

```

1  '''
2  authour: Mayur Kamat
3  affiliation: 201104032, TE-E&TC Engg. Sem V, 2021-22, GEC
4  last updated: 22/11/2022
5  '''
6
7  #importing necessary functions from libraries
8  from matplotlib import pyplot as plt
9  from matplotlib.widgets import Slider
10 from numpy import cos, real, abs
11 from numpy.fft import fft
12 from math import pi
13 from scipy.fft import fft
14 from scipy.signal import square
15 from plotconfig import *
16
17 #global (fig, ax) tuple, making it global makes it easier to update values and use
  GUI
18 fig1, ax = plt.subplots()
19
20 #keeps track of the currently displayed plot
21 CurrentGraph = 0
22
23 #plots, calculates and updates the signals using the global variables from plotconfig
24 #which are updated in the update functions below
25 def plotSingals():
26     global fig1, ax
27
28     #producing the message and carrier signals
29     vm = amp_vm*square(2*pi*fm*time) + amp_vm
30
31     #making a sync pulse (10000..)
32     vm[200:] = 0
33
34
35     vc = amp_vc*cos(2*pi*fc*time)
36
37     #FSK signal
38     vfsk = amp_vc*cos(2*pi*(fc+vm*fd)*time)
39
40     #calculating the FFT
41     spectrum = (fft(vfsk))
42
43     #functions below plot the singals
44     def plot_vm():
45         ax.clear()
46         ax.set_xlabel('time - (sec)')
47         ax.set_ylabel('amplitude - (volts)')
48         ax.set_title('message and quantized signal')
49         ax.plot(time, vm, 'b', label='Message')
50
51     def plot_vc():
52         ax.clear()
53         ax.set_xlabel('time - (sec)')
54         ax.set_ylabel('amplitude - (volts)')
55         ax.set_title('Carrier Signal')
56         ax.plot(time, vc, 'b', label='Carrier')
57
58     def plot_vfsk():
59         ax.clear()
60         ax.set_xlabel('time - (sec)')
61         ax.set_ylabel('amplitude - (volts)')
62         ax.set_title('FSK signal')
63         ax.plot(time, vfsk, 'r', label='Vfsk')
64
65     def plot_spectrum():
66         ax.clear()
67         ax.set_xlabel('frequency - (hertz)')
68         ax.set_ylabel('Amplitude - (volts)')
69         ax.set_title('FSK spectrum')
70         ax.plot(frequency, abs((real((spectrum)))/N), 'b', label='spectrum')
71         ax.set_xlim(-100, 100)
72
73
74     #dictionary to call the plotting functins as and when the graph slider value
  changes
75     GraphSelector = {
76         0 : plot_vm,

```

```
77     1 : plot_vc,
78     2 : plot_vfsk,
79     3 : plot_spectrum,
80     }
81
82     GraphSelector.get(CurrentGraph)()
83
84     #plot adjustments
85     fig1.tight_layout(h_pad=2)
86     fig1.set_size_inches(14, 7)
87     plt.subplots_adjust(bottom=0.4)
88
89     #draws the plot
90     ax.grid(True)
91     ax.legend()
92     plt.draw()
93
94
95 def update_graph(val):
96     global CurrentGraph
97     CurrentGraph = val
98     plotSingals()
99
100
101
102
103 #slider widgets
104 ax_graph = plt.axes([0.17, 0.27, 0.65, 0.03])
105 graph_Slider = Slider(ax_graph, 'Graph Select', valmin=0, valmax=3, valstep=1,
106                       valinit=0)
107
108 #plots the signal on run
109 plotSingals()
110
111 #handles updates on the sliders widgets
112 graph_Slider.on_changed(update_graph)
113
114 #needed in vscode to plot the fig in a new window...can be ignored in spyder
115 plt.show()
```