Abstract

This project aim to design adaptive filter operation that is used to filter the any given and reference signal with the help of LMS which mean least mean square values. In this project we have given that three question that depend one each other ,In the first question we have given that Adaptive filter case diagram that we need to create Adaptive filter of two signal one is reference signal and other is desired signal given that , Than we need to add Gaussian noise in desired signal and then observe the digital plotting case study at different value of sound to noise ratio.

INTRODUCTION

In this project we basically need to design the Adaptive filter designing, Actually we have given some signals in the form of Desired signal and Reference signal that we have already given that, Using Matlab processing we need to add filter in these signals and conclude the filter signal

Aim of Design

Designing aim of this project is quite simple firstly we need to design the Adaptive filter at given cases and we will add Gaussian noise at different values of SNR in it and analyze it clearly that what the different we observe than we will take a audio signal than implement the Adaptive signal on it.

Objective of Design

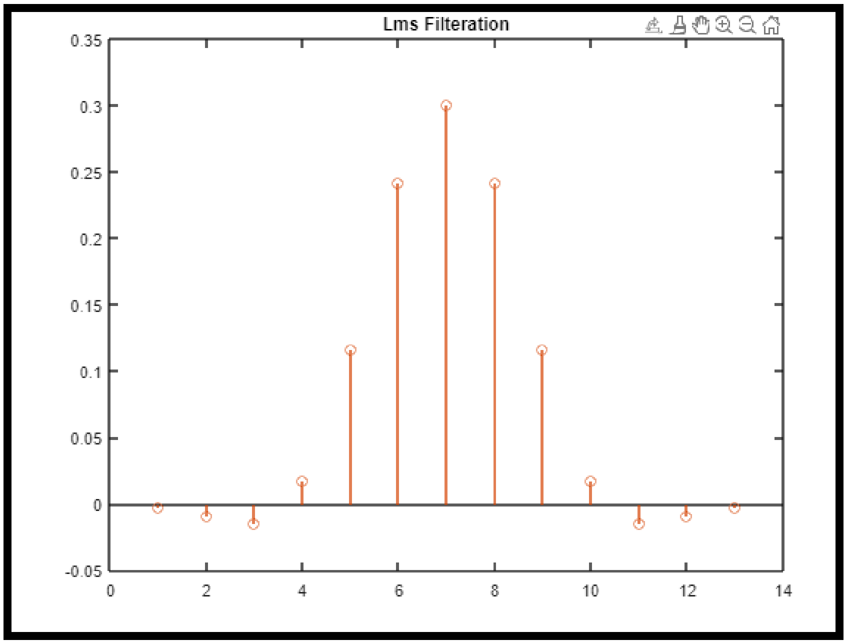
- To understand the Built in function of Matlab
- To analyze the filtration in Matlab
- Test Simulation

Question 1

Part a)

Implement Adaptive filter:

```
%%Question#1
%Part#1
%%=================================================================
%%Implement Adaptive Filter with size=0.4
Fs=1000;
Ts=1/Fs;
order=12;
t=0:Ts:1-Ts;
x=sin(2*pi*t/30);
noise=randn(size(x));
x=x+noise;
x=max(x)/x;
x=x';
b=fir1(order,0.3,'low');
filteraion=filter(b,1,x);
mu=0.4;
lms=dsp.LMSFilter(order+1,'StepSize',mu,'WeightsOutputPort',true);
[y,e,w]=step(lms,x,filteraion);
stem([b.' w]);
title('Lms Filteration');
```



Explanation:

In this task first we define some parameters like sampling frequency, size of filter and some given signals in the form of desired signal and reference signal. First we need to determine the sampling time to make the signal with respect to time, After that we add the signal in the code and manipulate it with respect to time range, Than add some noise in it to make it some noise but it not a Gaussian Noise, after that add the filtration of adaptive filter in the signal b , But code is not end here we need to measure the LMS so to measure it we used the digital processing library to implement it , it gives us Y,W and Errors values form least measurement value.

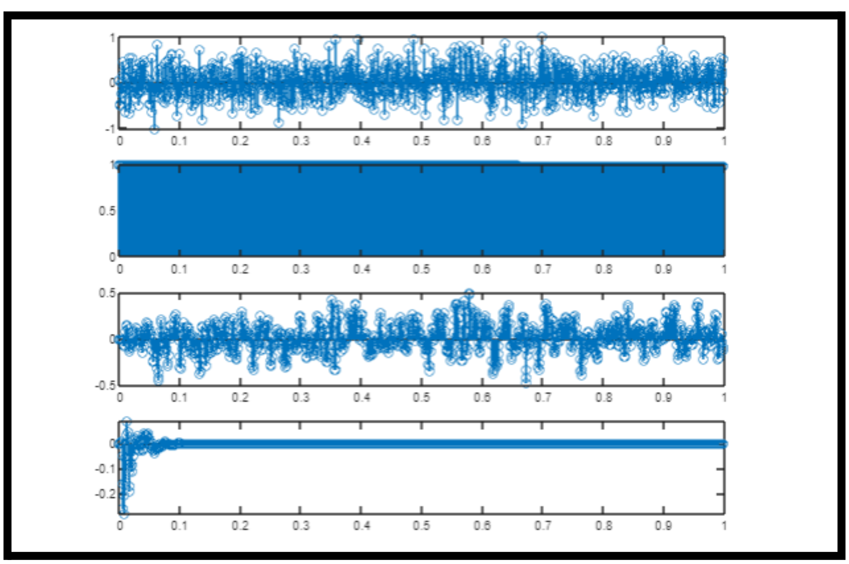
Part b)

Plotting the Parameters:

```
%Part 2
%%Plotting
%%Plot x[n]
figure(2)
subplot(4,1,1)
stem(t,x)
d=cos(2*pi*t/30);
subplot(4,1,2)
stem(t,d)
subplot(4,1,3)
stem(t,y)
subplot(4,1,4)
stem(t,e)
```

Explanation:

As filtration signal create in above with the parameters of Y,W and error values, but we need to plot graphs of them as shown below:



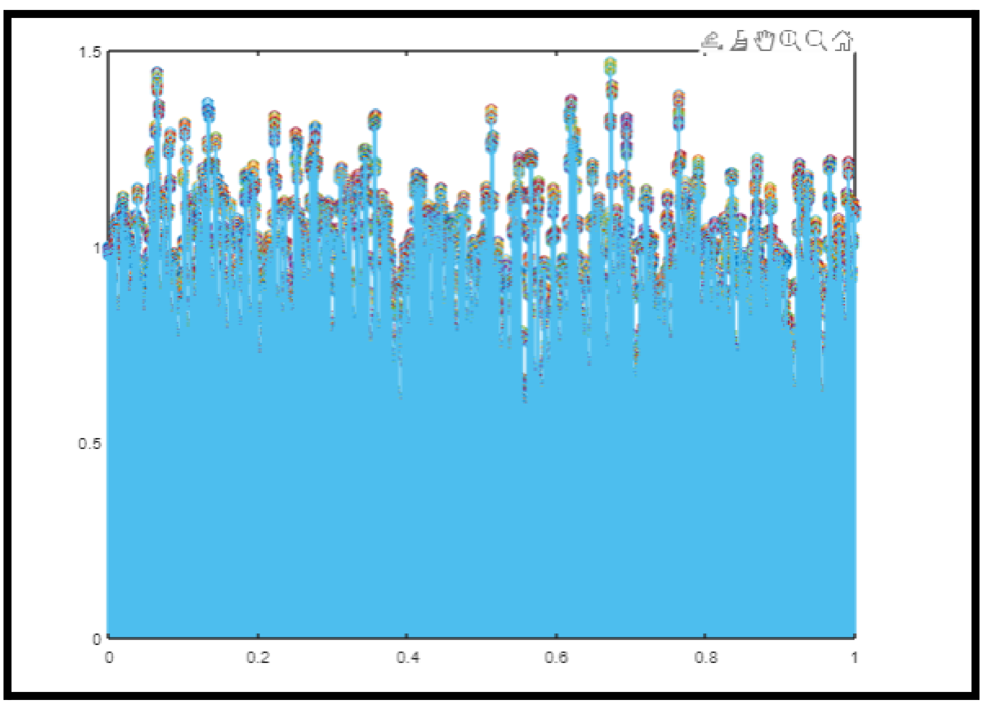
Part c)

i) Error Signal:

```
%Part 3
%% i) Error Signal
e=d-y;
disp('Error Signal is:')
disp(e);
figure(3)
stem(t,e)
```

Explanation:

In this case we need to measure the value of Error signal to measure the value of error signal we need two signal to analyze them one is desired and other is output signal, If output signal is desired signal than we will achieve our task as it is feedback function so it continuous work and give signal shown below:



ii) Mean Squared Error:

```
%% ii) Squared Error Calculate
E=0.02;
J=E*e*e;
```

17.2933	17.2926	17.2919	17.2912	17.2905	17.2898	17.2891	17.2884	17.2877	17.2870	17.2863	17.2856	17.2849	17.2842	17.2835							
19.1619	19.1612	19.1604	19.1596	19.1589	19.1581	19.1573	19.1566	19.1558	19.1550	19.1542	19.1535	19.1527	19.1519	19.1511	19.1504	19.1496	19.1488	19.1480	19.1473	19.1465	19.1457
19.8722	19.8714	19.8706	19.8698	19.8690	19.8682	19.8674	19.8666	19.8658	19.8650	19.8642	19.8634	19.8626	19.8618	19.8610	19.8602	19.8594	19.8586	19.8578	19.8570	19.8562	19.8554
19.8040	19.8032	19.8024	19.8016	19.8008	19.8000	19.7992	19.7984	19.7976	19.7968	19.7960	19.7952	19.7944	19.7936	19.7928	19.7920	19.7912	19.7904	19.7896	19.7888	19.7880	19.7872
20.5171	20.5163	20.5155	20.5146	20.5138	20.5130	20.5122	20.5114	20.5105	20.5097	20.5089	20.5081	20.5072	20.5064	20.5056	20.5047	20.5039	20.5031	20.5022	20.5014	20.5006	20.4997
22.2497	22.2488	22.2480	22.2471	22.2462	22.2453	22.2444	22.2435	22.2426	22.2417	22.2408	22.2399	22.2390	22.2381	22.2372	22.2363	22.2354	22.2345	22.2336	22.2327	22.2318	22.2309
23.4771	23.4761	23.4752	23.4742	23.4733	23.4724	23.4714	23.4705	23.4695	23.4686	23.4677	23.4667	23.4658	23.4648	23.4639	23.4629	23.4620	23.4610	23.4600	23.4591	23.4581	23.4572
22.9335	22.9326	22.9317	22.9307	22.9298	22.9289	22.9280	22.9271	22.9261	22.9252	22.9243	22.9234	22.9224	22.9215	22.9206	22.9197	22.9187	22.9178	22.9169	22.9159	22.9150	22.9141
21.2672	21.2664	21.2655	21.2647	21.2638	21.2629	21.2621	21.2612	21.2604	21.2595	21.2587	21.2578	21.2570	21.2561	21.2552	21.2544	21.2535	21.2527	21.2518	21.2509	21.2501	21.2492
19.8881	19.8873	19.8865	19.8857	19.8850	19.8842	19.8834	19.8826	19.8818	19.8810	19.8802	19.8794	19.8786	19.8777	19.8769	19.8761	19.8753	19.8745	19.8737	19.8729	19.8721	19.8713
19.0956	19.0949	19.0941	19.0934	19.0926	19.0918	19.0911	19.0903	19.0895	19.0888	19.0880	19.0872	19.0865	19.0857	19.0849	19.0841	19.0834	19.0826	19.0818	19.0810	19.0803	19.0795
18.3880	18.3873	18.3865	18.3858	18.3851	18.3843	18.3836	18.3829	18.3821	18.3814	18.3806	18.3799	18.3792	18.3784	18.3777	18.3769	18.3762	18.3754	18.3747	18.3739	18.3732	18.3724
17.6168	17.6161	17.6154	17.6147	17.6140	17.6133	17.6126	17.6119	17.6112	17.6105	17.6098	17.6090	17.6083	17.6076	17.6069	17.6062	17.6055	17.6048	17.6041	17.6033	17.6026	17.6019
17.2337	17.2330	17.2323	17.2316	17.2309	17.2302	17.2295	17.2288	17.2282	17.2275	17.2268	17.2261	17.2254	17.2247	17.2240	17.2233	17.2226	17.2219	17.2212	17.2205	17.2198	17.2191
17.4402	17.4395	17.4388	17.4381	17.4374	17.4367	17.4360	17.4353	17.4346	17.4339	17.4332	17.4325	17.4318	17.4311	17.4304	17.4297	17.4289	17.4282	17.4275	17.4268	17.4261	17.4254
17.9768	17.9760	17.9753	17.9746	17.9739	17.9732	17.9724	17.9717	17.9710	17.9703	17.9696	17.9688	17.9681	17.9674	17.9666	17.9659	17.9652	17.9645	17.9637	17.9630	17.9623	17.9615
18.2846	18.2839	18.2831	18.2824	18.2817	18.2809	18.2802	18.2795	18.2787	18.2780	18.2773	18.2765	18.2758	18.2750	18.2743	18.2736	18.2728	18.2721	18.2713	18.2706	18.2699	18.2691
17.9171	17.9164	17.9157	17.9150	17.9143	17.9136	17.9128	17.9121	17.9114	17.9107	17.9100	17.9092	17.9085	17.9078	17.9071	17.9063	17.9056	17.9049	17.9042	17.9034	17.9027	17.9020
16.8307	16.8300	16.8294	16.8287	16.8280	16.8273	16.8267	16.8260	16.8253	16.8246	16.8240	16.8233	16.8226	16.8219	16.8212	16.8206	16.8199	16.8192	16.8185	16.8178	16.8171	16.8164
15.6860	15.6854	15.6848	15.6841	15.6835	15.6829	15.6823	15.6816	15.6810	15.6804	15.6797	15.6791	15.6785	15.6778	15.6772	15.6766	15.6759	15.6753	15.6747	15.6740	15.6734	15.6727
15.4978	15.4972	15.4965	15.4959	15.4953	15.4947	15.4941	15.4934	15.4928	15.4922	15.4916	15.4909	15.4903	15.4897	15.4891	15.4884	15.4878	15.4872	15.4865	15.4859	15.4853	15.4847
16.5732	16.5726	16.5719	16.5712	16.5706	16.5699	16.5693	16.5686	16.5679	16.5673	16.5666	16.5659	16.5652	16.5646	16.5639	16.5632	16.5626	16.5619	16.5612	16.5605	16.5599	16.5592
18.2323	18.2315	18.2308	18.2301	18.2293	18.2286	18.2279	18.2271	18.2264	18.2257	18.2249	18.2242	18.2235	18.2227	18.2220	18.2213	18.2205	18.2198	18.2190	18.2183	18.2176	18.2168
19.9122	19.9114	19.9106	19.9098	19.9090	19.9082	19.9074	19.9066	19.9058	19.9050	19.9042	19.9034	19.9026	19.9018	19.9010	19.9002	19.8994	19.8986	19.8978	19.8970	19.8962	19.8954

Explanation:

In this task first we take the error signal from above part and then multiple it than we need to calculate the mean of the error signal that we calculate in the form of E and then multiple with E so in this way we calculate the Mean square errors.

iii) Derivation:

Question1 (iii)

$$J = E[e^2(n)]$$

take derivative with respect w1

$$\frac{dJ}{dw_1} = 2e(n)E+w_1$$

take derivative with respect

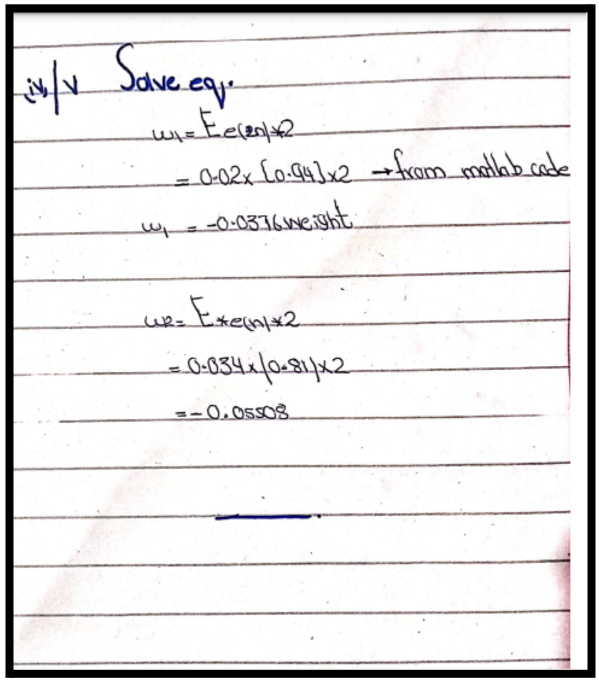
$$\frac{dJ}{dw_1} = 2e(n)E+w_2$$

equal to zero

$$2e(n)E+w_1=0 \quad -(1$$

$$2e(n)E+w_2=0 \quad -(2$$

iv) Solve for Weight and Minimum Error:


iv/v Solve eq.

$w_1 = E\,e(2n) * 2$

$= 0.02 \times [0.94] \times 2$ → from matlab code

$w_1 = -0.0376$ weight

$w_2 = E * e(n) * 2$

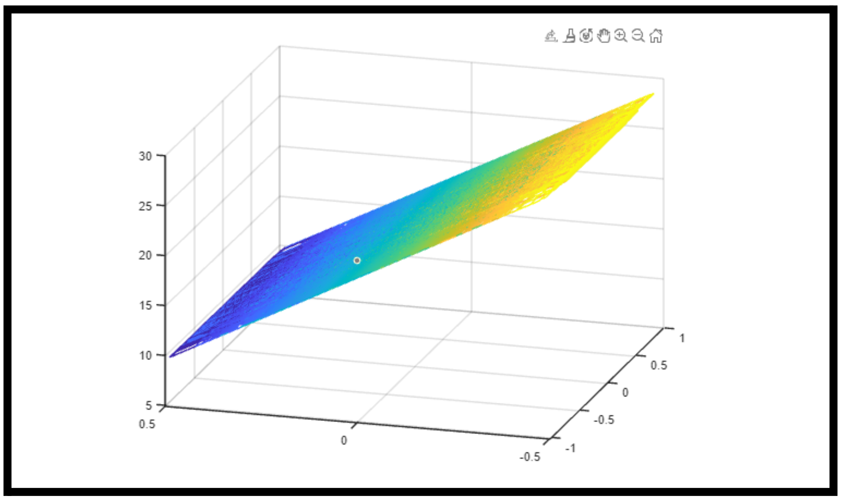
$= 0.034 \times |0.81| \times 2$

$= -0.05508$

Part d)

3D plot with J w0 and w1:

```
%%Part 4
%%3D plot
w0=x;
w1=y;
figure(4)
mesh(x,y,J);
```



Explanation:

As in above analysis we calculate the values of three side of 3d model one is J that is mean squared values calculated from error of the given system than other is w0 and w1 that is self generated from adaptive filter w so we design this 3d model from all three values:

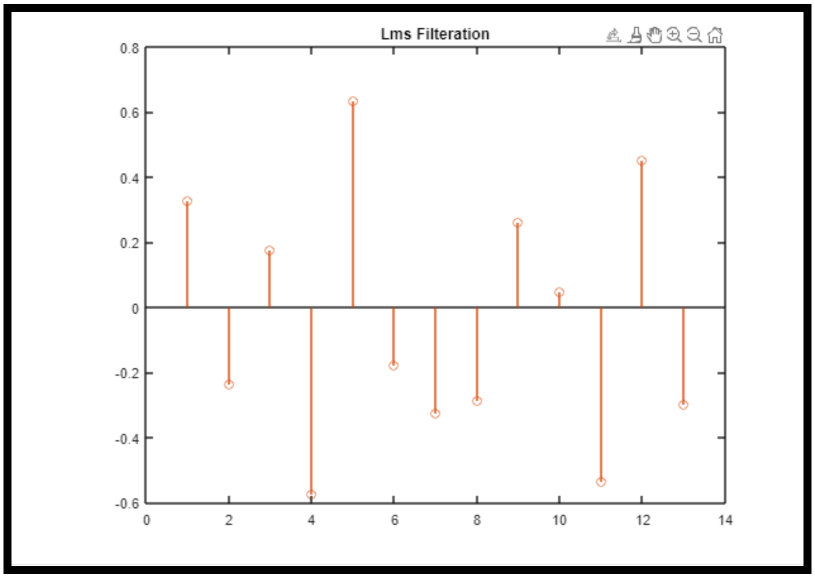
Question 2

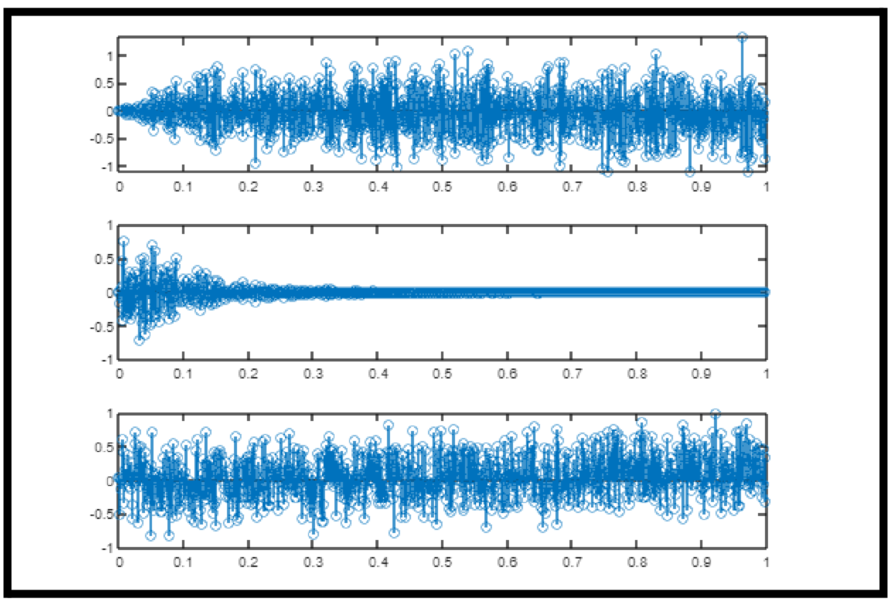
Part a)

When SNR=10dB and meo=0.1

```
%Part#1
%%===================================================================
%%Implement Adaptive Filter with size=0.1
Fs=1000;
Ts=1/Fs;
order=12;
mu=0.1;
SNR=10;
t=0:Ts:1-Ts;
x=sin(2*pi*t/15);
noise=randn(size(x));
x=x+noise;
x=max(x)\x;
x=x';
s=sin((2*pi*t/15)-pi/3);
b=fir1(order,0.3,'low');
d=awgn(b,SNR);

filteraion=filter(d,1,x);

lms=dsp.LMSFilter(order+1,'StepSize',mu,'WeightsOutputPort',true);
[y,e,w]=step(lms,x,filteraion);
stem([d.' w]);
title('Lms Filteration');
%%===================================================================

%Part a)
%plotting
figure(2)
subplot(3,1,1);



stem(t,y)
subplot(3,1,2);
```

```
stem(t,e);
subplot(3,1,3);
stem(t,x);
```





Explanation:

As we discussed adaptive filter before so in this task or question we need to analyze the signal with Gaussian Noise, First we analyze the signal as reference signal than add the AWGN and Gaussian noise in desired signal than all the process as before we discussed in adaptive filtration and result shown above.

Part b)

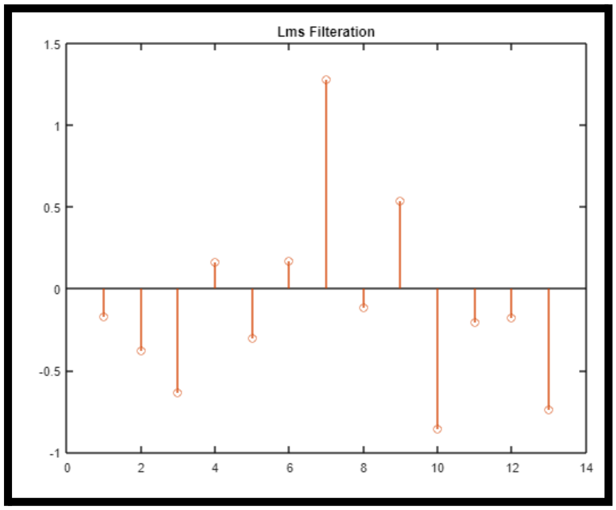
SNR varying of 10 to 5

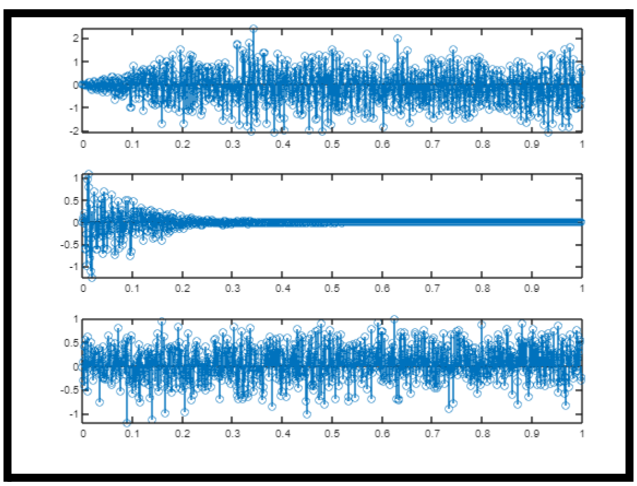
```
%%Part b)
%%=================================================================
%%Implement Adaptive Filter with size=0.4
Fs=1000;
Ts=1/Fs;
order=12;
mu=0.1;
SNR=5;
t=0:Ts:1-Ts;
x=sin(2*pi*t/15);
noise=randn(size(x));
x=x+noise;
x=max(x)\x;
x=x';
s=sin((2*pi*t/15)-pi/3);
b=fir1(order,0.3,'low');
d=awgn(b,SNR);

filteraion=filter(d,1,x);

lms=dsp.LMSFilter(order+1,'StepSize',mu,'WeightsOutputPort',true);
[y,e,w]=step(lms,x,filteraion);
stem([d.' w]);
title('Lms Filteration');

figure(3)
subplot(3,1,1);
stem(t,y)
subplot(3,1,2);
stem(t,e);
subplot(3,1,3);
stem(t,x);
```

Explanation:

It will be done as above task but in this task we take the SNR value to 10 to -10 while the size is 0.1, SNR value actually the sound to noise ration value that define the noise in Gaussian noise, that in which limit we need to add noise in signal, when SNR is -5 we can easily observe below that noise ratio is less while if we compared it with 5 it will show that noise ratio is high and Noise in the signal in high range.

When SNR=5

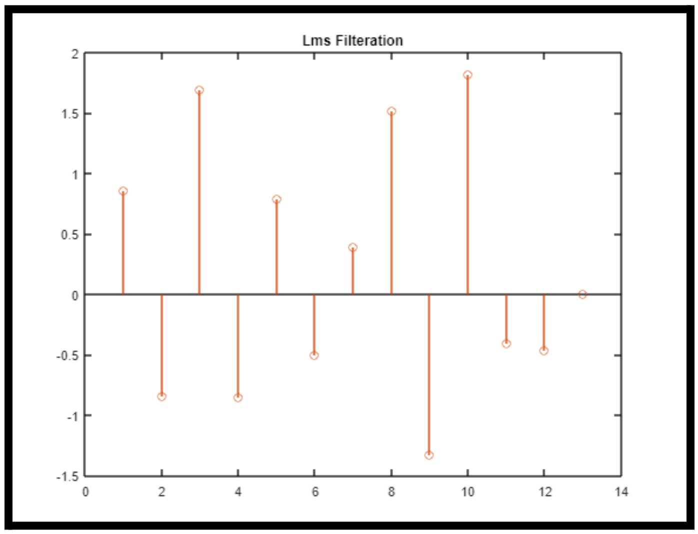


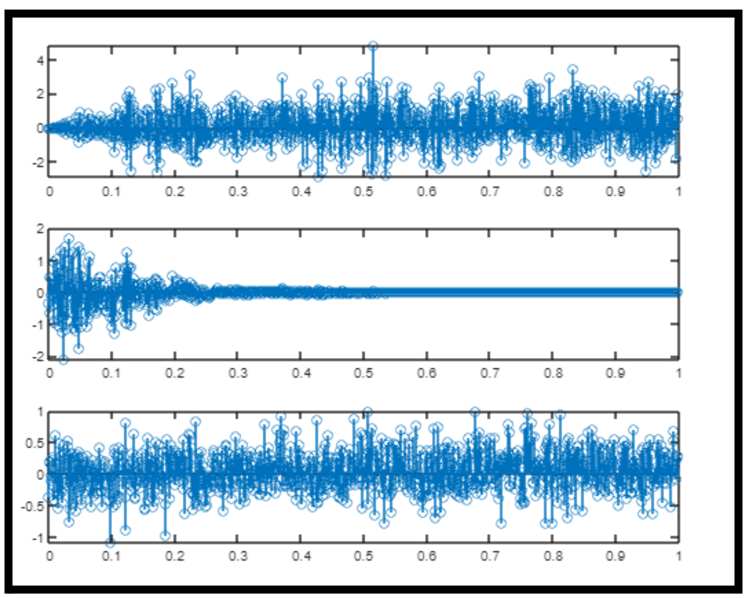


Explanation:

When SNR value is 5 least mean square value start from zero that shows that there is huge Noise added in our signal that is good for Gaussian noise now if we observe second graph parameters we can easily be observed that there is also huge noise in signal as campared to previous signals

When SNR=0





Explanation:

In this task now SNR value is zero mean sound to noise ratio is zero but as we observe it we will conclude that sound is zero mean in this signal everything is noise no sound because if noise zero SNR will be infinite and its not now, its SNR is zero mean sound is zero so above graphs just show the Noise .

Part c)

When SNR=5 , and N vary from 4,6,8,10

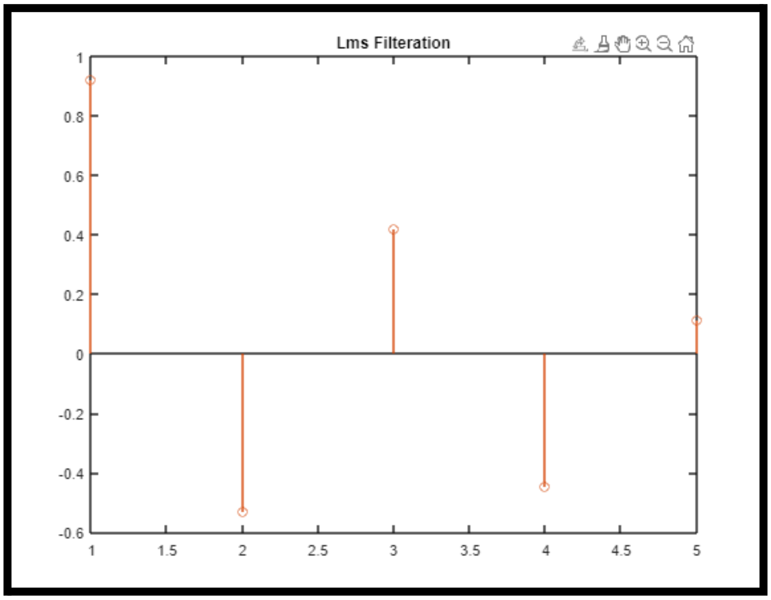
```
%%Part c)

%%Implement Adaptive Filter with size=0.4
Fs=1000;
Ts=1/Fs;
order=4;
mu=0.1;
SNR=5;
t=0:Ts:1-Ts;
x=sin(2*pi*t/15);
noise=randn(size(x));
x=x+noise;
x=max(x)\x;
x=x';
s=sin((2*pi*t/15)-pi/3);
b=fir1(order,0.3,'low');
d=awgn(b,SNR);

filteraion=filter(d,1,x);

lms=dsp.LMSFilter(order+1,'StepSize',mu,'WeightsOutputPort',true);
[y,e,w]=step(lms,x,filteraion);
stem([d.' w]);
title('Lms Filteration');

figure(4)
subplot(3,1,1);
stem(t,y)
```

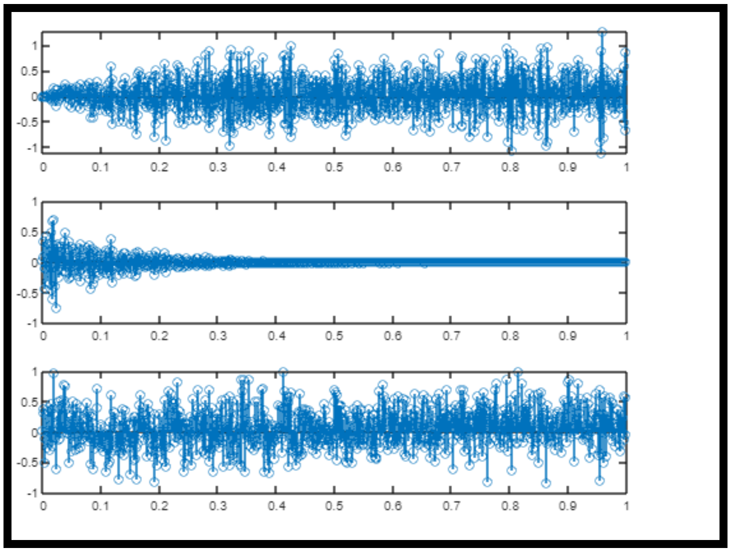
```
subplot(3,1,2);
stem(t,e);
subplot(3,1,3);
stem(t,x);
```

Explanation:

In this SNR remain the same as 5dB while the N number of orders will be change in every condition, In this we will easily observed that what effect on it when we changed the N values what the plotting of LMS change when N orders change and SNR remain the same

When N=4



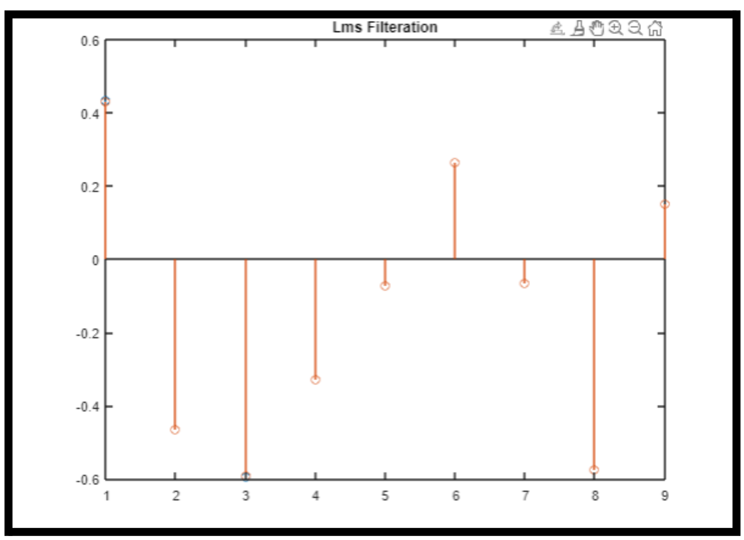


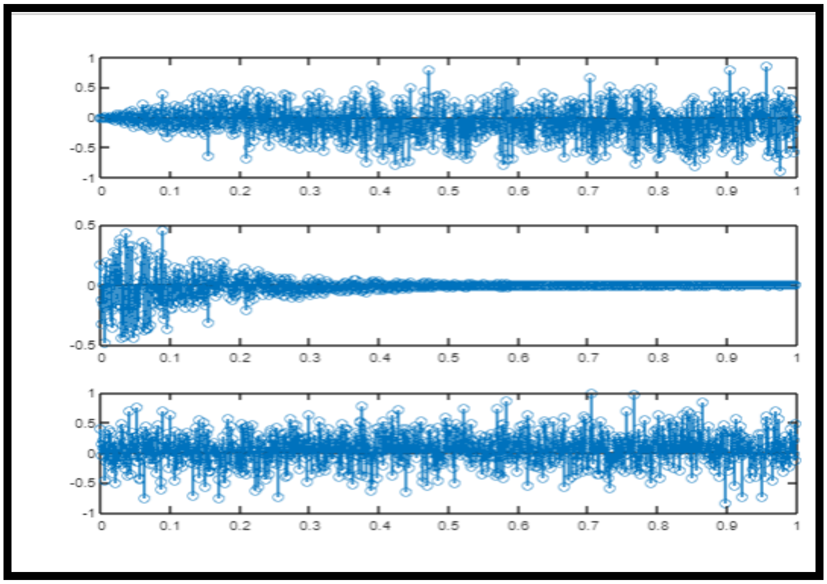
Explanation:

In the above graph is the result of N=4 when we take N=4 we get the LMS values up to 5 signal one signal will be added on the lean max square value it denote that at Number of orders 4 it

shows the some basic signals and same effect on other parameters plotting Ratio remain same while ordering change the time interval

When N=8





As we discussed in above signal we get the value of N=4 and analyze it now we have N=8 so LMS will show the signal of 9 as one signal is added as a reference signal, Same effect we will easily observed on other parameters that depend on the number of orders of noisy signals.

Part d)

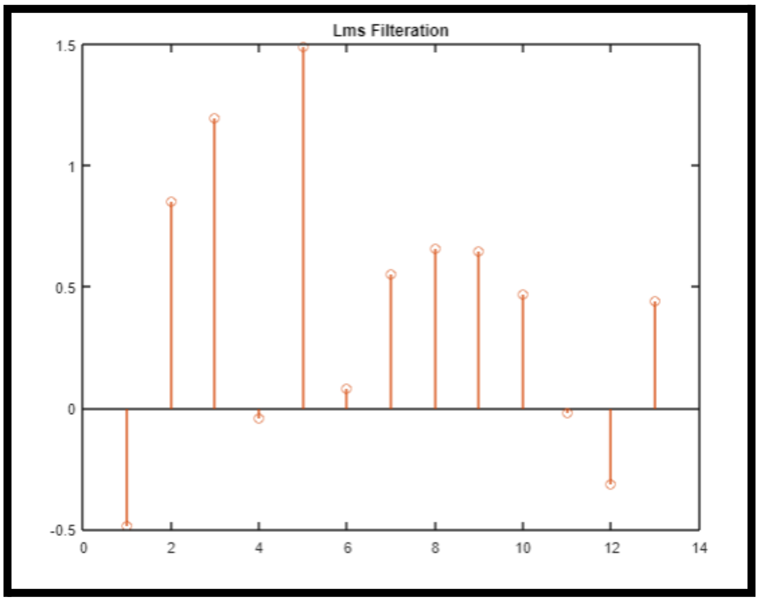
When SNR=5 and size=0.2,0.3,0.4,0.5,0.8

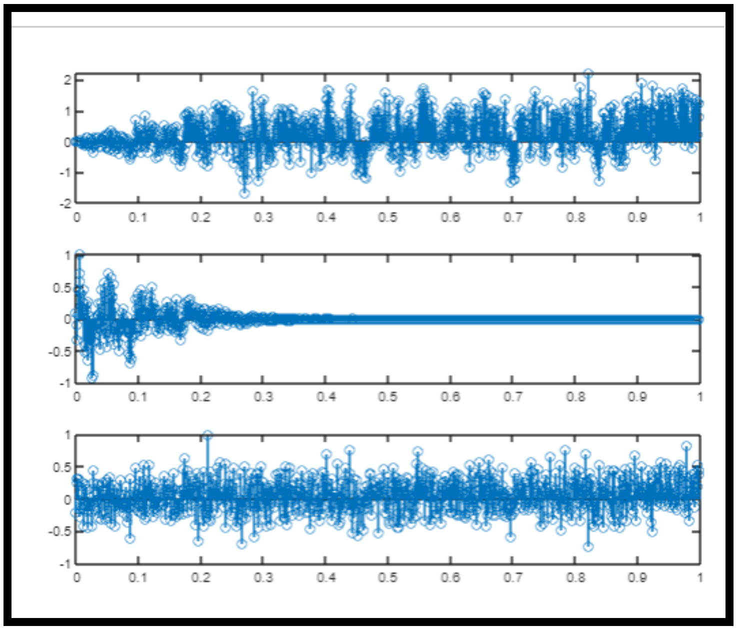
```
%%Implement Adaptive Filter with size=0.4
Fs=1000;
Ts=1/Fs;
order=12;
mu=0.1;
SNR=5;
t=0:Ts:1-Ts;
x=sin(2*pi*t/15);
noise=randn(size(x));
x=x+noise;
x=max(x)\x;
x=x';
s=sin((2*pi*t/15)-pi/3);
b=fir1(order,0.3,'low');
d=awgn(b,SNR);

filteraion=filter(d,1,x);

lms=dsp.LMSFilter(order+1,'StepSize',mu,'WeightsOutputPort',true);
[y,e,w]=step(lms,x,filteraion);
stem([d.' w]);
title('Lms Filteration');

figure(4)
subplot(3,1,1);
stem(t,y)
subplot(3,1,2);
stem(t,e);
subplot(3,1,3);
stem(t,x);
```

Explanation:

In this task we also take the SNR constant and change the size of signal, Actually size change of signal take effect on adaptive filtering and LMS values and plotting as you can easily observed the below graph, In previous analysis most of LMS values in range of 0.2 to 0.3 but now not like that.

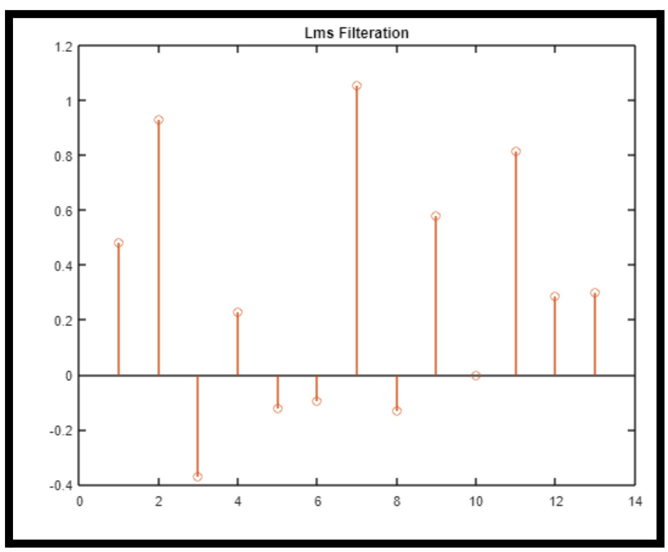
When SNR=5 and size=0.3

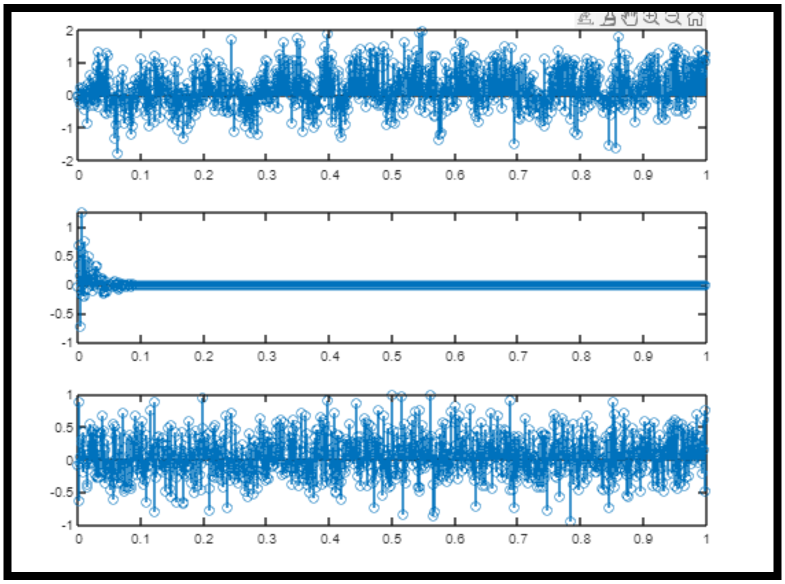




When size of filtering is increase we can easily be observed that LMS values changed and it touch the boundary of 1.5 mean as we increase the size of filtering it effect the LMS values and same effect can be observed in other parameters shown above graph.

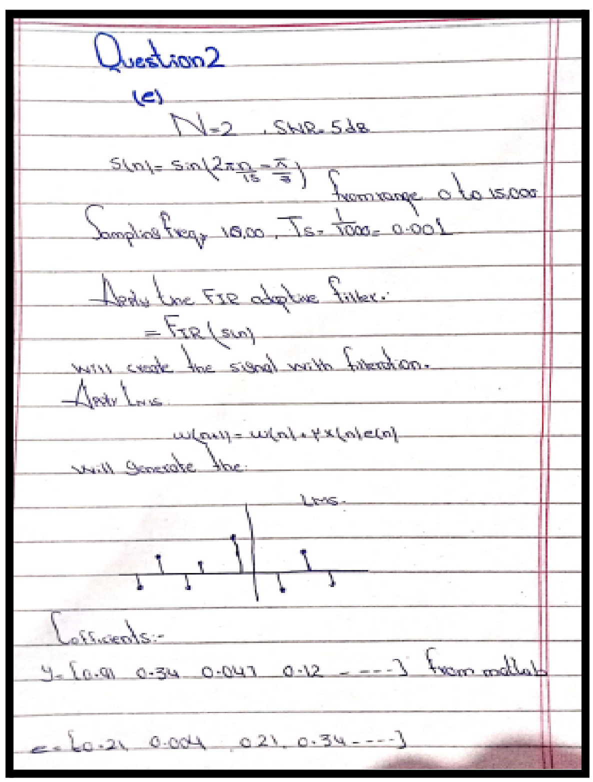
When SNR=5 and size=0.6





Part e)

Handwritten:



Question2

(e)

N=2 , SNR=5dB

$s(n) = \sin(2\pi \frac{n}{15} - \frac{\pi}{3})$ from range 0 to 15,000

Sampling freq = 10,00 , $T_s = \frac{1}{1000} = 0.001$

Apply the FIR adaptive filter:

= FIR(s(n))

will create the signal with filtration.

Apply LMS.

$w(n+1) = w(n) + \mu x(n) e(n)$

will generate the:

LMS.

Coefficients:-

y = {0.91 0.34 0.047 0.12 - - - -} from matlab

e = {0.21 0.004 , 0.21, 0.34 - - - -}

Part f)

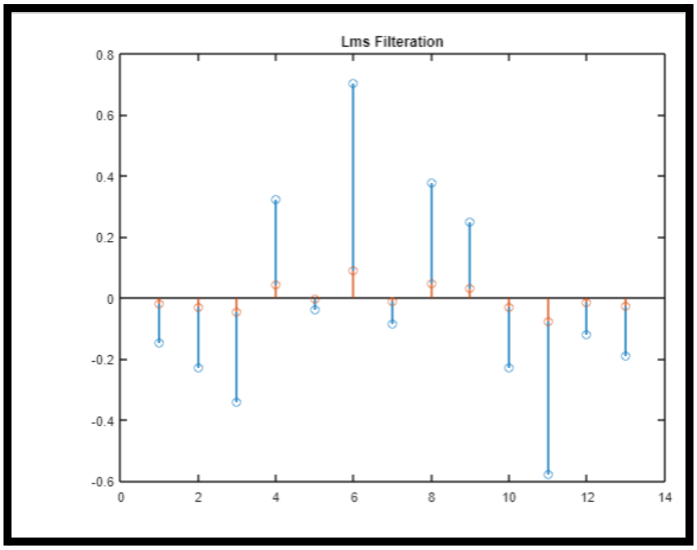
Explanation:

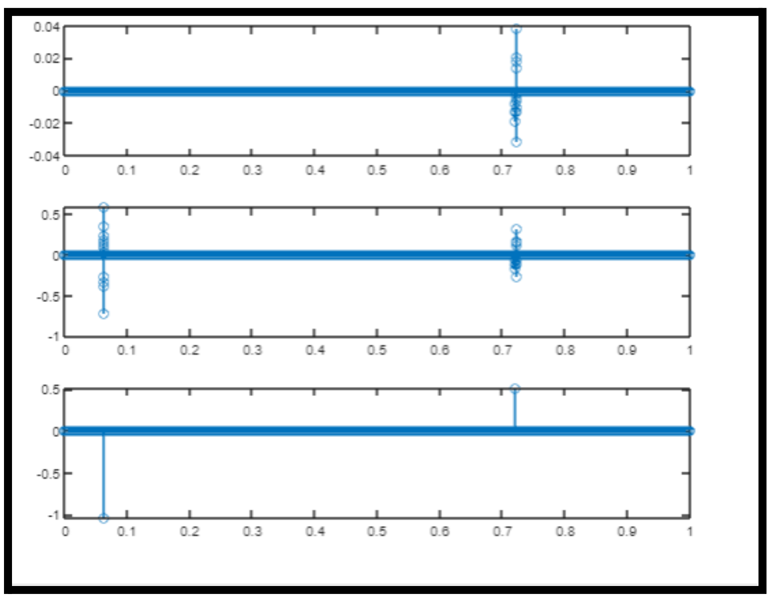
In this task we also take the SNR constant and change the size of signal, Actually size change of signal take effect on adaptive filtering and LMS values and plotting as you can easily observed the below graph, In previous analysis most of LMS values in range of 0.2 to 0.3 but now not like that.

As we discussed in above signal we get the value of N=4 and analyze it now we have N=8 so LMS will show the signal of 9 as one signal is added as a reference signal, Same effect we will easily observed on other parameters that depend on the number of orders of noisy signals.

Question 3

```
%%Question#3
%%=========================================================
%%Load .wav file
Fs=44074;
Ts=1/Fs;
order=12;
mu=0.1;
SNR=10;
t=0:Ts:1-Ts;
x=audioread('Tone.mp4');
noise=randn(size(x));
x=x+noise;
x=max(x)/x;
x=x';
plot(x)
s=sin((2*pi*t/15)-pi/3);
b=fir1(order,0.3,'low');
d=awgn(b,SNR);

filteraion=filter(d,1,x);

lms=dsp.LMSFilter(order+1,'StepSize',mu,'WeightsOutputPort',true);
[y,e,w]=step(lms,x,filteraion);
stem([d.' w]);
title('Lms Filteration');
%Part a)
%plotting
figure(2)
subplot(3,1,1);
stem(t,y)
subplot(3,1,2);
stem(t,e);
subplot(3,1,3);
stem(t,x);
```





Explanation:

In this task we need to analyze and implement the adaptive filter in the audio file that is .wav name . First we read the file that convert it into a one variable X than add some noise in it not

Gaussian noise then we get a desired signal from this analysis and then the same implementation as Problem 1 applies the adaptive filter method and then calculates the value of LMS as shown above.