EXPERIMENT 8 – LINEAR FILTERING OF LONG SEQUENCES:

CODE:

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

"""

Created on Sat Apr 20 11:25:11 2024

@author: sanja

"""

import numpy as np

import matplotlib.pyplot as plt

def plot\_sequence(x,t,title):

plt.stem(t,x)

plt.title(title)

plt.xlabel('n')

plt.ylabel('Amplitude')

plt.grid(True)

plt.show()

def plot\_sequence1(x,title):

plt.stem(x)

plt.title(title)

plt.xlabel('n')

plt.ylabel('Amplitude')

plt.grid(True)

plt.show()

def circ\_conv(x1,x2):

N=len(x1)

y=np.zeros\_like(x1)

for i in range(N):

for k in range(N):

if(i-k)>=0:

y[i]+=x1[k]\*x2[i-k]

else:

y[i]+=x1[k]\*x2[N-k+i]

return y[:2\*N-1]

def overlap\_add\_convolve(h, x, L):

Lenh = len(h)

Lenx = len(x)

Leny = Lenh+Lenx - 1

x\_pad = np.pad(x,(0, Lenh-1))

h\_pad = np.pad(h,(0, Lenx-1))

y = np.zeros(Leny)

for i in range(0, Leny, L):

x\_b = x\_pad[i:i + L]

conv\_b = np.convolve(x\_b, h\_pad)

y[i:i + len(conv\_b)] += conv\_b[:len(y[i:i + len(conv\_b)])]

return y

plt.figure(1)

plt.subplot(1,2,1)

tx=np.arange(0,9)

x=np.array([1,2,-3,-4,2,3,-1,-5,4])

plot\_sequence(x,tx,"Input sequence")

plt.subplot(1,2,2)

th=np.arange(0,3)

h=np.array([4,2,3])

M=len(h)

plot\_sequence(h,th,"Impulse Response")

L=3

plt.figure(2)

hpad=np.pad(h,(0,M-1))

plot\_sequence1(hpad,"Impulse response after padding")

plt.figure(3)

plt.subplot(1,3,1)

x1=x[0:L]

plot\_sequence1(np.pad(x1,(0,M-1)),"1st sequence with zero padding")

plt.subplot(1,3,2)

x2=x[L:2\*L]

plot\_sequence1(np.pad(x2,(0,M-1)),"2nd sequence with zero padding")

plt.subplot(1,3,3)

x3=x[2\*L:3\*L]

plot\_sequence1(np.pad(x3,(0,M-1)),"3rd sequence with zero padding")

plt.figure(4)

plt.subplot(1,3,1)

x1pad=np.pad(x1,(0,M-1))

y1=circ\_conv(x1pad,hpad)

plot\_sequence1(y1,"1st sequence result after circular convolution")

plt.subplot(1,3,2)

x2pad=np.pad(x2,(0,M-1))

y2=circ\_conv(x2pad,hpad)

plot\_sequence1(y2,"2nd sequence result after circular convolution")

plt.subplot(1,3,3)

x3pad=np.pad(x3,(0,M-1))

y3=circ\_conv(x3pad,hpad)

plot\_sequence1(y3,"3rd sequence result after circular convolution")

plt.figure(5)

plt.subplot(1,2,1)

output=overlap\_add\_convolve(h,x,L)

plot\_sequence1(output,"Final output")

print(output)

plt.subplot(1,2,2)

op=np.convolve(x,h)

plot\_sequence1(op,"built-in output")

OUTPUT:











