

*Title:* **Digital Communications II – Formal Report**

*Submitted by:* **Colm Doyle**

*Student Number:* **C15387731**

*Date Submitted:* **15 October 2018**

*Submitted for*

*Module:* **COMM3602 Digital Communications II**

*Programme :* **DT021A/4**

*Lecturers:* **Anthony Kelly**

Contents

**No table of contents entries found.**

**No table of figures entries found.**

**No table of figures entries found.**

# Lab 1 – Speech scrambling using Python

Provide a spectral plot of the input and output simulated voice signal from the frequency inversion scrambler. Insure that the frequency axis is scaled correctly in Hz or kHz. Discuss briefly, the validity of the spectral component frequency locations. Insert the segment of code that produces the inversion of each second sample and explain how it works.

# -\*- coding: utf-8 -\*-

"""

Spyder Editor

This is a temporary script file.

"""

import numpy as np

import scipy.fftpack

import matplotlib.pyplot as plt

from scipy.io import wavfile

# ---------------------creating an input signal---------------------------------

plt.close("all")

fs = 8000.0 # Analogue Freq, Sampling Freq, No. of display samples

fs, audio = wavfile.read('c:\Temp\speech\_dft.wav')

NN=len(audio)\*1.0

t=np.arange(0,NN)/fs

plt.figure(1)

plt.plot(t,audio)

plt.title('Audio Signal')

plt.xlabel('Time (secs)')

plt.ylabel('Amplitude (V)')

plt.grid()

plt.show()

#-------------------Generating the FFT of the signal---------------------------

NFFT=1024 # No. of values in FFT

M = 2\*np.abs(scipy.fftpack.fft(audio,NFFT))/NN

M = M[0:int(NFFT/2)] #slicing operation to avoid mirroring

freq = np.arange(0,NFFT/2) #frequency vector

freq = freq\*fs/NFFT

plt.figure(2)

plt.plot(freq,M)

plt.title('Spectrum of Audio Signal')

plt.xlabel('Frequency (Hz)')

plt.ylabel('Amplitude (V)')

plt.grid()

plt.show()

#----------------------Inverting every 2nd sample------------------------------

count=np.arange(1,len(audio)-1,2)

invaudio=audio

invaudio[count]=-audio[count]

plt.figure(3)

plt.plot(t, audio)

plt.title('Scrammbled Audio Signal')

plt.xlabel('Time (secs)')

plt.ylabel('Amplitude (V)')

plt.grid()

plt.show()

#-------------------Generating the FFT of the signal---------------------------

NFFT=1024 # No. of values in FFT

M = 2\*np.abs(scipy.fftpack.fft(audio,NFFT))/NN

M = M[0:int(NFFT/2)] #slicing operation to avoid mirroring

freq = np.arange(0,NFFT/2) #frequency vector

freq = freq\*fs/NFFT

plt.figure(4)

plt.plot(freq,M)

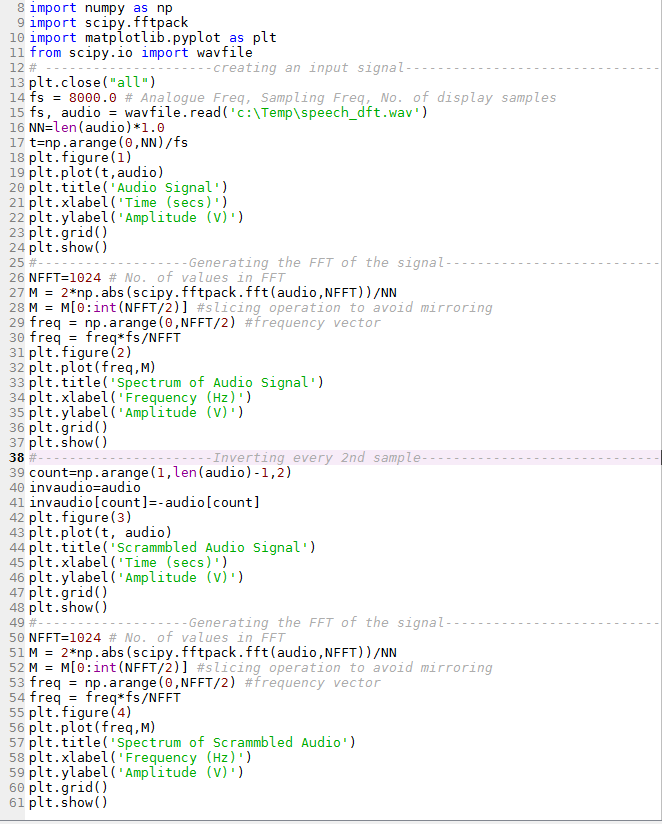
plt.title('Spectrum of Scrammbled Audio')

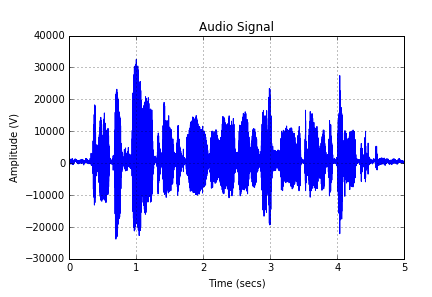
plt.xlabel('Frequency (Hz)')

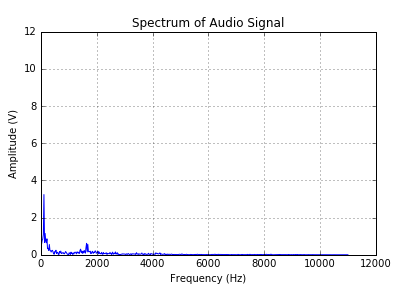
plt.ylabel('Amplitude (V)')

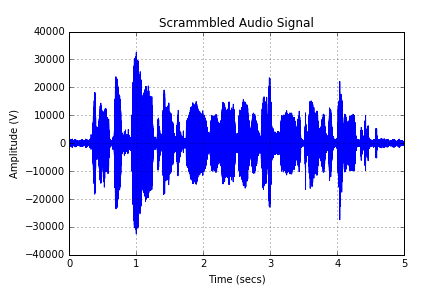
plt.grid()

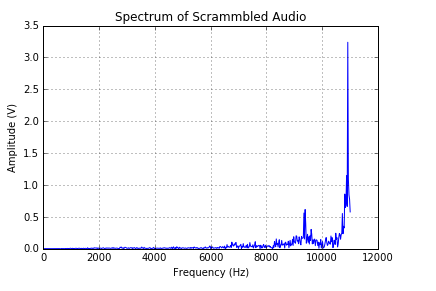
plt.show()













# Lab 2 – M-ary Coding

Provide your complete schematic. Include plots of (both simulated and real):

1. The eye diagram
2. The filtered M-ary signal

Why is M-ary coding used in practice? Discuss briefly the significance of the eye diagram.

LAB2

# Lab 3 – BER Analysis

Provide your complete schematic. Include a plot of the output from the AWGN channel. Show how the BER value obtained in your simulation relates to the calculation of the error probability. What does the BER tell you?

LAB3

# Lab 5 – Entropy Coding

1. Provide the Matlab code used in Lab 5.
2. What is the average level and entropy of the symbols with and without symbol 8? Comments?
3. What is the entropy of a data source with 5 symbols with probabilities of symbols 1 to 4 defined respectively as:

P1=0.12, P2=0.22, P3=0.36, P4=0.05 and P(1/1)= P(4/1)=0.5, P(1/3)=0.6 and P(5/3)=0.4.

LAB5