IEOR 140 Final Project Milestone 2 - 10/25/2012
Team 4: Nate Bailey and Raymond Ma

Responsibilities

In this project, Nate was in charge of program design and coding. Raymond was in charge of hardware design, experimental work, and project writing.

Hours Spent

Approximately 3 hours of work

Project Code

https://github.com/ieor140-team4/FinalProject

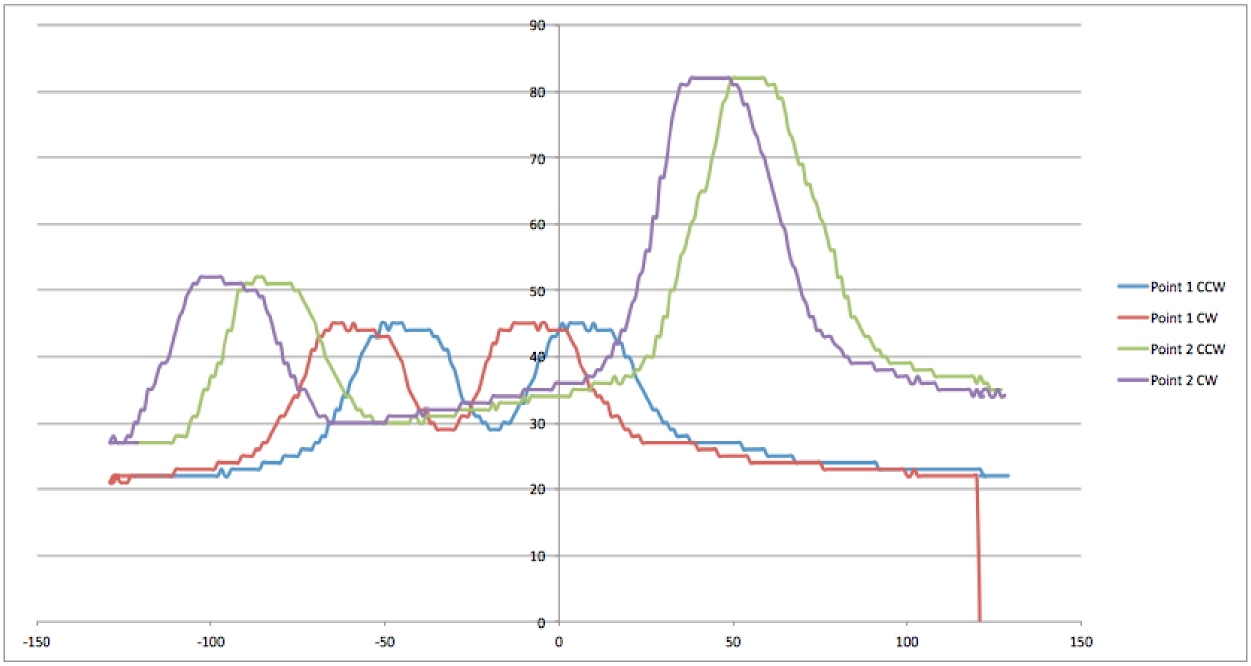
Performance Specifications

Our robot met all of the performance specifications (there were no bonus specifications to meet).

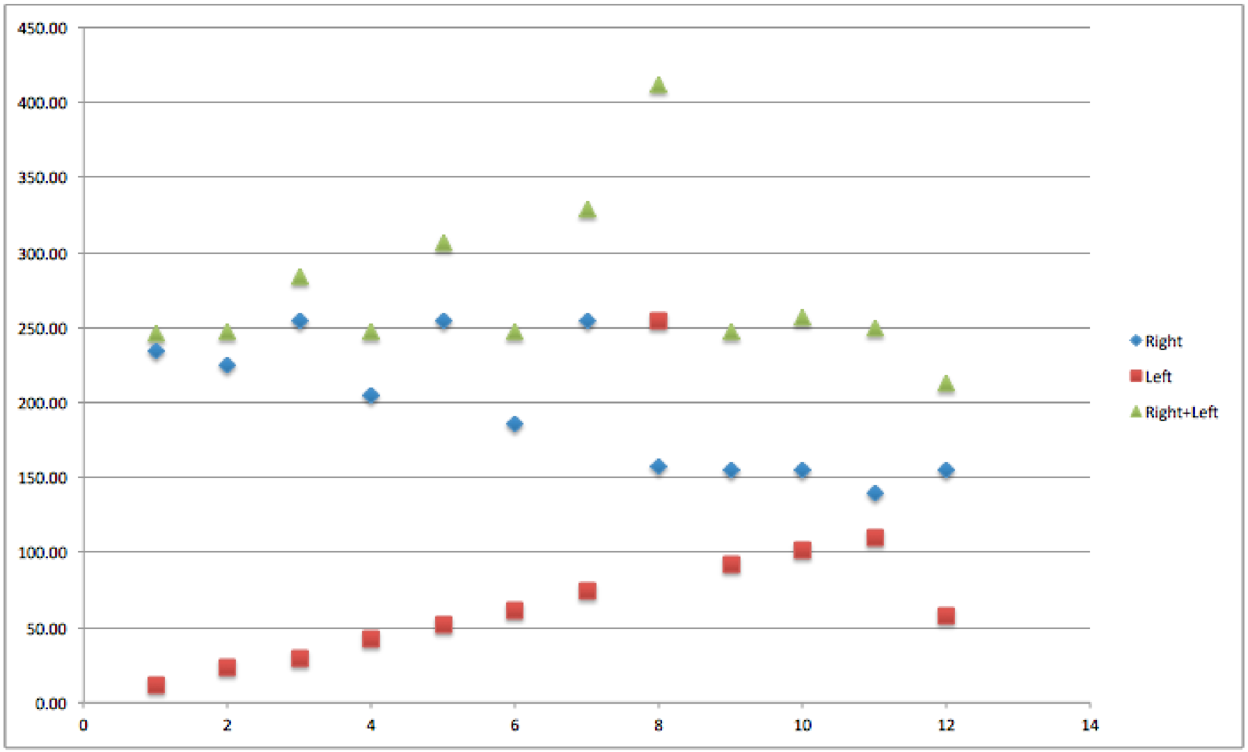
Hardware Design

Our hardware design consisted of the default GridNavigation robot. We had two touch sensors pointed at the ground in order to track both sides of a line and an ultrasonic sensor pointing forward in order to detect blocked nodes. We didn't use touch sensors for this robot. The robot was a two wheel with a castor wheel design with the center axle directly below the middle of our robot.

Experimental Work



Light sensor data with Point 1 being the far away point (240, 30) and Point 2 being the close point at (20, 20).



Ultrasonic data gathered starting 10cm from the way and moving over into the middle of the hallway in 10 cm intervals. The Green dots are the added left and right points, giving us the total width of the hallway. Most readings give us an average of 248 (which is the the actual width), but we have some major outliers, which brings questions of consistency.

Task Analysis

- Use the motor to rotate the sensor so that it can take various light readings at different angles.
- Take light readings at each angle
- Use logic to determine what light readings actually correspond to beacons
  - Ignore all values underneath a threshold light value (42 in our case, selected from our experimental work.)
  - Once the light value passes underneath the threshold after already seeing a peak and storing a highest light value, we know that we've passed the first peak and that the next highest light value we see will be the second peak.

Class Responsibilities

This milestone was performed pretty much entirely by the Scanner class in NXT Files. The scanner was in charge of rotating the motor associated with the light sensor, taking in readings from the light sensor at each angle, and using logic to determine how the light readings we observed translated into beacon headings.

Results from Final Scanner Test

Point 1:	(240, 30)		Point 2:	(20, 20)
R Bearing	L Bearing		R Bearing	Left Bearing
6	-41		-81	58
5	-43		-79	57
5	-42		-80	58
5	-43		-79	57
5	-42		-78	57
4	-42		-80	57
5	-41		-80	57
5	-45		-79	57
RB Mean:	5		RB Mean:	-79.5
RB StdDev:	0.534522484		RB StdDev:	0.9258201
LB Mean:	-42.375		LB Mean:	57.25
LB StdDev:	1.302470181		LB StdDev:	0.46291005

These results are pretty good. ¾ of the standard deviations are less than 1, and the only one that is over 1 is just barely over with a value of 1.3. It seems that the variance tends to be highest either when very far, as with the far away point looking towards the opposite beacon, or when very close up, as with the closest point looking to the close beacon.

Interesting/Challenging/Difficult

The most interesting, challenging, and difficult part was all in the logic of figuring out when we had passed one peak and to begin looking for the second peak. We solved this problem by experimentally finding a threshold and knowing when a peak was passed when the values dropped below said threshold.

Appendix

Source Code | Java Docs