Filter Banks

Rishi Nandha V - EE21B111

EE6311 - Multirate Digital Signal Processing

November 26, 2023

All codes used in these Experiments can be found here under "Experiment 2".

1 LPF Prototype

N was chosen to be **49** for the Type-2 LPF Prototype.

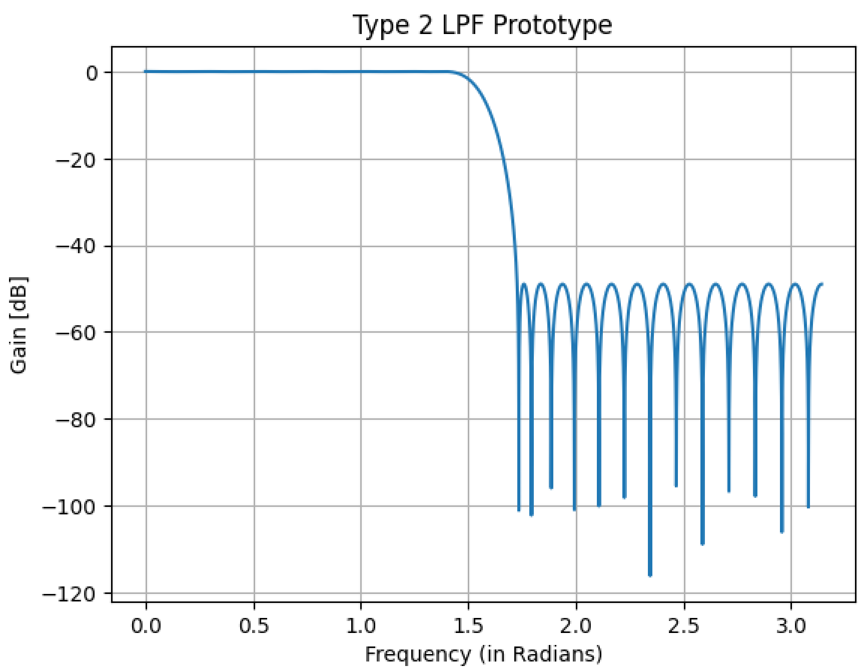


Figure 1: LPF Prototype Magnitude Response

```
omega_p_4 = 0.45 * 0.5
omega_s_4 = 0.55 * 0.5
delta_p_4 = 0.02
delta_s_4 = 0.02
H_naught = remez(50, [0, omega_p_4, omega_s_4, 0.5],[1,0], [delta_p_4 ,delta_
  ↪s_4])
w_4, h_4 = freqz(H_naught, worN=8000)
plt.plot(w_4, 20  * np.log10(np.abs(h_4)))
```

2 Synthesis Bank Filter Choices and Reconstruction

```
H0 = np.zeros(49)
H0[::2]=H_naught[0::2]
H1 = np.zeros(49)
H1[::2]=H_naught[0::2]
```

$$Y(z) = \frac{1}{2} \cdot \begin{bmatrix} z^{-1} & 1 \end{bmatrix} \cdot F(z^2) \cdot E(z^2) \cdot \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \cdot W^* \cdot \begin{bmatrix} X(z) \\ X(-z) \end{bmatrix}$$

$$E(z^2) = W^* \cdot \begin{bmatrix} H_0^0(z^2) & 0 \\ 0 & H_1^0(z^2) \end{bmatrix}$$

$$F(z^2) = \begin{bmatrix} K^0(z^2) & 0 \\ 0 & K^1(z^2) \end{bmatrix} \cdot W$$

$$\implies \boxed{T(z) = z^{-1}\left(K_0(z^2)H_0^0(z^2) + H_1^0(z^2)K_1(z^2)\right)}$$

$$\implies \boxed{A(z) = z^{-1}\left(K_0(z^2)H_0^0(z^2) - H_1^0(z^2)K_1(z^2)\right)}$$

2.1 A: No Aliasing (Not PR)

$T(z)$ is a type 1 filter, thus we can write $T_{zp}(w) = |T(w)|$

```
K0_a = H1.copy()
K1_a = H0.copy()
K0H0_a = np.convolve(H0, K0_a, mode='full')
K1H1_a = np.convolve(H1, K1_a, mode='full')
w_5, h_5 = freqz(K0H0_a+K1H1_a, worN=48000)
plt.plot(w_5, 20  * np.log10(np.abs(h_5)))
```

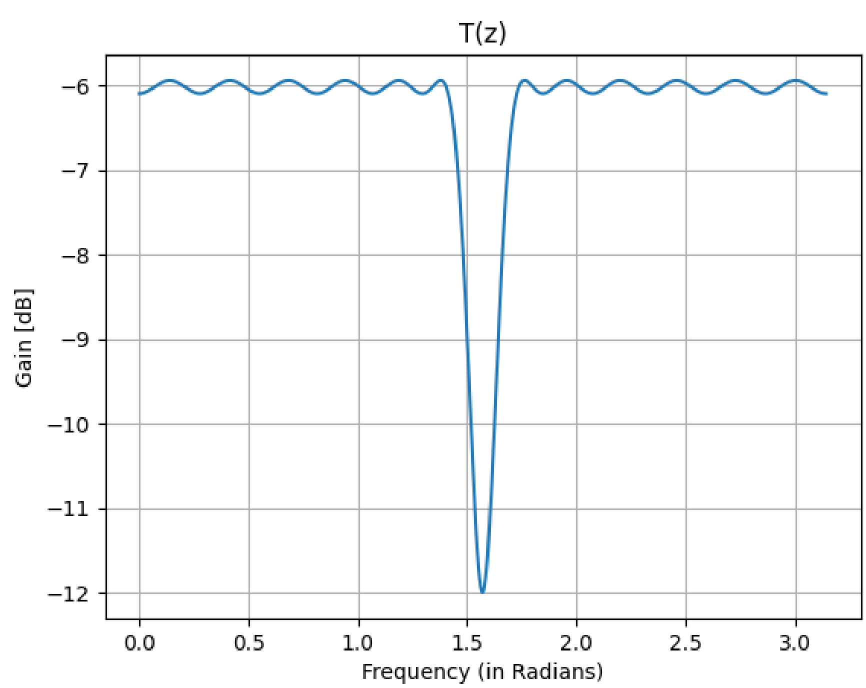


Figure 2: T(w) for FB without Aliasing

2.2 B: A Bad Choice of Filters

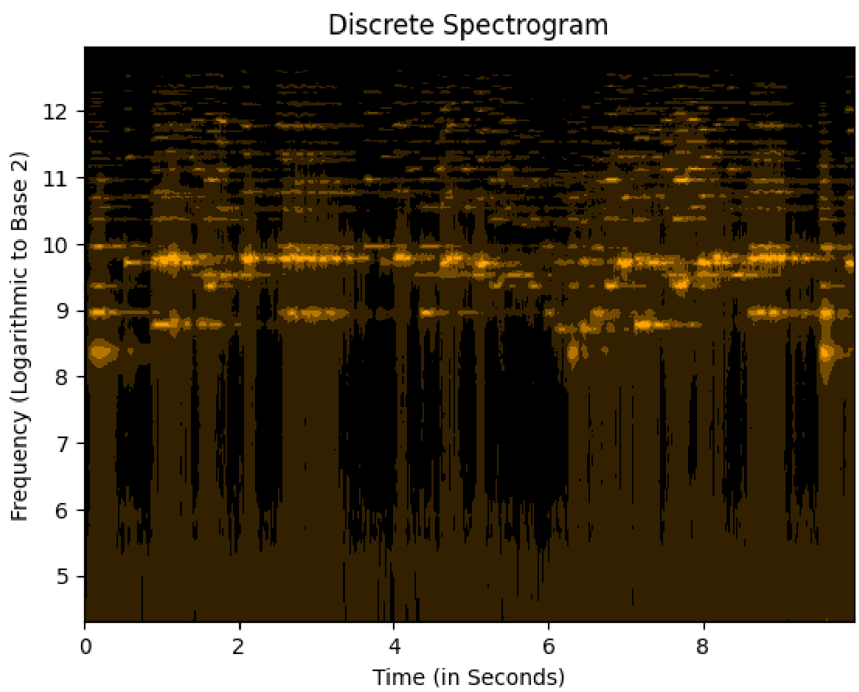
```
K0_b = H0.copy()
K1_b = H1.copy()

K0H0_b = np.convolve(H0, K0_a, mode='full')
K1H1_b = np.convolve(H1, K1_a, mode='full')
```

3 Testing With Audio

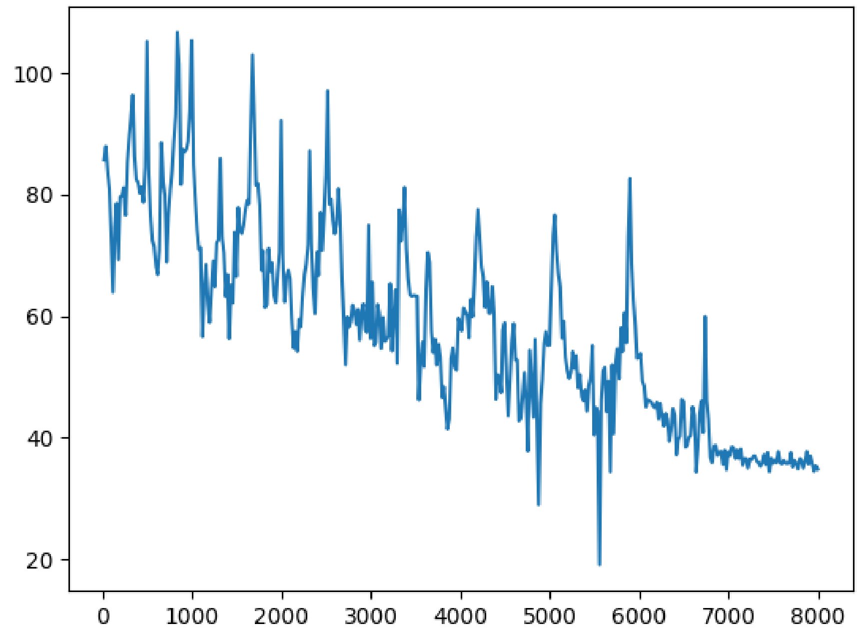
3.1 Music16kHz.wav

3.1.1 Non-PR, Without Aliasing

The Spectrogram looked like this:



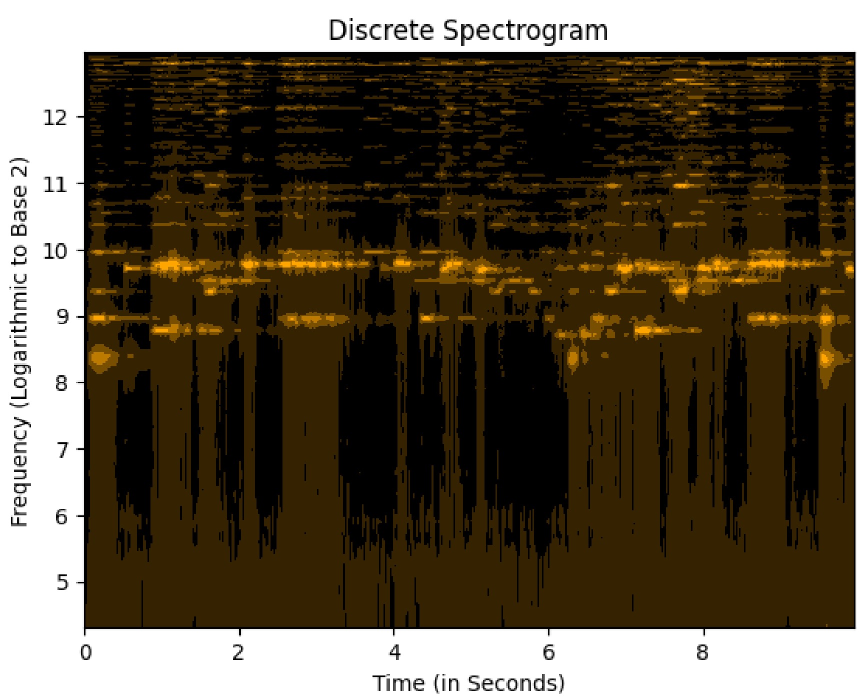
A Snippet of Magnitude Spectrum at a specific time-stamp:



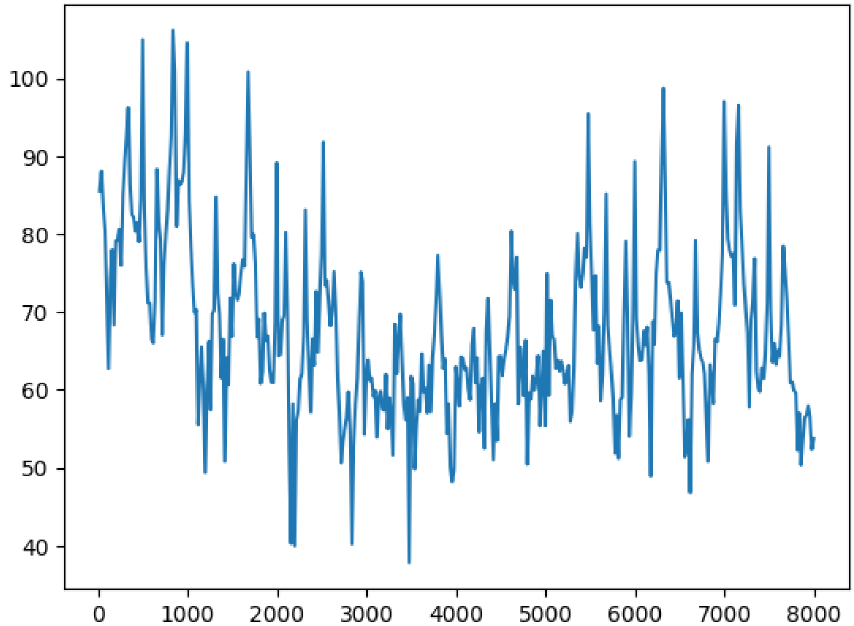
From the looks of the Magnitude Spectrum and how the Audio sounds, the audio is practically unchanged. Even if the $T(z)$ was not perfectly reconstruction, practically there is no noticable differences.

3.1.2 With Aliasing

The Spectrogram looked like this:



A Snippet of Magnitude Spectrum at a specific time-stamp:

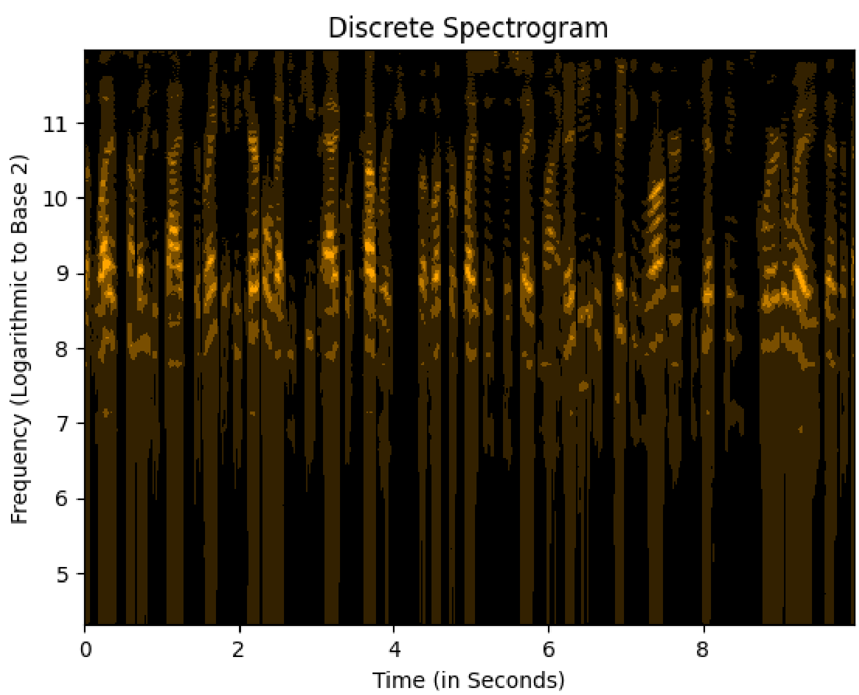


The playback sounds severly damaged. The magnitude spectrum is clearly displaying aliasing too.

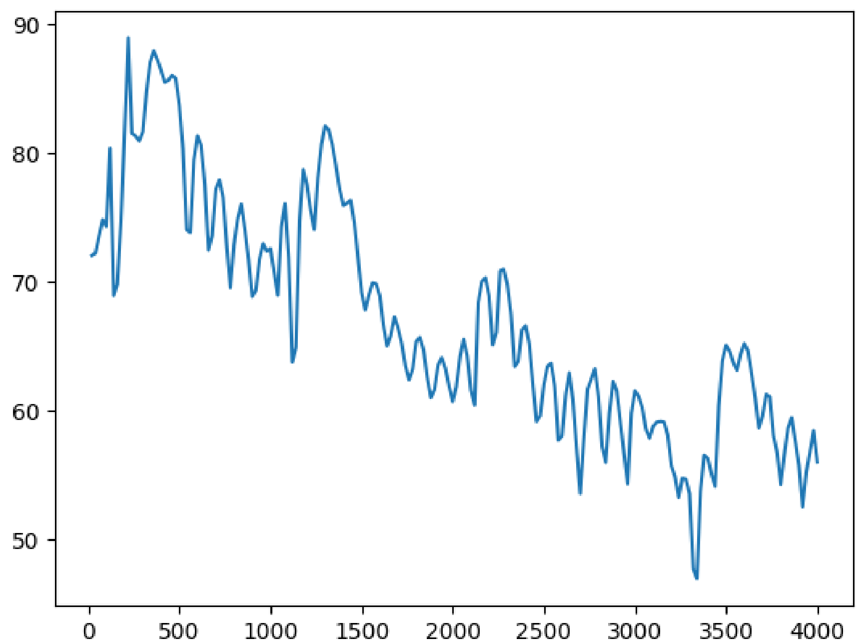
3.2 Speech8kHz.wav

3.2.1 Non-PR, Without Aliasing

The Spectrogram looked like this:



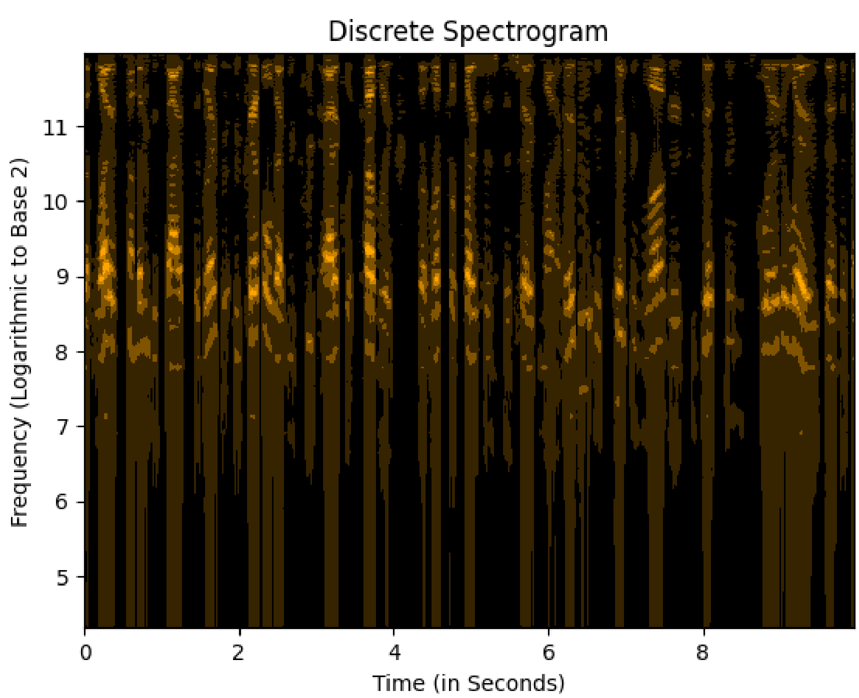
A Snippet of Magnitude Spectrum at a specific time-stamp:



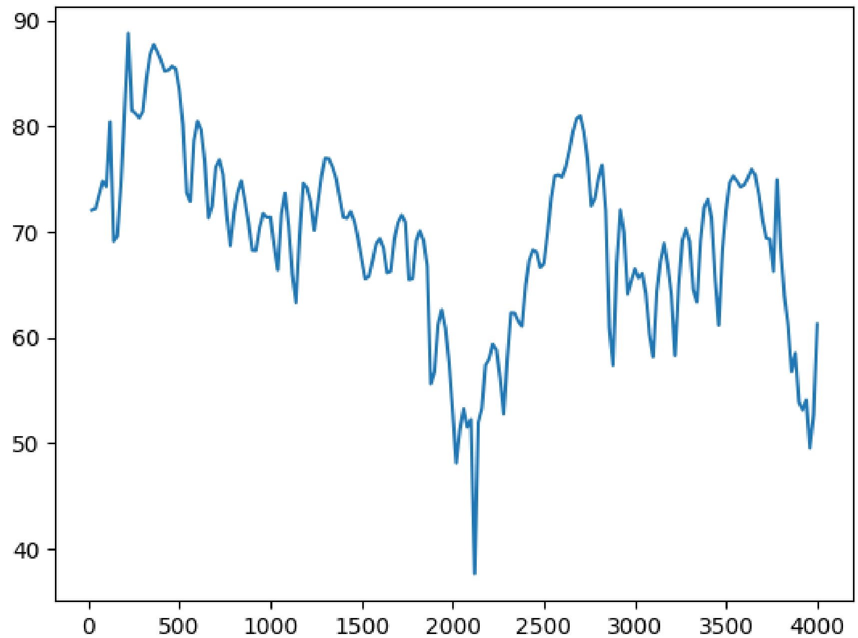
From the looks of the Magnitude Spectrum and how the Audio sounds, again, this sounds almost identical and unaffected.

3.2.2 With Aliasing

The Spectrogram looked like this:



A Snippet of Magnitude Spectrum at a specific time-stamp:



From the looks of the Magnitude Spectrum and how the Audio sounds, again, this sounds severly damaged and aliased

3.3 Source Code for Testing with Audio

Here's is the code used for one of the four cases above:

```
X_1,Y_1,Z_1,sampling_rate_1, audio_1 = openaudio("music16khz.wav")
from scipy.signal import lfilter

audio_1_negz = audio_1.copy()
audio_1_negz[1::2] -= audio_1_negz[1::-2]*2
audio_1_FB = lfilter(K0H0_a+K1H1_a, 1.0, audio_1).astype(np.uint16) +␣
↪lfilter(K0H0_a-K1H1_a, 1.0, audio_1_negz).astype(np.uint16)
samples = audio_1_FB.size

with wave.open('Music_FB_a.wav', 'wb') as wav_file:
    wav_file.setnchannels(1)
    wav_file.setsampwidth(2)
    wav_file.setframerate(sampling_rate_1)
    wav_file.setnframes(samples)
    wav_file.writeframes(audio_1_FB.tobytes())

X_3,Y_3,Z_3,sampling_rate_3,audio_3 = openaudio("Music_FB_a.wav")
spectrum2(X_3,Y_3,Z_3,"Music_FB_a.jpeg")
k = 12000
plt.plot(np.power(2,Y_3), Z_3[:,k].flatten())
plt.show()
```

4 Links To Delivarables:

1. Chosen value of N, Code for Prototype Filter and the Magnitude Spectrum of $H^0(z)$ can be found page 1
2. Plots of $T_{zp}(w)$ for the Filter Banks without Aliasing is in page 2
3. Verbal Comparison of Audio Outputs and Magnitude Spectrum can be found in pages 3 to 7
4. Source Code used for applying these FBs over Audio is in page 7