Sampling of Signals

Utkarsh Mahajan 201EC164

Arnav Raj 201EC109

Q1. Generate the following signals with the specified sampling rate and listen to it. Let t vary from $0-3$seconds. Let the sampling rate is 8000Hz or a sampling time period is $125\mu s$ (A sample is taken at every integer multiple of $125\mu s$)

1. $x(t) = sin(100\pi t)$ (50 Hz sinusoid)
2. $x(t) = cos(1000\pi t)$ (500Hz sinusoid)
3. $x(t) = sin(100\pi t^2)$ (Called a chirp signal)

To listen to a signal, write the signal to a file using wave write command from scipy Please look up the documentation here (https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.io.wavfile.write.html). Then download the signal and listen to it.

In [122…

```
import numpy as np
import scipy.io.wavfile as wav
import matplotlib.pyplot as plt
import IPython.display as ipd

# Set the sampling rate
rate = 8000;

# Genearate the t values use np.arange
t = np.arange(0,3,(1.0/8000))

#signal 1
#Generate signal
s1 = np.sin(100*np.pi*t)
#Write the signal
wav.write('sin100.wav',rate, s1)
#playing the audio here
ipd.Audio(s1, rate=rate)
```

Out[122]:

0:00 / 0:03

In [123…

```
#signal 2
#Generate signal
s2 = np.cos(1000*np.pi*t)
#Write the signal
wav.write('cos100.wav',rate, s1)
#playing the audio here
ipd.Audio(s2, rate=rate)
```

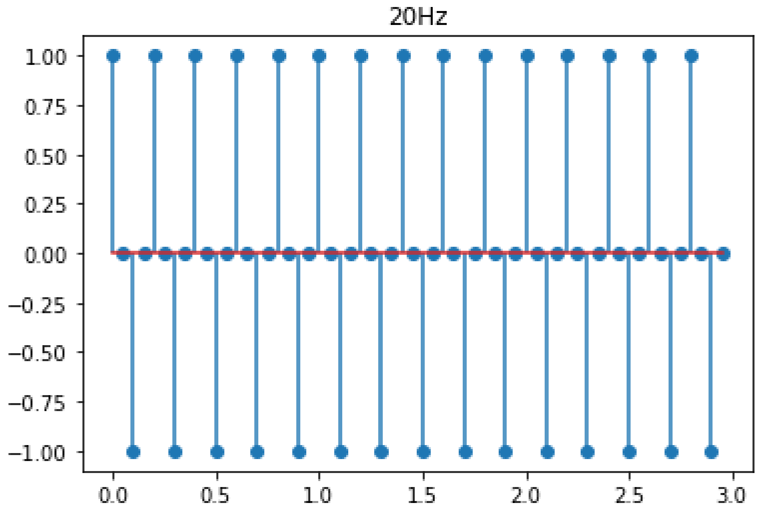
Out[123]: 0:00 / 0:03

In [124…
```
#signal 3
#Generate signal
s2 = np.sin(100*np.pi*t*t)
#Write the signal
wav.write('sin100t.wav',rate, s1)
#playing the audio here
ipd.Audio(s2, rate=rate)
```

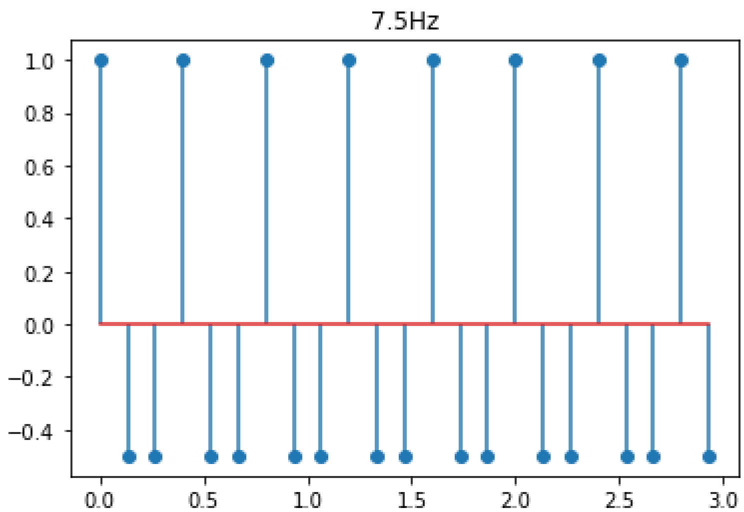
Out[124]: 0:00 / 0:03

Q2. Sample the signal $cos(10\pi t)$ using the following frequencies (a) 20 Hz (b) 7.5 Hz (c) 5 Hz (d) 2.5 Hz. In each case, plot the signal and determine its period.

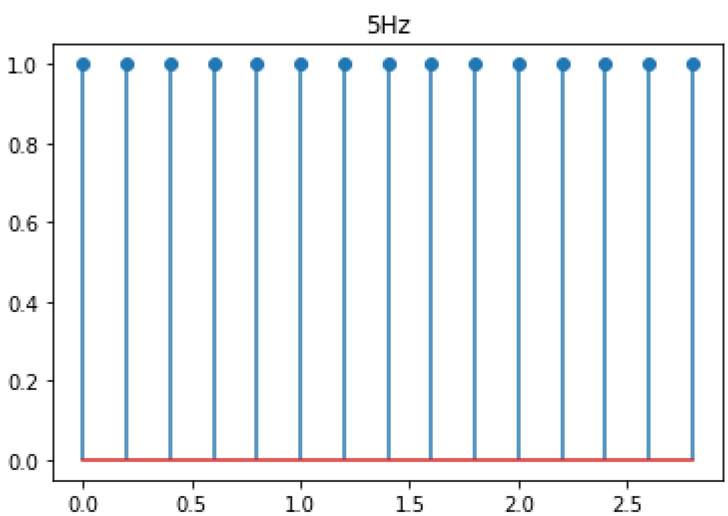
In [125…
```
import numpy as np
import matplotlib.pyplot as plt

def cosfuncplot(freq):
    # Genearate the t values use np.arange
    t = np.arange(0,3,(1.0/freq))
    #signal 1
    #Generate signal
    s = np.cos(10*np.pi*t)
    plt.stem(t, s)
    plt.title(str(freq) + 'Hz')
    plt.show()
    print('The Period of the signal is '+ str(freq/5) + ' units') # Tp = T/fre

# a) 20Hz
cosfuncplot(20);
# b) 7.5Hz
cosfuncplot(7.5);
# c) 20Hz
cosfuncplot(5);
# d) 7.5Hz
cosfuncplot(2.5);
```



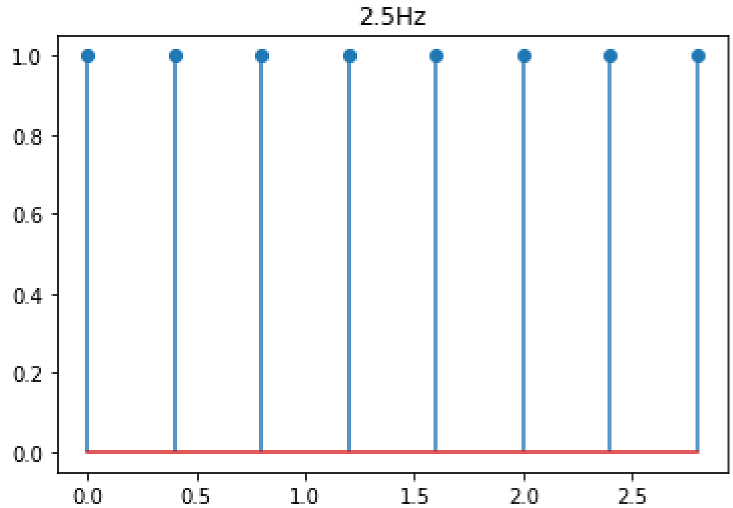

```
The Period of the signal is 4.0 units
```




```
The Period of the signal is 1.5 units
```




```
The Period of the signal is 1.0 units
```




```
The Period of the signal is 0.5 units
```

Q3. In DTMF dialling a number is represented by a dual frequency tone. Do a web search and find the frequencies of each digit. Generate DTMF tones corresponding to the telephone number 08242474040 by sampling the sum of sinusoids at the required frequencies at Fs = 8192 Hz. Concatenate the signals by putting 100 zeros between each signal (to represent silence) and listen to the signal. (Must sound like tone dialling the number from a phone)

In [126…

```
import numpy as np
import scipy.io.wavfile as wav
import IPython.display as ipd
rate=8192
def dtmf(i):
    t = np.arange(0,0.5,(1.0/rate))
    if(i==0):
        return ( np.sin(2*np.pi*1336*t) + np.sin(2*np.pi*941*t))
    elif(i==1):
        return ( np.sin(2*np.pi*1209*t) + np.sin(2*np.pi*697*t))
    elif(i==2):
        return ( np.sin(2*np.pi*1336*t) + np.sin(2*np.pi*697*t))
    elif(i==3):
        return ( np.sin(2*np.pi*1447*t) + np.sin(2*np.pi*697*t))
    elif(i==4):
        return ( np.sin(2*np.pi*1209*t) + np.sin(2*np.pi*770*t))
    elif(i==5):
        return ( np.sin(2*np.pi*1336*t) + np.sin(2*np.pi*770*t))
    elif(i==6):
        return ( np.sin(2*np.pi*1447*t) + np.sin(2*np.pi*770*t))
    elif(i==7):
        return ( np.sin(2*np.pi*1209*t) + np.sin(2*np.pi*852*t))
    elif(i==8):
        return ( np.sin(2*np.pi*1336*t) + np.sin(2*np.pi*852*t))
    elif(i==9):
        return ( np.sin(2*np.pi*1447*t) + np.sin(2*np.pi*852*t))

silence = np.zeros(100)
#test signal
zero = dtmf(0);
wav.write('zero.wav', rate, zero)
ipd.Audio(zero, rate=rate)
```

```
# for given telephone no
tele = np.concatenate((dtmf(0),silence, dtmf(8), silence,
                       dtmf(2),silence, dtmf(4),silence,
                       dtmf(2),silence, dtmf(4),silence,
                       dtmf(7),silence, dtmf(4),silence,
                       dtmf(0),silence, dtmf(4),silence, dtmf(0)))
print(tele)
# writing it to a wav file
wav.write('tele.wav', rate, tele)
#playing it in jupyteer notebook
ipd.Audio(tele, rate=rate)
```

```
[ 0.          1.51524827  1.87954532 ...  0.7610309   0.10426608
 -0.1938677 ]
```

Out[126]: 0:00 / 0:05

Q4. Record your own voice saying vowel /a/ as in cat (It can be done using a sound recording program in the computer). Please save in *.wav format, nd read it using wavread command in Python for further processing). Please look https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.io.wavfile.read.html for documentation

(a) Find the sampling rate used by the recorder

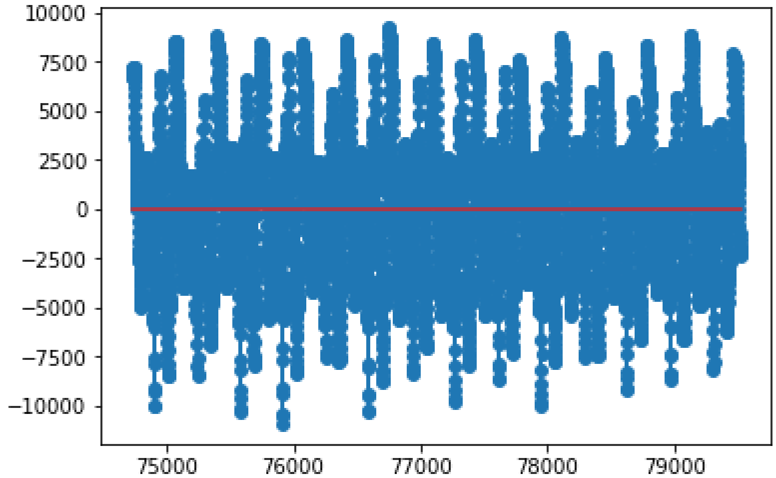
(b) Just zoom and plot only the middle 100 milliseconds of the data

(c) Find the mean (average value) and variance of the entire signal. Use np.mean and np.var commands

In [127…

```
import numpy as np
import scipy.io.wavfile as wav
import matplotlib.pyplot as plt
import scipy.signal as sig

rate, data = wav.read('aa.wav')
data = data[:,0]
print('(a)Sampling rate used by the recorder: ' + str(rate) + ' Hz')
datal = np.arange(0, len(data))
dl = int(len(data)/2 - (rate/20))
dr = int(len(data)/2 + (rate/20))
plt.stem(datal[dl:dr], data[dl:dr])
print('(b)')
plt.show()
print('(c)')
print('mean: ' + str(np.mean(data)) + ' Variance: '+ str(np.var(data)))
```

```
(a)Sampling rate used by the recorder: 48000 Hz
(b)
```



```
(c)
mean: 8.29575748145843 Variance: 15332052.125094287
```