

DSP Project.

**Instructor: Dr. Qadri Mayyala**

**Course: DSP – ENCS4310**

First semester 2022/2023

**2/13/2023**

**Introduction** : Finding the system's input-output response relationship, or its equivalent, identifying the system, is the project's major goal. achieving the goal of finding the filter coefficients that fit to a model of the unknown system

**Problem Specification**: we have a desired d[n] output that results from an input x[n] which has entered an unknown system .

## Idea

The fundamental principle underlying the LMS filter is to converge to the optimum filter weights by updating the filter weights in a certain way. The gradient descent algorithm is the foundation of this. The technique begins by assuming low weights (often zero) and updates the weights at each step by determining the gradient of the mean square error. That instance, Equation for the weight update is w(n+1)=w(n)+2µe(n)x(n)

% LMS Algorithm

for i = M:N

e(i) = d(i) - wi\* x(i:-1:i-M+1)';

w = wi + 2\*mu\*e(i)\*x(i:-1:i-M+1)

wi = w;

J(i) = (e(i))^2;

Jlog(i) = 10\*log(J(i));

end

For adaptive filters used in noise cancellation and many other applications, NLMS is the best choice. By setting the parameters properly, noise cancellation can be accomplished. The effectiveness of the NLMS filter is utilized in this paper to reduce noise in sinusoidal signals, Equation for the weight update is

w(n+1)=w(n)+2µe(n)x(n) / (epsilon + norm(x(i:-1:i-M+1))^2);

**where** epsilon = 1e-9;

**references :** [**https://asp-eurasipjournals.springeropen.com/articles/10.1186/s13634-015-0283-1**](https://asp-eurasipjournals.springeropen.com/articles/10.1186/s13634-015-0283-1)

**------------------------------A-------------------------**

%% A) Generating x[n]

N = 2000; % Number of Samples

n = 0:1:N-1;

x = cos(0.03\*pi\*n); %Input Signal

%% A Plotting x[n]

fs = 1/N;

f = linspace(-N\*fs/2, N\*fs/2, length(x)); % Frequency

M = fftshift(fft(x))./ length(x);

figure();

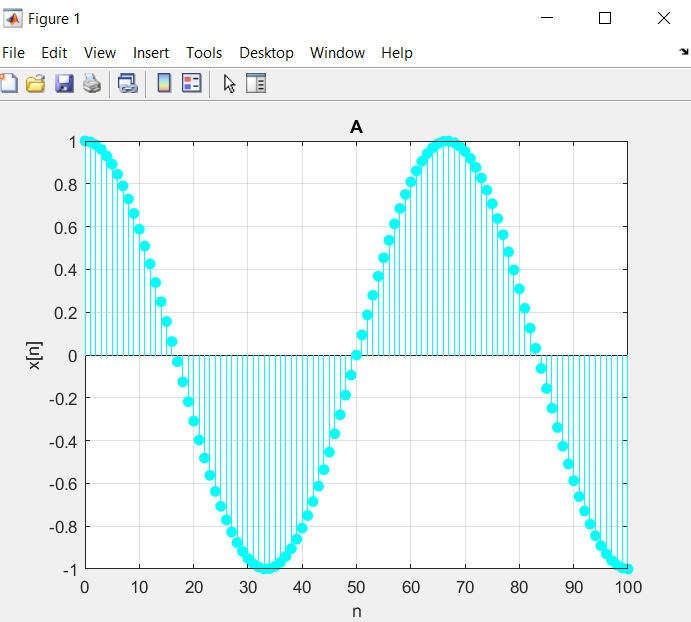
subplot();

stem(n, x,"filled",'c');

grid on;

title ('A');

xlabel('n');ylabel('x[n]'); xlim([0 100]);



**------------------------------B-------------------------**

B)) Plot the amplitude and phase response for the given FIR system.  
  
z = tf('z',0.1);

H = (4\*(z^-2)-2\*(z^-1)+1) /1

bode(H)

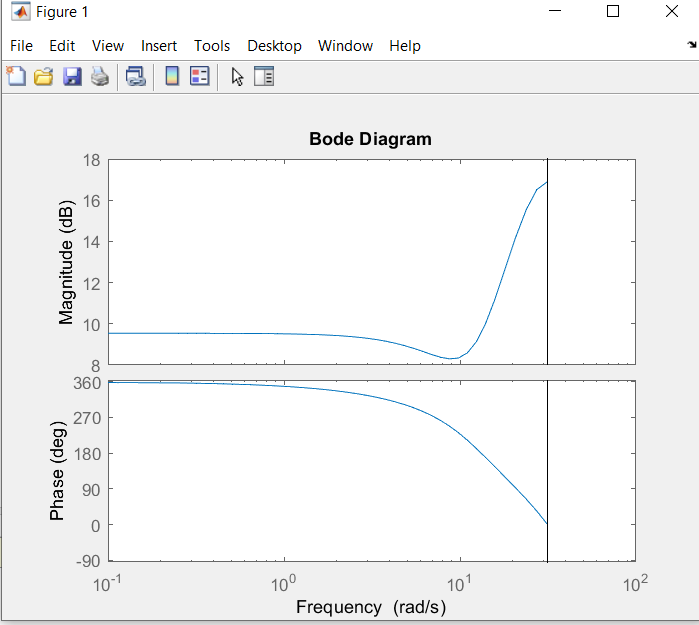
% H = z^3 - 2 z^2 + 4 z

% -----------------

% z^3

%Sample time: 0.1 seconds

%Discrete-time transfer function.



**------------------------------C-------------------------**

%% C) Plotting spectrum of x[n]

N = 2000; % Number of Samples

n = 0:1:N-1;

x = cos(0.03\*pi\*n); %Input Signal

fs = 1/N;

f = linspace(-N\*fs/2, N\*fs/2, length(x));

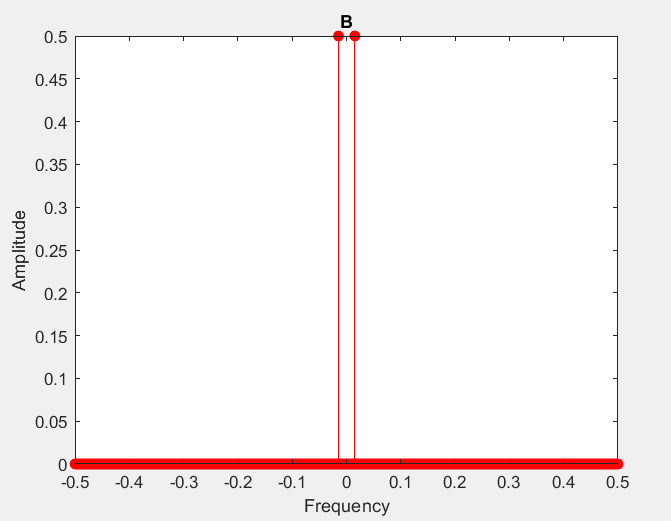
M = fftshift(fft(x))./ length(x);

figure();

subplot();

stem(f, abs(M),'r','filled');

title ('B'); xlabel('Frequency');ylabel('Amplitude');



**------------------------------D-------------------------**

%% D) LMS Algorithm

d = filter(filter1, 1, x); % Applying the FIR Filter to obtain d[n]

M = 4;

w = zeros(1,M); % the updated filter coefficients

wi = zeros(1,M); % Intial Weights

e = [];

mu = 0.01; % learning parameter

% LMS Algorithm

for i = M:N

e(i) = d(i) - wi\* x(i:-1:i-M+1)';

w = wi + 2\*mu\*e(i)\*x(i:-1:i-M+1);

wi = w;

J(i) = (e(i))^2;

Jlog(i) = 10\*log(J(i));

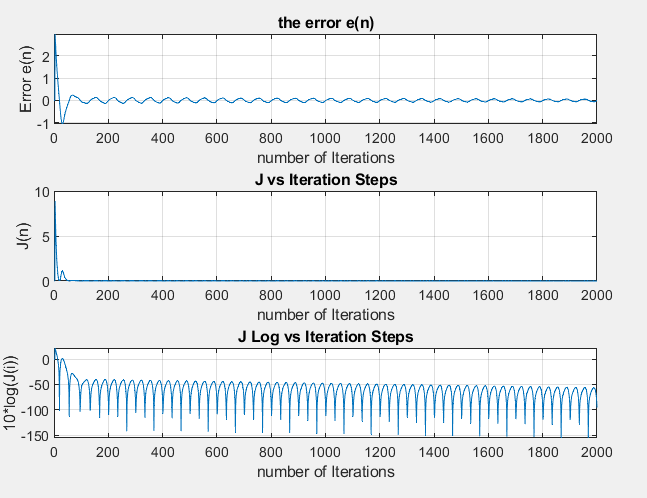
end

figure(3);

subplot(3, 1, 1), plot(e); grid; title('the error e(n)'); xlabel('number of Iterations');ylabel('Error e(n)');

subplot(3,1,2), plot(J); grid; title('J vs Iteration Steps'); xlabel('number of Iterations');ylabel('J(n)');

subplot(3,1,3), plot(Jlog); title('J Log vs Iteration Steps'); grid; xlabel('number of Iterations');ylabel('10\*log(J(i))');



**------------------------------E-------------------------**

%% E) Frequency Response of the obtained Filter

figure(1);

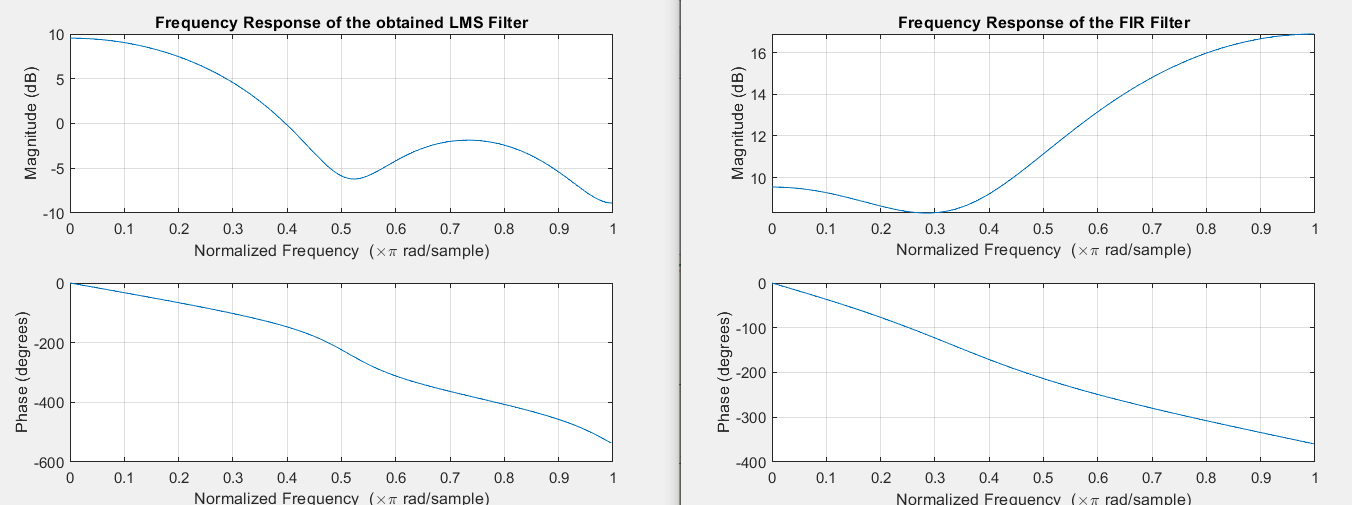
freqz(w,1);

title('Frequency Response of the obtained LMS Filter');

figure(2);

freqz(filter1,1); title('Frequency Response of the FIR Filter');

% They have the same phase respone. with different magnitude



**------------------------------F-------------------------**

%% F) LMS Algorithm Decreasing mu value

w1 = zeros(1,M); % the updated filter coefficients

wi1 = zeros(1,M); % Intial Weights with zeros

e1 = [];

mu1 = 0.0001; % learning parameter with different value

% LMS Algorithm

for i = M:N

e1(i) = d(i) - wi1\* x(i:-1:i-M+1)';

w1 = wi1 + 2\*mu1\*e1(i)\*x(i:-1:i-M+1);

wi1 = w1;

J1(i) = (e1(i))^2;

Jlog1(i) = 10\*log(J1(i));

end

figure();

subplot(3, 1, 1), plot(e1); grid; title('Error Signal mu = 0.0001'); xlabel('number of Iterations');ylabel('Thr error e(n)');

subplot(3,1,2), plot(J1); grid; title('J vs Iteration Step mu = 0.0001'); xlabel('number of Iterations');ylabel('J(n)');

subplot(3,1,3), plot(Jlog1); title('J Log vs Iteration Step mu = 0.0001'); grid; xlabel('number of Iterations');ylabel('10\*log(J(i))');

figure();

freqz(w1,1);

title('Frequency Response LMS Filter mu = 0.0001');

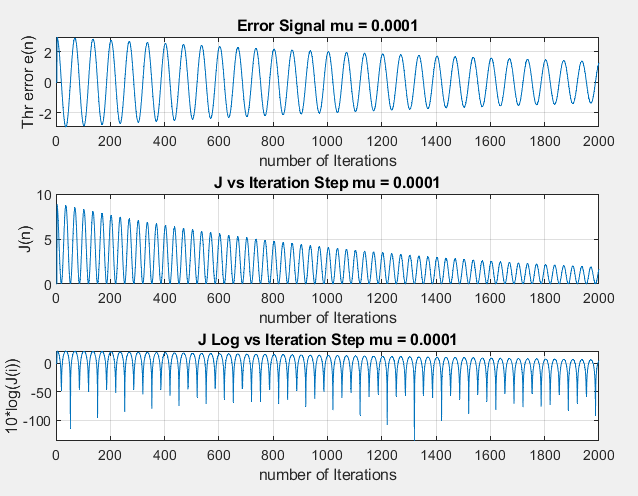
figure();

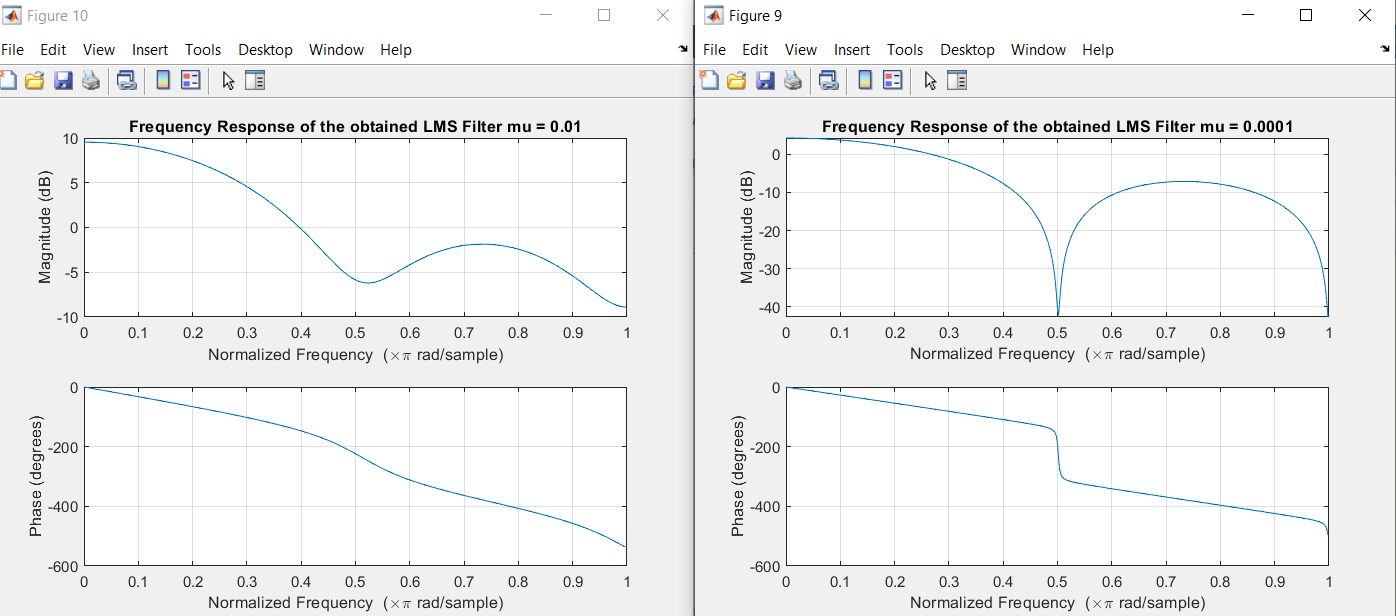
freqz(w,1); title('Frequency Response of the obtained LMS Filter mu = 0.01');

% They have the similar Frequency respone.

% Actually, deceasing the mu value will slower the learning rate which in

% turn increase the steady state error as shown in the error signal figure.





**------------------------------G-------------------------**

%% G) Add 40dB Noise using awgn to x[n]

xn = x;

x = awgn(xn, 40);

fs = 1/N;

f = linspace(-N\*fs/2, N\*fs/2, length(x));

M = fftshift(fft(x))./ length(x);

% repeat D) LMS Algorithm

d = filter(filter1, 1, x); % Applying the FIR Filter to obtain d[n]

M = 4;

w = zeros(1,M);

wi = zeros(1,M);

e = [];

mu = 0.01;

% LMS Algorithm

for i = M:N

e(i) = d(i) - wi\* x(i:-1:i-M+1)';

w = wi + 2\*mu\*e(i)\*x(i:-1:i-M+1);

wi = w;

J(i) = (e(i))^2;

Jlog(i) = 10\*log(J(i));

end

% Plot the requested Signals (e, J);

figure(12);

subplot(3, 1, 1), plot(e); grid; title('Error Signal'); xlabel('Iteration');ylabel('Error e(n)');

subplot(3,1,2), plot(J); grid; title('J vs Iteration Step'); xlabel('Iteration');ylabel('J(n)');

subplot(3,1,3), plot(Jlog); title('J Log vs Iteration Step'); grid; xlabel('Iteration');ylabel('10\*log(J(i))');

% repeat E) Frequency Response of the obtained Filter

figure(13);

freqz(w,1);

title('Frequency Response of the obtained LMS Filter');

figure(14);

freqz(filter1,1); title('Frequency Response of the FIR Filter');

% They have the same phase respone.

% repeat F) LMS Algorithm Decreasing mu value 0.0001

w1 = zeros(1,M);

wi1 = zeros(1,M);

e1 = []; %Error

mu1 = 0.0001;

% LMS Algorithm

for i = M:N

e1(i) = d(i) - wi1\* x(i:-1:i-M+1)';

w1 = wi1 + 2\*mu1\*e1(i)\*x(i:-1:i-M+1);

wi1 = w1;

J1(i) = (e1(i))^2;

Jlog1(i) = 10\*log(J1(i));

end

figure(15);

subplot(3, 1, 1), plot(e1); grid; title('Error Signal when mu = 0.0001'); xlabel('number of Iterations');ylabel('the error e(n)');

subplot(3,1,2), plot(J1); grid; title('J vs Iteration Step mu = 0.0001'); xlabel('number of Iterations');ylabel('J(n)');

subplot(3,1,3), plot(Jlog1); title('J Log vs Iteration Step mu = 0.0001'); grid; xlabel('number of Iterations');ylabel('10\*log(J(i))');

figure(16);

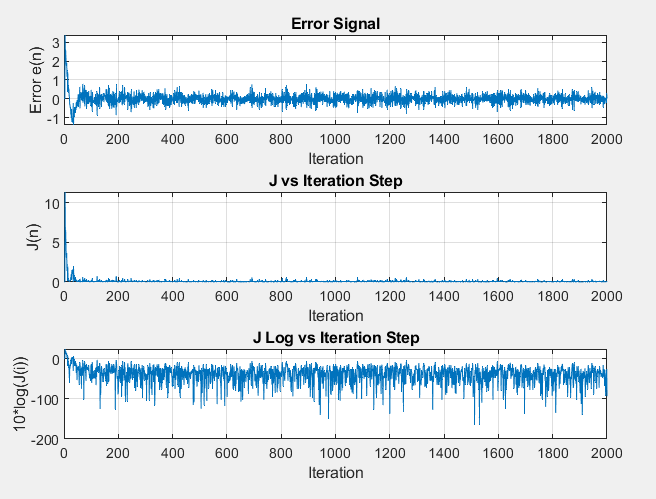
freqz(w1,1);

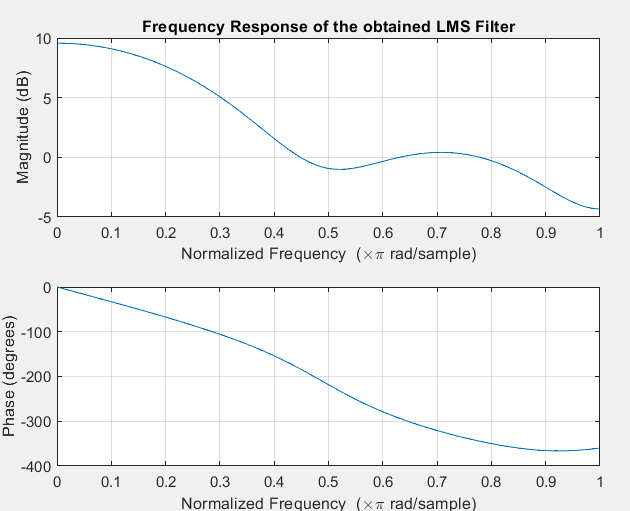
title('Frequency Response of the obtained LMS Filter mu = 0.0001');

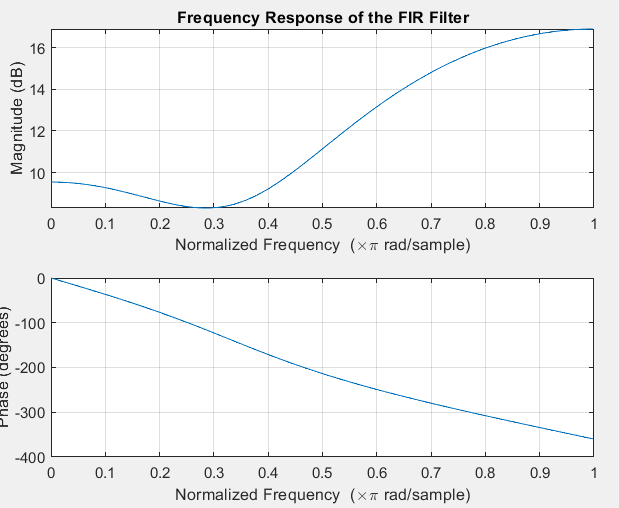
figure(16);

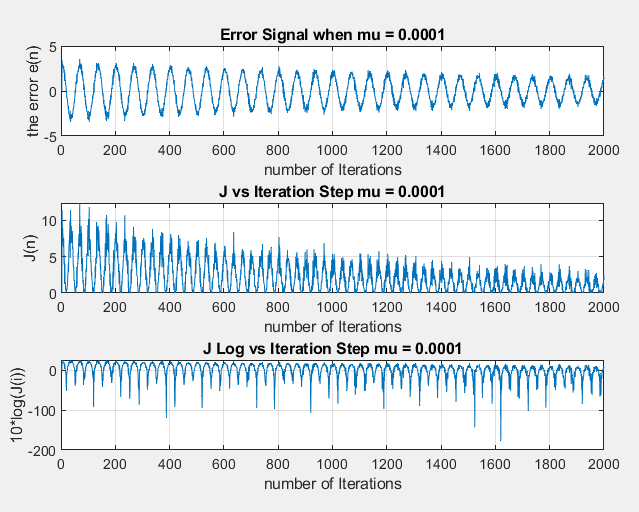
freqz(w,1); title('Frequency Response of the obtained LMS Filter mu = 0.01');

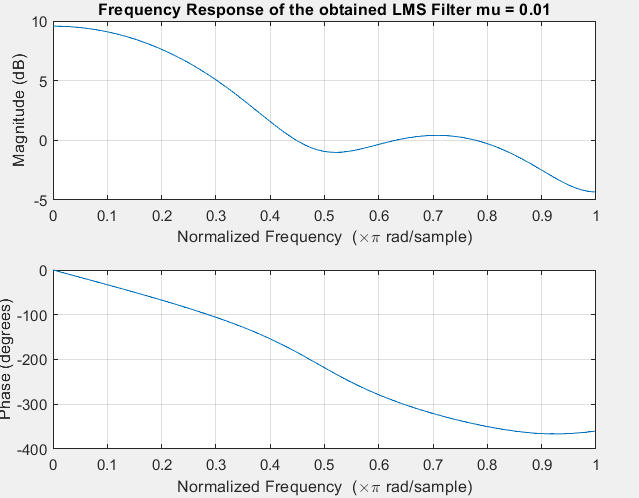
% They have the similar Frequency response.











%% H) Add 30dB Noise using awgn to x[n]

x = awgn(xn, 30);

fs = 1/N;

f = linspace(-N\*fs/2, N\*fs/2, length(x));

M = fftshift(fft(x))./ length(x);

% repeat D) LMS Algorithm

d = filter(filter1, 1, x);

M = 4;

w = zeros(1,M);

wi = zeros(1,M);

e = [];

mu = 0.01;

% LMS Algorithm

for i = M:N

e(i) = d(i) - wi\* x(i:-1:i-M+1)';

w = wi + 2\*mu\*e(i)\*x(i:-1:i-M+1);

wi = w;

J(i) = (e(i))^2;

Jlog(i) = 10\*log(J(i));

end

figure();

subplot(3, 1, 1), plot(e); grid; title('Error Signal'); xlabel('number of Iterations');ylabel('Error e(n)');

subplot(3,1,2), plot(J); grid; title('J vs Iteration Step'); xlabel('number of Iterations');ylabel('J(n)');

subplot(3,1,3), plot(Jlog); title('J Log vs Iteration Step'); grid; xlabel('number of Iterations');ylabel('10\*log(J(i))');

% repeat E) Frequency Response of the obtained Filter

figure();

freqz(w,1);

title('Frequency Response of LMS Filter');

figure();

freqz(filter1,1); title('Frequency Response FIR Filter');

% They have the same phase respone.

% repeat F) LMS Algorithm Decreasing mu value

w1 = zeros(1,M);

wi1 = zeros(1,M);

e1 = [];

mu1 = 0.0001;

% LMS Algorithm

for i = M:N

e1(i) = d(i) - wi1\* x(i:-1:i-M+1)';

w1 = wi1 + 2\*mu1\*e1(i)\*x(i:-1:i-M+1);

wi1 = w1;

J1(i) = (e1(i))^2;

Jlog1(i) = 10\*log(J1(i));

end

figure();

subplot(3, 1, 1), plot(e1); grid; title('Error Signal mu = 0.0001'); xlabel('number of Iterations');ylabel('Error e(n)');

subplot(3,1,2), plot(J1); grid; title('J vs Iteration Step mu = 0.0001'); xlabel('number of Iterations');ylabel('J(n)');

subplot(3,1,3), plot(Jlog1); title('J Log vs Iteration Step mu = 0.0001'); grid; xlabel('number of Iterations');ylabel('10\*log(J(i))');

figure();

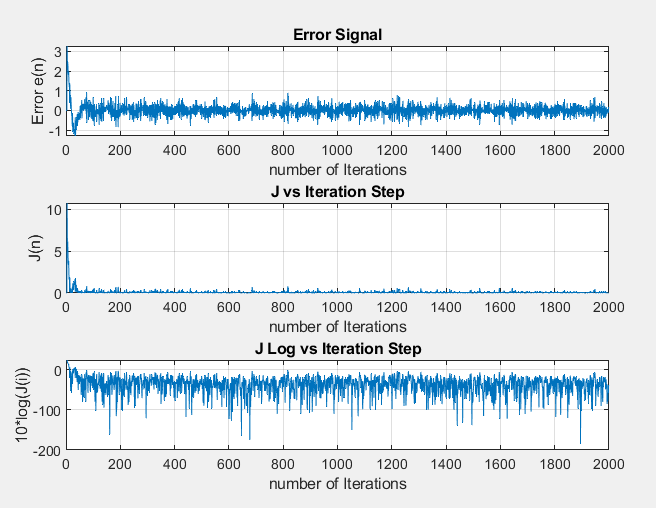
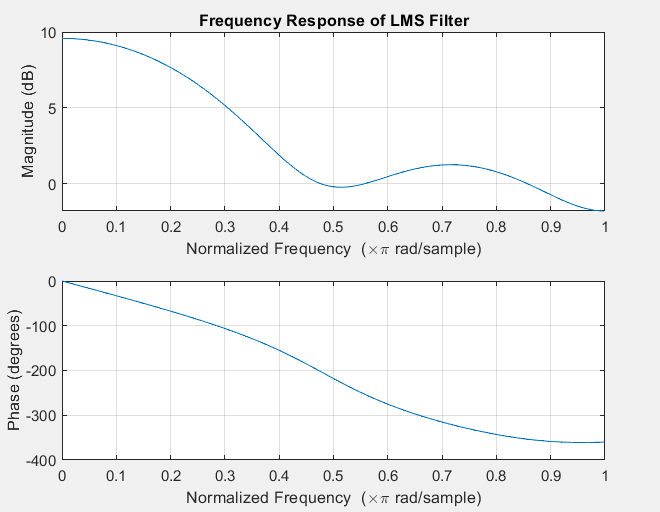
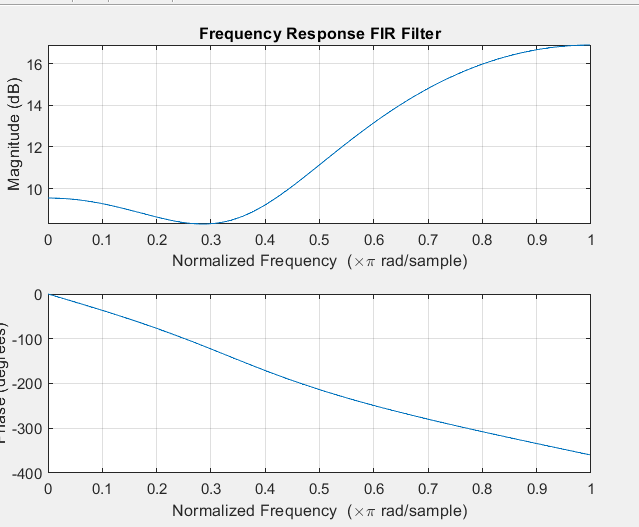
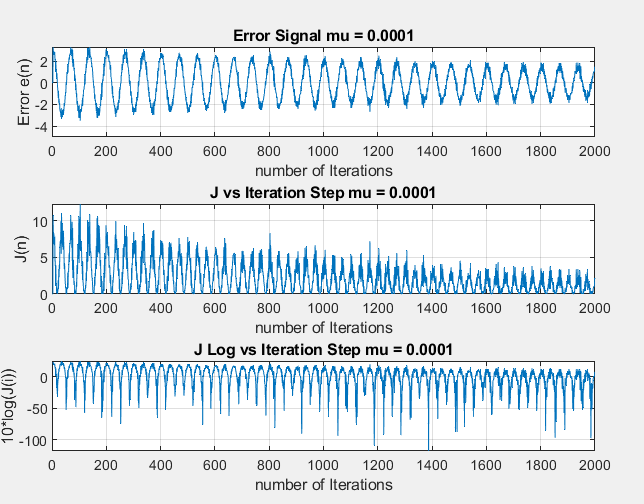
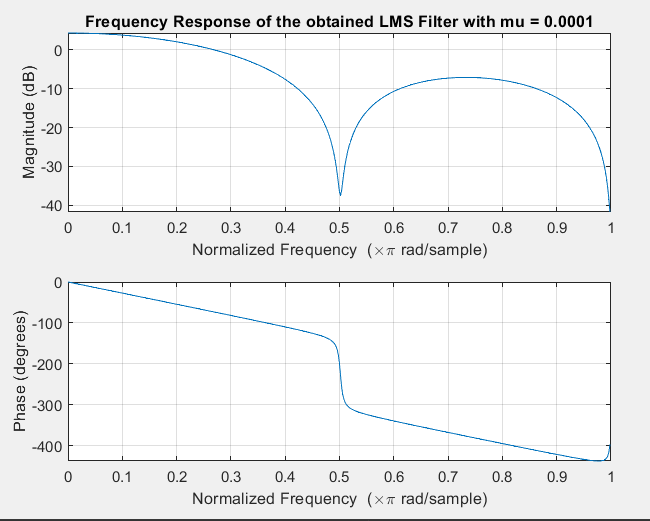
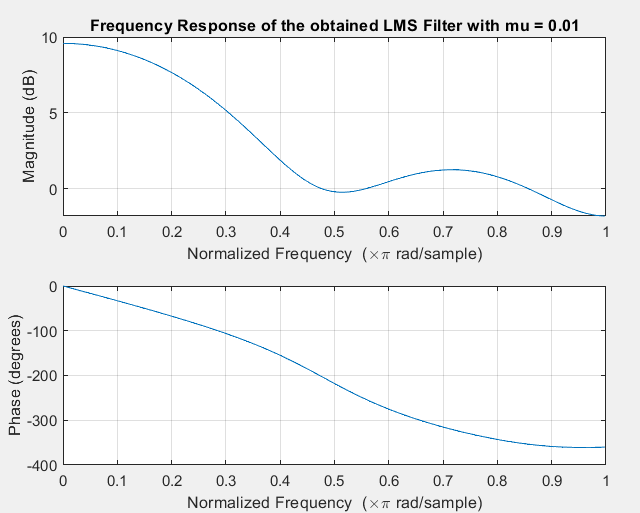
freqz(w1,1);

title('Frequency Response of the obtained LMS Filter with mu = 0.0001');

figure();

freqz(w,1); title('Frequency Response of the obtained LMS Filter with mu = 0.01');

% They have the similar Frequency respone.



%% I ) Ensemble Averaging

x = awgn(xn, 30);

% repeat D) LMS Algorithm

d = filter(filter1, 1, x);

M = 4;

w = zeros(1000,M);

wi = zeros(1000,M);

e = [];

mu = 0.01;

% LMS Algorithm

for k = 1:1:1000

for i = M:N

e(k,i) = d(i) - wi(k,:)\* x(i:-1:i-M+1)';

w(k,:) = wi(k,:) + 2\*mu\*e(k,i)\*x(i:-1:i-M+1);

wi = w;

end

end

% Averaging

e = mean(e);

w = mean(w);

J = e.^2;

Jlog = 10\*log(J);

figure();

subplot(3, 1, 1), plot(e); grid; title('Error Signal'); xlabel('number of Iterations');ylabel('Error e(n)');

subplot(3,1,2), plot(J); grid; title('J vs Iteration Step'); xlabel('number of Iterations');ylabel('J(n)');

subplot(3,1,3), plot(Jlog); title('J Log vs Iteration Step'); grid; xlabel('number of Iterations');ylabel('10\*log(J(i))');

% E) Frequency Response of the obtained Filter

figure();

freqz(w,1);

title('Frequency Response of the obtained LMS Filter');

figure();

freqz(filter1,1); title('Frequency Response of the FIR Filter');

% The steady state error is smaller

% repeat F) LMS Algorithm Decreasing mu value 0.0001

w1 = zeros(1000,M);

wi1 = zeros(1000,M);

e1 = []; %Error

mu1 = 0.0001;

% LMS Algorithm

for k = 1:1:1000 %for 1000 rounds

for i = M:N

e1(k,i) = d(i) - wi1(k,:)\* x(i:-1:i-M+1)';

w1(k,:) = wi1(k,:) + 2\*mu1\*e1(k,i)\*x(i:-1:i-M+1);

wi1 = w1;

end

end

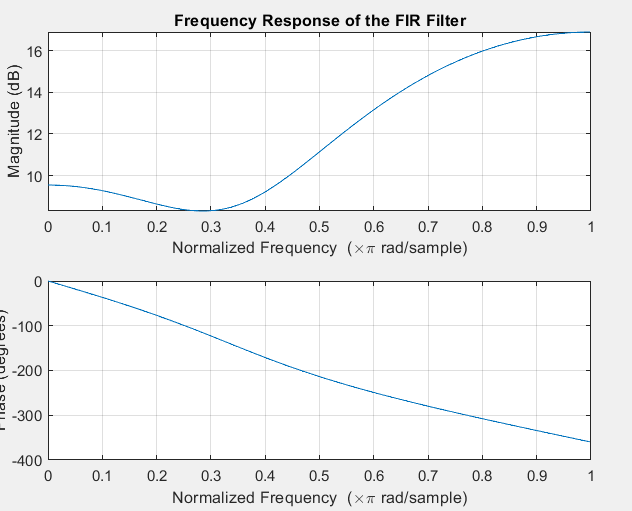
% Averaging

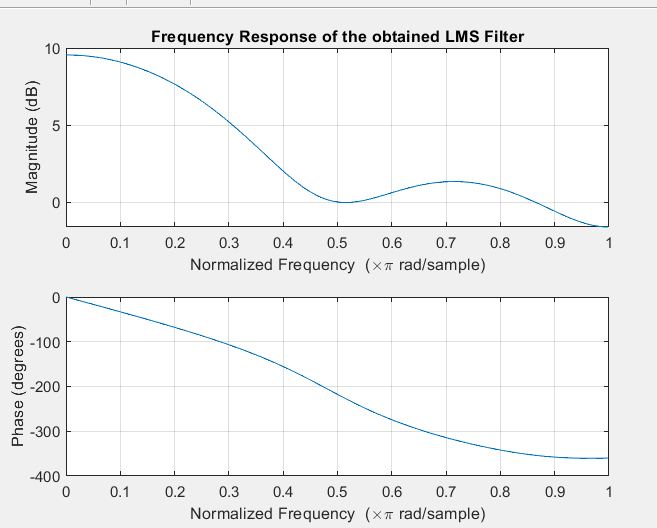
e1 = mean(e1);

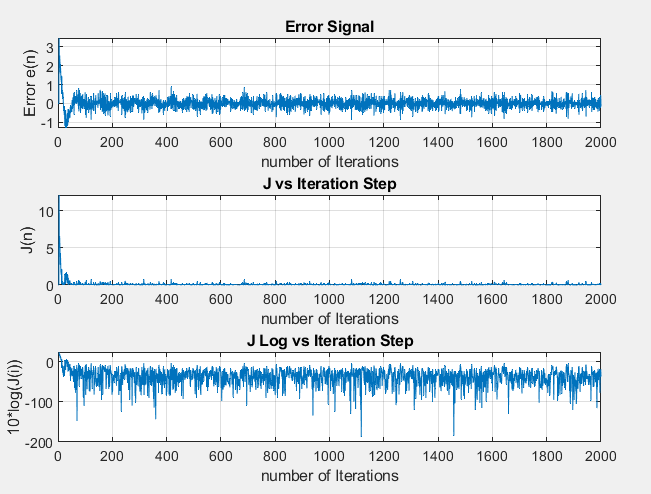
w1 = mean(w1);

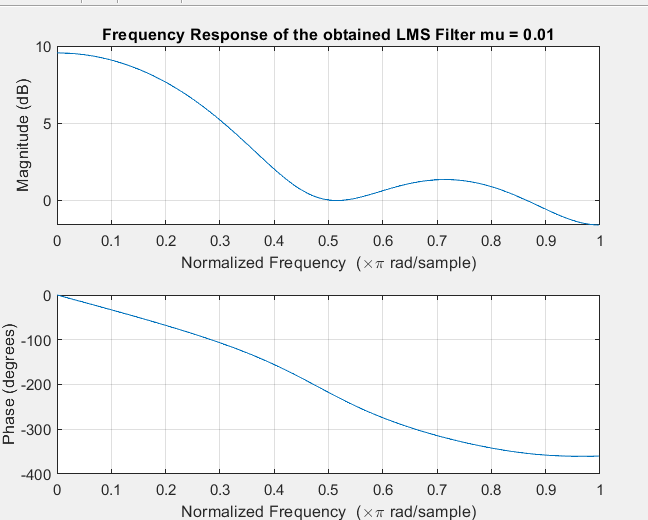
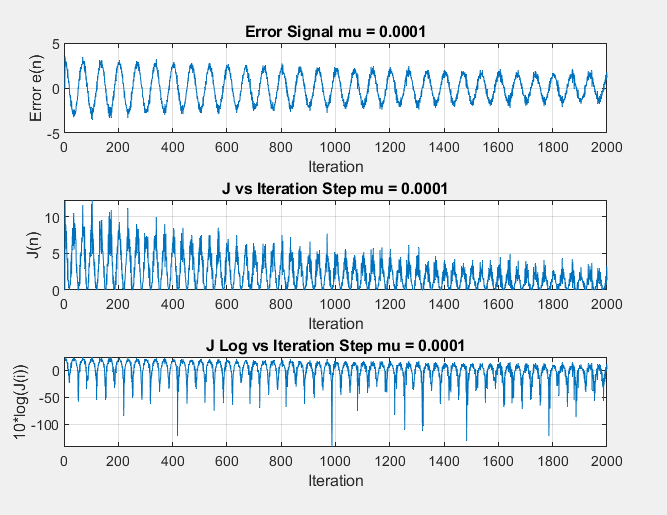
J1 = e1.^2;

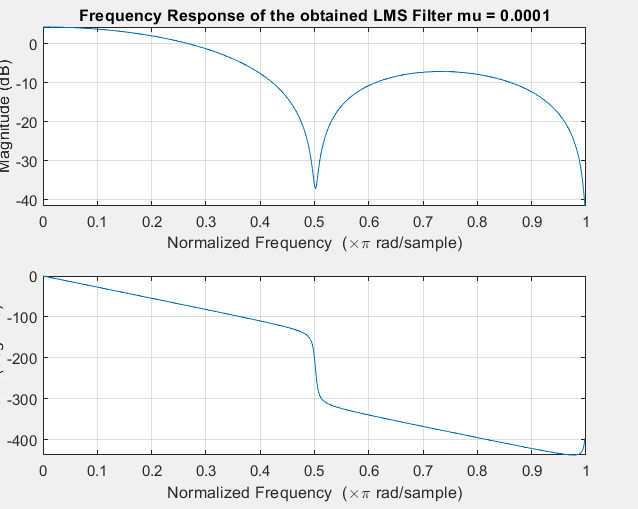
Jlog1 = 10\*log(J1);











**PART2**

Repeating PART A with NLMS Algorithm   
from A to C it’s the same no different

clear;

close all;

%% Generating x[n]

N = 2000; % Number of Samples

n = 0:1:N-1;

x = cos(0.03\*pi\*n); %Input Signal

epsilon = 1e-9;

%% D) LMS Algorithm

d = filter(filter1, 1, x); % Applying the FIR Filter to obtain d[n]

M = 4;

w = zeros(1,M); % the updated filter coefficients

wi = zeros(1,M); % Intial Weights with zeros

e = []; %Error

mu = 0.01;

% LMS Algorithm

for i = M:N

e(i) = d(i) - wi\* x(i:-1:i-M+1)';

w = wi + 2 \* mu \* e(i) \* x(i:-1:i-M+1) / (epsilon + norm(x(i:-1:i-M+1))^2);

wi = w;

J(i) = (e(i))^2;

Jlog(i) = 10\*log(J(i));

end

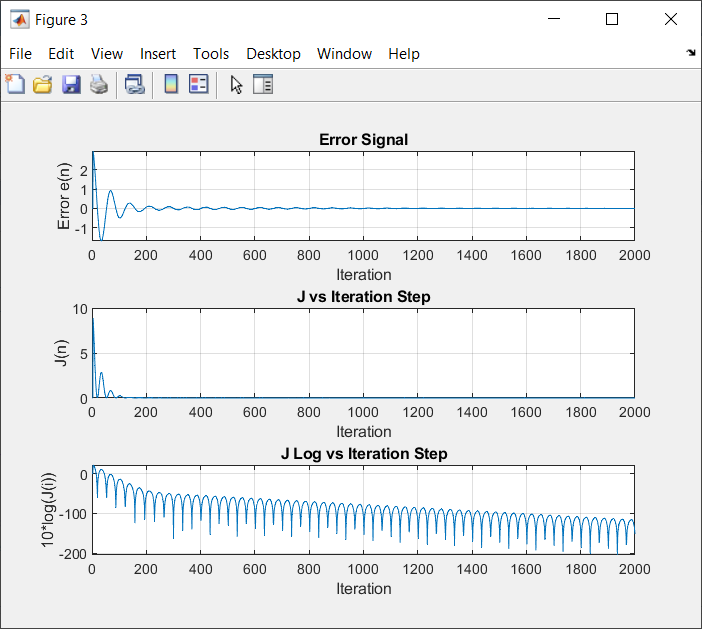
% Plot the requested Signals (e, J);

figure(3);

subplot(3, 1, 1), plot(e); grid; title('Error Signal'); xlabel('number of Iterations');ylabel('the error e(n)');

subplot(3,1,2), plot(J); grid; title('J vs Iteration Steps'); xlabel('number of Iterations');ylabel('J(n)');

subplot(3,1,3), plot(Jlog); title('J Log vs Iteration Steps'); grid; xlabel('number of Iterations');ylabel('10\*log(J(i))');



%% E) Frequency Response of the obtained Filter

figure(4);

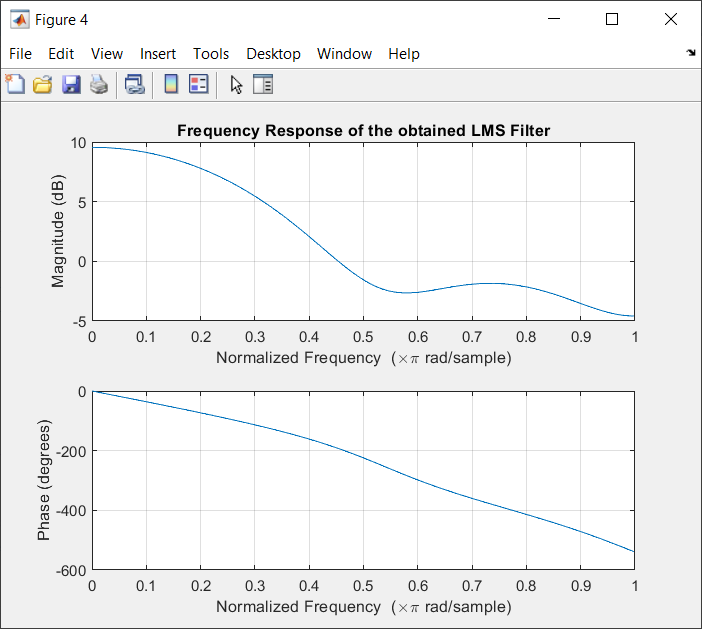
freqz(w,1);

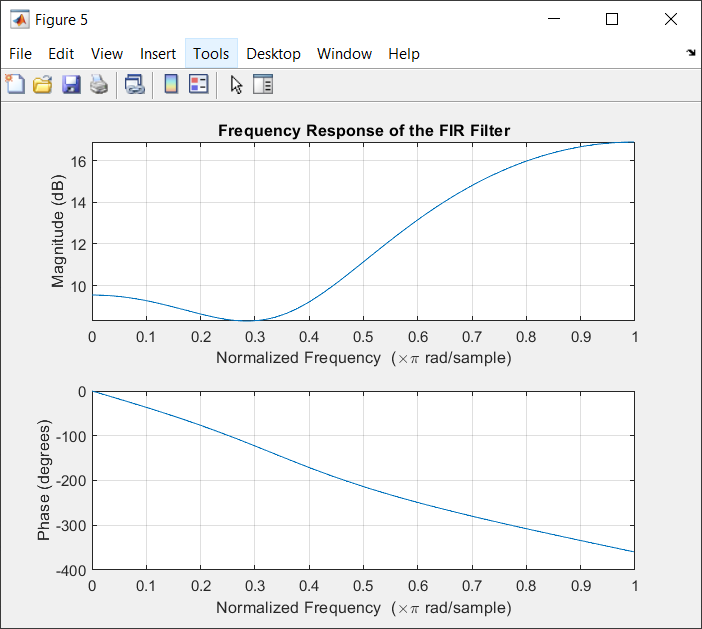
title('Frequency Response of LMS Filter');

figure(5);

freqz(filter1,1); title('Frequency Response of FIR Filter');

% They have the same phase respone.





%% F) LMS Algorithm Decreasing mu value

w1 = zeros(1,M); % the updated filter coefficients

wi1 = zeros(1,M); % Intial Weights with zeros

e1 = []; %Error

mu1 = 0.0001;

% LMS Algorithm

for i = M:N

e1(i) = d(i) - wi1\* x(i:-1:i-M+1)';

w1 = wi1 + 2\*mu1\*e1(i)\*x(i:-1:i-M+1) / (epsilon + norm(x(i:-1:i-M+1))^2);

wi1 = w1;

J1(i) = (e1(i))^2;

Jlog1(i) = 10\*log(J1(i));

end

% Plot the requested Signals (e, J);

figure(8);

subplot(3, 1, 1), plot(e1); grid; title('Error Signal mu = 0.0001'); xlabel('number of Iterations');ylabel('the error e(n)');

subplot(3,1,2), plot(J1); grid; title('J vs Iteration Step mu = 0.0001'); xlabel('number of Iterations');ylabel('J(n)');

subplot(3,1,3), plot(Jlog1); title('J Log vs Iteration Step mu = 0.0001'); grid; xlabel('number of Iterations');ylabel('10\*log(J(i))');

figure(9);

freqz(w1,1);

title('Frequency Response of the obtained LMS Filter mu = 0.0001');

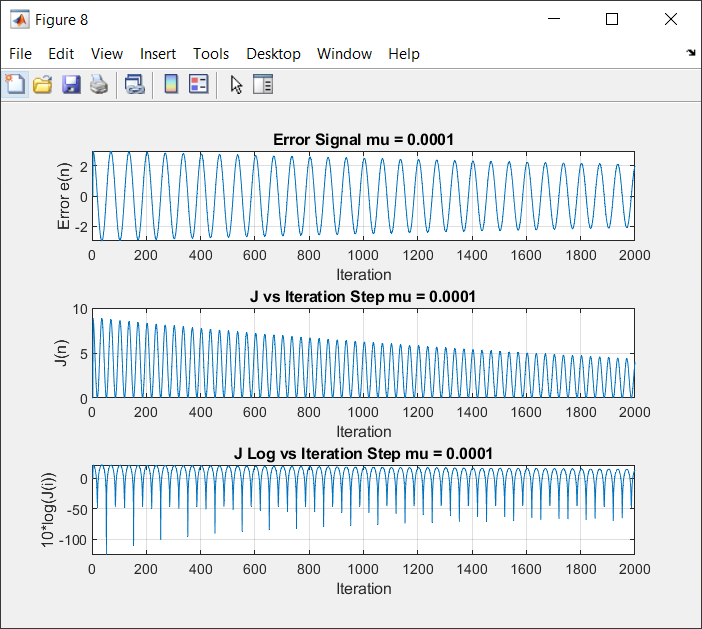
figure(10);

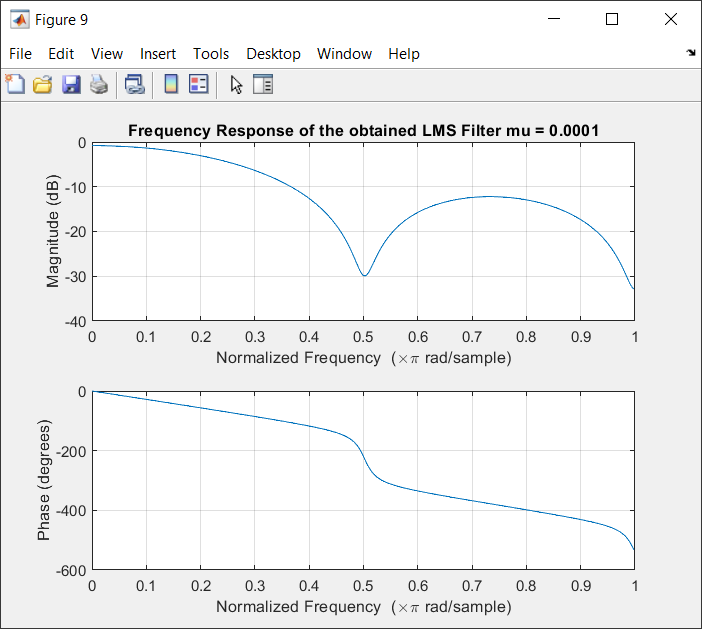
freqz(w,1); title('Frequency Response of the obtained LMS Filter mu = 0.01');

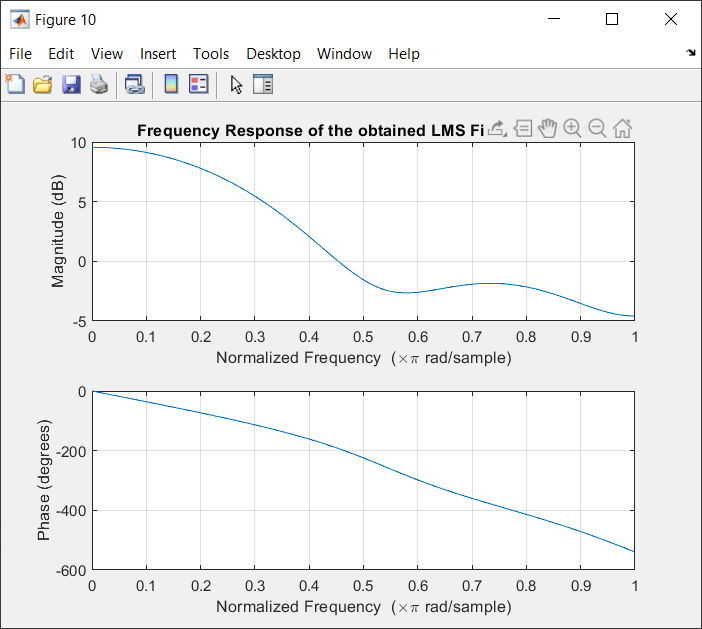
% They have the similar Frequency respone.

% Actually, deceasing the mu value will slower the learning rate which in

% turn increase the steady state error as shown in the error signal figure.







%% G) Add 40dB Noise using awgn to x[n]

xn = x;

x = awgn(xn, 40);

fs = 1/N;

f = linspace(-N\*fs/2, N\*fs/2, length(x)); % Frequency

M = fftshift(fft(x))./ length(x);

% D) LMS Algorithm

d = filter(filter1, 1, x); % Applying the FIR Filter to obtain d[n]

M = 4;

w = zeros(1,M); % the updated filter coefficients

wi = zeros(1,M); % Intial Weights with zeros

e = []; %Error

mu = 0.01;

% LMS Algorithm

for i = M:N

e(i) = d(i) - wi\* x(i:-1:i-M+1)';

w = wi + 2\*mu\*e(i)\*x(i:-1:i-M+1) / (epsilon + norm(x(i:-1:i-M+1))^2);

wi = w;

J(i) = (e(i))^2;

Jlog(i) = 10\*log(J(i));

end

% Plot the requested Signals (e, J);

figure(12);

subplot(3, 1, 1), plot(e); grid; title('Error Signal'); xlabel('number of Iterations');ylabel('the error e(n)');

subplot(3,1,2), plot(J); grid; title('J vs Iteration Steps'); xlabel('number of Iterations');ylabel('J(n)');

subplot(3,1,3), plot(Jlog); title('J Log vs Iteration Steps'); grid; xlabel('number of Iterations');ylabel('10\*log(J(i))');

% E) Frequency Response of the obtained Filter

figure(13);

freqz(w,1);

title('Frequency Response of LMS Filter');

figure(14);

freqz(filter1,1); title('Frequency Response of FIR Filter');

% They have the same phase respone.

% F) LMS Algorithm Decreasing mu value

w1 = zeros(1,M); % the updated filter coefficients

wi1 = zeros(1,M); % Intial Weights with zeros

e1 = []; %Error

mu1 = 0.0001;

% LMS Algorithm

for i = M:N

e1(i) = d(i) - wi1\* x(i:-1:i-M+1)';

w1 = wi1 + 2\*mu1\*e1(i)\*x(i:-1:i-M+1) / (epsilon + norm(x(i:-1:i-M+1))^2);

wi1 = w1;

J1(i) = (e1(i))^2;

Jlog1(i) = 10\*log(J1(i));

end

% Plot the requested Signals (e, J);

figure(15);

subplot(3, 1, 1), plot(e1); grid; title('Error Signal mu = 0.0001'); xlabel('number of Iterations');ylabel('the error e(n)');

subplot(3,1,2), plot(J1); grid; title('J vs Iteration Step mu = 0.0001'); xlabel('number of Iterations');ylabel('J(n)');

subplot(3,1,3), plot(Jlog1); title('J Log vs Iteration Step mu = 0.0001'); grid; xlabel('number of Iterations');ylabel('10\*log(J(i))');

figure(16);

freqz(w1,1);

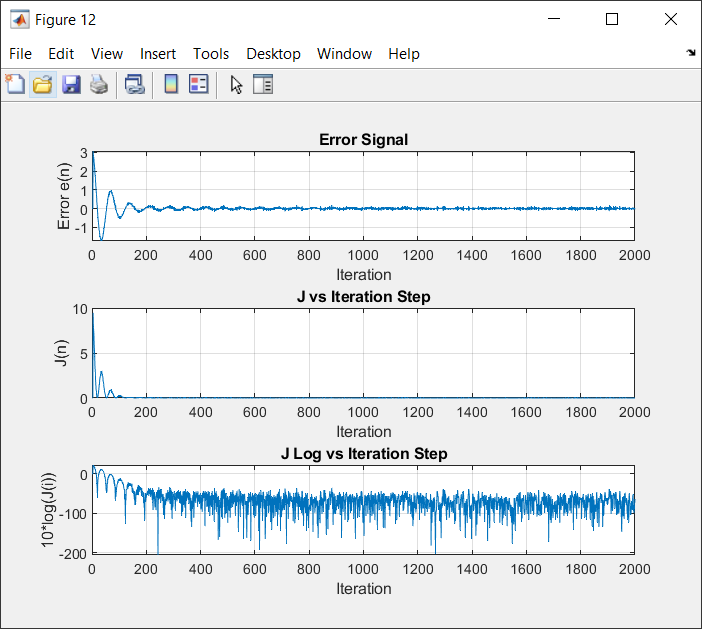
title('Frequency Response of the obtained LMS Filter mu = 0.0001');

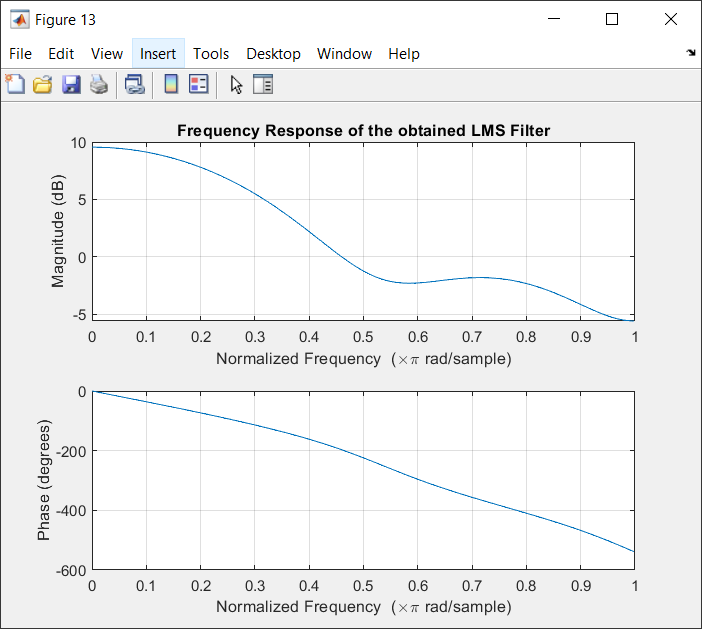
figure(16);

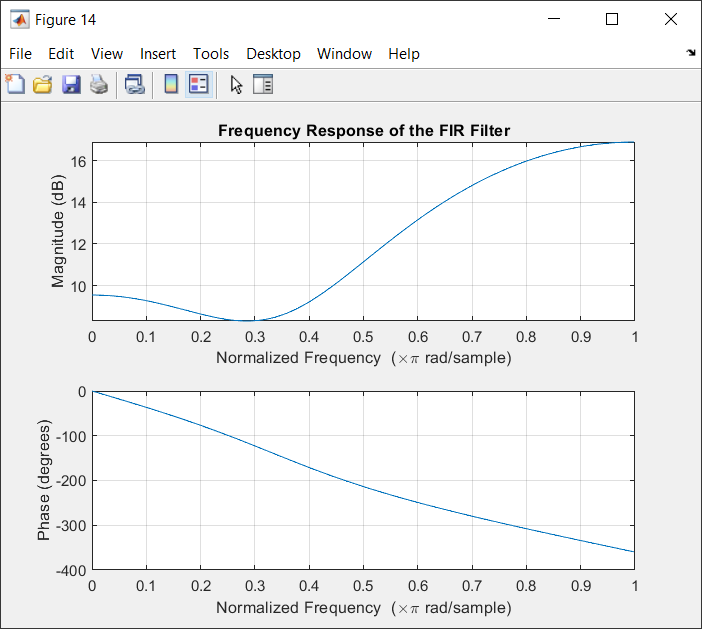
freqz(w,1); title('Frequency Response of the obtained LMS Filter mu = 0.01');

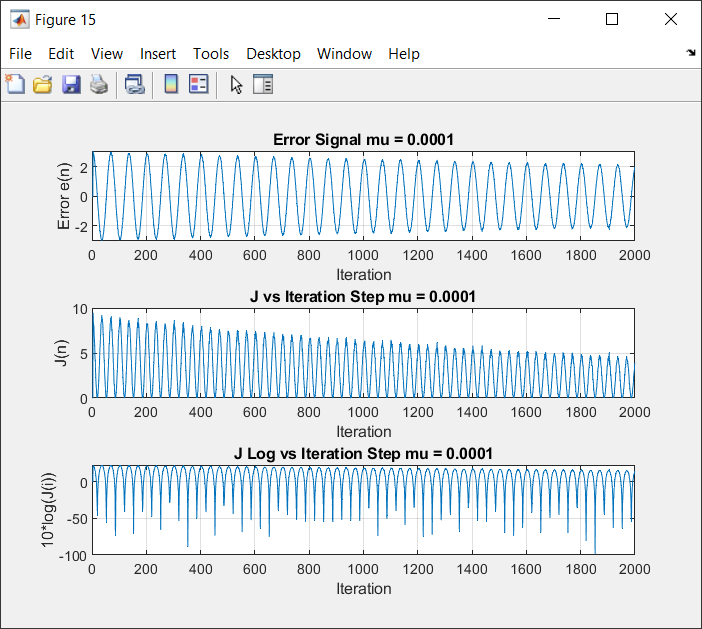
% They have the similar Frequency respone.

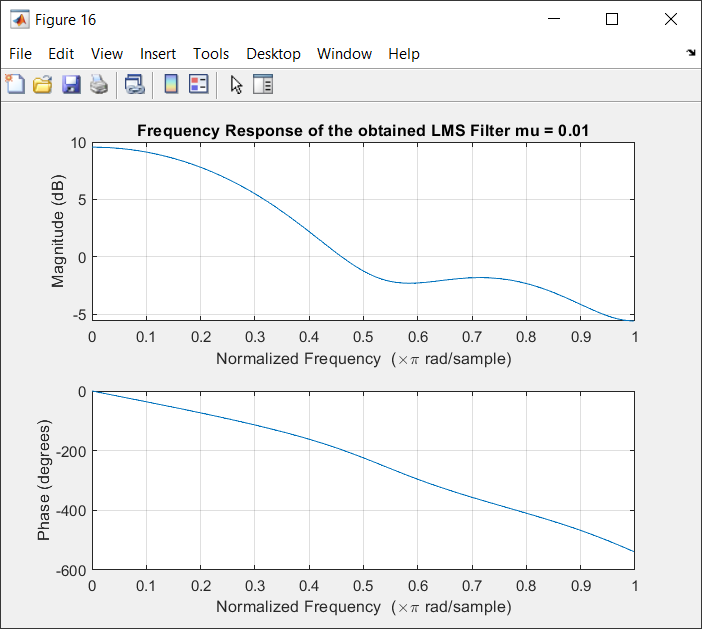
% Actually, adding awgn leads to an increase in the error and the steady state error as well as shown in the error signal figure.











%% H) Add 30dB Noise using awgn to x[n]

x = awgn(xn, 30);

% A, C) Plotting x[n] and its spectrum

fs = 1/N;

f = linspace(-N\*fs/2, N\*fs/2, length(x)); % Frequency Vector to plot

%Obtaining Fourier Transform for both x[n] signals.

M = fftshift(fft(x))./ length(x);

%Plotting the x[n] signal and its magnitude and phase spectrum.

figure(17);

subplot(3,1,1);

stem(n, x);

title ('x[n] signal');

xlabel('n');ylabel('x[n]'); xlim([0 200]);

subplot(3,1,2);

plot(f,abs(M));

title ('Magnitude Spectrum of the signal M(jw)'); xlabel('Frequency');ylabel('Amplitude');

subplot(3,1,3);

plot(f,(angle(M).\*(180/pi)));

title ('Phase Spectrum of the signal M(jw)'); xlabel('Frequency');ylabel('Angle');

% D) LMS Algorithm

d = filter(filter1, 1, x); % Applying the FIR Filter to obtain d[n]

M = 4;

w = zeros(1,M); % the updated filter coefficients

wi = zeros(1,M); % Intial Weights with zeros

e = []; %Error

mu = 0.01;

% LMS Algorithm

for i = M:N

e(i) = d(i) - wi\* x(i:-1:i-M+1)';

w = wi + 2\*mu\*e(i)\*x(i:-1:i-M+1) / (epsilon + norm(x(i:-1:i-M+1))^2);

wi = w;

J(i) = (e(i))^2;

Jlog(i) = 10\*log(J(i));

end

% Plot the requested Signals (e, J);

figure(18);

subplot(3, 1, 1), plot(e); grid; title('Error Signal'); xlabel('number of Iterations');ylabel('the error e(n)');

subplot(3,1,2), plot(J); grid; title('J vs Iteration Steps'); xlabel('number of Iterations');ylabel('J(n)');

subplot(3,1,3), plot(Jlog); title('J Log vs Iteration Steps'); grid; xlabel('number of Iterations');ylabel('10\*log(J(i))');

% E) Frequency Response of the obtained Filter

figure(19);

freqz(w,1);

title('Frequency Response of LMS Filter');

figure(20);

freqz(filter1,1); title('Frequency Response of FIR Filter');

% They have the same phase respone.

% F) LMS Algorithm Decreasing mu value

w1 = zeros(1,M); % the updated filter coefficients

wi1 = zeros(1,M); % Intial Weights with zeros

e1 = []; %Error

mu1 = 0.0001;

% LMS Algorithm

for i = M:N

e1(i) = d(i) - wi1\* x(i:-1:i-M+1)';

w1 = wi1 + 2\*mu1\*e1(i)\*x(i:-1:i-M+1) / (epsilon + norm(x(i:-1:i-M+1))^2);

wi1 = w1;

J1(i) = (e1(i))^2;

Jlog1(i) = 10\*log(J1(i));

end

% Plot the requested Signals (e, J);

figure(21);

subplot(3, 1, 1), plot(e1); grid; title('Error Signal mu = 0.0001'); xlabel('number of Iterations');ylabel('the error e(n)');

subplot(3,1,2), plot(J1); grid; title('J vs Iteration Step mu = 0.0001'); xlabel('number of Iterations');ylabel('J(n)');

subplot(3,1,3), plot(Jlog1); title('J Log vs Iteration Step mu = 0.0001'); grid; xlabel('number of Iterations');ylabel('10\*log(J(i))');

figure(22);

freqz(w1,1);

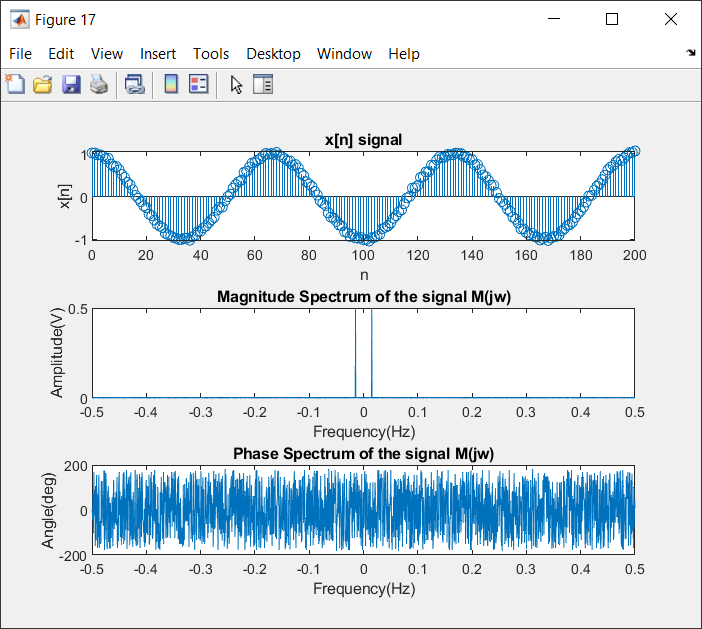
title('Frequency Response of the obtained LMS Filter mu = 0.0001');

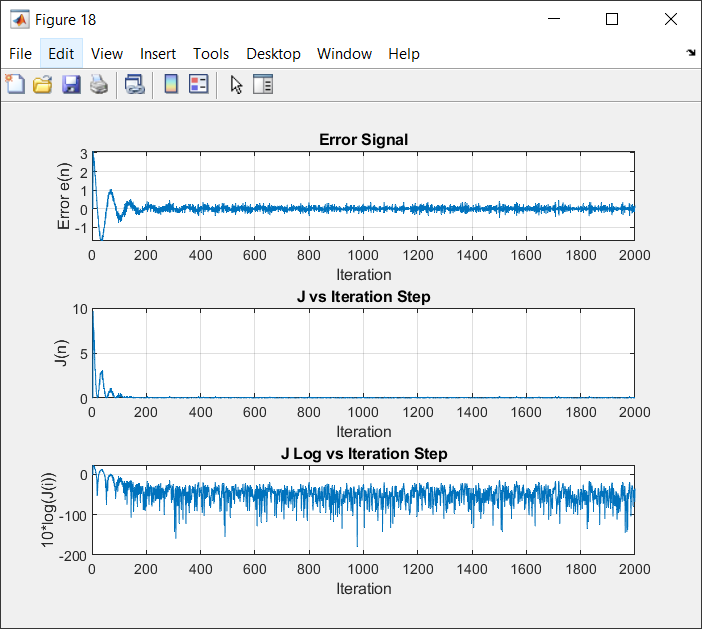
figure(23);

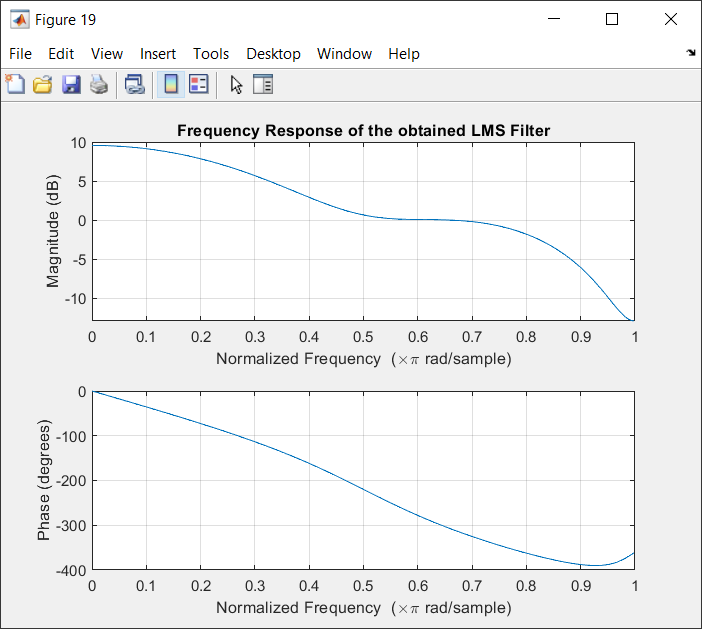
freqz(w,1); title('Frequency Response of the obtained LMS Filter mu = 0.01');

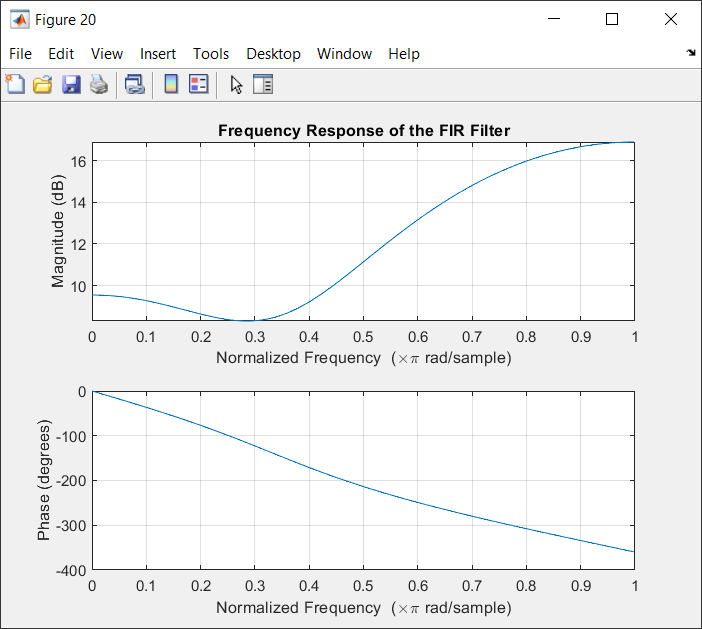
% They have the similar Frequency respone.

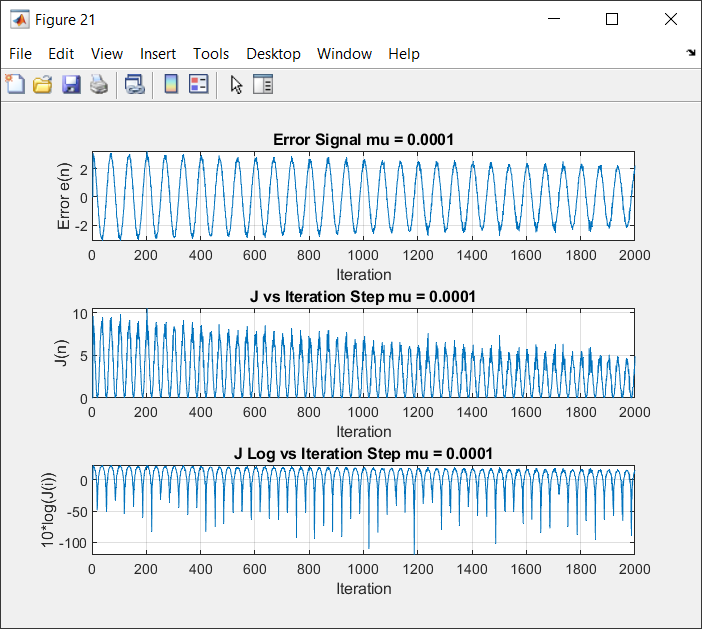
% Actually, adding awgn leads to an increase in the error and the steady state error as well as shown in the error signal figure.

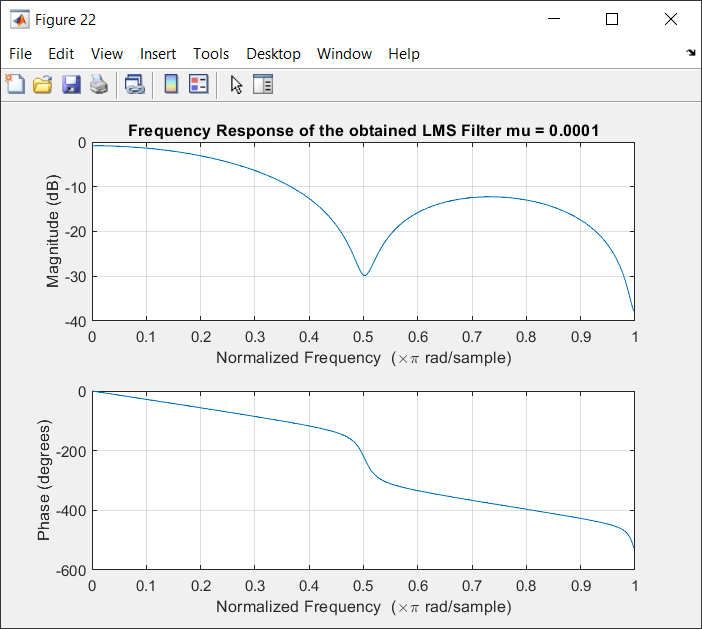


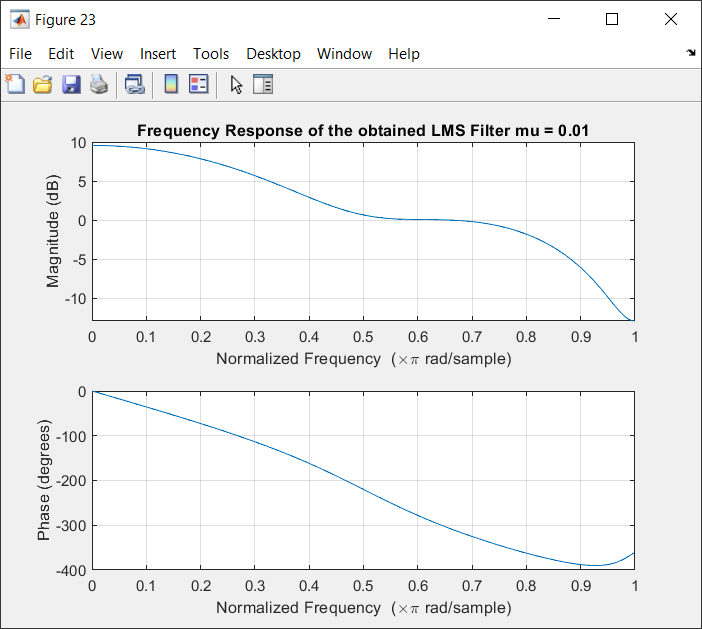












%% Ensemble Averaging

x = awgn(xn, 30);

% D) LMS Algorithm

d = filter(filter1, 1, x); % Applying the FIR Filter to obtain d[n]

M = 4;

w = zeros(1000,M); % the updated filter coefficients

wi = zeros(1000,M); % Intial Weights with zeros

e = []; %Error

mu = 0.01;

% LMS Algorithm

for k = 1:1:1000

for i = M:N

e(k,i) = d(i) - wi(k,:)\* x(i:-1:i-M+1)';

w(k,:) = wi(k,:) + 2\*mu\*e(k,i)\*x(i:-1:i-M+1) / (epsilon + norm(x(i:-1:i-M+1))^2);

wi = w;

end

end

% Averaging

e = mean(e);

w = mean(w);

J = e.^2;

Jlog = 10\*log(J);

% Plot the requested Signals (e, J);

figure(24);

subplot(3, 1, 1), plot(e); grid; title('Error Signal'); xlabel('number of Iterations');ylabel('the error e(n)');

subplot(3,1,2), plot(J); grid; title('J vs Iteration Steps'); xlabel('number of Iterations');ylabel('J(n)');

subplot(3,1,3), plot(Jlog); title('J Log vs Iteration Steps'); grid; xlabel('number of Iterations');ylabel('10\*log(J(i))');

% E) Frequency Response of the obtained Filter

figure(25);

freqz(w,1);

title('Frequency Response of LMS Filter');

figure(26);

freqz(filter1,1); title('Frequency Response of FIR Filter');

% The steady state error is smaller

% F) LMS Algorithm Decreasing mu value

w1 = zeros(1000,M); % the updated filter coefficients

wi1 = zeros(1000,M); % Intial Weights with zeros

e1 = []; %Error

mu1 = 0.0001;

% LMS Algorithm

for k = 1:1:1000

for i = M:N

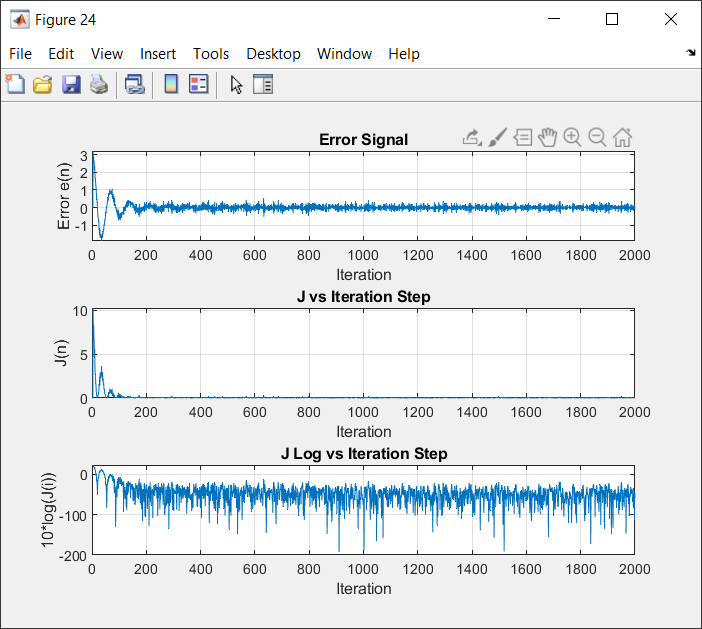
e1(k,i) = d(i) - wi1(k,:)\* x(i:-1:i-M+1)';

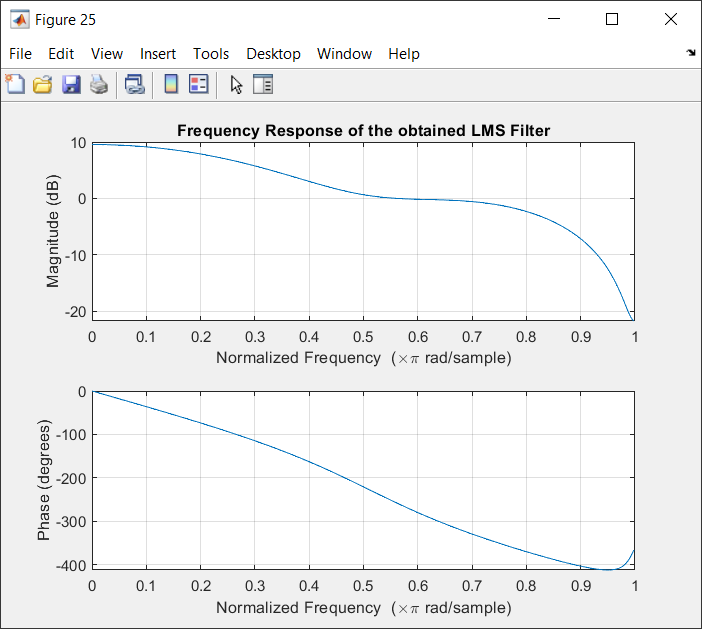
w1(k,:) = wi1(k,:) + 2\*mu1\*e1(k,i)\*x(i:-1:i-M+1) / (epsilon + norm(x(i:-1:i-M+1))^2);

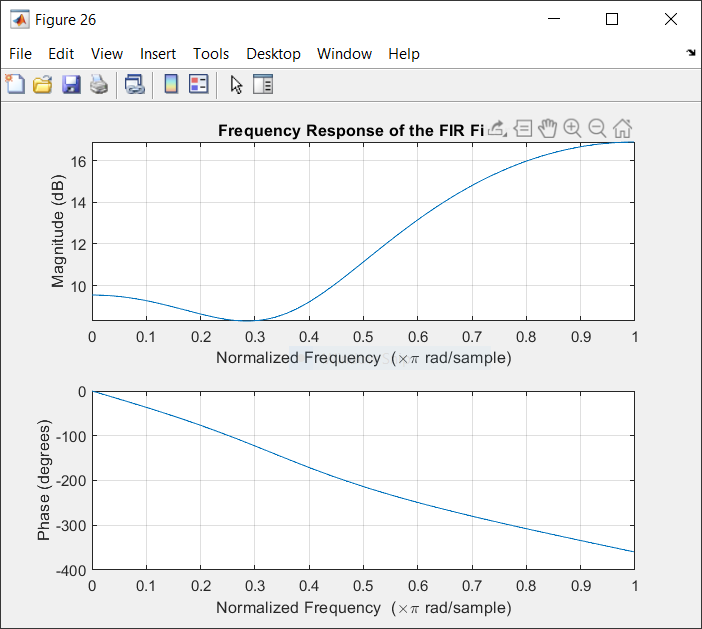
wi1 = w1;

end

end







% Averaging

e1 = mean(e1);

w1 = mean(w1);

J1 = e1.^2;

Jlog1 = 10\*log(J1);

% Plot the requested Signals (e, J);

figure(27);

subplot(3, 1, 1), plot(e1); grid; title('Error Signal mu = 0.0001'); xlabel('number of Iterations');ylabel('the error e(n)');

subplot(3,1,2), plot(J1); grid; title('J vs Iteration Step mu = 0.0001'); xlabel('number of Iterations');ylabel('J(n)');

subplot(3,1,3), plot(Jlog1); title('J Log vs Iteration Step mu = 0.0001'); grid; xlabel('number of Iterations');ylabel('10\*log(J(i))');

figure(28);

freqz(w1,1);

title('Frequency Response of the obtained LMS Filter mu = 0.0001');

figure(29);

freqz(w,1); title('Frequency Response of the obtained LMS Filter mu = 0.01');

% They have the similar Frequency respone.

% Actually, adding awgn leads to an increase in the error and the steady state error as well as shown in the error signal figure.

