Department of Electrical Engineering
Indian Institute of Technology Kharagpur

Digital Signal Processing Laboratory (EE39203)

Autumn, 2022-23

Experiment 5 **Digital Filter Design**

Slot: **X** Date: **13/10/2023**

Student Name: **P. Manoj Kumar** Roll No.: **21IE10027**

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)

P Manoj Kumar (21IE10027)

Experiment 5 - Digital Filter Design

1 Design of a Simple FIR Filter

1.1 Impulse Response and Difference Equation of the Filter

The given transfer function is:

$$H_f(z) = 1 - 2\cos\theta z^{-1} + z^{-2}$$

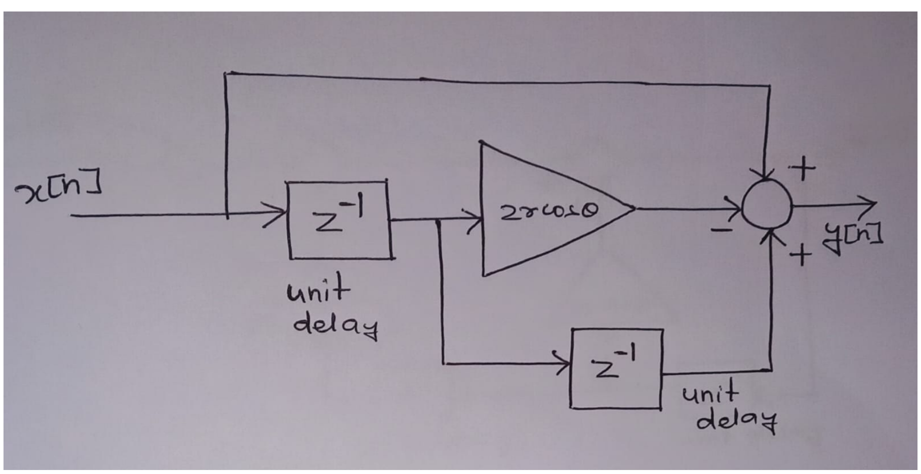
The impulse response of a system is the inverse Z-transform of its transfer function. So, taking inverse Z-transform on both sides of equation, The impulse response of the filter is given by:

$$h[n] = \delta[n] + 2\cos\theta\delta[n-1] + \delta[n-2]$$

In order to draw the block diagram of the system, first, we find the difference equation of the system from its impulse response. The output $y[n]$ of the system can be obtained by convolving the input $x[n]$ with its impulse response $h[n]$:

$$\begin{aligned} y[n] &= x[n] * h[n] \\ &= x[n] * (\delta[n] + 2\cos\theta\delta[n-1] + \delta[n-2]) \\ &= x[n] * \delta[n] + 2\cos\theta(x[n] * \delta[n-1]) + x[n] * \delta[n-2] \\ &= x[n] + 2\cos\theta x[n-1] + x[n-2] \end{aligned}$$

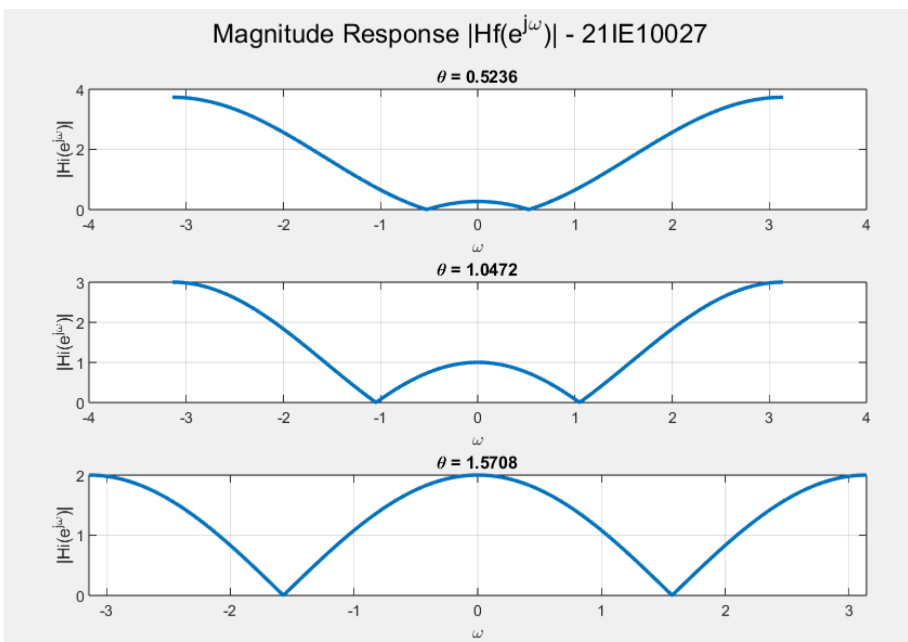
1.2 Block Diagram



1.3 Plot of Magnitude Response for Different Values of θ

The MATLAB code and plots for the magnitude response of $H_f(z)$ is given below for three different values of θ i.e. $\frac{\pi}{6}$, $\frac{\pi}{3}$, and $\frac{\pi}{2}$.

```
theta_values = [pi/6, pi/3, pi/2];% Define the values of theta
omega = linspace(-pi, pi, 1000);  % Define the frequency range
% Initialize an array to store the magnitude responses
magnitude_responses = zeros(length(theta_values), length(omega));
% Compute the magnitude responses for each theta
for i = 1:length(theta_values)
    theta = theta_values(i);
    Hf = abs(1 - 2*cos(theta)*exp(-1i*omega) + exp(-2*1i*omega));
    magnitude_responses(i, :) = Hf;
end
% Plot the magnitude responses
figure;
for i = 1:length(theta_values)
    subplot(length(theta_values),1, i);
    plot(omega, magnitude_responses(i, :), 'LineWidth', 2);
    title(['\theta = ', num2str(theta_values(i))]);
    xlabel('\omega');
    ylabel('|Hi(e^{j\omega})|');
    grid on;
end
sgtitle('Magnitude Response |Hf(e^{j\omega})| - 21IE10027');
xlim([-pi pi]); % Set the x-axis limits to -pi and pi
```



1.4 Effect of θ on Filter's Frequency Response

The value of θ directly influences the locations of the zeros of the transfer function $H_f(z)$, which in turn affects the magnitude response of the filter in the frequency domain.

The transfer function $H_f(z)$ is given by:

$$H_f(z) = (1 - z^{-1})(1 - z^{-1}) = 1 - 2\cos(\theta)z^{-1} + z^{-2}$$

This can be rewritten in terms of $z = e^{j\omega}$ as:

$$H_f(e^{j\omega}) = 1 - 2\cos(\theta)e^{-j\omega} + e^{-2j\omega}$$

The magnitude response $|H_f(e^{j\omega})|$ can be calculated as:

$$|H_f(e^{j\omega})| = \sqrt{1 + 4\cos^2(\theta) - 4\cos(\theta)\cos(\omega) + 4\cos^2(\omega)}$$

Here's how θ affects the magnitude response:

1. **Small θ (e.g., $\theta = \pi/6$):**
   - This corresponds to a filter with zeros close to the unit circle.
   - The magnitude response will exhibit narrower notches in the frequency domain.
   - It will be effective at attenuating frequencies close to θ and $-\theta$.
2. **Medium θ (e.g., $\theta = \pi/3$):**
   - This will lead to zeros that are moderately spaced from the unit circle.
   - The magnitude response will have wider notches compared to small θ.
   - It will provide moderate attenuation around θ and $-\theta$.
3. **Large θ (e.g., $\theta = \pi/2$):**
   - This corresponds to zeros far away from the unit circle.
   - The magnitude response will have even wider notches.
   - It will provide significant attenuation around θ and $-\theta$.

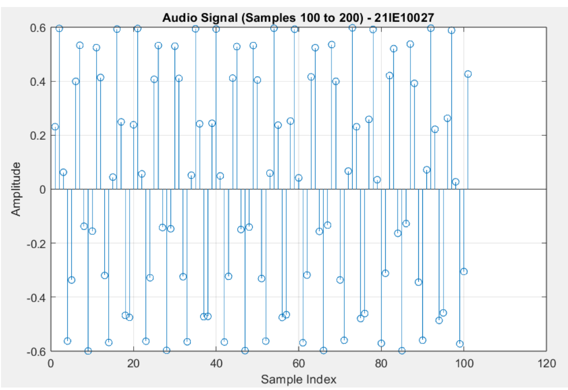
In summary, increasing θ spreads out the zeros of the transfer function, resulting in broader notches in the frequency response. This means that larger θ values lead to greater attenuation of frequencies around θ and $-\theta$.

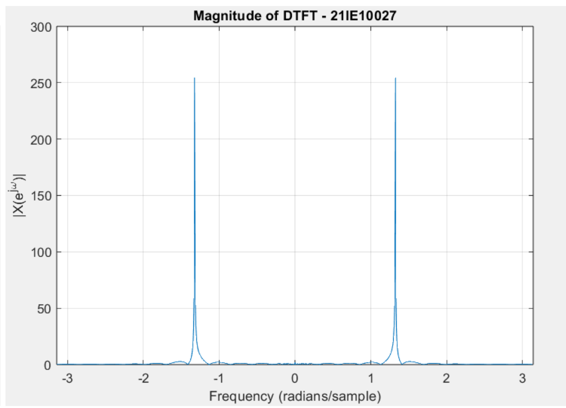
2 Filtering Audio Signal Using FIR Filter

2.1 Plot of 101 samples and DTFT of the original audio

The MATLAB code for generating the time domain plot of 101 samples of the original audio between the indices 100 to 200, as well as the plot for 1001 samples of the magnitude of the DTFT of the original audio between the indices 100 to 1100, is provided below.

```
% Load the audio signal "nspeech1"
load nspeech1;
% Play the loaded audio signal
sound(nspeech1);
% Plot the amplitude of samples 100 to 200 of the audio signal
figure;
stem(nspeech1(100:200));
title('Audio Signal (Samples 100 to 200) - 21IE10027');
xlabel('Sample Index');
ylabel('Amplitude');
grid on;
% Compute and plot the DTFT of the first 1001 samples
[X, w] = DTFT(nspeech1(1:1001), 0);
figure;
plot(w, abs(X));
title('Magnitude of DTFT - 21IE10027');
xlabel('Frequency (radians/sample)');
ylabel('|X(e^{j\omega})|');
xlim([-pi, pi]);
grid on;
```

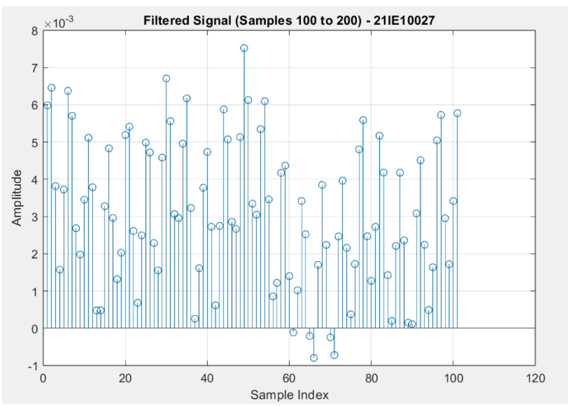


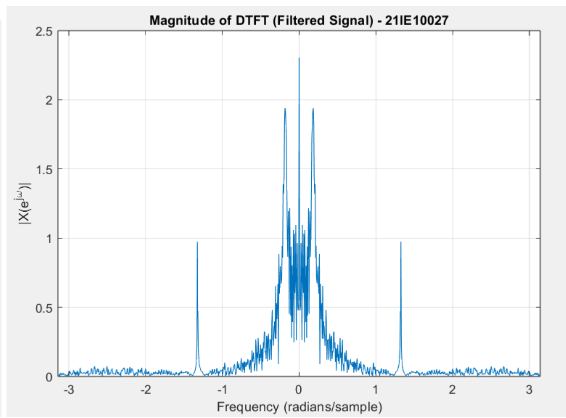


2.2 Plot of 101 samples and DTFT of the filtered audio

The MATLAB code for generating the time domain plot of 101 samples of the filtered audio between the indices 100 to 200, as well as the plot for 1001 samples of the magnitude of the DTFT of the filtered audio between the indices 100 to 1100, is provided below.

```matlab
% Find the maximum value and corresponding index in the DTFT
[Xmax, Imax] = max(abs(X));
theta = w(Imax);
% Apply a FIR filter with the selected frequency theta to the audio signal
filtered_signal = FIRfilter(nspeech1, theta);
% Play the filtered audio signal (amplified by 20)
sound(20*filtered_signal);
% Plot the amplitude of samples 100 to 200 of the filtered signal
figure;
stem(filtered_signal(100:200));
title('Filtered Signal (Samples 100 to 200) - 21IE10027');
xlabel('Sample Index');
ylabel('Amplitude');
grid on;
% Compute and plot the DTFT of first 1001 samples of the filtered signal
[X_filtered, w_filtered] = DTFT(filtered_signal(1:1001), 0);
figure;
plot(w_filtered, abs(X_filtered));
title('Magnitude of DTFT (Filtered Signal) - 21IE10027');
xlabel('Frequency (radians/sample)');
ylabel('|X(e^{j\omega})|');
xlim([-pi, pi]);
grid on;
```





2.3 MATLAB Code for FIR Filter

```matlab
function y = FIRfilter(x, theta)
    % Define the filter coefficients
    a = 1;
    b = [1 -2*cos(theta) 1];
    h = impz(b,a,50);
    % Apply the filter
    y = conv(x, h, 'same');
end
```

2.4 Discussion

- The peak of the DTFT of `nspeech1` is observed to occur at approximately $\theta = 1.32536$.
- A FIR filter should be designed in such a way that its magnitude response dips to zero at $\omega = \pm 1.32536$.
- When the speech is passed through this filter, it is noted that the peaks in the original audio's DTFT (that originally introduced the beep) are significantly attenuated.
- The filtering process effectively reduces the intensity of the beep to a significant extent.
- As a result, the sentence becomes more intelligible, with the voice now clearly saying, "Please get rid of this beep."

3 Design of a Simple IIR Filter

3.1 Impulse Response and Difference Equation of the Filter

Consider the given transfer function:

$$H_i(z) = \frac{1-r}{1-2r\cos\theta z^{-1} + r^2 z^{-2}} \quad (6)$$

Substituting $\frac{Y(z)}{X(z)}$ for $H(z)$ in the above equation, we get:

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

or

$$\left(1-2r\cos\theta z^{-1} + r^2 z^{-2}\right) Y(z) = (1-r)X(z) \quad (7)$$

Taking the inverse Z-transform on both sides of the above equation, we get:

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

Therefore, the required difference equation is:

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

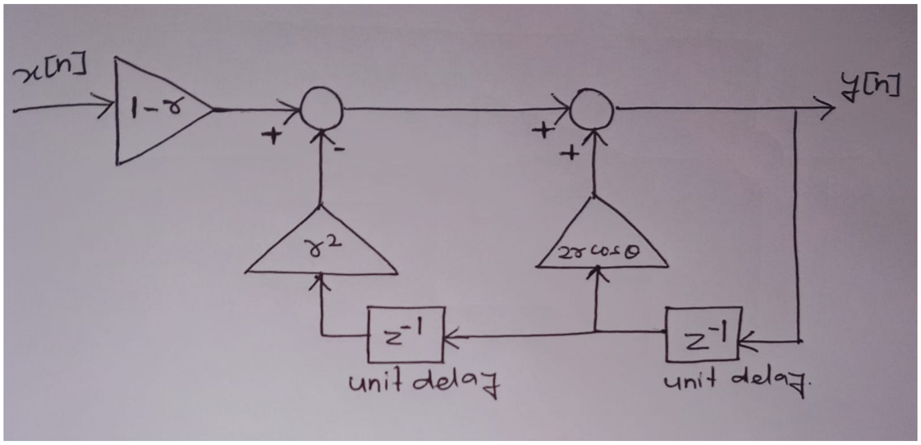
The impulse response of the system can be obtained by taking the inverse Z-transform of equation 6. That is:

$$\begin{aligned}
h[n] &= \mathcal{Z}^{-1}\left\{\frac{1-r}{1-2r\cos\theta z^{-1} + r^2 z^{-2}}\right\} \\
&= \mathcal{Z}^{-1}\left\{\frac{1-r}{1-e^{-2j\theta}\cdot\frac{1}{1-re^{j\theta}z^{-1}} + 1 - re^{2j\theta}\cdot\frac{1}{1-re^{-j\theta}z^{-1}}}\right\} \\
&= (1-r)r^n\left\{\frac{e^{j\theta(n+1)} - e^{-j\theta(n+1)}}{2j\sin\theta}\right\}u[n]
\end{aligned}$$

Therefore, the required impulse response is:

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

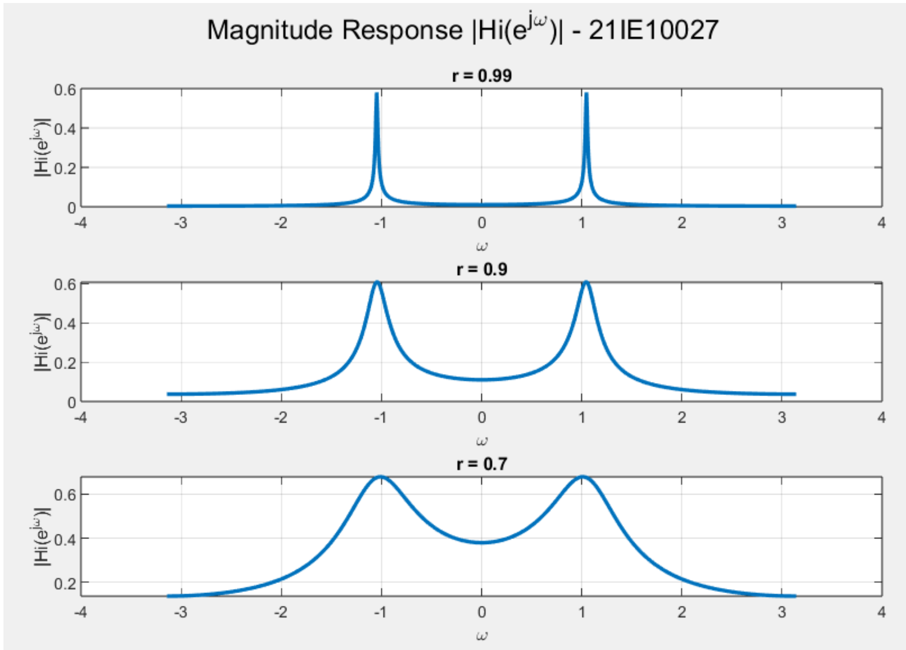
3.2 Block Diagram



3.3 Plot of Magnitude Response for Different Values of r

The MATLAB code and plots for the magnitude of the filter's frequency response $|H_i(e^{j\omega})|$ on $|\omega| < \pi$ for three values of r i.e. 0.99, 0.9, and 0.7 when $\theta = \frac{\pi}{3}$

```
r_values = [0.99, 0.9, 0.7]; % Define the values of r
theta = pi/3; % Value of theta
omega = linspace(-pi, pi, 1000);% Define the frequency range
% Initialize an array to store the magnitude responses
magnitude_responses = zeros(length(r_values), length(omega));
% Calculate magnitude responses for each value of r
for i = 1:length(r_values)
    r = r_values(i);
    Hi = abs((1-r)./(1-2*r*cos(theta)*exp(-1j*omega)+r^2*exp(-2j*omega)));
    magnitude_responses(i, :) = Hi;
end
% Plot the magnitude responses
figure;
for i = 1:length(r_values)
    subplot(length(r_values),1, i);
    plot(omega, magnitude_responses(i, :), 'LineWidth', 2);
    title(['r = ', num2str(r_values(i))]);
    xlabel('\omega');
    ylabel('|Hi(e^{j\omega})|');
    grid on;
end
sgtitle('Magnitude Response |Hi(e^{j\omega})| - 21IE10027');
```



3.4 Effect of r on Filter's Frequency Response

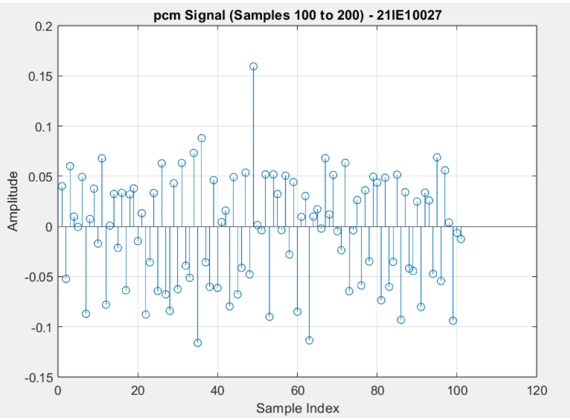
- **As r increases (moving closer to 1)**:
  - **Magnitude of Peaks Increases**: As r increases, the magnitude of peaks in the magnitude response also increases. This can be observed in the plots, where higher values of r lead to taller peaks.
  - **Narrowing of Frequency Range**: The frequency range around θ where the magnitude is significant becomes narrower. In other words, the filter becomes more selective in amplifying or attenuating frequencies around θ.
- **As r decreases (moving closer to 0)**:
  - **Magnitude of Peaks Increases**: On the contrary, as r decreases, the magnitude of peaks in the magnitude response decreases. This is evident in the plots, where lower values of r result in shorter peaks.
  - **Widening of Frequency Range**: The frequency range around θ where the magnitude is significant becomes wider. The filter becomes less selective and affects a broader range of frequencies around θ.
- **Impact of Pole Location**: By locating the poles of $H_i(z)$ close to the unit circle, the filter's bandwidth may be made extremely narrow around θ. This also makes sense mathematically, since when ω approaches θ, the magnitude of the frequency response increases. So, the closer the pole is to the unit circle, the more pronounced the peak.

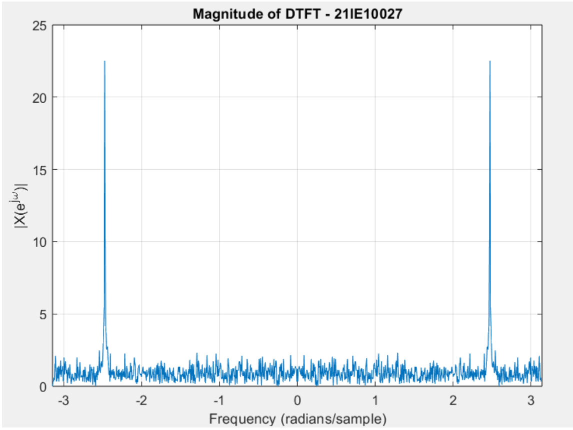
4 Filtering Audio Signal Using IIR Filter

4.1 Plot of 101 samples and DTFT of the original audio

The MATLAB code for generating the time domain plot of 101 samples of the original audio between the indices 100 to 200, as well as the plot for 1001 samples of the magnitude of the DTFT of the original audio between the indices 100 to 1100, is provided below.

```
% Load the signal named "pcm"
load pcm;
% Play the loaded signal
sound(pcm);
% Plot the amplitude of samples 100 to 200 of the signal
figure;
stem(pcm(100:200));
title('pcm Signal (Samples 100 to 200) - 21IE10027');
xlabel('Sample Index');
ylabel('Amplitude');
grid on;
% Compute and plot the DTFT of samples 100 to 1100
[X, w] = DTFT(pcm(100:1100), 0);
figure;
plot(w, abs(X));
title('Magnitude of DTFT - 21IE10027');
xlabel('Frequency (radians/sample)');
ylabel('|X(e^{j\omega})|');
xlim([-pi, pi]);
grid on;
```

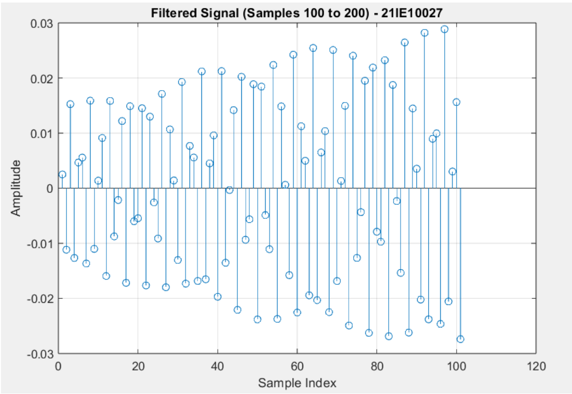


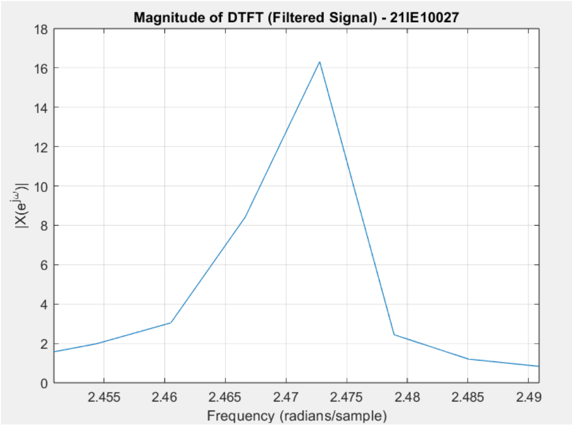


4.2 Plot of 101 samples and DTFT of the filtered audio for ω in [θ - 0.02, θ + 0.02]

The MATLAB code for generating the time domain plot of 101 samples of the filtered audio between the indices 100 to 200, as well as the plot for 1001 samples of the magnitude of the DTFT of the filtered audio between the indices 100 to 1100, is provided below.

```
% Define a specific frequency theta
theta = 3146 * 2 * pi / 8000;
% Define a parameter for an IIR filter
r = 0.995;
% Apply an IIR filter to the signal with specified parameters
filtered_signal = IIRfilter(pcm, r, theta);
% Play the filtered signal (amplified by 30)
sound(30*filtered_signal);
% Plot the amplitude of samples 100 to 200 of the filtered signal
figure;
stem(filtered_signal(100:200));
title('Filtered Signal (Samples 100 to 200) - 21IE10027');
xlabel('Sample Index');
ylabel('Amplitude');
grid on;
% Compute and plot the DTFT of samples 100 to 1100 of the filtered signal
[X_filtered, w_filtered] = DTFT(filtered_signal(100:1100), 0);
figure;
plot(w_filtered, abs(X_filtered));
title('Magnitude of DTFT (Filtered Signal) - 21IE10027');
xlabel('Frequency (radians/sample)');
ylabel('|X(e^{j\omega})|');
xlim([theta - 0.02, theta + 0.02]);
grid on;
```





4.3 MATLAB Code for IIR Filter

```
function y = IIRfilter(x, r, theta)
    % Initialize an array 'y' with zeros of the same size as 'x'
    y = zeros(size(x));
    for n = 3:length(x)
        % IIR filter equation
        y(n) = (1 - r)* x(n) + 2*r*cos(theta)*y(n-1) - r^2*y(n-2);
    end
end
```

5 Low Pass Filter Design Using Truncation

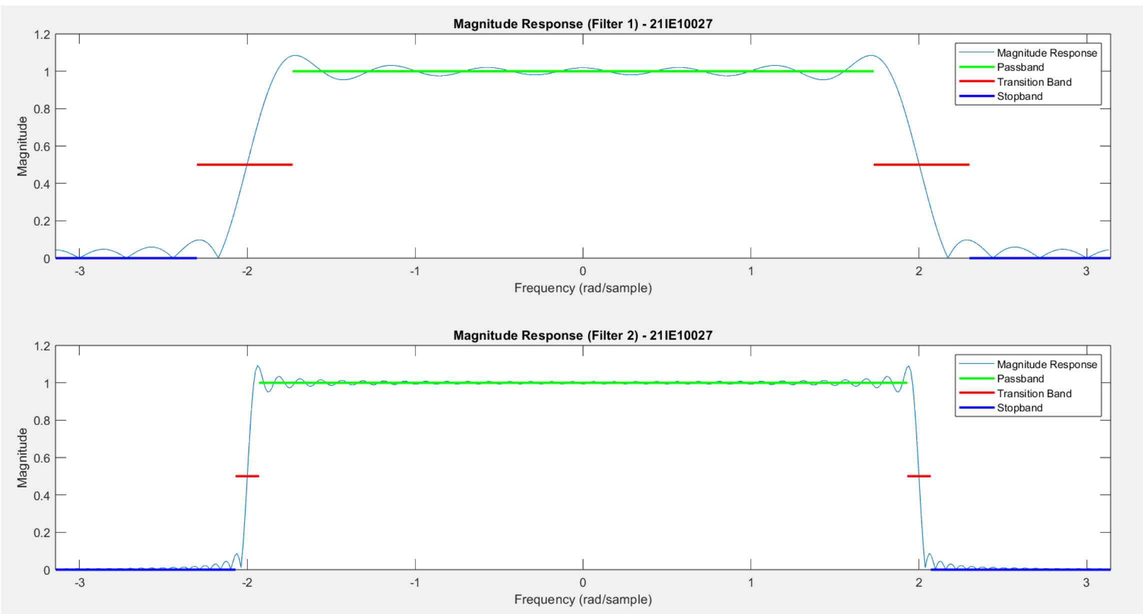
5.1 MATLAB Code for LPFTrunc

The function LPFtrunc(N) calculates the truncated and shifted impulse response with a size of N for a low pass filter featuring a cutoff frequency of ω_c= 2.0. The MATLAB code for the function is given below.

```
function h = LPFtrunc(N)
    % N: Length of the filter
    % M: Middle index of the filter
    M = (N-1)/2;
    % wc: Cutoff frequency
    wc = 2;
    % n: Discrete time index from 0 to N-1
    n = 0:N-1;
    % Calculate the impulse response using the sinc function
    h = (wc/pi) * sinc((wc/pi)*(n-M));
end
```

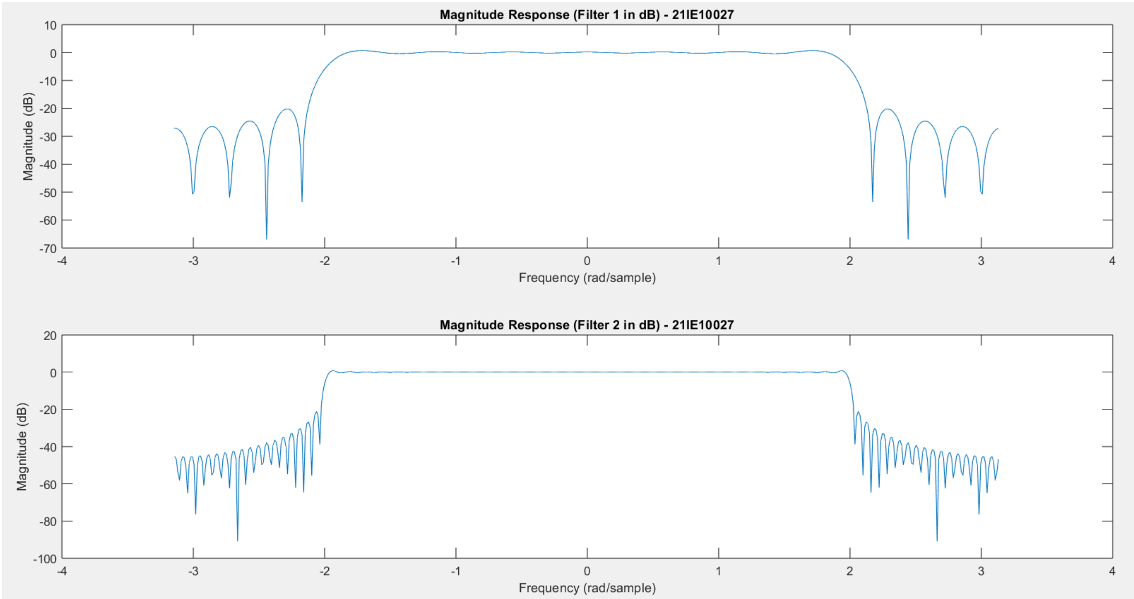
5.2 Magnitude response of the filter (not in decibels)

The magnitude response of the filter is plotted for two different filter orders: N = 21 and N = 101. For both cases, the pass-band, stop-band, and transition band characteristics are displayed below.



5.3 Magnitude response of the filter (in decibels)

The magnitude response of the filter is plotted for two different filter orders: N = 21 and N = 101, in decibels.



5.4 Discussion

- **Effect of Filter Size on Stopband Ripple**
  - Increasing filter size reduces stopband ripple.
  - Larger filter size leads to a truncated filter that approximates the ideal low-pass filter better.
  - However, ripple frequency increases.
- **Changes in Ripple Frequency and Transition Band**
  - Ripple frequency increases with larger filter size.
  - Transition band decreases with larger filter size.
  - Reduced ripple amplitude in the stop-band leads to less noise in the filtered signal.
  - Smaller transition band results in sharper rejection of frequencies beyond the cutoff.
- **Factors Influencing Filtered Signal Quality**
  - Other factors have minimal contribution to the quality of the filtered signal.
- **Effect of Filter Length on Audio Clarity**
  - Longer filters improve audio clarity.
  - Noise introduced at corners decreases with longer filters.
  - Observations:
    - N = 21: Small audible noise at normal volume, but speech is much clearer.
    - N = 101: No audible noise at normal volume.