22S-CSB150-EEB159 Lab7

LISA WANG

TOTAL POINTS

96 / 100

QUESTION 1

1 Ex.1 20 / 20

✓ **- 0 pts** *Correct*

- 3 pts 1 or 2 lines wrong

```matlab
dyn.gK=pars.gKbar*dyn.n.^4;
dyn.gNa=pars.gNabar*dyn.m.^3.*dyn.h;
dyn.gL=pars.gL*ones(length(dyn.t),1);
dyn.IK=pars.gKbar*dyn.n.^4.*(dyn.V-pars.EK);
dyn.INa=pars.gNabar*dyn.m.^3.*dyn.h.*(dyn.V-pars.ENa);
dyn.IL=pars.gL*(dyn.V-pars.EL);
```

- 10 pts half wrong

```matlab
dyn.gK=pars.gKbar*dyn.n.^4;
dyn.gNa=pars.gNabar*dyn.m.^3.*dyn.h;
dyn.gL=pars.gL*ones(length(dyn.t),1);
dyn.IK=pars.gKbar*dyn.n.^4.*(dyn.V-pars.EK);
dyn.INa=pars.gNabar*dyn.m.^3.*dyn.h.*(dyn.V-pars.ENa);
dyn.IL=pars.gL*(dyn.V-pars.EL);
```

- 15 pts Almost all wrong

```matlab
dyn.gK=pars.gKbar*dyn.n.^4;
dyn.gNa=pars.gNabar*dyn.m.^3.*dyn.h;
dyn.gL=pars.gL*ones(length(dyn.t),1);
dyn.IK=pars.gKbar*dyn.n.^4.*(dyn.V-pars.EK);
dyn.INa=pars.gNabar*dyn.m.^3.*dyn.h.*(dyn.V-pars.ENa);
dyn.IL=pars.gL*(dyn.V-pars.EL);
```

- 20 pts All wrong. Correct solution is:

```matlab
dyn.gK=pars.gKbar*dyn.n.^4;
dyn.gNa=pars.gNabar*dyn.m.^3.*dyn.h;
dyn.gL=pars.gL*ones(length(dyn.t),1);
dyn.IK=pars.gKbar*dyn.n.^4.*(dyn.V-pars.EK);
dyn.INa=pars.gNabar*dyn.m.^3.*dyn.h.*(dyn.V-pars.ENa);
dyn.IL=pars.gL*(dyn.V-pars.EL);
```

QUESTION 2

2 Ex.2 20 / 20

✓ **- 0 pts** *Correct*

- 15 pts Code not correct; should use `n`, `m`, `h` instead of `dyn.n`, `dyn.m`, `dyn.h` as those are not defined yet

- 5 pts Code not correct

```matlab
Vdot = (1/pars.C)*(I-pars.gKbar*n^4*(V-pars.EK)-pars.gNabar*m^3*h*(V-pars.ENa) ...
-pars.gL*(V-pars.EL));
```

```
ndot = pars.alphan(V)*(1-n)-pars.betan(V)*n;
mdot = pars.alpham(V)*(1-m)-pars.betam(V)*m;
hdot = pars.alphah(V)*(1-h)-pars.betah(V)*h;
```

QUESTION 3

3 Ex.3 20 / 20

✓ **- 0 pts** *Correct*

- 10 pts Incorrect code

```matlab
for i=1:length(t),
if (t(i)>2 & t(i)<2.5)
I(i,1)=2;
elseif (t(i)>10 & t(i)<10.5)
I(i,1)=25;
else
I(i,1)=0;
end
end
```

QUESTION 4

4 Ex.4 16 / 20

- 0 pts Correct

- 4 pts Correct method

- 4 pts Correct values

✓ **- 4 pts** *Correct plot*

- 20 pts Incorrect

QUESTION 5

5 Ex.5 20 / 20

✓ **- 0 pts** *Correct*

- 5 pts Correctly set up input impulses: one long-weak and one short-strong

- 4 pts Only one correct plot

- 8 pts No correct plots

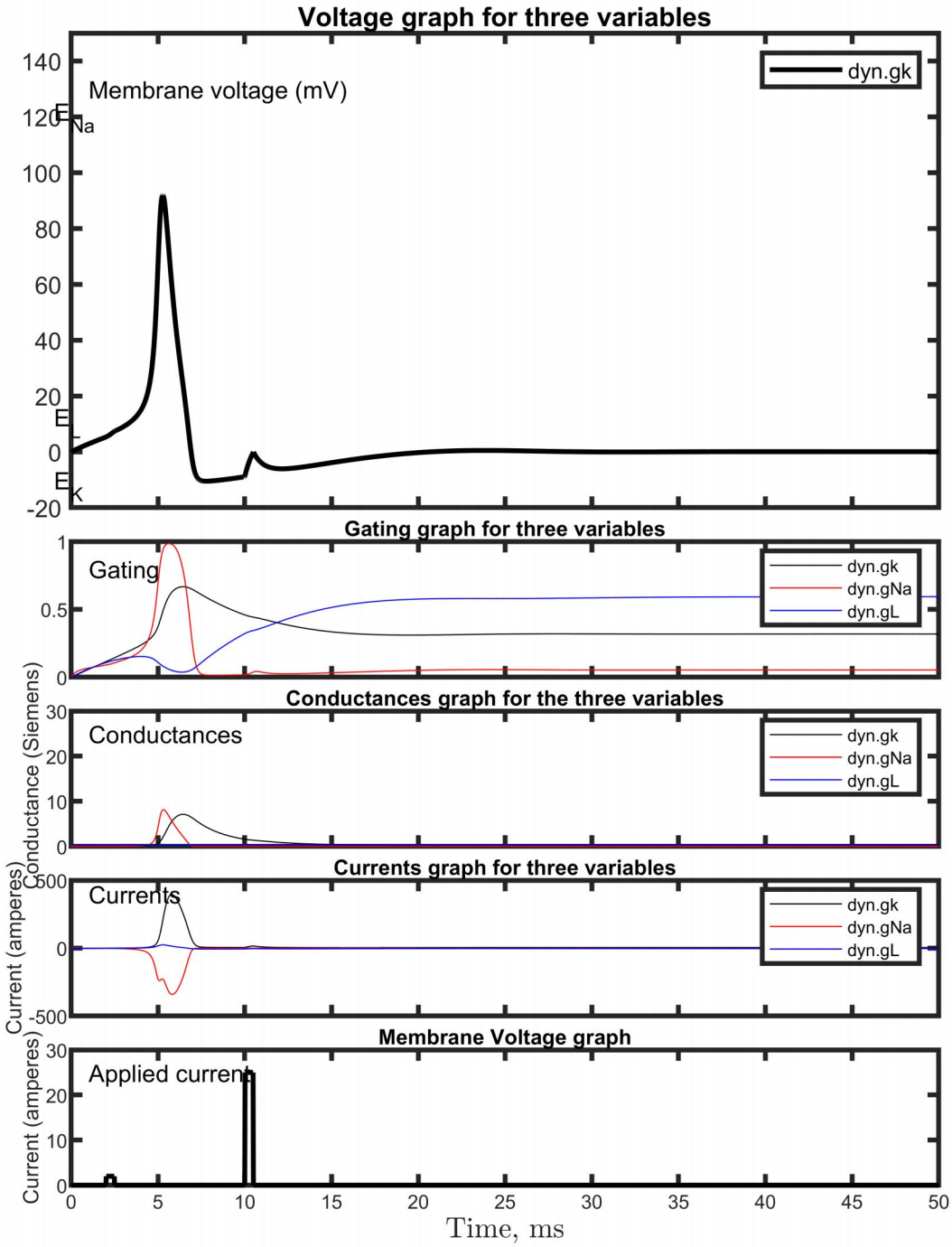
- 7 pts Conclusion: in general, short-strong should generate a signal while long-weak should not

- 1 pts Long-weak impulse generates signal (it should not)

```
%Exercise 1 - fill in the ...


% Joshua Weitz - BIOL 8814 - Neuron excitation - Fall 2018
%
% This simulates the HH model using parameters embedded in pars
% a stimulus current, and a time range over which
% the simulation should run

% Parameters, including on/off functions
pars.gKbar= 36; % mS/cm^2
pars.gNabar=120; % mS/cm^2
pars.gL=0.3; % mS/cm^2
pars.EK=-12; % mV
pars.ENa=120; % mV
pars.EL= 10.6; % mV
pars.C=1; % muF/cm^2
pars.alphan = @(V) 0.01*(10-V)./(exp(1-V/10)-1);
pars.betan = @(V) 0.125*exp(-V/80);
pars.alpham = @(V) 0.1*(25-V)./(exp(2.5-V/10)-1);
pars.betam = @(V) 4*exp(-V/18);
pars.alphah = @(V) 0.07*exp(-V/20);
pars.betah = @(V) (exp(3-V/10)+1).^-1;

% Initial conditions
pars.V0=0;
pars.n0= 0;
pars.m0 = 0;
pars.h0 = 0;

% Run the model
y0 = [pars.V0 pars.n0 pars.m0 pars.h0];
[t,y]=ode45(@model_hh,[0:0.02:50],y0,[],pars);

% Store results
% FILL IN WHEREVER YOU SEE ...
dyn.t=t;
dyn.V=y(:,1);
dyn.n=y(:,2);
dyn.m=y(:,3);
dyn.h=y(:,4);
dyn.gK=pars.gKbar*(dyn.n).^4;
dyn.gNa=(pars.gNabar)*(dyn.m).^3 .* (dyn.h);
dyn.gL=pars.gL * ones(length(dyn.t),1);
dyn.IK= pars.gKbar*(dyn.n).^4 .* (dyn.V-pars.EK);
dyn.INa= (pars.gNabar)*(dyn.m).^3 .* (dyn.h) .* (dyn.V - pars.ENa);
dyn.IL= pars.gL .* (dyn.V - pars.EL);
dyn.appliedI = impulse_t(dyn.t);
```

```
ploth = plot_hh(dyn, pars);
```

```
Warning: Ignoring extra legend entries.
```




```
% Thought problem:
% Here are my thoughts...

pars.gKbar= 36; % mS/cm^2
pars.gNabar=120; % mS/cm^2
```

```
pars.gL=0.3; % mS/cm^2
pars.EK=-12; % mV
pars.ENa=120; % mV
pars.EL= 10.6; % mV
pars.C=1; % muF/cm^2
pars.alphan = @(V) 0.01*(10-V)./(exp(1-V/10)-1);
pars.betan = @(V) 0.125*exp(-V/80);
pars.alpham = @(V) 0.1*(25-V)./(exp(2.5-V/10)-1);
pars.betam = @(V) 4*exp(-V/18);
pars.alphah = @(V) 0.07*exp(-V/20);
pars.betah = @(V) (exp(3-V/10)+1).^-1;

% Initial conditions
pars.V0=0;
pars.n0= pars.alphan(pars.V0) / (pars.alphan(pars.V0) + pars.betan(pars.V0)); %% n stars from e
pars.m0= pars.alpham(pars.V0) / (pars.alpham(pars.V0) + pars.betam(pars.V0)); %% plug in equili
pars.h0= pars.alphah(pars.V0) / (pars.alphah(pars.V0) + pars.betah(pars.V0)); % leave V(0) as i

% Run the model
y0 = [pars.V0 pars.n0 pars.m0 pars.h0];
[t,y]=ode45(@model_hh,[0:0.02:20],y0,[],pars);

% Store results
% FILL IN WHEREVER YOU SEE ...
dyn.t=t;
dyn.V=y(:,1);
dyn.n=y(:,2);
dyn.m=y(:,3);
dyn.h=y(:,4);
dyn.gK=pars.gKbar*(dyn.n).^4;
dyn.gNa=(pars.gNabar)*(dyn.m).^3 .* (dyn.h);
dyn.gL=pars.gL * ones(length(dyn.t),1);
dyn.IK= pars.gKbar*(dyn.n).^4 .* (dyn.V-pars.EK);
dyn.INa= (pars.gNabar)*(dyn.m).^3 .* (dyn.h) .* (dyn.V - pars.ENa);
dyn.IL= pars.gL .* (dyn.V - pars.EL);
dyn.appliedI = impulse_t(dyn.t);

ploth = plot_hh(dyn, pars);
```

```
Warning: Ignoring extra legend entries.
```

```
% generated. This shows how the neuron responds to an impulse (a sudden
% increase, positive feedback by the components) by culminating in an
% action potential.
```

```
function ploth = plot_hh(dyn,pars)
% function ploth = plot_hh(dyn,pars)
%
% Takes a HH output structure in dyn and plots
% dynamics of Voltage, gating variables, conductance
% currents and applied current
% Defaults
clf;
set(gcf,'DefaultLineMarkerSize',10);
set(gcf,'DefaultAxesLineWidth',2);
set(gcf,'PaperPositionMode','auto');
set(gcf,'Position',[300 40 600 750]);
% main data goes here
% Applied current on bottomA
tmppos= [0.1 0.1 0.8 0.1];
tmpa1 = axes('position',tmppos);
tmph=plot(dyn.t,dyn.appliedI,'k-');
set(tmph,'linewidth',2);
xlabel('Time, ms','fontsize',12,'verticalalignment','top','interpreter','latex');
text(1,25,'Applied current');
ylim([0 30]);
ylabel('Current (amperes)');
title('Membrane Voltage graph');

% Currents next
tmppos= [0.1 0.225 0.8 0.1];
tmpa2 = axes('position',tmppos);
tmph=plot(dyn.t,dyn.IK,'k-',dyn.t,dyn.INa,'r-',dyn.t,dyn.IL,'b-');
text(1,400,'Currents');
set(gca,'xticklabel',[]);
ylim([-500 500]);
ylabel('Current (amperes)')
legend('dyn.gk', 'dyn.gNa', 'dyn.gL');
title('Currents graph for three variables');


% Conductances next
tmppos= [0.1 0.35 0.8 0.1];
tmpa3 = axes('position',tmppos);
tmph=plot(dyn.t,dyn.gK,'k-',dyn.t,dyn.gNa,'r-',dyn.t,dyn.gL,'b-');
text(1,25,'Conductances');
set(gca,'xticklabel',[]);
ylim([0 30]);
ylabel(' Conductance (Siemens)');
legend('dyn.gk', 'dyn.gNa', 'dyn.gL');
```

```
title('Conductances graph for the three variables');


% Gating
tmppos= [0.1 0.475 0.8 0.1];
tmpa3 = axes('position',tmppos);
tmph=plot(dyn.t,dyn.n,'k-',dyn.t,dyn.m,'r-',dyn.t,dyn.h,'b-');
text(1,0.8,'Gating');
set(gca,'xticklabel',[]);
ylim([0 1]);
legend('dyn.gk', 'dyn.gNa', 'dyn.gL');
title('Gating graph for three variables');


% Voltage
tmppos= [0.1 0.6 0.8 0.35];
tmpa3 = axes('position',tmppos);
tmph=plot(dyn.t,dyn.V,'k-');
set(tmph,'linewidth',2);
text(1,130,'Membrane voltage (mV)');
set(gca,'xticklabel',[]);
ylim([-20 150]);
text(-1,pars.EK,'E_K');
text(-1,pars.ENa,'E_{Na}');
text(-1,pars.EL,'E_L');
legend('dyn.gk', 'dyn.gNa', 'dyn.gL');
title('Voltage graph for three variables');


% Return the plot handle
ploth=gcf;
end




function I = impulse_t(t)
% function I = impulse_t(t)
%
% specifies the applied time-varying current
% works if t is a single value or many values
for i=1:length(t),
    if (t(i)>2 & t(i)<4)
        I(i,1)=10;
    elseif (t(i)>10 & t(i)<10.5)
        I(i,1)=0;
    else
        I(i,1)=0;
    end
end
```

```
end



function dydt = model_hh(t,y,pars)
% Joshua Weitz - BIOL 8814 - Neuron excitation - Fall 2018
%
% function dydt = model_hh(t,y,pars)
% This simulates the HH model using parameters embedded in pars
% and requires a separate function termed impluse_t which
% has information on the time-dependent applied current
% Variables
V=y(1);
n=y(2);
m=y(3);
h=y(4);
% Impulses
I=impulse_t(t); % Specified in a function
% Dynamics
Vdot = (1/pars.C)*(I- (pars.gKbar*(n).^4 .* (V - pars.EK )) - ((pars.gNabar)*(m).^3 .* (h) .* (
ndot = (pars.alphan(V) * (1- n)) - (pars.betan(V) * (n));
mdot = (pars.alpham(V) .* (1-m)) - (pars.betam(V) .* m);
hdot = (pars.alphah(V).* (1-h)) - (pars.betah(V) .* h);
% Return the changes
dydt = [Vdot; ndot; mdot; hdot];
end
```

1 Ex.1 20 / 20

✓ **- 0 pts** *Correct*

- 3 pts 1 or 2 lines wrong

```matlab
dyn.gK=pars.gKbar*dyn.n.^4;
dyn.gNa=pars.gNabar*dyn.m.^3.*dyn.h;
dyn.gL=pars.gL*ones(length(dyn.t),1);
dyn.IK=pars.gKbar*dyn.n.^4.*(dyn.V-pars.EK);
dyn.INa=pars.gNabar*dyn.m.^3.*dyn.h.*(dyn.V-pars.ENa);
dyn.IL=pars.gL*(dyn.V-pars.EL);
```

- 10 pts half wrong

```matlab
dyn.gK=pars.gKbar*dyn.n.^4;
dyn.gNa=pars.gNabar*dyn.m.^3.*dyn.h;
dyn.gL=pars.gL*ones(length(dyn.t),1);
dyn.IK=pars.gKbar*dyn.n.^4.*(dyn.V-pars.EK);
dyn.INa=pars.gNabar*dyn.m.^3.*dyn.h.*(dyn.V-pars.ENa);
dyn.IL=pars.gL*(dyn.V-pars.EL);
```

- 15 pts Almost all wrong

```matlab
dyn.gK=pars.gKbar*dyn.n.^4;
dyn.gNa=pars.gNabar*dyn.m.^3.*dyn.h;
dyn.gL=pars.gL*ones(length(dyn.t),1);
dyn.IK=pars.gKbar*dyn.n.^4.*(dyn.V-pars.EK);
dyn.INa=pars.gNabar*dyn.m.^3.*dyn.h.*(dyn.V-pars.ENa);
dyn.IL=pars.gL*(dyn.V-pars.EL);
```

- 20 pts All wrong. Correct solution is:

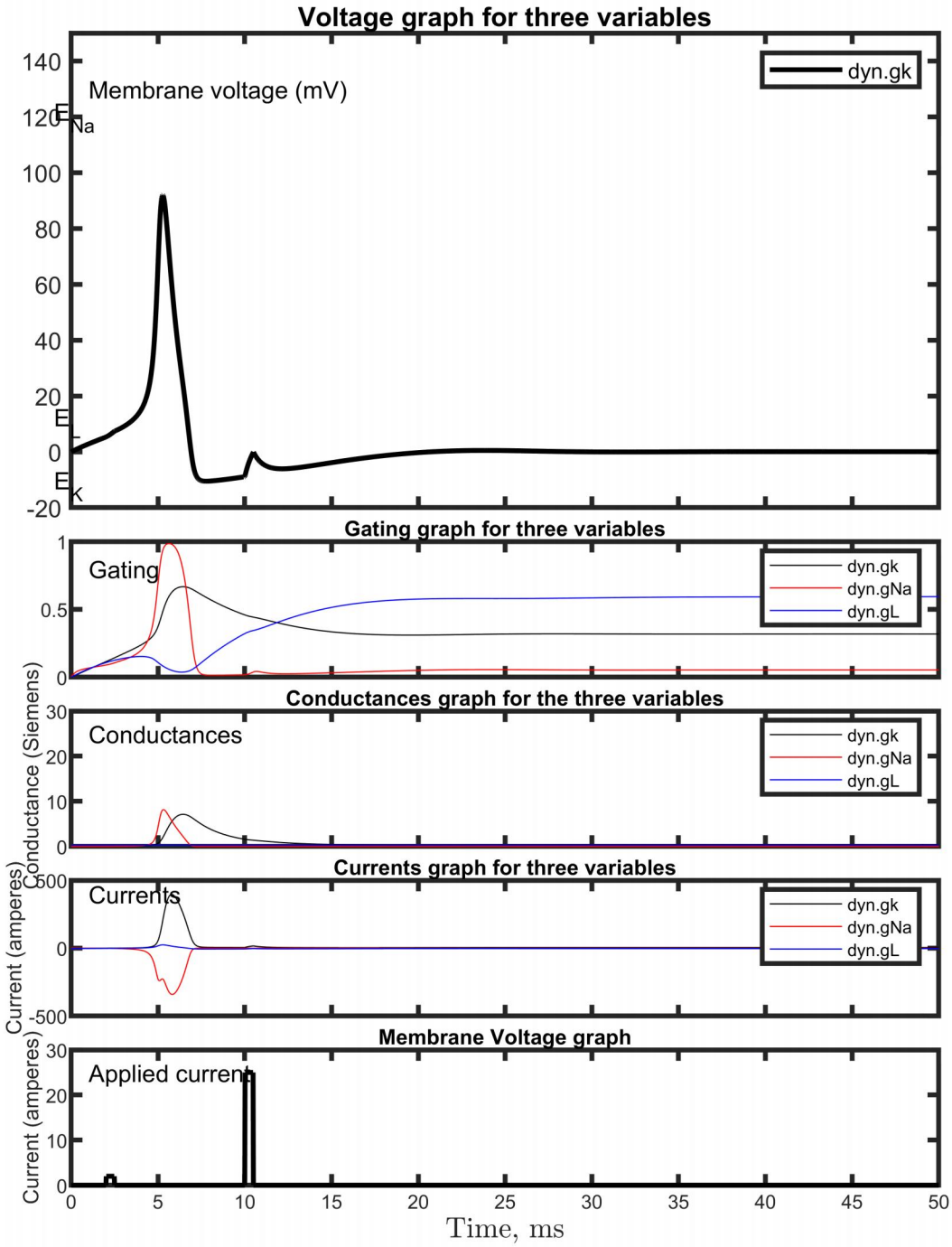
```matlab
dyn.gK=pars.gKbar*dyn.n.^4;
dyn.gNa=pars.gNabar*dyn.m.^3.*dyn.h;
dyn.gL=pars.gL*ones(length(dyn.t),1);
```

```
dyn.IK=pars.gKbar*dyn.n.^4.*(dyn.V-pars.EK);
dyn.INa=pars.gNabar*dyn.m.^3.*dyn.h.*(dyn.V-pars.ENa);
dyn.IL=pars.gL*(dyn.V-pars.EL);
```

```
%Exercise 1 - fill in the ...


% Joshua Weitz - BIOL 8814 - Neuron excitation - Fall 2018
%
% This simulates the HH model using parameters embedded in pars
% a stimulus current, and a time range over which
% the simulation should run

% Parameters, including on/off functions
pars.gKbar= 36; % mS/cm^2
pars.gNabar=120; % mS/cm^2
pars.gL=0.3; % mS/cm^2
pars.EK=-12; % mV
pars.ENa=120; % mV
pars.EL= 10.6; % mV
pars.C=1; % muF/cm^2
pars.alphan = @(V) 0.01*(10-V)./(exp(1-V/10)-1);
pars.betan = @(V) 0.125*exp(-V/80);
pars.alpham = @(V) 0.1*(25-V)./(exp(2.5-V/10)-1);
pars.betam = @(V) 4*exp(-V/18);
pars.alphah = @(V) 0.07*exp(-V/20);
pars.betah = @(V) (exp(3-V/10)+1).^-1;

% Initial conditions
pars.V0=0;
pars.n0= 0;
pars.m0 = 0;
pars.h0 = 0;

% Run the model
y0 = [pars.V0 pars.n0 pars.m0 pars.h0];
[t,y]=ode45(@model_hh,[0:0.02:50],y0,[],pars);

% Store results
% FILL IN WHEREVER YOU SEE ...
dyn.t=t;
dyn.V=y(:,1);
dyn.n=y(:,2);
dyn.m=y(:,3);
dyn.h=y(:,4);
dyn.gK=pars.gKbar*(dyn.n).^4;
dyn.gNa=(pars.gNabar)*(dyn.m).^3 .* (dyn.h);
dyn.gL=pars.gL * ones(length(dyn.t),1);
dyn.IK= pars.gKbar*(dyn.n).^4 .* (dyn.V-pars.EK);
dyn.INa= (pars.gNabar)*(dyn.m).^3 .* (dyn.h) .* (dyn.V - pars.ENa);
dyn.IL= pars.gL .* (dyn.V - pars.EL);
dyn.appliedI = impulse_t(dyn.t);
```

```
ploth = plot_hh(dyn, pars);
```

```
Warning: Ignoring extra legend entries.
```




```
% Thought problem:
% Here are my thoughts...

pars.gKbar= 36; % mS/cm^2
pars.gNabar=120; % mS/cm^2
```

```
pars.gL=0.3; % mS/cm^2
pars.EK=-12; % mV
pars.ENa=120; % mV
pars.EL= 10.6; % mV
pars.C=1; % muF/cm^2
pars.alphan = @(V) 0.01*(10-V)./(exp(1-V/10)-1);
pars.betan = @(V) 0.125*exp(-V/80);
pars.alpham = @(V) 0.1*(25-V)./(exp(2.5-V/10)-1);
pars.betam = @(V) 4*exp(-V/18);
pars.alphah = @(V) 0.07*exp(-V/20);
pars.betah = @(V) (exp(3-V/10)+1).^-1;

% Initial conditions
pars.V0=0;
pars.n0= pars.alphan(pars.V0) / (pars.alphan(pars.V0) + pars.betan(pars.V0)); %% n stars from e
pars.m0= pars.alpham(pars.V0) / (pars.alpham(pars.V0) + pars.betam(pars.V0)); %% plug in equili
pars.h0= pars.alphah(pars.V0) / (pars.alphah(pars.V0) + pars.betah(pars.V0)); % leave V(0) as i

% Run the model
y0 = [pars.V0 pars.n0 pars.m0 pars.h0];
[t,y]=ode45(@model_hh,[0:0.02:20],y0,[],pars);

% Store results
% FILL IN WHEREVER YOU SEE ...
dyn.t=t;
dyn.V=y(:,1);
dyn.n=y(:,2);
dyn.m=y(:,3);
dyn.h=y(:,4);
dyn.gK=pars.gKbar*(dyn.n).^4;
dyn.gNa=(pars.gNabar)*(dyn.m).^3 .* (dyn.h);
dyn.gL=pars.gL * ones(length(dyn.t),1);
dyn.IK= pars.gKbar*(dyn.n).^4 .* (dyn.V-pars.EK);
dyn.INa= (pars.gNabar)*(dyn.m).^3 .* (dyn.h) .* (dyn.V - pars.ENa);
dyn.IL= pars.gL .* (dyn.V - pars.EL);
dyn.appliedI = impulse_t(dyn.t);

ploth = plot_hh(dyn, pars);
```

```
Warning: Ignoring extra legend entries.
```

```
% generated. This shows how the neuron responds to an impulse (a sudden
% increase, positive feedback by the components) by culminating in an
% action potential.
```

```
function ploth = plot_hh(dyn,pars)
% function ploth = plot_hh(dyn,pars)
%
% Takes a HH output structure in dyn and plots
% dynamics of Voltage, gating variables, conductance
% currents and applied current
% Defaults
clf;
set(gcf,'DefaultLineMarkerSize',10);
set(gcf,'DefaultAxesLineWidth',2);
set(gcf,'PaperPositionMode','auto');
set(gcf,'Position',[300 40 600 750]);
% main data goes here
% Applied current on bottomA
tmppos= [0.1 0.1 0.8 0.1];
tmpa1 = axes('position',tmppos);
tmph=plot(dyn.t,dyn.appliedI,'k-');
set(tmph,'linewidth',2);
xlabel('Time, ms','fontsize',12,'verticalalignment','top','interpreter','latex');
text(1,25,'Applied current');
ylim([0 30]);
ylabel('Current (amperes)');
title('Membrane Voltage graph');

% Currents next
tmppos= [0.1 0.225 0.8 0.1];
tmpa2 = axes('position',tmppos);
tmph=plot(dyn.t,dyn.IK,'k-',dyn.t,dyn.INa,'r-',dyn.t,dyn.IL,'b-');
text(1,400,'Currents');
set(gca,'xticklabel',[]);
ylim([-500 500]);
ylabel('Current (amperes)')
legend('dyn.gk', 'dyn.gNa', 'dyn.gL');
title('Currents graph for three variables');


% Conductances next
tmppos= [0.1 0.35 0.8 0.1];
tmpa3 = axes('position',tmppos);
tmph=plot(dyn.t,dyn.gK,'k-',dyn.t,dyn.gNa,'r-',dyn.t,dyn.gL,'b-');
text(1,25,'Conductances');
set(gca,'xticklabel',[]);
ylim([0 30]);
ylabel(' Conductance (Siemens)');
legend('dyn.gk', 'dyn.gNa', 'dyn.gL');
```

```
title('Conductances graph for the three variables');


% Gating
tmppos= [0.1 0.475 0.8 0.1];
tmpa3 = axes('position',tmppos);
tmph=plot(dyn.t,dyn.n,'k-',dyn.t,dyn.m,'r-',dyn.t,dyn.h,'b-');
text(1,0.8,'Gating');
set(gca,'xticklabel',[]);
ylim([0 1]);
legend('dyn.gk', 'dyn.gNa', 'dyn.gL');
title('Gating graph for three variables');


% Voltage
tmppos= [0.1 0.6 0.8 0.35];
tmpa3 = axes('position',tmppos);
tmph=plot(dyn.t,dyn.V,'k-');
set(tmph,'linewidth',2);
text(1,130,'Membrane voltage (mV)');
set(gca,'xticklabel',[]);
ylim([-20 150]);
text(-1,pars.EK,'E_K');
text(-1,pars.ENa,'E_{Na}');
text(-1,pars.EL,'E_L');
legend('dyn.gk', 'dyn.gNa', 'dyn.gL');
title('Voltage graph for three variables');


% Return the plot handle
ploth=gcf;
end




function I = impulse_t(t)
% function I = impulse_t(t)
%
% specifies the applied time-varying current
% works if t is a single value or many values
for i=1:length(t),
    if (t(i)>2 & t(i)<4)
        I(i,1)=10;
    elseif (t(i)>10 & t(i)<10.5)
        I(i,1)=0;
    else
        I(i,1)=0;
    end
end
```

```
end



function dydt = model_hh(t,y,pars)
% Joshua Weitz - BIOL 8814 - Neuron excitation - Fall 2018
%
% function dydt = model_hh(t,y,pars)
% This simulates the HH model using parameters embedded in pars
% and requires a separate function termed impluse_t which
% has information on the time-dependent applied current
% Variables
V=y(1);
n=y(2);
m=y(3);
h=y(4);
% Impulses
I=impulse_t(t); % Specified in a function
% Dynamics
Vdot = (1/pars.C)*(I- (pars.gKbar*(n).^4 .* (V - pars.EK )) - ((pars.gNabar)*(m).^3 .* (h) .* (
ndot = (pars.alphan(V) * (1- n)) - (pars.betan(V) * (n));
mdot = (pars.alpham(V) .* (1-m)) - (pars.betam(V) .* m);
hdot = (pars.alphah(V).* (1-h)) - (pars.betah(V) .* h);
% Return the changes
dydt = [Vdot; ndot; mdot; hdot];
end
```

2 Ex.2 20 / 20

✓ **- 0 pts** *Correct*

- 15 pts Code not correct; should use `n`, `m`, `h` instead of `dyn.n`, `dyn.m`, `dyn.h` as those are not defined yet

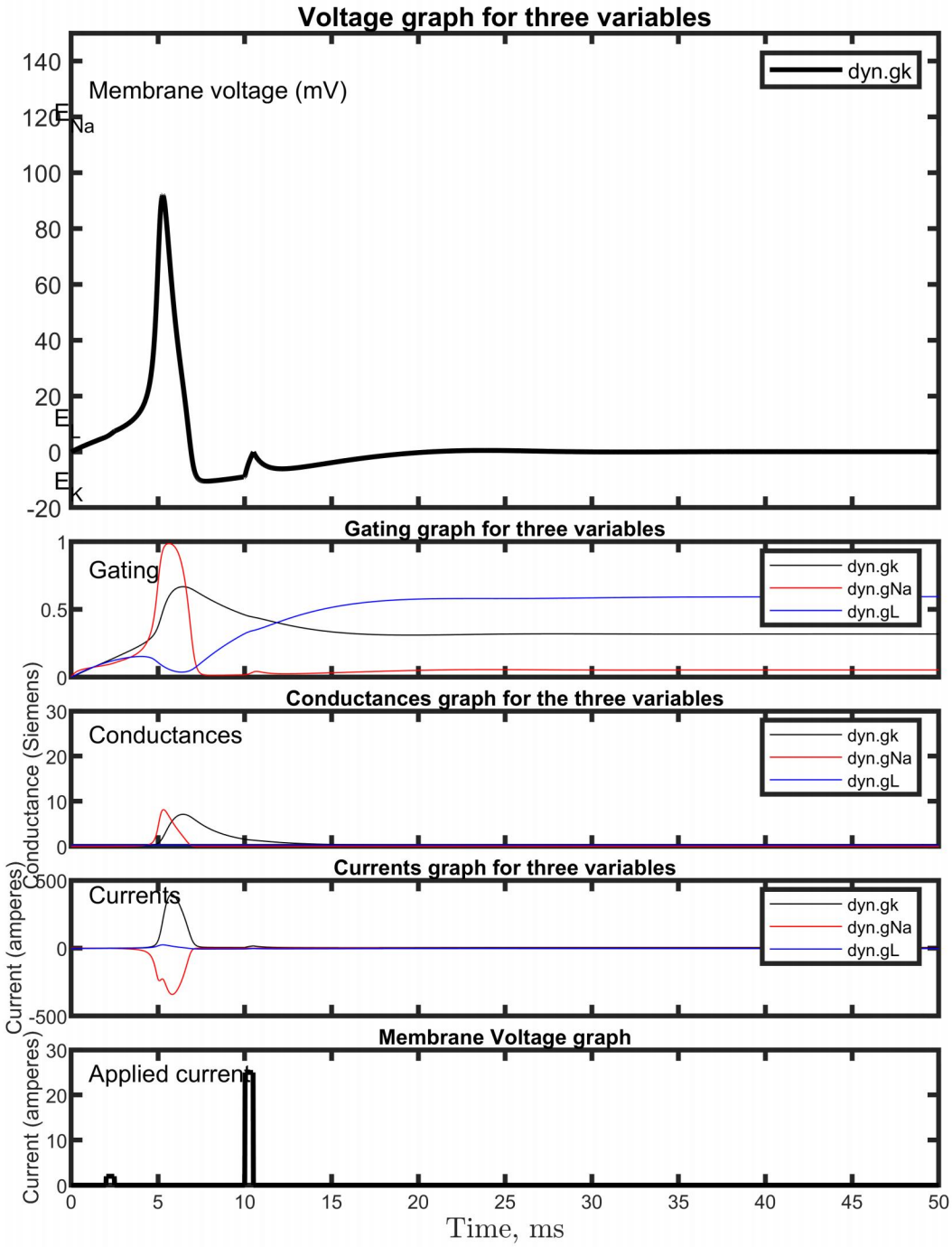
- 5 pts Code not correct

```matlab
Vdot = (1/pars.C)*(I-pars.gKbar*n^4*(V-pars.EK)-pars.gNabar*m^3*h*(V-pars.ENa) ...
-pars.gL*(V-pars.EL));
ndot = pars.alphan(V)*(1-n)-pars.betan(V)*n;
mdot = pars.alpham(V)*(1-m)-pars.betam(V)*m;
hdot = pars.alphah(V)*(1-h)-pars.betah(V)*h;
```

```
%Exercise 1 - fill in the ...


% Joshua Weitz - BIOL 8814 - Neuron excitation - Fall 2018
%
% This simulates the HH model using parameters embedded in pars
% a stimulus current, and a time range over which
% the simulation should run

% Parameters, including on/off functions
pars.gKbar= 36; % mS/cm^2
pars.gNabar=120; % mS/cm^2
pars.gL=0.3; % mS/cm^2
pars.EK=-12; % mV
pars.ENa=120; % mV
pars.EL= 10.6; % mV
pars.C=1; % muF/cm^2
pars.alphan = @(V) 0.01*(10-V)./(exp(1-V/10)-1);
pars.betan = @(V) 0.125*exp(-V/80);
pars.alpham = @(V) 0.1*(25-V)./(exp(2.5-V/10)-1);
pars.betam = @(V) 4*exp(-V/18);
pars.alphah = @(V) 0.07*exp(-V/20);
pars.betah = @(V) (exp(3-V/10)+1).^-1;

% Initial conditions
pars.V0=0;
pars.n0= 0;
pars.m0 = 0;
pars.h0 = 0;

% Run the model
y0 = [pars.V0 pars.n0 pars.m0 pars.h0];
[t,y]=ode45(@model_hh,[0:0.02:50],y0,[],pars);

% Store results
% FILL IN WHEREVER YOU SEE ...
dyn.t=t;
dyn.V=y(:,1);
dyn.n=y(:,2);
dyn.m=y(:,3);
dyn.h=y(:,4);
dyn.gK=pars.gKbar*(dyn.n).^4;
dyn.gNa=(pars.gNabar)*(dyn.m).^3 .* (dyn.h);
dyn.gL=pars.gL * ones(length(dyn.t),1);
dyn.IK= pars.gKbar*(dyn.n).^4 .* (dyn.V-pars.EK);
dyn.INa= (pars.gNabar)*(dyn.m).^3 .* (dyn.h) .* (dyn.V - pars.ENa);
dyn.IL= pars.gL .* (dyn.V - pars.EL);
dyn.appliedI = impulse_t(dyn.t);
```

```
ploth = plot_hh(dyn, pars);
```

```
Warning: Ignoring extra legend entries.
```




```
% Thought problem:
% Here are my thoughts...

pars.gKbar= 36; % mS/cm^2
pars.gNabar=120; % mS/cm^2
```

```
% generated. This shows how the neuron responds to an impulse (a sudden
% increase, positive feedback by the components) by culminating in an
% action potential.
```

```
function ploth = plot_hh(dyn,pars)
% function ploth = plot_hh(dyn,pars)
%
% Takes a HH output structure in dyn and plots
% dynamics of Voltage, gating variables, conductance
% currents and applied current
% Defaults
clf;
set(gcf,'DefaultLineMarkerSize',10);
set(gcf,'DefaultAxesLineWidth',2);
set(gcf,'PaperPositionMode','auto');
set(gcf,'Position',[300 40 600 750]);
% main data goes here
% Applied current on bottomA
tmppos= [0.1 0.1 0.8 0.1];
tmpa1 = axes('position',tmppos);
tmph=plot(dyn.t,dyn.appliedI,'k-');
set(tmph,'linewidth',2);
xlabel('Time, ms','fontsize',12,'verticalalignment','top','interpreter','latex');
text(1,25,'Applied current');
ylim([0 30]);
ylabel('Current (amperes)');
title('Membrane Voltage graph');

% Currents next
tmppos= [0.1 0.225 0.8 0.1];
tmpa2 = axes('position',tmppos);
tmph=plot(dyn.t,dyn.IK,'k-',dyn.t,dyn.INa,'r-',dyn.t,dyn.IL,'b-');
text(1,400,'Currents');
set(gca,'xticklabel',[]);
ylim([-500 500]);
ylabel('Current (amperes)')
legend('dyn.gk', 'dyn.gNa', 'dyn.gL');
title('Currents graph for three variables');


% Conductances next
tmppos= [0.1 0.35 0.8 0.1];
tmpa3 = axes('position',tmppos);
tmph=plot(dyn.t,dyn.gK,'k-',dyn.t,dyn.gNa,'r-',dyn.t,dyn.gL,'b-');
text(1,25,'Conductances');
set(gca,'xticklabel',[]);
ylim([0 30]);
ylabel(' Conductance (Siemens)');
legend('dyn.gk', 'dyn.gNa', 'dyn.gL');
```

```
title('Conductances graph for the three variables');


% Gating
tmppos= [0.1 0.475 0.8 0.1];
tmpa3 = axes('position',tmppos);
tmph=plot(dyn.t,dyn.n,'k-',dyn.t,dyn.m,'r-',dyn.t,dyn.h,'b-');
text(1,0.8,'Gating');
set(gca,'xticklabel',[]);
ylim([0 1]);
legend('dyn.gk', 'dyn.gNa', 'dyn.gL');
title('Gating graph for three variables');


% Voltage
tmppos= [0.1 0.6 0.8 0.35];
tmpa3 = axes('position',tmppos);
tmph=plot(dyn.t,dyn.V,'k-');
set(tmph,'linewidth',2);
text(1,130,'Membrane voltage (mV)');
set(gca,'xticklabel',[]);
ylim([-20 150]);
text(-1,pars.EK,'E_K');
text(-1,pars.ENa,'E_{Na}');
text(-1,pars.EL,'E_L');
legend('dyn.gk', 'dyn.gNa', 'dyn.gL');
title('Voltage graph for three variables');


% Return the plot handle
ploth=gcf;
end




function I = impulse_t(t)
% function I = impulse_t(t)
%
% specifies the applied time-varying current
% works if t is a single value or many values
for i=1:length(t),
    if (t(i)>2 & t(i)<4)
        I(i,1)=10;
    elseif (t(i)>10 & t(i)<10.5)
        I(i,1)=0;
    else
        I(i,1)=0;
    end
end
```

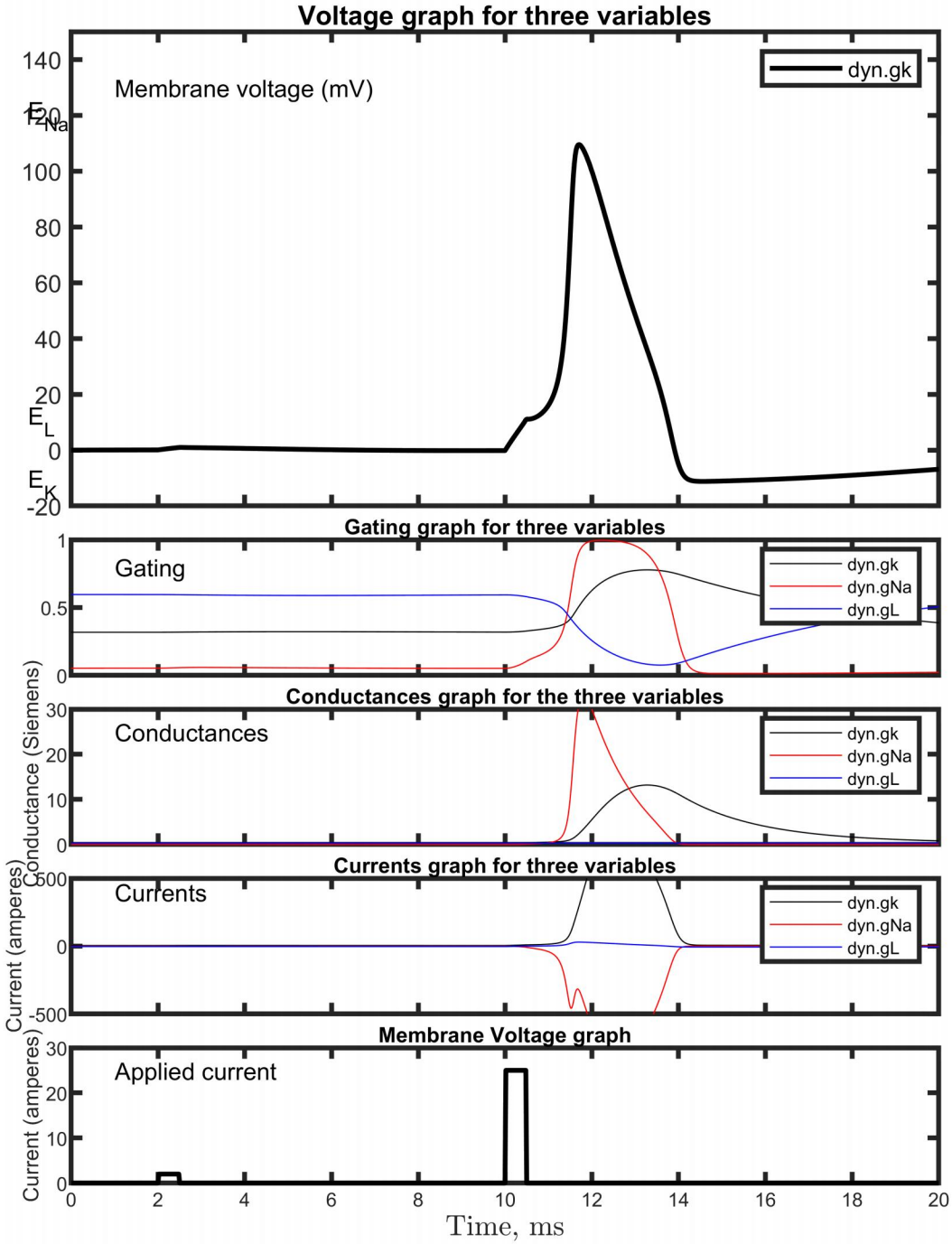
```
end



function dydt = model_hh(t,y,pars)
% Joshua Weitz - BIOL 8814 - Neuron excitation - Fall 2018
%
% function dydt = model_hh(t,y,pars)
% This simulates the HH model using parameters embedded in pars
% and requires a separate function termed impluse_t which
% has information on the time-dependent applied current
% Variables
V=y(1);
n=y(2);
m=y(3);
h=y(4);
% Impulses
I=impulse_t(t); % Specified in a function
% Dynamics
Vdot = (1/pars.C)*(I- (pars.gKbar*(n).^4 .* (V - pars.EK )) - ((pars.gNabar)*(m).^3 .* (h) .* (
ndot = (pars.alphan(V) * (1- n)) - (pars.betan(V) * (n));
mdot = (pars.alpham(V) .* (1-m)) - (pars.betam(V) .* m);
hdot = (pars.alphah(V).* (1-h)) - (pars.betah(V) .* h);
% Return the changes
dydt = [Vdot; ndot; mdot; hdot];
end
```

3 Ex.3 **20 / 20**

✓ **- 0 pts** *Correct*

- 10 pts Incorrect code

```matlab
for i=1:length(t),
if (t(i)>2 & t(i)<2.5)
I(i,1)=2;
elseif (t(i)>10 & t(i)<10.5)
I(i,1)=25;
else
I(i,1)=0;
end
end
```




```
%% exercise 4
% identify equilibrium voltage and gating variables for HH equations.

% equilibrium voltages are the voltage of the neuron at resting state, no
% impulse inputted or have recently regressed from the active state. The
% equilibrium is just V* = V0 = 0 mV.
% The gating variables' equilibrium states are the N*, M*, and H* ODEs.
```

```
% They represent corresponding steady states which are 0.6, 0.35, and 0.05
% for gL, gk, and gNa respectively. This is obtained by looking at the
% graphs corresponding with the variables, as t approaches to infinity.
```

```
% Exercise 5
%: With this result in hand, explore the effect of additional impulses of different durations
% and magnitudes while keeping the product of the current times the time the same. Are all "pul
% Can a long weak pulse generate a signal? Can a long but very strong pulse generate a signal?
% support your argument with plots)
```

```
% Exercise 5's first graph

pars.gKbar= 36; % mS/cm^2
pars.gNabar=120; % mS/cm^2
pars.gL=0.3; % mS/cm^2
pars.EK=-12; % mV
pars.ENa=120; % mV
pars.EL= 10.6; % mV
pars.C=1; % muF/cm^2
pars.alphan = @(V) 0.01*(10-V)./(exp(1-V/10)-1);
pars.betan = @(V) 0.125*exp(-V/80);
pars.alpham = @(V) 0.1*(25-V)./(exp(2.5-V/10)-1);
pars.betam = @(V) 4*exp(-V/18);
pars.alphah = @(V) 0.07*exp(-V/20);
pars.betah = @(V) (exp(3-V/10)+1).^-1;

% Initial conditions
pars.V0=0;
pars.n0= pars.alphan(pars.V0) / (pars.alphan(pars.V0) + pars.betan(pars.V0)); %% n stars from e
pars.m0= pars.alpham(pars.V0) / (pars.alpham(pars.V0) + pars.betam(pars.V0)); %% plug in equili
pars.h0= pars.alphah(pars.V0) / (pars.alphah(pars.V0) + pars.betah(pars.V0)); % leave V(0) as i

% Run the model
y0 = [pars.V0 pars.n0 pars.m0 pars.h0];
[t,y]=ode45(@model_hh,[0:0.02:20],y0,[],pars);

% Store results
% FILL IN WHEREVER YOU SEE ...
dyn.t=t;
dyn.V=y(:,1);
dyn.n=y(:,2);
dyn.m=y(:,3);
dyn.h=y(:,4);
dyn.gK=pars.gKbar*(dyn.n).^4;
dyn.gNa=(pars.gNabar)*(dyn.m).^3 .* (dyn.h);
dyn.gL=pars.gL * ones(length(dyn.t),1);
dyn.IK= pars.gKbar*(dyn.n).^4 .* (dyn.V-pars.EK);
dyn.INa= (pars.gNabar)*(dyn.m).^3 .* (dyn.h) .* (dyn.V - pars.ENa);
```

4 Ex.4 **16 / 20**

- **0 pts** Correct
- **4 pts** Correct method
- **4 pts** Correct values

✓ **- 4 pts** *Correct plot*

- **20 pts** Incorrect

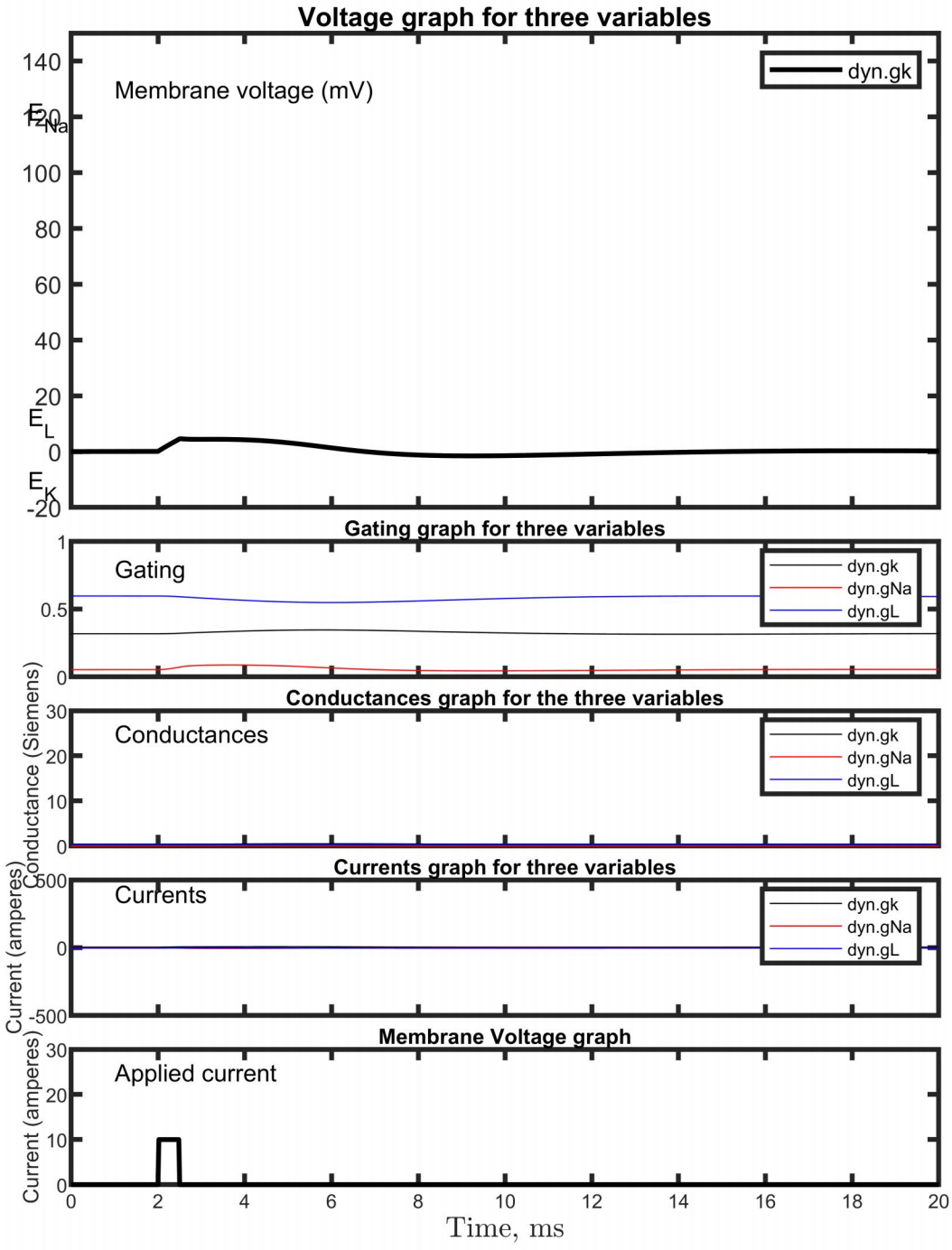
```
% They represent corresponding steady states which are 0.6, 0.35, and 0.05
% for gL, gk, and gNa respectively. This is obtained by looking at the
% graphs corresponding with the variables, as t approaches to infinity.
```

```
% Exercise 5
%: With this result in hand, explore the effect of additional impulses of different durations
% and magnitudes while keeping the product of the current times the time the same. Are all "pul
% Can a long weak pulse generate a signal? Can a long but very strong pulse generate a signal?
% support your argument with plots)
```

```
% Exercise 5's first graph

pars.gKbar= 36; % mS/cm^2
pars.gNabar=120; % mS/cm^2
pars.gL=0.3; % mS/cm^2
pars.EK=-12; % mV
pars.ENa=120; % mV
pars.EL= 10.6; % mV
pars.C=1; % muF/cm^2
pars.alphan = @(V) 0.01*(10-V)./(exp(1-V/10)-1);
pars.betan = @(V) 0.125*exp(-V/80);
pars.alpham = @(V) 0.1*(25-V)./(exp(2.5-V/10)-1);
pars.betam = @(V) 4*exp(-V/18);
pars.alphah = @(V) 0.07*exp(-V/20);
pars.betah = @(V) (exp(3-V/10)+1).^-1;

% Initial conditions
pars.V0=0;
pars.n0= pars.alphan(pars.V0) / (pars.alphan(pars.V0) + pars.betan(pars.V0)); %% n stars from e
pars.m0= pars.alpham(pars.V0) / (pars.alpham(pars.V0) + pars.betam(pars.V0)); %% plug in equili
pars.h0= pars.alphah(pars.V0) / (pars.alphah(pars.V0) + pars.betah(pars.V0)); % leave V(0) as i

% Run the model
y0 = [pars.V0 pars.n0 pars.m0 pars.h0];
[t,y]=ode45(@model_hh,[0:0.02:20],y0,[],pars);

% Store results
% FILL IN WHEREVER YOU SEE ...
dyn.t=t;
dyn.V=y(:,1);
dyn.n=y(:,2);
dyn.m=y(:,3);
dyn.h=y(:,4);
dyn.gK=pars.gKbar*(dyn.n).^4;
dyn.gNa=(pars.gNabar)*(dyn.m).^3 .* (dyn.h);
dyn.gL=pars.gL * ones(length(dyn.t),1);
dyn.IK= pars.gKbar*(dyn.n).^4 .* (dyn.V-pars.EK);
dyn.INa= (pars.gNabar)*(dyn.m).^3 .* (dyn.h) .* (dyn.V - pars.ENa);
```

```
dyn.IL= pars.gL .* (dyn.V - pars.EL);
dyn.appliedI = impulse_t(dyn.t);

ploth = plot_hh(dyn, pars);
```

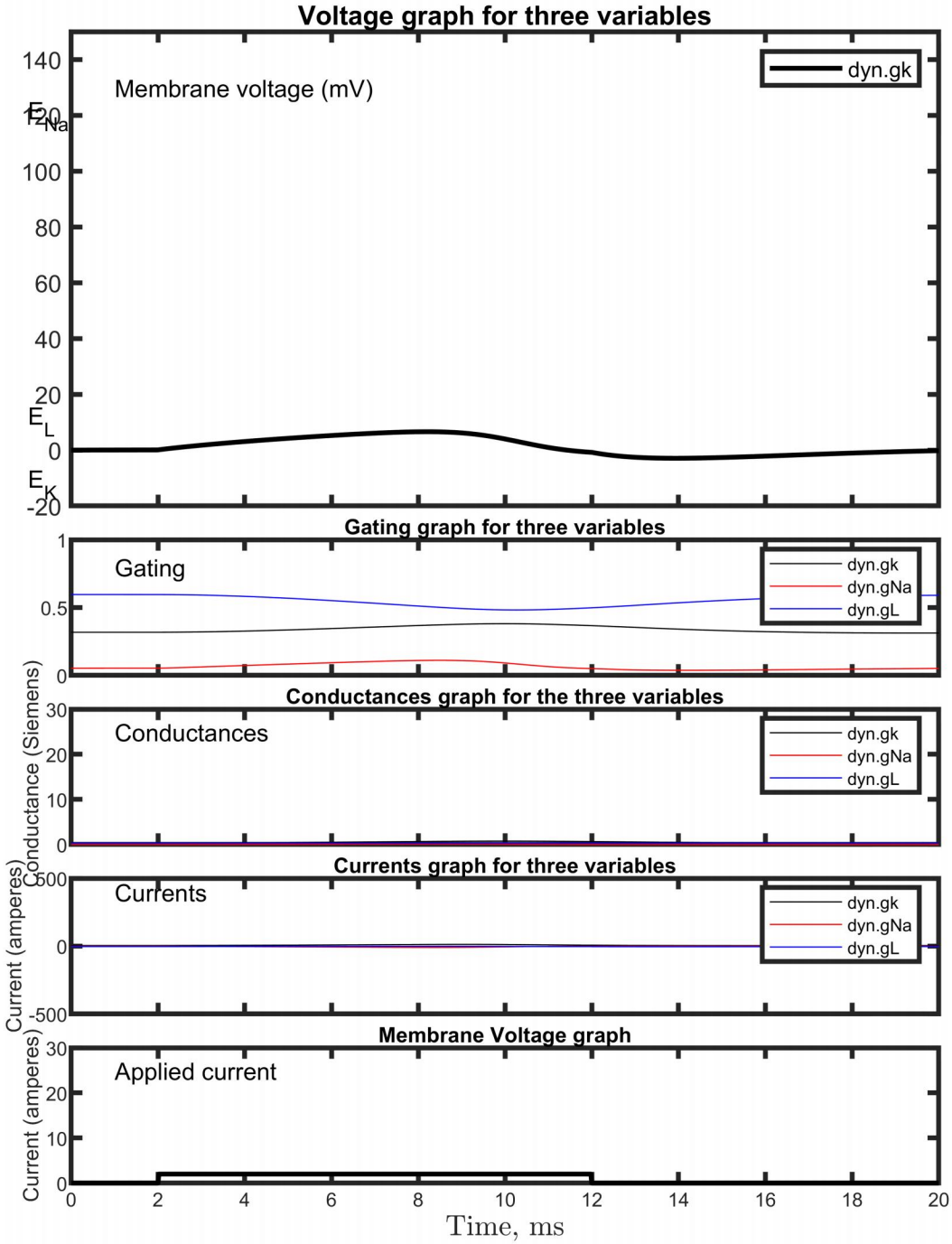
```
Warning: Ignoring extra legend entries.
```




```
% Exercise 5's second graph --- A weak long pulse
pars.gKbar= 36; % mS/cm^2
```

```
pars.gNabar=120; % mS/cm^2
pars.gL=0.3; % mS/cm^2
pars.EK=-12; % mV
pars.ENa=120; % mV
pars.EL= 10.6; % mV
pars.C=1; % muF/cm^2
pars.alphan = @(V) 0.01*(10-V)./(exp(1-V/10)-1);
pars.betan = @(V) 0.125*exp(-V/80);
pars.alpham = @(V) 0.1*(25-V)./(exp(2.5-V/10)-1);
pars.betam = @(V) 4*exp(-V/18);
pars.alphah = @(V) 0.07*exp(-V/20);
pars.betah = @(V) (exp(3-V/10)+1).^-1;

% Initial conditions
pars.V0=0;
pars.n0= pars.alphan(pars.V0) / (pars.alphan(pars.V0) + pars.betan(pars.V0)); %% n stars from e
pars.m0= pars.alpham(pars.V0) / (pars.alpham(pars.V0) + pars.betam(pars.V0)); %% plug in equili
pars.h0= pars.alphah(pars.V0) / (pars.alphah(pars.V0) + pars.betah(pars.V0)); % leave V(0) as i

% Run the model
y0 = [pars.V0 pars.n0 pars.m0 pars.h0];
[t,y]=ode45(@model_hh,[0:0.02:20],y0,[],pars);

% Store results
% FILL IN WHEREVER YOU SEE ...
dyn.t=t;
dyn.V=y(:,1);
dyn.n=y(:,2);
dyn.m=y(:,3);
dyn.h=y(:,4);
dyn.gK=pars.gKbar*(dyn.n).^4;
dyn.gNa=(pars.gNabar)*(dyn.m).^3 .* (dyn.h);
dyn.gL=pars.gL * ones(length(dyn.t),1);
dyn.IK= pars.gKbar*(dyn.n).^4 .* (dyn.V-pars.EK);
dyn.INa= (pars.gNabar)*(dyn.m).^3 .* (dyn.h) .* (dyn.V - pars.ENa);
dyn.IL= pars.gL .* (dyn.V - pars.EL);
dyn.appliedI = impulse_t(dyn.t);

ploth = plot_hh(dyn, pars);
```

```
Warning: Ignoring extra legend entries.
```

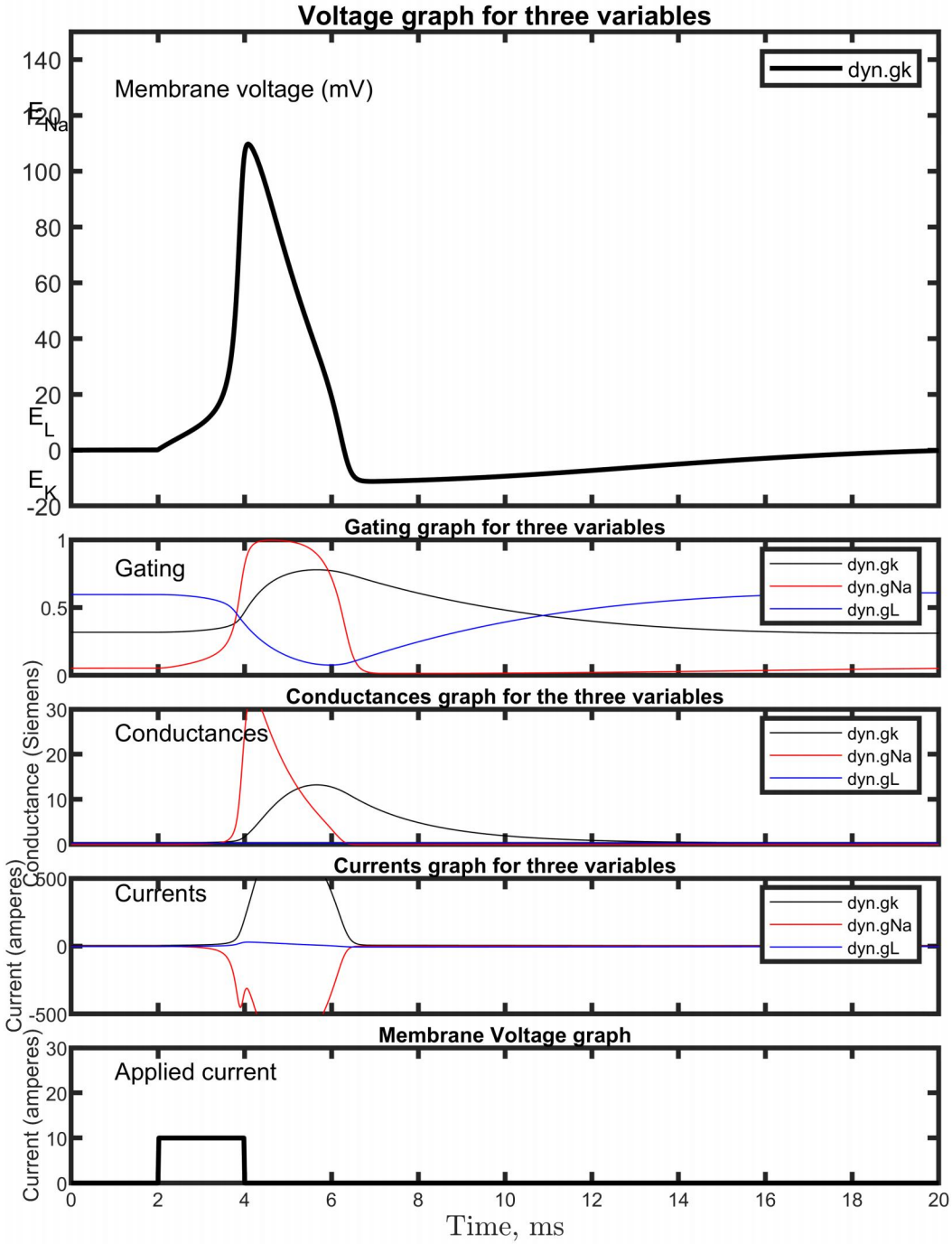



```
% where the long weak pulse is 2 but the duration is 10
```

```
% Exercise 5's third graph --- a short strong pulse
pars.gKbar= 36; % mS/cm^2
pars.gNabar=120; % mS/cm^2
pars.gL=0.3; % mS/cm^2
```

```
pars.EK=-12; % mV
pars.ENa=120; % mV
pars.EL= 10.6; % mV
pars.C=1; % muF/cm^2
pars.alphan = @(V) 0.01*(10-V)./(exp(1-V/10)-1);
pars.betan = @(V) 0.125*exp(-V/80);
pars.alpham = @(V) 0.1*(25-V)./(exp(2.5-V/10)-1);
pars.betam = @(V) 4*exp(-V/18);
pars.alphah = @(V) 0.07*exp(-V/20);
pars.betah = @(V) (exp(3-V/10)+1).^-1;

% Initial conditions
pars.V0=0;
pars.n0= pars.alphan(pars.V0) / (pars.alphan(pars.V0) + pars.betan(pars.V0)); %% n stars from e
pars.m0= pars.alpham(pars.V0) / (pars.alpham(pars.V0) + pars.betam(pars.V0)); %% plug in equili
pars.h0= pars.alphah(pars.V0) / (pars.alphah(pars.V0) + pars.betah(pars.V0)); % leave V(0) as i

% Run the model
y0 = [pars.V0 pars.n0 pars.m0 pars.h0];
[t,y]=ode45(@model_hh,[0:0.02:20],y0,[],pars);

% Store results
% FILL IN WHEREVER YOU SEE ...
dyn.t=t;
dyn.V=y(:,1);
dyn.n=y(:,2);
dyn.m=y(:,3);
dyn.h=y(:,4);
dyn.gK=pars.gKbar*(dyn.n).^4;
dyn.gNa=(pars.gNabar)*(dyn.m).^3 .* (dyn.h);
dyn.gL=pars.gL * ones(length(dyn.t),1);
dyn.IK= pars.gKbar*(dyn.n).^4 .* (dyn.V-pars.EK);
dyn.INa= (pars.gNabar)*(dyn.m).^3 .* (dyn.h) .* (dyn.V - pars.ENa);
dyn.IL= pars.gL .* (dyn.V - pars.EL);
dyn.appliedI = impulse_t(dyn.t);

ploth = plot_hh(dyn, pars);
```

```
Warning: Ignoring extra legend entries.
```




```
% where the long strong impulse is 10 but duration is 2
  The Pulses are equal in the sense that the area under the curve is the same, so the
  amperes*time in the applied current is the same. However, they are not the same
  because the action potential's preferential nature for short strong pulses.
% We can see that by keeping the area under the curve the same (current
% times time) ~20, we can simulate a long weak pulse and also
% a short strong pulse. In the long weak pulse, the action potential is not
% generated. However, in the short strong pulse, the action potential is
```

```
% generated. This shows how the neuron responds to an impulse (a sudden
% increase, positive feedback by the components) by culminating in an
% action potential.
```

```
function ploth = plot_hh(dyn,pars)
% function ploth = plot_hh(dyn,pars)
%
% Takes a HH output structure in dyn and plots
% dynamics of Voltage, gating variables, conductance
% currents and applied current
% Defaults
clf;
set(gcf,'DefaultLineMarkerSize',10);
set(gcf,'DefaultAxesLineWidth',2);
set(gcf,'PaperPositionMode','auto');
set(gcf,'Position',[300 40 600 750]);
% main data goes here
% Applied current on bottomA
tmppos= [0.1 0.1 0.8 0.1];
tmpa1 = axes('position',tmppos);
tmph=plot(dyn.t,dyn.appliedI,'k-');
set(tmph,'linewidth',2);
xlabel('Time, ms','fontsize',12,'verticalalignment','top','interpreter','latex');
text(1,25,'Applied current');
ylim([0 30]);
ylabel('Current (amperes)');
title('Membrane Voltage graph');

% Currents next
tmppos= [0.1 0.225 0.8 0.1];
tmpa2 = axes('position',tmppos);
tmph=plot(dyn.t,dyn.IK,'k-',dyn.t,dyn.INa,'r-',dyn.t,dyn.IL,'b-');
text(1,400,'Currents');
set(gca,'xticklabel',[]);
ylim([-500 500]);
ylabel('Current (amperes)')
legend('dyn.gk', 'dyn.gNa', 'dyn.gL');
title('Currents graph for three variables');


% Conductances next
tmppos= [0.1 0.35 0.8 0.1];
tmpa3 = axes('position',tmppos);
tmph=plot(dyn.t,dyn.gK,'k-',dyn.t,dyn.gNa,'r-',dyn.t,dyn.gL,'b-');
text(1,25,'Conductances');
set(gca,'xticklabel',[]);
ylim([0 30]);
ylabel(' Conductance (Siemens)');
legend('dyn.gk', 'dyn.gNa', 'dyn.gL');
```

5 Ex.5 **20 / 20**

✓ **- 0 pts** *Correct*

- 5 pts Correctly set up input impulses: one long-weak and one short-strong

- 4 pts Only one correct plot

- 8 pts No correct plots

- 7 pts Conclusion: in general, short-strong should generate a signal while long-weak should not

- 1 pts Long-weak impulse generates signal (it should not)