
NENS 230

Assignment #5

Spike Rasters and Peristimulus Time Histogram

Cell Arrays and Structures
Due: 9:30am on Tuesday, October 28th, 2014

Goals

- Become comfortable using *cell arrays* and *struct arrays* to organize and store data.
- Write functions to encapsulate units of code
- Learn to use Matlab debugging tools
- Practice using axis handles

Introduction

In this assignment you will draw upon what you have learned about data structures, functions, data manipulation, and plotting to analyze and plot some neural data. This assignment asks you to combine concepts from previous lectures, and learn new concepts and functions we haven't covered by reading the Matlab documentation and searching online. In practice, most programming involves problem solving in unfamiliar territory and this should start to feel comfortable as you gain experience.

About the data: we have provided data consisting of electrical recordings from neurons in a monkey's brain, while the monkey is moving its arm to touch targets on a virtual screen. In the experiment, electrodes are implanted in a monkey's brain that allow us to electrically record from the neural activity (consisting of action potentials, or spikes) in the monkey's brain. The monkey makes reaches to one of eight radial targets, spread out in a ring (we have labeled them with cardinal directions, N, NE, S, SE, etc). For each direction, we repeat the presentation of the target twelve times to generate twelve trials of data. On each trial, we record the activity from 96 electrode channels.

Neuroscientists commonly store neural activity as just a list of times when a given cell (or in this case, electrode) registers an action potential. These are called 'spike times'. You are provided with a list of spike times for each trial on each of the 96 electrode channels.

In these trials the target appears 50ms after trial onset. The trial ends after the monkey's finger reaches the target and stays there for 500ms. Thus, a typical trial will be 800-1200ms long, during which time each electrode may record anywhere between zero and 100 spikes. Note that the list of spike times is stored in milliseconds, and will thus have values that range from 0 to 1200. Each trial/electrode can have a variable number of spikes, so we store them using a cell array. We use a cell array in this case since we can put

anything we want into a cell. In this case, we're storing a $1 \times n$ array of spike times, and since n is different for each channel and trial, we need a container that can hold arrays of different sizes.

These spike times can be quite noisy on a given trial. To cut through the noise and see the underlying structure in the data, we are going to average the data across the 12 trials. To do this, you will write a function that 'bins' the recorded spikes into larger time bins by counting all of the spikes that fall within a certain time window or bin. Currently the spike times are stored with a resolution of 1ms, but we want to create discrete and non-overlapping time bins that are 50 ms wide, and count everything that falls into the range of 0-50ms, 51-100ms, 101-150, etc.

The way we'll do this is with a set of nested loops to loop over the time bins, channels, and trials. The outermost loop will be the time bin, then inside this loop we'll loop over channels and then over individual trials. The output of this function is a 2D array called `binnedSpikeCount` with size `[# time bins \times # Channels]`. We'll also output the starting times of each of the bins. Don't worry a significant code skeleton is set up for you. You have to figure out how to create the loops, select the correct subset of spikes, and then find the number of spikes that fall within the current bin.

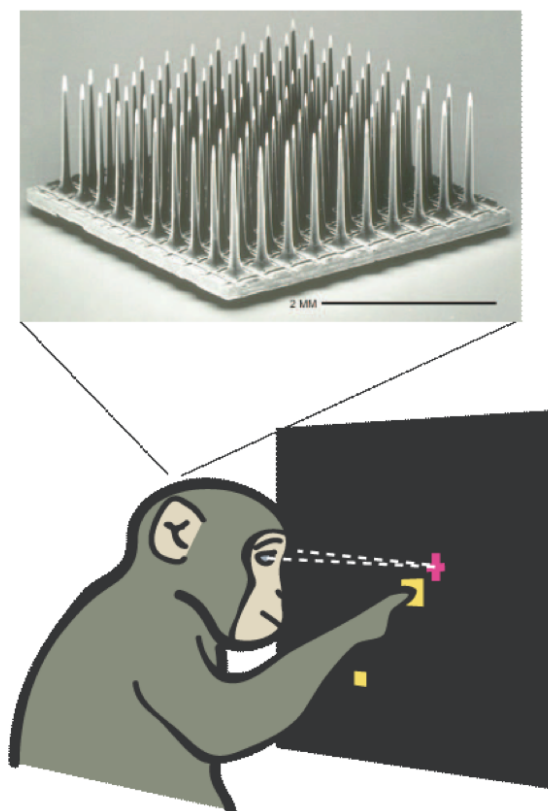


Figure 1: Cartoon of the experimental setup. A macaque makes reaches from a central hold target to one of eight radial targets while spiking activity is recorded from a 96-channel array implanted in M1 cortex. (from Afshar et al, 2011)

Data

Copy the Assignment 5 directory to your MATLAB directory. It contains the data you will need as well as incomplete functions to help you get started. Take some time to familiarize yourself with the format of this dataset.

Remember, **structures** are a data format in matlab consisting of **fields**. We've actually given you a structure array, which is just an array (or vector) of different structures. You can access a single structure in the array using parentheses (e.g., `reachingData(4)` gives you the reaching data for one of the target directions). Given a structure, you can access different fields by using a period (`.`), for example, `reachingData(4).targetDir` gives you the target direction (as a cardinal direction, the direction for the fourth structure is southeast, or SE).

Here is a list of all of the fields that are in each `reachingData` structure:

- **targetDir** - Which target the monkey was reaching to.
- **numTrials** - The number of reach trials to the specific target.
- **spikeTimes** - A cell array, size = [number of trials \times number of electrodes].

Figure 2 below may be useful to understand the organization of the `reachingData` structures `.spikeTimes` field:

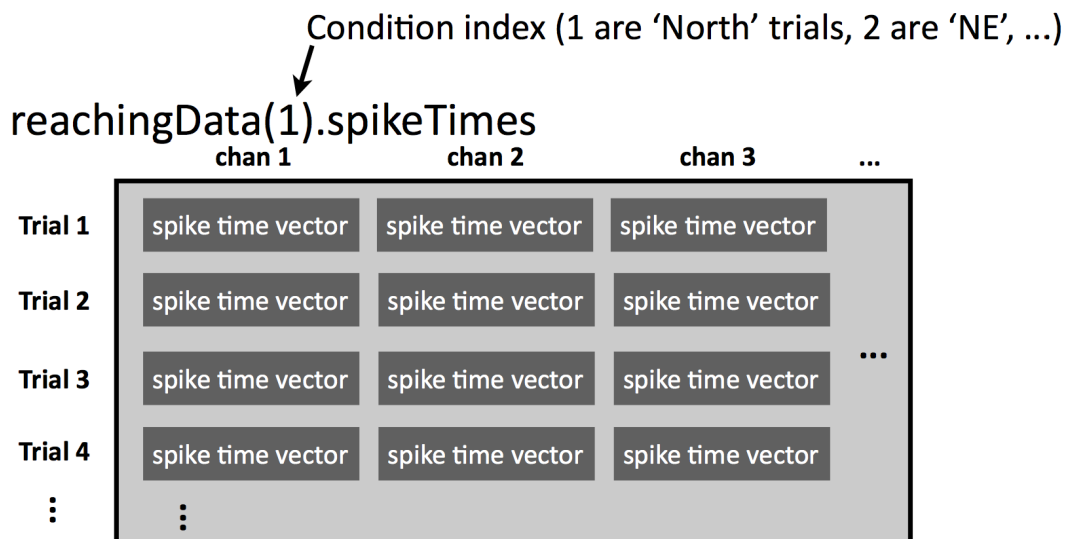


Figure 2:

This giant 12x96 cell array contains data corresponding to one of the 12 trials (along the rows) and for one of the 96 electrodes (along the columns). At each of these locations, we have stored a list of spike times (in milliseconds) as a cell. You can therefore access the spike times on trial 3 recorded by electrode 78 for the southeast (SE) reach using the command: `spikeTimes = reachingData(4).spikeTimes{3,78}`. Note the use of curly braces, which we use when indexing into cell arrays.

Your Assignment

Your assignment is to write the `binSpikeTimes.m` function and flesh out the script `assignment5.m`. This script just plots a bunch of binned spike time data for different target directions for a single electrode from the 96 in the data.

Skeletons of both of these files have been provided, you just need to flesh them out. *Note that because they are incomplete, they will not run unless you have added the necessary lines of code.* Once you are done, you can play around with the `assignment5.m` script to plot the binned data for different electrodes, but ***** please turn in the plot for channel 16 only *****.

Start by stepping through `assignment5.m` and executing cells one by one using either command-enter (mac) or ctrl-enter (win) to execute that cell only. You'll need to write the `binSpiketimes.m` function. There's a significant head start for you there.

Note that break points only work reliably when running the whole script, not just cells. Once you have a good sense of the data structure, write your `binSpikeTimes.m` function. Remember handy debugging tools such as 1) the `keyboard` keyword to stop program execution and give you a look at the variables, 2) hovering over a variable with the mouse in the editor (only newer versions of matlab), 3) break points, and 4) cell mode execution for running a few lines at a time. The final figure should look like the figure shown in Figure 3 below.

When you are finished, save your plot as a .png file, and submit the following:

- `assignment5.m`
- `binSpikeTimes.m`
- `channel_16.png`

Email your script (original code) and the generated .png figure to nens230-aut1415-staff@lists.stanford.edu to submit your assignment, and please include the tag (written verbatim) `[Assignment 5]` in the subject line.

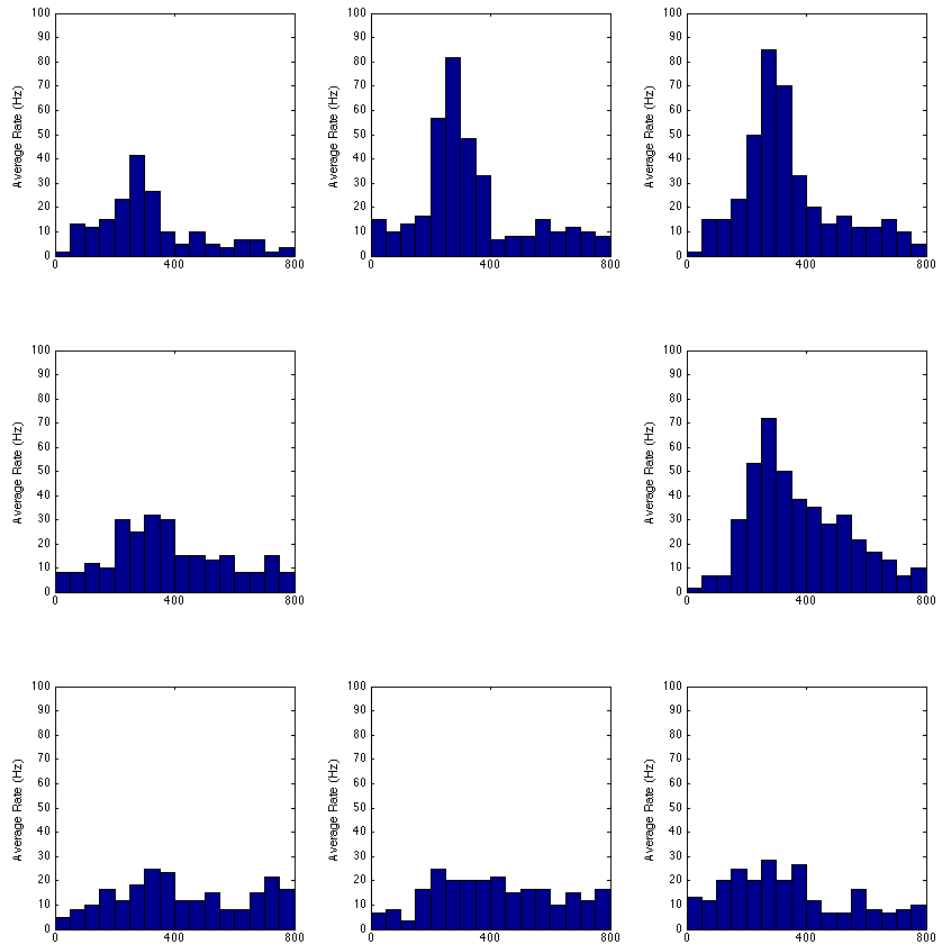


Figure 3: