

BE 521: Homework 2 Questions

Modeling Neurons

Spring 2020

46 points

Due: Tuesday, 2/4/2020 11:59 PM

Objective: Computational modeling of neurons.

We gratefully acknowledge Dr. Vijay Balasubramanian (UPenn) for many of the questions in this homework.

Andrew Clark

Collaborators: Alex Silva, Jack Bellwoar, Vishal Tien.

1 Basic Membrane and Equilibrium Potentials (6 pts)

Before undertaking this section, you may find it useful to read pg. 153-161 of Dayan & Abbott's *Theoretical Neuroscience* (the relevant section of which, Chapter 5, is posted with the homework).

1. Recall that the potential difference V_T when a mole of ions crosses a cell membrane is defined by the universal gas constant $R = 8.31 \text{ J/mol} \cdot \text{K}$, the temperature T (in Kelvin), and Faraday's constant $F = 96,480 \text{ C/mol}$

$$V_T = \frac{RT}{F}$$

Calculate V_T at human physiologic temperature (37°C). (1 pt)

Example Latex math commands that uses the align tag to make your equations neater. You can also input math into sentences with \$ symbol like $\pi + 1$.

$$E = MC^2 \quad \text{(not aligned)}$$

$$E = MC^2 \quad \text{(aligned at = by the \&)}$$

$$1 = \frac{2}{2} \quad \text{(aligned at = by \&)}$$

```
R=8.31;  
T=310.15;  
F=96480;  
  
V_T=(R*T)/F *1000
```

```
V_T =  
  
26.7138
```

V_T at human physiologic temperature (37 °C), using $V_T = \frac{RT}{F}$ was calculated using the code above and is equal to 26.7 mV.

Use this value V_T to calculate the Nernst equilibrium potentials (in mV) for the K^+ , Na^+ , and Cl^- ions, given the following cytoplasm and extracellular concentrations in the squid giant axon: K^+ : (120, 4.5), Na^+ : (15, 145), and Cl^- : (12, 120), where the first number is the cytoplasmic and the second the extracellular concentration (in mM). (2 pts)

```
%Define ion concentrations  
K_cyto=120;  
K_extra=4.5;  
  
Na_cyto=15;  
Na_extra=145;  
  
Cl_cyto=12;  
Cl_extra=120;  
  
%Calculate Nernst equilibrium potentials  
  
Eq_K=((R*T)/(1*F))*log(K_extra/K_cyto))*1000  
Eq_Na=((R*T)/(1*F))*log(Na_extra/Na_cyto))*1000  
Eq_Cl=((R*T)/-(1*F))*log(Cl_extra/Cl_cyto))*1000
```

```
Eq_K =  
  
-87.7124  
  
Eq_Na =  
  
60.6051  
  
Eq_Cl =  
  
-61.5108
```

As calculated using the code above the Nernst equilibrium potential for the K^+ ions is -87.7 mV for the Na^+ ions it is 60.6 mV, and for the Cl^- it is -61.5 mV.

1. Use the Goldmann equation,

$$V_m = V_T \ln \left(\frac{P_K \cdot [K^+]_{out} + P_{Na} \cdot [Na^+]_{out} + P_{Cl} \cdot [Cl^-]_{in}}{P_K \cdot [K^+]_{in} + P_{Na} \cdot [Na^+]_{in} + P_{Cl} \cdot [Cl^-]_{out}} \right) \quad (1)$$

to calculate the resting membrane potential, V_m , assuming that the ratio of membrane permeabilities $P_K : P_{Na} : P_{Cl}$ is 1.0 : 0.05 : 0.45. Use the ion concentrations given above in Question 1.2. (2 pts)

```
%Define constants  
p_k=1;  
p_na=0.05;  
p_cl=0.45;
```

```
% Goldman Eqn
V_m_g_a=(V_T)*log((p_k*K_extra+p_na*Na_extra+p_cl*Cl_cyto)/(p_k*K_cyto+p_na*Na_cyto+p_cl*Cl_extra))
```

```
V_m_g_a =
-62.0123
```

As calculated using the code above resting membrane potential is equal to -62.0 mV under these conditions.

2. Calculate the membrane potential at the peak action potential, assuming a ratio of 1.0 : 12 : 0.45, again using the ion concentrations given in Question 1.2. (1 pt)

```
p_k=1;
p_na=12;
p_cl=0.45;

V_m_g_b=V_T*log((p_k*K_extra+p_na*Na_extra+p_cl*Cl_cyto)/(p_k*K_cyto+p_na*Na_cyto+p_cl*Cl_extra))
```

```
V_m_g_b =
42.6891
```

As calculated using the code above the resting membrane potential is equal to 42.7 mV under these conditions.

The amplitudes of the multi-unit signals in HW0 and local field potentials (LFP) in HW1 had magnitudes on the order of 10 to 100 microvolts. The voltage at the peak of the action potential (determined using the Goldman equation above) has a magnitude on the order of 10 millivolts. Briefly explain why we see this difference in magnitude. Hint1: Voltage is the difference in electric potential between two points. What are the two points for our voltage measurement in the multi-unit and LFP signals? What are the two points for the voltage measurement of the action potential? Hint 2: The resistance of the neuronal membrane is typically much higher than the resistance of the extracellular fluid. (2 pts)

We see this difference in magnitude for two reasons. First, according to Ohm's law voltage = current * resistance ($V=I*R$), so since the resistance of the neuronal membrane is typically much higher than the resistance of the extracellular fluid, the voltage measured at the peak of an action potential will be much higher than the voltages measured for the local field potentials and multi-unit signals from HW0. This is because voltage is proportional to resistance (if resistance increases voltage will increase if current stays constant), and we see this relationship from our $V=I*R$ equation. The second reason ties into the first: for the local field potentials the resistance was measured in the extra-cellular fluid at different parts in the brain, but for the action potentials the resistance was measured across the neuronal cell membrane, so these respective resistances did in fact influence the respective voltages measured.

2 Integrate and Fire Model (38 pts)

You may find it useful to read pg. 162-166 of Dayan and Abbott for this section. The general differential equation for the integrate and fire model is

$$\tau_m \frac{dV}{dt} = V_m - V(t) + R_m I_e(t)$$

where $\tau_m = 10$ ms is the membrane time constant, describing how fast the current is leaking through the membrane, V_m in this case is constant and represents the resting membrane potential (which you have already calculated in question 1.3.a), and $V(t)$ is the actual membrane potential as a function of time.

$R_m = 10^7 \Omega$ is the constant total membrane resistance, and $I_e(t)$ is the fluctuating incoming current. Here, we do not explicitly model the action potentials (that's Hodgkin-Huxley) but instead model the neuron's behavior leading up and after the action potential.

Use a $\Delta t = 10 \mu s$ (Δt is the discrete analog of the continuous dt). Remember, one strategy for modeling differential equations like this is to start with an initial condition (here, $V(0) = V_m$), then calculate the function change (here, ΔV , the discrete analog to dV) and then add it to the function (here, $V(t)$) to get the next value at $t + \Delta t$. Once/if the membrane potential reaches a certain threshold ($V_{th} = -50 \text{ mV}$), you will say that an action potential has occurred and reset the potential back to its resting value.

1. Model the membrane potential with a constant current injection (i.e., $I_e(t) = I_e = 2 \text{ nA}$). Plot your membrane potential as a function of time to show at least a handful of “firings.” (8 pts)

```
I_e = 2e-9; % I in amps

Tau=0.01; %tau in seconds

R_m=1*10^7; % R in ohms

delta_t=1e-5; % time step in seconds

V_th=-50e-3;% threshold in volts

time1=0:delta_t:0.1;
%Write a for loop to calculate the V at each 10 microsecond interval

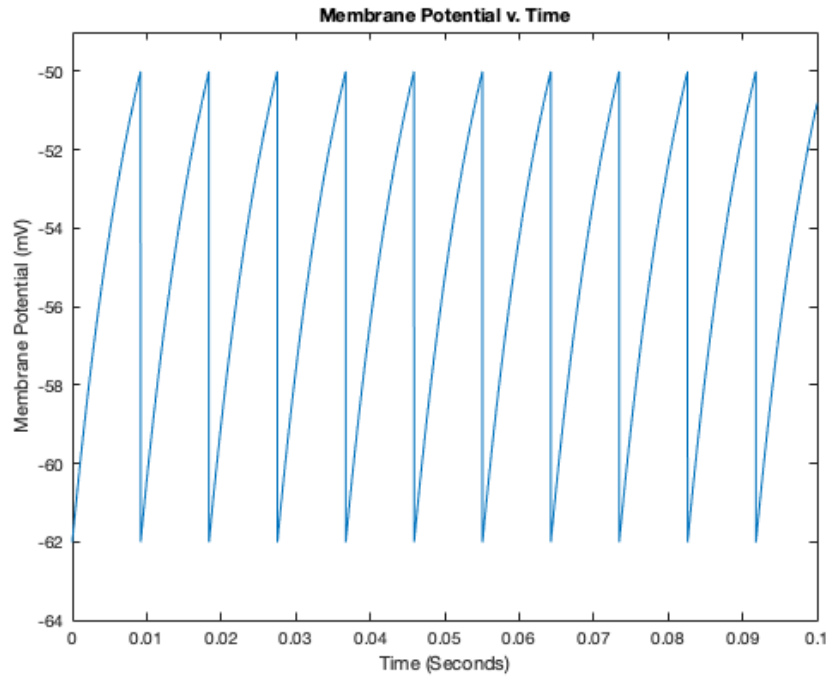
V=[];
V(1)=(V_m_g_a/1000);
num.peaks=0;

for i=2:length(time1)
    dv=(delta_t/Tau)*((V_m_g_a/1000)-V(i-1)+(R_m*I_e));
    dv;
    V(i)=dv+V(i-1);
    if V(i)>=V_th
        %V(i-1)=V_m_g_b/1000;
        num.peaks=num.peaks+1;
        V(i)=V_m_g_a/1000;
    end
end

%time=1:delta_t:length(V);

V=V.*1000; %convert to millivolts

figure
plot(time1,V)
ylabel('Membrane Potential (mV)')
xlabel('Time (Seconds)')
title('Membrane Potential v. Time')
ylim([-64 -49])
```



2. Produce a plot of firing rate (in Hz) versus injection current, over the range of 1-4 nA. (4 pts)

```

I_e = 2e-9; % I in amps

Tau=0.01; %tau in seconds

R_m=1*10^7; % R in ohms

delta_t=1e-5; % time step in seconds

V_th=-50e-3; % threshold in volts

time1=0:delta_t:1;
%Write a for loop to calculate the V at each 10 microsecond interval

V=[];
peak_vector=[];
V(1)=(V_m_g_a/1000);
%num_peaks=0;

currents=1e-9:1e-10:4e-9;

for j=1:length(currents)
    num_peaks=0;
    for i=2:length(time1)
        dv=(delta_t/Tau)*((V_m_g_a/1000)-V(i-1)+(R_m*currents(j)));
        dv;
        V(i)=dv+V(i-1);
        if V(i)>=V_th
            %V(i-1)=V_m_g_b/1000;
            num_peaks=num_peaks+1;
            %peak_vector(i)=num_peaks;
            V(i)=V_m_g_a/1000;
        end
    end
end

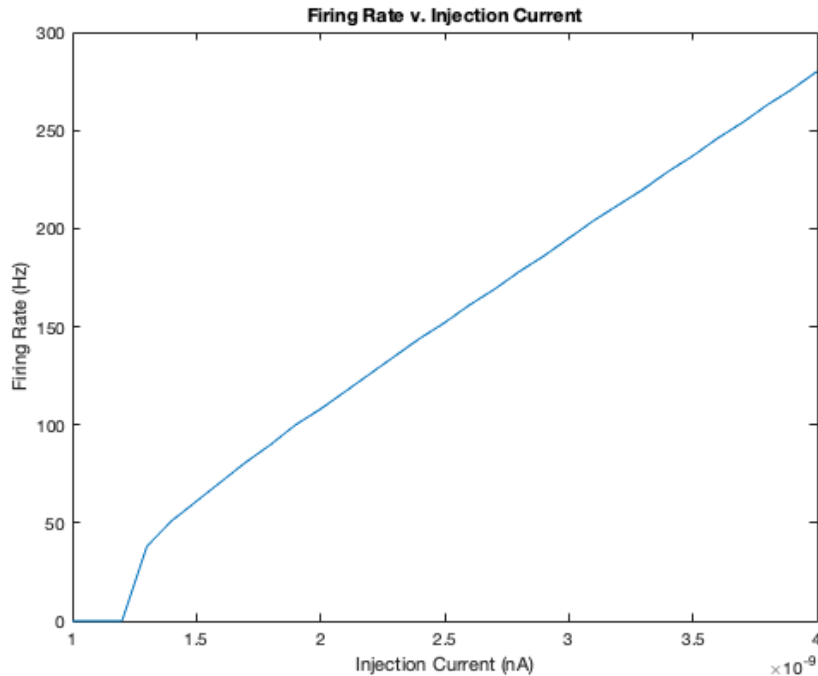
```

```

        end
        peak_vector =[peak_vector num_peaks];
    end

figure
plot(currents,peak_vector)
ylabel('Firing Rate (Hz)')
xlabel('Injection Current (nA)')
title('Firing Rate v. Injection Current')

```



3. I521.A0002.D001 contains a dynamic current injection in nA. Plot the membrane potential of your neuron in response to this variable injection current. Use Matlab's `subplot` function to place the plot of the membrane potential above the injection current so that they both have the same time axis. (Hint: the sampling frequency of the current injection data is different from the sampling frequency ($\frac{1}{\Delta t}$) that we used above.) (4 pts)

```

session = IEEGSession('I521.A0002.D001', 'andrewc', '/Users/andrewclark/Downloads/ieeg.password.bin' );

```

```

Warning: Objects of edu/upenn/cis/db/mefview/services/TimeSeriesDetails class
exist - not clearing java
Warning: Objects of edu/upenn/cis/db/mefview/services/TimeSeriesInterface class
exist - not clearing java
IEEGSETUP: Found log4j on Java classpath.
URL: https://www.ieeg.org/services
Client user: andrewc
Client password: ****

```

```
session.data
```

```
nr = ceil((session.data.rawChannels(1).get_tsdetails.getEndTime)/1e6*session.data.sampleRate);  
allData = session.data.getvalues(1:nr,1);
```

```
ans =
```

```
<a href="matlab:help('IEEGDataset')">IEEGDataset</a>:
```

```
    snapName: 'I521A0002.D001'  
    montage: As Recorded  
    filter: 'No Filter'  
    resample: 'Not resampled'  
    sampleRate: 1e+06  
    values:  
    channelLabels: [1x2 cell]  
    annLayer: []  
    rawChannels: [1x1 IEEGTimeseries]  
    allMontages: [1x44 IEEGMontage]
```

```
<a href="matlab:methods(IEEGDataset)">Methods</a>, <a href="matlab:IEEGObject.openPortalSite()">main.ieeg.
```

```
durationInUSec = session.data(1).rawChannels(1).get_tsdetails.getDuration;  
durationInSec = durationInUSec / 1e6;
```

```
%set integrate and fire model sampling rate  
sampling_rate.int_fire=(1/1e-5);  
  
%Downsample the current injection  
  
downsampled_currents=downsample(allData, 10);
```

```
downsampled_currents=downsampled_currents.*1e-9;  
  
I_e = 2e-9; % I in amps  
  
Tau=0.01; %tau in seconds  
  
R_m=1*10^7; % R in ohms  
  
delta_t=1e-5; % time step in seconds  
  
V_th=-50e-3;  
  
time=durationInSec;  
peak_vector_i_varied=[];  
  
time2=0:delta_t:(0.5-delta_t);  
  
V_new=[];  
V_new(1)=(V_m_g_a/1000);  
  
num_peaks=0;  
for i=2:length(time2)  
    dv=(delta_t/Tau)*((V_m_g_a/1000)-V_new(i-1)+(R_m*downsampled_currents(i)));  
    dv;
```

```

V_new(i)=dv+V_new(i-1);
if V_new(i)>=V_th
    num_peaks=num_peaks+1;
    V_new(i)=V_m_g_a/1000;
end
end

```

```

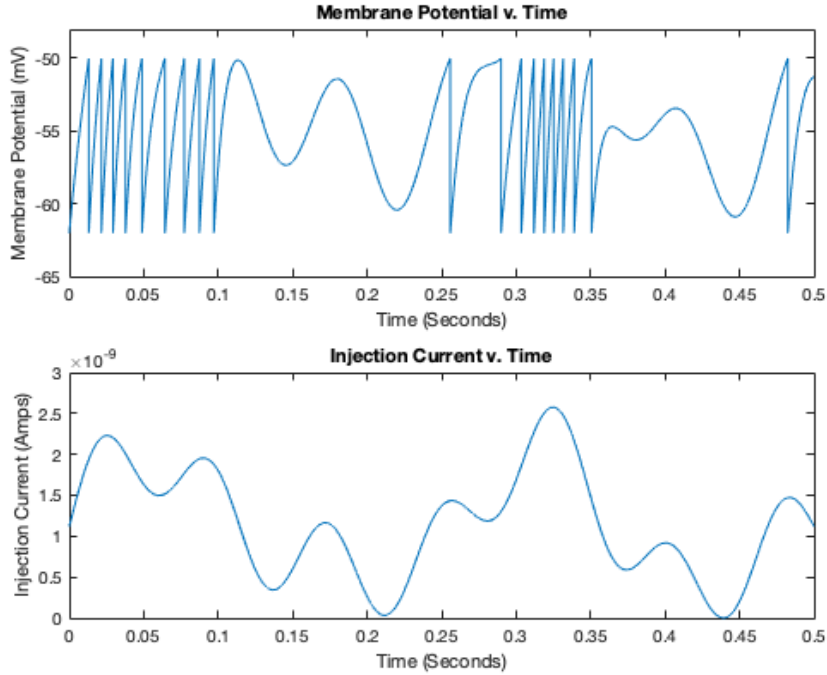
V_new=V_new.*1000;

```

```

figure
subplot(2,1,1)
plot(time2,V_new)
xlabel('Time (Seconds)')
ylabel('Membrane Potential (mV)')
title('Membrane Potential v. Time')
ylim([-65 -48])
%set(gcf, 'Position', [400, 400, 1000, 1000])
subplot(2,1,2)
plot(time2,downsampled_currents)
xlabel('Time (Seconds)')
ylabel('Injection Current (Amps)')
title('Injection Current v. Time')

```



4. Real neurons have a refractory period after an action potential that prevents them from firing again right away. We can include this behavior in the model by adding a spike-rate adaptation conductance term, $g_{sra}(t)$ (modeled as a potassium conductance), to the model

$$\tau_m \frac{dV}{dt} = V_m - V(t) - r_m g_{sra}(t)(V(t) - V_K) + R_m I_e(t)$$

where

$$\tau_{sra} \frac{dg_{sra}(t)}{dt} = -g_{sra}(t),$$

Every time an action potential occurs, we increase g_{sra} by a certain constant amount, $g_{sra} = g_{sra} + \Delta g_{sra}$. Use $r_m \Delta g_{sra} = 0.06$. Use a conductance time constant of $\tau_{sra} = 100$ ms, a potassium equilibrium potential of $V_K = -70$ mV, and $g_{sra}(0) = 0$. (Hint: How can you use the $r_m \Delta g_{sra}$ value to update voltage and conductance separately in your simulation?)

- (a) Implement this addition to the model (using the same other parameters as in question 2.1) and plot the membrane potential over 200 ms. (8 pts)

```
%Define Constants

I_e = 2e-9; % I in amps

Tau=0.01; %tau in seconds

tau_gsra= 0.1; % gsra tau in seconds

R_m=1*10^7; % R in ohms

delta_t=1e-5; % time step in seconds

V_th=-50e-3; % threshold in volts

rm_gsra_increase=0.06; %r_m gsra term (keep this grouped together)

V_k= -70e-3; % V_k in volts

time4=0:delta_t:0.2;

%For loop for iterating through time on the new equation

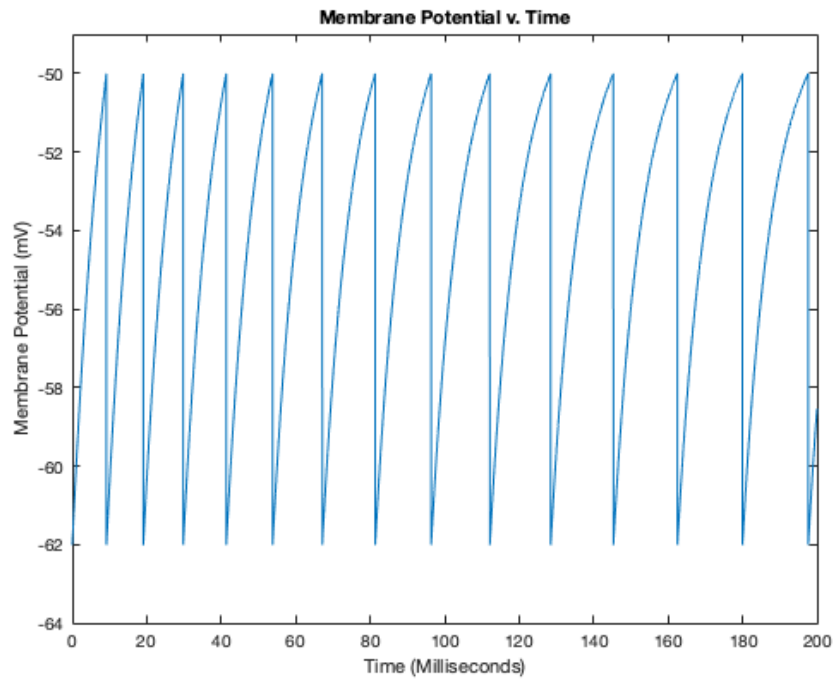
rm_gsra=[];
rm_gsra(1)=0;
V_4=[];
V_4(1)=(V_m_g_a/1000);
num_peaks=0;

for i=2:length(time4)
    %if rm_gsra <=
    dv_rm_gsra=(delta_t/tau_gsra)*(-rm_gsra(i-1));%R_m gsra differential eqn
    rm_gsra(i)=rm_gsra(i-1)+dv_rm_gsra;
    dv_4=(delta_t/Tau)*((V_m_g_a/1000)-V_4(i-1)-(rm_gsra(i-1)*(V_4(i-1)-V_k)))+(R_m*I_e));
    dv_4;
    V_4(i)=dv_4+V_4(i-1);
    if V_4(i)>=V_th
        rm_gsra(i)=rm_gsra(i)+rm_gsra_increase;
        %V(i-1)=V_m_g_b/1000;
        %num_peaks=num_peaks+1;
        V_4(i)=V_m_g_a/1000;%resting
    end
end
end
```

```
V_4=V_4.*1000;
```

```
time4=time4.*1000;
```

```
figure
plot(time4,V_4)
xlabel('Time (Milliseconds)')
ylabel('Membrane Potential (mV)')
ylim([-64 -49])
title('Membrane Potential v. Time')
```



- (b) Plot the inter-spike interval (the time between the spikes) of all the spikes that occur in 500 ms. (2 pts)

```
%Define Constants

I_e = 2e-9; % I in amps

Tau=0.01; %tau in seconds

tau_gsra= 0.1; % gsra tau in seconds

R_m=1*10^7; % R in ohms

delta_t=1e-5; % time step in seconds

V_th=-50e-3; % threshold in volts

rm_gsra_increas=0.06; %r_m gsra term (keep this grouped together)

V_k= -70e-3; % V_k in volts
```

```

time5=0:delta_t:0.5;

%For loop for iterating through time on the new equation

rm_gsra=[];
rm_gsra(1)=0;
V_5=[];
V_5(1)=(V_m_g_a/1000);
num_peaks=0;

for i=2:length(time5)
    %if rm_gsra <=
    dv_rm_gsra=(delta_t/tau_gsra)*(-rm_gsra(i-1));%R_m gsra differential eqn
    rm_gsra(i)=rm_gsra(i-1)+dv_rm_gsra;
    dv_4=(delta_t/Tau)*((V_m_g_a/1000)-V_5(i-1)-(rm_gsra(i-1)*(V_5(i-1)-V_k)))+(R_m*I_e));
    dv_4;
    V_5(i)=dv_4+V_5(i-1);
    if V_5(i)>=V_th
        rm_gsra(i)=rm_gsra(i)+rm_gsra_increase;
        %V(i-1)=V_m_g_b/1000;
        %num_peaks=num_peaks+1;
        V_5(i)=V_m_g_a/1000;%resting
    end
end

```

```

peak_threshold = -52;

[pks, locs]=findpeaks(V_5, 'MinPeakHeight', peak_threshold);

```

```

time_diff=time5(diff(locs)); %convert to seconds and find time difference

```

```

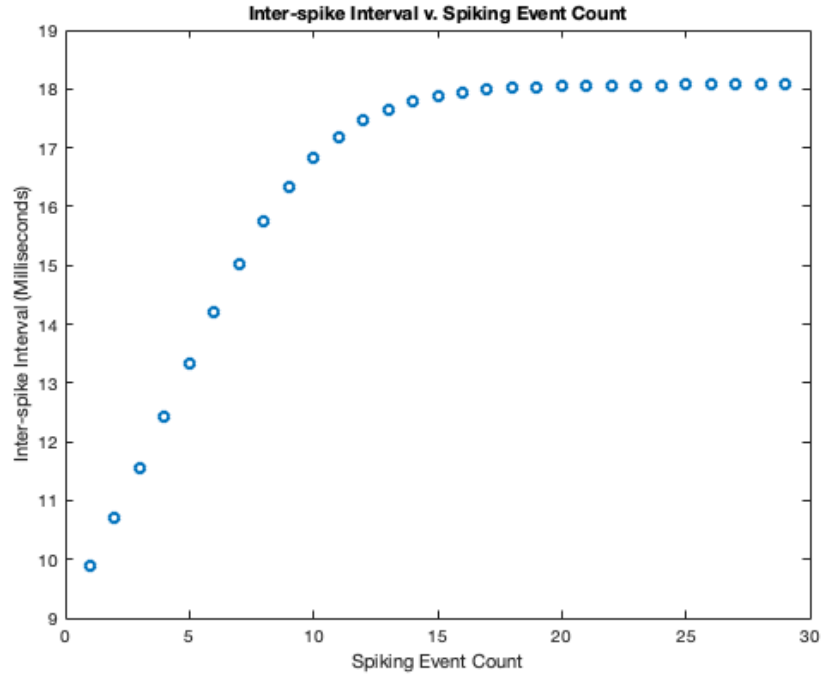
time_diff=time_diff.*1000; % convert to milliseconds

```

```

figure
plot(time_diff, 'o')
ylabel('Inter-spike Interval (Milliseconds)')
xlabel('Spiking Event Count')
title('Inter-spike Interval v. Spiking Event Count')

```



- (c) Explain how the spike-rate adaptation term we introduced above might be contributing to the behavior you observe in 2.4.b. (2 pts)

The spike-rate adaptation conductance term (termed " $g_{sra}(t)$ "), we introduced is intended to model the refractory period between action potentials. This refractory period is modeled by the $g_{sra}(t)$ term because the $g_{sra}(t)$ term increases the potassium ion conductance in the model, which increases the time needed before the next action potential ("spiking event") can occur. We can see this in our math because an increasing $g_{sra}(t)$ term will decrease the rate at which membrane potential increases (the absolute value of each " dv " will be decreased), which means more time (or in our simulation more loop iterations) will be needed before membrane potential can increase above the threshold and the next action potential spike can occur. It is also important to note that in the context of an actual neuron an increasing potassium conductance will allow potassium ions to flow across the membrane more easily, which will increase the time it takes for all ions to reset to the levels they need to be at for another action potential to occur.

5. Pursue an extension of this basic integrate and fire model. A few ideas are: implement the Integrate-and-Fire-or-Burst Model of Smith et al. 2000 (included); implement the Hodgkin-Huxley model (see Dayan and Abbot, pg. 173); provide some sort of interesting model of a population of neurons; or perhaps model what an electrode sampling at 200 Hz would record from the signal you produce in question 2.3. Feel free to be creative. We reserve the right to give extra credit to particularly interesting extensions and will in general be more generous with points for more difficult extensions (like the first two ideas), though it is possible to get full credit for any well-done extension.

- (a) Briefly describe what your extension is and how you will execute it in code. (6 pts)

The extension that will be pursued here will be the Hodgkin-Huxley model for the generation of an action potential as described by Dayan and Abbot in the 2001 book "Computational and Mathematical Modeling of Neural Systems." This model simulates the membrane potential of a neuron over time. It displays the behavior of the neuron membrane potential during an action potential. The model will be executed in code by numerically integrating the Hodgkin-Huxley Equations provided by Dayan and Abbot. Time will be discretized in order to recursively solve

each of the Hodgkin-Huxley equations provided by Dayan and Abbot. After defining initial conditions, the values of the membrane potential over time will be computed using a for loop and stored to a vector. After this computation is complete, plots of membrane potential v. time and activation probabilities v. time can be created. The code for this model is shown below.

Reference: Dayan, P., Abbott, L. F. (2001). Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems. The MIT Press.

```
%Define constants
p_k=1;
p_na=0.05;
p_cl=0.45;

% Goldman Eqn
V_m_g_a=(V.T)*log((p_k*K_extra+p_na*Na_extra+p_cl*Cl_cyto)/(p_k*K_cyto+p_na*Na_cyto+p_cl*Cl_extra));

%define constants
g_l = 0.003; % mS/mm^2
g_k= 0.36; % mS / mm^2
g_na = 1.2; % mS / mm^2

e_l = -54.387; % in mV
e_k = -77; %mv
e_na= 50; % mv

c_m = 0.01; % microF / mm^2
A = 0.1; % mm^2

I_e=0.01; % injection current

%Set dt
dt = 0.01; %10 microseconds in units of seconds

%time vector
time=0:dt:100; % 100 milliseconds

% v vector
V=zeros(1,length(time));
V(1)=(V_m_g_a); % resting membrane potential in millivolts

% i_m vector
i_m=zeros(1,length(time));
% n vector
n=zeros(1,length(time));
% m vector
m=zeros(1,length(time));
% h vector
h=zeros(1,length(time));

%set alpha n
alpha_n=(0.01*(V(1)+55)) / (1 - exp(-0.1 * (V(1) + 55)));

%set alpha m
alpha_m=(0.1*(V(1)+40)) / (1 - exp(-0.1 * (V(1) + 40)));

%set alpha h
alpha_h=0.07*exp(-0.05*(V(1)+65));

%set beta n
beta_n=0.125*exp(-0.0125*(V(1)+65));
```

```

%set beta m
beta_m=4*exp(-0.0556*(V(1)+65));

%set beta h
beta_h=1/(1+exp(-0.1*(V(1)+35)));

%set n of 1
n(1)=(alpha_n*V(1)) / (alpha_n*V(1) + beta_n*V(1));

%set m of 1
m(1)=(alpha_m*V(1)) / (alpha_m*V(1) + beta_m*V(1));

%set h of 1
h(1)=(alpha_h*V(1)) / (alpha_h*V(1) + beta_h*V(1));

```

```

for i = 2: length(time)

    %set alpha n
    alpha_n=(0.01*(V(i-1)+55)) / (1 - exp(-0.1 * (V(i-1) + 55)));

    %set alpha m
    alpha_m=(0.1*(V(i-1)+40)) / (1 - exp(-0.1 * (V(i-1) + 40)));

    %set alpha h
    alpha_h=0.07*exp(-0.05*(V(i-1)+65));

    %set beta n
    beta_n=0.125*exp(-0.0125*(V(i-1)+65));

    %set beta m
    beta_m=4*exp(-0.0556*(V(i-1)+65));

    %set beta h
    beta_h=1/(1+exp(-0.1*(V(i-1)+35)));

    % dn equation
    dn=(alpha_n*(1-n(i-1)) - (beta_n*n(i-1)))*dt;
    n(i)=n(i-1)+dn;

    % dm equation
    dm=(alpha_m*(1-m(i-1)) - (beta_m*m(i-1)))*dt;
    m(i)=m(i-1)+dm;

    % dh equation
    dh=(alpha_h*(1-h(i-1)) - (beta_h*h(i-1)))*dt;
    h(i)=h(i-1)+dh;

    %im equation
    im = (g_l*(V(i-1) - e_l)) + g_k*(n(i-1)^4)*(V(i-1)-e_k) + g_na*(m(i-1)^3)*h(i-1)*(V(i-1) - e_na);
    dv = (-im + (I_e/A))*(dt/c_m);
    V(i)=V(i-1)+dv;

end

```

(b) Provide an interesting figure along with an explanation illustrating the extension. (4 pts)

```

figure
subplot(2,1,1)
plot(time,V)
ylabel('Membrane Potential (mV)')
xlabel('Time (milliseconds)')

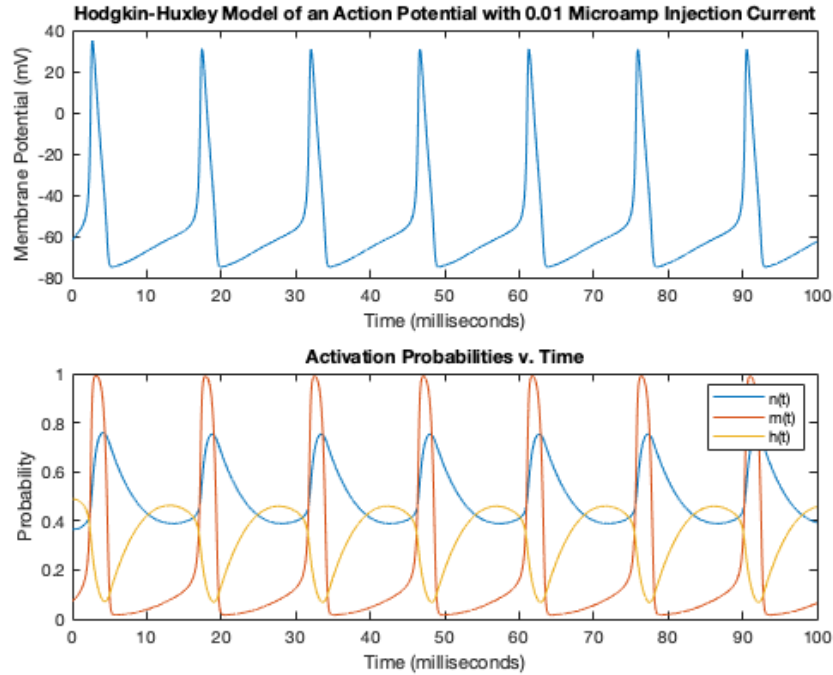
```

```

title('Hodgkin-Huxley Model of an Action Potential with 0.01 Microamp Injection Current')

%gating variable plot
subplot(2,1,2)
plot(time,n)
hold on
plot(time,m)
plot(time,h)
title('Activation Probabilities v. Time')
xlabel('Time (milliseconds)')
ylabel('Probability')
legend('n(t)', 'm(t)', 'h(t)')

```



The code above creates two plots. The top plot shows the Hodgkin-Huxley model of an Action Potential, with membrane potential on the y axis and time on the x axis. The viewer can see that once the membrane potential rises above the threshold an action potential ("spike") occurs and then the membrane potential decreases until a sufficiently long time has occurred before another action potential can be generated. The bottom plot shows the activation probabilities of the n , m , and h gating variables. We can see that each probability increases and decreases with respect to time. These increases and decreases correspond to the behavior of the membrane potential and the point in the action potential cycle that the neuronal cell is in, which is shown in the top plot.