ISTE Audio Processing SMP – Assignment 2

Code used for the creating the plots and spectrograms for Q1 and Q2 respectively:

import numpy

import librosa

sr = 22050

T = 5.0

t = numpy.linspace(0, T, int(T\*sr), endpoint = False)

x1 = 0.5\*numpy.sin(2\*numpy.pi\*256\*t)

x2 = 0.5\*numpy.sin(2\*numpy.pi\*256\*2\*t)

x3 = 0.5\*numpy.sin(2\*numpy.pi\*256\*3\*t)

x4 = 0.5\*numpy.sin(2\*numpy.pi\*256\*4\*t)

x=x1+x2+x3+x4

librosa.output.write\_wav('Resultant.wav', x, sr)

from IPython import get\_ipython

ipy = get\_ipython()

if ipy is not None:

ipy.run\_line\_magic('matplotlib', 'inline')

import matplotlib.pyplot as plt

import librosa.display

plt.figure(figsize=(14,5))

plt.plot(x1[1000:1100])

plt.grid()

plt.figure(figsize=(14,5))

plt.plot(x2[1000:1100])

plt.grid()

plt.figure(figsize=(14,5))

plt.plot(x3[1000:1100])

plt.grid()

plt.figure(figsize=(14,5))

plt.plot(x4[1000:1100])

plt.grid()

plt.figure(figsize=(14,5))

plt.plot(x[1000:1100])

plt.grid()

X1 = librosa.stft(x1)

Xdb1 = librosa.amplitude\_to\_db(abs(X1))

plt.figure(figsize=(14,5))

librosa.display.specshow(Xdb1, sr=sr, x\_axis='time', y\_axis='log')

plt.colorbar()

X = librosa.stft(x)

Xdb = librosa.amplitude\_to\_db(abs(X))

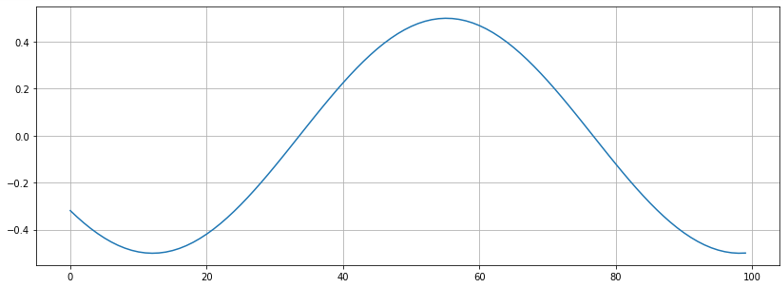
plt.figure(figsize=(14,5))

librosa.display.specshow(Xdb, sr=sr, x\_axis='time', y\_axis='log')

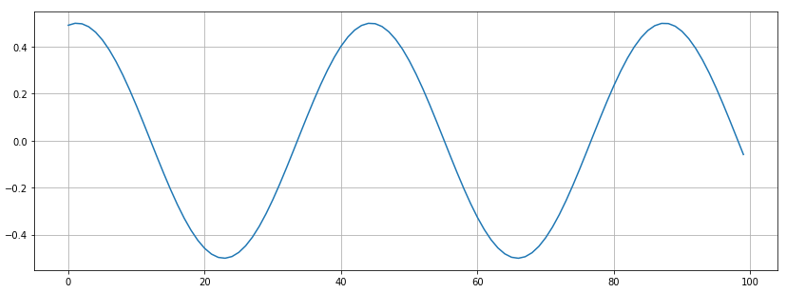
plt.colorbar()

**Q1**. Create an overtone series consisting of four harmonics. Plot each harmonic separately, and then plot the entire resulting audio signal too. You may use any fundamental frequency you like. Attach all the plots (there should be 5) and also the final audio file (which includes all the overtones).

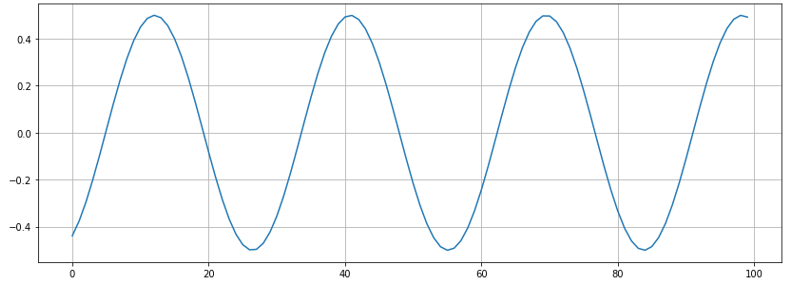
First harmonic (256 Hz):



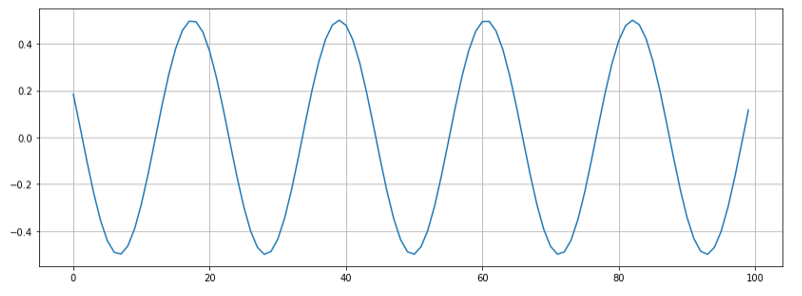
Second harmonic (512 Hz):



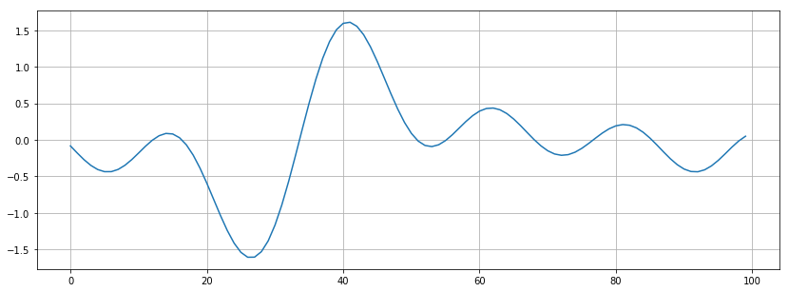
Third harmonic (768 Hz):



Fourth harmonic (1024 Hz):



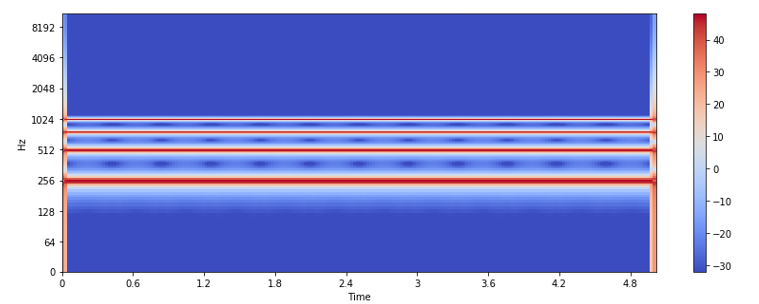
Resulting audio signal:



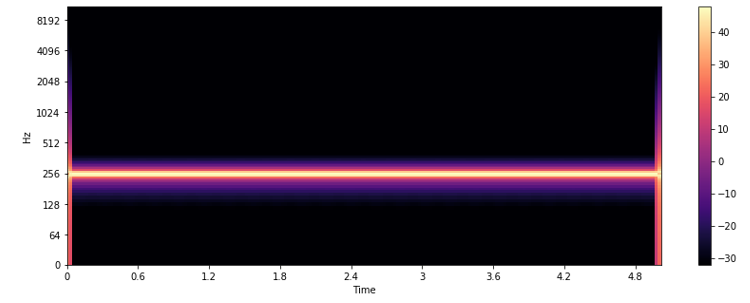
The final audio file is attached with the email as Resultant.wav

**Q2**. Make a spectrogram for the audio you created in the last question. Attach a screenshot of it and compare it to the spectrogram you obtained for the pure sin wave. How is it different?

Spectrogram for the final audio file Resultant.wav:



Spectrogram for pure sine wave of 256 Hz:



The spectrogram for Resultant.wav shows the existence of three overtones along with the fundamental frequency, while that of the 256 Hz pure sine wave shows the existence of only the fundamental frequency.

Due to the existence of more overtones, Resultant.wav has a brighter timber than the pure 256 Hz sine wave.

**Q3**. Record yourself first whistling a small tune and then humming the same tune. Save this as a .wav file and obtain its spectrogram. What differences do you note between the region where you were whistling and the region where you were humming, and what do you infer from it? Attach a screenshot of the spectrogram. (try to record in a quiet place)

Code used for creating the spectrogram:

import librosa

audio\_file = 'Recording.wav'

x, sr = librosa.load(audio\_file)

from IPython import get\_ipython

ipy = get\_ipython()

if ipy is not None:

ipy.run\_line\_magic('matplotlib', 'inline')

import matplotlib.pyplot as plt

import librosa.display

X = librosa.stft(x) # computing the STFT

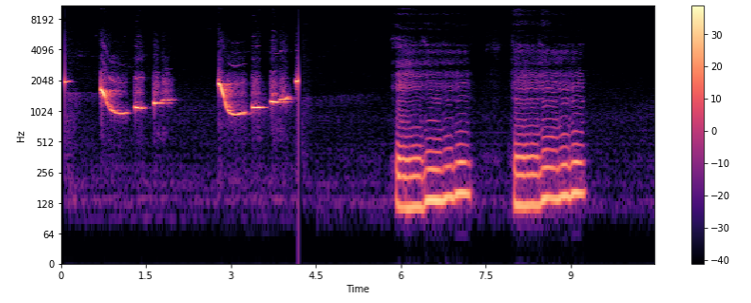
Xdb = librosa.amplitude\_to\_db(abs(X)) # convert amplitide to dB scale

plt.figure(figsize=(14,5))

librosa.display.specshow(Xdb, sr=sr, x\_axis='time', y\_axis='log')

plt.colorbar()

Spectrogram for the recording of whistling and humming:



In the above spectrogram, the first half corresponds to whistling, while the second half corresponds to humming. It seems that the humming part has a much brighter timbre than the whistling part as it has a lot more overtones.

Also, it is evident that the humming part has a lower frequency than the whistling part.