

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA



**DISEÑO DE UN ROBOT AUTÓNOMO DE LIMPIEZA
UTILIZANDO APRENDIZAJE DE MÁQUINA**

Tesis para optar el título profesional de Ingeniero Mecatrónico

AUTOR:

Grover Hernando Susanibar Sipan

ASESOR:

Mg. Jhon Manuel Portella Delgado

Lima, Septiembre, 2024

Informe de Similitud

Yo, Jhon Manuel Portella Delgado,

docente de la Facultad de ciencias e ingeniería de la Pontificia

Universidad Católica del Perú, asesor(a) de la tesis/el trabajo de investigación titulado

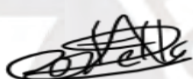
DISEÑO DE UN ROBOT AUTÓNOMO DE LIMPIEZA UTILIZANDO APRENDIZAJE DE MÁQUINA,

del/de la autor(a)/ de los(as) autores(as) Grover Hernando Susanibar Sipan,

dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 12 %. Así lo consigna el reporte de similitud emitido por el software *Turnitin* el 16/09/2024.
- He revisado con detalle dicho reporte y la Tesis o Trabajo de Suficiencia Profesional, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lugar y fecha: Lima, 16 de Septiembre del 2024

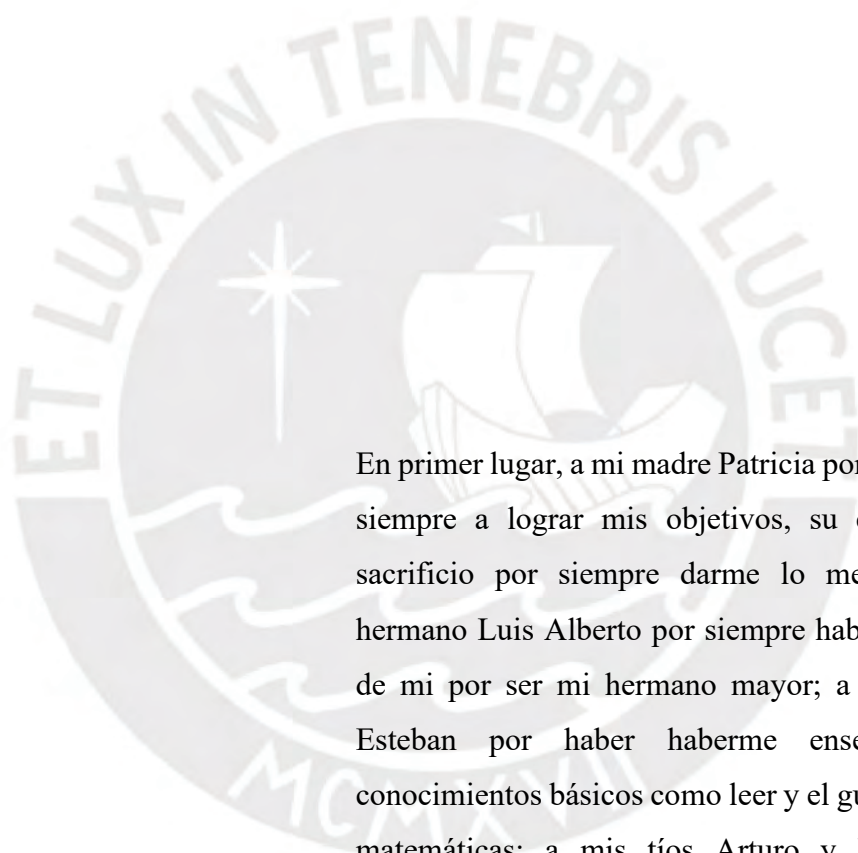
Apellidos y nombres del asesor / de la asesora: Portella Delgado, Jhon Manuel	
<u>Paterno Materno, Nombre1 Nombre 2</u>	
DNI:47099341	Firma 
ORCID: https://orcid.org/0000-0003-2778-686X	

RESUMEN

En la actualidad existen varias marcas de robots limpiadores en el mercado, sin embargo, estos robots no realizan su misión de forma eficiente, debido a que presentan errores sistemáticos al momento de desplazarse en un ambiente requerido, por lo que no logran limpiar adecuadamente la suciedad. Se realizó un estudio de los motivos por los cuales los robots actuales presentan problemas y se determinó que los algoritmos de control que usan para desplazarse sobre una habitación no son del todo eficientes, por lo que es necesario plantear otros algoritmos que complementen y así se puedan disminuir los errores.

La presente tesis es desarrollada con el objetivo de diseñar un robot de limpieza que sea capaz de desplazarse de forma autónoma sobre todos los espacios de una habitación y limpie de forma eficiente todas las partículas de suciedad presentes en el interior de dicha habitación. Asimismo, se propone el uso de un algoritmo de Aprendizaje de Máquina a través del entrenamiento de una Red Neuronal Convolutiva que le permite al robot aprender a reconocer objetos y partes de la habitación para que en su recorrido de limpieza se pueda guiar y le sirva como retroalimentación en su propósito de localizarse.

En el documento se detalla el diseño mecatrónico del robot limpiador describiendo los componentes mecánicos y de desplazamiento del robot; los sensores, actuadores y circuitos de alimentación de energía; así como la arquitectura de control y diagramas de flujo de los algoritmos involucrados. Con el objetivo de validar el funcionamiento del diseño de robot planteado, se realizaron simulaciones computacionales del recorrido y limpieza de un robot dentro de una habitación requerida. Con los resultados obtenidos se concluye que es posible mejorar el desempeño de los robots actuales a través del uso de algoritmos de Inteligencia Artificial que le permiten al robot ir aprendiendo a reconocer objetos a lo largo de su desplazamiento. Finalmente, el robot propuesto tiene un costo estimado del sistema mecánico y electrónico de S/. 5'610 y un costo de diseño de S/. 5'625 para el proyecto, teniendo en cuenta que el costo de diseño va a disminuir conforme el robot sea fabricado en grandes volúmenes de producción.



En primer lugar, a mi madre Patricia por apoyarme siempre a lograr mis objetivos, su esfuerzo y sacrificio por siempre darme lo mejor; a mi hermano Luis Alberto por siempre haber cuidado de mi por ser mi hermano mayor; a mi abuelo Esteban por haber haberme enseñado los conocimientos básicos como leer y el gusto por las matemáticas; a mis tíos Arturo y Edgar por siempre haberme apoyado desde pequeño; a mi asesor Jhon Portella por su apoyo total en el desarrollo de la presente tesis; finalmente a mis amigos de la universidad con los cuales nos llevamos los mejores recuerdos de esta época universitaria.

TEMA DE TESIS PARA OPTAR
EL TÍTULO DE INGENIERO MECATRÓNICO

Título : DISEÑO DE UN ROBOT AUTÓNOMO DE LIMPIEZA
UTILIZANDO APRENDIZAJE DE MÁQUINA

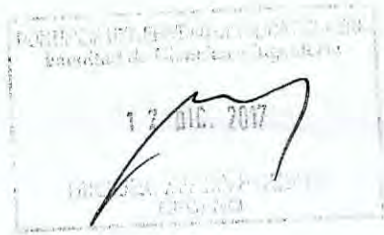
Asesor : Jhon Portella Delgado

Alumno : Grover Hernando Susanibar Sipan

Código : 20123105

Tema N° : # 214

Fecha : 24/11/2017



Descripción y Objetivos

Hace aproximadamente tres décadas se comenzó a desarrollar robots limpiadores y poco a poco han ido apareciendo en los hogares de las personas como también en la industria. Estos robots limpiadores surgen debido a la necesidad de automatizar el proceso de limpieza de una habitación. En la actualidad, existe una amplia diversidad de robots limpiadores disponibles en el mercado. Sin embargo, estos robots presentan algunos problemas que les impiden cumplir su función de forma eficiente. Estos robots necesitan navegar de forma autónoma en la habitación y justo aquí radican los problemas. Principalmente presentan problemas para percibir adecuadamente su entorno y al momento de planear su trayectoria de limpieza. Al no poder navegar correctamente en la habitación, entonces no pueden limpiarla de forma eficiente.

En este contexto, la presente tesis desarrolla el diseño de un robot autónomo de limpieza que cumple dos funciones: navegar de forma autónoma en una habitación y limpiar eficientemente todas las partículas de suciedad presentes en el interior de dicha habitación. El robot tiene dimensiones de 360x360x145mm, un peso de 2Kg y se desplaza fácilmente mediante una configuración diferencial de dos ruedas. Cuenta con escobillas, conductos de aire, un filtro de polvo, un almacén y un trapo para la limpieza de partículas de suciedad. El robot es capaz de construir un mapa geométrico 2D de la habitación y localizarse dentro de la habitación. Para limpiar la habitación, el sistema divide el mapa en zonas y va limpiando zona por zona. Finalmente, se diseñó y entrenó una Red Neuronal Convolucional, método de Aprendizaje de Máquina, que le permite al robot reconocer objetos o partes de la habitación para que pueda localizarse de acuerdo con estas referencias y además le sirva como retroalimentación para que corrija los errores que tenga y de esta forma se optimice la navegación autónoma del robot limpiador.

Máximo: 100 páginas

Luisa P. Madrid R.

i

Portella

B

INDICE DE CONTENIDO

RESUMEN	i
INDICE DE TABLAS	x
GLOSARIO DE TÉRMINOS	xii
INTRODUCCIÓN	1
1 ANTECEDENTES Y ESTADO DEL ARTE	3
1.1 Presentación de la problemática de robots limpiadores.....	3
1.2 Antecedentes y estado de tecnología de robots limpiadores	5
1.2.1 Robots limpiadores en el mercado	5
1.2.2 Publicaciones científicas sobre robots limpiadores	8
1.2.3 Publicaciones científicas sobre sistemas de navegación	9
1.2.4 Patentes.....	13
1.3 Formulación de propuesta de robot limpiador.....	15
1.3.1 Objetivo general	15
1.3.2 Objetivos específicos.....	15
2 MARCO TEÓRICO	17
2.1 Análisis cinemático	17
2.2 Localización y mapeamiento simultáneo	20
2.2.1 Localización	20
2.2.2 Mapeamiento	21
2.3 Planeamiento de trayectorias	23
2.4 Control de movimiento.....	24
2.5 Aprendizaje profundo	25
2.5.1 Redes Neuronales Convolucionales	25
2.5.2 Detección de objetos.....	29
3 DISEÑO CONCEPTUAL	33
3.1 Requerimientos y alcances	33
3.1.1 Requerimiento general.....	33
3.1.2 Requerimientos mecánicos	33
3.1.3 Requerimientos eléctricos y electrónicos	34
3.1.4 Requerimientos de control.....	35
3.2 Estructura de funciones	36
3.2.1 Caja negra.....	36
3.2.2 Funciones.....	37
3.2.3 Diagrama de estructura de funciones	40

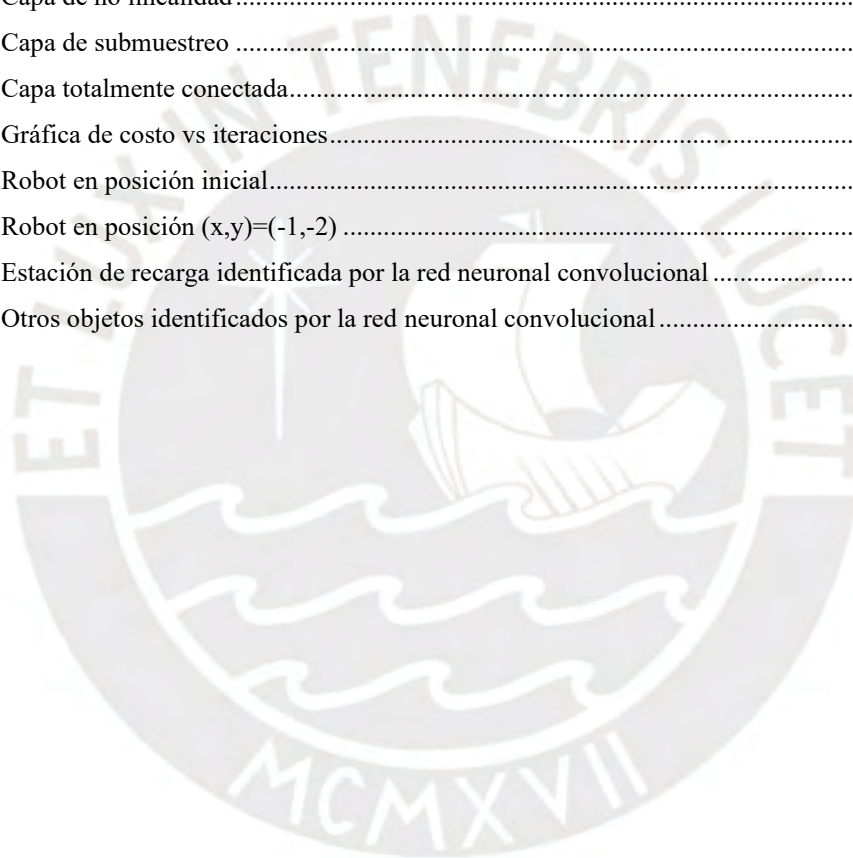
4 INGENIERÍA DE DETALLE DEL SISTEMA MECATRÓNICO.....	41
4.1 Presentación del robot autónomo de limpieza.....	41
4.2 Sistema mecánico	44
4.2.1 Carcasa exterior	45
4.2.2 Subsistema de limpieza	46
4.3 Sistema electrónico.....	53
4.3.1 Diagrama de bloques	53
4.3.2 Componentes eléctricos y electrónicos	55
4.4 Sistema de control	75
4.4.1 Arquitectura de control.....	75
4.4.2 Diagramas de flujo	76
4.4.3 Estrategias de control	83
5 SIMULACIONES Y RESULTADOS.....	84
5.1 Simulación de localización y mapeamiento simultáneo.....	84
5.2 Simulación de planeamiento de trayectoria usando descomposición celular.....	87
5.3 Simulación de aprendizaje profundo	89
5.3.1 Datos utilizados en el sistema.....	89
5.3.2 Arquitectura de red neuronal convolucional	90
5.3.3 Entrenamiento de la red neuronal convolucional	92
5.3.4 Desempeño de la red neuronal convolucional entrenada	93
5.3.5 Aplicación de la red neuronal convolucional en la navegación autónoma....	94
6 PLANOS Y COSTOS	96
6.1 Lista de planos.....	96
6.1.1 Planos mecánicos.....	96
6.1.2 Planos eléctricos y diagramas esquemáticos	97
6.2 Costos	98
6.2.1 Costos mecánicos	98
6.2.2 Costos de componentes eléctricos y electrónicos.....	99
6.2.3 Costo total.....	100
CONCLUSIONES Y RECOMENDACIONES	101
BIBLIOGRAFÍA	103

INDICE DE FIGURAS

Figura 1.1 <i>Roomba 980</i>	6
Figura 1.2: <i>Neato Botvac Connected</i>	6
Figura 1.3: <i>Samsung POWERbot VR9000</i>	7
Figura 1.4: Robot limpiador con sensor Kinect.....	8
Figura 1.5: Áreas diferenciadas por el algoritmo	9
Figura 1.6: Esquema del sistema de mapeo y navegación.....	10
Figura 1.7: Arquitectura de red neuronal convolucional utilizada	11
Figura 1.8: Simulación de un móvil basado en sistema neuro-difuso	12
Figura 1.9: Robot de limpieza de pisos	13
Figura 1.10: Robot autónomo de limpieza de pisos.....	14
Figura 1.11: Objetivo de diseño del robot autónomo de limpieza	16
Figura 2.1: Mecanismos de locomoción basados en ruedas	17
Figura 2.2: Modelo de mecanismo diferencial	18
Figura 2.3: Esquema en base a velocidades de traslación y angular.....	19
Figura 2.4: Modelamiento de mapa de celdas y lectura del sensor	22
Figura 2.5: Robot sensando distancias en el mapa	23
Figura 2.6: Esquema de planeamiento de trayectoria	24
Figura 2.7: Esquema de control de movimiento	24
Figura 2.8: Modelo de una red neuronal común y de una red neuronal convolucional	26
Figura 2.9: Ejemplo la estructura de una red neuronal convolucional.....	26
Figura 2.10: Esquema de capa convolucional	28
Figura 2.11: Aplicación de función <i>Softmax</i> en la salida	29
Figura 2.12: Tipos de tareas: (a) Clasificación, (b) Localización, (c) Detección de objetos	29
Figura 2.13: <i>Sliding Window Object Detection</i>	30
Figura 2.14: Implementación Convolutiva de <i>Sliding Window</i>	30
Figura 2.15: Etiquetas para entrenar una CNN de detección de objetos.....	31
Figura 2.16: Cómo entrenar una CNN de detección de objetos	31
Figura 2.17: <i>YOLO</i> algoritmo.....	32
Figura 3.1: Caja negra general.....	36
Figura 3.2: Caja negra dominio eléctrico	36
Figura 3.3: Caja negra dominio electrónico	36
Figura 3.4: Caja negra dominio de control	37
Figura 3.5: Caja negra dominio mecánico	37
Figura 3.6: Estructura de funciones	40
Figura 4.1: Diseño de robot autónomo propuesto.....	42
Figura 4.2: Esquema de limpieza del robot	42
Figura 4.3: Robot recargando energía	43
Figura 4.4: Vista frontal del robot	44

Figura 4.5: Vista inferior del robot.....	44
Figura 4.6: Carcasa exterior	45
Figura 4.7: Escobilla lateral.....	46
Figura 4.8: Esquema de escobilla central	47
Figura 4.9: Conducto de entrada	48
Figura 4.10: Almacén de polvo	49
Figura 4.11: Filtro de polvo.....	49
Figura 4.12: Conducto de salida.....	50
Figura 4.13: Trapo de pulido.....	50
Figura 4.14: Subsistema de desplazamiento	51
Figura 4.15: Soporte con componentes electrónicos	52
Figura 4.16: Estación de recarga	52
Figura 4.17: Diagrama de bloques del sistema.....	54
Figura 4.18: Raspberry Pi 3.....	56
Figura 4.19: Arduino Mega 2560	57
Figura 4.20: Sensor HCSR 04	58
Figura 4.21: Sensor FC-51	59
Figura 4.22: Sensor MPU 6050	60
Figura 4.23: Sensor RP LIDAR A1.....	61
Figura 4.24: Cámara Logitech C920	62
Figura 4.25: Cámara USBFHD01M.....	63
Figura 4.26: Motor Pololu 37Dx70L.....	63
Figura 4.27: Curva de funcionamiento Motor Pololu 37Dx70L.....	64
Figura 4.28: Motor DFRobot 6v.....	65
Figura 4.29: Ventilador BCB	66
Figura 4.30: Curva Presión vs Caudal	66
Figura 4.31: Controlador VNH 5019.....	67
Figura 4.32: Dual L298 H-Bridge	70
Figura 4.33: Sistema de alimentación.....	70
Figura 4.34: Batería Li-Ion 18650.....	71
Figura 4.35: Fuente DC Tenergy	72
Figura 4.36: Regulador LM2596.....	73
Figura 4.37: Sensor de voltaje FZ0430	74
Figura 4.38: Arquitectura de control del sistema.....	75
Figura 4.39: Diagrama de flujo general del sistema	77
Figura 4.40: Diagrama de flujo de proceso de construcción de mapa	79
Figura 4.41: Diagrama de flujo de proceso de localización en el mapa	78
Figura 4.42: Diagrama de flujo de proceso de planeación de trayectoria.....	80
Figura 4.43: Diagrama de flujo de proceso de limpieza de la habitación.....	81
Figura 4.44: Diagrama de flujo de proceso de recarga de batería	82

Figura 4.45: Lazo de control de motores de desplazamiento.....	83
Figura 5.1: Inicio de proceso de localización y mapeamiento simultáneo	85
Figura 5.2: Proceso de construcción de mapa de habitación	85
Figura 5.3: Culminación de proceso de construcción de mapa	86
Figura 5.4: Esquema de partículas en localización.....	86
Figura 5.5: Descomposición de mapa en zonas.....	87
Figura 5.6: Desplazamiento en forma de zigzag	88
Figura 5.7: Planeamiento de trayectoria	88
Figura 5.8: Imágenes del sistema	89
Figura 5.9: Arquitectura de red neuronal convolucional	90
Figura 5.10: Capa convolucional.....	90
Figura 5.11: Capa de no linealidad.....	91
Figura 5.12: Capa de submuestreo	91
Figura 5.13: Capa totalmente conectada.....	92
Figura 5.14: Gráfica de costo vs iteraciones.....	93
Figura 5.15: Robot en posición inicial.....	94
Figura 5.16: Robot en posición $(x,y)=(-1,-2)$	94
Figura 5.17: Estación de recarga identificada por la red neuronal convolucional	95
Figura 5.18: Otros objetos identificados por la red neuronal convolucional	95



INDICE DE TABLAS

Tabla 1.1: Especificaciones de <i>Roomba 980</i>	6
Tabla 1.2: Especificaciones de <i>Neato Botvac</i>	6
Tabla 1.3: Especificaciones de <i>POWERbot</i>	7
Tabla 1.4: Comparativa entre sistemas de navegación	12
Tabla 4.1: Parámetros de selección de controlador principal	55
Tabla 4.2: Características de Raspberry Pi 3	56
Tabla 4.3: Parámetros de selección de controlador esclavo	56
Tabla 4.4: Características de Arduino Mega 2560	57
Tabla 4.5: Parámetros de selección de sensor ultrasonido	57
Tabla 4.6: Características de sensor HCSR 04	58
Tabla 4.7: Parámetros de selección de sensor infrarrojo	58
Tabla 4.8: Características de sensor FC-51	59
Tabla 4.9: Parámetros de selección de sensor inercial	59
Tabla 4.10: Características de sensor MPU 6050	60
Tabla 4.11: Parámetros de selección de sensor LIDAR	60
Tabla 4.12: Características de sensor RP LIDAR A1	61
Tabla 4.13: Parámetros de selección de cámara matricial	61
Tabla 4.14: Características de cámara Logitech C920	62
Tabla 4.15: Parámetros de selección de cámara odometría visual	62
Tabla 4.16: Características de cámara USBFHD01M	63
Tabla 4.17: Parámetros de selección de motor de desplazamiento	63
Tabla 4.18: Características de motor Pololu	63
Tabla 4.19: Parámetros de selección de motores de limpieza	64
Tabla 4.20: Características de motor <i>DFRobot</i>	65
Tabla 4.21: Parámetros de selección de ventilador	65
Tabla 4.22: Características de ventilador	66
Tabla 4.23: Parámetros de selección de control de motores de desplazamiento	67
Tabla 4.24: Características de VNH 5019	67
Tabla 4.25: Parámetros de selección de controlador de ventilador	68
Tabla 4.26: Variación de señal PWM vs velocidad	68
Tabla 4.27: Parámetros de selección de control de motores de limpieza	69
Tabla 4.28: Características de <i>Dual L298 H-Bridge</i>	70
Tabla 4.29: Parámetros de selección de batería	70
Tabla 4.30: Características de batería Li-Ion 18650	71
Tabla 4.31: Parámetros de selección de Fuente DC	71
Tabla 4.32: Características de Fuente DC Tenenergy	72
Tabla 4.33: Componentes que requieren regulador de voltaje	72
Tabla 4.34: Características de regulador LM2596	73

Tabla 4.35: Parámetros de selección de sensor de voltaje	73
Tabla 4.36: Características de sensor de voltaje	74
Tabla 5.1: Desempeño de la red entrenada	93
Tabla 6.1: Lista de planos mecánicos	96
Tabla 6.2: Lista de planos eléctricos	97
Tabla 6.3: Lista de diagramas esquemáticos.	97
Tabla 6.4: Costos mecánicos	98
Tabla 6.5: Costos eléctricos y electrónicos.....	99
Tabla 6.6: Costo total del sistema.....	100



GLOSARIO DE TÉRMINOS

ANN	<i>Artificial Neural Network</i> (Red Neuronal Artificial)
CNN	<i>Convolutional Neural Network</i> (Red Neuronal Convolutacional)
FPS	<i>Frames per Second</i> (Imágenes por segundo)
GND	<i>Ground</i> (Tierra)
HEPA	<i>High Efficiency Particulate Air</i> (Filtro de Aire de Alta Eficiencia)
IMU	<i>Inertial Motion Unit</i> (Unidad de Movimiento Inercial)
LED	<i>Light Emitting Diode</i> (Diodo Emisor de Luz)
LIDAR	<i>Light Detection and Ranging</i> (Medición y detección por luz)
PID	<i>Proportional Integral Derivate</i> (Proporcional Integral Derivativo)
PWM	<i>Pulse Width Modulation</i> (Modulación por Ancho de Pulso)
ROS	<i>Robot Operating System</i> (Sistema Operativo de Robots)
RPM	<i>Revolutions per Minute</i> (Revoluciones por Minuto)
SLAM	<i>Simultaneous Localization and Mapping</i> (Localización y Mapeamiento Simultáneo)
SPI	<i>Serial Peripheral Interface</i> (Interfaz de Periféricos Serial)
USB	<i>Universal Serial Bus</i> (Bus universal en serie)
VAC	<i>Voltage Analog Current</i> (Voltaje de Corriente Alterna)
VDC	<i>Voltage Direct Current</i> (Voltaje de Corriente Continua)
VIN	<i>Voltage Input</i> (Voltaje de Entrada)
VOUT	<i>Voltage Output</i> (Voltaje de Salida)

INTRODUCCIÓN

Hace aproximadamente tres décadas se comenzaron a desarrollar robots limpiadores y poco a poco han ido apareciendo en los hogares de las personas como también en la industria. Estos robots limpiadores surgen de la necesidad de automatizar el proceso de limpieza de habitaciones. Mantener los ambientes limpios dentro de un hogar resulta ser una tarea que requiere tiempo. Aquellas personas que puedan presentar alguna discapacidad motora o alguna alergia al polvo les resulta muy complicado limpiar. Asimismo, contar con robots limpiadores en la industria, puede resultar en una reducción significativa de costos de operaciones.

En la actualidad, existe una amplia diversidad de robots limpiadores disponibles en el mercado. Sin embargo, estos robots aún presentan problemas que les impiden cumplir su función de forma eficiente. Estos robots requieren navegar de forma autónoma en las habitaciones y justo aquí radican algunos problemas. Principalmente presentan problemas para percibir adecuadamente su entorno y para planificar su trayectoria de limpieza. Al no poder navegar correctamente en la habitación, entonces no pueden limpiarla de forma eficiente.

En este contexto, la presente tesis desarrolla el diseño de un robot autónomo de limpieza que cumple principalmente dos funciones: navegar autónomamente en una habitación requerida y limpiar eficientemente todas las partículas de suciedad presentes en el interior de dicha habitación. El robot tiene dimensiones de 360 mm x 360 mm x 145 mm, un peso de 2kg y se desplaza fácilmente mediante una configuración diferencial de dos ruedas. Cuenta con escobillas, conductos de aire, filtro de polvo, un almacén y un trapo para la limpieza de partículas de suciedad. Asimismo, posee una estación de auto recarga, una batería y reguladores de voltaje que suministran energía y le da capacidad al robot de limpiar de forma continua durante una hora y media. El robot es capaz de construir un mapa geométrico 2D de la habitación y de localizarse dentro de la habitación. Para recorrer la habitación, el sistema divide el mapa en zonas y va limpiando zona por zona. Finalmente, como propuesta de mejora en la localización del robot, se diseñó y entrenó una Red Neuronal Convolutiva, método de Aprendizaje de Máquina, que le permite al robot reconocer objetos o partes de la habitación para que también pueda localizarse de

acuerdo con estas referencias y además le sirva como retroalimentación para que así corrija los errores acumulados que se tengan durante su recorrido de limpieza.

El desarrollo del presente trabajo se basa en la metodología basada en la norma alemana VDI 2206 (Metodología de diseño de sistemas mecatrónicos) cuyo procedimiento de trabajo contempla las fases de comprensión de la problemática planteada, elaboración de conceptos de solución, desarrollo de proyecto y elaboración de detalles. El diseño del sistema mecatrónico contempla el diseño de la parte mecánica del robot, el diseño eléctrico y electrónico, el diseño del sistema de control, así como la integración de todas las partes para el funcionamiento óptimo del robot. Finalizado el diseño mecatrónico del robot, se procederá a realizar simulaciones computacionales de los procesos que el robot limpiador realiza en su misión de navegar autónomamente en una habitación y limpiar las partículas de suciedad presentes en dicha habitación.

Este documento de tesis se encuentra estructurado en seis capítulos. En primer lugar, se describe la problemática actual de los robots limpiadores, así como el estado del arte. Luego se presenta un marco teórico que contempla los conocimientos necesarios en el diseño del robot. Seguidamente se formulan los requerimientos y alcances del sistema para de ahí pasar a describir el diseño conceptual del robot. En el capítulo 4 se desarrolla el sistema mecatrónico del robot limpiador describiéndose los componentes mecánicos, los sensores, actuadores y esquemas eléctricos involucrados en dominio electrónico, así como también la arquitectura de control y diagramas de flujo de la lógica de control del robot. Luego se describen las simulaciones computacionales realizadas y se presentan los resultados obtenidos. En el capítulo 6 se presenta la lista de planos y se describen los costos estimados en una eventual implementación del robot. Finalmente, se describen las conclusiones y recomendaciones del presente trabajo de tesis.

CAPÍTULO 1

ANTECEDENTES Y ESTADO DEL ARTE

En este capítulo se plantea la problemática de los robots limpiadores, se exponen los antecedentes y el estado de tecnología mencionando los actuales robots limpiadores en el mercado, publicaciones científicas sobre robot limpiadores y sistemas de navegación, y también se listan patentes relacionadas. Finalmente, se formula la propuesta de robot limpiador listando los objetivos de la presente tesis.

1.1 Presentación de la problemática de robots limpiadores

A lo largo de los años, la tecnología ha ido evolucionando progresivamente y al mismo tiempo cambiando la forma de vivir de los seres humanos en diversas áreas. En particular, el avance tecnológico ha impactado positivamente en el estilo de vivir de las personas en sus hogares, buscando tener sistemas de casas inteligentes. Actualmente, en el mercado existe una gran variedad de productos para el hogar que buscan contribuir a tener casas inteligentes. Según el reporte *Smart Home Market - Global Forecast to 2022*, en el año 2015, las ventas de diversos productos, software y servicios para casas inteligentes alcanzaban una cifra de 46.97 billones de dólares y se proyecta que para el año 2022, las ventas alcancen la cifra de 121.73 billones de dólares, es decir, aproximadamente un crecimiento del 160% de ventas [*MarketsandMarkets*, 2016].

Dentro de estos productos se encuentran los robots limpiadores, productos que han entrado a las vidas de las personas desde hace unas décadas atrás. De hecho, estos robots comienzan a aparecer alrededor del año 1985 como parte de investigaciones en diversas compañías y centros de investigación. Luego entran a la fase de comercialización y rápidamente toman buena posición dentro del mercado mundial. Según la *IRF (Internacional Federation of Robotic)*, los robots limpiadores representan el 96% de las ventas de robots domésticos. Se proyecta que las ventas aumenten de los 3.6 millones de unidades que se tenía en el año 2015, a alrededor de 30 millones de unidades dentro del periodo entre los años 2016 y 2019 [*International Federation of Robotic*, 2017].

Las cifras mostradas anteriormente son muy llamativas, de las cuales se puede apreciar que los robots limpiadores están cada vez más presentes en nuestras vidas. Entonces de aquí surge dos interrogantes respecto a estos robots que son las siguientes: ¿qué es un robot limpiador? y ¿cómo surge la necesidad de automatizar el proceso de limpieza dentro de los hogares de las personas?

Con el objetivo de responder la primera interrogante y definir un robot limpiador, se puede tomar como referencia a las aspiradoras convencionales. Estas aspiradoras son dispositivos que sirven como herramientas para que una persona pueda limpiar todo el polvo y otras partículas de suciedad, generalmente presentes en el suelo, dentro del interior de su hogar. A diferencia de las aspiradoras convencionales, un robot limpiador cuenta con un sistema de navegación que le permite cumplir de forma autónoma la función de limpiar el polvo y otras partículas de extrañas dentro de un ambiente. Estos robots pueden ser usados para limpiar ambientes dentro de un hogar, como también dentro de la industria.

Con respecto a la segunda interrogante, si bien es cierto que la tarea de limpiar los ambientes puede ser realizada por las personas, existen diversos factores por los cuales surge la necesidad de automatizar este proceso de limpieza. Un factor fundamental es el tiempo. Mantener los ambientes limpios dentro de un hogar resulta ser una tarea que requiere tiempo. Actualmente, las personas disponen de tiempos marcados para realizar sus actividades y el tiempo restante la disponen para su familia o descanso. Para aquellas personas que puedan presentar alguna discapacidad motora o alguna alergia al polvo o partículas extrañas, estos robots son ideales para mantener sus hogares limpios. Otro factor importante es el costo y se aprecia mejor en la industria. Las industrias disponen de un número amplio de ambientes que pueden ser espaciosos. Contar con robots limpiadores en una industria, puede resultarle una reducción significativa de costos.

Los robots limpiadores son comercializados desde hace unas décadas y, en la actualidad, existe una amplia diversidad de robots limpiadores disponibles en el mercado. Sin embargo, estos robots aún presentan algunos problemas que les impiden cumplir su función de forma totalmente eficiente. Como se sabe, estos robots cuentan con un sistema de navegación que les da la capacidad de ser autónomos, pero es aquí donde se encuentran los principales problemas. En primer lugar, los robots limpiadores presentan ineficiencia al momento de registrar su trayectoria de limpieza. Los robots deben realizar una limpieza completa dentro de los ambientes requeridos, pero para esto deben ser capaces de

distinguir las áreas que ya han limpiado de aquellas que aún les falta cubrir. De no distinguir adecuadamente, pueden limpiar ciertas áreas de forma repetitiva, lo cual resulta en pérdidas de tiempo, ineficiencia y malgasto de energía.

En segundo lugar, los robots limpiadores presentan problemas para percibir adecuadamente su entorno. Este aspecto es de suma importancia, ya que pueden tener inconvenientes para planificar su trayectoria, como también para evitar obstáculos en su camino. Actualmente, existen algoritmos de navegación que tratan de solucionar estos problemas. Un algoritmo clásico es *SLAM* (*Simultaneous localization and mapping*); no obstante, aún no funciona adecuadamente en robots limpiadores de ambientes.

Finalmente, ante lo expuesto anteriormente, la presente tesis plantea el diseño de un robot limpiador que limpie polvo y suciedad dentro un ambiente requerido. Adicionalmente, se presentará un método de Aprendizaje de Máquina con la finalidad de mejorar la navegación autónoma del robot y de esta forma presentar una alternativa de solución a los problemas que presentan los robots limpiadores actuales.

1.2 Antecedentes y estado de tecnología de robots limpiadores

Hace aproximadamente tres décadas se comenzaron a desarrollar robots limpiadores de ambientes y poco a poco han ido apareciendo en los hogares de las personas como también en la industria. La primera generación de robots limpiadores contaba con estrategias de navegación que se basaban en movimientos aleatorios para cubrir su ambiente, por lo que no eran capaces de diferenciar áreas limpiadas de aquellas que les faltaba cubrir. Luego apareció una segunda generación, en la cual los robots contaban con sensores más modernos que le permitían percibir mejor su entorno. Además, tenían estrategias de navegación con planificación, es decir, tenían métodos para cubrir una cierta área. Posteriormente se ha ido desarrollando robots limpiadores con sensores más sofisticados y con estrategias de navegación más robustas que en conjunto buscaban mejorar el rendimiento del proceso de limpieza. A continuación, se detallarán diferentes robots limpiadores que existen en el estado de la tecnología.

1.2.1 Robots limpiadores en el mercado

1.2.1.1 iRobot Roomba 980

Este robot es la versión más novedosa de la serie de robots limpiadores de la compañía *iRobot*. Cuenta con un sistema de navegación *iAdapt 2.0* con localización visual que le

permite al robot navegar dentro un nivel entero con múltiples habitaciones de una casa manteniendo un registro de su localización. Puede realizar su recorrido de limpieza en forma de zigzag o espiral. Mediante su aplicación *iRobot Home*, le permite al usuario programar horarios de limpieza y recibir notificaciones del estado de limpieza. Posee un filtro *AeroForce* que captura partículas tan pequeñas como de 10 micras.

Tabla 1.1: Especificaciones de *Roomba 980*¹

Tiempo de autonomía	2 horas
Tiempo de recarga	3 horas
Eficiencia en pisos de alfombra	95%
Eficiencia en pisos rugosos	80%
Altura capaz de subir	0.625 pulg
Batería	Ni-MH
Precio	\$900



Figura 1.1 *Roomba 980*¹

1.2.1.2 *Neato Botvac Connected*

Es un robot de la compañía *Neato Robotics*. Cuenta con dos modos de limpieza: primero el modo Eco, para un proceso de limpieza pausado con ahorro de energía, y segundo el modo Turbo, para un proceso rápido de limpieza. Tiene un diseño en forma de letra D, que le permite limpiar muy de cerca de las paredes y tener mejor acceso a las esquinas. Posee un sistema de navegación *LaserSmart* con detección de objetos en tiempo real que le permite mapear las habitaciones y planear metódicamente su camino.

Tabla 1.2: Especificaciones de *Neato Botvac*²

Tiempo de autonomía	2 horas
Tiempo de recarga	3 horas
Eficiencia en pisos de alfombra	95%
Eficiencia en pisos rugosos	90%
Altura capaz de subir	1 pulg
Batería	Li-Ion
Precio	\$800



Figura 1.2: *Neato Botvac Connected*²

¹ iRobot Corporation: <http://www.irobot.com/For-the-Home/Vacuuming/Roomba.aspx>

² Neato Robotics: <https://www.neatorobotics.com/robot-vacuum/botvac-connected-series/>

1.2.1.3 Samsung POWERbot VR9000

Es un robot limpiador perteneciente a la compañía Samsung. Posee una performance de limpieza ligeramente superior comparado a los robots de las otras compañías; sin embargo, es difícil de usar, no es tan intuitivo como los demás. Cuenta con una cámara digital y diez sensores de profundidad que le permiten determinar el óptimo camino de limpieza mediante la creación de un mapa completo del hogar, incluyendo paredes, muebles y escaleras. No cuenta con una aplicación para control remoto.

Tabla 1.3: Especificaciones de *POWERbot*³

Tiempo de autonomía	1 hora
Tiempo de recarga	2.5 horas
Eficiencia en pisos de alfombra	95%
Eficiencia en pisos rugosos	95%
Altura capaz de subir	0.59 pulg
Batería	Li-Ion
Precio	\$1000



Figura 1.3: *Samsung POWERbot VR9000*³

Adicionalmente, en el mercado existen más modelos de robots limpiadores tales como *HOM-BOT LrV5900* de la compañía *LG* que presenta cuatro modos de limpieza que son zigzag, celda por celda, espiral y manual. Sin embargo, estos robots no logran tener un buen rendimiento de limpieza, por lo que resultan ineficientes. Por el lado de la investigación, también se trata de desarrollar estos tipos de robots con el propósito de mejorar los algoritmos de navegación.

³ Samsung Electronics Co.: <http://www.samsung.com/us/home-appliances/vacuums/powerbot-robot>

1.2.2 Publicaciones científicas sobre robots limpiadores

1.2.2.1 A Novel Navigation System for Indoor Cleaning Robot

Esta publicación fue realizada por investigadores de la Universidad de Beihang, China, en el año 2016. Se dispone de un robot diferencial circular de dos ruedas con mecanismos de rodillos, auto recarga y en la parte superior tiene un sensor Kinect. Cuenta con dos modos de trabajo: independiente y manual. En ambos modos el robot capta información visual, por medio de imágenes capturadas por el sensor Kinect, y de su posición, a través de encoders acoplados en sus ruedas. En el modo independiente, el robot transmite esta información a un computador externo por medio de comunicación inalámbrica (WLAN) para el procesamiento de imágenes y planificación de la trayectoria y luego el computador envía comandos de movimiento. En el modo manual, el usuario puede controlar el robot por medio de una aplicación en un dispositivo móvil, es decir, el usuario es quien comanda al robot para limpiar el ambiente.



Figura 1.4: Robot limpiador con sensor Kinect [Zheng, 2016]

1.2.2.2 Experience based Domestic Environment and User Adaptive Cleaning Algorithm of a Robot Cleaner

Es una publicación realizada por investigadores de la Universidad de KAIST, Korea en el año 2014. Esta investigación no se centra en el diseño de un robot limpiador, sino más bien en mostrar un panorama de algunos algoritmos de navegación que usan dichos robots. Un robot limpiador común utiliza algoritmos basados en *SLAM* para construir un mapa geométrico del ambiente donde se encuentra. Sin embargo, estos algoritmos basados en *SLAM* presentan varios problemas cuando son aplicados a ambientes reales, debido a que ciertas áreas pueden presentar muchos obstáculos y el robot puede quedar estancado. Por tal motivo, se propone un algoritmo que inspeccione el ambiente, clasifique las áreas de acuerdo con la cantidad de obstáculos que presenta y luego determine el orden y la estrategia de limpieza.



Figura 1.5: Áreas diferenciadas por el algoritmo [Han-Gyeol, 2014]

1.2.3 Publicaciones científicas sobre sistemas de navegación

1.2.3.1 A Flexible and Scalable SLAM System with Full 3D Motion Estimation

Esta publicación presenta un sistema de navegación y mapeo que integra un subsistema SLAM 2D, un subsistema de navegación 3D y un conjunto de sensores que le permite al robot desplazarse autónomamente sobre diversos tipos de escenarios. El subsistema SLAM 2D recibe datos de escaneo de un sensor LIDAR para en un primer instante crear un mapa geométrico del escenario donde se encuentra y después localizarse de acuerdo a una comparación entre las medidas del sensor y del mapa construido. Luego el sistema de navegación 3D recibe múltiples medidas de sensores como IMU, GPS, giroscopios

que junto con la estimación de la posición 2D del subsistema anterior son fusionados mediante un Filtro de Kalman Extendido para tener una estimación precisa de la posición del robot en seis grados de libertad. Luego esta información se utiliza para controlar los actuadores del robot y navegar de forma autónoma. El filtro de navegación se ejecuta en tiempo real a una frecuencia de 100 Hz. El sistema consume bajos recursos computacionales por lo que puede usarse en procesadores de bajo procesamiento y bajo costo.

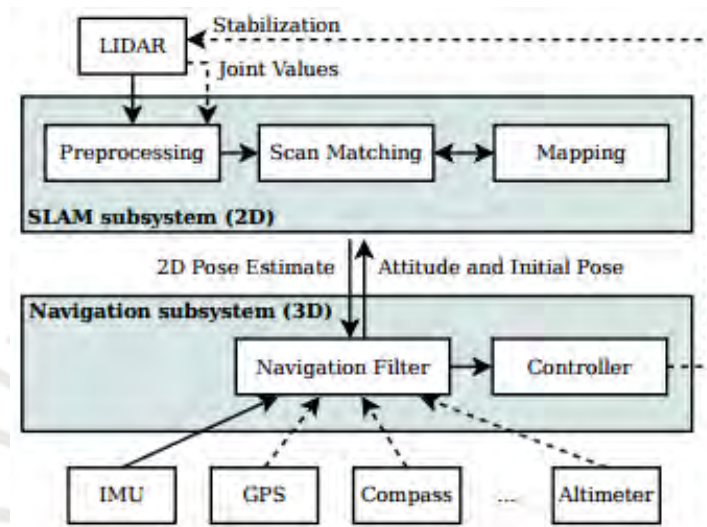


Figura 1.6: Esquema del sistema de mapeo y navegación [Kohlbrecher, 2011]

1.2.3.2 Door recognition and Deep Learning Algorithm for Visual Based Robot Navigation

En este artículo técnico, un nuevo método basado en Aprendizaje Profundo es utilizado para la navegación autónoma de robots. Se diseña un Red Neuronal Convolutiva con la finalidad de extraer información y lograr identificar escenas de ambientes como la localización de puertas. Luego esta información extraída es usada para la navegación móvil. La arquitectura de la red neuronal convolutiva está conformada por una serie de capas: una capa de entrada, capas convolucionales, capas de normalización, capas de agrupamiento y una capa de salida. Para el entrenamiento de la red neuronal se adquiere como data de entrada un set de 20500 imágenes de ambiente de tamaño 60x60 píxeles, del cual 2500 imágenes corresponden a imágenes de puertas y son etiquetadas como muestras positivas, mientras que el resto son de imágenes de ambiente y son etiquetadas como muestras negativas. Luego se prueba la red neuronal con un conjunto de 8200 imágenes y se consigue tener solo un porcentaje de error de 2.82%. A la salida de la red

neuronal convolucional se tiene información si es que existe una puerta y en qué dirección se encuentra. De esta forma, el robot navega sobre el ambiente tomando como referencia las puertas que están a su alrededor y las direcciones de estas para navegar de forma autónoma.

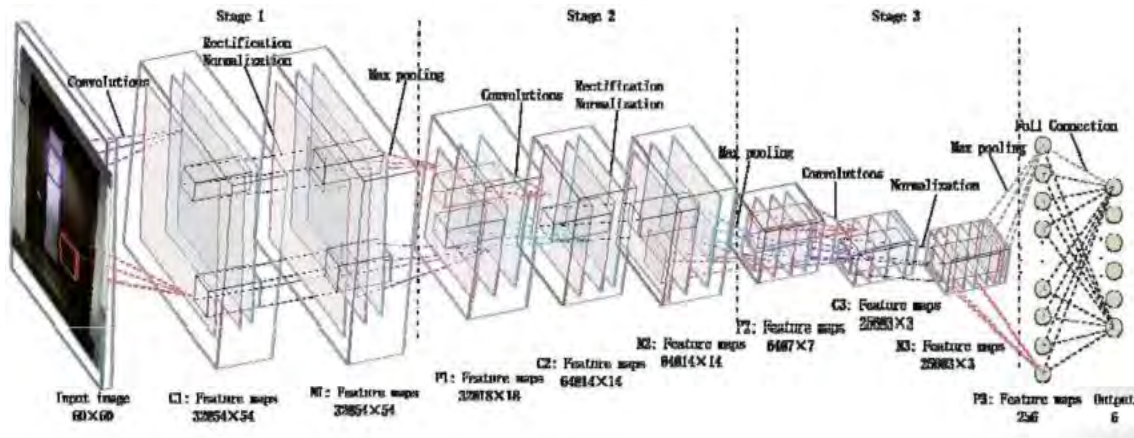


Figura 1.7: Arquitectura de red neuronal convolucional utilizada [Chen1, 2014]

1.2.3.3 Neuro-Fuzzy Based Autonomous Mobile Robot Navigation System

En este artículo técnico, un sistema basado en lógica neuro difusa es desarrollado para la navegación autónoma de un robot móvil. Se utiliza un control basado en comportamientos con el propósito de desarrollar un paradigma reactivo en la navegación. El sistema transforma las señales de los sensores del robot en señales de accionamiento de las ruedas del móvil. El sistema consiste en un sistema híbrido de dos etapas: la primera conformada por una red neuronal artificial y la segunda por lógica difusa. A la entrada de la red neuronal se tiene medidas de distancia del lado izquierdo, frontal y derecho del robot junto con su orientación. El robot recorre el ambiente evitando obstáculos y de esta forma se entrena la red neuronal. A la salida de la red neuronal se tiene un ángulo de direccionamiento de referencia que va como entrada a la siguiente etapa de lógica difusa. Los valores lingüísticos de la lógica difusa son cerca, medio y lejos y de esta forma se tienen un total de 81 reglas de entrenamiento. A la salida del sistema se obtienen velocidades de las ruedas del móvil. De esta forma se consigue tener un sistema de navegación autónoma.

1.2.4 Patentes

1.2.4.1 Robotic Floor Cleaning Device

Patente publicada en agosto de 2003 con serie US 6,605,156 B1 que presenta un dispositivo de limpieza de pisos de ambientes. Este dispositivo es un robot de dos ruedas que integra varias partes tales como un chasis, dos ruedas, un ventilador que succiona el aire, una sola escobilla central que recoge el polvo, un set de sensores y un sistema de control para la navegación dentro de un ambiente. El robot cuenta principalmente con dos modos de operación complementarios. En el primer modo, el robot se mueve sobre el ambiente con la misión de detectar los lugares donde hay polvo. Luego en el segundo modo de operación, el robot realiza la tarea de succionar el polvo del lugar anteriormente detectado.

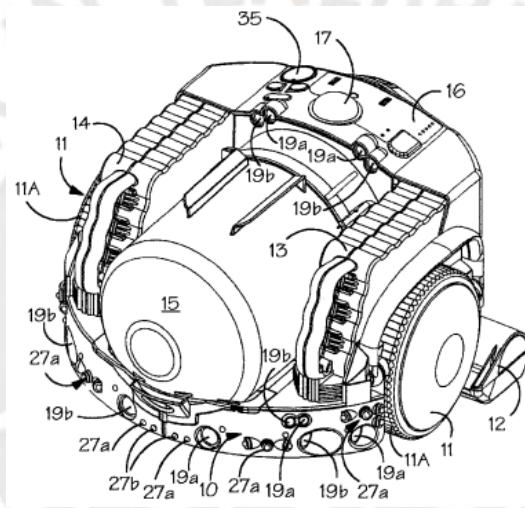


Figura 1.9: Robot de limpieza de pisos [Jones, 2011]

1.2.4.2 Autonomous Floor Cleaning Robot

Patente denominada con serie US 7,636,982 B2 que data de la fecha de diciembre de 2009. Esta patente corresponde al diseño de un robot autónomo limpiador de pisos que integra varios subsistemas con los que se basan los robots comerciales de ahora. En primer lugar, el diseño cuenta con una batería integrada al robot y que tiene la capacidad de ser recargada fácilmente por medio de una entrada a la que se conecta un cable de alimentación. Luego se discute la necesidad de optimizar los mecanismos de cepillado y absorción de polvo para lograr una sinergia de operaciones y reducir los requerimientos de energía. Para esto, el diseño contiene escobillas laterales en conjunto con la escobilla central para recolectar mejor las partículas de suciedad. Finalmente, plantea integrar diversos sensores como sensor detección de obstáculos, sensor de colisión de muros, y encoders en cada rueda con la finalidad de otorgarle al robot la capacidad de desplazarse autónomamente.

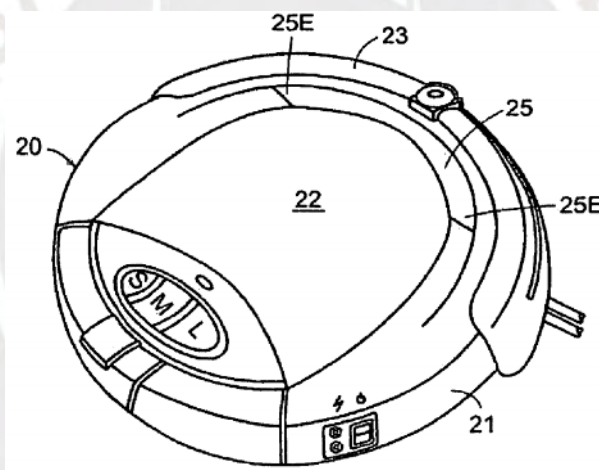


Figura 1.10: Robot autónomo de limpieza de pisos [Gerard, 2013]

1.3 Formulación de propuesta de robot limpiador

1.3.1 Objetivo general

El objetivo principal del presente trabajo de tesis es el diseño de un robot de limpieza que navegue de forma autónoma en una habitación y limpie todas las partículas de suciedad presentes en el interior de dicha habitación. El algoritmo de navegación del robot será propuesto tal que aproveche algunas técnicas de Aprendizaje de Máquina.

1.3.2 Objetivos específicos

- Diseñar mecanismos de locomoción de traslación y rotación del robot que permita su desplazamiento autónomo sobre diversos tipos de superficies.
- Diseñar mecanismos de conducción de aire, cepillado y filtros para la absorción de polvo e integrar actuadores de limpieza.
- Integrar un conjunto de sensores y actuadores que permita al robot reconocer su ambiente que lo rodea y desplazarse autónomamente sobre dicho ambiente.
- Desarrollar un sistema de energía que brinde autonomía energética durante el proceso de limpieza y tenga una estación de auto recarga.
- Desarrollar algoritmos de construcción de mapas geométricos y de localización. De esta forma el robot podrá tener un mapa del ambiente en que se encuentra y planear su recorrido de limpieza.
- Revisar y simular un método basado en aprendizaje de máquina con el objetivo de proponer una mejora al rendimiento de la navegación autónoma del robot.
- Estimar el presupuesto del sistema diseñado tomando en cuenta costos asociados al diseño e implementación del robot.

La Figura 1.11 muestra el objetivo de diseño del robot autónomo de limpieza. Se cuenta con herramientas como mecanismos de desplazamiento y limpieza, un sistema de suministro de energía y algoritmos de Aprendizaje de Máquina como mejora en la navegación para cumplir el objetivo de diseñar un robot autónomo de limpieza, el cual desarrollará las funciones principales de navegar autónomamente sobre una habitación y limpiar las partículas de suciedad.

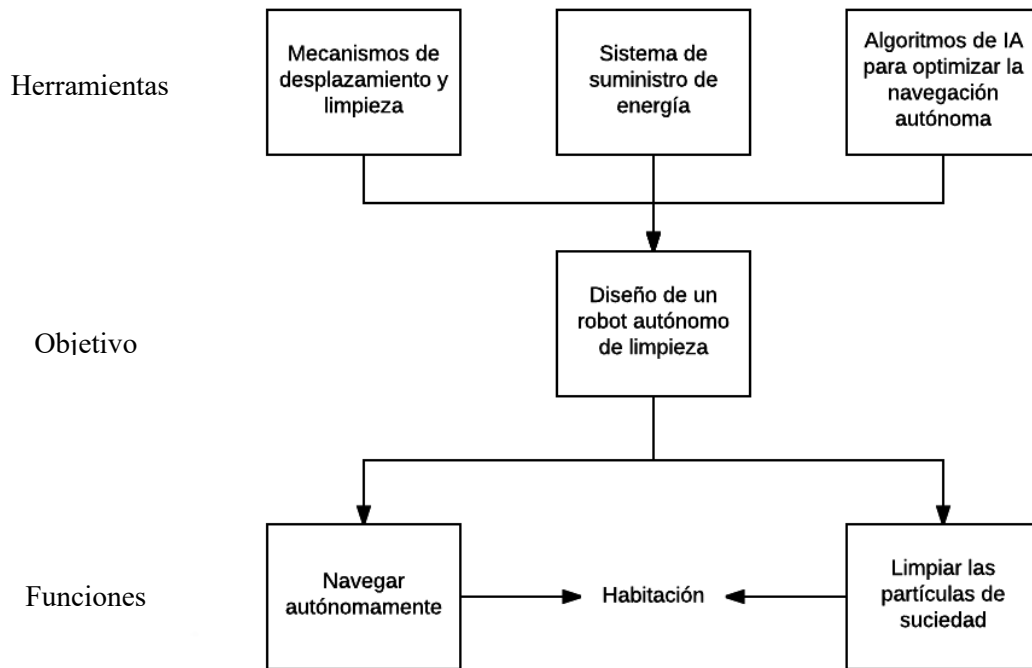
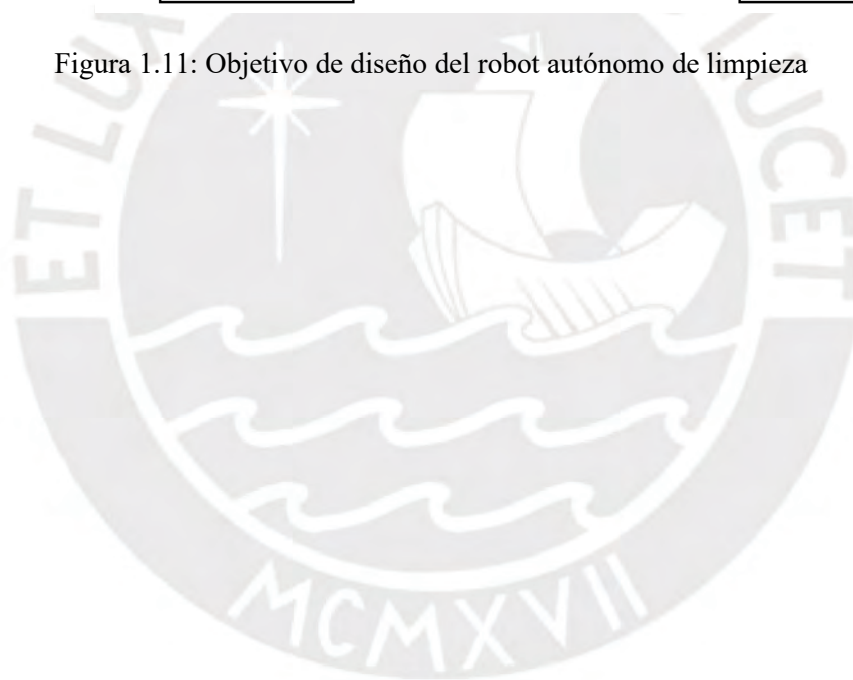


Figura 1.11: Objetivo de diseño del robot autónomo de limpieza



CAPÍTULO 2

MARCO TEÓRICO

En este capítulo se explica el marco teórico empezando por el análisis cinemático de robot móvil con configuración diferencial, seguidamente se aborda el algoritmo de localización y mapeamiento simultáneo, el proceso de planeamiento de trayectoria del robot dentro de una habitación, y el control de movimiento. Finalmente, se estudian algunos algoritmos de aprendizaje profundo como redes neuronales convolucionales y su proceso de entrenamiento para la detección de objetos.

2.1 Análisis cinemático

En el dominio de la robótica móvil en base a ruedas existen varios mecanismos de locomoción cuya elección se basa en la función que va a realizar el robot y parámetros como estabilidad, maniobrabilidad, controlabilidad, eficiencia y costo. En la Figura 2.1 se muestran algunos de los mecanismos de locomoción de robots de ruedas.

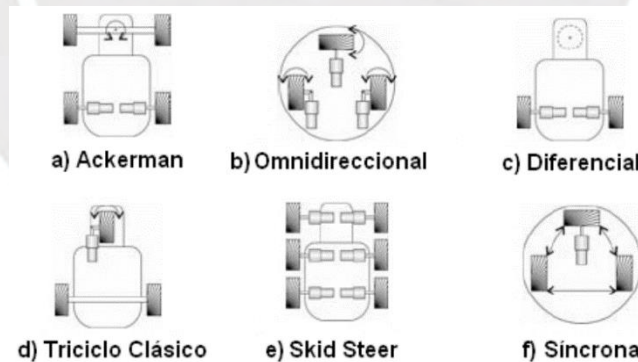


Figura 2.1: Mecanismos de locomoción basados en ruedas [Ollero, 2001]

En la presente tesis, se opta por la configuración diferencial que se basa en dos ruedas independientes montadas sobre un mismo eje de tracción debido a que es el mecanismo más simple y fácil de controlar. Con el objetivo de controlar al robot, resulta necesario modelar el comportamiento del robot y para esto se realiza un análisis cinemático del mecanismo.

Un robot con mecanismo diferencial posee dos ruedas que pueden girar a diferentes velocidades para así desplazarse. El robot ubicado en el plano tiene tres grados de libertad: posición (x,y) y orientación (θ) , por lo que la posición del robot en el plano a menudo se representa por la coordenada (x,y,θ) .

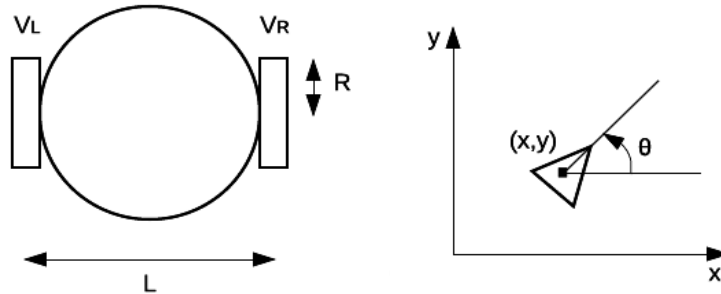


Figura 2.2: Modelo de mecanismo diferencial

En la Figura 2.2 se muestra el modelo del mecanismo diferencial, en el cual “ V_L ” representa la velocidad de giro de la rueda izquierda; “ V_R ”, la velocidad de giro de la rueda derecha; “ R ”, el radio de las ruedas y “ L ”, la distancia de separación entre las ruedas.

En el control del movimiento del robot, los parámetros de entrada son V_L y V_R ya que variando estas velocidades se modifica el movimiento del robot; por ejemplo, si ambas ruedas giran a la misma velocidad entonces el robot se mueve en línea recta, mientras que si una rueda gira a menor velocidad que la otra entonces el robot gira hacia la dirección de la rueda que gira a menor velocidad. Luego los estados del movimiento corresponden a la posición en el plano (x,y) y la dirección a la cual está apuntando (θ) . Entonces el modelamiento de este mecanismo diferencial debe relacionar los parámetros de entrada con los estados [Egerstedt, 2013]. Esta relación se aprecia en las siguientes ecuaciones de estado:

$$\dot{x} = \frac{R}{2} * (V_R + V_L) * \cos \theta \quad \dots (2.1)$$

$$\dot{y} = \frac{R}{2} * (V_R + V_L) * \sin \theta \quad \dots (2.2)$$

$$\dot{\theta} = \frac{R}{2} * (V_R - V_L) \quad \dots (2.3)$$

Estas ecuaciones expresan cómo V_L y V_R afectan a \dot{x} , \dot{y} que representan la variación de la posición en el plano y a $\dot{\theta}$ que representa la variación de la dirección del robot. No obstante, cuando el robot diferencial se está moviendo, lo más simple de expresar su

movimiento es a través de su velocidad de traslación “ v ” y su velocidad angular “ ω ” como se muestra en la siguiente figura.

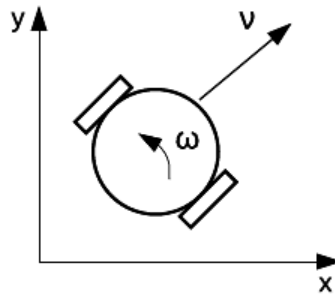


Figura 2.3: Esquema en base a velocidades de traslación y angular

Entonces ahora las entradas del sistema son v y ω , por lo que es necesario tener las relaciones entre estos parámetros y los estados. Estas relaciones se expresan mediante las siguientes ecuaciones de estado:

$$\dot{x} = v * \cos \theta \quad \dots (2.4)$$

$$\dot{y} = v * \sin \theta \quad \dots (2.5)$$

$$\dot{\theta} = \omega \quad \dots (2.6)$$

Estas ecuaciones expresan una forma más simple sobre cómo afecta la velocidad de traslación a las variaciones de la posición en el plano \dot{x}, \dot{y} , como también que la variación de la dirección del robot es igual a la velocidad angular. De los dos modelos anteriormente citados, si bien es cierto que ambos expresan la variación de los estados de la posición y direccionamiento del robot, resulta necesario diferenciarlos. El segundo modelo se utiliza en el diseño del control, ya que cuando se quiere variar los estados, el movimiento es caracterizado por las velocidades de traslación y angular. Por otro lado, el primer modelo es utilizado en la implementación debido a que las variaciones de la posición y el direccionamiento del robot son expresadas mediante las velocidades de giro de las ruedas. Luego es necesario encontrar las relaciones entre las velocidades de las ruedas y las velocidades de traslación y angular ya que ambos modelos entran en funcionamiento cuando el robot se desplaza. Estas relaciones se muestran a continuación:

$$V_R = \frac{2v + \omega L}{2R} \quad \dots (2.7)$$

$$V_L = \frac{2v - \omega L}{2R} \quad \dots (2.8)$$

Estas dos últimas relaciones muestran cómo los parámetros de diseño v y ω junto con los parámetros físicos R, L se relacionan con los parámetros de implementación V_L y V_R .

2.2 Localización y mapeamiento simultáneo

La localización y mapeamiento simultáneo (SLAM) es un proceso de mapeamiento de un entorno desconocido mientras que, al mismo tiempo, se mantiene un registro de la posición del robot móvil dentro del entorno. Este proceso es uno de los mayores desafíos en la robótica móvil para desarrollar robots autónomos capaces de operar dentro de ambientes reales. En cuanto a las partes de localización y mapeamiento, resulta necesario desarrollarlas simultáneamente ya que dependen una de la otra. En el mapeamiento es necesaria la localización debido a que las medidas de los sensores de distancia son relativas a la posición del robot, la cual se requiere que sea conocida para que las medidas sean llevadas a un sistema global de coordenadas. Por otro lado, en la localización es necesario el mapeamiento ya que se requiere reconocer algunos patrones del mapa para tener estimaciones de su posición absoluta [Siegwart, 2011].

2.2.1 Localización

La localización tiene el objetivo de determinar la posición del robot con respecto al mapa del entorno. Luego es posible diferenciar dos tipos de localización: local y global. Con respecto a la localización local, a partir de una posición inicial conocida del robot, el objetivo es que mientras el robot se vaya desplazando se calcule su posición en todo instante, y para eso se dispone de sensores internos para desarrollar un sistema de odometría. Por otro lado, en la localización global, la posición inicial del robot es desconocida y el objetivo es determinar su posición absoluta en todo instante basándose del conocimiento del mapa del entorno, por lo que debe disponer de sensores externos que le permitan medir distancias del entorno.

El desarrollo de un sistema de odometría dispone de sensores internos como encoders que miden las rotaciones de los ejes de los actuadores de las ruedas en intervalos de tiempo con la finalidad de conocer la velocidad y desplazamiento del robot. Luego el cálculo de la posición global se da a través de sensores externos como el sensor LIDAR. Mediante sus mediciones se busca encontrar patrones en el entorno (*landmarks*), que son estructuras características del entorno que pueden ser fácilmente distinguidas y son utilizadas por el robot para tomarlos como referencia y obtener su localización global en el entorno.

Considerar calcular la localización del robot teniendo solo como fuente la odometría no es confiable ya que presenta errores propios que se van acumulando, por lo que se requiere

tener también como fuente a sensores externos que permitan calcular la posición absoluta del robot y corregir los errores de la odometría.

Debido a los errores inherentes de los sensores, tanto internos como externos, no es posible conocer la posición del robot con completa certeza, por lo que se requiere el uso de métodos probabilísticos que estimen la posición dada las mediciones de los sensores. El desplazamiento del robot es generado por señales de control que le permite alcanzar una posición deseada. Entonces, suponiendo una posición inicial x_t y que el robot recibe una señal de control u_t , la localización del robot en el siguiente instante de tiempo puede ser modelada mediante una distribución de probabilidad: $p(x_{t+1}|x_t, u_t)$.

2.2.2 Mapeamiento

Un mapa es una representación del entorno y es necesario que un robot construya el mapa de su entorno para poder navegar en él. En la construcción del mapa, es necesario tener en consideración aspectos como la forma de representación del mapa, su precisión, los sensores que se utilizan y el procesamiento de las medidas de los sensores.

Existen varios tipos de mapas, entre los principales están los mapas métricos, mapas topológicos y los mapas semánticos. El tipo de mapa más utilizado en navegación de robots es el mapa métrico debido principalmente a que la localización es representada por una coordenada (x,y). Luego el proceso de mapeamiento corresponde a un problema de percepción que depende de las mediciones de los sensores de distancia y como las mediciones contienen ruido inherente, entonces se requiere de un método probabilístico de estimación. Además, dichas mediciones son relativas una posición local del robot, por lo que se requiere que sean interpretadas en un sistema de coordenadas global.

Existen varios algoritmos de mapeamiento cuya elección depende del tipo de mapa que se requiere construir. Con respecto al mapa métrico, el algoritmo más preciso es el Mapeamiento por ocupación de celdas (*Occupancy Grid Mapping*) que en esencia divide al entorno en celdas finas y define si cada celda está ocupada o libre. Para obtener información del entorno se dispone de un sensor que mide distancias. Debido a que se requiere de una buena cantidad de distancias medidas en un intervalo de tiempo, el sensor más utilizado es el sensor LIDAR. Este sensor emite rayos de láser hacia su alrededor y recibe las señales reflejadas que permiten calcular las distancias. Conforme el robot sensa distancias mientras se va desplazando, se va construyendo el mapa [Lee, 2016].

La ocupación de una celda es una variable aleatoria binaria $m_{x,y}: \{\text{libre}, \text{ocupado}\} \rightarrow \{0,1\}$, es decir, una celda tiene dos posibles estados: libre u ocupada. Entonces un mapa de ocupación de celdas es un arreglo de variables de ocupación. Luego este algoritmo utiliza un Filtro Bayesiano con el objetivo de actualizar recursivamente el mapa. En la Figura 2.4(a) se aprecia que el estado de cada celda es representado por una distribución de probabilidad $p(m_{x,y})$ y la Figura 2.4 (b) muestra una distribución de probabilidad $p(z|m_{x,y})$ de la medida del sensor dado el estado de la celda. Entonces si se tiene información previa del estado de la celda $p(m_{x,y})$, se obtiene la probabilidad posterior $p(m_{x,y}|z)$ de acuerdo con la regla de Bayes:

$$p(m_{x,y}|z) = \frac{p(z|m_{x,y}) * p(m_{x,y})}{p(z)} \quad \dots (2.9)$$

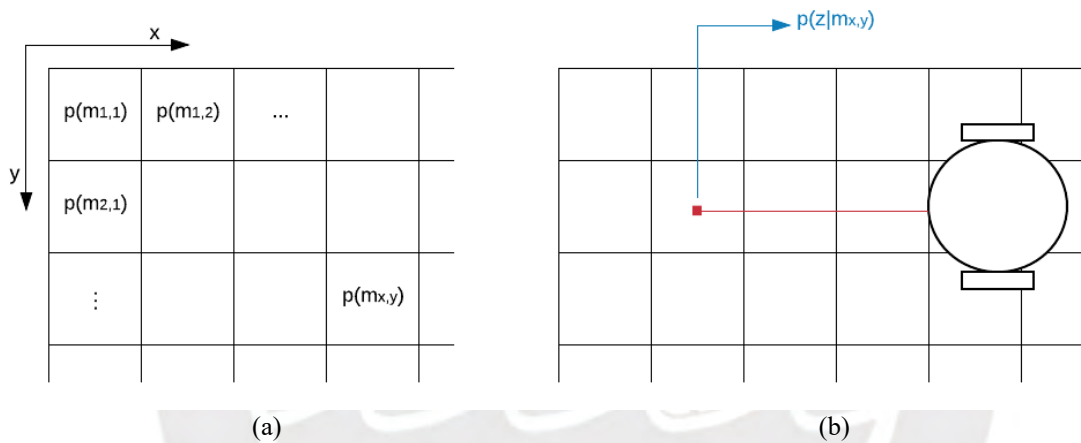


Figura 2.4: Modelamiento de mapa de celdas y lectura del sensor

La ecuación 2.9 expresa la actualización recursiva del mapa mediante el cálculo de la probabilidad posterior del valor de cada celda dada las mediciones del sensor de distancia. Sin embargo, estas mediciones son relativas a la posición actual del robot y necesitan ser transformadas hacia un sistema absoluto de coordenadas. En la Figura 2.5 se aprecia el robot ubicado en una posición (x_r, y_r) y orientación “ θ ” que mide la distancia “ d ” hacia un objeto. Entonces el valor de la celda debe asociarse a una posición dentro del sistema absoluto de coordenadas. Esta relación se muestra en la ecuación 2.10.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} * \begin{bmatrix} d \\ 0 \end{bmatrix} + \begin{bmatrix} x_r \\ y_r \end{bmatrix} \quad \dots (2.10)$$

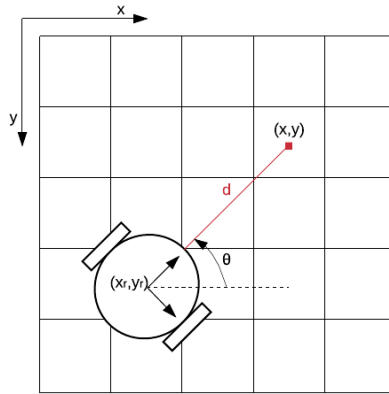


Figura 2.5: Robot sensando distancias en el mapa

2.3 Planeamiento de trayectorias

El planeamiento de trayectorias representa una de las tareas más importantes en el diseño de robots móviles autónomos. Consiste en encontrar el camino más corto para llegar desde una posición inicial hasta una posición deseada. Dicho camino más corto corresponde al camino óptimo, que al mismo tiempo evite que el robot colisione con obstáculos y que tome en cuenta las restricciones físicas del robot. Existen varios algoritmos que tratan de resolver este problema y pueden ser agrupados principalmente en dos grupos: búsqueda del camino y optimización de la trayectoria [Thrun, 2005].

En primer lugar, un algoritmo de búsqueda de camino tiene el objetivo de encontrar una trayectoria válida para llegar desde un punto inicial a un punto deseado. Este tipo de algoritmo requiere un modelo discretizado del entorno para calcular el camino. Algoritmos como Exploración rápida de árboles aleatorios (RRT) y Mapa de rutas probabilísticas básicamente lo que realizan es discretizar el entorno usando muestras aleatorias y luego conectar estas muestras evitando obstáculos. En cambio, si un mapa discretizado, como el mapa de ocupación de celdas, es provisto al algoritmo, entonces algoritmos como A* y Dijkstra solo se concentran en encontrar un camino válido. Por otro lado, un algoritmo de optimización de trayectoria recibe una trayectoria predefinida y en base a una función costo, trata de minimizar esta última con el objetivo de encontrar la trayectoria óptima.

Luego resulta necesario combinar estos dos enfoques anteriormente descritos ya que un algoritmo de búsqueda de camino retorna una trayectoria discreta válida que generalmente no es la óptima. Entonces esta trayectoria discreta es usada como entrada por un algoritmo de optimización de trayectoria y de esta forma se obtiene la trayectoria óptima como se muestra en la Figura 2.6.

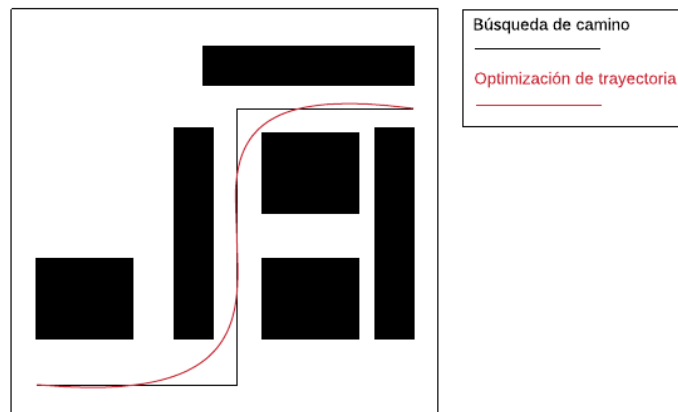


Figura 2.6: Esquema de planeamiento de trayectoria

2.4 Control de movimiento

Luego de calcular la trayectoria óptima que el robot tiene que seguir para llegar desde una posición inicial hasta una posición deseada, el siguiente problema que surge es cómo el robot realiza los movimientos necesarios para desplazarse según la trayectoria calculada. Lo que realiza el controlador es dividir la trayectoria en segmentos con formas definidas como líneas o circunferencias y posteriormente calcular las velocidades de las ruedas para que el robot se desplace. Este desplazamiento puede realizarse mediante un control lineal o un control no lineal. Desde otro enfoque, el control de movimiento puede realizarse con lazo abierto o con lazo cerrado. Entonces combinando estos dos enfoques, un control comúnmente utilizado es el control PID, control retroalimentado, que permite calcular la velocidad lineal y la velocidad angular que debe tener el robot para llegar de un punto inicial a un punto final como se muestra en la Figura 2.7. Finalmente, de acuerdo con el análisis cinemático del robot se calculan las velocidades de cada rueda.

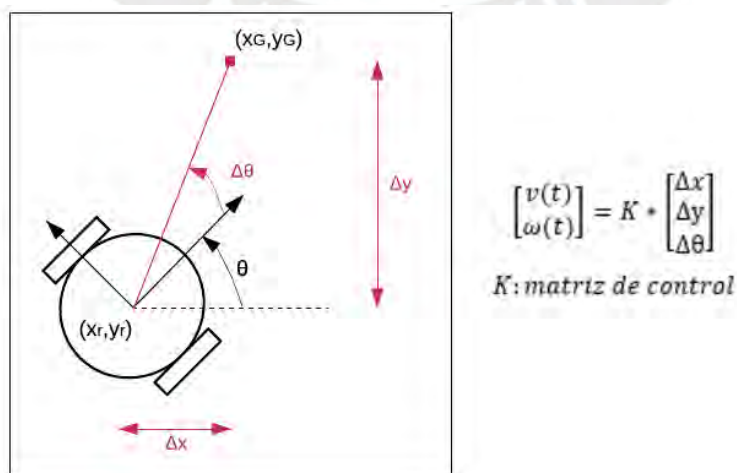


Figura 2.7: Esquema de control de movimiento

2.5 Aprendizaje profundo

Aprendizaje profundo (*Deep Learning*) es un área que pertenece al aprendizaje de máquina (*Machine Learning*). Los principios teóricos del aprendizaje profundo se basan en las redes neuronales artificiales, pero se diferencian en el grado de profundidad de las capas de neuronas, por lo que se construyen redes neuronales profundas. Al tener mayor profundidad, permite que se tengan nuevos paradigmas de aprendizaje y así tener un mayor grado de abstracción de la información. Existen varias teorías de Aprendizaje profundo que se pueden agrupar principalmente en dos grupos: aprendizaje supervisado y aprendizaje no supervisado. En particular, las Redes Neuronales Convolucionales (*Convolutional Neural Networks*) es el enfoque más utilizado en reconocimiento de patrones e imágenes debido a las numerosas ventajas que posee en comparación a los tipos de redes profundas [LeCun, 1995].

2.5.1 Redes Neuronales Convolucionales

Las redes neuronales convolucionales (CNN) son similares a las redes neuronales artificiales (RNA) ya que las principales propiedades de estas últimas se mantienen. De hecho, una red neuronal convolucional también presenta capas de neuronas con pesos sinápticos y *biases* que son modificados en un proceso de aprendizaje supervisado.

El surgimiento de las CNN se da cuando se encuentran algunas limitaciones de las redes neuronales artificiales para realizar determinadas tareas, como cuando se trabaja con imágenes o audio. Una imagen o un espectro de audio generalmente son representados mediante una larga cantidad de variables. Debido a que una red neuronal artificial tiene todos sus elementos procesadores completamente conectados, la dimensión del espacio de entrada no debe ser grande, ya que se tendría una gran cantidad de pesos a entrenar. Esto llevaría a tener problemas de ocupar una gran cantidad de memoria computacional porque se requeriría tener una gran cantidad de muestras para entrenar y adicionalmente conduciría a tener problemas de pérdida de precisión en la generalización.

Las redes neuronales convolucionales en su estructura combinan principalmente tres ideas que son las que las diferencian de las redes neuronales artificiales: uso de campos receptivos locales, compartimiento de pesos y submuestreo. Estas características son visualizadas en los tipos de capas que conforman una CNN. A diferencia de las RNA, las capas de una CNN están dispuestas en tres dimensiones: ancho, altura y profundidad, similar a la estructura de una imagen, en la cual la profundidad corresponde al número de canales de colores.

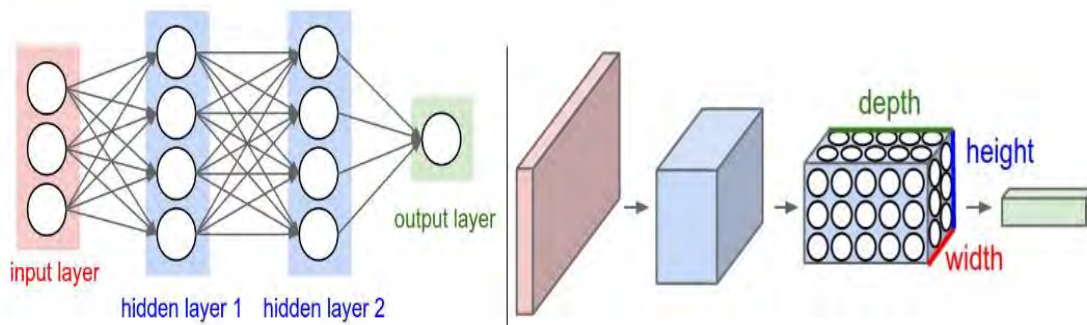


Figura 2.8: Modelo de una red neuronal común y de una red neuronal convolucional⁴

En la Figura 2.8 se muestra una comparación entre una red neuronal artificial, en la cual cada neurona está completamente conectada a sus entradas anteriores, y una red neuronal convolucional en la que cada capa está compuesta por neuronas dispuestas tridimensionalmente.

Una red neuronal convolucional está compuesta por una secuencia de capas o layers tridimensionales, entre las cuales se encuentran *Convolutional Layer*, *Pooling Layer*, *ReLU Layer*, y *Fully-Connected Layer*. La disposición de estas capas es definida por la arquitectura de la red.

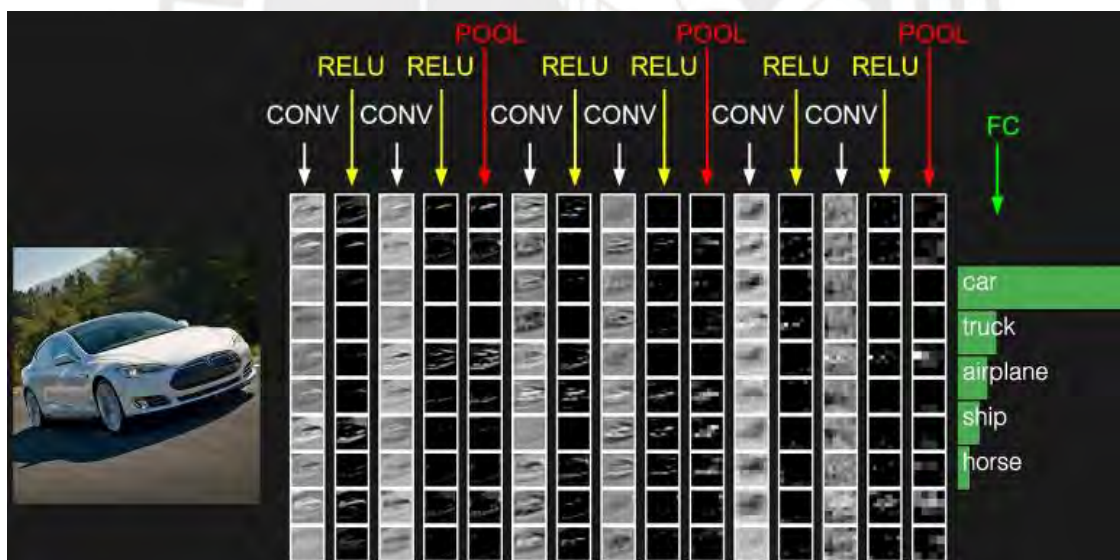


Figura 2.9: Ejemplo la estructura de una red neuronal convolucional⁴

En la Figura 2.9 se muestra cómo una red neuronal convolucional está estructurada por una secuencia de par de *Convolutional - ReLu Layers* seguido de un *Pooling Layer* y en

⁴ Convolutional Neural Networks for Visual Recognition: <https://cs231n.github.io/convolutional-networks/>

la parte final un *Fully-Connected Layer*. Dada la imagen de entrada, la CNN se encarga de clasificarla y dar una salida mediante una etiqueta que corresponde a dicha imagen.

- ***Convolutional Layer* o capa convolucional**

Es la capa más importante de una red neuronal convolucional y generalmente es la capa de entrada. Las propiedades de esta capa diferencian una CNN de una RNA. Los parámetros de esta capa consisten en un conjunto de filtros. Durante la propagación hacia adelante de una red neuronal artificial, cada neurona multiplica las entradas con sus correspondientes pesos sinápticos y luego suma estos resultados. Análogamente lo hace una red neuronal convolucional, con la diferencia de que ahora los pesos son unos filtros espaciales, cada filtro es pasado a lo largo del ancho y altura de la imagen y por tanto esta operación corresponde a la operación matemática de convolución. Es por esto esta capa recibe el nombre de *Convolutional Layer*.

Las neuronas de esta capa están conectadas a regiones locales de la imagen de entrada, por tanto, cada neurona puede extraer características visuales correspondientes a regiones de la imagen como orientación de bordes y concentración de colores. En una imagen, píxeles adyacentes tienden a estar muy correlacionados mientras que píxeles separados tienden a estar muy poco correlacionados.

Las neuronas están dispuestas de forma tridimensional, por lo que la capa tiene ancho, altura y profundidad que son definidos generalmente por cuatro parámetros que se escogen en el diseño de capa. Esta capa tridimensional puede ser dividida por planos, donde el número de planos es numéricamente igual a la profundidad, por lo que también corresponde al número de filtros. Cada plano contiene neuronas que comparten el mismo filtro ya que en la operación de convolución, el filtro es pasado a lo largo y ancho de la imagen. Es por esto que se da el compartimiento de pesos, característica propia de una CNN y es lógico usarlo ya que, si un detector de características es útil en un parte de una imagen, también será útil sobre las demás partes de la imagen.

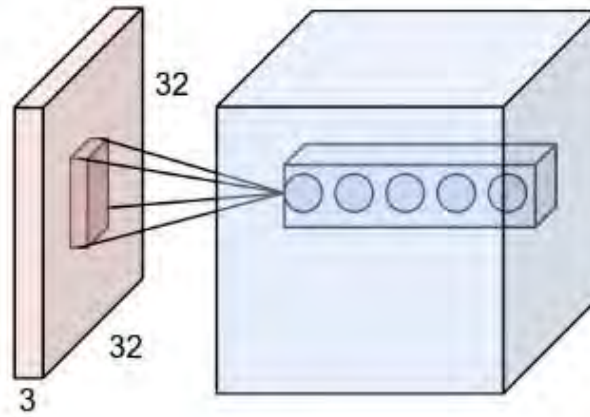


Figura 2.10: Esquema de capa convolucional⁵

- ***Pooling Layer* o capa de submuestreo**

Este layer se encarga de reducir el tamaño espacial, ancho y altura, del volumen de entrada con la finalidad de reducir la cantidad de parámetros y ayuda a minimizar el problema de pérdida de precisión en la generalización. Específicamente, aplica operaciones sobre cada plano del volumen de entrada, entre las más comunes están las operaciones de máximo y promedio. La profundidad del volumen de entrada es conservada.

- ***ReLU Layer* o capa de no-linealidad**

En este layer, cada neurona aplica una función de activación a cada elemento del volumen de entrada. La función de activación comúnmente usada es la que aplica a la entrada un umbral en cero, $\max(0, x)$, lo que permite incrementar las características de no linealidad de la red. Este layer mantiene las dimensiones del volumen de entrada.

- ***Fully-Connected Layer* o capa totalmente conectada**

Como las redes neuronales artificiales, cada neurona está completamente conectada a las que se encuentran en el volumen anterior. Su función es calcular puntajes para cada clase con la finalidad de etiquetar o clasificar a la imagen de entrada. Los valores de las neuronas de salida son calculados por la multiplicación de los pesos con los valores de la capa anterior. Luego una función *Softmax* es aplicada al resultado de la multiplicación con la finalidad de convertir los resultados en probabilidades como se muestra en la Figura 2.11.

⁵ Convolutional Neural Networks for Visual Recognition: <https://cs231n.github.io/convolutional-networks/>

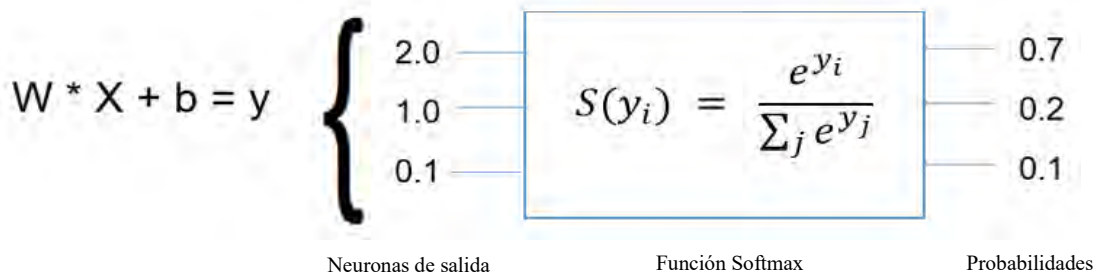


Figura 2.11: Aplicación de función *Softmax* en la salida

La probabilidad de una clase correctamente clasificada tiende al valor de 1 y las probabilidades de las demás clases tienden a ser cero. El volumen de salida tiene dimensiones $1 \times 1 \times n$, donde n corresponde al número de etiquetas a clasificar.

Finalmente, mediante estas capas descritas anteriormente, las redes neuronales convolucionales van transformando la imagen de entrada capa tras capa obteniendo como resultado un vector de puntajes, en el cual la clase que tenga mayor puntaje será la que etiquetará la imagen.

2.5.2 Detección de objetos

Mediante el entrenamiento de una simple Red Neuronal Convolucional es posible reconocer objetos, es decir, se presenta una imagen nueva a la CNN y esta la clasifica en un determinado objeto. Por ejemplo, si se le presenta una imagen que contiene un carro, la CNN la clasifica como una imagen de carro. ¿Pero qué pasa si es que en una imagen hay varios objetos? Es aquí donde se necesita abordar teorías más avanzadas de Aprendizaje de Máquina que toman como base el entrenamiento de una CNN.

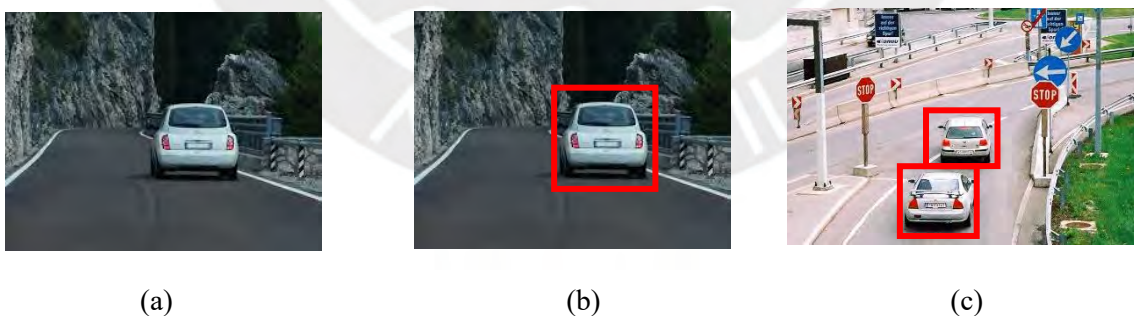


Figura 2.12: Tipos de tareas: (a) Clasificación, (b) Localización, (c) Detección de objetos⁶

En la Figura 2.12 se aprecia tres escenarios posibles en los que se usa una CNN. En el caso (a) la tarea de la CNN solo es clasificar a la imagen con una etiqueta, en este caso “carro”. En el caso (b) además de clasificar, se aprecia que localiza el carro dentro de la

⁶ Cars: <https://pixy.org/src/39/thumbs190/398506.jpg>

imagen. Mientras que en el caso (c) la tarea es detectar múltiples objetos dentro de una imagen, incluso de diferentes tipos. Este último caso es el que se requiere usar en los algoritmos del robot limpiador.

La detección de objetos es entonces una extensión del problema de clasificación. El reto añadido es identificar a un objeto y localizarlo dentro de la imagen. Existen varios métodos para identificar y localizar objetos dentro de una imagen. El más básico es mediante “*Sliding Windows*” el cual toma una ventana de cierto tamaño y la va pasando sobre el ancho y largo de una imagen, es decir, en cada pasada toma una porción de la imagen y la clasifica como se aprecia en la Figura 2.13.

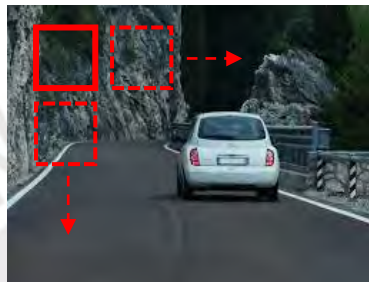


Figura 2.13: *Sliding Window Object Detection*⁷

Sin embargo, este algoritmo tiene un gran problema, el cual es que tiene un costo computacional muy alto debido a que se escogen ventanas de diferentes tamaños y por cada tamaño se tiene que ir pasando sobre la imagen para tratar de localizar un objeto. Con el objetivo de disminuir este costo computacional, se realiza una implementación convolucional de las ventanas deslizantes.

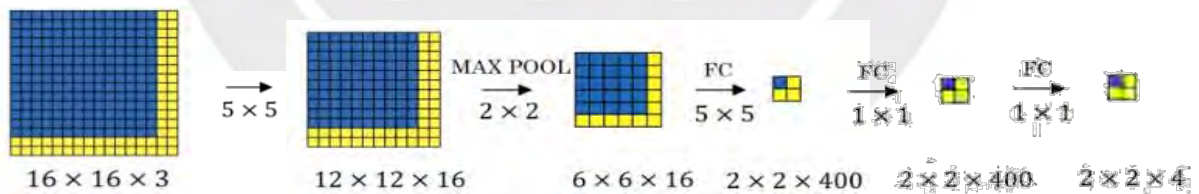


Figura 2.14: Implementación Convolutiva de *Sliding Window*⁸

⁷ Cars: <https://pixy.org/src/39/thumbs190/398506.jpg>

⁸ Coursera: <https://www.coursera.org/learn/convolutional-neural-networks/>

En la Figura 2.14 vemos que mediante una sola operación de convolución es posible obtener una salida de 4 predicciones que corresponden a como si se fuese recorrido una ventana de 14x14 cuatro veces a lo largo y ancho sobre la imagen inicial. De esta forma el alto costo computacional se disminuye. Ahora, ¿cómo es que se entrena una CNN para que detecte y localice objetos dentro de una imagen?



Figura 2.15: Etiquetas para entrenar una CNN de detección de objetos⁹

En la figura 2.15 se aprecia a modo de ejemplo, que una imagen se divide en nueve divisiones y cada celda tiene que etiquetarse con un vector de valores para luego entrenar una CNN. Para este caso, cada vector tiene ocho valores que corresponden a lo siguiente:

P_x : probabilidad de existencia de un objeto

B_x, B_y, B_w, B_h : Ubicación, ancho y altura de una celda dentro de la imagen

C_1, C_2, C_3 : Tipos de objetos

Entonces para entrenar una CNN, se tendría que tener 9 etiquetas de vectores “y” en la salida que representa si es que existe un objeto en dicha celda y qué tipo de objeto corresponde como se aprecia en la Figura 2.16.

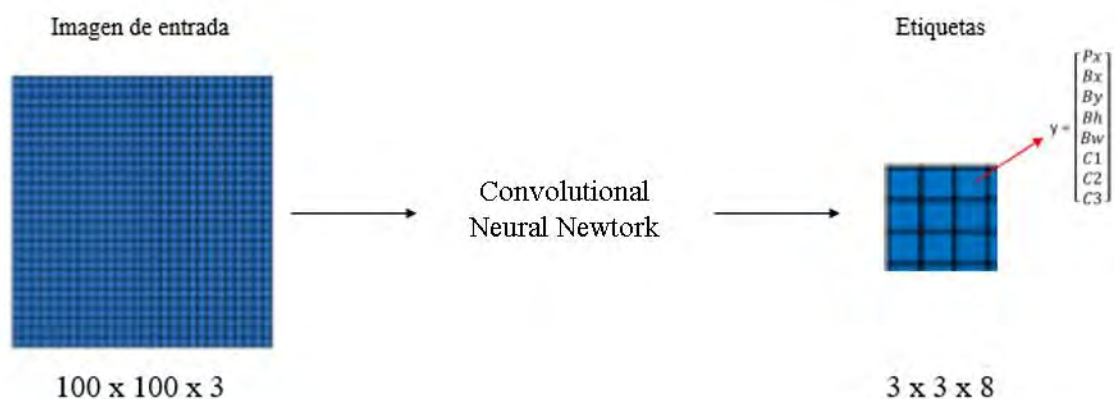


Figura 2.16: Cómo entrenar una CNN de detección de objetos⁹

⁹ Coursera: <https://www.coursera.org/learn/convolutional-neural-networks/>

- **Algoritmo YOLO**

El algoritmo “*You Only Look Once*” (YOLO) fue desarrollado para realizar en un solo performance las tareas de detección y localización de objetos dentro de una imagen. Específicamente, la ubicación de rectángulos que enmarcan los objetos (*bounding boxes*) y la predicción de clases son realizadas en una sola evaluación de la CNN sobre la imagen, y justo esto es la idea que diferencia este algoritmo sobre los demás [Redmon, 2016].

Primero, la imagen de entrada es dividida en $S \times S$ celdas. Luego “B” rectángulos enmarcadores son definidos sobre cada celda y cada uno se le calcula un puntaje de confianza. Este puntaje de confianza se refiere a la probabilidad de que exista un objeto dentro de un rectángulo de confianza. Luego mediante métodos de “*Intersection over Union*” y “*Non-max Supresion*” se van descartando los *bounding boxes* con probabilidad menos a umbral y que enmarquen mejor a cada objeto como se muestra en la Figura 2.17.

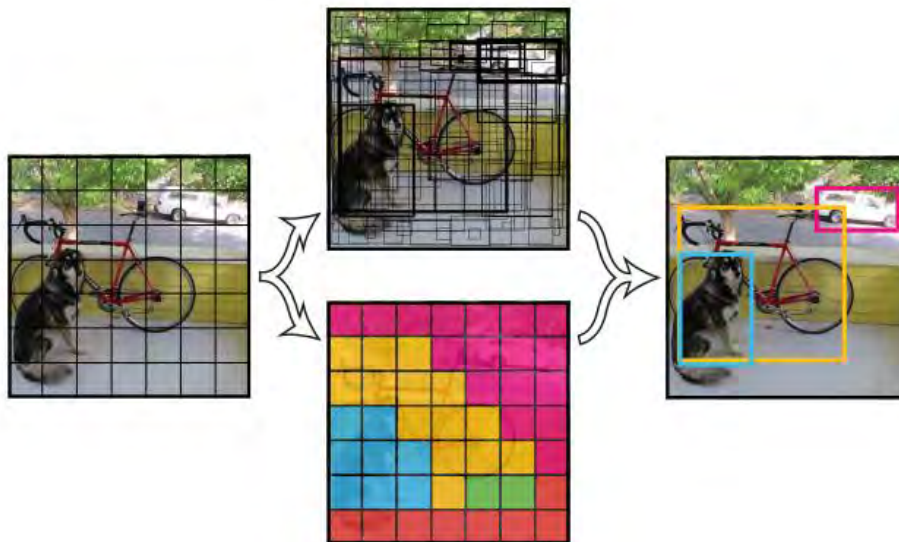


Figura 2.17: YOLO algoritmo [Redmon, 2016]

CAPÍTULO 3

DISEÑO CONCEPTUAL

En este capítulo se presenta el diseño conceptual del robot limpiador, se describen los requerimientos y alcances necesarios para cumplir con los objetivos de diseño propuestos en el primer capítulo. Asimismo, se presenta la estructura de funciones del robot y como resultado de la interacción de las funciones se genera el diagrama de estructura de funciones.

3.1 Requerimientos y alcances

3.1.1 Requerimiento general

El sistema mecatrónico principalmente tiene dos funciones que cumplir: navegar autónomamente en un ambiente requerido y limpiar eficientemente todas las partículas de suciedad presentes en el interior de dicho ambiente.

3.1.2 Requerimientos mecánicos

- El robot tendrá un diseño compacto y portable para su fácil guardado y transporte con las siguientes dimensiones como máximo: ancho no mayor a 50 cm, largo no mayor a 50 cm y altura no mayor a 25 cm.
- El sistema junto con sus accesorios no superará una masa de 6 kg.
- El robot tendrá mecanismos de cepillado, conducción de aire y filtros para la absorción de partículas de suciedad.
- Podrá limpiar partículas con tamaño mayor o igual a 0.3 micrómetros.
- El robot se desplazará a una velocidad de desplazamiento promedio de 0.2 m/s y una velocidad de giro promedio de 0.4 rad/s que le permita limpiar adecuadamente las partículas de suciedad.
- El robot podrá limpiar ambientes de diversos tipos de superficies como pisos laminados, pisos alfombrados o pisos con losetas.

- En la impresión 3D de los materiales del robot se utilizará ULTEM9085 ya que es un termoplástico de alta relación resistencia/peso ideal para la fabricación de piezas en impresión 3D.
- El polvo será almacenado en un compartimiento con capacidad no menor a 0.5 litros de fácil acceso para que pueda ser retirado de forma rápida cuando esté lleno.
- El robot será de fácil montaje y desmontaje que permita un adecuado mantenimiento y limpieza de componentes en un periodo aproximado de un año.
- Para una adecuada ergonomía, el usuario podrá accionar el robot por medio de una aplicación en su dispositivo móvil.
- El costo total del sistema incluyendo costo de diseño y costo de materiales no deberá superar el monto de S/. 12'000, precio promedio del mercado actual.

3.1.3 Requerimientos eléctricos y electrónicos

- El robot tendrá una batería que le permita tener un tiempo de limpieza no menor a una hora.
- El sistema contará con una estación de auto recarga que recibe energía eléctrica alterna 110-220 VAC (50-60 Hz) desde un tomacorriente. Cuando el robot tenga un nivel bajo de batería, se desplazará de forma autónoma hacia esta estación y comenzará su recarga de energía.
- El tiempo de recarga de la batería de robot no superará las 3 horas.
- El robot no deberá estar en contacto con agua, por lo que es necesario aislar y proteger los componentes eléctricos del sistema en consideración con la norma técnica IEC 61508 “Seguridad funcional de sistemas eléctricos / electrónicos / programables relacionados con la seguridad”.
- El sistema tendrá un sistema de regulación de voltaje que garantice la distribución adecuada de energía hacia los sensores, actuadores y otros componentes electrónicos.
- El consumo de energía del robot no superará los 100 W.
- El robot generará un ruido con un nivel máximo de intensidad de 75 dB.
- El robot tendrá un sistema de odometría que le permita estimar su posición dentro del mapa durante su recorrido.
- El robot integrará sensores para medir distancias hasta un rango de 4 metros que le permita sensor su ambiente que lo rodea, evitar obstáculos y caídas.

- El sistema tendrá un sistema embebido para el control del proceso y un microcontrolador para el control de sensores y actuadores.
- Los actuadores recibirán señales PWM desde el microcontrolador. Dichas señales PWM deberán pasar por una interfaz de potencia con una corriente máxima por canal de 5A para controlar a los actuadores como motores de desplazamiento, motores de las escobillas de limpieza y el motor del ventilador.
- Los motores de desplazamiento deberán tener un torque mínimo de 0.4 Nm y una velocidad sin carga mínima de 120 RPM para ser capaz de desplazar y girar toda la masa del robot sin problemas. Por otro lado, el ventilador centrífugo deberá tener una presión mínima de 800 Pa y un flujo de aire mínimo de 0.18 m³/min.
- La batería del robot deberá tener un voltaje nominal de 12V y una capacidad mínima de 5Ah.
- La comunicación entre el robot y la aplicación móvil se realizará por un módulo Wi-Fi o bluetooth.

3.1.4 Requerimientos de control

- El sistema tendrá un algoritmo de fusión de medida de sensores para interpretar las señales provenientes de los diversos sensores.
- El sistema integrará algoritmos que le permita construir un mapa geométrico del ambiente donde se encuentra.
- El sistema tendrá algoritmos de localización para que se ubique correctamente en cualquier instante dentro del ambiente.
- El robot tendrá un algoritmo de navegación eficiente que le permita realizar la limpieza de ambientes de forma totalmente autónoma.
- El robot deberá ser capaz de navegar un ambiente con múltiples habitaciones y diferenciar las zonas limpiadas de aquellas que le falta cubrir.
- El sistema contará con una aplicación que le permita conectarse con un dispositivo móvil por comunicación inalámbrica que permita al usuario programar horarios de limpieza y recibir notificaciones del estado del proceso.

Luego de definir los requerimientos del sistema se procede a realizar el diseño conceptual del robot limpiador. El objetivo de este diseño conceptual es elaborar un concepto de solución óptimo que satisfaga todos los requerimientos planteados. Se formulará una estructura de funciones que el robot a diseñar debe cumplir para lograr su objetivo

principal. Por cada función determinada se buscará principios de solución apropiados para armar una matriz morfológica. Luego en la matriz se buscará posibles caminos de solución que permita tener alternativas de solución. Finalmente, se realizará una evaluación técnica-económica de las alternativas para determinar el concepto de solución óptimo del robot limpiador.

3.2 Estructura de funciones

3.2.1 Caja negra

Antes de determinar las funciones necesarias, podemos representar el proceso general del robot limpiador y los procesos por cada dominio del sistema como cajas negras que suponen una transformación de magnitudes de entrada en magnitudes de salida.

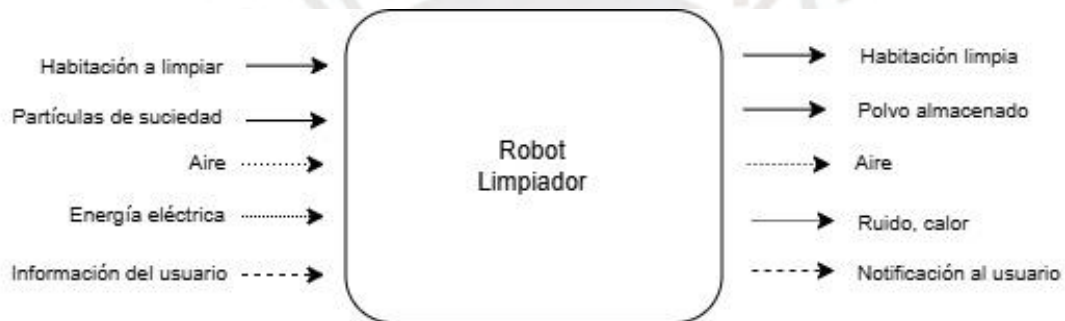


Figura 3.1: Caja negra general



Figura 3.2: Caja negra dominio eléctrico

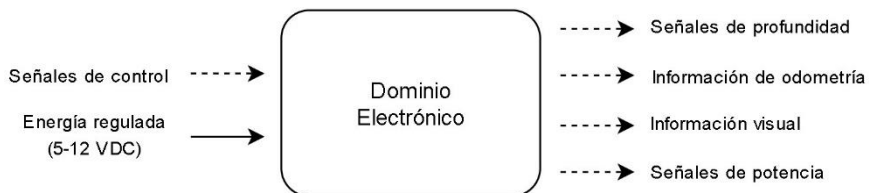


Figura 3.3: Caja negra dominio electrónico



Figura 3.4: Caja negra dominio de control



Figura 3.5: Caja negra dominio mecánico

3.2.2 Funciones

Luego del proceso de abstracción anterior, resulta necesario definir todas las funciones que el robot limpiador debe cumplir para transformar las magnitudes de entrada en magnitudes de salida y así de esta forma llevar a cabo su objetivo principal. A continuación, se listarán y describirán las funciones que el robot debe cumplir:

Funciones eléctricas:

- **Transformar energía:** Esta función la realizará la estación de recarga del sistema. Recibe energía eléctrica alterna 110-220 VAC (50-60 Hz) desde un tomacorriente y la transforma en energía continua para recargar energía al robot.
- **Almacenar energía:** El robot móvil necesita energía para desplazarse en el ambiente. Para esto, la energía transformada por la estación de recarga se debe almacenar en una batería que energice al robot.
- **Regular energía:** La energía almacenada en la batería del robot debe ser regulada para que alimente correctamente a los componentes electrónicos del sistema.

Funciones mecánicas:

- **Atraer partículas:** Cuando el robot se desplaza en el ambiente, debe atraer las partículas de suciedad que hay a sus alrededores.
- **Conducir partículas:** Una vez atraídas las partículas de suciedad, estas son conducidas hacia la zona interior del robot.

- **Filtrar partículas:** Las partículas conducidas hacia el interior deben ser filtradas para que sean separadas del aire que las conduce.
- **Almacenar polvo:** Después de ser filtradas las partículas de suciedad, estas son almacenadas en un depósito.
- **Pulir piso:** El robot también pule el piso con la finalidad de dar un acabado brillante a este.
- **Desplazamiento:** El robot debe desplazarse sobre la habitación que está limpiando.

Funciones electrónicas:

- **Sensar habitación:** En cada instante, el robot mide las distancias de los puntos que se encuentran a su alrededor. Esto le sirve para el mapeo y localización.
- **Estimar posición:** Esta función realiza odometría para estimar la posición del vehículo durante su recorrido.
- **Capturar imágenes:** Permite obtener información visual del ambiente.
- **Evitar obstáculos/caídas:** El móvil debe evitar los obstáculos que se encuentran a su alrededor y reconocer escaleras o espacios vacíos para evitar caerse.
- **Sensar nivel de batería:** Es necesario sensar constantemente el nivel de batería para que cuando presente un nivel bajo, se dirija hacia su estación de recarga.
- **Enviar/recibir información:** Intercambiar información con la aplicación del usuario.
- **Interfaz de potencia:** Permite adaptar las señales de control hacia señales de potencia para accionar los actuadores del sistema.

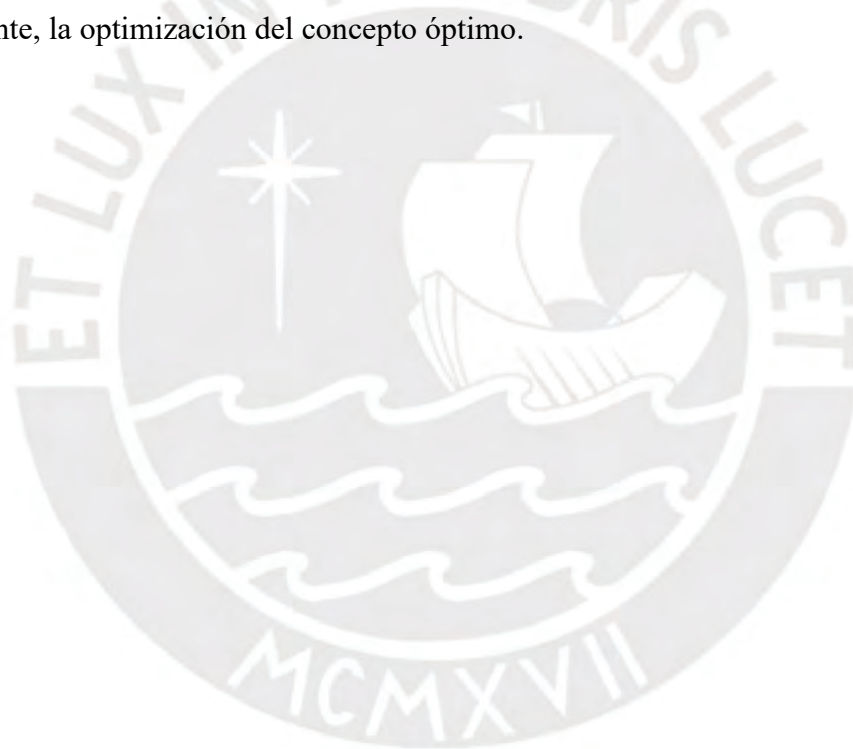
Funciones de control:

- **Inicializar sistema:** El sistema recibe información del usuario a través de la aplicación móvil como por ejemplo para que el robot comience a limpiar.
- **Adaptar señales de sensores:** El sistema recibe señales de los sensores y las adapta para ser información útil del sistema embebido.
- **Construcción de mapa:** Cuando el robot inicia su proceso, el primer paso que debe realizar es construir un mapa geométrico de su ambiente.
- **Localización:** En cualquier instante del recorrido de limpieza, el robot debe reconocer su ubicación exacta dentro del ambiente.

- **Navegación:** Construido el mapa y capaz de localizarse, ahora el robot debe navegar en dicho ambiente.
- **Verificar estado de limpieza:** Según la trayectoria planeada y el recorrido que ha hecho el robot, se puede saber cuánto le falta para acabar de limpiar la habitación. Este progreso se notifica al usuario mediante la aplicación móvil.

Luego como resultado de la interacción de estas funciones se genera la estructura de funciones, como se muestra en la Figura 3.6. Esta estructura representa en detalle la contribución de estas funciones en el objetivo principal.

Posteriormente en el Anexo A se muestra el desarrollo de la matriz morfológica, los conceptos de solución alternativos de solución, una evaluación técnica-económica de las alternativas que permite determinar el concepto de solución óptimo del robot limpiador y, finalmente, la optimización del concepto óptimo.



3.2.3 Diagrama de estructura de funciones

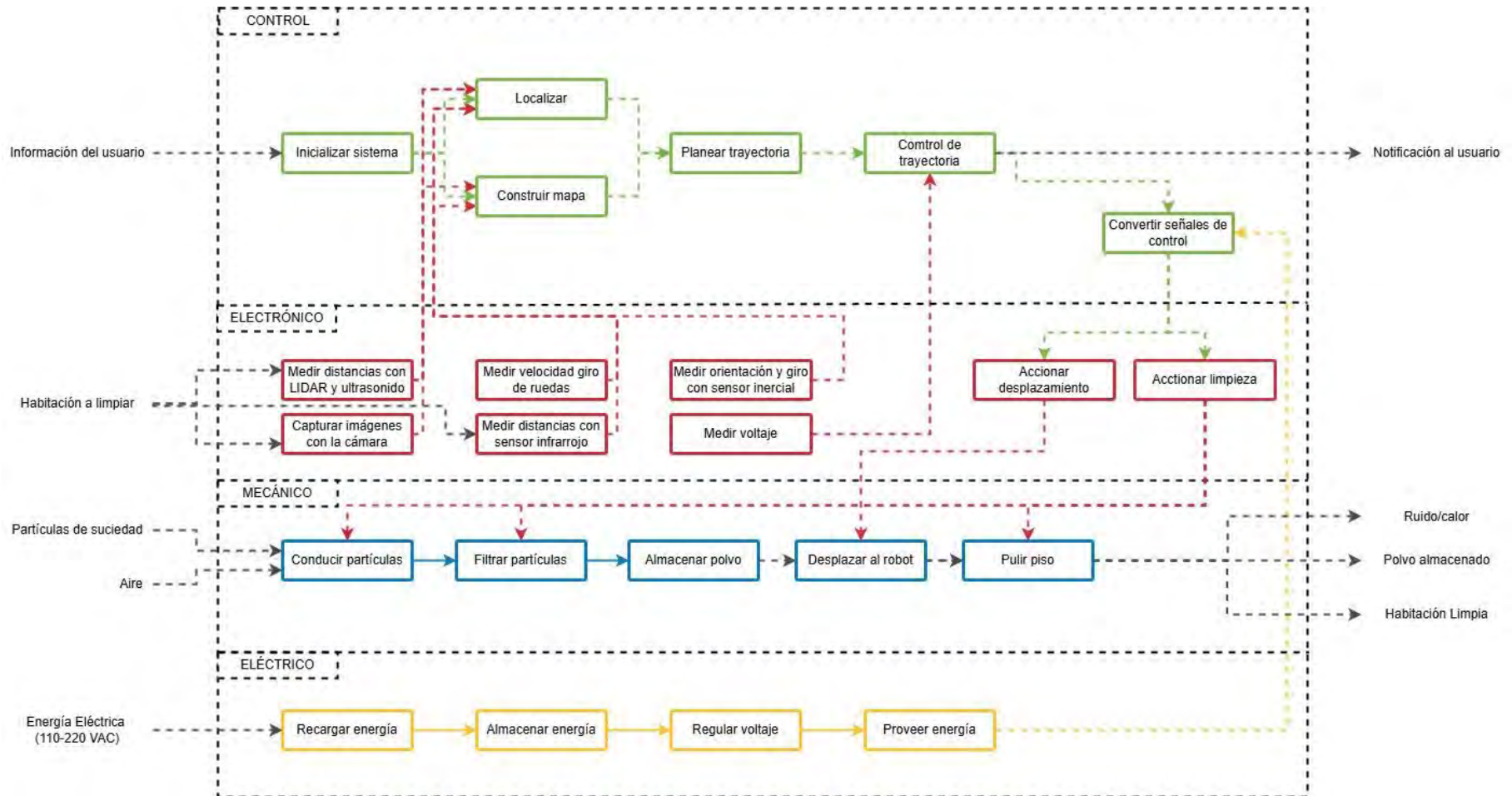


Figura 3.6: Estructura de funciones

CAPÍTULO 4

INGENIERÍA DE DETALLE DEL SISTEMA MECATRÓNICO

El objetivo de este capítulo es desarrollar el sistema mecatrónico del robot limpiador autónomo mostrando la integración de los dominios mecánico, electrónico y de control. En primer lugar, se presenta el robot limpiador diseñado mostrando sus principales características. Luego se describe el desarrollo del dominio mecánico del sistema describiendo todos los subsistemas que lo componen. Posteriormente, se presenta el desarrollo del dominio electrónico presentando todos los componentes electrónicos del sistema. Finalmente, se detalla el dominio de control describiendo el diagrama de flujo del sistema, así como también las estrategias de control.

4.1 Presentación del robot autónomo de limpieza

En la Figura 4.1 se presenta el robot limpiador autónomo diseñado para cumplir adecuadamente las funciones de navegar autónomamente en una habitación requerida y limpiar eficientemente todas las partículas de suciedad presentes en el interior de dicho ambiente. El robot limpiador tiene una masa de 2.3 kg y cuenta con las siguientes dimensiones: 360 mm de largo, 360 mm de ancho y una altura de 145 mm. Un punto que resaltar es el diseño compacto con que cuenta el robot que le permite un fácil mantenimiento. Asimismo, resaltar que tiene una baja altura de tan solo 14.5cm, que le da la capacidad de interactuar mejor con los objetos que puede tener una habitación arbitraria.

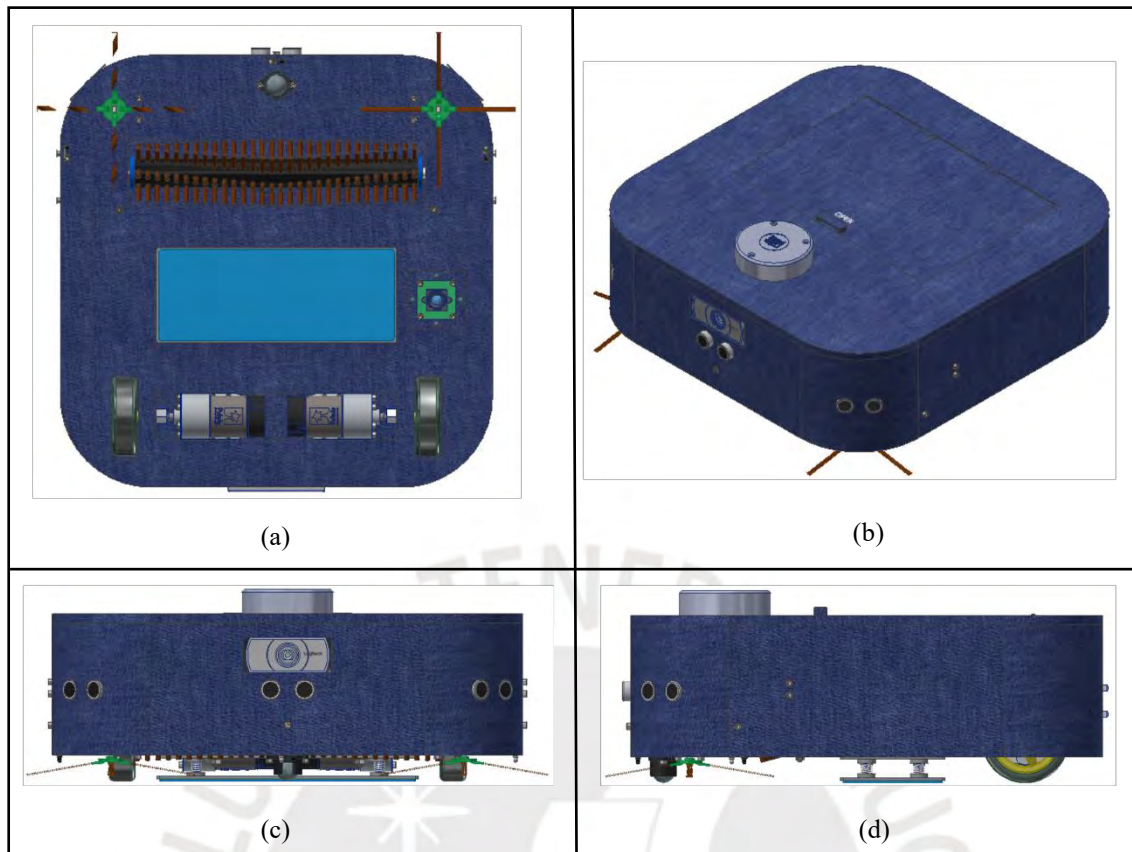


Figura 4.1: Diseño de robot autónomo propuesto
 (a) Vista inferior, (b) Vista isométrica, (c) Vista frontal, (d) Vista lateral

El robot está conformado principalmente por tres subsistemas: cubierta exterior, limpieza y desplazamiento. En las imágenes de la Figura 4.1 se aprecia como la cubierta exterior se encarga de soportar y proteger a todos los componentes del sistema. El subsistema de limpieza se encarga de atraer las partículas de suciedad, absorberlas, filtrarlas, almacenarlas en un depósito y posteriormente pulir el piso de la habitación.

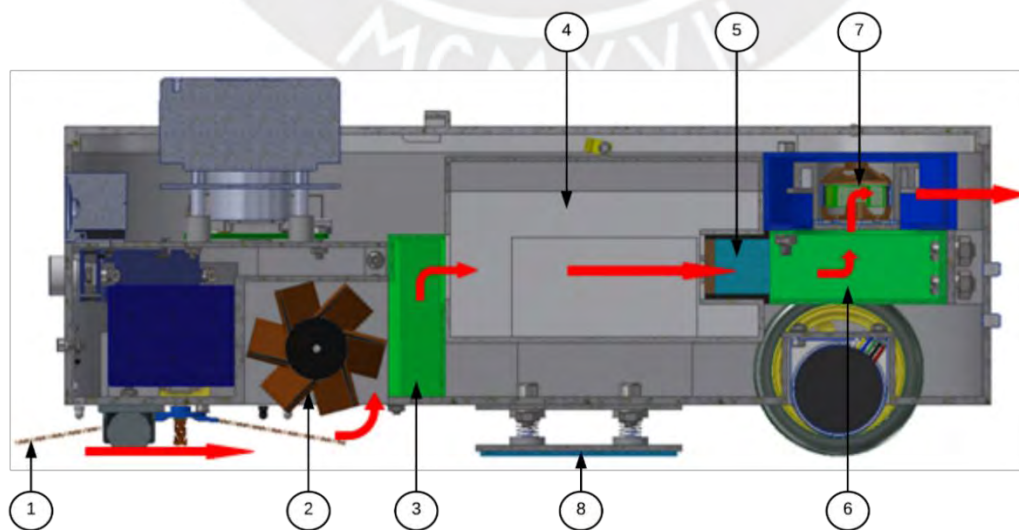


Figura 4.2: Esquema de limpieza del robot

En la Figura 4.2 se muestra el esquema de limpieza del robot. En primer lugar, el robot cuenta con dos escobillas laterales (1) y una escobilla central (2) que se encargan de atraer las partículas de suciedad que se encuentran alrededor del robot. Luego cuenta con un conducto de entrada (3) que facilita el ingreso del polvo. Las partículas de suciedad son almacenadas en un depósito (4) luego de ser filtradas (5). Posteriormente, el aire que sale del filtro es llevado hacia el ventilador mediante un conducto de salida (6). El ventilador centrífugo (7) se encarga de generar un flujo de aire que conduce las partículas de suciedad. Finalmente, el piso de la habitación es pulido mediante un trapo (8).

En el desplazamiento se utiliza una configuración diferencial, la cual consiste en dos ruedas accionadas que dirigen el movimiento y una rueda de libre giro que sirve como apoyo. Las ventajas de tener un desplazamiento diferencial son que dispone de la cantidad mínima de actuadores y es una configuración fácil de controlar.

El robot se encuentra equipado con un conjunto de sensores que le permite sentir la habitación que se encuentra limpiando, evitar obstáculos y caídas, estimar la posición durante su recorrido, capturar imágenes y conocer su orientación.

El robot limpia una habitación requerida de una forma totalmente autónoma. Para esto, el robot inicia su tarea posicionado en un lugar totalmente desconocido. Entonces lo primero que realiza es un recorrido sobre la habitación con la finalidad de captar información y de esta forma construir un mapa. Cuando el robot ya tiene construido el mapa del lugar donde se encuentra, se dispone a navegar de forma autónoma con el objetivo de limpiar dicho ambiente. Esto lo logra mediante la integración de algoritmos de localización y planeamiento de trayectorias. Asimismo, el sistema cuenta con una estación de recarga para que cuando la batería del robot tenga un nivel bajo de energía, el robot se desplace de forma autónoma hacia esta estación y comience a recargar la batería como se muestra en la Figura 4.3.

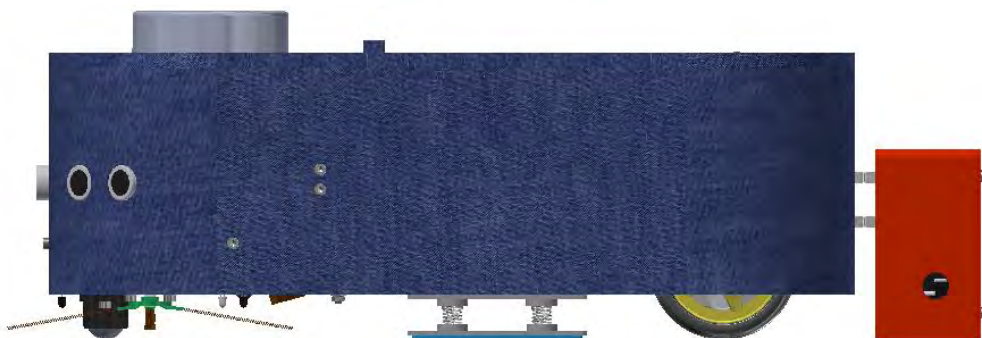


Figura 4.3: Robot recargando energía

4.2 Sistema mecánico

En primer lugar, el robot limpiador dispone de un diseño compacto y forma cuadrada con redondeos a las esquinas que le permite atraer mejor las partículas de suciedad. Sus dimensiones exteriores son de 360 mm ancho, 360 mm de largo y 145 mm de altura como se muestra a continuación.

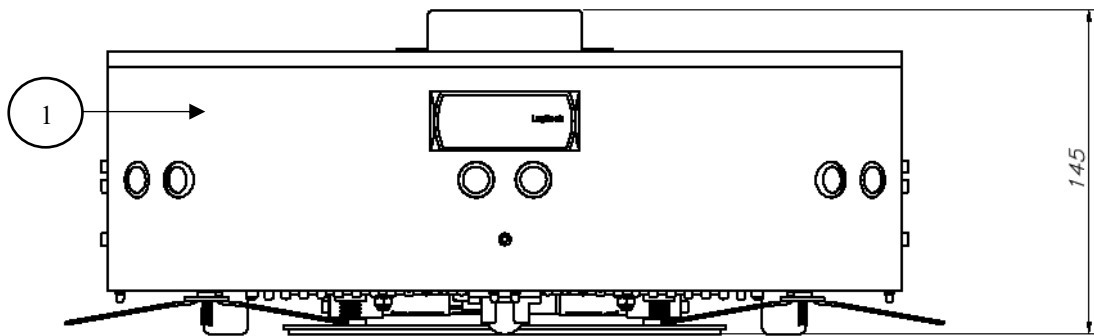


Figura 4.4: Vista frontal del robot

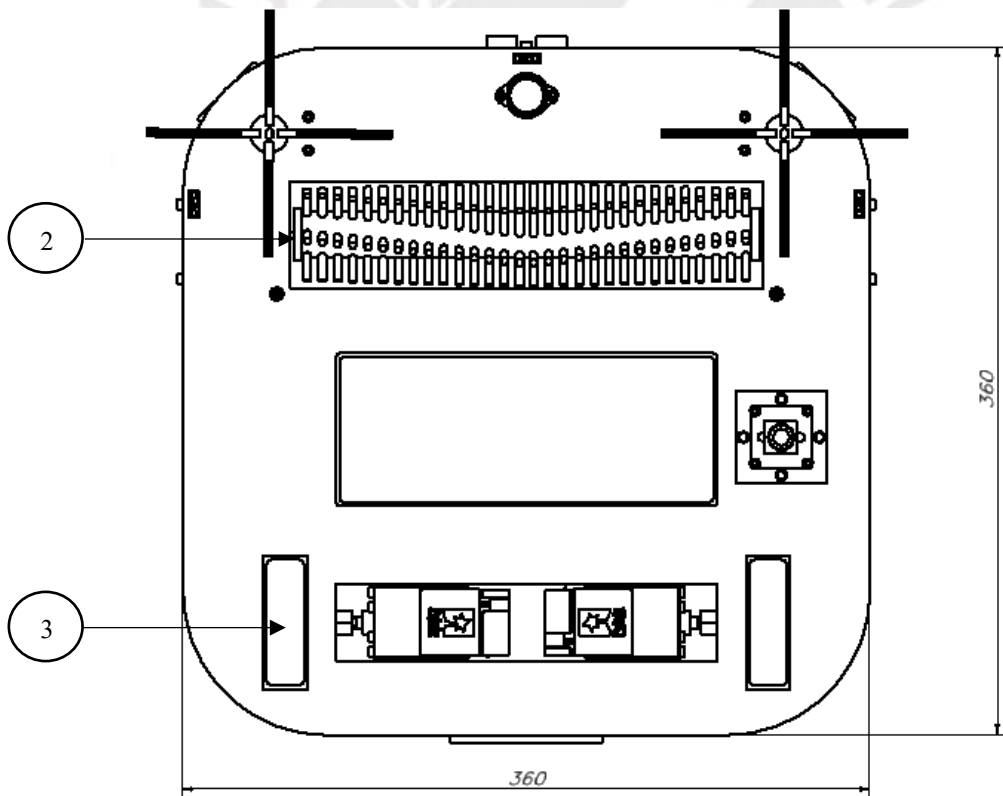


Figura 4.5: Vista inferior del robot

El robot posee una carcasa exterior (1) que protege a todos los componentes mecánicos y electrónicos. Cuenta con un subsistema de limpieza (2) con mecanismos de cepillado, conducción de aire y filtros para la absorción de partículas de suciedad. Finalmente, posee un subsistema de desplazamiento (3) que le permite trasladarse sobre una habitación.

4.2.1 Carcasa exterior

La carcasa exterior cumple la función de soportar y proteger a los demás componentes del sistema. Está conformada por dos partes: (1) carcasa inferior y (2) carcasa superior. La parte inferior protege a todos los componentes del interior y además se acopla a un soporte de componentes electrónicos para soportar los diversos sensores y controladores. La parte superior principalmente posee una compuerta que se abre cuando se retira el almacén de polvo. Este subsistema en conjunto tiene un largo de 360 mm, un ancho de 360 mm y una altura de 115 mm como se muestra en la Figura 4.6. Las paredes de ambas carcasas tienen un espesor de 3 mm. Tanto la parte inferior como la superior estarán hechas de material ULTEM 9085, termoplástico de alta relación resistencia/peso y serán fabricadas mediante impresión 3D.



Figura 4.6: Carcasa exterior

4.2.2 Subsistema de limpieza

Este subsistema cumple la función de limpiar las partículas de suciedad y pulir el piso de una habitación, por lo que cuenta con mecanismos de cepillado, conducción de aire y filtros para la absorción de partículas de suciedad. Estos mecanismos de limpieza están formados por los siguientes componentes: dos escobillas laterales y una escobilla central que se encargan de atraer las partículas de suciedad que se encuentran alrededor del robot; un conducto de entrada que facilita el ingreso del polvo; un almacén donde se depositan las partículas de suciedad; un filtro de polvo; un conducto de salida; un ventilador centrífugo que se encarga de generar un flujo de aire que conduce las partículas de suciedad; y, finalmente, un trapo que pule el piso de la habitación.

4.2.2.1 Escobillas laterales

Cada escobilla lateral está formada por dos partes: (1) la base de la escobilla y (2) las cerdas. La escobilla lateral tendrá una forma circular con diámetro de 128 mm y altura de 15 mm como se muestra en la Figura 4.7. El material de la base de la escobilla será de ULTEM 9085 ya que es un termoplástico con una alta relación de resistencia/peso ideal para fabricación mediante impresión 3D. Con respecto a las cerdas, estas estarán hechas de material sintético Nylon tipo 6.6 debido a que tiene excelente recuperación de doblado, resistente a la abrasión, duradero y de bajo costo.

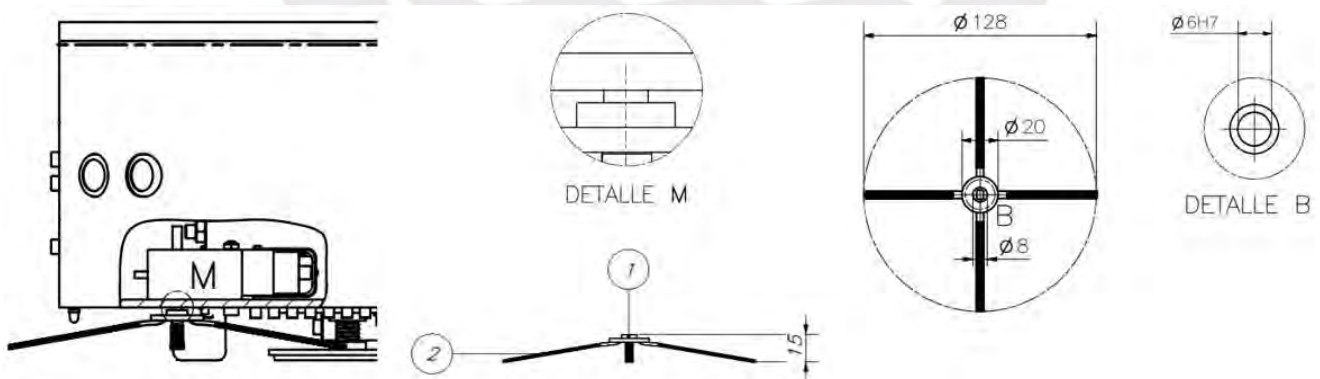


Figura 4.7: Escobilla lateral

4.2.2.2 Escobilla central

Este subsistema que corresponde a la escobilla central está formado por cinco partes: (1) eje central, (2) rodamientos (3) escobilla cilíndrica, (4) un acople lineal, (5) anillos de retención, (6) chumaceras de pared y (7) chaveta. La longitud de la escobilla cilíndrica es de 220 mm y su sección transversal tiene una forma circular con diámetro de 50 mm como se muestra en la Figura 4.8. El cuerpo de la escobilla central será de material ULTEM 9085, termoplástico de alta relación resistencia/peso. Las cerdas de esta escobilla serán de material sintético Nylon tipo 6.6, que presenta buena recuperación de doblado y es resistente a la abrasión. Luego es necesario tener un eje que le transmita movimiento a esta escobilla. Este eje tendrá diámetro 5mm con longitud de 260 mm y será de material Aluminio 6061. El eje transmite movimiento a la escobilla cilíndrica por medio de una chaveta tipo A de dimensiones 2 mm x 2 mm x 20 mm. En ambos extremos, el eje tiene insertado rodamientos SKF 623 con norma DIN 625 para tener un apoyo de giro. Estos rodamientos se encuentran insertados en sus respectivas chumaceras de pared con norma DIN 616 para que todo el subsistema pueda ser acoplado al robot. Asimismo, en ambos extremos de los rodamientos, el eje tiene insertado dos anillos de retención tipo A, diámetro 3 mm y norma DIN471 para evitar el desplazamiento axial del eje. Finalmente, en el extremo derecho, el subsistema tiene acoplado un acople lineal de 6 mm y material Aluminio 6061 para conectar el eje del motor DC.

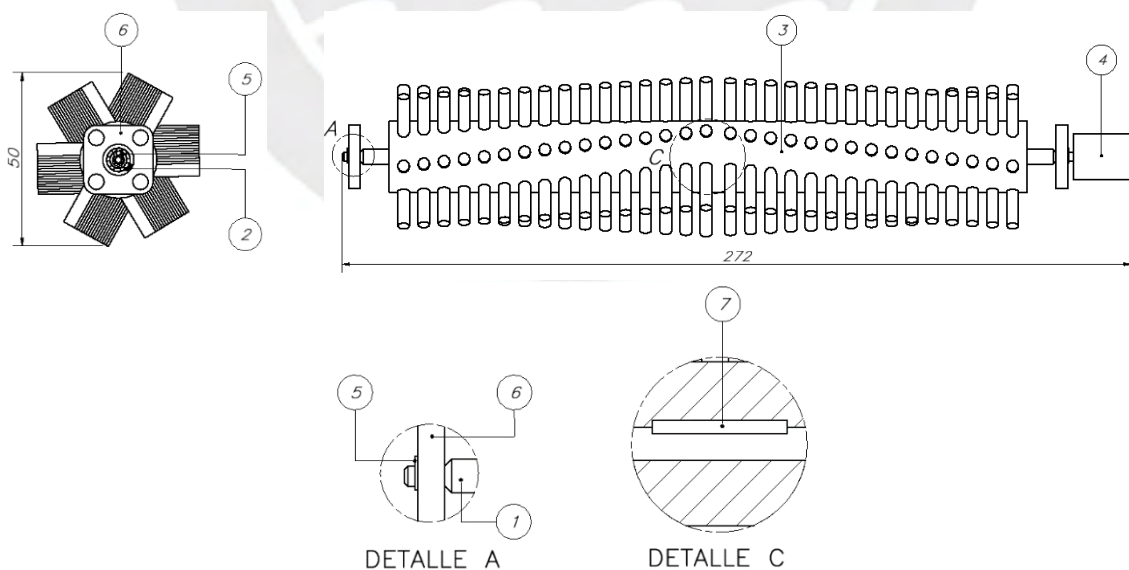


Figura 4.8: Esquema de escobilla central

4.2.2.3 Conducto de entrada

Las partículas de suciedad atraídas por las escobillas son conducidas hacia el interior de robot por medio del flujo de aire. El conducto de entrada tiene dos aberturas: la abertura de entrada por la cual ingresa el flujo de aire proveniente del exterior y una abertura de salida que direcciona este flujo hacia el almacén de polvo. Las dimensiones de este conducto de entrada son de 268 mm de ancho, 63 mm de altura y 22 mm de profundidad como se muestra en la Figura 4.9. La geometría del conducto en concepto es similar a la de un embudo, por lo que tiene una abertura grande de entrada y una abertura reducida de salida para conducir el polvo hacia el almacén. Para el diseño de la geometría de este conducto se tomó como referencia conductos de entrada de robots limpiadores en el mercado citados en el capítulo 1 de estado del arte como el *Samsung Powerbot* y el *Neato Botvac*. El material del conducto también es ULTEM 9085 para que sea fabricado mediante impresión 3D.

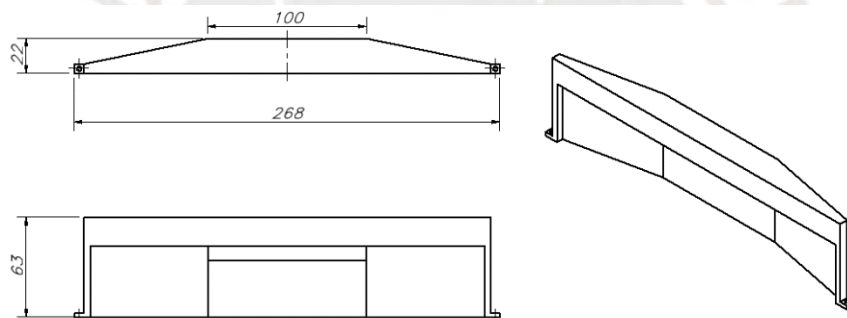


Figura 4.9: Conducto de entrada

4.2.2.4 Almacén de polvo

En este almacén se depositan las partículas de suciedad absorbidas. Está conformado de tres partes: (1) la base inferior, (2) la tapa del almacén y (3) la bisagra que une a las dos partes anteriores como se muestra en la Figura 4.10. Este almacén tiene una longitud de 204 mm, un ancho de 124 mm y una altura de 72 mm y una capacidad de almacenar 1.7 litros de polvo. El material de la base inferior como también de la tapa es ULTEM 9085. Por la parte frontal tiene un agujero rectangular que permite el ingreso del polvo y por la parte de atrás tiene otro agujero rectangular donde se coloca el filtro de polvo.

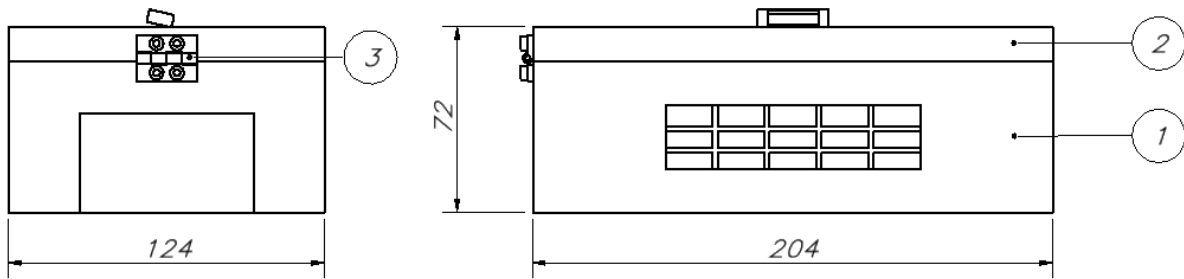


Figura 4.10: Almacén de polvo

4.2.2.5 Filtro de polvo

En el almacén se coloca un filtro para atrapar las partículas de suciedad y solo dejar pasar el flujo de aire. Este subsistema está formado de tres componentes: (1) carcasa exterior, (2) filtro HEPA y (3) una esponja, como se muestra en la Figura 4.11. El subsistema tendrá una longitud de 106 mm, una altura de 29 mm y una profundidad de 25 mm. La carcasa exterior estará hecha de material ULTEM 9085. Se usará un filtro de aire de alta eficiencia tipo HEPA 14 que tiene la capacidad de filtrar partículas con tamaño mayor o igual a 0.3 micrómetros. Este filtro atrapa diversos tipos de partículas como esporas de moho, polvo, ácaros y pelos de mascotas. Adicionalmente, se coloca una esponja rectangular de poliuretano como primer contacto de filtrado y atrapar partículas de mayor tamaño.

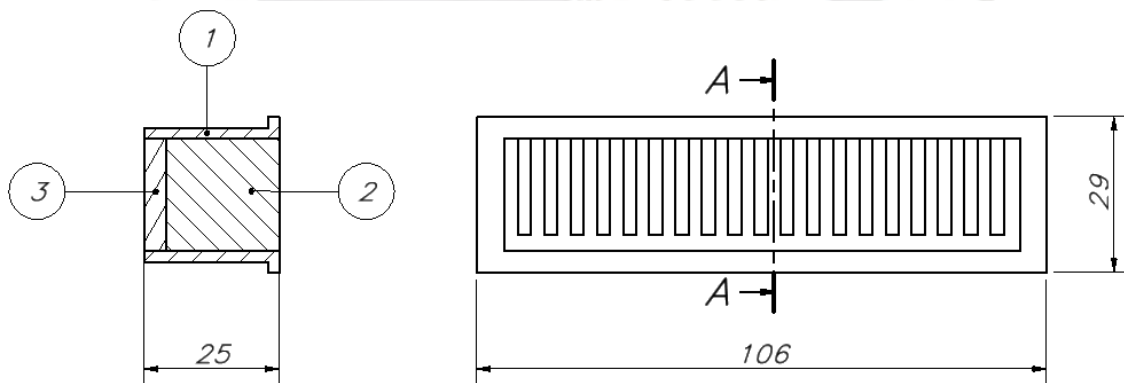


Figura 4.11: Filtro de polvo

4.2.2.6 Conducto de salida

Después de que las partículas de suciedad son filtradas y almacenadas, el flujo de aire continua su camino y es llevado hacia el ventilador por medio de un conducto de salida. Este conducto tiene dos aberturas: la abertura de entrada por la cual ingresa el flujo de aire proveniente del filtro y una abertura de salida que direcciona este flujo hacia el ventilador. Las dimensiones de este conducto de entrada son de 104 de ancho, 29mm de

altura y 69mm de profundidad como se muestra en la Figura 4.12. El material del conducto también es ULTEM 9085 para que sea fabricado mediante impresión 3D.

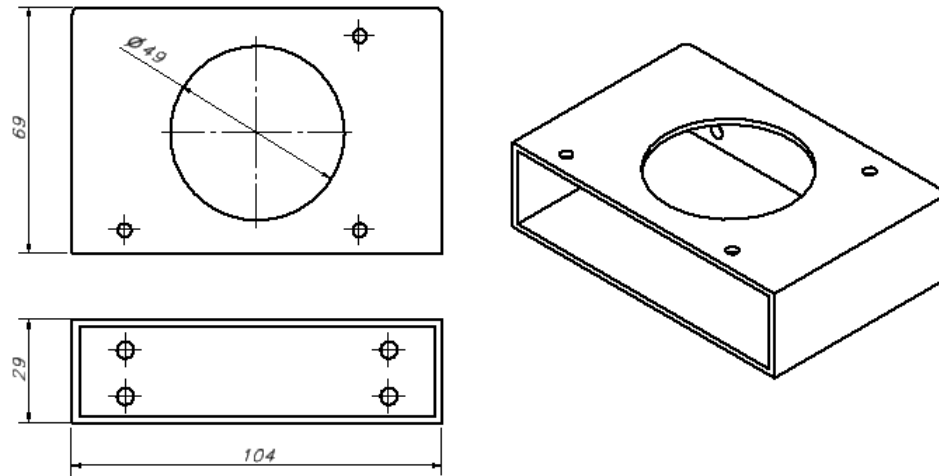


Figura 4.12: Conducto de salida

4.2.2.7 Trapo de pulido

Luego de que el robot ha absorbido y almacenado las partículas de suciedad, el robot cuenta con un trapo con la finalidad de pulir el piso que está limpiando. Este subsistema está conformado por cuatro partes: (1) base superior, (2) resortes, (3) base inferior y (4) trapo como se muestra en la Figura 4.13. Este subsistema tiene una longitud de 200 mm, un ancho de 80 mm y una altura de 20 mm. Tanto la base inferior como la base superior estará hecha de material ULTEM 9085. Los resortes son de material acero inoxidable AISI 1018 que cumplen la función de amortiguar en caso de que el trapo impacte con un objeto por debajo del robot. El trapo de limpieza será desechable, hecho de algodón y sirven para retirar la suciedad superficial remanente y así de esta forma pulir el piso.

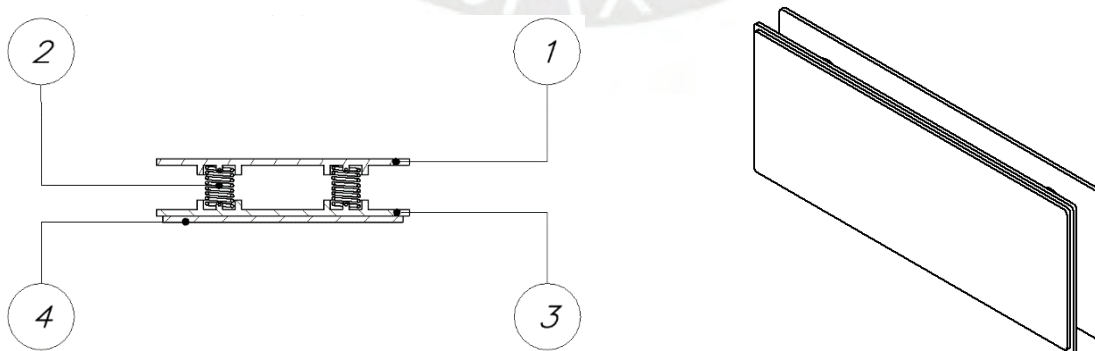


Figura 4.13: Trapo de pulido

4.2.2.8 Subsistema de desplazamiento

El robot limpiador se desplaza mediante una configuración diferencial cuyo movimiento se basa en dos ruedas accionadas cada una por un motor y alineadas sobre un eje común. En cada lado, el robot dispone de un motor (1) que se une a la carcasa mediante un soporte de sujeción (2). El eje de salida del motor es conectado a un acople lineal (3) que transmite movimiento a una rueda todo terreno (4) para así transmitir potencia como se muestra en la Figura 4.14. El soporte de sujeción es de material Aluminio 6061 y se acopla a la carcasa inferior (5) mediante seis tornillos M3 con norma DIN 912 (36), seis arandelas M3 con norma DIN 125 (31) y seis tuercas hexagonales M3 con norma DIN 934 (32). El motor es de la marca Pololu con diámetro 37 mm, largo de 57 mm y un eje de salida con diámetro de 6 mm. El motor se une al soporte de sujeción mediante seis tornillos M3 con norma DIN 912 (37). El acople lineal está hecho de material Aluminio 6061 y tiene un diámetro de 6 mm. El acople, por un lado, se une al eje del motor mediante ajuste con tolerancia h6 y por el otro lado, se une a la rueda con un eje (6) de material Aluminio 6061 tolerancia h6. La rueda es de la marca DFRobot con material de revestimiento de goma, tiene un diámetro de 68 mm y 15 mm de ancho.

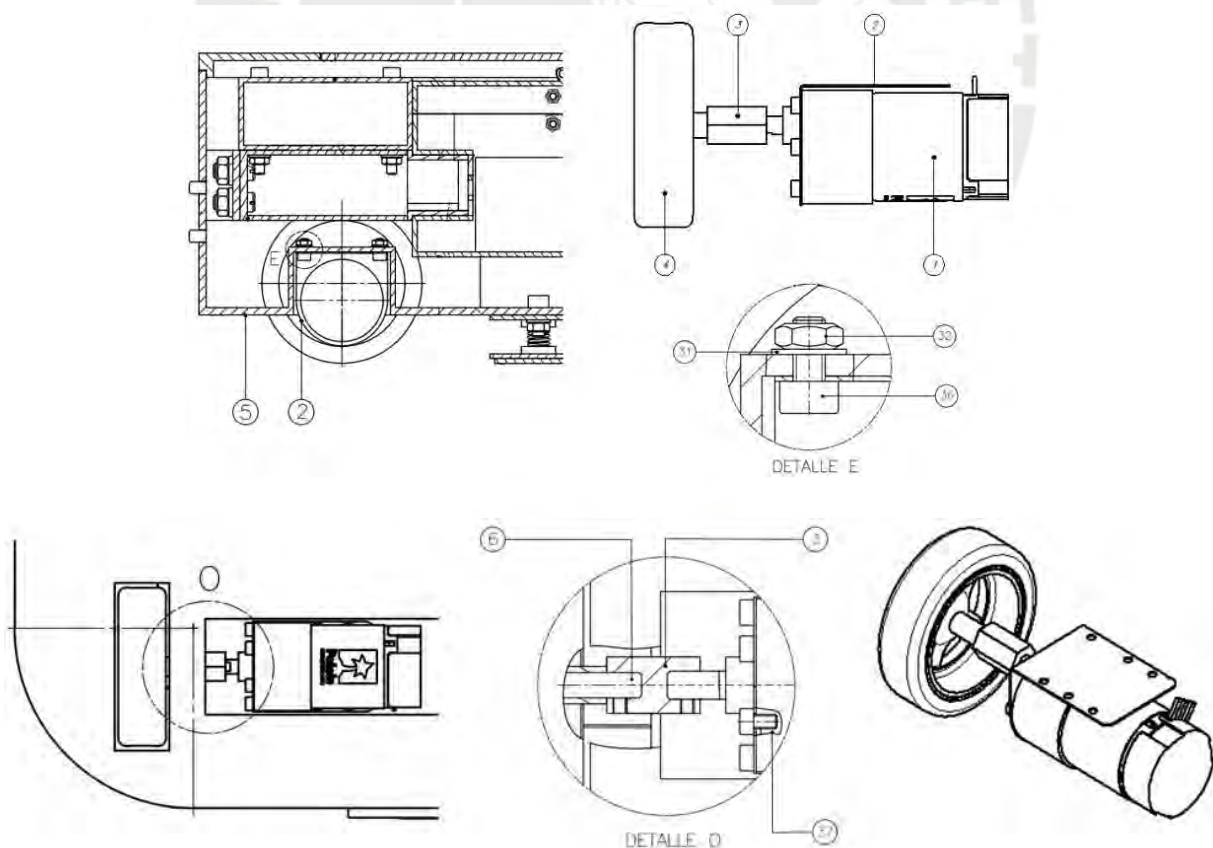


Figura 4.14: Subsistema de desplazamiento

4.2.2.9 Soporte de componentes electrónicos

El soporte de componentes electrónicos se acopla a la carcasa inferior y se encarga de soportar a los componentes electrónicos y de control como el controlador principal, el controlador esclavo, la cámara, el sensor LIDAR y los sensores ultrasonidos como se muestra en la Figura 3.14. Este soporte se encuentra ubicado en la parte superior delantera y contiene aberturas para reducir el cableado. Este soporte tiene dimensiones de 354 mm de largo, 122 mm de largo y un espesor de 2 mm. El material del soporte es ULTEM 9085 para ser fabricado mediante impresión 3D.

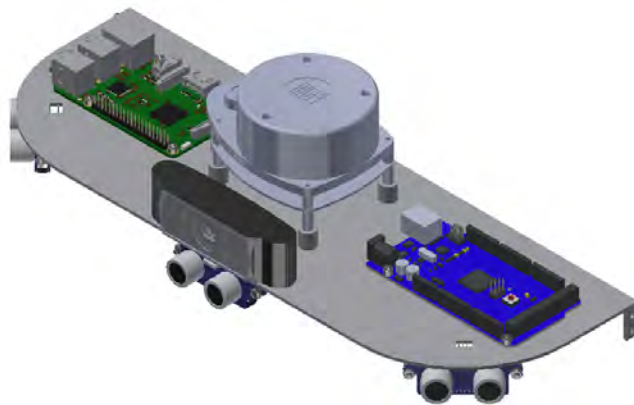


Figura 4.15: Soporte con componentes electrónicos

4.2.2.10 Estación de recarga

Este accesorio permite que cuando la batería del robot presente un nivel bajo de energía, el robot se desplace hacia esta estación y recargue la batería. Está conformado de tres partes: (1) carcasa, (2) contactores y (3) fuente DC como se muestra en la Figura 4.16. Este subsistema tiene una longitud de 150 mm, un ancho de 52 mm y una altura de 85 mm. El material de la carcasa es ULTEM 9085 para que sea fabricado mediante impresión 3D.

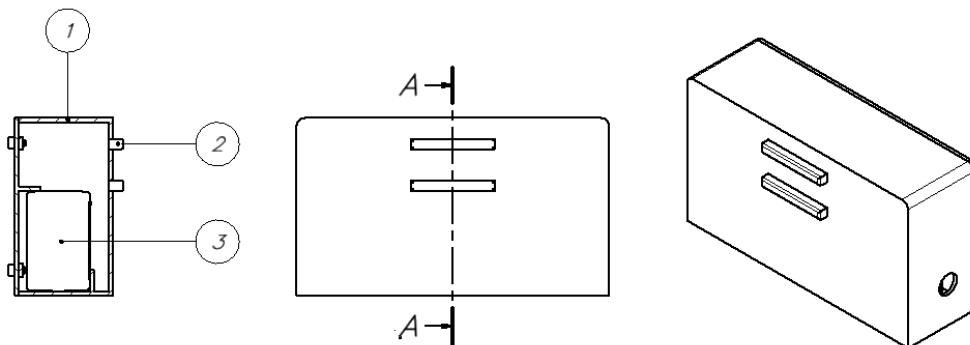


Figura 4.16: Estación de recarga

4.3 Sistema electrónico

En esta sección, en primer lugar, se presenta el diagrama de bloques del sistema y luego se presentan los componentes electrónicos del sistema describiendo sus principales características y sus diagramas esquemáticos. Los cálculos de selección y tablas comparativas de componentes se encuentran en el Anexo B.

4.3.1 Diagrama de bloques

En la Figura 4.17 se muestra el diagrama de bloques del sistema, en el cual se detalla cómo interactúan los componentes del sistema.

En primer lugar, el sistema recibe energía eléctrica alterna 110-220 VAC (50-60 Hz) desde un tomacorriente y mediante una fuente DC la transforma en energía continua. Esta energía transformada se almacena en una batería que se encarga de energizar al robot. Posteriormente, la energía de la batería es regulada a 5 VDC y 12 VDC para que energice correctamente a los componentes electrónicos del sistema.

Luego el sistema posee un conjunto de sensores que le permiten adquirir conocimiento acerca del ambiente que lo rodea. El robot toma medidas de los diferentes sensores y extrae información de esas medidas. Posee un sensor LIDAR para sensar la habitación que limpia, sensores ultrasonidos para evitar obstáculos, sensores infrarrojos para evitar caídas, un sensor inercial para conocer la orientación, encoders para conocer la velocidad y posición de las ruedas, una cámara para capturar imágenes de la habitación y otra cámara para estimar posición mediante odometría visual. Finalmente, dispone de un sensor de voltaje para medir el nivel de la batería.

El sistema posee dos controladores: controlador principal y controlador esclavo. El controlador principal se encarga del control del proceso; es decir, procesar los algoritmos de construcción de mapa, localización, navegación y limpieza, así como también recibe información de algunos sensores como el LIDAR y las cámaras. El controlador esclavo se encarga de recibir las señales de los demás sensores y adaptarlas para ser información útil del controlador principal. Por otro lado, como producto del procesamiento de los algoritmos de navegación, el controlador principal genera señales de control de movimiento del robot que son enviadas al controlador esclavo. Este último, traduce estas señales y genera señales PWM que van hacia una interfaz de potencia que controla a los motores de desplazamiento, motores de las escobillas de limpieza y el motor del ventilador.

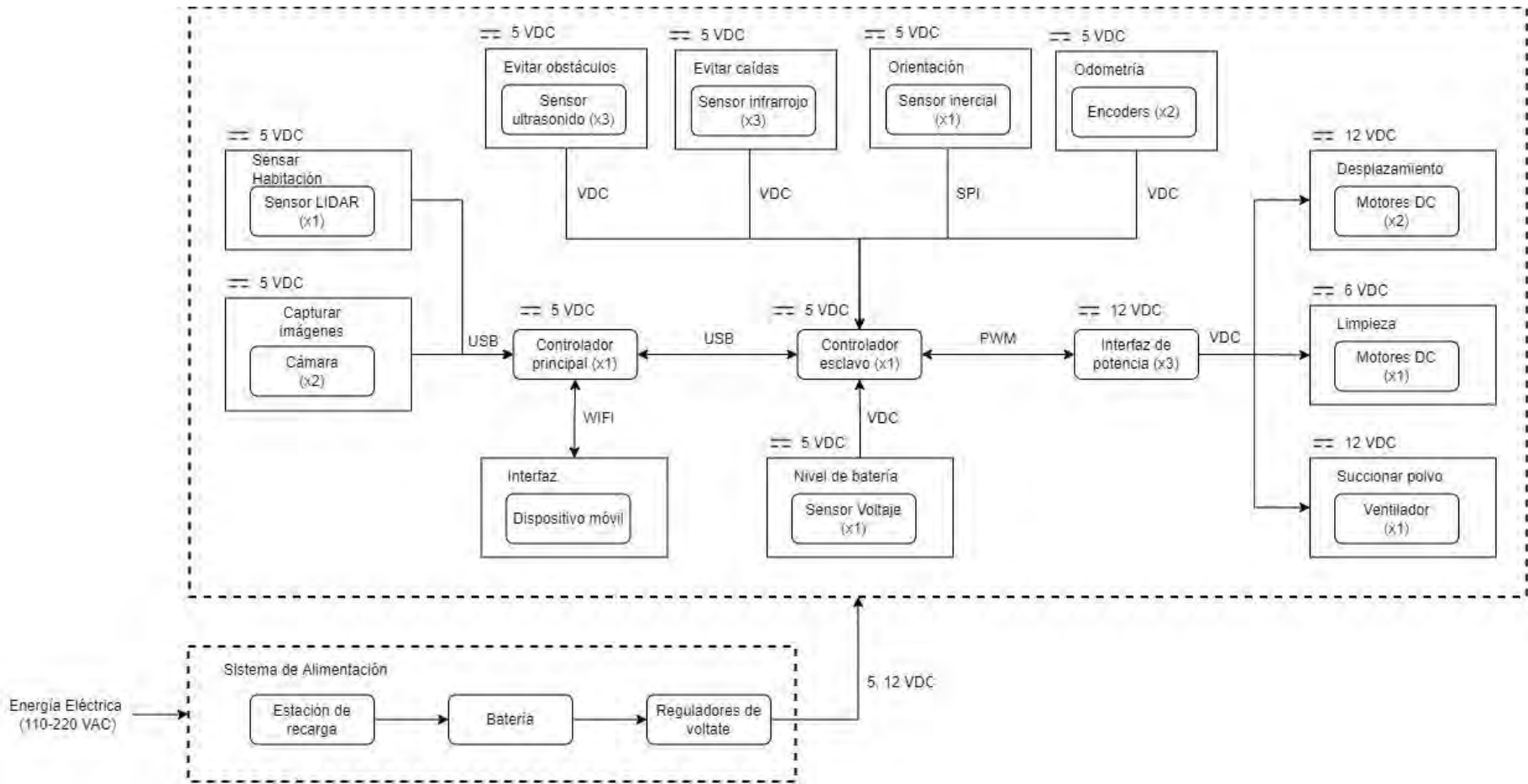


Figura 4.17: Diagrama de bloques del sistema

4.3.2 Componentes eléctricos y electrónicos

A continuación, se presentan los componentes eléctricos y electrónicos del sistema detallando sus principales características y parámetros de selección.

4.3.2.1 Controlador principal

El sistema requiere de un controlador principal que sea capaz de procesar información extraída de los sensores ultrasonidos, cámaras, LIDAR, infrarrojo, inercial y encoders; procesar los algoritmos de construcción de mapa de la habitación, localización, navegación y limpieza; y finalmente intercambiar información con la aplicación del usuario. De acuerdo con el diagrama de bloques, el controlador principal se comunica con el sensor LIDAR, dos cámaras y el controlador esclavo mediante protocolo USB. Además, se requiere que este controlador se pueda comunicar con la interfaz del usuario mediante comunicación WiFi. En la Tabla 4.1 se muestra los parámetros de selección del controlador principal.

Tabla 4.1: Parámetros de selección de controlador principal

Parámetro	Valor requerido
Núcleos	4
Velocidad	1 GHz
Memoria RAM	1 GB
Puertos USB	4

Se selecciona el sistema embebido Raspberry Pi 3 como controlador principal del sistema. Este embebido tiene un procesador ARMv8 con cuatro núcleos, una velocidad de procesamiento de 1.2 GHz, memoria RAM de 1GB, cuatro periféricos de comunicación USB, 40 pines de propósito general y capacidad de comunicarse de forma inalámbrica por protocolo WiFi. Para energizar este embebido se requiere de un voltaje de 5 V y una corriente mínima de 1 A. En la Tabla 4.2 se muestra las principales características de la Raspberry Pi 3.

Tabla 4.2: Características de Raspberry Pi 3¹⁰

Parámetro	Valor
Procesador	ARMv8
Núcleos	4
Memoria RAM	1 GB
Velocidad procesamiento	1.2 GHz
Pines multipropósito	40
Voltaje alimentación	5V
Corriente de consumo	1A



Figura 4.18: Raspberry Pi 3¹⁰

4.3.2.2 Controlador esclavo

Adicionalmente al controlador principal, el sistema requiere de un controlador esclavo con la finalidad de acondicionar las señales de los sensores ultrasónicos, infrarrojos, inercial y encoders, así como también para traducir las señales de control de movimiento del robot hacia los drivers de los motores. De acuerdo con el diagrama de bloques, el controlador esclavo se comunica con los sensores ultrasónicos por medio de señales VDC, con los sensores infrarrojo mediante señales VDC, con los encoders de los motores mediante señales VDC, con el sensor inercial mediante señales SPI y con el controlador principal mediante protocolo USB. En la Tabla 4.3 se muestra los parámetros de selección del controlador esclavo.

Tabla 4.3: Parámetros de selección de controlador esclavo

Parámetro	Valor requerido
Memoria Flash	256 KB
Frecuencia reloj	8 MHz
Comunicación	USB y SPI
Pines digitales	34

Se selecciona el microcontrolador Arduino Mega 2560 como controlador esclavo del sistema principalmente porque tiene 54 pines digitales, cantidad suficiente para interactuar los diversos sensores. Este Arduino está basado en el microcontrolador Atmega2560; tiene 54 pines de entrada/salida digital, de los cuales 15 pueden ser usados como salidas PWM; además tiene 16 pines analógicos. En la Tabla 4.4 se muestra las principales características del Arduino Mega 2560.

¹⁰ Raspberry Pi 3: www.raspberrypi.org

Tabla 4.4: Características de Arduino Mega 2560¹¹

Parámetro	Valor
Microcontrolador	Atmega2560
Frecuencia Reloj	16 MHz
Memoria Flash	256 KB
Memoria RAM	8KB
Pines digitales	54
Corriente máxima pin E/S	40 mA
Voltaje Operación	5V



Figura 4.19: Arduino Mega 2560¹¹

4.3.2.3 Sensor ultrasonido

Mientras el robot se desplaza dentro de la habitación, debe evitar colisionar con obstáculos; por ende, dispone de tres sensores ultrasonidos: uno ubicado en la parte central delantera y los otros dos ubicados en cada lado del robot. Estos sensores miden distancias desde el robot hacia los objetos que están a su alrededor. De acuerdo con el diagrama de bloques, los sensores ultrasonidos se comunican con el controlador esclavo por medio de señales VDC que corresponden a señales TTL. En la Tabla 4.5 se muestra los parámetros de selección del sensor ultrasonido.

Tabla 4.5: Parámetros de selección de sensor ultrasonido

Parámetro	Valor requerido
Rango	2cm – 1m
Frecuencia mínima	20 Hz
Ángulo mínimo	15°
Precisión mínima	10mm

Se selecciona el sensor ultrasonido HCSR 04 como sensores para evitar obstáculos debido principalmente a su buena resolución y frecuencia de lectura. Este sensor cuenta con cuatro pines: VCC, GND, TRIG y ECHO. Mediante el pin TRIG se envía una señal de activación y por el pin ECHO se recibe el eco del pulso emitido por el sensor. La distancia se calcula a través de la expresión $d = v * t$, en la cual “v” corresponde a la velocidad del sonido y “t” corresponde a la mitad del tiempo que transcurre entre el envío de la señal de activación y la recepción del eco del pulso emitido. En la Tabla 4.6 se muestra las principales características del sensor ultrasonido HCSR 04.

¹¹ Arduino: www.arduino.cc

Tabla 4.6: Características de sensor HCSR 04¹²

Parámetro	Valor
Rango	2cm – 4m
Resolución	3mm
Frecuencia de lectura	40 Hz
Ángulo de medición	15°
Voltaje de Operación	5V
Corriente de consumo	15 mA



Figura 4.20: Sensor HCSR 04¹²

4.3.2.4 Sensor infrarrojo

Mientras el robot se desplaza dentro de la habitación, además de evitar obstáculos, el robot debe detectar desniveles en la habitación para evitar caerse; por ello, el robot posee tres sensores infrarrojos: uno ubicado en la parte inferior delantera y los otros dos ubicados en cada lado del robot. Estos sensores miden distancias desde el robot hacia el piso. De acuerdo con el diagrama de bloques, los sensores infrarrojos envían señales VDC hacia el controlador esclavo. En la Tabla 4.7 se muestra los parámetros de selección del sensor infrarrojo.

Tabla 4.7: Parámetros de selección de sensor infrarrojo

Parámetro	Valor requerido
Rango	2cm – 10cm
Salida	Voltaje digital
Ángulo mínimo	15°

Se selecciona el sensor infrarrojo FC-51 como sensor para evitar caídas debido a su rango de 2cm – 30cm que permite detectar adecuadamente los espacios vacíos y a que su voltaje de salida es digital. El tener una salida digital indica que, al medir una distancia mayor a un umbral, el sensor genera un voltaje 3.3 VDC como salida, y si mide una distancia menor al umbral, genera 0 VDC. En la Tabla 4.8 se muestra las principales características del sensor ultrasonido HCSR 04.

¹² Sensor Ultrasonido: www.micropik.com

Tabla 4.8: Características de sensor FC-51¹³

Parámetro	Valor
Rango	2cm – 30cm
Salida	VDC
Ángulo de medición	35°
Voltaje de Operación	3-6V
Corriente de consumo	30 mA



Figura 4.21: Sensor FC-51¹³

4.3.2.5 Sensor inercial

En la estimación de la posición mediante odometría, se utiliza un sensor inercial con el objetivo de dar una corrección a dicha estimación y de esa forma disminuir el error acumulado. Este sensor mide la orientación e inclinación del robot. En particular, el sensor inercial que se requiere es uno que integre giroscopio y acelerómetro para poder obtener información de la aceleración y velocidad angular del robot. Según el diagrama de bloques, el sensor inercial se comunica con el controlador esclavo mediante protocolo SPI. En la Tabla 4.9 se muestra los parámetros de selección del sensor inercial.

Tabla 4.9: Parámetros de selección de sensor inercial

Parámetro	Valor requerido
Velocidad de rotación	180 °/s
Resolución de giro	0.5 °
Aceleración medible	1g

Se selecciona el sensor inercial MPU 6050 debido a que puede rotar a una velocidad máxima de 250°/s con una resolución de 0.2 °/s y una máxima aceleración medible de 2g. Este sensor integra un acelerómetro y un giroscopio. Este sensor se ubica en el soporte de componentes electrónicos debidamente calibrado. En la Tabla 4.10 se muestra las principales características del sensor inercial MPU 6050.

¹³ Sensor Infrarrojo: www.artofcircuits.com

Tabla 4.10: Características de sensor MPU 6050¹⁴

Parámetro	Valor
Velocidad rotación máxima	250 °/s
Resolución	0.2 °
Aceleración máxima	2g
Resolución ADC	16 bits
Voltaje Operación	5V
Corriente de consumo	40 mA



Figura 4.22: Sensor MPU 6050¹⁴

4.3.2.6 Sensor LIDAR

El robot limpiador en cada instante de su recorrido mide las distancias de los puntos que se encuentran a su alrededor para obtener información de la habitación. Esto le sirve para la construcción del mapa de la habitación y también para su localización. Con el objetivo de cumplir esta función, se utiliza un sensor LIDAR 2D, que es un sensor que mide distancias desde un emisor hacia un objeto o superficie mediante un haz de láser. Según el diagrama de bloques, el sensor LIDAR se comunica con el controlador principal mediante protocolo USB. En la Tabla 4.11 se muestra los parámetros de selección del sensor LIDAR.

Tabla 4.11: Parámetros de selección de sensor LIDAR

Parámetro	Valor requerido
Rango	20cm – 5m
Resolución	1mm
Ángulo de visión	360°
Frecuencia de muestreo	1 kHz

Se selecciona el sensor RP LIDAR A1 debido a que tiene un ángulo de visión de 360° en un campo 2D y cuenta con una adecuada frecuencia de muestreo de 2 kHz que permite obtener buena información de la habitación. Este sensor tiene un rango de visión desde 20 cm hasta 6 m con una resolución de 0.5 mm, un ángulo de visión de 360° ya que incluye un motor DC con frecuencia de rotación de 10 Hz. Se ubica en la parte superior del robot para que su campo de visión no sea afectado. En la Tabla 4.12 se muestra las principales características del sensor RP LIDAR A1.

¹⁴ Sensor Inercial: www.invensense.com

Tabla 4.12: Características de sensor RP LIDAR A1¹⁵

Parámetro	Valor
Rango	20cm – 6m
Resolución distancia	0.5 mm
Ángulo de visión	360°
Resolución angular	1°
Frecuencia de muestreo	2 kHz
Voltaje Operación	5V
Corriente de consumo	135 mA



Figura 4.23: Sensor RP LIDAR A1¹⁵

4.3.2.7 Cámara de captura de imágenes de habitación

Cuando el robot se desplaza sobre la habitación también puede recolectar información visual. Este tipo de información la utilizará para comprender mejor lo que se encuentra a su alrededor, por ejemplo, reconocer objetos específicos o interpretar sus movimientos. Entonces se utiliza una cámara matricial para capturar imágenes de la habitación. Según el diagrama de bloques, la cámara matricial se comunica con el controlador principal mediante protocolo USB. En la Tabla 4.13 se muestra los parámetros de selección de la cámara matricial.

Tabla 4.13: Parámetros de selección de cámara matricial

Parámetro	Valor requerido
Resolución	640x320
Tomas por segundo	30 FPS

Se sabe que el procesamiento de imágenes no depende exclusivamente del tipo de cámara que se usa, sino de la calidad de la imagen. Por ende, se selecciona la cámara matricial Logitech C920 debido a que tiene una resolución de 1920 x 1080 píxeles con una capacidad de tomar 30 tomas por segundo. Con esta cantidad de 30 FPS es posible detectar objetos en movimiento hasta una velocidad de 20 cm/s, velocidades superiores ocasionarían que la imagen resulte borrosa. En la Tabla 4.14 se muestra las principales características de la cámara Logitech C920.

¹⁵ Sensor LIDAR: www.slamtec.com

Tabla 4.14: Características de cámara Logitech C920¹⁶

Parámetro	Valor
Resolución	1920x1080 px
FPS	30
Voltaje de Operación	5V
Corriente de consumo	100mA



Figura 4.24: Cámara Logitech C920¹⁶

4.3.2.8 Cámara de odometría visual

En términos generales, odometría se refiere a estimar la posición del vehículo durante su recorrido. Esta estimación se puede realizar midiendo las rotaciones de las ruedas del robot. Sin embargo, medir solamente las rotaciones de las ruedas puede incurrir en tener un largo error acumulado de estimación debido principalmente a que las ruedas pueden deslizarse. Por tal motivo, surge la necesidad de optimizar la estimación de la posición, ya que, en la navegación autónoma, esto resulta ser un factor importante. Entonces, una alternativa eficaz es usar odometría visual, y recibe este término debido a que se dispone de una cámara acoplada al robot con la finalidad de construir la trayectoria recorrida del robot mediante el análisis de las imágenes en el tiempo. De acuerdo con el diagrama de bloques, la cámara de odometría visual se comunica con el controlador principal mediante protocolo USB. En la Tabla 4.15 se muestra los parámetros de selección de la cámara.

Tabla 4.15: Parámetros de selección de cámara odometría visual

Parámetro	Valor requerido
Resolución	320x240
Tomas por segundo	100 FPS

En el análisis de las imágenes en el tiempo para la estimación, se requiere que se tenga la mayor cantidad posible de imágenes en un rango de tiempo para tener una buena precisión y evitar tener pérdidas de información. Por lo tanto, se selecciona la cámara USBFHD01M debido a que puede tomar hasta 120 imágenes por segundo. En la Tabla 4.16 se muestra las principales características de la cámara Logitech USBFHD01M.

¹⁶ Cámara Logitech: www.logitech.com

Tabla 4.16: Características de cámara USBFHD01M¹⁷

Parámetro	Valor
Resolución	640x480 px
FPS	120
Voltaje de Operación	5V
Corriente de consumo	100mA



Figura 4.25: Cámara USBFHD01M¹⁷

4.3.2.9 Motores de Desplazamiento

El robot se desplaza mediante una configuración diferencial cuyo movimiento se basa en dos ruedas accionadas cada una por un motor. De acuerdo con el diagrama de bloques, cada motor es controlado por una interfaz de potencia que recibe señales PWM del controlador esclavo. En el anexo D se presentan los cálculos de los parámetros de selección de los motores de desplazamiento que se muestran en la Tabla 4.17.

Tabla 4.17: Parámetros de selección de motor de desplazamiento

Parámetro	Valor requerido
Torque	0.441 Nm
Velocidad	85 RPM

Según el torque y la velocidad de giro requeridos, se selecciona el motor Pololu 37Dx70L con caja reductora 70:1. Este motor viene integrado con un encoder de cuadratura que tiene una resolución de 64 cuentas por revolución. Este encoder posee dos canales que permiten conocer la velocidad y sentido de giro del motor. En la Tabla 4.18 se muestra sus principales características y en la Figura 4.27 se muestra su curva de funcionamiento.

Tabla 4.18: Características de motor Pololu¹⁸

Parámetro	Valor
Torque máximo	1.41 Nm
Velocidad sin carga	150 RPM
Reducción	70:1
Diámetro eje salida	6 mm
Resolución de Encoder	64 cuentas
Dimensiones	37Dx70L mm
Peso	0.22 kg
Voltaje de Operación	12 V
Corriente de libre giro	300 mA



Figura 4.26: Motor Pololu 37Dx70L¹⁸

¹⁷ Cámara USBFHD01M: www.elpccvtv.com

¹⁸ Motor Pololu: www.pololu.com

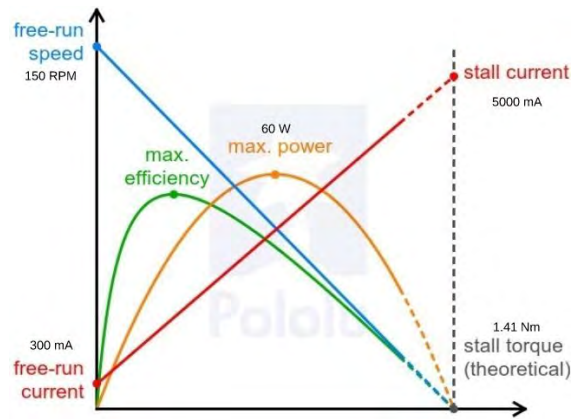


Figura 4.27: Curva de funcionamiento Motor Pololu 37Dx70L¹⁸

4.3.2.10 Motores de limpieza

El robot cuenta con dos escobillas laterales y una escobilla central que se encargan de atraer las partículas de suciedad que se encuentran alrededor del robot. Entonces, dispone de un motor por cada escobilla de limpieza. De acuerdo con el diagrama de bloques, cada motor es controlado por una interfaz de potencia que recibe señales PWM del controlador esclavo. En el anexo D se presentan los cálculos de los parámetros de selección de los motores de limpieza que se muestran en la en la Tabla 4.19.

Tabla 4.19: Parámetros de selección de motores de limpieza

Parámetro	Valor requerido
Motor de escobilla lateral	
Torque	0.1 oz-in
Velocidad	150 RPM
Motor de escobilla central	
Torque	0.8 oz-in
Velocidad	90 RPM

Según el torque y la velocidad de giro requerido por cada motor de limpieza, se selecciona el motor DFRobot de 6V con caja reductora 120:1. Debido a que este motor tiene un torque máximo de 2.7 oz-in y una velocidad de carga libre de 160 RPM, se opta por utilizar tres de estos motores: dos para las escobillas laterales y otro para la escobilla central. En la Tabla 4.20 se muestra las principales características de este motor.

Tabla 4.20: Características de motor DFRobot¹⁹

Parámetro	Valor
Torque máximo	2.7 oz-in
Velocidad sin carga	160 RPM
Reducción	120:1
Diámetro eje salida	5.4 mm
Peso	45 gramos
Voltaje de Operación	6 V
Corriente de libre giro	170 mA

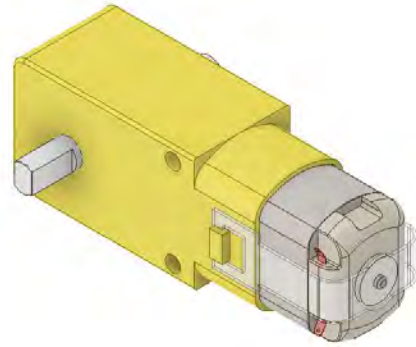


Figura 4.28: Motor DFRobot 6v¹⁹

4.3.2.11 Ventilador Centrífugo

El sistema dispone de un ventilador con la finalidad de generar un flujo de aire que transporte las partículas de suciedad. Existe varios tipos de ventiladores y una clasificación es según la trayectoria del flujo de aire, en la cual se dividen en tres grupos: ventiladores axiales, centrífugos y especiales. Para el sistema se opta por un ventilador centrífugo debido a que la dirección del aire cambia 90° en la salida con respecto a la dirección axial, lo cual permite ubicar al ventilador en una posición que ocupe menos espacio y de esta forma tener un diseño compacto. Además, este tipo de ventilador es el que genera menos ruido con respecto a los demás. En el anexo D se presentan los cálculos de los parámetros de selección del ventilador centrífugo que se muestran en la en la Tabla 4.21, donde además se especifica el volumen que recorre el flujo de aire dentro del robot.

Tabla 4.21: Parámetros de selección de ventilador

Parámetro	Valor requerido
Presión de aire	900 Pa
Flujo de aire	0.186 m ³ /min
Voltaje de Operación	12 V

Los parámetros mostrados corresponden a que, en primer lugar, la presión de aire corresponde a la capacidad de aspirar o succionar el aire que transporta las partículas de suciedad. Una presión a partir de 700 Pa es capaz de succionar relativamente bien el polvo del piso de una habitación²⁰, por lo que para tener un buen rendimiento se busca un ventilador que ejerza una presión de aire de 900 Pa ya que además también hay pérdida de presión en el recorrido. Luego es necesario que el flujo de aire se desplace con un

¹⁹ Motor DFRobot: www.dfrobot.com

²⁰ Ventilador centrífugo: www.delta-fan.com

caudal de 0.186 m³/min de acuerdo con el volumen de 3100 cm³ que se tiene que desplazar en el interior del robot (Anexo D). Finalmente, el voltaje de alimentación del ventilador centrífugo debe ser alrededor de 12V para evitar el uso de un convertidor DC-DC adicional.

Entonces de acuerdo con los parámetros requeridos, se selecciona el ventilador centrífugo Delta Electronics con número de serie BCB0812UHN con dimensiones 75 mm x 80 mm x 30 mm (longitud x altura x ancho) y álabes con diámetro 49 mm. Tiene un motor brushless que se energiza con 12 VDC y gira a una velocidad de 10800 RPM. El material de los álabes y de la carcasa es de plástico UL 94V-0 y su peso total es de 140g. En la Tabla 4.22 se muestra las principales características de este ventilador y en la Figura 4.30 se muestra la curva Presión vs Caudal del ventilador indicando su punto de operación.

Tabla 4.22: Características de ventilador²⁰

Parámetro	Valor
Máximo flujo de aire	0.854 m ³ /min
Máxima presión de aire	912 Pa
Velocidad máxima de giro	10800 RPM
Ruido	69.3 dB
Voltaje de Operación	6-13.2 V
Corriente de consumo	2.3 A



Figura 4.29: Ventilador BCB²⁰

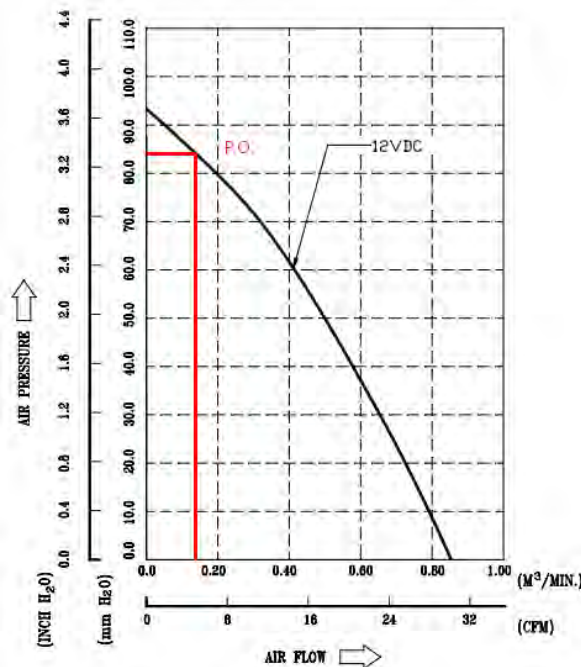


Figura 4.30: Curva Presión vs Caudal²⁰

²⁰ Ventilador centrífugo: www.delta-fan.com

4.3.2.12 Controlador de motores

- **Controlador de motores de desplazamiento**

Como resultado del procesamiento de los algoritmos de navegación, el controlador principal genera señales de control de movimiento del robot y son enviadas hacia el controlador esclavo. El controlador esclavo convierte estas señales de control en respectivas velocidades para cada motor de desplazamiento a través de señales PWM. Estas señales necesitan de una interfaz de potencia que genere los respectivos voltajes y así accionar a los motores de desplazamiento. En la Tabla 4.23 se muestra los parámetros de selección del controlador de motores de desplazamiento.

Tabla 4.23: Parámetros de selección de control de motores de desplazamiento

Parámetro	Valor requerido
Número de canales	2
Corriente máxima por canal	5A
Voltaje de alimentación	12V

Los parámetros mostrados corresponden a que, en primer lugar, el controlador debe manejar los dos motores al mismo tiempo. Luego debido a que la corriente máxima que puede consumir cada motor es 5A, entonces cada canal del controlador debe ser capaz de soportar este consumo. Por último, el voltaje de alimentación del controlador de motores debe ser alrededor de los 12V para evitar el uso de un convertidor DC-DC adicional. Entonces de acuerdo a los parámetros requeridos, se selecciona el controlador de motores Pololu VNH5019 debido a que posee dos canales y cada canal puede proveer una corriente continua en el rango de 0-12A y, en caso crítico, soportar un pico de corriente de hasta 30A por 10 segundos sin recalentamiento ni daños. En la Tabla 4.24 se muestra las principales características de este controlador de motores.

Tabla 4.24: Características de VNH 5019²¹

Parámetro	Valor
Corriente continua por canal	0 - 12A
Corriente pico por canal	30A
Voltaje de control de entrada	3.3 o 5V
Frecuencia señal PWM	Hasta 20KHz
Voltaje de Operación	5.5 – 24V



Figura 4.31: Controlador VNH 5019²¹

²¹ Driver VN5019: www.pololu.com

- **Controlador de motor de ventilador**

El ventilador centrífugo tiene un motor que gira a los álabes y de esta forma genera el flujo de aire. El caudal de este flujo tiene una relación proporcional a la velocidad de giro de los álabes: a mayor velocidad de giro del motor corresponde un mayor caudal de flujo de aire. Luego se apreció en la curva Presión vs Caudal del ventilador que a un determinado caudal corresponde una respectiva presión de aire. Entonces se aprecia cómo se interrelacionan estas variables y que dependen inicialmente de la velocidad de giro del motor. Por lo tanto, se necesita de un controlador para el motor del ventilador para poder variar la velocidad de giro de los álabes y trabajar en el punto de operación adecuado. En la Tabla 4.25 se muestra los parámetros de selección del controlador del motor del ventilador para elegir el óptimo.

Tabla 4.25: Parámetros de selección de controlador de ventilador

Parámetro	Valor requerido
Número de canales	1
Corriente máxima por canal	3A
Voltaje de alimentación	12V

La corriente máxima por canal corresponde a que el motor del ventilador puede consumir hasta un pico de 3A y el voltaje de alimentación del controlador de motores debe ser alrededor de los 12V para evitar el uso de un convertidor DC-DC adicional. Entonces de acuerdo a los parámetros requeridos, se selecciona el controlador de motores Pololu VNH5019 debido a que cada canal puede proveer una corriente continua en el rango de 0-12A y, en caso crítico, soportar un pico de corriente de hasta 30A por 10 segundos sin recalentamiento ni daños. Se aprecia que este controlador tiene dos canales, un canal se utiliza para controlar el motor del ventilador y otro canal servirá para poder controlar otro motor que requiera el sistema. Finalmente, en la Tabla 4.26 se muestra cómo la variación del ciclo de trabajo de la señal PWM que llega hacia este controlador afecta a la velocidad del motor y a la corriente de consumo.

Tabla 4.26: Variación de señal PWM vs velocidad

Ciclo de trabajo (%)	Velocidad (RPM)	Corriente consumo (A)
100	10800	2.3
60	6000	0.49
20	2000	0.11

- **Controlador de motores de limpieza**

El robot dispone de tres escobillas de limpieza y cada una será accionada por un motor DC. Entonces resulta necesario controlar la velocidad de giro de cada motor ya que afecta directamente a la atracción de polvo en la limpieza. Por tanto, se requiere de un controlador para estos motores que genere los respectivos voltajes de potencia de acuerdo con las señales PWM del controlador esclavo y así accionar a los motores de limpieza. Inicialmente se tendría que conseguir un controlador con tres canales; sin embargo, anteriormente se seleccionó un controlador VNH5019 para controlar el motor del ventilador y quedó un canal libre. Este canal libre se utilizará para controlar el motor de la escobilla central. Se verifica que este controlador VNH5019 funciona adecuadamente para este motor, ya que si bien es cierto que el voltaje de salida de potencia de este controlador varía de 0-12V y que el motor de la escobilla central recibe un voltaje nominal de 6V, con una señal PWM que varíe su ciclo de trabajo en el rango de 0-50% estaría controlando adecuadamente a este motor.

Luego se requiere un controlador para los dos motores de las escobillas laterales. En la Tabla 4.27 se muestra los parámetros de selección del controlador de motores de limpieza que permite seleccionar el óptimo.

Tabla 4.27: Parámetros de selección de control de motores de limpieza

Parámetro	Valor requerido
Número de canales	2
Corriente máxima por canal	2A
Voltaje de alimentación	12V

Los parámetros mostrados corresponden a que debido a que la corriente máxima que puede consumir cada motor es 2A, entonces cada canal del controlador debe ser capaz de soportar este consumo. Luego el controlador debe ser capaz de energizarse con voltaje de alimentación de 12V para evitar el uso de un convertidor DC-DC adicional. Entonces de acuerdo a los parámetros requeridos, se selecciona el controlador de motores Dual L298 H-Bridge debido a que posee dos canales y cada canal puede proveer una corriente continua en el rango de 0-2A. En la Tabla 4.28 se muestra las principales características de este controlador de motores para las escobillas laterales.

Tabla 4.28: Características de Dual L298 H-Bridge²²

Parámetro	Valor
Corriente continua por canal	0 - 2A
Corriente pico por canal	2A
Voltaje de control de entrada	3.3 o 5V
Voltaje de Operación	5 – 35V
Potencia máxima	25W



Figura 4.32: Dual L298 H-Bridge²²

4.3.2.13 Sistema de alimentación del sistema

El sistema de alimentación tiene la función de suministrar energía a todos los componentes del robot. Este sistema está conformado por una Fuente AC, una Fuente DC, una batería y reguladores de voltaje como se muestra en la Figura 4.33.

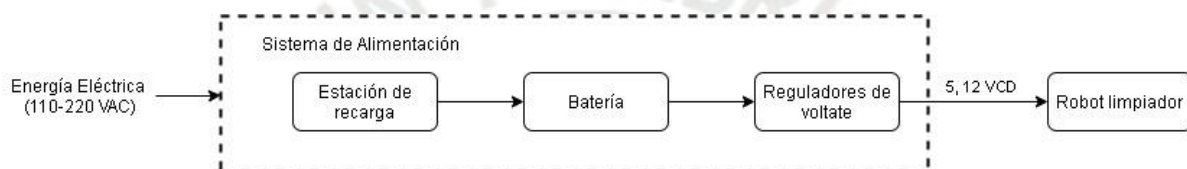


Figura 4.33: Sistema de alimentación

En primer lugar, el sistema recibe energía eléctrica alterna 110-220 VAC (50-60 Hz) desde un tomacorriente. Luego esta energía eléctrica AC es transformada en energía continua a través de la estación de recarga. Posteriormente, esta energía continua es almacenada en una batería. Finalmente, la energía de la batería es regulada para que energizar correctamente a los componentes eléctricos y electrónicos del robot limpiador.

- **Batería**

En la Tabla 4.29 se muestra los parámetros de selección de la batería que permiten seleccionar la óptima. Los cálculos de estos parámetros de selección se encuentran en el Anexo D de la presente tesis.

Tabla 4.29: Parámetros de selección de batería

Parámetro	Valor requerido
Capacidad	10 Ah
Voltaje	12V
Corriente de carga	0.5 A
Corriente de descarga	3 A

²² Driver Dual L298 H-Bridge: www.robotshop.com

Los parámetros mostrados corresponden a que, en primer lugar, el robot debe ser capaz de moverse como mínimo una hora y se opta a que el robot pueda moverse aproximadamente dos horas, por lo que la capacidad de la batería resulta 10 Ah (Anexo D). Luego el voltaje nominal de la batería debe ser 12V para evitar el uso de convertidores DC-DC para energizar los motores. Debe soportar una corriente de carga de 0.5 A, que equivale a 5Ah, para que el tiempo de carga de la batería no supere las 2 horas. Finalmente, debe ser capaz de suministrar una corriente de descarga hasta 30A, que es la corriente pico que puede consumir el controlador de motores VNH5019. Entonces de acuerdo con los parámetros requeridos, se selecciona la Batería Li-Ion 18650 ya que tiene una capacidad de 10 Ah y voltaje nominal de 12V. En la Tabla 4.30 se muestra las principales características de esta batería.

Tabla 4.30: Características de batería Li-Ion 18650²³

Parámetro	Valor
Voltaje nominal	12 V
Capacidad	10 Ah
Corriente de descarga	0.2 – 3A
Corriente de carga	0.2 – 1A
Voltaje de corte en descarga	2.75V



Figura 4.34: Batería Li-Ion 18650²³

- **Fuente DC y cargador**

Una vez definida la batería del robot, se procede a seleccionar la Fuente DC que abastecerá de energía a la batería. Esta Fuente DC se encarga de transformar la energía eléctrica alterna, que proviene de la Fuente AC, en energía continua para recargar a la batería. En la Tabla 4.31 se muestra los parámetros de selección de la Fuente DC que permiten seleccionar el óptimo según los cálculos detallados en el Anexo D.

Tabla 4.31: Parámetros de selección de Fuente DC

Parámetro	Valor requerido
Voltaje de entrada	220VAC-60Hz
Voltaje de salida	12V
Corriente de salida	5A

Los parámetros de la Tabla 4.31 corresponden primeramente a que el cargador de la batería pueda conectarse en cualquier tomacorriente de la habitación a limpiar. Luego de

²³ B atería Li-Ion 18650: www.gebattery.com.cn

transformar la energía alterna en energía continua, debe tener como salida un voltaje de 12V que corresponde al voltaje de la batería. Adicionalmente, como la batería tiene una capacidad de 10Ah, entonces la corriente que el cargador debe suministrar en la salida deberá ser aproximadamente 5A para que el tiempo de recarga de la batería sea dos horas. Entonces de acuerdo con los parámetros requeridos, se selecciona la Fuente DC Tenergy debido principalmente a que genera energía con voltaje 12 V y corriente 5A. En la Tabla 4.32 se muestra las principales características de esta Fuente DC. La fuente DC Tenergy cuenta con un conector standard macho de 2.5 mm x 5.5 mm que se acopla a la batería mediante su par análogo, es decir un conector hembra que se conecta a los cables de la batería Li-Ion 18650.

Tabla 4.32: Características de Fuente DC Tenergy²⁴

Parámetro	Valor
Voltaje de entrada	100-240 VAC
Voltaje de salida	12 V
Corriente de salida	5A
Dimensiones	115x50x34 mm



Figura 4.35: Fuente DC Tenergy²⁴

- **Reguladores de voltaje**

Definidos la Fuente DC y la batería, resta seleccionar los reguladores de voltaje necesarios para energizar adecuadamente a los componentes eléctricos y electrónicos del robot limpiador. De acuerdo con los voltajes de operación de los componentes eléctricos y electrónicos descritos anteriormente, se requiere reguladores de voltaje para energizar los siguientes componentes: controlador principal, controlador esclavo y sensor LIDAR. El sensor LIDAR no puede ser energizado mediante el controlador esclavo ya que posee un motor. El voltaje de operación y consumo de corriente de cada componente indicado se muestra en la siguiente tabla:

Tabla 4.33: Componentes que requieren regulador de voltaje

Componente	Voltaje de operación	Corriente de consumo
Controlador principal	5 V	2 A
Controlador esclavo	5 V	200 mA
Sensor LIDAR	5 V	135 mA
	Consumo total	2.335 A

²⁴ Fuente DC Tenergy: www.all-battery.com

Entonces se requiere de un regulador de voltaje que convierta el voltaje 12V de la batería a 5V y que suministre una corriente de 2.335A. De acuerdo con estos parámetros requeridos, se selecciona el regulador de voltaje LM2596 principalmente debido a que puede suministrar corriente hasta 3A. En la Tabla 4.34 se muestra las principales características de este regulador de voltaje.

Tabla 4.34: Características de regulador LM2596²⁵

Parámetro	Valor
Voltaje de entrada	4-35 V
Voltaje de salida ajustable	1.2-30 V
Corriente máxima de salida	3A
Eficiencia de conversión	92%
Dimensiones	66x39x18 mm



Figura 4.36: Regulador LM2596²⁵

- **Sensor de nivel de batería**

Finalmente, se necesita sensar constantemente el nivel de batería para que cuando presente un nivel bajo, el robot se dirija hacia su estación de recarga. Asimismo, se requiere medir el voltaje cuando la batería está completamente cargada. Para esto, el sistema requiere de un sensor de voltaje. En la Tabla 4.35 se muestra los parámetros de selección del sensor de voltaje que permiten seleccionar el óptimo.

Tabla 4.35: Parámetros de selección de sensor de voltaje

Parámetro	Valor requerido
Voltaje de entrada	0 – 15V
Resolución	0.1V
Compatibilidad	Arduino
Voltaje de salida	3.3 – 5V

Los parámetros de la Tabla 4.35 corresponden primeramente a que el sensor de voltaje pueda sensar el nivel de voltaje de la batería, cuyo nivel máximo podría llegar en caso extremo a 15V. Luego se requiere que tenga una resolución de 0.1V y que sea compatible

²⁵ Regulador LM2596: www.itead.cc

con el microcontrolador Arduino, por lo que su voltaje de salida debe estar en el rango de 3.3 V a 5 V. Entonces de acuerdo con los parámetros requeridos, se selecciona el sensor de voltaje FZ0430. En la Tabla 4.36 se muestra las principales características de este sensor de voltaje.

Tabla 4.36: Características de sensor de voltaje²⁶

Parámetro	Valor
Voltaje de entrada	0 – 25V
Resolución	0.02V
Compatibilidad	Arduino
Voltaje de salida	3.3 – 5V



Figura 4.37: Sensor de voltaje FZ0430²⁶



²⁶ Sensor de voltaje: www.tinkbox.ph

4.4 Sistema de control

En esta sección, en primer lugar, se presenta la arquitectura de control del sistema y posteriormente se presenta la lógica de control del sistema que incluye el diagrama de flujo general, así como diagramas de flujo de los procesos involucrados. Asimismo, se presenta las estrategias de control del sistema.

4.4.1 Funciones para el control del robot

En un nivel de abstracción, un robot cumple fundamentalmente tres funciones: (1) sensar, (2) decidir y (3) actuar. Un robot es una máquina que puede sensar su entorno, utilizar esta información extraída para decidir qué acción debe realizar y después actuar sobre el entorno. Con el objetivo de tener un sistema robusto, es importante que el robot complete este lazo de forma cerrada, de tal forma que le permita tener una retroalimentación y corregir los errores que se pueden presentar.



Figura 4.38: Funciones para el control del robot

En la Figura 4.38 se muestra las funciones involucradas para el control del robot. El robot dispone de seis tipos de sensores con la finalidad de sensar su entorno que lo rodea: LIDAR, cámaras, ultrasonidos, infrarrojos, inercial y encoders. A través de estos sensores el robot recibe información en bruto, es decir, una gran cantidad de medidas, por lo que es necesario que interprete esta información a través de algoritmos de fusión de sensores. El algoritmo que se usa es el Filtro de Kalman Extendido, que primero agrupa las medidas del LIDAR y de la cámara para obtener medidas de posición, luego agrupa las medidas de los encoders y el sensor inercial para obtener medidas de odometría y finalmente agrupa las medidas de los ultrasonido e infrarrojos para obtener distancias hacia obstáculos y vacíos. El robot utiliza esta información para localizarse y mapear simultáneamente con la finalidad de construir un mapa del entorno (SLAM). En dicho SLAM, se dispone de un algoritmo de optimización de Filtro de Partículas y de un

algoritmo de mapeo de ocupación de celdas que trabajan en conjunto. Como resultado se obtiene el mapa de la habitación.

Posteriormente, el robot ya puede planear la trayectoria para disponerse a limpiar la habitación. Luego el robot se desplaza sobre la habitación y la va limpiando de partículas de suciedad. Durante el recorrido el robot es capaz de controlar la trayectoria que viene realizando con la finalidad de saber el progreso y cuánto le falta para terminar el recorrido de limpieza.

Nótese que los algoritmos de aprendizaje de máquina se proponen como mejora en la parte de localización y mapeamiento simultáneo (SLAM), dado que el robot va a ser capaz de detectar objetos como puertas, ventanas, etc. y usarlos como landmarks y de esta forma ayudar a corregir los errores acumulados que se tenga.

4.4.2 Diagramas de flujo

- **Diagrama de flujo general**

En la Figura 4.39 se muestra el diagrama de flujo de general que permite programar el controlador principal del robot limpiador. En este diagrama se aprecia que el proceso inicia configurando los puertos y variables de comunicación. Luego el robot inicia su tarea posicionándose en un lugar totalmente desconocido. Entonces lo primero que debe realizar es la localización y mapeamiento simultáneo y de esta forma construir el mapa de la habitación.

Cuando el robot ya tiene construido el mapa, se dispone a desplazarse sobre la habitación con la finalidad de limpiarla. Luego el sistema divide el mapa de la habitación en zonas o células, para que después el robot vaya limpiando zona por zona. Seguidamente, el robot planea la trayectoria de recorrido y luego se desplaza y va limpiando la habitación. En cada instante de recorrido de limpieza, el robot va midiendo su nivel de batería. En caso de presentar un nivel bajo de batería, el robot se dirige a la estación de recarga, espera a que la batería se complete de cargar y se dirige nuevamente al lugar donde se quedó para continuar con la limpieza. Finalmente, de esta forma el robot recorre cada zona de la habitación hasta que termina de limpiarla.

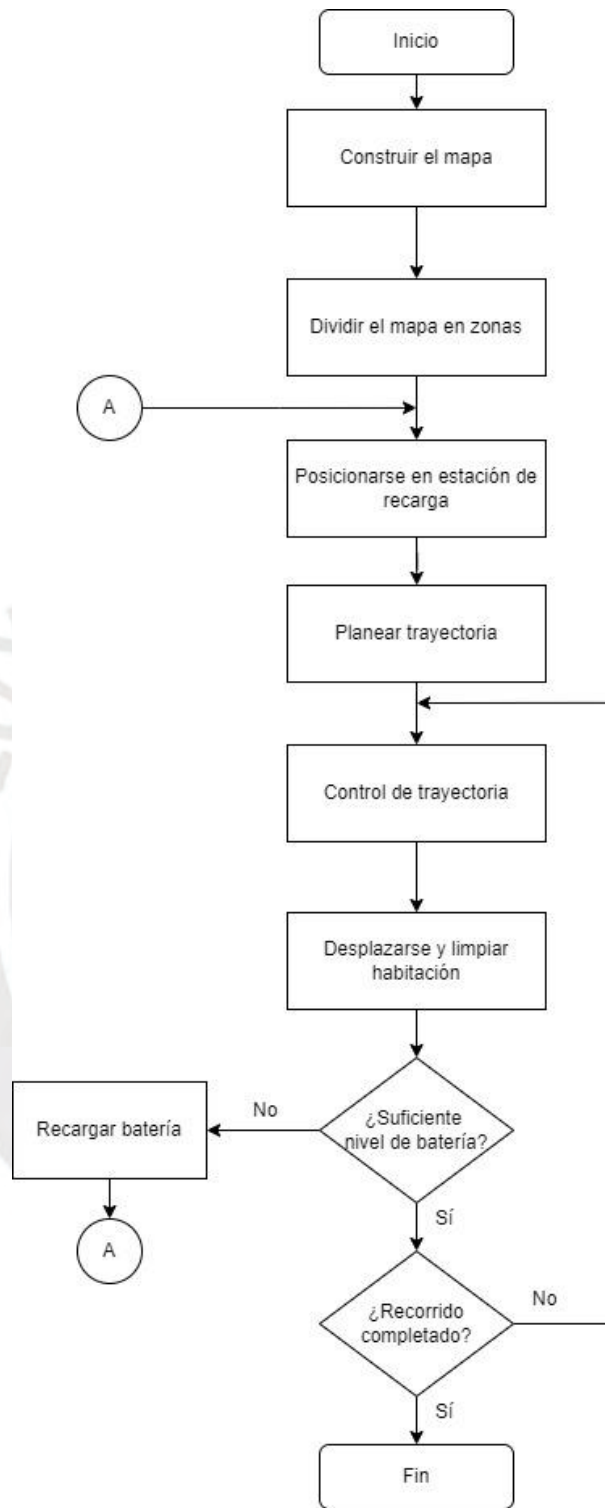


Figura 4.39: Diagrama de flujo general del sistema

- **Diagrama de flujo de proceso de localización**

En la Figura 4.40 se muestra el diagrama de flujo de proceso de localización en el mapa, que se realiza de manera simultánea con el mapeamiento. El robot inicia en una posición arbitraria desconocida y el objetivo es conocer su posición. Entonces el robot mide distancias usando el sensor LIDAR y procesa estas medidas mediante el algoritmo de optimización de Filtro de Partículas que permite estimar la posición en el mapa. Es necesario utilizar más sensores debido a que se sabe que un sensor como el LIDAR tiene un grado de incerteza que se va acumulando en el tiempo. Entonces al mismo tiempo el sistema obtiene información de odometría y también estima la posición dentro del mapa. Entonces estas dos fuentes de estimación de posición son fusionadas para obtener una estimación más precisa. Asimismo, se utiliza el algoritmo de aprendizaje de máquina para corregir la previa estimación. Mediante el algoritmo de aprendizaje de máquina el robot detecta objetos como puertas, ventanas, etc. y los usa como landmarks para de esta forma ayudar a corregir los errores acumulados que se tenga. De esta forma el robot es capaz de localizarse en el mapa.

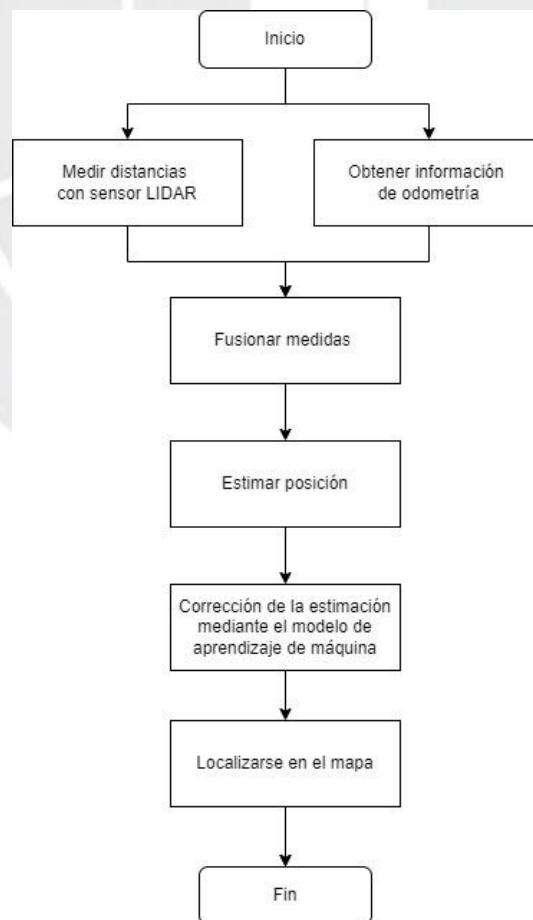


Figura 4.400: Diagrama de flujo de proceso de localización en el mapa

- **Diagrama de flujo de proceso de construcción de mapa**

En la Figura 4.41 se muestra el diagrama de flujo de proceso de construcción de mapa, que corresponde al primer proceso del diagrama de flujo general. Para ello se utiliza el algoritmo de mapeamiento por ocupación de celdas. Este proceso comienza con el robot colocado en una posición arbitraria. Entonces el robot mide distancias usando el sensor LIDAR. Estas distancias son asociadas a la posición actual del robot para que se relacionen con el mapa y de esta forma se actualice. Luego el robot se desplaza para seguir construyendo el mapa. La estimación de la posición es obtenida por medio del proceso de localización del sistema para que en todo instante se conozca la posición del robot utilizando los sensores LIDAR, cámaras, encoders y sensor inercial. Finalmente, este ciclo se repite hasta tener el mapa completo.

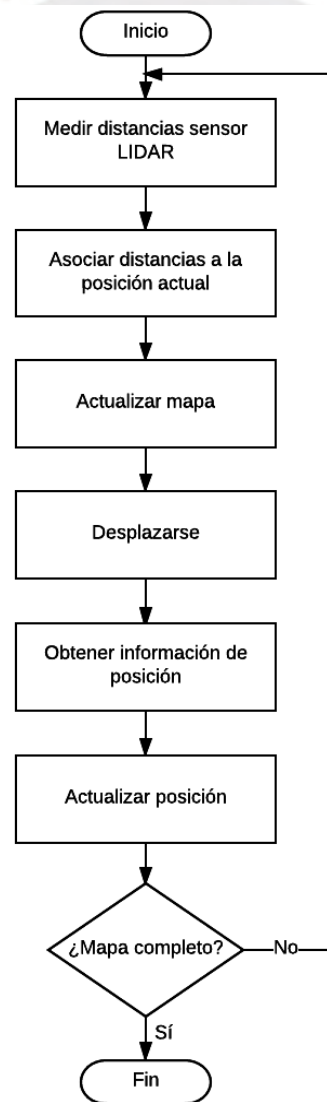


Figura 4.411: Diagrama de flujo de proceso de construcción de mapa

- **Diagrama de flujo de proceso de planeación de trayectoria**

En la Figura 4.42 se muestra el diagrama de flujo de proceso planeación de trayectoria. Este proceso inicia obteniéndose la posición inicial y la posición deseada a la que el robot quiere alcanzar. De acuerdo con estas posiciones, se procede a calcular la trayectoria con la distancia mínima para llegar a la posición deseada. Una vez calculada dicha trayectoria, el robot comienza a desplazarse. No obstante, durante la trayectoria puede aparecer un nuevo obstáculo y el robot debe ser capaz de evitar colisionar con dicho obstáculo. Entonces en caso de que aparezca un obstáculo, el robot nuevamente procede a calcular la trayectoria evitando el obstáculo. Por otro lado, mientras no aparezcan obstáculos nuevos, el robot se desplaza hasta llegar a la posición deseada.

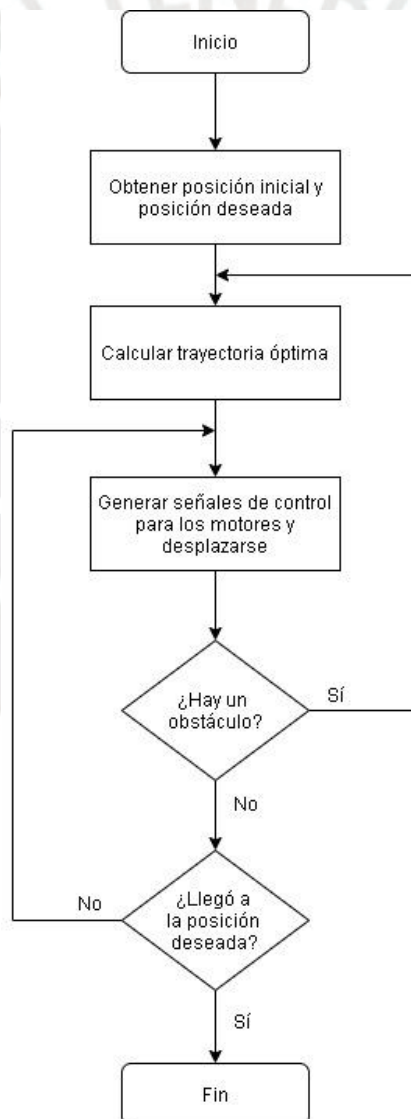


Figura 4.42: Diagrama de flujo de proceso de planeación de trayectoria

- **Diagrama de flujo de proceso de limpieza de habitación**

En la Figura 4.43 se muestra el diagrama de flujo de proceso limpieza de la habitación. Este proceso inicia atrayendo partículas de suciedad hacia el robot mediante el accionamiento de las escobillas. Seguidamente, se succiona el aire a través del ventilador que genera un flujo de aire que conduce las partículas de suciedad. Luego se filtra el flujo de aire que ingresa al robot para separar el polvo y posteriormente almacenarlo.

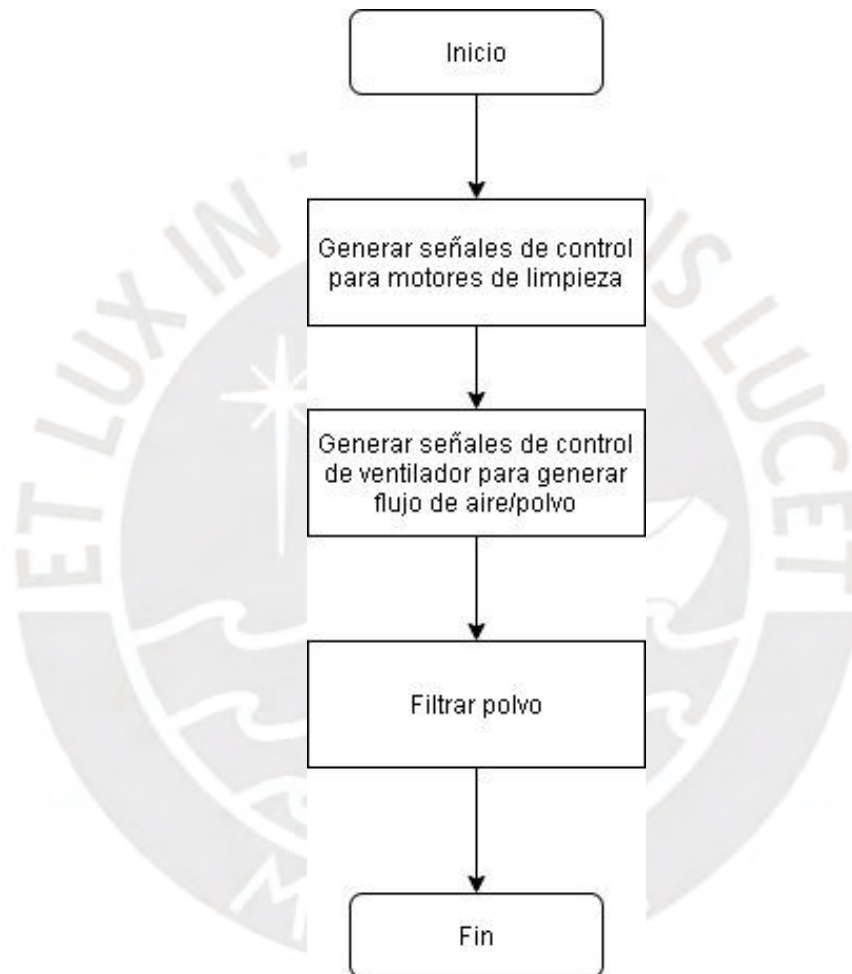


Figura 4.43: Diagrama de flujo de proceso de limpieza de la habitación

- **Diagrama de flujo de proceso de recarga de batería**

En la Figura 4.44 se muestra el diagrama de flujo del proceso de recarga de batería. Este proceso inicia cuando le llega una interrupción al controlador principal que indica que hay un nivel bajo de batería. Entonces el robot procede a guardar la última posición donde se quedó limpiando. Seguidamente, se desplaza hacia la estación de recarga. Una ventaja de poseer una estación de recarga es que esta estación tiene una ubicación conocida en el mapa. Cuando el robot llega hacia esta ubicación, posiblemente tiene errores acumulados de estimación de posición; sin embargo, al conocerse la posición de la estación, entonces permite corregir los errores. Luego el robot espera hasta que la batería este completamente cargada. Finalmente, el robot se desplaza hacia la última posición guardada para continuar con su tarea de limpieza.



Figura 4.44: Diagrama de flujo de proceso de recarga de batería

4.4.3 Estrategias de control

- **Estrategia de control de motores de desplazamiento**

De acuerdo con la planeación de trayectoria, el controlador principal genera señales de control de los motores de desplazamiento. Estas señales son enviadas al controlador esclavo para que genere señales PWM que van hacia una interfaz de potencia que controla a los motores de desplazamiento. Entonces debe existir un lazo de control en el controlador esclavo para manejar a los motores de las ruedas.

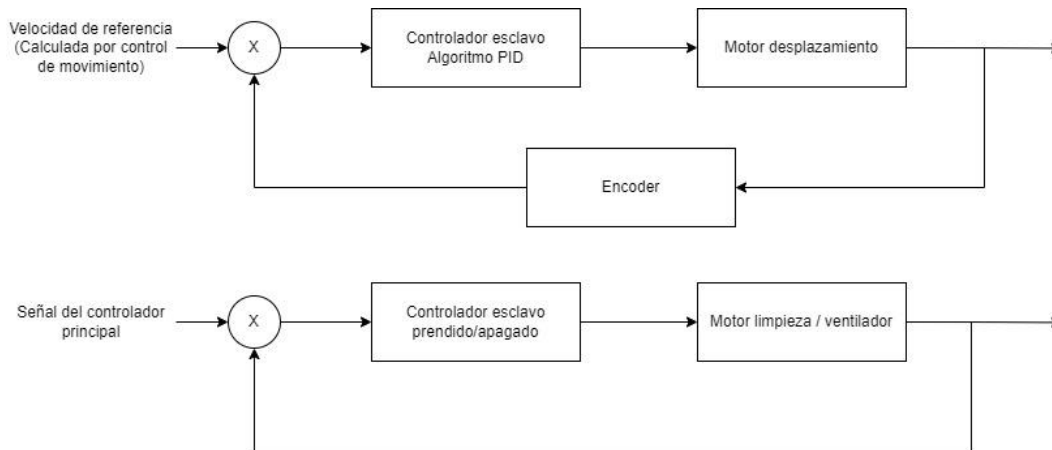


Figura 4.45: Lazo de control de motores de desplazamiento

En la Figura 4.45 se muestra el lazo de control de los motores de desplazamiento y los motores de limpieza. La velocidad de referencia proviene del controlador principal como producto de la planeación de trayectoria. Entonces el controlador esclavo tiene implementado un algoritmo PID que tiene como entrada el error entre la velocidad de referencia y la velocidad sensada por el encoder del motor. El algoritmo genera señales PWM que van hacia la planta, que en este caso está conformada por la interfaz de potencia junto con el motor de desplazamiento.

- **Odometría**

Generalmente, la posición del robot es estimada mediante odometría. Ahora existen varias fuentes de odometría. La comúnmente usada es odometría por medio de encoders en las ruedas del robot. Sin embargo, usar solo encoders no es conveniente, ya que rápidamente acumula errores. Esta odometría se mejora mediante el uso de un sensor inercial (IMU) que mide la orientación y aceleración del robot en tres ejes. No obstante, es recomendable mejorar aún más esta última combinación encoder-IMU. Se puede añadir otra fuente de odometría: visual. Mediante una cámara que mide variaciones de patrones es posible obtener variaciones de posición. Luego es necesario fusionar las fuentes de odometría, por lo que se usa un Filtro de Kalman Extendido (*EKF-filter*).

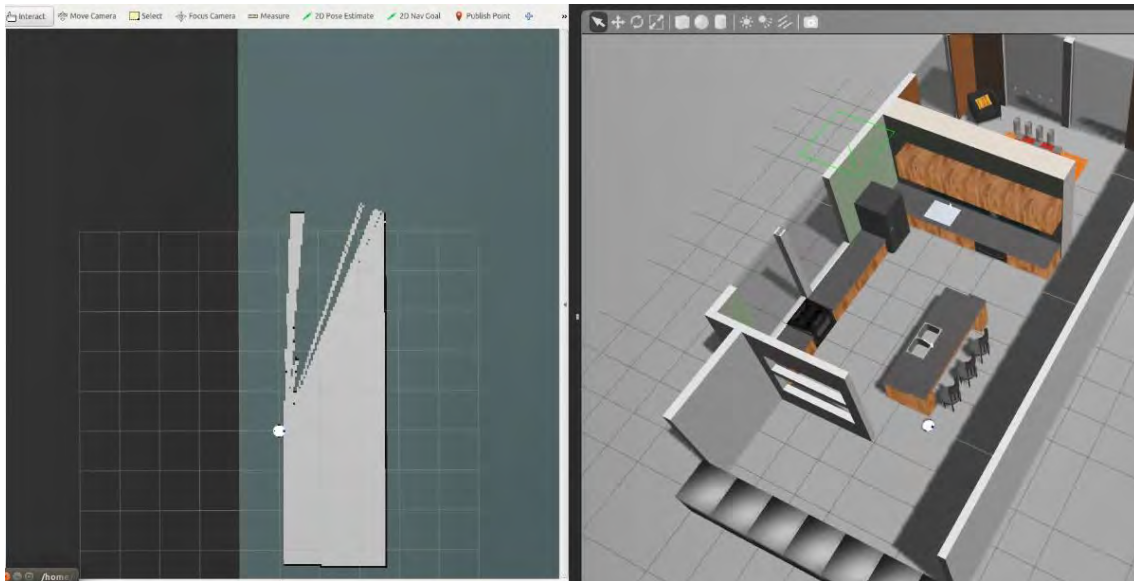
CAPÍTULO 5

SIMULACIONES Y RESULTADOS

En este capítulo se presentan simulaciones de los procesos que el robot limpiador cumple en su objetivo de navegar autónomamente en una habitación y realizar un recorrido que le permita limpiar las partículas de suciedad presentes en dicha habitación. Estas simulaciones se realizan utilizando la plataforma ROS (*Robot Operating System*). ROS es una plataforma de código abierto para el desarrollo de aplicaciones robóticas que tiene características como envío de mensajes, computación distribuida y reutilización de código, lo que permite administrar diversos hardware integrados con sensores y actuadores junto con paquetes de programas [Joseph, 2015].

5.1 Simulación de localización y mapeamiento simultáneo

El robot limpiador inicia su misión en una posición arbitraria de una habitación totalmente desconocida. Con el objetivo de que el robot pueda navegar autónomamente en dicha habitación, lo primero que debe realizar es construir un mapa de la habitación y poder localizarse dentro del mapa. Para esto, el robot realiza el proceso de localización y mapeamiento simultáneo. En la localización, el robot utiliza adicionalmente un algoritmo de optimización de filtro de partículas en conjunto con la información de la odometría proveniente de los encoders y el sensor inercial. Con respecto al mapeamiento, utiliza el algoritmo de mapeamiento de ocupación de celdas. En la Figura 5.1 (b) se muestra el robot limpiador ubicado dentro de una habitación conformada de una sala que colinda con un comedor. En la Figura 5.1 (a) se aprecia el inicio de la construcción del mapa de la habitación.



(a)

(b)

Figura 5.1: Inicio de proceso de localización y mapeamiento simultáneo

Luego el robot se desplaza sobre la habitación con la finalidad de obtener información del entorno e ir construyendo el mapa. Esto se aprecia en la Figura 5.2, la cual muestra que el robot se ha desplazado sobre la habitación y el mapa se ha ido actualizando.

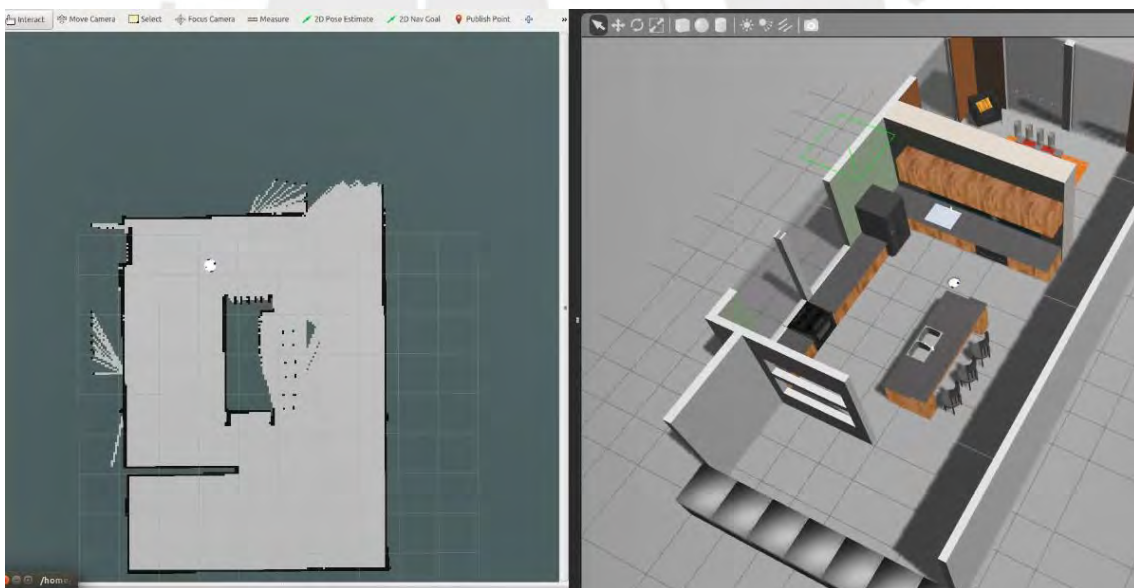


Figura 5.2: Proceso de construcción de mapa de habitación

Posteriormente, el robot culmina este proceso al recorrer todas las áreas de la habitación lo cual le permite terminar de construir el mapa geométrico de la habitación como se muestra en la Figura 5.3.

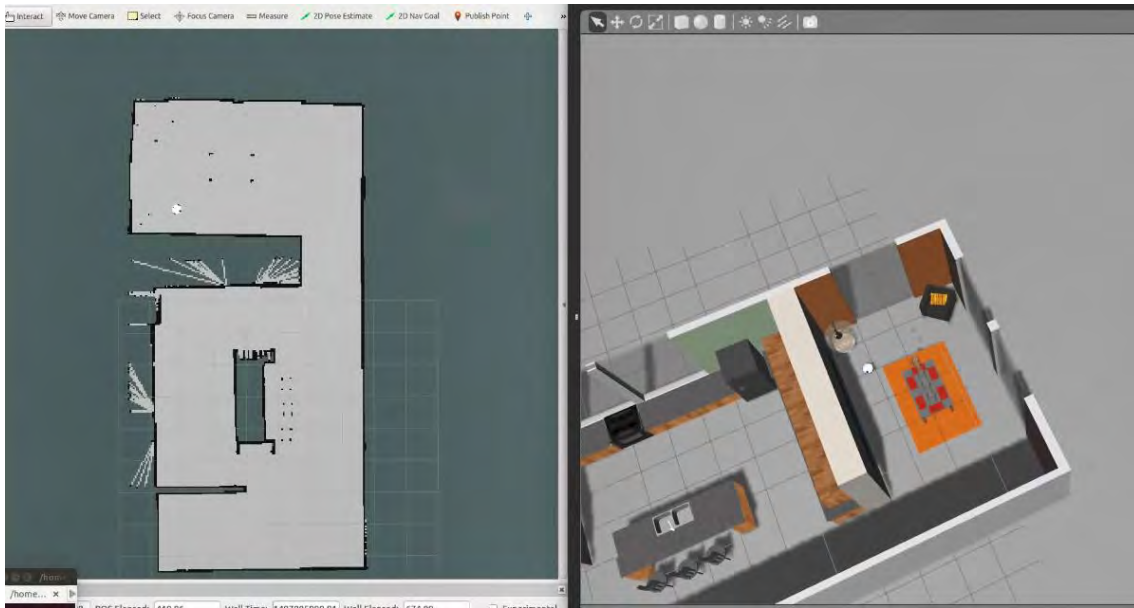
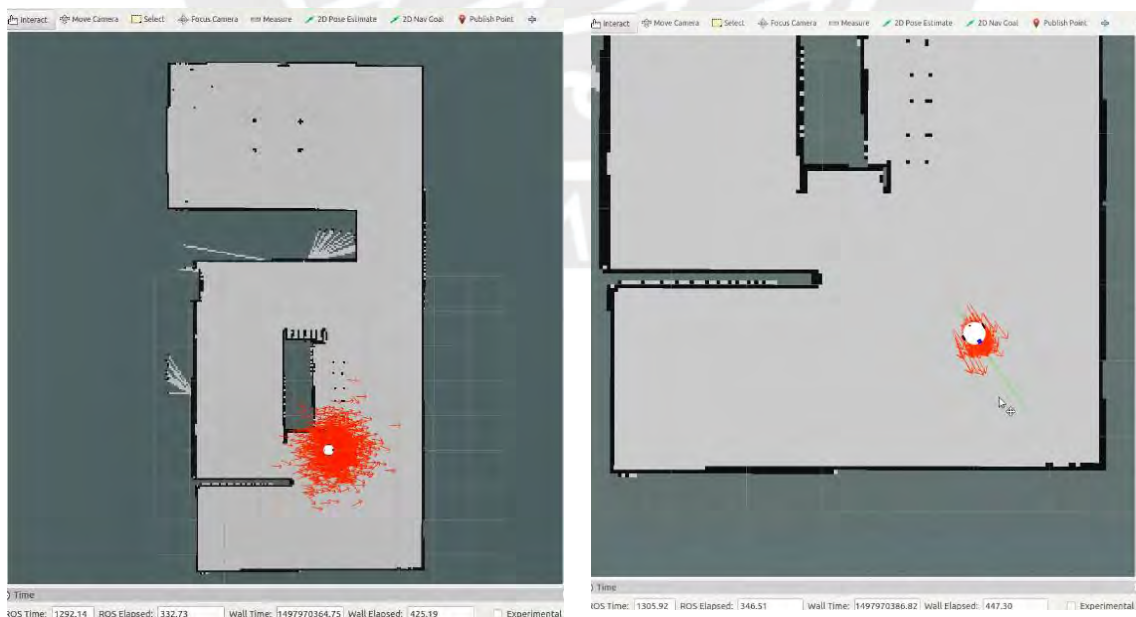


Figura 5.3: Culminación de proceso de construcción de mapa

Finalizado el proceso de localización y mapeo simultáneo, se aprecia que mediante el sensor LIDAR es posible sensor distancias del entorno que sirven para poder construir el mapa geométrico 2D. Asimismo, se aprecia que el mapa construido tiene buena resolución, pero presenta algunos errores en algunas zonas.

Luego de construir el mapa de la habitación, el robot debe desplazarse sobre la habitación con la finalidad de limpiarla. Nuevamente el robot inicia en una posición arbitraria dentro del habitación.



(a) (b)
Figura 5.4: Esquema de partículas en localización

En la Figura 5.4 (a) se muestra el robot ubicado en una posición arbitraria, por lo cual se aprecia que las partículas, representadas por flechas rojas, están bastante dispersas alrededor del robot. Luego conforme el robot se va desplazando, se aprecia cómo las partículas van convergiendo hacia la posición central del robot como se muestra en la Figura 5.4 (b). Esta convergencia indica que el robot ya es capaz de localizarse con mayor precisión dentro del mapa, indicando la posición y orientación estimada.

5.2 Simulación de planeamiento de trayectoria usando descomposición celular

Cuando el robot ya dispone de un mapa geométrico de la habitación, el siguiente paso que debe realizar es navegar de forma autónoma y simultáneamente limpiar las partículas de suciedad. Para esto, primero resulta necesario que el controlador principal disponga de un algoritmo de planificación de trayectoria de cobertura total del mapa con la finalidad de que el robot se desplace sobre todos los puntos de la habitación. El robot utiliza el algoritmo de descomposición celular de Boustrophedon [Trhun, 2005], el cual consiste básicamente en dividir el mapa de la habitación en zonas o células, para que después el robot vaya limpiando zona por zona como se muestra en la Figura 5.5.

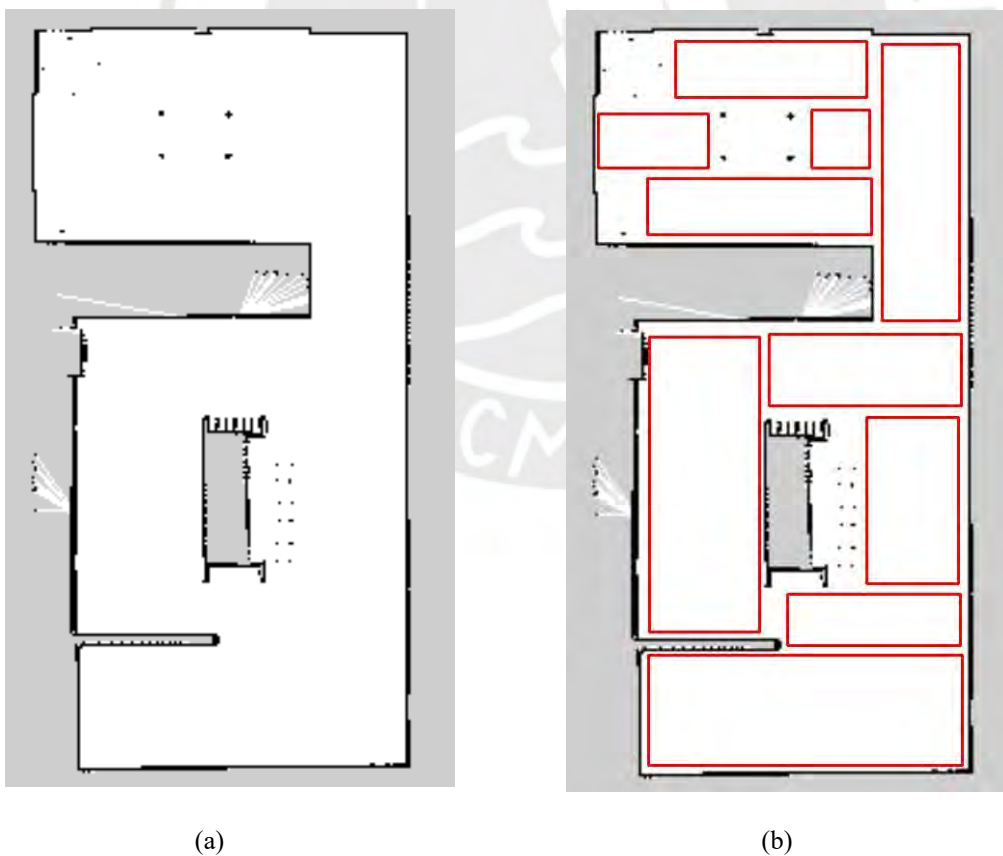


Figura 5.5: Descomposición de mapa en zonas

Luego de que el mapa es descompuesto en zonas, el robot se dirige hacia una zona y empieza a desplazarse en forma de zigzag, al mismo tiempo evitando obstáculos, sobre cada zona como se muestra en la Figura 5.6.

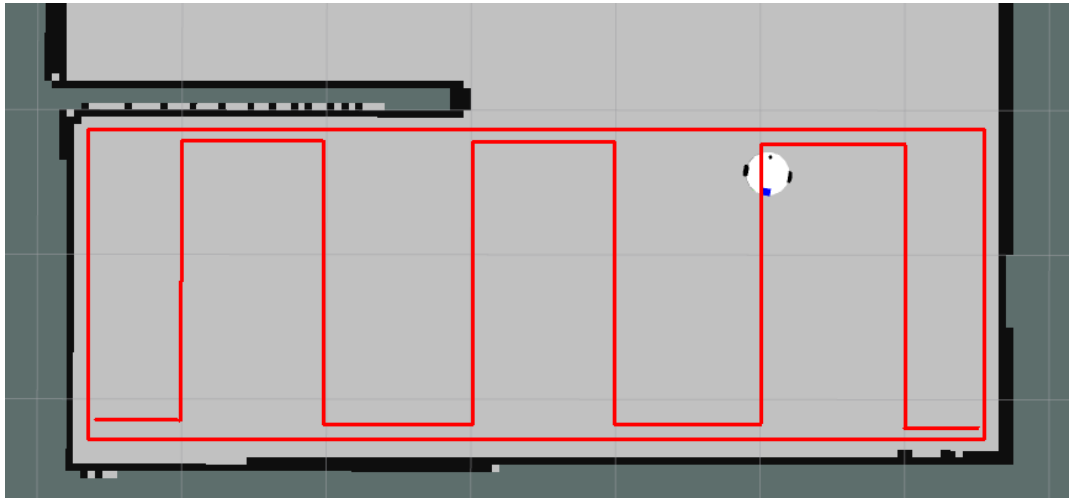


Figura 5.6: Desplazamiento en forma de zigzag

Para desplazarse, el robot debe planear la trayectoria óptima para llegar desde un punto inicial hasta un punto objetivo en cada tramo de la trayectoria en forma de zigzag como se muestra en la Figura 5.7. Adicional al resultado de la trayectoria óptima se generan los comandos de movimiento asociados a las velocidades a las que debe moverse cada rueda del robot. Adicionalmente, se propone el algoritmo de aprendizaje de máquina para ayudar en la localización del robot y corregir de cierta forma los errores acumulados que se tengan de las medidas de los sensores. La idea es que el robot sea capaz de reconocer algunos objetos como una puerta, ventana, etc. y lo tome como *landmark* o punto de referencia.



Figura 5.7: Planeamiento de trayectoria

5.3 Simulación de aprendizaje profundo

A continuación, se muestra el desarrollo del sistema de aprendizaje profundo que consiste en la construcción de una base de datos de imágenes para el entrenamiento y testeo de la red neuronal convolucional; en el diseño de la arquitectura de la red, en el entrenamiento de la red y, finalmente, en el testeo de la red. El objetivo es que luego del entrenamiento de la red neuronal convolucional, el robot sea capaz de reconocer objetos o partes de la habitación, se localice de acuerdo con estas referencias y le sirva como retroalimentación para que de esta forma corrija los errores acumulados que se tengan de los algoritmos de SLAM. El objetivo de este algoritmo de aprendizaje de máquina es mejorar el rendimiento de la navegación autónoma del robot limpiador.

5.3.1 Datos utilizados en el sistema

Los datos que se utilizan en el sistema son imágenes obtenidas de entornos de simulación como la habitación de la sala-comedor mostrada anteriormente. Cabe resaltar que el robot cuenta con una cámara frontal, la cual capturó imágenes del entorno del robot. Se construyó una base de datos que consiste en imágenes que se dividen en ocho categorías: estación de recarga, cocina, mesa, pasadizo, puerta, refrigerador, silla y ventana. Estas categorías serán los datos de salida y se determinaron que sean esas ocho porque son las partes más comunes de una casa. Las imágenes capturadas por la cámara fueron pre procesadas con el objetivo de tener imágenes uniformes con tamaño de 640x480 pixeles con una precisión de 8 bits para valores a escala de grises.



Figura 5.8: Imágenes del sistema

La Figura 5.8 muestra en la parte (a) las imágenes capturadas del entorno de las ocho categorías seleccionadas. En la parte (b) se muestra las imágenes pre procesadas que son imágenes del mismo tamaño. Además, se muestra que las imágenes se encuentran en

categorías: (0) estación de carga, (1) cocina, (2) mesa, (3) puerta, (4) pasadizo, (5) refrigerador, (6) silla, (7) ventana. La cantidad total de imágenes que tienen en la base de datos es de 6400 imágenes, 800 imágenes por cada categoría. Luego la base de datos es dividida en dos partes: 5440 imágenes (85%) se disponen para el entrenamiento de la red neuronal convolucional, mientras que 960 imágenes (15%) se disponen para el testeo de la red neuronal.

5.3.2 Arquitectura de red neuronal convolucional

En la Figura 5.9 se muestra la arquitectura diseñada de la red neuronal convolucional que se encuentra compuesta por una capa de convolución, seguida de una capa de no-linealidad, una capa de submuestreo y termina con una capa totalmente conectada.

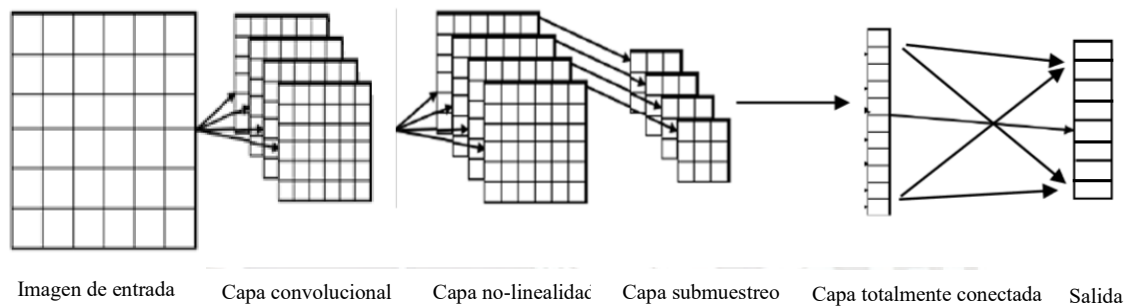


Figura 5.9: Arquitectura de red neuronal convolucional

Como entrada de la red se tiene las imágenes resultantes del preprocesamiento, es decir, imágenes con tamaño de 32x32 píxeles. En primer lugar, estas imágenes de entrada pasan por la capa convolucional, en el cual se usan 48 filtros de convolución de tamaño 7x7. La operación de convolución se realiza sin *zero padding* y con un paso de 1 píxel de cada filtro sobre la imagen, lo que resulta como salida 48 imágenes (activaciones de convolución) de tamaño 26x26 píxeles como se muestra en la Figura 5.10.

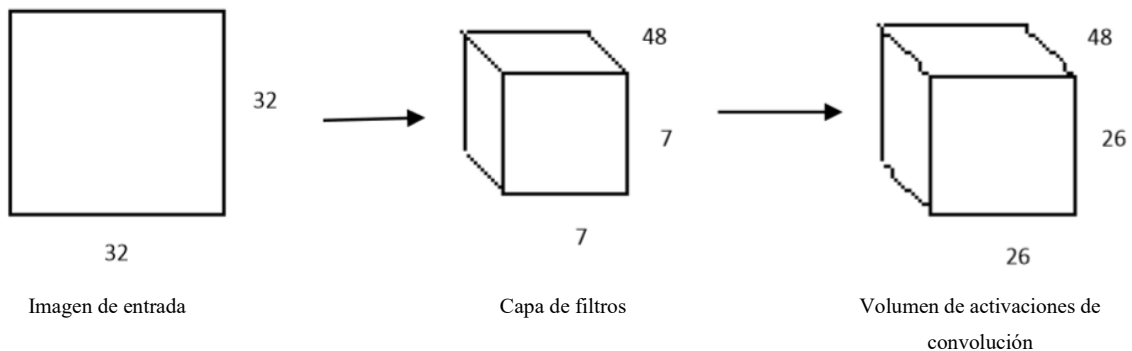


Figura 5.10: Capa convolucional

Luego de las operaciones de convolución continua la capa de no-linealidad, en la cual una función sigmoide es aplicada al volumen de activaciones de convolución. Esta función sigmoide se aplica a la salida de cada neurona de la capa de convolución, por lo que el tamaño del volumen de activaciones resultante permanece igual como se muestra en la Figura 5.11.

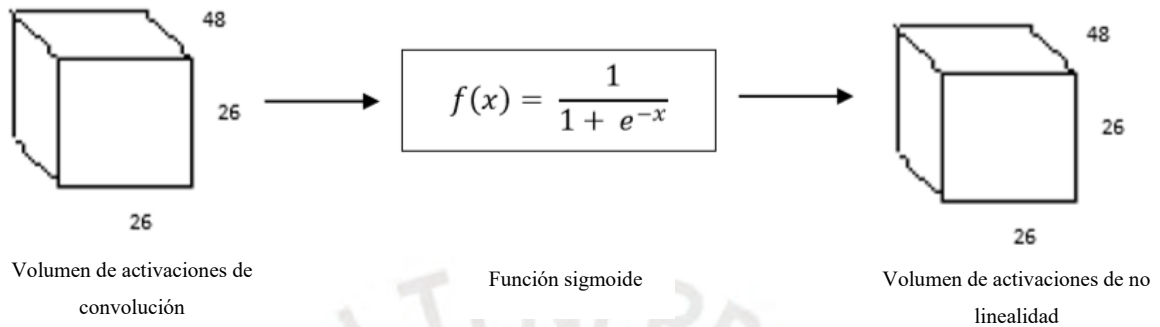


Figura 5.11: Capa de no linealidad

Después continúa la capa de submuestreo en la cual el tamaño del volumen de activaciones de no linealidad es reducido espacialmente. En esta capa, se dispone de un filtro de 2x2, que promedia el valor de cuatro neuronas de la capa de no-linealidad, por lo que cada región de 4 neuronas es sustituida por una neurona con valor igual al promedio de dicha región. Entonces el volumen de activaciones es reducido como se muestra en la Figura 5.12.

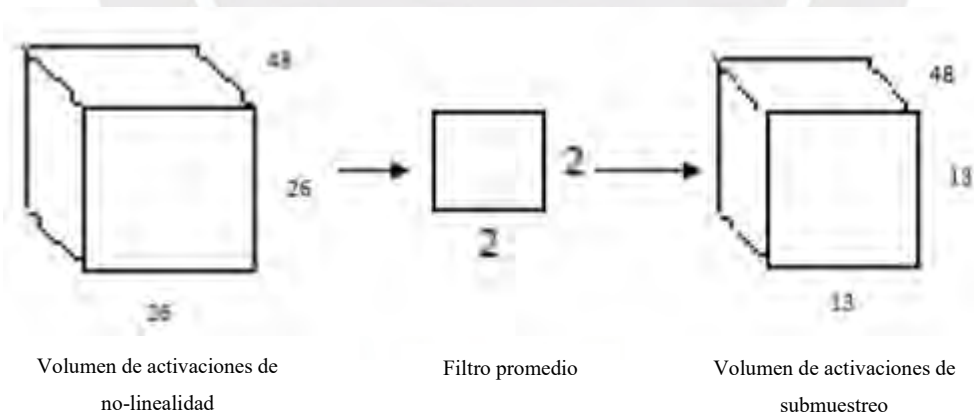


Figura 5.12: Capa de submuestreo

Luego de las operaciones de submuestreo, 48 mapas de activación de tamaño 13x13 son obtenidos y se tiene un volumen de 13x13x48. Este volumen de activaciones es reordenado para formar un vector de 8112 elementos. Este vector es la entrada de la capa totalmente conectada y que da como salida un vector de 8 elementos. Una función

Softmax es aplicada este vector de 8 elementos y se obtiene un vector de probabilidades del mismo tamaño como se muestra en la Figura 5.13. Este vector de probabilidades corresponde los valores de predicción de cada categoría de objetos que se encuentran en la habitación.

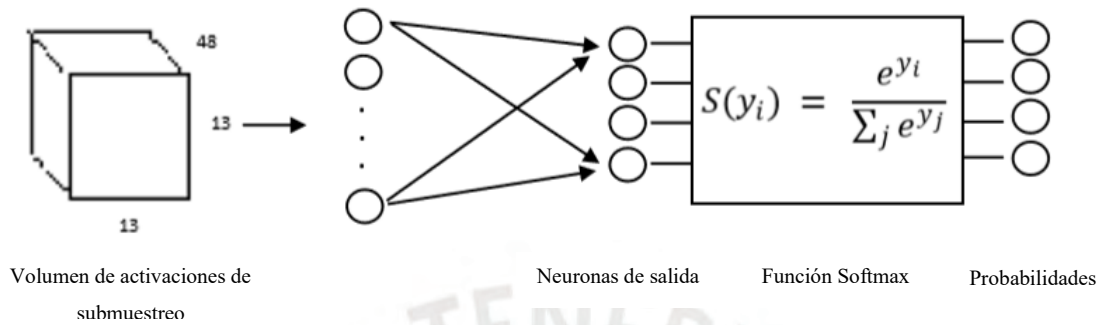


Figura 5.13: Capa totalmente conectada

5.3.3 Entrenamiento de la red neuronal convolucional

El entrenamiento de la red pasa por dos fases de *forward* y *backward* similar a una red neuronal artificial. Primeramente, se inicializan los parámetros de la red con valores aleatorios para evitar el problema de *symmetry breaking*, es decir, que en el caso de que se inicialicen todos los parámetros con valor cero se tendría el problema de que luego de la retropropagación los parámetros de entrada para una neurona en un layer serían iguales y no se optimizarían los parámetros.

Luego se realiza la propagación hacia adelante (*forward*) en la cual primero se convoluciona cada imagen de entrada con cada filtro de la capa de convolución, se aplica la función de no-linealidad y luego se realiza el submuestreo a través del promedio de cada región de cuatro elementos. Luego el volumen resultante del submuestreo se ordena en un vector que es la entrada al layer totalmente conectado y se propaga este vector con los parámetros de este último layer. A la salida se obtiene un vector de 8 elementos, donde cada elemento representa la probabilidad de cada categoría de objeto.

Después se pasa a la etapa de retropropagación, en el cual primero se calcula el error en la salida δ_{FC} , es decir el error en la capa totalmente conectada. Esto es la diferencia entre los valores de la salida de la red y los valores deseados. Luego este error debe ser propagado hacia atrás para calcular la variación de cada parámetro. El error que se obtiene en la salida debe ser propagado por las capas de convolución y submuestreo. Luego generalmente se utiliza el algoritmo de *Batch Gradient Descent* que usa todo el conjunto de entrenamiento completo para calcular la siguiente actualización de los parámetros de

la red en cada iteración hasta lograr converger en un valor óptimo local. Sin embargo, calcular el costo y los gradientes tomando en cuenta todo el conjunto de entrenamiento puede ser lento y computacionalmente caro. Por tanto, para este tipo de aplicaciones se usa otros algoritmos como *Stochastic Gradient Descent* (SGD) [Ng, 2013]. Este algoritmo usa solo uno o pocas muestras del conjunto de entrenamiento para el cálculo de los gradientes. Este método se utiliza en el presente sistema.

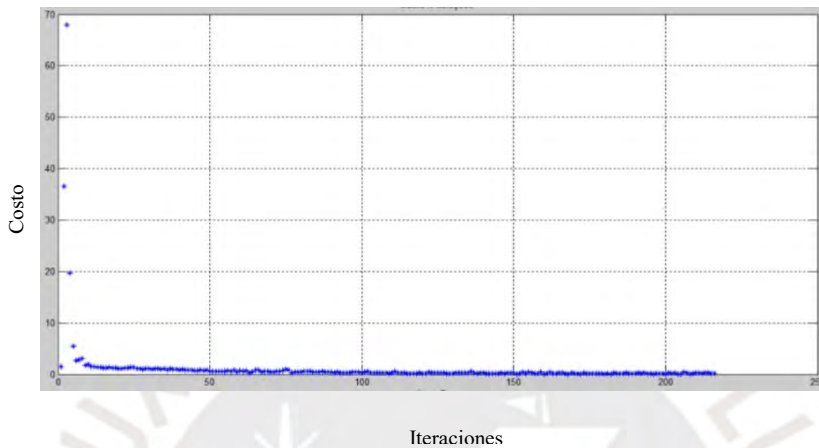


Figura 5.14: Gráfica de costo vs iteraciones

La Figura 5.14 muestra el costo de los errores en la salida de la red neuronal convolucional por cada iteración de entrenamiento. Se aprecia que el costo converge a un valor mínimo, lo que significa que los parámetros de la red neuronal convolucional fueron optimizados.

5.3.4 Desempeño de la red neuronal convolucional entrenada

Como producto del entrenamiento de la red neuronal convolucional, se obtienen los parámetros de las neuronas de las capas de la red. Cabe resaltar que la red clasifica una imagen de entrada en una de las ocho categorías: estación de recarga, cocina, mesa, pasadizo, puerta, refrigerador, silla y ventana. Entonces se pasa a verificar el desempeño de la red neuronal utilizando la parte de testeo de la base de datos.

Tabla 5.1: Desempeño de la red entrenada

Número de imágenes de testeo	960
Número de imágenes correctamente clasificadas	872
Porcentaje de acierto	90.83%

La Tabla 5.1 muestra el desempeño de clasificación de la red neuronal convolucional diseñada. Cuando se le presentan nuevas imágenes al sistema, este tiene un 90.83% porcentaje de acierto lo que significa que nueve de cada diez imágenes serán clasificadas correctamente por el robot limpiador.

5.3.5 Aplicación de la red neuronal convolucional en la navegación autónoma

La red neuronal convolucional fue diseñada y entrenada para que clasifique las imágenes que el robot capta en ocho categorías: estación de recarga, cocina, mesa, pasadizo, puerta, refrigerador, silla y ventana. Entonces mientras el robot se desplaza por la habitación, tiene la capacidad de reconocer uno de estos objetos que se encuentra en las categorías de clasificación para tenerla como referencia y corregir los errores acumulados por los algoritmos de SLAM. Específicamente, el robot al reconocer a un objeto lo toma como *landmark* o punto de referencia, estima la distancia a dicho objeto, compara esta distancia con la distancia que midió cuando estaba mapeando el ambiente, y luego corrige los errores acumulados para tener un mayor grado de certeza en la estimación de la posición.

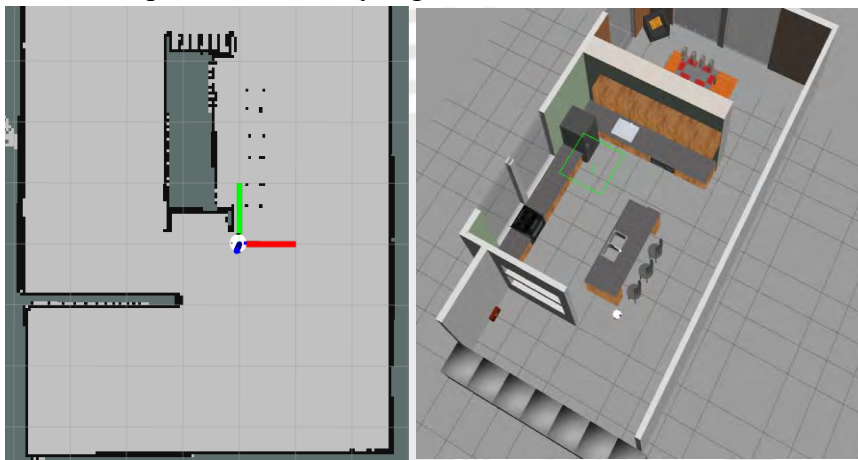


Figura 5.15: Robot en posición inicial

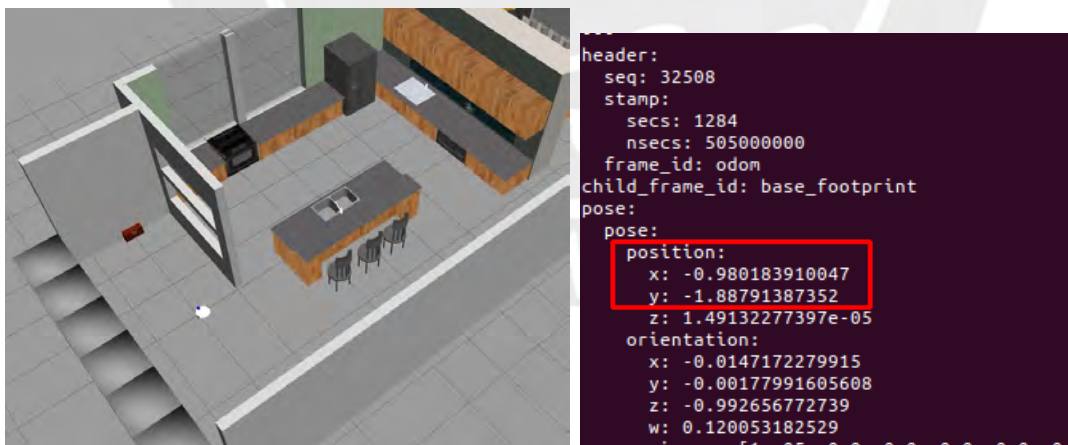


Figura 5.16: Robot en posición $(x,y)=(-1,-2)$

En la Figura 5.15 se aprecia la posición inicial del robot cuando comienza su proceso de navegación y limpieza. Luego de desplazarse por la habitación, el robot llega a la posición $(x,y) = (-1,-2)$; sin embargo, debido a errores acumulados de SLAM, la estimación de la posición no es precisa como se aprecia en la Figura 5.16.

Debido a que estos errores continúan acumulándose, el robot poco a poco se va perdiendo en la habitación, por lo que el proceso de limpieza se vuelve ineficiente. ¿Qué hacer para corregir estos errores acumulativos? o ¿qué hacer cuando el robot se ha perdido completamente? Entonces es aquí donde se utiliza el método de aprendizaje profundo a través de la red neuronal convolucional diseñada y entrenada.



Figura 5.17: Estación de recarga identificada por la red neuronal convolucional

La Figura 5.17 muestra que el robot ha podido reconocer la estación de recarga, por lo que le da la capacidad de tener como referencia a esta estación, entender que está cerca de la estación de recarga y corregir los errores acumulados de la odometría. Si es que el robot se había perdido por esa posición, entonces al reconocer este objeto ya es capaz de volver a ubicarse y poder continuar con la limpieza. Asimismo, aporta en la mejora de la trayectoria ya que inicialmente fue calculada sin errores, pero luego durante el desplazamiento del robot, la trayectoria hacia el objetivo ya no es la óptima por los errores acumulados.

Ahora también es posible que el robot pueda navegar tomando como referencia los objetos que va reconociendo por la red neuronal convolucional. Por ejemplo, tomando como referencia la imagen 5.17, cuando el robot presente un nivel bajo de batería, este puede buscar la estación por medio de la red neuronal. No obstante, este sistema de aprendizaje profundo puede tornarse aún más robusto, a través de una arquitectura más profunda de la red neuronal y de esta forma pueda identificar una cantidad mayor de categoría de objetos.

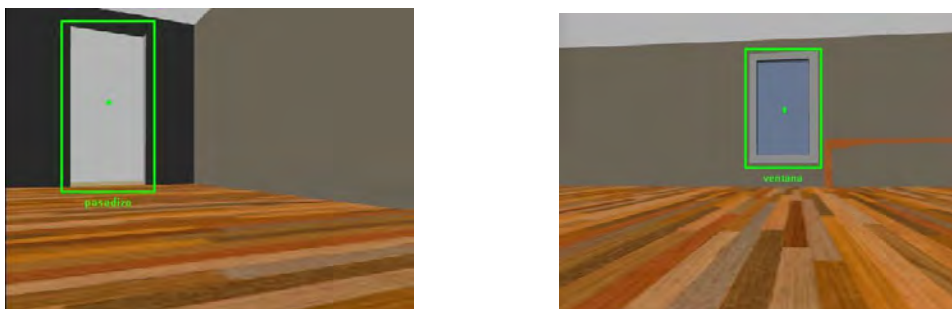


Figura 5.18: Otros objetos identificados por la red neuronal convolucional

CAPÍTULO 6

PLANOS Y COSTOS

En este capítulo, en primer lugar, se presenta la lista de planos mecánicos de ensamble y despiece del robot limpiador. Luego se presenta la lista de planos eléctricos y diagramas esquemáticos. Todos los planos y diagramas listados se encuentran en el Anexo C. Finalmente, se presenta la estimación de los costos mecánicos, costos de componentes eléctricos y electrónicos que permiten calcular el costo de elaboración del sistema mecatrónico propuesto.

6.1 Lista de planos

6.1.1 Planos mecánicos

En la Tabla 6.1 se muestra la lista de planos de ensamble, subensamble y despiece del robot limpiador autónomo propuesto.

Tabla 6.1: Lista de planos mecánicos

N° LÁMINA	TAMAÑO	TIPO	DESCRIPCIÓN
L0	A0	ENSAMBLE	ROBOT LIMPIADOR AUTÓNOMO
L1	A2	DESPIECE	CARCASA INFERIOR
L2	A3	SUBENSAMBLE	CARCASA SUPERIOR
L3	A3	DESPIECE	CONDUCTO DE ENTRADA
L4	A4	DESPIECE	CONDUCTO DE SALIDA
L5	A3	DESPIECE	SOPORTE DE COMPONENTES
L6	A3	SUBENSAMBLE	ALMACÉN DE POLVO
L7	A3	SUBENSAMBLE	FILTRO DE POLVO
L8	A3	SUBENSAMBLE	TRAPO DE PULIDO
L9	A4	DESPIECE	ESCOBILLA LATERAL
L10	A3	SUBENSAMBLE	ESCOBILLA CENTRAL
L11	A4	DESPIECE	EJE RUEDA ACOPLÉ

N° LÁMINA	TAMAÑO	TIPO	DESCRIPCIÓN
L12	A3	ENSAMBLE	ESTACIÓN DE RECARGA
L13	A3	DESPIECE	CARCASA SUPERIOR PARTE 1
L14	A4	DESPIECE	CARCASA SUPERIOR PARTE 2
L15	A4	DESPIECE	BASE DE ALMACÉN
L16	A4	DESPIECE	TAPA DE ALMACÉN
L17	A4	DESPIECE	CARCASA DE FILTRO
L18	A3	DESPIECE	BASE SUPERIOR DE TRAPO
L19	A4	DESPIECE	BASE INFERIOR DE TRAPO
L20	A3	DESPIECE	EJE ESCOBILLA
L21	A3	DESPIECE	ESCOBILLA CILÍNDRICA
L22	A4	DESPIECE	CARCASA ESTACIÓN PARTE 1
L23	A4	DESPIECE	CARCASA ESTACIÓN PARTE 2

6.1.2 Planos eléctricos y diagramas esquemáticos

En la Tabla 6.2 se muestra la lista de planos eléctricos y en la Tabla 6.3, la lista de diagramas esquemáticos del robot limpiador propuesto.

Tabla 6.2: Lista de planos eléctricos

N° LÁMINA	TAMAÑO	DESCRIPCIÓN
10	A4	ESQUEMA DE ALIMENTACIÓN
20	A4	ESQUEMA DE FUERZA DE MOTORES
30	A4	ESQUEMA DE CONEXIÓN CONTROLADORES Y SENSORES

Tabla 6.3: Lista de diagramas esquemáticos. Fuente: Propia.

N° LÁMINA	TAMAÑO	DESCRIPCIÓN
EE1	A2	ESQUEMÁTICO ROBOT LIMPIADOR

6.2 Costos

6.2.1 Costos mecánicos

En la Tabla 6.4 se muestra la lista de costos mecánicos del robot limpiador. Se aprecia que los primeros componentes requieren ser fabricados mediante impresión 3D, por lo cual se realizó una cotización en una empresa local. Con respecto a las demás piezas, se resalta que es posible adquirirlos en el mercado nacional.

Tabla 6.4: Costos mecánicos

Cantidad	Descripción	Precio(S/.)
1	Carcasa exterior - Impresión 3D	1200
1	Conducto entrada - Impresión 3D	
1	Conducto salida - Impresión 3D	
1	Soporte componentes - Impresión 3D	
1	Almacén - Impresión 3D	
1	Carcasa filtro - Impresión 3D	
1	Bases Trapo - Impresión 3D	
1	Carcasa estación - Impresión 3D	
2	Escobilla Lateral	36
1	Escobilla central	30
1	Filtro HEPA	35
1	Esponja	5
1	Tela de limpieza	6
2	Rueda 68x15mm	20
1	Rueda castor metálica	12
1	Soporte de sujeción motor	30
2	Acople lineal 6mm	32
2	Chumacera de pared	30
2	Anillo retención 3mm	10
4	Resorte	10
2	Rodamientos 3mm	12
1	Eje 5x260mm	17
1	Eje 6x20mm	6
3	Bisagra	9
1	Pernería en general	50
	Subtotal	1550

El costo de integración del diseño mecánico, fabricación de los componentes y también incluyendo las pruebas de diseño se dará en un tiempo de 20 horas y siendo el precio

por hora de mano de obra de S/. 25, el costo de integración mecánica es de S/.500. Esto resulta en un costo total de S/. 2'550 para el diseño mecánico.

6.2.2 Costos de componentes eléctricos y electrónicos

En la Tabla 6.5 se muestra la lista de costos de componentes eléctricos y electrónicos del robot limpiador. En esta tabla se especifican los proveedores, ya que algunos de los componentes requieren ser importados (precios unitarios ya incluyen costo de envíos).

Tabla 6.5: Costos eléctricos y electrónicos

Cantidad	Descripción	Proveedor	Precio Unitario(S/.)	Precio Total(S/.)
1	Raspberry Pi 3	Creatividad Ahora	239	239
1	Arduino Mega 2560	Perú Electrónico	70	70
3	Sensor Ultrasonido HCS04	Perú Electrónico	7	21
3	Sensor Infrarrojo FC-51	Electrónica HiFi	12	36
1	Sensor Inercial MPU6050	Perú Electrónico	13	13
1	Sensor LIDAR	Creatividad Ahora	1100	1100
1	Cámara Logitech	Logitech	260	260
1	Cámara USBFHD01M	Amazon	170	170
4	LED	Electrónica HiFi	1.5	6
2	Motor DC 37Dx70L	Pololu	150	300
3	Motor DC 6V	Creatividad Ahora	7	21
1	Ventilador 12V	Delta Electronics	60	60
2	Driver VNH5019	Pololu	170	340
1	Driver L298	Creatividad Ahora	13	13
1	Batería Li-Ion 18650	Alibaba	85	85
1	Fuente DC 12V	All-Batery	54	54
1	Regulador LM2596	Electrónica HiFi	12	12
1	Sensor de Voltaje	Electrónica HiFi	10	10
			Subtotal	2810

El costo de integración del diseño electrónico, tomando en cuenta también el tiempo de soldaduras y conexiones, se dará en un tiempo de 10 horas y siendo el precio por hora de mano obra de S/. 25, el costo de integración electrónica resulta S/. 250. Esto resulta en un costo total de S/. 3'060 para el diseño de electrónico.

6.2.3 Costo total

En la Tabla 6.6 se muestra que la suma de los subtotales mecánicos, eléctricos y electrónicos del sistema. Adicionalmente, se considera el costo de diseño del robot limpiador, para lo cual se considera un precio por hora de S/. 25, un tiempo de diseño de 15 horas por semana y 15 semanas de duración del presente proyecto. Esto da como resultado un costo de diseño de S/. 5'625. Finalmente, se tiene una estimación total del costo del sistema de S/. 11'235. Ahora hay que tomar en cuenta que el costo de diseño va a disminuir conforme el robot sea fabricado en grandes volúmenes de producción.

Tabla 6.6: Costo total del sistema

Subtotal Electrónica: componentes e integración	3'060
Subtotal Mecánica: componentes, fabricación e integración	2'550
Costo de diseño	5'625
Total	11'235



CONCLUSIONES Y RECOMENDACIONES

a. Conclusiones

- En la presente tesis se logró diseñar un sistema mecatrónico según los requisitos establecidos de navegar autónomamente en una habitación y limpiar eficientemente las partículas de suciedad presentes en el interior de dicho entorno.
- El robot es capaz de desplazarse mediante un mecanismo de locomoción diferencial de dos ruedas para recorrer una habitación.
- En la estimación de la posición mediante odometría, de acuerdo con las simulaciones se aprecia que usar solo encoders no es conveniente, ya que rápidamente acumula errores. Esta odometría se mejora mediante el uso de un sensor inercial (IMU), sensor LIDAR y de odometría visual, a través una cámara.
- Se entrenó una red neuronal convolucional para que el robot reconozca objetos o partes de la habitación con una precisión de 90.83% para los datos de validación.
- El robot es capaz de tomar los objetos reconocidos por la red neuronal convolucional como puntos de referencia y le sirve al algoritmo de localización y mapeamiento simultáneo para corregir los errores de estimación de ubicación del robot. De esta forma se verifica que es factible proponer un método de aprendizaje de máquina para mejorar el rendimiento de la navegación autónoma del robot limpiador.
- La estimación de costos muestra un presupuesto final de S/. 11'235 tomando en cuenta costos de los componentes mecánicos y electrónicos, costos de fabricación y costos de diseño.
- Las simulaciones y resultados obtenidos muestran que el sistema mecatrónico propuesto puede ser puesto en funcionamiento en situaciones reales y comparándolo en términos de planeación de trayectoria y tiempo de limpieza el robot puede llegar a tener un mejor rendimiento que muchos de los robots limpiadores existentes en el mercado. Esto debido a que se está proponiendo un

algoritmo de aprendizaje de máquina para que ayude en la corrección de errores acumulados en la estimación de la posición.

b. Recomendaciones

- Se puede mejorar el desempeño de la red neuronal convolucional a través de una arquitectura más profunda de la red neuronal con mayores capas convolucionales y submuestreo para que pueda identificar una cantidad mayor de categoría de objetos.
- El planeamiento de trayectoria puede complementarse con otros algoritmos de inteligencia artificial como Aprendizaje por Reforzamiento o Colonización de hormigas para mejorar aún más el rendimiento de la navegación autónoma.



BIBLIOGRAFÍA

- [1] MarketsandMarkets, "Smart Home Market by Product, Software & Service, and Geography - Global Forecast to 2022", Mayo 2016.
- [2] iRobot Corporation, "Roomba 980", [En línea]. Disponible en: <http://www.irobot.com/For-the-Home/Vacuuming/Roomba.aspx>. [Último acceso: 15 Mayo 2017].
- [3] Samsung Electronics Co., Ltd, "Samsung Powerbot VR 9000", [En línea]. Disponible en: <http://www.samsung.com/us/home-appliances/vacuums/powerbot-robot>. [Último acceso: 15 Mayo 2017].
- [4] Neato Robotics, "Neato Botvac Connected", [En línea]. Disponible en: <https://www.neatorobotics.com/robot-vacuum/botvac-connected-series/>. [Último acceso: 15 Mayo 2017].
- [5] International Federation of Robotic, "World Robotic 2016 Service Robots", [En línea]. Disponible en: <http://www.ifr.org/service-robots/statistics/>. [Último acceso: 15 Mayo 2017].
- [6] B. Barriga, "Métodos de diseño en Ingeniería Mecánica", Lima: Fondo Editorial PUCP, 1985.
- [7] W. Chen, T. Qu, Y. Zhou, K. Weng, G. Wang y G. Fu, "Door recognition and deep learning algorithm for visual based robot navigation", *IEEE International Conference on Robotics and Biomimetics*, pp. 1793-1798, 2014.
- [8] H. Choset y P. Pignon, "Coverage Path Planning: The Boustrophedon Cellular Decomposition", *International Conference on Field and Service Robotics*, 1997.
- [9] M. Egerstedt, "Control of Mobile Robots", Atlanta: Georgia Institute of Technology, 2013.
- [10] A. Gerard, D. Lindsey y M. David, "Robotic Floor Cleaning Device". United States Patente US 6,605,156 B1, Agosto 2013.
- [11] K. Han-Gyeol, Y. Jeong-Yean y K. Dong-Soo, "Experience based Domestic Environment and User Adaptive Cleaning Algorithm of a Robot Cleaner", *IEEE 11th International Conference on Ubiquitous Robots and Ambient Intelligence*, pp. 176-178, 2014.

- [12] S. Haykin, "Redes Neuronales-Principios y práctica", Porto Alegre: Bookman, 2001.
- [13] J. Jones, M. Newton, D. Nugent y P. Sandin, "Autonomous Floor Cleaning Robot". United States Patente US 7,636,982 B2, Diciembre 2009.
- [14] L. Joseph, "Mastering ROS for Robotics", Birmingham: Packt Publishing, 2015.
- [15] A. Karpathy, "Convolutional Neural Networks for Visual Recognition", 2016. [En línea]. Disponible en: <https://cs231n.github.io/convolutional-networks/>. [Último acceso: 20 Junio 2017].
- [16] S. Kohlbrecher, O. Von Stryk, J. Meyer y U. Klingauf, "A Flexible and Scalable SLAM System with Full 3D Motion Estimation", *IEEE International Symposium on Safety, Security and Rescue Robotics*, 2011.
- [17] Y. LeCun y Y. Bengio, "Convolutional networks for images, speech, and time series", *The Handbook of Brain Theory and Neural Networks*, pp. 255-258, 1995.
- [18] D. Lee, "Robotics: Estimation and Learning", Pensilvania: Universidad de Pensilvania, 2016.
- [19] J. Maulin y Z. Mukesh, "Neuro-Fuzzy Based Autonomous Mobile Robot Navigation System", *IEEE 11th International Conference on Control, Automation, Robotics and Vision*, pp. 384-389, 2010.
- [20] V. Nath y S. Levinson, "Autonomous Robotics and Deep Learning", London: Springer, 2014.
- [21] A. Ng, "Aprendizaje Automático", Stanford: Universidad de Stanford, 2013.
- [22] A. Ollero, "Robótica, Manipuladores y robots móviles", Barcelona: Alfaomega, 2001.
- [23] B. Siciliano y O. Khatib, "Springer Handbook of Robotics", Berlin: Springer Berlin Heidelberg, 2008.
- [24] R. Siegwart, I. Nourbakhsh y D. Scaramuzza, "Introduction to Autonomous Mobile Robots", London: The MIT Press, 2011.
- [25] S. Thrun, W. Burgard y D. Fox, "Probabilistic Robotics", Cambridge, 2005.
- [26] S. Tzafestas, "Introduction to Mobile Robot Control", Atenas: Universidad de Atenas, 2014.

- [27] Z. Zheng, C. Weihai, P. Chen y W. Xingming, "A Novel Navigation System for Indoor Cleaning Robot", *IEEE International Conference on Robotics and Biomimetics*, pp. 2159-2164, 2016.

