Problem Set #1: spike train statistics

Publishing and Turning in your homework

Hand in the homework by "publishing it" using Matlab. Instructions for publishing are describe in the Matlab Demo called Publishing Code from the Editor (in the Help Menu). Please generate a PDF with the following type of filename:

`PS1_Butts.pdf`

This filename clearly contains the solutions to the first problem set (PS1) from a student with the unfortunate last name of Butts. Please note the underscore and the case of letters (I hate having to rename them all the time). Also, separately turn in the m-file used to generate this code, so that I can look at details and debug if necessary. Use the same name, but with a .m for this file. Note that it will be necessary for me to run this, so be sure that it runs on its own (after typing **`clear all`**).

Submit these assignments directly to the ELMs site by the due date(s).

Clear easy-to-read code (with appropriate commenting) will be appreciated. For example, please separate MATLAB code from normal text by a blank line, before and after, and clearly label which problem (#, and letter, if applicable) each part is addressing. You should verify that the output of your own code displays the appropriate answers.

A note about the format of spike train data for this course:

Spike trains are ordered lists of the time of each spike, in units of seconds. We will assume the experiment starts at time zero and continues until time T, which is the duration of the experimental trial. Thus, all valid spike times should have a time associated with it between 0 and T.

Multiple repeats:

While spike trains within a given trial should be sorted, in many cases we will want to list the spike train response for a given neuron over multiple repeated trials. For these purposes, please use the following conventions. First, make the spikes from multiple repeats all part of the same list (array). Spike trains within each repeat should be ordered from 0 to T, and successive repeats should be separated by a -1 inserted after the last spike of each repeat. [Note that this extra number will not be confused with an extra spike because it is negative.] Thus, for example, the following array represents 4 repeated trials:

```
spks = [0.453 0.855 -1 0.103 -1 -1 0.023 0.634 0.914 -1];
```

Note that the third trial has no spikes in it. This will also allow you to use some programs that I have made to analyze spike trains, including **`raster.m`**, which you should use for this problem set.

Binning convention:

In many cases, it is necessary to histogram data (such as spike times for a PSTH, interspike intervals for an ISI distribution, etc). To do so, please use the following convention. For a bins of width dt, have the first element of the histogram (first bin) represent time intervals 0<= t < dt, the second represent intervals dt <= t <= 2*dt, etc. This will also give you the right number of bins. I recommend using the MATLAB function **`histc`** for this:

```
firingrate = histc(spks,0:0.001:1)/4;
```

Note that if you are analyzing a spike train that you generate yourself, it might make sense to shift the bins by an infinitesimal amount to prevent the bin boundaries from being directly over the spikes:

```
firingrate = histc(spks,-1e-8:0.001:1)/4;
```

Note that **`histc`** usually makes an extra bin at the end (which is empty). Also, the example above divides by 4, which is the number of repeated trials in this example, and you would probably want to use a different number.

1. Poisson neuron simulation and analysis.

   A. Generate an array in MATLAB called **`spks0`** that represents 100 repeated trials lasting one second each, recording a neuron firing with uniform probability (i.e., Poisson) at 20 Hz. Use a bin size of 1 ms (0.001 sec) to generate the spikes. [It is also possible to generate the spikes with arbitrarily fine precision, but it is not necessary to do this in this problem.] You should only generate this data once at the beginning of the problem, rather than generate a new instance for each part.

   B. Display the first 20 repeats of the spike train using the function **`raster.m`**, included on the ELMs website.

   C. Using 10 ms time bins, calculate and plot the instantaneous firing rate over the second-long experiment (using all of the 100 repeats that you generated to estimate). Make sure the firing rate is represented on the y-axis (in units of spikes/sec or Hz). Also be sure to label the x- and y-axes.

   D. Display a histogram showing the distribution of the number of spikes on each trial (over the entire second).

   E. Calculate the mean number of spikes over all trials. Also calculate the variance in spike count from trial-to-trial. Calculate the Fano factor.

   F. Display a histogram that shows the distribution of the number of spikes in the first 100 ms, and recalculate the mean number of spikes, variance, and Fano factor as above.

   G. Do the answers from D-G match what you would expect? Why or why not?

2. Analysis of real data. Download data from an LGN neuron from the elms website, called **`LGN_FFdata.mat`**. Load the data into MATLAB (**`load LGN_FFdata.mat`**): this will put several variables into memory. The only one you should worry about is **`FFspksR`** for this problem set. This is the response of an LGN neuron to multiple repeats of a type of visual stimulation called *full-field flicker*, to be explained later.

   A. How many repeats are given in this data? What is the mean firing rate? Display a raster plot for the first 200 ms of this data for all repeats.

   B. Display the interspike interval (ISI) distribution for this spike train, with 1 ms resolution, up to 50 ms. This ISI distribution should be a simple histogram of all ISIs less than 50 ms. Please describe the factors that you think shape the different features of this distribution.

   C. Calculate the Fano factor for the specific time window of 160-180 ms (this should encompass a large firing event). Did you expect this sort of answer for the Fano Factor? Why/why not?