Matlab function for frequency response

a. bb = [0.5, 0.5]; %-- Filter Coefficients ww = -pi:(pi/100):pi; %-- omega hat
HH = freqz(bb, 1, ww); %<--freekz.m is an alternative subplot(2,1,1);
plot(ww, abs(HH)) subplot(2,1,2); plot(ww, angle(HH))
xlabel('Normalized Radian Frequency')

b. d = designfilt('highpassfir','FilterOrder',18,'SampleRate',100, ...
'CutoffFrequency',30,'Window',{'kaiser',4});
impz(d,[],100)

$$H(e^{j\omega})=\frac{B(e^{j\omega})}{A(e^{j\omega})}=\frac{b(1)+b(2)\,e^{-j\omega}+b(3)\,e^{-j2\omega}+\cdots+b(M)\,e^{-j(M-1)\omega}}{a(1)+a(2)\,e^{-j\omega}+a(3)\,e^{-j2\omega}+\cdots+a(N)\,e^{-j(N-1)\omega}}.$$

mmaxc(centres, rev_ecdf)

mmaxc(centres, rev_ecdf)

$$a(1)y(n)=b(1)x(n)+b(2)x(n-1)+...+b(N_b)x(n-N_b+1)-a(2)y(n-1)-...-a(N_a)y(n-N_a+1)$$

is the following transfer function.

$$Y(z)=H(z^{-1})X(z)=\frac{b(1)+b(2)z^{-1}+...+b(N_b)z^{-N_b+1}}{a(1)+a(2)z^{-1}+...+a(N_a)z^{-N_a+1}}X(z)$$

Use the transfer function

$$H(z^{-1})=\frac{b(z^{-1})}{a(z^{-1})}=\frac{2+3z^{-1}}{1+0.2z^{-1}}$$

to modify the amplitude of the data in count.dat.

Load the data and assign the first column to the vector x.

load count.dat
x = count(:,1);

Create the filter coefficient vectors according to the transfer function $H(z^{-1})$.

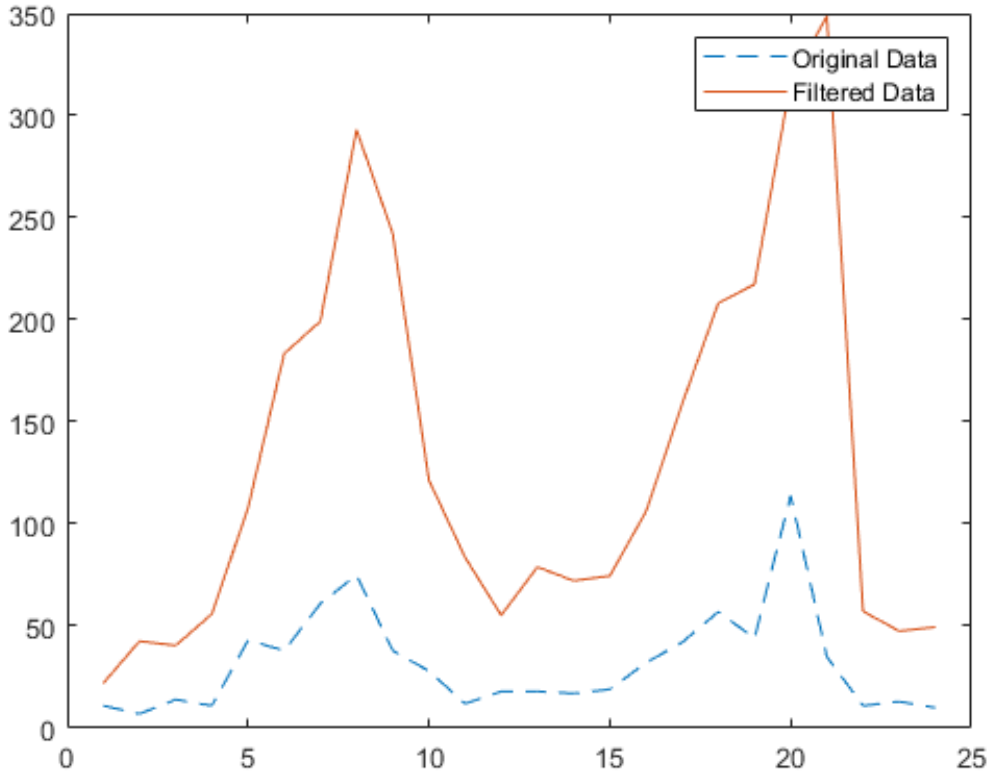
a = [1 0.2];
b = [2 3];

Compute the filtered data, and plot both the original data and the filtered data. This filter primarily modifies the amplitude of the original data.

y = filter(b,a,x);

t = 1:length(x);

```
plot(t,x,'--',t,y,'-')
legend('Original Data','Filtered Data')
```



c. y = filter(b,a,x)
   y = filter(b,a,x,zi)
   y = filter(b,a,x,zi,dim)
   [y,zf] = filter(___)

Multiband CLS Filter Design

fircls uses the same technique to design FIR filters with a specified piecewise constant magnitude response. In this case, you can specify a vector of band edges and a corresponding vector of band amplitudes. In addition, you can specify the maximum amount of ripple for each band.

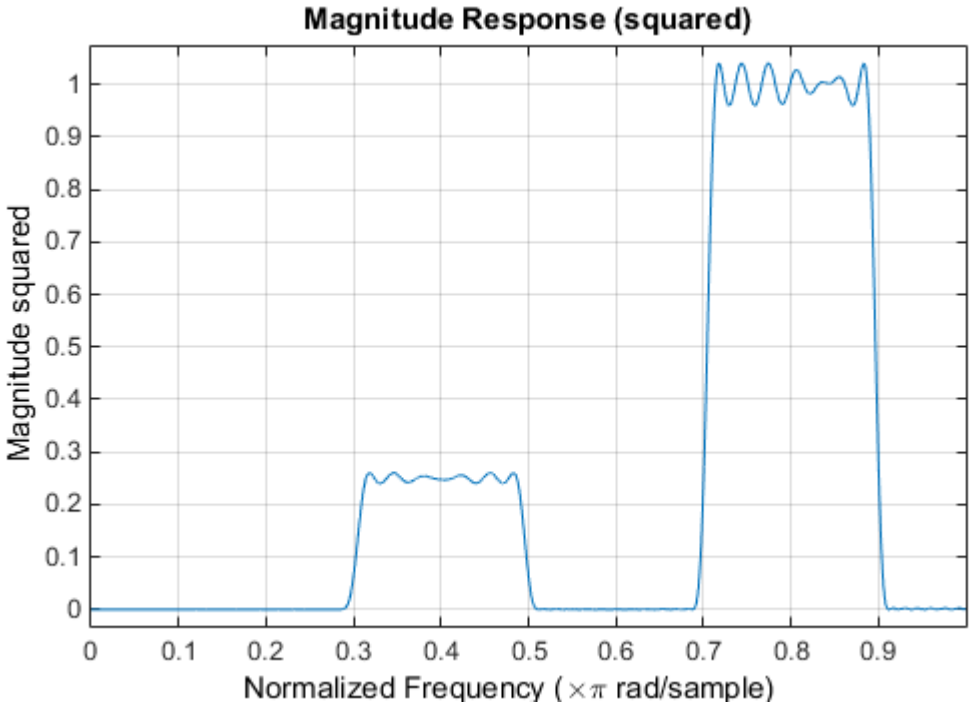
For example, assume the specifications for a filter call for:

- From 0 to 0.3 (normalized): amplitude 0, upper bound 0.005, lower bound –0.005
- From 0.3 to 0.5: amplitude 0.5, upper bound 0.51, lower bound 0.49
- From 0.5 to 0.7: amplitude 0, upper bound 0.03, lower bound –0.03
- From 0.7 to 0.9: amplitude 1, upper bound 1.02, lower bound 0.98
- From 0.9 to 1: amplitude 0, upper bound 0.05, lower bound –0.05

Design a CLS filter with impulse response order 129 that meets these specifications:

```
n = 129;
f = [0 0.3 0.5 0.7 0.9 1];
a = [0 0.5 0 1 0];
up = [0.005 0.51 0.03 1.02 0.05];
lo = [-0.005 0.49 -0.03 0.98 -0.05];
h = fircls(n,f,a,up,lo);
fvtool(h,1)
```

Note that the *y*-axis shown below is in Magnitude Squared. You can set this by right-clicking on the axis label and selecting **Magnitude Squared** from the menu.



Weighted CLS Filter Design

Weighted CLS filter design lets you design lowpass or highpass FIR filters with relative weighting of the error minimization in each band. The fircls1 function enables you to specify the passband and stopband edges for the least squares weighting function, as well as a constant k that specifies the ratio of the stopband to passband weighting.

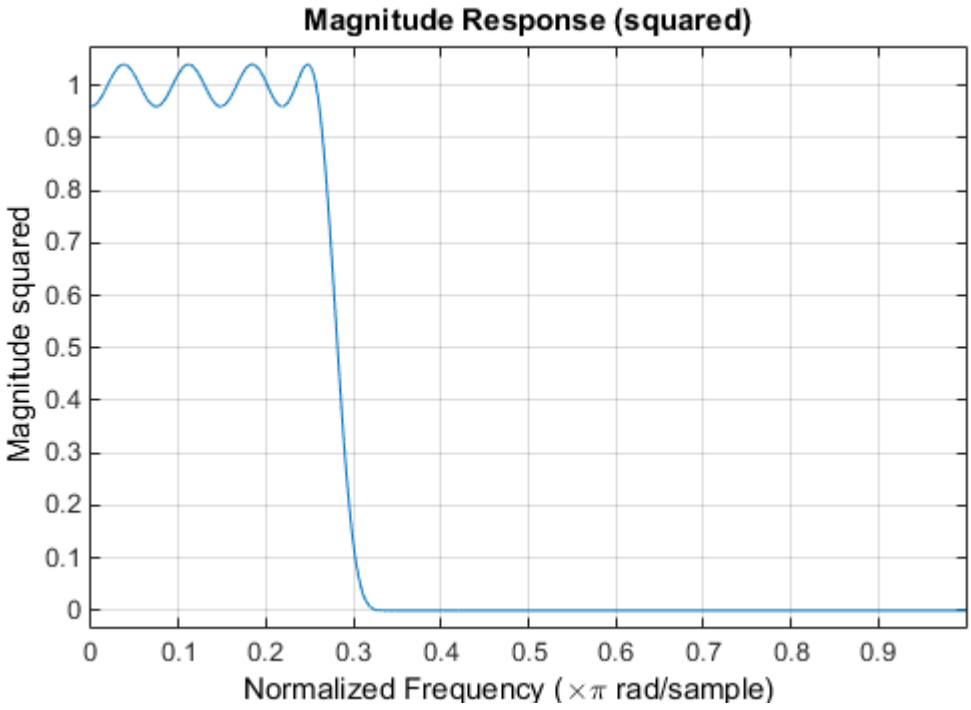
For example, consider specifications that call for an FIR filter with impulse response order of 55 and cutoff frequency of 0.3 (normalized). Also assume maximum allowable passband ripple of 0.02 and maximum allowable stopband ripple of 0.004. In addition, add weighting requirements:

- Passband edge for the weight function of 0.28 (normalized)
- Stopband edge for the weight function of 0.32
- Weight error minimization 10 times as much in the stopband as in the passband

To approach this using fircls1, type

```
n = 55;
wo = 0.3;
dp = 0.02;
ds = 0.004;
wp = 0.28;
ws = 0.32;
k = 10;
h = fircls1(n,wo,dp,ds,wp,ws,k);
fvtool(h,1)
```

Note that the *y*-axis shown below is in Magnitude Squared. You can set this by right-clicking on the axis label and selecting **Magnitude Squared** from the menu.



Arbitrary-Response Filter Design

The cfirpm filter design function provides a tool for designing FIR filters with arbitrary complex responses. It differs from the other filter design functions in how the frequency response of the filter is specified: it accepts the name of a function which returns the filter response calculated over a grid of frequencies. This capability makes cfirpm a highly versatile and powerful technique for filter design.

This design technique may be used to produce nonlinear-phase FIR filters, asymmetric frequency-response filters (with complex coefficients), or more symmetric filters with custom frequency responses.

The design algorithm optimizes the Chebyshev (or minimax) error using an extended Remez-exchange algorithm for an initial estimate. If this exchange method fails to obtain the optimal filter, the algorithm switches to an ascent-descent algorithm that takes over to finish the convergence to the optimal solution.

Multiband Filter Design

Consider a multiband filter with the following special frequency-domain characteristics.

Band	Amplitude	Optimization Weighting
[–1 –0.5]	[5 1]	1
[–0.4 +0.3]	[2 2]	10
[+0.4 +0.8]	[2 1]	5

A linear-phase multiband filter may be designed using the predefined frequency-response function multiband, as follows:

```
b = cfirpm(38, [-1 -0.5 -0.4 0.3 0.4 0.8], ...
            {'multiband', [5 1 2 2 2 1]}, [1 10 5]);
```

For the specific case of a multiband filter, we can use a shorthand filter design notation similar to the syntax for firpm:

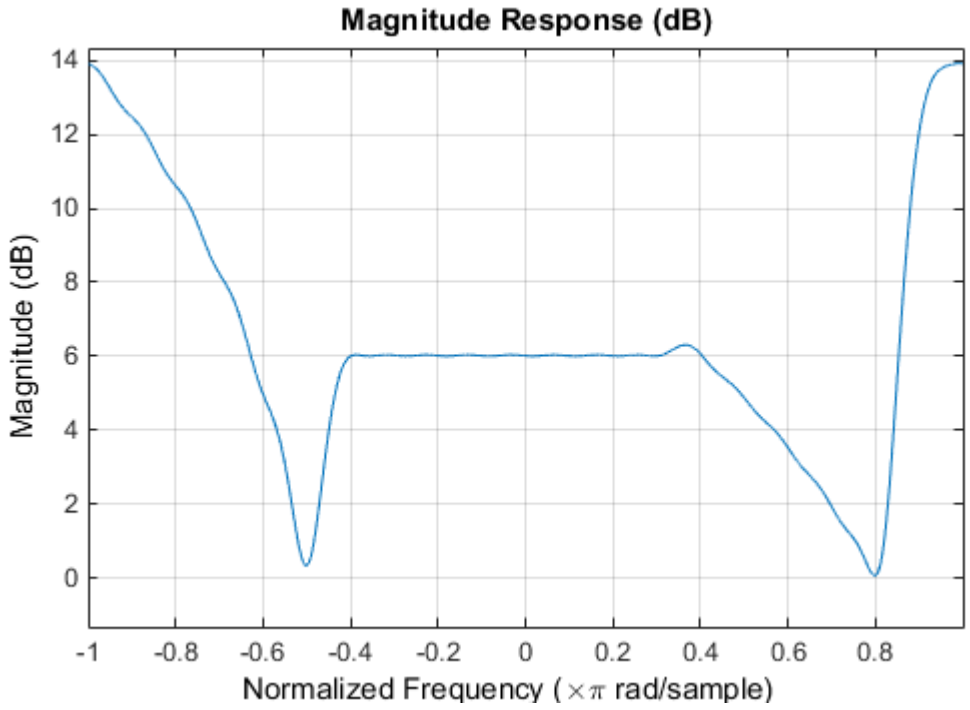
```
b = cfirpm(38,[-1 -0.5 -0.4 0.3 0.4 0.8], ...
           [5 1 2 2 2 1], [1 10 5]);
```

As with firpm, a vector of band edges is passed to cfirpm. This vector defines the frequency bands over which optimization is performed; note that there are two transition bands, from –0.5 to –0.4 and from 0.3 to 0.4.

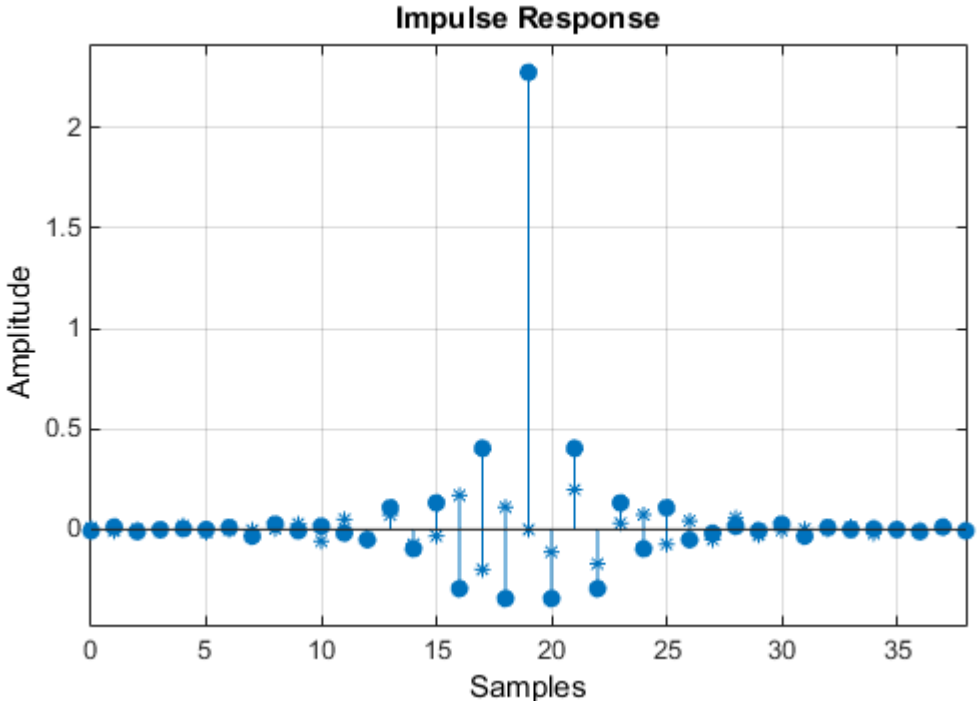
In either case, the frequency response is obtained and plotted using linear scale in FVTool:

```
fvtool(b,1)
```

Note that the range of data shown below is (-pi,pi).



The filter response for this multiband filter is complex, which is expected because of the asymmetry in the frequency domain. The impulse response, which you can select from the FVTool toolbar, is shown below.



Filter Design with Reduced Delay

Consider the design of a 62-tap lowpass filter with a half-Nyquist cutoff. If we specify a negative offset value to the lowpass filter design function, the group delay offset for the design is

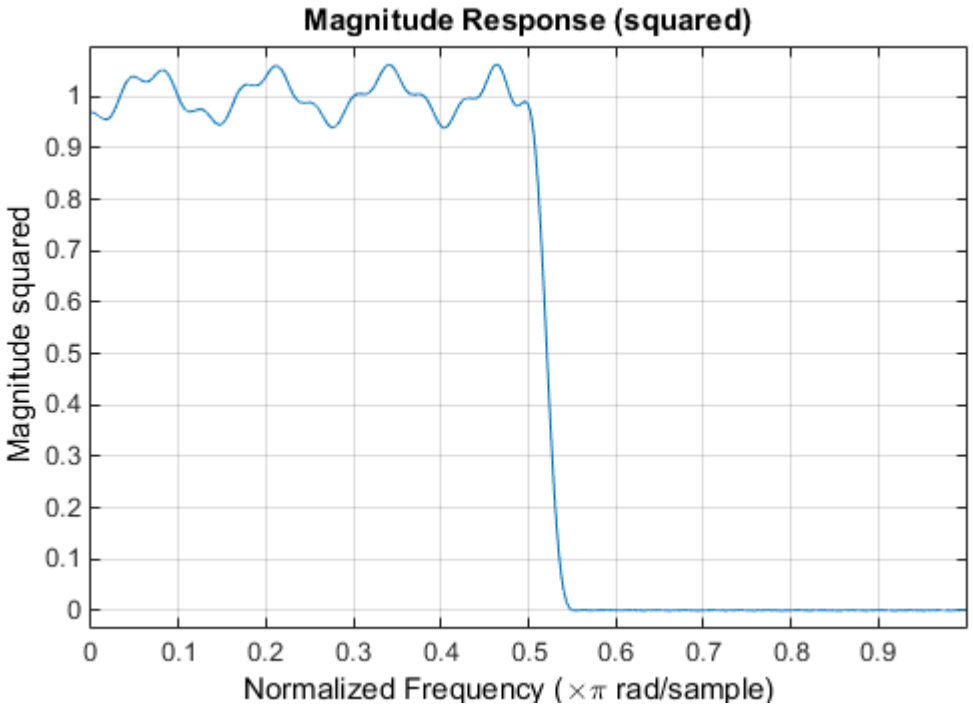
significantly less than that obtained for a standard linear-phase design. This filter design may be computed as follows:

```
b = cfirpm(61,[0 0.5 0.55 1],{'lowpass',-16});
```

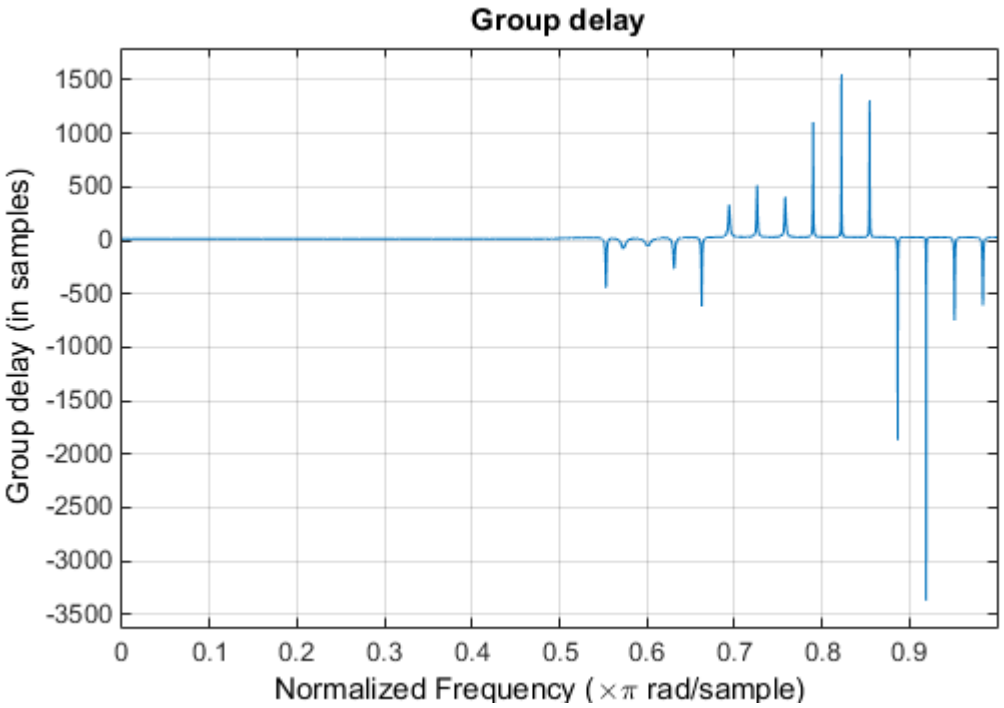
The resulting magnitude response is

```
fvtool(b,1)
```

The *y*-axis is in Magnitude Squared, which you can set by right-clicking on the axis label and selecting **Magnitude Squared** from the menu.



The group delay of the filter reveals that the offset has been reduced from N/2 to N/2-16 (i.e., from 30.5 to 14.5). Now, however, the group delay is no longer flat in the passband region. To create this plot, click the **Group Delay Response** button on the toolbar.



If we compare this nonlinear-phase filter to a linear-phase filter that has exactly 14.5 samples of group delay, the resulting filter is of order 2*14.5, or 29. Using b = cfirpm(29,[0 0.5 0.55 1],'lowpass'), the passband and stopband ripple is much greater for the order 29 filter. These comparisons can assist you in deciding which filter is more appropriate for a specific application.

%Magnitude response rectangular window defined over n = 0 to N-1

% h(n) = {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1} N = 11;

b11= ones(1, 11); a=[1];

w=-pi: pi/256: pi; r = w/pi;

%

%Normalized magnitude H(ω)/N

Hw11n= freqz(b11, a, w)/N;

subplot(2, 1, 1), plot(w, abs(Hw11n)); legend (' Length = 11'); xlabel('Frequency \omega, rad/sample'), ylabel('Magnitude of H(\omega)');grid; title ('Normalized magnitude')

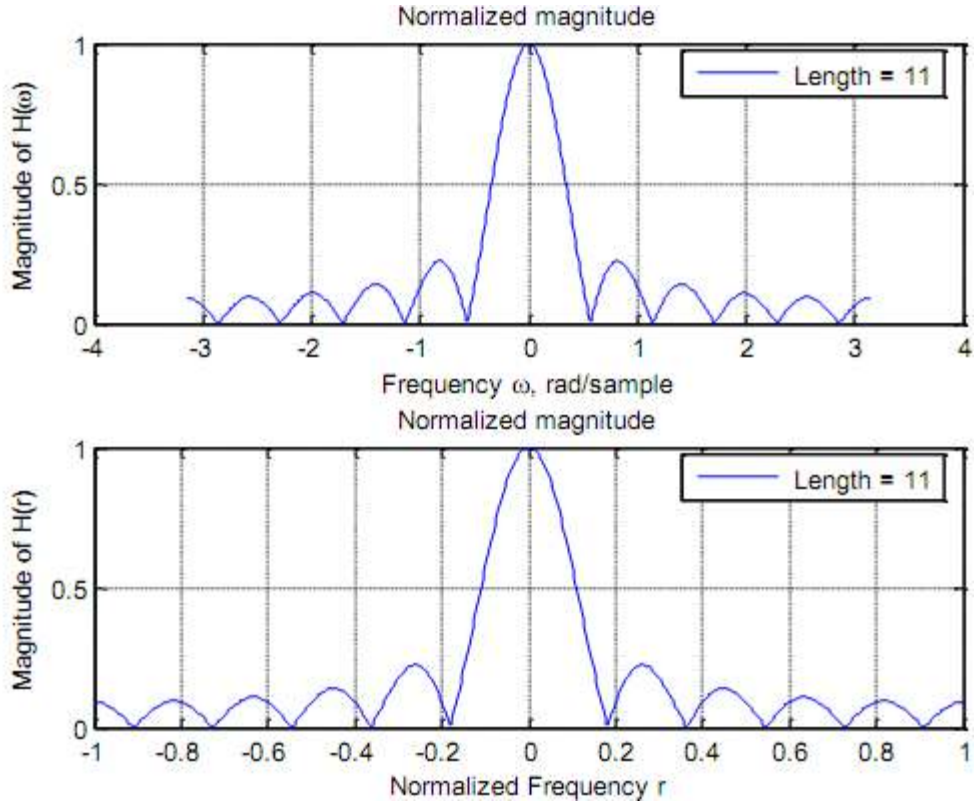
%

%Normalized magnitude and normalized frequency

Hr11n= freqz(b11, a, pi*r)/N;

subplot(2, 1, 2), plot(r, abs(Hr11n)); legend (' Length = 11'); xlabel('Normalized Frequency r'), ylabel('Magnitude of H(r)'); grid; title ('Normalized magnitude')

Note in the plot below that the height of the main lobe is 1 since it is normalized.



Email based Normalized magnitude assignment help