Bhargavi Poyekar
Ch33454

CMSC 691: Neural Eng Assignment 3

Q1. I.

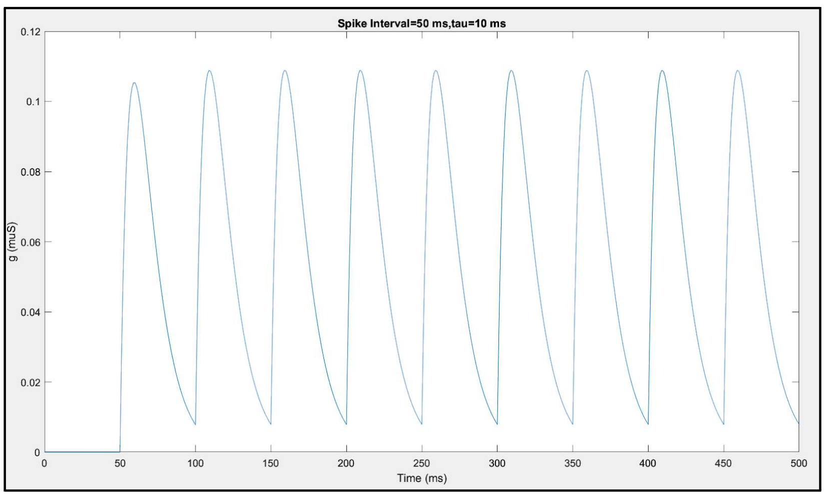
Code:

```
% input values
spike_interval=input("Enter spike interval: ");
tau=input("Enter value of tau: ");

time=0:500; % time range 0 to 500 (Tmax given 500)
delta_t=1; % dt
g_final=[]; % array for g values

% initial values
g=0;
z=0;

% iterative method for calculating z and g
for t=0:500
   g_final=[g_final,g]; % store g in array
   z=z+delta_t*z_func(z,tau,t,spike_interval);
   g=g+delta_t*g_func(g,tau,z);
end

% Plot conductance vs time
plot(time, g_final)
title(sprintf('Spike Interval=%d ms,tau=%d ms',spike_interval,tau))
xlabel('Time (ms)')
ylabel('g (muS)')

% Function for spike_interval
function u_t=u_t(spike_interval,t)
   if mod(t,spike_interval)==0 && t~=0
       u_t=1;
   else
       u_t=0;
   end
end

% function to calculate dg/dt
function g_func=g_func(g,tau,z)
   g_func=(-g/tau)+z;
end
% function to calculate dz/dt
```

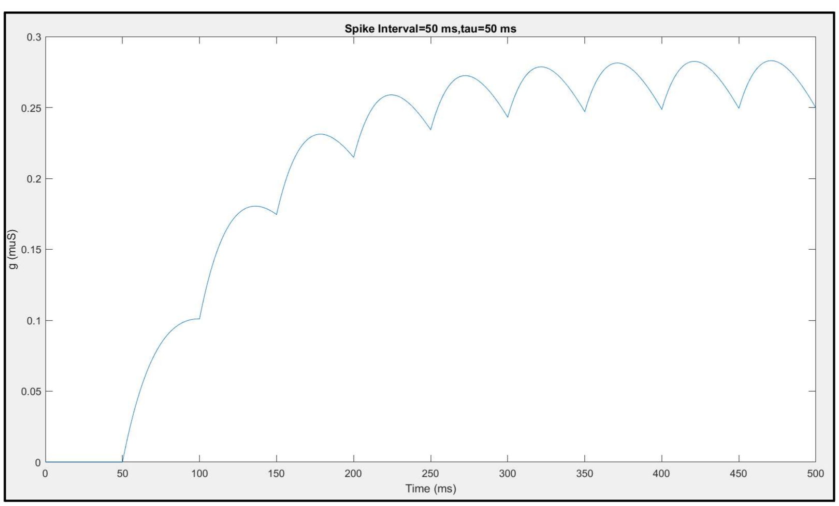
```
function z_func=z_func(z,tau,t,spike_interval)
   z_func=(-z/tau)+(Gnorm(tau)*u_t(spike_interval,t));
end

% function to calculate e^(-t/tau)
function eterm=eterm(t,tau)
   eterm=exp(-t/tau);
end

% function for Gnorm
function Gnorm=Gnorm(tau)
   e=2.718;
   gpeak=0.1;
   Gnorm=gpeak/(tau/e);
end
```

Output:

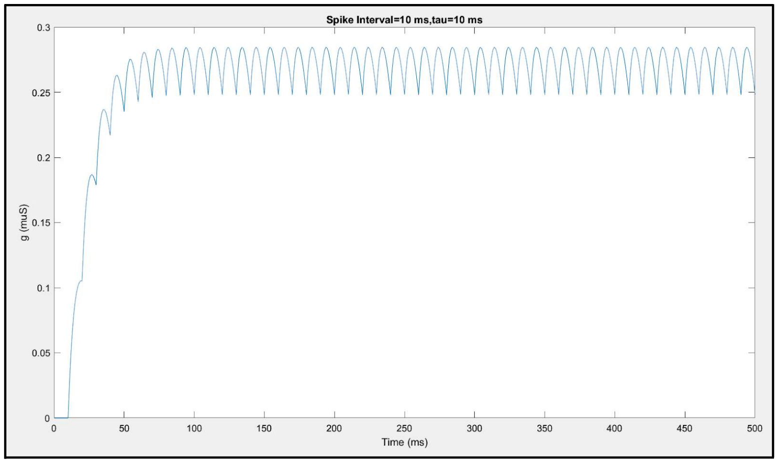
a) Spike interval=50 ms, Tau=10 ms:



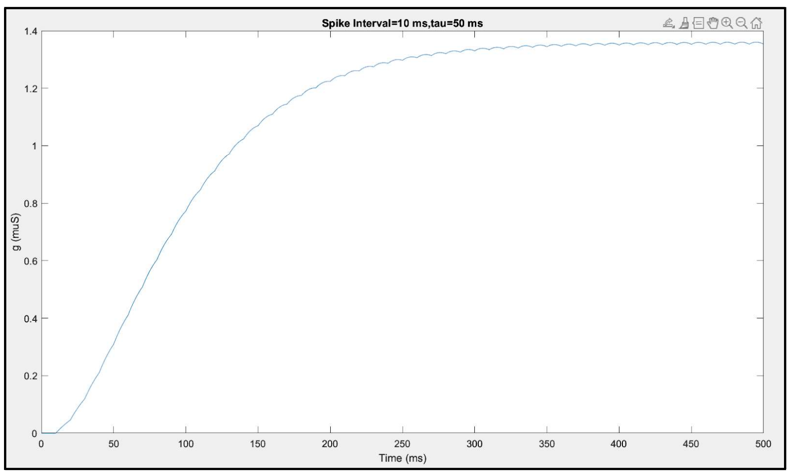
b) Spike interval=50 ms, Tau=50 ms:



c) Spike interval=10 ms, Tau=10 ms:



d) Spike interval=10 ms, Tau=50 ms:



Q1. II.

Code:

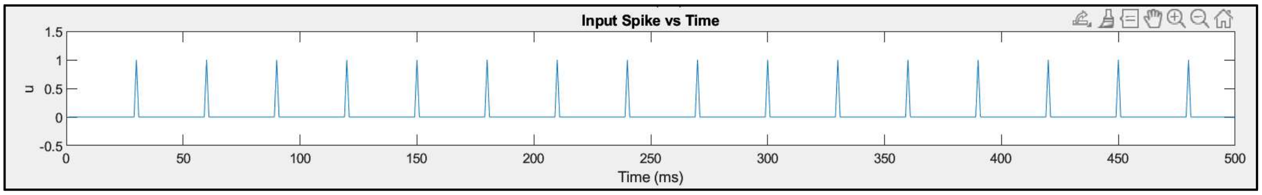
```
% input values
spike_interval=30;
tau=10;
Iinj=0; % Assumed to be zero
E=70;
Vthr=5;
Vspk=70;
time=1:500; % time range 0 to 500 (Tmax given 500)
delta_t=1; % dt
g_final=[]; % array for g values
u=[];
voltage=[];
```

```matlab
Isyn=[];
% initial values
g=0;
z=0;
v=0;
I=0;
% iterative method for calculating z and g
for t=1:500

   g_final=[g_final,g]; % store g in array
   voltage=[voltage,v];
   u=[u,u_t(spike_interval,t)];
   Isyn=[Isyn,I];
   I=g*(v-E); % Synaptic Current

   if v~=Vspk % If V(t-1) is not spiked
       v=v+delta_t*v_func(g,v,Iinj);
   else
       v=0; % if V(t-1) is spiked, new v=0
   end
   % Spiking V at Threshold
   if v>=Vthr
       v=Vspk;
   end
   % Calculating conductance and z
   g=g+delta_t*g_func(g,tau,z);
   z=z+delta_t*z_func(z,tau,t,spike_interval);

end
% Plot conductance vs time
subplot(4,1,1)
plot(time, g_final)
title('Synaptic Conductance vs Time')
xlabel('Time (ms)')
ylabel('g (muS)')
% Plot input spike train vs time
subplot(4,1,2)
plot(time, u)
title('Input Spike vs Time')
xlabel('Time (ms)')
ylabel('u')
ylim([-0.5,1.5])
% Plot Synaptic Current vs time
subplot(4,1,3)
plot(time,Isyn)
title('Synaptic Current vs Time')
xlabel('Time (ms)')
ylabel('Isyn')
% Plot Postsynaptic Membrane Voltage vs time
```

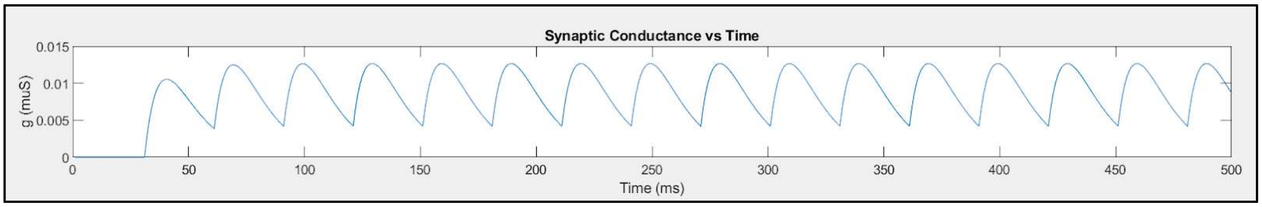
```
vthr_arr=linspace(5,5,length(time));
subplot(4,1,4)
plot(time, voltage);
hold on;
plot(time,vthr_arr,'r-');
hold off;
title('Postsynaptic Membrane Voltage vs Time')
xlabel('Time (ms)')
ylabel('v (volts)')
legend('','Vthr')
% Function for dv/dt
function v_func=v_func(g,v,Iinj)
   C=1;
   R=10;
   E=70;
   v_func=(1/C)*((-v/R)-(g*(v-E))+Iinj);
end
% Function for spike_interval
function u_t=u_t(spike_interval,t)
   if mod(t,spike_interval)==0 && t~=0
       u_t=1;
   else
       u_t=0;
   end
end
% function to calculate dg/dt
function g_func=g_func(g,tau,z)
   g_func=(-g/tau)+z;
end
% function to calculate dz/dt
function z_func=z_func(z,tau,t,spike_interval)
   z_func=(-z/tau)+(Gnorm(tau)*u_t(spike_interval,t));
end
% function to calculate e^(-t/tau)
function eterm=eterm(t,tau)
   eterm=exp(-t/tau);
end
% function for Gnorm
function Gnorm=Gnorm(tau)
   e=2.718;
   gpeak=0.01;
   Gnorm=gpeak/(tau/e);
end
```

Output:

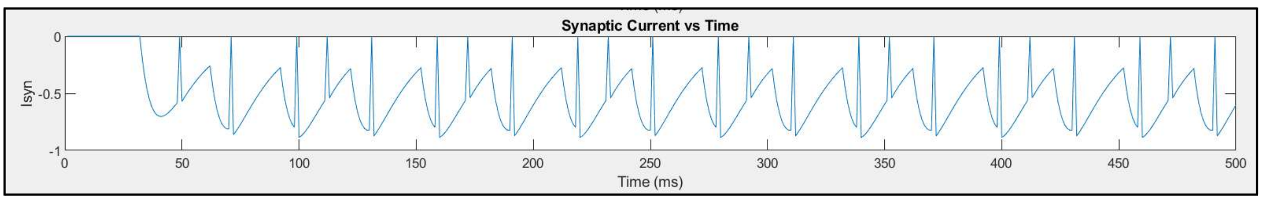
a) Input spike train vs Time:



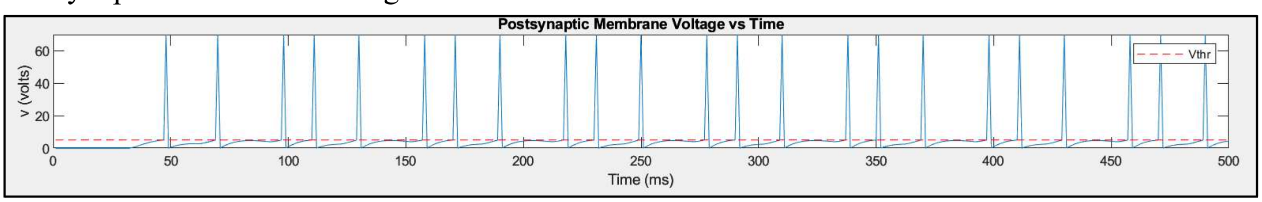
b) Synaptic conductance $g_{ex}(t)$ vs Time:



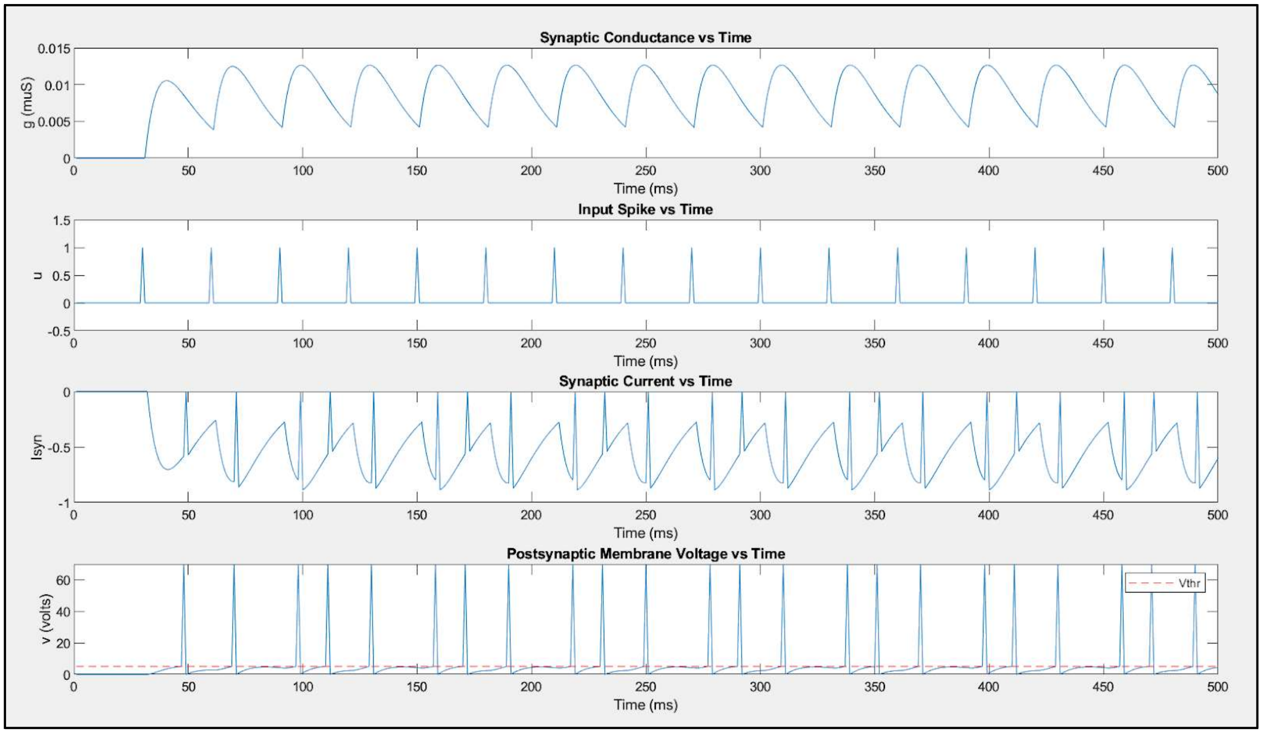
c) Synaptic current Isync(t) vs Time:



d) Postsynaptic Membrane Voltage:



All Outputs in One frame:



Q2. Two Neuron Oscillator:

Code:

```
tau_syn=15;
tau_thresh=50;

Iinj1=1.1; % Injected current in Neuron1
Iinj2=0.9; % Injected current in Neuron2

Einh=-15; % Synaptic reverse Potential
Vspk=70; % Spiked Voltage -> Action potential amplitude
time=1:1500; % time range 0 to 500 (Tmax given 500)
delta_t=1; % dt

voltage1=[]; % Membrane Potential for Neuron 1
voltage2=[]; % Membrane Potential for Neuron 1
vthresh1=[]; % Threhold level for neuron 1
vthresh2=[]; % Threhold level for neuron 2

% initial values
g1=0;
g2=0;
z1=0;
z2=0;
v1=0;
```

```
v2=0;
vthr1=0;
vthr2=0;

% iterative method
for t=1:1500

   voltage1=[voltage1,v1];
   voltage2=[voltage2,v2];
   vthresh1=[vthresh1,vthr1];
   vthresh2=[vthresh2,vthr2];

   % Spike inpu for Neuron 2
   if v1==Vspk
       u2=1;
   else
       u2=0;
   end

   % Spike inpu for Neuron 1
   if v2==Vspk
       u1=1;
   else
       u1=0;
   end

   % Neuron 1
   vthr1=vthr1+delta_t*vthr_func(vthr1,v1,tau_thresh);

   if v1~=Vspk
       v1=v1+delta_t*v_func(g1,v1,Iinj1);
   else
       v1=Einh;
   end

   if v1>=vthr1
       v1=Vspk;
   end

   g1=g1+delta_t*g_func(g1,tau_syn,z1);
   z1=z1+delta_t*z_func(z1,tau_syn,u1);

   % Neuron 2
   vthr2=vthr2+delta_t*vthr_func(vthr2,v2,tau_thresh);

   if v2~=Vspk
       v2=v2+delta_t*v_func(g2,v2,Iinj2);
   else
       v2=Einh;
```

```
    end

    if v2>=vthr2
        v2=Vspk;
    end

    g2=g2+delta_t*g_func(g2,tau_syn,z2);
    z2=z2+delta_t*z_func(z2,tau_syn,u2);

end

% Plot Membrane Voltage of Neuron 1 and Neuron 2 vs time
plot(time, voltage1,'b-');
hold on;
plot(time,vthresh1,'r--');
hold on;
plot(time, 100+voltage2,'g-');
hold on;
plot(time,100+vthresh2,'r--');
hold off;

title('Membrane Voltage vs Time for Neuron 1 and Neuron 2')
xlabel('Time (ms)')
ylabel('Membrane Potential')
legend('Neuron1','Vthr','Neuron2')

% Function for Threhold
function vthr_func=vthr_func(vthr,v,tau_thresh)
    vthr_func=((-vthr+v)/tau_thresh);
end

% Function for dv/dt
function v_func=v_func(g,v,Iinj)
    C=1;
    R=10;
    E=-15;
    v_func=(1/C)*((-v/R)-(g*(v-E))+Iinj);
end

% Function for spike_interval
function u_t=u_t(spike_interval,t)
    if mod(t,spike_interval)==0 && t~=0
        u_t=1;
    else
        u_t=0;
    end
end

% function to calculate dg/dt
```

```
function g_func=g_func(g,tau,z)
    g_func=(-g/tau)+z;
end

% function to calculate dz/dt
function z_func=z_func(z,tau,u)
    z_func=(-z/tau)+(Gnorm(tau)*u);
end

% function to calculate e^(-t/tau)
function eterm=eterm(t,tau)
    eterm=exp(-t/tau);
end

% function for Gnorm
function Gnorm=Gnorm(tau)
    e=2.718;
    gpeak=0.1;
    Gnorm=gpeak/(tau/e);
end
```

Output: