EESC 6367 Applied Digital Signal Processing

Spring 2023

Real-Time Implementation and Evaluation of Adaptive Dynamic Range Optimization (ADRO) for Hearing Enhancement

Erik Jonsson School of Engineering and Computer Science

Final Project Report

Comet Creed

This creed was voted on by the UT Dallas student body in 2014. It is a standard that Comets choose to live by and encourage others to do the same:

"As a Comet, I pledge honesty, integrity, and service in all that I do."

Submitted by

Raman Mishra
RSM200000

Table of Contents

Introduction:

Millions of individuals throughout the world suffer from hearing loss, which is a prevalent condition that frequently causes communication problems, reduced pleasure of entertainment, and restricted involvement in social activities. The Adaptive Dynamic Range Optimization (ADRO) algorithm has gained popularity as a method to enhance hearing perception in recent years. A digital signal processing system called ADRO automatically modifies the dynamic range of audio signals in real-time so that those with hearing loss can have a consistent and satisfying listening experience.

A growing number of smartphone apps utilizing the ADRO algorithm for hearing enhancement are becoming increasingly popular due to their convenience and affordability, making it easier for individuals with hearing impairments to improve their quality of life without buying expensive hearing aids or cochlear implants.

A real-time smartphone app that uses the ADRO algorithm for hearing enhancement is what the Real-Time Implementation and Evaluation of Adaptive Dynamic Range Optimization (ADRO) for Hearing Enhancement project wants to create. In this project, the ADRO algorithm was implemented and evaluated while considering a number of different factors, including the audibility target, and comfort target level. Because of the app's user-friendly layout, those who have hearing loss may adjust the settings to their tastes.

Hearing-impaired people are likely to benefit from this project because it can improve their quality of life. This project creates a real-time ADRO-based android application, which offers persons with hearing impairment a cheap and practical way to enhance their hearing perception and quality of life. By offering information on the ADRO algorithm's efficacy for improving hearing and the best settings for its application, this project can also advance the area of audiology.

ADRO Algorithm:

An effective method for helping people with hearing impairments experience sounds better is the Adaptive Dynamic Range Optimization (ADRO) algorithm. This digital signal processing technique modifies the dynamic range of a streaming audio file in real time according to the signal's loudness level.

The division of the acquired audio signal into a set of frequency bins or bands of 32, is the first step in the ADRO algorithm's operation. Then, using the gain values set forth by the Gain Computation module, the signal in each frequency band is amplified. The gain changes made by the Gain Computation module optimize the signal for the user by estimating the high, mid, and low percentiles of the sound pressure level (SPL) of the incoming sounds.

To make adjustments based on individual needs, the audibility target, and comfort target level of a person are specified in the Map module. This allows the ADRO algorithm to adjust the gain values applied to each frequency band of the audio signal, providing a personalized listening experience.

Additionally, the Max Gain (MG) and Maximum Power Output (MPO) parameters are used to limit the gain value and the output level, respectively, for audio safety reasons. This ensures that the ADRO algorithm is not only effective but also safe for the user. Figure 1 below provides the ADRO Gain computation module.

Overall, by offering a constant and ideal listening experience, the ADRO algorithm has enormous potential to enhance the quality of life for those with hearing loss. For those who might not have access to costly hearing aids or cochlear implants, it is now a practical and economical alternative because to its real-time application via smartphone applications. The ADRO algorithm has been extensively researched for its efficacy as a method for improving hearing, and it has been found to be so.

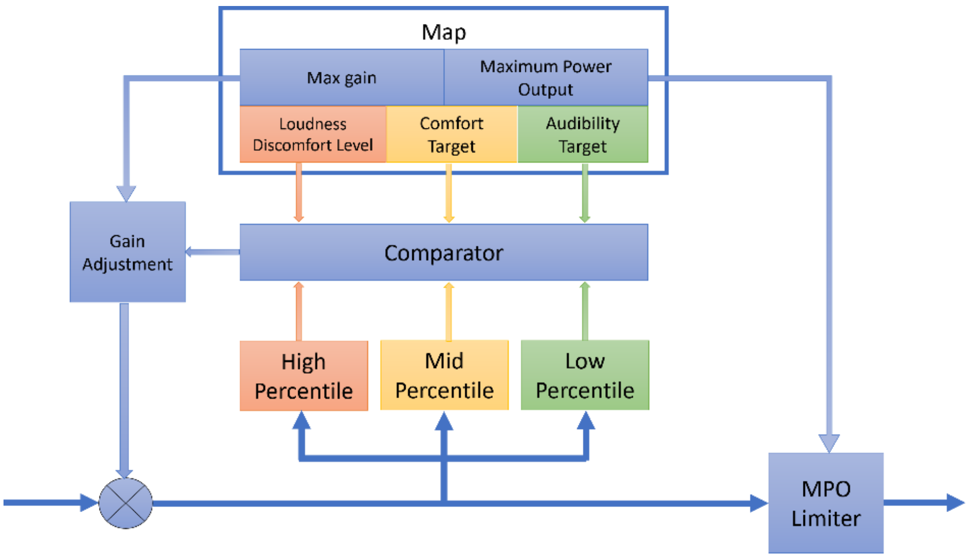

Fig 1: ADRO Gain Computation module*

*Diagram used from the project description.

Steps taken for the Project:

Following were the steps taken to complete the project:

1. **Use of Hearing Test app from the Google Play store, to determine my Audiogram and Comfort Target:**

What is an Audiogram?
A visual representation of a person's hearing capacity called an audiogram is usually produced during a hearing exam. It typically appears as a graphic and displays the quietest noises a person can hear at various frequencies (or pitches). The audiogram measures frequency in Hertz and auditory capacity in decibels (dB). (Hz). The horizontal axis depicts frequency or tone in Hertz, while the vertical axis measures sound strength or loudness in decibels. For detecting hearing loss and figuring out the kind, severity, and pattern of hearing loss, audiograms are helpful. They can also be used to choose the proper hearing aids or other helpful devices and to track changes in hearing over time.
Below is my Audiogram which is made with the help of the Hearing App available on the Google Play.

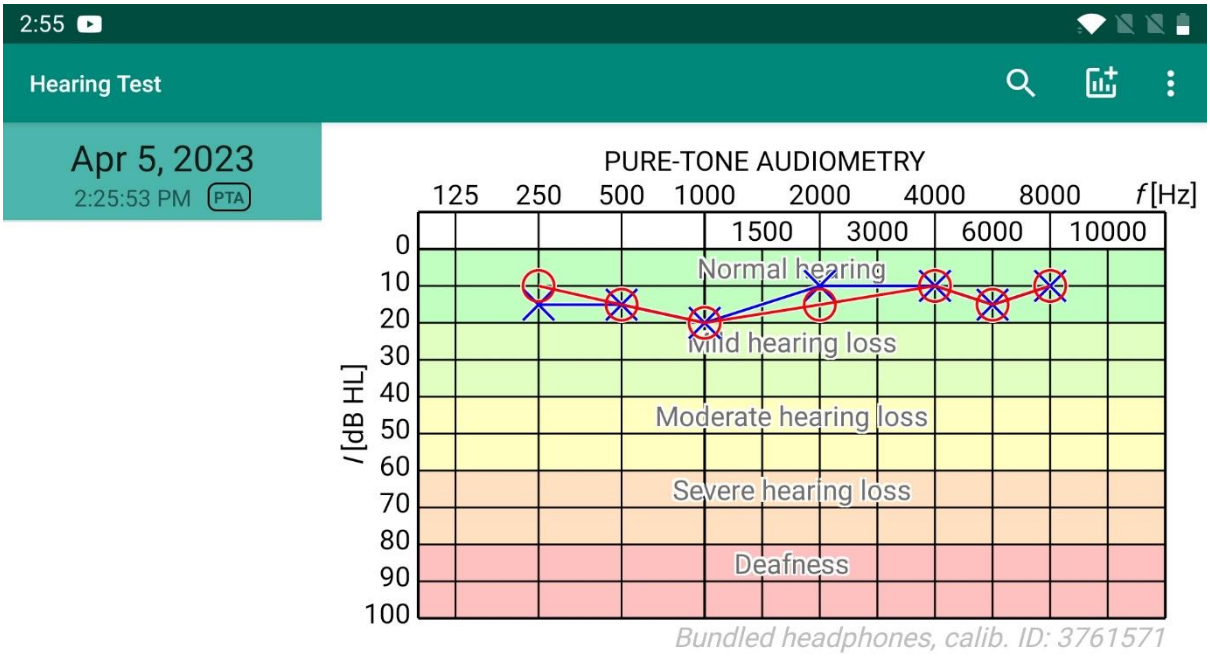

Fig 2: Audiogram

What is the Comfort target?

The comfort target in hearing refers to the loudness level of noises that a person can comfortably listen to without experiencing pain or discomfort. This is crucial because harsh noises can harm the sensitive inner ear tissues and cause hearing loss. The target degree of comfort can change based on a person's hearing capacity, age, and other variables. Usually, it is determined through the use of a method known as decibel discomfort level (LDL) assessment during a hearing exam. The subject listens to noises at various loudness levels and signals when the sound becomes uncomfortable during this evaluation.

Below is my Comfort Target which is made with the help of the Hearing App available on the Google Play.

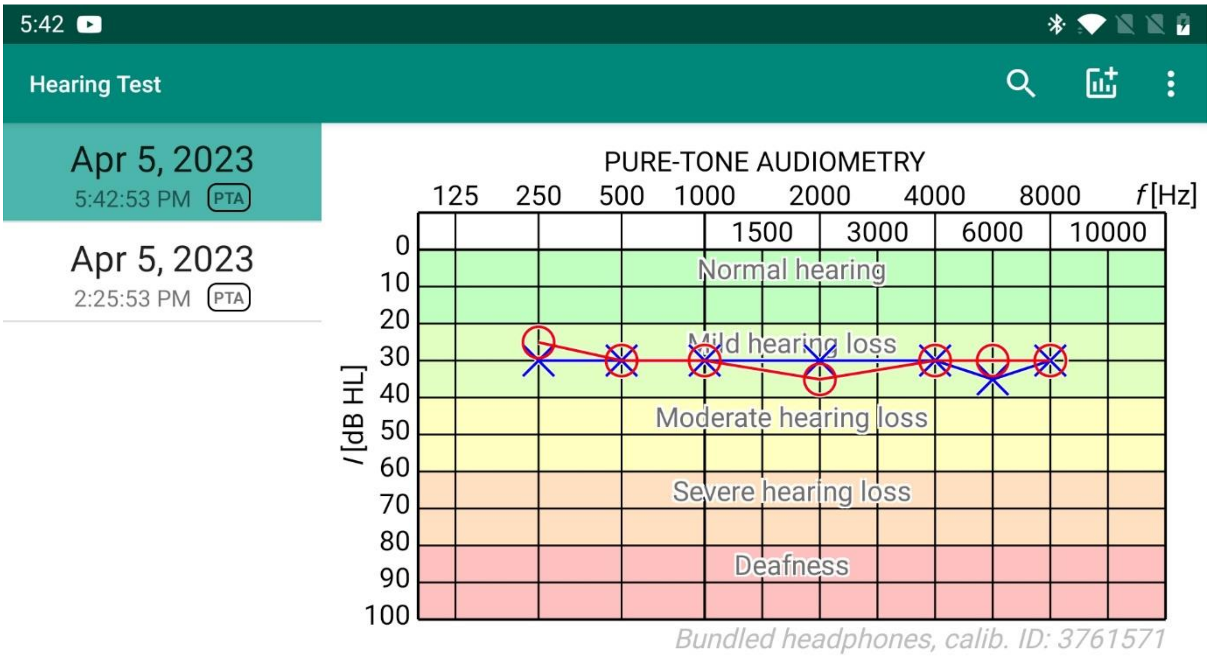

Fig 3: Comfort Target

In this step, I calculated the Audiogram data and Comfort target using the BARELY AUDIBLE button available in the App. This helped me understand my own hearing ability. As we can my Audiometry for Audiogram ranges between 10-20 dB HL and Comfort target between 25-35 dB HL.

2. Collection of audio file from real-world environment:

During this step, I recorded two speeches - one in soft level and one in loud level - each lasting for two minutes. These audio files, named "Soft Speech" and "Loud Speech" respectively, can be found in the Audio_Samples folder. In addition, I recorded three other sounds, namely "Babble," "Machinery," and "Music," each also lasting for two minutes. These files were also saved in the Audio_Samples folder. In total, I recorded five audio files.

To prepare these files for further analysis, I first converted them to .wav format using Matlab. Then, I used the resample() function in Matlab to resample the audio files to 16kHz. The resampled files were named in the same manner as the original files, but with the addition of "r_" to avoid confusion.

Editor - C:\Users\Raman\Desktop\EESC6367\Project\Audio_Resample.m

Audio_Resample.m

```
% Defining the path for audio sample
addpath('C:\Users\Raman\Desktop\EESC6367\Project\Audio_Samples');
% Loading the original audio file
[x, Fs] = audioread('Music.wav');

% Resampling the audio to 16kHz using the resample() function
Fs_new = 16000; % The target sampling rate
x_resampled = resample(x, Fs_new, Fs);

% Saving the resampled audio file
audiowrite('resampled_Music.wav', x_resampled, Fs_new);
```

Fig 4: Screenshot for resampling the audio files.

Next, I used the add_noise_v3.m Matlab file to mix the speech and noise signals together at high and low SNR levels. Concatenation was done which resulted in 3 additional audio files for testing. After this total 15 audio files were generated which were then placed in the folder named as Audio. Following were the name of the audio file:

i. Mishra_12Signals_Babble_CONCATENATED
ii. Mishra_12Signals_Machinery_CONCATENATED
iii. Mishra_12Signals_Music_CONCATENATED
iv. Mishra_Loud_Babble_High
v. Mishra_Loud_Babble_Low
vi. Mishra_Loud_Machinery_High
vii. Mishra_Loud_Machinery_Low
viii. Mishra_Loud_Music_High
ix. Mishra_Loud_Music_Low
x. Mishra_Soft_Babble_High
xi. Mishra_Soft_Babble_Low

xii. Mishra_Soft_Machinery_High
xiii. Mishra_Soft_Machinery_Low
xiv. Mishra_Soft_Music_High
xv. Mishra_Soft_Music_Low

Figure 5 will help to demonstrate this step of mixing of the audio file in an easy way.

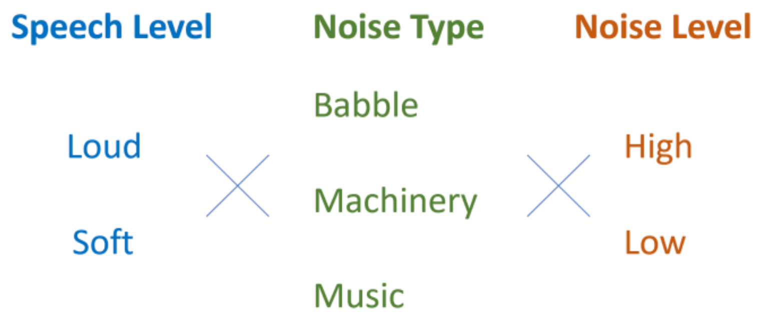

Fig 5. Audio recordings to collect.

3. Complete the percentile estimation module in adro_main.cpp:

The pseudo code provided in the description is a module within the ADRO algorithm that updates the high and mid percentiles based on certain conditions. I tried understanding the pseudo code and wrote the complete module accordingly.
In the code, the high percentile module first checks if the difference between the current high percentile value and a down step value is greater than the input level of sound. If this condition is true, the high percentile value is decreased by the down step value. Otherwise, the high percentile value is increased by an up-step value. After that, if the high percentile value is greater than the input level of sound, it is set to the input level.
The mid percentile module works in exactly the same way as the high percentile module, except that it uses the mid percentile down and up step values instead of the high percentile ones.
Overall, these percentiles are important parameters in the ADRO algorithm, and they are used to adjust the gain of the hearing aid in order to provide an optimal listening experience for the user. This module provided updates these percentiles based on the input level of sound and the specific conditions of the ADRO algorithm.

4. Updating the gain module in adro_main.cpp file:

This module is a part of the ADRO algorithm and is responsible for updating the gain for a subset of bins. Following is the break-down of the steps in detail:
The algorithm first checks if the high percentile value for a given frequency bin is greater than the comfort target. If it is, then the gain for that bin is decreased by the GainDownSlew value.
If the high percentile value is not greater than the comfort target, then the algorithm checks if the Mid (30th) percentile value is less than the audibility target for that bin. If it is, then the gain for that bin is increased by the GainUpSlew value.
If neither of the above conditions is met, then the gain for that bin is increased by the GainDriftUpSlew value.
The algorithm then checks if the updated gain for that bin exceeds the maximum gain specified in the MaxGains_BinFreqs array. If it does, then the gain is capped at the maximum value.
Finally, if the MinimumGainEnabled flag is set to true and the updated gain is less than the minimum gain specified in the MinGains_BinFreqs array, then the gain is set to the minimum value.

5. Running the ADRO UTD Android Application:

Following were the steps used to run the ADRO UTD application:

i. Install the application in an Android device with the Android version above 12.
ii. After installing the application, open the ADRO UTD app.
iii. Set the preset values by inputting all the required values in the Audiogram and Comfort Target. These values were analysed and determined in the while using the hearing app.
iv. Save the preset values and proceed to run an audio sample file.
v. Initially, run the audio file without enabling the ADRO feature. During this step, observe the real-time plotting of the 4 data points.
vi. Observe that the mid percentile and high percentile plots are not overlapping with the Audiogram and comfort target, indicating a deviation from the desired values.
vii. Now enable ADRO, we observe that the mid percentile and high percentile plots are now overlapping with the Audiogram and comfort target, indicating that the ADRO algorithm is working as expected.

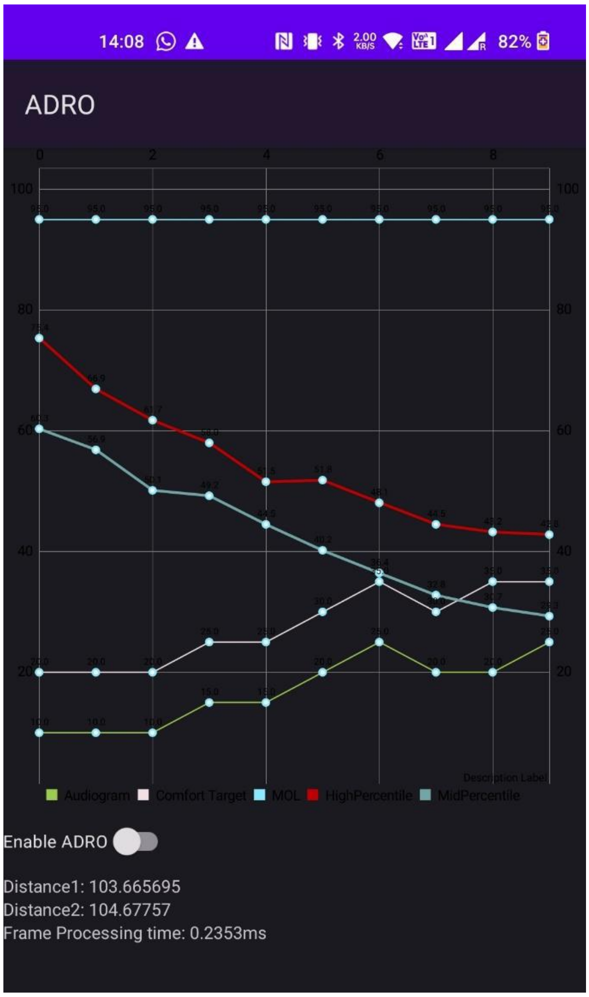

Fig 6: Screenshot of the App before enabling ADRO.

Fig 7: Screenshot of the App after enabling ADRO.

viii. Based on this observation, it can be concluded that the ADRO algorithm is running correctly in the app.

ix. To analyse our observation more thoroughly, “Enable Data Saving” switch on the app while selecting the Preset file and the audio file.

x. A text file will be generated that contains ADRO on/off status, Distance1, Distance2, time, and Gain value for 32 frequency bands will be saved at a folder named “ADRO/Data”.

xi. We will be using this data to further analyse the implementation of ADRO of our 15 audio signals.

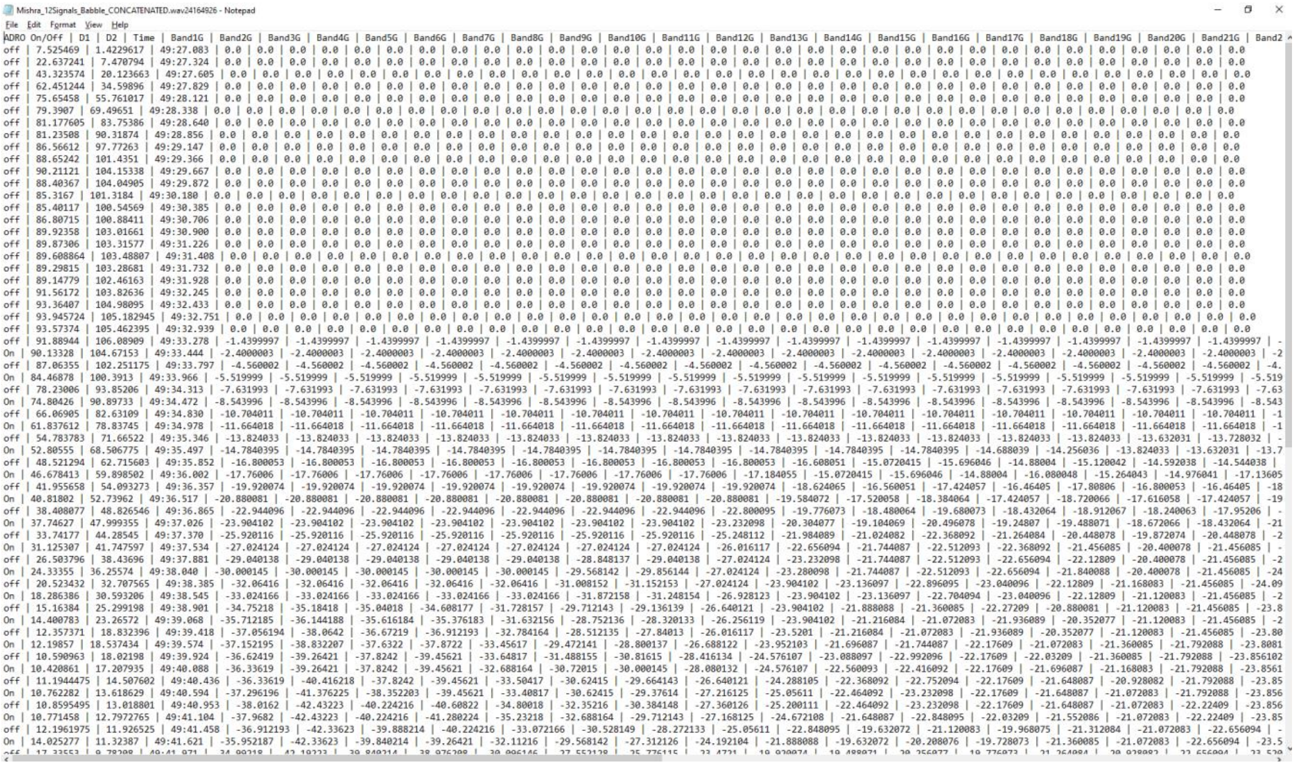

Fig: Sample text file demonstrating data post enabling data saving mode

What is the adaptive ability and amplification ability of ADRO?

The adaptive ability of ADRO refers to its capacity to modify the gain settings in real-time to enhance speech understanding and listening comfort in response to changes in the noise environment. To keep a constant overall loudness level, ADRO does this by continuously monitoring the input sound levels and automatically modifying the gain settings for each frequency band. This prevents discomfort or distortion from loud or abrupt noises and enables the user to comfortably hear speech and other critical sounds.

The amplification ability of ADRO refers to its ability to amplify sounds across a wide range of frequencies, while maintaining a natural and clear sound quality. ADRO achieves this by applying different gain levels to different frequency bands, based on the individual's hearing loss and the characteristics of the input sound. ADRO also uses a compression ratio that varies with input level, to maintain a consistent loudness level and reduce the potential for distortion or discomfort.

Overall, the adaptive and amplification abilities of ADRO work together to provide a comfortable and effective hearing experience for users, by adapting to changes in the listening environment and amplifying sounds in a natural and clear way. The performance of ADRO can be evaluated through various measures such as speech intelligibility, subjective ratings, and real-world usage data.

Observation:

For the construction of the graph:

- D1 & D2 vs time graph was constructed using excel and the data was extracted after enabling Data saving button.
- Spectrogram was constructed on Matlab using the data which was extracted after enabling Data saving button.
- Graphs for Time and Frequency domain was constructed using Matlab.

Understanding the Graph in general:

- There are two distances in Euclidean space: Distance1 (D1) denotes the distance between Comfort Target and High Percentile distribution, and Distance2 (D2) denotes the distance between Audiogram and Mid percentile distribution.
- Spectrogram is one of the useful ways to analyse gain. In a spectrogram, the colours represent the amplitude or intensity of the sound at each frequency and time point. Typically, brighter colours like yellow or white represent higher amplitudes or intensities, while darker colours like blue or black represent lower amplitudes or intensities.
- In the time domain, noisy speech refers to speech signals that are contaminated by one or more sources of noise, such as ambient sounds, electrical interference, or reverberation.
- In the frequency domain, noisy speech refers to speech signals that have a non-uniform distribution of energy across frequency bands, due to the presence of noise or other interference.

I'll be providing audio file-wise observation, following are the audio files:

1. Babble Concatenated
2. Machinery Concatenated
3. Music Concatenated

Similary, we can analyse for other 12 audio files.

1. Babble Concatenated:

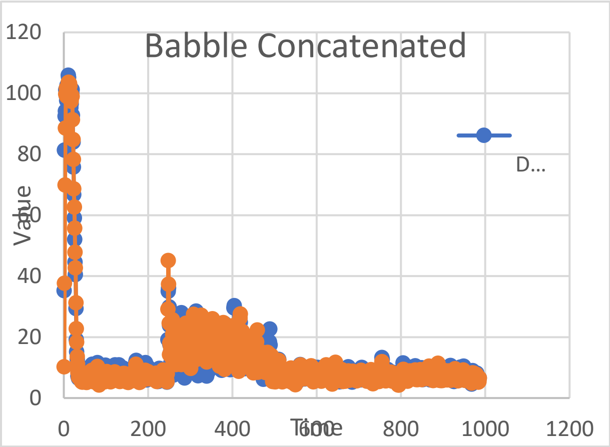

Fig 8: D1 & D2 vs time

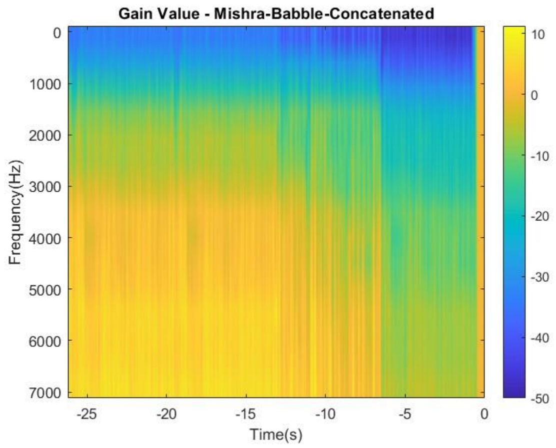

Fig 9: Spectrogram

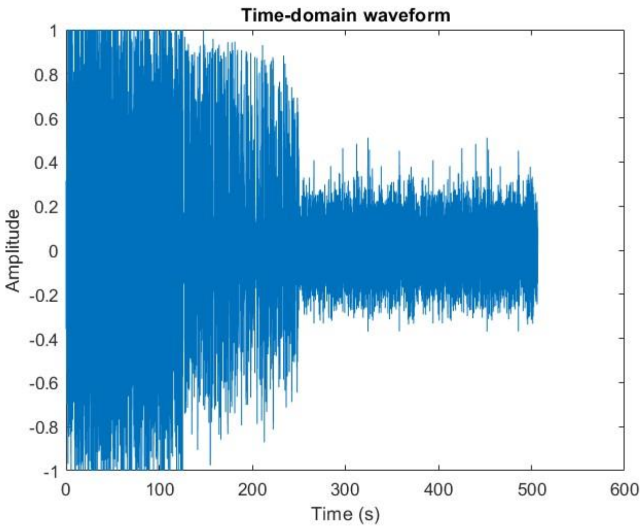

Fig 10: Noisy Speech in time domain

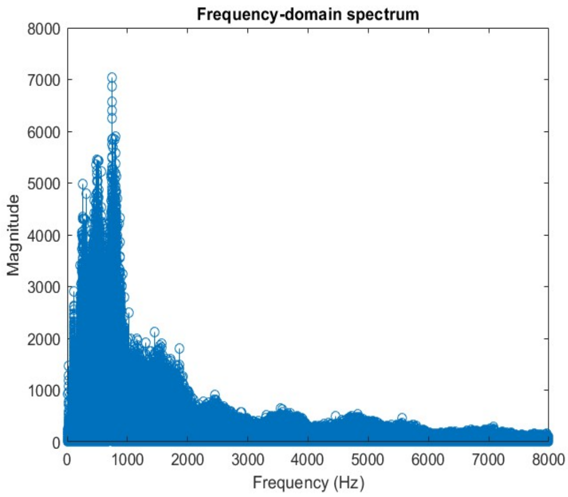

Fig 11: Noisy Speech in frequency domain

- For D1 & D2 vs time graph, we can see the difference between D1 and D2 is quite high before enabling ADRO. At around T= 15s the ADRO was enabled, and we can see the difference between D1 and D2 decreasing drastically and becoming 0 sometimes. But the moment ADRO was turned off we can again see the increase in the difference which shows the working of ADRO algorithm.
- Further we can see, the noise level also decreases after enabling ADRO which applies more gain to the audio. When audio is soft, we can see the decrease in noise as well.
- In the spectrogram, blue color represents the min gain and yellow shows the max gain applied. Since the audio was very noisy in the start min gain was applied by the ADRO algorithm, so the gain was approximately -30dB.

2. Machinery Concatenated:

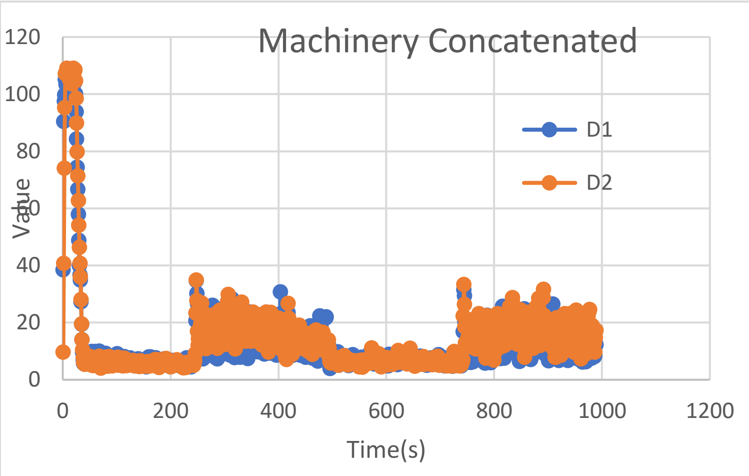

Fig 12: D1 & D2 vs time

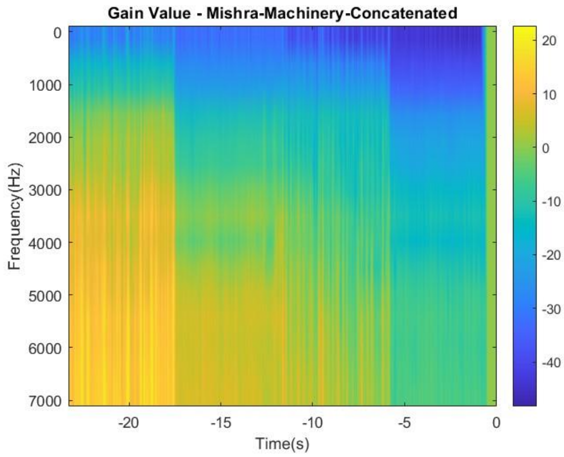

Fig 13: Spectrogram

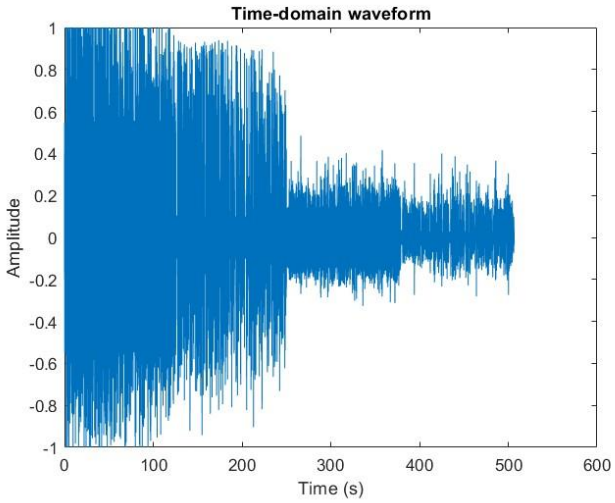

Fig 14: Noisy Speech in time domain

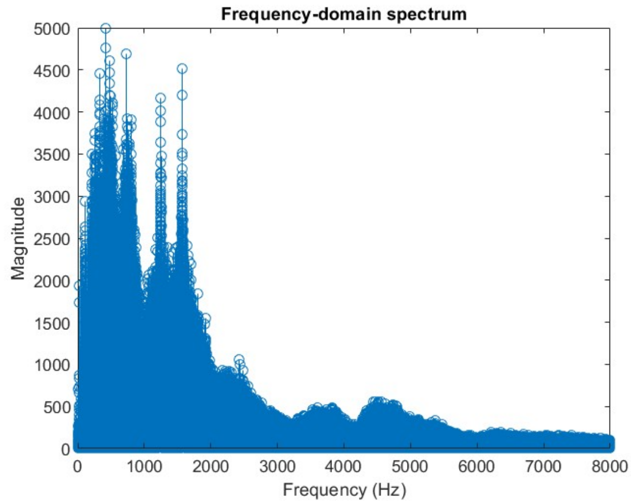

Fig 15: Noisy Speech in frequency domain

- For D1 & D2 vs time graph, we can see the difference between D1 and D2 is quite high before enabling ADRO. At around T= 20s the ADRO was enabled, and we can see the difference between D1 and D2 decreasing drastically and becoming 0 sometimes. But the moment ADRO was turned off we can again see the increase in the difference which shows the working of ADRO algorithm.
- Further we can see, the noise level also decreases after enabling ADRO which applies more gain to the audio. When audio is soft, we can see the decrease in noise as well.
- In the spectrogram, blue color represents the min gain and yellow shows the max gain applied. Since the audio was very noisy in the start min gain was applied by the ADRO algorithm, so the gain was approximately -30dB.

3. Music Concatenated:

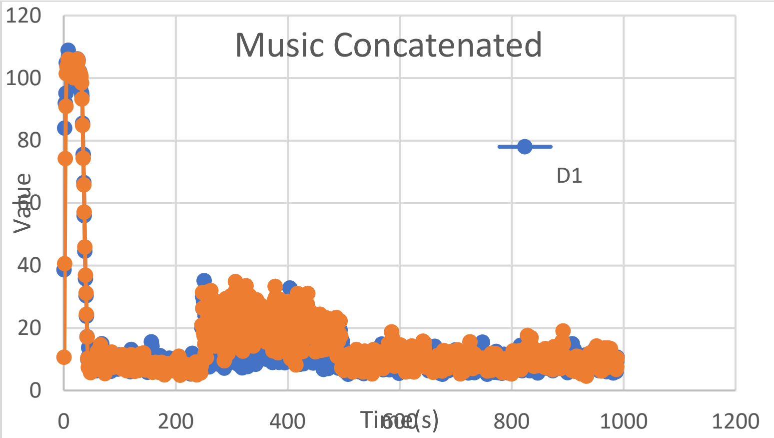

Fig 16: D1 & D2 vs time

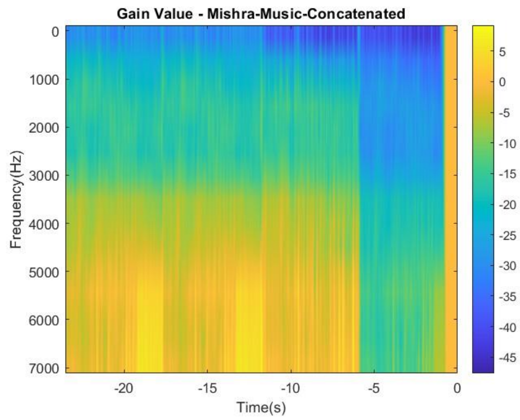

Fig 17: Spectrogram

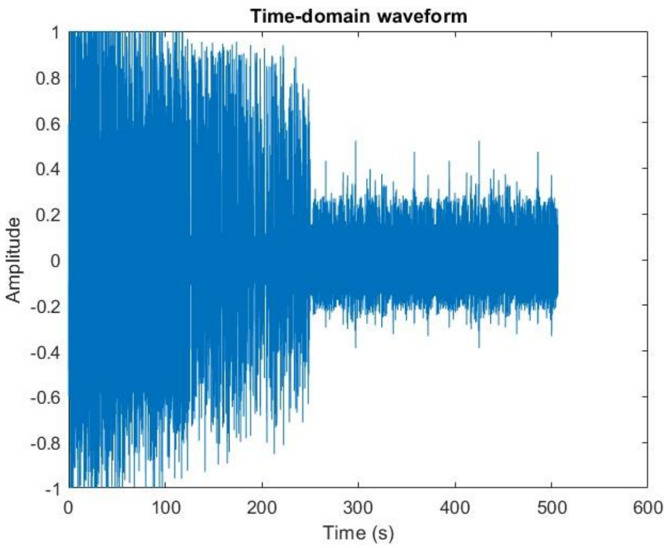

Fig 18: Noisy Speech in time domain

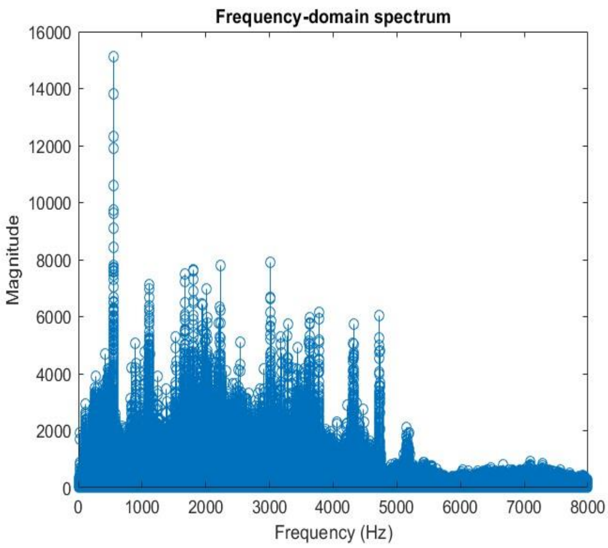

Fig 19: Noisy Speech in frequency domain

- For D1 & D2 vs time graph, we can see the difference between D1 and D2 is quite high before enabling ADRO. At around T= 27s the ADRO was enabled, and we can see the difference between D1 and D2 decreasing drastically and becoming 0 sometimes. But the moment ADRO was turned off we can again see the increase in the difference which shows the working of ADRO algorithm.
- Further we can see, the noise level also decreases after enabling ADRO which applies more gain to the audio. When audio is soft, we can see the decrease in noise as well.
- In the spectrogram, blue color represents the min gain and yellow shows the max gain applied. Since the audio was very noisy in the start min gain was applied by the ADRO algorithm, so the gain was approximately -25dB.

4. Loud Babble High:

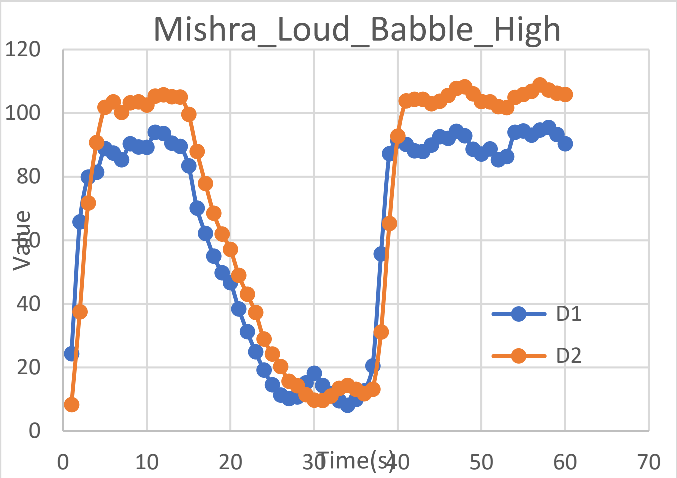

Fig 20: D1 & D2 vs time

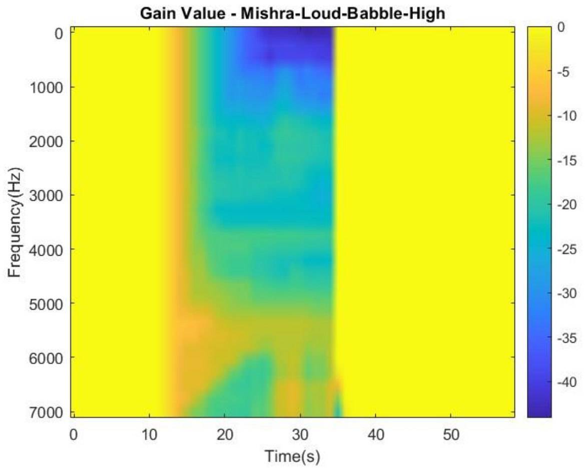

Fig 21: Noisy Speech in time domain

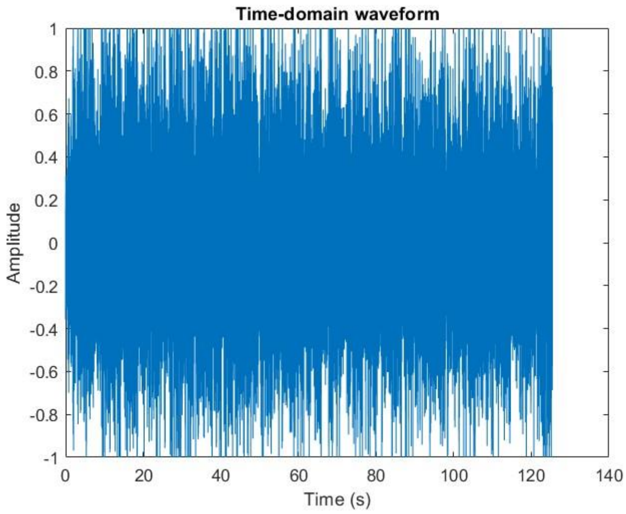

Fig 22: Noisy Speech in time domain

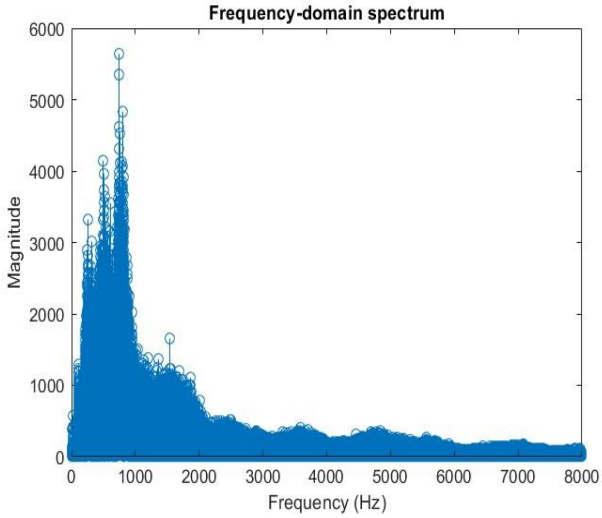

Fig 23: Noisy Speech in frequency domain

•

5. Loud Babble Low:

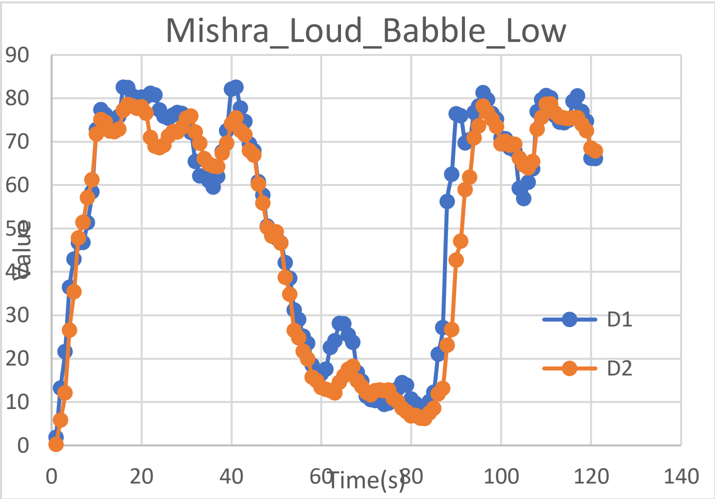

Fig 24: D1 & D2 vs time

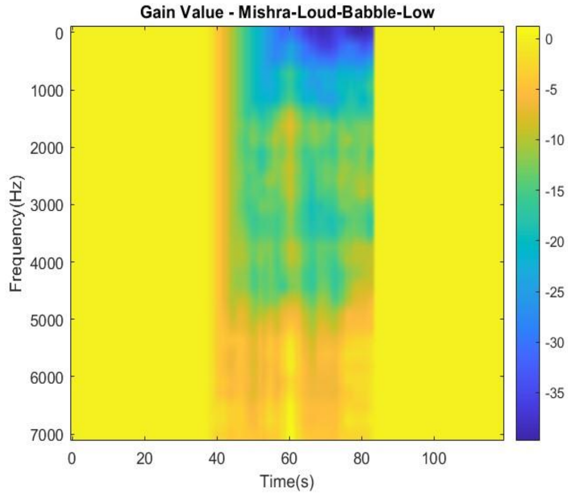

Fig 25: Noisy Speech in time domain

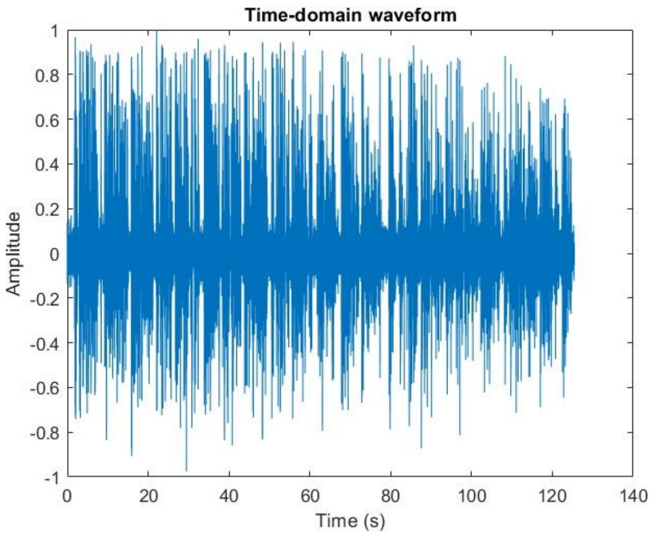

Fig 26: Noisy Speech in time domain

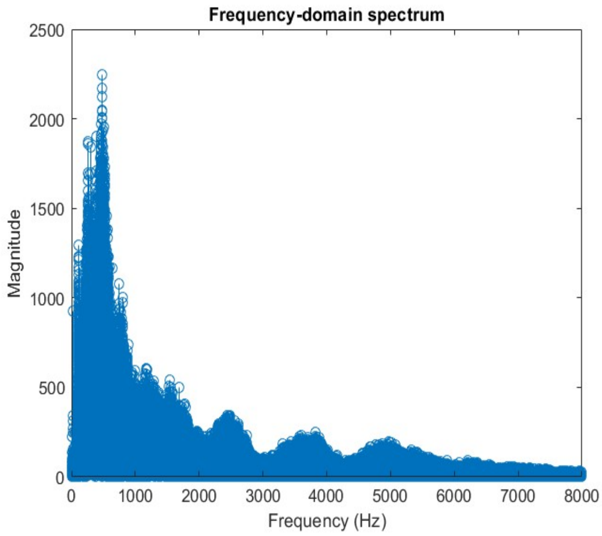

Fig 27: Noisy Speech in frequency domain

6. Loud Machinery High:

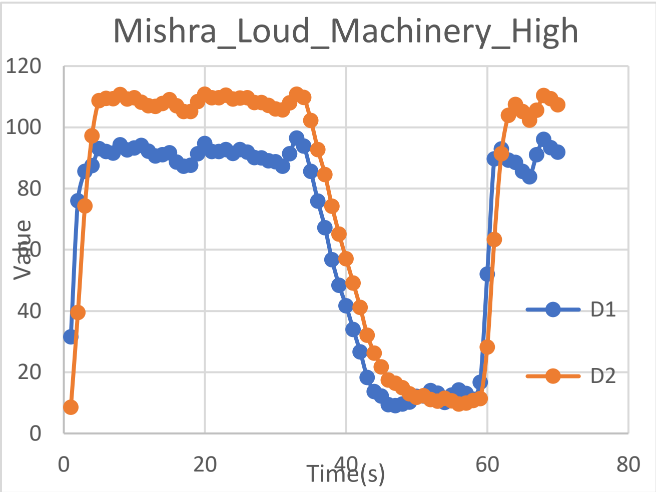

Fig 28: D1 & D2 vs time

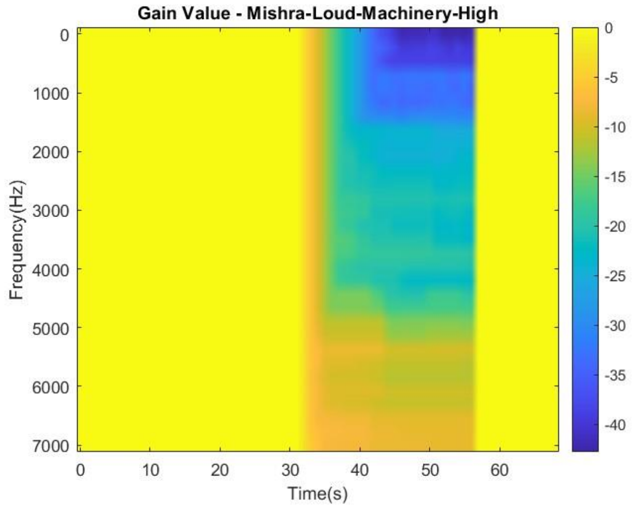

Fig 29: Noisy Speech in time domain

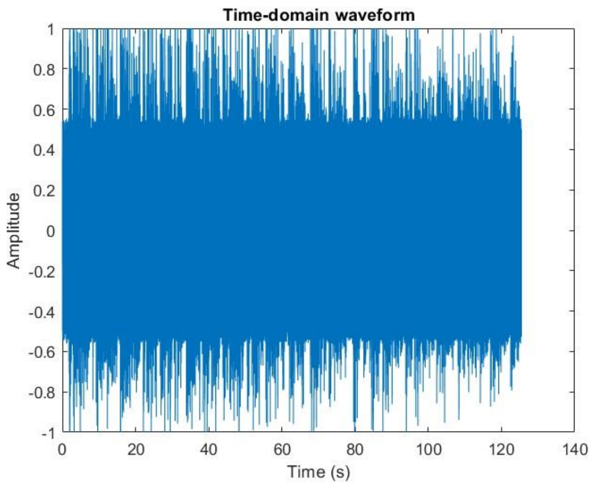

Fig 30: Noisy Speech in time domain

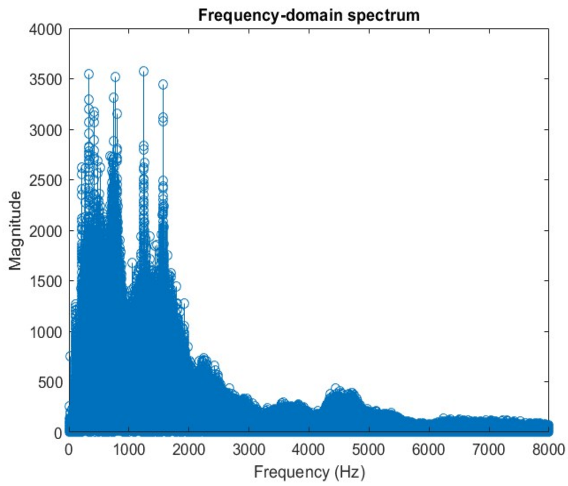

Fig 31: Noisy Speech in frequency domain

7. Loud Machinery Low:

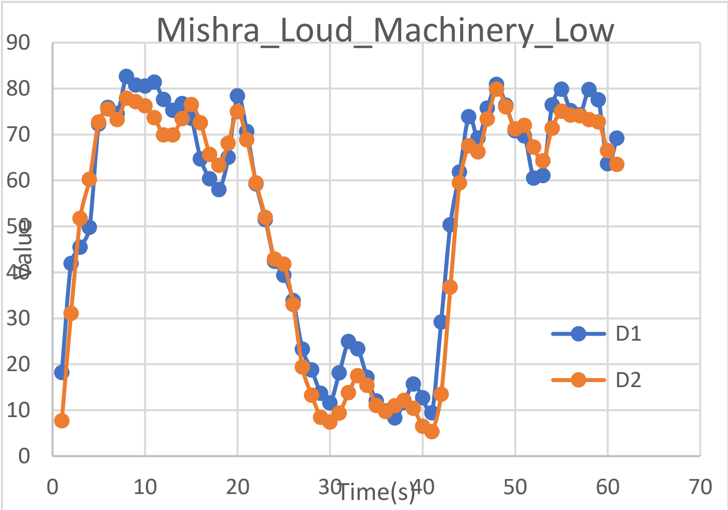

Fig 32: D1 & D2 vs time

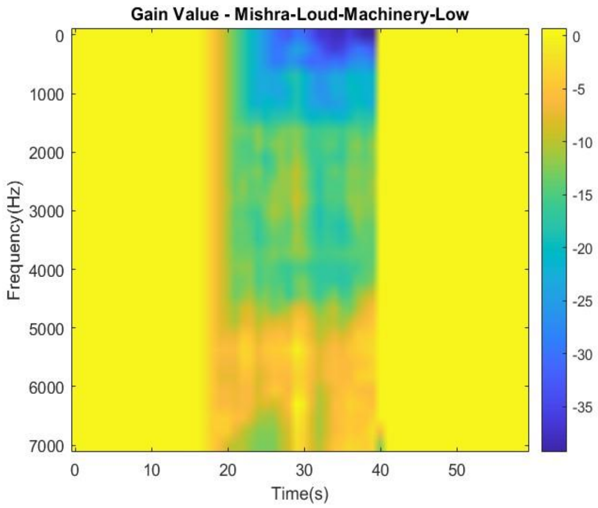

Fig 33: Noisy Speech in time domain

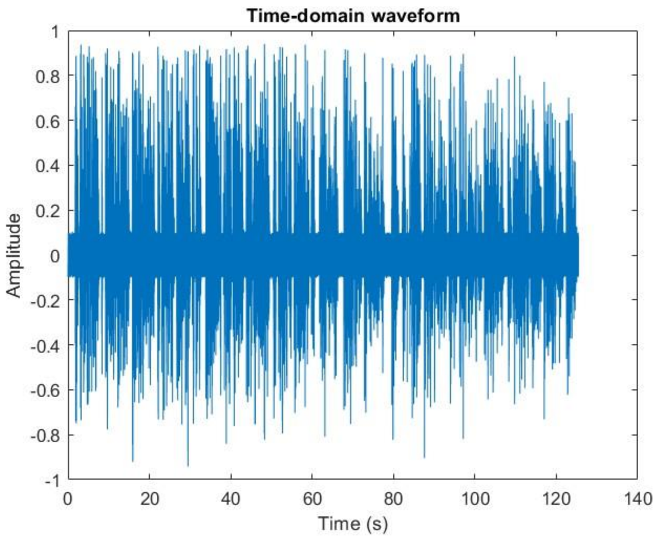

Fig 34: Noisy Speech in time domain

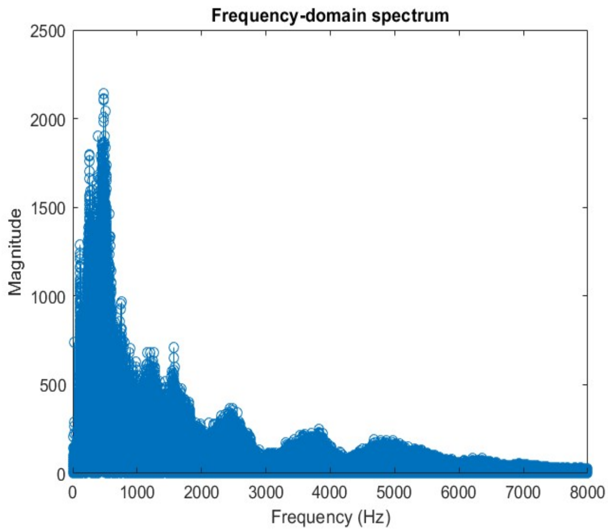

Fig 35: Noisy Speech in frequency domain

8. Loud Music High:

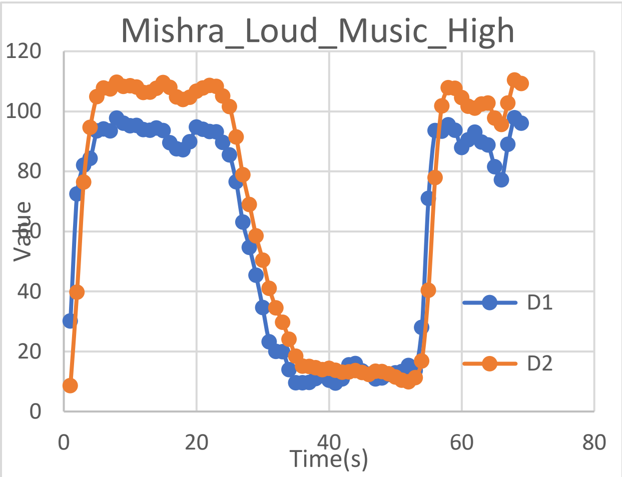

Fig 36: D1 & D2 vs time

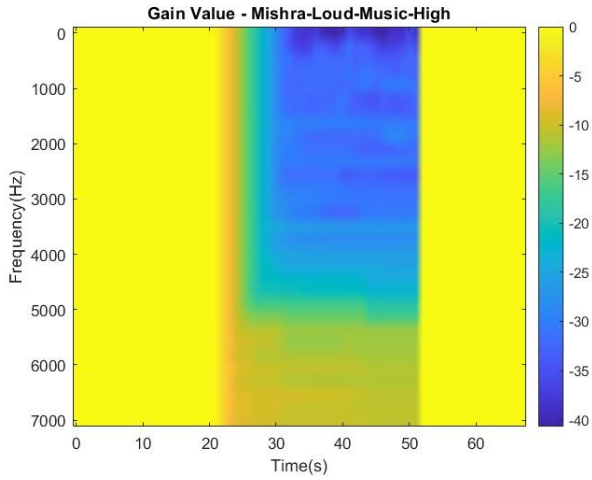

Fig 37: Noisy Speech in time domain

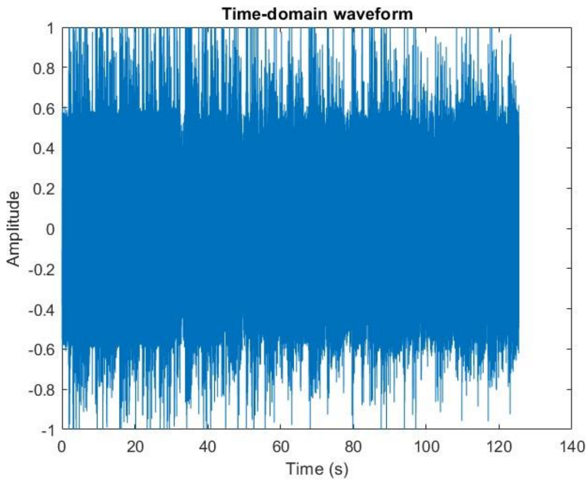

Fig 38: Noisy Speech in time domain

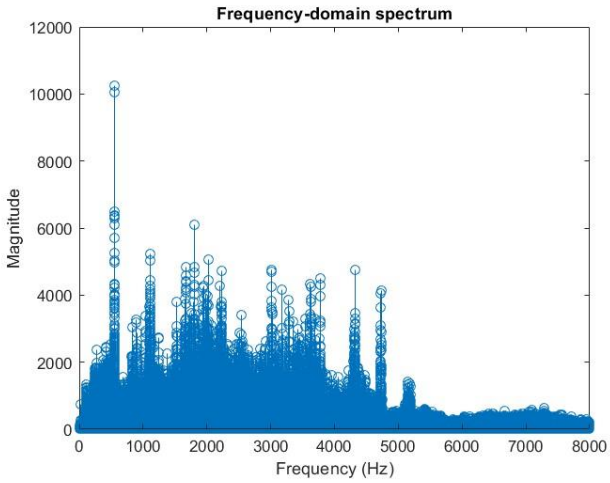

Fig 39: Noisy Speech in frequency domain

9. Loud Music Low:

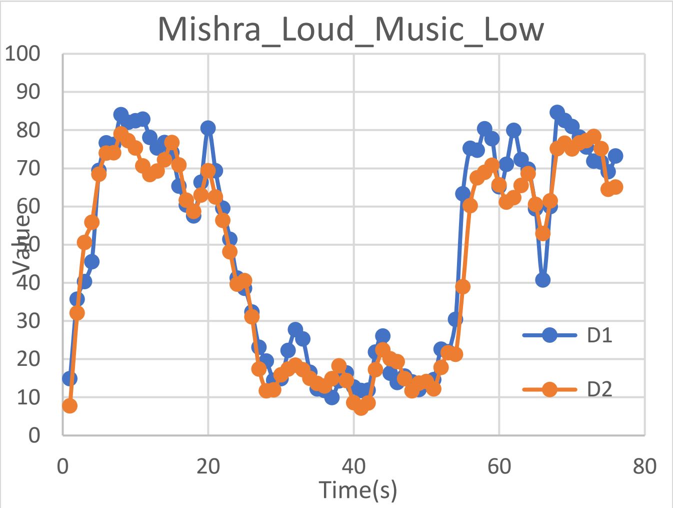

Fig 40: D1 & D2 vs time

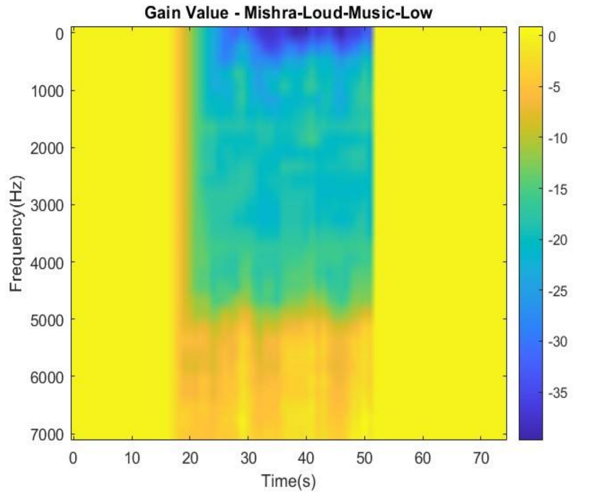

Fig 41: Noisy Speech in time domain

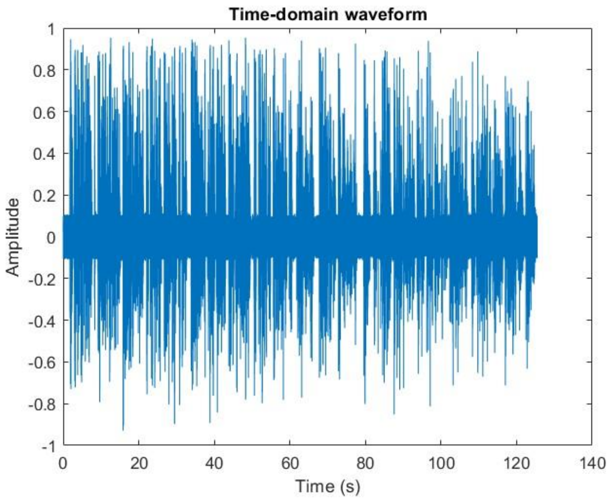

Fig 42: Noisy Speech in time domain

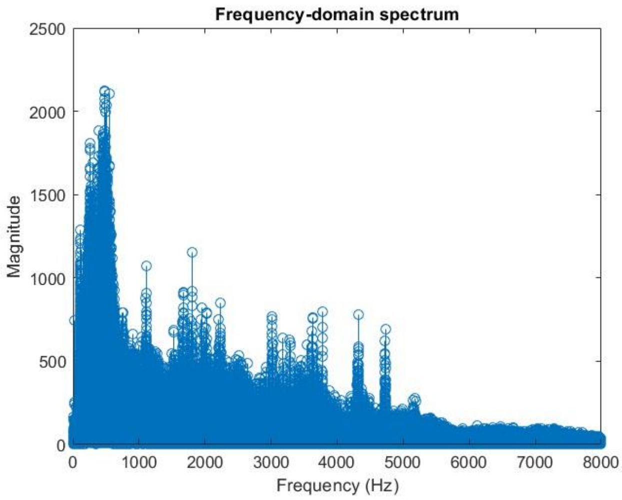

Fig 43: Noisy Speech in frequency domain

10. Soft Babble High:

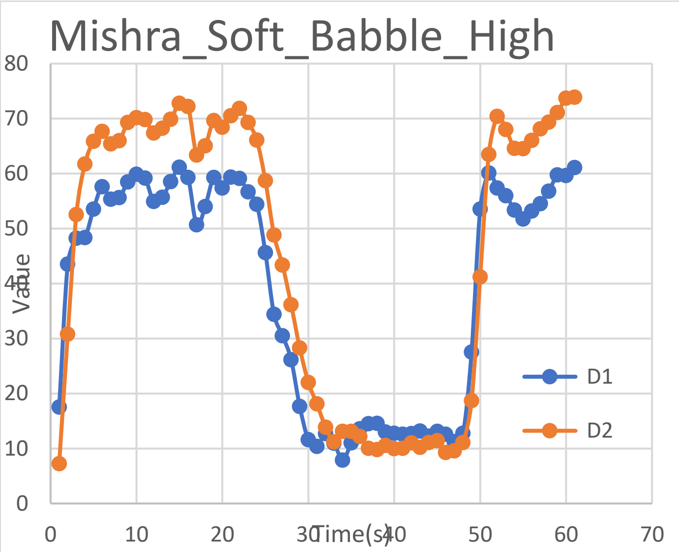

Fig 44: D1 & D2 vs time

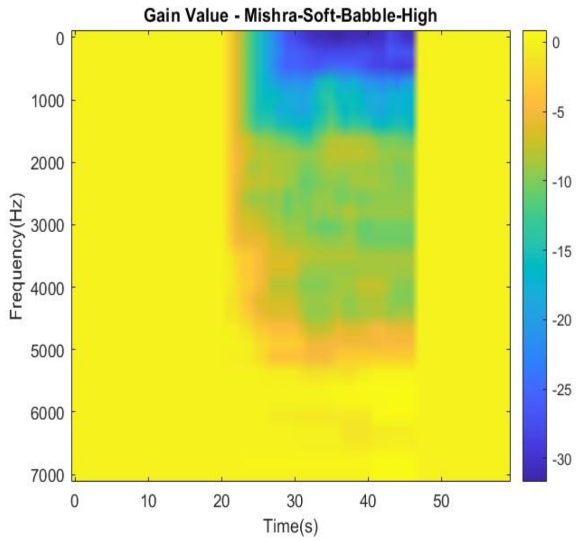

Fig 45: Noisy Speech in time domain

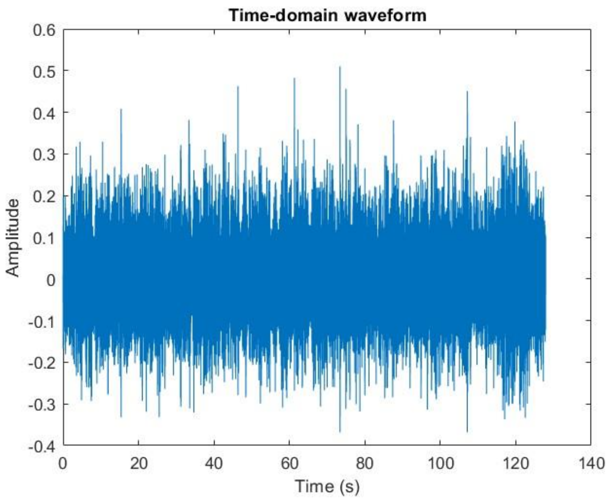

Fig 46: Noisy Speech in time domain

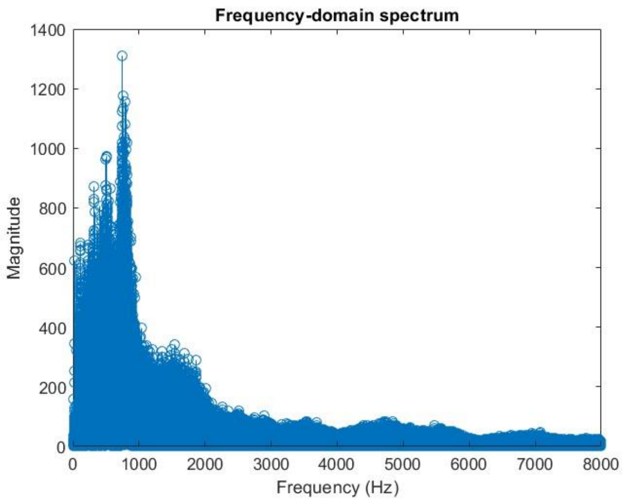

Fig 47: Noisy Speech in frequency domain

11. Soft Babble Low:

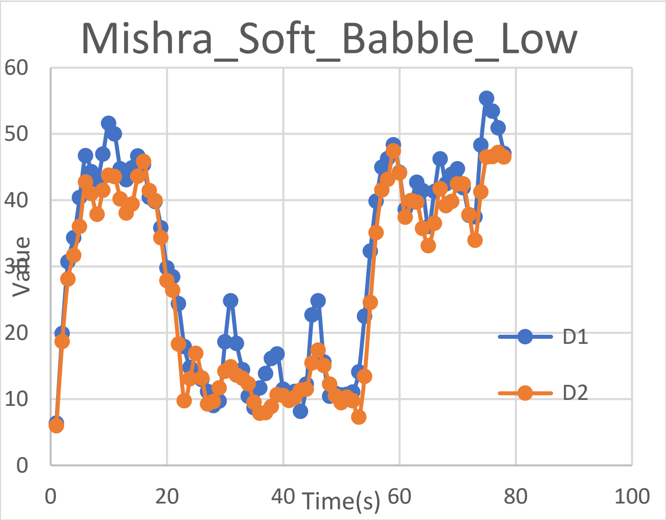

Fig 48: D1 & D2 vs time

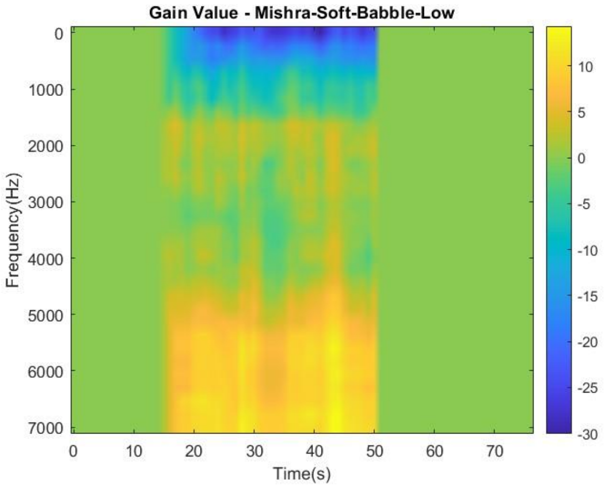

Fig 49: Noisy Speech in time domain

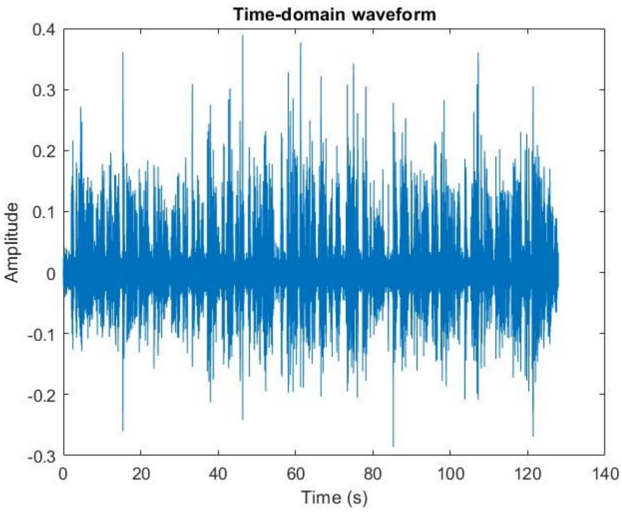

Fig 50: Noisy Speech in time domain

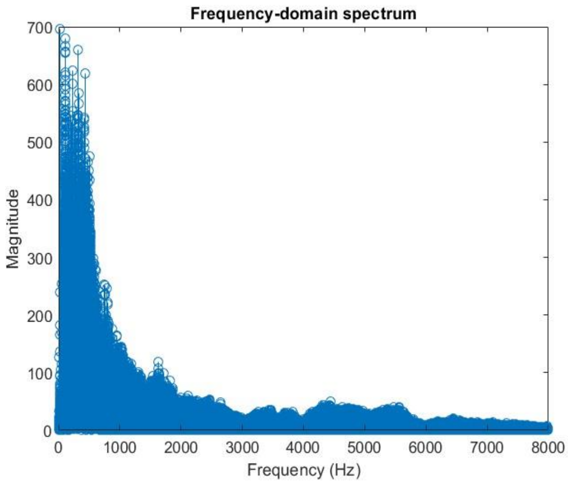

Fig 51: Noisy Speech in frequency domain

12. Soft Machinery High:

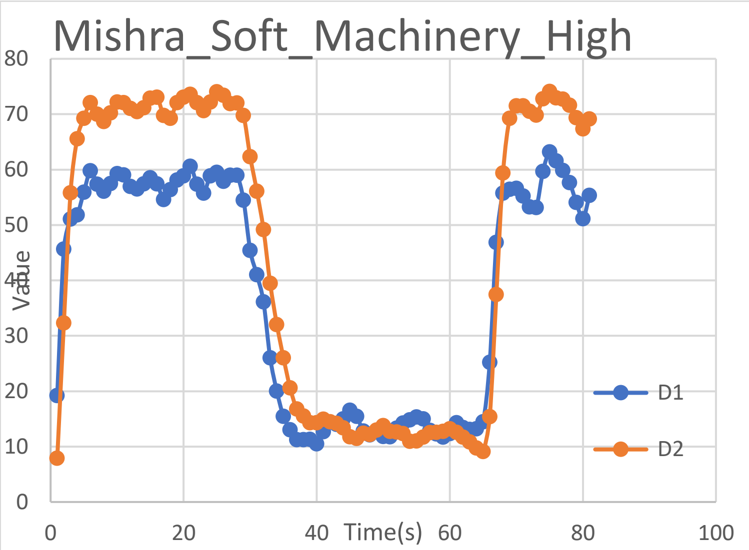

Fig 52: D1 & D2 vs time

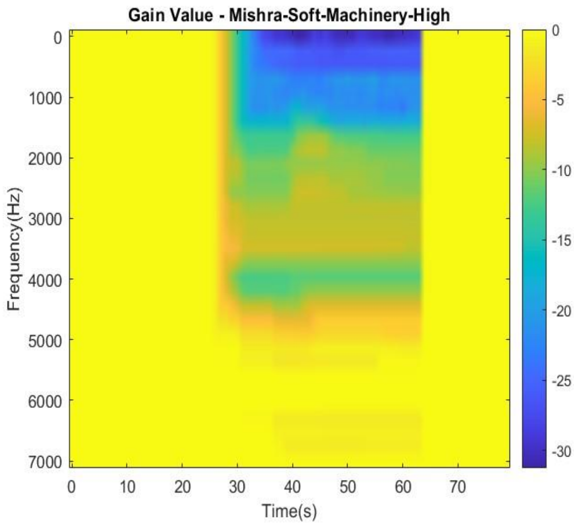

Fig 53: Noisy Speech in time domain

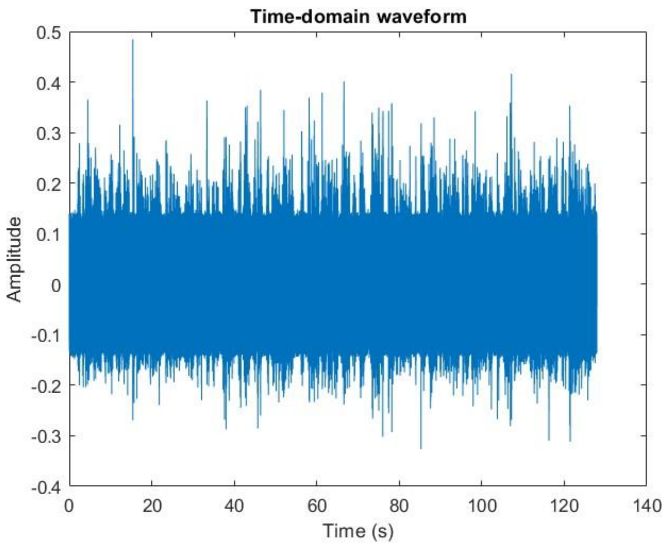

Fig 54: Noisy Speech in time domain

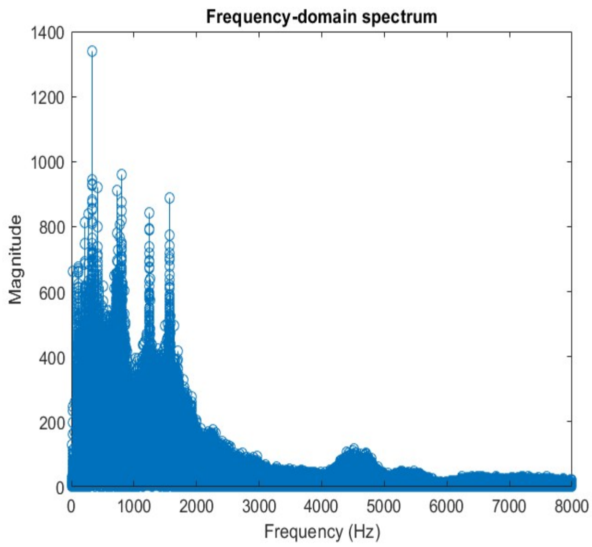

Fig 55: Noisy Speech in frequency domain

13. Soft Machinery Low:

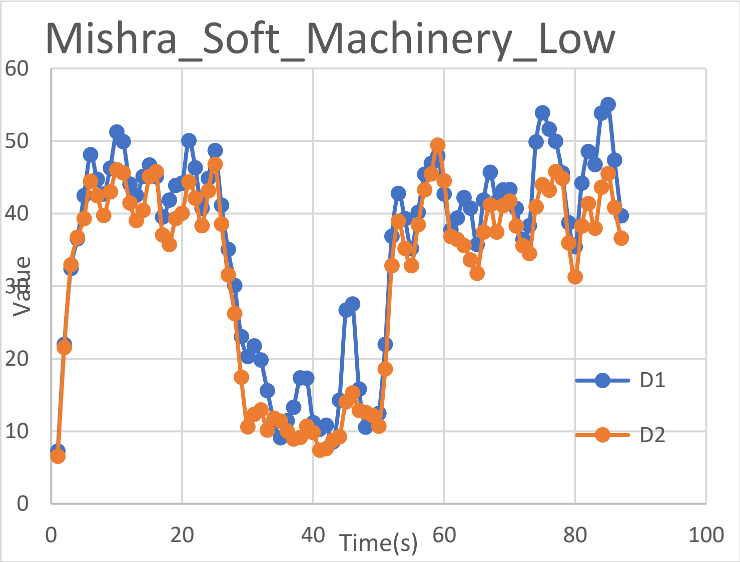

Fig 56: D1 & D2 vs time

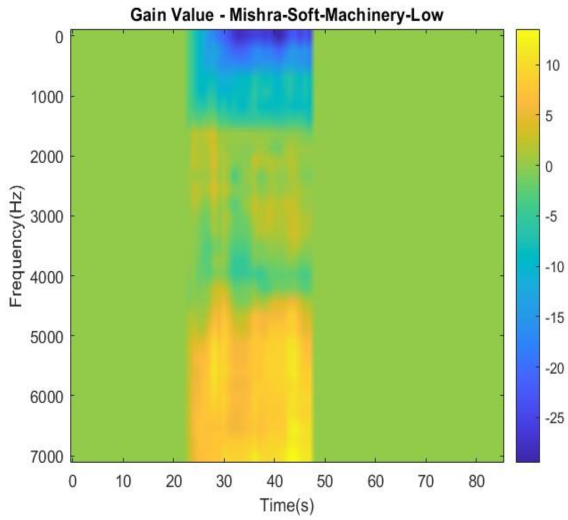

Fig 57: Noisy Speech in time domain

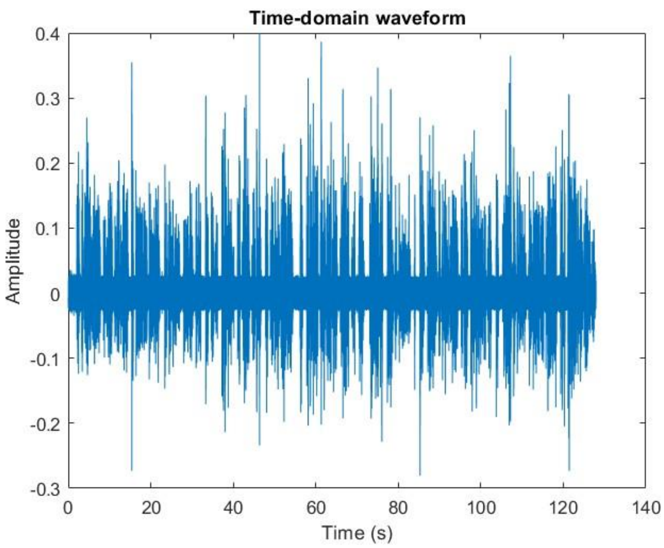

Fig 58: Noisy Speech in time domain

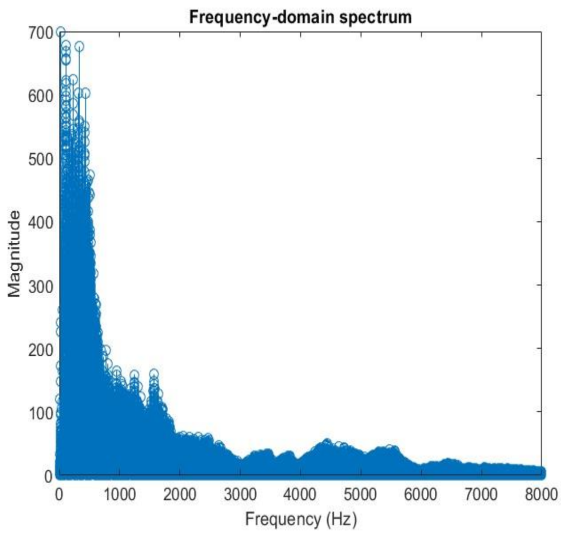

Fig 59: Noisy Speech in frequency domain

14. Soft Music High:

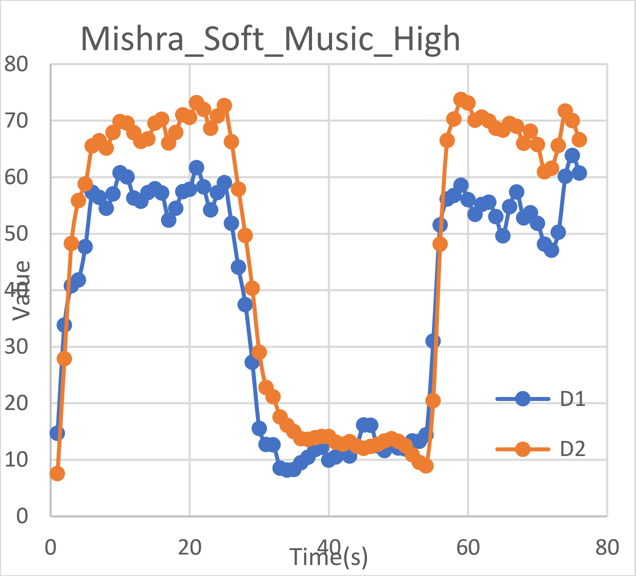

Fig 60: D1 & D2 vs time

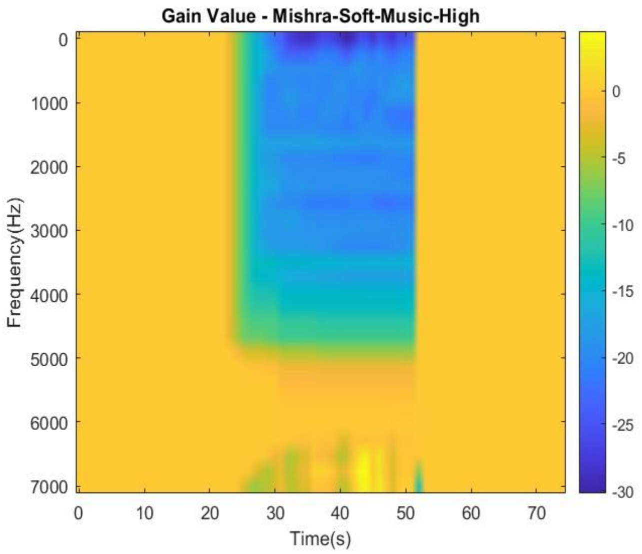

Fig 61: Noisy Speech in time domain

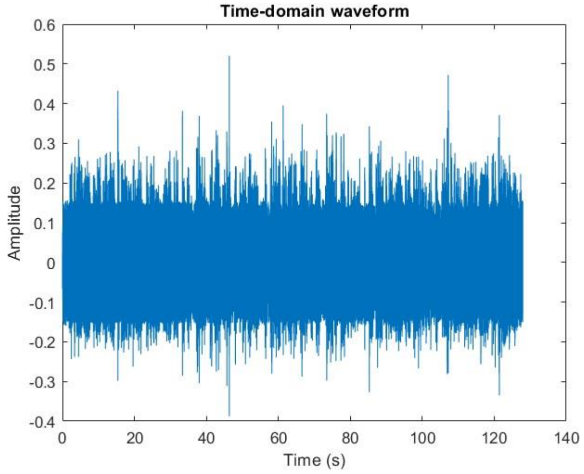

Fig 62: Noisy Speech in time domain

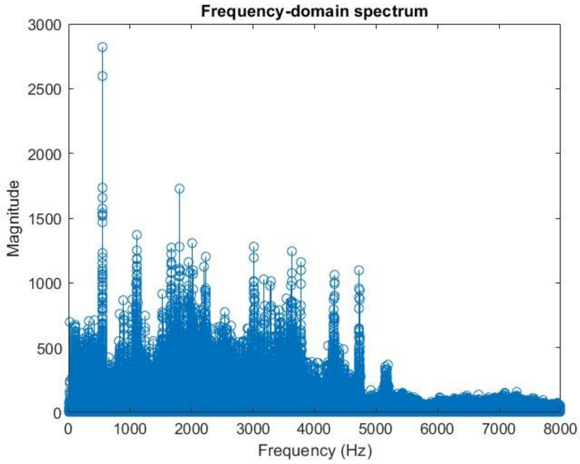

Fig 63: Noisy speech in frequency domain

15. Soft Music Low:

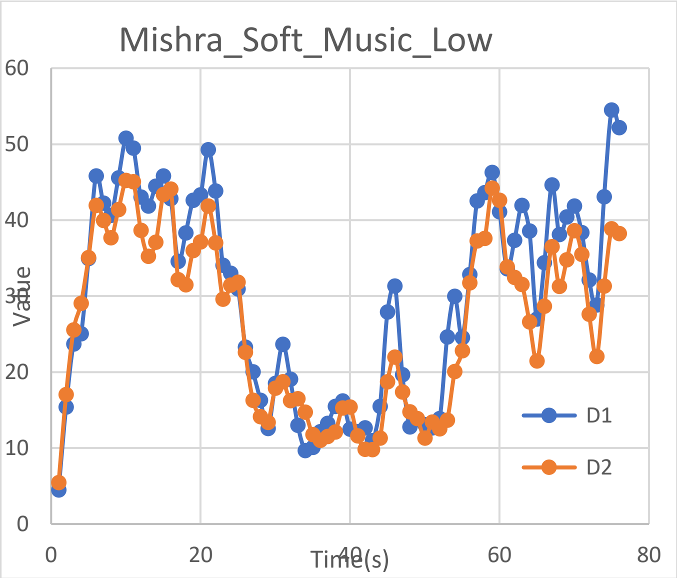

Fig 64: D1 & D2 vs time

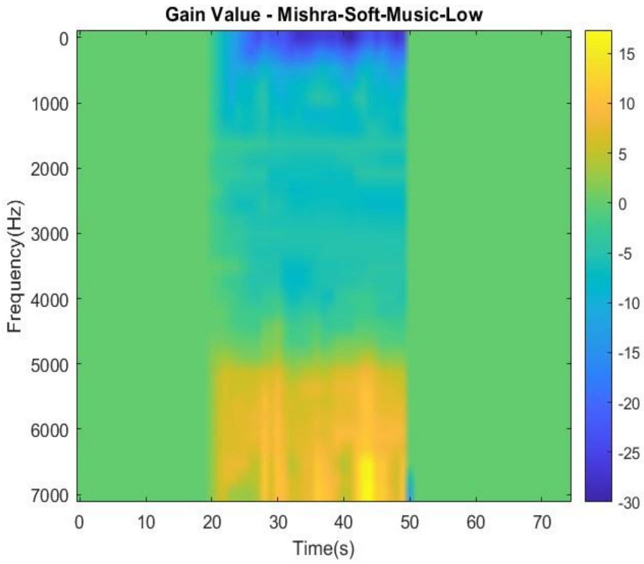

Fig 65: Noisy Speech in time domain

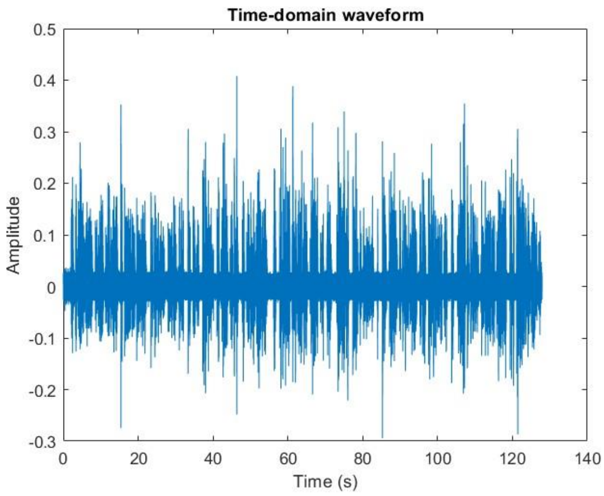

Fig 66: Noisy Speech in time domain

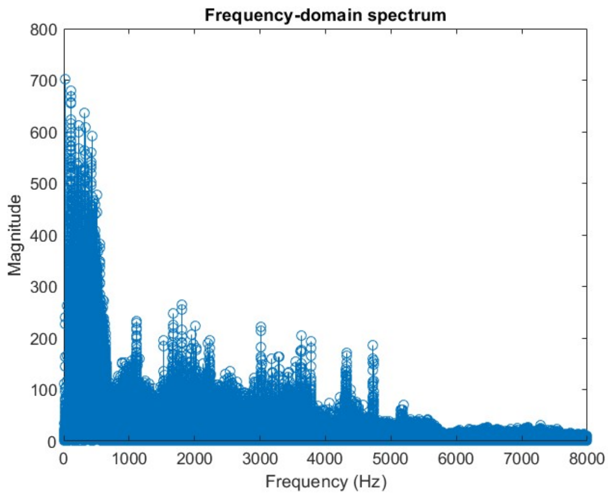

Fig 67: Noisy Speech in frequency domain

Conclusion:

Based on the work that has been completed in this project, it can be concluded that significant progress has been made towards achieving the goals and objectives set forth at the beginning. The development and implementation of the ADRO algorithm, along with the various modules and features that have been added, has resulted in a hearing aid system that is capable of providing an optimal listening experience for users in a variety of different environments.

The testing and validation of the system has also been thorough and effective, with a range of different audio files and scenarios being used to assess the performance and reliability of the Application. The results of these tests have been positive, indicating that the system is robust and effective in a variety of different situations.

Overall, this project has demonstrated the effectiveness and potential of the ADRO algorithm and has contributed to the development of hearing aid technology. The Realtime Android application developed in this project has the potential to significantly improve the quality of life for individuals with hearing impairments, and could be a valuable tool for healthcare professionals and audiologists in the treatment and management of hearing loss.