COMP470: Mobile Robotics

Spring 2020

Lab 4: Sonar Range Measurements

The purpose of this lab is to add range sensing capabilities to your robot and use feedback from this sensor to execute driving commands without running into objects.

There are four phases to this lab:

1. Connect the sonar sensor and obtain range measurements.
2. Modify the sensor library to use floats instead of long ints.
3. Calibrate the sonar sensor.
4. Use the calibrated sensor to drive in a "maze" race.

Phase 1

1. Obtain one of the 3D Printed sensor brackets and two screws from the instructor.
2. Attach the sensor to the bracket and use the two screws to attach the sensor to the robot.
3. Connect the cable to the sensor.
4. Install the sensor library.
    1. Download the zip file of the repository at this URL. Do not clone the repository. Simply download the zip file.
    2. Open the Arduino IDE on your laptop and follow the directions for Importing a .zip Library here.
    3. Connect your laptop to the USB-Serial board.
    4. Open Arduino IDE, go to "File"->"Examples"->"Grove*Ultrasonic*Ranger"
    5. Run this example program and then go to "Tools"->"Serial Monitor"
    6. You should see printed to the console a message like: "The distance to obstacles in front is: " with some **NONZERO** values printed below. If you see nonzero values you are ready to move on. If you see only zeros then you have a problem. See instructor.

Phase 2

In the previous phase you should have seen integer readings being printed to the console. I don't like this, so we will modify the library to use floats instead of long ints.

1. Modify the files Ultrasonic.h and Ultrasonic.cpp to return float values for the measurements.
2. Before moving on to the next phase demonstrate to the instructor that this works.

Phase 3

Now that we have a properly working sonar sensor we need to calibrate the sensor to make sure the distance we measure is actually the true distance. Place a board in front of the car at distances of 5 cm to 100 cm in increments of 5 cm using a meter stick. At each point record the reading returned by the sensor. Plot the best fit line through your data and find the parameters for slope and intercept.

Use MATLAB online to generate a plot of your data and to determine the best fit line.

Phase 4

In this phase we will combine the code you developed in Lab 2 with the sonar sensor to enable your robot to drive while avoiding walls. We will have a timed race in the wooden maze in the lab. Your robot will need to complete one circuit of the maze without hitting a wall. Each team will be timed and ranked. This means that you will need to drive as fast as you can, but not too fast so that you can make accurate measurements from your sensors and stop your car before you hit the wall. I recommend starting slow and speeding up after you get the hang of it.

Lab Report

Turn in a zip file containing all source code and a written lab report documenting the procedure you used for calibrating your sensor and a figure showing the data points and the best fit line.

Questions to include in report.

1. Would your robot have performed better in the race without using the sonar sensor? Why or why not?
2. Propose a modification to the robot that would improve the performance.