IEOR 140 Project 2 Milestone 2 - Due 9/7/2012

Team 4: Nate Bailey, Raymond Ma, Edison Park

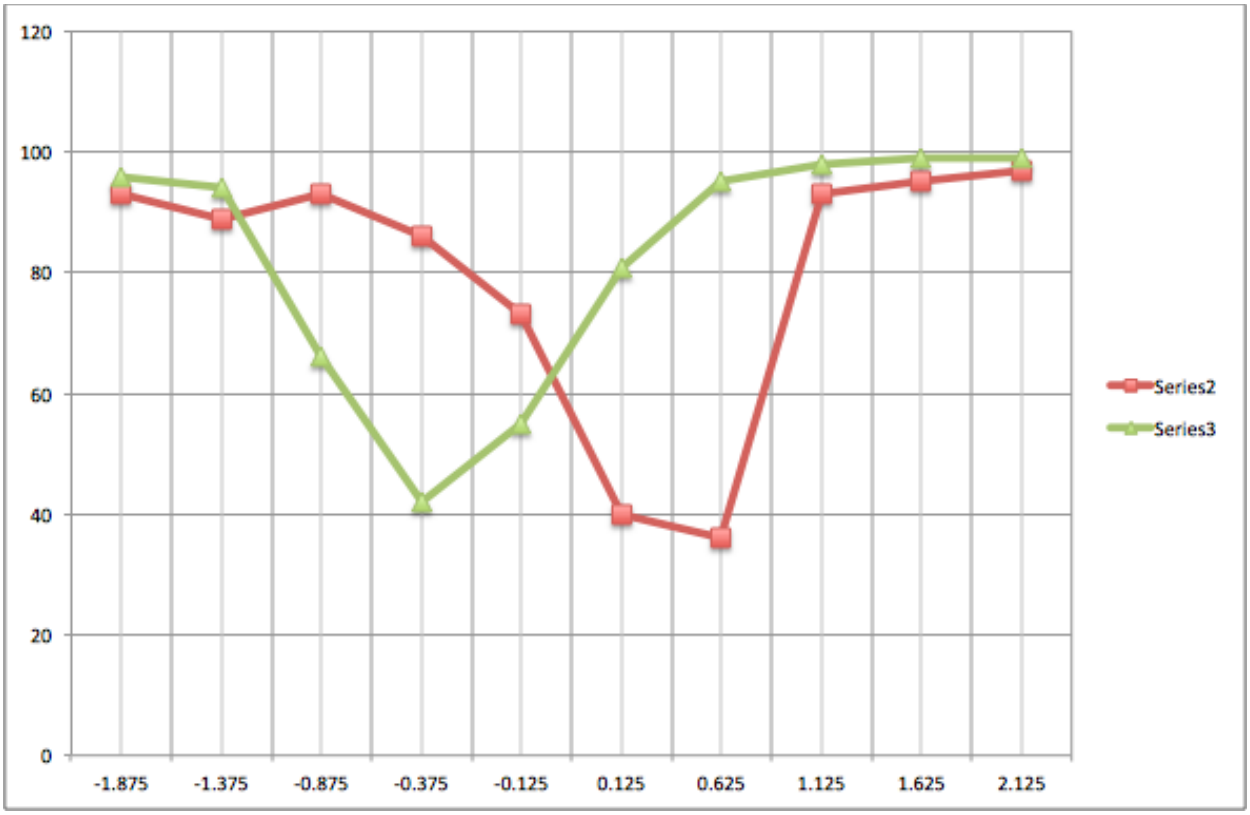
Project Delegation

On this project, Nate was once again the chief program coder. He implemented all of the methods required for the project and bug tested the individual parts of the code to make sure that the robot worked properly. Raymond was again the chief mechanical designer and worked on building the robot as well as making sure the sensors were attached properly. Raymond also worked with Nate to plan out the methods required to implement the new methods required.

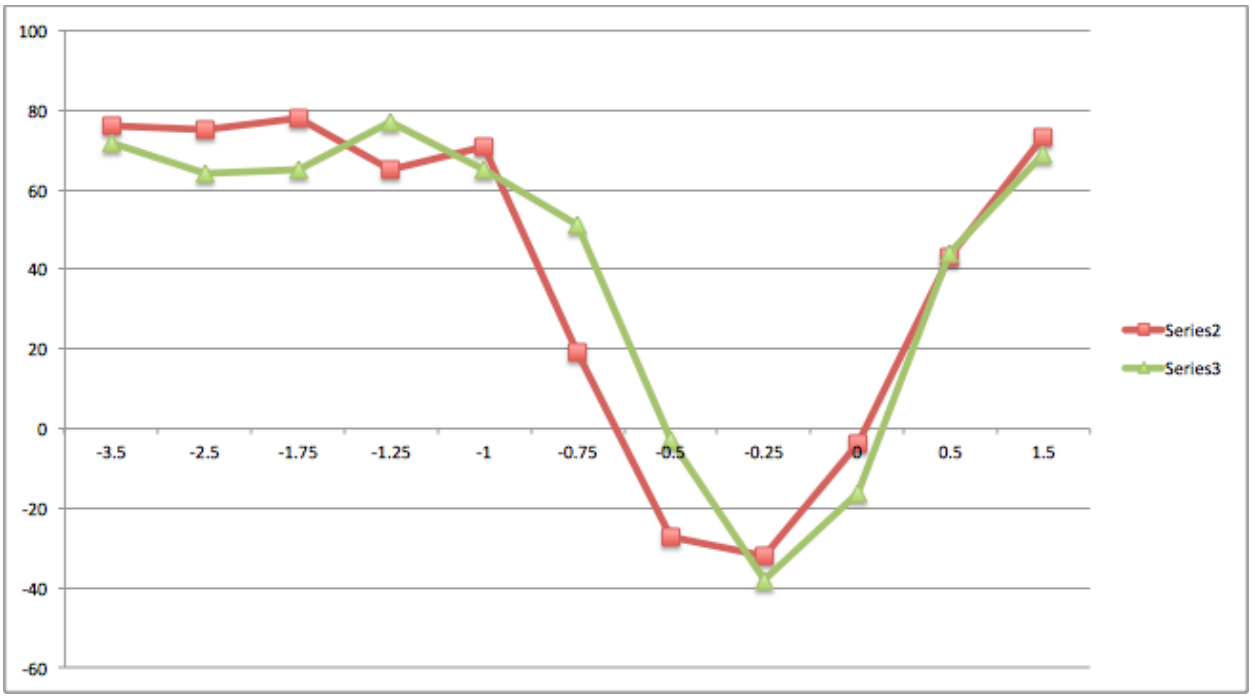
Time Spent on Project

3 hours (each day) x 1 (day on Friday) x 2 (people) = 6 man hours

Experimental Work

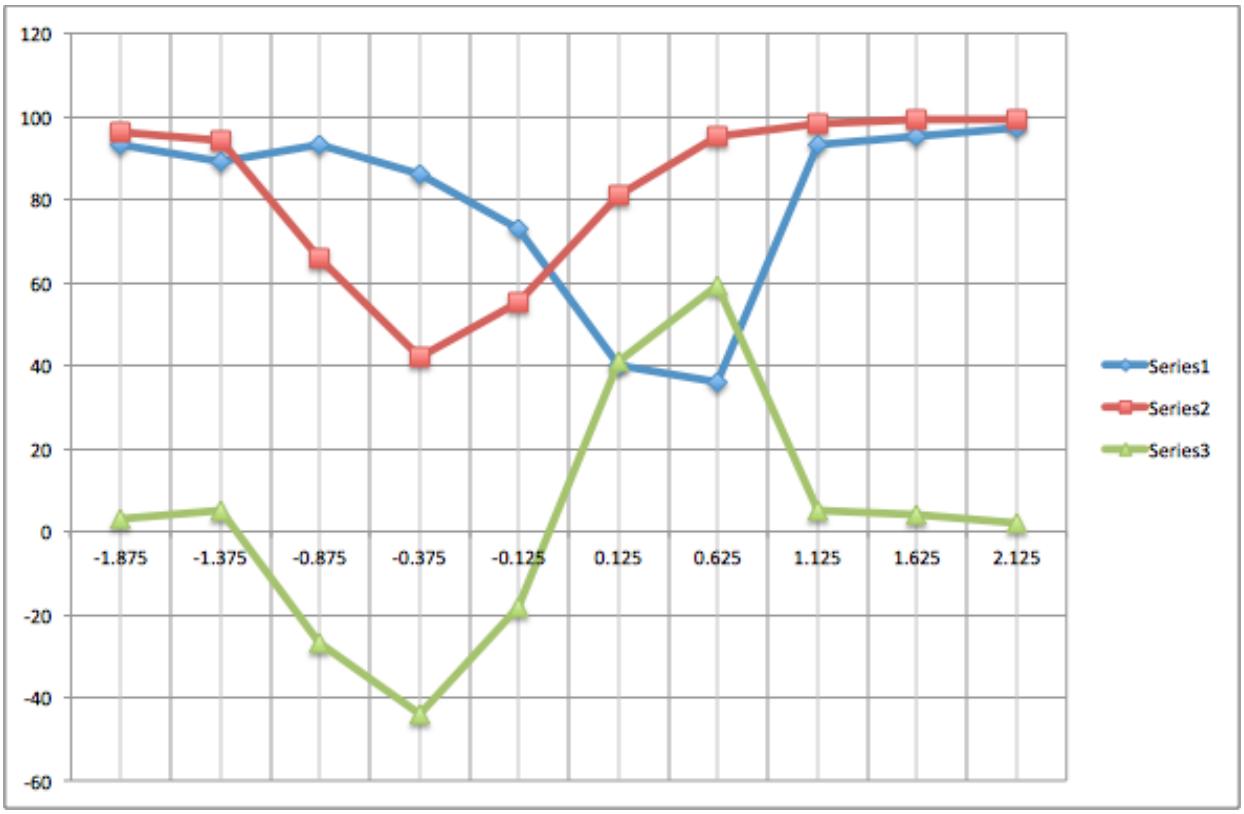


In order to see how the light sensors behaved at different distances from the center line, we used some of the code supplied to us in order to perform tests of the light sensors. The above graph shows light sensor readings as a function of the distance that the center of the light sensors was from the center of the line. The x-axis shows this distance while the y-axis shows readings from the leftmost light sensor (red) and from the rightmost light sensor (green).



In addition to that data, we also tried to find out how our light sensors were able to find the black square which marked the halfway points around the track. In the above graph, the x-axis represents the distance that the light sensors were from the center of the black square, while the y-axis represents light sensor readings from the left and right light sensors (same colors as above).

The first set of data actually did not turn out to be that useful for us, as we ended up doing more guesswork to find a good gain constant as opposed to using some kind of a function that translated the difference between the two light readings into an estimate of the distance from the center line. However, the second set of data proved very useful as we were able to use it to obtain accurate thresholds that told us when our robot was on the black square and when it had gone past it.



This is the graph that shows how our robot steers towards the center line. The value for green is calculated by taking the left eye measurement and subtracting it from the right eye. The closer the points of the left and right eye measurements are, the closer to zero the green line is. Our robot multiplies the green line with a gain constant of 1 and then passes this information into the differential pilot steer method.

Task Analysis

The main tasks that our code needed to perform in order to succeed on this project were:
1) Obtain measurements of the light readings from our sensors
2) Use those measurements to tell us something about where our robot was compared to the center line - in other words, measure our error.
3) Apply control theory and feed the error into a scheme that led our robot back to the center line.
4) Find the black square that marks the halfway points of the oval and stop tracking
5) Implement this in a loop that allowed our robot to perform four loops, turn around, four more loops, etc.

We accomplished these tasks by using a simple control scheme where we multiply the difference between the two light sensors by a gain constant. This means that a large difference between the two light sensors means a sharp turn, which usually happens when one eye is looking at the center line and the other is looking at the white of the mat. This can sometimes yield a jerky motion as it ventures slightly of the line and turns sharply back onto the line, although reducing the gain constant helped with that

problem.

The other thing we did was to implement a loop in our main function that called the trackLine() method, which exited when it saw a black square, then moved a little bit forward and called the trackLine() method again, and so on.

Observations on the Project

For this project, the most interesting part was being able to use real world information obtained through sensors to control robot movement. By using sensors, our robot was able to respond to real world objects such as a line and perform actions such as following a line as well as correcting motion to stay on the path of the line. For us, the most challenging/difficult part of this project was figuring out how to convert the deviation in light readings into useful commands for the robot. From the data collection above, we took various samples of light readings to understand how each light sensor responded to being under different color situations and then formulated a range for which an offset off the line would be responded to with a correction of motion for the robot.

Appendix

Source code listing on Github