



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN

**“DESARROLLO DE UN PROTOTIPO DE UN ROBOT MÓVIL
TERRESTRE AUTÓNOMO PARA FUMIGACIÓN AGRÍCOLA, EN
EL CULTIVO DE MORA DE CASTILLA.”**

Trabajo de Integración Curricular

Tipo: Dispositivo Tecnológico

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

AUTOR:

CHRISTIAN FERNANDO AGUAGALLO COELLO

Riobamba – Ecuador

2023



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN

**“DESARROLLO DE UN PROTOTIPO DE UN ROBOT MÓVIL
TERRESTRE AUTÓNOMO PARA FUMIGACIÓN AGRÍCOLA, EN
EL CULTIVO DE MORA DE CASTILLA.”**

Trabajo de Integración Curricular

Tipo: Dispositivo Tecnológico

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

AUTOR: CHRISTIAN FERNANDO AGUAGALLO COELLO

DIRECTOR: Ing. JOSÉ LUIS TINAJERO LEÓN

Riobamba – Ecuador

2023

© 2023, Christian Fernando Aguagallo Coello

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Yo Christian Fernando Aguagallo Coello, declaro que el presente Trabajo de Integración Curricular es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Integración Curricular. El patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 14 de noviembre de 2023



Christian Fernando Aguagallo Coello
0605153865

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN

El Tribunal de Trabajo de Integración Curricular certifica que: El Trabajo de Integración Curricular, Tipo: Dispositivo tecnológico; “**DESARROLLO DE UN PROTOTIPO DE UN ROBOT MÓVIL TERRESTRE AUTÓNOMO PARA FUMIGACIÓN AGRÍCOLA, EN EL CULTIVO DE MORA DE CASTILLA**”; realizado por el señor **CHRISTIAN FERNANDO AGUAGALLO COELLO**, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Integración Curricular, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal autoriza su presentación.

	FIRMA	FECHA
Ing. Andrés Fernando Morocho Caiza PRESIDENTE DEL TRIBUNAL		2023-11-14
Ing. José Luis Tinajero León DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR		2023-11-14
Ing. Diego Sebastián Santana Alarcón ASESOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR		2023-11-14

DEDICATORIA

Este trabajo de titulación lo dedico a mi familia. Mi padre, mi madre y mis hermanas que han sido siempre mi apoyo y motivación para aprender continuamente, un ejemplo constante de cómo no rendirme ante las adversidades y agradecer las oportunidades de la vida.

Christian Aguagallo

AGRADECIMIENTO

Agradezco a mi familia que ha sido mi respaldo para que pueda cumplir esta meta, al igual que agradezco a mis amigos, compañeros y maestros que han sido parte de este difícil, pero a la vez hermoso período donde he aprendido mucho y espero reflejar lo mejor de estas experiencias en mi vida profesional.

Christian Aguagallo

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS.....	x
ÍNDICE DE ILUSTRACIONES	xi
ÍNDICE DE ANEXOS	xiv
RESUMEN.....	xv
SUMMARY.....	xvi
INTRODUCCIÓN	1
CAPÍTULO I	
1. DIAGNÓSTICO DEL PROBLEMA.....	2
1.1. Antecedentes.....	2
1.2. Justificación Teórica	3
1.3. Justificación aplicativa.....	4
1.4. Objetivo General.....	6
1.4.1. <i>Objetivos específicos</i>	6
CAPÍTULO II	
2. MARCO TEÓRICO	7
2.1. Cultivo de Mora de castilla	7
2.1.1. <i>Enfermedades en la mora de castilla</i>	8
2.2. Fumigación Agrícola	9
2.2.1. <i>Pulverización de compuestos agrícolas para fumigación.</i>	9
2.3. Robótica en la Agricultura.....	11
2.3.1. <i>Fumigación autónoma en la agricultura</i>	11
2.4. Robots Móviles	12
2.4.1. <i>Sistemas de Locomoción</i>	12
2.4.2. <i>Configuraciones Cinemáticas</i>	13
2.5. Sistemas de navegación y localización en robots móviles.....	14
2.5.1. <i>Planificadores de trayectoria</i>	14
2.5.2. <i>Sistema de navegación basado en sensores</i>	15

2.5.3.	<i>Sistema de navegación basado en mapas.....</i>	16
2.6.	Tarjetas de desarrollo	17
2.7.	Sistemas de control para robots móviles	21
2.7.1.	<i>Análisis comparativo de los sistemas de control analizados</i>	22
2.8.	Software para Robótica	23
2.9.	Controladores de motor	24
2.10.	Sensores	24
2.10.1.	<i>Análisis comparativo de los sensores analizados</i>	27
2.11.	Actuadores.....	27
2.11.1.	<i>Motores DC</i>	27
2.11.2.	<i>Dimensionamiento de un motor DC.....</i>	28
2.11.3.	<i>Bomba DC.....</i>	29
2.11.4.	<i>Sistema de Pulverización.....</i>	30
2.12.	Sistemas de Alimentación	31
2.12.1.	<i>Dimensionamiento del banco de baterías</i>	32
CAPÍTULO III		
3.	MARCO METODOLÓGICO	35
3.1.	Requerimientos del prototipo	35
3.1.1.	<i>Requerimientos de Hardware.....</i>	35
3.1.2.	<i>Requerimientos de Software.....</i>	35
3.2.	Concepción de la arquitectura general del prototipo.....	36
3.3.	Diseño de los bloques del prototipo.....	36
3.3.1.	<i>Bloque de obtención y procesamiento de datos.....</i>	36
3.3.2.	<i>Bloque de control de motores</i>	37
3.3.3.	<i>Bloque de fumigación</i>	37
3.3.4.	<i>Bloque de Alimentación</i>	37
3.4.	Estructura del prototipo	38
3.5.	Selección de los elementos que conforman los bloques	39
3.5.1.	<i>Dispositivos de obtención y procesamiento de datos</i>	40

3.5.2.	<i>Dispositivos actuadores</i>	41
3.5.3.	<i>Dispositivos de Alimentación</i>	45
3.6.	Elementos adicionales	47
3.7.	Diagrama eléctrico de conexiones	48
3.8.	Ensamblaje de la estructura y hardware del prototipo	49
3.8.1.	<i>Estructura metálica del prototipo.</i>	49
3.8.2.	<i>Ensamblaje del sistema de locomoción</i>	50
3.9.	Ensamblaje de los elementos electrónicos	50
3.10.	Ensamblaje del sistema de pulverización	51
3.11.	Configuración de la Raspberry pi 4 y ROS	52
3.11.1.	<i>Configuración del sensor LiDAR LD06</i>	53
3.12.	Programación del prototipo	54
CAPÍTULO IV		
4.	VALIDACIÓN DE PROTOTIPO	59
4.1.	Prueba de estabilidad del sensor LiDAR LD06	59
4.2.	Prueba de alcance del sensor LiDAR LD06	61
4.3.	Prueba de estabilidad del prototipo	63
4.4.	Prueba de reconocimiento del camino y generación de trayectoria	64
4.5.	Prueba de rango de pulverización	66
4.6.	Prueba de optimización de la pulverización	69
4.7.	Prueba de autonomía del robot	71
4.8.	Análisis de costos del prototipo	72
CAPÍTULO V		
5.	CONCLUSIONES Y RECOMENDACIONES	73
5.1.	CONCLUSIONES	73
5.2.	RECOMENDACIONES	74
BIBLIOGRAFÍA		
ANEXOS		

ÍNDICE DE TABLAS

Tabla 1-2:	Comparación entre Tarjetas de desarrollo Arduino.....	17
Tabla 2-2:	Comparación entre los principales modelos de Raspberry Pi.....	18
Tabla 3-2:	Comparación entre las tarjetas Jetson.....	19
Tabla 4-2:	Características de la tarjeta de desarrollo Beaglebone Black.....	20
Tabla 5-2:	Comparación entre sistemas de control para robots móviles	22
Tabla 6-2:	Comparación entre sistemas de control para robots móviles	24
Tabla 7-2:	Comparación entre los sensores de detección para robots móviles	27
Tabla 8-2:	Comparación entre motores DC para robots móviles.....	27
Tabla 9-2:	Comparación entre las bombas DC volumétricas	29
Tabla 10-2:	Comparación entre boquillas de aspersión	30
Tabla 11-2:	Comparación entre los sistemas de alimentación para robots móviles.....	31
Tabla 1-3:	Comparación entre los sistemas de alimentación para robots móviles.....	40
Tabla 2-3:	Características de la Tarjeta Raspberry Pi 4	41
Tabla 3-3:	Características de la bomba Seaflo 12VDC.....	42
Tabla 4-3:	Características del controlador de velocidad PWM	43
Tabla 5-3:	Características del motor Ampflow G43	44
Tabla 6-3:	Características del controlador dual Sabertooth 2x32	45
Tabla 7-3:	Características del regulador de voltaje XL4015	45
Tabla 8-3:	Dimensionamiento de la batería.....	46
Tabla 9-3:	Características de la batería ChilWee 6-DZF-20.3.....	47
Tabla 1-4:	Análisis comparativo de la medición del sensor lidar LD06	60
Tabla 2-4:	Análisis comparativo del alcance del sensor lidar LD06.....	62
Tabla 3-4:	Análisis comparativo de la estabilidad del prototipo.....	64
Tabla 4-4:	Análisis comparativo de la distancia al borde de la trayectoria generada ...	65
Tabla 5-4:	Análisis del rango de pulverización	67
Tabla 6-4:	Análisis del rango de pulverización por cada distancia muestral.....	69
Tabla 7-4:	Análisis de autonomía del prototipo.....	71
Tabla 8-4:	Análisis de costos del prototipo	72

ÍNDICE DE ILUSTRACIONES

Ilustración 1-1:	Diagrama de bloques del funcionamiento básico del robot móvil autónomo para fumigación.....	5
Ilustración 1-2:	Cultivo de Mora de Castilla	7
Ilustración 2-2:	Botritis en la Mora de Castilla	8
Ilustración 3-2:	Antracnosis en la Mora de Castilla.....	8
Ilustración 4-2:	Oidium o Míldeo polvoso en plantas de mora de castilla	8
Ilustración 5-2:	Fumigación agrícola manual.....	9
Ilustración 6-2:	Fumigación agrícola mecanizada	9
Ilustración 7-2:	Pulverización agrícola	10
Ilustración 8-2:	Robots para agricultura.....	11
Ilustración 9-2:	Robot agrícola terrestre	11
Ilustración 10-2:	Clasificación de robots de acuerdo al tipo de locomoción	12
Ilustración 11-2:	Configuraciones de los RMR.....	13
Ilustración 12-2:	RRT en entornos simétricos	15
Ilustración 13-2:	RRT en entornos asimétricos	15
Ilustración 14-2:	Aplicación de RRT a un robot con conducción diferencial	15
Ilustración 15-2:	Elementos del robot móvil Pioneer	16
Ilustración 16-2:	Experimento SLAM multi-robot - DARPA Subterranean Challenge	16
Ilustración 17-2:	Arduino UNO	17
Ilustración 18-2:	Raspberry Pi 4 Model B	18
Ilustración 19-2:	NVIDIA Jetson Nano	19
Ilustración 20-2:	BeagleBone Black.....	20
Ilustración 21-2:	Sensor HC-SR501	25
Ilustración 22-2:	Sensor Ultrasónico HC-SR04	25
Ilustración 23-2:	Sensor infrarrojo de largo alcance JS200XF	26
Ilustración 24-2:	YDLIDAR X4	26
Ilustración 25-2:	Tipos de boquillas para pulverización.....	30
Ilustración 26-2:	Conexiones para bancos de baterías de 12V.....	33
Ilustración 27-2:	Conexión incorrecta de baterías tanto en descarga como en la carga	33
Ilustración 28-2:	Conexiones adecuadas para un banco de baterías equilibrado	34
Ilustración 1-3:	Concepción general del prototipo	36
Ilustración 2-3:	Arquitectura de los módulos del prototipo.....	36

Ilustración 3-3:	Diagrama de bloques para la obtención y procesamiento de datos	37
Ilustración 4-3:	Diagrama de bloques para el control de motores	37
Ilustración 5-3:	Diagrama de bloques para la fumigación	37
Ilustración 6-3:	Diagrama de bloques para la alimentación del prototipo.....	37
Ilustración 7-3:	Diseño del prototipo de robot autónomo para fumigación agrícola en mora de castilla en un software CAD.....	38
Ilustración 8-3:	Base soporte del prototipo	38
Ilustración 9-3:	Base con ruedas acopladas.....	38
Ilustración 10-3:	Motores acoplados a la base con sus soportes y poleas conectadas para el sistema de transmisión	39
Ilustración 11-3:	Acople de elementos en la base del prototipo.....	39
Ilustración 12-3:	Ensamblaje de la carcasa metálica y el sensor LIDAR	39
Ilustración 13-3:	Sensor LIDAR LD06	40
Ilustración 14-3:	Raspberry Pi 4.....	41
Ilustración 15-3:	Bomba de diafragma Seaflo DC 33 Series	42
Ilustración 16-3:	Controlador de velocidad PWM	42
Ilustración 17-3:	Motor Ampflow G43-500-M.....	43
Ilustración 18-3:	Controlador Sabertooth 2x32.....	44
Ilustración 19-3:	Regulador de voltaje XL4016.....	45
Ilustración 20-3:	Batería de GEL ChilWee 6-DZF-20.3	46
Ilustración 21-3:	Boquilla de pulverización 3mm.....	47
Ilustración 22-3:	Tubería 12V.....	48
Ilustración 23-3:	Diagrama eléctrico de conexiones para los dispositivos que integran el prototipo. Cable rojo positivo y cable azul negativo.....	49
Ilustración 24-3:	Estructura base del prototipo	49
Ilustración 25-3:	Sistema de transmisión para los motores del prototipo	50
Ilustración 26-3:	Estructura metálica del prototipo	50
Ilustración 27-3:	Placa base con dispositivos electrónicos	51
Ilustración 28-3:	Ubicación de las boquillas de pulverización a 50cm	51
Ilustración 29-3:	Sistema de pulverización: conexión a la bomba y al contenedor.....	52
Ilustración 30-3:	Activación del protocolo VNC desde la herramienta PuTTY	52
Ilustración 31-3:	Activación del sensor LiDAR LD06 mediante comunicación serial con la Raspberry.....	53
Ilustración 32-3:	Visualización de la nube de puntos del sensor LD06 en Rviz	53

Ilustración 33-3:	Diagrama de flujo del prototipo.....	54
Ilustración 34-3:	Clase principal del prototipo.....	55
Ilustración 35-3:	Adquisición y procesamiento de los datos LiDAR	55
Ilustración 36-3:	Método para filtrar los datos de la nube de puntos	56
Ilustración 37-3:	Método para identificar el camino adecuado para el prototipo.....	56
Ilustración 38-3:	Método para generar la trayectoria en el camino identificado.....	57
Ilustración 39-3:	Clase que define la red neuronal perceptrón multicapa.....	57
Ilustración 40-3:	Datos de entrenamiento de la MLP.....	58
Ilustración 1-4:	Medición a 3m del objetivo: flexómetro y sensor	59
Ilustración 2-4:	Prueba - Validación del sensor.....	61
Ilustración 3-4:	Prueba de alcance del sensor LiDAR LD06	62
Ilustración 4-4:	Prueba de medición del sensor LiDAR LD06.....	63
Ilustración 5-4:	Tiempo de inicio del software y detección del camino	63
Ilustración 6-4:	Prueba de medición del sensor LiDAR LD06.....	64
Ilustración 7-4:	Reconocimiento de caminos	65
Ilustración 8-4:	Generación de una trayectoria por la mitad del camino	65
Ilustración 9-4:	Prueba - Generación de Trayectoria.....	66
Ilustración 10-4:	Medición del alcance de la pulverización por el experto del cultivo	67
Ilustración 11-4:	Prueba - Alcance del sistema de pulverización.....	68
Ilustración 12-4:	Comparación entre fumigación manual y automatizada	69
Ilustración 13-4:	Prueba - Alcance del sistema de pulverización – tiempo	70
Ilustración 14-4:	Prueba - Alcance del sistema de pulverización – volumen	70
Ilustración 15-4:	Prueba - Autonomía del prototipo.....	71

ÍNDICE DE ANEXOS

ANEXO A:	Pines e interface de comunicación del sensor LiDAR LD06
ANEXO B:	Pines GPIO de la Raspberry PI 4
ANEXO C:	Guía de usuario de controlador Sabertooth 2x32
ANEXO D:	Conexiones y regulación de los dispositivos electrónicos
ANEXO E:	Sistema de Pulverización: ensamblaje y funcionamiento
ANEXO F:	Instalación de Raspbian Buster en la SD Card
ANEXO G:	Activación del protocolo VNC y adecuación del entorno de desarrollo.
ANEXO H:	Instalación del SDK del sensor LiDAR LD06 en ROS Noetic
ANEXO I:	Recolección de datos de entrenamiento de la MLP
ANEXO J:	Entrenamiento de la MLP y almacenamiento de los pesos resultantes
ANEXO K:	Clase “SabertoothController” y conexión de los pines
ANEXO L:	Clase “PumpController” para el control de la bomba

RESUMEN

La fumigación agrícola tradicional aplica un conjunto de conocimientos, técnicas y herramientas que el agricultor adquiere por experiencia en el manejo del cultivo, sin embargo, la fumigación con productos fitoquímicos debe ser precisa para tener una aplicación uniforme en todo el sembrío y así evitar la resistencia de plagas. El trabajo de integración curricular presenta el desarrollo de un prototipo autónomo para fumigación agrícola, enfocada al cultivo de mora de Castilla. La investigación correspondiente estableció las características de la estructura del prototipo y los algoritmos adecuados para cumplir con los requerimientos enfocados a mejorar la práctica de fumigación tradicional en la mora de Castilla. Para tener una percepción en tiempo real del entorno del prototipo se emplea un sensor LiDAR. Sus datos se analizan en la Raspberry pi mediante la implementación de ROS, Python y Raspbian, trabajando con nodos que facilitan la comunicación entre el software y el hardware. Además, para el control del movimiento del prototipo se implementó una red neuronal perceptrón multicapa, que se encarga de predecir los valores de velocidad y dirección del prototipo “RAPFA” de acuerdo a la trayectoria por el centro del camino entre hileras que se genera en cada iteración, para que se mueva a una velocidad constante de 4km/h. Pues a esta velocidad se evita la deriva debido a que la gota recibe más energía y se produce una mejor penetración en el cultivo. Las pruebas realizadas revelan un ahorro de alrededor del 27% del líquido aplicado al cultivo en fumigaciones tradicionales con bombas de mochila con un rango de alcance total del 1.60m entre hileras. De esta manera se reduce el tiempo de trabajo del agricultor hasta en un 73% de acuerdo a los resultados obtenidos y se minimiza la exposición prolongada a sustancias químicas perjudiciales para su salud.

Palabras clave: <ROBOT AUTÓNOMO>, <SENSOR LIDAR>, <FUMIGACIÓN AUTÓNOMA>, <GENERACIÓN DE TRAYECTORIA>, <SISTEMA OPERATIVO ROBÓTICO (ROS)>, <RED NEURONAL PERCEPTRÓN MULTICAPA>.

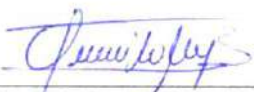


1745-DBRA-UPT-2023

SUMMARY

Traditional agricultural fumigation applies a set of knowledge, techniques and tools that the farmer acquires through experience in crop management; however, fumigation with phytochemical products must be precise to have a uniform application throughout the crop and thus avoid resistance of pests. The curricular integration work presents the development of an autonomous prototype for agricultural fumigation, focused on the cultivation of blackberries from Castilla. The corresponding research established the characteristics of the prototype structure and the appropriate algorithms to meet the requirements focused on improving the traditional fumigation practice of blackberries from Castilla. To have a real-time perception of the prototype's environment, a LiDAR sensor is used. Its data is analyzed on the Raspberry pi by implementing ROS, Python and Raspbian, working with nodes that facilitate communication between software and hardware. In addition, to control the movement of the prototype, a multilayer perceptron neural network was implemented, which is responsible for predicting the speed and direction values of the "RAPFA" prototype according to the trajectory through the center of the path between rows that is generated in each iteration, so that it moves at a constant speed of 4km/h since at this speed drift is avoided because the drop receives more energy and better penetration into the crop occurs. The tests carried out reveal a saving of around 27% of the liquid applied to the crop in traditional fumigations with backpack pumps with a total range of 1.60m between rows. In this way, the farmer's work time is reduced by up to 73% according to the results obtained and prolonged exposure to chemical substances harmful to their health is minimized.

Keywords: <AUTONOMOUS ROBOT>, <LIDAR SENSOR>, < AUTONOMOUS FUMIGATION>, <TRAJECTORY GENERATION>, < ROBOTIC OPERATING SYSTEM(ROS)>, <MULTILAYER PERCEPTRON NEURAL NETWORK>.



Lic. Nelly Padilla P. Mgs
DOCENTE FIE

INTRODUCCIÓN

El avance tecnológico en áreas referentes a la alimentación y salud humana es muy importante en la sociedad actual, pues aporta soluciones técnicas e innovadoras a procesos sistemáticos que pueden adoptar herramientas de control automático o ser reemplazados por algoritmos específicos. Las mejoras tecnológicas reducen los riesgos laborales a los que normalmente está expuesto el personal y permiten prevenir accidentes, mejorar la producción, el control de los procesos y generar nuevas bases de conocimiento para futuras líneas de investigación.

La agricultura es una actividad de interés universal, de ella depende la sustentabilidad de la sociedad humana y la seguridad alimentaria, que es uno de los problemas principales a resolver de los gobiernos en este siglo. La adopción de tecnologías y prácticas innovadoras por parte de los agricultores es el principal factor para impulsar la productividad agrícola, mejorar sus ingresos, aumentar los rendimientos, gestionar los insumos con mayor eficiencia, introducir nuevos cultivos y sistemas de producción, mejorar la calidad de sus productos, conservar los recursos naturales y adaptarse a los desafíos climáticos (BANCO MUNDIAL, 2019).

La robótica en la agricultura ha tenido grandes avances en países desarrollados, gracias a la innovación y recursos tecnológicos que poseen, y a que se han involucrado en el desarrollo de técnicas de producción agrícola autónomas, con el uso de robots, drones, sistemas de geolocalización y almacenamiento de datos por métodos de inteligencia artificial, y en base a sensores o sistemas satelitales, biotecnología, entre otros. Esto hace que su nivel productivo sea alto en comparación con países que practican agricultura tradicional o tienen poca injerencia tecnológica.

En base al estudio de la información disponible sobre los avances de la robótica en la agricultura y las características de los dispositivos electrónicos desarrollados por fabricantes especializados, se pueden desarrollar nuevos prototipos enfocados en satisfacer las necesidades de los agricultores en zonas donde la tecnología aún es escasa y se requieren aplicar nuevas técnicas en el manejo de los cultivos.

El presente trabajo de integración curricular plantea el desarrollo de un prototipo de robot móvil autónomo que pueda realizar las labores de fumigación en cultivos de mora de Castilla, considerando las características del cultivo y las ventajas que los robots móviles presentan en ambientes como los campos agrícolas, pues pueden incorporar herramientas como sensores, sistemas de navegación, generadores de trayectorias, controladores, actuadores y manipuladores, para maximizar su eficiencia en las labores agrícolas y disminuir la presencia de personal en los cultivos. Esto permite precautelar la salud de los agricultores de mora de Castilla y desarrollar una agricultura más sostenible y tecnológicamente avanzada.

CAPÍTULO I

1. DIAGNÓSTICO DEL PROBLEMA

1.1. Antecedentes

La compañía XAG (Advancing Agriculture) desarrolló un vehículo terrestre no tripulado denominado XAG R150, que cuenta con una potente escalabilidad y múltiples modos de operación, es la primera plataforma agrícola producido en masa de su tipo. Abarca desde la protección de cultivos de precisión, reconocimiento de campo hasta entrega de material en la finca. El robot puede trabajar de forma autónoma y por control remoto, y para la aplicación de la fumigación emplea un rociado omnidireccional con una inclinación panorámica sin escobillas de transmisión directa con giroscopio integrado, lo que facilita la pulverización puntual y una cobertura de 360 grados (XAG, 2021).

En Colombia, la Universidad Militar Nueva Granada diseñó y construyó un robot llamado CERES, el cual realiza labores como remoción de maleza y fumigación. Los resultados de las pruebas sobre una superficie de cemento indican que el tiempo de estabilización de la velocidad traslacional del sistema real se da cercana a los 2 segundos y su orientación alcanza la referencia sobre los 6 segundos, los valores de voltajes y potencia aplicados a los motores, se asemejan a los resultados de nivel dinámico y sugieren que para navegar en un cultivo siguiendo los surcos o la trayectoria referenciada por el agricultor, se deben desarrollar controles de seguimiento de trayectoria, cerrando el lazo de control en una jerarquía superior a los implementados en el diseño (GUZMÁN, y otros, 2019).

En cuanto a los trabajos realizados en Ecuador, en la Universidad Estatal Península de Santa Elena, se elaboró un prototipo de robot semiautónomo, especializado en fumigación agrícola bajo la tecnología “Arduino”. Se trata de una estructura para suelos llanos con irregularidades no mayores a 4 cm, con un peso bruto de 5.45 kg, un control PTZ (Pan, Tilt and Zoom, por sus siglas en inglés) desde el software LabVIEW para manejarlo a través del joystick (FIGUEROA SARCOS, 2015).

En 2013, en la Universidad de las Fuerzas Armadas ESPE, según (LEÓN VÁSQUEZ, 2013), se desarrolló el diseño e implementación de un robot móvil autónomo y teleoperado para labores agrícolas, en su primera fase para la fumigación de plantaciones de rosas, que en condiciones ideales en un campo de pruebas, se mostró funcional, sin fugas de líquido en las uniones entre las mangueras y las válvulas, sin inconvenientes en su modo manual pero con pequeñas fallas en el modo automático. Además, se concluyó que la técnica de control más adecuada para el manejo de motores de corriente continua de baja potencia, es el control proporcional, pues no se requiere controlar el arranque de los mismos, sino su velocidad de funcionamiento, lo que se hace mediante el uso de PWM (Pulse Width Modulation) configurados digitalmente en el controlador.

1.2. Justificación Teórica

La producción agrícola en Ecuador es primordial en los ejes de desarrollo del país, debido a sus regiones y microclimas que permiten la producción de varios tipos de frutas, legumbres, cereales, flores, entre otros. En este ámbito, la mora de castilla es una fruta de gran adaptabilidad en zonas de la sierra que cumplen con las características ambientales necesarias para su germinación, desarrollo y muy buena producción.

En Ecuador, las organizaciones de investigación del sector agropecuario, se han enfocado principalmente en alcanzar la mayor productividad de los cultivos. Es así que mucha de la oferta tecnológica se ha basado en la provisión de material genético (semillas) y recomendaciones de prácticas de cultivo (SÁNCHEZ, y otros, 2019).

Actualmente, los procesos de aplicación de pesticidas y fertilizantes a los cultivos, crece constantemente, y los elementos que éstos compuestos contienen, pueden afectar a las personas que trabajan en la manipulación de los mismos. A pesar de que hay normativas de trabajo y uso de equipos de seguridad para labores que implican exposición a elementos químicos toxicológicos, muchos de los trabajadores hacen uso omiso de estos implementos y esto hace que exista un riesgo a la salud de los agricultores por el uso permanente y el manejo inadecuado de plaguicidas químicos, para el control fitosanitario de los cultivos (JIMENEZ-QUINTERO, y otros, 2016).

Es evidente que la agricultura nacional actual, requiere de innovaciones tecnológicas, basadas en la utilización de ciencias debidamente integradas como la física, agronomía, química, robótica, entre otras, que en coordinación con una gestión de conocimiento, permita generar una mayor competitividad del sector, mejorar la calidad del ambiente de trabajo de los agricultores y la rentabilidad de la producción agroalimentaria (SANTOS, 2018).

El desarrollo de este prototipo, pretende introducir el manejo de herramientas tecnológicas en el proceso de producción de la mora de Castilla y ayudar a precautelar la salud de los agricultores que están expuestos constantemente a compuestos químicos y cargas pesadas durante el proceso de fumigación agrícola, lo que repercute en su salud y puede causar enfermedades graves a largo plazo.

1.3. Justificación aplicativa

El robot móvil terrestre autónomo para fumigación se desarrollará en base a dispositivos electrónicos y herramientas de programación específicas y de alta eficiencia como es la tarjeta de desarrollo Raspberry Pi, para el procesamiento de la información y la programación de algoritmo del robot. En base a las tareas principales del prototipo como la navegación autónoma dentro del cultivo de mora de castilla y la aplicación del líquido desde su contenedor hacia las plantas de manera controlada, se requiere de un sistema de control para el funcionamiento de la bomba encargada del sistema de pulverización, para evitar que la cantidad suministrada esté por fuera de los rangos requeridos por el cultivo y de las especificaciones del agricultor o técnico agrícola. Se implementará un sistema de control por PWM para la variación del voltaje de alimentación de la bomba para regular el flujo de salida de la bomba. Mientras que, para los motores, en base a los datos obtenidos por el sensor LIDAR, que será la herramienta principal para la localización dentro del área de trabajo, se pretende implementar una red neuronal perceptrón multicapa (MLP) para el control de los motores mediante el entrenamiento con los datos obtenidos por el sensor LIDAR, para alimentar un algoritmo para generar una trayectoria adecuada y evitar colisiones dentro del cultivo. Además, mediante las librerías de ROS, se podrá facilitar la integración del software y el hardware del robot.

Al ser un prototipo móvil terrestre, se debe contar con un sistema de tracción que soporte el peso del robot y además permita su libre movilidad dentro del sembrío. Así, los motores deben tener el suficiente torque para mover el peso total de la estructura y el líquido con agilidad y de acuerdo a la potencia que pueda suministrar la fuente de energía prevista para cumplir este requerimiento.

La estructura metálica, proporcionará robustez al prototipo, de modo que se adecúe a las condiciones de la superficie del terreno y del ambiente, a la carga a transportar y a las dimensiones de los surcos del cultivo.

Se pretende que el robot tenga una autonomía entre los veinte a treinta minutos de trabajo, tiempo que será suficiente para la aplicación total del contenido de su tanque y además se pueda recargar el líquido en el caso de que el área del sembrío sea muy grande, para esto se plantea el uso de un banco de baterías, con un amperaje y capacidad de descarga acorde a los motores y dispositivos que se deban implementar.

La capacidad del tanque del robot prototipo será proporcional al volumen que disponen las bombas de fumigar básicas tipo mochila, con la finalidad de poder comparar el rendimiento del líquido en una aplicación manual respecto a una aplicación autónoma.

Los rociadores o boquillas, que serán los encargados de la aplicación del contenido líquido a las plantas, deberán tener un rango de alcance adecuado para cubrir completamente el forraje de la planta y asegurar que la fumigación sea completa. Para el proceso de bombeo del líquido desde el tanque hacia las boquillas de nebulización, será por medio de una bomba con una presión correspondiente al tipo de boquilla y de corriente continua, a diferencia del tipo de alimentación que tienen las bombas comerciales comunes, que usan combustibles como diésel o gasolina.

El esquema básico del funcionamiento del prototipo, se presenta en la ilustración 1-1.

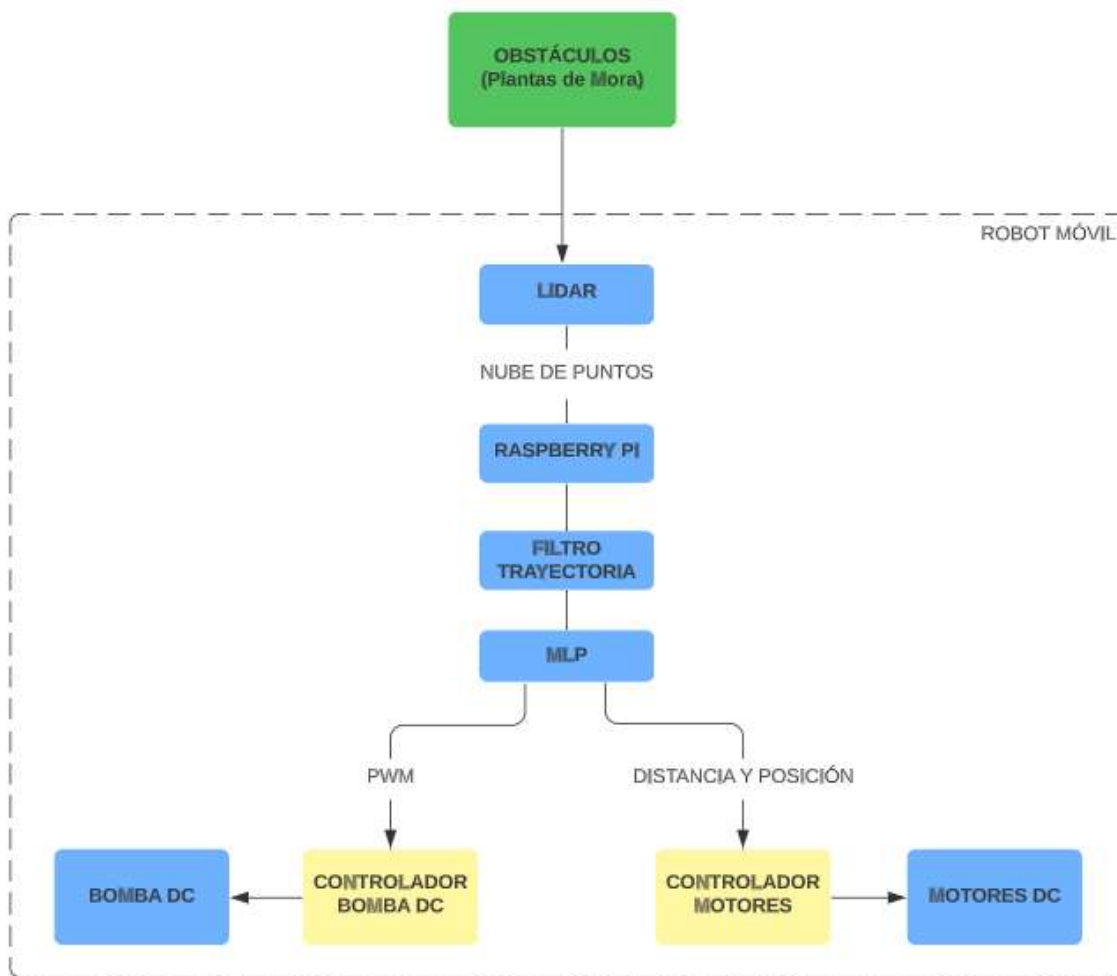


Ilustración 1-1: Diagrama de bloques del funcionamiento básico del robot móvil autónomo para fumigación en mora de Castilla

Realizado por: Christian Aguagallo, 2023

1.4. Objetivo General

Desarrollar un prototipo de robot móvil terrestre autónomo para fumigación agrícola en el cultivo de Mora de Castilla.

1.4.1. Objetivos específicos

- Investigar bibliografía referente a los estudios acerca de la implementación de la robótica en la agricultura y los beneficios que aporta a su desarrollo tecnológico.
- Seleccionar los dispositivos electrónicos que más se adecúen al prototipo, en base a las características de funcionamiento requeridas y del ambiente al cual estará expuesto.
- Desarrollar un prototipo robótico para fumigación agrícola que realice esta tarea de forma autónoma y con la mejor precisión posible.
- Evaluar la eficiencia del prototipo, de acuerdo a las especificaciones del cultivo y a los resultados de funcionalidad que se obtengan.

CAPÍTULO II

2. MARCO TEÓRICO

En el presente capítulo se realiza una revisión bibliográfica referente al cultivo de mora de castilla, abarcando la arquitectura de los surcos y las enfermedades más comunes en este tipo de fruto. Luego se analiza la fumigación agrícola, la importancia de la robótica en la agricultura y los conceptos de fumigación autónoma. A continuación, se presentan las características de los robots móviles, en base a la bibliografía disponible, sus sistemas de locomoción y las configuraciones cinemáticas, además de analizar los sistemas de navegación y localización más adecuadas para el proyecto. También se hace énfasis en las características de los sensores y controladores que se podrían implementar y se culmina el capítulo con la información de los sistemas de alimentación para robots móviles.

2.1. Cultivo de Mora de castilla

La mora de Castilla del género *Rubus Glaucus*, es una fruta generalmente cultivada en la sierra andina a lo largo del continente americano, con ciertos asentamientos en países europeos. Sus características de sabor, tamaño y resistencia, le permiten tener una buena producción durante su vida (INIAP, 2014). Las plantas de mora de castilla se siembran a una distancia promedio de 1.5m entre surcos y 1.2m a 1.5m entre plantas, se recomienda una distancia de 3 m entre surcos si se va a hacer uso de maquinaria agrícola convencional dentro del cultivo (INFOAGRO, 2022). La densidad de siembra es de alrededor de 2500 plantas por hectárea, en el caso de que el terreno presente pendientes, se recomienda aumentar las distancias de siembra (LÓPEZ CHOLOTA, 2018). De acuerdo a estas especificaciones, en la ilustración 1-2, se presenta un cultivo promedio de mora de castilla.



Ilustración 1-2: Cultivo de Mora de Castilla

Fuente: (GOBERNACION DE ANTIOQUIA, 2013)

La planta comienza a fructificar a los seis u ocho meses después del trasplante. Dependiendo del manejo y cuidado de la plantación, la planta presenta un período de 10 o más años de producción, la misma que aumenta a medida que crece y avanza en edad el cultivo (INIAP, 2014).

Esta fruta en todas sus variedades tiene una gran aceptación en el mercado, puesto que alrededor de un 92% de las familias riobambeñas gustan del consumo de este fruto (FONSECA, y otros, 2020).

2.1.1. Enfermedades en la mora de castilla

Debido a las características de la especie *Rubus Glaucus*, resulta sensible a enfermedades como la Botritis (o Moho gris), Antracnosis y el Oídium, entre otras (INFOAGRO, 2022). Que son las más comunes en este tipo de cultivo.

La Botritis causa la pudrición de los frutos, como se puede observar en la ilustración 2-2.



Ilustración 2-2: Botritis en la Mora de Castilla

Fuente: (ZULETA OSPINA, 2018)

La Antracnosis provoca que las ramas se sequen, resultando en una muerte regresiva de la planta, como se muestra en la ilustración 3-2.



Ilustración 3-2: Antracnosis en la Mora de Castilla

Fuente: (ZULETA OSPINA, 2018)

El oídium se presenta en las hojas, ramas jóvenes, peciolo y frutos, los tallos se tornan delgados y sus frutos se apiñan y deforman perdiendo así su valor comercial (CEVALLOS BERMEJO, 2020). En la ilustración 4-2 se presenta la deformación que causa el oídium en las hojas de la mora de castilla.



Ilustración 4-2: Oidium o Mildeo polvoso en plantas de mora de castilla

Fuente: (ASMOBEL, 2018)

Como respuesta al control de estas enfermedades, los agricultores llevan a cabo fumigaciones periódicas conforme lo recomiende el técnico agrónomo a cargo del cultivo. Además de las fumigaciones de control de enfermedades, se aplican fertilizantes, plaguicidas y sustratos, que, junto

al control hídrico del cultivo, permiten obtener plantas sanas, resistentes, con una producción rentable y de calidad (ROMO VINUEZA, 2022).

2.2. Fumigación Agrícola

La fumigación agrícola es un tratamiento de control de plagas y enfermedades basado en el uso de agroquímicos en los cultivos. Esta fumigación se la puede realizar de dos maneras, manual o mecanizada. En el caso de la fumigación manual, que se presenta en la ilustración 5-2, se hace uso de bombas de tipo mochila o en el mejor de los casos, una bomba de motor, que se carga en la espalda del agricultor o fumigador y se aplica en el líquido de manera directa a las plantas dentro del área de cultivo. Mientras que la fumigación mecanizada, que se observa en la ilustración 6-2, se realiza tanto por vía aérea, utilizando avionetas o drones, como por vía terrestre utilizando tractores acoplados para realizar esta labor correctamente (NAVIA ZAMORA, 2019).



Ilustración 5-2: Fumigación agrícola manual

Fuente: (JARDINEDIA, 2018)



Ilustración 6-2: Fumigación agrícola mecanizada

Fuente: (123RF, 2021)

Sin embargo, se debe destacar que el uso excesivo o inadecuado de productos químicos para el control de enfermedades puede tener efectos negativos en la salud de las plantas, la calidad de la producción y el medio ambiente, por lo que es necesario seguir las prácticas de manejo integrado de plagas solamente en el caso de ser necesario, además de combinarse con técnicas de control como la eliminación de plantas infectadas, la poda adecuada, rotación de cultivos y el uso de variedades de mora resistentes y adecuadas a la zona de cultivo (ZUMBA BRIONES, 2022).

2.2.1. Pulverización de compuestos agrícolas para fumigación.

La pulverización en la agricultura es el proceso de fraccionar una sustancia líquida en partículas pequeñas con la finalidad de que, al aplicarse en las plantas, estas partículas cubran completamente

el cultivo y se puedan obtener mejores resultados en el control y manejo fitosanitario (TECHNIDEA, 2019). En la ilustración 7-2 se presenta la fumigación de maíz por pulverización mediante el uso de maquinaria agrícola.



Ilustración 7-2: Pulverización agrícola

Fuente: (AGRICULTURERS, 2020)

Uno de los problemas existentes en la agricultura, es el incremento significativo del uso de herbicidas y pesticidas, debido a la presencia continua de enfermedades y plagas cada vez más resistentes, lo que puede deberse a la mala aplicación de los productos fitosanitarios mediante la pulverización hidráulica, que genera un rango de tamaños de gotas entre 900 μ m a <10 μ m (MERANI, y otros, 2021).

La dosis de aplicación de productos fitosanitarios se calcula regularmente con base al área de fumigación, generalmente expresado en gramos o litros por hectárea, pudiendo variar esta concentración según el volumen de agua que se utilice para su dosificación (MORENO F., y otros, 2021).

La dosis recomendada generalmente se refiere a equipos de pulverización normal con gastos de caldo de 1000 L/Ha, pero cuando se realizan aplicaciones con equipos de gasto reducido como los atomizadores o pulverizadores se requiere aumentar la concentración del producto fitosanitario, considerando también la velocidad de aplicación (METROFLOR-AGRO, 2021).

En el caso de las mochilas aspersoras para la aplicación de agroquímicos se recomienda una velocidad máxima de 2.5 Km/h a una presión entre 20 a 40 PSI, para evitar tanto la presencia de gotas grandes como la deriva (INTAGRO, 2022). La deriva o pérdida de líquido por evaporación o efectos ambientales se presenta cuando la gota sale del pulverizador, pero no llega al destino deseado, llegando a afectar a zonas que no requieren esa aplicación. Las partículas con diámetros de 100 a 200 micras derivan con mucha facilidad (NOYA, 2019).

El principal factor que más influye en la deriva es el viento, en relación directa con la velocidad de aplicación del producto. La velocidad adecuada es entre 1 y 2m/s (7km/h), porque con esta velocidad la gota recibe más energía y se produce una mejor penetración en el cultivo. A medida que aumenta la velocidad se debe aumentar el tamaño de la gota, utilizar boquillas anti deriva, etc., por lo que se recomienda no pasar de los 3m/s (BIZKAIA, 2020).

2.3. Robótica en la Agricultura

Los robots con aplicaciones en agricultura, operan en exteriores y en terrenos difíciles, tienen un conjunto único de desafíos de ingeniería que, no se presentan comúnmente en robots que trabajan en interiores y son el centro de los enfoques de diseño de los robots (CABRERA A, y otros, 2018). La finalidad de su uso es el reducir la exposición de las personas a labores pesadas y con riesgo biológico, además de agilizar los procesos agrícolas y permitir la innovación y el desarrollo de tecnologías enfocadas en la agricultura de precisión. En la ilustración 8-2 se puede observar al robot móvil “RB-VOGUI+ DUAL” de la empresa Robotnik en funcionamiento en un viñedo para el proyecto Bacchus, donde el robot con dos brazos UR integrados, imita la mecánica que realiza un operario para vendimiar, que es el proceso donde se corta, se recoge el racimo o se espera al punto óptimo de cosecha (UNIVERSAL ROBOTS, 2022).



Ilustración 8-2: Robots para agricultura

Fuente: (UNIVERSAL ROBOTS, 2022)

Las investigaciones actuales incluyen áreas como la navegación autónoma, el procesamiento de imágenes y el manejo de terrenos difíciles en agricultura (BACA, y otros, 2022).

2.3.1. Fumigación autónoma en la agricultura

La fumigación autónoma es una técnica de aplicación de pesticidas y productos químicos en el cultivo, que se basa en el uso tecnologías avanzadas y autónomas como drones, robots y vehículos autónomos sin necesidad de la intervención de un operador humano (RODRÍGUEZ PULIDO, y otros, 2017). Esta técnica utiliza algoritmos de inteligencia artificial y sensores para determinar la cantidad y la dirección de la aplicación de pesticidas, lo que permite una aplicación más precisa y eficiente.

En la ilustración 9-2, se aprecia al robot XAG R150 realizando su labor de fumigación agrícola dentro de un cultivo de manera autónoma.



Ilustración 9-2: Robot agrícola terrestre

Fuente: (IBERICADRON, 2022)

2.4. Robots Móviles

Un robot móvil es un sistema electromecánico con la capacidad de desplazarse, interpretar, planear y ejecutar tareas; además de navegar de manera autónoma en un ambiente desconocido. Las parte que lo componen son una estructura rígida (arreglo cinemático), un sistema de actuadores, que dotan de movimiento a la estructura, y un sistema de control (CRUZ CANDO, 2020).

2.4.1. Sistemas de Locomoción

Los robots móviles se pueden clasificar de acuerdo al tipo de locomoción que utilicen para desplazarse y realizar sus tareas en un área determinada, como se presenta en la ilustración 10-2, en robots de ruedas 10-2(a), robots de patas 10-2(b) y robots de oruga 10-2(c).

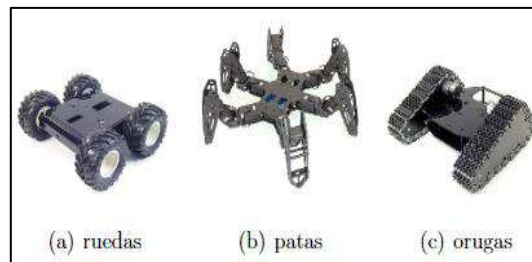


Ilustración 10-2: Clasificación de robots de acuerdo al tipo de locomoción

Fuente: (HERNANDEZ SANCHEZ, 2017).

- **Robots móviles con ruedas**

Este tipo de robots presentan mayor eficiencia energética en superficies duras y lisas, por lo que su utilización se ve más generalizada en la industria y en áreas como el almacenaje, la seguridad, maquinaria agrícola, educación, el ejército (RIVERA, y otros, 2019). Una ventaja que presenta el utilizar ruedas como medio locomotor es su facilidad de construcción, además, sólo es necesario suministrar energía al eje de las ruedas motrices; el robot siempre se encuentra en contacto con la superficie, así no presenta problemas de balance y no causa tanto desgaste en las superficies en las que trabaja (HUAMANCHAHA, y otros, 2021).

- **Robots de patas**

Los robots con patas se caracterizan por aislar el cuerpo del terreno empleando puntos discretos de soporte, consiguiendo movimientos omnidireccionales y reduciendo el deslizamiento. Así el robot puede superar obstáculos de manera más eficiente que un robot de ruedas, pero su desventaja es el mayor consumo de energía y que requiere algoritmos de control y planificación más complejos que otros tipos de locomoción (JIMÉNEZ CUESTA, 2014)

- **Robots tipo oruga**

Los robots tipo oruga resultan ser los más adecuados para labores en suelos inestables, ya que una de sus características es que integran bandas laterales para su desplazamiento, así se mejora la distribución de su peso para evitar estancamientos en terrenos con arena y grava, aunque su implementación por lo general resulta más costosa para máquinas de pequeñas dimensiones (ESCALONA, y otros, 2017).

De acuerdo a las especificaciones de cada sistema de locomoción, se pueden utilizar tanto ruedas como un sistema tipo oruga para el robot móvil planteado en este trabajo, pues, ambos sistemas pueden responder adecuadamente en labores agrícolas. Cabe recalcar que un robot tipo oruga tendría mejor desempeño en terrenos como los del cultivo de la mora de Castilla, aunque su costo sería una consideración importante en relación al sistema de ruedas, cuya implementación es más accesible y presenta una buena estabilidad en contacto con las superficies.

2.4.2. Configuraciones Cinemáticas

Existen algunas configuraciones cinemáticas para los robots móviles, estas se presentan en la figura 11-2, la aplicación de estas configuraciones dependerá principalmente de la funcionalidad del robot.

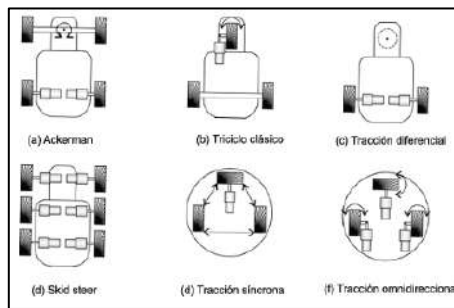


Ilustración 11-2: Configuraciones de los RMR

Fuente: (HUAMANCHAHA, y otros, 2021)

A continuación, se detallan las características de las principales configuraciones que se implementan en los robots móviles convencionales.

- **Configuración Ackerman**

Esta configuración es común en automóviles y vehículos eléctricos. Posee cuatro ruedas, las dos delanteras sirven para la dirección del robot, al efectuar un giro, la rueda interior presenta un ángulo mayor que la exterior, como se observa en la ilustración 11-2(a), de esta manera se evitan deslizamientos (JIMÉNEZ CUESTA, 2014).

- **Triciclo Clásico**

En esta configuración, la rueda delantera sirve tanto para la tracción como para el direccionamiento, mientras que el eje trasero con 2 ruedas laterales es pasivo, lo que permite que las ruedas giren

libremente, como se muestra en la ilustración 11-2(b). Tiene una mayor maniobrabilidad que la configuración Ackerman, pero presenta problemas de estabilidad en terrenos irregulares y en pendientes, su centro de gravedad suele desplazarse ocasionando pérdidas de tracción (OLLERO B., 2001).

- ***Tracción Diferencial***

Un robot móvil de tracción diferencial dispone de dos ruedas activas conectadas directamente o mediante un tren de engranajes a dos motores que se mueven al aplicar tensión a sus terminales, lo que induce velocidades angulares sobre las ruedas, produciendo cambios en la dirección del robot para que se desplace en línea recta, rote o cambie de dirección. Esta configuración se presenta en la ilustración 11-2(c). Además, esta configuración presenta mayor estabilidad en terrenos irregulares, mayor capacidad de carga, eficiencia y mejora la capacidad de maniobra (CARDONA, y otros, 2018).

La configuración adecuada para el prototipo será de tracción diferencial. Esta configuración permite trabajar con 2, 3 o 4 llantas y funcionar tanto con ruedas como en sistemas de oruga (ROBOTICOSS, 2021). Además, la tracción diferencial con 4 ruedas le dará más estabilidad en terrenos irregulares y transporte su peso.

2.5. Sistemas de navegación y localización en robots móviles

Un robot autónomo requiere de sistemas que le permitan determinar su ubicación y orientación para desplazarse adecuadamente en los entornos donde va a realizar sus actividades, además de poder detectar obstáculos y evitar colisiones con objetos presentes en el área (ZURITA, y otros, 2019).

2.5.1. Planificadores de trayectoria

Un planificador de trayectorias es un algoritmo que genera una secuencia de movimientos que un robot o vehículo autónomo deben seguir a lo largo de un camino para alcanzar un objetivo específico (PEREYRA, y otros, 2020). Los principales factores considerados son la geometría del entorno, la cinemática del robot, las restricciones de velocidad y aceleración, las prestaciones de los actuadores y los obstáculos que puedan presentarse en el camino (Universidad Miguel Hernández, 2018).

Un algoritmo de muestreo es un tipo de planificador que genera trayectorias mediante la selección aleatoria de puntos en el espacio de configuración del robot, denominados puntos de muestreo. A partir de la colección de estos datos, se implementa un algoritmo de conectividad para unir los puntos seleccionados y crear una trayectoria que puede ser suavizada para garantizar una trayectoria continua y suave para el robot (REY ARCENEGUI, 2019).

El algoritmo de muestreo de árbol rápido (RRT por sus siglas en inglés) es el algoritmo más común que utiliza una estructura de árbol para representar el espacio de configuración y construir una trayectoria óptima para desplazamiento del robot (PEREYRA, y otros, 2020). En las ilustraciones 12-

2 y 13-2 se presentan las progresiones de un RRT en distintos entornos, con densidad uniforme en las ramas sobre el espacio libre. Además, en la ilustración 14-2, se muestra la respuesta que presenta un RRT más avanzado con maniobras restringidas en un robot diferencial (LÓPEZ, y otros, 2016).

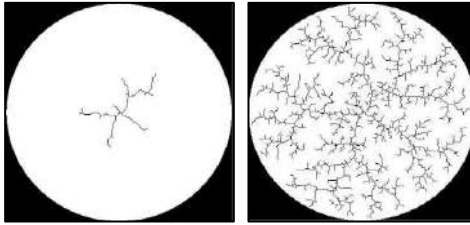


Ilustración 12-2: RRT en entornos simétricos

Fuente: (LÓPEZ, y otros, 2016)

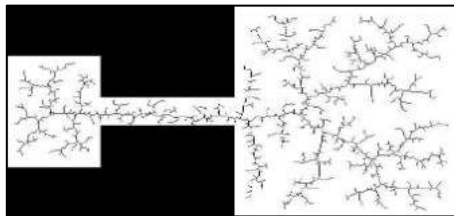


Ilustración 13-2: RRT en entornos asimétricos

Fuente: (LÓPEZ, y otros, 2016)

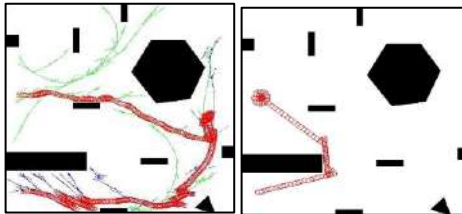


Ilustración 14-2: Aplicación de RRT a un robot con conducción diferencial

Fuente: (LÓPEZ, y otros, 2016)

El algoritmo RRT resulta muy útil para robots móviles que operan en entornos desconocidos o no estructurados, ya que puede adaptarse rápidamente a los cambios en el entorno (SALAS MEDINA, 2019). Las variantes de este algoritmo que se han desarrollado en los últimos años para mejorar la eficiencia de las trayectorias obtenidas, como el algoritmo Goal-bias RRT, el algoritmo Dynamic-RRT, el algoritmo RRT-Connect, Bi-RRT, Extend RRT, Local-tree RRT, etc. (TIAN, y otros, 2020).

2.5.2. Sistema de navegación basado en sensores

Los sistemas de navegación basados en sensores utilizan una variedad de dispositivos de medición para determinar la posición, velocidad y orientación de objetos o vehículos (SALAS, 2015).

Los sistemas de navegación basados en sensores pueden ser más precisos que los sistemas de navegación basados en satélites, como el GPS, ya que los sensores pueden proporcionar datos continuos y en tiempo real sobre la posición y la orientación del objeto (DE LA CRUZ SUSTAITA, 2017).

En la ilustración 15-2 se presentan los elementos más importantes del robot Pioneer 3DX, que utiliza un sistema de navegación basado en sensores y el software ARIA para el control y procesamiento de los datos (ROBOTS, 2022).

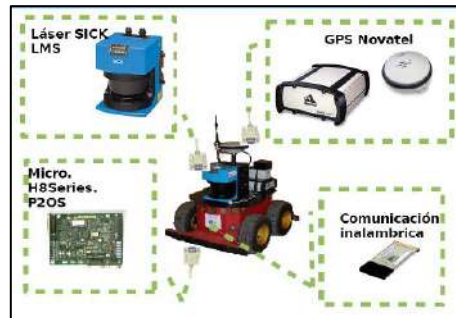


Ilustración 15-2: Elementos del robot móvil Pioneer

Fuente: (FLYABILITY, 2021)

Sin embargo, estos sistemas pueden ser más susceptibles a errores y desviaciones debido a la influencia de factores externos, como las condiciones climáticas y las interferencias electromagnéticas (SALAS, 2015).

2.5.3. Sistema de navegación basado en mapas

Para que un robot pueda trasladarse desde un punto a otro de manera autónoma, es necesario contar con un mapa del entorno, estático o dinámico, y también se requieren las coordenadas del objetivo en el mapa y la posición y orientación actual del robot en tiempo real (REY ARCENEGUI, 2019). El uso de técnicas como SLAM (Simultaneous Localization and Mapping), permiten crear mapas del entorno del robot y determinar su posición actual en todo momento, de esta manera el robot puede navegar adecuadamente ejecutando las órdenes asignadas, incluso sin tener información previa de la zona (DEL PINO BASTIDA, 2021). Para lograr esto, el robot puede requerir de sensores que le ayuden a obtener datos de su entorno. La ilustración 16-2 muestra un mapa generado por SLAM durante un experimento realizado en el DARPA Subterranean Challenge.

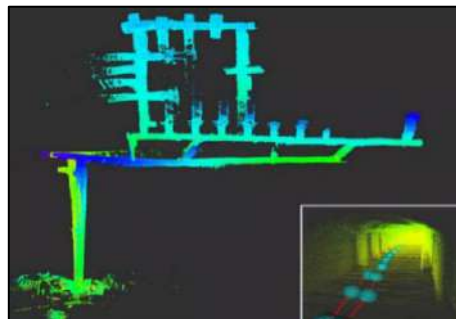


Ilustración 16-2: Experimento SLAM multi-robot - DARPA Subterranean Challenge

Fuente: (FLYABILITY, 2021)

Se propone implementar un sistema de navegación basado en mapas a partir de la información del sensor LiDAR. De esta manera, se podrá generar una trayectoria en tiempo real en función de la posición actual del robot.

2.6. Tarjetas de desarrollo

Las tarjetas de desarrollo son componentes electrónicos que trabajan en base a un microcontrolador o microprocesador y se utilizan para distintas aplicaciones como robótica, IOT, automatización industrial, educación, experimentación y electrónica de consumo (GIOVINO, 2020). Se caracterizan por su versatilidad, facilidad de programación, su amplia variedad de puertos y pines para trabajar con sensores, actuadores y otros dispositivos electrónicos.

- **Arduino**

Es una tarjeta de desarrollo basada en una placa electrónica de hardware libre, con un microcontrolador reprogramable, pines y puertos compatibles con la mayoría de los componentes electrónicos que se utilizan actualmente en proyectos de electrónica y robótica, resultando muy popular debido a su facilidad de uso y programación en su propio IDE de Arduino (ARDUINO, 2022).

La tarjeta Arduino UNO, más empleada en proyectos de robótica, se expone en la ilustración 17-2.

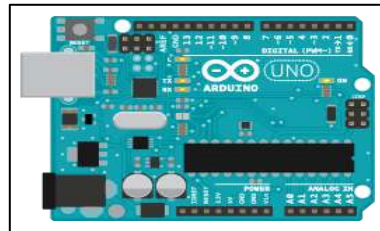


Ilustración 17-2: Arduino UNO

Fuente: (ARDUINO, 2022)

Existen en el mercado distintos modelos de tarjetas Arduino, con propósitos diferentes de acuerdo al tipo de microcontrolador, tamaño y número de pines, en la tabla 1-2 se presenta una comparación entre las principales tarjetas de Arduino.

Tabla 1-2: Comparación entre Tarjetas de desarrollo Arduino

Características	Arduino Nano	Arduino Uno	Arduino Mega	Arduino Leonardo	Arduino Due
Microcontrolador	ATmega328P	ATmega328	ATmega2560	Atmega32u4	SAM3X8E ARM
Memoria Flash	32 Kb	32 Kb	256 Kb	32 Kb	512 Kb
Terminales digitales	14/6	14/6	54/15	20/7	54/12
Terminales analógicos I/O	8/0	6/0	16/0	12/0	12/2
Voltaje de alimentación	5V	5V	5V	5V	5V
Voltaje de entrada	7-9V	7-12V	7-12V	7-12V	7-12V

SRAM	2 Kb	2 Kb	8 Kb	2 Kb	96 Kb
Tamaño	44x18mm	69x53mm	102x54mm	69x53mm	102x54mm

Realizado por: Aguagallo Christian, 2023

- **Raspberry PI**

La Raspberry Pi es una computadora de placa única que corre un sistema operativo Linux, muy popular entre desarrolladores de software libre. Se utiliza para una amplia gama de proyectos de robótica, proyectos educativos, servidores web y base de datos. Además de tener un bajo costo y una amplia disponibilidad de pines, puertos y excelente capacidad de procesamiento, por lo que se implementa en varias industrias en todo el mundo (RASPBerry, 2019). En la ilustración 18-2 se presenta la última versión de la placa de desarrollo Raspberry, la Raspberry Pi 4 Model B.



Ilustración 18-2: Raspberry Pi 4 Model B

Fuente: (PI AUSTRALIA, 2019)

Actualmente existen varios modelos de Raspberry Pi, los principales se comparan en la tabla 2-2.

Tabla 2-2: Comparación entre los principales modelos de Raspberry Pi

Características	3 Model A+	3 Model B+	4 Model B	Zero
CPU	1.4 GHz QUAD ARM Cortex-A53	1.4 GHz QUAD ARM Cortex-A53	1.5 GHz QUAD ARM Cortex-A72	1 GHz QUAD ARM 1176JZF-S
GPU	Video Core IV	Video Core IV	Video Core IV	Video Core IV
Memoria RAM	512 Mb	1 Gb	1,2,4 u 8 Gb	512 Mb
Alimentación	2.5A 12.5W/5V Micro USB GPIO	2.5A 12.5W/5V Micro USB GPIO PoE (HAT)	3A 12.5W/5V Micro USB GPIO	2.5A 12.5W/5V Micro USB GPIO
Puertos	1x USB 2.0	4x USB 2.0	2x USB 3.0, 2x USB 2.0	1x USB Micro
Red	Dual-band WIFI, BT	ETH10/100/300(USB) Dual-band WIFI, BT	ETH1000 Dual-band WIFI, BT	No
Tamaño	65x56 mm	85x56 mm	85x56 mm	65x30 mm

Realizado por: Aguagallo Christian, 2023

- **NVIDIA Jetson**

El NVIDIA Jetson es un sistema en módulo (SOM) de computación embebida de alta eficiencia energética diseñado para el desarrollo de aplicaciones de inteligencia artificial y aprendizaje automático en dispositivos IoT y sistemas embebidos. Es compatible con una variedad de bibliotecas y frameworks de aprendizaje automático (NVIDIA, 2022).

En la ilustración 19-2 se observa la tarjeta NVIDIA Jetson Nano, la más popular de la marca.

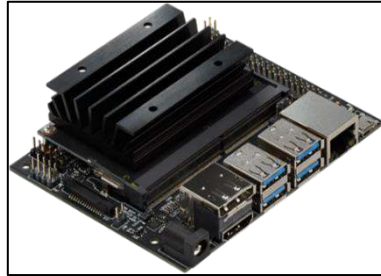


Ilustración 19-2: NVIDIA Jetson Nano
Fuente: (DUSTIN, 2019)

En la tabla 3-2 se muestra una comparación entre los principales módulos Jetson.

Tabla 3-2: Comparación entre las tarjetas Jetson

Características	Jetson Nano	Jetson TX1	Jetson TX2	Jetson AGX XAVIER
CPU	QUAD ARM Cortex-A57 MPCore	QUAD ARM Cortex-A57 MPCore	Dual-Core NVIDIA Denver2 64-Bit CPU and Quad-Core ARM Cortex-A72 MPCore	8-Core ARM v8.2 64-Bit NVIDIA Carmet CPU
GPU	128-core NVIDIA Maxwell GPU	256-core NVIDIA Maxwell GPU	256-core NVIDIA Pascal GPU	512-core NVIDIA Volta GPU – 512 Tensor Cores
Memoria RAM	4 Gb 64-Bit LPDDR4	4 Gb 64-Bit LPDDR4 25.6 Gb/s	4-8 Gb 128-Bit LPDDR4	16 Gb 256-Bit LPDDR4x 137 Gb/s
Puertos	3xUSB 3.0, 1xUSB 2.0 Micro B, HDMI GPIO, I2C, SPI, UART	4x USB 2.0	2x USB 3.0, 2x USB 2.0	1x USB Micro
Alimentación	5V-2A 5W	10W	7-5-20W	10-30W
Almacenamiento externo	16 Gb	16 Gb	16-32 Gb	32 Gb
Tamaño	65x56 mm	50x87 mm	50x87 mm	100x87 mm

Realizado por: Aguinaldo Christian, 2023

- **BeagleBone Black**

BeagleBone Black es una plataforma de desarrollo de hardware libre de bajo coste, respaldada por la comunidad, para desarrolladores y aficionados. Se utiliza en una gran variedad de proyectos electrónicos. Dispone de un procesador potente, conectividad y flexibilidad de hardware y software (BEAGLEBOARD, 2023). En la ilustración 20-2 se aprecia la tarjeta BeagleBone Black.

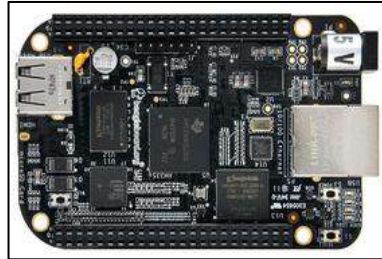


Ilustración 20-2: BeagleBone Black

Fuente: (BEAGLEBOARD, 2023)

Las características que presenta esta tarjeta de desarrollo se destallan en la tabla 4-2.

Tabla 4-2: Características de la tarjeta de desarrollo Beaglebone Black

Características	Jetson Nano
Procesador	AM335xARM A8 a 1GHz
GPU	SGX530
Memoria RAM	512 Mb DDR3
Puertos	2x USB 2.0
Alimentación	5V-2A 5W
Red	ETH10/100, RJ45
Almacenamiento interno	2 Gb
Tamaño	76.2x76.2 mm

Realizado por: Aguinaldo Christian, 2023

Al analizar las características de las principales tarjetas de desarrollo, se concluye que la Raspberry Pi 4 de 4 GB de RAM es la más adecuada para el robot móvil. Esto se debe a que se planea implementar herramientas como el SLAM y sensores que proporcionen una gran cantidad de datos para realizar un mapeo y generación de trayectorias adecuadas. Así, el robot podrá responder ágilmente y ejecutar sus funciones con eficiencia. Por lo tanto, no se consideran las tarjetas Jetson, ya que su capacidad de procesamiento es más elevada de lo necesario y su costo es muy elevado en comparación con otras opciones. Por otro lado, un Arduino no podría procesar tanta cantidad de datos, y la BeagleBoard tiene características inferiores a las de la Raspberry Pi.

2.7. Sistemas de control para robots móviles

- ***Controlador Proporcional***

El controlador proporcional ajusta la salida del motor en proporción a la entrada del sensor, lo que significa que cuanto mayor sea la desviación entre la entrada del sensor y el valor de referencia, mayor será la corrección aplicada al motor, así se pueden ajustar la velocidad y la dirección del robot en función de los datos obtenidos. Es uno de los controladores más simples de usar, aunque presenta ciertas limitaciones de precisión, requiere de una sintonización casi precisa, es sensible a las perturbaciones y no compensa el error en estado estable, lo que causaría que el robot tenga deficiencias en su movimiento y en el rendimiento general (GUTIÉRREZ, y otros, 2022).

- ***Controlador PID***

El control proporcional-integral-derivativo es la estrategia de control más empleada en la actualidad, se estima que más del 90% de los lazos de control emplean este tipo de control, a menudo con ganancia derivativa cero (BORASE, y otros, 2021) Este tipo de controladores, a diferencia del proporcional, también cuenta con un componente integral, que permite responder a los errores en la señal de entrada, además el componente derivativo permite responder a cambios o variaciones rápidas en la señal de entrada (ZAYAS-GATO, y otros, 2020).

- ***Controlador Difuso***

Los controladores basados en lógica difusa (FLC por sus siglas en inglés), permiten obtener respuestas dentro de un rango, asemejándose a la forma de trabajo del cerebro humano. Tienen la habilidad de capturar estrategias cualitativas de control y ofrecer un comportamiento de control altamente flexible, que permite que los sistemas puedan ajustarse a condiciones cambiantes o en ocasiones, difíciles de predecir (MEDINA, y otros, 2011). El empleo de la lógica difusa permite determinar la salida del motor en función de las entradas de los sensores, esto resulta útil cuando los sensores información variable y con cierta incertidumbre en los datos. El controlador FLC proporciona una respuesta precisa y rápida en entornos variables (KASSIR, 2015).

- ***Control por Redes Neuronales Artificiales***

Una red neuronal artificial (ARN por sus siglas en inglés), utiliza una red neuronal profunda, que se inspira en las neuronas del cerebro humano y animal, especialmente en el procesamiento del sistema visual de los mamíferos, obteniendo una capa de neuronas que pueden modelarse como una operación de filtrado lineal o convolución, seguida de una operación no lineal (RUIZ-DEL-SOLAR, y otros, 2018), su principal característica es que identifica automáticamente las características relevantes sin supervisión humana, empleando pesos compartidos y conexiones locales para aprovechar la

estructura de datos de entrada, empleando un número mínimo de parámetros y simplificando el proceso de entrenamiento (ALZUBAIDI, y otros, 2021).

Este tipo de redes neuronales, se han aplicado ampliamente en una variedad de campos diferentes, como visión por computadora, el procesamiento del habla, el reconocimiento facial, etc. (HUVAL, y otros, 2015). Para entrenar una red neuronal artificial para el control de un robot, se utiliza un conjunto de datos que incluyen escaneos LIDAR, así, cada escaneo LIDAR se convierte en una imagen 2D, que se usa posteriormente como entrada para la red neuronal, la red se entrena entonces para predecir las velocidades de los motores a partir de las imágenes previamente establecidas, además, se pueden aplicar técnicas de reducción de dimensionalidad para mejorar la eficiencia de la red neuronal (TEYMOURLOUEI, y otros, 2019), aunque su implementación resulta más costosa y más compleja que los otros controladores analizados.

2.7.1. Análisis comparativo de los sistemas de control analizados

Mediante la información presentada sobre los controladores que pueden resultar útiles para robots móviles, como el proporcional, proporcional integral derivativo (PID), controlador de lógica difusa (FLC) y los sistemas de control híbridos, basados en redes neuronales, se realiza una comparativa en la tabla 5-2 respecto a las ventajas y desventajas existentes en cada uno.

Tabla 5-2: Comparación entre sistemas de control para robots móviles

Controlador	Características	Ventajas	Desventajas
Proporcional	Controla la salida en proporción directa a la señal de entrada	Fácil de implementar, adecuado para sistemas estables y predecibles	Sensible a perturbaciones, no compensa el error de estado estable y presenta limitaciones en su precisión
PID	Controla la salida en proporción directa, integral y derivativa de la señal de entrada	Respuesta rápida, capaz de responder a cambios de la señal de entrada.	Complejidad de sintonización, en ciertas situaciones puede sufrir oscilaciones y overshoot
FLC	Utiliza lógica difusa para determinar la salida en función de la entrada de los sensores	Respuesta precisa, rápida en entornos variables y con datos inciertos, resulta eficiente cuando los valores de los sensores no son precisos	Requiere de un conocimiento especializado para su sintonización e implementación
Control por Redes Neuronales	Implementa una red neuronal para determinar la salida en función de los datos de entrada	Puede aprender y adaptarse a medida que la información que obtiene es mayor. Resulta ideal en entornos desconocidos o cambiantes en el tiempo	Es más complejo y costoso de implementar, también resulta menos transparente que otros controladores

Realizado por: Aguagallo Christian, 2023

Una de las características principales de las redes neuronales artificiales es su capacidad de aprendizaje (BASOGAIN, 2019). La arquitectura de una red neuronal perceptrón multicapa aprende a clasificar modelos por aprendizaje supervisado. Los modelos que suele clasificar suelen ser vectores con valores binarios y además presenta dos capas de unidades procesadoras, de las cuales sólo una tiene la capacidad de adaptar los pesos de las conexiones (VIVAS, y otros, 2018).

Debido a su flexibilidad y capacidad de aprender del error, una red neuronal resultaría más óptima para que el robot pueda trabajar en base a predicciones de una red neuronal preentrenada, de tipo perceptrón multicapa para que mediante los resultados que se obtengan, se puedan controlar los motores y así generar un movimiento autónomo adecuado.

2.8. Software para Robótica

Debido a los dispositivos y herramientas que se incorporan en el prototipo, se requieren trabajar con distintas herramientas de software que se detallan a continuación:

- ***Python 3.7***

Python es el lenguaje de programación ampliamente utilizado en proyectos, aplicaciones web, ciencia de datos, machine Learning, desarrollo de software, etc., su principal característica es su eficiencia y facilidad de implementación en distintas plataformas. Su versión 3.7 incluye todas las funciones de sintaxis, mejoras de intérprete, nuevas funciones de escritura para los desarrolladores que facilita la interacción con el lenguaje de programación y la estructura del código en su totalidad (PYTHON, 2021).

- ***Raspbian***

Raspbian también conocido como Raspberry Pi OS, es un sistema operativo optimizado específicamente para la Raspberry Pi, esto permite que el usuario tenga una experiencia sencilla y familiar en el desarrollo de proyectos que incluyan una Raspberry Pi. Además, Raspbian tiene una amplia comunidad de usuarios y soporte técnico, que facilita la ayuda en caso de problemas con la tarjeta de desarrollo. En algunos casos puede tener un mejor rendimiento que otros sistemas operativos disponibles (RASPBERRY PI, 2022)

- ***ROS (Robot Operating System)***

Es una plataforma de software de código abierto para el desarrollo de sistemas robóticos avanzados. ROS ha sido diseñado con un enfoque en la escalabilidad, la robustez y la seguridad, y es compatible con múltiples sistemas operativos y arquitecturas de hardware (LENTIN, y otros, 2022). Además, permite la integración de diferentes componentes, como sensores, actuadores, controladores, navegación, planificación de movimiento, entre otros, para construir sistemas robóticos personalizados y complejos (MARUYAMA, y otros, 2016). ROS tiene varias versiones, entre las cuales destacan ROS Melodic y ROS Noetic, que se diferencian porque Noetic es una evolución de ROS Melodic, debido

a que puede trabajar con Python 3 en lugar de Python 2, además que cuenta con una gran cantidad de librerías y herramientas que permiten a los desarrolladores crear, probar y depurar código de forma más eficiente (ROS.ORG, 2021).

Para implementar ROS en robots móviles, es necesario utilizar un sistema operativo compatible, como Ubuntu o Debian, y configurar el entorno de desarrollo adecuadamente. Los desarrolladores deben definir los mensajes y servicios que se utilizarán para la comunicación entre los diferentes componentes del robot. Además, es importante la configuración de los nodos, ya que son los procesos que se ejecutan en el robot y se comunican entre sí a través de ROS (RODAS CÓRDOVA, 2021).

2.9. Controladores de motor

Los controladores de motor son dispositivos específicamente diseñados para controlar la alimentación y la dirección de los motores. Son capaces de manejar altas corrientes y potencias, y ofrecen funciones de control de velocidad y dirección. El circuito eléctrico de un controlador generalmente lleva transistores de potencia o Mosfets, que son dispositivos que actúan como interruptores controlando el flujo de corriente a los devanados del inducido del motor (SMART-MOTOR DEVICES, 2020).

En la tabla 6-2 se realiza una comparación entre los controladores tipo puente H más utilizados en aplicaciones de robótica.

Tabla 6-2: Comparación entre controladores de motores DC.

Controlador	L298N	BTS7960	Sabertooth	RoboClaw
Potencia nominal	48W	Hasta 1200W	480W	2880W
Voltaje de operación	5-35V	5-27V	5-30V	6-34V
Corriente nominal	Hasta 2A	Hasta 43A	Hasta 32A	Hasta 60A
Canales de motor	2	2	2	2
Tipo de control	Velocidad y dirección	Velocidad y dirección	Velocidad y dirección	Velocidad y dirección
Frecuencia de PWM	Hasta 20KHz	Hasta 25KHz	Hasta 20KHz	Hasta 20KHz
Comunicación	UART, USB, etc	No dispone	UART, USB, etc	UART, USB, etc
Precio	Costo bajo	Costo medio	Costo medio	Costo medio a elevado

Realizado por: Aguagallo Christian, 2023

2.10. Sensores

Los sensores son dispositivos electrónicos de entrada que proveen una salida manipulable de la variable física medida, los sensores imitan la capacidad de percepción de los seres humanos e

interactúan con el entorno para proporcionar información de ciertas variables para procesarlas o activar procesos (QUIMIS SUÁREZ, 2021) (RAMÍREZ C., y otros, 2014).

- ***Sensores de movimiento***

Los sensores de movimiento son dispositivos electrónicos diseñados para detectar la presencia de movimiento en una zona determinada. Estos sensores utilizan una variedad de tecnologías para detectar el movimiento, como sensores infrarrojos pasivos (PIR), microondas, ultrasonido y cámaras (ESCALONA, y otros, 2017). En la ilustración 21-2 se muestra el sensor HC-SR501 que es un sensor compuesto por un sensor PIR y un circuito de estabilización y control, que le permite trabajar como un sensor digital directo (PROMETEC, 2018).



Ilustración 21-2: Sensor HC-SR501

Fuente: (PROMETEC, 2018)

- ***Sensores ultrasónicos***

Los sensores ultrasónicos son dispositivos electrónicos que utilizan ondas sonoras de alta frecuencia para detectar objetos y medir distancias. Los sensores ultrasónicos emiten una señal de sonido de alta frecuencia y miden el tiempo que tarda la señal en rebotar en un objeto y volver al (PINEDA-TORRES, 2019). Tienen la ventaja de ser no invasivos y no emitir radiación, sin embargo, pueden tener limitaciones en entornos ruidosos o en presencia de objetos que absorben o reflejan las ondas sonoras (BARRALAGA, 2021). En la ilustración 22-2 se presenta el sensor HC-SR04, común en proyectos de detección de objetos y medición de distancias.



Ilustración 22-2: Sensor Ultrasónico HC-SR04

Fuente: (NAYLAMP, 2021)

Estos sensores se utilizan comúnmente en aplicaciones que requieren medición de distancia y detección de objetos, como sistemas de seguridad, detección de obstáculos en robots y vehículos autónomos, control de nivel de líquidos y medición de espesor de materiales (ROCKWELL AUTOMATION, 2022).

- ***Sensores infrarrojos***

Los sensores infrarrojos (IR) son dispositivos electrónicos que detectan la radiación infrarroja emitida por objetos cercanos y los convierten en una señal eléctrica. Los sensores infrarrojos activos emiten una señal de luz infrarroja y miden el tiempo que tarda la luz en rebotar en un objeto y volver al sensor (ESCALONA, y otros, 2017). Mientras que, los sensores infrarrojos pasivos (PIR) detectan el cambio en la radiación infrarroja emitida por los objetos en su campo de visión (TORRES TAPIA, y otros, 2019). En la ilustración 23-2 se presenta el sensor infrarrojo JS200XF de la empresa JSUMO, que es un sensor infrarrojo de largo alcance y amplio ángulo de detección, pues puede medir hasta 2m con un tiempo de respuesta de 1.68 milisegundos (JSUMO, 2022).



Ilustración 23-2: Sensor infrarrojo de largo alcance JS200XF

Fuente: (JSUMO, 2022)

- **Sensores LIDAR**

El sistema LIDAR, por sus siglas en inglés (Ligh Detection and Ranging), es una tecnología de medición mediante el reflejo de luz láser de un objeto, utilizando una sola luz coherente en un espectro y frecuencia a través de la radiación electromagnética. El haz tiene una pequeña divergencia y alta intensidad para poder alcanzar objetos a largas distancias. Este método de medición de distancia se divide en dos categorías, a saber, tiempo de vuelo (ToF) y triangulación (RIVAI, y otros, 2020). La luz viaja a una velocidad de unos 300.000 kilómetros por segundo o 0,3 metros por nanosegundo en el vacío. La diferencia de tiempo entre la luz emitida y la recibida se puede utilizar para determinar la distancia del objeto (HUTABARAT, y otros, 2019).

En la ilustración 24-2 se observa el sensor YDLIDAR X4 de la empresa YDLIDAR, que es un sensor muy eficiente en aplicaciones de escaneo y posicionamiento.



Ilustración 24-2: YDLIDAR X4

Fuente: (YDLIDAR, 2018)

La utilización de datos LIDAR puede suponer un avance para la agricultura en esta dirección. Por ejemplo, a partir de los datos registrados por un sistema LIDAR se pueden calcular valores dendrométricos de los árboles frutales que forman las explotaciones. (ALANIZ, y otros, 2022).

2.10.1. Análisis comparativo de los sensores analizados

En la tabla 7-2 se presenta una comparativa de las características de los sensores revisados. Los voltajes de consumo pueden variar dependiendo del modelo específico de sensor que se utilice.

Tabla 7-2: Comparación entre los sensores de detección para robots móviles

Características	Movimiento	Infrarrojo	Ultrasónico	LIDAR
Tipo de tecnología	PIR (infrarrojo pasivo)	IR activo y pasivo	Ultrasónico	Láser
Rango de detección	< 15m	< 10m	< 10m	< 100m
Precisión	baja	alta	media	alta
Ángulo de detección	180°	120°	30°-60°	360°
Voltaje de alimentación	4.5V-20V	2.7V-5.5V	3.3V-5V	5V-12V
Voltaje de consumo	5V-20V	3.3V-5V	3V-5V	5V-12V

Realizado por: Aguagallo Christian, 2023

Se puede observar que las características del sensor LIDAR son mucho mejores en cuanto al rango de detección, precisión y ángulo de detección, por lo que se considera el más favorable para obtener datos e información del medio donde se encuentre el robot.

2.11. Actuadores

Los actuadores son los dispositivos que realizan la transformación de energía en movimiento, en función de la señal de entrada que puede ser eléctrica, hidráulica o neumática (PALLÁS Areny, 1993), para realizar un proceso y trabajar con el entorno del robot. A continuación, se describen los principales actuadores que trabajan en un robot móvil para fumigación agrícola.

2.11.1. Motores DC

Un motor DC o de corriente continua es una máquina que transforma energía eléctrica en mecánica. Además, un motor de corriente continua puede ser controlador por un driver específico según la aplicación que tenga, su configuración o las prestaciones que presente (ROLDÁN VILORIA, 2014). En la tabla 7-2 se realiza una comparación de los motores con escobillas, sin escobillas y motores paso a paso, comunes en aplicaciones dentro de la robótica.

Tabla 8-2: Comparación entre motores DC para robots móviles

Características	Motor de escobillas	Motor sin escobillas	Motor paso a paso
Funcionamiento y control de velocidad	Emplea escobillas para transferir la energía eléctrica al rotor	Utiliza un controlador electrónico para	Usa un sistema de imanes y bobinas para

		transferir la energía eléctrica al rotor	mover el rotor en pasos discretos
Voltaje de alimentación (DC)	3V-48V	12V-48V	3V-48V
Nivel de ruido	Alto (debido al desgaste de las escobillas)	Bajo a medio	Bajo a medio
Capacidad de carga	Hasta 20 Kg	Hasta 60 Kg	Hasta 10 Kg
Peso	0.5 – 10 Kg	0.5-5 Kg	0.3 – 5 Kg
Velocidad de rotación	Hasta 15000 RPM	Hasta 10000 RPM	Hasta 2000 RPM

Realizado por: Aguinaldo Christian, 2023

Cada fabricante tiene distintos valores de voltaje, capacidad de carga, peso, etc., por eso se definen rangos aproximados para cada tipo de motor.

2.11.2. Dimensionamiento de un motor DC

Para dimensionar un motor eléctrico de corriente continua, se requiere determinar el valor de la potencia que debe tener, así como el torque instantáneo.

La ecuación 1 describe el torque que se requiere para mover un objeto en estado estacionario:

$$T = F * r \quad [1]$$

Donde:

T: Torque instantáneo

F: Fuerza que se requiere para mover el objeto o vehículo (N).

r: Radio de la rueda (m)

Para calcular la fuerza requerida para mover una masa desde su estado de reposo, se requiere considerar su peso, como se muestra en la ecuación 2:

$$F = M * g \quad [2]$$

Donde:

F: Fuerza aplicada al movimiento

M: Masa del objeto

Finalmente se reemplaza el valor resultante en la ecuación 1 y se obtiene el torque que debe tener el motor. Cabe recalcar que, al usarse 2 motores, el torque resultante debe dividirse para 2 y entonces se tendrá el valor de torque para cada motor.

Mientras que, para determinar la potencia de cada motor para cumplir con los requerimientos de velocidad y carga a transportar, se debe aplicar la ecuación 3:

$$P = F * v \quad [3]$$

Donde:

P: Potencia

F: Torque calculado para cada motor

v: velocidad del prototipo (m/s)

Al reemplazar los datos, se obtienen los valores de Potencia y torque del motor, además se debe establecer un voltaje de acuerdo a las necesidades del prototipo (LEÓN, 2021).

2.11.3. Bomba DC

Las bombas DC que desplazan fluidos a través de accionamiento mecánico se clasifican en bombas volumétricas y bombas rotodinámicas. Las bombas volumétricas se implementan comúnmente debido a que generan un aumento de presión empleando variaciones de volumen en las cámaras donde se desplazan los fluidos, con la ventaja de que existen distintos mecanismos que se usan para producir ese efecto hidrostático (MÉRIDA LEÓN, 2019). En la tabla 8-2 se muestra una comparación entre las 3 bombas volumétricas más comunes, de acuerdo a sus características primordiales.

Tabla 9-2: Comparación entre los bombas DC volumétricas

Características	Bombas de diafragma	Bomba peristáltica	Bomba de engranajes
Funcionamiento	Dispone de un diafragma que se mueve hacia adelante y hacia atrás para comprimir el líquido	Tubo flexible que se aprieta y se suelta para mover el líquido	Dos engranajes que se mueven juntos para mover el líquido
Voltaje de alimentación (DC)	3V-24V	3V-24V	12V-48V
Nivel de ruido	Bajo	Bajo	Medio a alto
Control de flujo	Regulación de la frecuencia y el voltaje	Regulación de la velocidad y la presión	Regulación de la velocidad y la presión
Eficiencia energética	Media	Alta	Media
Peso	0,1 - 3 kg	0,1 - 0,5 kg	0,5 - 5 kg
Presión máxima	Hasta 4,5 bar	Hasta 1,5 bar	Hasta 10 bar
Caudal máximo	Hasta 12 L/min	Hasta 1 L/min	Hasta 30 L/min

Realizado por: Aguinaldo Christian, 2023

Estas características y rangos establecidos en la tabla, son una aproximación general, pues todo dependerá del fabricante y de aspectos como el tipo de líquido a transportar, el tamaño y capacidad del sistema, necesidades de mantenimiento y reparación, y de la frecuencia con que trabaje la bomba.

2.11.4. Sistema de Pulverización

2.11.4.1. Boquillas de Pulverización

El éxito de la aplicación de agroquímicos en un cultivo dependen de varios factores como las condiciones climáticas, el correcto funcionamiento de la maquinaria agrícola o la adecuada selección de equipos, como en el caso de las boquillas, pues son las que determinan la cantidad de producto aplicado, la uniformidad y la cobertura que alcance el producto en las plantas, así como la deriva durante la fumigación (INTAGRI, 2020). Las características que se deben considerar para el correcta elección de una boquilla son entre otras la resistencia a la abrasión, el diseño de la forma de la boquilla, el tamaño de gota producido el patrón de aspersión (MCCOY, y otros, 2021).

Los principales tipos de boquillas de acuerdo a la funcionalidad y eficiencia que presentan son las de abanico plano, deflectora, cono hueco y cono lleno, que se pueden observar en la ilustración 25-2.



Ilustración 25-2: Tipos de boquillas para pulverización

Fuente: (RESUSA, 2021)

En la tabla 9-2 se realiza una comparativa de las boquillas presentadas, considerando rangos de presión y velocidad de flujo dentro de los cuales se pueden trabajar con este tipo de dispositivos.

Tabla 10-2: Comparación entre boquillas de aspersión

Características	Abanico Plano	Cono Hueco	Cono Lleno	Deflectora
Distribución de gotas	Ancho y uniforme	Fino y concentrado	Grueso y uniforme	Muy grueso y uniforme
Ángulo de pulverización	80°-110°	30°-110°	60°-110°	120°
Tamaño de gota	Medio a grande (200-400 micras)	Pequeño a medio (100-300 micras)	Grande (300-600 micras)	Muy grande (800-1000 micras)
Deriva	Moderada a baja	Baja a moderada	Alta	Muy baja
Resistencia a la abrasión	Baja a media	Media a alta	Alta	Alta

Rango de presión	15-90 PSI	20-90 PSI	30-120 PSI	20-90 PSI
Rango de velocidad de flujo	0.2-0.8 GPM	0.2-1.0 GPM	0.2-1.2 GPM	0.2-0.8 GPM
Aplicación recomendada	Herbicidas, Insecticidas, Fungicidas	Insecticidas, Fungicidas	Fertilizantes, Herbicidas	Fertilizantes, Herbicidas

Realizado por: Aguagallo Christian, 2023

El rango de velocidad se estableció en galones por minuto (GPM por sus siglas en inglés), este rango dependerá de la presión y el flujo de líquido que se utilice, la distancia a la que se aplique la fumigación y del tamaño de la gota, que es el principal factor de deriva en las aplicaciones por aspersión (HOMER BANNISTER, y otros, 2020).

2.11.4.2. Tuberías

Para dimensionar las tuberías para el sistema de pulverización se requieren aplicar ecuaciones básicas de hidráulica y mecánica de fluidos, como la ecuación 4 que representa la ecuación del caudal, la ecuación de continuidad en su forma básica en la ecuación 5, que resulta ser la más importante pues considera que el caudal debe mantenerse constante a lo largo de un conducto.

$$Q = \frac{V}{t} \quad [4]$$

$$Q = v * A \quad [5]$$

Entonces para hallar el diámetro de la tubería, se despeja el área de la ecuación 5 y mediante los conceptos de área y diámetro se llega a la ecuación 6 que representa el diámetro de las tuberías que se deben implementar en el sistema de pulverización para trabajar con el flujo requerido.

$$A = \frac{Q}{v}$$

$$D = \sqrt{\frac{4A}{\pi}} \quad [6]$$

2.12. Sistemas de Alimentación

Los robots móviles requieren de sistemas de alimentación que les permitan operar de forma autónoma por un periodo adecuado sin necesidad de una fuente de energía extra (TIE, y otros, 2013). Estos sistemas pueden ser de diferentes tipos, en la tabla 10-2 se hace una comparativa de los principales sistemas como las baterías, paneles solares, combustibles y sistemas de inducción, la elección de la opción más adecuada depende de la finalidad del robot (MINNALEARN, 2022).

Tabla 11-2: Comparación entre los sistemas de alimentación para robots móviles

Características	Baterías	Paneles Solares	Combustible	Inducción
-----------------	----------	-----------------	-------------	-----------

Características eléctricas	Voltaje constante, CC o DC	Voltaje variable CC o DC	CC o AC, de acuerdo al tipo de motor o célula de combustible	Corriente alterna
Duración	Desde unas pocas horas hasta varios días, de acuerdo a su capacidad y la carga del robot.	Depende del clima, la ubicación y la capacidad del panel solar, desde unas pocas horas hasta varios días	Desde varios días hasta varias semanas, según el tipo de combustible, la capacidad del tanque y la carga del robot.	Depende de la eficiencia del sistema de carga y descarga del robot.
Eficiencia	Hasta 95%	Variable < 90%	Hasta 60%	Hasta 90%
Zonas de implementación	En cualquier zona, siempre y cuando se puedan cargar las baterías.	Zonas con alta exposición solar, como desiertos, playas y áreas abiertas	Zonas remotas o de difícil acceso, como desiertos, bosques y montañas	En cualquier zona, siempre y cuando haya una estación de carga por inducción
Ventajas	Alta densidad de energía, bajo peso, fácil de cargar, fuente de energía constante.	Fuente de energía limpia y renovable, silencioso	Alta eficiencia energética, larga duración de la operación, independencia energética	Requiere combustible para operar, emite gases y calor, necesita mayor espacio.
Desventajas	Tiempo limitado de operación, requieren recargas frecuentes, vida útil limitada.	Dependencia de la luz solar, necesita espacio y tiempo para cargar, no funciona bien en entornos nublados o con poca luz solar	Permite la carga de forma autónoma mientras está en movimiento, no se requiere contacto directo para cargar, no emiten gases	Requiere una estación de carga, la eficiencia disminuye con la distancia, presenta incompatibilidad con algunos robots

Realizado por: Aguagallo Christian, 2023

Se puede observar que, debido a las características presentes en la tabla anterior, el sistema más adecuado para la alimentación del robot sería por baterías, debido al tipo de corriente eléctrica, eficiencia y su factibilidad de implementación en la zona donde va a trabajar el robot. Será importante calcular adecuadamente la carga total del robot para que la duración de la batería cumpla con los requerimientos del prototipo.

2.12.1. Dimensionamiento del banco de baterías

El procedimiento para calcular la potencia y capacidad que debe tener un banco de baterías consiste en establecer el voltaje común que se requiere para los dispositivos dentro del prototipo, para que el banco de baterías pueda alimentarlos, además su capacidad se debe a la corriente total que resulta de la suma de las corrientes que requiere cada elemento. Entonces, de acuerdo a la corriente total y el voltaje general se establece la potencia que se necesita en el banco de baterías, generalmente esta potencia se calcula en W/h (SOLOELECTRONICOS, 2016).

Suelen presentarse errores típicos de conexionado que a corto plazo producen problemas devino a una conexión no óptima de las baterías como, por ejemplo, que cada conexión crea una pequeña resistencia sobre el cable, los terminales de engarzado y los terminales de tornillo de la batería (TRITEC-INTERVENTO, 2020).

Existen 3 maneras de conectar las baterías para obtener un banco acorde a las necesidades de la carga, en la ilustración 26-2 se muestran las 3 conexiones: serie, paralelo y mixta para baterías de 12V.

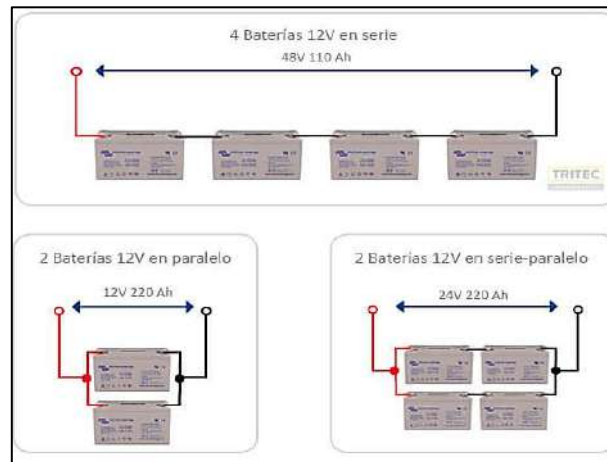


Ilustración 26-2: Conexiones para bancos de baterías de 12V

Fuente: (TRITEC-INTERVENTO, 2020)

Una mala conexión de los cables hace que en el caso de la ilustración 27-2(a) la batería superior proporcione menos corriente que la batería inferior, debido a las resistencias que se presentan durante todo el recorrido de la energía por el cableado, mientras que al cargar el banco de baterías como se muestra en la ilustración 27-2(b), la batería inferior se carga con una corriente más alta que la batería superior.

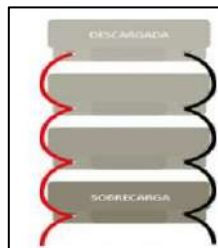


Ilustración 27-2: Conexión incorrecta de baterías tanto en descarga como en la carga

Fuente: (TRITEC-INTERVENTO, 2020)

Por lo tanto, se recomienda aplicar alguno de los conexionados de la ilustración 28-2, para que la corriente en cada batería sea la misma o trabaje de manera equilibrada.

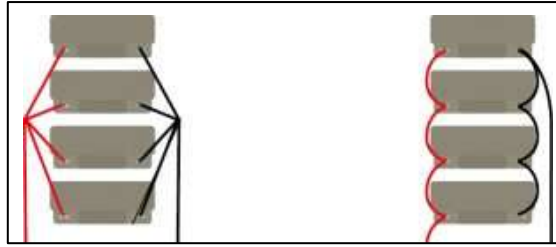


Ilustración 28-2: Conexiones adecuadas para un banco de baterías equilibrado

Fuente: (TRITEC-INTERVENTO, 2020)

Para calcular el tiempo que dura una batería se aplica la ecuación 7, en base a la capacidad del banco de baterías, la corriente de carga y la eficiencia de las baterías (QOITECH, 2022). Para esto se deben considerar las especificaciones técnicas del fabricante para evitar descargar la batería por debajo de su umbral de descarga y así prolongar el tiempo de vida útil de la batería.

$$TiempoBatería = \frac{Capacidad_Batería (Ah)}{Corriente_Carga (A)} * Eficiencia_Batería \quad [7]$$

Con estas consideraciones se puede establecer el tipo de conexión más adecuada para la alimentación del prototipo y dimensionar el espacio en la estructura para el banco de baterías.

CAPÍTULO III

3. MARCO METODOLÓGICO

En el presente capítulo se establecen los requerimientos, la concepción y arquitectura del prototipo, además de los elementos de hardware y software y los módulos que tendrá el robot para funcionar de manera autónoma.

3.1. Requerimientos del prototipo

De acuerdo a la información presente en el capítulo II, se presentan los requerimientos para el desarrollo del prototipo de robot para fumigación agrícola en un cultivo de mora de castilla.

3.1.1. *Requerimientos de Hardware.*

- Capacidad de traslación por los caminos entre surcos o hileras de entre 80cm a 100cm.
- Incluir un tanque de almacenamiento con capacidad de al menos 10 litros.
- Pulverización del líquido con un tamaño de gotas hasta 300 micras para que la aplicación de los productos agrícolas en la planta de mora de castilla sea completa.
- Sistema de bombeo de líquido con una presión adecuada para las boquillas, entre 30 a 45 PSI.
- Flujo de líquido adecuado para las boquillas de cono hueco, entre 0.25 a 0.5 litros por minuto.
- Distancia de aplicación desde las boquillas hacia las plantas de al menos 60cm, para cubrir la mitad del follaje desde cada lado.
- Sistema de fumigación acorde a la altura promedio de las plantas en el cultivo de hasta 2m.
- Velocidad de traslación del prototipo de 4Km/h (1.11m/s) dentro del cultivo.
- Sistema de locomoción apto para transportar el peso total del robot hasta 55 Kg.
- Sistema de alimentación que permita un funcionamiento adecuado por un tiempo aproximado de 20 minutos.
- Carcasa para la protección de los elementos y dispositivos electrónicos.
- Carcasa con ventilación para la Raspberry Pi.

3.1.2. *Requerimientos de Software.*

- Obtención de datos hasta 3m desde la posición actual del prototipo.
- Generación de una trayectoria por la mitad del camino entre hileras o surcos en base a los datos obtenidos en 2D del sensor LIDAR.
- Lenguaje de programación potente y que permita la implementación de múltiples librerías para el correcto funcionamiento del prototipo.
- Comunicación adecuada entre el sensor y la Raspberry pi 4, para evitar pérdidas de datos y trabajar con el Baud-Rate de 230400 del LD06.

3.2. Concepción de la arquitectura general del prototipo

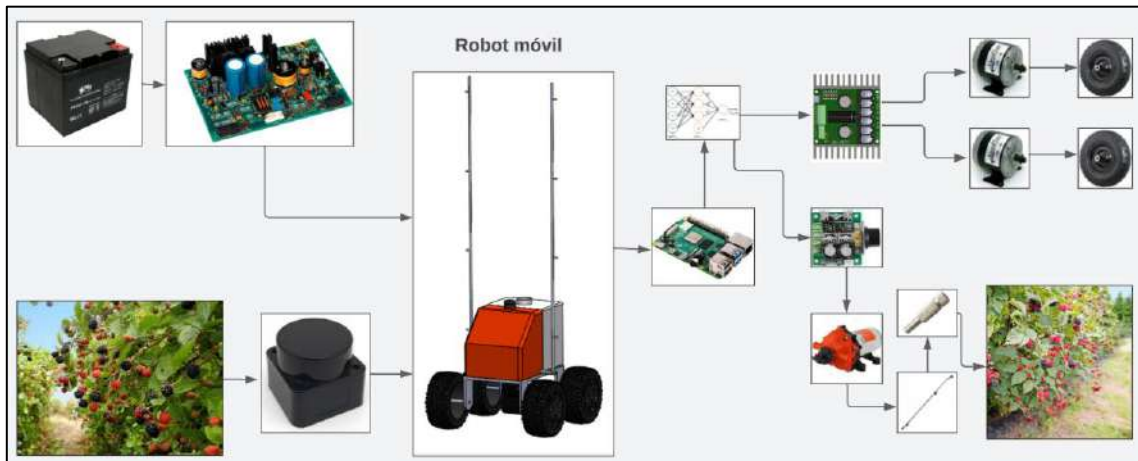


Ilustración 1-3: Concepción general del prototipo

Realizado por: Aguinaldo Christian, 2023

3.3. Diseño de los bloques del prototipo

Al analizar la arquitectura general del prototipo, se considera que se debe trabajar por bloques, de acuerdo a la funcionalidad que cada uno presenta en el robot.

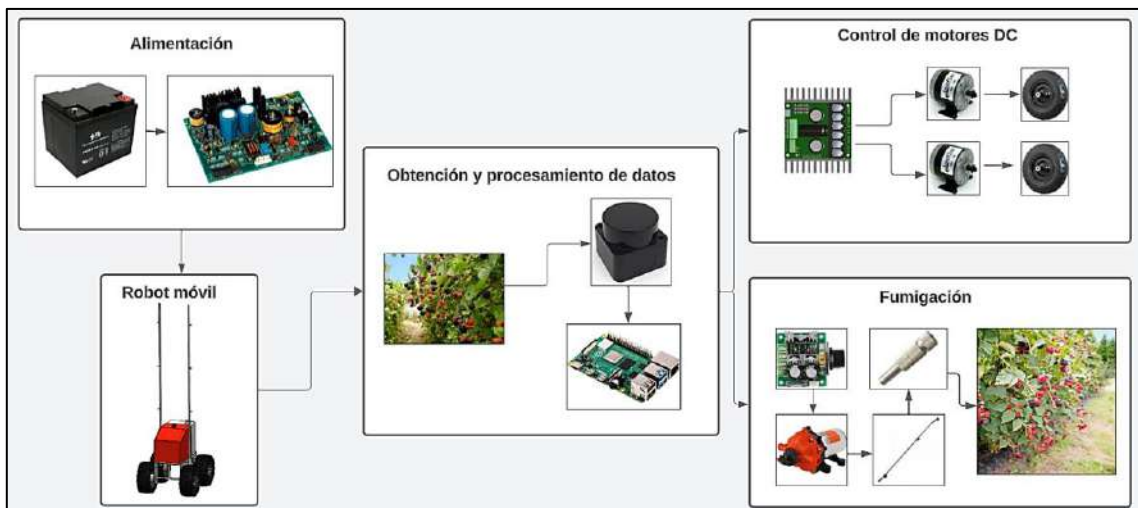


Ilustración 2-3: Arquitectura de los módulos del prototipo

Realizado por: Aguinaldo Christian, 2023

3.3.1. *Bloque de obtención y procesamiento de datos*

En este bloque se obtienen los datos mediante el sensor LIDAR. Se procesan estos datos mediante un filtro programado en la Raspberry Pi, con los datos filtrados se identifica el camino disponible entre hileras de plantas de mora, se genera una trayectoria y se calcula los valores de velocidad y dirección necesarios para mantenerse dentro de la trayectoria definida.

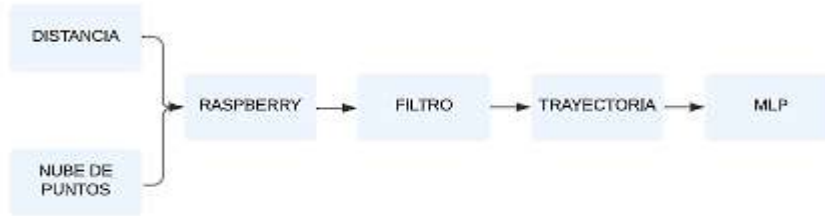


Ilustración 3-3: Diagrama de bloques para la obtención y procesamiento de datos
 Realizado por: Aguagallo Christian, 2023

3.3.2. *Bloque de control de motores*

Una vez calculados los valores de control, estos se envían al controlador de los motores DC a través de la Raspberry pi para establecer el movimiento de los motores de acuerdo a la velocidad y dirección que el robot debe tener para seguir la trayectoria establecida.

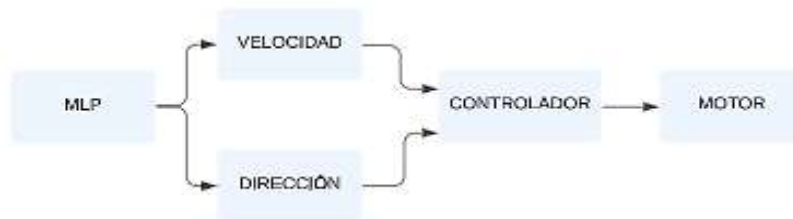


Ilustración 4-3: Diagrama de bloques para el control de motores
 Realizado por: Aguagallo Christian, 2023

3.3.3. *Bloque de fumigación*

En este bloque se encuentran la bomba DC que se activa de acuerdo a la posición del prototipo en el cultivo, que envía el líquido a las boquillas a una presión cercana a 40-45 PSI para que la pulverización se realice de acuerdo a las especificaciones de la gota.



Ilustración 5-3: Diagrama de bloques para la fumigación
 Realizado por: Aguagallo Christian, 2023

3.3.4. *Bloque de Alimentación*

Este bloque es el encargado de alimentar a los dispositivos y componentes electrónicos con los valores de voltaje y corriente adecuados. Se compone de baterías y reguladores de voltaje.



Ilustración 6-3: Diagrama de bloques para la alimentación del prototipo
 Realizado por: Aguagallo Christian, 2023

3.4. Estructura del prototipo

Para el diseño de la estructura del prototipo se hace uso de un software CAD de alto nivel para que la representación de los elementos sea lo más real posible, de acuerdo a sus dimensiones y la ubicación que se determina para cada uno en el ensamblaje final del prototipo. La ilustración 7-3 muestra el prototipo completo con los elementos ensamblados para visualizar la apariencia que tendría el robot.

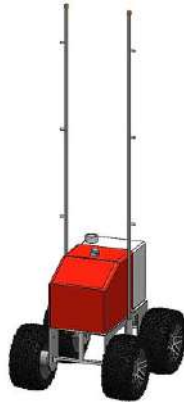


Ilustración 7-3: Diseño del prototipo de robot autónomo para fumigación agrícola en mora de castilla en un software CAD

Realizado por: Aguagallo Christian, 2023

Se parte de la ideal general del prototipo para diseñar la estructura tanto de la base como del ensamblaje de los elementos de hardware en el robot. Para lo que se toma en consideración el requerimiento de peso máximo del prototipo que son 55 Kg. Entonces la estructura de la base debe ser sólida y apta para soportar ese peso, por lo que en la ilustración 8-3 se muestra el diseño de la base del robot en acero inoxidable.



Ilustración 8-3: Base soporte del prototipo

Realizado por: Aguagallo Christian, 2023

Al tener la estructura de la base, se añaden las ruedas aptas para tracción en terrenos irregulares como en la ilustración 9-3.



Ilustración 9-3: Base con ruedas acopladas

Realizado por: Aguagallo Christian, 2023

Luego se añaden los motores como se observa en la ilustración 10-3, con su respectivo sistema de transmisión que permitirá no sobrecargar a los motores.



Ilustración 10-3: Motores acoplados a la base con sus soportes y poleas conectadas para el sistema de transmisión

Realizado por: Aguagallo Christian, 2023

Luego se ensamblan los elementos como: la bomba, el banco de baterías, el tanque y los tubos para el sistema de fumigación, como se muestra en la ilustración 11-3.



Ilustración 11-3: Acople de elementos en la base del prototipo

Realizado por: Aguagallo Christian, 2023

Luego se integra una carcasa metálica para proteger los elementos que integran el sistema electrónico y de alimentación, para en la parte superior instalar un sensor LIDAR, como se presenta en la ilustración 12-3.

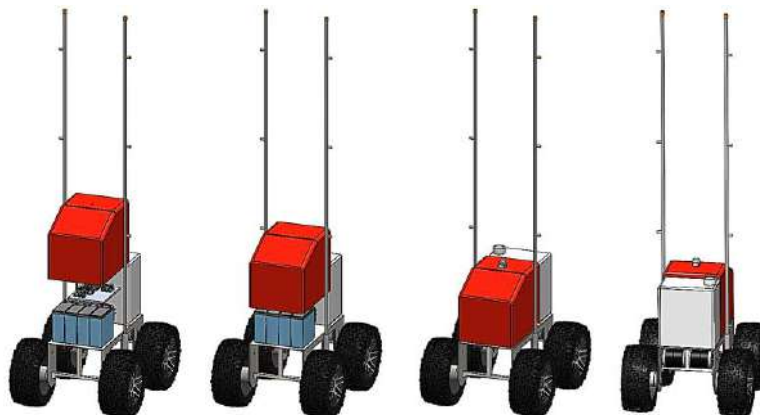


Ilustración 12-3: Ensamblaje de la carcasa metálica y el sensor LIDAR

Realizado por: Aguagallo Christian, 2023

3.5. Selección de los elementos que conforman los bloques

Luego de analizar los requerimientos de estructura y funcionalidad del sistema en general, y cada uno de los bloques que comprenden el prototipo, se seleccionan los elementos y dispositivos adecuados para que el robot cumpla adecuadamente su labor en el cultivo de mora de castilla.

3.5.1. Dispositivos de obtención y procesamiento de datos

Una vez analizados los requerimientos y los bloques que conforman el prototipo, y de acuerdo a la información del capítulo II se seleccionan los elementos y dispositivos para la implementación del prototipo de robot móvil para fumigación agrícola, las hojas de características técnicas necesarias se encuentran en el Anexo A.

3.5.1.1. Sensor Lidar LD06

Es un sensor de escaneo láser diseñado para aplicaciones de detección y mapeo de entornos. Se compone principalmente de un núcleo de medición láser que adopta la tecnología DTOF para medir 4500 veces por segundo. El LD06 emite un láser infrarrojo hacia adelante, el láser se refleja en la unidad receptora de fotón único al encontrar un objeto, así se obtienen los valores de tiempo de emisión y recepción, siendo este intervalo el tiempo de vuelo. Entonces emite una nube de puntos mediante comunicación inalámbrica (LDROBOT CO., 2021).

En la ilustración 13-3 se observa el sensor LD06 de la empresa LDROBOT.



Ilustración 13-3: Sensor LIDAR LD06

Realizado por: Aguagallo Christian, 2023

Las características técnicas des sensor LD06 se detallan en la tabla 1-3.

Tabla 1-3: Comparación entre los sistemas de alimentación para robots móviles

Características	Descripción
Voltaje	5V
Rango de Medición	12 m
Resolución angular	0.2° - 1.0°
Ángulo de medición	0° - 360°
Longitud de onda del láser	905 nm
Grado de protección	IPX – 4
Frecuencia de Muestreo	4500 Hz
Frecuencia de barrido	5 – 13 Hz
Dimensiones	38 x 38 x 34.3 mm

Realizado por: Aguagallo Christian, 2023

3.5.1.2. Tarjeta de desarrollo Raspberry pi 4

Es la cuarta generación de una serie de computadoras de placa única desarrolladas por Raspberry. Presenta una mayor potencia de procesamiento que las versiones anteriores, para ejecutar algoritmos de percepción, planificación y control necesarios en un robot autónomo (RASPBerry PI, 2019). Presenta una amplia variedad de puertos y opciones de conectividad, una gran flexibilidad y respaldo de una comunidad activa.

En la ilustración 14-3 se muestra la Raspberry Pi 4 de 4GB de RAM.



Ilustración 14-3: Raspberry Pi 4

Realizado por: Aguagallo Christian, 2023

Las características técnicas de la tarjeta de desarrollo Raspberry Pi 4 se detallan en la tabla 2-3.

Tabla 2-3: Características de la Tarjeta Rasberry Pi 4

Características	Descripción
Voltaje	5V
Corriente	3A
CPU	1.5 GHz QUAD ARM Cortex-A72
GPU	Video Core IV
Memoria RAM	1,2,4 u 8 Gb
Alimentación	3A 12.5W/5V Micro USB GPIO
Puertos	2x USB 3.0, 2x USB 2.0
Red	ETH1000 Dual-band WIFI, BT
Peso	45g
Dimensiones	85x56 mm

Realizado por: Aguagallo Christian, 2023

3.5.2. Dispositivos actuadores

3.5.2.1. Bomba de diafragma 12V DC

La bomba de diafragma SEAFLO DC, 12V es un dispositivo de bombeo de agua que se utiliza generalmente en sistemas de agua en embarcaciones, vehículos reactivos (RV), aunque también se

pueden implementar en aplicaciones agrícolas como sistemas de rociado agrícola, sistemas de purificación de agua, en la industria, entre otros (SEAFLO, 2018).

En la ilustración 15-3 se puede observar la bomba de presión de diafragma de agua Seaflo 12V DC.



Ilustración 15-3: Bomba de diafragma
Seaflo DC 33 Series

Realizado por: Aguinaldo Christian, 2023

Las características técnicas de la bomba Seaflo se detallan en la tabla 3-3.

Tabla 3-3: Características de la bomba Seaflo 12VDC.

Características	Descripción
Voltaje	12V
Corriente	3A (máximo 6A)
Caudal	3.0 GPM (11.4 LPM)
Tipo de bomba	Diafragma
Presión máxima	45 PSI (3.1 bar)
Material	Viton
Peso	4.81 libras
Dimensiones	198.736 mm x 127 mm x 117.348 mm

Realizado por: Aguinaldo Christian, 2023

3.5.2.2. Controlador de velocidad PWM 10A 12-40V

Es un controlador de motor que permite variar la velocidad mediante control PWM. Presenta una alta eficiencia, baja generación de calor y protección contra sobre corrientes y polaridad inversa (MEGAELECTRONICA, 2022).

En la ilustración 16-3 se presenta el controlador de velocidad PWM de la empresa INAFFLASH.



Ilustración 16-3: Controlador de
velocidad PWM

Realizado por: Aguinaldo Christian, 2023

Las características técnicas del control de velocidad PWM DC se detallan en la tabla 4-3.

Tabla 4-3: Características del controlador de velocidad PWM

Características	Descripción
Voltaje	12 – 40V DC
Corriente estática	0.02 A
Corriente máxima	10A
PWM ciclo de trabajo	10% - 100%
Frecuencia	13 KHz
Peso	78g
Dimensiones	60 mm x 56 mm x 27 mm

Realizado por: Aguagallo Christian, 2023

3.5.2.3. Motor Ampflow G43-500-M.

Para establecer la potencia del motor para cumplir con los requerimientos de velocidad y carga, se debe considerar el peso total del prototipo, así como el voltaje de alimentación. Debido a que el requerimiento especifica que se debe transportar un peso hasta 50Kg, se hace el cálculo de la fuerza que se requiere para mover el peso del prototipo de acuerdo a la ecuación 2, con una masa de 55kg, gravedad de 9.8m/s^2 , se obtiene 539N.

Este valor se reemplaza en la ecuación 1 y se obtiene el valor del torque total de 54 Nm

Debido a que se utilizan 2 motores, el torque se divide y el torque por cada motor sería de 27 Nm.

Uno de los requerimientos es que el prototipo no debería superar los 4Km/h dentro del cultivo que en m/s son 1.11m/s. Para obtener la potencia por motor se reemplaza esa velocidad en la ecuación 3, que da 598.29W. La potencia que se requiere dentro del cultivo denominada es de 598.29 W para trabajar a una velocidad máxima de 4Km/h.

Entonces cada motor debe tener una potencia de 299.15 W y un torque de parada de al menos 27 Nm, considerando la máxima eficiencia que presente el motor de acuerdo a sus especificaciones.

El motor Ampflow G43 cuenta con las características adecuadas para trabajar en las condiciones que se requieren para el prototipo. En la ilustración 17-3 muestra al motor brushless mencionado.



Ilustración 17-3: Motor Ampflow G43-500-M

Realizado por: Aguagallo Christian, 2023

Las características técnicas del motor Ampflow se detallan en la tabla 5-3.

Tabla 5-3: Características del motor Ampflow G43

Características	Descripción
Voltaje	12-36V
Corriente sin carga	1.2 A
Corriente máxima recomendada	30 A
Potencia continua	500 W
Par de parada	15 Nm
Eficiencia	80%
Material	Metal
Velocidad	3300 rpm
Peso	2.5 kg
Dimensiones	107.9mm x 94.95mm

Realizado por: Aguagallo Christian, 2023

De acuerdo a las recomendaciones técnicas mecánicas, se debe implementar un sistema de transmisión para no forzar el motor y extender su vida útil, además, para aumentar el torque del motor, se implementa una relación de transmisión 3:1 para asegurar el adecuado funcionamiento del sistema de tracción y desplazamiento.

3.5.2.4. Controlador dual Sabertooth 2x32

Es uno de los controladores de motor dual más eficientes, versátiles y fáciles de usar, se puede operar desde radio control analógico, TTL serial o USB. Contiene un accionamiento regenerativo y frenado para que su funcionamiento sea eficiente. Además, su limitación de corriente es configurable y se puede usar en robots de combate de hasta 100Lbs en robots normales y hasta 300 Lbs en robots de hobby (DIMENSION ENGINEERING, 2019).

En la ilustración 18-3 se presenta el controlador Sabertooth 2x32.



Ilustración 18-3: Controlador Sabertooth 2x32

Realizado por: Aguagallo Christian, 2023

Las características técnicas del controlador se detallan en la tabla 6-3.

Tabla 6-3: Características del controlador dual Sabertooth 2x32

Características	Descripción
Voltaje nominal	6-33.6 V
Voltaje regular	12-24 VDC
Corriente continua por canal	32A
Corriente estática	0.05A
Pico de Corriente por canal	64A
Control	Analógico, R/C, Simplificado de serie, packetized Serie
Protección	Térmica y contra sobre corrientes
Peso	240g
Dimensiones	70 mm x 90 mm x 25 mm

Realizado por: Aguagallo Christian, 2023

3.5.3. Dispositivos de Alimentación

3.5.3.1. Regulador de voltaje XL4016

El regulador Step-Down XL4016 tiene como función entregar un voltaje de salida constante inferior al voltaje de entrada frente a variaciones del voltaje. Soporta corrientes de salida de hasta 8A, con un voltaje de entrada entre los 4 a 36V y un voltaje de salida entre 1.25V a 32V. El voltaje de salida es ajustable mediante el potenciómetro multivuelta que tiene integrado este regulador (TECMIKRO, 2021).



Ilustración 19-3: Regulador de voltaje XL4016

Realizado por: Aguagallo Christian, 2023

Las características técnicas de este regulador de voltaje se detallan en la tabla 7-3.

Tabla 7-3: Características del regulador de voltaje XL4015

Características	Descripción
Voltaje de entrada	4 - 40 V
Voltaje de salida	1.25 – 36V
Corriente máxima	8A
Potencia	200W

Eficiencia	94%
Temperatura	-45°C – 85°C
Peso	16g
Dimensiones	61mm x 40mm x 29mm

Realizado por: Aguinaldo Christian, 2023

3.5.3.2. Batería

De acuerdo a los elementos seleccionados para el robot, se dimensiona la batería respecto al consumo total que tendría el prototipo, que se muestra en la tabla 8-3.

Tabla 8-3: Dimensionamiento de la batería

Elementos	Amperaje
sensor LD06	0.2 A
Raspberry Pi 4	3 A
Bomba Seaflo 12VDC	5 A
Motor Ampflow P40	2 x 25 A
Total	58.2 A

Realizado por: Aguinaldo Christian, 2023

Entonces, con este valor de corriente, debido a que un requerimiento dice que el prototipo debe trabajar por al menos 20 minutos, si consideramos una tasa de descarga de 59Ah para una hora de trabajo, para los 20 minutos un banco de baterías de 24V y 40.6 Ah duraría aproximadamente 41.2 minutos, considerando las características de la batería y las recomendaciones de descarga máxima, se considera una eficiencia del 80%, entonces de acuerdo a la ecuación 7, se obtiene un tiempo de 33.03 minutos. La ilustración 20-3 muestra la batería de 12V 20.3Ah recargable de la marca ChilWee, que se implementa en scooters y motos eléctricas con gran eficiencia y durabilidad (CHILWEE, 2023).



Ilustración 20-3: Batería de GEL ChilWee 6-DZF-20.3

Realizado por: Aguinaldo Christian, 2023

Las características de esta batería se presentan en la tabla 9-3.

Tabla 9-3: Características de la batería ChilWee 6-DZF-20.3

Características	Descripción
Voltaje	12V
Corriente	20.3 A
Eficiencia	80%
Ciclos	Hasta 600
Descarga máxima	150 A (5s)

Realizado por: Aguinaldo Christian, 2023

3.6. Elementos adicionales

- Boquillas de cono hueco

Para obtener una aplicación eficiente del líquido en el cultivo y se pueda abarcar el follaje de la planta de mora casi en su totalidad para que el producto aplicado reaccione correctamente, el requerimiento dice que el tamaño de las gotas no debe ser mayor a 300micras, esto se logra usando boquillas de cono hueco a una presión de 2.8 bar o 40.61 PSI. Como la bomba Seaflo proporciona esa presión y se seleccionan las boquillas de cono hueco de 0.3mm (300 micras) como las que se presentan en la ilustración 21-3.



Ilustración 21-3: Boquilla de pulverización 0.3mm

Realizado por: Aguinaldo Christian, 2023

- Tuberías

Para conocer el diámetro de tubería que se requiere para el sistema de pulverización, se debe considerar las especificaciones analizadas en el capítulo II, como el caudal requerido en el sistema, la velocidad a la que la bomba envía el líquido y el caudal máximo que entrega la bomba.

Aplicando la ecuación 5 referente a la velocidad de salida del líquido de la bomba se obtiene una velocidad de 1.45m/s de salida de la bomba.

Se considera el caudal que requieren las boquillas para aplicar la fumigación a las plantas, que ronda en los 0.3 litros por minuto por cada boquilla, entonces, al aplicar la ecuación 6 para definir el diámetro de la tubería en base al área calculada y conociendo que se requiere un total de 2.4 LPM en

cada tubería que son 0.00004m, dividiendo la velocidad de la bomba para cada tubería. Entonces el diámetro de cada tubería resulta en 12.2mm \approx 1/2 pulgada aproximándola a las medidas comerciales. Entonces las tuberías que se deben conectar a la salida de la bomba Seaflo son de 1/2 de pulgada para trabajar con un flujo aproximado en el sistema de 2.4 LPM para cumplir con el requerimiento de alcance de aplicación de la fumigación al follaje de las plantas.

La tubería adecuada para transportar el líquido se considera de acuerdo a la presión que debe soportar y la salida de la bomba DC. El tubo roscable de la marca plastigama de 1/2 pulgada que se observa en la ilustración 22-3 resulta adecuado debido a su resistencia al desgaste, además de soportar hasta 420 PSI.



Ilustración 22-3: Tubería 1/2 pulgada

Realizado por: Aguagallo Christian, 2023

3.7. Diagrama eléctrico de conexiones

Una vez determinados y establecidas las características de cada dispositivo y elemento que integra el prototipo, se realiza el diagrama eléctrico de conexiones, para lo cual se identifica el voltaje que requiere cada uno, así como los pines GPIO que se utilizan en la Raspberry (Anexo B).

La ilustración 23-3 presenta el diagrama eléctrico de conexiones, en el cual desde el banco de baterías se conectan los cables rojo-positivo y azul-negativo, hacia los elementos. El primer regulador Step-Down XL4016 reduce el voltaje de 24V a 5V y 3A para alimentar a la Raspberry Pi 4, mientras que el sensor LiDAR LD06 se conecta directamente a la Raspberry al puerto USB mediante el módulo CP2102 que establece una comunicación serial entre la Raspberry y el sensor.

El controlador Sabertooth recibe un voltaje de 24V de acuerdo al requerido por los motores, y se conecta por puerto serial con la Raspberry Pi al puerto serial 5. Del controlador salen los cables de alimentación para cada motor Ampflow G43 desde los pines M1 y M2 (Anexo C).

Para alimentar la bomba Seaflo DC se regula el voltaje de 24V a 12V y 8A, debido a que consume 5A regularmente con un máximo de 10A, pero se evita que se sobrecargue. Entonces se alimenta con esta potencia al controlador PWM, que recibe la señal PWM desde la Raspberry para regular la salida de voltaje hacia la bomba.

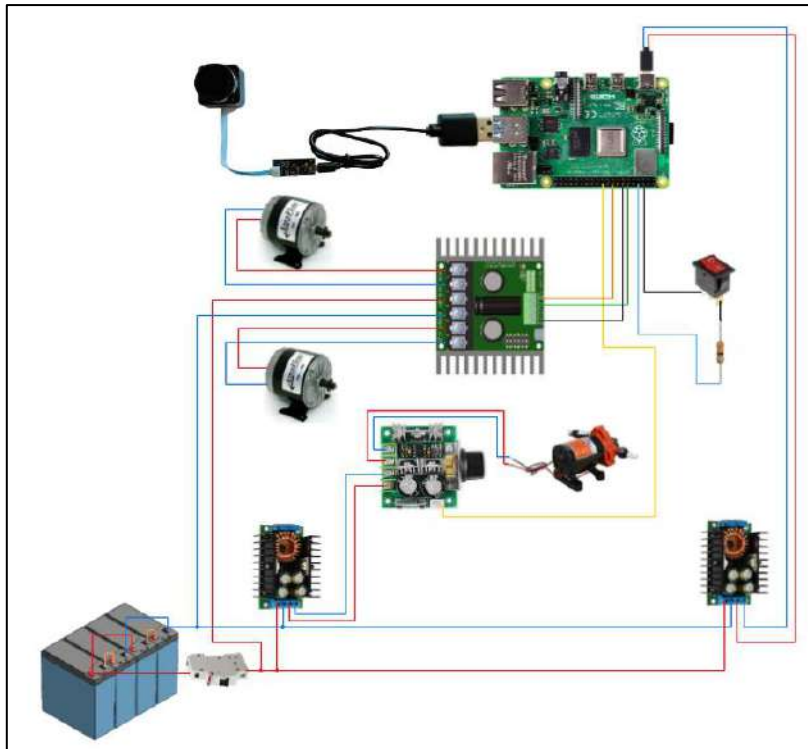


Ilustración 23-3: Diagrama eléctrico de conexiones para los dispositivos que integran el prototipo. Cable rojo positivo y cable azul negativo.

Realizado por: Aguagallo Christian, 2023

3.8. Ensamblaje de la estructura y hardware del prototipo

A partir del diseño CAD para el prototipo se procede a ensamblar la estructura metálica del prototipo para integrar los elementos hardware en la misma, con la finalidad de establecer la ubicación adecuada, las conexiones eléctricas y la conexión del sistema de pulverización.

3.8.1. Estructura metálica del prototipo.

La estructura metálica del prototipo se presenta en la ilustración 24-3, en la cual se ubicarán los elementos de manera equilibrada respecto a su peso para evitar desbalances en el prototipo y asegurar que los motores trabajen con la misma potencia.



Ilustración 24-3: Estructura base del prototipo

Realizado por: Aguagallo Christian, 2023

3.8.2. *Ensamblaje del sistema de locomoción.*

El sistema de locomoción se integra por los motores Ampflow, ubicados debajo de la base de la estructura, para trabajar mediante un sistema de transmisión por cadenas para mejorar el torque, reducir la velocidad y ampliar el tiempo de vida útil de los motores. Al eje de cada rueda se acopla un engranaje como se observa en la ilustración 25-3 y se completa el sistema de transmisión con el motor.



Ilustración 25-3: Sistema de transmisión para los motores del prototipo

Realizado por: Aguagallo Christian, 2023

La ilustración 26-3 muestra la estructura final del prototipo con los motores y el sistema de locomoción.



Ilustración 26-3: Estructura metálica del prototipo

Realizado por: Aguagallo Christian, 2023

3.9. **Ensamblaje de los elementos electrónicos**

De acuerdo al diagrama eléctrico de conexiones y una vez armada la estructura del prototipo se procede a instalar los elementos electrónicos como la Raspberry pi, los reguladores de voltaje y los controladores, tanto de los motores como de la bomba en una plancha de acrílico, como se presenta en la ilustración 27-3.

Debido a las corrientes que se manejan para los motores se establecen conexiones cableadas directas entre el banco de baterías y el controlador Sabertooth con calibres adecuados para evitar sobrecalentamientos de los conductores.

Las adecuaciones de voltajes y conexiones de los pines se detallan en el anexo D.

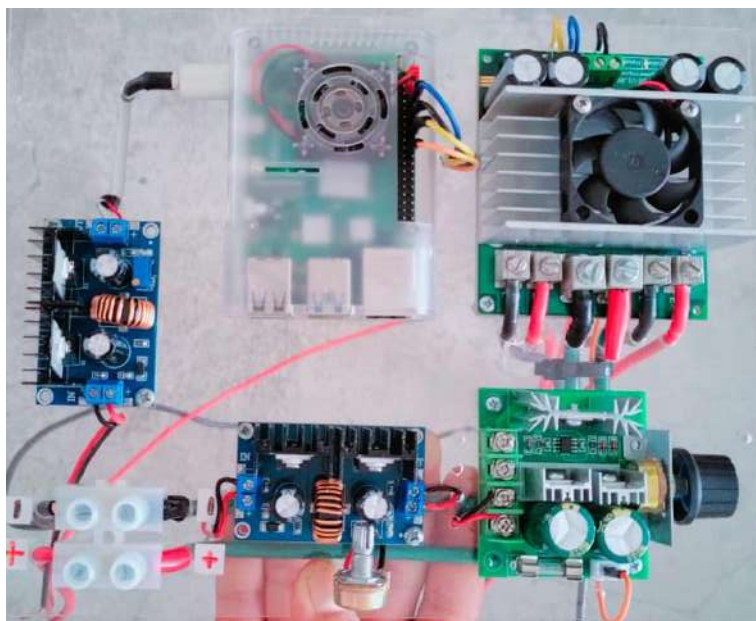


Ilustración 27-3: Placa base con dispositivos electrónicos

Realizado por: Aguinaldo Christian, 2023

3.10. Ensamblaje del sistema de pulverización

Para ensamblar el sistema de pulverización se toma en cuenta que las primeras boquillas deben abarcar la fumigación de los primeros 50cm de la planta, entonces de acuerdo a la altura de la base se establece que la tubería a cada lado del prototipo se ubica a una altura de 15 cm desde el suelo para evitar roces con el suelo, piedras o basuras presentes en el camino, desde esa medida se toman 10 cm desde el ángulo de la tubería para ubicar la primera boquilla y desde esa medida, las otras boquillas se ubican a 50cm una de otra, como se aprecia en la ilustración 28-3.



Ilustración 28-3: Ubicación de las boquillas de pulverización a 50cm

Realizado por: Aguinaldo Christian, 2023

Luego se procede a fijar las boquillas, los tapones, el filtro de la bomba, la llave de paso tipo palanca y el contenedor de 10L para almacenar el líquido a aplicar al cultivo (Anexo E). En la ilustración 29-

3 se muestra la conexión de la bomba Seaflo a las tuberías de PVC, así como la conexión al contenedor del líquido, completando así el sistema de pulverización.



Ilustración 29-3: Sistema de pulverización: conexión a la bomba y al contenedor

Realizado por: Aguagallo Christian, 2023

3.11. Configuración de la Raspberry pi 4 y ROS

Previo a realizar la programación del prototipo se debe adecuar el entorno de desarrollo con la Raspberry pi 4, ROS y Python, configurando el software, las dependencias y las librerías necesarias para cumplir con el requerimiento de funcionalidad del prototipo.

Para esto se inicia con la configuración de la Raspberry pi 4 con la instalación del sistema operativo Raspbian Buster, que se instala en la tarjeta SD con la imagen disponible en la página de Raspberry en la cual se activa la comunicación SSH, configurando la dirección IP de la Raspberry al asignarle una conexión Wi-Fi a la red del domicilio (Anexo F).

Una vez instalado el sistema operativo se coloca la SD a la tarjeta de desarrollo y se accede a la Raspberry mediante el programa PuTTY, que es un cliente de emulación de terminal y transferencia de archivos SSH para sistemas Windows. Desde esta herramienta se activa el protocolo de comunicación VNC para trabajar de manera remota desde la PC, como se muestra en la ilustración 30-3. Luego se procede a verificar la versión de Python 3.7 en el sistema, para proceder a instalar ROS Noetic que es la última versión de ROS 1 y es compatible con Raspbian Buster y esa versión de Python (Anexo G).

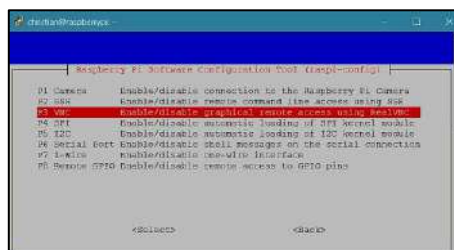


Ilustración 30-3: Activación del protocolo VNC desde la herramienta PuTTY

Realizado por: Aguagallo Christian, 2023

Posteriormente se instala Visual Studio Code para programar el código de Python y se instalan las librerías y bibliotecas adecuadas y que sean compatibles con el software configurado en la Raspberry, para luego proceder a instalar el SDK del sensor LiDAR para ROS.

3.11.1. Configuración del sensor LiDAR LD06

El sensor LiDAR LD06 genera una nube de puntos con las características de su entorno, esta nube de puntos se envía con una frecuencia de 10 veces por segundo a la Raspberry, por lo que requiere trabajar con ROS para poder almacenar estos datos en un tópic, en este caso llamado “/scan”. Para esto se debe tener ROS correctamente instalado para proceder a la integración del sensor LD06.

Se conecta el sensor a la Raspberry mediante una comunicación serial usando en módulo CP2102 que viene integrado en el paquete del sensor, como se muestra en la ilustración 31-3, y siguiendo los pasos desde el repositorio de LDrobot en GitHub se instala el SDK correspondiente a la versión del sensor en ROS (Anexo H)



Ilustración 31-3: Activación del sensor LiDAR LD06 mediante comunicación serial con la Raspberry

Realizado por: Aguinaldo Christian, 2023

Al terminar la instalación del SDK, se puede comprobar el envío y recepción de datos desde la visualización de la información del tópic “/scan” en ROS Noetic, mediante la herramienta de visualización Rviz, como se observa en la ilustración 32-3.

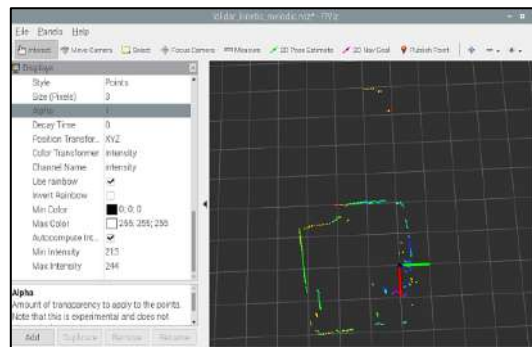


Ilustración 32-3: Visualización de la nube de puntos del sensor LD06 en Rviz

Realizado por: Aguinaldo Christian, 2023

3.12. Programación del prototipo

De acuerdo a los requerimientos y los dispositivos integrados en el prototipo, se desarrolla una programación en Python para la tarjeta de desarrollo Raspberry Pi 4, donde se integran los distintos procesos correspondientes a los módulos planteados. En la ilustración 33-3 se presenta el diagrama de flujo del programa en Python.

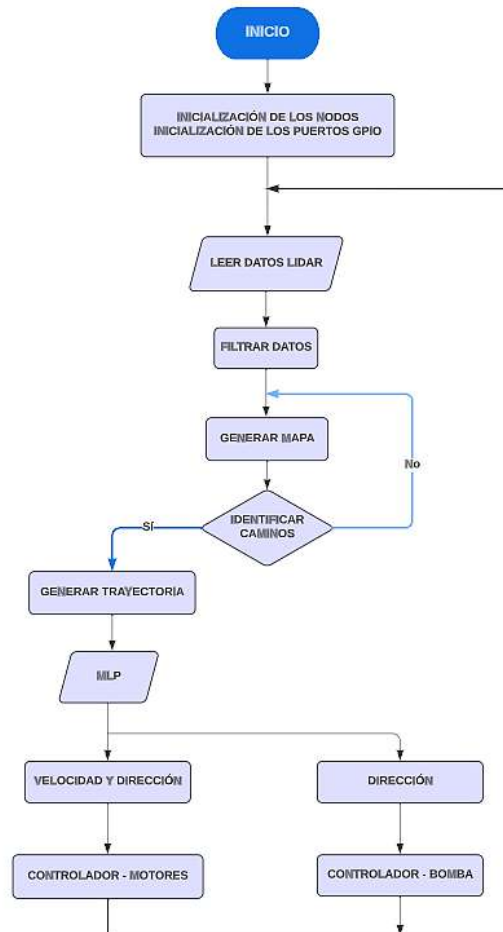


Ilustración 33-3: Diagrama de flujo del prototipo

Realizado por: Aguagallo Christian, 2023

- Importación de librerías e inicialización del nodo de ROS

Para poder trabajar con ROS Noetic en Python y Raspbian Buster, se deben importar las librerías necesarias, como se muestra en la ilustración 34-3 para el procesamiento de la nube de puntos, el filtrado de los datos, la identificación de caminos, el cálculo y generación de la trayectoria, la configuración de la MLP y el control de los motores y la bomba DC (Anexo I).

```

class RobotFumigacion:
    def __init__(self):
        rospy.set_param('/use_sim_time', False)
        rospy.init_node('robot_fumigacion')
        self.lidar_sub = rospy.Subscriber('/scan', LaserScan, self.lidar_callback)
        self.trayectoria_pub = rospy.Publisher('/trayectoria', Path, queue_size=10)
        self.lidar_data = None
        self.camino = None
        self.trayectoria = None
        self.rate = rospy.Rate(1) # Frecuencia de ejecución a 1 Hz
        # Variables de control
        self.pos_actual = (0, 0) # Inicializar la posición actual del robot
        self.dir_actual = 0.0
        self.lidar_pub = None
        self.tiempo_ultimo_calculo = rospy.Time.now()
        self.subarbottooth = SabertoothController()
        # Crear una instancia de la MLP preentrenada
        self.mlp_model = MLP()
        print("nodo iniciado")

```

Ilustración 34-3: Clase principal del prototipo
Realizado por: Aguinaldo Christian, 2023

- Adquisición y Procesamiento de Datos

Luego de importar las librerías se procede a crear la clase principal llamada “RobotFumigacion”, donde se define su método de inicialización, y se procede a realizar la configuración inicial del robot, estableciendo la comunicación entre los dispositivos y la Raspberry, y los valores iniciales de algunas variables de control y los métodos para cada nodo.

Se configuran los suscriptores y se inicia el bucle principal de ROS. Además, se crea el método de retroalimentación “lidar_callback” para que se ejecute cada vez que se reciba un mensaje en el tópico “lidar_scan” y se asignen los datos del tópico “/scan” a “lidar_data”, como se muestra en la ilustración 35-3.

```

class RobotFumigacion:
    def __init__(self):
        rospy.set_param('/use_sim_time', False)
        rospy.init_node('robot_fumigacion')
        self.lidar_sub = rospy.Subscriber('/scan', LaserScan, self.lidar_callback)
        self.trayectoria_pub = rospy.Publisher('/trayectoria', Path, queue_size=10)
        self.lidar_data = None
        self.camino = None
        self.trayectoria = None
        self.rate = rospy.Rate(1) # Frecuencia de ejecución a 1 Hz
        # Variables de control
        self.pos_actual = (0, 0) # Inicializar la posición actual del robot
        self.dir_actual = 0.0
        self.lidar_pub = None
        self.tiempo_ultimo_calculo = rospy.Time.now()
        self.subarbottooth = SabertoothController()
        # Crear una instancia de la MLP preentrenada
        self.mlp_model = MLP()
        print("nodo iniciado")

    def lidar_callback(self, data):
        self.lidar_data = data
        rospy.loginfo("received LIDAR data")

```

Ilustración 35-3: Adquisición y procesamiento de los datos LiDAR
Realizado por: Aguinaldo Christian, 2023

- Filtrado de los datos LIDAR

Para filtrar los datos y trabajar solamente con los necesarios de “lidar_data” en un radio de 3 metros, se implementa el método “filtrar_lidar” definiendo el rango y la eliminación de datos nulos, luego se almacenan esos datos en la lista vacía, se itera cada valor de tipo “ranges” que arroja el sensor, se almacenan los valores de intensidad y el ángulo de cada punto, para luego almacenar los puntos filtrados junto a las intensidades correspondientes en un array de numpy, cómo se presenta en la ilustración 36-3.


```

def filtrar_datos(self, lista_datos):
    puntos = []
    aplicar_datos = []
    for i in range(len(lista_datos)):
        rango = lista_datos[i]
        angulo = lista_datos[i+1] + lista_datos[i+2]*np.pi/180
        distancia = lista_datos[i+3]
        # verificar si el rango es un valor válido (si es más o está dentro del rango deseado)
        if not isinstance(rango, float) or rango <= 0:
            # calcular las coordenadas x, y a partir del ángulo y rango
            x = rango * math.cos(angulo)
            y = rango * math.sin(angulo)
            # crear el punto con las coordenadas x, y, z y la intensidad
            punto = (x, y, z)
            aplicar_datos.append(punto)
        else:
            # crear el punto con las coordenadas x, y, z y la intensidad
            punto = (x, y, z)
            aplicar_datos.append(punto)
    # convertir la lista de puntos a un ndarray
    self.puntos = np.array(aplicar_datos, dtype='f')
    # crear el objeto de puntos filtrados (self.puntos_filtrados)
    return self.puntos_filtrados

```

Ilustración 36-3: Método para filtrar los datos de la nube de puntos

Realizado por: Aguinaldo Christian, 2023

- Reconocimiento del camino

En el método “identificar_caminos” se toman los datos filtrados de la nube de puntos y se identifica el camino que tenga al menos 0.6 metros de ancho para luego almacenar los datos del camino en la variable “caminos”, como se muestra en la ilustración 37-3.

```

def identificar_caminos(self):
    if self.lista_datos is not None:
        puntos_filtrados = self.filtrar_datos(self.lista_datos)
        # obtener las coordenadas x e y de los puntos
        x = puntos_filtrados[:,0]
        y = puntos_filtrados[:,1]
        # encontrar el camino más cercano al robot con al menos 0.6 m de ancho
        distancia_minima = float('inf')
        camino_seleccionado = None
        for i in range(len(y)):
            if y[i] >= 0.3 and y[i] < distancia_minima:
                j = i
                while j < len(y) and y[j] >= 0.3:
                    j += 1
                if j - i >= 0.6 / 0.3 and j - i < distancia_minima:
                    camino_seleccionado = np.array([i[j], j[j]] for j in range(i, j))
                    distancia_minima = j - i
        if camino_seleccionado is not None:
            self.camino = [camino_seleccionado]
            rospy.loginfo("CAMINO: (%s)" % camino_seleccionado)
        else:
            self.camino = []
            rospy.loginfo("No hay CAMINO (%s)" % camino_seleccionado)
        # si hay más de un camino, seleccionamos el más cercano al robot
        if len(self.camino) > 1:
            distancias_a_robot = [np.linalg.norm(camino[i] - np.array(pos_robot)) for camino in self.camino]
            indice_camino_mas_cercano = np.argmin(distancias_a_robot)
            camino_seleccionado = self.camino[indice_camino_mas_cercano]
            self.camino = [camino_seleccionado]

```

Ilustración 37-3: Método para identificar el camino adecuado para el prototipo

Realizado por: Aguinaldo Christian, 2023

- Generación de trayectorias

Para la generación de la trayectoria se toman los datos del camino definido en el método anterior, se divide por la mitad de su ancho y se unen los puntos resultantes de la división, se aplica una interpolación para evitar que existan espacios vacíos entre puntos y asegurar una trayectoria suave para el robot. Esta trayectoria se publica en el tópico “path_msg”. En la ilustración 38-3 se presenta el método “generar_trayectoria”.

```

12 def generar_trayectoria(self, caminos):
13     if len(caminos) == 0:
14         self.trayectoria = [] # si no hay caminos disponibles, la trayectoria estara vacia
15         return
16
17     camino_seleccionado = caminos[0] # si solo hay un camino, seleccionarlo directamente
18
19     # obtener las coordenadas x e y del camino seleccionado
20     x_camino = camino_seleccionado[0]
21     y_camino = camino_seleccionado[1]
22
23     # Interpolacion para obtener la trayectoria suavizada del centro del camino
24     num_puntos_interpolados = 100 # Numero deseado de puntos interpolados en la trayectoria
25     t = np.linspace(0, 1, num_puntos_interpolados)
26     x_interp_centro = np.interp(t, np.linspace(0, 1, len(x_camino)), x_camino)
27     y_interp_centro = np.interp(t, np.linspace(0, 1, len(y_camino)), y_camino)
28
29     # Agregar punto final para el giro suave de 90 grados
30     x_interp_centro = np.append(x_interp_centro, x_interp_centro[-1] + 1.0)
31     y_interp_centro = np.append(y_interp_centro, y_interp_centro[-1])
32
33     # Combinar las trayectorias suavizadas de ambos caminos para generar la trayectoria final
34     trayectoria = list(zip(x_interp_centro, y_interp_centro))
35     self.trayectoria = trayectoria
36     return self.trayectoria

```

Ilustración 38-3: Método para generar la trayectoria en el camino identificado

Realizado por: Aguagallo Christian, 2023

- Seguimiento de trayectoria y control de velocidad y dirección por Redes Neuronales

Se define una red neuronal perceptrón multicapa (MLP) de una capa de entrada con función de activación ReLU, 2 capas ocultas y una capa de salida con 2 neuronas para generar las predicciones finales como se presenta en la ilustración 39-3.

```

class MLP:
    def __init__(self):
        self.modelo = self.build_mlp_model()
        self.load_weights()

    def build_mlp_model(self):
        modelo = tf.keras.Sequential()
        modelo.add(tf.keras.layers.Dense(32, activation='relu', input_shape=(2,)))
        modelo.add(tf.keras.layers.Dense(64, activation='relu'))
        modelo.add(tf.keras.layers.Dense(2))
        return modelo

```

Ilustración 39-3: Clase que define la red neuronal perceptrón multicapa

Realizado por: Aguagallo Christian, 2023

Esta red neuronal recibe los datos de la trayectoria generada y predice los valores de velocidad y dirección de acuerdo al pre-entrenamiento que recibe (Anexo J).

- Control de los motores por el Sabertooth

Luego se establecen los valores de velocidad y dirección para el controlador Sabertooth, se definen los pines GPIO de la Raspberry para la comunicación con el controlador y se crea el código para adecuar los valores al rango que maneja el Sabertooth (Anexo K).

- Control PWM para la bomba DC

Para generar la señal PWM para el control de la bomba, se crea la clase “PumpController”. Se inicializa el pin GPIO 25 para el envío de la señal desde la Raspberry hacia el controlador PWM, se configura la frecuencia a 1KHz. El método se utiliza para encender la bomba, calculando el ciclo de trabajo necesario para lograr el voltaje deseado de máximo 12V adecuados para la bomba. Mientras que el método apagar tiene la función de establecer un ciclo en 0 para el PWM (Anexo L).

- Inicializador del programa principal del prototipo.

Finalmente se crea una instancia de la clase “RobotFumigacion” llamada robot, y luego se llama al método “ejecutar” para iniciar la ejecución del programa principal del robot de fumigación.

- Obtención de datos y entrenamiento de la MLP.

Para usar la MLP en el seguimiento de la trayectoria y control de velocidad y dirección de los motores se debe entrenar la red neuronal antes de implementarla en el código principal, por lo que se realiza una toma de muestras con trayectorias similares a las que se obtendrán durante el funcionamiento del prototipo en un cultivo de mora de Castilla, considerando las dimensiones de los caminos, las dimensiones del prototipo, y la generación de la trayectoria por la mitad del camino disponible. Por lo tanto, se establece un código similar al generado para el nodo “RobotFumigacion”, desde la obtención de los datos hasta la generación de la trayectoria y se añade el cálculo de los valores de velocidad y dirección, con la finalidad de almacenar cada trayectoria con sus valores correspondientes de velocidad y dirección, para luego, al tener una cantidad adecuada para el entrenamiento de la red neuronal perceptrón multicapa, dividir los datos obtenidos en 3 grupos, un porcentaje para entrenamiento (80%), un porcentaje para validación (10%) y el resto para pruebas (10%) así se comprueba que la MLP genere valores adecuados de velocidad y dirección para cada trayectoria que será enviada y de esta manera se tenga un control adecuado de los motores.

En la ilustración 40-3 se puede observar una muestra de los datos de entrenamiento de la MLP, donde se presenta la numeración correspondiente a cada muestra, las 2 primeras columnas tienen los valores “x” y “y” de la trayectoria mientras que las 2 últimas tienen los valores de la velocidad y la dirección asignadas para llegar a ese punto desde la posición actual del robot.

9419	0.2323914911,	0.0034353662,	50.00,	50.00
9420	0.2895184731,	0.0058756655,	50.00,	50.00
9421	0.0076354525,	0.1414944580,	30.00,	25.00
9422	0.1107666409,	0.2694387719,	30.00,	31.00
9423	0.0001930331,	0.0133244553,	50.00,	50.00
9424	0.1042586592,	0.0759381149,	30.00,	26.00
9425	0.2787501402,	1.1545115606,	30.00,	28.00
9426	0.0161683321,	0.0297633324,	50.00,	50.00
9427	0.2135534104,	0.0092602043,	50.00,	50.00
9428	0.0131321235,	0.4013493150,	30.00,	25.00
9429	0.4400865847,	1.0325155236,	30.00,	31.00
9430	0.1439639742,	0.2838873972,	50.00,	32.00
9431	0.6890086943,	1.3577072141,	30.00,	32.00
9432	0.0360161017,	0.0293498656,	50.00,	50.00
9433	0.2081149630,	0.2730535517,	30.00,	35.00
9434	0.2085180582,	0.1756804267,	30.00,	38.00
9435	0.1259847898,	0.0950904563,	30.00,	39.00
9436	0.0069194137,	0.0222691813,	50.00,	50.00
9437	0.1153107942,	0.3220161900,	30.00,	30.00
9438	0.1877546969,	0.0184803925,	50.00,	50.00
9439	0.4292094599,	1.1586910389,	30.00,	30.00
9440	0.1535281546,	0.5758376501,	30.00,	29.00

Ilustración 40-3: Datos de entrenamiento de la MLP

Realizado por: Aguinaldo Christian, 2023

CAPÍTULO IV

4. VALIDACIÓN DE PROTOTIPO

En este capítulo se analizan los resultados obtenidos mediante las pruebas realizadas al prototipo para garantizar el cumplimiento de los requerimientos del capítulo III. Se realizan pruebas de alcance del sensor LiDAR, pruebas de rendimiento del banco de baterías, alcance de la fumigación a las plantas, rendimiento del líquido para fumigación y la autonomía del prototipo.

4.1. Prueba de estabilidad del sensor LiDAR LD06

Para comprobar si el sensor es estable respecto a las mediciones y obtención de datos, se realiza una prueba de estabilidad en un entorno cerrado con condiciones normales de temperatura e iluminación, para lo cual se toman 30 muestras de mediciones del sensor hacia un mismo objeto considerando el rango de 0.10m a 3m, con una iluminancia de promedio de 654 lux, medidos con la app “Lux Light Meter” para comprobar la veracidad de la iluminancia, que es el rango en que va a trabajar en el cultivo como se muestra en la ilustración 1-4.

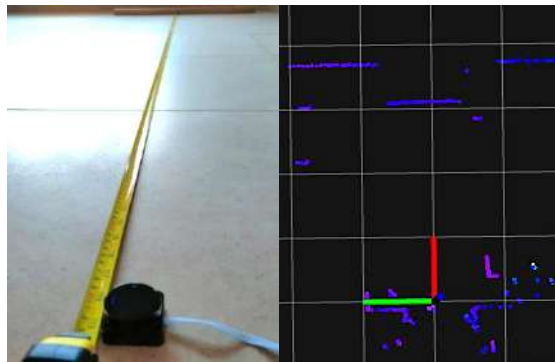


Ilustración 1-4: Medición a 3m del objetivo:
flexómetro y sensor

Realizado por: Aguagallo Christian, 2023

Para validar los resultados de las pruebas realizadas se usan herramientas como el error absoluto medio (MAE) y el error cuadrático medio (MSE), con la finalidad de obtener una precisión general del sistema y la magnitud promedio de los errores.

Para calcular el error absoluto se usa la ecuación 8, que se determina como el promedio de las diferencias absolutas entre los errores de las medidas obtenidas.

$$MAE = \frac{\sum_{i=1}^n |x_i - y_i|}{n} \quad [8]$$

Donde:

n = Número de muestras

x_i = valor de referencia

y_i = valor medido

El error cuadrático medio de la ecuación 9, mide la diferencia cuadrática media entre los valores estimados y los valores reales de una distribución de datos, este valor es siempre positivo y disminuye a medida que disminuye el error.

$$MSE = \frac{\sum_{i=1}^n (x_i - y_i)^2}{n} \quad [9]$$

Donde:

n = Número de muestras

x_i = valor de referencia

y_i = valor medido

Además, se aplica el coeficiente de variación (CV) que se expresa como un porcentaje y es útil para evaluar la proporción de homogeneidad o heterogeneidad relativa de los datos, la ecuación 10 presenta coeficiente de variación.

$$CV = \frac{\sigma}{\bar{x}} * 100\% \quad [10]$$

Donde:

σ = Desviación estándar

\bar{x} = Media (promedio) de los datos

Al realizar las mediciones se obtuvieron los datos representados en la tabla 1-4.

Tabla 1-4: Análisis comparativo de la medición del sensor lidar LD06

Muestras	Distancia de referencia (m)	Distancia medida (m)	Diferencia Absoluta (m)
1	0.100	0.118	0.018
2	0.200	0.214	0.014
3	0.300	0.313	0.013
4	0.400	0.416	0.016
5	0.500	0.515	0.015
6	0.600	0.612	0.012
7	0.700	0.713	0.013
8	0.800	0.811	0.011
9	0.900	0.918	0.018
10	1.000	1.017	0.017
11	1.100	1.116	0.016
12	1.200	1.212	0.012
13	1.300	1.313	0.013
14	1.400	1.412	0.012
15	1.500	1.514	0.014
16	1.600	1.613	0.013
17	1.700	1.712	0.012
18	1.800	1.816	0.016
19	1.900	1.913	0.013
20	2.000	2.012	0.012

21	2.100	2.117	0.017
22	2.200	2.214	0.014
23	2.300	2.315	0.015
24	2.400	2.416	0.016
25	2.500	2.516	0.016
26	2.600	2.617	0.017
27	2.700	2.719	0.019
28	2.800	2.815	0.015
29	2.900	2.917	0.017
30	3.000	3.015	0.015
Promedio:			0.015

Realizado por: Aguagallo Christian, 2023

El diagrama de barras representativo de los datos se presenta en la ilustración 2-4.

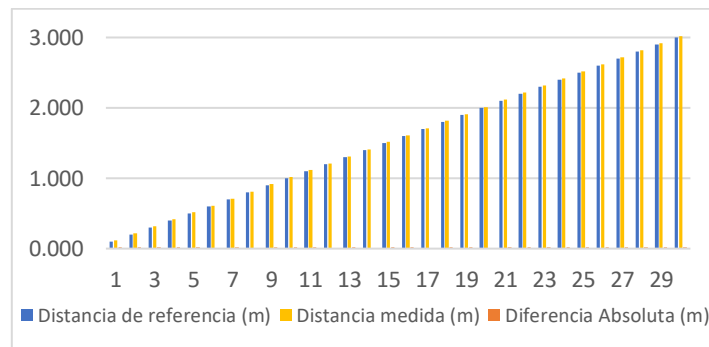


Ilustración 2-4: Prueba - Validación del sensor

Realizado por: Aguagallo Christian, 2023

Entonces, de acuerdo al valor del error absoluto se comprueba que el sensor está dentro del rango de precisión de 3cm de alcance definido por el fabricante, con un error absoluto de 1.5cm en un entorno controlado y un error cuadrático medio de 0.00021 que significa que los errores son pequeños y no presentan variabilidad constante con un coeficiente de variación del 14.748% lo que representa una desviación pequeña respecto a la media (Hernández, 2020).

4.2. Prueba de alcance del sensor LiDAR LD06

El desarrollo de esta prueba tuvo como objetivo verificar si el sensor LD06 puede tomar datos con un error mínimo en un entorno real de funcionamiento en un rango de 3m de radio, cabe destacar como se mencionó en el capítulo III que el sensor genera una nube de puntos con información de la distancia horizontal en metros desde el sensor al punto en grados e intensidad del láser, para lo cual se recabaron 30 muestras de acuerdo a la exposición a la luz en distintas horas de un día templado, en un entorno de dimensiones constantes y se comparó con un equipo de medición patrón, como se muestra en la ilustración 3-4. Además, se usó la app “Lux Light Meter”, la cual muestra un valor mínimo, un valor máximo y un valor promedio de la iluminancia presente en cada medición realizada.

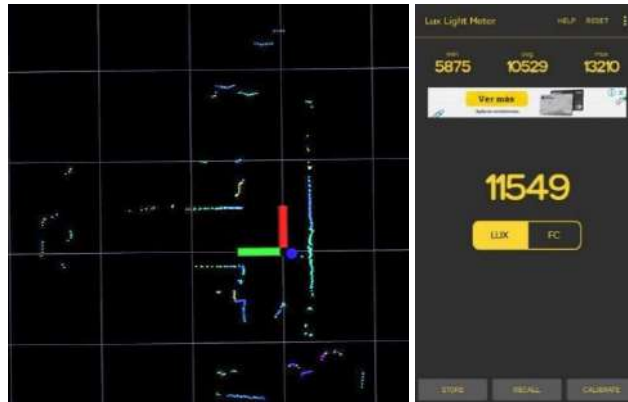


Ilustración 3-4: Prueba de alcance del sensor LiDAR LD06

Realizado por: Aguinaldo Christian, 2023

Los valores de distancia medidos por el sensor y el equipo patrón se comparan en la tabla 2-4.

Tabla 2-4: Análisis comparativo del alcance del sensor lidar LD06

Muestras	Distancia de referencia (m)	Distancia medida (m)	Iluminancia (lux)	Hora	Diferencia Absoluta (m)
1	0.500	0.511	2664.00	7:30:00	0.011
2	1.000	1.013			0.013
3	1.500	1.512			0.012
4	2.000	2.011			0.011
5	2.500	2.512			0.012
6	3.000	3.014			0.014
7	0.500	0.512	8590.00	10:30:00	0.012
8	1.000	1.013			0.013
9	1.500	1.514			0.014
10	2.000	2.012			0.012
11	2.500	2.515			0.015
12	3.000	3.015			0.015
13	0.500	0.516	17450.00	13:30:00	0.016
14	1.000	1.014			0.014
15	1.500	1.517			0.017
16	2.000	2.014			0.014
17	2.500	2.515			0.015
18	3.000	3.016			0.016
19	0.500	0.518	12795.00	15:30:00	0.018
20	1.000	1.015			0.015
21	1.500	1.516			0.016
22	2.000	2.017			0.017
23	2.500	2.516			0.016
24	3.000	3.014			0.014
25	0.500	0.515	10529.00	17:30:00	0.015
26	1.000	1.016			0.016
27	1.500	1.515			0.015
28	2.000	2.014			0.014
29	2.500	2.516			0.016
30	3.000	3.017			0.017
Promedio:					0.015

Realizado por: Aguinaldo Christian, 2023

El diagrama de barras representativo de los datos se presenta en la ilustración 4-4.

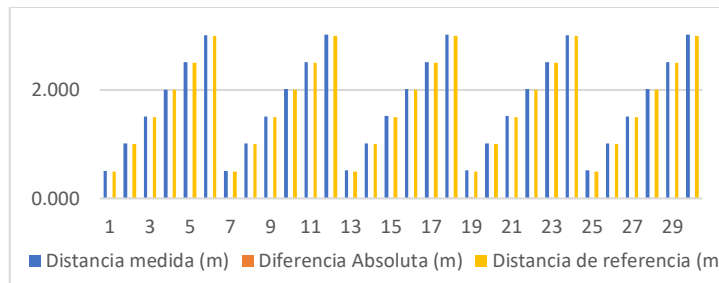


Ilustración 4-4: Prueba de medición del sensor LiDAR LD06

Realizado por: Aguinaldo Christian, 2023

Como se puede observar, el error o diferencia de valores medidos es pequeño respecto a las medidas tomadas manualmente, con un error absoluto de 0.015m o 1.5cm, dado que el fabricante estima un error de hasta 3cm el sensor estaría funcionando adecuadamente, pues las variaciones más grandes se presentaron en horarios con mayor exposición solar o lumínica y considerando que los valores de iluminancia se encuentran dentro de los 30000 lux que determina el fabricante como máxima resistencia a intensidad de luz, además se tiene un coeficiente de variación de 12.902% que es un valor aceptable respecto a la media de los datos analizados .

4.3. Prueba de tiempo de arranque del prototipo

Para comprobar el tiempo que el equipo tarda en iniciar y poder ejecutar sus tareas y su programa en la Raspberry se realiza una prueba de estabilidad, al encender el prototipo, iniciar el nodo “RobotFumigacion” y empezar a recibir los datos del sensor LiDAR desde el tópico “/scan” para identificar el camino y generar la trayectoria adecuada. Para esto se toman 10 muestras considerando el tiempo en que tarda el prototipo en reconocer el camino y empezar a moverse en la dirección correcta. La ilustración 5-4 presenta una muestra tomada con los tiempos correspondientes al inicio del software y a la detección del camino para generar la trayectoria adecuada para el prototipo.

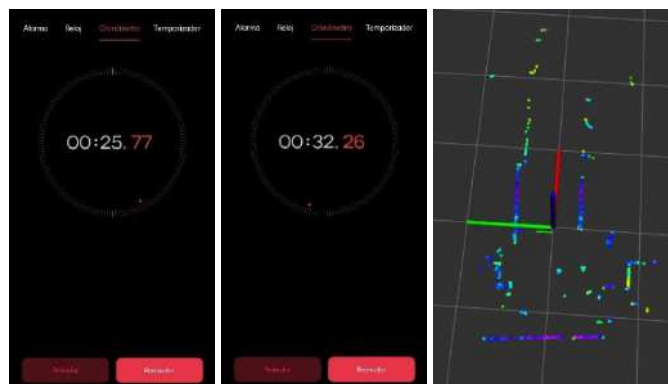


Ilustración 5-4: Tiempo de inicio del software y detección del camino

Realizado por: Aguinaldo Christian, 2023

En la tabla 3-4 se presenta el análisis comparativo de los tiempos empleados en cada muestra.

Tabla 3-4: Análisis comparativo de la estabilidad del prototipo

Muestras	Tiempo de inicio (s)	Detección del camino (s)	Tiempo empleado (s)
1	25.770	32.260	6.490
2	24.860	30.120	5.260
3	24.010	30.760	6.750
4	25.450	31.420	5.970
5	23.120	31.130	8.010
6	24.620	30.820	6.200
7	23.750	30.230	6.480
8	23.020	32.010	8.990
9	25.110	31.870	6.760
10	24.680	31.260	6.580
Promedio	24.439	31.188	6.749
Desviación Estándar	0.938	0.725	1.049
C. Variación	3.839%	2.324%	15.545%

Realizado por: Aguinaldo Christian, 2023

El diagrama de barras representativo de los datos se presenta en la ilustración 6-4.

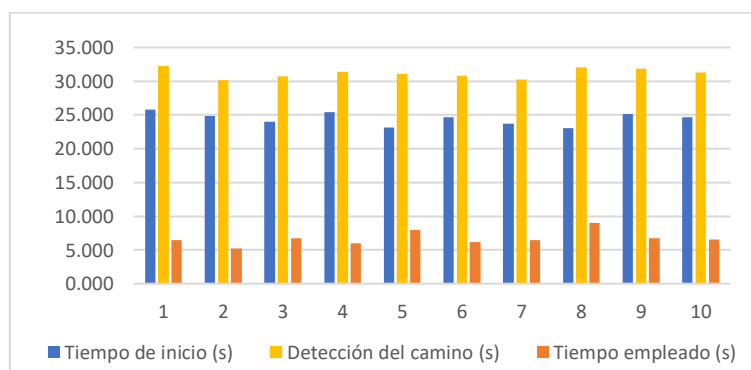


Ilustración 6-4: Prueba de medición del sensor LiDAR LD06

Realizado por: Aguinaldo Christian, 2023

El coeficiente de variación es menor al 30%, lo que indica una homogeneidad en los datos. Entonces, se obtiene un tiempo promedio de inicio del software del prototipo de 24.439 segundos, debido a que deben inicializarse los nodos y los tópicos de ROS además del sensor y los elementos hardware. Al prototipo le toma aproximadamente 6.75 segundos en reconocer un camino disponible de acuerdo a la ubicación que tenga en el cultivo, por lo que, para iniciar el trabajo del prototipo en el cultivo, resulta adecuado ubicarlo frente a la primera hilera para que el robot pueda ubicarse de mejor manera y empezar a ejecutar su labor de fumigación.

4.4. Prueba de reconocimiento del camino y generación de trayectoria

La generación de trayectoria se ha planteado que se realice por la mitad del camino, para lo cual se requiere medir si el software genera la trayectoria adecuadamente, para esto se han tomado 30 muestras de generación de trayectoria mediante el algoritmo generado en python y un equipo patrón

que permita medir la mitad del camino disponible en distintos espacios del cultivo. En la ilustración 7-4 se muestra el reconocimiento del camino desde el sensor LiDAR mientras que en la ilustración 8-4 se observa la trayectoria que se genera de acuerdo a ese camino disponible.

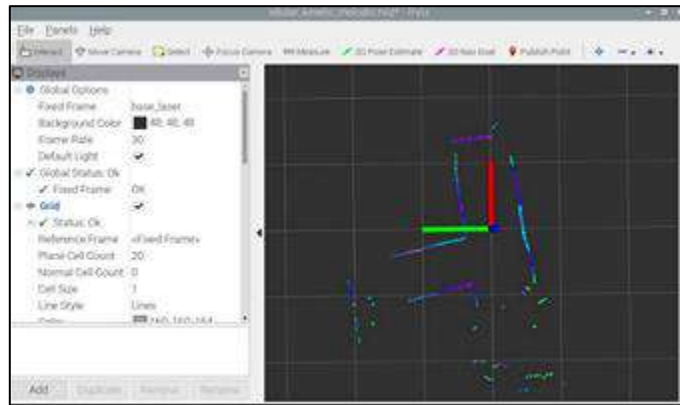


Ilustración 7-4: Reconocimiento de caminos

Realizado por: Aguagallo Christian, 2023

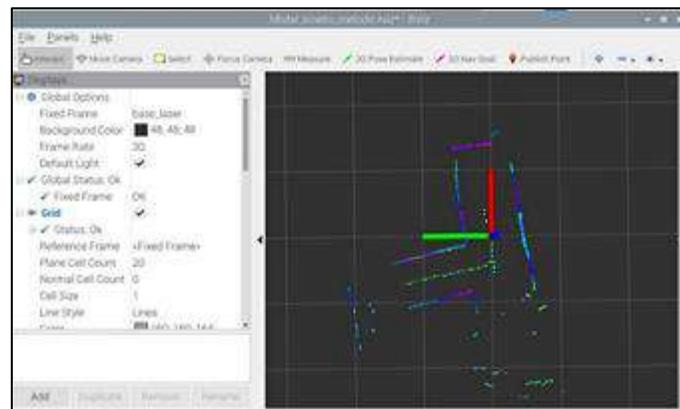


Ilustración 8-4: Generación de una trayectoria por la mitad del camino

Realizado por: Aguagallo Christian, 2023

La tabla 4-4 presenta las mediciones obtenidas durante el proceso

Tabla 4-4: Análisis comparativo de la distancia al borde de la trayectoria generada

Número de muestras	Distancia de referencia (m)	Distancia calculada (m)	Diferencia (m)
1	0.42	0.45	0.03
2	0.45	0.42	0.03
3	0.42	0.46	0.04
4	0.44	0.47	0.03
5	0.42	0.47	0.05
6	0.42	0.45	0.03
7	0.48	0.46	0.02
8	0.45	0.44	0.01
9	0.43	0.45	0.02
10	0.40	0.42	0.02

11	0.40	0.43	0.03
12	0.40	0.43	0.03
13	0.39	0.41	0.02
14	0.40	0.43	0.03
15	0.43	0.45	0.02
16	0.46	0.48	0.02
17	0.45	0.47	0.02
18	0.42	0.45	0.03
19	0.44	0.46	0.02
20	0.45	0.43	0.02
21	0.42	0.46	0.04
22	0.41	0.44	0.03
23	0.39	0.42	0.03
24	0.40	0.42	0.02
25	0.42	0.45	0.03
26	0.46	0.48	0.02
27	0.44	0.47	0.03
28	0.42	0.45	0.03
29	0.44	0.46	0.02
30	0.42	0.46	0.04
Promedio			0.03

Realizado por: Aguinaldo Christian, 2023

El diagrama de barras representativo de los datos se muestra en la ilustración 9-4.

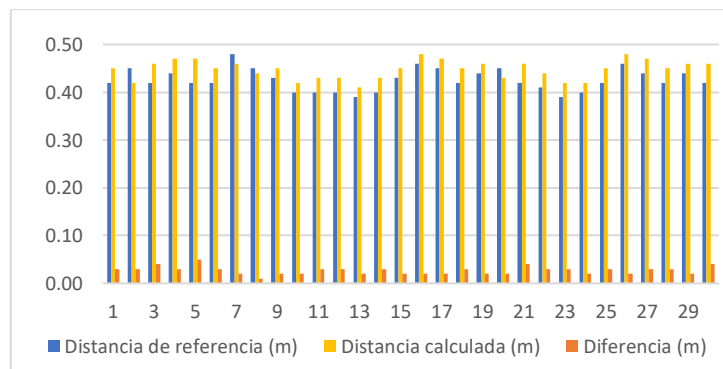


Ilustración 9-4: Prueba - Generación de Trayectoria

Realizado por: Aguinaldo Christian, 2023

De acuerdo a los datos y valores obtenidos, el error de generación de la trayectoria por la mitad del camino es de $\pm 3\text{cm}$, con una desviación estándar de 0.020 y un coeficiente de variación de 20.349% en los datos generados, lo que es aceptable dentro del camino disponible, entonces se puede determinar que el prototipo puede trasladarse por el camino disponible entre las hileras de tal manera que no choque con las plantas ni cause daños en el cultivo, pues la trayectoria se mantendría sin alejarse demasiado del centro del camino entre hileras.

4.5. Prueba de rango de pulverización

El requerimiento de pulverización es que el sistema de bombeo tenga una presión entre 30 a 45 PSI y que la salida de las boquillas llegue hasta 60cm, para esto se utiliza un equipo de medición patrón

para tomar 30 muestras en boquillas de manera aleatoria a la presión máxima de salida de la bomba Seaflo de 45 PSI. Conociendo que el error de la generación de la trayectoria tiene un error $\pm 3\text{cm}$ se considera esta variación para determinar si la distancia a la que llega el líquido desde la boquilla llegaría a la mitad de la hilera.

La ilustración 10-4 presenta la medición del rango de pulverización por parte del técnico del cultivo de mora “La Ilusión” ubicada en la provincia de Chimborazo, ciudad de Riobamba, parroquia Yaruquíes, en el barrio “Santa-Cruz”.



Ilustración 10-4: Medición del alcance de la pulverización por el experto del cultivo

Realizado por: Aguagallo Christian, 2023

La tabla 5-4 contiene los datos y las evaluaciones realizadas para determinar la validez de las medidas.

Tabla 5-4: Análisis del rango de pulverización

Número de muestras	Distancia de referencia (m)	Distancia medida (m)	Diferencia Absoluta (m)	Cuadrado de las diferencias	Cumple de acuerdo al valor de la distancia medida respecto a la distancia de referencia	Cumple de acuerdo a la validación del experto del cultivo
1	0.6	0.58	0.020	0.000400	0	1
2	0.6	0.63	0.030	0.000900	1	1
3	0.6	0.61	0.010	0.000100	1	1
4	0.6	0.67	0.070	0.004900	1	1
5	0.6	0.57	0.030	0.000900	0	0
6	0.6	0.65	0.050	0.002500	1	1
7	0.6	0.63	0.030	0.000900	1	1
8	0.6	0.66	0.060	0.003600	1	1
9	0.6	0.61	0.010	0.000100	1	1
10	0.6	0.68	0.080	0.006400	1	1
11	0.6	0.57	0.030	0.000900	0	0
12	0.6	0.59	0.010	0.000100	0	1
13	0.6	0.61	0.010	0.000100	1	1
14	0.6	0.6	0.000	0.000000	1	1
15	0.6	0.61	0.010	0.000100	1	1
16	0.6	0.64	0.040	0.001600	1	1
17	0.6	0.62	0.020	0.000400	1	1

18	0.6	0.61	0.010	0.000100	1	1
19	0.6	0.62	0.020	0.000400	1	1
20	0.6	0.59	0.010	0.000100	0	1
21	0.6	0.63	0.030	0.000900	1	1
22	0.6	0.63	0.030	0.000900	1	1
23	0.6	0.61	0.010	0.000100	1	1
24	0.6	0.64	0.040	0.001600	1	1
25	0.6	0.6	0.000	0.000000	1	1
26	0.6	0.56	0.040	0.001600	0	0
27	0.6	0.62	0.020	0.000400	1	1
28	0.6	0.65	0.050	0.002500	1	1
29	0.6	0.64	0.040	0.001600	1	1
30	0.6	0.63	0.030	0.000900	1	1
Promedio			0.028	0.001167	24	27
Porcentaje					80%	90%

Realizado por: Aguagallo Christian, 2023

El diagrama de barras representativo de los datos se presenta en la ilustración 11-4.

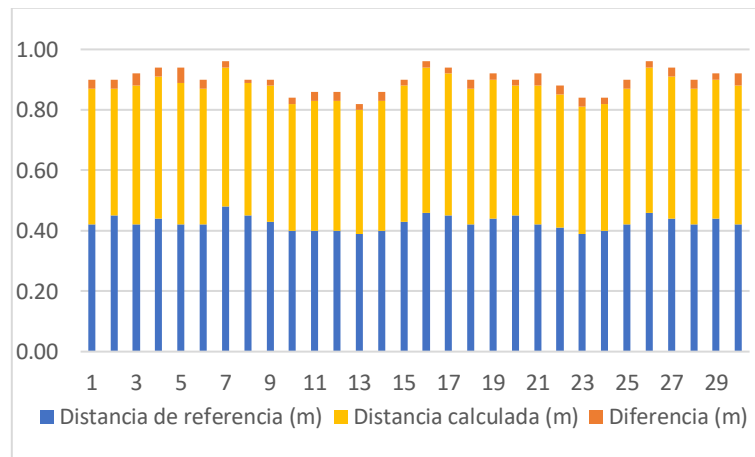


Ilustración 11-4:Prueba - Alcance del sistema de pulverización

Realizado por: Aguagallo Christian, 2023

En base a los resultados obtenidos, se comprueba que el rango de pulverización cumple el requerimiento con un 20% de variación respecto a la medida de 60cm establecida con un error de ± 2.8 cm, con un coeficiente de variación del 4.703%, que demuestra una invarianza respecto a la media del conjunto de datos. Sin embargo, la calificación del experto sugiere un porcentaje menor del error con un valor del 10% debido a que si bien, en ciertas muestras no se llega a 60 cm, el producto llegaría hasta la mitad de la hilera desde cada lado de la aplicación de la fumigación, ya que la distancia entre hileras desde la mitad de la planta es de 1.50m y el sistema de pulverización abarcaría ± 1.60 m, sumada la anchura del prototipo que es de 40cm y 120cm (60 cm de cada lado) desde las boquillas, por ejemplo, para la medida de 59cm, se tendría un rango de 158cm que es más que la distancia entre hileras, por lo tanto, el sistema de aspersión cumple con la prueba del rango de pulverización.

4.6. Prueba de optimización de la pulverización

En esta prueba se mide el rendimiento del líquido en una aplicación para 100m de fumigación dentro del cultivo, para esto se recolectan datos de mediciones realizadas cada 5 metros (sin contar las vueltas entre hileras), debido a que las hileras tienen una medida aproximada de 20m de longitud, para luego comparar esta aplicación con la fumigación manual para la misma distancia, como se presenta en la ilustración 12-4. Cabe recalcar que la prueba se realizó una mañana a temperatura regular en la sierra para evitar la presencia de vientos con velocidades mayores a 7 km/h.



Ilustración 12-4: Comparación entre fumigación manual y automatizada

Realizado por: Aguagallo Christian, 2023

La tabla 6-4 presenta un análisis de la cantidad de líquido que se emplea tanto en la aplicación manual con la bomba tipo mochila como con la aplicación por parte del prototipo RAPFA, también se incluyen los tiempos empleados para cada prueba en segundos.

Tabla 6-4: Análisis del rango de pulverización por cada distancia muestral

Número de muestras	Distancia de referencia (m)	Tiempo Manual (s)	Tiempo automatizado (s)	Diferencia de tiempo (s)	Cantidad de líquido empleado manual (L)	Cantidad de líquido empleado automatizado (L)	Diferencia de volumen (L)
1	5	17.49	4.87	12.62	0.25	0.22	0.03
2	10	35.07	9.85	25.22	0.50	0.42	0.08
3	15	52.74	14.71	38.03	0.80	0.65	0.15
4	20	70.38	19.52	50.86	1.10	0.85	0.25
5	25	87.93	24.27	63.66	1.35	1.05	0.30
6	30	105.48	28.93	76.55	1.60	1.25	0.35
7	35	123.05	33.74	89.31	1.85	1.45	0.40
8	40	140.70	38.48	102.22	2.15	1.65	0.50
9	45	158.34	43.36	114.98	2.45	1.85	0.60
10	50	175.90	48.08	127.82	2.70	2.05	0.65
11	55	193.56	52.74	140.82	3.00	2.25	0.75
12	60	211.18	57.43	153.75	3.30	2.45	0.85
13	65	228.79	62.19	166.60	3.60	2.65	0.95
14	70	246.42	66.87	179.55	3.90	2.85	1.05
15	75	263.97	71.58	192.39	4.15	3.05	1.10
16	80	281.53	76.37	205.16	4.40	3.25	1.15
17	85	299.14	81.12	218.02	4.70	3.45	1.25

18	90	316.77	85.73	231.04	5.00	3.65	1.35
19	95	334.33	90.47	243.86	5.25	3.85	1.40
20	100	351.97	95.26	256.71	5.55	4.05	1.50
Porcentaje de tiempo				72.94%	Porcentaje de volumen		27.03%

Realizado por: Aguinaldo Christian, 2023

El diagrama de barras representativo del tiempo empleado en cada muestra se presenta en la ilustración 13-4, mientras que en la ilustración 14-4 se representa el diagrama de barras de la cantidad de líquido aplicado.

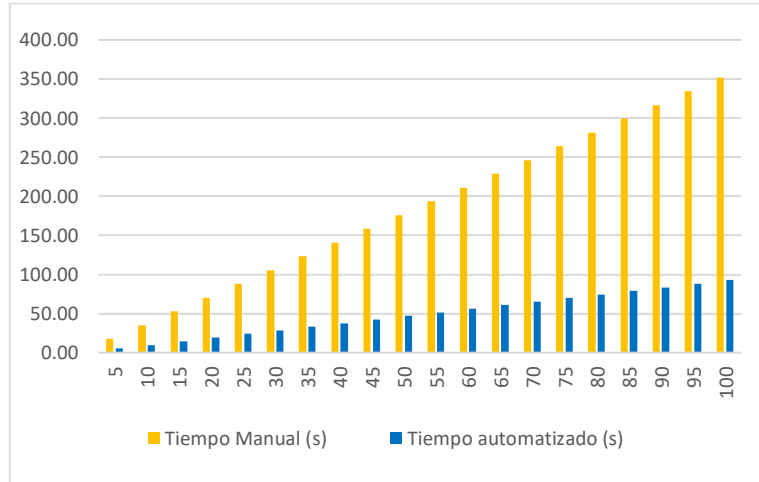


Ilustración 13-4:Prueba - Alcance del sistema de pulverización – tiempo

Realizado por: Aguinaldo Christian, 2023

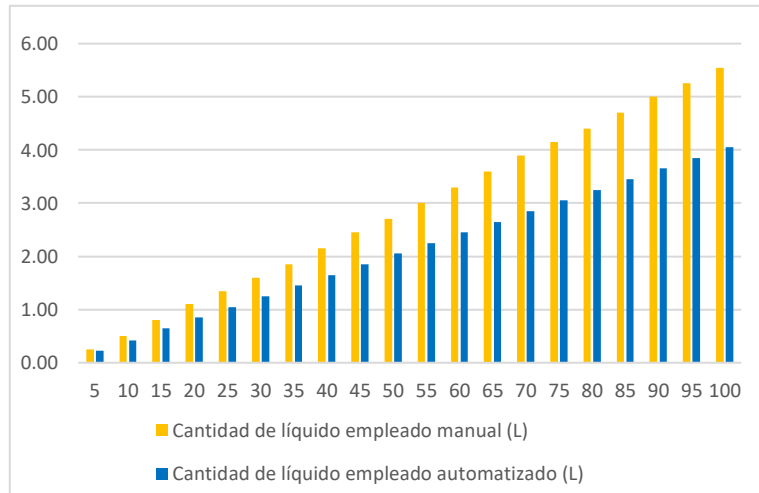


Ilustración 14-4:Prueba - Alcance del sistema de pulverización – volumen

Realizado por: Aguinaldo Christian, 2023

Como se puede observar, el tiempo que tardó el prototipo en recorrer una distancia de 100 metros a una velocidad de 4km/h es de 95.26 segundos, aproximadamente un 73% menos que el tiempo que

toma una aplicación manual a velocidad promedio. Además, se observa que se ahorra un volumen de 1.50 litros, cantidad suficiente para fumigar alrededor de 35 metros más del cultivo, esto representa según la validación del experto un “rendimiento satisfactorio”, pues estaría ahorrando 27.03% del líquido y hasta un 73% de tiempo en la aplicación de la fumigación.

4.7. Prueba de autonomía del robot

Para esta prueba se calcula el tiempo que el sistema de alimentación puede trabajar en el prototipo, con la finalidad de que se puedan aplicar los 10 litros de fumigación que lleva en el tanque de almacenamiento adecuadamente en el cultivo sin que se requiera de una carga instantánea.

Tabla 7-4: Análisis de autonomía del prototipo

Número de muestras	Cantidad de líquido (L)	Peso (kg)	Velocidad (m/s)	Tiempo de funcionamiento (min)
1.00	10.00	52.00	1.20	32.00
2.00	10.00	52.00	1.20	31.00
3.00	10.00	52.00	1.20	32.00
4.00	10.00	52.00	1.20	32.00
5.00	10.00	52.00	1.20	32.00
6.00	10.00	52.00	1.20	31.00
7.00	10.00	52.00	1.20	32.00
8.00	10.00	52.00	1.20	31.00
9.00	10.00	52.00	1.20	32.00
10.00	10.00	52.00	1.20	31.00
Promedio				31.60

Realizado por: Aguagallo Christian, 2023

El histograma respectivo se presenta en la ilustración 15-4.

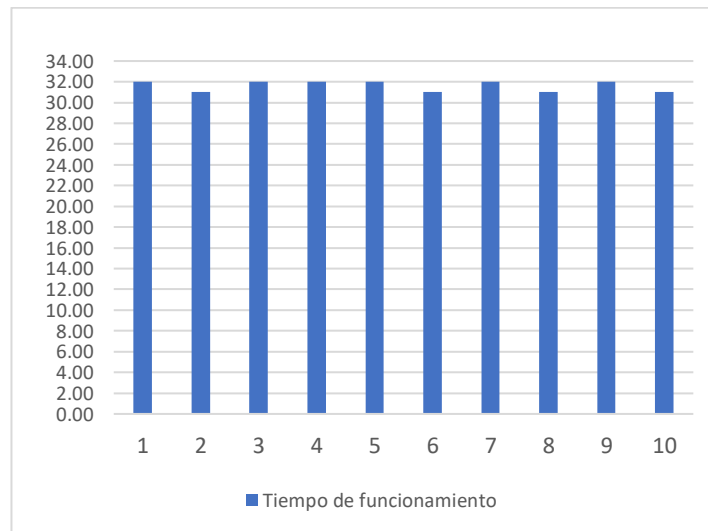


Ilustración 15-4: Prueba - Autonomía del prototipo

Realizado por: Aguagallo Christian, 2023

Se observa que el tiempo de funcionamiento del prototipo es de 31.6 minutos cargado su peso total con un volumen de 10 Litros, lo que indica que el prototipo mantendría un funcionamiento adecuado con una autonomía para la aplicación de la fumigación en un cultivo de hasta 1000 metros.

4.8. Análisis de costos del prototipo

Para esta prueba se calcula el costo del prototipo en su totalidad, para compararlo con otro de similares características. En la tabla 8-4 se enlistan los precios de cada componente y se calcula en costo total.

Tabla 8-4: Análisis de costos del prototipo

Cantidad	Materiales	Precio Unitario	Precio Total
1	Raspberry pi 4	220.00	220.00
1	Sensor LD06	110.00	110.00
2	Motores Ampflow G43	150.00	300.00
4	Batería 12V 20.3A	50.00	200.00
1	Sabertooth 2x32	150.00	150.00
2	Regulador AML4016	7.00	14.00
1	Controlador 10A	10.00	10.00
1	Bomba DC Seaflo 45 PSI	85.00	85.00
4	Llantas	15.00	60.00
1	Tanque 10 L	5.00	5.00
10	Terminales	0.40	4.00
1	Tubo PVC	7.50	7.50
8	Boquillas 0.3mm	6.00	48.00
2	Cable	2.50	5.00
1	Sistema de Transmisión	50.00	50.00
1	Estructura	200.00	200.00
100	Horas de Ingeniería	4.00	400.00
1	Otros	150.00	150.00
Total			2018.50

Realizado por: Aguinaldo Christian, 2023

El costo del prototipo RAPFA es de 2018.50 dólares, considerando los costos comerciales de todos los elementos y con el costo de importación debido a la inexistencia en Stock de los dispositivos utilizados en el prototipo. De acuerdo a las características del robot y la funcionalidad que presenta, resulta ser una herramienta accesible para poder implementarse en cultivos de mora de Castilla considerando que presentaría una inversión a largo plazo cuya rentabilidad se vería reflejada en la eficiencia del producto fitosanitario, reducción de tiempos de aplicación, reducción de carga para el agricultor y mejora en la aplicación de la fumigación.

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- De acuerdo a la bibliografía consultada, se ha establecido que los tamaños de gota recomendables para las aplicaciones de fungicidas y elementos fitosanitarios en cultivos como los de mora de Castilla deben ser de 300micras, además el sistema de locomoción debe recorrer a una velocidad de 4Km/h promedio recomendada para evitar pérdidas por deriva y asegurar que el líquido se distribuya homogéneamente, a una presión entre 42 – 45 PSI suficiente para que la pulverización tenga un alcance de 60cm desde la boquilla hasta la planta abarcando el follaje adecuado de cada lado de la hilera y con un filtro para evitar la presencia de residuos en el líquido a aplicar.
- Los dispositivos implementados en el desarrollo del prototipo se adecúan a las condiciones de trabajo en el cultivo, permiten la regulación de voltaje a valores requeridos por la Raspberry Pi 4, la bomba DC y los motores y presentan un rango de protección necesario como en el caso del sensor LD06 que tiene una protección IPx4 capaz de soportar la caída de gotas provenientes de la aplicación de la fumigación.
- El sensor LiDAR LD06 facilita el proceso de reconocimiento de caminos y hace factible la generación de trayectorias. La implementación de ROS hace que la comunicación entre nodos, clases y métodos sea más eficiente y rápida, lo que acelera el procesamiento de los datos LiDAR y las labores del robot RAPFA en el cultivo.
- El prototipo tiene la capacidad de identificar el camino disponible entre hileras de plantas de mora y generar una trayectoria por la mitad de ese camino con un error de ± 3 cm que permite su locomoción por el espacio disponible evitando colisiones con las plantas o los bordes del sembrío, cumpliendo con la aplicación del líquido que lleva en su contenedor de 10 Litros y tiene la capacidad de mover su peso total de 52Kg a la velocidad definida con una autonomía de 30 minutos.
- De acuerdo a las pruebas realizadas, el prototipo presenta un rendimiento en comparación con una aplicación manual con una bomba de mochila del 73% en tiempo de aplicación y un ahorro del 27.03% del líquido contenido para la fumigación en el cultivo, lo cual resulta en una optimización del proceso de fumigación tradicional, facilitando el trabajo del agricultor y mejorando las labores de control de plagas y enfermedades en vías a una agricultura de precisión.

5.2. RECOMENDACIONES

- El prototipo usa un sensor LiDAR LD06 que tiene un gran alcance de medición y es posible reconocer el entorno del robot en 2 dimensiones. Se podría añadir un sistema de visión artificial que potencie la identificación de caminos y obstáculos en 3 dimensiones para el prototipo e incluso que pudiera ayudar a reconocer plagas en las plantas y hojas enfermas para detectar las zonas más afectadas en el caso de requerirse una aplicación focalizada del compuesto fitosanitario.
- El error de la generación de la trayectoria si bien es pequeño y el prototipo puede seguirla sin presentar colisiones con las hileras, podría ser mejorado con técnicas más avanzadas para el control de trayectoria, que permitan reducir ese error hasta 1 cm, lo que mejoraría la capacidad de traslación del robot en este tipo de entornos.
- De acuerdo al avance de los sistemas de generación y alimentación eléctrica, se podría implementar un sistema más amigable con el ambiente y que reduzca el peso del prototipo. Siempre y cuando se cumpla con la potencia necesaria y una eficiencia que permita al robot realizar sus labores por tiempos más largos.

BIBLIOGRAFÍA

123RF. 2021. Tractor de fumigación de un campo de maíz. [En línea] 2021. [Citado el: 20 de 04 de 2023.] https://es.123rf.com/photo_58629753_tractor-de-fumigaci%C3%B3n-de-un-campo-de-ma%C3%ADz.html.

AGRICULTURERS. 2020. Pulverización agrícola. *Agriculturers, Red de especialistas en agricultura.* [En línea] 11 de 04 de 2020. [Citado el: 23 de 04 de 2023.] <https://agriculturers.com/agriculture-imm/>.

ALANIZ, Raúl Gibran Porras, ZÁRATE, Gilberto Rivera y GARCÍA, Vicente. 2022. Metaheurística para prescribir estructuras de poda en nogales: Un caso de estudio en el norte del estado de Chihuahua usando tecnologías de LiDAR: Núm. 3CP22-31. *Memorias Científicas y Tecnológicas.* 2022, Vol. 2, 1, págs. 58-59.

ALZUBAIDI, Laith, y otros. 2021. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data.* 2021, Vol. 8, págs. 1-74.

ARDUINO. 2022. ¿Qué es Arduino? *ARDUINO.cl.* [En línea] 2022. [Citado el: 18 de 04 de 2023.] <https://arduino.cl/que-es-arduino/>.

ASMOBEL. 2018. Mora de Castilla. [En línea] 2018. [Citado el: 16 de 04 de 2023.] <https://asmobel.com.co/castle.php>.

BACA, Victor Giudice y TITO, Rosario Zapata. 2022. Robotización en la Industria y la Agricultura. *Iberoamerican Business Journal.* 2022, Vol. 6, 1, págs. 28-46.

BARRALAGA, F. 2021. Sensores ultrasónicos y arduino en experiencias de laboratorio de cinemática. 2021, Vol. 9, 1, págs. 65-72.

BASOGAIN, Xabier. 2019. Redes Neuronales Artificiales y sus aplicaciones. 2019, págs. 15-30.

BEAGLEBOARD. 2023. BeagleBone Black. *beagleboard.org.* [En línea] 2023. [Citado el: 21 de 04 de 2023.] <https://beagleboard.org/black>.

BIZKAIA. 2020. Aplicación de fitosanitarios. *EQUIPOS DE APLICACIÓN DE PRODUCTOS FITOSANITARIOS.* 2020, págs. 3-9.

BORASE, Rakesh P., y otros. 2021. A review of PID control, tuning methods and applications. *International Journal of Dynamics and Control.* 2021, Vol. 9, págs. 818-827.

CABRERA A, , Fausto Ramiro y BARCIA M., Ronald Marcelo. 2018. Robótica en la agricultura. 2018, págs. 7-9.

CARDONA, Juan Pablo, LEAL, John Jairo y RAMIREZ, Jose L. 2018. Modelado Matemático de la Posición del Centro de Masa de un Robot de Tracción Diferencial. Un Enfoque desde la Mecánica Lagrangiana. 2018, Vol. 29, 6, págs. 307-320.

CEVALLOS BERMEO, Luis A. 2020. Manejo agronómico del cultivo de mora de castilla (*Rubus glaucus*). *Tesis Doctoral. UNIVERSIDAD AGRARIA DEL ECUADOR.* 2020.

CHILWEE. 2023. Batería de gel 6-DZF-20.3 VRLA. *ChiWee Battery.* [En línea] 2023. [Citado el: 27 de 05 de 2023.] <https://es.chilweebattery.com/motive-power-battery/6-dzf-20-lead-acid-battery-electric-bike.html>.

CRUZ CANDO, Edison Patricio. 2020. Aplicación de un algoritmo de resiliencia para la identificación de agentes no cooperativos y la reestructuración de formaciones en sistemas multiagente heterogéneos. 2020.

DE LA CRUZ SUSTAITA, Luis David. 2017. Sistema de navegación basado en sensor de visión dinámica para un robot hexápodo. 2017.

DEL PINO BASTIDA, Iván. 2021. Desarrollo de un sistema completo de navegación autónoma basado en GNSS y LiDAR para robots terrestres que operan en entornos dinámicos y no estructurados. 2021.

DIMENSION ENGINEERING. 2019. Sabertooth dual 60A motor driver. *Dimension Engineering - Robotics radio control power electronics.* [En línea] 2019. [Citado el: 25 de 05 de 2023.] <https://www.dimensionengineering.com/products/sabertooth2x60>.

DUSTIN, , Franklin. 2019. Jetson Nano Brings AI Computing to Everyone. *NVIDIA DEVELOPER.* [En línea] 18 de 05 de 2019. [Citado el: 16 de 04 de 2023.] <https://developer.nvidia.com/blog/jetson-nano-ai-computing/>.

ESCALONA, Uriel, RUBIO-ESPINO, Elsa y AZUELA S., Juan Humberto. 2017. Visión por computadora en un robot móvil tipo oruga. 2017, Vol. 135, págs. 145-157.

FIGUEROA SARCOS, José Antonio. 2015. Prototipo de robot semiautónomo, especializado en fumigación agrícola bajo la tecnología “Arduino”. *Tesis de Licenciatura. La Libertad; Universidad Estatal Península de Santa Elena.* 2015, págs. 3-136.

FLYABILITY. 2021. What Is SLAM (Simultaneous Localization and Mapping)? *FlyAbility.* [En línea] 2021. [Citado el: 24 de 04 de 2023.] <https://www.flyability.com/simultaneous-localization-and-mapping>.

FONSECA, G., y otros. 2020. Cadena Agroproductiva Del Cultivo De Mora En La Parroquia El Altar, Chimborazo-Ecuador/Blackberry Crop Agroproductive Chain in El Altar Parish, Chimborazo-Ecuador. 2020, págs. 576–596.

GIOVINO, Bill. 2020. Uso de las placas de desarrollo de microcontroladores como computadoras monoplaça. *Digi-Key*. [En línea] 29 de 07 de 2020. [Citado el: 26 de 04 de 2023.] <https://www.digikey.com.mx/es/articles/using-microcontroller-development-boards-as-single-board-computers>.

GOBERNACION DE ANTIOQUIA. 2013. De la fruta a la pulpa. [En línea] Territorio Verde, 2013. [Citado el: 18 de 04 de 2023.]

GUTIÉRREZ, Jaime Franco, VILLANUEVA, Moisés García y ZAVALA, Salvador Ramírez. 2022. Desempeño de los controladores clásicos y difuso proporcional para el seguimiento de pared de un robot móvil. *South Florida Journal of Development*. 5, 2022, Vol. 3, págs. 6017-6027.

GUZMÁN, Leonardo Enrique Solaque, HERRERA, Guillermo Sánchez y GUEVARA, Adriana Riveros. 2019. Control de velocidad traslacional y orientación de un robot dedicado a agricultura de precisión. 2019, Vol. 15, 28, págs. 9-19.

HERNANDEZ SANCHEZ, Daniel Eduardo. 2017. Diseño e implementación de un robot miniatura para proveer estimulación iterativa a la extremidad posterior del mutante de mielina taiep. s.l. : BUAP, 2017.

HOMER BANNISTER, Ian, PRIETO VARÓN, Mayerly y DÍAZ MARTÍNEZ, Verónica. 2020. Deriva de la aspersión con diferentes boquillas por la acción del viento en un vivero de vid. 2020, Vol. 37, 4, págs. 327-346.

HUAMANCHAHUA, D. y YALLI-VILLA, D., BELLO-MERLO, A., & MACURI-VÁSQUEZ, J. 2021. Ground Robots for Inspection and Monitoring: A State-of-the-Art. *IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. 2021, págs. 0768-0774.

HUTABARAT, D, y otros. 2019. Lidar-based obstacle avoidance for the autonomous mobile robot. *12th International Conference on Information & Communication Technology and System (ICTS)*. 2019, págs. 197-202.

HUVAL, Brody, y otros. 2015. An empirical evaluation of deep learning on highway driving. *arXiv preprint arXiv:1504.01716*. 2015.

IBERICADRON. 2022. Robot terrestre agrícola. Drones Agricultura. [En línea] 2022. [Citado el: 19 de 04 de 2023.] <https://www.ibericadron.com/tienda/agricultura/robot-agricola-terrestre/>.

INFOAGRO. 2022. Cultivo de la Mora (Parte I). *InfoAgro*. [En línea] 2022. [Citado el: 17 de 04 de 2023.] https://www.infoagro.com/documentos/el_cultivo_mora__parte_i_.asp#:~:text=Las%20plantas%20de%20mora%20se,y%201.7%20a%202.0%20m.

- INIAP.** 2014. Mora (*Rubus glaucus* Benth). *Instituto Nacional de Investigaciones Agropecuarias*. [En línea] 2014. <http://tecnologia.iniap.gob.ec/index.php/explore-2/mfruti/rmora>.
- INTAGRI.** 2020. Selección de Boquillas en la Aplicación de Agroquímicos. *Instituto para la Innovación Tecnológica en la Agricultura*. [En línea] 2020. [Citado el: 28 de 04 de 2023.] <https://www.intagri.com/articulos/fitosanidad/seleccion-de-boquillas-en-la-aplicacion-de-agroquimicos>.
- INTAGRO.** 2022. Calibración de mochilas aspersoras para la aplicación de agroquímicos. *Intagro*. [En línea] 2022. [Citado el: 26 de 05 de 2023.] <https://www.intagri.com/articulos/fitosanidad/calibracion-de-mochilas-aspersora-para-la-aplicacion-de-agroquimicos>.
- JARDINEDIA.** 2018. Fumigar | Qué es y métodos de fumigación. [En línea] 2018. <https://www.jardinedia.com/fumigar-que-es-y-metodos-de-fumigacion/>.
- JIMÉNEZ CUESTA, Jaime.** 2012. Simulación de vehículos eléctricos ligeros. 2012.
- JIMENEZ-QUINTERO, Cristian Alexander, PANTOJA-ESTRADA, Andrés y LEONEL, Hugo Ferney.** 2016. Riesgos en la salud de agricultores por uso y manejo de plaguicidas, microcuena. 12 de 12 de 2016, Vol. 18, 3, págs. 417-431.
- JSUMO.** 2022. JS200XF Infrared Long Range Sensor. [En línea] 2022. [Citado el: 29 de 04 de 2023.] <https://www.jsumo.com/js200xf-infrared-long-range-sensor>.
- KASSIR, Eduardo Estrada.** 2015. Sistemas De Control Difuso. *Eslava Zuluaga, AF*. 2015.
- LDROBOT CO.** 2021. *LiDAR Sensor LD06. Development Manual*. s.l. : SHEZHEN LDROBOT CO., LTD., 2021.
- LENTIN, J. y ALEENA, J.** 2022. Programming with ROS. *Robot Operating System (ROS) for Absolute Beginners*. 2022, págs. 173-240.
- LEÓN VÁSQUEZ, Iván & VÁSQUEZ TAPIA, Adrián.** 2013. Artículo Científico.-Diseño e implementación de un Robot Móvil Autónomo y Teleoperado para labores agrícolas. Fase 1: fumigación para plantaciones de rosas. 2013.
- LEÓN, Fabián.** 2021. ¿Como seleccionar un motor dc con reducción? *Dynamo Electronics*. [En línea] 10 de 01 de 2021. [Citado el: 02 de 05 de 2023.] <https://dynamoelectronics.com/como-seleccionar-un-motor-dc/>.
- LÓPEZ CHOLOTA, Jorge Washington, et al.** 2018. Caracterización morfo-agronómica del fenotipo de mora (*Rubus glaucus* Benth) provenientes de tres zonas agroecológicas de la provincia

Bolívar. *Tesis de Licenciatura. Guaranda. Universidad Estatal de Bolívar. Facultad de Ciencias Aropecuarias, Recursos Naturales y del Ambiente. Ingeniería Agronómica.* 2018.

LÓPEZ, D., y otros. 2016. Planificación de trayectorias con el algoritmo RRT. Aplicación a robots no holónomos. *Revista Iberoamericana de Automática e Informática Industrial.* 2016, Vol. 3, 3, págs. 56-67.

MARUYAMA, Yuya, KATO, Shinpei y AZUMI, Takuya. 2016. Exploring the performance of ROS2. *Proceedings of the 13th International Conference on Embedded Software.* 2016, págs. 1-10.

MCCOY, Margaret, MOYER, Michelle y HOHEISEL, Gwen-Alyn. 2021. Boquillas intercambiables de uso común para pulverizadoras o aspersoras usadas en cultivos perennes con copa o follaje. 2021.

MEDINA, M. A. P. L., y otros. 2011. Controladores PID y controladores difusos. *Revista de ingeniería industrial.* 2011, Vol. 5, 1.

MEGAELECTRONICA. 2022. Controlador de velocidad PWM 12V-40V 10A DC. *MEGATRONICA - Robótica y Electrónica.* [En línea] 2022. [Citado el: 22 de 05 de 2023.] <https://megatronica.cc/producto/controlador-de-velocidad-pwm-12v-40v-10a-dc/>.

MERANI, Victor Hugo, y otros. 2021. *Pulverizaciones Agrícolas: Comparación de metodologías para su evaluación agrícola sprays: Comparison of evaluation methods.* s.l. : Chilean journal of agricultural & animal sciences, 2021. págs. 325-337.

MÉRIDA LEÓN, Julio Daniel. 2019. Diseño e implementación del módulo mecánico de un dispositivo de limpieza automatizada para cigarrillos electrónicos IQOS 2.4. *Tesis Doctoral. Universidad del Valle de Guatemala.* 2019.

METROFLOR-AGRO. 2021. Importancia del cálculo y uso de dosis en coadyuvantes y agroquímicos. [En línea] 24 de 03 de 2021. [Citado el: 01 de 06 de 2023.] <https://agro.app/Lo-que-Todo-Agricultor-debe-Saber-sobre-las-Consideraciones-Meteorologicas-relativas-a-los-Pesticidas/>.

MINNALEARN. 2022. ¿Cómo funcionan los robots? [En línea] 06 de 2022. [Citado el: 01 de 05 de 2023.]

MORENO F., Álvaro y PEÑARANDA, María Isabel. 2021. Importancia del cálculo y uso de dosis en coadyuvantes y agroquímicos. *Metroflor-agro.* [En línea] 24 de 03 de 2021. [Citado el: 18 de 04 de 2023.] <https://www.metroflorcolombia.com/importancia-del-calculo-y-uso-de-dosis-en-coadyuvantes-y-agroquimicos/>.

MUNDIAL, Banco. 2019. La innovación agrícola y la tecnología son la clave para reducir la pobreza en los países en desarrollo, según un informe del Banco Mundial. *Banco Mundial.* [En línea] 2019. [Citado el: 12 de 04 de 2023.] <https://www.bancomundial.org/es/news/press->

release/2019/09/16/agricultural-innovation-technology-hold-key-to-poverty-reduction-in-developing-countries-says-world-bank-report.

NAVIA ZAMORA, José Rolando. 2019. *Optimización del proceso de fumigación agrícola mediante la utilización de los drones*. s.l. : Tesis de Licenciatura. Quevedo-UTEQ, 2019. págs. 8-12.

NAYLAMP. 2021. Sensor ultrasonido HC-SR04. *NAYLAMP MECHATRONICS*. [En línea] 2021. [Citado el: 28 de 04 de 2023.] <https://naylampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hc-sr04.html>.

NOYA, Ramiro. 2019. Pulverización: La deriva. *Plan Agropecuario*. 2019.

NVIDIA. 2022. NVIDIA Jetson . *NVIDIA*. [En línea] 2022. [Citado el: 15 de 04 de 2023.] <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/>.

OLLERO B., Aníbal. 2001. Robótica: Manipuladores y Robots Móviles. [aut. libro] Aníbal Ollero Baturone. *Robots Móviles*. Barcelona : MARCOMBO, S.A., 2001, págs. 28-36.

PALLÁS Areny, Ramón. 1993. Sensores y actuadores. *Adquisición y distribución de señales*. Barcelona : Marcombo, 1993.

PEREYRA, M. E., y otros. 2020. Planificación de caminos y trayectorias de una formación de robots autónomos. 2020, 5.

PI AUSTRALIA. 2019. Raspberry Pi 4 Model B. *Pi Australia*. [En línea] 24 de 06 de 2019. [Citado el: 20 de 04 de 2023.] <https://raspberrypi.ustralia.com.au/products/raspberry-pi-4>.

PINEDA-TORRES, Franklin. 2019. Técnicas de slam con filtros probabilísticos: caracterización y resultados en robots móviles. 2019, Vol. 9, 18, págs. 7-15.

PROMETEC. 2018. SENSORES DE MOVIMIENTO PIR. [En línea] 2018. [Citado el: 28 de 04 de 2023.] <https://www.prometec.net/sensor-pir/>.

PYTHON. 2021. Python 3.10.0. *Python.org*. [En línea] 04 de 10 de 2021. [Citado el: 31 de 04 de 2023.] <https://www.python.org/downloads/release/python-3100/>.

QOITECH. 2022. Calculador de vida útil de una batería. [En línea] Digi-Key, 2022. [Citado el: 14 de 05 de 2023.] <https://www.digikey.com/es/resources/conversion-calculators/conversion-calculator-battery-life>.

QUIMIS SUÁREZ, Joselyn Estefanía. 2021. Diseño de un prototipo experimental de medición de señales de ambientes de contaminación acústica. *Tesis Doctoral. Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas. Carrera de Ingeniería en Networking y Telecomunicaciones*. 2021.

- RAMÍREZ C., Leonel G., JIMÉNEZ A., Griselda S. y CARREÑO, Jesús M.** 2014. Sensores. *Sensores y actuadores*. México D.F. : Grupo Editorial Patria, 2014.
- RASPBERRY.** 2019. ¿Que es Raspberry Pi? *Raspberry Pi.cl*. [En línea] 2019. [Citado el: 17 de 04 de 2023.] <https://raspberrypi.cl/que-es-raspberry/>.
- RASPBERRY PI.** 2019. Raspberry Pi 4. *Raspberry Pi*. [En línea] 2019. [Citado el: 15 de 05 de 2023.] <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>.
- . 2022. Raspberry Pi OS. *Raspberry Pi*. [En línea] 03 de 05 de 2022. [Citado el: 31 de 04 de 2023.] <https://www.raspberrypi.com/software/>.
- REMOLINA-CAVIEDES, E., HIGUERA, J. y BAHENA, A.** 2020. Procedimiento para la estimación del coeficiente de fricción neumático-vía. *Revista Logos Ciencia & Tecnología*. 2020, Vol. 12, 1, págs. 71-83.
- RESUSA.** 2021. Equipos de uso agrícola. *Resusa.co*. [En línea] 2021. [Citado el: 30 de 04 de 2023.] <https://resusa.co.cr/productos/>.
- REY ARCENEGUI, Rafael.** 2019. Planficación de trayectorias para robots autónomos en entornos industriales. 2019.
- RIVAI, Muhammad, HUTABARAT, Dony y NAFIS, Zishwa Muhammad.** 2020. 2D mapping using omni-directional mobile robot equipped with LiDAR. 2020, Vol. 18, 3, págs. 1467-1474.
- RIVERA, Zandra B., DE SIMONE, Marco C. y GUIDA, Domenico.** 2019. Unmanned ground vehicle modelling in Gazebo/ROS-based environments. *Machines*. 2019, Vol. 7, 2, pág. 42.
- ROBOTICOSS.** 2021. Modelo cinemático y simulación de un robot móvil diferencial. *Roboticcoss*. [En línea] 2021. [Citado el: 12 de 04 de 2023.] <https://roboticcoss.com/modelo-cinematico-y-simulacion-con-python-robot-movil-diferencial/>.
- ROBOTS.** 2022. Pioneer 3. *Robots IEEE*. [En línea] 2022. [Citado el: 22 de 04 de 2023.] <https://robots.ieee.org/robots/pioneer/>.
- ROCKWELL AUTOMATION.** 2022. Sensores ultrasónicos. *Dispositivos de detección de presencia*. [En línea] 2022. [Citado el: 13 de 04 de 2023.] <https://www.rockwellautomation.com/es-mx/products/hardware/allen-bradley/sensors-and-switches/presence-sensors/ultrasonic-sensors.html>.
- RODAS CÓRDOVA, Andrea Estefanía.** 2021. Diseño e implementación de un sistema de generación de trayectoria para el control de un robot móvil, utilizando inteligencia artificial. *Tesis de Licenciatura*. 2021.

- RODRÍGUEZ PULIDO, Edwin A. y ZABALA MENESES, Martin E.** 2017. Desarrollo de sistema autónomo y prototipado robótico par fumigación de cultivos de tomate. 2017.
- ROLDÁN VILORIA, José.** 2014. Motores CC. *Motores de corriente continua*. Madrid : Ediciones Paraninfo, SA, 2014.
- ROMO VINUEZA, Carlos Guillermo.** 2022. Estrategias para la reducción de riesgos causados por el fungicida cymoxanil en la producción de mora (*rubus glaucus benth*) en la provincia del Carchi, Ecuador. 2022.
- ROS.ORG.** 2021. What is ROS. *ROS - Robot Operating System*. [En línea] 2021. [Citado el: 23 de 04 de 2023.] <https://www.ros.org/>.
- RUIZ-DEL-SOLAR, Javier, LONCOMILLA, Patricio y SOTO, Naiomi.** 2018. A survey on deep learning methods for robot vision. *arXiv preprint arXiv:1803.10862*. 2018.
- SALAS MEDINA, Raul Alfredo.** 2019. Generación de trayectorias para robot móvil mediante una variación del método rrt. *nstituto de Ingeniería y Tecnología*. 2019.
- SALAS, Luis Miguel Enciso.** 2015. Diseño de un sistema de navegación autónomo para robots móviles usando fusión de sensores y controladores neuro difusos. 2015.
- SÁNCHEZ, Víctor Hugo y ZAMBRANO MENDOZA, José Luis.** 2019. Adopción e impacto de las tecnologías agropecuarias generadas en el Ecuador. 2019, Vol. 30, 2, págs. 28-39.
- SANTOS, Lisi Katherine Carpio.** 2018. El uso de la tecnología en la agricultura. 2018, Vol. 2, 14, págs. 25-32.
- SEAFLO.** 2018. SEAFLO 12v diaphragm water pump | 3.0 GPM 45 PSI Marine Diaphragm Water Pump for Sale. *SEAFLO*. [En línea] 2018. [Citado el: 15 de 05 de 2023.] <https://www.seaflo.com/en-us/product/detail/604.html>.
- SMART-MOTOR DEVICES.** 2020. Controladores para motores. *Smart Motor Devices- Accionamiento y controladores para motores eléctricos*. [En línea] 2020. [Citado el: 01 de 05 de 2023.] <https://smd.ee/esp/brushcontroller.htm>.
- SOLOELECTRONICOS.** 2016. Dimensionamiento de baterias en embarcaciones electricas. *Cálculo Batería*. [En línea] 27 de 08 de 2016. [Citado el: 08 de 05 de 2023.] <https://soloelectronicos.com/tag/calculo-bateria/>.
- TECHNIDEA.** 2019. ¿Pulverización o Aplicación? [En línea] 2019. [Citado el: 12 de 04 de 2023.] https://www.produccion-animal.com.ar/produccion_y_manejo_pasturas/pasturas_combate_de_plagas_y_malezas/79-pulverizacion_o_aplicacion.pdf.

- TECMIKRO.** 2021. XL4015 MODULO REGULADOR DE VOLTAJE 30V 5A. *TECMikro Ecuador*. [En línea] 2021. [Citado el: 02 de 06 de 2023.] <https://tecmikro.com/alimentacion-pilas-baterias/515-xl4015-modulo-regulador-de-voltaje-30v-5a.html>.
- TEYMOUROUEI, Farzan, SADEGHI, Amir y KARRAY, Feras.** 2019. Efficient deep learning for 2d laser scan-based robot control. *IEEE Transactions on Robotics*. 2019, Vol. 35, 4, págs. 1164--1177.
- TIAN, Zeyu, y otros.** 2020. An improved RRT robot autonomous exploration and SLAM construction method. *2020 5th International Conference on Automation, Control and Robotics Engineering (CACRE)*. 2020, págs. 612-619.
- TIE, Siang y TAN, Chee Wei.** 2013. A review of energy sources and energy management system in electric vehicles. 2013, Vol. 20, págs. 82-102.
- TORRES TAPIA, M. A., y otros.** 2019. Caracterización y agrupamiento de cebadas imberbes mediante sensores infrarrojos y rendimiento de forraje. 2019, Vol. 10, 5, págs. 1125-1137.
- TRITEC-INTERVENTO.** 2020. Cómo calcular y diseñar un banco de baterías. *TRITEC INTERVENTO*. [En línea] 28 de 05 de 2020. [Citado el: 30 de 04 de 2023.] <https://tritec-intervento.cl/calculo-y-diseno-de-un-banco-de-baterias/>.
- UNIVERSAL ROBOTS.** 2022. Robots para Agricultura: Hacia un mayor aprovechamiento de los recursos. *Universal Robots*. [En línea] 16 de 05 de 2022. [Citado el: 19 de 04 de 2023.] <https://www.universal-robots.com/es/blog/robots-para-agricultura/>.
- VIVAS, Hevert, MARTÍNEZ, Héctor Jairo y PÉREZ, Rosana.** 2018. Método secante estructurado para el entrenamiento del perceptrón multicapa. 2018.
- XAG.** 2021. XAG® R150 Unmanned Ground Vehicle. *XAG*. [En línea] 2021. [Citado el: 19 de 04 de 2023.] https://www.xa.com/en/xauv_r150.
- YDLIDAR.** 2018. YDLIDAR X4. *YDLIDAR*. [En línea] 01 de 2018. [Citado el: 30 de 04 de 2023.] <https://www.ydlidar.com/products/view/5.html>.
- ZAYAS-GATO, Francisco, y otros.** 2020. El control PID. *Diseño de controladores PID*. Coruña : Servicio de Publicacións, 2020.
- ZULETA OSPINA, I. A.** 2018. ANTRACNOSIS (*Colletotrichum gloeosporioides*, *C. acutatum*). *MANEJO INTEGRADO DE PALGAS EN EL CULTIVO DE LA MORA *Rubus glaucus* Benth.* 2018.
- ZUMBA BRIONES, Joyce Anahí.** 2022. Manejo agro-técnico del cultivo de mora de castilla (*Rubus glaucus*), en la sierra ecuatoriana. Babahoyo : s.n., 2022. págs. 7-17.

ZURITA, R., y otros. 2019. Sistema de localización para robots móviles de bajo costo utilizando marcas de referencia artificiales en ambientes de interiores. 2019, págs. 1081-1090.

ANEXOS

乐动机器人
LD06 Development manual

2. Communication Interface

LD06 uses Z-HL5T-4P 1.5mm connectors to connect with external system to implement power supply and data receiving. Below please find the definition and parameter requirements for specific interfaces:

Number	Signal Name	Type	Description	Min	Typical	Max
1	Tx	Output	Radar data output	0V	3.3V	3.5V
2	PWM	Input	Motor control signal	0V	-	3.3V
3	GND	Power Supply	Power negative	-	0V	-
4	PSV	Power Supply	Power positive	4.5V	5V	5.5V

LD06 is equipped with a stepless speed adjusting mode motor driver which can control the start, stop and speed of the motor via the PWM signal in the interface. Due to the individual differences of each motor, the actual speed may be different when the duty rate is set as typical value. To accurately control the motor speed, it is needed to according to the speed information in the receiving data to control in closed-loop.

The LD06 takes use of standard asynchronous serial port (UART) to transmit data in one way, the transmission parameters are shown as below table:

Baud Rate	Data Bits	Stop Bits	Parity Check Bit	Flow Control
230400	8 Bits	1	No	No

3. Communication Protocol

3.1. Data Packet Format

LD06 adopts one-way communication. It begins to send measuring data packet once

4/15 sales@ldrobot.com
COPYRIGHT © 2020 SHENZHEN LDROBOT CO., LTD. ALL RIGHTS RESERVED

乐动机器人
LD06 Development manual

working stably, without any instruction. The format of the data packet is as below:

Start Character	Data Length	Radar Speed	Start Angle	Data	End Angle	Timestamp	CRC check
54H	1 Byte	LSB MSB	LSB MSB	LSB MSB	LSB MSB	1 Byte

- ◆ starting character: Length 1 Byte, fixed value 0x54, means the beginning of data packet.
- ◆ Data Length: Length 1 Byte, the last three digits reserved, the last five digits represent the number of measured points in a packet, currently fixed value 12;
- ◆ Radar speed: Length 2 Byte, in degrees per second;
- ◆ Start angle: Length 2 Byte; unit: 0.01 degree;
- ◆ Data: A measurement data length is 3 bytes, please refer to next section for detailed explanation;
- ◆ End Angle: Length 2 Byte; unit: 0.01 degree;
- ◆ Timestamp: Length 2 Bytes in ms, recount if reaching to MAX 30000;
- ◆ CRC check: Checksum of all previous data;

Please refer to the data structure as following:

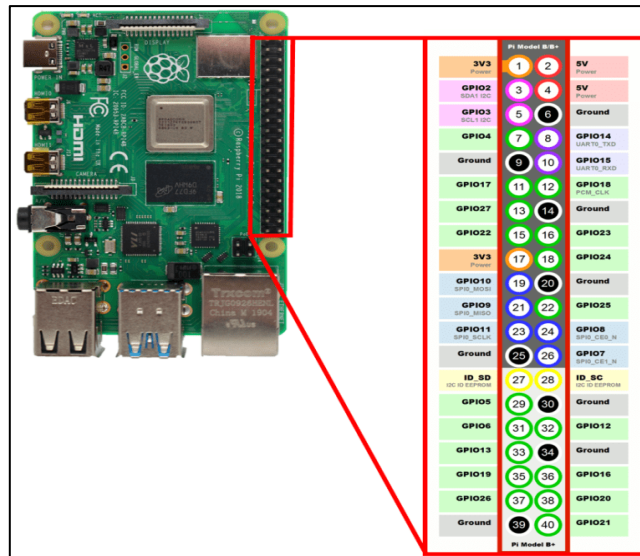
```

#define ANGLE_PER_FRAME 12
#define HEADER 0x54
typedef struct __attribute__((packed))
{
    uint8_t header;
    uint8_t wor_len;
    uint16_t speed;
    uint16_t start_angle;
    uint16_t end_angle;
    uint16_t timestamp;
    uint8_t crc8;
}LD06FrameTypeDef;
    
```

5/15 sales@ldrobot.com
COPYRIGHT © 2020 SHENZHEN LDROBOT CO., LTD. ALL RIGHTS RESERVED

ANEXO A: Pines e interface de comunicación del sensor LiDAR LD06
Para revisar la documentación completa revisar:

<https://www.ldrobot.com/editor/file/20220416/1650094739136874.pdf>



ANEXO B: Pines GPIO de la Raspberry Pi 4
Para conocer más información acerca de los pines GPIO de la Raspberry Pi 4 revisar:

<https://www.raspberrypi-spy.co.uk/2012/06/simple-guide-to-the-rpi-gpio-header-and-pins/>

Sabertooth 2x32

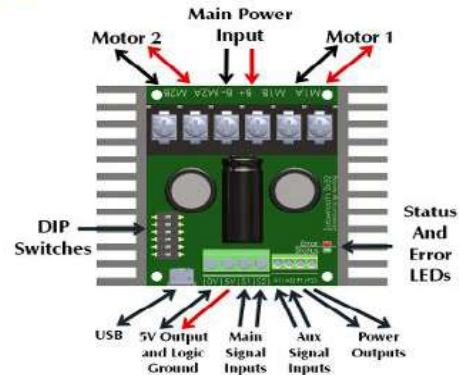
Dimension Engineering



Sabertooth 2x32 is a dual channel motor driver capable of supplying 32 amps to two motors, with peak currents up to 64 amps per motor. It can be operated from radio control, analog, TTL serial or USB inputs. It uses regenerative drive and braking for efficient operation. A variety of operating modes including tank style mixing and automatic calibration allow most projects to work immediately out of the box. In addition to the standard operating modes, Sabertooth 2x32 features additional signal inputs and power outputs, as well as enhanced configuration options. User-defined operating modes allow for custom operation, such as switching between radio control and computer-driven inputs on the fly, emergency stops or front panel control overrides. The auxiliary power outputs can be configured to allow the Sabertooth 2x32 to operate from a power supply without a parallel battery, or automatic control of electromagnetic brakes. When combined with Dimension Engineering's Kangaroo x2 motion control module, the Sabertooth 2x32 can be used for closed-loop position or speed control with encoder or analog feedback. The state of the driver can be monitored in real time using the USB port in any operating mode, making debugging your project faster and easier. Sabertooth 2x32 is more flexible, robust and powerful than previous motor drivers, while also being easier to use.

www.dimensionengineering.com/products/Sabertooth2x32

Overview



Main Power Input: Connect to a 6V-33.6V Battery or Power Supply.

Motor 1 and Motor 2: Connect Motor 1 to the M1A and M1B. Connect Motor 2 to M2A and M2B.

DIP Switches: These are used to set the operating mode and options. Can be changed while operating.

USB: A standard Micro USB port. Connect to a PC or other USB host to control, monitor or modify.

Logic Ground: The 0V logic ground is connected internally to B-

5V Output: 5V is a regulated 5 volt output. You can use it to power additional circuitry up to 1 amp.

Main Signal Inputs: Connect your main analog, R/C or serial signals here.

Aux Signal Inputs: These may be used for additional control. Optional in most modes.

Power Outputs: These connect to voltage clamp resistors, electromagnetic brakes, field windings, or other moderate power loads. 8A max current per channel.

Status and Error LEDs: These glow and flash to indicate the status of the Sabertooth 2x32.

www.dimensionengineering.com/products/Sabertooth2x32

Specifications

		<i>Mechanical specifications</i>		
Dimensions		2.75 x 3.5 x 1.0 inches (70mm x 90mm x 26mm)		
Weight		4.5 ounces (125 grams)		
		Minimum	Typical	Maximum
Wire size, battery		16 gauge	10 gauge	10 gauge
Wire size, motors		16 gauge	12 gauge	10 gauge
Wire size, signal		28 gauge	24 gauge	18 gauge
Operating temperature		0F (-20C)	70F (25C)	160F (70C) ¹
		<i>Electrical Characteristics</i>		
		Minimum	Typical	Maximum
Input voltage, B+ and B-		6.0 Volts	12 or 24 Volts	33.6 Volts
Continuous output current, M1 and M2		-	-	32 amps ¹
Peak output current, M1 and M2		-	-	64 amps ²
Output voltage, M1 and M2		-95% of input voltage (average)	-	95% of input voltage (average)
Voltage, P1 and P2		0V	-	Input voltage +3V
Output current, P1 and P2				8 amps, sink only
Output voltage, 5V		4.85	5.0	5.15
Output Current, 5V		-	-	1A
Input voltage, S1 and S2		-3V	0V to 5V	12V ³
Input voltage, A1 and A2		-3V	0V to 5V	12V ³
Output Voltage, S2 and A2		0V		3.5V

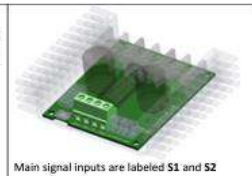
¹ Maximum continuous output current derates linearly above 40C ambient. At 70C ambient the maximum continuous output current is 10 amps per channel.

² Can be reduced by software setting

³ Stress rating only, signals over 5 volts will be read as 5 volts by the Sabertooth 2x32

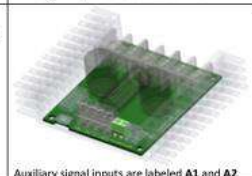
Inputs

Main Signal Inputs
The main inputs are labeled **S1** and **S2**. They can be set for analog, R/C or serial communications. When using serial, S1 is the RX port and S2 is the TX port.



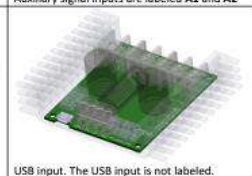
Main signal inputs are labeled **S1** and **S2**

Auxiliary Inputs
The auxiliary inputs are labeled **A1** and **A2**. They can be set for analog, R/C or serial communications. When using serial, A1 is the RX port and A2 is the TX port.



Auxiliary signal inputs are labeled **A1** and **A2**

USB Input
The Sabertooth connects via a standard micro USB plug. A windows driver installs with the DESCRIBE PC software. Linux and Mac will work out of the box with no driver. The USB port is also used to communicate with DESCRIBE to configure the Sabertooth 2x32.



USB input. The USB input is not labeled.

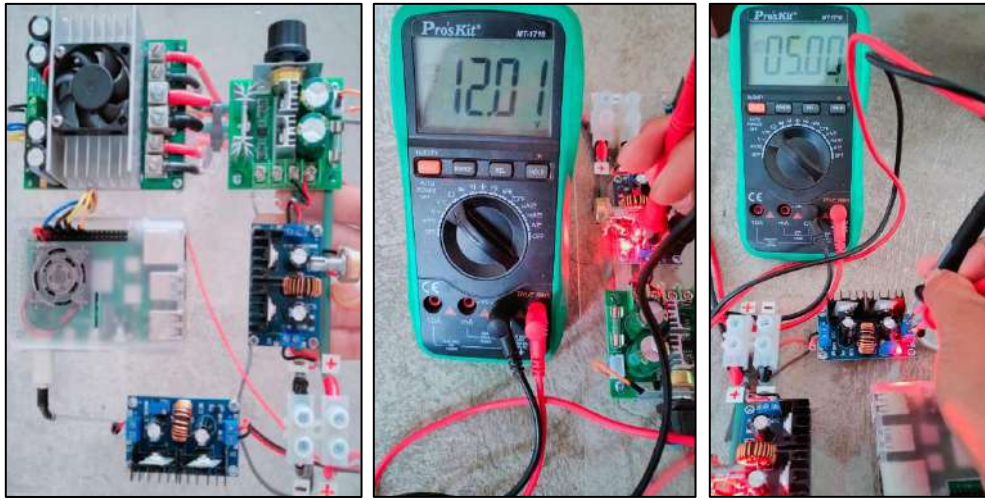
Sabertooth 2x32 has four logic inputs S1, S2, A1 and A2. The analog input range is 0 to 5 volts. Digital signals can be 3, 3.3 or 5v logic. All third generation drivers such as Sabertooth 2x32 include a USB port. The USB port can be used for control from a PC or embedded processor like a Raspberry Pi. It can also be used to monitor the inputs and outputs, set up user modes and custom settings, and update the firmware.

www.dimensionengineering.com/products/Sabertooth2x32

ANEXO C: Guía de usuario de controlador Sabertooth 2x32

Para conocer las especificaciones del controlador revisar:

<https://www.dimensionengineering.com/datasheets/Sabertooth2x32.pdf>

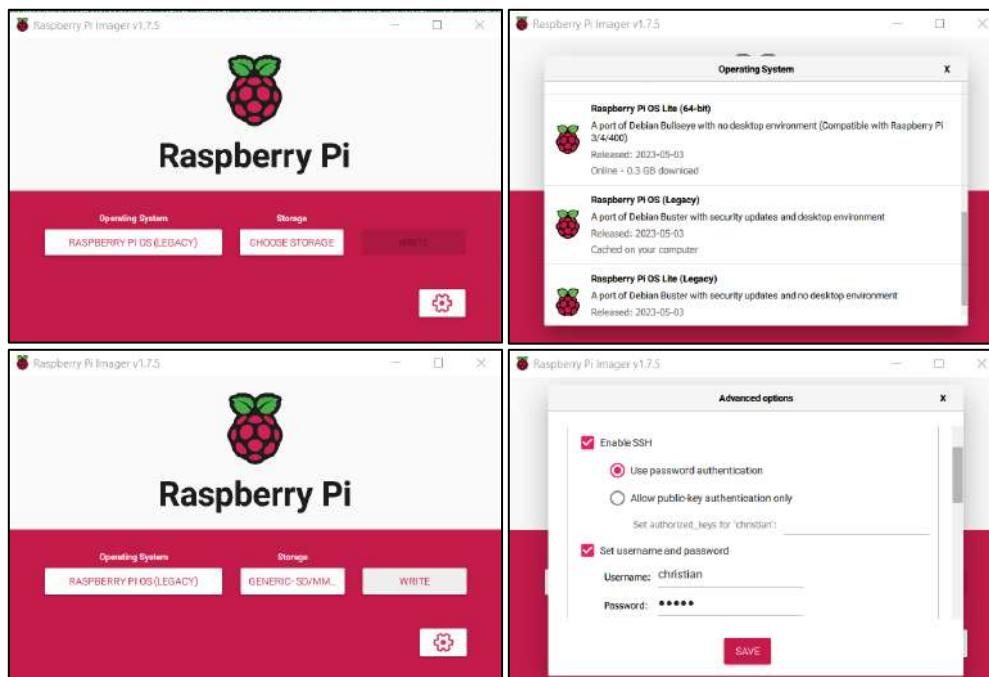


ANEXO D: Conexiones y regulación de los dispositivos electrónicos

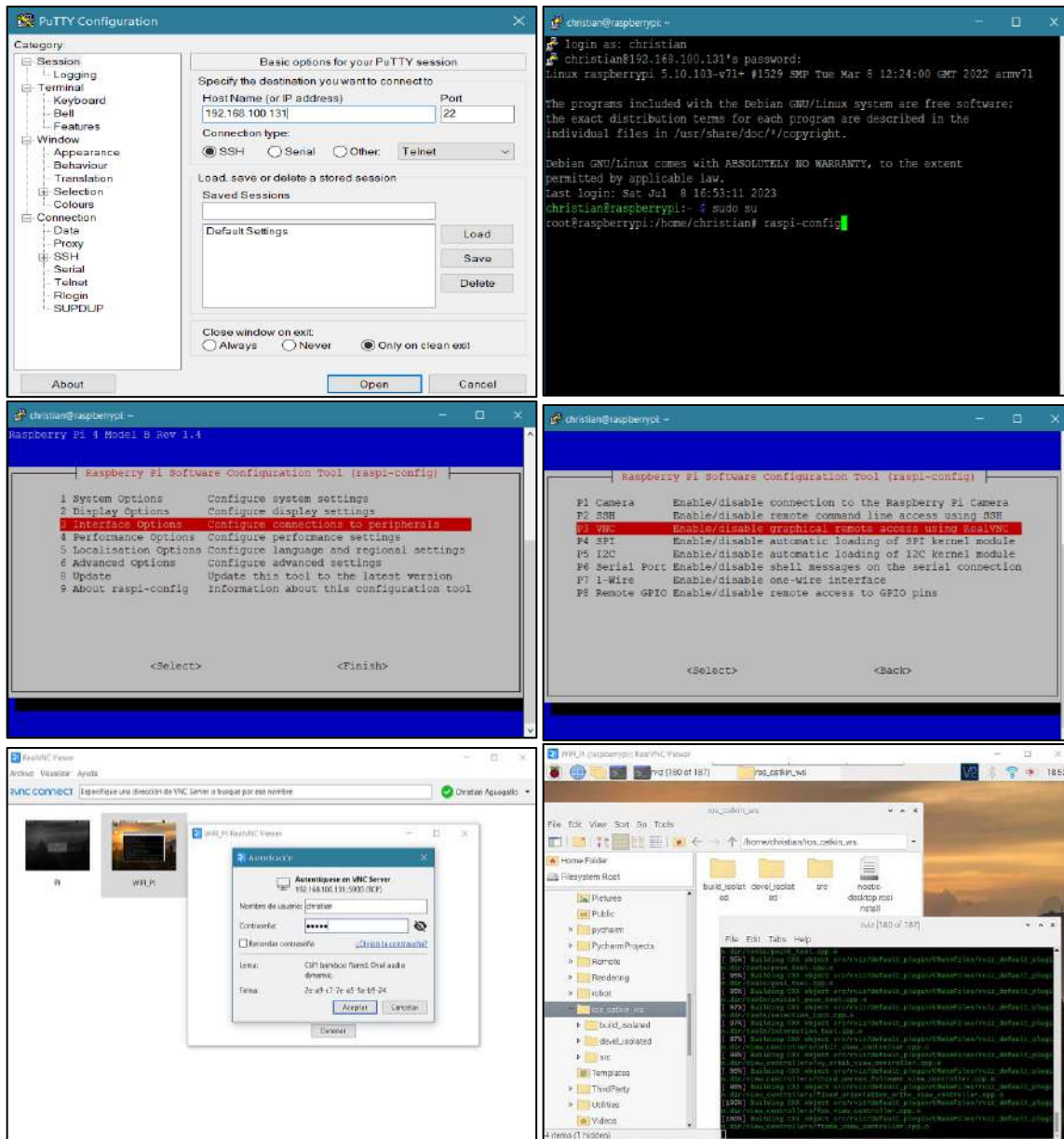




ANEXO E: Sistema de Pulverización: ensamblaje y funcionamiento



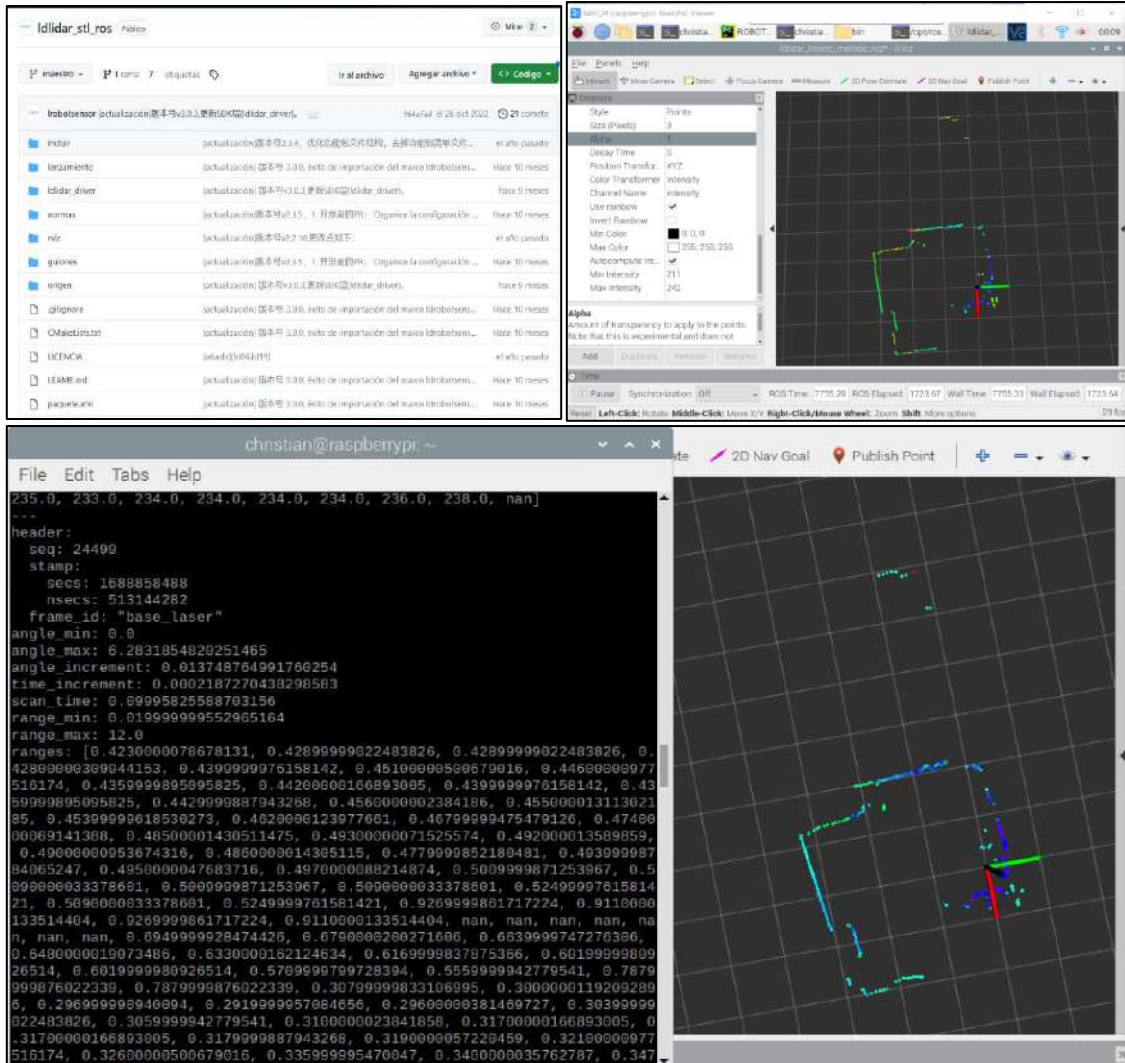
ANEXO F: Instalación de Raspbian Buster en la SD Card



ANEXO G: Activación del protocolo VNC y adecuación del entorno de desarrollo.

Para conocer los pasos concretos para la instalación revisar:

<https://www.skulltrap.co/2019/11/instalar-raspbian-en-micro-sd-raspberrypi.html>



ANEXO H: Instalación del SDK del sensor LiDAR LD06 en ROS Noetic
 Para conocer los pasos para la instalación del SDK revisar:

https://github.com/ldrobotSensorTeam/ldlidar_stl_ros

```

1 import rospy
2 import math
3 import numpy as np
4 from geometry_msgs.msg import PoseStamped
5 from nav_msgs.msg import Path
6 from sensor_msgs.msg import LaserScan
7 import pandas as pd
8 from scipy.interpolate import UnivariateSpline
9
10 class RobotFulguracion:
11     def __init__(self):
12         rospy.set_param('/use_sim_time', False)
13         rospy.init_node('robot_fulguracion')
14         self.lidar_sub = rospy.Subscriber('/scan', LaserScan, self.lidar_callback)
15         self.lidar_data = None
16         self.casinos = None
17         self.trayectoria = None
18         self.rate = rospy.Rate(30)
19         # Variables de control
20         self.pos_actual = (0, 0) # Inicializar la posición actual del
21         self.dir_actual = 0.0
22         self.lidar_pub = None
23         print("nodo iniciado")
24
25
26     def lidar_callback(self, data):
27         self.lidar_data = data
28         rospy.loginfo("Received LIDAR data")
29         #print(data)
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

ANEXO I: Recolección de datos de entrenamiento de la MLP

```

class MLP:
    def __init__(self):
        self.modelo = self.build_mlp_model()
        self.load_weights()

    def build_mlp_model(self):
        model = tf.keras.Sequential()
        model.add(tf.keras.layers.Dense(12, activation='relu', input_shape=(2,)))
        model.add(tf.keras.layers.Dense(12, activation='relu'))
        model.add(tf.keras.layers.Dense(1))
        return model

    def train(self, x_train, y_train, x_val, y_val):
        optimizer = tf.keras.optimizers.Adam()
        loss = tf.keras.losses.MeanSquaredError()
        self.model.compile(optimizer=optimizer, loss=loss)

        epochs = 10
        batch_size = 32

        # Definir el callback de EarlyStopping
        early_stopping_callback = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=1, verbose=1, restore_best_weights=True)

        # Definir el callback de ModelCheckpoint
        model_checkpoint_callback = tf.keras.callbacks.ModelCheckpoint(filepath='mlp_weights.h5', monitor='val_loss', save_best_only=True, save_weights_only=True)

        # Entrenar el modelo con los callbacks
        self.model.fit(x_train, y_train, epochs=epochs, batch_size=batch_size,
                      validation_data=(x_val, y_val), callbacks=[early_stopping_callback, model_checkpoint_callback])

    def test(self, x_test, y_test, verbose=0):
        # Evaluar el modelo en el conjunto de prueba
        test_loss = self.model.evaluate(x_test, y_test, verbose=0)
        print("Pérdida en el conjunto de prueba:", test_loss)

# Cargar el modelo con los mejores pesos (seleccionados por el callback ModelCheckpoint)
mlp.load_weights()

# Evaluar el modelo en el conjunto de prueba
test_loss = mlp.model.evaluate(x_test, y_test, verbose=0)
print("Pérdida en el conjunto de prueba:", test_loss)

# Pesos cargados exitosamente.
Epoch 2/10
2/12 [=====] - 1s 26ms/step - loss: 18.2182 - val_loss: 79.1180
Epoch 3/10
3/12 [=====] - 0s 98ms/step - loss: 18.8600 - val_loss: 51.5008
Epoch 4/10
4/12 [=====] - 0s 96ms/step - loss: 21.0230 - val_loss: 16.6185
Epoch 5/10
5/12 [=====] - 0s 96ms/step - loss: 22.7280 - val_loss: 29.7098
Epoch 6/10
6/12 [=====] - 0s 96ms/step - loss: 21.2180 - val_loss: 28.8710
Epoch 7/10
7/12 [=====] - 0s 96ms/step - loss: 26.0030 - val_loss: 21.6590
Epoch 8/10
8/12 [=====] - 0s 96ms/step - loss: 25.7640 - val_loss: 19.2600
Epoch 9/10
9/12 [=====] - 0s 96ms/step - loss: 26.7870 - val_loss: 20.7320
Epoch 10/10
10/12 [=====] - 0s 96ms/step - loss: 29.6180 - val_loss: 17.6154
Epoch 11/10
11/12 [=====] - 0s 96ms/step - loss: 24.7240 - val_loss: 22.7240
Epoch 12/10
12/12 [=====] - 0s 96ms/step - loss: 26.8240 - val_loss: 20.1240
Epoch 13: early stopping
Pesos cargados exitosamente.
Pérdida en el conjunto de prueba: 1.1499113068908

```

ANEXO J: Entrenamiento de la MLP y almacenamiento de los pesos resultantes

Mode 3: Simplified Serial Mode

Simplified serial uses TTL level single-byte serial commands to set the motor speed and direction. This makes it easy to interface to microcontrollers and PCs, without having to implement a packet-based communications protocol. Simplified serial is a one-direction only interface. The transmit line from the host is connected to S1. The host's receive line is not connected to the Sabertooth. Because of this, multiple drivers can be connected to the same serial transmitter. If using a true RS-232 device like a PC's serial port, it is necessary to use a level converter to shift the -10V to 10V to 232 levels to the 0V-5V TTL levels the Sabertooth is expecting. This is usually done with a Max232 type chip. If using a TTL serial device like a microcontroller, the TX line of the microcontroller may be connected directly to S1.

Because Sabertooth controls two motors with one 8 byte character, when operating in Simplified Serial mode, each motor has 7 bits of resolution. Sending a character between 1 and 127 will control motor 1. 1 is full reverse, 64 is stop and 127 is full forward. Sending a character between 128 and 255 will control motor 2. 128 is full reverse, 192 is stop and 255 is full forward. Character 0 (hex 0x00) is a special case. Sending this character will shut down both motors.

Baud Rate Selection

Simplified Serial operates with an 8N1 protocol - 8 data bytes, no parity bits and one stop bit. The baud rate is selected by switches 4 and 5 from the following 4 options

2400 Baud: 01x0x	9600 Baud: 01x10x
19200 Baud: 01x01x	38400 Baud: 01x11x

What baud rate to use is dependent on what your host can provide and the update speed necessary. 9600 baud or 19200 baud is recommended as the best starting points. If communication is unreliable, decrease the baud rate. If communications are reliable, you may increase the baud rate. The maximum update speed on the Sabertooth is approximately 2000 commands per second. Sending characters faster than this will not cause problems, but it will not increase the responsiveness of the controller either.

```

class SabertoothController:
    def __init__(self):
        GPIO.setmode(GPIO.BOARD)
        GPIO.setup(12, GPIO.OUT)
        GPIO.setup(18, GPIO.OUT)
        self.pwm_speed = GPIO.PWM(12, 350) # frecuencia de 350Hz para control de velocidad
        self.pwm_direction = GPIO.PWM(18, 350) # frecuencia de 350Hz para control de dirección

        # Iniciar señales PWM con un ciclo de trabajo del 0% (detener el motor al inicio)
        self.pwm_speed.start(0)
        self.pwm_direction.start(0)

    def enviar_datos(self, velocidad, direccion):
        # Escalar los valores de velocidad a valores enteros para el controlador sabertooth
        velocidad_pwm = int(velocidad) # valores enteros
        direccion_pwm = int(direccion) # valores enteros

        # Actualizar las señales PWM con los valores ajustados
        self.pwm_speed.ChangeDutyCycle(velocidad_pwm)
        self.pwm_direction.ChangeDutyCycle(direccion_pwm)

    def detener(self):
        # Detener el motor (0% de ciclo de trabajo en ambos pines)
        self.pwm_speed.ChangeDutyCycle(0)
        self.pwm_direction.ChangeDutyCycle(0)

    def __del__(self):
        # Detener las señales PWM y limpiar los recursos al finalizar
        self.pwm_speed.stop()
        self.pwm_direction.stop()
        GPIO.cleanup()

```

ANEXO K: Clase "SabertoothController" y conexión de los pines

```
class PumpController:
    def __init__(self):
        self.pin_bomba = 22

        GPIO.setwarnings(False)
        GPIO.setmode(GPIO.BOARD) # Establecer el modo a "BOARD"
        GPIO.setup(self.pin_bomba, GPIO.OUT)
        self.pwm_bomba = GPIO.PWM(self.pin_bomba, 1000) # Frecuencia de 1000 Hz
        self.pwm_bomba.start(0) # Iniciar el PWM con un ciclo de trabajo del 0%

    def encender(self):
        voltaje = 12.0 # Voltaje de encendido
        ciclo_trabajo = voltaje * 100 / 10.0 # Mapear el rango (0, 10) a (0, 100)
        self.pwm_bomba.ChangeDutyCycle(ciclo_trabajo)
        GPIO.output(self.pin_bomba, GPIO.HIGH) # Establecer el pin en estado alto (bomba encendida)

    def apagar(self):
        self.pwm_bomba.ChangeDutyCycle(0) # Establecer el ciclo de trabajo en 0 (apagar el PWM)
        GPIO.output(self.pin_bomba, GPIO.LOW) # Establecer el pin en estado bajo (bomba apagada)
```



ANEXO L: Clase “PumpController” para el control de la bomba



**ESCUELA SUPERIOR POLITÉCNICA DE
CHIMBORAZO**

**DIRECCIÓN DE BIBLIOTECAS Y RECURSOS DEL
APRENDIZAJE**



UNIDAD DE PROCESOS TÉCNICOS
REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 22/11/2023

INFORMACIÓN DEL AUTOR
Nombres – Apellidos: Christian Fernando Aguagallo Coello
INFORMACIÓN INSTITUCIONAL
Facultad: Informática y Electrónica
Carrera: Electrónica y Automatización
Título a optar: Ingeniero en Electrónica y Automatización
f. Analista de Biblioteca responsable:  Ing. Fernanda Arévalo M.

