



**IMPLEMENTACIÓN DE UN ROBOT MOVIL PROTOTIPO PARA DESCARGA
DE SEMILLA EN HILERAS DE TERRENO PREPARADO PARA CULTIVO**

**DIEGO ANDRÉS CHÁVEZ BENÍTEZ
CÓD: 20441711894
WILLIAM ALFONSO APACHE CARO
COD: 20441811550**

Universidad Antonio Nariño

Programa Ingeniería Electrónica

Facultad de Ingeniería Mecánica, Electrónica y Biomédica

Villavicencio, Colombia

2023

**IMPLEMENTACIÓN DE UN ROBOT MOVIL PROTOTIPO PARA
DESCARGA DE SEMILLA EN HILERAS DE TERRENO PREPARADO
PARA CULTIVO**

**DIEGO ANDRÉS CHÁVEZ BENÍTEZ
WILLIAM ALFONSO APACHE CARO**

Proyecto de grado presentado como requisito parcial para optar al título de:
Ingeniero Electrónico

Director (a):
ING. ALBERTO ALFONSO VILLARRAGA

Línea de Investigación:
Robótica

Universidad Antonio Nariño
Programa Ingeniería Electrónica
Facultad de Ingeniería Mecánica, Electrónica y Biomédica
Villavicencio, Colombia

2023

NOTA DE ACEPTACIÓN

El trabajo de grado titulado

_____ ,

Cumple con los requisitos para optar

Al título de _____ .

Firma del Tutor

Firma Jurado

Firma Jurado

Contenido

1. RESUMEN	1
2. Abstract	2
3. INTRODUCCION Y ANTECEDENTES	3
OBJETIVOS.....	6
OBJETIVO GENERAL	6
OBJETIVOS ESPECÍFICOS	6
4. MARCO TEORICO.....	7
4.1 <i>Sistemas de control</i>	7
4.2 <i>Sistema de control de lazo cerrado</i>.....	7
4.3 <i>Sistema de control de lazo abierto</i>	8
4.4 <i>Raspberry Pi 4</i>.....	9
4.5 <i>Encoder</i>	11
4.6 <i>Sensor Inercial</i>.....	11
4.7 <i>Tecnología LIDAR</i>.....	12
4.8 <i>Puentes H</i>	12
4.9 <i>Lenguaje de programación Python</i>	13
4.10 <i>Lógica Difusa</i>	14
5. METODOLOGIA	15
5.1 FASE 1: RECONOCIMIENTO DE LA PLATAFORMA ROBÓTICA A UTILIZAR Y SUS SUBSISTEMAS	15
5.1.1 <i>MOTOR CON ENCODER</i>	16
5.1.2 <i>RASPBERRY PI 4B</i>	17
5.1.3 <i>SENSOR YDLIDAR X4</i>	20
5.1.4 <i>IMU BNO 055</i>.....	22
5.1.5 <i>CONTROL DE MOTORES</i>	23
5.1.6 <i>SISTEMA ELECTROMECAÁNICO DE CARGA Y DESCARGA DE SEMILLAS</i>	25
5.2 FASE 2: INSTALACION DE DISPOSITIVOS	27
6.2.1 <i>Sensor YDLIDAR X4</i>.....	27
5.2.2 <i>SENSOR INERCIAL IMU BNO055</i>	29
5.2.3 <i>ENCODERS DEL MOTOR</i>	32

5.3 FASE 3: IMPLEMENTAR UN SISTEMA DE CONTROL DE NAVEGACION.....	34
5.3.1 PERCEPCION DEL TELAR.....	34
5.3.2 PERCEPCION DEL TELAR CON ERROR DE ORIENTACION	35
5.4 FASE 5: IMPLEMENTAR UN SISTEMA DE CONTROL DE NAVEGACION.....	36
5.5 DEFINIR VARIABLES DE ENTRADA	37
5.6 FUZIFICACION.....	37
5.7 REGLAS DIFUSAS	38
5.8 DEFUZIFICACION	39
6. RESULTADOS.....	40
6.1 Prueba N°1. Prueba de Inclinación	41
6.1.1 Prueba de inclinación a 10°	41
6.1.2 Prueba de inclinación a 15°	43
6.1.3 Prueba de inclinación a 20°	45
6.1.4 Prueba de inclinación a 25°	46
6.2 Prueba N°2. Prueba en terreno irregular	48
6.3 Pruebas en terreno regular.	52
6.4 Prueba N°3. Navegación del robot expuesto a la luz solar durante el día.	57
6.4.1 Pruebas realizadas a las 8 y 10 Am	57
6.4.2 Pruebas realizadas de 2 a 4 Pm	58
6.5 Prueba realizada a las 12 Pm.....	60
7. Conclusiones	63
8. Anexos	64
8.1 ANEXO 1: CONFIGURACION DEL SENSOR LIDAR.....	64
8.1 ANEXO 2: CONFIGURACION DEL SENSOR BNO055	65
9. Referencias Bibliográficas	66

Lista de Figuras

Figura 1.	Diagrama de bloques de un sistema de control de lazo cerrado.	8
Figura 2.	Diagrama de bloques de un sistema de control de lazo abierto.	9
Figura 3.	Tarjeta Raspberry Pi 4	9
Figura 4.	Estructura de un puente H	12
Figura 5.	Líneas de código para el ejemplo de “hello world” en C++ y en Python	14
Figura 6.	Chasis del robot.	15
Figura 7.	Motor con encoder integrado.	16
Figura 8.	Placa Raspberry pi 4 B.	18
Figura 9.	YDLIDAR X4.	20
Figura 10.	Funcionamiento del sensor YDLIDAR X4.....	21
Figura 11.	IMU BNO055.	22
Figura 12.	Puente H L298n.	24
Figura 13.	Mecanismo de carga y descarga de semillas	25
Figura 14.	Sistema electromecánico final.	26
Figura 15.....	Robot sowwall.	26

Figura 16. Vista frontal y lateral del robot sowwall	27
Figura 17. Adaptador USB.	28
Figura 18. Conexión Raspberry pi 4 y IMUBNO055.	30
Figura 19. Esquema encoder.....	32
Figura 20. Ventana angular definida en el LIDAR.	34
Figura 21. Percepción con error de orientación.....	35
Figura 22. Conjuntos difusos de entrada.....	38
Figura 23. Conjuntos difusos de salida.	38
Figura 24. Control difuso, error de orientación 0°.....	39
Figura 25. Control difuso, error de orientación -3°.....	40
Figura 26. Grafica de los datos tomados tomados en una inclinación de 10°	42
Figura 27. Robot prueba de inclinación a 10°	43
Figura 28. Robot prueba de inclinación a 15°	44
Figura 29. Grafico de los datos tomados para la prueba de 10° y 15°	44
Figura 30. Robot prueba de inlcinacion a 20°	45
Figura 31. Grafico de los datos tomados para la prueba de 10°, 15° y 20°	46
Figura 32. Robot prueba de inclinación a 25°	47

Figura 33. Grafico de los datos tomados para las prueba de 10°, 15°, 20 y 25°	48
Figura 34. Terreno con irregularidades en la superficie	49
Figura 35. Dimensiones de los surcos a lo largo.....	50
Figura 36. Profundiada de los surcos.	50
Figura 37. Datos arrojados por pruebas realizadas en un terreno irregular.	51
Figura 38. Correcciones realizada por el prototipo en un terreno irregular.	52
Figura 39. Terreno regular adecuado	53
Figura 40. Prueba de campo.	54
Figura 41. Datos tomados de las pruebas realizadas en un terreno regular.	55
Figura 42. Esparcimiento de las semillas durante las pruebas	56
Figura 43. Lúmenes a las 8 y 10 de la mañana.	57
Figura 44. Datos arrojados en las pruebas realizadas a las 8 y 10 AM.....	58
Figura 45. Lumenes medidos por el luxometro a las 2 y 4 de la tarde.	59
Figura 46. Datos arrojados en las pruebas realizadas a las 2 y 4 PM.	60
Figura 47. Registro tomado por el luxómetro a las 12 Pm del día.....	60
Figura 48. Robot con la cubierta diseñada para porteger el Lidar	61
Figura 49. Datos arrojados para la pruebas realizadas con la cubierta y sin ella.	62

Lista de Tablas

Tabla 1.	Especificaciones Raspberry pi 4.....	10
Tabla 2.	Especificaciones del motor.....	17
Tabla 3.	Características de la Raspberry pi 4B.....	19
Tabla 4.	Características del sensor YDLIDAR X4.....	21
Tabla 5.	Características del Puente H.....	24
Tabla 6.	Pruebas del sensor YDLIDAR X4.....	28
Tabla 7.	Errores absolutos de las medidas del sensor YDLIDAR X4.....	29
Tabla 8.	Pruebas del sensor IMU BNO055.....	31
Tabla 9.	Errores absolutos de los datos de la IMU BNO055.....	31
Tabla 10.	Pruebas tomadas con los encoders.....	33
Tabla 11.	Errores absolutos del encoder.....	33
Tabla 12.	Pasos del sistemas de control difuzo.....	36

Lista de Símbolos y Abreviaturas

Símbolos con letras latinas

Símbolo	Término	Unidad SI	Definición
A	Área	m^2	$\iint dx dy$
A_{BET}	Área interna del sólido	$\frac{m^2}{g}$	ver DIN ISO 92
A_g	Área transversal de la fase gaseosa	m^2	Ec. 3.2
A_s	Área transversal de la carga a granel	m^2	Ec. 3.6
a	Coefficiente	1	Tabla 3-1

Símbolos con letras griegas

Símbolo	Término	Unidad SI	Definición
α_{BET}	Factor de superficie	$\frac{m^2}{g}$	$(W_{F,wat})(A_{BET})$
β_i	Grado de formación del componente i	1	$\frac{m_j}{m_{bm} \rho}$
r	Wandhaufreiwinkel (Stahlblech)	1	Sección 3.2
ϵ	Porosidad de la partícula	1	$1 - \frac{\rho_s}{\rho_w}$
η	mittlere Bettneigungswinkel (Stürzen)	1	Figura 3-1

Subíndices

Subíndice	Término
	Materia orgánica
DR	Dubinin-Radushkevich
E	Experimental

Superíndices

Superíndice	Término
n	Exponente, potencia

Abreviaturas

Abreviatura	Término
1.LT	Primera ley de la termodinámica
DF	Dimension fundamental
RFF	Racimos de fruta fresca

(Dedicatoria)

En dedicatoria a nuestras familias queremos presentar este proyecto de grado, gracias a ellos, a su apoyo incondicional podemos el día de hoy con mucho orgullo decir que siempre serán para nosotros, aquel tesoro que no brilla como el oro, pero cuestan lo mismo y mucho más.

En este momento nos encontramos terminando de redactar este documento, agotados y exhaustos, pero con una sonrisa en el rostro...

“La alegría de ver y entender es el más perfecto don de la naturaleza”

Albert Einstein

Agradecimientos

Antes que nada quiero dar gracias a Dios por permitirnos terminar este proceso, que con mucho esfuerzo y dedicacion sacamos adelante. No fue un proceso facil, pues desde que iniciamos nuestros estudios para optar por el titulo de profesional en Ingenieria Electronica, hemos pasado momentos dificiles.

A su vez queremos expresar el mas sincero agradecimiento al Doctor Hnerry Borrero quien nos compartio su tiempo y conocimiento para hacer posible el desarrollo exitoso de este proyecto. A la Ingeniera Diana Diaz por su guia experta, dedicacion y apoyo, gracias a ellos por creer en nostoros en todo momento, por insistir una y otra vez. Nos sentimos muy afortunados de haber contado con su valioso apoyo en cada paso del camino.

A nuestras familias, quienes nos dieron ese apoyo moral, tambien a nuestros compañeros con quienes hemos compartido mano a mano y codo a codo, la gran trayectoria de preocupaciones, alegrías, tristezas para superar cada una de las dificultades que se presentaron, a ellos y a toda la UAN queremos decirles GRACIAS.

1. RESUMEN

Este documento presenta un trabajo de grado, en el programa de ingeniería electrónica de la sede Villavicencio, de la Universidad Antonio Nariño, en el cual se desarrolla un robot móvil prototipo autónomo que pueda descargar semillas en hileras de terreno preparado para el cultivo. El robot cuenta con un sistema de locomoción tipo oruga lo que lo hace más estable y con mejor maniobrabilidad en terrenos irregulares, además de contar con un sensor LIDAR, que se utiliza para recolectar información del entorno e implementar un sistema de navegación autónoma reactiva, permitiendo una mejor precisión del movimiento del robot tomando como punto de referencia un obstáculo a un costado del robot, también cuenta con una IMU (Unidad de Medición Inercial), que se utiliza para mantener una dirección fija durante todo el proceso de la descarga de semillas. La IMU es capaz de detectar cualquier cambio en la orientación del robot, y envía señales al sistema de control para corregir cualquier desviación, permitiendo así que el sistema electromecánico del robot, responsable de almacenar y liberar las semillas en el terreno de cultivo a intervalos regulares, lo haga de una manera más eficiente. Para llevar a cabo todo el funcionamiento autónomo del robot se utiliza como controlador central una Raspberry Pi 4, siendo esta nuestro componente principal para el desarrollo del proyecto presentado en este informe.

Palabras Claves: (IMU, Robot Móvil, Sistema Electromecánico, Controlador)

2. Abstract

This document presents a degree work, in the electronic engineering program of the Villavicencio branch of the Antonio Nariño University, in which an autonomous prototype mobile robot is developed that can unload seeds in rows of land prepared for cultivation. The robot has a caterpillar type locomotion system which makes it more stable and with better maneuverability on uneven terrain, in addition to having a LIDAR sensor, which is used to collect information from the environment and implement a reactive autonomous navigation system, allowing a better precision of the robot's movement taking as a reference point an obstacle on one side of the robot, it also has an IMU (Inertial Measurement Unit), which is used to maintain a fixed direction throughout the process of unloading seeds. The IMU is able to detect any changes in the robot's orientation, and sends signals to the control system to correct any deviations, thus allowing the robot's electromechanical system, which is responsible for storing and releasing the seeds into the field at regular intervals, to do so more efficiently. To carry out all the autonomous operation of the robot, a Raspberry Pi 4 is used as a central controller, being this our main component for the development of the project presented in this report.

Keywords: (IMU, Mobile Robot, Electromechanical System, Controller)

3. INTRODUCCION Y ANTECEDENTES

En la agricultura moderna la siembra es una tarea crucial para el éxito productivo de los cultivos, de tal manera que, en los últimos años desde la robótica se ha demostrado interés en el desarrollo de aplicaciones que contribuyan al respecto especialmente cuando se trata de sembrar en grandes extensiones de tierra. Una manifestación destacada, orientada al desarrollo de una solución a este problema corresponde a la implementación del robot dispensador de semillas reportado en [1][2].

Una revisión bibliográfica en una ventana de observación correspondiente a los últimos 6 años, permite verificar que existen varios tipos de robots móviles para la siembra de cultivos. Por ejemplo, algunos robots utilizan tecnología de visión por computadora para identificar la posición y la distancia entre las semillas. Otros robots utilizan sensores de profundidad para medir la profundidad de siembra y, además, algunos robots están equipados con sistemas de guiado por GPS para mantener una dirección fija, como lo puede ser el proyecto presentado por Santhi, Nellore, Vijay KR, & CH Venkata, 2017, el cual consiste en la implementación de un robot agrícola, basado en sensores y visión por computadora para la siembra de semillas a su vez que navega de forma autónoma mediante un sistema de posicionamiento global (GPS) [3].

Por otra parte, el proyecto presentado por SM Khot, Alabado, Nitesh, Jabes, Atul, Aditya, & Joel, 2021, habla sobre la implementación de un robot sembrador de semillas que es capaz de realizar la acción de arado, descargue de semillas mediante un sistema de rueda dosificadora y cubrimiento de la semilla con tierra. Para la navegación autónoma se tomó

las entradas de un magnetómetro y se mejoró la precisión de navegación alineando la entrada a un controlador integral proporcional (PI) [4].

A pesar de los avances en la tecnología robótica para la agricultura, aún existen desafíos que limitan la adopción de robots en el campo. Uno de los principales desafíos es el costo de los robots, que puede ser prohibitivo para muchos agricultores, por esta razón Jayakrishna, Suryavamsi, Jaswanth , N Susheel , & KP Peeyush, 2018, presentaron un robot agrícola autónomo de siembra de semillas; el robot presenta tracción en las cuatro ruedas y se diseñó para tierras agrícolas aradas, para la siembra de las semillas se tuvieron en cuenta los parámetros del área de campo y los intervalos de espacio entre semillas dadas por el usuario dando como consecuencia que el robot pueda con estos datos a su vez navegar de forma autónoma manteniendo una distancia de protección a cada lado del campo de 0.74 pies [5].

En el área de sensores de profundidad Taha, Patel, Ahmed, Manik, & Dalwinder, 2022 presentaron el diseño y desarrollo de un robot agrícola para siembra de semillas, aspersión de agua y fertirrigación, donde se utilizan sensores de temperatura y un sensor de humedad para detectar el PH del suelo, junto a otros más que determinaran la cantidad de semillas, agua y fertilizantes disponibles en cada tanque. Para la navegación del robot se determinó el uso de sensores infrarrojo (IR) que detectaran las paredes circundantes, a su vez se implementó una cámara que permitirá al agricultor visualizar el suelo y los obstáculos durante las operaciones [7]. Por su parte Tanmay, Pranay, Hrushikesh, Sushobhit, neema, & saurabh, 2020 realizaron el diseño de un robot de siembra para semillas de algodón BT utilizando un mecanismo que cuenta de a dos semillas con precisión mediante una

configuración de Arduino Laser LDR para luego dejarlas caer a través de un tubo guía. Con sensores ultrasónicos (Hc-SRo4) se realiza la navegación del robot siguiendo un mismo camino recto de una profundidad constante, comunicándose continuamente con la unidad de control para evitar que este se desvíe [8].

Para la presentación de este informe, se plantea a lo largo del trabajo tres capítulos principales estructurados en el siguiente orden: Marco teórico, Metodología y Resultados. En el primer capítulo se muestran los conceptos necesarios que se utilizan con el fin de comprender el desarrollo teórico del proyecto, para el segundo capítulo se aborda toda la parte técnica, como lo es el planeamiento, diseño y construcción del prototipo; Finalmente, el último capítulo, muestra los resultados que se obtienen al poner en práctica nuestro diseño sobre diferentes campos y condiciones.

En resumen, el desarrollo de esta tesis busca presentar una contribución innovadora para la siembra de semillas en la agricultura, mediante el diseño y desarrollo de un robot dispensador de semillas que navega de forma autónoma.

OBJETIVOS

OBJETIVO GENERAL

Implementar un robot móvil prototipo que descargue semillas en hileras de un terreno preparado para cultivo de acuerdo con una separación homogénea.

OBJETIVOS ESPECÍFICOS

- Configurar un sistema de percepción que permita detectar el entorno para recorrer las hileras sobre las cuales se debe descargar la semilla.
- Implementar un subsistema electromecánico para almacenar y liberar semillas durante los momentos de desplazamiento autónomo del prototipo
- Configurar un sistema de control de lazo cerrado para manejar la navegación autónoma del prototipo en las hileras correspondientes para siembra.
- Evaluar a través de la realización de pruebas de campo abierto, el desempeño general del robot móvil en la realización de la tarea propuesta.

4. MARCO TEORICO

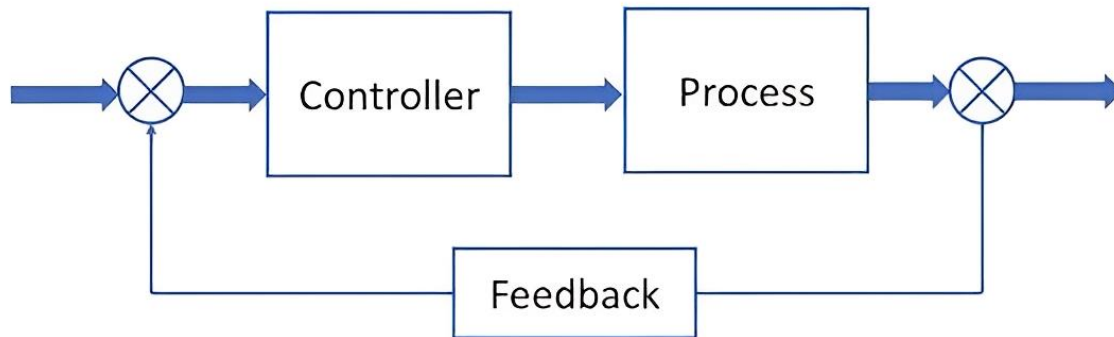
4.1 *Sistemas de control*

Un sistema de control es un conjunto de elementos que se utilizan para mantener un proceso o sistema en un estado deseado o estable. Estos sistemas se utilizan en una amplia variedad de áreas, desde la industria hasta la robótica, pasando por la ingeniería, la electrónica y la automatización. Existen diferentes tipos de sistemas de control, como los sistemas de control de lazo abierto, los sistemas de control de lazo cerrado, los sistemas de control en cascada, entre otros. Cada uno de ellos se adapta a diferentes necesidades y características del proceso que se desea controlar [9].

4.2 *Sistema de control de lazo cerrado*

El lazo cerrado se llama así porque hay una "retroalimentación" que cierra el circuito entre la salida y la entrada del sistema. La retroalimentación proporciona información al sistema sobre su desempeño en tiempo real, lo que le permite ajustar su entrada para mantener la salida en el valor deseado. En otras palabras, el sistema mide continuamente su salida y la compara con el valor deseado, y ajusta su entrada para corregir cualquier desviación [9].

Figura 1. Diagrama de bloques de un sistema de control de lazo cerrado.



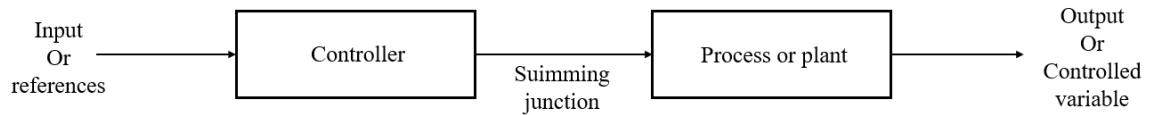
Fuente: (Devasia, 2023)..

Un ejemplo de control de lazo cerrado podemos decir el sistema de control de temperatura en un horno: El termostato mide continuamente la temperatura dentro del horno y ajusta la entrada de combustible para mantener la temperatura constante [9].

4.3 Sistema de control de lazo abierto

Un sistema de control de lazo abierto es un tipo de sistema de control en el que la salida del sistema no se retroalimenta para ajustar la entrada. En otras palabras, el sistema no utiliza la información de salida para corregir o ajustar la entrada. En lugar de eso, la entrada se establece en un valor fijo y se espera que el sistema produzca la salida deseada. Aunque los sistemas de control de lazo abierto son más simples que los sistemas de control de lazo cerrado, son menos precisos y no pueden adaptarse a cambios en las condiciones del sistema o a perturbaciones externas. Por lo tanto, se utilizan con mayor frecuencia en aplicaciones donde se requiere una precisión menor o donde las condiciones del sistema son estables y predecibles constante [10].

Figura 2. Diagrama de bloques de un sistema de control de lazo abierto.



Fuente: “Autor” (2023).

El sistema de control de lazo cerrado se divide en dos partes que son el controlador y el proceso controlado. Al ingresar al sistema una señal de entrada o una referencia, esta actúa directamente con el controlador, para luego interactuar con una variable o proceso que regulará el proceso de tal forma que la salida se desempeñe a ciertas especificaciones [10].

4.4 Raspberry Pi 4

Es una computadora de placa única (SBC, por sus siglas en inglés) diseñada por la Fundación Raspberry Pi con un sistema operativo Linux que permite al usuario explorar la computación y aprender lenguajes de programación como lo es Python. Es una de las últimas versiones de la serie Raspberry Pi que se lanzó en el año 2019 [11].

Figura 3. Tarjeta Raspberry Pi 4



Fuente: “¿Qué es Raspberry Pi?” (2019).

Raspberry Pi 4 es una plataforma muy popular para proyectos de computación de bajo costo, desde sistemas de automatización del hogar hasta proyectos educativos. Además, su tamaño compacto la hace ideal para aplicaciones de Internet de las cosas (IoT) [11].

Tabla 1. Especificaciones Raspberry pi 4

Procesador	Broadcom BCM2711
CPU	1.5Ghz Quad Core
GPU	500MHz
RAM	2GB/4GB/8GB
USB	2 USB 2.0 / 2 USB 4.0
Video	Jack, 2 Micro HDMI
Audio	Jack, Micro HDMI
Memoria	Micro SD
Wireless	WiFi 2.4 y 5 GHz Blouethooth 5.0BLE
Ethernet	Gigabit Ethernet 1000
Alimentación	5V 3Amp
GPIO	40 pines
Dimensiones	88x58x19 mm

Fuente: “¿Qué es Raspberry Pi?” (2019).

4.5 *Encoder*

Es un dispositivo que convierte la posición o el movimiento de un objeto físico en una señal digital o analógica que puede ser procesada por un sistema electrónico. En otras palabras, un encoder se utiliza para medir la posición, la velocidad o la dirección de rotación de un eje o de una parte móvil de una máquina, motores, sistemas de control de movimiento, robots, entre otros (Cruz Contreras Edgar A. & Portilla Ramon Silva, 2018).

Un encoder se basa en un principio físico llamado "efecto Hall", el dispositivo consta de un disco giratorio que tiene un patrón de marcas o ranuras a lo largo de su borde, estas marcas se detectan mediante un sensor llamado sensor de efecto Hall. Cuando el disco gira, el sensor de efecto Hall detecta las marcas y envía una señal eléctrica al procesador que convierte esta señal en información sobre la posición y la velocidad de rotación del disco (Cruz Contreras Edgar A. & Portilla Ramon Silva, 2008).[13].

4.6 *Sensor Inercial*

Un sensor inercial o también denominado Unidad de mediad inercial (IMU) mide la aceleración y la velocidad angular de un objeto en movimiento. Estos sensores funcionan midiendo las fuerzas que actúan sobre un conjunto de piezas mecánicas o electrónicas que se mueven dentro del sensor. La aceleración se mide midiendo la fuerza que actúa sobre un objeto en movimiento, mientras que la velocidad angular se mide midiendo la velocidad de rotación del objeto. Los sensores inerciales se utilizan comúnmente en sistemas de navegación y control de movimiento, como en aviones, robots y vehículos autónomos. Estos sensores pueden proporcionar información crítica sobre la posición, la velocidad y la

orientación del objeto en el espacio, lo que permite a los sistemas de control ajustar y corregir el movimiento del objeto de manera precisa y rápida [6][14].

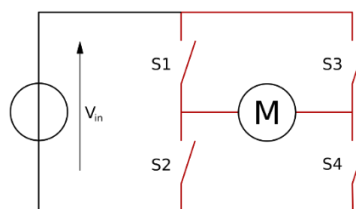
4.7 Tecnología LIDAR

LIDAR es el acrónimo de "Light Detection and Ranging" y se refiere a un tipo de sensor que utiliza pulsos láser para medir la distancia entre el sensor y un objeto o superficie. Los sensores LIDAR emiten pulsos de luz láser y miden el tiempo que tarda en regresar la luz reflejada por los objetos del entorno. A partir de esta información, se puede crear un mapa 3D de la zona escaneada, lo que permite al sensor "ver" los objetos y medir su distancia con gran precisión [15].

4.8 Puentes H

Un puente H es un circuito electrónico que se utiliza para controlar la dirección del flujo de corriente eléctrica a través de un motor, permitiéndole girar en ambos sentidos, son bastante usados en robótica y como convertidores de potencia. El nombre de "puente H" proviene de la forma en que los interruptores están conectados en el circuito, creando una configuración que se asemeja a la letra "H".

Figura 4. Estructura de un puente H



Fuente: "Diseño y construcción de un controlador inalámbrico a través de microprocesadores para el funcionamiento de un automóvil eléctrico a escala, controlado desde un computador" (2013)

El circuito está formado por cuatro interruptores (transistores o relés), dispuestos en una configuración en forma de puente. Al controlar la activación de los interruptores de manera adecuada, es posible invertir la polaridad de la corriente que llega al motor, permitiéndole girar en ambos sentidos (Roberto C. Guamán & Ángel E. Villava, 2013).

4.9 Lenguaje de programación Python

Python es un lenguaje de programación de alto nivel, interpretado y orientado a objetos. Fue creado por Guido van Rossum en la década de 1990 y ha ganado una gran popularidad debido a su sintaxis clara y legible, lo que lo hace ideal para principiantes en la programación. Python se utiliza en una amplia variedad de aplicaciones, como desarrollo web, análisis de datos, inteligencia artificial, aprendizaje automático, automatización de tareas, entre otros. Python es multiplataforma, lo que significa que puede ser utilizado en diferentes sistemas operativos, como Windows, Linux y macOS. Además, cuenta con una gran comunidad de usuarios que contribuyen al desarrollo de bibliotecas y módulos para hacer más fácil la programación en Python. Si hacemos una comparación entre el programa “Hola Mundo” de python y el programa “Hola Mundo” en C++, podemos notar la gran diferencia que conlleva realizar la misma acción en cantidad de líneas de código que requiere cada lenguaje de programación. [18][17].

Figura 5. Líneas de código para el ejemplo de “hello world” en C++ y en Python

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Hello World" <<
endl;
    return 0;
}
```

a) Programa “Hola Mundo” en C++

```
print "Hello World"
```

b) Programa “Hola Mundo” en Python

Fuente: “The programming language Python” (2018)

4.10 Lógica Difusa

La lógica difusa fue planteada por Dr. Zadeh en 1965, esta hace referencia al comportamiento humano a la hora de tomar decisiones, donde se tienen en cuenta muchas consideraciones y experiencias. La lógica difusa se creó con la intención de preservar el concepto de vaguedad, que se ha mantenido a través de la historia y dado a conocer por las afirmaciones realizada desde Aristóteles hasta Zadeh manifestadas en sus artículos. Sin embargo, la lógica difusa a diferencia de la lógica clásica, tiene fronteras imprecisas como *estatura media* o *temperatura baja* adaptándose así mejor al mundo real en el que vivimos hasta tal punto que puede comprender y funcionar con expresiones como: “Mucho”, “Muy”, “Un poco” [19][20].

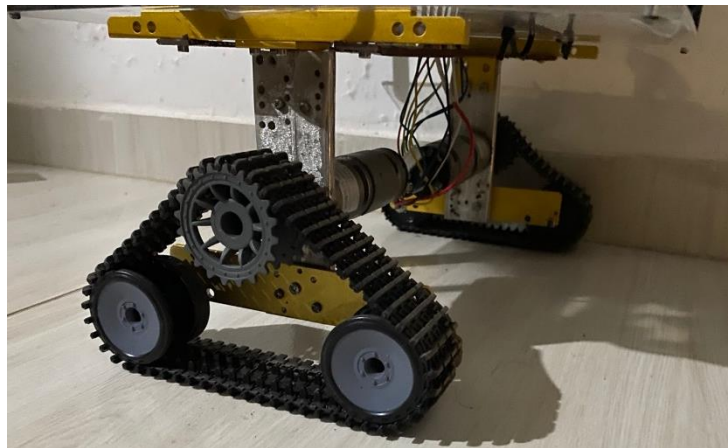
5. METODOLOGIA

El enfoque principal del proyecto propuesto se centra en implementar un sistema de control que le permita a un robot móvil navegar de forma autónoma, mientras realiza la descarga de semillas en hileras de un terreno preparado para cultivo. La metodología planteada se divide en las siguientes fases:

5.1 FASE 1: RECONOCIMIENTO DE LA PLATAFORMA ROBÓTICA A UTILIZAR Y SUS SUBSISTEMAS

El proceso comenzó con la identificación de los elementos clave utilizados en la creación del robot y como resultado, se obtuvo una comprensión más clara de cada elemento utilizado en la creación de este proyecto.

Figura 6. Chasis del robot.



Fuente: imagen propia

Pensando en una mejor movilidad para diferentes tipos de terreno, se decidió elaborar un chasis de tracción diferencial con ruedas tipo oruga (*figura 6*), El cual está hecho en aleación de aluminio metálico y un porcentaje pequeño en solo aluminio, cuenta con suspensión

independiente en cada una de las orugas, ya que esto le permite mejor movilidad en diferentes tipos de terreno. Sus dimensiones son:

Longitud: 19.5 cm

Ancho: 30 cm

Alto: 18 cm

Peso: 3.5 libras

5.1.1 MOTOR CON ENCODER

Para la realización del proyecto, se hace uso de unos motores con encoders incorporados, con un torque de 60 kg.cm, lo necesario para abordar un terreno previamente preparado para cultivo. Los encoders serán usados para contar sus pulsos por revolución (PPR) y así poder tener una estimación de la distancia recorrida por el robot, el motor con encoder utilizado para este proyecto se muestra en la siguiente figura.

Figura 7. Motor con encoder integrado.



Fuente: (Motor de Engranajes Planetarios ABHL 36GP 555.)

En la siguiente tabla se muestra alguna de las especificaciones del motor:

Tabla 2. Especificaciones del motor.

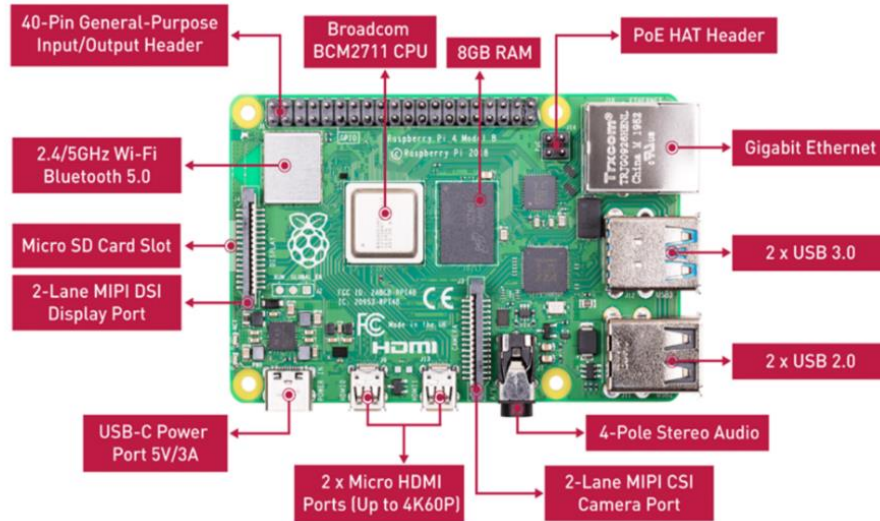
Tipo	Motor de engranajes
Voltaje motor	12 V
Torque	Max 60 kg.cm
Numero de pulsos (encoders)	17 PPR
Respuesta en frecuencia (encoders)	100 kHz
Voltaje funcionamiento (enconder)	3.3 V/5.0V

Fuente: (Motor de Engranajes Planetarios ABHL 36GP 555.)

5.1.2 RASPBERRY PI 4B

Para desarrollar el proyecto era necesario buscar una unidad de procesamiento que estuviera en la capacidad de procesar los datos suministrados por los diferentes sensores, además que no tuviera un costo muy elevado por lo cual se hizo uso de una tarjeta Raspberry pi 4 modelo B, ya que cuenta con una GPU (Unidad de Procesamiento Grafico), una CPU (Unidad Central de Procesamiento) Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) SoC de 64 bits 1,5 GHz, además de tener a su disposición muchos otros sistemas operativos [21].

Figura 8. Placa Raspberry pi 4 B.



Fuente: (Raspberry Pi 4 Computadoras Modelo B de 8 GB)

Esta placa es desarrollada por la compañía británica Raspberry Pi Foundation y fue anunciada en junio del 2019, a diferencia de sus predecesoras la Raspberry pi 3 e incluso las Raspberry pi 2 esta nueva placa tiene algunas mejoras que la diferencia de las anteriores tanto por más rendimiento en su procesador, así como por el uso incluso de dos pantallas. La tarjeta se trabajó con el sistema operativo de Linux, distribuido como Raspbian disponible de forma libre en la página oficial de raspberry.org [21].

Sus principales características se presentan en la siguiente tabla:

Tabla 3. Características de la Raspberry pi 4B.

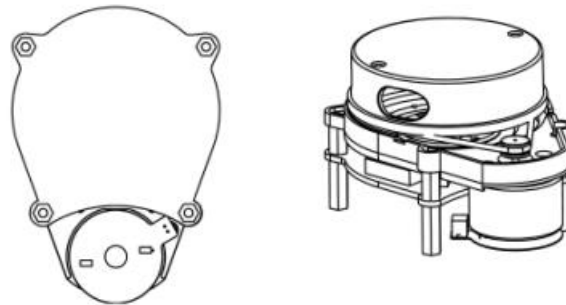
Hardware	Descripción
Procesador	Broadcom BCM2711
CPU	1.5GHz 64-bit Quad-core Cortex-A72
GPU	Broadcom Video Core VI, OpenGL ES 3.0, 1080p30 H.264/MPEG-4 AVC, 4kp60 H.265
Memoria	8GB
Puertos USB	2 puertos USB 2.0 2 puertos USB 3.0
Salidas de audio	Jack de 3.5 mm, 2 puertos microHDMI
Conectividad de red	Puerto RJ-45 10/100/1000Mbps vía hub USB 3.0 Wi-Fi 802.11ac de doble banda Bluetooth 5.0 BLE
Periféricos de bajo nivel	17 x GPIO y un bus HAT ID
Consumo energético	800 mA (4.0 W)

Fuente: (Raspberry Pi Documentation, 2023).

5.1.3 SENSOR YDLIDAR X4

El sensor seleccionado para la detección del entorno del proyecto es un YDLIDAR X4, desarrollado por EAI equipo. El cual realiza un escaneo laser omnidireccional con un rango de hasta 10 metros del entorno en el que se encuentre.

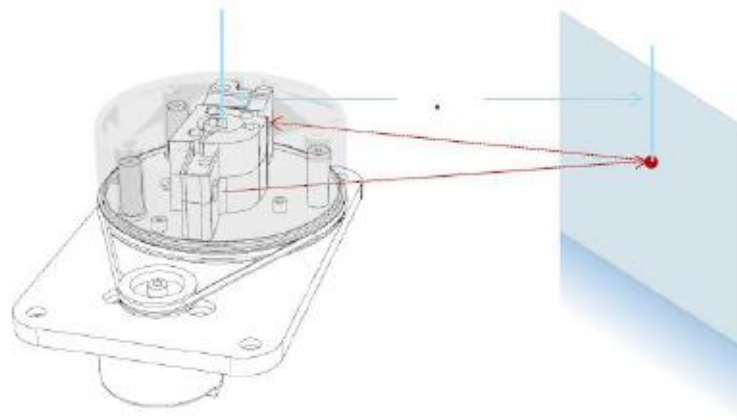
Figura 9. YDLIDAR X4.



Fuente: (SLAMTEC, 2016).

Este sensor funciona en el sentido de las manecillas del reloj, usando el principio de triangulación laser (ver figura 10), adquisición y procesamiento de datos. Funciona emitiendo una señal infrarroja modulada la cual es reflejada por el objeto al ser detectado como se observa en la siguiente figura. El sensor YDLIDAR X4 se conecta a través de un puerto serial USB a la Raspberry [22].

Figura 10. Funcionamiento del sensor YDLIDAR X4.



Fuente: (SLAMTEC, 2016).

Algunas de las características del sensor YDLIDAR X4 utilizado para el presente proyecto, se muestran a continuación en la siguiente tabla.

Tabla 4. Características del sensor YDLIDAR X4

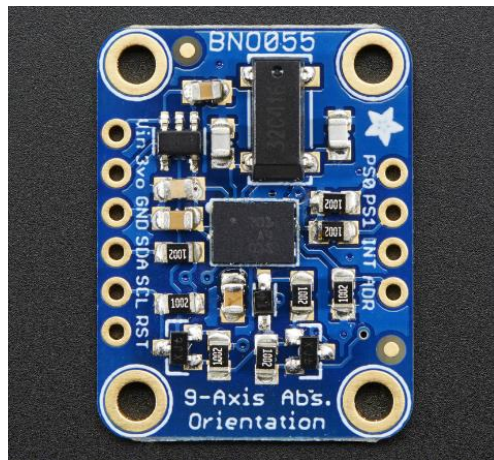
Alcance máximo	10 metros
Rango	360°
Resolución	0,86°
Frecuencia	5 KHz
Estándar de seguridad	Clase 1

Fuente: (YDLIDAR/Focus on Lidar Sensor Solutions, n.d.)

5.1.4 IMU BNO 055

La unidad de medida inercial (IMU) que se utilizó para la realización del presente proyecto fue la placa BNO055 de Adafruit, (*Figura 11*), la BNO055 es un sensor que posee 9 grados de libertad, además tiene la facilidad de que se pueden leer los datos arrojados a través de los puertos I2C (inter integrated circuits) o por medio de la UART (transmisor-receptor asincrónico universal) [23].

Figura 11. IMU BNO055.



Fuente: (Overview | Adafruit BNO055 Absolute Orientation Sensor | Adafruit Learning System, n.d.)

El sensor inercial IMU BNO055 nos puede generar los siguientes datos:

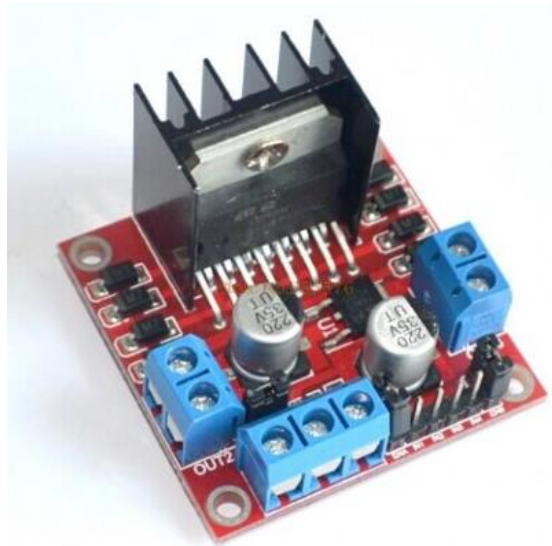
- Orientación absoluta. (Vector de Euler, 100Hz) datos de orientación de tres ejes basado en una esfera de 360°.
- Orientación absoluta. (Quaternion, 100 Hz).
- Vector de velocidad angular. (100 Hz), tres ejes de ‘velocidad de rotación’ en rad/s.
- Vector de aceleración. (100 Hz), tres ejes de aceleración (gravedad + movimiento lineal) en m/s^2 .

- Vector de fuerza de campo magnético. (20 Hz), tres ejes de detección de campo magnético en micro Tesla (uT).
- Vector de aceleración lineal (100 Hz). Tres ejes de datos de aceleración lineal (aceleración menos gravedad) en m/s^2
- Vector de gravedad (100 Hz). Tres ejes de aceleración gravitacional (menos cualquier movimiento) en m/s^2
- Temperatura (1 Hz). Temperatura ambiente en grados centígrados

5.1.5 CONTROL DE MOTORES

Ya que nuestro robot móvil prototipo es de tracción diferencial y sus ruedas son de tipo oruga para mejor maniobrabilidad en campo y en los diferentes terrenos, es necesario la utilización de un driver L298n (*Figura 12*). Se utilizaron dos de estos drivers, uno para el control de los motores de las ruedas y otro para el motor del sistema electromecánico de carga y descarga de las semillas, dicho un módulo cuenta con dos canales de salida ideal para el control de cambio de giro para cada uno de los motores dc, trabajan con un nivel lógico de 5V y admite entradas de PWM para el control de velocidad de los mismos [23].

Figura 12. Puente H L298n.



Fuente: (Módulo Controlador Puente H L298N,)

En la siguiente tabla se observan alguna de las características que posee el módulo puente H L298n.

Tabla 5. Características del Puente H

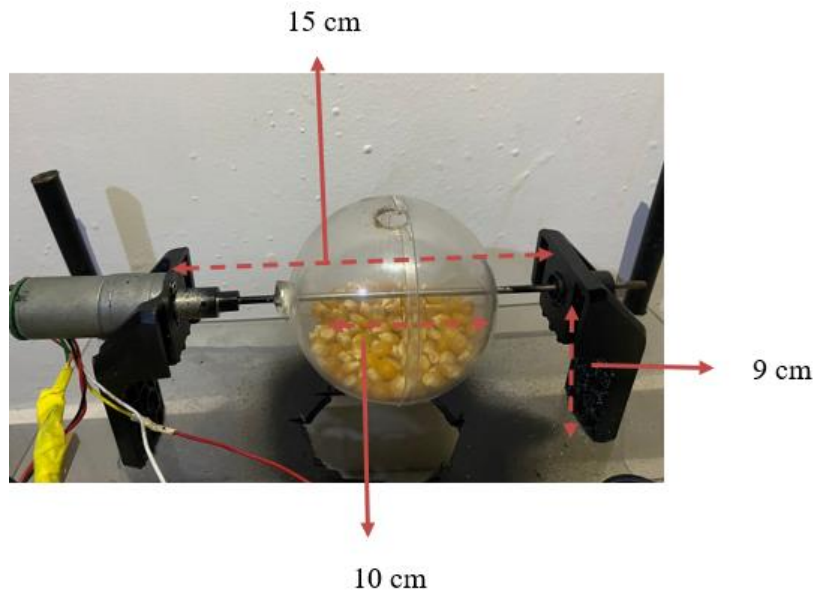
Canales	2 canales de manejo
Voltaje lógico	5V
Voltaje de operación	5V-35V
Consumo de corriente (Digital)	0 a 36 mA
Capacidad de corriente	2A (picos de hasta 3A)
Potencia máxima	25W
Peso	30g
Dimensiones	43 * 43 * 27 mm

Fuente: (Módulo Controlador Puente H L298N, n.d.)

5.1.6 SISTEMA ELECTROMECAÁNICO DE CARGA Y DESCARGA DE SEMILLAS

Ya que el proyecto consiste en la descarga de semillas cada cierta distancia en un cultivo preparado para siembra, para lograr tal objetivo era necesario la implementación de un sistema que nos dosificara la descarga de semillas, por lo anterior, como se observa en la *figura 13* se hace el montaje sobre el robot de dos plataformas en paralelo unidas por un hilo giratorio que en su centro lleva una esfera con un orificio en uno de sus extremos que es por donde saldrán las semillas, además se presentan las medidas de dicho mecanismo.

Figura 13. Mecanismo de carga y descarga de semillas

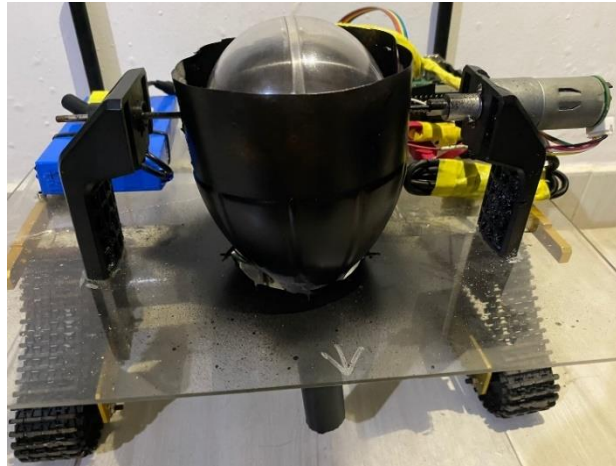


Fuente: imagen propia

El sistema de descarga de semillas dosificaba lo suficiente para evitar grandes cantidades de caída [24]. El sistema entra en actividad cada cierta distancia según lectura de los encoders, en donde se energiza el motor, proporcionando giro y por ley de fuerza de gravedad es como las semillas se descargan.

Fue necesario la incorporación de una protección externa en forma de cono para mejorar el riego de la semilla y evitar dispersiones mayores, así como se observa en la siguiente figura.

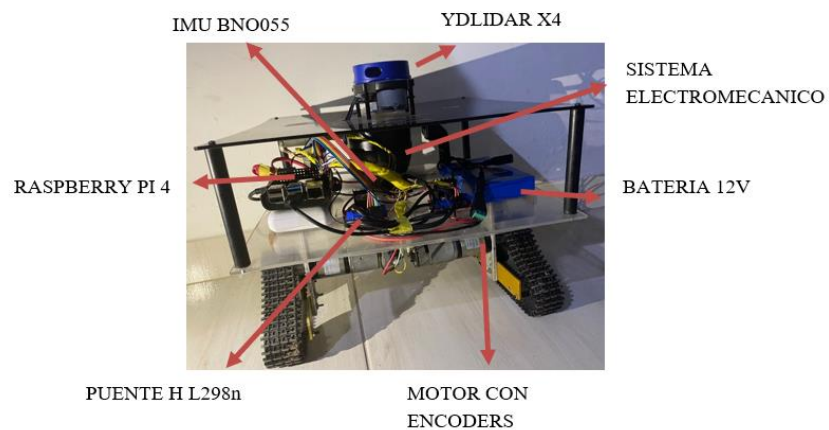
Figura 14. Sistema electromecánico final.



Fuente: imagen propia

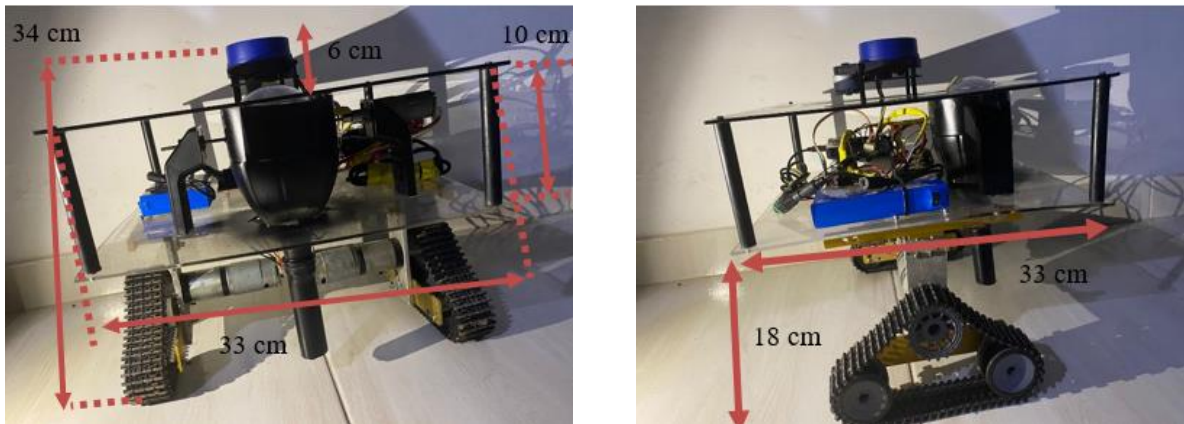
Una vez se identificaron los elementos y dispositivos que se requerían para el proyecto, se llevó a cabo su distribución y montaje sobre la estructura, por lo tanto, en la *figura 15*, se muestran los principales elementos y dispositivos instalados en el chasis, y en la *figura 16*, se observan las dimensiones del mismo.

Figura 15. Robot sowwall.



Fuente: Imagen propia

Figura 16. Vista frontal y lateral del robot sowwall



Fuente: Imagenes propias

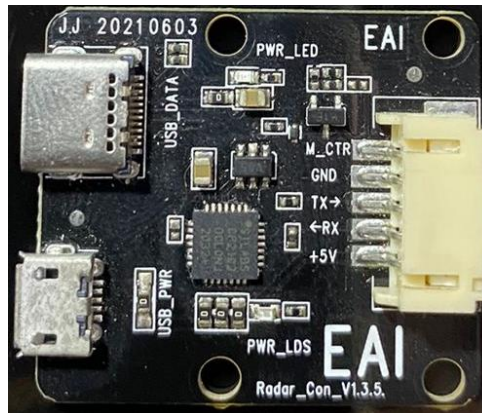
5.2 FASE 2: INSTALACION DE DISPOSITIVOS

Después de la identificación de los diferentes componentes que se usaran en el siguiente proyecto, se prosigue con la instalación de los sensores, YDLIDAR X4, IMUBNO055, y el enconder incremental, todo en conjunto con la placa de desarrollo Raspberry pi 4B.

6.2.1 Sensor YDLIDAR X4

El sensor YDLIDAR X4 utiliza el protocolo de comunicación serial UART 3.3V. con tecnología TTL, para llevar a cabo la conexión con la placa Raspberry se hace uso del adaptador USB de EAI (*figura 17*), que es el encargado de proporcionar la fuente de alimentación para el sensor y de convertir la interfaz serial UART interna de YDLIDAR X4 en una interfaz USB [22].

Figura 17. Adaptador USB.



Fuente: (YDLIDAR/Focus on Lidar Sensor Solutions, n.d.)

Una vez conectado el sensor a la placa Raspberry pi 4, se prosigue a identificar el nombre del puerto al que fue conectado y descargamos un paquete de configuración para el sensor YDLIDAR X4 que se puede encontrar en el siguiente dirección <https://github.com/YDLIDAR/sdk/tree/master>. El sensor laser requiere de una configuración rápida para empezar el proceso de captura de datos, la cual se describe en el anexo 1. Luego de realizar los pasos anteriores se esta listo para empezar la captura de datos, para corroborar una buena obtencion de datos se lleva la realizacion de 5 pruebas con diferentes muestras de medidad observadas en la *tabla 6*.

Tabla 6. Pruebas del sensor YDLIDAR X4

Dato Real	Prueba1	Prueba2	Prueba3	Prueba4	Prueba 5
20 cm	19.7	19.7	19.6	19.4	19.6
30 cm	29.1	29.5	29.1	29.1	29.1
40 cm	39.5	39.5	39.5	39.5	39.5
50 cm	49.5	49.5	49.5	49.5	49.5
60 cm	59.6	59.6	59.6	59.6	59.6
70 cm	69.5	69.5	69.5	69.5	69.5
80 cm	79.3	79.4	79.3	79.5	79.5

Fuente: Tabla propia

Después de tomar las 5 pruebas con muestras diferentes, se procede a obtener el error absoluto para cada una de las muestras para tener una idea más clara del comportamiento del sensor. El error absoluto se calcula teniendo el valor real medido menos el valor medido por el sensor.

$$\text{Error absoluto} = \text{valor real} - \text{valor medido} \quad (1)$$

Tabla 7. Errores absolutos de las medidas del sensor YDLIDAR X4.

	prueba1	prueba2	prueba3	prueba4	prueba 5
	0,30	0,3	0,4	0,6	0,4
	0,9	0,5	0,9	0,9	0,9
	0,5	0,5	0,5	0,5	0,5
	0,5	0,5	0,5	0,5	0,5
	0,4	0,4	0,4	0,4	0,4
	0,5	0,5	0,5	0,5	0,5
	0,7	0,6	0,7	0,5	0,5
Promedio error	0,54	0,5	0,6	0,6	0,5
Promedio total	0,54				

Fuente: Tabla propia

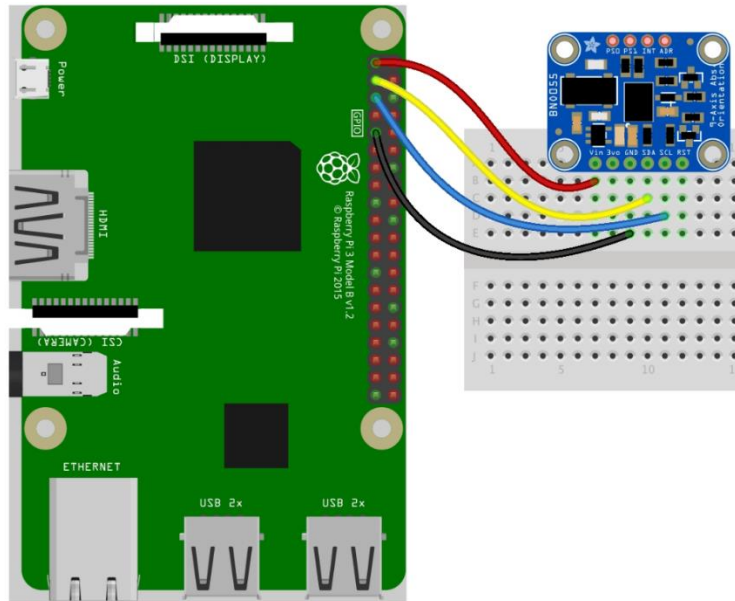
Posteriormente de la toma de muestras y del cálculo del error se tiene una estimación de cuanto es el error de distancia que presenta el lidar y se toma en cuenta para un mejor diseño de precisión en los algoritmos.

5.2.2 SENSOR INERCIAL IMU BNO055

Para realizar la conexión de la IMU BNO055 se realiza por medio del modo I2C que se debe habilitar directamente en la Raspberry pi 4 como lo indica (Anón s/f-g), este método solo es posible con Raspberry pi 4 o superiores debido a que la Raspberry 3 al usar el método

I2C presenta un error de estiramiento de reloj I2C y se debería hacer mediante el modo UART.

Figura 18. Conexión Raspberry pi 4 y IMUBNO055.



Fuente: (Python & CircuitPython | Adafruit BNO055 Absolute Orientation Sensor | Adafruit Learning System, n.d.)

En la *figura 18* se observa la conexión realizada entre la placa de desarrollo Raspberry pi 4 y el sensor inercial.

Adafruit cuenta con un paquete de instalación y de configuración para el sensor BNO055 por modo de I2C que se puede encontrar en <https://docs.circuitpython.org/projects/bno055/en/latest/index.html> y se puede seguir en el anexo 2.

Al tener instalados todos los requerimientos necesarios para el uso del sensor BNO055, se hace uso de la librería de Adafruit para la adquisición de datos, más exactamente de solo la

orientación absoluta presentado por el eje z, ya que es lo requerido para este proyecto y por lo mismo en la siguiente tabla se presentan diferentes muestras tomadas de los valores arrojados por la IMU, esto para conocer un poco la precisión otorgada por este sensor.

Tabla 8. Pruebas del sensor IMU BNO055.

Angulo real	Angulo Imu	Angulo imu2	Angulo imu3	Angulo imu4	Angulo imu5
0°	-0,1°	0,04°	0,2°	0,02°	-0,19°
10°	13,8°	11,1°	10,2°	9,7°	12,8°
20°	19,5°	21,06°	20,2°	22,5°	21,1°
30°	29,3°	30,43°	31,5°	30,4°	30,2°
40°	39,5°	41,2°	39,9°	43,04°	40,5°
50°	48,9°	51,3°	51,3°	51,7°	52,07°
60°	59,8°	61,6°	61,3°	61,4°	62,02°

Fuente: Tabla propia

Usando la misma ecuación 1, de cómo se encontró el error absoluto para el sensor YDLIDAR X4, se hace uso para identificar el error en cada una de las muestras tomadas por el sensor IMU BNO055 que se presentan en la siguiente tabla.

Tabla 9. Errores absolutos de los datos de la IMU BNO055.

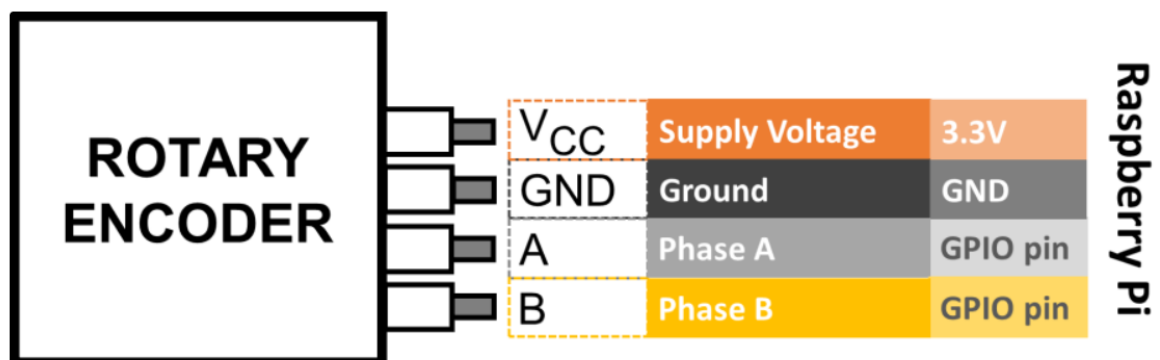
	prueba1	prueba2	prueba3	prueba4	prueba 5
	0,10	0,0	-0,2	0,0	0,2
	-3,8	-1,1	-0,2	0,3	-2,8
	0,5	-1,1	-0,2	-2,5	-1,1
	0,7	-0,4	-1,5	-0,4	-0,2
	0,5	-1,2	0,1	-3,0	-0,5
	1,1	-1,3	-1,3	-1,7	-2,1
	0,2	-1,6	-1,3	-1,4	-2,0
Promedio error	- 0,10	-1,0	-0,7	-1,3	-1,2
Promedio total	0,84°				

Fuente: Tabla propia

5.2.3 ENCODERS DEL MOTOR

Para la conexión de los encoders de cuadratura a la Raspberry se hace usando una fuente alterna de 5V y usando la librería de Rotary encoder para la configuración de los mismos, así como se indica en https://gpiozero.readthedocs.io/en/stable/api_input.html#rotaryencoder, en la cual se describe un poco más sobre su configuración y en la figura que se presenta a continuación se puede observar un esquema del encoder y sus salidas.

Figura 19. Esquema encoder.



Fuente: Codificador Con Raspberry Pi – Things DAQ, n.d.)

Al tener instalados los requerimientos para el funcionamiento de los encoders, procedemos a hacer conteo de los pulsos del motor y tomar muestras de diferentes distancias, según los pulsos del motor, en la siguiente tabla se puede observar los datos tomados.

Tabla 10. Pruebas tomadas con los encoders.

Dato Real	Prueba1	Prueba2	Prueba3	Prueba4	Prueba 5
4 cm	6	3,5	3,5	6	3
9 cm	10	8,5	9,5	10	10,8
13 cm	14	14,5	13,5	14	14.5
18 cm	19	19	17	19	19
20 cm	21	21	20	22	22
28 cm	30	29	29	30	29
33 cm	34	34	34	35	36

Fuente: tabla propia

Luego de tomar las muestras se hace uso de la ecuación (1) del error absoluto para conocer el error en cada una de las muestras tomadas y así conocer un poco más la precisión de los encoders. Esta información se presenta en la siguiente tabla.

Tabla 11. Errores absolutos del encoder.

	Prueba1	Prueba2	Prueba3	Prueba4	Prueba 5
	- 2,00	0,5	0,5	-2,0	1,0
	-1,0	0,5	-0,5	-1,0	-1,8
	-1,0	-1,5	-0,5	-1,0	-1,5
	-1	-1,0	1,0	-1,0	-1,0
	-1	-1,0	0,0	-2,0	-2,0
	-2	-1,0	-1,0	-2,0	-1,0
	-1,0	-1,0	-1,0	-2,0	-3,0
Promedio error	- 1,29	-0,6	-0,2	-1,6	-1,3
Promedio total	1,01 cm				

Fuente: tabla propia

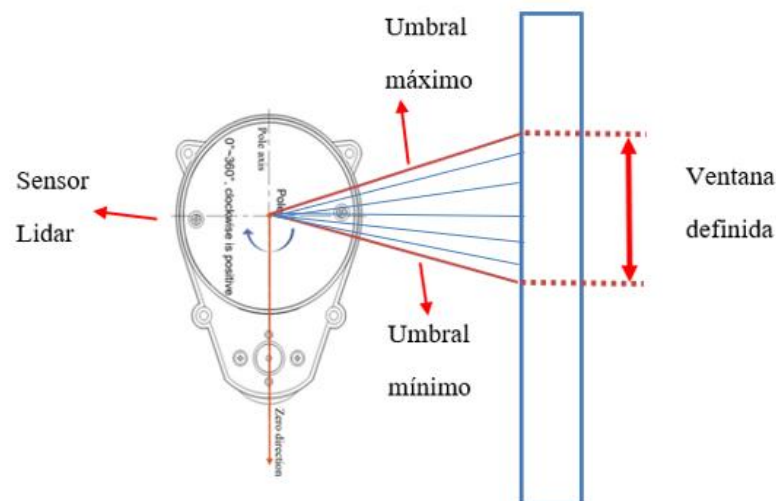
5.3 FASE 3: IMPLEMENTAR UN SISTEMA DE CONTROL DE NAVEGACION

5.3.1 PERCEPCION DEL TELAR

Teniendo en cuenta que los datos arrojados por el LIDAR corresponden a la detección de objetos a su alrededor, el sensor nos retorna un valor de distancia y de grados en un rango de 0° a 360° .

Para la detección del telar se efectúa utilizando un algoritmo en el cual se implementa el concepto de ventana que vendría siendo una pequeña parte delimitada por un umbral máximo y un umbral mínimo, en donde solo se recibirá datos de distancia y de los ángulos de la ventana definida, con los cuales se promediara la distancia del sensor al telar, para tener una mejor medida de los datos que se están recibiendo. Una representación del concepto de ventana se muestra en la siguiente figura.

Figura 20. Ventana angular definida en el LIDAR.

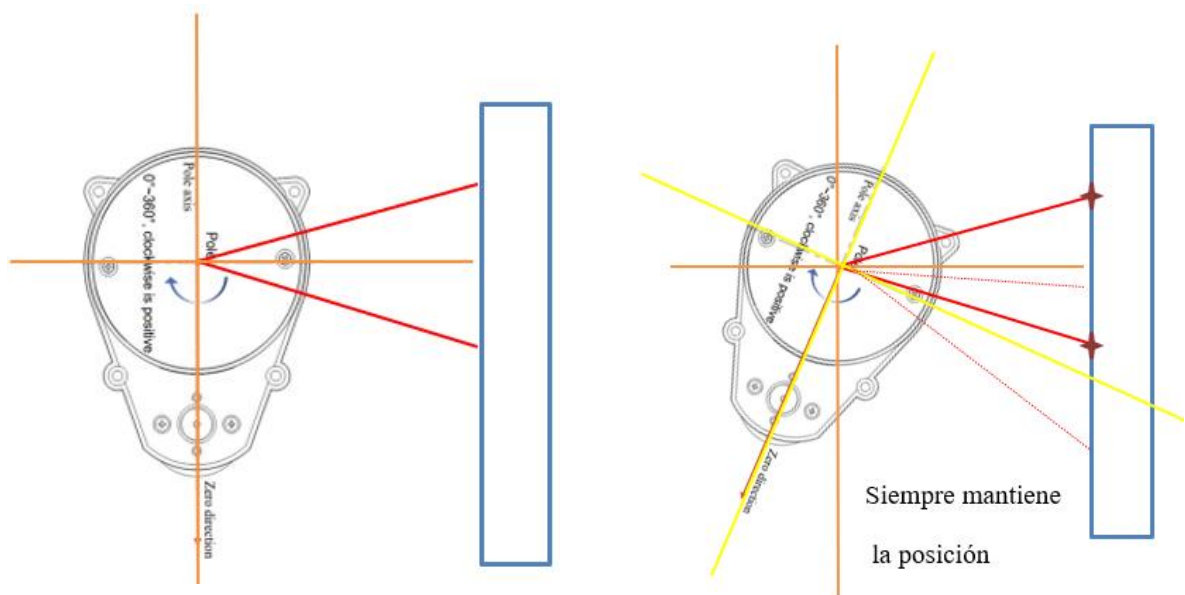


Fuente: Imagen propia

5.3.2 PERCEPCION DEL TELAR CON ERROR DE ORIENTACION

Al tener configurado el LIDAR para que solo envíe datos de la ventana definida, se procede a incorporar el sensor inercial BNO055, para identificar el error de orientación presentado por el robot que se consigue teniendo un punto fijo de referencia y logrando que la ventana definida se mantenga fija sin importar la perturbación del robot. En la presente figura se explica un poco mejor el concepto.

Figura 21. Percepción con error de orientación.



Fuente: imagen propia

El sensor inercial entrega el desplazamiento del robot en grados respecto de su eje central z, en donde de esta forma se integra al algoritmo del sensor LIDAR permitiendo que la ventana de percepción se mantenga siempre en un punto, respecto del error de orientación.

5.4 FASE 5: IMPLEMENTAR UN SISTEMA DE CONTROL DE NAVEGACION

Para la implementación del sistema de control fue necesario el uso de un sistema de control de lazo cerrado, debido a que es necesario que mientras el robot avanza y detecta el telar, se debe retroalimentar el controlador con el error de posición y de orientación, proporcionado por los sensores lidar e imu, evitando también que si el robot sufre alguna perturbación o alguna afectación de su desplazamiento, el controlador este en la capacidad de recobrar su posición y orientación respecto a la trayectoria paralela del telar.

Para tal fin se escoge hacer un controlador de lógica difusa para que el robot se adapte de una manera más sencilla, además de no necesitar un modelo matemático o caracterización de los motores para que funcione adecuadamente como se puede observar en la siguiente tabla se explica un poco mejor el paso a paso para la elaboración de un algoritmo de lógica difusa.

Tabla 12. Pasos del sistemas de control difuzo.

Pasos	Nombre	Descripción
1	Inicialización	Definir variables de entrada y control
2	Fuzificación	Convierta datos de entrada nítidos en conjuntos difusos utilizando la membresía
3	Reglas	Conjunto de reglas de compilación
4	Defuzzificación	Convierta el conjunto difuso a valores de salida nítidos

Fuente: (Norris, Basic FLS, 2017)

5.5 *DEFINIR VARIABLES DE ENTRADA*

Se inicia definiendo las variables que se van a utilizar para el controlador tanto de entrada que viene siendo los datos de orientación y distancia dados por la imu y el lidar, teniendo en cuenta que para las variables de salida se tiene presente la velocidad de los motores del robot para lograr tener control de la posición y de la distancia la que se encuentra el robot del telar.

5.6 *FUZIFICACION*

En la fase de la fuzificacion se inicia creando las funciones de membresía o pertenencia para ello se nombra las diferentes variables lingüísticas utilizadas, y usando el método triangular se prosigue con la elaboración de los conjuntos difusos:

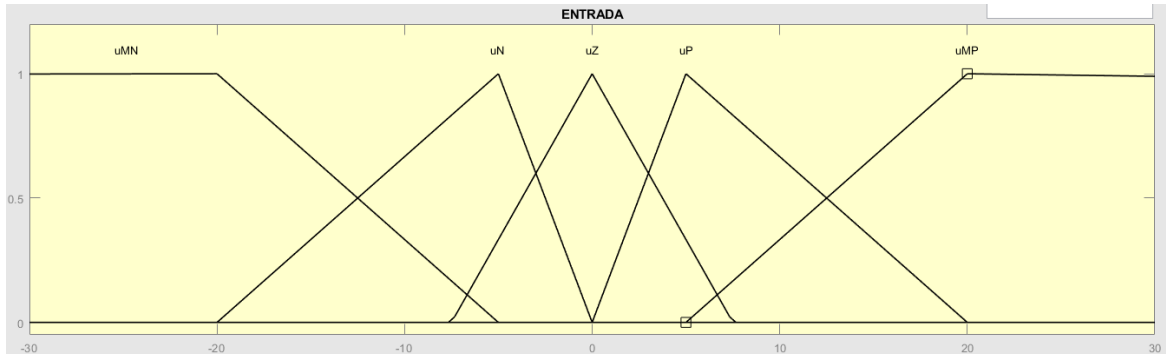
VARIABLES DE ENTRADA:

- uMN: conjunto muy negativo
- uN: conjunto negativo
- uZ: conjunto cero
- uP: conjunto positivo
- ump: conjunto muy positivo

VARIABLES DE SALIDA:

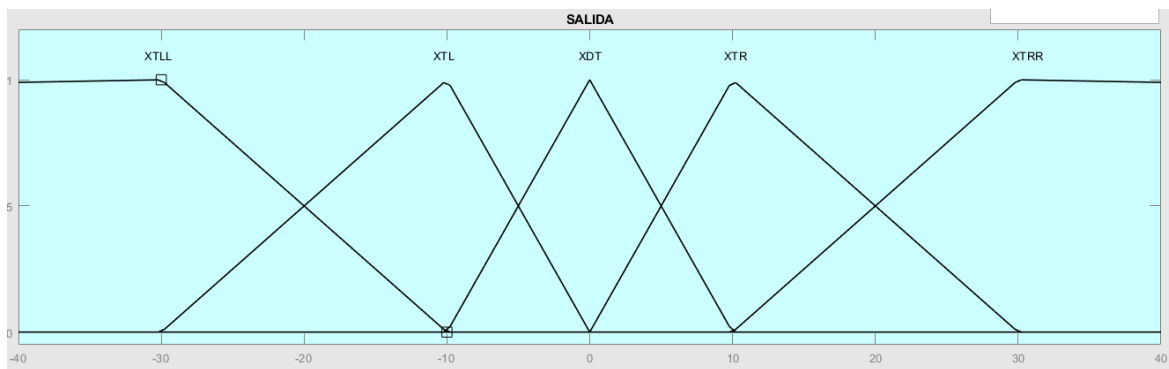
- XTLL
- XTL
- XDT
- XTR
- XTTRR

Figura 22. Conjuntos difusos de entrada.



Fuente: imagen propia

Figura 23. Conjuntos difusos de salida.



Fuente: imagen propia

5.7 REGLAS DIFUSAS

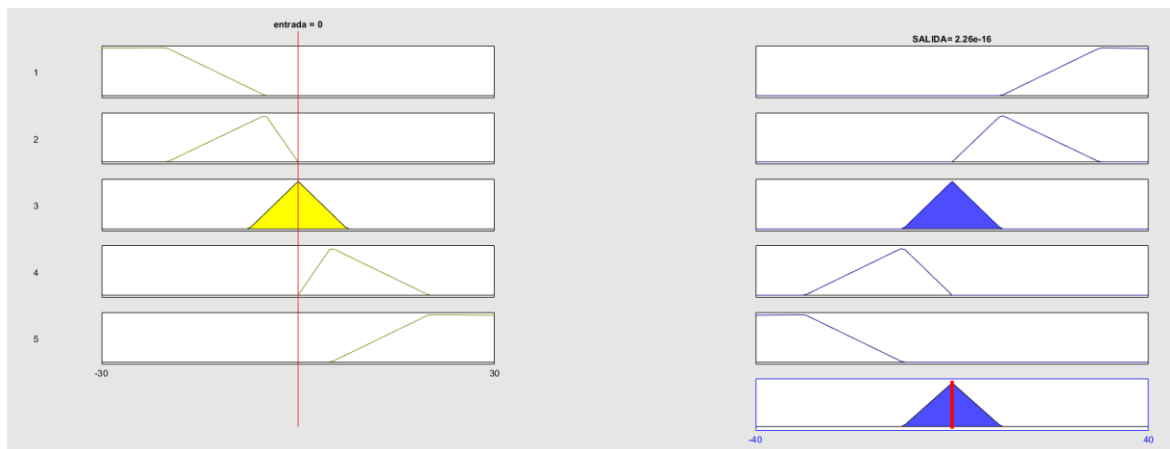
Posteriormente de tener los diferentes conjuntos difusos de entrada y salida procedemos a declarar las reglas difusas las cuáles serán las encargadas de conectar los diferentes conjuntos difusos y quienes pondrán el controlador en funcionamiento.

1. SI error de orientación es uMN ENTONCES la salida es XTLL
2. SI error de orientación es uN ENTONCES la salida es XTL
3. SI error de orientación es uZ ENTONCES la salida es XDT
4. SI error de orientación es uP ENTONCES la salida es XTR
5. SI error de orientación es uMP ENTONCES la salida es XTRR

5.8 DEFUZZIFICACION

La Defuzzificación se realizó usando el método del centroide debido a que es el más utilizado y tiene una gran precisión a la hora de dar un valor de resultado, en la siguiente imagen se presenta una representación de como actuaría el control difuso luego de haberle puesto las reglas.

Figura 24. Control difuso, error de orientación 0° .



Fuente: imagen propia

Figura 25. Control difuso, error de orientación -3° .



Fuente: imagen propia

En la imagen anterior se observa un correcto funcionamiento del control difuso, se observa que si se presenta un disturbio o alguna causa que desequilibra el sistema el inmediatamente nos da una salida que será la que corregirá nuestro sistema, con esto queda claro que nuestro controlador funciona según las reglas que le dimos y tenemos una salida satisfactoria que regula nuestro sistema de control.

6. RESULTADOS

Para analizar el funcionamiento del robot, este se puso a disposición de diferentes pruebas realizadas en distintas condiciones, logrando así obtener resultados de las capacidades del robot, las limitaciones a las se puede someter, la efectividad del controlador, la eficiencia del sistema encargado de soltar las semillas, entre muchos otros aspectos.

Las pruebas realizadas consistieron en tres inicialmente, en la primera se evaluó el comportamiento del robot a diferentes ángulos de inclinación, iniciando con un ángulo de

10 Grados y variándolo progresivamente cada 5. En la segunda prueba se propuso colocar en funcionamiento el robot en un terreno con irregularidades y en un terreno previamente preparado, para así ver su error de lectura y conocer cómo afecta en su precisión el riego de semillas. Durante la última prueba, se sometió el prototipo a realizar recorridos cada dos horas durante todo un día partiendo de las 8 AM hasta las 5PM, analizando así su comportamiento y efectividad del sensor Lidar a la variación de la luz solar durante el día. Todas las pruebas se realizaron colocando el robot a 43 cm de un telar, distancia que mantendrá el robot durante su navegación. El telar representa el punto de referencia que tomara el prototipo su orientación autónoma.

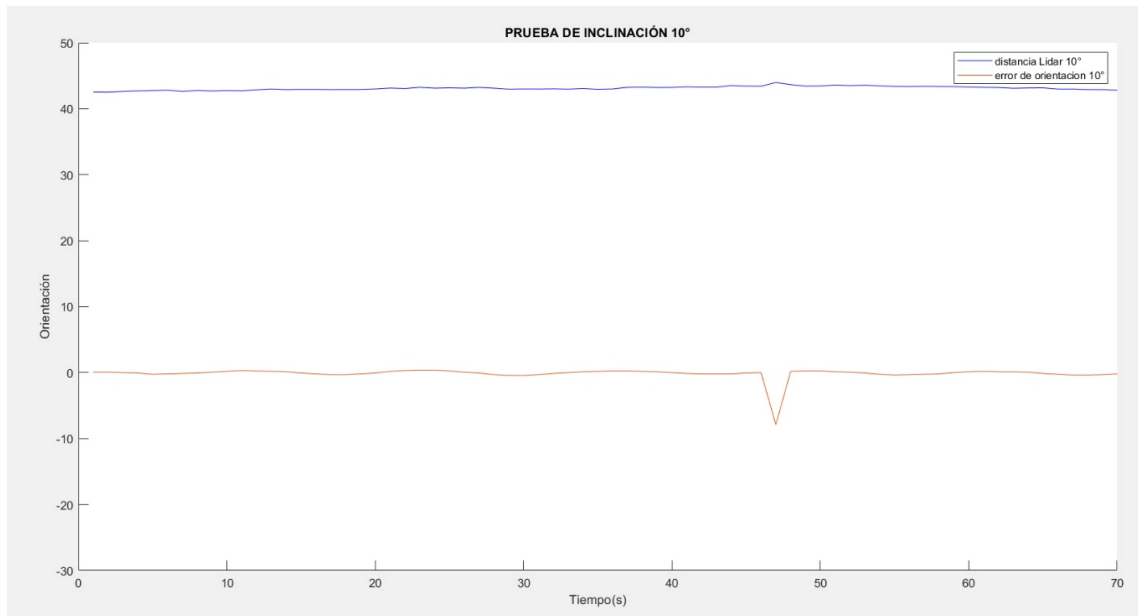
6.1 Prueba N°1. Prueba de Inclinación

Para la realización de esta prueba se usaron materiales caseros que nos permitieron ajustar la inclinación a la cual se pondría a prueba el robot, la inclinación se ajustó con un compás casero y algunas tablas, a las cuales se les modificaba la altura de un lado para conseguir el declive requerido con respecto al suelo.

6.1.1 Prueba de inclinación a 10°

Se inicia sometiendo el robot a una pendiente lo más aproximado a 10°, en esta primera prueba al colocar el robot en funcionamiento a 43 cm del telar, se logra percibir que el comportamiento del mismo es eficiente y no presenta desviaciones exageradas que comprometan el funcionamiento óptimo del prototipo. Los datos adquiridos muestran como el robot conserva una orientación con respecto al telar y su navegación es exacta. En la *figura 26* se puede observar el comportamiento del robot de manera gráfica.

Figura 26. Grafica de los datos tomados en una inclinación de 10°

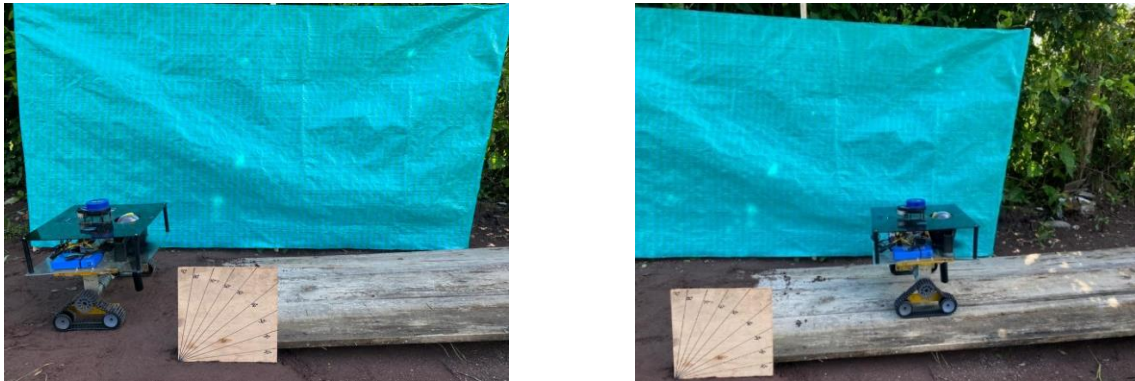


Fuente: imagen propia

Como podemos ver en la gráfica (*figura 26*), la línea superior de color azul, nos muestra los datos de distancia que toma el sensor Lidar con respecto al telar dando a entender que el prototipo mantiene su posición, mientras que en la línea inferior de color café se observa el error de orientación, error que es mínimo durante toda la prueba, excepto en una pequeña instancia de tiempo dada por irregularidad en la superficie en la cual se movía el robot o perturbaciones externas.

En la *figura 27* se observa el robot al momento de realizar la prueba. En la imagen del lado izquierdo podemos ver el robot antes de tomar la inclinación de 10° , y en la imagen del lado derecho se aprecia el robot después de terminar la prueba.

Figura 27. Robot prueba de inclinación a 10°



Fuente: Imágenes propias

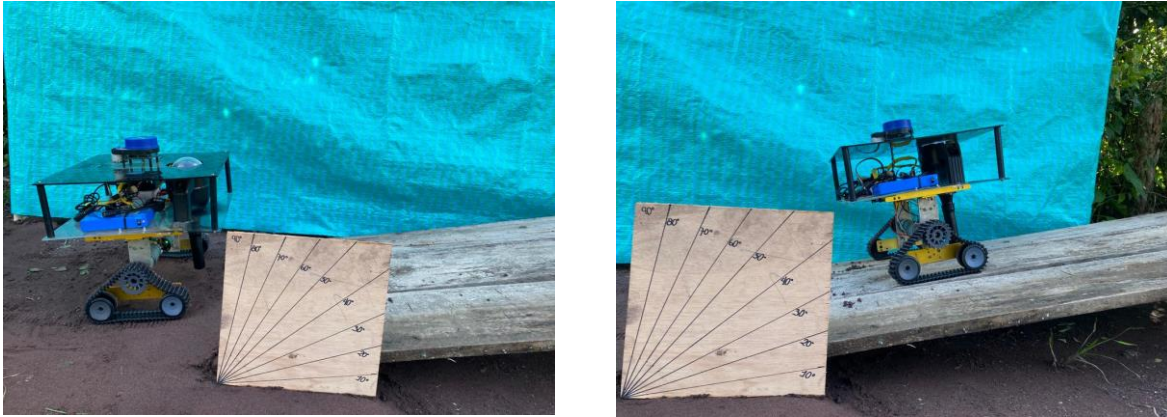
Durante el recorrido, el robot no presentó una desviación fuera de lo común, su funcionamiento fue óptimo y conservó su orientación en línea recta mientras dejó caer semillas, en las imágenes se puede observar que el prototipo no perdió el equilibrio y se mantuvo estable hasta el final de la prueba. La velocidad del robot fue constante, no aumentó ni disminuyó considerablemente.

6.1.2 Prueba de inclinación a 15°

Al igual que en la prueba de 10° de inclinación, tomando como ayuda el compás casero, se inclinaron las tablas hasta obtener 15° lo más exacto posible, una vez se logra la inclinación requerida para el ejercicio, se coloca en funcionamiento el robot desde el inicio para que este tome la pendiente a medida que va avanzando.

En la siguiente figura podemos observar la posición inicial del robot y la posición final del robot después de haber recorrido con éxito el trayecto.

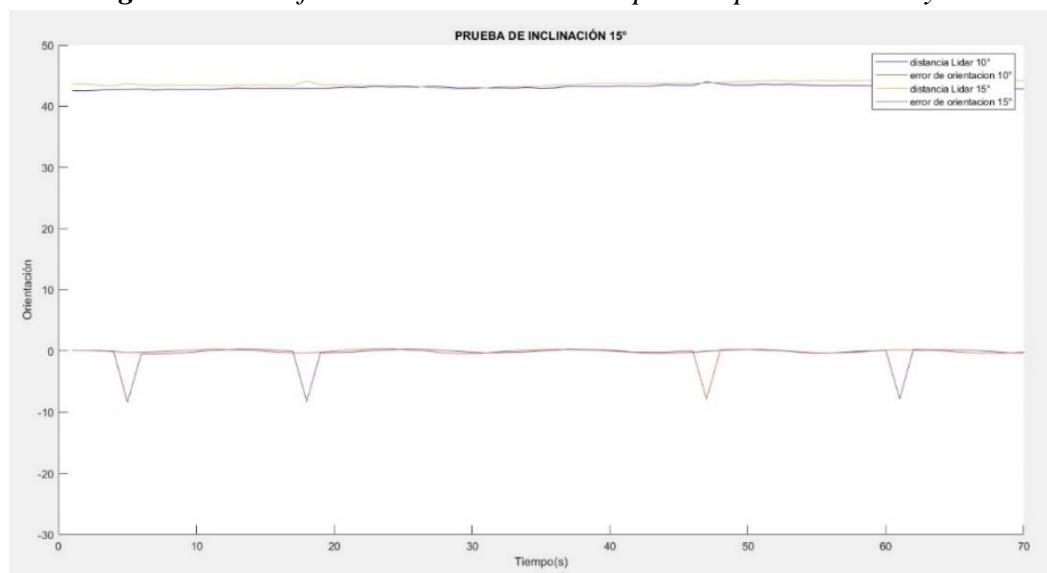
Figura 28. Robot prueba de inclinación a 15°



Fuente: Imágenes propias

Durante la prueba para 15° de inclinación el robot mantuvo su equilibrio y velocidad desde el inicio de la prueba hasta el final, sus motores respondieron correctamente y su sistema electromecánico encargado de soltar las semillas, de igual forma funciono correctamente. la navegación del robot no se vio afectada, y los datos adquiridos por el sensor permitieron orientar correctamente al robot, esto lo podemos observar en la *figura 29* donde nos muestra gráficamente los datos tomados.

Figura 29. Gráfico de los datos tomados para la prueba de 10° y 15°



Fuente: imagen propia

Para la prueba de 15° podemos observar que el robot presento más perturbaciones, lo que aumento las veces en que el error de orientación es mayor, aun así, el desempeño del robot sigue siendo eficiente y no presenta inconvenientes para realizar sus funciones. Las mediciones tomadas por el sensor Lidar son exactas en las dos pruebas, su variación es muy mínima y permanece casi constante, mostrando que la posición del prototipo de mantiene en todo momento.

6.1.3 Prueba de inclinación a 20°

Continuando con las pruebas, se ajusta el declive de las tablas a 20° lo más exacto posible con respecto al suelo y nuevamente se posiciona el robot de tal forma que vuelva a realizar el recorrido. Una vez el robot está en su lugar se da inicio para que este comience su trayectoria y mantenga los 43 cm con respecto al telar que se muestra en la *figura 30*.

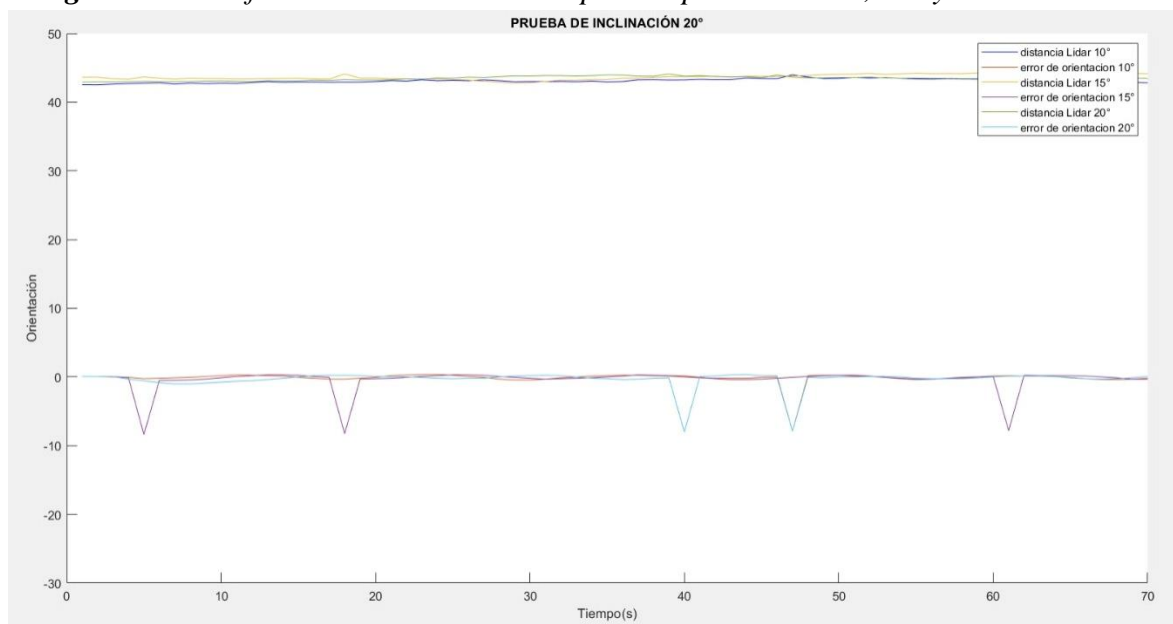
Figura 30. Robot prueba de inclinación a 20°



Fuente: Imagenes propias

Durante la prueba se determinó óptimo el funcionamiento del robot en una superficie con una inclinación de 20° , no presento desviaciones fuera de lo común y mantuvo una estabilidad y velocidad constante durante todo el trayecto de principio a fin. El sistema encargado de soltar las semillas tampoco se vio afectado y funciono correctamente, los datos tomados por el sistema durante la navegación, muestran que el robot conservo su distancia en todo momento, presentando solo unas muy leves variaciones por irregularidades en la superficie de las tablas. El error de orientación fue mínimo, al igual que en la prueba de 10° , solo se presentó una pequeña variación en una instancia de tiempo.

Figura 31. Gráfico de los datos tomados para la prueba de 10° , 15° y 20°



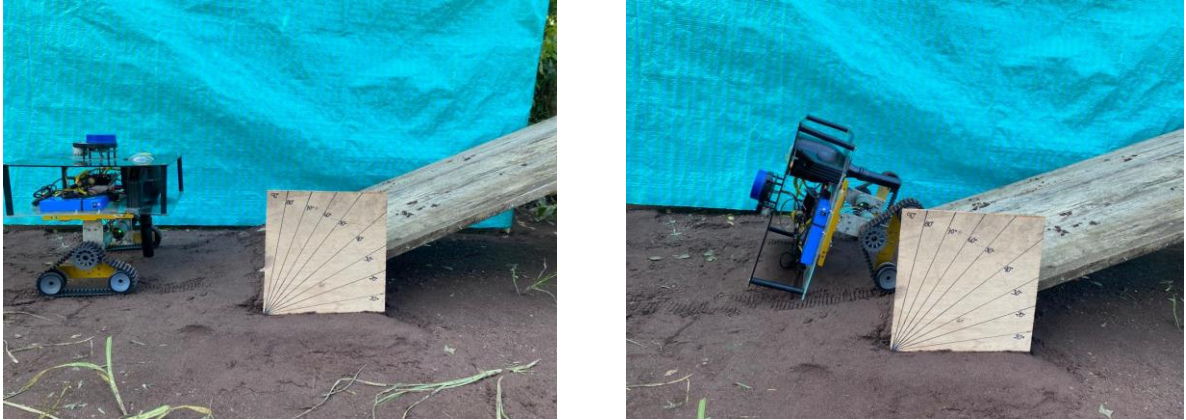
Fuente: imagen propia

6.1.4 Prueba de inclinación a 25°

En la prueba de inclinación para 25° se ajustó nuevamente el ángulo en la estructura, una vez posicionadas con ayuda del compás, se llevó el robot a su posición inicial para

realizar la trayectoria sobre la superficie inclinada. Tomando como referencia el telar, se da inicio para que el prototipo comience su recorrido.

Figura 32. Robot prueba de inclinación a 25°

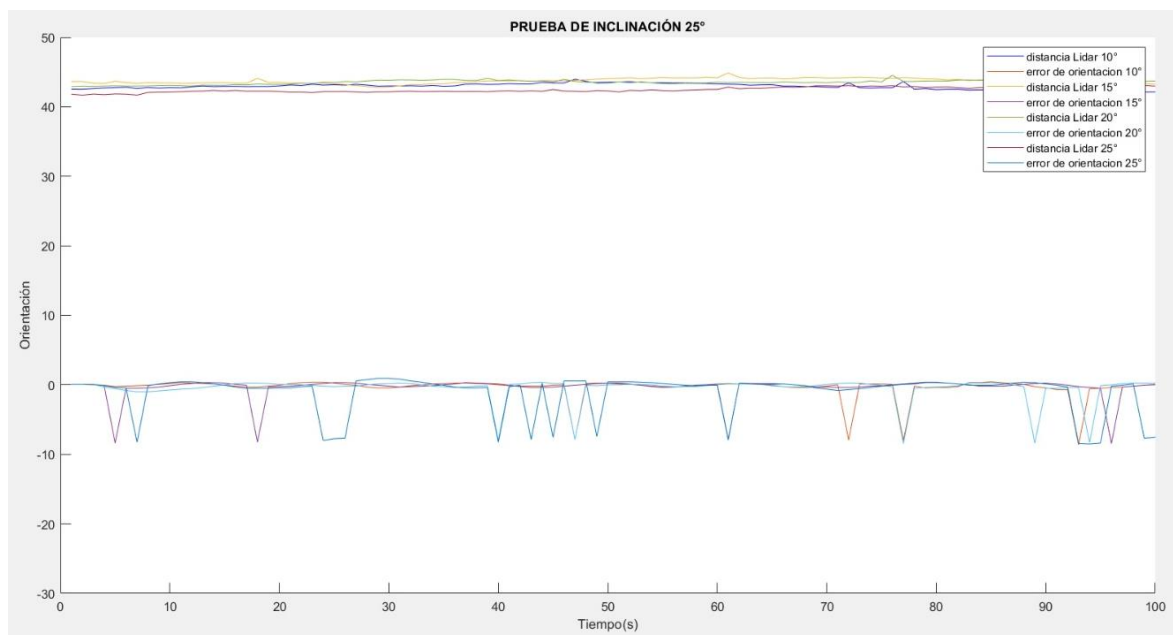


Fuente: imágenes propias

Como podemos observar en la *figura 32*, el robot presento problemas para mantener el equilibrio, las razones principales por el cual el prototipo no tiene la capacidad de navegar en pendientes mayores o iguales a 25° son el peso, la forma cuadrada del robot y el tamaño de las orugas. Durante el ensamblaje se implementaron los componentes más pesados en la parte posterior del prototipo para que el sistema encargado de soltar las semillas contara con espacio en la parte frontal, lo que produjo como consecuencia que el peso en el robot no sea equitativo y no logre navegar sobre una superficie inclinada a 25°. A su vez la estructura del robot y el tamaño de las orugas no son los más óptimos para superficies con desnivel pues su forma cuadrada y sus orugas pequeñas no se diseñaron para este tipo de terrenos inclinados.

Los datos tomados muestra que el robot mantuvo su distancia correctamente con respecto al telar, mientras que el error de orientación fue mínimo hasta el momento donde el robot pierde el equilibrio y cae, en la *figura 33* podemos observar la variación que presento justo en ese instante.

Figura 33. Grafico de los datos tomados para las prueba de 10°, 15°, 20 y 25°



Fuente: imagen propia

6.2 Prueba N°2. Prueba en terreno irregular

Para la realización de esta prueba, se adecuo el terreno sobre el cual el robot realizaría la trayectoria, diseñando una pista de aproximadamente un metro y medio de largo, con algunas inclinaciones y surcos, simulando las irregularidades en la superficie (*Figura 34*). Se realizaron varias pruebas sobre el terreno, para determinar gráficamente como varia el distanciamiento del riego, realizado por el sistema electromecánico implementado en el robot.

Figura 34. Terreno con irregularidades en la superficie



Fuente: imagen propia.

Durante las pruebas sobre este tipo de terreno, el prototipo mantuvo un buen funcionamiento, aspectos como la arena y suciedad no intervinieron en el desempeño del robot.

En las *figuras 35 y 36*, se pueden observar las dimensiones de algunos surcos presentes en el terreno por el cual el prototipo realizo la prueba de campo. Como se muestran en las imágenes algunos de estos vacíos contaban con profundidades de 4 a 6 cm y un largo de 16 a 21 cm, que equivalente a una escala de 1:10 es aproximadamente 0.4 a 0.6 metros de profundidad y 1.6 a 2.1 metros de largo. Estas medidas indican que un modelo del prototipo a mayor escala es funcional para superficies con irregularidades de gran tamaño, convirtiéndose en un gran desarrollo tecnológico para la industria en la agricultura.

Figura 35. Dimensiones de los surcos a lo largo.



Fuente: Imágenes propias

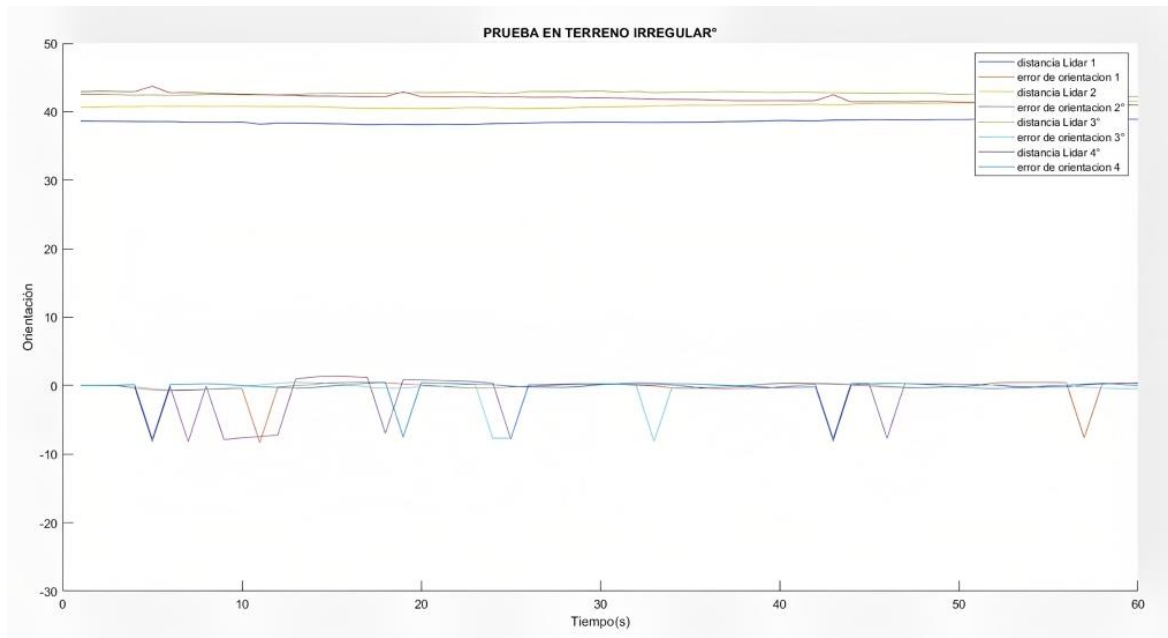
Figura 36. Profundidad de los surcos.



Fuente: imágenes propias

En la *figura 37* se observa los resultados de algunas pruebas tomadas por el prototipo, donde se muestra el comportamiento del robot a medida que avanza por el terreno irregular.

Figura 37. Datos arrojados por pruebas realizadas en un terreno irregular.

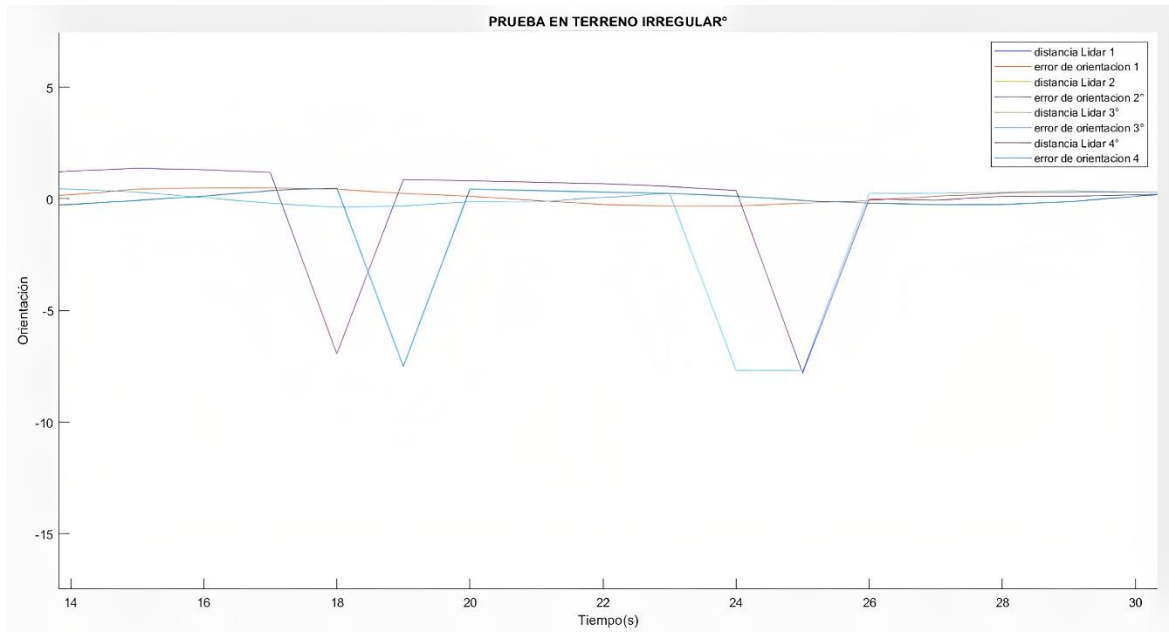


Fuente: imagen propia

El comportamiento del robot según la gráfica y los datos arrojados, muestran que el prototipo aunque presenta algunos errores de orientación en ciertas instancias de tiempo, su distanciamiento con respecto al telar se mantiene, pues como se observa en la parte superior de la imagen la lectura del lidar para cada prueba realizada es constante y no presenta gran variación en el tiempo.

Para apreciar mejor las correcciones que realiza el robot durante su trayectoria podemos ver la *figura 38*, donde realizando un acercamiento a la gráfica de la *figura 37* podemos observar pequeñas ondas que muestran el amortiguamiento del robot para mantener su posición en un solo rumbo, los picos que se muestran hacia abajo, son correcciones rápidas que realiza el robot cuando este salía de un surco ya que el prototipo se desorientaba levemente.

Figura 38. Correcciones realizada por el prototipo en un terreno irregular.

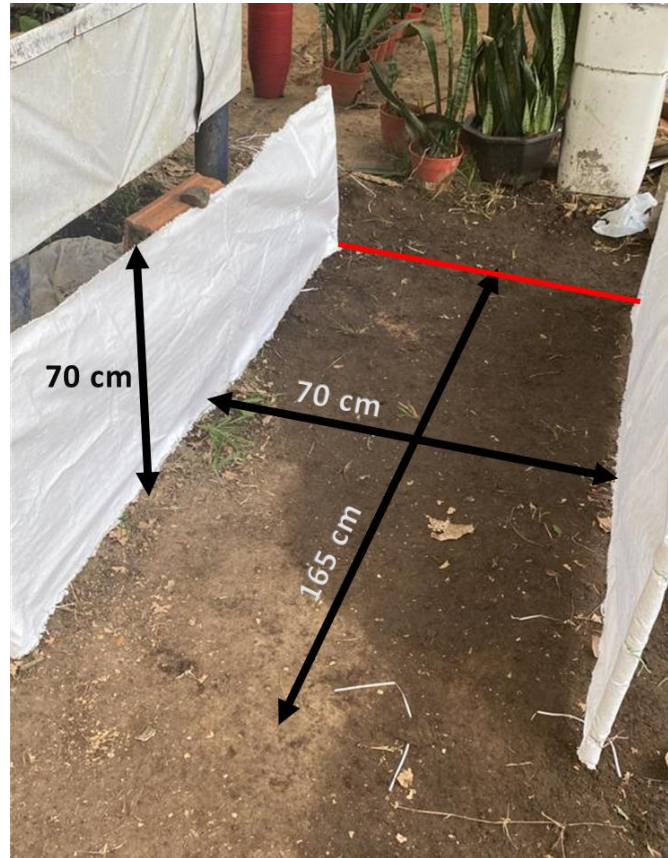


Fuente: imagen propia

6.3 Pruebas en terreno regular.

Como bien se sabe el prototipo está diseñado para un terreno previamente preparado, por lo cual se inicia adecuando el área donde se dispone a colocar a prueba el robot, de tal manera que durante su recorrido presente las mínimas perturbaciones posibles.

Figura 39. Terreno regular adecuado



Fuente: imagen propia

En la *figura 39* se observa el campo ya preparado con las medidas de largo, ancho y alto. El robot iniciará su trayectoria a partir de la línea roja tomando como referencia el telar del lado derecho de la imagen.

Al ser este un terreno previamente preparado todo el circuito que recorre el prototipo está ubicado bajo una polisombra de color negro que reducirá la incidencia de la luz solar para evitar daños en el robot o minimiza posibles interferencias que se puedan presentar.

Durante el desarrollo de las pruebas se realizó un pequeño arado donde las semillas caen como se observa en la *figura 40*.

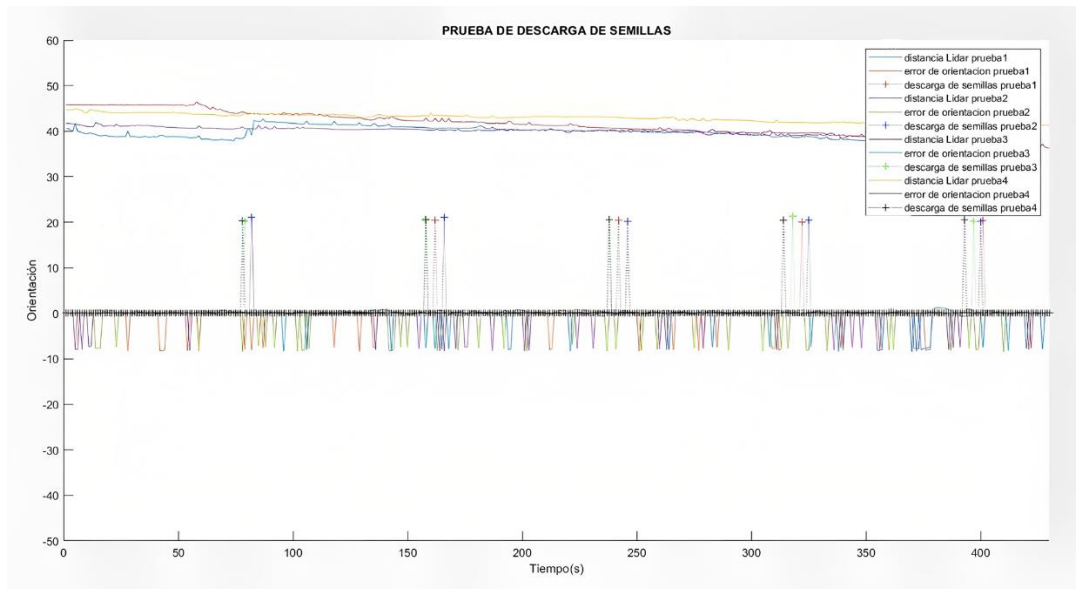
Figura 40. Prueba de campo.



Fuente: imagen propia

Dado que se realizaron múltiples pruebas se toma 4 para mostrar gráficamente las distancias en el momento de regar las semillas durante cada prueba (*Figura 41*). Es importante aclarar que el prototipo nunca presento inconvenientes de lectura por parte del sensor Lidar durante su trayectoria y su error de orientación para cada prueba fue mínimo, manteniendo siempre su dirección en línea recta con pequeñas correcciones en tiempo real.

Figura 41. Datos tomados de las pruebas realizadas en un terreno regular.



Fuente: imagen propia

En la *figura 41* se puede observar de forma gráfica el comportamiento del robot, en esta se muestran los puntos donde el sistema electromecánico diseñado para el riego de las semillas se activa a su vez que el sensor lidar va tomando la distancia del telar y el control fuzzy corrigiendo la trayectoria del prototipo. En la gráfica también se observan variaciones al inicio de la toma de datos por parte del Lidar para cada prueba, estas lecturas del Lidar son debido al posicionamiento del robot en el inicio del terreno siendo un posible error humano, a medida que el robot avanza este se ajusta lentamente a la distancia que se configuró del telar, acción que se refleja en la misma grafica.

Algunas de las imágenes donde se muestran de forma física el esparcimiento de las semillas y la distancia entre grupos, las podemos observar en la *figura 42*

Figura 42. *Esparcimiento de las semillas durante las pruebas*



Fuente: imágenes propias

El distanciamiento entre cada riego de semillas es aproximadamente entre 23 y 26 cm una a la otra, para cada prueba tomada varía levemente y si lo relacionamos con la gráfica de la *figura 41*, podremos observar esas diferencias en donde las semillas caen pero sus distancia entre ellas no concuerdan siempre en un mismo punto, siendo preciso pero no exacto.

Las pruebas tanto para el terreno irregular como el previamente preparado concluyen en que el robot es capaz de funcionar con gran eficiencia, pues su desempeño fue el esperado realizando su función principal que es el riego de semillas y navegar de forma totalmente autónoma. La precisión en la medición del Lidar fue con gran asombro mejor de lo que se esperaba para el terreno irregular, la capacidad de navegar a través de una superficie con huecos bastante profundos es un gran logro que adquiere el robot como alcance extra.

6.4 Prueba N°3. Navegación del robot expuesto a la luz solar durante el día.

Esta prueba consiste en poner en funcionamiento el prototipo al aire libre, donde este expuesto a la luz directa del sol durante diferentes horas del día, para este estudio el robot se sometió a 5 pruebas realizadas cada dos horas, partiendo de las 8Am hasta las 4Pm, la carga de la luz solar se midió en lúmenes usando un luxómetro.

Para adecuar el terreno se uso un telar verde de dos metros de largo el cual tomara el robot como guía para navegar durante las pruebas. La superficie en la que se trabajó fue plana para evitar interferencias externas del suelo en las medidas del Lidar.

6.4.1 Pruebas realizadas a las 8 y 10 Am

Durante las pruebas realizadas para las 8 y 10 de la mañana, el robot estuvo expuesto a 1595lm y 1847lm como se muestra en la siguiente figura:

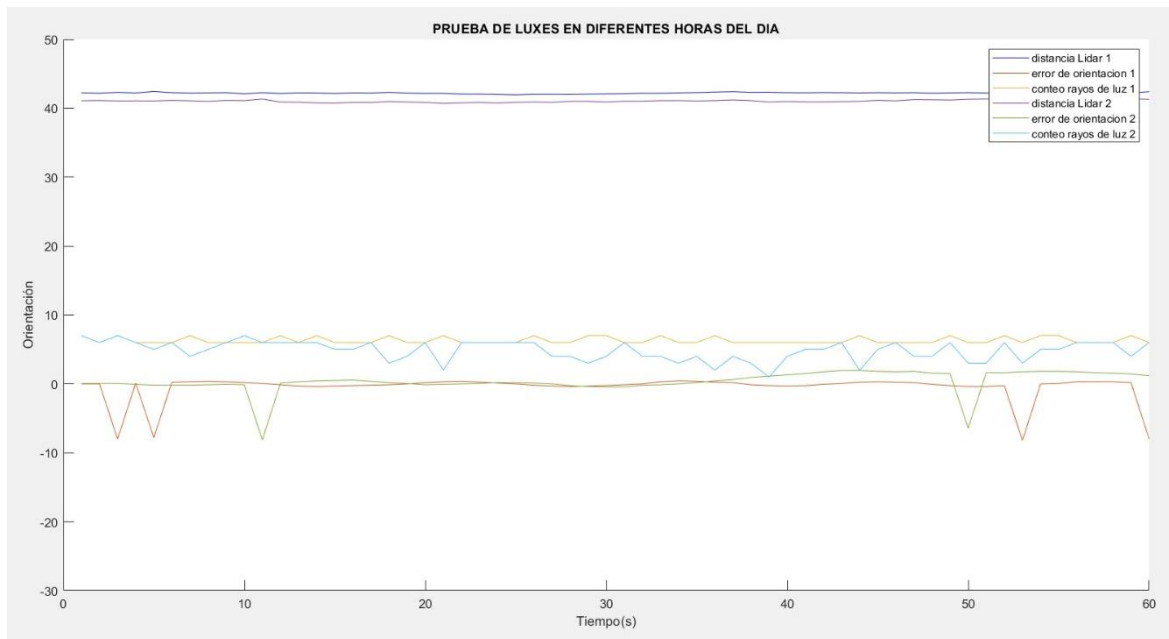
Figura 43. Lúmenes a las 8 y 10 de la mañana.



Fuente: imágenes propias

Los datos arrojados por las dos pruebas (*figura 44*), muestran que el funcionamiento del robot en este rango de horas, es óptimo y eficaz sin ningún inconveniente, la lectura del sensor Lidar no se vio afectada por la incidencia directa de la luz solar, y el error de orientación se mantuvo dentro de lo normal.

Figura 44. Datos arrojados en las pruebas realizadas a las 8 y 10 AM.



Fuente: imagen propia

6.4.2 Pruebas realizadas de 2 a 4 Pm

Para las pruebas realizadas de 2 a 4 Pm, al igual que para las de 8 a 10 Am el robot se sometió a tomar mediciones de distanciamiento y control en el horario de la tarde, exponiéndose de forma directa a la luz solar. que para en esos momentos la luminosidad del sol medida por el luxómetro era de 35900lm a las 2 Pm y 1167lm a las 4 Pm como se muestra en la *figura 45*.

Figura 45. Lúmenes medidos por el luxómetro a las 2 y 4 de la tarde.

2 PM



4 PM

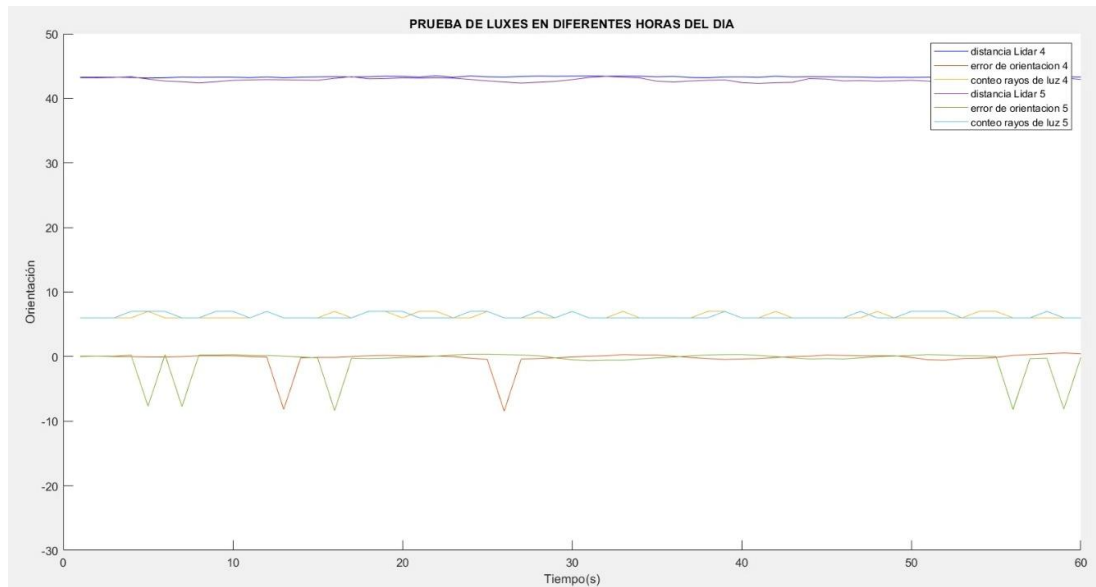


Fuente: imágenes propias

El prototipo durante las pruebas realizadas para estas dos horas del día, no presento inconvenientes con su navegación, el comportamiento del Lidar fue el adecuado y el error de orientación fue mínimo permitiendo que el robot sea funcional a estas horas del día. Los datos adquiridos por los sistemas se muestran gráficamente en la *figura 46*, allí podemos observar que el robot en ningún momento perdió su rumbo y se mantuvo en línea recta todo el tiempo.

Se esperaba que durante la prueba de las 2 Pm el robot presentara inconvenientes con su navegación reflejando desorientación del prototipo por la cantidad de luz solar en el lugar, pero las pruebas y los datos tomados comprobaron lo contrario dando a conocer que el robot si es capaz de navegar estando expuesto a 35900lm.

Figura 46. Datos arrojados en las pruebas realizadas a las 2 y 4 PM.



Fuente: imagen propia

6.5 Prueba realizada a las 12 Pm

Durante la prueba realizada a medio día, la luz solar llego 49900lm (figura 47), punto en el cual fue el mayor registro tomado por el luxómetro para el desarrollo de las practica.

Figura 47. Registro tomado por el luxómetro a las 12 Pm del día.



Fuente: imagen propia

La prueba realizada a las 12 Pm llevo a un gran descubrimiento, al poner en funcionamiento el robot para la toma de datos y que navegara de forma autónoma, este se desorienta y pierde el control. Se realizaron varias pruebas y en todas el prototipo no respondió correctamente, para comprobar que la cantidad de luz presente en el momento era la razón de esta desorientación se puso a prueba el robot en un lugar con algo de sombra y funciono correctamente, dando como resultado que el prototipo no es funcional entre las 11:30 Am y las 2 Pm.

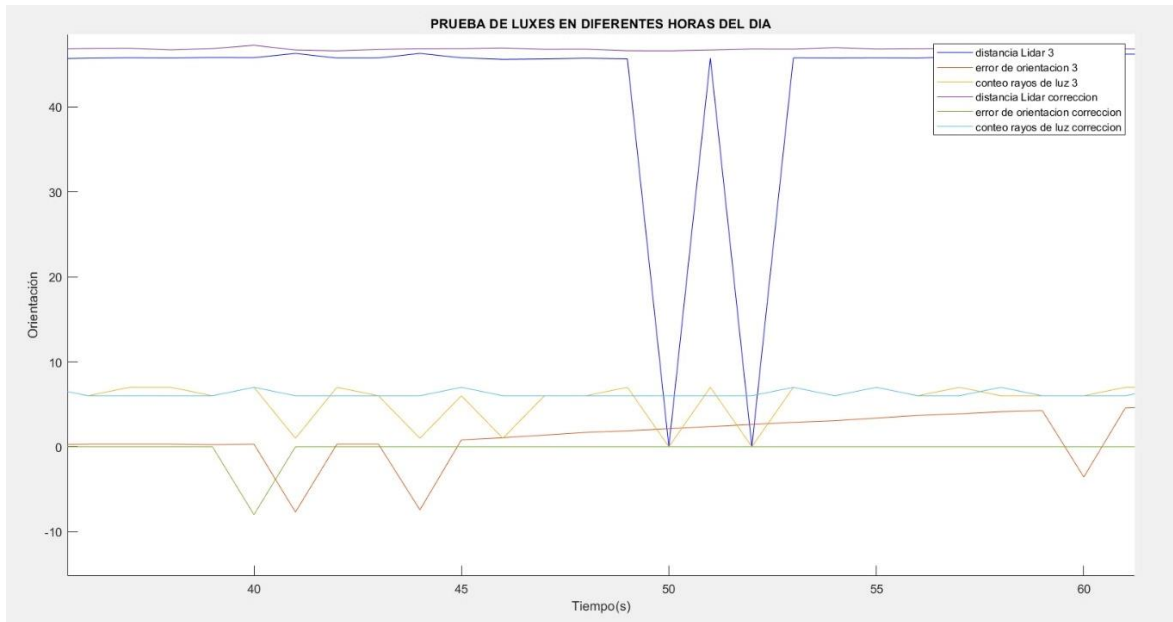
Durante las pruebas, se tomó la decisión de implementar una pequeña cubierta temporalmente que protegiera al sensor Lidar, dejando al descubierto de este solo aquel lado por donde se envía el As de luz infrarrojo que mide la distancia a la que se encuentra el robot del telar (figura 48), se intentó nuevamente realizar la prueba, en buscas de una posible solución a esta desorientación que presenta por culpa de los rayos del sol. dando como resultado que al cubrir el Lidar el prototipo si es capaz de navegar en el horario donde mas luz natural se da en el día.

Figura 48. Robot con la cubierta diseñada para porteger el Lidar



Fuente: Imágenes propias

Figura 49. Datos arrojados para las pruebas realizadas con la cubierta y sin ella.



Fuente: imagen propia

En la gráfica (*figura 49*), si observamos la línea azul, esta se puede asociar con la lectura que realiza el sensor Lidar cuando el prototipo fue expuesto a la luz directa durante la prueba. Como bien se ve, presentó errores en las mediciones bastante significativos dando paso a que el robot se desorientara y perdiera el control. La línea morada que se encuentra en la parte superior, representa las mediciones tomadas por el Lidar una vez se le adapta la cubierta, convirtiéndose en una posible solución a futuro si se planea mejorar el rendimiento del prototipo a cualquier hora del día.

7. Conclusiones

El informe presenta la implementación de un robot, capaz de navegar de forma autónoma y a su vez realizar la descarga de semillas sobre un terreno previamente preparado para cultivo.

Para que el robot realizara la navegación, se elaboró un control difuso que junto a un microcontrolador permitían adquirir la información suministrada por los sensores Lidar e IMU, siendo este capaz de poder hacer una percepción lateral, logrando así un posicionamiento y orientación constante siempre referenciado a un obstáculo que deba seguir.

Para dejar caer las semillas, se implementó un sistema electromecánico, el cual posee un motor que actúa cada cierto tiempo dependiendo de los rpm suministrados al sistema por encoders. Las semillas son almacenadas en una esfera de plástico, que gira y deja caer las semillas cuando el motor entra en funcionamiento una vez llega a los rpm indicados. Para que las semillas al caer no se dispersen, se implementó un conducto casero con materiales reciclados que mejora la precisión y evita que las semillas queden separadas unas a otras al caer.

Los resultados muestran que el prototipo funciona con gran eficiencia en campos de cultivos con inclinación no mayor a 25° , a su vez es eficiente en horarios donde la luz solar es menor como lo pueden ser en las mañanas o en la tarde, los alcances del robot permiten que este pueda navegar en terrenos irregulares, aunque fue diseñado para terrenos previamente preparados.

8. Anexos

8.1 ANEXO 1: CONFIGURACION DEL SENSOR LIDAR

Para tener una buena configuración de nuestro sensor YDLIDAR X4 nos dirigimos al siguiente enlace <https://github.com/YDLIDAR/sdk/tree/master> en el cual encontraremos todo el paquete necesario para la configuración del sensor.

El desarrollo del proyecto se realiza con Python3 por lo que primero es necesario descargar todas las librerías para python3, posteriormente en la raspberry digitamos los comandos.

```
git clone https://github.com/YDLIDAR/sdk.git
cd sdk/build
cmake ..
make
sudo make install
```

al finalizar la instalación en la misma página se encuentra un programa de prueba para verificar que la instalación fue correcta.

8.1 ANEXO 2: CONFIGURACION DEL SENSOR BNO055

En la instalación de los controladores para el sensor inercial es de gran importancia tener el software actualizado, y estar conectado a una red wifi, posteriormente en la terminal digital de la raspberry se inicia ejecutando los siguientes comandos:

1. `sudo apt-get update`
2. `sudo apt-get install -y build-essential python-dev python-smbus`

Posterior a esto se procede a descargar e instalar el paquete de instalación del módulo BNO055.

```
pip3 install adafruit-circuitpython-bno055
sudo pip3 install adafruit-circuitpython-bno055
mkdir project-name && cd project-name
python3 -m venv .venv
source .venv/bin/activate
pip3 install adafruit-circuitpython-bno055
```

Después de la instalación si se presenta algún mensaje de error, se debe volver atrás y revisar que todo se haya instalado correctamente, de lo contrario en el siguiente link se puede encontrar un poco más de informacion

<https://docs.circuitpython.org/projects/bno055/en/latest/index.html>.

9. Referencias Bibliográficas

- [1] Dutta, Sweety & Shanker, Udit & Katiyar, Sulekha & Singh, Venkatesh & Zafar, Mohd & Mohanta, Jagadish. (2019). *Development and Fabrication of an Autonomous Seed Sowing Robot*. IOP Conference Series: Materials Science and Engineering. 691. 012023. 10.1088/1757-899X/691/1/012023.
- [2] *El uso de robots en tareas agrícolas*. (2016). Interempresas. Recuperado el 5 de mayo de 2023, de <https://www.interempresas.net/Horticola/Articulos/151745-El-uso-de-robots-en-tareas-agricolas.html>.
- [3] V. Santhi, N. Kapileswar, V. K. R. Chenchela and C. H. V. S. Prasad, "Sensor and vision based autonomous AGRIBOT for sowing seeds," 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, India, 2017, pp. 242-245, doi: 10.1109/ICECDS.2017.8389873.
- [4] S. M. Khot et al., "Automated Seed Sowing Robot," 2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE), NaviMumbai, India, 2021, pp. 1-4, doi: 10.1109/ICNTE51185.2021.9487573.
- [5] P. V. S. Jayakrishna, M. S. Reddy, N. J. Sai, N. Susheel and K. P. Peeyush, "Autonomous Seed Sowing Agricultural Robot," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, 2018, pp. 2332-2336, doi: 10.1109/ICACCI.2018.8554622.
- [6] *Introduction to inertial sensors · VectorNav.* (2021). <https://www.vectornav.com/resources/inertial-navigation-primer/theory-of-operation/theory-inertial>.
- [7] T. Fadhaeel, P. C. H, A. Al Ahdal, M. Rakhra and D. Singh, "Design and development an Agriculture robot for Seed sowing, Water spray and Fertigation," 2022 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), Greater Noida, India, 2022, pp. 148-153, doi: 10.1109/CISES54857.2022.9844341.

- [8] T. Nagdeve, P. Jangde, H. Tandulkar, S. Dhara, N. Ukani and S. Chakole, "*Design of Automated Seed Sowing Robot for BT Cotton Seed*," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2020, pp. 303-307, doi: 10.1109/ICIRCA48905.2020.9183259.
- [9] Devasia, A. (2023). An Introduction to Types of Closed-loop Motor Control. *Technical Articles*. <https://control.com/technical-articles/in-introduction-to-types-of-closed-loop-motor-control/>
- [10] Agarwal, T. (2020, 21 julio). *Open Loop & Closed Loop Control System and Their Differences*. ElProCus - Electronic Projects for Engineering Students. <https://www.elprocus.com/difference-between-open-loop-closed-loop-control-system/>.
- [11] Paguayo. (2022). ¿Que es Raspberry Pi? - Raspberry Pi. *Raspberry Pi*. <https://raspberrypi.cl/que-es-raspberry/>.
- [12] Agustín Cruz Contreras, E. A. P. F. y R. S. O. (2018). *Multiplicador Electrónico para Encoder Incremental*. Recuperado el 3 de abril de 2023, de chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.redalyc.org/pdf/4026/402640451007.pdf
- [13] *Motor de engranajes planetarios ABHL 36GP 555, codificador Hall, motor de CC de alto Torque, 6V, 12V, 24V, calidad Original/Motor CC/ - AliExpress*. (n.d.). Retrieved April 6, 2023, from <https://n9.cl/6g528>.
- [14] Castellanos-Ruíz, J., Montealegre-Mesa, L. M., Martínez-Toro, B. D., Gallo-Serna, J. J., & Almanza-Fuentes, O. (2020). Uso de sensores inerciales en fisioterapia: Una aproximación a procesos de evaluación del movimiento humano. *Universidad y Salud*, 23(1), 55–63. <https://doi.org/10.22267/rus.212301.214>
- [15] GeoSLAM. (2022, 14 noviembre). *What is LiDAR and How Does it Work?* <https://geoslam.com/what-is-lidar/>.
- [16] Roberto Carlos Guamán Choca, Ángel Eucebio Villava. (2013). *Diseño y construcción de un controlador inalámbrico a través de microprocesadores para el funcionamiento de un automóvil eléctrico a escala, controlado desde un computador*. Recuperado el 3 de abril de 2023, de <https://bibdigital.epn.edu.ec/bitstream/15000/5921/1/CD-4736.pdf>

- [17] Holguín, C., Díaz-Ricardo, Y., & Antonio Becerra-García, R. (2018). *El lenguaje de programación Python/The programming language Python Ivet Challenger-Pérez*. <http://www.linuxjournal.com/article/2959>
- [18] [FrontPage - Python Wiki](https://wiki.python.org/moin/). (2018). <https://wiki.python.org/moin/>.
- [19] Ruvalcaba Coyaso, U., & Javier, F. (2015). *Lógica difusa para la toma de decisiones y la selección de personal*. 17, 239–256. <http://www.redalyc.org/articulo.oa?id=187243745010>
- [20] Hurtado, J. P., Universidad, P., Quindío, D., De Ciencias Básicas, F., Tecnologías, Y., & De, P. (2021). *LÓGICA DIFUSA: PERSPECTIVA Y APLICACIONES*.
- [21] Raspberry Pi - Wikipedia, la enciclopedia libre. (n.d.). Retrieved April 4, 2023, from https://es.wikipedia.org/wiki/Raspberry_Pi.
- [22] *Downloads & Support_YDLIDAR/Focus on lidar sensor solutions*. (n.d.). Retrieved April 4, 2023, from https://www.ydlidar.com/service_support/download.html?gid=5.
- [23] *Overview / Adafruit BNO055 Absolute Orientation Sensor / Adafruit Learning System*. (n.d.). Retrieved April 4, 2023, from <https://learn.adafruit.com/adafruit-bno055-absolute-orientation-sensor/overview>
- [24] *El semillero - Horturbà*. (2018). Retrieved April 6, 2023, from https://www.horturba.com/castellano/cultivar/ficha_manejo.php?ID=16.
- [25] *Python & CircuitPython / Adafruit BNO055 Absolute Orientation Sensor / Adafruit Learning System*. (n.d.). Retrieved April 10, 2023, from <https://learn.adafruit.com/adafruit-bno055-absolute-orientation-sensor/python-circuitpython>,