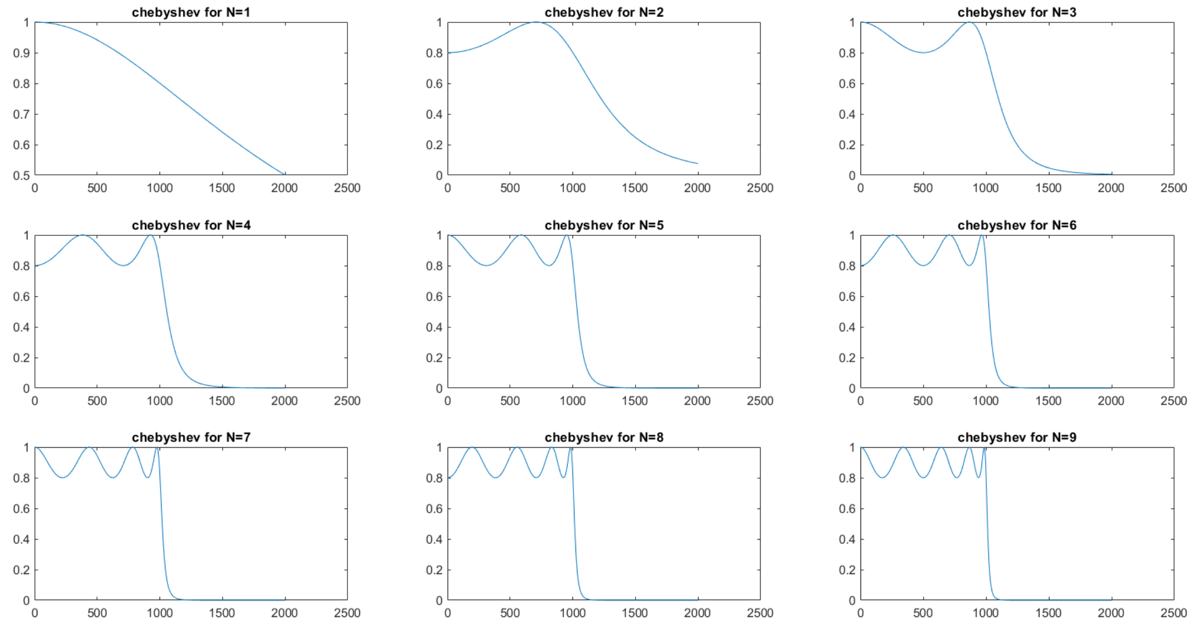
Question 1:   Mohammed Khaleel(21ee01027)

Q.1. a) Implement the equation of Analog Chebyshev Low Pass Filter Approximation and observe the effect of order N on the magnitude response.

Chebyshev filter for different N

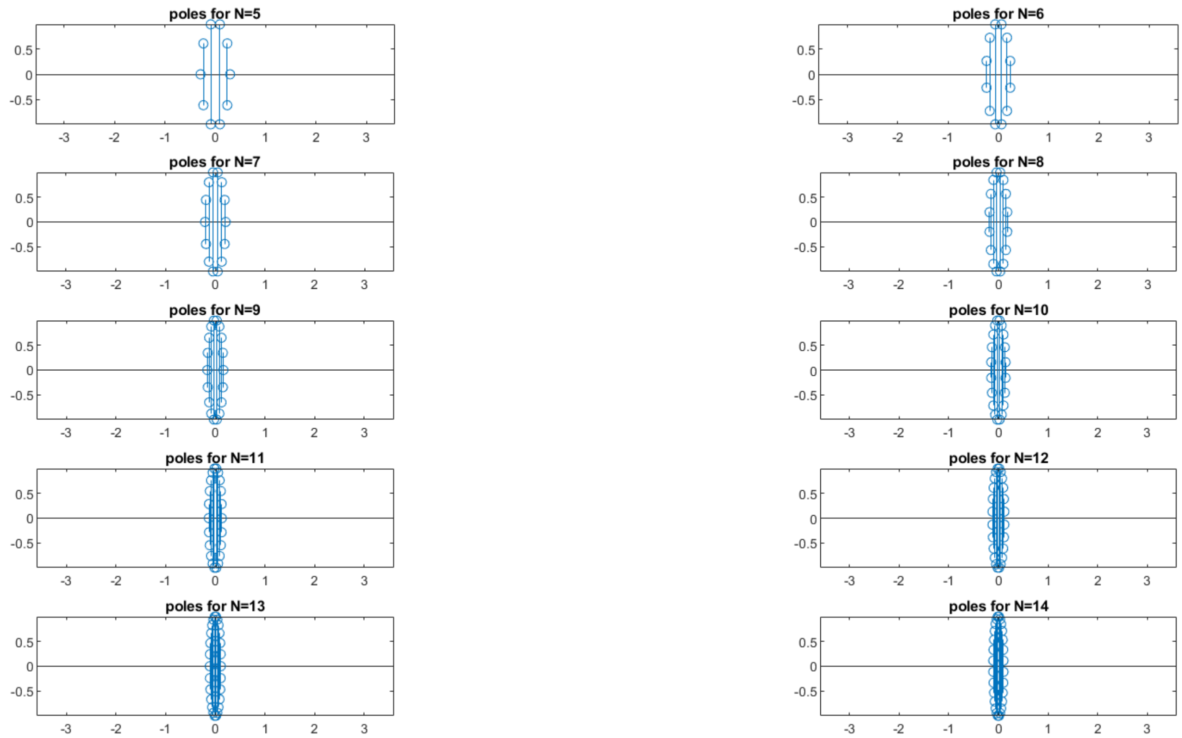
```
for N=1:9
n = 0:0.001:2;
Cn = ones(1,length(n));
e = 0.5;

for i = 1:length(n)
    if n(i)<=1
        Cn(i) = cos(N*acos(n(i)));
    else
        Cn(i) = cosh(N*acosh(n(i)));
    end
end
H = 1./(1 + (e.^2).*(Cn.^2));
subplot(3,3,N)
plot(H)
title(['chebyshev for N=',num2str(N)])
end
```



Poles of chebishev:

b) Plot the poles of the filter for different values of order N and observe the shape of its contour.

```
clf
for i = 5:14
syms k
N1 = i;

Ak = 1/N1*(2*k+1)*pi/2;
Bk = 1/N1*asinh(1/e);

sk = sin(Ak)*sinh(Bk) + 1j*cos(Ak)*cosh(Bk);

k_subs = 0:2*N1-1;
sk_subs = subs(sk, k, k_subs);

subplot(5,2,i-4)
stem(real(sk_subs),imag(sk_subs))
xlim([-1,1])
ylim([-1,1])
title(['poles for N=',num2str(N1)])
end
```



Question 2: Chebyshev LPF

Q.2. a) Design and realize the IIR Chebyshev LP filter with the specifications as discussed in the tutorial.

Given Parameters

```
Sampling_Frequency = 2000;
Pass_Band_frequency  = 100;
Stop_band_frequency = 500;
Pass_Band_attenuation = -1;
Stop_Band_attenuation = -40;
```

getting dicrete signal parameters

```
dp = 10^(Pass_Band_attenuation/20);
ds = 10^(Stop_Band_attenuation/20);
wp = 2*pi*Pass_Band_frequency/Sampling_Frequency;
ws = 2*pi*Stop_band_frequency/Sampling_Frequency;
Wp = tan(wp/2);
Ws = tan(ws/2);
```

N,e,Ak,B

```
N = ceil(acosh(((1/(ds^2)-1)/(1/(dp^2)-1))^0.5)/acosh(Ws/Wp))
```

```
N = 3
```

```
e = (1/dp^2-1)^0.5
```

```
e = 0.5088
```

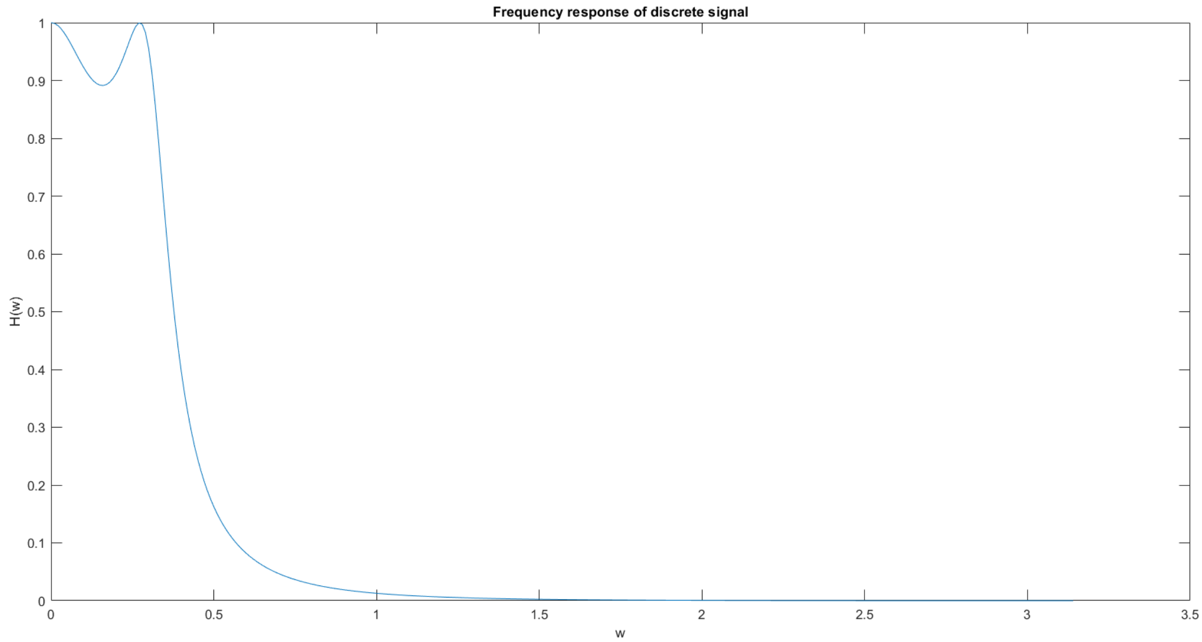
```
k = 0:2*N-1;
Ak = (2*k+1)*pi/(2*N);
B = 1/N*asinh(1/e);
```

using cheb1ord

```
[N,wn] =
cheb1ord(2*Pass_Band_frequency/Sampling_Frequency,2*Stop_band_frequency/Sampli
ng_Frequency,-Pass_Band_attenuation,-Stop_Band_attenuation);
[a,b] = cheby1(3,-Pass_Band_attenuation,wn);
```

plotting its response

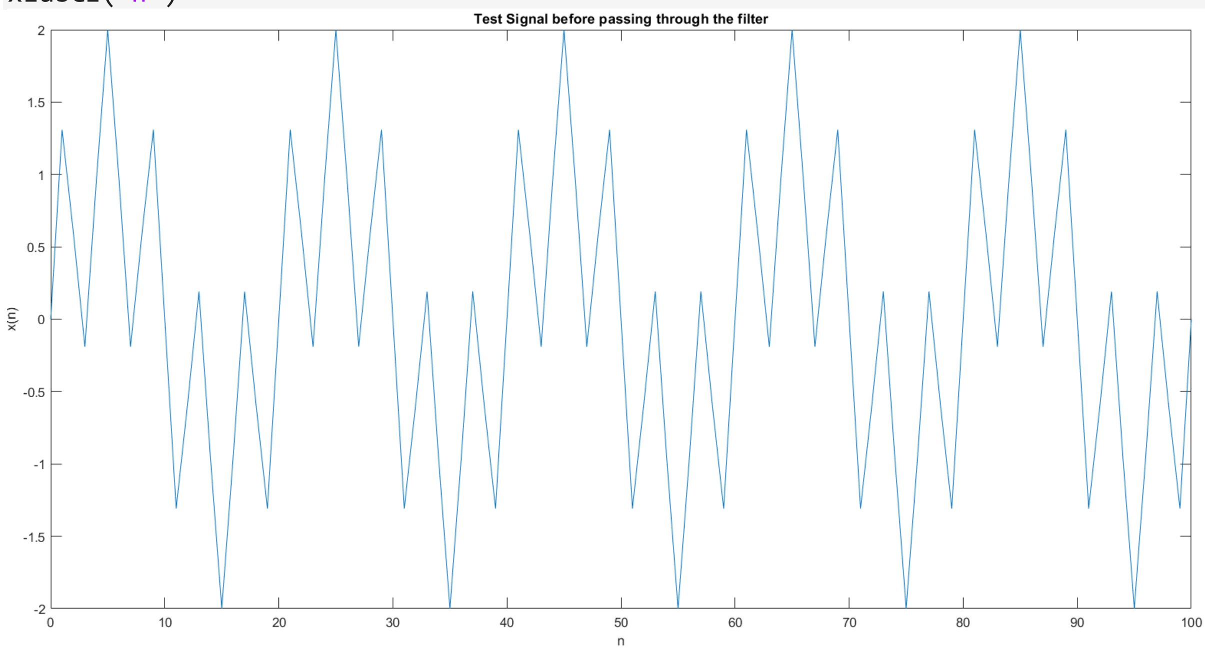
```
w = 0:0.01:pi;
Hw = freqz(a,b,w);
plot(w,abs(Hw))
title('Frequency response of discrete signal')
ylabel('H(w)')
xlabel('w')
```



b) Generate two sinusoids one within passband and other out of passband, add them and pass through the filter as designed in part a). Plot the input and output signals and verify whether the desired specifications are satisfied or not.

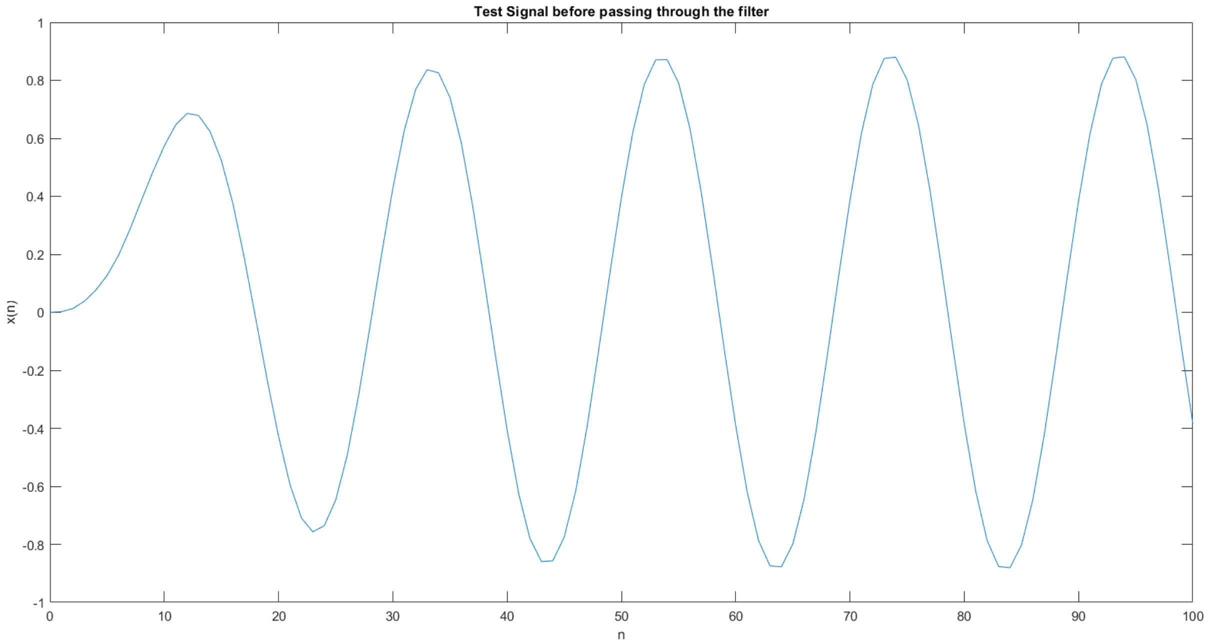
Checking using sine signals

```
n = 0:100;
test_signal = sin(pi/10*n) + sin(pi/2*n);
plot(n,test_signal)
title('Test Signal before passing through the filter')
ylabel('x(n)')
xlabel('n')
```



Passing through filter

```
output = filter(a,b,test_signal);
plot(n,output)
title('Test Signal before passing through the filter')
ylabel('x(n)')
xlabel('n')
```



Question 3:

Q.3. a) Decrease the passband and/or stopband tolerance level (thus making the filter closer to ideal) as compared to the above question and then design the corresponding IIR Chebyshev and Butterworth filter. Repeat this for different set of tolerance specifications and observe the effect on the resulting filter parameters of both the filters. Comment on the results observed.

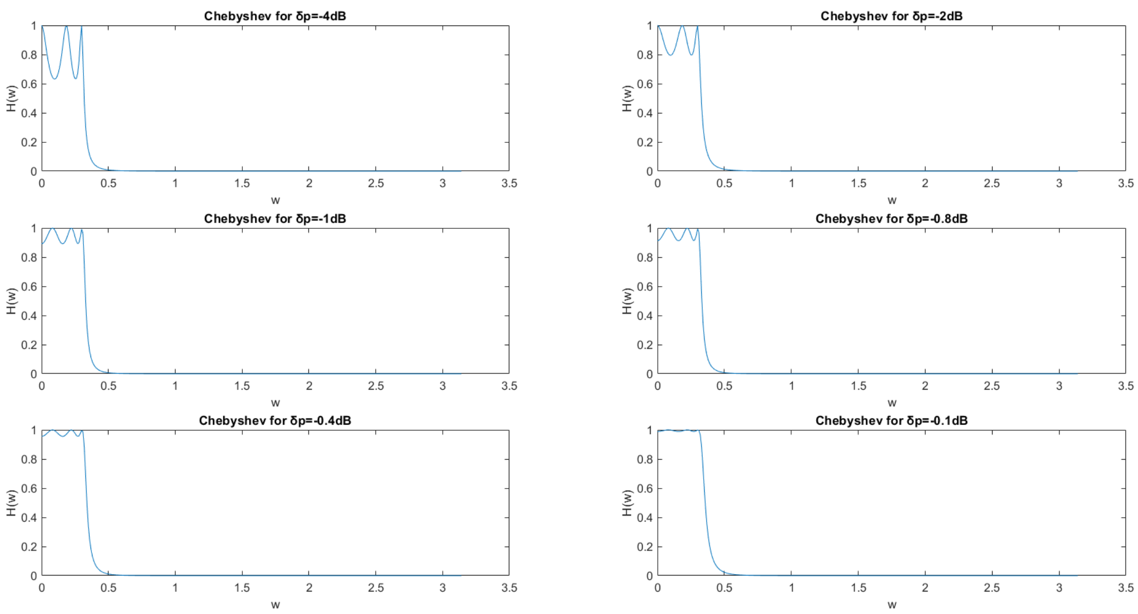
```
Sampling_Frequency = 2000;
Pass_Band_frequency  = 100;
Stop_band_frequency = 500;
Pass_Band_attenuation = -0.1;
Stop_Band_attenuation = -100;
```

Part 1: changing the tolerance

```
Pass_Band_analysis = [-4,-2,-1,-0.8,-0.4,-0.1];
```

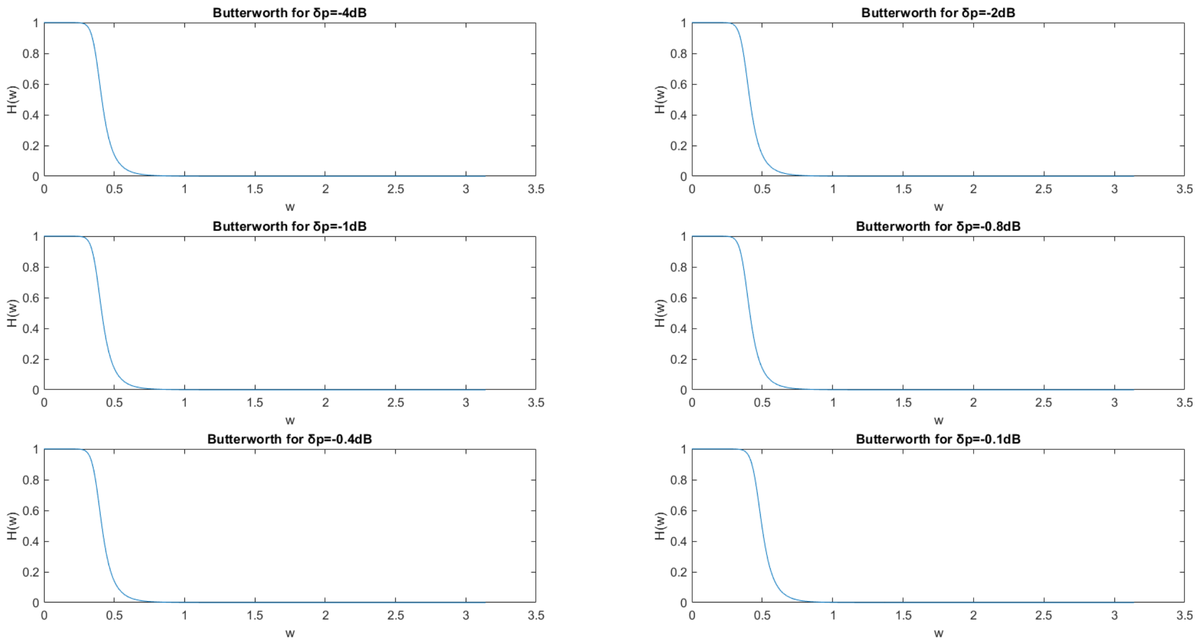
Chebyshev

```
clf
for i= 1:6
Pass_Band_attenuation = Pass_Band_analysis(i);
[n1,wn1] =
cheb1ord(2*Pass_Band_frequency/Sampling_Frequency,2*Stop_band_frequency/Sampli
ng_Frequency,-Pass_Band_attenuation,-Stop_Band_attenuation);
[a1,b1] = cheby1(n1,-Pass_Band_attenuation,wn1);

w1 = 0:0.01:pi;
Hw1 = freqz(a1,b1,w1);

subplot(3,2,i)
plot(w1,abs(Hw1))
title(['Chebyshev for δp=',num2str(Pass_Band_attenuation),'dB'])
ylabel('H(w)')
xlabel('w')
end
```



ButterWorth:

```
clf
for i= 1:6
Pass_Band_attenuation = Pass_Band_analysis(i);
[n2,wn2] =
buttord(2*Pass_Band_frequency/Sampling_Frequency,2*Stop_band_frequency/Samplin
g_Frequency,-Pass_Band_attenuation,-Stop_Band_attenuation);
[a2,b2] = butter(n2,wn2);

w2 = 0:0.01:pi;
Hw2 = freqz(a2,b2,w2);

subplot(3,2,i)
plot(w2,abs(Hw2))
title(['Butterworth for δp=',num2str(Pass_Band_attenuation),'dB'])
ylabel('H(w)')
xlabel('w')
end
```



Part 2: changing the bandwidth using Wn

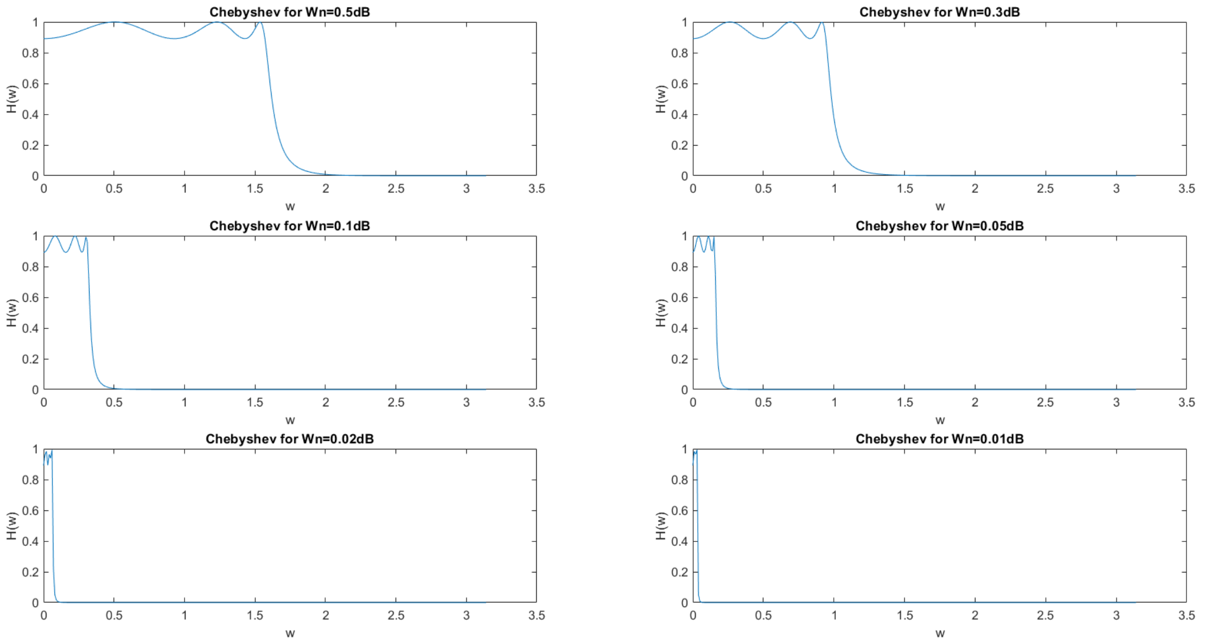
```
Pass_Band_attenuation=-1;
bandwidths = [0.5,0.3,0.1,0.05,0.02,0.01];
```

Chebyshev

```
clf
for i= 1:6
[a1,b1] = cheby1(n1,-Pass_Band_attenuation,bandwidths(i));
w1 = 0:0.01:pi;
Hw1 = freqz(a1,b1,w1);

```

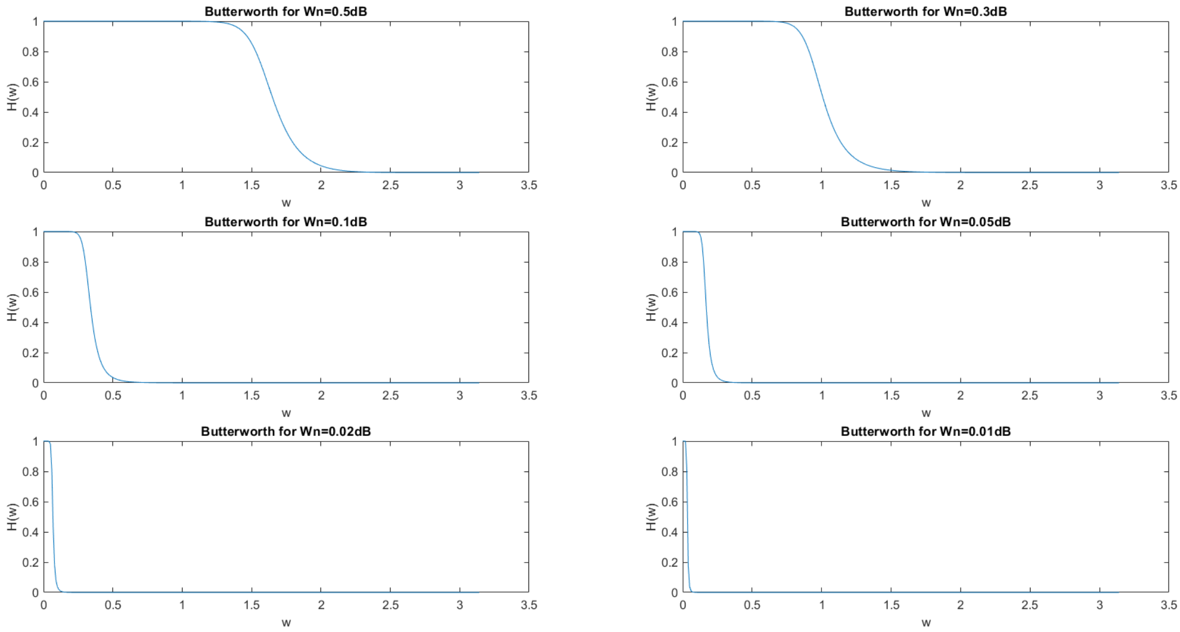
```
subplot(3,2,i)
```

```
plot(w1,abs(Hw1))
title(['Chebyshev for Wn=',num2str(bandwidths(i)),'dB'])
ylabel('H(w)')
xlabel('w')
end
```



ButterWorth

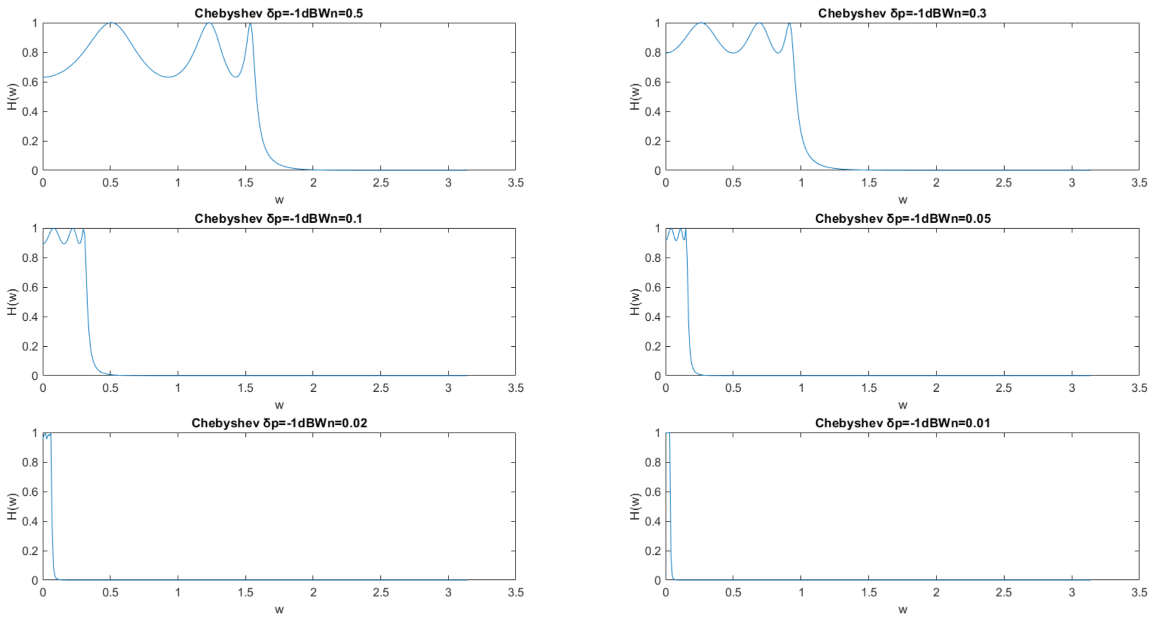
```
clf
for i= 1:6
[n2,wn2] =
buttord(2*Pass_Band_frequency/Sampling_Frequency,2*Stop_band_frequency/Samplin
g_Frequency,-Pass_Band_attenuation,-Stop_Band_attenuation);
[a2,b2] = butter(n2,bandwidths(i));


w2 = 0:0.01:pi;
Hw2 = freqz(a2,b2,w2);


subplot(3,2,i)
plot(w2,abs(Hw2))
title(['Butterworth for Wn=',num2str(bandwidths(i)),'dB'])
ylabel('H(w)')
xlabel('w')
end
```



Part 3: Changing both tolerance and bandwidth

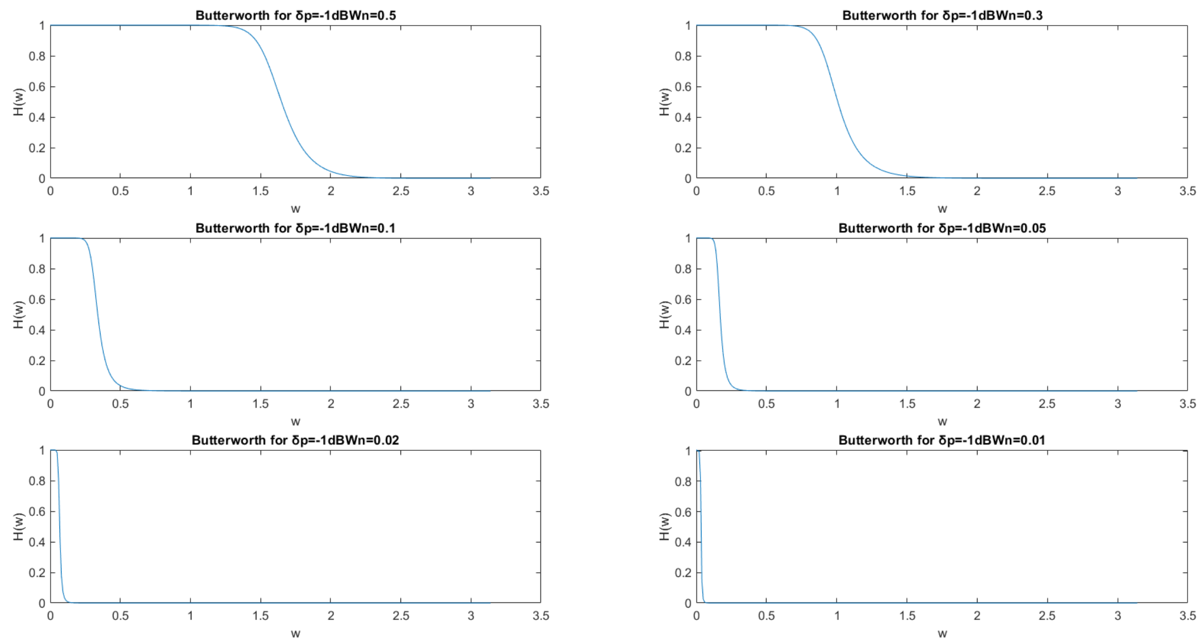
Chebyshev

```
for i = 1:6
[a1,b1] = cheby1(n1,-Pass_Band_analysis(i),bandwidths(i));
w1 = 0:0.01:pi;
Hw1 = freqz(a1,b1,w1);
subplot(3,2,i)
plot(w1,abs(Hw1))
title(['Chebyshev
δp=',num2str(Pass_Band_attenuation),'dB','Wn=',num2str(bandwidths(i))])
ylabel('H(w)')
xlabel('w')
end
```



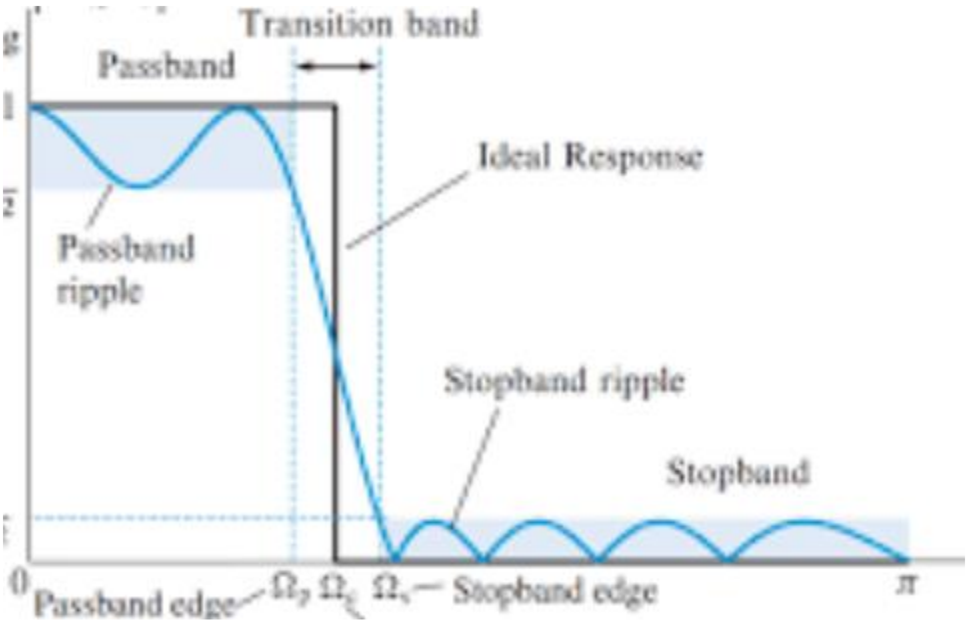
Butterworth

```
for i=1:6
[n2,wn2] =
buttord(2*Pass_Band_frequency/Sampling_Frequency,2*Stop_band_frequency/Samplin
g_Frequency,-Pass_Band_analysis(i),-Stop_Band_attenuation);
[a2,b2] = butter(n2,bandwidths(i));


w2 = 0:0.01:pi;
Hw2 = freqz(a2,b2,w2);


subplot(3,2,i)
plot(w2,abs(Hw2))
title(['Butterworth for
δp=',num2str(Pass_Band_attenuation),'dB','Wn=',num2str(bandwidths(i))])
ylabel('H(w)')
xlabel('w')
end
```



Question 4:

Q.4. a) Design an FIR digital low pass filter for the same set of filter specifications as used in the above IIR LPF design using different windows discussed in the class, i.e,



dp = 0.94, ds = 0.01, wp (digital)= 0.314 , ws = 1.57,

wc = (0.314+1.57)/2 = 0.94

digital frequency = 2*pi*f / fs, where fs = sampling frequency = 2*max analog frequency.

Now we have all values in which domain ?

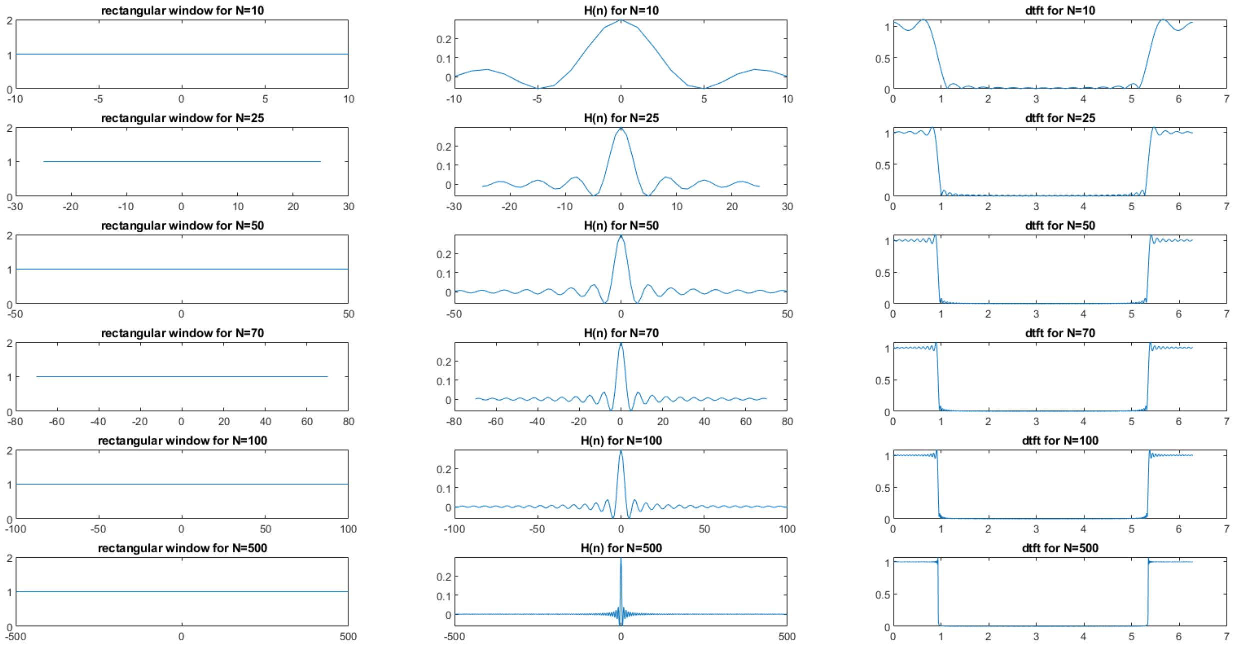
Defining

```
syms n
wp = 0.314;
ws = 1.57;
wc = 0.94;


N = [10,25,50,70,100,500];
ideal_lpf = sin(wc*n)/(pi*n);
```

Rectangular window

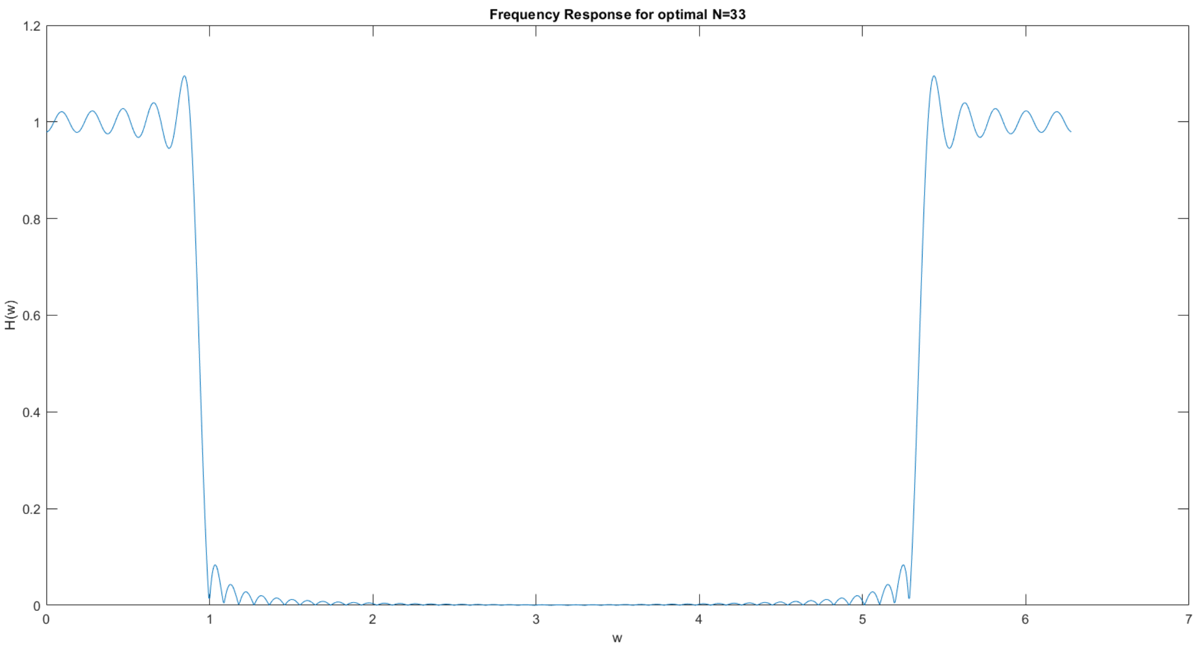
```
for index = 1:length(N)
    rect_window = ones(1,2*N(index)+1);
    n_subs = -N(index)+0.0001:N(index)+0.0001;
    practical_lpf = subs(ideal_lpf,n,n_subs);
```

```
    subplot(length(N),3,3*index-2)
    plot(-N(index):N(index),rect_window)
    title(['rectangular window for N=',num2str(N(index))])


    subplot(length(N),3,3*index-1)
    plot(n_subs-0.001,practical_lpf)
    title(['H(n) for N=',num2str(N(index))])


    signal_length = 2*N(index)+1;
    k = 0:signal_length+1000-1;
    wdtft = 2*pi*k/(signal_length+1000);
    dtft = fft(double(practical_lpf),signal_length+1000);
    %a = dtft(double(practical_lpf),-
N(index):N(index),0:2*pi/signal_length:2*pi*(signal_length-1)/signal_length)
    subplot(length(N),3,3*index)
    plot(wdtft,abs(dtft))
    title(['dtft for N=',num2str(N(index))])
end
```



Finding optimal N (Rect)

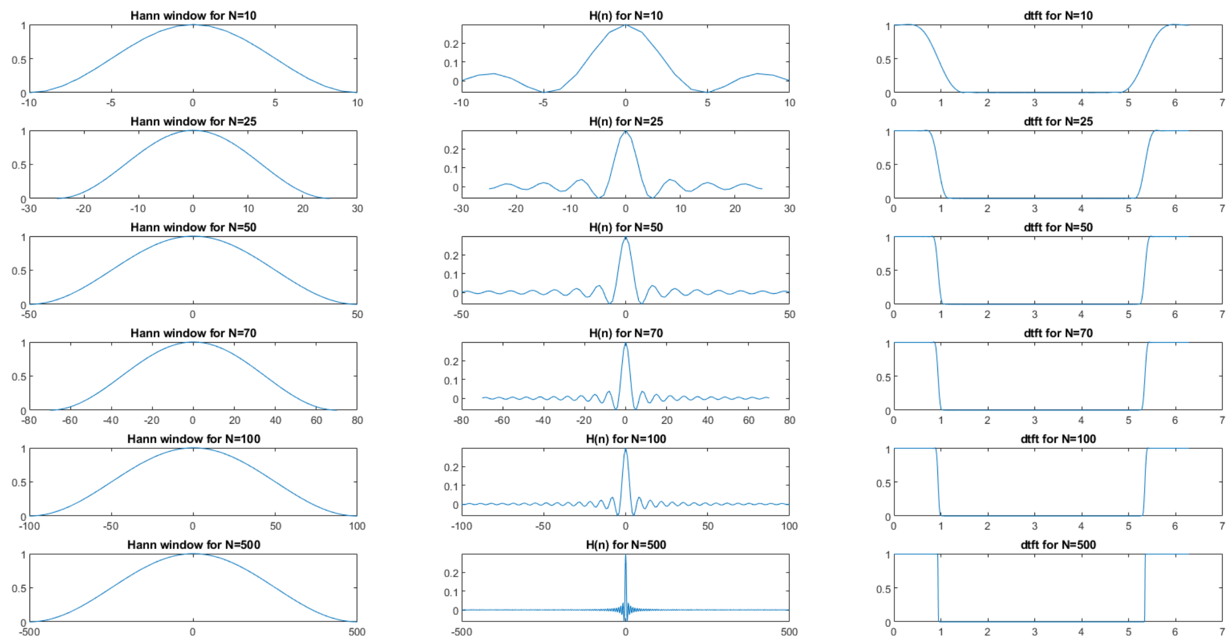
```
for i = 1:100


N_optimal_rect = i;
practical_lpf = subs(ideal_lpf,n,-
N_optimal_rect+0.0001:N_optimal_rect+0.0001);
rect_window = -N_optimal_rect:N_optimal_rect;


signal_length = 2*N_optimal_rect+1;
k = 0:signal_length+1000-1;
wdtft = 2*pi*k/(signal_length+1000);
dtft = fft(double(practical_lpf),signal_length+1000);
clf
passes = evaluate(abs(dtft),wdtft);
if(passes==1)
    break
end
end

plot(wdtft,abs(dtft))
title(['Frequency Response for optimal N=',num2str(N_optimal_rect)])
ylabel('H(w)')
xlabel('w')
```



Hann window

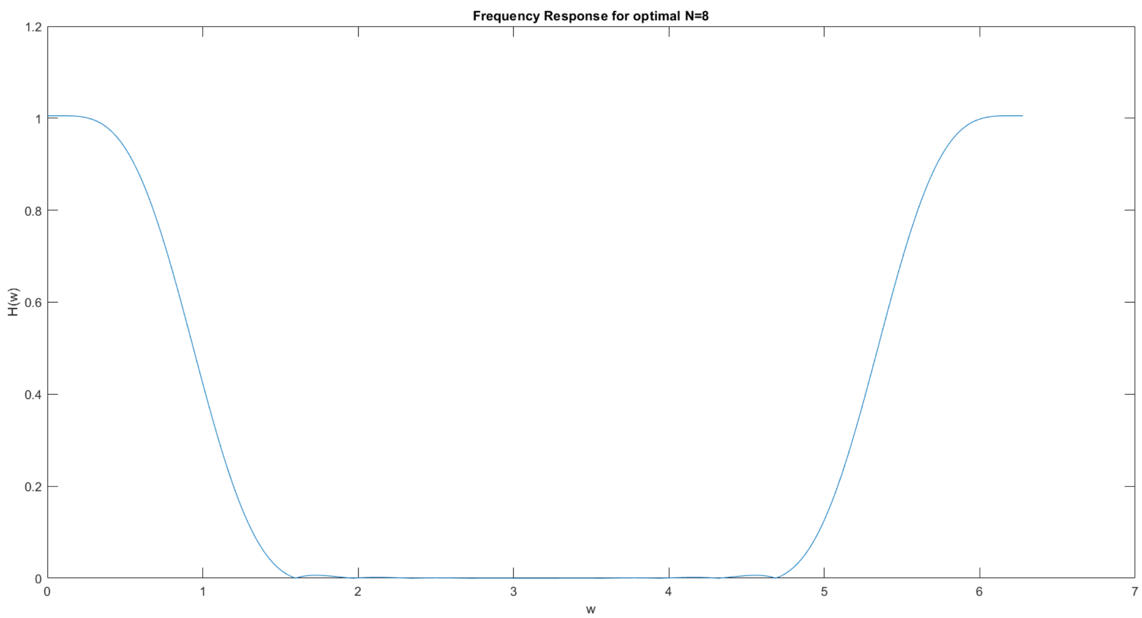
```
for index = 1:length(N)
    rect_window = ones(1,2*N(index)+1);

    %% Hann window
    n_hann = 0:2*N(index);
```

```
    hann_window = 0.5.*(1-cos((2.*pi.*n_hann)/(2*N(index)))).*rect_window;

    %% Truncated h(n)
    n_subs = -N(index)+0.0001:N(index)+0.0001;
    practical_lpf = subs(ideal_lpf,n,n_subs);


    %plotting hanning window
    subplot(length(N),3,3*index-2)
    plot(-N(index):N(index),hann_window)
    title(['Hann window for N=',num2str(N(index))])


    %plotting Truncated h(n)
    subplot(length(N),3,3*index-1)
    plot(n_subs-0.001,practical_lpf)
    title(['H(n) for N=',num2str(N(index))])


    signal_length = 2*N(index)+1;
    k = 0:signal_length+1000-1;
    wdtft = 2*pi*k/(signal_length+1000);


    % dtft of h(n)*hann window
    dtft = fft(double(practical_lpf.*hann_window),signal_length+1000);
    subplot(length(N),3,3*index)
    plot(wdtft,abs(dtft))
    title(['dtft for N=',num2str(N(index))])
end
```



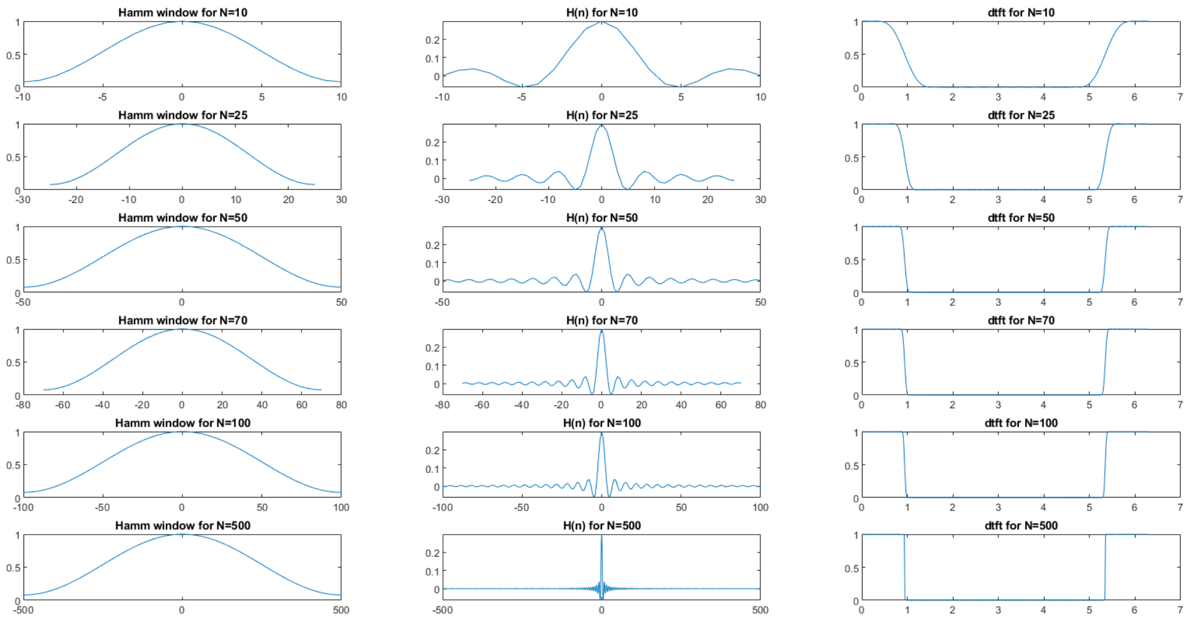
Finding optimal N (Hann)

```
for i = 1:100


N_optimal_hann = i;
practical_lpf = subs(ideal_lpf,n,-
N_optimal_hann+0.0001:N_optimal_hann+0.0001);
rect_window = ones(1,2*N_optimal_hann+1);


n_hann = 0:2*N_optimal_hann;
hann_window = 0.5.*(1-cos((2.*pi.*n_hann)/(2*N_optimal_hann))).*rect_window;


signal_length = 2*N_optimal_hann+1;
k = 0:signal_length+1000-1;
wdtft = 2*pi*k/(signal_length+1000);
dtft = fft(double(practical_lpf.*hann_window),signal_length+1000);


clf
passes = evaluate(abs(dtft),wdtft);


if(passes==1)
    break
end
end


plot(wdtft,abs(dtft))
title(['Frequency Response for optimal N=',num2str(N_optimal_hann)])
ylabel('H(w)')
xlabel('w')
```



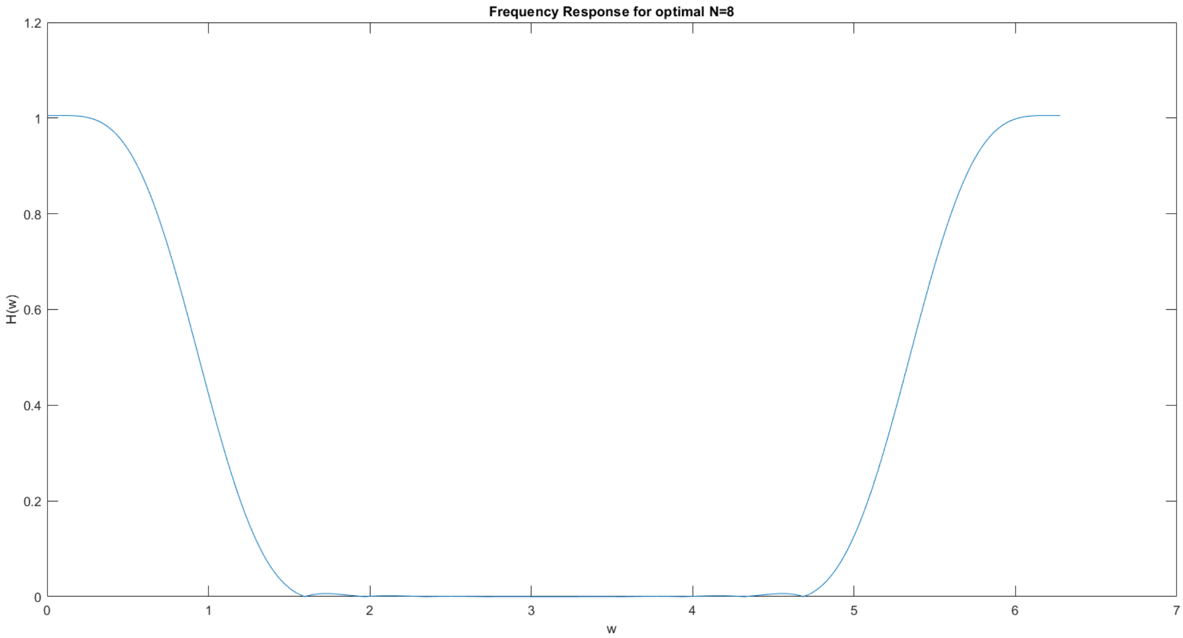
Hamming window

```
for index = 1:length(N)
    rect_window = ones(1,2*N(index)+1);

    %% Hann window
    n_hamm = 0:2*N(index);
    hamm_window = (0.54-0.46*cos((2.*pi.*n_hamm)/(2*N(index)))).*rect_window;

    %% Truncated h(n)
    n_subs = -N(index)+0.0001:N(index)+0.0001;
    practical_lpf = subs(ideal_lpf,n,n_subs);


    %plotting hamming window
    subplot(length(N),3,3*index-2)
    plot(-N(index):N(index),hamm_window)
    title(['Hamm window for N=',num2str(N(index))])


    %plotting Truncated h(n)
    subplot(length(N),3,3*index-1)
    plot(n_subs-0.001,practical_lpf)
    title(['H(n) for N=',num2str(N(index))])


    signal_length = 2*N(index)+1;
    k = 0:signal_length+1000-1;
    wdtft = 2*pi*k/(signal_length+1000);


    % dtft of h(n)*hamm window
    dtft = fft(double(practical_lpf.*hamm_window),signal_length+1000);
    subplot(length(N),3,3*index)
    plot(wdtft,abs(dtft))
    title(['dtft for N=',num2str(N(index))])
end
```



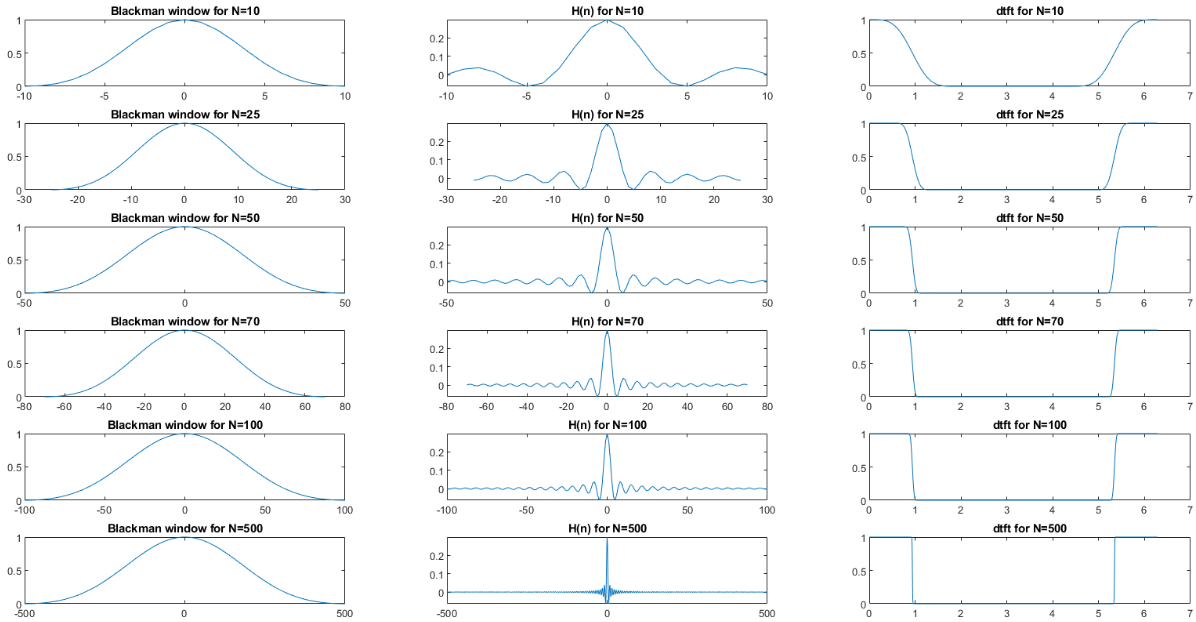
Finding optimal N (Hamm)

```
for i = 1:100
N_optimal_hamm = i;
practical_lpf = subs(ideal_lpf,n,-
N_optimal_hamm+0.0001:N_optimal_hamm+0.0001);
rect_window = ones(1,2*N_optimal_hamm+1);


n_hamm = 0:2*N_optimal_hamm;
hamm_window = 0.5.*(1-cos((2.*pi.*n_hamm)/(2*N_optimal_hamm))).*rect_window;


signal_length = 2*N_optimal_hamm+1;
k = 0:signal_length+1000-1;
wdtft = 2*pi*k/(signal_length+1000);
dtft = fft(double(practical_lpf.*hamm_window),signal_length+1000);


clf
passes = evaluate(abs(dtft),wdtft);
if(passes==1)
    break
end
end


plot(wdtft,abs(dtft))
title(['Frequency Response for optimal N=',num2str(N_optimal_hamm)])
ylabel('H(w)')
xlabel('w')
```



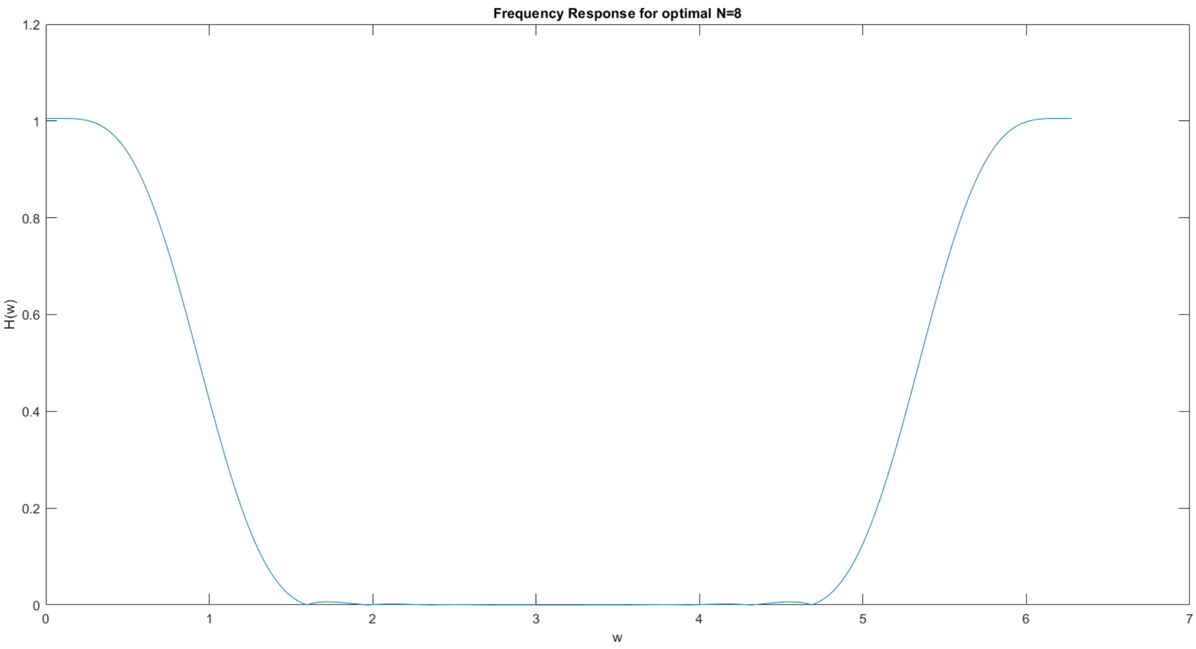
Blackman window

```
for index = 1:length(N)
    rect_window = ones(1,2*N(index)+1);

    %% Blackman window
    n_bw = 0:2*N(index);
    bw_window = (0.42-
0.5*cos((2.*pi.*n_bw)/(2*N(index)))+0.08*cos((4.*pi.*n_bw)/(2*N(index)))).*rec
t_window;

    %% Truncated h(n)
    n_subs = -N(index)+0.0001:N(index)+0.0001;
    practical_lpf = subs(ideal_lpf,n,n_subs);


    %plotting Blackman window
    subplot(length(N),3,3*index-2)
    plot(-N(index):N(index),bw_window)
    title(['Blackman window for N=',num2str(N(index))])


    %plotting Truncated h(n)
    subplot(length(N),3,3*index-1)
    plot(n_subs-0.001,practical_lpf)
    title(['H(n) for N=',num2str(N(index))])


    signal_length = 2*N(index)+1;
    k = 0:signal_length+1000-1;
    wdtft = 2*pi*k/(signal_length+1000);


    % dtft of h(n)*Blackman window
    dtft = fft(double(practical_lpf.*bw_window),signal_length+1000);
    subplot(length(N),3,3*index)
    plot(wdtft,abs(dtft))
    title(['dtft for N=',num2str(N(index))])
end
```



Finding optimal N (Blackman)

```
for i = 1:100


N_optimal_bw = i;
practical_lpf = subs(ideal_lpf,n,-N_optimal_bw+0.0001:N_optimal_bw+0.0001);
rect_window = ones(1,2*N_optimal_bw+1);


n_bw = 0:2*N_optimal_bw;
bw_window = 0.5.*(1-cos((2.*pi.*n_bw)/(2*N_optimal_bw))).*rect_window;


signal_length = 2*N_optimal_bw+1;
k = 0:signal_length+1000-1;
wdtft = 2*pi*k/(signal_length+1000);
dtft = fft(double(practical_lpf.*bw_window),signal_length+1000);


clf
passes = evaluate(abs(dtft),wdtft);


if(passes==1)
    break
end
end


plot(wdtft,abs(dtft))
title(['Frequency Response for optimal N=',num2str(N_optimal_bw)])
ylabel('H(w)')
xlabel('w')
```



<u>Evaluate Function:</u> (I have defined evaluate function which tells whether my filter satisfies the required specifications or not)

```
function bool = evaluate(dtft,wdtft)

bool = 1;
len = length(dtft);

for i = 1:len/2
    if(bool==0)
        break
    end

    if(wdtft(i)<0.314)
        if(dtft(i)>1.054 || dtft(i)<0.94)
            bool=0;
        end
    elseif(wdtft(i)>1.57)
        if(dtft(i)>0.01)
            bool=0;
        end
    end
end
end
```

Discussions:

2b(a) Plot the input and output signals and verify whether the desired specifications are satisfied or not.

We have defined a test signal to be sum of 2 sine signals of frequency 0.1π (low frequency) and 0.5π (High frequency), where the cutoff frequency of filter is 0.1π. By sending the signal which consisted of a low and a high frequency component through a Butterworth Low pass filter, only the low frequency term (0.1π) remains which we see as a sine wave as observed in the last figure. Hence the designed Low pass filter functions properly.

3a. Decrease the passband and/or stopband tolerance level (thus making the filter closer to ideal) as compared to the above question and then design the corresponding IIR Chebyshev and Butterworth filter. Repeat this for different set of tolerance specifications and observe the effect on the resulting filter parameters of both the filters. Comment on the results observed.

Chebyshev:

By decreasing the pass band tolerance we can see that the width of ripples is getting reduced and by further decreasing its almost smooth to a naked eye.

By decreasing Wn (cutoff frequency), which is directly proportional to Bandwidth, we can observe that the graph moves left. Becoming close to an Ideal LPF.

ButterWorth:

By decreasing the pass band tolerance we can see that there is no major difference in the graph but slightly the response in pass band becomes more horizontal indicating close to ideal nature.

By decreasing Wn (cutoff frequency), which is directly proportional to Bandwidth, we can observe that the graph moves left. Becoming close to an Ideal LPF.

4b. Compare the order of these filters with the order of the analog filters, i.e., Butterworth and Chebyshev filters.

For the given specifications, orders of
Chebyshev = 6,
Butterworth = 7,
FIR filter using rectangular window = 33
FIR filter using Hann window = 8
FIR filter using Hamming window = 8
FIR filter using Blackmann window = 8

We can observe that the order of FIR filters in all cases is greater than that of IIR filters in the given case.

4c) Observe the effect of increasing the window lengths on the filter frequency response and comment about the same.

By increasing N, in all the above cases the transition region decreases. Hence transition region acts ideally at very high values of N. But it doesn't affect the overshoot or undershoot.