Spectral Speech Denoising

Yasin Emir Ariturk

Dilay Kilavuz

Eren Oz

I. Introduction

Audio signals are often corrupted by background environment noise and buzzing. Audio denoising aims at attenuating the noise while retaining the underlying signals. The aim of the project is to clean/extract the noise from a speech signal in spectral manner.

Following the theoretical background from the paper "Audio Denoising by Time-Frequency Block Thresholding" practical solution has been provided to speech denoising problem with different approaches/techniques. These techniques and terms will be covered in the report. In Section II, Short Time Fourier Transform on Speech Denoising is given.

II. Short-Time Fourier Transform on Speech Denoising

One of the most common way to analyze frequency components of a signal is performing Discrete Fourier Transform on that signal. But this method is useful at constant or fairly changing frequencies. When it comes to the signals, of which frequency contents vary over the time, we use Short-Time Fourier Transform (STFT) and spectograms. The signal types of which frequencies vary with time speech, music, seismology etc.

The application of STFT depends on classical DFT. But since the frequency changes with the time it's not logical to use DFT. Instead, the signal is separated into segments or windowed in each time under certain window lengths. The effect of the window length and different windowing methods (Hamming, Chebyshev, Kaiser etc.) are important concepts and to be discussed in the project.

I. Effect Of Window Length On Speech Signal and Spectograms

If we know the sampling frequency F_s, we can label the axis of spectogram as follows:

Frequency resolution = $\frac{F_s}{L}$

Width of time slices = $\frac{L}{F_s}$

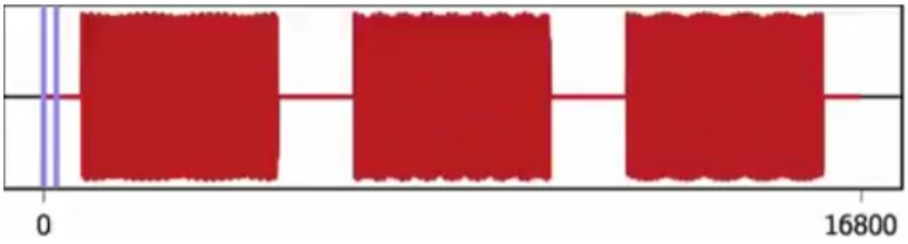


Figure 1: Analyzed Signal with 16800 Samples

To see the the effect of window length on an audio signal above which has 16800 samples and windowed under 2 different length of windows that have 32 and 256 respectively. Following 2 figures will be helpful to understand different window lengths effect on speech signal and spectograms;

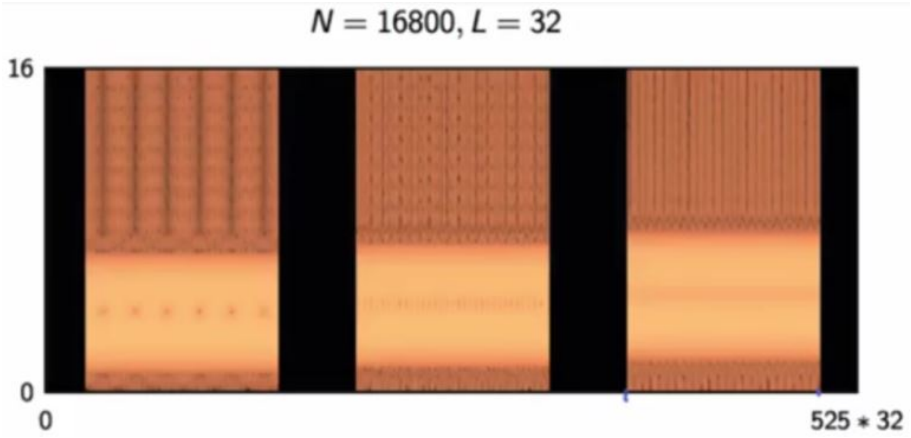


Figure 2: Short Windowed Signal

As you can see in the figure, in Short Window which called as Wide-band Spectogram, there are many time slices therefore precise location of transitions can obtain. On the other hand, there is not much Discrete Fourier Transform (DFT) points, therefore frequency resolution is uncertain.

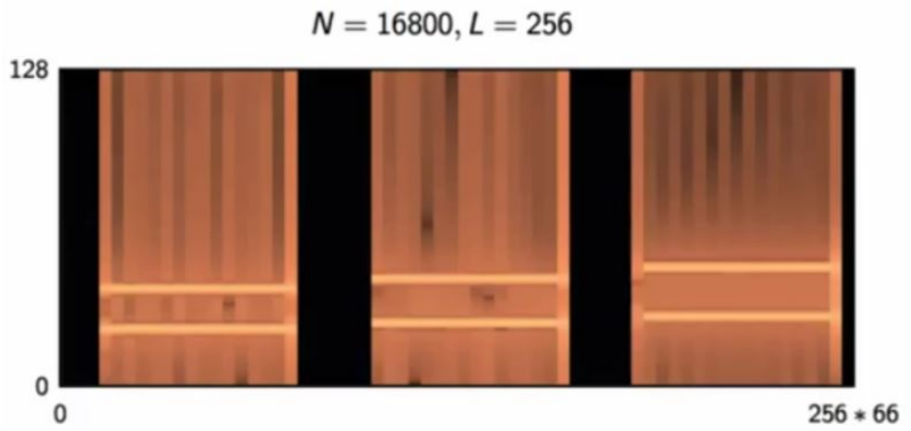


Figure 3: Long Windowed Signal

As you can see in the figure, in Long Window which called as Narrow-band Spectogram, there are more DFT points therefore frequency resolution of the signal is clear. On the other hand, more sound signals can be obtained in this period, therefore precision in time is less confidential.

II. Approach to the Problem

In our project, as the first practice we have performed is diagonal estimation and Wiener attenuation rule in time-frequency aspect, based on the reference paper 'Audio Denoising by Time-Frequency Block Thresholding'. Solution steps can be summarized into the steps below:

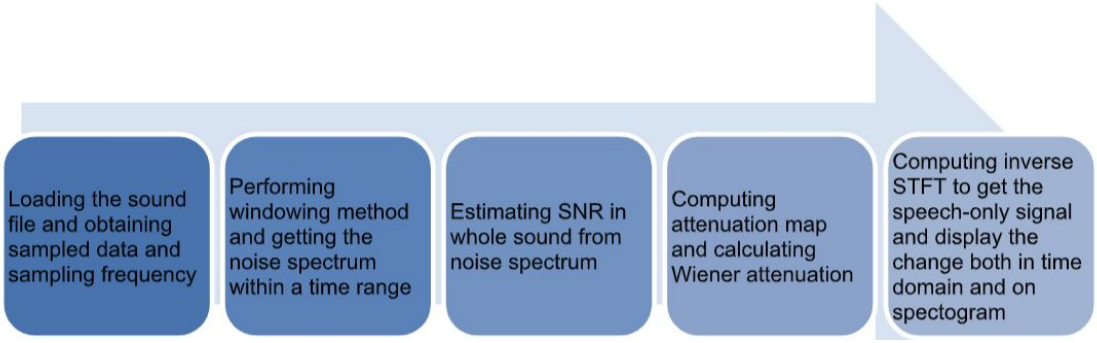


Figure 4: Summarized Solution Steps

III. Diagonal Estimation and Wiener Attenuation Rule

To detail the solution blocks, the approaches in each block are visualized as below:

Figure 5: After Loading the Sound File, Performing a Windowing Method and getting the Noise Spectrum within a time range

And after getting the Noise Spectrum, SNR is estimated in whole sound from noise spectrum. First step of estimation is calculation of the priori SNR which formulated as;

$$\xi[l,k] = F^2[l,k]/\sigma^2[l,k] \tag{1}$$

We also estimate posterior SNR, in order to estimate posterior SNR;

$$\gamma[l,k] = Y[l,k]^2/\sigma^2[l,k] \tag{2}$$

must be found, and posterior SNR formulated as;

$$\hat{\xi}[l,k] = \gamma[l,k] - 1 \tag{3}$$

After calculation of posterior SNR, we are computing attenuation map and calculating Wiener attenuation;

$$a[l,k] = \left(1 - \lambda \left[\frac{1}{\hat{\xi}[l,k]+1}\right]^{\beta_1}\right)_+^{\beta_2}$$

And after the calculation of Wiener attenuation, computing inverse STFT to get the speech-only signal and display the change both in time domain and on spectogram. Time domain results shown below; (Hamming filter used)

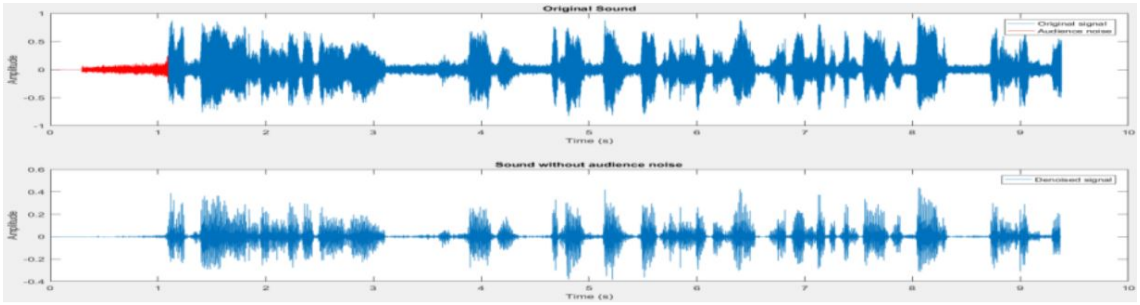


Figure 6: After the process, sound without audience noise is received.

IV. Windowing Techniques

I. Rectangular Window

Rectangular Window (known as the boxcar or Dirichlet window) is the simplest window, equivalent to replacing all but M values of a data sequence by zeros, making it appear as though the waveform suddenly turns on and off.

$$\omega_R(n) = 1, for -\frac{M-1}{2} \leq n \leq \frac{M-1}{2} \tag{4}$$

where M is the window length in samples. And, ω_R is equal to zero for other n values.

Rectangular window has narrowest main lobe width however dynamic range limitations of the rectangular window are inefficient that's why other types of windowing techniques proposed.

II. Hamming Window

The best result in our project is obtained by Hamming Window. The equation for Hamming Window sequence can be defined by;

$$\omega(n) = \alpha - \beta \cos\left(\frac{2\pi n}{N-1}\right), for -\frac{N-1}{2} \leq n \leq \frac{N-1}{2} \tag{5}$$

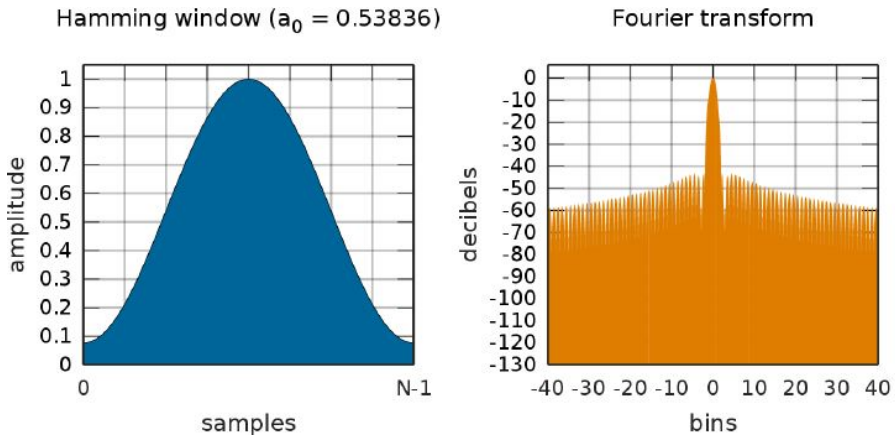


Figure 7: Time and Frequency Domain of Hamming Window

For $\alpha = 0.54$ and $\beta = 0.46$, Hamming Window equation becomes;

$$\omega(n) = 0.54 - 0.46\cos\left(2\pi\frac{n}{N}\right) \tag{6}$$

It is also known as raised cosine, because the zero-phase version, ω_0 is one lobe of an elevated cosine function.

The figure above illustrates that time and frequency domain of Hamming Window for a_0=0.54 which is optimal parameter for Hamming window. In this figure, in time domain it has the form of raised cosine, turns on slowly and turns off slowly, in addition, the end points never reach to 0.

III. Rectangular Window and Hamming Window Comparison

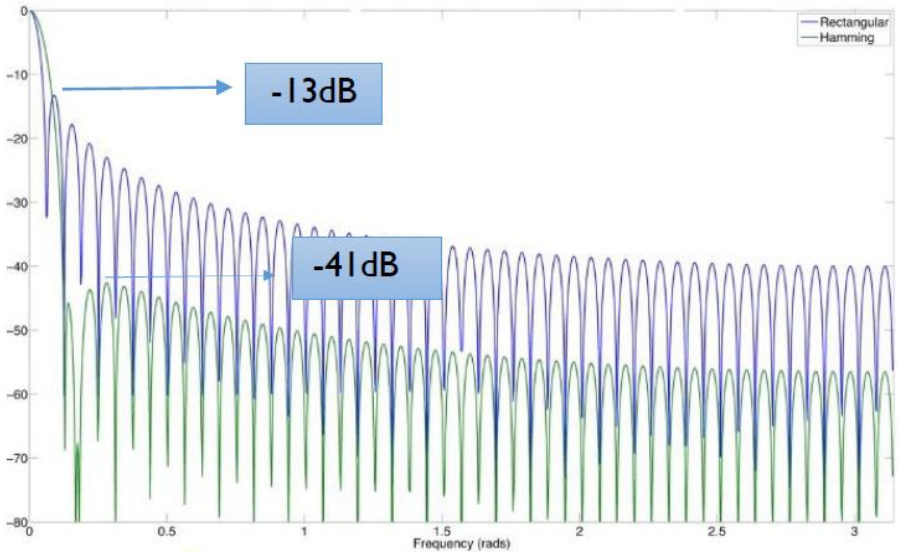


Figure 8: Rectangular Window and Hamming Window

The figure shows that while the main lobe width expands, side lobe height drops. For the Rectangular window, main lobe width is $4\pi/N$ which is considerably narrow, for Hamming window, this value is doubled. With the expansion of main lobe width, the side lobes level decreases from -13 dB which is the peak side lobe value for Rectangular Window to -41 dB which is the peak side lobe value for Hamming Window, however dynamic range is increasing.

Window	Main lobe width	Side lobe height
Rectangular	4π /N	-13dB
Hamming	8π /N	-41dB

IV. Chebyshev Window

The Dolph-Chebyshev Window (or Chebyshev window, or Dolph window) minimizes the Chebyshev norm of the side lobes for a given main-lobe width $2\omega_0$. The optimal Dolph-Chebyshev window transform can be written in closed form;

$$W(\omega_k) = \frac{\cos\left[M\cos^{-1}[\beta\cos(\frac{\pi k}{M})]\right]}{\cosh\left[M\cosh^{-1}(\beta)\right]}, k = 0, 1, \ldots, M-1 \tag{7}$$

$$\beta = \cosh\left[\frac{1}{M}\cosh^{-1}(10^{\alpha})\right], (\alpha \approx 2, 3, 4) \tag{8}$$

The α parameter controls the side-lobe level via the formula:

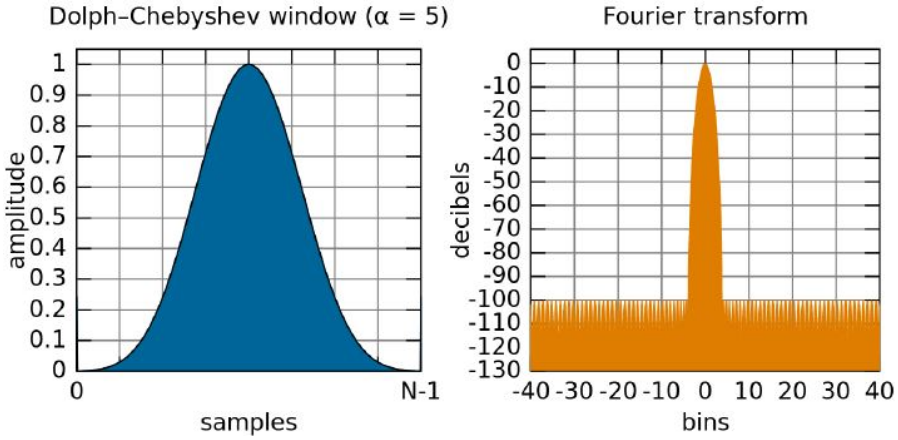


Figure 9: Time and Frequency Domain of Chebyshev Window (Side-Lobe level is -20α)

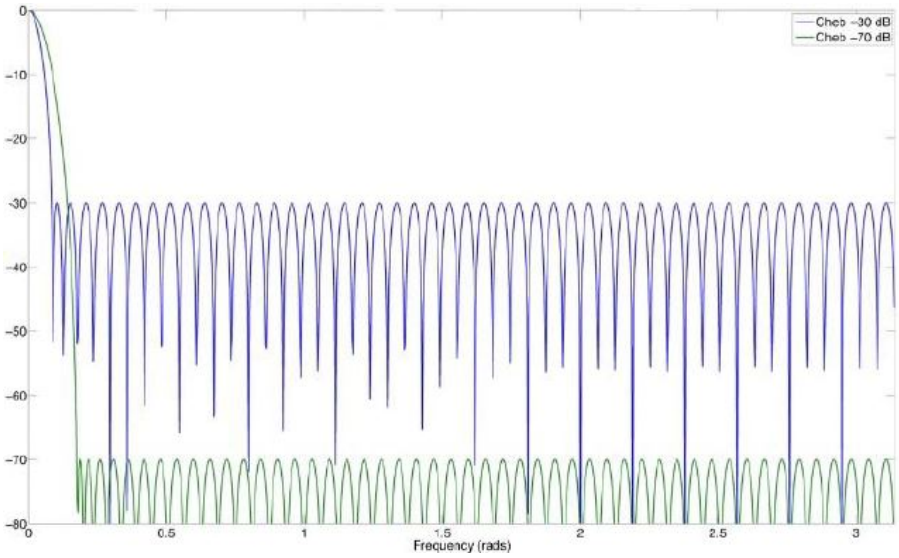


Figure 10: Frequency Domain of Chebyshev Window for different α parameters

Figure 9 illustrates that time and frequency domain of Chebyshev Window for α=5 parameter. In addition for frequency domain, Chebyshev window has uniform and equal height side lobes, therefore, by specifying α parameter directly control dynamic range. As expected, pushing the side lobes down generates the main lobe wider.

Figure 10 shows that frequency domain of Chebyshev Window for different α parameters. As mentioned before, unlike the Hamming window, the side lobes heights are uniform and equal. Also as expected, when α parameter increases, the main lobe width extends.

For this project, lots of different type windowing methods were tried, and as mentioned before the best result obtained by Hamming Window. On the other hand, when the Chebyshev Window applied to the sound, the result was quite poor compared to other types of windows.

V. Hamming Window and Chebyshev Window Comparison

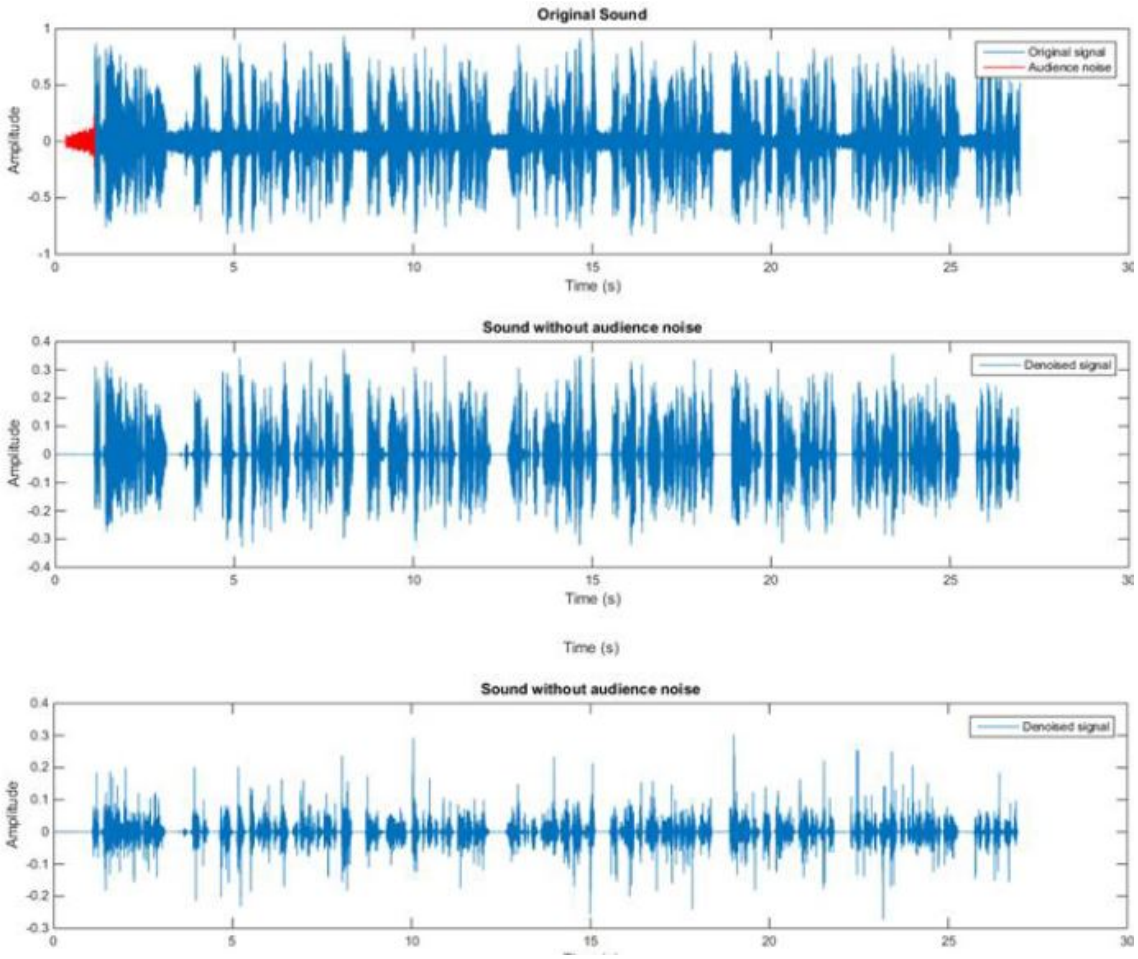


Figure 11: a) Original Sound, b) Hamming Windowed Result, c) Chebyshew Windowed Result

This figure illustrates the results of Hamming and Chebyshev Windows for project. First one, is original signal, and second one is Hamming Window result, and third one is Chebyshev Window result. Chebyshev Window reduces the requested signal while removes the audience noise.

VI. Kaiser Window

The coefficients of a Kaiser window are computed from the following equation:

$$w(n) = \frac{I_0\left(\beta\sqrt{1-\left(\frac{n-N/2}{N/2}\right)^2}\right)}{I_0(\beta)}, 0 \leq n \leq N \tag{9}$$

where I_0 is the zero-order modified Bessel function of the first kind ,and β is the Kaiser window parameter that affects the side lobe attenuation of the Fourier transform of the window.

To obtain a Kaiser window that designs an FIR filter with side lobe attenuation of α dB, $\beta = \pi\alpha$ is using. When β parameter equals to zero, it reduces to rectangular window.

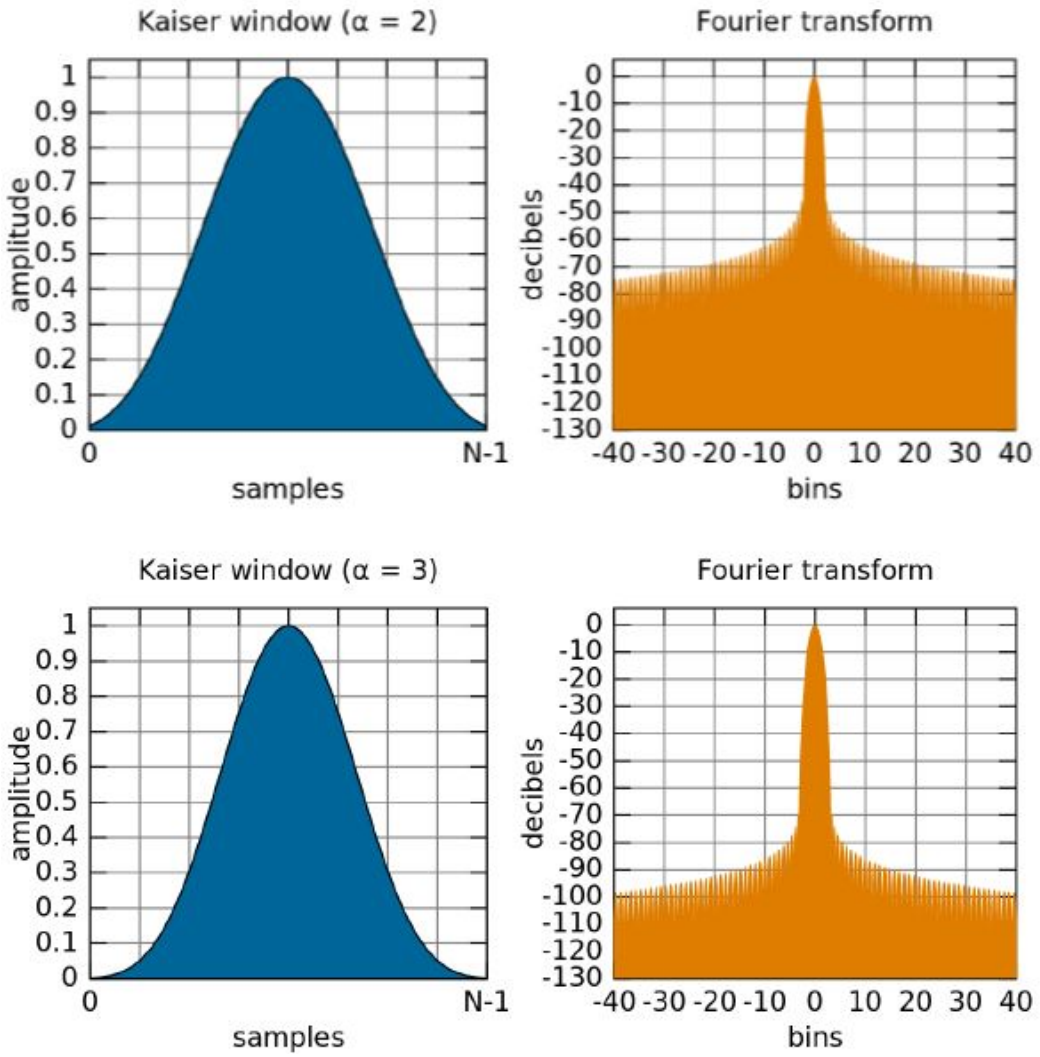


Figure 12: Kaiser Window Frequency Domain for different α parameters.

In Figure 12, it can be easily seen that in the Fourier transform of Kaiser Window as increasing of α or β parameter, main lobe width increases also side lobe heights decreases. The frequency

resolution decreases, and dynamic range increases with the main lobe width extends.

VII. Kaiser Window for different Beta parameters

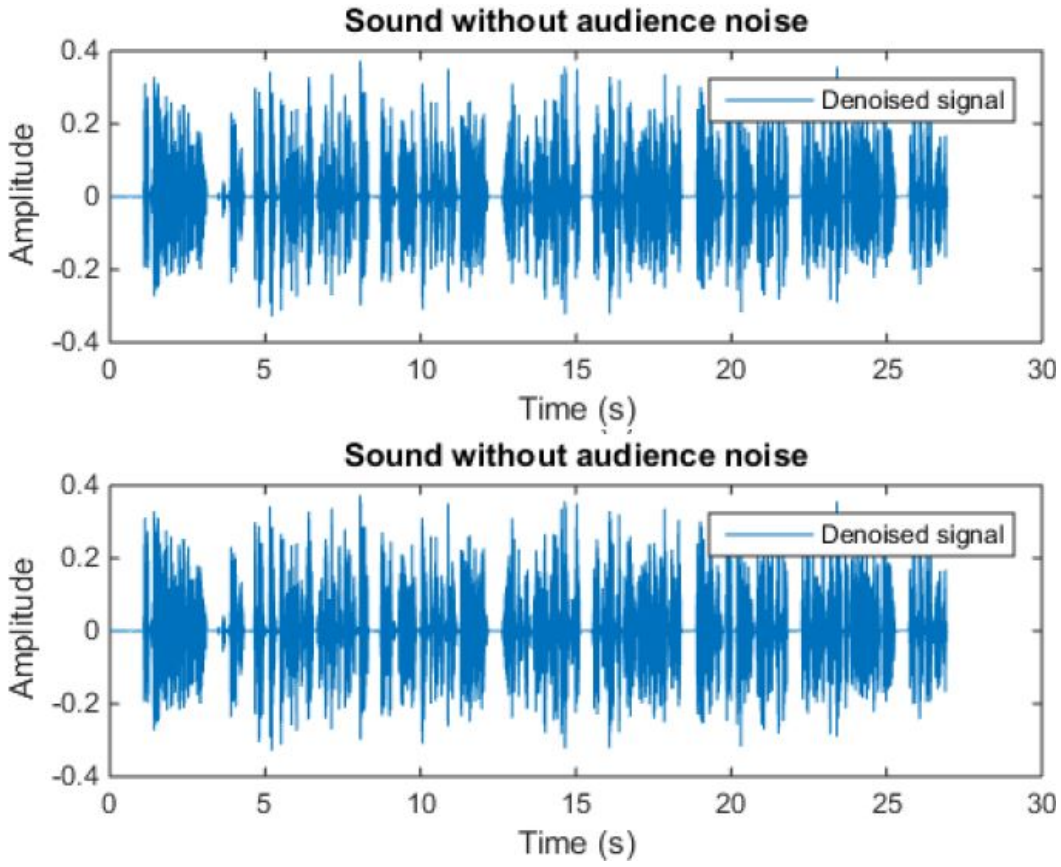


Figure 13: a) Kaiser Window β=5, b) Kaiser Window β=15

In our project, two Kaiser Window with different β parameters were compared, the results are shown in figure. For β=5, we got almost same result with Hamming window, however for β=15 the result was worst as Chebyshev Window.

V. Spectograms

A spectrogram is a visual way of representing the signal strength, or "loudness", of a signal over time at various frequencies present in a particular waveform. Not only can one see whether there is more or less energy at, but one can also see how energy levels vary over time. In other sciences spectrograms are commonly used to display frequencies of sound waves produced by humans, machinery, animals, etc., as recorded by microphones.

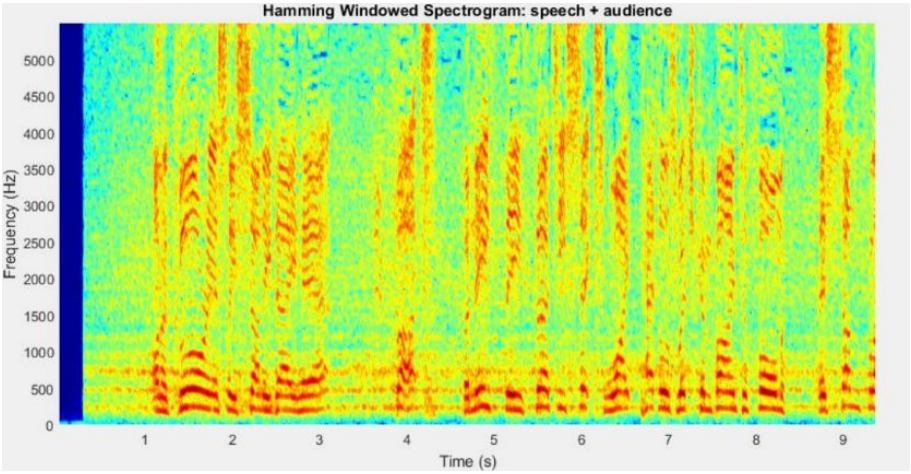
I. Spectogram Simulation Results

A spectrogram graph shows the evolution of the spectrum (the frequency contents) of a signal over time. Often, the frequency is on the vertical axis and time is on the horizontal axis. A spectrogram

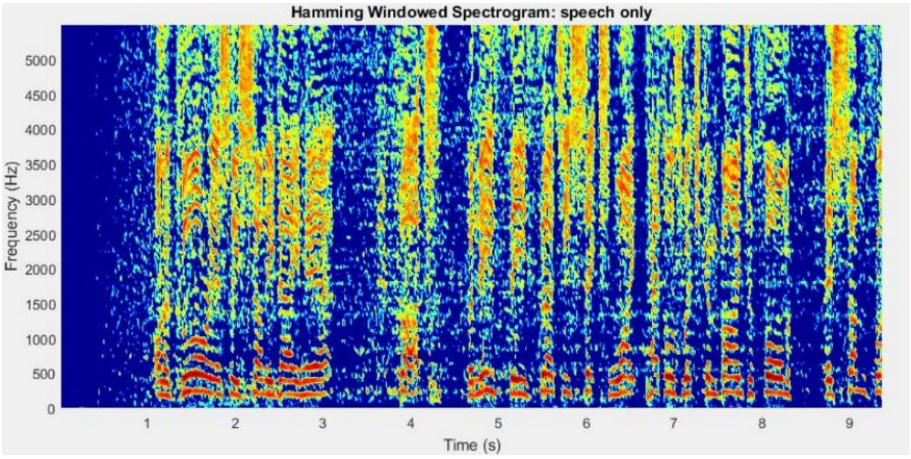
is computed by "chopping up" the signal into chunks and computing a spectrum for each of those. These different spectra are then put next to each other (as vertical lines) to form a 2D image.

In our project, we analyzed different types of windows spectogram results, and spectogram obtained for both Speech+Noise signal and Speech-only signal.

Here, spectogram obtained for Speech + Noise for the Hamming window;



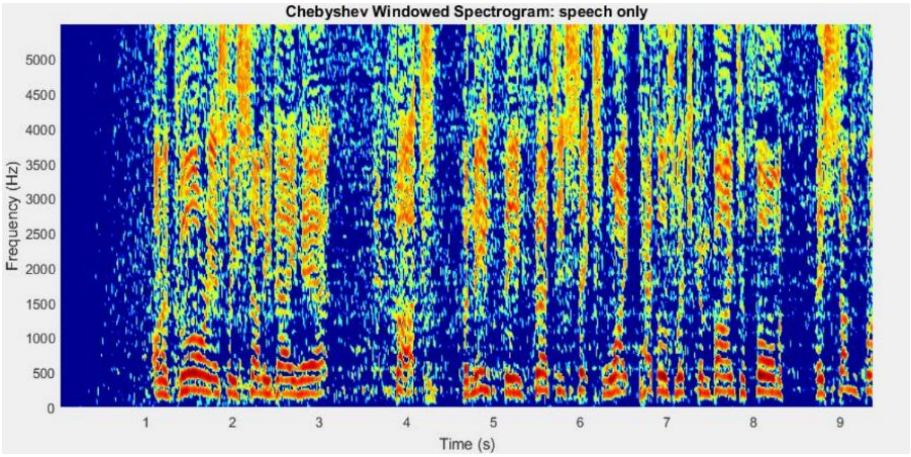
And here, spectogram after the noise removal from the speech signal obtained (speech-only signal) for the Hamming Window;



It can be easily seen from the graph above that signal frequency bands can be easily allocated (narrow) and also we have time-knowledge of the signal, so Hamming window is qualified windowing technique for this project.

Hamming Window is the optimum windowing technique for our speech signal. But also another windowing techniques applied to the signal and different results obtained from them. Chebyshev window was one of them, and performs quite inefficient and poor compared to the other types of windowing methods. Here, we used Chebyshev Window spectogram in order to show the difference between Chebyshev and Hamming Window spectograms.

Spectogram after the noise removal from the speech signal (which becomes speech-only signal) for the Chebyshev Window is shown below;



It can be seen from the graph above that signal frequency bands are wider than the Hamming window and can not be allocated easily and therefore, frequency resolution is inefficient. So, we can say that Hamming Window outperforms Chebyshev Window.

Another windowing technique that applied to the signal was Kaiser Window. Here, two spectograms of the Kaiser Window which has 2 different β parameters are shown below;

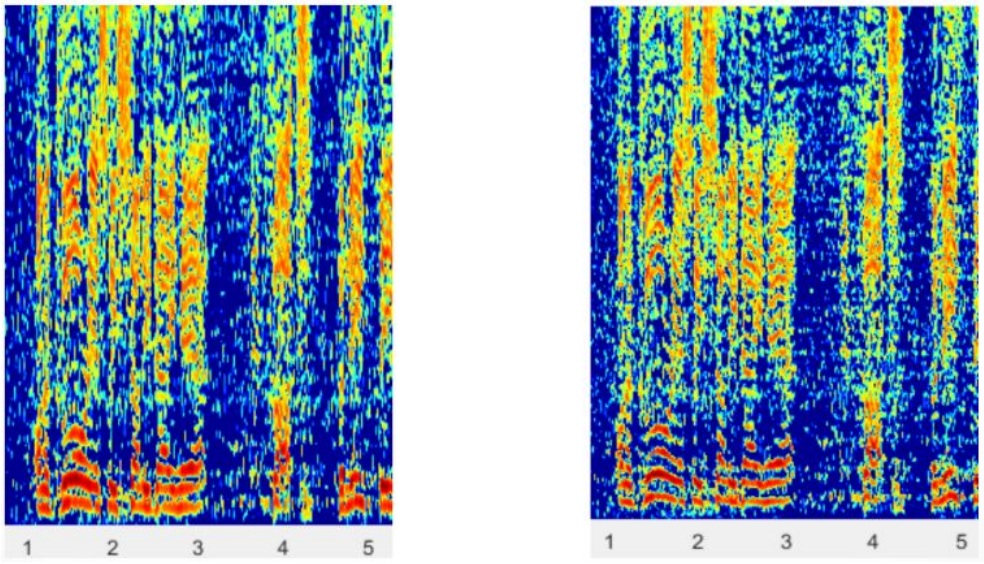


Figure 14: Kaiser Windowed signal spectograms, Left one is for β=15, Right one is for β=5

β=5 is the optimum value for the Kaiser Window for our project. If we increase or decrease this value we get worse results.

Hamming Window, Chebyshev Window and Kaiser Window spectogram results shown in this section. To sum up, Hamming Window performs best among them. And also it is observed that, Chebyshev Window's results are quite poor compared to Hamming Window. On the other hand, Kaiser Window has a special β parameter which has a optimum value of 5, performs nearly as good as Hamming Window, but increasing or decreasing this parameter from 5 ends up getting worse results like Chebyshev Window.

VI. Non Diagonal Estimation

Non-diagonal Estimation has fixed parameters that are chosen empirically. For audio signal denoising, we describe an adaptive block thresholding non-diagonal estimator that automatically adjusts all parameters. It relies on the ability to compute an estimate of the risk, with no prior stochastic audio signal model, which makes this approach particularly robust.

A time-frequency block thresholding estimator regularizes power subtraction estimation by calculating a single attenuation factor over time-frequency blocks;

$$\hat{f}[n] = \sum_{i=1}^{I} \sum_{(l,k)\in B_i} a_i Y[l,k] g_{l,k}[n] \tag{10}$$

In this equation, B_i are blocks, f is the signal estimator, (calculated from noisy data Y) with a attenuation factor α_i.

It is verified that the upper bound is minimized by choosing attenuation factor as;

$$a_i = \left(1 - \frac{\lambda}{\hat{\xi}_i + 1}\right)_+ \tag{11}$$

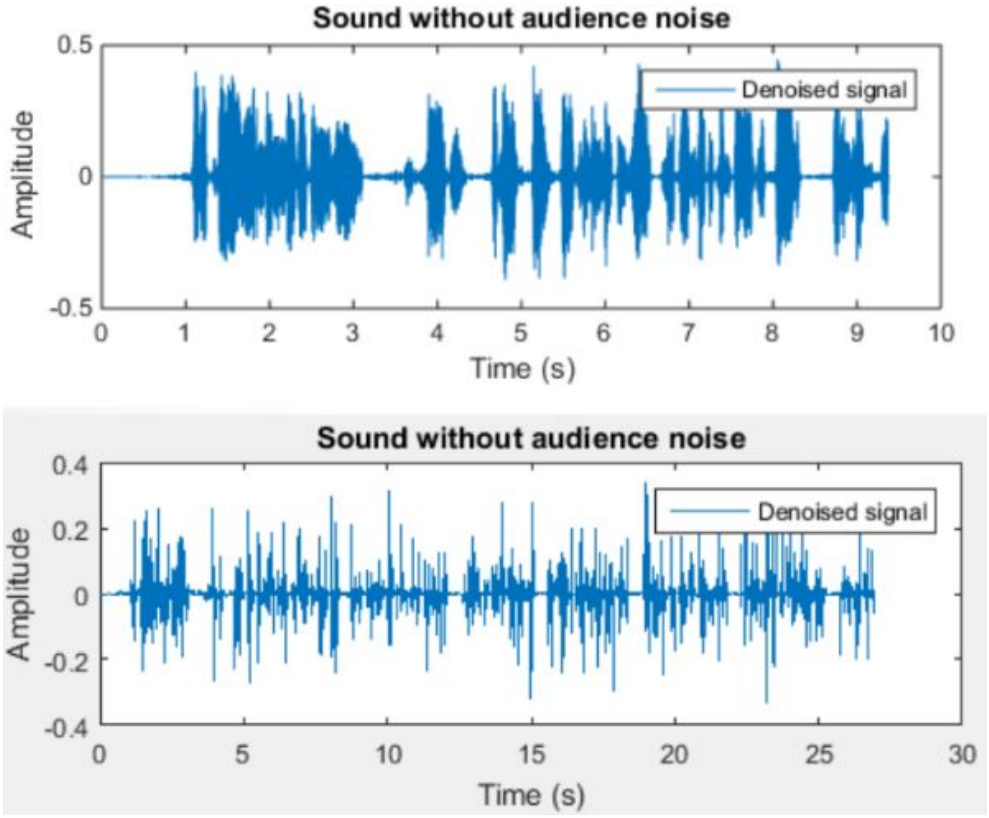


Figure 15: Non-Diagonal Estimation Results, First one is for Hamming Window, Second one is for Chebyshev Window

Lambda is an over-subtraction factor to compensate variation of noise amplitude. ξ_i is the average priori SNR in B_i. A block thresholding estimator can thus be interpreted as a non-diagonal

estimator derived from averaged SNR estimations over blocks. Each attenuation factor is calculated from all coefficients in each block, which regularizes the time-frequency coefficient estimation. Non-diagonal block thresholding attenuation factors are much more regular than the diagonal power subtraction attenuation factors.

It can be easily seen from the Figure 15 that, like diagonal method, Hamming window outperforms Chebyshev window in non-diagonal estimation.

Finally, Spectogram graph results of noisy and speech-only signal comparison using Hamming window in non-diagonal estimation is shown below;

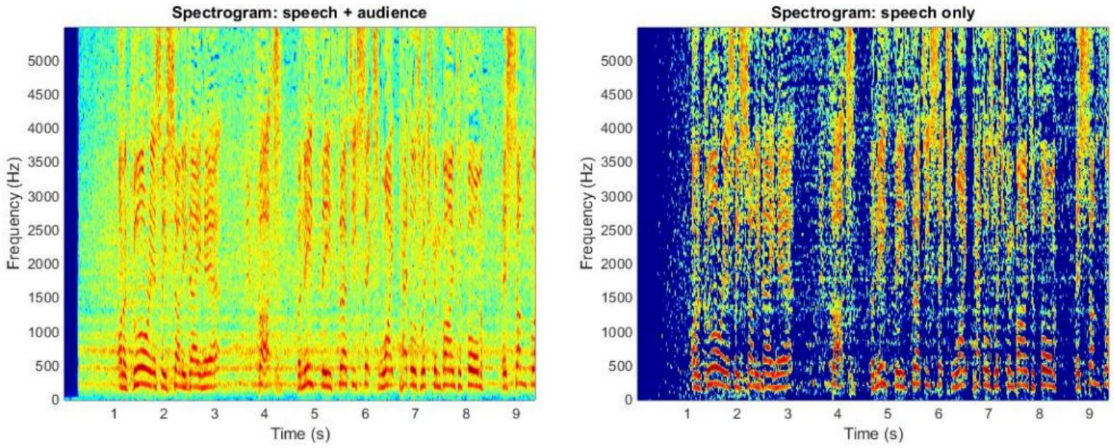


Figure 16: Non-Diagonal Estimation Spectogram Results for Hamming Window

VII. Conclusions

In this project, sound signals noise is extracted and speech only signal obtained by using Hamming, Chebyshev and Kaiser windowing techniques. In addition to these windowing techniques, sound extraction methods used to get speech-only signal. This methods were based on our reference and were used respectively; Noise Spectrum Extraction, SNR estimation and Wiener attenuation rule. After the process, signal noise completely removed and desired speech-only signal obtained successfully. Compared to the other windowing techniques, Hamming Window and Kaiser Window with β=5 parameter performs best to obtain desired signal.

References

[1] 'Audio Denoising by Time-Frequency Block Thresholding', Guoshen Yu, Stephane Mallat, Fellow, IEEE, and Emmanuel Bacry, - *IEEE Transactions On Signal Processing, Vol. 56, No.5, May 2008*.