EE 321: Digital Signal Processing

Python Group No. P02

Group Members:

Yash Kataria (210102100)

Tejas Kadre (210102104)

Lakshit Sethia (210102123)

Task 1: The audio file goodmorning.wav contains saying "good morning" with an echo noise added to it. Load the audio signal from the file into a numpy array. Use Python to design a digital filter that will eliminate the echo contamination from the audio file. Derive the transfer function, and the difference equation. Plot the magnitude and phase response of the designed filter. Use Python to apply your inverse filter to the sound file and show that it has removed the echo. Plot the original noisy signal and the filtered signal in the same graph to compare their frequency content. (Do not use any DSP Python libraries for filter design. Build your own filter function.)

Approach: The goal of this project is to design a digital filter to eliminate echo contamination from an audio file using Python. The chosen approach involves implementing a custom FIR filter with a band-pass characteristic.

Filter Design Choices:

- **Frequency Pass Band [250 Hz,2200 Hz] :** A band-pass filter was selected to target and filter the specific frequency range containing the echo. From the graphs below, we figured out that the human voice typically fell within the range of 250 Hz to 2200 Hz with weak echo and noise falling below 250 and above 2200 Hz. Therefore, a band-pass filter centered around these frequencies was designed to preserve the speech components while attenuating the echo with some noise.
- **Filter Order 1023:** The filter order (number of taps) was chosen based on a trade-off between computational complexity and frequency resolution. A higher filter order allowed for a sharper roll-off and hence a steeper graph in the frequency response, but it also increases the computational load.
- **Hamming Window:** The Hamming window provides a good compromise between main lobe width and side lobe attenuation, thus reducing spectral leakage and resulting in a smooth frequency roll-off.
- **Pass-Band and Stop-Band Frequencies and Scaling:** The stop-band frequencies for the band-pass filter were chosen as [125 Hz, 1100 Hz], corresponding to half of the original

range due to the Nyquist theorem. The scaling factor (2) was applied to normalize the filter response, frequencies, and maintain the overall amplitude of the signal.

Filter Equations:

- **Ideal LPF impulse response function**, which is basis for every filter:

$$hd[n] = (fs/fc)\cdot sinc(2fc(n - \frac{(N-1)}{2})/fs)$$

where f_s is the sampling frequency, *fc* is the cutoff frequency, N is the number of taps.

- **Hamming Window Equation:**

$$w(n) = \alpha - \beta cos(2\pi n/(N-1))$$

where α=0.54, β=0.46, and N is number of taps and n is the index vector.

- **Filtering Process Equation:**

The ideal filter kernel is transformed to ideal band-pass and multiplied by the Hamming window to form the final filter kernel coefficients:

$$h(n) = hd'[n] \cdot w(n)$$

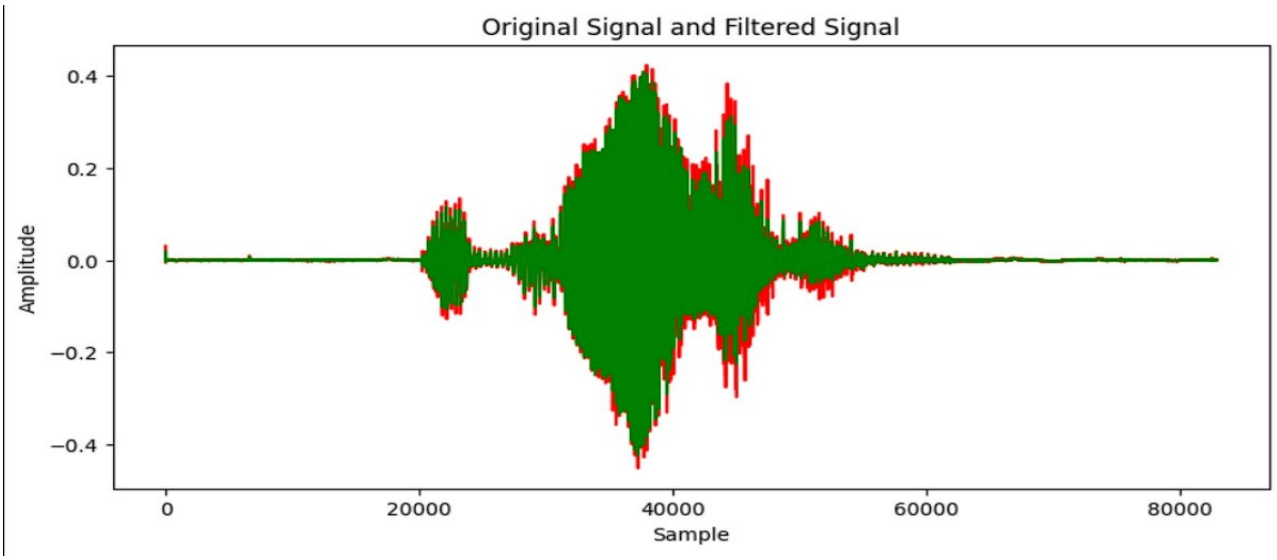
where $hd'[n]$ is the impulse response of ideal band-pass filter.

The filtering process involves convolving the input signal *x(n)* with the filter coefficients *h(n):*

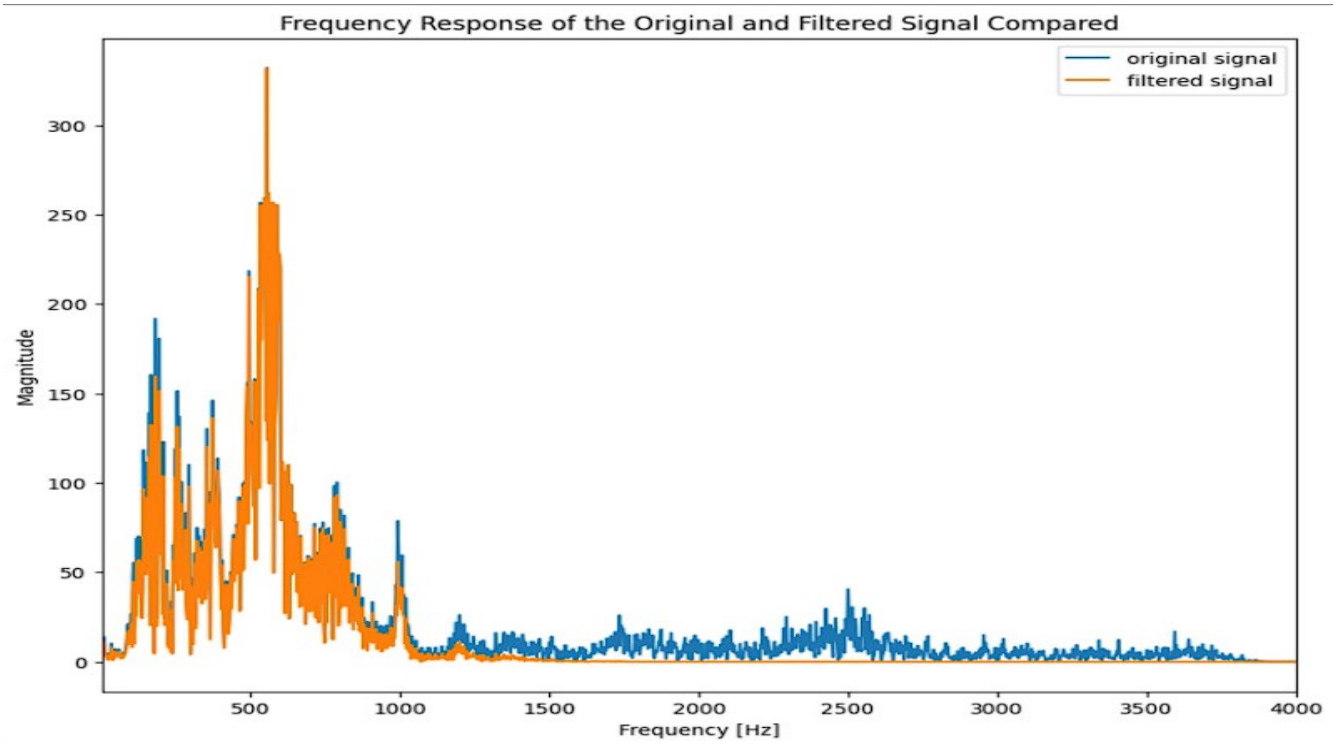
$$y(n) = x(n) * h(n)$$

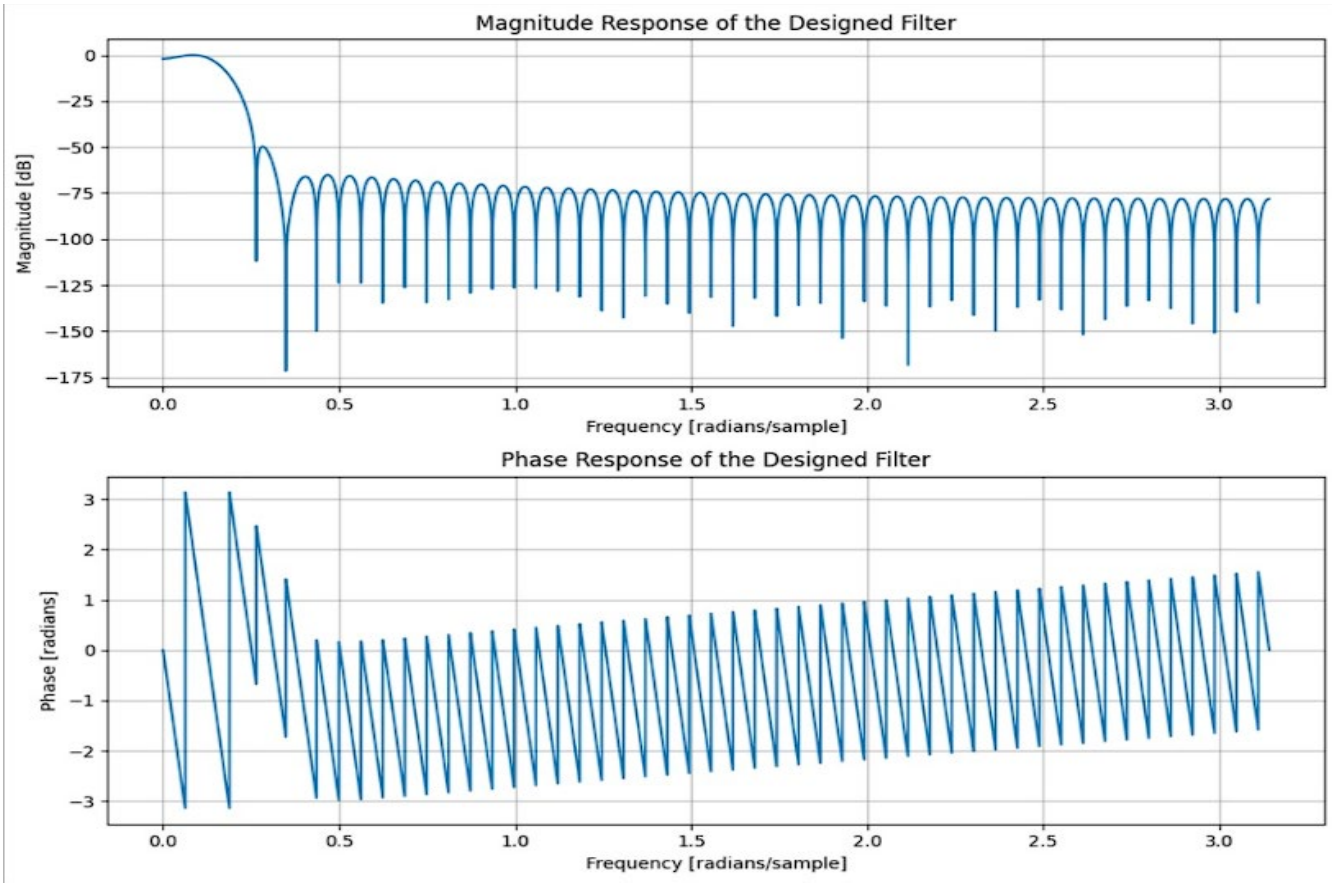
Graphs and Conclusion:

The original signal and the filtered signal were also plotted against each other for comparison. The magnitude and phase responses of the designed filter were plotted to visualize its characteristics. The desired results were hence obtained using the above filter.



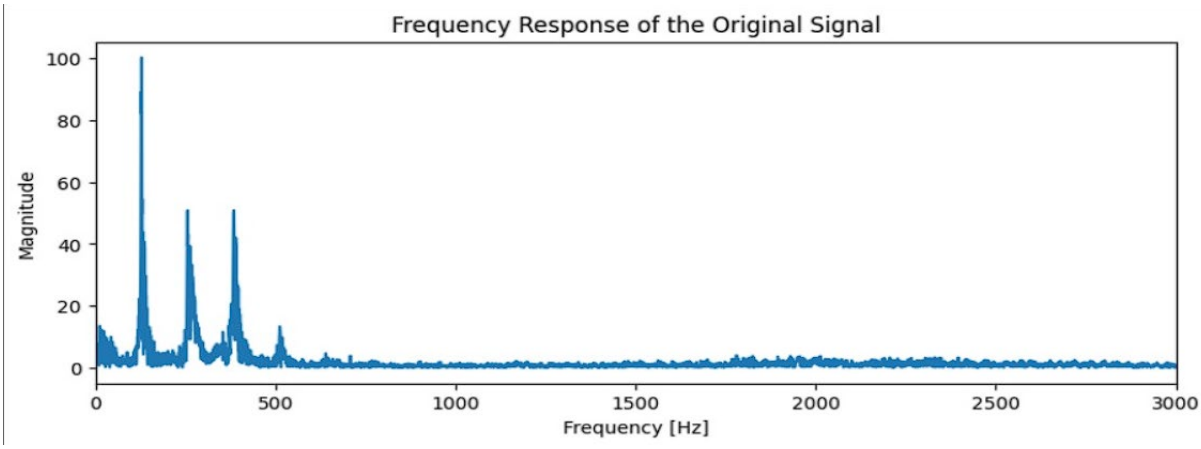
AMPLITUDE VS TIME OF FILTERED (GREEN) AND ORIGINAL (RED)





Task 2: The file safety.wav contains the speech signal for the word "safety" uttered by a male speaker, sampled at 8 kHz. The signal has a significant amount of background noise (as it was recorded in a computer laboratory). Load the audio signal from the file into a numpy array. Develop procedures to segment the signal into voiced, unvoiced, and silence (background noise) portions using Python. Plot the magnitude and phase response of the designed filter. Also, plot the original raw audio signal and segmented audio signals. Compute and plot the Power Spectral Density (PSD) for each segment that you obtain and study its characteristics. (Do not use any DSP python libraries. Build your own functions.)

Approach: We identified the three components of the given audio clip from its frequency response – the frequencies corresponding to maximum magnitude if of voiced component, medium/average magnitude is of unvoiced component, and minimum magnitude is of noise. Voiced components also tend to have periodic characteristics. Hence we designed three band-pass filters with different bandwidths to extract the three components.



Filter Design Choices:

- **First Band-Pass Range [1,850]:** These frequencies correspond to voiced segment with max magnitude.
- **Second Band-Pass Range [950,1050]:** These frequencies correspond to unvoiced segment with medium magnitude.
- **Third Band-Pass Range [1051,3000]:** These frequencies correspond to noise segment with min magnitude.
- **Filter Order 101:** To reduce the computational complexity and time complexity of the algorithm, an order of 101 was taken as we designed three filters so the program needed to be efficient to run in less time.
- **Hamming Window:** Like in the previous task, the Hamming window is symmetric, simplifying the design and implementation of filters. It is also easy to implement to the ideal LPF and transform it using band-pass transformation technique.

Filter Equations:

- **Ideal LPF impulse response function**, which is basis for every filter:

$$hd[n] = (fs/fc)\cdot sinc(2fc(n - \tfrac{(N-1)}{2})/fs)$$

where f_s is the sampling frequency, fc is the cutoff frequency, N is the number of taps.

Band-Pass impulse response function:

$$hd'[n] = \frac{\left(\sin\ (wc2(n-\mu)) - \sin\ (wc1(n-\mu))\right)}{\pi\,(n-\mu)}$$

when n≠μ where μ=(N-1)/2, wc2 and wc1 are angular frequencies corresponding to high and low band cutoffs. But when n=μ,

$$hd'[n] = \frac{(wc2 - wc1)}{\pi}$$

- **Hamming Window Equation:**

$$\boldsymbol{w(n) = \alpha - \beta cos(2\pi n/(N-1))}$$

where α=0.54, β=0.46, and N is number of taps and n is the index vector.

- **Filtering Process Equation:**

The ideal filter kernel is transformed to ideal band-pass and multiplied by the Hamming window to form the final filter kernel coefficients:

$$\boldsymbol{h(n) = hd'[n] \cdot w(n)}$$

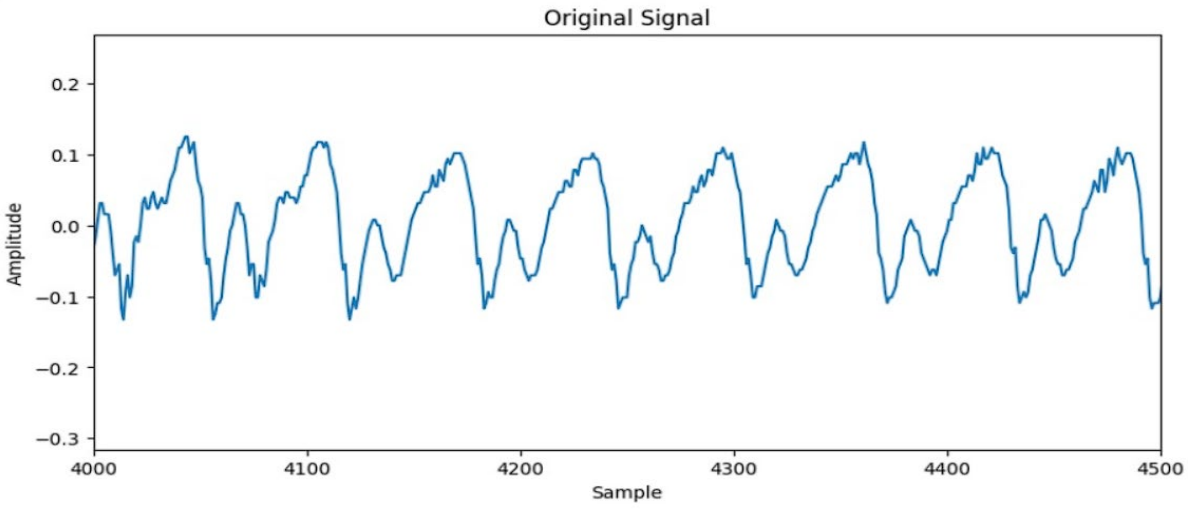
The filtering process involves convolving the input signal $x(n)$ with the filter coefficients $h(n)$:

$$\boldsymbol{y(n) = x(n) * h(n)}$$

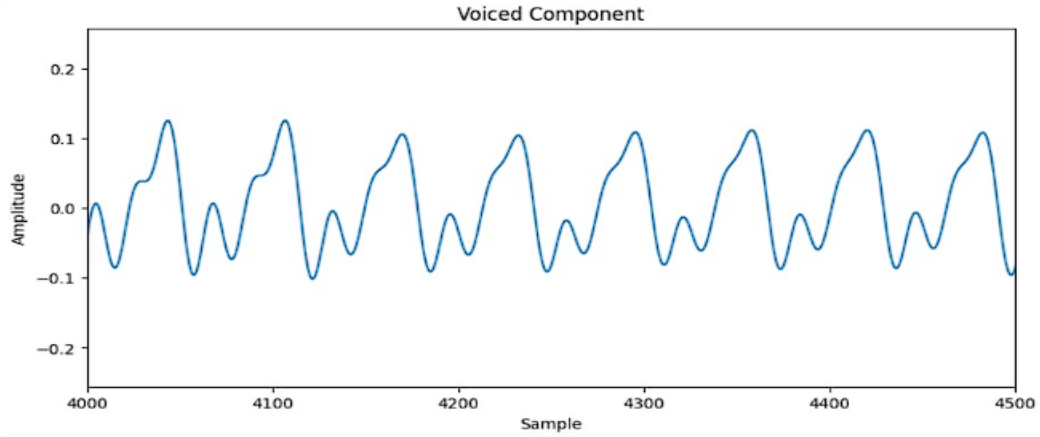
The convolution is performed in both the forward and backward directions to ensure a linear phase response.

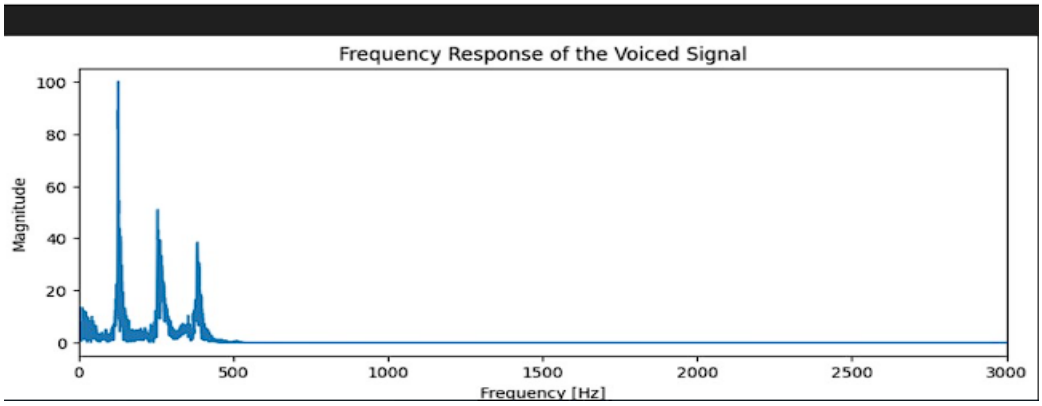
- **Normalization:** The filter is normalized to have unity gain at DC (0 Hz), preventing overall magnitude changes in the signal.

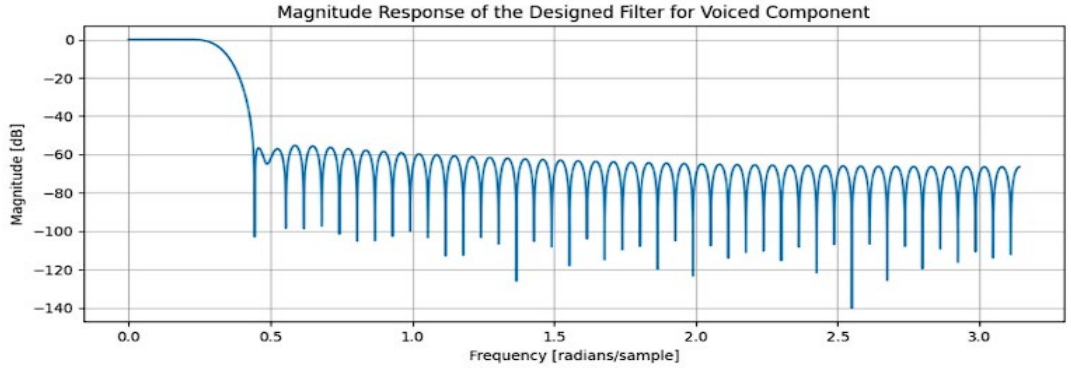
Graphs and Conclusion:

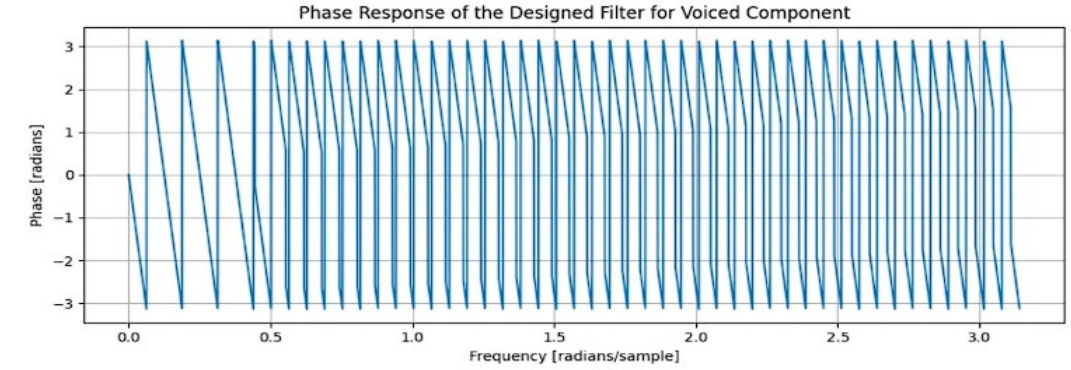


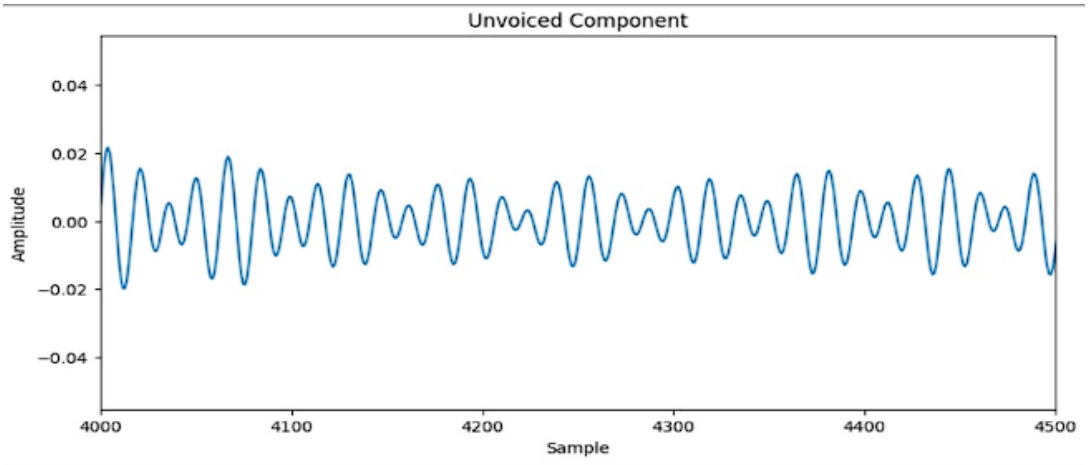
AMPLITUDE VS TIME OF ORIGINAL SIGNAL

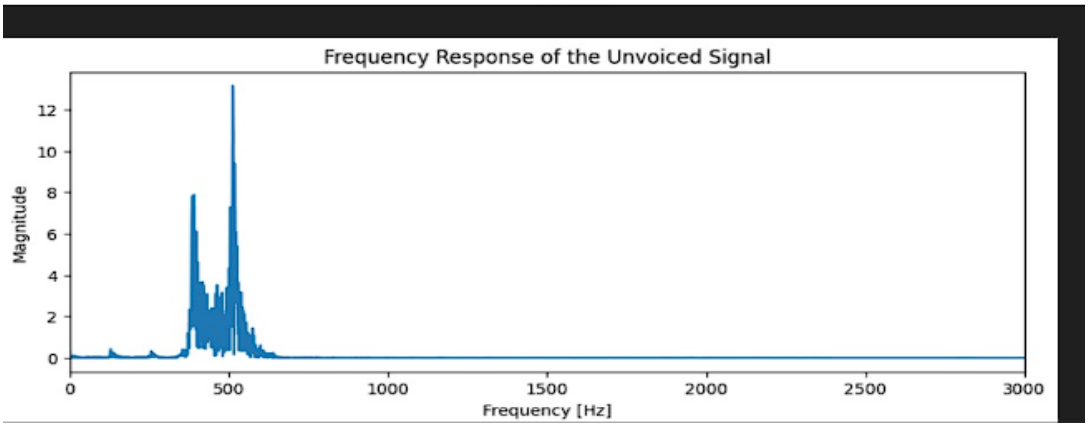

Voiced Component
Amplitude
Sample
0.2
0.1
0.0
−0.1
−0.2
4000
4100
4200
4300
4400
4500

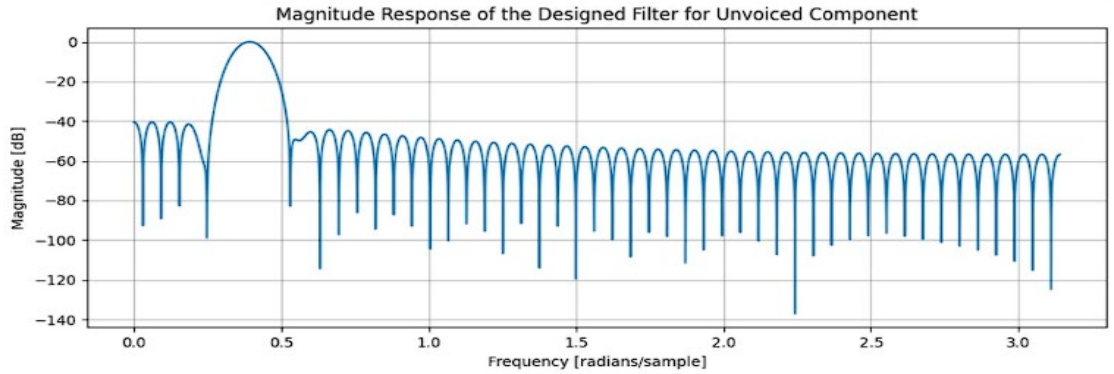

Frequency Response of the Voiced Signal
Magnitude
Frequency [Hz]
100
80
60
40
20
0
0
500
1000
1500
2000
2500
3000

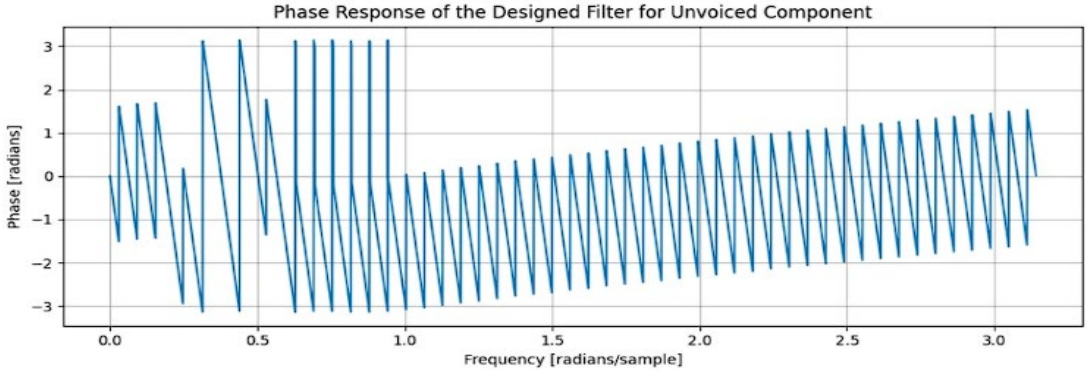

Magnitude Response of the Designed Filter for Voiced Component
Magnitude [dB]
Frequency [radians/sample]
0
−20
−40
−60
−80
−100
−120
−140
0.0
0.5
1.0
1.5
2.0
2.5
3.0

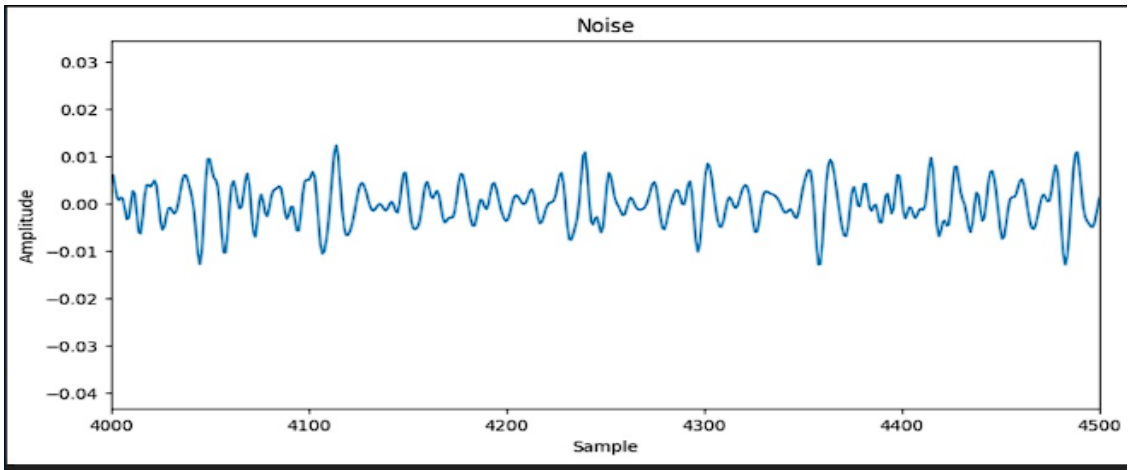

Phase Response of the Designed Filter for Voiced Component
Phase [radians]
Frequency [radians/sample]
3
2
1
0
−1
−2
−3
0.0
0.5
1.0
1.5
2.0
2.5
3.0

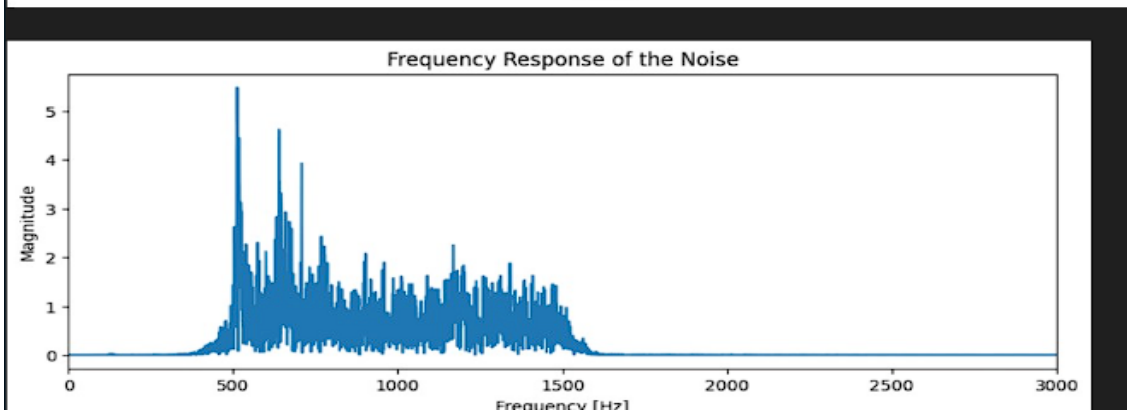

Unvoiced Component
Amplitude
Sample
0.04
0.02
0.00
−0.02
−0.04
4000
4100
4200
4300
4400
4500

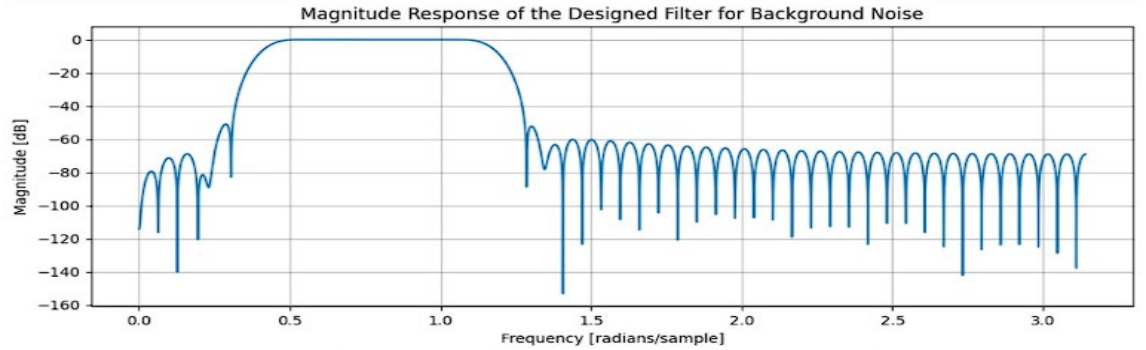

Frequency Response of the Unvoiced Signal
Magnitude
Frequency [Hz]
12
10
8
6
4
2
0
0
500
1000
1500
2000
2500
3000

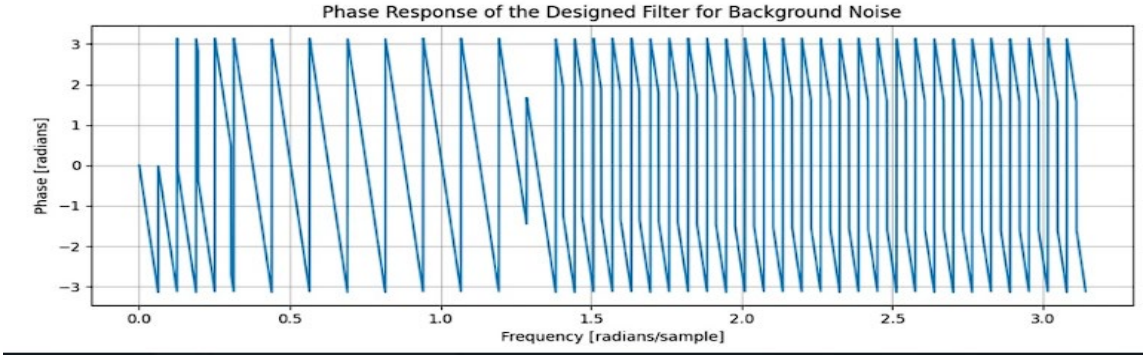

Magnitude Response of the Designed Filter for Unvoiced Component
Magnitude [dB]
Frequency [radians/sample]
0
−20
−40
−60
−80
−100
−120
−140
0.0
0.5
1.0
1.5
2.0
2.5
3.0


Phase Response of the Designed Filter for Unvoiced Component
Phase [radians]
Frequency [radians/sample]
3
2
1
0
−1
−2
−3
0.0
0.5
1.0
1.5
2.0
2.5
3.0


Noise
Amplitude
Sample
0.03
0.02
0.01
0.00
−0.01
−0.02
−0.03
−0.04
4000
4100
4200
4300
4400
4500

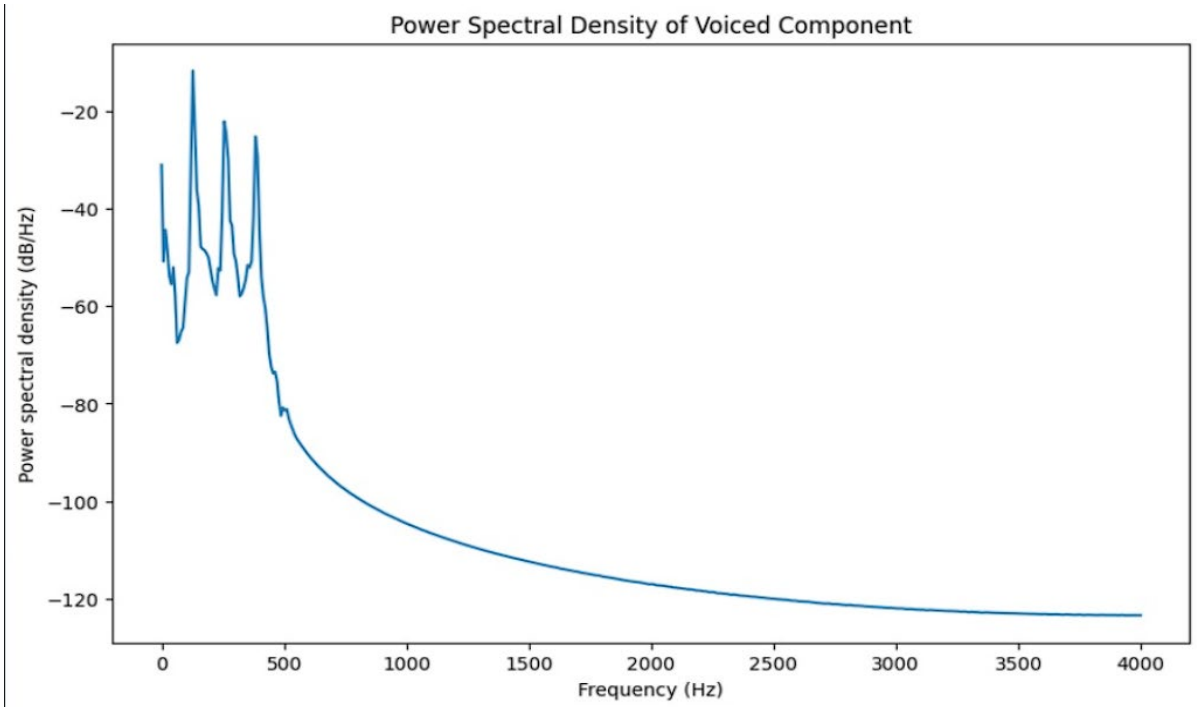

Frequency Response of the Noise
Magnitude
Frequency [Hz]
5
4
3
2
1
0
0
500
1000
1500
2000
2500
3000

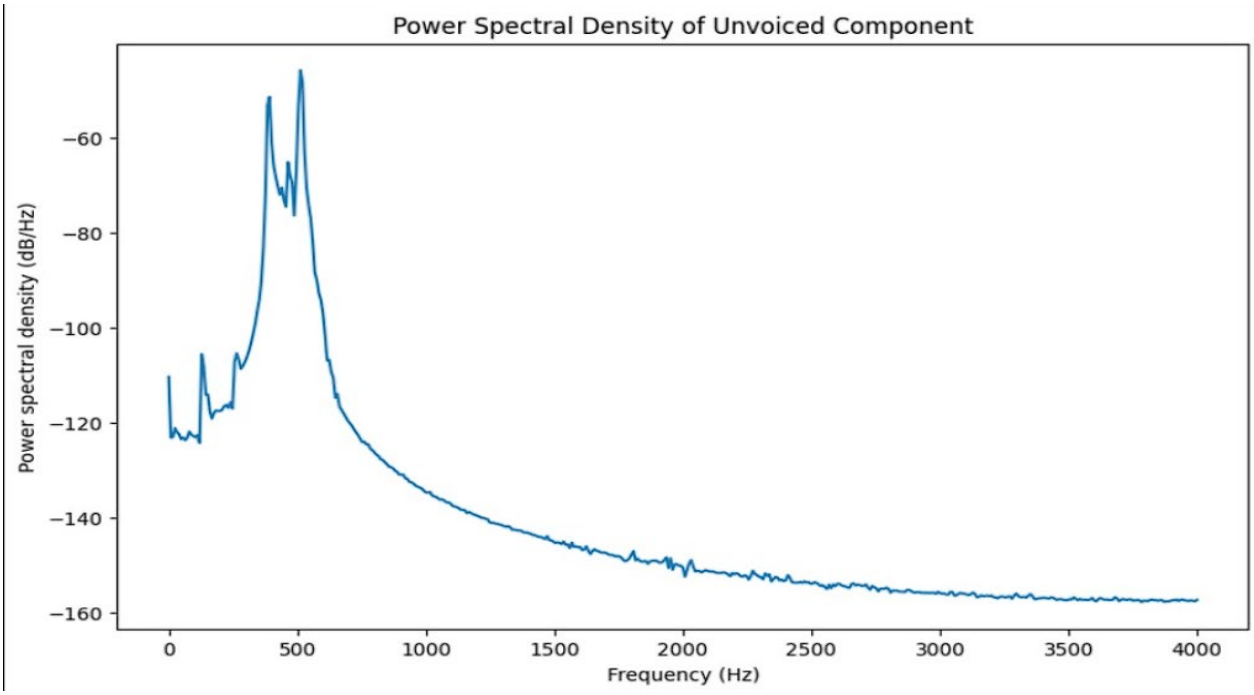

Magnitude Response of the Designed Filter for Background Noise
Magnitude [dB]
Frequency [radians/sample]
0
−20
−40
−60
−80
−100
−120
−140
−160
0.0
0.5
1.0
1.5
2.0
2.5
3.0

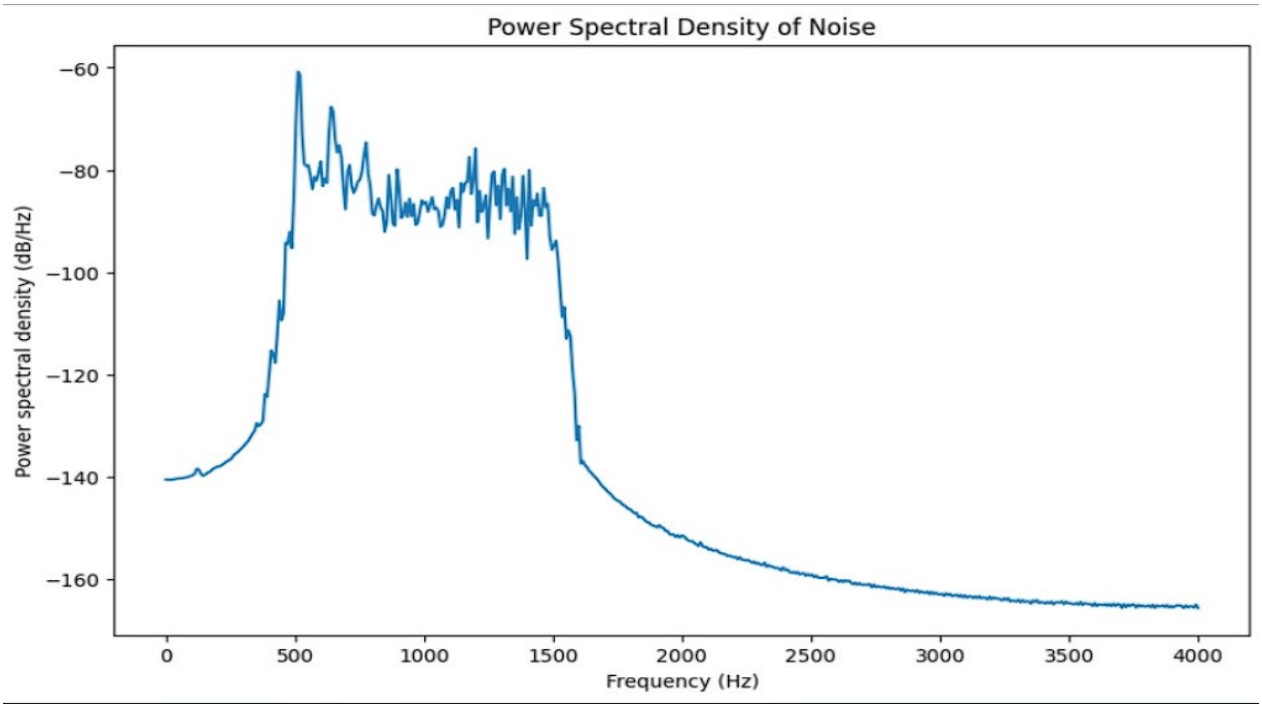

Phase Response of the Designed Filter for Background Noise
Phase [radians]
Frequency [radians/sample]
3
2
1
0
−1
−2
−3
0.0
0.5
1.0
1.5
2.0
2.5
3.0

POWER SPECTRAL DENSITY:

The PSD was computed by squaring the magnitude spectrum values and normalizing them by the length of the signal. As we can see, PSD of voiced component is maximum (around –20 dB) under 600 Hz. PSD of unvoiced component is medium (around –50 dB between 500 – 800 Hz). PSD of noise component is lowest (around –60 dB between 800 – 1750 Hz).






Power Spectral Density of Noise
Power spectral density (dB/Hz)
−60
−80
−100
−120
−140
−160
0
500
1000
1500
2000
2500
3000
3500
4000
Frequency (Hz)