Cathy Zhuang

Dr. Calabrese

BIOL 450

28 February 2022

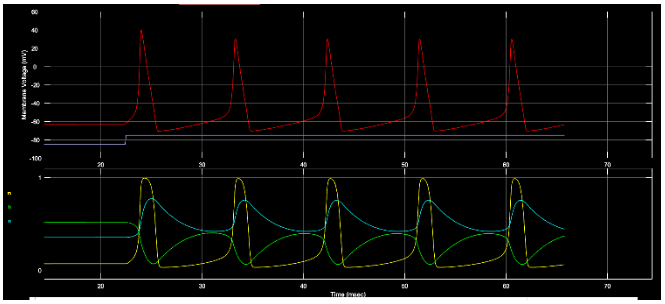
Assignment 5: Biophysical Neurons

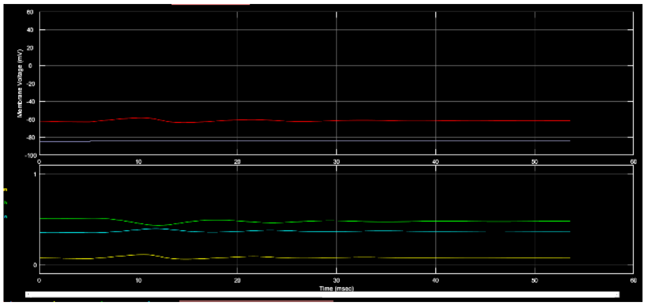
Part A

Question 1

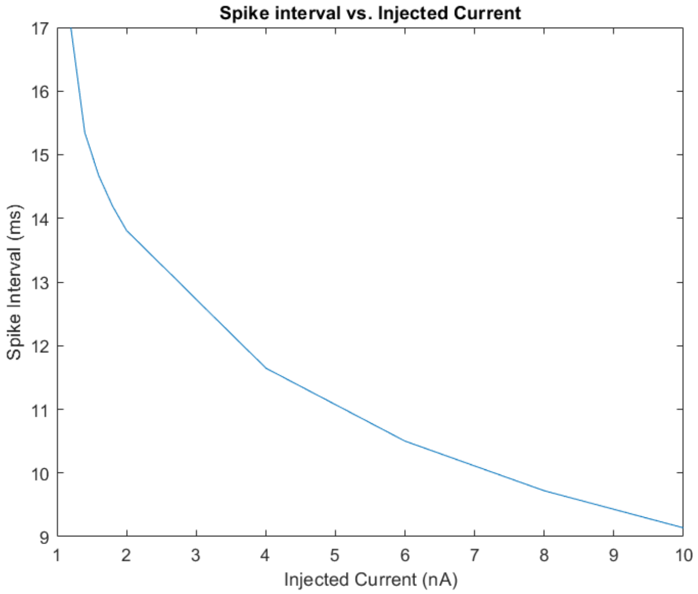
The longest interval that we elicited between successive spikes was around a 17-millisecond interval, which was given by an injected current of 1.2 nA. We found that after 1.2 nA, should one attempt a 1.19 nA injection, the spiking fails. Overall, we found that as the injected current decreases in amplitude, the spike intervals increase, or becomes longer.

Below are the plots for 10 nA and 1.19 nA, respectively. From this comparison, it is evident that spiking fails at 1.19 nA.



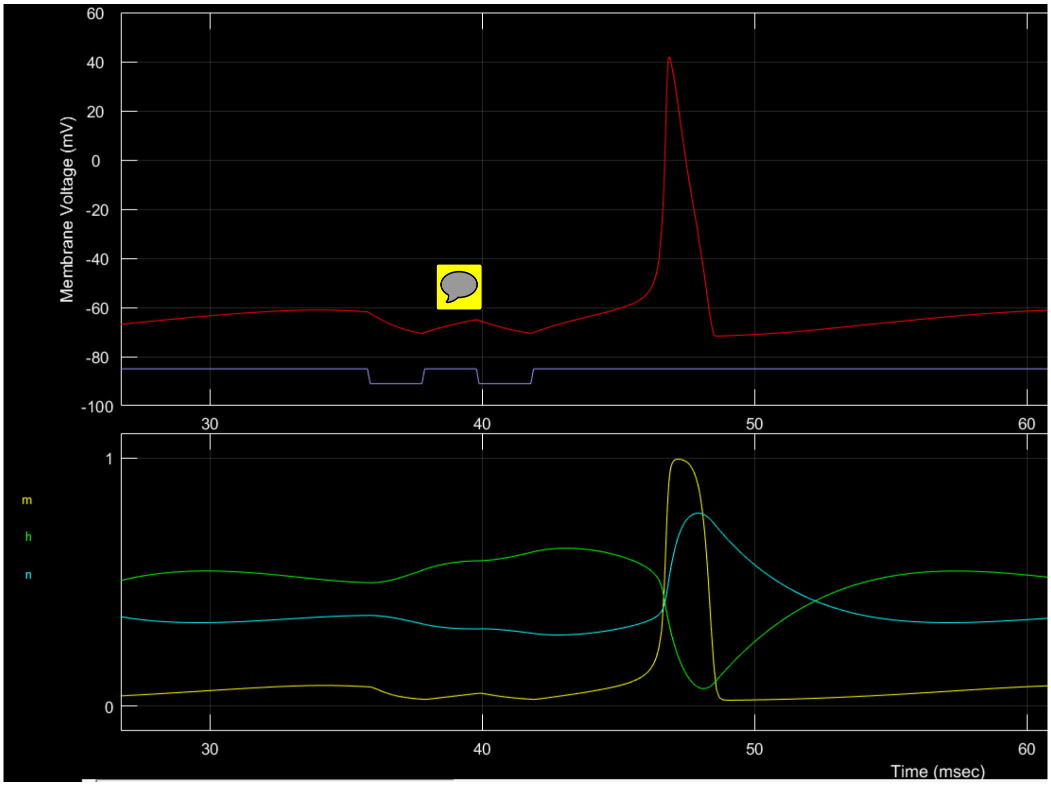


Below is the plot of the spike interval (Y axis) vs. injected current (X axis). As the injected current decreases in amplitude, the spike intervals increase in duration. This may be because an injection of a lower amplitude current means that it takes longer to charge the membrane and reach the threshold for an action potential. See **Appendix A** for the MATLAB code that produced this figure.



Question 2

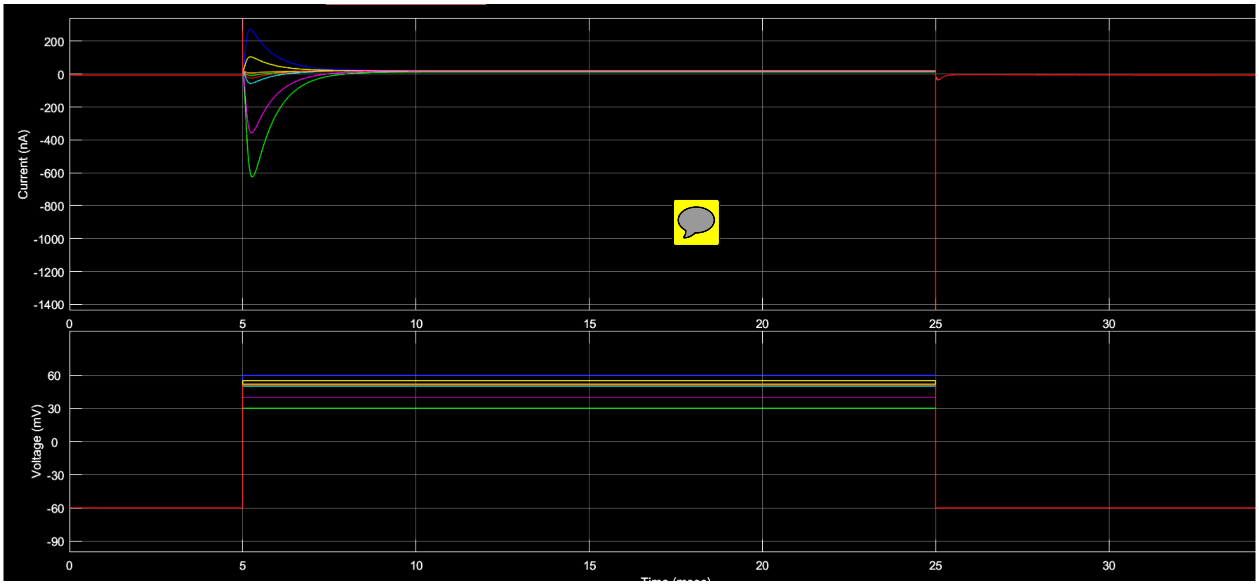
A spike is generated from the release from hyperpolarization because of the inactivation and activation gates of the voltage-gated sodium channels. When we apply the hyperpolarizing current, we observe that the membrane potential overall decreases. We also see m, which represents the activation gate, moves very close to zero, or closed. However, h, which represents the inactivation gate, increases, meaning that it opens. Then, when we remove the hyperpolarizing current, m begins increasing quickly, and h starts to decrease, but it does not have enough time to reach its resting value. This may be because the time constant for h is larger than m at this membrane potential, so it will take longer for h to reach its resting value. Thus, at a point, both the inactivation and activation gates are open, leading to a sodium current that flows inward, pushing towards the threshold for an action potential and causing a spike.



Part B

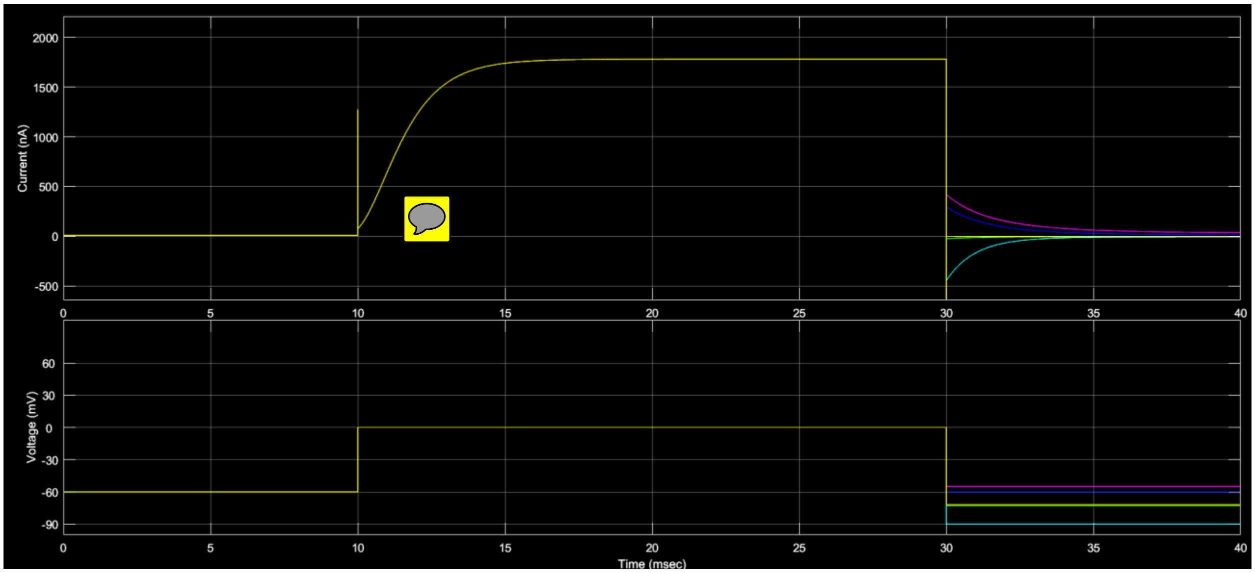
Question 1

After using TEA to inhibit 99-100% of the K+ current and using fast sodium, we ran the HHsim simulation with multiple voltage values to find where the current for fast sodium would equal net zero, as this is where the reversal potential would be. Our first voltage began at -60 mV, and then the second step varied, and then we returned to -60 mV. At a voltage of approximately 52.35 mV, the net current of Na+ becomes zero. Thus, the reversal potential for Na+ in this case is around 52.35 mV.



Question 2

After using TTX to inhibit 99-100% of the fast Na+ current, we ran HHsim with multiple voltage values to find where the reversal potential was. We did this first by using voltage steps of -60 mV, 0 mV, and then various voltages to observe the tail current of K+ and see which one reached a net current of zero the fastest. We found that, in this case, the reversal potential of K+ is approximately -72 mV.



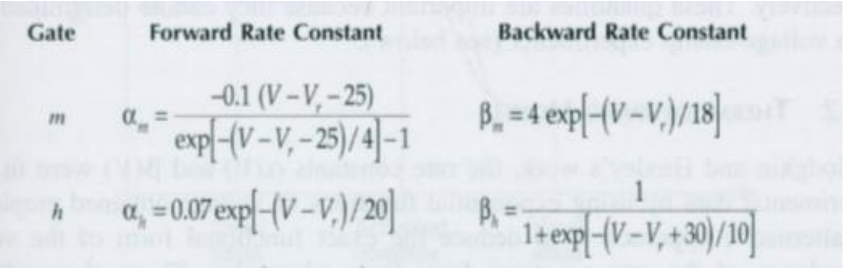
Question 3

We know that the steady state activation and inactivation characteristics of $g_{\mathrm{NaVG}}\ (\mathrm{Fast\ Sodium})$, which are represented respectively as m_∞ and h_∞ , can be modeled by the following equations[1]:

$$m_\infty = \frac{\alpha_m}{\alpha_m + \beta_m} \qquad (2.17)$$

$$h_\infty = \frac{\alpha_h}{\alpha_h + \beta_h} \qquad (2.18)$$

where alpha and beta represent the forward and backward rate constants. We also know that alpha and beta can be represented by:

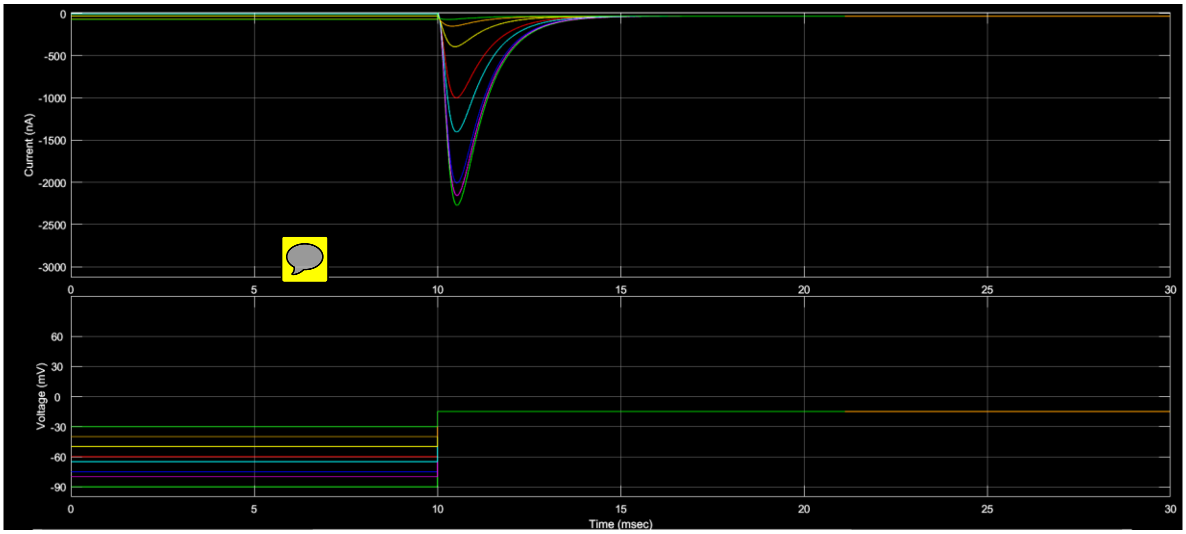


Gate	Forward Rate Constant	Backward Rate Constant
m	$\alpha_m = \dfrac{-0.1\,(V - V_r - 25)}{\exp[-(V - V_r - 25)/4] - 1}$	$\beta_m = 4\exp[-(V - V_r)/18]$
h	$\alpha_h = 0.07\exp[-(V - V_r)/20]$	$\beta_h = \dfrac{1}{1 + \exp[-(V - V_r + 30)/10]}$

with V_r representing the resting membrane potential.

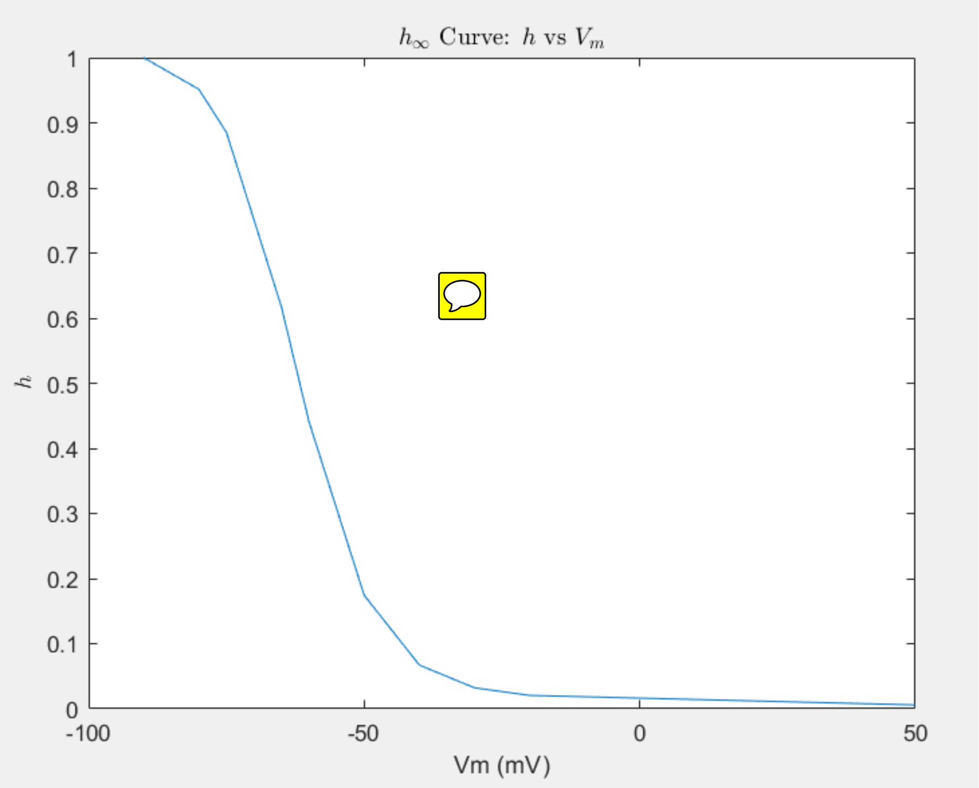
The variables h and m both represent aspects of the fast sodium channel. The variable m represents the activation characteristics of fast sodium, and the activation gates quickly open upon depolarization. In contrast, the variable h represents the inactivation gates, which close upon depolarization. Sodium can only flow through when both the activation and inactivation gates are open. Additionally, inactivation must be removed, often when the membrane is either repolarized or hyperpolarized.

We are able to model the h_∞ curve by plotting the relative peak size of the current versus the pre-pulse potential. In HHsim, we can accomplish this by first eliminating the potassium current through TEA, and then setting various pre-pulse potentials, ranging anywhere from -90 to 50



[1] Equations taken from Hille Ch. 2

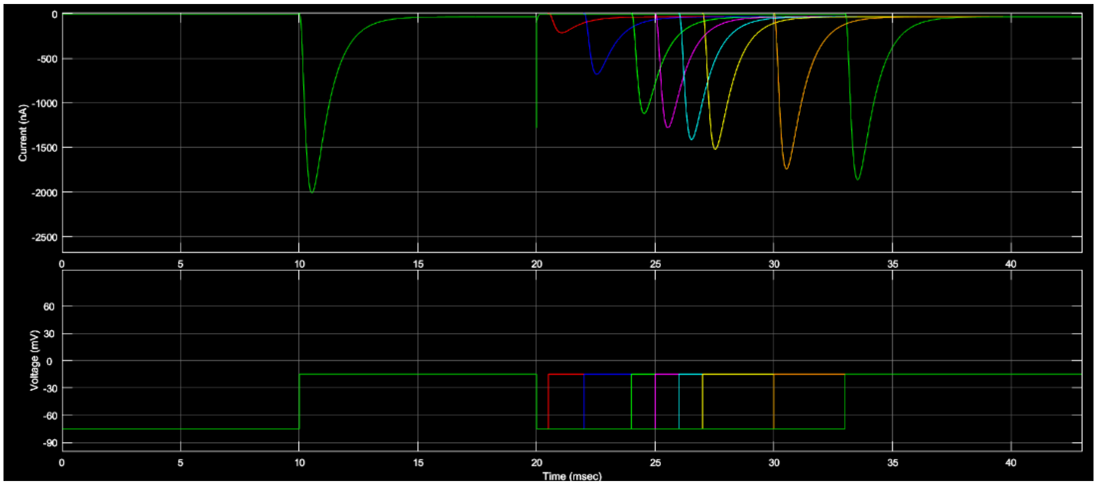
mV. Then, we initiate a second test pulse to produce the current. By measuring the currents, we can find the relative peak size and plot them against the pre-pulse potentials.



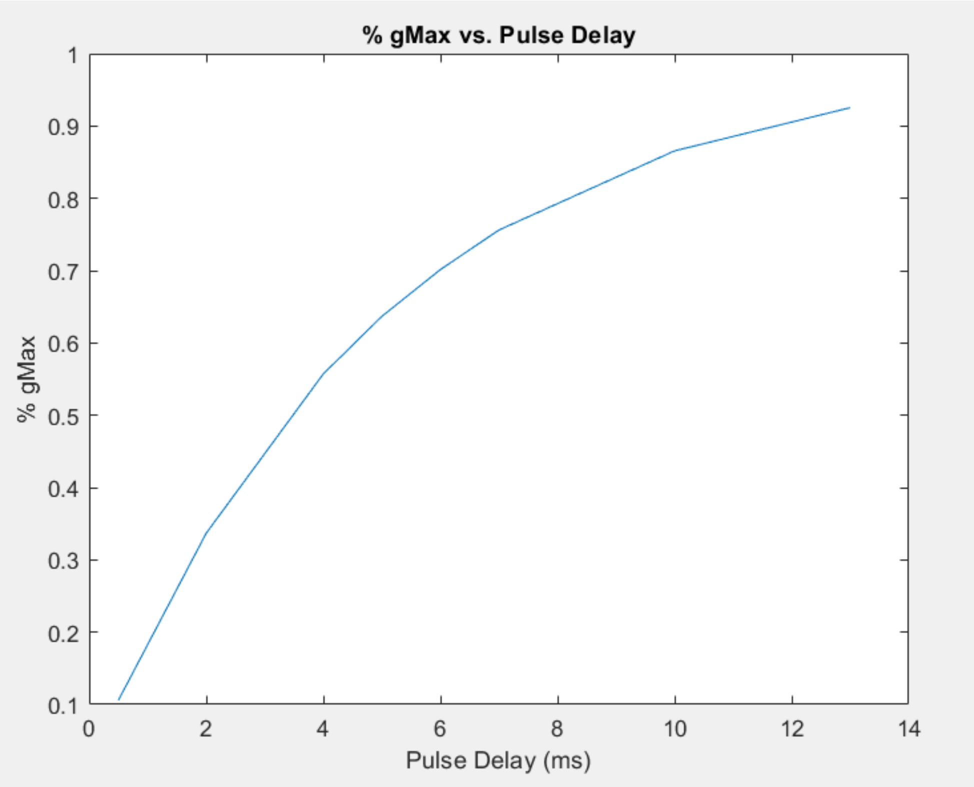
See **Appendix B** for the MATLAB code that produced this figure.

Question 4

To find the time course of recovery from sodium inactivation, we used a two-pulse experiment. The first pulse, from -75 mV to -15 mV, creates a current that is able to inactivate some Na+ channels. Then, we repolarize back to -75 mV to start de-inactivation. After varying the number of milliseconds that this potential is held, we depolarize back to -15 mV to see the proportion of channels that have recovered from inactivation, which is the amplitude of the current of pulse 2



divided by the amplitude of the current of pulse 1. We can then plot this over the duration of the pulse delay.

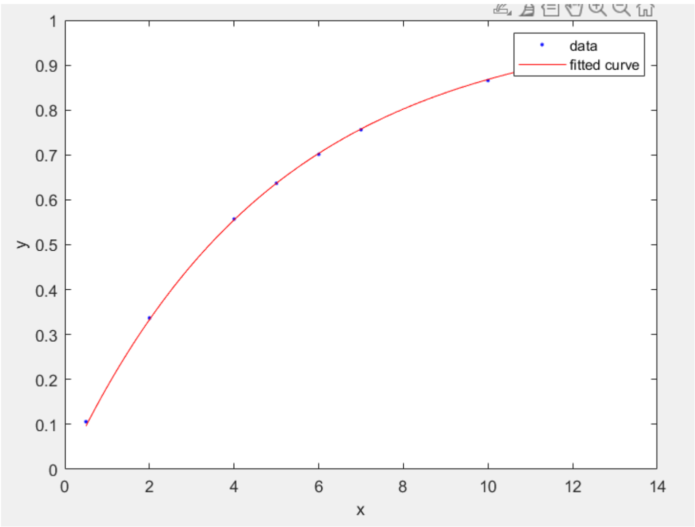


After plotting this graph, we notice that this is an exponential function, modeled by:

$$1 - e^{-t/\tau_h}$$

where τ_h is the time constant for inactivation of the Na+ channels.

After fitting the data to the function in MATLAB, we arrive at a τ_h of approximately 4.9. This τ_h is specific to the voltage steps conditions we have specified above.



```
General model:
val(x) = 1-exp(-x/tau)
Coefficients (with 95% confidence bounds):
  tau =       4.935  (4.875, 4.995)
```

See **Appendix C** for the MATLAB code that produced these figures and time constant calculations.

Question 5

To find n_∞, which represents the steady state activation characteristics of gKVg(Delayed Rectifier), we know that we can model it using the following equations[2]:

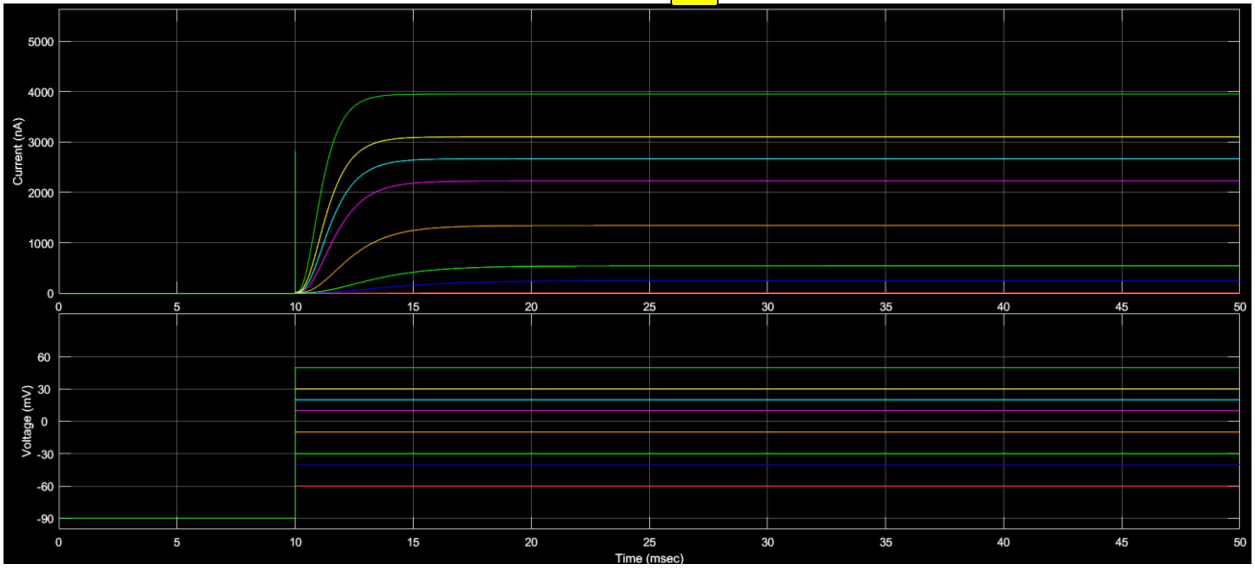
$$n_\infty = \frac{\alpha_n}{\alpha_n + \beta_n} \qquad (2.8)$$

where alpha and beta represent the forward and backward rate constants. We also know that alpha and beta can be represented by:

$$n \qquad \alpha_n = \frac{-0.01\,(V - V_r + 10)}{\exp[-(V - V_r + 10)/10] - 1} \qquad \beta_n = 0.125\exp[-(V - V_r)/80]$$

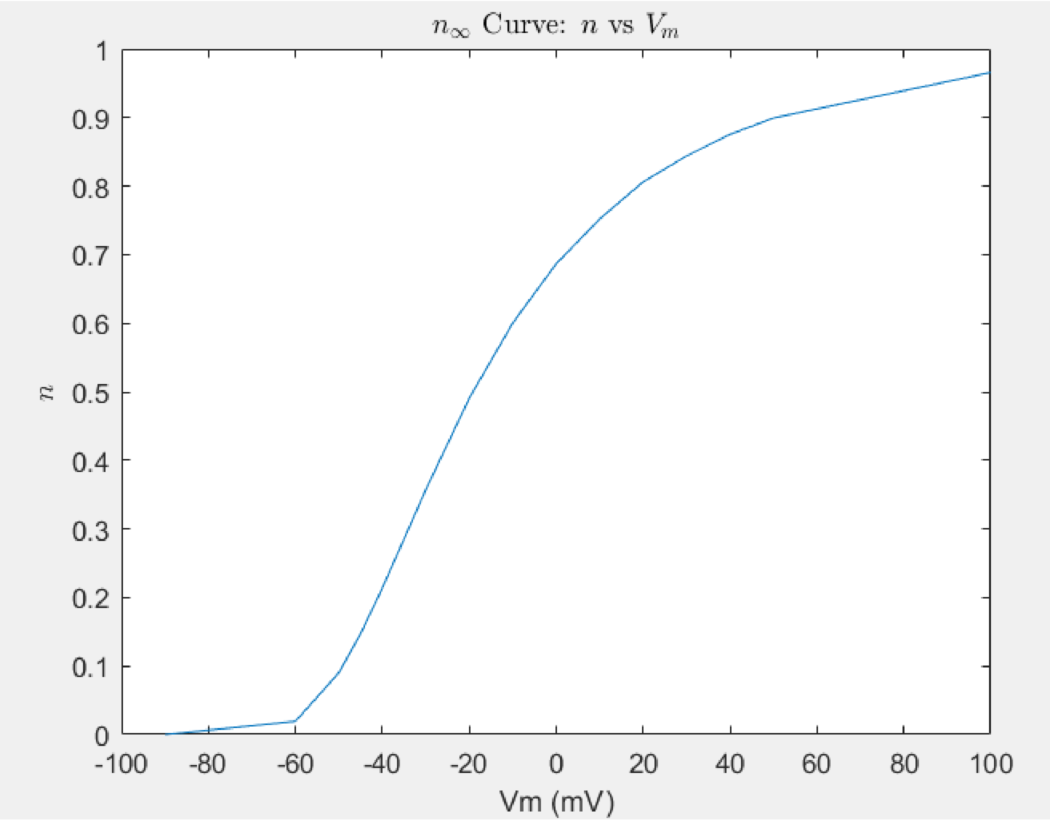
with V_r representing the resting membrane potential.

The variables n represents characteristics of the delayed rectifier. This means that, after depolarization and after a delay, there is a potassium current that repolarizes the membrane. To model the steady state n, we can plot the n over the voltages. In HHsim, we accomplish this by first eliminating any sodium using TTX, and then setting the first voltage to be low or around resting membrane potential (here we set it to -90). Then, we can use voltage steps to measure the amplitude of the potassium current.



After exporting the data into MATLAB, we can plot the steady state n with Vm.

[2] Equations taken from Hille Ch. 2



See **Appendix D** for the MATLAB code that produced this figure.

Appendices

Appendix A

Note: "one" through "nine" are the datasets exported from HHsim to MATLAB.

10 nA, 50 ms

```
[pks1, locs1] = findpeaks(one.('ecMemb'));
times1 = one(locs1,:);
toDelete = times1.raneVo < 10;
times1(toDelete,:) = [];
times1diff = mean(diff(times1.timems));
disp(times1diff)
```

8 nA

```
[pks2, locs2] = findpeaks(two.('ecMemb'));
times2 = two(locs2,:);
toDelete2 = times2.raneVo < 8;
times2(toDelete2,:) = [];
times2diff = mean(diff(times2.timems));
disp(times2diff)
```

6 nA

```
[pks3, locs3] = findpeaks(three.('ecMemb'));
times3 = three(locs3,:);
toDelete3 = times3.raneVo < 6 | times3.ecMemb < 0;
times3(toDelete3,:) = [];
times3diff = mean(diff(times3.timems));
disp(times3diff)
```

4 nA

```
[pks4, locs4] = findpeaks(four.('ecMemb'));
times4 = four(locs4,:);
toDelete4 = times4.raneVo < 4 | times4.ecMemb < 0;
times4(toDelete4,:) = [];
times4diff = mean(diff(times4.timems));
disp(times4diff)
```

2 nA

```
[pks5, locs5] = findpeaks(five.('ecMemb'));
times5 = five(locs5,:);
toDelete5 = times5.raneVo < 2 | times5.ecMemb < 0;
times5(toDelete5,:) = [];
times5diff = mean(diff(times5.timems));
disp(times5diff)
```

1.8 nA

```
[pks6, locs6] = findpeaks(six.('ecMemb'));
times6 = six(locs6,:);
toDelete6 = times6.raneVo < 1.8 | times6.ecMemb < 0;
times6(toDelete6,:) = [];
times6diff = mean(diff(times6.timems));
disp(times6diff)
```

1.6 nA

```
[pks7, locs7] = findpeaks(seven.('ecMemb'));
times7 = seven(locs7,:);
toDelete7 = times7.raneVo < 1.6 | times7.ecMemb < 0;
times7(toDelete7,:) = [];
times7diff = mean(diff(times7.timems));
disp(times7diff)
```

1.4 nA

```
[pks8, locs8] = findpeaks(eight.('ecMemb'));
times8 = eight(locs8,:);
toDelete8 = times8.raneVo < 1.4 | times8.ecMemb < 0;
times8(toDelete8,:) = [];
times8diff = mean(diff(times8.timems));
disp(times8diff)
```

1.2 nA

```
[pks9, locs9] = findpeaks(nine.('ecMemb'));
times9 = nine(locs9,:);
toDelete9 = times9.raneVo < 1.2 | times9.ecMemb < 0;
times9(toDelete9,:) = [];
times9diff = mean(diff(times9.timems));
disp(times9diff)
```

Graph

```
allTimes = [times1diff, times2diff, times3diff, times4diff, times5diff,
times6diff, times7diff...
    times8diff, times9diff];
allnanoAmps = [10, 8, 6, 4, 2, 1.8, 1.6, 1.4, 1.2];


plot(allnanoAmps, allTimes);
title('Spike interval vs. Injected Current');
xlabel('Injected Current (nA)');
ylabel('Spike Interval (ms)')
```

Appendix B

```
amplitude = [-2270, -2160, -2010, -1400, -1000, -395.72, -152.52, -73.15, -46.70,
-13.47];
prepulsevoltage = [-90, -80, -75, -65, -60, -50, -40, -30, -20, 50];
amplitude1 = amplitude./-2270;
plot(prepulsevoltage, amplitude1);
title('$h_\infty$ Curve: $h$ vs $V_m$', 'Interpreter', 'latex');
xlabel('Vm (mV)');
ylabel('$h$', 'Interpreter', 'latex');
```

Appendix C

```
pulseinterval = [0.5, 2, 4, 5, 6, 7, 10, 13];
amplitude2 = [-212.85, -677.16, -1120, -1280, -1410, -1520, -1740, -1860]
amplitude3 = amplitude2./-2010;
plot(pulseinterval, amplitude3);
title('% gMax vs. Pulse Delay');
xlabel('Pulse Delay (ms)');
ylabel('% gMax');




ft = fittype('1-exp(-x/tau)',...
    'dependent',{'y'},'independent',{'x'},...
    'coefficients',{'tau'});
f = fit(pulseinterval(:),amplitude3(:),ft, 'StartPoint', [3])
plot(f,pulseinterval, amplitude3)
```

Appendix D

```
voltages = [-90, -60, -50, -45, -40, -30, -20, -10, 0, 10, 20, 30, 40, 50, 100];
amplitude4 = [0, 8.37, 71.48, 143.06, 246.23, 540.43, 918.69, ...
     1340, 1780, 2220, 2670, 3100, 3530, 3950, 5980];


conductance = amplitude4./((voltages+72)*1000);
plot(voltages, conductance./0.036);
title('$n_\infty$ Curve: $n$ vs $V_m$', 'Interpreter', 'latex');
xlabel('Vm (mV)');
ylabel('$n$', 'Interpreter', 'latex');
```