CSCI 4950/6950 Assignment 3

In this assignment, you will be working on remove noise from sensor data and implement a simple live step counter (pedometer).

The following parts of the assignment contains 1 Jupyter notebook and 1 python file. You are required to add your own code in them wherever mentioned. Make sure you read each file carefully so you don't miss any part of the assignment.

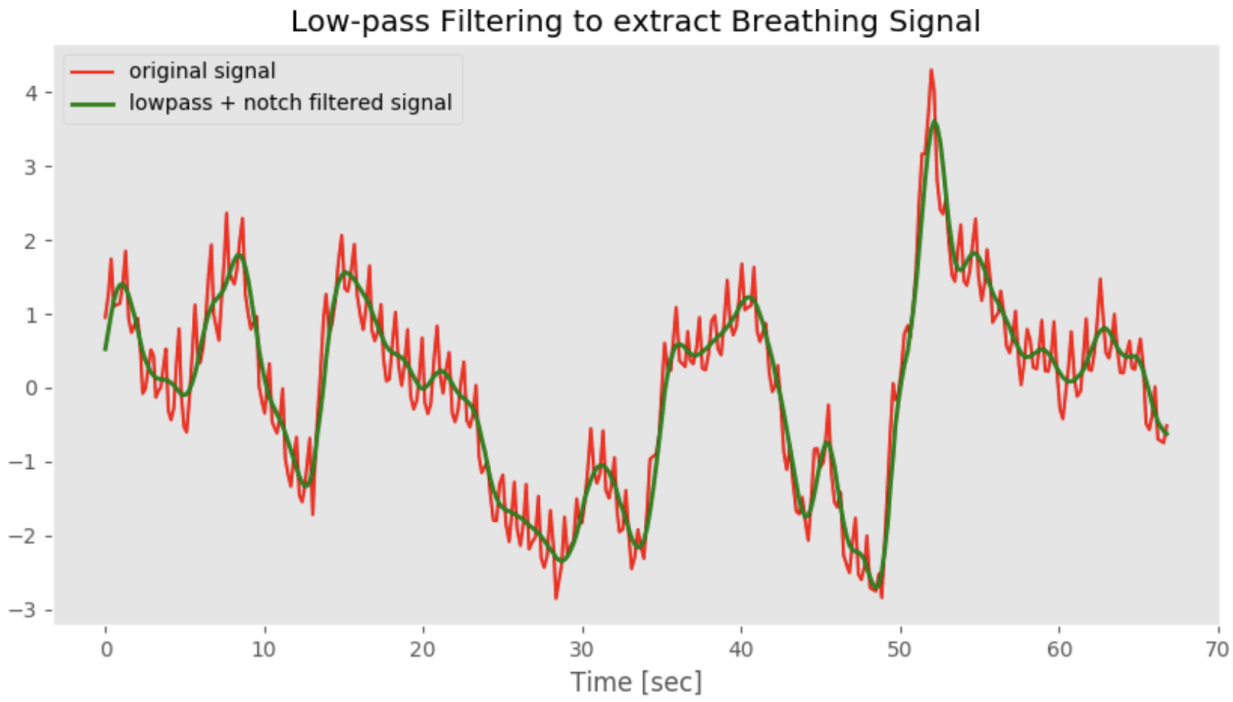
GitHub Classroom Link: https://classroom.github.com/a/S5Y9fiEC

Part 1: PPG Data Processing (50 Points).

- This assignment contains one Jupyter notebook, "PPG-Assignment.ipynb", that you are required to complete and submit. You will be estimating heart rate and breathing rate from a given PPG signal after filtering it appropriately.

- You can import any methods you choose from `scipy.signal`, but other additional imports are not allowed.

- Sample graphs of filtered signals are given below to help you tune your filter parameters. Note that these are just guidelines and your graphs may not turn out to be exactly similar.



- More instructions are contained in the Jupyter notebook in the places where you are required to fill in the code.

- Commit and push your changes periodically. Only the last commitment before the deadline will be graded.

Part 2: Live Step Counting (25 Points)

We'll be using Sensor Logger (https://www.tszheichoi.com/sensorlogger) app for streaming, you can also think about using this app for your final project:

For iOS: https://apps.apple.com/app/id1531582925

For Android:
https://play.google.com/store/apps/details?id=com.kelvin.sensorapp&pcampaignid=pcampaignidMKT-Other-global-all-co-prtnr-py-PartBadge-Mar2515-1

The above link is also available on the app's website.

1. Connect both your laptop and phone to the same WiFi router. If you are on campus, eduroam and Au-Secure may not work due to restriction, in which case, you can connect your laptop to your phone's hotspot.

2. Reopen the app and navigate to the "Logger" tab.

3. Tap on the radio button beside the word “Accelerometer”.

   IMPORTANT: If you are unable to install and get the app running, let me know.

4. Once your app is up and running, find and open the python file “sensor_logger.py” from the starter repository in any code editor of your choice.

5. When you run the script in terminal “python3 sensor_logger.py” you will see something like this

   You will see something like these on the command prompt:

   ```
   Dash is running on http://0.0.0.0:8000/

    * Serving Flask app 'sensor_logger'
    * Debug mode: off
   WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
    * Running on all addresses (0.0.0.0)
    * Running on http://127.0.0.1:8000
    * Running on http://192.168.212.126:8000
   Press CTRL+C to quit
   ```

   The line:

   ```
    * Running on http://192.168.212.126:8000
   ```

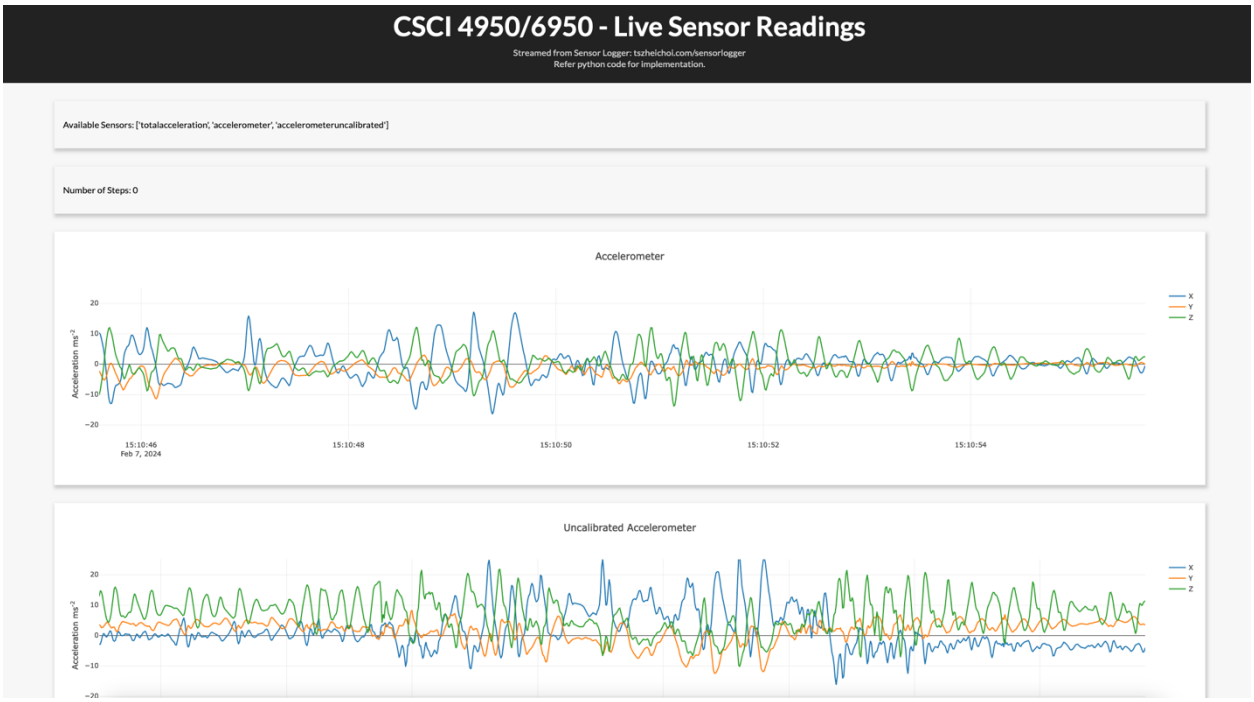
   shows the IP address the server is running. **set the same IP address on the senor logger app.**

   ---

6. Copy the ip address. Go to settings ⚙ and enable http push. Then paste the ip address e.g, `http://192.168.212.126:8000/data` in the push url and tap to test pushing.

7. Go back to the logger page and press start recording.

8. Now navigate to the ip address on a web browser of your choice (other than Safari) and you will see the following web page if you run your updated script.



9. To complete the assignment, you need to edit the Python script and add code to do the following:

    - count steps by following a pipeline similar to assignment 2: filter the raw signal using appropriate filters and apply your own step detection algorithm on the filtered data.
    - plot the points where the steps occurred on the magnitude graph (as markers on the magnitude line).

10. Rerun the script and repeat the steps from 5 to 8 until you have a simple live step counter.