LAB 5: Circular Convolution

Theory:

Circular Convolution:

Circular convolution is an operation that combines two sequences using circular shifts and element-wise multiplication. It is often used in signal processing and image processing applications. Circular convolution differs from linear convolution in that it assumes the sequences to be periodic, wrapping around at the endpoints.

To perform circular convolution, you need two input sequences of equal length, let's say sequence A and sequence B, each containing N elements. The circular convolution of A and B, denoted as C, is computed as follows:

1. Extend both sequences A and B to a length of 2N by appending N-1 zeros to each sequence.
2. Take the discrete Fourier transform (DFT) of A and B, resulting in A_fft and B_fft, respectively.
3. Compute the element-wise multiplication of A_fft and B_fft, denoted as C_fft.
4. Take the inverse discrete Fourier transform (IDFT) of C_fft to obtain the circular convolution C.

The resulting sequence C will also have N elements, representing the circular convolution of A and B.

Mathematically, circular convolution can be expressed as:

C = IDFT(DFT(A) * DFT(B))

Zero Padding:

Zero padding in convolution refers to the technique of adding zeros to the input signal or image before performing the convolution operation. It involves extending the boundaries of the input with zeros to create a larger signal or image, which allows for a larger output size and can help preserve important spatial or spectral information during the convolution process.

The main purposes of zero padding in convolution are:

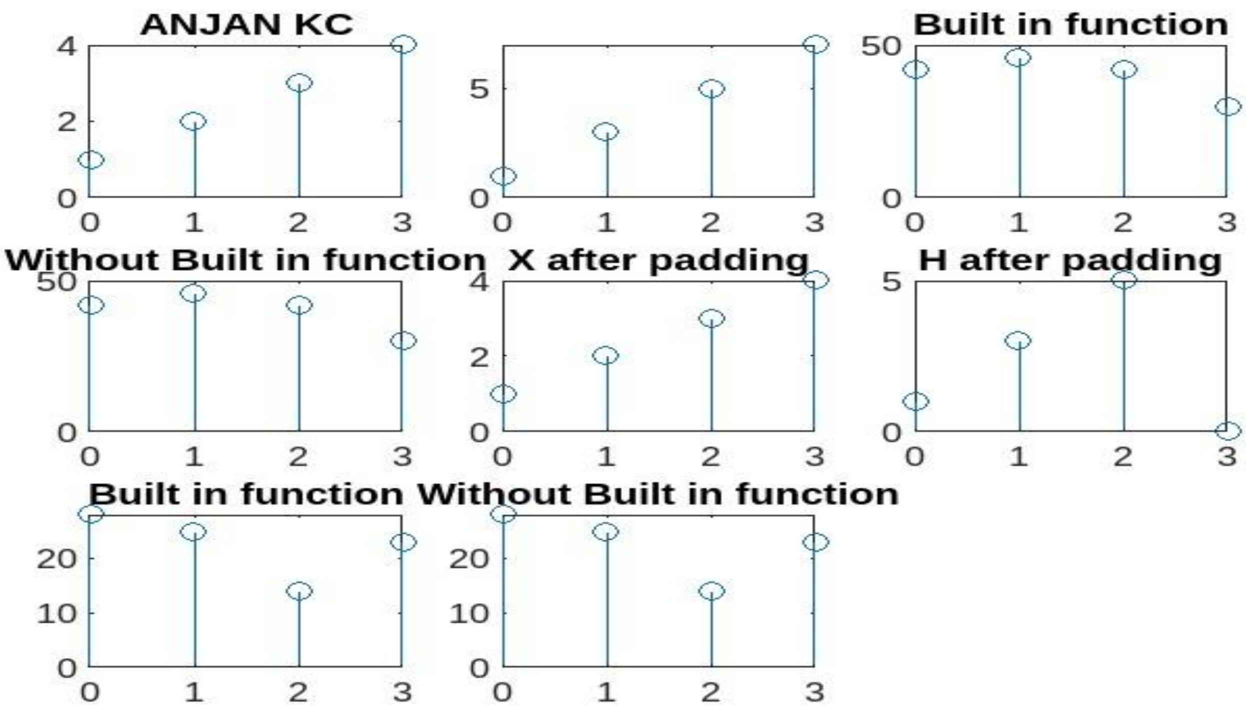
1. Preserving spatial dimensions: When applying convolution to an input signal or image, the output size is typically smaller than the input size.
2. Avoiding border artifacts: Without padding, the convolution operation can cause border artifacts, such as the "border effect" or "ringing effect," where the output values near the edges are distorted or influenced by the edge values of the input.
3. Preserving frequency information: In frequency domain processing, such as using the Fast Fourier Transform (FFT) for convolution, zero padding is used to increase the resolution of the frequency spectrum. By padding the input signal or image with zeros, the resulting spectrum has more samples and provides finer frequency details.

Code:

```
x=[1 2 3 4];
h=[1 3 5];
l1=length(x);
l2=length(h);
N=max(l1,l2);
t=0:1:N-1;
subplot(3,3,1);
stem(t,x);
title('ANJAN KC');
h=[1 3 5 7];
subplot(3,3,2);
stem(t,h);
l1=length(x);
l2=length(h);
N=max(l1,l2);
t=0:1:N-1;
y=cconv(x,h,N);%Using built in function
subplot(3,3,3);
stem(t,y);
title('Convolution using Built in function');
%Without using Builtin Function
k=zeros(1,N);
for n=1:N
    for m=1:N
        j= mod(n-m,N);
        j=j+1;
        k(n)=k(n)+x(m)*h(j);
    end
end
subplot(3,3,4);
stem(t,k);
title('Convolution without Built in function');
%ZERO padding
x=[1 2 3 4];
h=[1 3 5];
l1=length(x);
l2=length(h);
N=max(l1,l2);
t=0:1:N-1;
x=[x zeros(1,N-l1)];
h=[h zeros(1,N-l2)];
subplot(3,3,5);
stem(t,x)
title('Signal x after padding');
subplot(3,3,6);
stem(t,h);
title('Signal h after padding');
y=cconv(x,h,N);%Using built in function
subplot(3,3,7);
stem(t,y);
title('Convolution using Built in function');
%Without using Builtin Function
k=zeros(1,N);
```

```
for n=1:N
    for m=1:N
        j= mod(n-m,N);
        j=j+1;
        k(n)=k(n)+x(m)*h(j);
    end
end
subplot(3,3,8);
stem(t,k);
title('Convolution without Built in function');
```

Output:



Conclusion:

We became familiar with Circular Convolution. We implement input signals and calculate convolution by using graphical method. Also we use zero padding to make the size if input signal equal.