Experiment No: 5 Date: 23/09/2024

LINEAR CONVOLUTION FROM CIRCULAR CONVOLUTION

Aim

To get Linear Convolution from Circular Convolution.

Theory

Linear convolution is a fundamental operation in digital signal processing (DSP) used to analyze the relationship between input signals and systems. It combines two signals to produce a third signal, which represents the output of a system when an input signal is applied. This process allows us to analyze how an input signal is transformed by the system's characteristics, often described by its impulse response. The operation involves sliding the impulse response across the input signal and calculating the output at each position, considering the overlapping portions. One of the key features of linear convolution is its order independence; the result remains unchanged regardless of the order in which the signals are combined. Linear convolution finds wide applications, particularly in filtering, where it is used to modify signals by enhancing or reducing certain characteristics. It is also crucial in system analysis and various image processing tasks, such as blurring and edge detection. While conceptually straightforward, linear convolution can be computationally intensive, often leading to the use of efficient algorithms like the Fast Fourier Transform (FFT) to improve processing speed.

Circular convolution is a process used in signal processing to combine two periodic signals. Unlike linear convolution, where signals extend infinitely, circular convolution assumes that the signals wrap around, creating a continuous loop. This means that when the end of one signal meets the beginning of the other, they interact as if they were connected. This approach is particularly useful for analyzing signals in systems where periodicity is important, such as in digital signal processing and communications. Circular convolution can be efficiently computed using the Fast Fourier Transform (FFT), allowing for rapid processing of data. In practical applications, circular convolution helps in tasks like filtering, where the output signal is influenced by the entire input signal, taking into account the periodic nature of the data.

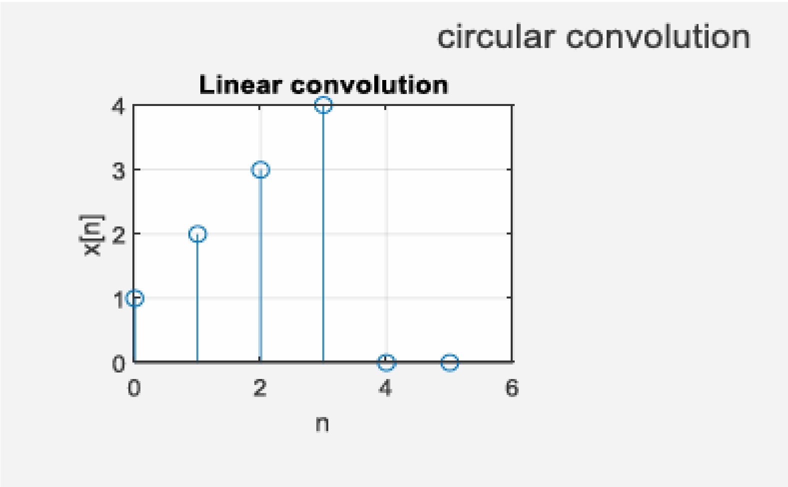
OBSERVATION

INPUT=

x = [1 2 3 1]

h= [1 1 1]

OUTPUT=



Program

```
clc;
clf;
close all;
clear all;
x=input ('enter x values=');
h=input ('enter h values=');
N=length(x);
m=length(h);
L=N+m-1
y_n=zeros(1,L);
x=[x zeros(1,(L-N))];
h=[h zeros(1,(L-m))];
for (i=1:L)
for (k=1:L)
y_n(i)=y_n(i)+x(k)*h(mod(i-k,L)+1);
end
end
disp(y_n);
y_conv=cconv(x,h,L);
disp(y_conv);
tile=tiledlayout(2,2);
title(tile,'circular convolution');
nexttile;
stem(0:L-1,x);
xlabel('n');
ylabel('x[n]');
title('Linear convolution');
grid on;
```

Result

The Linear Convolution is obtained from Circular Convolution and the graph is plotted.