Assignment 5: Event Finding

First steps

Understand the code in section 6.1 of the book *Hands-on Signal Analysis with Python*. Make sure that you understand:

- How to create a logical vector by thresholding
- How to use logical indexing
- How to combine logical vectors with bitwise logical operators
- Why it can be useful to take the derivative before finding features
- How to use `plt.subplots`

1 Simulate experimental data

Generate a vector which looks like the data in Figure 6.3 in the book, and save it to the file `HorPos.txt` (with no header). If you do that correctly, you should be able to run the file `event_detection.py`, generating a figure similar to Figure 6.3 in the book.

For `event_detection.py` to run smoothly, your data vector should have the following properties:

- 500 points of random noise about 0 (a vector of 500 random data, normally distributed about 0 with a standard deviation of 1, can be generated with `np.random.randn(500)`),
- then 500 points of values with random noise with an average value of -15,
- then the same thing for average values up to +15, in steps of 5.
- And at the end, again 500 points of random noise about 0.

2 EMG Activity

2.1 Background

EMG-data are some of the most common signals in movement analysis. But sometimes the data analysis is not that simple. For example, data can be superposed by spurious drifts. And short drops in EMG activity can obscure extended periods of muscle contractions.

The data in `Shimmer3_EMG_Calibrated.csv` have been taken from https://www.shimmersensing.com/support/sample-data/, where also the data description is given in detail. The first column of EMG-data describes the muscle activity in the forearm, the second one the activity in the biceps. Sample rate is 512 Hz, and the data are in mV.

2.2 Problem specification

Write a function that does the following:

- Import the EMG data from the data file `Shimmer3_EMG_Calibrated.csv`. Thereby the command `pd.read_csv` can be used with the parameter `delim_whitespace=True`, to ensure that any mixture of white-spaces is taken as a single separator.

  Select the column corresponding to the EMG of the forearm.
- Remove the offset of the EMG-recording with a Butterworth highpass filter.
- Rectify the data, and smooth them with a Savitzky-Golay filter to produce a rough envelope of the signal.
- Interactively select a threshold, and use this to find the start- and end-points of muscle activity, using the `numpy` command `where`. Watch out that activities that last less than 0.5 sec are probably measurement artefacts!
- Eliminate further artefacts, by cutting away transients at the beginning of the file.
- Calculate and display the mean contraction time.

3 Gait Events

Heel Strike and Toe Off are two important events within the human gait, and give the examined an insight in to the timing during walking. Their occurrence can be measured using a force plate.

- Load `GroundReactionForce.mat`, which can be found in the directory `Data`.
- Plot the z-axis of the data
- Choose a threshold to detect when the foot is on the force plate
- Set a green marker when an heel strike occurs
- Set a red marker when an toe off occurs

4 R-peaks

R-peaks are the most prominent events in an ECG signal, therefore they can be used to calculate the heart rate.

- Load `ecg-hfn.mat` (fs=1000Hz) , which can be found in the directory `Data`.
- Plot the z-axis of the data
- Find R-peaks automatically and highlight them in the plot with a marker.
  (Hint: find zero-crossings of the 1^{st} derivative, by checking where the product of two subsequent velocities is negative, and the ECG is simultaneously higher than a selected threshold.)
- Calculate the mean and standard deviation of the heart bpm (beats-per-minute). Be careful to note that the heart rate in `[Hz]` is giving the beats per second!!