

```

%Adaptive signal Equalizer.
%G.Anjan Kumar , 2020PEB5360.
%contents:
%Matlab code.
%Figures.
%Conclusions.

%-----%
clear
clc
%Adaptive FIR equalizer using LMS algorithm

%using matlab inbuilt functions
bpsk=comm.BPSKModulator;

%input samples of [-1,1] randomly generated.
xn = bpsk(randi([0 1],1000,1));
%index= (0:1000);
figure(1)

%plotting discrete graph of bpsk signal values are either -1,1.
%stem(index,xn);
stem(xn);
title('Input bpsk signal -1,1')

%convolution of input symbols with the system.
y = conv(xn,[1 0.8 0.3]);%  $H(z)=1+0.8z^{-1}+0.3z^{-2}$ .

%plotting discrete values of 'y'.
figure(2)
stem(y);
title('Output from the filter H(z)')

%'y' is combined with random noise of 30dB.
y1 = awgn(y,30);

%plotting discrete values of 'y1' after addition of noise.
figure(3)
stem(y1);
title('Signal after awgn noise addition')
%equalizing

eq=
comm.LinearEqualizer("Algorithm","LMS","NumTaps",8,"StepSize",0.04,"InputDelay",0,
"InputSamplesPerSymbol",1,"TrainingFlagInputPort",false,"AdaptAfterTraining",true,
"WeightUpdatePeriod",1);

```

```

%desired signal is considered with zero delay of input signal.
%i.e desired signal dn=xn.
[y2,e,weights] = eq(y1,xn(1:1000));

%plotting error
figure(4)
plot((1:1002),e)
grid on;
xlabel('symbols'); ylabel('e'); title('Error signal');

%plotting Absolute error
figure(5)
plot(abs(e))
grid on; xlabel('Symbols'); ylabel('|e|');title('Equalizer Error Signal')

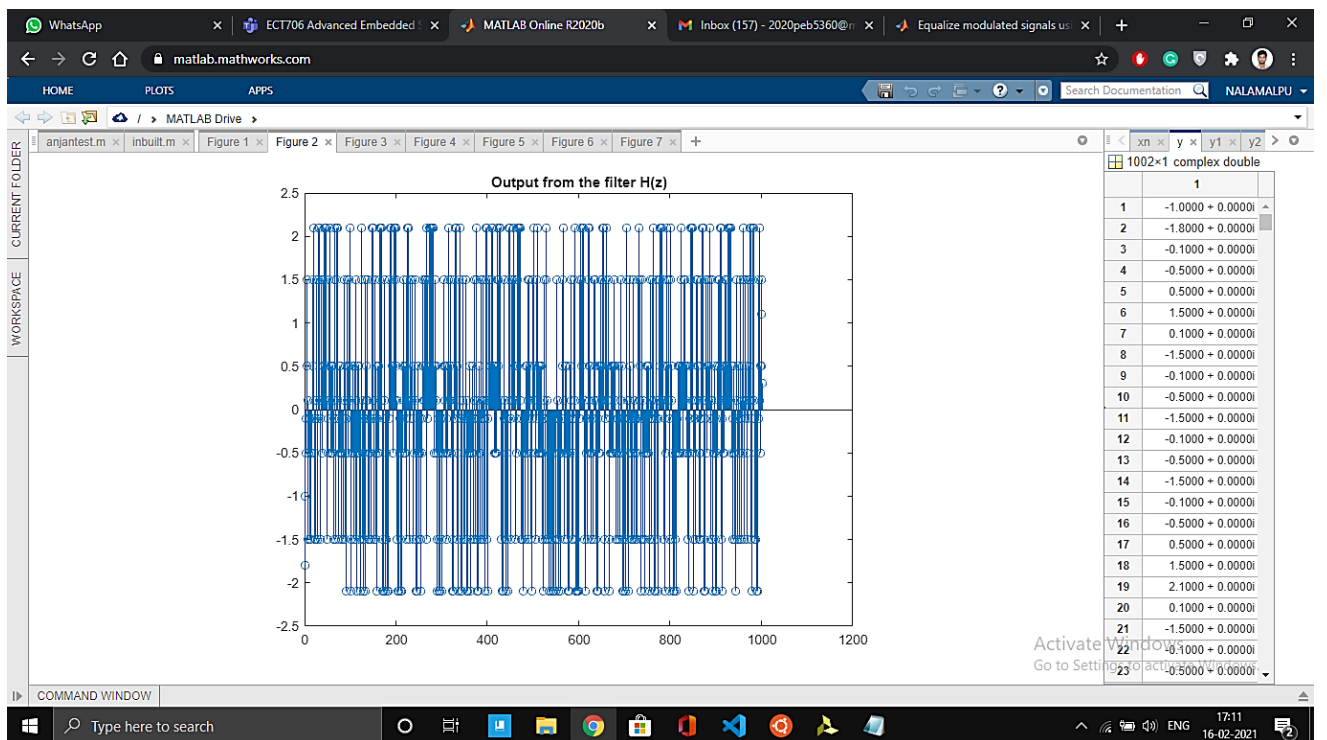
k=abs(e);

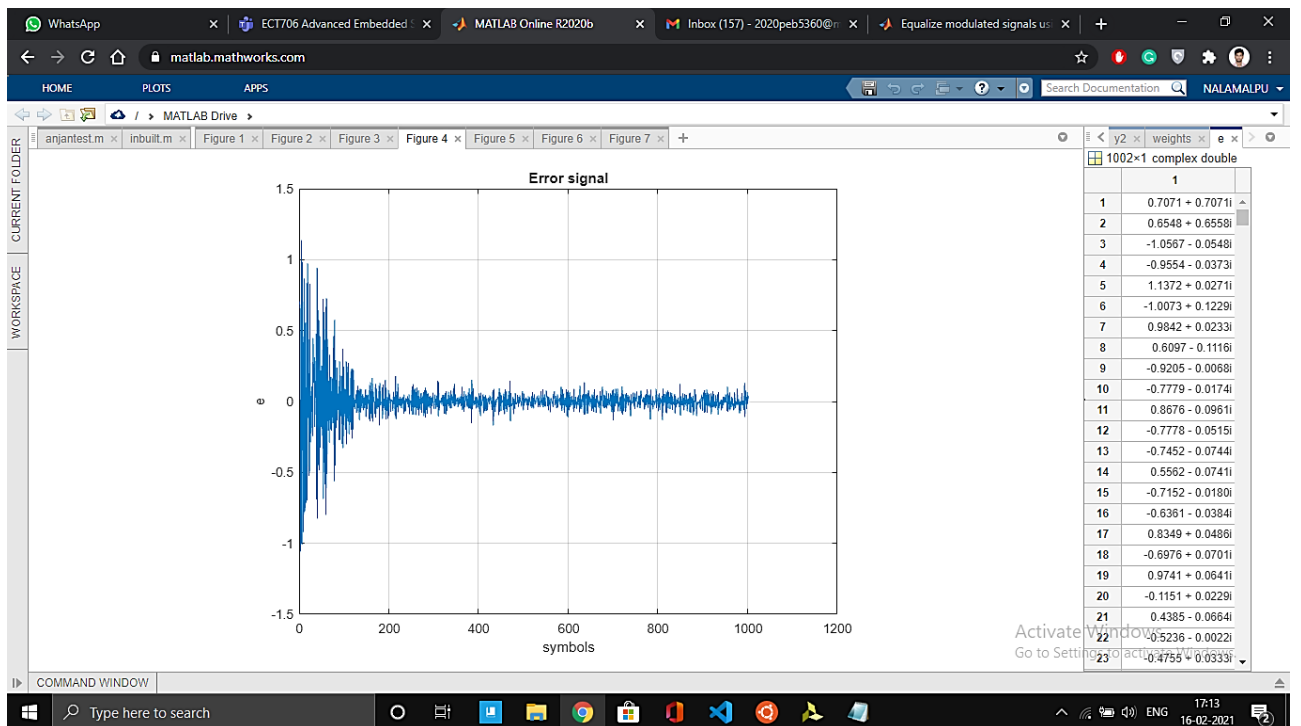
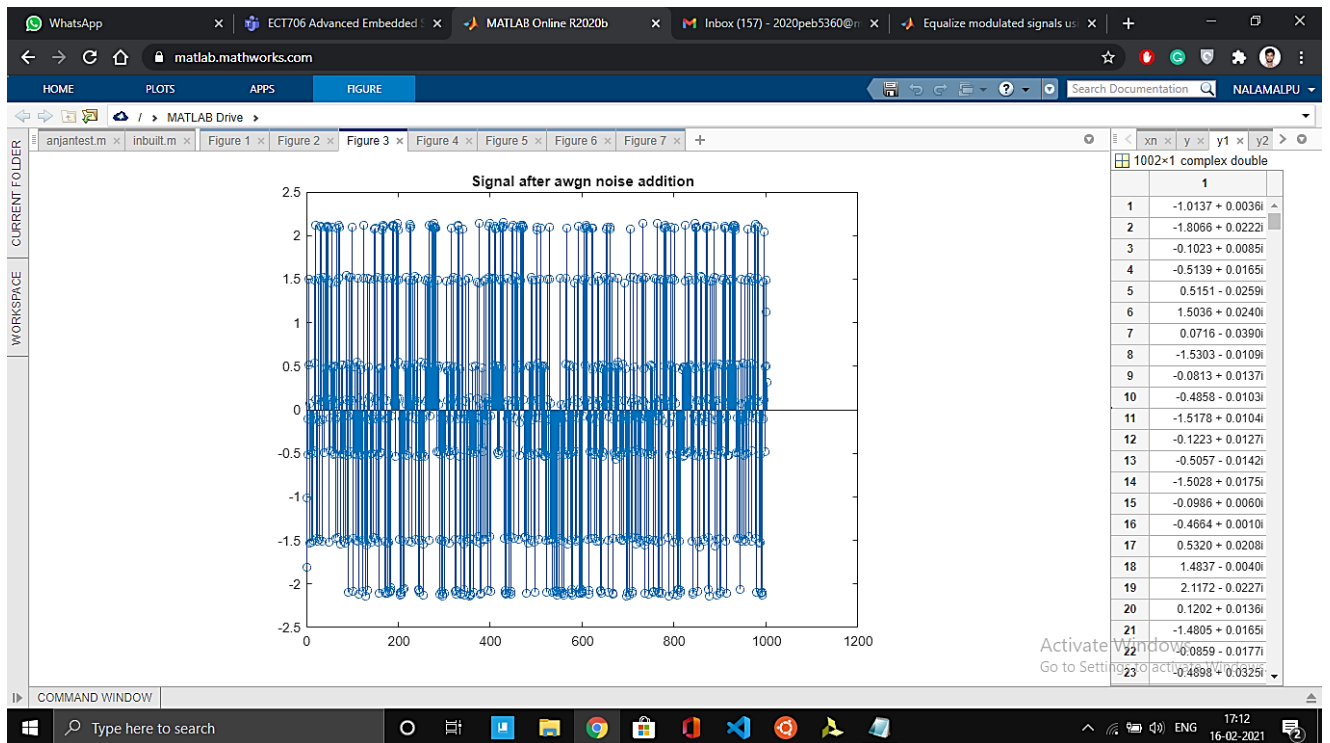
%plotting square of the error signal
figure(6)
plot(k.^2)
xlabel('symbols'); ylabel('e^2'); title('Mean square Error signal');

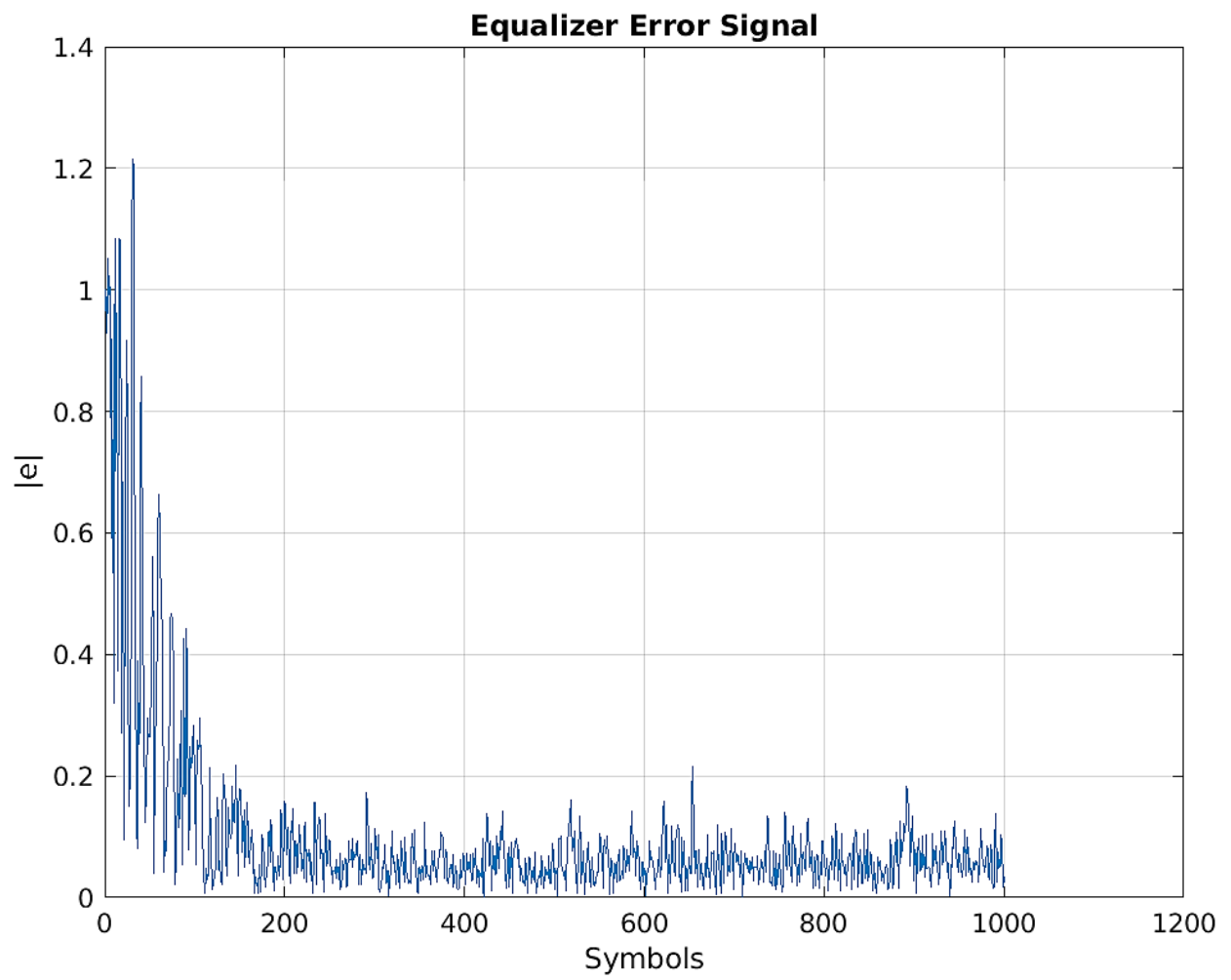
%plotting weights
figure(7)
subplot(3,1,1);
stem(real(weights)); ylabel('real(weights)'); xlabel('Tap'); grid on; axis([0 8 -1
1])
title('Equalizer Tap Weights')
subplot(3,1,2);
stem(imag(weights)); ylabel('imag(weights)'); xlabel('Tap'); grid on; axis([0 8 -
0.01 0.01])
subplot(3,1,3);
stem(abs(weights)); ylabel('abs(weights)'); xlabel('Tap'); grid on; axis([0 8 -1
1])

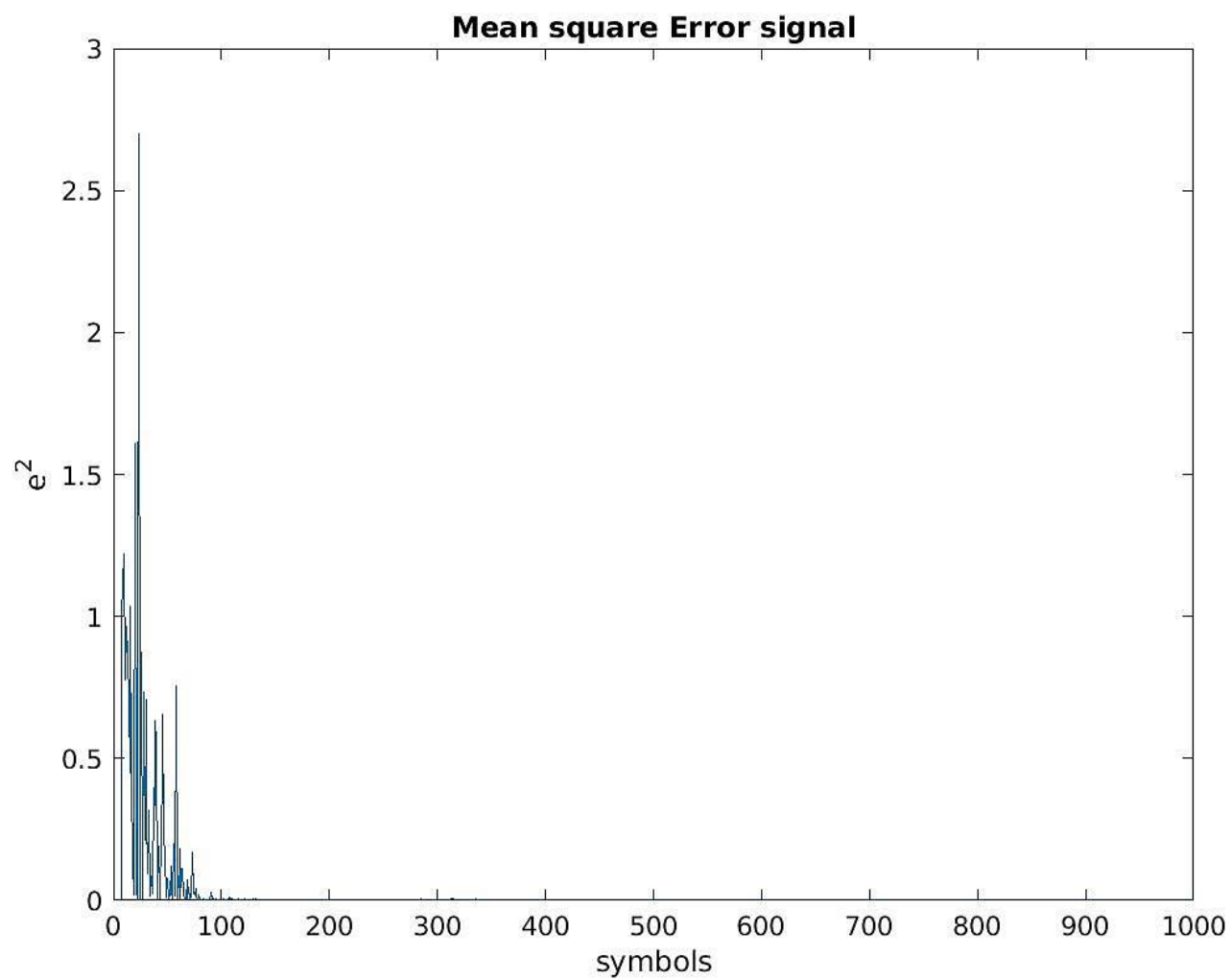
%-----%

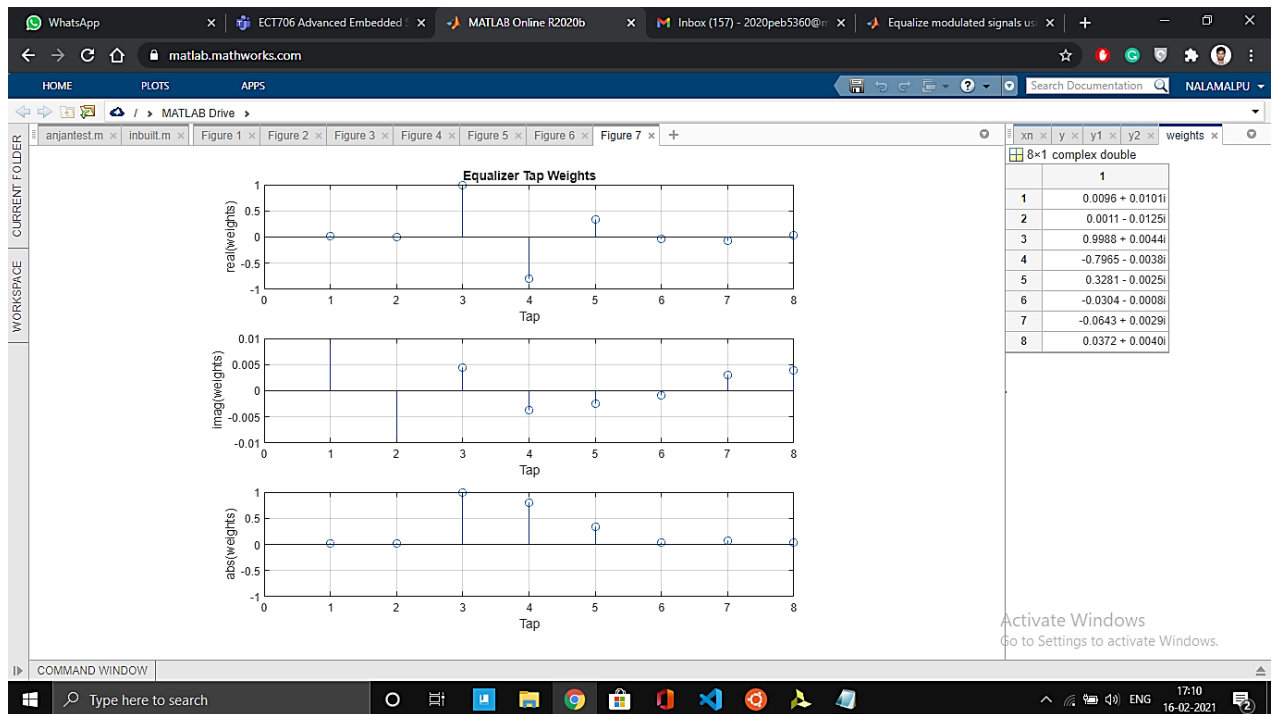
```











Conclusions:- I used LMS algorithm to design a system equalizer. I have designed using the matlab libraries. Here I have considered the channel coefficients as  $H(z) = [1 \ 0.8 \ 0.3]$ . I have considered number of weights in the adaptive to be 8 (more than twice of channel coefficients for better converging), so the system converges to zero. I have also varied the step sizes and saw how the error is converging to zero.