

POE Lab 3: Line Following Robot

Junwon Lee, Alex Bahner

09/24/2018

1 Introduction

In this lab we built a 3d Scanner based on pan-tilt mechanism. The scanner is composed of an infrared sensor and two servo motors. The Arduino reads the angles of two servos along with the measurement from the infrared sensor and records the data. Then, the python script uses the data to plot the points detected by the infrared sensor.

2 Source Code

The Arduino controls the servos and reads the IR sensor and sends that data to the computer, which is read in Python and turns the data from the Arduino into a scatter plot. The Arduino runs the nested for loops so that it scans up in the y axis and shifts slightly in the x axis and repeats the y-axis rotation scan. At each individual point, it collects the value from the IR sensor along with the angular position of two servos. Ultimately, we have raw data for the spherical coordinate (r, θ, ϕ) , where r is non-calibrated IR sensor value and θ and ϕ are the angle values from two servos.

```
for (pos1 = 40; pos1 <= 80; pos1 += 1) {
  myservo1.write(pos1);
  for (pos2 = 100; pos2 <= 150; pos2 +=1){
    int val = analogRead(sensorPin);
    Serial.print(val); Serial.print(",");
    Serial.print(pos1); Serial.print(",");
    Serial.println(pos2);
    myservo2.write(pos2);
    delay(100);
  }
  for (pos2 = 150; pos2 >= 100; pos2 -= 1) { // goes from 180 degrees to 0 degrees
    myservo2.write(pos2);                // tell servo to go to position in variable 'pos'
    delay(10);                          // waits 15ms for the servo to reach the position
  }
  delay(500);
}
```

In Python script, we parsed the serial data and was able to take out the IR sensor value along with angles of servos 1 and 2. We were then able to take the IR sensor data and apply our calibration equation which was a log-log of the calibrated data shown in this figure below. The equation was

```
Dis = (val**-1.1080) * math.exp(9.9496)
```

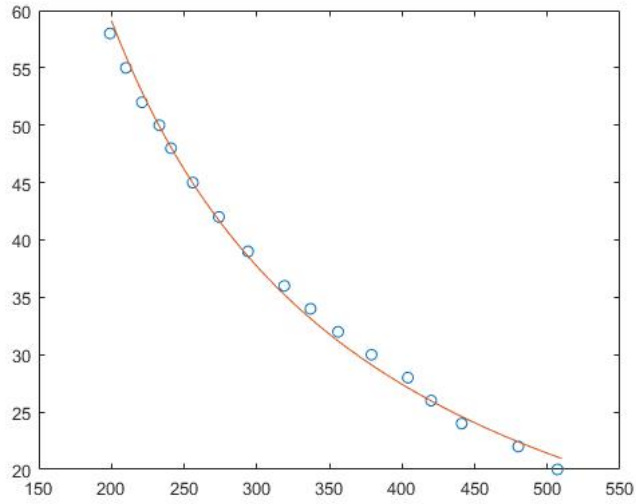


Figure 1: Calibration Graph

Using the distance, we can take our data that's in the spherical coordinate and convert it into Cartesian coordinate using the following scripts:

```
x = Dis*sin(radians(yangle))*cos(radians(xangle))
y = Dis*sin(radians(yangle))*sin(radians(xangle))
z = Dis*cos(radians(yangle))
```

Then, using the Cartesian coordinate values, we plotted the slice across the letter along with the full letter using matplotlib's scatterplot. This is a slice for the Letter B

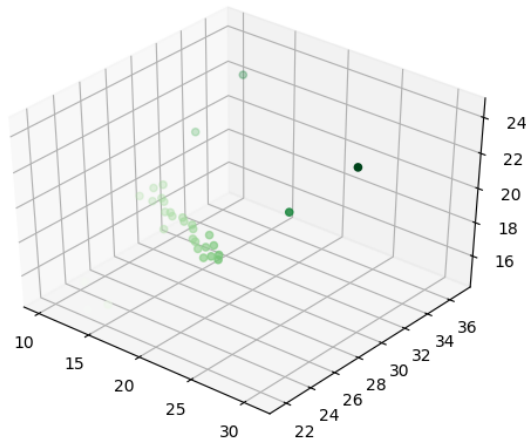


Figure 2: Slice of B

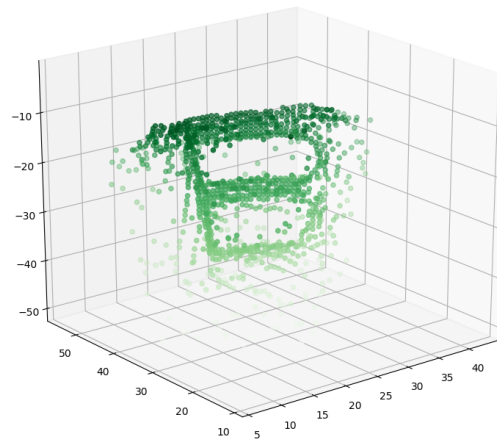


Figure 3: Full B

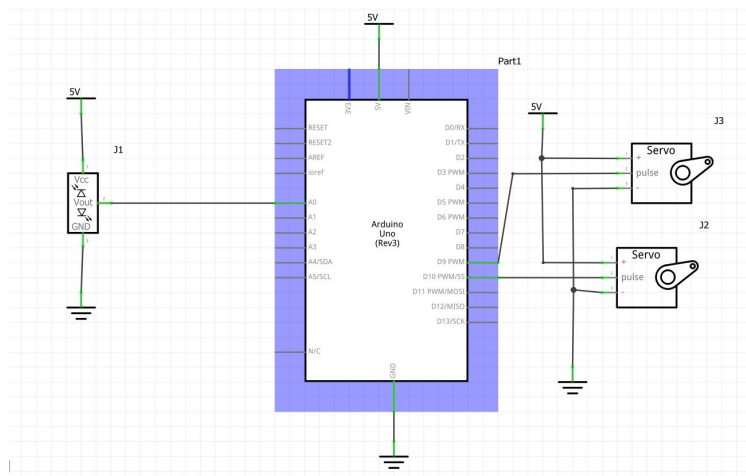


Figure 4: Schematic for this Lab

3 Circuit Diagram

4 Reflection

We started off the lab with decent pace, where we calibrated the sensor and had the mechanical mount printed by the first week. However, because one of us was sick for few days, we slowed down and had to finish the scan in the last couple of days. Fortunately, we still finished the project on time.

5 Full Code

Arduino code

```

#include <Servo.h>

Servo myservo1; // create servo object to control a servo
Servo myservo2;
// twelve servo objects can be created on most boards

int pos1 = 0; // variable to store the servo position
int pos2 = 0;

int sensorPin = 0; //analog pin 0

void setup() {
  Serial.begin(9600);

  myservo1.attach(9); // attaches the servo on pin 9 to the servo object
  myservo2.attach(10);

  myservo1.write(30);
  myservo2.write(80);
  delay(5000);
}

void loop() {
  // put your main code here, to run repeatedly:
  for (pos1 = 40; pos1 <= 80; pos1 += 1) {
    myservo1.write(pos1);
    for (pos2 = 100; pos2 <= 150; pos2 +=1){
      int val = analogRead(sensorPin);
      Serial.print(val); Serial.print(",");
      Serial.print(pos1); Serial.print(",");
      Serial.println(pos2);
      myservo2.write(pos2);
      delay(100);
    }
    for (pos2 = 150; pos2 >= 100; pos2 -= 1) { // goes from 180 degrees to 0 degrees
      myservo2.write(pos2); // tell servo to go to position in variable 'pos'
      delay(10); // waits 15ms for the servo to reach the position
    }
    delay(500);
  }

  for (pos1 = 90; pos1 >= 40; pos1 -= 1) { // goes from 180 degrees to 0 degrees
    myservo1.write(pos1); // tell servo to go to position in variable 'pos'
    delay(500); // waits 15ms for the servo to reach the position
  }
}

```

Python script

```

import serial
import math
from math import *
import csv
import matplotlib.pyplot as plt

```

```

from mpl_toolkits.mplot3d import Axes3D

arduinoComPort = "/dev/ttyACM0"
baudRate = 9600
serialPort = serial.Serial(arduinoComPort, baudRate, timeout=1)
data_x = []
data_y = []
data_z = []
Zs = []

for x in range(2000):
    lineOfData = serialPort.readline().decode()
    if len(lineOfData) > 0:

        val, xangle, yangle = (int(x) for x in lineOfData.split(','))

        Dis = (val**-1.1080) * math.exp(9.9496)
        print("d = " + str(Dis), end="")
        print(" d = " + str(xangle), end="")
        print(" d = " + str(yangle))

        x = Dis*sin(radians(yangle))*cos(radians(xangle))
        y = Dis*sin(radians(yangle))*sin(radians(xangle))
        z = Dis*cos(radians(yangle))
        if Dis < 60 :
            data_x.append(x)
            data_y.append(y)
            data_z.append(z*-1)

fig = plt.figure()
ax =fig.add_subplot(111, projection='3d')
ax.scatter(data_x,data_y,data_z, c=data_z, cmap = 'Greens')
plt.show()

```