To: Dr. Berry
From: Peter Heath, Matthew Schack, and Data
Date: 12/15/15
RE: Lab 03 making the robot move with sensor input

The purpose of this lab was to make many behaviors that were dependent on sensor input. The behaviors were divided into levels starting with the lowest level (level 0) with the two behaviors of: shy kid and aggressive kid. The next level of behaviors (level 1) were: random wander and random wander with obstacle avoidance.

The first step of making any of the code work was calibrating our sensors. We tested and obtained calibration curves for both the Sonar and IR sensors. The calibration curves are located in Appendix A. The next step was to put our sensors on Data. We chose to put the sensors all up front to have redundant sensing and because we only planned on moving forwards. This gave us better results for everything we would be facing, and since we were driving forward most of the time having the sensors up front was very helpful.

Obstacle avoidance behavior is handled two ways. The first way we poll each sensor and add up the ones lower than a threshold as vectors. We then use this vector to control Data's movement. We used this method for shy kid and as the obstacle avoidance part of random wander with obstacle avoidance. The other method we used was a true-false statement linked to the front sensor. If the front sensor was returning a number lower than a threshold then we would stop. Otherwise, we would go straight. This method was useful for the aggressive kid behavior.

The higher level code was a lot easier to implement. For random wander we would pick a random angle and turn to it. We would then go forward at a random speed for a set amount of time, and then repeat the entire process. To make random wander with obstacle avoidance we just combined both random wander and shy kid together. We were able to combine them like this because our programs were modular. Every function did a specific thing and we linked them together. For random wander with obstacle avoidance Data would only poll some sensors, but if those sensors said something was close we would move into shy kid behavior otherwise we would move into random wander behavior. To make debugging easier we would have the screen print what behavior it was doing at the current time to the LCD.

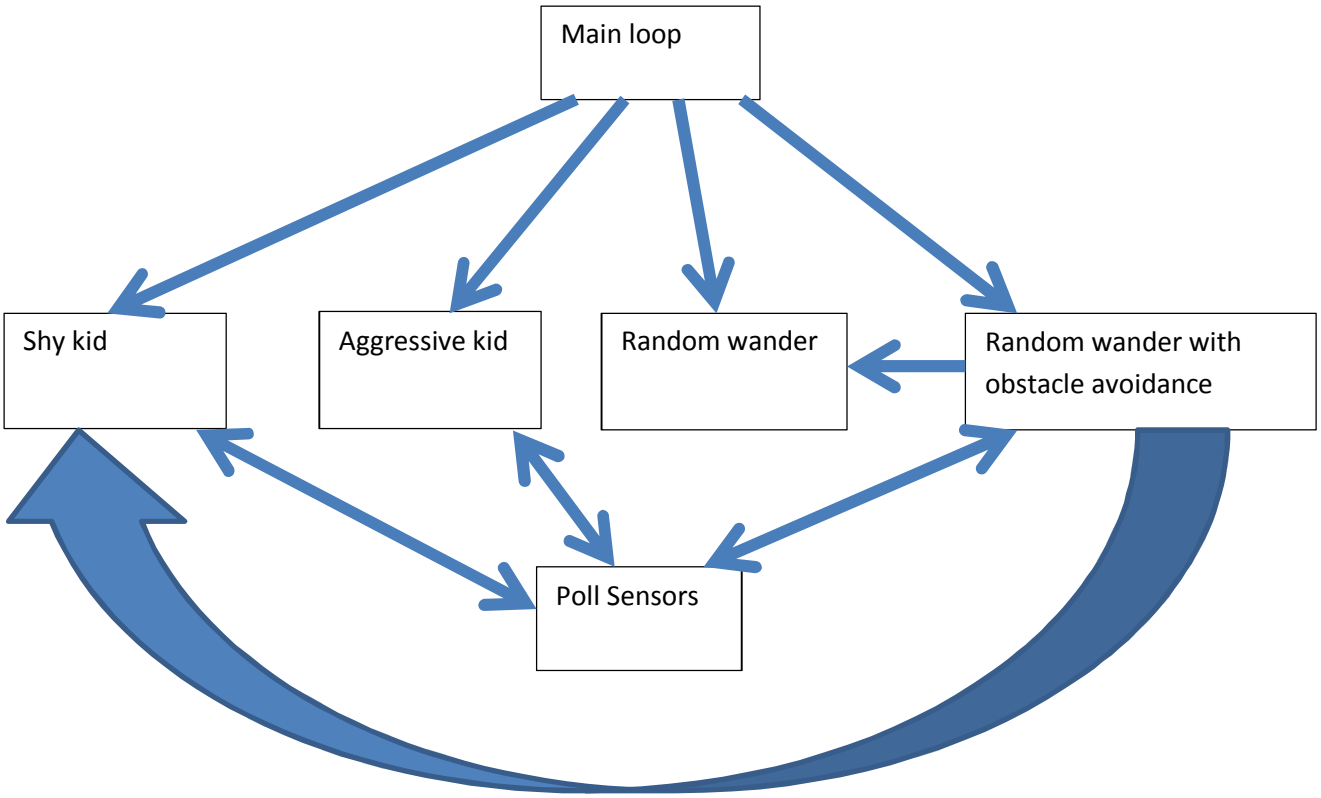
The code we currently have could be improved to make a smart wander routine by replacing random wander code with code that keeps going until it sees something then tries to go around it. If the object is small the robot will avoid it, but if the object is the wall of the room the robot will follow the wall.

If we wanted a go-to-goal and avoid obstacle behavior it could be accomplished easily by replacing the random wander code with go-to-goal while keeping all the sensors polling. This would

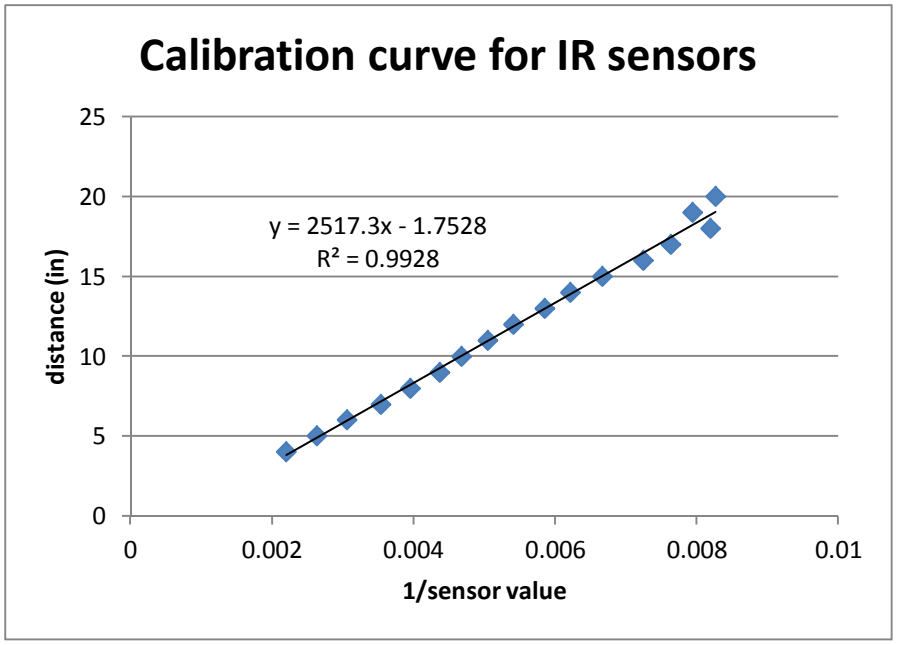
keep Data from running into an obstacle by making it start implementing shy kid behavior if any obstacle comes too close.

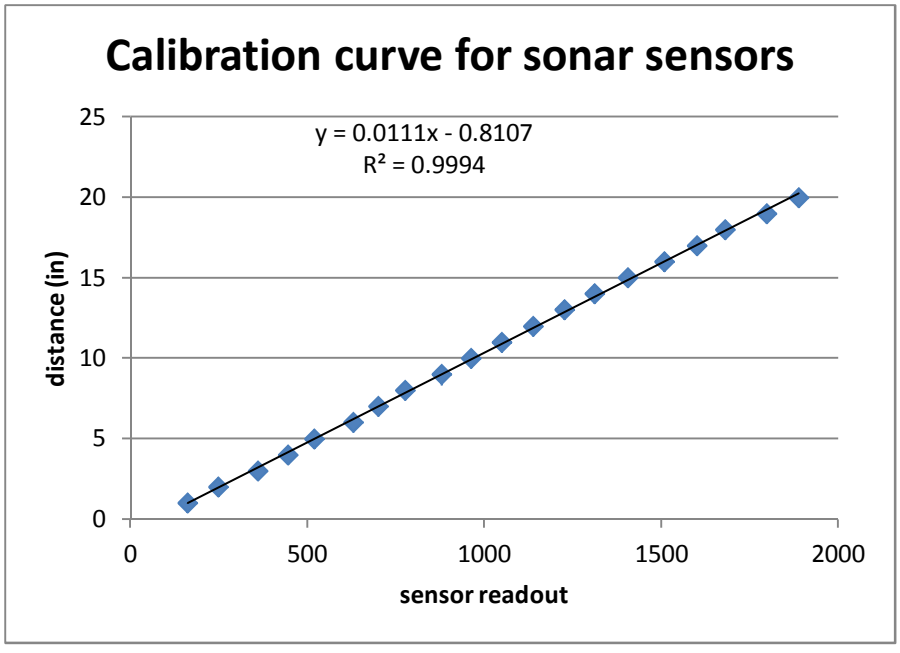
We encountered some errors with the obstacle avoidance behavior. The aggressive kid would occasionally stop at an object then start back up again. We could improve this by making the code have a higher threshold for starting movement again. There were also some obstacles Data could not detect. They were small objects that were either too low for the sensors to see or thin enough that it would slip in between two sensors. The IR sensors also had a really hard time seeing anything close to them. This makes sense because the calibration curve for the IR sensors breaks down at low values.

When changing between random wander and shy kid sometimes shy kid would move too slowly to break away from the shy kid behavior. We fixed this by putting in a minimum motor strength for shy kid to make sure that if the motors are running the robot would always move at least a little bit.



Appendix A:


Calibration curve for IR sensors
y = 2517.3x - 1.7528
R² = 0.9928
distance (in)
1/sensor value
0
5
10
15
20
25
0
0.002
0.004
0.006
0.008
0.01


Calibration curve for sonar sensors
y = 0.0111x - 0.8107
R² = 0.9994
distance (in)
sensor readout
0
5
10
15
20
25
0
500
1000
1500
2000