LMI: The Lidar Mapping Instrument

Jocelyn Reahl ‘19

ASTR 202: Hands-On Planetary Exploration

May 22, 2018

Prof. Wes Watters

Abstract

The Scanning Outcrop Rover (ScOut) is a semi-autonomous, radio-enabled, path-finding rover that was built during the Spring 2018 semester. ScOut was built in order to explore a bedrock outcrop in the Wellesley Hinterlands, located just west of Lake Waban in Wellesley, Massachusetts. ScOut had two primary scientific objectives: to construct a three-dimensional map of a bedrock outcrop and to collect images of the outcrop surface. Due to time constraints, ScOut was not fully constructed by the end of the semester. However, individual components of this rover were partially or fully complete by the end of this project. This paper focuses on one of the two scientific instruments: the Lidar Mapping Instrument (LMI; pronounced “Lemmy”). LMI constructed a three-dimensional point cloud map using a specialized Lidar scanning mount. These data were used to: 1) interpret an environment and 2) to create a DEM (digital elevation model) for the path-finding algorithm.

Introduction

Lidar, an acronym for Light Detection and Ranging, is an active form of remote sensing where information about an object is obtained from a signal sent by a transmitter that reflects off of the object and is received back at the source. While there are many types of Lidar, the most common and commercially available Lidar is topographic Lidar, also known as Laser Altimetry for airborne applications. Topographic Lidar sensors use near-infrared (NIR) light in the form of a pulsed laser to measure distances of objects relative to the sensor. These sensors employ the following time-of-flight (ToF) equation to determine the distance *D* that the object is away from the sensor:

$$D = \frac{tc}{2} \tag{1}$$

Where *t* is the elapsed time between sending the laser pulse and recording the backscattered signal, and *c* is the speed of light (3.0×10^8 m s^{-1}).

LMI’s Lidar sensor is Garmin’s Lidar Lite v3, which uses a Class 1 NIR laser to measure the distances of objects at a range of up to 40 meters. In LMI’s case, the LIDAR Lite v3 works on an I^2C bus under the control of an Arduino Nano microcontroller. A hookup guide for this sensor is available on SparkFun’s website. Finally, the Lidar Lite v3 operates on the Arduino IDE using Garmin’s LIDARLite library (available on GitHub).

This sensor is only able to measure distances; therefore, it is unable to independently construct a three-dimensional point cloud map. In order to construct a map, LMI records three variables in order to record a Lidar measurement in a spherical coordinate system: distance D, azimuth angle θ, and altitude angle ϕ. θ and ϕ are known using two servo motors, hereby referred to as the azimuth and altitude motors, respectively. A diagram of the relationship between the spherical and Cartesian coordinate systems is shown in Figure 1, and the following equations are used to convert spherical coordinates into Cartesian coordinates:

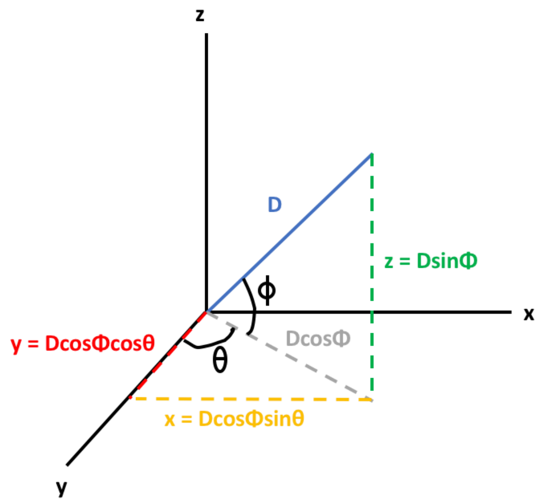


Figure 1. Diagram of the relationship between spherical and Cartesian coordinates.

$$
\begin{aligned}
x &= Dcos\phi sin\theta \\
y &= Dcos\phi cos\theta \\
z &= Dsin\phi
\end{aligned}
\tag{2}
$$

LMI was also meant to be equipped with SparkFun's MPU 6050 (developed by InvenSense) and SparkFun's MAG 3110 Triple Axis Magnetometer. However, there was not enough time to integrate these sensors into LMI.

Instrument Description

LMI uses the following electronic parts: the Lidar Lite v3, a 680 µF capacitor, one standard size servo motor (azimuth servo), one 8 g mini servo motor (altitude servo), an Arduino Nano, a ¼ size Adafruit protoboard, a ½ size Adafruit protoboard, and a 6-wire slip ring. The 6-wire slip ring prevents rotating wires from tangling and disconnecting. The 680 µF capacitor is recommended by Garmin and makes it easier for the Lidar sensor to make measurements. There is room available on the ½ size protoboard to install the MPU 6050 and MAG 3110 sensors in future iterations of this project. There are three male header pins located on the ¼ and ½ size protoboards that accommodate female servo headers. There is also a male header pin soldered to pin A0 on the ½ size protoboard that allows for Parallax Feedback 360°™ functionality. Figure 2 shows a Fritzing diagram of the LMI circuit.

LMI's Lidar Lite v3 sensor uses a Class 1 NIR laser with a nominal wavelength of 905 nm and an 8 mRadian beam divergence to measure the distances of objects at a range of up to 40 meters. The sensor has an accuracy of ± 2.5 cm between 2 and 5 meters and an accuracy of ± 10 cm at greater than 5 meters. The sensor is able to sample 270 times per second and operates under an I^2C or pulse width modulation (PWM) interface at a voltage between 4.5 and 6 volts. More information on this sensor can be found in Garmin's Lidar Lite v3 Manual.

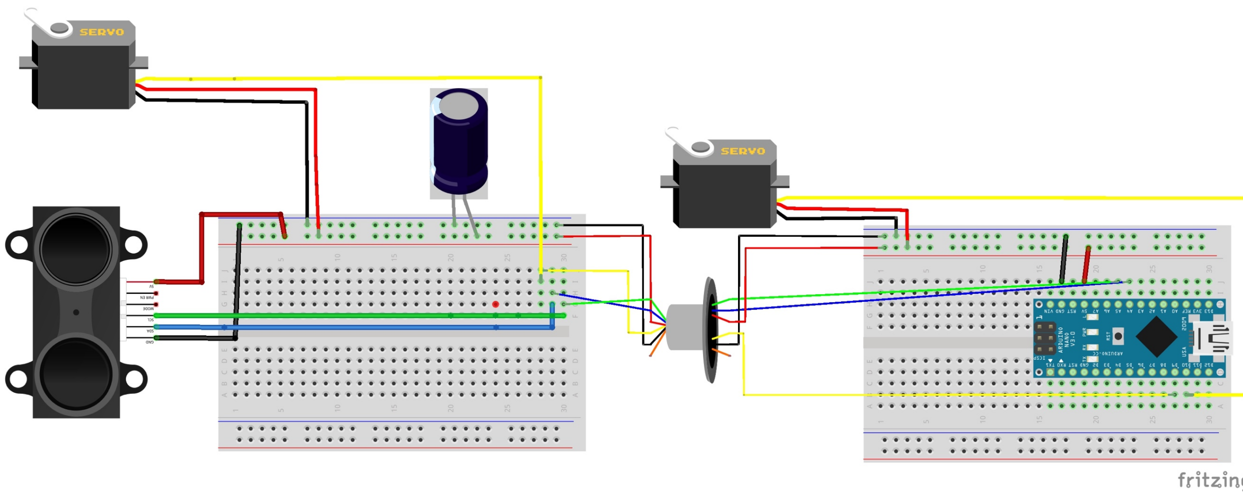


Figure 2. Fritzing diagram of the LMI circuit. The altitude servo and Lidar are attached to the same ¼ size protoboard (here represented as a ½ size breadboard) with a 680 µF capacitor. The connections on the ¼ size protoboard are then connected to the 6 wire slip ring. The wires from this slip ring are then connected to the Arduino Nano. The azimuth servo is directly connected to the Arduino at digital pin 9 (D9) while the altitude servo is connected via the slip ring at digital pin 10 (D10). The Lidar's SDA and SCL wires (blue and green, respectively) are connected to the Nano's analog pins 4 (A4) and 5 (A5), respectively.

The LMI mount was constructed using a combination of laser-cut pieces of plywood and delrin, 3D printed parts, and hot glue. These parts can be found in a variety of formats (SolidWorks, DXF, CDR, and STL) at this project's associated GitHub repository. Figure 3 shows pictures of the mount assembly and 3D printed parts in SolidWorks. Figure 4 is a photograph of the completed LMI assembly.

The 3D printed parts were printed in Clapp Library's MakerLab. There were three 3D printed parts: the sensor mount, which secures the Lidar Lite v3 and ¼ size protoboard using screws; the bracket, which suspends the sensor mount in the air and serves as the location where the altitude servo is held in place, and the wirefeed, which allows the gear below the bracket to securely rotate in place while allowing for wires from the slip ring to pass through into the housing's interior.

The main mount housing was built with laser-cut plywood pieces. Every piece of the housing was held together with hot glue with the exception of the mount base, which was screwed into place. The screws were used to facilitate easy removal and access into the housing's interior. There are three openings on the housing: a place for the 3D printed wirefeed piece; a place to install a standard-size servo motor; and a place to plug in the micro USB for the Arduino Nano. The wirefeed was hot glued with the gear attached onto the housing. The gears used to rotate the Lidar sensor in the azimuth direction were laser cut out of ¼" white delrin. The bracket was secured to a bracket mount (which was also laser cut and hot glued to a gear) using six screws.

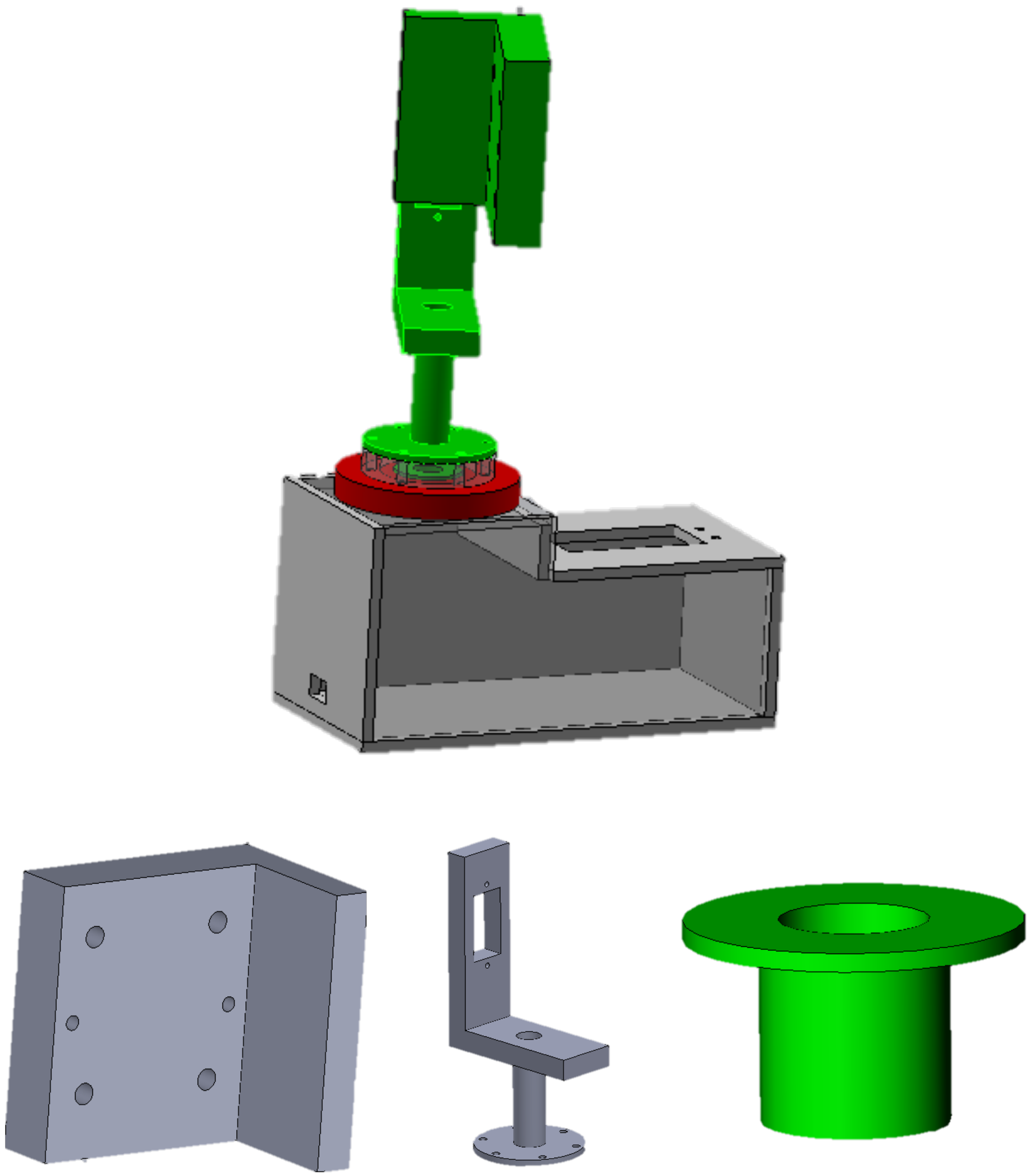


Figure 3. The mount assembly (top) and the three 3D printed parts: the sensor mount (bottom left), bracket (bottom center), and wirefeed (bottom right).

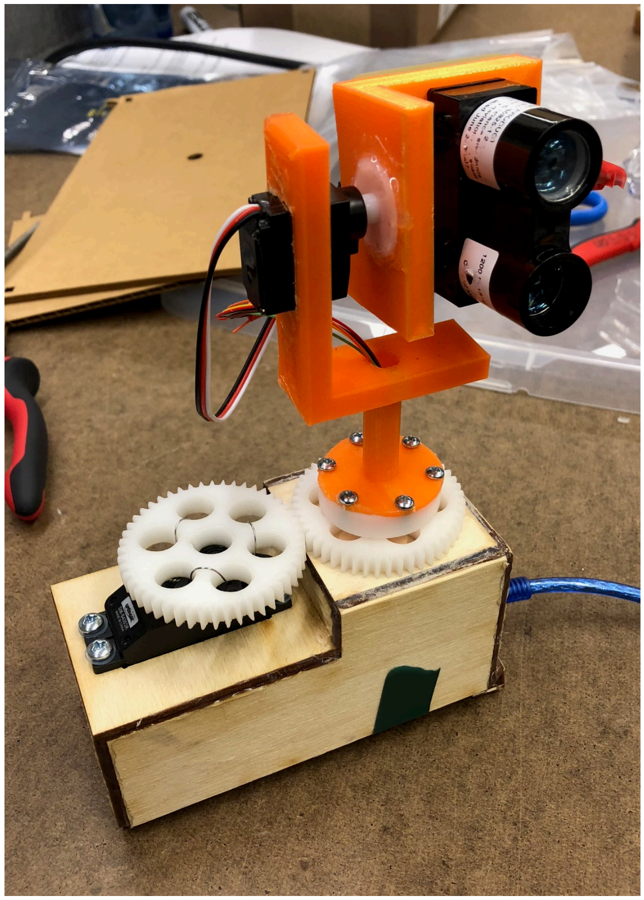


Figure 4. Photograph of LMI version 1.0.

Code

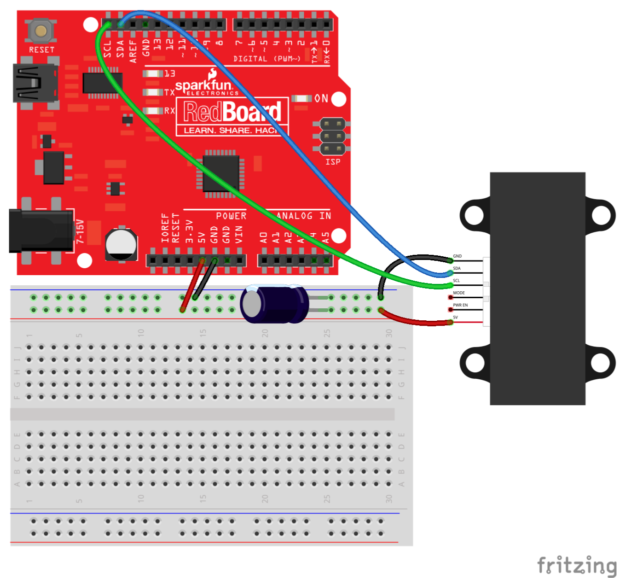
The scripts used to operate LMI can be found on this project's GitHub repository. LMI.ino runs on the Arduino IDE while LidarViewer.pde (by Charles Grassin) runs on Processing. LMI.ino is based on a template called LidarTurret.ino by Charles Grassin, which is also included in the repository. The links to the libraries used in these programs (Servo, LIDARLite, and Wire) can be found in the References section of this paper. To plot LMI's data, we used Wes Watter's (2018) cloudplot.py, which constructed not only point cloud maps but also constructed a digital elevation model (DEM) that can be applied to the path-finding algorithm.

Process and Log

The following is a more streamlined version of LMI's project log. An informal version, written in bullet point format, can be found as a PDF on this project's GitHub repository.

1. Received the Lidar Lite v3, MAG 3110 (magnetometer), and MPU 6050 (IMU) from SparkFun and began research on operating specifications. Ordered an Arduino Nano from Amazon.com.

2. Downloaded the Arduino IDE from the Arduino website with the help of a SparkFun tutorial. Also downloaded FTDI drivers so that my Mac OS X High Sierra operating system can use the Arduino IDE.
3. Made my first Arduino circuit using the Lidar Lite v3 Hookup Guide from SparkFun with the following circuit diagram:



4. Investigated SLAM algorithms and determined that applying this algorithm would be too complex for the scope of this project. With that being said, it would be interesting to apply these to this instrument in the future.
5. Learned about a program called SlugView, which is a free point cloud analysis software on OpenTopography. However, for our purposes, we realized that using a Python script would also plot our point clouds.
6. Learned about the differences between stepper and servo motors and came to the conclusion that servo motors would be easier to use than stepper motors.
7. Received Arduino Nano from Amazon, but after endless troubleshooting with my computer I realized that the Nano was a clone. In the end I purchased another Nano from the Arduino website.
8. Learned how to solder and began working with the MPU 6050. However, I realized that it wasn't working given everything that I had tried. Future iterations of this project should investigate how to implement the MPU 6050 into this scanner. Successfully programmed the MAG 3110 but determined that it wouldn't have much use without the MPU 6050.
9. Ordered a slip ring, continuous rotation servo, Parallax Feedback 360°™ servo, and mini servo.
10. Received all of the previously ordered products and constructed the first LMI prototypes.
11. Designed mount housing in SolidWorks.
12. Laser-cut and 3D printed housing components.
13. Constructed LMI v. 1.0 while simultaneously soldering circuit together on ¼ and ½ size protoboards.
14. Created a scan of the WeLab and of the Outdoor Classroom by the Observatory.

Preliminary Data

Since the LMI was not installed on the rover, it did not collect data during a “drive” around an outcrop; there was simply not enough time to integrate all of the individual components into the rover. However, LMI was able to complete a trial scan of Wellesley’s Outdoor Classroom, which is located just outside of the Observatory. This location was chosen because it was: 1) nearby and easily accessible, 2) had interesting structures in the field of view (e.g. a tree, a slope), and 3) had randomly distributed rocks that could be used as a proxy for obstacles.

The full scan swept 0 to 75° in ϕ (altitude) and 0 to 180° in θ (azimuth). The scan took 25 minutes to complete one whole sweep. The plotted point cloud and DEM are shown in Figure 5. Both plots were created using cloudplot.py (by Wes Watters (2018), available in the repository). All point data recorded as of 5/22/2018, including Test 3 b (the data plotted in Figure 5) are also available in the repository.

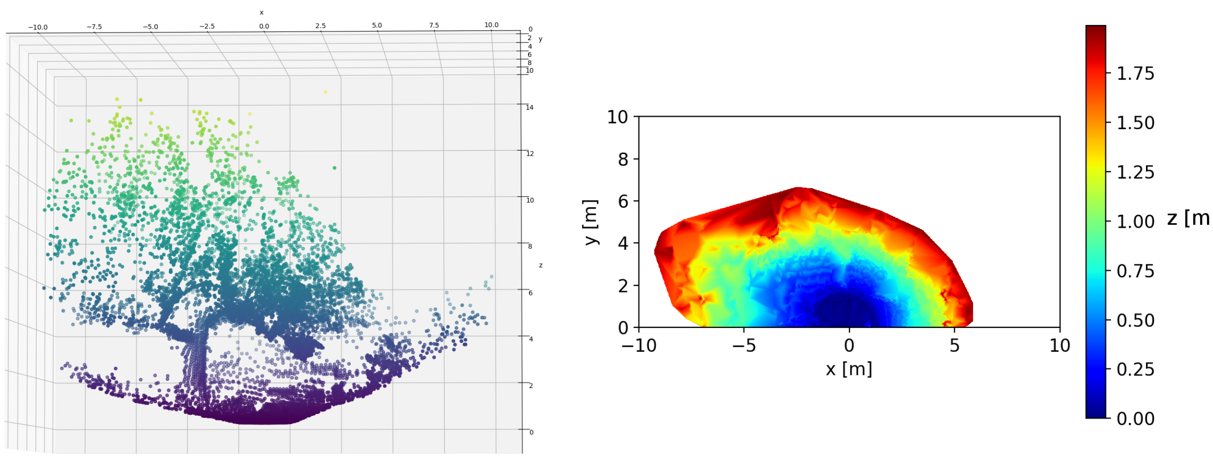


Figure 5. 3D point cloud (left) and top-down view of the scan < 2 m from the surface (right).

Discussion and Future Paths

The LMI was able to successfully make multiple scans of both indoor and outdoor structures (the plots in Figure 4 are only one of many more scans). However, the LMI needs to take data during a rover drive. In order to get closer to achieving this goal, future workers need to perform the following tasks, in order of priority:

1. Set up communication between ScOut’s Rasberry Pi and LMI’s Arduino Nano. The Nano should be sending LMI data directly to the RasPi either as a stream of data from the Serial Monitor or as a text (.txt) file that is compiled on the Nano and then sent to the RasPi for processing. This communication also involves setting up the Pi to tell the Nano when to start and stop scans.
2. Implementing the inertial measurement unit (IMU; sensor: MPU 6050) and the magnetometer (sensor: MAG 3110) into the LMI circuit on the ½ size protoboard. The MAG 3110 would operate under the control of the MPU 6050, and the MPU 6050 would send acceleration, gyroscope, orientation, and magnetic bearing data to the Arduino

Nano, which would then be sent to the RasPi. These data act as calibrations that would allow the map to be properly rendered and oriented after multiple scans. Although we were able to get the MAG 3110 to work (using the SparkFun MAG 3110 Magnetometer Breakout Library), we were unable to get the MPU 6050 working after multiple attempts. Links to "starting points" for the MPU 6050 in the form of Arduino forum posts and I^2Cdev forums can be found in the References section. The issue might potentially lie in how the Arduino addresses the sensor (0 x 68 versus 0 x 69), and there might be a way to easily fix this in Jeff Rowberg's example code from his MPU 6050 library.

- It is important to note that we were able to control the MAG 3110 sensor independently of the MPU 6050; the final iteration of the MAG 3110 sensor needs to be connected to the MPU 6050. This is another challenge.
- The MAG 3110 also needs to be programmed to record magnetic bearings (e.g. programmed as a compass) instead of raw magnetic field data.
- If the MPU 6050 is a bust, an ADXL 3xx three-axis accelerometer sensor in combination with a three-axis gyroscope sensor would be an acceptable replacement.

3. Implementing the Parallax Feedback 360°™ continuous rotation servo onto the LMI. This servo allows 360° azimuth rotation while simultaneously providing position feedback. This not only allows for full 360° views of the rover's surroundings, but it would in theory allow our Arduino program to tell the servo (and thereby the Lidar sensor) to move to a specific position that is determined by the direction of the magnetic bearing provided by the MAG 3110 sensor. However, this servo does not have much documentation related to Arduino, as documentation from the website is written in Java. Some Arduino users like JavelinPoint have created code to run this servo, but it relies on PID control and is difficult to use. LMI has an extra header installed at pin A0 in order to allow for easy Parallax Feedback 360°™ installation and use with JavelinPoint's code.
    - An alternative is implementing a stepper motor with a full step size of 0.9°. The advantage of this motor is that it can be programmed to move in smaller than 1 degree increments of θ. However, the motor does not provide position feedback unless it has an encoder.
4. Improving the LMI housing and stability, which includes but is not limited to:
    - 3D printing the main mount housing (this housing is currently made out of plywood)
    - Using wood-joinery techniques to put housing pieces together instead of hot glue.
    - Shifting the Lidar sensor position on the altitude servo so that the sensor can move freely from 0 to 180° in altitude (assuming that the Parallax Feedback 360°™ servo or a stepper motor cannot be installed with LMI)
    - Purchasing a longer micro-USB to USB cable that can both charge and transmit data. The previously purchased cable is not the right size micro-USB for the Nano.
    - Creating more secure wire connections that can still be taken apart (ex. if the LMI needs to be taken apart piece by piece). The current setup has the electronic wiring and the physical mount deeply intertwined; pieces of the physical mount cannot be removed without removing the wires, and vice versa.

References

LMI GitHub Repository

Reahl, J. LMI-Lidar-Mapping-Instrument, *GitHub*, accessed May 22, 2018, https://github.com/jreahl/LMI-Lidar-Mapping-Instrument.

Lidar Background

"What is LIDAR?" *NOAA*, https://oceanservice.noaa.gov/facts/lidar.html.

Corrigan, Fintan. "12 Top LIDAR sensors for UAVs and So Many Great Uses," *DroneZon*, January 17, 2018, https://www.dronezon.com/learn-about-drones-quadcopters/best-lidar-sensors-for-drones-great-uses-for-lidar-sensors/.

Cameron, Oliver. "An Introduction to LIDAR: The Key Self-Driving Car Sensor," *Voyage*, May 9, 2017, https://news.voyage.auto/an-introduction-to-lidar-the-key-self-driving-car-sensor-a7e405590cff.

NEON Science. "How Does LiDAR Remote Sensing Work? Light Detection and Ranging," *Youtube*, November 24, 2014, https://www.youtube.com/watch?v=EYbhNSUnIdU.

Marcoe, K. "LIDAR: an Introduction and Overview," *Portland State University*, Fall 2007, http://web.pdx.edu/~jduh/courses/Archive/geog481w07/Students/Marcoe_LiDAR.pdf.

Lidar Mount Examples

TechsOnTube. "LIDAR-Lite v3 – 3D-Scan," *Youtube*, November 9, 2016, https://www.youtube.com/watch?v=gCpCGkwwy8I.

Guy, A. "LIDAR Sensor – 3D Printed Rotating Platform – Robotic Vehicle #8," *Youtube*, March 10, 2017, https://www.youtube.com/watch?v=RHPAJ3kjc3Q.

Guy, A. "Rotating 360 LIDAR Scanner – 3D Printed up and down scanning – Robotic Vehicle #9," *Youtube*, May 6, 2017, https://www.youtube.com/watch?v=zizrY-cOS9k.

Grau, A. "DIY 360 DEGREE REALTIME OUTDOOR LIDAR WITH ROS SUPPORT (GRAULIDAR)," *Don't worry be creative*, 2015, http://grauonline.de/wordpress/?page_id=1233.

Grassin, C. "3D LIDAR SCANNER," *Charles' Labs*, October 10, 2017, http://www.charleslabs.fr/en/project-3D+Lidar+Scanner.

Product Pages

Garmin Lidar Lite v3, *SparkFun*, https://www.sparkfun.com/products/14032.

SparkFun Triple Axis Accelerometer and Gyro Breakout – MPU-6050, *SparkFun*, https://www.sparkfun.com/products/11028.

SparkFun Triple Axis Magnetometer Breakout – MAG3110, *SparkFun*, https://www.sparkfun.com/products/12670.

Arduino Nano, *Arduino*, https://store.arduino.cc/usa/arduino-nano.

Adafruit Perma-Proto Quarter-sized Breadboard PCB – 3 Pack!, *Adafruit*, https://www.adafruit.com/product/589.

Adafruit Perma-Proto Half-sized Breadboard PCB – 3 Pack!, *Adafruit*, https://www.adafruit.com/product/571.

Slip Ring with Flange – 22mm diameter, 6 wires, max 240 V @ 2A, *Adafruit*, https://www.adafruit.com/product/736.

Servo – Generic (Sub-Micro Size), *SparkFun*, https://www.sparkfun.com/products/9065.

Servo – Generic High Torque Continuous Rotation (Standard Size), *SparkFun*, https://www.sparkfun.com/products/9347.
Servo – Generic High Torque (Standard Size), *Sparkfun*, https://www.sparkfun.com/products/11965.
Parallax Feedback 360° High Speed Servo, *Parallax Inc.*, https://www.parallax.com/product/900-00360.

Software

Arduino Libraries. Servo Library for Arduino, *GitHub*, accessed May 22, 2018, https://github.com/arduino-libraries/Servo.
ESP8266 Community Forum. Wire.h, *GitHub*, accessed May 22, 2018, https://github.com/esp8266/Arduino/tree/master/libraries/Wire.
Garmin International. LIDAR-Lite v3 Arduino Library, *GitHub*, accessed May 22, 2018, https://github.com/garmin/LIDARLite_v3_Arduino_Library.
Arduino IDE 1.8.5, *Arduino*, https://www.arduino.cc/en/Main/Software.
Processing 3.3.7, *Processing*, https://processing.org/download/.
Rowberg, J. MPU 6050 Arduino Library, *GitHub*, accessed May 22, 2018, https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050.
SparkFun Electronics. SparkFun MAG3110 Breakout Board Arduino Library, *GitHub*, accessed May 22, 2018, https://github.com/sparkfun/SparkFun_MAG3110_Breakout_Board_Arduino_Library.
Fritzing 0.9.3b, *Fritzing*, http://fritzing.org/download/.
I2CScanner, *Arduino*, https://playground.arduino.cc/Main/I2cScanner?action=sourceblock&num=1.
Grassin, C. LidarTurret.ino, *Charles' Labs*, October 10, 2017, http://www.charleslabs.fr/en/project-3D+Lidar+Scanner.
Grassin, C. LidarViewer.pde, *Charles' Labs*, October 10, 2017, http://www.charleslabs.fr/en/project-3D+Lidar+Scanner.

LIDAR Lite v3

"Lidar Lite v3 Operation Manual and Technical Specifications," *Garmin International*, https://static.garmin.com/pumac/LIDAR_Lite_v3_Operation_Manual_and_Technical_Specifications.pdf.

MPU 6050

"MPU-6000 and MPU-6050 Product Specification Revision 3.4," *InvenSense Inc.*, August 19, 2013, http://43zrtwysvxb2gf29r5o0athu.wpengine.netdna-cdn.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf.
"Arduino Playground: MPU-6050 Accelerometer + Gyro," *Arduino*, https://playground.arduino.cc/Main/MPU-6050#intro.
Sanjeev, A. "How to Interface Arduino and the MPU 6050 Sensor," *Maker Pro Beta*, https://maker.pro/arduino/tutorial/how-to-interface-arduino-and-the-mpu-6050-sensor.
Arduino Forum Topic: mpu6050 problem, *Arduino*, thread started January 13, 2013, http://forum.arduino.cc/index.php?topic=142490.0.

Arduino Forum Topic: Easily Changing I2C Address Of MPU6050 Accelerometer + Gyroscope, *Arduino*, thread started July 14, 2014, https://forum.arduino.cc/index.php?topic=255033.0.
Rowberg, J. "MPU-6050 6-axis accelerometer/gyroscope," *I^2Cdevlib*, https://www.i2cdevlib.com/devices/mpu6050#help.
"Wiring the MPU 6050 Sensor Mems Accelerometer GYRO," *14Core*, http://www.14core.com/wiring-the-mpu-6050-sensor-accelerometer-gyro-with-nano/.

Parallax Feedback 360°™
Arduino Forum Topic: Controlling Parallax Servo 360 in a certain angle, *Arduino*, thread started December 13, 2017, https://forum.arduino.cc/index.php?topic=516986.0.

Tutorials
b_e_n. "Installing Arduino IDE," *SparkFun*, https://learn.sparkfun.com/tutorials/installing-arduino-ide.
Smith, P. "How to Install FTDI Drivers: Mac," *SparkFun*, https://learn.sparkfun.com/tutorials/how-to-install-ftdi-drivers/mac.
Hymel, S. "LIDAR-Lite v3 Hookup Guide," *SparkFun*, https://learn.sparkfun.com/tutorials/lidar-lite-v3-hookup-guide.
b_e_n. "What is an Arduino?" *SparkFun*, https://learn.sparkfun.com/tutorials/what-is-an-arduino.