SYDE552/BIOL487 Assignment: Neuron Responses

25 marks total
Due March 15

This assignment can be done in Matlab or Python. Some code you will need is given on the next page. You will also need the data files *c1p8* and *MT-direction-tuning* (either .mat or .pkl for Matlab or Python, respectively).

Submit the following:

- A PDF file with the requested figures and brief explanations.
- Your Matlab or Python code

1. Tuning Curves
    a. Load the synthetic data file *MT-direction-tuning*. The file contains two variables: "direction" is a list of stimulus directions for 200 trials. "spikeTimes" contains spike times for each trial.
    b. Plot the spike raster and multi-trial firing rate (5ms bins) for 0° trials. Trial length is 2s.
    c. Plot together the single-trial rate estimate for trial 9 using a Gaussian kernels with SD=5ms and SD=50ms. (Use an appropriate sampling period of your choice for which the rate fluctuations are not visibly distorted in the plot.)
    d. Plot the tuning curve with standard deviation error bars using data from 50-250ms. (Tip: use a library function to find unique direction values.)
    e. Search the electrophysiology literature to find a (real) tuning curve from a mouse. Include and explain a figure that shows the tuning curve.
2. Spike-Triggered Averages
    a. Load the *c1p8* data file. This data is from Dayan & Abbott's web site and contains H1 neuron spike data collected by de Ruyter van Steveninck. There are two variables: "stim" is stimulus velocity, and "rho" is the response function (Δt=2ms).
    b. Plot the spike-triggered average stimulus. You can omit spikes that occur less than a window length after the start of recording.
    c. Generate 100 seconds of approximate white noise by drawing independent Gaussian-distributed samples every ms with mean 0 and SD 1. Save this noise for part (d) below. Use this as input to the function syntheticNeuron function defined below and calculate the spike-triggered average from the output.
    d. Create coloured noise by convolving the white noise you generated above with a Gaussian kernel (SD 20ms). Feed this signal into the syntheticNeuron function and calculate the spike-triggered average of this signal from the output. How and why is it different?

```python
import numpy as np
def synthetic_neuron(drive):
    """
    Simulates a mock neuron with a time step of 1ms.

    Arguments:
    drive - input to the neuron (expect zero mean; SD=1)

    Returns:
    rho - response function (0=non-spike and 1=spike at each time step)
    """
    dt = .001
    T = dt*len(drive)
    time = np.arange(0, T, dt)
    lagSteps = .02/dt
    drive = np.concatenate((np.zeros(lagSteps), drive[lagSteps:]))
    system = scipy.signal.lti([1], [.03**2, 2*.03, 1])
    _, L, _ = scipy.signal.lsim(system, drive[:,np.newaxis], time)
    rate = np.divide(30, 1 + np.exp(50*(.05-L)))
    spikeProb = rate*dt
    return np.random.rand(len(spikeProb)) < spikeProb
```

```
% rho = syntheticNeuron(drive) simulates a mock neuron with
a time
% step of 1ms.
%
% drive: input to the neuron (expect zero mean; SD=1)
% rho: response function (0=non-spike and 1=spike at each
time step)
function rho = syntheticNeuron(drive)
    if size(drive, 2) > size(drive, 1)
        drive = drive';
    end

    dt = .001;
    T = dt*length(drive);
    lagSteps = .02/dt;
    sys = tf([1], [.03^2 2*.03 1]);
    drive = [zeros(lagSteps,1); drive(1:end-lagSteps)];
    L = lsim(sys, drive, 0:dt:(T-dt));
    rate = 30 ./ (1 + exp(50*(.05-L)));
    spikeProb = rate*dt;
    rho = rand(size(spikeProb)) < spikeProb;
end
```