

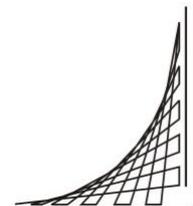
Maestría en Ingeniería Electrónica

**Evaluación de la viabilidad en la aplicación de tecnología
LiDAR para la detección de trayectorias y finales de surco en
un cultivo de papa usando un robot Rover.**

Autor

Manuel Enrique Wagner Mendivelso

Bogotá, 07-07-2018



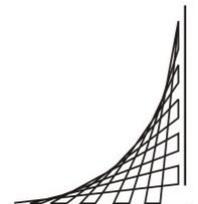
**Evaluación de la viabilidad en la aplicación de tecnología
LiDAR para la detección de trayectorias y finales de surco en
un cultivo de papa usando un robot Rover.**

**Tesis para optar al título de magíster en Ingeniería
Electrónica, con énfasis en automatización.**

Director

PhD. Javier Alberto Chaparro Preciado

Bogotá, 07-07-2018



Nota de aceptación:

La tesis de maestría titulada “**Evaluación de la viabilidad en la aplicación de tecnología LiDAR para la detección de trayectorias y finales de surco en un cultivo de papa usando un robot Rover**”, presentada por Manuel Enrique Wagner Mendivelso, cumple con los requisitos establecidos para optar al título de Magíster en Ingeniería Electrónica con énfasis en Automatización.

Firma del jurado

Firma del jurado

Firma del director de tesis

Bogotá, Julio de 2018

DEDICATORIA

A mí querida esposa Marina, a mis hijos Danna y David, y a mi bella madre María; quienes siempre me han apoyado y han sido la motivación en cada proyecto que he emprendido.

AGRADECIMIENTOS

Agradezco a Dios primeramente por la vida y la oportunidad que me brindó de continuar mis estudios, a todos y cada uno de quienes aportaron un granito de arena para lograr que este proyecto se hiciera realidad.

Al Ingeniero Javier Chaparro, director de mi proyecto de grado, quien estuvo siempre asesorándome y facilitándome las herramientas para lograr finalizar con éxito el trabajo planteado. Así mismo, al Ingeniero Alexánder Pérez, por compartir sus conocimientos y ser un buen guía en cada oportunidad de diálogo que tuvimos.

A todos los docentes de la Escuela Colombiana de Ingeniería que me brindaron, de la mejor manera, sus conocimientos, dejando una huella imborrable para el resto de mi vida.

A cada uno de mis compañeros, con quienes compartí muchas experiencias y con los cuales superamos trabajos y actividades que me enseñaron, además, el valor del trabajo en equipo.

Finalmente, agradezco a mi familia y amigos, a quienes tuve que dejar de lado en varias ocasiones, para dedicar tiempo al estudio de esta maestría, pero que, aun así, me apoyaron incondicionalmente y fueron el mejor aliciente para culminar con éxito mis estudios.

RESUMEN

La Escuela Colombiana de Ingeniería – Julio Garavito, en su Decanatura de Ingeniería Electrónica, cuenta con un grupo de investigación que impulsa la aplicación de la tecnología en la agricultura de precisión, uno de sus proyectos, en el cual se enmarca este trabajo es el de “Metodología para la caracterización y modelado matemático automático de áreas de cultivos de papa”. Como parte de los proyectos asociados a este grupo, se encuentra la creación de Modelos de Digitalización Terrestre (MDT), a lo cual se pretende aportar a través del presente trabajo de grado. Las actividades se encaminan a realizar el diseño y construcción de un sistema de captura de datos provenientes de un sensor láser (LiDAR, *Light detection and Ranging*), el cual se encuentra instalado sobre un robot diferencial y cuyo funcionamiento se basa en el software *Robot Operating System* (ROS). Una vez controlado el movimiento del Robot, se obtienen una serie de paquetes de datos provenientes del sensor láser, los cuáles son analizados y utilizados para realizar un algoritmo que detecte los surcos y finales de surco en un cultivo de papa. Los datos de ángulo de la señal láser, así como los de distancia de reflexión y tiempo, son los elegidos para determinar, de manera lógica, los surcos y finales de surco dentro del cultivo.

TABLA DE CONTENIDO

DEDICATORIA	4
AGRADECIMIENTOS	5
RESUMEN	6
TABLA DE CONTENIDO	7
LISTA DE FIGURAS	9
LISTA DE TABLAS	10
1. CAPÍTULO UNO. PRELIMINARES	11
1.1 INTRODUCCIÓN.....	11
1.2 DESCRIPCIÓN DEL PROYECTO.....	12
1.2.1 Antecedentes del problema	12
1.2.2 Descripción del problema.....	16
1.2.3 Justificación	17
1.3 OBJETIVOS DEL PROYECTO	17
1.3.1 Objetivo general.....	17
1.3.2 Objetivos específicos	17
2. CAPÍTULO DOS. MARCO TEÓRICO	18
2.1 AGRICULTURA DE PRECISIÓN	18
2.2 TECNOLOGÍA LiDAR	19
2.3 SISTEMA DE MAPEO MÓVIL.....	20
2.4 SENSOR LÁSER.....	21
2.5 ROS (Robot Operating System).....	22
2.6 TARJETA DE PROCESAMIENTO	23
3. CAPÍTULO TRES. PROCEDIMIENTO	24
3.1 DISEÑO DE HARDWARE	25
3.1.1 Robot Rover 6x6.....	25
3.1.2 Sensor laser RPLiDAR 360	26
3.1.3 Tarjeta NVIDIA Jetson TK1.....	27
3.1.4 Control de servos.....	28
3.1.5 IMU BNO055	30
3.2 DISEÑO DE SOFTWARE	31
3.2.3 Configuración de la tarjeta embebida.....	32
3.2.4 Instalación del Sistema Operativo para Robots ROS.....	33
3.2.5 Espacio de trabajo.....	34

3.2.6	Comunicación entre máquinas.....	34
3.2.8	Configuración del sensor LiDAR.....	34
3.2.9	Configuración del control de servos.....	35
3.2.10	Teleoperación.....	35
3.2.11	Configuración del IMU.....	35
3.2.12	Generación de la carpeta para archivo de datos – rosbag.....	36
3.2.13	Construcción de mapas.....	37
3.3	PRESUPUESTO.....	37
4.	CAPÍTULO CUATRO. PRUEBAS DE CAMPO.....	39
4.1	DISEÑO EXPERIMENTAL.....	39
4.1.1	Tipo de investigación.....	39
4.1.2	Fuentes de información.....	39
4.1.3	Técnicas de recolección de información.....	39
4.1.4	Análisis de datos LiDAR.....	41
5.	CAPÍTULO CINCO. CONCLUSIONES Y RECOMENDACIONES.....	48
5.1	CONCLUSIONES.....	48
5.2	RECOMENDACIONES.....	48
	REFERENCIAS BIBLIOGRÁFICAS.....	49
	ANEXOS.....	51
	Anexo 1: Pasos para la instalación de ROS.....	51
	Anexo 2. Configuración del espacio de trabajo.....	52
	Anexo 3. Comunicación entre máquinas.....	53
	Anexo 4. Configuración del sensor LiDAR.....	55
	Anexo 5. Configuración del control de servos.....	56
	Anexo 6. Configuración del IMU.....	57
	Anexo 7. Configuración de la carpeta para archivo de datos – rosbag.....	58
	Anexo 8. Clasificación de datos en formato LAS.....	60
	Anexo 9. Datos – Pruebas de campo.....	62
	Anexo 11. Gráficas en Excel de los datos LiDAR.....	64

LISTA DE FIGURAS

Figura 1. Crecimiento poblacional mundial.	12
Figura 2. Porcentaje de tierras cultivables en el mundo.	13
Figura 3. Método del barbecho mediante el uso de caballos.	14
Figura 4. Sembradora neumática.	14
Figura 5. Funcionamiento de la tecnología LiDAR.	20
Figura 6. Mapeo móvil.	21
Figura 7. Funcionamiento sensor laser.	21
Figura 8. ROS, utilizado para localización del robot.	23
Figura 9. Robot Rover de 6x6.	25
Figura 10. Funcionamiento del sensor láser RPLiDAR 360.	26
Figura 11. Tarjeta NVIDIA Jetson TK1.	27
Figura 12. Servo controlador PWM Adafruit PCA9685.	29
Figura 13. Conexión Jetson TK1 y PCA9685.	29
Figura 14. Jetson Tk1, puerto I2C.	29
Figura 15. IMU placa Adafruit BNO055.	31
Figura 16. Conexión Jetson TK1 y BNO055.	31
Figura 17. Configuración de la pila de navegación del robot.	32
Figura 18. Conexión de dispositivos externos a la tarjeta NVIDIA para la configuración inicial.	33
Figura 19. Ventana de RViz en la que visualizan los datos del láser por medio de puntos.	35
Figura 20. Simulación IMU.	36
Figura 21. Robot Rover en el invernadero de la Escuela Colombiana de Ingeniería.	40
Figura 22. Movimiento del vehículo a través de los surcos del invernadero controlado por el joystick.	40
Figura 23. Nube de puntos generada por el LiDAR en uno de los surcos del invernadero.	41
Figura 24. Nube de puntos que representan los surcos del invernadero.	42
Figura 25. Nube de puntos que identifica parte de la pared del invernadero. ...	42
Figura 26. Variables a analizar en la nube de puntos.	43
Figura 27. Datos obtenidos del LiDAR expresados en formato .csv.	44
Figura 28. Datos del LiDAR en formato excel.	44
Figura 29. Ángulo del haz de luz del sensor LiDAR.	46
Figura 30. Respuesta del sistema al detectar el final del surco.	46
Figura 31. Diagrama de flujo de la activación de los motores de velocidad y dirección del Rover mediante el Joystick.	47
Figura 32. Diagrama de flujo para la detección de surcos y finales de surco. ...	47
Figura 24. Creación del bag ROS.	58
Figura 25. Metadatos del bag creado.	59
Figura 39. Prueba de campo - Luminosidad en el invernadero.	62
Figura 40. Bus I2C J3A1 2x25 pines - Jetson TK1.	63
Figura 41. Bus I2C J3A2 3x25 pines - Jetson TK1.	63

LISTA DE TABLAS

Tabla 1. Características tarjetas de procesamiento.....	24
Tabla 2. Presupuesto del proyecto.	38
Tabla 3. Códigos de clasificación: formato LAS 1.1 a 1.4.....	60

1. CAPÍTULO UNO. PRELIMINARES

1.1 INTRODUCCIÓN

El presente trabajo pretende evaluar la implementación de un sistema LiDAR terrestre para la detección de surcos y finales de surcos en un cultivo de papa. El trabajo se plantea como base fundamental para el proyecto de investigación en agricultura de precisión que viene desarrollando la Escuela Colombiana de Ingeniería Julio Garavito en la ciudad de Bogotá, y cuyo objetivo es aportar conocimiento y avances tecnológicos en el área agrícola del territorio colombiano.

El proyecto se desarrolla en tres fases, las cuales parten del diseño de hardware y software para finalmente realizar pruebas de campo, en las que se incluyen la captura y posterior análisis de la información. Inicialmente se utilizan fuentes secundarias, tales como, textos y páginas web, que orientan la construcción del sistema LiDAR, para luego, continuar con la utilización del sistema en un cultivo de papa, obteniendo datos primarios directamente del sensor, y que finalmente, luego de ser analizados y evaluados, brindan conclusiones acerca de la viabilidad de implementar el sistema LiDAR y obtener datos confiables para la detección de surcos y finales de surco en un cultivo de papa.

Aunque en la actualidad se han implementado sistemas LiDAR para diversas aplicaciones, en su mayoría para levantamientos topográficos, basados en sensores *outdoor*, este proyecto busca identificar los datos que se obtienen de un sensor láser *indoor*, aprovechando la cercanía que tiene el mismo con los surcos de un cultivo. Los altos costos actuales en los sensores LiDAR *outdoor* limitan su implementación en proyectos de investigación, por lo que se evalúa el presente proyecto con un sensor *indoor*.

La estructura del trabajo consta de cinco capítulos. En el primero se presentan los antecedentes históricos del problema, su planteamiento y justificación, además, los objetivos y la metodología que se usa para el desarrollo del proyecto. El segundo capítulo relaciona la definición de algunos términos importantes para la comprensión del trabajo junto con los principales avances y aplicaciones hasta el día de hoy en los sistemas LiDAR. El capítulo tres describe los procedimientos realizados a nivel de software y hardware para la construcción del sistema LiDAR. En el capítulo cuatro se presentan los datos obtenidos en las pruebas de campo, junto con su respectivo análisis.

Finalmente, el capítulo cinco brinda las conclusiones y recomendaciones producto del proyecto y pruebas de campo.

1.2 DESCRIPCIÓN DEL PROYECTO

1.2.1 Antecedentes del problema

El hombre desde la antigüedad ha basado su subsistencia en la caza de animales y el cultivo de alimentos, manteniendo un balance entre la producción y el consumo de los mismos. Sin embargo, en los últimos 60 años, el mundo ha experimentado un crecimiento poblacional vertiginoso, pasando de 3 mil millones de personas en el año 1960 a 7 mil trescientos millones en el año 2015, según datos del banco mundial (Banco mundial, 2016a), figura 1, lo que ha obligado a un cambio en el estilo de vida, especialmente en las grandes ciudades del mundo, y a la generación de alternativas para la producción agrícola.

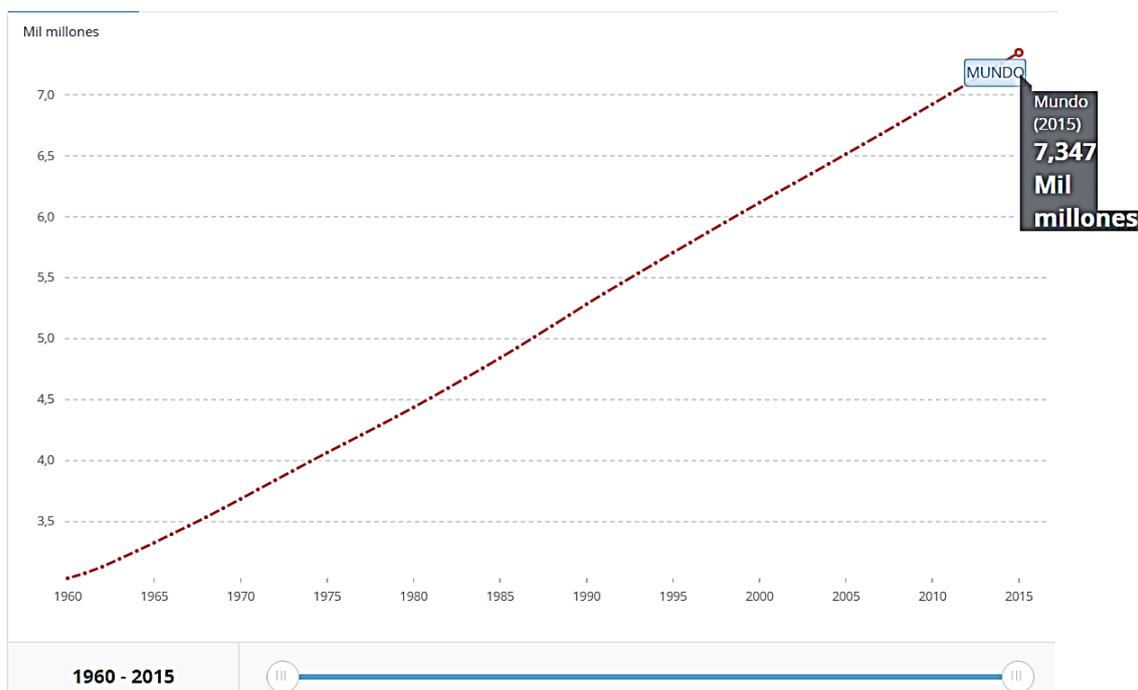


Figura 1. Crecimiento poblacional mundial.
Fuente: Banco mundial (2016).

Este crecimiento poblacional ha traído también muchos cambios culturales y una desaceleración y estancamiento notorio en el porcentaje de tierras cultivadas en el mundo, debido a que muchos de estos espacios, óptimos para el cultivo de alimentos, han sido utilizados para la construcción de viviendas, empresas y grandes centros comerciales. Prueba de lo anterior es el informe presentado por la Organización de las

Naciones Unidas para la agricultura y la alimentación en el año 2016, figura 2 (Banco mundial, 2016b), en el cual se observa que entre los años de 1960 y 1990 se obtuvo un crecimiento de alrededor de 1.2% de tierras cultivadas en el mundo, sin embargo, desde 1990 hasta la actualidad se ha mantenido el mismo porcentaje, de alrededor de 10.8%. Aunque en la actualidad no se siente una fuerte escases de alimentos, si es notoria la disminución de ciertos productos y un leve aumento de precios. Lo anterior sumado a los fuertes cambios climáticos que se vienen presentando, hacen temer por la dificultad de brindar alimentación adecuada a las generaciones futuras.

La disminución de tierras cultivables en el mundo ha llevado a analizar la manera en que se desarrollan las actividades agrícolas, métodos y prácticas utilizadas para la producción de alimentos.



Figura 2. Porcentaje de tierras cultivables en el mundo.
Fuente: Banco mundial (2016).

La disminución de tierras cultivables en el mundo ha llevado a analizar la manera en que se desarrollan las actividades agrícolas, métodos y prácticas utilizadas para la producción de alimentos.

La agricultura es uno de los modelos económicos más antiguos, pero paradójicamente el desarrollo tecnológico no ha sido el mejor. Las prácticas desarrolladas para la producción de alimentos han sido realizadas en gran medida de manera empírica,

tomando el campo de manera uniforme y con necesidades similares en cada espacio. La aplicación de abonos y químicos se hace por conocimiento general del terreno estimando lo que se cree es lo indicado para toda la plantación.

Entre los años de 1875 y 1975, y debido al incremento poblacional, la agricultura pasó a utilizar maquinaria para aumentar su producción. En este lapso de tiempo, también se incrementó el uso de fertilizantes para mejorar los cultivos, los cuales ya se usaban desde el año 1820,(Carmona, 2004); se realizaba rotación de cultivos y era común el uso de abono natural. Los procedimientos realizados provenían de la observación de la naturaleza y la aplicación de técnicas exitosas en otros cultivos similares.

A partir de 1910, la maquinaria, basada en tractores (figura 4), incursionó las grandes granjas agrícolas, logrando tareas como plantación y cosecha. Esta tecnología permitió el aumento en la producción agrícola y la reducción en la utilización de animales para las tareas del campo (Figura 3). (Moss, Andrew Schmitz, 2015)



Figura 3. Método del barbecho mediante el uso de caballos.
Fuente: La Enciclopedia de Saskatchewan



Figura 4. Sembradora neumática.
Fuente: Central Header Parts

Los tractores siguieron evolucionando durante el siglo XX, originando máquinas para la

realización de tareas específicas como por ejemplo la cosechadora de caña de azúcar. A finales del siglo XX, los desarrollos en el área de la agricultura se habían dado en la generación de especies más resistentes a cambios climáticos y plagas, productos con mayores dimensiones y alteración de sus sabores. Sin embargo, los desarrollos en maquinaria y herramientas no eran tan notorios. Fueron apareciendo entonces algunas técnicas y tecnologías que se incorporaron paulatinamente a los procesos del campo, inicialmente a manera de investigación para luego ser implementada satisfactoriamente a los cultivos. El objetivo primordial de estas técnicas y tecnologías es el mejoramiento de los procesos, que hasta el momento eran en su mayoría empíricos. Estos nuevos avances en la producción agrícola mundial originaron lo que se denomina Agricultura de precisión, AP (Espinosa, 2015).

Existen dos técnicas asociadas a la agricultura de precisión. La primera consiste en la generación de imágenes satelitales o aéreas para el estudio del suelo y la segunda en la utilización de sensores en vehículos terrestres para el muestreo y pruebas de laboratorio del suelo (Barreiro, 2007). Precisamente, asociada a esta última técnica, investigadores de los EEUU iniciaron con pruebas de sensores instalados en maquinaria agrícola complementados con monitores de rendimiento, sistemas GPS y sistemas de Información Geográfica SIG (Se trata de un sistema que permite mostrar complejas vistas de variables en las parcelas y tomar decisiones para realizar los cultivos (IDAE, 2010)); con el objetivo de realizar mapas de rendimiento que brinden datos que permitan la toma de decisiones en los procesos agropecuarios (Valero, 2002).

En España, diversos grupos de investigación han comenzado a adaptar las tecnologías de AP provenientes de EE.UU., Australia, Alemania o Reino Unido; tales como sistemas de posicionamiento GPS, sensores electrónicos, software de corrección diferencial de GPS, monitores electrónicos y diferentes programas informáticos para obtener mapas de rendimiento o para combinar los datos de fertilidad del suelo con los de humedad e infestación de malas hierbas y generar mapas de aplicación fitosanitarios (Valero Ubierna, 2004).

Latinoamérica no ha quedado atrás en el cambio que se ha iniciado en los procesos agrícolas, observando que la aplicación de ciertos avances tecnológicos generará mejoramiento en la calidad de los productos, aprovechamiento de los recursos y estandarización de los procesos; sin embargo, es claro que se debe dar un cambio de ideas, costumbres, cultura, debido a que los agricultores realizan muchas de sus actividades de manera empírica. La evolución, recomposición y modernización de la

agricultura permitirá delinear la historia de los países de la región (Perfetti, Balcázar, Hernández, & Leibovich, 2013). La producción agrícola debe tomar un nuevo camino, el cual se enfoca al uso de nuevas tecnologías de la información y la comunicación para el mejoramiento de los resultados y toma de decisiones. (Rodríguez, Rodríguez, & Sotomayor, 2014).

En Chile, el gobierno, a través del ministerio de agricultura, ha venido trabajando de la mano con instituciones académicas teniendo como meta ser un país con potencia alimentaria y forestal; para ello cuentan con los recursos más avanzados en tecnología de información disponibles, como los sistemas de posicionamiento global (GPS), los sistemas de información geográfica (SIG), los sistemas de control y adquisición de datos, sensores y actuadores, entre otros, están cada vez más presentes en el campo (Best & Zamora, 2008).

En Colombia, la industria azucarera Riopaila Castilla ha aplicado la agricultura de precisión y cumplido a cabalidad sus objetivos: optimizar el insumo fertilizante, obtener beneficios ambientales, lograr ahorros económicos en insumo y aumentar la productividad (Castilla, 2013). Los incrementos en la productividad física de campo se pretenden realizar a través de la renovación de cultivos, introducción de variedades promisorias, cambio tecnológico (agricultura de precisión) y prácticas de manejo (Uniandes. Castilla, 2014).

La investigación en Colombia en el campo de la Agricultura de precisión también ha dado frutos e incursionado con fuerza para proponer soluciones y mejoras al agro, tal es el caso de la Escuela Colombiana de Ingeniería Julio Garavito, en cuya institución se generó un proyecto de grado denominado “Módulo de Visión Artificial para una Aplicación de Agricultura de Precisión”. El caso de estudio en el que se aplica este proyecto es una solución de automatización de agricultura de precisión que consta de un robot controlado a través de Plexil, el cual corre en una tarjeta SoC (System on a chip) Intel Galileo (Martínez & Álvarez, 2016).

1.2.2 Descripción del problema

Buscando el mejoramiento en la producción agrícola, al pasar de un proceso netamente empírico a uno apoyado en tecnología y bases de datos, que lleve a procesos de cultivo más eficientes; el presente trabajo busca diseñar, probar y evaluar un sistema basado en la tecnología LiDAR para detección de trayectorias y finales de surco en zonas de

cultivo de papa para un vehículo terrestre (Robot Rover). Lo anterior, enmarcado dentro del proyecto de investigación “**Caracterización de áreas de cultivo mediante sensor RPLiDAR 360° láser scanner y robot Rover**” de la Escuela Colombiana de Ingeniería Julio Garavito, que busca facilitar el desplazamiento autónomo a través de surcos en un campo agrícola del robot Rover, con el fin de capturar datos y realizar mapeo de la zona de cultivo, obteniendo así información importante que permita la posterior implementación de mejoras en actividades de siembra, manejo de plagas y control de enfermedades.

1.2.3 Justificación

Para el proceso de mapeo de zonas de cultivo, se han implementado algunos sistemas basados en la adquisición de señales a través de GPS (Sistemas de posicionamiento global), SIG (Sistemas de información geográfica) o sensores instalados en maquinaria pesada como cosechadoras, los cuales brindan información sobre el estado general del cultivo, sin embargo, no se cuenta con un robot móvil de dimensiones pequeñas, que realice ésta actividad, entre otras, facilitando su uso en los diferentes tipos de terreno con que cuenta el territorio colombiano, apoyado por tecnología LiDAR.

1.3 OBJETIVOS DEL PROYECTO

1.3.1 Objetivo general

Evaluar, mediante pruebas de campo, la viabilidad del uso de tecnología LIDAR como sistema de detección de trayectorias y finales de surco en zonas de cultivo de papa para un vehículo terrestre (Robot Rover).

1.3.2 Objetivos específicos

- Obtener el estado del arte de los sistemas de tecnología LiDAR y sus aplicaciones, haciendo énfasis en las tareas realizadas en el campo agrícola.
- Implementar los dispositivos de hardware y software necesarios para el funcionamiento del sistema LiDAR sobre el robot Rover.
- Diseñar y desarrollar el experimento para la realización de pruebas de campo y evaluación del sistema LiDAR en la detección de trayectorias y finales de surco.

- LiDAR el sistema de tecnología LiDAR propuesto, basado en los datos obtenidos en las pruebas de campo.

2. CAPÍTULO DOS. MARCO TEÓRICO

A continuación, se encuentra la definición de algunos conceptos propios del proyecto y la descripción de las principales características de elementos de hardware y software, de los que se hará uso en el presente proyecto.

2.1 AGRICULTURA DE PRECISIÓN

La agricultura de precisión (AP) se basa en las técnicas implementadas en los sistemas de producción agrícola, con el objetivo de mejorar la calidad de los productos, reduciendo a la vez el gasto de fertilizantes, plaguicidas y agua. Estas técnicas brindan precisión para aplicar la cantidad correcta de insumos, en el momento adecuado y en el lugar exacto; para ello hace uso de la tecnología de la información en la adecuación y el manejo de suelos y cultivo, a la variabilidad presente dentro de un terreno. La agricultura de precisión involucra el uso de sistemas de posicionamiento global (GPS) y de otros medios electrónicos para obtener datos de cultivo (Marote, 2010).

La revista *Ambienta*, de España, clasifica las técnicas de agricultura de precisión en dos grandes grupos:

1. Técnicas orientadas a la gestión de la variabilidad espacial de las propiedades de los suelos y estado de los cultivos.
2. Técnicas orientadas a la ayuda, al guiado y a la uniformidad de las operaciones mecanizadas.(Vega, 2013)

El primero, teniendo en cuenta que los espacios en los cultivos no son homogéneos, sino que varían significativamente en relación a su salinidad, humedad, dureza y composición; lo que lleva a diseñar técnicas para diferentes partes dentro del cultivo. El segundo encaminado al uso de equipos y maquinaria de campo, buscando uniformidad en las tareas mecánicas dependiendo de las necesidades de la tierra y cultivos en general.

Una de las actividades importante en un sistema de agricultura de precisión es el de mapeo. Los mapas de rendimiento se obtienen a partir de los datos recolectados por una cosechadora, que incluye un sistema de posicionamiento global. Un mapa de rendimiento es una representación espacial de datos de rendimiento registrados durante la cosecha de un cultivo. La representación espacial se realiza utilizando un formato vectorial de puntos y coordenadas (x, y, z) asociadas a un sistema de proyección (latitud y longitud, WGS84) que referencia geográficamente cada uno de los puntos. (Cooperativo, Tecnol, & Sur, 2014). Otro de los mapas relevantes en la agricultura de precisión es el de fertilización, el cual permite la aplicación variable de abonos y sustancias orgánicas e inorgánicas, que mejoran la calidad del cultivo, en lotes con variabilidad del suelo. (González, Sepúlveda, & Lorenzo, 2011)

2.2 TECNOLOGÍA LiDAR

Un sistema LiDAR está basado en la emisión de pulsos de luz láser desde una plataforma aérea o terrestre, tal como se observa en la Figura 5. La medición precisa del tiempo de retorno de las posiciones del pulso al sensor permite calcular la distancia que separa a éste de la superficie terrestre y de los objetos que existen sobre ella.

Dado que la posición y orientación del sensor son conocidas para cada pulso emitido, cada señal de retorno tiene unas coordenadas tridimensionales únicas, lo cual permite la captura remota de la información topográfica. La tecnología LiDAR se basa en el uso de sensores activos, por lo que la captura de información no depende de las condiciones meteorológicas, y además las misiones pueden realizarse durante la noche (Magdaleno & Martínez, 2006).

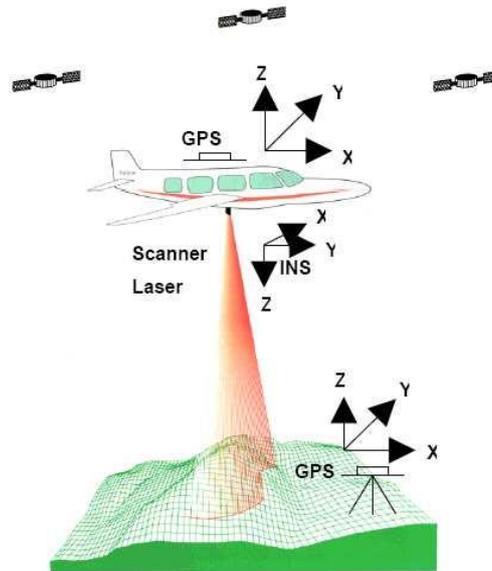


Figura 5. Funcionamiento de la tecnología LiDAR.

Fuente: (Aguilar, Fernando & M.A., 2010) Recuperado de: <https://goo.gl/images/XbkS17>

La tecnología LiDAR tiene aplicaciones en hidrología, urbanismo, topografía, geografía, geología, sismología y física de la atmósfera. (Magazine, 2015).

2.3 SISTEMA DE MAPEO MÓVIL

Un Sistema de Mapeo Móvil (MMS, *Mobile Mapping Systems*), figura 6, está conformado principalmente de tres partes, que son: Sensores (dispositivos de captura de datos, basados en tecnología LiDAR), una unidad de posicionamiento y navegación (IMU, Unidad de medida inercial, la cual está constituida por giroscopios y acelerómetros) y una unidad de referencia de tiempo. (Puente, González-Jorge, Martínez-Sánchez, & Arias, 2013).

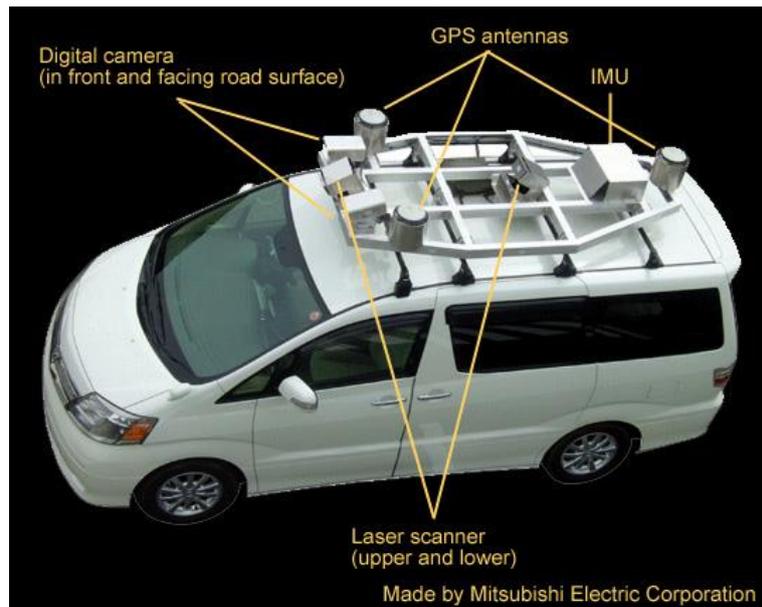


Figura 6. Mapeo móvil.

Fuente: PASCO, (2018). Recuperada de: <https://goo.gl/images/UMdmzA>

2.4 SENSOR LÁSER

Es un dispositivo electrónico que hace parte del sistema LiDAR, el cual es el encargado de capturar datos de distancias provenientes del medio en el que se desenvuelve el sistema y presentarla en forma una nube de puntos. Este sensor es de tipo activo, emite haces de luz láser infrarroja, (en el rango de 600nm – 1000nm para fines no científicos, y los 1550 nm de longitud de onda) y que rebotan en los objetos que se encuentran en su espacio de cobertura.(LiDAR UK, 2018).

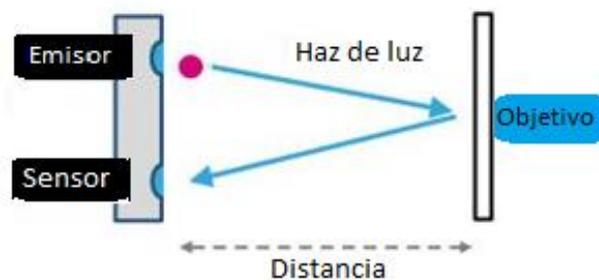


Figura 7. Funcionamiento sensor laser

Fuente: Electronic products (2016). Recuperado de: <http://m.eepw.com.cn/article/201608/296180.html>

La distancia medida entre el sensor y el objeto se ilustra en la figura 7 y está dada por la siguiente ecuación:

$$Distancia (R) = \frac{c * t}{2}$$

Donde,

R = Distancia entre el sensor y el objeto.

c = Velocidad de la luz.

t = Tiempo entre la emisión y recepción del pulso.

Los sensores láser LiDAR tienen una visión de 360°, lo que permite obtener información completa de todo su alrededor, además, pueden ser para uso en interiores (*indoor*) o exteriores (*outdoor*); estos últimos funcionan en ambientes al aire libre, soportando incluso la exposición de luz solar, sin embargo, cuentan con una gran desventaja, su alto costo.

Algunas propiedades de los sensores láser LiDAR son las siguientes:

- Menor longitud de onda que el radar.
- Mayor detalle en la observación.
- No penetra en capas espesas como nubes.
- Láser clase 1. Protección visual.

El uso de los sensores LiDAR se halla en diferentes aplicaciones como escaneo de edificios para generar modelos 3D, topografía para aplicaciones de mapas, pistolas láser para detectar velocidades, sistemas de navegación de robots y óptica adaptativa, entre muchos otros.

2.5 ROS (Robot Operating System)

El Sistema Operativo de Robots (ROS) es una plataforma que cuenta con un conjunto de bibliotecas y herramientas de software que ayudan a construir aplicaciones para robots. Cuenta con controladores, algoritmos de última generación, y potentes herramientas de desarrollo. Se trata de un software de código abierto.

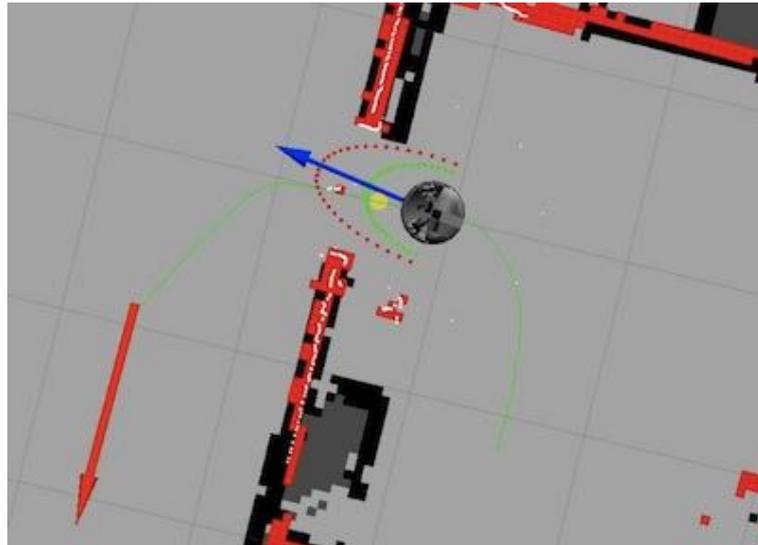


Figura 8. ROS, utilizado para localización del robot.
Fuente: ROS (2017). Recuperada de:<http://www.ros.org>

ROS incluye paquetes que ayudan a resolver problemas de estimación, localización, mapeo y navegación de robots móviles (Figura 8). (Open Source Foundation Robotics, 2018).

Entre sus características se pueden mencionar:

- ROS es modular.
- Cada módulo es autónomo.
- Multilingüe
- Gratis y de código abierto.

ROS busca estandarizar el código utilizado para la programación de robots ayudando a que los nuevos desarrolladores cuenten con código ya disponible para su reutilización, logrando, además, crear una comunidad que comparta sus conocimientos adquiridos.(Autracen, 2018).

2.6 TARJETA DE PROCESAMIENTO

El núcleo del sistema, tanto de hardware como de software, es la tarjeta de procesamiento, la cual, es una computadora de placa reducida, con características similares a una computadora de escritorio. Es la encargada de integrar los diferentes dispositivos y periféricos con los que cuenta el robot móvil. Además, contiene el Sistema Operativo del Robot con el cual se logra una comunicación a nivel de software con

elementos internos y externos al robot.

En el mercado actual existen diferentes opciones de tarjetas de procesamiento apropiadas para integrar en el actual trabajo, entre las cuales se puede resaltar la serie NVIDIA Jetson, las cuales cuentan con un procesador Tegra de NVIDIA que integra una Unidad de procesamiento central de arquitectura ARM. (elinux, 2015)

En la tabla 1, se resumen algunas de las características de cinco tarjetas de procesamiento que se encuentran en el mercado actual, y que se consideraron para el presente trabajo.

Tabla 1. Características tarjetas de procesamiento

	Raspberry pi 3B	Beaglebone Black	Intel Joule T 5700	nVidia Jetson TK1	Dragon Board
Procesador	Broadcom BCM2837 based on ARM Cortex A-53	TI AM3358 based in ARM cortex-A8	Intel Atom T5700	nVidia 4-plus-1 based on ARM Cortex-A15	Qualcomm Snapgragon 410C based on ARM Cortex A-53
Velocidad de reloj	1.2 Ghz	1Ghz	1.7 Ghz	2.3 Ghz	1.2 Ghz
GPU	Broadcom Video core IV	Power VRSGX530	Intel HD	nVidia Kepler GPU	Qualcomm adreno 306
Almacenamiento	No	4 GB eMMC flash	16 GB eMMC flash	16 GB eMMC flash	8 GB eMMC flash
4k video encode	No	No	4kx2kp60 (H.264)	4kx2kp24 (H.264)	1080 P30 (H.264)
4k video decode	No	No	4kx2kp60 (H.264)	4kx2kp30 (H.264)	1080 P30 (H.264)
Interface de cámara	Yes 1xMIPI-CSI	Yes 1xGPMC	Yes 2xMIPI-CSI	Yes 2xMIPI-CSI, x4 and x1	Yes 1xMIPI-CSI
Consumo de potencia	250-750 mA. @ 5V	210-460 mA. @ 5V	130-600 mA. @ 12V	250-4800 mA. @ 12V	50-400 mA. @ 12V

Fuente: Griffith, B (2018). Recuperada de: <https://www.arrow.com/es-mx/research-and-events/articles/what-is-a-single-board-computer>

3. CAPÍTULO TRES. PROCEDIMIENTO

El presente proyecto dispone de un diseño físico a partir del Robot Rover, un diseño a nivel de software y finalmente una etapa de recolección de datos en campo. Este

capítulo presenta la forma, elementos y pasos que se realizaron en la elaboración de cada etapa.

3.1 DISEÑO DE HARDWARE

El sistema de navegación del robot Rover, cuenta con algunos requisitos de hardware para lograr un buen funcionamiento, los cuales son:

1. Se enfoca a robots con ruedas de accionamiento diferencial. Asume que la base móvil se controla enviando comando de velocidad en los planos x y z.
2. Requiere un láser plano montado en algún lugar de la base móvil. Este láser se utiliza para la creación y localización de mapas.
3. El rendimiento del software ROS es mejor en robots con geometría cuadrada o circular (Foundation Robotics, 2017).

3.1.1 Robot Rover 6x6



Figura 9. Robot Rover de 6x6.
Recuperada de: Minds-i (2018)

El robot Rover que se utiliza en el proyecto es de tipo diferencial de 6x6 (Figura 8). Cuenta con suspensión independiente en cada una de las 6 ruedas y dirección en cuatro de ellas (Las dos ruedas centrales son fijas). Sus dimensiones son:

Longitud: 597 mm

Alto: 190 mm

Ancho: 425 mm

Base de rueda: 404 mm

Longitud de la suspensión: 64 mm

Diámetro de las rudas: 140 mm

Sus dimensiones, geometría, suspensión y poder de maniobrabilidad hacen que el robot Rover tenga características que lo hacen apropiado para el desplazamiento entre los surcos de un cultivo de papa, dado que dichos cultivos presentan distancias de siembra que van de 90 a 100 cm entre surcos, diferentes condiciones de irregularidad del terreno y múltiples obstáculos de diferentes tamaños y geometría. (MinAgricultura & DANE, 2013)

3.1.2 Sensor láser RPLiDAR 360

El láser seleccionado es el sensor RPLiDAR 360 A1, el cual es de tipo *indoor* y realiza escaneo láser omnidireccional, en un rango de 6 metros y es apto para proyectos de navegación y localización de robots con uso de SLAM. Otras características son:

- Escáner láser 2D de 360 grados.
- Rango de escaneo 5.5 hz.
- Resolución angular: 1 grado.

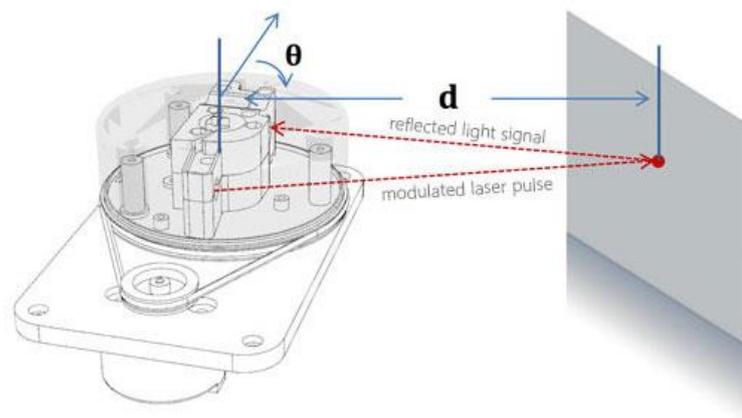


Figura 10. Funcionamiento del sensor láser RPLiDAR 360.
Recuperada de: Roboshop. 2017.

Específicamente, el sensor RPLiDAR, desarrollado por Robopeak, basa su funcionamiento en el principio de triangulación láser, adquisición y procesamiento de hardware. El sistema toma datos de distancia a más de 2000 veces por segundo y con

alta resolución (< 1% de la distancia). RPLIDAR emite una señal láser infrarroja modulada y ésta es reflejada por el objeto a ser detectado (Figura 10). La señal de retorno es muestreada por el sistema de adquisición del sensor y el DSP incorporado. (RoboPeak Team, 2014).

El sensor está conectado a través de un puerto USB 2.0 a la tarjeta NVIDIA Jetson TK1.

3.1.3 Tarjeta NVIDIA Jetson TK1

El proyecto requiere de una unidad de procesamiento bastante robusta, por lo que se ha hecho uso de la tarjeta NVIDIA Jetson TK1. Esta cuenta con una CPU (Unidad Central de Procesamiento), una GPU (Unidad de Procesamiento Gráfico) y una ISP (Programación en el Sistema) en un solo procesador. Algunas de sus aplicaciones son los sistemas robóticos y la visión computarizada; temas centrales del presente proyecto. Además, existe una mayor facilidad para su implementación, ya que, en los laboratorios de la Escuela Colombiana de Ingeniería se cuenta con varias de ellas.

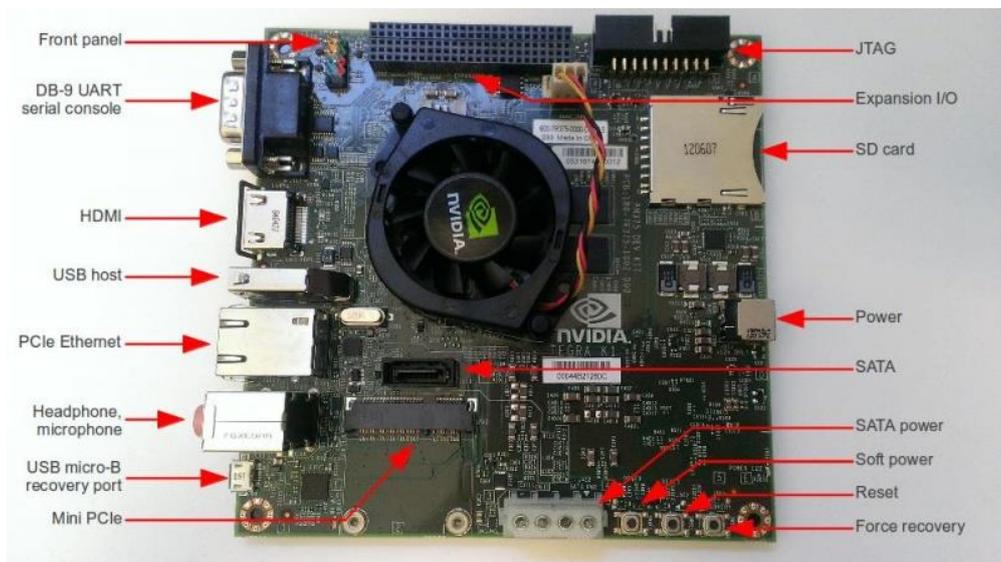


Figura 11. Tarjeta NVIDIA Jetson TK1.

Fuente: Community Jetson TK1 (2017). Recuperada de: http://elinux.org/Jetson_TK1

La tarjeta viene con un sistema operativo Linux, Ubuntu 14.04, y controladores pre-configurados (Figura 10). Sus principales características de acuerdo a la información del fabricante (NVIDIA, 2017), son:

- Procesador Tegra K1 SOC(CPU + GPU + ISP)
- Consumo de energía 1 - 5 vatios

- GPU NVIDIA Kepler GPU "GK20a" con núcleos CUDA (Arquitectura de cálculo paralelo) 192 SM3.2 (hasta 326 GFLOPS)
- CPU: NVIDIA "4-PLUS-1" 2.32GHz ARM de cuatro núcleos Cortex-A15
- DRAM: 2 GB DDR3L 933MHz x16 EMC utilizando anchura de los datos de 64 bits
- Almacenamiento: 16 GB rápida eMMC 4,51 (encaminan a SDMMC4)
- mini-PCIe : una ranura de media altura de un solo carril PEX (como por Wifi, SSD RAID, FireWire o addon Ethernet tarjetas)
- SD / tarjeta MMC: una ranura de tamaño completo (enrutado a SDMMC3)
- Puerto USB 3.0
- Puerto USB 2.0
- Puerto HDMI
- Puerto RS232
- Audio: un códec de audio Realtek HD ALC5639 con micrófono en línea.
- Ethernet: a / 1000BASE-T Puerto Gigabit LAN Realtek RTL8111GS 10/100
- SATA
- JTAG : un 2x10 pines
- Un conector de alimentación de 12V DC barril y un conector de alimentación IDE PC de 4 pines
- Ventilador

3.1.4 Control de servos

El control de servos es un dispositivo electrónico que permite controlar un servo motor a través de PWM (Modulación por ancho de pulsos). El presente proyecto hace uso de la tarjeta Adafruit PCA9685 (Figura 12), la cual es un módulo controlador de servos con 16 salidas PWM, 12 bits de resolución (0-4095) y una frecuencia máxima de 1600 Hz. Es recomendable conectar una fuente externa, para suplir la demanda de corriente de los servos. Las características del módulo son:

- Voltaje de funcionamiento 5V.
- Frecuencia ajustable PWM alrededor de 1,6Khz.
- Salida configurable *push-pull* u *open-drain*.
- Pin de habilitación de salida para desactivar rápidamente todas las salidas.

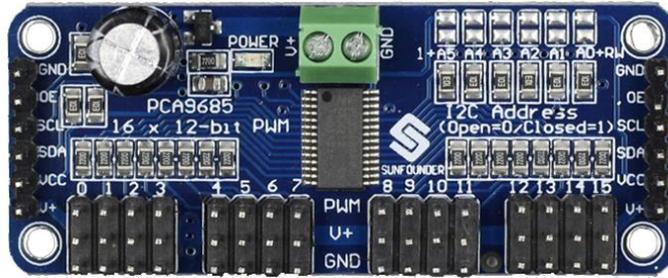


Figura 12. Servo controlador PWM Adafruit PCA9685.
Fuente propia. 2018

Esta placa está conectada directamente con la tarjeta Jetson Tk1 a través del puerto I2C bloque J3A1 (Anexo 10), cuyos pines a conectar se detallan en la figura 13.

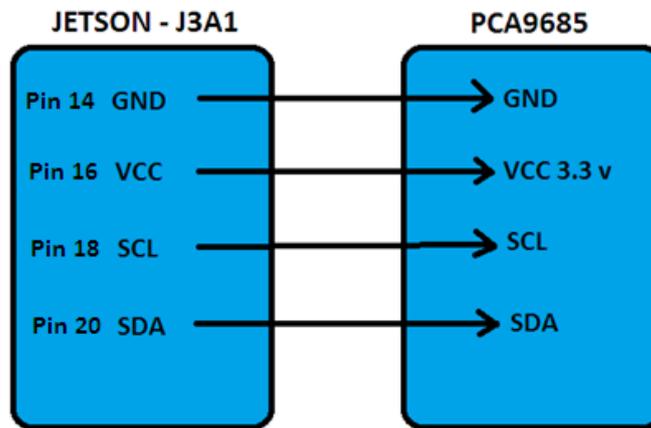


Figura 13. Conexión Jetson TK1 y PCA9685.
Fuente propia. 2018

En la tarjeta Jetson Tk1 se pueden encontrar dos buses I2C, los cuales se identifican en la siguiente figura:

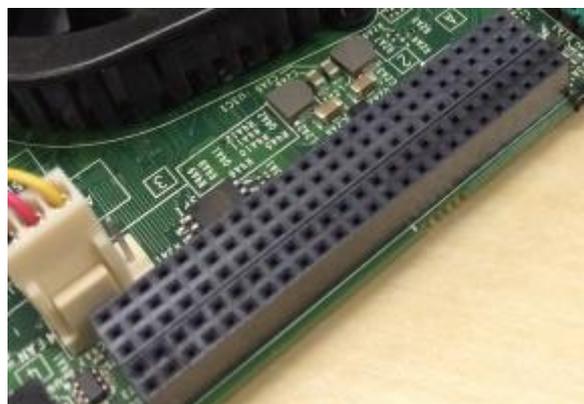


Figura 14. Jetson Tk1, puerto I2C.
Fuente: elinux (2018). Recuperada de: http://elinux.org/Jetson_TK1

Es importante analizar el consumo en potencia de los motores, para evitar corrientes excesivas que puedan dañar alguno de los componentes del sistema. La tarjeta PCA 9685, cuenta con un terminal denominado V+, el cual puede ser utilizado para alimentar los servos. Se recomienda conectar los motores mediante una fuente externa, dado que el consumo de potencia puede ser superior al entregado por el módulo de control de servos.

3.1.5 IMU BNO055

La Unidad de Medida Inercial (IMU) que realiza el proceso de odometría en el presente proyecto es la placa BNO055 de Adafruit, la cual, es un sensor inteligente de 9GDL que brinda la posibilidad de leer los datos directamente sobre el puerto I2C.

El dispositivo BNO055 puede generar los siguientes datos:

- Orientación absoluta. (Vector de Euler, 100 Hz) Datos de orientación de tres ejes basados en una esfera de 360 °
- Orientación absoluta. (Quaternion, 100Hz).
- Vector de velocidad angular (100 Hz) Tres ejes de 'velocidad de rotación' en rad / s.
- Vector de aceleración (100Hz) 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).
- Linear Acceleration Vector (100Hz) Tres ejes de datos de aceleración lineal (aceleración menos gravedad) en m / s^2 .
- Gravity Vector (100Hz) Tres ejes de aceleración gravitacional (menos cualquier movimiento) en m / s^2 .
- Temperatura (1Hz) Temperatura ambiente en grados centígrados.

Datos tomados de (Adafruit, 2017)

La tarjeta tiene dimensiones pequeñas, 20mm x 27mm x 4mm, y se puede observar figura 14:

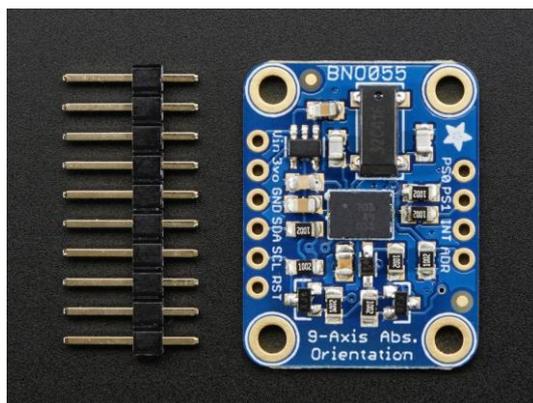


Figura 15. IMU placa Adafruit BNO055.
Recuperada de: Adafruit. (2017)

Su conexión se hace de manera directa con la tarjeta Jetson TK1 en el puerto I2C – J3A1. Sus conexiones son las siguientes:

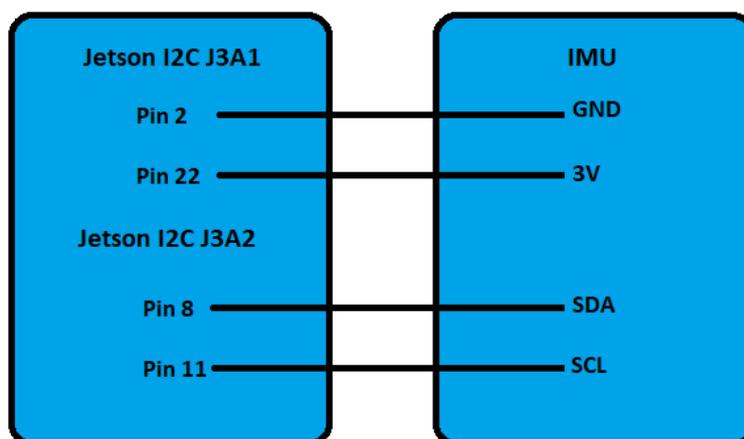


Figura 16. Conexión Jetson TK1 y BNO055.
Fuente propia. 2018

3.2 DISEÑO DE SOFTWARE

El proyecto realiza su funcionamiento basado en software de alto nivel, denominado ROS (*Robot Operating System*), el cual es el encargado de proveer herramientas de control, visualización y comunicación.

Es de recordar que, el presente proyecto es parte de un macro proyecto de investigación de la Escuela Colombiana de Ingeniería denominado “Caracterización de áreas de cultivo mediante sensor RPLiDAR 360° láser scanner y robot Rover” y es por eso que el diseño busca crear herramientas que puedan ser retomadas en trabajos posteriores, como, por ejemplo, el sistema de navegación del robot. Las configuraciones de los

diferentes paquetes de ROS, que se incluyen en el sistema asociado con la navegación del robot Rover en el cultivo de papa, se realizan mediante una terminal de Ubuntu. El sistema de navegación de ROS basa su funcionamiento en datos de Odometría y sensores, dando, a partir de estos, señales de control al robot móvil.

De acuerdo a la información de configuración de la pila de navegación de wiki.ros, los nodos, tópicos y mensajes que se deben configurar, son los que se encuentran en la siguiente imagen.

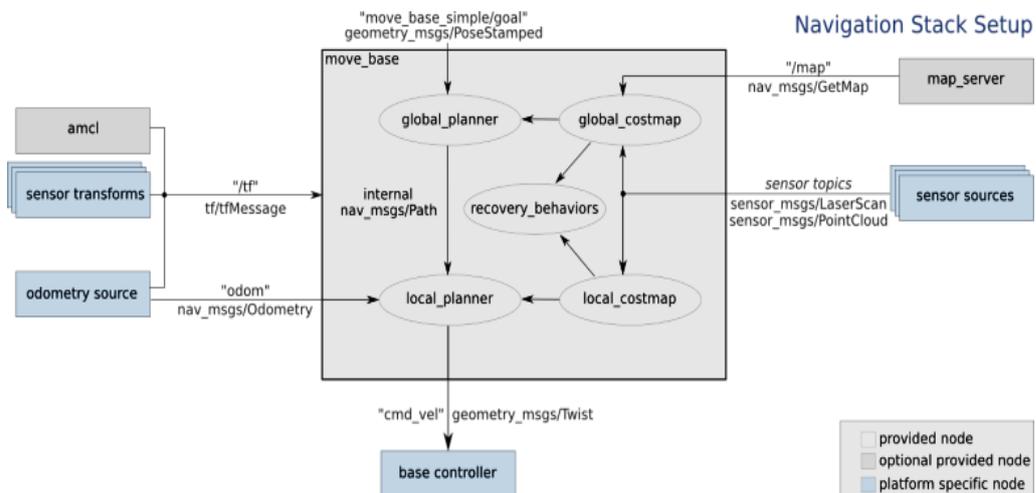


Figura 17. Configuración de la pila de navegación del robot.

Fuente: Wiki ROS (2018). Recuperada de: <http://wiki.ros.org/navigation/Tutorials/RobotSetup>

En la figura 17, los cuadros en azul son los temas que se deben realizar para lograr una adecuada integración con los demás temas de la pila de navegación.

3.2.3 Configuración de la tarjeta embebida

La tarjeta NVIDIA Jetson TK1 requiere de una configuración previa, antes de iniciar su uso. Para una correcta configuración se deben conectar dispositivos de entrada (Teclado y ratón) y salida (Pantalla), con el fin de realizar la instalación del software (Figura 18).

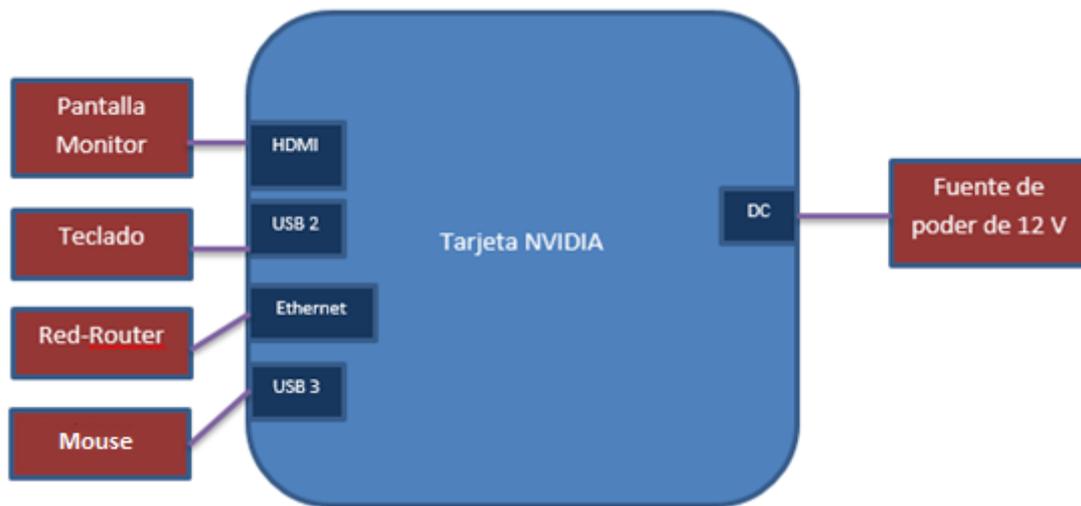


Figura 18. Conexión de dispositivos externos a la tarjeta NVIDIA para la configuración inicial.
Fuente Autor, 2018.

La tarjeta viene con el sistema operativo Linux4Tegra preinstalado, que corresponde a Ubuntu 14.04. Este sistema operativo es el más estable sobre el cuál se puede instalar ROS. Además, los controladores de cada uno de sus puertos ya se encuentran preinstalados, lo que facilita aún más el diseño.

Precaución: Antes de conectar el sistema a internet, se debe digitar el siguiente comando:

```
$ sudo apt-bodega marca servidor X-xorg-core
```

Esto evitará que, al momento de actualizar el sistema, se carguen archivos incorrectos para el controlador de gráficos de la tarjeta.

3.2.4 Instalación del Sistema Operativo para Robots ROS

La versión seleccionada de ROS para desarrollar el presente trabajo es Indigo, debido a que es la distribución más estable para el sistema operativo 14.04.2 de Ubuntu. Esta distribución fue lanzada en julio de 2014 y tendrá soporte hasta abril de 2019. Los pasos para su instalación se pueden encontrar en el anexo 1.

A continuación, se realiza una descripción de las etapas llevadas a cabo para la configuración en ROS de cada uno de los paquetes, necesarios para el funcionamiento del proyecto.

3.2.5 Espacio de trabajo

Lo primero fue crear el espacio de trabajo o comúnmente llamado *workspace* (*ws*) en el cual se alojaron los paquetes necesarios para el correcto funcionamiento del proyecto (Un paquete en ROS, *packages*, es la unidad funcional más pequeña, la cual puede contener código, datos o documentación) específicamente en el directorio *src*. Anexo 2.

3.2.6 Comunicación entre máquinas

El proyecto se diseña con un equipo base (Computador portátil) y un equipo embebido (Tarjeta de comunicación NVIDIA adaptada al robot Rover), por lo que, para lograr una comunicación remota, se realiza una configuración con una arquitectura cliente/servidor, mediante el uso del protocolo SSH (Secure SHell, intérprete de órdenes seguro). La configuración específica se resume en el anexo 3.

3.2.7 Sistema de navegación

Para lograr la navegación del robot, sobre un espacio determinado, es preciso que conozca en donde se encuentra y hacia donde debe ir. Es preciso aclarar que el objetivo final del presente proyecto no es lograr la navegación del robot, sino la detección de surcos y finales de surcos en un cultivo de papa; sin embargo, sí es la base para un sistema de navegación, por lo cual, también se buscará, dejar los nodos utilizados, con los mensajes que requiere un sistema de navegación en ROS.

3.2.8 Configuración del sensor LiDAR

El paquete necesario para la configuración del sensor láser RPLiDAR se puede descargar o copiar de la siguiente dirección: https://github.com/robopeak/rpLiDAR_ros

El sensor láser requiere de una rápida configuración para iniciar el proceso de captura de datos, la cual se describe en el anexo 4. Una vez terminada la configuración, automáticamente se abre una ventana en RViz, en la que se observa la nube de puntos captada por el sensor (En rojo, figura 19).

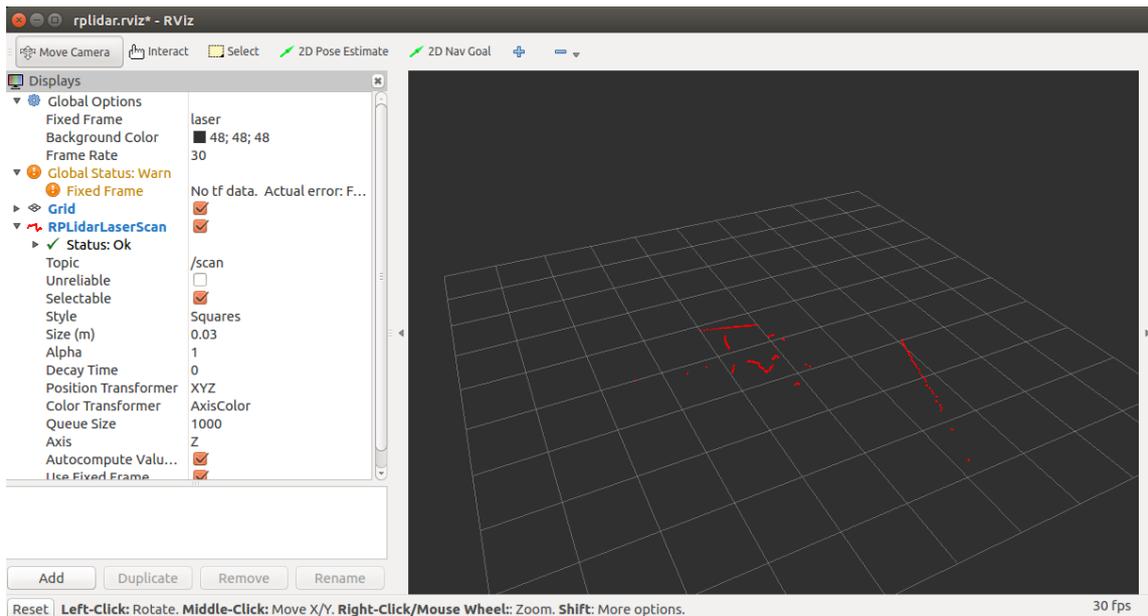


Figura 19. Ventana de RViz en la que visualizan los datos del láser por medio de puntos.
Fuente autor (2017).

3.2.9 Configuración del control de servos

La placa PCA9685 está directamente conectada a la tarjeta Jetson TK1 y al regulador DC/DC y es la encargada de controlar los servos de dirección y velocidad del robot Rover. El paquete del PCA9685 se descarga de la página <https://github.com/jetsonhacks/JHPWMDriver.git> y la respectiva configuración se encuentra en el anexo 5. Las modificaciones que se hacen, en el ángulo de giro y velocidad de los motores, se ejecutan en el software QtCreator, el cual es un entorno de desarrollo multiplataforma, que brinda la posibilidad de editar en código C++, entre otros.

3.2.10 Teleoperación

El Control de velocidad y dirección del robot Rover se lleva a cabo a través de un *joystick* inalámbrico. La configuración del *Joystick* se hace descargando el paquete desde la dirección http://wiki.ros.org/teleop_twist_joy.

3.2.11 Configuración del IMU

ROS cuenta con un paquete llamado RTIMULib, el cual se descarga de <https://github.com/jetsonhacks/RTIMULib.git> para la respectiva instalación y configuración del nodo del sensor IMU. El código de este nodo, se abre y compila

mediante el uso de QtCreator (Anexo 6), obteniendo una imagen como la que se aprecia en la figura 20, en la que se visualizan las medidas obtenidas.

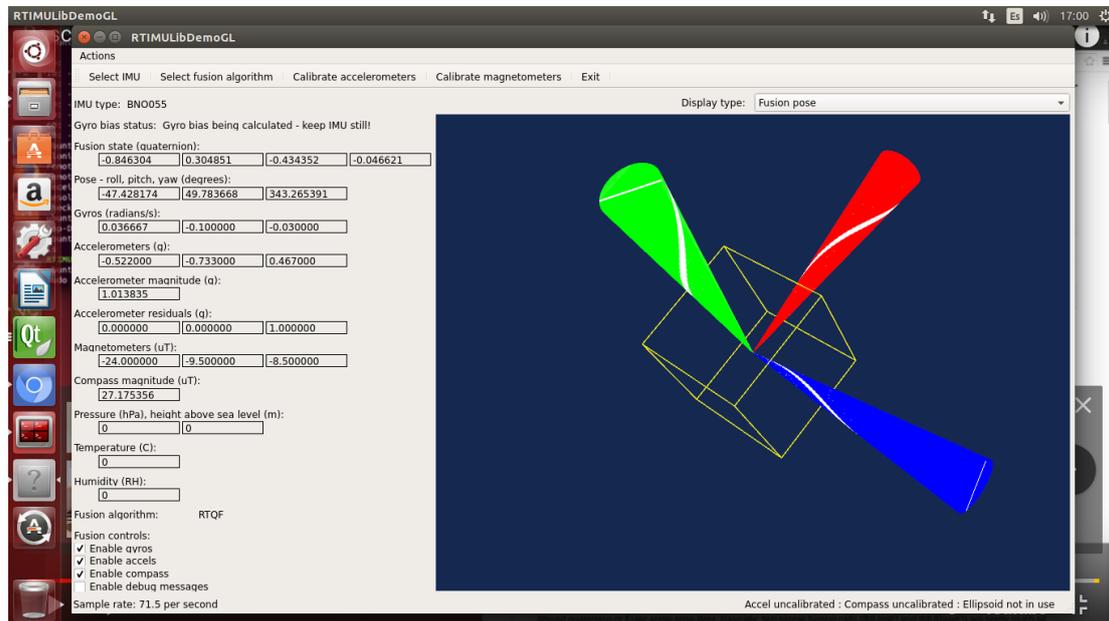


Figura 20. Simulación IMU.
Fuente autor. (2018)

La información de configuración se basa en (JetsonHacks, 2018)

3.2.12 Generación de la carpeta para archivo de datos – rosbag

Rosbag es una herramienta de ROS cuya función es la de guardar y reproducir mensajes. Se puede publicar mensajes de diferentes bolsas, resumir los contenidos de las bolsas y realizar filtrado de mensajes. Es muy útil para contar con información capturada de sensores instalados en el robot y lograr un posterior análisis a través de algún algoritmo computacional.

Para la construcción de un mapa lo más recomendable es guardar los datos obtenidos por el sistema de odometría, gps y demás sensores, para luego, a partir de dichos datos, generar el mapa. Lo anterior permitirá evitar posibles errores al momento de realizar la lectura de datos para la construcción del mapa, ya que, de hacerse en línea, se puede generar una duplicidad en los datos. La creación y configuración del Ros bag se explica en detalle en el anexo 7.

3.2.13 Construcción de mapas

Para que el robot pueda desplazarse en un determinado espacio es necesario que conozca su entorno, es decir, su ubicación en el espacio y hacia dónde se dirige, por lo que es importante la construcción de un mapa para lograr un desplazamiento del robot de acuerdo a las necesidades del proyecto. Para lograr obtener un buen mapa es conveniente incorporar diferentes sensores en el robot. Sin embargo, dichos sensores cuentan con diferentes porcentajes de error, que sumados, generan un grado de incertidumbre a tener en cuenta al momento de iniciar algún tipo de pruebas.

El sistema operativo ROS, cuenta con diferentes herramientas que facilitan la construcción de mapas del mundo. Los mapas son construidos a partir de celdas 2D, las cuales tienen valores correspondientes a la ocupación del espacio. Los puntos blancos representan espacio abierto; los negros, ocupado; y los grises, desconocido. Los mapas generados son almacenados como imágenes en diferentes formatos (PNG, JPG y PGM). Junto con estas imágenes, se crea un archivo YAML, el cual guarda información adicional, como resolución, el origen del mapa y límites para determinar si una celda está ocupada o desocupada.

Las imágenes van siendo construidas desde la parte superior izquierda de la pantalla, desplazándose por el eje y, guardando valores de su recorrido, los cuales van entre 0 y 255. Los valores altos, como 255, corresponden a blanco y los bajos a negros (0). Una celda está ocupada si tiene un valor mayor al 65% en la escala de grises, o desocupada, si se encuentra con un porcentaje inferior al 19.6%.

El presente trabajo configura los nodos utilizados, de tal manera, que sirvan para que en trabajos posteriores se aplique el meta paquete de navegación de manera correcta.

3.3 PRESUPUESTO

El proyecto requirió de elementos con los que ya se contaba en la Escuela Colombiana de Ingeniería y que fueron facilitados para el desarrollo del mismo. Estos elementos no se incluyen en el costo total del proyecto. El software usado es de uso libre, por lo que tampoco tiene costo en la tabla 2. A continuación, se describen los elementos de hardware, software y transporte que se necesitaron para el proyecto.

Tabla 2. Presupuesto del proyecto.

TIPO	CATEGORIA	RECURSO	DESCRIPCIÓN	MONTO (\$ Pesos)
Recursos disponibles	Software	Configuración de equipo	ROS (Robot Operating System)	
			ROS Indigo Igloo	
			Linux4 Tegra (L4T). Ubuntu 14.04.	
	Hardware	Equipo	Vehículo diferencial terrestre. Robot Rover de tres ejes	
			Tarjeta de comunicación NVIDIA Jetson TK1.	
			RPLiDAR A1M-360 Degree Laser Scanner Development Kit.	
			TV LED LG 32" entrada HDMI	
			Teclado Genius alámbrico	
			Mouse Genius alámbrico	
			Cable HDMI	
Recursos necesarios	Hardware	Equipo	Control de servos Adafruit PCA9685	\$18.000
			Unidad de Medida Inercial (IMU) BNO085 de Adafruit	\$172.312
			Módulo Lm2596 Convertidor Fuente DC-DC Arduino	\$5.500
			Batería LiPo de 2200 mAh 11.1 V ZIPPY Compact.	\$86.800
			Indicador de tensión con alarma para baterías LiPo 1-8S	\$10.000
			Cargador de baterías LiPo B3AC 7.4V/11.1V 2S/3S	\$54.000
			Dual Band Wireless -N 7260 Uso para Intel 7260	\$97.000
			Antena Wifi modem internet	\$35.000
	Gastos de trabajo en campo	Transporte	Transporte Gasolina (\$10.000) y peaje (\$8.300) por desplazamiento a la Escuela Colombiana de Ingeniería. Total 18 desplazamientos.	\$329.400
	Total			

Fuente: Autor (2018).

4. CAPÍTULO CUATRO. PRUEBAS DE CAMPO

4.1 DISEÑO EXPERIMENTAL

La estrategia metodológica que se utiliza para el presente proyecto es la de validación de instrumentos, dado que se evalúa el sistema móvil LiDAR para detección de surcos y finales de surco en un cultivo de papa.

4.1.1 Tipo de investigación

Se trata de una investigación experimental, en la cual, partiendo de la manipulación de una variable independiente, en este caso la distancia obtenida por el sensor en cada una de las mediciones, se logra la obtención de valores binarios para una variable dependiente. Estas variables se encuentran en la nube de puntos que entrega el sensor LiDAR.

4.1.2 Fuentes de información

Fuente primaria: Los puntos generados por el sensor láser son tomados como información primaria para el análisis posterior del proyecto.

Fuentes secundarias: Se realiza una revisión bibliográfica para obtener referencias cualitativas y cuantitativas de mediciones realizadas con sensores LiDAR en espacios libres, especialmente en áreas de cultivos.

4.1.3 Técnicas de recolección de información

Los datos primarios se obtienen colocando directamente el Robot Rover en los surcos del invernadero de la Escuela Colombiana de Ingeniería e iniciando el sistema LiDAR, tal como se aprecia en la figura 21.



Figura 21. Robot Rover en el invernadero de la Escuela Colombiana de Ingeniería.
Fuente autor, 2018.

El mando a través del Joystick permite que el Robot Rover se desplace adecuadamente a través del surco y recopile la respectiva nube puntos, los cuales van siendo almacenados automáticamente en un archivo bag creado en la raíz del sistema operativo de la tarjeta Jetson TK1. Como el proyecto requiere que el robot identifique el surco y final de surco por donde se está desplazando, se decide tomar bags de datos por cada surco del cultivo.



Figura 22. Movimiento del vehículo a través de los surcos del invernadero controlado por el joystick.
Fuente autor, 2018.

El desplazamiento del Rover se realiza de extremo a extremo de cada uno de los surcos (Figura 22), y el sensor van captando información en las siguientes variables:

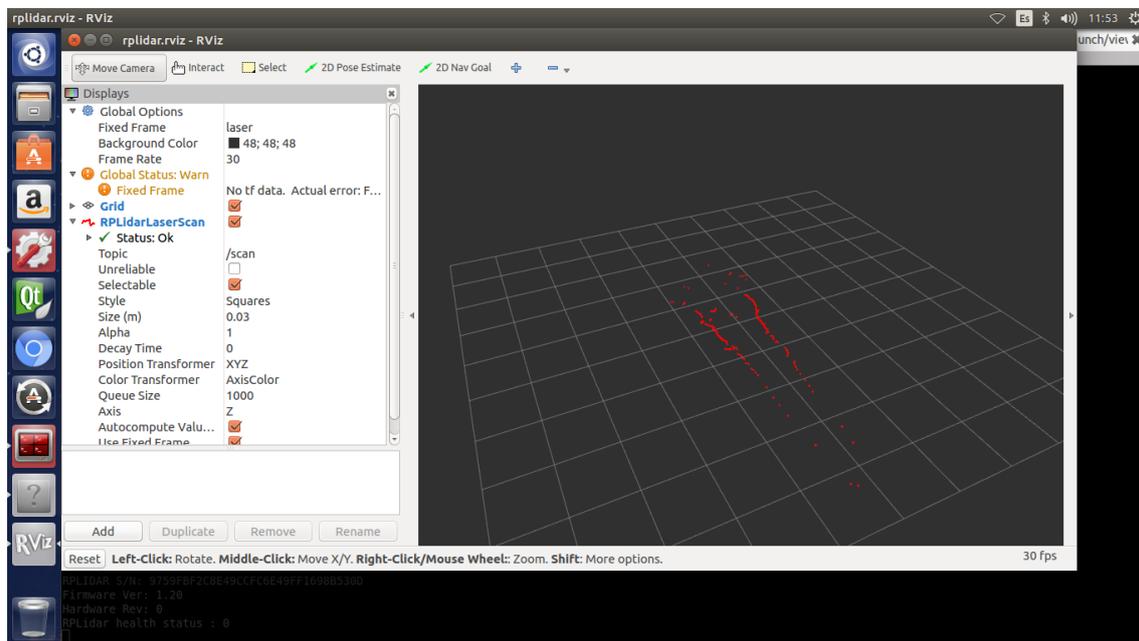


Figura 23. Nube de puntos generada por el LiDAR en uno de los surcos del invernadero.
Fuente autor, 2018.

La figura 23 permite apreciar, mediante RVIZ, la nube de puntos generada por el sensor RPLiDAR. El robot inicia su desplazamiento desde un extremo del surco, que en la figura corresponde a la parte central del gráfico; es por eso que los puntos que se encuentran hacia el centro de la imagen poseen una mayor intensidad que los que se ubican en la parte inferior. Se detallan claramente dos agrupaciones de puntos, que corresponden a los surcos laterales que observa el sensor en su desplazamiento, lo que ubica al robot en medio de estos dos grupos de puntos.

Cada una de las tomas de datos realizadas en el invernadero deja un archivo `.bag`, el cual puede ser revisado en su meta-información a través del comando

`$rosbag info [Nombre del archivo]`

4.1.4 Análisis de datos LiDAR

Debido a que se obtienen datos primarios y se trabaja en un ambiente con la afectación de diferentes variables naturales (como luz solar, imperfecciones del suelo, obstáculos

de tierra y plantas) y artificiales (reflexión del plástico que recubre las paredes del invernadero), existe una probabilidad de error en las medidas obtenidas.

En la figura 24, se puede observar que la nube de puntos describe los dos surcos que se encuentran a cada costado del robot y en el intermedio de estos surcos está ubicado el camino por donde se desplaza el Rover.

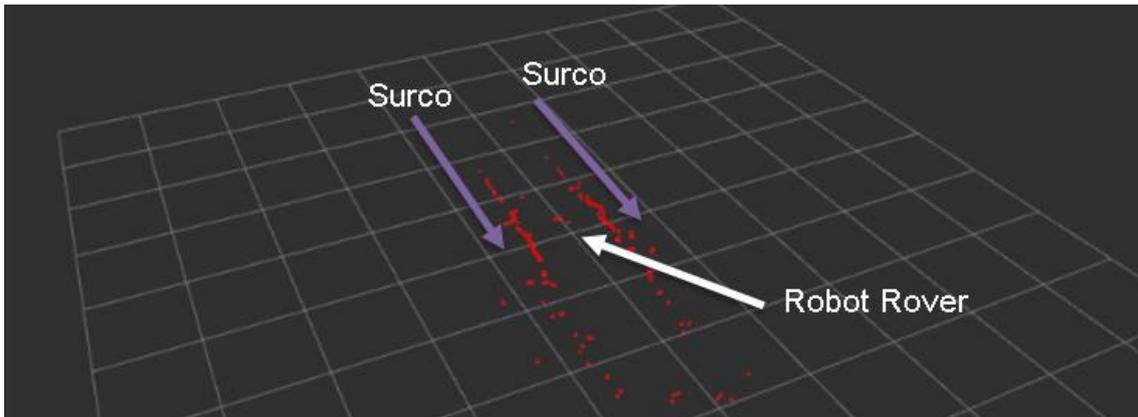


Figura 24. Nube de puntos que representan los surcos del invernadero.
Fuente autor, 2018.

A medida que se desplaza el robot a través de los surcos, van apareciendo algunos puntos en la parte inferior de la gráfica, los cuales corresponden a la pared del invernadero y que se encuentra construida con plástico, tal como se aprecia en la figura 25.

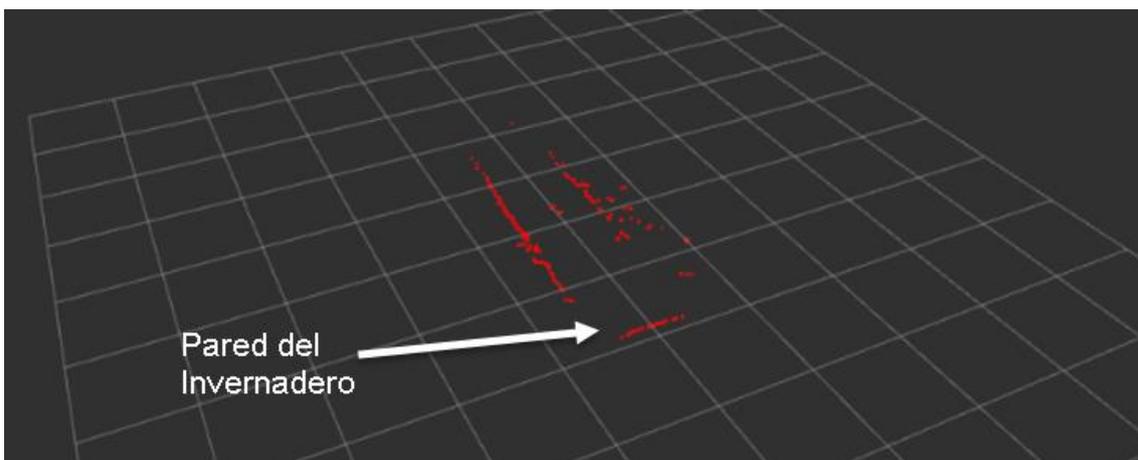


Figura 25. Nube de puntos que identifica parte de la pared del invernadero.
Fuente autor, 2018.

Una vez el robot ha llegado al final del surco se puede detectar nueva información (Figura 26), el final de la secuencia de puntos, que corresponde al final del surco, y el espacio sin puntos que se encuentra entre el final del surco y la pared del invernadero.

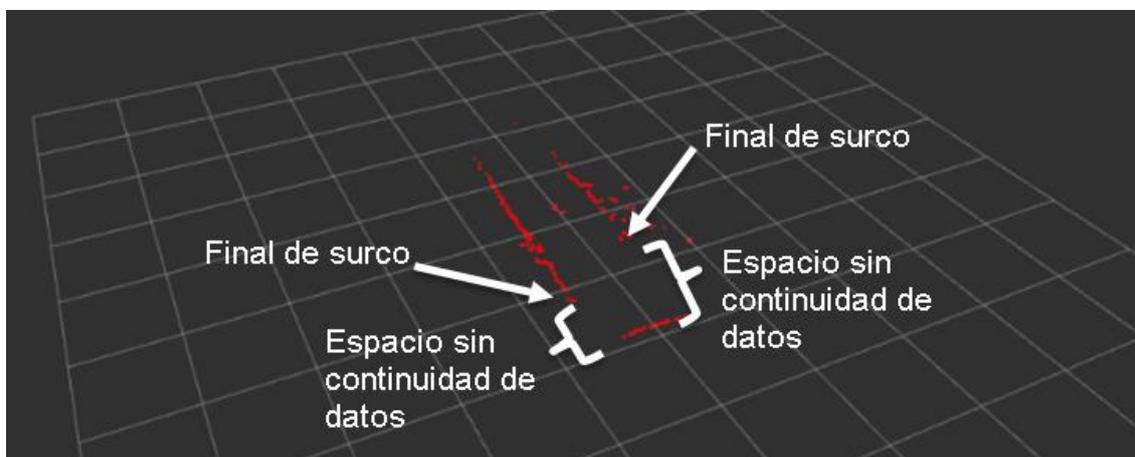


Figura 26. Variables a analizar en la nube de puntos.
Fuente autor, 2018.

La observación y análisis de las gráficas que se generan en RVIZ, permiten tener una idea del algoritmo a desarrollar para la detección de los surcos y finales de surco.

Ahora se pasa al análisis cuantitativo de los datos obtenidos por el sensor LiDAR en cada una de las muestras tomadas en el invernadero y que han sido almacenadas en igual número de archivos *.bag*.

Los sensores RPLiDAR generan una nube de puntos, que representan los datos tomados, los cuales generalmente son consolidados en un archivo *.LAS*. La Sociedad Americana de Fotogrametría y Teledetección (ASPRS), clasificó estos puntos mediante códigos, para los formatos LAS 1.1, 1.2, 1.3 y 1.4. Anexo 8. Los puntos LIDAR se pueden clasificar en varias categorías que incluyen suelo o terreno desnudo, parte superior de cubierta forestal y agua. Las diversas clases se definen mediante códigos numéricos de enteros en el archivo LAS. Sin embargo, en la utilización de ROS, el formato en que se entregan los datos es de tipo *.bag*

Para facilitar el trabajo de análisis de cada uno de los archivos, se realiza la conversión de los datos a un formato *.csv*,. Los datos son entregados de manera continua como se muestra en la figura 27:

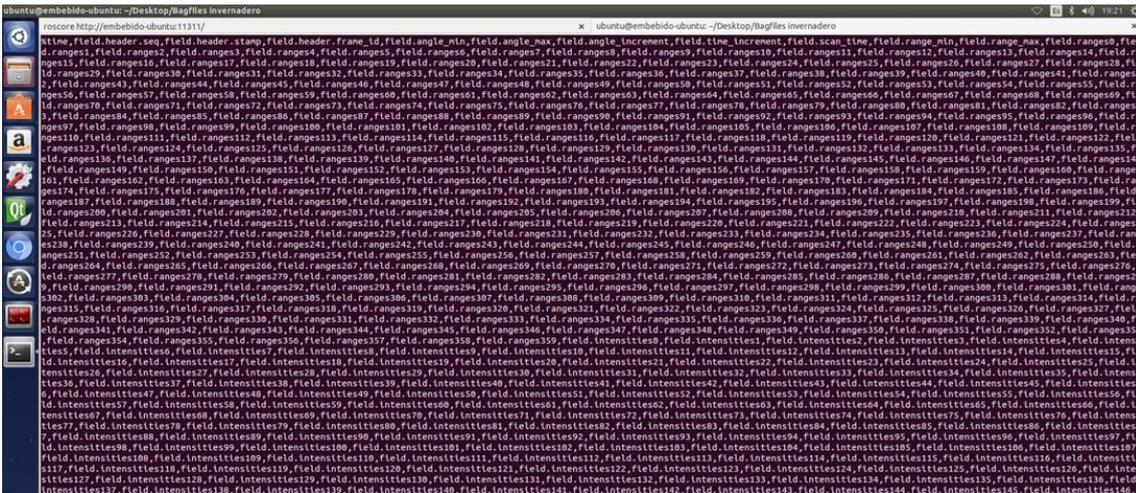


Figura 27. Datos obtenidos del LiDAR expresados en formato .csv.
Fuente autor, 2018.

Una vez se pasan los datos a un archivo .csv, estos son fácilmente exportados a un formato Excel, lo que mejora significativamente el análisis. El formato adquirido se muestra en la figura 28.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	
1	header:																
2	seq:		6018														
3	stamp:																
4	secs:		1529346124														
5	nsecs:		871065478														
6	frame_id:		laser														
7	angle_min:		-3.12413907														
8	angle_max:		3.14159274														
9	angle_incr:		0.017459329														
10	time_incr:		4.28E-07														
11	scan_time:		0.00015382														
12	range_min:		0.15000001														
13	range_max:		8														
14	ranges:	[inf	1.21300006	1.21500003	1.21300006	1.22099996	1.23000002	1.23199999	inf	inf	inf	inf	inf	0.20099999	0.20099999	inf	inf
15	intensities:	[0.0	47	47	47	47	47	47	0	0	0	0	0	0	47	47	0
16	...																
17	header:																
18	seq:		6019														
19	stamp:																
20	secs:		1529346125														
21	nsecs:		34285395														
22	frame_id:		laser														

Figura 28. Datos del LiDAR en formato excel.
Fuente autor, 2018.

Los datos que se pueden identificar en este archivo son:

- Ángulo mínimo (*angle_min*), en grados.
- Ángulo máximo (*angle_max*), en grados.
- Incremento de ángulo (*angle_increment*), en grados.

- Incremento de tiempo (*time_increment*), en segundos.
- Tiempo de escanéo (*scan_time*), en segundos.
- Rango de distancia mínimo (*range_min*), en metros.
- Rango de distancia máximo (*range_max*), en metros.
- Rangos (*ranges*)
- Intensidades (*intensities*)

Si se grafican en un diagrama polar, de los que se incluyen en Excel, es posible obtener las mismas figuras que se mostraban en RVIZ, lo que muestra la coherencia de los datos y su representación. Estas gráficas se incluyen en el anexo 11.

Para el desarrollo del algoritmo se han seleccionado los ítems de rangos y ángulos debido a que ofrecen mayor información y su análisis se puede comprobar con las figuras obtenidas en RVIZ.

El sensor LiDAR tiene una resolución de un grado, por lo que cada revolución logra el escaneo de los 360° de su alrededor. Estos son los datos que se muestran en la figura 32 en cuanto a rangos, nombrados como *field.range1.....field.range269 field.range270 field.range271.....field.range359*.

Los rangos de medida están dados en mm y en el sensor RPLiDAR utilizado para este proyecto van desde los 150 hasta los 6000.

Como el sensor estará escaneando constantemente en su desplazamiento, se observa que en los ángulos de 90° y 270° (Según la configuración del LiDAR y que se encuentra en la figura 34), las mediciones estarán entre 150 mm y 600mm, dado que corresponden a las distancias entre el sensor y el surco, esto da a entender que hay un surco al lado del sensor. Así mismo, cuando el robot llegue al final del surco, el sensor obtendrá medidas, en los mismos ángulos anteriores, superiores a los 600 mm, lo que evidenciará que el vehículo ha llegado al final del surco.

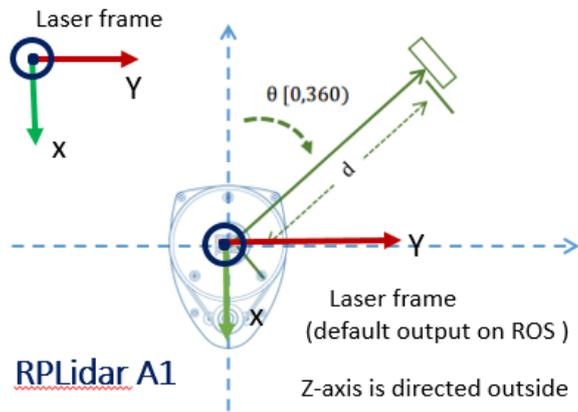


Figura 29. Ángulo del haz de luz del sensor LiDAR.
Tomado de: <http://rosclub.cn/post-1050.html>

Para evidenciar que el robot llega al final del surco, se realiza un programa, basado en el pseudocódigo del párrafo anterior, y que entregue a través de pantalla un mensaje que diga "Final de surco", lo cual se muestra a continuación, figura 30.



Figura 30. Respuesta del sistema al detectar el final del surco.
Fuente autor, 2018.

Este programa se realiza en nodo del control de motores, con el fin de aprovechar la estructura y librerías de este nodo. Debido a que se genera un bucle infinito en el programa, es decir, el sistema se queda generando información continuamente, se incluyen algunas líneas de código adicional para que solo se genere información (Final del surco, si es el caso) cuando se presione uno de los botones del *joystick*. El diagrama de flujo del programa se muestra a continuación, figura 36:

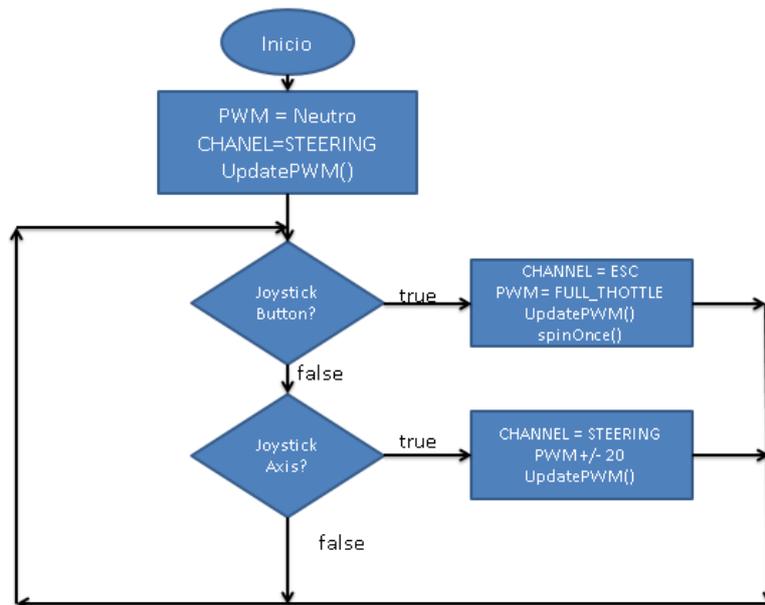


Figura 31. Diagrama de flujo de la activación de los motores de velocidad y dirección del Rover mediante el Joystick.
Fuente autor, 2018.

El algoritmo implementado para la detección de surcos y finales de surco se realiza en lenguaje C, con la estructura que se muestra en diagrama de flujo de la figura 32.

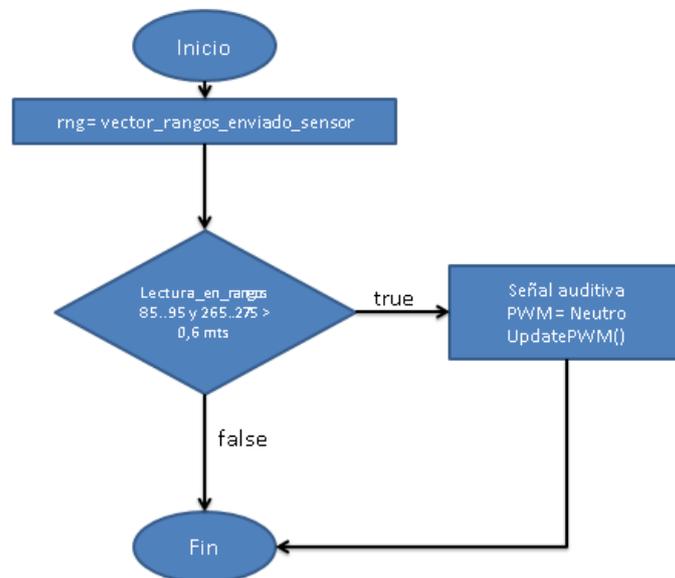


Figura 32. Diagrama de flujo para la detección de surcos y finales de surco.
Fuente autor, 2018.

La nube de puntos entrega datos tomados en los 360° de exploración del LiDAR, los cuales son analizados y utilizados para determinar donde se encuentran los surcos y

final de surco en el cultivo.

5. CAPÍTULO CINCO. CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

El sensor RPLiDAR 360 ofrece un completo escaneo en los ejes X y Y con una excelente resolución, lo que permite obtener información muy precisa de distancias e intensidades, esta última determinada por el tipo de material sobre el que el haz de luz realiza la reflexión.

El sistema implementado es apropiado para detectar los surcos y finales de surco de un cultivo de papa dentro de un invernadero con condiciones de luminosidad en el rango de 8000 lux a 18000 lux (Anexo 9), con un espacio entre surcos mayor a 50 cm y una altura de los surcos mayor a 25 cm.

Para implementar un sistema de navegación con el presente sistema, es recomendable que la distancia entre surcos sea de por lo menos 50 cm, para facilitar el desplazamiento del robot Rover.

5.2 RECOMENDACIONES

La versión de ROS que funciona adecuadamente sobre el sistema operativo Ubuntu 14.04, es la distribución Indigo Igloo, la cual cuenta con soporte hasta el año 2019.

Una vez se esté trabajando con la tarjeta de comunicación, no se recomienda apagarla bruscamente, sino, hacer uso del modo correcto desde el sistema operativo (desde el terminal o desde el botón apagar del sistema).

El adaptador de wi-fi, necesita una versión de Kernel, superior al 3.10, para su funcionamiento.

En el desplazamiento del vehículo a través de los surcos, el vehículo perdía fuerza y se dificultaba la dirección, por lo que es recomendable el cambio de los motores tanto de dirección como de velocidad por unos con mayor torque.

REFERENCIAS BIBLIOGRÁFICAS

- Andrés, E. (2015). Agricultura de Precisión para la Producción de Cultivos de Seguridad Alimentaria y de Agro Exportación.
- Bailey, B. Y. T. I. M., & Durrant-whyte, H. (2006). Simultaneous Localization and Mapping (SLAM), (September), 108–117.
- Banco mundial. (2016a). Población total. Recuperado de:
<https://datos.bancomundial.org/indicador/SP.POP.TOTL>
- Banco mundial. (2016b). Tierras cultivables (% del área de tierra). Recuperado de:
<https://datos.bancomundial.org/indicador/AG.LND.ARBL.ZS?view=chart>
- Barreiro, P. (n.d.). Sensores para la caracterización del suelo agrícola usados en agricultura de precisión. Retrieved from http://oa.upm.es/6209/1/Barreiro_87.pdf
- Best, S., & Zamora, I. (2008). Tecnologías aplicables en agricultura de precisión. Recuperado de:
<http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Tecnolog?as+Aplicables+en+Agricultura+de+Precisi?n#0>
- Castilla, R. (2015). Informe Sostenibilidad y Gestion Riopaila Castilla 2013. *Statewide Agricultural Land Use Baseline 2015*, 1. Recuperado de:
<https://doi.org/10.1017/CBO9781107415324.004>
- IDAE. (2010). *Ahorro y eficiencia energética en la agricultura de precisión*. Recuperado de:
http://www.idae.es/uploads/documentos/documentos_10995_aee_agr_precision_agr16_a2010_69054ba0.pdf
- José A. Sobrino. (2001). *Teledetección*. (G. Impresores, Ed.). Valencia.
- Magdaleno, F., & Martínez, R. (2006). Aplicaciones de la teledetección láser (LIDAR) en la caracterización y gestión del medio fluvial. *Ingeniería Civil*, 142, 1–15.
- Marote, M. L. (2010). Agricultura de Precisión. *Ciencia Y Tecnología*, 10, 143–166. Recuperado de:
http://www.palermo.edu/ingenieria/pdf2012/cyt/numero10/10N_ISEU_CyT10.pdf
- Martínez, J., & Álvarez, G. (2016). Módulo de visión artificial para una aplicación de agricultura de precisión.
- Melrose, J., Perroy, R., & Careas, S. (2015). Argentina-Referente-Mundial-AP. *Statewide Agricultural Land Use Baseline 2015*, 1. <https://doi.org/10.1017/CBO9781107415324.004>
- Morgan Quigley, Brian Gerkey, W. D. S. (2015). *Programming Robots with ROS* (First). Gravenstein Highway North Sebastopol.
- Moss, Andrew Schmitz, C. B. (2015). Mechanized Agriculture: Machine Adoption, Farm Size, and Labor Displacement.

- NVIDIA, T. community N. (2017). Jetson TK1. Recuperado de: https://elinux.org/Jetson_TK1
- Perfetti, J. J., Balcázar, Á., Hernández, A., & Leibovich, J. (2013). *Políticas para el desarrollo de la agricultura en Colombia*.
- Puente, I., González-Jorge, H., Martínez-Sánchez, J., & Arias, P. (2013). Review of mobile mapping and surveying technologies. *Measurement: Journal of the International Measurement Confederation*, 46(7), 2127–2145. <https://doi.org/10.1016/j.measurement.2013.03.006>
- Raspberry PI Foundation. (2017). Raspberrypi. Recuperado de: <https://www.raspberrypi.org/>
- RoboPeak Team. (2014). RPLIDAR Low Cost 360 degree 2D Laser Scanner (LIDAR) System Introduction and Datasheet, 12.
- Rodríguez, A., Rodríguez, M., & Sotomayor, O. (2014). Perspectivas de la agricultura y del desarrollo rural en las Américas: Una mirada hacia América Latina y el Caribe. Recuperado de: <https://www.cepal.org/es/publicaciones/37136-perspectivas-la-agricultura-desarrollo-rural-americas-mirada-america-latina>
- Uniandes. Castilla, R. (2010). Riopaila Castilla S.A. Proyecto Veracruz en Santa Rosalía y la Primavera, Vichada. Recuperado de: https://ceo.uniandes.edu.co/riopaila_web.pdf
- Valero, C. (2002). Agricultura de precisión en EE.UU. y Argentina: dos ejemplos a seguir. *Vida Rural*, 151, 54–56. Recuperado de: http://oa.upm.es/6294/1/Valero_39.pdf
- Valero, C. (2004). Agricultura de precisión en España. E.T.S.I Agrónomos (UPM). Recuperado de: <https://dialnet.unirioja.es/servlet/articulo?codigo=930015>

ANEXOS

Anexo 1: Pasos para la instalación de ROS.

A continuación, se realizan algunos pasos básicos e importantes para lograr una adecuada instalación de ROS. Estos comandos se deben digitar en la terminal de Ubuntu.

1. Configuración del equipo para que acepte el sistema operativo para robots:

```
$sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $ (-SC lsb_release) principal"> /etc/apt/sources.list.d/ros-latest.list'
```

2. Luego se configuran las claves:

```
$sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key 421C365BD9FF1F717815A3895523BAEED01FA116
```

3. Actualizar el sistema mediante el comando:

```
$sudo apt-get update
```

4. Se inicia con la instalación completa de ROS digitando lo siguiente:

```
$sudo apt-get install ros-indigo-desktop-full
```

5. Inicializar *rosdep*, el cual es un comando importante para ejecutar algunos componentes básicos del sistema:

```
$sudo rosdep init  
$rosdep update
```

6. El siguiente comando, por ejemplo, es útil para añadir de manera automática, variables cada vez que se inicia una nueva instrucción.

```
$echo "/opt/ros/kinetic/setup.bash fuente" >> ~ /.bashrc  
source ~ /.bashrc
```

Anexo 2. Configuración del espacio de trabajo

Los pasos a seguir para la construcción del *workspace* se realizan mediante el terminal de Ubuntu y son los siguientes:

Para revisar que se tenga el sistema completo de *setup* se digita:

```
$ source /opt/ros/indigo/setup.bash
```

Ahora si, se inicia con la creación del espacio de trabajo:

```
$ mkdir -p ~/catkin_ws/src  
$ cd ~/catkin_ws/src  
$ catkin_init_workspace
```

Ya se ha creado un *workspace* en el directorio *catkin_ws*, con un subdirectorio *src* para colocar el código del proyecto (Morgan Quigley, Brian Gerkey, 2015).

Se deben crear algunos otros archivos para terminar:

```
$ cd ~/catkin_ws  
$ catkin_make
```

Con las dos últimas líneas se han creado dos carpetas más denominadas *build* y *devel* las cuales se alojan en la carpeta *catkin*. Ha quedado listo el espacio de trabajo.

Anexo 3. Comunicación entre máquinas

Para obtener una comunicación entre los dos equipos, éstos deben tener la misma información de *hostname* e IPs. Para asignar nombre al equipo, es decir el *hostname*, se debe ingresar al terminal de UBUNTU y digital los siguientes comandos:

```
$ sudo gedit /etc/hostname/ /etc/host
```

En la interfaz abierta se puede cambiar el *hostname* por el nombre deseado, luego, se procede a guardar y restaurar el equipo para que tome los cambios realizados.

Para este proyecto se han dado los siguientes *hostname*:

Equipo base: manuelwagner-PC

Equipo embebido: embebido-ubuntu

Una vez se tengan los cambios de *hostname*, se debe verificar la IP de cada equipo, lo cual se realiza con el siguiente comando en terminal:

```
$ ifconfig
```

Con la información de *hostname* e IP de cada uno de los equipos, se procede a agregar las siguientes líneas en el equipo base:

```
$ sudo nano /etc/hosts
```

Se abrirá una interfaz, en la cual se realizarán los cambios:

```
127.0.0.1          localhost
127.0.1.1          manuelwagner-PC
192.168.100.5      manuelwagner-PC
192.168.100.33     embebido-ubuntu
```

Para verificar la comunicación entre máquinas se realiza auto-ping:

```
$ ping embebido-ubuntu
```

Hacer ping al equipo base

```
$ ping manuelwagner-PC
```

Conectarse por *ssh* al equipo base

```
$ ssh manuel@manuelwagner-PC
```

Debe haber un unico ROS master.

Se edita el archivo `~/.bash` de los dos equipos, agregando las siguientes líneas (Se utilizará el ordenador embebido como *master*, pero el *hostname* debe corresponder a cada uno)

Primero en el equipo embebido:

```
## ROS multiple computers configuration
export ROS_MASTER_URI=http://embebido-ubuntu:11311
#export ROS_MASTER_URI=http://manuelwagner-PC:11311
export ROS_HOSTNAME=embebido-ubuntu
```

Luego en el equipo base:

```
## ROS multiple computers configuration
export ROS_MASTER_URI=http://embebido-ubuntu:11311
#export ROS_MASTER_URI=http://manuelwagner-PC:11311
export ROS_HOSTNAME=manuelwagner-PC
```

Para evitar inconvenientes en la comunicación es necesario desactivar *firewalls*, antivirus u otros programas que puedan bloquear puertos o conexiones entrantes.

```
$ sudo ufw disable
```

Anexo 4. Configuración del sensor LiDAR

El paquete necesario para la configuración del sensor láser RPLiDAR se puede descargar de https://github.com/robopeak/rpLiDAR_ros y se descomprime dentro de la carpeta `src` que se encuentra en la carpeta `catkin`, en la raíz de los archivos.

Se comienza con la configuración del puerto donde se recibirán los datos provenientes del sensor RPLiDAR, de la siguiente manera:

En una terminal se escribe lo siguiente:

```
$ ls -l /dev | grep ttyUSB  
$ sudo chmod 666 /dev /ttyUSB0  
$ cd ~/catkin_ws  
$ catkin_make
```

Una vez configurado el puerto, se abre una nueva terminal, en la que se da inicio a la captura de datos:

```
$ source devel /setup.bash  
$ roslaunch rpLiDAR_ros view_rpLiDAR.launch
```

La configuración se ha completado. Automáticamente se abrirá una ventana, en la que se visualizan los datos capturados por el sensor RPLiDAR a través de Rviz.

Anexo 5. Configuración del control de servos.

El control de servos utilizado para el proyecto es el Adafruit PCA9685, de 12 canales y 16 bits, el cual utiliza el controlador JHPWMDriver, especialmente creado para su uso con la tarjeta de desarrollo NVIDIA Jetson TK1. A continuación se describen los pasos a seguir para su configuración.

Instalación del puerto I2C

```
$ sudo apt-get install libi2c-dev i2c-tools
$ sudo i2cdetect -y -r 1
```

El controlador del PCA9685 se configura de la siguiente manera:

```
$ git clone https://github.com/jetsonhacks/JHPWMDriver.git
$ cd JHPWMDriver
$ cd example
```

Luego, en la carpeta 'servoExample.cpp' se quita el comentario de las siguientes dos líneas:

```
// int servoMin = 120;
// int servoMax = 720;
```

Los puntos de partida más apropiados para la exploración inicial son 150/600.

Para construir el paquete con los cambios realizados, se digita en el terminal

```
$ Make
```

Finalmente, para correr el control

```
$ sudo ./servoEjemplo
```

Este último comando permite ver el movimiento de dirección que realizan las 4 ruedas del Rover, lo que evidencia el buen funcionamiento del programa.

Anexo 6. Configuración del IMU

El IMU seleccionado es el Sensor BNO055 de *Adafruit*. Su configuración inicia con la instalación de las herramientas del puerto I2C.

```
$ sudo apt-get install libi2c-dev i2c-tools  
$ sudo i2cdetect -y -r 1
```

En la terminal se observa una matriz de 0x28, la cual es la dirección del IMU por defecto.

Se descarga la carpeta RTIMULib, la cual se encuentra en la dirección <https://github.com/jetsonhacks/RTIMULib.git> y se instala en la carpeta *catkin_ws/src*.

Luego se abre el icono de archivos y se navega hasta la carpeta */RTIMULib/Linux/RTIMULibDemoGL*, en la cual se busca el documento *RTIMULibDemoGL.pro*, se da click derecho sobre él y se abre con *QtCreator* (Previamente instalado). Una vez finalice la compilación hecha por *QtCreator*, se abre un terminal y se coloca `$sudo RTIMULibDemoGL`.

Este último comando, abre una nueva ventana en donde se visibilizan los movimientos que realiza el IMU en los tres ejes.

Anexo 7. Configuración de la carpeta para archivo de datos – rosbag

Para iniciar la configuración del *rosbag* se digita el siguiente comando, el cual genera una carpeta llamada *bagfiles* en la raíz del sistema:

```
$ mkdir bagfiles
```

Luego, es necesario dirigirse a la carpeta creada, para ello se utiliza el comando:

```
$cd bagfiles
```

Se corre el *roscore* para hacer posible la comunicación con el archivo creado.

```
$roscore
```

Una vez en la carpeta, se da inicio a la captura y grabación de los datos. Para el inicio de la grabación es necesario el siguiente comando, en una ventana nueva:

```
$rosbag record -a
```

Para la captura de los datos se da inicio a la configuración del sensor LiDAR.

El archivo generado, con los datos del sensor, tendrá un nombre por defecto con la estructura YYYY-MM-DD-HH-mm-ss.bag, el cual relaciona el tiempo en el cual fueron tomados los datos, sin embargo, el nombre puede ser cambiado por uno más corto.

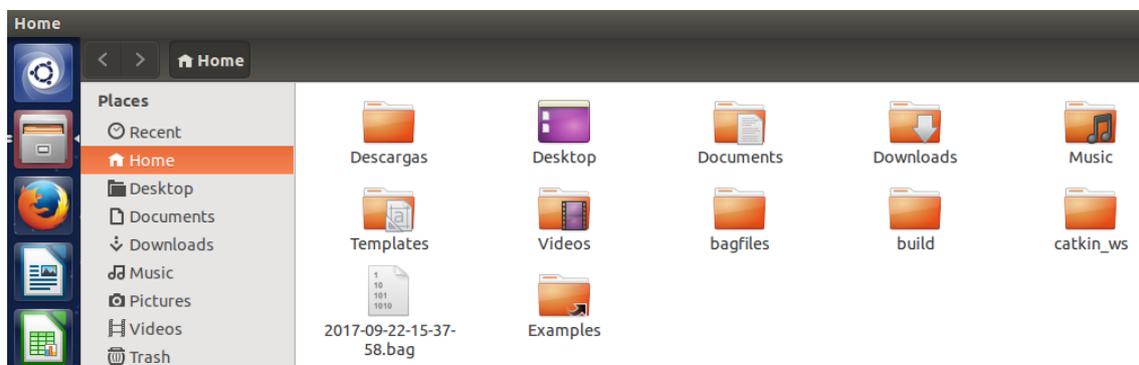


Figura 33. Creación del bag ROS.
Fuente autor (2018).

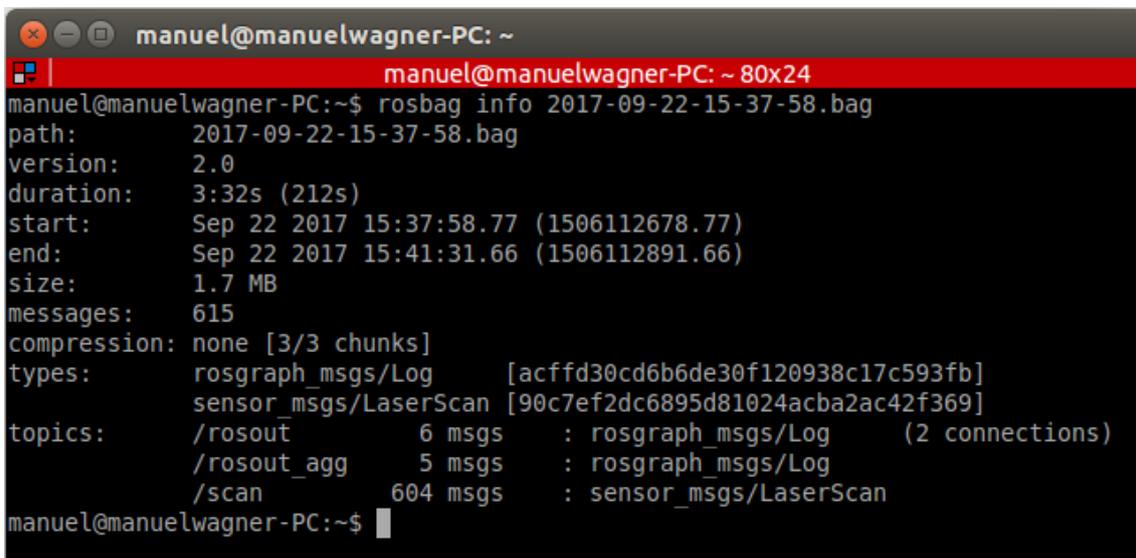
Luego de este proceso, se cuenta con una serie de datos, que pueden ser reproducidos mediante el siguiente comando:

```
$rosviz play --clock foo.bag
```

Es posible verificar los metadatos del archivo generado, mediante el comando:

```
$rosviz play --clock foo.bag
```

En la siguiente figura se puede apreciar la información obtenida en un bag, creado el 22 de septiembre de 2017.



```
manuel@manuelwagner-PC: ~
manuel@manuelwagner-PC: ~ 80x24
manuel@manuelwagner-PC:~$ rosviz info 2017-09-22-15-37-58.bag
path:      2017-09-22-15-37-58.bag
version:   2.0
duration:  3:32s (212s)
start:     Sep 22 2017 15:37:58.77 (1506112678.77)
end:       Sep 22 2017 15:41:31.66 (1506112891.66)
size:      1.7 MB
messages:  615
compression: none [3/3 chunks]
types:     rosgraph_msgs/Log      [acffd30cd6b6de30f120938c17c593fb]
           sensor_msgs/LaserScan [90c7ef2dc6895d81024acba2ac42f369]
topics:    /rosout                6 msgs      : rosgraph_msgs/Log      (2 connections)
           /rosout_agg           5 msgs      : rosgraph_msgs/Log
           /scan                 604 msgs    : sensor_msgs/LaserScan
manuel@manuelwagner-PC:~$
```

Figura 34. Metadatos del bag creado.
Fuente autor (2018).

Anexo 8. Clasificación de datos en formato LAS

Tabla 3. Códigos de clasificación: formato LAS 1.1 a 1.4

VALOR DE CLASIFICACIÓN	SIGNIFICADO
0	Nunca clasificado
1	No asignado
2	Base
3	Vegetación baja
4	Vegetación media
5	Vegetación alta
6	Edificio
7	Punto bajo
8	Reservado *
9	Agua
10	Ferrocarril
11	Superficie de la carretera
12	Reservado *
13	Protector de cable (señal)
14	Conductor de cable (fase)
15	Torre de transmisión
16	Conector de la estructura de cables (aislante)
17	Plataforma del puente

VALOR DE CLASIFICACIÓN	SIGNIFICADO
18	Ruido alto
19-63	Reservado
64-255	Definido por el usuario

Fuente: ArgMap. (2016). Recuperado de:

http://desktop.arcgis.com/es/arcmap/10.3/manage-data/las-dataset/LiDAR-point-classification.htm#ESRI_SECTION1_570719D89812478598FB633D71EBAD06

Nota:

* El sistema de clasificación LAS 1.4 más actual tiene reservados estos códigos (código 8: Clave del modelo/Reservada y código 12: Superposición/Reservado), y los puntos con estas características deberían usar los marcadores de clase correspondientes en su lugar.

- LAS 1.1-1.3 solo es compatible con las clases 0-31.¹

¹ ArcGIS. (2016). Clasificación de puntos LiDAR. Recuperado de:
http://desktop.arcgis.com/es/arcmap/10.3/manage-data/las-dataset/LiDAR-point-classification.htm#ESRI_SECTION1_570719D89812478598FB633D71EBAD06

Anexo 9. Datos – Pruebas de campo

Las pruebas fueron llevadas a cabo en el invernadero de la Escuela Colombiana de Ingeniería, mediante el sensor RPLiDAR, un luxómetro y un portátil. Los datos obtenidos se visualizaron en la aplicación RVIZ.

Prueba 1

Lugar: Invernadero, Escuela Colombiana de Ingeniería Julio Garavito

Fecha: Julio 05 de 2017

Hora: 8:10 am Luminosidad: 8250 lux (nublado)

Hora: 8:40 am Luminosidad: 13570 lux (Nublado)

Observaciones:

- No hay energía eléctrica en el invernadero – Se gestiona el arreglo inmediato por parte del Ingeniero Javier Chaparro.
- Las pruebas permiten evidenciar que el sensor detecta las paredes del invernadero.
- El sensor adquiere gran cantidad de partículas polvo, producto del recorrido en los surcos.

La siguiente figura permite apreciar las señales captadas por el sensor en el primer día de pruebas.

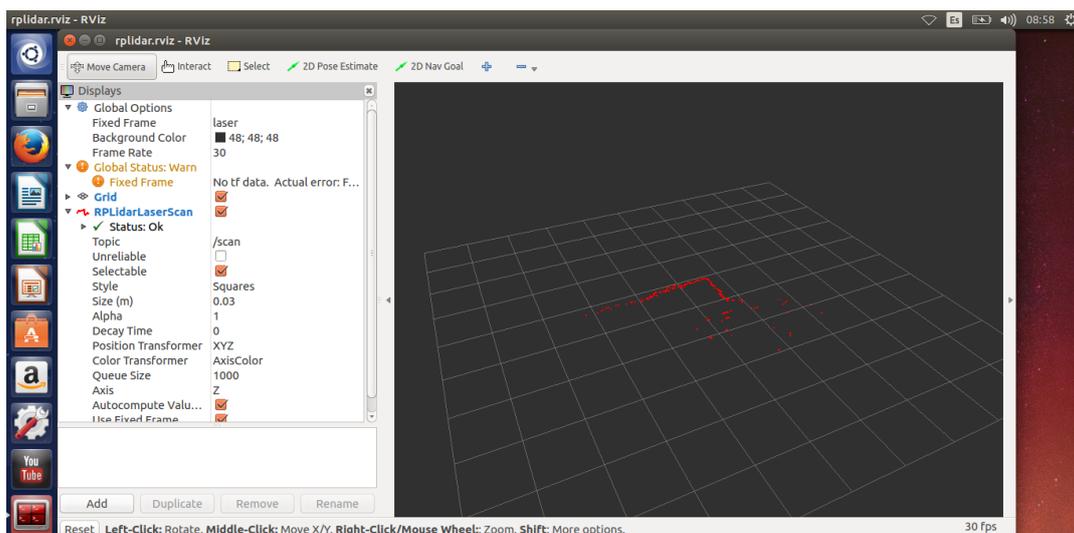


Figura 35. Prueba de campo - Luminosidad en el invernadero.
Fuente autor, 2017

Anexo 10. Jetson TK1 – Buses I2C

Hay seis buses I2C disponibles en Tegra K1, y cuatro están disponibles en el encabezado de expansión. La configuración se puede apreciar en la figura 40 y 41.

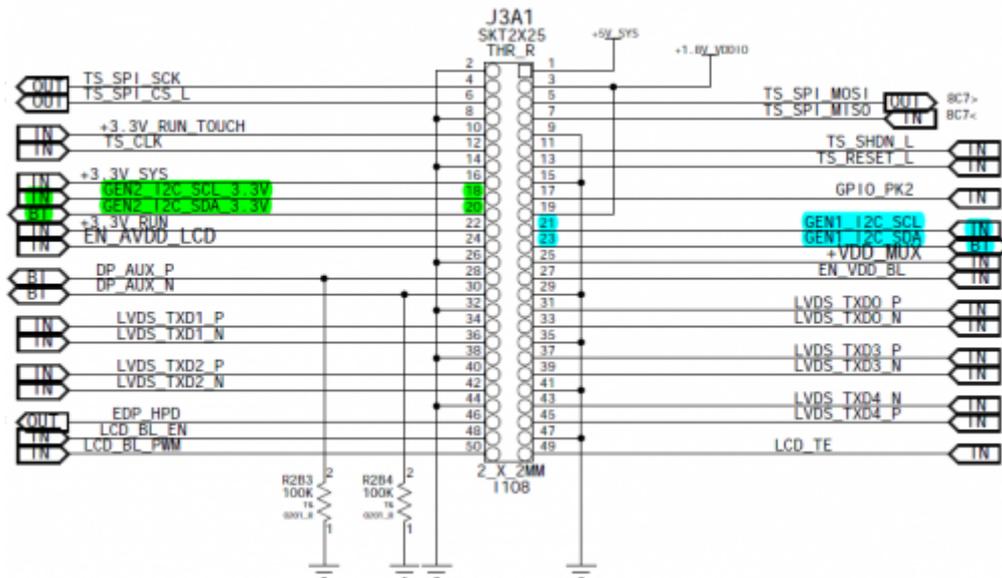


Figura 36. Bus I2C J3A1 2x25 pines - Jetson TK1.
Recuperada de: Jetson TK1. (2016). <https://elinux.org/Jetson/I2C>

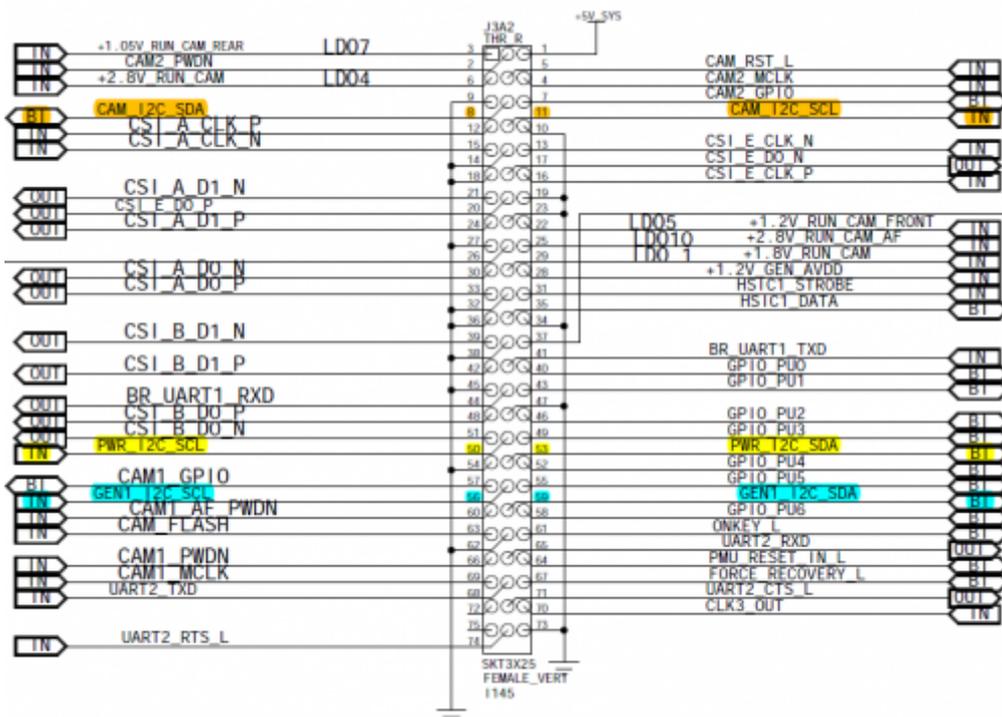
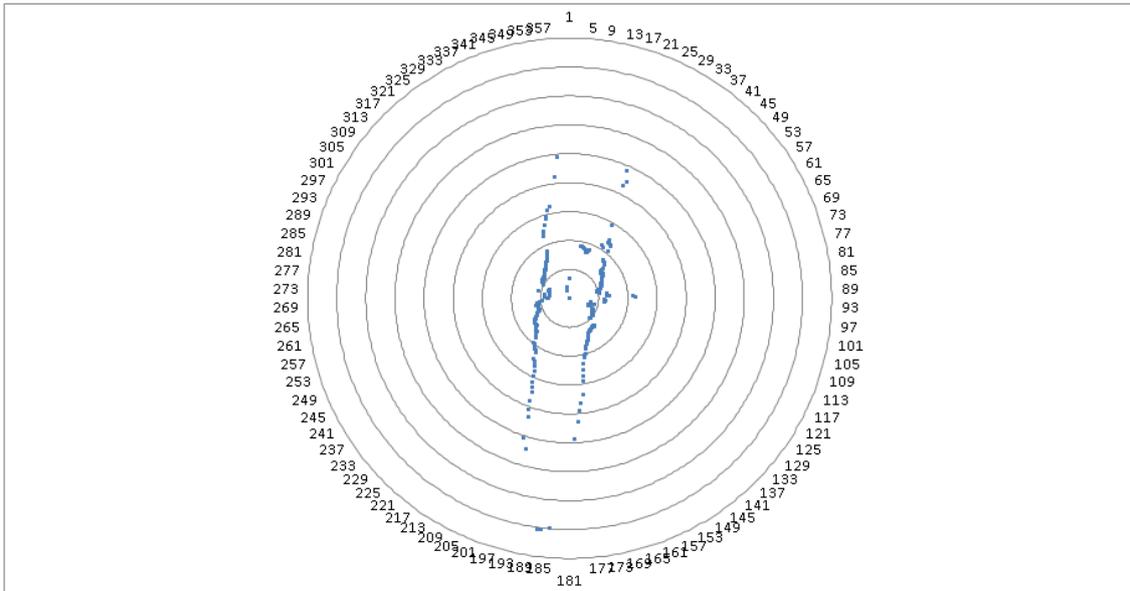


Figura 37. Bus I2C J3A2 3x25 pines - Jetson TK1.
Recuperada de: Jetson TK1. (2016). <https://elinux.org/Jetson/I2C>

Anexo 11. Gráficas en Excel de los datos LiDAR.

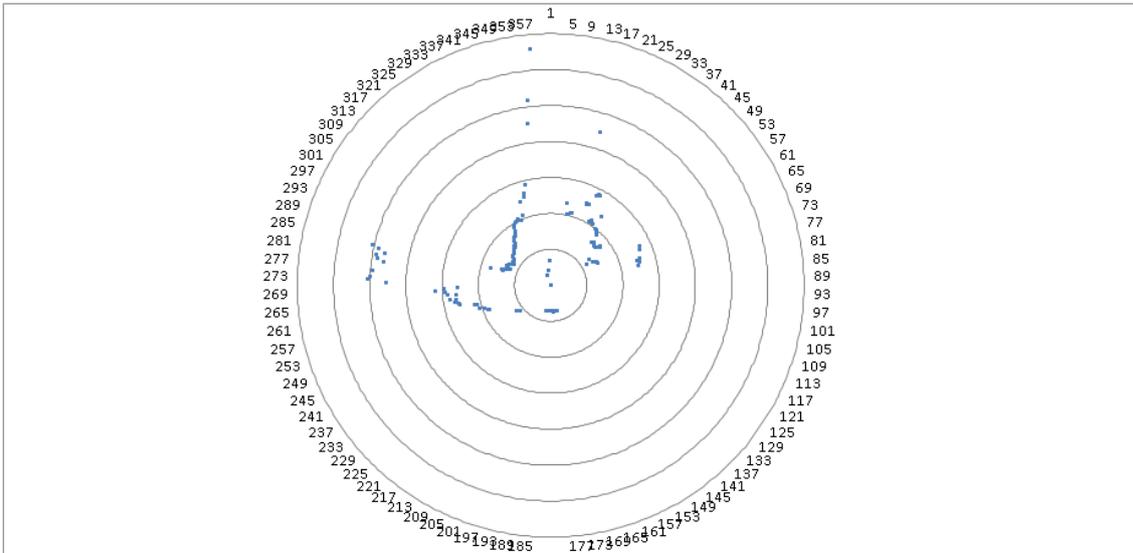
Las pruebas de campo realizadas en el invernadero de la Escuela Colombiana de Ingeniería Julio Garavito, arrojaron datos que fueron almacenados en bags y posteriormente pasados a archivos .cvs. Estos archivos se abren en Excel y se crean un gráfico axial, lo que genera una imagen muy similar a la que presenta RVIZ.

Muestra 1.



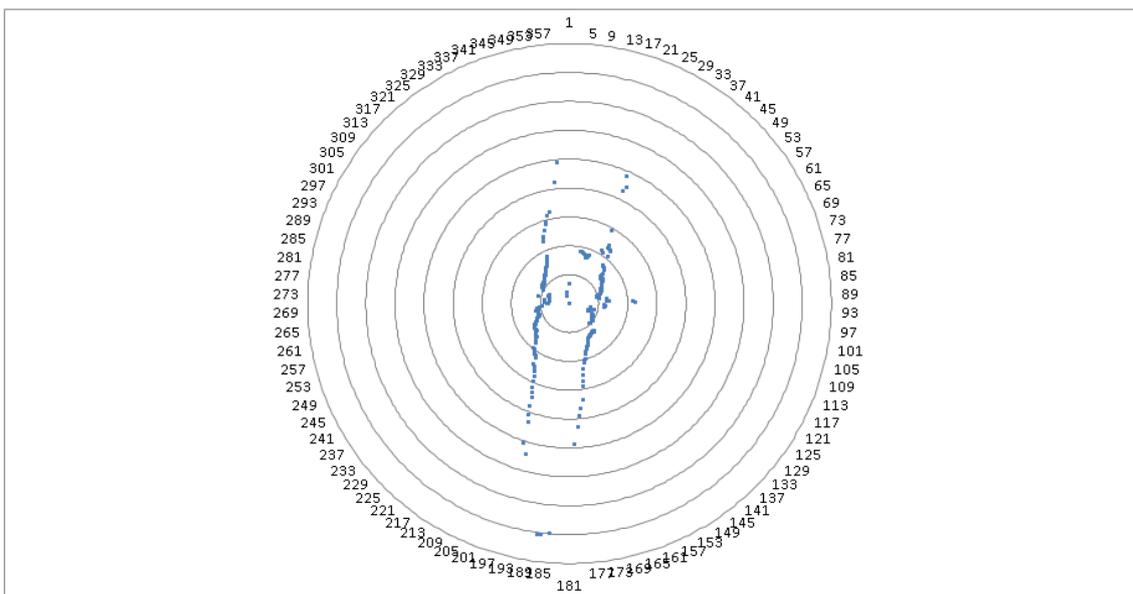
Los datos, representados por puntos, crean dos líneas verticales, que, de acuerdo a lo observado en el invernadero, al momento de la toma de los datos, corresponden a los surcos laterales, donde se encuentra el LiDAR. La concentración de los puntos alrededor del círculo central, es debido a la cercanía con los surcos, aproximadamente 70 cm, es el diámetro del círculo menor.

Muestra 2.



En la muestra 2, el robot se ha desplazado hasta el final del surco. Los datos que se encuentran en la secante horizontal que corta el círculo menor, pertenecen a los datos correspondientes a los puntos que identifican la pared del invernadero, que, por ser plástica, no se obtiene una concentración alta. Tomando el punto (0, 0) como la ubicación del sensor LiDAR, a los 270° (que en la escala de la imagen corresponde a 180°, debido a la configuración de fábrica del sensor, y que será la escala a la que se hará referencia de aquí en adelante), junto al límite del círculo menor, se encuentra el frente del robot, y los puntos generados por la presencia de la pared del invernadero y que se encuentran más cerca, aproximadamente a 30cm del robot.

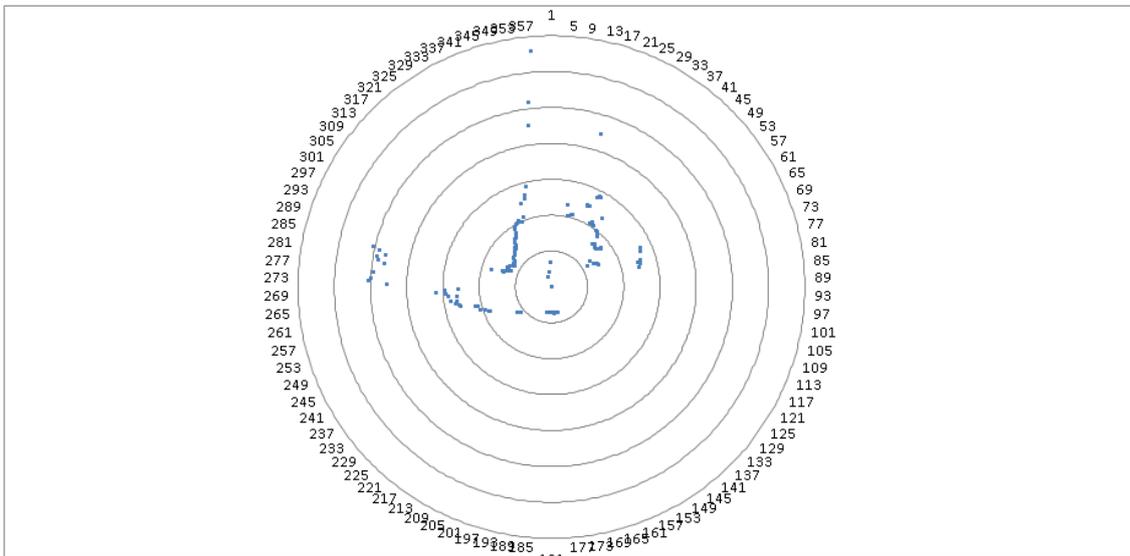
Muestra 3.



En la muestra 3, el robot se encuentra en la parte central de uno de los surcos del

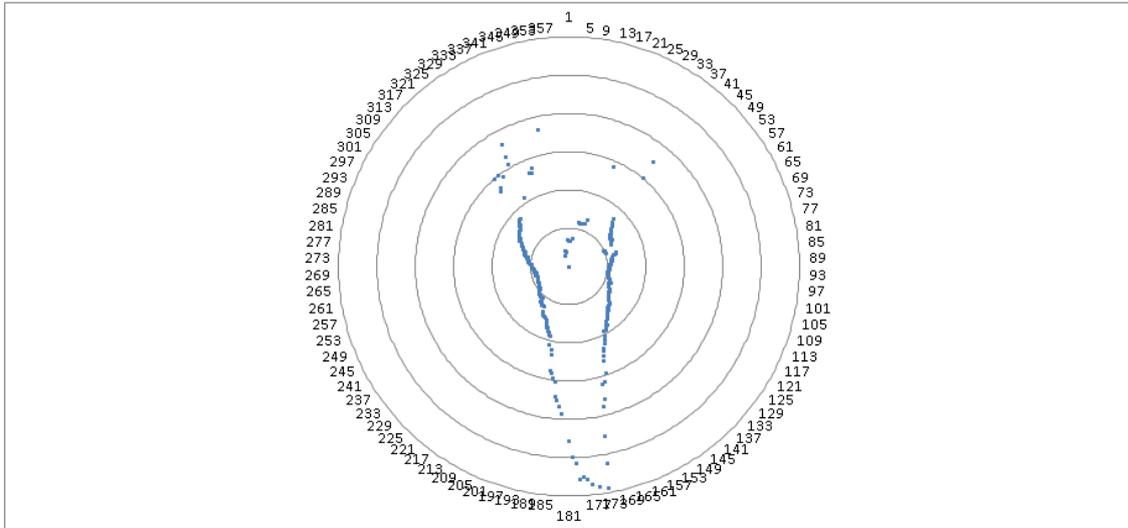
invernadero. Nuevamente se aprecia una gran concentración de puntos cercanos al sensor. Tomando la escala que brinda el gráfico, a los 185° aproximadamente, en el círculo 8, contando a partir del círculo inferior hacia el mayor, se aprecian tres puntos que representan la pared del invernadero. A medida que la distancia aumenta, los puntos se observan más distantes, debido a la definición del sensor.

Muestra 4.



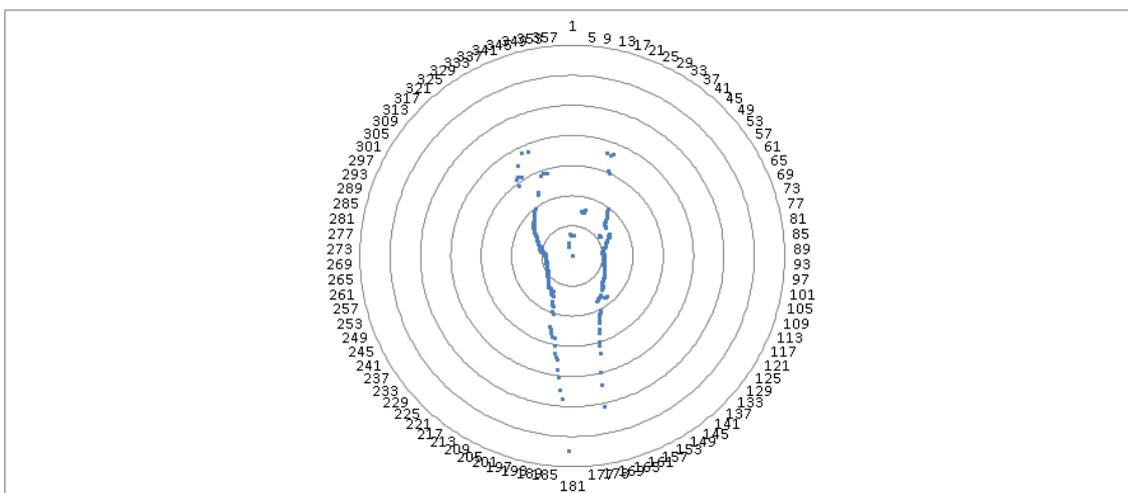
En la muestra 4, el robot se encuentra al final de uno de los surcos del invernadero. Se observa que se forma una línea vertical entrecortada, junto al límite izquierdo del círculo menor, lo que corresponde a uno de los surcos por donde se desplaza el robot. En el límite vertical derecho, del círculo menor, los puntos obtenidos no tienen una figura definida, y, por lo tanto, no se cuenta con información clara. Lo que se analiza, es que, de acuerdo a lo visto en la toma de la muestra, esa parte del surco tiene una altura inferior a la que se encuentra el sensor LiDAR, por lo que los puntos obtenidos en este espacio, corresponden a los tallos de las plantas que se encuentran en esa parte del surco y no al surco mismo.

Muestra 5.



En la muestra 5, el robot se encuentra en la parte central de uno de los caminos del cultivo en el invernadero. Esta muestra permite observar que los datos obtenidos, y que forman dos líneas entrecortadas verticales, tienden a unirse a medida que la distancia aumenta (Aquellos puntos que se alejan del centro de la imagen). Esto da una información errónea, ya que, la distancia entre los surcos, por donde se desplaza el robot, permanece constante; la causa, es que al momento de tomar la muestra, el robot se encuentra en un desnivel del camino, haciendo que el sensor se incline levemente y obtenga datos no de una trayectoria totalmente horizontal, sino, en línea recta con inclinación hacia el piso, lo que ocasiona que los datos se obtengan, a medida que aumenta la distancia con el sensor, de una posición cada vez más hacia la base del surco, en donde la distancia entre surcos es menor.

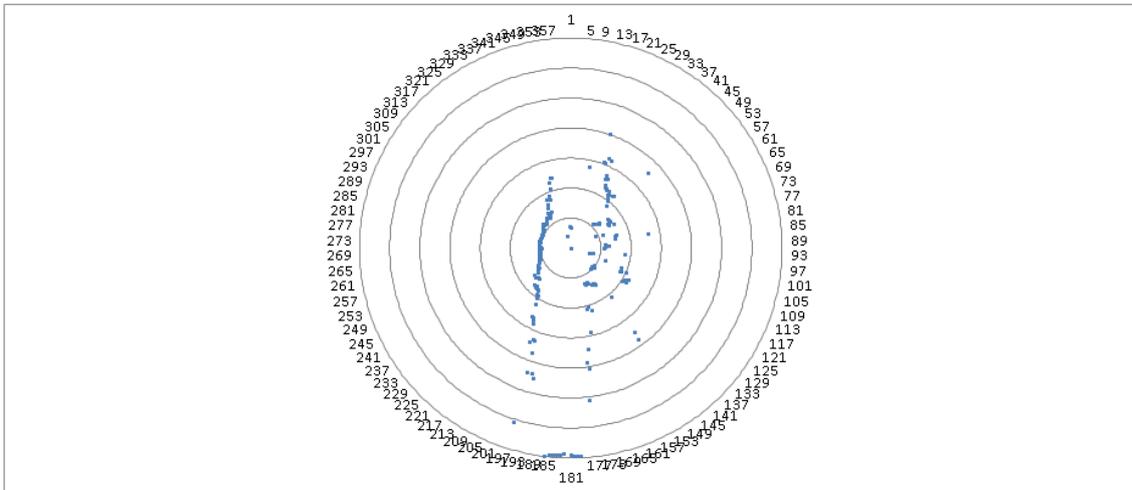
Muestra 6.



En la muestra 6, el robot se encuentra en la parte central de uno de los caminos del

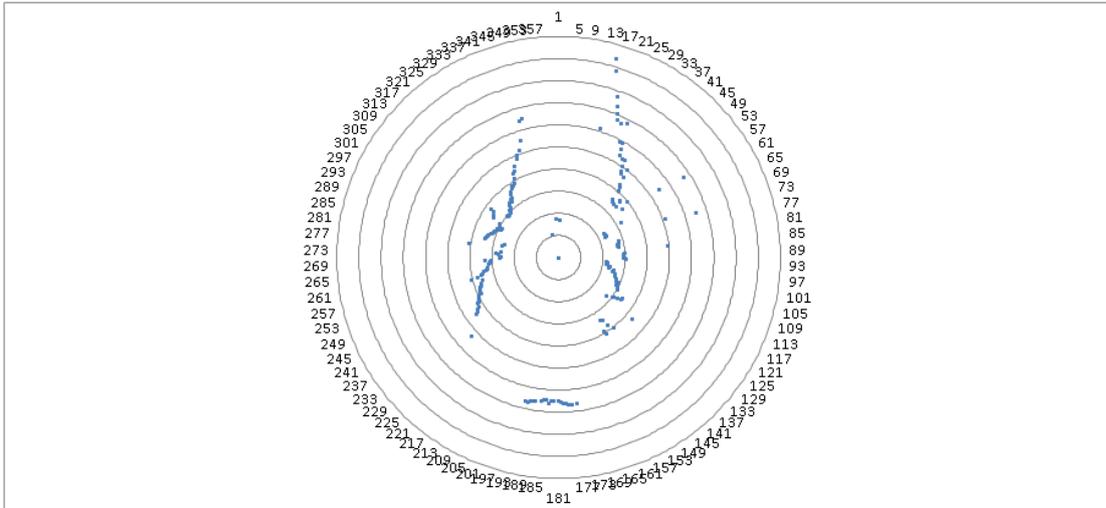
cultivo en el invernadero. Se observa que se forman dos líneas entrecortadas verticales, la cuáles tienden a acercarse a medida que la distancia respecto al sensor es mayor; esto es debido a que el sensor presenta una leve inclinación respecto al eje horizontal, generando a su vez, que los datos obtenidos en la parte superior al círculo central de la gráfica, no tengan una figura clara, y que se debe a que el sensor está captando los elementos por encima de la altura del surco, y que en este caso, corresponden a los tallos de las plantas que se encuentran en la parte posterior a la dirección que lleva el robot.

Muestra 7.



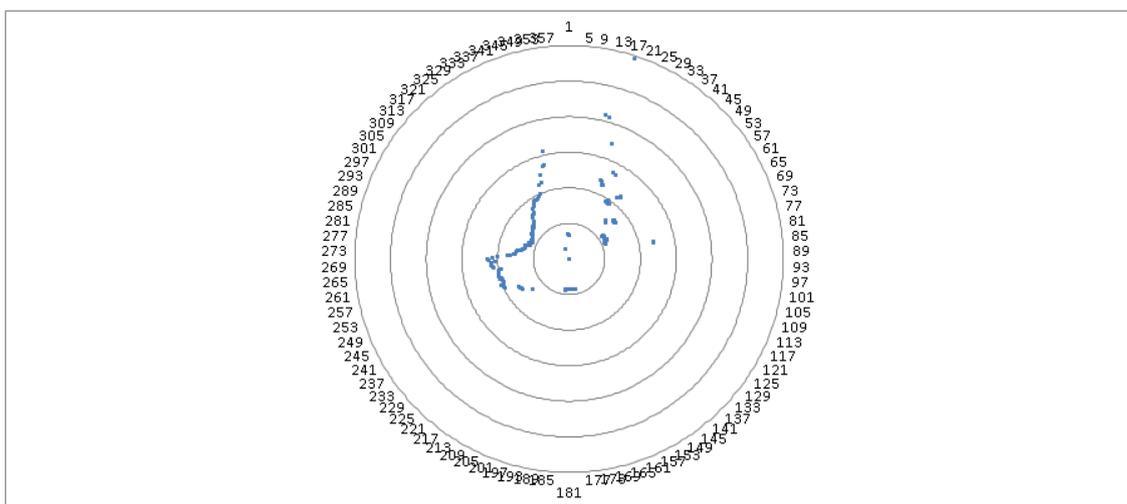
En la muestra 7, el robot se encuentra en la parte central de uno de los caminos del cultivo en el invernadero. El robot tiene una inclinación lateral de aproximadamente 5° , lo que genera que en la parte derecha obtenga puntos con mayor concentración y linealidad, que corresponden al surco, pero en la izquierda, los haces de luz del sensor no chocan contra el surco, sino que, lo hacen contra algunos de los tallos de las plantas, por ello, los puntos se presentan con una mayor desviación.

Muestra 8.



En la muestra 8, el robot se encuentra llegando al final de uno de los caminos del cultivo en el invernadero. Hacia los 180° en la gráfica, se observa una línea horizontal que corresponde a la pared del invernadero, estando a una distancia de 4,2 m aproximadamente. La distancia que se observa, entre surcos, es de aproximadamente de 2m, sin embargo, los puntos que se observan a la derecha de la gráfica y que forman una línea entrecortada vertical, corresponden a los tallos de las plantas, ya que la altura del surco es menor a la altura en donde se encuentran el sensor.

Muestra 9.



En la muestra 9, el robot se encuentra al final de uno de los caminos del cultivo en el invernadero. Siendo el punto (0,0) el lugar en donde se encuentra el robot, en los 180°, según la escala de la gráfica, se observan algunos puntos, los cuales representan la pared del invernadero, y que se encuentra a aproximadamente 40 cm del robot.

los 270°, se observa que la linealidad, de los puntos capturados por el sensor, cambia, esto debido al final del surco, en la que el haz de luz ya no choca con el surco, sino con objetos o espacios ubicados a mayor distancia. Los puntos ubicados entre los 13 y 60 grados respecto al sensor, de acuerdo a la gráfica, corresponden a tallos de plantas en el surco del costado izquierdo del robot; en este espacio el surco presenta una menor distancia, respecto a la altura del sensor.