Dept. of Electronics and Electrical Communication Engineering   
Indian Institute of Technology Kharagpur

**DIGITAL SIGNAL PROCESSING  
LAB (EC39201)**



**Experiment No: 6**

**Title:** Adaptive Line Enhancer

**Date of submission: 20th October, 2023**

Name: Gaurav Milind Kulkarni

Roll Number: 21EC30021

Group number: 39

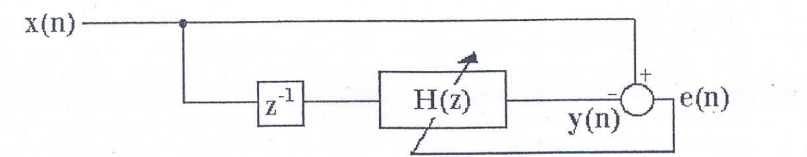
**Aim:**Take a sinusoidal message signal with noise and pass it through the adaptive line enhancer system and obtain the transfer function plot of of the filter. Compare the plots for different frequencies like 2, 3, 10 kHz.

**Theory:**An Adaptive Line Enhancer (ALE) is a digital signal processing technique used to enhance or extract a specific signal from a noisy environment, particularly when the frequency of the desired signal is unknown. It's a powerful tool in applications like communications, radar, and speech processing.

Basic Principle:  
1. Filtering and Adaptation: ALEs are based on the concept of adaptive filtering. An adaptive filter adjusts its parameters based on the input data to minimize the error between the desired output and the actual output.  
2. LMS Algorithm: The Least Mean Squares (LMS) algorithm is commonly used in ALE. It's a type of gradient descent algorithm that iteratively adjusts filter coefficients to minimize the mean square error between the filter output and the desired signal.

Components of an ALE:  
1. Input Signal (x[n]): The input signal to the ALE is a mixture of the desired signal and noise. It's represented as x[n], where 'n' is the discrete time index.  
2. Adaptive Filter: The core of the ALE is an adaptive filter. This filter has adjustable coefficients that are updated iteratively to minimize the error. The filter is represented by the transfer function H(z).  
3. Reference Input (d[n]): This is the reference signal, which is assumed to contain only the noise. In practice, it's obtained by passing the input signal through a delay element.  
4. Error Signal (e[n]): The error signal represents the discrepancy between the actual output of the filter and the reference input. It's calculated as e[n] = x[n] - d[n].

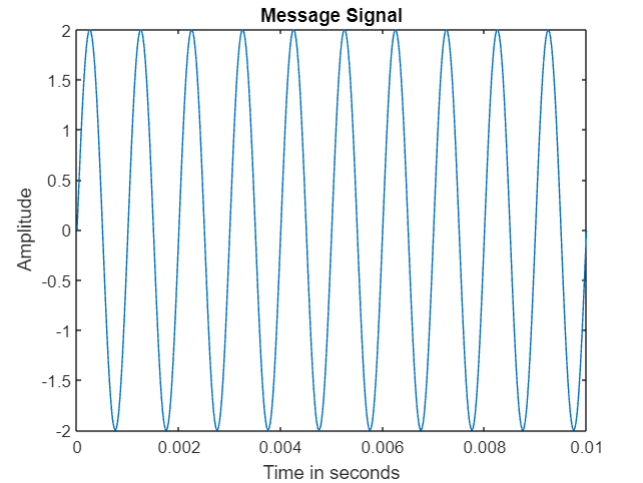
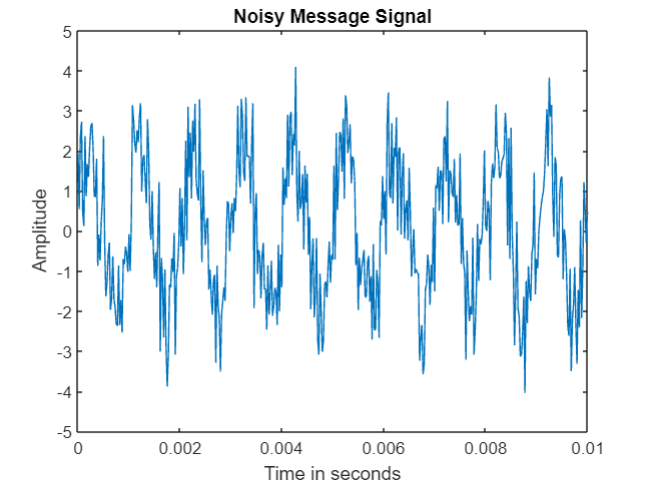
Operation of an ALE:  
1. Initialization: Initially, the filter coefficients are set to arbitrary values.  
2. Adaptation Process: The LMS algorithm updates the filter coefficients iteratively to minimize the mean square error. This is done using the formula:  
 W(k+1) = W(k) + μ \* e(n) \* X(n)  
where W(k) is the filter coefficient vector at iteration 'k', μ is the step size (a small positive constant), e(n) is the error signal, and X(n) is the input signal vector.  
3. Filtered Output (y[n]): The filter processes the input signal, producing an output y[n] = H(z) \* x[n], which ideally contains only the desired signal.  
4. Updated Reference (d[n]): The reference signal is continuously updated using the delayed version of the filtered output, which helps to adapt to changes in the input signal.

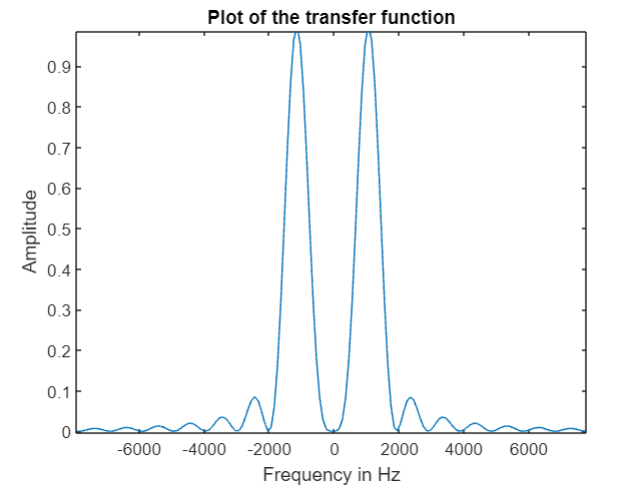


Schematic of the Adaptive Line Enhancer

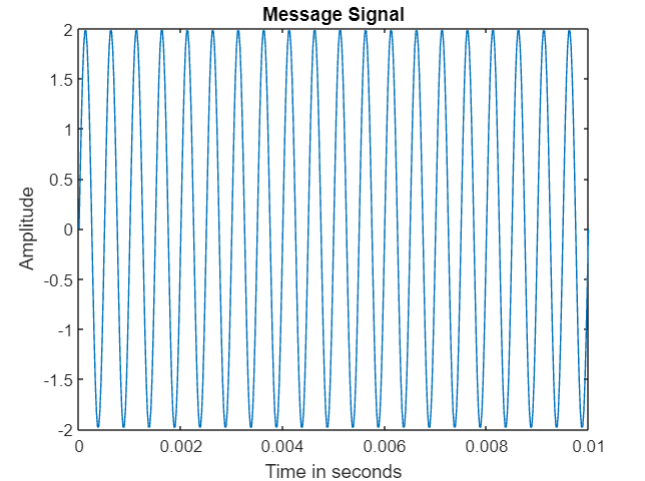
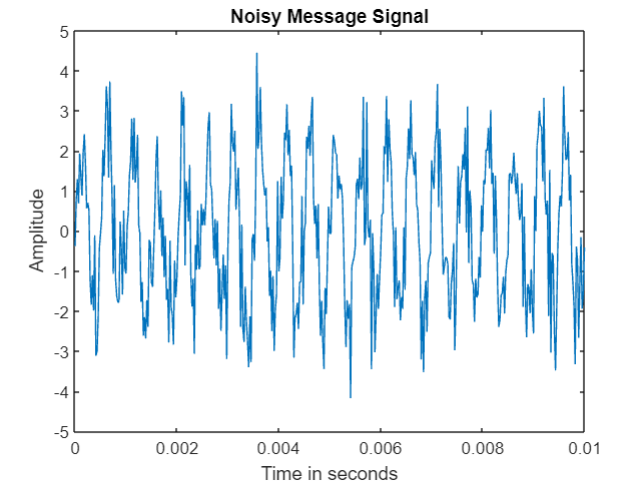
**Observations:**

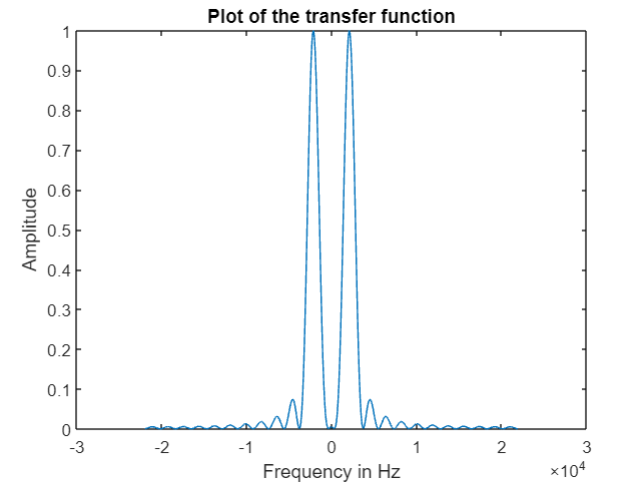
1) F­0 = 1000Hz   
 P = 45

** **

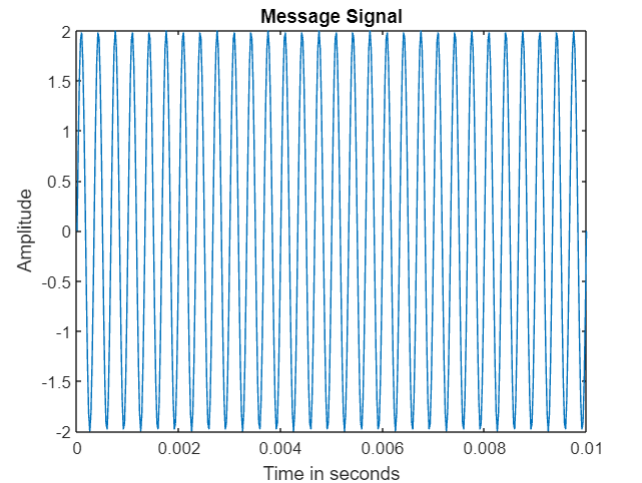
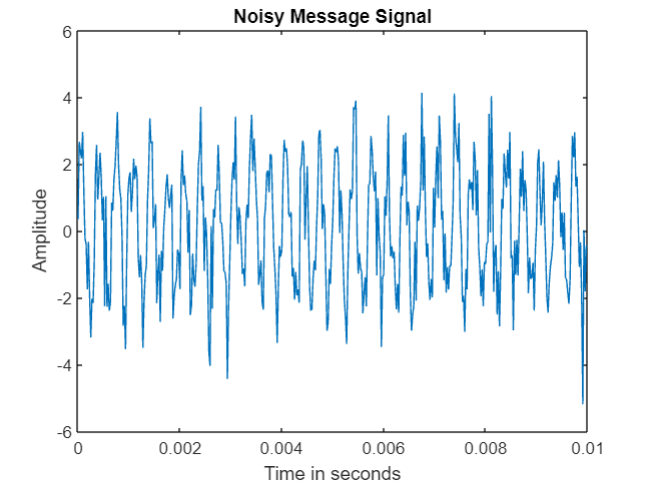
****

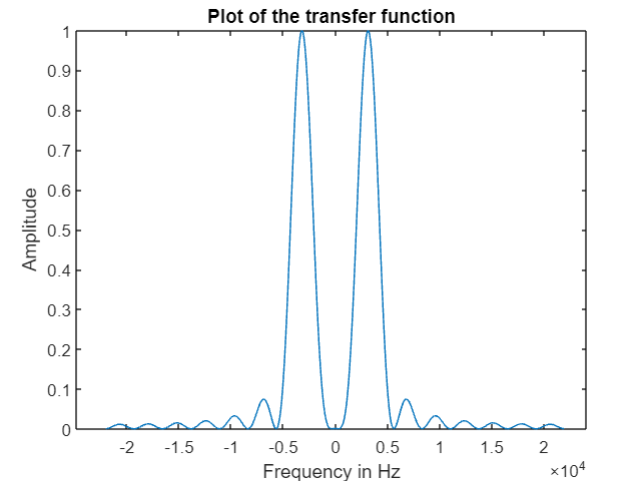
2) F0 = 2000Hz  
 P = 24

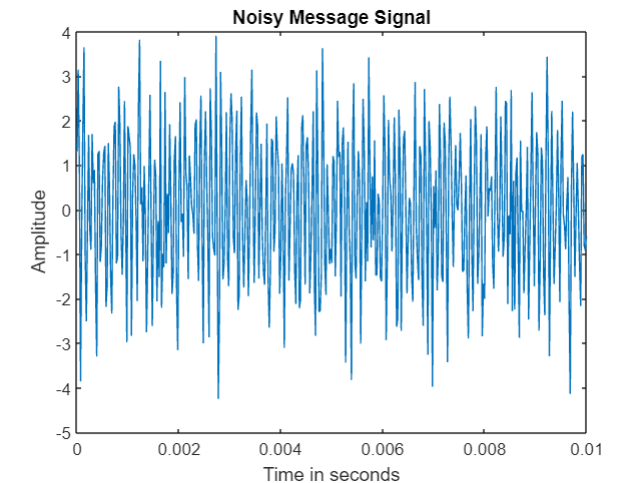
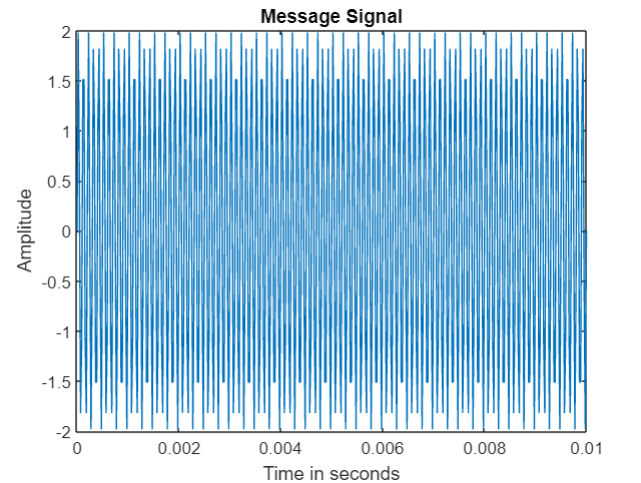


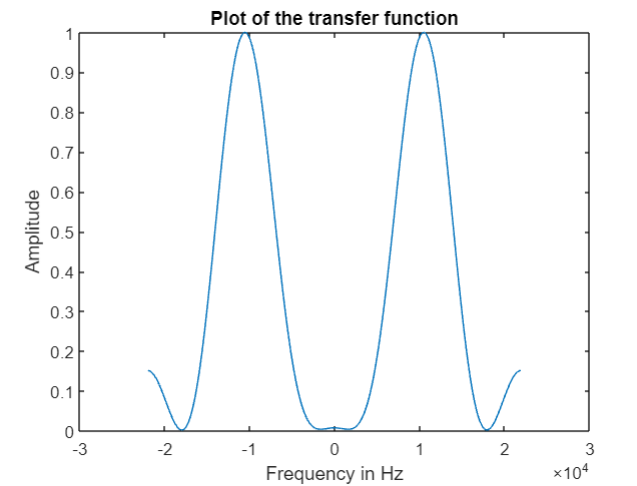
3) F0 = 3000kHz  
 P = 16



4) 10000kHz  
 P = 5





**Discussion:**

In the experiment, we implemented a adaptive line enhancer and were able to extract a sinusoidal wave from a noisy signal. Multiple observations were made for various values of frequency and accordingly, the parameter P had to be adjusted to get the required transfer function, where P is the number of coefficients in the transfer function.

It can be clearly observed that as the frequency signal to be extracted increases, the value of the parameter P decreases. The observation that as the frequency of the signal to be extracted increases, the number of coefficients P in the transfer function of an Adaptive Line Enhancer (ALE) decreases is likely related to the nature of the adaptive filtering process and the phenomenon of spectral sparsity. Spectral sparsity refers to the idea that in many real-world signals, the energy is concentrated at certain frequencies, while other frequencies contribute little to the overall signal. The number of coefficients P in the transfer function of an ALE determines the length of the adaptive filter. A longer filter can potentially capture more detailed spectral information. As the frequency of the signal to be extracted increases, the ALE must strike a balance between adapting to the changing spectral characteristics and avoiding overfitting to noise or irrelevant frequency component. At higher frequencies, due to the spectral sparsity, it's often more efficient to use a shorter filter with fewer coefficients. This allows the ALE to focus on extracting the energy in the relevant frequency range. The choice of P is often a trade-off between accurately capturing the signal's spectral characteristics and avoiding overfitting to noise

The advantages of using an ALE are as follows: ALE is particularly effective when the frequency of the desired signal is unknown, as it adapts to the signal characteristics. ALE can significantly attenuate noise. ALE can operate in real-time, making it suitable for applications where timely processing is crucial. ALE can adapt to variations in the input signal characteristics.

The limitations are: The convergence of the adaptive filter depends on the step size and the characteristics of the input signal. Choosing an appropriate step size is critical. If the step size is too large, the filter may over-adjust and distort the desired signal. ALEs can become unstable if the step size is not carefully chosen. The performance of the ALE may depend on the initial filter coefficients. In our experiment, we just set it to a very small value.

**Code:**clc;

clear all;

A = 2;

fm = 10000;

Fs = 44000;

t = 0:1/Fs:0.01;

message\_signal = A\*sin(2\*pi\*fm\*t);

noise = wgn(1, length(t), 0);

input\_signal = message\_signal + noise;

figure(1)

plot(t, message\_signal);

title("Message Signal");

xlabel("Time in seconds");

ylabel("Amplitude");

figure(2)

plot(t, input\_signal);

title("Noisy Message Signal");

xlabel("Time in seconds");

ylabel("Amplitude");

figure(3)

plot(t,noise);

p = 5;

N = length(t);

w = ones(1,p).\*0.0001;

message\_in = [];

for j = 1:N-p+1

for i = 1:p

message\_in(j, i) = message\_signal(j+i-1);

end

end

j = 2;

y = zeros(p);

epsilon = 10;

while epsilon > 1e-3

y = filter(w, 1, message\_in((j-1), :));

error\_signal = y - message\_in(j, :);

w\_ = w + 1e-4\*error\_signal(p)\*message\_in(j, :);

epsilon = (abs(norm(w\_ - w)))^2/norm(w)^2;

w = w\_;

j = j+1;

end

sys\_ = filt(w, 1)

[H, om] = freqz(w, 1, 'whole', length(t));

H\_ = (abs(H)).^2;

f\_scale = (-length(om)/2:1:length(om)/2-1)\*Fs/(3\*length(om));

figure

plot(f\_scale, abs(fftshift(H\_))/abs(max(H\_)));

title("Plot of the transfer function");

xlabel("Frequency in Hz");

ylabel("Amplitude");

% H\_fft = fftshift(fft(H\_));

% figure

% plot(f\_scale, abs(H\_fft)));