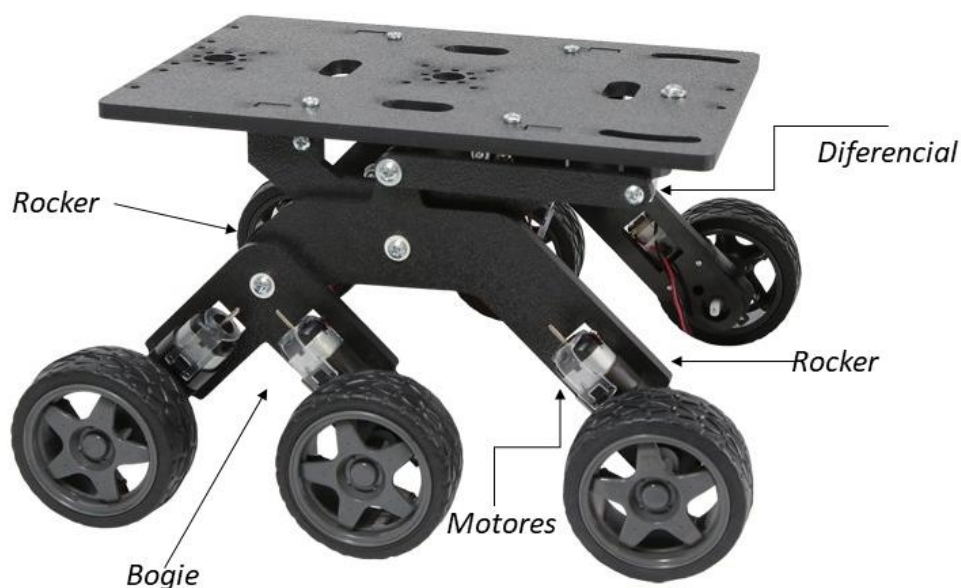


Figura 2.1: Estructura del robot.

### 2.2.1. Dimensiones

Las dimensiones en cm de la plataforma se muestran en la Figura 2.2. Estas son de mucha importancia a la hora de diseñar el controlador y el algoritmo de generación de trayectorias.

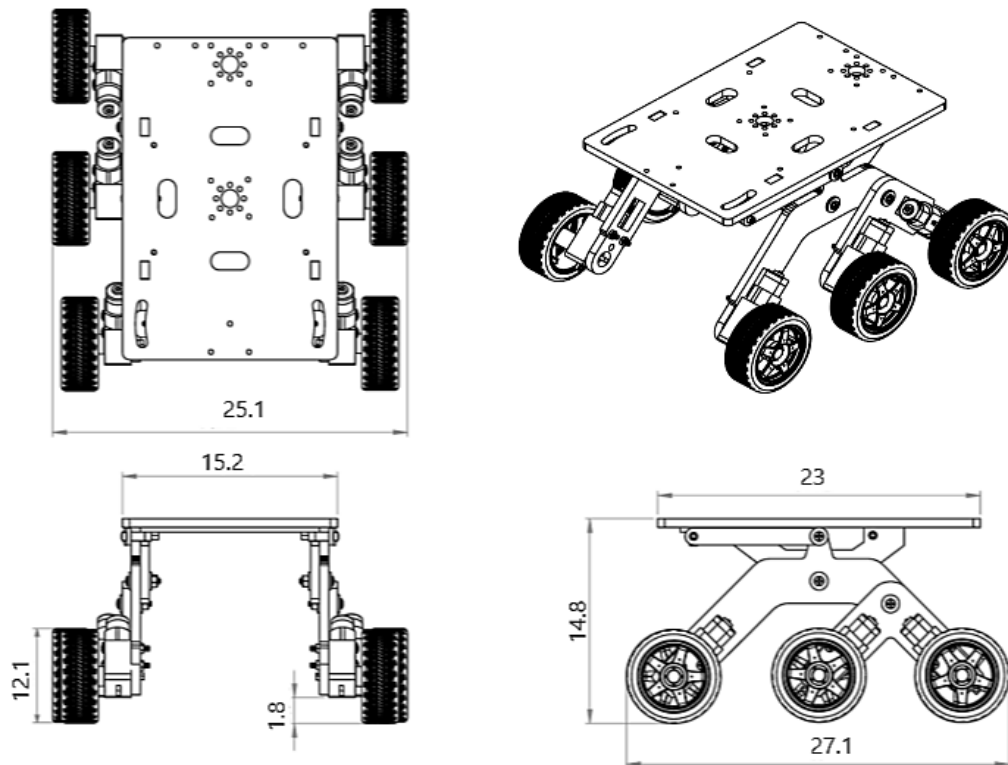


Figura 2.2: Dimensiones de la plataforma.

### 2.3. Motores

Para el movimiento de las ruedas se utilizaron seis motores de CC con caja reductora. El motor puede ser alimentado en un rango de tensión de 4.5-6 VDC. Serán alimentados con 5 VDC para mantener concordancia con el resto de los componentes del sistema. En la Figura 2.3 se muestra el motor y sus dimensiones. En la Tabla 2.1 se muestran las especificaciones obtenidas de [56].



Figura 2.3: Motor de CC utilizado para el movimiento de las ruedas.

Tabla 2.1: Especificaciones del motor utilizado.

Tensión (nominal)	4.5V
Rango de tensión (recomendado)	4.5V - 6.0V
Velocidad (Sin carga)	140RPM
Corriente (Sin carga)	190mA
Corriente (Carga máxima)	250mA
Par (parada)	800 gf-cm (11.11 oz-in)
Relación de transmisión	48:1
Material del engranaje	Plástico
Estilo de caja de cambios	Engranaje frontal
Tipo de motor	CC con escobillas
Diámetro del eje de salida	5.3mm
Soporte del eje de salida	Contenedor de plástico
Peso	0.07 lb

### 2.3.1. Controlador de motores

La dirección de los motores se controla mediante el circuito integrado TB6612 de Adafruit, Figura 2.4. Este circuito contiene dos puentes H para el control bidireccional de dos motores de corriente continua. Soporta hasta 1.2 A por puente y picos de 3 A durante 20 ms, opera en un rango de tensión de 4.5 a 13.5 V y de tensión lógico entre 3 y 5.5 V [57], por lo tanto, se puede utilizar sin problema para controlar los motores.

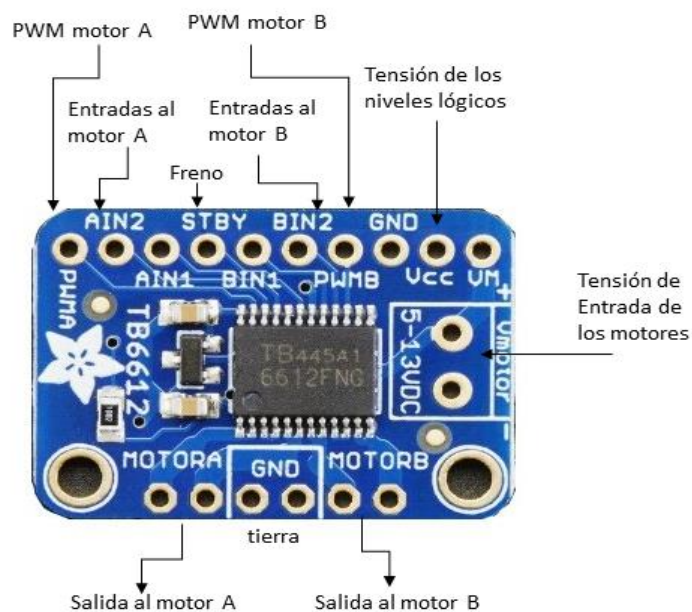


Figura 2.4: Controlador de motores TB6612.

Un Arduino Mega2560 es el encargado de realizar el control de velocidad y sentido de giro de los motores. Se emplea este microcontrolador, en adición al controlador principal, con el objetivo de aliviar esfuerzo y previendo la expansión de sus funciones en el futuro, agregando, por ejemplo, sensores de corriente y tensión para la batería, además por la cantidad de fuentes de interrupciones y pines PWM. Se seleccionó la plataforma Arduino por ser *open-source*, tener varios pines dedicados a PWM, la facilidad de programación, amplia información sobre su uso y la posibilidad de comunicación UART mediante USB. El Arduino Mega tiene 54 pines digitales de entrada/salida (de los cuales 15 se pueden usar como salidas PWM), 16 entradas analógicas, cuatro UART, un oscilador de cristal de 16 MHz, seis temporizadores y seis fuentes de interrupción externa [58].

## 2.4. Sensores

Para realizar las tareas de navegación y de reconocimiento del entorno se utilizan sensores que provean un estimado de la configuración espacial del robot.

### 2.4.1. *Encoder*

Se utilizaron *encoders* de cuadratura de efecto Hall, Figura 2.5, para realizar la odometría y la realimentación de velocidad de los motores, en la Tabla 2.2 se muestran las características. El sensor de efecto Hall utilizado es el TLE4946-2K de alta precisión. El motor se conecta a las entradas M1 y M2, luego A y B son las salidas del *encoder*. La placa debe ser soldada al motor, el eje pasa por el orificio central para ser acoplado al imán. Las características fueron obtenidas de [59].

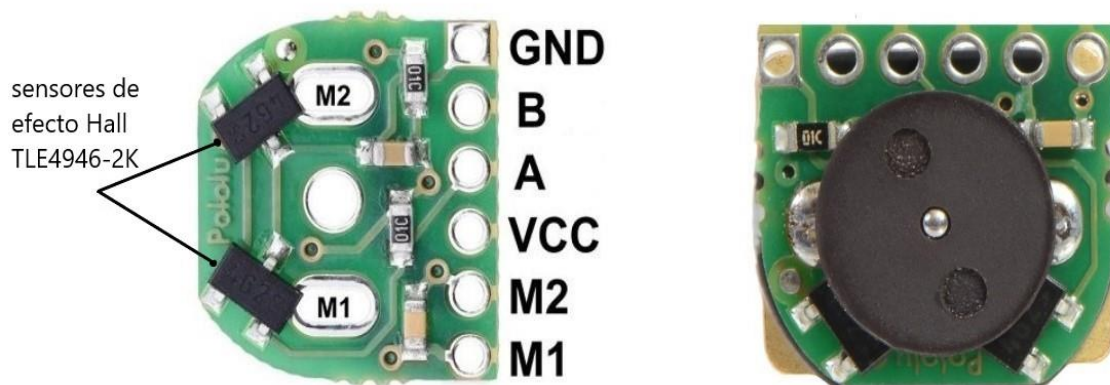


Figura 2.5: *Encoder* de cuadratura.



Tabla 2.2: Especificaciones del *encoder*

Rango de tensión (recomendado)	2.7-18 V
Resolución	12 pulsos por revolución
Temperatura de operación	-40-150 °C
Tamaño	10.6 mm × 11.6 mm
Peso	0.004 lb

### 2.4.2. *Sense-Hat*

Para estimar los ángulos de orientación del robot (*yaw*, *pitch* y *roll*) se utilizó el Sense-Hat para Raspberry Pi, Figura 2.6. Esta placa fue diseñada para aplicaciones espaciales y es utilizada actualmente en la Estación Internacional Espacial, ISS en inglés. Incluye una serie de sensores que permiten medir el entorno, una matriz de leds como visualizador y un *joystick* para interactuar con la placa. Incorpora sensores de presión, temperatura y humedad.

Además, posee una IMU (LSM9DS1) de 9 grados de libertad, que contiene un giróscopo, un acelerómetro y un magnetómetro de tres grados para la medición de aceleración, rotación y campos magnéticos en cada eje, estas mediciones se combinan para obtener el *pitch*, *yaw* y *roll* (ver figura 1.9) de la unidad.

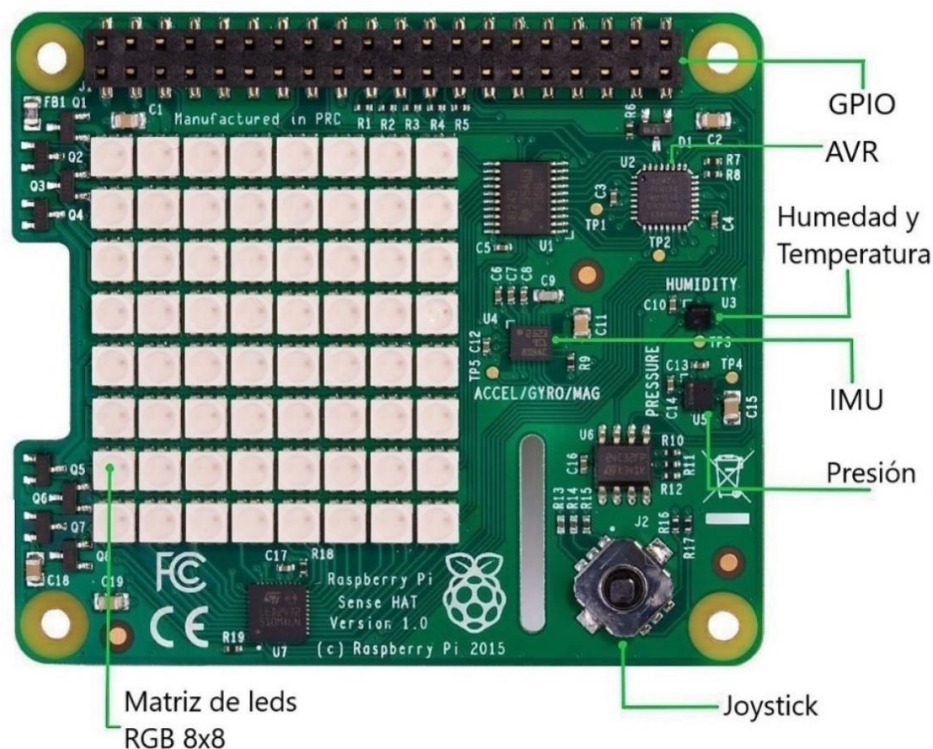


Figura 2.6: Sense-Hat.

Se seleccionó esta unidad por su compatibilidad con el procesador principal del sistema. Las bibliotecas que incluye permiten su fácil utilización y la obtención de los ángulos de orientación de forma muy sencilla. Incorpora un microcontrolador AVR 8-bits de 8 MHz en donde se realizan los cálculos para no cargar al procesador principal. La comunicación con la placa es por *I<sup>2</sup>C* (*Inter-Integrated Circuit*), lo que permite su manejo con solo dos pines. Sin embargo, no es necesario preocuparse por esto, ya que las librerías garantizan un manejo bastante cómodo de la placa.

## 2.5. Controlador principal y comunicación

El controlador del robot debe ser capaz de realizar las tareas de navegación, leer el estado de sensores, establecer la comunicación con una interfaz para la recepción y envío de datos y realizar tareas programadas o de contingencia. Se empleó la Raspberry Pi 3 modelo B, por su buena capacidad de procesamiento y bajo consumo de potencia, las facilidades de programación en varios lenguajes y bibliotecas útiles, los periféricos que puede utilizar, la capacidad de comunicación por WiFi y Bluetooth integrada y la relación calidad-precio. Además, se consideró la posibilidad de expandir las funcionalidades de la plataforma añadiendo la visión por cámaras, mediante una de las cámaras específicas para Raspberry Pi (PiCamera). Las características se muestran en la Tabla 2.3, obtenidas de [20].

La comunicación entre la Raspberry Pi y el Arduino se realiza mediante UART (*Universal Asynchronous Receiver-Transmitter*) sobre USB. El Arduino recibe los datos enviados por USB desde la Raspberry Pi y los convierte a UART para interpretarlos.

Para la comunicación con una estación se empleó el estándar 802.11 conocido como WiFi para realizar comunicación inalámbrica. La Raspberry Pi 3 tiene una antena integrada que permite la comunicación con los estándares 802.11 b/g/n.

Tabla 2.3: Características del Raspberry Pi 3 modelo B.

<b>SoC</b>	Broadcom BCM28367
<b>Arquitectura</b>	64 bits Quad-core ARM Cortex-A53
<b>CPU</b>	1.2GHz Quad-Core ARM Cortex-A53 802.11 b/g/n Wireless LAN y Bluetooth 4.1
<b>GPU</b>	Co-procesador multimedia Dual Core Video Core IV a 400 MHz capaz de llegar a los 28.8 GFLOPS de potencia de cálculo y compatible con OpenGL ES 2.0.
<b>Memoria</b>	1GB LPDDR2



<b>Sistema operativo</b>	Arranca desde la tarjeta Micro SD, ejecutando una versión del sistema operativo Linux o Windows 10 IoT
<b>Dimensiones</b>	85 x 56 x 17mm
<b>Potencia</b>	Micro USB 5 V, 2.5 A

## 2.6. Sistema de generación y almacenamiento de energía

La autonomía del *rover* no debe estar sujeta solamente a la navegación. Según la ECSS, debe ser capaz de sobrevivir durante un tiempo determinado sin intervención del hombre. Para lograr esto contará con un sistema de generación y almacenamiento de energía basado en baterías recargables y paneles solares.

El *rover* operará durante las horas de mayor incidencia de sol, el resto del tiempo estará en modo de hibernación. Durante la hibernación el movimiento y las funciones de navegación estarán suspendidas para reducir lo mayor posible el consumo de energía y recargar las baterías. Durante la noche la batería debe estar cargada en caso de que surja cualquier contingencia y deban reanudarse las funciones. No se contempla la adición de un reloj de tiempo real ni el conteo de tiempo, sin embargo, es de suma importancia en estos análisis, ya que el robot debe mantener conocimiento de la hora para saber cuándo debe reanudar la operación y cuándo suspenderla.

Un robot potenciado por energía solar se puede considerar un sistema aislado. Los sistemas fotovoltaicos aislados se caracterizan por utilizar paneles solares como fuente primaria de energía y baterías para el almacenamiento. En general, un sistema de energía fotovoltaico aislado, que opere con corriente continua, está compuesto por paneles solares, baterías y un regulador de carga.

La eficiencia de los paneles solares no es muy alta por lo que para garantizar el funcionamiento de los módulos fotovoltaicos en el punto de máxima potencia (punto de mayor generación de potencia del panel para las condiciones medioambientales del momento) es común utilizar un algoritmo de control llamado MPPT, *Maximum Power Point Tracking*, por sus siglas en inglés. Otra forma es hacer que los paneles busquen el punto de máxima potencia, esto lleva un mayor control automático y el diseño de un sistema mecánico y eléctrico para garantizar la rotación y movimiento de los paneles.

Los sistemas aislados pueden disponerse en serie o en paralelo. Un sistema en serie tiene como desventaja que toda la energía generada va directamente a la batería y la batería se encarga de suministrar potencia a la carga, descargándose y cargándose continuamente lo que disminuye el tiempo de vida de la batería. El regulador de carga se encarga de controlar la tensión de alimentación de las baterías y, además, garantiza el punto de máxima potencia de los módulos. Los sistemas en paralelo emplean las baterías en paralelo con el flujo de energía del sistema, la diferencia radica en que, en esta configuración, los paneles pueden alimentar también la carga, evitando que la batería se descargue más frecuentemente, las baterías se accionarán cuando la energía generada por los paneles sea inferior a la demandada por la carga.

### 2.6.1. Consumo de energía

En la Tabla 2.4 se describe el consumo potencia de los componentes del rover. Se asumió la corriente máxima de los motores. El consumo de la Raspberry Pi 3B se asume sin la conexión de ningún periférico, utilizando los cuatro núcleos y la comunicación WiFi, la información del consumo se obtuvo de los resultados de varias pruebas publicadas en el sitio oficial [60]. Para calcular la potencia del Sense-Hat se suman los consumos de los sensores que componen la IMU y el microcontrolador incorporado. La potencia del acelerómetro, el magnetómetro y los sensores de temperatura, presión y humedad se desprecia ya que las corrientes están en el orden de los micro amperes quedando de la siguiente forma  $Potencia_{SenseHat} = Potencia_{AVR} + Potencia_{IMU}$ . La potencia total requerida es de 12 W y la corriente consumida es de 2300 mA.

Tabla 2.4: Consumo de potencia de los componentes

Componente	Tensión	Corriente	Cantidad	Potencia total
<b>Motores</b>	5 V	250 mA	6	7.5 W
<b>RPi + antena</b>	5 V	650 mA	1	3.25 W
<b>Arduino</b>	5 V	80 mA	1	400 mW
<b>Sense-Hat</b>	3.3 V	53 mA	1	175.5 mW
<b>Encoders</b>	5 V	4 mA	6	120 mW
<b>Total</b>				12 W

## 2.6.2. Almacenamiento de energía

Como la energía solar es variable y depende grandemente de las condiciones atmosféricas, se emplea un sistema de almacenamiento de energía para el funcionamiento en los periodos de poca iluminación.

La tensión de operación de todo el sistema es de 5V CC, por lo que con una sola fuente de alimentación capaz de entregar esta tensión y la corriente necesaria sería suficiente. Se seleccionó un PowerBank de 20000 mAh de tres puertos de salida, con salida de 5V CC regulados por USB. Este PowerBank permite extraer un máximo de 6Ah y 3Ah por puerto USB. Esto permite utilizar una de las salidas para la alimentación de la Raspberry Pi y Arduino y otra para los motores, proporcionando la misma tensión a los componentes, pero con entradas independientes. El consumo de corriente del sistema es de 2300mAh, por lo que esta batería otorga una autonomía de aproximadamente 8 horas. De esta forma se elimina la necesidad de incluir una etapa de regulación de tensión y un controlador de carga para la batería.



Figura 2.7: Batería del robot.

## 2.6.3. Generación de energía

La configuración de un sistema de energía fotovoltaico aislado puede ser en serie o en paralelo como se explicó al inicio del capítulo, dependerá de si el panel puede proporcionar suficiente corriente para alimentar el sistema. Para la aplicación se necesitaría un panel solar capaz de entregar 2300mA, los encontrados capaces de suministrar esta corriente son muy grandes para ser montados en el *rover* o muy costosos, por lo que se seleccionó un panel solar de 5V y capaz de entregar 1000mA por USB para hacerlo compatible con la entrada del PowerBank. De esta forma se tiene un sistema en serie, donde el panel carga la batería y la batería alimenta el

circuito. El panel se muestra en la Figura 2.8 y las características, obtenidas de [61], en la Tabla 2.5.

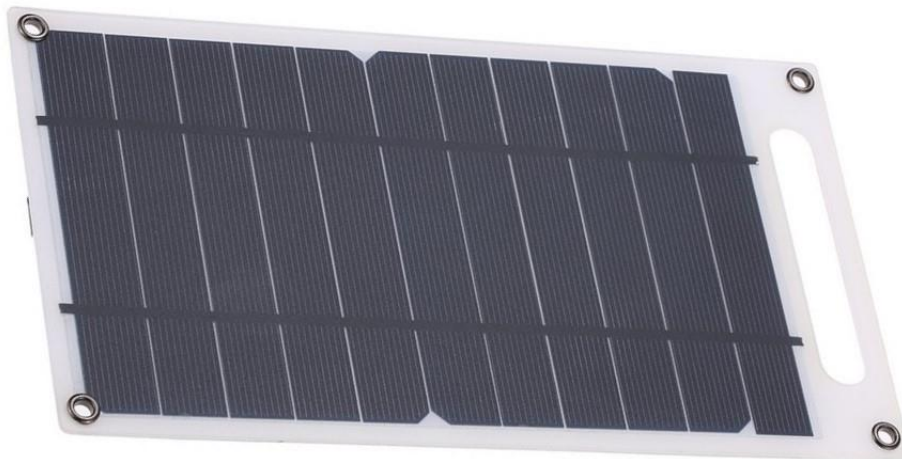


Figura 2.8: Panel solar con regulador interno.

Tabla 2.5: Características del panel solar

Salida del panel	6V 1300mA
Potencia	7.8W
Eficiencia de conversión	20%
Salida USB	5V 1000mA
Material	Silicio Monocristalino
Dimensiones	22cm x 12.6 cm
Peso	127g

#### 2.6.4. Esquema general

El robot debe funcionar en las horas de mayor incidencia solar y debe asegurarse que las baterías queden cargadas luego del periodo de operación para operar con poca iluminación en caso que sea necesario. Para asegurar la carga completa de las baterías, se considera que el *rover* opera durante 2 horas diarias solamente, de 10am a 12pm, lo que da un consumo de 4600mA diarios, si se consideran 5 horas de sol pleno, el panel será capaz de entregar 5000mA diarios, lo cual es suficiente para cargar las baterías completamente antes de la noche. Para un mayor tiempo de operación es necesario un panel solar que pueda suministrar más corriente. Si un panel con estas características fuera utilizado en Marte, por ejemplo, tardaría casi el doble del tiempo en cargar totalmente la batería, es decir 10 horas, ya que la radiación por metro cuadrado es casi la mitad de la incidente en la Tierra, como se explicó en

la sección 1.7.1. Este problema puede ser resuelto cambiando el material de las celdas de Silicio Monocristalino a Arseniuro de Galio para aumentar la eficiencia a casi el doble, de esta forma se compensaría el problema de la intensidad de radiación. En la Figura 2.9 se muestra el esquema de conexiones general del sistema.

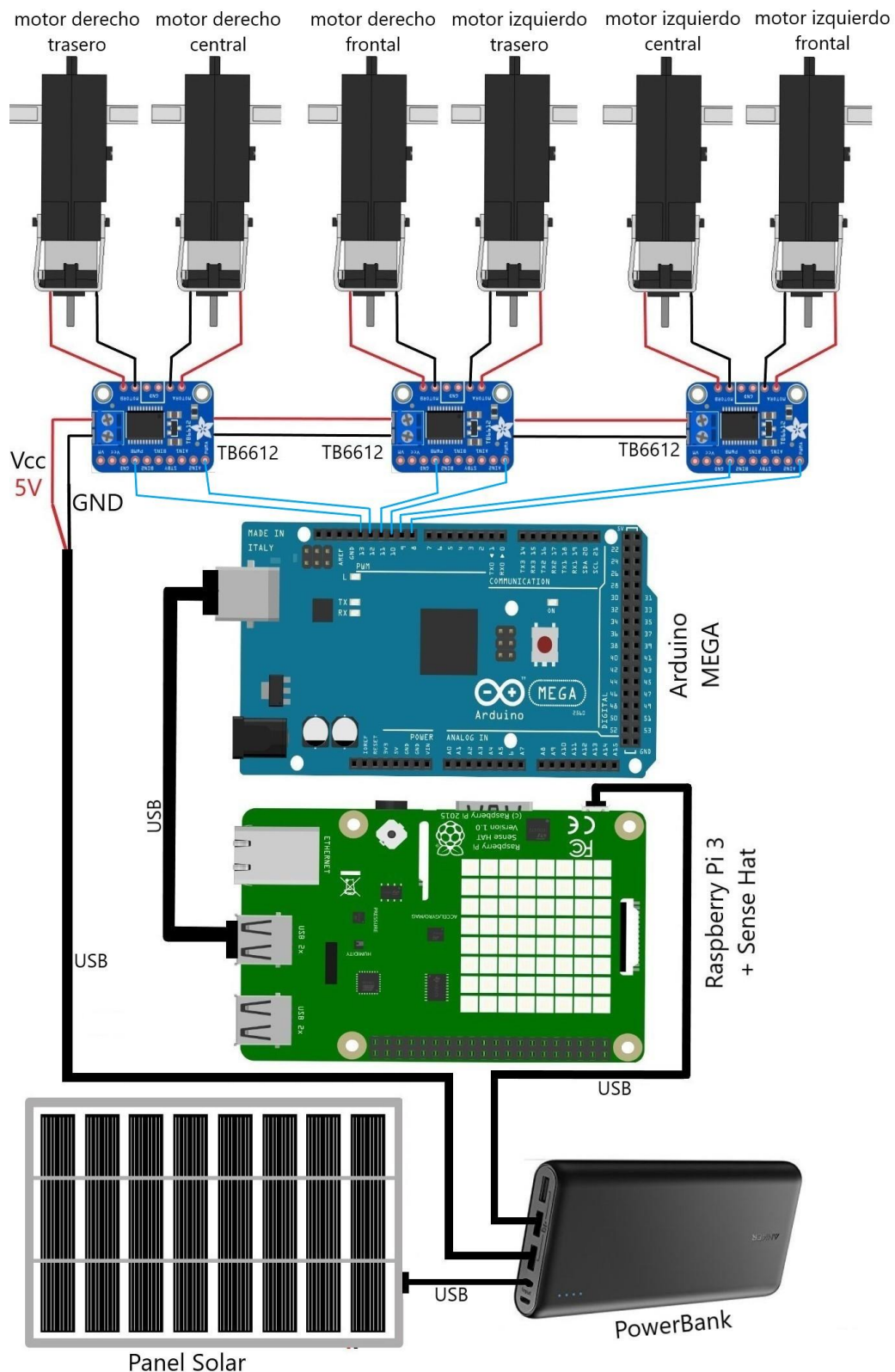


Figura 2.9: Diagrama general del sistema.

El panel solar se conecta al PowerBank mediante USB por uno de los puertos de entrada de este, proporcionando 5 VCC y corriente variable en dependencia de la intensidad solar incidente. Una de las salidas del PowerBank alimenta la Raspberry Pi, que a su vez comparte la alimentación con el Sense-Hat y el Arduino Mega mediante otro cable USB, por el cual, también se realiza la comunicación serie entre los dispositivos. El Sense-Hat se acopla a la Raspberry Pi mediante el GPIO y unos tornillos en las esquinas para mantener fija su posición y no incurrir en errores en la medición. La comunicación entre ambos se realiza mediante los pines SDA y SCL por I2C, el resto del GPIO se mantiene disponible a través del Sense-Hat.

El Arduino Mega se conecta a los tres controladores TB6612, mediante sus pines digitales. Para cada controlador se necesitan 5 pines del Arduino, lo cual hace un total de 15 pines digitales. Se utilizan 6 pines dedicados a PWM hardware del Arduino para regular la velocidad de los motores. Los 6 motores, 3 por cada lado del robot, se conectan a los TB6612, cada uno controla dos motores. En el esquema no se muestran los *encoders*, sin embargo, estos se encuentran acoplados al eje de cada motor por la parte contraria al reductor, el eje pasa por el centro del disco magnético y por el centro del circuito integrado, quedando entre los dos sensores de efecto Hall, como se muestra en la Figura 2.5. Las salidas de los *encoders* también se conectan al Arduino a través de los pines digitales.

## 2.7. Programa

### Arduino

El Arduino deberá encargarse de mantener la velocidad deseada en las ruedas para seguir la trayectoria correctamente. El controlador de seguimiento de trayectoria provee dos vectores de velocidad,  $u_1, u_2$ , que corresponden a las velocidades angular y longitudinal de la plataforma, ver Figura 2.10, luego se transforman a las velocidades angulares deseadas de las ruedas derechas e izquierda, obteniéndose  $\omega d_d$  y  $\omega d_i$ . Como el robot tiene tres ruedas en cada lado (trasera, media y frente), las velocidades de referencia se dividen entre 3 para lograr que cada rueda mantenga la misma velocidad. Cada rueda tendrá una velocidad de  $1/3$  de la de referencia correspondiente a su lado, esto se debe a que las velocidades de referencia son calculadas para un robot con estas dimensiones, pero de una sola rueda a cada lado. Esto disminuye la carga de los motores para una misma fuerza de empuje del robot, lo cual es una característica de la configuración Rocker Bogie.

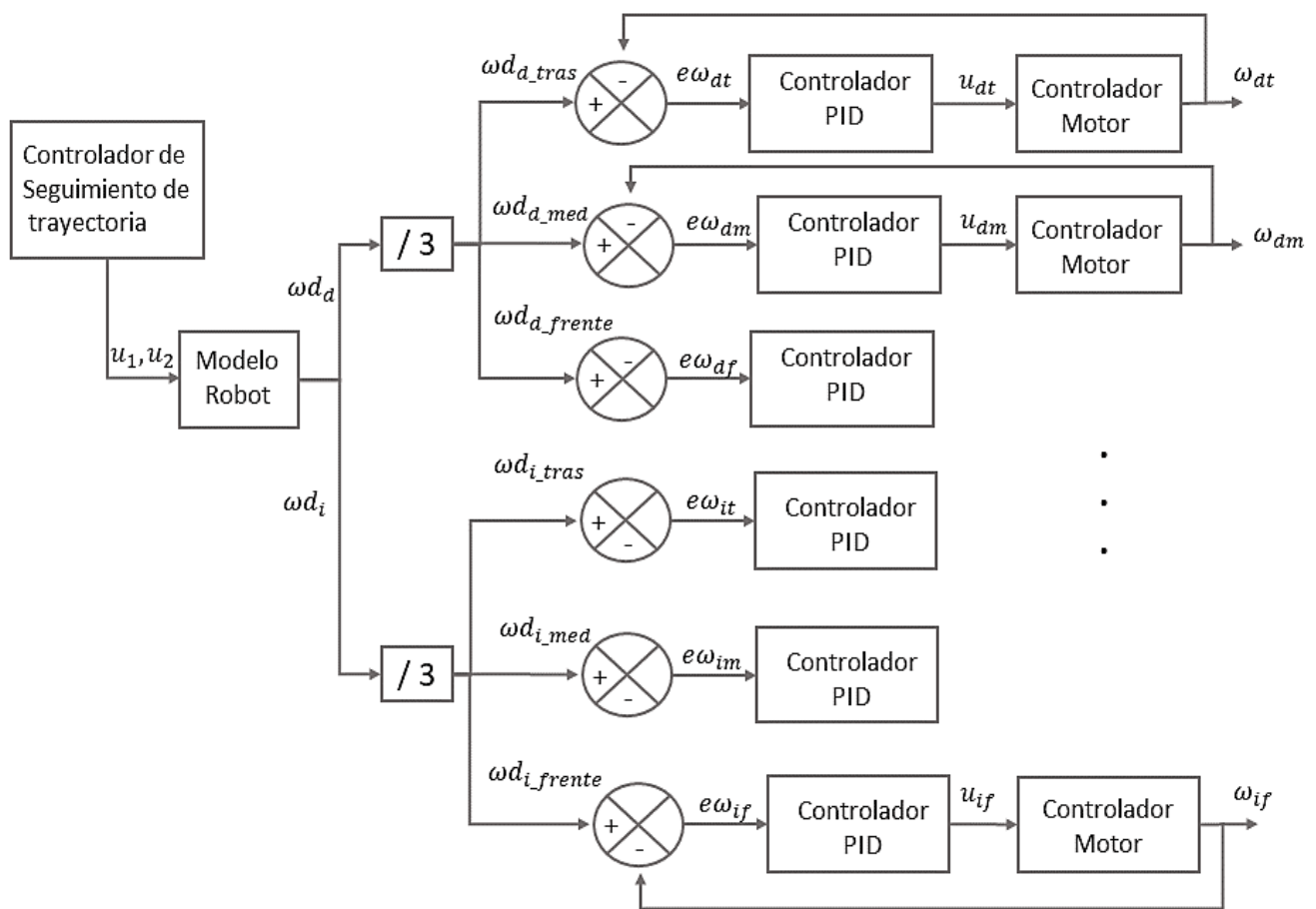


Figura 2.10: Esquema de control de velocidad.

Las velocidades de las ruedas entran como referencia al lazo de control de tiempo real para compararse con la velocidad medida mediante la lectura de los *encoders*. La atención de los *encoders* puede realizarse mediante interrupciones por frentes de subida. En cada interrupción se chequea el estado del otro canal, si es cero, se aumenta el contador, de lo contrario se decrementa, esto permite que se mantenga la cuenta independiente del programa principal. Los *encoders* propuestos tienen una resolución de 12 pulsos por revolución, y los motores una relación de transmisión de 48:1, por lo que la cantidad de pulsos de subida del *encoder* equivalentes a una vuelta de la rueda es de 576.

El Arduino realiza el control de velocidad de los motores mediante PWM cada 30 ms, por lo que se utiliza una demora software de 30ms, luego deberá leer el estado de los contadores para calcular la posición y la velocidad angular real mediante las ecuaciones 1.8 y 1.9. Con la velocidad real y la deseada se obtiene el error, mediante la ecuación 1.10, que se utiliza para el controlador PID que proporciona el nuevo valor de ciclo útil para la señal PWM, esto se determina para cada uno de los motores. El

esquema de control de motores se muestra en la Figura 2.10 y el diagrama de flujo del programa se muestra en la Figura 2.11.

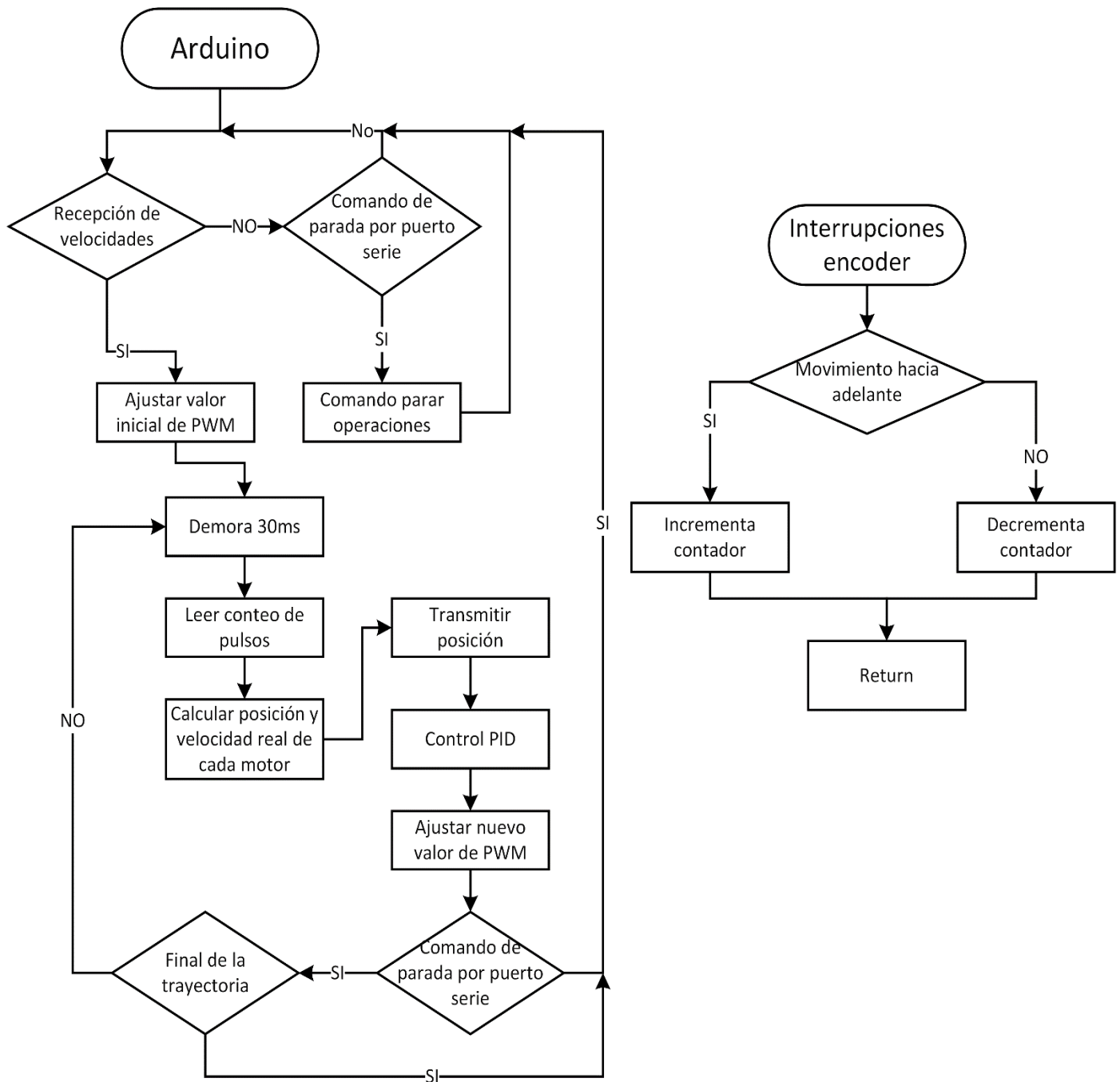


Figura 2.11: Diagrama de flujo del programa en el Arduino.

La programación del Arduino se realiza en el entorno de desarrollo integrado (*Integrated Development Environment: IDE*) Arduino versión 1.8.4. El software está basado en el entorno de Processing, de código abierto, y lenguaje de programación basado en Wiring. El microcontrolador se programa mediante una conexión USB con la PC.



## Raspberry Pi

El programa principal se realiza en la Raspberry Pi. Esta se mantendrá a la espera de un mapa para realizar la planificación, generación y el control de seguimiento de trayectoria, luego envía las velocidades angulares de las ruedas al Arduino por puerto serie. Una vez que el Arduino recibe los datos, se encargará de realizar el movimiento. El Arduino enviará los datos de la posición de los *encoders* a lo largo de la trayectoria a la Raspberry Pi. Cada vez que se reciba un dato del Arduino, se calculan los valores de los ángulos de orientación mediante el Sense-Hat. En la Figura 2.12 se puede ver lo explicado anteriormente.

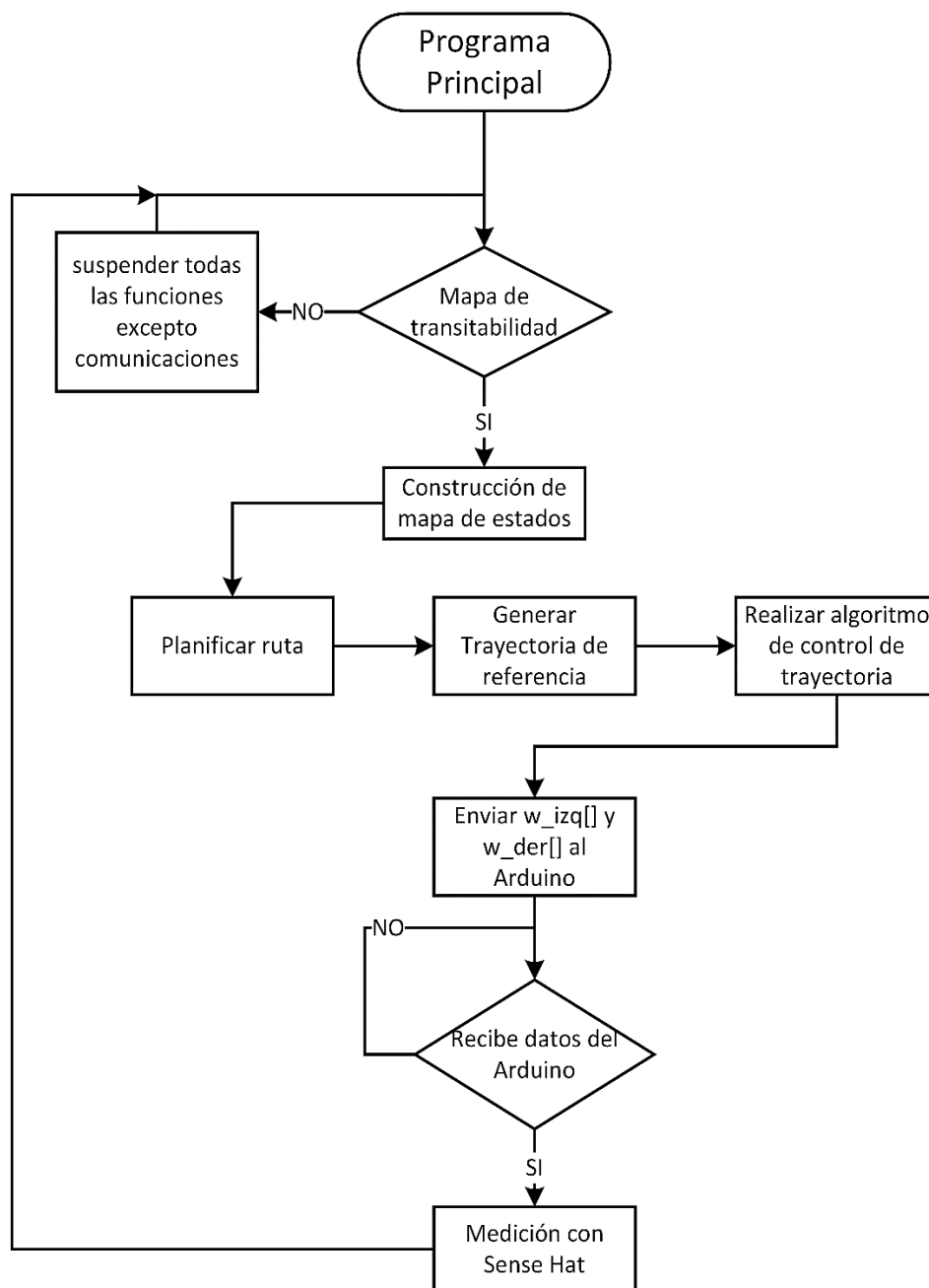


Figura 2.12: Diagrama de flujo del programa en la Raspberry Pi.

La programación en la Raspberry Pi se realiza en Python 3.3.2 con el IDE JetBrains PyCharm Pro 2017.2.3 de IntelliJ IDEA. La Raspberry Pi crea un punto de acceso (*hotspot*) al que se conecta la PC, luego se establece la conexión por SSH (*Secure Shell*) mediante el propio PyCharm. Una vez establecida la conexión se tiene total control de la Raspberry Pi mediante la consola. El código es compilado en el PyCharm, el cual ofrece muchas facilidades para programar en Python, y automáticamente se copia a un directorio de la Raspberry Pi para ser ejecutado. De esta forma se puede hacer uso de todos los recursos de la RPi sin tener que conectar un monitor y un teclado.

## 2.8. Conclusiones

La propuesta de diseño garantiza un robot con el sistema de locomoción capaz de superar obstáculos, moverse por terrenos irregulares, la capacidad de medir los ángulos necesarios para la evaluación del terreno y realizar la odometría móvil mediante *encoders*. El sistema de generación y almacenamiento de energía desarrollado permite el movimiento del *rover* sin necesidad de la intervención del hombre. Los componentes seleccionados garantizan la capacidad de una futura expansión de las capacidades del robot, haciendo posible la adición de cámaras compatibles con las Raspberry Pi para realizar el proceso de mapeo y localización mediante visión. Los controladores utilizados son muy fáciles de reemplazar, por lo que cambiar el diseño es relativamente fácil.

## Capítulo 3. Algoritmos de navegación

### 3.1. Introducción

En este capítulo, se explican los algoritmos utilizados para realizar la navegación autónoma, la cual se descompone en la planificación de la trayectoria, la generación de curvas, para mejorar el camino planificado, y el control de seguimiento de trayectoria. Los resultados obtenidos son mostrados durante cada una de las etapas.

### 3.2. Consideraciones previas

La estación en tierra proporciona un mapa general con algunos puntos de referencia para ayudar a la localización y, además, un punto de interés hacia el cual el *rover* tendrá que navegar. El mapa proporcionado por la estación no es lo suficientemente bueno como para realizar la planificación de forma segura, por lo tanto, es necesario que el *rover* construya un mapa de su entorno.

Una vez construido el mapa y realizada la localización, el *rover* deberá analizar el terreno, esto incluye los análisis de cinemática inversa, análisis de fuerza y análisis de las características claves del terreno para construir un mapa de transitabilidad. Una vez construido el mapa de transitabilidad, se realiza la transformación al C-espacio mediante la descomposición en celdas para aplicar un algoritmo de planificación.

A continuación, se asume que las tareas de localización y construcción de un mapa de transitabilidad ya han sido realizadas a bordo del robot, por lo que se cuenta con un mapa representado en el C-espacio en el que ya se puede aplicar un algoritmo de planificación. La Figura 3.1 muestra las etapas de navegación. Para las simulaciones se utilizó el software MATLAB® R2017b, versión 9.3.0.713579.

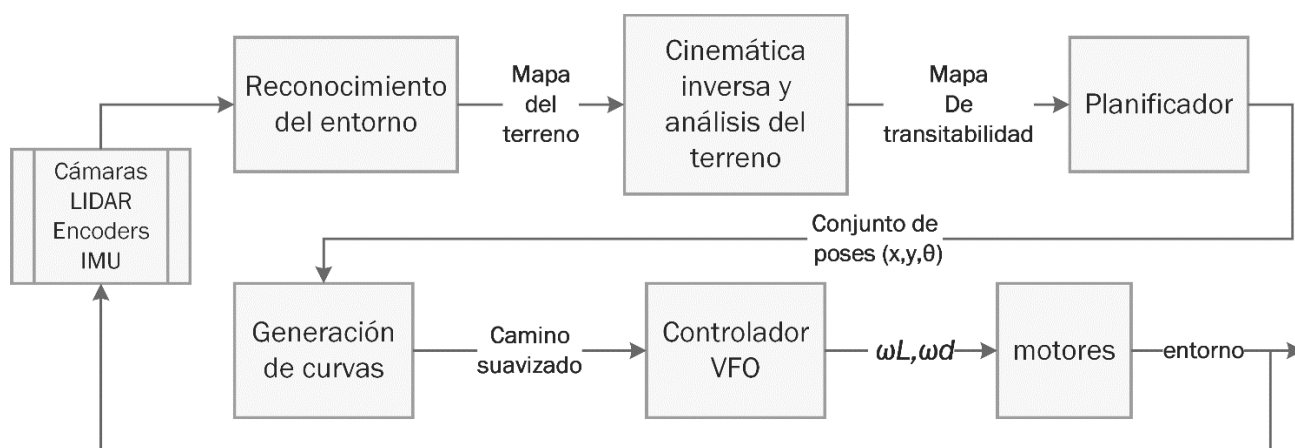


Figura 3.1: Etapas del proceso de navegación.

### 3.3. Planificación de trayectoria

El mapa del entorno se transformó al C-espacio mediante la descomposición en celdas cuadradas. Para evitar obstáculos se considera un sobredimensionamiento de estos, agregando el factor de expansión del entorno. El robot será modelado como un círculo de radio  $r = 30 \text{ cm}$  siendo este el factor de expansión, ver Figura 2.2 para las dimensiones. La distancia entre las celdas deberá ser mayor que el radio para asegurar que el *rover* no se acerque demasiado a un estado inválido. El mapa será de  $1200 \times 1200 \text{ cm}$  y la resolución  $Re = 66 \text{ cm}$ , quedando un matriz de estados de  $18 \times 18$ .

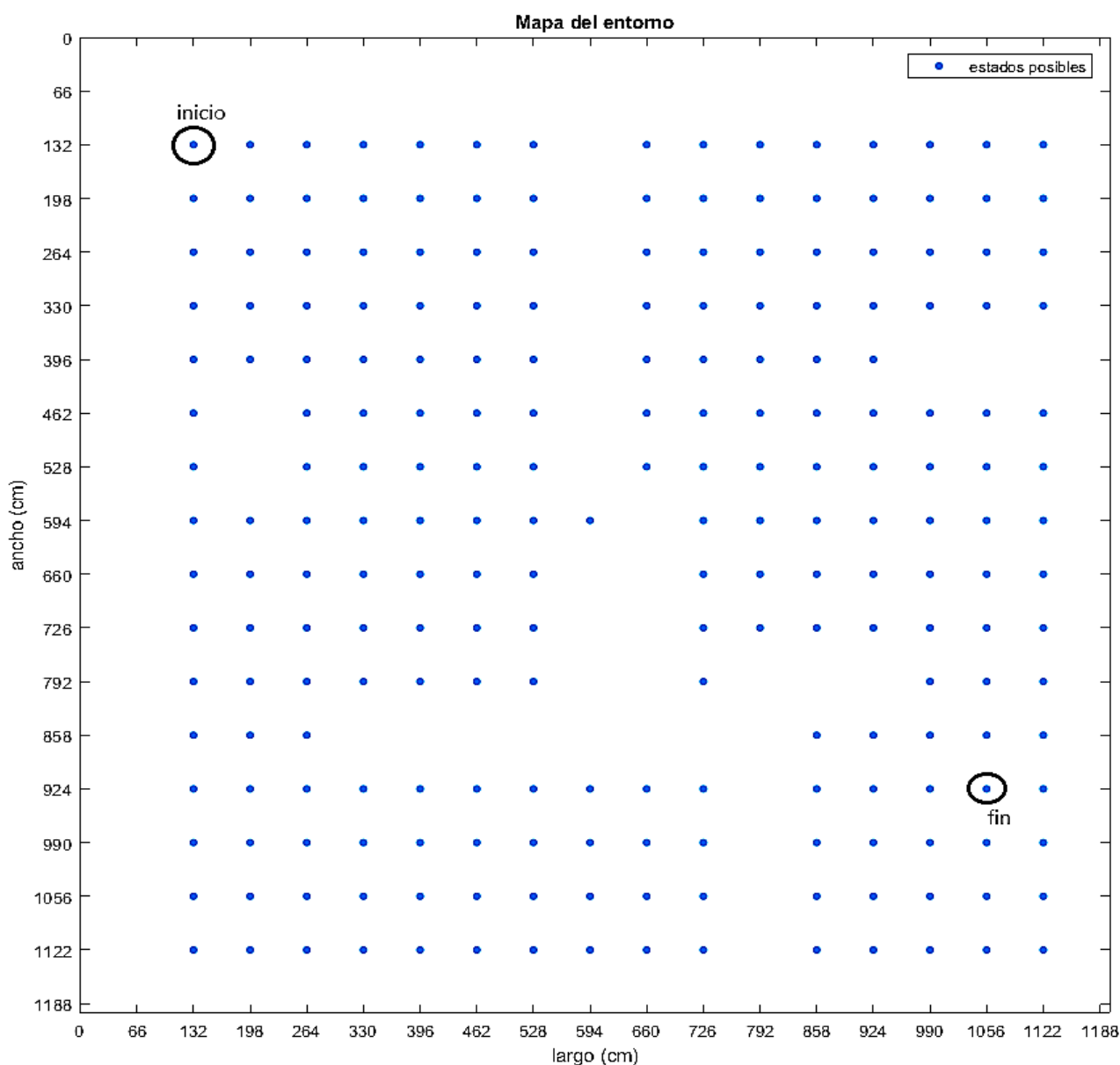


Figura 3.2: Mapa de estados.

La Figura 3.2 muestra el mapa del entorno representado mediante la descomposición en celdas, cada localización del mapa tiene dos posibles valores: 1 si es un estado libre (punto), y 0 si es un estado prohibido (espacio en blanco).

Las coordenadas de interés son las siguientes:

- Inicio: 132 *cm*; 132 *cm*
- Fin: 1056 *cm*; 924 *cm*

### 3.3.1. Algoritmo de Bellman-Ford

El algoritmo de planificación utilizado es el de Bellman-Ford descrito en [7, 8, 36, 62] para programación dinámica. Las ventajas de la utilización de programación dinámica, en este tipo de problemas, es que el camino entre dos puntos siempre será óptimo debido al principio de optimalidad de Bellman y, además, se puede utilizar cualquier función de costo u optimización [8].

Este es un método de búsqueda en grafos que calcula el camino más corto entre un vértice fuente y el resto de los vértices o estados. Es un método de optimización combinatorio que de forma recursiva va evaluando posibles soluciones a partir de una función objetivo que optimiza un criterio determinado. Primero calcula las distancias más cortas que tienen una arista en el camino, luego calcula las rutas más cortas con 2 aristas, y así sucesivamente hasta  $|V|-1$ , donde  $V$  es el conjunto de vértices.

En el caso de un robot móvil, los estados o vértices corresponden a las posibles localizaciones pertenecientes al C-espacio. Las localizaciones podrán ser definidas de la siguiente forma:

- Localizaciones prohibidas: estados por los que no podrá transitar el robot, ya sea por la existencia de un obstáculo o por otras condiciones preestablecidas.
- Localizaciones permitidas: el robot podrá ocuparlos sin problemas.

Su aplicación a un robot móvil en el C-espacio puede ser representada de la siguiente forma:

$$\delta_0 = \begin{cases} 0, & \text{si } v = s \text{ y } v \notin P \\ \infty, & \text{si } v \neq s \text{ o } v \in P \end{cases} \quad (3.1)$$

$$\delta_k = \begin{cases} \min\{\delta_{k-1}(v), \min_{\omega \in N^-(v)}(\delta_{k-1}(\omega) + C_{v\omega})\}, & \forall v \notin P \\ \infty, & \forall v \in P \end{cases} \quad (3.2)$$

- $P$  es el conjunto de estados prohibidos.
- $\delta_k$  es la función objetivo.
- $N^-(v)$  es el conjunto de vecinos del vértice  $v$ .
- $S$  el estado inicial.
- $v, w$  son dos estados adyacentes cualesquiera.
- $C_{vw}$  es el costo de ir del estado  $w$  al  $v$ .

El algoritmo fue diseñado para calcular la trayectoria más corta según la distancia. Para el caso de un robot móvil esto es muy útil, sin embargo, permite cambiar la función objetivo por otra sin perder generalidad, por ejemplo, se pueden considerar condiciones de terreno, tiempo, potencia consumida o una combinación.

La función de costo seleccionada es la distancia euclidiana entre los vértices, por tanto, se seleccionará la ruta óptima basada en la distancia entre los estados en dependencia de los movimientos que se puedan realizar. El pseudo-código puede ser descrito de forma siguiente:

*Determinar para cada  $v \in V$ :*

$$\begin{cases} 0, & \text{si } v = s \text{ y } v \notin O \\ \infty, & \text{si } v \neq s \text{ o } v \in O \end{cases}$$

*Para  $i$  desde 1 hasta  $|V| - 1$ :*

*Para cada  $(v, w)$  en  $V$ :*

*Calcular  $C_{vw}$  del movimiento:  $C_{vw} = \text{euclid}(v, w)$*

*relajar:*

$$\text{Si } w.d > v.d + C_{vw} \text{ y } w \notin P$$

$$w.d = v.d + C_{vw}$$

$$\Pi(v) = w$$

*Fin*

*Para cada  $(v, w)$  en  $V$ :*

*Si  $w.d > v.d + C_{vw}$*

*existe un ciclo negativo y el camino no puede ser definido*

*si no, el camino más corto estará definido en  $\Pi$ ,*

*return  $\Pi$*

Donde  $euclid(v, w)$  es una función que calcula la distancia euclidiana entre los estados  $v$  y  $w$ ,  $v.d$  es el peso de  $v$ ,  $w.d$  es el peso de  $w$ ,  $O$  es el conjunto de estados prohibidos y  $C_{vw}$  es el costo del movimiento.

En un entorno descompuesto en celdas cuadradas solamente hay ocho posibles movimientos, entonces el costo  $C_{vw}$  varía en dependencia en dependencia del movimiento analizado, estos se muestran en la Figura 3.3.

Primero se inicializan todos los estados,  $\forall v \in V$  el estado inicial  $S$  tendrá peso cero y el resto infinito. Luego se definen los movimientos permitidos y sus costos y se calcula la distancia mínima a cada uno de los vértices mediante el proceso de relajar las aristas. Se comparan los pesos del vértice  $v$  con el de su adyacente  $w$  (o adyacentes) y el costo de ir de  $w$  a  $v$ , siempre que el estado vecino no sea uno prohibido. El camino más corto desde el inicio hasta una posición objetivo se conforma con la unión de los vértices padres en  $\pi$ . Por último, se realiza una búsqueda de todos los estados para buscar ciclos negativos, si se encuentra algún ciclo negativo no se podrá definir el camino. Una vez termine el algoritmo se podrá encontrar el camino más corto hasta cualquier punto del terreno siempre que este no sea un obstáculo. En la Figura 3.4 se muestra el camino planificado por el algoritmo.

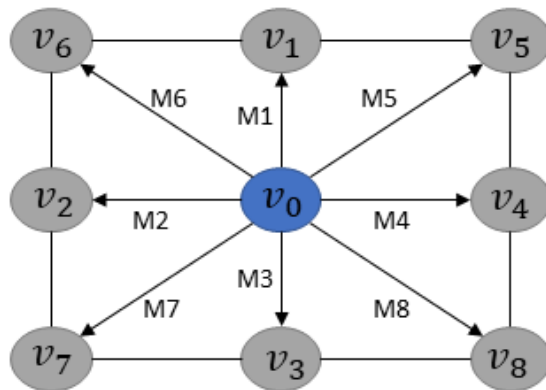


Figura 3.3: Movimientos posibles en un entorno descompuesto en celdas cuadradas.

La ruta planificada tiene muchos cambios bruscos en la orientación, como se muestra en la Figura 3.4, lo que provocaría que el robot tenga que detenerse para girar sobre su eje, por lo que se incorpora una etapa de generación de trayectoria. No se consideran las orientaciones inicial y final ya que el robot tiene la habilidad de girar en el lugar, por lo que solo con un giro puede ajustar su orientación en caso de no coincidir con la de la trayectoria.

La trayectoria es segura debido a que se consideraron las dimensiones del robot, al ser la resolución igual a  $66\text{ cm}$  la distancia entre estados es de  $66\text{ cm}$  y por tanto mayor que el radio del robot.

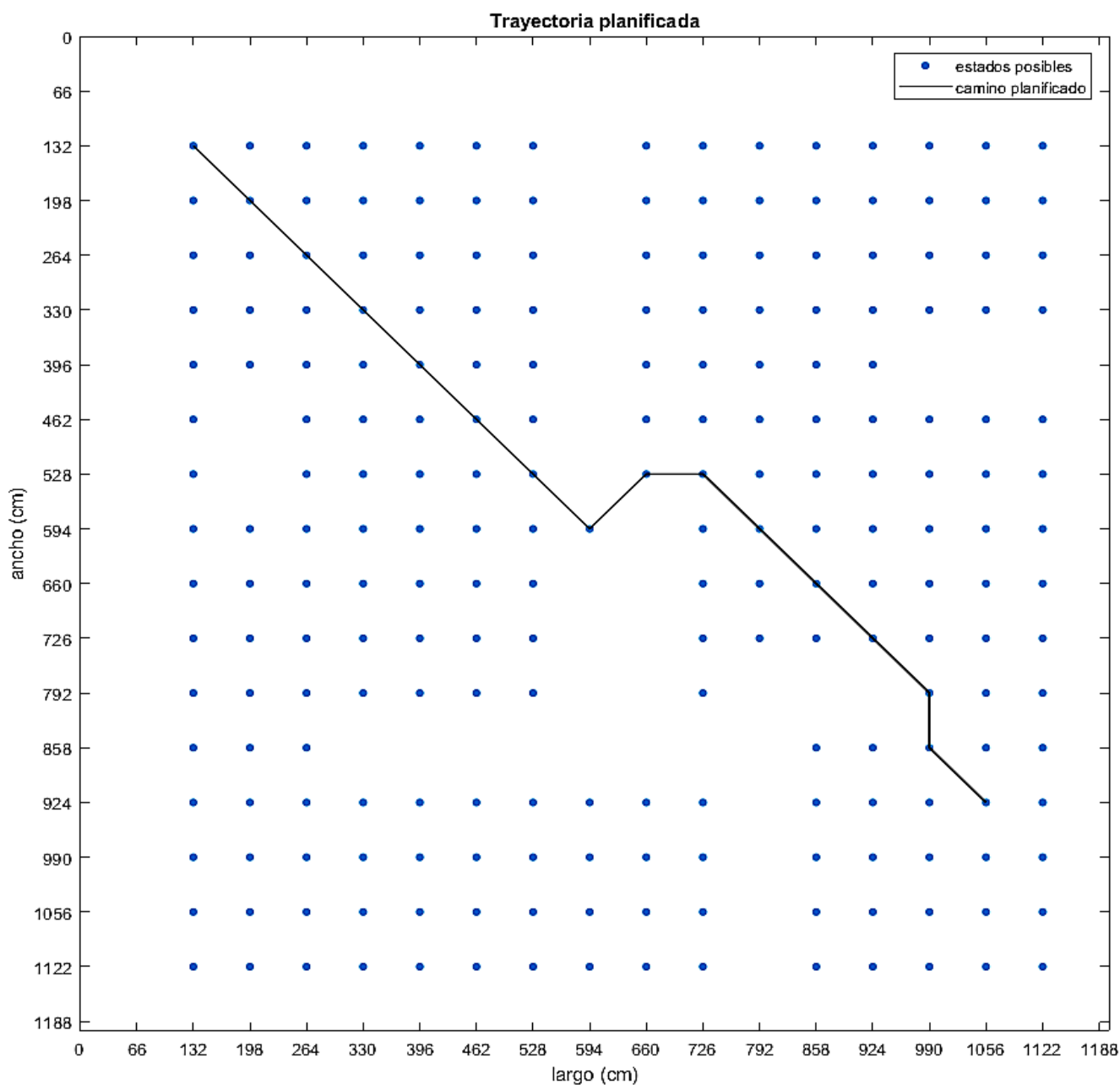


Figura 3.4: Trayectoria planificada.



Al analizar la trayectoria planificada se puede ver que siempre se selecciona el camino más corto, en dependencia de la dirección en que se realiza la búsqueda en el grafo, en este caso se realizó la búsqueda de izquierda a derecha y de arriba hacia abajo, con relación al mapa.

### 3.4. Generación de trayectoria

Una vez planificada la trayectoria se pasa a la etapa de generación para lograr cambios continuos de orientación, optimizando el camino planificado. Se utilizarán las curvas clotoides para la generación de trayectorias mediante el procedimiento descrito en [7, 9, 40, 63], estas curvas garantizan un camino suave, con continuidad en la orientación y posición para lograr mayor eficiencia en el camino del robot.

#### 3.4.1. Curvas clotoides

Las clotoides son conocidas también como espiral de Cornú o espiral de Euler, son curvas ampliamente utilizadas en el sector de transporte, principalmente en las carreteras para darle a los vehículos recorridos seguros [64].

Son curvas que varían su curvatura linealmente con la longitud del arco recorrido, ver Figura 3.5, pueden ser expresadas mediante las siguientes ecuaciones:

$$k(s) = \sigma s + k_0 \quad (3.3)$$

$$\theta(s) = \int_0^s k(s) ds \quad (3.4)$$

$$x(s) = \int_0^s \cos \theta(s) ds \quad (3.5)$$

$$y(s) = \int_0^s \sin \theta(s) ds \quad (3.6)$$

Donde  $k$  es la curvatura y  $\sigma$  es la razón de cambio de la curvatura o *sharpness*,  $\theta$  y  $(x, y)$  son el ángulo de orientación y la posición de cualquier punto sobre la curva, respectivamente. Esta forma de representar una curva es conocida como representación natural de una curva [41].

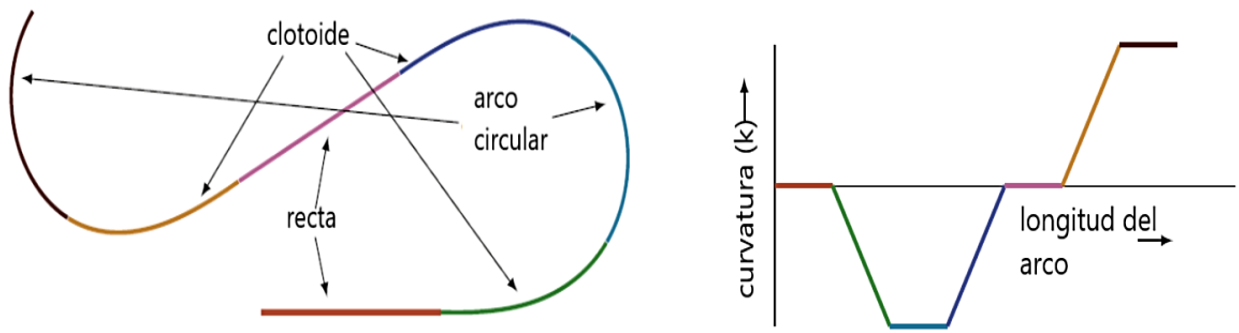


Figura 3.5: Comparación de la curvatura de una clotoide y un arco circular.

**Configuraciones simétricas:** Sean dos configuraciones  $q_1 = (p_1, \theta_1) = (x_1, y_1, \theta_1)$  y  $q_2 = (p_2, \theta_2) = (x_2, y_2, \theta_2)$ , con  $\beta = \text{atan2}(y_2 - y_1, x_2 - x_1)$  que denota la dirección de la recta que une a  $p_1, p_2$  y  $\theta_1, \theta_2$  es la dirección de cada configuración. El par  $q_1, q_2$  será simétrico si se cumple lo siguiente:

$$\frac{\theta_1 + \theta_2}{2} = \text{atan2}(y_2 - y_1, x_2 - x_1) \quad (3.7)$$

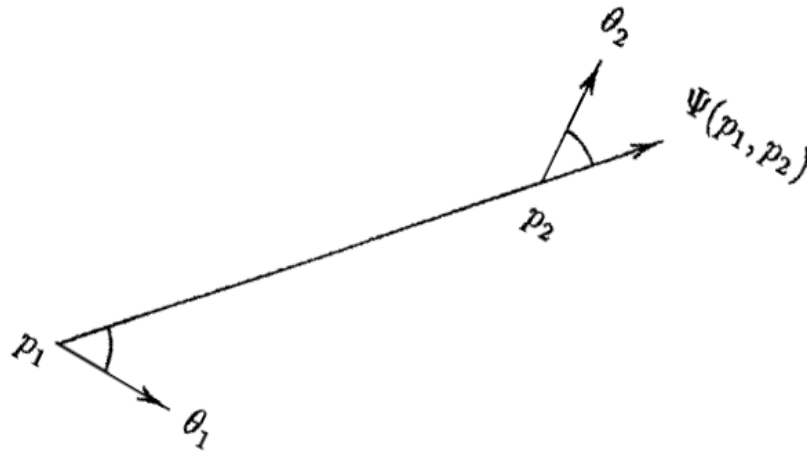


Figura 3.6: Configuraciones simétricas.

**Curva o camino elemental:** Un camino elemental es una curva clotoide que une dos configuraciones simétricas. Estos se caracterizan por tener una variación lineal de la curvatura simétrica como se muestra en la ecuación 3.8, sus parámetros se muestran en las ecuaciones 3.11 y 3.12. En la Figura 3.6 se muestran dos configuraciones simétricas.

$$k(s) = \begin{cases} \min(\sigma s, k_{max}), & \forall s \in [0, \frac{l}{2}] \\ \min(\sigma(l - s), k_{max}), & \forall s \in [\frac{l}{2}, l] \end{cases} \quad (3.8)$$

$$\beta = \text{atan}\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \quad (3.9)$$

$$\alpha = \frac{\theta_2 - \theta_1}{2} = \beta - \theta_1 = \theta_2 - \beta \quad (3.10)$$

$$\sigma = 4\pi \text{sign}(\alpha) \frac{D(|\alpha|)}{r^2} \quad (3.11)$$

$$l = 2 \sqrt{\frac{2\alpha}{\sigma}} \quad (3.12)$$

$$D(\alpha) = \cos\alpha Fc\left(\sqrt{\frac{2\alpha}{\pi}}\right) + \sin\alpha Fs\left(\sqrt{\frac{2\alpha}{\pi}}\right) \quad (3.13)$$

$$Fc(u) = \int_0^u \cos\left(\frac{\pi}{2}u^2\right) du \quad (3.14)$$

$$Fs(u) = \int_0^u \sin\left(\frac{\pi}{2}u^2\right) du \quad (3.15)$$

Donde  $\sigma$  y  $l$  son la razón de cambio de la curvatura y la longitud del arco respectivamente,  $r$  es la distancia euclidiana entre las configuraciones (Figura 3.6),  $Fc$  y  $Fs$  son integrales de Fresnel, las cuales no tienen solución analítica y por tanto requieren de métodos numéricos. Para resolver la integral se utilizará la aproximación racional empleada en [9, 35, 40] que garantiza una reducción del tiempo de cómputo y la posibilidad de ejecución en tiempo real, con un error máximo de  $1.7 \times 10^{-3}$ .

$$Fc(t) \approx \frac{1}{2} - R(t) \sin\left(\frac{1}{2}\pi(A(t) - t^2)\right) \quad (3.16)$$

$$Fs(t) \approx \frac{1}{2} - R(t) \cos\left(\frac{1}{2}\pi(A(t) - t^2)\right) \quad (3.17)$$

Con,

$$R(t) = \frac{0.506t^2 + 1}{1.79t^2 + 2.054t + \sqrt{2}} \quad (3.18)$$

$$A(t) = \frac{1}{0.803t^3 + 1.886t^2 + 2.524t + 2} \quad (3.19)$$

**Caminos bi-elementales:** Un camino bi-elemental permite unir dos configuraciones cualesquiera utilizando un par de caminos elementales. Estos caminos deben unirse en un punto  $q$ , cuyo lugar geométrico es la circunferencia que pasa por ambos puntos de centro  $p_c = \left( \frac{x_1 + x_2 - D(y_2 - y_1)}{2}, \frac{y_1 + y_2 - D(x_2 - x_1)}{2} \right)$ , donde  $D = \cot\left(\frac{\theta_2 - \theta_1}{2}\right)$  y el radio es la distancia euclidiana entre  $q$  y  $p_c$  [41].

En un entorno descompuesto en celdas cuadradas se pueden ajustar cuatro curvas básicas  $C_1, C_2, C_3$  y  $C_4$ , que permiten unir tres configuraciones consecutivas  $(P_1, P_v, P_2)$  y que describen cualquier movimiento posible en un entorno de este tipo, formadas por caminos elementales, bi-elementales y rectas. En la Figura 3.7 se muestran las cuatro curvas básicas ajustadas.

La utilización de estos caminos elementales garantiza una trayectoria suave y continua para el robot, pero aún es necesario asegurar una curvatura permitida para un robot con estas dimensiones. Un robot móvil con la configuración utilizada y una distancia entre las ruedas delanteras y traseras igual a  $dr$  y variación máxima del ángulo de dirección  $\varphi_{max}$  podrá desplazarse de una configuración  $P_1 = (x_1, y_1, \theta_1)$  a  $P_2 = (x_2, y_2, \theta_2)$  mediante un camino elemental si se cumple que el parámetro  $\sigma$  de la curva satisface  $\sigma \leq \sigma_{max}^{rest}$  con,  $\sigma_{max}^{rest} = \frac{1}{2\alpha dr^2} \tan^2 \varphi_{max}$  y  $2\alpha = \theta_2 - \theta_1$  [40], siendo  $\sigma_{max}^{rest}$  la máxima variación de curvatura.

En un entorno descompuesto en celdas la separación entre las celdas es la resolución del entorno  $Re$ . Para determinar la resolución mínima que debe tener el entorno para que todos los caminos sean realizables por el robot se debe cumplir que:

$$Re_{min} = d_{min}^{ejex} = d_{min}^{ejey} = |\max(f_{ci})(p_{fx} - p_{ix})|, C_i = C_1, C_2, C_3, C_4 \quad (3.20)$$

$$f_{ci} = \begin{cases} \sqrt{\frac{\sigma_{ci}}{\sigma_{max}^{rest}}}, & C_i = C_1, C_3 \\ \max\left(\sqrt{\frac{\sigma_{1,ci}}{\sigma_{max1}^{rest}}}, \sqrt{\frac{\sigma_{2,ci}}{\sigma_{max2}^{rest}}}\right), & C_i = C_2 \end{cases} \quad (3.21)$$

$$f_c = \max(f_{ci}) \quad (3.22)$$

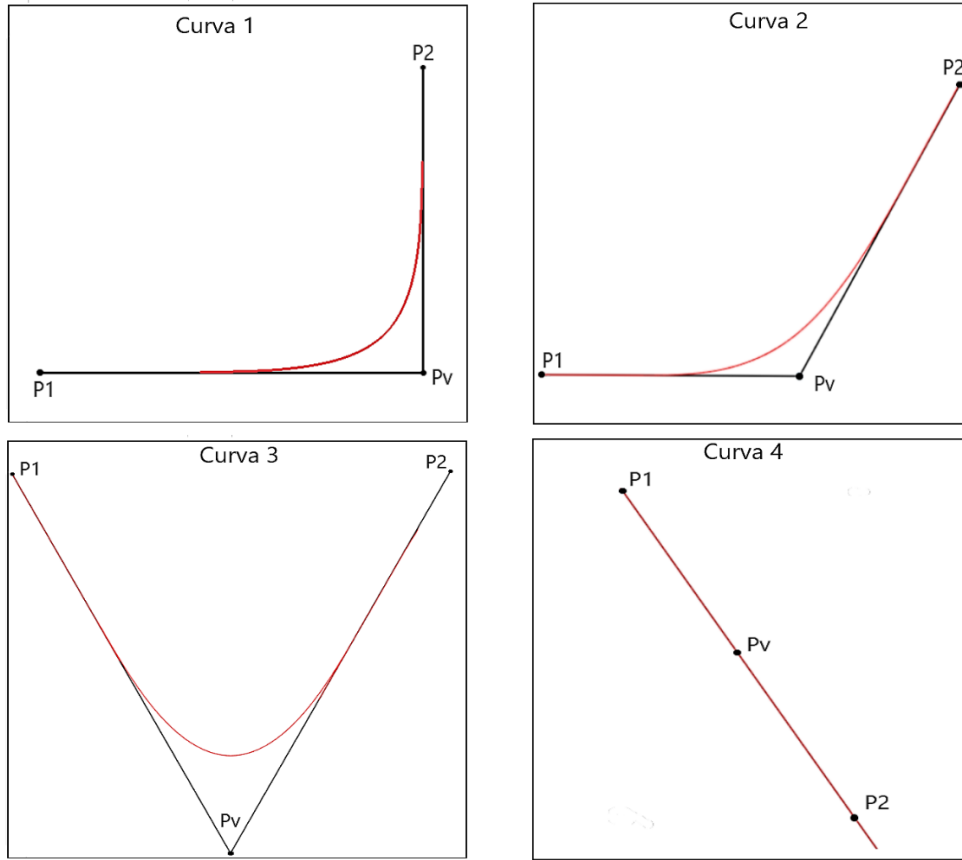


Figura 3.7: Caminos básicos en una descomposición en celdas cuadradas.

Donde  $f_{ci}$  es el factor de escala del camino básico para hacer que las curvas sean realizables y  $d_{min}^{ejex}$ ,  $d_{min}^{ejey}$  las distancias mínimas sobre los ejes a las que puede estar el punto  $p_f$  de  $p_i$ .

En la sección 3.3 se definen las dimensiones del terreno y se toma  $Re = 66 \text{ cm}$ . Esto es porque se escogió  $Re = 2f_c$ , para asegurar que se pudieran realizar todas las curvas y teniendo en cuenta el factor de expansión.

Se utiliza como función de costo para elección del punto de unión:

$$Costo_{qi} = \min(f_{1i} + f_{2i}) \quad (3.23)$$

El problema queda reducido a calcular los parámetros de las curvas básicas, en función del ángulo de dirección máximo de las ruedas, y luego ajustar estas curvas cada tres puntos consecutivos de la secuencia de poses planificada. Si cambia el ángulo de dirección máximo hay que recalcular los parámetros, pero para un mismo robot esto no debe pasar.

### 3.4.2. Algoritmo de generación

La trayectoria de referencia se construye ajustando las curvas básicas entre los puntos planificados. Se define una trayectoria auxiliar que estará formada por los puntos de la trayectoria planificada más el punto medio entre cada par de puntos planeados, queda definida de la siguiente forma:

$$t_{aux} = \{ta_1, \dots, ta_j, \dots, ta_{2n-1}\} \quad (3.24)$$

$$t_{auxj} = \begin{cases} P_i, & i = \frac{j+1}{2}, \quad \forall j \text{ impar} \\ P_i + \frac{(P_{i-1} - P_i)}{2}, i = \frac{j}{2}, \forall j \text{ par} \end{cases} \quad (3.25)$$

Donde  $n$  es la cantidad de puntos del camino planificado,  $P_i$  son los puntos del camino. Luego se ajustará una curva básica entre cada tres puntos consecutivos de la trayectoria auxiliar, en dependencia de la orientación inicial y final de los puntos. Siempre entre puntos impares se ajusta la curva C4 (una recta) y entre puntos pares se debe ajustar una de las restantes curvas (C1, C2, C3) en dependencia de la variación de orientación entre los puntos final e inicial mediante la ecuación 3.26.

$$\theta = \text{atan}\left(\frac{P2_y - P1_y}{P2_x - P1_x}\right) \quad (3.26)$$

Las curvas son creadas resolviendo las ecuaciones 3.3-3.6 a partir del cálculo de los parámetros de las clotoides mediante las ecuaciones 3.8-3.15. En la Figura 3.8 se muestra el diagrama de flujo del algoritmo de generación, en la Figura 3.9 se muestra un acercamiento de las curvas y los puntos auxiliares que se forman. La Tabla 3.1 muestra los parámetros plantilla de las curvas, obtenidos para un ángulo de giro máximo de 45 grados. Se puede ver como la única curva cuyos segundos parámetros no son cero es la C2, esto es porque está formada por un camino bi-elemental.

Tabla 3.1: Parámetros plantilla de las curvas básicas

Curva	$\sigma_1$	$\sigma_2$	$\sigma_{1max}^{rest}$	$\sigma_{2max}^{rest}$	$l1$	$l2$	$\Delta\theta$
<b>Curva C1</b>	2.230	0	$7.07 \times 10^{-4}$	0	1.678	0	$\frac{\pi}{2}$
<b>Curva C2</b>	0.846	-0.052	0.0014	-0.489	1.93	0.418	$\frac{\pi}{4}$
<b>Curva C3</b>	1.115	0	$7.073 \times 10^{-4}$	0	2.374	0	$\frac{\pi}{2}$
<b>Curva C4</b>	0	0	0	0	2	0	0

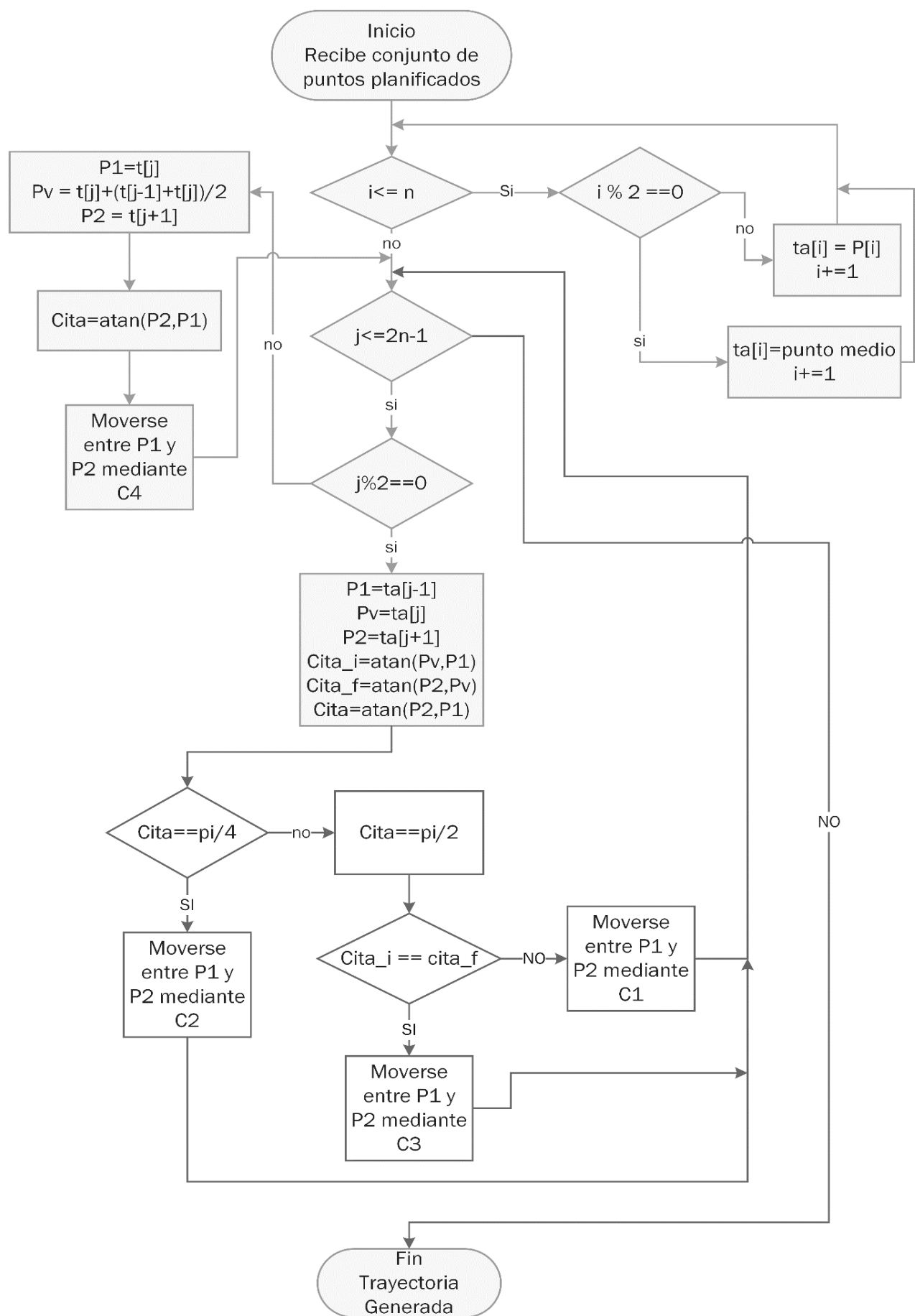


Figura 3.8: Diagrama de flujo del algoritmo de generación.

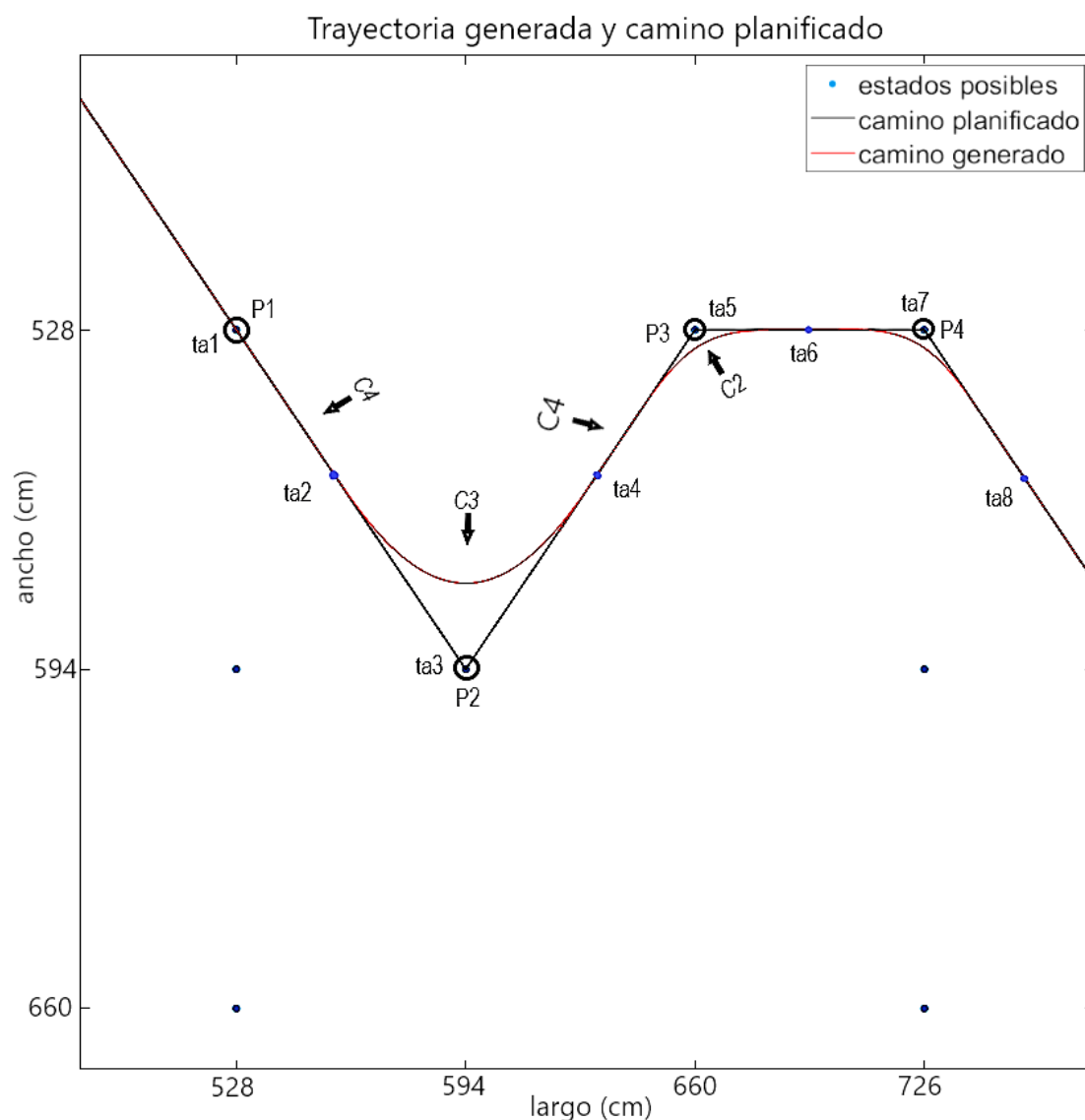


Figura 3.9: Trayectoria planificada y generada.

La distancia que recorre el *rover* es de 1345.4 cm. Se considera que el robot se mueve a una velocidad de 3 cm/s, por lo que demora en realizar el recorrido aproximadamente 448.5 s. No se tiene en cuenta la orientación inicial y final del robot ya que el mismo cuenta con la capacidad de reorientarse sin desplazarse, por lo que no hay que generar giros de orientación. En la Figura 3.10 se puede comprobar la duración de la trayectoria.

En la Figura 3.10 se muestra la variación en las orientaciones de la trayectoria de referencia a lo largo del camino. La Figura 3.11 es un acercamiento para mostrar con más detalles la suavidad en los cambios de orientación, se realiza en la parte que corresponde a las curvas C4, C3, C2. En la Figura 3.12 se muestran la trayectoria generada y la planificada en el mapa, a simple vista solo resalta la curva C3.



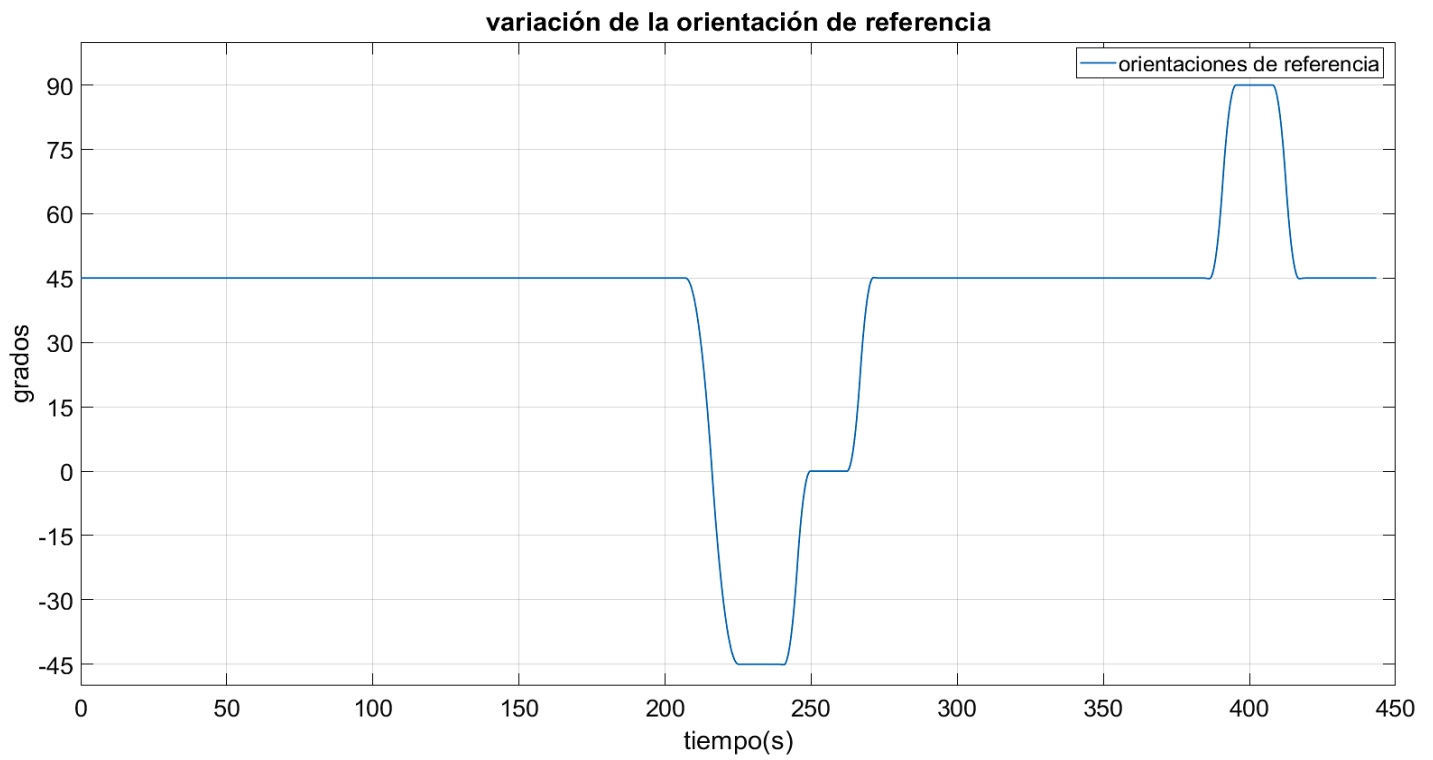


Figura 3.10: Variación de orientaciones de referencia de la trayectoria generada.

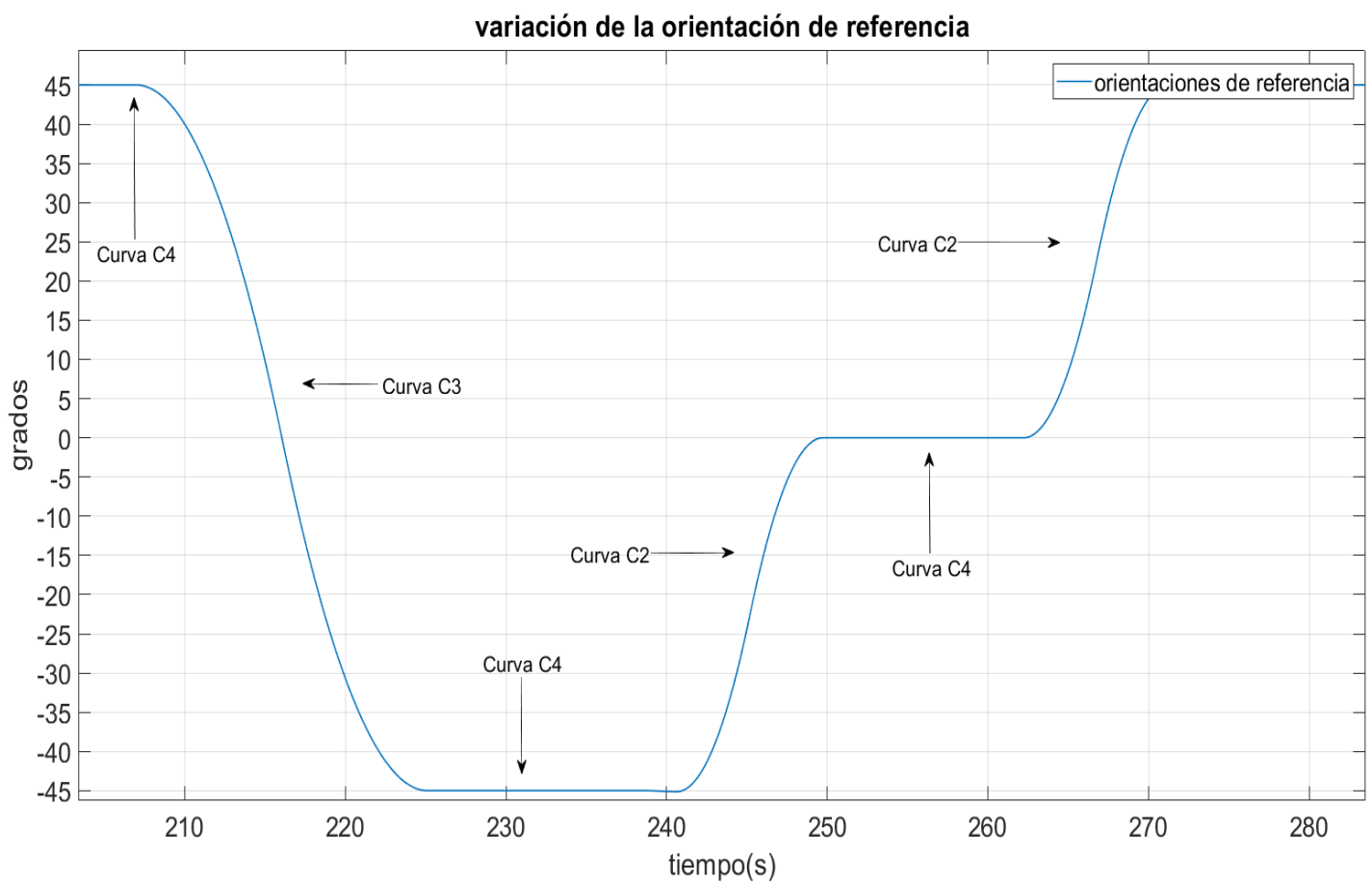


Figura 3.11: Segmento de variaciones con las curvas.

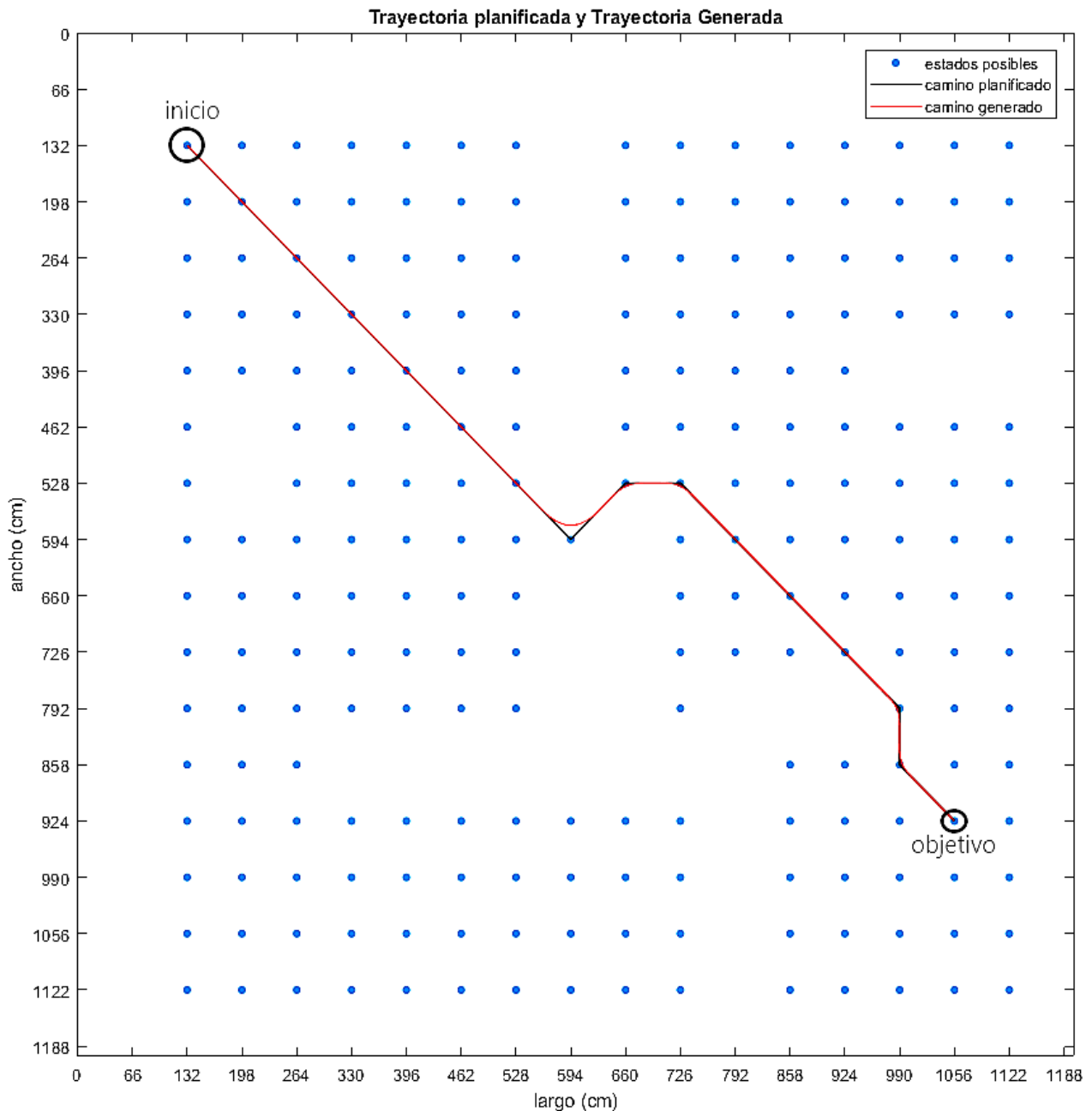


Figura 3.12: Trayectoria generada.

### 3.5. Seguimiento de trayectoria

El *rover* tiene las ruedas fijas al cuerpo, por lo que su movimiento será mediante tracción diferencial. Para el seguimiento de trayectorias se utilizó el algoritmo de control VFO para un robot con configuración diferencial y unicycle propuesto en [46, 47]. Este método permite realizar el control de seguimiento de trayectoria, para un robot con esta configuración, proporcionando un vector de control que garantiza la convergencia del error. El controlador queda definido por las siguientes ecuaciones:

$$\begin{bmatrix} u_2 \\ u_1 \end{bmatrix} = \begin{bmatrix} h_2 \cos \varphi + h_3 \sin \varphi \\ k_\varphi e_a + \varphi_a \end{bmatrix} \quad (3.27)$$

$$\varphi_a = \text{atan2}(\text{sign}(u_{ref})h_3, \text{sign}(u_{ref})h_2) \quad (3.28)$$

$$\varphi = \text{atan2}(\text{sign}(u_{ref})\dot{y}_{ref}, \text{sign}(u_{ref})\dot{x}_{ref}) \quad (3.29)$$

$$h_2 = k_p e_x + \dot{x}_{ref} \quad (3.30)$$

$$h_3 = k_p e_y + \dot{y}_{ref} \quad (3.31)$$

$$e_a = \varphi_a - \varphi \quad (3.32)$$

$$e_x = x_{ref} - x \quad (3.33)$$

$$e_y = y_{ref} - y \quad (3.34)$$

Donde  $k_p$  y  $k_\varphi$  son parámetros de diseño,  $v_{2ref}$  es la velocidad nominal del robot. De aquí se obtienen las velocidades con respecto al cuerpo del robot, luego es necesario obtener las velocidades de las ruedas mediante las ecuaciones 1.6 y 1.7. Este algoritmo se puede utilizar también en un robot con configuración doble de Ackerman de seis ruedas si se realizan las aproximaciones señaladas en la sección 1.8.4 y se mantiene una velocidad de unidades de centímetros.

Se coloca al robot en una posición de inicio diferente de la planificada para verificar el efecto del controlador. La posición inicial planificada es en (132;132) y la posición real de la que parte el robot es en (162;92). En la Figura 3.13 se muestra como el controlador se encarga de corregir la posición y mantenerse cercano a la curva de referencia, aproximando el error a cero mayormente. La velocidad de referencia  $v_{ref}$  se toma como 3 cm/s hacia adelante y -3 cm/s hacia atrás.

El ajuste de los parámetros se realiza mediante prueba y error. Luego de varias simulaciones se pudo observar que valores pequeños de  $k_p$  producen oscilaciones al comienzo de la trayectoria y que a medida que disminuye el valor de  $k_\varphi$  se producen oscilaciones cuando se realizan cambios de orientación, ocasionando mayores desviaciones en la trayectoria. Finalmente quedaron ajustados los parámetros a  $k_p = 10$  y  $k_\varphi = 8$ .

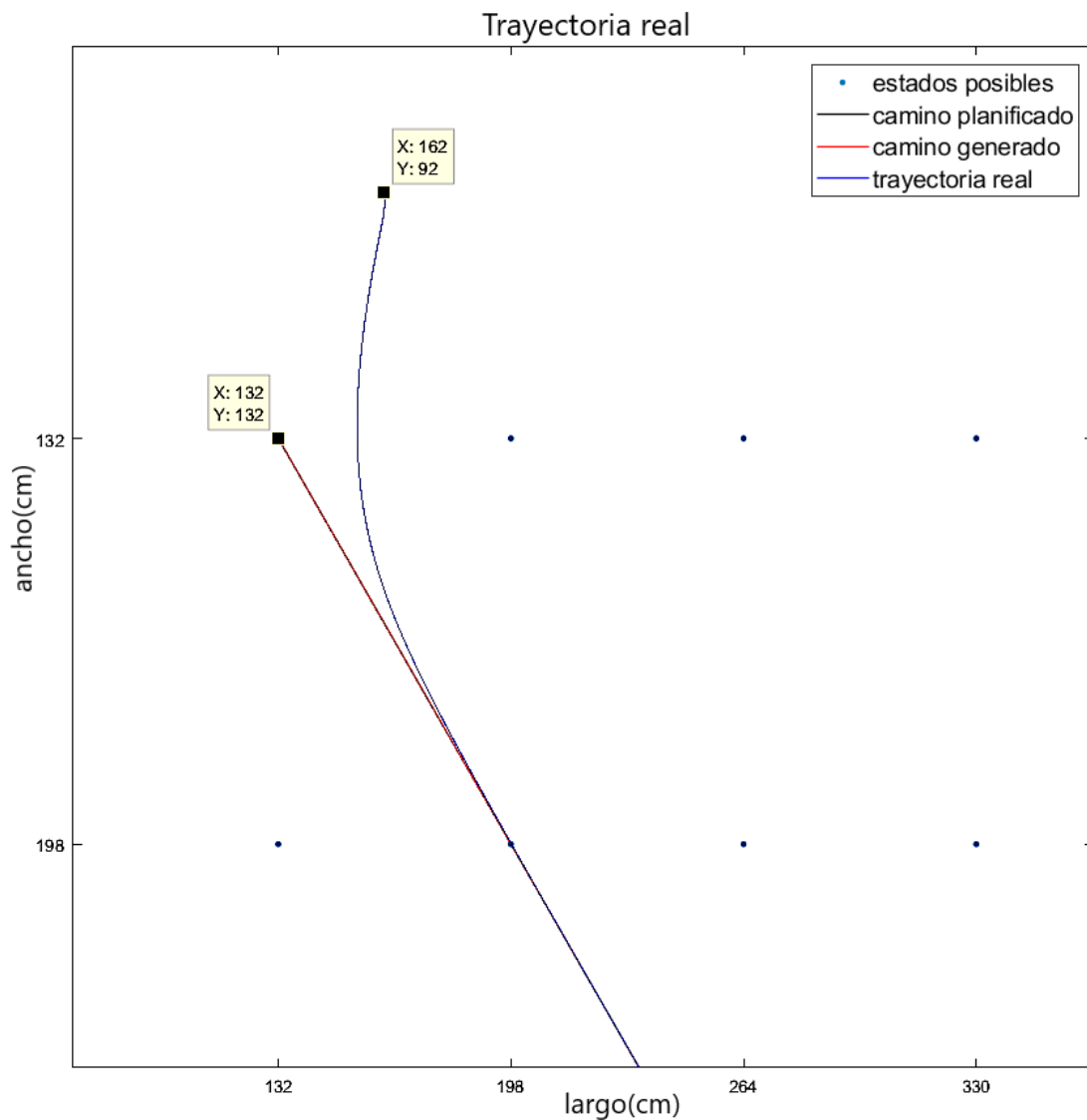


Figura 3.13: Segmento de inicio de la trayectoria real y la de referencia.

En las Figuras 3.14 y 3.15 se muestra el comportamiento del error en el tiempo en los ejes de coordenadas. Se observa que el error se mantiene muy cercano a cero a lo largo de la trayectoria excepto al principio, cuando la trayectoria de referencia no corresponde con la real. El error en el eje x comienza en un valor negativo porque la posición real en x es mayor que la de referencia, es decir, el robot está más adelante. En el caso del eje y, el error empieza siendo positivo porque la posición real en y es menor que la de referencia.

El robot demora aproximadamente 16 s en hacer cero el error, es decir, en alcanzar la trayectoria de referencia. En la Figura 3.16 se muestra la diferencia entre las trayectorias de referencia y la real en una de las curvas realizadas. En esta figura se pueden ver los tres caminos, el planificado, el generado y el real.

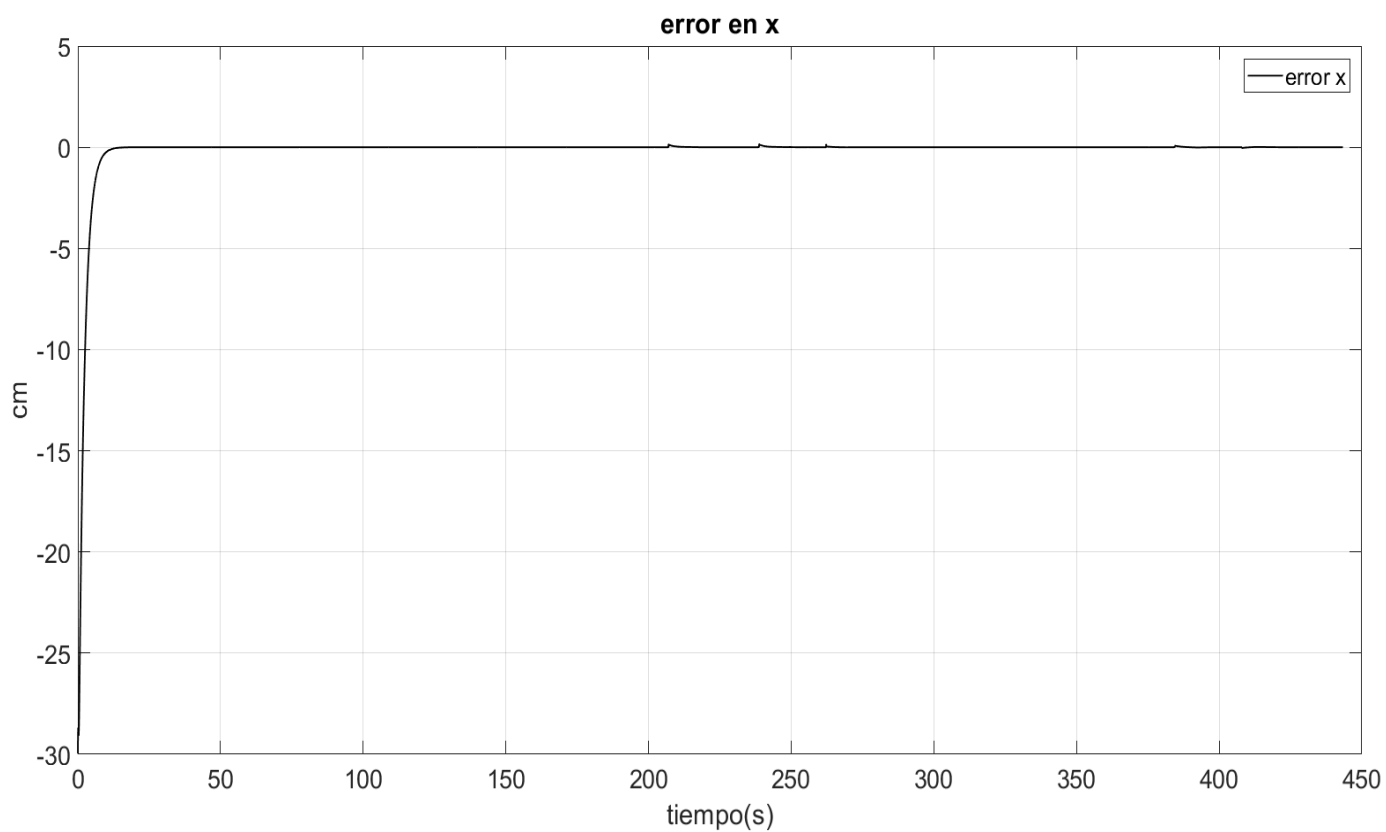


Figura 3.14: Comportamiento del error del seguimiento de trayectoria en el eje x.

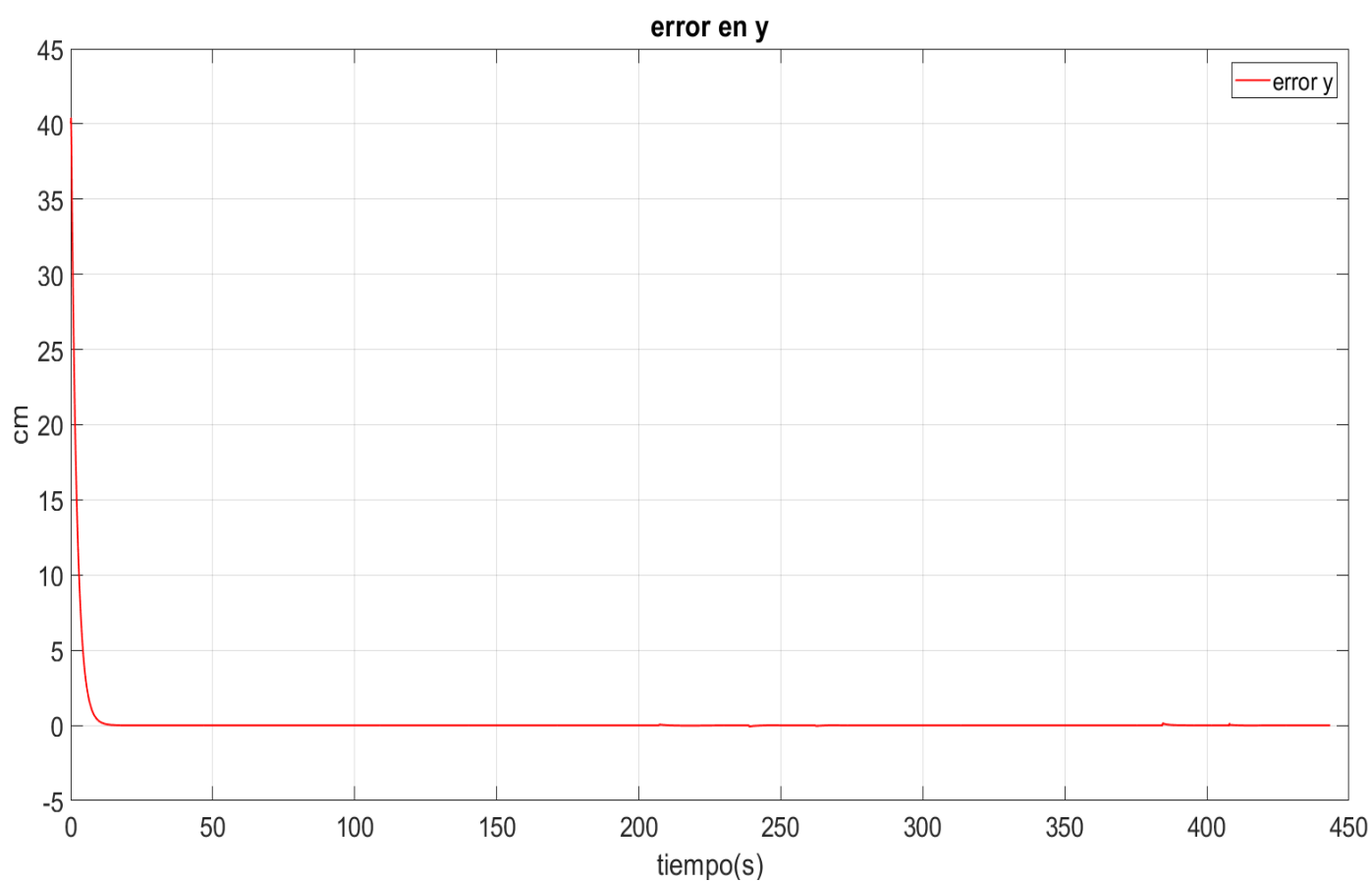


Figura 3.15: Comportamiento del error del seguimiento de trayectoria en el eje y.

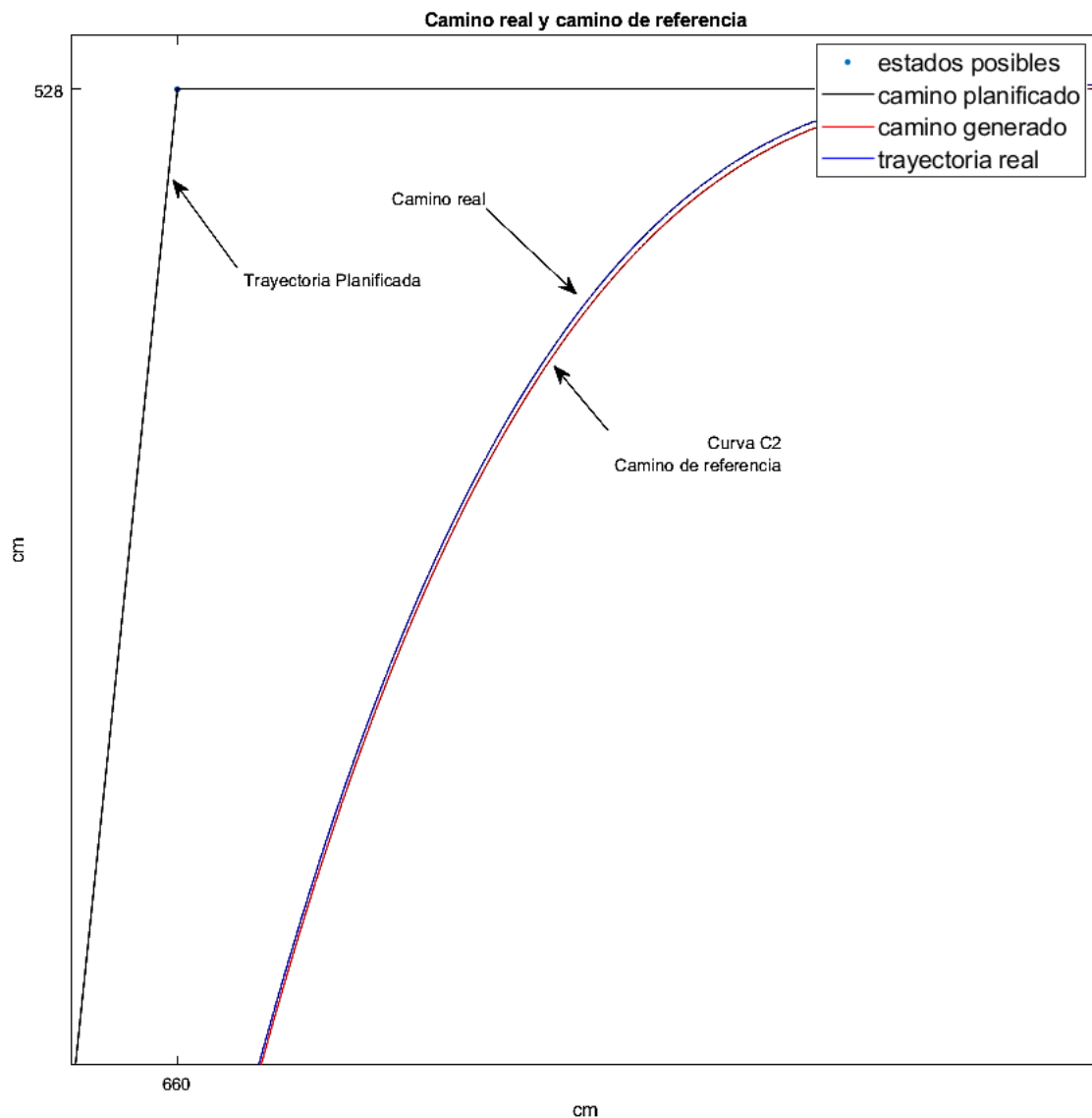


Figura 3.16: Acercamiento del camino en la curva C2.

En la Figura 3.17 se muestra como el robot realiza el seguimiento de los ángulos de orientación. Al principio de cada curva hay una pequeña oscilación debido al cambio de orientación, pero rápidamente se realiza la corrección. La Figura 3.18 muestra un acercamiento del gráfico anterior en el instante en que la posición y el ángulo de orientación no coincide con los de la trayectoria de referencia. Se observan las variaciones de orientación que realiza el controlador para alcanzar la referencia.

En la Figura 3.19 se muestra la trayectoria real, la planificada y la de referencia en el mapa. A simple vista solo se observa la diferencia cuando las posiciones real y generada iniciales de la trayectoria no coinciden. De esta forma se demuestra que este controlador permite el seguimiento de la trayectoria de referencia para un robot

móvil con configuración diferencial, con las dimensiones descritas, creando un nuevo camino que será un subóptimo muy cercano del camino óptimo planificado.

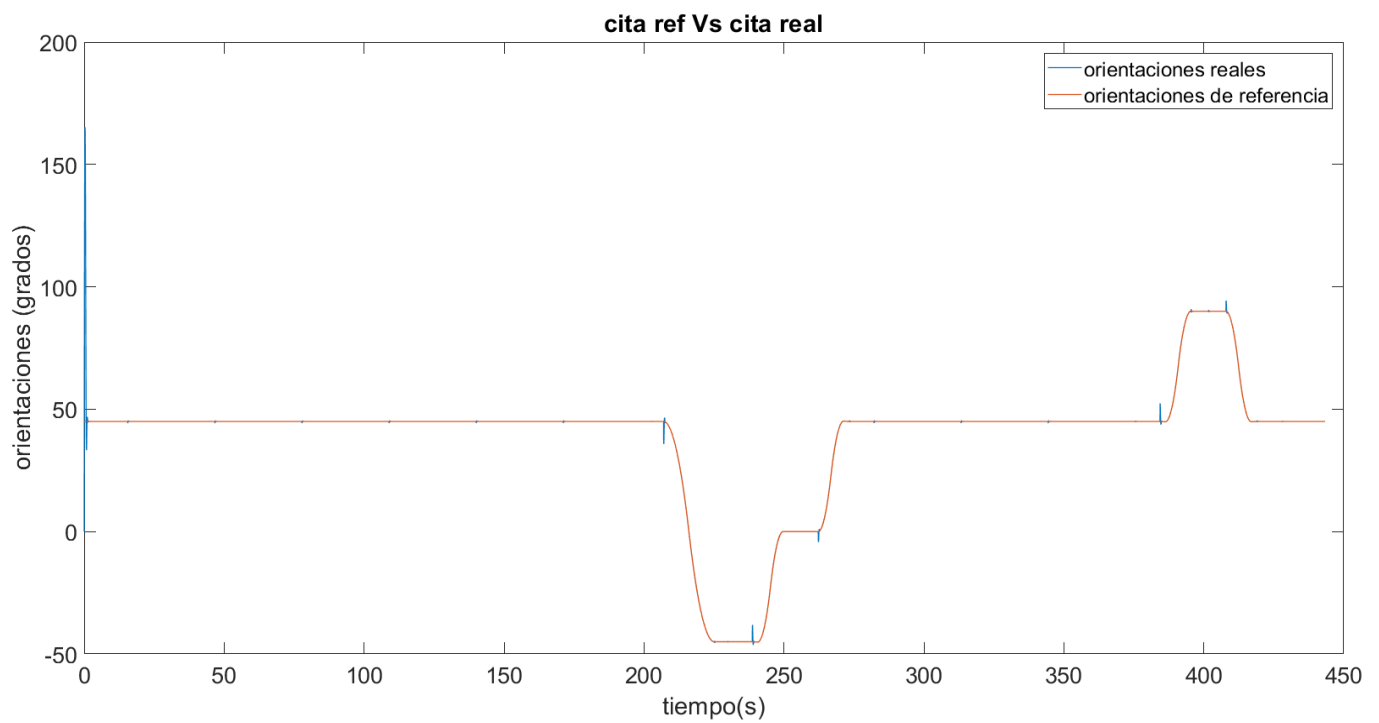


Figura 3.17: Seguimiento de la variación de la orientación de referencia.

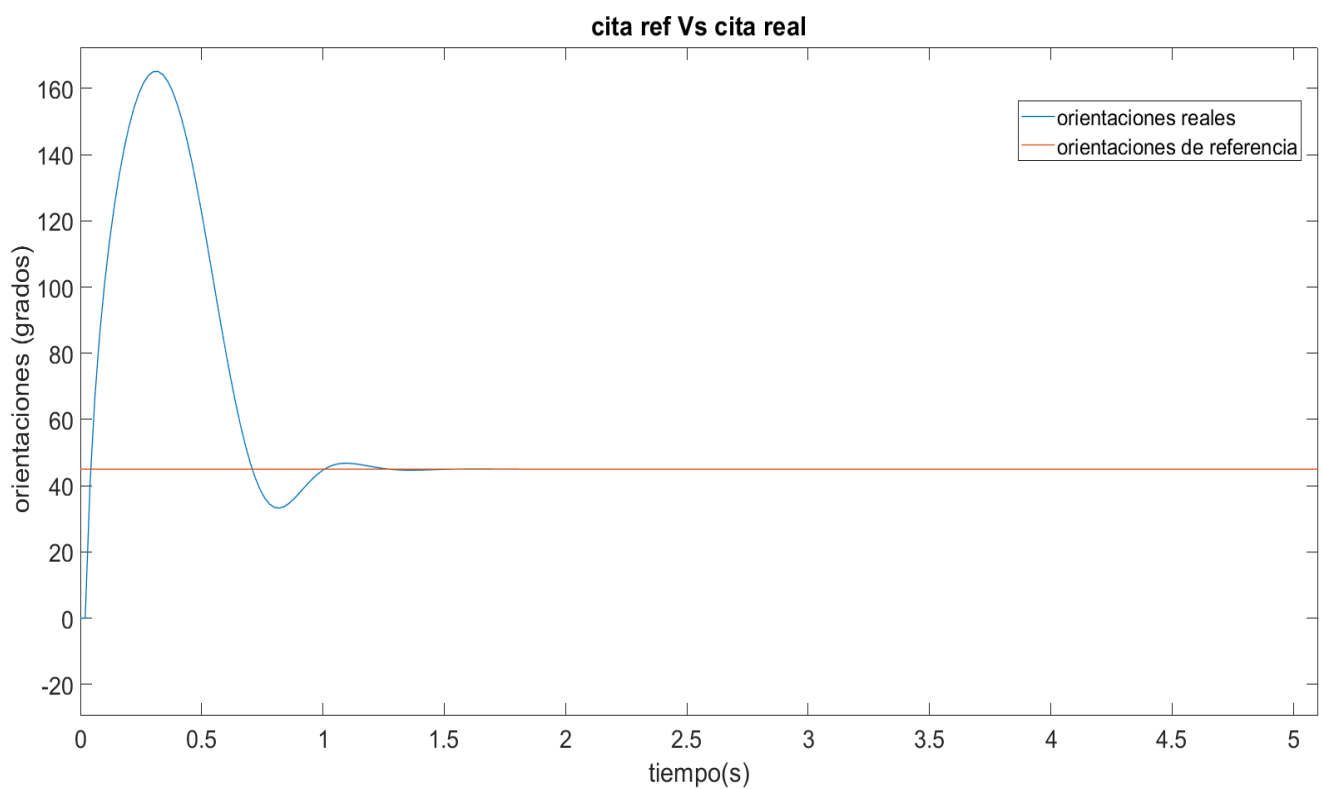


Figura 3.18: Acercamiento del seguimiento de la orientación de referencia.

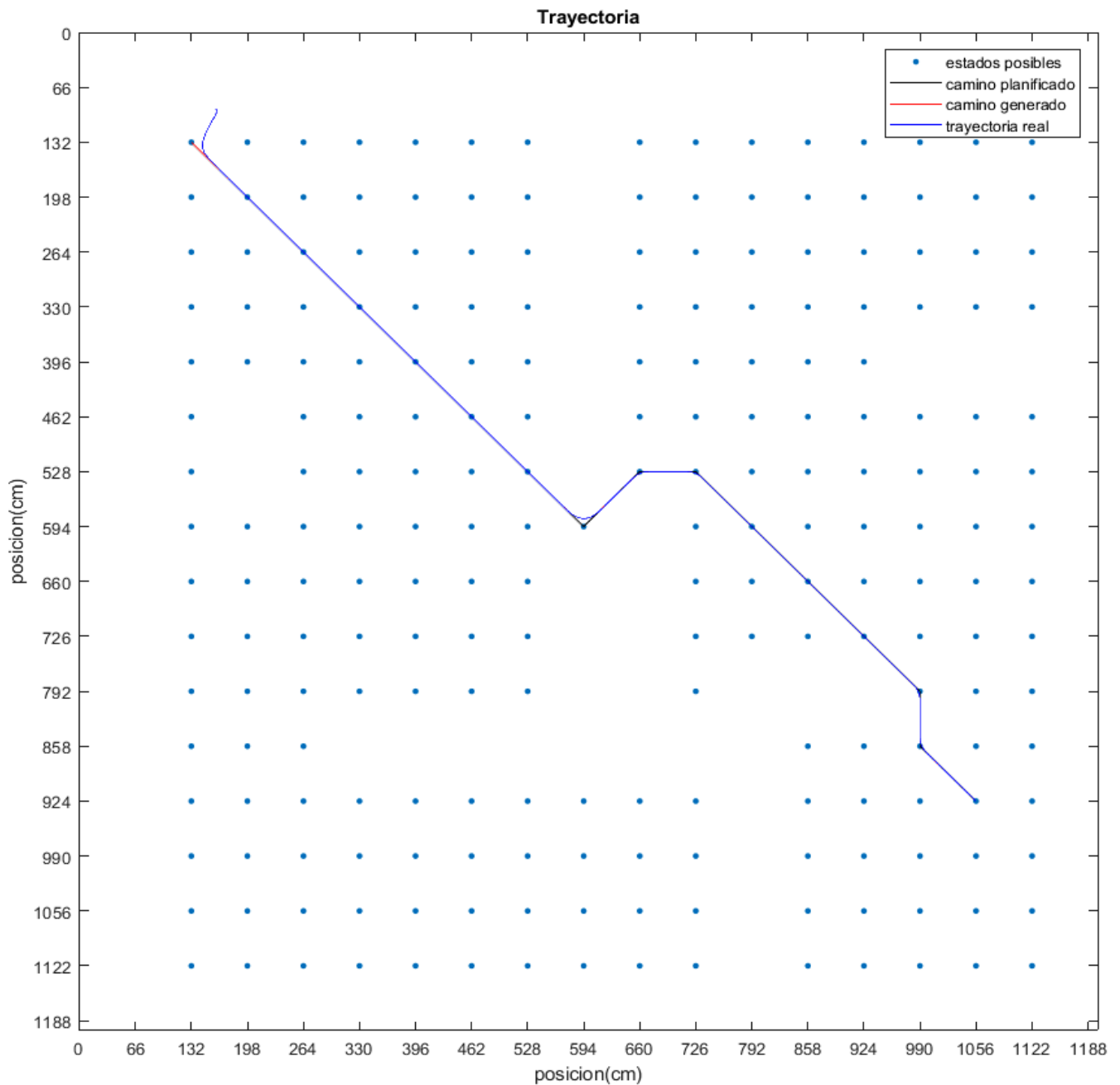


Figura 3.19: Trayectoria real a partir de la de referencia.

## 3.6. Conclusiones

La implementación del algoritmo de planificación permitió que el robot planificara una trayectoria libre de obstáculos al considerar el factor de expansión del entorno y la representación del entorno en el C-espacio, obteniéndose el camino de menor distancia. Al incluir la etapa de generación de curvas clotoideas se provee al robot de una trayectoria continua y suave, optimizando el recorrido del robot por el camino planificado y evitando que tenga que detenerse para girar sobre su eje o alejarse de



la trayectoria para luego reincorporarse, lo cual aumenta la distancia recorrida y puede provocar errores. Se demostró que el algoritmo de VFO es aplicable al robot descrito y que puede realizar el seguimiento de la trayectoria de referencia obtenida.

Los algoritmos garantizaron el seguimiento de la trayectoria permitiendo que el robot tenga cierto grado de autonomía y pueda llevar a cabo el cumplimiento de la misión con menor intervención del ser humano. Se pudo comprobar que estos algoritmos son aplicables para un robot de estas dimensiones a la velocidad de referencia propuesta.

## Capítulo 4. Análisis técnico-económico

### 4.1. Introducción

Se realizará el análisis cualitativo y cuantitativo del costo total del proyecto y los beneficios que se obtendrían con su aplicación. El cálculo económico de la investigación se realiza en cuanto a costo, precio y efecto económico.

### 4.2. Costo de la elaboración del proyecto

El Costo Total (CT) consiste en la suma del Costo Directo (CD) y el Costo Indirecto (CI), se calcula mediante:

$$CT = CD + CI \quad (4.1)$$

El CI se refiere a los gastos realizados durante la investigación, dígame consumo eléctrico, gastos de administración, etc. Se calcula multiplicando un coeficiente de costo por el salario básico, el coeficiente considerado es de 0.85. Queda de forma siguiente:

$$CI = S_{básico} * 0.85 \quad (4.2)$$

El salario básico total se considera como el que se le paga a un estudiante de 5to año durante el tiempo de realización del proyecto, sin incluir las vacaciones ni la seguridad social. Queda definido por la ecuación 4.3.

$$SB_{total} = \sum_{i=1}^n (a_i + b_i) \quad (4.3)$$

Donde,  $a_i$  es la cantidad de días de investigación de cada participante,  $b_i$  el diario de cada participante y  $n$  la cantidad de participantes.

El CD se calcula mediante la suma del  $SB_{total}$ , el salario complementario total ( $SC_{total}$ ), el seguro social total ( $SS_{total}$ ), los medios o materiales directos ( $MD$ ), pasajes y otros gastos ( $DP$ ) que no se incluyen en este caso.

$$CD = SB_{total} + SC_{total} + SS_{total} + MD + OG + DP \quad (4.4)$$

El  $SC_{total}$  se calcula mediante la suma de los salarios complementarios individuales, el salario complementario individual es el 9.09% del salario básico, que se destina para el pago de las vacaciones, está definido por normativas del estado. Quedan definidos por las ecuaciones 4.5 y 4.6.

$$SC_i = 9.09\% * SB_n \quad (4.5)$$

$$SC_{Total} = \sum_{i=0}^n SC_1 + SC_2 + SC_3 + \dots + SC_n \quad (4.6)$$

La  $SS_{total}$  se calcula mediante se calcula mediante la suma de las seguridades sociales individuales. La Seguridad Social Individual es el 10% del salario básico, se calculan como se muestra en las ecuaciones 4.7 y 4.8.

$$SS_i = 10\% * (SB_n + SC_n) \quad (4.7)$$

$$SS_{Total} = \sum_{i=0}^n SS_1 + SS_2 + SS_3 + \dots + SS_n \quad (4.8)$$

Los gastos por medios o materiales directos se refieren a los materiales utilizados directamente en la investigación, o sea, equipos y componentes que se incorporan en el resultado. Los otros gastos (OG) incluyen el precio de utilización de los equipos, se considera el gasto por concepto de tiempo de máquina, el cual es de 10 pesos cubanos por hora de trabajo. Se muestra en la ecuación 4.9.

$$OG = t_{máquina} * costo \quad (4.9)$$

#### 4.2.1. Personas involucradas en realización del proyecto

Se consideran 24 días laborales para los tutores y 24 días de actividad docente, así como 8 horas de jornada laboral, según las leyes cubanas.

##### Dr. Valery Moreno Vega

Su salario básico es de 950 pesos cubanos mensuales, considerando 24 días al mes y durante 5 meses y en cada día laborable se dedicó como promedio 1 hora al trabajo de diploma. Por cada hora laboral el ingreso del Dr. Valery Moreno Vega fue de 4.94 pesos cubanos.

$$SB_{Dr.Moreno} = 4.94 MN * 120 horas = 592.8 MN \quad (4.10)$$

$$SC_{Dr.Moreno} = 0.0909 * SB_{Dr.Moreno} = 53.81 MN \quad (4.11)$$

$$SS_{Dr.Moreno} = 0.05(SB_{Dr.Moreno} + SC_{Dr.Moreno}) = 32.3 MN \quad (4.12)$$

##### Dra. Ivón O. Benítez González

Su salario básico es de 790 pesos cubanos mensuales, considerando 24 días al mes y durante 5 meses y en cada día laborable se dedicó como promedio 1 hora al trabajo de diploma. Por cada hora laboral el ingreso de la Dra. Ivón O. Benítez González fue de 4.11 pesos cubanos.

$$SB_{Dra.Benítez} = 4.11 \text{ MN} * 120 \text{ horas} = 493.2 \text{ MN} \quad (4.13)$$

$$SC_{Dra.Benítez} = 0.0909 * SB_{Dra.Benítez} = 44.83 \text{ MN} \quad (4.14)$$

$$SS_{Dra.Benítez} = 0.05(SB_{Dra.Benítez} + SC_{Dra.Benítez}) = 26.9 \text{ MN} \quad (4.15)$$

### **Estudiante Carlos Raúl Morales Hernández**

La compensación de un estudiante de 5to de año ingeniería es de 100.00 pesos cubanos mensuales, tomando 24 días laborales durante 5 meses de trabajo, dedicando 8 horas de trabajo diario como promedio a la investigación, ganado por cada hora el equivalente a 0.521 pesos.

$$SB_{carlos} = 0.521 \text{ pesos cubanos} * 960 \text{ horas} = 500.16 \text{ MN} \quad (4.16)$$

$$SC_{carlos} = 0.0909 * SB_{carlos} = 45.46 \text{ MN} \quad (4.17)$$

$$SS_{carlos} = 0.05(SB_{carlos} + SC_{carlos}) = 27.28 \text{ MN} \quad (4.18)$$

El salario básico total, el salario complementario total y la seguridad social total se obtienen sustituyendo los salarios de los participantes obtenidos por las ecuaciones 4.10-4.18 en la ecuación 4.3, 4.6 y 4.8.

$$SB_{total} = 493.2 + 500.16 + 592.8 \text{ MN} = 1586.16 \text{ MN} \quad (4.19)$$

$$SC_{total} = 32.3 + 44.83 + 27.28 = 104.41 \text{ MN} \quad (4.20)$$

$$SS_{total} = 54.56 + 53.8 + 64.61 = 197.97 \text{ MN} \quad (4.21)$$

### **4.2.2. Otros gastos**

Se considera un total de 960 horas de tiempo de máquina, sustituyendo en la ecuación 4.9 se obtiene que:

$$OG = 960 \text{ h} * 10.0 \text{ MN/h} = 9600 \text{ MN} \quad (4.22)$$

### **4.2.3. Gastos por medios o materiales directos**

Se incluyen todos los gastos realizados para la adquisición de los componentes y materiales utilizados directamente. En la Tabla 4.1 se muestra el costo de los componentes para el diseño propuesto del rover.

Tabla 4.1: Gastos por materiales directos

Componente	Proveedor	Precio (USD)	Cantidad	Total (USD)
Plataforma Bogie-Runt Rover	Servo City	70	1	70
Raspberry Pi 3B	Adafruit	30	1	30
Raspberry Pi Sense-Hat	Amazon	35	1	35
Controlador TB6612	Adafruit	5	3	15
Encoders	Robot Shop	9x2 unidades	3	27
Panel Solar	Amazon	15	1	15
Baterías	Amazon	21	1	21
<b>TOTAL</b>				<b>213</b>

#### 4.2.4. Costo indirecto, directo y total

Se utiliza la ecuación 4.2 para el cálculo del costo indirecto, la ecuación 4.4 para el costo directo y la ecuación 4.1 para el costo total.

$$CI = SB_{total} * 0.85 = 856.53$$

$$\begin{aligned}
 CD &= SB_{total} + SC_{total} + SS_{total} + MD + OG + DP \\
 &= 1586.16 \text{ MN} + 104.1 \text{ MN} + 197.7 \text{ MN} + 213 \text{ USD} + 9600 \text{ MN} \\
 &= 11487.9 \text{ MN} + 213 \text{ USD}
 \end{aligned}$$

$$CT = CD + CI = 11487.9 \text{ MN} + 213 \text{ USD} + 856.53 = 12344.4 \text{ MN} + 213 \text{ USD}$$

### 4.3. Precio

El precio se refiere a lo que se paga por contrato o lo que pagaría la entidad que recibe el servicio o resultado, se calcula como el costo total más su 10%. De esta forma, se permite hasta un 10% del costo total en que se incurre como ganancia de la investigación.

$$\begin{aligned}
 P &= CT + (0.1CT) = 12344.4 \text{ MN} + 213 \text{ USD} + 0.1(12344.4 \text{ MN} + 213 \text{ USD}) \quad (4.23) \\
 &= 12344.4 \text{ MN} + (1234.44 \text{ MN} + 21.3 \text{ USD}) \\
 &= 13578.84 \text{ MN} + 234.3 \text{ USD}
 \end{aligned}$$

#### **4.4. Análisis cuantitativo y cualitativo del efecto económico como resultado de la implementación del trabajo de diploma**

Con la realización del *rover* se garantiza una plataforma robótica con la cual se pueden realizar nuevas investigaciones en este campo. Se podrán realizar el estudio de movimiento sobre terrenos accidentados y el control de seguimiento de trayectorias con análisis de tracción, además, la utilización del Raspberry Pi garantiza la expansión de las funcionalidades del robot y el Arduino la adición de otros sensores, como sensores de corriente y tensión para implementar un algoritmo de MPPT en el panel. Para recuperar esta inversión se pueden generar proyectos de investigación que tributen a resultados en un futuro cercano.

# Conclusiones

Durante el desarrollo del proyecto se realizó una búsqueda de la información más reciente sobre robótica móvil terrestre y espacial. Se investigaron las formas de movimiento, estructuras mecánicas, los sensores que se utilizan, las unidades de procesamiento. Además, se estudiaron los algoritmos de navegación utilizados y las formas de generación y almacenamiento de energía utilizadas en los *rovers* y otros vehículos eléctricos. Esto permitió obtener mayor conocimiento sobre la robótica móvil actual y seleccionar adecuadamente los componentes a integrar en la plataforma robótica.

La propuesta de diseño realizada garantiza un robot con algunas características de un *rover*, tales como la configuración del sistema de suspensión, un sistema de generación y almacenamiento de energía que permite operar sin necesidad de la intervención del hombre, la capacidad de medir ángulos de orientación, realizar la planificación de trayectorias y el control de seguimiento de estas. Se realizó el diseño basado en una Raspberry Pi 3B, la cual permite la expansión de las funcionalidades y además garantiza un gran poder de procesamiento y facilidad de programación en varios lenguajes. Los controladores utilizados son muy fáciles de reemplazar, por lo que cambiar el diseño reutilizando la plataforma es relativamente sencillo.

Se logró planificar un camino óptimo en cuanto distancia al utilizar el algoritmo de programación dinámica de Bellman Ford como planificador. Al considerar el factor de expansión del entorno se logró que el camino planificado fuera libre de obstáculos, quedando un camino seguro de transitar.

Al incluir una etapa de generación de trayectorias, se realizó un estudio de las curvas clotoideas para generar caminos suaves. El algoritmo de generación proporcionó un camino continuo en orientación y posición, de esta forma el robot no tendrá que detenerse para girar sobre su eje o alejarse de la trayectoria para luego reincorporarse. La trayectoria de referencia obtenida fue muy cercana a la trayectoria planificada, por lo tanto, esta constituye un subóptimo muy próximo del camino óptimo planificado.

Se implementó un algoritmo de control de seguimiento de trayectoria para un robot con configuración diferencial, obteniéndose un error en la trayectoria muy cercano a cero. Se demostró que el algoritmo se encarga de corregir la posición inicial del robot en caso de no coincidir con la esperada, lo que permitió concluir que el algoritmo es

aplicable a un robot con las características del seleccionado y capaz de seguir la trayectoria generada.

Luego de lo expresado anteriormente, es posible afirmar que los objetivos propuestos fueron cumplidos satisfactoriamente.



# Recomendaciones

Entre las recomendaciones fundamentales para futuros trabajos con la plataforma se encuentran:

- Adicionar una cámara compatible con la Raspberry Pi para dotar de visión al *rover*.
- Incluir algoritmos para la construcción de mapas del terreno.
- Implementar algoritmos para realizar análisis de cinemática inversa y análisis de terreno.
- Realizar la unión de la medición de los sensores de la IMU y los odométricos para implementar un algoritmo de localización preciso.
- Construir un soporte que permita la incorporación de un panel solar más grande para aumentar el número de horas de operación.
- Incluir un sistema mecánico que permita desplegar y recoger los paneles solares.
- Incluir un regulador con MPPT para aumentar la eficiencia del panel solar.
- Incluir un sensor de sol para realizar la localización absoluta.
- Incorporar un reloj de tiempo real para realizar tareas programadas en el tiempo.
- Incorporar sensores para la medición de corriente y tensión de la batería.

# Bibliografía

1. Siciliano B, Khatib O. Springer handbook of robotics: Springer; 2016.
2. Kumar Saha S. Introducción a la robótica 2010.
3. Murphy R. Introduction to AI robotics: MIT press; 2000.
4. Gao Y. Contemporary Planetary Robotics An Approach Toward Autonomous Systems: John Wiley & Sons; 2016.
5. Viera Lopez G. Diseño e implementación de una plataforma robótica móvil con aplicaciones basadas en procesamiento digital de imágenes. Habana: CUJAE; 2015.
6. Delgado Morales JS. Implementación de un Robot Móvil con Ruedas de Accionamiento Diferencial. La Habana: CUJAE; 2013.
7. Torres Piñeiro MO. Planificación y Generación de trayectorias para robot móvil recolector de objetos. Ciudad de La Habana, Cuba: Instituto Superior Politécnico José Antonio Echeverría; 2010.
8. Torres Piñeiro MO, Moreno Vega Valery. APLICACIÓN DE LA PROGRAMACIÓN DINÁMICA EN LA PLANIFICACIÓN DE TRAYECTORIAS PARA LA ROBÓTICA MÓVIL. Revista Electrónica, Automática y Comunicaciones. 2010;Vol. XXXI, No.2.
9. Torres Piñeiro MO, Moreno Vega Valery, Mellado Martin. Generación de trayectorias para robots móviles mediante curvas clotoideas. Revista Electrónica, Automática y Comunicaciones. 2011;Vol. XXXVI, No.2.
10. Siegwart R, Nourbakhsh IR, Scaramuzza D. Introduction to autonomous mobile robots: MIT press; 2011.
11. Ortigoza RS, Sánchez JG, Sotelo VB, Vilchis MM, Hernández V. Guzmán, G. Silva Ortigoza Una panorámica de los robots móviles Télématique, vol. 6, núm. 3, 2007, pp. 1-14, Universidad Rafael Belloso Chacín. 2007.
12. Silva Ortigoza R, García Sánchez J, Barrientos Sotelo V, Molina Vilchis MA, Hernández Guzmán VM, Silva Ortigoza G. Una panorámica de los robots móviles. Télématique. 2007;6(3).
13. Ellery A. Planetary rovers: robotic exploration of the solar system: Springer; 2015.
14. Iagnemma K, Dubowsky S. Mobile Robots in Rough Terrain. Mobile Robots in Rough Terrain: Springer; 2004. p. 1-15.
15. Chinchkar G, Panchal R. Design of Rocker Bogie Mechanism. International Advanced Research Journal in Science, Engineering and Technology. 2017;4(1).

16. Scherz P. Practical electronics for inventors: McGraw-Hill, Inc.; 2006.
17. Ellery A. Planetary Rovers: Robotic Exploration of the Solar System. The Aeronautical Journal. 2016;121(1236):277-8.
18. Bräunl T. Embedded robotics: mobile robot design and applications with embedded systems: Springer Science & Business Media; 2008.
19. Williams A. Microcontroller projects using the Basic Stamp: CMP Books Gilroy, CA; 2002.
20. Monk S. Raspberry Pi cookbook: Software and hardware problems and solutions: " O'Reilly Media, Inc."; 2016.
21. Molloy D. Exploring Raspberry Pi: interfacing to the real world with embedded Linux: John Wiley & Sons; 2016.
22. Upton E, Halfacree G. Raspberry Pi user guide: John Wiley & Sons; 2014.
23. Velazco Estrada J. Diseño e implementación de una plataforma robótica móvil de configuración diferencial para el grupo de robótica del Departamento de Automática y Computación.: INSTITUTO SUPERIOR POLITÉCNICO JOSÉ ANTONIO ECHEVERRÍA; 2012.
24. Pei JDZ. An electrical power system of Mars rover. Transportation Electrification Asia-Pacific (ITEC Asia-Pacific), 2014 IEEE Conference and Expo. 2014.
25. Alta\_Devices. ¿Por qué utilizar células solares de arseniuro de galio? 2017. Available from: <https://www.altadevices.com/use-gallium-arsenide-solar-cells/>.
26. Stachniss C. Robotic mapping and exploration: Springer; 2009.
27. Hacot H. Analysis and traction control of a rocker-bogie planetary rover: Massachusetts Institute of Technology; 1998.
28. Hacot H, Dubowsky S, Bidaud P. Analysis and simulation of a rocker-bogie exploration rover. COURSES AND LECTURES-INTERNATIONAL CENTRE FOR MECHANICAL SCIENCES. 1998:95-102.
29. Barraquand J, Latombe J-C. Robot motion planning: A distributed representation approach. The International Journal of Robotics Research. 1991;10(6):628-49.
30. Kwok KS, Driessen BJ, editors. Path planning for complex terrain navigation via dynamic programming. American Control Conference, 1999 Proceedings of the 1999; 1999: IEEE.
31. Cai C, Ferrari S. Information-driven sensor path planning by approximate cell decomposition. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). 2009;39(3):672-89.

32. Moreno Vega V, Piñeiro MOT. Diseño y simulación de un robot móvil recolector de objetos. Revista Ingenieria Electronica, Automatica y Comunicaciones. 2010;30(3):16-20.
33. Sleumer N, Tschichold-Gürmann N. Exact cell decomposition of arrangements used for path planning in robotics. Technical report/ETH Zürich, Department of Computer Science. 1999;329.
34. Lingelbach F, editor Path planning using probabilistic cell decomposition. Robotics and Automation, 2004 Proceedings ICRA'04 2004 IEEE International Conference on; 2004: IEEE.
35. Torres MO, editor Planificación y generación de trayectorias para un robotmóvil recolector de objetos. I Jornadas científicas UAH-CES de Cuba [Recurso electrónico]: Universidad de Alcalá-Universidades Cubanas, 2010: workshop de la acción AECID 08-CAP-0655: resultados de cooperación universitaria en red regional; 2010: Servicio de Publicaciones.
36. Torres Piñeiro MO, Moreno Vega V. Un estudio de dos métodos de planificación de trayectorias en entornos estáticos. Revista Cubana de Ciencias Informáticas. 2009;3(1-2).
37. Maikel O. Torres Piñeiro VMV. Modificación del método de Bellman-Ford para la planificación de trayectorias en entornos estáticos. 2011.
38. LUNDBERG M. Path planning for autonomous vehicles using clothoid based smoothing of A\* generated paths and optimal control. Stockholm, Sweden: KTH ROYAL INSTITUTE OF TECHNOLOGY SCHOOL OF ENGINEERING SCIENCES; 2017.
39. Filip J, Azkarate M, Visentin G. TRAJECTORY CONTROL FOR AUTONOMOUS PLANETARY ROVERS. Automation and Robotics Section, European Space Agency, ESA, Noordwijk, The Netherlands. 2017.
40. Torres Piñeiro MO, Moreno Vega V. Un nuevo método de generación de caminos para robots móviles mediante curvas clotoides. Ingeniería Electrónica, Automática y Comunicaciones. 2015;36(2):98-118.
41. Yutaka Kanayama BIH. Smooth Local Path Planning for Autonomous Vehicles. 1991.
42. Kanayama YJ, Hartman BI. Smooth local-path planning for autonomous vehicles1. The International Journal of Robotics Research. 1997;16(3):263-84.

43. Sanchez CM, Sanchez JRG, Cervantes CYS, Ortigoza RS, Guzman VMH, Juarez JNA, et al. Trajectory generation for wheeled mobile robots via Bézier polynomials. *IEEE Latin America Transactions*. 2016;14(11):4482-90.
44. Gómez-Bravo F, Cuesta F, Ollero A, Viguria A. Continuous curvature path generation based on  $\beta$ -spline curves for parking manoeuvres. *Robotics and autonomous systems*. 2008;56(4):360-72.
45. Brady M, Hollerbach JM, Johnson TL, Lozano-Pérez T, Mason MT, Winston PH. *Robot motion: Planning and control*: MIT press; 1982.
46. Michalek M, Kozłowski K. Vector-Field-Orientation feedback control method for a differentially driven vehicle. *IEEE Transactions on Control Systems Technology*. 2010;18(1):45-65.
47. Michałek M, Kozłowski K. Motion planning and feedback control for a unicycle in a way point following task: The VFO approach. *International Journal of Applied Mathematics and Computer Science*. 2009;19(4):533-45.
48. Michalek M, Kozłowski K, editors. Trajectory tracking for a threecycle mobile robot: The vector field orientation approach. *Decision and Control, 2005 and 2005 European Control Conference CDC-ECC'05 44th IEEE Conference on*; 2005: IEEE.
49. Michałek M, Kozłowski K, editors. Unified approach to trajectory tracking and set-point control for a front-axle driven car-like mobile robot. *American Control Conference (ACC)*, 2011; 2011: IEEE.
50. Michałek M, Kozłowski K. Feedback control framework for car-like robots using the unicycle controllers. *Robotica*. 2012;30(4):517-35.
51. U. Cortés AC, A. Benítez, A. Díaz. Control de Movimiento de un Robot Móvil Tipo Diferencial Robot ȳot-32b. *Congreso Nacional de Control Automático, AMCA*. 2015.
52. Matijevic J. Mars pathfinder microrover-implementing a low cost planetary mission experiment. 1996.
53. Rieder R, Gellert R, Brückner J, Klingelhöfer G, Dreibus G, Yen A, et al. The new Athena alpha particle X-ray spectrometer for the Mars Exploration Rovers. *Journal of Geophysical Research: Planets*. 2003;108(E12).
54. Greicius T. Curiosity Rover: National Aeronautics and Space Administration (NASA); 2017 [updated 3 agosto de 2017]. Available from: [https://www.nasa.gov/mission\\_pages/msl/overview/index.html](https://www.nasa.gov/mission_pages/msl/overview/index.html).

55. Kucinskis F, Ferreira M, editors. Taking the ECSS autonomy concepts one step further. SpaceOps 2010 Conference Delivering on the Dream Hosted by NASA Marshall Space Flight Center and Organized by AIAA; 2010.
56. City S. 2018. Available from: [www.servocity.com/right-angle-gearmotor](http://www.servocity.com/right-angle-gearmotor).
57. Ada L. Adafruit TB6612 1.2A DC/Stepper Motor Driver Breakout Board 2016. Available from: [www.adafruit.com](http://www.adafruit.com).
58. Arduino. Available from: [www.arduino.cc/en/guide/arduinomega2560](http://www.arduino.cc/en/guide/arduinomega2560).
59. Shop R. polulu encoder. Available from: [www.robotshop.com/poluluencoder](http://www.robotshop.com/poluluencoder).
60. Foundation R. The Raspberry Pi official site 2018. Available from: [www.raspberrypi.org](http://www.raspberrypi.org).
61. Amazon. Monocrystalline Silicon Solar Panel 2018. Available from: [www.amazon.com/lixada/portable-ultra-thin-monocrystalline-silicon-solar-panel-chrger](http://www.amazon.com/lixada/portable-ultra-thin-monocrystalline-silicon-solar-panel-chrger).
62. Piñeiro MOT, Vega VM. Planificación de trayectorias mediante Programación Dinámica. Revista Ingeniería Electrónica, Automática y Comunicaciones ISSN: 1815-5928. 2011;31(2):31-7.
63. Maikel O. Torres Piñeiro VMV, Martin Mellado. GENERACIÓN DE CAMINOS MEDIANTE CURVAS CLOTOIDES SIMÉTRICAS. 2011.
64. McCrae J, Singh K. Sketching piecewise clothoid curves. Computers & Graphics. 2009;33(4):452-61.