Communication System Lab 2

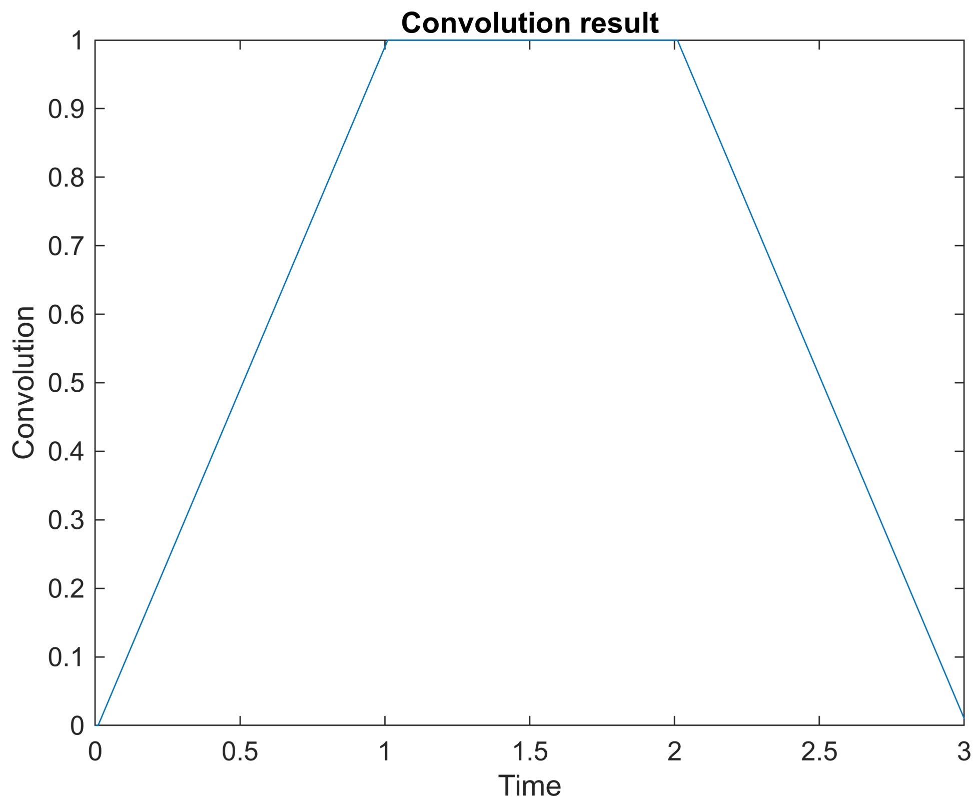
By: Saurabh Kumar

SC22B146

1. Suppose x(t) = I[0,1](t) and y(t) = I[0,2](t).

How do you obtain x(t) * y(t) using Matlab?

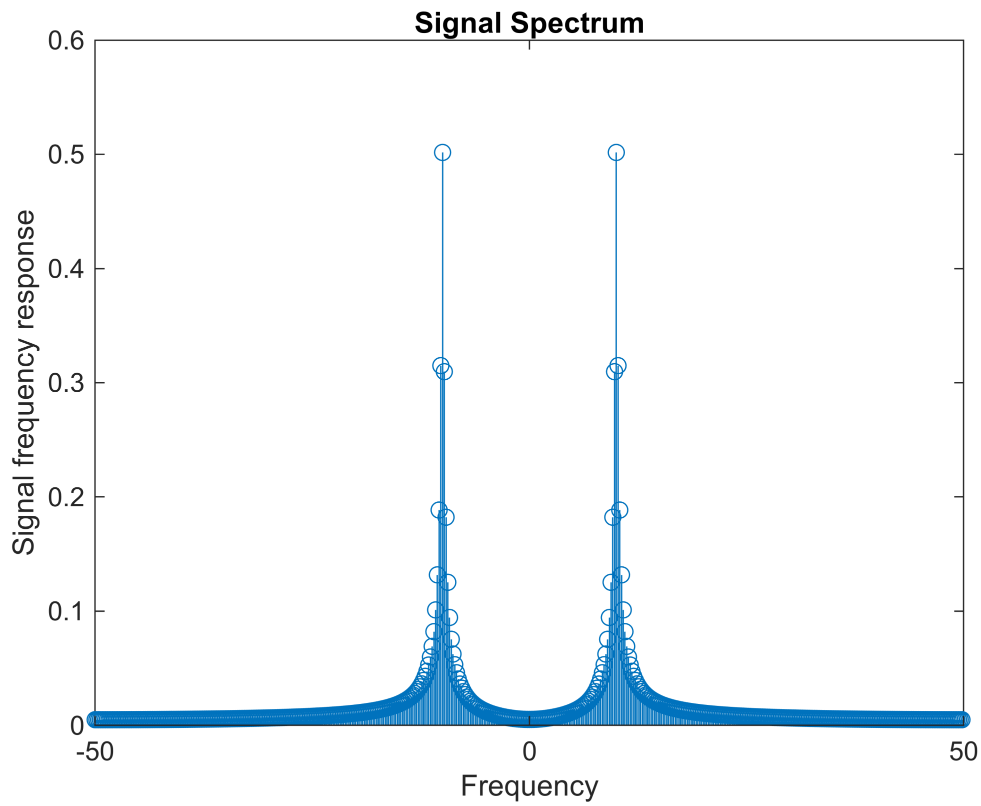
```
% time definition
Ts = 0.01;
t1 = 0:Ts:1;
t2 = 0:Ts:2;
tfinal = 0:Ts:3;

% signal definition
x = t1>0;
y = t2>0;

result = Ts * conv(x,y);

figure;
plot(tfinal,result);
title("Convolution result");
xlabel("Time");
ylabel("Convolution");
```



Inference: Convolution can be computed by in-built function 'conv' or the custom function as defined in the previous lab session. To get a scaled singal, it is multiplied by sampling time, Ts. The convolution of two square pulse shows a trapezium signal.

2. Write a function to compute and plot the spectrum for a general input signal.

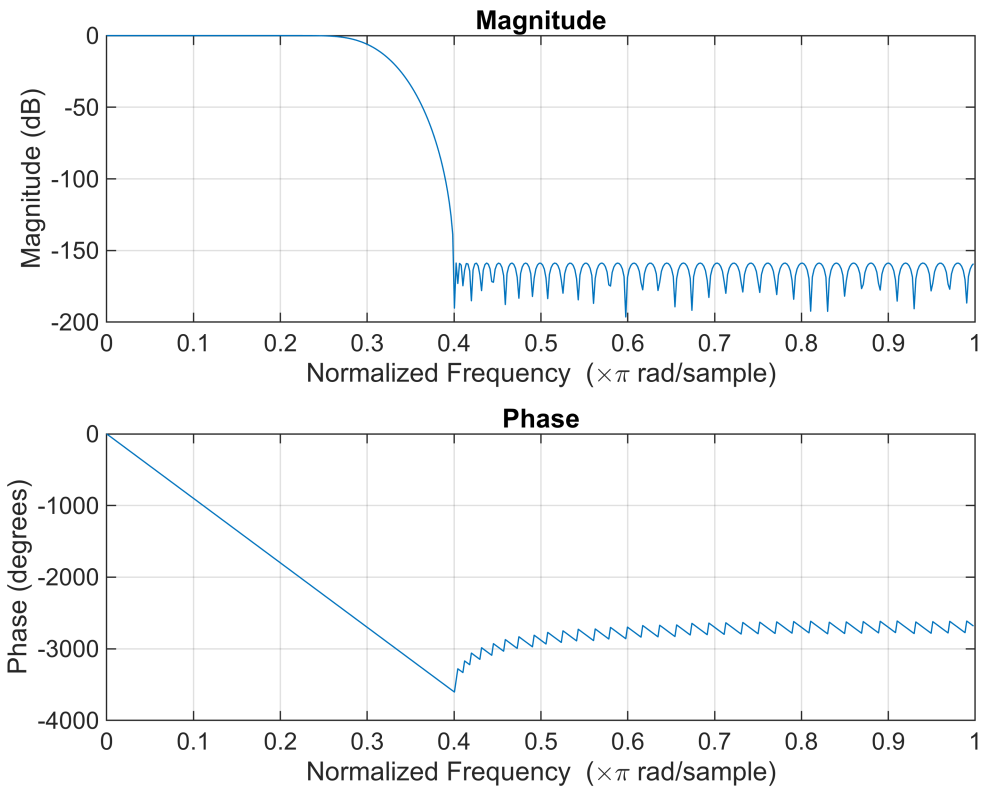
```
Ts = 0.01;
fs = 0.2;
Ttotal = 1/fs;
t = 0:Ts:(Ttotal - Ts);
% using spectrum function defined later
spectrum(cos(2 * pi * 10 * t) .* exp(-t),Ts,fs);
```



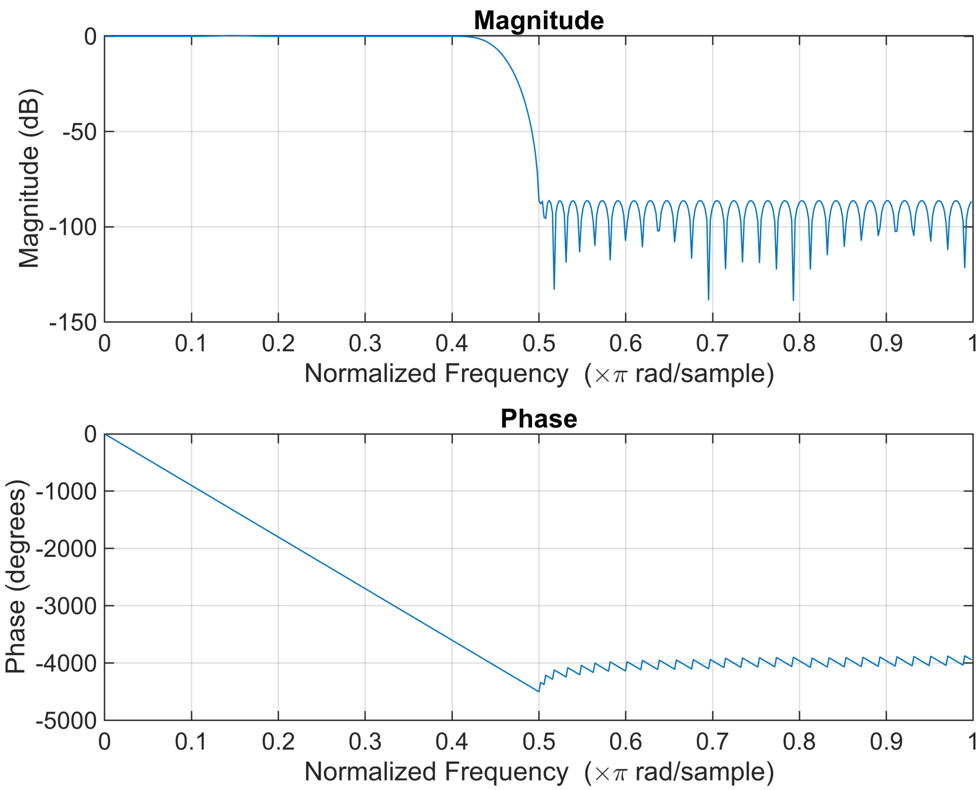
Inference: Spectrum of the input signal is plotted using the custom function 'spectrum' which computes the DFT, properly scales it and plots it. It shows the frequencies of +10 Hz and -10 Hz as the frequency response.

3. Visualize the frequency response of the low pass filters – assuming that these filters have been designed to process continuous time signals sampled at fs = 100 Hz.

```
% LP filter - 1
filter1 = firpm(100, [0, 0.2, 0.4, 1], [1, 1, 0, 0]);
figure;
freqz(filter1);
```




```
% LP filter 2
filter2 = firpm(100, [0, 0.1, 0.2, 0.4, 0.5, 1], [1, 1, 1, 1, 0, 0]);
figure;
freqz(filter2);
```



Inference: The frequency response of the low pass filters is plotted using firpm which takes input parameters that define its response. It gives the filter's time response which can be plotted with 'freqz' to get its frequency response. The first filter is having 3dB bandwidth around 0.3x100 = 30 Hz and the second has 3dB bandwidth around 0.4x100 = 40 Hz.

4. Consider a signal x(t) that consists of two cosines - one at a frequency of 10 Hz and the other at 50 Hz.

• Design a filter to filter out the 50 Hz cosine.

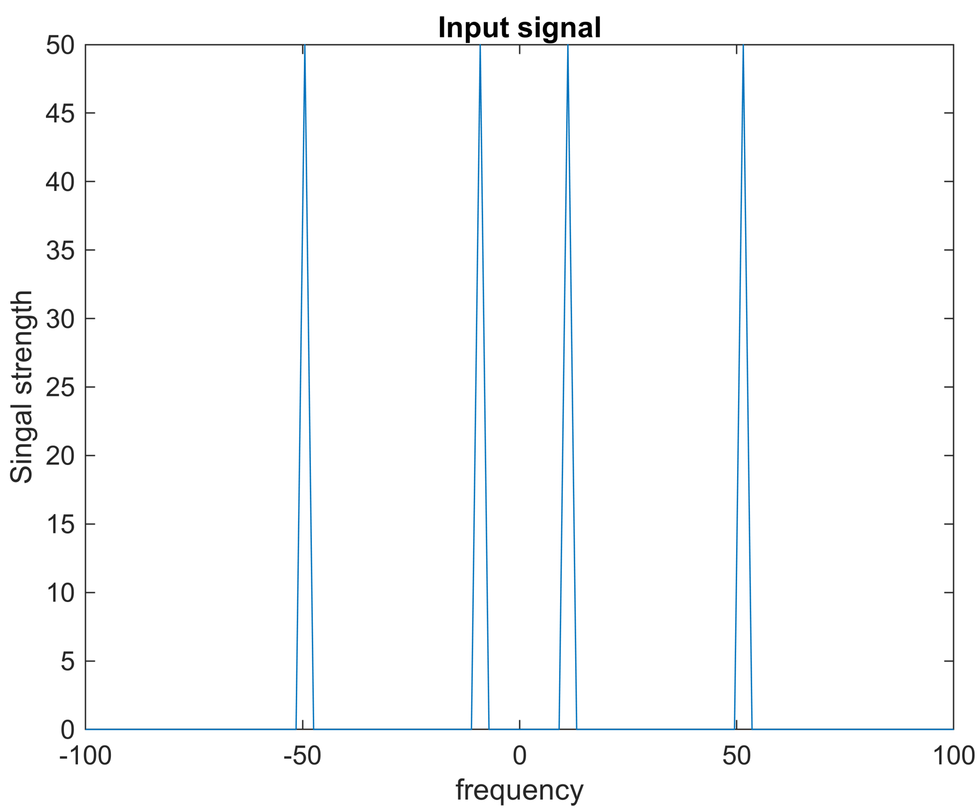
• Simulate this sytem in Matlab.

• Plot the input and output signals in time domain.

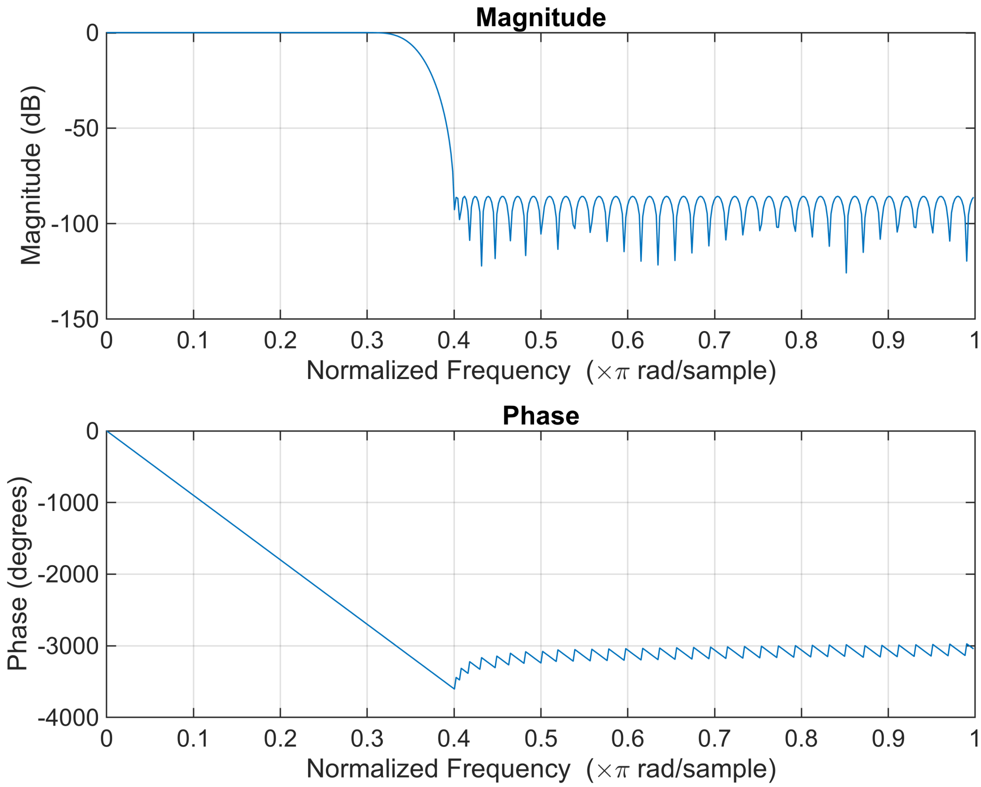
• Plot the input and output signals' spectrum and observe whether the filtering operation is successful.

```
% signal definition
fs = 200;
N = 100;
```

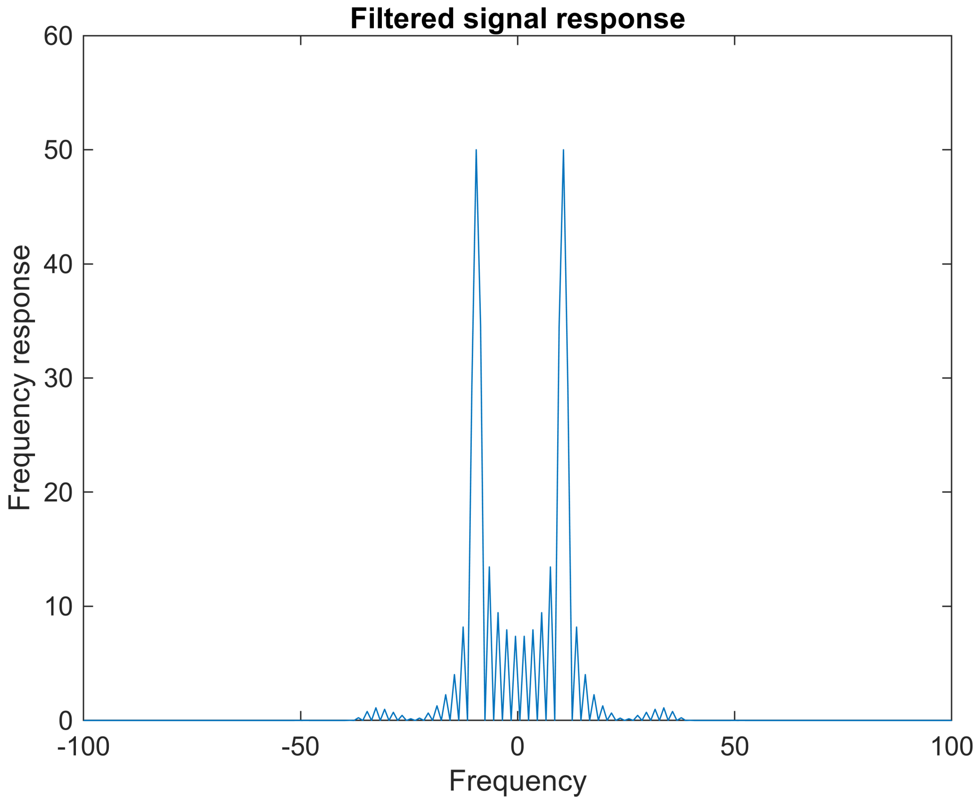
```
n = 0:N-1;
x = sin(2*pi*10*n/fs) + sin(2*pi*50*n/fs);

figure;
w = linspace(-pi,pi,N);
freq = w*fs/(2*pi);
plot(freq,abs(fftshift(fft(x))));
title("Input signal");
xlabel("frequency");
ylabel("Singal strength");
```




```
% filter definition - to filter out 50 Hz frequency
filter_50 = firpm(100, [0, 0.3, 0.4, 1], [1, 1, 0, 0]);
figure;
freqz(filter_50);
```




```
% filtering signal using filter
filtered_sig = conv(x,filter_50);
w = linspace(-pi,pi,200);
f = w*fs/(2*pi);
figure;
plot(f, abs(fftshift(fft(filtered_sig))));
title("Filtered signal response");
xlabel("Frequency");
ylabel("Frequency response");
```



Inference: The filter response is computed using 'firpm' function with properly chosen parameters to filter out 50 Hz frequency. The signal is filtered out with convolution of signal and filter, and the final filtered signal's frequency response is computed using 'fft' function. The final signal is observed to have completely filtering out 50 Hz frequency, with adding noise around 10 Hz frequency.

```
% spectrum function - for question 2
```

```
function Xk = spectrum(func,Ts,fs)
    N = length(func);
    dft_result = zeros(1,N);

    if N > length(func)
        func = [func zeros(1,N-length(func))]; % zero padding to ensure
correct size of x
    end

    for k = 0:N-1
        for n = 0:N-1
```

```
            dft_result(k+1) = dft_result(k+1) +
func(n+1)*exp(-1i*2*pi*n*k/N);
        end
    end

    Xk = Ts * dft_result;
    frequencies = (-N/2:(N/2)-1) * (fs);
    figure;
    stem(frequencies, abs(fftshift(Xk)));
    title("Signal Spectrum");
    xlabel("Frequency");
    ylabel("Signal frequency response");
end
```