CSCI 445 Lab 2 — Sonar Characterization

Prof. Culbertson, University of Southern California Spring 2020

In this lab we will extend our usage of the robots' functionalities by using a sonar mounted on a servomotor such that we can rotate the sonar. Furthermore, we will characterize the quality of the sonar by experimentation.

1 Prerequisites

- Be familiar with pulse-width-modulation, motors, and ultrasound sensors (sonar).
- Bring your laptop.

2 Test Sonar

Start by downloading `lab2.zip` from Piazza and extract the containing files into your lab directory. Note that the archive just contains the new files, so make sure that `Lab2.ttt`, `lab2_servo_test.py` and `lab2_sonar_test.py` are in the same directory as `Lab1.ttt` from the first lab session.

Start V-REP and open `Lab2.ttt`. Use the following command to get sonar readings in simulation:

```
[laptop]$ python3 run.py --sim lab2_sonar_test
```

Now copy the files to the robot and repeat the test on the robot.

```
[create]$ python3 run.py lab2_sonar_test
```

We implemented the sonar driver for you using a GPIO on the odroid. Please skim the documentation of the sensor[1] and read the code in `pyCreate2\robot\sonar.py`. We will ask questions about that as part of your lab grade!

3 Test Servo in Simulation

Run the following example program to test the servo in simulation:

```
[laptop]$ python3 run.py --sim lab2_servo_test
```

The sonar should move from the center to the left and then to the right. The servo is limited to 180 degree movements.

4 Implement Servo Support on Robot

Please read the documentation of the servo[2]. As mentioned there, it is controlled by PWM. Fortunately, the Odroid has hardware support for PWM and we wrote a small python wrapper for you in `pyCreate2\robot\pwm.py`. Add code to `pyCreate2\robot\servo.py` to actually go to a specified angle with the same interface as provided for the simulation (see `pyCreate2\simulation\servo.py`). Once you are done, test your implementation on the robot using

```
[create]$ python3 run.py lab2_servo_test
```

Hint: You'll need to find the pulse length and interval length in the documentation.

[1] https://www.parallax.com/sites/default/files/downloads/28015-PING-Sensor-Product-Guide-v2.0.pdf

[2] https://www.parallax.com/sites/default/files/downloads/900-00005-Standard-Servo-Product-Documentation-v2.2.pdf

5 Characterize Sonar

You will now execute several experiments with the physical sonar to characterize its behavior. Furthermore, you will update the model in V-REP based on your findings.

5.1 Distance Range Characterization

Use the wooden blocks and the tape measures provided to find the minimum and maximum distance sensing range of the sonar sensor. In the task sheet, write down these values (in meters) in the designated sections, along with a short description of your method used to do the measurements.

Hint: Place the blocks exactly opposite the sonar for your measurements. You will find out why this is important once you do the task in Section 5.4.

5.2 Measurement Accuracy Characterization

Pick 10 distances within the range you found in Section 5.1 and write these down as ground truth distances (or actual distances) for your measurements in your task sheet. For each value selected, place a wooden block at that distance away from the sonar sensor (place in the opposite). Do at least 100 measurements for each distance selected, calculate the measurement errors (differences between the ground truth distances and the measured distances), and store these errors for each measurement. Use the recorded errors to calculate the mean and the standard deviation (SD) of the errors in measurements.

You should also plot a histogram diagram of the data errors for all your measurements and demonstrate it to the TAs.

Hint I: You should use a loop in your script to automate the measurements.

Hint II: For calculating the discrete mean and standard deviation, you should use the `numpy` module for python (installed on the robots). You can use `np.std` and `np.mean` for the computation.

Hint III: When you want to plot on the robot, you need take into account that there is no display attached to the robot. We recommend plotting using `matplotlib`. To plot on the robot and save it, you can adapt the following snippet:

```python
import matplotlib
matplotlib.use('Agg') # use off-screen rendering (must go before any plotting code)
import matplotlib.pyplot as plt

# other code here

plt.plot([0,1,2])  # do your plotting here
plt.savefig("test.png")
```

5.3 Angular Range Characterization

Here, you will determine the angular sensing range (a.k.a. the cone size) of the sonar sensor provided. The angular sensing range of a sonar is the angle of the right cone within which the sonar will detect objects. Determine the size of this angle and write it down in the task-sheet.

5.3.1 Modify V-REP Model

Now that you have determined the cone size of the sonar sensor provided, you must modify the V-REP model of the sonar sensor to match the determined angular range of the physical sensor provided. Please read the documentation at http://www.coppeliarobotics.com/helpFiles/en/proximitySensorVolumeDialog.htm to find out how to do that.

5.4 Critical Angle Characterization

Here, you will find the critical incidence angle for the sonar sensor. Critical angle is the incidence angle beyond which the measurements will become highly inaccurate due to lost/misdirected reflected sound waves necessary for time-of-flight[3] measurements of distance. Determine this angle for the sonar sensor provided. In your task-sheet, write down this angle, and explain briefly the method you used to determine this angle.

5.5 Sound Reflection from Various Interfaces

Sound waves reflect at different rates from different materials[4]. Try using foam as an object for proximity sensing and write a short summary describing your observation, the distance value you measured, and reasons as to why that is the case.

[3] Time-of-flight measurement: https://en.wikipedia.org/wiki/Time_of_flight

[4] Sound waves striking an interface: http://goo.gl/1aEcY3