c) Implement this filter using Matlab or python. Show the frequency response of the filter.

import functools

from scipy.signal import butter, freqz

import matplotlib.pyplot as plt

from math import pi

import numpy as np

plt.close('all')

f\_s = 22000    # Sample frequency in Hz

f\_c = 5000     # Cut-off frequency in Hz

order = 99   # Order of the butterworth filter

omega\_c = 2 \* pi \* f\_c       # Cut-off angular frequency

omega\_c\_d = omega\_c / f\_s    # Normalized cut-off frequency (digital)

# Design the digital fir Butterworth filter

b, a = butter(order, omega\_c\_d / pi)

print('Coefficients')

print("b =", b)                           # Print the coefficients

print("a =", a)

w, H = freqz(b, a, 4096)                  # Calculate the frequency response

w \*= f\_s / (2 \* pi)                       # Convert from rad/sample to Hz

# Plot the amplitude response

plt.subplot(2, 1, 1)

plt.suptitle('Bode Plot')

H\_dB = 20 \* np.log10(abs(H))              # Convert modulus of H to dB

plt.plot(w, H\_dB)

plt.ylabel('Magnitude [dB]')

plt.xlim(0, f\_s / 2)

plt.ylim(-80, 6)

plt.axvline(f\_c, color='red')

plt.axhline(-3, linewidth=0.8, color='black', linestyle=':')

# Plot the phase response

plt.subplot(2, 1, 2)

phi = np.angle(H)                         # Argument of H

phi = np.unwrap(phi)                      # Remove discontinuities

phi \*= 180 / pi                           # and convert to degrees

plt.plot(w, phi)

plt.xlabel('Frequency [Hz]')

plt.ylabel('Phase [°]')

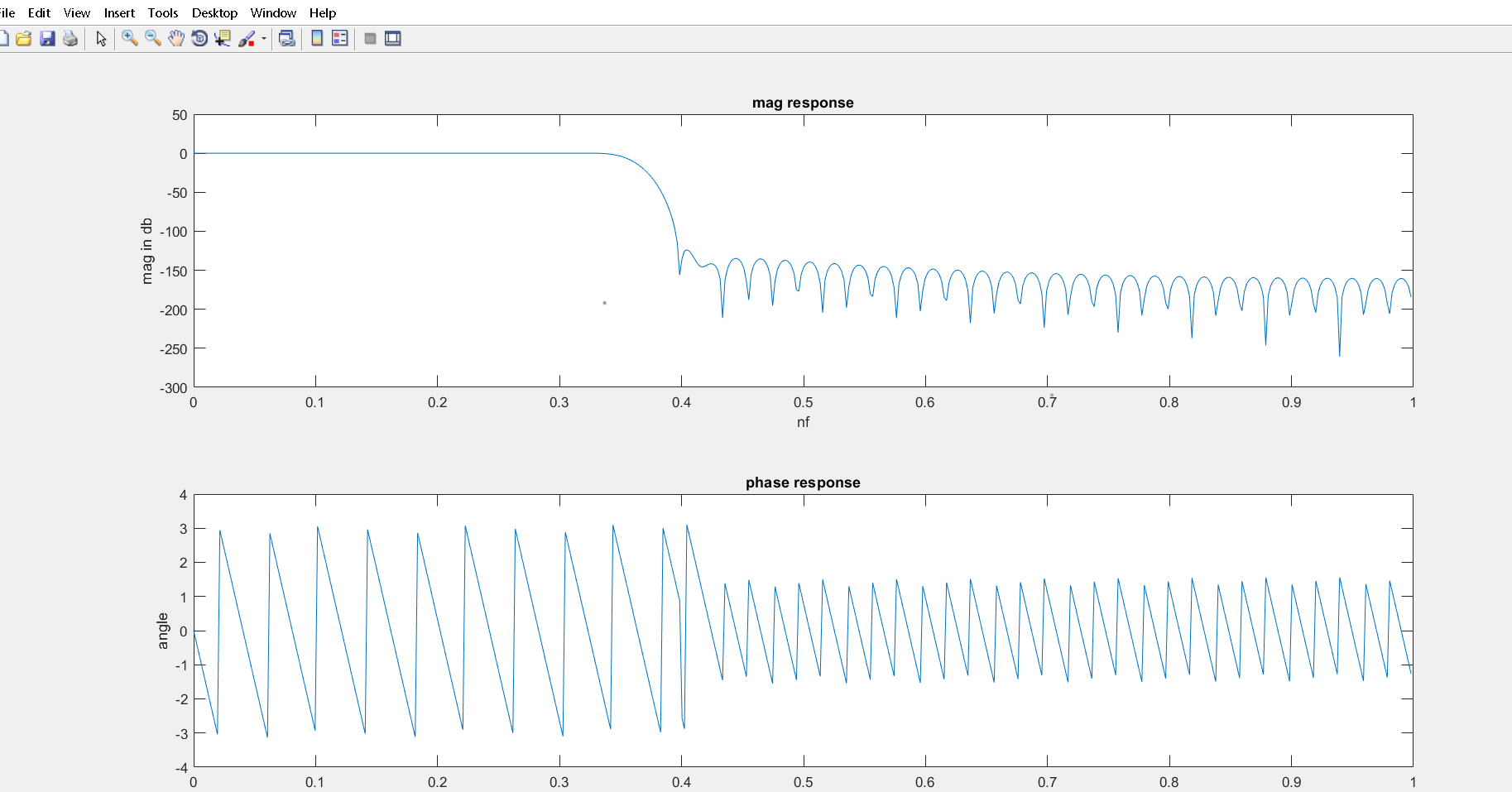
plt.xlim(0, f\_s / 2)

plt.ylim(-360, 0)

plt.yticks([-360, -270, -180, -90, 0])

plt.axvline(f\_c, color='red')

plt.show()



d) Test the filter using the following signal

clc; %clear the command window

close all; %clear all windows except those of imtods

clear ;

workspace; % Make sure the workspace panel is showing.

format long g;

format compact;

fontSize = 20;

% Let's print out the periods so we know what to expect.

% The period is just 1 over the frequency.

%We define thr specifications of the filter

fs=22e3;

%normalized frequencies in rad

Wp=(2\*4e3/fs);

Ws=2\*(4.5e3/fs);

Ap=0.8;

As=50;

%to design the fir low pass filter

%normalized cuff off frequency in rad

Wn=2\*(4e3/fs);

order=((As\*22e3)/(22\*0.5e3))-1;

%to design

h=fir1(order,Wn,'low');

% Let's print out the periods so we know what to expect.

% The period is just 1 over the frequency.

periods = 1 ./ [.08, .1, .22, .4];

% Let's have the max value of x be large enough to contain 4 cycles of the lowest frequency.

xMax = 4\*periods(1);

% Let's have, say, 500 samples over that range [0, xMax].

x = linspace(0, xMax, 500);

% Now we have x, so compute y using the formula.

y = sin(2\*pi\*0.08\*x) + sin(2\*pi\*0.1\*x) + sin(2\*pi\*0.22\*x) + sin(2\*pi\*0.4\*x);

% Now we're done and we can plot y.

plot(x, y, 'b-', 'LineWidth', 3);

% Fancy up the plot.

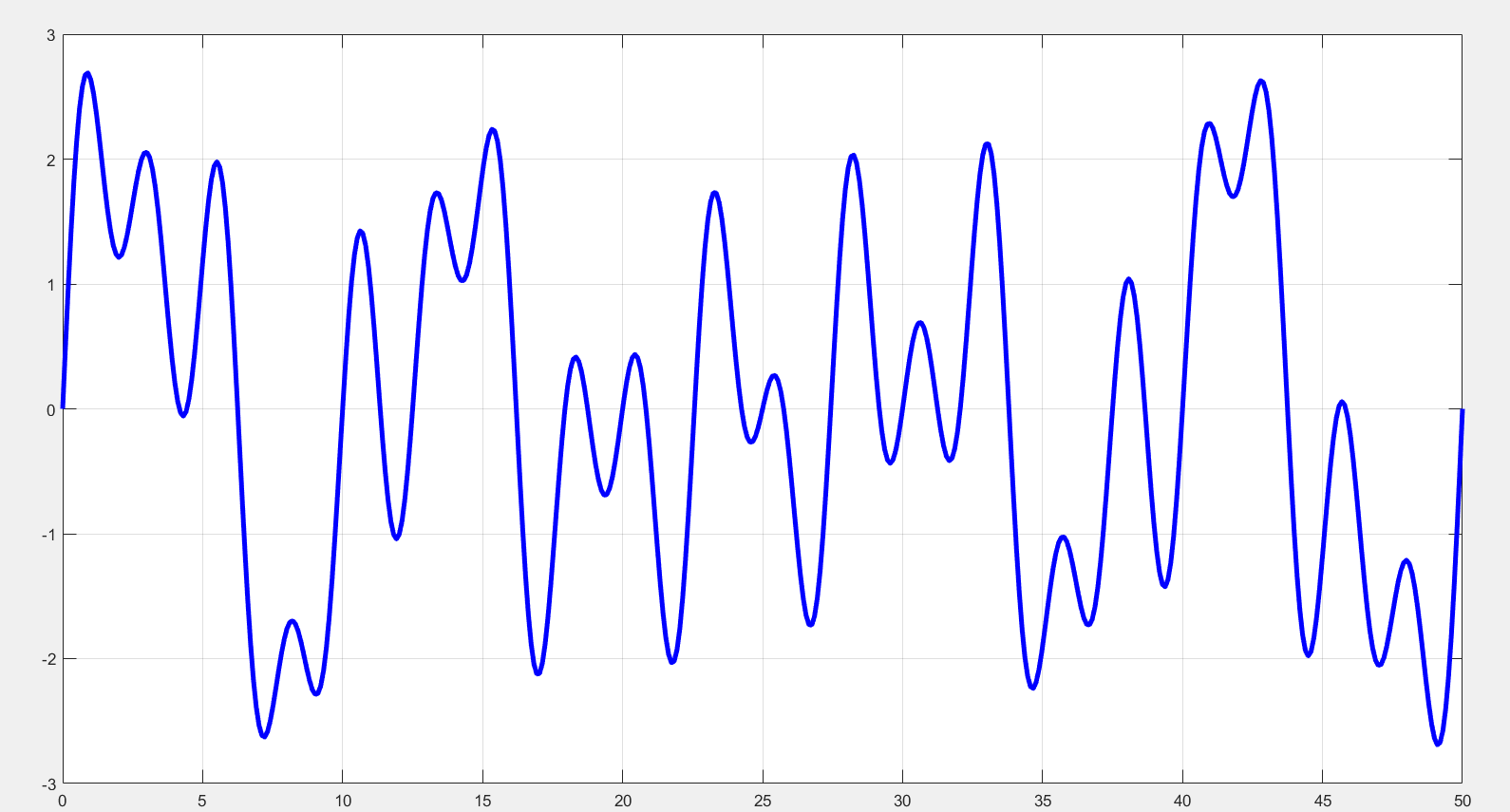
grid on;

title('Four Frequencies', 'FontSize', fontSize);

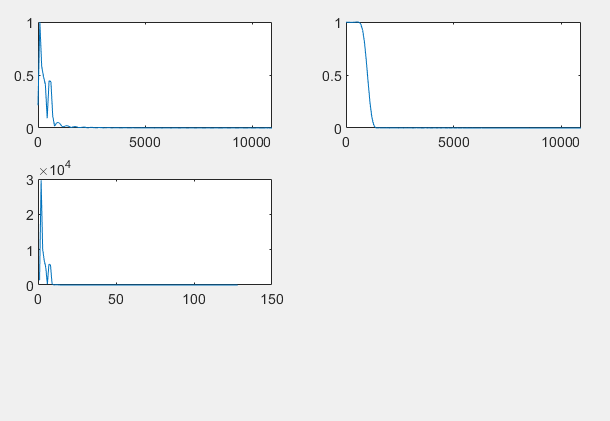
xlabel('X', 'FontSize', fontSize);

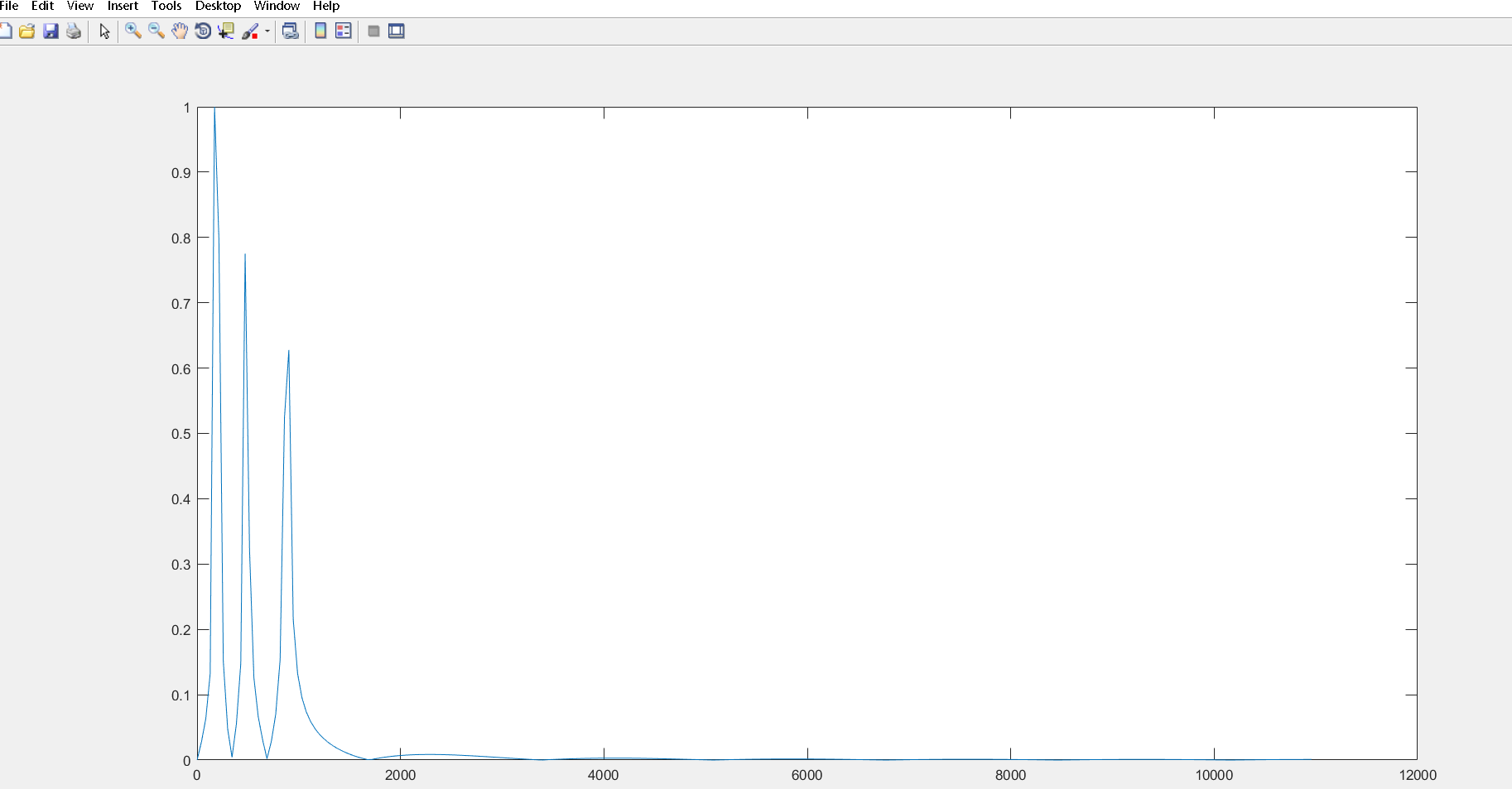
ylabel('Y', 'FontSize', fontSize);

**unfiltered signal**

****

**Filtered frequency domain**





e) Redo the filtering by changing the window type (use: Henning, Hamming). Compare the results with the above part and comment.

Hamming

clc; %clear the command window

close all; %clear all windows except those of imtods

clear ;

workspace; % Make sure the workspace panel is showing.

format long g;

format compact;

fontSize = 20;

% Let's print out the periods so we know what to expect.

% The period is just 1 over the frequency.

%We define thr specifications of the filter

fs=22e3;

%normalized frequencies in rad

Wp=(2\*4e3/fs);

Ws=(4.5e3/fs\*2);

Ap=0.8;

As=50;

%to design the fir low pass filter

%normalized cuff off frequency in rad

Wn=(4e3/(fs\*2));

order=((As\*22e3)/(22\*0.5e3))-1;

%to design

h=fir1(order,Wn,'low',hamming(order+1));

% Let's print out the periods so we know what to expect.

% The period is just 1 over the frequency.

periods = 1 ./ [.08, .1, .22, .4];

% Let's have the max value of x be large enough to contain 4 cycles of the lowest frequency.

xMax = 1\*periods(1);

% Let's have, say, 500 samples over that range [0, xMax].

x = linspace(0, xMax, 200);

% Now we have x, so compute y using the formula.

y = sin(2\*pi\*0.08\*x) + sin(2\*pi\*0.1\*x) + sin(2\*pi\*0.22\*x) + sin(2\*pi\*0.4\*x);

%change to frequency domain.

%to pass it through filter, first find signal length

nfft=length(y);

nfft2=2.^nextpow2(nfft);

%Transform impulse response of filter

fh=fft(h,nfft2);

fh=fh(1:nfft2/2); %dividing x-axis into half

%converting to frequency domain.

fy=fft(y,nfft2);

fy=fy(1:nfft2/2); %we will only use magnitude response.

%assign axis of frequency domain.

xfft=fs.\*(0:nfft2/2-1)/nfft2;

%plot(xfft,abs(fy/max(fy))); %used to plot frequency domain %we have to take the absolute function for magnitude.

mul=fy.\*fy; %we multiply in frequency domain to get convolution.

subplot(3,2,1);

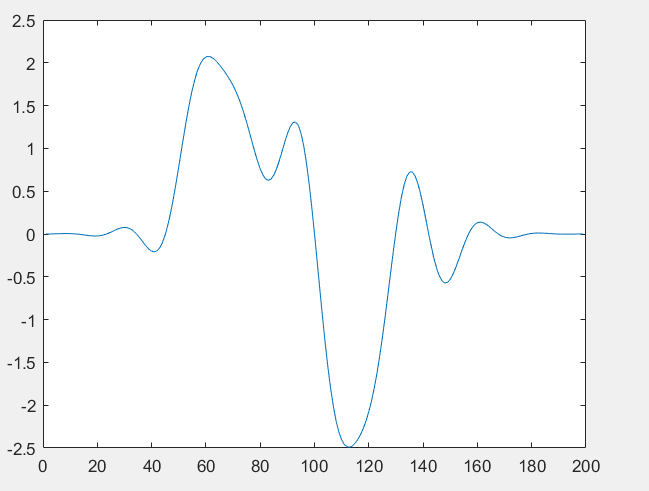
plot(xfft,abs(fy/max(fy)));

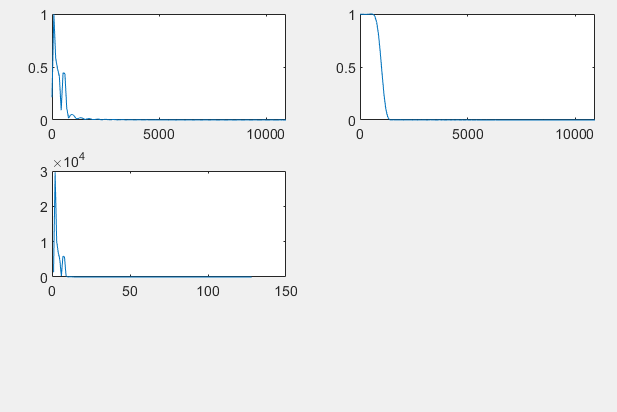
subplot(3,2,2);

plot(xfft,abs(fh/max(fh)));

subplot(3,2,3);

plot(abs(mul));





Hanning

clc; %clear the command window

close all; %clear all windows except those of imtods

clear ;

workspace; % Make sure the workspace panel is showing.

format long g;

format compact;

fontSize = 20;

% Let's print out the periods so we know what to expect.

% The period is just 1 over the frequency.

%We define thr specifications of the filter

fs=22e3;

%normalized frequencies in rad

Wp=(2\*4e3/fs);

Ws=(4.5e3/fs\*2);

Ap=0.8;

As=50;

%to design the fir low pass filter

%normalized cuff off frequency in rad

Wn=(4e3/(fs\*2));

order=((As\*22e3)/(22\*0.5e3))-1;

%to design

h=fir1(order,Wn,'low',hanning(order+1));

% Let's print out the periods so we know what to expect.

% The period is just 1 over the frequency.

periods = 1 ./ [.08, .1, .22, .4];

% Let's have the max value of x be large enough to contain 4 cycles of the lowest frequency.

xMax = 1\*periods(1);

% Let's have, say, 200 samples over that range [0, xMax].

x = linspace(0, xMax, 200);

% Now we have x, so compute y using the formula.

y = sin(2\*pi\*0.08\*x) + sin(2\*pi\*0.1\*x) + sin(2\*pi\*0.22\*x) + sin(2\*pi\*0.4\*x);

%change to frequency domain.

%to pass it through filter, first find signal length

nfft=length(y);

nfft2=2.^nextpow2(nfft);

%Transform impulse response of filter

fh=fft(h,nfft2);

fh=fh(1:nfft2/2); %dividing x-axis into half

%converting to frequency domain.

fy=fft(y,nfft2);

fy=fy(1:nfft2/2); %we will only use magnitude response.

%assign axis of frequency domain.

xfft=fs.\*(0:nfft2/2-1)/nfft2;

%plot(xfft,abs(fy/max(fy))); %used to plot frequency domain %we have to take the absolute function for magnitude.

mul=fy.\*fy; %we multiply in frequency domain to get convolution.

subplot(3,2,1);

plot(xfft,abs(fy/max(fy)));

subplot(3,2,2);

plot(xfft,abs(fh/max(fh)));

subplot(3,2,3);

plot(abs(mul));

