

```

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 plotconfig import *
15
16 #global (fig, ax) tuple, making it global makes it easier to update values and use
17 #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 = 0*time
30     vm[:120] = amp_vm
31     vc = amp_vc*cos(2*pi*fc*time)
32
33     #ASK signal
34     vask = vm*vc
35
36     #calculating the FFT
37     spectrum = (fft(vask))
38
39     #functions below plot the singals
40     def plot_vm():
41         ax.clear()
42         ax.set_xlabel('time - (sec)')
43         ax.set_ylabel('amplitude - (volts)')
44         ax.set_title('message and quantized signal')
45         ax.plot(time[:400], vm[:400], 'b', label='Message')
46
47     def plot_vc():
48         ax.clear()
49         ax.set_xlabel('time - (sec)')
50         ax.set_ylabel('amplitude - (volts)')
51         ax.set_title('Carrier Signal')
52         ax.plot(time[:400], vc[:400], 'b', label='Carrier')
53
54     def plot_vask():
55         ax.clear()
56         ax.set_xlabel('time - (sec)')
57         ax.set_ylabel('amplitude - (volts)')
58         ax.set_title('ASK signal')
59         ax.plot(time[:400], vask[:400], 'r', label='Vask')
60
61     def plot_spectrum():
62         ax.clear()
63         ax.set_xlabel('frequency - (hertz)')
64         ax.set_ylabel('Amplitude - (volts)')
65         ax.set_title('ASK spectrum')
66         ax.plot(frequency, abs((real((spectrum))))/N, 'b', label='spectrum')
67         ax.set_xlim(-1500, 1500)
68
69     #dictionary to call the plotting functins as and when the graph slider value
70     #changes
71     GraphSelector = {
72         0 : plot_vm,
73         1 : plot_vc,
74         2 : plot_vask,
75         3 : plot_spectrum,
76     }

```

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