EE 746 Neuromorphic Engineering
Assignment 2

Devesh Kumar, 16D070044
Yojit Srivastava, 160020073

1. AEF NEURON DRIVEN BY A SYNAPSE RECEIVING POISSON STIMULUS

1.1 Part a

```
function [stimulus] = stimulus_gen(t,del_t,lambda)
 threshold = lambda*del_t;
 ans = rand(t/del_t,1);
 stimulus =logical(ans<threshold);
end
```

1.2 Part b

1.2.1 matlab code

```
        clear;
close all;
t_max = 0.5;
 dt = 0.1*10^-3;
 lambda = 10;
 io = 1*10^-12;
 we = 500;
 tou = 15*10^-3;
 tou_s = tou/4;
 m = t_max/dt;
 %iapp = ones(m,1).*250* 10^(-12);
 [stimulus,iapp] = gen_iapp(t_max,dt,lambda,io,we,tou, tou_s,m);
    [u_in,v_in] = steadystate();
    [c,gl,el,vt,del_T,a,tw,b,vr] = getvalue(1);
    %iapp =[ 250* 10^(-12), 350* 10^(-12), 450* 10^(-12)];

    V = zeros(m,1);
    U = zeros(m,1);
    V(1,:) = v_in(1);
    U(1,:) = u_in(1);
    i = 1;
    for j  = 2:m
        V(j,i) = V(j-1,i) + (gl*(del_T*exp((V(j-1,i)-vt)/del_T)) - gl*(V(j-1,i)-el) - U(j-1,i) + iapp(j))
        U(j,i) =  U(j-1,i) +(a*(V(j-1,i)- el) - U(j-1,i))*(dt/tw);
        if(V(j-1,i)==0)
            V(j,i) = vr;
            U(j,i) = U(j-1,i) + b;
        end
        if(V(j,i)>=0)
            V(j,i) = 0;
        end
    end
```

```
    x = zeros(1,m);
    for i = 2:m
        x(i) = x(i-1) +dt;
    end
    figure(1)
    plot(x,V(:,1))
    title(["Neuron "+"Rs"])
    xlabel('Time(in s)')
    ylabel('Voltage (in V)')
    figure(2)
    plot(x,stimulus)
    title("Stimulus")
    xlabel('Time(in s)')
    figure(3)
    subplot(2,1,1)
    plot(x,V(:,1))
    title(["Voltage"])
    xlabel('Time(in s)')
    ylabel('Voltage (in V)')

    subplot(2,1,2)
 % figure(3)
    plot(x,iapp(:,1))
    title(["input current"])
    xlabel('Time(in s)')
    ylabel('iapp (in A)')


        function [stimulus,ans] = gen_iapp(t_max,dt,lambda,io,we,tou, tou_s,m)
threshold = lambda*dt;
dum = rand(t_max/dt,1);
stimulus =logical(dum<threshold);

i_app = zeros(m,1);
tm = [];
for i = 1:m
   if length(tm) ~= 0
       for j  = 1:length(tm)
           dum = (i*dt - tm(j));
           i_app(i) = i_app(i) + io*we*(exp(-1*dum/tou) - exp (-1*dum/tou_s));
       end
   end
   if stimulus(i) == 1
       tm = cat(2,tm,[i*dt]);
   end
end
ans = i_app;


end
```

1.2.2 Plots

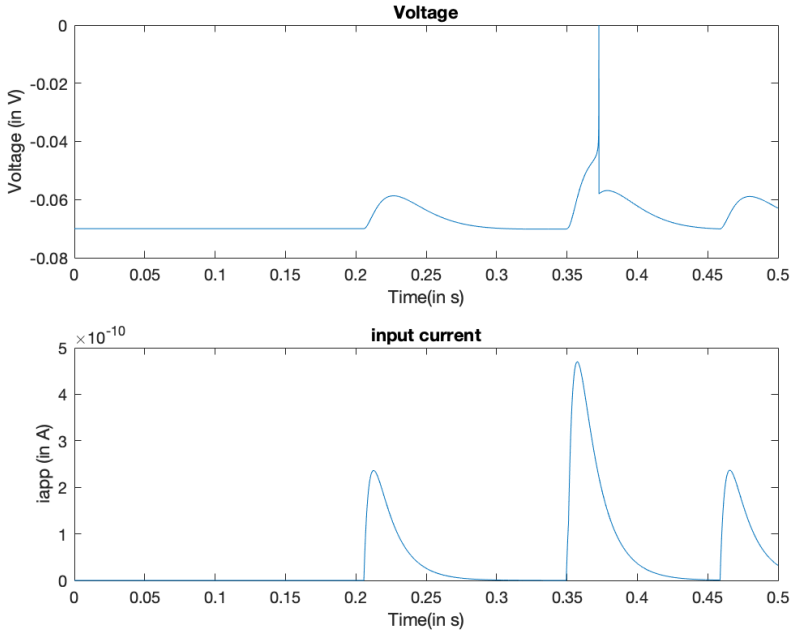

Figure 1. Average Time Interval Between Spikes

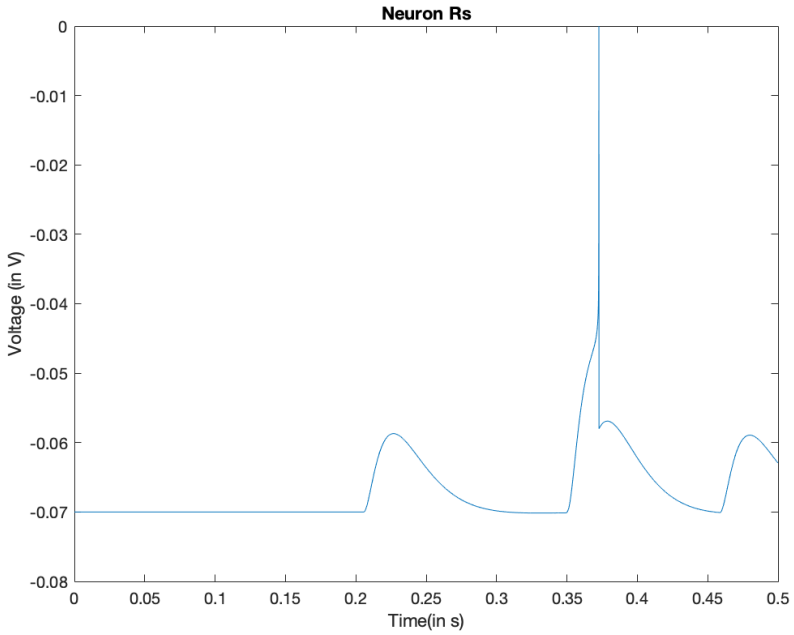

Figure 2. Average Time Interval Between Spikes

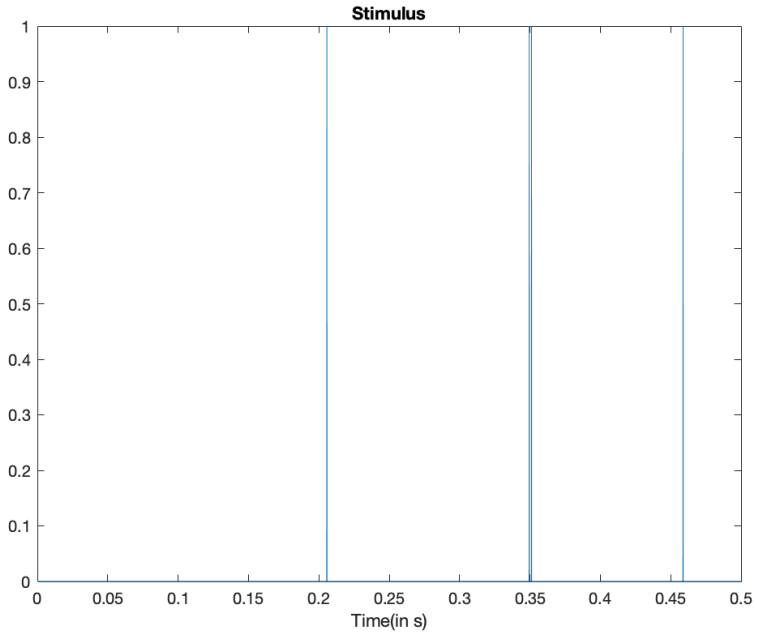

Figure 3. Average Time Interval Between Spikes

2. AEF NEURON DRIVEN BY MULTIPLE SYNAPSES

2.1 Part A

No of spikes = 0
total current = 2.5707e-07 A

2.1.1 Plots

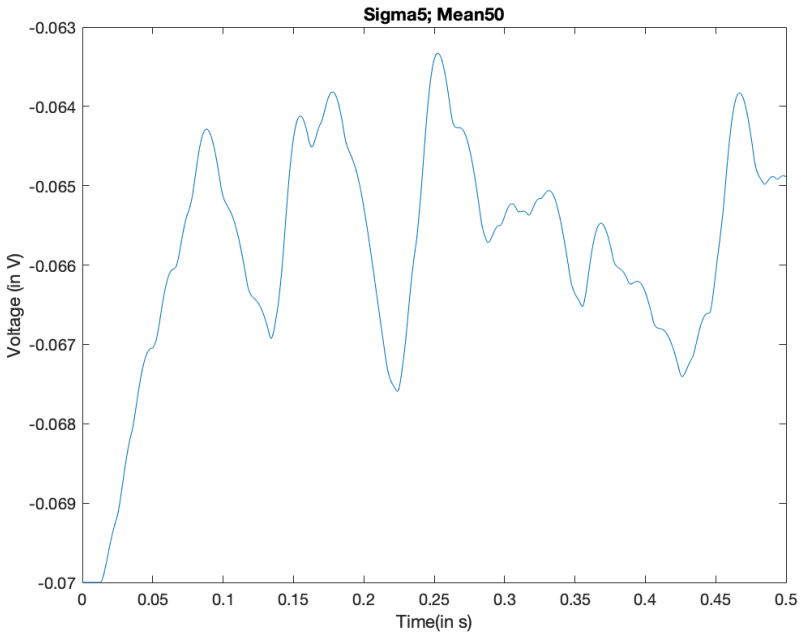

Figure 4. Response of the neuron

2.2 Part B

No of spikes =14
total current = 1.2773e-06 A

2.2.1 Plots

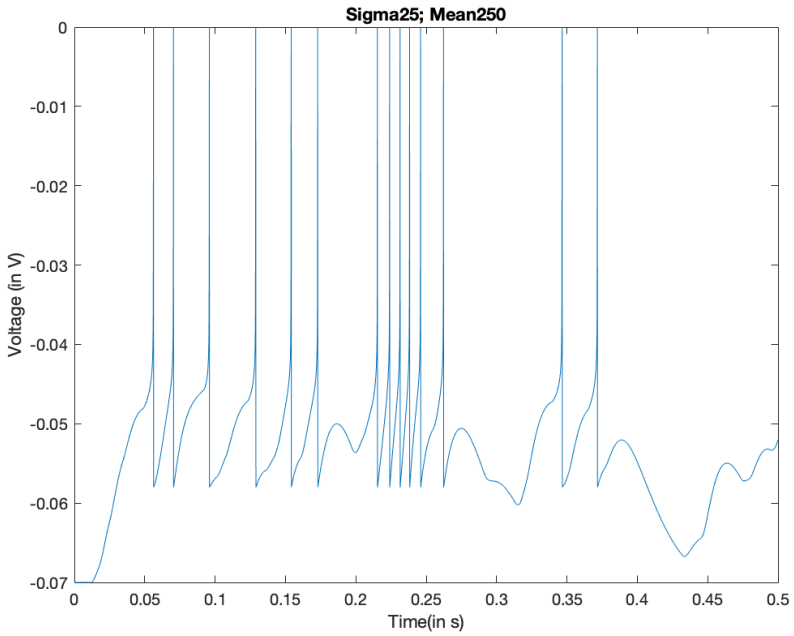

Figure 5. Response of the neuron

2.3 Matlab code

```
    function [stimulus,ans] = gen_iapp(t_max,dt,lambda,io,tou, tou_s,m,wo,sig_w)
w(1:100,1)=(wo+sig_w*randn(100,1));


 i_app = zeros(m,100);
 for k = 1:100
     tm = [];
     threshold = lambda*dt;
     dum = rand(t_max/dt,1);
     stimulus =logical(dum<threshold);
     for i = 1:m
        if length(tm) ~= 0
            for j  = 1:length(tm)
                dum = (i*dt - tm(j));
                i_app(i,k) = i_app(i,k) + io*w(k)*(exp(-1*dum/tou) - exp (-1*dum/tou_s));
            end
        end
        if stimulus(i) == 1
            tm = cat(2,tm,[i*dt]);
        end
     end
 end

 ans = sum(transpose(i_app));
```

```
end
```

3. ADJUSTING THE WEIGHTS TO ELICIT A SPIKE RESPONSE

3.1 PartA

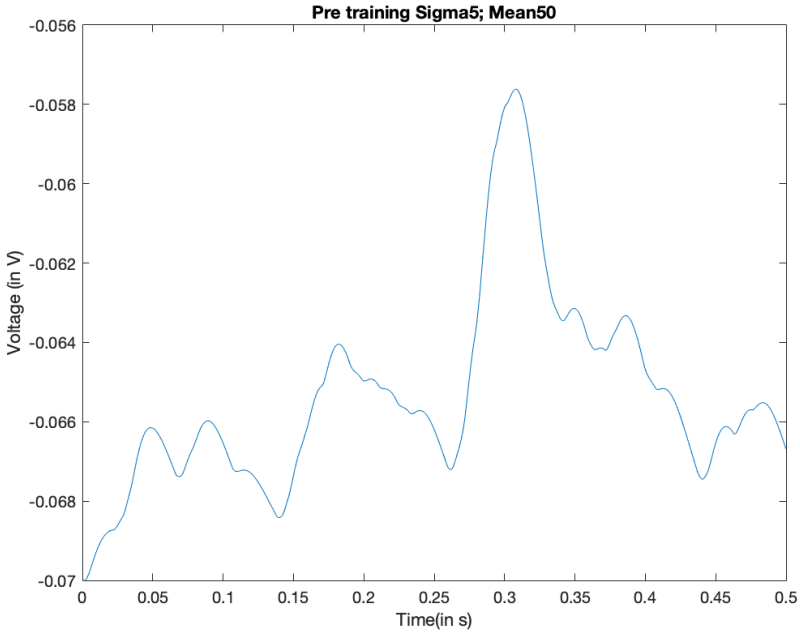

Figure 6. Pre training

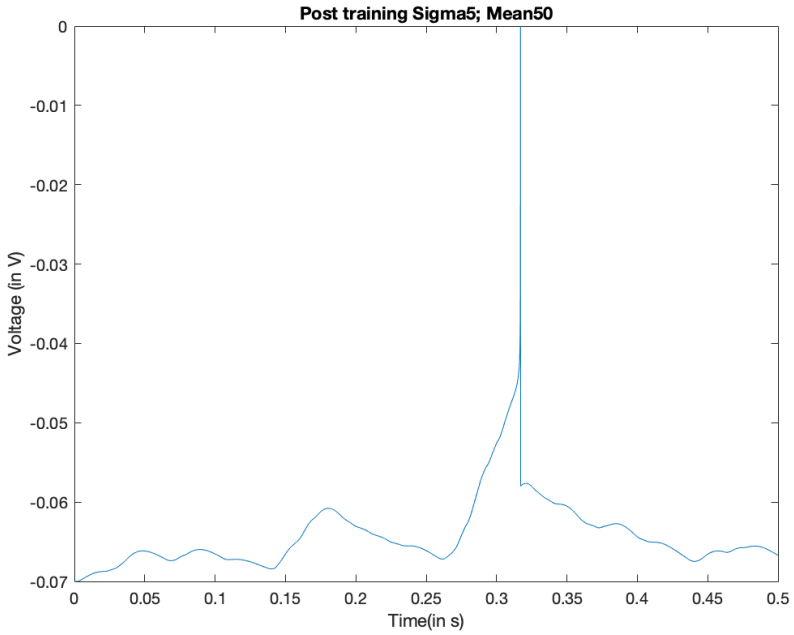

Figure 7. Post traning

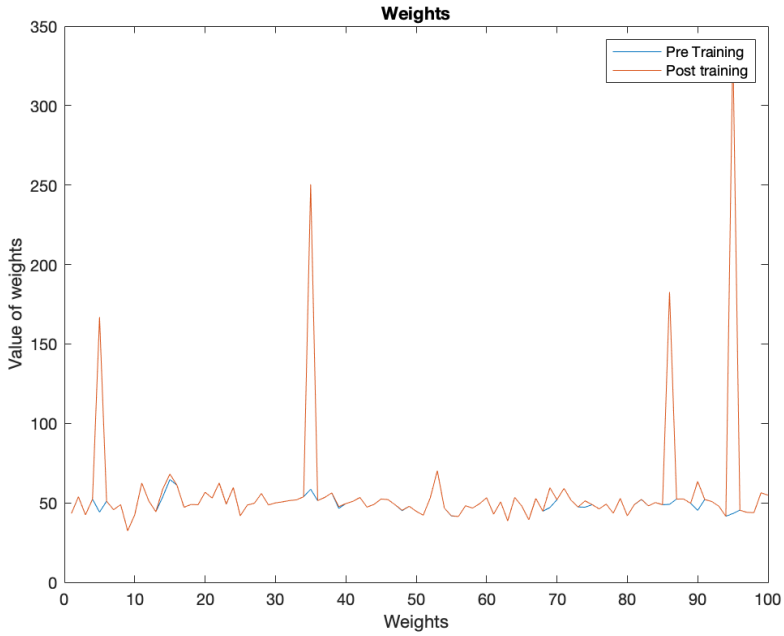

Figure 8. Weights

3.2 PartB

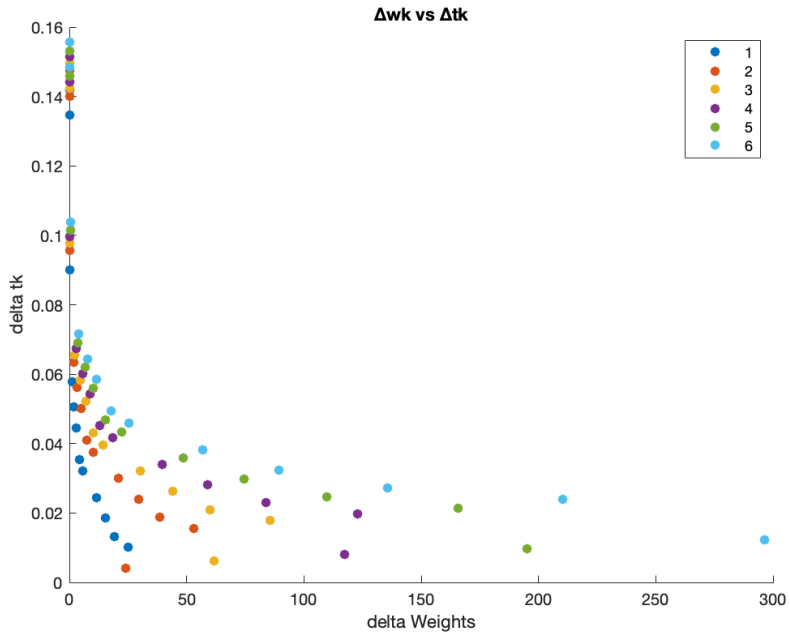

Figure 9. weights

3.3 Matlab code

```
    close all;
clear;
t_max = 0.5;
 dt = 0.1*10^-3;
 lambda = 1;
 io = 1*10^-12;
 tou = 15*10^-3;
 tou_s = tou/4;
 m = t_max/dt;
 wo = 50;
 sig_w = 5;
 w(1:100,1)=(wo+sig_w*randn(100,1));
 stimulus = zeros(m,100);
 for k = 1:100
     threshold = lambda*dt;
     dum = rand(t_max/dt,1);
     stimulus(:,k) =logical(dum<threshold);
 end

[pre_iapp] = gen_iapp(stimulus,t_max,dt,lambda,io,tou, tou_s,m,w);
[V,x] = aef(pre_iapp);
itr = 1
pre_V = V;
pre_w = w;
```

```
w_all =[];
del_tk_all = [];
%cat(2,w_all,w);
    figure(4)
    title(["wk vs tk"])
    %legend("Pre Training","Post training")
    xlabel('delta Weights')
    ylabel('delta tk')
    hold on
while(1)

    [post_w,del_tk] = training(w,dt,stimulus,V);
    [post_iapp] = gen_iapp(stimulus,t_max,dt,lambda,io,tou, tou_s,m,post_w);
    [post_V,x] = aef(post_iapp);
    del_w = post_w - pre_w;
    del_w = nonzeros(del_w);
   % w_all = cat(2,w_all,del_w(1:size()));
   % del_tk_all = cat(2,del_tk_all,del_tk);
    scatter(del_w,del_tk,'filled')
    if max(post_V) == 0
        break
    end

    w = post_w;
    V = post_V;
    itr = itr +1
end
legend("1","2","3","4","5","6")
hold off
    figure(1)
    plot(x,pre_V(:,1))
    title(["Pre training Sigma"+sig_w+"; Mean" + wo])
    xlabel('Time(in s)')
    ylabel('Voltage (in V)')
    figure(2)
    plot(x,post_V(:,1))
    title(["Post training Sigma"+sig_w+"; Mean" + wo])
    xlabel('Time(in s)')
    ylabel('Voltage (in V)')

    y = [1:100] ;
    figure(3)
    plot(y,pre_w,y,post_w)
    title(["Weights"])
    legend("Pre Training","Post training")
    xlabel('Time(in s)')
    ylabel('Voltage (in V)')


    function [V,x] = aef(iapp)
t_max = 0.5;
 dt = 0.1*10^-3;
 lambda = 1;
```

```
 io = 1*10^-12;
 tou = 15*10^-3;
 tou_s = tou/4;
 m = t_max/dt;

 %iapp = ones(m,1).*250* 10^(-12);
    %[stimulus,iapp] = gen_iapp(t_max,dt,lambda,io,tou, tou_s,m,wo,sig_w);
    [u_in,v_in] = steadystate();
    [c,gl,el,vt,del_T,a,tw,b,vr] = getvalue(1);
    %iapp =[ 250* 10^(-12), 350* 10^(-12), 450* 10^(-12)];

    V = zeros(m,1);
    U = zeros(m,1);
    V(1,:) = v_in(1);
    U(1,:) = u_in(1);
    i = 1;
    for j  = 2:m
        V(j,i) = V(j-1,i) + (gl*(del_T*exp((V(j-1,i)-vt)/del_T)) - gl*(V(j-1,i)-el) - U(j-1,i) + iapp(j))
        U(j,i) =  U(j-1,i) +(a*(V(j-1,i)- el) - U(j-1,i))*(dt/tw);
        if(V(j-1,i)==0)
            V(j,i) = vr;
            U(j,i) = U(j-1,i) + b;
        end
        if(V(j,i)>=0)
            V(j,i) = 0;
        end
    end

    x = zeros(1,m);
    for i = 2:m
        x(i) = x(i-1) +dt;
    end
end

    function [new_w,del_tk] = training(w,dt,stimulus,V)

 tou = 15*10^-3;
 tou_s = tou/4;
del_tk = [];

[dum,t_max] = max(V);
new_w = w;

for i = 1:100
    for j = t_max-1:-1:1
        if stimulus(j,i) == 1
            dt_k = (t_max - j)*dt ;
            del_tk = cat(1,del_tk,dt_k);
            new_w(i) = new_w(i)  + w(i)*(exp(-1*dt_k/tou) - exp(-1*dt_k/tou_s));
            new_w(i) = min(new_w(i),500);
            break;
        end
    end
```

```
    %if new_w(i) == w(i)
   %     del_tk = cat(1,del_tk,0);
   % end

end
end
```

4. ADJUSTING THE WEIGHTS TO REMOVE ALL SPIKE RESPONSES

4.1 PartA

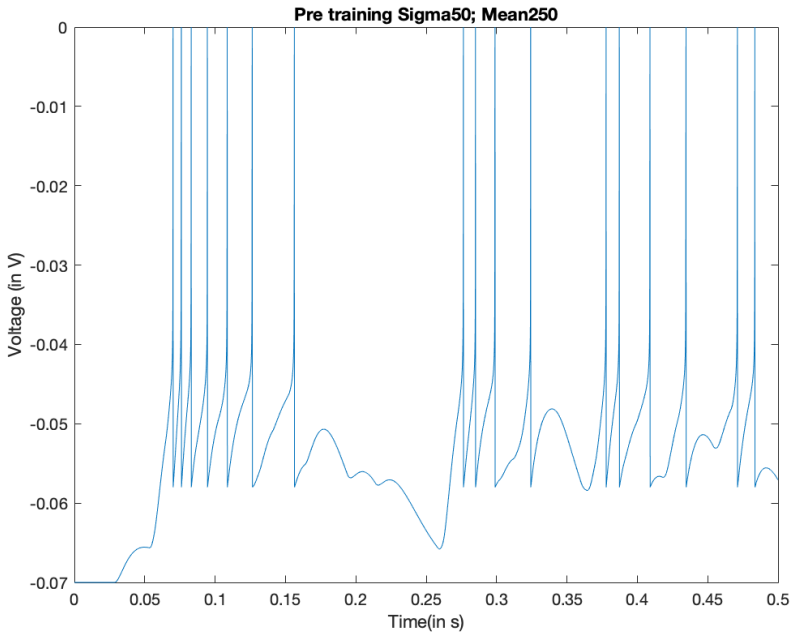

Figure 10. Pre traning

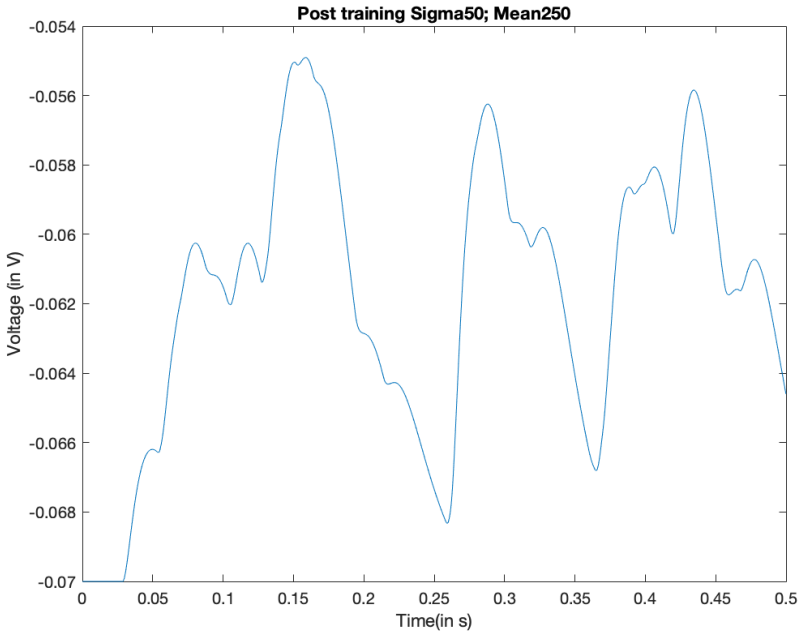

Figure 11. Post traning

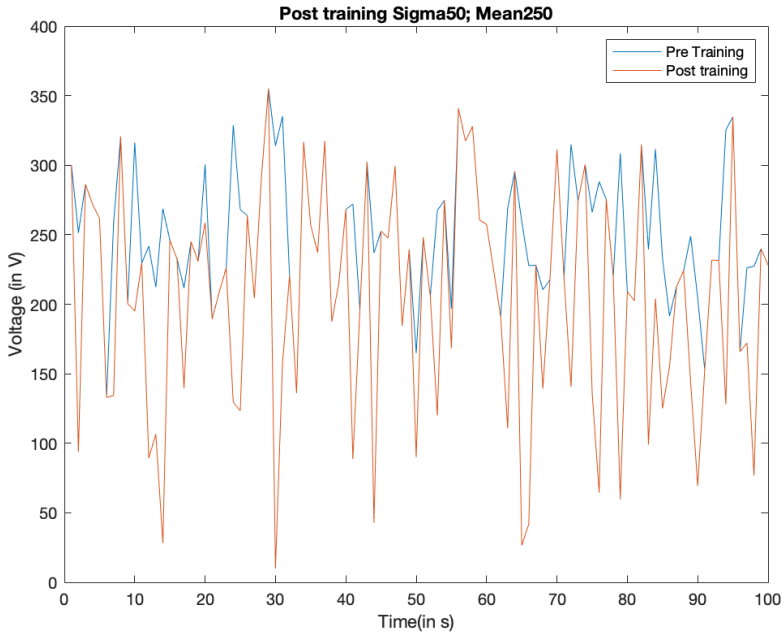

Figure 12. Post traning

4.2 PartB

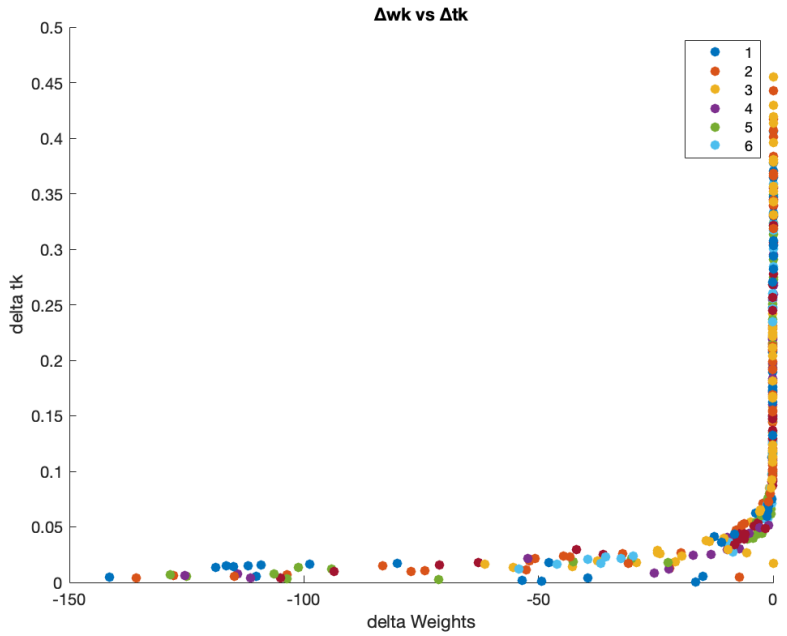

Figure 13. Post traning

4.3 Matlab Code

```
    function [new_w,del_tk]= training(w,dt,stimulus,V)

 tou = 15*10^-3;
 tou_s = tou/4;
del_tk = [];

peaks = find(V==0);
new_w = w;
    figure(4)
    title(["wk vs tk"])
    xlabel('delta Weights')
    ylabel('delta tk')
    hold on
for k = 1:length(peaks)
    t_max = peaks(k);
    del_w = [];
    del_tk = [];
    for i = 1:100
        for j = t_max-1:-1:1
            if stimulus(j,i) == 1
                dt_k = (t_max - j)*dt ;
                del_tk = cat(1,del_tk,dt_k);
                new_w(i) = new_w(i)  - w(i)*(exp(-1*dt_k/tou) - exp(-1*dt_k/tou_s));
                new_w(i) = max(new_w(i),10);
```

```
                del_w = cat(1,del_w,new_w(i) - w(i));
                break;
            end
        end
        scatter(del_w,del_tk,'filled')
    end

end
% legend("1","2","3","4","5","6")
% hold off
end
```

5. DISCRIMINATING STIMULI WITH SIMILAR STATISTICAL CHARACTERISTICS

5.1 PartA

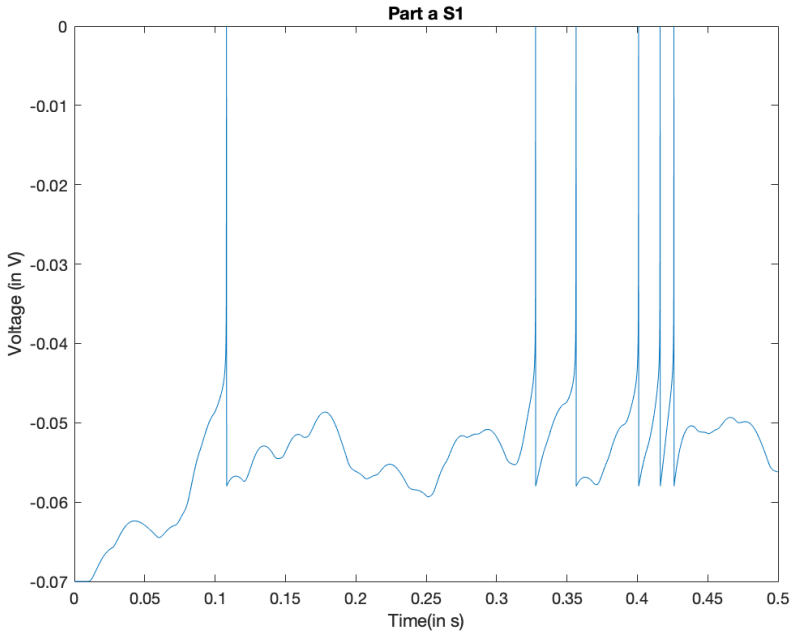

Figure 14. S2

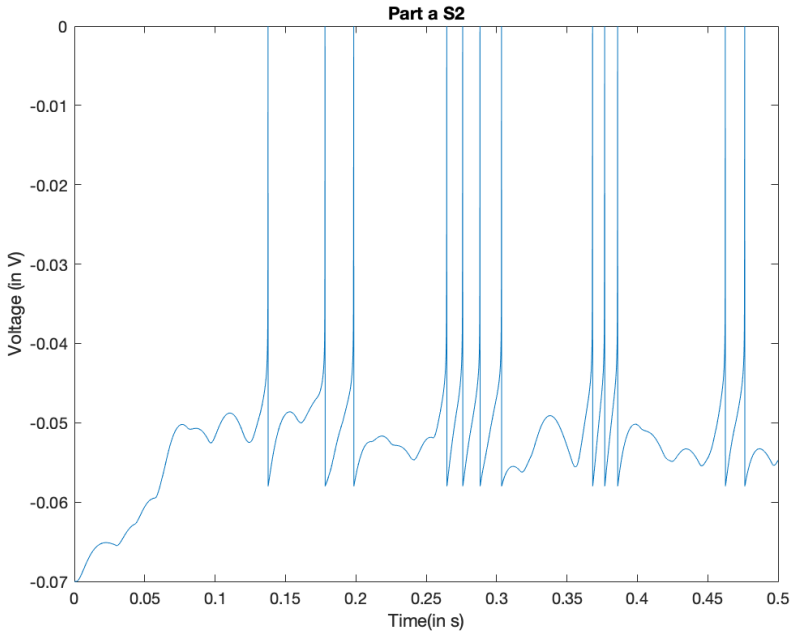

Figure 15. S2

5.2 PartB

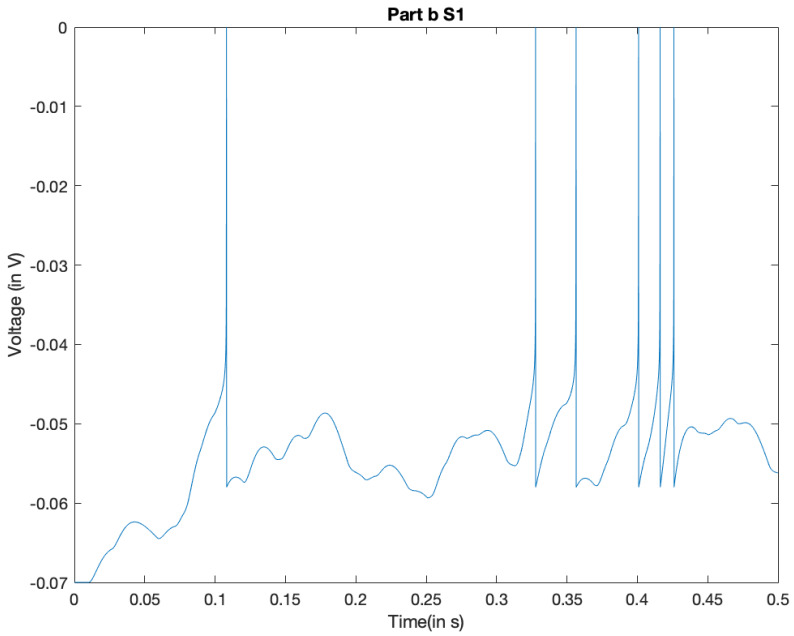

Figure 16. S1

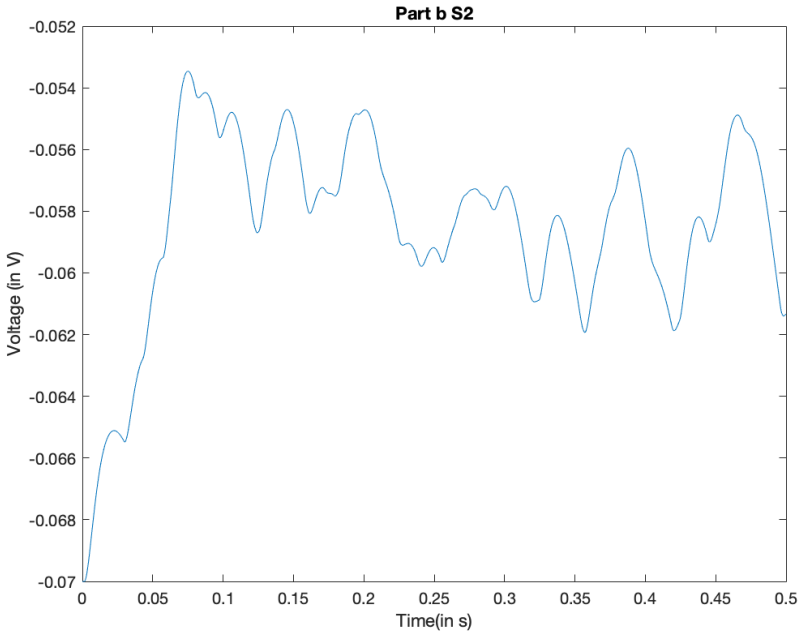

Figure 17. S2

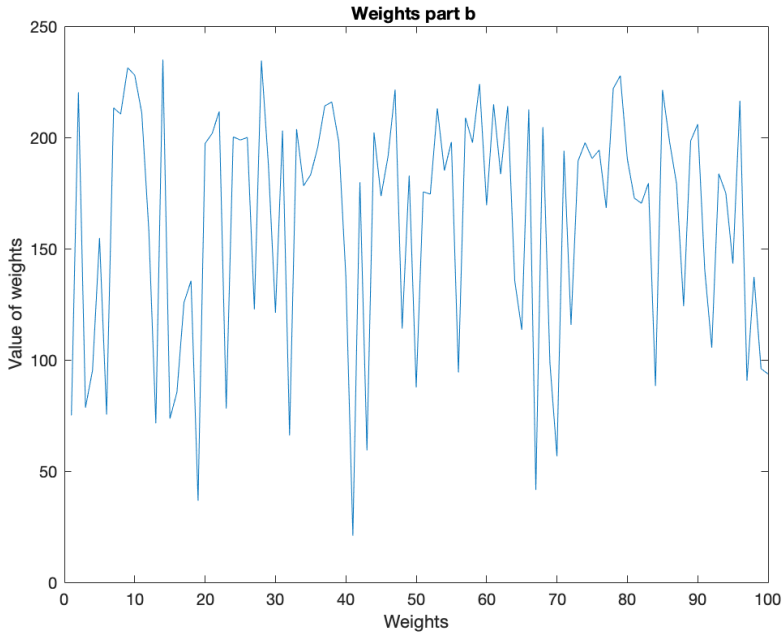

Figure 18. Resulting Weight

5.3 PartC

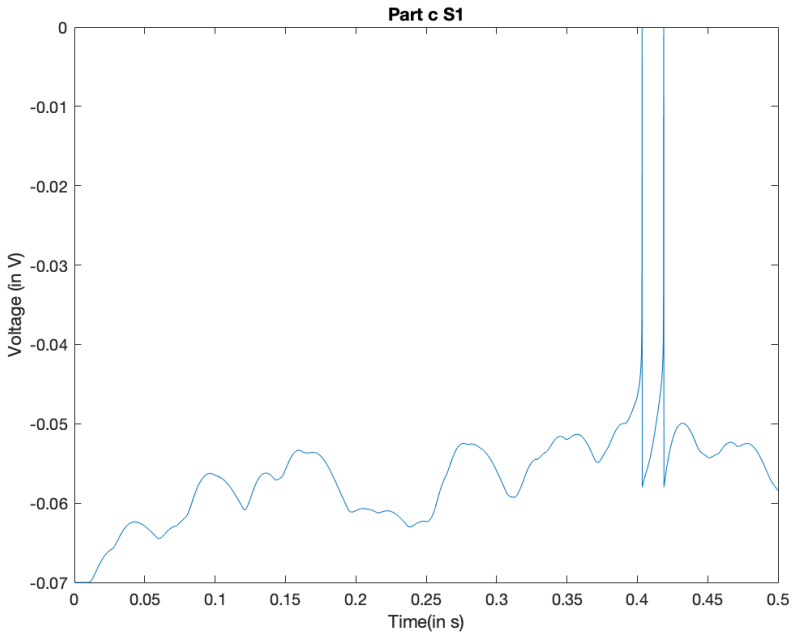

Figure 19. S1

5.4 PartD

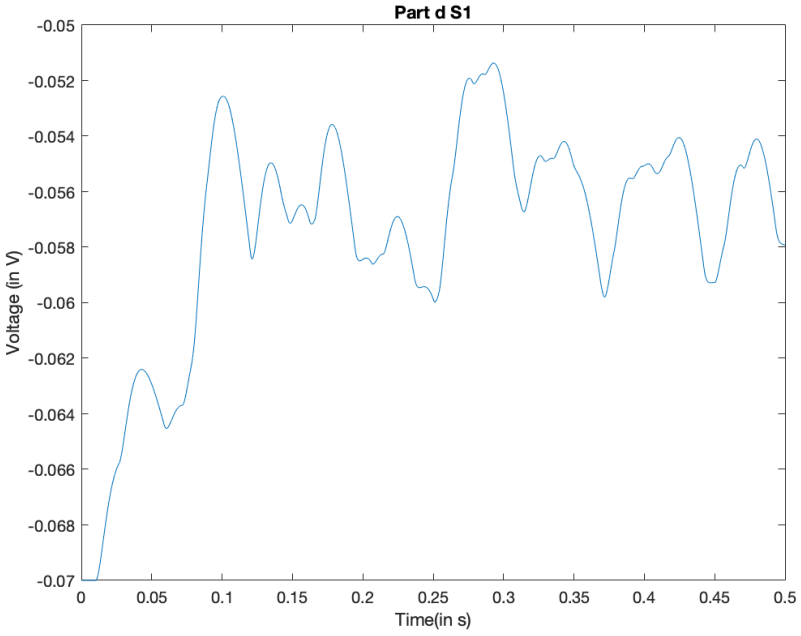

Figure 20. S1

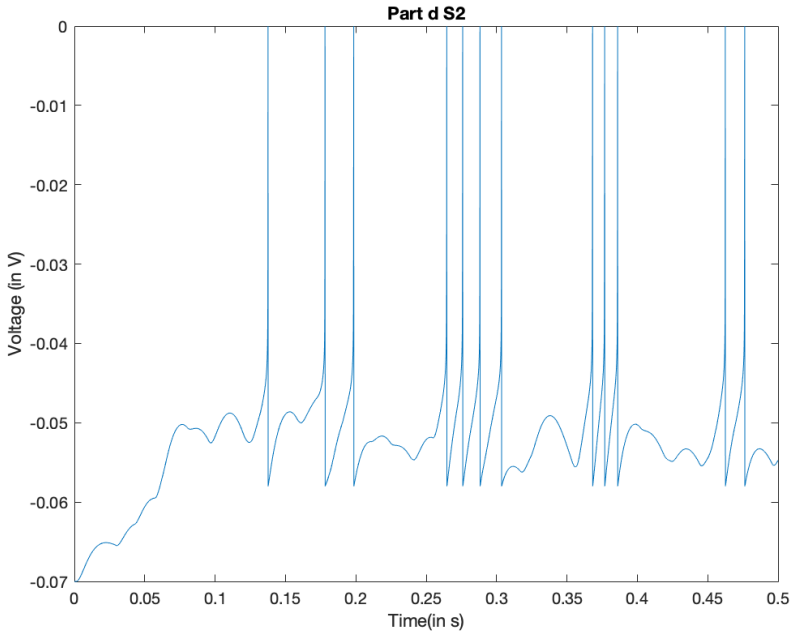

Figure 21. S2

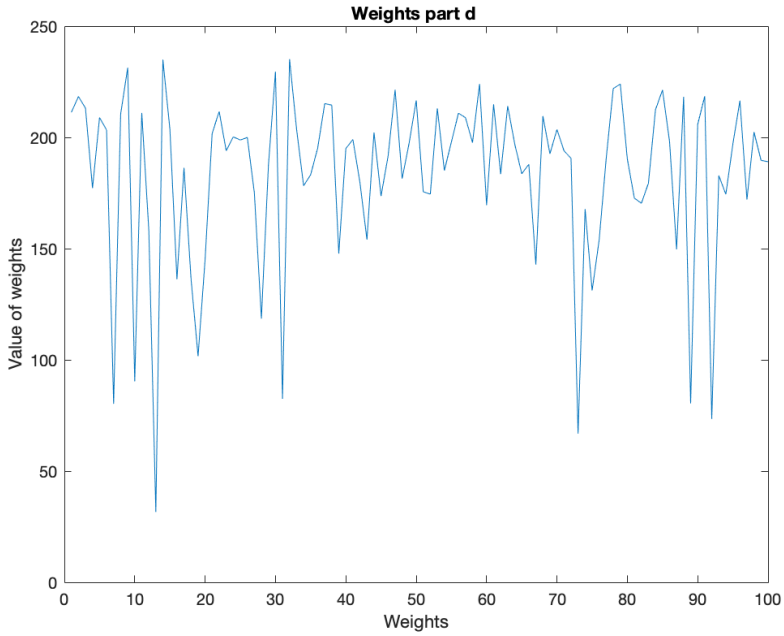

Figure 22. Resulting Weight

5.5 Matlab Code

```
close all;
```

```
clear;
t_max = 0.5;
dt = 0.1*10^-3;
lambda = 1;
io = 1*10^-12;
tou = 15*10^-3;
tou_s = tou/4;
m = t_max/dt;
wo = 200;
sig_w = 20;
w1(1:100,1)=(wo+sig_w*randn(100,1));
w2(1:100,1)=(wo+sig_w*randn(100,1));
w1 = w2;
w1_d = w1;
stimulus1 = zeros(m,100);
stimulus2 = zeros(m,100);
stimulus1_d = zeros(m,100);
stimulus2_d = zeros(m,100);
 for k = 1:100
     threshold = lambda*dt;
     dum1 = rand(t_max/dt,1);
     dum2 = rand(t_max/dt,1);
     stimulus1(:,k) =logical(dum1<threshold);
     stimulus2(:,k) =logical(dum2<threshold);
     stimulus1_d = stimulus1(:,k);
     stimulus2_d = stimulus2(:,k);
 end

[pre_iapp1] = gen_iapp(stimulus1,t_max,dt,lambda,io,tou, tou_s,m,w1);
[V1,x1] = aef(pre_iapp1);
[pre_iapp2] = gen_iapp(stimulus2,t_max,dt,lambda,io,tou, tou_s,m,w2);
[V2,x2] = aef(pre_iapp2);
V1_d = V1;
V2_d = V2;
    figure(1)
    plot(x1,V1(:,1))
    title(["Part a S1"])
    xlabel('Time(in s)')
    ylabel('Voltage (in V)')
        figure(2)
    plot(x2,V2(:,1))
    title(["Part a S2"])
    xlabel('Time(in s)')
    ylabel('Voltage (in V)')
%%%%%%%end part a(weights same)


%%%%%%start partb
itr = 1;
pre_V = V2;
pre_w = w2;
while(1)
```

```
    while(1)
        [post_w ,del_tk] = training(w2,dt,stimulus2,V2);
        [post_iapp] = gen_iapp(stimulus2,t_max,dt,lambda,io,tou, tou_s,m,post_w);
        [post_V,x] = aef(post_iapp);
        w2 = post_w;
        V2 = post_V;
        if max(post_V) ~= 0
            break
        end

        itr = itr +1;
    end
    [pre_iapp_c] = gen_iapp(stimulus1,t_max,dt,lambda,io,tou, tou_s,m,post_w);
    [V1_c,x1] = aef(pre_iapp_c);
    if max(V1_c) >= 0
        break
    end
    V = V1_c;
    w = post_w;
    while(1)
        [post_w,del_tk] = training2(w,dt,stimulus1,V);
        [post_iapp] = gen_iapp(stimulus,t_max,dt,lambda,io,tou, tou_s,m,post_w);
        [post_V,x] = aef(post_iapp);
        del_w = post_w - pre_w;
        del_w = nonzeros(del_w);
        if max(post_V) == 0
            break
        end
        w = post_w;
        V = post_V;
        itr = itr +1
end
w2 = post_w;

end
w2_partb =  post_w;
V2_partb =  post_V;
figure(3)
plot(x1,V1(:,1))
title(["Part b S1"])
xlabel('Time(in s)')
ylabel('Voltage (in V)')
figure(4)
plot(x2,post_V(:,1))
title(["Part b S2"])
xlabel('Time(in s)')
ylabel('Voltage (in V)')

  %%%%%part c

figure(5)
plot(x1,V1_c(:,1))
```

```
title(["Part c S1"])
xlabel('Time(in s)')
ylabel('Voltage (in V)')
     y = [1:100] ;
    figure(6)
    plot(y,w2_partb )
    title(["Weights part b "])

    xlabel('Weights')
    ylabel('Value of weights')



    %%%%%partd

itr = 1;
pre_V = V1_d;
pre_w = w1_d;
while(1)
    while(1)
        [post_w ,del_tk] = training(w1_d,dt,stimulus1,V1_d);
        [post_iapp] = gen_iapp(stimulus1,t_max,dt,lambda,io,tou, tou_s,m,post_w);
        [post_V,x] = aef(post_iapp);
        w2 = post_w;
        V2 = post_V;
        if max(post_V) ~= 0
            break
        end

        itr = itr +1;
    end
    [pre_iapp_c] = gen_iapp(stimulus2,t_max,dt,lambda,io,tou, tou_s,m,post_w);
    [V2_c,x1] = aef(pre_iapp_c);
    if max(V2_c) >= 0
        break
    end
    V = V2_c;
    w = post_w;
    while(1)
        [post_w,del_tk] = training2(w,dt,stimulus2,V);
        [post_iapp] = gen_iapp(stimulus,t_max,dt,lambda,io,tou, tou_s,m,post_w);
        [post_V,x] = aef(post_iapp);
        del_w = post_w - pre_w;
        del_w = nonzeros(del_w);
        if max(post_V) == 0
            break
        end
        w = post_w;
        V = post_V;
        itr = itr +1
end
w2 = post_w;
```

```
end
w2_partd =  post_w;
V2_partd=  post_V;

figure(7)
plot(x1,post_V(:,1))
title(["Part d S1"])
xlabel('Time(in s)')
ylabel('Voltage (in V)')
figure(8)
plot(x2,V2_d(:,1))
title(["Part d S2"])
xlabel('Time(in s)')
ylabel('Voltage (in V)')


    figure(9)
    plot(y,w2_partd )
    title(["Weights part d "])

    xlabel('Weights')
    ylabel('Value of weights')
```