Spike Sorting Utilizing Principal Component Analysis in Matlab (February 2023)

Paul Wanczuk

***Abstract*— Chestek Laboratory provided neural data recording of action potentials with 41,568 observations and 32 data samples. From this information, a spike sorting program was designed in Matlab to attempt to determine the number of different neural signals that are present within this dataset. Utilizing principal component analysis and k-means clustering, a conclusion on the most appropriate number of clusters was determined and visual data in the form of mean and standard deviation graphs were utilized to support these findings.**

***Index Terms*— Spike sorting, Principal component analysis (PCA), K-means clustering, Action potential**

I. Introduction

THE goal of this project was to develop a spike sorting program in Matlab utilizing principal component analysis (PCA). The data used to develop this program were extracellular data recordings of a rodent from the Chestek Laboratory. The data provided 41,568 observations for 32 data samples which had action potential voltages expressed in the microvolt range. In addition to this data, Chestek Laboratory provided the sampling frequency of 2000 Hz which was utilized for collecting the data.

With the acquisition of neural signal data in the form of action potentials, an electrode will encounter multiple neurons that each provide individual signals that overlap in the recording. As a result of this, spike sorting becomes an essential process in being able to identify and classify each individual signal. In spike sorting, the electrode signals are filtered and then analyzed to determine thresholding and clustering for different activities of neurons [1].

The most common method for performing spike sorting is PCA. This method allows for the reduction of dimensionality of data while retaining most of the variation that is present within the dataset [2]. More specifically, PCA identifies new variables which are linear combinations of the original variables [3][4]. In the context of spike sorting, it processes each signal and allows for estimation of a signal's origin relative to the electrode that collected the information.

II. Methods For Spike Sorting

The rodent data which was provided by the Chestek Laboratory included a sampling frequency (Fs) for the data collected rather than the time at which each data sample was taken. In order to plot the data provided, a time (T) has to be defined for each signal.

$$T = \frac{1}{Fs} \cdot 1000 \tag{1}$$

(*1*) describes the equation utilized in Matlab to generate the time that would be necessary for plotting. This equation converts the frequency into time (seconds) and then is multiplied by 1000 to utilize a more reasonable time scale of milliseconds. This time value would then need to be multiplied by the number for each data sample to get all the time samples required for plotting. For this data, it resulted in the generation of 32 time points, one for each data sample.

Following this, the mean and standard deviation of the data is collected to understand the distribution or variability of signals within the data. Large variation in the data can express the presence of different neuron action potentials being present.

Next, PCA is utilized to reduce the dimensionality of the signals as well as to linearize it. With the first two principal components PC1 and PC2, a plane can be generated that then has the original data projected onto it. With this plot, any data sample can be selected and reconstructed utilizing the mean (M), as well as the first and second principal component coefficients (E_1 and E_2) and their score values (A_1 and A_2).

$$Reconstruct = M + A_1 \cdot E_1 + A_2 \cdot E_2 \tag{2}$$

(*2*) expresses how the original data sample can be reconstructed by the sum of mean and products for coefficients and scores of the first two principal components.

After projecting the data onto a principal component plot, a Matlab function "k-means" can be utilized to categorize data into clusters based on the number of clusters that the user expects there to be.

A third dimension will then be established utilizing the third principal component, and this will allow for visualizing clusters in three dimensions to allow for another

This was submitted for review on February 17th 2023 for Project 1 in BME 4361 Neural Engineering course. This work has not received any support or funding.

P. Wanczuk, is with the University of Florida, Gainesville, FL 32605 USA (e-mail: pwanczuk@ufl.edu).

method of visualizing clusters and would utilize the same "k-means" function to categorize these clusters.

III. Results

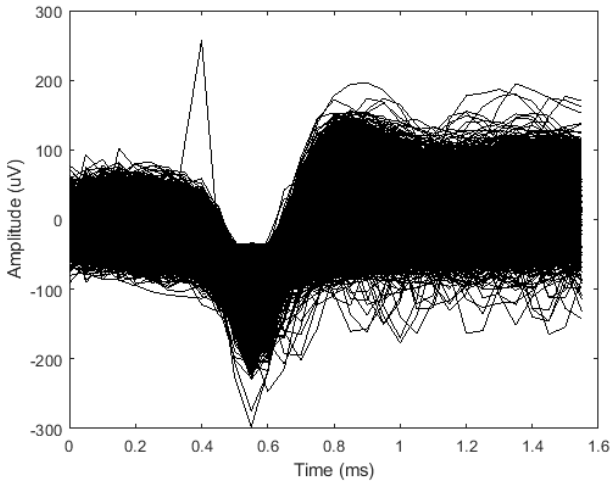

Fig. 1. Raw data of neuron action potential firing in microvolts(uV) over time milliseconds (ms).

The neuron firing is presented in *Fig. 1.* and it expresses the lack of clarity or ability to distinguish individual neuron firing within the data provided. Therefore, the mean and standard deviation were then collected to understand how the signals act in general and the range in variation within this data which is depicted in *Fig. 2.* Comparing these two figures, the range in voltage for *Fig.1.* was much larger and had a larger range of -300uV to -300uV than in *Fig. 2.*

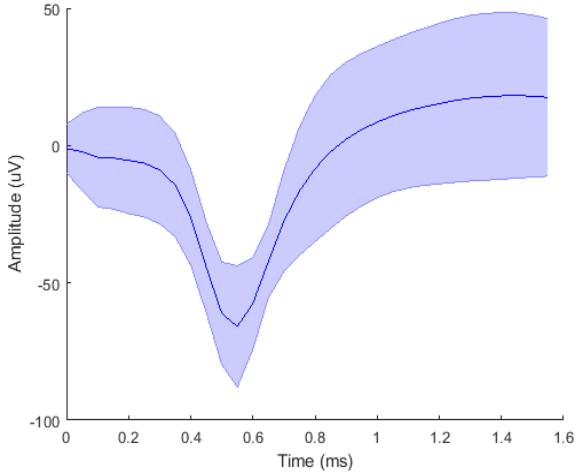

Fig. 2. This depicts the mean and standard deviation of the data presented in *Fig. 1.* with amplitude over time. Mean is depicted by the solid blue line and standard deviation by the lighter blue shaded region.

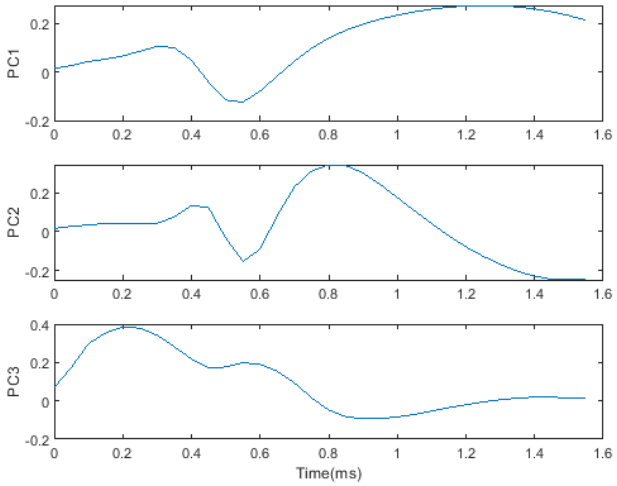

Fig. 3. Plot depicting the graphs for the first three principal components (PC1, PC2, and PC3) over time.

The first three principal components were generated in *Fig. 3.*, for developing both 2-dimensional (*Fig. 4.*) and 3-dimensional (*Fig. 9.*) plots for which the neural data would be projected onto and form clusters.

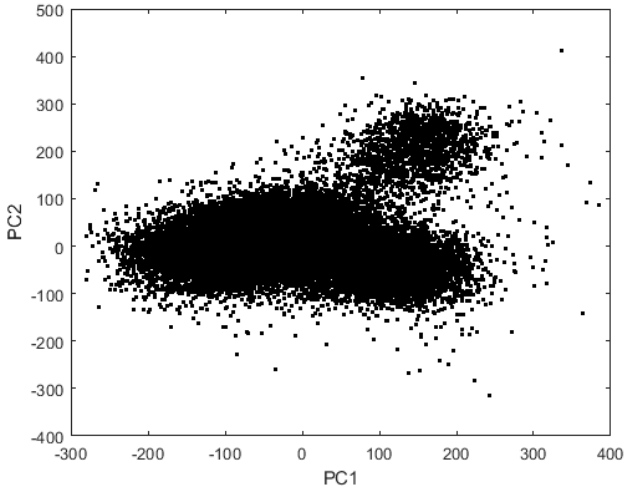

Fig. 4. Representation of the data observations being projected onto the first principal component (PC1 on the x-axis) and second principal component (PC2 on the y-axis) two-dimensional plot.

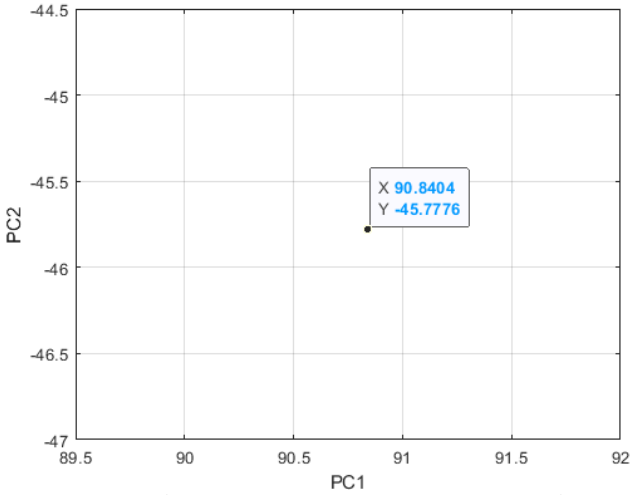

Fig. 5. Depiction of the PC1 (X) and PC2 (Y) component values for the second data sample.

Next, one of the 32 data samples was selected to be reconstructed. In *Fig. 5*., data sample 2 was chosen to be reconstructed and the first and second principal components were taken and utilized in the reconstruction of the sample. Then utilizing equation (2) the values were utilized to generate a reconstructed signal which is compared to the original in *Fig. 6.*

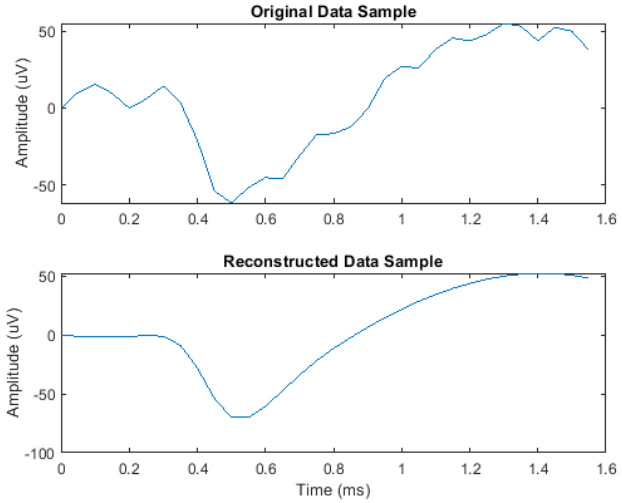

Fig. 6. Original data sample (TOP) amplitude over time compared to the reconstructed data (BOTTOM). The reconstruction equation (2) was utilized to generate the reconstruction which appears smoother than the original.

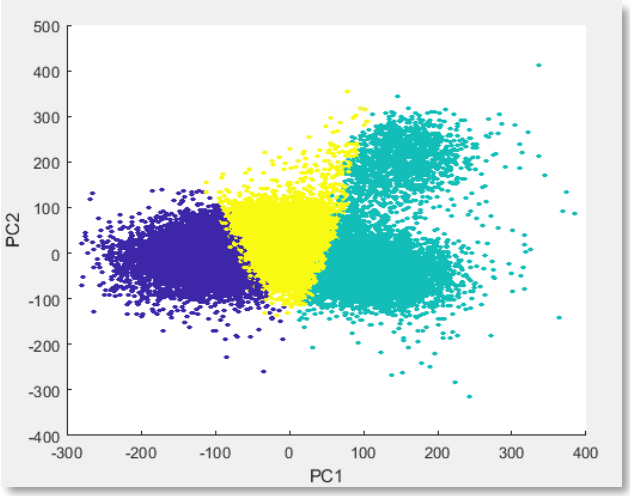

Fig. 7. Representation of k-means clustering using 3 different cluster regions. Cluster 1 is blue, cluster 2 is green, and cluster 3 is yellow. This result was achieved after two iterations of the Matlab program.

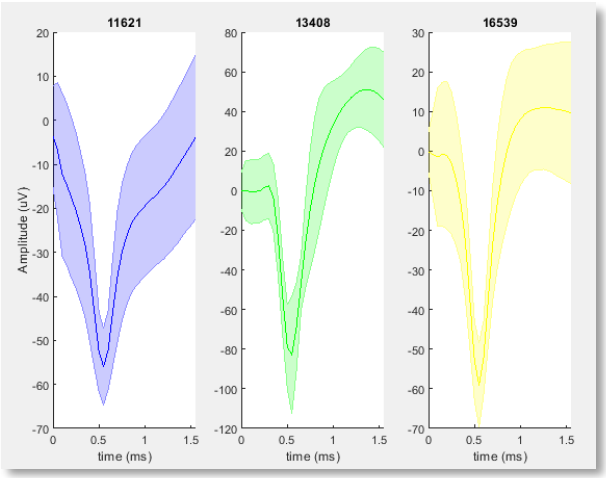

Fig. 8. The mean (solid line) and standard deviation (lighter filled regions) for each of the cluster's data samples. The colors are associated with the same colors as for Fig. 7. (Blue 1, Green 2, Yellow 3). The numbers above each plot represent the number of observations for each cluster.

Fig. 7. presents the k-means clustering for the data based on the claim that there are 3 neural signals present in the system. This was evaluated by looking at the ration between clusters which appeared approximately equal based on the number of observations in each cluster for *Fig. 8*.

Then, the 3rd principal component was introduced. To present the data in a 3-dimensional space *(Fig. 9*.). The same process of reconstruction was utilized in order to reconstruct data sample 2, but this time including the third principal component in the calculations (*Fig. 10*.).

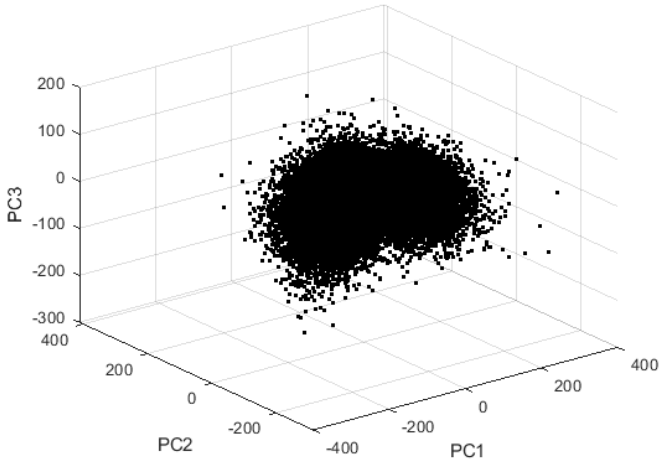

Fig. 9. Projection of data onto a 3-dimensional plot with PC1 on the x-axis, PC2 on the y-axis, and PC3 on the z-axis.

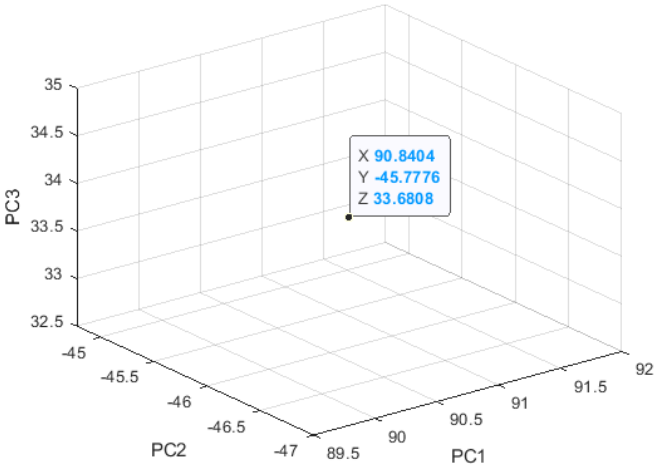

Fig. 10. This is expressing data sample 2 as a point on the 3-dimensional plane with PC1 coefficient being X, PC2 being Y, and PC3 being Z.

Following the identification of coefficient values for data sample 2, equation *(2)* was utilized with the added dimension which would include the addition of A3 and E3 to that equation which results in *Fig. 11*.

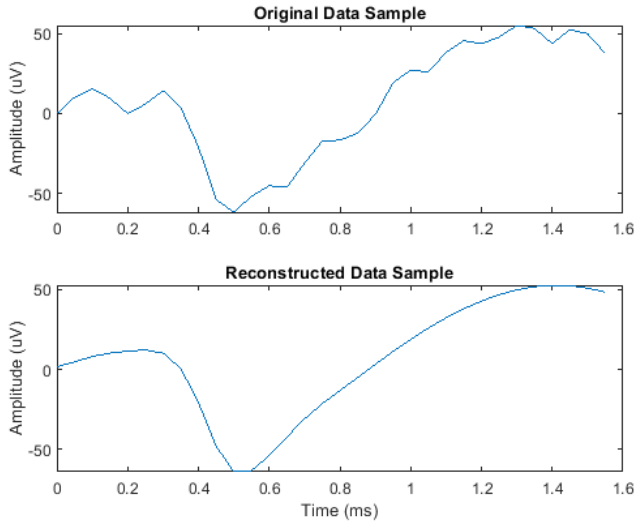

Fig. 11. Original data sample (TOP) amplitude over time compared to the reconstructed data (BOTTOM). The reconstruction equation (2) with the addition PC3 component was utilized to generate the reconstruction which appears smoother than the original.

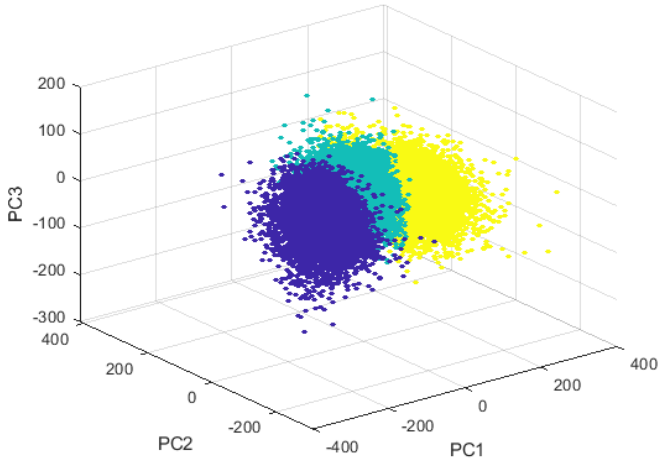

Fig. 12. K-means clustering in the 3-dimensional plane with blue representing cluster 1, green representing cluster 2, and yellow representing cluster 3.

Next, k-means clustering was conducted utilizing 3 cluster regions again *(Fig. 12.)* which resulted in means and standard deviation that were very similar to the 2-dimensional results *(Fig. 8., Fig. 13.)*.

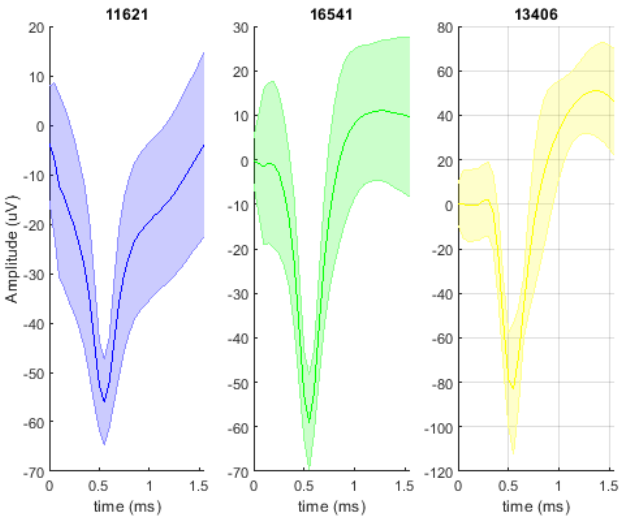

Fig. 13. The mean (solid line) and standard deviation (lighter filled regions) for each of the cluster's data samples. The colors are associated with the same colors as for Fig. 12. (Blue 1, Green 2, Yellow 3). The numbers above each plot represent the number of observations for each cluster.

IV. Discussion

When analyzing the results of this spike sorting program, it is important to note the variability that can occur with utilizing the k-means algorithm. This method for identifying spikes appeared to be most effective when distinguishing clusters of well defined regions or shapes. With the data in this program, in the 2-dimensional space (*Fig. 4.)*, there appeared to be two distinct clusters; however, in the 3-dimensional space there appeared to be only one large cluster (*Fig. 9.).* Since this data had over 40 thousand observations, outliers in the plots had very minimal effect on the clustering results.

This leads to the major limitation of k-means clustering being that the user has to identify how many clusters or neurons there would be when generating an action potential which would lead to bias on the user's input. In this project, 3 clusters appeared to be the most effective prediction when utilizing circular clusters. Looking at the cluster distribution in *Fig. 8*. and *Fig. 13*. The distribution among clusters appeared fairly equal with the 2-dimensional having a distribution ratio of 11,621:13,408:16,539 and the 3-dimensional having a distribution having a ratio of 13,406:11,621: 16,541 which are nearly identical results. It is also important to identify that the clustering ratio for the 2-dimensional plot resulted from running the program twice while the 3-dimensional only required one run to achieve this even clustering.

Based on this result, it can be concluded that clustering utilizing the first 3 principal components was more effective in spike sorting than only utilizing the first 2. This is further justified when comparing the reconstructions of data sample 2 utilizing the first 2 and first 3 critical components (*Fig. 14.*). Specifically, the 3 principal component reconstruction appears to more accurately model the steepness of the drop of the original signal at around 0.4 milliseconds than the reconstruction utilizing only the first 2.

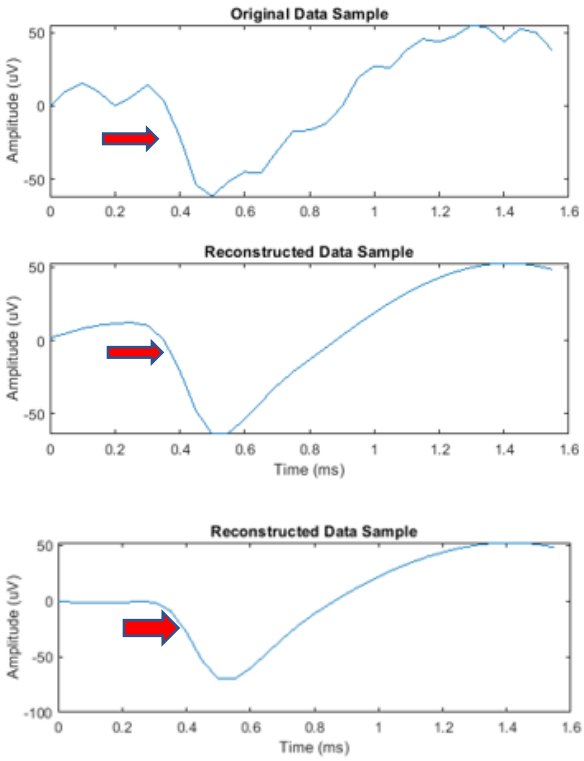

Fig. 14. Depicting the original data sample 2 (TOP) and reconstruction utilizing the first 3 principal components (MIDDLE), and first 2 principal components (BOTTOM). Red arrows denoting the steepness of curves.

References

[1] K. Oweiss, "Spike sorting," Spike Sorting - an overview | ScienceDirect Topics, 2016.[Online]. Available: https://www.sciencedirect.com/topics/neuroscience/spike-sorting.

[2] Jolliffe, I.T. *Principal Component Analysis* (Springer, New York, 2002).

[3] Ringnér, M. What is principal component analysis?. *Nat Biotechnol* 26, 303–304 (2008). https://doi.org/10.1038/nbt0308-303

[4] L. Xu, J. G. Taylor, T. D. Sanger, M. D. Plumbley, E. Oja, J. Karhunen, C. Jutten, P. Comon, G. Burel, P. Baldi, T. W. Anderson, P. Bekker, H. Bourlard, A. Cichocki, P. A. Devijver, A. Gifi, G. H. Golub, T. Hastie, S. Haykin, J. Hertz, A. K. Jain, and J. Joutsensalo, "Generalizations of principal component analysis, optimization problems, and Neural Networks," *Neural Networks*, 20-Apr-2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0893608094000987.