```
'''
authour: Mayur Kamat
affiliation: 201104032, TE-E&TC Engg. Sem V, 2021-22, GEC
last updated: 22/11/2022
'''

#importing necessary functions from libraries
from matplotlib import pyplot as plt
from matplotlib.widgets import Slider
from numpy import sin, real, abs
from numpy.fft import fft
from math import pi
from scipy.fft import fft
from scipy.signal import square
from plotconfig import *

#global (fig, ax) tuple, making it global makes it easier to update values and use GUI
fig1, ax = plt.subplots()

#keeps track of the currently displayed plot
CurrentGraph = 0

#plots, calculates and updates the signals using the global variables from plotconfig
#which are updated in the update functions below
def plotSingals():
    global fig1, ax

    #producing the message and carrier signals
    vm = amp_vm*square(2*pi*fm*time)
    vm[200:] = -amp_vm #making a sync bit (10000)
    vc = amp_vc*sin(2*pi*fc*time)

    #PSK signal
    vpsk = vm*vc

    #calculating the FFT
    spectrum = (fft(vpsk))

    #functions below plot the singals
    def plot_vm():
        ax.clear()
        ax.set_xlabel('time - (sec)')
        ax.set_ylabel('amplitude - (volts)')
        ax.set_title('message and quantized signal')
        ax.plot(time[:400], vm[:400], 'b', label='Message')

    def plot_vc():
        ax.clear()
        ax.set_xlabel('time - (sec)')
        ax.set_ylabel('amplitude - (volts)')
        ax.set_title('Carrier Signal')
        ax.plot(time[:400], vc[:400], 'b', label='Carrier')

    def plot_vask():
        ax.clear()
        ax.set_xlabel('time - (sec)')
        ax.set_ylabel('amplitude - (volts)')
        ax.set_title('PSK signal')
        ax.plot(time[:400], vpsk[:400], 'r', label='Vpsk')

    def plot_spectrum():
        ax.clear()
        ax.set_xlabel('frequency - (hertz)')
        ax.set_ylabel('Amplitude - (volts)')
        ax.set_title('PSK spectrum')
        ax.plot(frequency, abs((real((spectrum)))/N), 'b', label='spectrum')
        ax.set_xlim(-300, 300)


    #dictionary to call the plotting functins as and when the graph slider value changes
    GraphSelector = {
        0 : plot_vm,
        1 : plot_vc,
        2 : plot_vask,
        3 : plot_spectrum,
        }
```

```
77
78      GraphSelector.get(CurrentGraph)()
79
80      #plot adjustments
81      fig1.tight_layout(h_pad=2)
82      fig1.set_size_inches(14, 7)
83      plt.subplots_adjust(bottom=0.4)
84
85      #draws the plot
86      ax.grid(True)
87      ax.legend()
88      plt.draw()
89
90
91  def update_graph(val):
92      global CurrentGraph
93      CurrentGraph = val
94      plotSingals()
95
96
97
98
99  #slider widgets
100 ax_graph = plt.axes([0.17, 0.27, 0.65, 0.03])
101 graph_Slider = Slider(ax_graph, 'Graph Select', valmin=0, valmax=3, valstep=1,
    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()
111
```