

```
1 # Author: Ramesh Pai
2 # Affiliation: 201104047, TE-E&TC Engg, Sem.5, 2021-22, GCE.
3
4 # importing python modules
5 import matplotlib.pyplot as plt
6 from scipy.fft import fft, fftshift
7 import numpy as np
8 import random
9
10 # vc: Amplitude of Carrier Signal
11 # fc: frequency(Hz) of Carrier Signal
12 # Td: Number of bits in the Binary Data(Bipolar NRZ)
13
14 def values(vc, fc, Td):
15
16     fs = 600 #sampling frequency
17     dt = 1/fs #sample time interval or time-steps for time-domain signal
18     t = np.arange(0, 5, dt) #time indices for time-domain signal
19     n = np.size(t) #number of samples
20     df = fs/n #frquency interval or frequency-steps for frequency-spectrum
21     f = np.arange(-fs/2, fs/2, df) #frequency indices for frquency-spectrum
22
23     # plot1: Carrier Signal v/s Time
24
25     v_c = vc*np.sin(2*np.pi*fc*t) #Carrier Signal
26     plt.subplot(2, 2, 1)
27     plt.plot(t, v_c)
28     plt.title("Carrier Signal", loc='left')
29     plt.xlabel("t(sec)", loc='right')
30     plt.ylabel("v_c(Volts)")
31
32     # plot2: Modulating Signal(Bipolar NRZ) v/s Time
33
34     #list for storing random input bit sequence generated
35     bitData = []
36     for x in range(Td):
37         x = random.randint(0, 1)
38         bitData.append(x)
39
40     print(f"input bit sequence: {bitData}") #printing the input bit sequence
41
42     #number of points in the time duration
43     Td_len = int(len(t)/Td)
44
45     #list for storing number of points in time duration
46     Td_arr = []
47     for x in range(Td+1):
48         x *= Td_len
49         Td_arr.append(x)
50
51     sqSignal = [] #Modulating Signal(Bipolar NRZ)
52     for x in range(len(bitData)):
53         if bitData[x] == 1:
54             x = [1]*(Td_arr[x+1] - Td_arr[x])
55             sqSignal.extend(x)
56         else:
57             x = [-1]*(Td_arr[x+1] - Td_arr[x])
58             sqSignal.extend(x)
59
```

```
60 sqSignal = np.array(sqSignal) #converting list into numpy array
61 plt.subplot(2, 2, 2)
62 plt.plot(t, sqSignal)
63 plt.title(f"Modulating Signal (Binary Data: {bitData})", loc='left')
64 plt.xlabel("t(sec)", loc='right')
65 plt.ylabel("sqSignal(Volts)")
66
67 # plot3: FSK(Frequency Signal Keying) Signal v/s Time
68
69 fd = 1 #frequency deviation
70 v_f = vc*np.sin(2*np.pi*(fc + sqSignal*fd)*t)
71 plt.subplot(2, 2, 3)
72 plt.plot(t, sqSignal, color = 'r', linestyle = 'dotted', label = 'modulating
signal')
73 plt.plot(t, v_f, label = 'FSK Signal')
74 plt.title("FSK Signal", loc='left')
75 plt.xlabel("t(sec)", loc='right')
76 plt.ylabel("v_f(Volts)")
77
78 # plot4: Spectrum of the FSK(Frequency Shift Keying) Signal v/s Frequency
79
80 v_f_spec = fftshift(fft(v_f)) #FFT of FSK Signal(Complex in nature)
81 plt.subplot(2, 2, 4)
82 plt.plot(f, abs(v_f_spec)/n) #Plotting frequency indices v/s Normalised magnitude
of FFT ASK signal
83 plt.xlim(-20, 20)
84 plt.title("Spectrum of FSK Signal", loc='left')
85 plt.xlabel("frequency(Hz)", loc='right')
86 plt.ylabel("Magnitude")
87
88 plt.subplot_tool()
89 plt.show()
90
91 values(1, 3, 5) #assigning values to parameters
```