%Ben Zhu bjz2107

t\_end = .5; % time of sim in seconds

dt = .0001;

t\_vec = 0:dt:t\_end; %vectorized time

N = length(t\_vec); %keep track of spike train dts

Vth = -.054;

Vrest = -.070;

Vreset = -.080;

Ie\_values = [1.5 1.600 1.602 1.605 1.610 1.615 1.620:.02:1.7 1.8:.1:2.2 2.5]\*10^-9;

Rm = 10^7;

tau = .010;

Vspike = 0.02

i = 0;

inject = zeros(1,N); %inject into this window

center = .1/dt:.4/dt; %window in dts

rate = zeros(size(Ie\_values)); %filler array for the rate values

for Ie = Ie\_values

i = i+1; %counter

V = Vrest\*ones(1,N);

inject(center) = Ie; % set electrode current at specified value in 300 ms window

for n = 2:N

if V(n-1) < Vth % want it to spike

V\_derivative = (Vrest - V(n-1) + inject(n)\*Rm)/tau; % eq. 5.8

V(n) = V(n-1) + dt\*V\_derivative;

elseif (V(n-1) > Vth) && (V(n-1) < Vspike) % this is the spike

V(n) = Vspike; % spike to this voltage

elseif V(n-1) == Vspike % that's it's spiked

V(n) = Vreset; %reset to this

end

end

rate(i) = sum(V == Vspike)/(length(center)\*dt); % find the rate at each point

figure(i)

plot(t\_vec,V,'-',t\_vec,Vreset\*ones(1,N),':',t\_vec,Vrest\*ones(1,N),':',t\_vec,Vth\*ones(1,N),':')

grid on

ylabel('Membrane Potential V', 'fontsize', 16)

set(gca,'YLim',[-.1 .04])

title(['Injected Current: ' num2str(Ie) ' A'])

end

figure(i)

plot(t\_vec,V,'-',t\_vec,Vreset\*ones(1,N),':',t\_vec,Vrest\*ones(1,N),':',t\_vec,Vth\*ones(1,N),':')

grid on

ylabel('Membrane Potential V', 'fontsize', 16)

set(gca,'YLim',[-.1 .0])

title(['Injected Current: ' num2str(Ie) ' A'])

figure(i+1)

plot(Ie\_values,rate,'-\*')

hold on

grid on

title('Interspike Interval Firing Rate vs Model Firing Rate', 'fontsize', 16)

ylabel('Rate')

xlabel('Injected Current (A))')

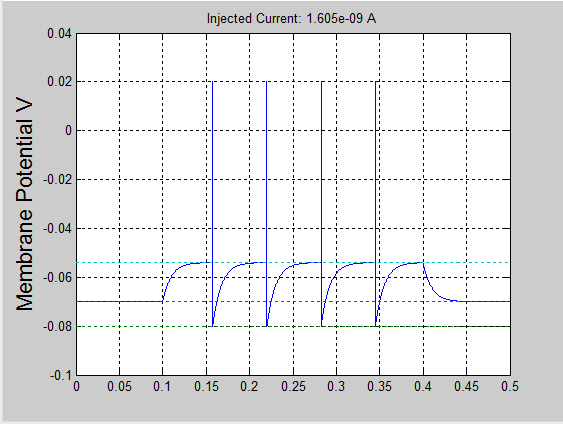
r\_isi = 1./(log(max(eps,(Vrest+Rm\*Ie\_values-Vreset)\*tau./(Rm\*Ie\_values+Vrest-Vth)))); %5.11

r\_isi(logical(Ie\_values<(Vth-Vrest)/Rm)) = 0;

plot(Ie\_values,r\_isi,'r')

hold off

legend('Model Results', 'Theoretical', 'location', 'best')

a)

This is just one example of an injected current that causes spikes to occur - this one is 1.605nA injected

PROBLEM 2

%Ben Zhu bjz2107

t\_end = .5;

dt = .0001;

Vrest = -.070; % resting potential

Vth = -.054; % firing threshold

Vreset = -.080; % reset potential

Ie\_values = [1.5 1.55 1.600 1.601 1.602 1.603 1.604 1.605:0.005:1.620 1.620:.02:1.7 1.7:.1:2.2 2.3 2.5]\*10^-9; % electrode current values (A)

Rm = 10^7; % membrane resistance (ohm)

tau\_m = .010; % membrane time constant (sec)

Vspike = 0.03;

%Using the new stuff

E\_k= -.070; % K+ equilibrium potential,

rmdgsra = .06;

tau\_sra = .1;

t\_vec = 0:dt:t\_end;

N = length(t\_vec);

i = 0;

inject = zeros(1,N); % electrode current will be zero everywhere but stimulus

center = .1/dt:.4/dt; %in dts

rate = zeros(size(Ie\_values));

for Ie = Ie\_values

i = i+1;

rmgsra = zeros(1,N);

V = Vrest\*ones(1,N);

inject(center) = Ie;

for n = 2:N

if (V(n-1) < Vth)

V\_derivative = (Vrest - V(n-1) + rmgsra(n-1)\*(E\_k - V(n-1)) + Rm\*inject(n))/tau\_m; %5.13

V(n) = V(n-1) + dt\*V\_derivative;

rmdgsraderiv = - rmgsra(n-1)/tau\_sra; %5.14

rmgsra(n) = rmgsra(n-1) + dt\*rmdgsraderiv;

elseif (V(n-1) > Vth) && (V(n-1) < Vspike)

V(n) = Vspike;

rmgsra(n) = rmgsra(n-1) + rmdgsra;

elseif V(n-1) == Vspike

V(n) = Vreset;

rmgsra(n) = rmgsra(n-1);

end

end

rate(i) = sum(V==Vspike)/(length(center)\*dt);

end

figure(i)

plot(t\_vec,V,'-',t\_vec,Vreset\*ones(1,N),':',t\_vec,Vrest\*ones(1,N),':',t\_vec,Vth\*ones(1,N),':')

grid on

ylabel('Membrane Potential V', 'fontsize', 16)

set(gca,'YLim',[-.1 .0])

title(['Injected Current: ' num2str(Ie) ' A'])

r\_isi = 1./(tau\_m\*log(max(eps,(Rm\*Ie\_values+Vrest-Vreset)./(Rm\*Ie\_values+Vrest-Vth)))); % equation 5.11

r\_isi(logical(Ie\_values<(Vth-Vrest)/Rm)) = 0;

figure(i+1)

plot(Ie\_values,rate,'-\*')

hold on

grid on

title('Interspike Interval Firing Rate vs Model Firing Rate', 'fontsize', 16)

ylabel('Rate')

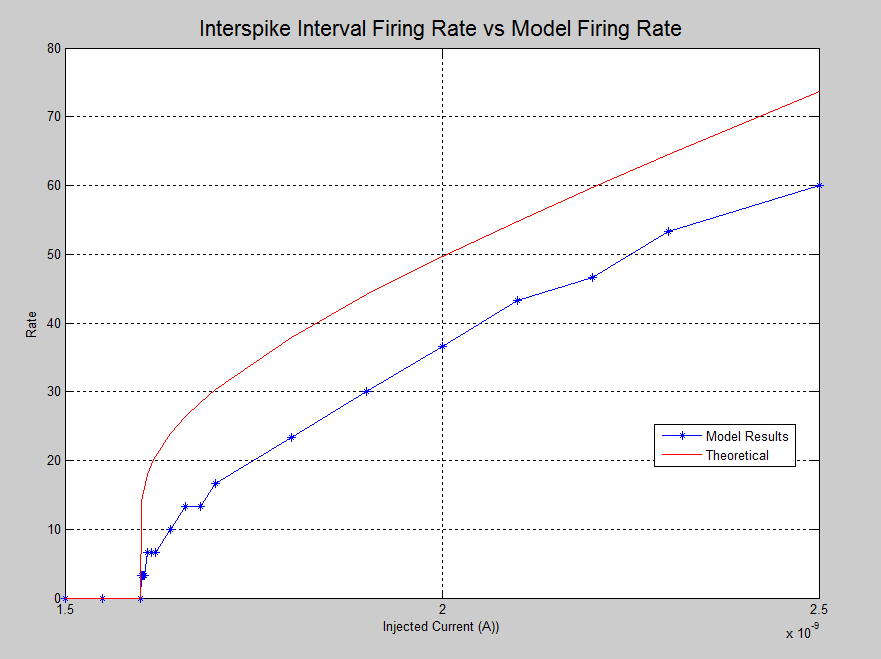
xlabel('Injected Current (A))')

plot(Ie\_values,r\_isi,'r')

hold off

legend('Model Results', 'Theoretical', 'location', 'best')

Added Spike Rate adaptation



Tested values from 1.5nA to about 2nA, and there were significant changes in the spike train around 1.601- 1.61 so did multiple points in that range to determine the simulation firing rate as opposed to the theoretical firing rate - which is also plotted here.

PROBLEM 3

%Ben Zhu

%bjz2107

%Problem number 3

t\_end = .5;

dt = .0001;

Vrest = -.070;

Vth = -.054;

Vreset = -.080;

tau\_m = .010;

Vspike = 0.03;

E\_s = 0; % synaptic reversal potential

r\_m\_gbar\_s = .5; % value of r\_m\*gbar\_s

tau\_s = .010; % synaptic decay

Pmax = .5; % msynaptic open probability

t\_vec = 0:dt:t\_end; % time axis

N = length(t\_vec); % number in the spike train

inc\_s = dt/tau\_s;

V = Vrest\*ones(1,N);

P\_s = zeros(1,N);

s\_input = zeros(1,N);

s\_current = zeros(1,N);

s\_input(round([.05 .15 .19 .3 .32 .4 .41]/dt)) = 1; %Trigger at these points

z = zeros(1,N);

for n = 2:N

sin = s\_input(n);

z(n) = sin + (1-sin)\*(1 - inc\_s)\*z(n-1);

P\_s(n) = (1 - inc\_s)\*P\_s(n-1) + inc\_s\*exp(1)\*Pmax\*z(n);

if V(n-1) < Vth

s\_current(n) = r\_m\_gbar\_s\*P\_s(n)\*(E\_s-V(n-1));

V\_derivative = (Vrest - V(n-1) + s\_current(n))/tau\_m; % eq. 5.43

V(n) = V(n-1) + dt\*V\_derivative;

elseif V(n-1) >= Vth && V(n-1) < Vspike

V(n) = Vspike;

else

V(n) = Vreset;

end

end

figure(1)

plot(t\_vec,z)

grid on

ylabel('z')

set(gca,'YLim',[0 1])

figure(2)

plot(t\_vec,P\_s)

grid on

ylabel('Synaptic Probability of Being Open')

set(gca,'YLim',[0 1])

figure(3)

plot(t\_vec,s\_current)

grid on

ylabel('Synaptic Current')

figure(4)

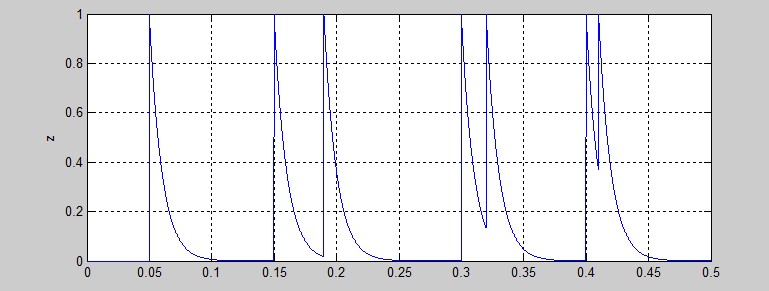
plot(t\_vec,V,'-',t\_vec,Vreset\*ones(1,N),':',t\_vec,Vrest\*ones(1,N),':',t\_vec,Vth\*ones(1,N),':')

grid on

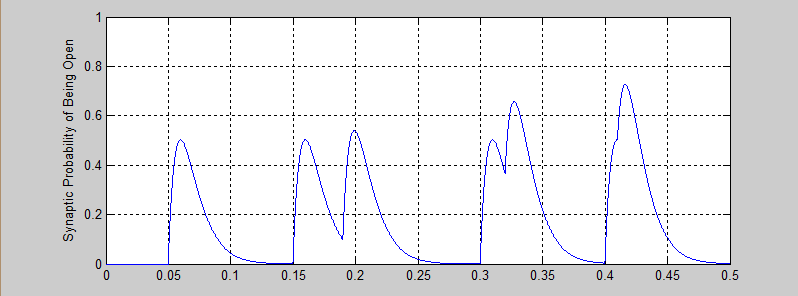
title('HOly')

ylabel('V')

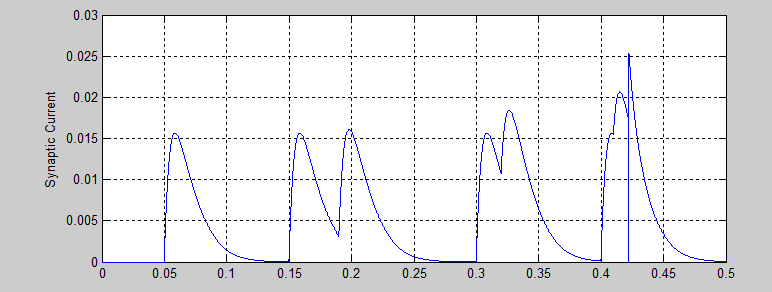
xlabel('Time (sec)')



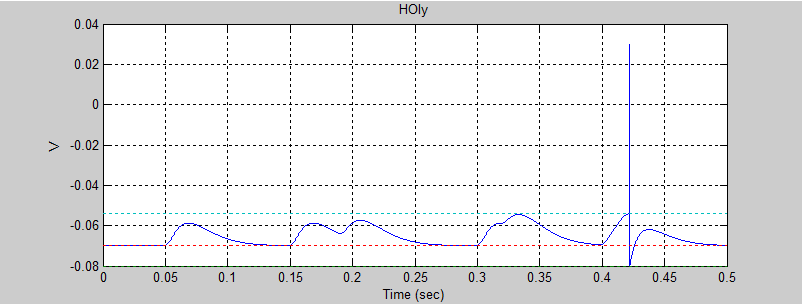
This figure represents the z's against time



This figure is the Synaptic Open Probability vs time



This is Synaptic Current vs Time



This is the V(t) given our input train of 05 .15 .19 .3 .32 .4 .41

Problem 4:

%Ben Zhu

%bjz2107

%Problem number 3

t\_end = .5;

dt = .0001;

Vrest = -.070;

Vth = -.054;

Vreset = -.080;

tau\_m = .010;

Vspike = 0.03;

E\_s = -80; % synaptic reversal potential

rmgbs = .5; % value of rmgbars

tau\_s = .010; % synaptic decay

Pmax = .5; % msynaptic open probability

t\_vec = 0:dt:t\_end; %time

N = length(t\_vec); % number in the spike train

inc\_s = dt/tau\_s;

V1 = Vrest\*ones(1,N);

P\_s1 = zeros(1,N);

s\_input1 = zeros(1,N);

s\_current1 = zeros(1,N);

%Second coupled one

V2 = Vrest\*ones(1,N);

P\_s2 = zeros(1,N);

s\_input2 = zeros(1,N);

s\_current2 = zeros(1,N);

s\_input1(round([.05 .15 .19 .3 .32 .4 .41]/dt)) = 1; %Trigger at these points

s\_input2;

z1 = zeros(1,N);

z2 = zeros(1,N);

for n = 2:N

sin1 = s\_input1(n);

sin2= s\_input1(n);

z1(n) = sin1 + (1-sin1)\*(1 - inc\_s)\*z1(n-1);

z2(n) = sin2 + (1-sin2)\*(1 - inc\_s)\*z2(n-1);

P\_s1(n) = (1 - inc\_s)\*P\_s1(n-1) + inc\_s\*exp(1)\*Pmax\*z1(n);

P\_s2(n) = (1 - inc\_s)\*P\_s2(n-1) + inc\_s\*exp(1)\*Pmax\*z2(n);

if V1(n-1) < Vth && V2(n-1) < Vth

s\_current1(n) = rmgbs\*P\_s1(n)\*(E\_s-V1(n-1));

s\_current2(n) = rmgbs\*P\_s2(n)\*(E\_s-V2(n-1));

V\_deriv1 = (Vrest - V1(n-1) + s\_current1(n))/tau\_m; % eq. 5.43

V1(n) = V1(n-1) + dt\*V\_deriv1;

V\_deriv2 = (Vrest - V2(n-1) + s\_current2(n))/tau\_m;

V2(n) = V2(n-1) + dt\*V\_deriv2;

elseif V1(n-1) >= Vth && V1(n-1) < Vspike && V2(n-1) >= Vth && V2(n-1) <Vspike

V1(n) = Vspike;

V2(n) = Vspike;

else

V1(n) = Vreset;

V2(n) = Vreset;

end

end

figure(1)

plot(t\_vec,z1)

grid on

ylabel('z')

set(gca,'YLim',[0 1])

figure(2)

plot(t\_vec,P\_s1)

grid on

ylabel('Synaptic Probability of Being Open')

set(gca,'YLim',[0 1])

figure(3)

plot(t\_vec,s\_current1)

grid on

ylabel('Synaptic Current')

figure(4)

plot(t\_vec,V1,'-',t\_vec,Vreset\*ones(1,N),':',t\_vec,Vrest\*ones(1,N),':',t\_vec,Vth\*ones(1,N),':')

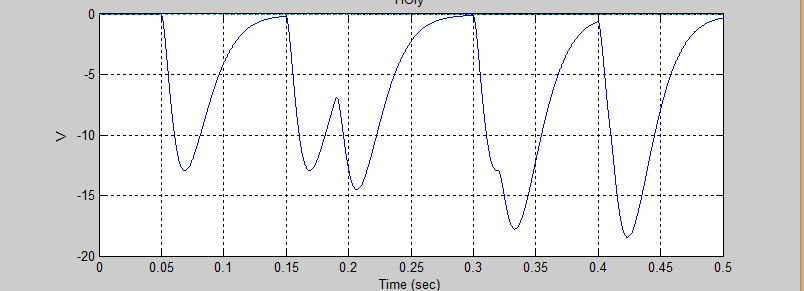
grid on

title('HOly')

ylabel('V')

xlabel('Time (sec)')

This was my attempt at getting them to couple together successfully... It did not work very well.



I don't really know what's going on here...

I tried a bunch of different things in the code in a way to successfully couple them but I wasn't able to really do it. With the different initial membrane voltages and Es being either inhibitory and excitatory, I was able to see a general trend where they were clearly firing and then would eventually reach an equilibrium where they would tend to the state that Es said they should be in - Inhibitory would cause the following neuron to be inhibitory as well. This is the expected behavior, but I was unable to reproduce this effectively.

Problem 5. Hodgkin Huxley simulation

%Current Stimulus

I\_external=zeros(1,4000);

I\_external(1000:1200)=9;

T=length(I\_external);

v=zeros(1,T);

dt=0.1;

C=10;

E\_K = 77;

E\_Na=50;

E\_L=-54.387;

G\_L=0.3;

G\_Na=120;

G\_K=36;

g\_L=G\_L;

g\_Na=zeros(1,T);

g\_K=zeros(1,T);

m=zeros(1,T);

n=zeros(1,T);

h=zeros(1,T);

n(1) = 0.3177;

m(1) = 0.0529;

h(1) = 0.5961;

v(1)=-65;

for t=2:T

v(t)= v(t-1)+(dt/C)\*(I\_external(t-1)-g\_L\*(v(t-1)-E\_L)-g\_K(t-1)\*(v(t-1)+E\_K)-g\_Na(t-1)\*(v(t-1)-E\_Na));

m(t)=m\_fun(m(t-1),v(t-1),dt);

n(t)=n\_fun(n(t-1),v(t-1),dt);

h(t)=h\_fun(h(t-1),v(t-1),dt);

g\_Na(t)=G\_Na\*(m(t)^3)\*h(t);

g\_K(t)=G\_K\*(n(t)^4);

end

figure

plot(v);

xlabel('Time');

ylabel('Membrane Potential V');

ANDDDD

function y=alpha\_n(V)

y=(0.01 \* ((V+55) / (1- exp(-0.1\*(V+55)) )));

function y=alpha\_m(V)

y=0.1\*( (V+40) / (1-exp(-0.1\*(V+40))));

function y=alpha\_h(V)

y=0.07\*exp(-0.05\*(V+65));

function y=beta\_n(V)

y=0.125\*exp(-0.0125\*(V+65));

function y=beta\_m(V)

y=4\*exp(-0.0556\*(V+65));

function y=beta\_h(V)

y=1/(1+ exp(-0.1\*(V+35)));

function n=n\_fun(n0,v,t)

n\_0=1 ./(1 + beta\_n(v) ./ alpha\_n(v));

tau\_n=1 ./(alpha\_n(v) +beta\_n(v));

n=n\_0 -(n\_0 -n0)\*exp(-t/tau\_n);

function m=m\_fun(m0,v,t)

m\_0=1 ./(1 + beta\_m(v) ./ alpha\_m(v));

tau\_m=1 ./(alpha\_m(v) +beta\_m(v));

m=m\_0 -(m\_0 -m0)\*exp(-t/tau\_m);

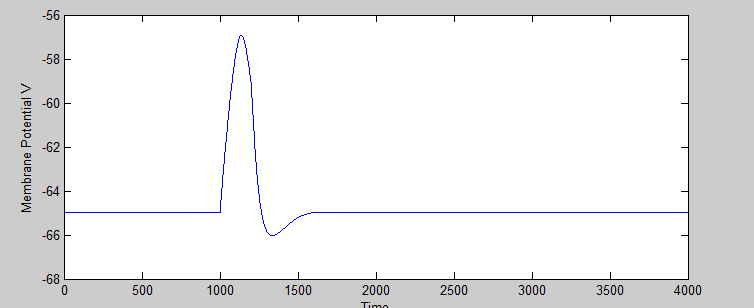
function h=h\_fun(h0,v,t)

h\_0=1 ./(1 + beta\_h(v) ./ alpha\_h(v));

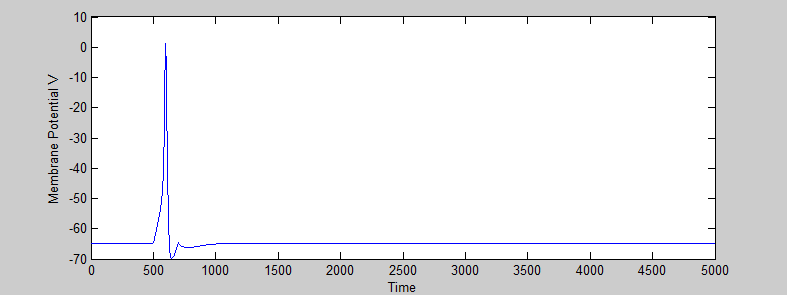
tau\_h=1 ./(alpha\_h(v) +beta\_h(v));

h=h\_0 -(h\_0 -h0)\*exp(-t/tau\_h);

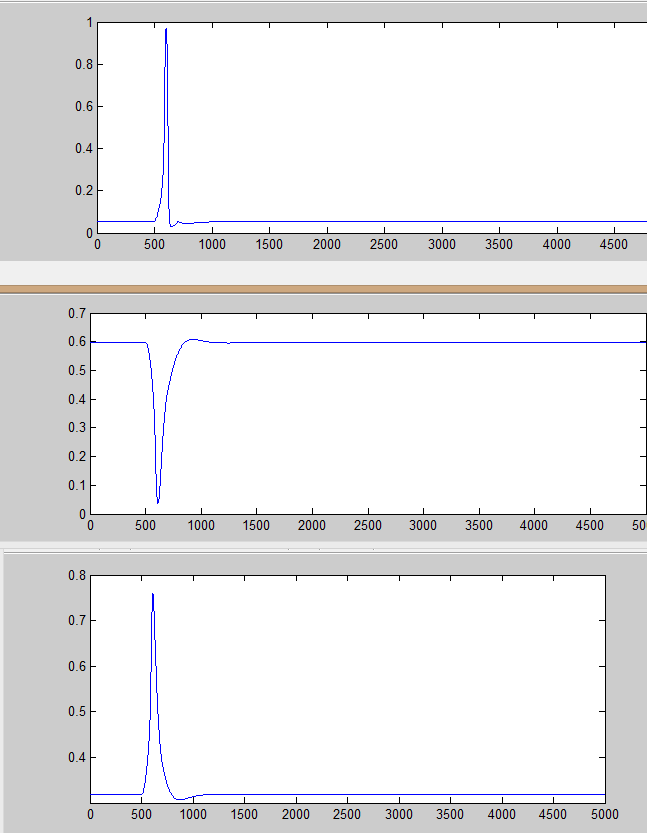
All the functions distilled into individual functions for easier calling in the loop.

I tried a different set of code where I iteratively solved the look for everything each time but it was impossible to debug so I went with this way instead.

So using 200nA

I had some issues. My simulation would not perfectly reach the expected voltage of E\_Na, but it was pretty close on each run.

This is V

These are m,h and n versus time... they're kinda weird but it's acceptable and makes sense for the displayed behavior of V

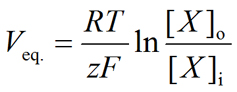
m

h

n

Problem 6:

OKAY SO FINALLY SOME MATH



We know that z = 2 for Mg2+, [x] outside = 200mM, and [x]inside = 15mM.

THEREFORE

**V membrane due to Mg2+ = 34.6 mV**

b) Set membrane potential at 20mV

Plug back into the above equation, 20mV = RT/zF (ln[xo] - ln[xi])

This tell us that internal concentratio**n of Mg2+ will be 22.38 mM**

7) I honestly have no clue how to do the wilson cowan model. I wrote the equations in matlab, plugged in some values, nothing works, no simulation works, this thing is a piece of junk