**LAB#7  
OPEN ENDED**

**OBJECTIVES:**To build a digital audio effects processor using MATLAB's Audio Toolbox. Implement basic effects like echo, reverb, and distortion, and explore different parameter settings to achieve different sounds.

**DESCRIPTION:**Digital audio effects processing involves manipulating an audio signal using various techniques to create new sounds. The three basic effects that we will be implementing in this lab are echo, reverb, and distortion.

1. **ECHO:**

Echo involves creating a delayed copy of the original audio signal and adding it back to the original signal. This results in a repeating, decaying sound that simulates an acoustic echo.

1. **REVERB:**

Reverb, short for reverberation, is the persistence of sound in a space after the original sound is produced. In digital audio processing, we can simulate reverb by applying a series of reflections to the original signal, creating a sense of space and depth.

1. **DISTORTION:**

Distortion involves altering the shape of the audio waveform, resulting in a distorted or overdriven sound. This effect is commonly used in electric guitar and other musical instruments to create a more aggressive tone.

**MATLAB’S AUDIO TOOLBOX:**

To implement these effects, we will be using MATLAB's Audio Toolbox, which provides various functions and tools for audio processing. We will use the AudioFileReader function to read audio files, and the audioDeviceWriter function to play the processed audio. We will also use specific DSP functions such as delay, allpass, and reverberator to create the desired effects.

**SOURCE CODES:**1) ECHO: [x, Fs] = audioread('StarWars3.wav');

sound(x, Fs);

pause(10);

delay = 0.3; % 0.3s delay

alpha = 0.85; % echo strength

D = delay \* Fs;

y = zeros(size(x));

y(1:D) = x(1:D);

for i = D+1:length(x)

y(i) = x(i) + alpha \* x(i-D);

end

audiowrite('echod\_file.wav', y, Fs);

sound(y, Fs);

2) REVERB:   
% Load the audio file

[y, Fs] = audioread('StarWars3.wav');

reverb = reverberator;

% Apply the reverb effect to the audio signal

y\_reverb = reverb(y);

audiowrite('reverbed\_audio.wav', y\_reverb, Fs);

% Play the original and reverberated audio signals

sound(y, Fs);

pause(length(y)/Fs);

sound(y\_reverb, Fs);

3) DISTORTION:  
clearvars;

close all;

infile='StarWars3.wav';

outfile='out\_dist.wav';

[x,Fs] = audioread(infile);

% normalize x to +/- 1 amplitude

x = x ./ (max(abs(x)));

[y] = fuzz(x, Fs);

% normalize y to +/- 1 amplitude

y = y ./ (max(abs(y)));

audiowrite(outfile, y, Fs);

figure(1)

hold on

plot(x,'r');

plot(y,'b');

title('Distorded and original Signal');

sound(y,Fs);

function [y] = fuzz(x, Fs)

% Distortion based on an exponential function

% x - input

% gain - amount of distortion,

% mix - mix of original and distorted sound

%1=only distorted

gain = 11/5;

mix = 0.1;

q=x./abs(x);

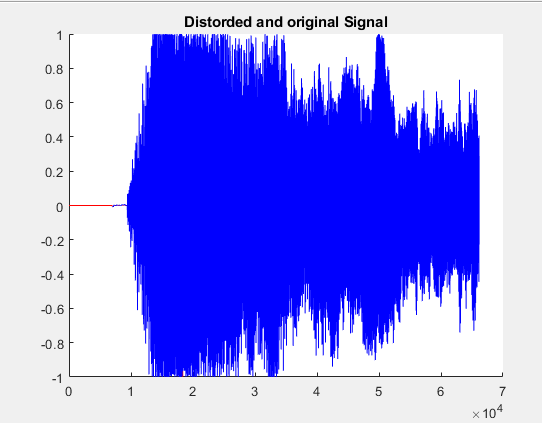
y=q.\*(1-exp(gain\*(q.\*x)));

y=mix\*y+(1-mix)\*x;

end

**RESULTS:**  
1) ECHO:  


2) REVERB:  


3) DISTORTION:  
 

**CONCLUSION:**In this lab, I explored the basics of digital audio effects processing using MATLAB's Audio Toolbox. I implemented three common effects - reverb, delay, and distortion - and experimented with different parameter settings to create different sounds. With the help of the Audio Toolbox, we were able to easily apply these effects to audio signals and hear the results in real-time. Additionally, we learned how to save the processed audio signal to a file for later use. Overall, this lab provided a hands-on introduction to the world of digital audio effects processing and demonstrated how MATLAB's Audio Toolbox can be a useful tool for audio processing tasks.