

Escuela Superior Politécnica del Litoral

Facultad de Ingeniería en Mecánica y Ciencias de la Producción

Diseño y simulación de un robot de seis ruedas para navegación autónoma y superación de obstáculos mediante sensores de visión y control predictivo

INGE – 2522

Proyecto Integrador

Previo la obtención del Título de:

Ingeniero en Mecatrónica

Presentado por:

Ángel Josué Espín Lumbano

Dana Paola Soriano Santana

Guayaquil - Ecuador

Año: 2024

Dedicatoria

Con profunda gratitud dedico este trabajo a las personas que han sido pilares fundamentales en mi vida. A mi madre, cuyas enseñanzas y apoyo inquebrantable me han guiado siempre. A mi hermana y a mi sobrina, quienes con su amor y energía me inspiran a dar lo mejor de mí cada día. También quiero reconocer a mis amigos, cuyo respaldo y compañía han sido esenciales para superar cada desafío en este camino.

Ángel Espín

Dedicatoria

Este proyecto va dedicado especialmente a mis padres, por su amor incondicional y por ser mi guía en los momentos más difíciles. Gracias por su apoyo constante, por creer en mí y por enseñarme el valor del esfuerzo y la dedicación.

A mis hermanas, que me dieron las fuerzas de seguir adelante y me impulsaron a dar siempre lo mejor de mí.

A mis amigos que gané a lo largo de la carrera. Gracias por dejarme compartir con ustedes y que este desafío llamado carrera fuera muy llevadero.

Este proyecto es un reflejo de la influencia y apoyo de cada uno de ustedes en mi vida.

Dana Soriano Santana

Agradecimientos

Quiero expresar mi sincero agradecimiento a mi madre, Carol Lumbano, por su apoyo incondicional, y a mi hermana Julissa y a mi sobrina, Alana Villon, quienes me han motivado en cada paso. Agradezco también a mis amigos, en especial a mi compañera de tesis, Dana Soriano, por su compañerismo y apoyo constante a lo largo de la carrera. Mi gratitud va también al M.Sc. Efraín Terán y al Ph.D. Carlos Saldarriaga, quienes fueron tutores fundamentales para la culminación exitosa de este trabajo. Finalmente, extendo un especial agradecimiento al Ph.D. Nabih Pico, cuya orientación y experiencia fueron cruciales en el desarrollo de este proyecto.

Ángel Espín

Agradecimientos

Mi más sincero agradecimiento a todas las personas que contribuyeron a la realización de este proyecto. Primero, al Ph.D. Nabih Pico, por brindarnos su dedicación, conocimiento y tiempo, guiando y enseñando tanto a mi como a mi compañero de tesis, Ángel Espín, en el desarrollo de este proyecto. También al M.Sc. Efraín Terán por ofrecernos su orientación y compartir su experiencia a lo largo de la culminación del mismo. A todos ustedes, mi más profundo agradecimiento por su valiosa contribución a este logro.

Dana Soriano Santana

Declaración Expresa

Nosotros, Espín Lumbano Ángel Josué y Soriano Santana Dana Paola acordamos y reconocemos que:

La titularidad de los derechos patrimoniales de autor (derechos de autor) del proyecto de graduación corresponderá al autor o autores, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor o autores.

La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, software o información no divulgada que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada por nosotros durante el desarrollo del proyecto de graduación, pertenecerán de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio del porcentaje que nos corresponda de los beneficios económicos que la ESPOL reciba por la explotación de mi/nuestra innovación, de ser el caso.

En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique a los autores que existe una innovación potencialmente patentable sobre los resultados del proyecto de graduación, no se realizará publicación o divulgación alguna, sin la autorización expresa y previa de la ESPOL.

Guayaquil, 22 de mayo del 2024



Espín Lumbano Ángel

Josue



Soriano Santana Dana

Paola

Evaluadores

Carlos Saldarriaga Mero, Ph.D.

Profesor de Materia

Efraín Terán Calle, M.Sc.

Tutor de proyecto

Resumen

El proyecto consistió en el diseño y simulación de un robot de seis ruedas capaz de navegar de forma autónoma y superar obstáculos, utilizando sensores de visión y control predictivo. El objetivo fue crear un robot que pudiera detectar y modelar escaleras y rampas en tiempo real, y ajustar su velocidad para superar estos obstáculos con estabilidad. Para ello, se emplearon técnicas de procesamiento de imágenes con una cámara de profundidad RGB-D, se diseñó un mecanismo de tracción y se implementó un Control Predictivo por Modelo (MPC). Los resultados mostraron que el robot propuesto puede superar escaleras de hasta 11 cm de altura y pendientes de hasta 15° sin volcamiento, con una precisión del 97% en la subida y del 94% en la bajada. Finalmente, a través de las distintas validaciones se concluye que el robot es capaz de navegar de manera autónoma, integrando adecuadamente los sensores y actuadores.

Palabras Clave: Navegación autónoma, detección de obstáculos, robot de seis ruedas, control predictivo por modelo (MPC), cámara de profundidad RGB-D.

Abstract

The project involved the design and simulation of a six-wheeled robot capable of autonomous navigation and obstacle overcoming using vision sensors and predictive control. The objective was to create a robot capable of detecting and modeling stairs and ramps in real-time, adjusting its speed to overcome these obstacles with stability. Image processing techniques with an RGB-D camera were employed, a traction mechanism was designed, and Model Predictive Control (MPC) was implemented. The results showed that the robot could overcome stairs up to 11 cm high and slopes up to 15° without tipping over, with an accuracy of 97% when ascending and 94% when descending. It is concluded that the robot is capable of autonomous navigation, adequately integrating sensors and actuators.

Keywords: Autonomous navigation, obstacle detection, six-wheeled robot, model predict control (MPC), RGB-D depth camera.

Índice general

Resumen.....	I
Abstract.....	II
Índice general.....	III
Abreviaturas.....	V
Simbología.....	VII
Índice de figuras.....	VIII
Índice de tablas.....	X
Índice de planos.....	XI
Capítulo 1.....	1
1.1 Introducción.....	2
1.2 Descripción del problema.....	3
1.3 Justificación del problema.....	4
1.4 Objetivos.....	5
1.4.1 Objetivo general.....	5
1.4.2 Objetivos específicos.....	5
1.5 Marco teórico.....	6
1.5.1 Detección y mapeo de escaleras en robots cuadrúpedos.....	6
1.5.2 Detección y superación autónoma de escaleras en robots de orugas.....	7
1.5.3 Reconocimiento de la geometría del terreno en tiempo real.....	9
1.5.4 Navegación autónoma y salto de obstáculos en robots cuadrúpedos.....	9
1.5.5 Diseño de robots con ruedas modificables.....	10
1.5.6 Estimación de escaleras con redes neuronales y cámara RGB-D.....	11
1.5.7 Model Predictive Control (MPC) para la navegación del robot.....	12
Capítulo 2.....	14
2.1 Metodología.....	15
2.1.1 Propuesta y selección de la solución.....	15

2.2 Proceso de diseño	22
2.2.1 Requisitos de diseño.....	24
2.2.2 Selección del motor.....	25
2.2.3 Selección de recursos electrónicos.....	30
2.2.4 Medidas de los eslabones	36
2.2.5 Diseño de la estructura principal del robot.....	41
2.2.6 Algoritmo de reconocimiento de obstáculos	46
2.2.7 Model predict control (MPC) para controlar las ruedas.....	53
Capítulo 3.....	57
3.1 Resultados y análisis.....	58
3.1.1 Diseño final del robot.....	58
3.1.2 Análisis de esfuerzos.....	60
3.1.3 Mapeo del lugar mediante gmapping	63
3.1.4 Navegación autónoma	67
3.1.5 Detección y modelado de obstáculo escalera.....	70
3.1.6 Detección y visualización de obstáculo rampa.....	74
3.2 Análisis de costos	79
Capítulo 4.....	82
4.1 Conclusiones y recomendaciones	83
4.1.1 Conclusiones	83
4.1.2 Recomendaciones.....	84
Referencias.....	85
Apéndices.....	88

Abreviaturas

2D	Dos dimensiones
3D	Tres dimensiones
CAD	Diseño Asistido por Computadora
CUDA	Arquitectura Unificada de Dispositivos de Computo
DCL	Diagrama de cuerpo libre
DWA	Enfoque de Ventana Dinámica
ESPOL	Escuela Superior Politécnica del Litoral
GPU	Unidad de Procesamiento Gráfico
IA	Inteligencia Artificial
ICCAS	Conferencia Internacional sobre Control, Automatización y Sistemas
IMU	Unidad de Medición Inercial
LiDAR	Detección y medición de la luz
MPC	Control Predictivo por Modelo
NaN	No es un número
PCA	Análisis de Componentes Principales
PCL	Policaprolactona
PID	Proporcional – Integral – Derivativo
PLA	Ácido Poliláctico
PWM	Modulación por ancho de pulso
RANSAC	Consenso de Muestra Aleatoria
RGB-D	Rojo, Verde, Azul - Profundidad
ROS	Sistema Operativo del Robot

RPM	Revoluciones por minuto
RVIZ	Visualización en ROS
SKKU	Universidad Sungkyunkwan
SLAM	Localización y Mapeo Simultáneos

Simbología

°	Grados
cm	Centímetro
kg	Kilogramo
m	Metro
mm	Milímetro
MPa	Megapascal
N	Newton
Nm	Newton metro
s	Segundo

Índice de figuras

Figura 1.1 Robot móvil de seis ruedas subiendo una escalera.....	2
Figura 1.2 Nube de puntos tridimensional mediante cámara de profundidad	7
Figura 1.3 Robot móvil con orugas.....	8
Figura 1.4 Superación de obstáculos mediante sensores laser.....	9
Figura 1.5 Comparación de seis funciones de recompensa en el escenario de objetos	10
Figura 1.6 Modelo que junta las interacciones del robot en multitudes	11
Figura 1.7 Navegación en base a dificultad de ambiente.....	12
Figura 1.8 Enfoque de optimización simple con un número discreto de maniobras	13
Figura 2.1 Robot con mecanismo de tracción.....	15
Figura 2.2 Robot con mecanismo de elevación	16
Figura 2.3 Robot con mecanismo bogie	17
Figura 2.4 Metodología de diseño	23
Figura 2.5 DCL del robot en una pendiente.....	26
Figura 2.6 Diagrama velocidad vs torque del motor.....	29
Figura 2.7 Servomotor DC DYNAMIXEL XL430-W250-T	30
Figura 2.8 Nvidia Jetson Nano.....	31
Figura 2.9 Intel Realsense D435i.....	32
Figura 2.10 Dynamixel XL430-W250-T	33
Figura 2.11 Controlador U2D2 para servomotor Dynamixel	34
Figura 2.12 Suspensión bogie	35
Figura 2.13 Robot con mecanismo de tracción.....	36
Figura 2.14 Máxima altura que el robot puede subir y bajar un escalón	39
Figura 2.15 Robot con sistema de bogie	42
Figura 2.16 Caja del robot bogie.....	42
Figura 2.17 Eslabones posteriores del mecanismo	43
Figura 2.18 Eslabones anteriores del mecanismo	43
Figura 2.19 Diseño preliminar del bogie	44
Figura 2.20 Modelo del robot en Onshape.....	44
Figura 2.21 Modelo del robot en el ambiente de Gazebo	45
Figura 2.22 Diagrama de flujo de detección de escaleras.....	46
Figura 2.23 Diagrama de flujo de la detección de pendientes	50
Figura 2.24 Model predict control con sus paquetes	53

Figura 2.25 Transformaciones entre frames para usar SLAM generando un mapa	55
Figura 3.1 Robot con sistema de bogie. con componentes	58
Figura 3.2 Mecanismo de la cámara	59
Figura 3.3 Robot con sistema de bogie dentro de Gazebo.....	59
Figura 3.4 Diseño final del robot subiendo una escalera	60
Figura 3.5 Robot con sistema de bogie en el peor escenario de tensión.....	61
Figura 3.6 Análisis de esfuerzos en el chasis para robot con sistema de bogie.....	62
Figura 3.7 Análisis de esfuerzos en ruedas para robot con sistema de bogie	62
Figura 3.8 Sistema local y global de la cinemática del robot	64
Figura 3.9 Algoritmo de cálculo de odometría para el robot tipo bogie.....	65
Figura 3.10 Mapeo del entorno para navegación.....	66
Figura 3.11 Mapa del entorno generado para la navegación	67
Figura 3.12 Sistema de control de la navegación autónoma.....	68
Figura 3.13 Model predict control implementado en Gazebo	69
Figura 3.14 Model predict control con rviz	69
Figura 3.15 Detección y modelado de escaleras en un entorno real	71
Figura 3.16 Algoritmo que calcula las velocidades necesarias para superar el obstáculo.....	72
Figura 3.17 Robot superando el obstáculo detectado por la cámara.....	73
Figura 3.18 Detección de pendientes en un entorno real	76
Figura 3.19 Modelado de pendientes detectadas en Gazebo	77
Figura 3.20 Robot superando las pendientes detectadas por la cámara.....	77

Índice de tablas

Tabla 2.1 Evaluación de los criterios de diseño del robot de seis ruedas	18
Tabla 2.2 Evaluación de criterio, estabilidad.....	19
Tabla 2.3 Evaluación de criterio, adaptabilidad.....	19
Tabla 2.4 Evaluación de criterio, costo.....	20
Tabla 2.5 Evaluación de criterio, durabilidad.....	20
Tabla 2.6 Evaluación de criterio, simplicidad del mecanismo	21
Tabla 2.7 Matriz de decisión.....	22
Tabla 2.8 Requisitos de diseño	25
Tabla 2.9 Pendiente vs Torque.....	28
Tabla 2.10 Parámetros de la cinemática del robot	37
Tabla 3.1 Tabla de resultados detección y modelado de las escaleras.....	74
Tabla 3.2 Tabla de resultados detección de pendientes	78
Tabla 3.3 Costes de los equipos usados	81

Índice de planos

PLANO 1	Vista isométrica del robot
PLANO 2	Lista de piezas
PLANO 3	Perspectivas ortogonales del robot
PLANO 4	Chasis del robot
PLANO 5	Placa de PLA
PLANO 6	Placa de aluminio
PLANO 7	Eslabón del <i>bogie</i>
PLANO 8	Eslabón posterior
PLANO 9	Ensamble de la cámara
PLANO 10	Carcasa de la cámara
PLANO 11	Placa de aluminio de la cámara
PLANO 12	Placa posterior de la cámara

Capítulo 1

1.1 Introducción

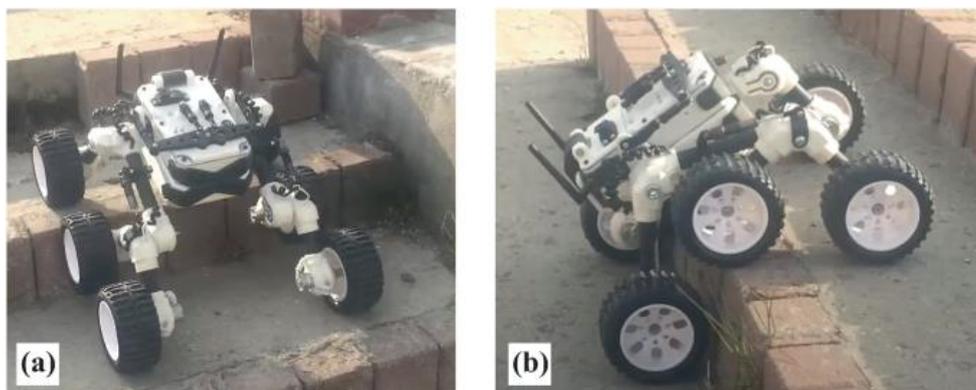
La robótica moderna se encuentra en un punto de inflexión, impulsada por avances tecnológicos que permiten a los robots realizar tareas cada vez más complejas. Entre los desafíos más importantes en este ámbito se encuentran la navegación autónoma y la superación de obstáculos. Estos aspectos son esenciales para el desarrollo de robots móviles que puedan desempeñarse en entornos diversos y ofrecer soluciones prácticas a problemas reales.

La navegación autónoma permite a los robots desplazarse de manera inteligente por su entorno sin necesidad de intervención humana. Esto los convierte en herramientas valiosas para una amplia gama de aplicaciones, desde la exploración de áreas peligrosas o inaccesibles hasta la optimización de procesos logísticos en almacenes y fábricas.

La superación de obstáculos es otro aspecto crucial para la robótica móvil. Los robots que pueden superar obstáculos como escaleras, pendientes y terrenos irregulares tienen un potencial enorme para mejorar la calidad de vida de las personas, especialmente en áreas como la asistencia en el hogar para personas mayores o con discapacidades, o en misiones de rescate en zonas afectadas por desastres naturales. Este proyecto busca contribuir a este campo desarrollando un sistema que ofrezca una solución práctica y económica.

Figura 1.1

Robot móvil de seis ruedas subiendo una escalera



Nota: Imagen obtenida de [1]

1.2 Descripción del problema

El desafío que enfrenta este proyecto se centra en el diseño y simulación de un robot móvil de seis ruedas capaz de navegar de manera autónoma y superar obstáculos como escaleras y pendientes. Esta necesidad surge debido a la creciente demanda de aplicaciones donde los terrenos son impredecibles y presentan varios desafíos significativos. El sistema del robot debe detectar obstáculos a una distancia inferior a 5 metros utilizando una cámara de profundidad RGB-D para determinar las dimensiones de los obstáculos detectados, como su altura, profundidad y ancho, para garantizar una navegación segura.

El control del robot también representa un aspecto importante de este problema. Se requiere desarrollar un controlador dinámico y adaptable que use un algoritmo de control predictivo para ajustar la velocidad de las ruedas del robot en función de los obstáculos identificados. Esto permitirá al robot anticipar y reaccionar adecuadamente a los cambios en el entorno optimizando su trayectoria para superar obstáculos.

Además, es necesario diseñar un mecanismo que mantenga estable la cámara de profundidad mientras el robot se mueve y detecta obstáculos, garantizando una percepción precisa y continua del entorno sin mucho ruido que impida la detección de los obstáculos.

Para garantizar el éxito de este robot, es fundamental que pueda superar obstáculos específicos como escaleras de hasta 11 cm de altura, considerando sus limitaciones físicas. Esto requiere un diseño mecánico robusto que asegure la estabilidad durante la escalada.

Para validar el funcionamiento del robot y su sistema de control predictivo, se realizarán simulaciones detalladas en un entorno virtual como Gazebo. Estas simulaciones permitirán probar y ajustar el diseño en diversas condiciones, asegurando su desempeño en el mundo real.

1.3 Justificación del problema

La resolución del desafío de navegación autónoma y superación de obstáculos en robots móviles de seis ruedas tiene varias implicaciones importantes. Estos robots son muy versátiles y pueden utilizarse en aplicaciones críticas como misiones de rescate en áreas afectadas por desastres naturales y en entornos industriales como almacenes y fábricas. Por otro lado, también es esencial que estos robots puedan operar eficazmente en entornos cotidianos, con obstáculos menores, como escaleras, pendientes a superar. La capacidad de estos robots para navegar de manera autónoma y superar obstáculos en terrenos complejos optimiza los procesos y mejora la eficiencia operativa, pues al superar obstáculos y terrenos irregulares se ahorra energía y tiempo los cuales son esenciales tanto en situaciones industriales como en contextos cotidianos.

En segundo lugar, el avance y la integración de tecnologías avanzadas, como los sensores de visión y el control predictivo, son cruciales para mejorar la capacidad de navegación y superación de obstáculos de los robots móviles. Estas tecnologías permiten al robot percibir su entorno con precisión y ajustar su comportamiento en tiempo real, lo que no solo mejora la seguridad, sino que también impulsa la innovación en el diseño y aplicación de robots móviles.

Además, la mejora en la seguridad es un aspecto crítico que se aborda con la autonomía de estos robots. Al poder superar obstáculos de forma autónoma, pueden desplegarse en situaciones peligrosas, reduciendo el riesgo para los humanos. Esto es especialmente relevante en aplicaciones como la exploración de áreas contaminadas o la intervención en incendios, donde los robots móviles pueden ingresar a zonas inaccesibles o peligrosas para los rescatistas, aumentando la seguridad y eficacia de las operaciones.

Otro aspecto clave es la optimización de recursos. Un sistema de navegación autónoma eficiente y robusto puede reducir los costos operativos al aumentar la durabilidad y autonomía

del robot. La correcta gestión de sus recursos es esencial para prolongar las misiones del robot y asegurar su funcionalidad en diversos entornos.

1.4 Objetivos

1.4.1 Objetivo general

Diseñar y simular un robot móvil de seis ruedas que sea capaz de superar obstáculos simples como escaleras y pendientes, manteniendo su estabilidad mientras navega en entornos urbanos.

1.4.2 Objetivos específicos

1. Diseñar un robot de seis ruedas con una suspensión tipo *bogie* que sea capaz de superar los obstáculos.
2. Diseñar un algoritmo de detección que permita percibir obstáculos a una distancia menor a tres metros utilizando una cámara de profundidad RGB-D.
3. Diseñar un mecanismo de estabilización para la cámara de profundidad que asegure su estabilidad durante la detección de obstáculos.
4. Crear un mecanismo que permita al robot subir escaleras de hasta 11 cm de altura sin volcamiento.
5. Diseñar un algoritmo que ajuste la velocidad de las ruedas del robot para superar los obstáculos identificados de manera estable.
6. Implementar un algoritmo que permita integrar los sensores y controladores necesarios para la superación de los obstáculos.
7. Simular el funcionamiento del robot en un entorno virtual para validar que el modelo de control predictivo proporciona velocidades adecuadas para la navegación.

1.5 Marco teórico

El marco teórico de este proyecto se basa en una amplia revisión de investigaciones previas que abordan la navegación autónoma y la superación de obstáculos en robots móviles. Estos aspectos son fundamentales para garantizar el funcionamiento efectivo de los robots en una variedad de entornos. A través de esta revisión, se han identificado principios fundamentales, enfoques innovadores y tecnologías emergentes que orientan el diseño y desarrollo del robot de seis ruedas para navegación autónoma y superación de obstáculos.

1.5.1 *Detección y mapeo de escaleras en robots cuadrúpedos*

La detección y el mapeo de escaleras son aspectos fundamentales en la navegación autónoma de robots móviles en ambientes urbanos. Woo *et al.* propusieron un método que utiliza nubes de puntos y modelado de escaleras para mejorar la detección y navegación en entornos con limitaciones de alcance sensorial [2]. Esta investigación fusiona datos de sensores de profundidad con modelos estimados de escaleras, permitiendo a los robots cuadrúpedos sortear este tipo de obstáculos de manera más eficiente.

Figura 1.2

Nube de puntos tridimensional mediante cámara de profundidad



Nota: Imagen extraída de [2]

Matsumura y Premachandra desarrollaron una técnica utilizando datos de nubes de puntos 3D y técnicas de deep learning para la detección de escaleras [3]. Su método demostró una alta precisión en la detección de escaleras, especialmente relevante para la prevención de accidentes en entornos donde la movilidad segura es crucial, como en el caso de personas con discapacidad visual.

1.5.2 Detección y superación autónoma de escaleras en robots de orugas

La detección y superación autónoma de escaleras son desafíos importantes en la robótica móvil. Guo *et al.* presentaron un algoritmo que utiliza sensores Kinect v2 para detectar y superar escaleras en robots de orugas [4]. Este sistema mejora la capacidad del robot para

calcular el ángulo de inclinación de las escaleras y planificar su trayectoria de manera eficiente, lo que resulta en una navegación más segura y fluida.

Figura 1.3

Robot móvil con orugas



Nota: Robot con orugas adaptado de [4]

Ben-Tzvi, Ito y Goldenberg desarrollaron un diseño de robot móvil capaz de subir y bajar escaleras de manera autónoma [5]. Este robot utiliza sensores y algoritmos de inteligencia artificial para ajustar su configuración de orugas y superar obstáculos de manera efectiva. Su diseño reconfigurable permite una adaptabilidad excepcional en una variedad de entornos, lo que lo convierte en una solución versátil para la navegación autónoma en diferentes terrenos.

1.5.3 Reconocimiento de la geometría del terreno en tiempo real

El reconocimiento de la geometría del terreno en tiempo real es esencial para la navegación segura de robots móviles. Pico *et al.* propusieron un método para reconocer la geometría del terreno utilizando sensores de escaneo láser [6]. Este enfoque permite que los robots ajusten su control según la información del terreno, mejorando la estabilidad y la capacidad de superar terrenos diversos. La capacidad de adaptación en tiempo real a diferentes condiciones del terreno es crucial para garantizar una navegación segura y eficiente en entornos dinámicos.

Figura 1.4

Robot superando obstáculos usando sensores laser



Nota: Imagen del robot obtenida de [6]

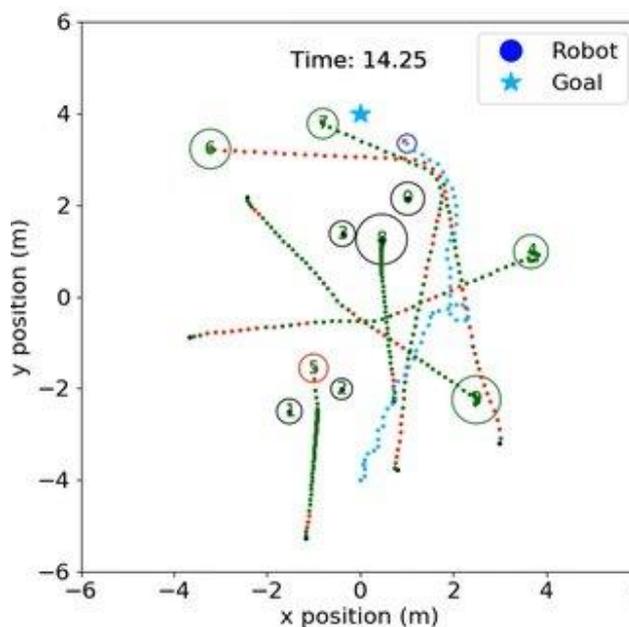
1.5.4 Navegación autónoma y salto de obstáculos en robots cuadrúpedos

La navegación autónoma y el salto de obstáculos son áreas de investigación activas en la robótica móvil. Scott *et al.* investigaron la navegación autónoma y el salto de obstáculos en robots cuadrúpedos [7]. Utilizaron un modelo dinámico simplificado y una optimización basada en la colisión para generar trayectorias de salto que permitan al robot evitar obstáculos mientras se encuentra en vuelo. Este enfoque representa un avance significativo en la capacidad

de los robots para navegar en entornos desafiantes, como terrenos irregulares o con obstáculos imprevistos.

Figura 1.5

Comparación de seis funciones de recompensa en el escenario de objetos



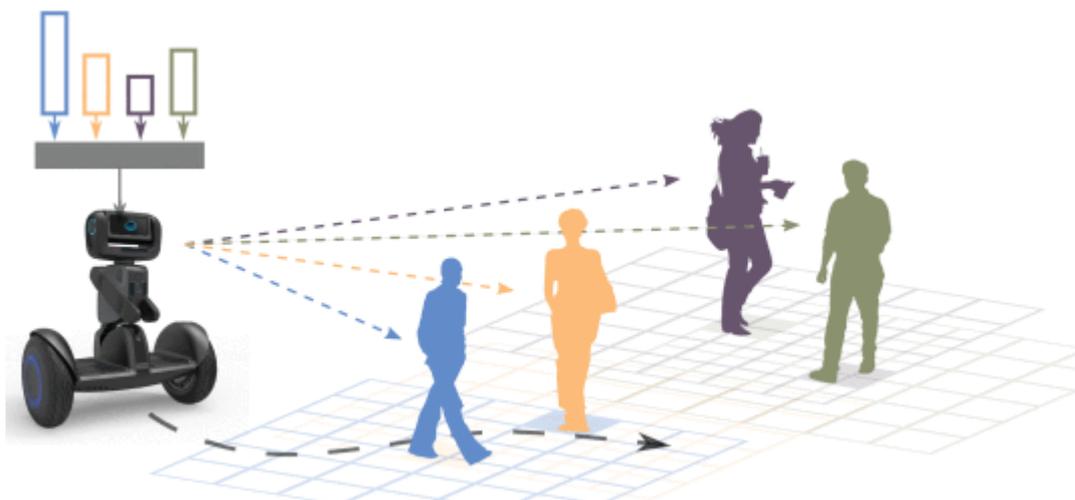
Nota: Gráfico comparativo de las funciones de recompensa obtenida de [7]

1.5.5 Diseño de robots con ruedas modificables

El diseño de robots con ruedas modificables es una estrategia innovadora para mejorar la capacidad de superar obstáculos. Chen *et al.* presentaron un robot con ruedas modificables que pueden cambiar su forma para adaptarse al terreno [8]. Este diseño permite que el robot altere su modo de locomoción según el terreno, mejorando su adaptabilidad y eficiencia en la superación de obstáculos. La capacidad de cambiar dinámicamente entre diferentes modos de locomoción aumenta la versatilidad del robot y su capacidad para enfrentar una variedad de desafíos en entornos no estructurados.

Figura 1.6

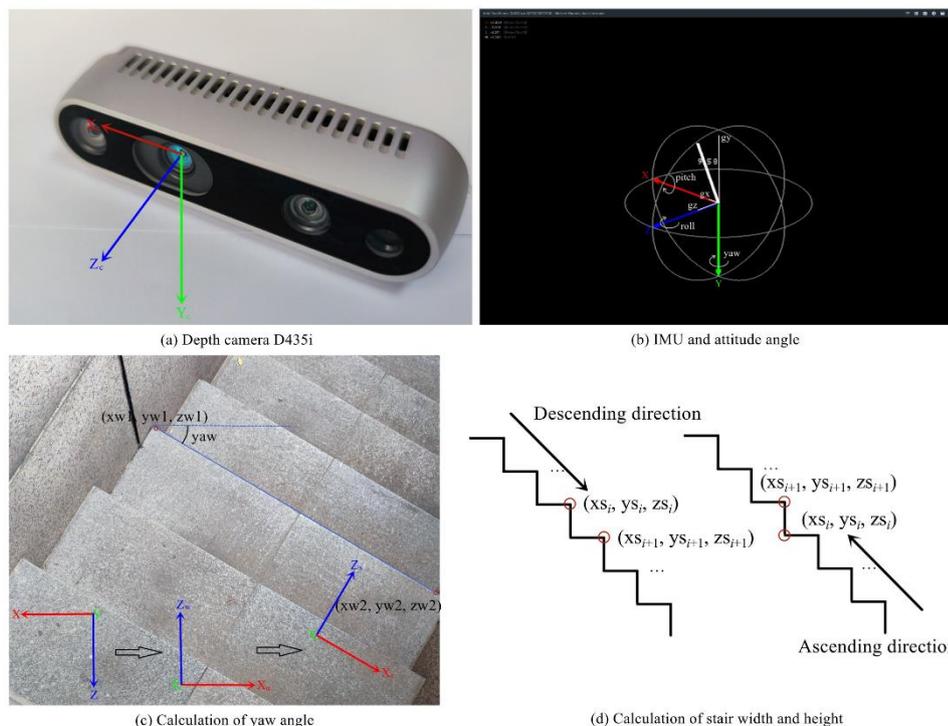
Modelo que junta las interacciones del robot en multitudes



Nota: Generación de trayectorias en base a multitudes, tal como se describe en [8]

1.5.6 Estimación de escaleras con redes neuronales y cámara RGB-D

En su investigación, Wang y colaboradores proponen una red neuronal con una cámara RGB-D para realizar una estimación precisa de escaleras [9]. Esta contribución es relevante para el proyecto actual, que se enfoca en mejorar la capacidad de los robots para superar obstáculos, incluidas las escaleras. El enfoque de Wang *et al.* en la detección precisa de escaleras, especialmente en condiciones de iluminación reducida, en lo que destacaron la efectividad en baja iluminación logrando errores de 15 mm en ascenso y 25 mm en descenso. Dentro de su investigación se proporcionaron mapas de profundidad y anotaciones sobre futuras investigaciones asociadas con este método.

Figura 1.7*Navegación en base a dificultad de ambiente**Nota: Imagen extraída de [9]*

1.5.7 Model Predictive Control (MPC) para la navegación del robot

El Model Predictive Control (MPC) es una técnica de control avanzada que se adapta bien a la navegación de robots móviles. Se utiliza para ajustar la velocidad de las ruedas del robot según los obstáculos a superar. Este algoritmo, como se discute en la investigación sobre navegación de robots móviles utilizando MPC basado en pasividad, realizado por Tahirovic, Magnani y Rocco, emplea ecuaciones de movimiento para predecir cómo la velocidad de las ruedas afectará la posición y orientación del robot en el tiempo [10]. Esto permite calcular trayectorias óptimas para anticipar su respuesta a diferentes obstáculos, garantizando una navegación segura hacia el objetivo deseado mediante ajustes continuos en la trayectoria en respuesta a cambios en el entorno o los objetivos de navegación.

Capítulo 2

2.1 Metodología

Se analizó los diversos diseños del robot propuesto que puede llegar a navegar de manera autónoma superando obstáculos en entornos urbanos. De estos diseños se seleccionó el más adecuado que cumpla con las funciones esperadas conociendo sus limitaciones.

2.1.1 Propuesta y selección de la solución

El principal desafío en la creación de un robot capaz de superar diversos obstáculos es desarrollar un diseño que garantice estabilidad y proteja sus componentes de posibles daños. Dado que el robot se empleará en entornos urbanos, debe ser capaz de superar escalones que pueden variar en tamaño, pero en este diseño se ha optado por especificaciones directas del cliente para superar escalones de hasta 11 cm sin riesgo de volcamiento. Además, el algoritmo debe ser capaz de identificar obstáculos a una distancia de hasta 3 metros. Por lo tanto, se consideraron las siguientes alternativas de solución:

2.1.1.1 Solución A: Robot con mecanismo de tracción.

Figura 2.1

Robot con mecanismo de tracción



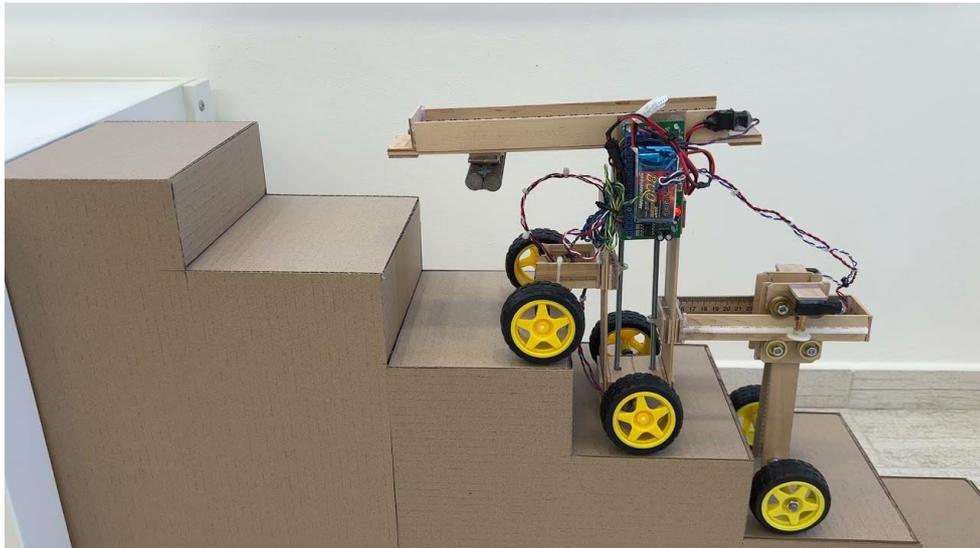
Nota: Imagen extraída de [11]

Este diseño presenta un robot equipado con un mecanismo de tracción, el cual es responsable de mover el robot hacia adelante y hacia atrás, así como girar. Este mecanismo es ideal para superficies relativamente planas y proporciona una tracción confiable en condiciones variadas. El diseño puede incluir ruedas o cadenas que permiten una movilidad eficiente y controlada, adecuado para entornos donde la estabilidad y la simplicidad son clave.

2.1.1.2 Solución B: Robot con mecanismos de elevación.

Figura 2.2

Robot con mecanismo de elevación



Nota: Imagen extraída de [12]

En esta alternativa el robot está equipado con un mecanismo de elevación que le permite ajustar su altura o levantar objetos. Este mecanismo es útil en tareas donde se requiere acceder a diferentes niveles o manipular objetos a distintas alturas. El diseño puede incorporar plataformas elevadoras o sistemas de actuadores lineales, facilitando la interacción con el entorno en aplicaciones industriales o de servicio.

2.1.1.3 Solución C: Robot con mecanismo *bogie*.

Figura 2.3

Robot con mecanismo bogie



Nota: Imagen extraída de [13]

Este diseño de robot presenta un mecanismo de suspensión tipo *bogie*, conocido por su capacidad de adaptarse a terrenos irregulares y superar obstáculos. Este sistema se caracteriza por sus múltiples puntos de pivote y su capacidad de mantener las ruedas en contacto con el suelo, distribuyendo el peso del robot de manera uniforme. Es especialmente adecuado para exploraciones en terrenos accidentados, como los que se encuentran en misiones de exploración planetaria.

Para poder seleccionar de manera adecuada el diseño del mecanismo del robot adecuado se evaluaron las alternativas con los siguientes criterios:

- *Estabilidad:* El diseño debe mantener la estabilidad del robot durante la navegación y al superar obstáculos.
- *Adaptabilidad:* La capacidad del diseño para adaptarse a diferentes tipos de terreno y obstáculos.

- Costo: El costo de fabricación y mantenimiento del robot.
- Durabilidad: La vida útil del diseño y la resistencia a daños.
- Simplicidad: La simplicidad del diseño en términos de elementos usados para que el mecanismo funcione de manera correcta.

En base a los criterios anteriormente mencionados se evaluaron los criterios que tienen mayor peso y se seleccionó la solución ganadora en base a los mismos.

2.1.1.4 Matriz de decisión para evaluar las alternativas.

Tabla 2.1

Evaluación de los criterios de diseño del robot de seis ruedas

Estabilidad > Adaptabilidad = Costo > Durabilidad > Simplicidad							
Criterio	Estabilidad	Adaptabilidad	Costo	Durabilidad	Simplicidad	$\Sigma+1$	Ponderación
Estabilidad		1	0.5	1	0.5	4	0.267
Adaptabilidad	0		0.5	1	1	3.5	0.233
Costo	0.5	0.5		0.5	1	3.5	0.233
Durabilidad	0	0	0.5		1	2.5	0.167
Complejidad	0.5	0	0	0		1.5	0.100
						15	1

Al evaluar los criterios de diseño se resumió la relevancia de estos en la *Tabla 2.1*. El criterio más determinante es la estabilidad del robot, pues como se va a usar un algoritmo que reconoce obstáculos la estabilidad de la cámara es esencial para que el robot pueda superar obstáculos de manera correcta. Mientras que, el criterio con menor peso es la complejidad del

mecanismo, pues a pesar de que se busca un diseño simple que cumpla con todas las funciones, no es tan relevante el número de elementos usados para que el mecanismo funcione.

2.1.1.5 Evaluación de cada alternativa según los criterios.

Tabla 2.2

Evaluación de criterio, estabilidad

Solución A > Solución C > Solución B					
Estabilidad	Solución A	Solución B	Solución C	$\Sigma+1$	Ponderación
Solución A		1	0.5	2.5	0.417
Solución B	0		0.5	1.5	0.250
Solución C	0.5	0.5		2	0.333
Suma				6	1

Tabla 2.3

Evaluación de criterio, adaptabilidad

Solución C > Solución B > Solución A					
Adaptabilidad	Solución A	Solución B	Solución C	$\Sigma+1$	Ponderación
Solución A		0.5	0	1.5	0.250
Solución B	0.5		0	1.5	0.250
Solución C	1	1		3	0.5
Suma				6	1

Tabla 2.4*Evaluación de criterio, costo*

Solución C > Solución B > Solución A					
Costo	Solución A	Solución B	Solución C	$\Sigma+1$	Ponderación
Solución A		0.5	0.5	2	0.308
Solución B	0.5		0.5	2	0.308
Solución C	1	0.5		2.5	0.385
	Suma			6.5	1

Tabla 2.5*Evaluación de criterio, durabilidad*

Solución C > Solución B > Solución A					
Durabilidad	Solución A	Solución B	Solución C	$\Sigma+1$	Ponderación
Solución A		0.5	0.5	2	0.308
Solución B	0.5		0.5	2	0.308
Solución C	1	0.5		2.5	0.385
	Suma			6	1

Tabla 2.6

Evaluación de criterio, simplicidad del mecanismo

Solución C > Solución B > Solución A					
Complejidad	Solución A	Solución B	Solución C	$\Sigma+1$	Ponderación
Solución A		0.5	0.5	2	0.308
Solución B	0.5		0.5	2	0.308
Solución C	1	0.5		2.5	0.385
Suma				6	1

Se evaluaron cada uno de los diseños en base a los criterios previamente establecidos, y en la mayoría de las tablas la solución C permanece como opción ganadora, a excepción de la *Tabla 2.2* la cual compara la estabilidad de cada una de las opciones. A pesar de tener mayor peso la estabilidad en la *Tabla 2.1* todavía se tenía que evaluar las opciones en base a la información anteriormente presentada, generando así la siguiente tabla que evalúa las opciones en orden de prioridad:

Tabla 2.7*Matriz de decisión*

Alternativas	Criterios					Resultados	
	Estabilidad	Adaptabilidad	Costo	Durabilidad	Simplicidad	Σ	Prioridad
Solución A	0.111	0.058	0.071	0.051	0.031	0.322	2
Solución B	0.067	0.058	0.071	0.051	0.031	0.278	3
Solución C	0.089	0.117	0.089	0.064	0.031	0.390	1

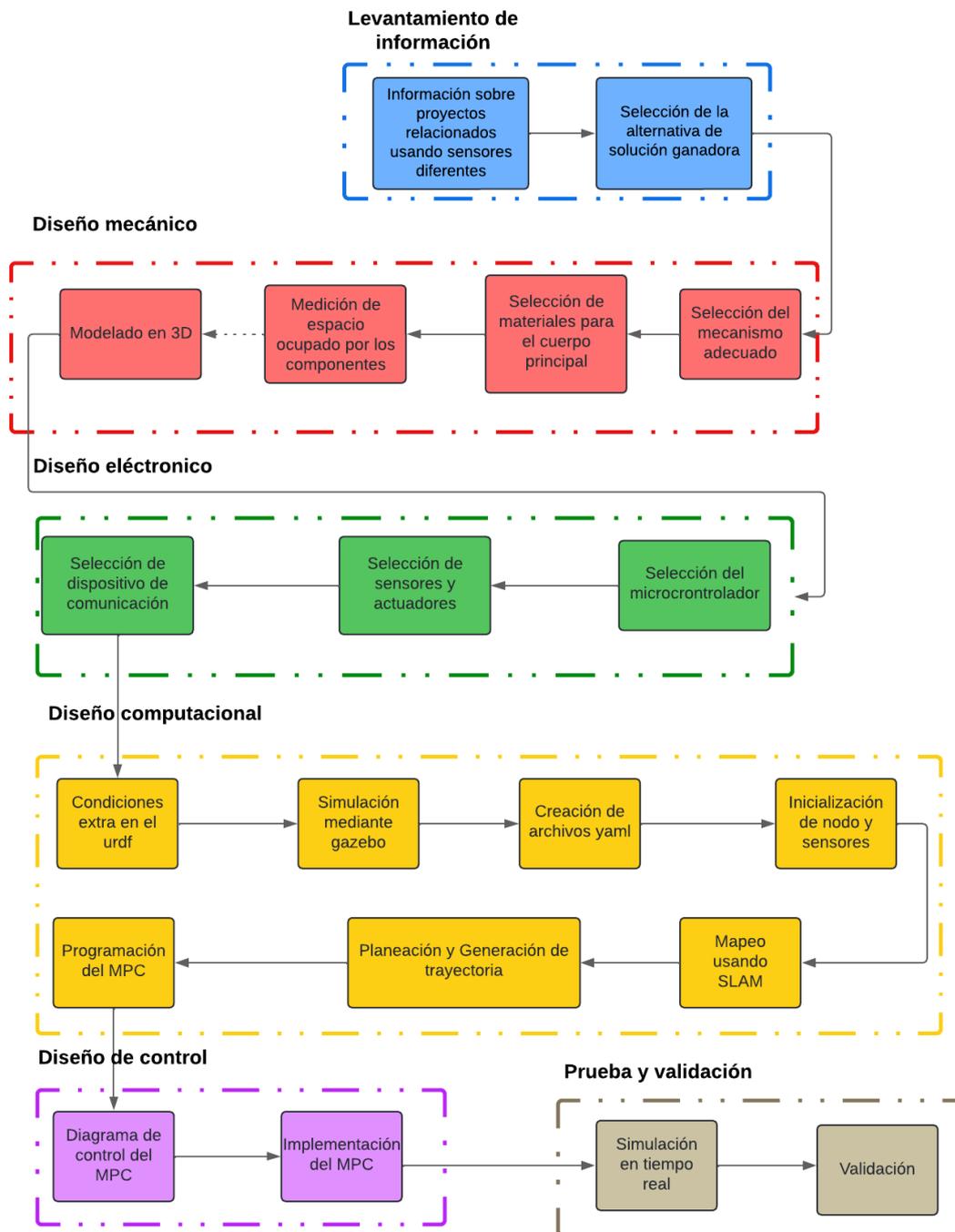
Al evaluar las distintas alternativas de solución con los criterios de diseño adecuados, se establecieron en la *Tabla 2.6* las prioridades de dichas opciones teniendo en cuenta los criterios anteriormente establecidos, concluyendo que la Solución C es la mejor opción.

2.2 Proceso de diseño

Para poder diseñar de manera correcta la solución previamente establecida, se establecieron diferentes etapas de diseño en las que se dividió el proyecto, las cuales fueron las siguientes:

Figura 2.4

Metodología de diseño



La metodología de diseño para un MPC de navegación autónoma de un robot de 6 ruedas fue organizada en varias fases clave. Comenzó con la recopilación de información y selección de la mejor solución (azul). Luego, se desarrolló el diseño mecánico mediante modelado 3D y selección de materiales y mecanismos adecuados (rojo). El diseño electrónico incluyó la elección de dispositivos de comunicación, sensores, actuadores y el microcontrolador (verde). En el diseño computacional, se configuraron condiciones en el URDF, simulación en Gazebo, creación de archivos YAML, inicialización de nodos y sensores, mapeo con SLAM, y generación de trayectorias (amarillo). Finalmente, se diseñó el MPC (morado) y se realizan pruebas y validación mediante simulaciones en tiempo real (marrón).

2.2.1 Requisitos de diseño

Para poder realizar un robot de 6 ruedas que fuera capaz de navegar de manera autónoma superando obstáculos fue necesario comprender cuales requisitos debe de cumplir la solución. En este caso los requisitos que la solución debió cumplir para navegar de manera autónoma fueron los siguientes:

Tabla 2.8*Requisitos de diseño*

Costo	El material y fabricación debe ser de bajo costo para un prototipo.
Medidas de los escalones	Se estableció que el robot sería capaz de superar escalones de 11 cm.
Reconocimiento de obstáculos	La distancia establecida para que el algoritmo reconociera los obstáculos fue de 3 m.
Model Predict Control (MPC)	El MPC debe ser capaz de calcular la velocidad adecuada de cada una de las 6 ruedas para tomar la ruta óptima.
Velocidad controlable	La velocidad a la que giran las ruedas debe ser controlable y medible.
Diámetro de las ruedas	Se utilizaron ruedas de 10 cm de diámetro.

2.2.2 Selección del motor

Para garantizar que el robot pueda superar obstáculos se tomó en cuenta las limitaciones mecánicas del robot y los requisitos técnicos, por lo que es esencial elegir los motores adecuados.

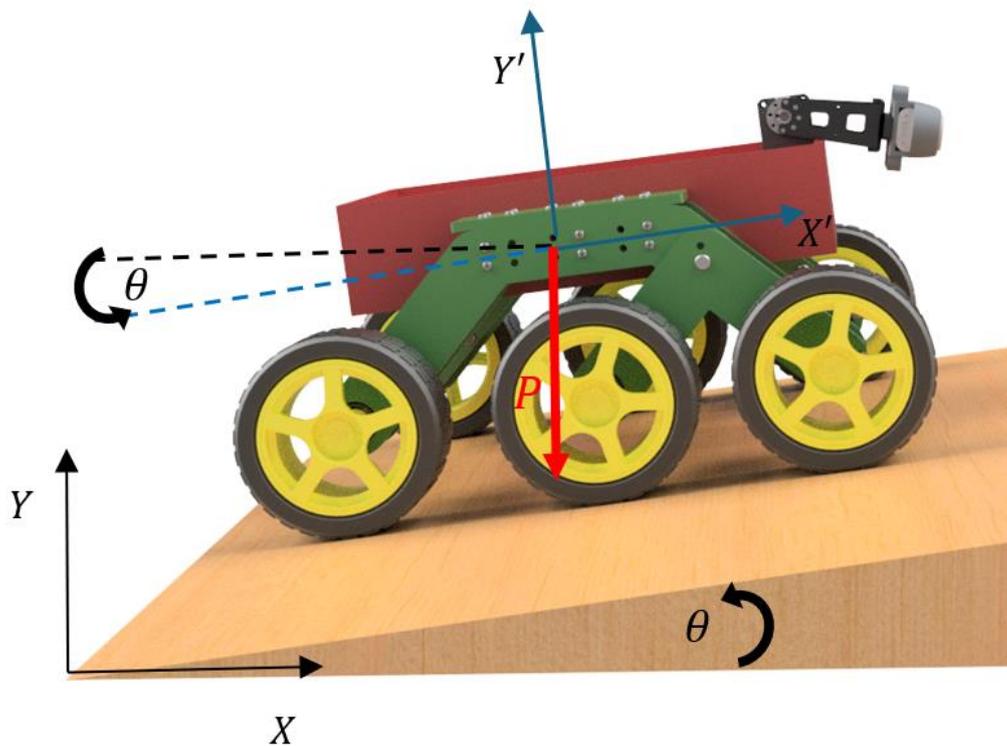
Para determinar los requisitos específicos del motor, se tuvo en cuenta los siguientes datos:

- El peso total del robot se estimó en 10 kg, cada rueda soporta aproximadamente 1.67 kg.
- El ángulo de pendientes es desde 5° hasta los 15°.
- La altura máxima del escalón (11 cm) en el cual se estima el ángulo desde el punto de contacto de la rueda con el suelo hasta el punto de contacto con la parte superior del primer escalón en 72.65°.

Se realizó cálculos del torque para diferentes pendientes y obstáculos con las fórmulas mencionadas a continuación:

Figura 2.5

DCL del robot en una pendiente



- Fuerza Total:

$$F = Pgsi n \theta \quad (1)$$

Donde:

- P es el peso del robot
 - g es la aceleración debido a la gravedad ($9.81 \frac{m}{s^2}$)
 - θ es el ángulo de la pendiente
- Distribución de la fuera total por rueda:

$$F_{rueda} = \frac{F}{6} \quad (2)$$

- Torque necesario (τ):

$$\tau = F_{rueda} \times \frac{D}{2} \quad (3)$$

Donde:

- D es el diámetro de la rueda (0.1 m).

Para determinar el ángulo correspondiente a un obstáculo de 11 cm de altura (un escalón), consideramos las dimensiones del robot y la geometría involucrada en la superación del obstáculo.

- Dimensiones relevantes:

- Altura del obstáculo (h): 11 cm = 0.11 m.
- Diámetro de la rueda (D): 10 cm = 0.1 m.
- Radio de la rueda (r): $D/2 = 0.05$ m.

- Hipotenusa del triángulo formado:

Cuando la rueda se encuentra en la parte superior del obstáculo, forma un triángulo rectángulo donde:

$$H = \sqrt{(h + r)^2 + r^2} \quad (4)$$

Usamos la relación trigonométrica para encontrar el ángulo θ (en grados) en el triángulo formado:

$$\tan \theta = \frac{h + r}{r} \quad (5)$$

De donde:

$$\theta = \tan^{-1} \left(\frac{h + r}{r} \right) \quad (6)$$

Sustituyendo los valores:

$$\theta = \tan^{-1} \left(\frac{0.16}{0.05} \right) \quad (7)$$

Calculamos:

$$\theta = \tan^{-1}(3.2) \quad (8)$$

Finalmente:

$$\theta = 72.65^\circ \quad (9)$$

Donde, la altura del obstáculo es uno de los catetos, el radio de la rueda es el otro cateto. La hipotenusa es la línea desde el punto de contacto de la rueda con el suelo hasta el punto de contacto con la parte superior del obstáculo.

A partir de estos requisitos a cumplir se calculó el torque necesario para esta configuración del robot descritos en la *Tabla 2.9*.

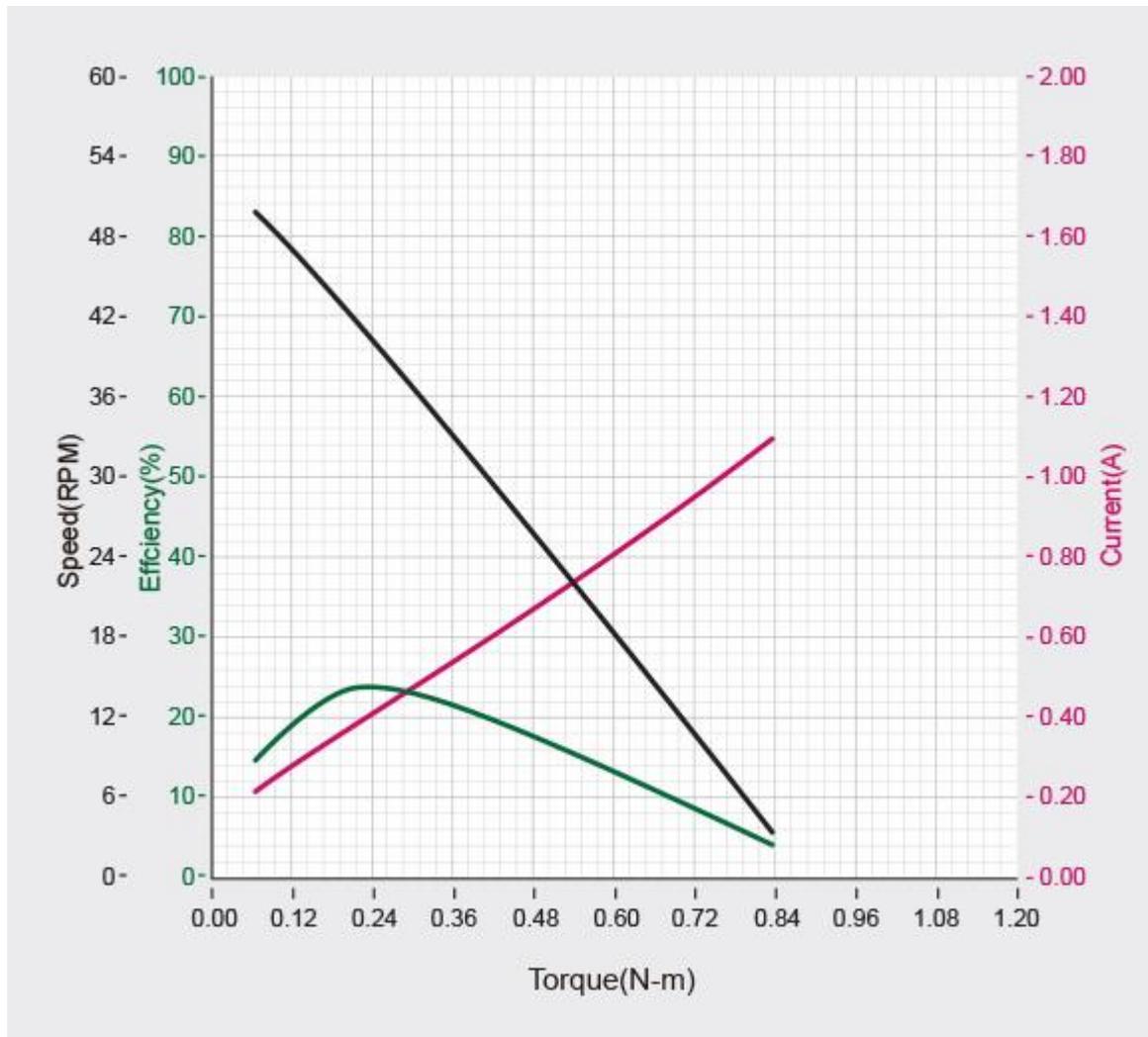
Tabla 2.9

Pendiente vs Torque

Pendiente (θ)	Fuerza total (f) [N]	Fuerza por rueda (f_{rueda}) [N]	Torque necesario (τ) [Nm]
5°	8.54	1.42	0.07
10°	17.03	2.84	0.14
15°	25.39	4.23	0.21
72.65°	93.64	15.6	0.78

Figura 2.6

Diagrama velocidad vs torque del motor



Nota: Imagen extraída de [14]

Basado en los datos de la *Tabla 2.9*, se seleccionó el servomotor DYNAMIXEL XL430-W250-T debido a sus especificaciones técnicas, que cumplen y superan los requisitos de torque necesarios para las pendientes y la altura máxima de escalones. Este servomotor fue seleccionado debido a su construcción robusta y su capacidad de control avanzado, incluyendo modos de velocidad, posición y PWM, ideales para la integración en el sistema de control del robot. Las especificaciones clave del servomotor son las siguientes:

- Torque máximo: 1.4 Nm (suficiente para superar el máximo torque calculado de 0.78 Nm).
- Modos de Operación: Incluye modos de control de velocidad, posición, y PWM.
- Control y Comunicación: Utiliza un algoritmo PID y comunicación TTL, necesarios para la integración en el sistema de control del robot.

Figura 2.7

Servomotor DC DYNAMIXEL XL430-W250-T



Nota: Imagen extraída de [14]

2.2.3 Selección de recursos electrónicos

1. Plataforma de procesamiento:

- Nvidia Jetson Nano: Esta plataforma proporciona un rendimiento computacional adecuado para aplicaciones de inteligencia artificial y aprendizaje automático. Es compatible con varios *frameworks* de inteligencia artificial (IA) y cuenta con una GPU de 128 núcleos CUDA, lo que facilita la ejecución de tareas complejas en tiempo real.

Figura 2.8

Nvidia Jetson Nano



Nota: Imagen extraída de [15]

2. Sensores y cámaras:

- Intel RealSense D435i (cámara de profundidad): La cámara Intel RealSense D435i ofrece una visión estéreo precisa con un rango de profundidad de hasta 10 metros y una resolución de 1280x720 píxeles. Su IMU incorporada facilita el seguimiento de movimientos y la navegación en entornos tridimensionales.

Figura 2.9

Intel Realsense D435i



Nota: Imagen extraída de [16]

3. Actuadores y sistemas de movilidad:

- *Servomotores DYNAMIXEL XL430-W250-T:* Estos servomotores tienen un torque de hasta 1.4 Nm y una velocidad de 46 RPM, adecuados para la movilidad y la manipulación precisa de la velocidad del robot.

Figura 2.10

Dynamixel XL430-W250-T

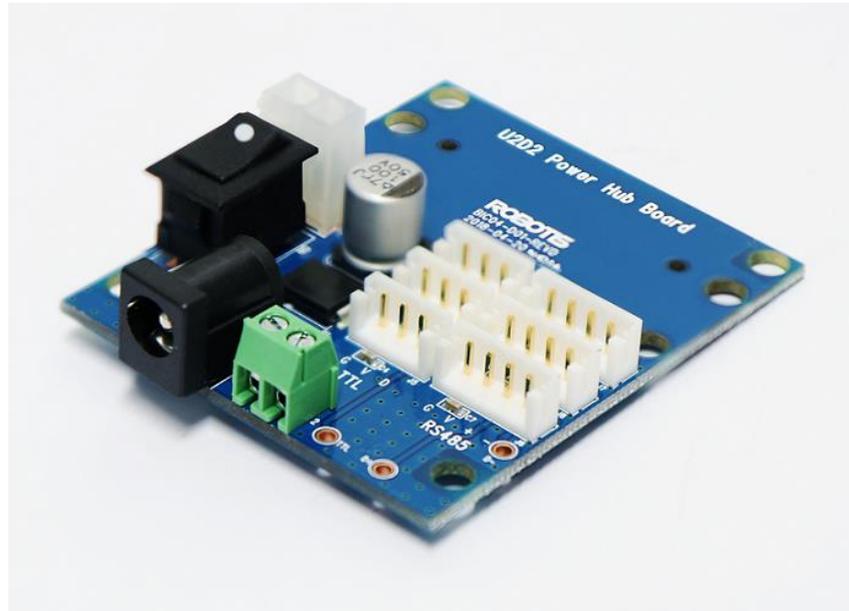


Nota: Imagen extraída de [14]

- Controlador para los servomotores U2D2: Este controlador permite una comunicación eficiente entre los servomotores y el sistema de control central.

Figura 2.11

Controlador U2D2 para servomotor Dynamixel

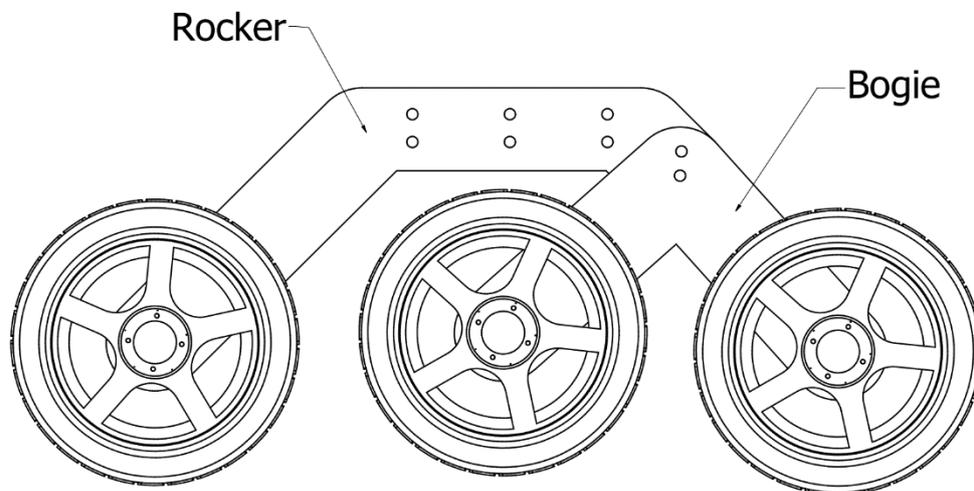


Nota: Imagen extraída de [17]

- Ruedas de 10 cm: Las ruedas de 10 cm de diámetro proporcionan una buena tracción y estabilidad en diversos terrenos.
- Sistema de suspensión adaptable (bogie): Este sistema de suspensión permite que el robot mantenga el equilibrio y se adapte a superficies irregulares, facilitando la navegación en terrenos accidentados.

Figura 2.12

Suspensión bogie



4. Software y licencias:

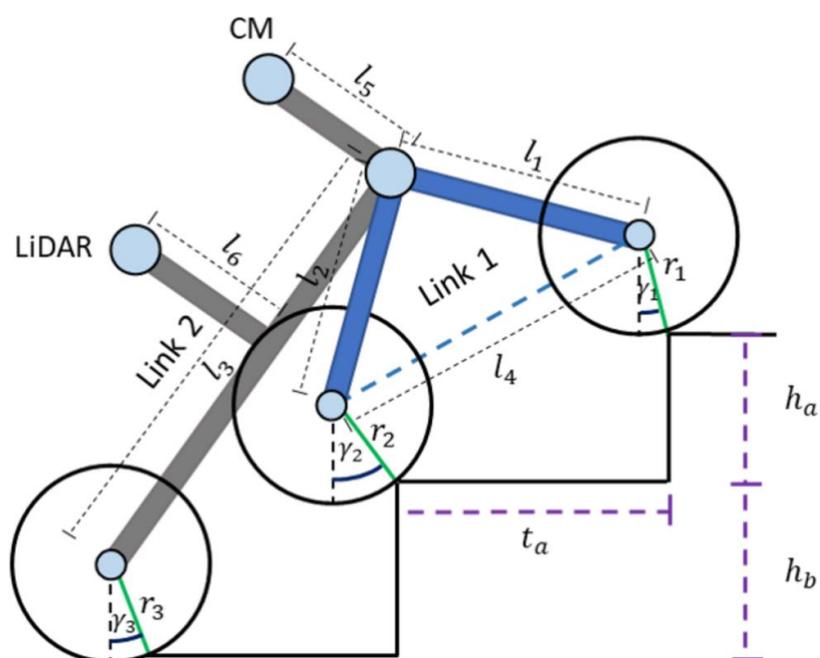
- ROS Melodic: Utilizado para la programación y control de robots, ROS *Melodic* es un sistema operativo que facilita la integración de diferentes componentes de *hardware* y *software*.
- Ubuntu 18.04: Sistema operativo que proporciona un entorno estable y compatible para el desarrollo y ejecución de aplicaciones de robótica.
- Inventor: Software de diseño asistido por computadora (CAD) utilizado para crear y optimizar modelos 3D de los componentes del robot.
- Onshape: Plataforma de CAD basada en la nube que permite la colaboración en tiempo real y la gestión eficiente de los diseños.
- Gazebo: Simulador de robots que permite la prueba y validación de algoritmos en un entorno virtual antes de su implementación en el hardware real.

- RViz: Herramienta de visualización en 3D utilizada en ROS para la interpretación y análisis de datos de sensores y la visualización del estado del robot.

2.2.4 Medidas de los eslabones

Figura 2.13

Robot con mecanismo de tracción



Nota: Imagen extraída de [18]

Tabla 2.10*Parámetros de la cinemática del robot*

Parámetros	Valores [m]
$(L \times W \times H)$	$0.26 \times 0.16 \times 0.06$
r_1	0.05
r_2	0.05
r_3	0.05
l_1	0.085
l_2	0.085
l_3	0.20
l_4	0.14

El robot móvil está formado por una estructura principal (chasis), un compartimento para los componentes internos y seis ruedas que se mueven de forma independiente gracias a unos motores (servomotores), como se puede observar en la Figura 2.13. Las ruedas delanteras y centrales se encuentran unidas mediante una articulación rotatoria pasiva llamada *bogie* y el eslabón 1, que tiene forma triangular y está definido por los parámetros l_1 , l_2 y l_4 . En este caso, l_1 representa la distancia entre el centro de la rueda delantera y el centro del *bogie*, l_2 la distancia entre el centro de la rueda central y el centro del *bogie*, y l_4 la distancia entre las ruedas delantera y central. La rueda trasera se conecta al chasis a través de un enlace imaginario que va desde el centro de la rueda hasta el centro del *bogie*, denominado eslabón 2, y que tiene como parámetro l_3 .

El punto de equilibrio del robot se considera fijo en el eslabón 1 y se mantiene estable durante el movimiento gracias a la rotación de la caja, la cual permanece horizontal en relación

con el suelo. La distancia entre el *bogie* y el centro de masa se denomina l_5 , mientras que l_6 representa la distancia entre el chasis y el *LiDAR*.

Las dimensiones y parámetros específicos del robot se detallan en la *Tabla 2.10*, y el diagrama esquemático del modelo 2D se muestra en la *Figura 2.12*.

Las restricciones cinemáticas del robot se analizan con base en los parámetros y eslabones del robot. La *Tabla 2.10* muestra los ocho parámetros del robot, incluyendo el radio de las ruedas y el centro de masa. Las siguientes ecuaciones detallan las restricciones basadas en las relaciones geométricas para prevenir colisiones entre los enlaces y las ruedas, así como la posición del *LiDAR* según sus especificaciones definidas de la siguiente manera:

1. Los parámetros l_1 , l_2 y l_4 forman el Enlace 1, y las restricciones están basadas en mantener el enlace como una forma triangular para mejorar la estabilidad del robot al subir y bajar escaleras.
2. El radio de las ruedas r_1 , r_2 y r_3 debe ser menor que los parámetros l_1 , l_2 y l_3 , respectivamente, para evitar colisiones entre las ruedas y los parámetros.
3. La suma de r_1 y r_2 debe ser mayor que l_4 para evitar colisiones entre las ruedas delanteras y centrales, r_1 y r_2 .
4. La diferencia entre l_3 y r_3 debe ser mayor que la suma de l_2 y r_2 para evitar colisiones entre las ruedas centrales y traseras, r_2 y r_3 , al subir o bajar escaleras.

$$l_2 + l_4 > l_1; l_1 + l_4 > l_2; l_1 + l_2 > l_4; \quad (10)$$

$$r_1 > l_i; \quad (11)$$

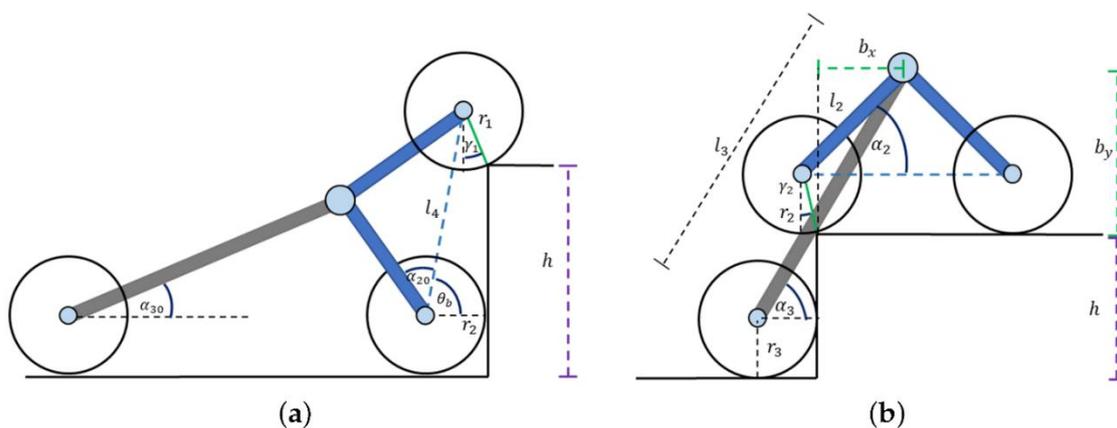
$$r_1 + r_2 < l_4; \quad (12)$$

$$l_3 - r_3 \geq l_2 + r_2; \quad (13)$$

Si un obstáculo no puede ser escalado, el robot debe buscar una ruta alternativa. Mantener el contacto de las ruedas con el suelo es esencial para la estabilidad, aunque en la práctica pueden perderse, comprometiendo la capacidad de movimiento. Para determinar la capacidad de escalar del robot, se calcula la altura máxima y la longitud mínima del escalón que puede superar, como se muestra en la Figura 2.11. La altura máxima puede variar según el tamaño del robot que está predefinida a 11 cm., γ_i es el ángulo de contacto entre cada rueda y el suelo, α_{20} denota el ángulo entre l_2 y l_4 , α_2 denota el ángulo entre l_2 y el eje horizontal global, θ_B denota la rotación máxima del *bogie*, α_{30} denota el ángulo inicial entre l_3 y el eje horizontal global cuando el robot está en una superficie plana, y α_3 denota el ángulo entre l_3 y el eje horizontal que varía con la inclinación del chasis. b_x es la distancia horizontal entre el centro del *bogie* y el punto de contacto de la rueda central, y b_y es la distancia vertical desde el centro del *bogie* hasta el punto de contacto de la rueda central.

Figura 2.14

Máxima altura que el robot puede subir y bajar un escalón



Nota: Imagen extraída de [18]

La altura máxima que el robot puede subir o bajar se determina con la Ecuación (20) después de obtener las alturas máximas para cada restricción.

- h_1 previene que las ruedas delantera y central suban o bajen juntas, como se muestra en la Figura 2.11. La rueda central mantiene contacto con el suelo antes de comenzar a escalar, y la rueda delantera debe mantener un ángulo de contacto inferior a 65 grados, basado en experimentos.
- h_2 es la rotación máxima del *bogie* en dirección contraria a las agujas del reloj (Figura 2.11).
- h_3 es la rotación máxima del *bogie* en dirección de las agujas del reloj (Figura 19b).
- Similar a h_1 , h_4 restringe que las ruedas central y trasera suban o bajen simultáneamente (Figura 2.11).
- h_5 es la altura máxima del escalón según la inclinación máxima del chasis para evitar caídas, obtenida de la Figura 2.11.

$$h_1 = \sqrt{l_2^4 - (r_2 - r_1 \sin(\gamma_1))^2} - r_1 \cos(\gamma_1) + r_2 \quad (14)$$

$$h_2 = l_4 \sin(\alpha_2 - \alpha_{20}) - r_1 \cos(\gamma_1) + r_2 \quad (15)$$

$$h_3 = l_3 \sin(\theta_B + \alpha_{30} + \alpha_2 - \alpha_{20}) - b_y + r_3 \quad (16)$$

$$h_4 = \sqrt{l_3^2 - (r_3 + b_x)^2} - b_y + r_3 \quad (17)$$

$$h_5 = l_3 \sin \alpha_3 - b_y + r_3 \quad (18)$$

$$\min(h_i) = h_{max} \quad (19)$$

La altura del escalón (h), que debe ser menor o igual a la altura máxima permitida (h_{max}), determina la longitud mínima del escalón para que el robot suba o baje. Si la altura del escalón es menor que la altura máxima, la longitud mínima es la suma de la longitud de l_4 , el

radio de las ruedas delanteras y traseras (r_1 y r_2) y la distancia horizontal entre los puntos de contacto de las ruedas centrales y traseras. Si h es igual a h_{max} y el robot es bajo, la longitud mínima del robot es la longitud total.

Se introduce un parámetro " d " para representar la distancia entre la rueda central y el escalón cuando hay dos ruedas en contacto. Hay un espacio entre el escalón y la rueda si d es mayor que cero. Se calculan t_1 y t_3 , que representan las distancias horizontales entre las ruedas delantera y central y trasera en contacto con la banda de rodadura, respectivamente, si h es menor que h_{max} . Considerando el caso de dos ruedas en contacto simultáneo, la distancia mínima requerida es la mayor distancia entre t_1 y t_2 .

Finalmente, en el tercer caso, el robot puede subir y bajar mientras las tres ruedas permanecen en el suelo en tres escalones distintos.

2.2.5 Diseño de la estructura principal del robot

El diseño simplificado del robot se compone de las siguientes partes mecánicas: una caja de dimensiones (16 x 6 x 26) m, 4 eslabones con longitud l_3 , 4 eslabones con longitud l_2 y l_3 . Además de utilizar el mecanismo de estabilización de la cámara y los diversos componentes electrónicos para que el robot pueda moverse. Por lo cual utilizando la *Tabla 2.10*, se realizaron los diseños de cada eslabón, además de dimensionar la caja para insertar los componentes electrónicos.

El robot tuvo que ser diseñado de tal manera que tuviera un mecanismo similar al siguiente:

Figura 2.15

Robot con sistema de bogie

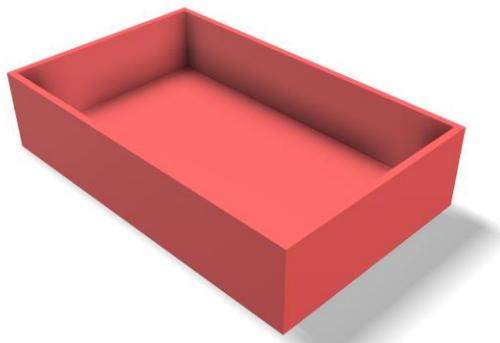


Nota: Imagen extraída de [19]

En primer lugar, las dimensiones de la caja seleccionadas y presentadas la *Tabla 2.10* se realizó un diseño en inventor que tuviera esas dimensiones, siendo el siguiente:

Figura 2.16

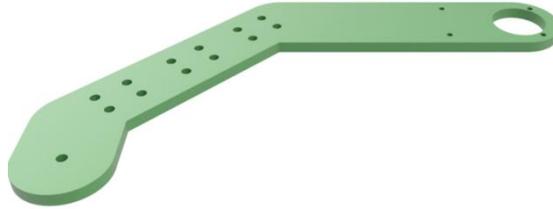
Caja del robot bogie



Además, el eslabón que representa l_3 también se lo diseñó de la siguiente manera:

Figura 2.17

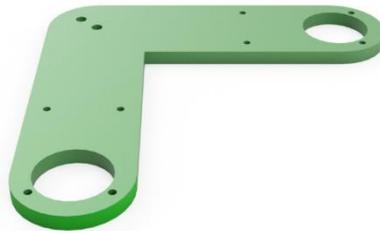
Eslabones posteriores del mecanismo



En cuanto a las medidas de l_2 y l_3 sirvieron para crear el siguiente eslabón de las ruedas delanteras:

Figura 2.18

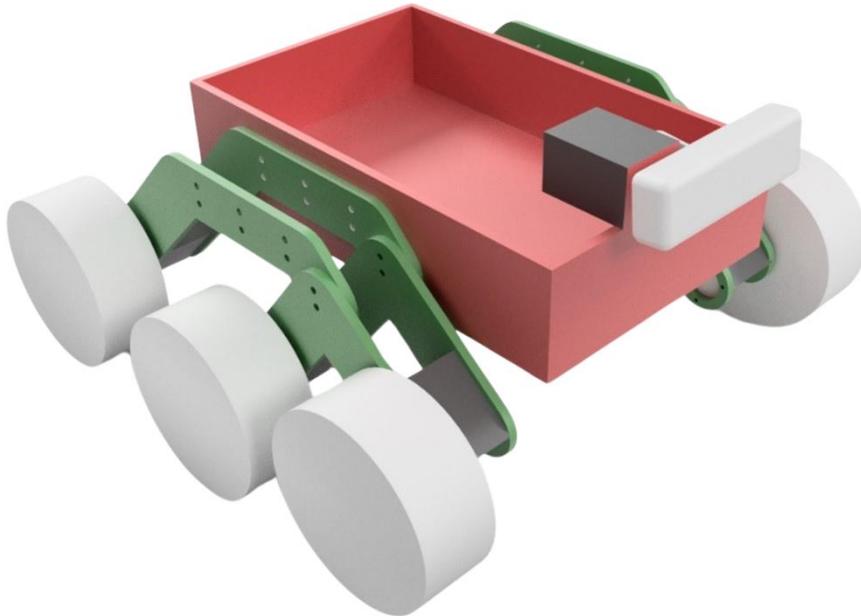
Eslabones anteriores del mecanismo



Cuando se ensambla todos los componentes mecánicos del robot se tiene el siguiente resultado preliminar:

Figura 2.19

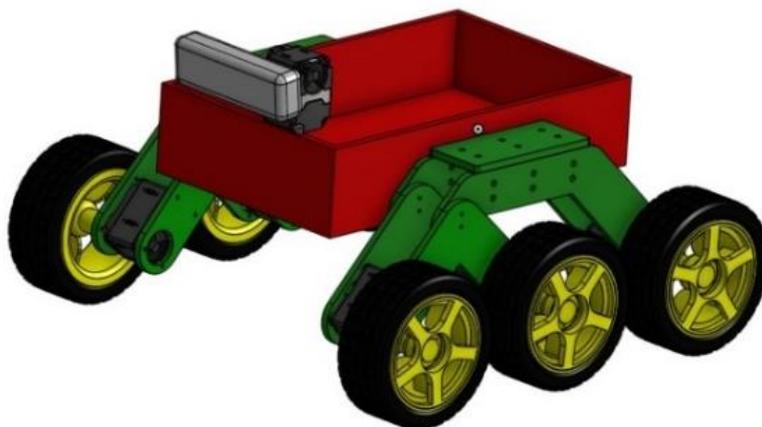
Diseño preliminar del bogie



Tomando en cuenta el diseño mecánico simplificado del robot anteriormente hecho, para poder pasar ese diseño al entorno virtual de ROS, fue necesario el uso del paquete *Onshape-to-robot*, el cual trabaja con el diseño del robot en la herramienta de diseño online *Onshape*, siendo el siguiente:

Figura 2.20

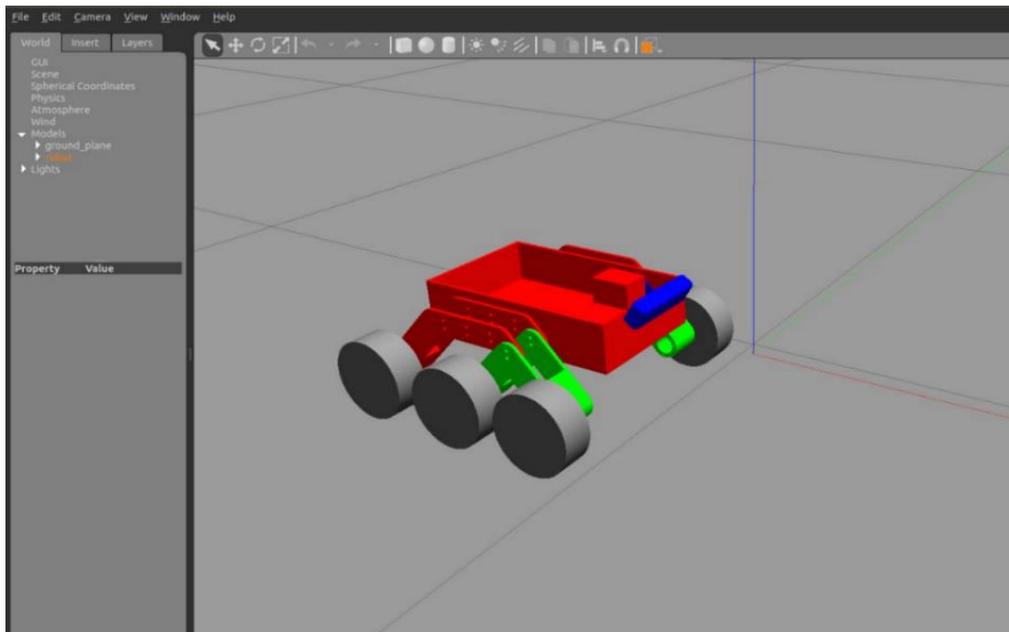
Modelo del robot en Onshape



Una vez teniendo dicho diseño, en un entorno virtual se lo convirtió a urdf, teniendo el siguiente diseño como resultado:

Figura 2.21

Modelo del robot en el ambiente de Gazebo

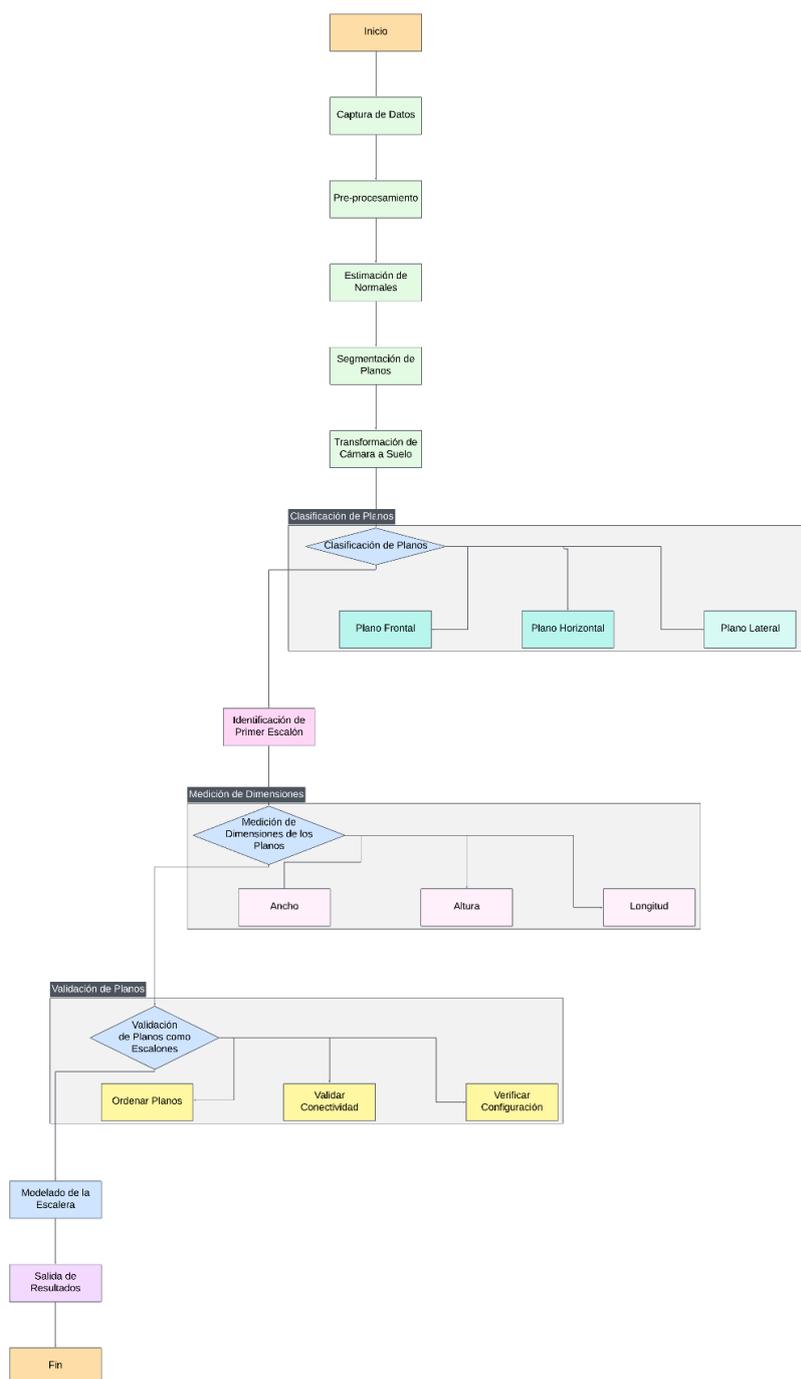


2.2.6 Algoritmo de reconocimiento de obstáculos

2.2.6.1 Detección y modelado de escaleras.

Figura 2.22

Diagrama de flujo de detección de escaleras



La detección de escaleras y pendientes es un proceso complejo que involucra varias etapas, desde la identificación inicial de planos en la nube de puntos hasta la validación final de la estructura modelada. A continuación, se describe paso a paso la metodología utilizada:

2.2.6.1.1 Detección y clasificación de planos.

- ***Identificación de Planos:*** Se detectan superficies planas en la nube de puntos utilizando algoritmos de segmentación. Estos planos se clasifican en función de su orientación espacial relativa a la cámara.
- ***Determinación del Primer Escalón:*** Para el caso de la escalera, se identifican los planos frontales como posibles candidatos para el primer escalón basándose en su distancia y orientación respecto al suelo. La distancia se calcula utilizando la fórmula de la distancia euclidiana en el espacio 3D.
- ***Dimensiones de los Escalones:*** Las dimensiones del escalón, como la longitud, el ancho y la altura, se derivan de las ecuaciones geométricas basadas en los puntos extremos del plano horizontal seleccionado.

2.2.6.1.2 Modelado de la escalera.

- ***Clasificación Inicial de Planos:*** Los planos detectados se clasifican como parte del suelo o posibles escalones según su altura relativa al suelo y para pendientes de acuerdo con un rango de ángulos posibles a ser pendientes.
- ***Separación de Vectores de Escalones:*** Los planos horizontales se evalúan para determinar si caen dentro del rango de altura adecuado para ser considerados escalones. Se utilizan límites de altura para clasificar los escalones ascendentes y descendentes.

- Orden de Planos: Los planos candidatos se ordenan según su altura para identificar correctamente el plano del suelo y los escalones subsecuentes en el caso de escaleras.

2.2.6.1.3 Verificación de conectividad y fusión de planos.

- Conectividad de Planos: Se verifica la conectividad de los planos con los niveles existentes utilizando una función de búsqueda de vecinos que calcula la distancia euclidiana entre puntos.
- Fusión de Planos Conectados: Si un plano está conectado al nivel más alto actual y la diferencia de altura está dentro de un umbral predefinido, el plano se fusiona con el nivel actual. La fusión implica combinar las nubes de puntos de los dos planos y recalcular la normal y el centroide del plano resultante.
- Estandarización de vectores Normales: Las normales de los planos fusionados se promedian y se normalizan para asegurar que el plano resultante tenga una orientación consistente.

2.2.6.1.4 Cálculo de dimensiones con pca y método manhattan.

- Análisis de Componentes Principales (PCA): Se utiliza PCA para alinear los datos de la nube de puntos con las direcciones principales de la escalera.
- Método Manhattan: Este método calcula las dimensiones basándose en la alineación de los puntos a lo largo de los ejes principales del sistema de coordenadas.

2.2.6.1.5 *Validación y ajuste de dimensiones.*

- Validación de Dimensiones: Las dimensiones del modelo de la escalera se validan contra especificaciones predefinidas para asegurar que cumplen con los rangos aceptables de seguridad y funcionalidad.
- Ajuste Iterativo: Si alguna dimensión cae fuera de los límites establecidos, el algoritmo ajusta los límites o detiene las revisiones después de un número máximo de iteraciones, asegurando que las dimensiones de la escalera reflejen con precisión la estructura real capturada por la cámara RGB-D.
- Captura de Dimensiones en Tiempo Real: El sistema captura dimensiones en tiempo real, incluyendo posibles planos no pertenecientes a la escalera y rampa, y ajusta los límites superior e inferior de cada dimensión para manejar estos errores.

2.2.6.1.6 *Visualización 3D.*

- Generación de Nubes de Puntos: El algoritmo calcula las posiciones de los puntos que forman cada plano tipo pendiente, peldaño y contrahuella, y los añade a las nubes de puntos.
- Transformación y Dibujo: Se transforman estas nubes utilizando una matriz para ajustar su posición, se dibujan los escalones y las contrahuellas, y se añaden textos descriptivos para indicar el número de cada escalón, asegurando una visualización tridimensional precisa de la escalera.

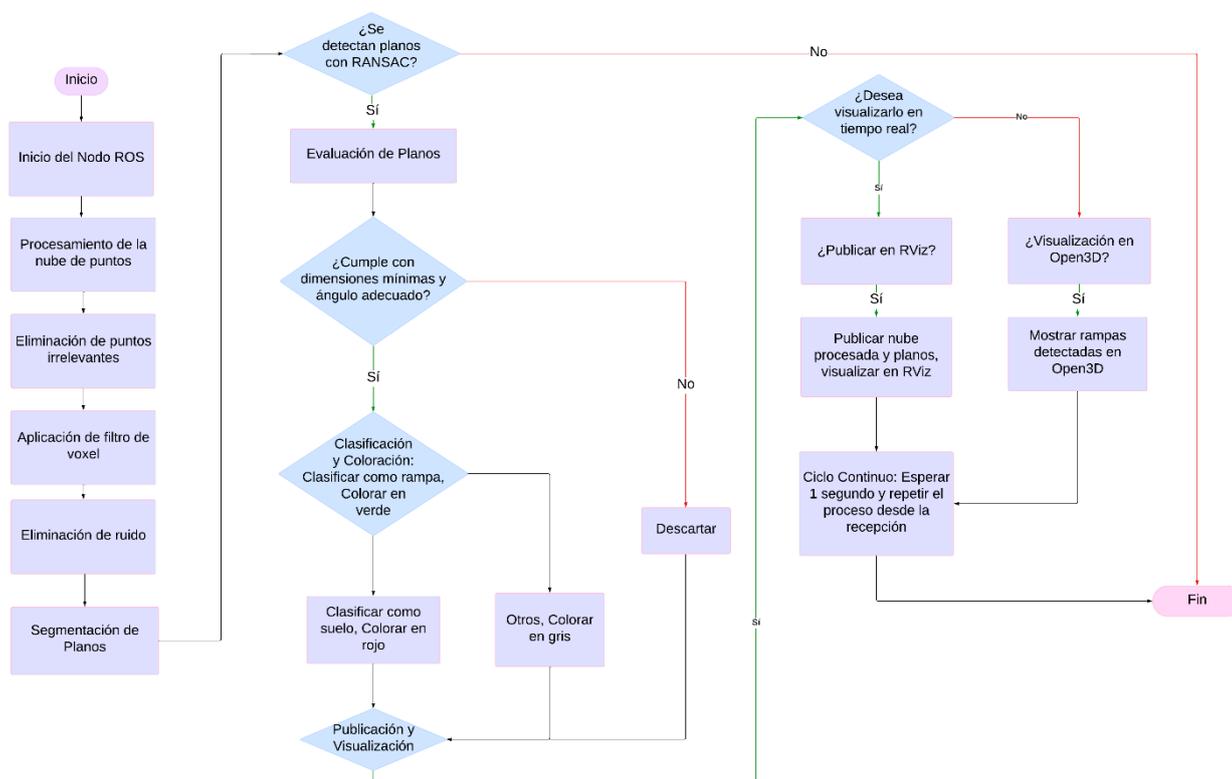
2.2.6.2 **Detección de rampas y visualización.**

El código para detección de rampas implementa un nodo de ROS (Robot Operating System) especializado en la detección de rampas a partir de nubes de puntos 3D capturadas por

la cámara de profundidad D435i. Este nodo opera en tiempo real y está diseñado para integrarse en sistemas de navegación autónoma, permitiendo la detección y clasificación de superficies inclinadas que puedan ser consideradas rampas.

Figura 2.23

Diagrama de flujo de la detección de pendientes



2.2.6.2.1 Adquisición y Preprocesamiento de Datos.

El nodo se suscribe al tópico de ROS que transmite las nubes de puntos generadas por la cámara de profundidad. Utilizando la biblioteca Open3D, los datos de la nube de puntos se preprocesan mediante varios pasos críticos:

- Eliminación de puntos irrelevantes: Se eliminan los valores NaN de la nube de puntos para asegurar la integridad de los datos.

- Reducción de la resolución (downsampling): Se aplica una voxelización para reducir la densidad de puntos y así optimizar el procesamiento.
- Eliminación de ruido: Mediante un filtro estadístico, se elimina el ruido presente en la nube de puntos, mejorando la calidad de los datos para la posterior detección de planos.

2.2.6.2.2 Segmentación y Detección de Planos.

Una vez preprocesada la nube de puntos, el nodo aplica un algoritmo de segmentación basado en RANSAC para detectar múltiples planos dentro de la nube. Estos planos son evaluados en función de su longitud y su orientación respecto al eje Z, que se asume como la dirección vertical en el sistema de coordenadas de la cámara.

- Evaluación de planos: Se utilizan umbrales predefinidos para determinar si un plano detectado puede considerarse una rampa. Los criterios incluyen un rango de ángulos específicos (entre 4° y 16°) y dimensiones mínimas (1 metro de longitud). Los planos que no cumplen con estos criterios son descartados.
- Clasificación de planos: Los planos que cumplen con los criterios de rampa se clasifican como tales y se les asigna un color para su visualización. Los planos horizontales identificados como suelo se colorean en otro color, mientras que los puntos restantes en la nube de puntos se visualizan en gris.

2.2.6.2.3 Publicación y Visualización de Resultados.

El nodo publica dos mensajes principales a través de ROS:

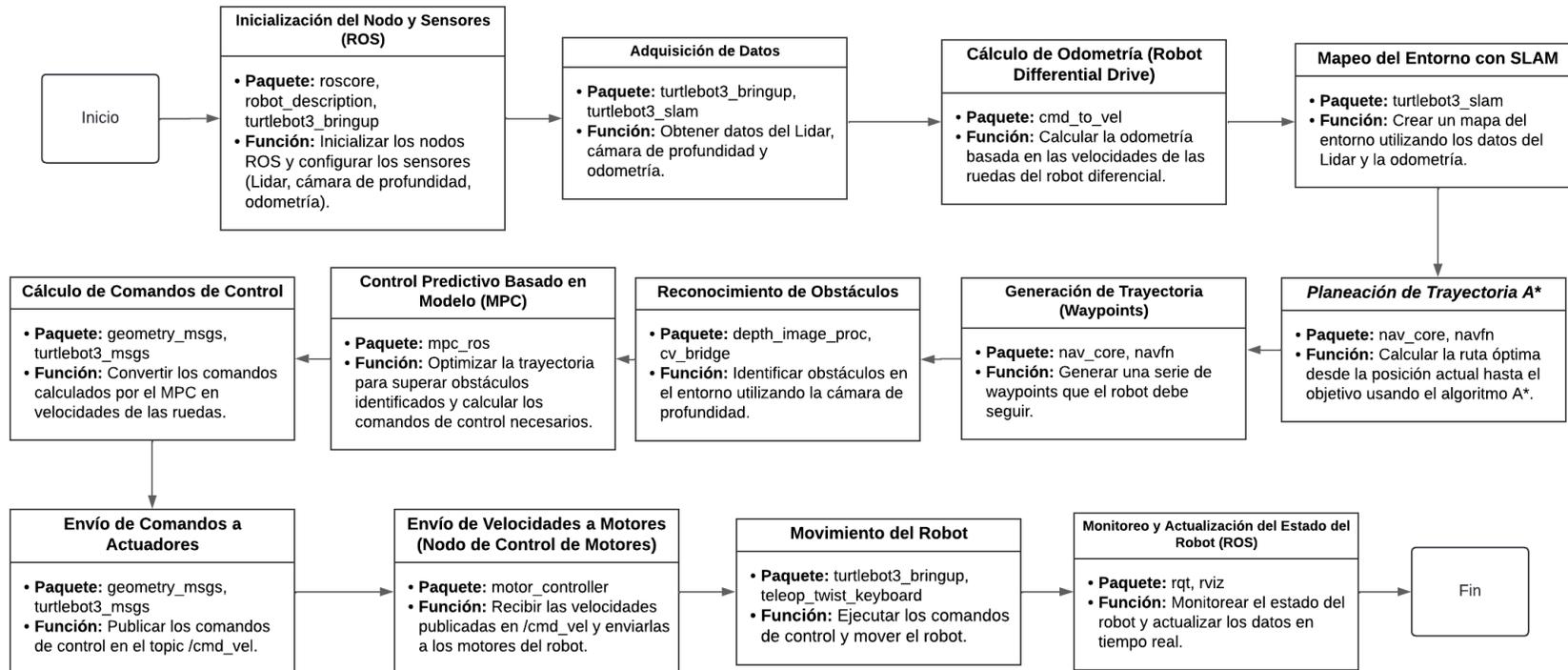
- Nube de puntos procesada: La nube de puntos, después de ser preprocesada y filtrada, se publica para su uso en otros nodos.
- Planos detectados: Los planos que cumplen con los criterios de detección de rampas se publican como un mensaje separado.

Ambos mensajes pueden ser visualizados en *RViz*, una herramienta de visualización en ROS, lo que permite observar en tiempo real la nube de puntos procesada y los planos de rampas detectados. Esta visualización es necesaria para el análisis del algoritmo, proporcionando una interfaz visual directa para observar cómo se comporta la detección en diversos entornos.

2.2.7 Model predict control (MPC) para controlar las ruedas

Figura 2.24

Model predict control con sus paquetes

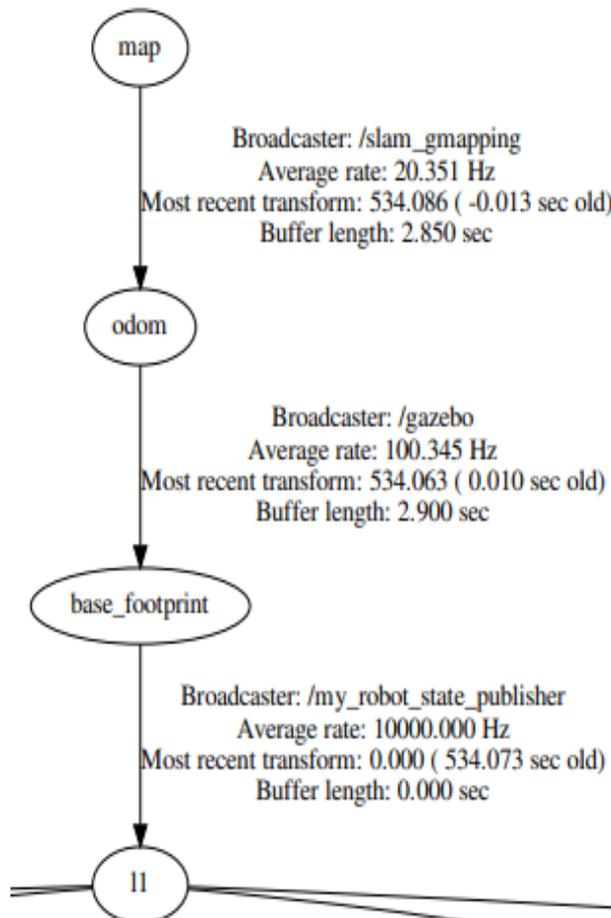


El proceso para implementar un sistema de control predictivo basado en modelos (MPC) para la navegación autónoma de un robot se divide en varias etapas cruciales. Comenzó con la inicialización de nodos y sensores ROS, configurando el *LiDAR* y la cámara de profundidad. Luego, se adquirieron datos para calcular la odometría del robot, esencial para conocer su posición y orientación en tiempo real. Utilizando los datos del *LiDAR* y la odometría, se creó un mapa del entorno mediante SLAM. Con este mapa, se planeó la trayectoria global usando el algoritmo A* y se generan *waypoints*, reconociendo obstáculos para evitar colisiones.

En el caso del TurtleBot3 normalmente se utiliza el *Dynamic Window Approach (DWA)* como planificador local. Sin embargo, en este proyecto se optó por utilizar MPC como planificador local. La elección de MPC se debe a su capacidad para prever y ajustar la trayectoria del robot de manera más eficiente y segura, considerando restricciones dinámicas y garantizando una mejor respuesta ante cambios en el entorno, lo cual mejora la autonomía y la robustez del sistema de navegación del robot.

Figura 2.25

Transformaciones entre frames para usar SLAM generando un mapa



Para poder utilizar el paquete *SLAM* se realizaron transformaciones desde el *frame map* hasta el *frame odom*, luego se hizo una transformación hasta el *frame* base del robot llamado *base_foorprint*. Aunque se podía haber realizado la secuencia inversa iniciando desde el *frame* base, esta secuencia de transformaciones asegura que el robot esté bien referenciado en *RViz*, para que se mueva de una manera similar a como lo hace en gazebo.

El *frame base_footprint* representó la posición actual del robot y fue crucial para coordinar sus movimientos en relación con la posición estimada por la odometría y ajustada por *SLAM*. Estas transformaciones permitieron integrar y corregir datos, asegurando que el robot mantuviera un conocimiento preciso de su posición y entorno. Sin estas transformaciones, el robot no podría haber ejecutado movimientos precisos ni navegado de manera eficiente, lo que fue fundamental para una navegación autónoma efectiva.

Capítulo 3

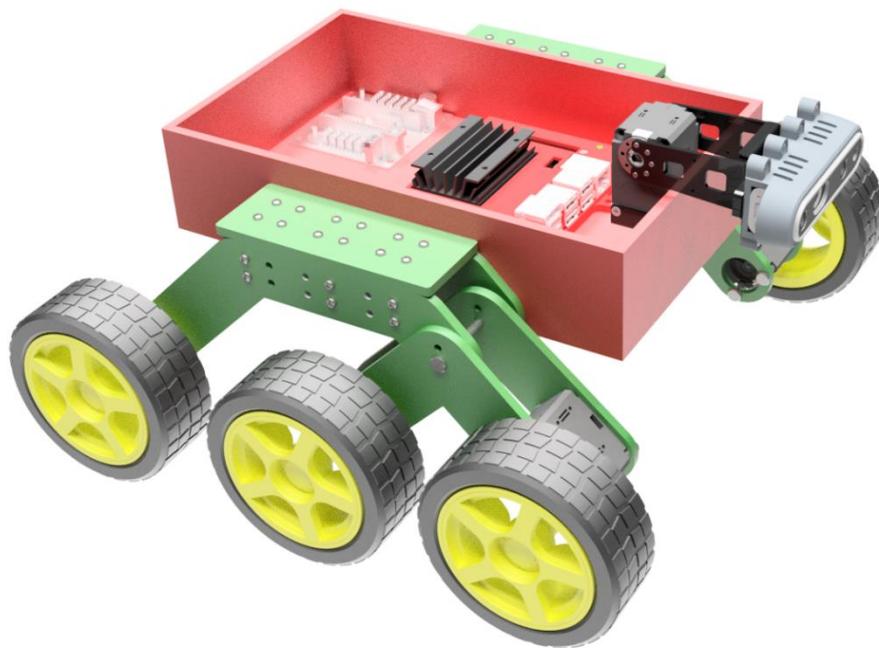
3.1 Resultados y análisis

3.1.1 Diseño final del robot

En base a los componentes electrónicos considerados se agregaron los mismos al diseño final. Los componentes principales son la cámara de profundidad, la cual sirvió para poder reconocer las pendientes y escaleras a superar. Además de constar del mecanismo para el movimiento de la cámara el cual permite que la imagen de los obstáculos se estabilice para poder reconocerlos.

Figura 3.1

Robot con sistema de bogie. con componentes



Además se modificó el diseño anterior en gazebo para poder agregar los sensores adecuados. Se agregó la cámara de profundidad D435i y un sensor *LiDAR*. También la cámara de profundidad se encuentra conectada al eslabón que representa el mecanismo de estabilidad para que la misma pueda rotar de manera adecuada y realizar el reconocimiento de obstáculos.

Figura 3.2

Mecanismo de la cámara

**Figura 3.3**

Robot con sistema de bogie dentro de Gazebo

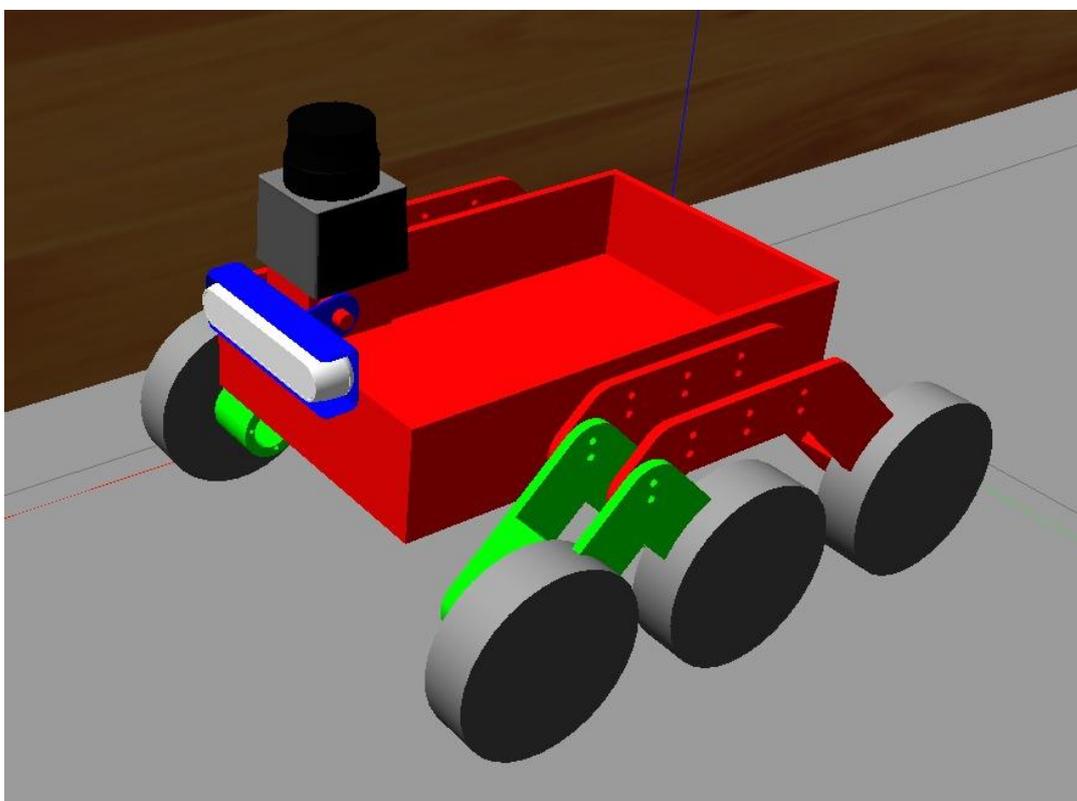
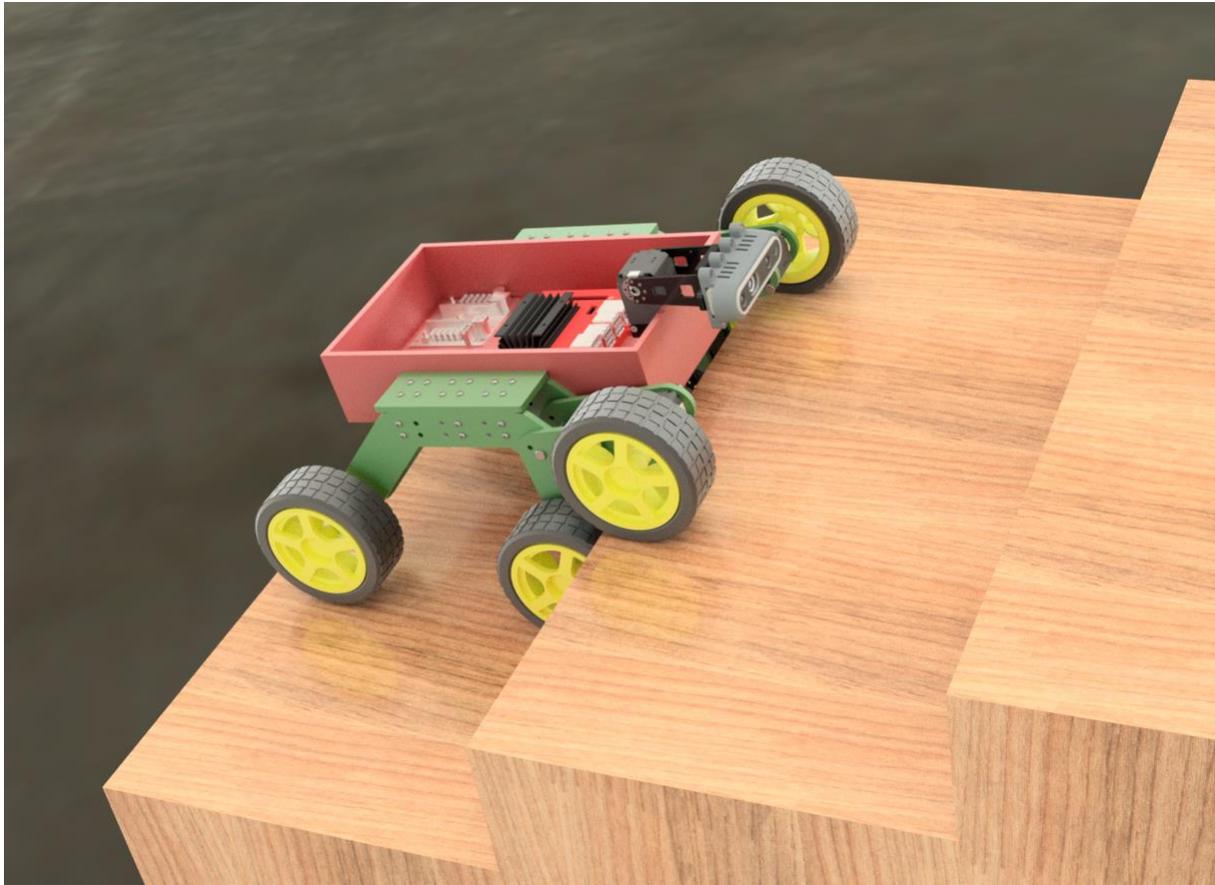


Figura 3.4

Diseño final del robot subiendo una escalera

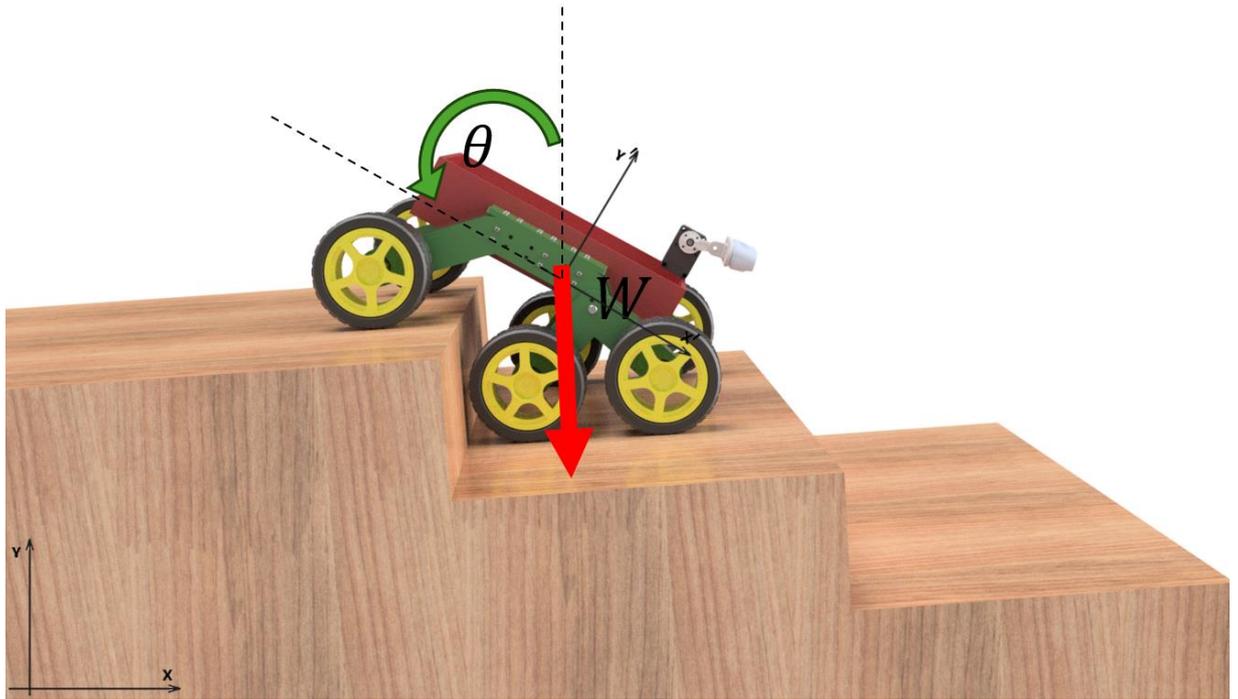


3.1.2 Análisis de esfuerzos

Para verificar que el material y diseño del robot fueran los adecuados en un entorno real se realizó un análisis de tensión usando la gravedad como fuerza externa. Teniendo en cuenta el siguiente diagrama de cuerpo libre cuando el robot soporta la mayor tensión, en este caso es cuando baja la escalera, pues los eslabones de la rueda delantera deben soportar el peso de la cámara.

Figura 3.5

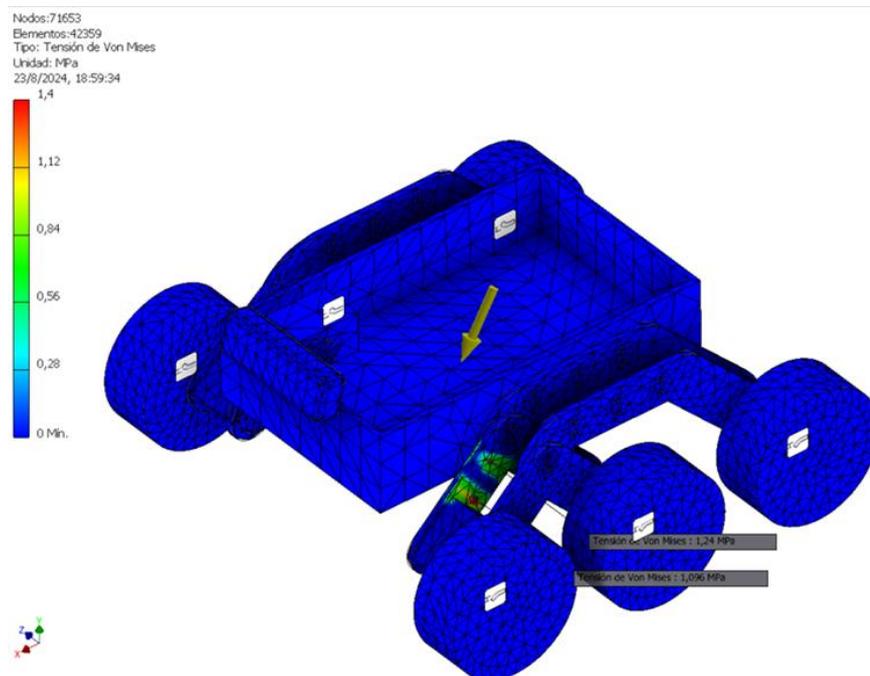
Robot con sistema de bogie en el peor escenario de tensión



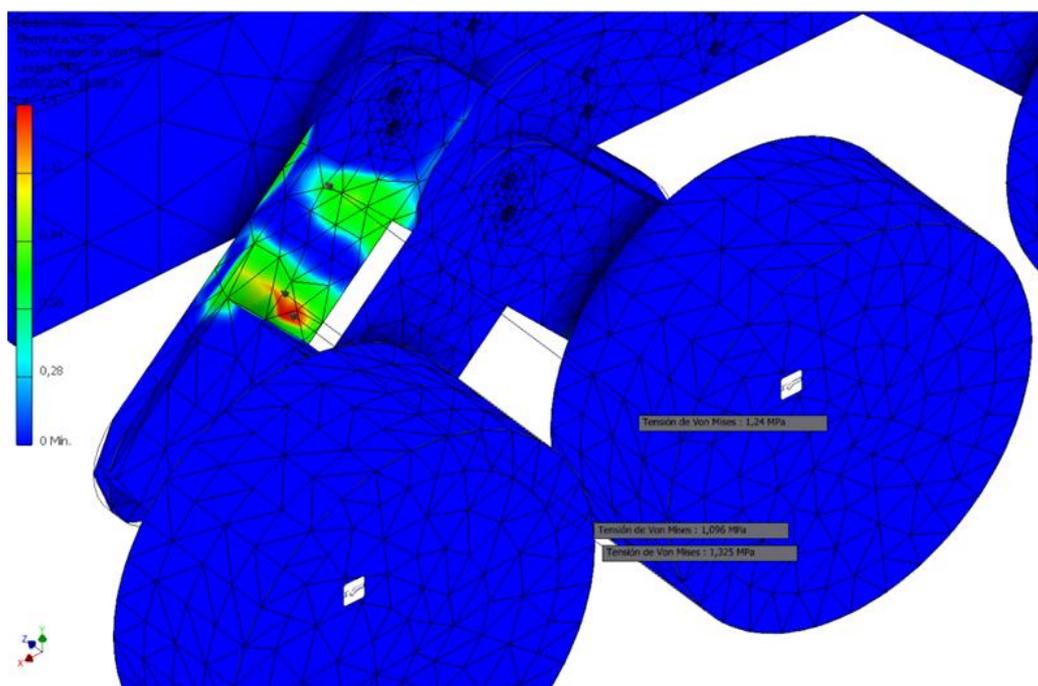
Por lo que cuando se realice el análisis de tensión en el modelo simplificado es necesario aplicar la fuerza del peso del robot a un ángulo θ de 60° con respecto al eje x global. Como el peso del robot se estimó en 10kg la componente horizontal del peso tiene un valor de 49.05 N y la vertical tiene un valor de - 98.1 N dando como resultado el siguiente análisis de tensión:

Figura 3.6

Análisis de esfuerzos en el chasis para robot con sistema de bogie

**Figura 3.7**

Análisis de esfuerzos en ruedas para robot con sistema de bogie



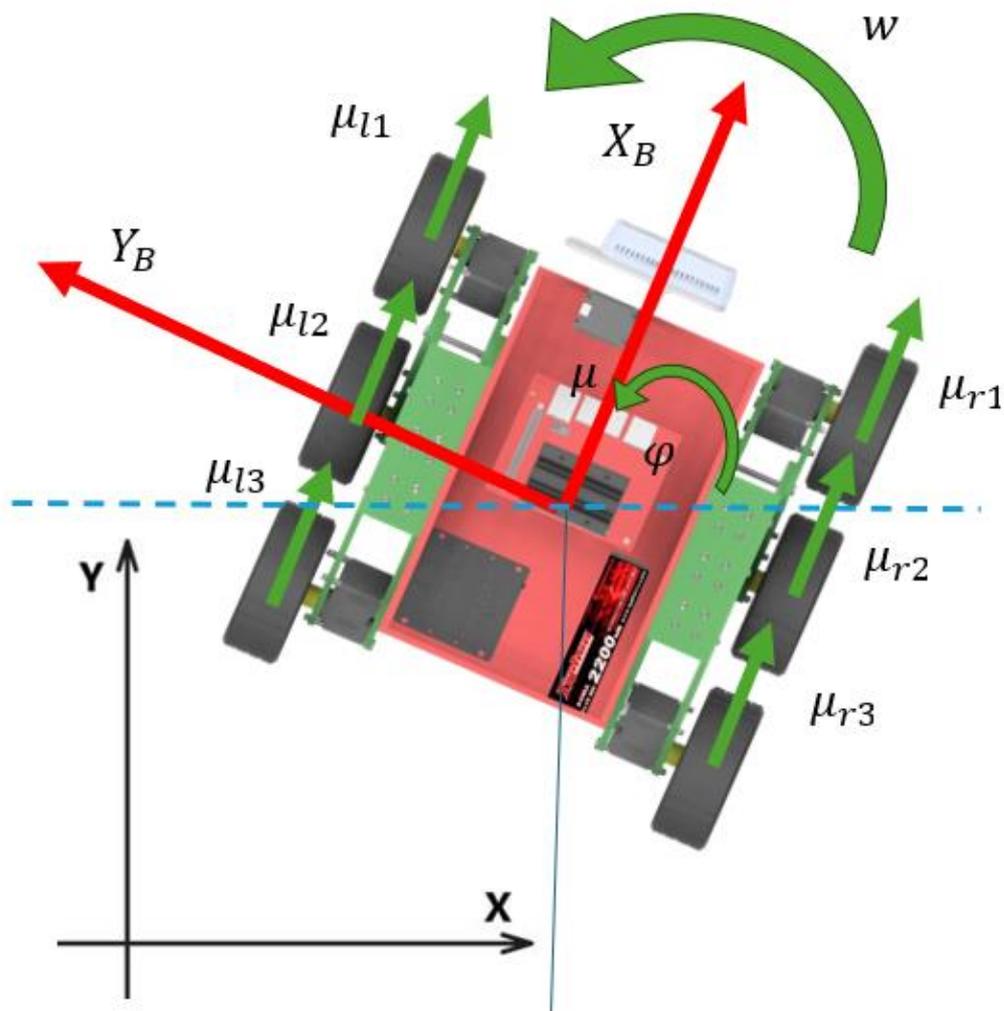
El análisis de tensión reveló que el diseño es el adecuado, pues se tuvo una tensión máxima de von Mises de 1.325 MPa. Para el PLA según la gráfica de resultados prueba de tensión para PLA, PCL y lay [20], su esfuerzo último de tensión teniendo un porcentaje de relleno de 30% es de 6.79 MPa, teniendo un factor de seguridad de 5 por lo que el robot no tendrá problemas mecánicos. Cabe recalcar que la simulación del análisis de tensión se lo realizó asumiendo que se tiene un 100% de relleno es por eso que no se observa deformación del material, sin embargo al fabricar el robot no surgirán problemas de tensión.

3.1.3 Mapeo del lugar mediante gmapping

El primer paso para que el robot pueda desplazarse en un entorno es tener un mapa del mismo, por lo que se utilizó *SLAM* o por sus siglas en inglés (Localización y mapeo simultáneos) para poder crear el mapa y mediante el paquete *gmapping* de ros melodic se genero el mapa adecuado del entorno que se encuentra en el paquete de turtlebot 3 titulado "*turtlebot3_house.world*". Además para poder localizar el robot de manera correcta fue necesaria la creación de un código que calcule la odometría del mismo, el cual estima el cambio de la posición del robot respecto del tiempo en base a las velocidades que tienen las ruedas.

Figura 3.8

Sistema local y global de la cinemática del robot



Tomando en cuenta las velocidades de las ruedas derechas e izquierdas (μ_r y μ_l), el radio de las ruedas ($r = 5 \text{ cm}$), la posición inicial del sistema de referencia global (X_B y Y_B) y la distancia entre las ruedas izquierdas y derechas ($b = 15 \text{ cm}$). Se generó el siguiente algoritmo el cual calcula la odometría del robot en base a las velocidades lineales y angulares de las ruedas.

Figura 3.9

Algoritmo de cálculo de odometría para el robot tipo bogie

Algorithm 1 Odometry Calculation

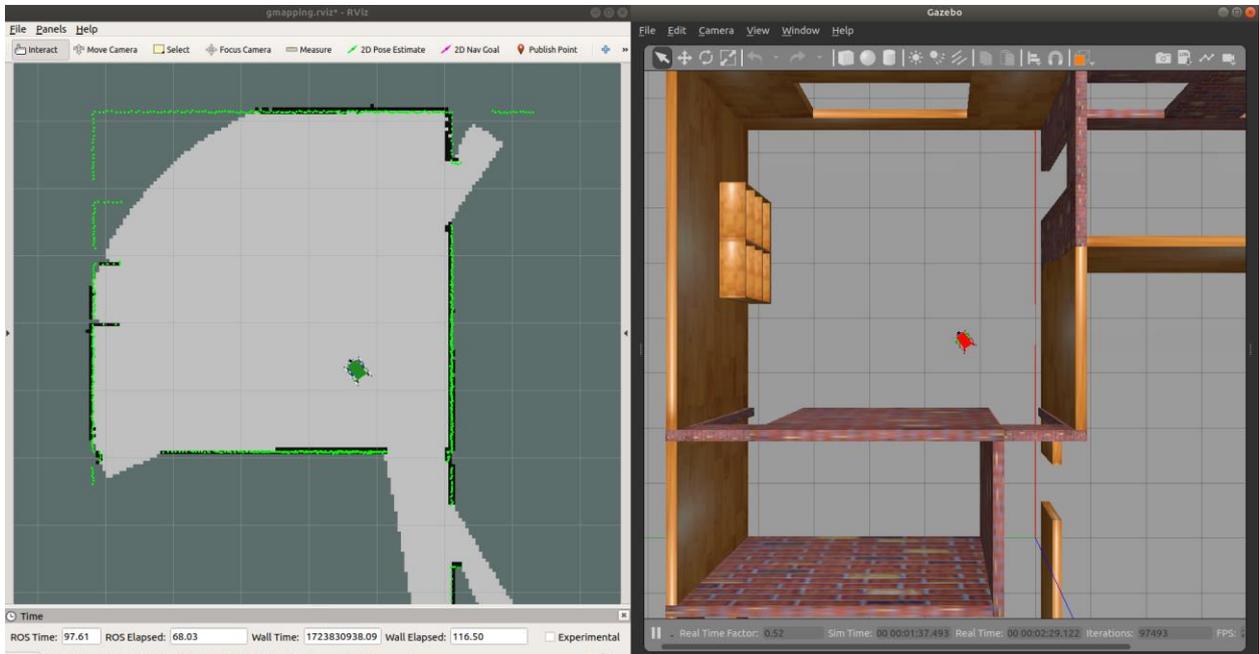
```

1:  $x, y, th \leftarrow 0.0$ 
2:  $wheel\_radius \leftarrow 0.05$ 
3:  $wheel\_base \leftarrow 0.15$ 
4:  $wheel\_velocities \leftarrow joint1: 0.0, joint2: 0.0, \dots, joint6: 0.0$ 
5: while ROS is running do
6:    $current\_time \leftarrow get\_current\_time()$ 
7:    $left\_wheel\_velocity \leftarrow$ 
    $mean(wheel\_velocities['joint1'], wheel\_velocities['joint3']$ 
8:    $, wheel\_velocities['joint5'])$ 
9:    $right\_wheel\_velocity \leftarrow$ 
    $mean(wheel\_velocities['joint2'], wheel\_velocities['joint4']$ 
10:    $, wheel\_velocities['joint6'])$ 
11:    $vx \leftarrow -1 \times (wheel\_radius \times (left\_wheel\_velocity)$ 
12:    $- right\_wheel\_velocity)$ 
13:    $vth \leftarrow (wheel\_radius \times (left\_wheel\_velocity)$ 
14:    $+ right\_wheel\_velocity) / (wheel\_base \times)$ 
15:    $dt \leftarrow current\_time - last\_time$ 
16:    $delta\_x \leftarrow vx \times \cos(th) \times dt$ 
17:    $delta\_y \leftarrow vx \times \sin(th) \times dt$ 
18:    $delta\_th \leftarrow vth \times dt$ 
19:    $x \leftarrow x + delta\_x$ 
20:    $y \leftarrow y + delta\_y$ 
21:    $th \leftarrow th + delta\_th$ 
22:    $odom\_quat \leftarrow convert\_to\_quaternion(th)$ 
23:    $broadcast\_transform(x, y, th, odom\_quat, current\_time)$ 
24:    $publish\_odometry(x, y, th, vx, vth, odom\_quat$ 
25:    $, current\_time)$ 
26:    $last\_time \leftarrow current\_time$ 
27: end while

```

Figura 3.10

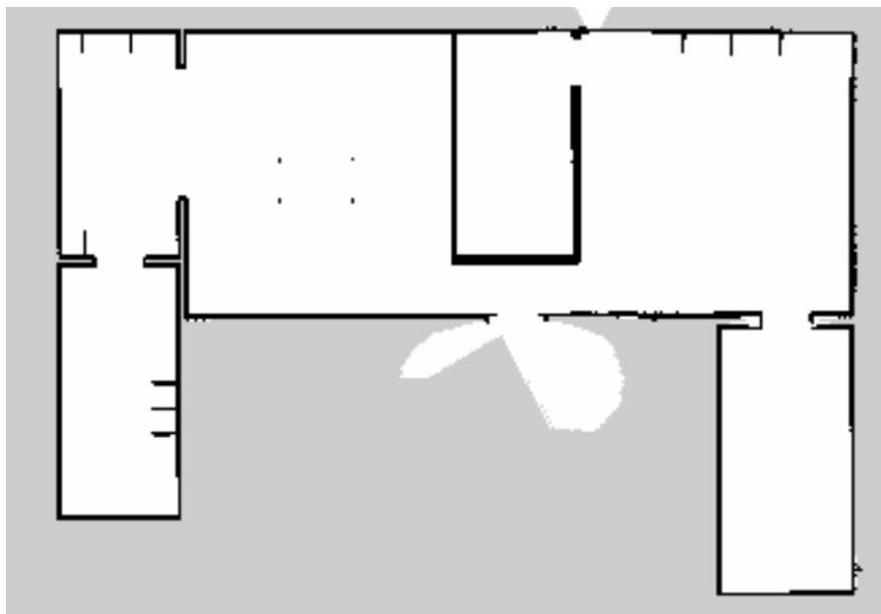
Mapeo del entorno para navegación



Después de utilizar *gmapping* para generar el mapa del entorno se procedió a guardar dicho mapa generado. Este mapa es el que se utilizó para poder realizar la navegación del robot ya que en base al mismo se puede generar la trayectoria adecuada esquivando las paredes.

Figura 3.11

Mapa del entorno generado para la navegación

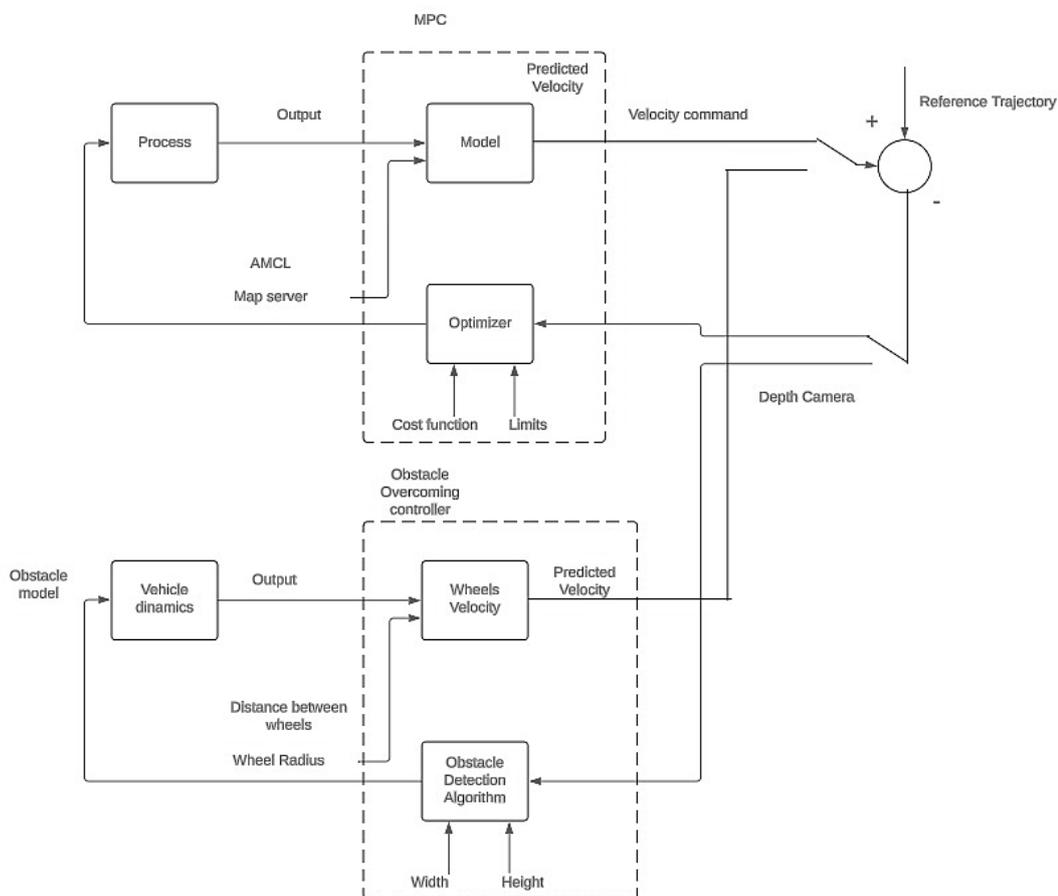


3.1.4 Navegación autónoma

Para poder realizar la navegación autónoma para superar obstáculos fue necesario entender cuál es el comportamiento de esta. Mediante *RViz* se manda un punto objetivo al que el robot debe ir y mediante el paquete *mpc_global_planner* se genera una trayectoria de referencia. Mientras la cámara de profundidad no detecte ningún obstáculo en el camino el sistema de control solo se guiará mediante las velocidades brindadas por el paquete *mpc_local_planner*, el cual es capaz de evitar obstáculos previamente mapeados. Sin embargo, cuando la cámara de profundidad detecte un obstáculo se utiliza el algoritmo de control de velocidad de ruedas en base a las dimensiones del obstáculo para superarlo sin volcamiento, y una vez superado el obstáculo se sigue usando el MPC para la navegación autónoma.

Figura 3.12

Sistema de control de la navegación autónoma



Una vez realizado el mapa se procedió a realizar la navegación autónoma. Se generó un código que suscribía la velocidad generada por el paquete *mpc_local_planner* hacia las 6 ruedas del robot, y dichas velocidades además servían para generar la odometría adecuada. Esquivando las paredes para llegar a su objetivo de manera exitosa.

Figura 3.13

Model predict control implementado en Gazebo

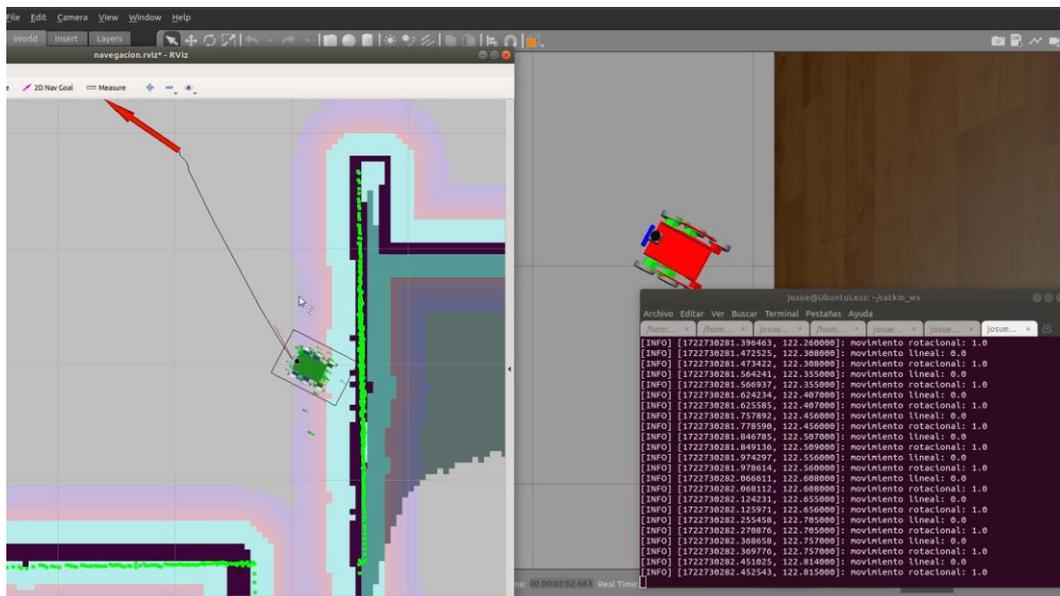
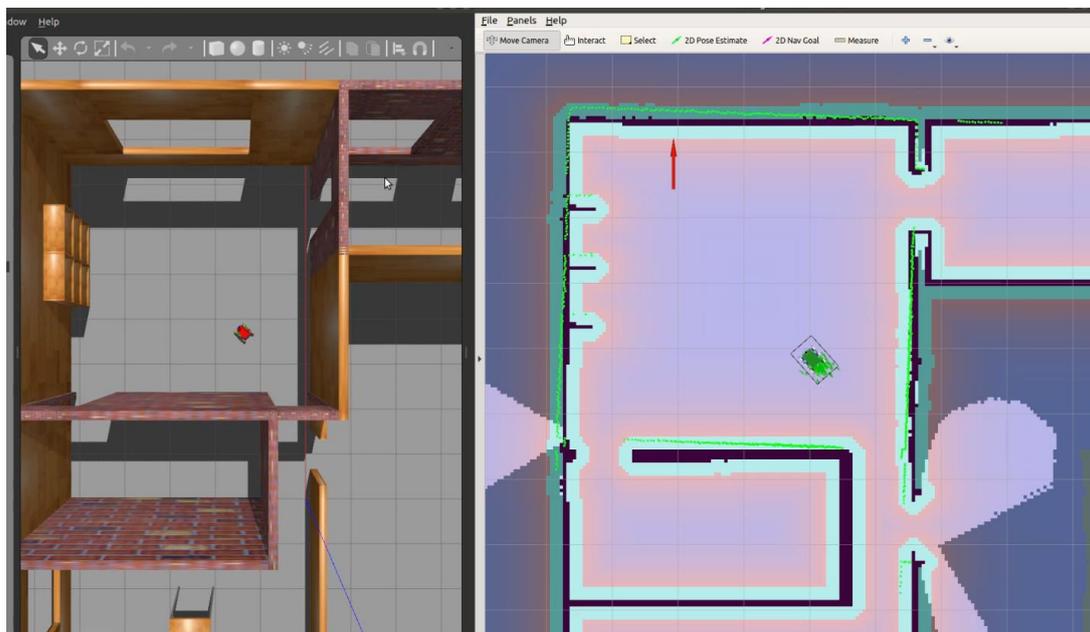


Figura 3.14

Model predict control con RViz



Nota: Imagen extraída del video de navegación

3.1.5 Detección y modelado de obstáculo escalera

Utilizando la cámara RGB-D Intel RealSense D435i se realizó la detección y el modelado de tres escaleras de diferentes dimensiones en un entorno real se realizaron con respecto a la vista desde arriba (ascenso) y desde abajo (descenso), como se ilustra en las Figuras . El sistema puede modelar automatizadamente las escaleras en tiempo real utilizando la nube de puntos para identificar los planos laterales, horizontales y verticales. A partir de estos, mediante la detección de planos horizontales (huella), se calculan el ancho, la altura y la longitud.

En este proceso, el valor real corresponde a las dimensiones físicas exactas de las escaleras, como la altura de los escalones, la longitud de cada huella y el ancho total de la escalera. Por su parte, el valor medido es el resultado de las mediciones realizadas por el algoritmo, utilizando los datos obtenidos de la cámara RGB-D. Aunque el algoritmo logra capturar estos valores con un alto grado de precisión, siempre existe un pequeño margen de error debido a las limitaciones inherentes al sistema de percepción basado en la nube de puntos.

A partir de estos, se utilizan huellas horizontales y verticales para calcular el ancho, la altura y la longitud. Gracias a refinamientos iterativos y verificaciones de consistencia, se mantuvo una alta precisión, lo que permitió eliminar mediciones inexactas. La *Tabla 3.1* contiene los valores medidos y reales, así como los errores y la precisión de las dimensiones.

Los resultados muestran que el algoritmo es efectivo en la creación precisa de modelos de escaleras con una precisión promedio del **97.4%** en la vista de ascenso y del **94.49%** en la vista de descenso.

Es importante destacar que durante las pruebas de descenso, el sistema de percepción tuvo un desempeño menor debido a la reducción de información en la nube de puntos en comparación con la vista de ascenso. Por lo tanto, el algoritmo modela las escaleras utilizando aproximaciones de las medidas de altura entre los planos horizontales.

Figura 3.15

Detección y modelado de escaleras en un entorno real



Nota: Vista en ascenso y descenso en el lado izquierdo y detección y modelado de la escalera en el lado derecho

Figura 3.16

Algoritmo que calcula las velocidades necesarias para superar el obstáculo

Algorithm 2 Control de Velocidades basado en Dimensiones de Obstáculo

```

1:  $base\_speed \leftarrow 0.8$ 
2:  $obstacle\_dimensions \leftarrow$  Esperar datos del sensor
3: while ROS is running do
4:   if  $obstacle\_dimensions$  está disponible then
5:     if  $obstacle\_dimensions.length == 3$  then ▷ Es
        escalera
6:        $height, width, depth \leftarrow$ 
         $obstacle\_dimensions$ 
7:        $is\_stairs \leftarrow$  True
8:        $is\_slope \leftarrow$  False
9:     else if  $obstacle\_dimensions.length == 2$  then
        ▷ Es pendiente
10:       $angle, width \leftarrow obstacle\_dimensions$ 
11:       $is\_stairs \leftarrow$  False
12:       $is\_slope \leftarrow$  True
13:    else
14:      Mostrar advertencia: Dimensiones no válidas
15:      continue
16:    end if
17:     $speed\_factor \leftarrow$  Calcular según dimensiones
18:     $linear\_x \leftarrow base\_speed \times speed\_factor$ 
19:    if  $is\_stairs$  then ▷ Aumentar velocidad en
        ruedas delanteras
20:       $front\_speed \leftarrow linear\_x \times 1.1$ 
21:    else
22:       $front\_speed \leftarrow linear\_x$ 
23:    end if
24:     $left\_speed, right\_speed \leftarrow linear\_x$ 
25:    Publicar velocidades a las ruedas
26:    log_info de velocidades calculadas
27:  else
28:    Mostrar advertencia: No se han recibido
29:    dimensiones de obstáculo
30:  end if
31: end while

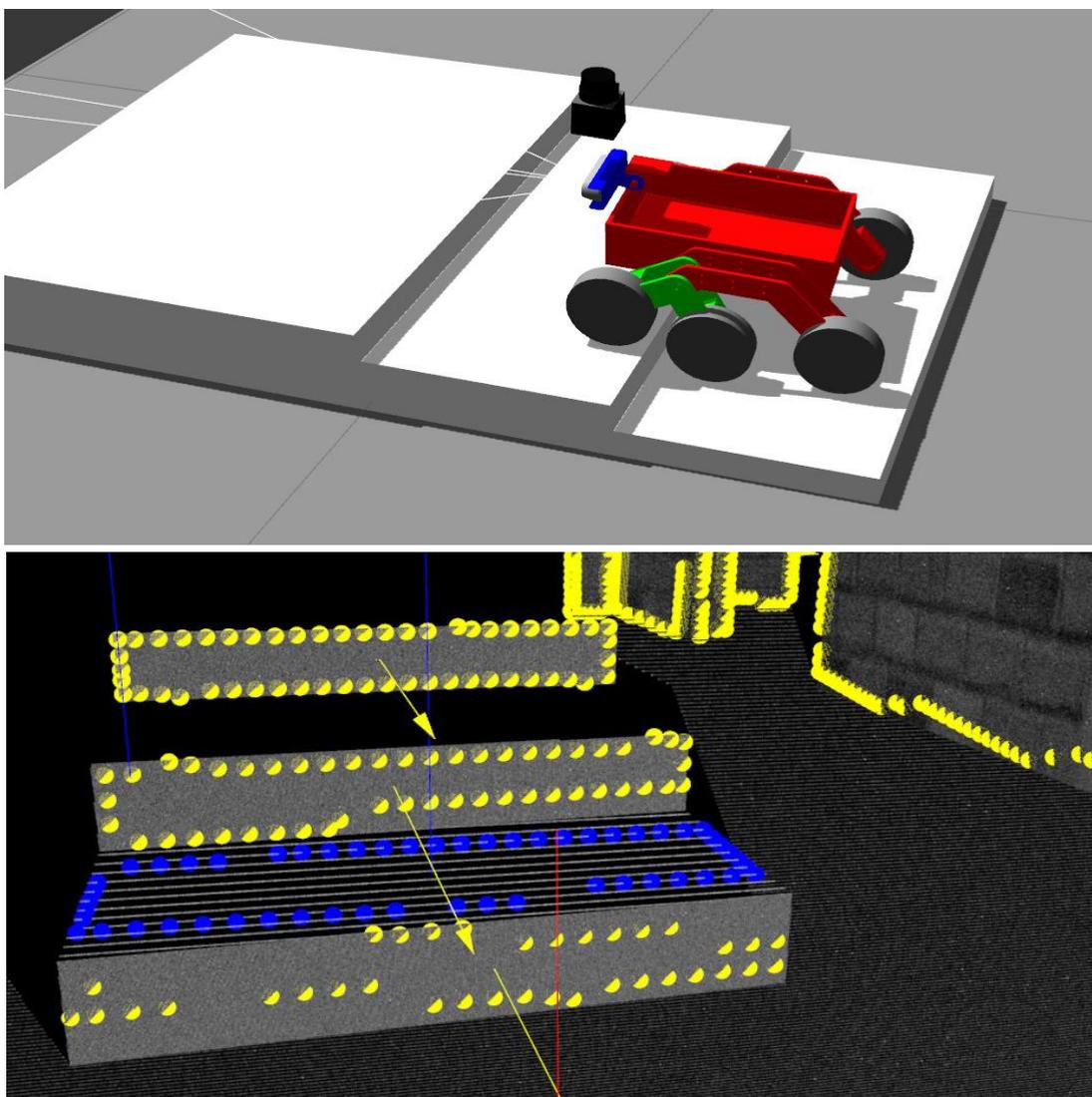
```

El algoritmo desarrollado controló las velocidades de las ruedas de un robot para enfrentar escaleras y pendientes, ajustando la velocidad según las dimensiones del obstáculo. Comenzó identificando si el obstáculo era una escalera o pendiente y evaluó si era difícil de superar, considerando la altura, profundidad y ancho para escaleras, o el ángulo de inclinación para pendientes. Cuando el obstáculo resultó complicado, la velocidad base se incrementó de

0.8 a 1.0. Además, se aplicó un factor de conversión que ajustó la velocidad en función de las dimensiones del obstáculo, asegurando que el robot se adaptara a diferentes condiciones. En el caso de las escaleras, se incrementó la velocidad de las ruedas delanteras para facilitar la subida. Finalmente, estas velocidades ajustadas se publicaron en los controladores de las ruedas, lo que permitió al robot superar los obstáculos de manera eficiente.

Figura 3.17

Robot superando el obstáculo de escalera detectado por la cámara



Nota: Navegación autónoma para superación de obstáculos en subida y bajada de escaleras

Tabla 3.1*Tabla de resultados detección y modelado de las escaleras*

Escalón	Dimensión	Real	Medido	Error	Precisión			Precisión
						<i>Ascenso</i>	<i>Descenso</i>	
Escalón 1	<i>Altura</i>	0.150	0.145	0.005	96.7%	0.143	0.007	95.3%
	<i>Longitud</i>	0.350	0.328	0.022	93.7%	0.323	0.027	92.3%
	<i>Ancho</i>	1.450	1.36	0.090	93.8%	1.32	0.130	91.0%
Escalón 2	<i>Altura</i>	0.130	0.128	0.002	98.5%	0.135	0.005	96.2%
	<i>Longitud</i>	0.350	0.343	0.007	98.0%	0.336	0.014	96.0%
	<i>Ancho</i>	1.50	1.495	0.005	99.6%	1.46	0.040	97.3%
Escalón 3	<i>Altura</i>	0.150	0.147	0.003	98.0%	0.142	0.008	94.7%
	<i>Longitud</i>	0.350	0.346	0.004	98.9%	0.334	0.016	95.4%
	<i>Ancho</i>	1.55	1.543	0.007	99.6%	1.43	0.12	92.3%

3.1.6 Detección y visualización de obstáculo rampa

La detección de rampas se realizó utilizando una cámara RGB-D Intel Relasense D435i en un entorno real y se evaluó en un entorno simulado en Gazebo. En tiempo real, se analizaron tres rampas de diferentes dimensiones desde las vistas de ascenso y descenso. El sistema permite la visualización en *RViz* de los puntos correspondientes a la detección de rampas.

Los resultados muestran una precisión promedio del **97.8%** para el ángulo en la vista de ascenso y del **99.2%** en la vista de descenso. En cuanto a la longitud, la precisión promedio es del **98.9%** en ascenso y del **99.3%** en descenso. La mayor precisión en la vista de descenso puede deberse a una mejor calidad de la nube de puntos en esa dirección como se puede observar en la *Tabla 3.2*.

Es importante destacar que el valor real hace referencia a las dimensiones físicas exactas de la rampa, tales como la longitud y el ángulo de inclinación, que son previamente conocidos y medidos en el entorno real. Por otro lado, el valor medido corresponde a las dimensiones obtenidas por el algoritmo a través de la cámara RGB-D. Si bien el sistema logra capturar estos valores con gran precisión, siempre existe un pequeño margen de error debido a las limitaciones del procesamiento de la nube de puntos o la calidad de los datos capturados. Este margen de error, en el caso del sistema propuesto, es bajo, lo que demuestra la eficacia del algoritmo al detectar y medir las pendientes con una precisión muy cercana a los valores reales.

En el entorno simulado de Gazebo, el sistema también demostró su capacidad para detectar rampas de manera efectiva, proporcionando una visualización clara y precisa de las rampas detectadas.

Este análisis destaca la eficacia del sistema para visualizar rampas con alta precisión en ambos entornos. Las diferencias en precisión entre las vistas reflejan variaciones en la calidad de los datos, siendo el descenso ligeramente más preciso. En general, el sistema ofrece una visualización confiable y precisa tanto en el entorno real como en el simulado.

Figura 3.18

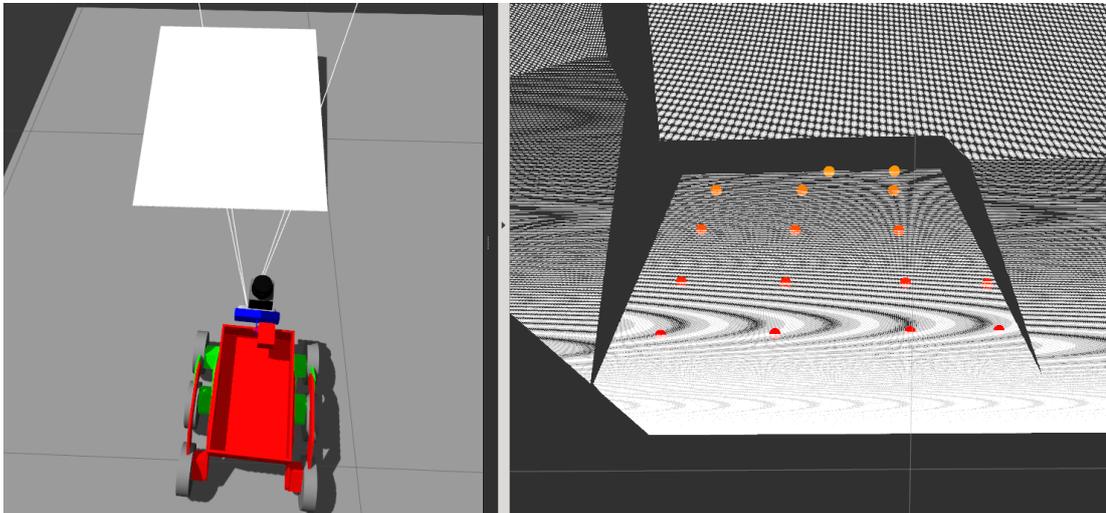
Detección de pendientes en un entorno real



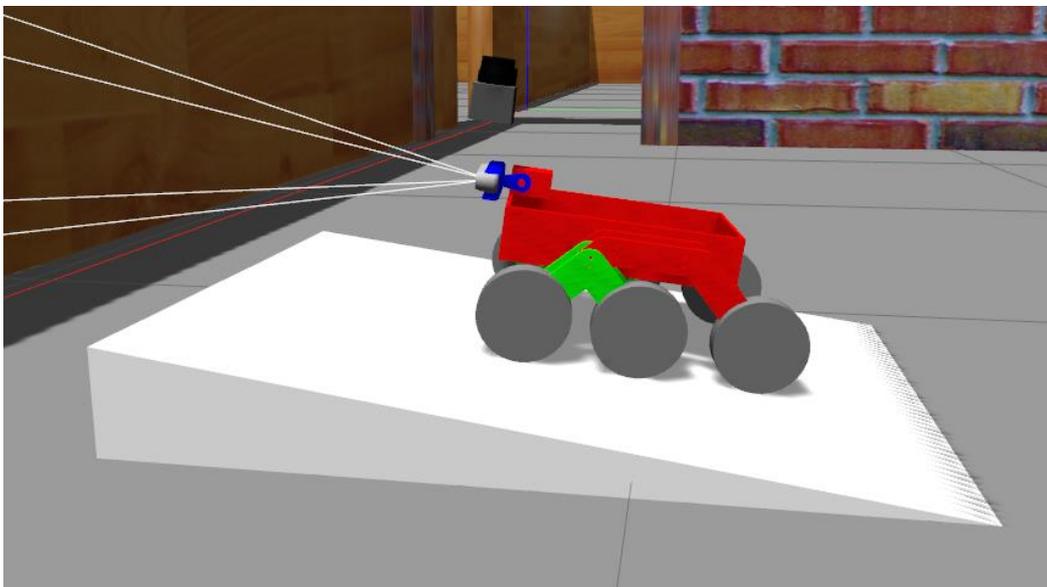
Nota: Vista en ascenso y descenso en el lado derecho y modelo de las pendientes en RViz en el lado izquierdo

Figura 3.19

Modelado de pendientes detectadas en Gazebo

**Figura 3.20**

Robot superando las pendientes detectadas por la cámara



Nota: Navegación autónoma para superación de obstáculos en subida y bajada de pendientes

Tabla 3.2*Tabla de resultados detección de pendientes*

Pendiente	Dimensión	Real	Medido	Error	Precisión	Medido	Error	Precisión
1	<i>Ángulo</i> [grados]	8°	7.92°	0.08	99.0%	7.89°	0.11	98.6%
	<i>Longitud</i> [cm]	156	153.6	2.37	98.5%	154.7	1.23	99.2%
2	<i>Ángulo</i> [grados]	15°	14.16°	0.84	94.4%	14.93°	0.07	99.5%
	<i>Longitud</i> [cm]	1400	1377.2	22.85	98.4%	1381.7	18.44	98.7%
3	<i>Ángulo</i> [grados]	12°	11.88°	0.12	99.0%	11.95°	0.05	99.6%
	<i>Longitud</i> [cm]	108	108.3	0.25	99.8%	107.9	0.07	99.9%

3.2 Análisis de costos

El análisis de costos es fundamental para comprender los recursos necesarios para el desarrollo de este proyecto. A continuación, se desglosan los costos asociados a los distintos componentes involucrados, incluyendo tanto el hardware como el software y el tiempo de programación. La *Tabla 3.3* resume el presupuesto requerido para llevar a cabo el modelado 3D, el diseño CAD, la integración de sistemas y las pruebas, entre otros aspectos. Además, se incluyen los costos de capacitación y recursos esenciales, como el hardware especializado, necesarios para asegurar el correcto funcionamiento y éxito del proyecto. A continuación, se detallan los costos asociados al proyecto:

- Modelado 3D y diseño CAD: A \$40 por hora, el costo refleja la especialización requerida para utilizar Autodesk Inventor, un software avanzado de diseño 3D que requiere habilidades técnicas avanzadas, justificando el gasto en función del nivel de experiencia del diseñador y la complejidad del modelado.
- Visión por cámara RGB-D: Con un costo de \$45 por hora, este gasto cubre la programación de algoritmos para la detección y modelado de escaleras y pendientes utilizando cámaras RGB-D, lo que exige experiencia en procesamiento de nube de puntos.
- Planeador Local (MPC): A \$45 por hora, el costo está relacionado con el desarrollo del Control Predictivo de Modelos (MPC), para el control de la velocidad de las ruedas del robot, lo que requiere conocimientos especializados en control automático y optimización de trayectorias.
- Integración y pruebas: El costo de \$45 por hora cubre la integración de sistemas en ROS y pruebas en Gazebo, asegurando que todos los componentes programables del robot funcionen en conjunto, lo cual es esencial para la funcionalidad del sistema.

- Capacitación en herramientas (ROS, Gazebo, RViz): A \$30 por hora, la capacitación en estas herramientas es crucial para que el equipo pueda utilizar efectivamente el software necesario para el proyecto, con un costo que refleja la importancia de preparar adecuadamente al equipo.
- Recursos (Laptop): Con un costo de \$2,500, esta laptop es esencial para realizar simulaciones y modelado 3D de alto rendimiento, justificando el gasto por la necesidad de un hardware potente para tareas computacionales intensivas.
- Software CAD (Licencia de Autodesk Inventor): La licencia anual de \$2,190 para Autodesk Inventor es necesaria para el diseño y modelado en 3D del robot de 6 ruedas, representando una inversión estándar en una herramienta profesional de CAD de alta calidad.

Tabla 3.3*Costes de los equipos usados*

Concepto	Categoría	Descripción	Costo Unitario	Cantidad	Costo Total
Modelado 3D y diseño CAD	Programación	Modelado 3D y diseño CAD (Autodesk Inventor)	\$ 40,00	50 horas	\$ 2.000,00
Visión por cámara RGB-D	Programación	Detección y modelado de escaleras y pendientes	\$ 45,00	100 horas	\$ 4.500,00
Control de velocidad (MPC)	Programación	Desarrollo de MPC para control de ruedas	\$ 45,00	100 horas	\$ 4.500,00
Integración y pruebas	Programación	Integración en ROS y pruebas en Gazebo	\$ 45,00	100 horas	\$ 4.500,00
Capacitación en herramientas	Capacitación	Capacitación en ROS, Gazebo, RViz y robótica	\$ 30,00	80 horas	\$ 2.400,00
Laptop con GPU potente	Hardware	Laptop con GPU potente para simulación	\$ 2.500,00	1 dispositivo	\$ 2.500,00
Software CAD (Inventor)	Software	Licencia de Autodesk Inventor	\$ 2.190,00	1 año	\$ 2.190,00
Total					\$ 22.590,00

Capítulo 4

4.1 Conclusiones y recomendaciones

En esta sección se resume la importancia del trabajo desarrollado y se presenta conclusiones a partir de los resultados obtenidos.

4.1.1 Conclusiones

- Se desarrolló un sistema de percepción en tiempo real para detectar y modelar escaleras tanto en la vida real como en Gazebo con una precisión de subida de 97% y de bajada 94% respecto a las dimensiones reales de la escalera.
- El uso de la cámara RGB-D, permitió la extracción de las nubes de puntos para detectar y modelar escaleras y rampas, facilitando la creación de modelos 3D precisos.
- La percepción de los obstáculos permitió ajustar la velocidad del robot para superar escaleras y rampas de manera estable.
- El Control Predictivo de Modelos (MPC) optimizó las trayectorias del robot asegurando una navegación estable en el entorno mapeado.
- Se diseñó el robot para que fuese capaz de superar escaleras de hasta 11 cm de altura y pendientes de hasta los 10° sin volcamiento.
- Se creó un mecanismo para asegurar la estabilidad de la cámara de profundidad.
- El algoritmo de detección y modelado de escaleras, desarrollado en colaboración con la ESPOL y la Universidad Sungkyunkwan (SKKU) de Corea, fue aprobado en la conferencia ICCAS 2024 (The International Conference on Control, Automation, and Systems).
- El algoritmo de detección implementado pudo detectar escaleras y pendientes en un rango de 3 metros.
- Se integró de manera adecuada los sensores, controladores y actuadores para la navegación autónoma.

- Se simuló mediante el uso de Gazebo y Ros Melodic el comportamiento del robot asegurando que pueda navegar de manera autónoma.

4.1.2 Recomendaciones

Debido a las limitaciones y el alcance del proyecto, se identificaron oportunidades para proyectos a futuro. A continuación, se enlistan las recomendaciones derivadas de los resultados y el análisis realizado, con la finalidad de servir como guías de futuros trabajos hacia la mejora continua.

- Construir un prototipo físico del robot utilizando materiales ligeros y resistentes que aseguren durabilidad y estabilidad en entornos reales.
- Realizar pruebas de campo iniciales en un entorno controlado para ajustar la calibración de sensores y verificar la eficacia del sistema de percepción y navegación.
- Implementar mecanismos de seguridad adicionales, como sensores de inclinación, para prevenir vuelcos y mejorar la estabilidad en terrenos más exigentes.
- Optimizar la distribución del peso y la ubicación de los componentes para modificar el centro de gravedad y garantizar una mejor estabilidad en pendientes.
- Evaluar el rendimiento del robot en diferentes condiciones climáticas para asegurar que los sensores y actuadores funcionen correctamente en situaciones de lluvia, polvo o cambios de temperatura.

Referencias

- [1] O. Antonescu, D. Valeanu, and P. Antonescu, "Mobile Robots on Adapted Wheels for Stair Climbing," in Proceedings of SYROM 2022 & ROBOTICS 2022. IISSMM 2022. Mechanisms and Machine Science, vol. 127, I. Doroftei, M. Nitulescu, D. Pislă, and E. C. Lovasz, Eds. Cham: Springer, 2023, pp. 415-426. doi: 10.1007/978-3-031-25655-4_35.
- [2] S. Woo et al., "Stair-mapping with Point-cloud Data and Stair-modeling for Quadruped Robot," *2019 16th International Conference on Ubiquitous Robots (UR)*, 2019, pp. 81-86.
- [3] H. Matsumura and C. Premachandra, "Deep-Learning-Based Stair Detection Using 3D Point Cloud Data for Preventing Walking Accidents of the Visually Impaired," *IEEE Access*, vol. 10, pp. 56249-56255, 2022.
- [4] J. Guo et al., "Approach to autonomous stair climbing for tracked robot," *2017 IEEE International Conference on Unmanned Systems (ICUS)*, 2017, pp. 182-186.
- [5] P. Ben-Tzvi, S. Ito, and A. A. Goldenberg, "A mobile robot with autonomous climbing and descending of stairs," *Robotica*, vol. 27, no. 2, pp. 171-188, 2009.
- [6] N. Pico et al., "Geometric Recognition of Diverse Terrain in Real-Time for a Six-Wheeled Robot based on Laser Scanning Sensors," *2022 22nd International Conference on Control, Automation and Systems (ICCAS)*, 2022, pp. 1924-1929.
- [7] S. Scott et al., "Enhancing Autonomous Robot Navigation Based on Deep Reinforcement Learning: Comparative Analysis of Reward Functions in Diverse Environments," *International Conference on Control, Automation and Systems*, 2023, pp. 1415-1420.

- [8] C. Chen et al., "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," *Proceedings - IEEE International Conference on Robotics and Automation*, 2019, pp. 6015-6022.
- [9] H.-C. Wang, S.-C. Huang, and I.-C. Wu, "Curriculum Reinforcement Learning From Avoiding Collisions to Navigating Among Movable Obstacles in Diverse Environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2740-2747, 2023.
- [10] A. Tahirovic, G. Magnani, and P. Rocco, "Mobile robot navigation using passivity-based MPC," in *IEEE Transactions on Control Systems Technology*, vol. 20, no. 3, pp. 681-696, May 2012, doi: 10.1109/TCST.2011.212064
- [11] Ó. F. Bits and Ó. F. Bits, "No solo Amazon se apunta a los robots de entrega: así es Roxo, el vehículo de reparto inteligente de FedEx," *20bits*, Nov. 08, 2020. [En línea]. Disponible: <https://www.20minutos.es/noticia/4465793/0/no-solo-amazon-se-apunta-a-los-robots-de-entrega-asi-es-roxo-el-vehiculo-de-reparto-inteligente-de-fedex/>
- [12] Liberman, "Innovative Stair Robot: Taking efficiency to new heights," *YouTube*. Feb. 19, 2023. [En línea]. Disponible: <https://www.youtube.com/watch?v=sHyekTDxaH8>
- [13] H. Staff, "Extreme Terrain Navigation with CrazyNK's Rocker-Bogie Model," *Hackster.io*, Aug. 30, 2024. [En línea]. Disponible: <https://www.hackster.io/news/extreme-terrain-navigation-with-crazynk-s-rocker-bogie-model-be78789c2b79>
- [14] Y. Name, "ROBOTIS e-Manual," *ROBOTIS e-Manual*. <https://emanual.robotis.com/docs/en/dxl/x/x1430-w250/>

[15] Ubuy Ecuador, "NVIDIA Jetson Nano Developer Kit - Plataforma de Ecuador | Ubuy," *Ubuy Ecuador*. [En línea]. Disponible: <https://www.ubuy.ec/es/product/35P1TJS-nvidia-jetson-nano-developer-kit-945-13450-0000-100>

[16] Intel RealSense, "Depth Camera D435i – Intel® RealSense™ depth and tracking cameras," *Intel® RealSense™ Depth and Tracking Cameras*, Jul. 03, 2024. [En línea]. Disponible: <https://www.intelrealsense.com/depth-camera-d435i/>

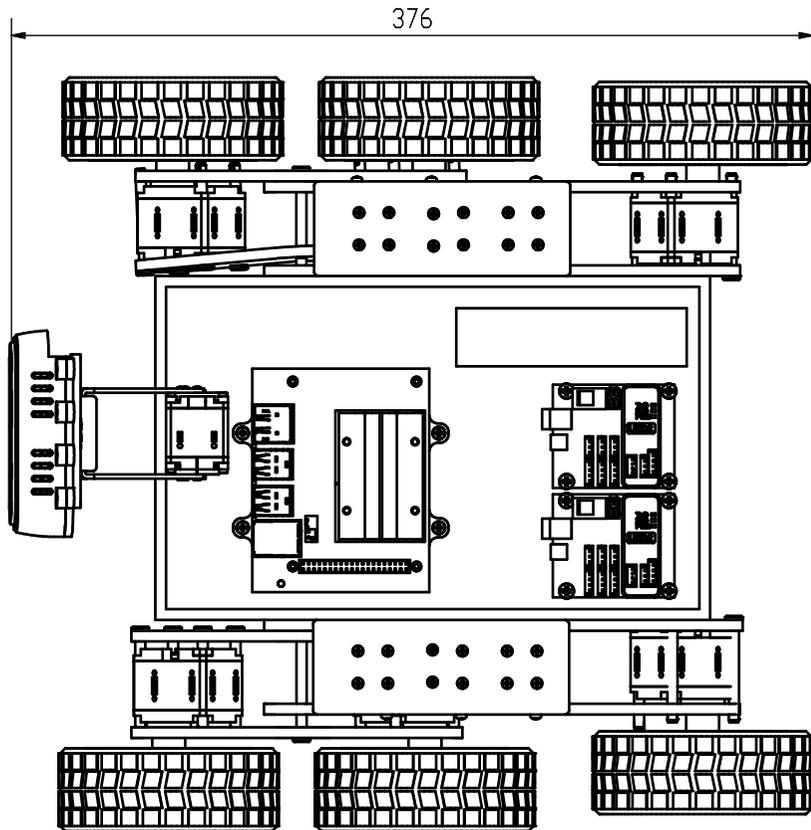
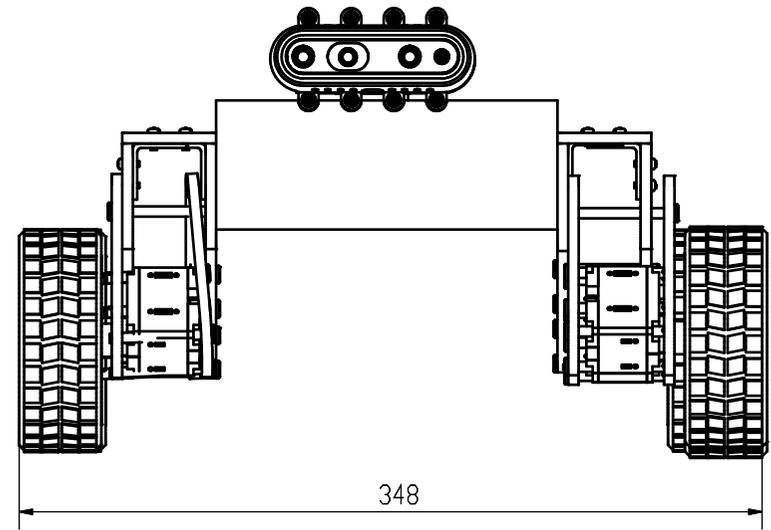
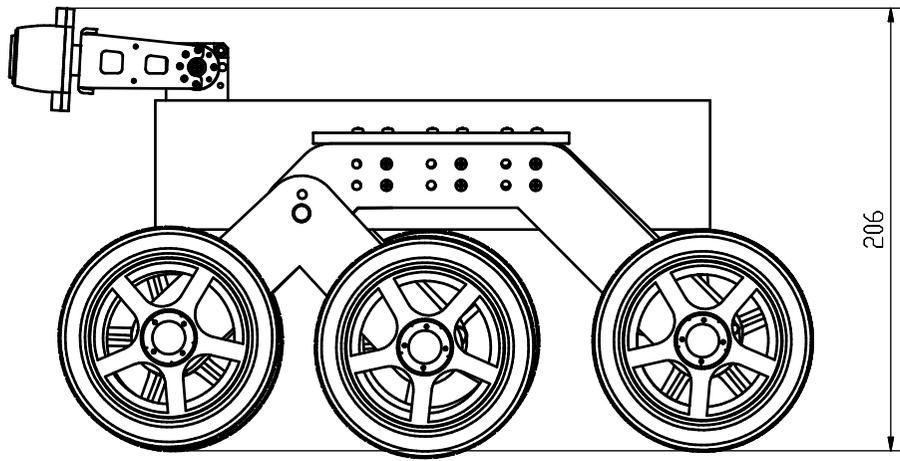
[17] P. David, "U2D2 Power Hub" *ROBOTIS e-Manual*. [En línea]. Disponible: https://emanual.robotis.com/docs/en/parts/interface/u2d2_power_hub/

[18] N. Pico, S.-H. Park, J.-S. Yi, and H. Moon, "Six-Wheel Robot Design Methodology and Emergency Control to Prevent the Robot from Falling down the Stairs," *Applied Sciences*, vol. 12, no. 9, p. 4403, Apr. 2022, doi: 10.3390/app12094403.

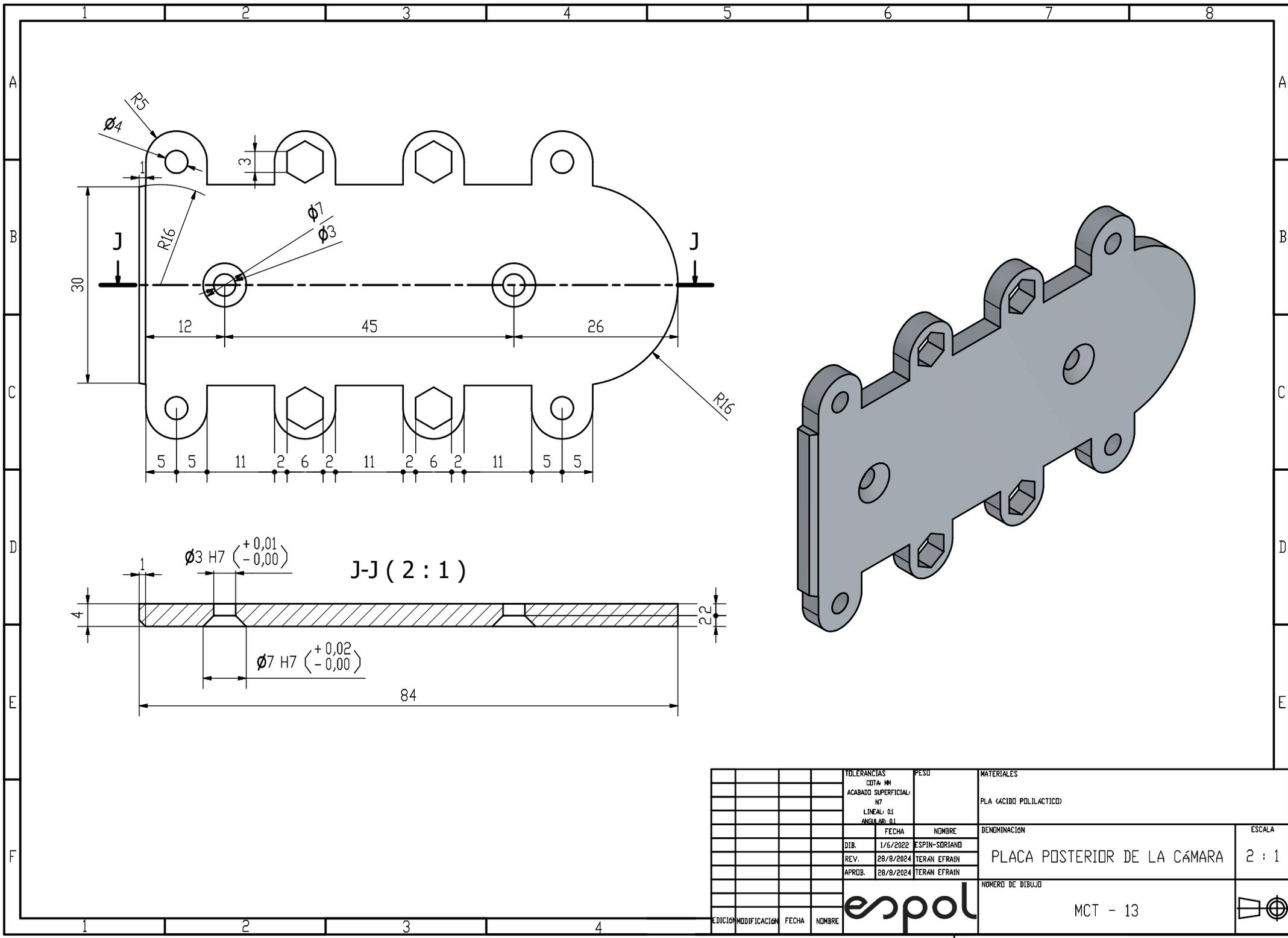
[19] "Amazon.com: Kit4Curious acrílico 6 ruedas curioso chasis para bricolaje robótica (negro): Juguetes y Juegos." <https://www.amazon.com/-/es/Kit4Curious-acr%C3%ADlico-curioso-bricolaje-rob%C3%B3tica/dp/B07D5T2DN3>

[20] J. Z. Parrado-Agudelo and C. Narváez-Tovar, "Caracterización mecánica de piezas de ácido poliláctico, policaprolactona y Lay-Fomm 40 fabricadas por modelado de deposición fundida, en función de los parámetros de impresión," *Iteckne*, vol. 16, no. 2, pp. 111-117, Dec. 2019. [En línea]. Disponible: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1692-17982019000200111&lng=en&nrm=iso.

Apéndices



TOLERANCIAS COTA: MM		PESO	MATERIALES PLA (ACIDO POLILACTICO) CAUCHO MADERA ALUMINIO 6061	ESCALA	
ACABADO SUPERFICIAL: N7					
LINEAL: 01		FECHA	NOMBRE	PERSPECTIVAS ORTAGONALES DEL ROBOT	
ANGULAR: 01		DIB.	18/5/2024		ESPIN-SORIANO
		REV.	28/8/2024		TERAN EFRAIN
		APROB.	28/8/2024		TERAN EFRAIN
		NOMERO DE DIBUJO		1 : 2.5	
		MCT - 4			
EDICION	MODIFICACION	FECHA	NOMBRE		



TOLERANCIAS		PESO	MATERIALES	
COTA: MM			PLA (ACIDO POLILACTICO)	
ACABADO SUPERFICIAL:				
N7				
LINEAL: 01				
ANGULAR: 01				
	FECHA	NOMBRE	DENOMINACION	ESCALA
DIB.	1/6/2022	ESPIN-SORIANO	PLACA POSTERIOR DE LA CÁMARA	2 : 1
REV.	28/8/2024	TERAN EFRAIN		
APROB.	28/8/2024	TERAN EFRAIN		
			NOMERO DE DIBUJO	
			MCT - 13	
EDICION	MODIFICACION	FECHA	NOMBRE	