Exercises

Chapter 1

1. Generate spikes for 10 s (or longer if you want better statistics) using a Poisson spike generator with a constant rate of 100 Hz, and record their times of occurrence. Compute the coefficient of variation of the interspike intervals, and the Fano factor for spike counts obtained over counting intervals ranging from 1 to 100 ms. Plot the interspike interval histogram.

2. Add a refractory period to the Poisson spike generator by allowing the firing rate to depend on time. Initially, set the firing rate to a constant value, $\mathrm{r}(t) = \mathrm{r}_0$. After every spike, set $\mathrm{r}(t)$ to 0, and then allow it to recover exponentially back to r_0 with a time constant τ_{ref} that controls the refractory recovery rate. In other words, have $\mathrm{r}(t)$ obey the equation
$$\tau_{\mathrm{ref}} \frac{d\mathrm{r}}{dt} = \mathrm{r}_0 - \mathrm{r}$$
except immediately after a spike, when it is set to 0. Plot the coefficient of variation as a function of τ_{ref} over the range 1 ms $\leq \tau_{\mathrm{ref}} \leq$ 20 ms, and plot interspike interval histograms for a few different values of τ_{ref} in this range. Compute the Fano factor for spike counts obtained over counting intervals ranging from 1 to 100 ms for the case $\tau_{\mathrm{ref}} = 10$ ms.

3. Compute autocorrelation histograms of spike trains generated by a Poisson generator with a constant firing rate of 100 Hz, a constant firing rate of 100 Hz together with a refractory period modeled as in exercise 2 with $\tau_{\mathrm{ref}} = 10$ ms, and a variable firing rate $\mathrm{r}(t) = 100(1 + \cos(2\pi t/25\text{ ms}))$ Hz. Plot the histograms over a range from 0 to 100 ms.

4. Generate a Poisson spike train with a time-dependent firing rate $\mathrm{r}(t) = 100(1 + \cos(2\pi t/300\text{ ms}))$ Hz. Approximate the firing rate from this spike train using a variable $\mathrm{r}_{\mathrm{approx}}$ that satisfies
$$\tau_{\mathrm{approx}} \frac{d\mathrm{r}_{\mathrm{approx}}}{dt} = -\mathrm{r}_{\mathrm{approx}}\,,$$
except that $\mathrm{r}_{\mathrm{approx}} \rightarrow \mathrm{r}_{\mathrm{approx}} + 1/\tau_{\mathrm{approx}}$ every time a spike occurs. Make plots of the true rate, the spike sequence generated, and the estimated rate. Experiment with a few different values of τ_{approx} in the range of 1 to 100 ms. Determine the best value of τ_{approx} by computing the average squared error of the estimate, $\int dt(\mathrm{r}(t) - \mathrm{r}_{\mathrm{approx}}(t))^2$, for different values of τ_{approx}, and finding the value of τ_{approx} that minimizes this error.

5. For a constant rate Poisson process, every specific (up to a finite resolution) sequence of N spikes occurring over a given time interval

is equally likely. This seems paradoxical because we certainly do not expect to see all N spikes appearing within the first 1% of the time interval. Resolve this paradox.

6. Build an approximate white-noise stimulus by choosing random values at discrete times separated by a time-step interval Δt. Plot its autocorrelation function and power spectrum (use the matlab® function `spectrum` or `psd`). Discuss how well this stimulus matches an ideal white-noise stimulus given the value of Δt you used.

7. Consider a model with a firing rate determined in terms of a stimulus $s(t)$ by integrating the equation

$$\tau_r \frac{d\mathrm{r}_{\mathrm{est}}(t)}{dt} = [\mathrm{r}_0 + s]_+ - \mathrm{r}_{\mathrm{est}}(t)\,,$$

where r_0 is a constant that determines the background firing rate and $\tau_r = 20$ ms. Drive the model with an approximate white-noise stimulus. Adjust the amplitude of the white-noise and the parameter r_0 so that rectification is not a big effect (i.e. $\mathrm{r}_0 + s > 0$ most of the time). From the responses of the model, compute the stimulus-response correlation function, $Q_{\mathrm{r}s}$. Next, generate spikes from this model using a Poisson generator with a rate $\mathrm{r}_{\mathrm{est}}(t)$, and compute the spike-triggered average stimulus from the spike trains produced by the white-noise stimulus. By comparing the stimulus-response correlation function with the spike-triggered average, verify that equation 1.22 is satisfied. Examine what happens if you set $\mathrm{r}_0 = 0$, so that the white-noise stimulus becomes half-wave rectified.

8. matlab® file `c1p8.mat` contains data collected and provided by Rob de Ruyter van Steveninck from a fly H1 neuron responding to an approximate white-noise visual motion stimulus. Data were collected for 20 minutes at a sampling rate of 500 Hz. In the file, `rho` is a vector that gives the sequence of spiking events or nonevents at the sampled times (every 2 ms). When an element of `rho` is one, this indicates the presence of a spike at the corresponding time, whereas a zero value indicates no spike. The variable `stim` gives the sequence of stimulus values at the sampled times. Calculate and plot the spike-triggered average from these data over the range from 0 to 300 ms (150 time steps). (Based on a problem from Sebastian Seung.)

9. Using the data of problem 8, calculate and plot stimulus averages triggered on events consisting of a pair of spikes (which need not necessarily be adjacent) separated by a given interval (as in figure 1.10). Plot these two-spike-triggered average stimuli for various separation intervals ranging from 2 to 100 ms. (Hint: in matlab® , use convolution for pattern matching: e.g. `find(conv(rho,[1 0 1])==2)` will contain the indices of all the events with two spikes separated by 4 ms.) Plot, as a function of the separation between the two spikes, the magnitude of the difference between the two-spike-triggered average and the sum of two single-spike-triggered averages (obtained

in exercise 8) separated by the same time interval. At what temporal separation does this difference become negligibly small. (Based on a problem from Sebastian Seung.)

10. Using the data of problem 8, find the spike-triggered average stimulus for events that contain exactly two adjacent spikes separated by various different intervals ranging from 2 to 100 ms (e.g. for 4 ms, the event `[1 0 1]` but not the event `[1 1 1]`). This is distinct from exercise 9 in which we only required two spikes separated by a given interval, but did not restrict what happened between the two spikes. Compare results of the exclusive case considered here with those of the inclusive two-spike-triggered average computed in exercise 9. In what ways and why are they different? (Based on a problem from Sebastian Seung.)