Project 3: Neural Spike Sorting

George Rabadi, Julianna Caton, Lauren Terry

BIOMEDE 599.19

Introduction

The following report focuses on peak finding and subsequent analysis. Specifically, it focuses on finding individual neural spikes then grouping spikes through clustering. From there, individual characteristics of the individual clusters were analyzed to better understand the “average” spike associated with the cluster. This exploration was undertaken on data collected from 5 separate electrodes which had been implanted into the brain of a rat.

In terms of computational tools, all of this analysis was done in MATLAB. All of the code was written from scratch, using the provided documentation as guidance. Outside of the basic calculations, this analysis relied on *findpeaks* to pull the neural spikes from the signal. Principal Component Analysis was then used in tandem with the k-means clustering algorithm to connect individual spikes into groups. All the analysis and visualization code can be found here: https://drive.google.com/drive/u/1/folders/1nX2PhZt5G3OnVHnIxXkN-PdBOLMrwG6h.

Results and Analysis

With a sampling frequency of 24,414.1 Hz and a length of 22,624,650 data points, the recorded signal comes out to about 927 seconds. The signal in each channel has generally the same characteristics, with large peaks and an incredibly noisy baseline signal. The frequency and magnitude of the peaks in each channel vary, which can be seen in **Figures 1-10**. Included are both the full signals and the first 40 seconds of each channel to better visualize the structure of the base signal and peaks. The means and standard deviations of each channel can be found in **Table 1**. To better understand the relationships between each channel, a Pearson’s Correlation Coefficient was calculated for each pair of channels, found in **Table 2**. The pairs with the highest correlations consisted of Channels 1 and 2 and Channels 4 and 5, with correlation coefficients of approximately 0.5. While these are relatively weak correlations, they are substantially higher than the correlations found for the other pairings at about 0.15. This implies a relationship between Channels 1 and 2 and Channels 4 and 5. While we don't know the physical placement of the electrodes, we can assume that Channels 1 and 2 were close enough to pick up electrical activity from the same neurons, with the same holding for Channels 4 and 5.

Once the data was loaded and preliminary analysis was completed, the next step was to pick the actual spikes. At a basic level MATLAB's built in *findpeaks* function was used to pick out the large peaks from the data in each channel. In practice, the difficulty came from identifying actual spikes compared to noise in the signal. *findpeaks* allows the user to set thresholds and minimum values, among other properties, but these values were difficult to determine algorithmically. In the end, the plots were used to determine reasonable thresholds for the peaks, while the widths of the peaks came from the knowledge that neural spikes last on the order of 1ms. The threshold values were found separately for each channel, and the list of peaks was recorded for each channel. The found peaks for Channel 1 can be seen in **Figure 11** and **Figure 12**, superimposed on the raw data. Once the indices of the peaks were found, the shapes of the associated spikes were pulled from the signal. Assuming the spikes last 1ms and are symmetric about the peak, 12 data points were pulled before and after each peak. 12 data points was picked based on the sampling frequency of the data, as 12 points is approximately 0.5ms. An example spike can be seen in **Figure 13**, which was pulled from Channel 4. It should be noted that the positive and negative spikes were found separately (a limitation of the *findpeaks* function). To prevent a spike that had both a positive and negative peak from being double-counted, any overlapping spikes were removed, keeping the spike which had the largest magnitude peak.

With the peaks and corresponding spikes in hand, it was time to cluster the spikes. Clustering was done using the k-means algorithm, which is built around clustering data points around a predetermined number of centroids. Before clustering could occur, it was necessary to choose features to cluster against. Many approaches could be used here, but in this analysis, clustering was done based on the results from Principal

Component Analysis, or PCA. PCA is an extension of Singular Value Decomposition, and reduces a matrix into principal components. The components are sorted in descending order of importance, based on how much of the variability in the data they account for. For simplicity, the neural spikes were clustered by the first two principal components. In all analyses, the number of clusters was determined using the “Elbow Method”, which looks at the inertia of each clustering. In this context, inertia refers to the distance from each data point to its centroid, and is used as a measure of the clustering’s accuracy, with a lower inertia being a more accurate clustering. In general, the greater the number of clusters, the lower the inertia of the clustering, but at the cost of computation time. As the number of clusters increases, there will initially be a sharp drop off in inertia followed by diminishing decreases. This point of inflection, or “elbow” is picked as the number of clusters.

The first analysis was done combining the spikes found from all the channels then following the PCA-clustering procedure outlined above. The PCA results of these spikes can be seen in **Figure 14** before any clustering was performed, with the clustering results seen in **Figure 15** and the corresponding Elbow Plot in **Figure 16**. Based on the Elbow Method, 4 clusters were used, implying 4 neurons contributing to the raw data. **Figure 17** shows what the “average” neuron looks like in each cluster, which came from finding the average value in each time point for the neurons in a given cluster. The transparent polygon around the average neuron comes from the standard deviations of the individual neuron data and gives an indication of how varied the neurons are.

The above analysis was repeated for each channel separately. Channel 1 and 2 were each found to again have 4 clusters, and can be seen in **Figures 18-21** and **Figures 22-25**, respectively. Channel 3 was found to have 6 clusters, which was an anomaly, and can be seen in **Figures 26-29**. Channels 4 and 5 were again found to have 4 clusters, and can be seen in **Figures 30-33** and **Figures 34-37**, respectively. As can be seen in the raw signal, the data from channel 3 was incredibly noisy without clearly visible peaks, which made finding distinct spikes much more difficult, leading to the poor clustering. This provides further evidence that there were 4 main neurons for which data was collected, with Channels 1, 2, 4, and 5 holding the bulk of the informative data.

After clustering, some work was done to better understand the firing rates of the neurons. To find the firing rate of each neuron, a 10 second long moving window was used on the data, moving 10ms between frames. Within each window, the number of peaks was counted, with this data being used to calculate the neuron's firing rate in Hertz. The firing rates for each neuron can be seen in **Figure 38**. Each neuron had a typical firing rate on the order of 1-10Hz, or firing every 0.1-1s. It should be noted that each neuron seemed to have an independent firing pattern, with the firing rates seeming basically independent.

At this point, the stimulation data was explored, and can be seen in **Figure 39**. This stimulation was applied to peripheral nerves on a visceral organ. Trying to understand the relationship between the two signals was rather difficult. Looking at the signals, there does not seem to be much of any visual similarity or connection. To quantify this, correlation coefficients were calculated, between the neuron firing rates and the stimulation, but were all effectively 0. This implies that there is very little relationship between the stimulation in the peripheral nerves and the actual firings in the rat's brain. At a naive level, this does not make sense, as one would expect stimulation in peripheral nerves to directly affect the firing of nerves in the brain. Some simple explanations, such as a neurological disorder or improper electrode placement, could be theorized, but substantially more information and exploration would be required to draw a stronger conclusion.

Figures

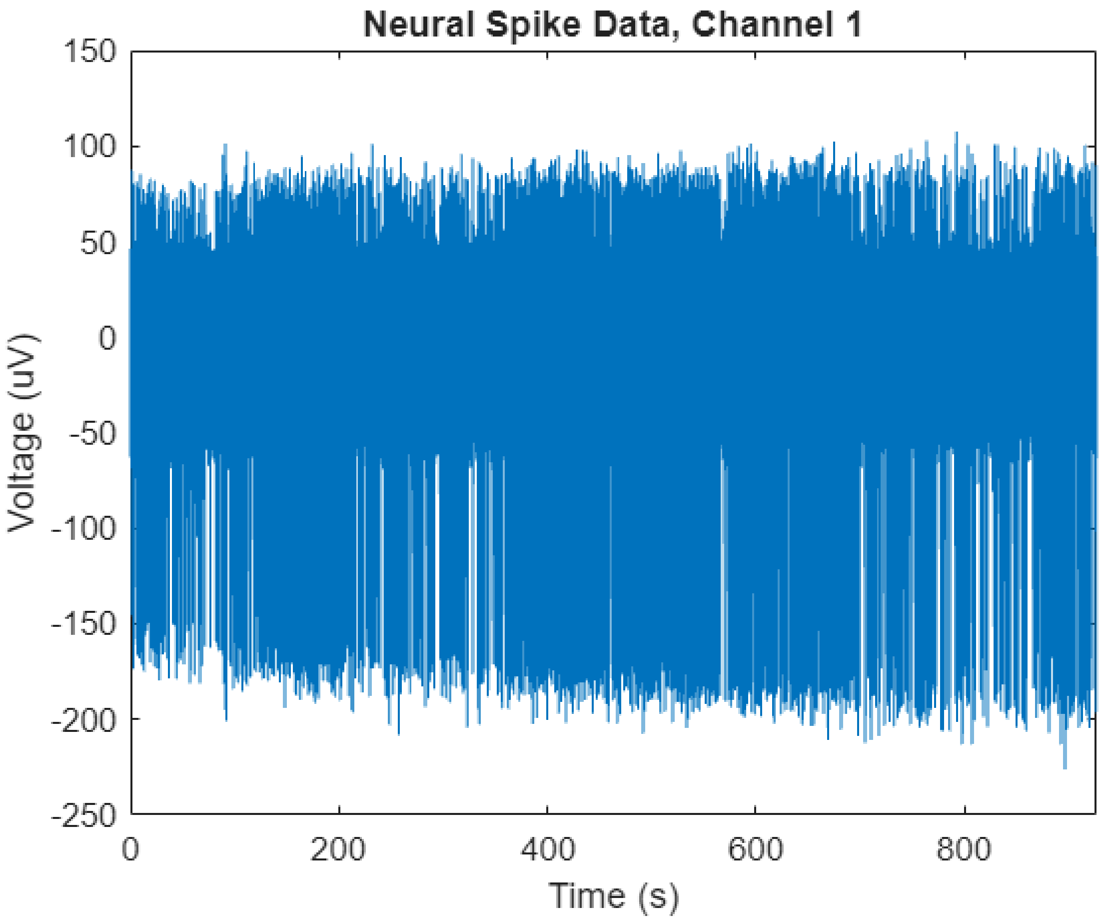

Figure 1. Raw Neural Spike Data, Channel 1

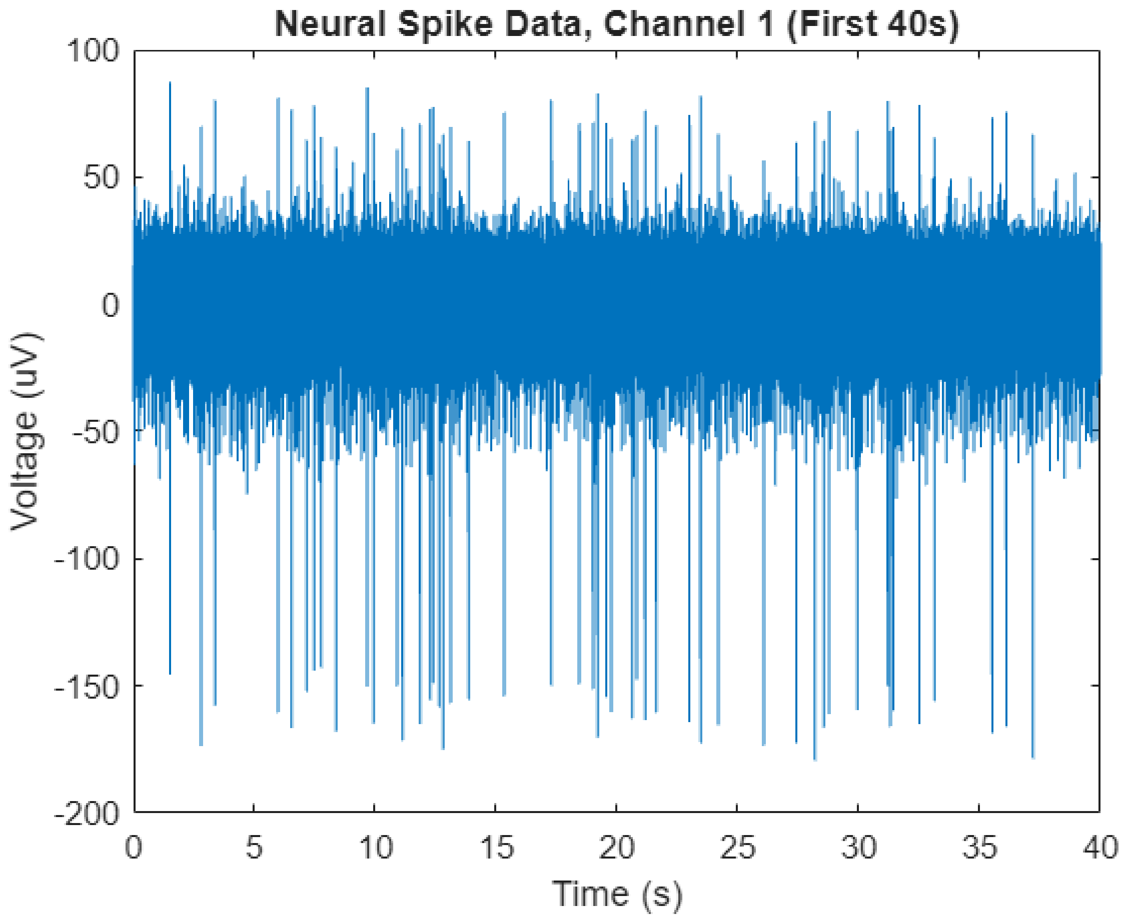

Figure 2. Raw Neural Spike Data, Channel 1 (First 40 Seconds)

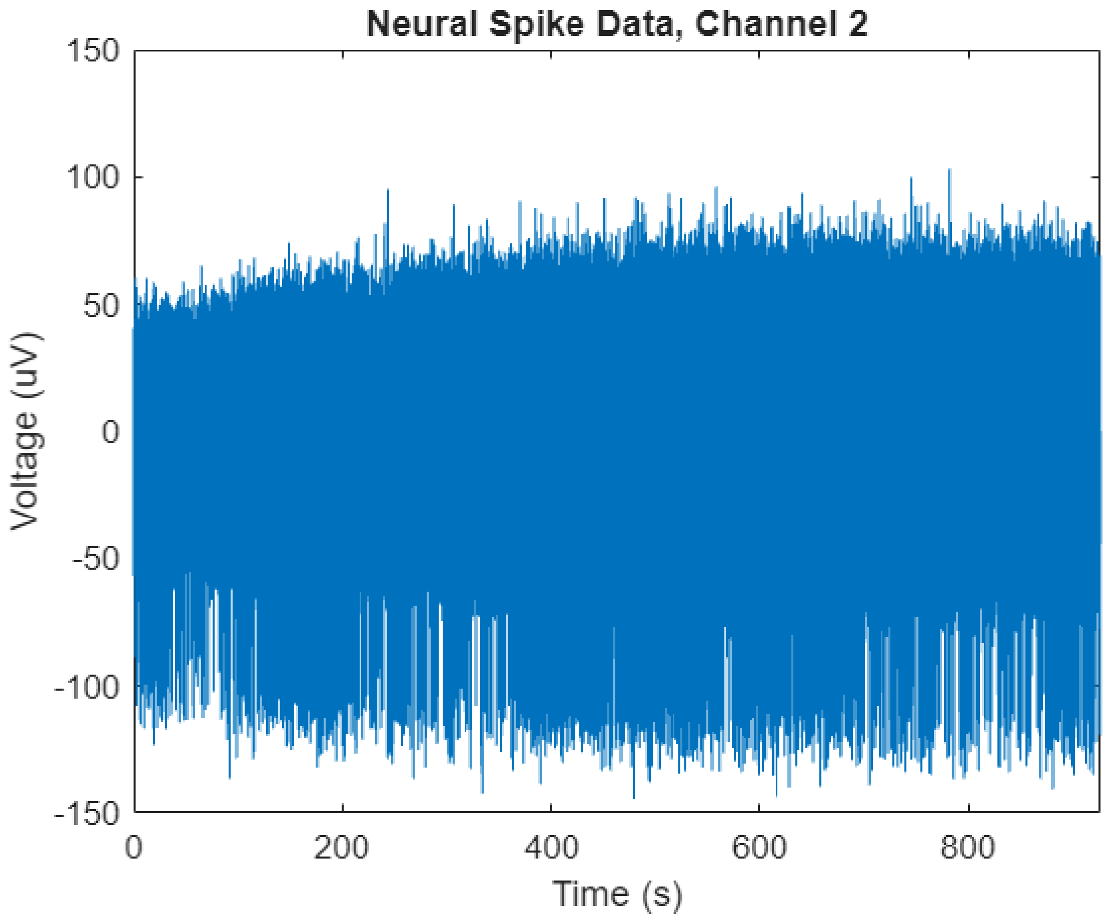

Figure 3. Raw Neural Spike Data, Channel 2

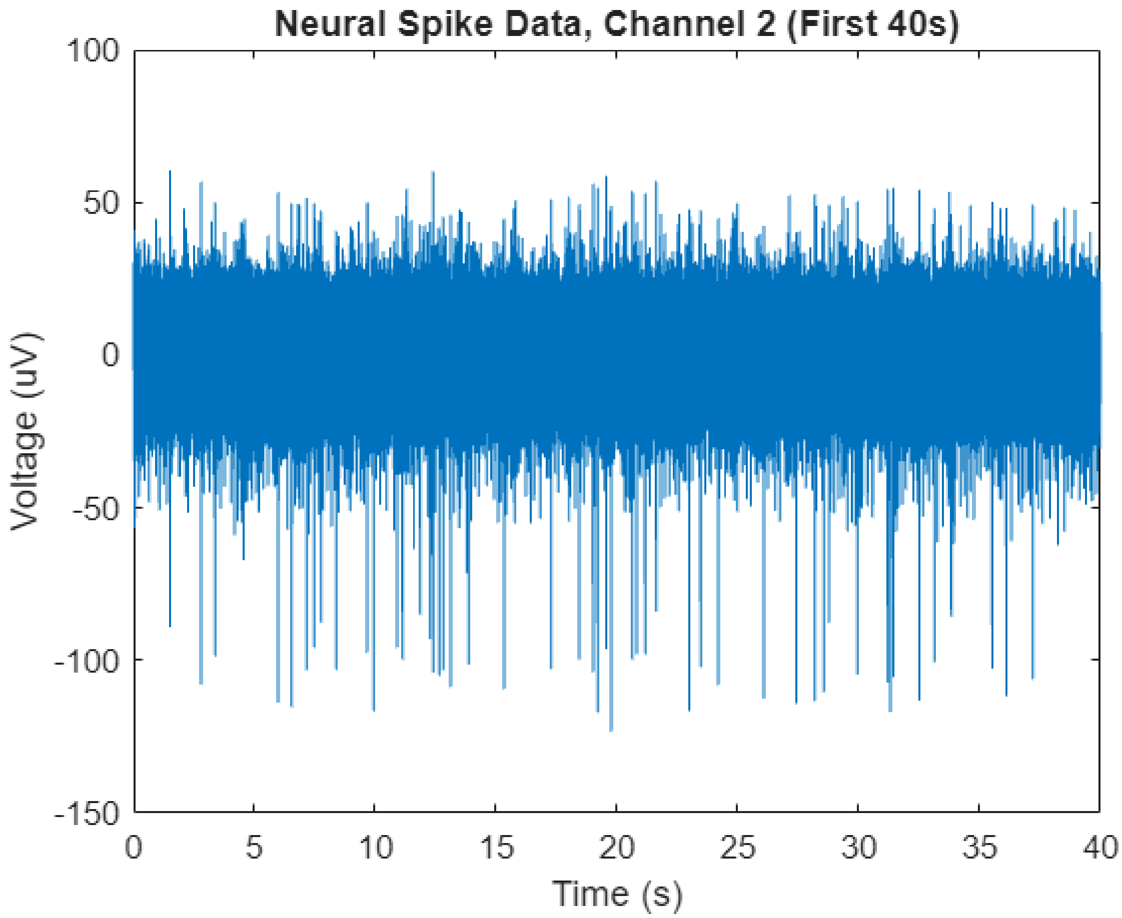

Figure 4. Raw Neural Spike Data, Channel 2 (First 40 Seconds)

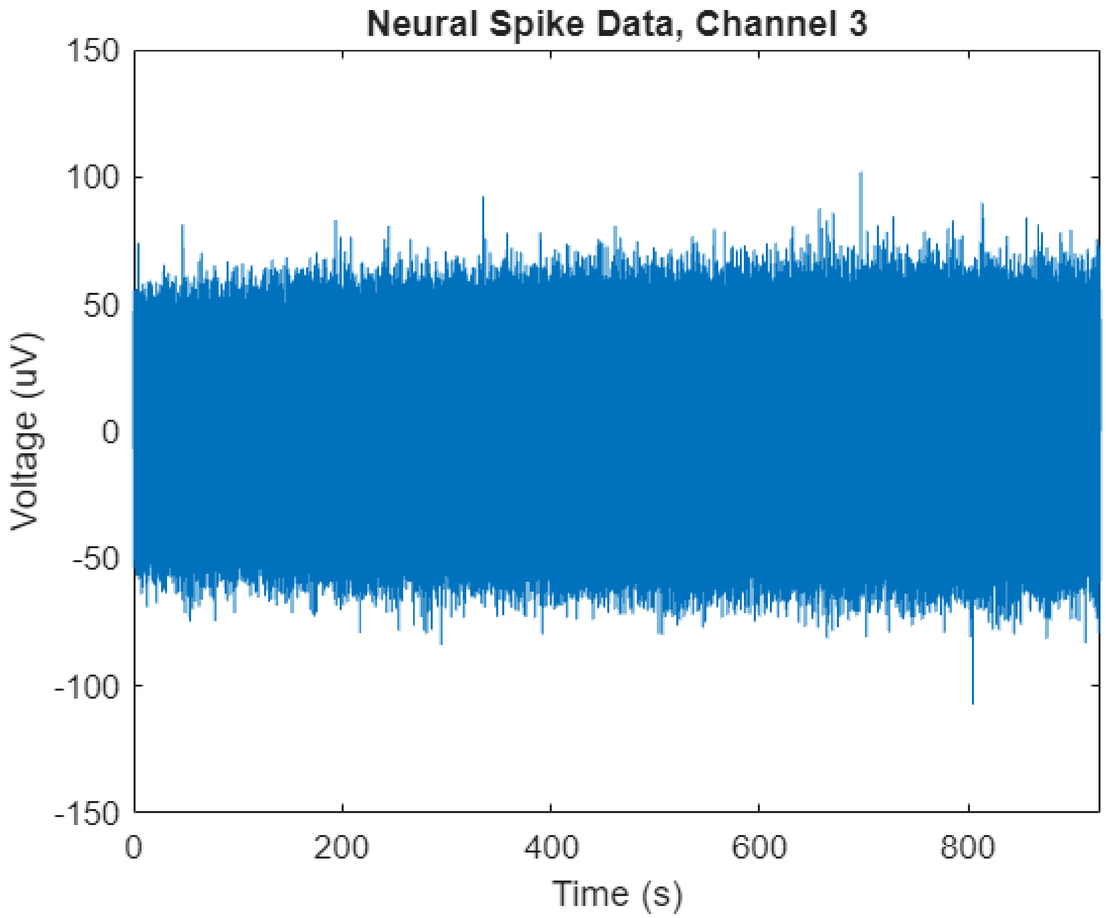

Figure 5. Raw Neural Spike Data, Channel 3

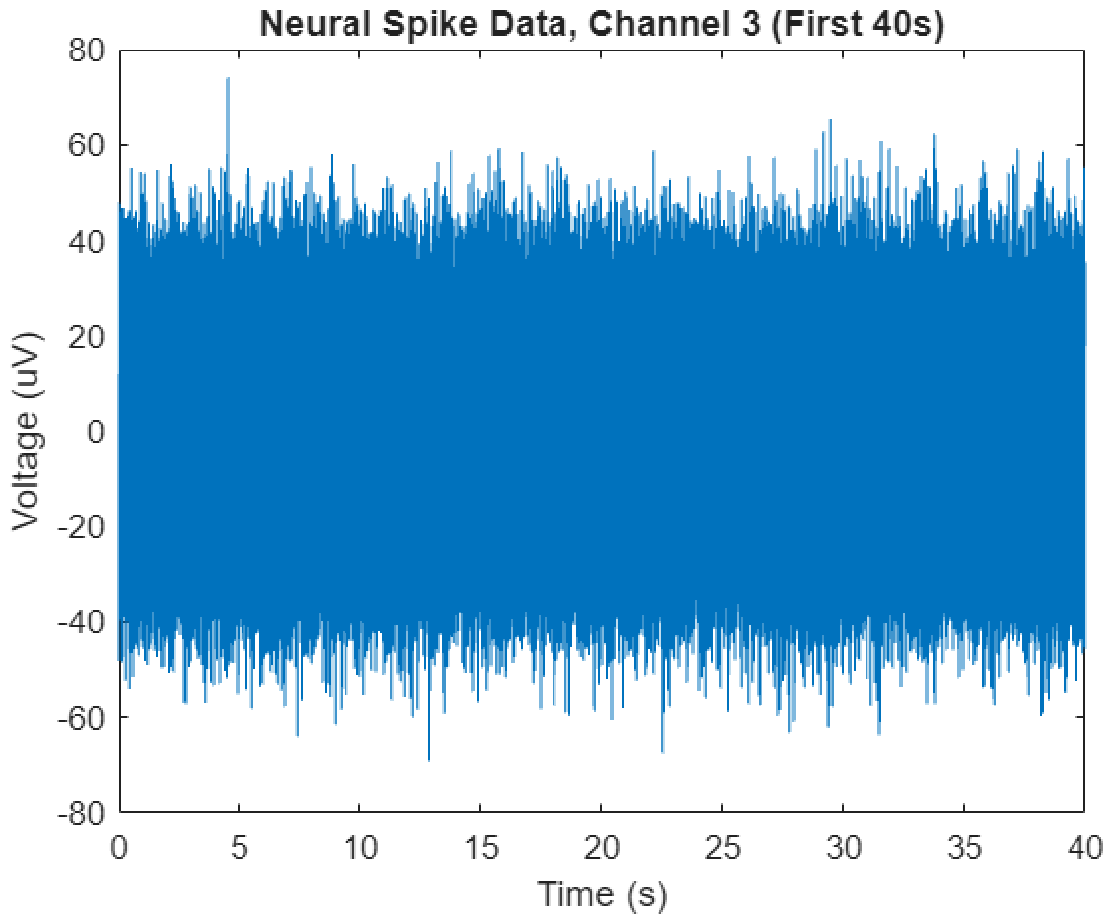

Figure 6. Raw Neural Spike Data, Channel 3 (First 40 Seconds)

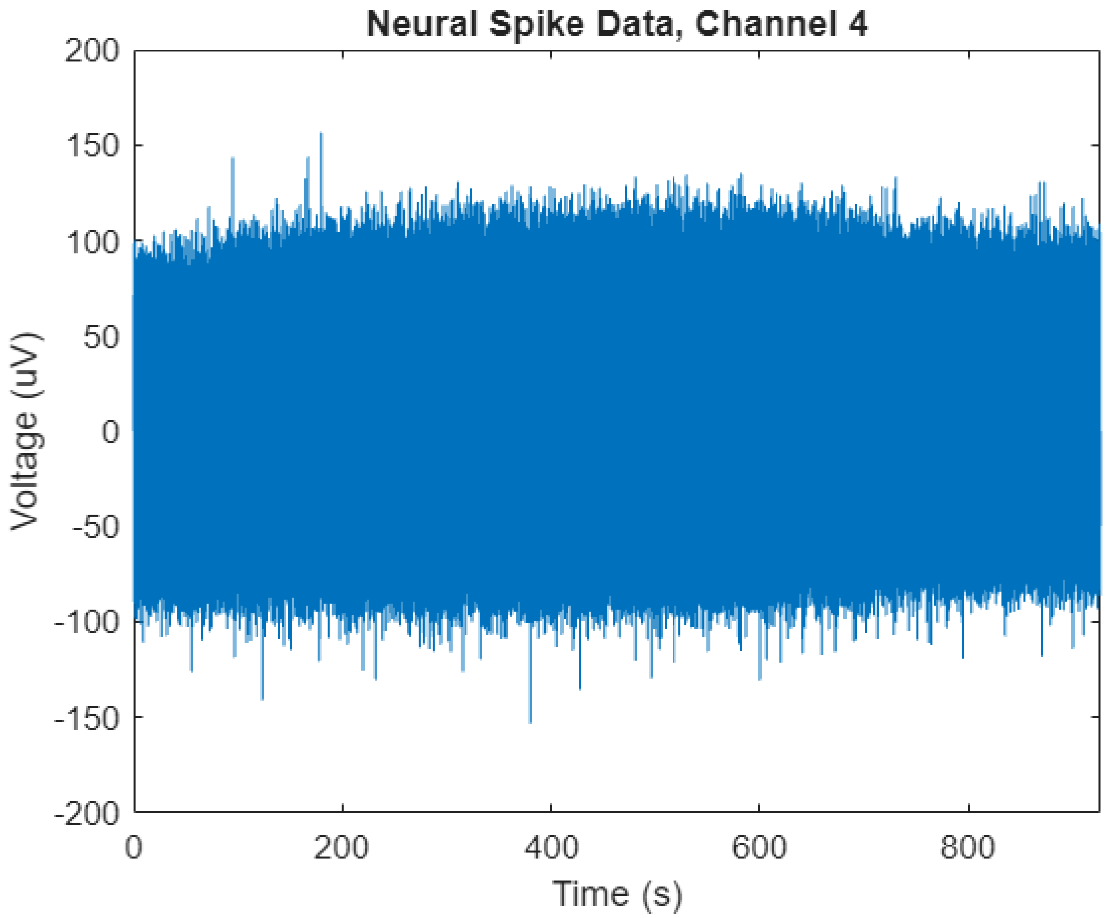

Figure 7. Raw Neural Spike Data, Channel 4

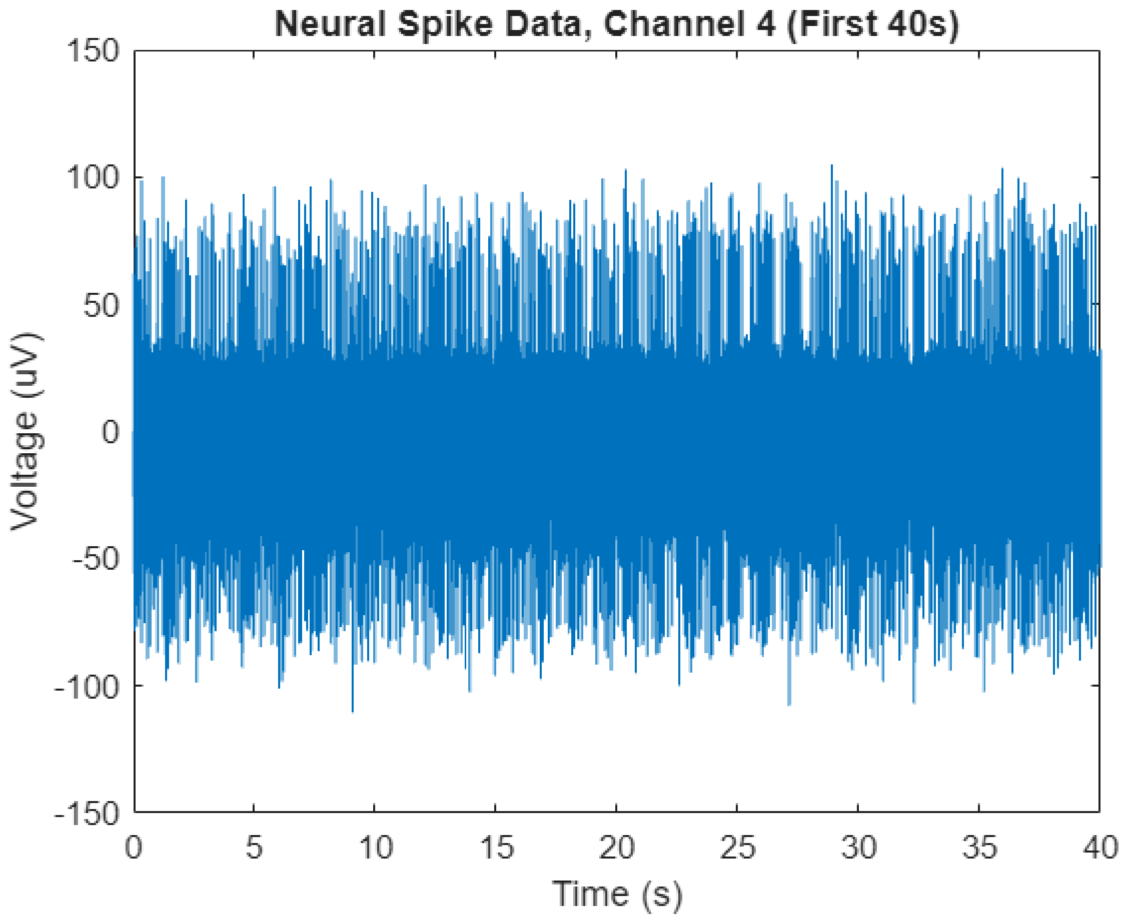

Figure 8. Raw Neural Spike Data, Channel 4 (First 40 Seconds)

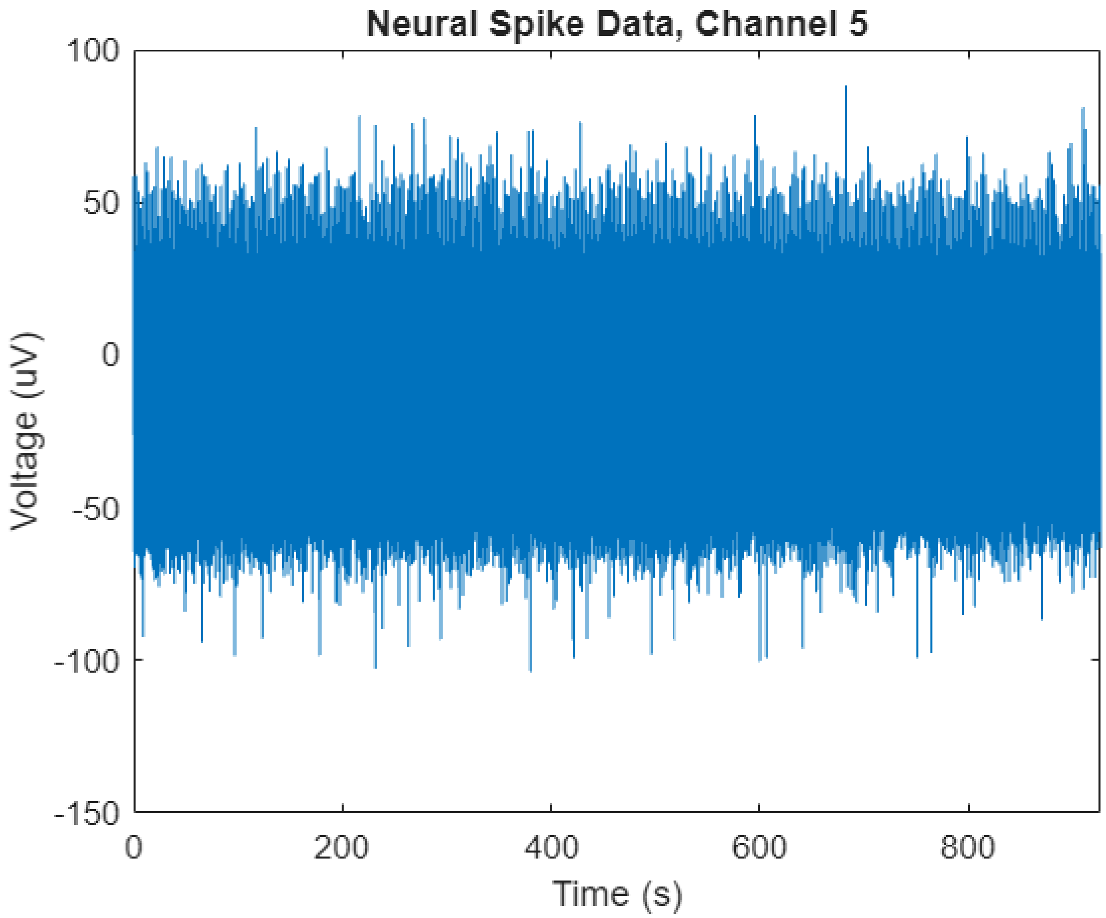

Figure 9. Raw Neural Spike Data, Channel 5 (First 40 Seconds)

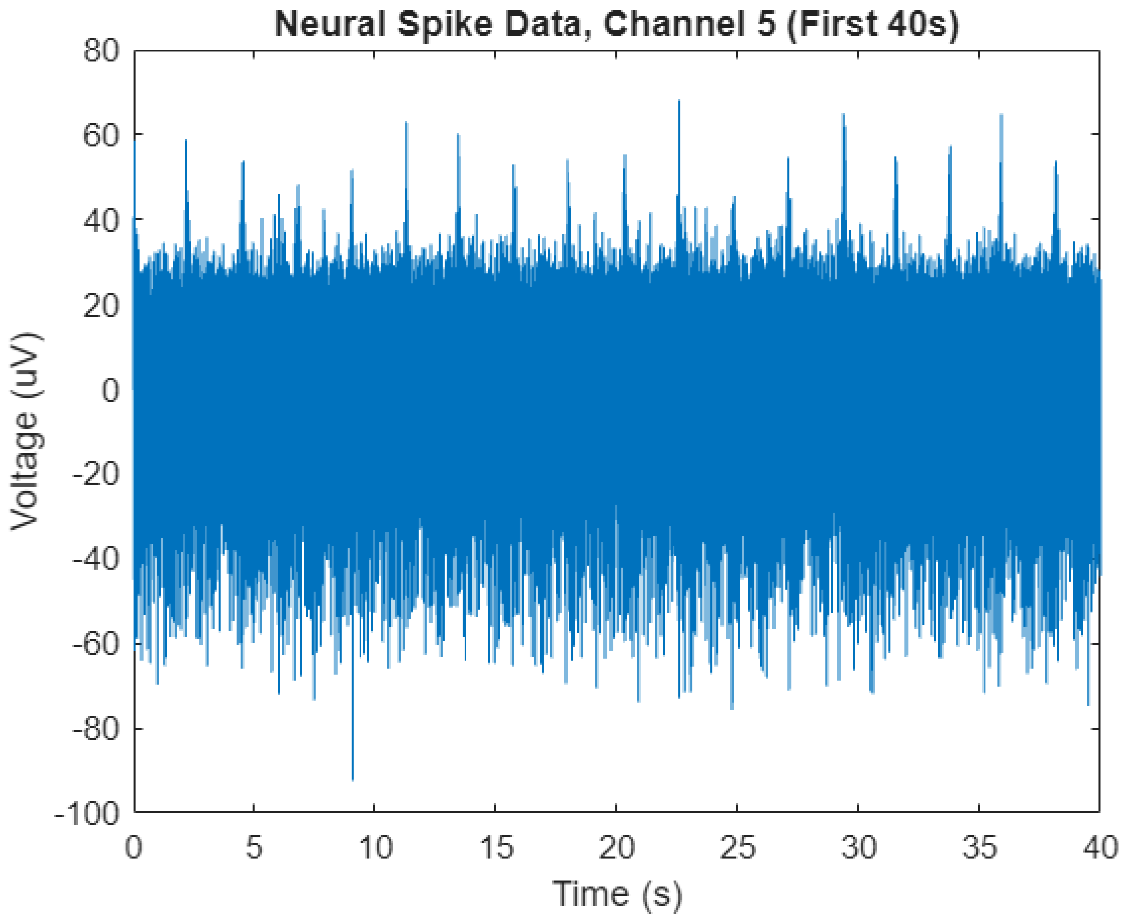

Figure 10. Raw Neural Spike Data, Channel 5 (First 40 Seconds)

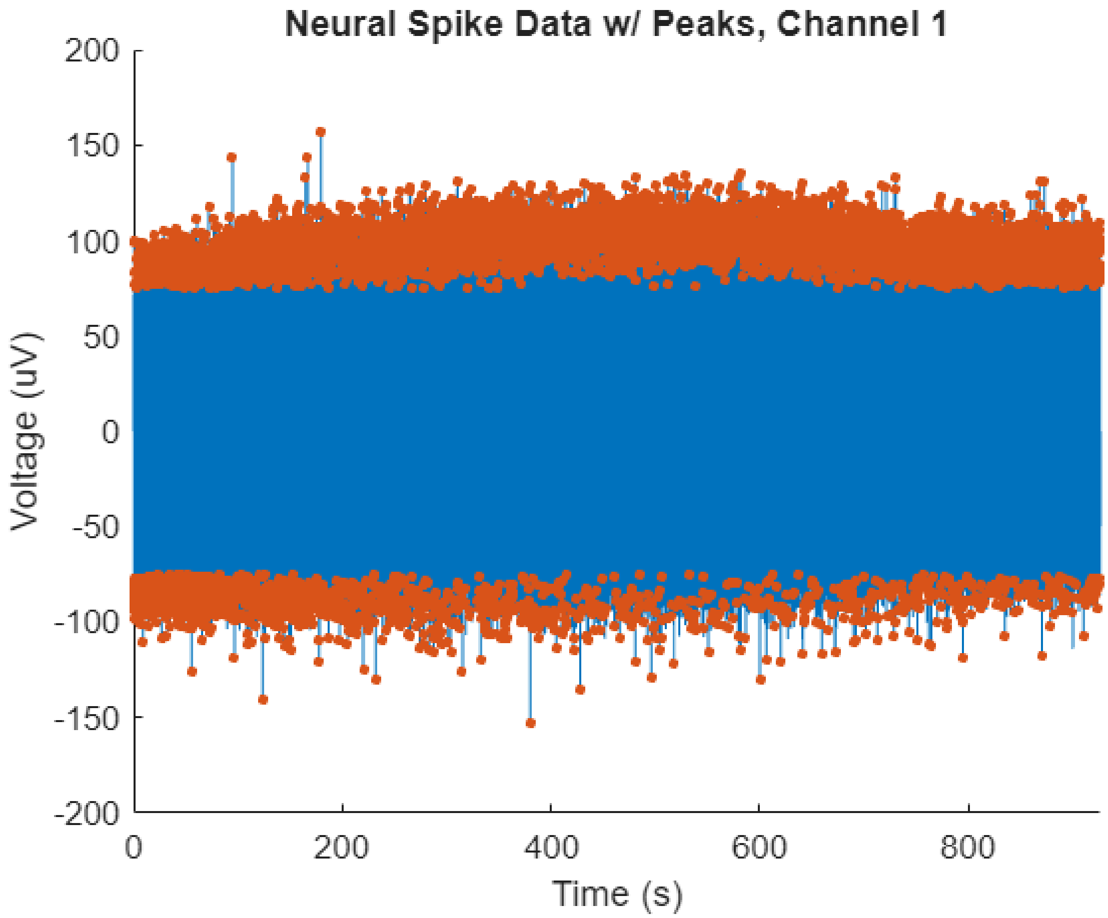

Figure 11. Raw Neural Spike Data Plotted With Spike Peaks, Channel 1

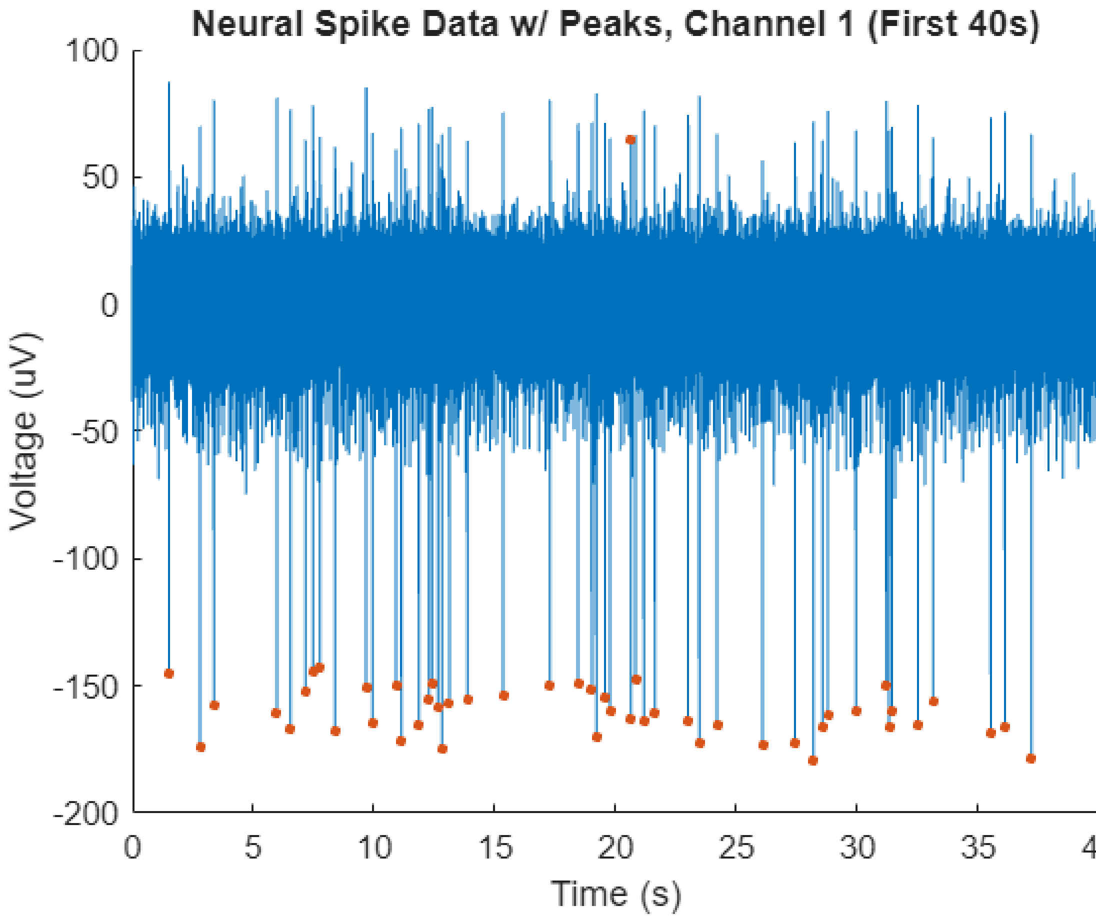

Figure 12. Raw Neural Spike Data Plotted With Spike Peaks, Channel 1 (First 40 Seconds)

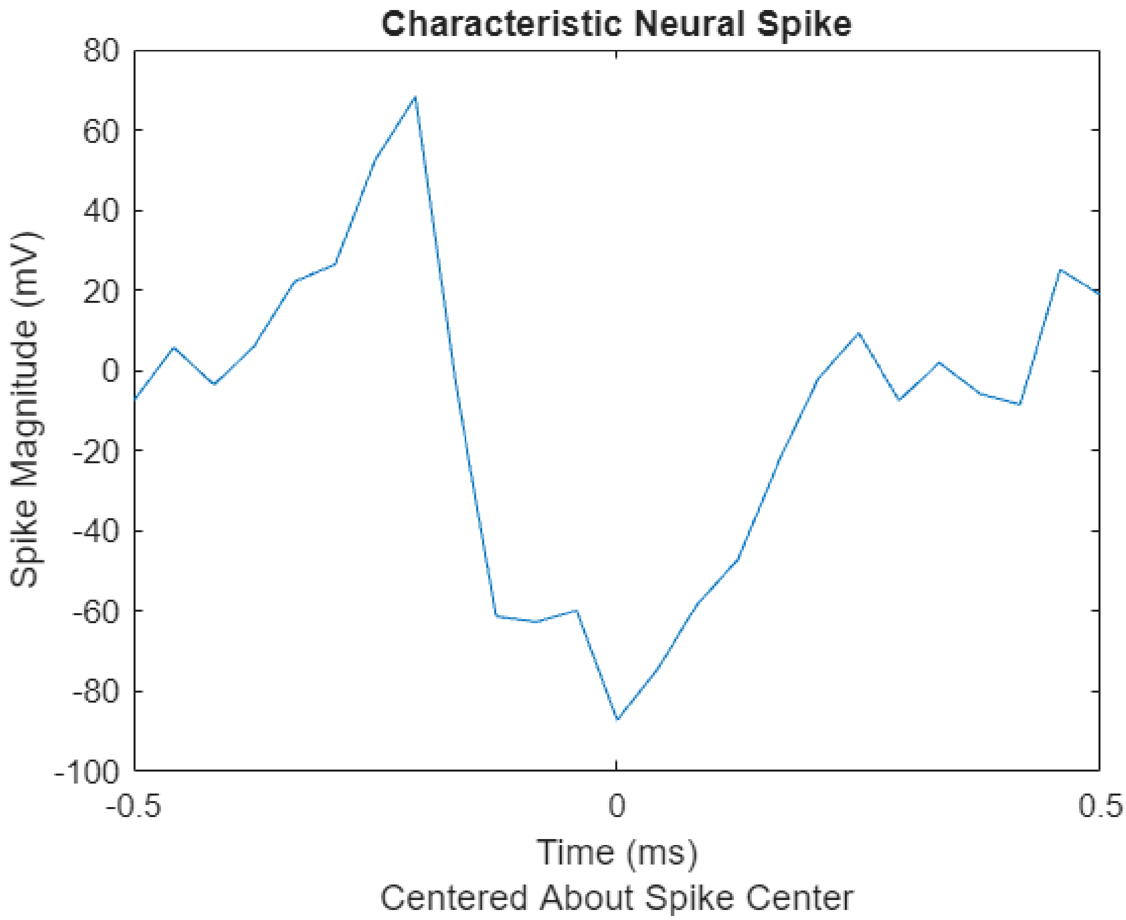

Figure 13. Characteristic Neural Spike. Pulled from the 103rd Spike of Channel 4

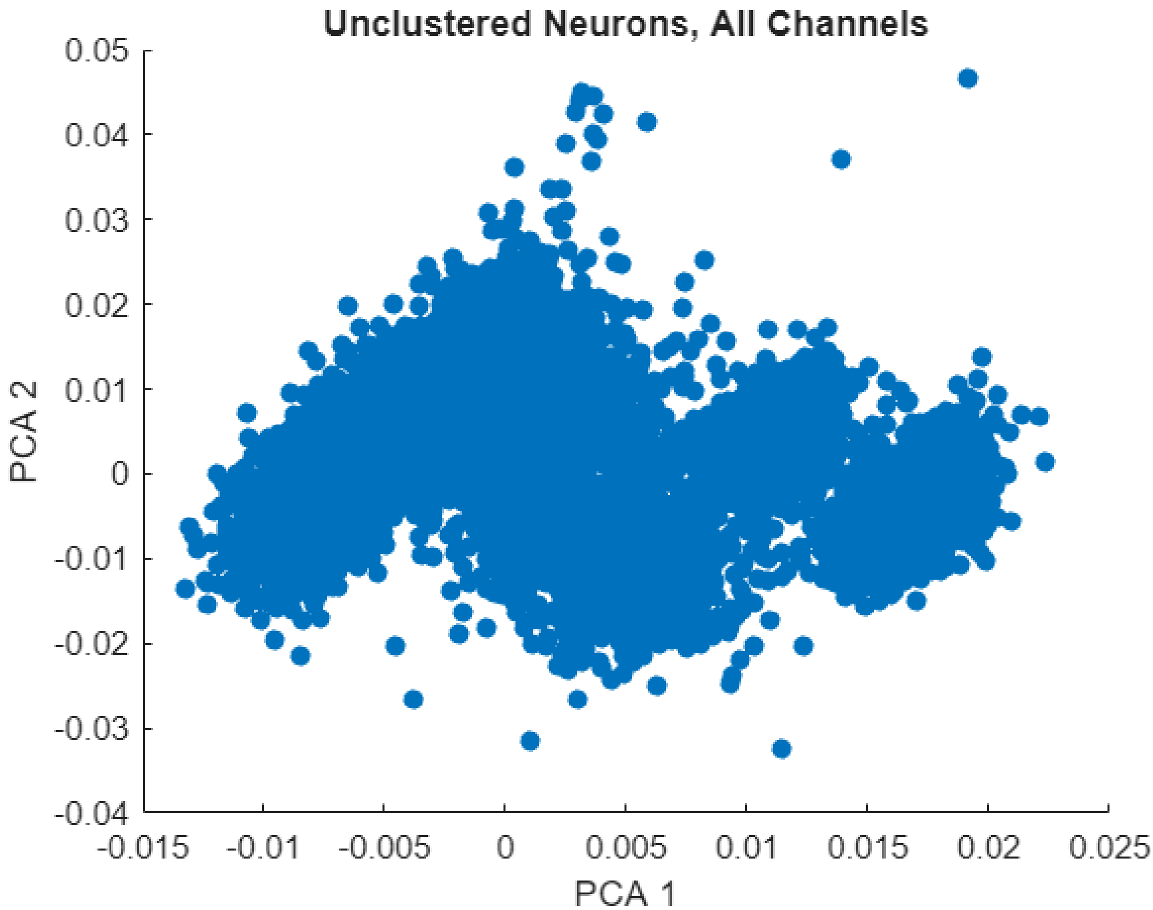

Figure 14. Unclustered Neural Spikes, All Channels

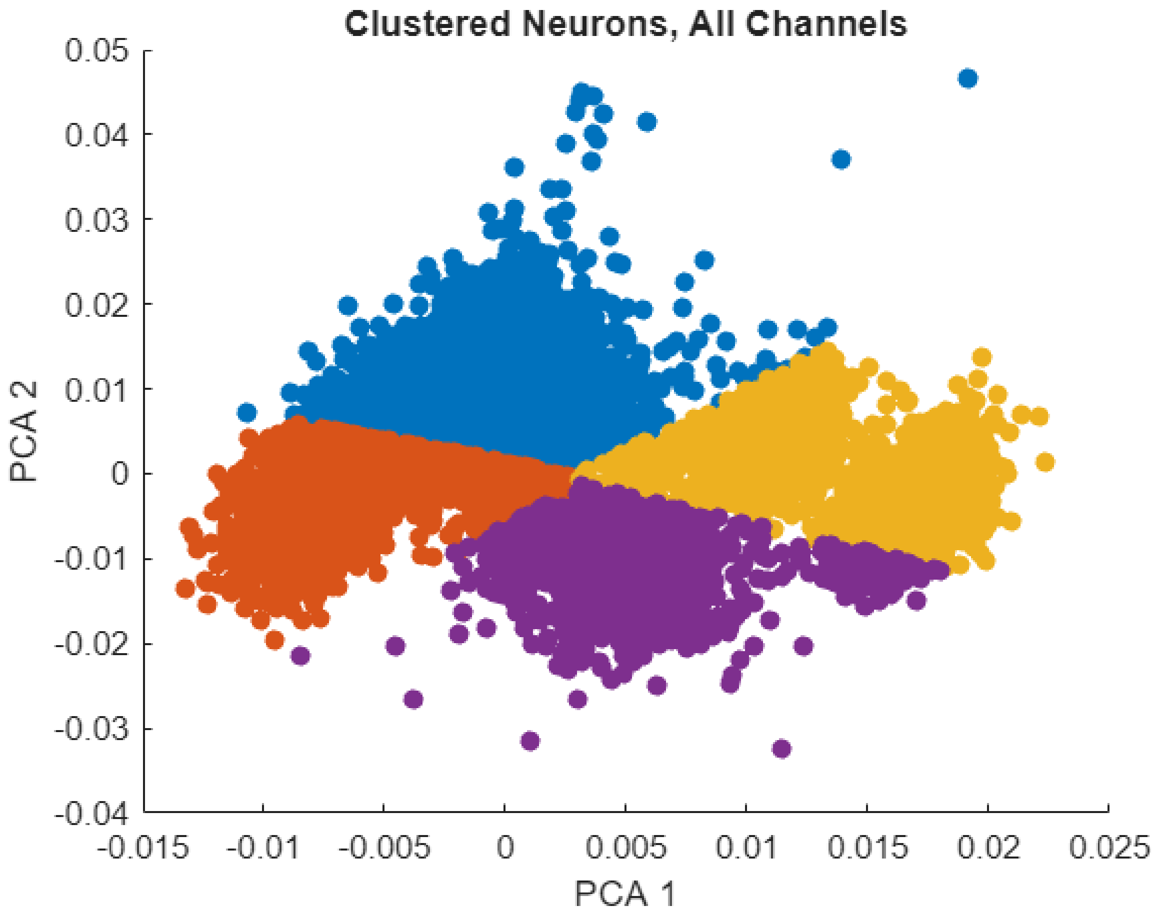

Figure 15. Clustered Neural Spikes, All Channels

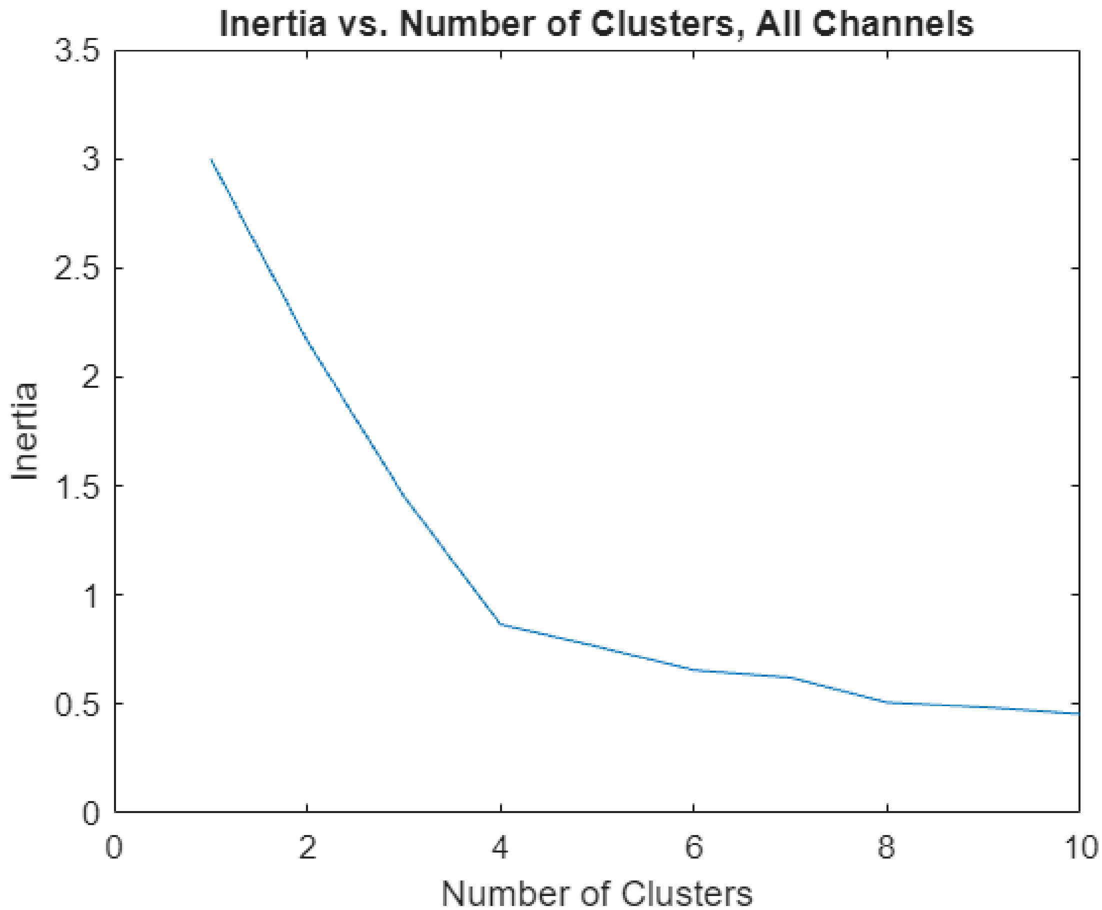

Figure 16. Elbow Plot for k-means Clustering, All Channels

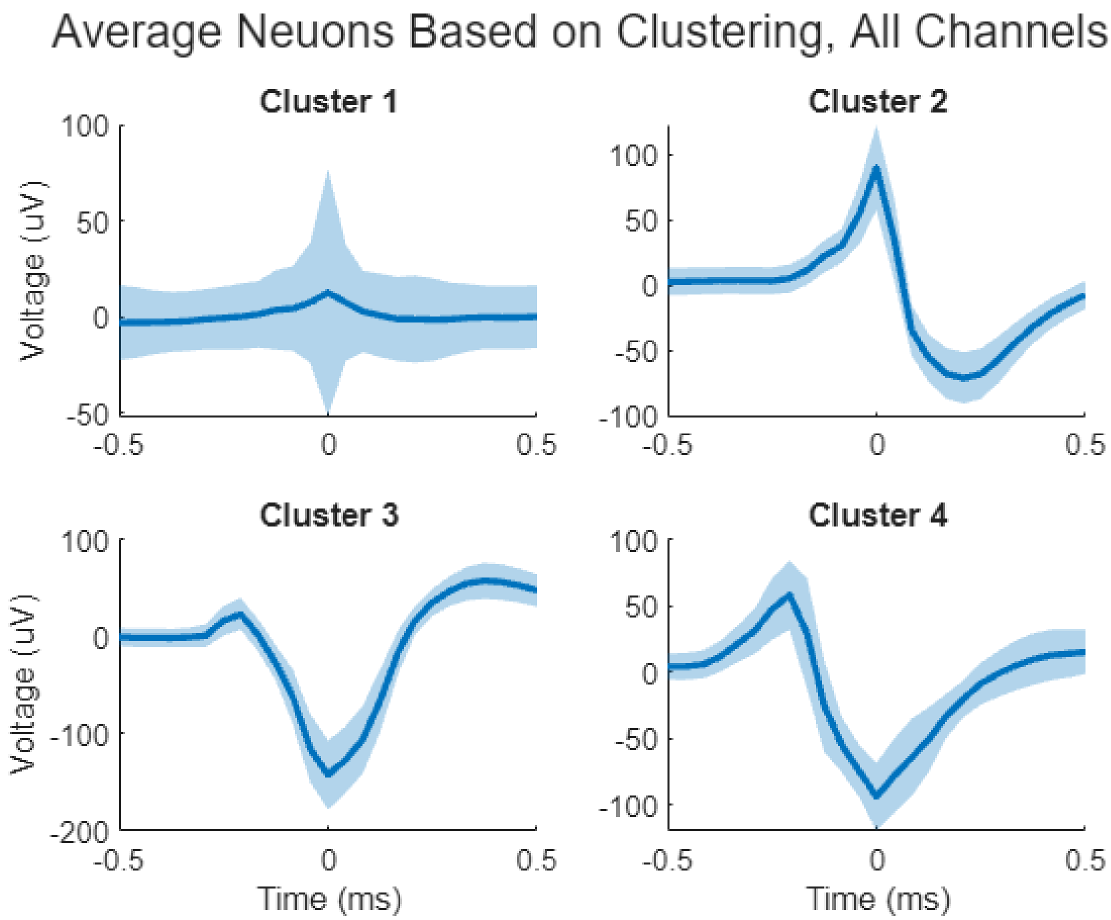

Figure 17. Average Neural Spikes, Based on k-means Clustering Results, All Channels

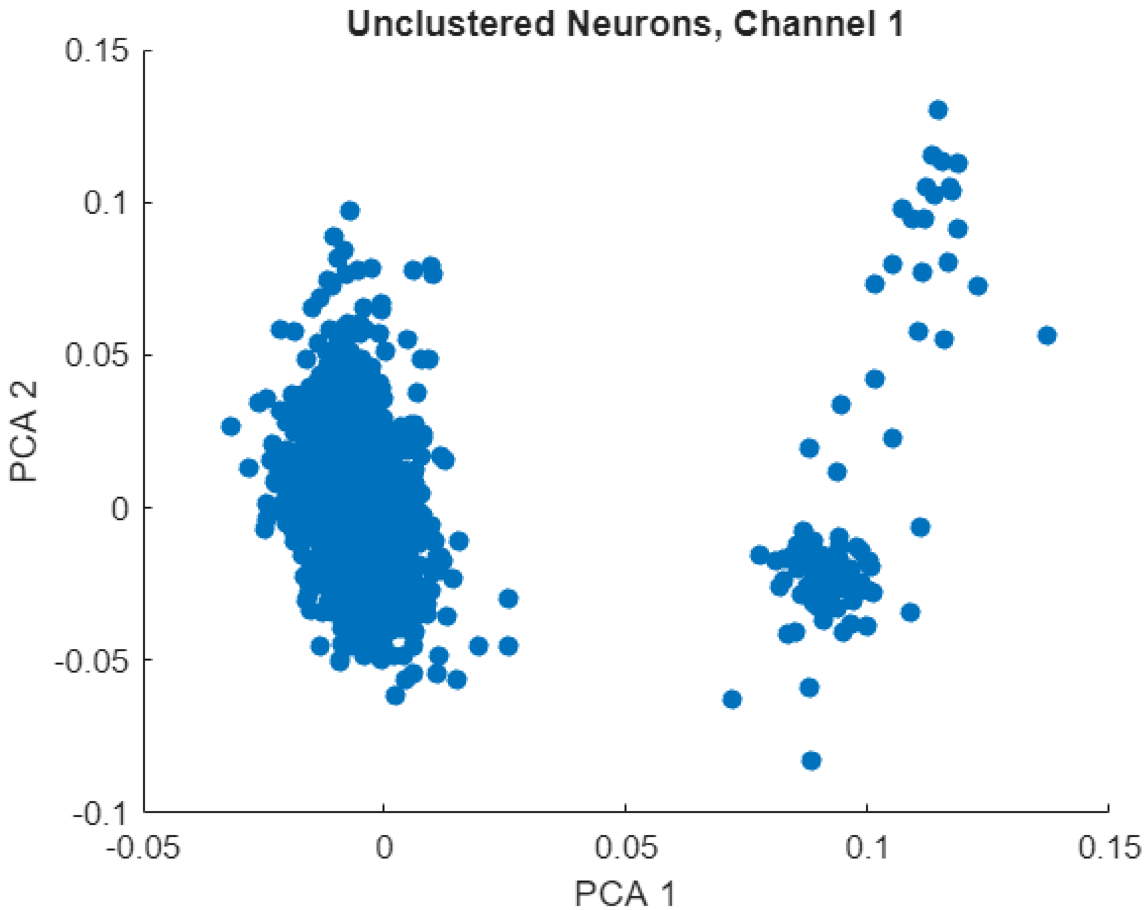

Figure 18. Unclustered Neural Spikes, Channel 1

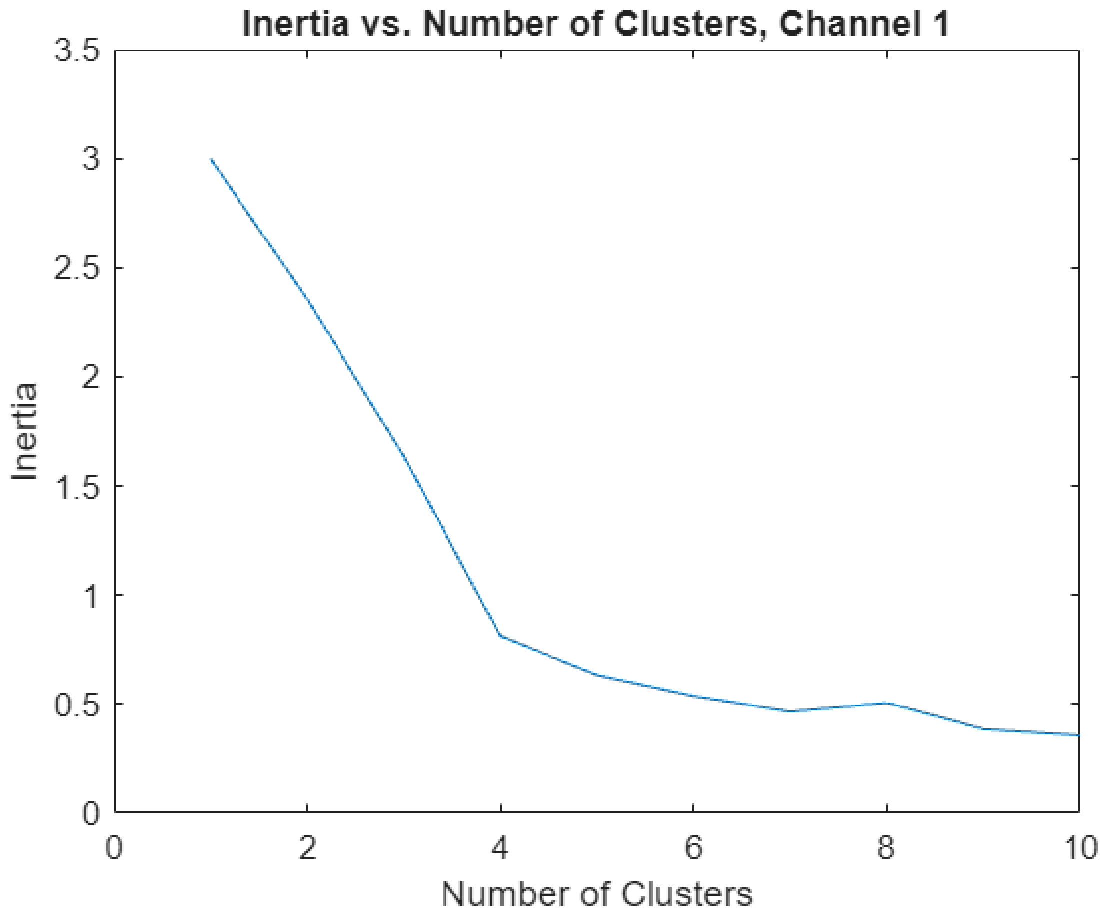

Figure 19. Elbow Plot for k-means Clustering, Channel 1

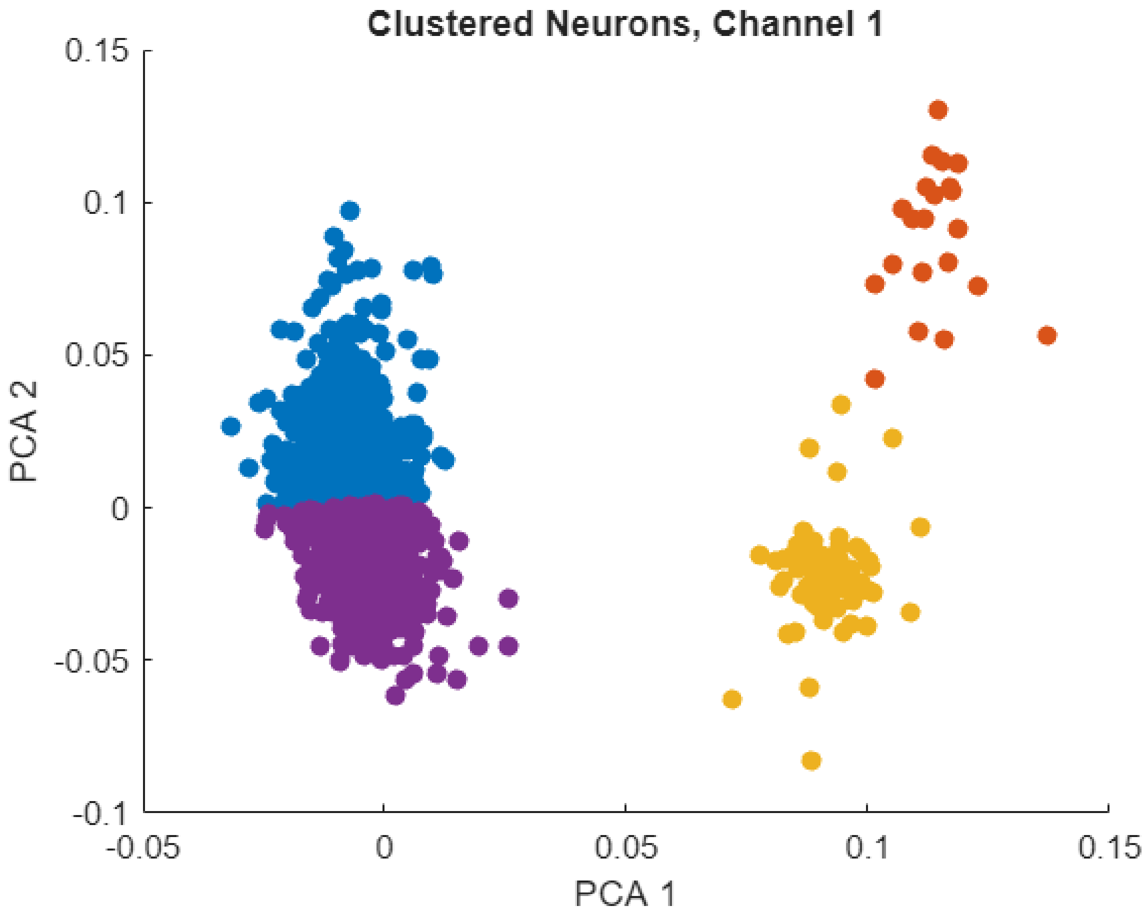

Figure 20. Clustered Neural Spikes, Channel 1

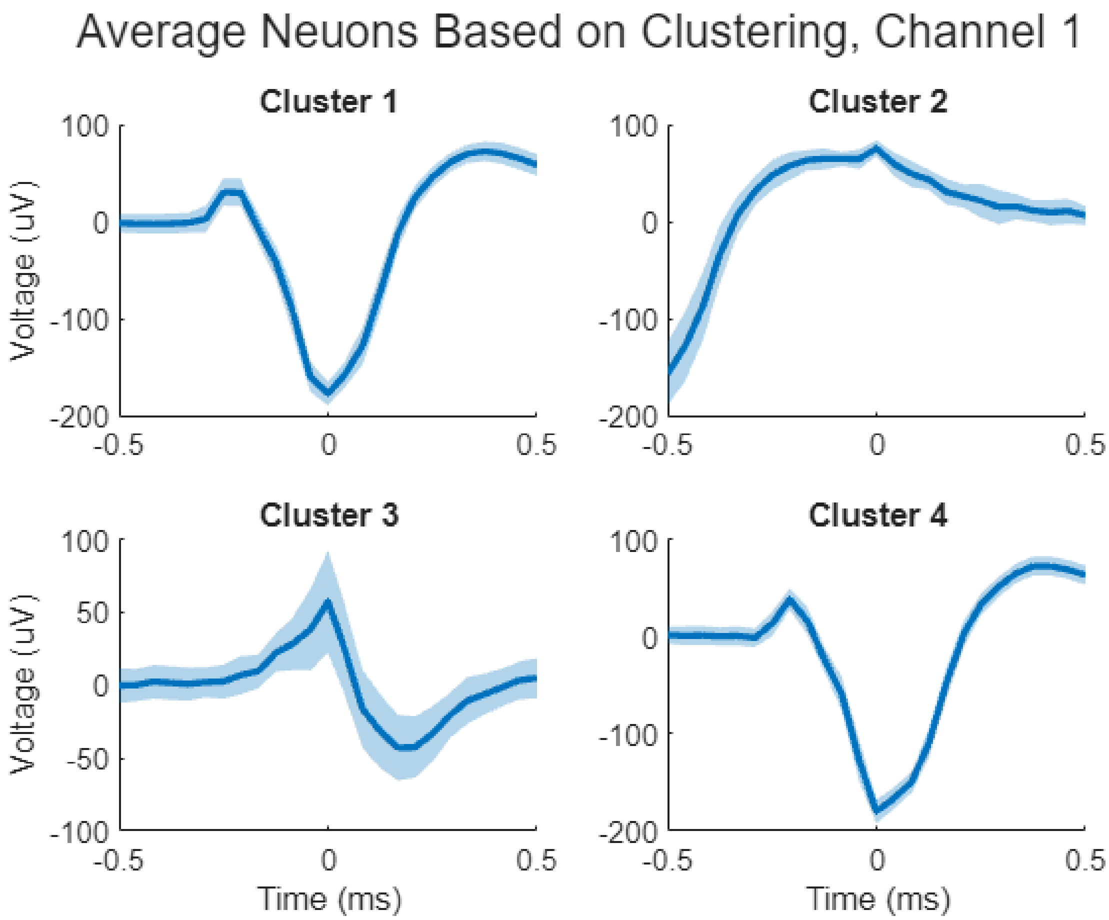

Figure 21. Average Neural Spikes, Based on k-means Clustering Results, Channel 1

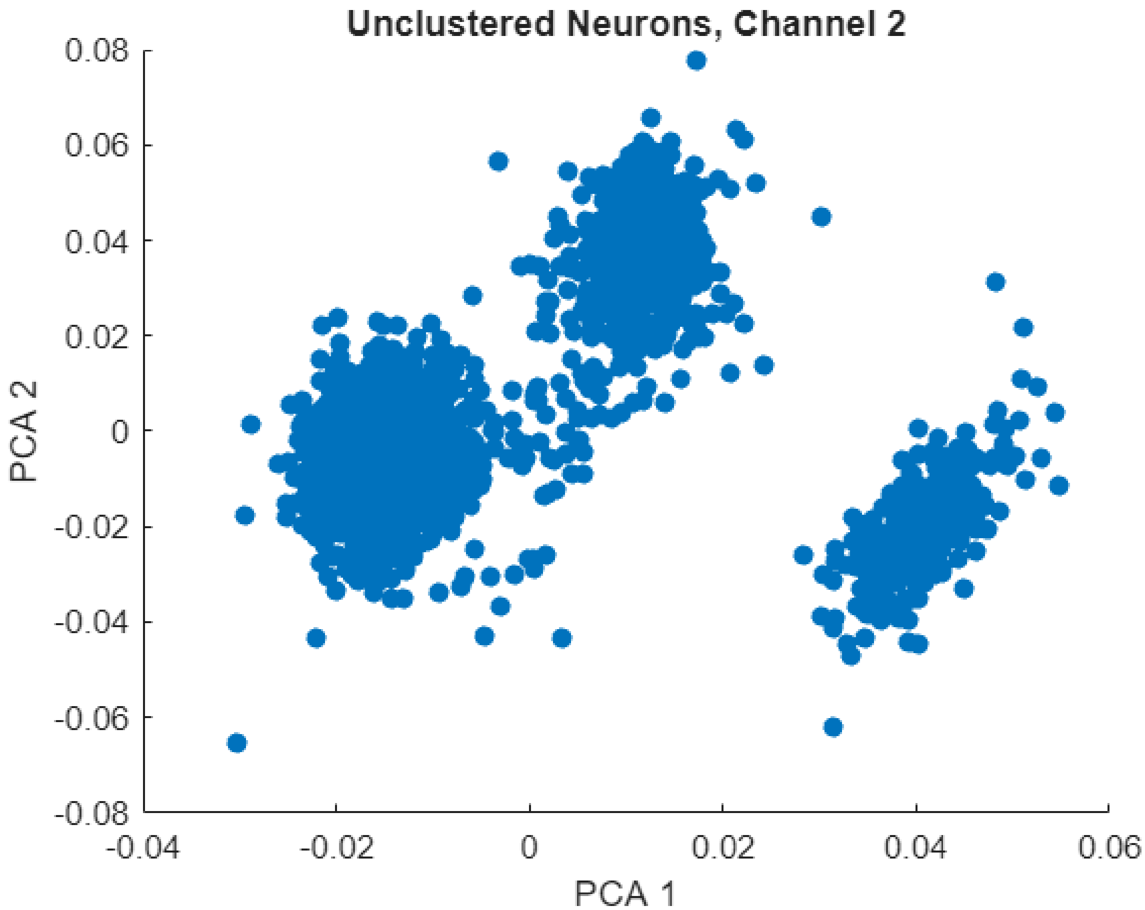

Figure 22. Unclustered Neural Spikes, Channel 2

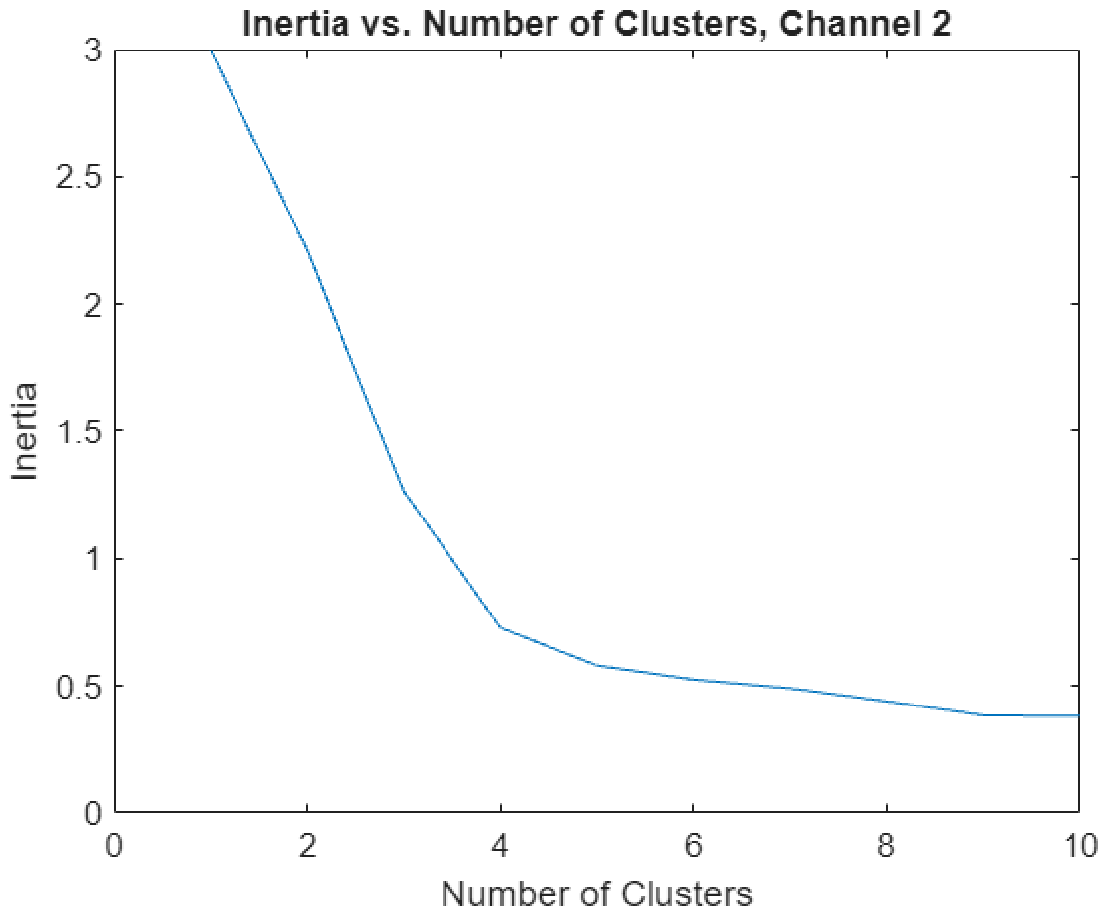

Figure 23. Elbow Plot for k-means Clustering, Channel 2

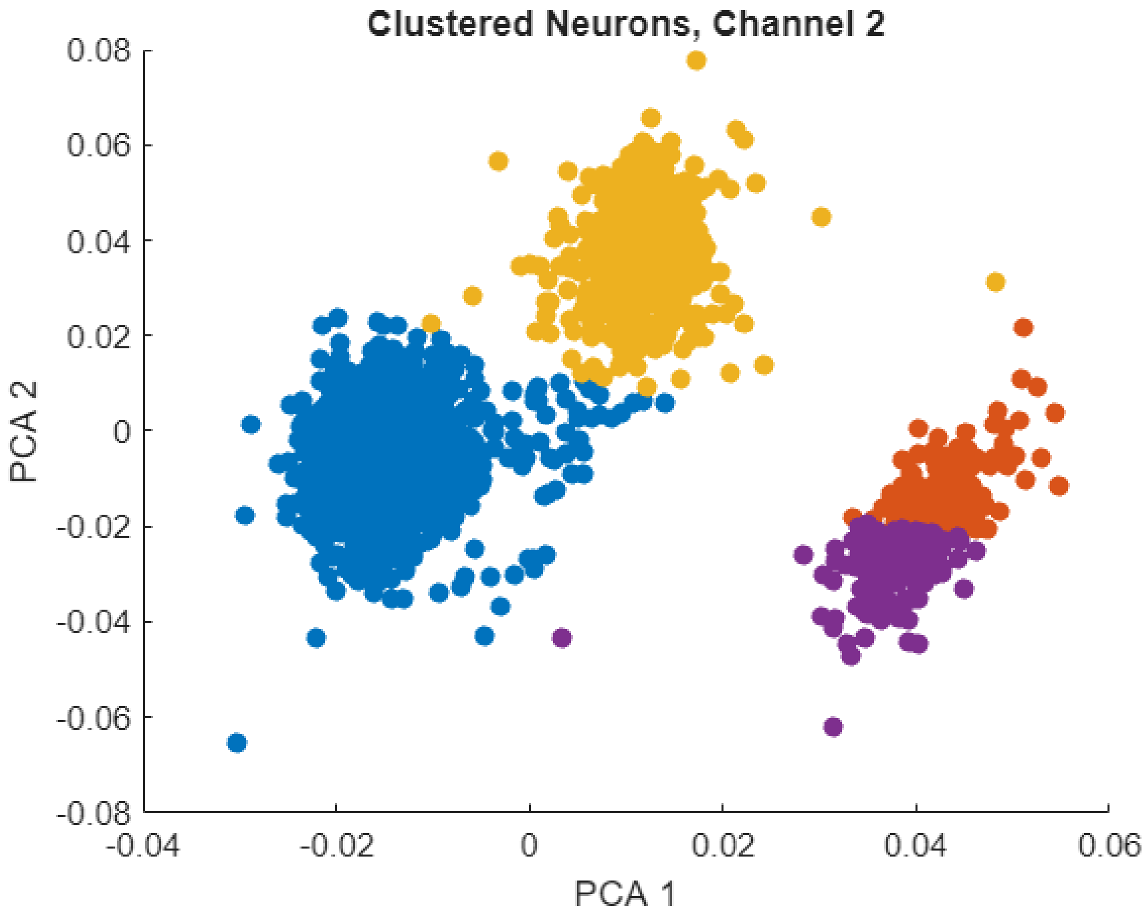

Figure 24. Clustered Neural Spikes, Channel 2

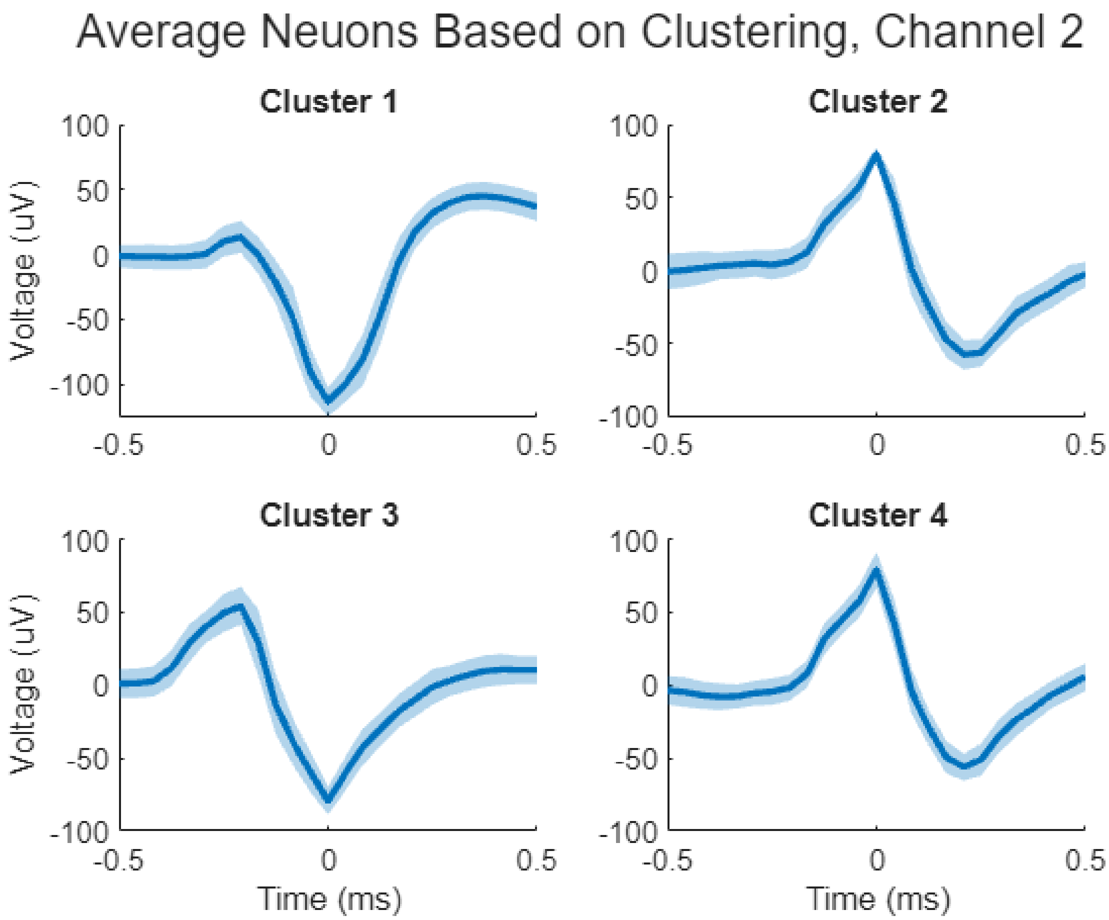

Figure 25. Average Neural Spikes, Based on k-means Clustering Results, Channel 2

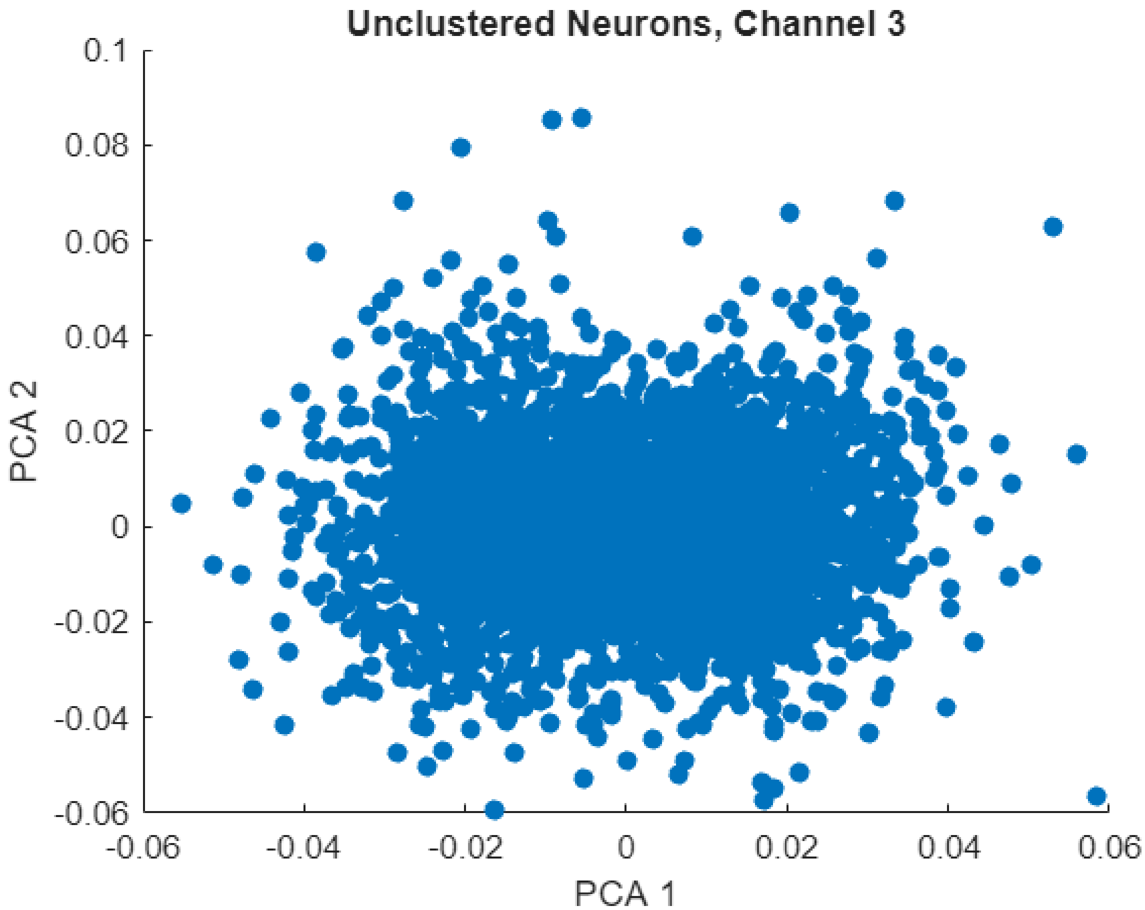

Figure 26. Unclustered Neural Spikes, Channel 3

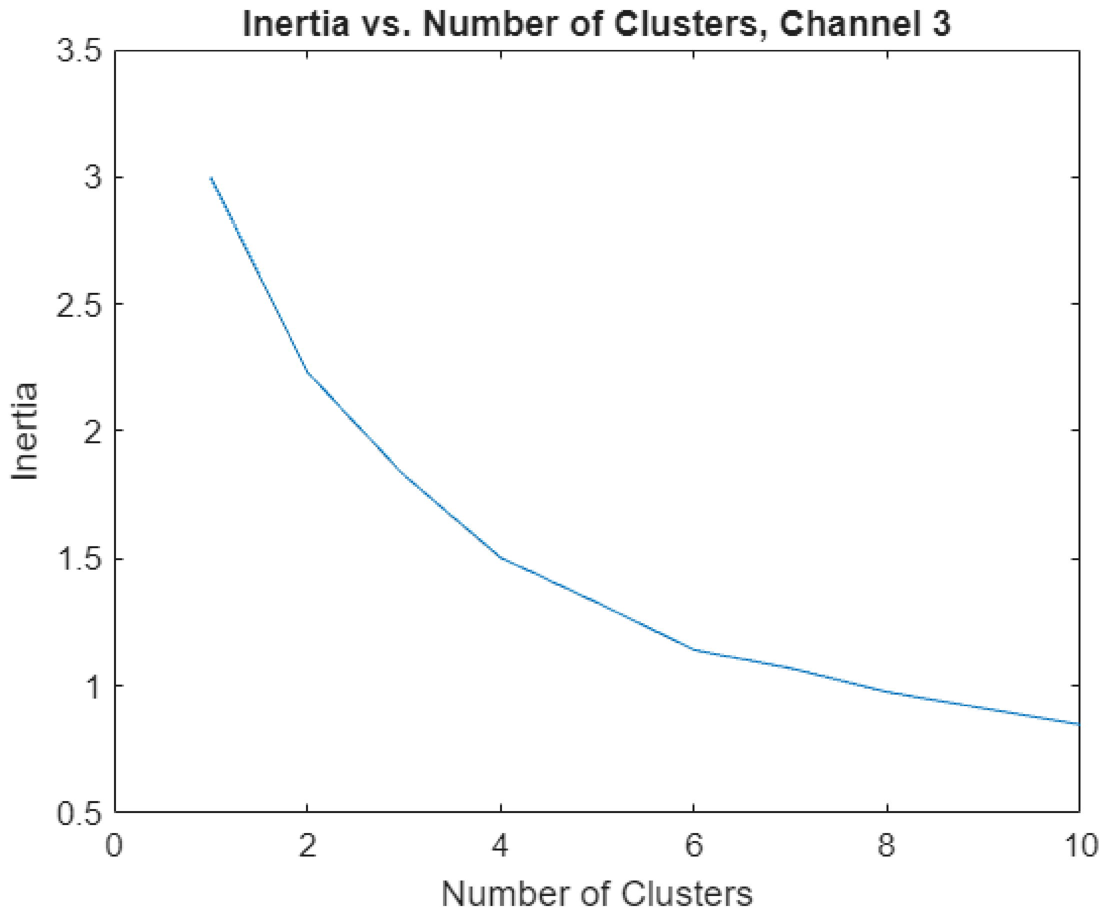

Figure 27. Elbow Plot for k-means Clustering, Channel 3

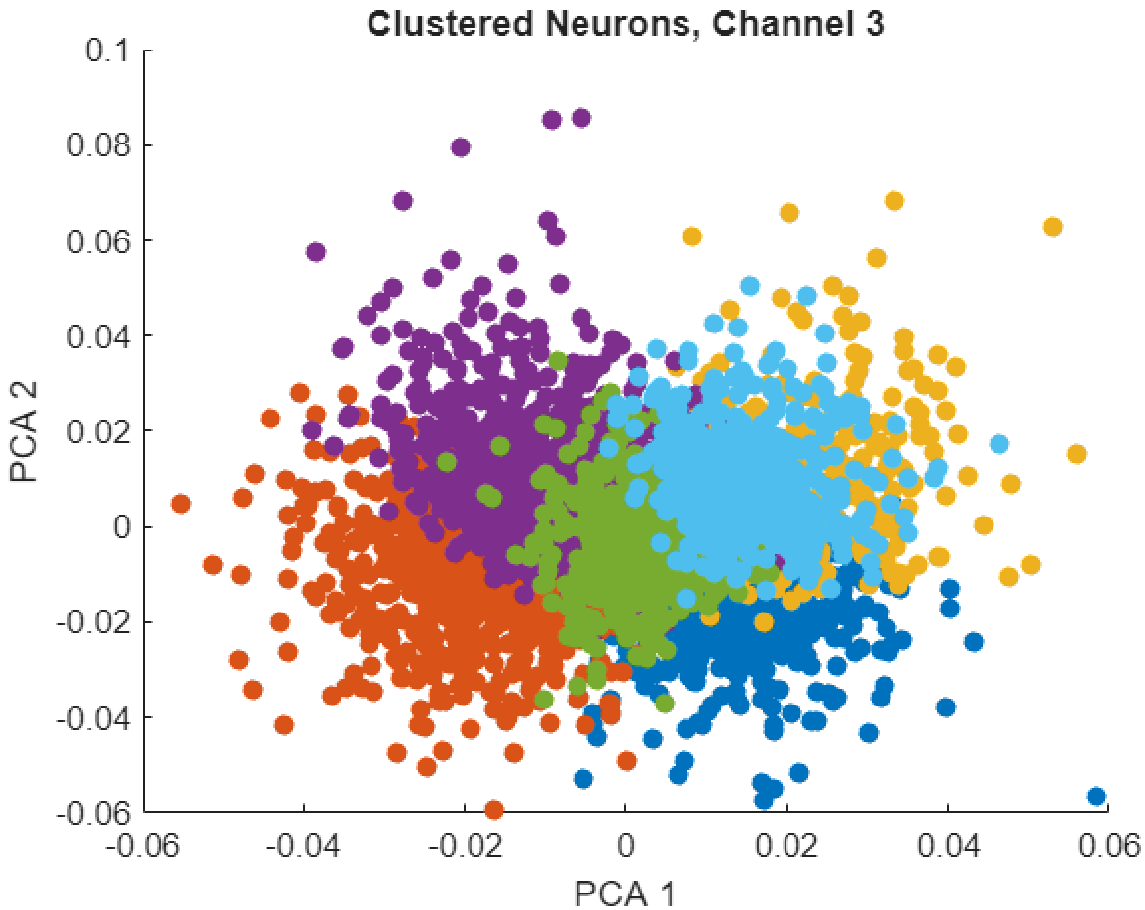

Figure 28. Clustered Neural Spikes, Channel 3

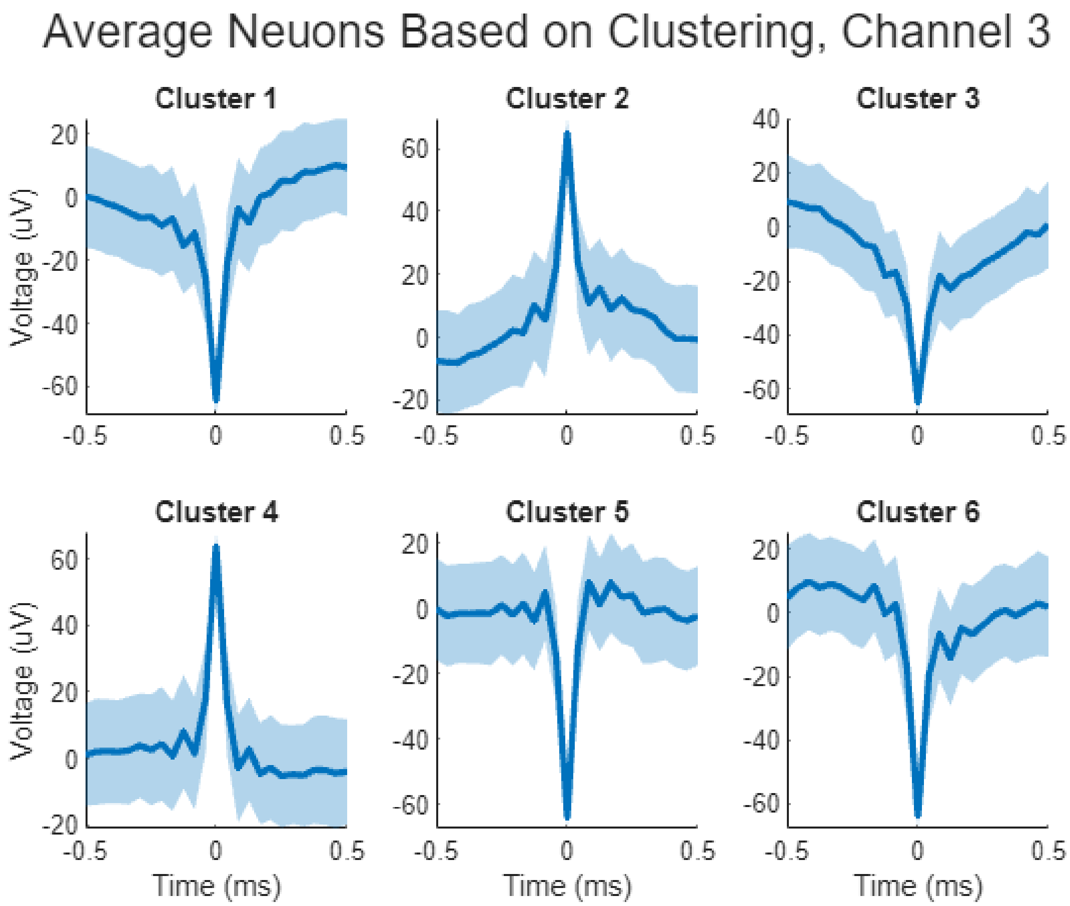

Figure 29. Average Neural Spikes, Based on k-means Clustering Results, Channel 3

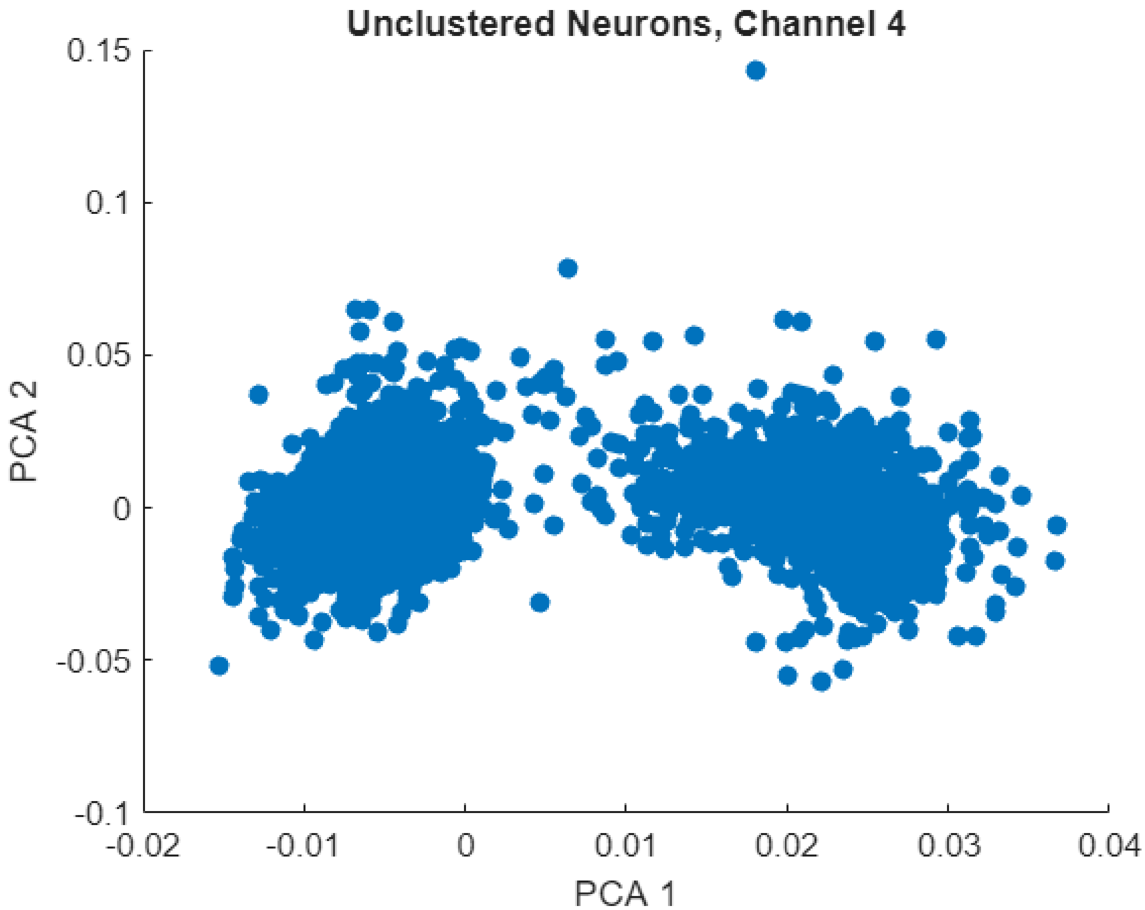

Figure 30. Unclustered Neural Spikes, Channel 4

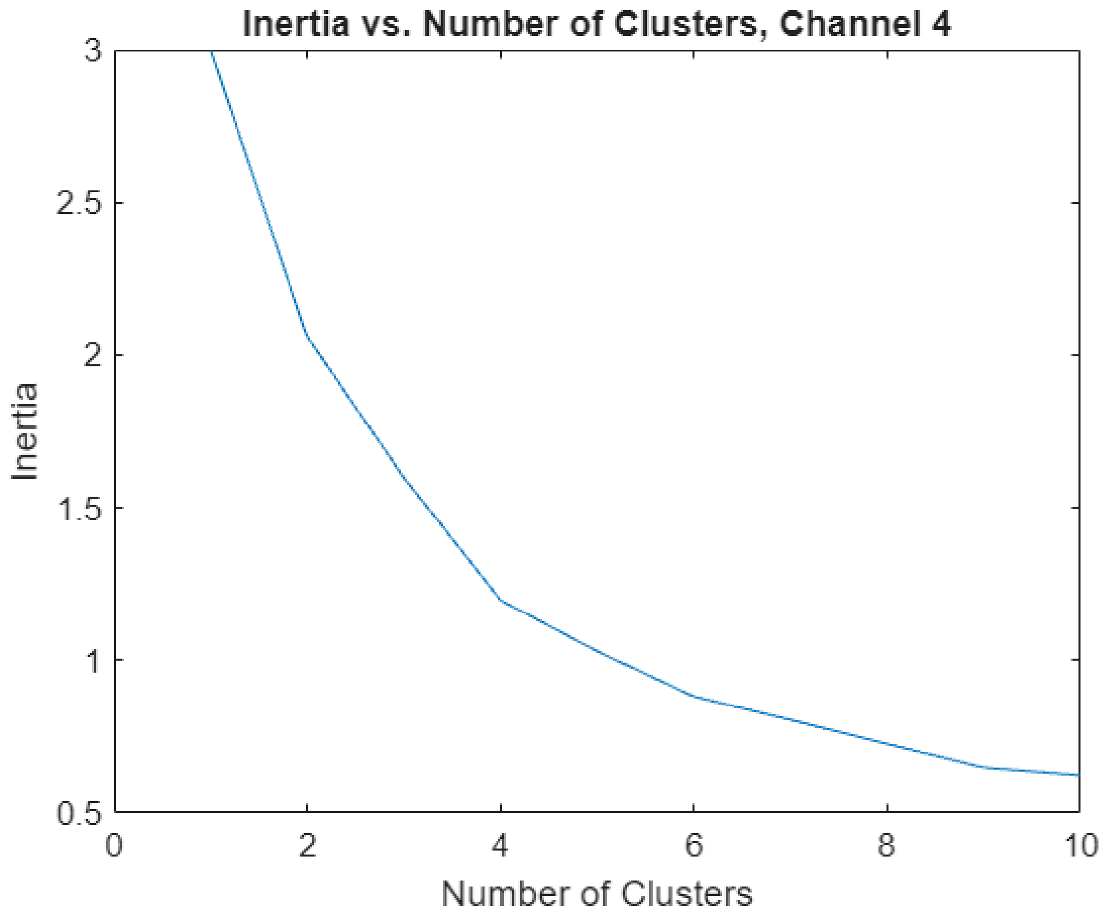

Figure 31. Elbow Plot for k-means Clustering, Channel 4

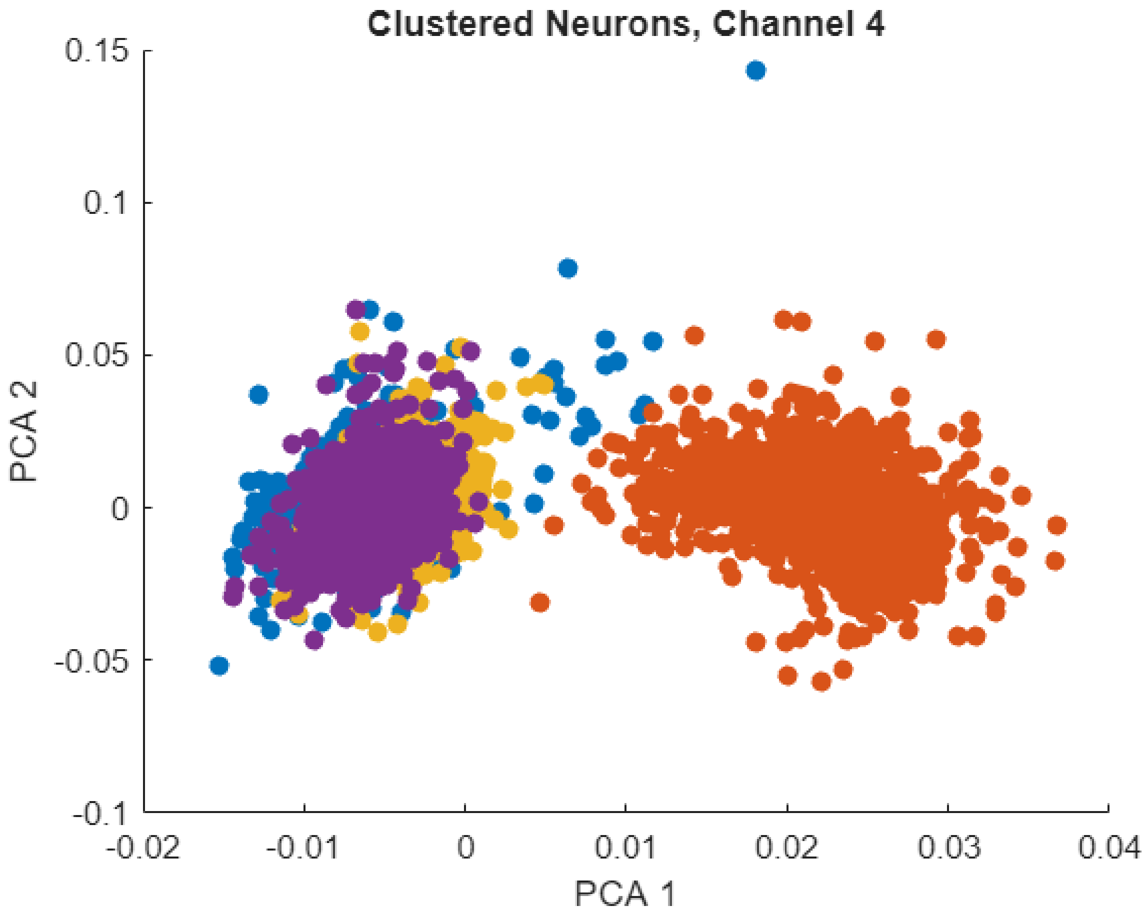

Figure 32. Clustered Neural Spikes, Channel 4

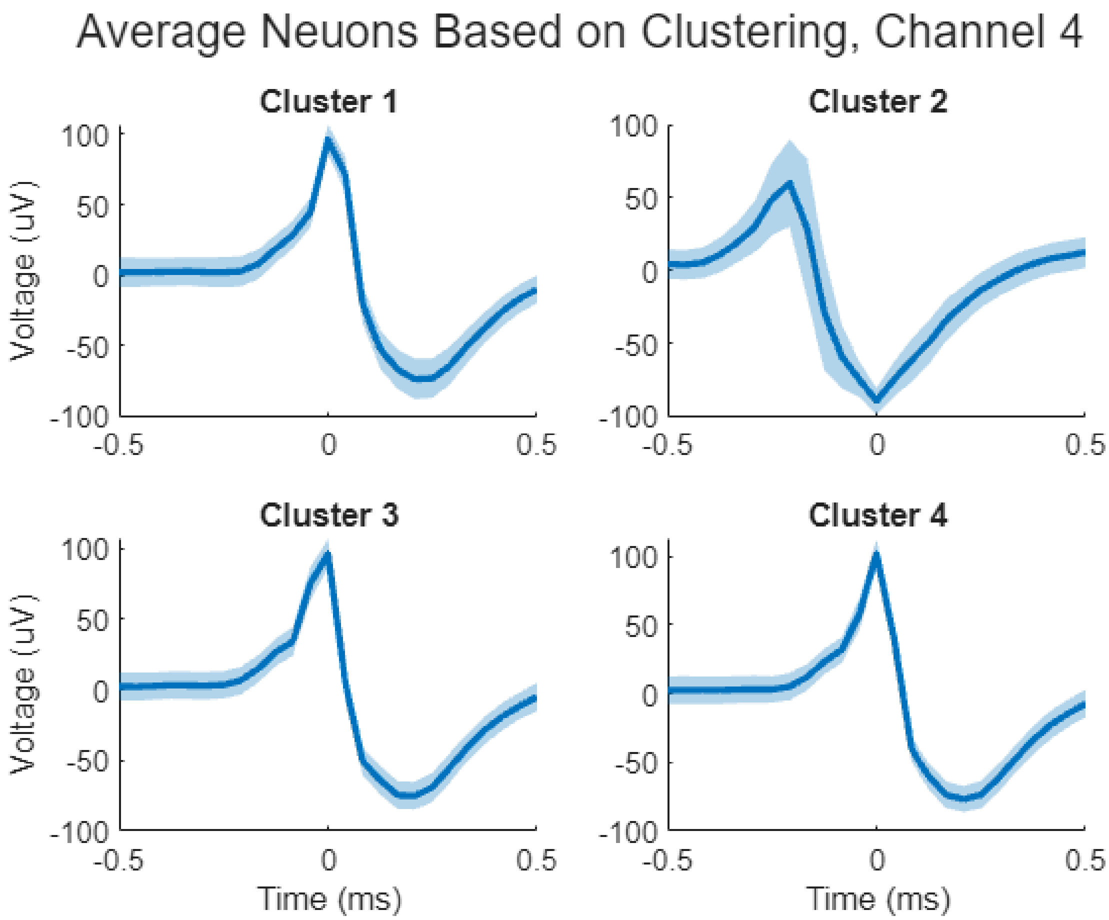

Figure 33. Average Neural Spikes, Based on k-means Clustering Results, Channel 4

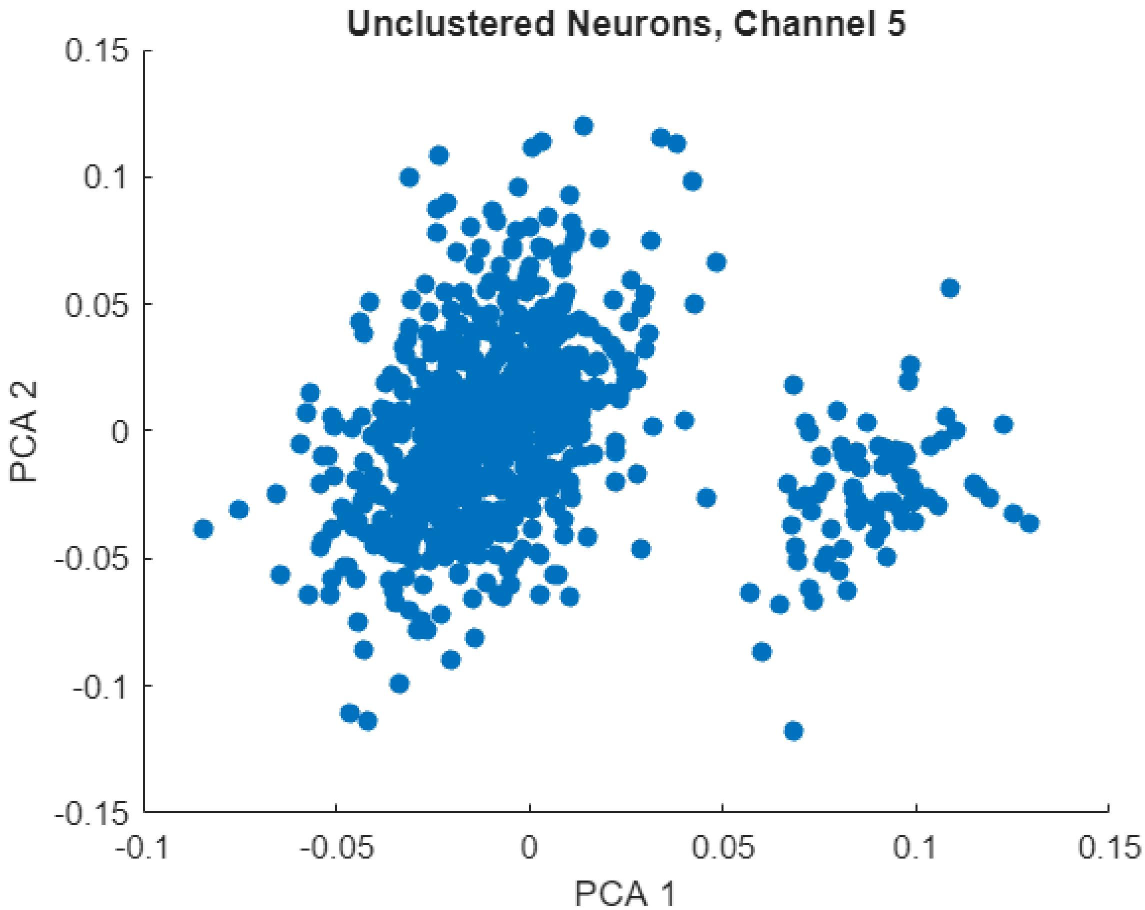

Figure 34. Unclustered Neural Spikes, Channel 5

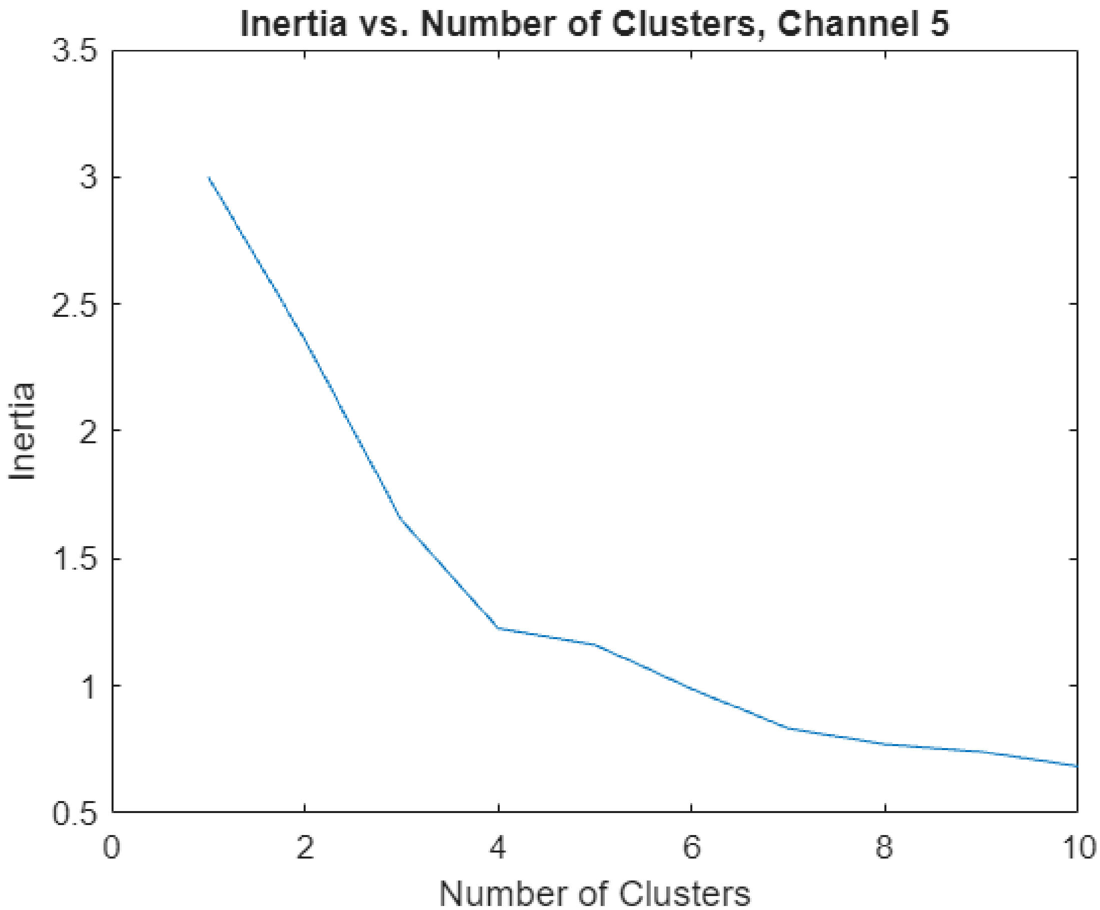

Figure 35. Elbow Plot for k-means Clustering, Channel 5

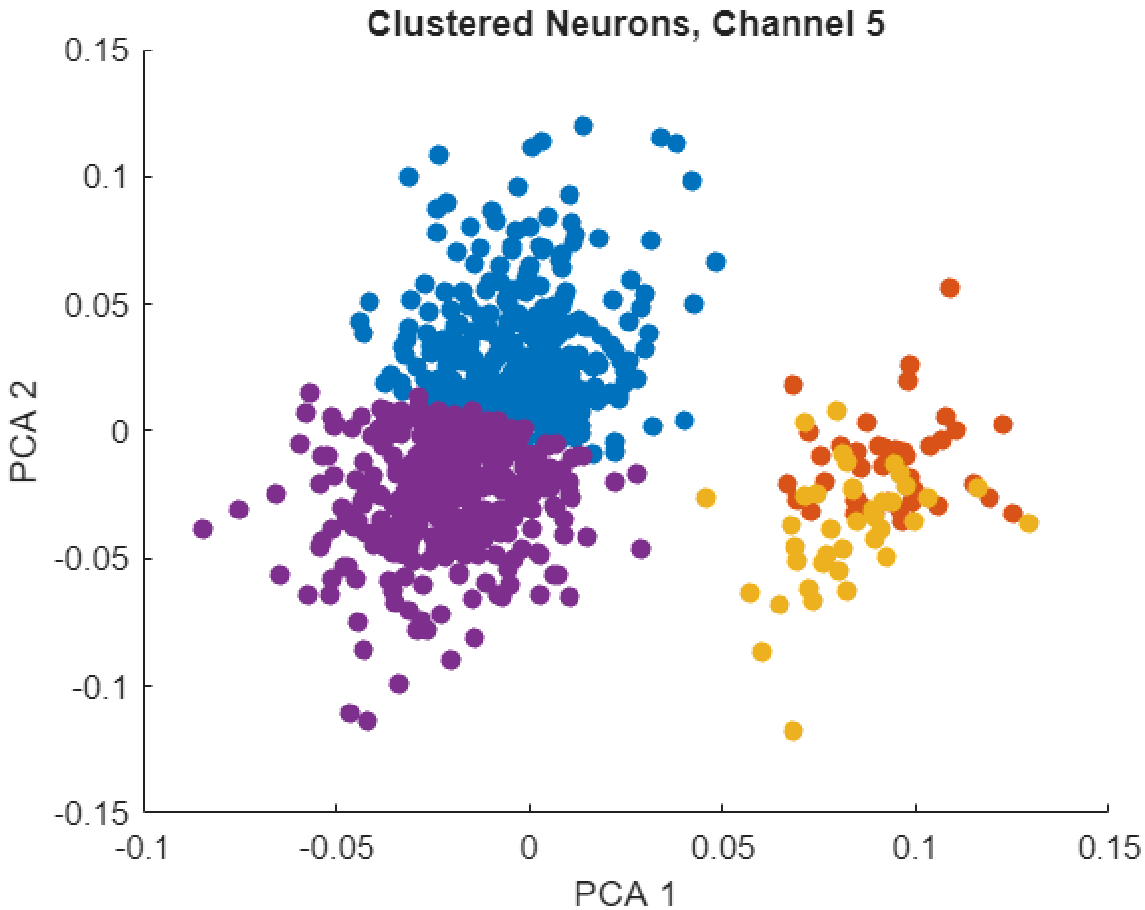

Figure 36. Clustered Neural Spikes, Channel 5

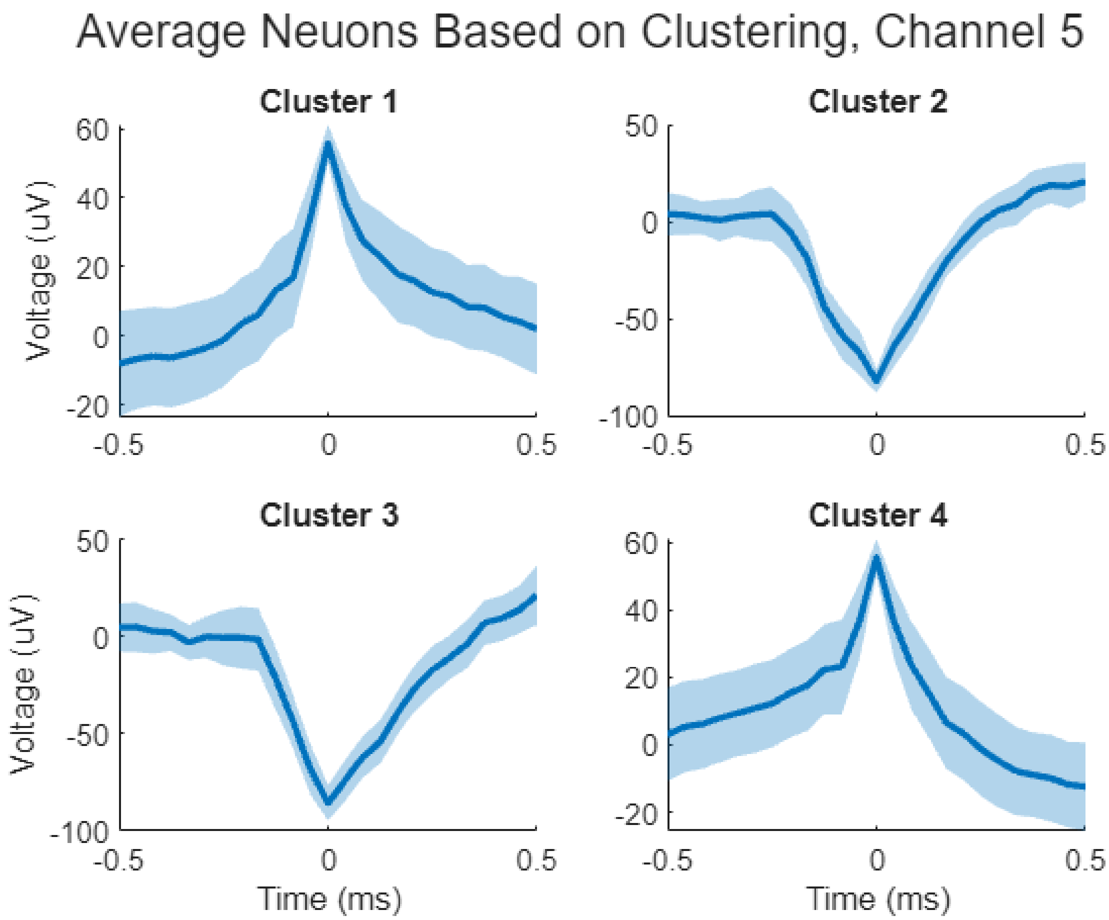

Figure 37. Average Neural Spikes, Based on k-means Clustering Results, Channel 5

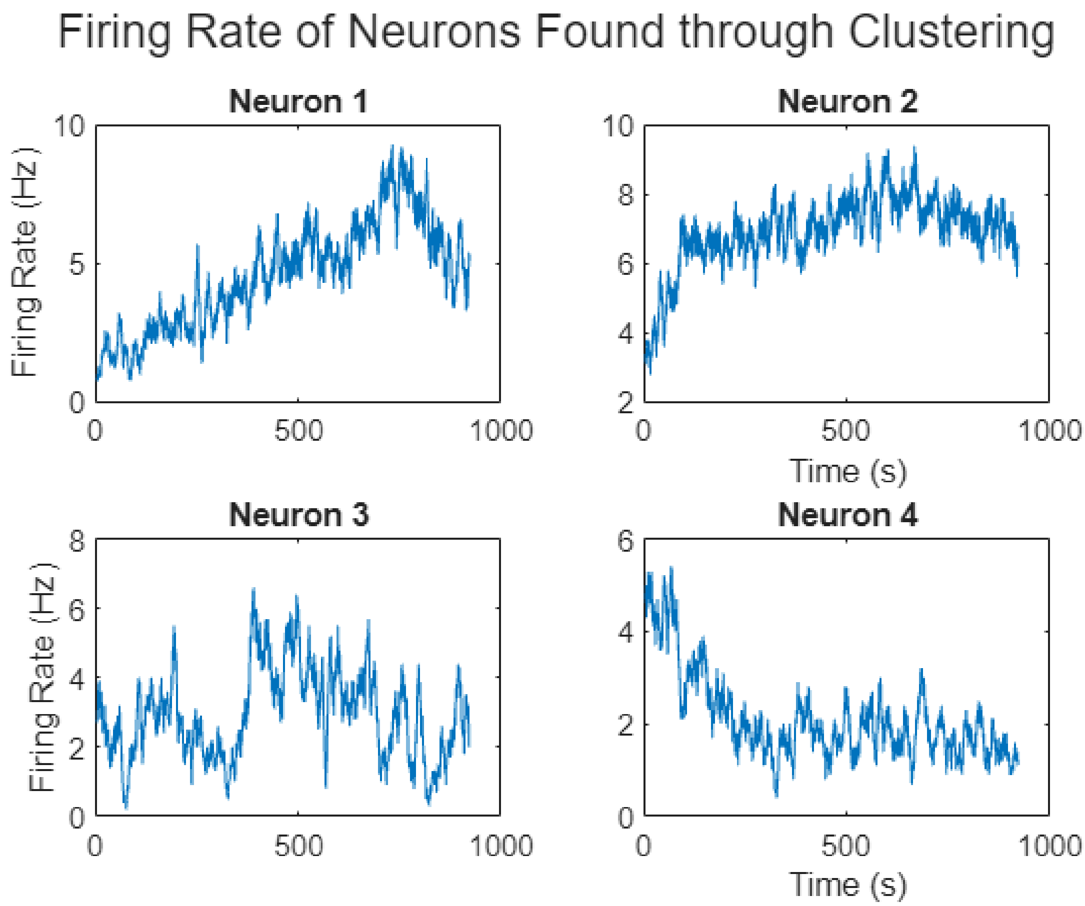

Figure 38. Neuron Firing Rates for Clustered Neurons from All Channels

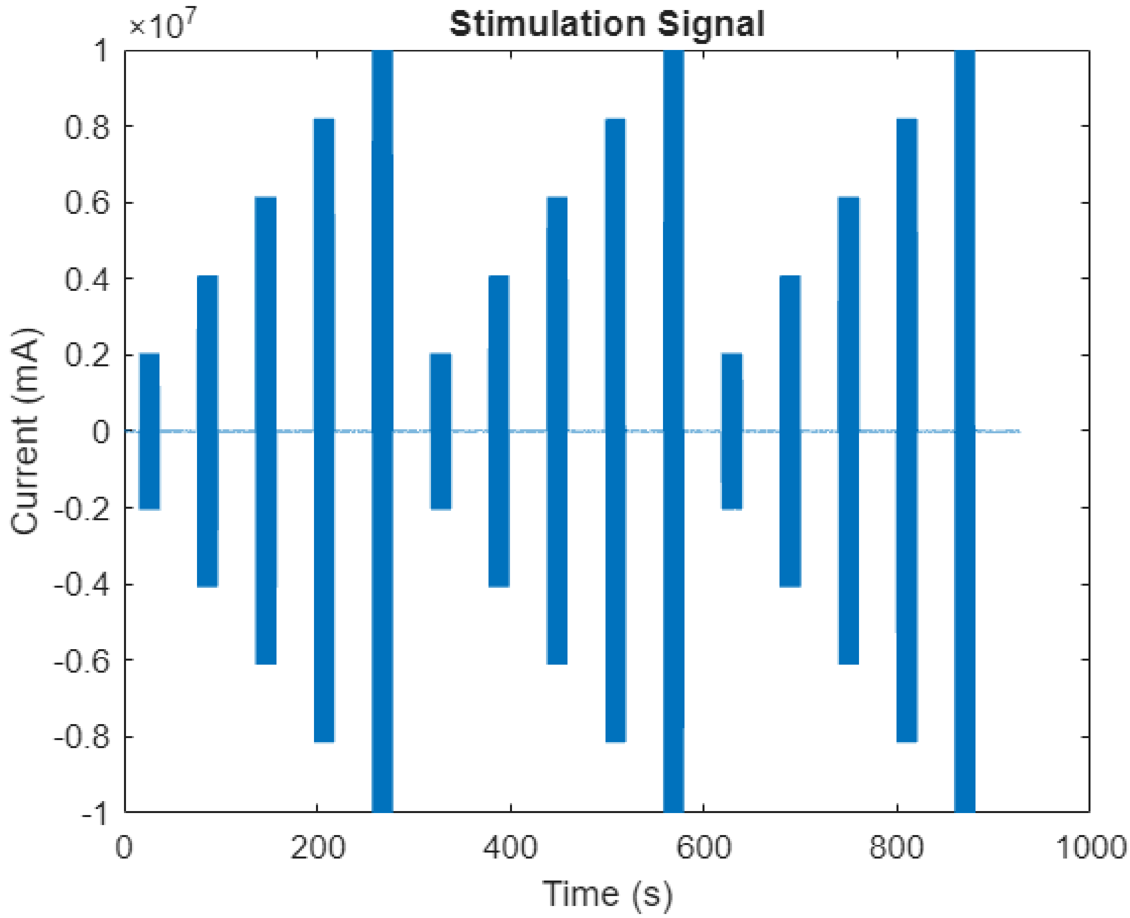

Figure 39. Peripheral Nerve Stimulation Signal

Tables

Channel	Mean	Standard Dev.
1	-9.23E-06	10.21
2	-7.74E-06	9.88
3	-1.10E-05	15.52
4	-4.06E-07	10.58
5	-1.03E-05	9.43

Table 1. Means and Standard Deviations of Raw Data for Each Channel

Channel	1	2	3	4	5
1	1	0.496	0.108	0.142	0.159
2		1	0.115	0.153	0.170
3			1	0.109	0.119
4				1	0.458
5					1

Table 2. Pearson's Correlation Coefficient Between Each Pair of Channels