

```
1 # Author: Ramesh Pai
2
3 # Affiliation: 201104047, TE-E&TC Engg, Sem.5, 2021-22, GCE.
4
5 # importing python modules
6 import matplotlib.pyplot as plt
7 import numpy as np
8 from scipy.fft import fft, fftshift
9
10 # vc: amplitude of Carrier Signal
11 # m: modulation index of FM Signal
12 # fc: frequency(Hz) of Carrier Signal
13 # fm: frequency(Hz) of Message Signal
14
15 def values(vc, m, fc, fm):
16
17     fs = 60000 #sampling frequency
18     dt = 1/fs #sample time interval or time-steps for time-domain signal
19     t = np.arange(0, 1, dt) #time indices for time-domain signal
20     n = np.size(t) #number of samples
21     df = fs/n #frequency interval or frequency-steps for frequency-spectrum
22     f = np.arange(-fs/2, fs/2, df) #frequency indices for frequency-spectrum
23
24     # plot1: Tone modulated FM Signal v/s Time
25
26     v_fm = vc*np.cos(2*np.pi*fc*t + m*np.sin(2*np.pi*fm*t)) #FM Signal
27     plt.subplot(3, 1, 1)
28     plt.plot(t, v_fm)
29     plt.title("Frequency Modulated Signal", loc='left')
30     plt.xlabel("time(sec)", loc='right')
31     plt.ylabel("v_fm(volts)")
32
33     # plot2: Spectrum of FM signal v/s Frequency
34
35     xf_1 = fftshift(fft(v_fm)) #FFT of FM Signal(Complex in nature).
36     plt.subplot(3, 1, 2)
37     plt.plot(f, abs(xf_1)/n) #Plotting frequency indices v/s Normalised magnitude of
    FFT of FM signal
38     plt.xlim(-200, 200)
39     plt.title("Spectrum of FM signal", loc='left')
40     plt.xlabel("frequency(Hz)", loc='right')
41     plt.ylabel("Magnitude")
42
43     # plot3: Spectrum of FM signal v/s Frequency(ZOOMED)
44
45     xf_1 = fftshift(fft(v_fm)) #FFT of FM Signal(Complex in nature).
46     plt.subplot(3, 1, 3)
47     plt.plot(f, abs(xf_1)/n) #Plotting frequency indices v/s Normalised magnitude of
    FFT of FM signal
48     plt.xlim(30, 70) #for (mf = 1)
49     # plt.xlim(10, 90) #for (mf = 2.4)
50     # plt.xlim(10, 90) #for (mf = 4)
51     plt.title("Spectrum of FM signal(ZOOMED)", loc='left')
52
53
54     if m == 1:
55         plt.title(f"(mf={m}) Bandwidth={fm*4} Hz", loc='right', fontsize = 10,
    fontstyle = 'italic')
56     elif m == 2.4:
```

```
57     plt.title(f"(mf={m}) Bandwidth={fm*6} Hz", loc='right', fontsize = 10,  
fontstyle = 'italic')  
58     elif m == 4:  
59         plt.title(f"(mf={m}) Bandwidth={fm*10} Hz", loc='right', fontsize = 10,  
fontstyle = 'italic')  
60  
61  
62     plt.xlabel("frequency(Hz)", loc='right')  
63     plt.ylabel("Magnitude")  
64  
65  
66     plt.subplot_tool()  
67     plt.show()  
68  
69 values(2, 1, 50, 5) #assigning values to parameters(mf = 1)  
70 # values(2, 2.4, 50, 5) #assigning values to parameters(mf = 2.4)  
71 # values(2, 4, 50, 5) #assigning values to parameters(mf = 4)
```