Spike Sorting of Rodent Extracellular Recordings using Principal Component Analysis and K-Means Clustering

Jacey L. Davies, *University of Florida*

***Abstract*—Spike sorting is the method of isolating signals generated by individual neurons from recordings with multiple neurons. This is a necessary process for examining the effect of single neurons on an action or behavior. Sorting manually by hand is time consuming, so sorting is done digitally by combining principal component analysis (PCA) and k-means clustering in MATLAB. PCA is a process by which the most important components of the data, the principal components, are isolated and the data reconstructed using these components. The first two and the first three components are used here. K-means clustering uses k number of randomly selected centers and sorts the data into k clusters based on their distance from each center. This process was tested using extracellular signals from rodent recordings. Three clusters worked best with the two-dimensional data and four clusters worked best with the three-dimensional data. Despite the variation still present in the spike clusters, the spikes were sorted well enough to be preferred over manual sorting.**

***Index Terms*—Spike sorting, feature extraction, clustering, k-means clustering, principal component analysis**

I. Introduction

THE extracellular recording of neuronal activity allows for exploration of how the activity of individual neurons influence certain actions or behaviors. However, an implanted electrode will inevitably contain action potentials (spikes) from several nearby neurons. These spikes need to be sorted in order to isolate the targeted neuron(s). Additionally, any background noise from the multitude of more distant neurons needs to be attenuated. This sorting is based on the shape of the spikes, which is influenced both by differences between individual neurons and by the neurons' distance from the electrode. Manual sorting of spikes can be done by observing differences in the peak amplitude, width, and after-hyperpolarization [1]. This process is time-consuming, costly, and prone to human error. Thus, this process has become automized to increase efficiency.

The spike sorting process usually consists of three main steps: spike detection, feature extraction, and clustering [1]. Spike detection is in which a voltage threshold is set to determine if a spike occurred. The spikes are then aligned together. The provided rodent data have already undergone spike detection processing and thus that will not be explored in this report. Feature extraction consists of reducing the data into lower feature dimensions. Although using all of the features would be more accurate to the original data, it would complicate the spike sorting process and thus reduce its efficiency. Several algorithms are available for this, but principal component analysis (PCA) is used here. After this, the spikes are sorted into clusters based on these features. Once again, there are several clustering algorithms for processing the features, but k-means clustering was used for simplicity. Different principal components and numbers of clusters (k) were experimented with to find the best results.

J. L. Davies is with the Biomedical Engineering Department, University of Florida, Gainesville, FL 32611 USA.

II. Materials And Methods

The software utilized for the spike sorting was MATLAB 2020. Both PCA and k-means clustering algorithms are built into the program. The shadedErrorBar function was provided to plot the mean signals with shaded error bars. The rodent extracellular data were imported into the program, including the given sampling frequency, 20,000 Hz. There was a total of 41,568 recorded spikes, each with 32 given time samples. The first time sample was at time 0 and thus the end time was 31/20,000 = 1.55ms. The time increments were the inverse of the sampling frequency, or 0.05ms. Thus, a time vector was created in MATLAB starting at 0, ending at 1.55ms, and with 0.05ms to plot the data against. To show the general shape of the spikes, the mean of the spikes was plotted along with shaded error bars to show the variance.

A. PCA

PCA is a multidimensional analysis that aims to extract important information from a set of data and present this information as uncorrelated variables called principal components [2]. PCA puts most of the information of the data into the first few principal components to make analysis easier; the dimensionality of the data is reduced without a large amount of information loss. It begins with subtracting the average values (mean) from the actual values (demeaning the data). The sample covariance matrix is calculated and used to find the eigenvectors and eigenvalues. An eigenvector points to the direction of larger spread/variance and the eigenvalue is the magnitude of the spread. The eigenvalues are sorted in

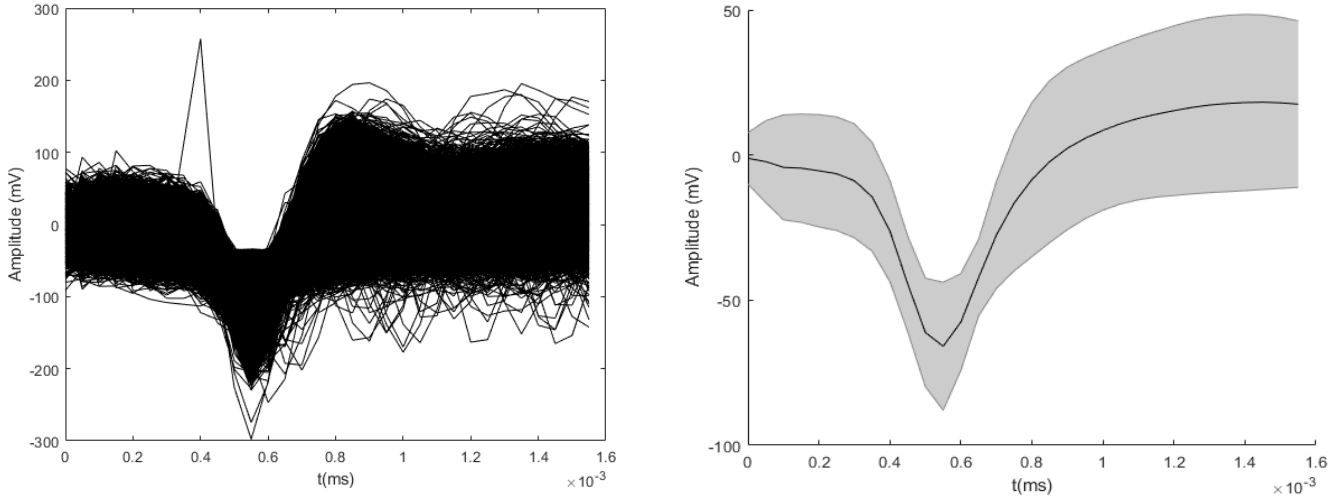

Fig. 1. The extracellular rodent spike data plotted against time (left) and a plot of the mean of the spikes with error bars (right). The spikes were already detected and aligned along the time axis. The left plot shows the data that will be sorted and analyzed. The right plot shows the overall shape of the spikes and their variability.

descending order, so the first few principal components have the most spread and thus the most information from the data. The number of eigenvectors/eigenvalues kept can be calculated [2], but in this report only 2-3 were kept. Using the remaining principal components, the data can be reconstructed to have fewer dimensions (m):

$$X_2 = M + A \cdot E'. \quad (1)$$

Where X_2 is the reduced data, M is the mean of the original data, A is the projection of the data onto the principal component space, and E is the transposed principal components. Reducing the initial data set gets rid of noninformative attributes and noise in the data [2]. However, PCA is highly sensitive to noise and can fail to extract meaningful features when the spike shapes are very similar [1].

The given data first went through PCA using the MATLAB pca() function. The output of this function is a matrix of the transposed principal components (E), the projections of the data in the principal component space (A), and the eigenvalues (L). The first three principal components were found and plotted from E. The observations were projected onto the first two principal components and the resulting projections were plotted onto the 2-dimensional (m=2) space from A. The fifth observation was individually put through this projection and reconstructed using the mean, principal components, and the projections. Later, the observations were projected onto the first three principal components and the resulting projections were plotted onto the 3-dimensional (m=3) space. Once again the fifth observation was used to demonstrate this process.

B. K-means Clustering

After the features were extracted from the data with PCA, they were clustered using k-means clustering. In MATLAB, the kmeans() function takes a set of points (the projected coefficients in this case) and the number k of desired clusters. It will randomly select k points to act as the cluster means. It will then take another data point and assign it to one of the clusters based on which cluster mean it is closest to. This distance is calculated using the Euclidean distance,

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}. \quad (2)$$

The cluster's mean is then recalculated with the new point in mind. This is iterated with all the points [3]. The k-means algorithm does have some limitations. It is very sensitive to the initial positions of the cluster means and thus should be iterated multiple times. It also requires prior knowledge of the number of clusters (k) to be expected [1]. Since the k-means algorithm depends on the Euclidean distance, it sorts circular/spherical clusters of data best.

The projected coefficients from the PCA were clustered using k-means clustering. For m=2, both k=2 and k=3 were tested. For m=3, both k=3 and k=4 were tested. Based on the calculated clusters, the original spikes were sorted. A plot of the mean of each cluster with error bars was made.

III. Results

The given rodent spike data along with its overall mean and standard deviation are shown in Fig. 1. The initial data is very extensive with 41,568 spikes and thus must be sorted digitally. As seen in the mean plot, the overall shape is that of an inverted action potential. There is also a high error, showing that there are multiple spikes present.

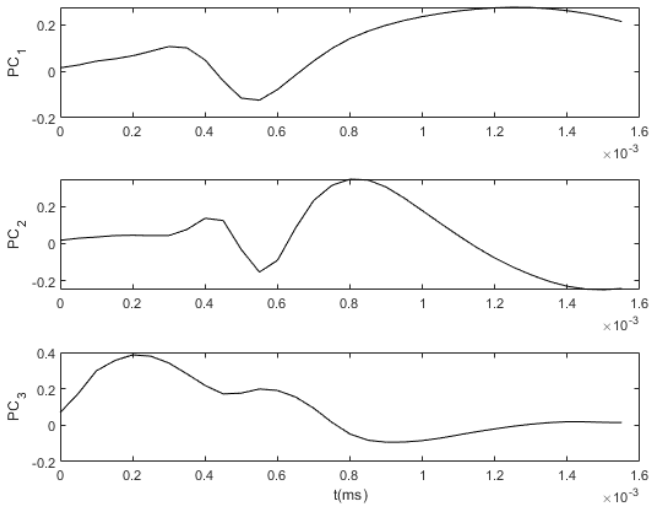

Fig. 2. The first three principal components from the principal component analysis. These components contain the most information from the data and thus are the most defining shapes for the spikes.

A. *PCA in two dimensions (m=2)*

The given data were put through PCA using the MATLAB pca() function. The first three principal components are plotted in Fig. 2. As mentioned previously, the principal components are ordered so the ones with the most information are first. Therefore, these three contain the most defining information for the spikes and are enough for the spike sorting.

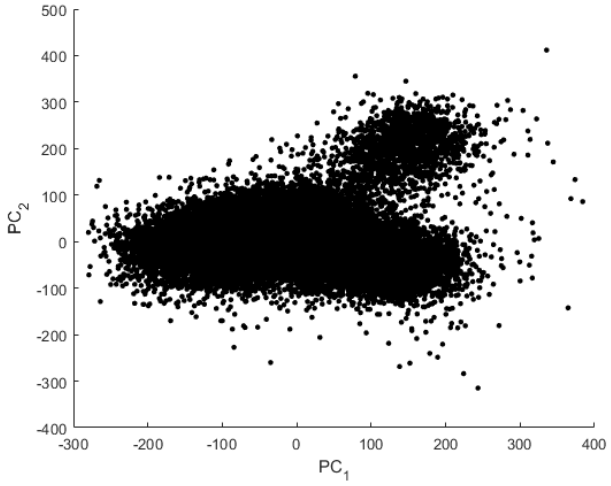

Fig. 3. The projection of the data onto the first two principal components. As there are only two principal components being projected onto, the plot is two dimensional. The points cluster into multiple areas that are not well defined.

The projections of the spikes onto the principal components were contained in the A matrix. The coefficients for the first two components were plotted against each other in Fig. 3. Since only two components were considered, the plot is two dimensional (m=2). The points seem to cluster into certain areas, but these are not well defined. The k-means clustering algorithm will be used to sort these later.

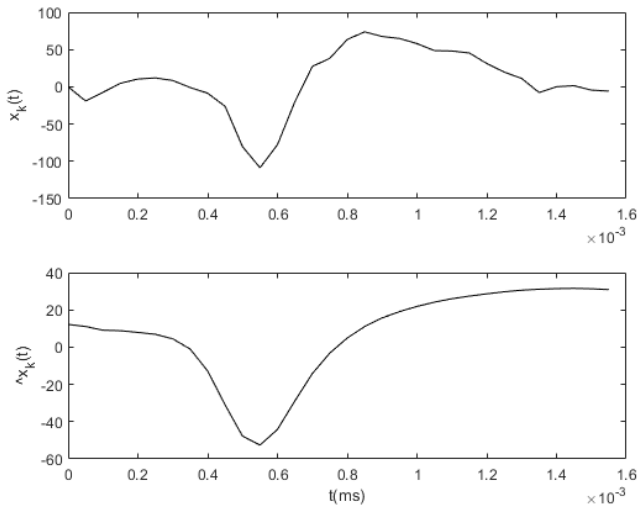

Fig. 4. The fifth observation in the original data set (top) and the reconstruction using the first two principal components (bottom). The reconstruction is missing some of the smaller details of the original but matches the overall shape.

To demonstrate the reconstruction of the data in the lower dimensional space, the fifth observation was examined. The reconstruction was done with (1), specifically with only the first two sets of projected coefficients and the first two principal components. The original spike and its reconstruction can be seen in Fig. 4. The reconstruction is much smoother and is missing some details, but the overall shape is the same.

B. *K-means clustering in two dimensions (m=2)*

The projected coefficients in Fig. 3 then were sorted into clusters using the k-means clustering algorithm in MATLAB. The code was run multiple times to get the best initial cluster mean positions. Only the best-fitting trials are shown here. Both k=2 clusters and k=3 clusters were tested to see which better categorized the data. The former is seen in Fig. 5.

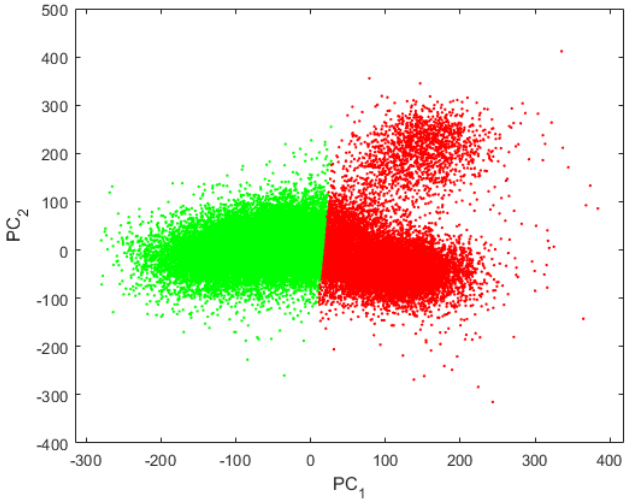

Fig. 5. The projection of the data onto the first two principal components sorted into two clusters by k-means clustering. The red cluster is cluster 1 and the green cluster is cluster 2. The clusters split the data almost in half.

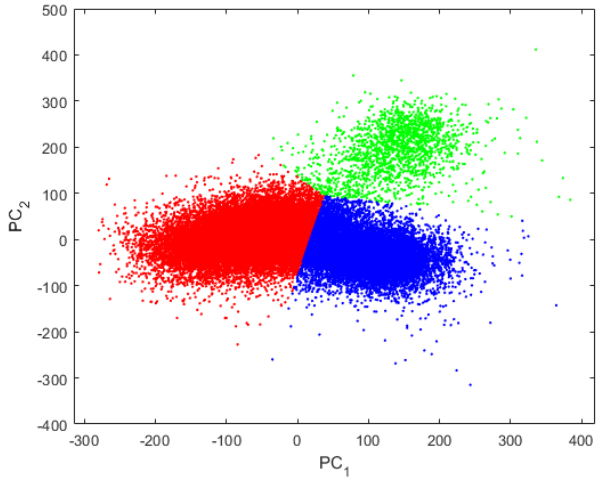

Fig. 6. The projection of the data onto the first two principal components sorted into three clusters by k-means clustering. The red cluster is cluster 1, the green cluster is cluster 2, and the blue cluster is cluster 3. The clusters split the data into three regions, separating the top right cluster.

With two clusters, the data was split almost evenly in half. This does not accurately separate the top right cluster from the larger bottom cluster(s) and thus is not an accurate number of clusters. The red cluster corresponds to the first index generated by the k-means algorithm and the green corresponds to the second. Since the cluster means are randomly selected, the cluster order (and thus colors) may flip.

With three clusters, the data was split into three regions (Fig. 6). This does separate the top right cluster from the bottom ones and better represents the data. However, the distinction between the green and blue is not well defined, and the green cluster may be larger than it should be.

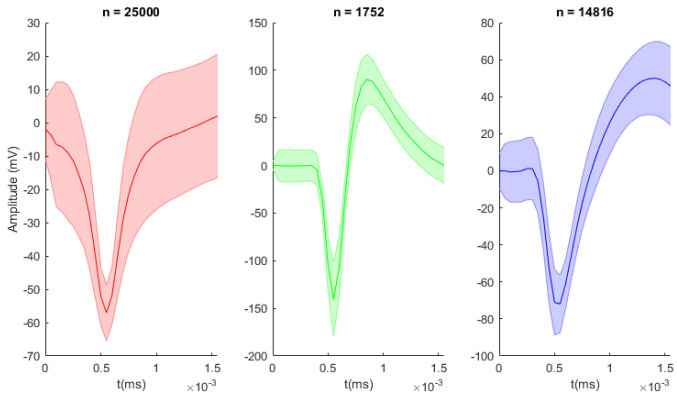

Fig. 7. The means of the sorted spikes with their respective standard deviation error bars. The colors of the graphs correspond to the cluster colors in Fig. 6. The number of spikes in each cluster are in the titles.

The means of the now sorted spikes and the number of spikes per cluster are shown in Fig. 7. The red cluster was the largest and thus has the most error. This suggests a need for further clustering. The green cluster has the least spikes and is well defined as shown by the low error. To try and get a better estimate of the data, the third principal component will be considered as well.

C. PCA in three dimensions (m=3)

Now the data will be analyzed using three principal components rather than two. Theoretically, adding another principal component should add more information from the original data. Thus, the analysis should be more accurate, although more complicated. As before, the projection of the data onto the principal components will be plotted and sorted.

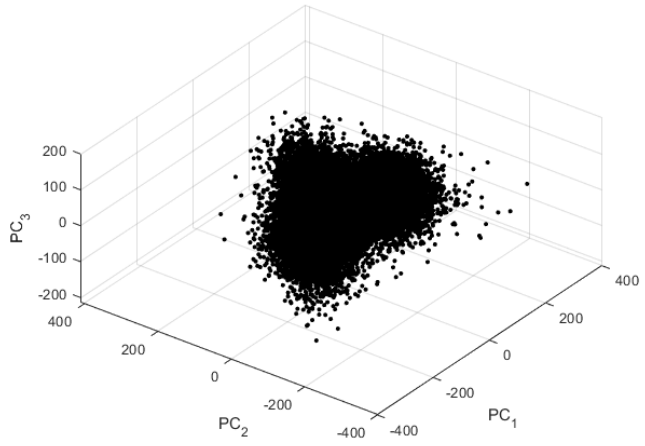

Fig. 8. The projection of the data onto the first three principal components. Since there are three principal components being projected onto, the plot is three dimensional.

Now there are three principal components being used, the coefficients are plotted in three dimensions (Fig. 8). The clusters are still not very well defined and thus k-means clustering must be used later. First the data must be reconstructed into the lower, three-dimensional space.

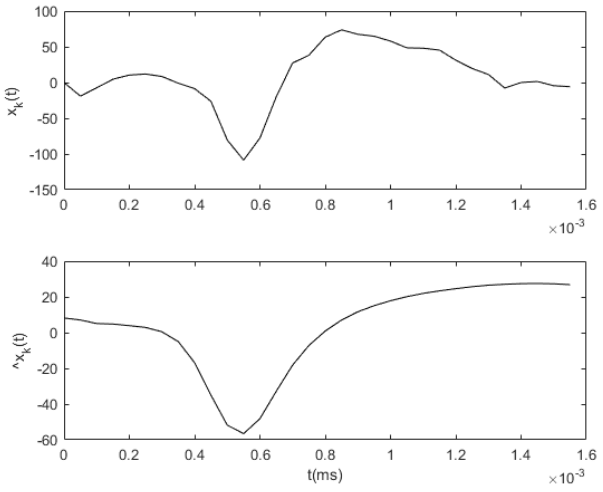

Fig. 9. The fifth observation in the original data set (top) and the reconstruction using the first three principal components (bottom). The reconstruction is missing some of the details of the original but matches the shape. It is very similar to Fig. 4 but has slightly different values.

The reconstruction was done with (1) again, this time with the first three sets of projected coefficients and the first three principal components. The original spike and its reconstruction can be seen in Fig. 9. As before, the same general shape is there, but some details are lost. The reconstruction looks almost

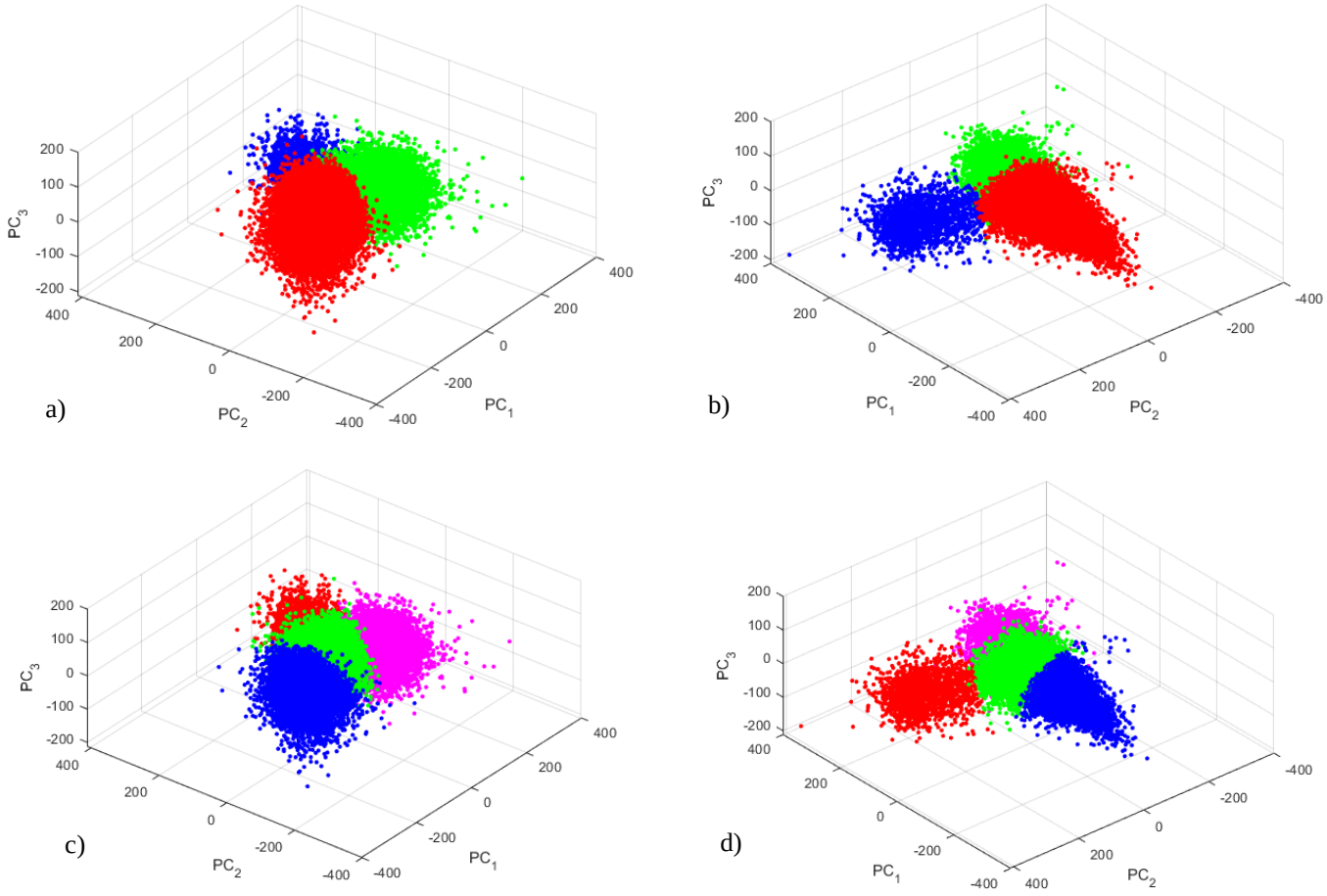

Fig. 10. Different views of the k-means clustering of the projections of the spikes onto the three principal components. For k=3 clusters, a general view (a) and a view from the opposite side showing how large the red cluster is (b). For k=4 clusters, a general view (c) and a view from the opposite side showing the same large cluster split in half now (d). Due to the random assigning of initial cluster means, the colors change between the top and bottom row.

identical to the one with only three principal components, just with slightly different values. This implies that the third principal component may not have been as present in this observation as the first two were. The data supports this. The values of A for this observation for the first three components were 87.2, 171.1, and -10.4 respectively; the third coefficient was much smaller than the other two in magnitude.

D. K-means clustering in three dimensions (m=3)

Adding a new dimension results in another axis by which the projected coefficients can differ. Therefore, the number of clusters tested were greater, with k=3 and k=4.

The plotted and sorted projections can be seen in Fig. 10. It is difficult to visualize three-dimensional data from a two-dimensional image, so more than one view was used. From an initial look at the plot (Fig. 10. a.), it would seem that three clusters are enough. However, looking from the side, the red cluster is much larger than the other two and is more elongated (Fig. 10. b.). This implies that it may consist of two clusters. Therefore, k-means clustering was applied again on the data, with k=4 clusters this time. The larger cluster from Fig. 3. is now split into three clusters (Fig. 10. c.). Looking from the side, this makes the distribution of spikes much more even (Fig. 10. d.). However, the border between the green and blue clusters is not well defined by the shape of the data. One large cluster is still a possibility.

The four final clusters of spikes were averaged, and their means and error bars can be seen in Fig. 11. The green and blue clusters still have relatively high error bars, but the red and magenta clusters seem to be consistent.

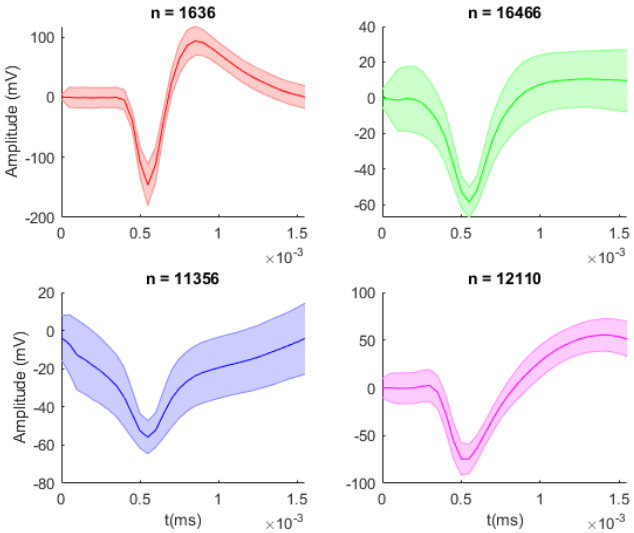

Fig. 11. The means of the sorted spikes with their respective standard deviation error bars. The colors of the graphs correspond to the cluster colors in Fig. 10. c-d. The number of spikes in each cluster are in the titles.

IV. Discussion

The use of PCA and k-means clustering for spike sorting was straightforward and did not require nearly as much time as manual sorting with more accuracy. However, both parts of the process have their limitations.

As mentioned previously, PCA is highly sensitive to noise and can fail to extract meaningful features when the spike shapes are very similar [1]. For this data, the spike shapes were distinct enough to find three to four clusters, even if the clusters themselves have some ambiguity towards the edges. Another limitation of PCA is the loss of information from discarding later principal components. The data reconstructed from PCA had fewer dimensions and thus was easier to analyze but lost some of the smaller details as seen in Fig. 9. Therefore, it is important to determine how many principal components need to be kept. Using three components instead of two made it easier to differentiate more clusters since there was an additional axis to separate the data. The additional component did not improve the consistency of each cluster, though, as seen in Fig. 11. Looking at the eigenvalues, the first three values were much higher than the others. The third value was 1,440 whereas the fourth was 796, almost half as much. Since the eigenvalues represent the spread in the data, and the greater spread means the more information stored, this would mean that the first three components hold the majority of the information needed for spike sorting. Thus, I believe that including the third component was necessary.

K-means clustering also has several limitations. It is very sensitive to the initial positions of the cluster means, which are generated randomly here. Thus, there is little control and multiple trials must be done. It also requires the assumption of the number of clusters, which is unknown for this data. The algorithm also shifts the mean of each cluster based on the points included in it. Therefore, it is sensitive to outliers as they can distort the mean. Since the k-means algorithm depends on the Euclidean distance, it sorts circular/spherical clusters of data best, which may not fit the data analyzed here. Nonetheless, the use of four clusters worked well with the data in the three-dimensional space.

References

[1] J. Eom et al., "Deep-learned spike representations and sorting via an ensemble of auto-encoders," *Neur. Net.*, vol. 134, pp. 131-142, Nov. 2020, doi: 10.1016/j.neunet.2020.11.009.

[2] O. Užga-Rebrovs and G. Kulešova, "Initial Dataset Dimension Reduction Using Principal Component Analysis," *Info. Tech. and Manag. Sci*., vol. 23, pp. 41-44, 2020, doi: 10.7250/itms-2020-0006.

[3] G. A. Wilkin and X. Huang, "K-Means Clustering Algorithms: Implementation and Comparison," in *Sec. Internatl. Multisymp. on Comp. and Comp. Sci*., 2007, pp. 133-136, doi: 10.1109/IMSCCS.2007.51.