**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);

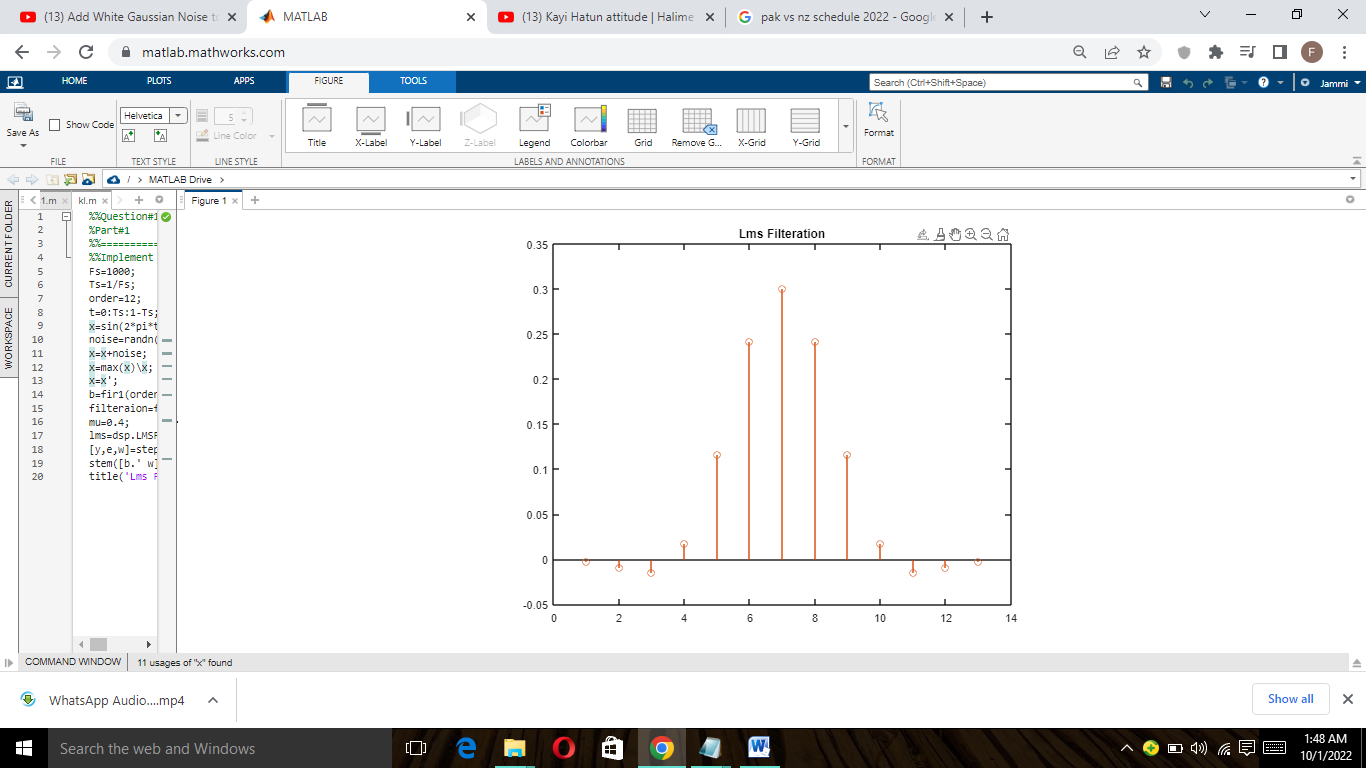
mu=0.4;

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)

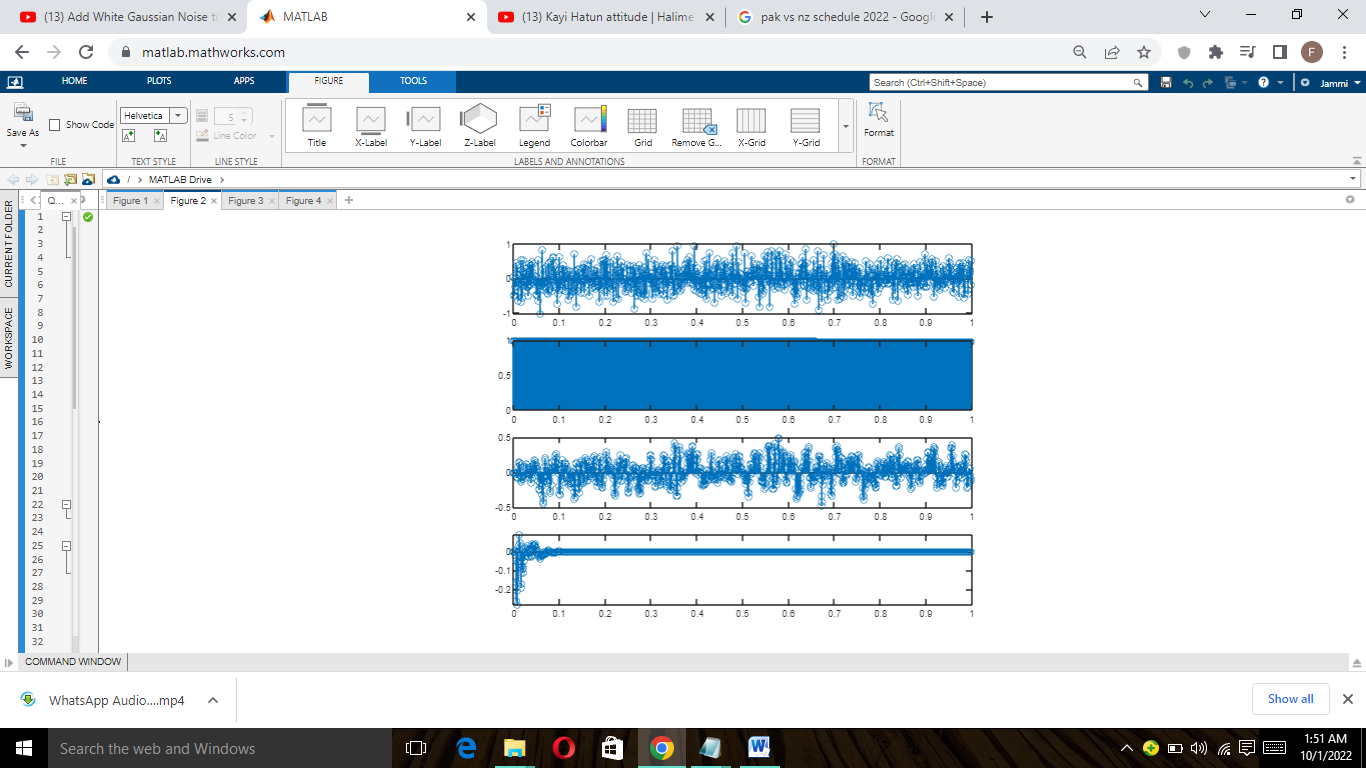
stem(t,y)

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)**

1. **Error Signal:**

%Part 3

%% i) Error Signal

e=d-y;

disp('Error Signal is:')

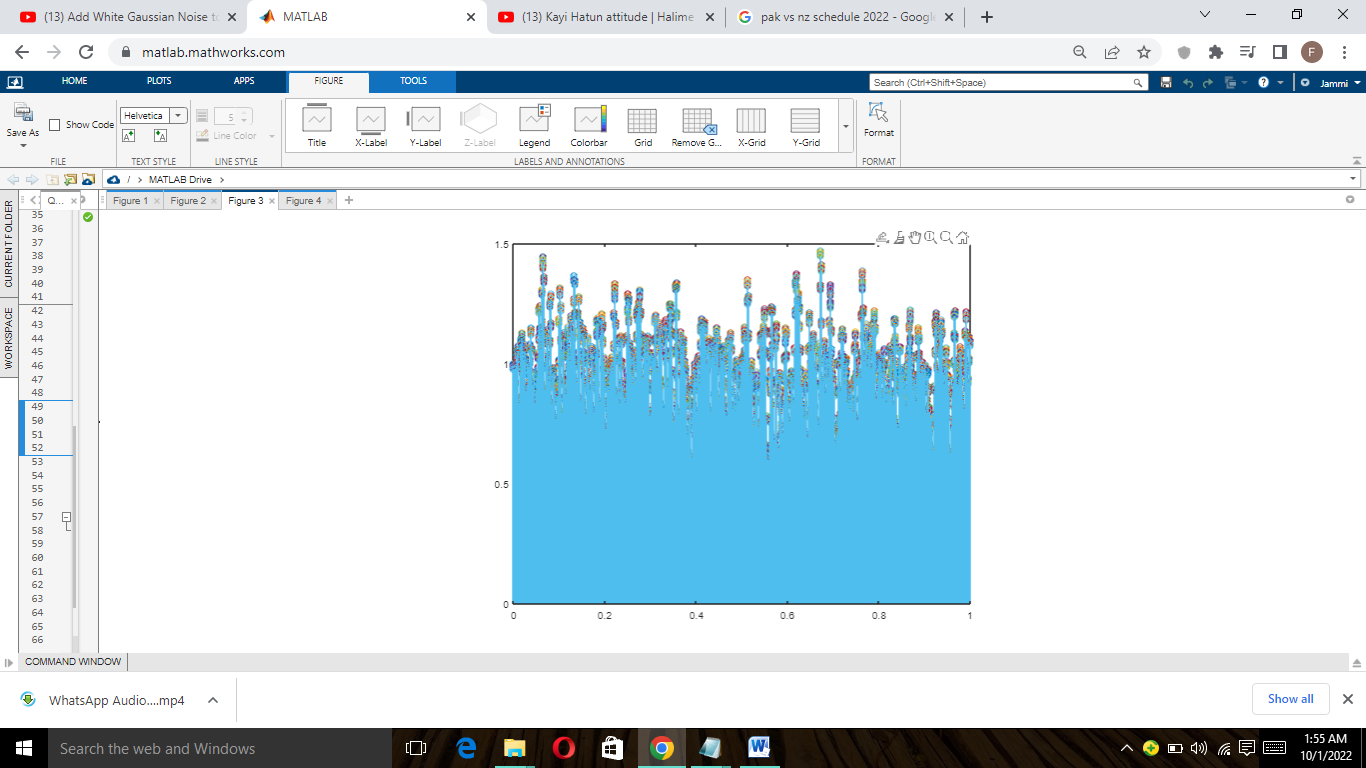
disp(e);

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:

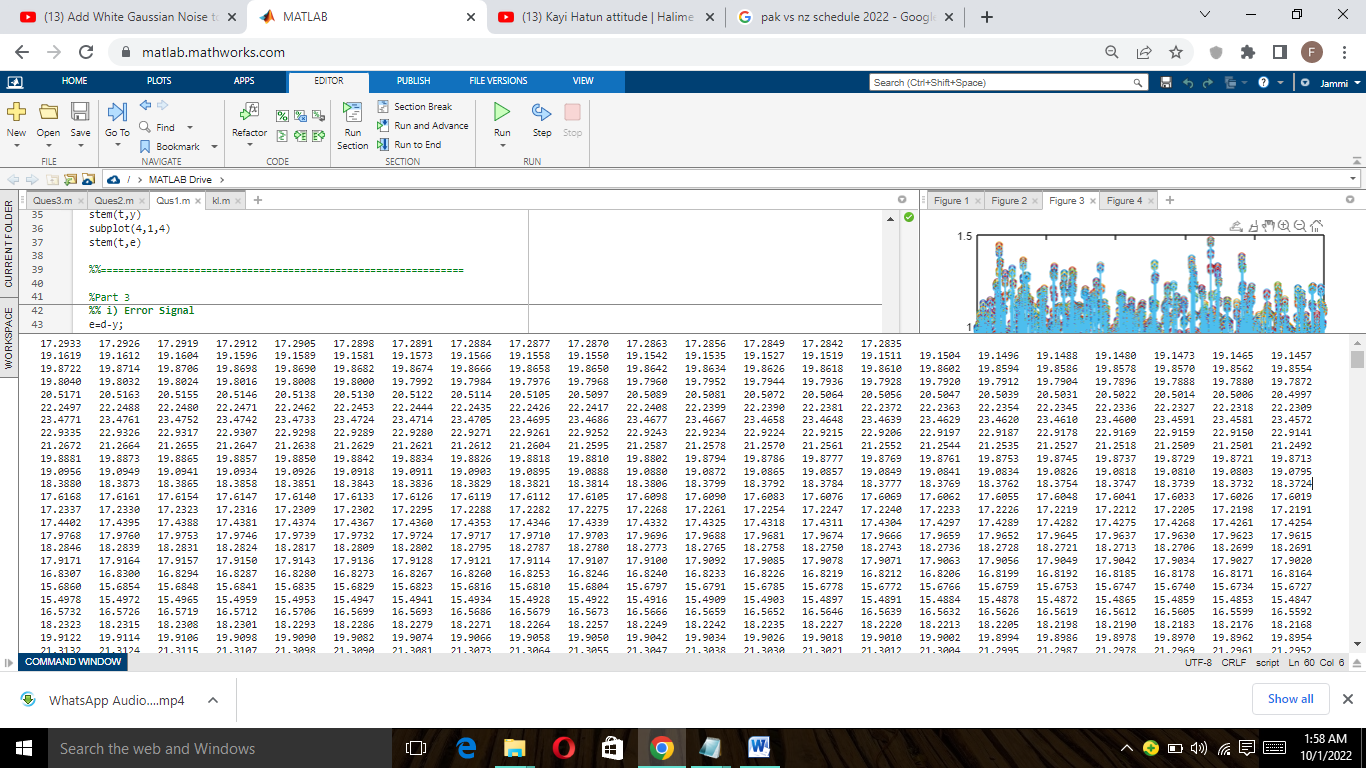


1. **Mean Squared Error:**

%% ii) Squared Error Calculate

E=0.02;

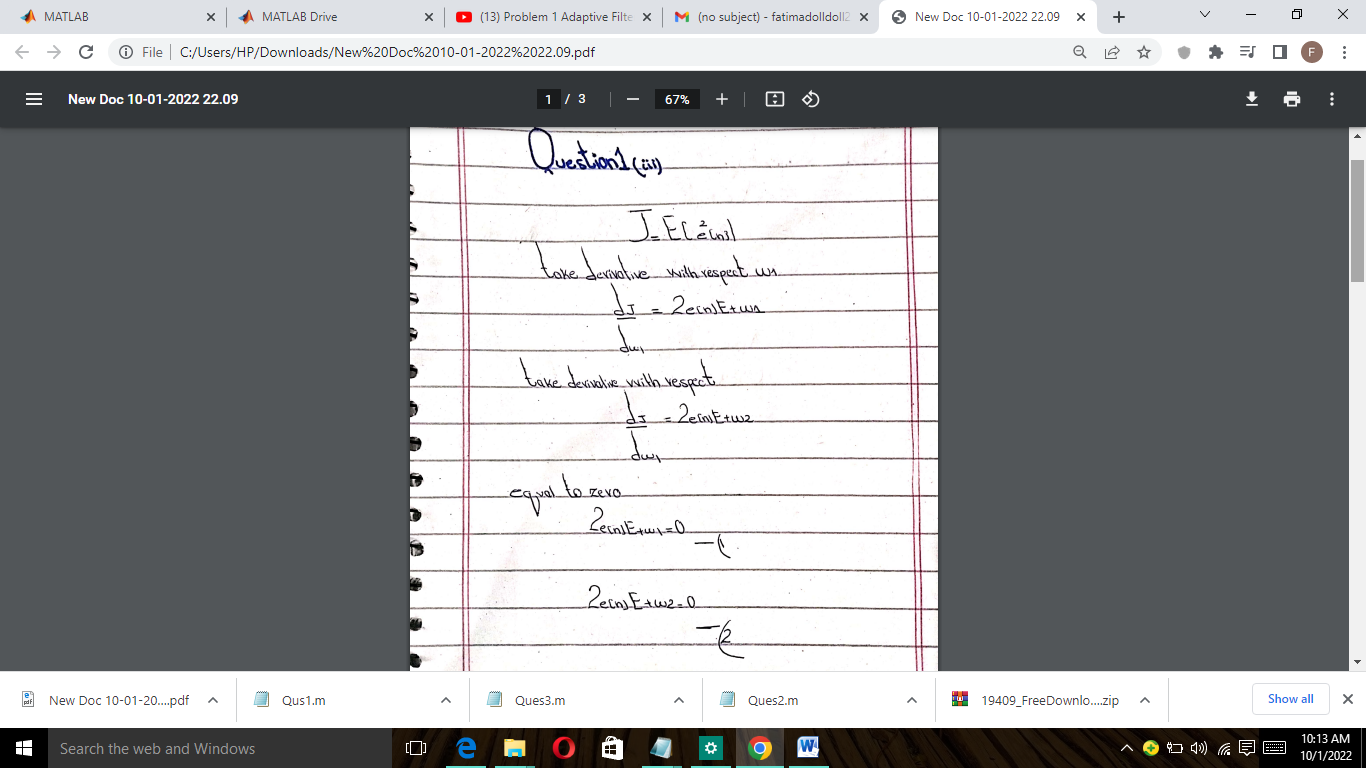
J=E\*e\*e;



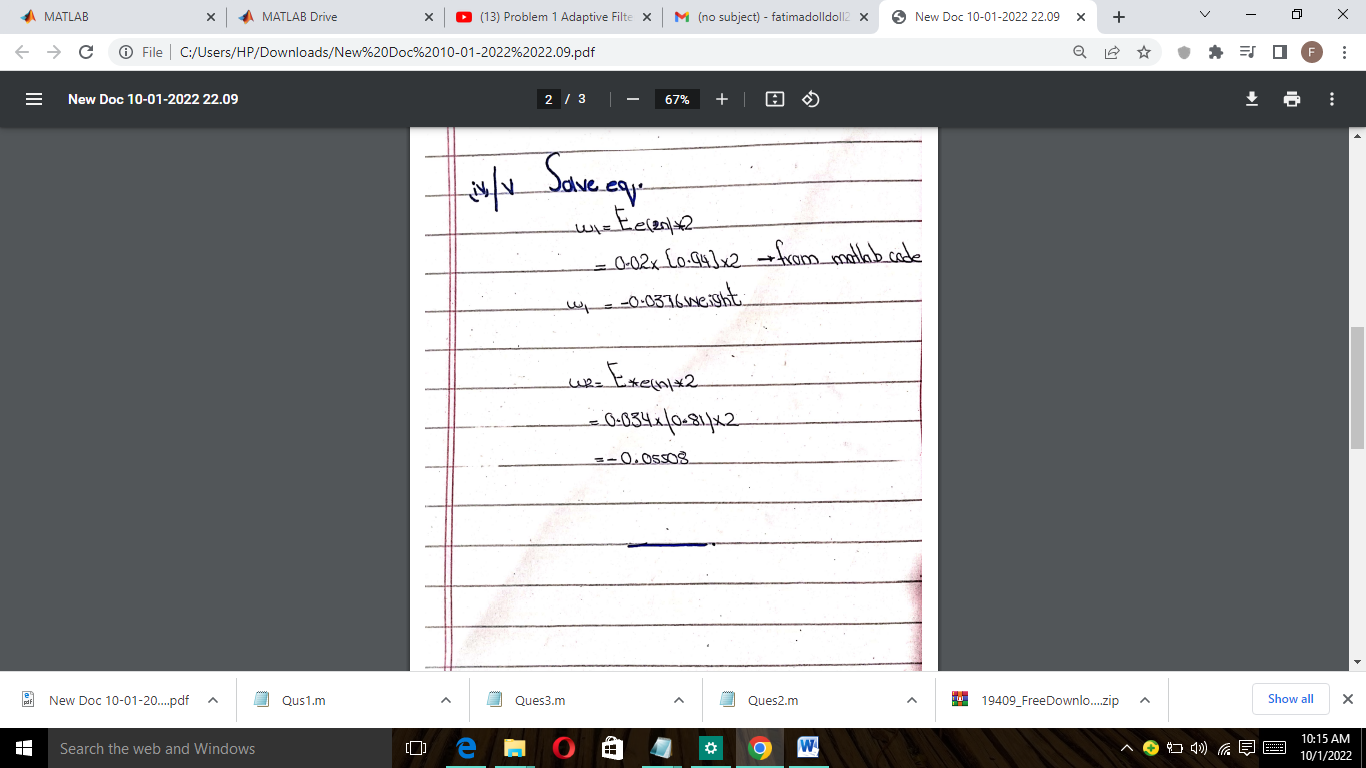
**Explanation:**

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.

1. **Derivation:**



1. **Solve for Weight and Minimum Error:**



**Part d)**

**3D plot with J w0 and w1:**

%%Part 4

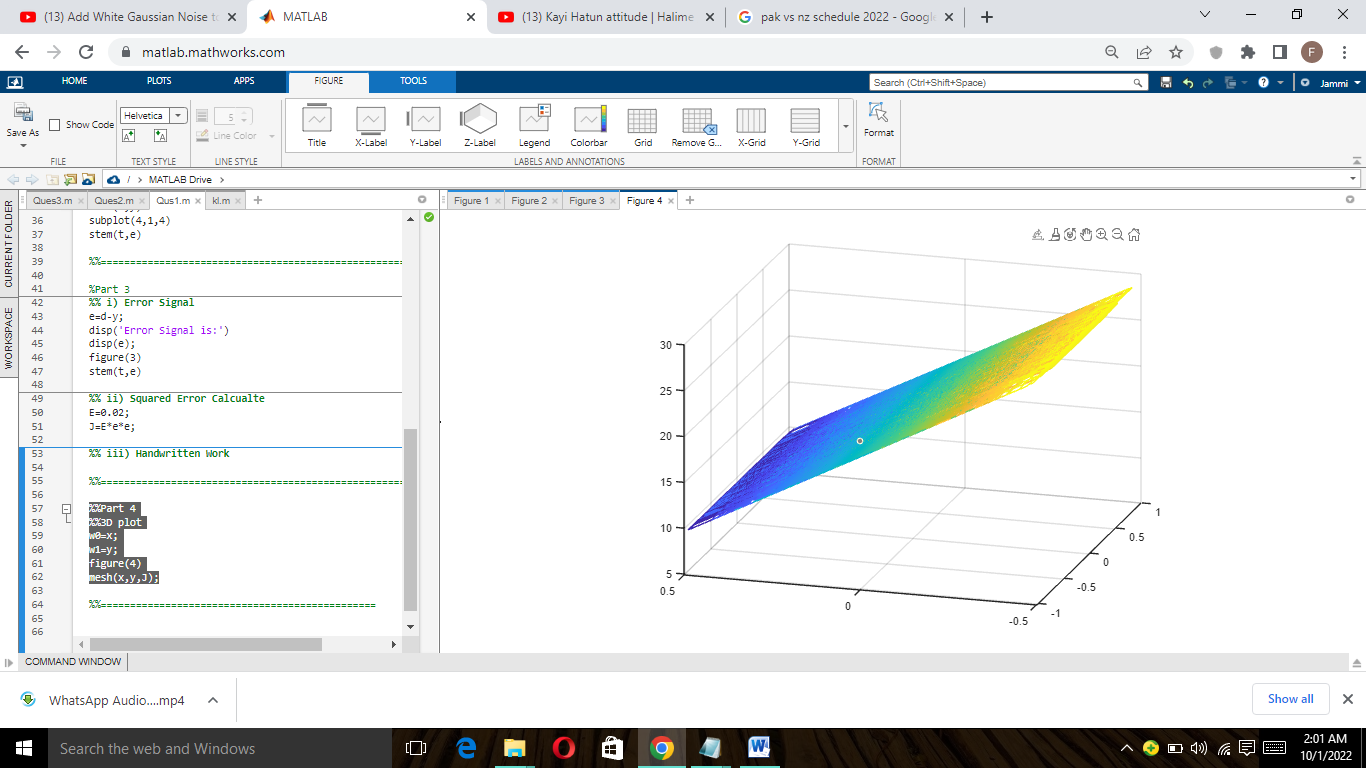
%%3D plot

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);

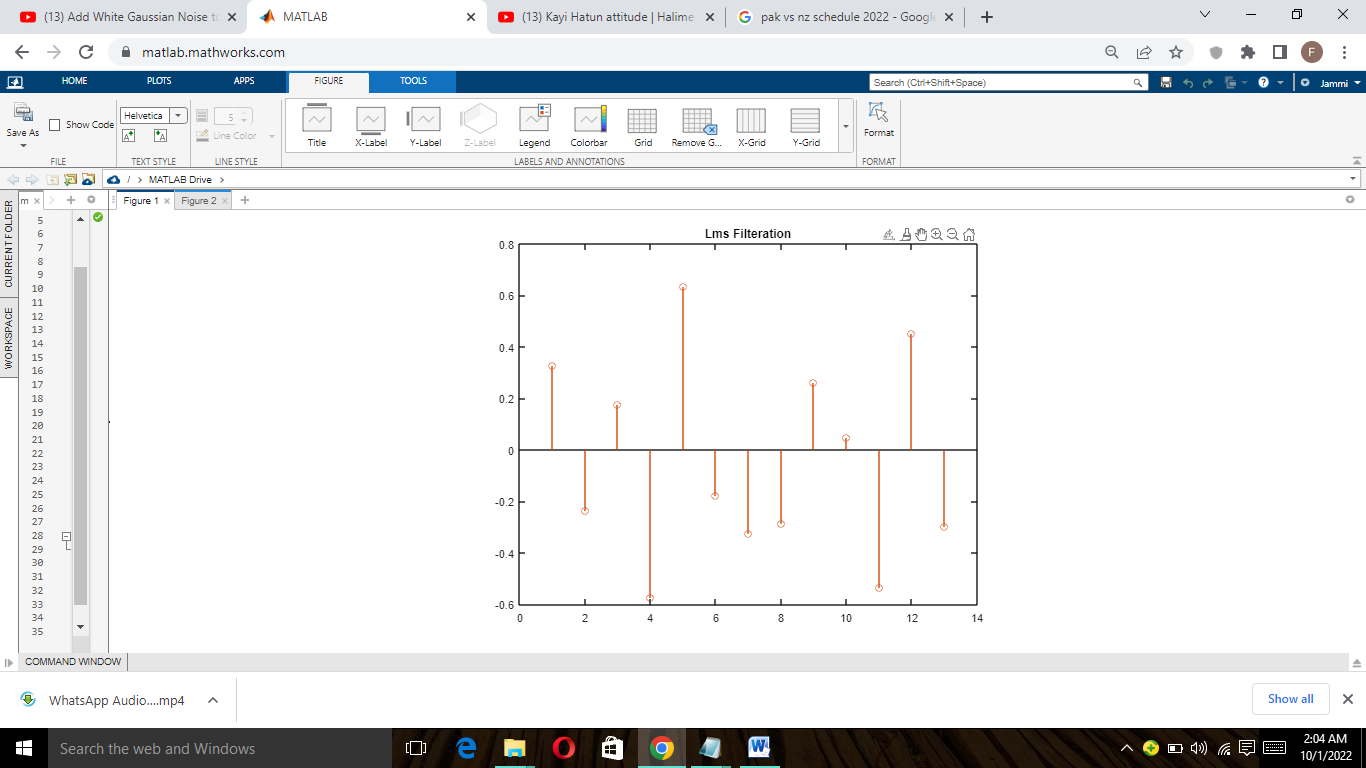
stem(t,y)

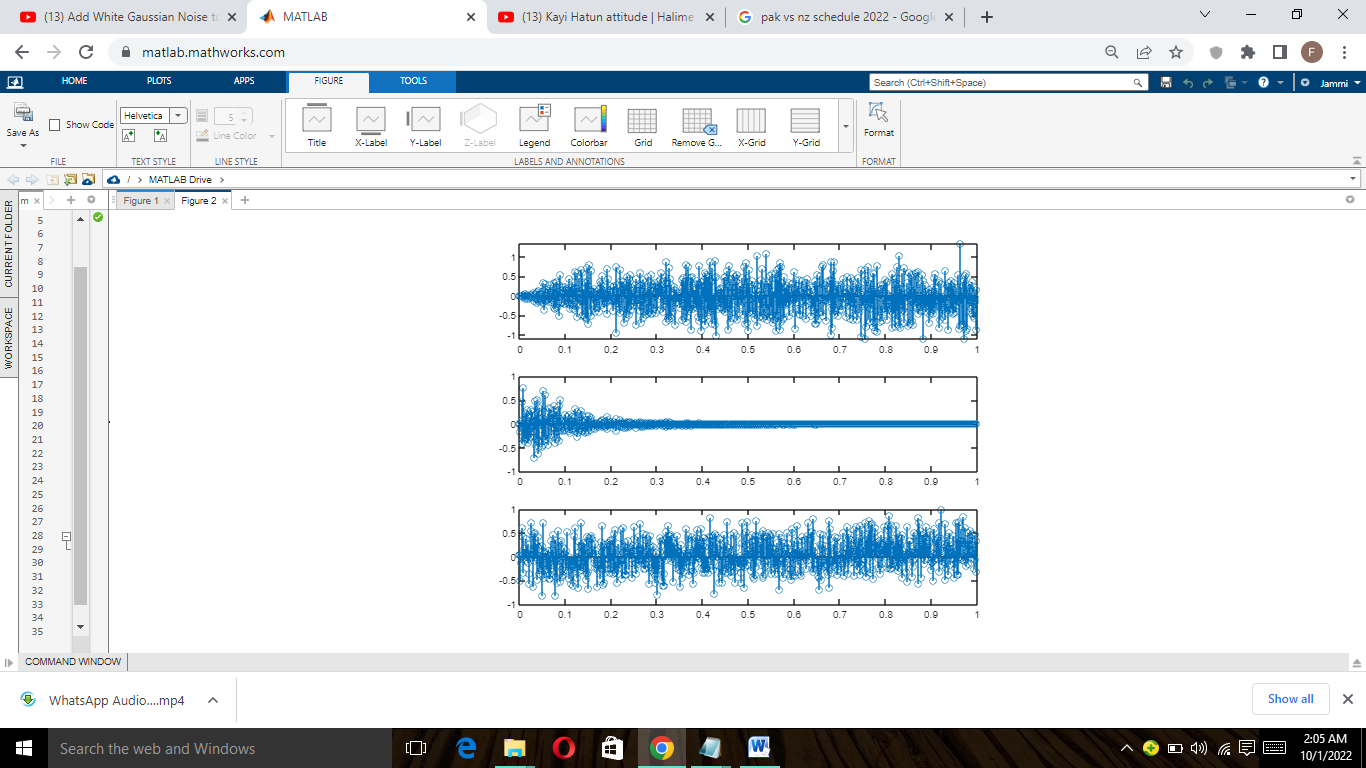
subplot(3,1,2);

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);

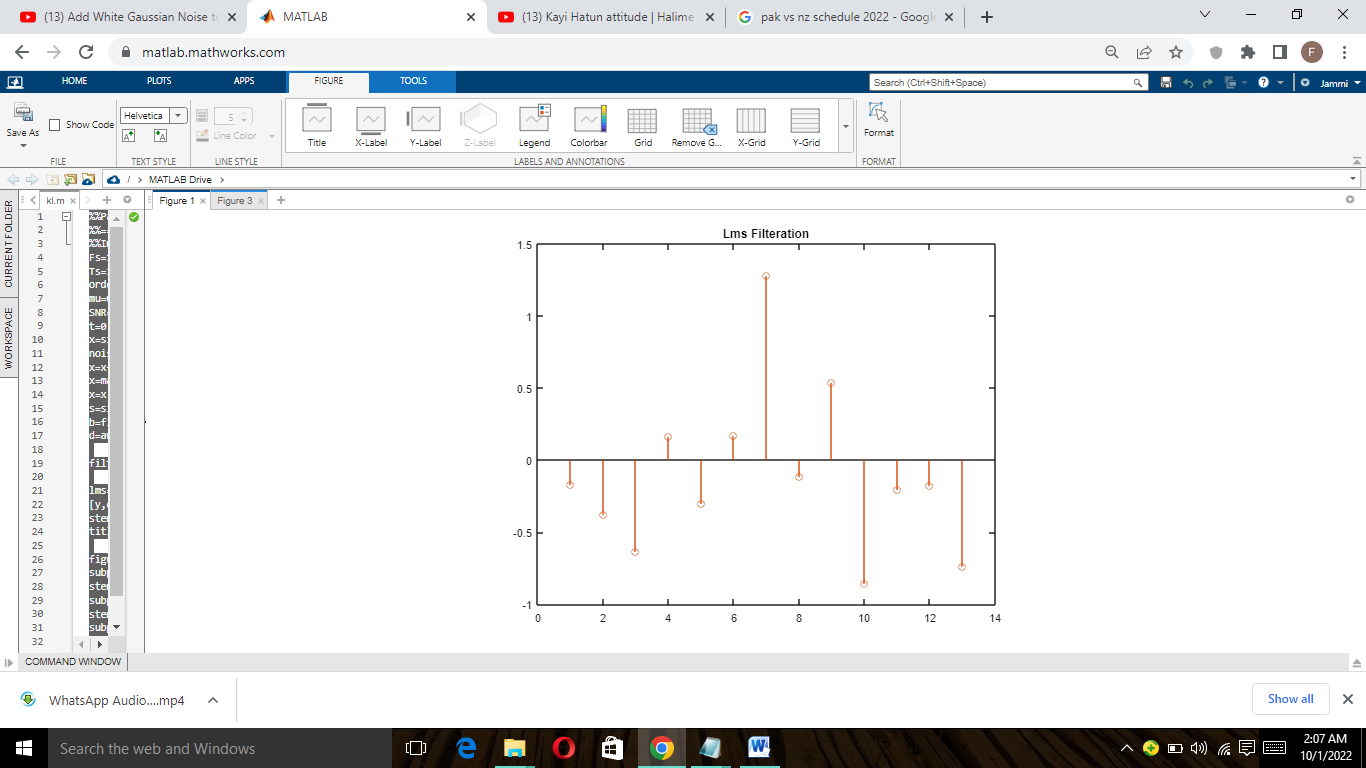
subplot(3,1,3);

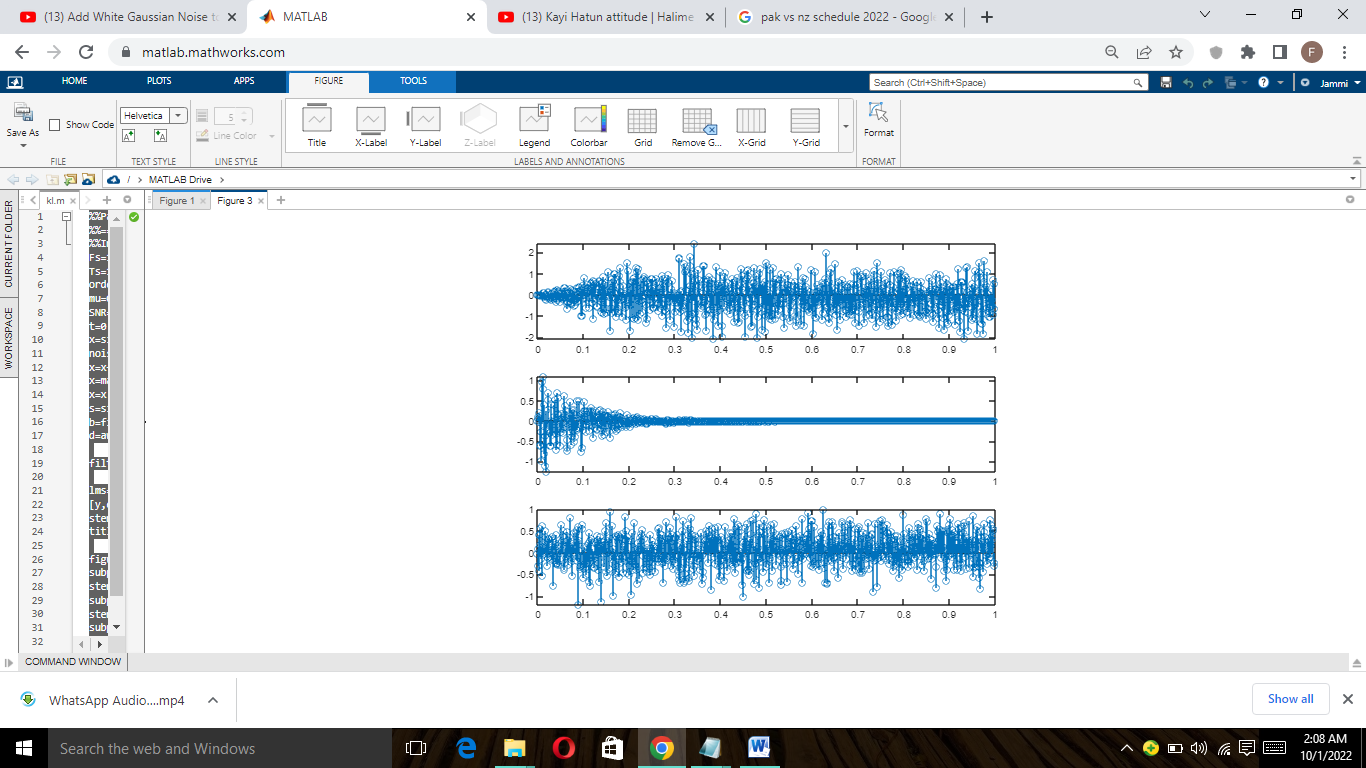
stem(t,x);

**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**

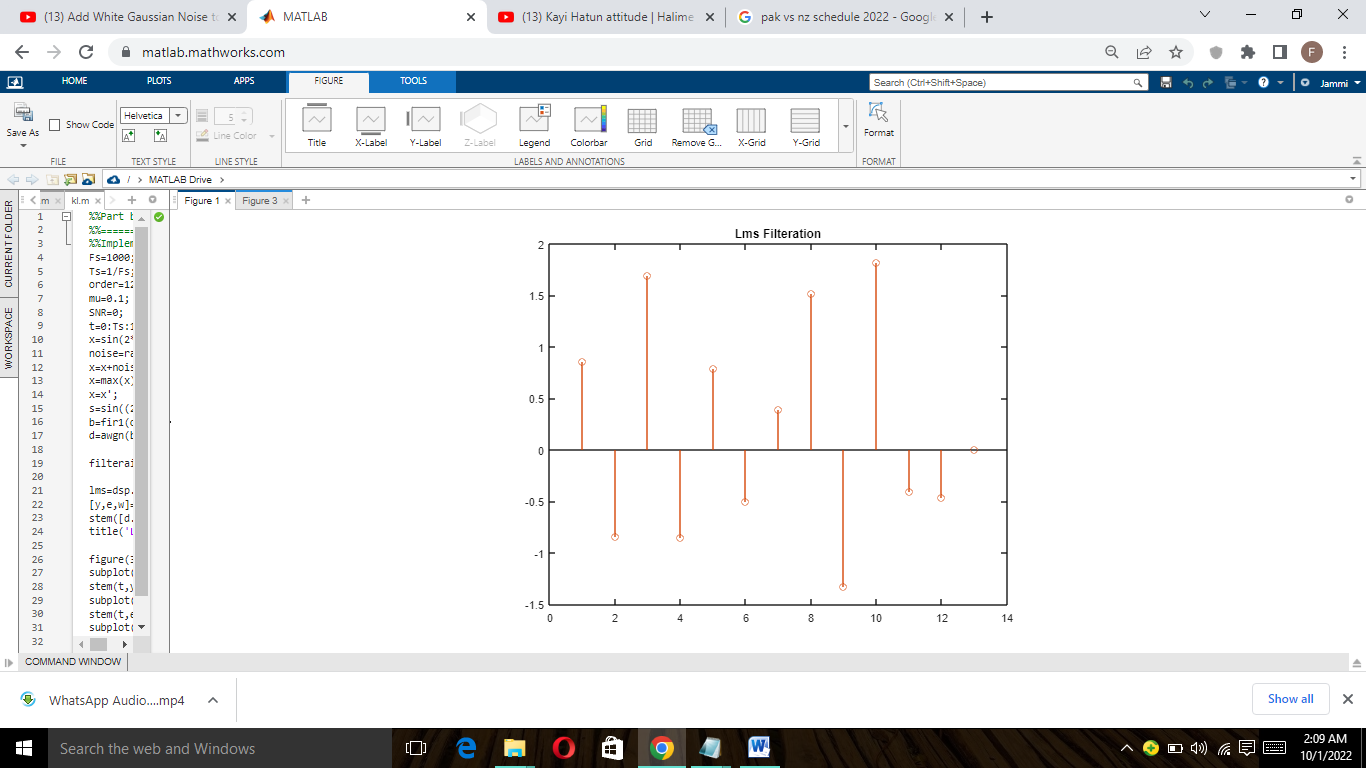


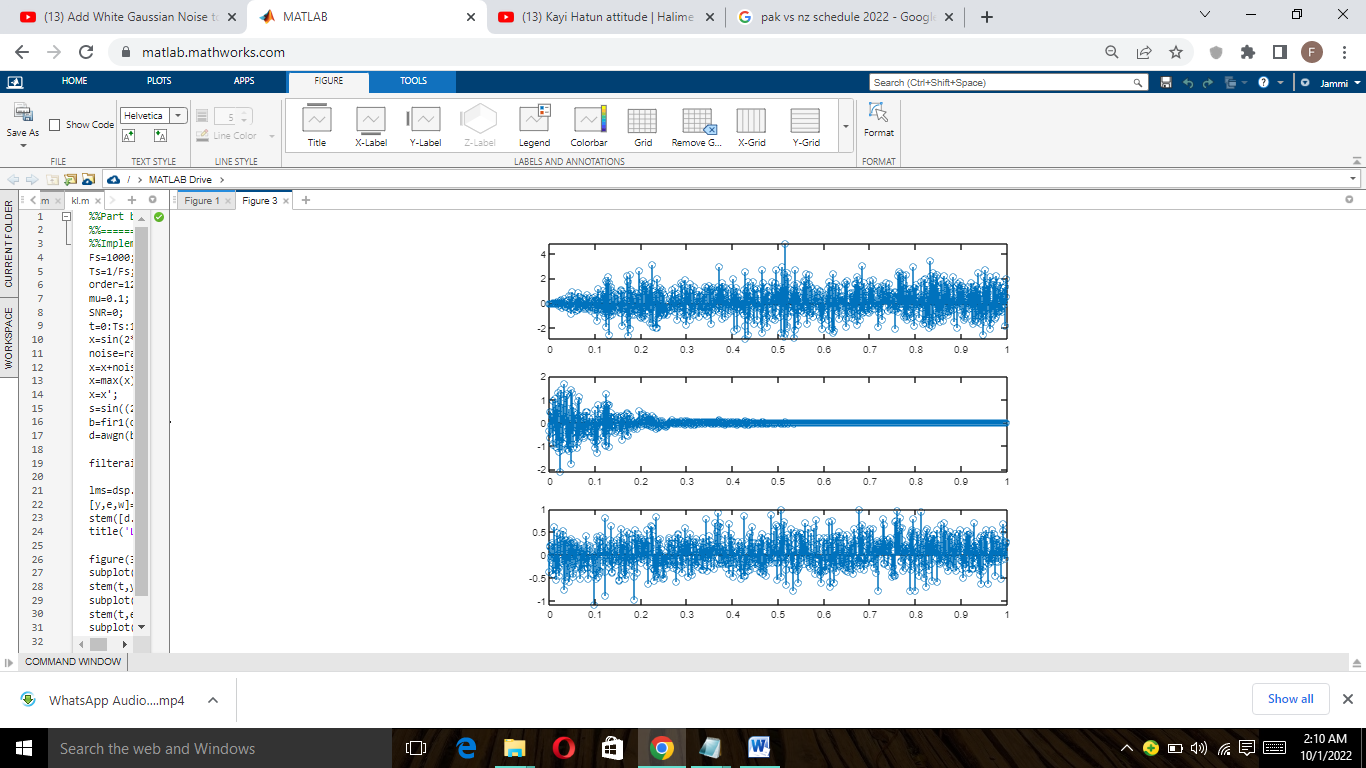


**Explanation:**

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);

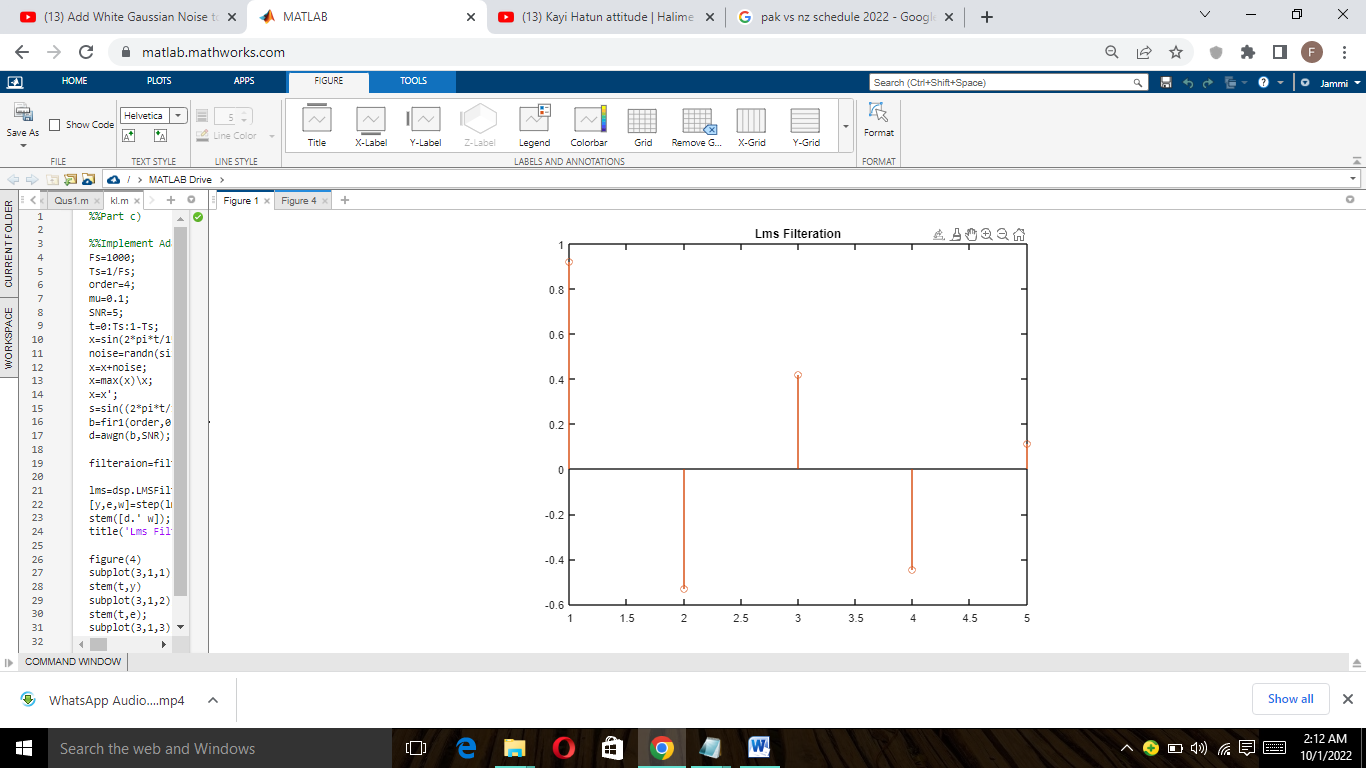
subplot(3,1,3);

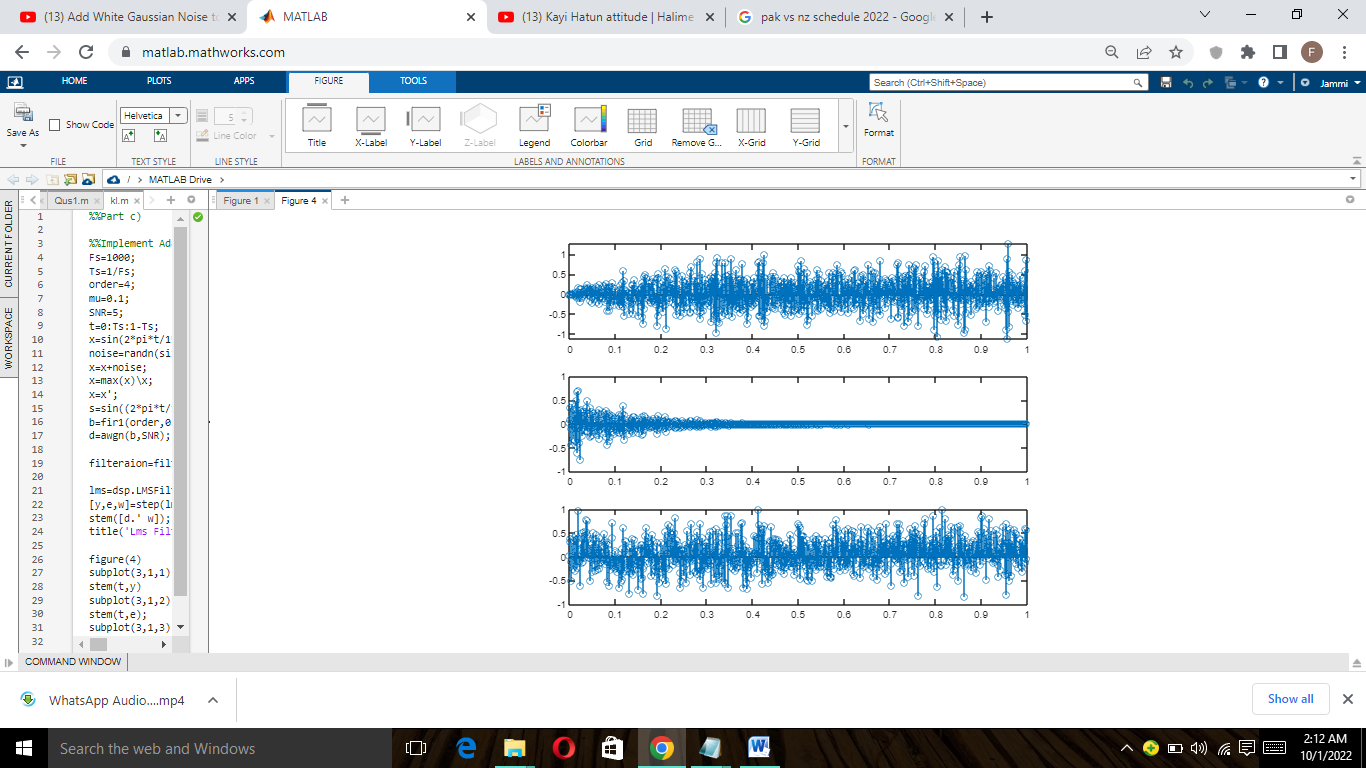
stem(t,x);

**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**



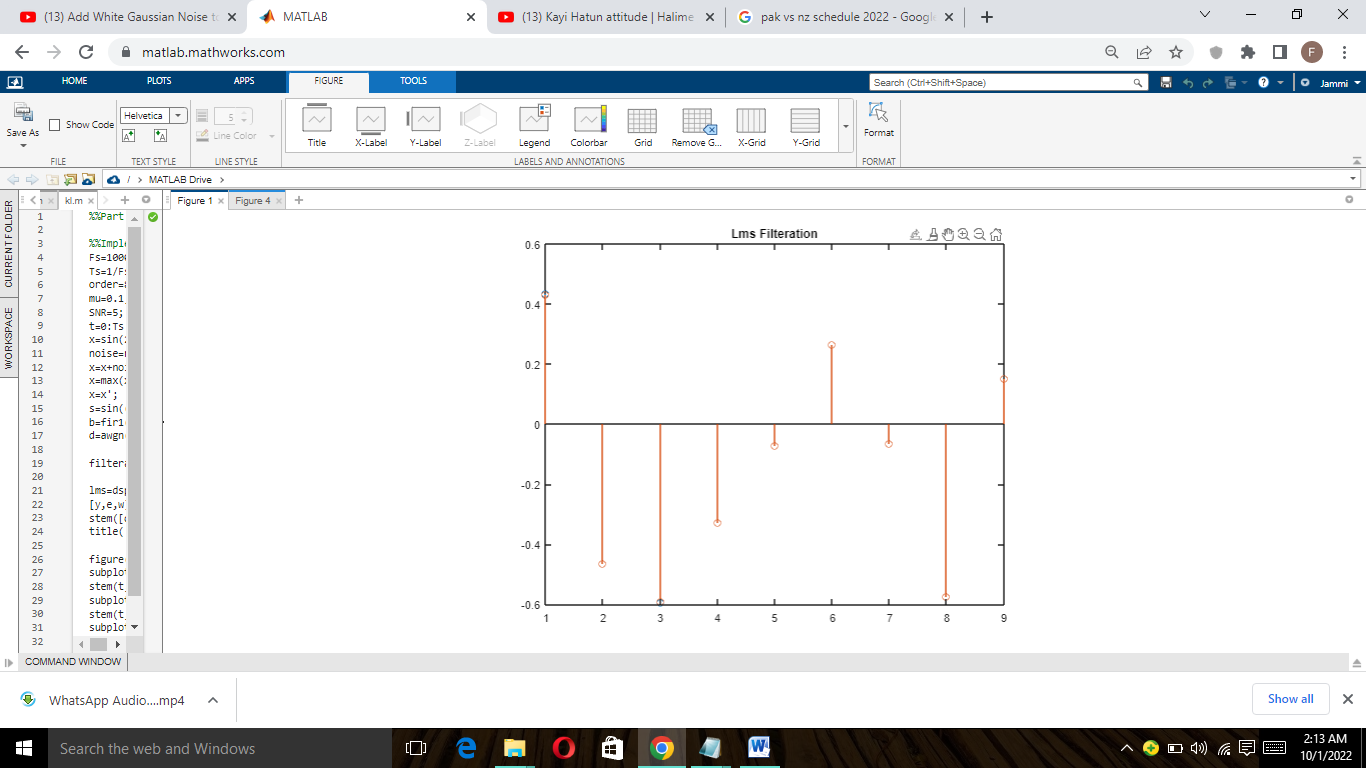


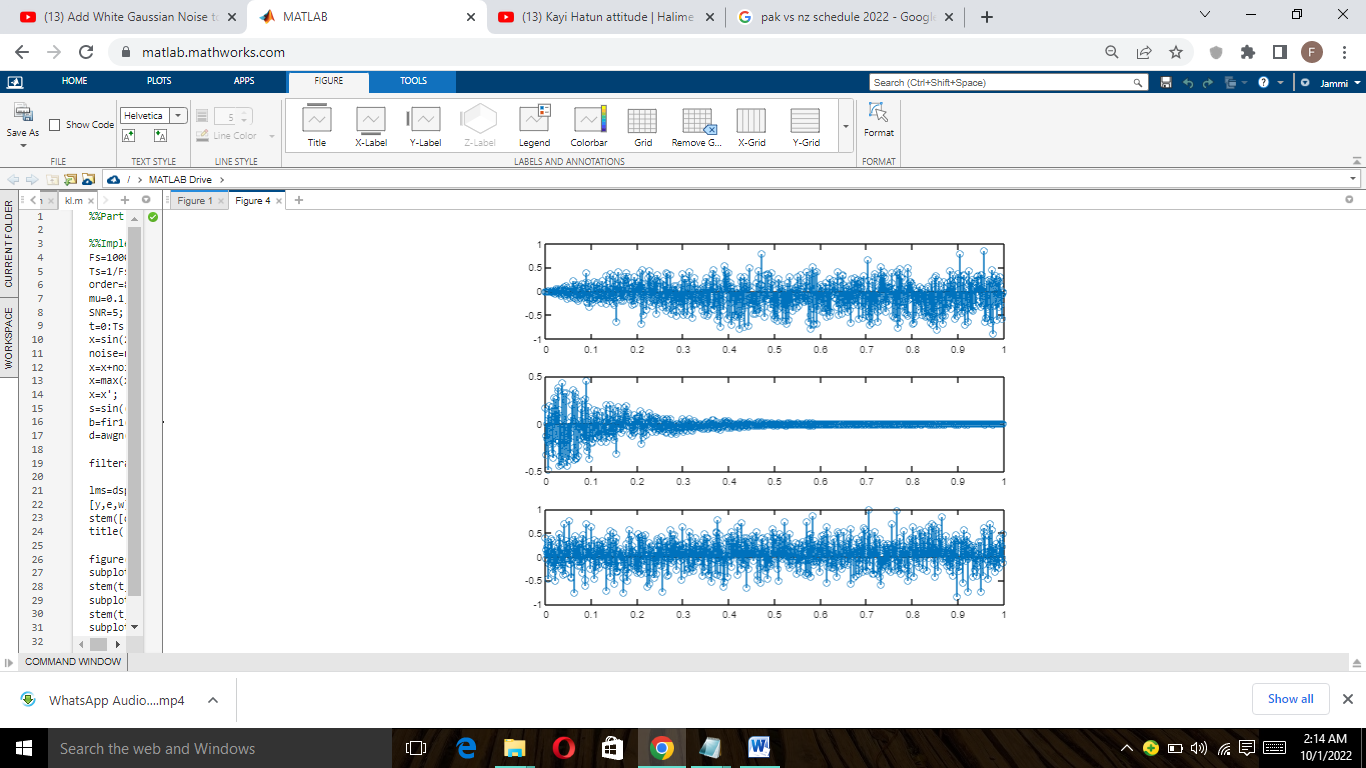
**Explanation:**

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);

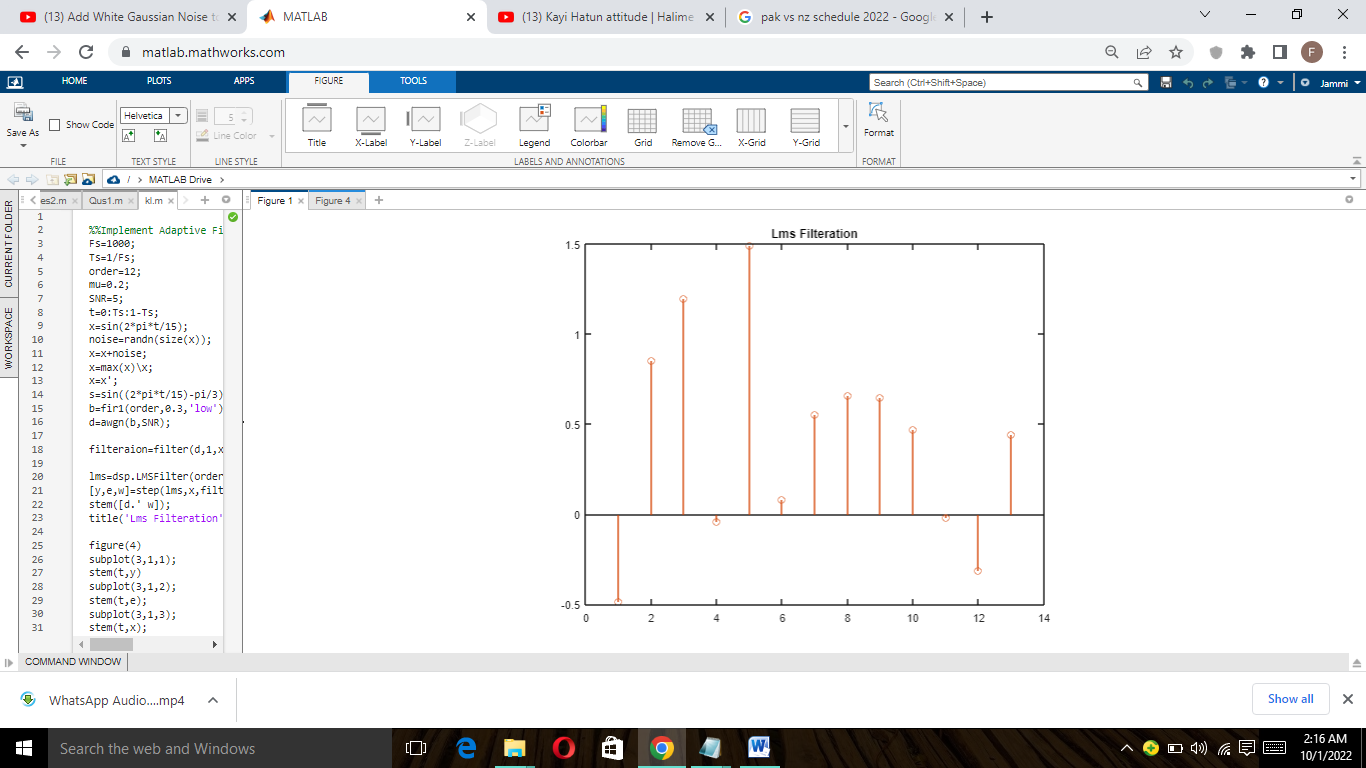
subplot(3,1,3);

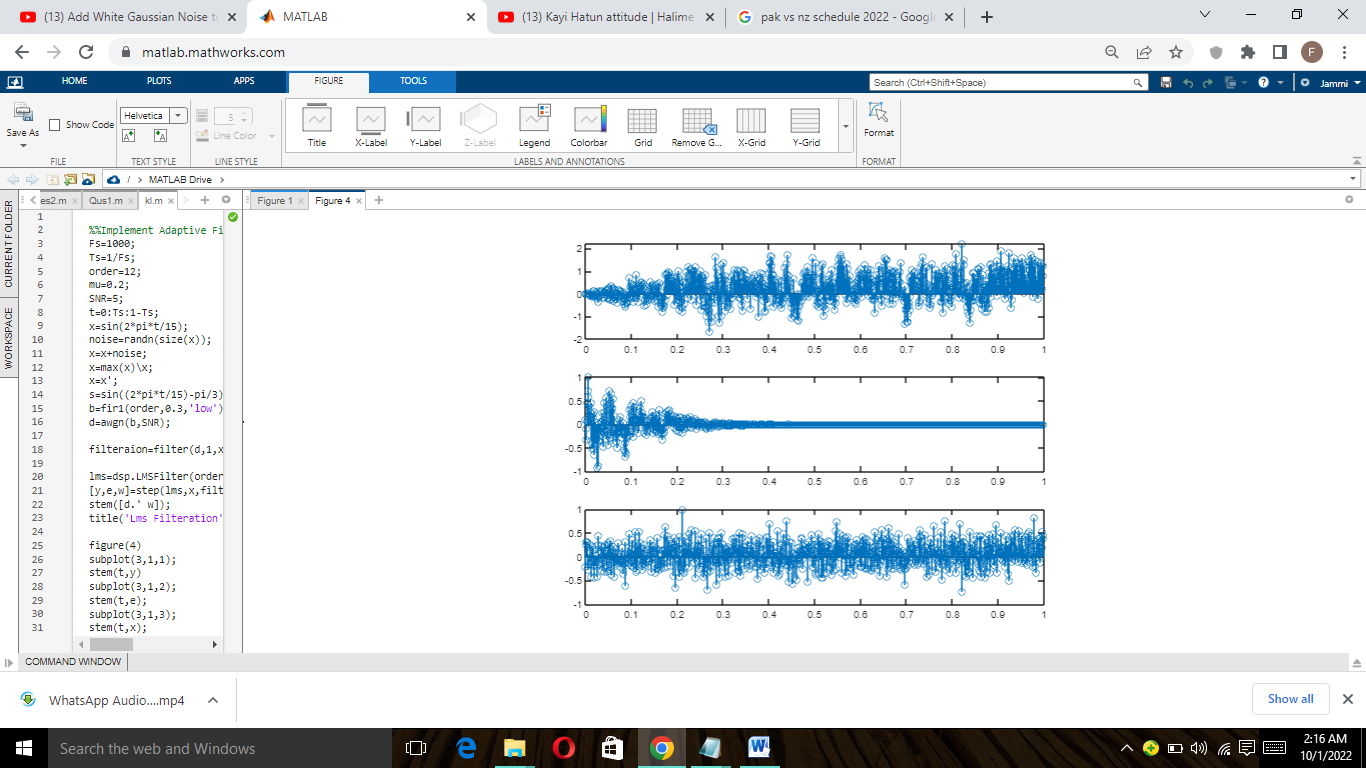
stem(t,x);

**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.

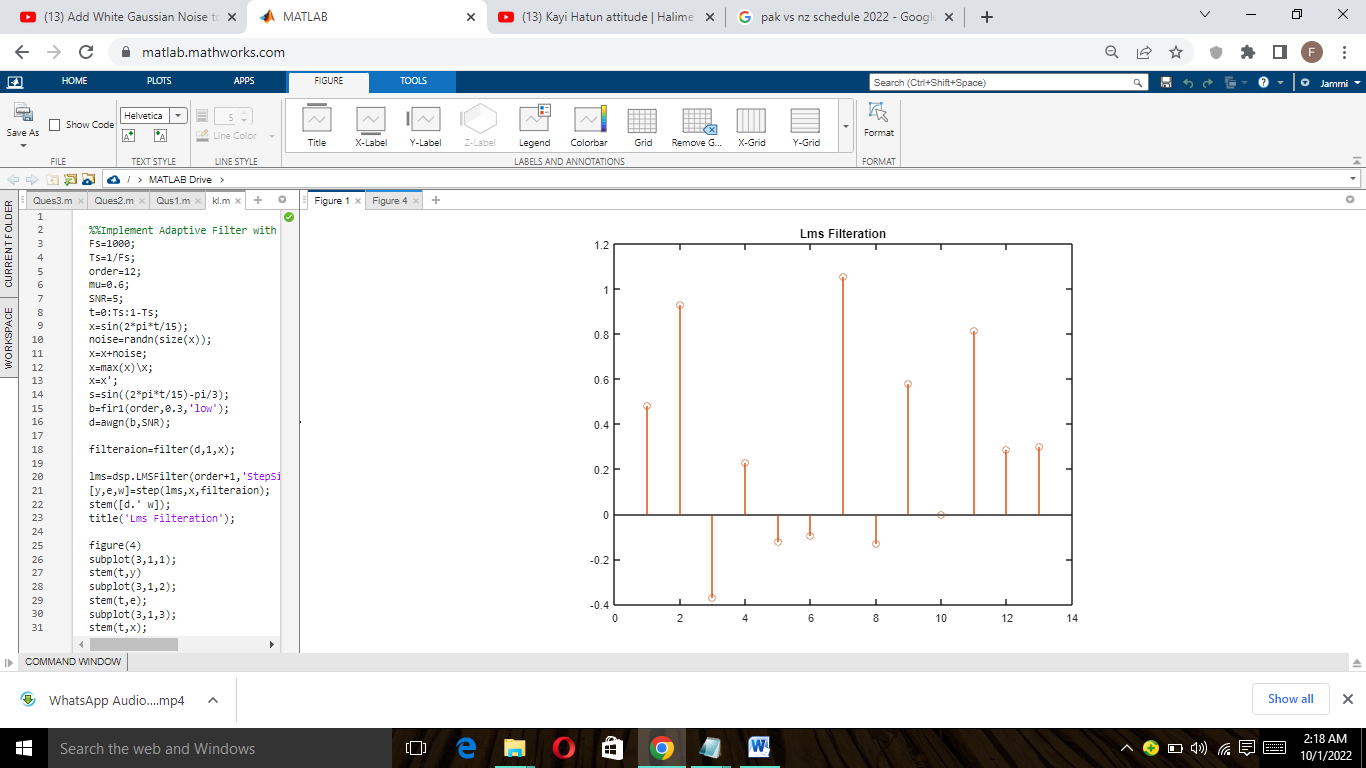
**When SNR=5 and size=0.3**

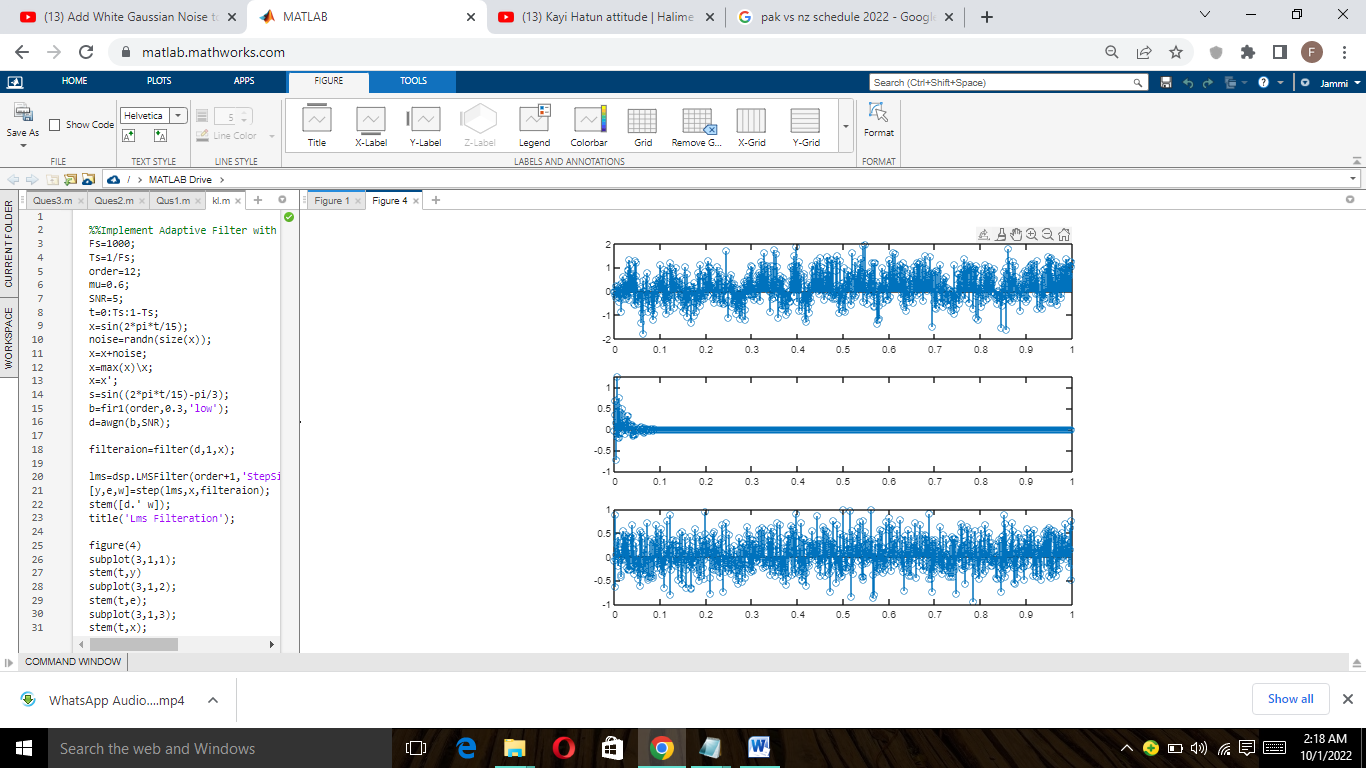




When size of filtering is increase we can easily be observed that LMS values changed and it touch the boundary of 1.5 mean as we increase the size of filtering it effect the LMS values and same effect can be observed in other parameters shown above graph.

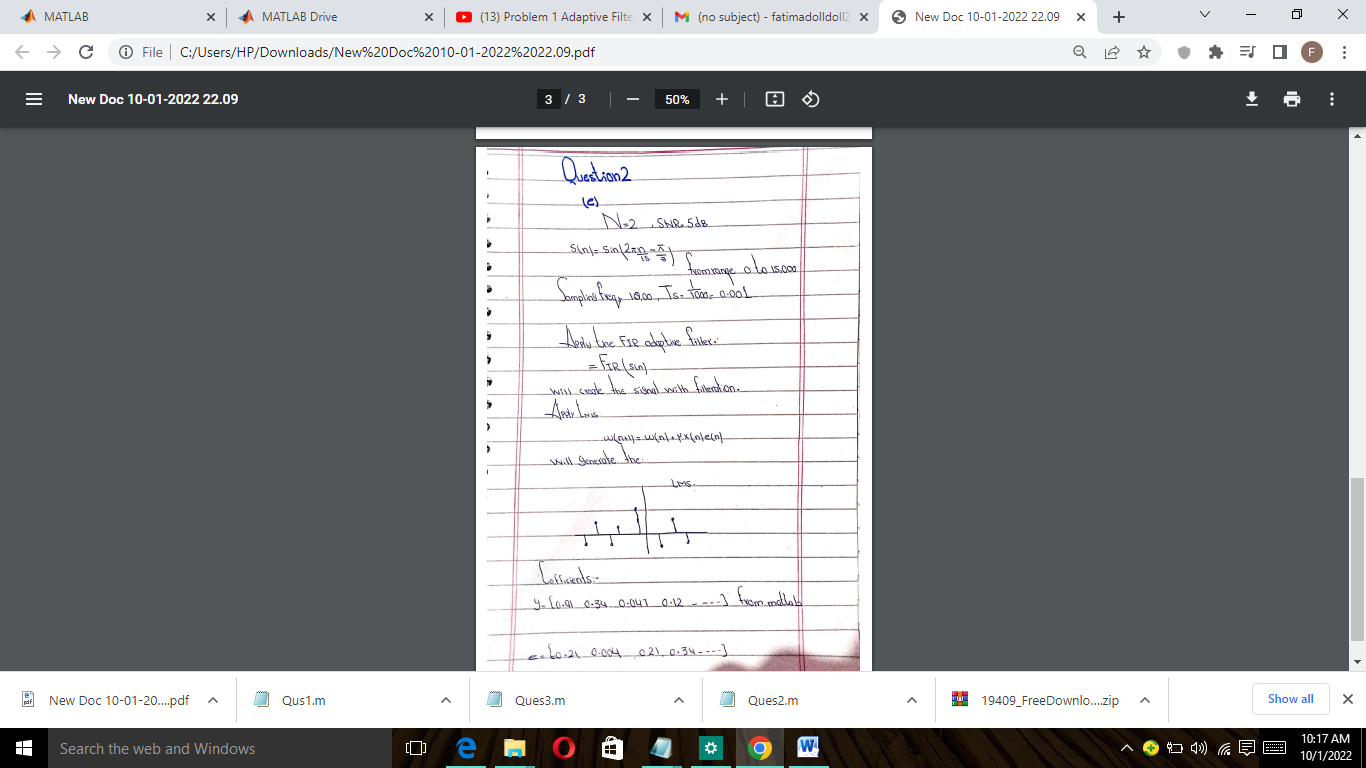
**When SNR=5 and size=0.6**





**Part e)**

**Handwritten:**



**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);

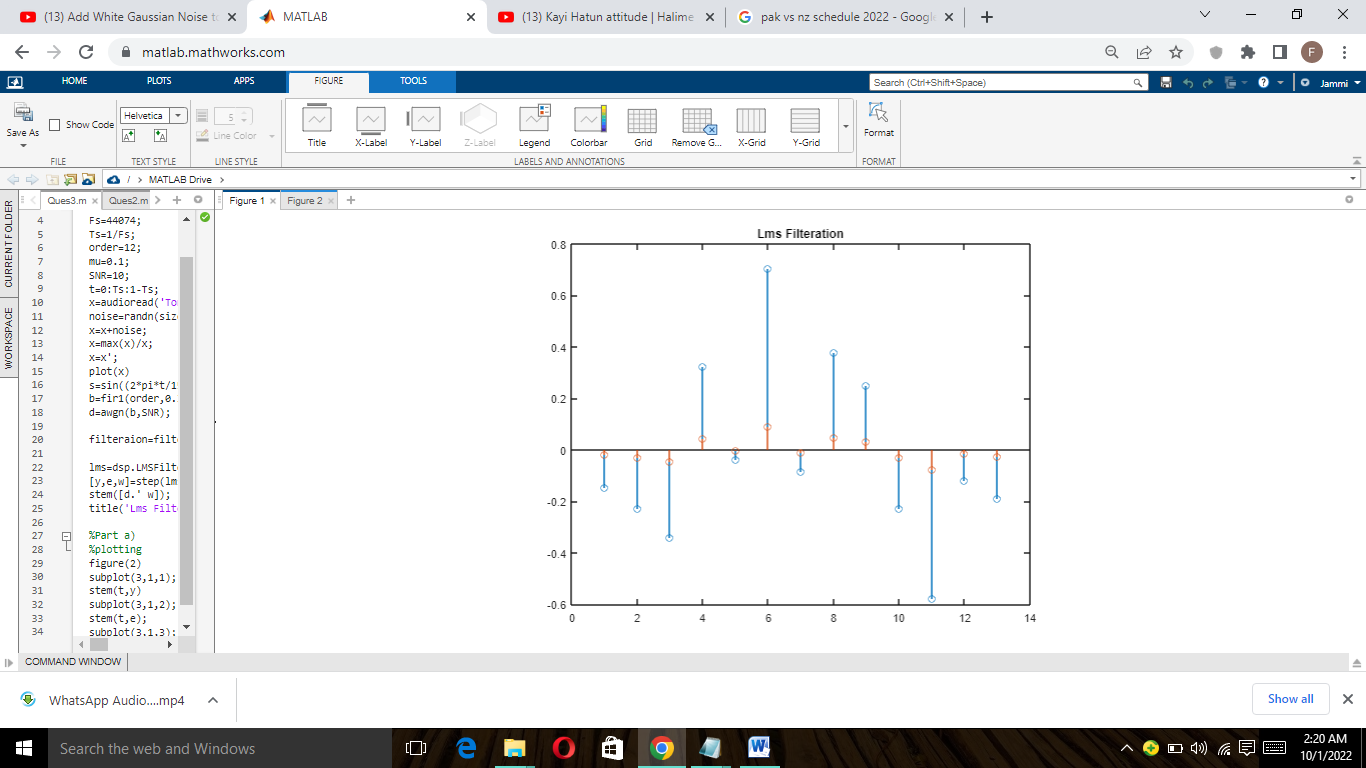
stem(t,y)

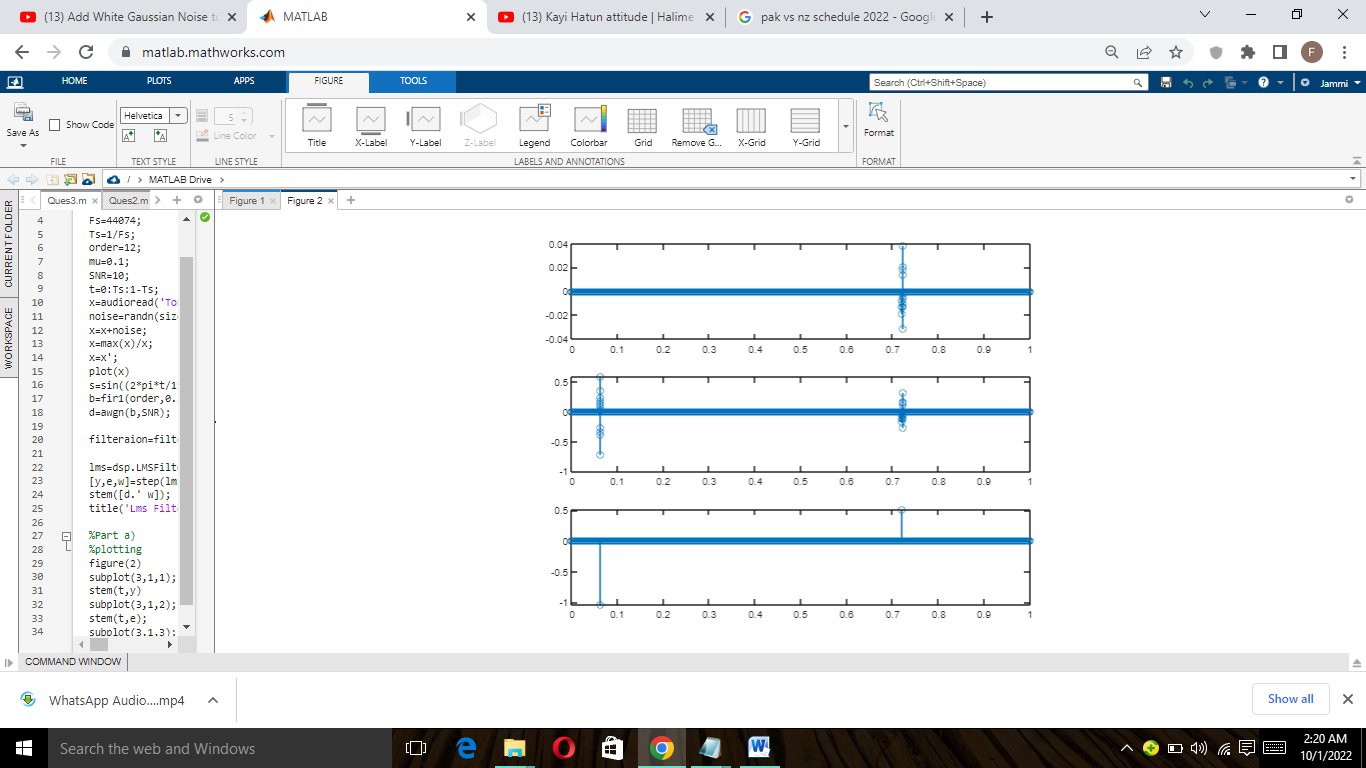
subplot(3,1,2);

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.