In-Class Exercises for Analyzing Spike Trains (1/21/2025)

Introduction to Spike-Rate Smoothing:

1) Download the Matlab code for Lecture #3 from Canvas and put it in your path.

2) Use the function `[times,spikes] = generateSpikingData_poisson(meanRate,simulationTime)` function to create a 10-second-long spike train with a mean firing rate of 10. This collection of spikes is an example of a Poisson process, which means that spikes are equally likely to occur at any point in time.

3) Use the function `plotSpikeTrain(times,spikes)` to plot the spike train. Can you observe any pattern in the spikes?

4) Before plotting the firing rates, sketch what you think the firing rate vs. time should look like?

5) Now use `plotAllWindowTypes(times,spikes,windowSize)` to plot firing rates vs. time for windows of size .01, .05, .1, .25, and 1 seconds. Explain what is happening as the time window becomes larger.

6) Repeat steps 2) to 5), but now start with generating an oscillatory time series using `[times,spikes] = generateSpikingData_oscillations(max_rate,min_rate,frequency)`. This process is called an "inhomogeneous Poisson process" because it represents a Poisson process whose mean changes with time. You can the default settings except that you should have the maximum firing rate be 200, the minimum firing rate be 100, and the frequency should be 1.

7) For the four plot types, which looks most accurate at each of the time bin sizes? Why might this be?

8) Let's look explicitly at the difference between gaussian-windowed firing rates and causally windowed rates. Use `calculateGaussianWindowedFiringRates(times,spikes,sigma)` and `calculateCausalFilteredFiringRates(times,spikes,alpha)` to generate filtered plots for sigma = 0.05 (and alpha = 1/0.05). Plot the resulting firing rate plots on the same axes (you can use "`hold on`" to do this). What is the difference between the two plots?

Calculating a Tuning Curve:

1) Use `linspace(min,max,N)` to create a set of 100 evenly-space angles between 0 and 2π.

2) For each of the angles, use `generateSpikingData_angles.m` to calculate an average firing rate using a simulation time of 5 seconds and a dt of .001 seconds (the default parameters).

3) Plot the observed tuning curve. What is the preferred angle?

4) Now do the same thing but reducing `simulationTime` to 2. Describe what happened to the plot?

5) The underlying curve is given by $r(\phi) = 15\cos\left(\phi - \frac{\pi}{3}\right) + 20$. For your previous plot, calculate $\chi(\phi) = calculated\ rate - r(\phi)$. This function is called the residual.

6) Lastly, change `simulationTime` from 1 to 20, advancing by 1. Plot $\sum_\phi \chi(\phi)^2$ as a function of `simulationTime` on a log-log plot (`loglog` in MATLAB). What is the slope of this line on a log-log plot?

Calculating a Spike Triggered Average

1) Load the data set in `H1_data.mat` into your workspace. This data set provides the spike times of the fruit fly H1 neuron (`H1_spikes`) when presented with a stimulus (`H1_stimulusData`). `H1_times` are the times associated with the stimulus values.

2) Use the function `calculateSpikeTriggeredAverage.m` to calculate and plot the Spike Triggered Average (STA) for the data using only the first 100 spike times and an `averagingWindow` of 100.

3) Now do the same for the first 50, 100, 1000, and 10000, and all spikes (note that this last one may take a minute to run). How does the curve change as you increase the amount of data? Do the changes look random or systematic (i.e., do the curves consistently increase or decrease with the number of spikes included, or do they fluctuate above and below the large-number-of-spikes solution)?

4) Calculate the Coefficient of Variation (CV) (standard deviation divided by the mean) for the inter-spike intervals this data set. Does it look more ordered, random, bursting, or somewhere in between?