7. Convolution: Linear & Circular

AIM

To find linear & circular convolution of two sequences in DSP kit (Using VSK-6748 KIT) and using MATLAB

Tools Used

MATLAB, Code composer Studio

In mathematical terms, given two finite, discrete-time signals x[n]$x[n]$ and h[n]$h[n]$, both of length NN, and their DFTs

Linear Convolution Definition

$$x[n] * h[n] = \sum_{m=-\infty}^{\infty} x[m]h[n-m], \quad n \in \mathbb{Z}.$$

Circular Convolution

Here, $\circledast$ symbol denotes the circular convolution. It is defined as

$$x[n] \circledast h[n] = \sum_{m=0}^{N-1} x[m]h[(n-m)\%N],$$

where % denotes the modulo operation,

Linear convolution - Program

C Program:

```
#include<fastmath67x.h>
void Vi_DSKC6747_init( );
void main()
{
int *Xn,*Hn,*Yout; // Input ,Output Sequence
int *xnlen,*hnlen; // Sequence Length
int i,k,n,l,m;
Xn=(int *)0x80000100;
Hn=(int *)0x80000200;
xnlen = (int *)0x80000300;
hnlen = (int *)0x80000304;
Yout=(int *)0x80000400;
l = *xnlen;
m = *hnlen;
Vi_DSKC6747Rev1_init( ); // Board Initialization
for(i=0;i<50;i++) // Memory Clear
{
Yout[i]=0;
Xn[l+i]=0;
Hn[m+i]=0;
xnlen[2+i]=0;
}
for(n=0;n<(l+m-1);n++)
{
for(k=0;k<=n;k++)
{
Yout[n]=Yout[n]+(Xn[k]*Hn[n-k]);
}
}
}
```

Input & Output:
Sequence Memory Address Values
Xn - First input 0x80000100 1,2,1,1
Hn - Second input 0x80000200 1,1,1
xnlen – 1st Seq length 0x80000300 4
hnlen – 2nd Seq length 0x80000304 3
Yout – Output Seq 0x80000400 1,3,4,4,2,1

Circular convolution – Program

```
#include<fastmath67x.h>

void main()

{

int Xn[4]= {1,2,3,4};      // x(n) input

int Hn[4] = {1,1,2,2};   // h(n) input

int xn = 4;              // length of x(n)

int hn= 4;                                        // length of h(n)

int *Yn;                 // output array

int i,n,m,l;

Yn=(int *)0x80010000;   // o/p starting address

for(i=0;i<xn+hn;i++)    // memory clear

        Yn[i]=0;

for(n=0;n<xn;n++)

        {

        for(m=0;m<hn;m++)

                {
```

```
        l=n-m;
            if(l<0)
            {
            l=l+xn;
            }
        Yn[n]=Yn[n]+(Xn[m]*Hn[l]);    // circular convolution
        }
    }
}
```

MATLAB :

```
x= input('ENTER THE FIRST SEQUENCE');
subplot(3,1,1);
stem(x);
title('FIRST SEQUENCE');
ylabel('x(n)');
xlabel('n');
h= input('ENTER THE SECOND SEQUENCE');
subplot(3,1,2);
stem(h);
title('SECOND SEQUENCE');
ylabel('h(n)');
xlabel('n');
N1=length(x);
N2=length(h);
N= max(N1,N2);
x=[x zeros(1,N-N1)];
h=[h zeros(1,N-N2)];
for n=0:N-1
y(n+1)=0;
for i=0:N-1
j=mod(n-i,N);
y(n+1)=y(n+1)+x(i+1)*h(j+1);
end
end
disp('OUTPUT SEQUENCE IS');
disp(y);
subplot(3,1,3);
stem(y);
title('OUTPUT SEQUENCE')
ylabel('y(n)');
xlabel('n');
```