**DEPARTMENT OF COMPUTER & SOFTWARE ENGINEERING**

**COLLEGE OF E&ME, NUST, RAWALPINDI**

**Subject Name**

**Digital Signal Processing**

**Lab Number**

**6**

**SUBMITTED TO:**

**LE Sundas Ashraf**

**SUBMITTED BY:**

**Student Name**

1. Wahaaj Nasir

**Reg#413238**

**DE- 44 Dept C&SE**

**Objectives:**

Processing in MATLab

**Related Topic/Chapter in theory class:**

Basics Of Digital Signal Processing

**Hardware/Software required:**

Hardware: PC

Software Tool: MATLab

**Task 1:**

**Objective: Visualize the Fourier Transform of simple signals**

**Steps:**

* 1. **Generate a simple sinusoidal signal (e.g., x(t)=sin(2πft)**
  2. **Compute its Fourier Transform using the FFT algorithm.**
  3. **Plot the magnitude spectrum and phase spectrum.**

**Solution:**

%% Task 1

T= 2;

fs = 100;

t = 0:1/fs:T;

sinosoid = sin(2\*pi\*80\*t);

X\_f = fft(sinosoid);

N = length(X\_f);

frequencies = (0:N-1) \* (fs / N);

magnitude = abs(X\_f);

phase = angle(X\_f);

subplot(2, 1, 1)

plot(frequencies, fftshift(magnitude))

ylabel("Amplitude")

xlabel("Samples")

title("Magnitude Plot")

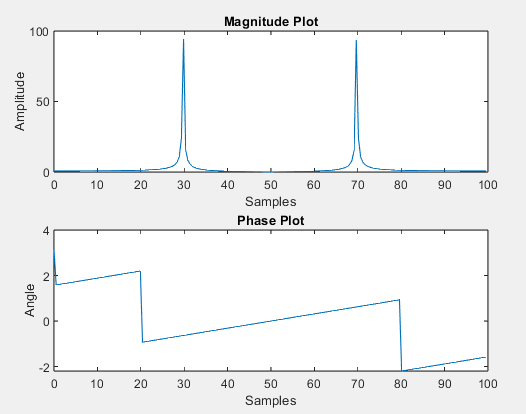
subplot(2, 1, 2)

plot(frequencies, phase)

ylabel("Angle")

xlabel("Samples")

title("Phase Plot")

****

**Task 2:**

**Objective: Analyze a composite signal using the Fourier Transform.**

**Steps:**

1. **Create a composite signal by adding multiple sinusoids of different frequencies.**
2. **Compute the FFT of the composite signal.**
3. **Identify the frequency components from the magnitude spectrum.**

**Solution**

%% Task 2

T= 2;

fs = 1000;

t = 0:1/fs:T;

f1 = 5;

f2 = 50;

f3 = 100;

sin\_f1 = sin(2\*pi\*f1\*t);

sin\_f2 = sin(2\*pi\*f2\*t);

sin\_f3 = sin(2\*pi\*f3\*t);

composite\_signal = sin\_f1 + sin\_f2 + sin\_f3;

X\_cf = fft(composite\_signal);

N = length(X\_cf);

frequencies = (0:N-1) \* (fs / N);

magnitude = abs(X\_cf);

phase = angle(X\_cf);

subplot(2, 1, 1)

plot(frequencies, fftshift(magnitude))

ylabel("Amplitude")

xlabel("Samples")

title("Magnitude Plot")

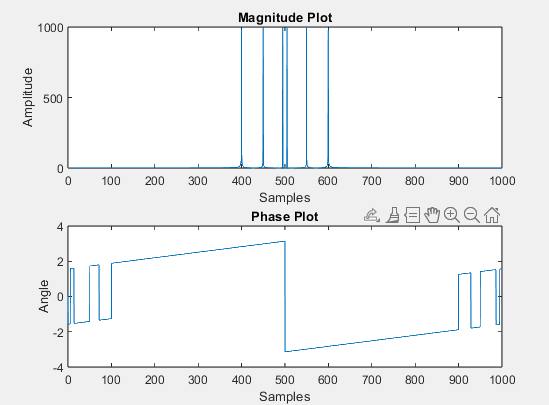
subplot(2, 1, 2)

plot(frequencies, phase)

ylabel("Angle")

xlabel("Samples")

title("Phase Plot")



**Task 3:**

**Objective: Filter a noisy signal using Lowpass band filter.**

**Steps:**

1. **Generate cosine signal in such a way that first value should be 0 with frequency 5Hz, Amplitude 5 and Fs=5000.**
2. **Compute Fourier Transform of respective signal using MATLAB command fft() or user defined function. Plot the signal in frequency domain**
3. **Add Gaussian noise in input signal using (Y =awgn(x,10,'measured')) command. Here x is input signal. Plot the resultant signal in time domain.**
4. **Pass DTFT of input signal from LTI system (Lowpass band) analyze the results.**
5. **Plot the resultant signal in time domain.**

**Solution**

%% Task 3

f = 5;

fs = 5000;

A = 5;

t = 0:1/fs:T;

cos\_sig = A \* cos((2\*pi\*f\*t)-pi/2);

X\_cos = fft(cos\_sig);

magnitude = abs(fftshift(X\_cos));

N = length(X\_cf);

noisy\_signal = awgn(cos\_sig, 10, 'measured');

order = 4;

cut\_off = 1000/(fs/2);

[b, a] = butter(order, cut\_off);

filtered = filter(b, a, noisy\_signal);

subplot(3, 1, 1)

plot(magnitude)

ylabel("Amplitude")

xlabel("Samples")

title("Fourier Magnitude Of COS")

subplot(3, 1, 2)

plot(noisy\_signal)

ylabel("Amplitude")

xlabel("Time")

title("Noisy Signal")

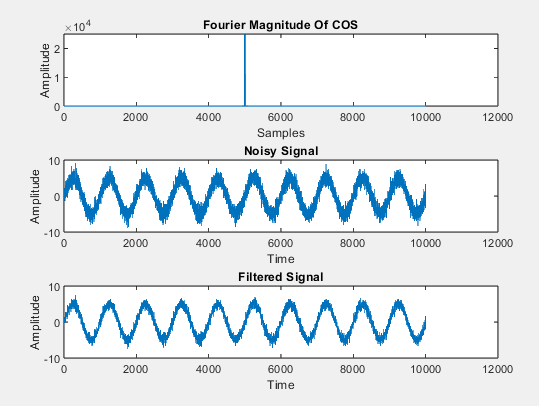
subplot(3, 1, 3)

plot(filtered)

ylabel("Amplitude")

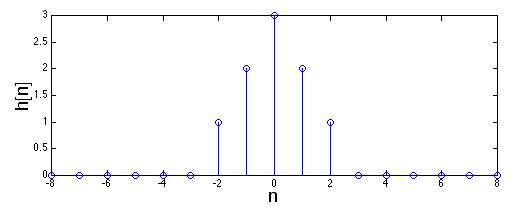
xlabel("Time")

title("Filtered Signal")



**Task 4:**

**Consider an LTI system with an even unit sample response.**

****

1. **Plot the Frequency response of this filter**
2. **Plot the phase response of this filter**
3. **Plot the magnitude response of this filter**

**Solution**

%% Task 4

n = -8:8;

h\_n = [0 0 0 0 0 0 1 2 3 2 1 0 0 0 0 0 0];

[h, w] = freqz(h\_n, 1, 1024);

subplot(2,1,1);

plot(w/pi, abs(h));

title('Magnitude Response');

xlabel('Normalized Frequency');

ylabel('|H(w)|');

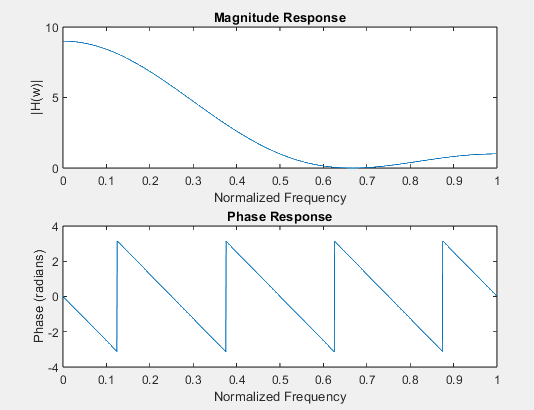
subplot(2,1,2);

plot(w/pi, angle(h));

title('Phase Response');

xlabel('Normalized Frequency');

ylabel('Phase (radians)');



**Task 5:**

**Objective: Apply the Fourier Transform to a real-world signal (e.g., audio, ECG).**

**Steps:**

* 1. **Load a real-world signal (e.g., an audio file or ECG data).**
  2. **Compute the FFT and analyze the frequency components.**
  3. **Identify dominant frequencies and their significance.**

**Solution**

%% Task 5

filepath = "D:/Uni/Semester 6/DSP/Lab/Lab 3/Record1.wav";

[signal, Fs] = audioread(filepath);

N = length(signal);

Y = fft(signal);

f = linspace(-Fs/2, Fs/2, N);

magnitude = abs(fftshift(Y))/N;

[~, peakIndex] = max(magnitude);

dominantFreq = f(peakIndex);

subplot(2,1,1);

t = (0:N-1)/Fs;

plot(t, signal);

title('Time-Domain Signal');

xlabel('Time (seconds)');

ylabel('Amplitude');

subplot(2,1,2);

plot(f, magnitude);

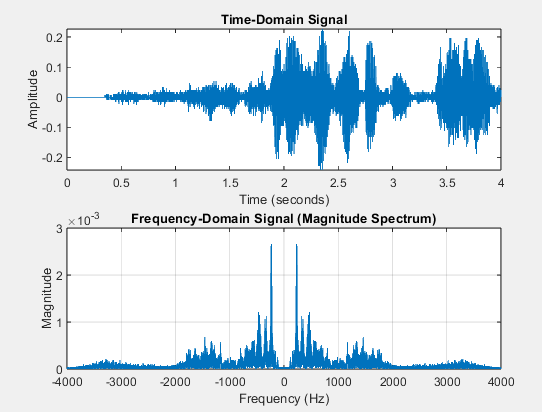
title('Frequency-Domain Signal (Magnitude Spectrum)');

xlabel('Frequency (Hz)');

ylabel('Magnitude');

grid on;

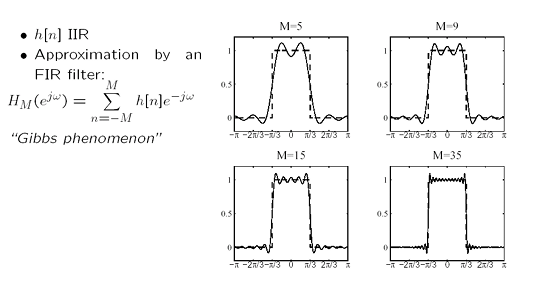
disp(['Dominant Frequency: ', num2str(dominantFreq), ' Hz']);





**Task 6:**

**You have to design the ideal low pass filter with fixed length approximation of IIR Filter, by using different values of M. Sample output is given below. Find h[n] for each, make it causal and plot the frequency, phase, and magnitude response of the system.**

****

**Solution**

%% Task 6

clc; clear; close all;

M\_values = [5, 9, 15, 35]; % Different values of M

wc = pi/4; % Cutoff frequency (adjustable)

N = 512; % FFT size for frequency response

figure;

for i = 1:length(M\_values)

M = M\_values(i);

n = -M:M; % Define the time index range

h = zeros(size(n)); % Initialize impulse response

% Compute ideal low-pass impulse response (sinc function)

h(n~=0) = sin(wc \* n(n~=0)) ./ (pi \* n(n~=0));

h(n==0) = wc / pi;

% Apply Hamming window to reduce Gibbs phenomenon

w = 0.54 - 0.46 \* cos(2 \* pi \* (n + M) / (2 \* M));

h\_w = h .\* w;

% Make the filter causal by shifting

h\_causal = [h\_w, zeros(1, M)];

% Compute Frequency Response

[H, w\_axis] = freqz(h\_causal, 1, N, 'whole');

% Plot Frequency Response

subplot(2,2,i);

plot(w\_axis - pi, abs(fftshift(H)), 'LineWidth', 1.5);

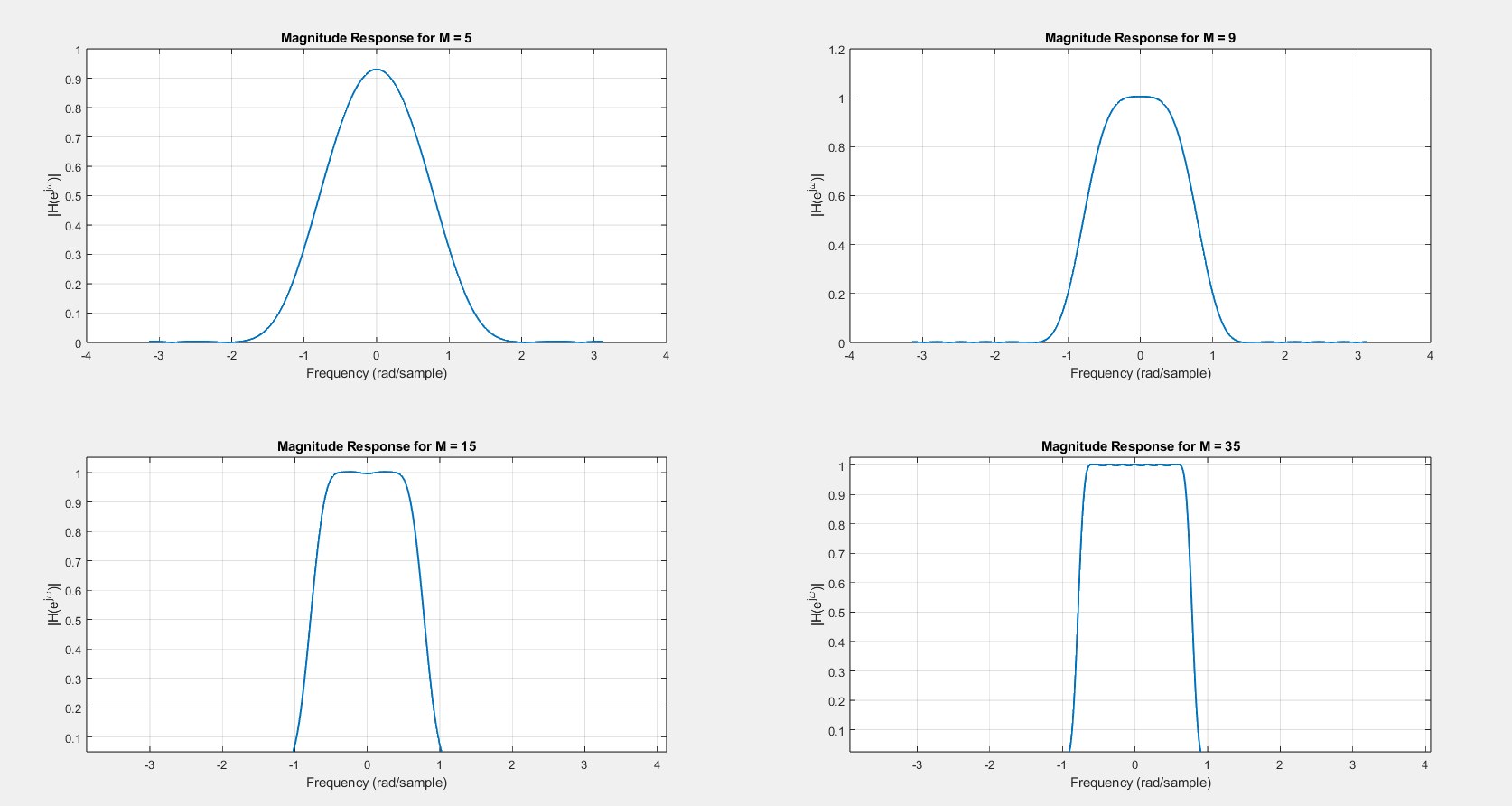
title(['Magnitude Response for M = ', num2str(M)]);

xlabel('Frequency (rad/sample)');

ylabel('|H(e^{j\omega})|');

grid on;

end

****