Multidimensional Sensing Techniques

Assignment 1 – Step counter

You can collect up to 16 points in this assignment

Deadline: 29.09.2023 at 23:55

The aim of this assignment is to familiarize with reading data from sensors as well as visualizing and parsing data on a concrete example. In this assignment you will store data from an accelerometer using MATLAB on your smartphone. After reading the data from the sensor you will use MATLAB to shape the data structure such that it can be exploited by a python application.

You will need to implement a function to read and parse the data file obtained from MATLAB and then implement two different algorithms to count the number of physical steps recorded in your data.

All course work is individual. *Discussing the course tasks, assignments and specific issues with other course participants is allowed and even encouraged. However, you should be the only author of all the solutions you provide in this assignment. Team work, pair programming or copying solutions or program code from other persons is consider plagiarism and it will be handled following the Åbo Akademi protocol for such cases.*

MATLAB and MATLAB Mobile sensor setup

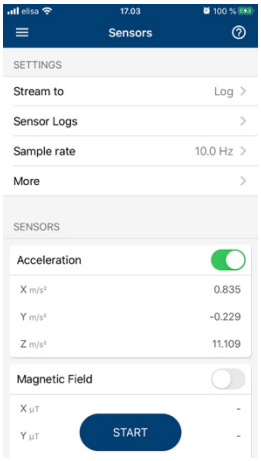
First you must download MATLAB and create an account using your university credentials. Do the following steps:

- Go on https://se.mathworks.com/academia/tah-portal/abo-akademi-40655967.html
- Click on *Sign in to get started* and log with your ÅA username and password
- Create a MathWorks Account if you do not have one yet
- You can access your MathWorks account anytime from the link http://www.mathworks.com/mwaccount
- Install MATLAB on your computer from your MathWorks account
    - Click on *Download Installer* from the *Install and activate tab* and install MATLAB (use your created account and select the ÅA license)

Open and sign into MATLAB on your computer (upper right corner) and enter the command: “license”. This will yield a **license key** which is needed to upgrade the mobile version of MATLAB to access the sensors.

Find MATLAB Mobile on the App Store or Google Play Store install it on your phone. Log in with your MathWorks account**.**

The sensors may now be activated and used on your phone. Enable the accelerometer, you do not have to change the settings (Acceleration in the SENSORS section). Your sensor logs will be saved and available from MATLAB Drive cloud (https://drive.matlab.com/files/).



You can now walk around with the sensor activated (after pressing start on the application). You can generate few logs of different types (for example slow walking, fast walking, jumping around, etc..). Remember to count how many steps you did for each log (for example write down the number of steps in the file name of your logs).

Data pre-processing

In MATLAB you may now use the command:

- load("name.mat")

This will load the data into memory. Accelerometer is now available for use; you will now index the data in the order it is given. This is because of a dependency in the lab1.py file. Enter the following command:

- temp = [[1: length(Acceleration.Timestamp)]', Acceleration{:, :}]

This command will generate a data matrix. *[1: length(Acceleration.Timestamp)]'* creates a column-vector which is concatenated with the accelerometer values in *Acceleration{:, :}*. The *{:, :}* suffix is added to convert the table *Acceleration* into a matrix of doubles.

Finally, execute the command:

- csvwrite("out.csv", temp)

This last command will write the double-matrix into a csv-file which may be used in by the provided *lab1.py* python script (note that used delimiter is ',')

Lab instructions

You need to have python install on your machine and have some basic knowledge of python programming.

Based on the visualization of the captured data you need to figure out an algorithm to count the number of steps taken during the measurements. Open the *stepcounter.py* python script and understand the source code and successive function calls starting from *main()*.

(2 point) Your very first task is to implement the function *read_data(filename)* which read the content of the MATLAB generated csv file and return a set of arrays. Open one .csv file with a text editor to see the structure of the input data.

(8 points) You will then implement a basic step counting algorithm in the *count_steps(timestamps, x_arr, y_arr, z_arr)* function. Here you are expected to provide a very simple solution using static threshold values to detect the number of steps.

(6 points) You will implement an enhanced step counting algorithm using dynamic threshold levels calculated over a fixed number of samples, and a defined time window used to discard possible invalid vibrations. You can get inspired by the two external resources provided bellow.

Compare the number of steps counted by your algorithms to the real number of steps done during the recording of the data. Run your algorithms over the few recorded data sets. How accurate are your algorithm?

External resources

Learn Python on w3schools.com:
https://www.w3schools.com/python/

Full-Featured Pedometer Design Realized with 3-Axis Digital Accelerometer
https://www.analog.com/media/en/analog-dialogue/volume-44/number-2/articles/pedometer-design-3-axis-digital-acceler.pdf

A Pedometer in the Real World
http://aosabook.org/en/500L/a-pedometer-in-the-real-world.html

Report instructions

You should upload all the implemented and used code for this lab on a GitHub Classroom repository. You should create your group repository from the following link:

https://classroom.github.com/a/E3bOxRid

If you do not have any GitHub account, please create one. The created repository will be private, and only you and teaching staff will have access to it. You will get full admin access on it. If you do not have any previous experience with git, please look for online tutorials and material introducing version control system in general and git in particular. Google will be you friend on this, but here are some pointers:

https://git-scm.com/book/en/v2/Getting-Started-Git-Basics

https://guides.github.com/introduction/git-handbook/

https://www.atlassian.com/git/tutorials

You should write a report documenting the work performed during this assignment:

The report document should contain:

- Your full name and student number at ÅA
- A direct link to your source code on GitHub.
- Each page of your submission should have a page number
- **Pay attention to the readability of your report!**

The expected size for this report is 3-5 pages of content. The report should contain information on:

- What you did
- How you did it
- Why you did it this way
- The results of your work
- How your solution could be improved/generalized

You are encouraged to provide pictures, graphs, screenshots, diagrams etc. Anything that help understanding your solutions should be included in your report. If you use content (pictures, graphs,

etc.) from other sources, remember to properly cite and provide a reference to the used external source(s).

At the end of the report you should also provide a reflection on what you learned during this exercise. This section could provide answers to the following questions:

- What did you learn?
- Did anything surprise you?
- Did you find anything challenging? Why?
- Did you find anything satisfying? Why?

Name you report file Assignment1_YourName.pdf and upload your report in PDF format on moodle before the deadline.