**SIDDAGANGA INSTITUTE OF TECHNOLOGY**

Tumkur-572103



**Speech Processing Lab Report**

Submitted to:

**Dr.Veena Karjigi**

**Associate Professor**

**Department of Electronics and Communication**

Submitted by:

**Ritabrata Sarkar 1SI19EC080**

**1.Generation of sinusoid**

**Code:**

import numpy as np

import matplotlib.pyplot as plt

#initial parameters

Fs=8000 #sampling rate

fm=1000 #frequency of sinusoid

tlen=1.0 #length in second

#generate time axis

tt=np.arange(np.round(tlen\*Fs))/float(Fs)

#generate

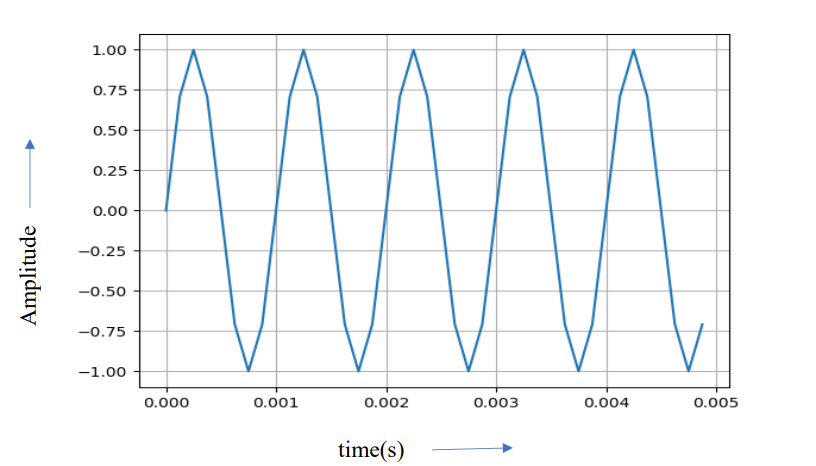
sine xt=np.sin(2\*np.pi\*fm\*tt)

#print the first 24 values of x(t)

plt.plot(tt[:40],xt[:40])

plt.grid() plt.show()

**Output:**

****

**2. DFT of a signal**

**Code:**

import numpy as np

import matplotlib.pyplot as plt

from array import \*

x=array(‘1’,[1,2,3,4])

N=len(x) n=np.arange(N)

k=n.reshape((N,1))

e=np.exp(-2j\*np.pi\*k\*n/N)

X=np.dot(e,x)

W=2\*np.pi\*n/N

print(n)

print(k)

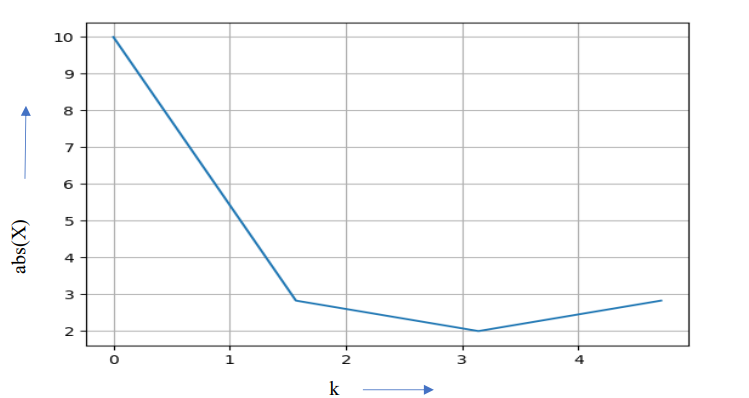
print€

print(X)

plt.plot(W,np.abs(X))

plt.grid()

plt.show()

**Output: **

**3.Spectrum**

**Code:**

from scipy.io import wavfile

import scipy.io

import numpy as np

import math

from scipy.signal import hann

from scipy.fftpack import fft

import matplotlib.pyplot as plt

samplerate, data = wavfile.read(r'C:\Users\ritab\Desktop\speech\_lab\fa.wav')

print(samplerate)

window = hann(40)

data\_win=np.multiply(data[:40],window)

mags = abs(fft(data\_win))

mags\_dB=20\*np.log10(mags)

N=len(data\_win)

k=np.arange(N)

f=k\*samplerate/N

plt.plot(f,mags\_dB)

plt.ylabel("Magnitude (dB)")

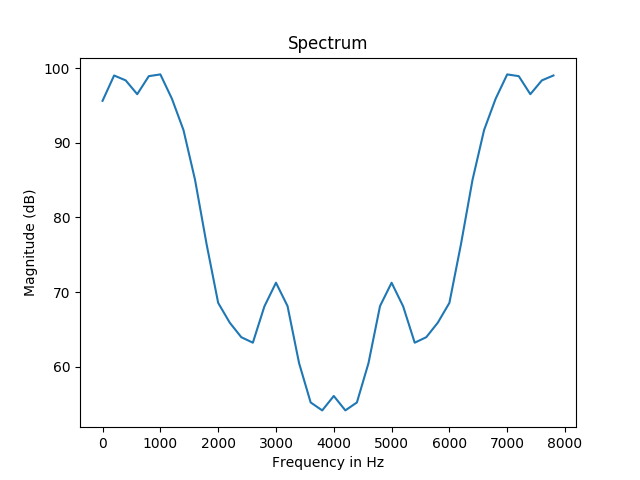
plt.xlabel("Frequency in Hz")

plt.title("Spectrum")

plt.show()

**Output:**

Sample Rate=8KHZ

****

**4.Spectrogram**

**Code:**

from scipy.io import wavfile

import scipy.io

import numpy as np

import math

from scipy.fftpack import fft

from scipy.signal import hann

import matplotlib.pyplot as plt

samplerate, data = wavfile.read(r'C:\Users\ritab\Desktop\speech\_lab\fa.wav')

print(len(data)/samplerate)

window=hann(240)

nvrlp=120

nframes=len(data)/nvrlp

nframes=int(nframes)

nframes=nframes-1

rows=nframes

frames=np.arange(nframes)

dft\_frame=[[0]\*512]\*rows

time=[[0]\*1]\*rows

for i in frames:

frame\_start=i\*nvrlp

data\_select=data[frame\_start:frame\_start+240]

data\_win=np.multiply(data\_select,window)

dft\_frame[i]=20\*np.log10(np.abs(fft(data\_win,512)))

time[i]=i\*(120/8000)

print(np.shape(dft\_frame))

dft\_frame\_t=np.transpose(dft\_frame)

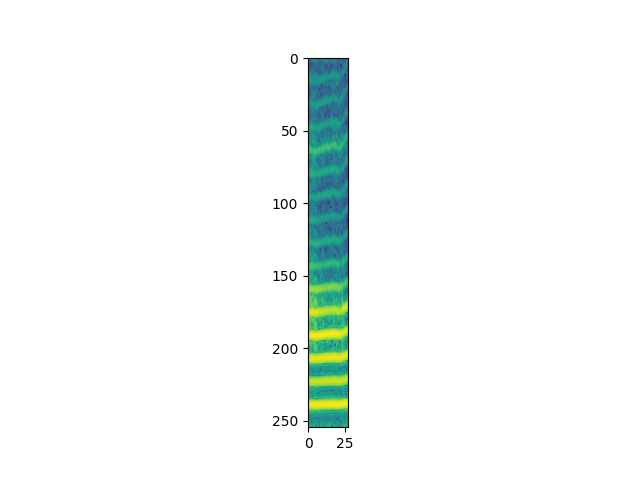
plt.imshow(dft\_frame\_t[257:512])

plt.show()

**Output:**

print(len(data)/samplerate)=0.441

print(np.shape(dft\_frame))= (28, 512)



**5.Energy and ZCR:**

**Code:**

from scipy.io import wavfile

import scipy.io

import numpy as np

import math

from scipy.signal import hann

import matplotlib.pyplot as plt

samplerate, data =wavfile.read(r'C:\Users\ritab\Desktop\speech\_lab\fa.wav')

print(len(data)/samplerate)

window=hann(240)

nvrlp=120

nframes=len(data)/nvrlp

nframes=int(nframes)

nframes=nframes-1

cols=240

rows=nframes

frames=np.arange(nframes)

samples=np.arange(cols)

Energy=[[0]\*1]\*rows

ZCR=[[0]\*1]\*rows

time=[[0]\*1]\*rows

for i in frames:

frame\_start=i\*nvrlp

data\_select=data[frame\_start:frame\_start+240]

data\_win=np.multiply(data\_select,window)

Energy[i]=sum(pow(data\_win,2))

for j in samples:

ZCR[i]=ZCR[i]+abs(np.sign(data\_win[j])-np.sign(data\_win[j-1]))

time[i]=i\*(120/8000)

signal\_samples=np.arange(len(data))/samplerate

plt.subplot(311)

plt.plot(signal\_samples,data)

plt.subplot(312)

plt.plot(time,Energy)

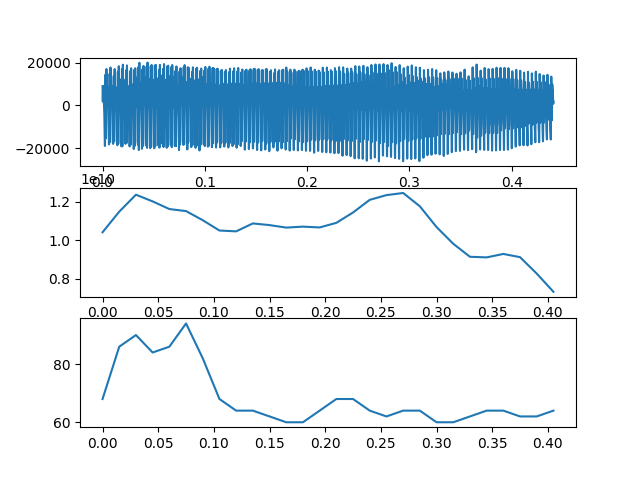
plt.subplot(313)

plt.plot(time,ZCR)

plt.show()

**Output:**

print(len(data)/samplerate)=0.441



**6.Linear Predictive Co-efficients:**

**Code:**

from scipy.io import wavfile

import scipy.io

import numpy as np

import math

from scipy.fftpack import fft

from scipy.signal import hann

import matplotlib.pyplot as plt

from scipy.linalg import solve\_toeplitz, toeplitz

samplerate, data = wavfile.read(r'C:\Users\ritab\Desktop\speech\_lab\fa.wav')

frame=data[:480]/32768

lp\_order=10;

len\_sig=len(frame)

auto\_corr=np.correlate(frame,frame,"full")

st=len\_sig-1

en=2\*len\_sig-1

norm\_auto=auto\_corr[st:en]/auto\_corr[st]

b=np.array((norm\_auto[1:lp\_order+1]))

c=norm\_auto[:lp\_order];

r=c;

lpcoeffs=solve\_toeplitz((c,r),b)

print(lpcoeffs)

**Outputs:**

[ 2.14681112 ,-1.89570141,0.53654544, -0.09261142, 0.76286507,

-0.86370883, 0.11689183, 0.35352519, -0.19824816 ,-0.01905677]

**7.Cepstrum of given signal**

**Code:**

from scipy.io import wavfile

import scipy.io

import numpy as np

import math

from scipy.signal import hann

from scipy.fftpack import fft

from scipy.fftpack import ifft

import matplotlib.pyplot as plt

samplerate, data = wavfile.read(r'C:\Users\ritab\Desktop\speech\_lab\fa.wav')

print(samplerate)

mags = np.abs(fft(data[:480],1024))

log\_mags=np.log(mags)

ceps=ifft(log\_mags)

t=(np.arange(480))/samplerate

#print(np.abs(ceps[:480])

plt.plot(t,np.abs(ceps[:480]))

plt.show()

**Output:**

