Lab 20

Name :- Aryan Dilipbhai Langhanoja

Date :- 10-10-2023

Enrollment No :- 92200133030

CO1: To write, test, and debug simple Python programs

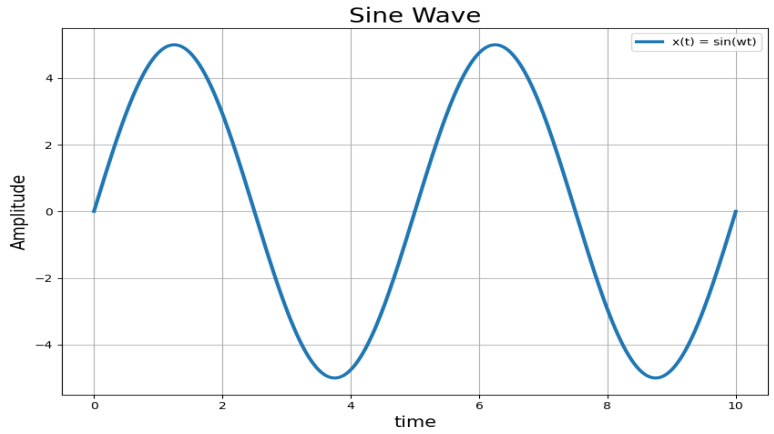
CO2: To implement Python programs with conditional, loops and functions

Task 1:- Generating and Plotting a Sine Wave Signal

Python Code:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal as sg
freq = 200000
amp = 5
t = np.linspace(0, 10, 1000)
sig_sine = amp*np.sin(2*np.pi*freq*t)
plt.figure(figsize=(10, 8))
plt.title('Sine Wave', fontsize=20)
plt.plot(t, sig_sine, linewidth=3, label='x(t) = sin(wt)')
plt.xlabel('time', fontsize=15, )
plt.ylabel('Amplitude', fontsize=15)
plt.legend(loc="upper right")
plt.grid()
plt.show()
```

Output:

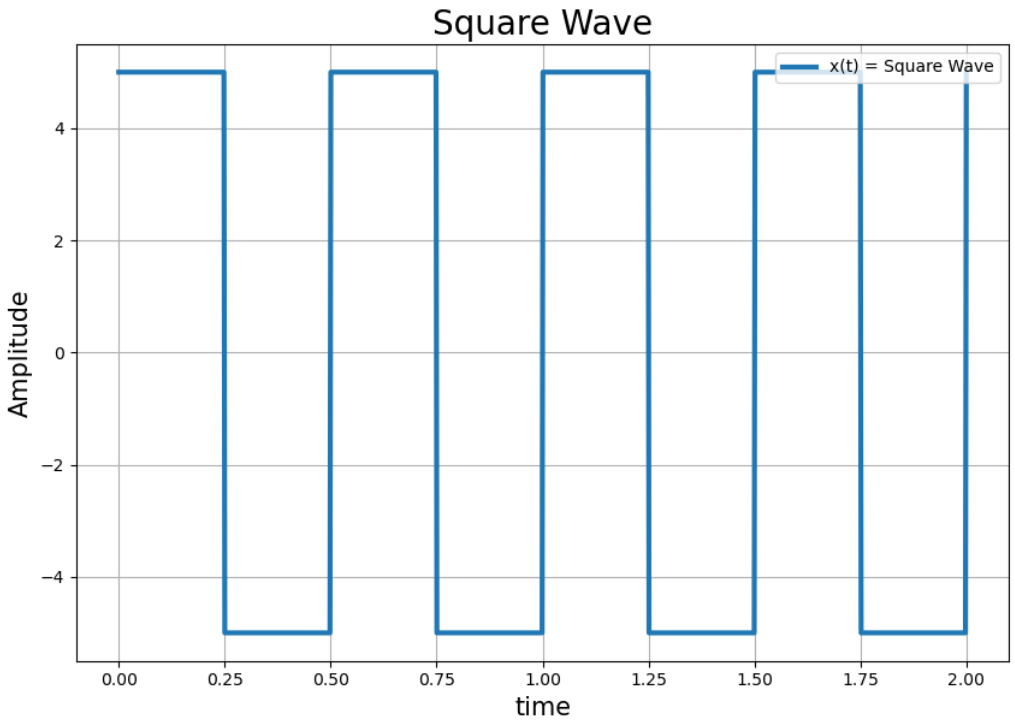


Task 2:- Generating and Plotting a Square Wave Signal with Python.

Python Code:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal as sg
freq = 2
amp = 5
t = np.linspace(0, 2, 1000)
sig_square = amp*sg.square(2*np.pi*freq*t, duty=0.5)
plt.figure(figsize=(10, 8))
plt.title('Square Wave', fontsize=20)
plt.plot(t, sig_square, linewidth=3, label='x(t) = Square Wave')
plt.xlabel('time', fontsize=15, )
plt.ylabel('Amplitude', fontsize=15)
plt.legend(loc="upper right")
plt.grid()
plt.show()
```

Output:

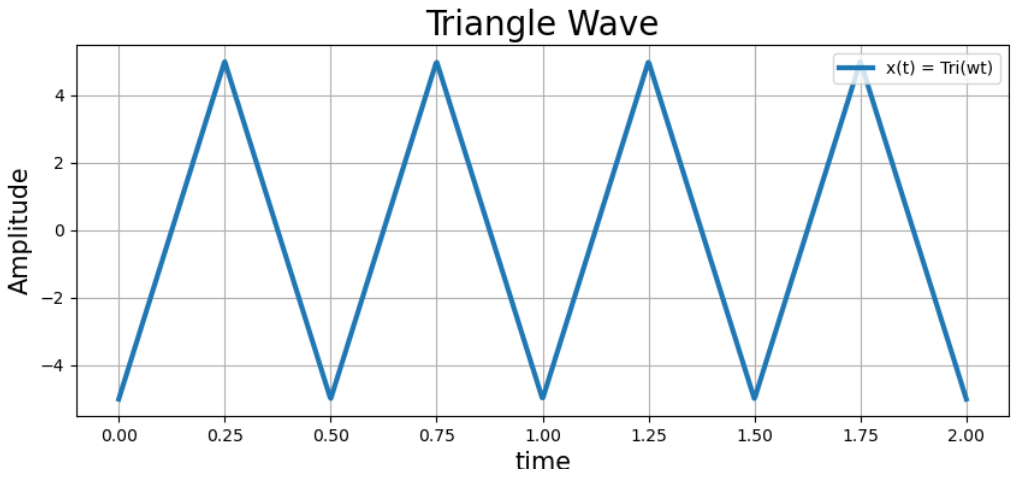


Task 3:- Generating and Plotting a Triangle Waveform in Python with Matplotlib

Python Code:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal as sg
freq = 2
amp = 5
t = np.linspace(0, 2, 1000)
sig_triangle = amp*sg.sawtooth(2*np.pi*freq*t, width=0.5)
plt.figure(figsize=(10, 4))
plt.title('Triangle Wave', fontsize=20)
plt.plot(t, sig_triangle, linewidth=3, label='x(t) = Tri(wt)')
plt.xlabel('time', fontsize=15, )
plt.ylabel('Amplitude', fontsize=15)
plt.legend(loc="upper right")
plt.grid()
plt.show()
```

Output:



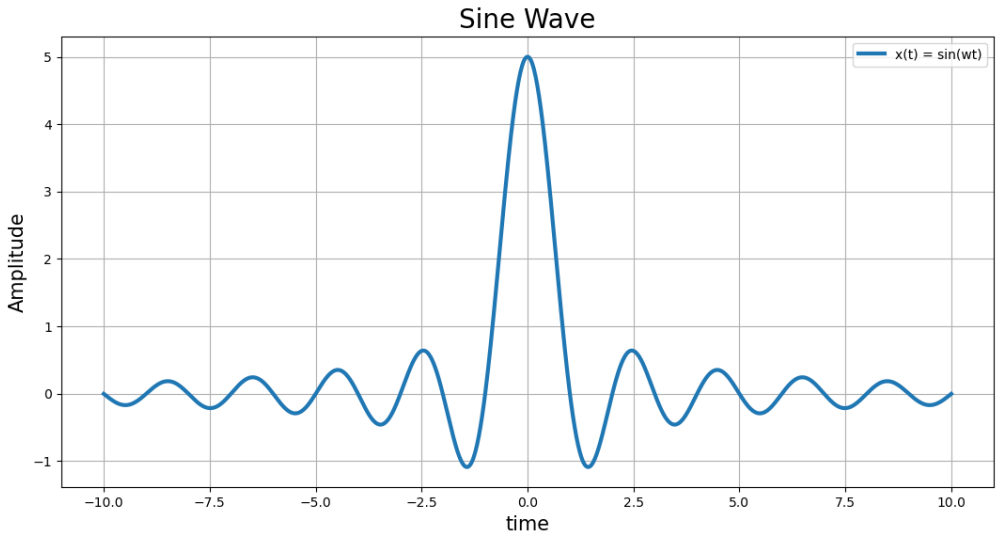
Task 4:- Plot of Sinc Function with Numpy and Matplotlib

Python Code:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal as sg
freq = 200000
amp = 5
t = np.linspace(-10, 10, 1000)
sig_sine = amp*np.sinc(t)
plt.figure(figsize=(10, 8))
plt.title('Sine Wave', fontsize=20)
plt.plot(t, sig_sine, linewidth=3, label='x(t) = sin(wt)')
```

```
plt.xlabel('time', fontsize=15, )
plt.ylabel('Amplitude', fontsize=15)
plt.legend(loc="upper right")
plt.grid()
plt.show()
```

Output:



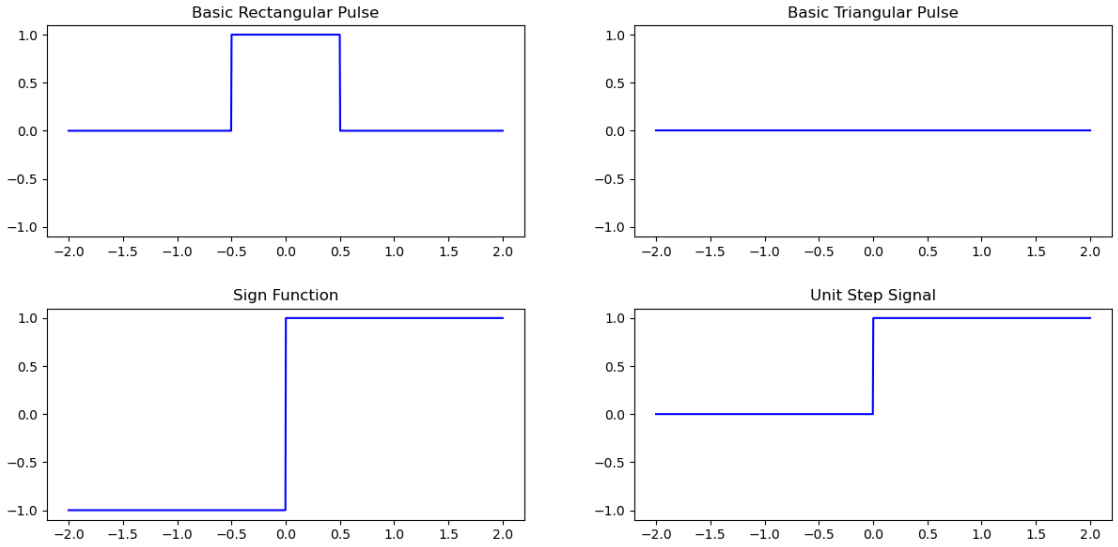
Task 5:- Generating and Plotting a Triangle Waveform in Python with Matplotlib

Python Code:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal as sg
def p(t):
    '''Basic Rectangular Pulse'''
    return 1 * (abs(t) < 0.5)
def pt(t):
    """Basic Triangular Pulse"""
    return (1 - abs(t)) * (abs(t) < t)
def sgn(t):
    """Sign Function"""
    return 1 * (t >= 0) - 1 * (t < 0)
def u(t):
    """Unit Step Signal"""
    return 1 * (t >= 0)
functions = [p, pt, sgn, u]
t = np.linspace(-2, 2, 1000)
plt.figure()
```

```
for i, function in enumerate(functions, start=1):
    plt.subplot(2, 2, i)
    plt.plot(t, function(t), '-b')
    plt.ylim((-1.1, 1.1))
    plt.title(function.__doc__)
plt.tight_layout()
plt.show()
```

Output:

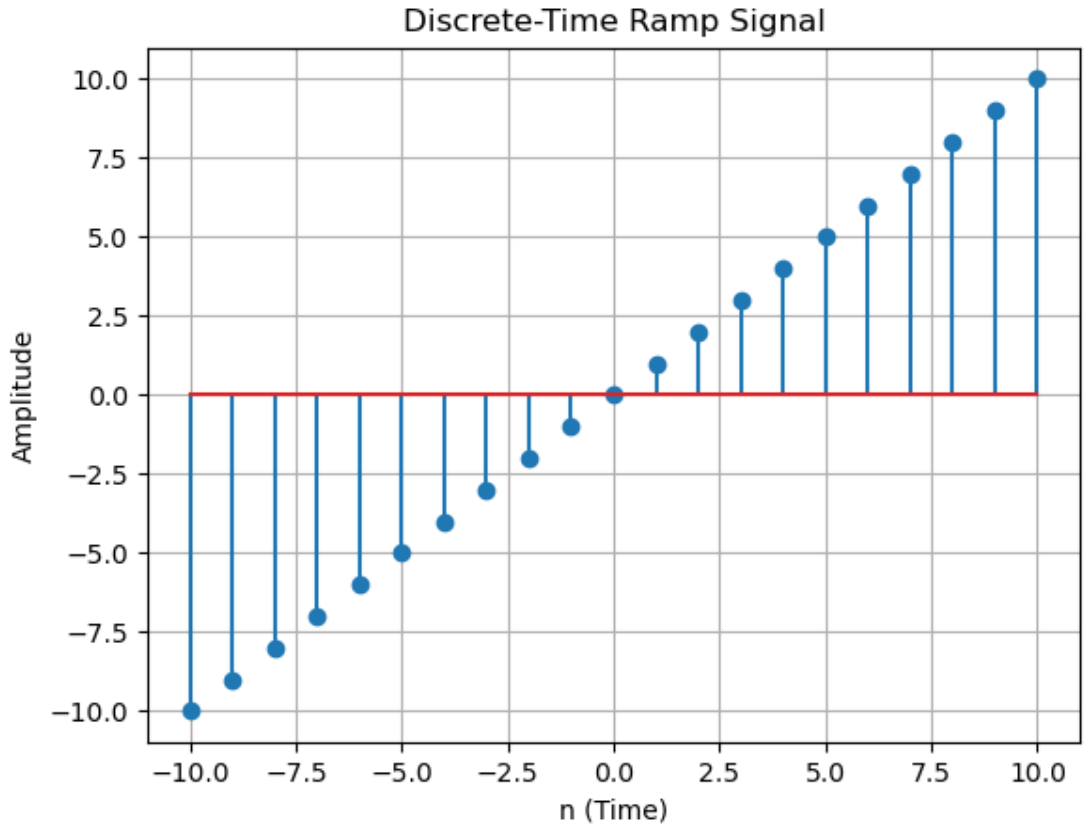


Post – Lab

Task 1:- Discrete-Time Ramp Signal Plot

Python Code:

```
import numpy as np
import matplotlib.pyplot as plt
n = np.arange(-10, 11)  # Values from -10 to 10
ramp_signal = n
plt.stem(n, ramp_signal, use_line_collection=True)
plt.title('Discrete-Time Ramp Signal')
plt.xlabel('n (Time)')
plt.ylabel('Amplitude')
plt.grid(True)
plt.show()
```


Discrete-Time Ramp Signal
Amplitude
n (Time)
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

Output: