Experiment No: 04 Date: 22-08-2024

Circular Convolution

Aim

To find circular convolution

a) Using FFT and IFFT.
b) Using Concentric Circle Method.
c) Using Matrix Method.

Theory

Circular convolution is a mathematical operation that is like linear convolution but is performed in a periodic or circular manner. This is particularly useful in discrete-time signal processing where signals are often represented as periodic sequences.

Mathematical Definition:

Given two periodic sequences x[n] and h[n], their circular convolution is defined as:

$$y[n] = (x[n] \circledast h[n]) = \sum_{k=0}^{N-1} x[k]h[(n-k) \bmod N]$$

Applications:

• Discrete-Time Filtering: Circular convolution is used for filtering discrete-time signals.

• Digital Signal Processing: It's a fundamental operation in many digital signal processing algorithms.
• Cyclic Convolution: In certain applications, such as cyclic prefix OFDM, circular convolution is used to simplify the implementation of linear convolution.

Program

a)Using FFT and IFFT

```
clc;
clear all;
close all;
x=input("Enter the elements in x[n]:");
x_ind=input("Enter the index of x[n]:");
h=input("Enter the elements in h[n]:");
h_ind=input("Enter the index of h[n]:");
```

```
figure;
subplot(3,1,1);
stem(x_ind,x);
title("x[n]");
xlabel("time ");
ylabel("amplitude");
grid;
subplot(3,1,2);
stem(h_ind,h);
title("h[n]");
xlabel("time");
ylabel("amplitude");
grid;
len_x=length(x);
len_h=length(h);
N=max(len_x,len_h);
new_x=[x zeros(1,N-len_x)];
new_h=[h zeros(1,N-len_h)];
x1=fft(new_x);
h1=fft(new_h);
y1=x1.*h1;
y=ifft(y1);
ny=0:N-1;
disp(y);
subplot(3,1,3);
stem(ny,y);
title("Circular convolution output y[n]");
xlabel("time ");
ylabel("amplitude");
grid;
```

b)Using concentric circle method

```
clc;
clear all;
close all;
x=input("Enter the elements in x[n]:");
x_ind=input("Enter the index of x[n]:");
h=input("Enter the elements in h[n]:");
h_ind=input("Enter the index of h[n]:");
x1=x;
x=x(:,end:-1:1);
for i=1:length(x)
    x=[x(end) x(1:end-1)];
    y(i)=sum(x.*h);
end
disp(y);
figure;
subplot(3,1,1);
stem(x_ind,x1);
title("x[n]");
xlabel("time ");
ylabel("amplitude");
grid;
subplot(3,1,2);
stem(h_ind,h);
title("h[n]");
xlabel("time ");
ylabel("amplitude");
grid;
subplot(3,1,3);
```

```
Ny=0:3;
stem(Ny,y);
title("circular convolution output y[n]");
xlabel("time ");
ylabel("amplitude");
grid;
```

c)Using matrix method

```
clc;
clear all;
close all;
x=input("Enter the elements in x[n]:");
x_ind=input("Enter the index of x[n]:");
h=input("Enter the elements in h[n]:");
h_ind=input("Enter the index of h[n]:");
hr=[];
h1=h;
h=h(:,end:-1:1);
for i=1:length(h);
    h=[h(end) h(1:end-1)];
    hr=[hr;h];
end
y=hr*x';
disp(y);
figure;
subplot(3,1,1);
stem(x_ind,x);
title("x[n]");
xlabel("time ");
```

```
ylabel("amplitude");
grid;
subplot(3,1,2);
stem(h_ind,h);
title("h[n]");
xlabel("time ");
ylabel("amplitude");
grid;
subplot(3,1,3);
Ny=0:3;
stem(Ny,y);
title("circular convolution output y[n]");
xlabel("time ");
ylabel("amplitude");
grid;
```

Result

Performed Circular Convolution using a) FFT and IFFT; b) Concentric Circle method; c) Matrix method and verified result.

Observation

a)Using FFT and IFFT

INPUT:

Enter the elements in x[n]:

[2 1 2 1]

Enter the index of x[n]:

0:3

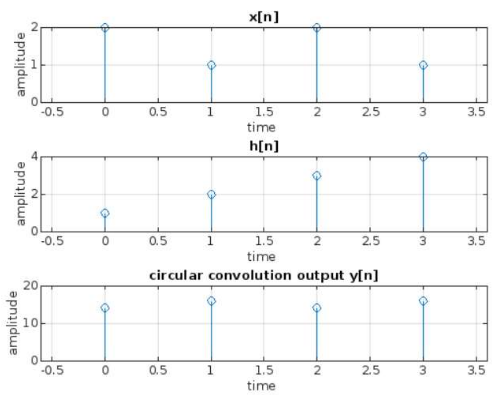
Enter the elements in h[n]:

[1 2 3 4]

Enter the index of h[n]:

0:3

OUTPUT:



b)Using concentric circle method

INPUT:

Enter the elements in x[n]:

[2 1 2 1]

Enter the index of x[n]:

0:3

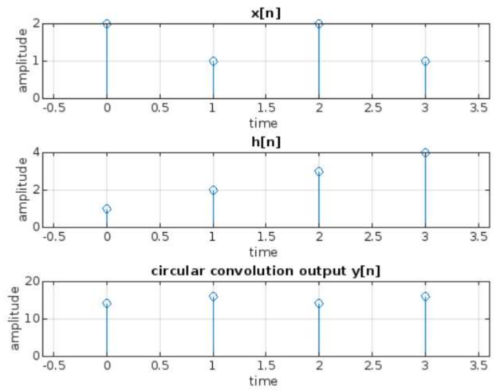
Enter the elements in h[n]:

[1 2 3 4]

Enter the index of h[n]:

0:3

OUTPUT:



c)Using matrix method

INPUT:

Enter the elements in x[n]:

[2 1 2 1]

Enter the index of x[n]:

0:3

Enter the elements in h[n]:

[1 2 3 4]

Enter the index of h[n]:

0:3

OUTPUT: