Spike Sorting and Behavioral analysis software

Ajinkya Kokate
Department of Computational Science
University of California, San Diego
La Jolla, CA 92092
akokate@ucsd.edu

December 14, 2012

Abstract

In this project, Graphical User Interfaces (GUI) are designed in MATLAB to implement spike sorting and behavioral analysis for the interactive playback experiments and are tested for a behavioral experiment. The basic aim of the software is to output both spike times and behavioral times, so that dynamics of the neurons could be studied with respect to various behavioral events in the interactive playback experiments.

In the first part of the project, a GUI for spike sorting was designed to extract spike times belonging to different neurons from neural recordings. For the second part of the project, a GUI was implemented for extracting the timings of various behavioral events from files containing playback recordings.

Finally, plots concerning dynamics of the neurons were made using the spike times and behavioral times, to show the practical use of the software in the interactive playback experiments.

1 Introduction

1.1 Significance of interactive playback

In this study (Miller and Wang 2006), it is showed that interactive playbacks were significantly more effective at eliciting antiphonal calls than traditional playback experiments. Interactive playbacks differ from traditional playback experiments in that the timing of stimulus presentation is determined entirely by subjects' behavior and occurs in response to subjects' vocalizations. In other words, during interactive playback experiments, stimulus presentation timing is based on when subjects produce calls, rather than being presented at a specifc timing interval.
The vocalizations are not the only sources of information. Aspects of each species vocal behaviors are likely to be communicatively rich as well. During vocal interactions, for example, the latency delay between the calls could communicate an important message to the signal receiver, such as an interest and willingness to socialize. Hence, we need interactive playback to address this issue in the antiphonal calling behavior of common marmosets.

1.2 Steps in interactive playback

The logic of the system is as shown in figure 1. Subjects initiated the software by producing a phee call. Once that initial phee was produced, the system broadcasted an 'antiphonal' phee stimulus at a preset latency interval: 'antiphonal latency'. Subsequently, each time subjects produced a phee, an 'antiphonal' phee stimulus was broadcast at the 'antiphonal latency'. If subjects did not respond to the 'antiphonal' phee stimulus within a predetermined period of time, labeled as 'spontaneous period 1', a 'spontaneous' phee stimulus was broadcast. If subjects did not respond to two consecutive 'spontaneous' phee stimuli, the interval between the 'spontaneous' phee stimuli was increased to a preset interval, labeled as 'spontaneous period 2', and the level of the stimulus was decreased to a preset gain. If at any point subjects produced a phee, an 'antiphonal' phee stimulus was broadcast and the same process for broadcasting an 'antiphonal' phee and 'spontaneous' phee occurs.

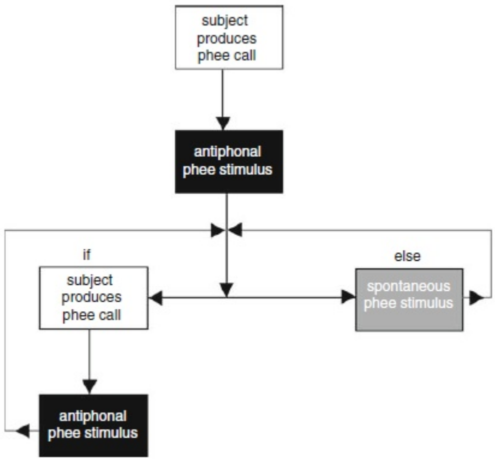

Figure 1: *Interactive Playback system*

1.3 Goals

As seen above, the interactive playback system will have behavioral events at non pre-determined timings and they would occur randomly because of the 'interactive' nature of the system. So, the commercial softwares available for the processing of neural and vocal recordings can only be used for 'traditional' playback experiments and not for these 'interactive' experiments. Hence, I designed a software that would process the neural and vocal recordings of interactive system and give the timings of spikes and classify them according to their originating neuron and also give out timings of the behavioral events.

2 Methods

The software essentially consists of two sections: Spike sorting and behavioral analysis and each of them is described below:

2.1 Spike sorting

The aim of this part of the software is to separate the spikes belonging to different neurons and get their timings. The basic steps involved in spike sorting are as shown (Fig. 2). The essential steps involved in this part of software are also decribed individually below:

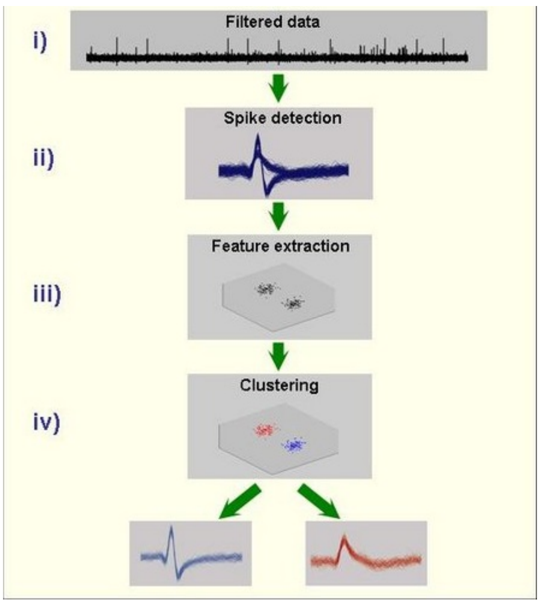

Figure 2: *Spike sorting*

1. **Sampling Data:**
After loading the entire data from the neural recording, the software will take out a sample of data (5 seconds from every minute) and further processing is done only on the sampled data to reduce computational time.

2. **Spike Detection:**
From the sampled data, spikes are detected by setting an amplitude threshold. Although the threshold is set manually, the user has a provision to check whether the threshold is good enough for the entire recording, thereby reducing the error of missed spikes.

3. **Feature Extraction:**
Transforming the input data into the set of features is called feature extraction which involves simplifying the amount of resources required to describe a large set of data accurately. In general, the more features we have, the better we will be able to distinguish different spike shapes. The result of this step is an M x K-matrix, where K is the number of detected spikes and M is the number of extracted features. But the technique of Principal Component Analysis (PCA) is used in the software to reduce the dimensionality of the M x K-matrix by extracting the most important features of the detected spikes. The result is a new matrix of reduced dimensions, L x K, where L < M is the number of extracted features per spike (Tiganj and Mboup, 2011). So, in this step the software displays a 2D and 3D PCA plot of the detected spikes.

4. **Clustering:**
Cluster analysis or clustering is the task of classifying a set of objects into groups (called clusters) so that the objects in the same cluster are more similar (in some sense or another) to each other than to those in other clusters. The fourth and final step of spike sorting is to group spikes with similar features into clusters, corresponding to the different neurons. The method used in the software is a technique called cluster cutting. In this approach, the user defines a boundary for a particular set of features. If a data point falls within the boundary, it is classified as belonging to that cluster; if it falls outside the boundary, it is discarded. Figure 3 shows an example of boundaries placed around the primary clusters. In this off-line analysis the cluster boundaries are determined after the data has been collected by looking at a sample from the data over the collection period. This allows the experiment to verify that the spike shapes were stable for the duration of the collection period (Lewicki, 1998). As a result of this step, we get spike forms belonging to the different neurons.

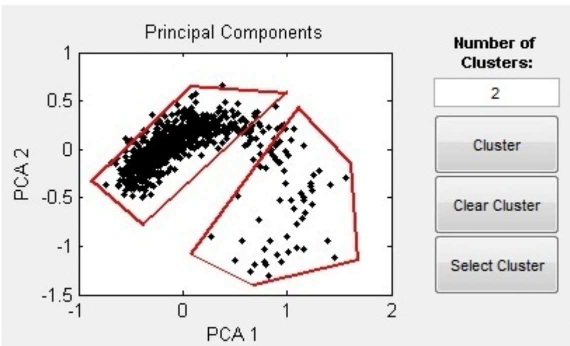

Figure 3: *Clustering*

5. **Spike-times Extraction:**
Although the spike times obtained from the previous step are classified as per the originating neuron, they are only from a sample of data. So in this step, the spike forms of each neuron is run through the entire recording to get the spike times from the entire data for that particular neuron.

2.2 Behavioral analysis

This part of the software does acoustic analysis on the animal and playback calls to get the accurate on-set and off-set times of these events. The methods involved in this part are as described below:

1. **Approximate times:** When the header file for the vocal recordings (behavioral events) is loaded, the software gets the approximate times of the events and classifies them into animal and playback calls depending on the channel of the event.

2. **Acoustic analysis:** In this step the user first selects the type of call to be analysed and then software displays the data for the first event from the vocal recordings using the approximate on-set and off-set times. After getting the data, the software calculates the spectrogram for the loaded data using the following formulae.

$$spectrogram\,(t, w) = |STFT\,(t, w)|^2\,; \tag{1}$$

3. **Event-time extraction:** Looking at the spectrogram the user can easily correct the on-set and off-set times, which were initially approximated by software. Thus at the end of this step, we have the accurate timings for the various behavioral events.

2.3 Interactive playback experiment

The software was then used for a interactive playback experiment in which the animal was tested for Individual recognition during bouts of antiphonal calling. The process of this experiment is as shown in the figure 4:

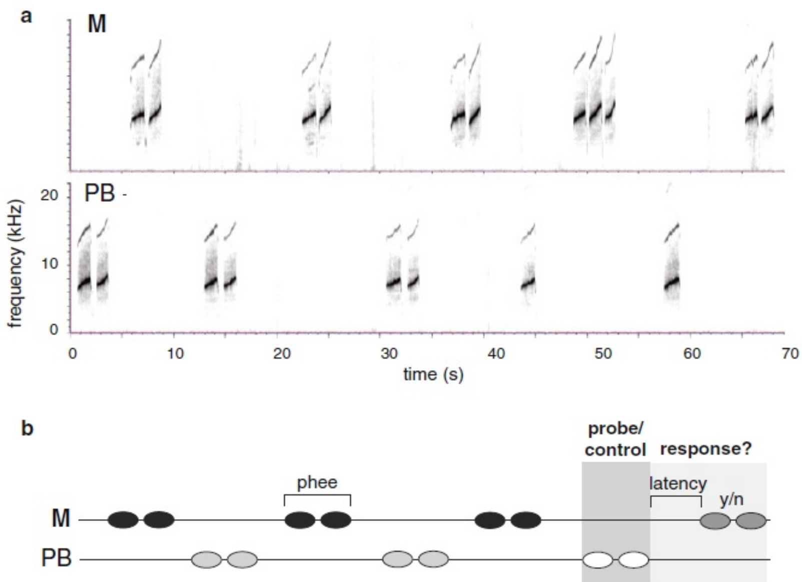

Figure 4: *a) Spectrogram showing an antiphonal calling bout. Marmoset is shown above, while Playback is shown below. The vocalizations depicted in the spectrogram are common marmoset phee calls. b) Schematic drawing of the probe playback procedure. M represents the phee calls produced by the subject marmoset. PB represents the phee call stimuli presented by the software. The 'probe/ control' stimulus is represented as a white colored phee call. That time period is shaded in dark grey. The response period is shaded in light grey.*

Next we used the behavioral times and spike-times obtained from the software to check for the dynamics of the neurons across all the channels to indicate the probe call recognition.

3 Results and discussion

The resulting simulations for both parts of the software were shown during the presentation. Download links for both spike sorting part (spikesorting.wmv) and behavioral part (behavioral.wmv) are available.

At the end of these simulations, spike-times between the on-set and off-set times of the animal calls in response to both, normal calls and a probe call, were used to plot the poststimulus time histogram (PSTH) and cross-correlogram (CCG). The plots for the same are shown in sections below.

3.1 Poststimulus time histogram

1. PSTH for normal calls

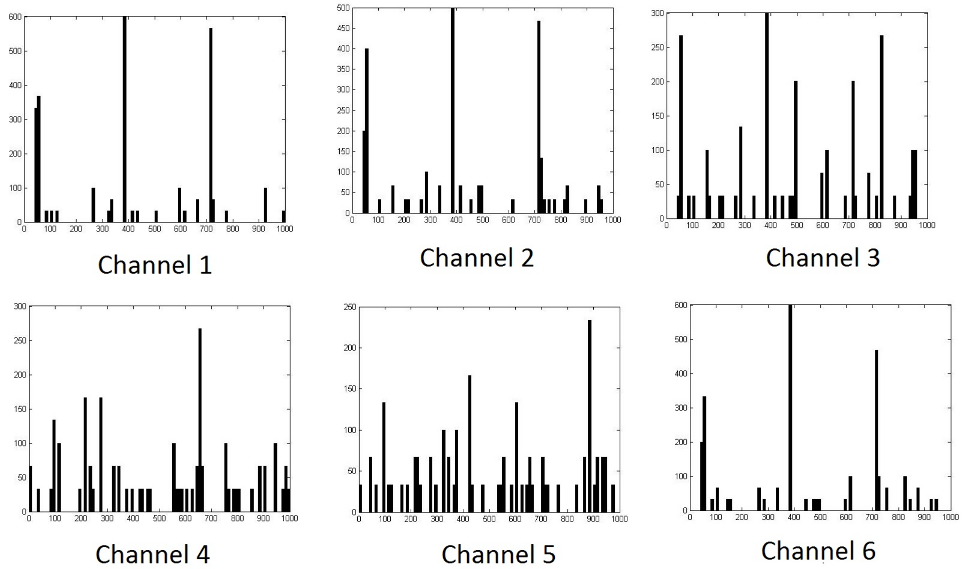

Figure 5: *PSTH-normal calls*

2. PSTH for probe calls

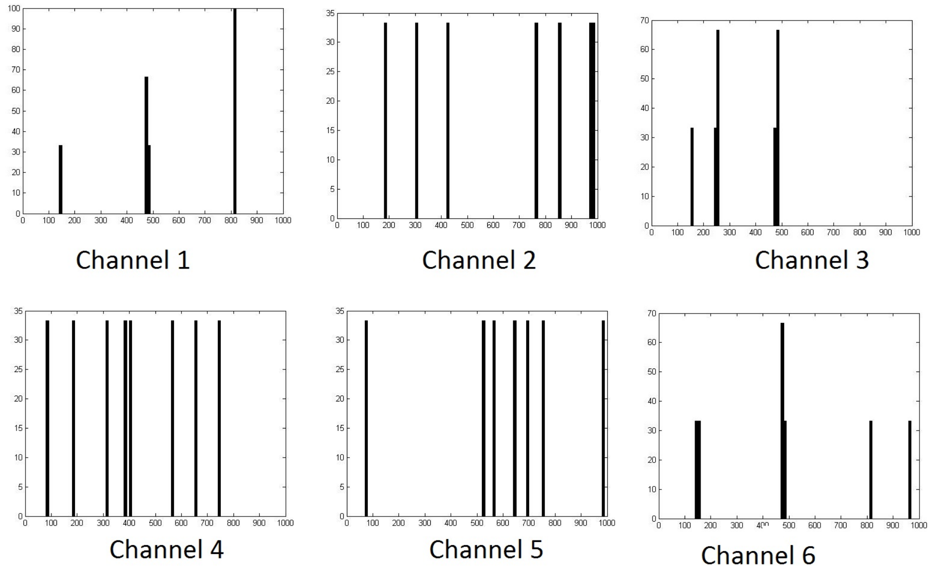

Figure 6: *PSTH-probe call*

Poststimulus time histograms were used to visualize the rate and timings of neuronal spike discharges in relation to an event. As we can see in figures 5 and 6, the PSTH for normal calls indicate a higher firing rate, while it is low in case of the probe call. Thus, the subject was able to recognize the probe call.

3.2 Crosscorrelogram

The crosscorrelogram (CCG) compares the output of 2 different neurons, so first you need to select the 2 cells you want to analyze. You arbitrarily choose one cell to be the reference cell – the spikes of this cell's spike train will provide the reference marker. Discovering relationships between neurons is important, since behavior results not from a single cell but from collections, or networks, of neurons acting together. From correlation analysis, we can get some idea of how specific neurons interact during the process of analyzing sensory information or producing a complex goal-directed movement.

1. CCG for normal calls

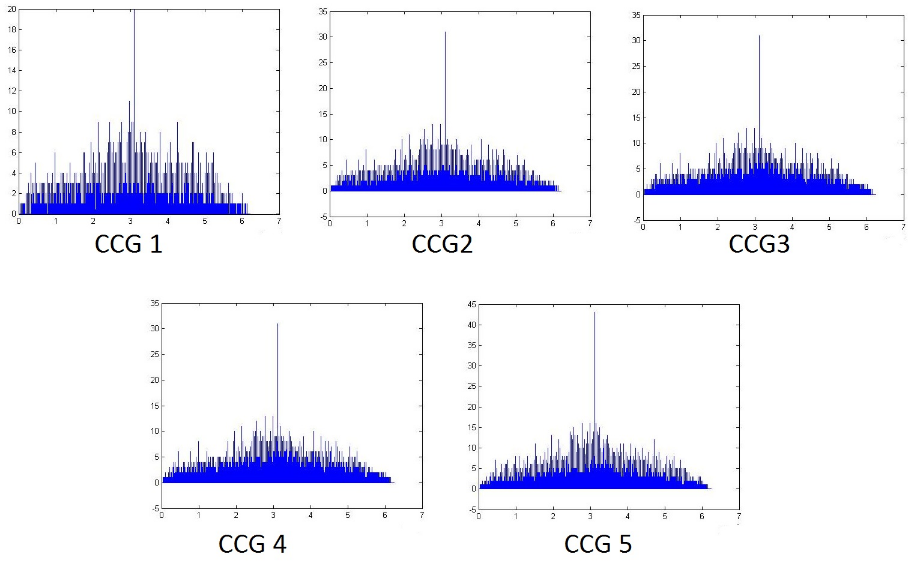

Figure 7: *CCG-normal calls*

2. CCG for probe calls

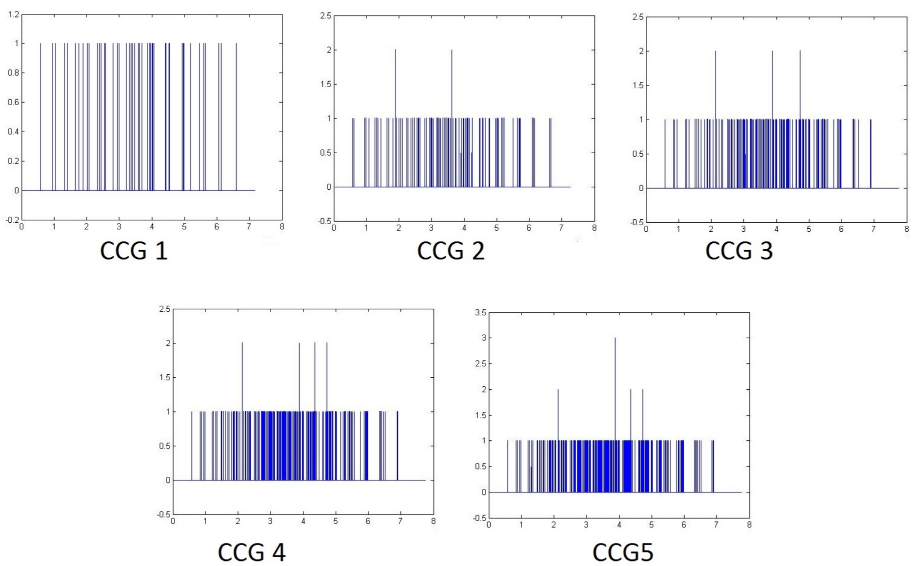

Figure 8: *CCG-probe call*

As we can see in figures 7 and 8, the CCG across channels for normal calls and CCG across channels for a probe call were significantly different. The peaked CCG in the normal calls indicates synaptic coupling or common input, while the probe flat CCG for probe call indicates no dependence. Thereby implying that the behavior of the subject was different for the probe call.

4 Conclusion

In this project, a GUI was designed in MATLAB for spike sorting and behavioral analysis, which gave spike-times and event times at the end of the simulations. These outputs of the software were then effectively used for testing vocal signal recognition and categorization at the neural levels. Hence, this software can be used for various social categorization and behavioral experiments. It also determine the reason for a particular behavior of a subject by studying the dynamics of neurons as shown above.

5 References

[1] Miller CT, Wang X (2006) Sensory-motor interactions modulate a primate vocal behavior: antiphonal calling in common marmosets. *J Comp Physiol* A 192:27–38.

[2] Miller CT, Beck K, Meade B, Wang X (2009a) Antiphonal call timing in marmosets is behaviorally significant: interactive playback experiments. *J Comp Physiol* A 195:783–789

[3] Michael S Lewicki (1998) A review of methods for spike sorting: the detection and classification of neural action potentials *Comput. Neural Syst*