Contents

1. Introduction:

Reverberation is the persistence of sound in an environment due to the reflection of sound waves off surfaces, creating a prolonged and diminishing series of echoes after the original sound source ceases. It is caused by the reflection of sound waves off surfaces in an enclosed environment. When sound waves encounter surfaces such as walls, ceilings, and floors, they bounce back and forth, creating a complex pattern of reflections.

Reverberation vs Echo

Reverberation and echo are practically the same concepts. Both are time-based audio effects that result from the reflection of sound on hard surfaces. The difference between them lies in time. Echo is a long reflection of sound on a hard and distant surface, while reverberation has a much shorter reflection time, or in this case, reverberation time. It reflects from a surface near another around the listener.

Since, Reverberation is a fundamental aspect of how we perceive sound in different environments. It adds a sense of space and depth to audio, mimicking the reflections and echoes that occur in real-world settings. In this context, MATLAB provides a powerful platform for audio processing, allowing us to simulate reverberation effects and enhance the auditory experience of the audio files.

The process involves creating an impulse response that models the characteristics of reverberation, and then convolving this response with the original audio signal. This MATLAB code involves the steps of setting up parameters, reading an audio file, generating echoes, applying the reverb effect, and finally, playing back the original and reverberated sounds.

.

2. Objectives:

a) To add reverberation to the given audio file.

Adding reverberation to an audio file involves applying a simulated echo effect to create the illusion of the sound reflecting off surfaces in a physical space. This process aims to enhance the spatial characteristics of the audio, making it more immersive and dynamic. By adjusting parameters such as decay time, pre-delay, and diffusion, the desired level and type of reverberation can be tailored to match the aesthetic goals of the audio. This step is commonly employed in audio production to introduce depth, warmth, and a natural ambiance to the original recording, contributing to an overall richer and more engaging auditory experience for the listener.

b) To analyze the phase and magnitude of the audio signal and the reverberated signal.

Analyzing the phase and magnitude of both the original audio signal and the reverberated signal is a crucial step in understanding and manipulating the sonic characteristics of a sound recording. Analyzing the phase allows for precise control over the temporal aspects of the audio, ensuring that the addition of reverberation does not introduce unwanted phase cancellations or distortions. Proper phase alignment is essential for maintaining clarity and fidelity in the audio.

Examining the magnitude spectrum helps in comprehending the frequency distribution and intensity of the original audio and the reverberated signal. This analysis aids in adjusting the reverberation parameters to avoid overemphasis or attenuation of specific frequencies, ensuring a balanced and natural-sounding result.

3. Theoretical Background:

Reverb mimics the way sound waves bounce off surfaces in a room, adding a natural and authentic quality to the audio. It can transform a dry, flat recording into a dynamic and engaging auditory experience. Whether applied subtly for a touch of warmth or more prominently for a dramatic effect, reverberation contributes to the overall texture and ambiance of the audio, making it more captivating for the listener.

Since Reverberation is a fundamental aspect of how we perceive sound in different environments. It adds a sense of space and depth to audio, mimicking the reflections and echoes that occur in real-world settings. In this context, MATLAB provides a powerful platform for audio processing, allowing us to simulate reverberation effects and enhance the auditory experience of the audio files.

The process involves creating an impulse response that models the characteristics of reverberation, and then convolving this response with the original audio signal. This MATLAB code involves the steps of setting up parameters, reading an audio file, generating echoes, applying the reverb effect, and finally, playing back the original and reverberated sounds.

Reverberation in Acoustics:

Reverberation is a phenomenon in acoustics that occurs when sound waves reflect off surfaces in an environment, creating a persistence of sound after the original source ceases. It is a crucial aspect of how we perceive sound in various spaces, such as concert halls, auditoriums, or natural environments.

Impulse Response:

In the context of audio processing, the impulse response represents the acoustic fingerprint of a system. It characterizes how the system responds to an instantaneous input, often resembling the pattern of echoes and reflections in a particular space. In this script, the impulse response (\`h\`) is created by strategically placing delays and amplitudes, simulating the echoes that would occur in a reverberant environment.

Convolution Operation:

The convolution operation, a central element in this script, is a mathematical operation that combines two functions to produce a third. In the context of audio processing, it blends the original audio signal with the impulse response. The resulting signal represents the sound as it would be heard in a space with the defined reverberation characteristics.

Randomness for Naturalness:

The introduction of randomness to the delays and amplitudes is a nod to the variability found in natural reverberant environments. This randomness adds an organic quality to the simulated echoes, making the reverberation effect more realistic and pleasing to the human ear.

Nature of Decay:

The exponential decay is a characteristic of reverberation in acoustic environments. When a sound is produced in a room, it reflects off surfaces, creating a series of echoes. Each echo arrives later than the original sound, and these echoes gradually decrease in amplitude as they bounce around the room and are absorbed by surfaces. The exponential function is a common way to model this decay, as it closely mimics the natural behavior of sound waves in a reverberant space.

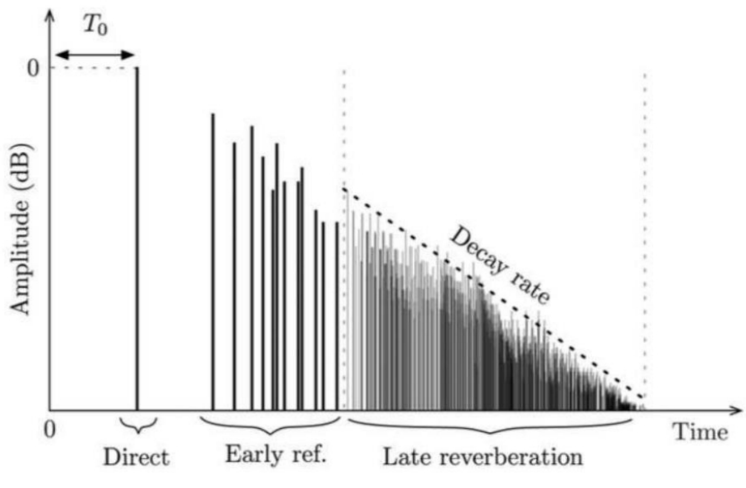


Figure:- Nature of decay rate

6

Normalization:

After applying the convolution, the resulting signal is normalized to ensure that its amplitude doesn't exceed certain limits. This step is essential for preventing distortion and maintaining a balanced and natural-sounding output.

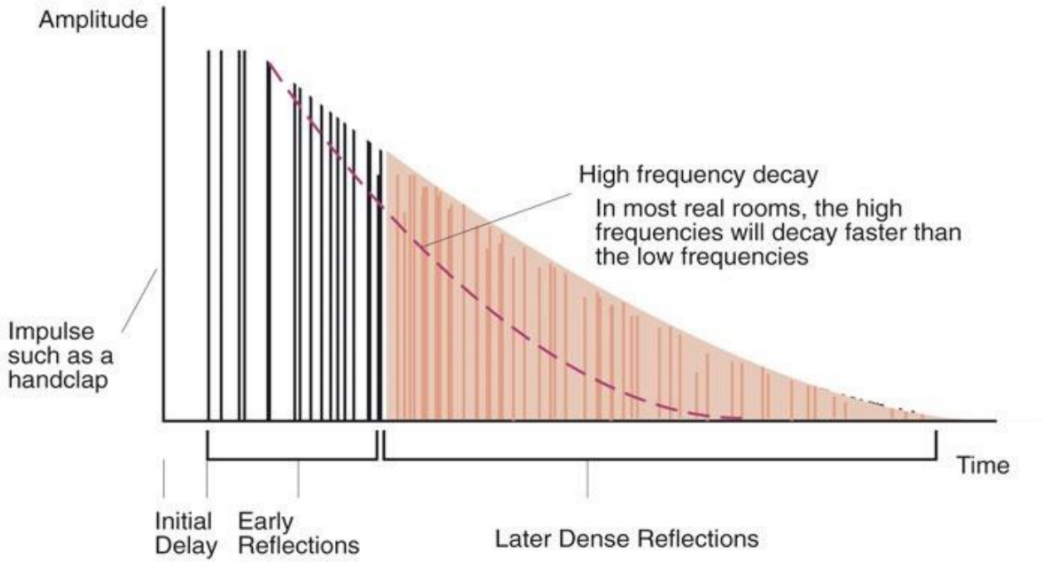


Figure:- Stages of reverberation

In the setup phase for audio processing, the first essential component is the definition of the filename, which serves as the path to the input audio file. This parameter, denoted as 'filename,' necessitates replacement with the actual file path of the audio file intended for processing—typically a file in the .WAV format or any other compatible audio format. Moving forward, the parameter 'delay_spread' plays a crucial role in configuring the amount of reverberation to be applied. This value influences the spread and distribution of delays, directly impacting the perceptual quality of the reverberation effect. Moreover, 'max_delay' is a derived parameter, calculated as a multiple of 'delay_spread.' This calculated value establishes the maximum delay duration, contributing to the overall temporal characteristics of the simulated echoes.

Additionally, the parameter 'num_delays' assumes significance in this setup, determining the number of echoes that will be simulated in the reverberation effect.

By adjusting 'num_delays,' one can control the density and richness of the reverberated sound, offering a means to tailor the auditory experience based on specific preferences or requirements. This comprehensive file and parameters setup is foundational to the subsequent stages of audio processing, providing the necessary inputs for creating a nuanced and controlled application of the reverb effect to the selected audio file.

In this part of the script, the MATLAB function audioread() takes center stage, acting like a specialized tool to open and explore an audio file specified by the variable filename. It's as if this function is the script's way of peeking into the musical treasure chest. Once the file is cracked open, it reveals two key treasures: the actual sound of the music, referred to as the signal (x), and a crucial detail about how fast the script is taking snapshots of this sound, known as the sampling rate (fs). Think of it as capturing the essence of the music and understanding how many moments in time are frozen per second. This dynamic duo, the signal and sampling rate, becomes the foundation for the magic that unfolds in the subsequent parts of the script, where the wizardry of audio manipulation truly begins.

The generation of echoes in the context of creating an impulse response involves a systematic process. Initially, random delays are generated, and these delays are then scaled by the product of 'max_delay' and the sampling rate 'fs.' This scaling ensures that the delays align with a specific temporal range relative to the original sound. Subsequently, amplitudes for each of these delays are determined using an exponential profile. This profile imparts a characteristic decay to the echoes, reflecting a natural acoustic phenomenon where subsequent reflections tend to diminish in intensity. To introduce an element of variability and emulate the organic nature of real-world sound, randomness is incorporated into the amplitudes. This addition of randomness contributes to a more natural and dynamic sonic experience.

Next step involves the creation of the impulse response 'h,' which serves as the model for the reverberation effect. Initially, 'h' is created as an array of zeros, and it is then populated at positions corresponding to the randomly generated delays. The amplitude information, determined by the exponential profile and introduced randomness, is assigned to these positions. This process results in the formulation of the impulse response 'h' that encapsulates the characteristics of the simulated echoes. This impulse response, when convolved with the original audio signal, effectively imparts the desired reverberation effect, enriching the audio with spatial depth and complexity.

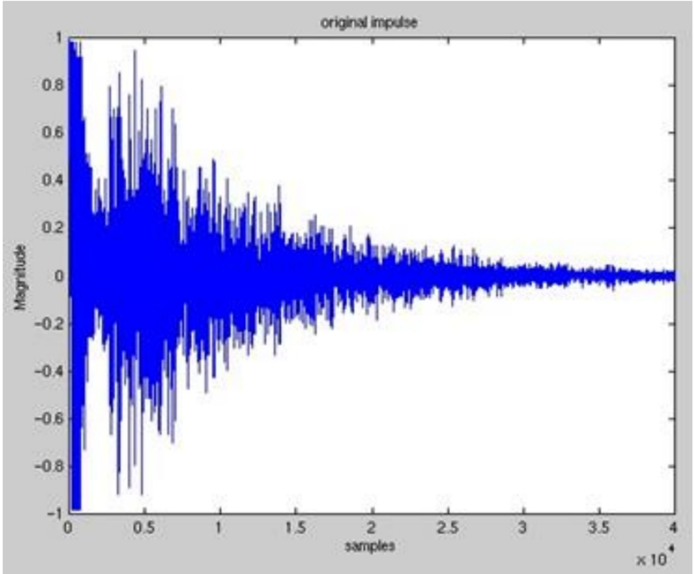
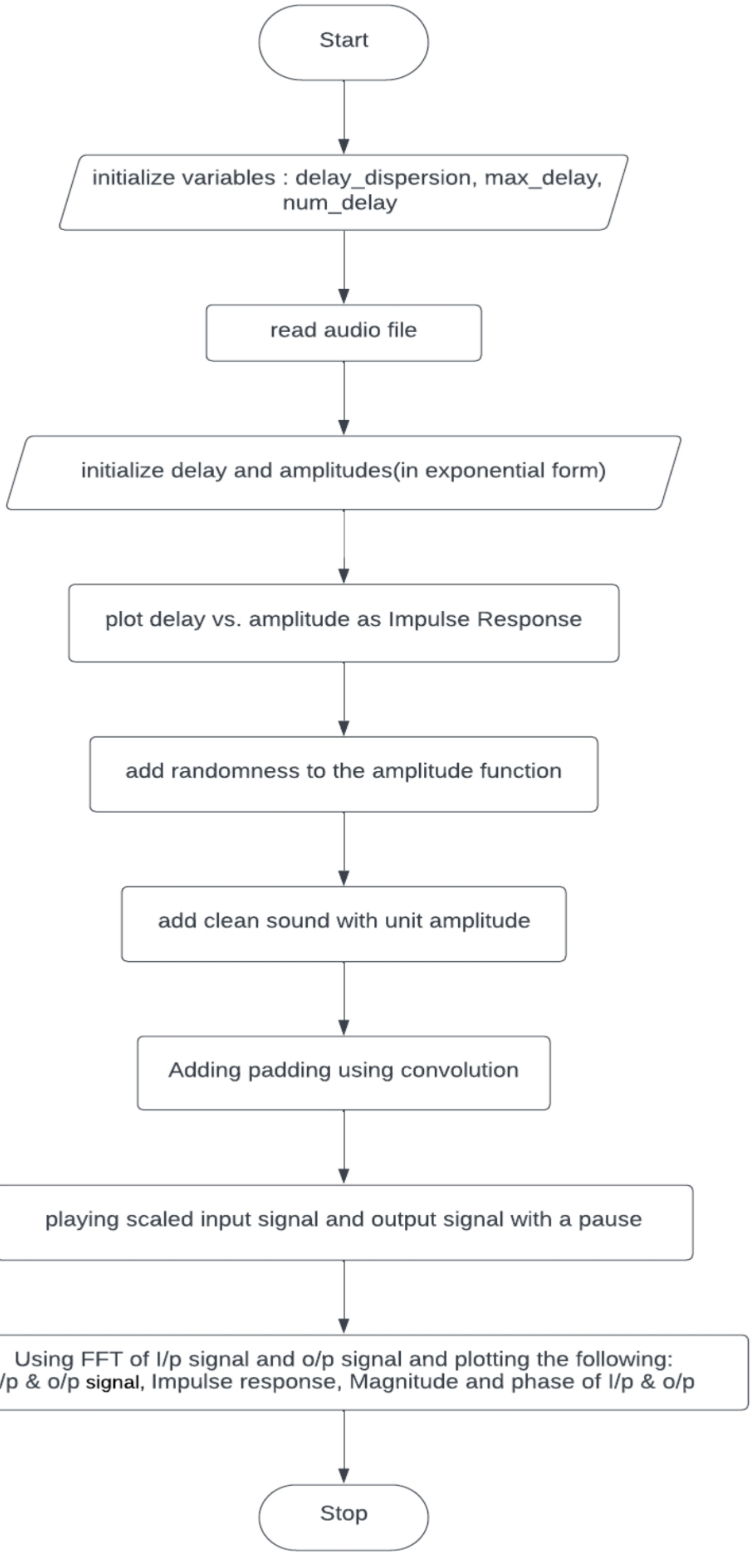


Figure:- Reverberation Impulses realized

In the process of audio playback, the initial step involves playing the unaltered, dry sound denoted as 'x' through the utilization of the sound() function in MATLAB. This represents the original state of the audio before any additional effects or modifications. Subsequently, a deliberate pause is introduced, serving the purpose of allowing sufficient time for the initial, unprocessed sound ('x') to complete its playback. This temporal intermission is crucial to ensure a clear distinction between the original and processed auditory experiences. Following this pause, the second phase commences, wherein the modified version of the audio, referred to as the reverberated sound denoted as 'y', is played. This modified sound incorporates the application of a reverberation effect, introducing echoes and reflections to simulate a more spatial and immersive auditory environment. The combination of these steps provides a sequential and controlled playback experience, allowing for the distinct appreciation of both the unaltered and reverberated sonic characteristics.

4. Flow chart:


Start
initialize variables : delay_dispersion, max_delay, num_delay
read audio file
initialize delay and amplitudes(in exponential form)
plot delay vs. amplitude as Impulse Response
add randomness to the amplitude function
add clean sound with unit amplitude
Adding padding using convolution
playing scaled input signal and output signal with a pause
Using FFT of I/p signal and o/p signal and plotting the following: I/p & o/p signal, Impulse response, Magnitude and phase of I/p & o/p
Stop

5. Matlab Program:

```
clc
close all
clear all
filename = 'Speech_sample.wav';
delay_dispersion = 0.0155;
max_delay = delay_dispersion*4;
num_delays = 1000;

[x, fs] = audioread(filename);
N = length(x);
t = 1:1/fs:11;
delays = round(rand(1,num_delays)*max_delay*fs);
amplitudes = exp(-delays/delay_dispersion/fs);
stem(delays, amplitudes), title('Impulse response'), xlabel('Delay'),ylabel('Amplitude')
amplitudes = 0.7*amplitudes+0.3*rand(size(amplitudes));
delays = [1 delays];
amplitudes = [1 amplitudes];
h = zeros(1,ceil(max_delay*fs));
h(delays) = amplitudes;
y = conv2(x,h(:));
clc
disp('Playing original audio');
soundsc(x,fs);
pause(15);
disp('Playing reverberated audio');
```

```
soundsc(y,fs);


X = fft(x);

Y = fft(y);

k = 1:N;


figure,

subplot(2,1,1),plot(t(1:20000),x(1:20000)),title('Input audio signal'),xlabel('t'),
ylabel('Input audio amplitude')

subplot(2,1,2),plot(t(1:20000),y(1:20000)),title('Reverberated audio signal'),xlabel('t'),
ylabel('Output audio amplitude')


figure,

subplot(2,2,1),plot(k(1:1000),mag2db(abs(X(1:1000)))), title('Magnitude of I/p
audio'), xlabel('k'), ylabel('DFT Magnitude of I/p audio (|X|)')

subplot(2,2,2),plot(k(1:1000),angle(X(1:1000))), title('Phase of I/p audio'), xlabel('k'),
ylabel('Phase of I/p audio (∠X)')

subplot(2,2,3),plot(k(1:1000),mag2db(abs(Y(1:1000)))), title('Magnitude of
reverberated audio'), xlabel('k'), ylabel('DFT Magnitude of reverberated audio (|Y|)')

subplot(2,2,4),plot(k(1:1000),angle(Y(1:1000))), title('Phase of reverberated audio'),
xlabel('k'), ylabel('Phase of reverberated audio (∠Y)')


figure,

subplot(2,1,1),plot(mag2db(abs(X(1:20000)))), title('Magnitude of Input audio'),
xlabel('k'), ylabel('DFT Magnitude of I/p audio (|X|)')

subplot(2,1,2),plot(mag2db(abs(Y(1:20000)))), title('Magnitude of reverberated
audio'), xlabel('k'), ylabel('DFT Magnitude of reverberated audio (|Y|)')
```

6. Results:

Input signal: The input signal is a clean Human speech recorded audio. The specifications of the input audio are:

i. Sampling Frequency = 16000 Hz

ii. Duration = 11 seconds

Reverberated signal: The output signal has a degree of reverberation clearly noticeable by Human ears. The specifications of the output audio are:

i. Delay spread = 0.0155

ii. Maximum Delay = Delay spread * 4 = 0.062

iii. Number of Delays = 1000

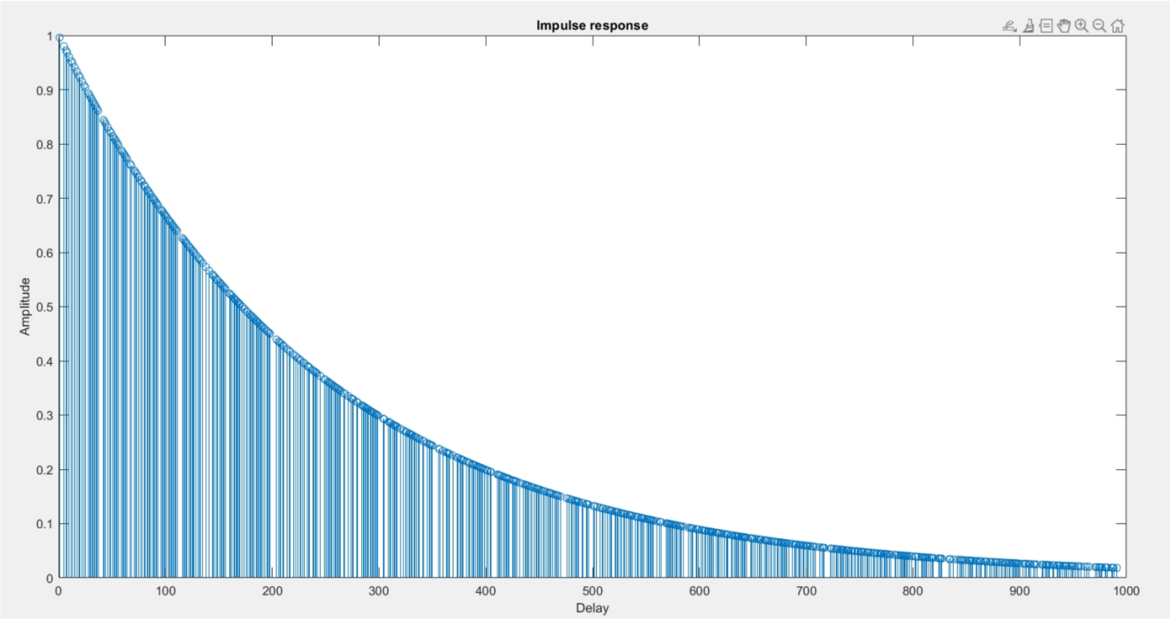


Figure:- Exponential decay in Impulse response

13

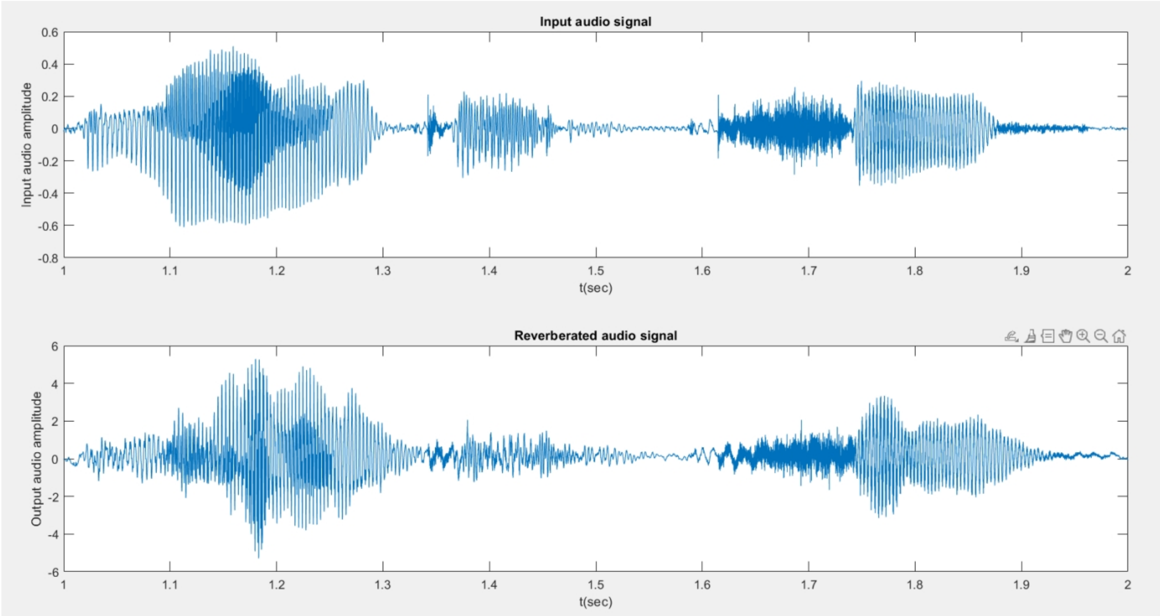


Figure:- Input and Output signal representation in time domain

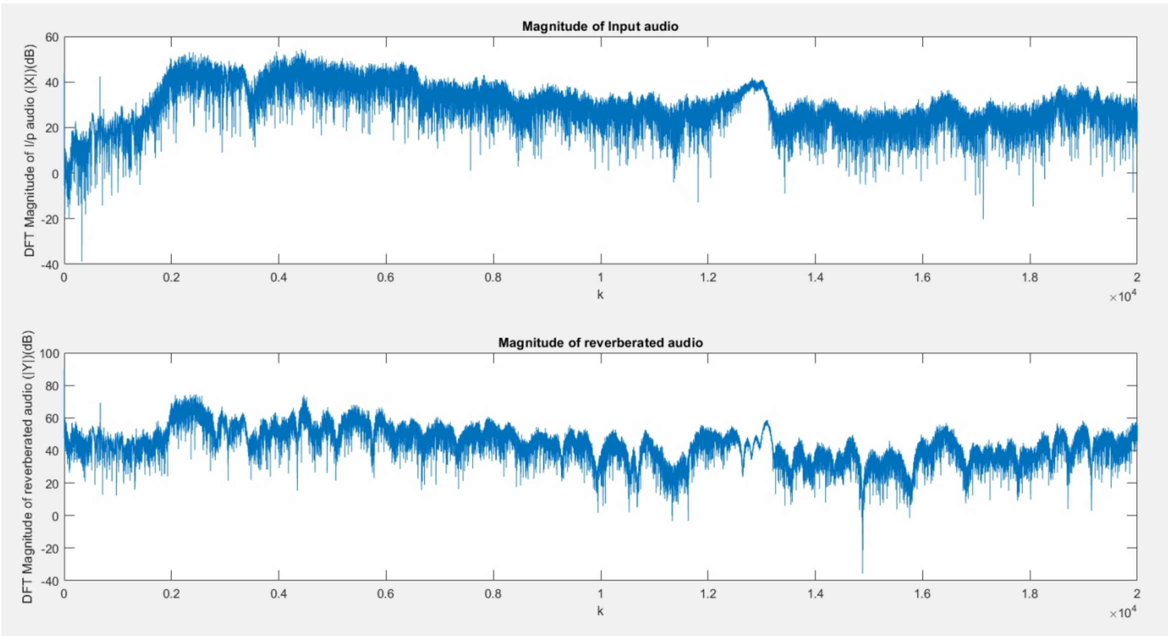


Figure:- Input and Output signal representation in frequency domain

14

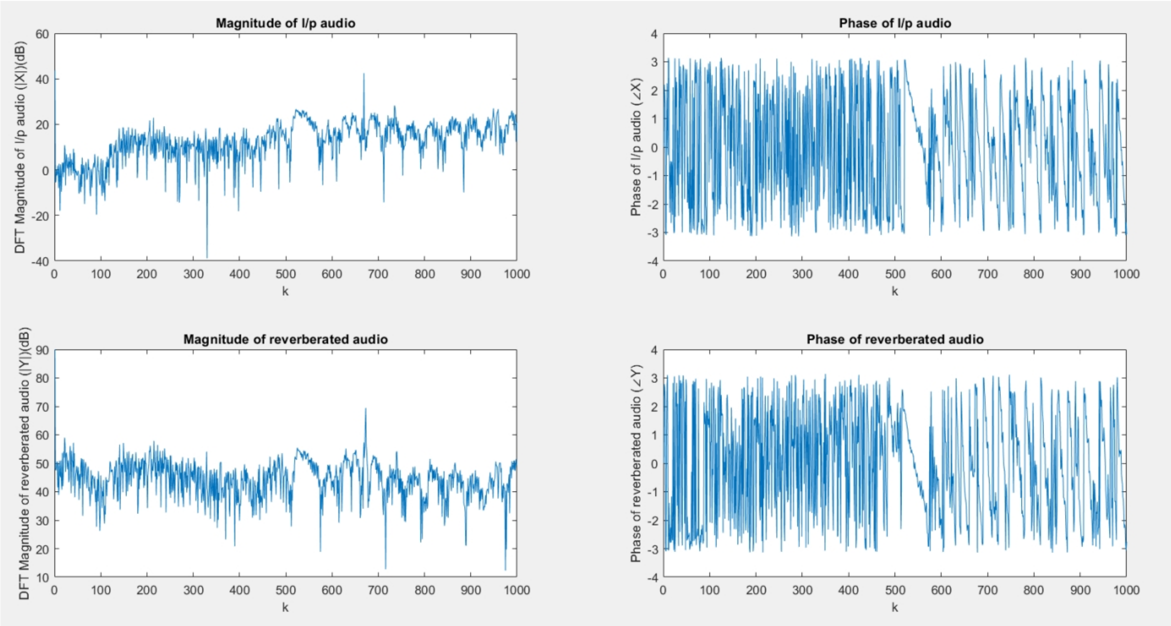


Figure:- Comparison Magnitude & Phase of Input audio signal & Reverberated signal

1. The signal is convolved with the impulse response, simulating the reflection and diffusion of sound in a reverberant environment.
2. The amplitude increases due to constructive interference between the original signal and the echoes, creating a more pronounced and immersive reverberated sound.
3. A slight phase shift occurs in the reverberated output as a result of the added delays, contributing to the spatial and temporal characteristics of the reverb effect.
4. Particularly in the case of instrumental audio input signals, the reverberated output retains significantly more information, enhancing the richness and complexity of the sound, unlike human speech signals which may exhibit different characteristics in the reverb process.
5. Maximum gain in input signal = 54.5450 dB
6. Maximum gain in reverberated signal = 89.8161 dB

7. Applications:

Applying a reverberation effect to an audio file has various applications across different fields. Here are some common scenarios where the use of reverberation is prevalent:

Extrapolation of dimensions of a room:

Echo Profiling: By analyzing the characteristics of added reverberation, it may be possible to estimate the dimensions and features of a room. The decay rate and pattern of echoes can provide information about the size, shape, and materials in the space.

Music Production:

Enhancing Spatial Characteristics: In music production, reverberation is often used to simulate different acoustic environments. For example, adding reverb to a vocal track can create the impression of the vocalist performing in a large hall or a small studio. When recordings are made in acoustically dry environments, adding a touch of reverb in post-production can help to mitigate the unnatural dryness and create a more pleasing sound.

Emphasizing Environments:

In film and video production, reverb is applied to dialogue and sound effects to match the acoustic characteristics of the on-screen environment. For instance, a scene in a cathedral may have more pronounced and extended reverb compared to a scene in a small room.

Creating Immersive Environments:

In the gaming industry, reverb is crucial for creating realistic and immersive audio environments. It's used to simulate the acoustics of virtual spaces, enhancing the overall gaming experience by making the environment sound more authentic.

Virtual Reality (VR) and Augmented Reality (AR) (Spatial Audio):

In VR and AR applications, spatial audio is essential for creating a realistic and immersive experience. Applying reverberation helps to accurately place sounds in a virtual space, allowing users to perceive the environment based on the audio cues.

Educational and Training Simulations:

Realistic Simulations: In educational and training simulations, such as flight or medical simulations, adding reverberation contributes to a more realistic experience. It helps users feel as though they are in a specific environment, enhancing the training effectiveness.

8. Limitations and Future scope:

Limitations:

Artifacts and Coloration:

Excessive application of reverberation can introduce artifacts and coloration to the audio, making it sound unnatural or distorted. Careful consideration of parameters such as reverb time and damping is necessary to avoid these issues

Computational Complexity:

Simulating realistic reverberation effects can be computationally intensive, especially in real-time applications. This limitation may impact the performance of systems with limited computational resources.

Future Scope:

Personalized Audio Environments:

Future developments could focus on technologies that allow users to personalize and adapt the reverberation characteristics based on individual preferences or specific applications, providing a more immersive and tailored listening experience.

Real-Time Adaptive Algorithms:

Advancements in real-time adaptive algorithms could lead to systems that dynamically adjust reverberation parameters based on the changing characteristics of the audio content or the acoustic environment.

Energy-Efficient Algorithms:

Developing energy-efficient algorithms for simulating reverberation effects could be crucial for applications in resource-constrained devices, such as mobile phones and IoT devices, without compromising audio quality.

Interactive Audio Production Tools:

Future tools for audio production could incorporate more intuitive and interactive interfaces for adjusting reverberation parameters, making it easier for musicians and sound engineers to achieve desired effects.

9. References:

[1]https://en.m.wikipedia.org/wiki/Delay_spread

[2] https://in.mathworks.com/help/stateflow/ug/convert-MATLAB-to-flow-chart.html

[3]https://www.nearity.co/apac-en/blog/Reverberation-and-Dereverberation-Effects-and-Solutions