EE 183DA - TEAM BUFFALO - LAB 1 REPORT

AUTHOR: IOU-SHENG (DANNY) CHANG

Date: January 24th 2019.

Contents

1. Introduction

The purpose of this lab is to build and characterize a 2 wheeled robot containing simple instrumentation. This instrumentation comes in the form of sensors and is to be used for the purpose of determining the state of the robot in future labs. Over the course of this lab, the objective is to develop three separate models of input-output dynamics. All code used modified, or created will be uploaded into the team's git repository in order for it to be easily accessible to the reader of this report, or anyone else who may have an interest in what is done in this lab. Additionally, pictures, videos, and data will be uploaded to the git repository.

The microcontroller of this robot is an ESP8266. It is programmed by a computer and can be coded such that it sends data from some or all of its sensors back to the computer it is wired too. This ESP8266 has a Wi-Fi module built into it, so it can also be configured to wirelessly communicate with other devices over Wi-Fi. The two wheels on this two wheel car are connected directly to continuous servos, with model FS90R. In this way, the car can be driven either forwards or backwards, as well as turned in any direction, by varying the PWM signal to the servos. The car contains two laser range finders, specifically the model GYVL53LOX, placed on the front and side of the car. The position of the car in a closed environment (such as a box) can be determined using these two sensors; the car stays in 2D space and therefore only 2 values are required for determining the car's translation in 2D space. The other type of sensor used is a magnetometer, specifically the MPU9250 IMU. This sensor, once calibrated, gives the value of the angle between the vector normal to the front edge of the sensor and magnetic north. It is used to determine what angle of rotation about the Z-axis of the car at any given time. The motor driver breakout board for the ESP8266 chip is also used in this lab. This breakout board allows the continuous servos to be plugged in and controlled without soldering directly to the ESP8266 chip. It also allows the two laser range finder sensors and the magnetometer to connect to the ESP8266's pins without being soldered on. Powering the robot when it is not connected to a computer is a rechargeable lithium-ion battery. The body of the robot itself is made out of strategically designed and folded sheet of paper.

The radius of our robots paper wheels is about 24 mm, with the two wheels 99mm apart and mounted one on each side to the front of the car. It's third contact point with the ground for stability is a tail that it drags 75 mm behind the front wheel center line.

In order to model the equations describing the input-output characteristics of the sensors, sample code is adapted and used to calibrate and collect data on the sensors, such that two data driven models, one describing the relation of the wheel speed to the PWM values sent to the servos, and the other describing the sensor output that will result in the input value. In total, the sensors on the robot produce 3 output values, one from the side laser range sensor, one from the front laser range sensor, and one from the magnetometer. The models of each individual sensor will be used in the subsequent labs in order to build a state estimator for the robot.

2. Symbols and Conventions

2.1. Symbols.

List of all symbols used for deriving the theoretical, geometric model.

W	distance between left and right wheels
D	distance between rotational center and car reference point
v	linear velocity of the car
v_L	linear velocity of the left wheel
v_R	linear velocity of the right wheel
ω	angular velocity of the car
ω_L	angular velocity of the left wheel
ω_R	angular velocity of the right wheel
p_x	position: absolute x coordinate of the car
p_y	position: absolute y coordinate of the car
θ	absolute orientation of the car
θ_{MPU}	MPU9250 magnetometer angle measurement
l_f	front laser range sensor reading
l_r	right laser range sensor reading
δt	time traveled by the car
δt_L	time traveled by the left wheel
δt_R	time traveled by the right wheel
N_v	linear velocity noise
N_ω	angular velocity noise

2.2. Assumptions.

The following assumptions are made according the the car's geometry and for the derivation of theoretical, geometric model.

(1) (r_x, r_y) is taken as the car's reference point, which is in the top right corner of the car.
(2) When the car is going straight (either forward or backward), $v = v_L = v_R$, having same sign and magnitude.
(3) When the car is turning (either clockwise or counter-clockwise), it is turning about its rotational center, which is the mid-point between the two wheels. In this case, $|v| = |v_L| = |v_R|$, however v_L and v_R have opposite sign.
(4) $\theta = \theta_{MPU}$ assuming the MPU9250 magnetometer is calibrated.

2.3. **Systems.**

Input

$$u = \begin{bmatrix} v_L \\ v_R \end{bmatrix}$$

State

$$x = \begin{bmatrix} p_x \\ p_y \\ \theta \end{bmatrix}$$

Sensor Measurements

$$y = \begin{bmatrix} l_f \\ l_r \\ \theta_{MPU} \end{bmatrix}$$

3. Methods

3.1. Set-Up.

To get started with the lab, we have to assemble the paperbot into its physical structure before any work with the electronics can be done. The given paper frame was folded according to the document given, which can also be found in the git repository under the name Paperbot Assembly. During this process, tape was added into places on the car as it was seen fit, both for the purpose of holding the car together and for adding structural integrity to the car. The end result was a folded car with the battery on the bottom and the ESP8266 module plugged into the breakout board on top. Once the car was set-up, it was necessary to set-up one or more computers belonging to team members to be able to modify, create, and upload programs to the ESP8266 chip from the Arduino IDE. This process is detailed in the Paperbot Assembly file that can be found in the git repository and can be summarized as the adding of the necessary libraries and board information for the ESP8266 module to work on either a new or already existing install of the Arduino IDE. Team Buffalo's paperbot assembly work is shown in Figure 1

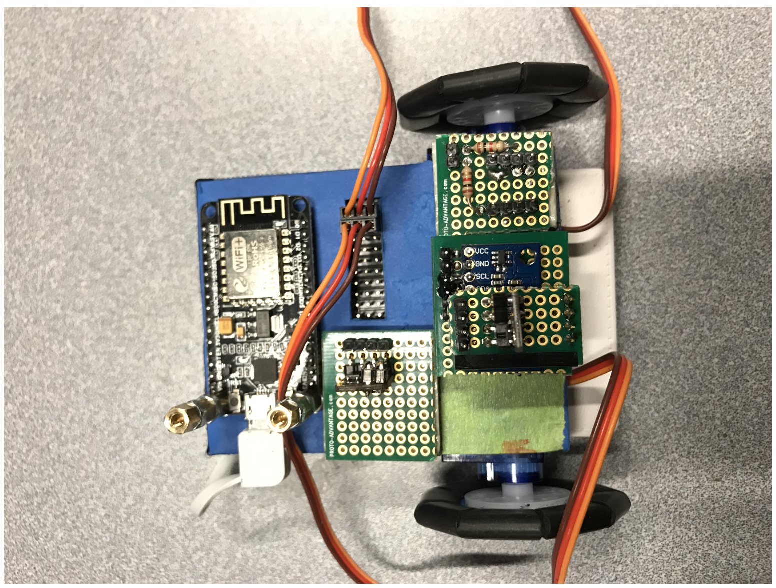

Figure 1. Team Buffalo's paperbot

3.2. Robot state update dynamics.

In order to generate a theoretical model of the system dynamics, the car's movements were separated into movements in a straight line and movements in a circle (turning). In order to make the calculations for the model of the car while turning easier, the magnetometer was placed on the car such the point from which it measured angle was the center of the line between the front wheels. This meant that the behavior (always turning about its rotation axis) during rotation was same whether it was going clockwise or counterclockwise.

3.3. **Data-driven actuator model.**

In order to get the car running, servos were then being set up. The goal was to find the input-output relationship, where the the input is the PWM signal sent to the servos from the ESP8266 chip and the output is the speed at which the servos rotate when they are sent that PWM signal. The sample code for setting up a network on the ESP8266 was used and the corresponding web page to wireless control the servos was used for all testing and adjustment of the PWM signal sent to the servos in this section. As the car will be driving in both a straight line and turning, it was necessary to generate the input-output relationship for the cases of going in a straight line, forward and backward, and turning about the rotational axis, clockwise and counterclockwise.

The first method attempted to gather data was placing the car on its side so the left wheel was up in the air, and then record the time it took for that wheel to spin 20 revolutions in both the forward and backward directions at different PWM values, thereby allowing the servo's angular velocity and linear velocity at different PWM signals to be calculated from the data collected. The process was then repeated with the right wheel in the air. Set-up is shown in Figure 2

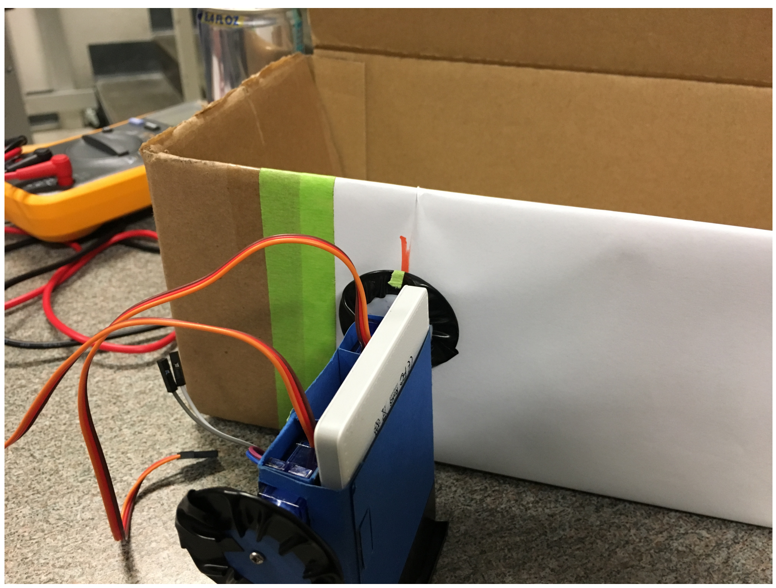

Figure 2. Servo freespin setup

Using the figure as shown in Figure 3, it should have been theoretically possible to draw a horizontal line across the graph at the desired speed, and then use the PWM signals given by that line's intersection with the trend curve of the data to move forward in a straight line at any speed desired, or move in a circle about the center of the car, given by the midpoint of the line between the two servos. The straight line movement would be done by choosing values such that both servos rotated forward at the same speed, where as the circular movement would be done by choosing values such that both servos rotated in opposite directions at the same speed. When this process was carried out in an attempt to verify the model, however, it proved that the model was faulty. The car did not move in a straight line or in a perfect circle when the model said it was supposed to. After considering the way the data was collected, the team came to the conclusion that the data collected with the wheel in the air was not valid data, as it represented an idealized and unrealistic situation for the servos operation. Specifically, determining car speed with the wheels in the air did not take into account of noise seen at normal driving, specifically the lack of friction of the wheels, resulting in wheel slippage, as well as the weight of the car on the wheels.

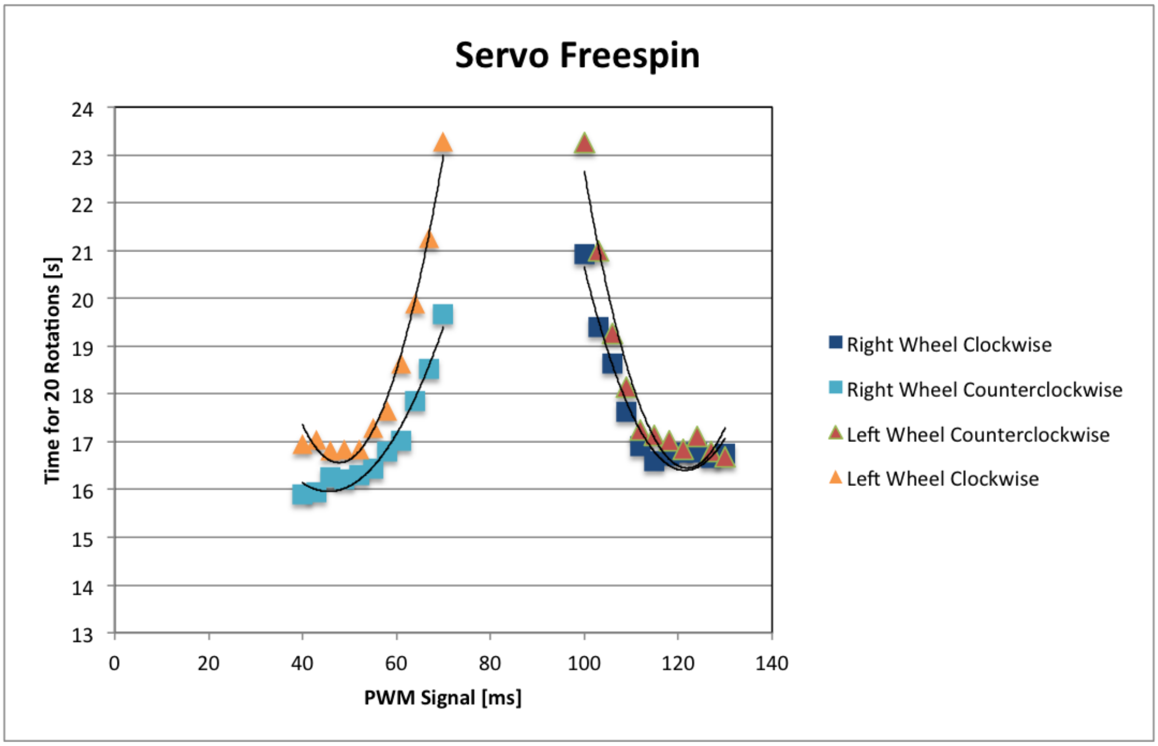

Figure 3. Data Collected from the Servo freespin experiment

After determining that the previous set of data collected was faulty, the team altered its method for collecting data on how the value of the PWM signal relates to servo speed. The car was placed on a flat table, with a start and finish marked out and the distance between them measured, setup as shown in Figure 4. Then the PWM signal of the right servo was fixed, and the PWM signal of the left servo was found through experimentation such that the car would travel in a straight line when both the right and left PWM signals were sent to the right and left servos, respectively, at the same time. Once a matching pair of PWM signals was found, the time it took the car to travel the set distance of 45 centimeters was recorded and then used to calculate the speed of the car at that pair of PWM signals. This process was used to find 5 pairs of PWM signals that moved the car in a straight line going forward and 5 pairs of PWM signals that moved the car in a straight line going backward. The same logic was used to find pairs of PWM signals that resulted in the car spinning in a circle such that the axis of rotation was the midpoint of the line between the two servos on the car.

As this data was recorded under realistic car driving scenarios, it can be used to create a model relating PWM input to a servo to the speed at which the servo rotates, taking into

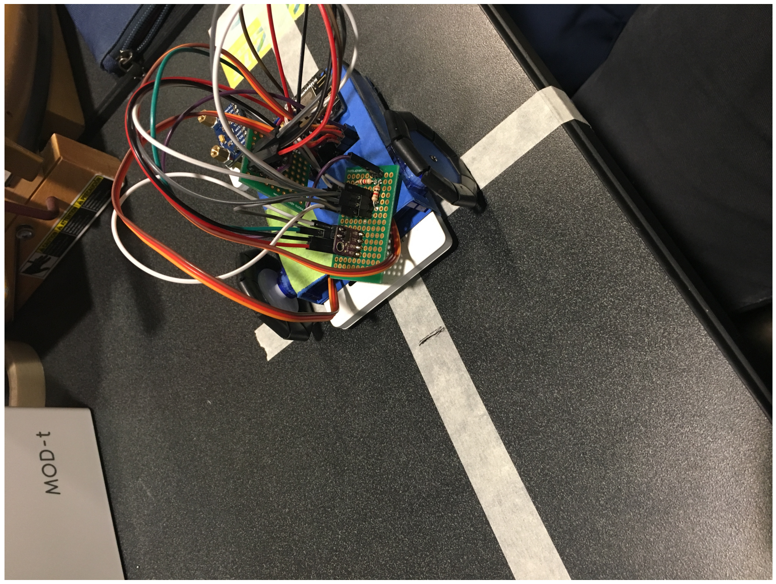

FIGURE 4. Servo experiment setup

account noise such as car weight on the servo and wheel slippage. These plots and the data that was used to create them were then used to create a data driven actuator model which translates the motor commands of PWM signals into the generated motion, with uncertainty taken into account in the model, as the data was collected in the real world scenario that takes uncertainty into account.

3.4. **Data-driven sensors model.**

3.4.1. *Laser range sensor.*

The next step of the procedure was to collect data on the laser range sensors on the car; specifically to measure the distance measured by each one of the laser range sensors and the corresponding actual distance for varying distances. Sources of noise for this step include the sensors not being exactly parallel to the object from which the distance was being measured, and then either measuring the distance to a different object than the one intended, or mis-measuring the distance to the object in front of it. Another source of uncertainty is the fact that the actual distance is measured from the object to the edge of the car, while the sensor does not necessarily sit on the edge of the car, meaning its value will have an offset.

The software used to measure these distances was a modified version of the sample code given for the laser range sensors. The code was edited so that it would print out to the serial monitor of the Arduino IDE only a continuous stream of readings from the sensor and nothing else. The purpose of this is to generate data that can be easily processed in MS Excel or a similar program after collection. As with all code used, this code can be found in the git repository for this lab.

The data was collected by placing the car a known and measured distance from an obstacle, covered by white paper to aid in the sensor's reading, per the suggestion of the documentation of the sensor's vendor found online, setup is shown in Figure 5 for Front lidar setup and Figure 6 for the Right lidar setup. The code generated for this part was uploaded to the ESP8266 chip and 200 sensor readings were recorded with the car stationary and averaged using MS Excel, to determine the average value that the sensor read with the car stationary at that known distance. This was done for both the front and the side sensors at several known distances. The corresponding data collected was used to generate a model for each sensor. Due to the fact that this data was collected in a real usage type setting, the data driven sensor model created for each sensor with this data will take into account any noise and uncertainty that comes along with this data.

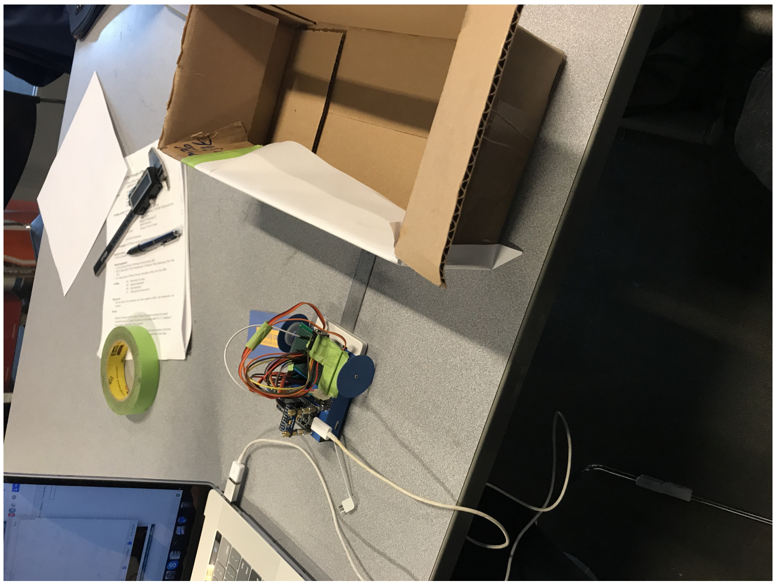

FIGURE 5. Front Lidar Experiment setup

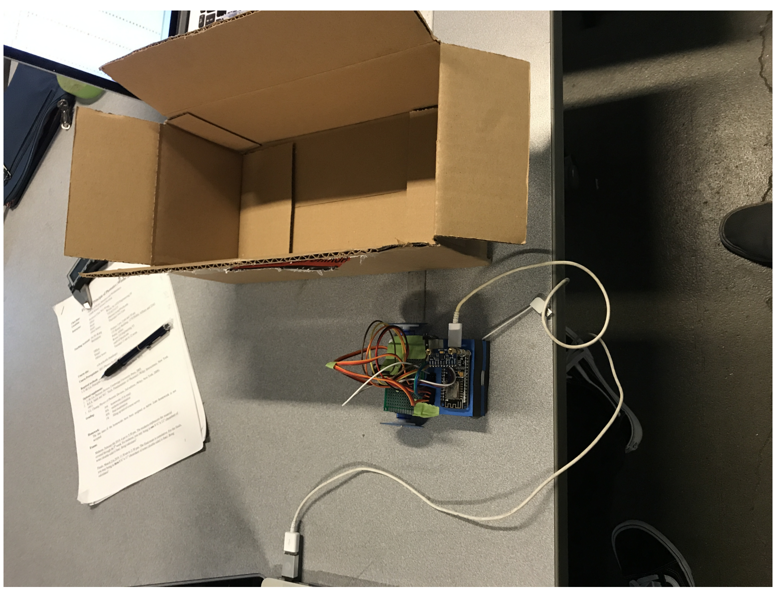

FIGURE 6. Right Lidar Experiment setup

3.4.2. *MPU9250 Magnetometer.*
Our goal is to calibrate the magnetometer and collect data for the purpose of creating a data-driven sensor model for the magnetometer. Before any data could be taken with the magnetometer, the declination was found using the online source “http://www.magnetic-declination.com/” to find what the declination angle for UCLA campus is, and then edit the code to take this angle into account taking measurements.

The next step was to calibrate the magnetometer, as these sensors typically do not come from the factory with any sort of accurate calibration. In order to calibrate the magnetometer, the code written by Kriswiner and linked by the professor was adapted. This code can be found in the git repository for this lab. The purpose of this code is to correct hard iron and soft iron biases. The hard iron biases were corrected by taking a lot of magnetometer data and then subtracting the average value of minimum and maximum value along each axis from all data collected so that the data is then centered and appears as a uniform sphere about the origin when plotted. The soft iron biases were corrected by re-scaling the magnetometer data so that the data gathered is equal in magnitude along all axes. This scaling is done using the average, minimum, and maximum value of each axis. The code displayed the six values needed to correct hard and soft iron biases, with the 3 bias values used to correct hard iron biases and the 3 scaling values used to correct soft iron biases.

Once this calibration was completed, the magnetometer’s reading was compared to that of a compass, in order to determine if an offset was needed. Since the magnetometer’s 0 degrees was perfectly aligned to the compass’s 0 degrees, as shown in Figure 7 it was determined that no offset was necessary.

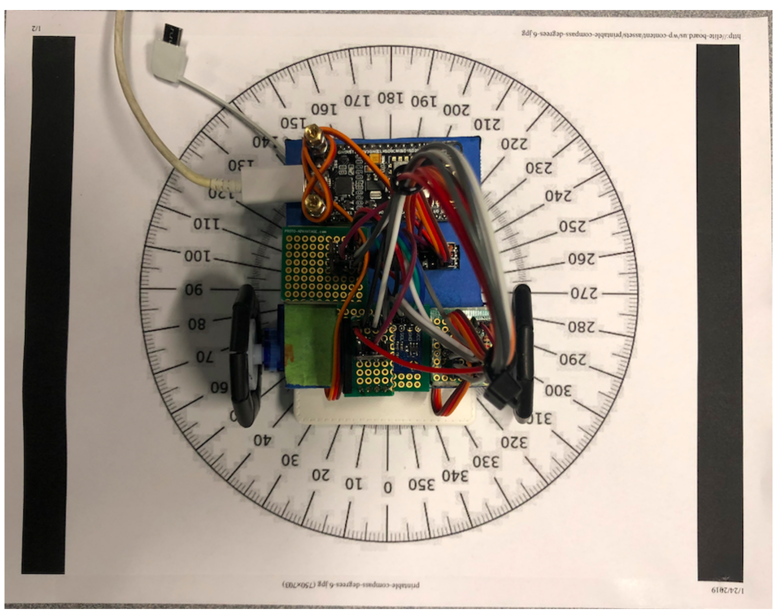

Figure 7. Right Lidar Experiment setup

4. Results

4.1. Robot state update dynamics.

The goal for this subsection is to find a theoretical, geometric model of the robot state update dynamics, which will have the form:

$$x'(t) = x_{t+1} = f(x_t, u_t, t) = Ax_t + Bu_t$$

In order to have complete control of the car, we split the movements of the car into two cases, traveling along a straight line (forward/backward) and turning about its rotational axis (clockwise/counter-clockwise).

Note that the following equations work for $\theta_t = [0, 2\pi]$, (all values of θ).

4.1.1. *Case 1: Traveling along a straight line (forward/backward).*

$$\begin{aligned} p_{x,t+1} &= p_{x,t} + (v_t + N_v)t \cdot \cos\theta_t \\ p_{y,t+1} &= p_{y,t} + (v_t + N_v)t \cdot \sin\theta_t \\ \theta_{t+1} &= \theta_t \end{aligned}$$

Expressing in matrix form,

$$x'(t) = x_{t+1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_{x,t} \\ p_{y,t} \\ \theta_t \end{bmatrix} + \begin{bmatrix} (v_t + N_v)t \cdot \cos\theta_t \\ (v_t + N_v)t \cdot \sin\theta_t \\ 0 \end{bmatrix}$$

which agrees with the form:

$$x'(t) = x_{t+1} = I_3 x_t + Bu_t = Ax_t + Bu_t$$

4.1.2. *Case 2: Turning about its rotational axis (clockwise/counter-clockwise).*

Note that $p_{x,t+1} \neq p_{x,t}$ and $p_{y,t+1} \neq p_{y,t}$ due to the fact that rotational axis is different from the car reference point

$$\begin{aligned} .p_{x,t+1} &= p_{x,t} - D[\cos\theta_t - \cos\theta_{t+1}] \\ p_{y,t+1} &= p_{y,t} + D[\sin\theta_t - \sin\theta_{t+1}] \\ \theta_{t+1} &= \theta_t + (\omega_t + N_\omega)t \end{aligned}$$

Expressing in matrix form,

$$x'(t) = x_{t+1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_{x,t} \\ p_{y,t} \\ \theta_t \end{bmatrix} + \begin{bmatrix} -D[\cos\theta_t - \cos(\theta_t + (\omega_t + N_\omega)t)] \\ D[\sin\theta_t - \sin(\theta_t + (\omega_t + N_\omega)t)] \\ (\omega_t + N_\omega)t \end{bmatrix}$$

which agrees with the form:

$$x'(t) = x_{t+1} = I_3 x_t + Bu_t = Ax_t + Bu_t$$

4.2. **Data-driven actuator model.**
The Forward experiment result is shown in Figure 8. As shown in Figure 8, the input PWM (x) - output Linear Velocity (y) relationship is characterized by

$$Leftservo : y = 0.0028x - 0.1641$$
$$Rightservo : y = -0.0029x + 0.317$$

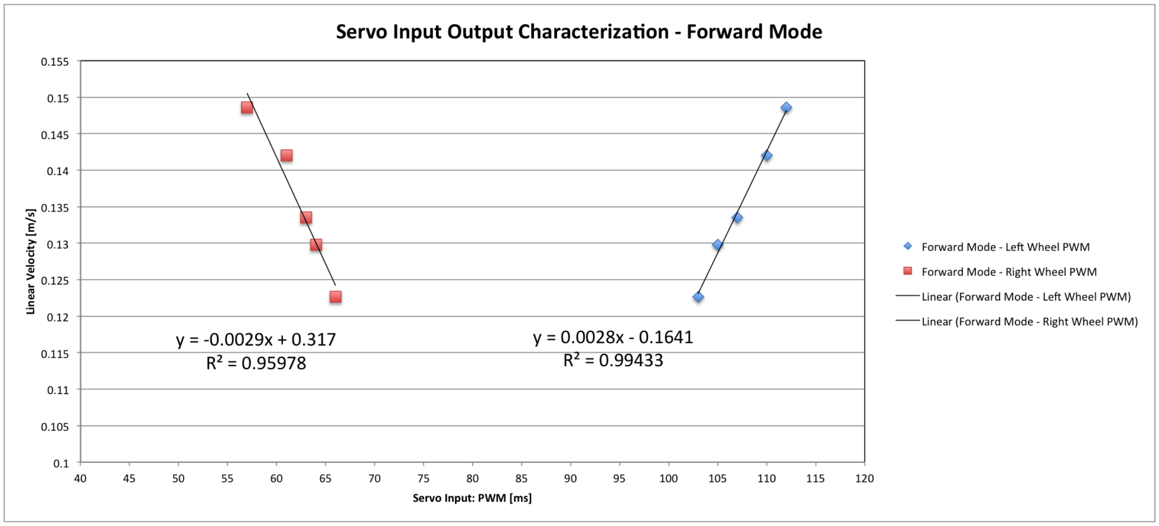

FIGURE 8. Servo Experiment: Forward Mode

The Backward experiment result is shown in Figure 9. As shown in Figure 9, the input PWM (x) - output Linear Velocity (y) relationship is characterized by

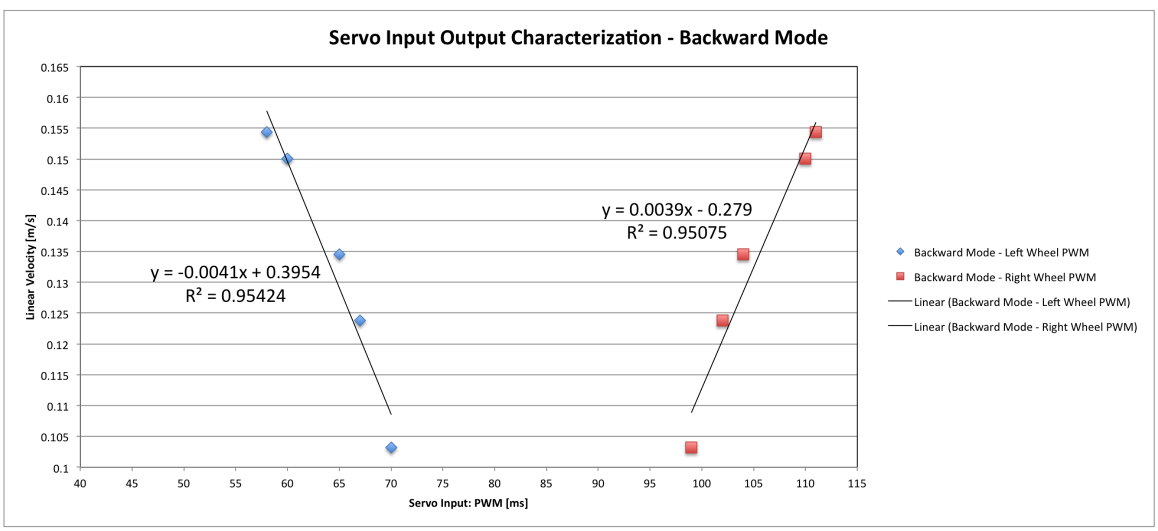

FIGURE 9. Servo Experiment: Backward Mode

$$Leftservo : y = -0.0041x + 0.3954$$
$$Rightservo : y = 0.0039x - 0.279$$

The Counter-clockwise rotation experiment result is shown in Figure 10. As shown in Figure

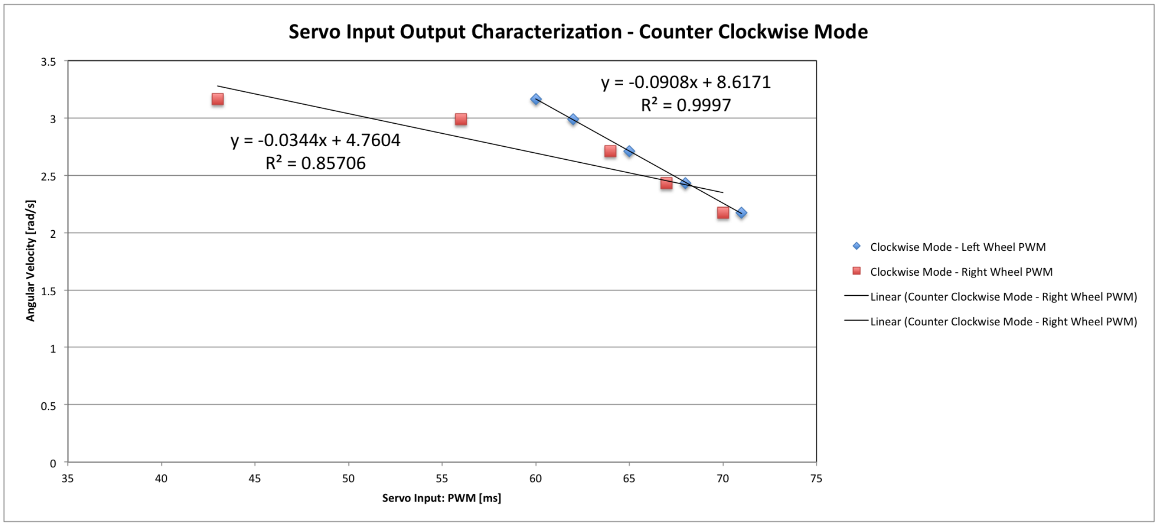

FIGURE 10. Servo Experiment: Counter-Clockwise Mode

10, the input PWM (x) - output Angular Velocity (y) relationship is characterized by

$$Leftservo : y = -0.0908x + 8.6171$$

$$Rightservo : y = -0.0344x + 4.7604$$

The Clockwise rotation experiment result is shown in Figure 11. As shown in Figure 11, the

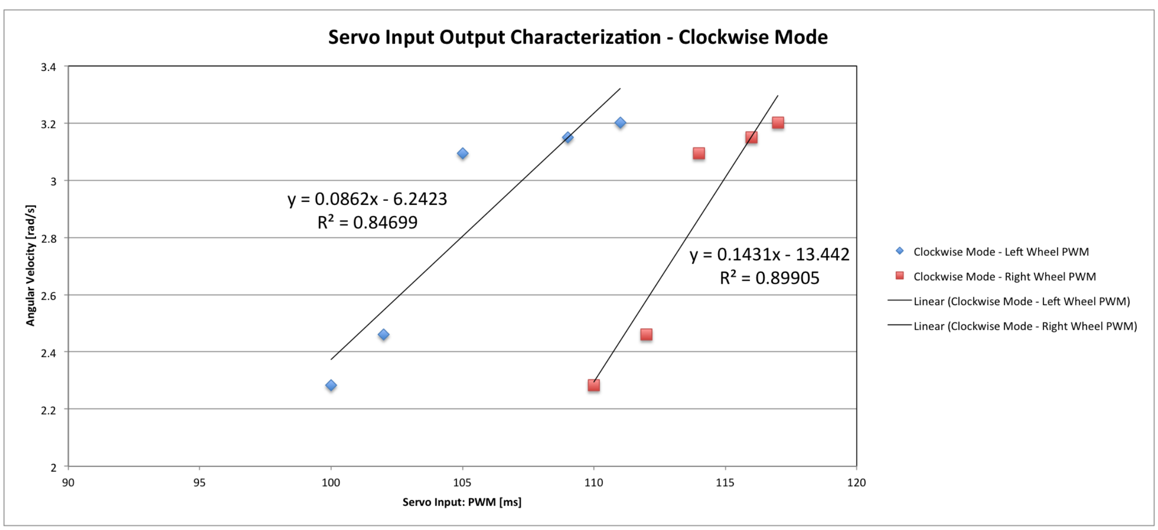

FIGURE 11. Servo Experiment: Clockwise Mode

input PWM (x) - output Angular Velocity (y) relationship is characterized by

$$Leftservo : y = 0.0862x - 6.2423$$

$$Rightservo : y = 0.1431x - 13.442$$

4.3. **Data-driven sensors model.**

4.3.1. *Laser range sensor.*
The Front laser range sensor experiment result is shown in Figure 12. As shown in Figure 12, the input distance (y) - output Lidar distance reading (x) relationship is characterized by

$$FrontLidar : y = 1.0153x + 57.117$$

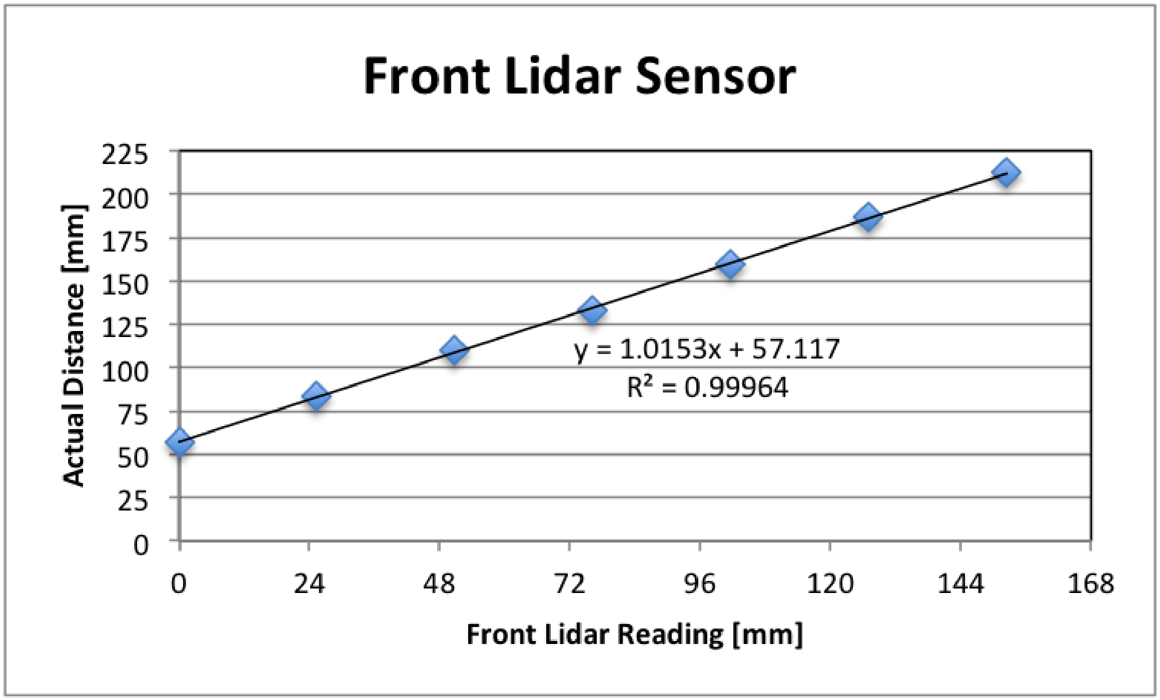

FIGURE 12. laser Range Sensor Experiment: Front

The Right laser range sensor experiment result is shown in Figure 13. As shown in Figure 13, the input distance (y) - output Lidar distance reading (x) relationship is characterized by

$$RightLidar : y = 1.0135x + 71.409$$

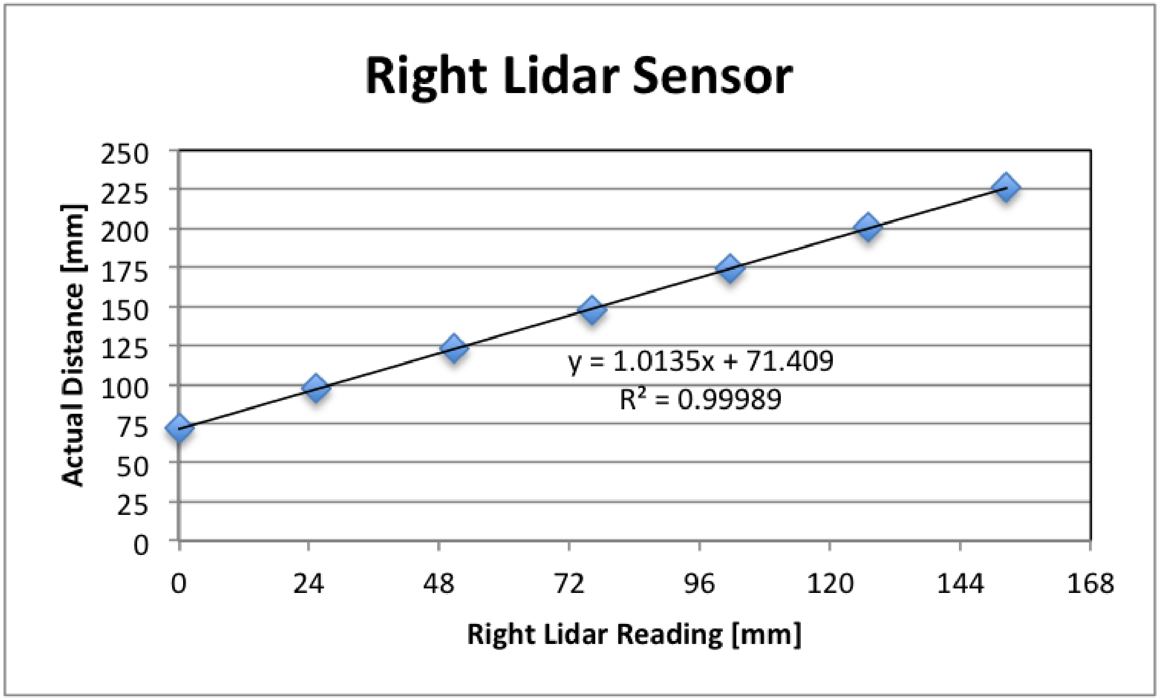

FIGURE 13. laser Range Sensor Experiment: Right

4.3.2. *MPU9250 Magnetometer.*
The declination angle obtained on UCLA campus is shown in Figure 14

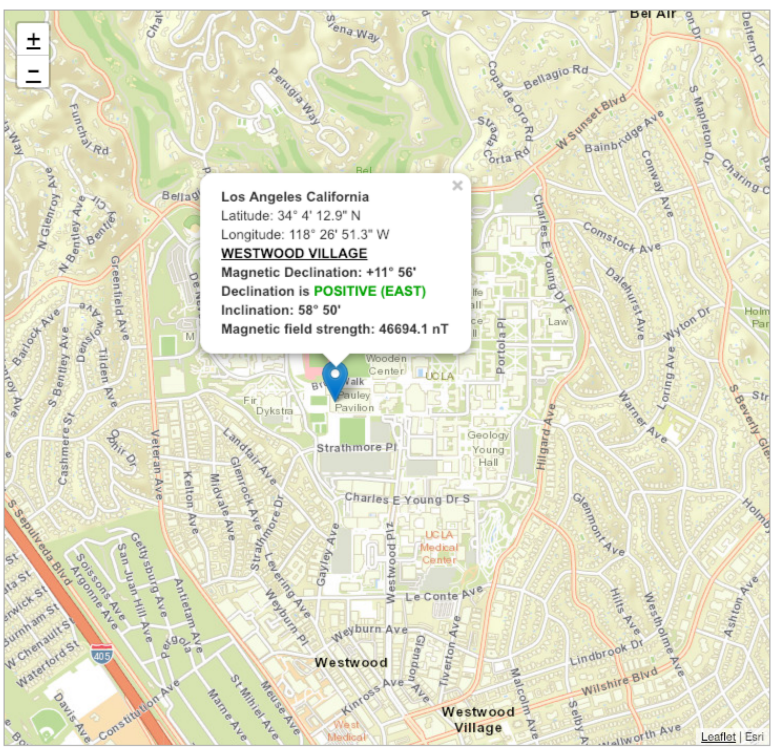

FIGURE 14. Declination Angle measured on UCLA campus

The hard iron and soft iron biased and scaling vector are found using the calibration code, the vectors are shown in Figure 15

```
Finished for loop!
Mag Calibration done!
MPU9250 Biases:
        Mx Bias: 79.000000
        My Bias: 35.000000
        Mz Bias: 2.000000
MPU9250 Scales:
        Mx Scaling Factor: 0.679012
        My Scaling Factor: 0.833333
        Mz Scaling Factor: 3.055556
```

FIGURE 15. Hard iron biased vector and soft iron scaling vecotr

After revising the magnetometer test code with the biased and scaling vectors obtained, we collect data to show that the calibration result is acceptable. As shown in Figure 16, the

BUFFALO MPU9250

Compass Value (Calibrated to Magnetic North) [Degree]	0	0 (Normalized)	30	60	90	10	150	180	210	240	270	300	330	360	360 (Normalized)
mpu9250 Magnetometer readings' mean	1.80676		30.81276	58.69012	90.72288	118.95164	151.47224	181.1892	209.96312	239.96692	271.94148	301.46128	329.18488	2.61448	
mpu9250 Magnetometer readings' variance	5.879187452		5.01068834	6.52214095	4.56856436	8.21718646	5.16533472	3.87354313	5.33391312	9.43043666	7.97855001	10.8770369	6.21220661	5.557719608	
	0.56	0.56	33.25	59.81	89.69	118.53	153.13	181.54	212.8	237.97	268.47	301.56	323.88	3.23	3.23
	2.47	2.47	33.83	59.81	95.29	122.8	154.13	181.13	209.43	242.34	272.03	298.35	330.09	2.87	2.87
	356.26	-3.74	33.83	53.88	90.22	118.53	149.81	181.54	209.43	241.91	274.99	304.63	326	359.11	-0.89
	4.99	4.99	26.95	56	95.56	119.43	153.13	178.2	211.85	240.09	269.11	304.63	328.12	0.22	0.22
	0.22	0.22	27.7	56	90.22	117.7	154.62	183.87	208.65	238.12	268.47	300.44	332.34	5.56	5.56
	2.03	2.03	29.43	57.97	92.03	117.7	155.16	180.7	211.22	236	271.54	295.74	328.29	2.87	2.87
	359.11	-0.89	32.8	57.97	92.47	118.53	151.53	177.61	210.98	236	269.11	306.02	330.4	5.81	5.81
	2.87	2.87	25.74	59.81	90.22	118.53	151.53	179.69	209.43	239.81	274.99	304.96	323.71	3.23	3.23
	2.03	2.03	31.85	63.59	90.22	115.04	151.91	179.23	206.26	236	277.99	297.39	330.4	0.7	0.7
	2.03	2.03	30.29	59.81	89.69	119.43	149.81	181.54	211.22	240.4	272.03	296.52	330.4	359.69	-0.31
	2.47	2.47	33.83	59.81	92.87	118.53	153.13	183.87	206.95	240.09	274.99	301.56	328.12	0.22	0.22
	359.69	-0.31	29.43	61.91	97.99	121.22	151.53	181.13	210.29	240.09	274.66	304.63	330.4	5.29	5.29
	5.29	5.29	33.83	58.12	92.03	115.74	153.59	181.13	207.7	238.12	271.54	306.2	330.4	2.87	2.87
	359.69	-0.31	26.95	56	94.99	114.39	151.53	180.7	206.95	238.12	272.47	301.56	330.4	2.47	2.47
	4.99	4.99	28.65	57.97	88.47	115.74	149.81	185.81	208.65	240.09	267.76	300.44	330.4	2.87	2.87
	2.47	2.47	30.29	54.03	90.99	118.53	151.91	180.7	210.18	237.97	269.11	306.2	328.12	5.56	5.56
	4.99	4.99	31.22	59.81	89.11	120.29	151.53	181.54	211.85	237.97	272.03	306.2	330.4	0.22	0.22
	5.29	5.29	32.8	56	89.11	118.53	151.53	181.54	209.43	238.12	272.03	304.96	326	2.87	2.87
	4.99	4.99	33.83	62.34	92.87	114.39	149.81	181.54	213.83	238.12	277.99	301.22	326	359.69	-0.31
	4.99	4.99	26.95	59.81	89.69	120.29	151.91	180.22	211.85	240.09	268.47	301.56	326	356.26	-3.74
	2.03	2.03	32.8	59.81	95.56	121.22	153.13	180.7	212.8	238.12	269.11	304.63	330.4	359.11	-0.89
	5.29	5.29	26.95	60.4	89.11	117.7	153.13	180.7	206.26	236	268.47	300.44	328.12	359.69	-0.31
	4.99	4.99	31.85	60.09	95.29	118.53	153.59	178.74	206.95	236	272.03	301.56	330.4	2.87	2.87
	4.66	4.66	33.83	64.13	92.87	121.22	153.59	183.57	209.43	241.91	274.99	301.56	330.4	359.69	-0.31
	2.03	2.03	27.7	58.12	87.61	117.7	149.81	183.23	210.98	242.34	269.11	301.56	330.4	5.81	5.81
	359.11	-0.89	33.83	56	89.11	115.04	151.53	181.13	211.85	240.09	267.76	304.63	330.4	6.24	6.24
	353.47	-6.53	30.29	59.81	89.11	115.74	153.13	180.22	209.43	236	272.03	298.35	330.4	0.7	0.7
	4.99	4.99	30.29	60.4	90.22	116.95	151.53	183.23	210.29	239.81	274.99	295.04	328.12	0.7	0.7
	359.11	-0.89	34.2	56	89.11	118.53	153.13	180.22	213.83	240.09	271.54	300.44	328.12	2.87	2.87
	2.03	2.03	31.22	57.97	92.87	115.74	153.13	175.74	210.29	242.34	274.66	307.57	326	5.56	5.56
	2.47	2.47	29.43	60.09	92.03	115.04	147.97	180.7	206.95	241.91	270.99	304.63	328.12	5.81	5.81
	2.03	2.03	31.22	57.97	90.22	115.04	149.58	181.13	208.65	237.97	269.11	298.35	323.88	3.23	3.23
	2.03	2.03	33.83	56	90.22	118.53	153.59	186.03	210.29	237.97	269.11	297.39	330.4	5.56	5.56
	5.56	5.56	31.85	60.4	86.26	121.22	151.53	183.87	210.29	238.12	269.11	297.39	332.34	2.87	2.87
	359.11	-0.89	26.95	62.84	92.03	121.22	153.13	181.54	208.65	236	274.99	301.56	332.84	2.87	2.87
	359.11	-0.89	29.43	57.97	90.22	123.83	149.81	181.13	208.65	237.97	272.03	304.63	332.34	4.99	4.99

FIGURE 16. Magnetometer Calibrated Test Experiment

mean of the recorded data and variance both tell that the calibration process is successful.

5. Appendix and Demonstration

All class materials and documents can be found on GitHub
`https://github.com/IouSC/UCLA_EE183DA_TeamBuffalo/`
All codes, videos, excel sheets and related documents for Lab 1 can be found on GitHub
`https://github.com/IouSC/UCLA_EE183DA_TeamBuffalo/tree/master/TeamBuffalo_EE183DA_Lab%201`

For experiment videos and demonstrations please visit links below:

- MPU9250 magnetometer calibration
  `https://github.com/IouSC/UCLA_EE183DA_TeamBuffalo/blob/master/TeamBuffalo_EE183DA_Lab%201/BUFFALO_MPU9250/BUFFALO_MPU9250_Calibration.mp4`
- Servo Forward Mode
  `https://github.com/IouSC/UCLA_EE183DA_TeamBuffalo/blob/master/TeamBuffalo_EE183DA_Lab%201/BUFFALO_Servo/BUFFALO_Servo_forward.mp4`
- Servo Backward Mode
  `https://github.com/IouSC/UCLA_EE183DA_TeamBuffalo/blob/master/TeamBuffalo_EE183DA_Lab%201/BUFFALO_Servo/BUFFALO_Servo_backward.mp4`
- Servo Clockwise Mode
  `https://github.com/IouSC/UCLA_EE183DA_TeamBuffalo/blob/master/TeamBuffalo_EE183DA_Lab%201/BUFFALO_Servo/BUFFALO_Servo_cw.mp4`
- Servo Counter-Clockwise Mode
  `https://github.com/IouSC/UCLA_EE183DA_TeamBuffalo/blob/master/TeamBuffalo_EE183DA_Lab%201/BUFFALO_Servo/BUFFALO_Servo_ccw_long.mp4`