91.450, Robotics I
Fall 2003
Prof. F. Martin

Lab 4: Wall-Following

Out: 26 September 2003
Due: 3 October 2003, beginning of class

Part I: Exercises to be done individually

Problem 1, Combining Behaviors: Exercise 4.4 of Murphy, p. 150.

Problem 2, Potential Fields: Exercise 4.12 of Murphy, p. 150.

Problem 3, Broken Sensor: Exercise 4.13 of Murphy, p.150.

Problem 4, Frame Problem & Open-World Assumption: Exercise 4.14 of Murphy, p. 151.

Part II: Lab

In this lab, you'll use the optical rangefinder sensor (aka the "ET sensor" because it looks like a pair of big innocent eyes) to build a wall-following robot. This is a floating analog sensor and must be plugged into one of the floating analog ports (16-19). You can access the sensor with the analog(#) function.

The sensor's range is 4 to 30 inches. Low values indicate a large distance and high values indicate a distance approximating 4 inches. If something is closer than 4 inches, the values will go high again.

Mount the sensor on one side of your robot (again, you can continue to use the Handy Bug base or redesign your robot to your liking). If you mount on the left, your robot will follow walls to its left; if you mount on the right, your robot will follow walls to its right. (If you'd like, you may mount sensors on both sides, and then choose to follow whichever side is near a wall.)

Exercise 4a in Section 3.6.5 of Martin on p. 127 discusses two ways to write a wall following program. The first is to have two states: one if the sensor reading indicates that the robot is too close to the wall and the other that the robot is too far. In the too close state, the robot should turn away from the wall. In the too far state, the robot would turn towards the wall. You may select the distance at which you'd like to follow the wall. I'd recommend that you pick a distance a bit away from the 4" minimum, since you'll get readings that increase from this point whether you get closer or farther away. Implement this solution. Turn in your code.

The second way to implement wall following code is to have three states: too close, just right, and too far. Modify your code to include the third state, the ideal distance from the wall. In this state, you'll drive forward. Experiment with the width of the "just right" range. What works best? Turn in your code.

How do the two programs compare? Which one does a better job following the wall? How far can your robot travel down the hall? What does it do when there are doorways with open doors? What does it do with recessed doors?

Show me your hall following robot that uses the method that you think is best. You can show me your robot up until the end of class on Tuesday, October 3.