ME 499/599: Computer Programming for Mechanical Systems

Winter 2016

Lab 2: Basic Sensor Processing

Purpose

The goal of this lab is to give you some more experience writing Python functions, and to introduce you to some simple sensor processing ideas.

Preparation

Make sure you're familiar with all of the material and concepts from the previous two labs.

Assignment

1. Do exercise 5.3 from the book. Make sure you pay attention to *when* Fermat's Last Theorem applies.

2. Do exercise 6.8.

3. **Graduate students:** Do exercise 7.5.

4. Get a copy of `sensor.py` and `null_filter.py`. Read them, and make sure you understand what they're doing. Generate a file of (simulated) sensor data, and plot it with `Gnuplot`. Notice that it's noisy.

5. Write a new filter that replaces each sensor reading with the mean of itself and the readings on either side of itself. For example, if you have these measurements in the middle of the list

   10 13 13

   then the middle reading would be replace by 12 (since (10 + 13 + 13) / 3 =

12). Generate two files, one for the data before filtering, and one for the filtered data. Plot both of these on the same graph in `Gnuplot`.

6. Modify your code to have a variable filter width. Instead of a width of 3, add a parameter to the filter function that specifies the filter width. Test this with a variety of widths, and see what it does to the data.

7. Write a variable-width median filter. This is like the mean filter that you just wrote, except that it replaces the values with the *median* of the values, rather than the mean.

8. Write a function that calculates the statistics of the data, and prints them out. You should print the number of data points, the mean measurement, standard deviation, median measurement, the maximum and minimum measurements, and what percentage of measurements are more than one standard deviations from the mean.

Grading

To get points for this lab, you should show your code to the class TA. If you need some extra time to complete the lab, then you can show your solutions to the TA at the start of next week's first lab with no penalty.

There are 16 points available for 499 students, and 20 points for 599 students. 599 students are expected to do more work, since they're taking the graduate version of the class, hence the larger number of points.

1. Exercise 5.3: 1 point for a function that does the right thing. 1 point for dealing with user input correctly. 1 point for some sort of input validation. 1 point for dealing with the specific setting in which Fermat's Last Theorem applies. [4 points total]

2. Exercise 6.8: 2 points for getting the function right. [2 points total]

3. **Graduate students:** Exercise 7.5: 2 points for getting the basic function right. 1 point for getting the iteration right. 1 point for correctly comparing it to pi. [4 points total]

4. 1 point for being able to explain what sort of noise that the fictional sensor is injecting into the system. [1 point total]

5. 1 point each for being able to demonstrate your mean filter does the right thing for filter widths of 1, 3, 9, and 27. [4 points total]

6. 2 points for being able to demonstration that your median filter does the

right thing for filter widths of 1, 3, 9, and 27. [2 points total]

7. 2 points for calculating all of the statistics and printing them out. 1 point for getting the percentage correct. [3 points total]

8. Uploading your code to the TEACH web site, before the end of the day after *next lab* on the 26th. (0 points, but we won't assign a grade for the lab until you do this)

Thoughts

1. You should put in some error checking in your code, but don't worry about checking for *every* thing that could go wrong, especially in user-entered data.

2. For exercise 7.5, you should split this up in to parts, and should implement a separate function to calculate factorials.

3. The null filter doesn't actually do anything to the sensor readings. It's really just to show you how you might write a filter for a list of sensor values. Don't worry if you don't understand the stuff in `sensor.py`; we'll cover that in class soon. All you really need to know is that it generates a list of sensor measurements for a fictional sensor.

4. To see the noise in the simulated sensor signal, either make the `Gnuplot` window really big, or pass in a larger `noise` parameter to the function. Google and the Python documentation will help you answer the question associated with this part of the exercise.

5. The data from the mean filter will have two fewer measurements that the original data set, because the mean of the first and last elements is not well-defined. Don't worry about this.

6. You might find this line of `Gnuplot` helpful for plotting two graphs together.

```
plot "raw" with lines, "filtered" with lines
```

7. Mean filter widths only really make sense for odd values. You should decide what to do with an even value. You can either reject it, or you can do a lop-sided filter. Your choice.

8. `numpy` contains a useful set of functions. Google knows about it.

The Rules

Everything you do for this lab should be your own work. Don't look up the answers on the web, or copy them from any other source. You can look up general information about Python on the web, but no copying code you find there. Read the code, close the browser, then write your own code.

Page written by Bill Smart.