```
% Tutorial 4.3
% Email: jlindbloom@smu.edu

clear all
close all

% Define parameters.
global A_s A_d g_s_leak g_d_leak
global g_max_na g_max_k g_max_ca g_max_kca g_max_kahp g_link
global e_na e_ca e_k e_leak C_s C_d I_s_app I_d_app tau_ca k
A_s = 1/3;
A_d = 2/3;
g_s_leak = A_s*5e-9;
g_d_leak = A_d*5e-9;
g_max_na = A_s*3e-6;
g_max_k = A_s*2e-6;
g_max_ca = A_d*2e-6;
g_max_kca = A_d*2.5e-6;
g_max_kahp = A_d*40e-9;
g_link = 50e-9;
e_na = 60e-3;
e_ca = 80e-3;
e_k = -75e-3;
e_leak = -60e-3;
C_s = A_s*100e-12;
C_d = A_d*100e-12;
I_s_app = 0;
I_d_app = 0;
tau_ca = 50e-3;
k = (5*(10^6))/A_d;



% Create vectors for membrane potential and calcium concentration.
v = [-0.085:0.005:0.05];
ca = [0:0.1e-3:2e-3];
[alpha_m, beta_m, alpha_h, beta_h , alpha_n, beta_n] =
 PR_soma_gating(v);
[alpha_mca, beta_mca, alpha_kca, beta_kca, alpha_kahp, beta_kahp ] =
 PR_dend_gating(v, ca);

% Plot for somatic gating variables.
f1 = figure;
figure(f1);
subplot(1,3,1);
plot(v, alpha_m);
hold on
plot(v, beta_m);
legend("\alpha_m", "\beta_m");
xlabel("Membrane Potential (Volts)");

subplot(1,3,2);
```

```
plot(v, alpha_h);
hold on
plot(v, beta_h);
legend("\alpha_h", "\beta_h");
xlabel("Membrane Potential (Volts)");

subplot(1,3,3);
plot(v, alpha_n);
hold on
plot(v, beta_n);
legend("\alpha_n", "\beta_n");
xlabel("Membrane Potential (Volts)");

a = axes;
t1 = title('Somatic Rate Constants');
a.Visible = 'off';
t1.Visible = 'on';

% Plot for dendritic gating variables.
f2 = figure;
figure(f2);
subplot(1,3,1);
plot(v, alpha_mca);
hold on
plot(v, beta_mca);
legend("\alpha_{mca}", "\beta_{mca}");
xlabel("Membrane Potential (Volts)");

subplot(1,3,2);
plot(v, alpha_kca);
hold on
plot(v, beta_kca);
legend("\alpha_{kca}", "\beta_{kca}");
xlabel("Membrane Potential (Volts)");

subplot(1,3,3);
plot(ca, alpha_kahp);
hold on
plot(ca, beta_kahp);
legend("\alpha_{kahp}", "\beta_{kahp}");
xlabel("Calcium Concentration");

a = axes;
t1 = title('Dendritic Rate Constants');
a.Visible = 'off';
t1.Visible = 'on';


% Setup time vector.
global dt
dt = 2e-6;
t = 0:dt:2;

% Run simulation.
```

```
[v_s_sim, v_d_sim, m_sim, h_sim, n_sim, m_ca_sim, m_kca_sim, ...
    x_sim, m_kahp_sim, ca_conc, spikes] = prsim(t);

f3 = figure;
figure(f3);
subplot(2,1,1);
plot(t, v_s_sim)
ylabel("Somatic Membrane Potential (Volts)");
xlabel("Time (seconds)");
title(sprintf("Part 3 and 4. # of Somatic Spikes = %d", sum(spikes)));
subplot(2,1,2);
plot(t, v_d_sim)
ylabel("Dendritic Membrane Potential (Volts)");
xlabel("Time (seconds)");


% Run simulations for part 5.
g_link = 0e-9;
[v_s_sim, v_d_sim, m_sim, h_sim, n_sim, m_ca_sim, m_kca_sim, ...
    x_sim, m_kahp_sim, ca_conc, spikes] = prsim(t);

f4 = figure;
figure(f4);
subplot(4,1,1);
plot(t, v_s_sim)
ylabel("Somatic Potential (Volts)");
xlabel("Time (seconds)");
title(sprintf("Part 5. G_{link} = 0 nS. # of Somatic Spikes = %d",
 sum(spikes)));
subplot(4,1,2);
plot(t, v_d_sim)
ylabel("Dendritic Potential (Volts)");
xlabel("Time (seconds)");
subplot(4,1,3);
plot(t, m_sim)
hold on
plot(t, h_sim)
hold on
plot(t, n_sim)
ylabel("Gating Variables");
xlabel("Time (seconds)");
subplot(4,1,4);
plot(t, g_link*(v_d_sim-v_s_sim))
xlabel("Time (seconds)")
ylabel("I_{Link}")

g_link = 10e-9;
[v_s_sim, v_d_sim, m_sim, h_sim, n_sim, m_ca_sim, m_kca_sim, ...
    x_sim, m_kahp_sim, ca_conc, spikes] = prsim(t);

f5 = figure;
figure(f5);
subplot(4,1,1);
```

```
plot(t, v_s_sim)
ylabel("Somatic Potential (Volts)");
xlabel("Time (seconds)");
title(sprintf("Part 5. G_{link} = 10 nS. # of Somatic Spikes = %d",
 sum(spikes)));
subplot(4,1,2);
plot(t, v_d_sim)
ylabel("Dendritic Potential (Volts)");
xlabel("Time (seconds)");
subplot(4,1,3);
plot(t, m_sim)
hold on
plot(t, h_sim)
hold on
plot(t, n_sim)
ylabel("Gating Variables");
xlabel("Time (seconds)");
subplot(4,1,4);
plot(t, g_link*(v_d_sim-v_s_sim))
xlabel("Time (seconds)")
ylabel("I_{Link}")

g_link = 100e-9;
[v_s_sim, v_d_sim, m_sim, h_sim, n_sim, m_ca_sim, m_kca_sim, ...
    x_sim, m_kahp_sim, ca_conc, spikes] = prsim(t);

f6 = figure;
figure(f6);
subplot(4,1,1);
plot(t, v_s_sim)
ylabel("Somatic Potential (Volts)");
xlabel("Time (seconds)");
title(sprintf("Part 5. G_{link} = 100 nS. # of Somatic Spikes = %d",
 sum(spikes)));
subplot(4,1,2);
plot(t, v_d_sim)
ylabel("Dendritic Potential (Volts)");
xlabel("Time (seconds)");
subplot(4,1,3);
plot(t, m_kahp_sim)
ylabel("m_{KAHP}");
xlabel("Time (seconds)");
subplot(4,1,4);
plot(t, g_link*(v_d_sim-v_s_sim))
xlabel("Time (seconds)")
ylabel("I_{Link}")



% Apply current to the neuron.
I_app1 = zeros(1, length(t)) + 50e-12;
I_app2 = zeros(1, length(t)) + 100e-12;
I_app3 = zeros(1, length(t)) + 200e-12;
I_app_none = zeros(1, length(t));
```

```
% Apply to the dendrite.
g_link = 50e-9;

[v_s_sim, v_d_sim, m_sim, h_sim, n_sim, m_ca_sim, m_kca_sim, ...
    x_sim, m_kahp_sim, ca_conc, spikes] = prsim(t, I_app_none,
 I_app1);

f8 = figure;
figure(f8);
subplot(2,1,1);
plot(t, v_s_sim)
ylabel("Somatic Membrane Potential (Volts)");
xlabel("Time (seconds)");
title(sprintf("Part 6. Dendrite, I_app = 50e-12. # of Somatic Spikes =
 %d", sum(spikes)));
subplot(2,1,2);
plot(t, v_d_sim)
ylabel("Dendritic Membrane Potential (Volts)");
xlabel("Time (seconds)");

[v_s_sim, v_d_sim, m_sim, h_sim, n_sim, m_ca_sim, m_kca_sim, ...
    x_sim, m_kahp_sim, ca_conc, spikes] = prsim(t, I_app_none,
 I_app2);

f9 = figure;
figure(f9);
subplot(2,1,1);
plot(t, v_s_sim)
ylabel("Somatic Membrane Potential (Volts)");
xlabel("Time (seconds)");
title(sprintf("Part 6. Dendrite, I_app = 100e-12. # of Somatic Spikes
 = %d", sum(spikes)));
subplot(2,1,2);
plot(t, v_d_sim)
ylabel("Dendritic Membrane Potential (Volts)");
xlabel("Time (seconds)");

[v_s_sim, v_d_sim, m_sim, h_sim, n_sim, m_ca_sim, m_kca_sim, ...
    x_sim, m_kahp_sim, ca_conc, spikes] =  prsim(t, I_app_none,
 I_app3);

f10 = figure;
figure(f10);
subplot(2,1,1);
plot(t, v_s_sim)
ylabel("Somatic Membrane Potential (Volts)");
xlabel("Time (seconds)");
title(sprintf("Part 6. Dendrite, I_app = 200e-12. # of Somatic Spikes
 = %d", sum(spikes)));
subplot(2,1,2);
plot(t, v_d_sim)
ylabel("Dendritic Membrane Potential (Volts)");
```

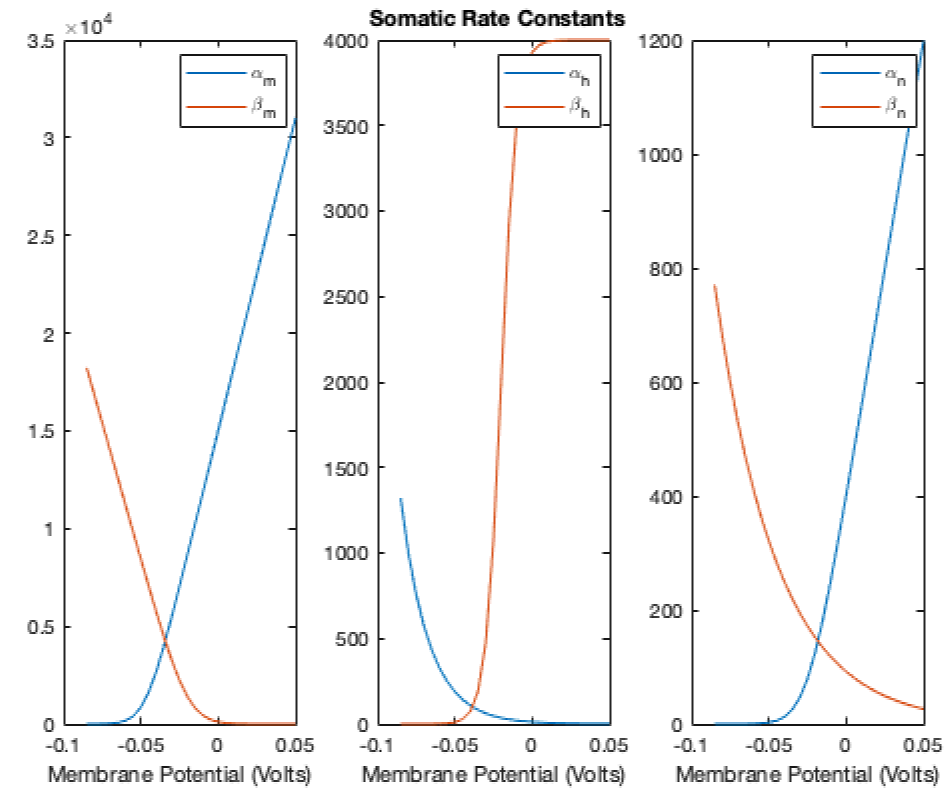
```
xlabel("Time (seconds)");


% Apply to the soma.
[v_s_sim, v_d_sim, m_sim, h_sim, n_sim, m_ca_sim, m_kca_sim, ...
    x_sim, m_kahp_sim, ca_conc, spikes] = prsim(t, I_app1,
 I_app_none);

f11 = figure;
figure(f11);
subplot(2,1,1);
plot(t, v_s_sim)
ylabel("Somatic Membrane Potential (Volts)");
xlabel("Time (seconds)");
title(sprintf("Part 6. Soma, I_app = 50e-12. # of Somatic Spikes =
 %d", sum(spikes)));
subplot(2,1,2);
plot(t, v_d_sim)
ylabel("Dendritic Membrane Potential (Volts)");
xlabel("Time (seconds)");

[v_s_sim, v_d_sim, m_sim, h_sim, n_sim, m_ca_sim, m_kca_sim, ...
    x_sim, m_kahp_sim, ca_conc, spikes] = prsim(t, I_app2,
 I_app_none);

f12 = figure;
figure(f12);
subplot(2,1,1);
plot(t, v_s_sim)
ylabel("Somatic Membrane Potential (Volts)");
xlabel("Time (seconds)");
title(sprintf("Part 6. Soma, I_app = 100e-12. # of Somatic Spikes =
 %d", sum(spikes)));
subplot(2,1,2);
plot(t, v_d_sim)
ylabel("Dendritic Membrane Potential (Volts)");
xlabel("Time (seconds)");

[v_s_sim, v_d_sim, m_sim, h_sim, n_sim, m_ca_sim, m_kca_sim, ...
    x_sim, m_kahp_sim, ca_conc, spikes] =  prsim(t, I_app3,
 I_app_none);

f13 = figure;
figure(f13);
subplot(2,1,1);
plot(t, v_s_sim)
ylabel("Somatic Membrane Potential (Volts)");
xlabel("Time (seconds)");
title(sprintf("Part 6. Soma, I_app = 200e-12. # of Somatic Spikes =
 %d", sum(spikes)));
subplot(2,1,2);
plot(t, v_d_sim)
ylabel("Dendritic Membrane Potential (Volts)");
xlabel("Time (seconds)");
```

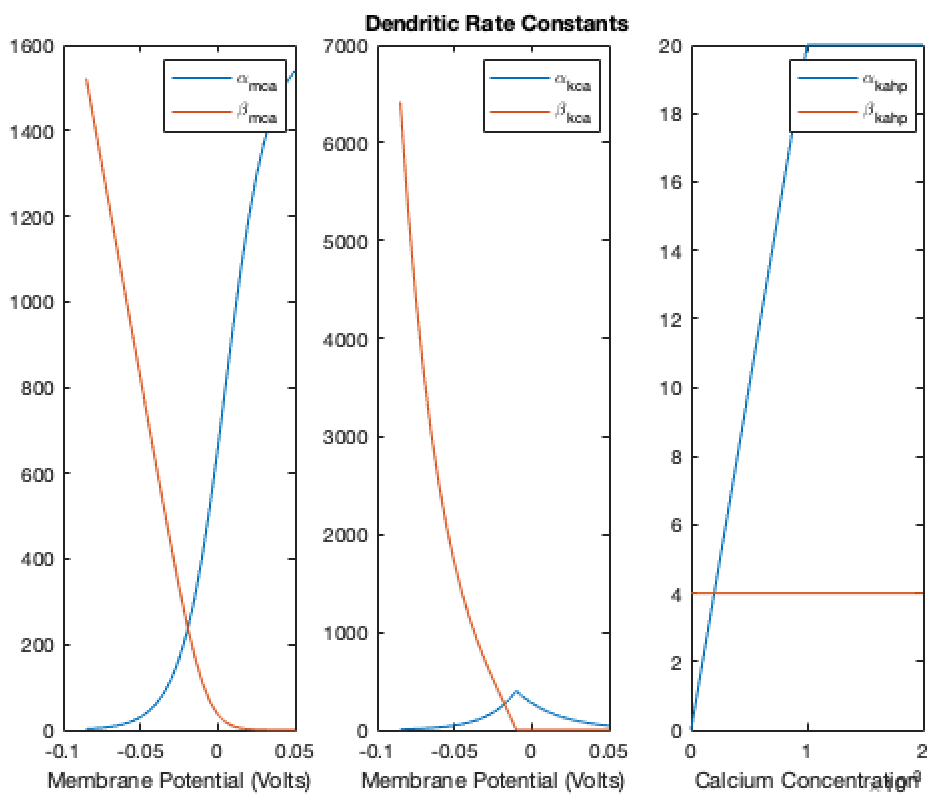
```
%%%%%%%%%%%%%%%%%%%%%%%
% Function Definitions:

function [v_s_sim, v_d_sim, m_sim, h_sim, n_sim, m_ca_sim,
 m_kca_sim, ...
    x_sim, m_kahp_sim, ca_conc, spikes] = prsim(t, i_s_applied, ...
    i_d_applied, v_s_init, v_d_init, m_init, h_init, n_init,
 m_ca_init, ...
    m_kca_init, x_init, m_kahp_init, ca_conc_init)
% Simulates the thalamocortical neuron model with a T-type calcium
 current given the input time vector and
% applied current.

global A_s A_d g_s_leak g_d_leak dt
global g_max_na g_max_k g_max_ca g_max_kca g_max_kahp g_link
global e_na e_ca e_k e_leak C_s C_d I_s_app I_d_app tau_ca k

% Default parameters if not inputted.
if (~exist('v_s_init'))
    v_s_init = 0;
end
if (~exist('v_d_init'))
    v_d_init = 0;
end
if (~exist('m_init'))
    m_init = 0;
end
if (~exist('h_init'))
    h_init = 0;
end
if (~exist('n_init'))
    n_init = 0;
end
if (~exist('m_ca_init'))
    m_ca_init = 0;
end
if (~exist('m_kca_init'))
    m_kca_init = 0;
end
if (~exist('x_init'))
    x_init = 0;
end
if (~exist('m_kahp_init'))
    m_kahp_init = 0;
end
if (~exist('i_s_applied'))
    i_s_applied = zeros(1, length(t));
end
if (~exist('i_d_applied'))
    i_d_applied = zeros(1, length(t));
```

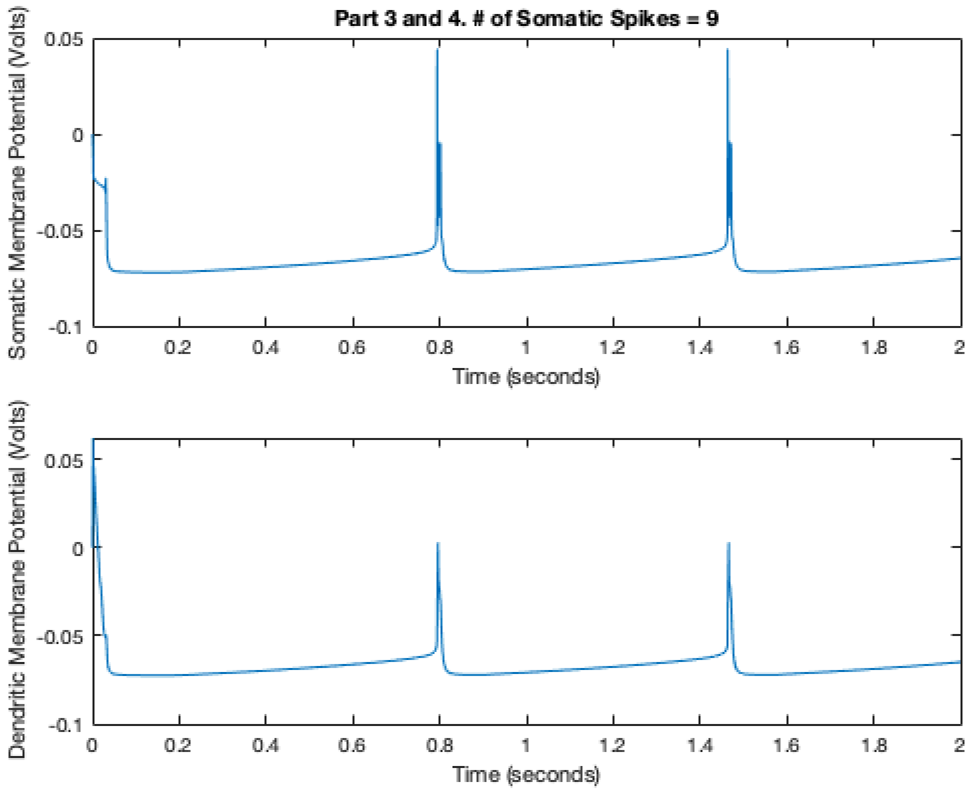
```
end
%{
if (~exist('i_ca_init'))
    i_ca_init = 0;
end
%}
if (~exist('ca_conc_init'))
    ca_conc_init = 0;
end


% Setup vectors.
v_s_sim = zeros(1, length(t));
v_d_sim = zeros(1, length(t));
m_sim = zeros(1, length(t));
h_sim = zeros(1, length(t));
n_sim = zeros(1, length(t));
m_ca_sim = zeros(1, length(t));
m_kca_sim = zeros(1, length(t));
x_sim = zeros(1, length(t));
m_kahp_sim = zeros(1, length(t));
%i_ca_sim = zeros(1, length(t));
ca_conc = zeros(1, length(t));
spikes = zeros(1, length(t));

v_s_sim(1) = v_s_init;
v_d_sim(1) = v_d_init;
m_sim(1) = m_init;
h_sim(1) = h_init;
n_sim(1) = n_init;
m_ca_sim(1) = m_ca_init;
m_kca_sim(1) = m_kca_init;
x_sim(1) = x_init;
m_kahp_sim(1) = m_kahp_init;
%i_ca_sim(1) = i_ca_init;
ca_conc(1) = ca_conc_init;


% To count the number of somatic spikes, a spike will be recorded when
 the membrane
% potential exceeds v_exceeds. The variable blocking will be set to 1,
% which will prevent more spikes from being recorded until the
 membrane
% potential falls below v_unblock, upon which the variable blocking
 will
% be set back to 0.

blocking = 0;
v_exceeds = -10e-3;
v_unblock = -30e-3;

% March forward in time.
for n = 1:(length(t)-1)
```

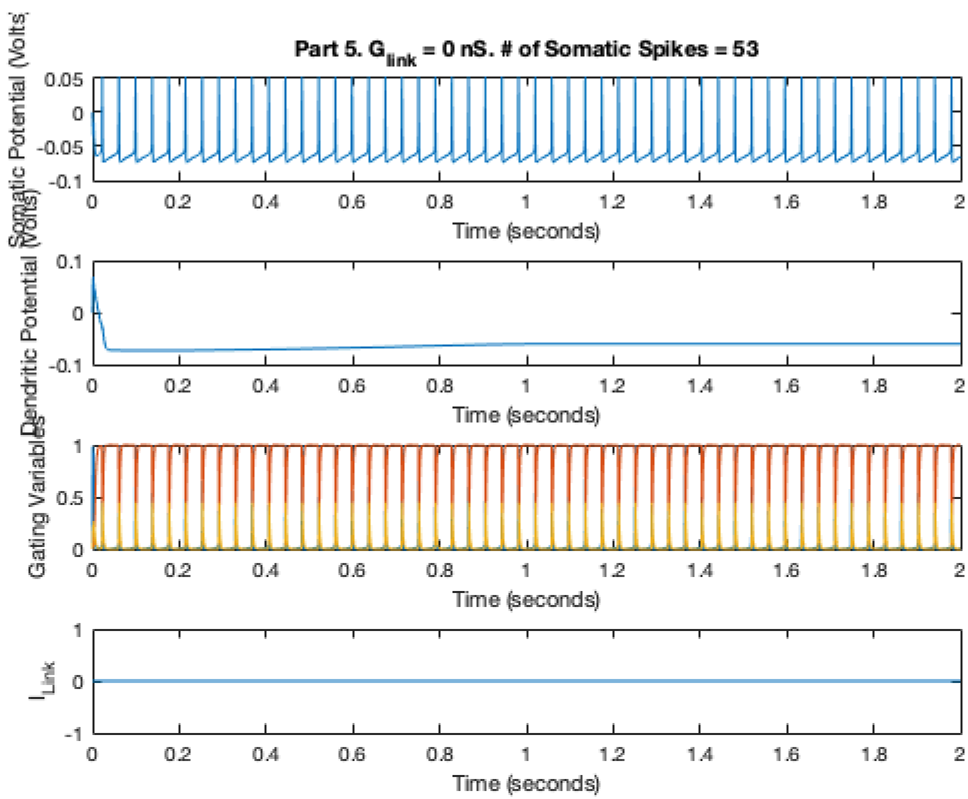
```
    if v_s_sim(n) > v_exceeds
        if blocking == 0
            spikes(n) = 1;
            blocking = 1;
        end
    end

    if v_s_sim(n) < v_unblock
        blocking = 0;
    end

    % Update v_s_sim.
    term1 = g_s_leak*(e_leak-v_s_sim(n));
    term2 = g_max_na*(m_sim(n)^2)*h_sim(n)*(e_na-v_s_sim(n));
    term3 = g_max_k*(n_sim(n)^2)*(e_k-v_s_sim(n));
    term4 = g_link*(v_d_sim(n)-v_s_sim(n));
    term5 = i_s_applied(n);
    v_s_sim(n+1) = v_s_sim(n) + (dt/
C_s)*(term1+term2+term3+term4+term5);

    % Update v_d_sim.
    term1 = g_d_leak*(e_leak-v_d_sim(n));
    term2 = g_max_ca*(m_ca_sim(n)^2)*(e_ca-v_d_sim(n));
    term3 = g_max_kca*(m_kca_sim(n))*x_sim(n)*(e_k-v_d_sim(n));
    term4 = g_max_kahp*m_kahp_sim(n)*(e_k-v_d_sim(n));
    term5 = -g_link*(v_d_sim(n)-v_s_sim(n));
    term6 = i_d_applied(n);
    v_d_sim(n+1) = v_d_sim(n) + (dt/
C_d)*(term1+term2+term3+term4+term5+term6);

    % Compute somatic rate constants.
    [alpha_m, beta_m, alpha_h, beta_h , alpha_n, beta_n] =
 PR_soma_gating(v_s_sim(n));

    % Update m_sim.
    m_inf = alpha_m/(alpha_m + beta_m);
    tau_m = 1/(alpha_m + beta_m);
    m_sim(n+1) = m_sim(n) + dt*((m_inf - m_sim(n))/tau_m);

    % Update h_sim.
    h_inf = alpha_h/(alpha_h+beta_h);
    tau_h = 1/(alpha_h+beta_h);
    h_sim(n+1) = h_sim(n) + dt*((h_inf - h_sim(n))/tau_h);

    % Update n_sim.
    n_inf = alpha_n/(alpha_n + beta_n);
    tau_n = 1/(alpha_n + beta_n);
    n_sim(n+1) = n_sim(n) + dt*((n_inf - n_sim(n))/tau_n);

    % Compute dendritic rate constants.
    [alpha_mca, beta_mca, alpha_kca, beta_kca, alpha_kahp, ...
        beta_kahp ] = PR_dend_gating( v_d_sim(n), ca_conc(n));

    % Update m_ca_sim.
```

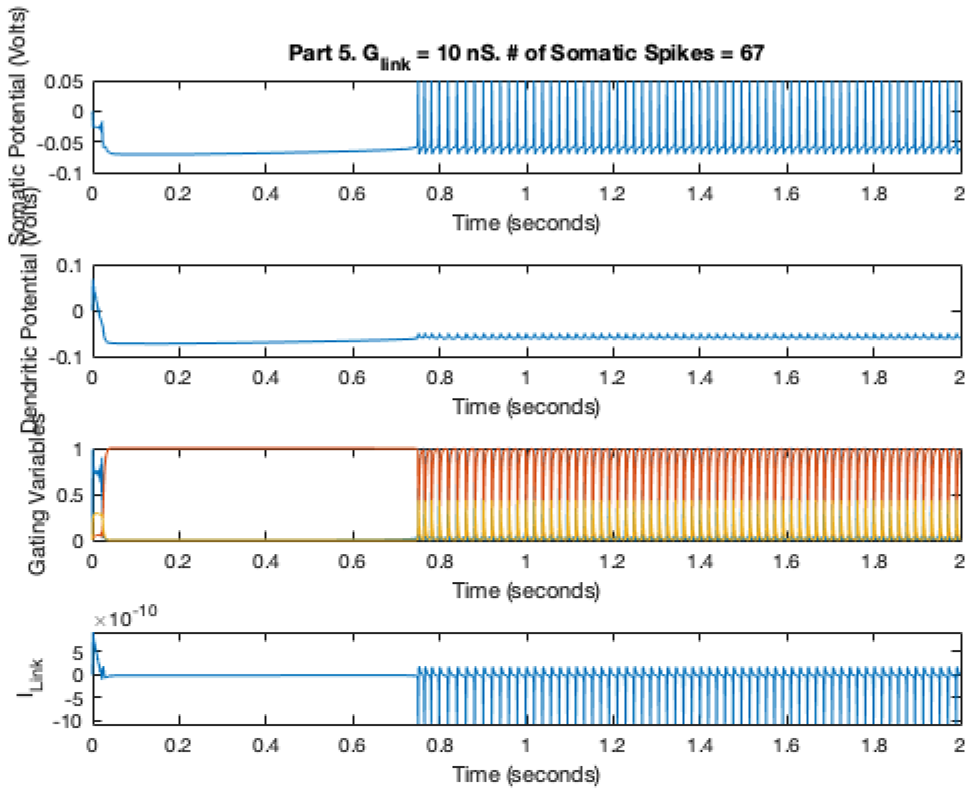
```
    inf = alpha_mca/(alpha_mca+beta_mca);
    tau = 1/(alpha_mca+beta_mca);
    m_ca_sim(n+1) = m_ca_sim(n) + dt*((inf - m_ca_sim(n))/tau);

    % Update m_kca_sim.
    inf = alpha_kca/(alpha_kca+beta_kca);
    tau = 1/(alpha_kca+beta_kca);
    m_kca_sim(n+1) = m_kca_sim(n) + dt*((inf - m_kca_sim(n))/tau);

    %Update m_kahp.
    inf = alpha_kahp/(alpha_kahp + beta_kahp);
    tau = 1/(alpha_kahp + beta_kahp);
    m_kahp_sim(n+1) = m_kahp_sim(n) + dt*((inf - m_kahp_sim(n))/tau);

    % Update ca_conc.
    term1 = g_max_ca*(m_ca_sim(n)^2)*(e_ca-v_d_sim(n));
    ca_conc(n+1) = ca_conc(n) + dt*( (-ca_conc(n)/tau_ca) + k*term1);

    % Update x_sim.
    %{
    inf = min([4000*ca_conc(n) 1]);
    x_sim(n+1) = x_sim(n) + dt*(inf - x_sim(n));
    %}
    x_sim(n+1) = min(4000*ca_conc(n+1),1);
end
end

function [I_applied] = applied_current(baseline, step, t)
% Returns a vector for applied current given input baseline and step
% currents.
global dt
    I_applied = zeros(1, length(t));
    third = floor(length(t)/3);
    twothird = 2*third;
    I_applied(1:third) = baseline;
    I_applied(third+1:twothird) = baseline+step;
    I_applied(twothird+1:length(I_applied)) = baseline;
end
```

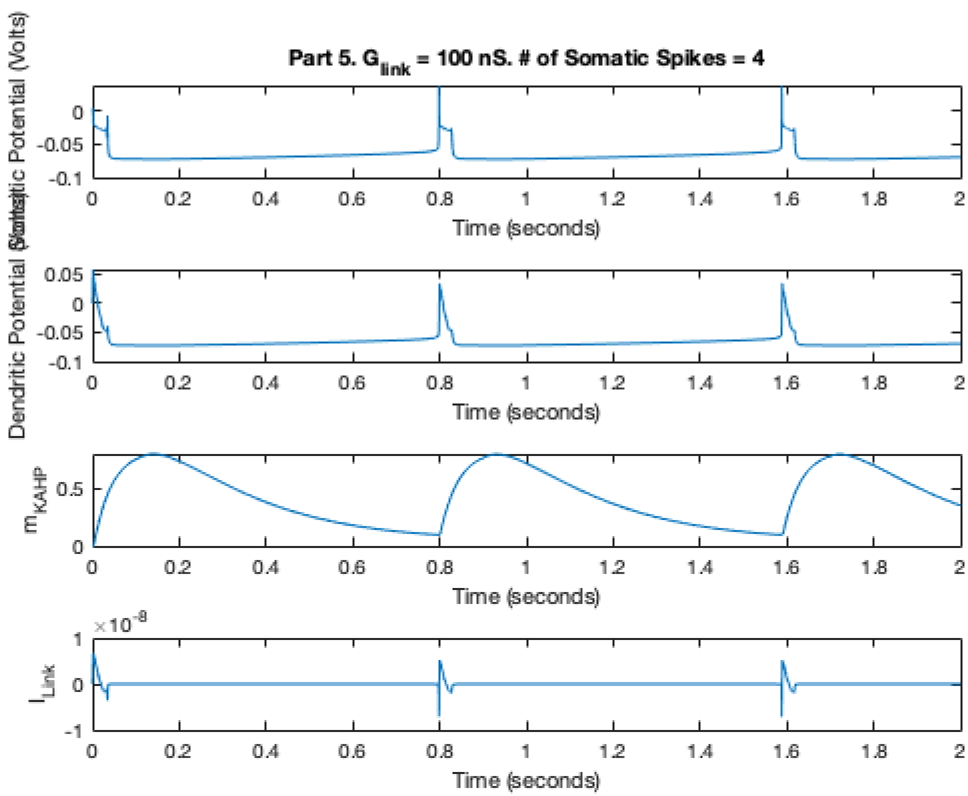

Somatic Rate Constants
×10^4
α_m
β_m
α_h
β_h
α_n
β_n
Membrane Potential (Volts)
Membrane Potential (Volts)
Membrane Potential (Volts)

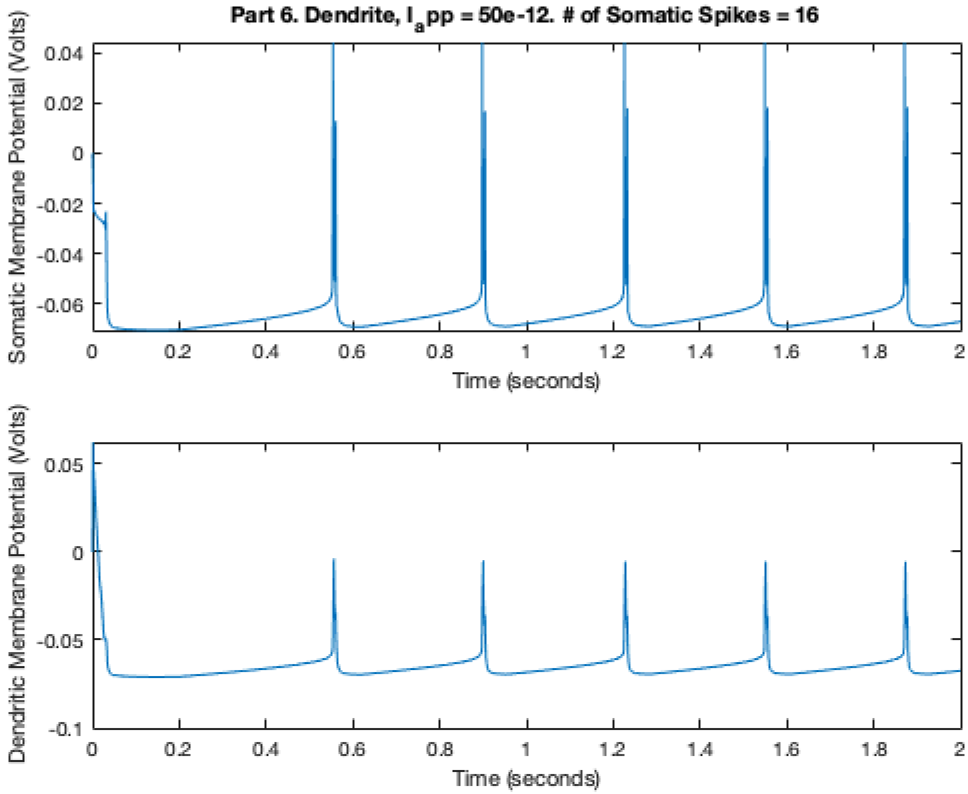

Dendritic Rate Constants
α_{mca}
β_{mca}
α_{kca}
β_{kca}
α_{kahp}
β_{kahp}
Membrane Potential (Volts)
Membrane Potential (Volts)
Calcium Concentration

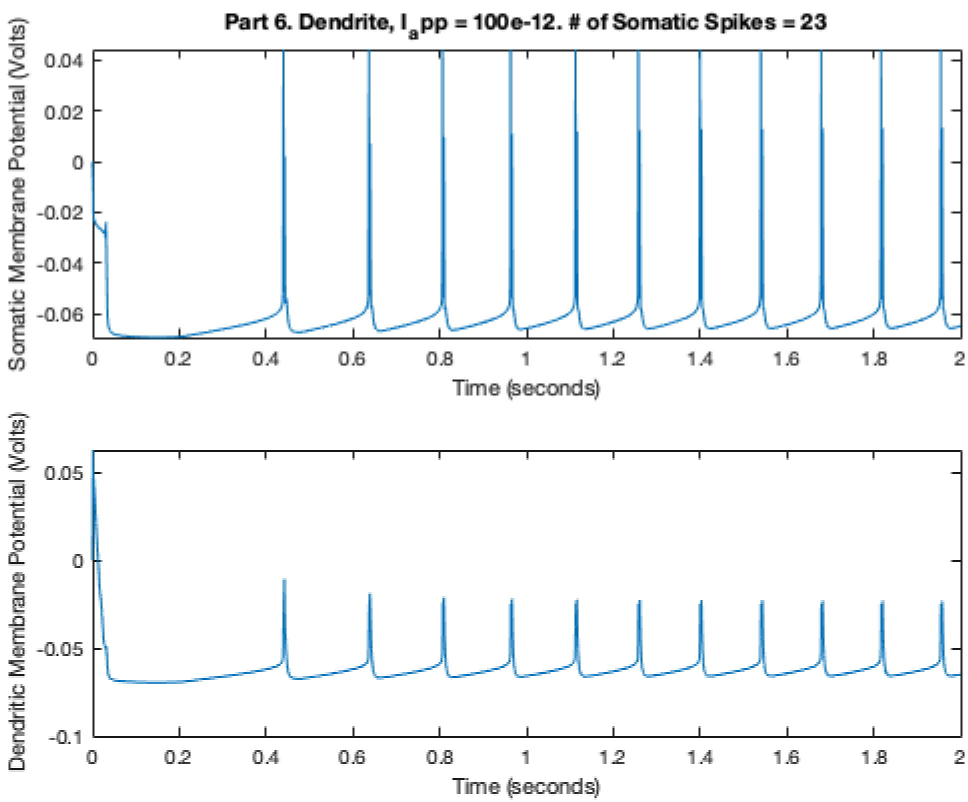

Part 3 and 4. # of Somatic Spikes = 9
Somatic Membrane Potential (Volts)
Time (seconds)
Dendritic Membrane Potential (Volts)
Time (seconds)

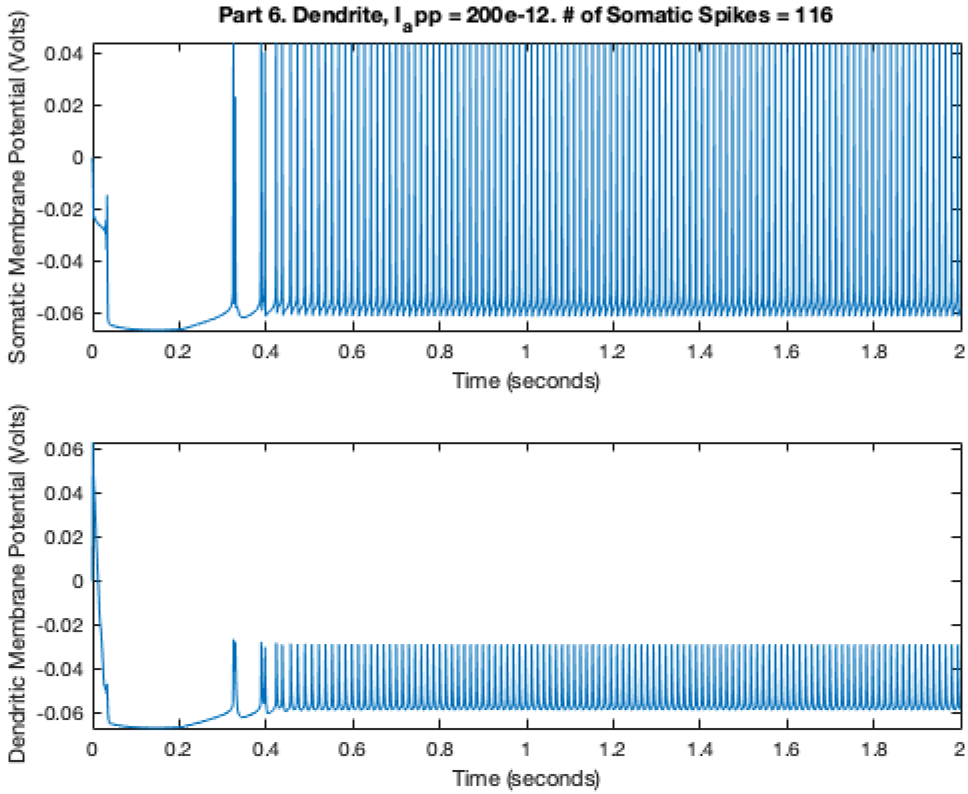

Part 5. G_link = 0 nS. # of Somatic Spikes = 53
Somatic Potential (Volts)
Time (seconds)
Dendritic Potential (Volts)
Time (seconds)
Gating Variables
Time (seconds)
I_Link
Time (seconds)


Part 5. G_{link} = 10 nS. # of Somatic Spikes = 67
Somatic Potential (Volts)
Dendritic Potential (Volts)
Gating Variables
I_{Link}
$\times 10^{-10}$
Time (seconds)

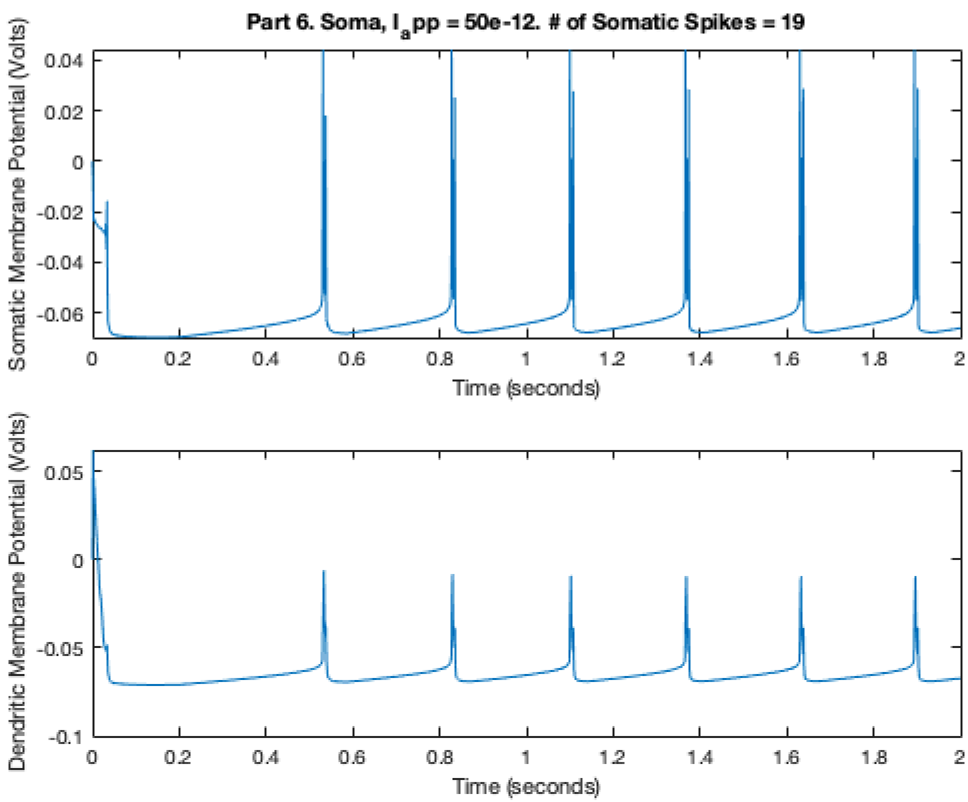

Part 5. G_{link} = 100 nS. # of Somatic Spikes = 4
Somatic Potential (Volts)
Dendritic Potential (Volts)
m_{KAHP}
I_{Link}
$\times 10^{-8}$
Time (seconds)


Part 6. Dendrite, I_app = 50e-12. # of Somatic Spikes = 16
Somatic Membrane Potential (Volts)
Time (seconds)
Dendritic Membrane Potential (Volts)
Time (seconds)

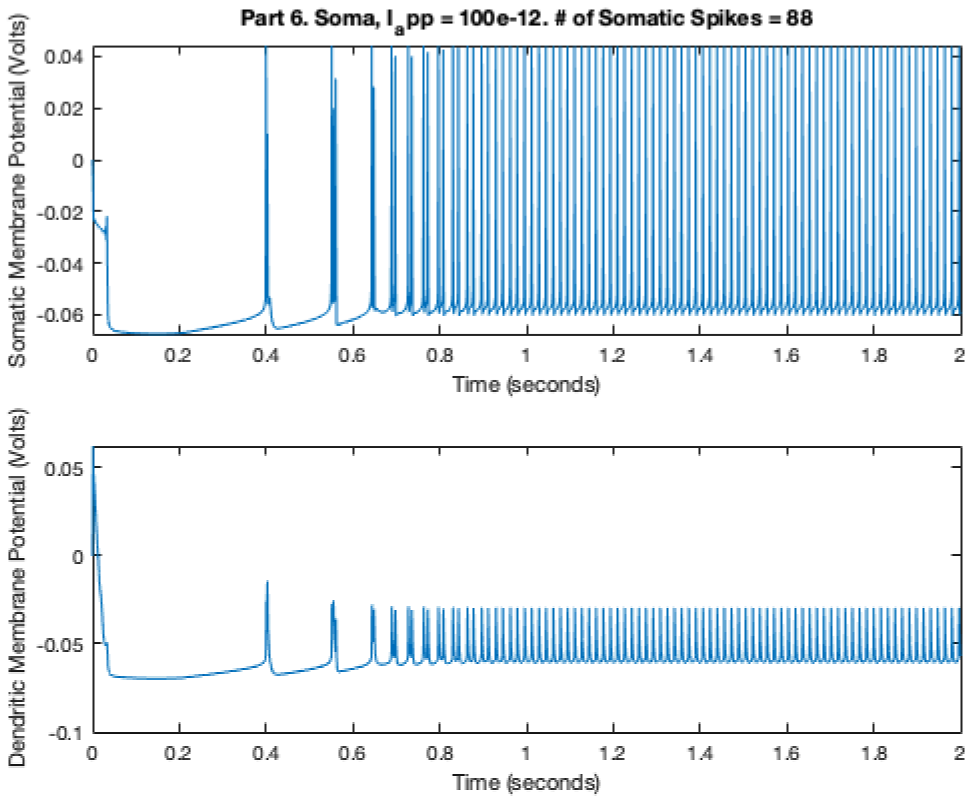

Part 6. Dendrite, I_app = 100e-12. # of Somatic Spikes = 23
Somatic Membrane Potential (Volts)
Time (seconds)
Dendritic Membrane Potential (Volts)
Time (seconds)

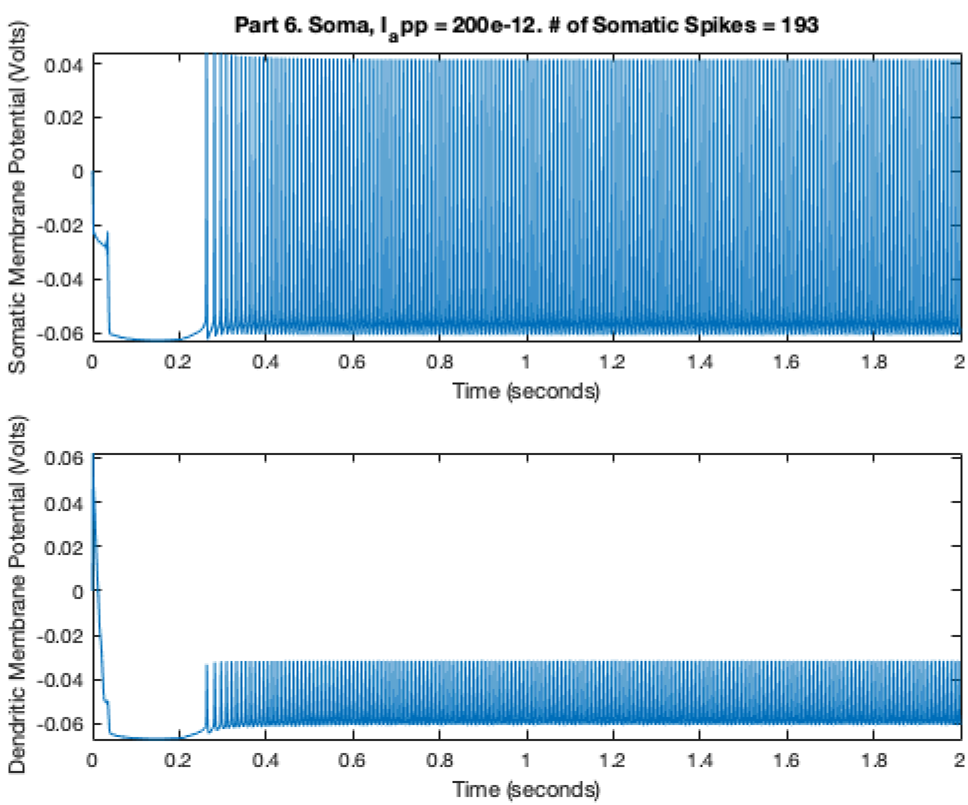


Part 6. Dendrite, I_app = 200e-12. # of Somatic Spikes = 116
Somatic Membrane Potential (Volts)
Time (seconds)
Dendritic Membrane Potential (Volts)
Time (seconds)



Part 6. Soma, I_app = 50e-12. # of Somatic Spikes = 19
Somatic Membrane Potential (Volts)
Time (seconds)
Dendritic Membrane Potential (Volts)
Time (seconds)


Part 6. Soma, $I_a$pp = 100e-12. # of Somatic Spikes = 88
Somatic Membrane Potential (Volts)
Time (seconds)
Dendritic Membrane Potential (Volts)
Time (seconds)


Part 6. Soma, $I_a$pp = 200e-12. # of Somatic Spikes = 193
Somatic Membrane Potential (Volts)
Time (seconds)
Dendritic Membrane Potential (Volts)
Time (seconds)

Published with MATLAB® R2018a