Auto-, reverse- and cross-correlation

a) Descriptions of the structure of a single spike train
   i) Interspike interval plot
      (1) Create a histogram showing the relative distributions of interspike intervals
         (a) Can be particularly useful for determining the quality of unit isolation
      (2) Log isi plot:
         (a) If s is the list of spike times and bin is the bin size in log units (0.01 is a good start):

```
%get the log of the list of intervals;
s = log(diff(s));
% nbins is the maximum difference / binsize + the negative bins
nbins = ceil(max(s) / bin);
if (min(s) < 0)
   bins = min(s):bin:(nbins * bin);
else
   bins = 0:bin:(nbins * bin);
end

i.data = zeros(length(bins)-1, 2);

[i.data(:,2) i.data(:,1)] = hist(s, bins);
i.data(:,1) = exp(i.data(:,1));
h = semilogx(i.data(:,1), i.data(:,2));
```

b) Autocorrelation
   i) For each spike, create a histogram of the times of other spikes relative to that spike
   ii) Divide by the number of spike to turn into a measure of the probability of seeing a spike at time t given that you saw a spike at time 0.
   iii) Useful for examining
      (1) Short time scale firing (bursting, refractory period)
      (2) Rhythmicity
c) Reverse correlation
   i) Given a visual (or auditory stimulus) and a spike train, how can we construct an estimate of the "preferred stimulus" of the neuron.
   ii) Example: retinal ganglion cell, stimulus with changing luminance over entire visual field, reverse correlation over time
      (1) Take a snapshot of the stimulus that precedes each spike
      (2) Average the snapshots
   iii) Example: spatio-temporal reverse correlation
      (1) White noise stimulus (flickering pixels, no correlations between pixels or between times)
      (2) Take snapshot of stimulus that precedes each spike going back in time. This produces a stack of 2D images.
      (3) Average each image at each time.
   iv) Math for one dimensional case:

(1) Spike triggered average stimulus

$$C(t) = \frac{1}{N}\sum_{i=1}^{N} S(u_i - t)$$

where $C(u)$ is the spike-triggered average stimulus

N is the number of spikes

u_i is the time of the i-th spike

$S(t)$ is the value of the stimulus at time t

Correlation between stimulus and spike train

$$Q_{rs}(t) = \frac{1}{T}\int_0^T \lambda(\tau) s(\tau + t)\, d\tau$$

where T is the total length of the experiment

$\lambda(\tau)$ is the firing rate at time τ

$s(\tau)$ is the value of the stimulus at time τ

$$C(t) = \frac{1}{\langle\lambda\rangle} Q_{rs}(-t)$$

v) We can also use the same concept of spike triggered average for comparing spikes and local field potentials by taking the average field potential in the neighborhood of a spike

vi) Related topic: smoothing

(1) It is often helpful to be able to smooth analysis to produce more realistic curves (e.g. smoothing a histogram to get rid of sharp bin edges)

(2) To smooth:

(a) Create a "filter" which sums to 1. The width of the filter will determine which values get averaged together to produce the smoothed estimate.

(i) Example filters

1. Boxcar

a. f = [.2 .2 .2 .2 .2] is a five point wide boxcar filter

2. Gaussian

a. f = gaussian(npoints, stdev) is a npoint side gaussian filter. Note that the f must sum to 1 to avoid changing the overall height of the original signal

(b) Convolve filter with data

(i) newhist = conv(f, orighist)

(c) Remove length(f) / 2 points from each side

(i) The convolution returns a vector whose length is the sum of the lengths of the filter and the original signal – 1, so we need to realign the data

(d) See smoothvect.m on the course website

(3) Issues: width of filter determines "frequency response"

(a) A wider filter emphasizes low frequencies.

d) The above are examples of correlations

i) Spike triggered average is the correlation between the signal and the firing rate of the neuron
ii) Cross correlations of spike trains
 (1) Histogram approach
  (a) Just like autocorrelegram, but using spike from train 1 as the reference and spikes from train 2 to make the histogram.
iii) Problem:
 (1) What constitutes a significant cross correlation?
 (2) Possible sources of correlation:
  (a) Similar behavioral correlates
  (b) Common input
  (c) Synaptic connection
 (3) We want to distinguish (a) from other possibilities
  (a) Determine whether neurons A and B are conditionally independent given the stimulus.
  P(A|B) may not equal P(A), but if A and B are independent, then P(A|B, stimulus) = P(A|stimulus)
iv) Normal approach:
 (1) Compare measured correlation to a many permuted (shuffled) test versions.
 (2) Given two simultaneously recorded neurons, compute cross correlation histogram
 (3) Using the same two neurons, shuffle the trials and recomputed the histogram over many shuffled versions.
  (a) For one shuffled version on five trials you might compute the cross correlation histogram comparing trials 1, 2, 3, 4 and 5 from neuron A to trials 3, 4, 5, 1 and 2 from neuron B.
 (4) Compare the measured cross correlation histogram to probability density produced by the shuffled trials
 (5) The correlation estimated from the shuffled trials is called the “signal correlation” because it represents the correlation resulting from the external signal (e.g. the stimulus)
 (6) The computed correlation is often called the “noise correlation” because it estimate the correlation within a trial that is not due to the stimulus.
 (7) This works well when you have defined trials, but is more difficult to apply to continuous recordings.
v) Brillinger method
 (1) Compute normal cross correlation histogram of B with respect to A
 (2) Transform the count in each bin as follows

$$newbin(i) = \sqrt{bin(i)/(binsize * nspikes_B)}$$

where $newbin(i)$ is the new value for bin i

$bin(i)$ is the original count in bin i

$binsize$ is the size of the bin in seconds

$nspikes_B$ is the number of spikes from neuron B

(3) Calculate a mean and a 95% confidence bound as follows:

$mean = \sqrt{nspikes_A / totaltime}$

where $nspikes_A$ is the number of spikes from neuron A

$totaltime$ is the total length of the period used for data collection

95% bounds: $mean \pm 1/\sqrt{binsize / nspikes_B}$

(4) This transformation changes the counts in each bin to be approximately normal, making it possible to compute confidence bounds.

e) Assumptions:

i) Non-dynamic data