Digital Signal Processing Lab
Experiment 4
Importance of low-frequency temporal structure of speech and frequency content of speech in speech perception.

Name: Anand Jhunjhunwala
Roll No: 17EC30041
Group No: 63

Objective:
The goal of the assignment is to implement a part of the paper Shannon et al 1995 and try to gain an understanding of relative importance of the low-frequency temporal structure of speech and frequency content of speech in speech perception.

Theory:
This experiment is based on the paper implementation mentioned above and to understand the importance of the low-frequency temporal structure of speech and frequency content of speech in speech perception.

According to the paper:
Traditionally the recognition of speech has been thought to require frequency-specific cues. Spectral energy peaks in speech reflect the resonant properties of the vocal tract and thus provide acoustic information on the production of the speech sound.

The above methods, however, met only limited success in identifying acoustic cues under various listing conditions and with various talkers. To perform better in these situations amplitude compression and spectral reduction techniques have been used, however, these manipulations resulted in stimuli that were still highly complex in their temporal spectral characteristics.

It was found that even total removal of spectral cues carries a relevant amount of information. Keeping this in mind, we preserved amplitude and temporal cues while systematically varying the amount of spectral information to understand its effect on speech identification.

To do so,
We passed the audio signal to a number of continues bandpass filter whose bandwidth is in GP, then its frequency component is removed by envelope detection which preserves amplitude and temporal information, to bring randomness in the spectral information noise is added with this extracted envelope after passing it to the same bandpass filter.
The number of bandpass filters was varied to see its effect on the output audio file.

Matlab code [File name: assign4_mod.m]

- This code is responsible for performing all the above-mentioned steps.

Code snippet:

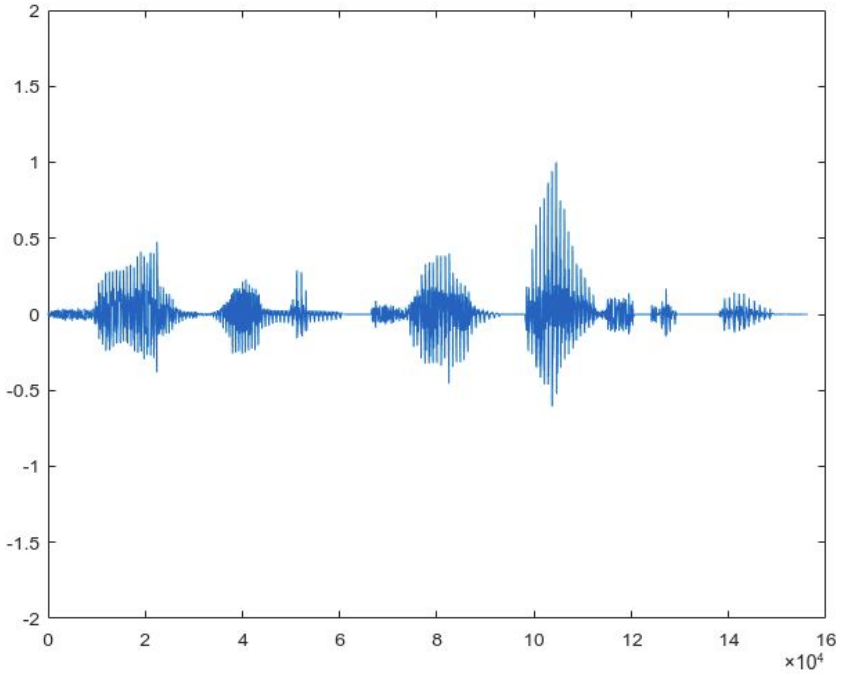
```
order = 4;
filename = "fivewo.wav";
[X,Fs] = audioread(filename);
norm = Fs/2;
value = input('Enter the number of bandpass filters  \n');
[B_l, A_l] = butter(order*2, 240/norm);
noise = rand(1,length(X));
result = zeros(1,length(X));
B = [;];
A = [;];
Y = [;];
Y_e = [;];
Y_el = [;];
r = nthroot(5760/90, value);
for i = 1:value
    [B(i,:), A(i,:)] = butter(order/2, [(90*(r.^(i-1)))/norm, (90*(r.^(i)))/norm]);
    Y(i,:) = filter(B(i,:),A(i,:),X);
    Y_e(i,:) = abs(hilbert(Y(i,:)));
    Y_el(i,:) = filter(B_l, A_l, Y_e(i,:));
    n = filter(B(i,:),A(i,:),noise);
    result = result + n.*Y_el(i,:);
end
result = result';
s1 = 'result';
s2 = '.wav';
r1 = strcat(s1,int2str(value));
r2 = strcat(r1,s2);
audiowrite(r2,result,Fs);
```

The code is fully modular which takes the input as a number of the bandpass filter (let XY) to be used and produce the output audio file named resultXY.wav.

Code explanation:
Order: It is the order of the bandpass filter to be used, and filename takes the name of the audio file to be used for processing.
Norm: is just a normalization factor, with Fs as the sampling frequency of input audio signal.
Value: holds a user-defined number of bandpass filters.
Noise: and the **result** are used to create noise and store the final result.
B, A: is two arrays which hold Butterworth filter coefficients of all filter.
Y: It holds the filtered output of the audio signal from all the above filters.
Y_e: It holds the extracted envelope of the filtered output.
Y_el: It holds the low-passed output of the envelope.
n: It holds the bandpass output for noise.
At last, the result is created by multiplying this noise with a low-passed envelope and adding all these values for each bandpass filter.

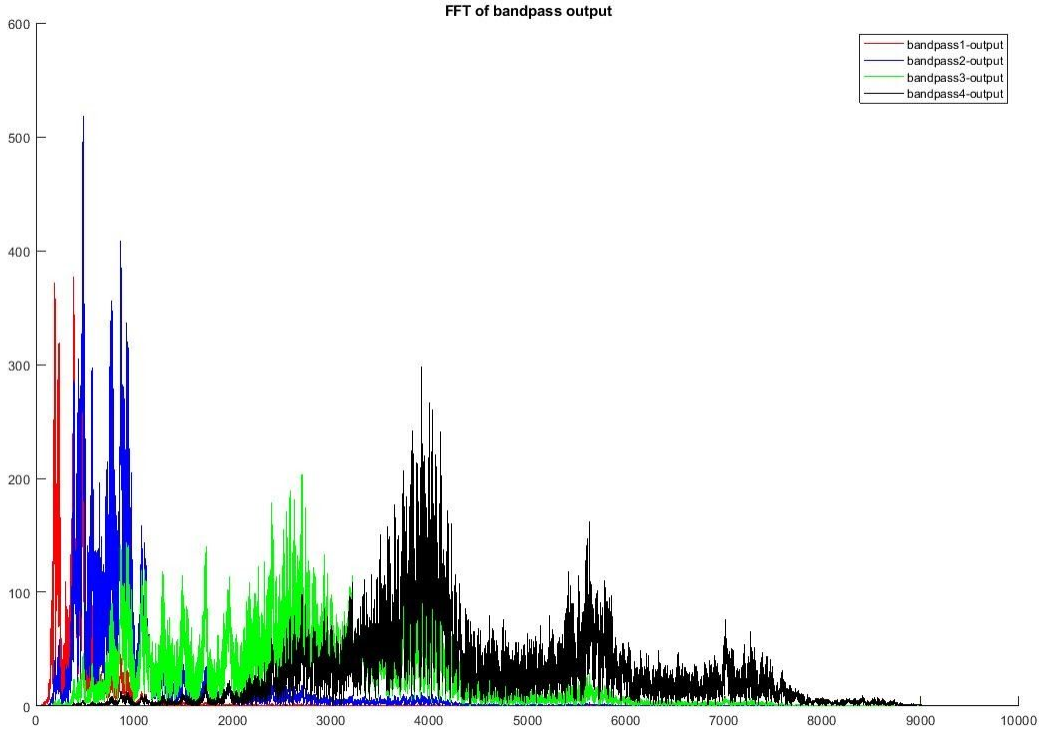
At last, this result is converted into the final audio file.

Results:
Audio Signal:



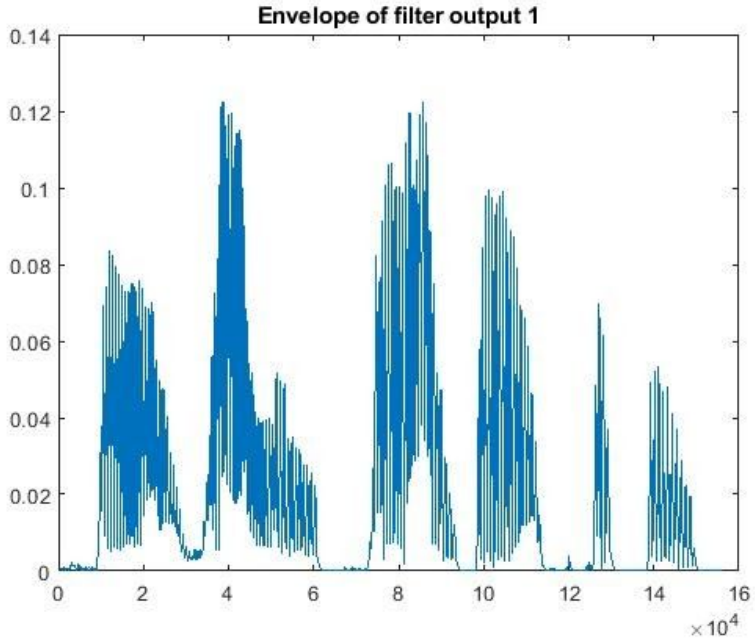
For N=4

Output of bandpass filters.



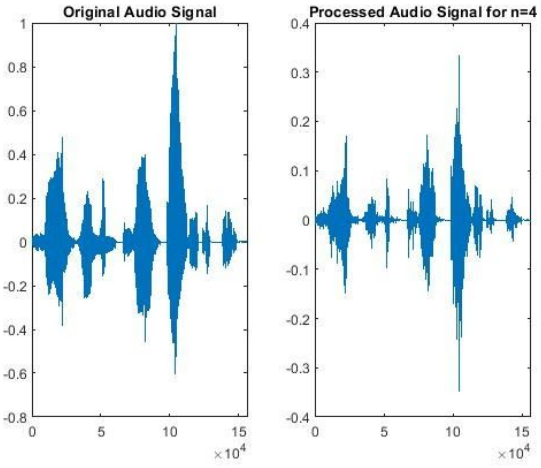
Clearly from the graph, we can see that bandwidth of filters increases making a GP.
And the resulting FFT is consistent within the passband.

Envelope for N=4

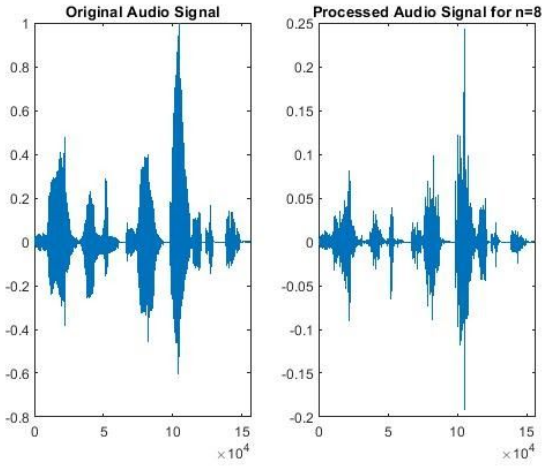


Results for different N

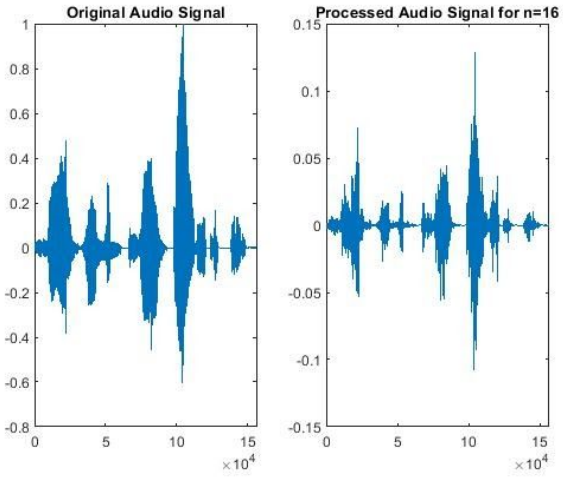
For N=4:



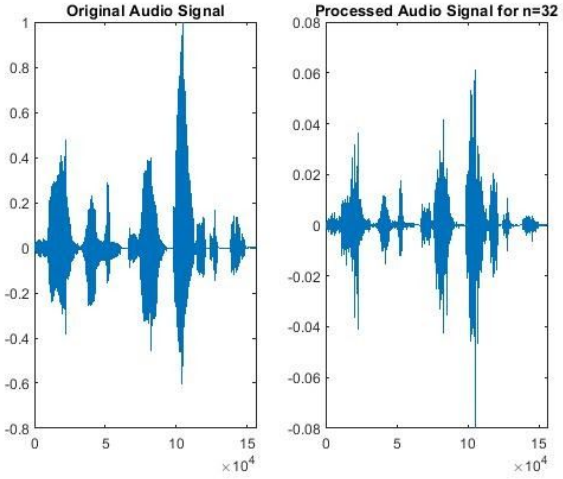
For N=8:



For N=16:



For N=32:



This link contains all the original Matlab code and audio signals:
https://drive.google.com/open?id=13Y_2JPKSv4RZlRz_b5T_LRVnIm4pIOQh

Discussions:

- From the bandpass output I observed that bandwidth is in GP as expected, and filter output was confined within its passband.
- From the envelope of the signal, I observed that the upper half of the envelope resembles exactly with the upper half of the audio signal.
- With the increase in N, I observed that the signal becomes more close to the original signal but amplitude values decrease with increasing N, this is due to the fact that with an increase in N, more filters are introduced which causes more attenuation resulting in less amplitude.
- The audibility of the signal obtained by this process is somewhat low when compared to that of the original signal, this is because of attenuation as discussed earlier.
- The intelligibility of the signal is mainly affected by the quality of the signal.
- 8 bands are sufficient to clearly understand the sound, there wasn't much improvement in the signal quality when the number of bands is increased from 8 to 16.
- For the 128 bands, audio signal was very clear almost similar to the original sound.
- The signal quality was not that good when the number of bands is 1,2,3,4 and the words of the speech were not clearly understood.
- Thus we can conclude that the low-frequency temporal structure of speech and frequency content of speech is sufficient for speech recognition to a greater extent.
