Department of Electrical Engineering
Indian Institute of Technology Kharagpur

Digital Signal Processing Laboratory (EE39203)

Autumn, 2022-23

Experiment 5 **Digital Filter Design**

Slot: Date:

Student Name: Roll No.:

Grading Rubric

	Tick the best applicable per row			Points
	Below Expectation	Lacking in Some	Meets all Expectation	
Completeness of the report				
Organization of the report (5 pts) *With cover sheet, answers are in the same order as questions in the lab, copies of the questions are included in report, prepared in LaTeX*				
Quality of figures (5 pts) *Correctly labelled with title, x-axis, y-axis, and name(s)*				
Understanding and implementation of simple FIR filter (35 pts) *Difference eq., flow diagram, impulse response, plots of magnitude response, plots of original and filtered signals and their DTFT, matlab code, questions*				
Understanding and implementation of simple IIR filter (35 pts) *Difference eq., flow diagram, impulse response, plots of magnitude response, plots of original and filtered signals and their DTFT, matlab code, questions*				
Understanding parameters of lowpass filter design (20 pts) *Magnitude response plots with marked regions, questions.*				
			TOTAL (100 pts)	

Total Points (100): TA Name: TA Initials:

Digital Signal Processing Laboratory (EE39203)
Experiment 5
Digital Filter Design

Name: Bandaru Sri Naga Akhil
Roll Number: 20EE10085

1 Design of a Simple IIR Filter

Design a simple second order IIR filter with the transfer function given by

$$H_i(z) = \frac{1-r}{1-2r\cos\theta z^{-1}+r^2 z^{-2}}$$

Calculate the magnitude of the filter's frequency response $|H_i\left(e^{j\omega}\right)|$ on $|\omega| < \pi$ for $\theta = \pi/3$ and the following three values of r.

- $r = 0.99$
- $r = 0.9$
- $r = 0.7$

Put all three plots on the same figure using the subplot command.

Submit the difference equation, system diagram and the analytical expression of the impulse response for $H_i(z)$. (Hint: The frequency response of the system can be obtained by restricting the z-transform to the unit circle. So the DTFT of $h_i[n]$ is $H_i\left(e^{j\omega}\right)$. Therefore, to get $h_i[n]$, you can take the inverse Fourier transform of $H_i\left(e^{j\omega}\right)$.)

Also submit the plot of the magnitude of the frequency response for each value of r. Explain how the value of r affects this magnitude.

Difference Equation:

$$H_i(z) = \frac{1-r}{1-2r\cos\theta z^{-1} + r^2 z^{-2}}$$

Linearity Property:

$$a \times h[n] + b \times g[n] \rightarrow a \times H(z) + b \times G(z)$$

with ROC atleast $R_h \cap R_g$

Time shifting Property:

$$h[n-n_0] \rightarrow z^{-n_0} H(z)$$

with ROC R_h, except z=0 if $n_0 > 0$ and z=0 if $n_0 < 0$.

Hence the inverse Z-Transform of 1 is $\delta[n]$, z^{-1} is $\delta[n-1]$ and z^{-2} is $\delta[n-2]$

Let input be x[n] with Z-transform be X(z) and output be y[n] with Z-transform Y(z).

$$H_i(z) = \frac{Y(z)}{X(z)} = \frac{1-r}{1-2r\cos\theta z^{-1} + r^2 z^{-2}}$$

Cross-multiplying and using time shifting, linearity properties,

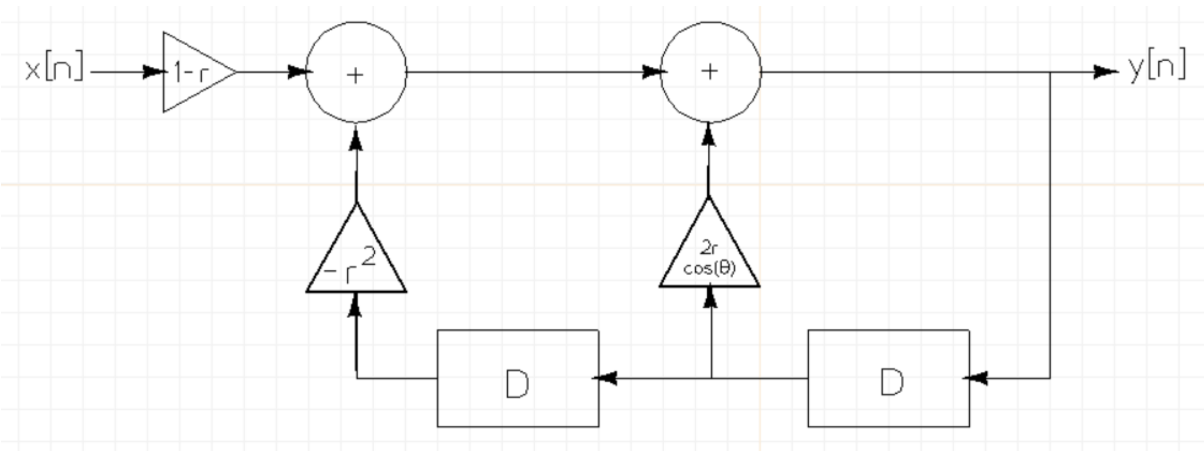
we get the Difference Equation as

$$y[n] - 2r\cos(\theta)y[n-1] + r^2 y[n-2] = (1-r)x[n]$$

For $\theta = \pi/3$, The Difference Equation is

$$y[n] - ry[n-1] + r^2 y[n-2] = (1-r)x[n]$$

System Diagram:



$$y[n] = ry[n-1] - r^2 y[n-2] + (1-r)x[n]$$

Impulse Response for the filter $H_i(z)$:

$$H_i(z) = \frac{1-r}{1-2r\cos(\theta)z^{-1}+r^2z^{-2}} = \frac{1-r}{(1-re^{j\theta}z^{-1})(1-re^{-j\theta}z^{-1})}$$

Using partial fractions, we can write

$$H_i(z) = \frac{(1-r)}{e^{j\theta}-e^{-j\theta}} \times \left(\frac{e^{j\theta}}{1-re^{j\theta}z^{-1}} - \frac{e^{-j\theta}}{1-re^{-j\theta}z^{-1}}\right)$$

Since its ROC must include the unit circle, we get the Impulse Response as

$$h_i[n] = \frac{(1-r)}{e^{j\theta}-e^{-j\theta}} \times \left(e^{j\theta}(re^{j\theta})^n u[n] - e^{-j\theta}(re^{-j\theta})^n u[n]\right)$$

$$h_i[n] = (1-r)r^n \times \frac{\frac{e^{j(n+1)\theta}-e^{-j(n+1)\theta}}{2j}}{\frac{e^{j\theta}-e^{-j\theta}}{2j}} \times u[n]$$

Therefore, we get the impulse response for the filter $H_i(z)$ which is

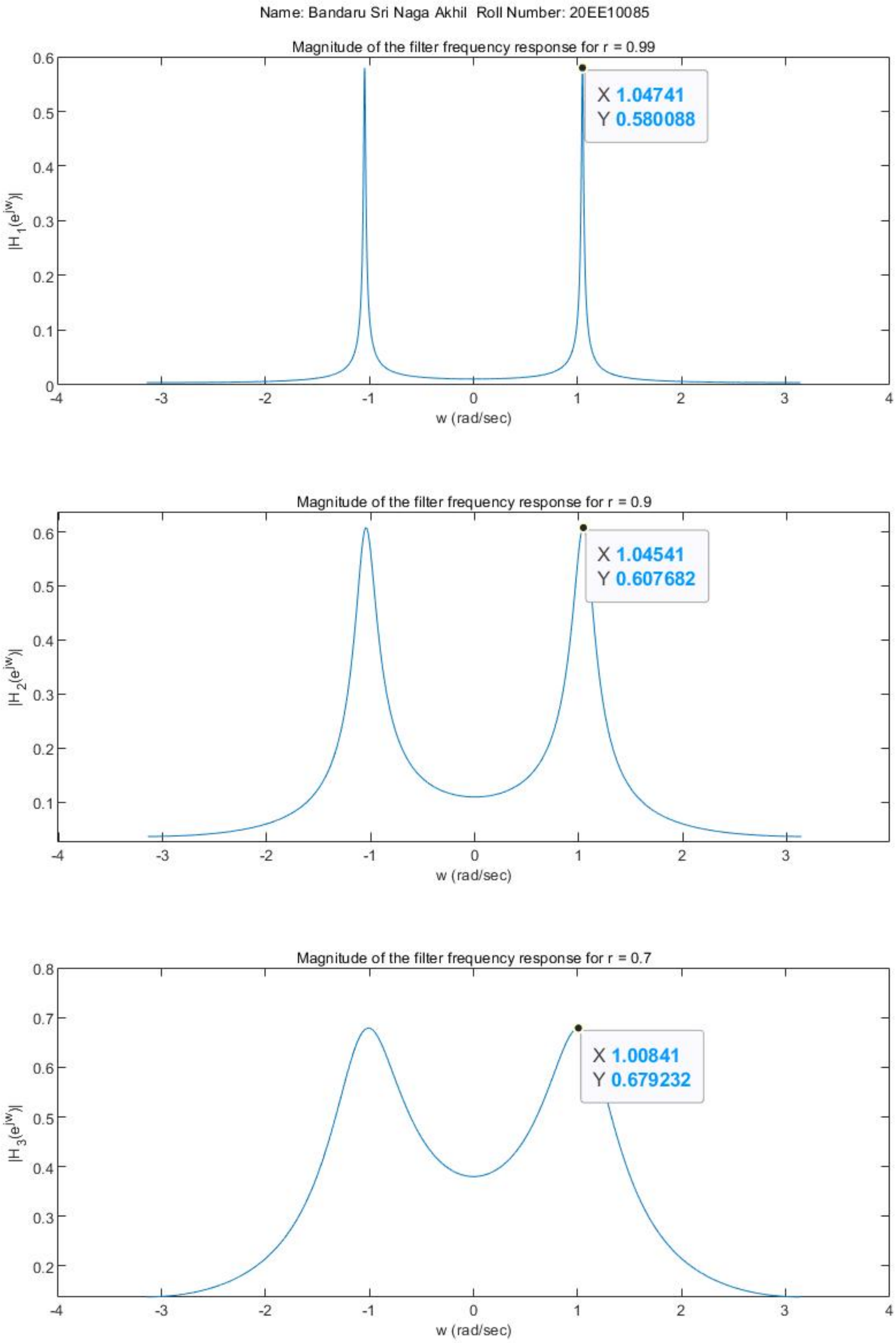
$$h_i[n] = (1-r)r^n \frac{\sin((n+1)\theta)}{\sin(\theta)} u[n]$$

Matlab code for plotting the magnitude of the filter's frequency response $|H_i(e^{j\omega})|$ as a function of ω on the interval $-\pi < \omega < \pi$ for r = 0.99, 0.9, 0.7 with $\theta = \pi/3$:

```
w = -pi:0.001:pi; % given input frequency range w = (-pi,pi)

ft1 = ft(w,0.99); % For r = 0.99
ft2 = ft(w,0.9);  % For r = 0.9
ft3 = ft(w,0.7);  % For r = 0.7

subplot(311)
sgtitle('Name: Bandaru Sri Naga Akhil  Roll Number: 20EE10085');
plot(w,abs(ft1));
subtitle('Magnitude of the filter frequency response for r = 0.99');
xlabel('w (rad/sec)');
ylabel('|H_1(e^j^w)|');

subplot(312)
plot(w,abs(ft2));
subtitle('Magnitude of the filter frequency response for r = 0.9');
xlabel('w (rad/sec)');
ylabel('|H_2(e^j^w)|');

subplot(313)
plot(w,abs(ft3));
subtitle('Magnitude of the filter frequency response for r = 0.7');
xlabel('w (rad/sec)');
ylabel('|H_3(e^j^w)|');
orient('tall');
```

```
%function for the frequency response
%H(e^jw) = (1-r)/(1-2r cos(theta)e^(-jw) + r^2 e^(-2jw))
%For theta = pi/3


function y = ft(w,r)
    j = sqrt(-1);
    y = zeros(1,length(w));
    for i = 1:length(w)
        y(i) = (1-r)/(1-2*cos(pi/3)*r*exp(-j*w(i)) + r^2 * exp(-2*j*w(i)));
    end
end
```

Plots of magnitude response for r = 0.99, 0.9 and 0.7 with $\theta = \pi/3$



Discussion:

The transfer function of a second order IIR filter is given by

$$H_i(z) = \frac{1-r}{1-2r\cos\theta z^{-1} + r^2 z^{-2}} = \frac{1-r}{(1-re^{j\theta}z^{-1})(1-re^{-j\theta}z^{-1}}$$

It has the poles at $z = re^{j\theta}$ and $z = re^{-j\theta}$.

The peak of the magnitude response is obtained at frequency ω_c which is given by

$$\cos(\omega_c) = \frac{1+r^2}{2r} \times \cos(\theta)$$

The peak is lower than the pole frequency for $0 < \theta < \pi/2$ and higher than the pole frequency for $\pi/2 < \theta < \pi$. Hence, we get different peak locations for different values of r in the plots which are less than the pole frequency ($\theta = \pi/3$).

As the poles become closer to the unit circle, the peak occurs at $\omega_c \approx \theta$. Hence, in the case of r = 0.99, we get peak at $\theta = \pi/3 = 1.047$ rad/sec.

Poles amplify the signals that are nearer to their frequencies. Hence, these filters are used to filter certain frequencies from other adjacent frequencies. These are also known as Discrete Time Resonators because, it has large magnitude response, that is, it resonates in the vicinity of the pole locations.

From the plots, we can observe that, as the value of r becomes closer to 1 (r is always less than 1, to keep the system stable), the magnitude of the frequency response, becomes narrower at the pole locations compared to its adjacent frequencies leading to a better filtering action.

2 Using the IIR Filter to separate a modulated sinusoid from background noice

Use the IIR filter $H_i(z)$ to seperate a modulated sinusoid from the background noise.

Download the file pcm.mat and load it into the Matlab workspace using the command load pcm. Play pcm using the sound command. Plot 101 samples of the signal for indices $(100 : 200)$, and then compute the magnitude of the DTFT of 1001 samples of pcm using the time indices (100:1100). Plot the magnitude of the DTFT samples versus radial frequency for $|\omega| < \pi$. Use the IIR filter described above to amplify the desired signal, relative to the background noise.

The pcm signal is modulated at 3146 Hz and sampled at 8kHz. Use these values to calculate the value of θ for the filter $H_i(z)$.

Write a Matlab function IIRfilter(x) that implements the filter $H_i(z)$. Use a for loop to implement the recursive difference equation. Use the calculated value of θ and $r = 0.995$. Assume that $y[n]$ is equal to 0 for negative values of n. Apply the new command IIRfilter to the signal pcm to separate the desired signal from the background noise, and listen to the filtered signal to hear the effects.

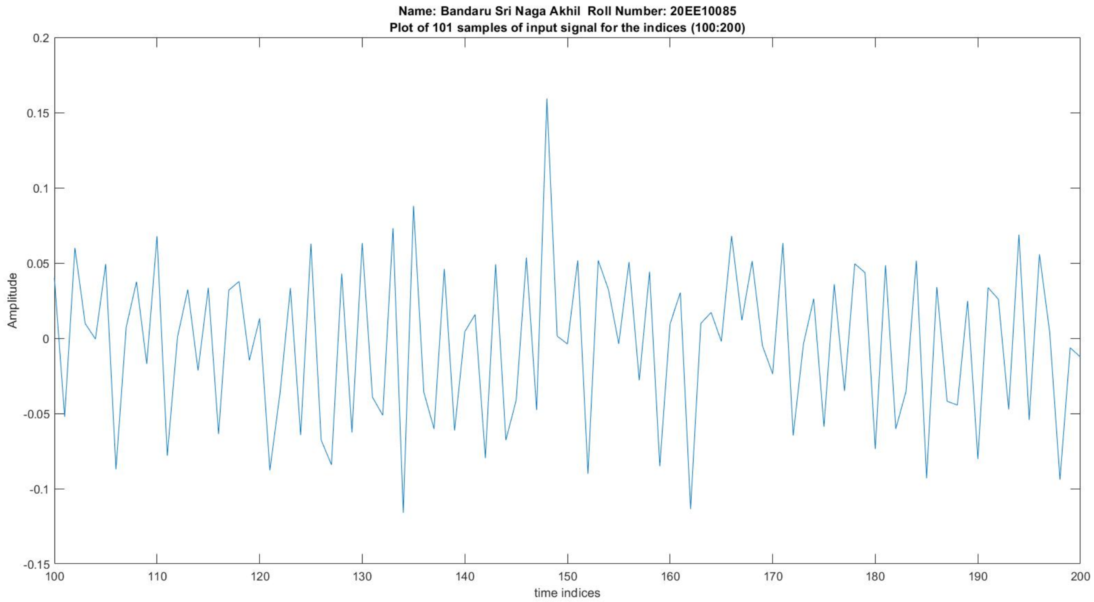
Plot the filtered signal for indices (100:200), and then compute the DTFT of 1001 samples of the filtered signal using the time indices (100:1100). Plot the magnitude of this DTFT. In order to see the DTFT around $\omega = \theta$ more clearly, plot also the portion of this DTFT for the values of ω in the range $[\theta - 0.02, \theta + 0.02]$. (Use the calculated value of θ)

Comment on how the signal looks and sounds before and after filtering. How would you expect changes in r to change the filtered output? Would a value of $r = 0.9999999$ be effective for this application? Why might such a value for r be ill-advised? (Consider the spectrum of the desired signal around $\omega = \theta$.)

Matlab code for plotting 101 samples of the input signal for indices (100:200):

```
load pcm;      % importing pcm file into MATLAB
sound(pcm);    % playing the audio of pcm file
audiowrite('pcm1.ogg',pcm,8000); % saving the audio into .ogg format
index = 100:1:200;  % defining an index interval from 100 to 200 samples

% plotting the 101 samples of input signal for indices (100,200)
plot(index,pcm(index));
title({'Name: Bandaru Sri Naga Akhil  Roll Number: 20EE10085';
    'Plot of 101 samples of input signal for the indices (100:200)'});
xlabel('time indices');
ylabel('Amplitude');
```

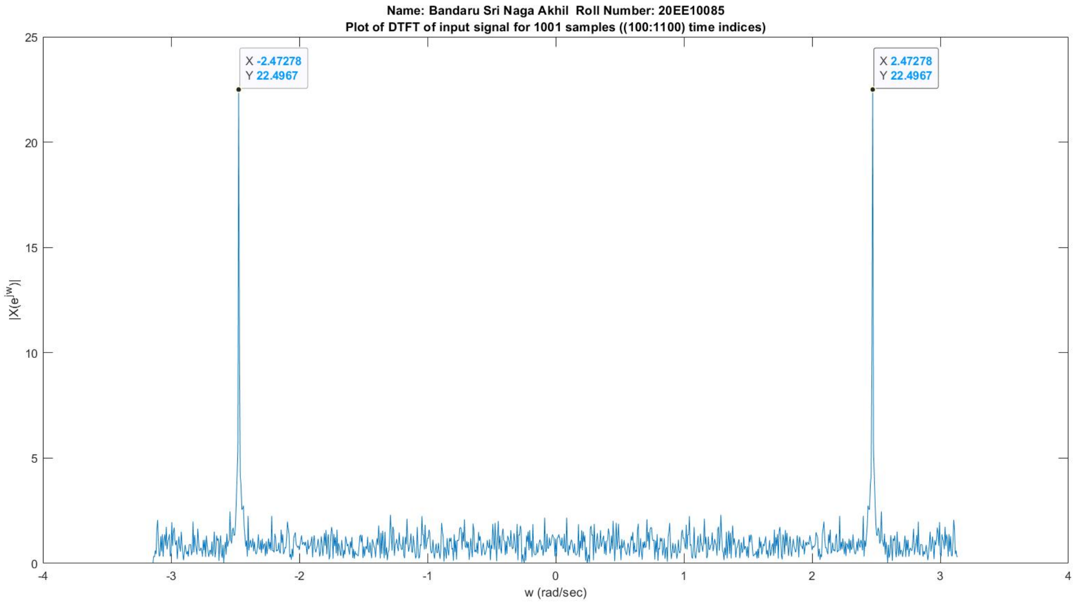
Plot of 101 samples of input signal for indices (100:200):



Matlab code for plotting the magnitude of DTFT of 1001 samples of input signal (using the time indices (100:1100)):

```
load pcm;      % importing pcm file into MATLAB

index = 100:1:1100; % defining a index interval from 100 to 1100
% Computing the DTFT of input signal over the above range
[X,W] = DTFT(pcm(index),0);

% plotting the magnitude of the DTFT of 1001 samples of input signal
% (for the time indices (100:1100))
plot(W, abs(X));
title({'Name: Bandaru Sri Naga Akhil  Roll Number: 20EE10085';
    'Plot of DTFT of input signal for 1001 samples ((100:1100) time indices)'});
ylabel('|X(e^j^w)|');
xlabel('w (rad/sec)');
```

Plot of magnitude of DTFT of 1001 samples of input signal (using the time indices (100:1100)):



Matlab code for implementing the IIRfilter function:

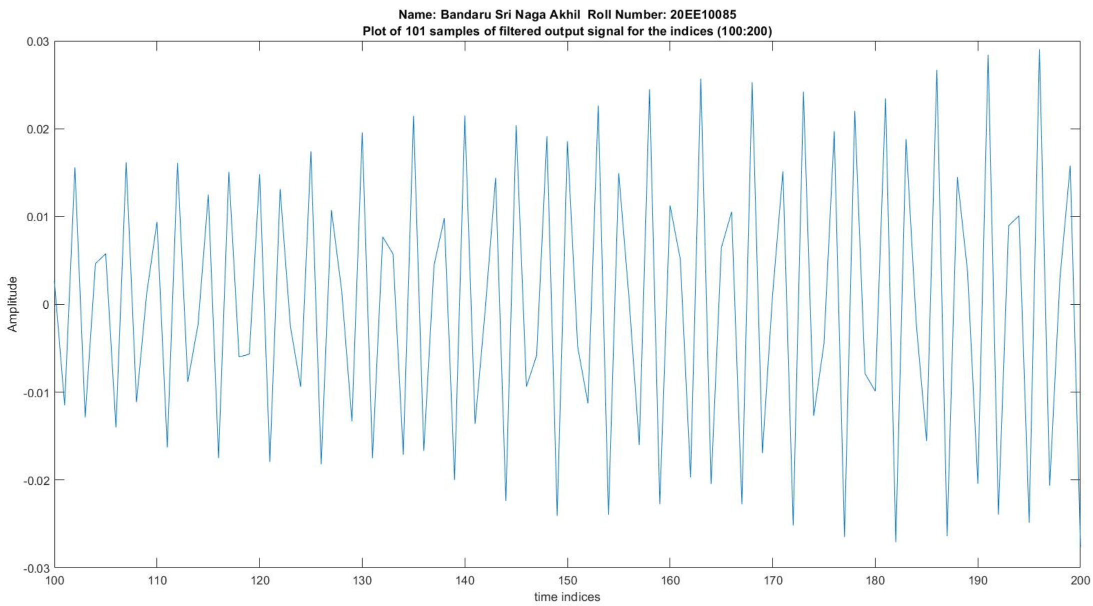
```
% IIR Filter Function
function y = IIRFilter(x,theta,r)
    y = zeros(1,length(x));    % Initialising an output vector with zeros
    % assuming y[n] = 0 for negatvie values of n
    % y[n] = -r^2 y[n-2] + r y[n-1] + (1-r) x[n]
    for i = 1:1:length(x)
        if i==1
            y(i) = x(i)*(1-r);
        elseif i==2
            y(i) =  2*r*cos(theta)*y(i-1) + (1-r)*x(i);
        else
            y(i) = -r^2 * y(i-2) + 2*r*cos(theta)*y(i-1) + (1-r)*x(i);
        end
    end
end
```

Matlab code for plotting 101 samples of the filtered output signal for indices (100:200):

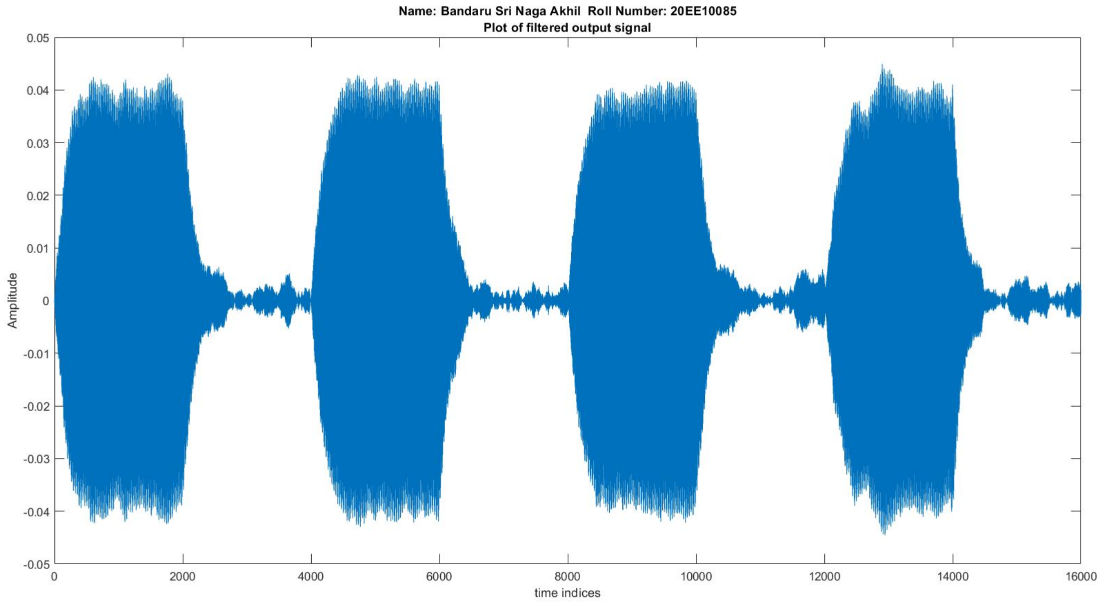
```
load pcm;      % importing pcm file into MATLAB
index = 100:1:200; % defining an index interval from 100 to 200 samples
% defining an index interval over all the entire samples of pcm
index1 = 1:1:length(pcm);
```

```
f1 = 3146; % modulated frequency
f2 = 8000; % sampled frequency
theta1 = (f1/f2)*2*pi; % calculation of frequency at which peak occurs
r1 = 0.995; % given r value

fs = IIRFilter(pcm,theta1,r1); % filtered output signal
sound(fs); % playing the audio of the filtered signal
audiowrite('pcm2.ogg',fs,8000); % saving the audio into .ogg format

% plotting the IIR filter output for 101 samples (for indices (100:200))
figure (1)
plot(index,fs(index));
title({'Name: Bandaru Sri Naga Akhil  Roll Number: 20EE10085';
    'Plot of 101 samples of filtered output signal for the indices (100:200)'});
xlabel('time indices');
ylabel('Amplitude');

% plotting the IIR filter output
figure(2)
plot(index1,fs);
title({'Name: Bandaru Sri Naga Akhil  Roll Number: 20EE10085';
    'Plot of filtered output signal'});
xlabel('time indices');
ylabel('Amplitude');
```

Plot of 101 samples of filtered output signal for indices (100:200):



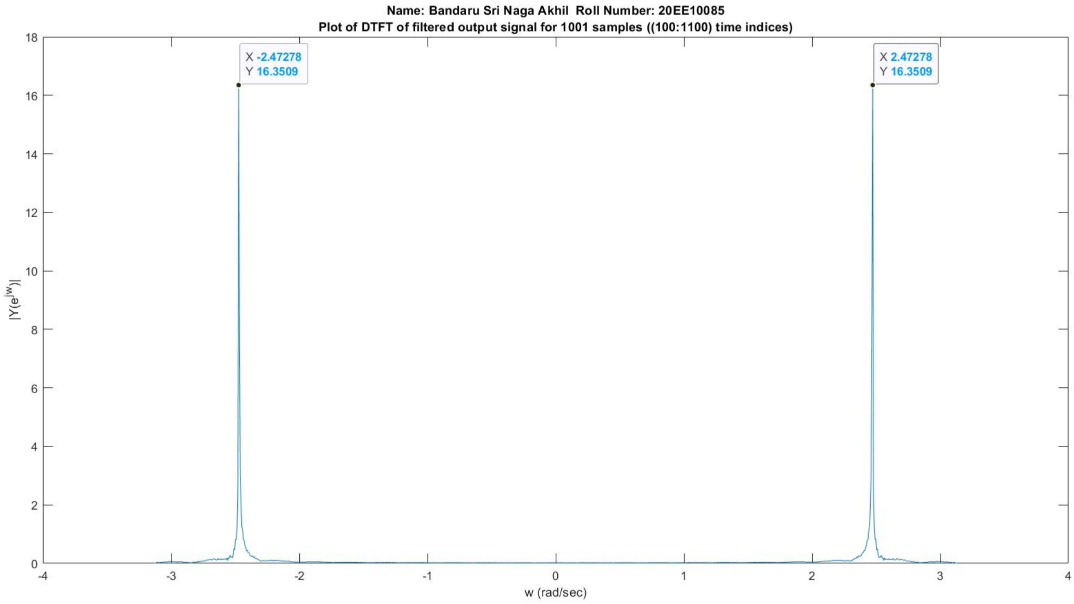
Plot of the filtered output signal:



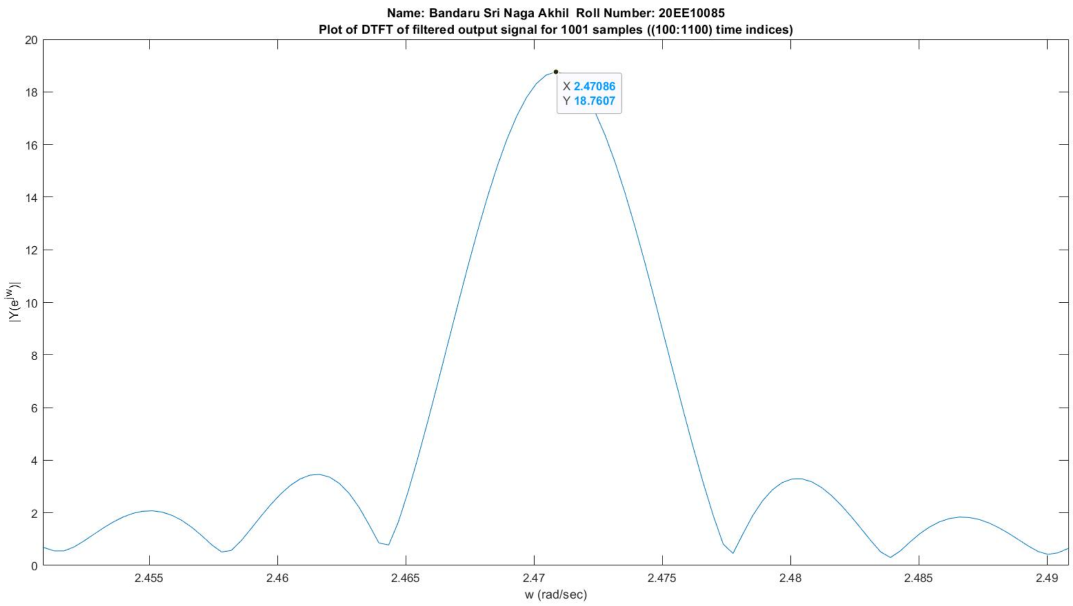
Matlab code for plotting the magnitude of DTFT of 1001 samples of the filtered output signal (using the time indices (100:1100)):

```
load pcm;      % importing pcm file into MATLAB
index = 100:1:1100; % defining a index interval from 100 to 1100

f1 = 3146; % modulated frequency
f2 = 8000; % sampled frequency
theta1 = (f1/f2)*2*pi; % calculation of frequency at which peak occurs
r1 = 0.995; % given r value

fs = IIRFilter(pcm,theta1,r1); % filtered output signal

% Computing the DTFT of filtered output signal over the above range
[Y,W] = DTFT(fs(index),0);

% plotting the magnitude of the DTFT of 1001 samples of filtered output signal
% (for the time indices (100:1100))
plot(W, abs(Y));
title({'Name: Bandaru Sri Naga Akhil  Roll Number: 20EE10085';
'Plot of DTFT of filtered output signal for 1001 samples ((100:1100) time indices)'});
xlabel('w (rad/sec)');
ylabel('|Y(e^j^w)|');
```

Plot of magnitude of DTFT of 1001 samples of the filtered output signal (using the time indices (100:1100)):



Plot of magnitude of DTFT of 1001 samples of the filtered output signal for the values of ω in the range of $[\theta - 0.02, \theta + 0.02]$:



Link for the audio signals - pcm1 (input signal) and pcm2 (filtered output signal):

pcm1 (input signal) and pcm2 (output signal)

Discussion:

From the magnitude response plot of the DTFT of the input signal, we can observe that, the input signal consists of a modulated sinusoid signal and background noice.

The peaks in this plot corresponds to the central frequency of the modulated sinusoid and the rest of the frequency contents correspond to the background noice.

Hence, inorder to remove the background noice, we design a FIR filter with r value close to 1 but less than 1 (for the system to be stable) and also keep its poles at the frequencies corresponding to the sinusoidal component, so that they amplify these frequencies with higher magnitude compared to other frequencies, resulting in noice elimination.

Given that, the pcm signal is modulated at 3146 Hz and sampled at 8k Hz. Hence, the value of θ at which the peak occurs in the DTFT plot of the input signal is

$$\theta = 2\pi \times \frac{3146}{8000} = 2.47086 rad/sec$$

Hence, we keep the poles at $re^{j\theta} = 0.995e^{j2.47086}$ and $re^{-j\theta} = 0.995e^{-j2.47086}$ to amplify these frequencies.

From the magnitude response plot of the DTFT of the output signal, we can observe that, the background noice gets eliminated by using this filter and the DTFT contains only the frequencies corresponding to the modulated signal. This background noice elimination can be observed from the audio of the output signal as well.

As the value of r becomes closer to 1, we expect that the background noise gets eliminated to a greater extent. But as the value of r becomes very close to 1 (consider r = 0.9999999), the output signal gets attenuated. This is because such large value of r filters out some of the frequencies of the modulated sinusoid as well including the background noice, resulting in attenuated output signal. In this case, we do not hear any audio from the output signal. So, it is ill-advised to use a filter with such a value of r.

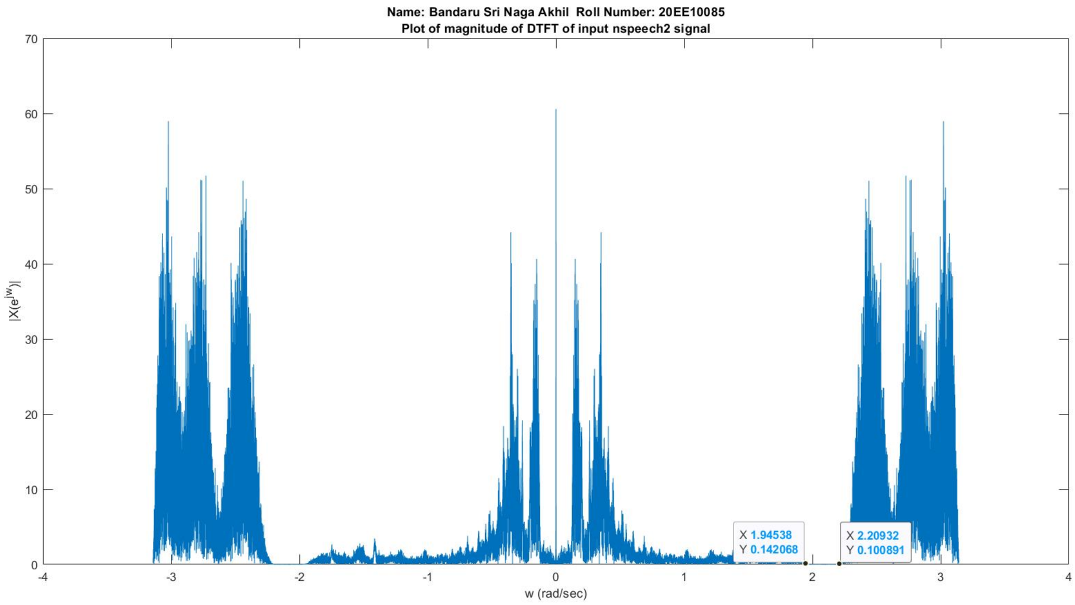
3 Lowpass Filter Design Parameters:

Download the file nspeech2.mat and load it into the Matlab workspace. It contains the signal nspeech2. Play the nspeech2 using the sound command and note the quality of the speech and background noise.

Matlab code for plotting the magnitude of DTFT for input nspeech2 signal:

```
load nspeech2.mat  % importing nspeech2.mat file into MATLAB
sound(nspeech2);   % Playing the audio of nspeech2 signal
audiowrite('inputnspeech2.ogg',nspeech2,8000); % saving the audio into .ogg format

% Computing the DTFT of input nspeech2 signal
[X,W] = DTFT(nspeech2,0);

% Plotting the DTFT of input nspeech2 signal
plot(W, abs(X));
title({'Name: Bandaru Sri Naga Akhil  Roll Number: 20EE10085';
    'Plot of magnitude of DTFT of input nspeech2 signal'});
ylabel('|X(e^j^w)|');
xlabel('w (rad/sec)');
```

Plot of magnitude of DTFT for input nspeech2 signal:



Discussion:

From the plot, we can observe that, there are two main components in nspeech2 signal: one at low frequencies and the other at high frequencies.

The high frequency signal is noice which is band limited to $|\omega| > 2.2$. The low frequency signal is the speech signal which is bandlimited to $|\omega| < 1.8$

3.1 Filter Design Using Truncation

Write a Matlab function LPFtrunc(N) that computes the truncated and shifted impulse response of size N for a low pass filter with a cutoff frequency of $\omega_c = 2.0$ using the equation

$$h(n) = \begin{cases} \frac{\omega_c}{\pi} \operatorname{sinc}\left(\frac{\omega_c}{\pi}\left(n - \frac{N-1}{2}\right)\right) & n = 0, 1, \ldots, N-1 \\ 0 & \text{otherwise} \end{cases}$$

For each of the following filter sizes, plot the magnitude of the filter's DTFT in decibels. (The magnitude of the response in decibels is given by $|H_{dB}(e^{j\omega})| = 20\log_{10}|H(e^{j\omega})|$.)

- $N = 21$
- $N = 101$

Submit the plots of the magnitude response for the two filters (not in decibels). On each of the plots, mark the passband, the transition band and the stopband.

Submit the plots of the magnitude response in decibels for the two filters.

Explain how the filter size effects the stopband ripple. Why does it have this effect?

Matlab code for implementing LPFtrunc function:

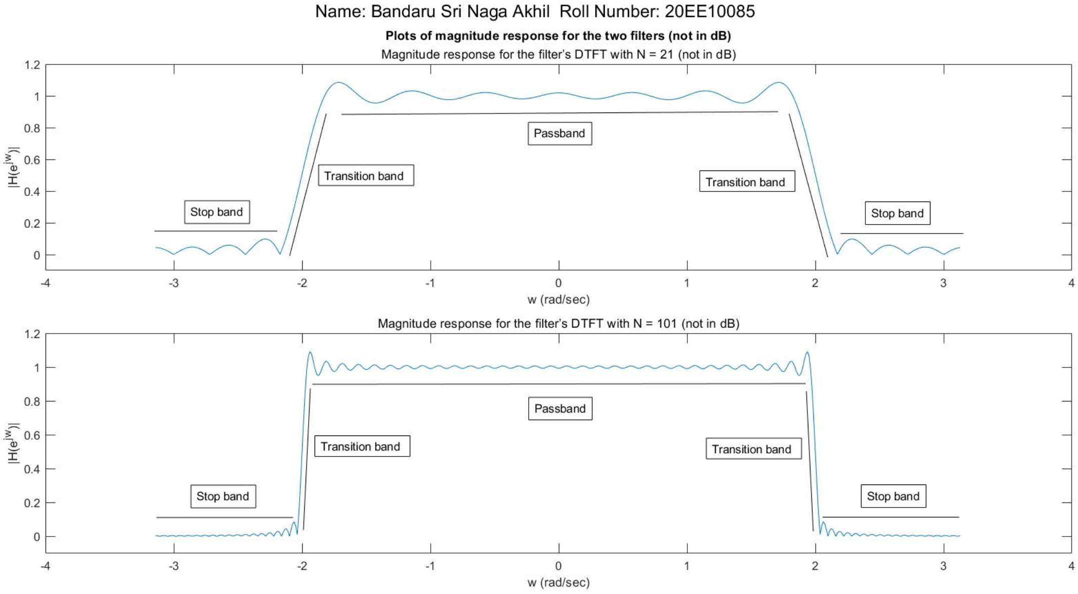
```
% Implementing LPFtrunc function
function h = LPFtrunc(N)
wc = 2.0;      % cutoff frquency of the low pass filter
% truncated impulse response
% h[n] = (wc/pi) sinc((wc/pi)(n - (N-1)/2)) for n = 0,1,2..,N-1
% h[n] = 0 otherwise
n = 0:1:N-1;
h = (wc/pi) * sinc((wc/pi) * (n - (N-1)/2));
end
```

Matlab code for plotting the magnitude response of the filter's DTFT for different values of N (N = 21 and N = 101):

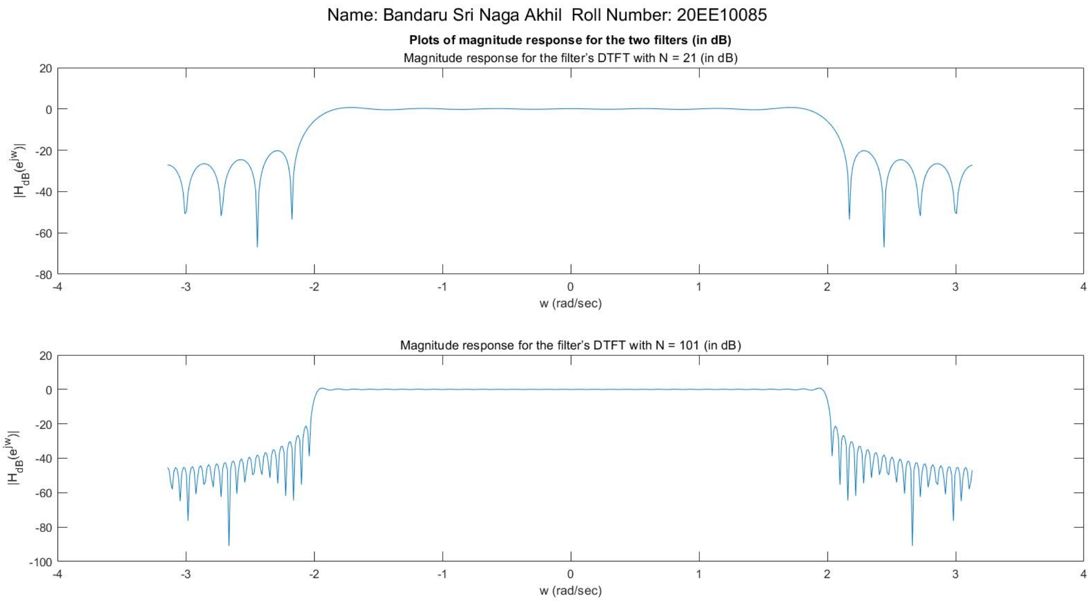
```
N1 = 21;  % size of rectangular window N = 21 for filter 1
h1 = LPFtrunc(N1);  % impulse response of the truncated lowpass filter
% Computing DTFT of impulse response of truncated lowpass filter
[X11,w1] = DTFT(h,512);
% Magnitude of DTFT of impulse response of the truncated lowpass filter in
% dB
X1 = 20*log10(abs(X11));
```

```
N2 = 101;  % size of rectangular window N = 101 for filter 2
h2 = LPFtrunc(N2);  % impulse response of the truncated lowpass filter
% Computing DTFT of impulse response of truncated lowpass filter
[X12,w2] = DTFT(h2,512);
% Magnitude of DTFT of impulse response of the truncated lowpass filter in
% dB
X2 = 20*log10(abs(X12));

% Plots of magnitude response for the two filters N = 21 and N = 101
% (not in dB)
figure (1)
subplot(211)
plot(w1,abs(X11));
sgtitle('Name: Bandaru Sri Naga Akhil  Roll Number: 20EE10085');
title('Plots of magnitude response for the two filters (not in dB)');
subtitle('Magnitude response for the filter DTFT with N = 21(not in dB)');
ylabel('|H(e^j^w)|');
xlabel('w (rad/sec)');
axis([-4 4 -0.1 1.2]);


subplot(212)
plot(w2,abs(X12));
subtitle('Magnitude response for the filter DTFT with N = 101(not in dB)');
ylabel('|H(e^j^w)|');
xlabel('w (rad/sec)');
axis([-4 4 -0.1 1.2]);


% Plots of magnitude response for the two filters N = 21 and N = 101 in dB
figure (2)
subplot(211)
plot(w1,X1);
sgtitle('Name: Bandaru Sri Naga Akhil  Roll Number: 20EE10085');
title('Plots of magnitude response for the two filters (in dB)');
subtitle('Magnitude response for the filter DTFT with N = 21 (in dB)');
ylabel('|H_d_B(e^j^w)|');
xlabel('w (rad/sec)');
axis([-4 4 -80 20]);

subplot(212)
plot(w2,X2);
subtitle('Magnitude response for the filter DTFT with N = 101 (in dB)');
ylabel('|H_d_B(e^j^w)|');
xlabel('w (rad/sec)');
axis([-4 4 -100 20]);
```

Plots of magnitude response of the filter's DTFT for different values of N (N = 21 and N = 101) (not in dB):



Plots of magnitude response (in dB) of the filter's DTFT for different values of N (N = 21 and N = 101):



Discussion:

Ideally, a low-pass filter with cutoff frequency ω_c should have a frequency response of

$$H_{\text{ideal}}\left(e^{j\omega}\right)=\begin{cases}1 & |\omega|\leq\omega_c\\ 0 & \omega_c<|\omega|\leq\pi\end{cases}$$

and a corresponding impulse response of

$$h_{\text{ideal}}(n)=\frac{\omega_c}{\pi}\operatorname{sinc}\left(\frac{\omega_c n}{\pi}\right)\text{ for }-\infty<n<\infty$$

Since,no real filter can have a frequency response of infinite length duration. Hence, we approximate the impulse response by truncating it. This results, in the oscillatory behaviour of the magnitude response near the cutoff frequency and large amount of ripple in the stop band.

Inorder to make the the filter causal, we shift it to the right and hence, we get the impulse response of the truncated signal as

$$h(n)=\begin{cases}\frac{\omega_c}{\pi}\operatorname{sinc}\left(\frac{\omega_c}{\pi}\left(n-\frac{N-1}{2}\right)\right) & n=0,1,\ldots,N-1\\ 0 & \text{otherwise}\end{cases}$$

From the modulation property of the DTFT, the frequency response of the truncated signal can be obtained by convolving the magnitude response of ideal low pass filter (which is a rect signal) with the DTFT of the truncating window.

Due to large sidelobes in the DTFT of truncating window, we get large stop band ripple in the final filter's magnitude response. These sidelobes can be minimized by considering a larger truncating window, which minimizes the stop band ripple in the filter's magnitude response.

From the plots, we can observe that, as the value of N increase, the filter's response has small ripple in the passband, high attenuation in the stopband and very narrow transition band leading to a closer approximation to an ideal low pass filter.

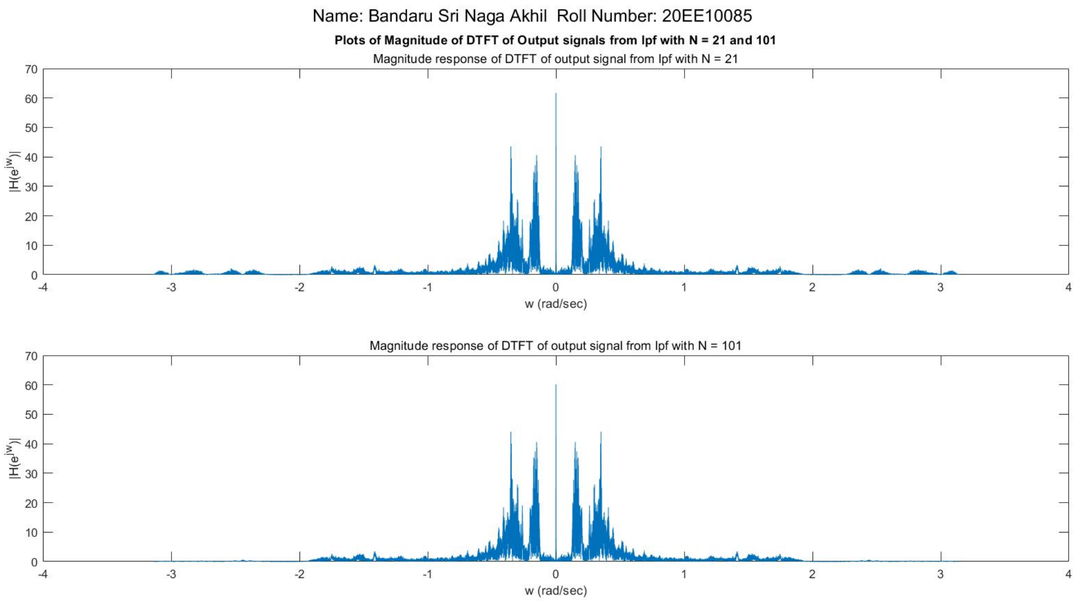
Download the noisy speech signal nspeech2.mat, and load it into the Matlab workspace. Apply the two filters with the sizes N =21 and N =101 to this signal. Since these are FIR filters, simply convolve them with the audio signal.

Comment on the quality of the filtered signals. Does the filter size have a noticeable effect on the audio quality? (In order to hear the filtered signals better, you may want to multiply each of them by 2 or 3 before using sound.)

Matlab code for plotting the magnitude of DTFT for the output signals from the low pass filter with sizes N = 21 and N =101:

```
load nspeech2.mat  % importing nspeech2.mat file into MATLAB

N1 = 21;  % size of rectangular window N = 21 for filter 1
h1 = LPFtrunc(N1);  % impulse response of the truncated lowpass filter

N2 = 101;  % size of rectangular window N = 101 for filter 2
h2 = LPFtrunc(N2);  % impulse response of the truncated lowpass filter

y1 = conv(nspeech2,h1); % output of truncated low pass filter (N=21)
y2 = conv(nspeech2,h2); % output of truncated low pass filter (N=1011)

y11 = 3*y1;   % multipliying the signal with 3 to hear the sounds better
y12 = 3*y2;   % multipliying the signal with 3 to hear the sounds better

sound(y11);   % Playing the audio of nspeech2 signal
audiowrite('outputN=21.ogg',y11,8000); % saving the audio into .ogg format

sound(y12);   % Playing the audio of nspeech2 signal
audiowrite('outputN=101.ogg',y12,8000); % saving the audio into .ogg format

[X1,w1] = DTFT(y1,512); % DTFT of output from low pass filter with N =21
[X2,w2] = DTFT(y2,512); % DTFT of output from low pass filter with N =101

subplot(211)
plot(w1,abs(X1));
sgtitle('Name: Bandaru Sri Naga Akhil  Roll Number: 20EE10085');
title('Plots of Magnitude of DTFT of Output signals from lpf with N = 21 and 101');
subtitle('Magnitude response of DTFT of output signal from lpf with N = 21');
ylabel('|H(e^j^w)|');
xlabel('w (rad/sec)');

subplot(212)
plot(w2,abs(X2));
subtitle('Magnitude response of DTFT of output signal from lpf with N = 101');
ylabel('|H(e^j^w)|');
xlabel('w (rad/sec)');
```

Plots of magnitude of DTFT for the output signals from the low pass filter with sizes N = 21 and N =101:



Link for the audio signals - inputnspeech2 (input signal) and outputN=21, outputN=101 (output signals from the low pass filters with sizes N = 21 and N = 101 respectively):

inputnspeech2 (input signal) and outputN=21, outputN=101 (output signals)

Discussion:

From the output audio signals obtained from the low pass filters, we can conclude that as the value of N increases, the quality of the audio of the filtered signals increases.

This is because, as the value of N increases, the truncating window length increases, which decreases the sidelobes obtained in its DTFT and hence decreasing the ripples in the stopband. Therefore, it better attenuates the frequencies in the stop band leading to a higher quality in the filtered output signals.

From the plots of DTFT of the output signals, we can observe that, for N = 21, there are some frequency components corresponding to background noice. But in the second case, i.e.; for N = 101, these frequency components are mostly attenuated.
Hence, we can conclude that the filter size have a noticeable effect on the audio quality of the filtered signals,i.e.; the more the size of N, the better the quality of the filtered signals.

*****