**Final Project**

**Team Members**

|  |  |  |  |
| --- | --- | --- | --- |
| **Full Name** | **Sec.** | **BN** | **Role in the project** |
| **Mohammed Ibrahim GabAllah** | **2** | **14** | **Task1(a, b, c) & Task3(a) & coordination between members & drawing the figures of task 1 and decorating them & playing the audio files & writing task 1 documentation.** |
| **Mahmoud Amr Nabil** | **2** | **26** | **Task1(d,e) & Task3(e,f) & output audio files and images for user in task 1** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of contents:

Project Tasks:

Task One: Echo generation and removal 3

Task Two: Audio steganography 3

Task Three: Image compression 3

Appendices 6

Appendix A: Codes for Task One: 6

Appendix B: Codes for Task Two: 5

Appendix C: Codes for Task Three: 5

References: 6

**List of Figures**

Figure 1: The audio with and before adding echo………………………………………………………………………………………………….3

Figure 2: Impulse response of echo the system……………………………………………………………………………………………………. 3

Figure 3: Manipulated Audio files signals. you will find them in the folder of task 1……………………………………………. 3

# Project Tasks:

## Task One: Echo generation and removal

* See appendix A for info about how to run the code
* GitHub link to the code and output files(“<https://github.com/marait123/Signals-Final/tree/master/1st-task>”)
* You will find in the previous link the sound files after echo and after echo removal
* The python script of task 1 draws the audio signal before and after echo in 1 figure that is displayed
* Figure 1 shows the audio x(t) before adding the echo and y(t) after adding the echo
* Figure 2 shows the impulse response of the system
* The python script also save the audio files before echo adding and after echo adding and after echo removal

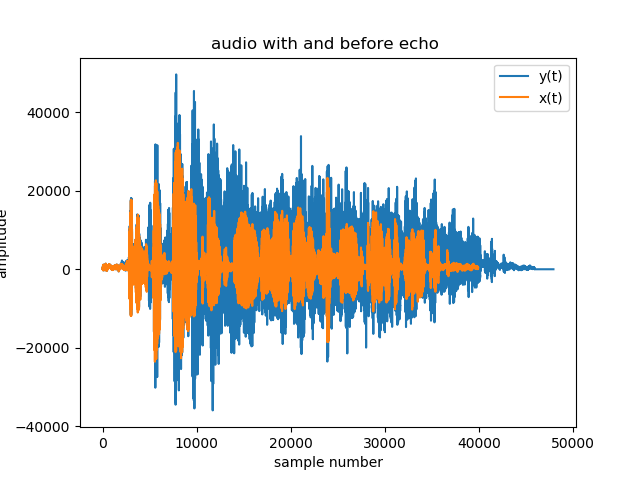
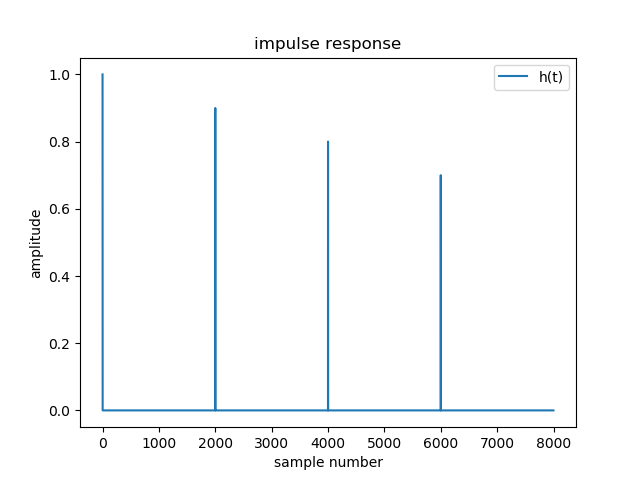


Figure 1

The audio with and before adding echo

Figure 2

Impulse response of echo the system

Figure 3

Manipulated audio files(signals). you will find them in the folder of task 1 in the accompanying link .







Figure 1: Example

## Task Two: Audio steganography

All the required results and answers to questions.

All the required figures.

Label your figures properly

## Task Three: Image compression

All the required results and answers to questions.

All the required figures.

Label your figures properly

# Appendices

## Appendix A: Codes for Task One: (“<https://github.com/marait123/Signals-Final/tree/master/1st-task>”)

from scipy.io import wavfile

import numpy as np

import sounddevice as sd

import time

import matplotlib.pyplot as plt

from scipy import fft,ifft

duration\_s = 4

# a. read the the file audio1.wav

fs, audio = wavfile.read('./audio1.wav')

# let's add echo to the sound

# impulse response has 2001

# b. find y(t) = x(t)+.9 \* x(t-.25) + .8 \* x(t-.5) + .7 \* x(t-.75)

audioLength = len(audio)

secondSize = fs

x25 = np.zeros(audioLength + fs)

x25[int(.25 \* fs):int(audioLength + .25 \* fs)] = .9 \* audio

x50 = np.zeros(audioLength + fs)

x50[int(.5 \* fs):int(audioLength + .5 \* fs)] = .8 \* audio

x75 = np.zeros(audioLength + fs)

x75[int(.75 \* fs):int(audioLength + .75 \* fs)] = .7 \* audio

y = (list(audio) + list(np.zeros(fs))) + x25 + x50 + x75

sd.play(y[0:audioLength].astype(np.dtype('i2')), fs)

plt.title('audio with and before echo ')

plt.xlabel('sample number')

plt.ylabel('amplitude')

plt.plot(y, label = 'y(t)')

plt.plot(audio, label = 'x(t)')

plt.legend()

plt.savefig('audio with and before echo.png')

plt.show()

time.sleep(duration\_s)

#output the audio\_with\_echo\_after\_using\_given\_formula.wav

wavfile.write("audio\_with\_echo\_after\_using\_given\_formula.wav",fs,y[0:audioLength].astype(np.dtype('i2')))

# C. find and plot the impulse response of the echo generation system

# h is the impluse response of the system

h = np.zeros(secondSize)

h[int(0 \* secondSize)] = 1

h[int(.25 \* secondSize)] = .9

h[int(.5 \* secondSize)] = .8

h[int(.75 \* secondSize)] = .7

# plot h

plt.title('impulse response')

plt.xlabel('sample number')

plt.ylabel('amplitude')

plt.plot(h, label = 'h(t)')

plt.legend()

#output impulse\_response.png

plt.savefig('impulse\_response.png')

plt.show()

#ly=lx+lh-1

h\_len=len(h)

audio\_len=len(audio)

N=len(audio)+len(h)-1

h=list(h)+list(np.zeros(N-len(h)))

h=np.array(h)

audio=list(audio)+list(np.zeros(N-len(audio)))

audio=np.array(audio)

#convolve the audio with the impulse response to generate echo

echoedAudio = np.convolve(audio, h)

sd.play(echoedAudio[0:audioLength].astype(np.dtype('i2')), fs)

time.sleep(duration\_s)

#output the audio\_with\_echo\_after\_convolution.wav

wavfile.write("audio\_with\_echo\_after\_convolution.wav",fs,echoedAudio[0:audioLength].astype(np.dtype('i2')))

#get x[n] given y[n](echoedAudio) and h[n] (h)

xn\_fft=fft(echoedAudio,N)/fft(h,N)

xn=ifft(xn\_fft)

xn=np.real(xn)

xn=np.around(xn)

xn=np.array(xn)

xn=xn[0:audioLength]

xn=xn.astype(np.dtype('i2')) # xn is the sound after echo removal

sd.play(xn, fs)

time.sleep(duration\_s)

#output the audio\_after\_echo\_removal.wav

wavfile.write("audio\_after\_echo\_removal.wav",fs,xn)

**How to run task1?**

To run the code you need to do the following

1. Install python3.7
2. Install the following packages (matplolib, sound device, scipy, nump), to install run the packages
   1. pip install matplotlib
   2. pip install sounddevice
   3. pip install numpy
   4. pip install scipy
3. After installation
   1. Run python soundEcho.py (make sure current terminal in the folder of task 1 if you are using terminal to run it.)

# References:

1. Oppenheim Signals and Systems