

DEVELOPMENT OF AN AUTONOMOUS ROVER  
FOR FIELD APPLICATIONS

By

AUSTIN PICKERING

Bachelor of Science in Biosystems Engineering

Oklahoma State University

Stillwater, Oklahoma

2020

Submitted to the Faculty of the  
Graduate College of the  
Oklahoma State University  
in partial fulfillment of  
the requirements for  
the Degree of  
MASTER OF SCIENCE  
May 2022

DEVELOPMENT OF AN AUTONOMOUS ROVER  
FOR FIELD APPLICATIONS

Thesis Approved:

Dr. Ning Wang

---

Thesis Adviser

Dr. Paul Weckler

---

Dr. John Long

---

Name: AUSTIN PICKERING

Date of Degree: MAY 2022

Title of Study: DEVELOPMENT OF AN AUTONOMOUS ROVER

## FOR FIELD APPLICATIONS

Major Field: BIOSYSTEMS ENGINEERING

Abstract: Agriculture is a labor-intensive industry that requires human interactions for even the most basic actions, such as spraying and weeding. Autonomous rovers can be used for many of these tasks. This research focuses on the use of depth sensors to detect the crop and navigate the field. Initial sensor testing was conducted to determine which sensor would be used on the final system. A 2D Laser Range Finder sensor (LiDAR) was chosen for its accuracy and its relatively small data sets. The LiDAR was configured to scan the crop in front of the rover. The rover corrections were determined by an on-board computer running an algorithm written in Python. The system used a PID loop to adjust motor speed and rover heading. Three different tunings were tested. The corrections were sent to a Cube Orange Auto Pilot that allowed an integration of GPS in future works. To validate the system, indoor tests were conducted using rows made from wood and windrows made of Windrow. The data from the LiDAR scans were compared to overhead images to determine the accuracy of the system. The system showed promising results as it maintained an accuracy of 15cm 95% of the time, and an accuracy of 5 cm roughly 80% of the time at 100% power (2 m/s).

## ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Ning Wang for her support and guidance. Without her help I would not be where I am today. Her passion for research has helped push me in my efforts as a researcher.

I would also like to thank my committee members, Dr. Paul Weckler and Dr. John Long, for their assistance throughout my time at Oklahoma State. They have always provided useful insight when I am unsure of what steps to take next.

I would like to thank my fellow graduate students Wade Young and Peyman Nematzadeh for all their assistance. We have spent countless hours in the lab working together and I consider them both great friends. I wish you both the best of luck in your future endeavors.

Lastly, I want to thank my wonderful wife Brittany and my parents Cody and Tracy for all of their support. Without them I truly could not have made it this far.

## TABLE OF CONTENTS

CHAPTER I.....	1
INTRODUCTION .....	1
OBJECTIVES .....	3
CHAPTER II.....	4
REVIEW OF LITERATURE .....	4
2.1 Agricultural robotics and their current applications.....	4
2.2 Sensors for agricultural robot navigation.....	6
2.3 Agricultural Robot Control Strategies .....	9
CHAPTER III .....	12
MATERIALS AND METHODS.....	12
3.1 Equipment Overview .....	12
3.2 Sensor Setup and Calibration.....	20
3.3 Systems Control Methods .....	23
3.4 System Testing and Data Collection.....	28
CHAPTER IV .....	37
Results and Analysis .....	37
4.1 Sensors Results and Analysis.....	37
4.2 Preliminary field tests results.....	39

CHAPTER V .....	49
Conclusions.....	49
5.1 Performance of system.....	49
5.2 Future Work.....	50
REFERENCES .....	51

## LIST OF TABLES

Table 1. The major specifications for the RPLIDAR A1.....	12
Table 2 Major parameters in the specification of the Intel RealSense cameras.....	13
Table 3 Major parameters in the specification of HC-SR04 Ultrasonic Sensors.....	14
Table 4 Major parameters in the specification of MaxBotix Ultrasonic Sensors .....	15
Table 5 Comparison of potential on-board computers.....	15
Table 6 Specifications for Jetson Nano.....	16
Table 7 Mini-PC specifications .....	16
Table 8 Cube Orange Specifications.....	17
Table 9 TF-Mini Range finder specifications.....	18
Table 10 LiDAR Calibration Data.....	37
Table 11 TF-Mini range finder calibration data.....	38
Table 12 The results of the indoor test between the D415 and D435 Camera.....	38
Table 13 The results of the zero-light test between the D415 and D435 Camera.....	39
Table 14 The results of the full sun test between the D415 and D435 Camera .....	39
Table 15 PID values of 60% Speed. ....	41
Table 16 PID Values for 100% Speed .....	41
Table 17 Results of 60% Speed .....	42
Table 18 Results of Wood Test at 100% Speed.....	43
Table 19 Results of Angled Approach at 100% speed.....	44
Table 20 Results of Windrow test at 100% Speed.....	45
Table 21 Average Error for each test.....	48

## LIST OF FIGURES

Figure 1 The data format of the LiDAR sensor (Slamtec, 2022).....	13
Figure 2 The Rover Platform .....	19
Figure 3 Electronic Control Box.....	19
Figure 4 LiDAR Calibration setup.....	20
Figure 5 Range finder calibration setup.....	21
Figure 6 Ultrasonic sensor triangulation setup. ....	22
Figure 7 Test setup for the RealSense cameras.....	23
Figure 8 System Communication Flowchart .....	24
Figure 9 Non-Inverting Summing Amplifier Circuit.....	25
Figure 10 Systems Control Flowchart .....	27
Figure 11 A raw Lidar scan pf peanut canopy (left) and the scan after smoothing (right) .....	28
Figure 12 Mountings of a LiDAR sensor and a range finder sensor .....	29
Figure 13 Mounting of the Cube Orange .....	30
Figure 14 An overhead image of the simulated wood row. ....	31
Figure 15 Processed overhead image of the simulated wood row. ....	32
Figure 16 Angled Approach setup .....	32
Figure 17 overhead image of Windrow test.....	33
Figure 18 Processed overhead image of Windrow test.....	33
Figure 19 Binary filter of a processed point cloud.....	34
Figure 20 Masked overhead image .....	35
Figure 21 Data Analysis Flowchart .....	36

Figure 22 The rover with the LiDAR mounted on the front .....	40
Figure 23 A single LiDAR scan.....	41
Figure 24 A point cloud of the field after being scanned by the LiDAR.....	41
Figure 25 Results of 60% Speed .....	42
Figure 26 Results of Wood Test at 100% Speed .....	43
Figure 27 Results of Angled Approach at 100% speed .....	44
Figure 28 Results of Windrow test at 100% Speed .....	45
Figure 29 Comparison of each Trials Test1 Values.....	46
Figure 30 Comparison of each Trials Test2 Values.....	47
Figure 31 Comparison of each Trials Test3 Values.....	47

## CHAPTER I

### INTRODUCTION

Agriculture is a labor-intensive industry that requires human interactions for even the most basic actions, such as spraying and weeding. Growers are facing a costly, long-term labor shortage. With the global population expected to rise from 7.7 billion to 9.7 billion in just over 30 years, food demand is poised to rise significantly (Gossett, 2019). One way to lower these labor demands is the use of autonomous robots to complete simple repetitive tasks, such as moving produce from field to market or field operations such as spraying or weeding.

Crop specialists and researchers test new varieties of plants and their resistance to disease and other factors that affect plant health and yield. These can be very tedious tasks. In recent years, the use of rovers has helped reduce the amount of labor needed to evaluate the plants; however, a human team is still needed to operate the rover in the field.

There has been research conducted to make autonomous systems for crop evaluation however, most attempts have used platforms that do not fit typical row spacing and are focused more on the traditional object avoidance rather than in field navigation. Industry leaders have created systems that are fully autonomous and meet the size requirements, but they are expensive and hard to modify to fit agricultural research purposes.

Common issues with designing equipment for field applications include size, weight, and temperature ratings. When designing a system for working in a field, certain size requirements must be met. Row spacing is the biggest factor as the wheels and all other components must not interfere with the plants. Weight is also a concern as compaction can occur to the soil if improper weight distribution is used. Meanwhile, an appropriate weight distribution provides moving stability, and adequate traction. The newest issue to arise is the selection of electrical components and their proper installation. Agricultural equipment is used in a variety of conditions and are exposed to extreme weather such as intense heat. Finding components that can withstand heat, cold, humidity, and the dusty environment is a challenge that must be addressed.

There are several products currently being used to solve the design challenges presented in agriculture. The promising solutions are an autonomous fleet-based rover from Fendt (Xaver), the Case IH full size autonomous tractor (Magnum), and the unmanned ground vehicles (UGV) produced by Clearpath Robotics (Husky). The Xaver solved several issues including compaction by using smaller rovers as a fleet-based system. It proved that autonomous systems could work and withstand the agricultural harsh conditions (Fendt, 2017). Magnum had proven larger machines could also successfully be controlled autonomously (Bedord, 2016). The Husky showed that small robotic platforms could be used to allow versatility and let users choose the purpose of their rovers (Maja et al., 2021).

## OBJECTIVES

The purpose of this project is to develop a system to autonomously navigate a field containing row crops. The system should be able to run while completing simple tasks such as field data collections. The development of the system will require a rover platform, a suite of software, and electronics. The specific objectives include:

1. design and construct a rover platform;
2. development of software to operate and control the rover autonomously;
3. design and conduct a set of experiments to test the performance of the rover navigation system; and
4. develop the methods to evaluate the performance of the rover.

The research was conducted with the following steps:

1. Sensor Selection
2. Sensor Calibration
3. Software Development and Testing
4. System Testing
5. System Validation

## CHAPTER II

### REVIEW OF LITERATURE

#### 2.1 Agricultural robotics and their current applications

Given limited land, water, and labor resources, it is estimated that agricultural productivity must increase by 25% to provide for the expected rise in population, while simultaneously limiting the growing pressure that agriculture puts on the environment. Robotics and automation can play a significant role in society to meet its future agricultural production needs (Bergerman et al., 2013). Agricultural robots, which are designed specifically for use in the agricultural industry, have become very popular for tasks such as weeding.

In recent years, there has been a push for autonomous systems in agricultural research. Maja et al. (2021) used a prebuilt Husky platform from Clear Path Robotics, which was configured with Robot Operating System (ROS). The benefit of using the Husky platform was that the manufacturer has done years of testing making ROS work flawlessly and with ease for users. The Husky can also be purchased with several pre-configured and programmed sensors, such as Light Detection and Ranging (LiDAR). Whereas if a user is to develop the system on their own, it could take several years to achieve the desired tasks. Researchers at Kansas State University designed a rover that could navigate a single row of corn and then maneuver through the plants at the head rows to avoid damaging any plants (Schmitz, 2017). While the rover was small, it showed promise for further research to build on. The system used ultrasonic sensors for object

detection and using tracks rather than tires for driving. The rover could only run at a low speed of 0.44 m/s (1mph). Research at the University of Illinois has led to the development of the TerraSentia robot, used for data collection for plant breeding (ACES, 2020).

Work towards autonomy in agriculture is not limited to research purposes. Over the past 20 years, the leaders in agricultural machinery have been working on systems for autonomous navigation. Fendt (2017) has taken a fleet-based approach with their Xaver planter. It is claimed these robots do not use complex sensors. There has also been work done to automate full size machinery. The Case Magnum has shown full size concept tractors using LiDAR and onboard cameras for detecting objects and a GPS for navigation (Bedord, 2016). Kubota has also been working towards autonomous navigation, but with a mid-sized tractor (Coxworth, 2020). The fully electric tractor uses a similar system to the Magnum, but the cameras are also used to make navigation decisions. The Magnum will stop driving if a potential threat is detected and contact the farmer for assistance (Collie, 2016).

John Deere has developed several conceptual tractors, including a compact electric tractor, and an autonomous sprayer. The sprayer system can even refill its tank without an operator to minimize exposure to chemicals (Deere, 2021). More Recently, Deere has released a fully autonomous version of their 8R series tractor and is planning to have actual farmers using this machine in the summer of 2022.

While Caterpillar is not working with field crops, they have been developing autonomous systems for the past 30 years. Caterpillar has approached autonomous machines slightly differently, as they have created a system that is installed on machines typically driven by a human. Caterpillar's system allows any machine to be autonomous, while being more cost effective. The new autonomous machines can communicate with other non-autonomous machines, adding safety to the workplace. Caterpillar utilizes a 3-dimensional (3D) LiDAR on their system to create a map

of the environment. If the system detects a potential obstacle in the path, the machine will stop and wait for the obstacle to leave the road. A GPS is used to keep the machine on the road. Caterpillar claims that their system is fully functional by driving more than 72.4 million kilometers and hauling 2.1 billion tons (Caterpillar, 2022).

There are countless startup companies trying to get their feet in the door of agriculture autonomous navigation. The ROWBOT (Rowbot, MN, USA) is an in-row system capable of planting a cover crop between rows prior to harvest or applying fertilizers. ROWBOT uses a GPS, ultrasonic sensors, and a LiDAR (ROWBOT, 2022). NAIO is a French company that has created an autonomous robot that can remove weeds. The primary sensor used for navigation is an RTK-GPS, while a LiDAR is used for object detection (Technologies, 2020).

## 2.2 Sensors for agricultural robot navigation

With the ever-evolving world of technology, there is an abundance of sensors for navigation and obstacle detection available on the market. The most common sensors are ultrasonic, LiDAR and stereoscopic cameras. A few of these sensors have been mentioned in the previous section. More information on their uses and capabilities will be presented below.

Ultrasonic sensors measure the distance between the sensor and the nearest object by sending ultrasound waves from a transmitter that then bounce off an object and received by the sensor. By using the speed of sound and the time between sending and receiving the signal, the distance can be calculated.

Ultrasonic sensors are a widely used sensor for obstacle avoidance and detection. They can be used in a simple configuration that detects an object in viewing area, or they can be used in a more advanced way illustrated by William H. Kusewich (2007). It was found that if the sensor is placed on a servo oscillating back and forth that a rough location can be found. Zhmud et al. (2018) shows that an HC-SR04 (SparkFun Electronics, CO, USA) is an appropriate sensor for

detecting objects in non-critical situations. However, they found the accuracy is affected by temperature, propagation path, and angle of incidence. They recommend using a laser range finder if a higher accuracy is needed.

LiDAR is a sensor that uses a laser emitter and scanner on a rotating platform to create a scan of the environment. The emitter sends a pulsed light as the platform spins and the scanner receives the pulses bounced off the objects. Each pulse is sent at a known angle therefore the distance of the object can be found. The LiDAR uses the same principles of time of flight as the ultrasonic sensor, but the laser has a much smaller area of focus giving more precise results. LiDAR is one of the most common sensors used for robotics and navigation. The Husky platform can come with either the LMS-111 (SICK, MN, USA) or the VLP-16 3D LiDAR (Velodyne, CA, USA). Clear Path also offers upgrades to several different LiDAR and GPS units (Husky-UGV). Weiss and Biber (2011) found that a 3D LiDAR sensor, 3D fx6 LiDAR (Nippon Signal, Japan), worked well for plant mapping, having a position accuracy of .03m. They state that 3D laser sensors are the most promising and reliable for ag robots to sense the environment.

Meiser et al. (2014) created a system using a custom rover. The rover utilized an ArduPilot board for waypoint navigation. They proposed several sensors, Point Grey BumbleBee XB3 Stereo Vision, Point Grey GigE mono Camera (FLIR, OR, USA) with IR filter, and a TIM51 LiDAR (SICK, MN, USA), which is the same LiDAR used by Maja et al. (2021)

Stereoscopic cameras are another popular option for navigation. Stereoscopic cameras use stereo vision, a method that uses triangulation to calculate the distance from the camera to an object. Two lenses are used on the camera and the distance between the two is known. By finding the distance of the object to each lens, a triangle is created, and the location of the object can be found.

One of the more popular options is the Intel RealSense D4xx series (Intel, CA, USA). The RealSense cameras are used on rovers such as Simple Autonomous Wheeled Robot (SAWR), which Intel recommends as a platform. The RealSense cameras are well documented and tested by the manufacturer. There have been numerous tests performed to find the limits of both the d415 and the d435 cameras. It has been found that both cameras are good options for an autonomous rover, but the d415 gives a higher resolution image than the d435 (Tadic et al., 2019). The RealSense cameras are also customizable to help with exposure when being used outdoors (Anders Grunnet-Jepsen, 2020). Intel also offers the T265 camera which is a double fish eyed lens capable of simultaneous localization and mapping (SLAM). Bayer and Faigl (2019) found the T265 to be competitive with other sensors on the market, and to have a lower computational need. Alapetite et al. (2020) confirms that the T265 is a competitive camera, as well as that the ZED cameras from Stereo Labs (San Francisco, CA, USA) are another good option for navigation.

There are plenty of sensors and configurations available, but some stand out more than others. Ultrasonic sensors are cheap and require little computational power, but they cannot detect an actual location. LiDAR can be obtained for a variety of prices depending on if a 2 dimensional (2D) or 3D sensor is needed. A 2D sensor offers a high-resolution scan that can offer plenty of useful data, is much cheaper than a 3D sensor, and requires less computational power. However, a 3D sensor can give much more data and provide a 3D image of the rover's surroundings. A stereoscopic camera has the added benefit of using RGB in future applications, while being in the same price range as a LiDAR.

The choice among sensors can also be made by evaluating the methods in which the data can be processed and used for controlling the rover. The 2D LiDAR data is much easier to process as the data is received in a single row, rather than a large array like the stereoscopic cameras. However,

the cameras data can be processed in more ways which can be seen in the agricultural robot control strategies section.

### 2.3 Agricultural Robot Control Strategies

There has been much discussion in academia about the best control strategies for autonomous and semi-autonomous robots and rovers in agriculture. Shalal et al. (2013) found that it depended on the system as SLAM was ideal for estimation and tracking. A Hough Transfer could also be used followed by k-means for segmentation which could be useful for detecting row crops. It was also found that while SLAM could be effective, it was not ideal for highly non-linear systems, therefore Extended Kalman Filter (EKF) should be used. Shalal et al. (2013) determined that a proportional integral derivative controller (PID) should not be used, but a Fuzzy Logic (FL) should. Zhang et al. (2019) confirmed both previous claims on a robot using a gyroscope as the sole sensor for rover location.

Like the Hough transform used for image segmentation, Hugh, Saturation and Value (HSV) could be used (Kanagasingham et al., 2020). HSV combined with Inverse Perspective Mapping (IPM) removed the distortion in an image and created a plane for row detection. Linear regression was then used to find the line of best fit. This method was tested on a rover in rice fields with three ranges of weed density to determine its accuracy. The low weed density trials proved successful with an average deviation of 45.9 mm. The high weed density fields had a much larger deviation of 94 mm, but the robot still navigated the fields without damaging the plants.

Line extraction from a point cloud is a similar method to those previously mentioned. Malavazi et al. (2018) proposed a new method based on the previous PEARL algorithm from Isack and Boykov (2012). The resultant RUBY algorithm was found to work well for LiDAR based navigation, but still had limitations. However, these limitations were primarily regarding the sensors used. They proposed that future works should use a 3D LiDAR. Even with the limitations

of the LiDAR, it was found when using simulated rows, that the new RUBY algorithm had a 100% success rate with an average error of 1.15% while the previous PEARL algorithm had an 80% success rate with an average error of 3.53%. Error was defined by total distance traveled regarding the perfect path.

Iqbal et al. (2020) performed a study using a simulated robot equipped with a 2D LiDAR using ROS. The purpose of the rover was to navigate simulated cotton fields using the Gazebo simulation package. The system tested control strategies using the LiDAR in different configurations. While the LiDAR was 2-dimensional, multiple scans were used to create a point cloud. The point cloud was then down sampled and voxelated before being split into the left and right rows. Once the rows were separated, a radius outlier filter was applied and a RANSAC algorithm was used to create a line of best fit. The program then compared the angle of the two lines to create a path for the rover. The simulation proved to be successful with less than 1% drift from the ideal path.

It has been found that LiDAR data could contain high amounts of noise causing variance in readings. One method of smoothing the data was the use of a Savitzky-Golay filter. Singh et al. (2020) found that the Savitzky-Golay filter was a good light weight solution for smoothing LiDAR data collected by a drone.

Along with the standard controls methods used by a standard robot, rovers being used in the field have interactions with various soils and ground materials that can affect steering controls.

Research found that a canonical digital controller could help with variance in lateral position of a skid steer style rover (Fernandez Guzman et al., 2018). The results showed that the system never had an error of more than 2cm. However, they recommended adding heading and speed sensors to the same controller.

A further need for field-based robots includes path planning and waypoint generation. A system using an ArduPilot board can have a predefined mission loaded using Mission Planner (Moeller, 2020). Mission Planner is an open-source ground control station that allows the user to change parameters, and to plan and start missions. Using an ArduPilot board, Moeller was able to guide a robot effectively to go to specific plants in a field to check for disease. If a fleet of rovers is to be used, an added level of complexity is added. Conesa-Muñoz et al. (2016) proposed a method for determining the optimal route of each rover. The algorithm focused primarily on sprayer applications and having the knowledge of where high weed populations were.

## CHAPTER III

### MATERIALS AND METHODS

#### 3.1 Equipment Overview

##### *3.1.1 Range sensors*

Three types of sensors were evaluated to determine the best fit for the project. A Light Detection and Ranging (LiDAR) sensor, two stereoscopic cameras, and two types of ultrasonic sensors. Each of these sensors underwent initial testing to determine the best fit for the project.

##### *LiDAR*

The LiDAR used in this study was an RPLIDAR A1 (Slamtec, Shanghai, China). The LiDAR was used to detect the location of the rover with respect to the plant rows. Table 1. The major specifications for the RPLIDAR A1.

Table 1. The major specifications for the RPLIDAR A1

Distance	.15 -6 m
Distance Resolution	.5 mm
Angular Range	0-360 Degrees
Angular Resolution	1 Degree
Sample Duration	.5 ms
Sample Frequency	2000 Hz
Scan Rate	1-5.5 Hz

The LiDAR communicated with an onboard computer via USB through a USB to Serial converter that came with the sensor. The format of the data can be seen Figure 1 (Slamtec, 2022).

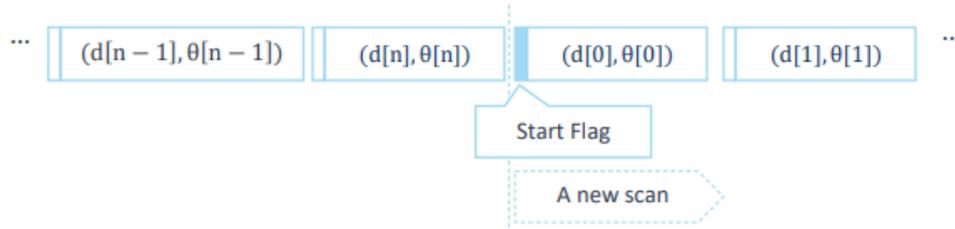


Figure 1 The data format of the LiDAR sensor (Slamtec, 2022).

### *Stereoscopic Camera*

The stereoscopic cameras tested were the Intel RealSense D415 and D435. The RealSense Depth cameras (RealSense) performed with a principle like that of the LiDAR, but a depth image would represent multiple scans of the LiDAR. Using an image instead of a line also allowed for predictive corrections based on the angle of the rows relative to the rover. Using images for predictive navigation was similar to the work of Shalal et al. (2013)) and Kanagasingham et al. (2020), except depth was being used instead of a traditional RGB image. The parameters of both cameras can be seen in Table 2.

Table 2 Major parameters in the specification of the Intel RealSense cameras.

	D415	D435
Min Depth (m)	.3	.2
Max Depth (m)	10	10
Sampling Rate (frame/second)	6, 15, 30	6, 15, 30
Shutter	Rolling	Global
Max Resolution of depth image (pixels x pixels)	1280x720	1280x720
Max Resolution of RGB image (pixels x pixels)	1920 x 1080	1920 x 1080
Depth Field of View	>70°	>90°

### *Ultrasonic Sensors*

Two ultrasonic sensors were tested, HC-SR04 (ElecFreaks, New Delhi, India) and a MaxBotix MB7380 (MaxBotix Inc, MN, USA). Ultrasonic sensors are a good choice as they are simple and low-cost sensors, especially for outdoor environment. They could be used to detect the distance from a row on either side of the rover. They could be installed facing the center row passing through the rover, as well as facing the rows on either side of the rover. The difference in each distance could then be compared. If one row was closer, then corrections on motor drive could be made. The specifications of both sensors can be seen in Table 3 and Table 4.

Table 3 Major parameters in the specification of HC-SR04 Ultrasonic Sensors

Parameter	Value
Working Voltage	DC 5 V
Working Current	15 mA
Working Frequency	40 Hz
Max Range	4 m
Min Range	2 cm
Distance resolution	.3 cm
Measuring Angle	15 degrees
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion

Table 4 Major parameters in the specification of MaxBotix Ultrasonic Sensors

Parameter	Value
Voltage	DC 3 – 5.5 V
Frequency	10 Hz
Max Range	10 m
Min Range	20 cm
Distance Resolution	5 mm
Measuring Angle	20 degrees
Output Signal	RS232/TTL Serial, Analog

### 3.1.2 Central Control Unit

The system included an on-board computer for handling data collection and processing, as well as sending the motor corrections. Table 5 shows the comparison of the single-board computers that were considered. The needs of the system included low power consumption, with a GPU and USB 3.0 ports. The Jetson Nano was chosen as it met the needs of the system and was cost effective.

Table 5 Comparison of potential on-board computers

	Raspberry Pi 4	ODROID-XU4	Jetson Xavier	Jetson Nano
Processor	1.5GHz	2GHz	1.4-1.9GHz	1.43GHz
RAM	8	2	8	4
USB 3.0	2	2	3	4
Power	15 W	20W	15W	10W
GPU	none	6 cores	384 core Volta with 58 tensors	128 core maxwell
Price	\$100	\$80	\$400	\$100

### *Jetson Nano*

The on-board computer being used is a Jetson Nano Developer Kit (NVIDIA, CA, USA). The Jetson Nano is a single board computer with an added graphics processing unit (GPU). The specifications can be seen in Table 6.

Table 6 Specifications for Jetson Nano

Jetson Nano	
Processor	1.43GHz Arm Cortex A57
RAM	4 GB 1600MHz DDR4
Power	10W
GPU	128 core Maxwell GPU 921Mhz
Price	\$100

### *Mini-PC*

After problems were found with the Jetson Nano, a mini-PC was used as a replacement. The PC was running Ubuntu 20.04 and the specs can be seen in Table 7.

Table 7 Mini-PC specifications

Model	Technovare PN-TPC2W7A
CPU	Intel G620 2.6GHz
RAM	8GB DDR3
Graphics	Integrated
Storage	500GB SSD

### *Cube Orange Autopilot*

A Cube Orange Auto Pilot (ProfiCNC, Australia) was also mounted on the rover. The Cube was used to communicate motor signals from the on-board computer to the motor driver. It could run GPS guided missions, manual RC control, and receive commands from the onboard computer to do sensor based autonomous driving. The specifications of the Cube can be seen in Table 8.

Table 8 Cube Orange Specifications

Cube Orange + ADS-B	
Processor	32 Bit Arm Cortex M7 400Hz, 1MB Ram, 2MB Flash
Power	4.1-5.7V/ 2.5A
Accelerometer/Gyroscope	ICM 20649 30G/ ICM20948/ ICM20602
Compass	ICM20948
Barometric Pressure Sensor	MS5611 x2

The Cube communicated through the Micro Air Vehicle Link (MAVLink) protocol. The on-board computer sent the Cube commands through a USB to a Serial converter. The Cube could also be controlled over Telemetry using a 915MHz radio allowing the operator to monitor the system from a laptop or handheld device.

### *3.1.3 System Evaluation Sensor Package*

To test the performance of the developed autonomous platform, a laboratory test setup was established. A single-point, laser range finder and a GoPro camera was used.

#### *TF Mini Range Finder Sensor*

A TF Mini range finder was used to collect the travel distance of the developed system during testing. The range finder was not part of the autonomous navigation, but for the system validation. The TF Mini is a single-point, time of flight (TOF) LiDAR sensor. It was able to

report distances from the start to the end of a test, which would help turn the 2D LiDAR scans into a 3D point cloud. Its specifications are listed in Table 9.

Table 9 TF-Mini Range finder specifications

Range	.1m-12m
Accuracy	5cm
Resolution	5mm
Ambient Light Immunity	70klux
Frame Rate	1-1000Hz
Light Source	LED
Central Wavelength	850nm
Field of View (FOV)	3.6 degrees
Supply Voltage	5V
Average Current	110mA
Peak Current	140mA
Power	550mW
Communication	UART, I2C, I/O

#### *GoPro Camera*

A GoPro Hero 5 Black (GoPro Inc., CA, USA) was used to collect images of the rows during testing. The camera was set in linear mode to avoid distortion of the image. The Hero 5 has a maximum resolution of 4000×3000 and has Bluetooth capabilities so it can be controlled remotely.

#### *3.1.4. Rover Platform*

A rover platform was constructed which used two electric motors in a differential steering configuration. One motor was setup on each of the rear wheels and the front wheels were caster style wheels that were free to spin 360 degrees. The system also had a two 100-watt solar panels to help maintain the battery levels during use, and a GPS. The platform can be seen in Figure 2.

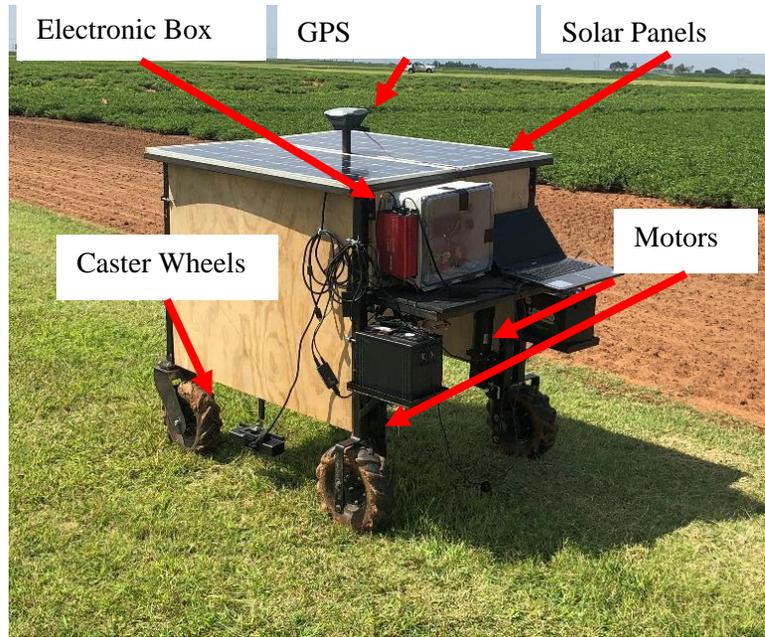


Figure 2 The Rover Platform

The electronic controls were mounted inside a plastic case to help protect them from dust or other debris. The contents of the case included a 24 V solenoid, a 12 V solenoid, a Sabertooth motor driver, a solar panel power inverter, and a Jetson Nano single board computer. The electronic control box is shown in Figure 3.

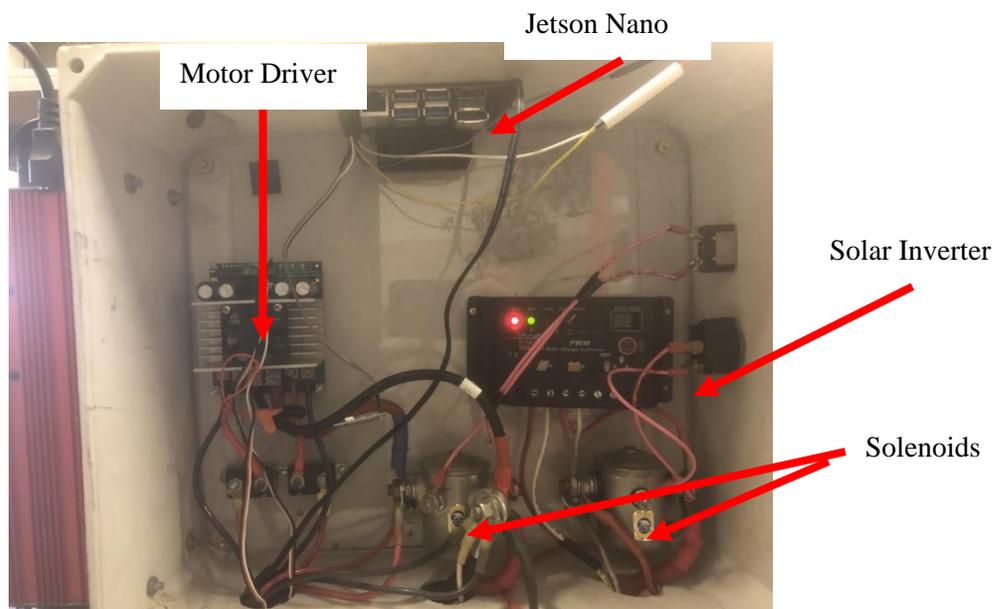


Figure 3 Electronic Control Box

The rover platform had an RC controller that allowed an operator to steer the rover through the field. The rover controller was a Spektrum DXE paired with an AR620 receiver. (Horizon Hobby, Champaign, IL). The receiver of the controller was connected to a Sabertooth 2x60 Amp motor driver (Dimension Engineering, Hudson, Ohio).

### 3.2 Sensor Setup and Calibration

#### 3.2.1 LiDAR Calibration Setup

Figure 4 shows a setup for the LiDAR calibration. Distance was measured from the sensor to a test stand with adjustable height. The test consisted of three repetitions at 3 heights (55cm, 70cm, and 80cm).

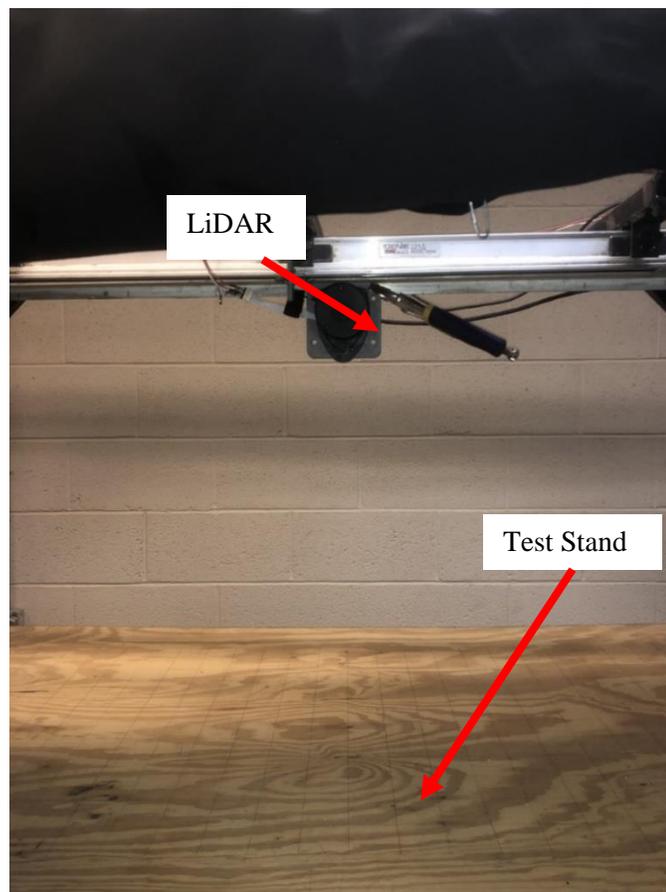


Figure 4 LiDAR Calibration setup.

### 3.2.2 Range Finder Calibration Setup

Because the range finder would scan at distances further than allowed by the test stand, the range finder was setup facing a large flat target at 140cm, 280cm, and 570cm. A measurement was recorded every 15 seconds over 15 minutes, and three trials were run. The setup can be seen in Figure 5.

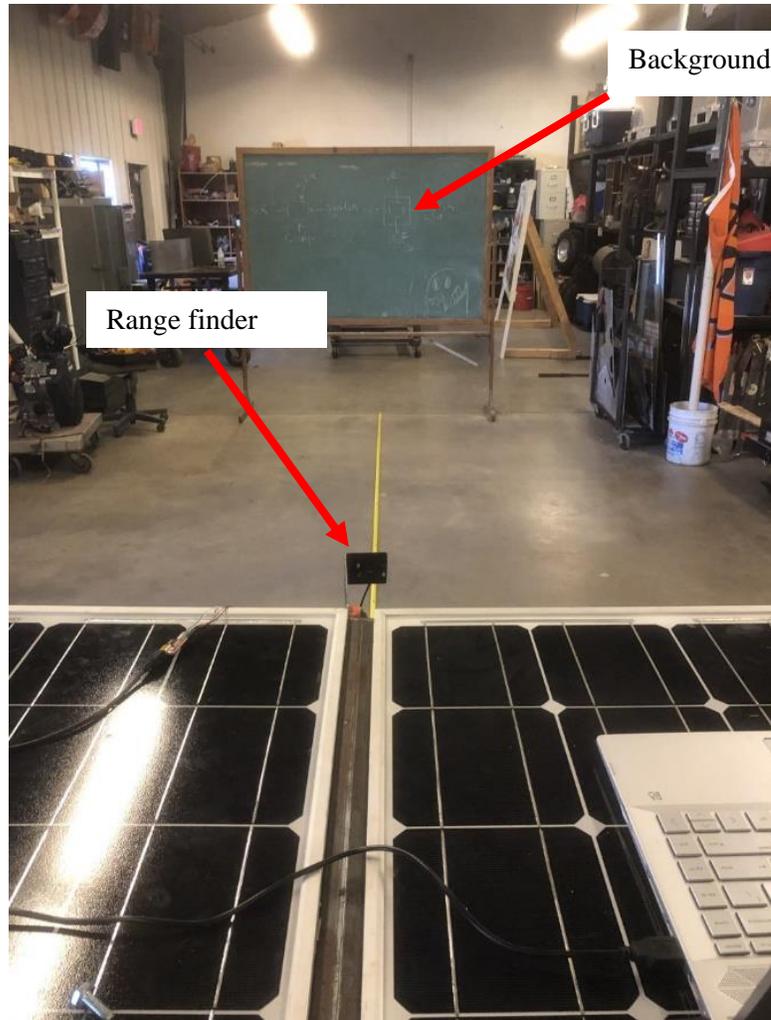


Figure 5 Range finder calibration setup.

### 3.2.3 Ultrasonic Calibration Setup

The HC-SRO4 ultrasonic sensors were tested to determine their accuracy at the measurement of the location of an object using triangulation. Two sensors were used in this experiment. One was used in the traditional manner, while the second had the emitter covered. The sensors were setup seventy centimeters from each other and slowly rotated until a reading was found. The setup can be seen in Figure 6.

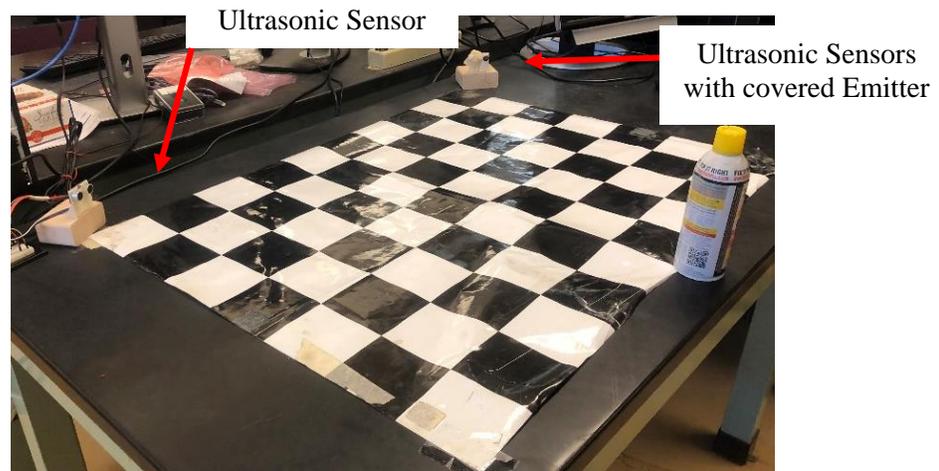


Figure 6 Ultrasonic sensor triangulation setup.

### 3.2.4 Stereo Camera Calibration Setup

Two Intel RealSense cameras, D415 and D435, were tested in the same manner using their default settings. The cameras were attached to a beam facing a flat target and readings were taken at four heights (.5m, .7m, .9m, and 1.1m). The test was run three times for each height. Figure 7 shows the setup.



Figure 7 Test setup for the RealSense cameras

### *3.2.5 Methods of data analysis for the calibrations*

The LiDAR and range finder data were analyzed using Excel. The average, standard deviation, and the maximum and minimum values were found for each distance test.

The RealSense Camera data was also analyzed using Excel. The average, standard deviation, number of dropped pixels, and outliers were all found. The number of dropped pixels provided a clear picture of which camera performed better and stable under each condition.

## 3.3 Systems Control Methods

### *3.3.1 System Communication Protocol*

To communicate between the Cube and the on-board computer, Pymavlink was used. Pymavlink is a python implementation of MAVLink. The protocol for communication between the devices did not monitor messages and no checks were used to determine if messages were being acknowledged. To prevent any failures or safety concerns, the on-board computer would request the Cube to send heartbeat messages periodically to ensure it was still connected. If the heartbeat

was not found, the Jetson Nano would be able to kill power to the motors, through a relay, to prevent any accidents. The relay was controlled by a signal from the Jetson Nano, which closed the coil. Therefore, if the Jetson were to ever shutdown for any reason the relay would lose the signal and the coil would open. Figure 8 shows a flowchart of the control system.

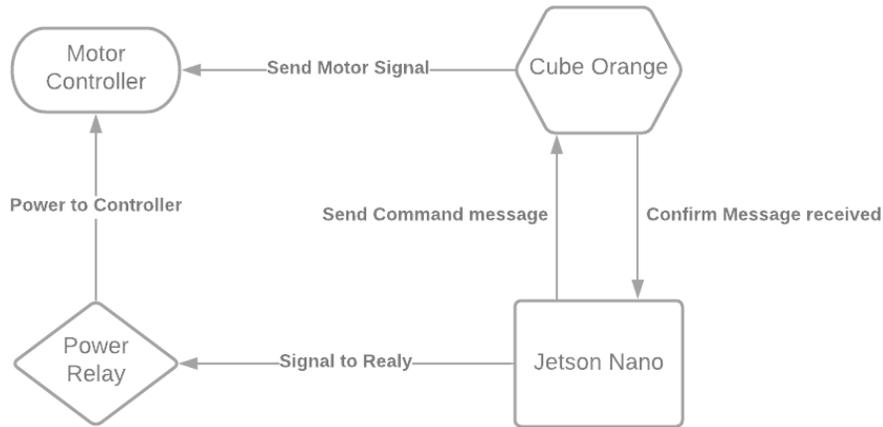


Figure 8 System Communication Flowchart

### 3.3.2 Rover Control Methods

#### *Control Commands*

Several methods were available to send corrections to the Cube, but eventually the Manual Control command was chosen for its simplicity and its ability to work without GPS. The ability to work without GPS allowed the system to be used without premade missions, and without the need of plant maps or other GPS coordinates. Another benefit for Manual Control was that the system could have a GPS added when needed, and with minimal changes, the system would have the ability to perform GPS guided missions with the added functionality of the sensors to help guide the rover.

Manual Control was a simple function included in the Pymavlink library. The purpose of Manual Control was to give the user a way to control the system output with a non-RC control. The available commands were pitch, roll, thrust, yaw, and two optional button inputs.

By using Manual Control, the use of special circuitry was eliminated and the ability to add a joystick or keyboard control in the future was much easier. Another option that was explored was RC Channels Override. The use of the override method was possible, but it would either require the loss of the RC controller, or special circuitry for signal mixing. The use of overrides was the initial choice, but it was found that the channels of the Cube with predefined outputs could not be controlled with this method. As a result, signal mixing was explored. The predefined channels for the left and right motors would still be connected to the RC Controller, and two undefined channels would be controlled by the on-board computer. The outputs from the Cube would then be sent to a non-inverting summing amplifier as seen in Figure 9. However, it was decided that Manual Control would be the better option as it lacked additional components that could fail.

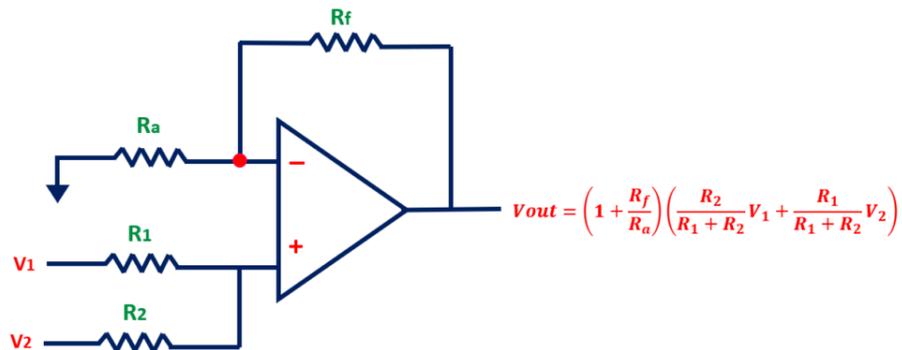


Figure 9 Non-Inverting Summing Amplifier Circuit

### *Control Strategy*

The system started with a pre-arm check, where it verified all connected sensors, and the Cube was arm able. Once these checks were finished, the rover was Armed. “Armed” means the Cube could send signals to the motors. While “Disarmed” the Cube would ignore all motor related commands. Once Armed, the main loop began and would not stop unless an error was detected, such as a sensor disconnecting, a user input was received, or the LiDAR depth timeout was detected.

The rover controls were based on the LiDAR data received. The LiDAR was mounted at the center of the front frame of the rover at a height of 86 cm. The scan data were read through the RPLiDAR package provided with the LiDAR and stored in an array as scan angle and depth.

Each scan data array was then run through a function which turned the scan array into useable data. The data was converted from angles in degrees in polar coordinates to ground location and height of scan in Cartesian coordinates. The converted data was then run through a Savitzky-Golay filter to smooth the reading and help determine the center of the plant canopy. Once the center of the scan was found, it was returned from the function.

The returned value from the previous step was then run through a PID loop. The PID loop determined the ideal rpm for each motor to either correct or maintain the current course. The PID loop returned the updated rpm values, which were stored in a MAVLink message and sent to the Cube.

It was important to note that the previous steps were only implemented if the new value had a height reading greater than the threshold set by the user. For the testing performed, the height threshold was 70 cm. If the height was less than 70 cm, the speed command was set at 15% forward speed until a height value within the threshold was observed or the timeout value was received. The timeout was added as a safety protocol to ensure the rover would stop at the end of the rows. The timeout loop was added to ensure a bad reading did not stop the rover, but rather slowly moved it forward to ensure the rover was not in a spot with a potentially dead plant. The entire flow of the system control can be seen in Figure 10.

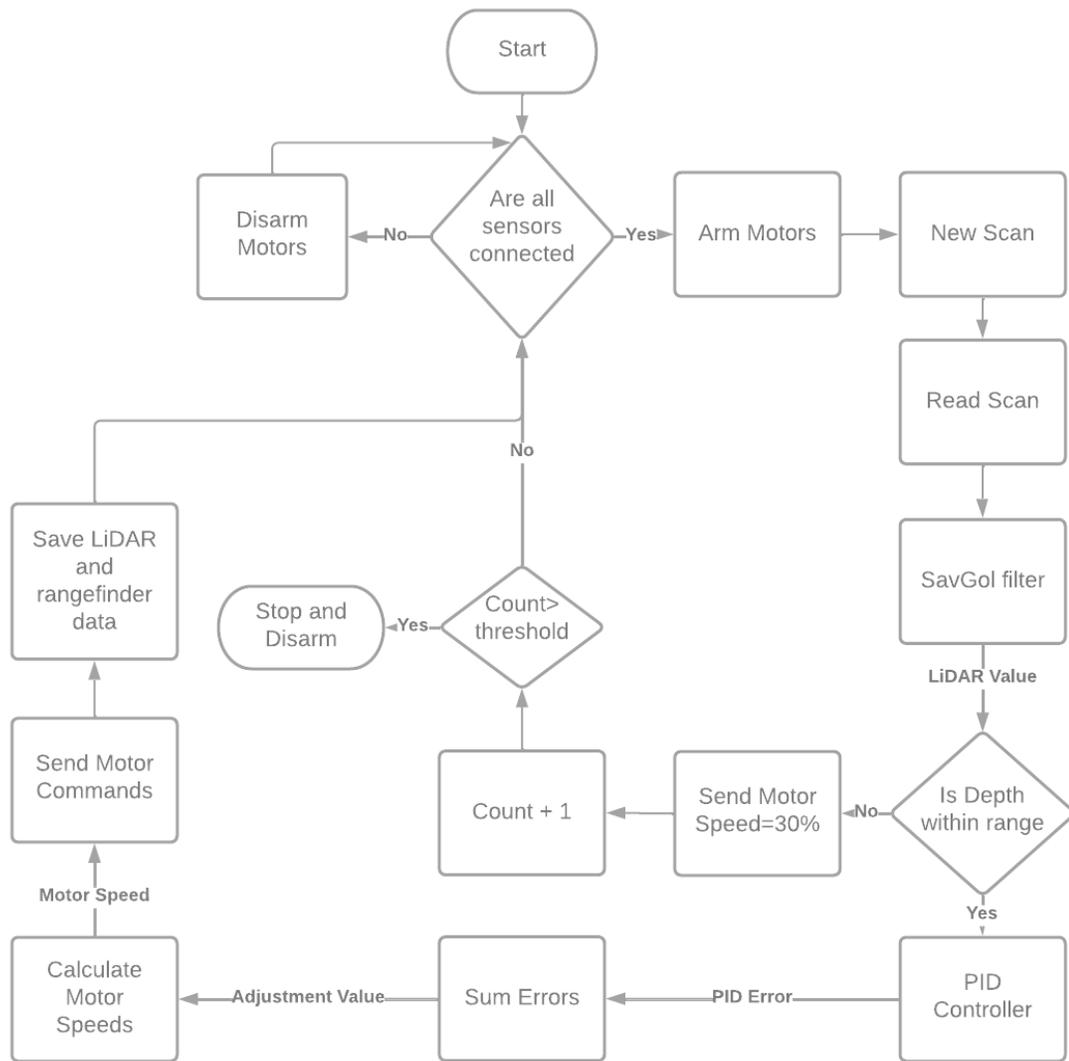


Figure 10 Systems Control Flowchart

To validate the system, the LiDAR scans were recorded and saved. The range finder was also recording a measurement each time a LiDAR scan was saved, and these values were saved. At the end of the test runs, these values were saved to a “.npz” file using NumPy and used in Chapter IV- Results and Analysis to help validate the system.

The control strategy was chosen based on prior knowledge and from methods learned during literature review. From the simple field tests performed with the LiDAR, it was seen that the data was not being read as a smooth curve and would provide unnecessary corrections to the rover’s

path. A Savitzky-Golay filter was chosen for its easy implementation in Python and its proven success. Savitzky-Golay is a low pass filter that allows the smoothing of data. The filter works by fitting successive sets of adjacent data points with a low-degree polynomial by linear least squares. A raw LiDAR scan and a corrected scan can be seen in Figure 11.

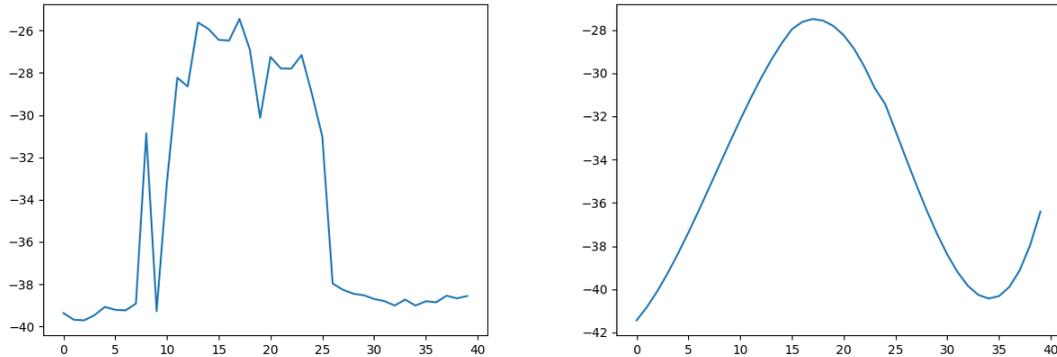


Figure 11 A raw Lidar scan pf peanut canopy (left) and the scan after smoothing (right)

### 3.4 System Testing and Data Collection

The goal of testing was to determine the performance of the LiDAR on detecting the rows, navigate along the row, and which PID values worked the best for the system.

#### 3.4.1 System Overview

The final system consisted of the rover platform outfitted with the LiDAR sensor, a range finder, a Cube Orange Auto Pilot, and a mini-PC. The LiDAR was mounted at the front of the system at a height of 84cm (33in). The range finder was mounted at a height of 127cm (50in). The mounting of both sensors could be seen in Figure 12.

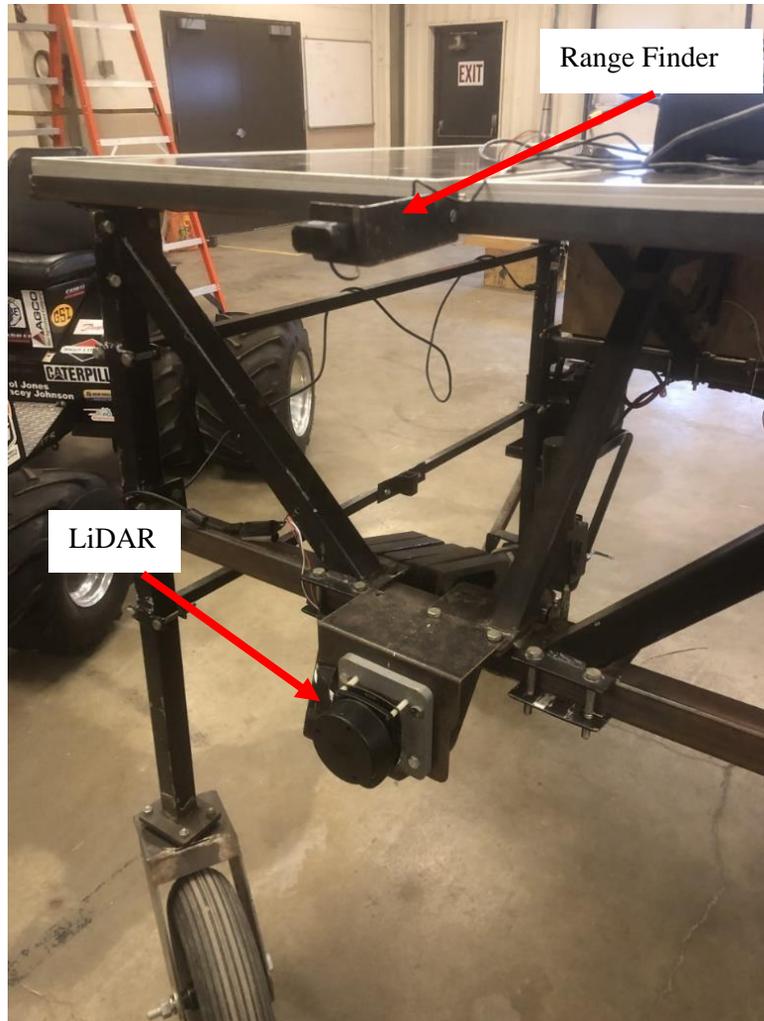


Figure 12 Mountings of a LiDAR sensor and a range finder sensor

The Cube was packaged inside a plastic project box with the RC receiver. The box included a GPS mast on the top that was not used for this project. The Servo Output of the Cube were connected to the Saber tooth motor driver which was housed inside the rover's main electronics box (The electronics box was the original mounting location of the Jetson Nano before it was replaced with the mini-PC). The Cube was then mounted on top of the rover 13cm from the center. The offset from the center was due to the previous GPS mount on the rover's frame. Figure 13 shows the final mounting of the Cube.

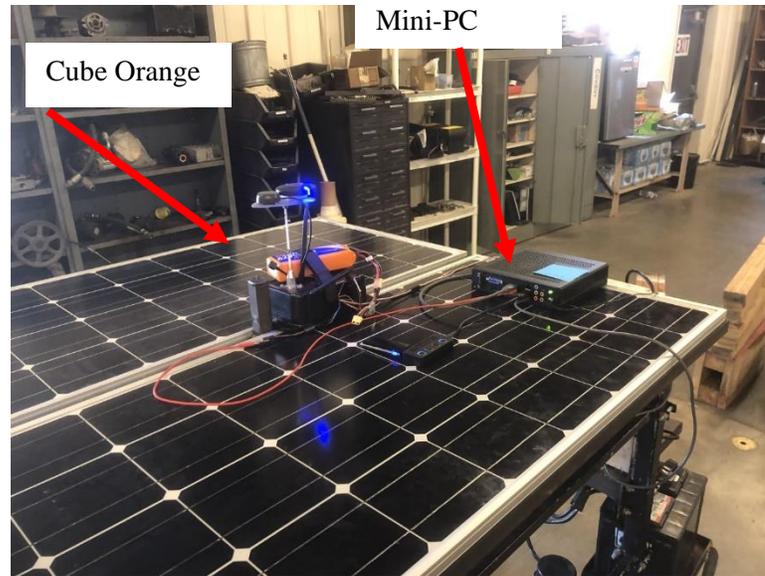


Figure 13 Mounting of the Cube Orange

### *3.4.2 Initial Tuning*

An initial tuning was done on the system to find a range of values that provided adequate control to the system. To begin tuning the PID controller, a Proportional (P) value was added to create oscillation. The rover used differential steering, so the oscillation resulted in a change in speed between the independent motors. Once the system was oscillating evenly, the P value was halved, and an Integral (I) value was added to smooth out the oscillation. After the oscillations smoothed out a Derivative (D) value was then added. A range of possible PID values was recorded to be used in further testing.

### *3.4.3 Testing Setup*

The system was tested using the PID values found during the initial testing. To have a quantitative way to validate the system, simulated rows were used in the lab. An overhead image was taken from the simulated row with a GoPro Hero 5 Black configured in a linear mode. During the tests, the system saved the LiDAR data and a corresponding distance from the range

finder. It was ensured that the rover had the same starting position for each test by using markers on the floor where the tires were supposed to be located.

### *Wood Rows*

To determine which values would provide the most accurate controls, lab tests were setup to simulate row crops. The first test was performed using wood to simulate rows. The system was started at 60% speed, then moved to 80%, and finally 100% with the selected PID values. The PID values were different for each of the speeds as the overall speed of the rover was important to the overall tuning of the system. The length of the rows was 700 cm long, and the width of the row was 17 cm.

As mentioned before, overhead images were taken from the simulated row. The image for the wood row can be seen in Figure 14. The yellow lines represent the boundary of the tests. The boundaries were 700 cm long by 100 cm wide. The image was then cleaned using segmentation and masked. These steps were performed for later analysis of the trials. The final image can be seen in Figure 15.

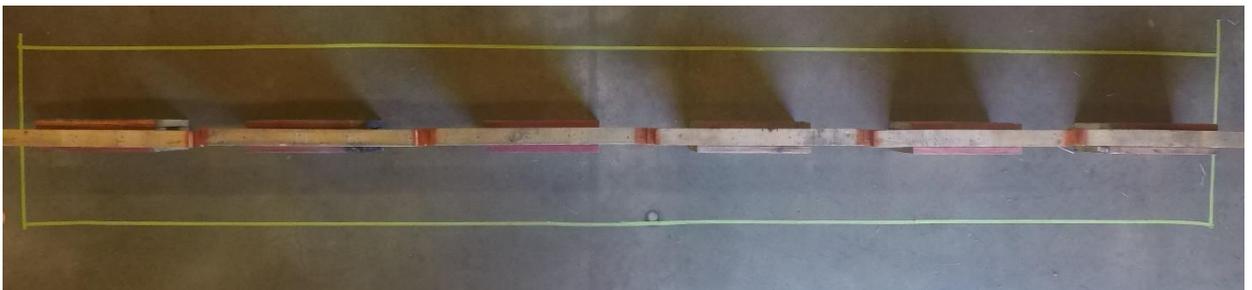


Figure 14 An overhead image of the simulated wood row.

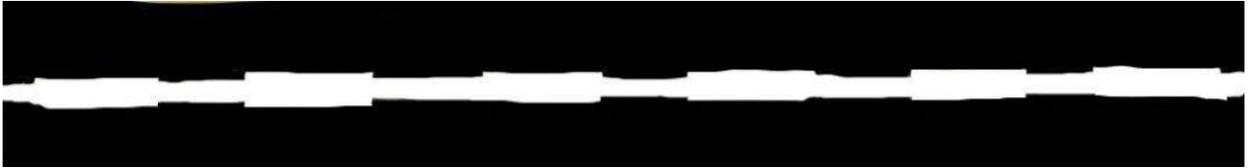


Figure 15 Processed overhead image of the simulated wood row.

### *Angled Approach*

Using the same setup as the wood test, the system was tested with an initial approach angle. An angle of 10 degrees was used to test how the system responded to a large error. The system was tested at 100% speed with the three PID values. After the testing, the same data processing methods were used. The starting position can be seen in Figure 16.

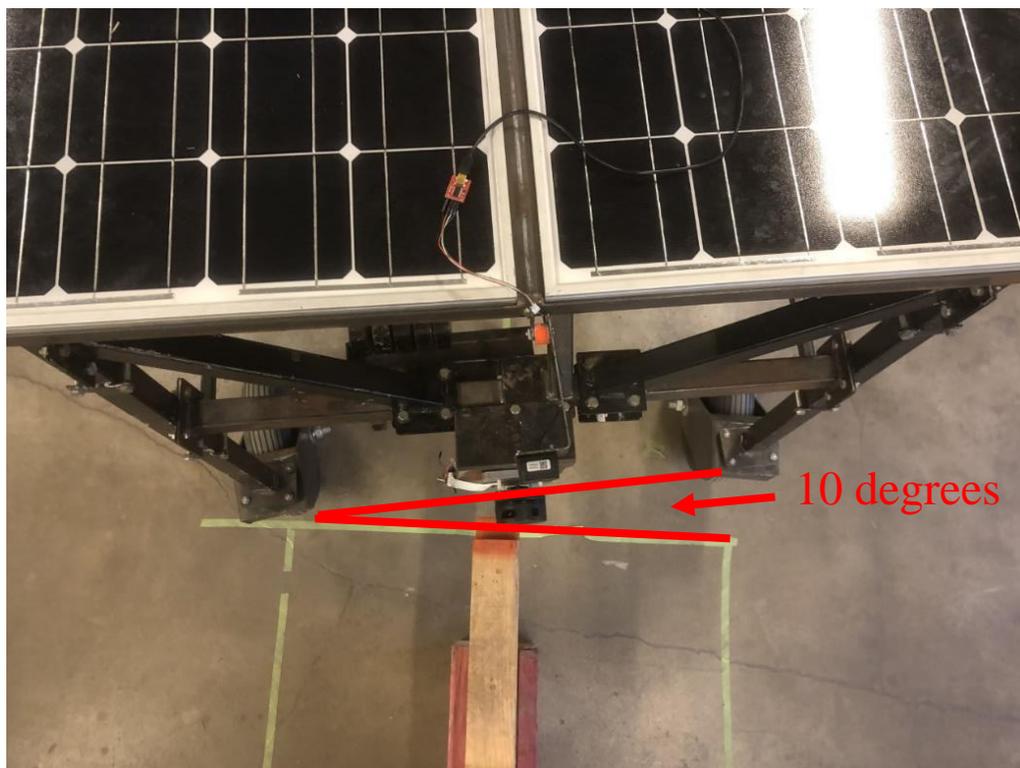


Figure 16 Angled Approach setup

### *Windrow*

To simulate a more realistic situation and still have quantitative data, a windrow was setup in the lab. The length of the row was 700 cm with an average width of 30 cm. The use the windrow allowed tests on a non-uniform pattern. As done in the previous tests, an overhead image was taken of the row, and the same data was collected during the tests. The overhead image and its corresponding segmented image can be seen in the Figure 17 and Figure 18.



Figure 17 overhead image of Windrow test



Figure 18 Processed overhead image of Windrow test

#### *3.4.4 Test Data Interpretation and Analysis*

##### *System Evaluation Process*

After the tests were finished, the data was processed using the developed Python programs. Each test resulted in an overhead image and three NumPy (.npy extension) files. The NumPy files contained all the LiDAR data for the entire test, range finder distances for each LiDAR scan, and the angular error values used to determine the correction needed.

The data analysis process was realized using four separate python programs. The code could be combined, but it was kept separate to ensure no errors occurred between steps.

The full LiDAR scans and the range finder data were read and turned into NumPy arrays. Each LiDAR scan was then paired with its corresponding distance reading. The LiDAR data consisted of a distance and a location (y, and z coordinates). As each LiDAR scan consisted of a different number of readings, the LiDAR scans were iterated through to get a range finder distance for each location giving x coordinates to the data. After the process, the data contained x, y, and z coordinates and was saved to an Excel sheet. The Excel sheet could then be examined to make sure there was no missing point cloud data such as dropped range finder reading. If any range finder readings were listed as “None”, they were turned to 0 which would remove that reading from the data in the next steps.

As the data consisted of decimal values and not whole numbers, it could not be placed directly onto an image. To use the data with an image, it had to be interpolated. The point cloud data was turned into a 2D image using SciPy interpolation. The x and y coordinates were interpolated to estimate the rover’s location. The z values were used to create a greyscale colormap. The interpolation resulted in a  $1080 \times 143$  image which was the same size as the overhead image. Afterward the image was filtered using a binary mask to turn the greyscale image into black and white. An example can be seen in Figure 19.

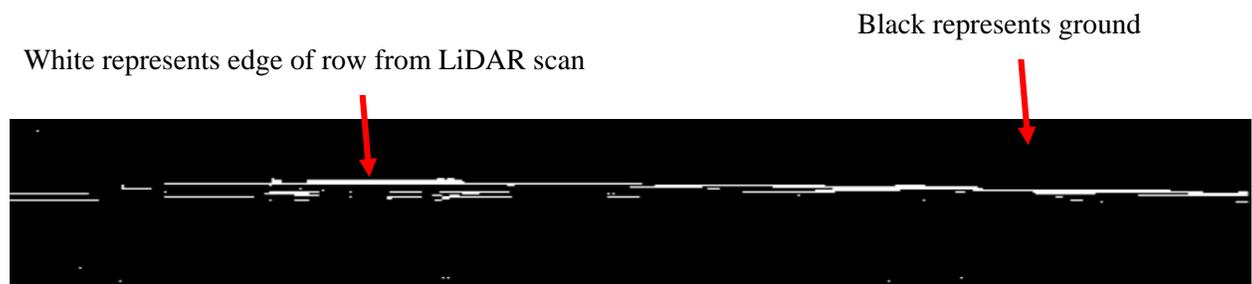


Figure 19 Binary filter of a processed point cloud

Once the image was binarized, it was used as a mask on the overhead image. A bitwise comparison was used as it returned an image that showed the full overhead image with the masked point cloud image on top. An example can be seen in Figure 20.

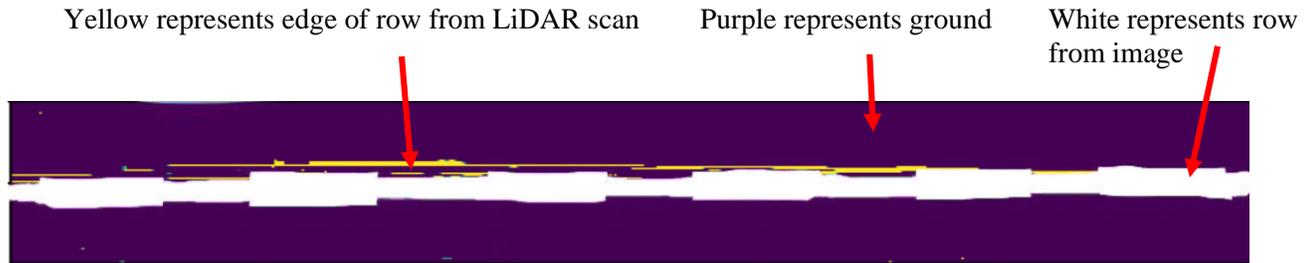


Figure 20 Masked overhead image

Finally, the masked image was analyzed, and the pixel location of an error were saved to an excel sheet. The error was found by iterating through every pixel of the image and determining if the color code was that of the error (yellow). Figure 21 shows a flowchart of the data analysis process.

The final steps of the data analysis were all done in Excel. Using similar parameters by many industry leaders, the accuracy of the system to stay within 15cm, 5cm, and 3cm of the target was found (Deere, 2022). The image had a width of 143 pixels which represented 100cm giving a ground resolution of .699cm/pixel. Meaning that an error of four pixels was 3cm, seven pixels was 5cm, and twenty-one pixels was 15 cm. The large wood blocks were 18 pixels wide and centered at pixel 71, giving an outer boundary from 62 to 80. The data was then sorted using the COUNTIF function in excel to find how many points were outside each range. Knowing there was a maximum of 1080 points a percentage could be found for each range.

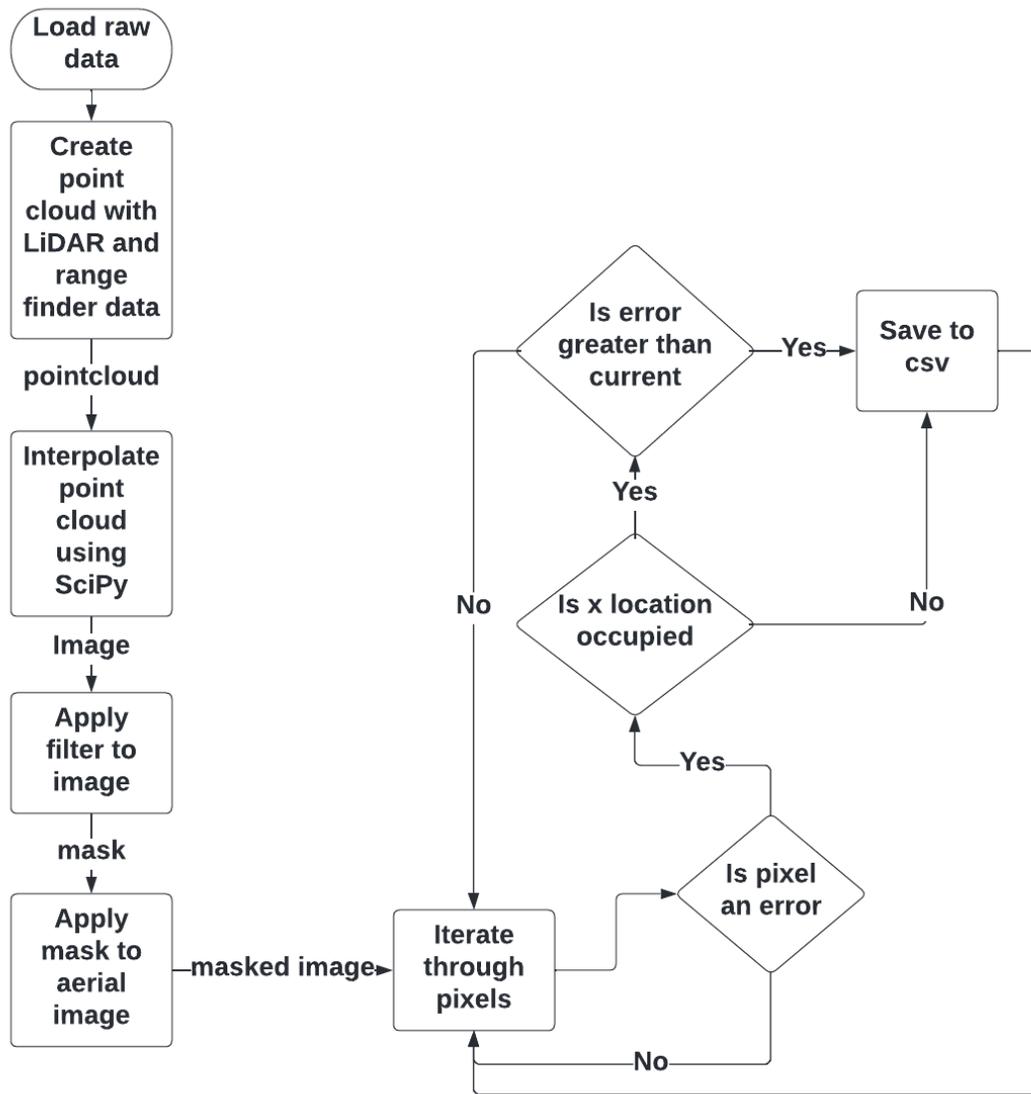


Figure 21 Data Analysis Flowchart

## CHAPTER IV

### Results and Analysis

#### 4.1 Sensors Results and Analysis

##### 4.1.1 LiDAR calibration data

According to the calibration results, the LiDAR was able to acquire accurate readings in the tested range of 56 to 80cm. Most of the variance could be placed on the surface not being perfectly smooth. There were a few outliers included in the results, particularly a reading of 92.9cm in the 70cm test. The averaged data for each test can be seen in the Table 10.

Table 10 LiDAR Calibration Data

	Mounting Height to the Object		
	55cm	70cm	80cm
Max (cm)	57.78	92.91	82.40
Min (cm)	55.28	69.96	78.68
Avg (cm)	56.47	71.30	80.61
SD (cm)	0.52	0.48	0.79

##### 4.1.2. TF Mini calibration data

The TF Mini range finder was able to acquire accurate readings for all three distances. The readings at 140 and 280cm yielded very consistent results. The tests at 570 cm were also acceptable having a standard deviation of only 1.3cm. The results can be seen in Table 11.

Table 11 TF-Mini range finder calibration data.

Distance to Object			
	140 cm	280 cm	570 cm
Avg (cm)	140	281.09	569.76
Max (cm)	140	282	574
Min (cm)	139	280	567
SD (cm)	.39	0.51	1.32

#### 4.1.3. Ultrasonic Sensors

The ultrasonic sensor testing did not return any usable data as the readings were inconsistent and rarely received. The poor readings are likely a result of the low quality of the ultrasonic sensors tested. The use of the sensors in this configuration, while intriguing, was not suitable for an application when consistent readings were critical.

#### 4.1.4 Stereoscopic Cameras

The results showed that the D435 needed to be tuned to be used outside as the exposure was remarkably high during the outside tests. The comparisons between the two cameras can be seen in Table 12, Table 13, and Table 14. It was showed that the configurations should be optimized for each camera before a decision was made.

Table 12 The results of the indoor test between the D415 and D435 Camera

Control	Camera	Distance (m)	SD (mm)	% Zeroes	% Outliers
	D415	0.5	152.01	1.94	0
0.7		10.90	0.003	0.03	
0.9		51.27	0.90	0.002	
1.1		20.55	0.16	0	
D435	0.5	142.58	1.60	0	
	0.7	109.69	1.50	0	
	0.9	85.52	1.52	0	
	1.1	59.36	1.51	0	

Table 13 The results of the zero-light test between the D415 and D435 Camera

Zero Light	Camera	Distance (m)	SD (mm)	% Zeroes	% Outliers
	D415	0.5	184.11	2.23	0
		0.7	96.24	1.89	0
		0.9	15.07	0.01	0
		1.1	141.26	1.56	0
	D435	0.5	58.97	1.49	0
		0.7	84.88	1.50	0
		0.9	109.67	1.51	0
		1.1	141.26	1.56	0

Table 14 The results of the full sun test between the D415 and D435 Camera

Full Sun	Camera	Distance (m)	SD (mm)	% Zeroes	% Outliers
	D415	0.5	114.63	5.61	0
		0.7	77.19	1.29	0
		0.9	168.24	3.18	0.005
		1.1	188.96	2.95	0
	D435	0.5	8448.85	85.37	4.45
		0.7	18300.24	84.38	8.54
		0.9	13898.80	37.00	1.69
1.1		414.19	17.18	0	

## 4.2 Preliminary field tests results

### 4.2.1 LiDAR tests

As the LiDAR was a 360-degree scan, it resulted in data in Polar coordinates. A simple program was made to convert the scans into Cartesian coordinates for visualization purposes. The angle of the scan was also limited to eliminate data from outside the region of interest (ROI). With the new scan area defined, preliminary field tests were performed. The setup can be seen in Figure 22. Figure 23 shows a LiDAR scan as a 2D plot, while Figure 24 shows a 3D plot of the data. The test was performed without any wheel encoders or GPS; therefore, the x direction (direction of

travel) could not be validated. From this test, it was decided that future testing should be done indoors, and the range finder was added to the system.



Figure 22 The rover with the LiDAR mounted on the front.

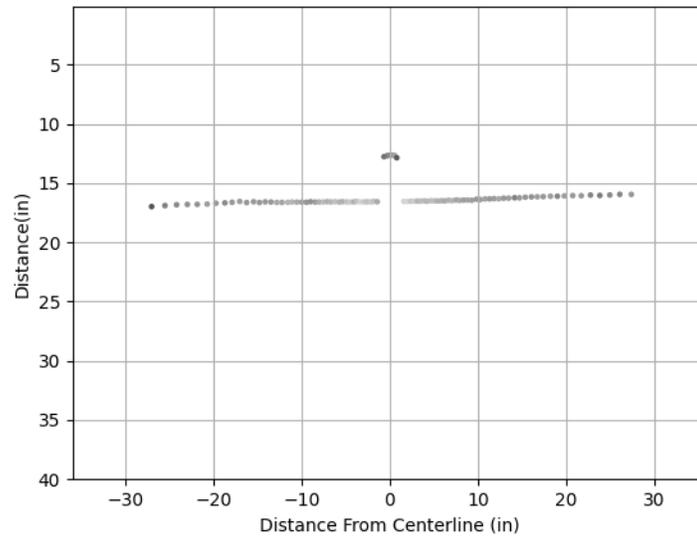


Figure 23 A single LiDAR scan.

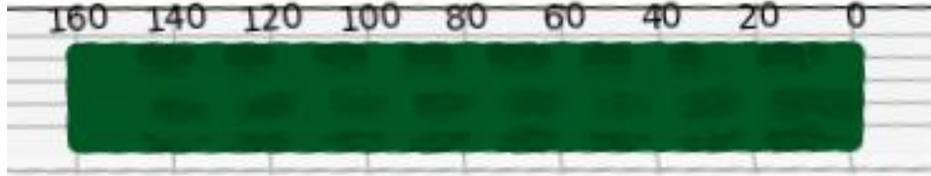


Figure 24 A point cloud of the field after being scanned by the LiDAR.

#### 4.2.2 Autopilot tests

The communication between the Cube and the Jetson Nano proved to be troublesome. It was found that the UART pins of the Jetson Nano had connection issues, and completely stopped working during initial testing. To solve the connection issue a USB to serial converter was used to connect the Cube to the Jetson Nano. The USB to Serial converter worked well for the initial testing, but then for unknown reasons it also failed. The Jetson Nano was then traded out for a mini-PC. The mini-PC worked well, but it did draw more power than the Jetson Nano.

#### 4.3 System Evaluation

Table 15 and Table 16 show the PID values used in the system tests. These were determined through initial tuning of the system. Test 1, Test 2, and Test 3 will be used to reference the different PID values.

Table 15 PID values of 60% Speed.

Test 1	P = .15, I = .005, D = .001
Test 2	P = .35, I = .005, D = .001
Test 3	P = .25, I = .005, D = .001

Table 16 PID Values for 100% Speed

Test 1	P=.3, I=.005, D=.008
Test 2	P=.3, I=.005, D=.005
Test 3	P=.35, I=.005, D=.01

### 4.3.1 Tests on the wood row

The use of wood to create rows worked very well. The ability to test various PID tunings without any changing parameters in the row greatly assisted with the tuning process. The first tests were conducted at 60% speed to make sure no issues occurred with the code. The ranges of the Proportional (P) values for the PID control were also spread out to better see how the tuning of the PID loop affected results. From Figure 25, the results show that the best results came from Test 2 followed closely by Test 3. Test1 was using the smallest P value and had a large amount of error, While Test 2 had the largest P value with the least amount of error. The values used in the plot can be seen in Table 17.

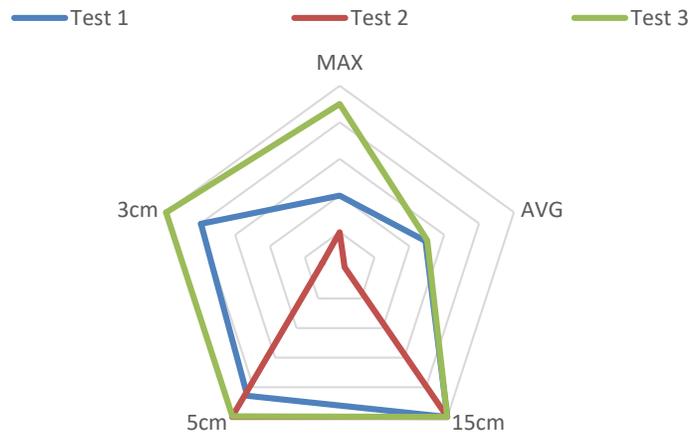


Figure 25 Results of 60% Speed

Table 17 Results of 60% Speed

	Test 1	Test 2	Test 3
MAX Error (cm)	4	2	9
AVG Error (cm)	0.49	0.03	.50
% In 15cm Range	100	100	100
% In 5cm Range	85.71	100	99.63
% In 3cm Range	79.60	99.97	99.60

Moving to the 100% speed trials it was decided to use slightly larger P values to compensate for the added speed. The Derivative (D) values were also increased. The results of the straight approach wood test can be seen in Figure 26 and Table 18. Test 2 had the best results staying in the 5cm range more often than the other tests and having a lower max and average error.

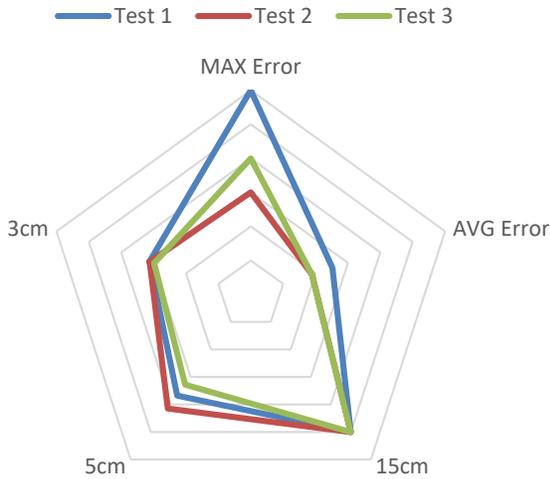


Figure 26 Results of Wood Test at 100% Speed

Table 18 Results of Wood Test at 100% Speed

	Test 1	Test 2	Test 3
MAX Error (cm)	12	6	8
AVG Error (cm)	5.04	3.78	3.78
% In 15cm Range	100	100	100
% In 5cm Range	73.55	82.96	65.56
% In 3cm Range	62.59	62.45	59.66

### 4.3.2 Test with the angled approach

To further test the system an approach angle of 10 degrees was used. Figure 27 and Table 19 show the results of the tests. Again Test 2 was the most reliable keeping inside the 5cm range 99% of the time. Test 1 was still the worst setup having a maximum error of 15cm.

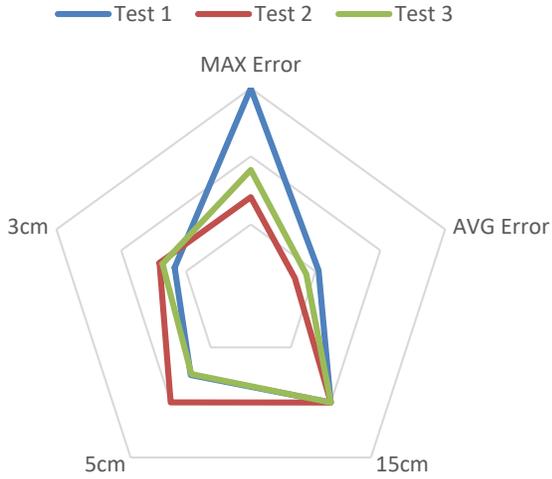


Figure 27 Results of Angled Approach at 100% speed

Table 19 Results of Angled Approach at 100% speed

	Test 1	Test 2	Test 3
MAX Error (cm)	15	7	9
AVG Error (cm)	5.23	3.40	4.29
% In 15cm Range	99.95	100.00	100.00
% In 5cm Range	74.98	99.97	74.24
% In 3cm Range	58.82	70.42	68.08

### 4.3.3 Tests on the windrow

The last test used windrows created in the lab to simulate a plant canopy. Figure 28 and Table 20 show the results of the tests. All the tests did well, but Test 3 did slightly better than Test 1.

Looking at the data in Table 20, the results showed that the maximum error was larger than the other two tests.

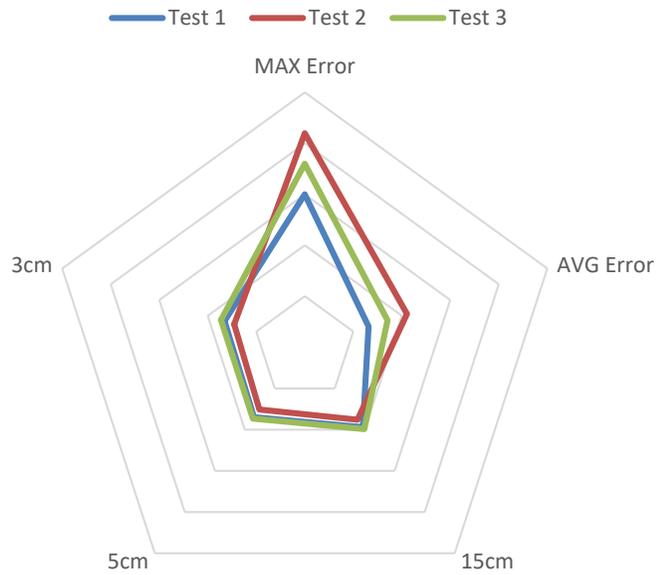


Figure 28 Results of Windrow test at 100% Speed

Table 20 Results of Windrow test at 100% Speed

	Test 1	Test 2	Test 3
MAX Error (cm)	15	21	18
AVG Error (cm)	6.59	10.54	8.53
% In 15cm Range	97.01	87.87	99.26
% In 5cm Range	84.85	75.65	86.60
% In 3cm Range	82.78	72.75	86.27

#### 4.3.4 Comparison of the system tests

Figure 29 shows a comparison among the results from Test 1 using different targeted rows. Test 1 had similar maximum errors with the Windrow tests having the highest accuracy at 3cm, but smallest accuracy at 15cm, meaning when it did have an error it was greater than the errors of the other two tests.

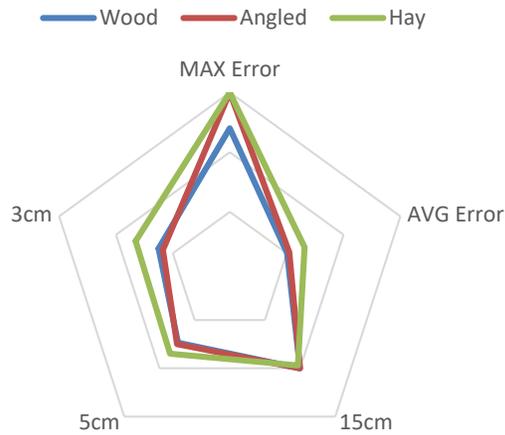


Figure 29 Comparison of each Trials Test1 Values

Figure 30 shows a comparison among the results from Test 2 using different targeted rows. The Windrow trials did significantly worse than the other two setups, having a maximum error of 21 cm, meaning that it was nearly running over the row. The results are surprising since the other two setups were nearly perfect.

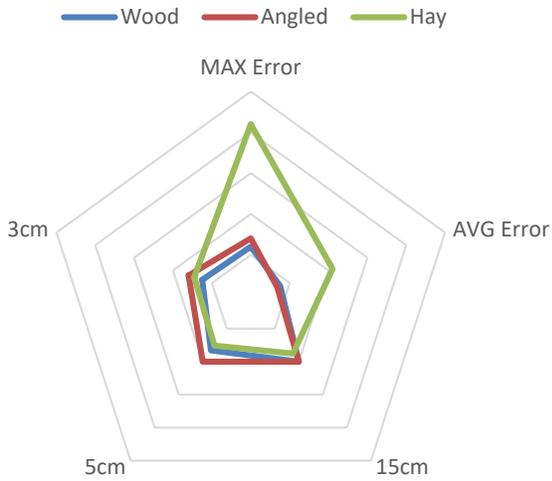


Figure 30 Comparison of each Trials Test2 Values

Figure 31 shows a comparison among the results from Test 3 using different targeted rows. The Windrow test had the largest error again, but it had the highest accuracy at 3cm. The wood and angled tests were nearly identical for this PID value.

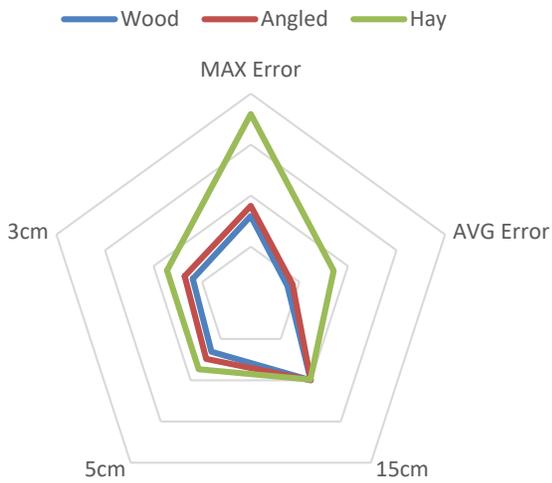


Figure 31 Comparison of each Trials Test3 Values

The average error of each PID test and the average error for each setup can be seen in Table 21. The Windrow test had roughly double the average error of the wood and angled tests. The error between each PID value was roughly the same, but Test 2 for the Windrow was nearly 20% larger than the next highest error.

Table 21 Average Error for each test

	Test1	Test2	Test3	Average
Wood	5.04	3.78	3.78	4.20
Angled	5.23	3.40	4.29	4.31
Windrow	6.59	10.54	8.53	8.55
Average	5.62	5.91	5.53	

#### *4.4 System Safety*

Due to the Jetson Nano not being used the proposed safety interlocks were not accomplished.

Without the GPIO pins of the Jetson, there was not a straightforward way to control the relays.

The code was, however, setup to unarm the system if any of the sensors were ever disconnected.

## CHAPTER V

### Conclusions

#### 5.1 Performance of system

An autonomous rover was developed using off-the-shelf parts and tested in this study. The tests were conducted under lab conditions with simulated object rows. Data collection algorithms and data and image processing algorithms were developed for the rover controls and to evaluate the performance of the system.

The use of the LiDAR over the cameras was determined by accuracy and processing power needed. The LiDAR worked well at detecting the rows but had connection issues. Due to several factors, such as inconsistent scan rates, or overheating, the laser could power off and require being reset to turn back on.

The Cube worked as expected but was not necessarily needed after GPS was removed from the system. It was a good device to have for future development of the system. It was later found that the Jetson Nano would not communicate with the Cube due to a corruption on the boot drive. The Jetson Nano was re-flashed and is now operating as intended. The mini-PC proved to be sufficient as the datasets were relatively small.

The data shows that the system could successfully navigate through rows on its own. The Windrow trials had larger maximum errors which could be accounted to the uneven shape of the canopy. There were several locations where the Windrow was much taller on one side than the other potentially resulting in over corrections. To correct these errors, the Savitzky-Golay filter could be fine-tuned. However, due to the LiDAR limitations, it could result in a scan not being

read or a system timeout. To correct the issue a LiDAR with constant scan angles is suggested. Even with these errors the system ran well and never interfered with the rows.

## 5.2 Future Work

There is still much work to be done to create a fully autonomous system. Future work includes head row turns as this system would only navigate through a single row at a time and still requires an operator for turning around. Another solution was to include a First Person View (FPV) camera and to transmit the feedback to a ground station. By adding the video stream, the operator could control the rover from a set location.

The system could be programmed using a faster language, such as C++, or could be moved to ROS. There was the need for added sensors to accomplish complete autonomy, and ROS could allow for simulation of the system. Without the addition of object detection, the rover could potentially cause damage to the crop or itself, or cause injury to a human. Potential sensors could be included such as the TF Mini range finders used in the validation process. These sensors had the ability to connect directly to the Cube autopilot. With this simple connection, the Cube would have the ability to Disarm the system if an object were detected within the set threshold.

Additional work could also be done to find a different LiDAR solution. While the current LiDAR worked and was certainly cost affective, it also has several issues. The LiDAR tended to timeout causing the entire system to stop and wait for reconnection.

## REFERENCES

- ACES. (2020). *TerraSentia robots, Agricultural and Biological Engineering faculty featured in New York Times*. Retrieved 04/20/2022 from <https://aces.illinois.edu/news/terrasentia-robots-agricultural-and-biological-engineering-faculty-featured-new-york-times>
- Alapetite, A., Wang, Z., Hansen, J. P., Zajączkowski, M., & Patalan, M. (2020). Comparison of Three Off-the-Shelf Visual Odometry Systems. *Robotics*, 9(3), 56.
- Anders Grunnet-Jepsen, J. N. S., John Woodfill. (2020). *Best-Known-Methods for Tuning Intel® RealSense™ D400 Depth Cameras for Best Performance*. Retrieved 04/20/2022 from [https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/BKMs\\_Tuning\\_RealSense\\_D4xx\\_Cam.pdf](https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/BKMs_Tuning_RealSense_D4xx_Cam.pdf)
- Bayer, J., & Faigl, J. (2019, 4-6 Sept. 2019). On Autonomous Spatial Exploration with Small Hexapod Walking Robot using Tracking Camera Intel RealSense T265. 2019 European Conference on Mobile Robots (ECMR),
- Bedord, L. (2016). *Case IH Unveils Autonomous Concept Tractor*. Retrieved 04/20/2022 from <https://www.agriculture.com/news/machinery/case-ih-unveils-autonomous-concept-tractor>
- Bergerman, M., Van Henten, E. J., Billingsley, J., Reid, J., & Deng, M. (2013). IEEE Robotics and Automation Society Technical Committee on Agricultural Robotics and Automation [TC Spotlight]. *Robotics & Automation Magazine, IEEE*, 20, 20-125. <https://doi.org/10.1109/MRA.2013.2255513>
- Caterpillar. (2022). *Cat® MineStar™ Command for Hauling*. Retrieved 04/20/2022 from [https://www.cat.com/en\\_US/by-industry/mining/surface-mining/surface-technology/command/command-hauling.html](https://www.cat.com/en_US/by-industry/mining/surface-mining/surface-technology/command/command-hauling.html)
- Collie, S. (2016). *Self-driving tractors promise to get themselves to work, plow without complaint*. Retrieved 04/20/2022 from [https://newatlas.com/self-driving-tractor/45169/?itm\\_source=newatlas&itm\\_medium=article-body](https://newatlas.com/self-driving-tractor/45169/?itm_source=newatlas&itm_medium=article-body)
- Conesa-Muñoz, J., Bengochea-Guevara, J. M., Andujar, D., & Ribeiro, A. (2016). Route planning for agricultural tasks: A general approach for fleets of autonomous vehicles in site-specific herbicide applications. *Computers and Electronics in Agriculture*, 127, 204-220. <https://doi.org/https://doi.org/10.1016/j.compag.2016.06.012>
- Coxworth, B. (2020). *Kubota unveils radical autonomous electric tractor in Japan*. Retrieved 04/20/2022 from <https://newatlas.com/automotive/kubota-autonomous-electric-tractor/>.
- Deere, J. (2021). Retrieved 04/20/2022 from <https://www.deere.co.uk/en/agriculture/future-of-farming/>
- Deere, J. (2022). Retrieved 04/20/2022 from <https://www.deere.com/en/technology-products/precision-ag-technology/guidance/starfire-6000-receiver/>
- Fendt. (2017). *Fendt Develops Robot System for Planting and Accurate Documentation*. Retrieved 04/20/2022 from <https://www.oemoffhighway.com/trends/equipment-launches/agriculture/press-release/20983953/fendt-fendt-develops-robot-system-for-planting-and-accurate-documentation>
- Fernandez Guzman, B., Herrera, P., & Cerrada, J. (2018). Robust digital control for autonomous skid-steered agricultural robots. *Computers and Electronics in Agriculture*, 153, 94-101. <https://doi.org/10.1016/j.compag.2018.07.038>
- Gossett, S. (2019). *Robots Working in Farming and Agriculture*.
- Iqbal, J., Xu, R., Sun, S., & Li, C. (2020). Simulation of an Autonomous Mobile Robot for LiDAR-Based In-Field Phenotyping and Navigation. *Robotics*, 9(2). <https://doi.org/10.3390/robotics9020046>

- Isack, H., & Boykov, Y. (2012). Energy-Based Geometric Multi-model Fitting. *International Journal of Computer Vision*, 97(2), 123-147. <https://doi.org/10.1007/s11263-011-0474-7>
- Kanagasingham, S., Ekpanyapong, M., & Chaihan, R. (2020). Integrating machine vision-based row guidance with GPS and compass-based routing to achieve autonomous navigation for a rice field weeding robot. *Precision Agriculture*, 21(4), 831-855. <https://doi.org/10.1007/s11119-019-09697-z>
- Maja, J. M., Polak, M., Burce, M. E., & Barnes, E. (2021). CHAP: Cotton-Harvesting Autonomous Platform. *AgriEngineering*, 3(2). <https://doi.org/10.3390/agriengineering3020013>
- Malavazi, F. B. P., Guyonneau, R., Fasquel, J.-B., Lagrange, S., & Mercier, F. (2018). LiDAR-only based navigation algorithm for an autonomous agricultural robot. *Computers and Electronics in Agriculture*, 154, 71-79. <https://doi.org/https://doi.org/10.1016/j.compag.2018.08.034>
- Meiser, V., Henke, R., Šeatovic, D., Rotach, T., & Hesselbarth, H. (2014). *Autonomous Unmanned Ground Vehicle as Sensor Carrier for Agricultural Survey Tasks*
- Moeller, R. (2020). *GPS-Guided Autonomous Robot with Obstacle Avoidance and Path Optimization*
- ROWBOT. (2022). Retrieved 04/20/2022 from <https://www.rowbot.com>
- Schmitz, A. (2017). *Row crop navigation by autonomous ground vehicle for crop scouting* [Kansas State University].
- Shalal, N., Low, T., McCarthy, C., & Hancock, N. H. (2013). A review of autonomous navigation systems in agricultural environments.
- Singh, J., Dhuheir, M., Refaey, A., Erbad, A., Mohamed, A., & Guizani, M. (2020, 30 Aug.-2 Sept. 2020). Navigation and Obstacle Avoidance System in Unknown Environment. 2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE),
- Slamtec. (2022). *RPLIDAR-A1 Laser Range Scanner*. Retrieved 04/20/2022 from
- Tadic, V., Odry, Á., Kecskes, I., Burkus, E., Király, Z., & Odry, P. (2019). Application of Intel RealSense Cameras for Depth Image Generation in Robotics. *WSEAS Transactions on Computers*, 18, 107-112.
- Technologies, N. (2020). *Naio Technologies closes Series A funding to ready weeding robots for mass production*. Retrieved 04/20/2022 from <https://www.therobotreport.com/naio-technologies-closes-series-a-readies-weeding-robots-production/>
- Weiss, U., & Biber, P. (2011). Plant detection and mapping for agricultural robots using a 3D LIDAR sensor. *Robotics and Autonomous Systems*, 59, 265-273. <https://doi.org/10.1016/j.robot.2011.02.011>
- William H. Kusewich, W. H. (2007). *Using an Ultrasonic to Determine Location of An Object On the Co-ordinate Grid* SUMMIT, NYU.
- Zhang, C., Yong, L., Chen, Y., Zhang, S., Ge, L., Wang, S., & Li, W. (2019). A Rubber-Tapping Robot Forest Navigation and Information Collection System Based on 2D LiDAR and a Gyroscope. *Sensors*, 19(9), 2136.
- Zhmud, V. A., Kondratiev, N. O., Kuznetsov, K. A., Trubin, V. G., & Dimitrov, L. V. (2018). Application of ultrasonic sensor for measuring distances in robotics. *Journal of Physics: Conference Series*, 1015, 032189. <https://doi.org/10.1088/1742-6596/1015/3/032189>

VITA

Austin Pickering

Candidate for the Degree of

Master of Science

Thesis: DEVELOPMENT OF AN AUTONOMOUS ROVER FOR FIELD APPLICATIONS

Major Field: Biosystems Engineering

Biographical:

Education:

Completed the requirements for the Master of Science in Biosystems Engineering at Oklahoma State University, Stillwater, Oklahoma in May, 2022.

Completed the requirements for the Bachelor of Science in Biosystems Engineering at Oklahoma State University in Stillwater, Oklahoma in May, 2020.

Experience:

Graduate Research Assistant at Oklahoma State University from 2020 until 2022, working on manned and unmanned ground rovers.

Undergraduate Research Assistant at Oklahoma State University from 2019 through 2020, working on ground based robot designs and data collection methods.

Professional Memberships:

American Society of Agricultural and Biological Engineers