
EEM255 – Project 2017

EMG Decomposition

Zaurbek Tsorojev

12/12/2017

INTRODUCTION

- ❖ The goal of phase II is to realize an EMG decomposition of a given signal. We want to find the number of neurons responsible for the EMG signal.

This presentation contains the results and figure of 8 building sections of the decomposition, for 3 test databases. It ends with a discussion of the results for each section, and gives different methods to explore that could potentially improve the decomposition.

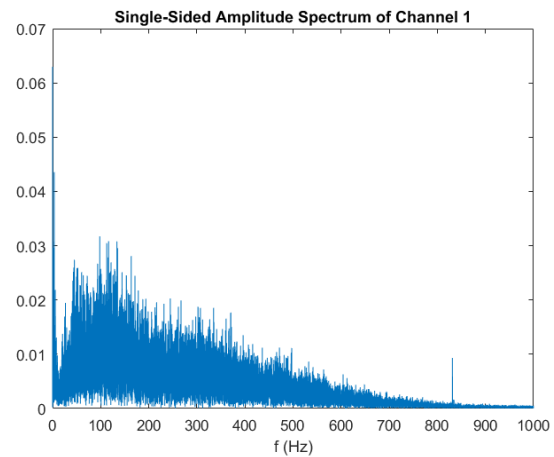
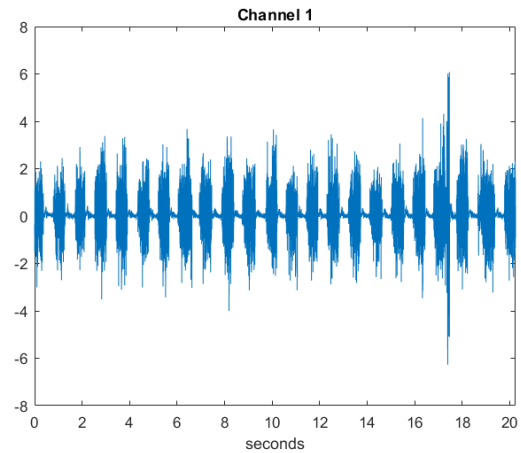
Note:

1. *EMG_example_2_fs_2k*: I used **channel 1**
2. *EMG_example_20s_2000Hz*: I used **channel 1**
3. *EMG_example_1_90s_fs_2k*: I used **channel 2**

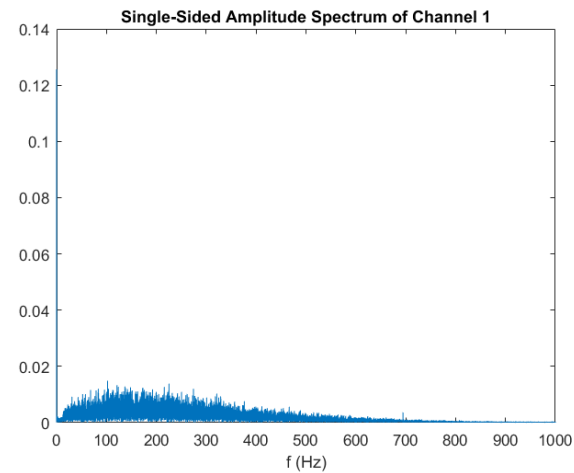
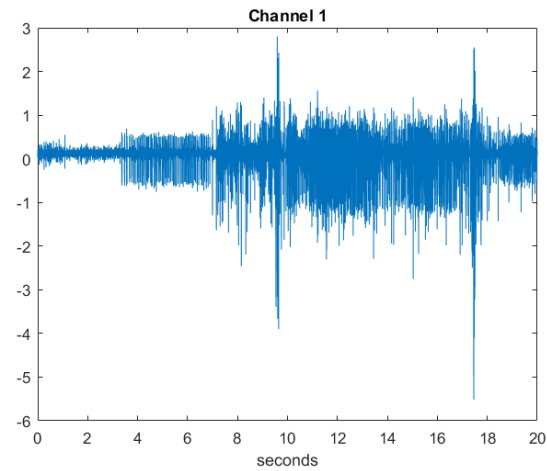
The columns of each step are presented in this order of databases.

INPUT FILE

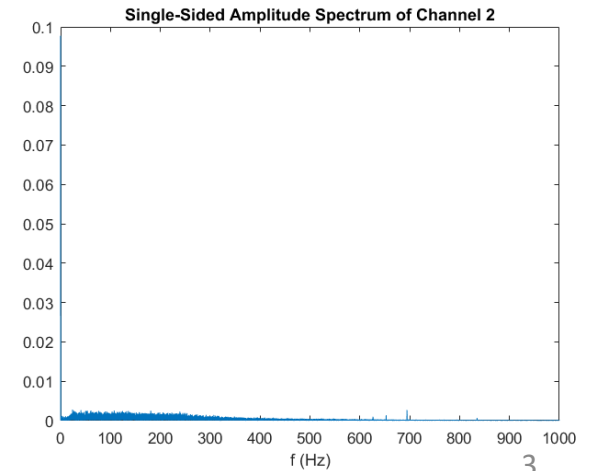
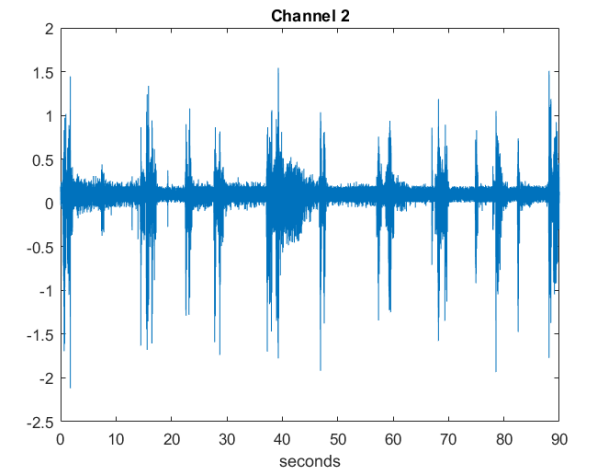
EMG_example_2_fs_2k



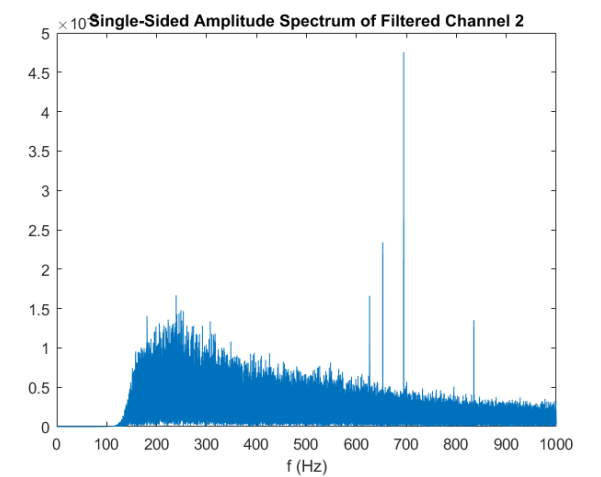
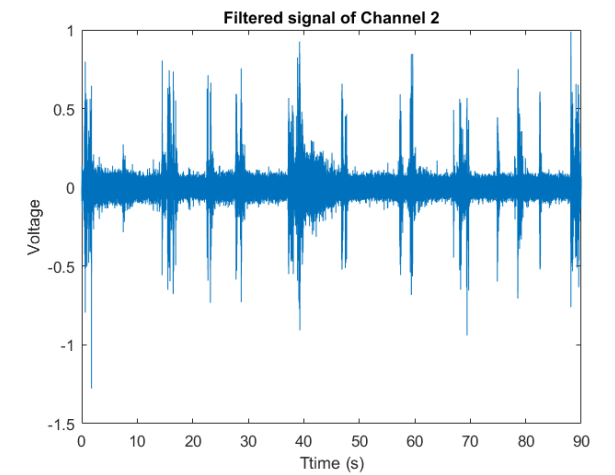
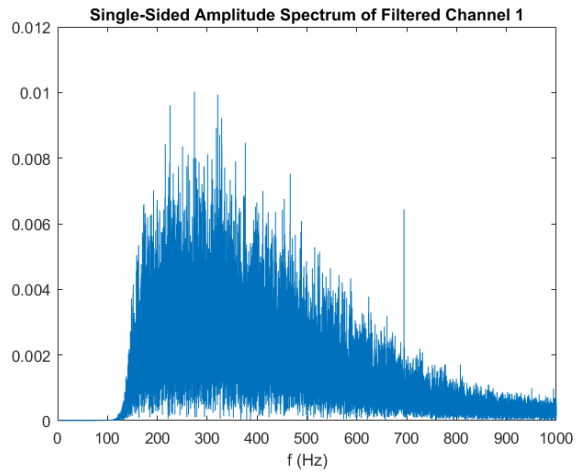
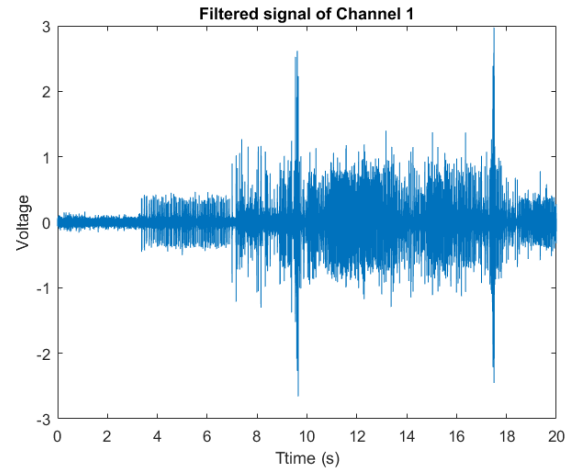
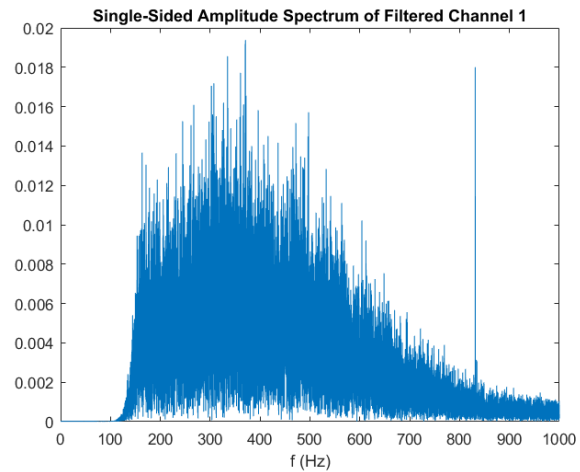
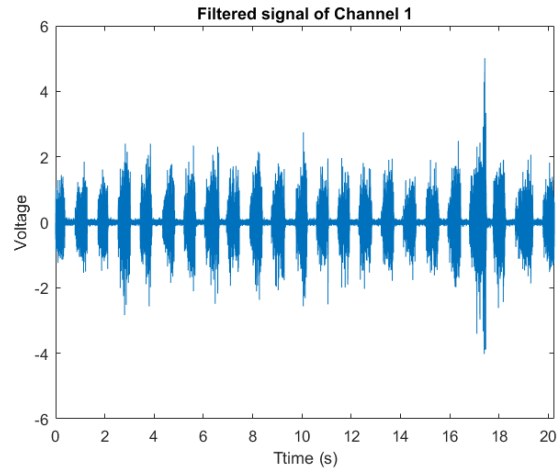
EMG_example_20s_2000Hz



EMG_example_1_90s_fs_2k



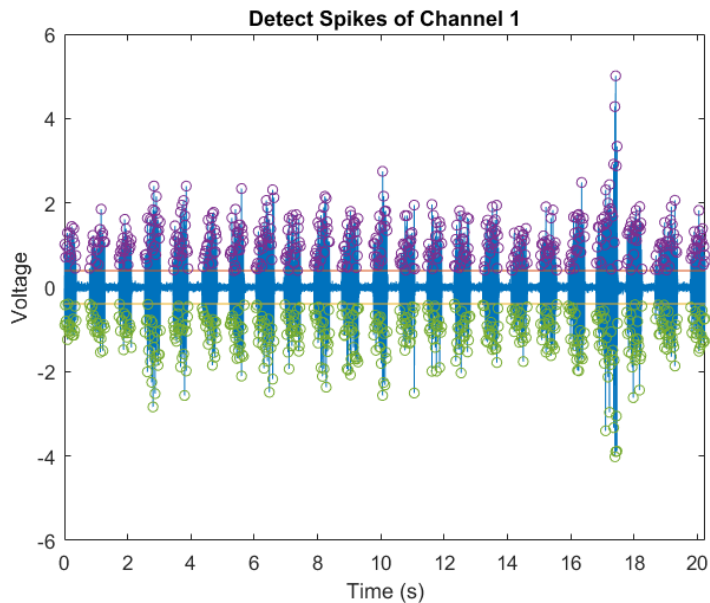
FILTER SIGNAL



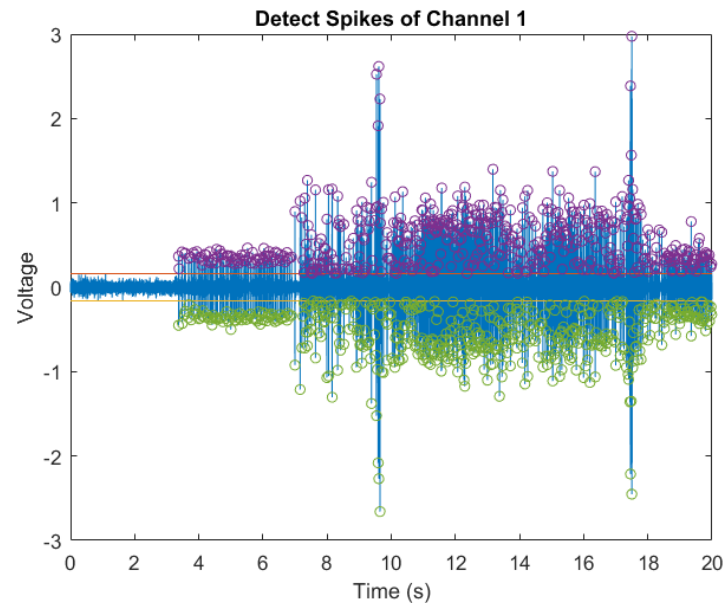
DETECT SPIKES

Find all the peaks above the threshold.

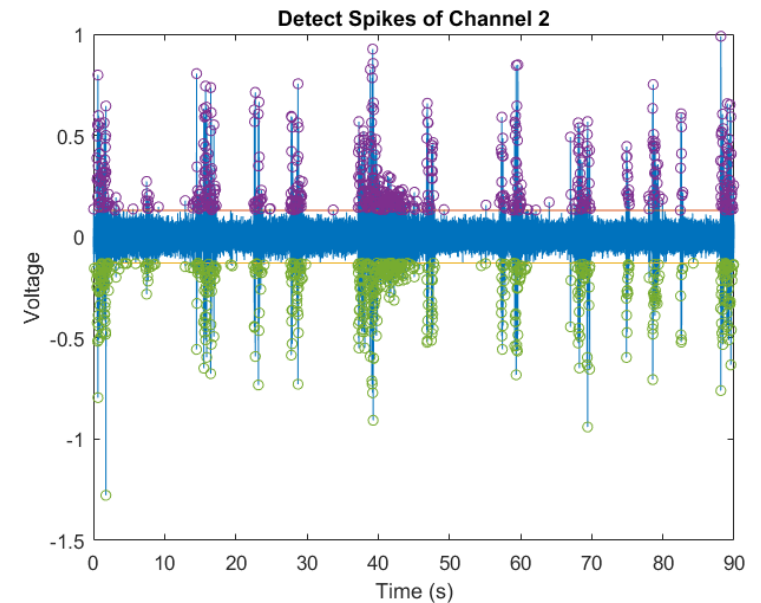
I noticed that using a slightly higher threshold than the $3 \times \text{RMS}$ gives a better result for clustering.



Threshold = $4 \times \text{RMS}$ of
baseline noise



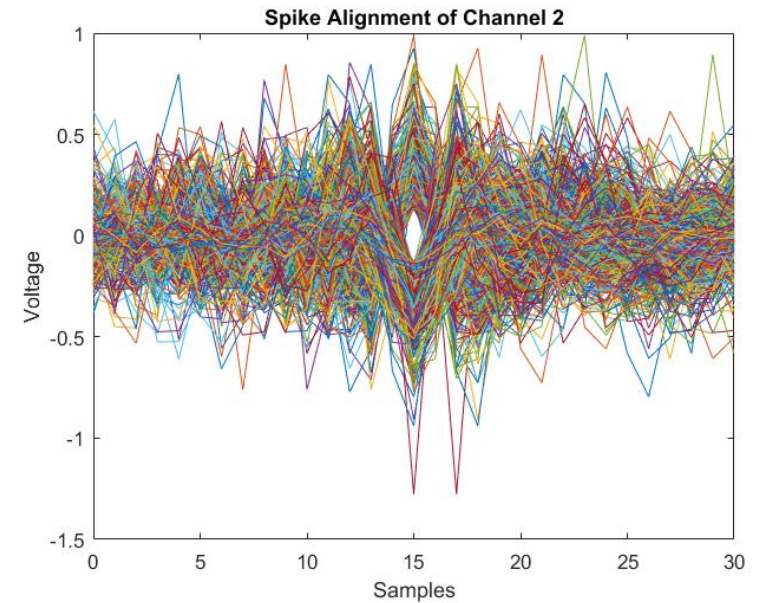
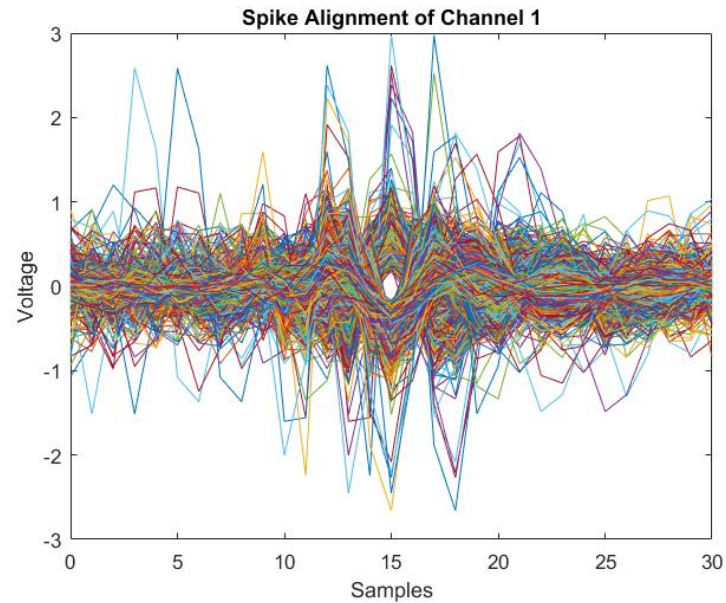
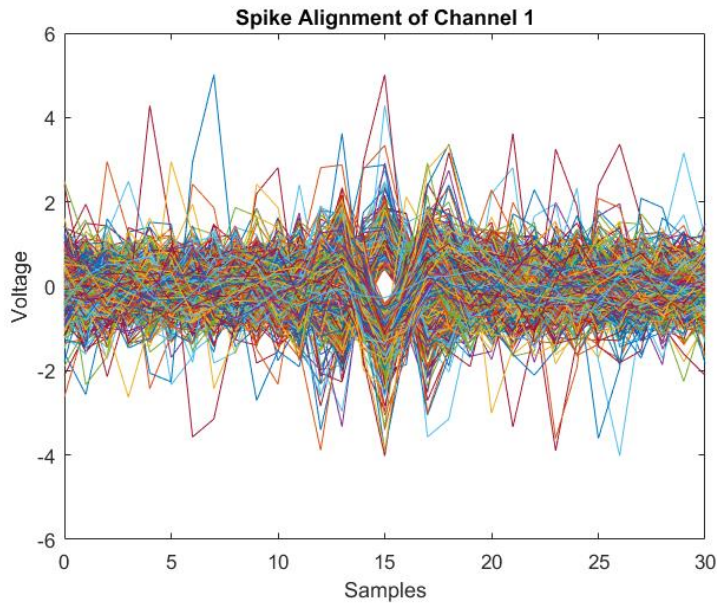
Threshold = $4 \times \text{RMS}$ of
baseline noise



Threshold = $5 \times \text{RMS}$ of
baseline noise

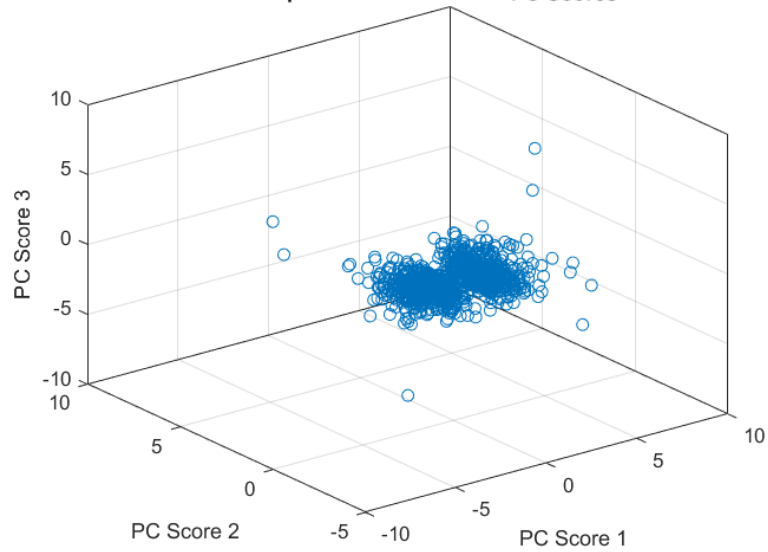
ALIGN SPIKES

Plot all the spikes in the same time window.

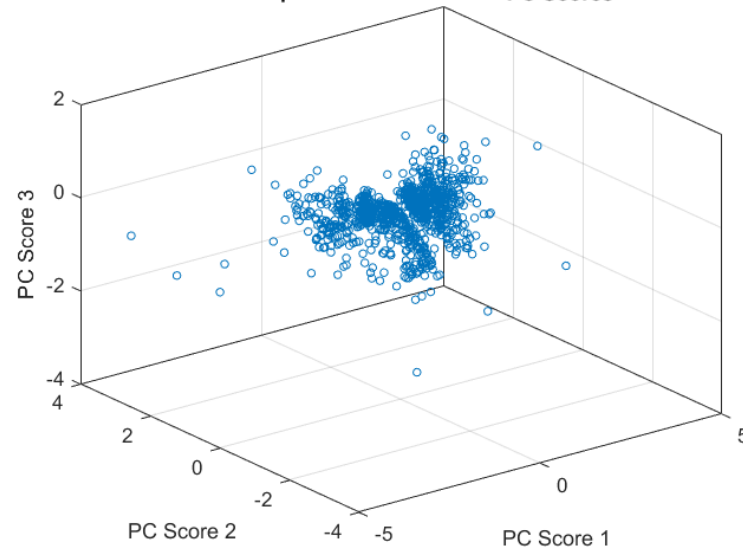


EXTRACT FEATURES

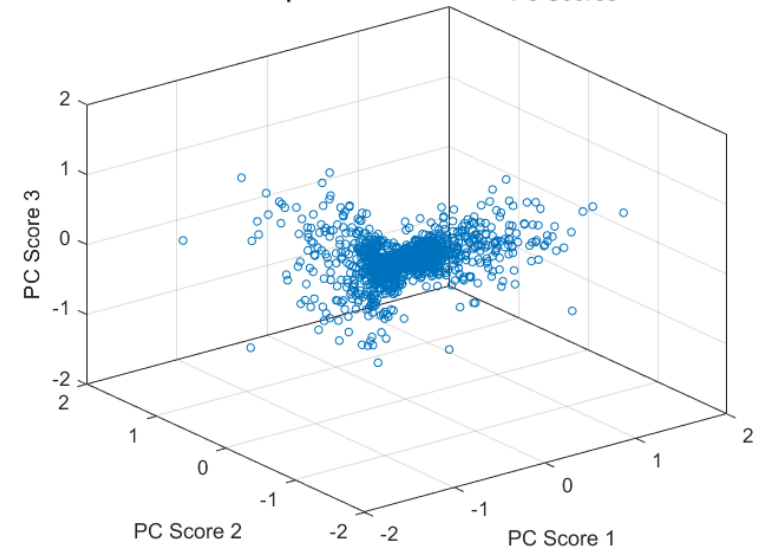
Scatterplot of the First Three PC Scores



Scatterplot of the First Three PC Scores

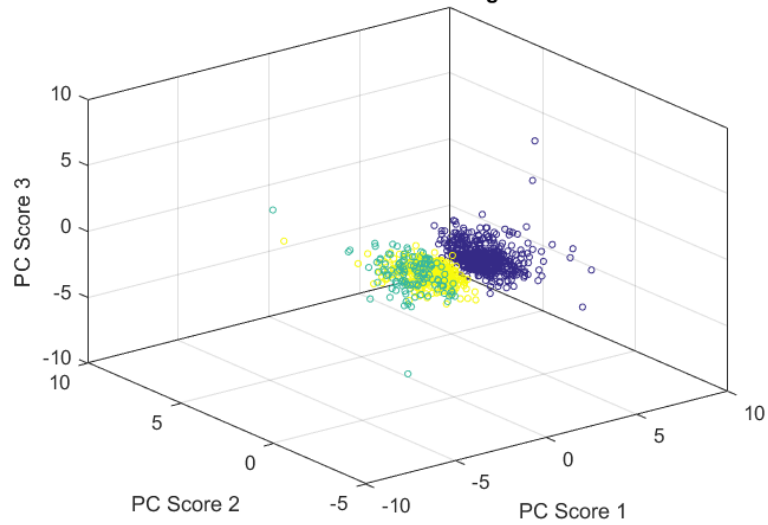


Scatterplot of the First Three PC Scores

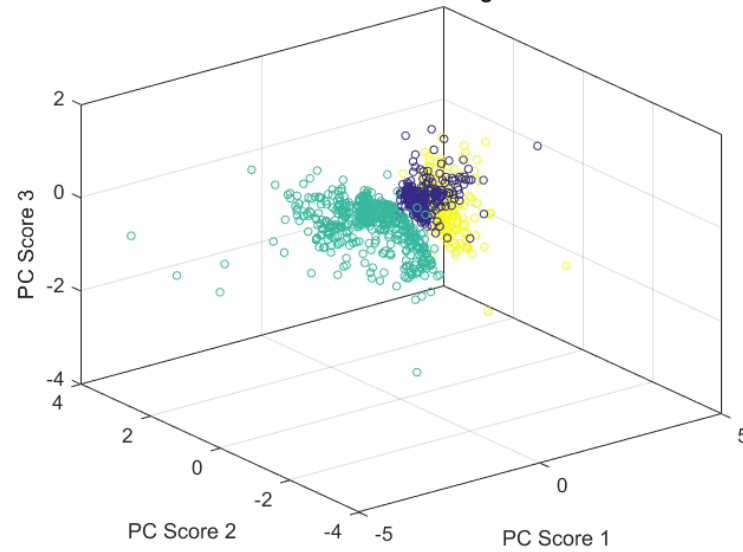


CLUSTER SPIKES

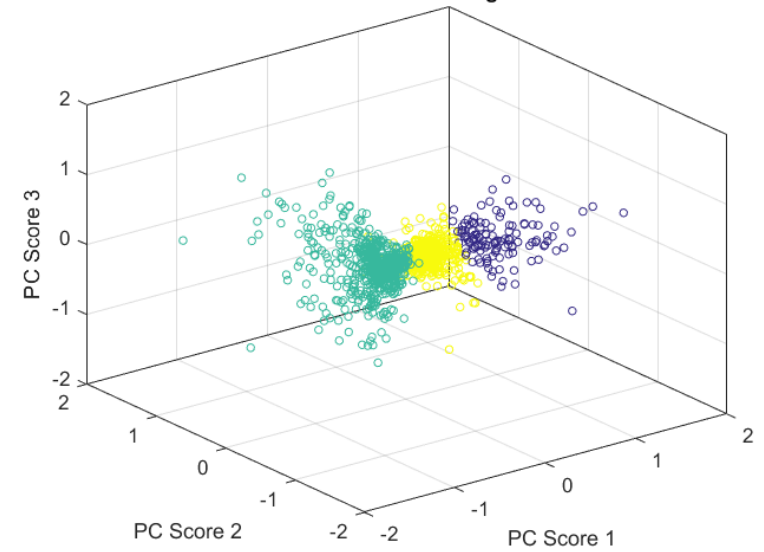
Kmean Clustering



Kmean Clustering



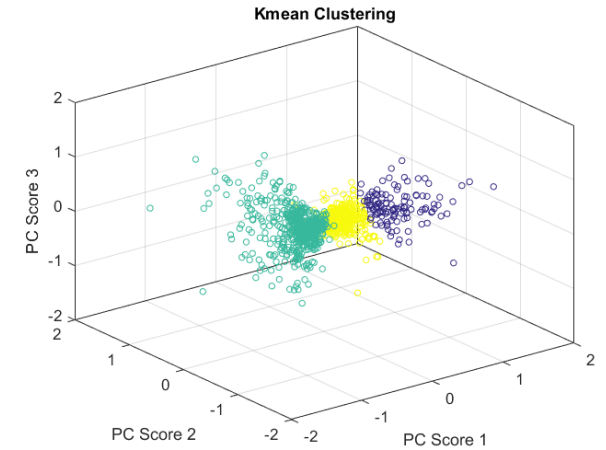
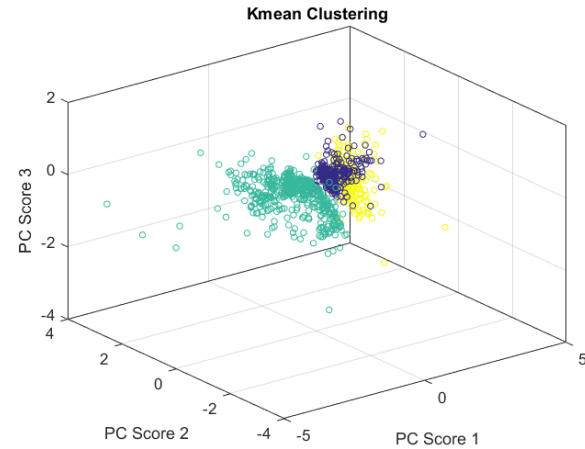
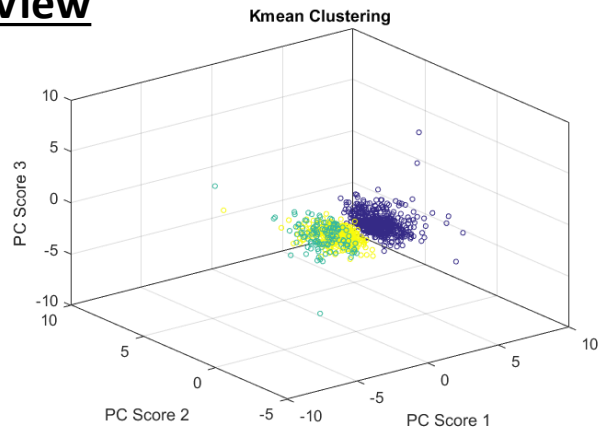
Kmean Clustering



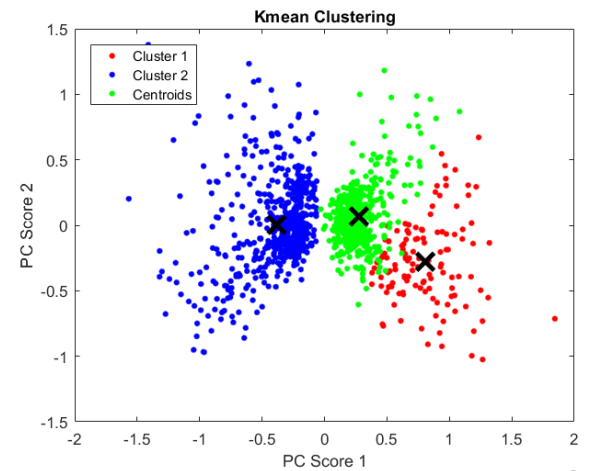
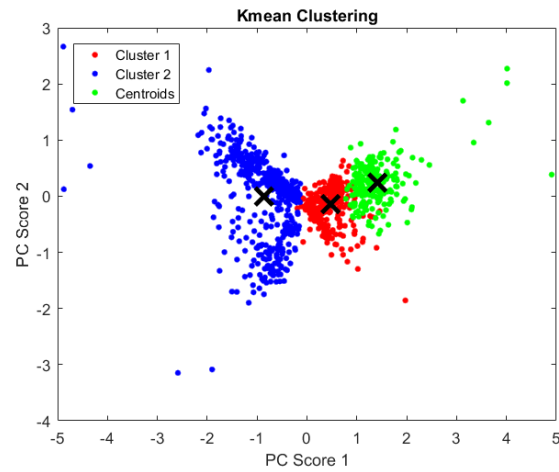
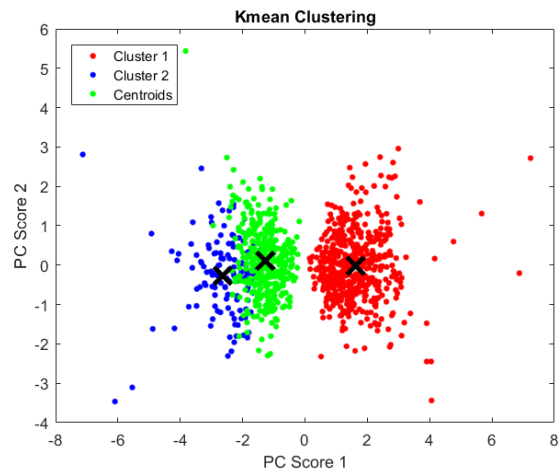
CLUSTER SPIKES

Cyan = Blue
Yellow = Green
Magenta = Red

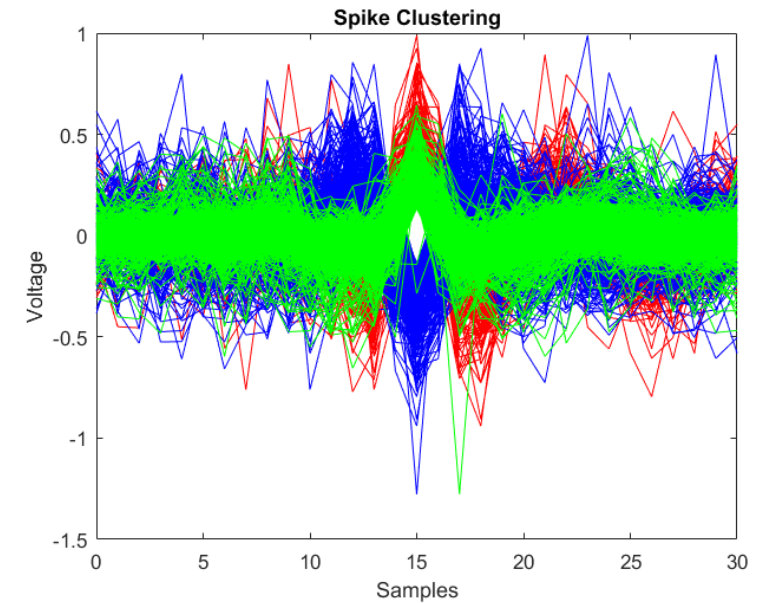
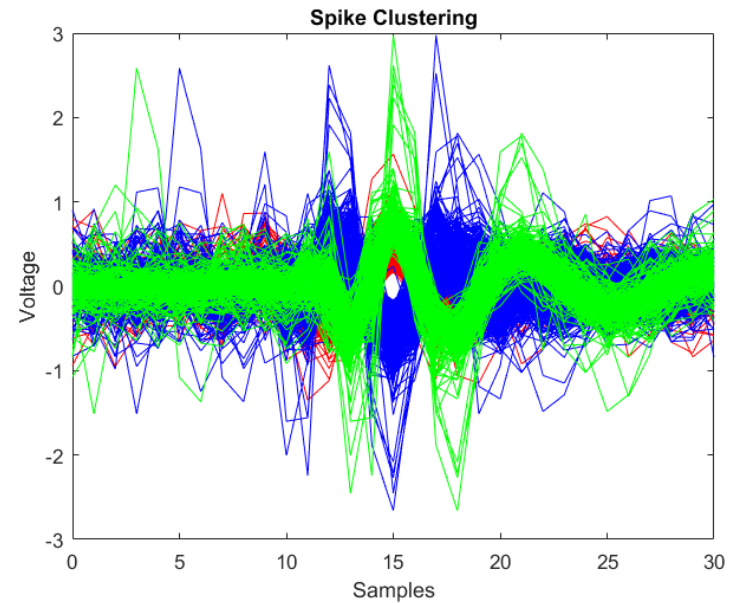
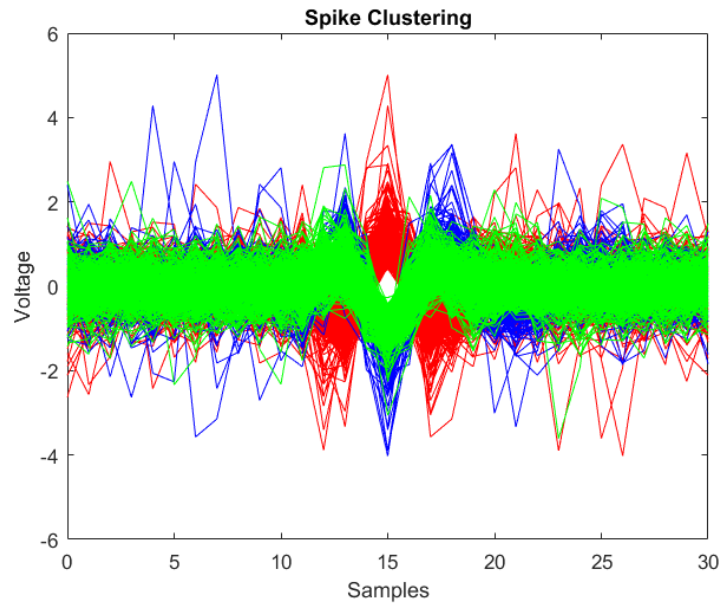
3D View



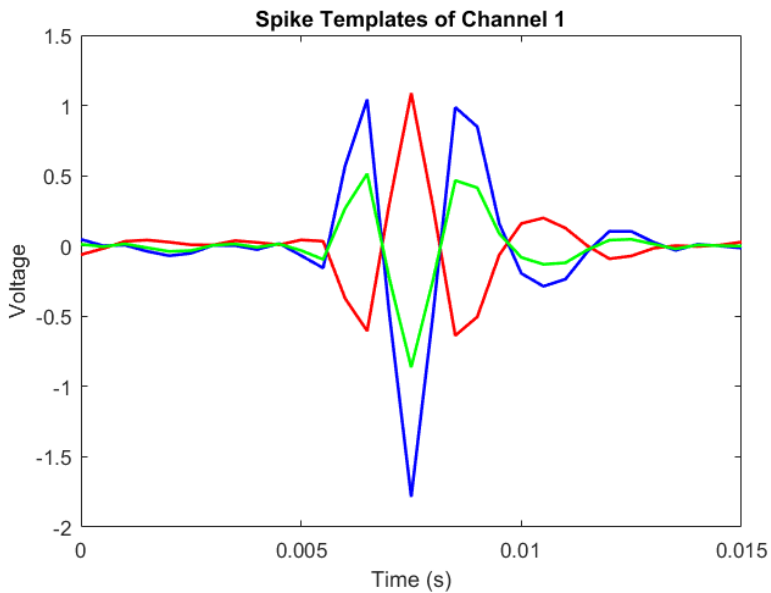
2D View



CLASSIFY SPIKES



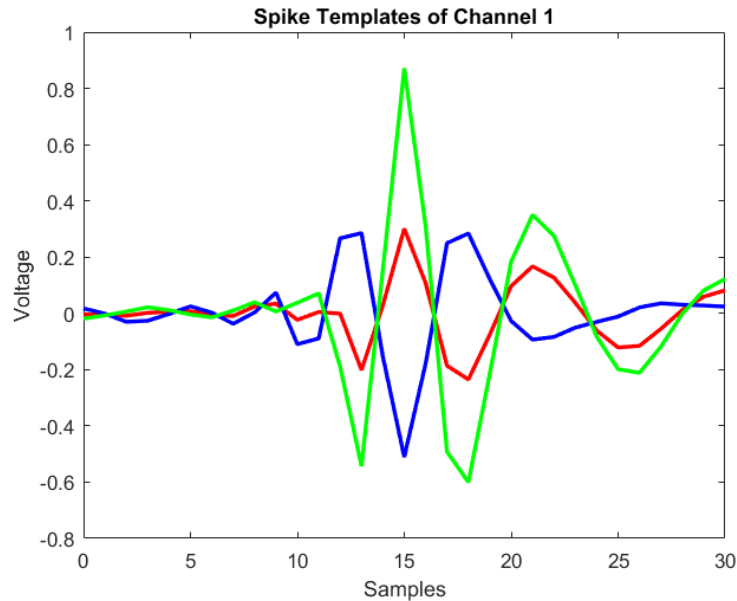
ANALYSIS (1)



3 clusters, but spikes **red** and **green** are \approx same (just opposite amplitude)



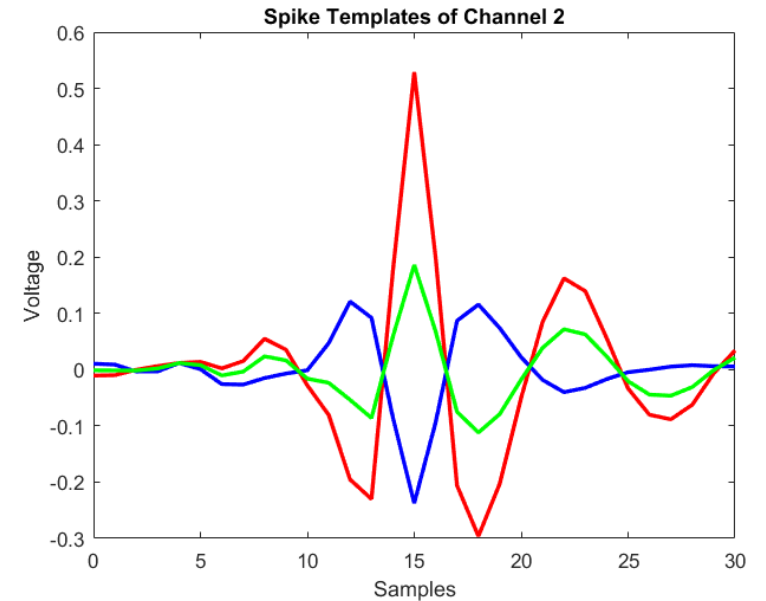
2 neurons



3 clusters, and all the spikes are different



3 neurons

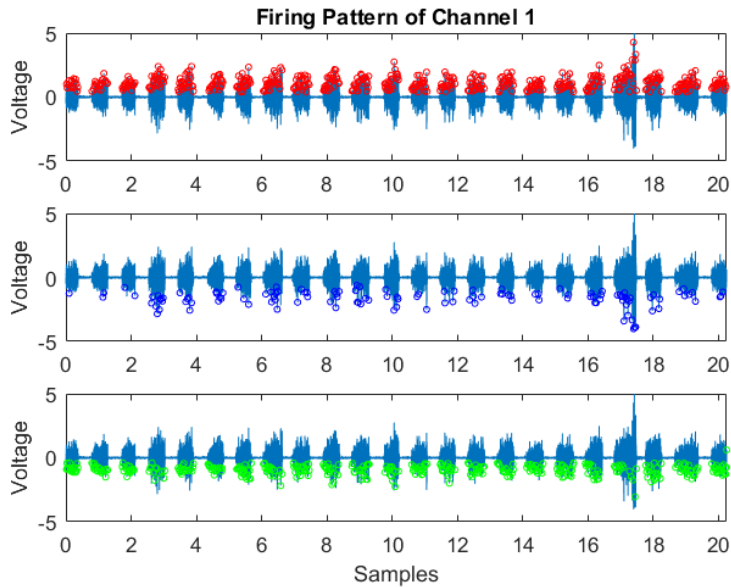


3 clusters, and all the spikes are different



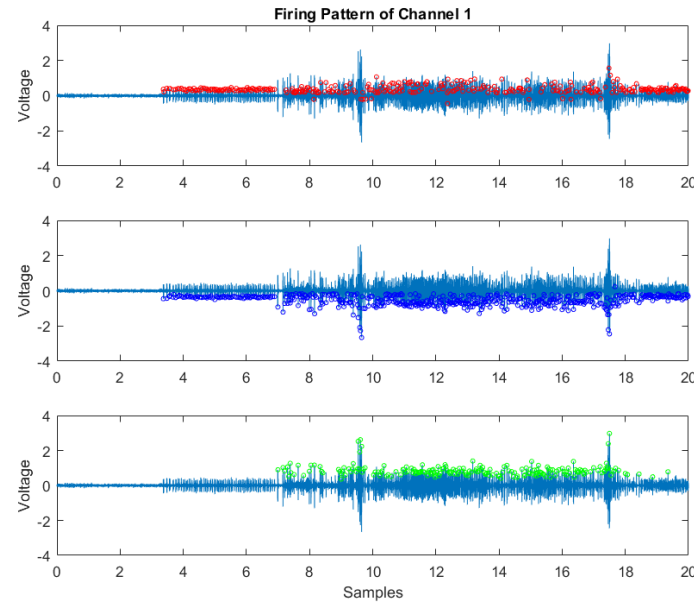
3 neurons

ANALYSIS (2)



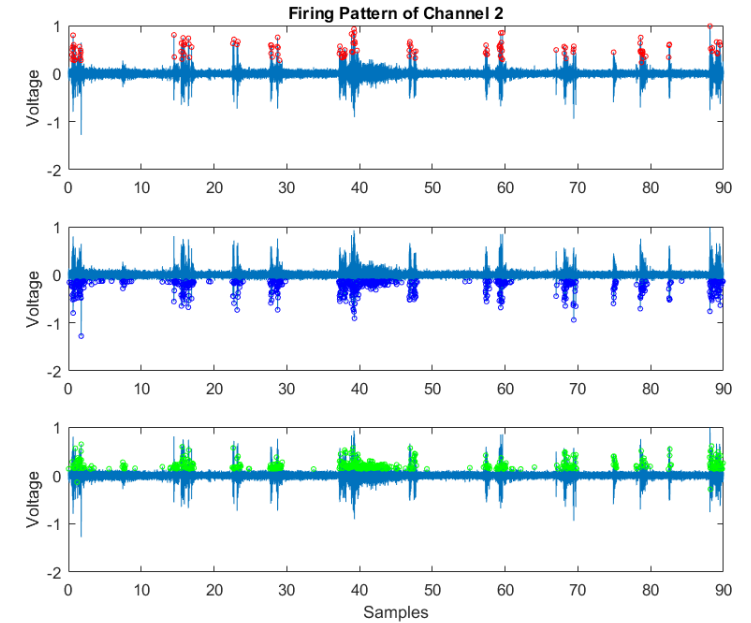
1110 spikes

- **540** in cluster **red**
- **159** in cluster **blue**
- **411** in cluster **green**



1054 spikes

- **332** in cluster **red**
- **518** in cluster **blue**
- **204** in cluster **green**



1208 spikes

- **486** in cluster **red**
- **606** in cluster **blue**
- **116** in cluster **green**

DISCUSSION (1)

1. **Input File:** Plot the EMG data as well as the spectrum to find the cutoff frequency for the filter.
2. **Filter Signal:** Differentiating the signal to improve the SNR and applying a high-pass at 150Hz helps removing the noise. The range [100,250]Hz seems to be the best for the cutoff frequency. Beyond this range, the filter does no help clustering and distorts the signal too much for higher cutoff frequency.
3. **Detect Spikes:** I noticed best results when using a **4*RMS** or **5*RMS** threshold. Increasing the threshold will take less spikes and give less clusters. Decreasing it will take too many and also give less clusters.
4. **Align Spikes:** I chose a window of 15ms to align the spikes and to represent them with enough accuracy. It corresponds to 30 samples as the sampling frequency is 2kHz.

DISCUSSION (2)

5. **Extract Features:** Using the *PCA* method, I am able to extract the principal components and output a scatterplot on the 3 most significant PC scores.

Important note: Matlab *pca()* function gives slightly **variable** results each time you run it. There might be a random component inside the method.

6. **Cluster Spikes:** Using the *kmeans* method, I group the features from the last step into clusters.
7. **Classify Spikes:** Based on this clustering, I show the plot from step 4, where each spikes is attributed to a cluster (color).
8. **Analysis:** Extract the spike templates allows to see the average action potentials. From that, we can find the number of neurons as being the number of **different** templates. I also extract the number of spikes for each cluster to see the most 'active' neuron.

CONCLUSION

The EMG decomposition gave me:

1. *EMG_example_2_fs_2k*: **2 neurons** for channel 1
2. *EMG_example_20s_2000Hz*: **3 neurons** for channel 1
3. *EMG_example_1_90s_fs_2k*: **3 neurons** for channel 2

A couple of tracks that can be explored to improve the decomposition:

- Nonlinear Energy Operator (NEO) based spike detection for a better threshold.
- Use a different filter than Butterworth.
- Manually code PCA to optimize it (instead of using the built-in Matlab function).
- Use a different method for clustering than kmeans, such as: *clusterdata* or *linkage*.