Computational Neuroscience
Assignment for Dr. Chacron

What you'll hand in:

1) The code (i.e. m-file) that you used to generate ALL graphs, quantities. Please include all your code into one m-file and e-mail it to Dr. Pack (christopher.pack@mcgill.ca). Include your name, McGill ID#, and the course code in the subject line. I should be able to simply run your m-file from matlab and obtain ALL your results. Also, please add comments to your code (i.e. explain what you are doing).

2) printouts of ALL graphs, clearly labeled (i.e. what quantities are being plotted, units, etc…). Or, if you want to save paper, you can e-mail the document in a commonly used format (i.e. word, pdf, etc…).

3) Answers to ALL questions. These answers must be justified with proper argumentation. Again, you can include them in a text document and e-mail it.

Here are some command lines in matlab that you may find useful:

```
>>[pxx,f]=pwelch(X,bartlett(2048),1024,2048,2000); %
computes the power spectrum of vector X which is sampled
at 2 kHz. Note that you have to divide the estimate by 2
since this will be a one-sided estimate.
```

```
>>[pxy,f]=cpsd(X,Y,bartlett(2048),1024,2048,2000); %
computes the cross-spectrum between vectors X and Y
which are sampled at 2kHz. Note that you have to divide the
estimate by 2 since this will be a one-sided estimate.
```

```
>> [c,lags]=xcorr(X,200,’unbiased’); % computes the
autocorrelation function of vector X up to 200 lags on each
side (i.e. c has length 201).
```

```
>>[c,lags]=xcorr(X,Y,200,’unbiased’); % computes the cross-
correlation function between vectors X and Y up to 200 lags
on each side (i.e. c has length 201).
```

```
>>[c,lags]=xcorr(X-mean(X),200,’coeff’); % computes the
auto-correlation function of vector X subtracting the bias and
normalizes so that the autocorrelation function at lag 199 is 1.
This is useful when computing the autocorrelation function of
the ISIs.
```

Part I) Spike train statistics.

A) Making a binary representation of a spike train.

Load file “spiketrain1.mat” into matlab. You’ll see two variables “timeaxis” and “Vm”. “Vm” is the membrane potential of a pyramidal cell obtained from an intracellular sharp electrode recording. “timeaxis” gives the times (in sec) at which the values of Vm where sampled.

1) What is the frequency at which Vm was sampled at? (hint: look at timeaxis).

2) Set an appropriate threshold and compute the indices at which the membrane potential crosses this threshold from below. (hint: you can loop over the values of Vm with a command like “find(Vm(j)>thresh & Vm(j-1)<thresh)”). Finally, you can convert these indices into times by multiplying them by the sampling time step (i.e. the inverse of the sampling frequency).

3) Finally, make a binary representation of the spike train with the following time steps: 1 msec, 0.5 msec, 0.1 msec. Keep these at you’ll need them later. (hint: first figure out how many elements the binary vector will have. Then you have to you have to convert the spike times into the appropriate bin numbers, which you can do by “indices=round(spiketimes/binwidth)”, and then set the elements of the appropriate bins to 1/binwidth).

B) Interspike interval Statistics.

1) Compute the interspike interval sequence from the spike times (hint: the diff function may be useful or you can use a for loop).

2) Compute the interspike interval histogram with a binwidth of 1 msec going up to 200 msec.

3) How variable is this neuron? Answer the question by computing the CV. Is it more variable than a Poisson process?

4) Compute the interspike interval correlation coefficients. Is this spike train a renewal process?

C) Autocorrelation function and Power spectrum

1) Compute the autocorrelation functions of the binary representations that you computed in part A). From these, what is the neuron’s absolute refractory period? Does this neuron have a tendency to produce packets of action potentials?

2) Compute the power spectra of these binary representations. Remember that they are sampled at different frequencies. Compare these power spectra. What is the effect of changing the sampling frequency? What is the neuron's firing rate? Finally, compare these spectra with the one obtained from a Poisson process with the same firing rate.

Part II) Measures of Neural Encoding:

Load spiketrain2.mat and you'll see that the workspace now contains a variable called "data". A given neuron has been stimulated with "frozen noise". I.e. the stimulus has been repeated many times and each epoch lasts 50 sec. The stimulus is sampled at 2kHz and is the variable "stim" in the workspace. The stimulus is Gaussian white noise that has been low-pass filtered (8th order butterworth) at a cutoff frequency of 30 Hz, its power spectrum should be flat up to about 20 Hz.

The first column contains the trial indices, the second column contains the spike times (in msec), and the third column contains the interspike intervals (in msec).

A) Raster plots and PSTH

1) Plot the data in raster format. (hint: you simply have to plot the appropriate two columns of data, you may also have to zoom in).

2) Build a PSTH from the data with binwidth 1 msec. To do this, you have to make a vector of appropriate length (i.e. the length of one epoch divided by the binwidth) and count the number of spikes in each bin. Then divide that number by the number of epochs and by the binwidth to obtain the firing rate.

B) Cross-correlation function and transfer function

1) Make a binary representation of each trial at 2kHz and compute the cross-correlation function between trial 1 and the stimulus. Repeat for trial 20. Are the cross-correlation functions similar? Finally, compute the cross-correlation function averaged across all trial.

2) Compute the cross-spectrum between the stimulus and the spike train from epoch 1. Use the formula seen in class to estimate both gain and phase as a function of frequency. Normalize the gain by its value at f=1 Hz. Remember that the stimulus only has power up to about 30 Hz.

C) Signal-to-noise Ratio

1) Take the average across trials of the binary vector and compute its power spectrum. Now, subtract the average binary vector from each trial and compute the power spectrum, then average these over the trials. Finally, compute the signal-to-noise ratio as a function of frequency. Compare the shape of the gain and signal-to-noise ratio curves.