19CCE303 – DIGITAL COMMUNICATION

ASSIGNMENT I

PROBLEM:

Using an audio source, perform the following in Python:

- Sample at 41.2 kHz;
- 128-Level Quantization; and
- Encode the signal in terms of a binary number.
- Reconstruct the signal from the binary number.

Hint: Take the sample and multiply with the inverse. Compute the weighted sum of shifted versions of the signal.

SOFTWARE REQUIRED:

- Windows 10 Operating System
- Anaconda3 2021.11 (Python 3.9.7 64-bit)
- The Scientific Python Development Environment (Spyder) 5.1.5

PYTHON CODE:

```
import librosa # Python package for music and audio analysis
import matplotlib.pyplot as plt # Collection of command style functions that
make matplotlib work like MATLAB
import numpy as np # Support for large, multi-dimensional arrays and matrices

y, sampling_rate = librosa.load('Impact_Moderato.wav', sr=41200, mono=False) #
Load an audio file as a floating point time series
ts = 1/len(y[1]) # Choose sampling rate
x = np.arange(0,1,ts) # Return evenly spaced values within a given interval
print("\nTo perform quantization:\nMinimmum Value: " + str((min(y[1]))) +
"\nMaximum Value: " + str((max(y[1]))))

# Plot Audio File: -
plt.plot(x,y[1])
plt.title("Sampled Version of Audio File")
plt.xlabel("Time (t)"); plt.ylabel("Amplitude (A)")
plt.grid(True); plt.show()

encoding_delta = []
delta = 0.01 # Initial Delta Value

# Perform Delta Modulation: -
for i in range(len(y[1])):
```

```
    if delta < y[1][i]:
        encoding_delta.append(0)
        delta = delta + 0.01

    else:
        encoding_delta.append(1)
        delta = delta - 0.01
print("\nLength of Encoded Data Array: ", len(encoding_delta))
print("First 50 Encoded Data: ", encoding_delta[0:50])

start = -0.3
quantization_levels = []
quantization_levels.append(start)

# Perform 128-Level Quantization: -
for i in range(128):
    start = start + 0.0047
    quantization_levels.append(start)
print("\nNumber of Quantization Levels: ", len(quantization_levels))
print("First 50 Quantization Levels: ", quantization_levels[0:50])

quantized_values = []
for i in range(len(quantization_levels)-1):
    temp = (quantization_levels[i]+quantization_levels[i+1])/2
    quantized_values.append(temp)
print("\nNumber of Quantized Values: ", len(quantized_values))
print("First 50 Quantized Values: ", quantized_values[0:50])

encoding = encoded_signal = []
for i in range(len(y[1])):
    flag = 0
    for j in range(len(quantized_values)):

        if y[1][i]<quantized_values[j]:
            encoding.append(j+1)
            encoded_signal.append(bin(j+1))
            flag = 1

        if flag == 1:
            break
print("\nLength of Encoded Signal Array: ", len(encoded_signal))
print("First 50 Encoded Signal: ", encoded_signal[0:50])

delta = 0.01 # Initial Delta Value
reconstruction = reconstruction_delta = []

# Perform Reconstruction for the Encoded Binary Numbers:
```

```
for i in range(len(encoding_delta)):
    if encoding_delta[i] == 0:
        delta = delta + 0.001
        reconstruction_delta.append(delta)
    else:
        delta = delta - 0.001
        reconstruction_delta.append(delta)

plt.plot(x, reconstruction_delta)
plt.title("Reconstructed Audio File from the Encoded Binary Numbers")
plt.xlabel("Time (t)"); plt.ylabel("Amplitude (A)")
plt.grid(True); plt.show()
```

RESULTS:

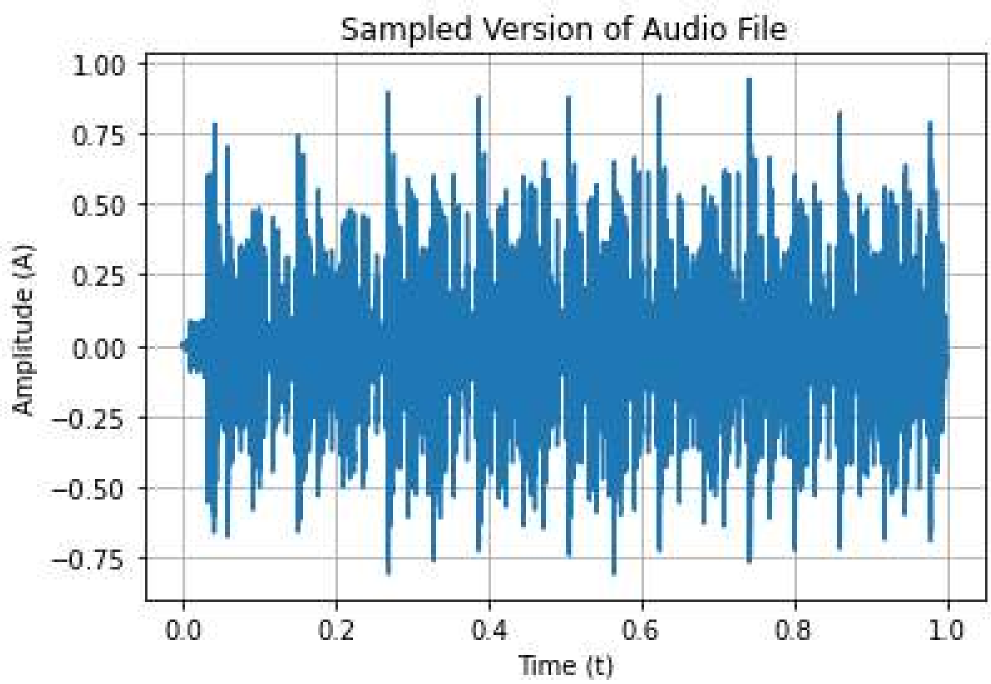

Figure 1 – Sampled Version of Audio File

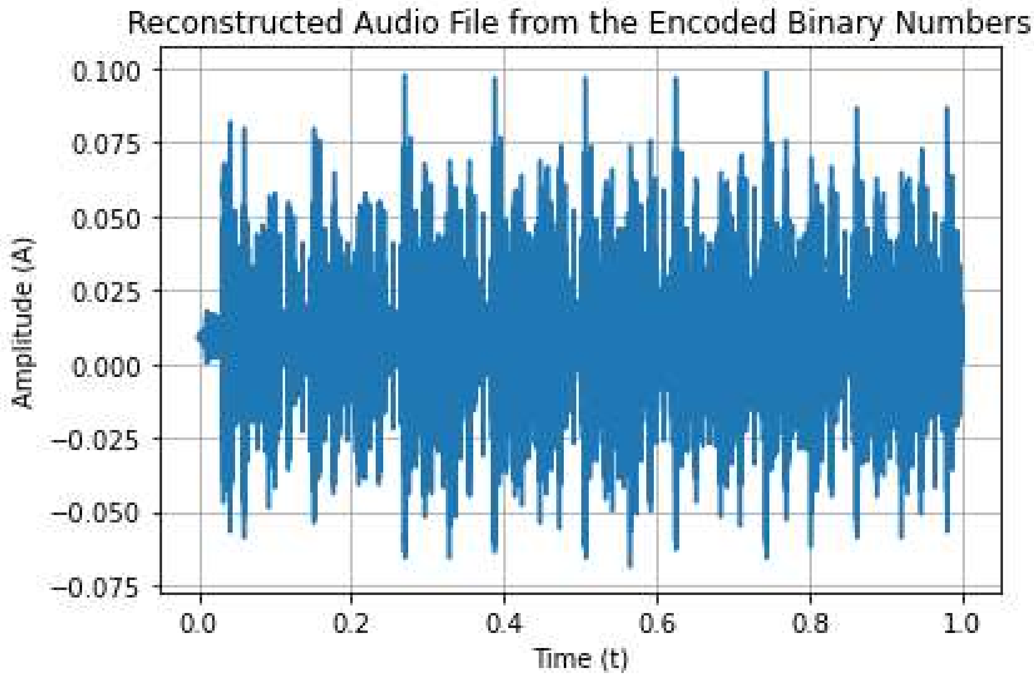

Figure 2 – Reconstructed Audio File from the Encoded Binary Numbers

```
Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.29.0 -- An enhanced Interactive Python.

Restarting kernel...
```

```
In [1]:         'E:/Plan B/Amrita Vishwa Vidyapeetham/Subject Materials/Semester V/
19CCE303 - Digital Communication/Assignments/19CCE303_Assignment_I_Code.py'      ='E:/Plan
B/Amrita Vishwa Vidyapeetham/Subject Materials/Semester V/19CCE303 - Digital
Communication/Assignments'

To perform quantization:
Minimmum Value: -0.8076387
Maximum Value: 0.9423016

Length of Encoded Data Array:  1381421
First 50 Encoded Data:  [1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0,
1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1]

Number of Quantization Levels:  129
First 50 Quantization Levels:  [-0.3, -0.2953, -0.2906, -0.28590000000000004,
-0.28120000000000006, -0.2765000000000001, -0.2718000000000001, -0.2671000000000001,
-0.26240000000000013, -0.25770000000000015, -0.25300000000000017, -0.24830000000000016,
-0.24360000000000015, -0.23890000000000014, -0.23420000000000013, -0.22950000000000012,
-0.2248000000000001, -0.2201000000000001, -0.2154000000000001, -0.21070000000000008,
-0.20600000000000007, -0.20130000000000006, -0.19660000000000005, -0.19190000000000004,
-0.18720000000000003, -0.18250000000000002, -0.1778, -0.1731, -0.1684,
-0.16369999999999998, -0.15899999999999997, -0.15429999999999996, -0.14959999999999996,
-0.14489999999999995, -0.14019999999999994, -0.13549999999999993, -0.13079999999999992,
-0.1260999999999999, -0.12139999999999991, -0.11669999999999991, -0.11199999999999992,
-0.10729999999999992, -0.10259999999999993, -0.09789999999999993, -0.09319999999999994,
-0.08849999999999994, -0.08379999999999994, -0.07909999999999995, -0.07439999999999995,
-0.06969999999999996]

Number of Quantized Values:  128
First 50 Quantized Values:  [-0.29764999999999997, -0.29295000000000004, -0.28825,
-0.2835500000000001, -0.27885000000000004, -0.2741500000000001, -0.2694500000000001,
-0.26475000000000015, -0.2600500000000001, -0.2553500000000002, -0.25065000000000015,
-0.24595000000000017, -0.24125000000000013, -0.23655000000000015, -0.2318500000000001,
-0.22715000000000013, -0.2224500000000001, -0.2177500000000001, -0.21305000000000007,
-0.2083500000000001, -0.20365000000000005, -0.19895000000000007, -0.19425000000000003,
-0.18955000000000005, -0.18485000000000001, -0.18015000000000003, -0.17545, -0.17075,
-0.16604999999999998, -0.16135, -0.15664999999999996, -0.15194999999999997,
-0.14724999999999994, -0.14254999999999995, -0.13784999999999992, -0.13314999999999994,
-0.1284499999999999, -0.12374999999999992, -0.1190499999999999, -0.11434999999999992,
-0.10964999999999991, -0.10494999999999993, -0.10024999999999992, -0.09554999999999994,
-0.09084999999999993, -0.08614999999999995, -0.08144999999999994, -0.07674999999999996,
-0.07204999999999995, -0.06734999999999997]

Length of Encoded Signal Array:  2671652
First 50 Encoded Signal:  [65, '0b1000001', 65, '0b1000001', 65, '0b1000001', 65,
'0b1000001', 65, '0b1000001', 65, '0b1000001', 65, '0b1000001', 65, '0b1000001', 65,
```

```
'0b1000001', 65, '0b1000001', 65, '0b1000001', 65, '0b1000001', 65, '0b1000001', 65,
'0b1000001', 65, '0b1000001', 65, '0b1000001', 65, '0b1000001', 65, '0b1000001', 65,
'0b1000001', 65, '0b1000001', 65, '0b1000001', 65, '0b1000001', 65, '0b1000001', 65,
'0b1000001', 65, '0b1000001']
```

In [2]: