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](#_Toc31631312)

[1.1 Setting up GitHub 1](#_Toc31631313)

[1.2 Preparing Python 3.7 and PyCharm IDE 2](#_Toc31631314)

[**2. Basic Data Manipulation** 2](#_Toc31631315)

[2.1 Directory Structure and Naming 2](#_Toc31631316)

[2.2 Graphing Test Data 3](#_Toc31631317)

[2.3 Graphing your own data 3](#_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).

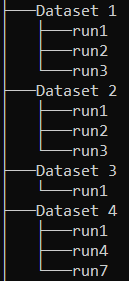
Follow these steps to prepare Python 3.7:

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:

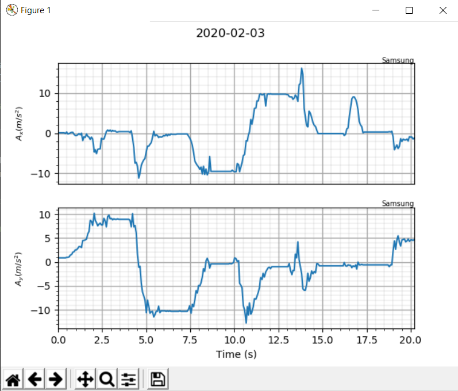
Example of Directory Structure

* + - * “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”.
3. 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”.