LAB TITLE AND CODE : SOFTWARE DEFINED RADIO LAB 19CCE284

EXPERIMENT NUMBER : 1

DATE : 18/04/2022 (MONDAY)

SAMPLING OF LOW PASS SIGNALS

* **AIM:**

Generating signal and sketching the spectrum for various sampling rates. Perform the analysis for both low pass and band limited signal of bandwidth B.

* **SOFTWARE REQUIRED:**

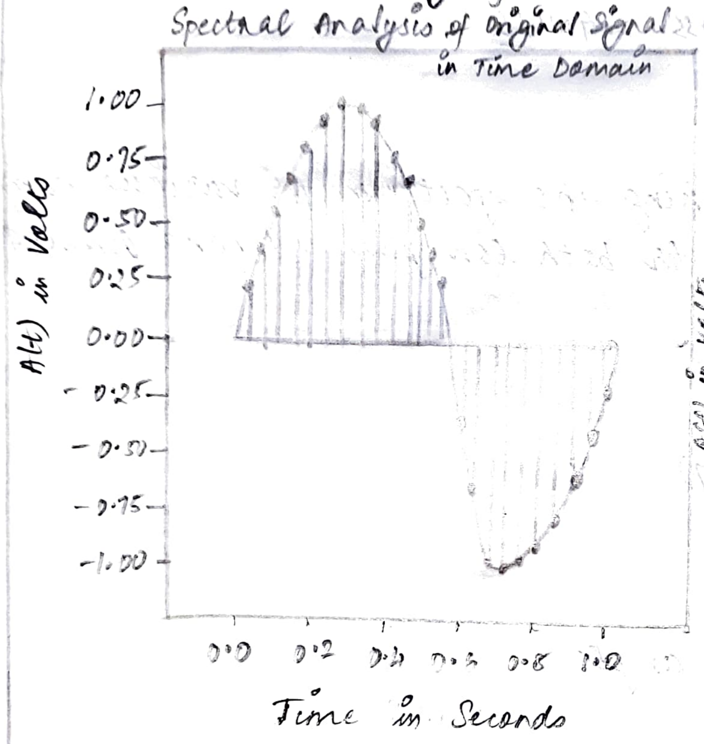
Spyder IDE (Python 3.9.7)

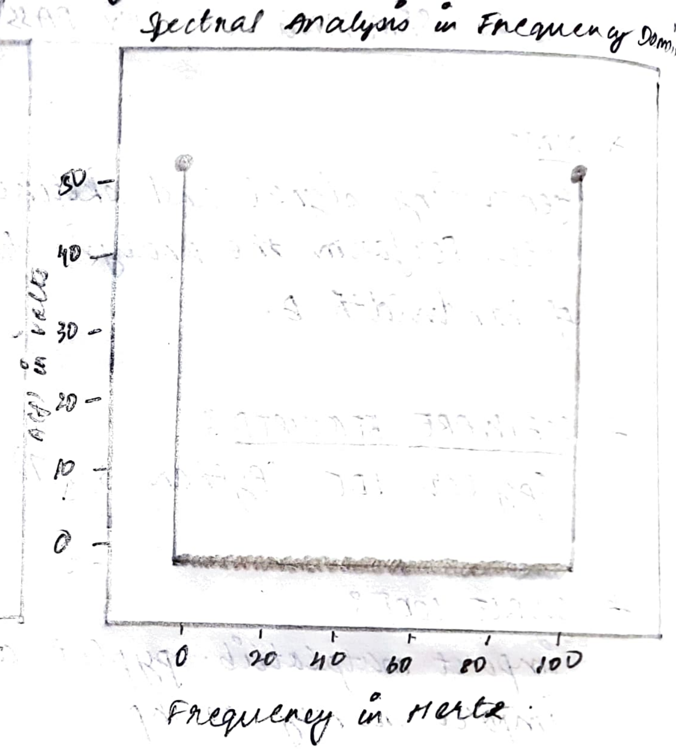
* **SOURCE CODE:**

```
import matplotlib.pyplot as plt
import numpy as np

# Compute DFT coefficients using linear transformation method:
def DFT(x):

    # Compute w(N) 1D Array:
    r1 = c1 = len(x)
    wn = []
    for i in range(r1):
        for j in range(c1):
            wn.append(np.exp(-2j * np.pi * i * j / len(x)))

    # numpy.reshape() is used to give a new shape to an array
    without changing its data.

    wn_multidim = np.reshape(wn, (r1, c1)) # An N*N w(N) matrix
    r2 = len(x); c2 = 1
    x_multidim = np.reshape(x, (r2, c2)) # An N*1 x(N) matrix
```

* CONSOLE WINDOW OUTPUT :

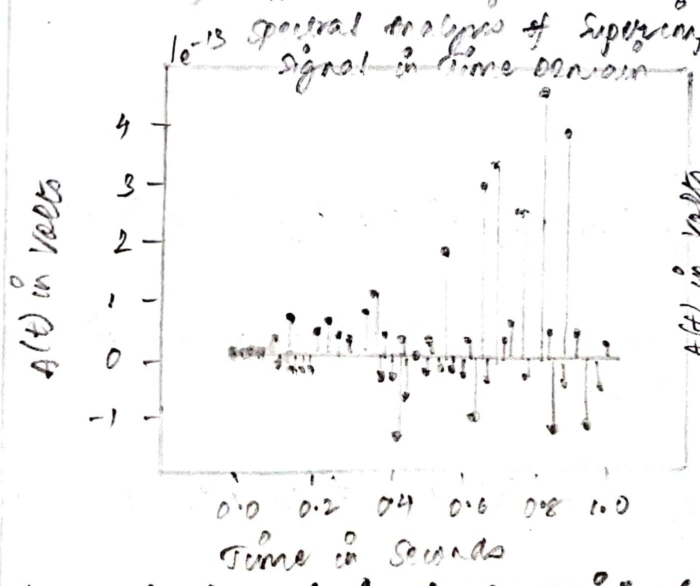
Enter the desired sampling frequency (in Hertz) : 100
enter the frequency of the sine signal (in Hertz) : 1

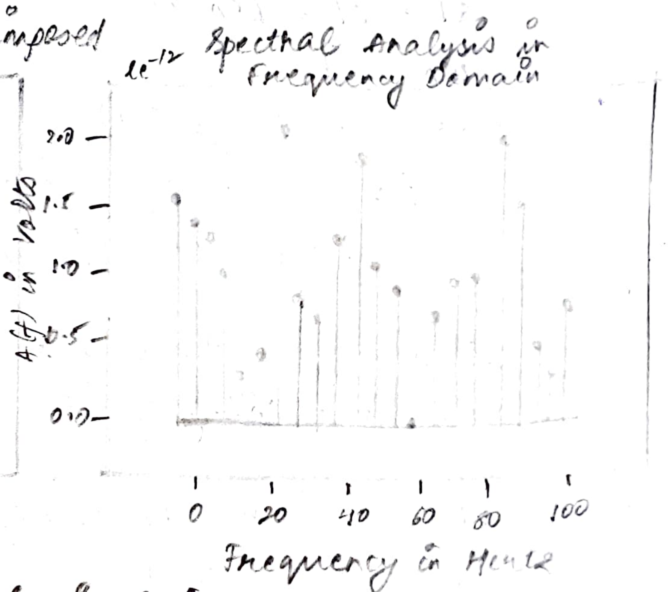




The spectral analysis of the original signal in time and frequency domain is shown in the above plot.

Enter the bandwidth for the analysis of the low pass signal (in Hertz) : 100



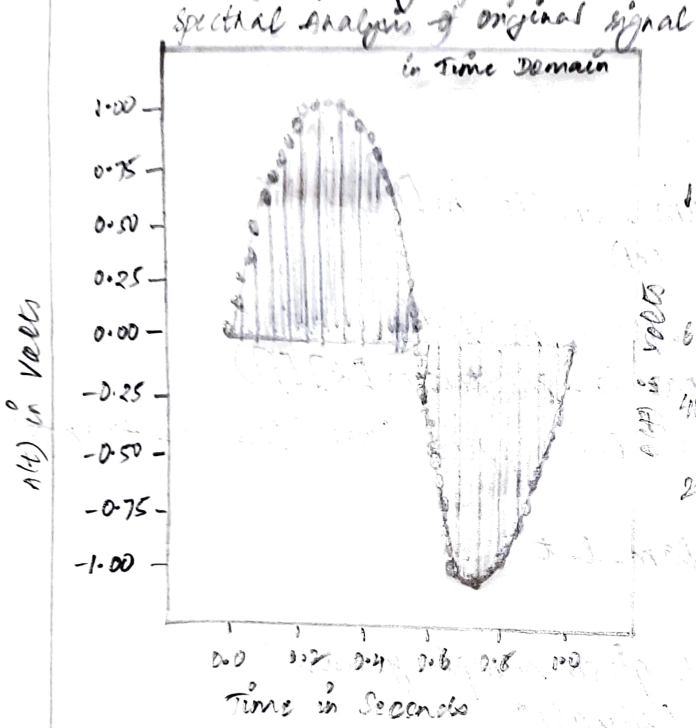


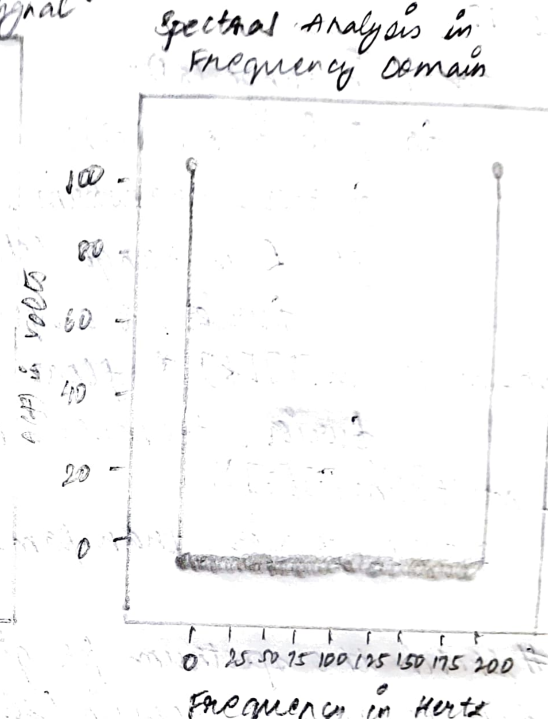
The spectral analysis of the original signal in time and frequency domain is shown in the above plot.

```
# Compute X(N) = W(N) * x(N), an N*1 matrix
    fourier_transform_multidim = [[0] * c2] * r1 # Null
Multidimensional Array
    fourier_transform_l_t = [] # Convert Multidimensional Array
to 1D
    for i in range (r1):
        for j in range (c2):
            fourier_transform_multidim [i][j] = 0
            for k in range (c1):
                fourier_transform_multidim [i][j] +=
wn_multidim [i][k] * float (x_multidim [k][j])
            fourier_transform_l_t.append (abs (fourier_transform
_multidim [i][j]))
    return fourier_transform_l_t

# Sketch the spectrum for given sampling rate:
def frequency_domain (signal):

    length_of_signal = len (signal)
    time_period = length_of_signal / fs
    temp_frequency_axis = np.arange (length_of_signal)
    frequency_axis = temp_frequency_axis / time_period

    plt.xlabel ("Frequency Domain")
    plt.ylabel (" A(f) ")
    plt.title ("Spectral Analysis in Frequency Domain")
    plt.stem (frequency_axis, signal)
    plt.grid (True)
    plt.show ()

# Generate the sine signal:
fs = int (input ("Enter the desired sampling frequency in Hertz: "))
time_interval = 1/ fs
```

* CONSOLE WINDOW OUTPUT:

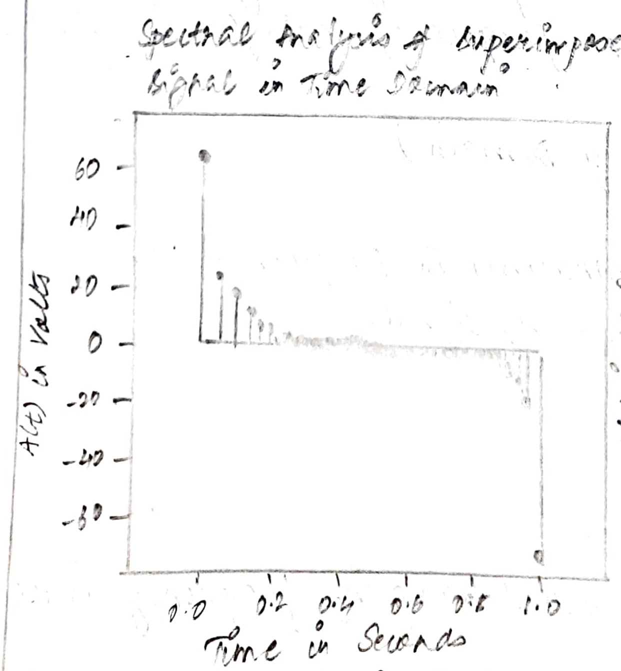
Enter the desired sampling frequency (in Hertz): 200
Enter the frequency of the sine signal (in Hertz): 1

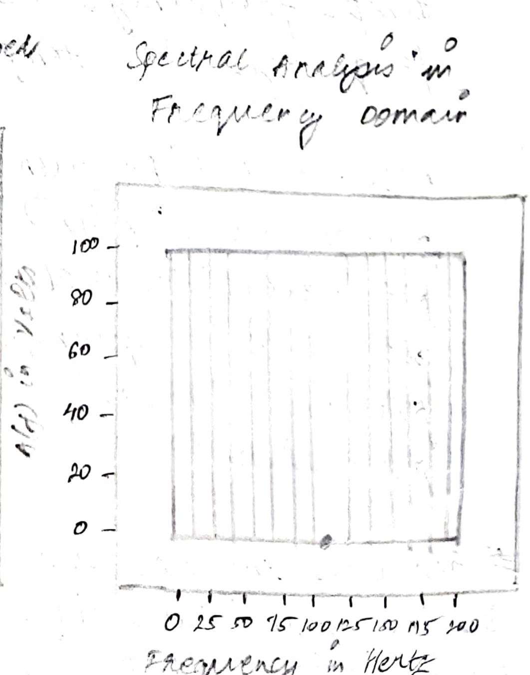




The spectral analysis of the original signal in time and frequency domain is shown in the above plot.

Enter the bandwidth for the analysis of the low pass signal (in Hertz): 100

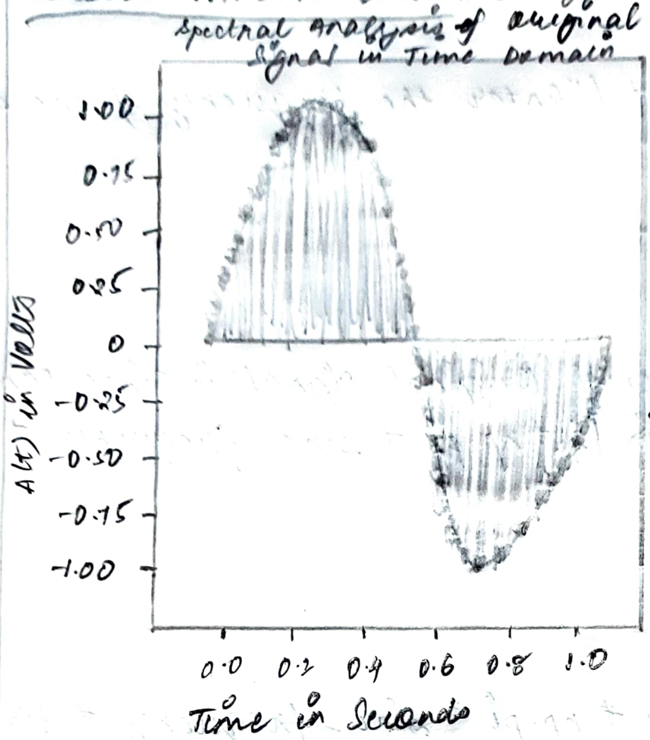


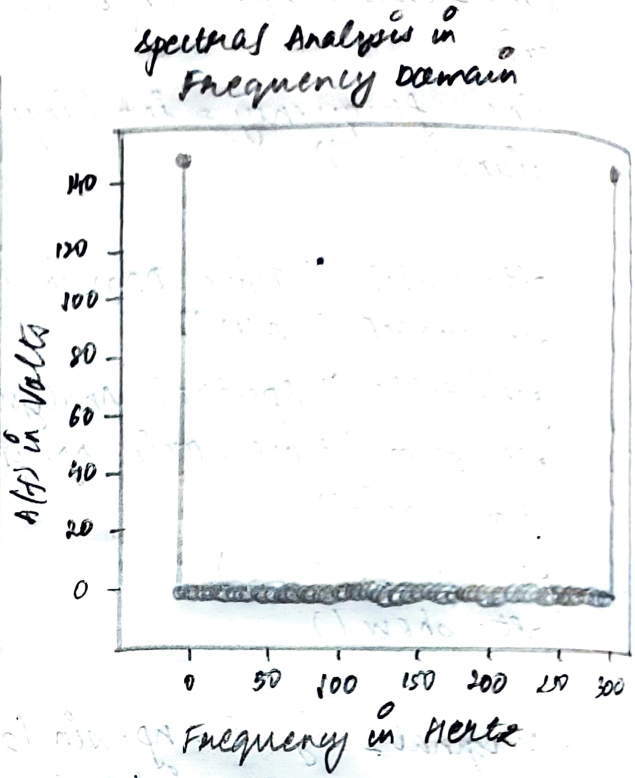


The spectral analysis of the superimposed signal in time and frequency domain is shown in the above plot.

```
time_axis = np.arange (0, 1, time_interval) # Define the time
axis
sine_frequency = int (input ("Enter the frequency of the sine
signal : "))

plt.xlabel ("Time Domain")
plt.ylabel ("A(t)")
plt.title ("Spectral Analysis of Original Signal in Time Domain")
plt.stem (time_axis, np.sin (2 * np.pi * sine_frequency *
time_axis))
plt.grid (True)
plt.show ()

original_signal = np.sin (2 * np.pi * sine_frequency * time_axis)
frequency_domain(DFT (original_signal))
print ("\nThe spectral analysis of the original signal in frequency
domain is shown in the above plot")

# Perform the analysis for low pass signal of given bandwidth:
superimposed_signal = 0
for i in range (1, 101, 1):
    superimposed_signal = superimposed_signal + np.sin (2 *
np.pi * i * time_axis)

plt.xlabel ("Time Domain")
plt.ylabel ("A(t)")
plt.title ("Spectral Analysis of Superimposed Signal in Time
Domain")
plt.stem (time_axis, superimposed_signal)
plt.grid (True)
plt.show ()
```

+ CONSOLE WINDOW OUTPUT :



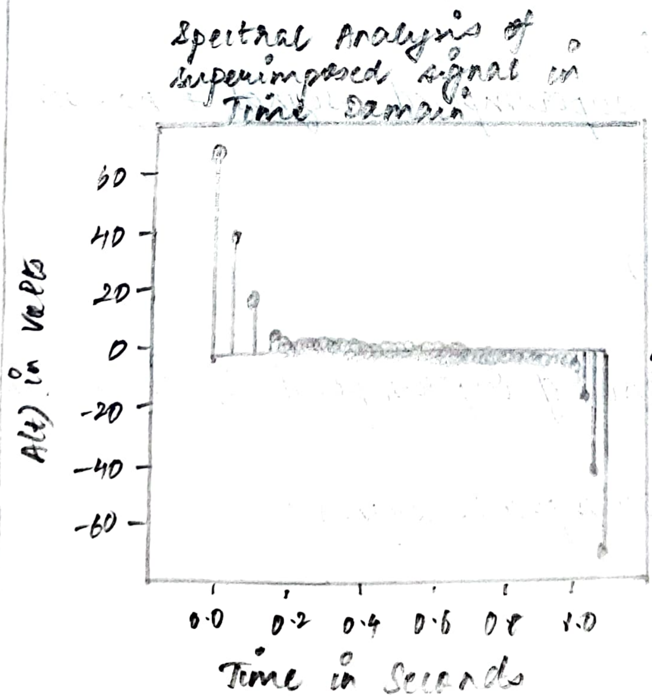


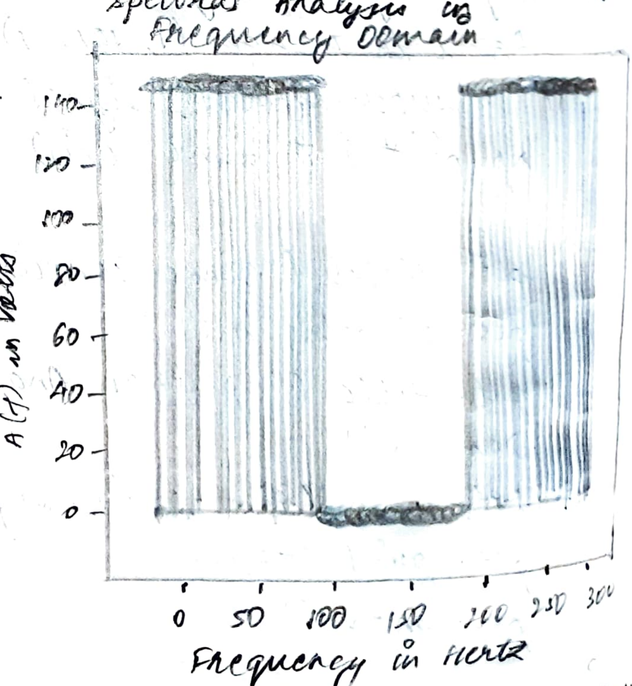
enter the desired sampling frequency (in Hertz) : 300

Enter the frequency of the sine signal (in Hertz) : 1

The spectral analysis of the original signal in time and frequency domain is shown in the above plot.

enter the bandwidth for the analysis of the low pass signal (in Hertz) : 10





The spectral analysis of the superimposed signal in time and frequency domain is shown in the above plot.

frequency_domain (DFT (superimposed_signal))
print ("In The spectral analysis of the superimposed signal in the frequency domain is shown in the above plot.")

* INFERENCE:

Generated signal and sketched the spectrum for various sampling rates. Also, performed the analysis for low pass signal of given bandwidth. All the simulation results were verified successfully.