SPEECH RECOGNITION

AIM:

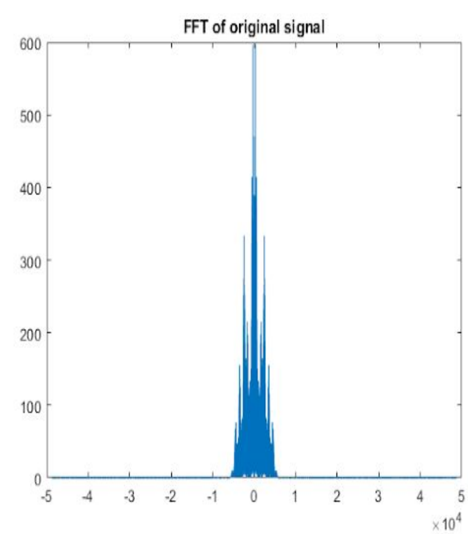
Relative Importance Of Low-frequency Temporal Structure Of Speech And Frequency Content Of Speech In Speech Perception.

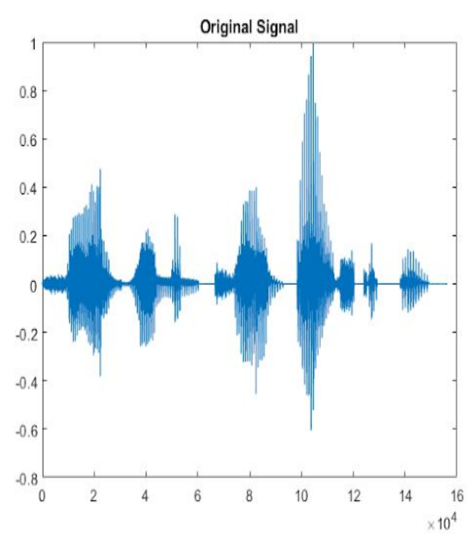
THEORY:

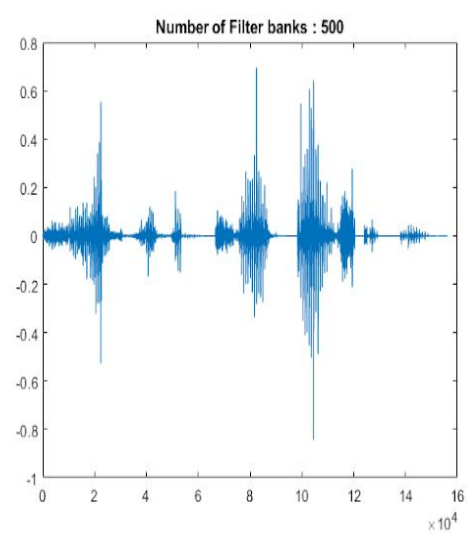
The goal of the assignment is to implement a part of the paper (1) Shannon et al 1995 and try to gain an understanding of relative importance of the low-frequency temporal structure of speech and frequency content of speech in speech perception. Use the methods described in the paper (see Note 7 in the paper) to modify the provided speech signal. Have 6 cases: 1 band, 2 bands, 3 bands, 4 bands and 8 bands and 16 bands– the filters should be logarithmic ally spaced and should span 90 Hz to 5.76 kHz (6 octaves). Thus, for 1 band case, the lower and higher cut-offs should be 90 Hz and 5.76 kHz, and for the 2 band case, the 2 filters should be 90 Hz to 720 Hz and 720 Hz to 5.76 kHz. Use fourth-order band pass Butter worth (MATLAB function butter) filters instead of elliptic IIR and no need for the reemphasize filter. For the filtering operation use suitable filters you have learned. For extraction of envelope use the Hilbert transform (https://in.mathworks.com/help/signal/ug/envelope-extraction-usingtheanalytic-signal.html) and then low pass filter (again use butter for the low pass filter) with a cutoff of 240 Hz. Create the new sounds and have someone who has not heard the sentence tell you what they hear (give comments). No need to perform elaborate statistics with multiple speeches sounds as in the paper. Get a qualitative idea of whether intelligibility increases or not and by how many bands is the sound clearly understood.

SOURCE CODE AND RESULTS:

```
[signal, Fs] = audioread('fivewo.wav');

figure;
signal_fft = fftshift(abs(fft(signal,2^nextpow2(length(signal)))));
f = linspace(-Fs/2, Fs/2, length(signal_fft));
plot(f, signal_fft);
title('FFT of original signal');

figure;
plot(signal);
title('Original Signal');

noise = rand(1, length(signal));

%figure;
%out1 = my_butter_filter(90, 576, Fs, signal);
%signal_fft = fftshift(abs(fft(out1,2^nextpow2(length(out1)))));
%f = linspace(-Fs/2, Fs/2, length(signal_fft));
%plot(f, signal_fft);
%title('FFT of bandpass signal');

for i = 500:500
    interval = (log(5760)-log(90))/i;
    env_noise = zeros(1, length(signal));
    for j = 1:i
        F_low = exp(log(90)+interval*(j-1));
        F_high = exp(log(90)+interval*j);
        bandpass_signal = my_butter_filter(F_low, F_high, Fs, signal);
        [yup, ylow] = envelope(bandpass_signal, 25, 'analytic');
        bandpass_noise = my_butter_filter(F_low, F_high, Fs, noise);
        env_noise = env_noise + my_vector_elementwise_multiply(yup, bandpass_noise);
    end
    figure;
    plot(env_noise);
    title("Number of Filter banks : " + i);

end

audiowrite("Env_Noise.wav",env_noise',Fs);
```







DISCUSSION:

- In this experiment The temporal information of a speech signal was kept whereas the spectral information is not retained totally.(By extracting the envelope of speech signal at several frequency bands and modulating a noise signal with it).
- We noticed that as we increased the number of bands the output was more prominent or closer to actual speech signal as more frequency components were obtained in this case.
- One problem that we faced was real time plotting is very difficult in this experiment, if we plot while computing, the computation time is becoming very high and not very feasible.
- In this experiment 3 to 4 bands should be sufficient for hearing the sound again by the person who has heard it earlier. For the person who has not heard more bands may be required.
- Here we used butter worth 3rd order filter as the computation speed will be much higher in case of butter worth fourth order filter.
- The output is amplified at last so that the sound can be heard clearly.