EXPERIMENT 2 – GENERATION OF SIGNALS:

CODE:

# -\*- coding: utf-8 -\*-

"""

Created on Tue Jan 30 10:54:30 2024

@author: cb.en.u4ece22151

"""

import numpy as np

import matplotlib.pyplot as plt

def unit\_step(t):

return np.heaviside(t,1)

def unit\_impulse(t):

return np.where(t==0,1,0)

def ramp(t):

return np.where(t>=0,t,0)

def exp\_decay(t):

return np.exp(-t)

def exp\_rising(t):

return np.exp(t)

c=np.linspace(-5, 5,100)

d=np.linspace(-5, 5,20)

plt.subplot(8,2,1)

a=unit\_step(c)

plt.plot(c,a)

plt.xlabel("Time")

plt.ylabel("Amplitude")

plt.title("Continuous unit step")

plt.subplot(8,2,2)

b=unit\_step(d)

plt.stem(d,b)

plt.xlabel("N index")

plt.ylabel("Amplitude")

plt.title("Discrete unit step")

plt.subplot(8,2,3)

q=unit\_impulse(c)

plt.plot(c,q)

plt.stem([0],[1],"--")

plt.xlabel("Time")

plt.ylabel("Amplitude")

plt.title("Continuous Unit impulse")

plt.subplot(8,2,4)

w=unit\_impulse(d)

plt.stem(d,w)

plt.stem([0],[1])

plt.xlabel("N index")

plt.ylabel("Amplitude")

plt.title("Discrete Unit impulse")

plt.subplot(8,2,5)

r=ramp(c)

plt.plot(c,r)

plt.xlabel("Time")

plt.ylabel("Amplitude")

plt.title("Continuous Ramp")

plt.subplot(8,2,6)

t=ramp(d)

plt.stem(d,t)

plt.xlabel("N index")

plt.ylabel("Amplitude")

plt.title("Discrete Ramp")

plt.subplot(8,2,7)

y=exp\_decay(c)

plt.plot(c,y)

plt.xlabel("Time")

plt.ylabel("Amplitude")

plt.title("Continuous Exponential Decay")

plt.subplot(8,2,8)

u=exp\_decay(d)

plt.stem(d,u)

plt.xlabel("N index")

plt.ylabel("Amplitude")

plt.title("Discrete Exponential Decay")

plt.subplot(8,2,9)

i=exp\_rising(c)

plt.plot(c,i)

plt.xlabel("Time")

plt.ylabel("Amplitude")

plt.title("Continuous Exponential Rising")

plt.subplot(8,2,10)

o=exp\_rising(d)

plt.stem(d,o)

plt.xlabel("N index")

plt.ylabel("Amplitude")

plt.title("Discrete Exponential Decay")

c1=np.linspace(-3, 3,100)

z=np.arange(np.pi,5\*np.pi,0.5)

plt.subplot(8,2,11)

l=np.sin(z)\*2

plt.plot(z/(60\*np.pi),l)

plt.xlabel("Time")

plt.ylabel("Amplitude")

plt.title("Continuous Sinusoidal")

plt.subplot(8,2,12)

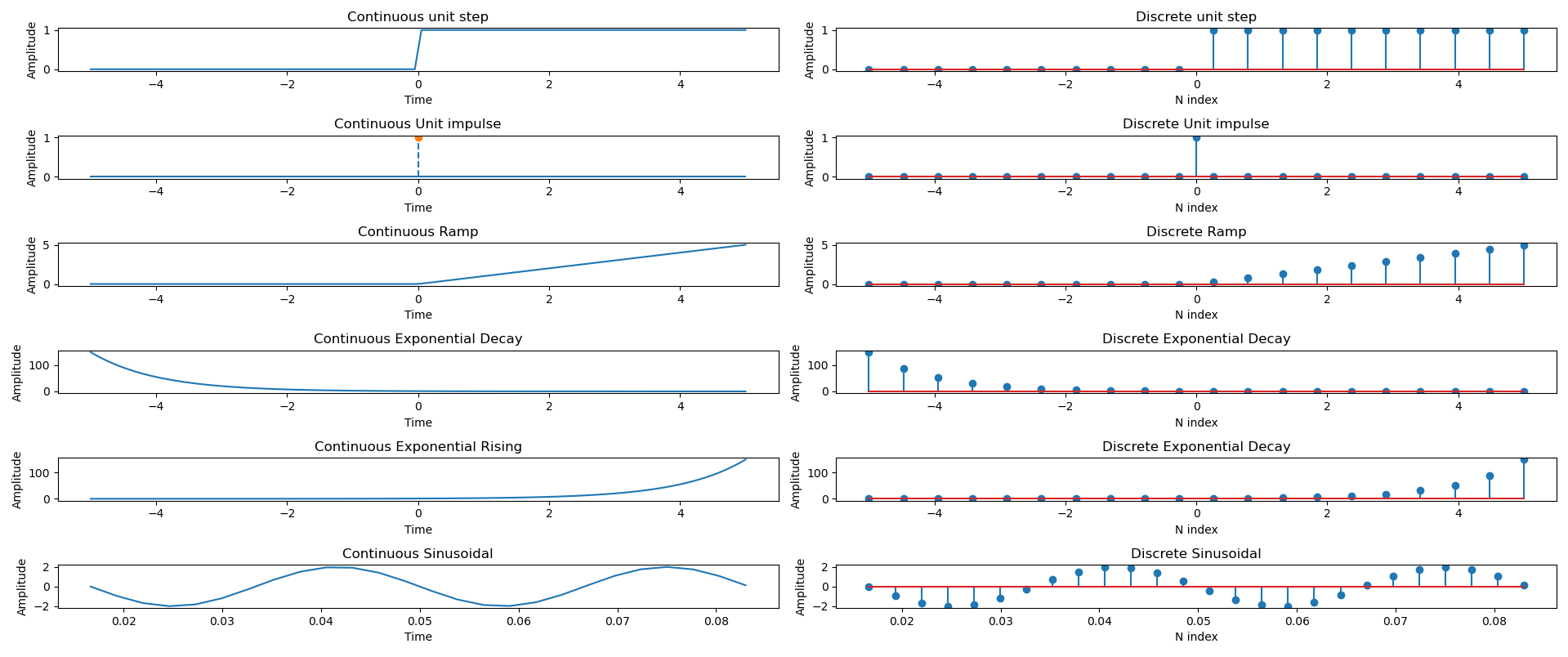
k=np.sin(z)\*2

plt.stem(z/(60\*np.pi),k)

plt.xlabel("N index")

plt.ylabel("Amplitude")

plt.title("Discrete Sinusoidal")

OUTPUT: