Week 5. Spike Data Analysis

In this exercise we will analyze spike train data collected in a real experiment. You will load the data in MATLAB, and then analyze the data using the techniques we discussed in lecture.

Part 1: Load & visualize the data

The spike train data will consist of 1000 trials, each lasting 500 ms with a sampling rate of 1000 Hz.

Q: Given this, how many samples will you have per trial?

Load the data. To do so, visit the class website, and download the file 'Spike_data_1.mat' in the Schedule section (Week 5), and load the data:

```
load('Spike_data_1.mat')
```

Upon loading Spike_data_1, your workspace should hold a single variable "d".

Q: What are the dimensions of d?

It's useful to define two new variables that record the number of trials, and the length of each trial:

```
n_trials = 1000;
T = 500; % 500ms
```

To visual the **raster plot** of the whole data, plot the 2D matrix as an image:

```
figure(1)
imagesc(d)
```

Q: What do you see? What does 'imagesc' do?

To get a sense for the data, let's plot the results from the first trial.

```
figure(2)
plot(d(1,:), '-o'); % plot the first trial
xlabel('Time (ms)');
ylabel('Spikes');
```

Q: What values do the data assume? What constitutes a spike?

Q: Visual inspection of data is a powerful tool. Describe what you "see" in the plot. In this single trial, does the neuron spike "a lot" or "a little"? Estimate the number of spikes in this trial.

Q: The results for one trial can be deceiving. Visualize the output of additional trials. Do you notice any consistency?

Q: One way to characterize the data is to count the number of spikes per trial. For trials 1-5, plot each trial (as above) and count the number of spikes. Save your results and show them to me, or to someone else in class, and compare your results. Check point #1

Part 2: Quantify the spiking

Through visual inspection, we've examined the spiking for several trials, and counted the number of spikes per trial. Let's now try to summarize the data for ALL trials.

We could try to do this by hand (as in the previous exercise). But counting spikes by hand is TEDIOUS. Imagine repeating this procedure for 1000 trials! To count in MATLAB, consider the following:

```
n_spikes_per_trial = sum(d,2);
```

Q: Does the above command make sense? Read the MATLAB Help section for the command 'sum' to make sure you understand it. Determine why we use the command 'sum(d,2)', but not the command 'sum(d,1)'

Q: Compare your results counting the number of spikes in trials 1-5 (found through visual inspection above) with the first five elements of the variable 'n_spikes_per_trial'. Do the two match?

Now, compute the average number (or mean number) of spikes per trial.

```
n_avg = mean(n_spikes_per_trial);
```

Q: Have you used the command 'mean' before? If not, look it up in MATLAB Help!

Q: What is the numerical value for n_avg? Is this consistent with your visual inspection of the data?

Part 3: Average firing rate

Q: Use your results above to compute the average firing rate of the data. Tell me what you find, or compare with a classmate. HINT: It's one line of code in MATLAB! Check point #2

Part 4: Spike number histograms

We've learned one quantitative element about the data: the mean number of spikes per trial. To learn more about the data, let's compute a histogram of the number of spikes per trial. This is the "spike number histogram". Consider the following:

```
figure(3)
hist(n_spikes_per_trial, [0:1:40])
xlabel('n = number of spikes');
ylabel('Number of trials containing n spikes');
title('Spike number histogram')
hold on
plot([n_avg, n_avg], [0, max(hist(n_spikes_per_trial, [0:1:40]))],
'r', 'LineWidth', 2)
hold off
```

Q: Remember the command 'hist'? Check out MATLAB Help!

Q: Why do we input two arguments to the 'hist' command? What is the purpose of the second argument? Vary the second argument in 'hist' and re-run the code above. Examine how the histogram plot changes.

Q: Can you explain the line of code started with 'plot'?

Q: Include a title on the plot that prints the numerical value of 'n_avg'. HINT: use 'num2str' to convert a number to a string so you can put it in the title using 'title'. Check point #3

Part 5: PSTH

We also discussed in class the post-stimulus time histogram, or PSTH. To compute the PSTH, we first need to compute the probability of a spike at any moment in time. Consider the following code:

```
prob_spike = sum(d,1)/n_trials;
```

Q: Why does this line of code reveal the probability of spiking at any moment in time?

Q: Use the above line of code to compute the PSTH, and plot it as a function of time. In the line of code above, what is the size of the tiny bin used to divide the time axis? How does the instantaneous frequency evolve over time? Check point #4

Part 6: ISI

We've spent some time considering the number of spikes per trial, and visualized these results in the spike number histogram. Let's now investigate the time intervals between spikes. Namely, let's compute the inter-spike intervals (ISIs).

First, we have to convert the observed data "d" to a list of ISIs. Remember, the data "d" indicates the *time* of each spike (in each trial). What we'd like to know now is the time *between* spikes. To compute the time between spikes, we can do the following:

```
ISI = [];
for k=1:n_trials
    spike_times = find(d(k,:) == 1);
    isi0 = diff(spike_times);
    ISI = [ISI, isi0];
end
```

Q: That's a complicated series of commands! Study it and explain it to a classmate or me. Use MATLAB Help to look up commands you don't yet know.

Q: Execute the commands above and examine the vector ISI. What is it's size? Does it make sense to you?

Once we've constructed the vector ISI, we can calculate its mean,

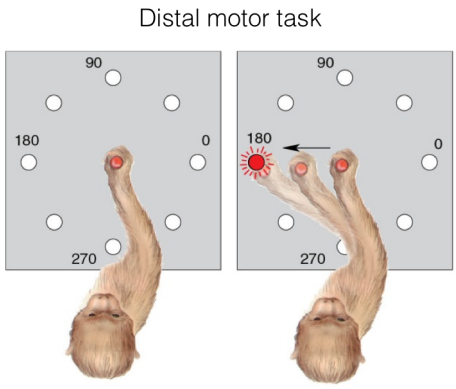
```
mean_ISI = mean(ISI);
```

Q: Examine the value of the mean ISI. Does it seem reasonable? Why (or why not)?

Q: Can you plot the histogram of ISI? Check point #5 Go try EXERCISE 2.1

Part 7: Tuning curve

Download 'Spike_data_4.mat' from the class website. The spike data were collected during a distal motor task. There was a total of 158 trials. During each trial, the subject first needs to hold the joystick at the center position. An "INSTRUCTION" cue will later appear at one of the 8 positions to instruct the moving direction, which will be followed by a "GO" cue. The subject needs to move the joystick toward the cued direction after the GO cue.



The data contains following key variables:

1. **direction**: 158X1 vector that indicates the cued direction in each trial
2. **instruction**: 158X1 vector indicates the timings of the INSTRUCTION cues.
3. **go**: 158X1 vector indicates the timings of the GO cues.
4. **unit** (a structure array):
    - **unit(i).times**: the timings of spikes of neuron i (a total of 143 neurons recorded)
    - **unit(i).area**: the brain area where neuron i locates (MI: Primary motor cortex, PMd: Dorsal premotor cortex)

Try the following code:

```
clear
load('Spike_data_4')

neuronNum=129; %examine neuron 129

%bin firing rates for each direction
spikeCount=zeros(8,1);
for i=1:8
    indDir=find(direction==i); %find trials in a given direction
    numTrials(i)=length(indDir);

    for j =1:numTrials(i)
        %to center on start of go cues
        centerTime=go(indDir(j));

        %center spike times
        allTimes=unit(neuronNum).times-centerTime;

        %pick 2 seconds around center time
        spikeCount(i)=spikeCount(i)+sum(allTimes>-1 & allTimes<1);
    end
    %divide by the number of trials & bin size (2s) for an average
firing rate
    spikeCount(i)=spikeCount(i)/numTrials(i)/2;
end

%plot tuning curve
figure
ang = [0:45:315]; %angles for 8 directions
plot(ang,spikeCount,'o-')
xlabel('Angle')
ylabel('Avg Firing Rate (Hz)')
areaName=unit(neuronNum).area;
title([areaName '- neuron ' num2str(neuronNum)])

%preferred direction (in degrees)
[p, prefDir]=max(spikeCount);
ang(prefDir)
```

Q: What do you see? What is the preferred direction for neuron 129?

Q: Try a different neuron. What is the preferred direction for neuron 132? Do all neurons exhibit similar average firing rates?

Check point #6 Go try EXERCISE 2.5

References:

Reference Book: Pascal Wallisch『MATLAB for Neuroscientists, Second Edition』(Academic Press) ISBN: 0123838363

Tutorial Videos: Adapted from Coursera: Computational Neuroscience taught by Rajesh P. N. Rao & Adrienne Fairhall at University of Washington.

MATLAB Basics

Learning Module	Learning Goals
What is MATLAB?	https://youtu.be/UPdcCSAslqo
Getting Started with MATLAB Online	https://youtu.be/LSHC3M7xrvg
MATLAB Variables	https://youtu.be/FYPaH9pnLCA
MATLAB as a Calculator	https://youtu.be/yaRWYjvqqXk
MATLAB Functions	https://youtu.be/pwzAmsPhTik

Creating Vectors & Matrices

Learning Module	Learning Goals
Creating Vectors	https://youtu.be/e9JAJj54zIE
Creating Uniformly Spaced Vectors	https://youtu.be/OK-3V0NXgs8
Accessing Elements of a Vector	https://youtu.be/NWd7olaA9sU
Creating Matrices	https://youtu.be/ZxhyjmugQaY
Array Creation Functions	https://youtu.be/aQLu91JFPpg
Determining Array Size and Length	https://youtu.be/1eRF0yewPqI

Calculations with Vectors and Matrices

Learning Module	Learning Goals
Calculations with Vectors	https://youtu.be/moXftAx0fMc
Calculations with Matrices	https://youtu.be/JJIfnL7l-gw
Statistical Functions with Matrices	https://youtu.be/-A70d2cr-cM
Matrix Multiplication	https://youtu.be/IcXBakNZmhI
Solving Systems of Linear Equations	https://youtu.be/y4Ie6gTDFUg
Calculating Eigenvalues and Eigenvectors	https://youtu.be/KjnOOJShE58

Plotting

Learning Module	Learning Goals
Line Plots	https://youtu.be/0UnTjAMER0c
Exploring Figures in MATLAB Online	https://youtu.be/z_IZAf3sOig
Multiple Plots	https://youtu.be/kSCKVALwGmI
Annotating Graphs	https://youtu.be/S4CWcrYnGe0

Calculations with Vectors and Matrices

Learning Module	Learning Goals
Managing Files in MATLAB Online	https://youtu.be/NJP-iRGnfuM
Writing a FOR Loop	https://youtu.be/GoSOE_BLbd4
Generating Random Numbers	https://youtu.be/qiFLKLQYfEs
Logical Variables	https://youtu.be/--Pe2xdh92Q
Writing a WHILE Loop	https://youtu.be/Yw2o0SmhtgM
IF-ELSE Statement	https://youtu.be/K7JEcd5JBL0
Accessing Elements of Vectors Using Conditions	https://youtu.be/3g6Ck-mla1Q