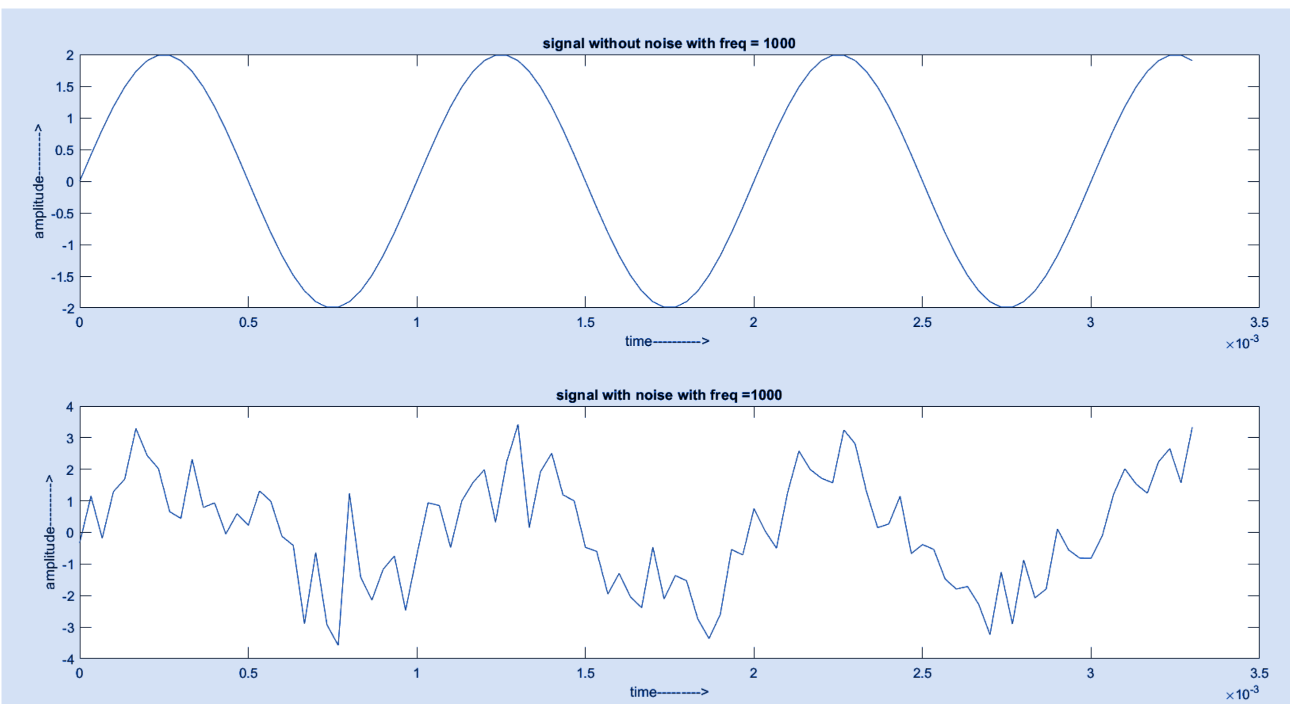
Digital Signal Processing Lab
Experiment -6

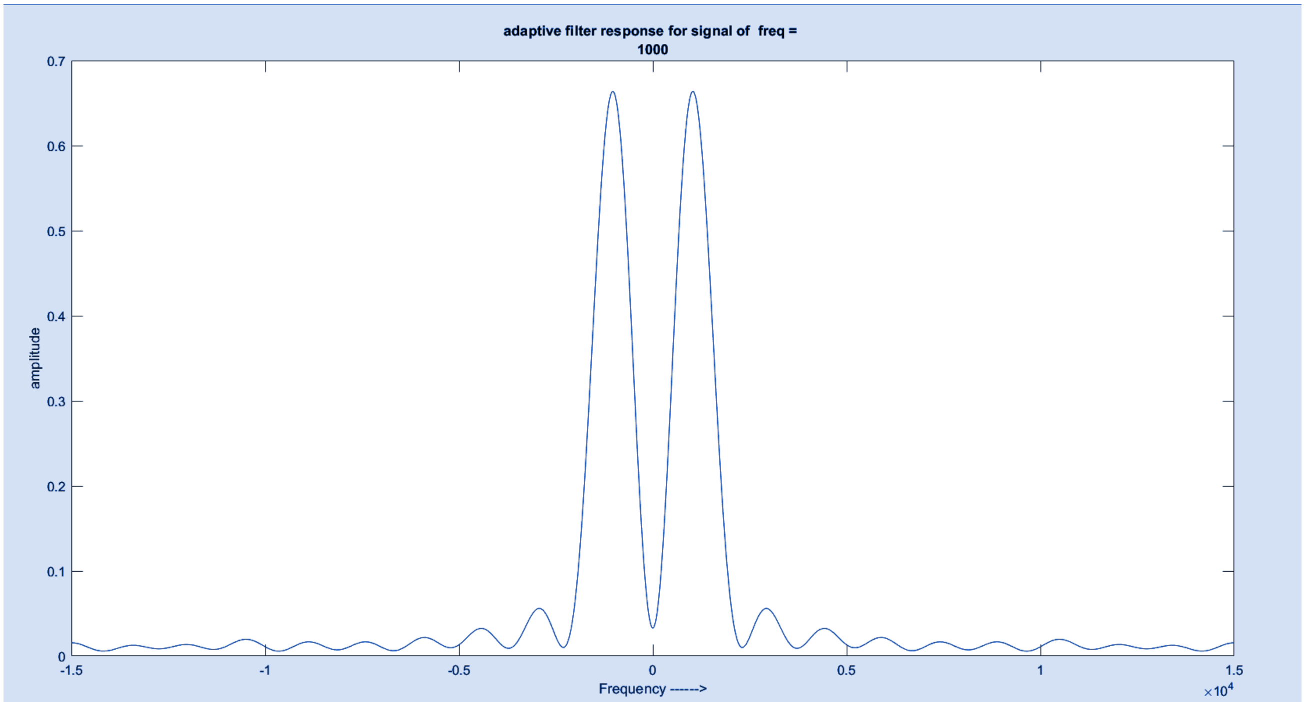
Name :Harshavardhan Alimi
Roll No.:18EC10021

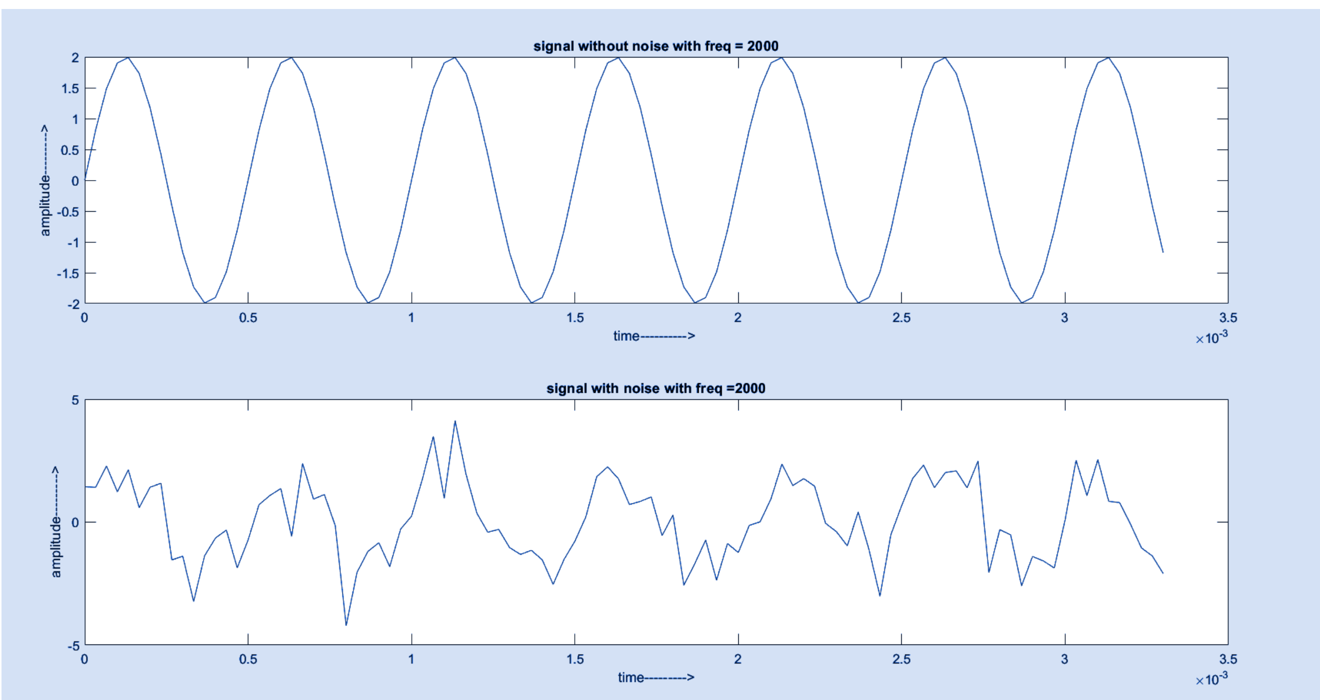
Adaptive Line Enhancer

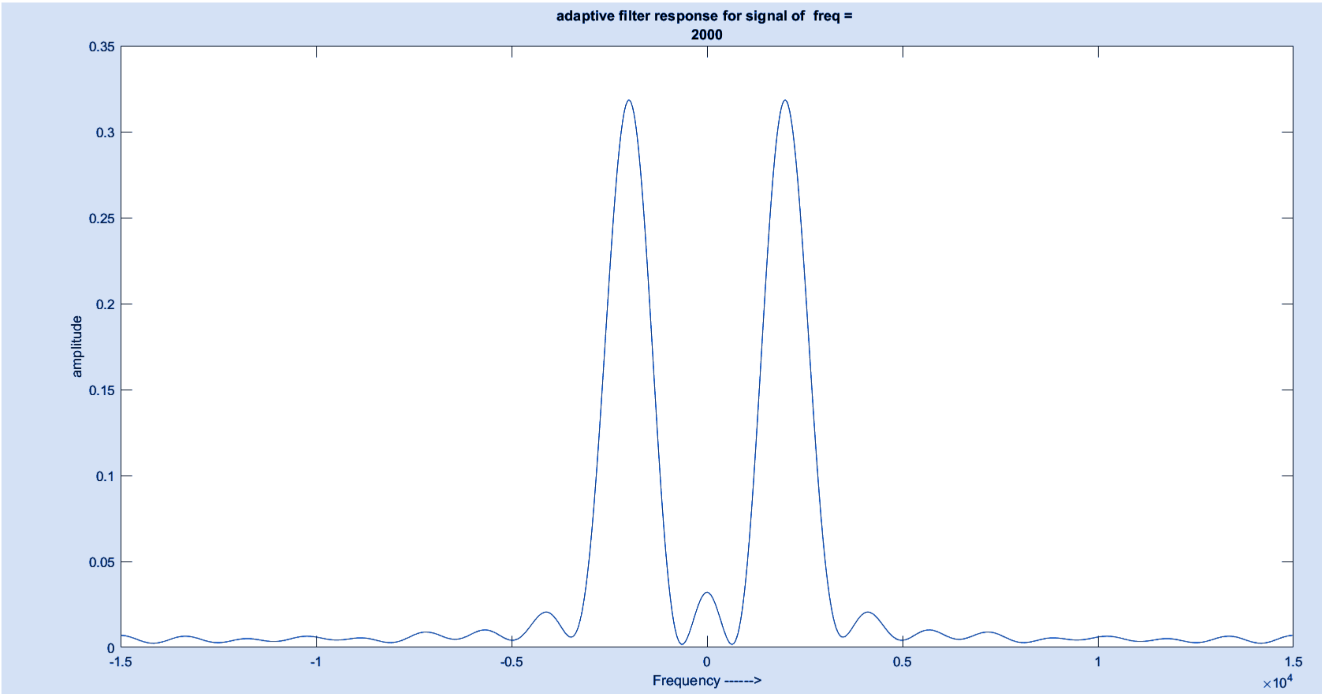
Aim: Design an Adaptive Line Enhancer to reduce the gaussian Noise

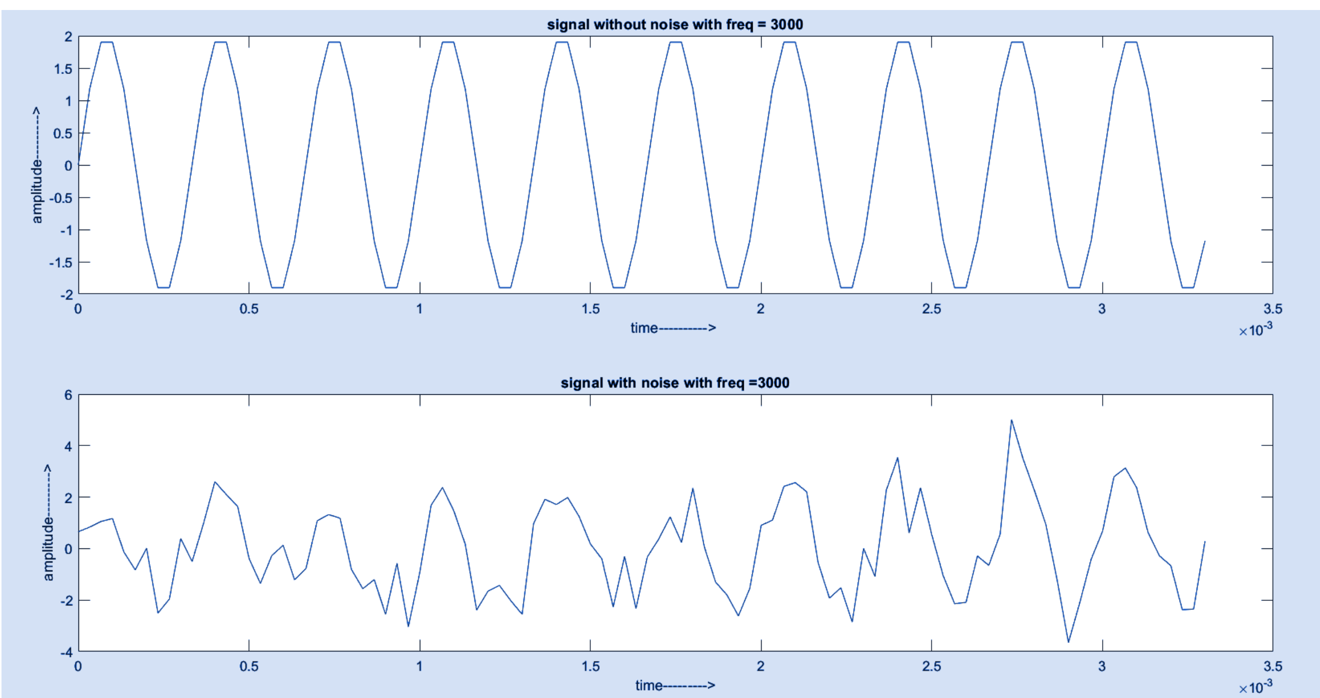
Results/Plots:-

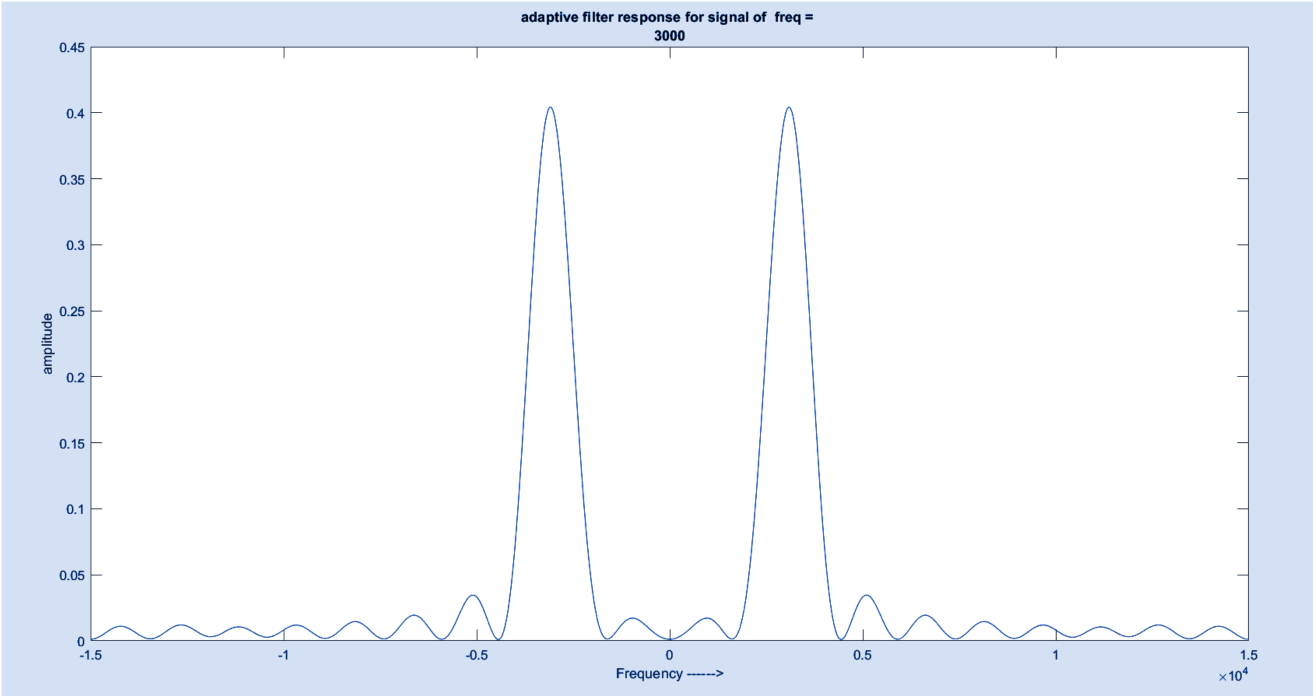


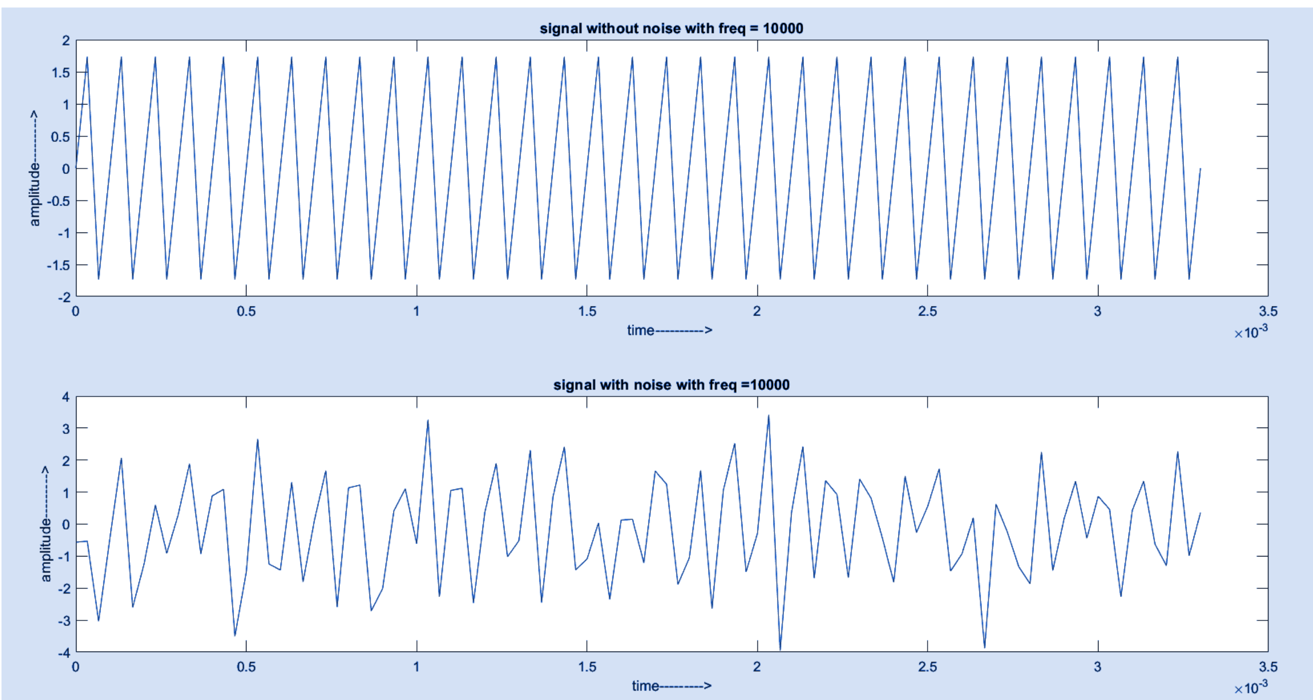

adaptive filter response for signal of freq =
1000
amplitude
Frequency ------>
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1.5
-1
-0.5
0
0.5
1
1.5
×10^4

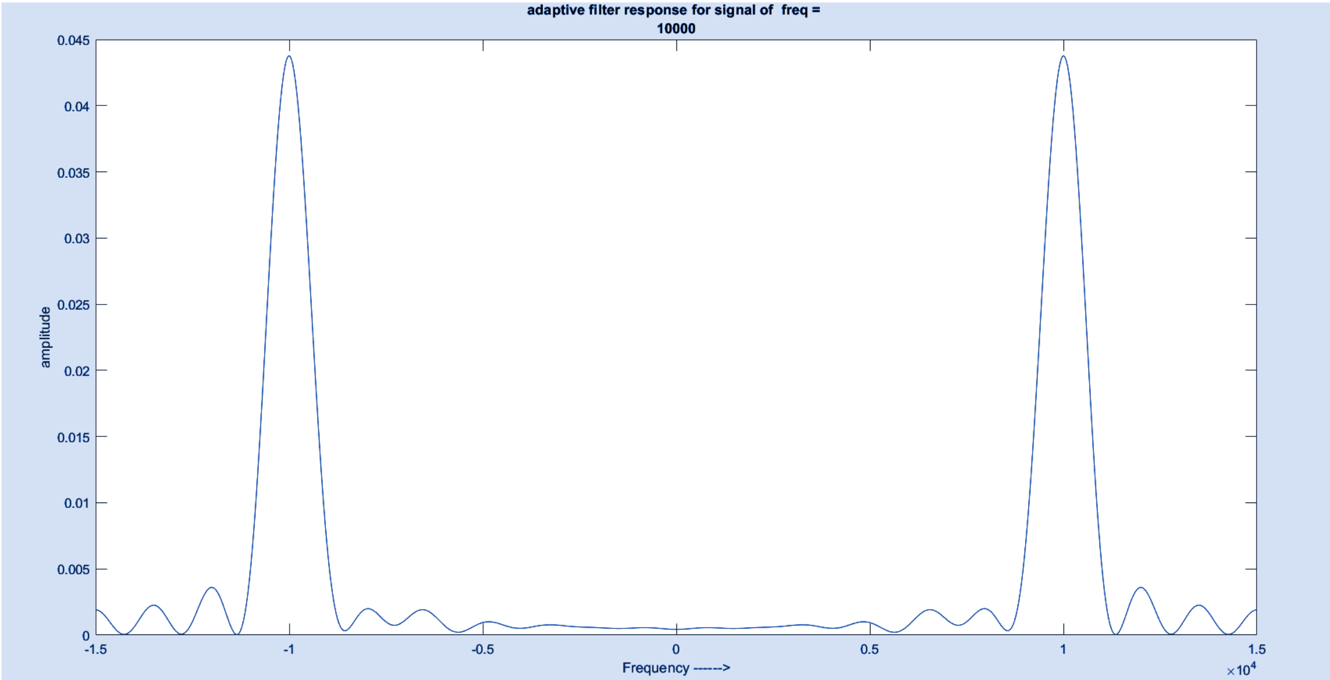

signal without noise with freq = 2000
amplitude--------->
time---------->
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
0.5
1
1.5
2
2.5
3
3.5
×10^-3
signal with noise with freq =2000
amplitude--------->
time---------->
5
0
-5
0
0.5
1
1.5
2
2.5
3
3.5
×10^-3


adaptive filter response for signal of freq =
2000
amplitude
Frequency ------>
×10^4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-1.5
-1
-0.5
0
0.5
1
1.5


signal without noise with freq = 3000
amplitude--------->
time---------->
×10^-3
signal with noise with freq =3000
amplitude--------->
time--------->
×10^-3


adaptive filter response for signal of freq =
3000
amplitude
Frequency ------>
×10^4


signal without noise with freq = 10000
amplitude-------->
time---------->
×10^-3
signal with noise with freq =10000
amplitude-------->
time--------->
×10^-3



Discussion & Observations:

- In this experiment we have designed an adaptive line enhancer to reduce the gaussian noise in a signal.
- The adaptive line enhancer has been realised using the concept of adaptive filter.(An adaptive filter is a digital filter that has self-adjusting characteristics. It is capable of adjusting its filter coefficients automatically to adapt the input signal via an adaptive algorithm.)
- The algorithm we use to adjust(change) the filter coefficients are called the adaptive algorithm. In this experiment we have used the LMS algorithm.
- In this experiment, we have desired that the output of the filter is the same as the input.

- In this algorithm , the base condition to end the loop(steady state is achieved) when the relative change becomes less than maximum change.
- From plots we can observe that the output signal is nearly the same as the input with reduction in noise which is to be achieved in the experiment.
- We can observe that the filter coefficients also depends on the value of frequency of the input signal
- We usually use this algorithm for the signals with low frequency components , because for the signals with higher frequency requires the higher sampling frequency .
- So, In this experiment , when we change the frequency of the input we need to take about the sampling frequency of the signal too(nyquist rate is satisfied)
- We can conclude that the adaptive line enhancer is very useful in the communication networks because the signal received at the receiver will be of very low strength and it is combined with noise . to detect the signal, we can use the above mentioned algorithm.

Appendix

Code :

```
N =1024;
Fs =30000;                                  %sampling frequency
A =2;
u = 10^-4;
e_max = 10^-5;                             %max relative change


for F = [1000,2000,3000,10000]             %different signal frequencies
      t = 0:1/Fs:(N-1)/Fs;
```

```
m = A*sin(2*pi*F*t);                          %sinusoidal signal
n = normrnd(0,1,1,N);                         %Noise signal
v = var(n);
n = n/sqrt(v);

x = m+n;                                      %input signal with noise

figure()
subplot(211)
plot(t(1:100),m(1:100));               %input signal plot without noise
xlabel('time----------> ')
ylabel('amplitude--------->')
title(['signal without noise with freq = ',num2str(F)]);

subplot(212)
plot(t(1:100),x(1:100));                %input signal plot with noise
xlabel('time--------->')
ylabel('amplitude--------->')
title(['signal with noise with freq =',num2str(F)]);




rchange = 1;
M =20;
w = zeros(M,1);                         %filter coefficient initialization

x1 = buffer(x,M,M-1);                  %dividing x into vectors of length M
x1 = flip(x1,1);

i =1;
while (rchange>e_max)
    y = x1(:,i)'*w;                               %filter output
    e = x(1,i)-y;                               % error signal(n)
    w_new = w + u*x1(:,i)*e;                   %updated filter coefficients
    if(i>1)
        rchange = (rssq(w_new(1,:)-w(1,:)))/(rssq(w(1,:)))^2;
                                                         %relative change
```

```
        end
        w = w_new;

        i =i+1;
    end

    f = -Fs/2 : Fs/N : (N-1)*Fs/(2*N);                %frequency points

    figure()
    plot(f,(abs(fftshift(fft(w,N)))).^2);             %plotting the spectrum
    xlabel('Frequency ------>');
    ylabel("amplitude");
    title(["adaptive filter response for signal of  freq = ",num2str(F)]);
end
```