EE3: Intro to Electrical Engineering
Path Following Robot

Brian Dionigi Raymond
Kevin Ke-En Sun

Instructor: Prof. Briggs

December 3, 2017

Introduction

Introduction plaintext - To be added

Testing Methology

How We Designed the Test

Testing Methodology Design plaintext - To be added

How We Conducted the Test

Testing Methodology Conduction plaintext - To be added

How We Analyzed the Test Data

Testing Methodology Analysis plaintext - To be added

How We Interpreted the Data

Testing Methodology Interpretation plaintext - To be added

Results and Discussion

Test Discussion

Test Discussion plaintext - To be added

Race Day Results

Results Discussion Conduction plaintext - To be added

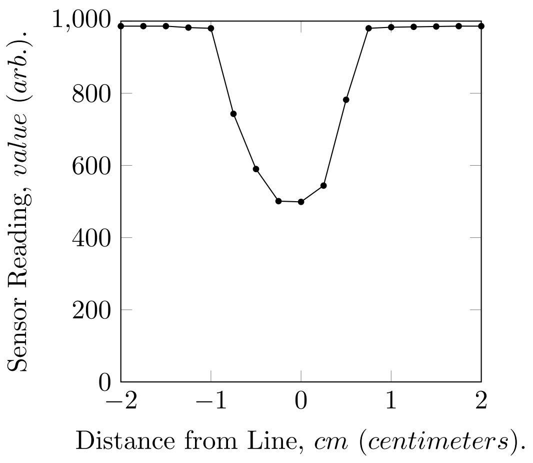
Conclusions and Future Work

Conclusions plaintext - To be added

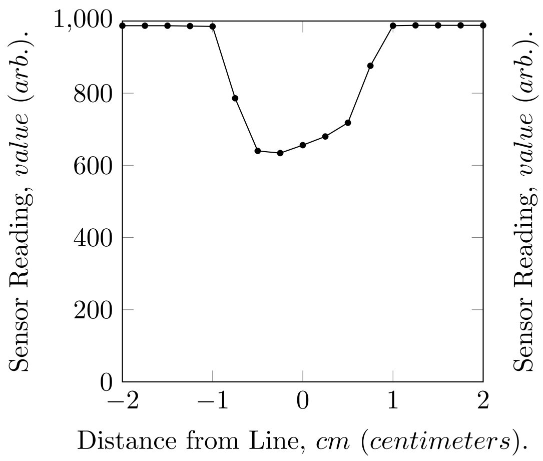
Future Work plaintext - To be added

References

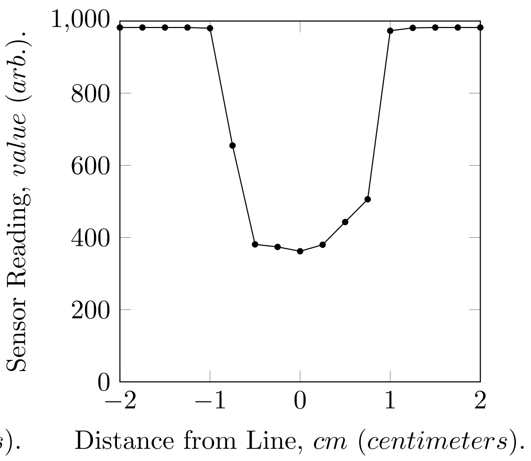
References plaintext - To be added



(a) Front Sensor

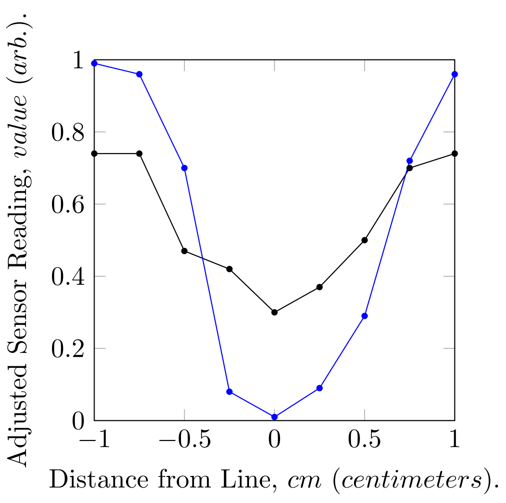


(b) Left Sensor

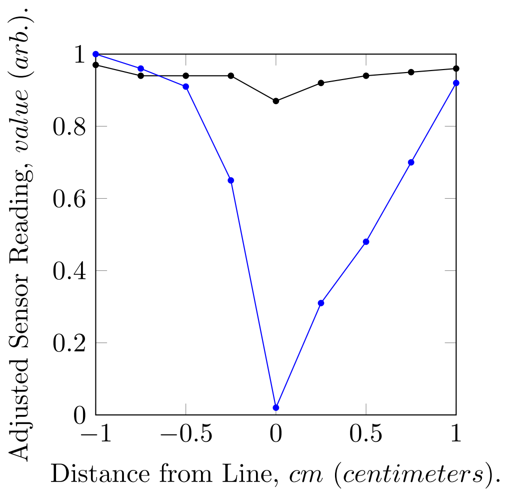


(c) Right Sensor

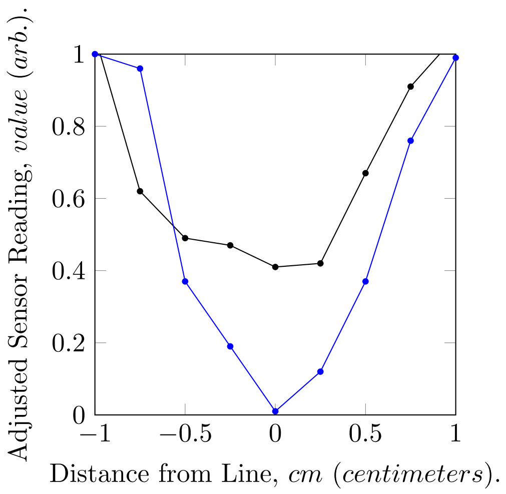
Figure 1: **Raw readings from the IR sensor, IR phototransistor combinations used on our path-following robot.** From the graphs, the value of the maximum readings range from about 400 for Graph 2c to about 600 for Graph **??**. Also worth noting is that the sensors each reach a global minimum around $0cm$ and slope downwards when moving from $\pm 1cm$ from the black line. This is optimal since the symmetry of the readings allow us to use an even polynomial function to obtain the distance from the sensor readings.



(a) Front Sensor



(b) Left Sensor



(c) Right Sensor

Figure 2: **Adjusted sensor readings using a crude polynomial fit of the data from Graphs 1a, 1b, and 1c with a degree $n = 2$ and then mapping to expand the range across the sensors line of sight (0 to $1cm$).** By using a polynomial fit of the sensor reading (y) versus the distance from the line (x) and solving in terms y, we are able to both find the distance from our readings and preserve each sensors local minimum at $0cm$. From there, we need to adjust the values to fit our known distribution and therefore simply map it as shown. By doing this, we see that our data better fits the distribution and is able to give us more valuable information to use with our PID function as the same change in the sensor reading returns a more significant change (and accurate) change in calculated distance from the line.

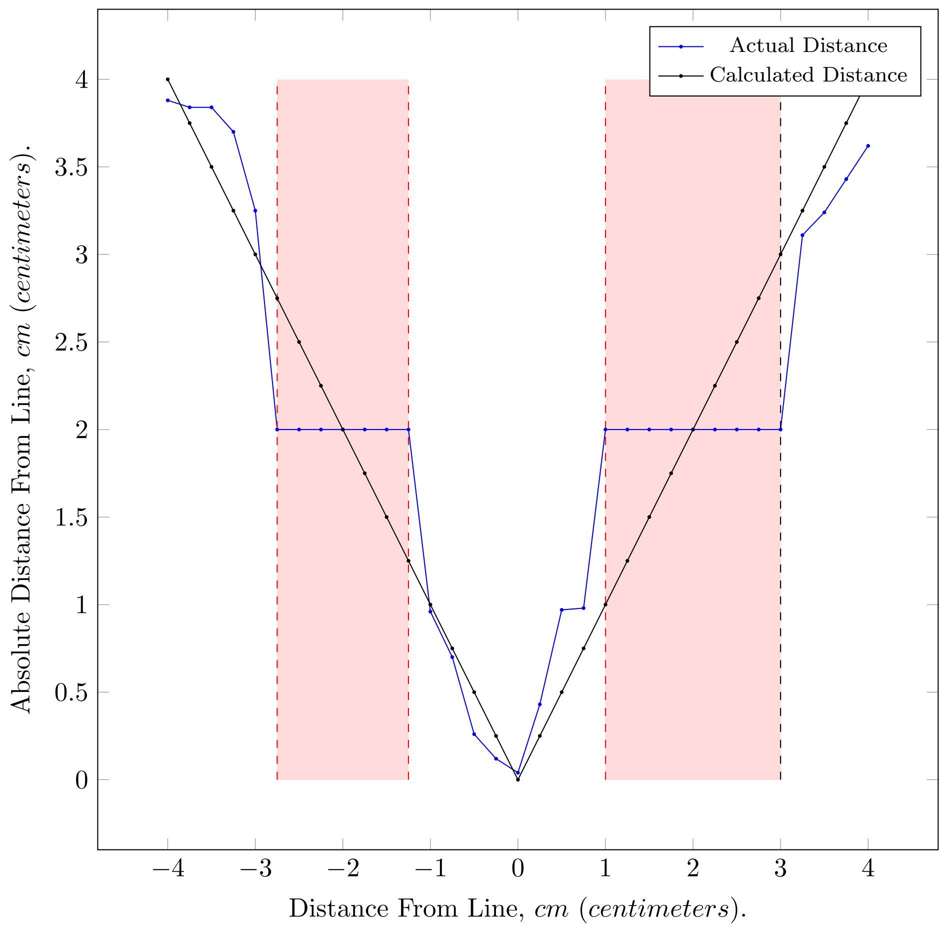


Figure 3: Comparison between actual and measured distance (from our getLocation() function) between center of $2cm$ thin black line and center of robot's front sensor. The red zone in the figure represent the area for which the black line is not within the range of the left, front, or right sensors. The calculated distance for these areas is just the average value of the actual distance. Looking at this graph, the calculated distance is generally accurate to within less than half a centimeter. Most importantly, the graph slopes to a minimum from both sides, an important feature if a function is to be used for a PID controller.