**PRACTICAL 4 : Convolution**

|  |  |
| --- | --- |
| **U19EC046 - Harsh Suthar** | **Date: 13-08-2021** |

**AIM**

To find the discrete Convolution of given two sequences.

**THEORY**

Convolution, one of the most important concepts in electrical engineering, can be used to determine the output a system produces for a given input signal. It can be shown that a linear time invariant system is completely characterized by its impulse response. The sifting property of the discrete time impulse function tells us that the input signal to a system can be represented as a sum of scaled and shifted unit impulses. Thus, by linearity, it would seem reasonable to compute of the output signal as the sum of scaled and shifted unit impulse responses. That is exactly what the operation of convolution accomplishes. Hence, convolution can be used to determine a linear time invariant system's output from knowledge of the input and the impulse response.

Discrete time convolution is an operation on two discrete time signals defined by the integral

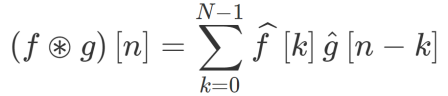
IMG_256

for all signals f,g defined on Z. It is important to note that the operation of convolution is commutative, meaning that

IMG_257

#### Circular Convolution

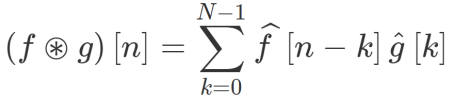
Discrete time circular convolution is an operation on two finite length or periodic discrete time signals defined by the sum



for all signals f,g defined on Z[0,N-1] where f^,g^ are periodic extensions of f and g. It is important to note that the operation of circular convolution is commutative, meaning that



for all signals f,g defined on Z[0,N-1]. Thus, the circular convolution operation could have been just as easily stated using the equivalent definition



**ALGORITHM**

1. Define two sequences x and h to be convoluted.
2. Define N as length of convoluted signal which is equal to length(x) + length(h) -1.
3. Create an empty array y of length N, which will be filled with result of convolution.
4. Create two for loops, the outer for loop will fill the convolution result into y from 1 to N, inner loop will evaluate convolution for particular n using summation, recreating the formula for convolution.

**CODE**

1. Convolution with and without inbuilt function

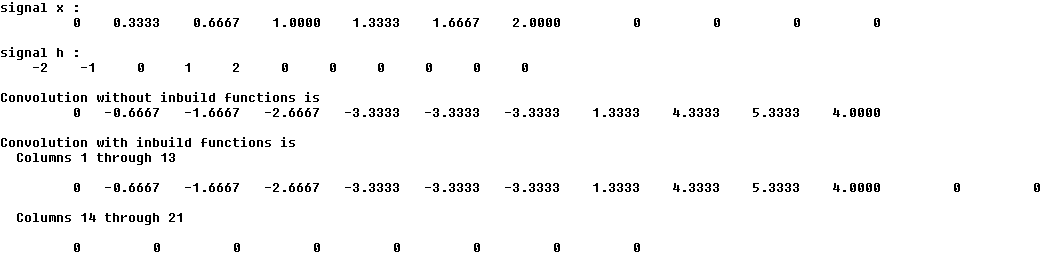
|  |
| --- |
| **clc**  **clear all;**  **x = 0:6;**  **h = -2:2;**  **x = x./3;**  **n1=length(x);**  **n2=length(h);**  **N=n1+n2-1;**  **x=[x, zeros(1, N-n1)];**  **h=[h, zeros(1, N-n2)];**  **disp('signal x :')**  **disp(x)**  **disp('signal h :')**  **disp(h)**  **y= zeros(1, N);**  **for n=1:N**  **for k= 1:n**  **y(n) = y(n)+x(k)\*h(n-k+1);**  **end**  **end**  **disp('Convolution without inbuild functions is' );**  **disp(y);**  **y\_inBuilt= zeros(1, N);**  **y\_inBuilt= conv(x, h);**  **disp('Convolution with inbuild functions is' );**  **disp(y\_inBuilt)** |

1. Circular convolution with and without inbuilt function

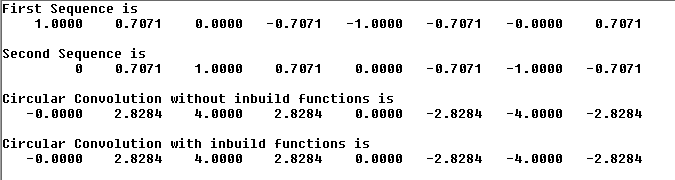
|  |
| --- |
| **clc**  **n=0:7;**  **N=8;**  **x = cos (2\*pi\*n./N);**  **h = sin(2\*pi\*n./N);**  **disp( 'First Sequence is' );**  **disp(x);**  **disp( 'Second Sequence is');**  **disp(h);**  **y=zeros(1, N);**  **for n=1:N**  **for m=1:N**  **z=mod( n-m, N);**  **y(n)=y(n)+x(m)\*h(z+1) ;**  **end**  **end**  **disp('Circular Convolution without inbuild functions is' );**  **disp(y);**  **z=cconv(x, h, N);**  **disp('Circular Convolution with inbuild functions is');**  **disp(z)** |

**OUTPUT**

1. Convolution



1. Circular convolution



**CONCLUSION**

Hence, during this practical we have found out Convolution and circular convolution of two signal using Matlab with and without using inbuilt functions provided by Matlab