**Boston University**

**Electrical & Computer Engineering**

**EC463 Senior Design Project**

Prototype Testing Plan

**Neuron Spike Identification with Machine Learning**

Chen Yang, PhD

Yang Research Laboratory

cheyang@bu.edu

by

Team 2

Spike Sorters

Team Members

Victoria Carlsten [carlsten@bu.edu](mailto:email1@bu.edu)

Hao Chen [ha0chen@bu.edu](mailto:email1@bu.edu)

Claire Cropper [ccropper@bu.edu](mailto:email1@bu.edu)

Shi Gu [bengushi@bu.edu](mailto:email1@bu.edu)

**Required Materials:**

Hardware:

* Personal Laptop

Software:

* Python, 2 scripts
* 2 data files
  + *DataImport.py*
    - Load recording data
    - Detect spikes
  + *spikesortingVTJason.py*
    - Load recording data
    - Detect spikes
    - Delete unnecessary data

**Setup:**

The setup only consists of one part: using our personal laptops to generate three graphs via two Python scripts. Firstly, running the DataImport.py using the data file ‘30 min\_0001.abf’ will produce the graph that depicts a bandpass filter. Then, running the spikesortingVTJason.py will produce 2 graphs. One is to generate the graph of Raw, LFP and Spike data after the raw data has been processed by the bandpass filters. The second one is to generate the graph of spikes that have been deleted unnecessary data (fake oscillation spikes, 3 ms spikes).

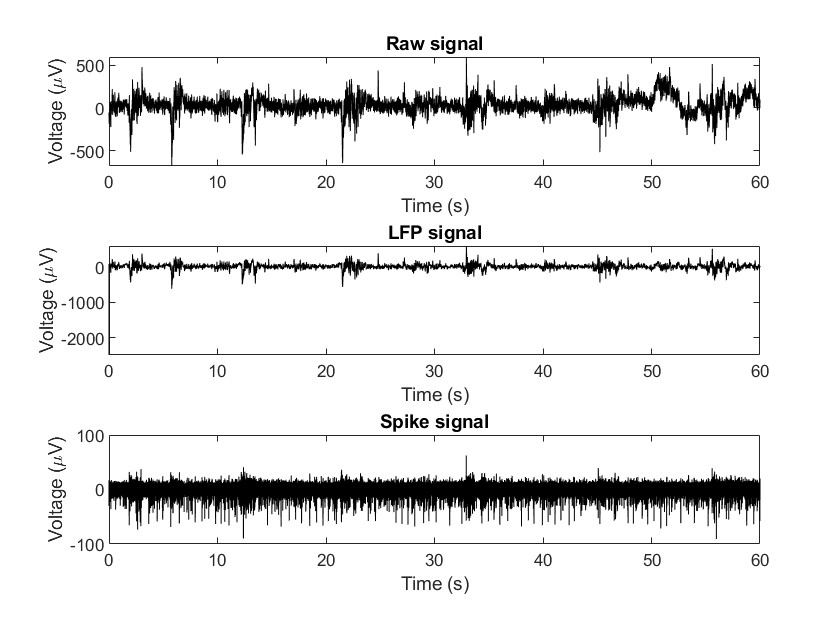
**Pre-testing Setup Procedure:**

DataImport file:

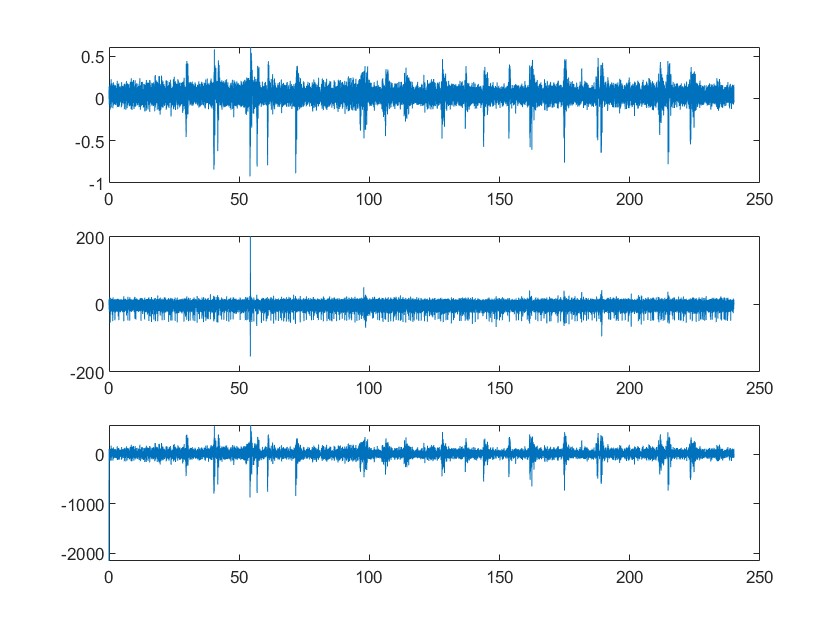
1. Make sure user is in the correct working directory
2. Make sure ‘30 min\_0001.abf’ is in the working directory
3. Run the python script, *DataImport.py*

spikesortingVTJason file:

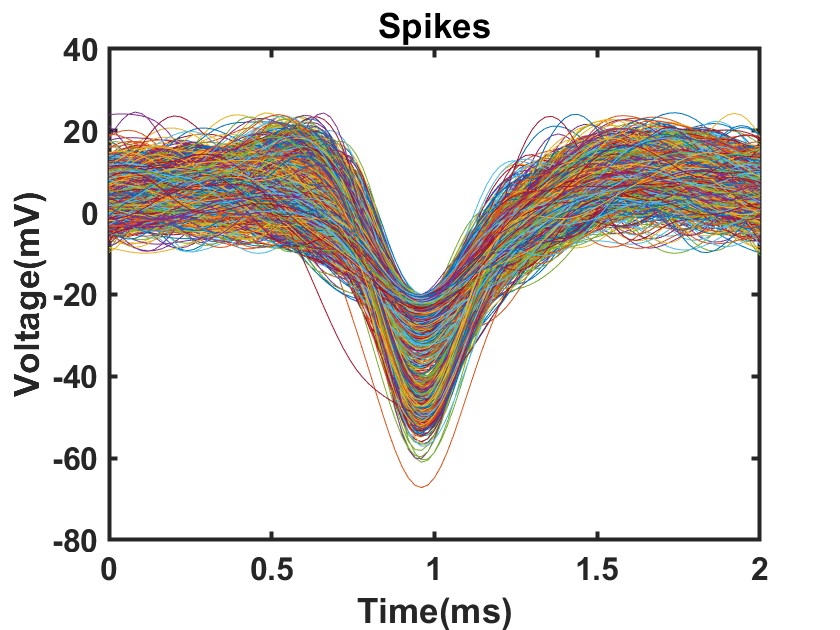
1. Make sure user is in the correct working directory
2. Make sure ‘10 min recording1.mat’ is in the working directory
3. Run lines 1 - 118 of the python script, *spikesortingVTJason.py*



*Figure 1. Bandpass filter for picking up the LFP and Spike signal in DataImport.m*

****

*Figure 2. Bandpass filter for Spikes and LFP in spikesortingVTJason.m*



*Figure 3. Spikes graph in spikesortingVTJason.m*

**Testing Procedure:**

1. Run DataImport.py
2. The computer will load a plot identical to Figure 1.
3. Run spikesortingVTJason.py
4. The computer will load a plot identical to Figures 2 and 3.

**Measurable Criteria:**

1. DataImport.py should successfully accept and load the recording data from the file ‘30 min\_0001.abf’, which is an .abf file of pre-recorded 30 minutes of Electrophysiological Data.
2. Spikes should be detected according to the threshold value and output a plot similar to Figure 1.
3. spikesortingVTJason.py (lines 1 - 118) should correctly load the recording data from the ‘10 min recording1.mat’, which is a mat file of pre-recorded 10 minutes of Electrophysiological Data, and then select the part of interest.
4. Spikes should be detected according to the threshold value and output a plot similar to Figure 2.
5. First step of PCA analysis of the spike array should output a plot similar to Figure 3.

**Score Sheet**

| **Object** | **Category** | **Correct? (Y/N)** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| **Result →** | | |