Lecture 4 Problem Set, MATLAB

We're going to make some simple simulated neural data and play with analyzing it. Our super-simple "neuron" will have almost no dynamics in it, and will not be very realistic at all, but it will be useful nevertheless. Later in the class we'll deal with some more realistic simulations and data.

We're going to simulate an experiment in which you've patched onto a neuron in a slice (the neuroscientists can explain what this means), stimulate it with a current, and record the resulting membrane potential. Then we'll identify moments when the simulated neuron would spike, and we'll look at those spike times. We'll bin them in time, to get estimates of number of spikes/sec for each time bin. And then we'll look at what happens when we stimulate two independent neurons. The final questions we'll asses are, "how correlated are the firing rates of the two neurons, and how does the strength of that correlation depend on experimental parameters such as number of trials and stimulation amplitude?"

1. Make a vector, called "t", that will stand for your time axis. Its elements should climb, in uniform steps of 0.001 (i.e., one millisecond) from 0 to 2 seconds. Now make a vector called "drive" that is the same length as t and whose elements are a sine wave, frequency 4 cycles/sec, with a mean of zero. The amplitude of the sine wave modulation should be "drive_amplitude" (this is a parameter that we will later change, so it is good to define it as a variable, not as a fixed number). Thus, when drive_amplitude = 1, if you plot(t, drive), you should see an amplitude 1 sine wave, of 8 cycles total (4 in each of the two seconds).

2. Now we'll simulate injecting this drive (think of it as a current) into a neuron. Write a function called "neuron.m" that takes a drive vector as its input parameter, and creates a simulated membrane potential that, at each timestep, is -60 (the resting membrane potential, in millivolts) plus a sample from a gaussian distribution with standard deviation 5 (to represent the typical neural noise observed in experiments; the randn.m function will be useful here) plus the value of the drive at that timestep. If the voltage goes above -50, we'll say it has reached threshold and the neuron spikes: replace each element of your membrane potential vector that is greater than -50 with +10, the peak voltage of a spike. (Our super-simple neuron has no dynamics in that the membrane potential at one timestep doesn't depend on any other timesteps; in week 3 we'll change that and go to more realistic neurons.) Your "neuron" function should return a vector that represents the final membrane potential, v. Plot v versus t to see what v looks like. Run it several time to see it several times-- since you're using random numbers, it should look different each time.

3. We will call each time you run your neuron a "trial". Run the neuron for many trials, and plot the resulting spike times in a "raster plot": this is a plot in which the y-axis represents trials, so each row will represent one trial; and the x-axis represents time

within each trial. At the time of each spike in each trial, you should place a small vertical line in your plot indicate that a spike occurred. What does the raster plot look like for different numbers of trials and different drive amplitudes? Plot it for 100 trials, with drive_amplitude=1. Can you see the sine pattern?

4. We will call a set of n trials an "experiment." Divide the time in each trial into bins of b milliseconds (start with b = 25 milliseconds). For each trial, and for each of these time bins, find the number of spikes fired by the neuron. Divide by b to obtain, in each bin and for each trial, an estimate of the number of spikes/sec fired by the neuron. This is an estimate of its firing rate. Using imagesc.m, plot the estimated firing rates for each trial as a function of time during each trial.

5. Take your estimated firing rates, and average over trials to obtain an estimated firing rate as a function of time that we'll call the "PeriStimulus Time Histogram" or PSTH. Plot the PSTH as a function of time t. What does the PSTH look like for different numbers of trials and different drive amplitudes?

6. Now run *two* experiments, both with same number of trials and the same drive amplitude, to represent two neurons that you patched simultaneously, injected current into simultaneously, and recorded simultaneously. Compute the two PSTHs that correspond to these two neurons in the two experiments. Calculate the correlation coefficient between the two PSTHs. (If σx and σy are the standard deviation of x and y, respectively, and < > represents averaging, the correlation coefficient is defined as:

$$\frac{\langle xy\rangle - \langle x\rangle \langle y\rangle}{\sigma_x \sigma_y}$$

7. Run the pair of experiments (as in question 6) many times, using different numbers of trials, and different drive amplitudes. Use imagesc to plot correlation coefficient as a function of drive amplitude and number of trials. Replot the same data as a "surface plot" using surf.m. If you're trying to observe a correlation coefficient, you could use this simulation and its final plot to try to decide what drive amplitude to use, and how many trials to record from, in each experiment. Does correlation coefficient become stronger or weaker as number of trials or drive amplitude grow?