Sensor Data Refinement

New Mexico

Supercomputing Challenge

Final Report

April 3, 2013

Team Members

Danielle Garcia

Denton Shaver

Keva Howe

Teacher(s)

Creighton Edington

Jerry Esquivel

Project Mentor

Creighton Edington

Abstract

Our project was first decided when our teacher and mentor, Creighton Edington, came to us with the idea since we involved the competition, Botball. Botball is an autonomous competition that requires student to learn how to build and fully program robots. There are multiple sensors but the one we focused on specifically was the proximity "ET" sensors. This sensor uses infrared beams to find how far away an object is, but the only problem is that the data tends to show a parabola which means it shows the same value twice.

To fix this problem, we started out by collecting the output values under different lighting conditions. After this step, we took all the values of the sensors and plugged them into different tables according to the lighting and whether we used the white of the black side of our wooden block. After graphing the tables we found that the sensors are more accurate under the bright and dim lighting.

Once this was understood, we started on the creation of the program in accordance to what the lighting was. Other than our final report, this was the longest part. The code had many mistakes, but we persevered and worked to make sure the code was perfect. After this, testing began.

The testing of the final code was long due to the amount of time it took to create an average value per centimeters away from the wooden board. Though it was time consuming, it proved to be worth it because the parabola in values disappeared and were replaced by a semi linear line. The line was not as straight as we wished it could have been, but it still worked excellent in terms of accuracy.

Table of Contents

Introduction

1.1 Problem Statement

As times goes on into the future our technology continues to improve. As problem solvers of the future generation, we took this upon ourselves. The data collection of a Rangefinder sensor can be difficult. The Rangefinder gives the distance from one point back to the sensor. Once the sensor reads a certain point, the data will repeat itself. To make the sensor more reliable for other people to use, a function was made the ideal solution for this problem. Once the function was created, a step further was taken. Using the concept of parallel processing, attaching multiple sensors to five brains reading the distance to get the most accurate reading and sending it to one brain to collect the data and displaying the collected data.

The reason for creating a function is to make the sensor more user-friendly for students and professionals. The theory behind the sensor is to make distance recognition more accurate not only for this project but for future vehicles with auto-recognition. Sensor Data Refinement is the building blocks for future technology.

1.2 Objective

Botball, an autonomous competition, is known to give out multiple sensors such as light and top hat sensors. One sensor that is used in particular by many teams, is the Proximity “ET” sensor. What the proximity sensor does, is shoot an infrared beam at an object and, by taking in how much time it takes for the beam to bounce back, it can show how far away an object is. The only problem with the sensor is that the sensor data values it records, tends to form a parabola. This means that a value that it gets when it is close to an object can be the same as when it is far away, or vice versa. This can majorly affect a robot in a way that it can be at a wrong distance, but thinks its distance is right. Our goal is to write a program that stops the ET sensor that stops it from forming a parabola and forming a linear line instead or at least make it so that a parabola rarely forms. The programming language we will be using is the KISS-C programming because not only is it efficient and user-friendly, but it is also the only coding used in Botball.

1.3 Background

At the School of Dreams Academy, we participate in many competitions outside of Supercomputing Challenge; such as Botball. Botball is a robotics competition that uses C programming to make student-built robotics to navigate a game field autonomously. “The robot's actions are based on information from the sensors, combined with the computer program written by the students in advance” (Botball). All of the members of our team have participated in Botball for three years, and has attended at least one of the many kickoffs that they offer. At each kick off, Botball gives each team a kit of parts that includes a Rangefinder or ET sensor. This Rangefinder Sensor is not as accurate as it could be.

1.3a Problem Found

Keva Howe used a Rangefinder Sensor last year on her robot. It would often miscalculate the distance it was away from a pole and would never collect the pieces of paper it was supposed to carry, but this did not happen at our school practice course. Instead, on the day of competition it would not read the distance accurately. The reason for this was the difference in lighting.

1.3b Rangefinder Sensor

Our project is uses older Kit of Parts pieces from Botball. Our main testing materials are the CBC (Chumby Bot Controller) and Rangefinder sensor. The Rangefinder Sensor has a maximum detection distance is 80cm, and a light sensitivity wavelength of 940 to 800 nm according to the *Sensor and Motor Manual* provided by KIPR. “This

Figure 1

sensor works by sending out a modulated frequency IR [inferred] beam and measures the angle the reflected IR light returns at and triangulates the distance to an object. Because of the modulated frequency, this sensor is less susceptible to error due to changing lighting conditions," as it states in the *Sensor and Motor Manual*.

2 Mathematical Models

The Rangefinder Sensor can only be modeled using a parabola. In creating the following function, the Rangefinder Sensor has to be tested in three lighting conditions: dim lighting, relative lighting, and highly- lit lighting. With each set of data, a graph was created with a separate function. The data was collected using a wooden board with two sides, a white side and a black side, a particle board with each centimeter up to 80 was marked, and a Rangefinder Sensor attached to a block of wood to keep the sensor steady.

2.1 Relative Lighting

The first test that was conducted with relative lighting; below, is the table of which the data was collected:

2.1.1 Results from White Side

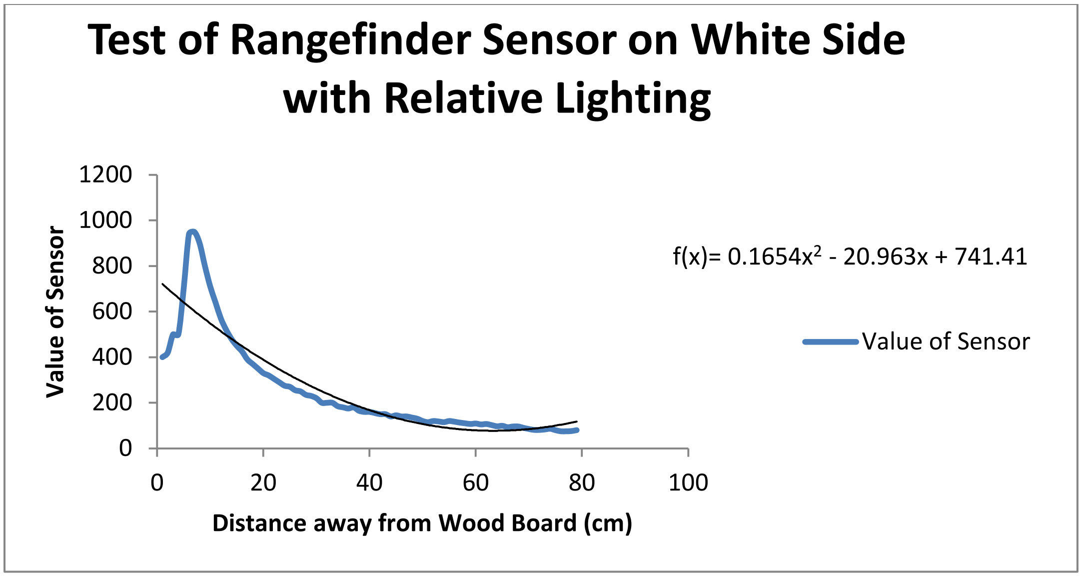
Distance away from wood board (cm)	Value of Sensor
1	400
2	420
3	500
4	500
5	700
6	940
7	950
8	900
9	800
10	710
11	640
12	570
13	520
14	480
15	450

16	425
17	390
18	370
19	350
20	330
21	320
22	305
23	290
24	275
25	270
26	255
27	250
28	235
29	230
30	220
31	200
32	200
33	200
34	185
35	180
36	175
37	180
38	165
39	160
40	160
41	155

42	150
43	150
44	140
45	145
46	140
47	140
48	135
49	130
50	120
51	115
52	120
53	118
54	115
55	120
56	117
57	113
58	110
59	107
60	109
61	105
62	107
63	102
64	97
65	99
66	93
67	96

68	96
69	90
70	85
71	81
72	81
73	83
74	88
75	80
76	75
77	75
78	76
79	80
80	80

With this table, a graph was created to plot the data and create the function needed for the overall function. The graph below was expected. KIPR, founder of Botball, did a video with plotting the values and was very close to the graph below.



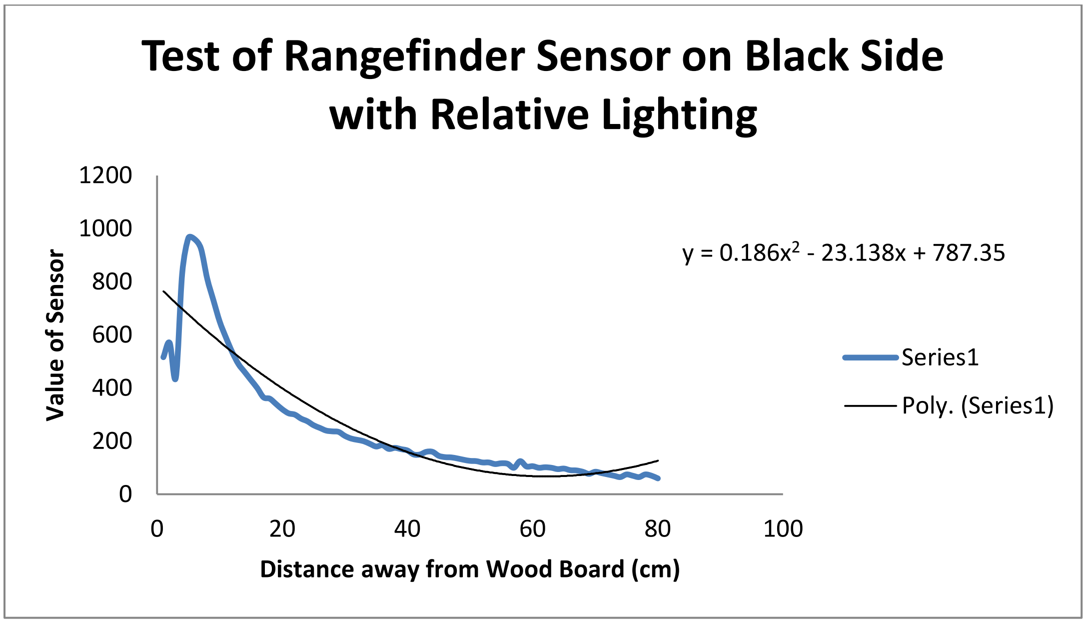
To go further and improve this data, the testing went on to on white and one black. Below is the test results and table:

2.1.2 Results from Black Side

Distance away from Wood Board (cm)	Value of Sensor
1	516
2	570
3	440
4	830
5	965
6	960
7	925
8	812
9	730
10	650
11	590
12	535
13	490
14	460
15	430
16	400
17	365
18	360
19	340
20	320
21	305
22	300

23	285
24	275
25	260
26	250
27	240
28	237
29	235
30	220
31	210
32	205
33	200
34	190
35	180
36	185
37	171
38	175
39	170
40	165
41	150
42	150
43	160
44	160
45	145
46	140
47	139
48	135
49	130
50	126
51	125
52	120
53	120
54	114
55	117
56	115
57	100
58	125
59	105
60	106
61	100
62	102
63	100
64	95
65	97
66	91
67	90
68	85

69	77
70	85
71	80
72	75
73	70
74	65
75	75
76	70
77	65
78	75
79	70
80	60



The data sets above have both proven two aspect of our project that color affects the distance the sensor sends back and the data the sensor receives can vary and is not consistent. It also showed that the black side had the opposite affect than the white side of the board had, but also finished the parabola on the white side.

2.2 Dim Lighting

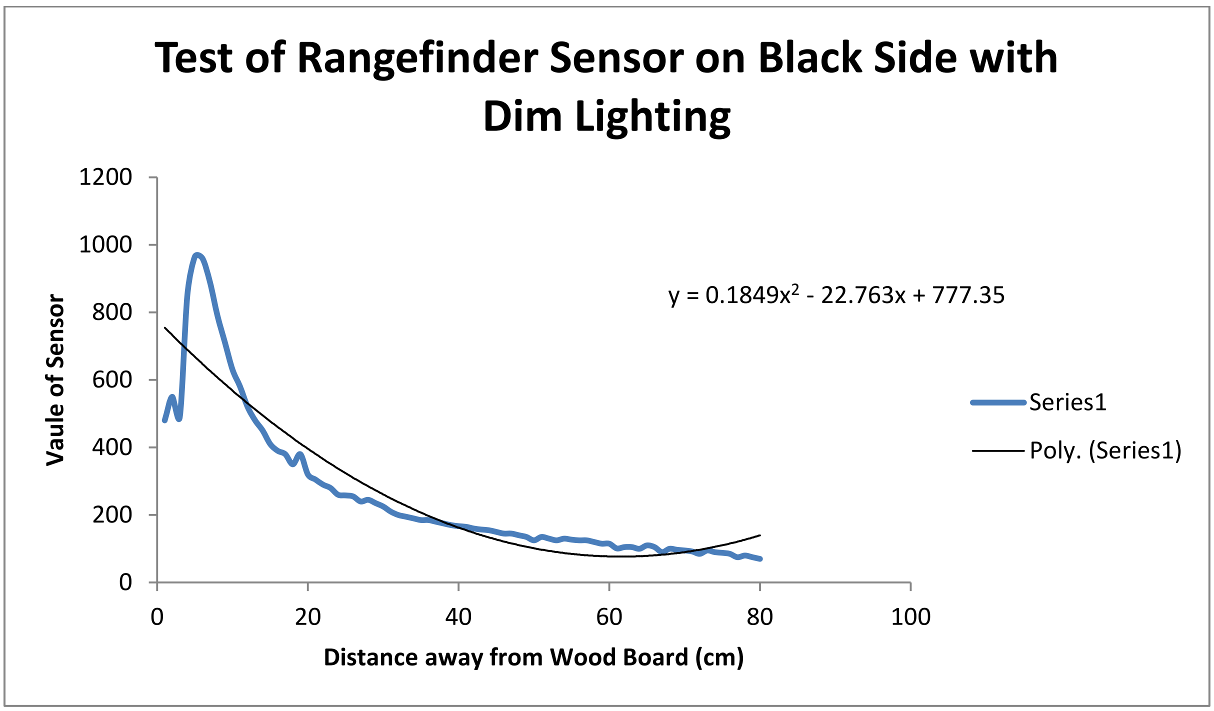
For the next test, dim lighting was counting in as a factor if students could not use the regulated lighting for their practice sections. In the data below are the results and table from both the white side and the black side.

2.2.1 Results from White Side

Distance away from wood board (cm)	Value of Sensor
1	480
2	520
3	830
4	970
5	960
6	915
7	800
8	720
9	650
10	585
11	530
12	460
13	455
14	425
15	420
16	385
17	380
18	360
19	345
20	320
21	310
22	295
23	285
24	280
25	265
26	258
27	246
28	235
29	235

30	225
31	220
32	275
33	205
34	200
35	190
36	185
37	180
38	175
39	175
40	170
41	160
42	160
43	155
44	150
45	150
46	145
47	140
48	135
49	131
50	132
51	135
52	130
53	125
54	126
55	125
56	120
57	115
58	115
59	110
60	110
61	110
62	105
63	102
64	105
65	102

66	102
67	95
68	100
69	88
70	95
71	90
72	82
73	85
74	78
75	76
76	77
77	70
78	75
79	80
80	70



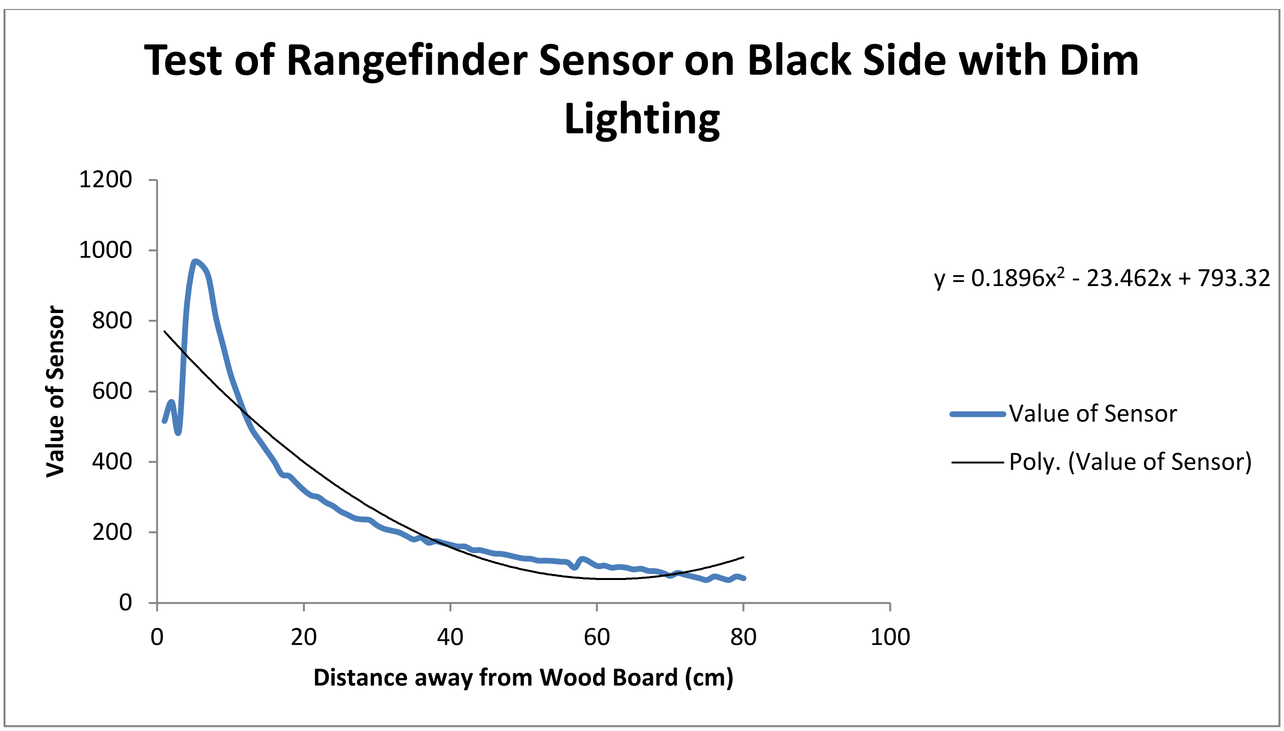
In the Dim Lighting Test, the white stayed the similar to the first graph, but the values were smaller and more evenly spaced. These sets of graphs, though, was the most useful in determining where the average values for the sensor where located with the help of the other graphs.

2.2.2 Results of Black Side

Distance away from wood board (cm)	Value of Sensor
1	516
2	570
3	490
4	830
5	965
6	960
7	925
8	812
9	730
10	650
11	590
12	535
13	490
14	460
15	430
16	400
17	365
18	360
19	340
20	320
21	305
22	300
23	285
24	275
25	260
26	250
27	240
28	237
29	235
30	220
31	210
32	205

33	200
34	190
35	180
36	185
37	171
38	175
39	170
40	165
41	160
42	160
43	150
44	150
45	145
46	140
47	139
48	135
49	130
50	126
51	125
52	120
53	120
54	119
55	117
56	115
57	100
58	125
60	105
61	106
62	100
63	102
64	100
65	95
66	97

67	91
68	90
69	85
70	77
71	85
72	80
73	75
74	70
75	65
76	75
77	70
78	65
79	75
80	70



From this set of data, it was concluded that dim lighting makes the Rangefinder sensor more concise. Unfortunately, the game field used at Botball Competitions uses highly lit arenas. This information did give us the individual functions we needed to make the main function.

2.3 Highly-Lit Lighting

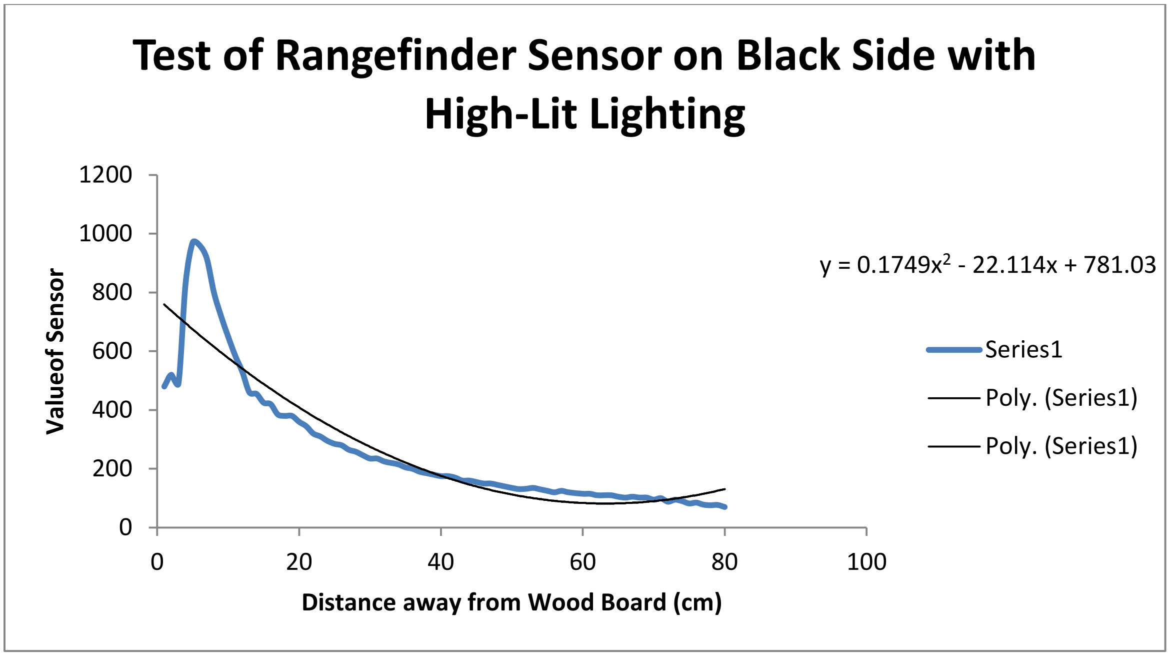
The last test that was conducted was under a highly- lit area to compensate for the Botball game fields and the dim lighting test. The table and results of this test are what follows.

2.3.1 Black Side Results from Highly-Lit Lighting

Distance away from wood board (cm)	Value of Sensor
1	480
2	520
3	490
4	830
5	970
6	960
7	915
8	800
9	720
10	650
11	585
12	530
13	460
14	455
15	425
16	420
17	385
18	380
19	380
20	360
21	345
22	320
23	310
24	295
25	285
26	280

27	265
28	258
29	246
30	235
31	235
32	225
33	220
34	215
35	205
36	200
37	190
38	185
39	180
40	175
41	175
42	170
43	160
44	160
45	155
46	150
47	150
48	145
49	140
50	135
51	131
52	132
53	135
54	130
55	125
56	120
57	125

58	120
60	115
61	115
62	110
63	110
64	110
65	105
66	102
67	105
68	102
69	102
70	95
71	100
72	88
73	95
74	90
75	82
76	85
77	78
78	76
79	77
80	70

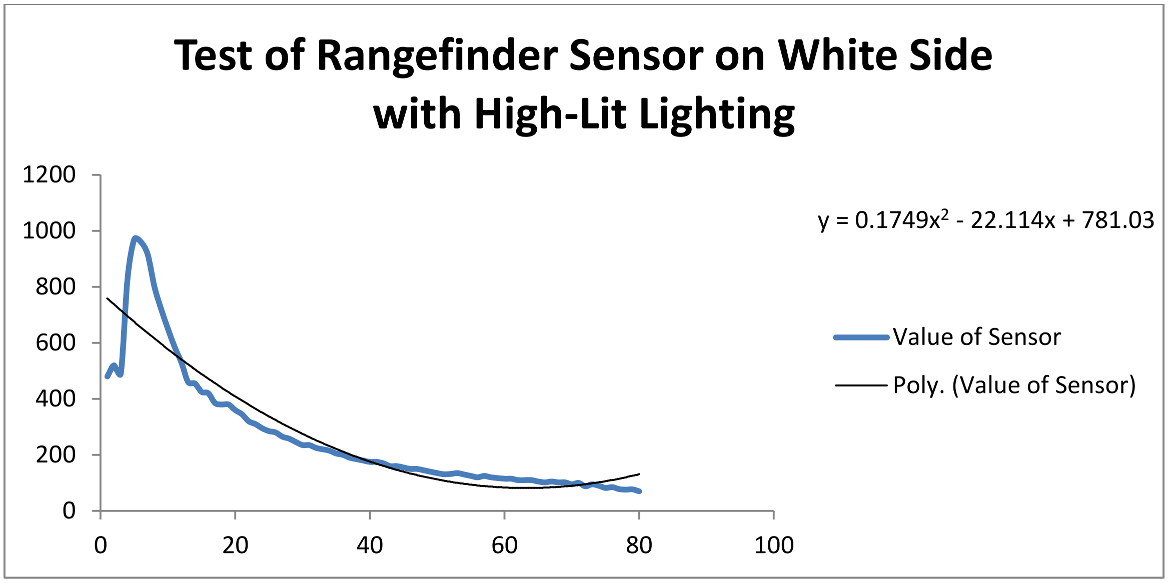


2.3.2 White Side Results from Highly-Lit Lighting

Distance away from wood board (cm)	Value of Sensor
1	480
2	520
3	490
4	830
5	970
6	960
7	915
8	800
9	720
10	650
11	585
12	530
13	460
14	455
15	425
16	420
17	385
18	380
19	380
20	360
21	345

22	320
23	310
24	295
25	285
26	280
27	265
28	258
29	246
30	235
31	235
32	225
33	220
34	215
35	205
36	200
37	190
38	185
39	180
40	175
41	175
42	170
43	160
44	160
45	155
46	150
47	150
48	145
49	140
50	135
51	131
52	132
53	135

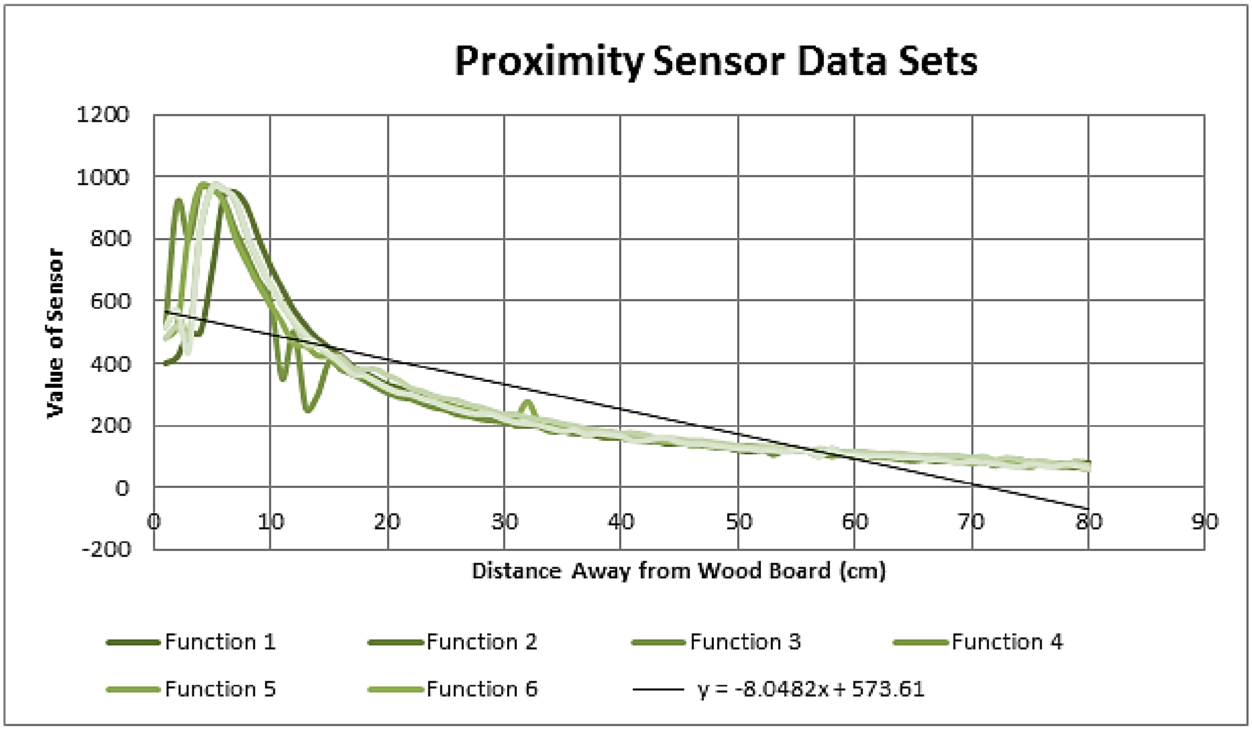
54	130
55	125
56	120
57	125
58	120
60	115
61	115
62	110
63	110
64	110
65	105
66	102
67	105
68	102
69	102
70	95
71	100
72	88
73	95
74	90
75	82
76	85
77	78
78	76
79	77
80	70



The test results shown for the Highly-Lit Lighting were very similar to the dim lighting results. This gave us the information that was proving how inconsistent the proximity sensor was to light. The next step taken was to combine the graphs to form one function.

2.4 Combined Function

The reason for combining Functions 1-6 is to give a visual of our raw data and to put our problem into perspective. This caused the graph to go from a parabola to a linear function.



Function 1. White; Relative, Function 2. Black; Relative, Function 3. White; Dim, Function 4. Black; Dim, Function 5. Black; High Lit, Function 6. White; High Lit

$$y = -8.482x + 573.61$$

The equation on the "Proximity Sensor Data Sets" graph is the ideal model for the sensor after we create the program for the proximity sensor.

3 Computational Model

Originally, if a proximity sensor were to be attached to the robot without any change, the result would be it would not know the difference between how close and how far an object is away. This program is designed in such a way that once the sensor reaches the peak in the data output parabola, it will reset itself so it can continue getting different values. To do this we would combine the data functions that we have from all the tests so it is extremely accurate. With this function, the robot will be able to measure distance far more accurately because the data we get from the Proximity sensor will form a linear line rather than a parabola.

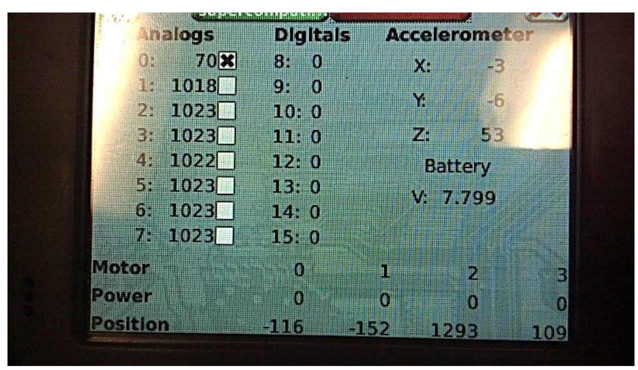
In the running of our code there is only one way to test if our code works; that way is by testing the sensor with the on screen sensor out puts, as shown in Figure 2 and 3.

Notice that in Figure 2 the sensor has not begun to read the sensor, but in Figure 3 the sensor begins to have outputs.

This is how we began the test for raw data, for our final testing we used code to see the amount of time the reading and sending back the code takes as well as the value of the sensor. In section 4, our models will go into more detail.

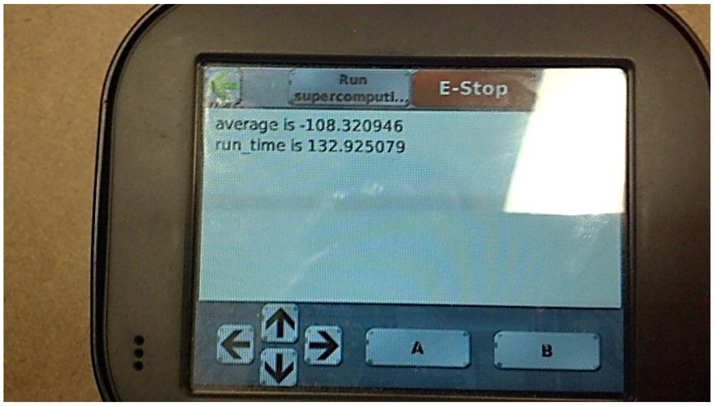
Figure 2

Figure 3



4 Code

Our code below is made to determine the amount of time the data comes back and the value. Although the value on in the code prints to the screen as negative, as shown in Figure 5, the value is positive for our results.



The code we developed used KISS-C. This programming language is in C, but can be used to its most simplies functions or can be used to the programmers more advanced functions, such as parabolas and other functions.

4.1 Developed Code

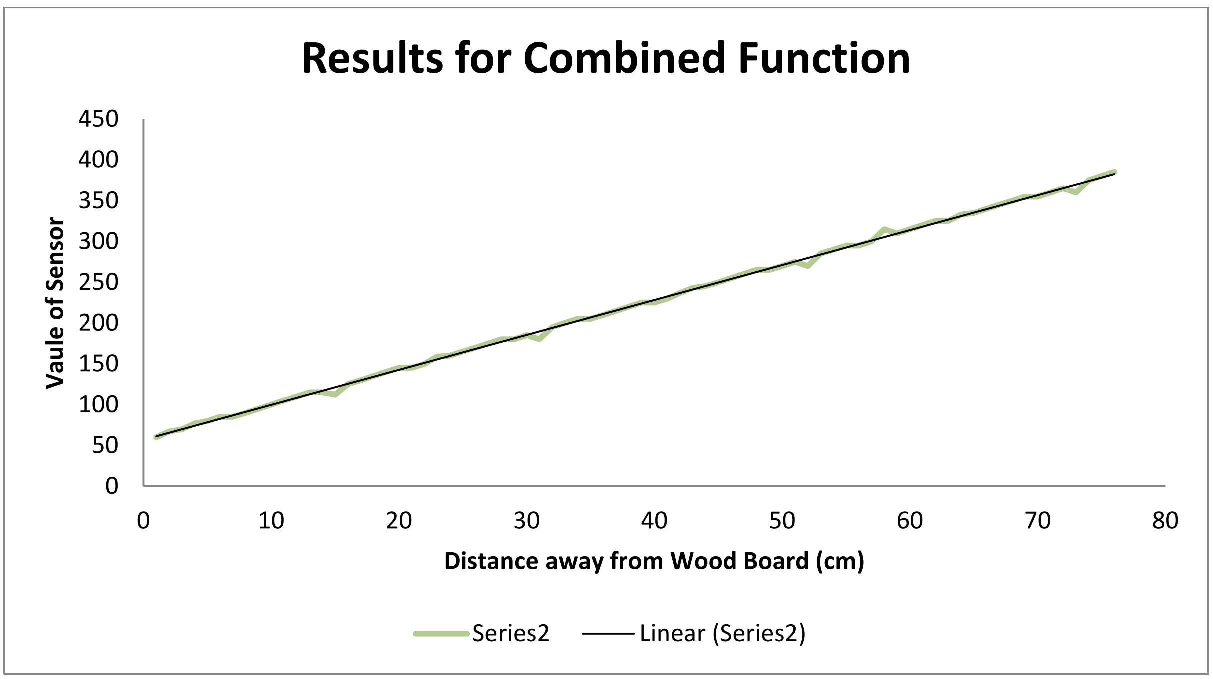
Below, is the code that we used for our project. Although, it does not have much length, it provided us the information that was needed to complete our project for this year.

```
int main()
{
    int c = 0;
    float running_total = 0.0;
    float average_value = 0.0;
    float start_time;
    float end_time;
    float run_time;
    float function_value;



    set_each_analog_state(1,0,0,0,0,0,0,0);

    start_time = seconds();
    while( c <= 20000)
    {


        function_value = -8.0482 * analog10(0) + 573.61;

       running_total = running_total + function_value;
       c = c + 1;
    }
    end_time = seconds();
    run_time = end_time - start_time;
    average_value = running_total / c;
    printf(" average is %f\n",average_value);
    printf(" run_time is %f\n",run_time);

}


```

5 Results

The results of our project are only the beginning of a large project. The results shown are only the data outputs we have so far which show all of the parabolas combined together. It is shown in our report in the mathematical model under section 2.4. The results we received from the new code create a semi linear line that waves because the program resets itself at the peak of its parabola. The code is explained in section 4.

The function that that was received at the end of our testing was in a different direction than originally planned, but worked better. The graph is the results that we got with our new function:



The code resets itself every time it reaches the peck of the parabola.

6 Conclusions

In conclusion our project has shown us that our problem was not only in our Proximity sensor, but in our lighting as well. Turns out, with either dim or very bright lighting, the sensor is actually accurate longer than when it has relative lighting. This contributes in a way that we can find the proper lighting needed for the testing. After we found this, we made our code so that is designed for that specific lighting. The affected the outcomes in a way that the data outputs we needed were very accurate.

The coding part of our project came out almost exactly like we needed it to, other than the fact that the linear line was not as perfect as we would have liked to be. This was due to the fact that it reset itself at the peak of the parabola which caused the line to become wavy rather than straight. Nevertheless, the program worked incredibly well and will most definitely help our team and others in this very common bug.

7 Recommendations

In many of today's cars, there are sensors placed in the back of the car that prevent crashes from behind. We figured that the more sensors there are, the more safe the car would be. So, what our future plan would involve sensors placed around the car so that it does not matter where a danger is located around a car, the car will forewarn all passengers of the car of the oncoming danger. This scan keeps passenger and other drivers on the road safe.

The accuracy of this code is vital for the passenger's safety. If the warning goes off to early due to the data intake forming a parabola, it can be dangerous for the driver in a way that they can be surprised and cause a wreck. If the coding is consistent, then it makes that danger go away.

8 Acknowledgements

We would like to thank our teacher/mentor, Creighton Edington and Jerry Esquvel, for supporting us and helping us with our project.

The non-profit organization, KIPR, helped us with our materials needed for research and all their help.

We would like to thank the School of Dreams Academy for being our main supporter in our project.

Thank you to our parents, for without them we would not be able to do our project, and for helping us push forward.

Lastly, we would like to thank The Supercomputing Challenge for the opportunity that they gave us.

Appendix

I. Resources

"CBC V2 Manual." *CBC V2 Manual*. KISS Institute for Practical Robotics. Web. 30 Mar. 2013. <http://www.kipr.org/sites/default/files/CBC_V2_Getting_Started_Manual_BB2011.1.pdf>.

"Sensor and Motor Manual." *Sensor and Motor Manual*. KISS Institute for Practical Robotics. Web. 1 Mar. 2013. <http://www.cs.uml.edu/~holly/teaching/91450/fall2011/Sensor_and_Motor_Manual_BB2011.pdf>.

II. Software/ Tools

Software

KISS-C

Tools

Proximity Sensor

CBC

Microsoft Office

Google Docs

Particle Board

Wood Board with black and white sides

III. Glossary

CBC: CBC stands for Chumby Bot Controller. This “brain” was created by KIPR

KIPR: KIPR stands for the KISS Institute of Practical Robotics, a non-profit organization, and creator of Botball.

KISS-C: KIPR's programming software used in Botball and other competitions.

Rangefinder/ET/Proximity Sensor: This sensor goes by many names, but they are all the same. This sensor provided distance reading using inferred.