LAB TITLE AND CODE : SOFTWARE DEFINED RADIO LAB 19CLE284
EXPERIMENT NUMBER : 2
DATE : 25/04/2022, MONDAY

INTERPOLATION AND DECIMATION – TIME AND FREQUENCY DOMAIN ANALYSIS

* **AIM :**
Generate signals of definite frequency and apply sampling rate conversion using decimation and interpolation. study the time domain and frequency domain characteristics.

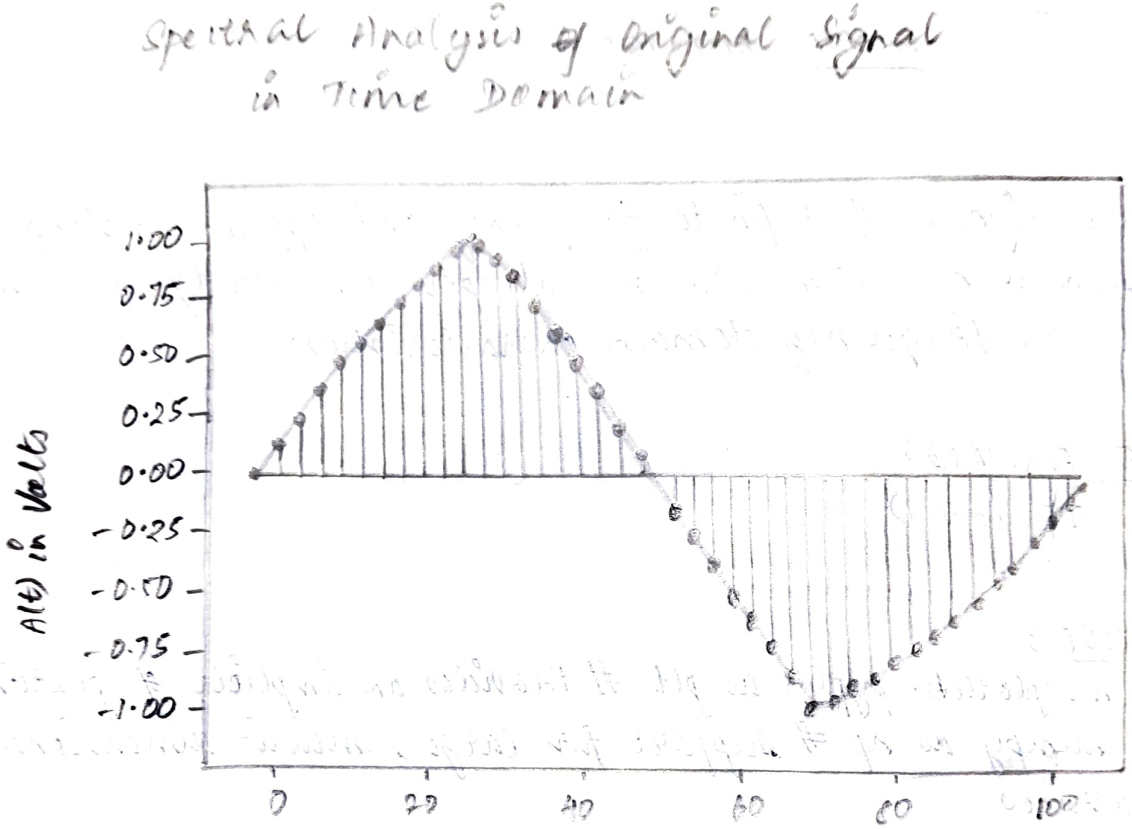
* **SOFTWARE REQUIRED :**
spyder IDE (3.9.7)

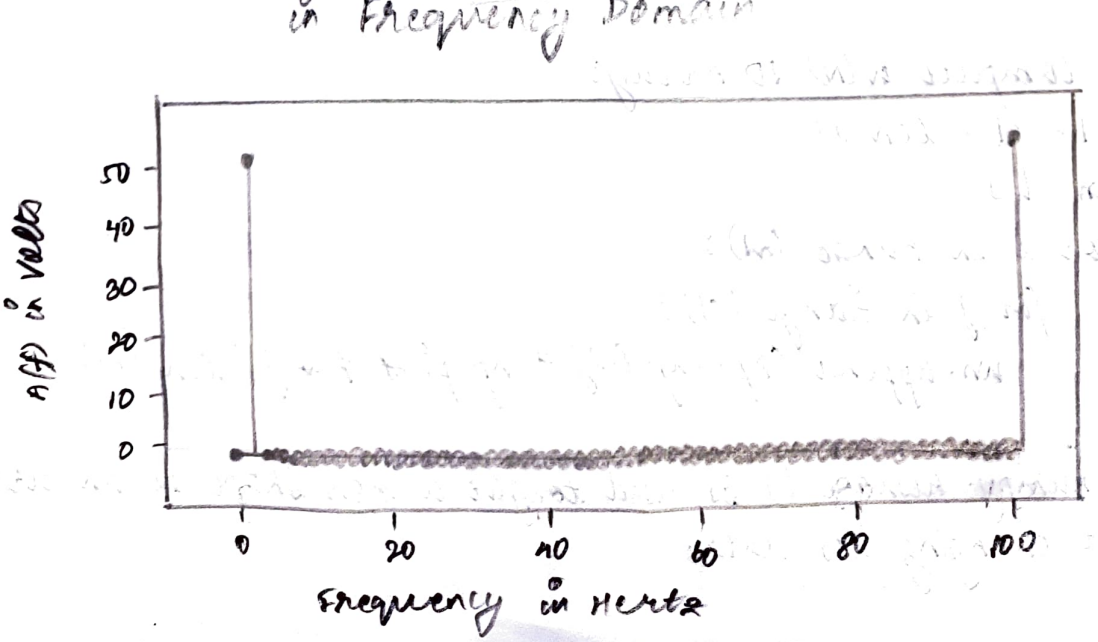
* **SOURCE CODE :**

```
import matplotlib.pyplot as plt # Provides an implicit of plotting
import numpy as np # support for large, multi-dimensional array and matrices

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

    # Compute W(N) 1D Array:
    x1 = c1 = len(x)
    wn = []
    for i in range(x1):
        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.
```

OUTPUT :

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

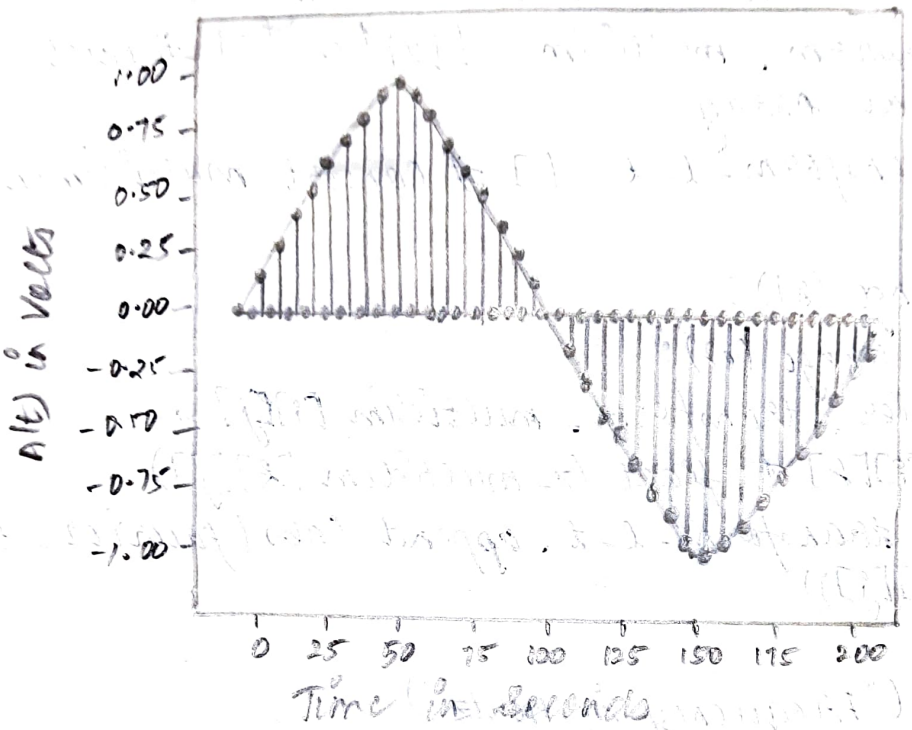




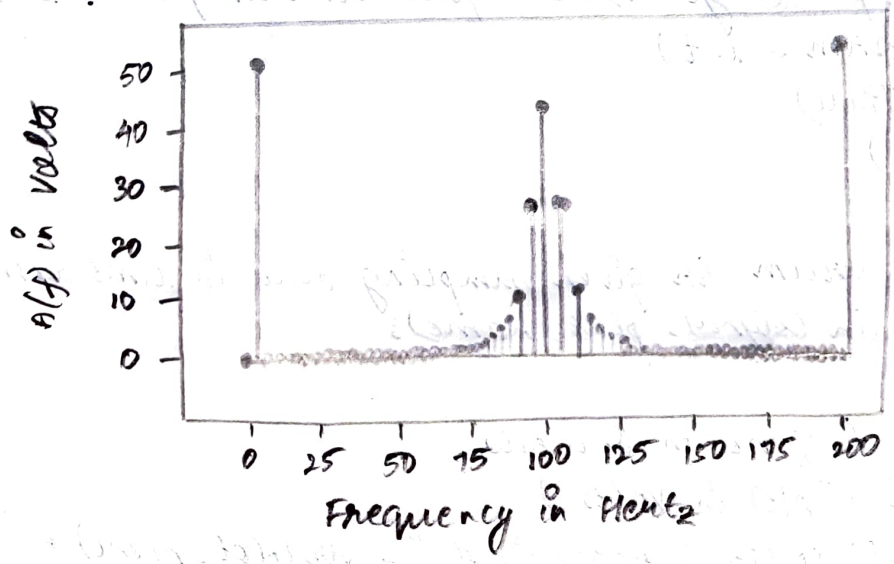
```
    wn_multidim = np.reshape(wn, (r1, c1)) # An N*N W(N) matrix
    r2 = len(x); c2 = 1
    x_multidim = np.reshape(x, (r2, c2)) # An N*1 x(N) matrix

    # Compute x(N) = W(N) * x(N), an N*1 matrix
    fourier_transform_multidim = [[0]*c2]*r1 # Null Multidimensional Array
    fourier_transform_1_t = [] # Convert multidimensional Array to 1D
    for i in range(r1):
        for j in range(c2):
            fourier_transform_multidim[i][j] += wn_multidim[i][k] * float(x_multidim[k][j])
        fourier_transform_1_t.append(abs(fourier_transform_multidim[i][j]))

    plt.xlabel("Frequency in Hertz")
    plt.ylabel("A(f) in Volts")
    plt.title("Spectral Analysis of " + str(plot_name) + " in Frequency Domain")
    plt.stem(np.arange(0, len(fourier_transform_1_t)), fourier_transform_1_t)
    plt.grid(True)
    plt.show()

# sketch the spectrum for given sampling rate in time domain:
def time_domain(signal, plot_name):

    plt.xlabel("Time in Seconds")
    plt.ylabel("A(t) in Volts")
    plt.title("Spectral Analysis of " + str(plot_name) + " in Time Domain")
    plt.stem(np.arange(0, len(signal)), signal)
```

Enter the value of L-fold expander : 2

Spectral Analysis of Up-Sampler in Time Domain



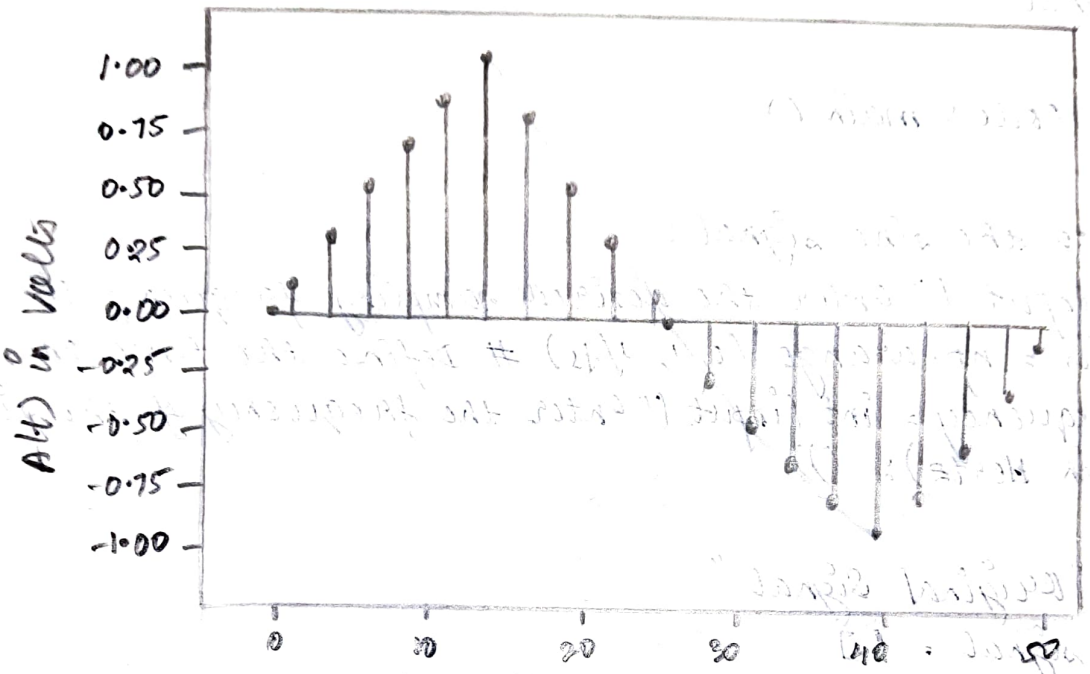
Spectral Analysis of Down-Sampler in Frequency Domain



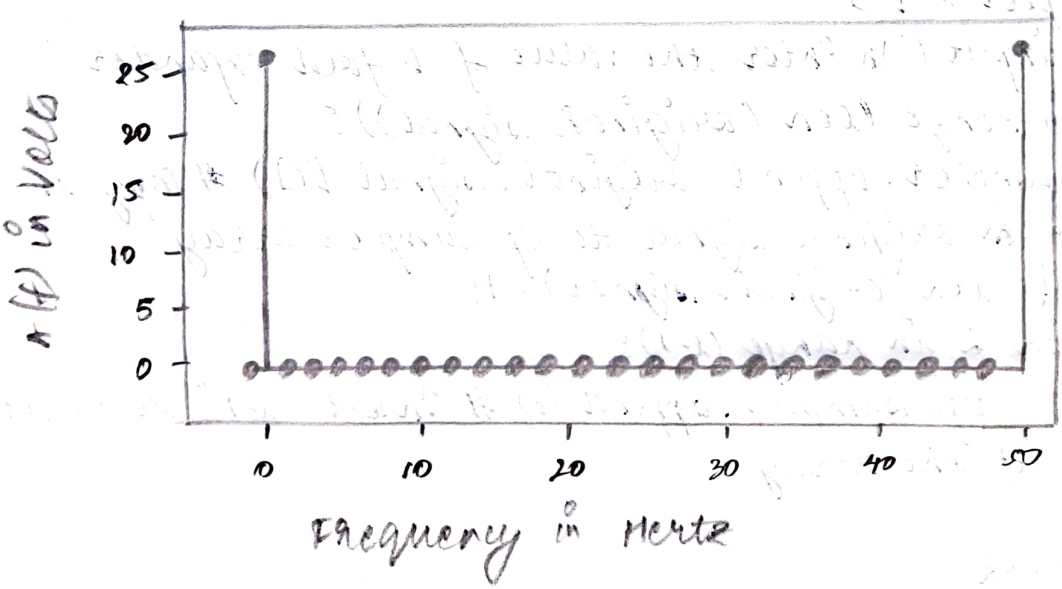
```
    plt.grid()
    plt.show()

    DFT(signal, plot_name) # Frequency domain analysis of the
given signal

# Driver code: main()

# Generate the sine signal:
fs = int(input("Enter the desired sampling frequency (in Hertz): "))
time_axis = np.arange(0, 1, 1/fs) # Define the time axis
sine_frequency = int(input("Enter the frequency of the sine
signal (in Hertz): "))

title = "Original Signal"
original_signal = []
original_signal = np.sin(2 * np.pi * sine_frequency * time_axis)
time_domain(original_signal, title)

title = "up-sampler"
up_sampler = []
l = int(input("In Enter the value of L-fold expander:"))
for i in range(len(original_signal)):
    up_sampler.append(original_signal[i]) # Copy the element
value from original signal to up sampler array
    if i != len(original_signal)-1:
        for k in range(l-1):
            up_sampler.append(0) # Insert "l-1" zeros between the
elements of the array
    else:
        break
time_domain(up_sampler, title)
```

Enter the value of M-fold expander : 2

Spectral Analysis of Down-Sampler in Time Domain



Spectral Analysis of Down-Sampler in Frequency Domain



```
title = "Down-sampler"
down-sampler = []
m = int (input ("\n Enter the value of M-fold expander : ")) # Copy
the element value from original signal to down sampler array with
the increment value set to "m":
for i in range (0, len (original_signal), m):
    down_sampler.append (original_signal[i])
time_domain (down_sampler, title)
```

* **RESULT:**

Generated signals of definite frequency and applied sampling rate conversion using decimation and interpolation. Also, studied the time domain and frequency domain characteristics. All the simulation results were verified successfully.