HW8 - filter design

Sunday, April 24, 2022 7:01 PM

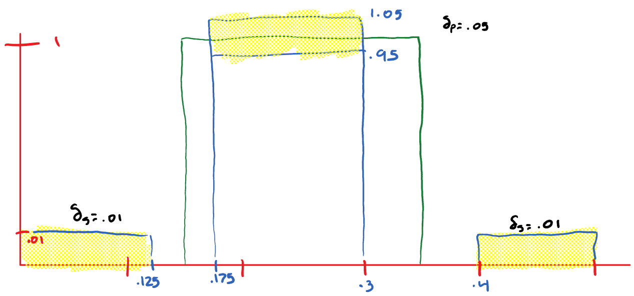
1. This problem explores the design a symmetric real-valued linear phase FIR bandpass filter that meets the following specifications.

Amplitude Spec.	Frequency Spec.
$\lvert H(f)\rvert \leq 0.01$	$0 \leq \lvert f\rvert \leq 0.125$
$0.95 \leq \lvert H(f)\rvert \leq 1.05$	$0.175 \leq \lvert f\rvert \leq 0.3$
$\lvert H(f)\rvert \leq 0.01$	$0.4 \leq \lvert f\rvert \leq 0.5$

- Design this filter using the window technique using the best window from the set {rectangular, Bartlett, Hann, Hamming, Blackman}. Indicate which window shape you used.
- Design this filter using the window technique using the Kaiser window. Indicate the Kaiser window parameters (β and M).
- Using FDATool in Matlab, design this filter using equiripple design method.

Make a table with one row for each filter you designed (window, Kaiser, equiripple). The table should have a column for the following parameters: length, order, group delay, type (I, II, III, or IV), and window parameters.

For each filter you design, turn in plots of the impulse response, magnitude response (linear and dB), and phase response. The frequency domain plots should show response over the interval $-\frac{1}{2} \leq f \leq \frac{1}{2}$. Also include a pole-zero plot.



$20 \log_{10}(.01) = -40$

$10^{\frac{-40}{20}} = .01$

$A_s = 20 \cdot \log_{10}(\delta_s) = -40 \longrightarrow$ Hann

$A_p = 20 \cdot \log_{10}\left(\frac{1+\delta_p}{1-\delta_p}\right) = .8693$

$\Delta f = \min(\text{transition band}) = .05$

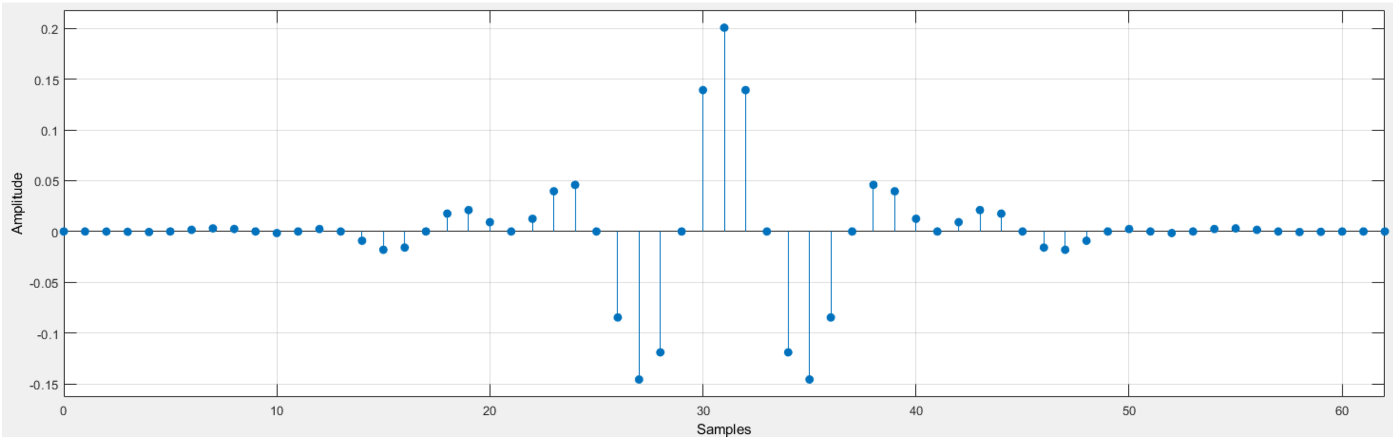
$\Delta\omega = 2\pi \cdot \Delta f = .3142$

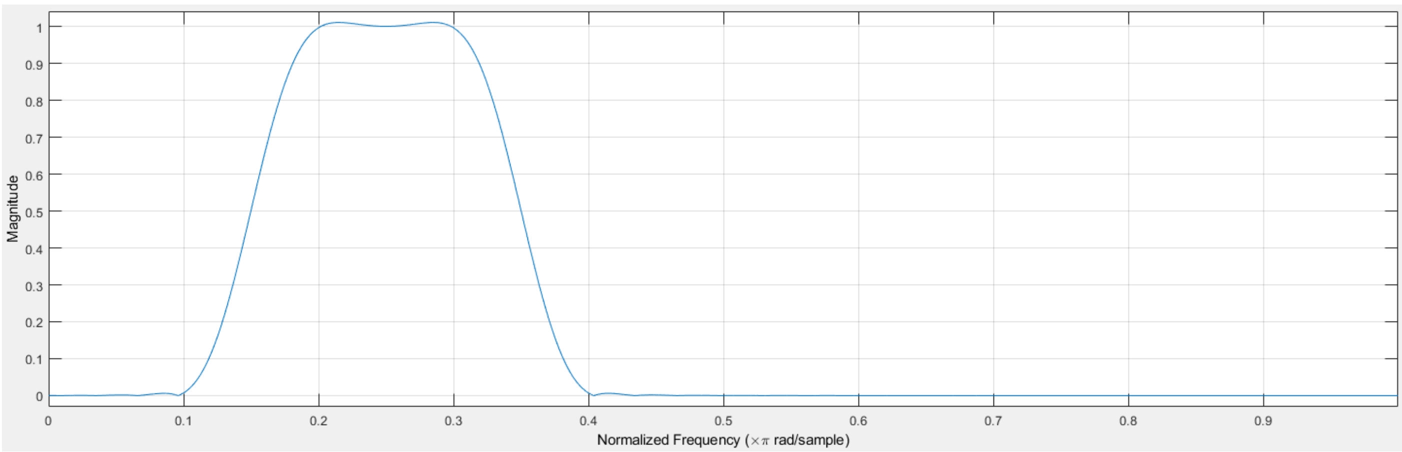
$L = \frac{K_{win}\pi}{\Delta\omega} = \frac{6.2\pi}{.3142} = 62$

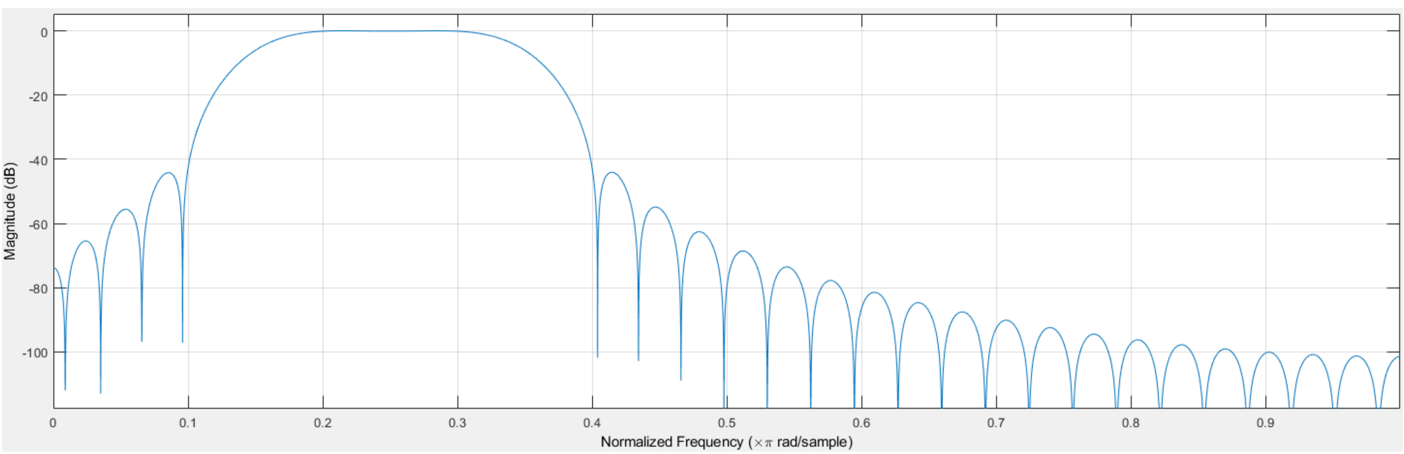
$M = L - 1 = 61$ (type II)

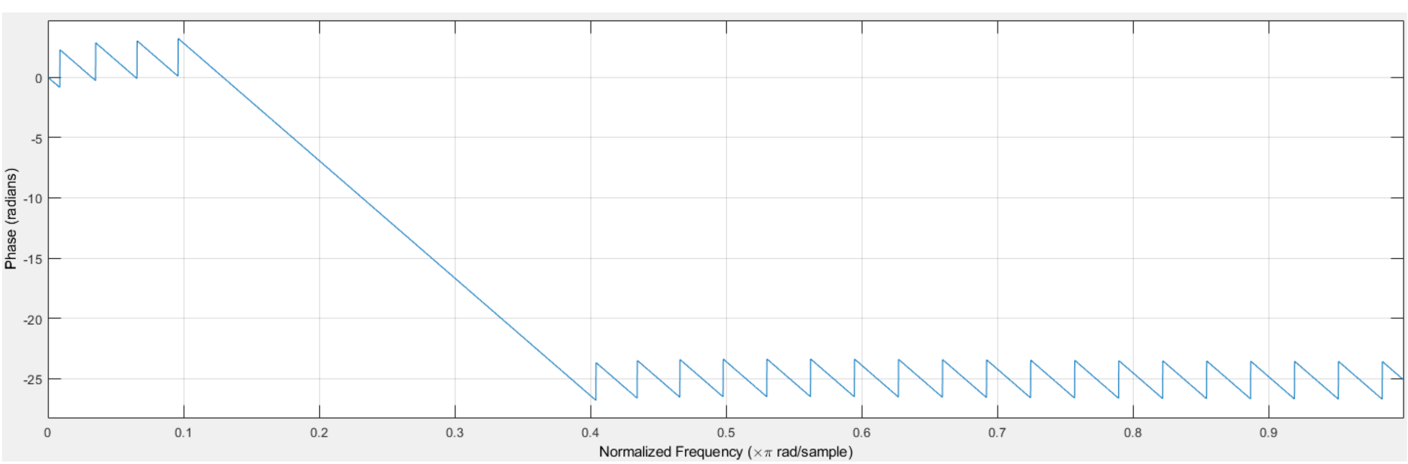
$f_{c_1} = \frac{.175 - .125}{2} = .15$

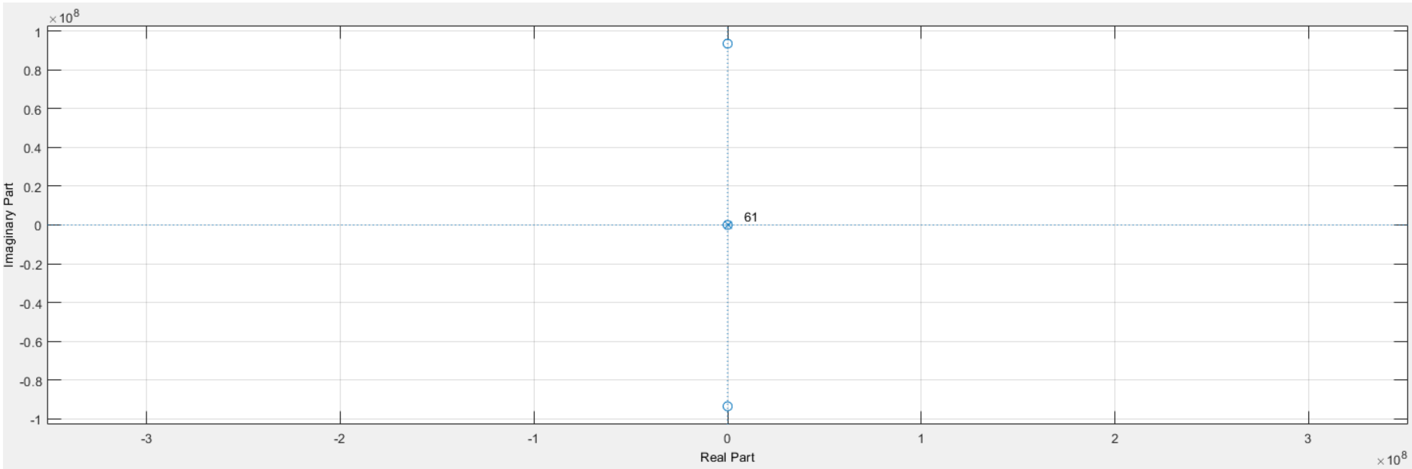
$f_{c_2} = \frac{.4 - .3}{2} = .35$


Amplitude
Samples


Magnitude
Normalized Frequency (×π rad/sample)


Magnitude (dB)
Normalized Frequency (×π rad/sample)


Phase (radians)
Normalized Frequency (×π rad/sample)

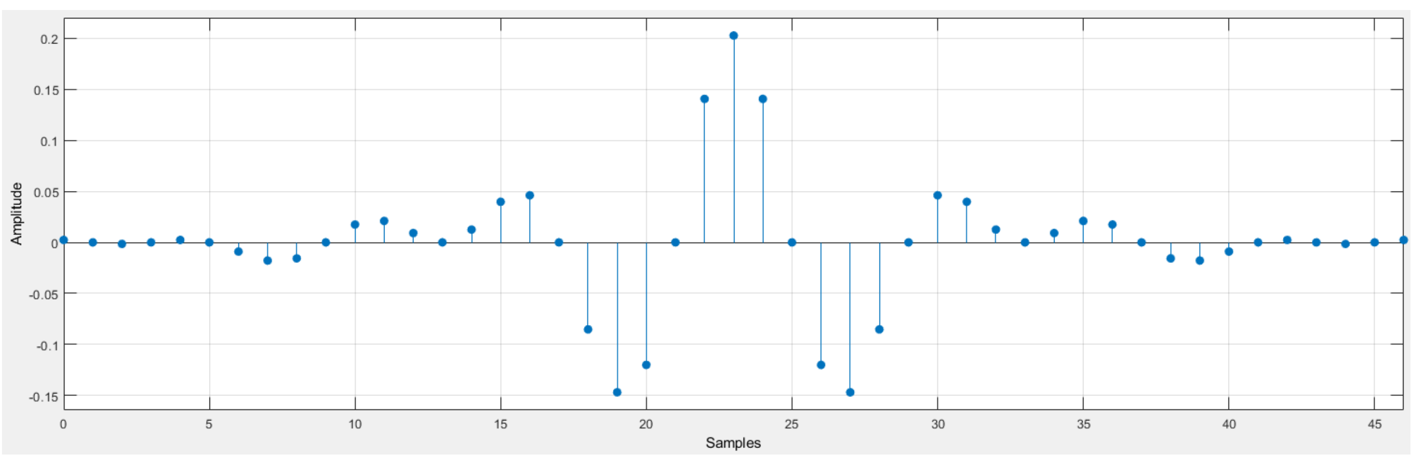


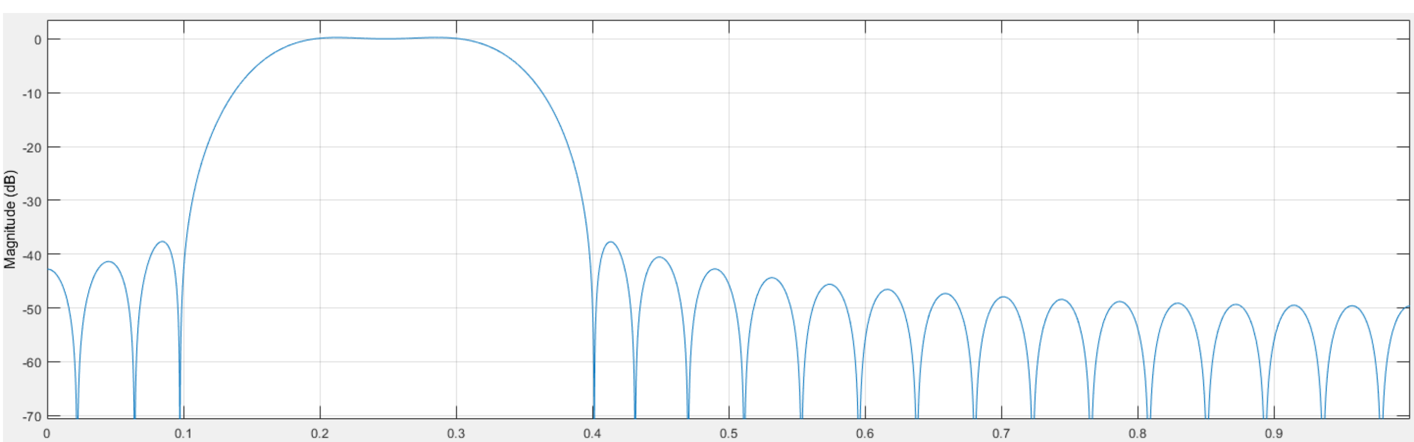
Kaiser: $A = -20 \log_{10}(.01) = 40$

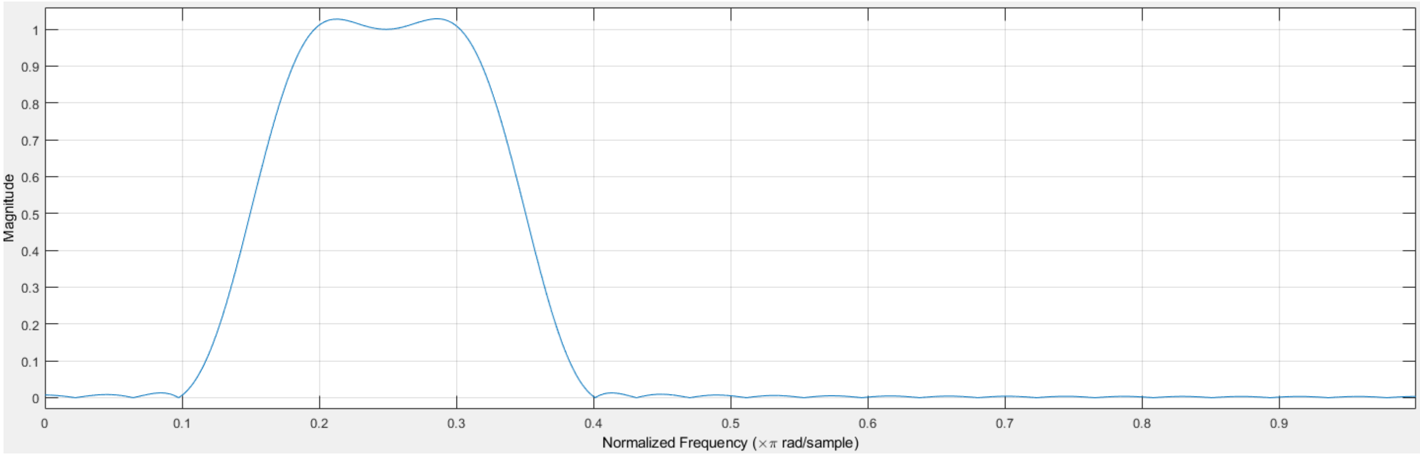
$\beta = .5842(40-21)^{.4} + .07886(40-21) = 3.395$

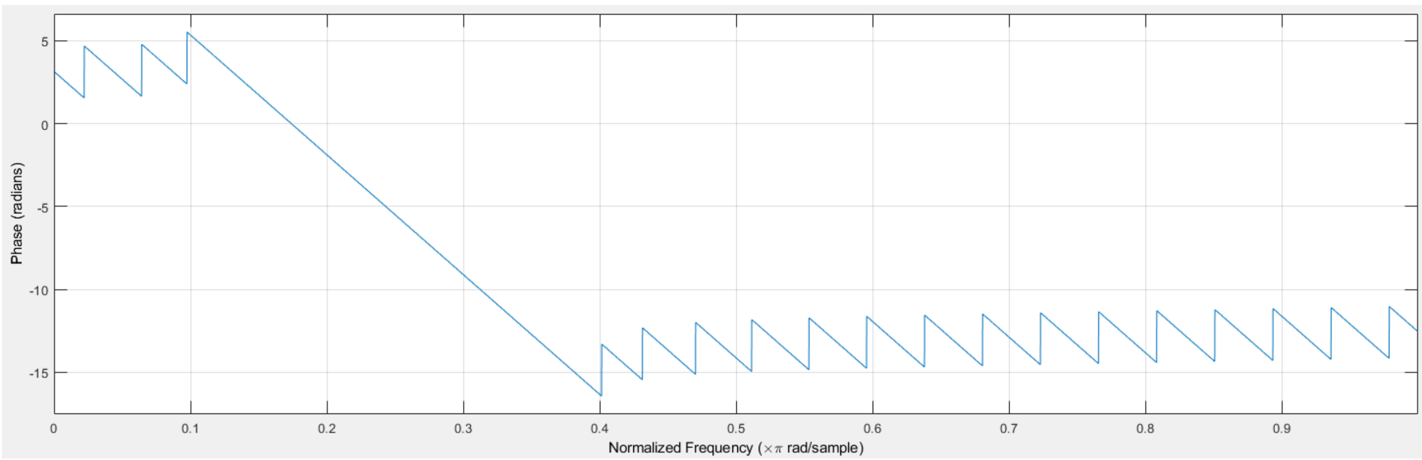
$M = \dfrac{A-8}{2.258\Delta\omega} = 44.577 \rightarrow 45$ (type II)

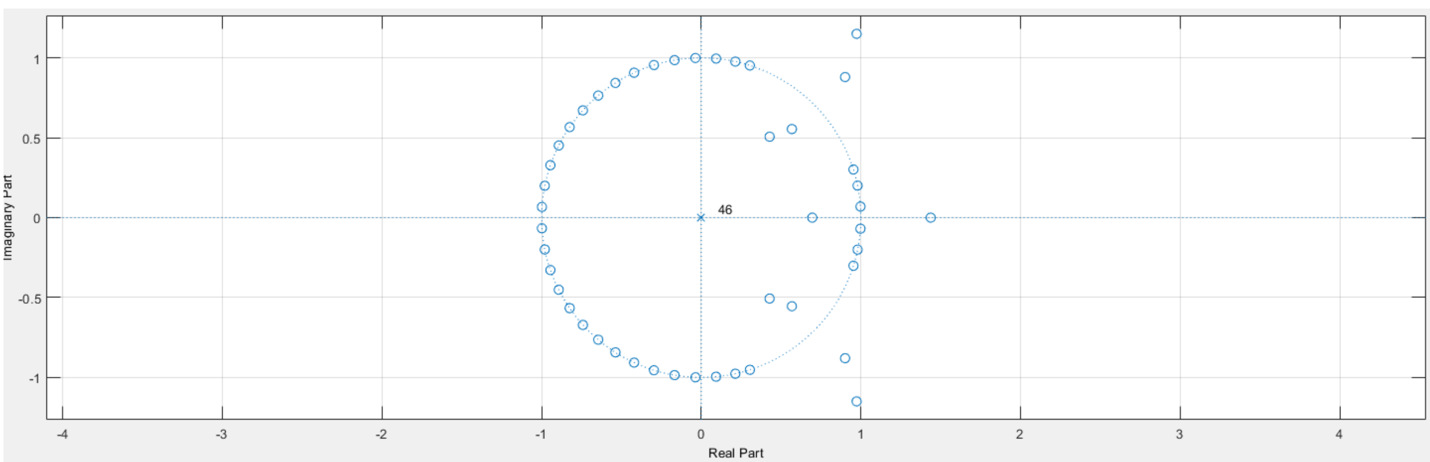
$L = M+1 = 46$



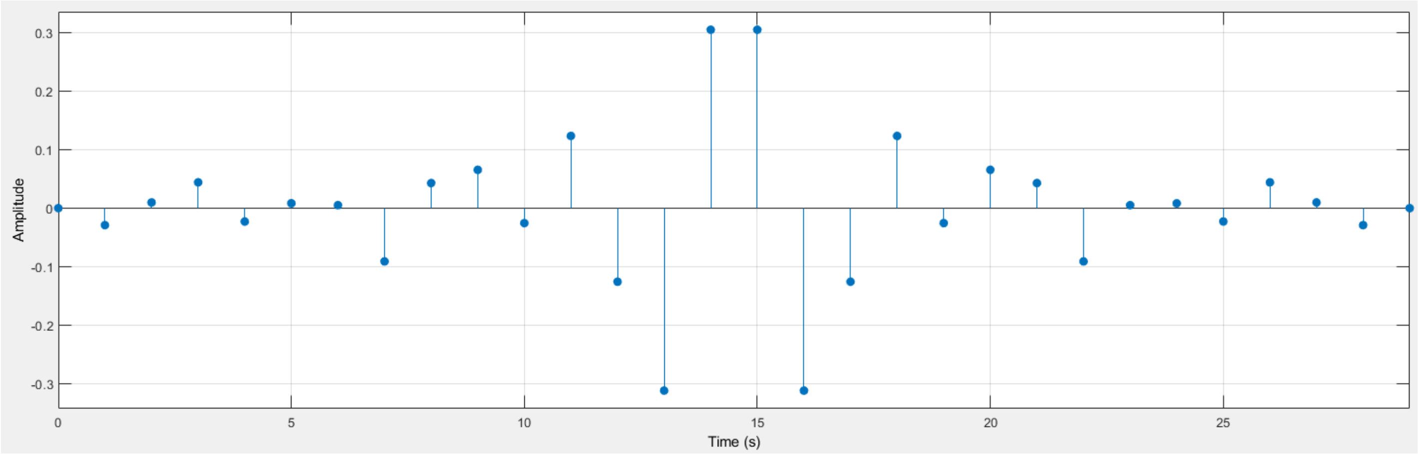



Magnitude
Normalized Frequency (×π rad/sample)

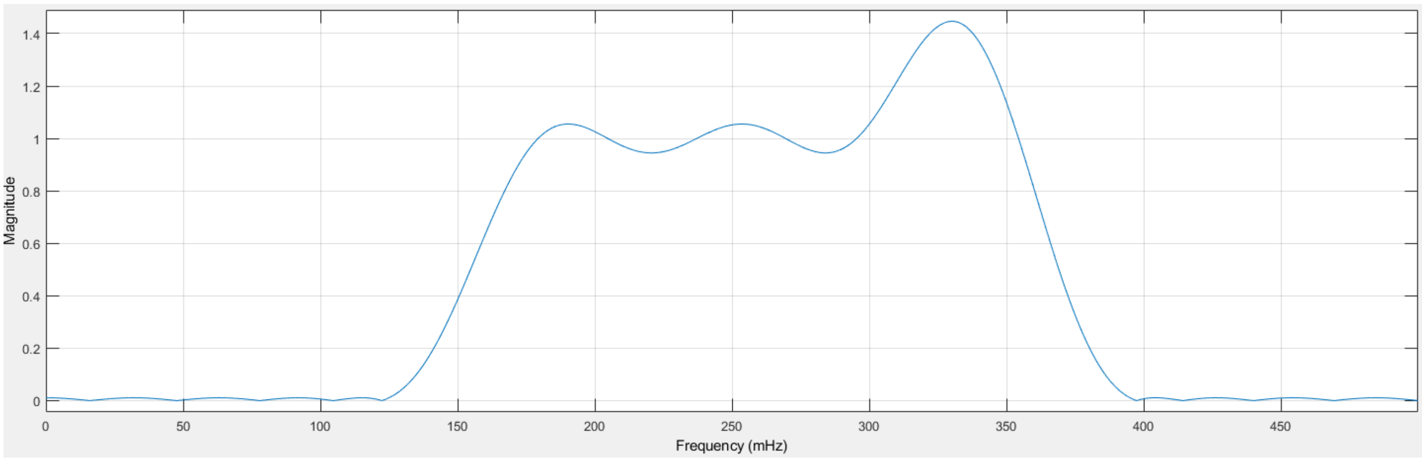

Phase (radians)
Normalized Frequency (×π rad/sample)


Imaginary Part
46
Real Part

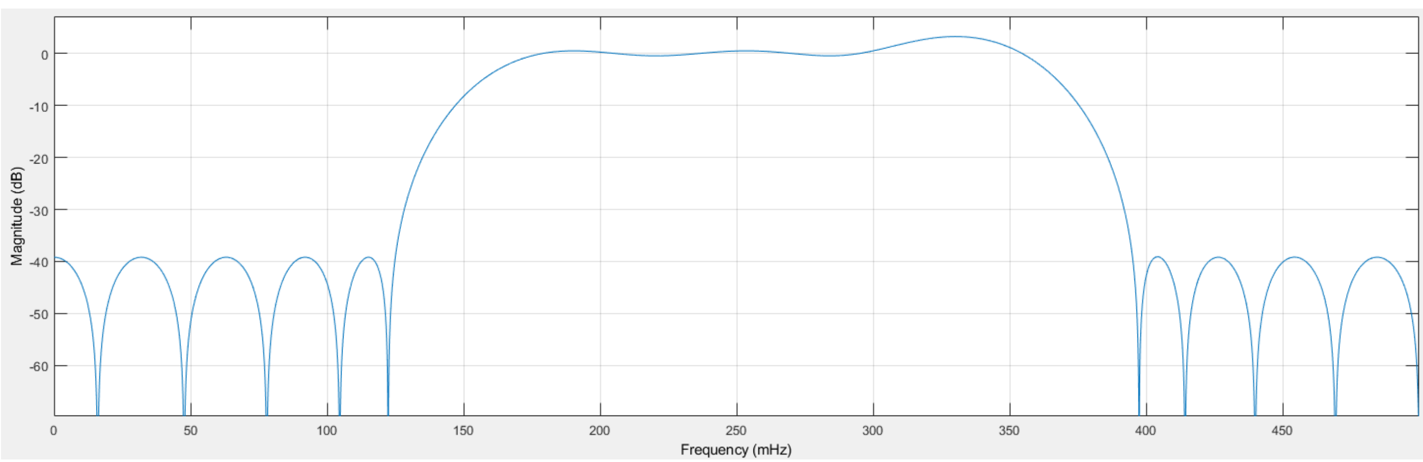
equiripple



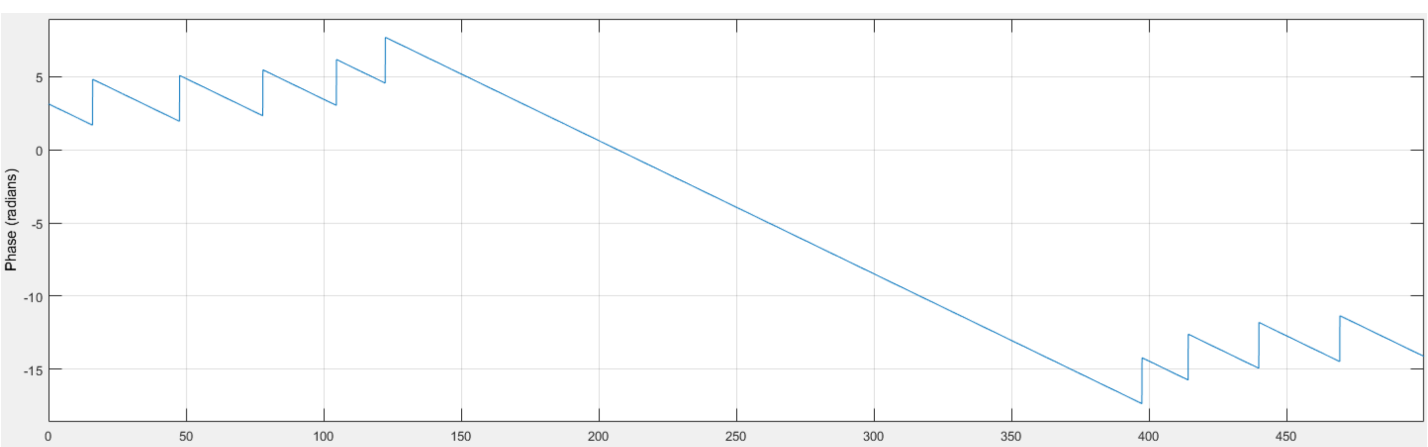
Amplitude
Time (s)



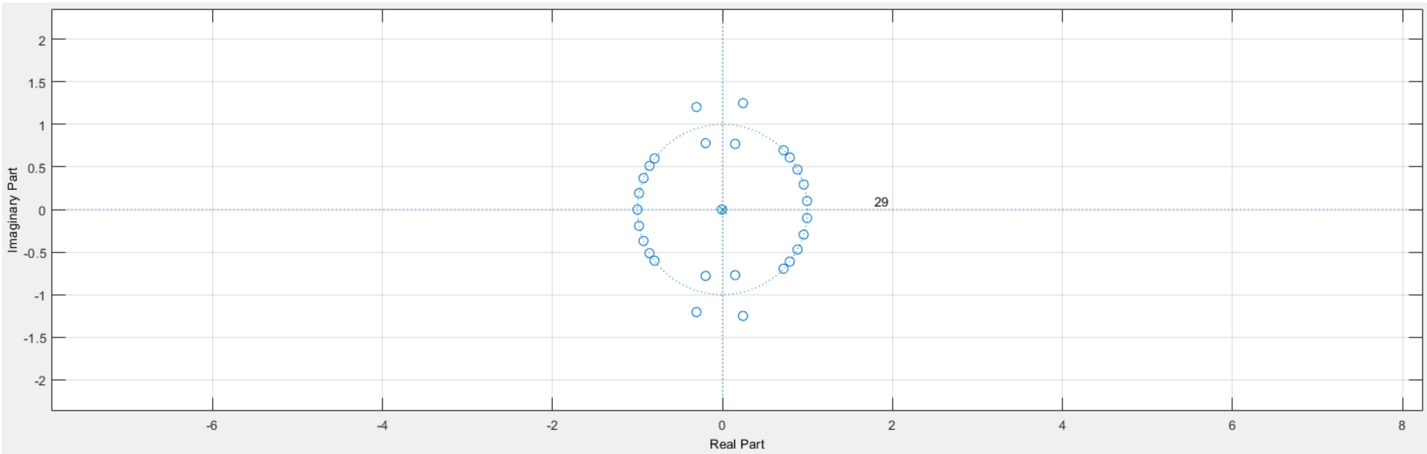
Magnitude
Frequency (mHz)



Magnitude (dB)
Frequency (mHz)



Phase (radians)
Frequency (mHz)



Filter	Length	Order	Group Delay	Type	δ or β	A
Hann	62	61	31	2	.01	-40
Kaiser	46	45	23	2	3.395	40
Equiripple	30	29	14.5	2	.01	40

2. This problem explores designing a lowpass filter with a very narrow transition band that achieves the following specifications: $A_p = 0.1$ dB ripple in the passband ($0 \le |f| \le 0.24$) and $A_s = 60$ dB suppression in the stopband ($0.26 \le |f| \le 0.5$).

- Design this filter using the window technique using the best window from the set {rectangular, Bartlett, Hann, Hamming, Blackman}. Indicate which window shape you used.
- Design this filter using the window technique using the Kaiser window. Indicate the Kaiser window parameters (β and M).
- Using FDATool in Matlab, design this filter using equiripple design method.

Make a table with one row for each filter you designed (window, Kaiser, equiripple). The table should have a column for the following parameters: length, order, group delay, type (I, II, III, or IV), and window parameters.

For each filter you design, turn in plots of the impulse response, magnitude response (linear and dB), and phase response. The frequency domain plots should show response over the interval $-\frac{1}{2} \le f \le \frac{1}{2}$. Also include a pole-zero plot.

$A_p = -.1\, dB$

$$\delta_p = \frac{10^{\frac{.1}{20}} - 1}{10^{\frac{.1}{20}} + 1} = .005756$$

$A_s = 60\, dB \rightarrow$ Blackman

$\delta_s = 10^{\frac{-60}{20}} = .001$

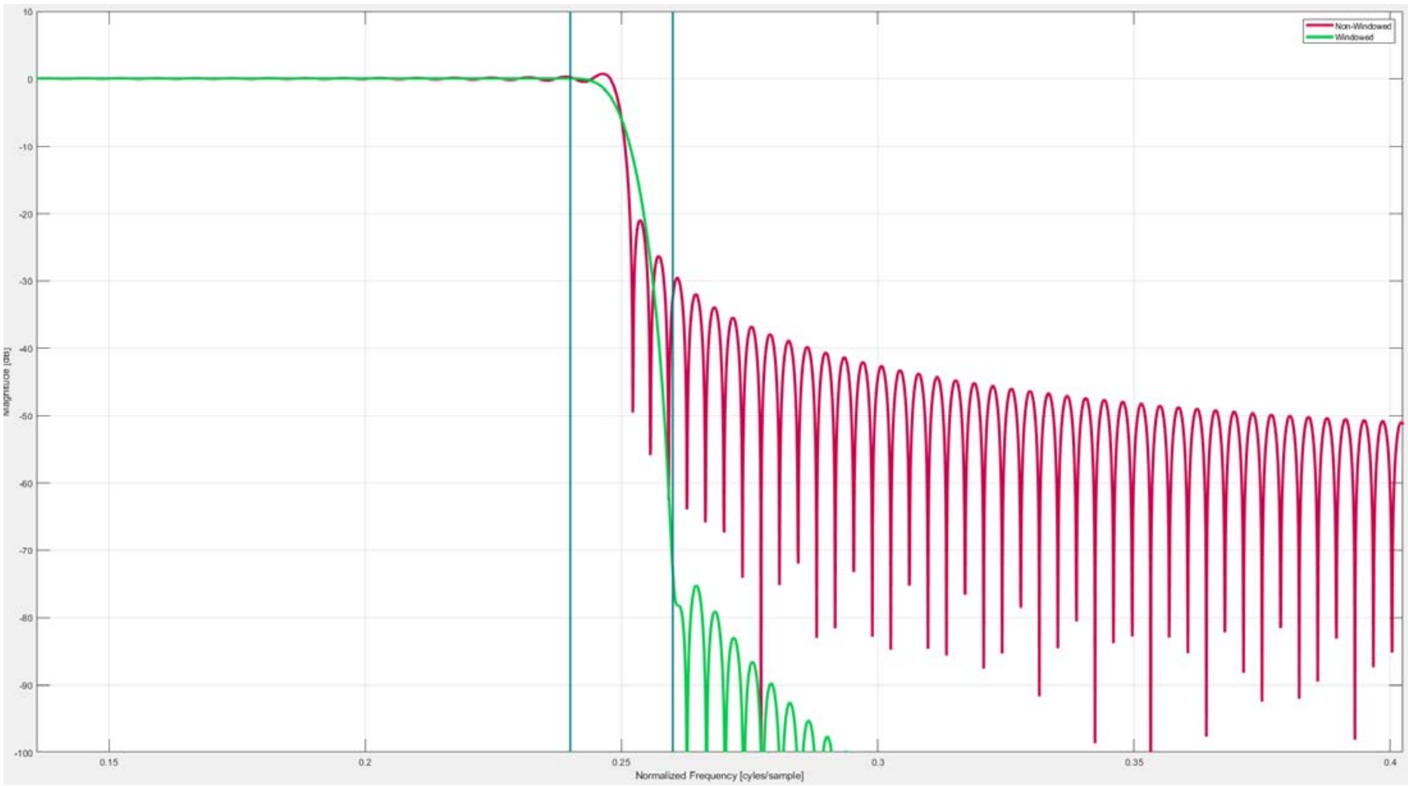
$\Delta f = .26 - .24 = .02$

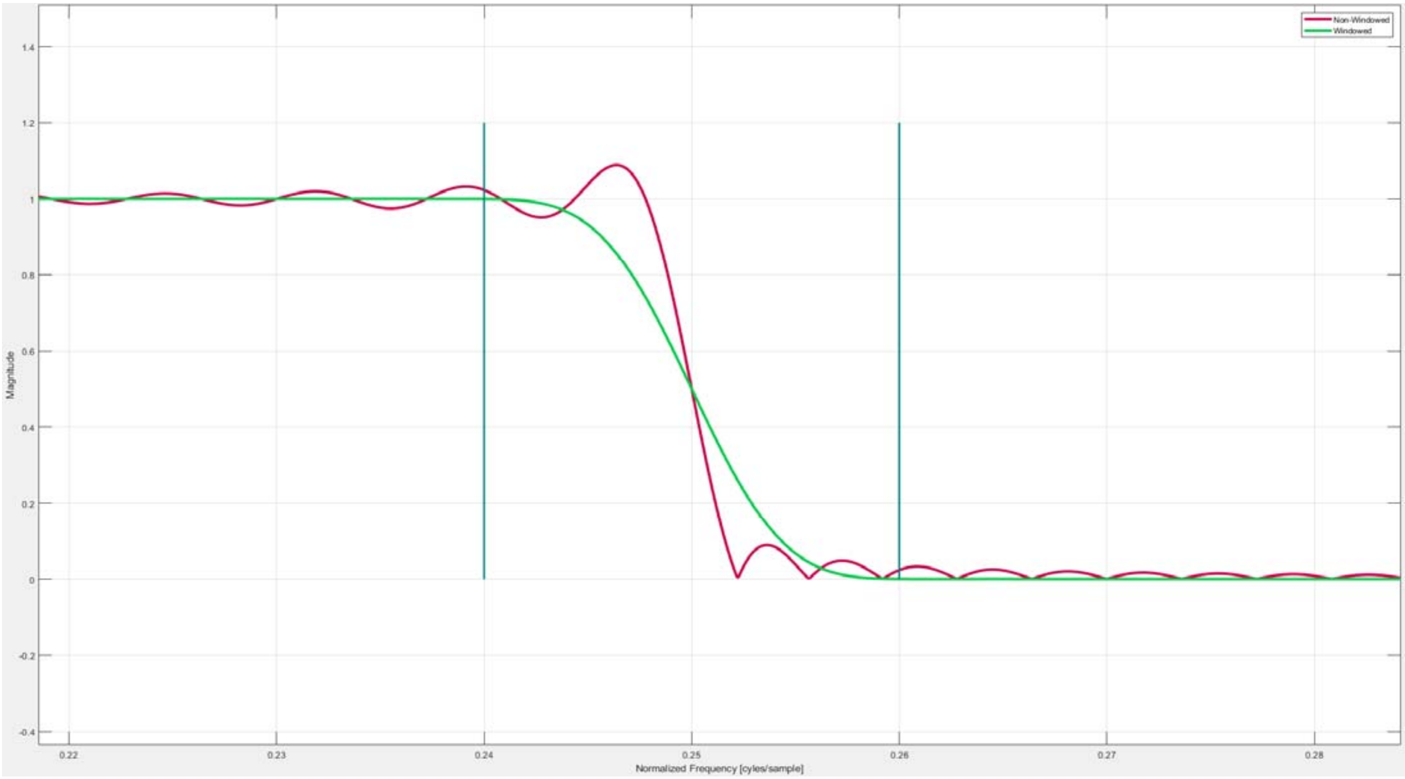
$\Delta\omega = 2\pi \cdot .02$

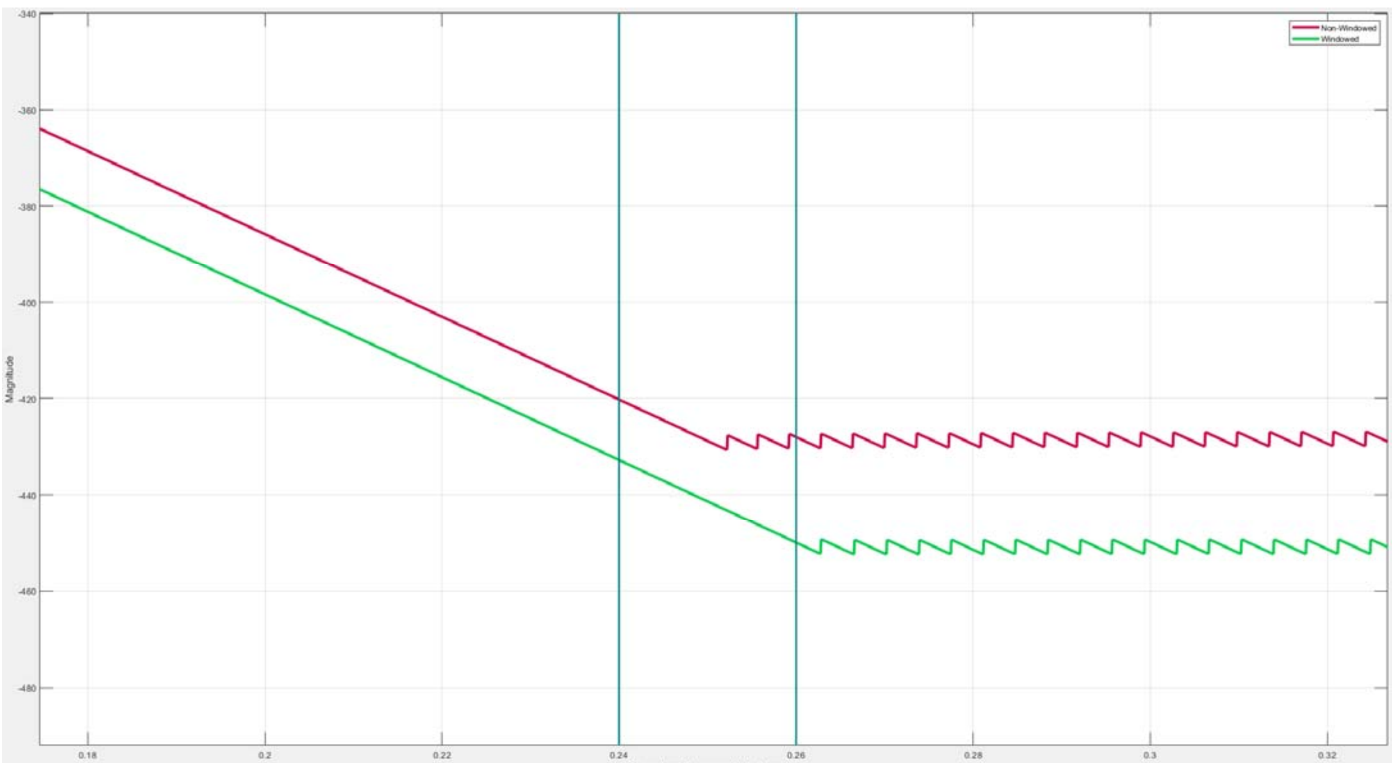
$L = \frac{11\pi}{\Delta\omega} = 274.\bar{9} \rightarrow 275$

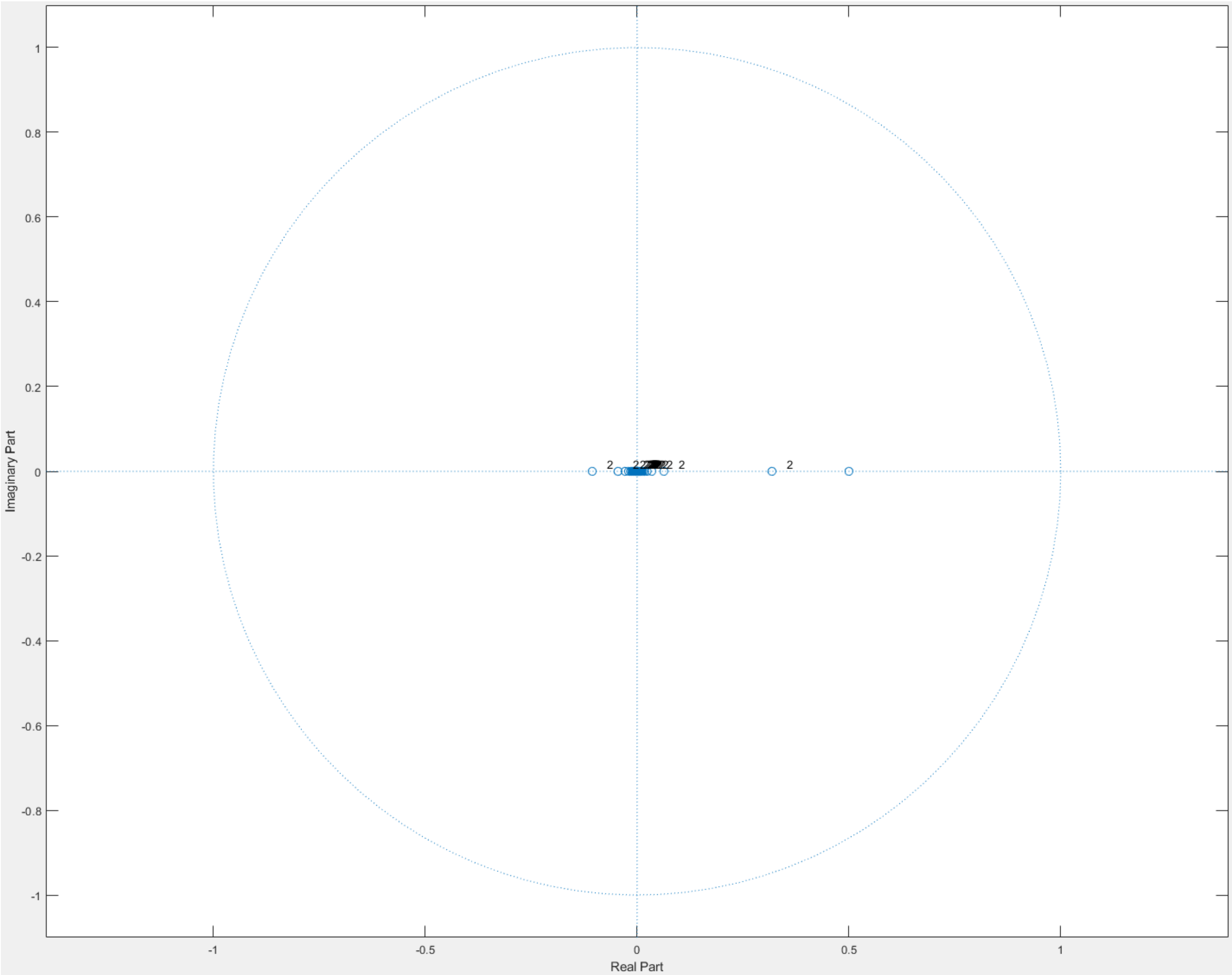
$M = 274$ (type I)

$f_c = .25$




Non-Windowed
Windowed
Magnitude
Normalized Frequency [cyles/sample]


Non-Windowed
Windowed
Magnitude
Normalized Frequency [cyles/sample]



Kaiser:

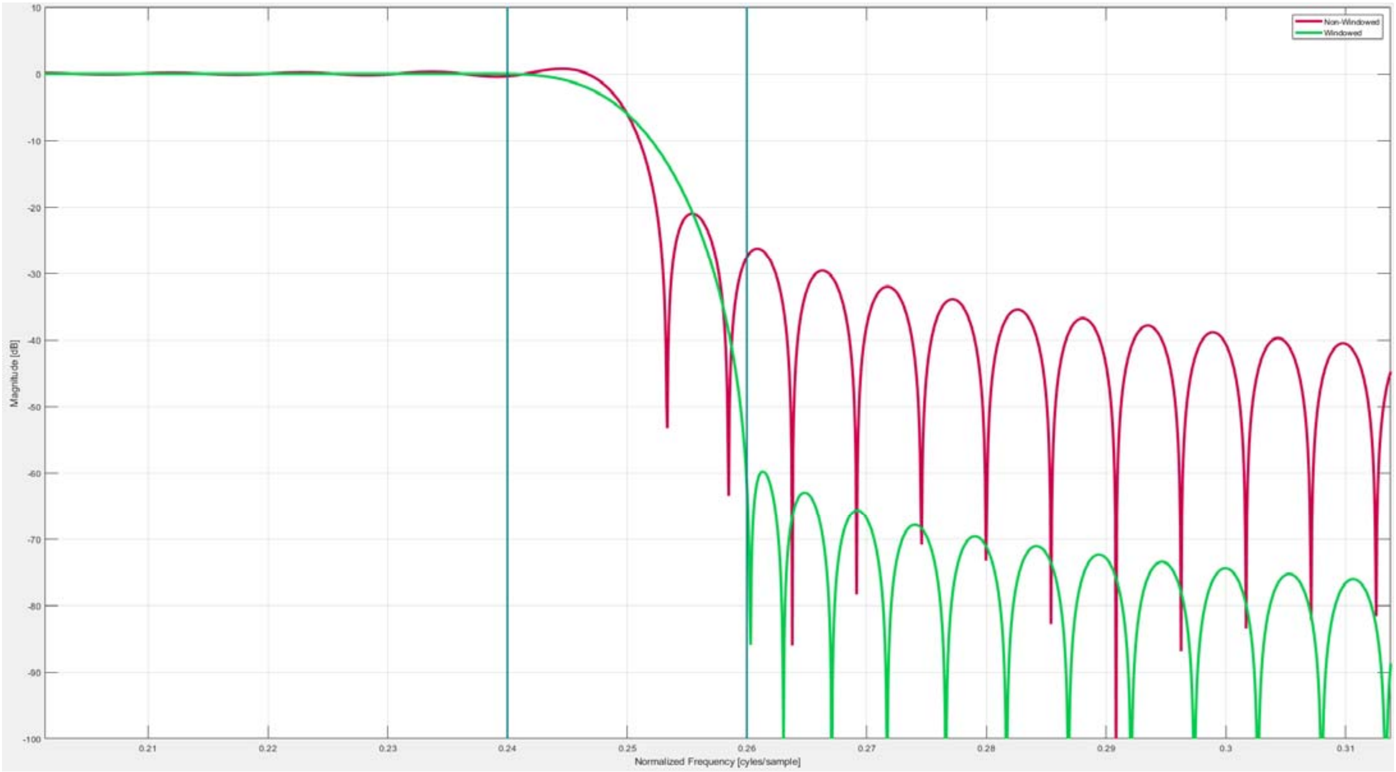
$$A = -20 \log_{10}(.001) = 60$$

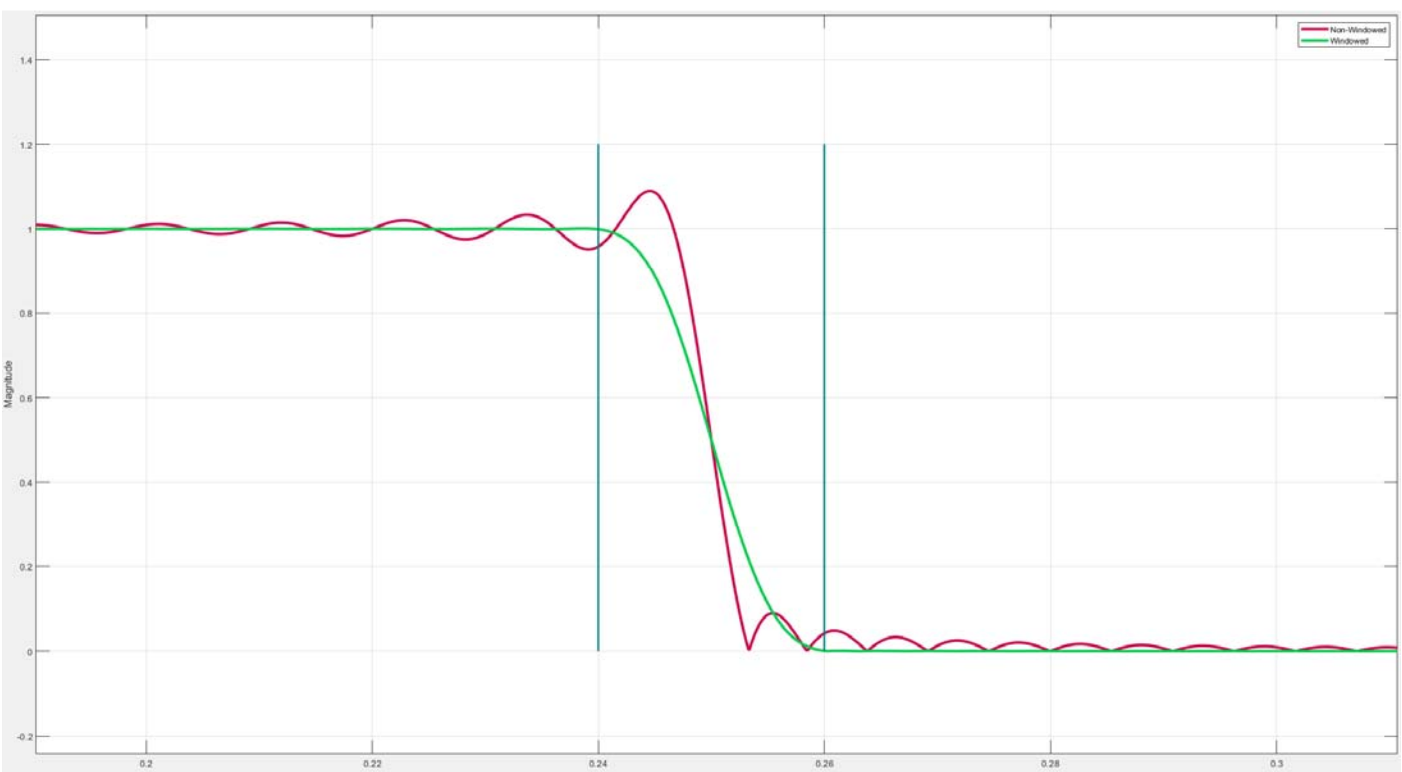
$$\beta = .1102(60-8.7) = 5.653$$

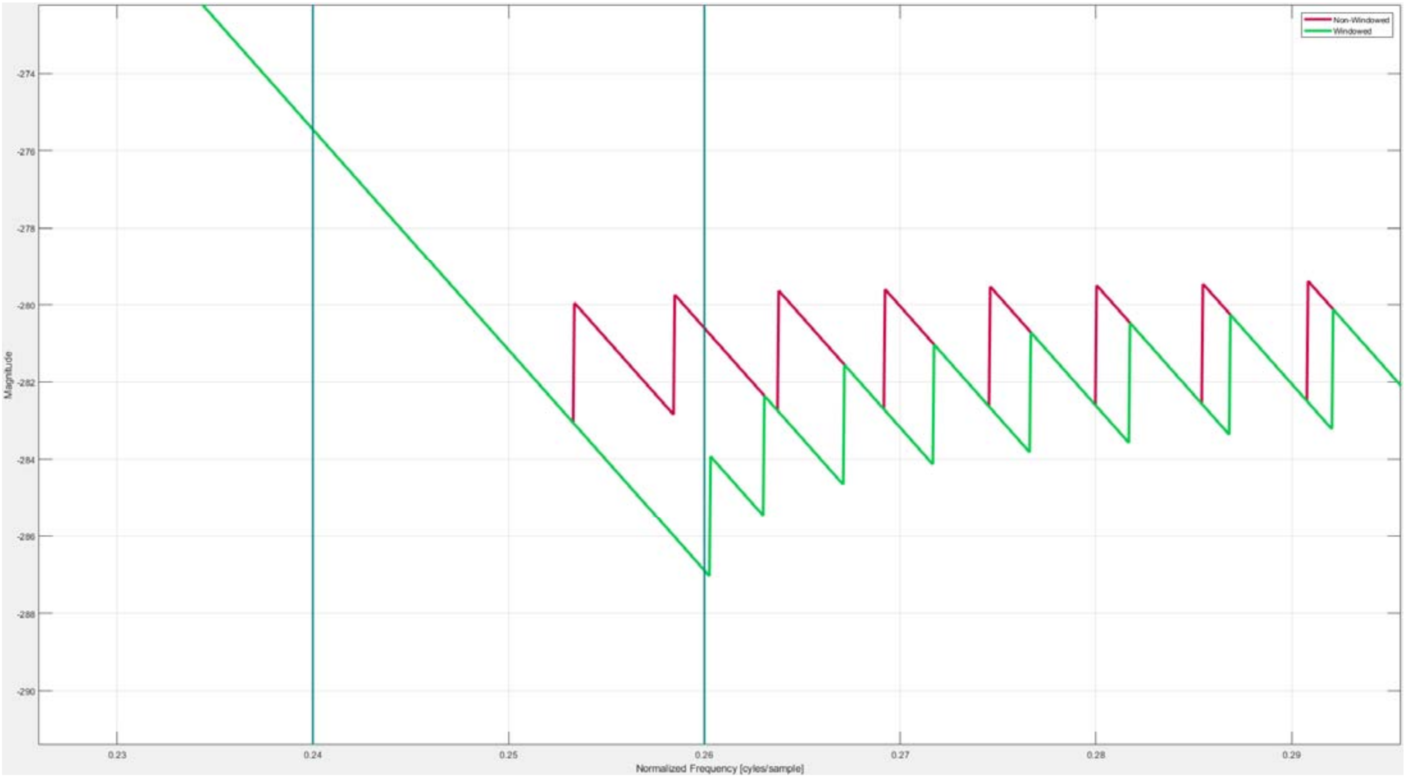
$$M = \frac{60-8}{2.285 \cdot 2\pi \cdot .02} = 181.1 \rightarrow 182$$

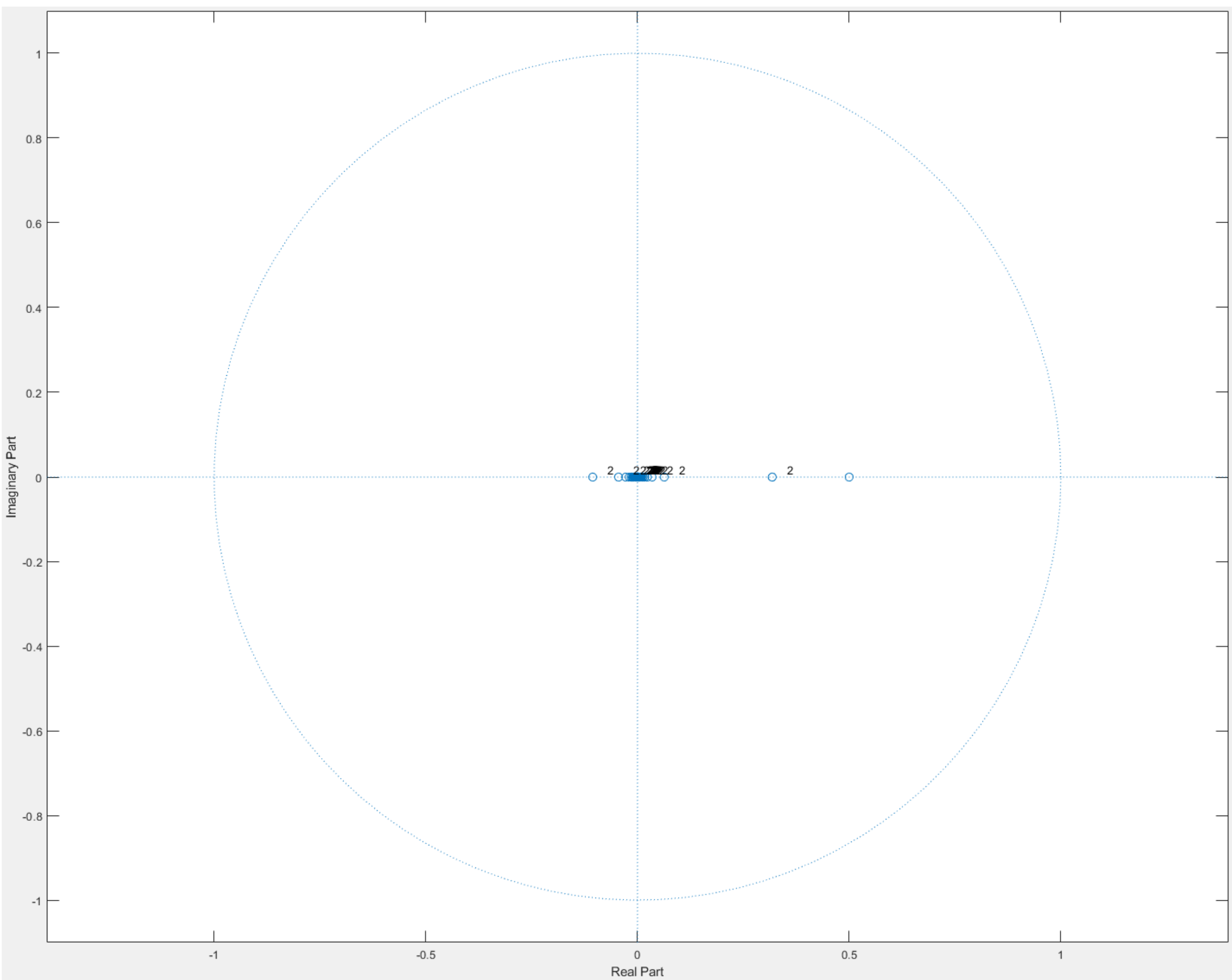
(type I)

$$L = 183$$

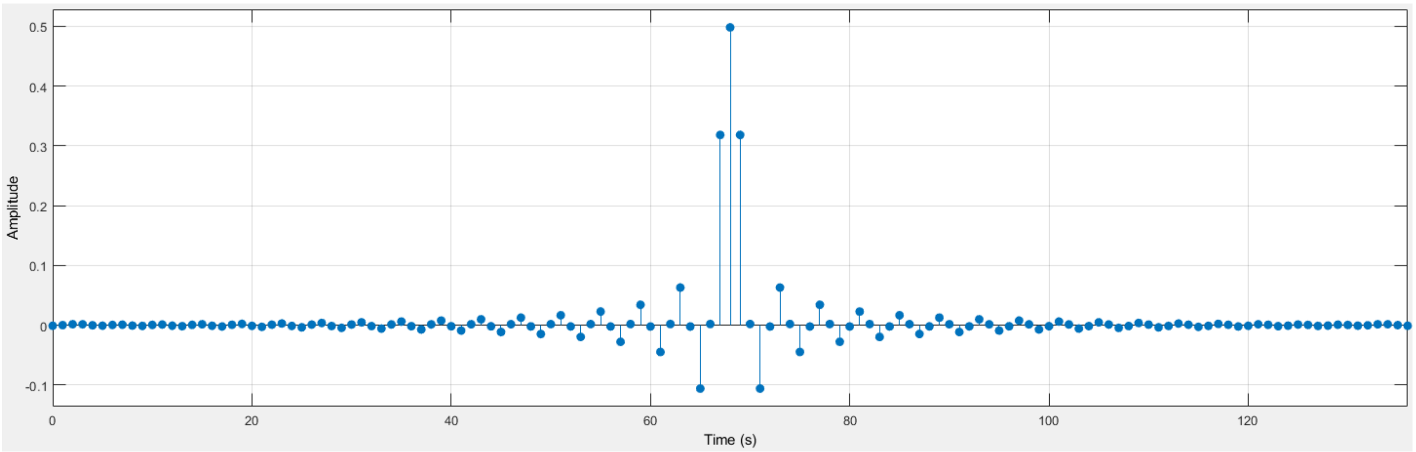

Non-Windowed
Windowed
Magnitude [dB]
Normalized Frequency [cyles/sample]


Non-Windowed
Windowed
Magnitude
Normalized Frequency [cyles/sample]

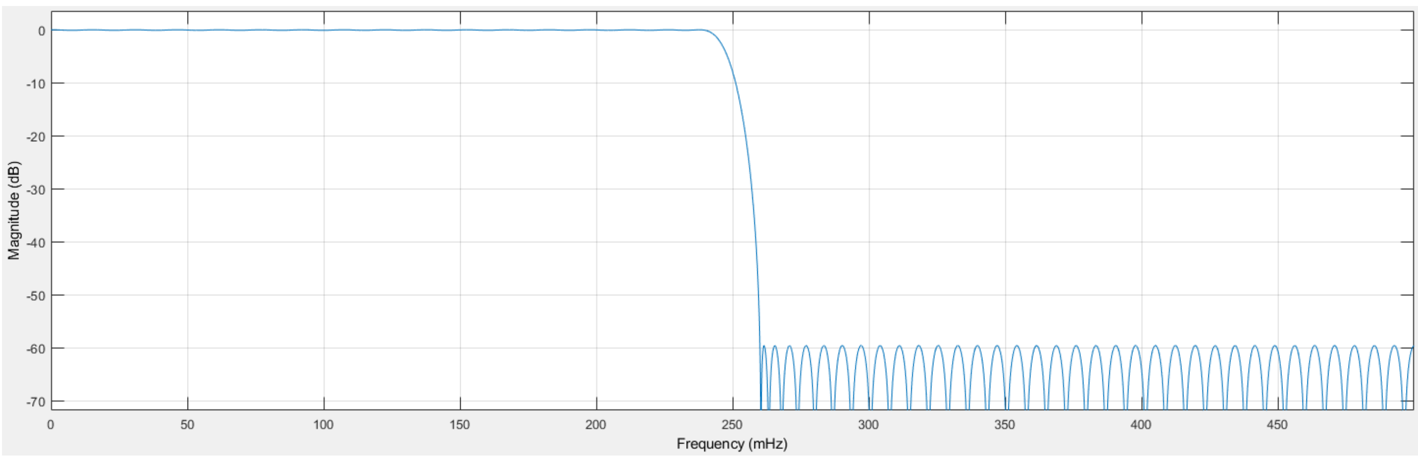

Non-Windowed
Windowed
Magnitude
Normalized Frequency [cyles/sample]
-274
-276
-278
-280
-282
-284
-286
-288
-290
0.23
0.24
0.25
0.26
0.27
0.28
0.29


Imaginary Part
Real Part
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.5
0.5
2

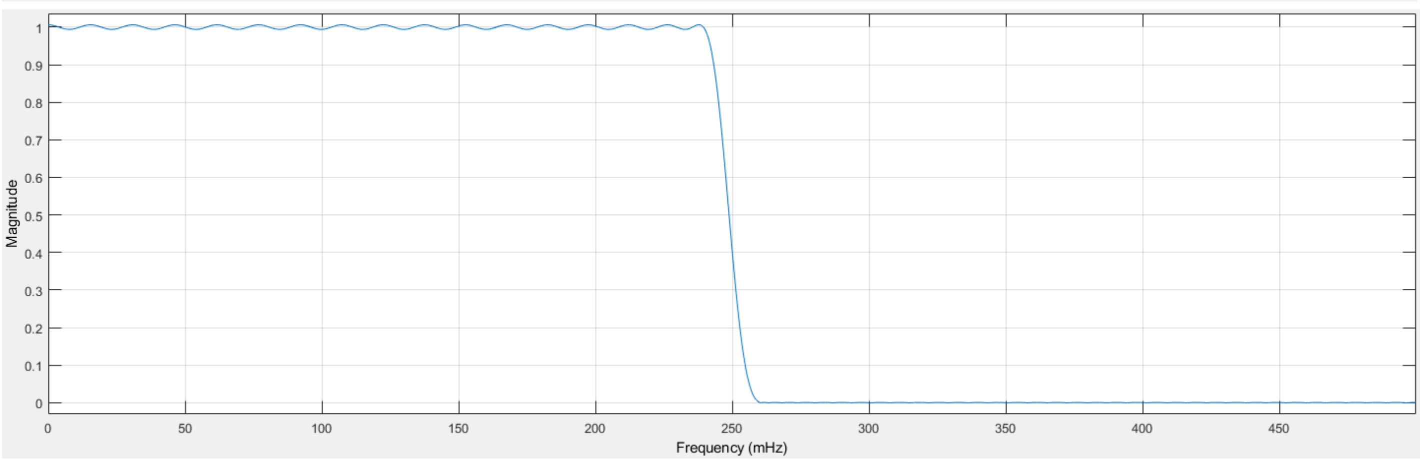
Equiripple:



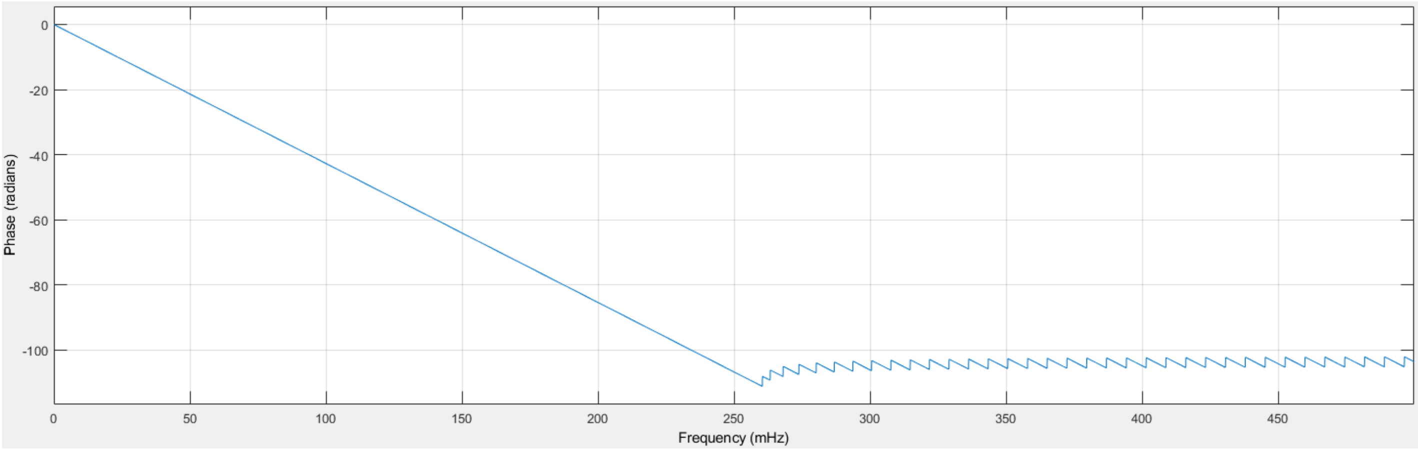
Amplitude
Time (s)

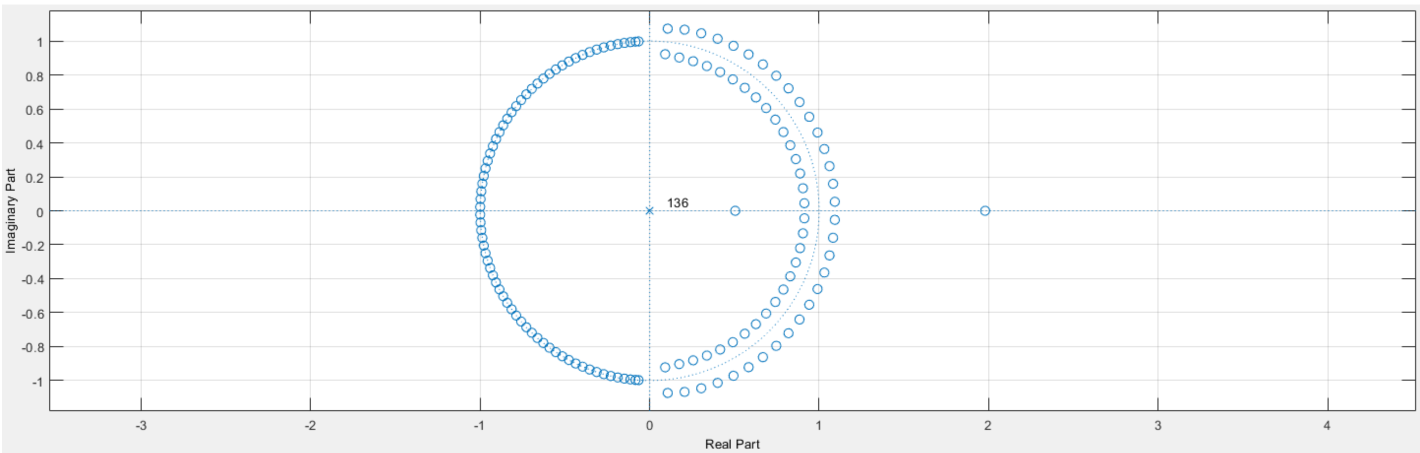


Magnitude (dB)
Frequency (mHz)



Magnitude
Frequency (mHz)





Filter	Length	Order	Group Delay	Type	δ or β	A
Hann	275	274	137	1	.001	60
Kaiser	183	182	91	1	5.653	60
Equiripple	137	136	68	1	.001	60

3. This problem explores designing a lowpass filter for sample rate conversion using the system depicted in Fig. 6.47 (c) in the textbook (page 379) in which the upsampling factor is $U = 3$ and the downsampling factor is $D = 4$. The maximum frequency in the signal $x[n]$ is $f_{\text{max,x}} = \frac{1}{3}$. Design the lowpass filter to have $A_p = 0.1$ dB of ripple in the pass band and $A_s = 60$ dB of attenuation in the stopband. Use your knowledge of upsampling and downsampling to determine the passband and stopband edge frequencies, f_p and f_s. What are f_p and f_s?

- Design this filter using the window technique using the best window from the set {rectangular, Bartlett, Hann, Hamming, Blackman}. Indicate which window shape you used.
- Design this filter using the window technique using the Kaiser window. Indicate the Kaiser window parameters (β and M).
- Using FDATool in Matlab, design this filter using equiripple design method.

Make a table with one row for each filter you designed (window, Kaiser, equiripple). The table should have a column for the following parameters: length, order, group delay, type (I, II, III, or IV), and window parameters.

For each filter you design, turn in plots of the impulse response, magnitude response (linear and dB), and phase response. The frequency domain plots should show response over the interval $-\frac{1}{2} \leq f \leq \frac{1}{2}$. Also include a pole-zero plot.

$A_p = .1\,dB$

$$\delta_p = \frac{10^{\frac{.1}{20}} - 1}{10^{\frac{.1}{20}} + 1} = .005756$$

$A_s = 60\,dB \rightarrow$ Blackman

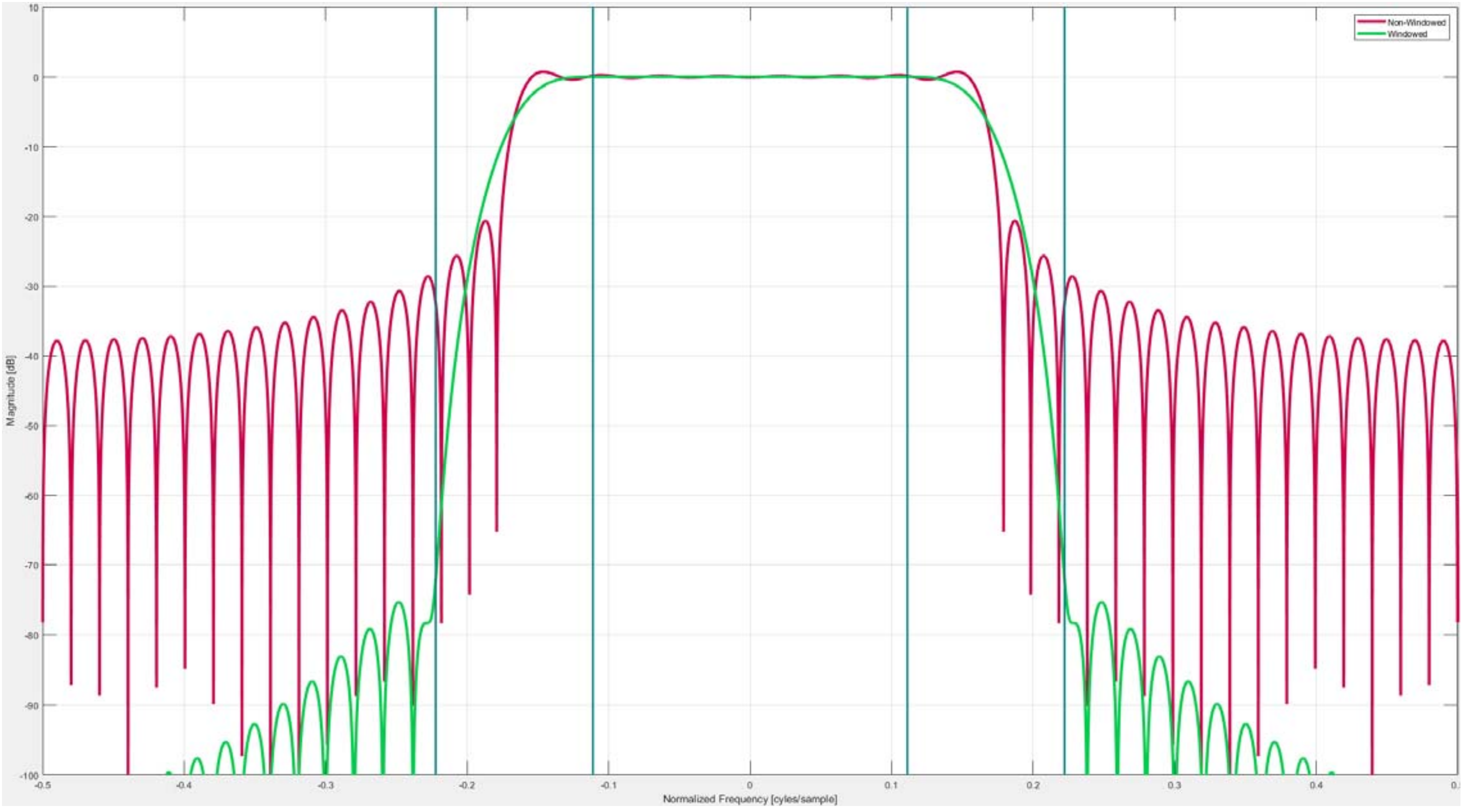
$\delta_s = 10^{\frac{-60}{20}} = .001$

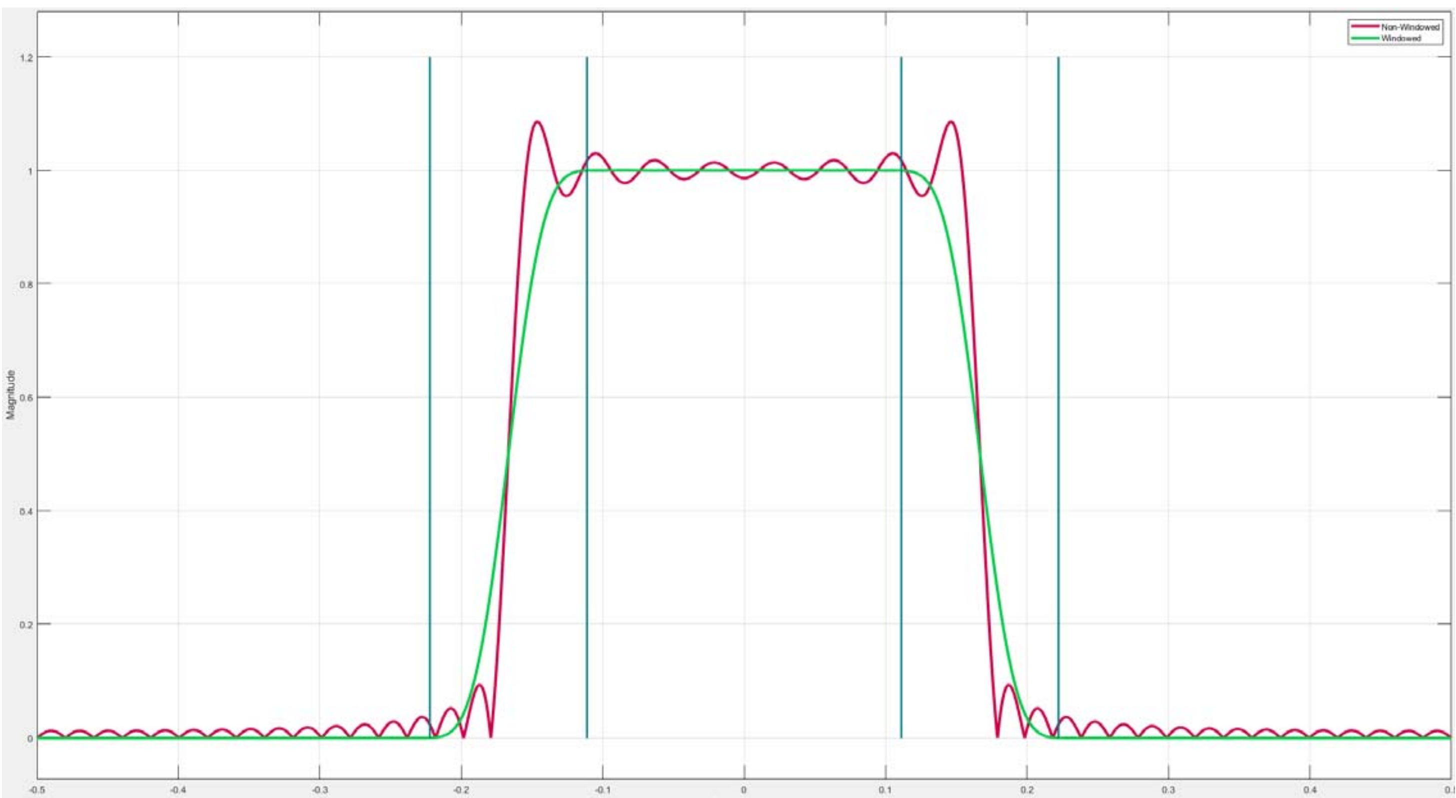
$\Delta f = \frac{2}{9} - \frac{1}{9} = \frac{1}{9}$

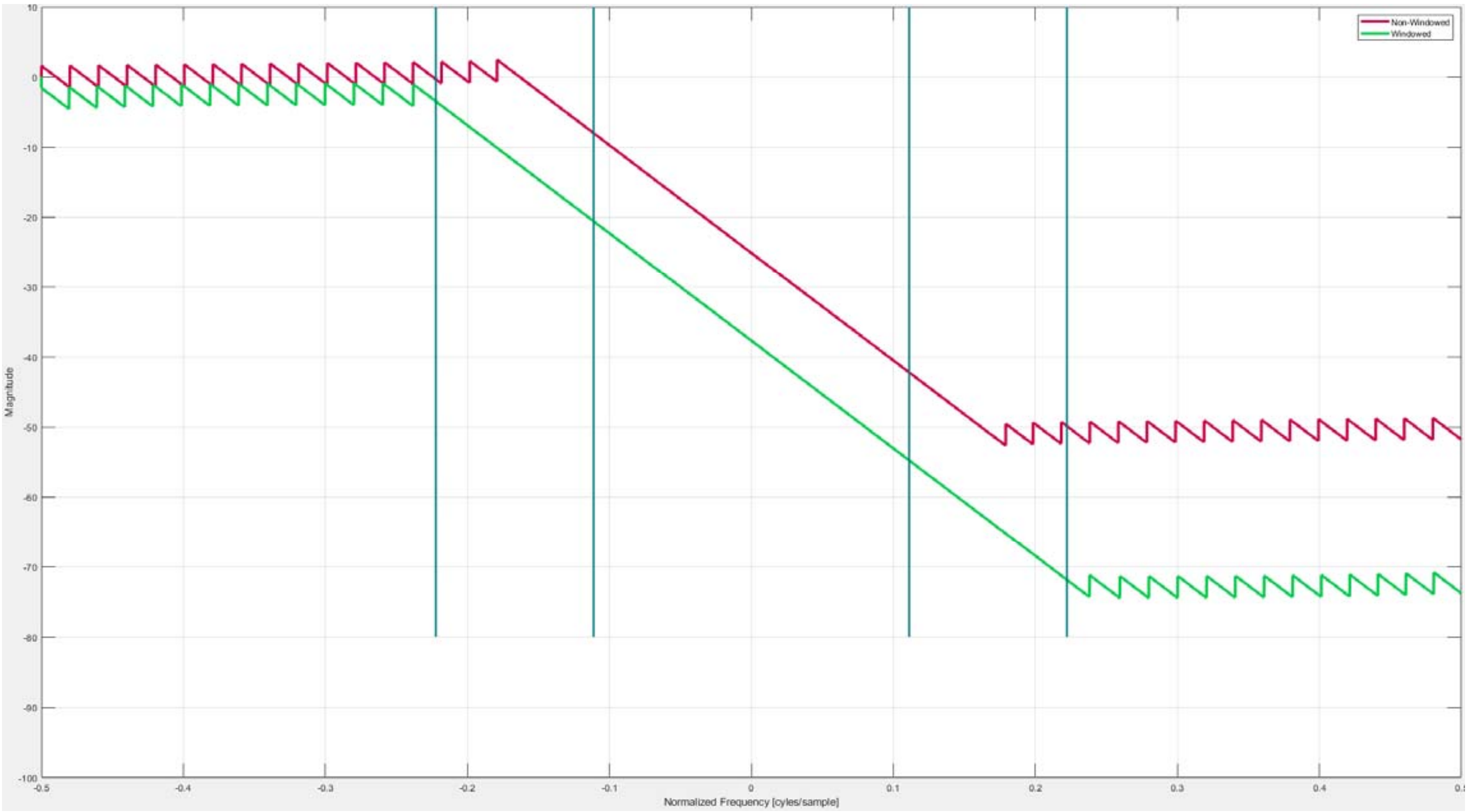
$\Delta \omega = \frac{2}{9}\pi$

$$L = \frac{11\pi}{\frac{2}{9}\pi} = 49.5 \rightarrow 50$$

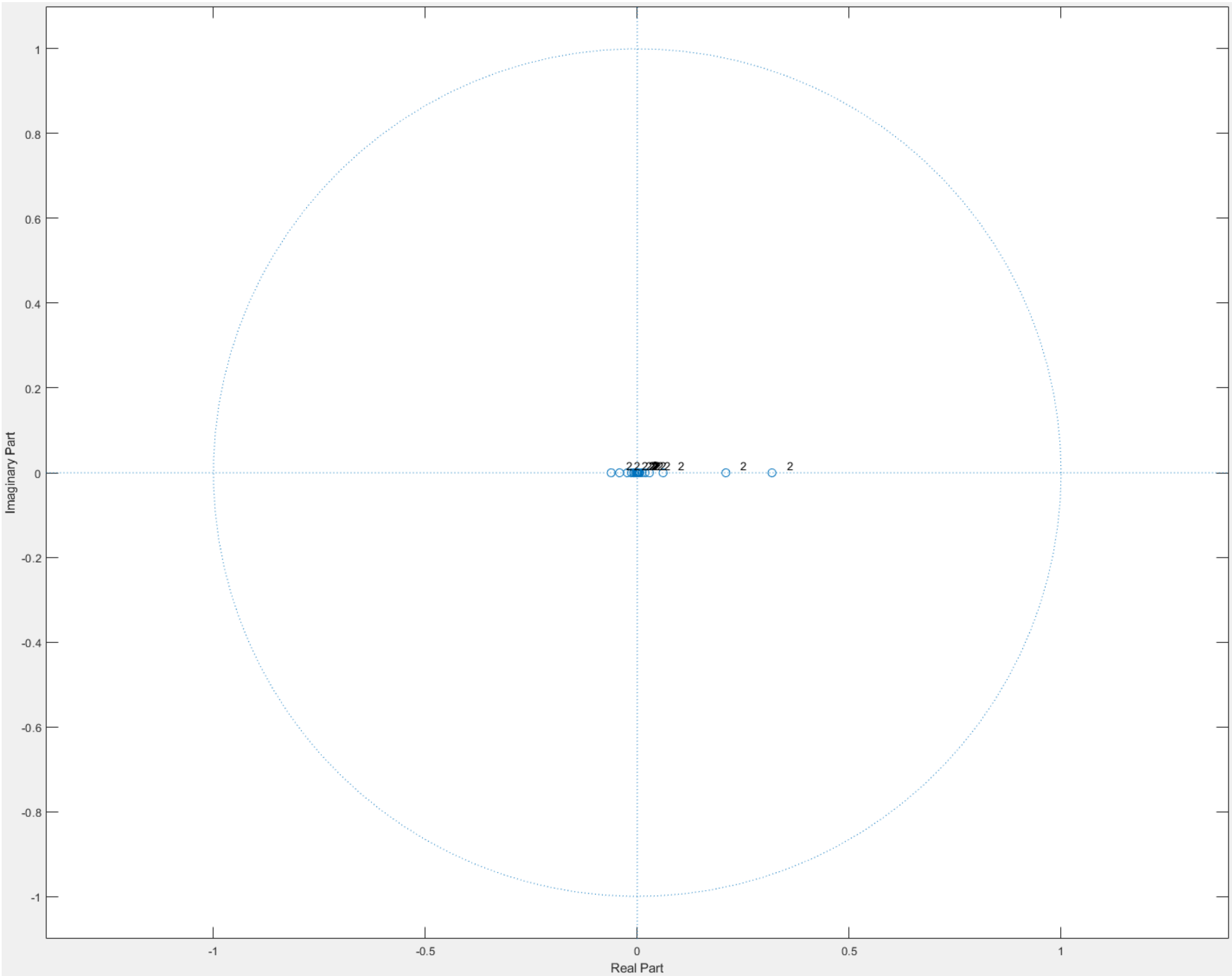
$M = 49$ (type II)


Non-Windowed
Windowed
Magnitude [dB]
Normalized Frequency [cyles/sample]


Non-Windowed
Windowed
Magnitude
Normalized Frequency [cyles/sample]



Non-Windowed
Windowed
Magnitude
Normalized Frequency [cyles/sample]



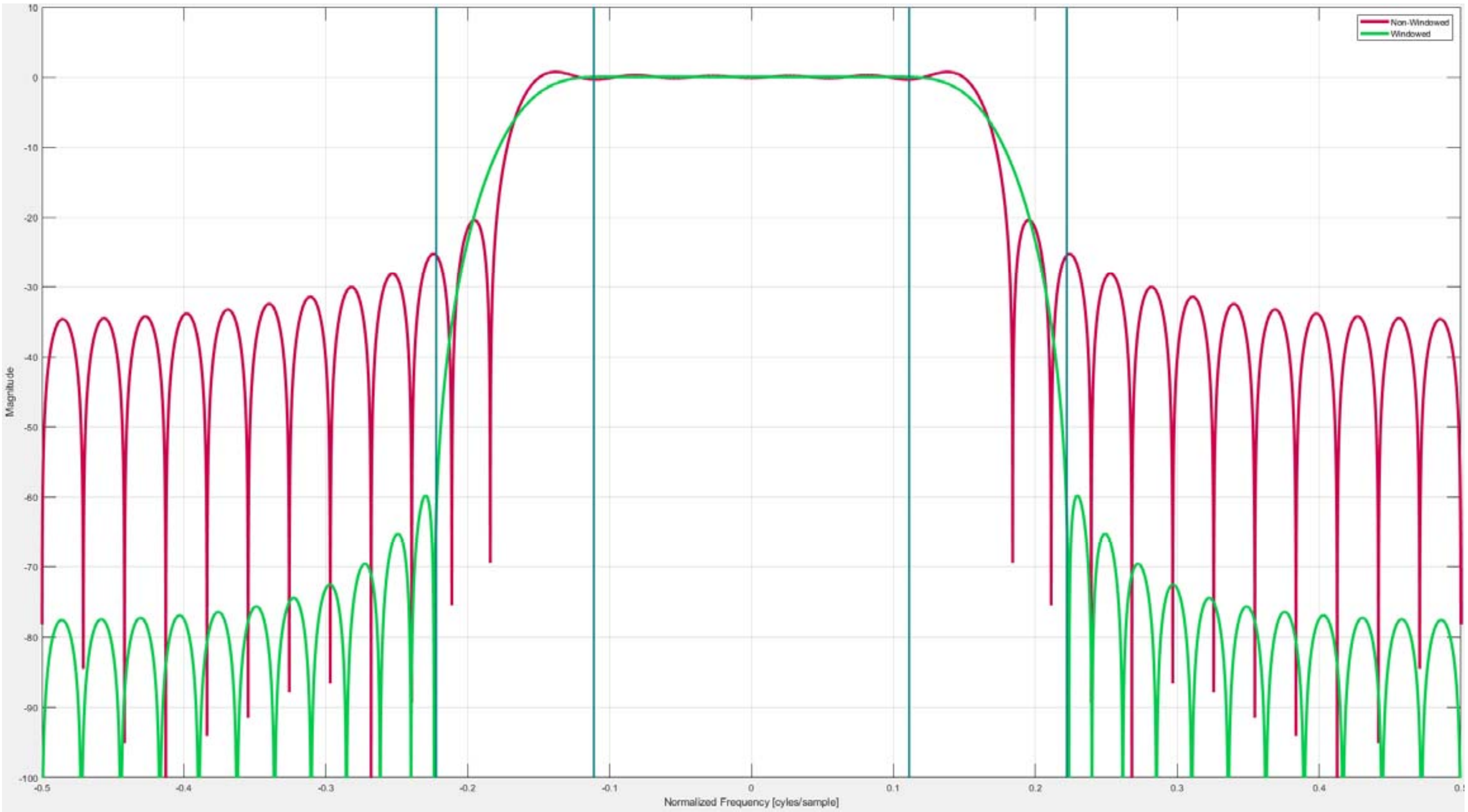
Kaiser:

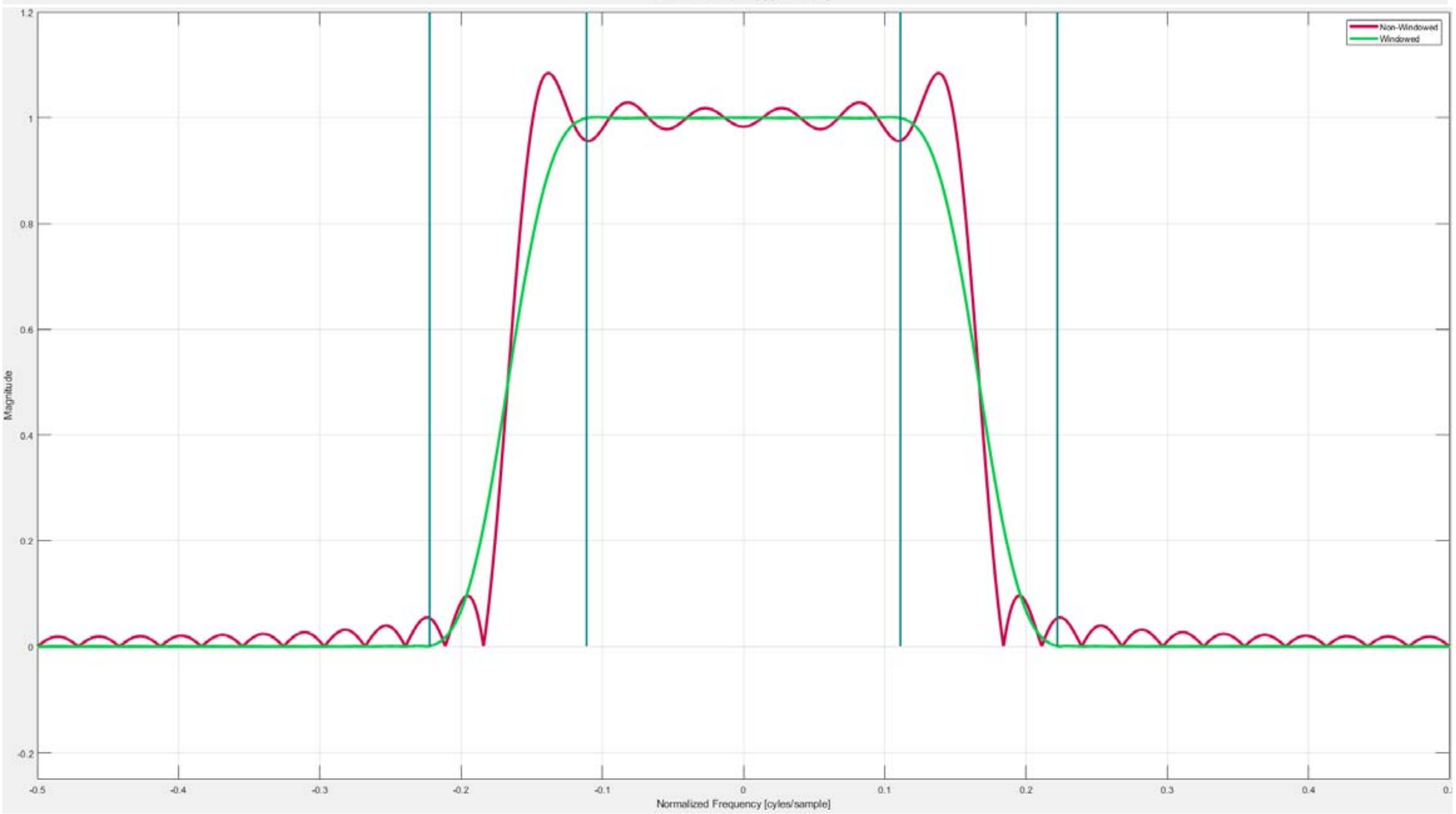
$A = -20 \log_{10}(.001) = 60$

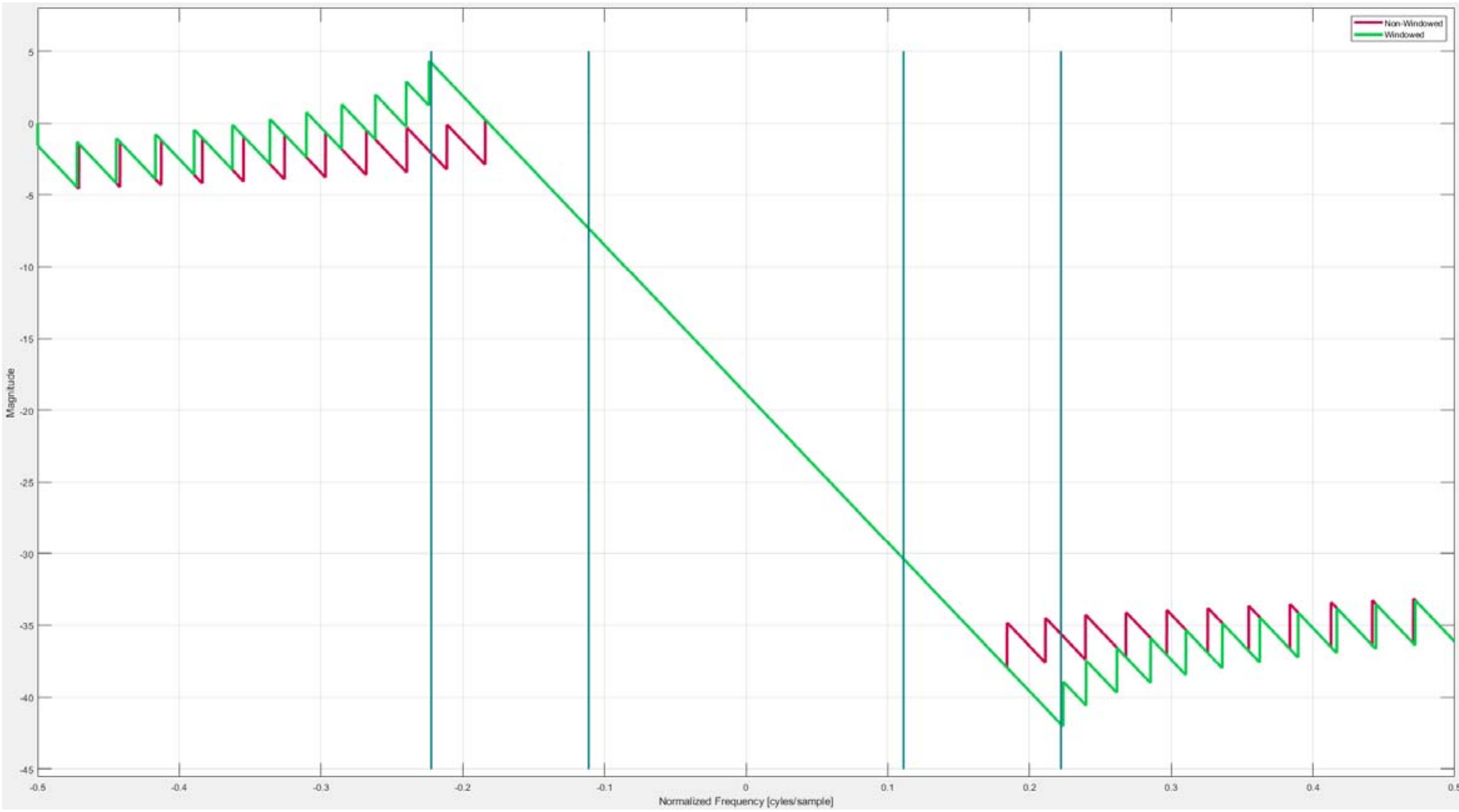
$\beta = .1102(60 - 8.7) = 5.653$

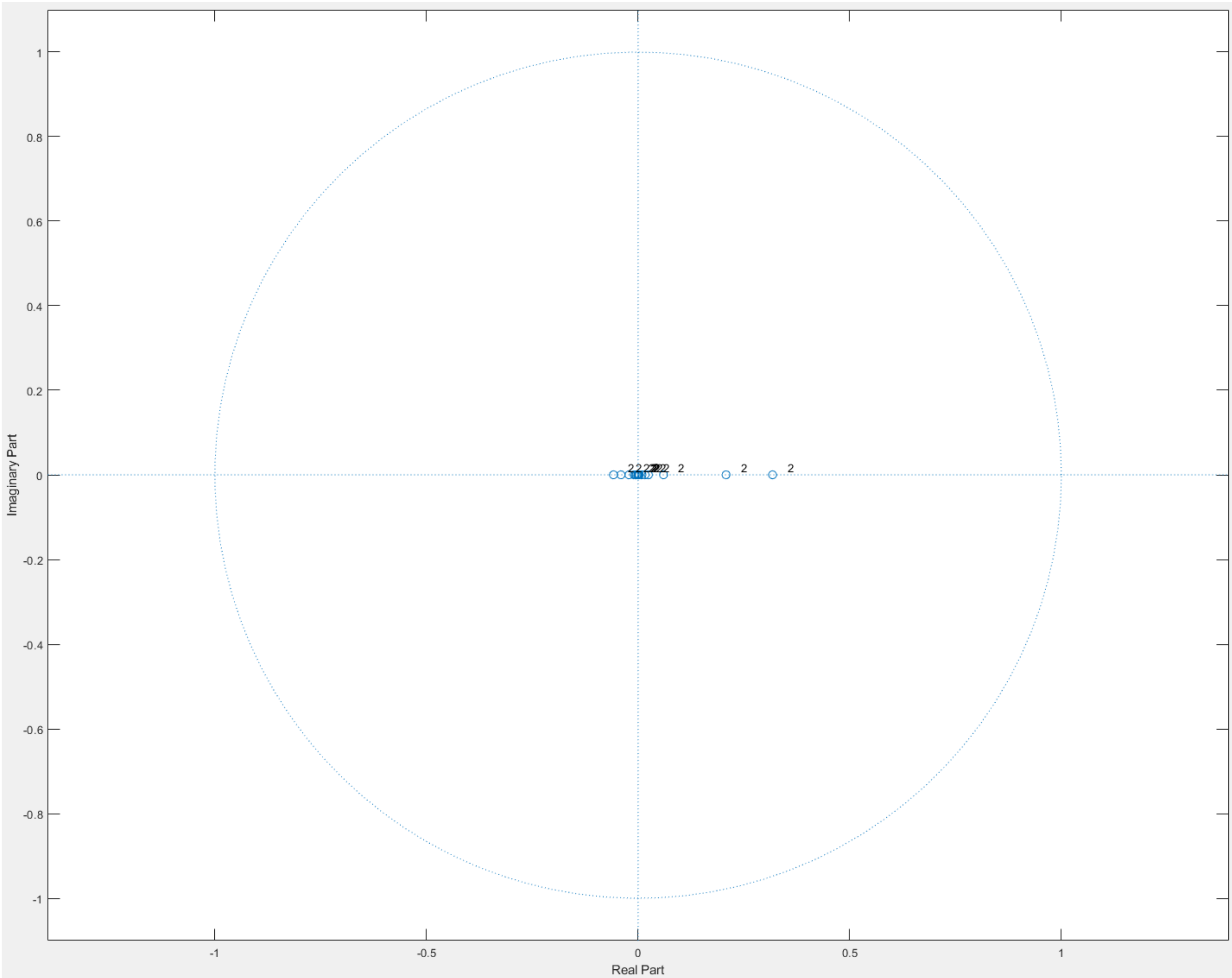
$M = 32.6 \rightarrow 33$ (type II)

$L = 34$

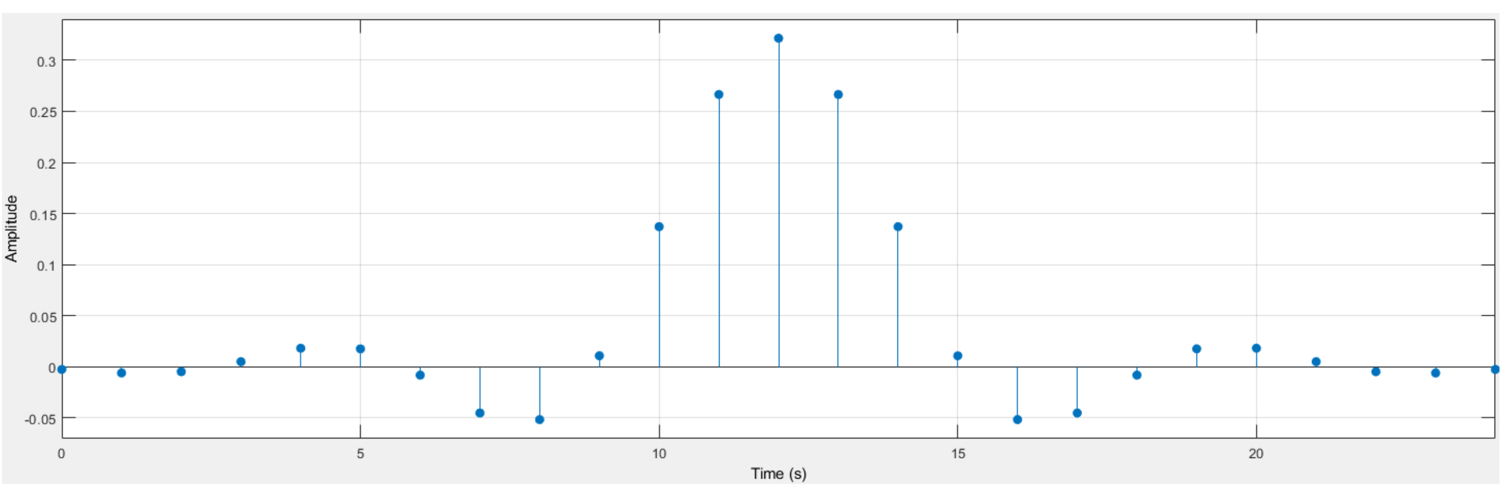

Non-Windowed
Windowed
Magnitude
Normalized Frequency [cyles/sample]

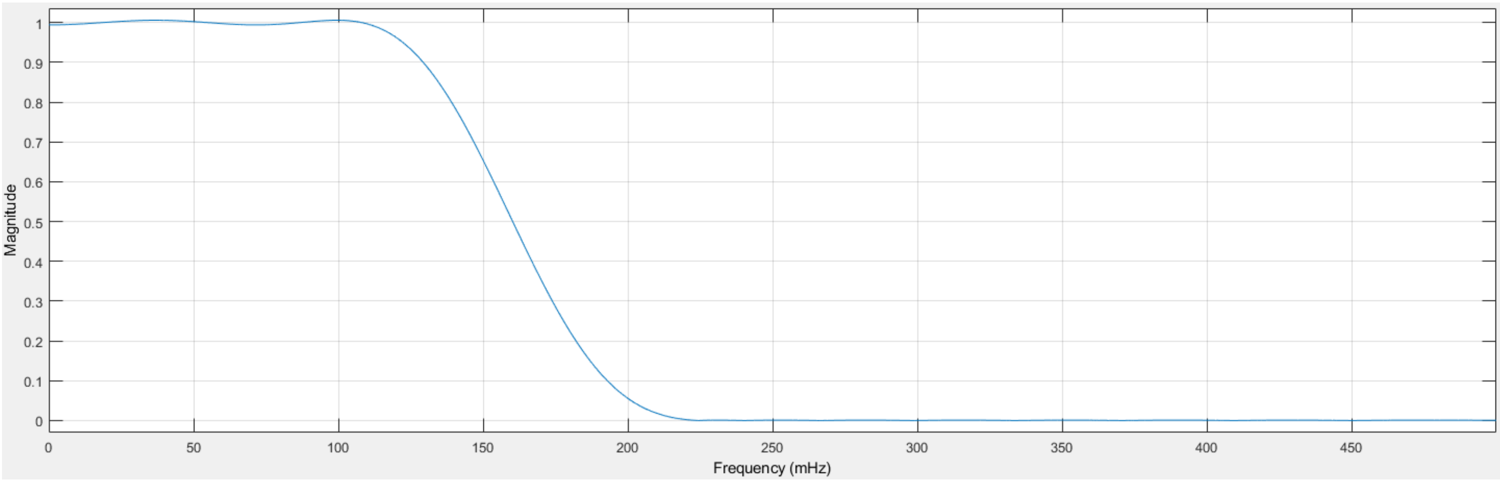

Non-Windowed
Windowed
Magnitude
Normalized Frequency [cyles/sample]

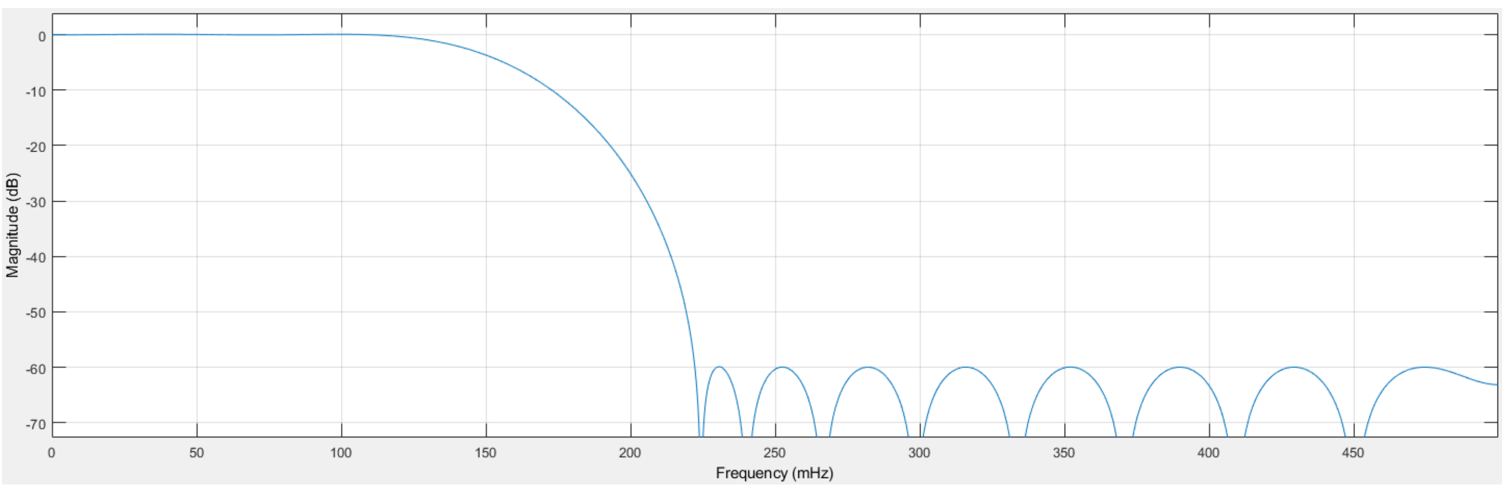

Non-Windowed
Windowed
Magnitude
Normalized Frequency [cyles/sample]

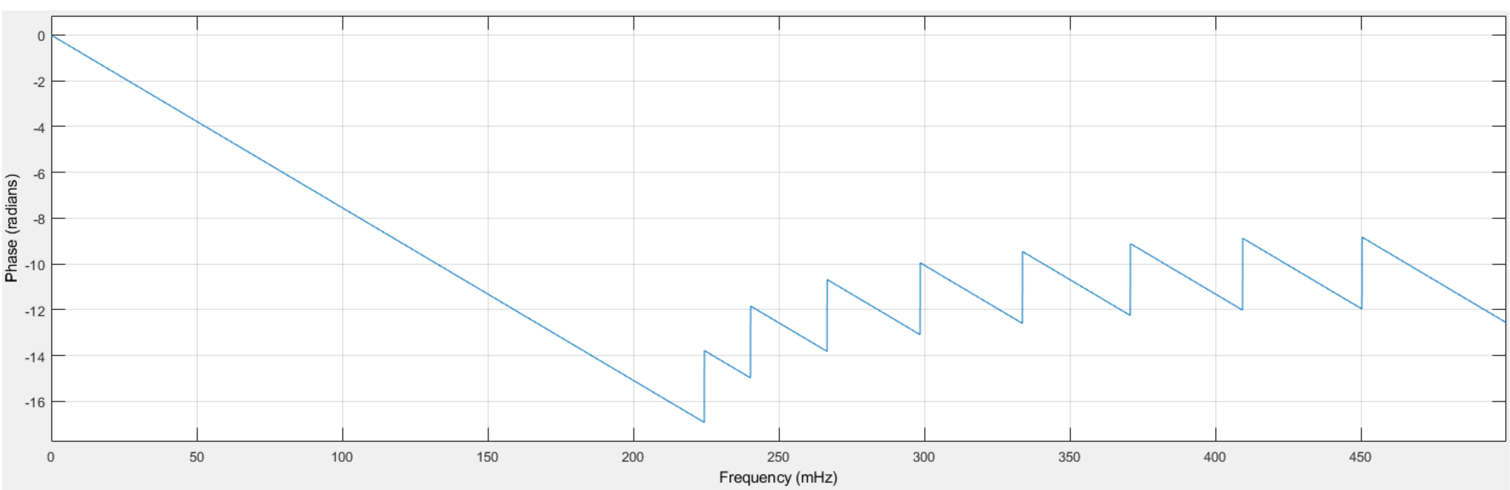


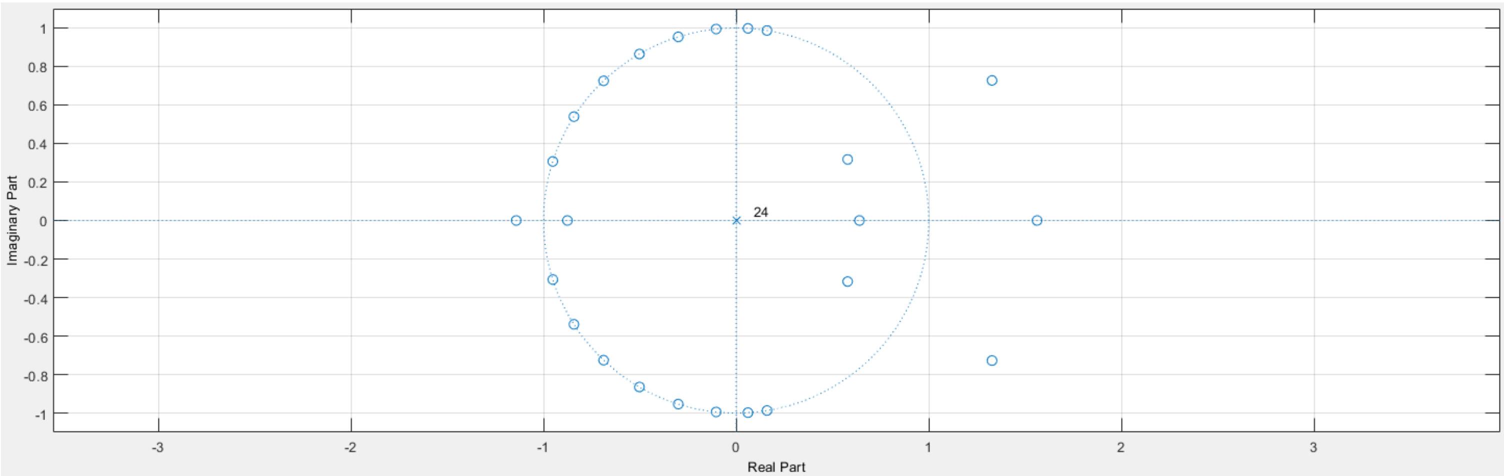
Equiripple:




Magnitude
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
50
100
150
200
250
300
350
400
450
Frequency (mHz)


Magnitude (dB)
0
-10
-20
-30
-40
-50
-60
-70
0
50
100
150
200
250
300
350
400
450
Frequency (mHz)


Phase (radians)
0
-2
-4
-6
-8
-10
-12
-14
-16
0
50
100
150
200
250
300
350
400
450
Frequency (mHz)



Filter	Length	Order	Group Delay	Type	δ or β	A
Hann	50	49	24.5	2	.001	60
Kaiser	34	33	16.5	2	5.653	60
Equiripple	25	24	12	1	.001	60

4. Suppose you would like to boost by a factor of 4 the low frequencies of an audio signal between the range of $0 \leq |F| \leq 150$ Hz and leave the scale of frequencies above 300 Hz unchanged. This type of filter has two passbands and no stopband. It is called a shelving filter. The transition band extends from 150 Hz to 300 Hz. You plan to perform this processing using the system depicted in Fig. 6.1 of the textbook (page 331) using a sample rate of $\frac{1}{T} = F_s = 20$k samples/second. Convert this information to a specification for a discrete-time filter and design a linear phase filter to meet this spec. Require your filter to limit deviations from the desired response to $\delta = 0.01$ in both pass bands.

- Design this filter using the window technique using the best window from the set {rectangular, Bartlett, Hann, Hamming, Blackman}. Indicate which window shape you used.
- Design this filter using the window technique using the Kaiser window. Indicate the Kaiser window parameters (β and M).
- Using FDATool in Matlab, design this filter using equiripple design method.

Make a table with one row for each filter you designed (window, Kaiser, equiripple). The table should have a column for the following parameters: length, order, group delay, type (I, II, III, or IV), and window parameters.

For each filter you design, turn in plots of the impulse response, magnitude response (linear and dB), and phase response. The frequency domain plots should show response over the interval $-\frac{1}{2} \leq f \leq \frac{1}{2}$. Also include a pole-zero plot.

$A = 40$ dB

$\delta = .01$

$f_1 = 150/20k = .0075$

$f_2 = 300/20k = .015$

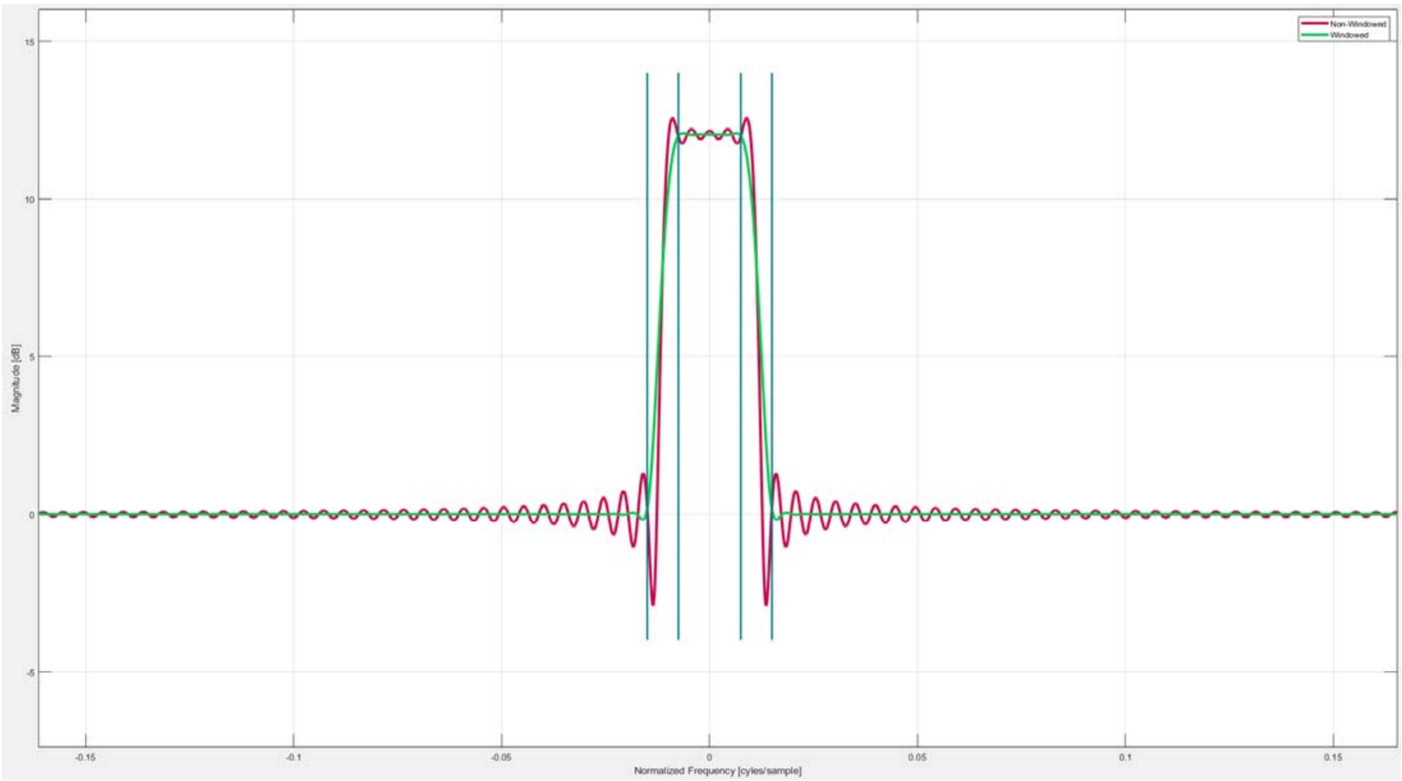
$\Delta f = .015$

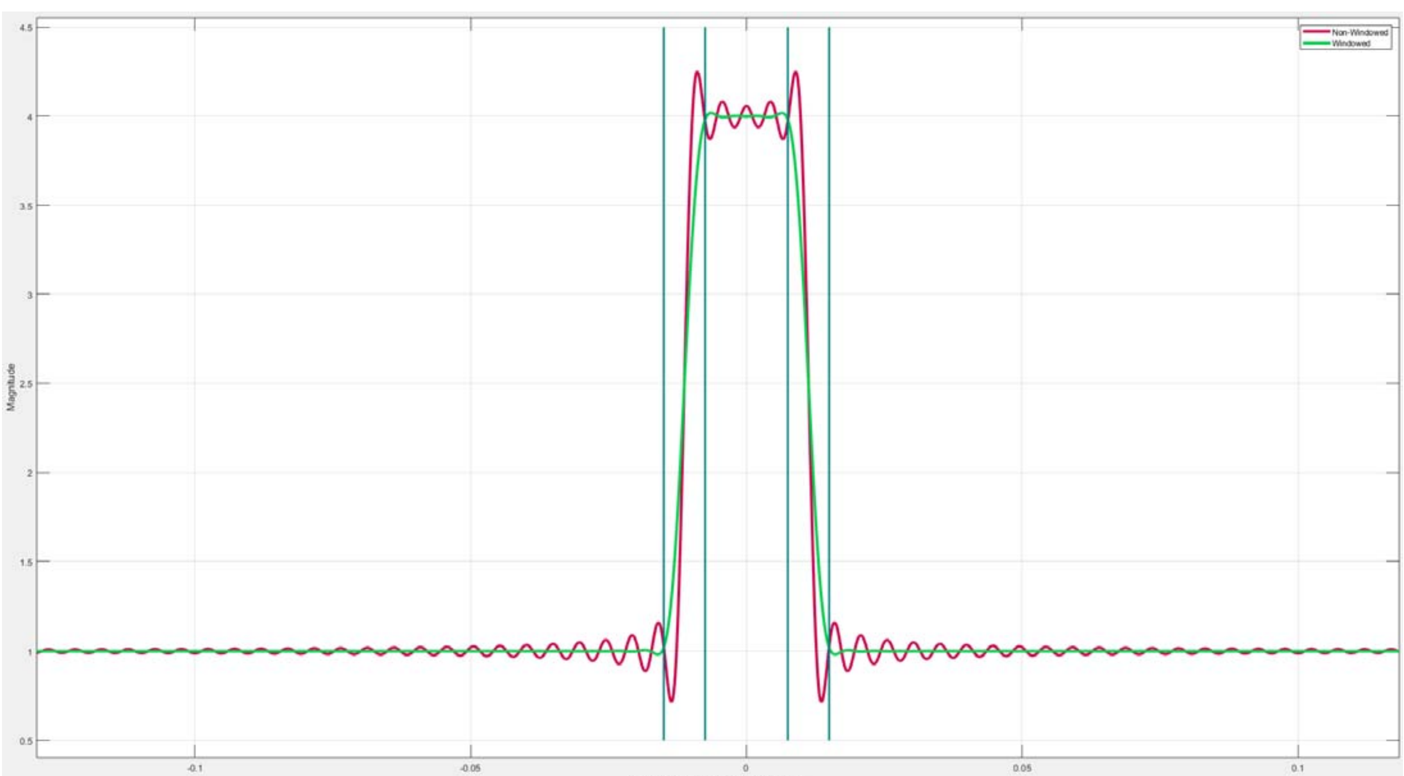
$\Delta\omega = 2\pi \cdot .015$

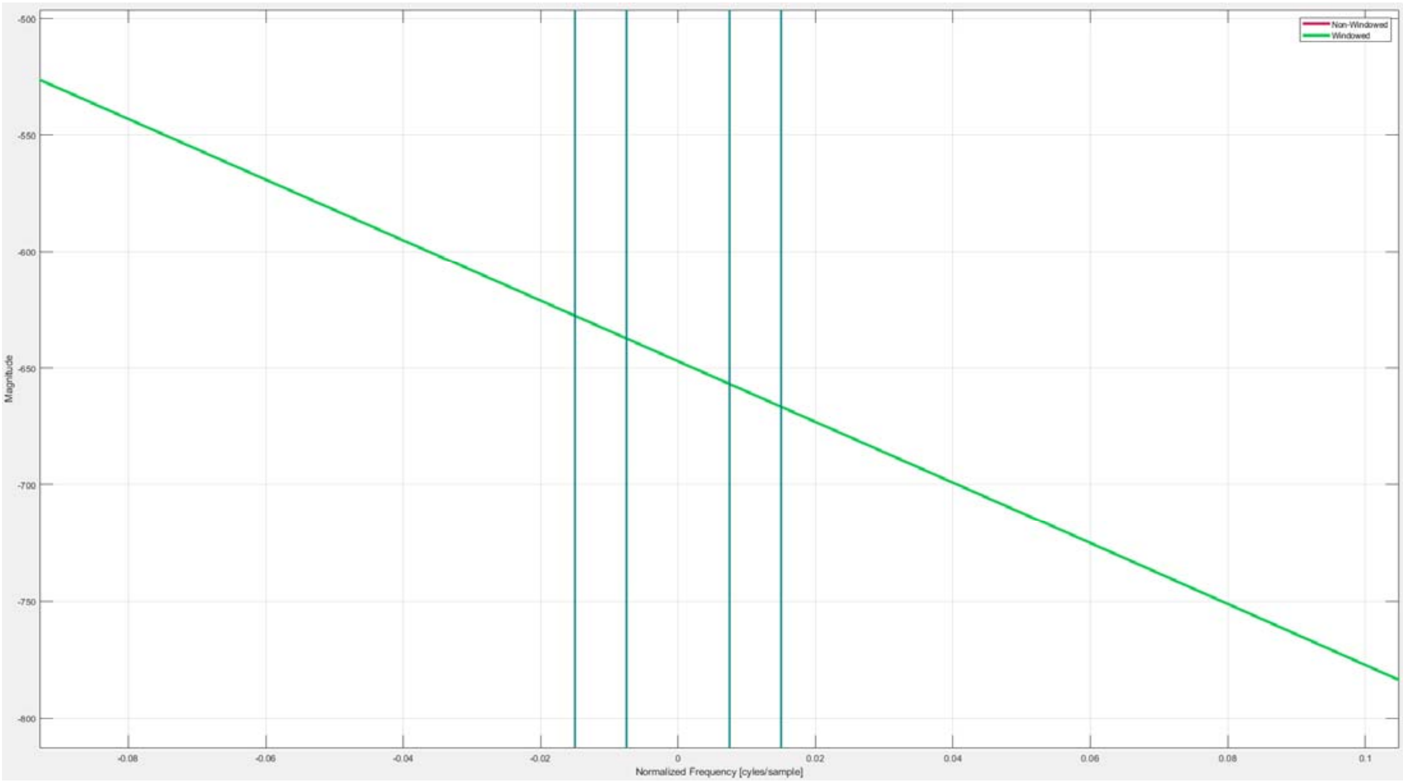
$L = \frac{6 \cdot 2\pi}{\Delta\omega} = 415$

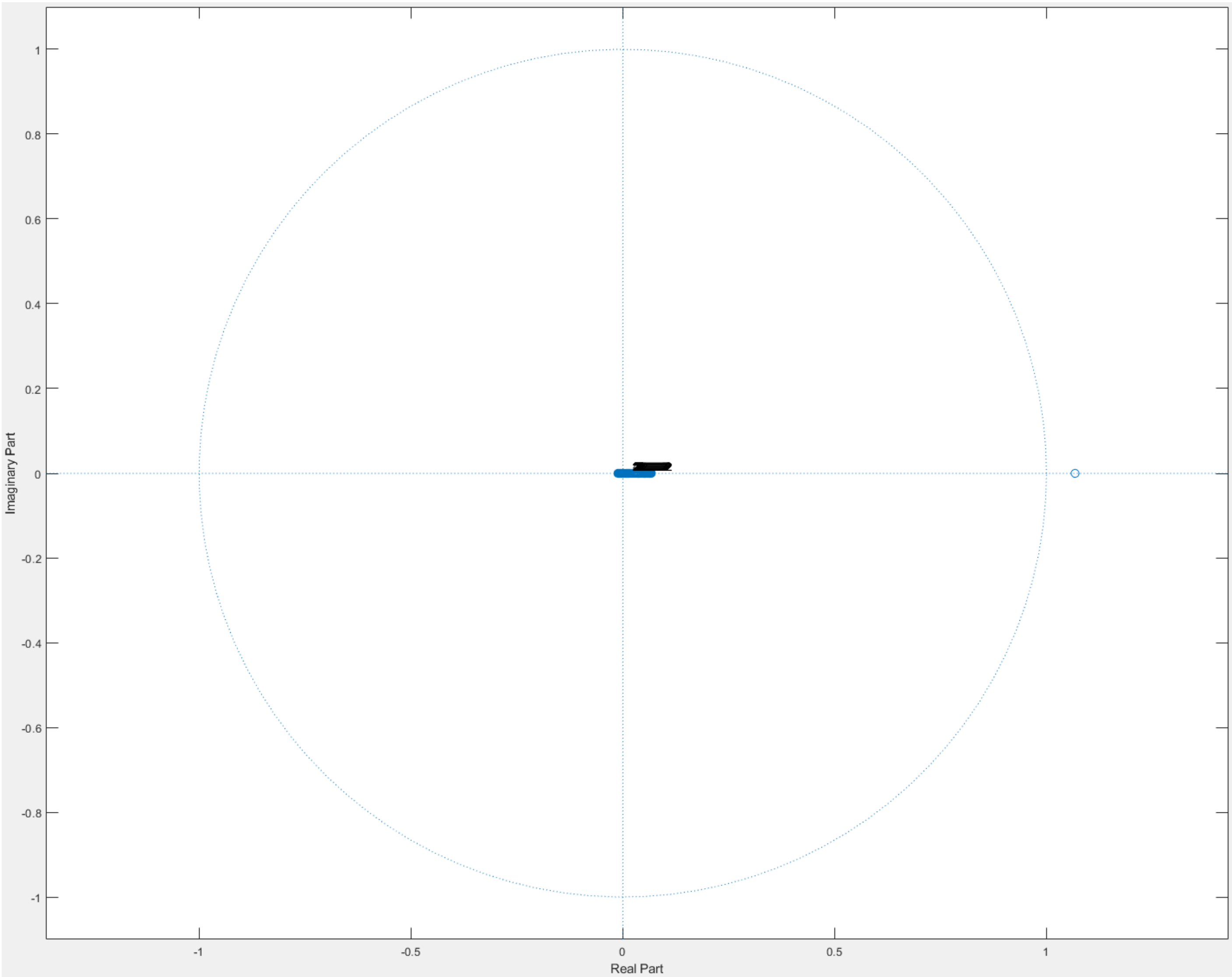
$M = 414$

$\alpha = 207$






Non-Windowed
Windowed
Magnitude
Normalized Frequency [cyles/sample]



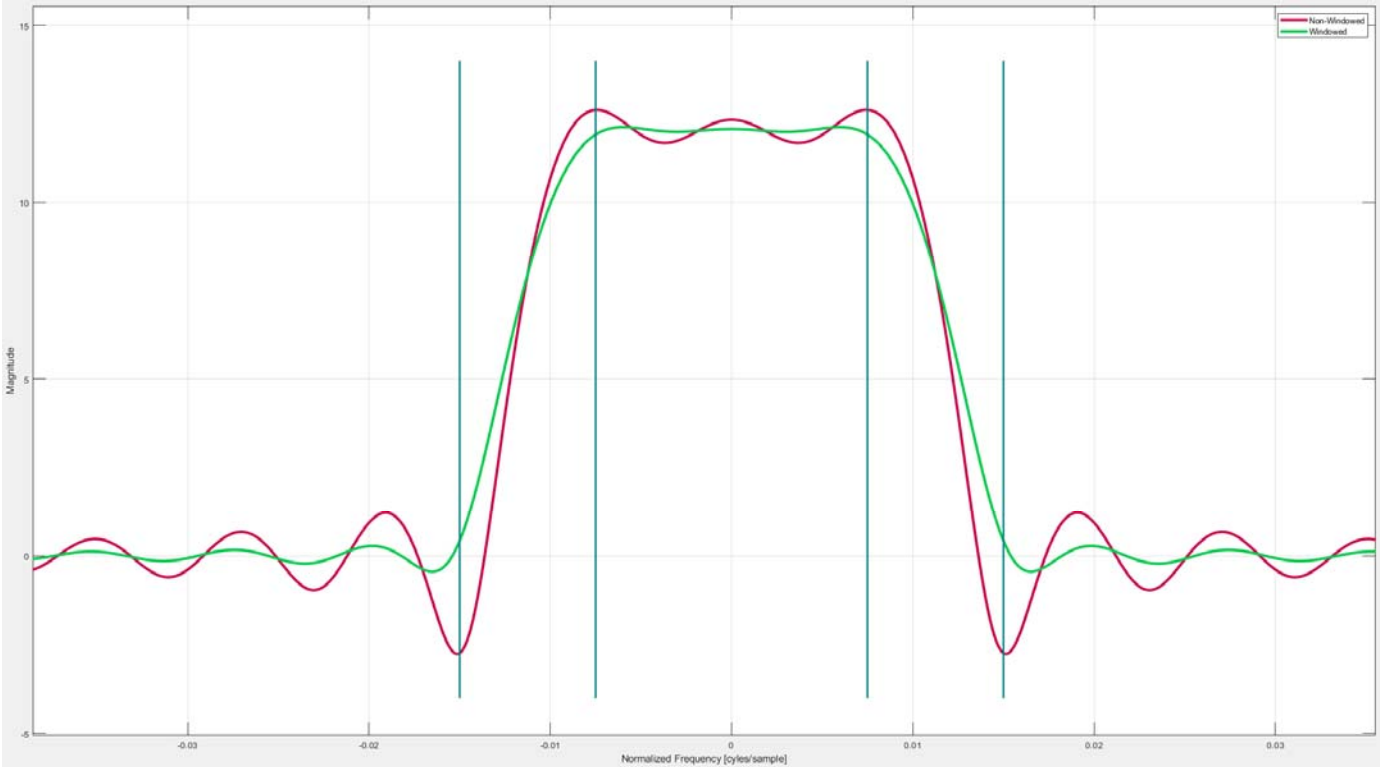
Kaiser:

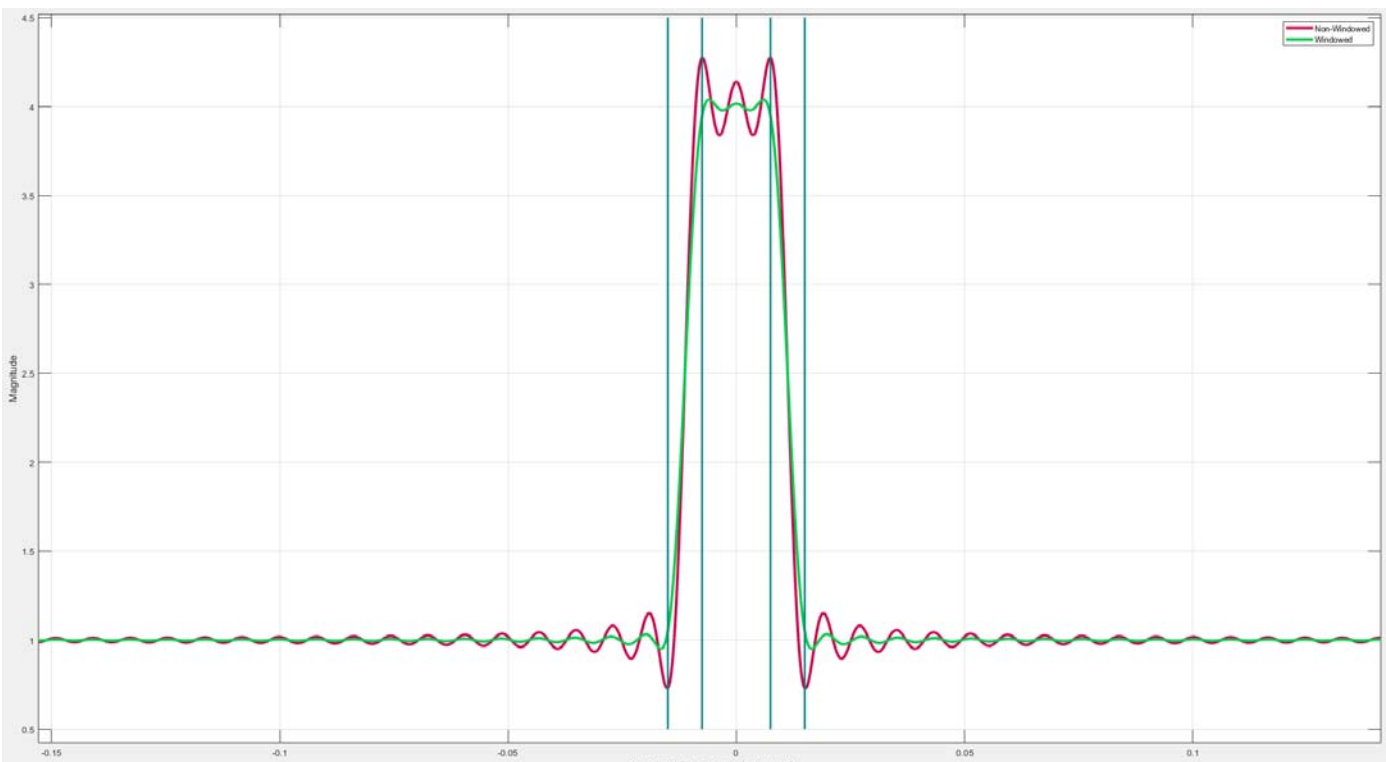
$\beta = 2.6635$

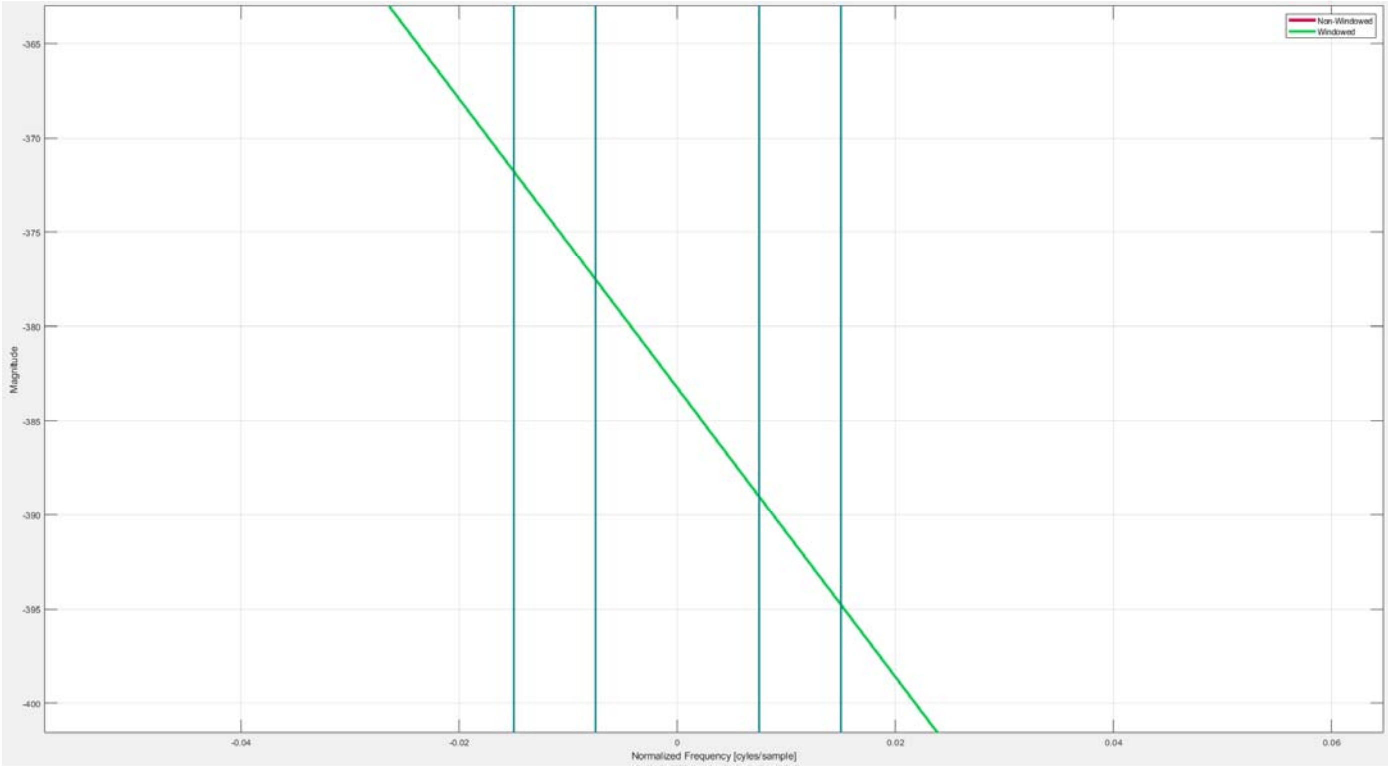
$L = 245$

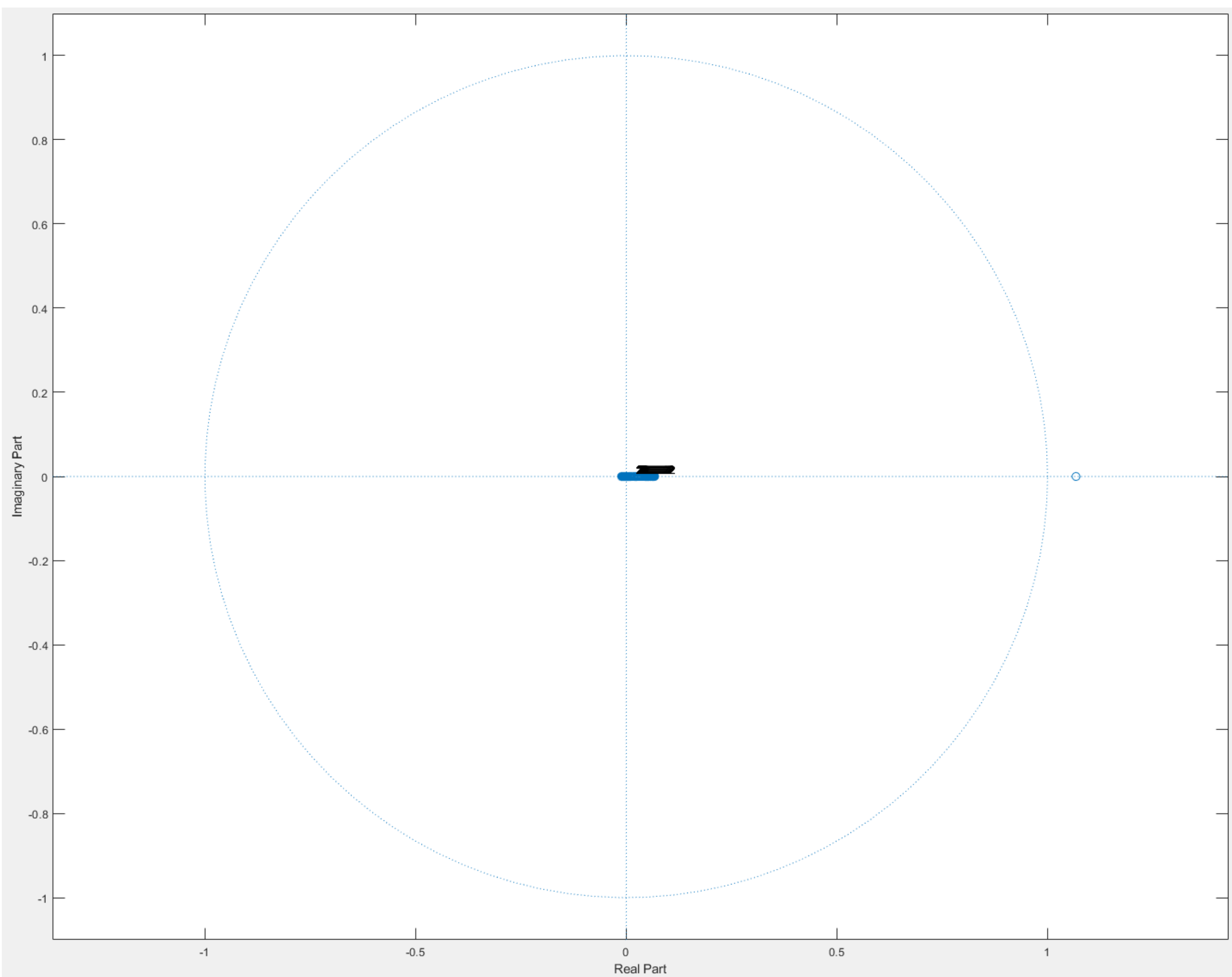
$M = 244$

$\alpha = 122$


Non-Windowed
Windowed
Magnitude
Normalized Frequency [cyles/sample]


Non-Windowed
Windowed
Magnitude
Normalized Frequency [cyles/sample]


Non-Windowed
Windowed
Magnitude
-365
-370
-375
-380
-385
-390
-395
-400
-0.04
-0.02
0
0.02
0.04
0.06
Normalized Frequency [cyles/sample]

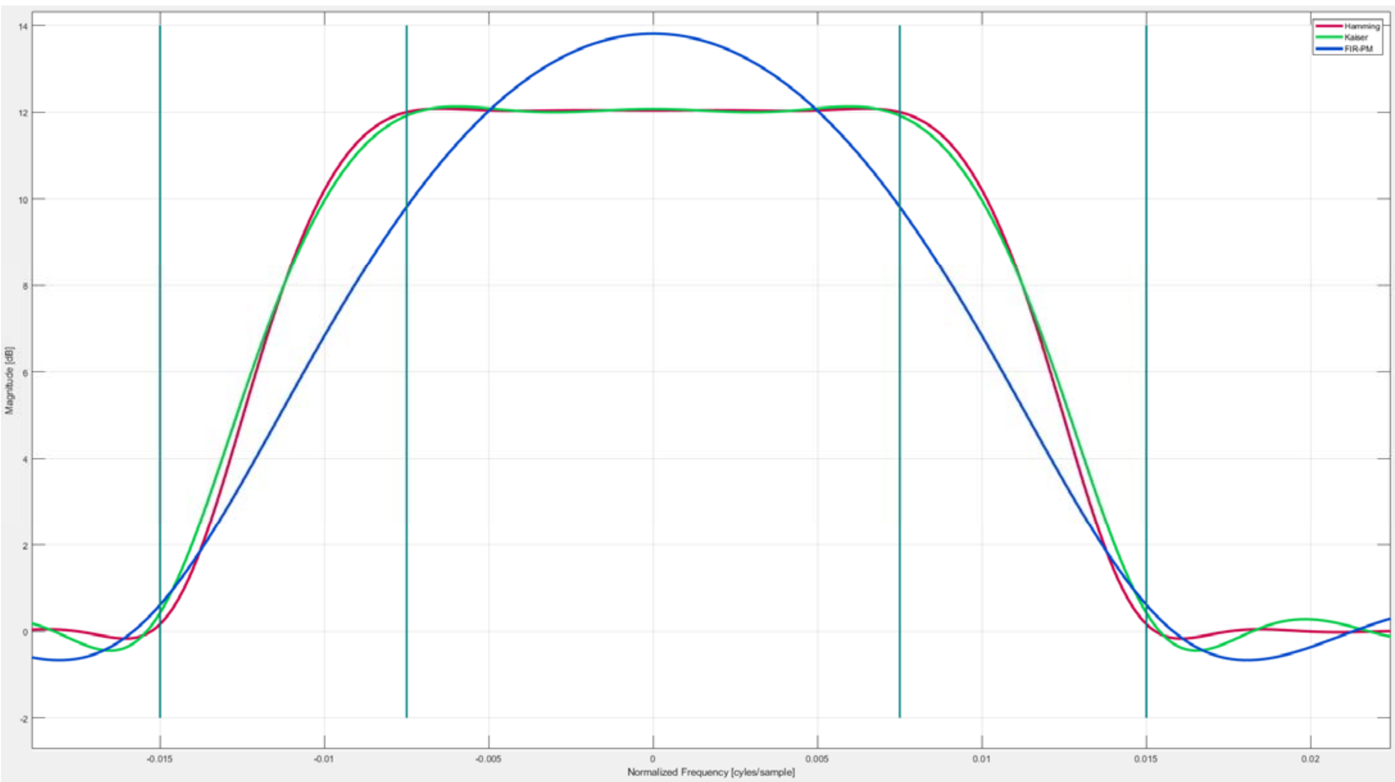

Imaginary Part
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1
Real Part

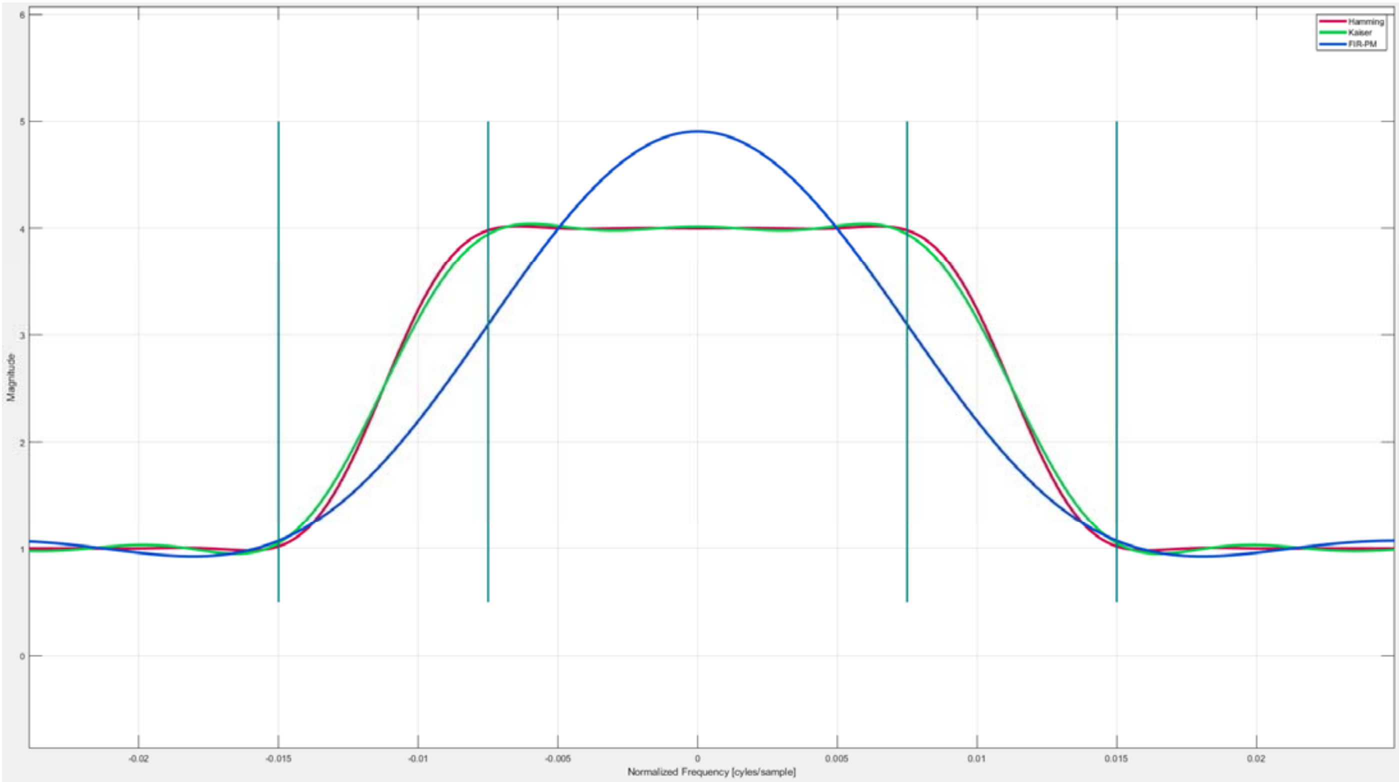
equiripple

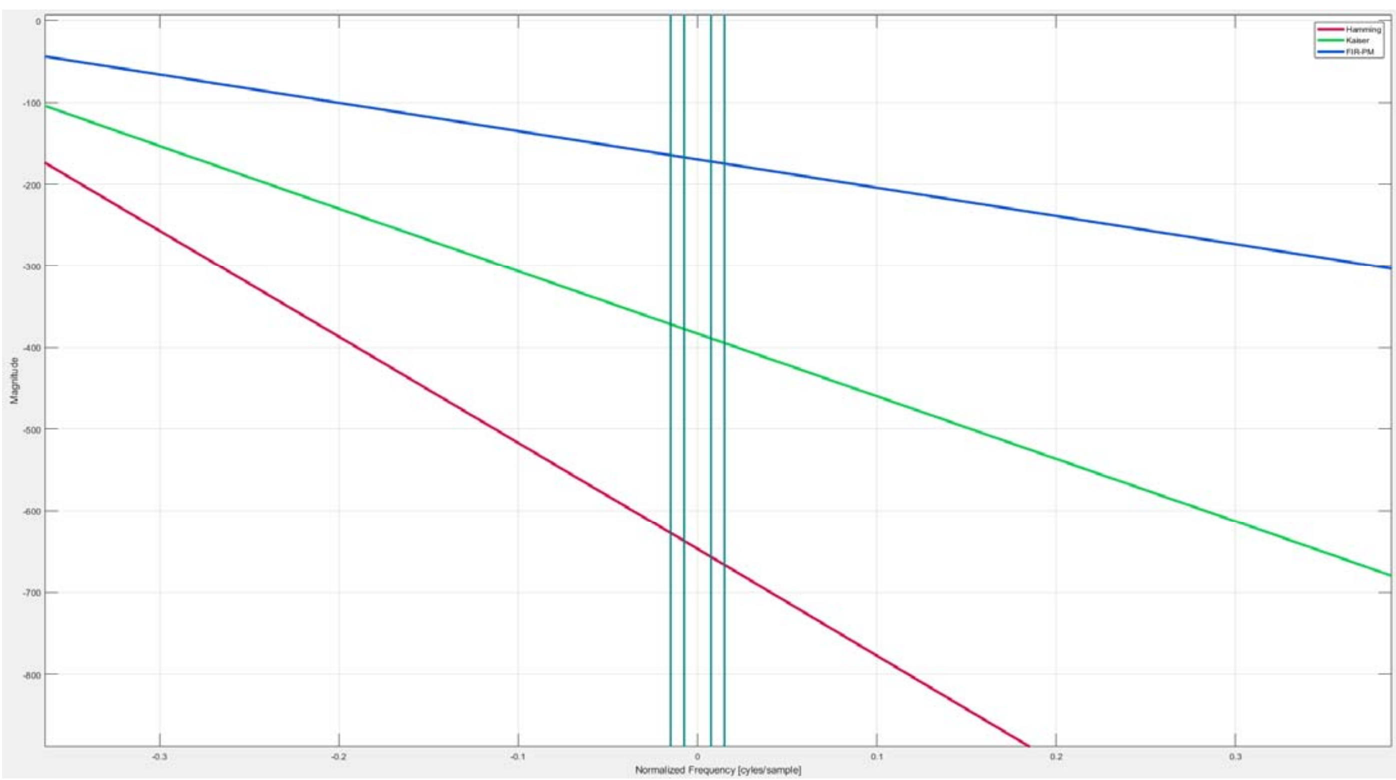
$L = 111$

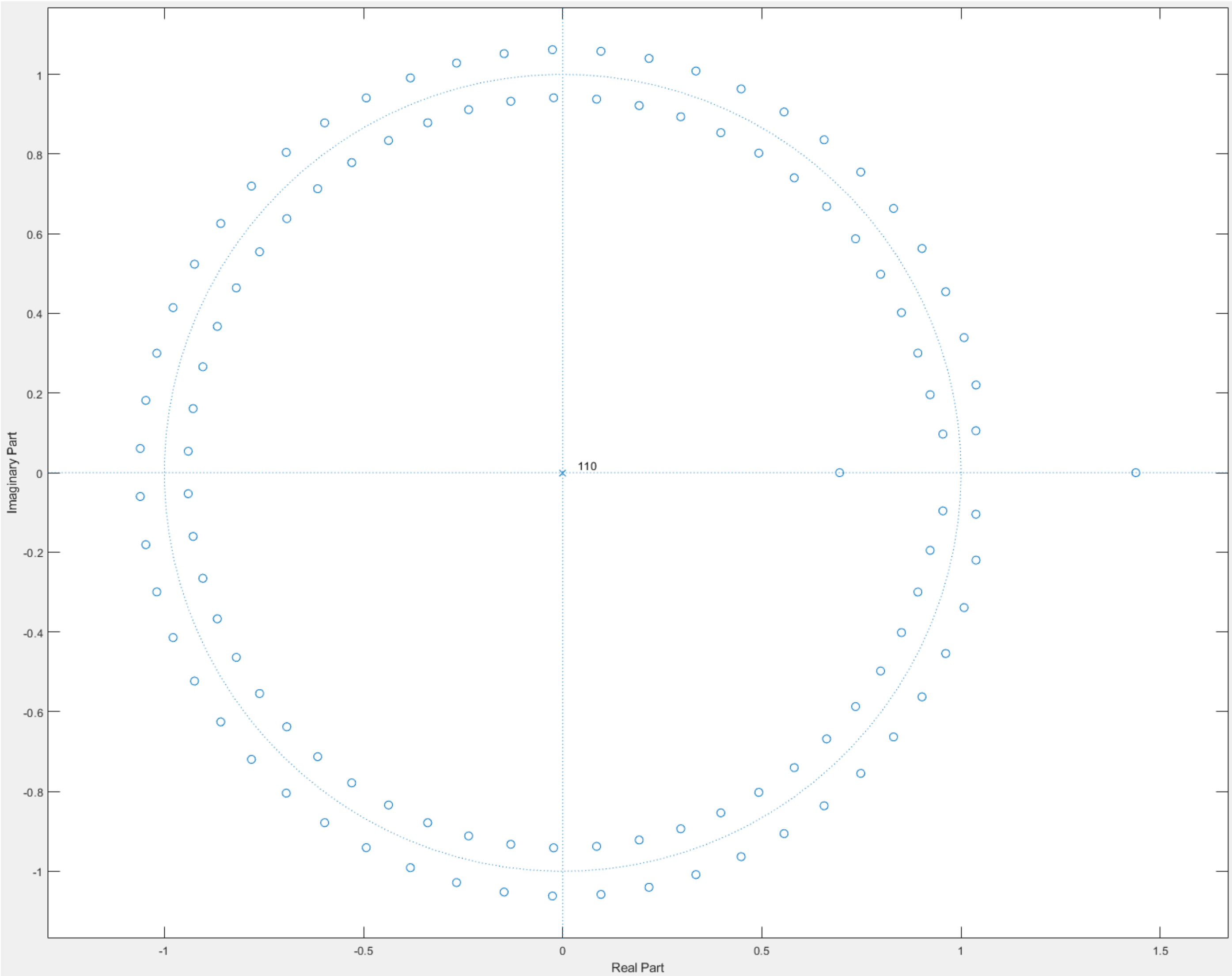
$M = 110$

$\alpha = 55$




Hamming
Kaiser
FIR-PM
Magnitude
Normalized Frequency [cyles/sample]


Hamming
Kaiser
FIR-PM
Magnitude
Normalized Frequency [cyles/sample]



Filter	Length	Order	Group Delay	Type	δ or β	A
Hann	415	414	207	1	.01	40
Kaiser	245	244	122	1	2.663	40
Equiripple	111	110	55	1	.01	40

5. After working the previous 4 problems, summarize your observations about the best filter design method to use if the goal is to minimize the filter length.

In all 4 problems, the equiripple method has had the shortest filter length. the filter itself may not always be as accurate as the other methods, but it has always had the shortest length.

6. Design a differentiator/delay filter pair to be used in FM demodulation. Let the received signal be $x[n]$. The FM demodulated signal is

$$y[n] = \text{imag}\left(\text{deriv}\{x[n]\} \cdot x^*[n]\right).$$

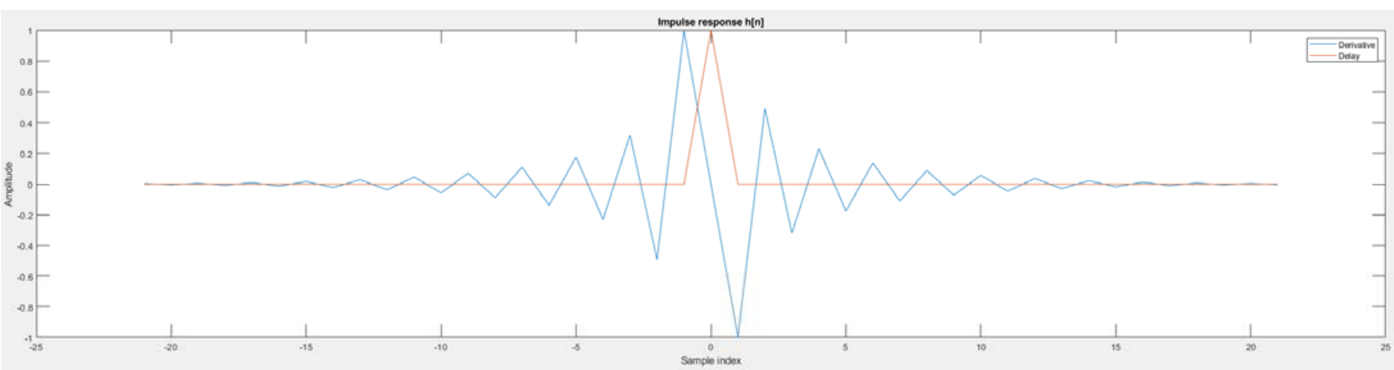
In FM demodulation (as in many other applications), it is vital to keep signal time aligned. Note that $\textbf{deriv}\{x[n]\}$ is the output of a derivative filter. Because the derivative filter delays the signal, the signal $x[n]$ must also be delayed. Thus it would be more correct to express FM demodulation as

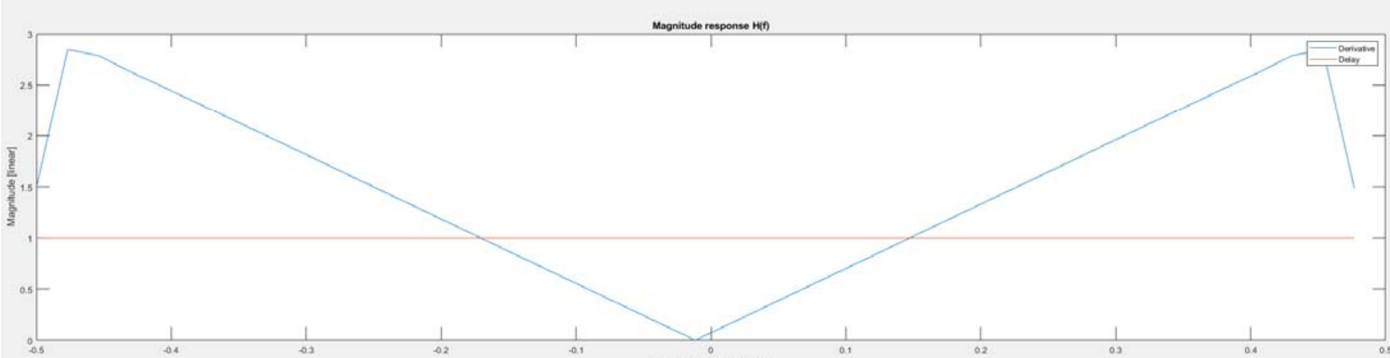
$$y[n] = \text{imag}\left(\text{deriv}\{x[n]\} \cdot \text{delay}\{x^*[n]\}\right),$$

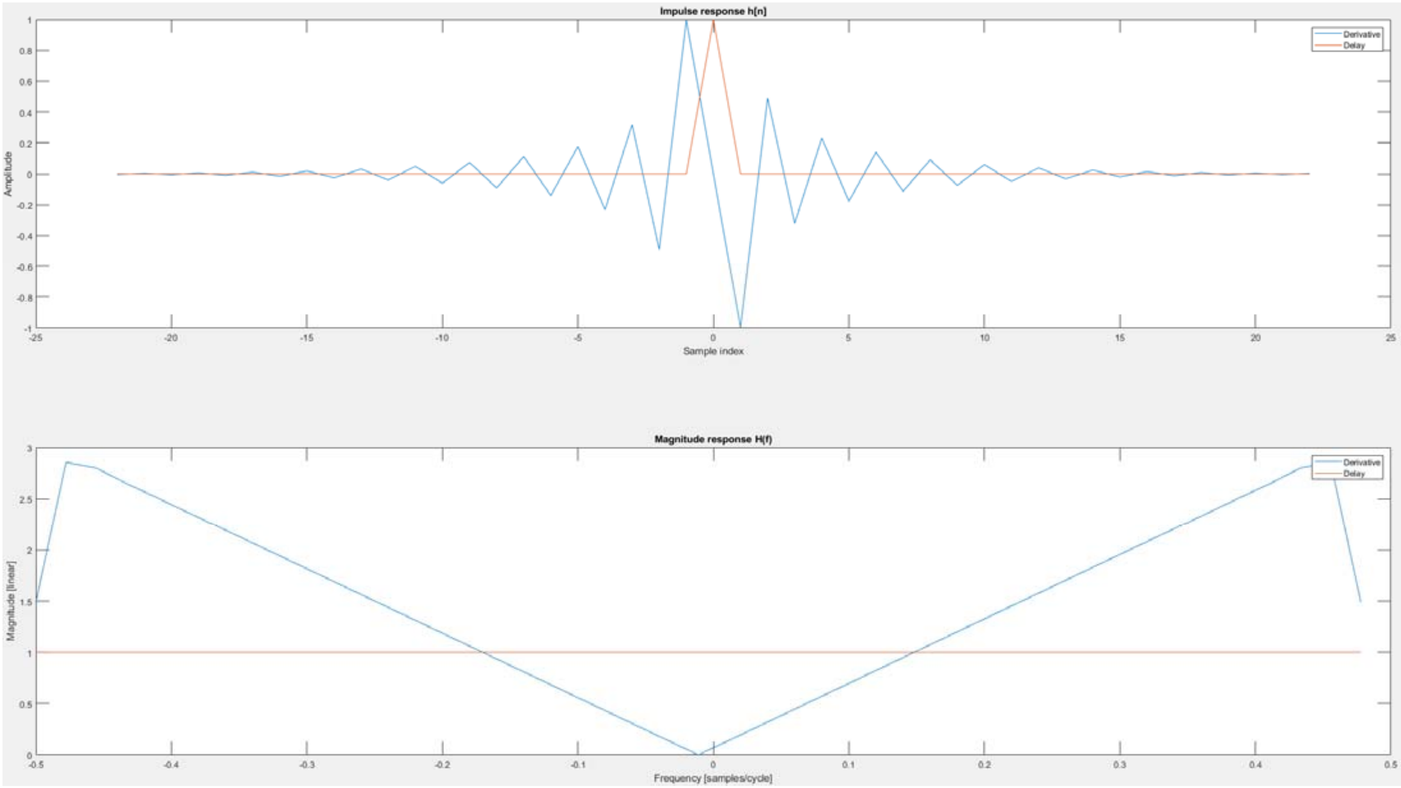
where $\textbf{delay}\{x[n]\}$ is the output of an all pass filter having the same delay as the derivative filter.

- Design type III derivative and delay filters using a Hamming window with length $L = 21$.
- Design type IV derivative and delay filters using a Hamming window with length $L = 22$.

In each case, plot the impulse response, and frequency response over $-\frac{1}{2} \leq f \leq \frac{1}{2}$.






Impulse response h[n]
Derivative
Delay
Amplitude
Sample index
Magnitude response H(f)
Derivative
Delay
Magnitude [linear]
Frequency [samples/cycle]

Code:

```
NFFT = 2^14;
freq = [0:NFFT-1]/NFFT - 0.5;

fp = 1/9;
fs = 2/9;
Ap = 0.1; % dB
As = 60; % dB

Bp = 10^(Ap/20);
dp = (Bp-1)/(Bp+1); % pass band ripple
Bs = (1+dp)*10^(-As/20);
ds = Bs; % stop band ripple (approx.)

%% Windowed filter design
Fw = fs - fp;
fc = 0.5*(fs + fp);
[d,dind] = min([dp ds]);
if(dind == 1)
    A = Ap; % design is dominated by pass band ripple
else
    A = As; % design is dominated by stop band ripple
end;
% Blackman window cuz 74 dB
L = ceil(11*pi/(2*pi*Fw));
% Now we know the shape (Blackman) and the length.
win = blackman(L);
% Generate the impulse response
M = L-1; % filter order
alpha = M/2; % group delay
n = [0:L-1].' - alpha;
hd = (2*fc)*sinc(2*fc*n);
h = hd.*win;
hblackman = h;

%magnitude dB
plot(freq,20*log10(abs(fftshift(fft(hd,NFFT)))),'Color',[0.8 0 0.3],'LineWidth',3);
hold on;
plot(freq,20*log10(abs(fftshift(fft(h ,NFFT)))),'Color',[0.0 0.8 0.3],'LineWidth',3);
ax = axis;
plot((fp)*[1 1],ax(3:4),'Color',[0 0.5 0.5],'LineWidth',2);
plot(-(fp)*[1 1],ax(3:4),'Color',[0 0.5 0.5],'LineWidth',2);
plot((fs)*[1 1],ax(3:4),'Color',[0 0.5 0.5],'LineWidth',2);
plot(-(fs)*[1 1],ax(3:4),'Color',[0 0.5 0.5],'LineWidth',2);
hold off;
grid on;
xlabel('Normalized Frequency [cyles/sample]');
ylabel('Magnitude [dB]');
legend('Non-Windowed','Windowed');
ylim([-100 10]);
%magnitude linear
plot(freq,(abs(fftshift(fft(hd,NFFT)))),'Color',[0.8 0 0.3],'LineWidth',3);
hold on;
plot(freq,(abs(fftshift(fft(h ,NFFT)))),'Color',[0.0 0.8 0.3],'LineWidth',3);
ax = axis;
plot((fp)*[1 1],ax(3:4),'Color',[0 0.5 0.5],'LineWidth',2);
plot(-(fp)*[1 1],ax(3:4),'Color',[0 0.5 0.5],'LineWidth',2);
plot((fs)*[1 1],ax(3:4),'Color',[0 0.5 0.5],'LineWidth',2);
plot(-(fs)*[1 1],ax(3:4),'Color',[0 0.5 0.5],'LineWidth',2);
hold off;
grid on;
xlabel('Normalized Frequency [cyles/sample]');
ylabel('Magnitude');
legend('Non-Windowed','Windowed');
ylim([-.5 1.5]);


%phase
plot(freq,unwrap(angle(fftshift(fft(hd,NFFT)))),'Color',[0.8 0 0.3],'LineWidth',3);
hold on;
plot(freq,unwrap(angle(fftshift(fft(h ,NFFT)))),'Color',[0.0 0.8 0.3],'LineWidth',3);
ax = axis;
plot((fp)*[1 1],ax(3:4),'Color',[0 0.5 0.5],'LineWidth',2);
plot(-(fp)*[1 1],ax(3:4),'Color',[0 0.5 0.5],'LineWidth',2);
plot((fs)*[1 1],ax(3:4),'Color',[0 0.5 0.5],'LineWidth',2);
plot(-(fs)*[1 1],ax(3:4),'Color',[0 0.5 0.5],'LineWidth',2);
hold off;
grid on;
xlabel('Normalized Frequency [cyles/sample]');
ylabel('Magnitude');
legend('Non-Windowed','Windowed');
ylim([-100 10]);
```

```matlab
%% Kaiser design
[M,Wn,beta,filtype]=kaiserord([fp fs],[1 0],[dp ds],1)
L = M+1;
alpha = M/2;
win = kaiser(L,beta);
n = [0:L-1].' - alpha;
hd = (2*fc)*sinc((2*fc)*(n));
h = hd.*win;
hkaiser = h;

plot(freq,20*log10(abs(fftshift(fft(hd,NFFT)))),'Color',[0.8 0 0.3],'LineWidth',3);
hold on;
plot(freq,20*log10(abs(fftshift(fft(h ,NFFT)))),'Color',[0.0 0.8 0.3],'LineWidth',3);
ax = axis;
plot((fp)*[1 1],ax(3:4),'Color',[0 0.5 0.5],'LineWidth',2);
plot(-(fp)*[1 1],ax(3:4),'Color',[0 0.5 0.5],'LineWidth',2);
plot((fs)*[1 1],ax(3:4),'Color',[0 0.5 0.5],'LineWidth',2);
plot(-(fs)*[1 1],ax(3:4),'Color',[0 0.5 0.5],'LineWidth',2);
hold off;
grid on;
xlabel('Normalized Frequency [cyles/sample]');
ylabel('Magnitude');
legend('Non-Windowed','Windowed');
ylim([-100 10]);

plot(freq,(abs(fftshift(fft(hd,NFFT)))),'Color',[0.8 0 0.3],'LineWidth',3);
hold on;
plot(freq,(abs(fftshift(fft(h ,NFFT)))),'Color',[0.0 0.8 0.3],'LineWidth',3);
ax = axis;
plot((fp)*[1 1],ax(3:4),'Color',[0 0.5 0.5],'LineWidth',2);
plot(-(fp)*[1 1],ax(3:4),'Color',[0 0.5 0.5],'LineWidth',2);
plot((fs)*[1 1],ax(3:4),'Color',[0 0.5 0.5],'LineWidth',2);
plot(-(fs)*[1 1],ax(3:4),'Color',[0 0.5 0.5],'LineWidth',2);
hold off;
grid on;
xlabel('Normalized Frequency [cyles/sample]');
ylabel('Magnitude');
legend('Non-Windowed','Windowed');
ylim([-.25 1.2]);

plot(freq,unwrap(angle(fftshift(fft(hd,NFFT)))),'Color',[0.8 0 0.3],'LineWidth',3);
hold on;
plot(freq,unwrap(angle(fftshift(fft(h ,NFFT)))),'Color',[0.0 0.8 0.3],'LineWidth',3);
ax = axis;
plot((fp)*[1 1],ax(3:4),'Color',[0 0.5 0.5],'LineWidth',2);
plot(-(fp)*[1 1],ax(3:4),'Color',[0 0.5 0.5],'LineWidth',2);
plot((fs)*[1 1],ax(3:4),'Color',[0 0.5 0.5],'LineWidth',2);
plot(-(fs)*[1 1],ax(3:4),'Color',[0 0.5 0.5],'LineWidth',2);
hold off;
grid on;
xlabel('Normalized Frequency [cyles/sample]');
ylabel('Magnitude');
legend('Non-Windowed','Windowed');
ylim([-100 10]);
```