To: Dr. Berry

From: Team Moravec, Devon Adair and Hunter LaMantia

Date: 12/21/2018

Re: Avoid-Obstacle and Random Wander Lab2

The purpose of this lab was to demonstrate the use of obstacle avoidance techniques such as shy-kid and aggressive-kid. The avoidance was done by using the IR sensors on the robot to determine the distance from objects. Once the robot sees how far away it is from obstacles, it move away at a proportional speed for shy-kid or just stops for aggressive-kid. Shy kid moves away from the object while aggressive kid waits for it to move. The last portion of the lab was to do random wander but with the obstacle avoidance. This uses state machines to determine if it needs to wander or avoid. With our go-to-goal we gave it inputs and it stays in go-to-goal until it detects that it needs to switch to obstacle avoidance. Then last it random wanders when it is done. The state machine for go-to-goal can be seen in Figure 1 in the appendix.

We wrote our code to function in a way that we can enable and disable features quickly. It allowed us to test shy-kid, random wander, and go-to-goal separately and when we wrote the function to make the state machine we could quickly verify each function if we found issues with them.

We also calibrated our sensors, which can be seen on Tables 1 and 2 in the appendix. This was done through a separate file called SensorCalibration.ino, which we included in our zip file. We found that the left and right sensors were poor sensors and the equations were difficult to accurately find due to poor sensor data. We can reliably read between 2-12 inches approximately. Because of this, we only used values up to 6 inches and ignored the negative values obtained from distances under 2 inches. The errors and equations can be found in Tables 2 and 3 of the appendix then Equations 1-6 of the appendix. The data was taken by printing the average values of the sensors at measured distances from a wall. After that we found the equations to linearize and convert to inches. We then took the same measurements with the inches printed out then compared them against the actual distances from the wall which gave us the percent error.

We tried to use the Sonar as redundant sensing but could not get the sonar to work correctly, so we ended up not using them in the functions to detect objects. Objects in the corners were unable to be detected because we didn’t have Sonar and the IR cannot be aimed at the corners. We resolved some of these issues by overshooting obstacles to make sure that there wasn’t an obstacle where we couldn’t see.

Smart wander could be set up to randomly wander until a wall is detected and follow it. If it detects an obstacle, it would go into avoidance. This would last until it returns to its starting point. This method would remove some of the randomness but add some more functionality to make sure it didn’t go where it came from.

Shy-kid had issues with using multiple sensors so we had to use special cases to get the robot out of them and then go back to proportional control. This way we could guarantee that it wouldn’t get stuck in a weird limbo state. When we had to do another function to do obstacle avoidance for go-to-goal, we made it calculate where it was and where it needed to go. This, however, ended up causing recursion issues in avoiding different obstacles. We resolved these issues by making a special version of runToStop that waited until it finished avoiding one obstacle.

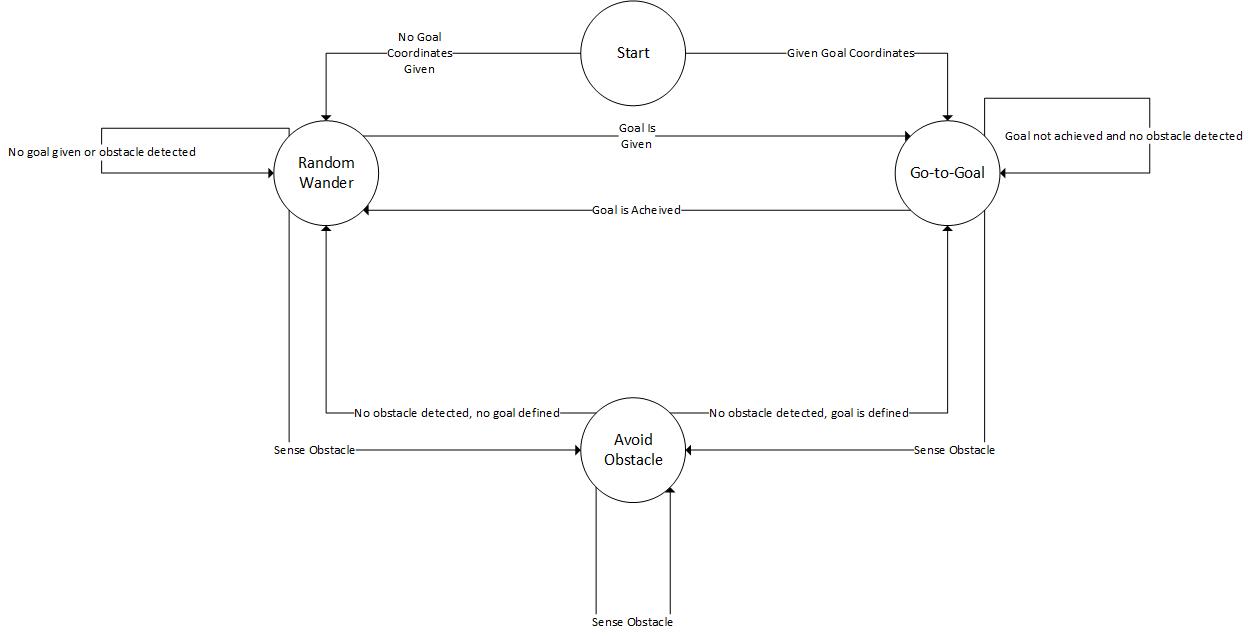
One way we could improve obstacle avoidance is by implementing a PID controller to make going to and around objects smoother. Right now our robot has rough stops and basic proportionality which makes avoidance a little jittery.

We used some booleans to tell the robot when it is doing things like dodging, avoiding, going to goal, or wandering. We used these variables to determine behavior of the other functions. For example, if it was already dodging, then the methods that decided if it needs to dodge did not get called, because the robot knew it was already in that state.

When an obstacle was detected when going-to-goal it would save the target position, current position and track how far it moved to dodge the obstacle. After dodging it recalculated the goal position and called a new instance of goToGoal to get there.

Appendix

Figures



**Figure 1: State Machine for Random Wander and go-to-goal with Obstacle Avoidance**

**Figure 2: Front IR plotted with linear equation**

**Figure 3: Rear IR plotted with linear equation**

**Figure 4: Left IR plotted with linear equation**

**Figure 5: Right IR plotted with linear equation**

**Figure 6: Left Sonar plotted with linear equation**

**Figure 7: Right Sonar plotted with linear equation**

Tables

**Table 1: State Machine States**

|  |  |  |  |
| --- | --- | --- | --- |
| **State** | **Input** | **Output** | **Next State** |
| Random Wander | No coordinates | Goal is given | Go-To-Goal |
| Random Wander | No obstacles | Sense obstacle | Avoid Obstacle |
| Random Wander | Goal is achieved | No goal or obstacle | Random Wander |
| Go-To-Goal | Coordinates given | Goal not achieved, no obstacle | Go-To-Goal |
| Go-To-Goal | No obstacles | Sense obstacle | Avoid Obstacle |
| Go-To-Goal | Coordinates given | Goal achieved | Random Wander |
| Avoid | Sense Obstacle | Sense obstacle | Avoid |
| Avoid | Sense Obstacle | No obstacle, goal defined | Go-To-Goal |
| Avoid | Sense Obstacle | No obstacle, no goal defined | Random Wander |

**Table 2: IR sensor data with error**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Front IR |  |  | Rear IR |  |  | Left IR |  |  | Right IR |  |
| Acutal Inches | Analog Value | Measured Inches | % error | Analog Value | Measured Inches | % error | Analog Value | Measured Inches | % error | Analog Value | Measured Inches | % error |
| 1 | 530 | 0 | 100 | 525 | 0 | 100 | 500 | -1 | 101 | 565 | 0.5 | 99.50495 |
| 2 | 370 | 1 | 50 | 345 | 1 | 98 | 655 | -3 | 103.06 | 667 | -1 | 100.9703 |
| 3 | 270 | 2 | 33.33 | 240 | 2 | 94 | 600 | -1 | 101.06 | 625 | 0.5 | 99.505263 |
| 4 | 210 | 3 | 25 | 190 | 3 | 88 | 490 | 3 | 96.59 | 500 | 4 | 95.858824 |
| 5 | 170 | 4 | 20 | 155 | 4 | 80 | 405 | 7 | 91.25 | 415 | 6 | 93.424658 |
| 6 | 140 | 5 | 16.66 | 135 | 5 | 70 | 350 | 9 | 87.14 | 340 | 8 | 90.819672 |
| 7 | 125 | 6 | 14.28 | 110 | 6.5 | 54.5 | 305 | 10 | 81.65 | 305 | 9 | 88.977528 |
| 8 | 110 | 6.5 | 18.75 | 95 | 7.5 | 60 | 275 | 11 | 81.66 | 275 | 10 | 87.755102 |
| 9 | 95 | 7.5 | 16.66 | 80 | 8.5 | 49 | 250 | 12 | 75.51 | 245 | 11 | 85.432432 |
| 10 | 80 | 9 | 10 | 80 | 9 | 10 | 235 | 13 | 30 | 225 | 12 | 60 |
| 11 | 75 | 9.5 | 13.63 | 70 | 10 | 26.66 | 215 | 14 | 47.5 | 205 | 12 | 74.736842 |
| 12 | 70 | 10 | 16.66 | 65 | 12 | 28 | 205 | 14 | 50 | 190 | 13 | 74 |
| 13 | 60 | 11 | 15.38 | 50 | 12 | 22 | 200 | 15 | 31.81 | 180 | 13 | 59.142857 |
| 14 | 50 | 12 | 14.28 | 60 | 13 | 9 | 190 | 15 | 66.66 | 170 | 13.5 | 79.75 |
| 15 | 50 | 13.5 | 10 | 45 | 13.5 | 35 | 185 | 15 | 57.14 | 160 | 13.5 | 76.375 |
| 16 | 50 | 14 | 12.5 | 45 | 14 | 12 | 180 | 16 | 33.33 | 150 | 14 | 58 |
| 17 | 50 | 14 | 17.64 | 40 | 14.5 | 17.83 | 175 | 16.5 | 7.47 | 145 | 14 | 87.25 |
| 18 | 45 | 16 | 11.11 | 40 | 14.5 | 30.5 | 180 | 16.5 | 45.90 | 140 | 14.5 | 68.410714 |
| 19 | 40 | 17 | 10.52 | 45 | 15 | 42.5 | 175 | 16.5 | 61.17 | 135 | 14.5 | 76.298077 |
| 20 | 45 | 18 | 10 | 50 | 16 | 60 | 180 | 17 | 71.66 | 130 | 15 | 79.069767 |

**Table 3: Sonar Sensor data with error**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | Left Sonar |  |  | Right Sonar |  |
| Acutal Inches | Analog Value | Measured Inches | % error | Analog Value | Measured Inches | % error |
| 1 | 390 | 2 | 100 | 350 | 2 | 100 |
| 2 | 450 | 2 | 0 | 430 | 2.5 | 25 |
| 3 | 555 | 3 | 0 | 490 | 3 | 0 |
| 4 | 645 | 4 | 0 | 580 | 4 | 0 |
| 5 | 790 | 5 | 0 | 765 | 5 | 0 |
| 6 | 940 | 6 | 0 | 910 | 5 | 16.6666667 |
| 7 | 1090 | 7 | 0 | 1050 | 6 | 14.2857143 |
| 8 | 1215 | 8 | 0 | 1235 | 7 | 12.5 |
| 9 | 1380 | 9 | 0 | 1370 | 7 | 22.2222222 |
| 10 | 1535 | 10 | 0 | 1450 | 8 | 20 |
| 11 | 1665 | 11 | 0 | 1670 | 8 | 27.2727273 |
| 12 | 1810 | 12 | 0 | 1620 | 9 | 25 |
| 13 | 1975 | 13 | 0 | 1770 | 10 | 23.0769231 |
| 14 | 2095 | 14.5 | 3.571428571 | 1910 | 10 | 28.5714286 |
| 15 | 2220 | 16 | 6.666666667 | 2155 | 12 | 20 |
| 16 | 2370 | 17 | 6.25 | 2385 | 13 | 18.75 |
| 17 | 2535 | 19 | 11.76470588 | 2410 | 16 | 5.88235294 |
| 18 | 2670 | 20 | 11.11111111 | 2500 | 17 | 5.55555556 |
| 19 | 3620 | 22 | 15.78947368 | 2620 | 18 | 5.26315789 |
| 20 | 2950 | 23 | 15 | 2775 | 18 | 10 |

Equations

**Equation 1: Front IR Sensor**

**Equation 2: Rear IR Sensor**

**Equation 3: Left IR Sensor**

**Equation 4: Right IR Sensor**

**Equation 5: Left Sonar Sensor**

**Equation 6: Right Sonar**