Teachers Assessment Activity

DIGITAL SIGNAL PROCESSING ECT 354

5th Semester B.Tech. Session-2024-25

Project Report

Title:"Applying Band pass filter to audio file with variable cutoff frequency input by user"

Project Group Students:

1. **Ojas Surjuse(49, B)**
2. **Paritosh Patra(51, B)**
3. **Parth Bais (52, B)**
4. **Pranay Pawar (53, B)**
5. **Shreya Bangde (67,B)**

Course Coordinator

Prof. Vipul S. Lande

Assistant Professor

Electronics and Communication Engineering Department

Ramdeobaba College of Engineering & Management, Nagpur

Contents

1.Introduction:

In digital signal processing (DSP), filters play a crucial role in manipulating signals to enhance or suppress specific frequencies, allowing users to isolate or eliminate certain aspects of a sound. One of the most commonly used filters is the band-pass filter, which allows frequencies within a specified range to pass through while attenuating frequencies outside that range. The ability to dynamically control which frequency range passes through a filter is a powerful tool in audio engineering, communication systems, and various multimedia applications.

In this project, we aim to apply a band-pass filter to an audio file, with the cutoff frequencies defined by the user. The flexibility of variable cutoff frequencies allows for greater control and precision in audio manipulation. This capability is especially useful in fields like music production, noise reduction, and speech processing, where different frequency ranges are emphasized or suppressed to enhance the audio quality or remove unwanted noise.

The band-pass filter is an essential tool in audio processing, as sound waves typically consist of a wide range of frequencies. In some situations, it is necessary to isolate a specific frequency band, such as the range of human speech or certain musical instruments, while suppressing other frequencies that contribute to noise or interference. For example, low-frequency sounds like background hums or high-frequency hisses can be reduced while preserving the clarity of a vocal performance. In other cases, focusing on specific frequency bands can enhance the audio's texture and richness, making the sound more engaging and pleasant to the listener.

One of the key aspects of this project is allowing the user to input the desired cutoff frequencies for the band-pass filter. This user-driven control offers significant flexibility and adaptability to different scenarios. For instance, the user may want to emphasize frequencies between 300 Hz and 3000 Hz, the typical range of human speech, in a communication system to ensure clear transmission. Alternatively, the user may want to filter out certain frequencies in music production to make room for other instruments or effects. This approach allows the filter to be tailored to the specific requirements of the task, making it more versatile than a filter with fixed cutoff frequencies.

MATLAB provides a powerful environment for implementing digital filters and processing audio signals. With its extensive set of built-in functions for filter design, users can quickly create filters that meet their specific requirements. For this project, we use a Butterworth band-pass filter. The Butterworth filter is chosen because of its flat frequency response in the passband, which ensures that the desired frequencies are passed through without significant distortion. The user inputs the low and high cutoff frequencies, which define the range of frequencies that the filter will allow. MATLAB then applies the filter to the audio signal, and the filtered signal can be analyzed or played back.

In addition to applying the filter, the project also involves visualizing the audio signal and the filter's frequency response. Visualization is a crucial step in understanding how the filter affects the audio signal. By plotting the original and filtered signals, users can observe the changes in amplitude and how the filter has attenuated the undesired frequencies. The frequency response of the filter is also plotted to show the range of frequencies that the filter allows to pass through, providing a clear picture of the filter's behavior.

The band-pass filter designed in this project can be used in a variety of applications. In music production, for instance, it can be used to enhance the clarity of vocals by removing unnecessary low and high-frequency noise. In communication systems, the filter can isolate the frequency range of human speech, making it easier for systems to recognize and process speech in noisy environments. The filter can also be used in noise reduction applications, where specific frequencies that contribute to noise, such as electrical hums or static, are suppressed, resulting in cleaner audio. Additionally, the filter can be applied to speech processing systems to improve the accuracy of voice recognition algorithms, especially in environments with reverberation or background noise.

One of the most powerful aspects of this project is its adaptability. By allowing the user to define the cutoff frequencies, the filter can be tailored to the specific needs of the user. Whether the goal is to isolate a certain instrument in a piece of music or to filter out background noise in a voice recording, this project provides a versatile solution. The user-driven approach empowers the user to experiment with different frequency ranges and find the optimal settings for their specific use case.

2.Objectives:

a) Design of a Band-Pass Filter with Variable Cutoff Frequencies Input by User and applying it to an audio file

The goal of this project is to design and implement a band-pass filter that can be applied to an audio signal, where the user can input the lower cutoff frequency and upper cutoff frequency to define the passband. The project will involve reading an audio file, applying the filter, and analyzing the results in both time and frequency domains. The filter will be adaptive based on user-defined parameters, and the results will be played back and saved for further analysis. This will involve key signal processing steps using MATLAB.

b) Selective Frequency Isolation for Enhanced Audio Quality

The primary goal of applying a bandpass filter is to isolate and preserve only the desired range of frequencies (the passband) while attenuating the unwanted frequencies outside of this range, thus improving the audio signal's quality.Audio signals often contain a mixture of desired sounds (such as vocals or instruments) and unwanted noise or interference. A bandpass filter allows you to target a specific frequency band, preserving only the frequencies that contain the audio of interest and filtering out the noise or other unwanted components. By allowing the user to input variable cutoff frequencies, the system becomes flexible, enabling them to fine-tune the filter to match the characteristics of the audio file. For example, in a music recording, the user can retain the midrange frequencies where vocals are typically present, while attenuating low-frequency noise or high-frequency hiss.

c) Customization for Adaptive Noise Reduction

Providing variable cutoff frequency inputs empowers the user to adapt the filter to different types of noise environments, achieving optimal noise reduction across different audio contexts.Audio recordings can suffer from various forms of noise that affect different frequency ranges. For instance, environmental noise like rumble and hum is often found in low frequencies, while high-pitched interference or hiss resides in higher frequencies. By allowing the user to specify both upper and lower cutoff frequencies, the filter can be customized to selectively attenuate the noise specific to the recording environment. This flexibility ensures that the filtered output preserves as much of the desired signal as possible while minimizing noise, making the audio clearer and more intelligible.

3.Theoretical Background:

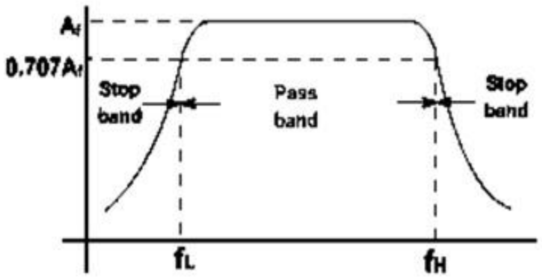
In digital signal processing (DSP), filters play an essential role in manipulating signals to achieve desired outcomes, such as enhancing certain frequencies, reducing noise, or isolating specific elements within an audio signal. A **band-pass filter (BPF)** is one such filter, used widely to allow frequencies within a specific range to pass through while attenuating frequencies outside that range. This functionality is pivotal in numerous applications, including audio processing, telecommunications, and biomedical engineering.

Overview of Band-Pass Filters

A band-pass filter is designed to pass frequencies within a designated range and attenuate frequencies outside that range. The filter is characterized by two key parameters:

- **Lower cutoff frequency (f_L):** The frequency below which all frequencies are attenuated.
- **Upper cutoff frequency (f_U):** The frequency above which all frequencies are attenuated.

The frequencies that lie between these two thresholds (f_L and f_U) are allowed to pass, forming the passband. Frequencies outside this range are part of the stopband, where attenuation is applied.
Mathematically, the bandwidth (BW) of the band-pass filter is defined as:
BW=f U-f L
Mathematically, the center frequency of the band-pass filter is defined as
CF=√(f U-f L)



Digital Implementation of Band-Pass Filters:

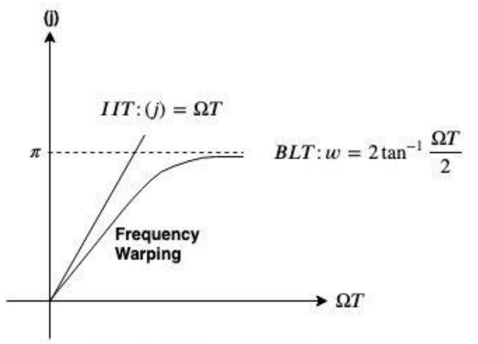
In the digital domain, band-pass filters are constructed by applying a combination of low-pass and high-pass filters. A low-pass filter attenuates frequencies higher than its cutoff frequency, while a high-pass filter attenuates frequencies lower than its cutoff frequency. By cascading a high-pass filter and a low-pass filter, the band-pass filter can isolate frequencies within the desired range.

For a band-pass filter, the filter's design equation is shaped by combining the equations for both high-pass and low-pass filters. The choice between FIR and IIR filters depends on the application, with FIR filters offering better phase linearity but requiring more computation, while IIR filters are computationally efficient but can introduce phase distortion. For our project we chose a butterworth filter.[5]

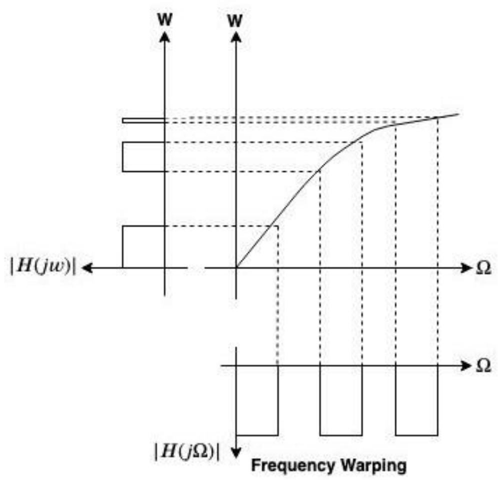
Bilinear transformation:

Bilinear transformation is a frequency domain method of converting the analog filter transfer function H(s) into a digital one H(z). The transformation is performed by a change of variables
s=(2/T)*(z−1)/(z+1)
It is called bilinear because both the numerator and denominator of this transformation equation are linear. This transformation is reversible in that H(s) can be obtained from H(z) by the substitution
z=((2/T)+s)/((2/T)−s)

WARPING EFFECT IN BILINEAR TRANSFORMATION:



Mapping between (j) & Ω in IIT & BLT



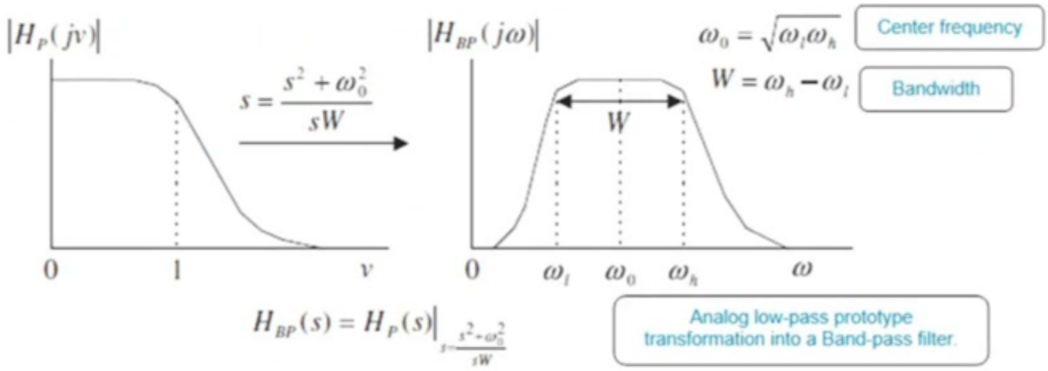
Warping effects happen because of tan function w=2*tan-1(Ω*T/2),
Prewarping needs to be done to eliminate warping effect u0=(2/T)*tan(ω0/2) where u0 is the analog frequency.[6]

Low pass to bandpass:

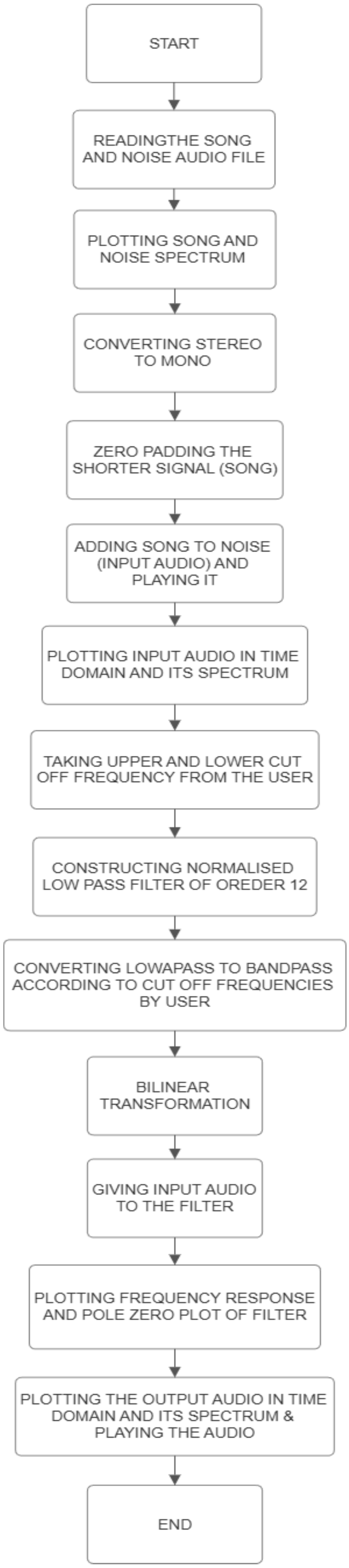
Low pass to bandpass can be transformed using frequency transformation in analog domain as:

s $\longrightarrow$ Ωc[(s^2+(Ω_0)^2) /(s*(BW))]

where Ωc is cut off frequency, Ω_0 is centre frequency, BW is bandwidth



4.Flow chart:


START
READINGTHE SONG AND NOISE AUDIO FILE
PLOTTING SONG AND NOISE SPECTRUM
CONVERTING STEREO TO MONO
ZERO PADDING THE SHORTER SIGNAL (SONG)
ADDING SONG TO NOISE (INPUT AUDIO) AND PLAYING IT
PLOTTING INPUT AUDIO IN TIME DOMAIN AND ITS SPECTRUM
TAKING UPPER AND LOWER CUT OFF FREQUENCY FROM THE USER
CONSTRUCTING NORMALISED LOW PASS FILTER OF OREDER 12
CONVERTING LOWAPASS TO BANDPASS ACCORDING TO CUT OFF FREQUENCIES BY USER
BILINEAR TRANSFORMATION
GIVING INPUT AUDIO TO THE FILTER
PLOTTING FREQUENCY RESPONSE AND POLE ZERO PLOT OF FILTER
PLOTTING THE OUTPUT AUDIO IN TIME DOMAIN AND ITS SPECTRUM & PLAYING THE AUDIO
END

5. Code:

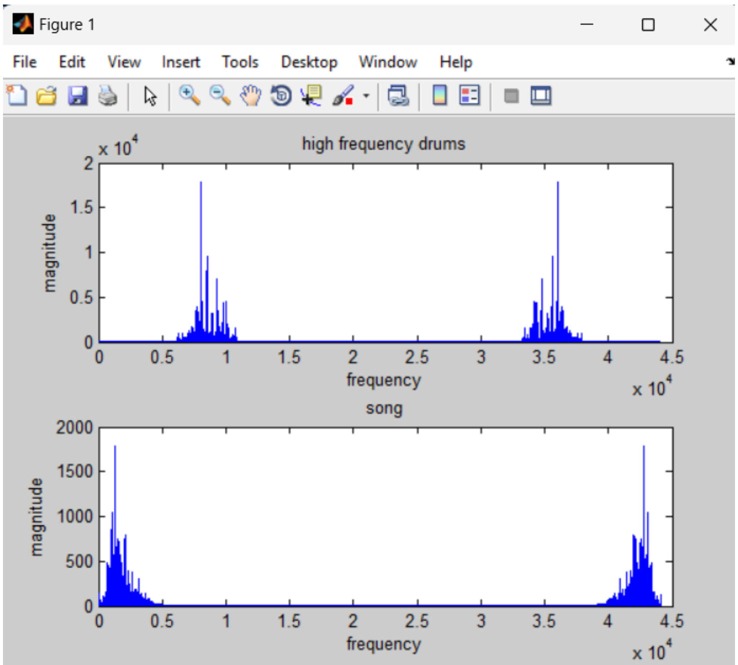
```
%DSP-TA:Applying Band pass filter to audio file with variable cutoff frequency input by user.
clc
clear all
close all
%loading the audio file
[song1,fs2]=audioread('C:\Users\Ojas
Surjuse\Downloads\d9-slumdog-millionaire-13-jai-ho-2775-[AudioTrimmer.com] (2) (1).mp3');
[noise1,fs1]=audioread('C:\Users\Ojas
Surjuse\Downloads\amen-break-no-copyright-remake-120bpm-25924 (1).mp3');
%converting stereo signal to mono signal
noise=noise1(:,1);
song=song1(:,1);
N2=length(noise);
N5=length(song);
fs=44100;
%zero padding of noise signal
song=[song; zeros((N2-N5),1)];
N3=length(song);
%adding noise to song
audio=(song+noise);
N=length(audio);
%frequency axis construction
f3=fs*(0:N2-1)/N2;
f4=fs*(0:N3-1)/N3;
%FFT of the signals
NO=fft(noise);
SONG=fft(song);
%plot of song spectrum and noise spectrum
figure,subplot(2,1,1),plot(f3,abs(NO)),title('high frequency
drums'),subplot(2,1,2),plot(f4,abs(SONG)),title('song');
ts=1/fs;%sampling time
t=0:ts:(length(audio)/fs)-ts;%time axis in seconds
f2=fs*(0:N-1)/N;
Aud=fft(audio);
```

```
%plot of input time domain and frequency domain
figure,subplot(2,1,1),plot(t,audio),title('time domain input signal(combined)'),xlabel('time in
seconds'),ylabel('amplitude')
subplot(2,1,2),plot(f2,abs(Aud)),title('input audio
spectrum'),xlabel('frequency'),ylabel('magnitude');
disp('playing input audio file...');
sound(audio,fs);%playing input audio
pause(length(audio)/fs);
disp('audio spectrum between 1k to 5k');
fcu=input(['enter the upper cut-off frequency:']);
fcl=input(['enter the lower cut-off frequency:']);
n1=12;
disp('order is 12');
ohmfcu=2*fs*tan(2*pi*fcu/(2*fs)); %prewarping technique
ohmfcl=2*fs*tan(2*pi*fcl/(2*fs));
bw=ohmfcu-ohmfcl;%bandwidth calculation
cf=sqrt(ohmfcu*ohmfcl);%center frequency calculation
[z1, p1, k1]=buttap(n1/2);%normalised low pass filter zeros,poles,gains
[num1,den1] = zp2tf(z1,p1,k1);%transfer function
[B1, A1]=lp2bp(num1,den1,cf,bw);%scale the lpf to bpf
[bz1, az1]= bilinear(B1,A1,fs);%bilinear transformation
[h1, f1]= freqz(bz1,az1,22.05*1000,fs);%frequency response
% Magnitude Spectrum of filter
figure,subplot(2,1,1)
plot(f1,mag2db(abs(h1))),title('magnitude response of
filter'),xlabel('frequency'),ylabel('Attenuation(db)')
subplot(2,1,2)
% phase Spectrum of filter
plot(f1,angle(h1)),title('phase response of filter'),xlabel('frequency'),ylabel('phase angle')
y1 = filter(bz1,az1,audio);
Y1=fft(y1);
%plot of output time domain and frequency domain
figure,subplot(2,1,1),plot(t,y1),title('time domain output signal'),xlabel('time in
seconds'),ylabel('amplitude')
subplot(2,1,2),plot(f2,abs(Y1)),title('output audio
spectrum'),xlabel('frequency'),ylabel('magnitude');
%plot of pole-zero plot of normalised lpf and bpf
figure,subplot(2,1,1)
```

```
zplane(num1,den1),title(['Normalised low pass filt of order: ' num2str(n1/2)])
subplot(2,1,2)
zplane(bz1,az1),title(['bandpass filter of order: ' num2str(n1)]);
disp('playing filtered audio...');
sound(y1,fs);%playing output audio
pause(length(audio)/fs);
```

6. Results:

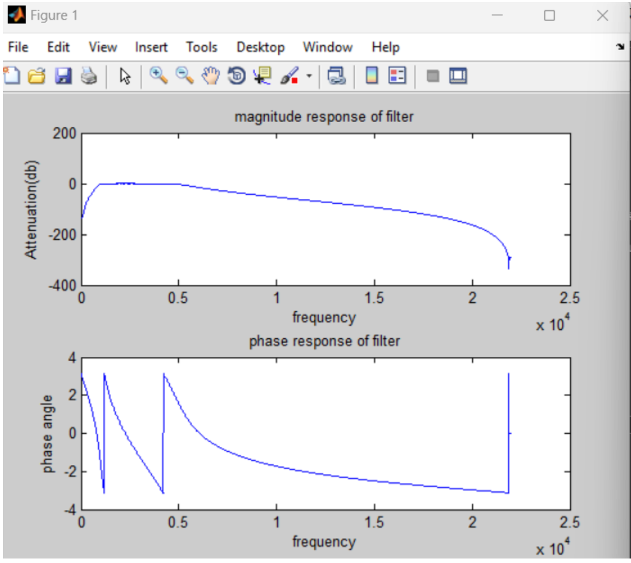
In this MATLAB code, a bandpass filter is applied to an audio signal, where the cutoff frequencies are input by the user. The main goal is to isolate a specific frequency range (between the lower cutoff fcl and upper cutoff fcu from the input audio, which is a mixture of a song and noise. The sampling frequency of the mixture is 44.1KHz and the duration of the signal is 18 seconds. We first plot spectrum of noise and song (high frequency drums) to understand the bandwidth of both the signal.

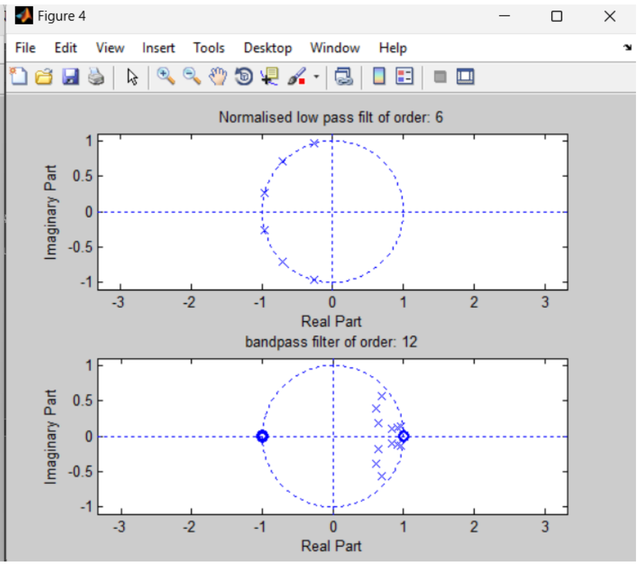


Plot of song and noise spectrum separately

From the above plot we conclude that the song lies between 1khz to 5khz and the high frequency drums lies from 8khz to 11khz.The mixture is then passed through the band pass filter with order 12 and fh=5khz and fl=1khz provided by the user.

Filter Plots:

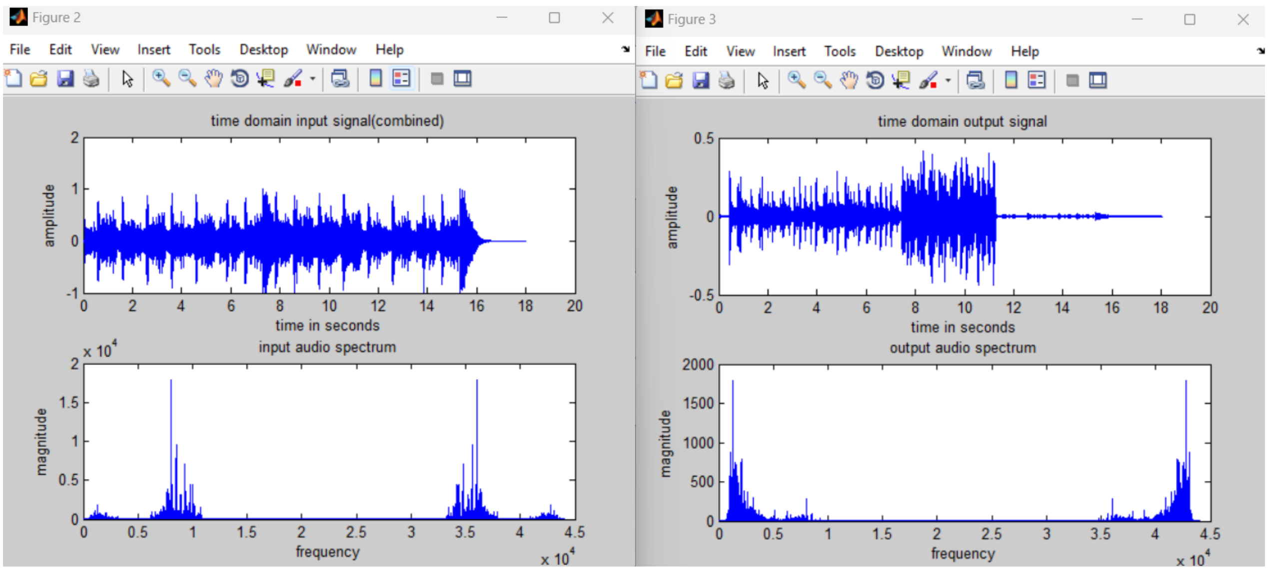




Frequency response of bandpass filter

Pole-zero plot of low pass and bandpass filter

As from the above frequency response of butterworth bandpass filter we conclude that butterworth has flat response in pass band also it gives non-linear phase response because it being IIR. From the pole-zero plot we see the difference between low pass filter of order 6 and scaled up band pass filter of order 12 made from low pass. The difference being the normalized low-pass scales to achieve the cut off frequency, then its characteristics are reversed to form high pass filter which are cascaded together doubling the order.



Plot of time domain and spectrum of input and output

From the output we conclude that high frequency drums which are in stopband are suppressed and the song which is in passband 1khz to 5khz is passed removing the unwanted noise and isolating the selective frequency for enhanced audio.

7. Applications:

1.Noise reduction in audio signals

Bandpass filters remove unwanted noise from audio recordings by attenuating frequencies outside the desired range. This improves clarity in speech or music by reducing background noise like low-frequency hum or high-frequency hiss .Filter helps clean up audio recordings, eliminating background noise and enhancing the main signal, whether it's vocals or instruments.[8]

2.Communication system

Bandpass filters in communication systems isolate signals within a specific frequency range, such as in radio receivers, selecting the desired signal while attenuating adjacent channel interference. Relevance: Your filter can be adjusted to pass only the desired communication frequencies, filtering out noise and interference outside the passband.[9]

3.Biomedical Signal Processing (e.g., ECG or EEG Filtering)

In medical devices, bandpass filters remove noise from biomedical signals like ECG (Electrocardiogram) or EEG (Electroencephalogram) data, focusing on critical physiological frequencies while filtering out high-frequency noise or interference. Bandpass filter could be applied to similar tasks, removing unwanted frequency components and preserving important signal features.[10]

4.Hearing Aids and Audio Enhancements

Bandpass filters are commonly used in hearing aids to improve speech clarity by amplifying speech frequencies and filtering out background noise. They focus on specific frequencies where human speech occurs while reducing unwanted sounds, making it easier for users to hear and understand conversations in noisy environments. Similarly, your filter can be applied to hearing aids to enhance important sounds like speech and reduce distracting noise, improving overall audio quality for the user.

8. Limitations and Future scope:

Limitations

The filter's effectiveness depends on accurate user inputs for cut-off frequencies, and a fixed 12th-order filter may not suit all signals. Its computational complexity could cause delays, making real-time applications challenging on less powerful hardware. Noise outside the passband is assumed, so overlapping noise may affect performance. The command-line interface isn't user-friendly, and manual filter parameter adjustment limits versatility. Additionally, ringing artifacts or distortion can occur near the cut-off frequencies due to the fixed order. Delays may arise based on the system's processing speed and the length of the audio file.

Future Scope

To enhance the filter project, real-time processing can be implemented using DSP hardware or real-time software frameworks, allowing for live audio filtering. Adaptive filtering can automatically adjust parameters based on incoming audio, improving performance in different noise environments. A user-friendly GUI will simplify input for non-technical users, making the tool more accessible. Multi-band filtering can be added to handle complex audio applications, and the project can be optimized for mobile and embedded systems. Machine learning integration could predict and adjust filter parameters, while automated noise analysis could dynamically identify and filter noise. Exploring advanced filter designs and 3D audio processing for VR/AR, as well as developing the tool for educational purposes, will further expand its functionality and impact.

9.References:

Audio

[1] https://pagalfree.com/music/jai-ho-2008.html

Noise

[2] https://pixabay.com/sound-effects/search/drums/

[3] https://audioalter.com/equalizer

Code

[4] https://in.mathworks.com/help/signal/ref/bandpass.html

Theoretical Background

[5] https://eepower.com/technical-articles/how-to-create-bandpass-filters/#

[6] https://www.sciencedirect.com/topics/computer-science/bilinear-transformation

[7]https://en.wikipedia.org/wiki/Band-pass_filter#:~:text=A%20band%2Dpass%20filter%20or,of%20a%20band%2Dstop%20filter

Application

[8] https://en.wikipedia.org/wiki/Noise_reduction#In_audio

[9] https://en.wikipedia.org/wiki/Band-pass_filter#Applications

[10] https://pmc.ncbi.nlm.nih.gov/articles/PMC3388307/