Group 4

- Mirza Ananta Hermawan (2206822212)
- Farizi Yufli Alrantisi (2206821595)
- Arya Fadhlurrahman Mulia (2206822010)

Type Filter : **Stop Band**

Type Window : **Bartlett**

Filter Characteristics:

1. Cut Off Frequency: 2000 Hz and 3100 Hz
2. Jumlah sinyal impulse yang digunakan, M=150

Tulis persamaan frekuensi response filter yang ideal, $H_d(e^{j\Omega})$:

Ideal Band-Stop frequency response

$$|H[e^{j\omega}]| = \begin{cases} 1, & 0 \leq |\omega| \leq \omega_{c1} \\ 0, & \omega_{c1} \leq |\omega| \leq \omega_{c2} \\ 1, & \omega_{c2} \leq |\omega| \leq \pi \end{cases}$$

Tulis persamaan impulse response filter yang ideal, $H_d[n]$:

Ideal impulse response filter

$$h[n] = \frac{1}{2\pi}\left[\int_{-\omega_{c1}}^{\omega_{c1}} e^{j\omega n}\, d\omega + \int_{\omega_{c2}}^{\pi} e^{j\omega n}\, d\omega + \int_{-\pi}^{-\omega_{c2}} e^{j\omega n}\, d\omega\right]$$

$$= \begin{cases} 1 + \dfrac{\omega_{c1} - \omega_{c2}}{\pi}, & \text{for } n = 0 \\ \dfrac{1}{\pi n}\left[\sin[\omega_{c1} n] - \sin[\omega_{c2} n]\right] & \text{for } n \neq 0 \end{cases}$$

Tulis persamaan metode window yang Anda gunakan, $w[n]$:

Bartlet windowing method

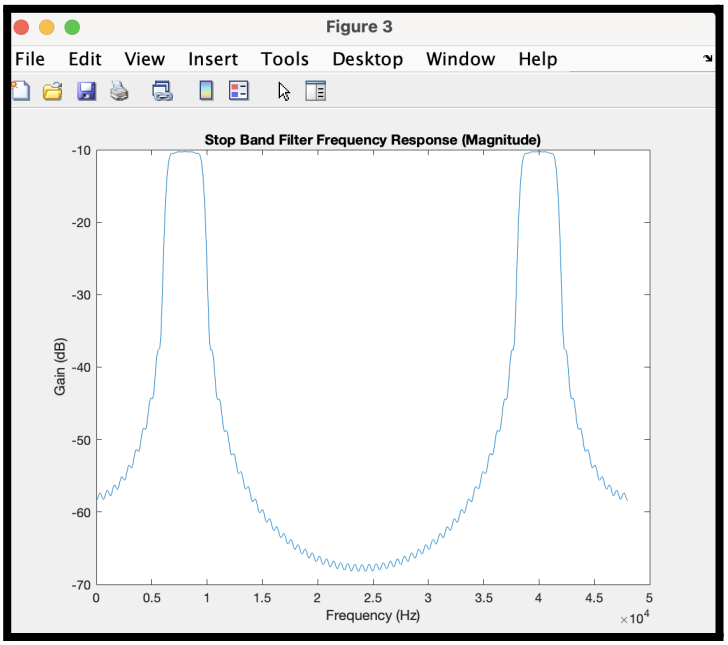
$$W[n] = 1 - \left| \frac{n - \frac{N}{2}}{\frac{L}{2}} \right|, \quad 0 \leq n \leq N$$

Tulis persamaan impulse response filter yang Anda rancang, $H[n]$:

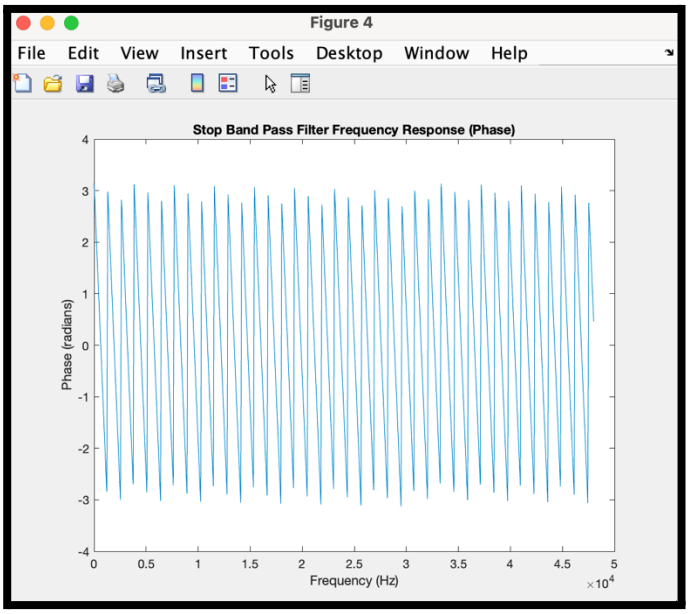
Band-stop impulse response filter

$$H[n] = \frac{2 f_1 \sin[n\omega_1]}{n\omega_1} - \frac{2 f_2 \sin[n\omega_2]}{n\omega_2}$$

Plot Magnitude dari $H[n]$:



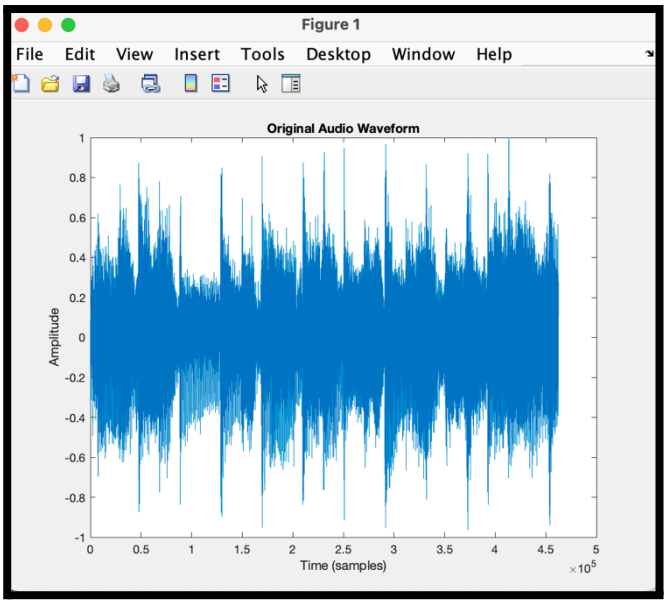
Plot Phase dari $H[n]$:



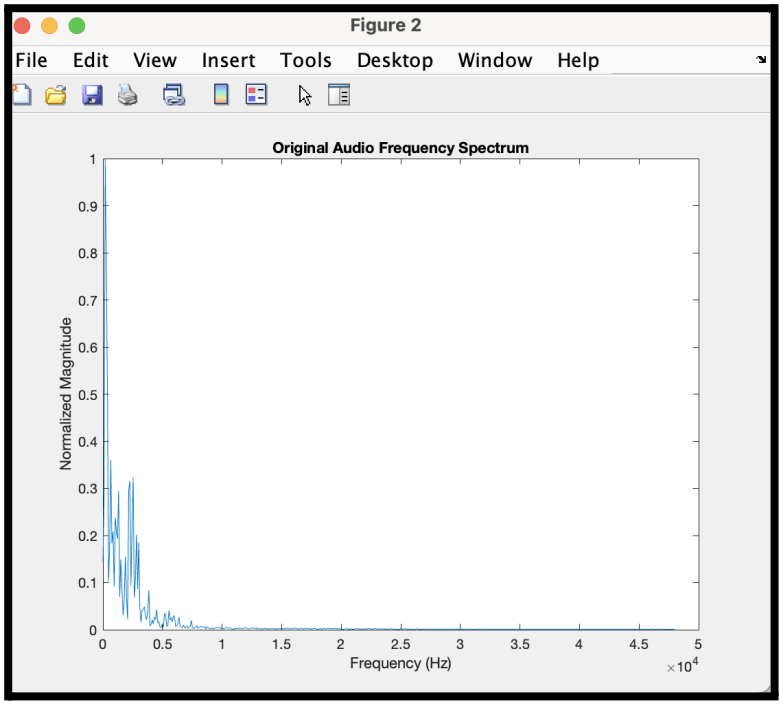
Carilah file audio dengan format *.wav dari internet, dan copy-paste kan link download nya di sini:

https://drive.google.com/file/d/18z97Q2j4Uxj7Q_zOGJxGOAwt6bpGagpR/view?usp=sharing

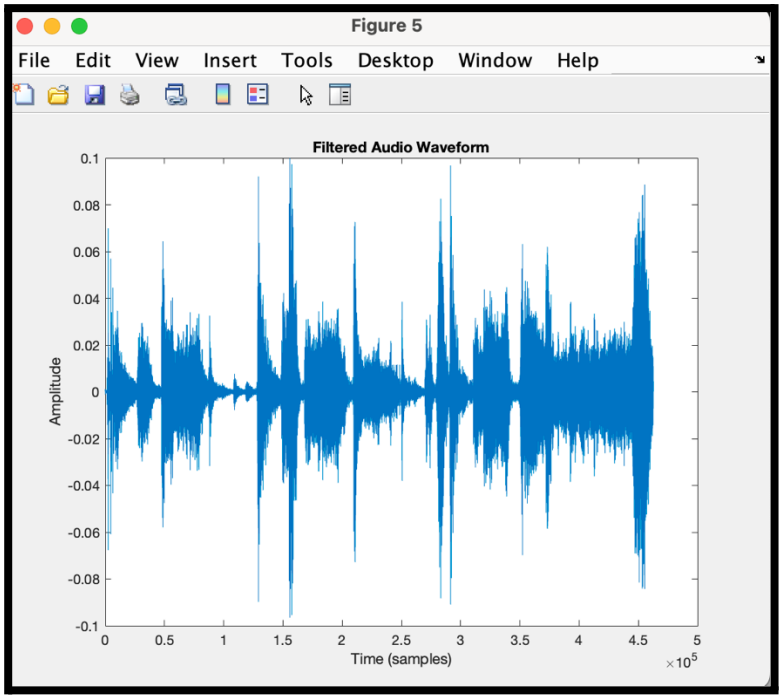
Plot audio dalam domain waktu:



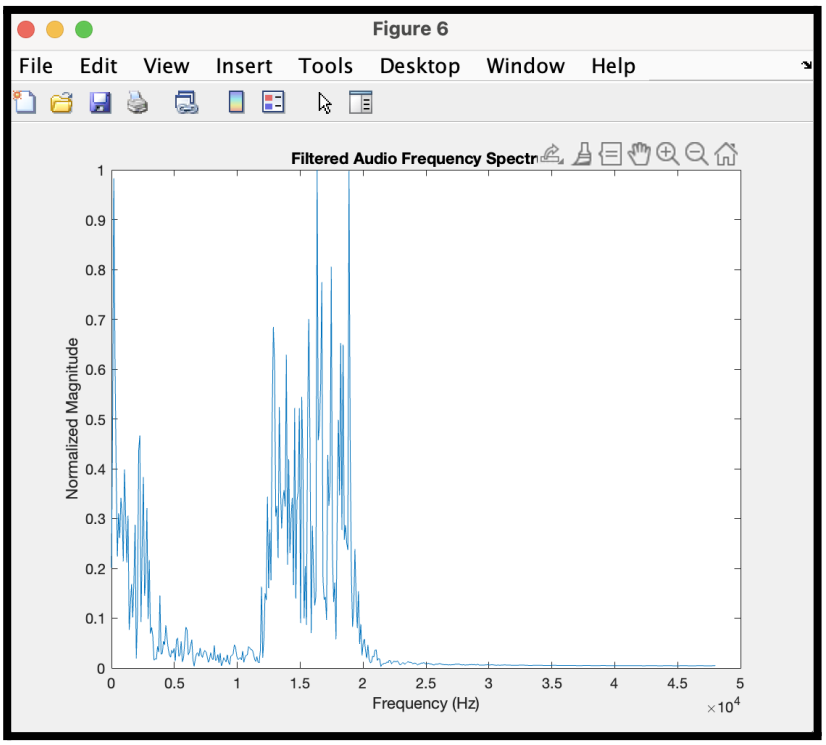
Plot audio dalam domain frequency:



Plot audio yang sudah terfilter dalam domain waktu



Plot audio yang sudah terfilter dalam domain frekuensi



Berikan analisis Anda:

Brief Analysis

In this project we designed an audio digital filter using finite impulse response. We created a stop band filter with Bartlett windowing method, the cut off frequencies for this filter are 2000 Hz and 3100 Hz, and we also used M=150 for the amount of impulse signal used. In this matlab code we can import an audio with .wav format to undergo the stop band filter. As the output of this matlab digital filter we plotted 6 figures to illustrate how the filter works. In this project we used paint my love from MLTR to demonstrate the audio digital filter.

Analysis of the figures (or graphs)

1. Figure 1: original audio waveform

   This figure shows the original audio waveform in the time domain. The Y axis represents the amplitude, while the X axis represents the audio signal over time

2. Figure 2: original Audio frequency spectrum

   This figure shows the original audio's frequency spectrum after performing FFT or fast fourier transform. This plot depicts the frequency of the original audio signal. The X axis represents frequency in Hertz, while the Y axis represents the normalized magnitude.

3. Figure 3: stop band fFilter frequency response (magnitude)

This figure shows the magnitude response of the designed band stop filter. It illustrates how the filter affects the magnitude or gain of different frequencies in the frequency domain. This figure specifically shows the attenuation of signal magnitude in the stop band or the frequencies between fc1 and fc2 of the filter.

4. Figure 4: stop band pass filter frequency response (phase)

   This shows the phase shift by the filter across different frequencies. The X axis represents frequency in Hertz, and the Y axis shows the phase in radians. This figure indicates how the filter delays or advances different frequencies within its passband or stopband.

5. Figure 5: filtered audio waveform

   This figure shows the waveform of the audio signal after it has been filtered using the band stop filter. This figure represents how the filtered audio signal's amplitude changes over time.

6. Figure 6: filtered audio frequency spectrum

   This figure represents the frequency content of the filtered audio signal. This figure displays the distribution of frequencies present in the filtered audio signal after the application of the band stop filter.

Bartlett Windowing Method Analysis

Bartlett windowing is a method that is used in signal processing to modify the signal on the output by making the start and the end is 0 gradually and showing the maximum of the signal in the middle of each band. The purpose of the Bartlett window is to make the signal distortion as minimal as possible and to reduce the discontinued signal and there can be spectral leakage so we can do the analysis. Bartlett window has various advantages. The first one is bartlett window will have better frequency resolution, the second one is bartlett window will reduce spectral leakage, and lastly its more simple to compute and more easier for the implementation. Meanwhile, bartlett window also have its own disadvantage and the important note while doing bartlett windowing is when it reducing the spectral leakage it might be not reducing entirely, but it’s all depend on the application that we’re going to use. And also the important disadvantage is bartlett window increasing the main lobe width in the frequency domain. Overall, bartlett windowing is not the only method to design digital filter and we might need a strong consideration to choose which windowing method that we need to use. However, by using bartlett windowing method it will offer the benefit to digital filter design. In this case, bartlett window might not be the right choice for stop band pass, we sure that it’s more suitable for low pass filter and its because the transition will be more smooth.

Matlab Filter Code Analysis

1. Load Audio:

Extract sample data and sample rate, and convert the audio to mono.

```
[sampleAudio, sampleRate] = audioread('MLTRPaintMyLove.wav');
sampleAudio = sampleAudio(:, 1);  % Convert the audio file into mono
```

2. Analyze Original Audio:

Plot the original audio waveform and see how the amplitude changes over time. After that run the audio through a frequency spectrum plotting program to observe how the frequencies are distributed.

```
% Plot the original audio in the time domain (waveform)
figure(1);
plot(sampleAudio);
title('Original Audio Waveform');
xlabel('Time (samples)');
ylabel('Amplitude');

% Plot the original audio in the frequency domain after performing FFT
N = 1024; % FFT size
frequency = 0:sampleRate/(N/2 - 1):sampleRate;
sampleAudio_fft = fft(sampleAudio, N);
figure(2);
plot(frequency, abs(sampleAudio_fft(1:N/2)) / max(abs(sampleAudio_fft(1:N/2))));
title('Original Audio Frequency Spectrum');
xlabel('Frequency (Hz)');
ylabel('Normalized Magnitude');
```

3. Design Stopband Filter:

Set the filter length, cutoff frequencies, and window type. Then we calculate the ideal impulse response with sinc functions and normalized cutoff frequencies.

Apply the Bartlett window to smoothen the ideal response and obtain final filter coefficients. Then, calculate the filter's frequency response using FFT.

```
% Design a Stop Band Pass filter with cutoff frequencies
fc1 = 2000 Hz and fc2 = 3100 Hz, using M=150 and Bartlett windowing
M = 150; % Filter length
fc1 = 2000; % Lower cutoff frequency
```

```
fc2 = 3100; % Upper cutoff frequency
FC1 = fc1 / sampleRate; % Normalized lower cutoff frequency
FC2 = fc2 / sampleRate; % Normalized upper cutoff frequency
omega1 = 2 * pi * FC1;
omega2 = 2 * pi * FC2;
n = -75:1:75; % Filter coefficients range
hd = ((2 * FC1 * sinc(n * omega1)) - (2 * FC2 * sinc(n * omega2))); % Filter impulse response
window = 1 - ((2 * abs(n)) / M); % Bartlett window
hn = hd .* window; % Final filter coefficients
hn_fft = fft(hn, N); % Compute FFT of the filter
```

4. Filter the Audio:

Apply the intended filter using convolution to change the frequency from the original audio.

```
% Plot the filter frequency response (magnitude)
figure(3);
% Calculate the frequency response of the filter
filter_response = freqz(hn, 1, frequency, sampleRate);
plot(frequency, 20 * log10(abs(filter_response)));
title('Stop Band Filter Frequency Response (Magnitude)');
xlabel('Frequency (Hz)');
ylabel('Gain (dB)');

% Calculate the phase response of the filter
phase_response = angle(hn_fft(1:1:N/2));

% Plot the filter frequency response (phase)
figure(4);
plot(frequency, phase_response);
title('Stop Band Pass Filter Frequency Response (Phase)');
xlabel('Frequency (Hz)');
ylabel('Phase (radians)');



% Apply the filter to the audio using convolution
sampleAudio_filtered = conv(sampleAudio, hn);
```

```
% Plot the filtered audio in the time domain (waveform)
figure(5);
plot(sampleAudio_filtered);
title('Filtered Audio Waveform');
xlabel('Time (samples)');
ylabel('Amplitude');

% Plot the filtered audio in the frequency domain after performing FFT
sampleAudio_fft_filtered = fft(sampleAudio_filtered, N);
figure(6);
plot(frequency, abs(sampleAudio_fft_filtered(1:1:N/2)) /
max(abs(sampleAudio_fft_filtered(1:1:N/2))));
title('Filtered Audio Frequency Spectrum');
xlabel('Frequency (Hz)');
ylabel('Normalized Magnitude');
```