Accelerometer Time-Series Analysis Tutorial

You will develop skills in manipulating time series data using accelerometer signals as an example. The progression builds from simple visualization to smoothing, optimization, and elements of machine learning. There is sample data available for each experiment to get you started quickly.

Completion of each task, however, requires demonstration of end-to-end proficiency in working with a scientific workflow: The data you analyse will be the data that you obtain yourself using our collection of accelerometer sensors. At first you will follow a prescribed procedure and provided apparatus, but it will lead towards the creation of your own designs for apparatus and experimental procedures.

Table of Contents

[**1. Configuration for the Project**. 1](https://word-edit.officeapps.live.com/we/wordeditorframe.aspx?ui=en%2DUS&rs=en%2DUS&hid=gtxs10CGukKEs1zrKiT78A%2E0&WOPISrc=https%3A%2F%2Fwopi%2Eonedrive%2Ecom%2Fwopi%2Ffiles%2F403200EF655E37B1%211632&&&sc=host%3D%26qt%3DFolders&wdLOR=cC68866DE%2D4247%2D4A64%2DBA46%2DF04DE8F955C9&wdo=2&wde=docx&wdp=3&wdPid=5E5787A5&wdModeSwitchTime=1587396670343&wdPreviousSession=21ce5ab2-e6cd-4abe-8681-bc39fd082bf2&pdcn=pdc3cdc&wdOrigin=AppModeSwitch#_Toc31631312)

[1.1 Setting up GitHub￼1](https://word-edit.officeapps.live.com/we/wordeditorframe.aspx?ui=en%2DUS&rs=en%2DUS&hid=gtxs10CGukKEs1zrKiT78A%2E0&WOPISrc=https%3A%2F%2Fwopi%2Eonedrive%2Ecom%2Fwopi%2Ffiles%2F403200EF655E37B1%211632&&&sc=host%3D%26qt%3DFolders&wdLOR=cC68866DE%2D4247%2D4A64%2DBA46%2DF04DE8F955C9&wdo=2&wde=docx&wdp=3&wdPid=5E5787A5&wdModeSwitchTime=1587396670343&wdPreviousSession=21ce5ab2-e6cd-4abe-8681-bc39fd082bf2&pdcn=pdc3cdc&wdOrigin=AppModeSwitch#_Toc31631313)

[1.2 Preparing Python 3.7 and PyCharm IDE￼2](https://word-edit.officeapps.live.com/we/wordeditorframe.aspx?ui=en%2DUS&rs=en%2DUS&hid=gtxs10CGukKEs1zrKiT78A%2E0&WOPISrc=https%3A%2F%2Fwopi%2Eonedrive%2Ecom%2Fwopi%2Ffiles%2F403200EF655E37B1%211632&&&sc=host%3D%26qt%3DFolders&wdLOR=cC68866DE%2D4247%2D4A64%2DBA46%2DF04DE8F955C9&wdo=2&wde=docx&wdp=3&wdPid=5E5787A5&wdModeSwitchTime=1587396670343&wdPreviousSession=21ce5ab2-e6cd-4abe-8681-bc39fd082bf2&pdcn=pdc3cdc&wdOrigin=AppModeSwitch#_Toc31631314)

[**2. Basic Data Manipulation￼2**](https://word-edit.officeapps.live.com/we/wordeditorframe.aspx?ui=en%2DUS&rs=en%2DUS&hid=gtxs10CGukKEs1zrKiT78A%2E0&WOPISrc=https%3A%2F%2Fwopi%2Eonedrive%2Ecom%2Fwopi%2Ffiles%2F403200EF655E37B1%211632&&&sc=host%3D%26qt%3DFolders&wdLOR=cC68866DE%2D4247%2D4A64%2DBA46%2DF04DE8F955C9&wdo=2&wde=docx&wdp=3&wdPid=5E5787A5&wdModeSwitchTime=1587396670343&wdPreviousSession=21ce5ab2-e6cd-4abe-8681-bc39fd082bf2&pdcn=pdc3cdc&wdOrigin=AppModeSwitch#_Toc31631315)

[2.1 Directory Structure and Naming￼2](https://word-edit.officeapps.live.com/we/wordeditorframe.aspx?ui=en%2DUS&rs=en%2DUS&hid=gtxs10CGukKEs1zrKiT78A%2E0&WOPISrc=https%3A%2F%2Fwopi%2Eonedrive%2Ecom%2Fwopi%2Ffiles%2F403200EF655E37B1%211632&&&sc=host%3D%26qt%3DFolders&wdLOR=cC68866DE%2D4247%2D4A64%2DBA46%2DF04DE8F955C9&wdo=2&wde=docx&wdp=3&wdPid=5E5787A5&wdModeSwitchTime=1587396670343&wdPreviousSession=21ce5ab2-e6cd-4abe-8681-bc39fd082bf2&pdcn=pdc3cdc&wdOrigin=AppModeSwitch#_Toc31631316)

[2.2 Graphing Test Data￼3](https://word-edit.officeapps.live.com/we/wordeditorframe.aspx?ui=en%2DUS&rs=en%2DUS&hid=gtxs10CGukKEs1zrKiT78A%2E0&WOPISrc=https%3A%2F%2Fwopi%2Eonedrive%2Ecom%2Fwopi%2Ffiles%2F403200EF655E37B1%211632&&&sc=host%3D%26qt%3DFolders&wdLOR=cC68866DE%2D4247%2D4A64%2DBA46%2DF04DE8F955C9&wdo=2&wde=docx&wdp=3&wdPid=5E5787A5&wdModeSwitchTime=1587396670343&wdPreviousSession=21ce5ab2-e6cd-4abe-8681-bc39fd082bf2&pdcn=pdc3cdc&wdOrigin=AppModeSwitch#_Toc31631317)

[2.3 Graphing your own data￼3](https://word-edit.officeapps.live.com/we/wordeditorframe.aspx?ui=en%2DUS&rs=en%2DUS&hid=gtxs10CGukKEs1zrKiT78A%2E0&WOPISrc=https%3A%2F%2Fwopi%2Eonedrive%2Ecom%2Fwopi%2Ffiles%2F403200EF655E37B1%211632&&&sc=host%3D%26qt%3DFolders&wdLOR=cC68866DE%2D4247%2D4A64%2DBA46%2DF04DE8F955C9&wdo=2&wde=docx&wdp=3&wdPid=5E5787A5&wdModeSwitchTime=1587396670343&wdPreviousSession=21ce5ab2-e6cd-4abe-8681-bc39fd082bf2&pdcn=pdc3cdc&wdOrigin=AppModeSwitch#_Toc31631318)

# 1. Configuration for the Project

## 1.1 Setting up GitHub

To gain access to the project files, you will need a GitHub account. GitHub is a version control service that allows for collaborators to work on the same files from different locations. It allows for organization of the files, and the changes made to these files.

In order to create a GitHub account if you don’t already have one, please follow this link: <https://github.com/join>.

Once you have an account, download GitHub Desktop from this link: [desktop.github.com](https://desktop.github.com/).

Next, please follow these instructions to access the project from your machine.

1. Open GitHub Desktop.
2. Go to File > Clone Repository.
3. Select the URL tab.
4. Paste this link into the URL dialog box: “<https://github.com/larnder/2019_06_AccelerationCamp>”.
5. Choose the file location you would like this repository to be cloned into.
6. Click “Clone”.

## 1.2 Preparing Python 3.7 and PyCharm IDE

Python is the main programming language used for this project and will be necessary to access and edit the files. We use Python 3.7 because that is the latest version that works with TensorFlow, which is another crucial tool for the project that will be discussed later. You will also need an environment in which to edit and view Python code. We recommend PyCharm, and the steps to download this environment are provided in the following procedure (Step 5).

Follow these steps to prepare Python 3.7:

1. Download and install the latest version of Python 3.7 from the following link: <https://www.python.org/downloads/>.
2. Open Command Prompt.
3. Navigate to the folder where you installed Python. i.e. “cd C:/User/…”
4. To install necessary libraries for Python, type “python -m pip install matplotlib numpy tensorflow”
5. Download and install the Community version of PyCharm from the following link: <https://www.jetbrains.com/pycharm/download/#section=windows>.

# 2. Basic Data Manipulation

## 2.1 Directory Structure and Naming

When saving a data set into the GitHub repository, please use the following convention:

* Directory Structure:￼
  + data / Dataset # / run# / files
* File Structure:
  + Name.type.device.csv
    - Name: File identifier
    - Type:
      * “accel” for acceleration data
      * “omega” for angular velocity data
    - Device (Supported devices):
      * “x2” for GCDC X2-2 accelerometer sensor
      * “x16” for GCDC X6-1a accelerometer sensor
      * “pocket” for PocketLab Voyageur accelerometer sensor

## 2.2 Graphing Test Data

In this exercise, you will be using *TestLoadPlot.py* to graph a data set. This data set is called “RawData\_SamsingJS.csv” and it can be found in tutorials/data. This data set was captured using the GCDC X2-2 accelerometer sensor. The sensor was tapped twice on the table and wiggled horizontally, and then left immobile. Once the graph is generated, you should see two spikes on the vertical axis, then a smoother variation on another axis, and finally, zero variations on all 3 axes.

To generate the graph:

1. In the folder “src”, run the file *TestLoadPlot.py*.
2. In the file explorer popup, select the folder “tutorials/data”.

If the result is as follows, you are set up correctly for the project.

## 2.3 Graphing your own data

This time, you will record your own data to graph.

Tasks:

1. Acquire a PocketLab Voyageur accelerometer.
   1. Connect to one of your devices using the instruction manual.
2. Record accelerometer data and save it according to the naming convention described in 2.1.
3. Graph your data using *TestLoadPlot.py*.

# 3. Position via Regression

This project is dedicated to finding the position of an accelerometer on a rotating frame by using the Linear Regression method. This is useful, because manufacturers usually do not specify the exact location of an accelerometer sensor in a device. This method can provide an estimate of where the sensor will be, without having to physically open the device.

## 3.1 Required Materials

1. Record player (must have 78 RPM option)
2. SpinFrame (circular 3d printed base built for this project)
3. Accelerometer (recommended: PocketLab Voyageur)
4. Ruler/measuring device
5. Tape

## 3.2 Procedure

1. Ensure that the power switch of the record player is in the OFF position then plug it in.
2. Set the rotational speed of the record player to 78 RPM.
3. Place the SpinFrame on the shaft of the record player.
4. Place a loop of tape underneath the accelerometer.
5. Place the accelerometer in the top left corner of the SpinFrame’s borders, taking note of the direction of the accelerometer’s axes.
6. Begin recording data on the accelerometer.
7. Turn on the record player.
8. Once the rotational velocity of the SpinFrame is constant, wait approximately 10 seconds, then turn off the record player and the accelerometer.
9. Remove the accelerometer from the SpinFrame and save the data according to the naming convention described in section 2.1.
10. Move the accelerometer 1cm to the right and repeat steps 5-9 until the accelerometer reaches the end of the SpinFrame.
11. Place the accelerometer in its starting position (at the top left of the SpinFrame).
12. Repeat step 10 but moving downwards instead of rightwards.

## 3.3 Processing the Data

In each acceleration file, remove any data that was recorded outside of the constant rotational velocity period.

1. Add each run to an Excel spreadsheet.
2. For each run,
   1. Calculate the average magnitude in acceleration in the x-y plane.
   2. Using the average acceleration and the angular velocity, calculate the radius of rotation.
   3. Calculate the sensor’s angular position relative to the center of rotation using the average accelerations.
3. Create a Calculations tab inside of the Excel spreadsheet.
4. In this Calculations tab, create one section that graphs ax vs rx, and one that graphs ay vs ry.
5. Using the equation of a linear line of best fit, find the y-intercept of both graphs.

The y-intercept on each graph is the

# 4. Handwriting Recognition

This project is dedicated to writing letters and numbers using accelerometers instead of writing utensils.

## 4.1 Required Materials

1. Accelerometer

## 4.2 Training Models to Recognize the English Alphabet

This section describes how to train machine learning software to recognize handwritten letters.

1. In the **dev** branch, go to src/modules/Alphabet and open *Train.py* in an editor.
2. In the MLPClassifier function and set the values of the arguments as desired. If you want to learn more, visit [https://scikit-learn.org/stable/modules/generated/sklearn.neural\_network.MLPClassifier.html](https://youtu.be/dQw4w9WgXcQ).
   1. “hidden\_layer\_sizes”: Controls the number of layers and their sizes.
   2. “max\_iter”: Sets the maximum on the number of iterations of training that can occur.
   3. “alpha”: Penalty parameter/regularization term.
   4. “solver”: Determines solving method used by the program.
   5. “verbose”: Determines how often to print loss and accuracy during training and testing.
   6. “tol”: Tolerance of training. If the loss does not decrease by this amount over 10 runs, then training will stop.
   7. “random\_state”: Indicates number of random states.
   8. “learning\_rate\_init”: The initial learning rate used. Controls the step size in updating the weights.

## 4.3 Procedure

1. Prepare a free horizontal flat surface.
2. Place the accelerometer on the surface such that the positive y axis is pointing forward, and the positive x axis is pointing rightward. **Make sure that this orientation is maintained throughout the entire procedure.**
3. Begin recording.
4. Draw your desired figure while maintaining contact with the surface throughout.
5. Stop recording.
6. Save the accelerometer to the desired folder using the naming convention described in section 2.1.
7. To graph the figure, go to src/modules and run *AlphabetPlotter.py*.
   1. Follow the instructions in the dialog boxes and select the saved dataset.
8. To access a model prediction of the drawing, go to src/modules/Alphabet.
   1. If you are running Windows, run *Predict.py* and follow the instructions in the dialog boxes and select the saved dataset.
   2. If you are running in WSL, run *run.sh* and follow the instructions in the dialog boxes and select the saved dataset.

# 5.