

Activity 1: Sensor Data Analysis

MUC Fall 2023

Hello! This activity is designed to familiarize you with using sensors to record data, and then analyze and manipulate the data you get from them. This activity has multiple tasks in increasing order of difficulty.

You will have 5 Tasks for this activity. Tasks include a mix of coding and reflection questions. First 4 tasks are mandatory questions which make up 100 points while Task 5 is a bonus question.

Please follow the proper template (naming convention for the coding tasks). We want the following 2 submissions:

A. PDF Report (filled out deliverable template)

B. Jupyter Notebook file, relevant code, and collected/processed data as a single .zip folder

*We will not look at anything else that is submitted.

1. **GitHub link for template code (login with gatech credentials):**
<https://github.gatech.edu/dmace3/muc-fall-2022-sensor-data>
2. Please make sure that you **don't get stuck** for long at any place and feel free to **ask for help**, that is what we are here for. You can also leave a message on canvas.
3. Tasks 1 through 4 - You will be majorly graded on what you put in the report. Please add in your plots and explanations in the report. We will only be looking at the code only where necessary.
4. Link to the assignment [slides](#) and [deliverable format document](#):

Preparation for Class

Step 1: PYTHON INSTALLATION AND EYE BALLING SENSOR DATA

Python 3.7 is required for this assignment

- a. Install Python (3.7+) with scientific packages like scipy, numpy, matplotlib. We recommend using [anaconda](#). The advantage of using anaconda is that it automatically installs scientific packages for you. For those new to python, this may be a good resource to get started: https://github.gatech.edu/omscs6601/assignment_0

(If you want to use Jupyter notebook please complete the following step, if not skip part b.)

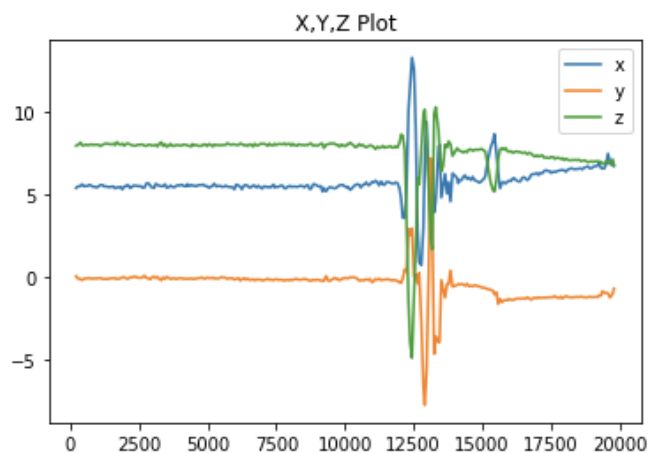
- b. Install Jupyter using the instructions here (in case you don't already have it, though it is typically installed with anaconda): <https://jupyter.org/install>. These instructions will help you get started: <https://jupyter.readthedocs.io/en/latest/running.html#running>.
 - i. Basics [here](#)
 - ii. Running code [here](#)

Use the Jupyter notebook template to execute the in class tasks (1 to 5).

You can create your own Jupyter notebook for the class prep.

(You can use Google Colab. If you do, skip the above two steps)

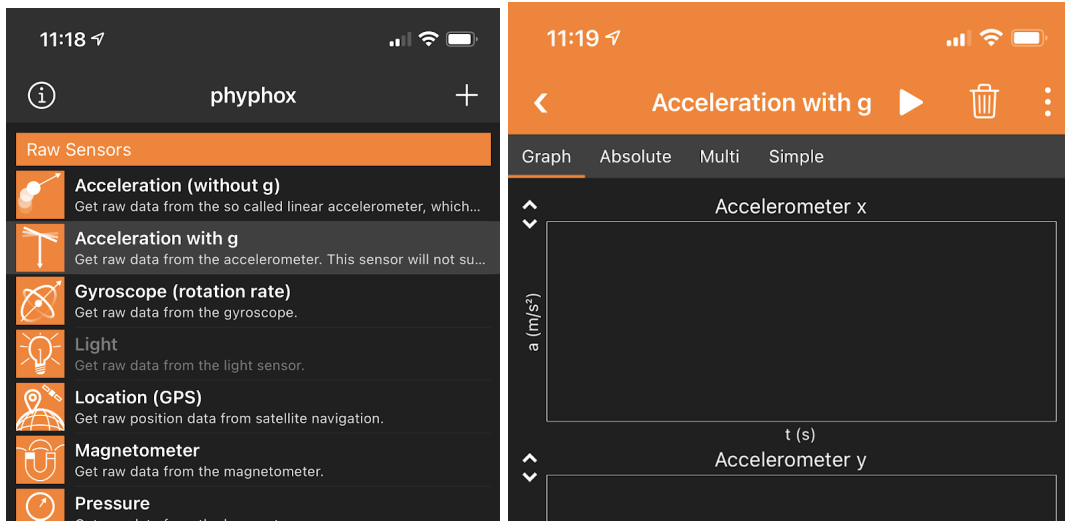
- c. Download the code from the GitHub [link](#) given. Run the `test.py` file. It should plot a sample accelerometer data for you as shown below. If you see this, then you are all set for the activity.



Step 2: LEARNING TO COLLECT DATA

We will need to first collect data we want to analyze. Please read the instructions below to learn.

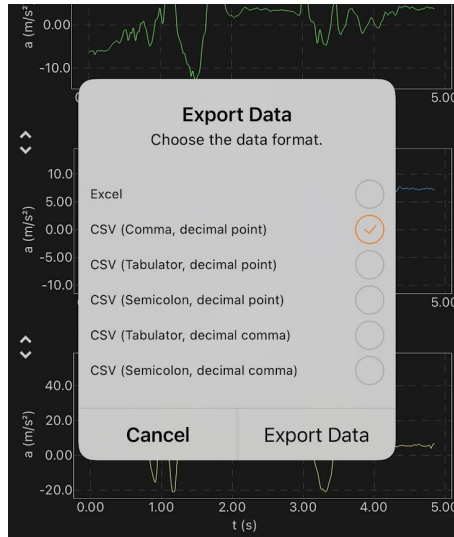
- a. Install phyphox (<https://phyphox.org/download/>) on your phone. You can find the apps for both android and iPhone in the above link.
- b. After launching the app choose “Acceleration with g” from the available sensors and try recording some data by pressing the play button.



- c. Once you have familiarized yourself with recording some sensor data, **try exporting the data from the phone to your laptop and view it.**

To export data to CSV follow these instructions:

1. Hit the three dot button on the top right of the app
2. Select Export Data from the menu options
3. When you export **use option of CSV (Comma, decimal point)** in the screenshot below:



D. Eyeball your exported data to check collected data. You should have data columns like below. You don't have to show us this data, this is to make sure that you know how the software works. :

	A	B	C	D	E	F	G
1	Time (s)	Acceleration x (m/s^2)	Acceleration y (m/s^2)	Acceleration z (m/s^2)	Absolute acceleration (m/s^2)		
2	0.00E+00	2.20E-01	3.84E+00	8.16E+00	9.02E+00		
3	1.01E-02	3.80E-01	3.87E+00	8.15E+00	9.03E+00		
4	2.01E-02	5.76E-01	3.97E+00	8.39E+00	9.30E+00		
5	3.02E-02	7.64E-01	4.12E+00	8.72E+00	9.67E+00		
6	4.02E-02	8.38E-01	4.26E+00	8.98E+00	9.98E+00		
7	5.03E-02	8.76E-01	4.26E+00	8.97E+00	9.97E+00		
8	6.03E-02	7.84E-01	4.16E+00	8.75E+00	9.72E+00		
9	7.04E-02	7.14E-01	4.04E+00	8.49E+00	9.43E+00		
10	8.04E-02	6.94E-01	3.96E+00	8.35E+00	9.27E+00		
11	9.05E-02	5.75E-01	3.98E+00	8.36E+00	9.28E+00		
12	1.01E-01	3.45E-01	4.04E+00	8.30E+00	9.24E+00		
13	1.11E-01	9.76E-02	4.05E+00	8.09E+00	9.05E+00		
14	1.21E-01	-1.47E-01	3.98E+00	8.13E+00	9.05E+00		

Note: In general, column names change with the sensors you choose to record. Also, take note of relative_time units. For example, these time samples are in milliseconds.

Another Note: You may not have recorded values for every timestamp in your csv. This happens sometimes, and it is up to you to figure out how to best deal with it. If you have a lot of missed values, it is probably not a great idea to simply delete all of those timestamps. **Some phones might have less missed values than others, so it is good to test out multiple phones within your group.**

Step 3: COLLECT DATA FOR THE THURSDAY CLASS

In step 2 you learned how to collect sensor data and in this step we prepare for the next activity. Please collect the following data before coming to the class on Thursday.

Data 1: walking_steps_1.csv

Keep your phone inside your jeans/pant pocket



Data to be recorded: Record three different axes for the accelerometer (X, Y, Z) to collect **100 steps walking data**.

Export file to your laptop and name it (auto-exports as a single file): walking_steps_1.csv

Data 2: walking_steps_2.csv

Now change the placement of your phone. Instead of keeping it inside your jeans/pant pocket, you can try your back pocket, another pocket or hand as well.

Data to be recorded: Repeat accelerometer (X, Y, Z) records for **100 steps walking data**

Export file to your laptop and name it: walking_steps_2.csv

Method of data collection and ground truth:

1. Start sensor recording in the phone app
2. Put phone in your front pocket
(Orientation matters! For each data collection activity below we have mentioned orientation of the phone, please follow that)
3. Stand still for 3 seconds, then start “doing the activity” as mentioned below. **Keep a mental count of steps taken. That will be used as your ground truth.**
4. Stop the activity and stand still for 3 seconds.
5. Take the phone out of your pocket and Stop sensor recording.
6. Export your data Name it correctly as mentioned at top - e.g walking_steps_2.csv
7. Add to the data folder in the directory which you downloaded from github.

TASK 1 (15 points): What is the sampling rate

Sampling rate is the number of samples of sensor data per second. It is measured in Hz.

Calculate:

- a. Open **walking_steps_1.csv** collected and check the **time (s)** column. What is the sampling rate for the accelerometer data?

Reflect :

- b. Is the sampling rate stable for the file **walking_steps_1.csv**? What is the variation? If it is not stable, explain what could be affecting the stability of the rate? (Hint: https://pure.au.dk/ws/files/93103132/sen099_stisenAT3.pdf, no need to thoroughly read the whole paper, this is just to lead you to think in the right direction. Looking at the first two pages should be sufficient)
- c. How does your understanding from **part (a)/(b)** affect your data processing pipeline?
- d. How can we make the data easier to work with (hint: moving average, [re-sampling the data](#), and refer to [this](#) for nice visualizations of Fourier transforms)? Choose one method of your choice and **briefly** explain how the math works.
- e. Do you think the current sampling rate is a good basis for capturing human movements like walking? Explain your reasons for or against your case.

Learning objective: Familiarize yourself with the sampling rate, its meaning, and units of frequency, need for resampling

TASK 2 (15 points): Clean your data

Data collection instructions mention standing still for 3 seconds after putting your phone in the pocket, before starting the activity (walking), and again standing still after the activity is finished. This was instructed so you can **easily remove garbage data** at the beginning and the end of the file (basically readings for putting the phone in the pocket and taking it out). To clean up the data, please follow the steps below:

1. Plot walking_steps_1.csv.
2. Through **visual inspection**, from the plot **find the timestamp** (note time is in ms!) window when the phone was being put in the pocket and removed from the pocket. This is garbage data. You need to take it out.
Note : You can automate the process of garbage data removal. Data science researchers usually do this as well. For the purpose of this class, and keeping tasks manageable we are asking you to find timestamp windows containing garbage data and remove it.
3. Once you've figured timestamp windows in the previous step, write code in **clean_data** function to clean up the data. Save your cleaned data in the **data folder**. For example - you will **add a new file in the data folder for walking_steps_1.csv as walking_steps_1_clean.csv**
4. Now repeat the process for walking_steps_2.csv. Save walking_steps_2_cleaned.csv in the data folder.

Learning objective : Understanding that data collected usually contains garbage data at the ends, and the activity of interest data is in between the file.

TASK 3 (40 points): Step counts - walk

Next, we want you to use your smartphone to count how many steps a user walks indoors. Develop a method to do this. Validate your method using the **cleaned data and ground truth** you have recorded. Example: You could record sensor data from your phone while walking say 100 steps. Your analysis method would then need to automatically and correctly count those 100 steps. You have to explain your method, provide the source code you used, and report the results of your automated pedometer.

For this task, you already have `walking_steps_1_clean.csv`, `walking_steps_2_clean.csv` in the data folder

- 1) Implement **vector_magnitude** function for calculating the magnitude of **walking_steps_1_clean.csv**. Plot the magnitude and compare it with the raw data.
- 2) Implement **moving_average** function to use as a **low pass filter for magnitude** you calculated in step1. Plot the **raw magnitude** and the **filtered magnitude** of **walking_steps_1_clean.csv**. Observe the difference between the two.
- 3) **Develop** a method for automated step counting. Utilize the functions **vector_magnitude** and **moving_average** you implemented while doing this (you might want to try different `window_size` values for your `moving_average` function). Use **count_steps** function for this.
- 4) For `walking_steps_2_clean.csv` repeat steps 1-3.

Learning objective: Familiarize yourself with concepts of the magnitude of an accelerometer, filtering using moving average, calculate steps counting using these simple methods, understand the difference between ground truth and implemented algorithms.

TASK 4 (30 points): Steps counts - climb

This task is a variation of task 3.

We want you to use your smartphone to count the number of steps climbed on a staircase, specifically three floors on a spiral staircase. A fire exit staircase in an apartment building should be fine for this. The staircase should have a landing space or flat area between flights. The goal here is to record data where you are walking and data where you are climbing.



Develop an analysis method to achieve the same. Validate your method using the data you have recorded. You could, for example, record sensor data from your phone while walking from the 1st floor to 4th floor. Your analysis method would then need to **automatically and correctly count the number of steps climbed**. You have to explain your method, provide the source code you used, and report the results.

Hint: you can use **ANY combination of sensors** on your phone (think about what these might be. Which all sensor *changes with the change in height and movement?*)

**** NOTE:** we know normally you would use Barometer data but there is an issue with the app where the sampling rate (especially on iOS) of the barometer is very slow. Please do not use barometer data and use different sensor(s) **

- 1) EITHER:
 - a) Collect data using your smartphone, this time the task being to walk up three flights of stairs on a spiral staircase.
 - b) Utilize the data from Canvas contained in the **stairs.zip** file
- 2) Import the recorded climbing data - **climb_steps.csv** into your data folder in Python program.
- 3) While collecting data on stairs there were times when you were also walking rather than climbing. It is important to remove the parts from the data where you were walking in

between the flight of stairs. Write your own algorithm to find segments in data which corresponds to climbing only.

- 4) Implement **segment_climbing_walking function** to segment periods of climbing from walking. Write a plotting function to represent segmented data. Next, re-use the count_steps function you implemented previously (of course you will have to modify it a bit!) to find the number of steps climbed. Running this code should give plots for segmented data, plots of labels for number of steps climbed.

Learning objective: understand concepts of segmentation, comprehensively mixing concepts of data cleaning, segmenting, filtering, recognition (Bulling's pipeline).

Task 5: Bonus (10 points): Whose step?

For this task, you are given two datasets (combined_data folder in Canvas) that each contain accelerometer data recorded from each of the TA while walking. The **training set contains the ground truth information** (which data belongs to which walking TA). The **test data** does not provide this ground truth information and is merely **unlabeled accelerometer data** (three different sessions). Your task is to develop a method (using the training data) that would allow you to automatically determine **which of the sessions of the test set belongs to which walking TA**. You are free to use whatever method you deem appropriate. The deliverable will need to explain your method, provide the source code, and, of course, provide the answer to the question on which dataset was recorded by what TA.

Task 6: Contributions

At the end of your report, fill in the contributions section and detail what each member of your team (including yourself!) completed in the group activity.