DIGITAL SIGNAL PROCESSING LAB (IV-I sem)

INDEX

DSP PROGRAMS IN C & MATLAB

1. Linear Convolution

AIM:

To verify Linear Convolution.

EQUIPMENTS:

Operating System	– Windows XP
Constructor	- Simulator
Software	- CCStudio 3 & MATLAB 7.5

THEORY:

Convolution is a formal mathematical operation, just as multiplication, addition, and integration. Addition takes two *numbers* and produces a third *numbe*r, while convolution takes two *signals* and produces a third *signa*l. Convolution is used in the mathematics of many fields, such as probability and statistics. In linear systems, convolution is used to describe the relationship between three signals of interest: the input signal, the impulse response, and the output signal.

$$y(n) = \sum_{k=0}^{N-1} x_1(k)x_2(n-k) \quad 0 < n < N-1 \qquad (1)$$

In this equation, x1(k), x2(n-k) and y(n) represent the input to and output from the system at time n. Here we could see that one of the input is shifted in time by a value every time it is multiplied with the other input signal. Linear Convolution is quite often used as a method of implementing filters of various types.

PROGRAM:

// Linear convolution program in c language using CCStudio

```
#include<stdio.h>
int x[15],h[15],y[15];
main()
{
int i,j,m,n;

printf("\n enter value for m");
scanf("%d",&m);
printf("\n enter value for n");
scanf("%d",&n);

printf("Enter values for i/p x(n):\n");
for(i=0;i<m;i++)
scanf("%d",&x[i]);
printf("Enter Values for i/p h(n) \n");
for(i=0;i<n; i++)
scanf("%d",&h[i]);
```

```
// padding of zeors
for(i=m;i<=m+n-1;i++)
x[i]=0;
for(i=n;i<=m+n-1;i++)
h[i]=0;
/*  convolution operation */
for(i=0;i<m+n-1;i++)
{
y[i]=0;
for(j=0;j<=i;j++)
{
y[i]=y[i]+(x[j]*h[i-j]);
}
}
//displaying the o/p
for(i=0;i<m+n-1;i++)
printf("\n The Value of output y[%d]=%d",i,y[i]);
}
```

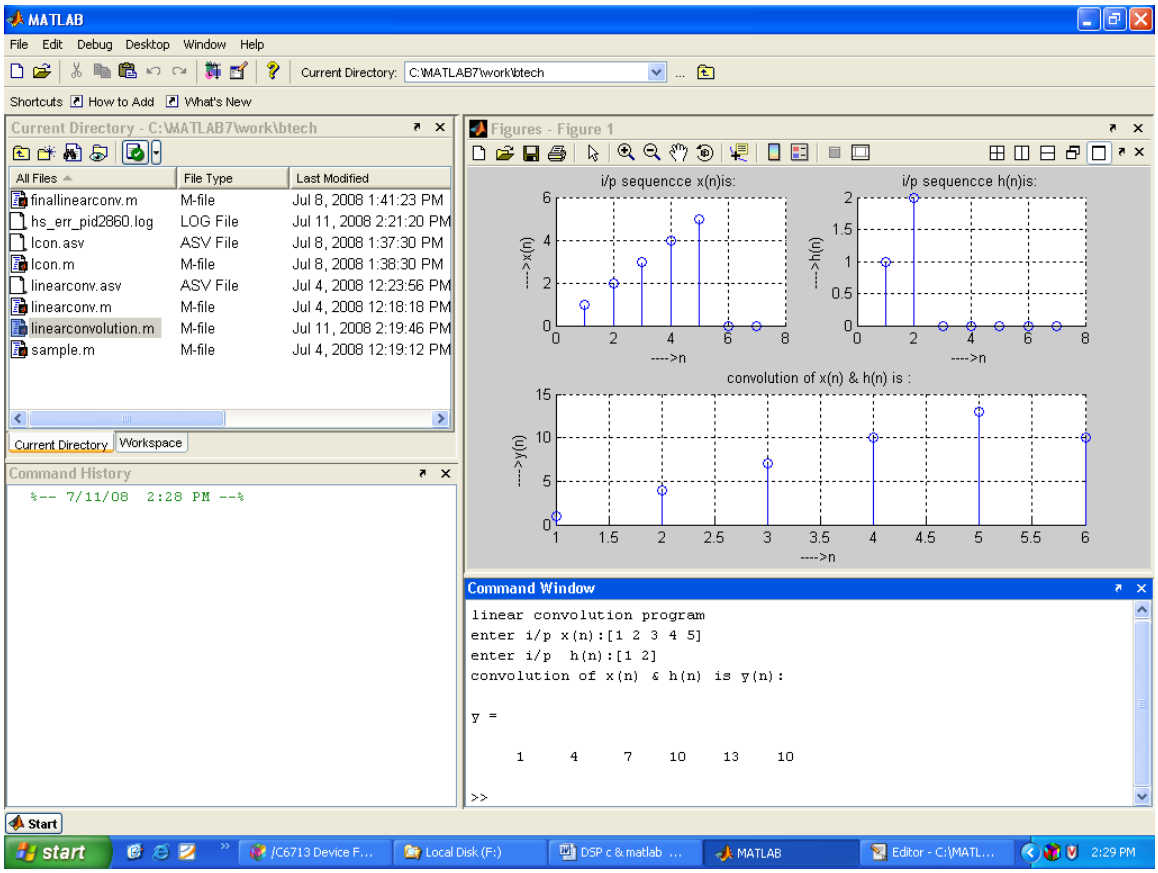
Result:

```
 enter value for m 5
 enter value for n 5
Enter values for i/p
1
2
3
4
5
Enter Values for h
1
2
3
4
5
 The Value of output y[0]=1
 The Value of output y[1]=4
 The Value of output y[2]=10
 The Value of output y[3]=20
 The Value of output y[4]=35
 The Value of output y[5]=44
 The Value of output y[6]=46
 The Value of output y[7]=40
 The Value of output y[8]=25
```

% MATLAB program for linear convolution

```
%linear convolution program

clc;
clear all;
close all;
disp('linear convolution program');

x=input('enter i/p x(n):');
m=length(x);
h=input('enter i/p  h(n):');
n=length(h);

x=[x,zeros(1,n)];
subplot(2,2,1), stem(x);
title('i/p sequencce x(n)is:');
xlabel('---->n');
ylabel('---->x(n)');grid;

h=[h,zeros(1,m)];
subplot(2,2,2), stem(h);
title('i/p sequencce h(n)is:');
xlabel('---->n');
ylabel('---->h(n)');grid;

disp('convolution of x(n) & h(n) is y(n):');

y=zeros(1,m+n-1);

for i=1:m+n-1
   y(i)=0;
   for j=1:m+n-1
      if(j<i+1)
      y(i)=y(i)+x(j)*h(i-j+1);
      end
   end
end
y
subplot(2,2,[3,4]),stem(y);
title('convolution of x(n) & h(n) is :');
xlabel('---->n');
ylabel('---->y(n)');grid;
```

Result :


MATLAB
File Edit Debug Desktop Window Help
Current Directory: C:\MATLAB7\work\btech
Shortcuts How to Add What's New
Current Directory - C:\MATLAB7\work\btech
All Files File Type Last Modified
finallinearconv.m M-file Jul 8, 2008 1:41:23 PM
hs_err_pid2860.log LOG File Jul 11, 2008 2:21:20 PM
Icon.asv ASV File Jul 8, 2008 1:37:30 PM
Icon.m M-file Jul 8, 2008 1:38:30 PM
linearconv.asv ASV File Jul 4, 2008 12:23:56 PM
linearconv.m M-file Jul 4, 2008 12:18:18 PM
linearconvolution.m M-file Jul 11, 2008 2:19:46 PM
sample.m M-file Jul 4, 2008 12:19:12 PM
Current Directory
Workspace
Command History
%-- 7/11/08 2:28 PM --%
Figures - Figure 1
i/p sequencce x(n)is:
i/p sequencce h(n)is:
convolution of x(n) & h(n) is :
----->x(n)
----->h(n)
----->y(n)
----->n
Command Window
linear convolution program
enter i/p x(n):[1 2 3 4 5]
enter i/p h(n):[1 2]
convolution of x(n) & h(n) is y(n):
y =
1 4 7 10 13 10
>>
Start
start
/C6713 Device F...
Local Disk (F:)
DSP c & matlab ...
MATLAB
Editor - C:\MATL...
2:29 PM

2. **Circular Convolution**

AIM

To verify Circular Convolution.

EQUIPMENTS:

Operating System – Windows XP
Constructor - Simulator
Software - CCStudio 3 & MATLAB 7.5

THEORY

Circular convolution is another way of finding the convolution sum of two input signals. It resembles the linear convolution, except that the sample values of one of the input signals is folded and right shifted before the convolution sum is found. Also note that circular convolution could also be found by taking the DFT of the two input signals and finding the product of the two frequency domain signals. The Inverse DFT of the product would give the output of the signal in the time domain which is the circular convolution output. The two input signals could have been of varying sample lengths. But we take the DFT of higher point, which ever signals levels to. For eg. If one of the signal is of length 256 and the other spans 51 samples, then we could only take 256 point DFT. So the output of IDFT would be containing 256 samples instead of 306 samples, which follows N1+N2 – 1 where N1 & N2 are the lengths 256 and 51 respectively of the two inputs. Thus the output which should have been 306 samples long is fitted into 256 samples. The 256 points end up being a distorted version of the correct signal. This process is called circular convolution.

PROGRAM:

```
/* program to implement circular convolution */

#include<stdio.h>
 int m,n,x[30],h[30],y[30],i,j, k,x2[30],a[30];
void main()
{
  printf("  enter the length of the first sequence\n");
  scanf("%d",&m);
  printf("  enter the length of the second sequence\n");
  scanf("%d",&n);
  printf("  enter the first sequence\n");
  for(i=0;i<m;i++)
  scanf("%d",&x[i]);
  printf("  enter the second sequence\n");
  for(j=0;j<n;j++)
  scanf("%d",&h[j]);

 if(m-n!=0)                          /*If length of both sequences are not equal*/
   {
            if(m>n)                  /* Pad the smaller sequence with zero*/
           {
```

```
            for(i=n;i<m;i++)
            h[i]=0;
            n=m;
           }
           for(i=m;i<n;i++)
           x[i]=0;
           m=n;
  }
     y[0]=0;
     a[0]=h[0];
    for(j=1;j<n;j++)                                    /*folding h(n) to h(-n)*/
    a[j]=h[n-j];

 /*Circular convolution*/
   for(i=0;i<n;i++)
     y[0]+=x[i]*a[i];
   for(k=1;k<n;k++)
   {
     y[k]=0;
      /*circular shift*/
     for(j=1;j<n;j++)
           x2[j]=a[j-1];

     x2[0]=a[n-1];
     for(i=0;i<n;i++)
     {
           a[i]=x2[i];
           y[k]+=x[i]*x2[i];
     }
   }

/*displaying the result*/
    printf("  the circular convolution is\n");
     for(i=0;i<n;i++)
    printf("%d \t",y[i]);
 }
```

OUTPUT:-

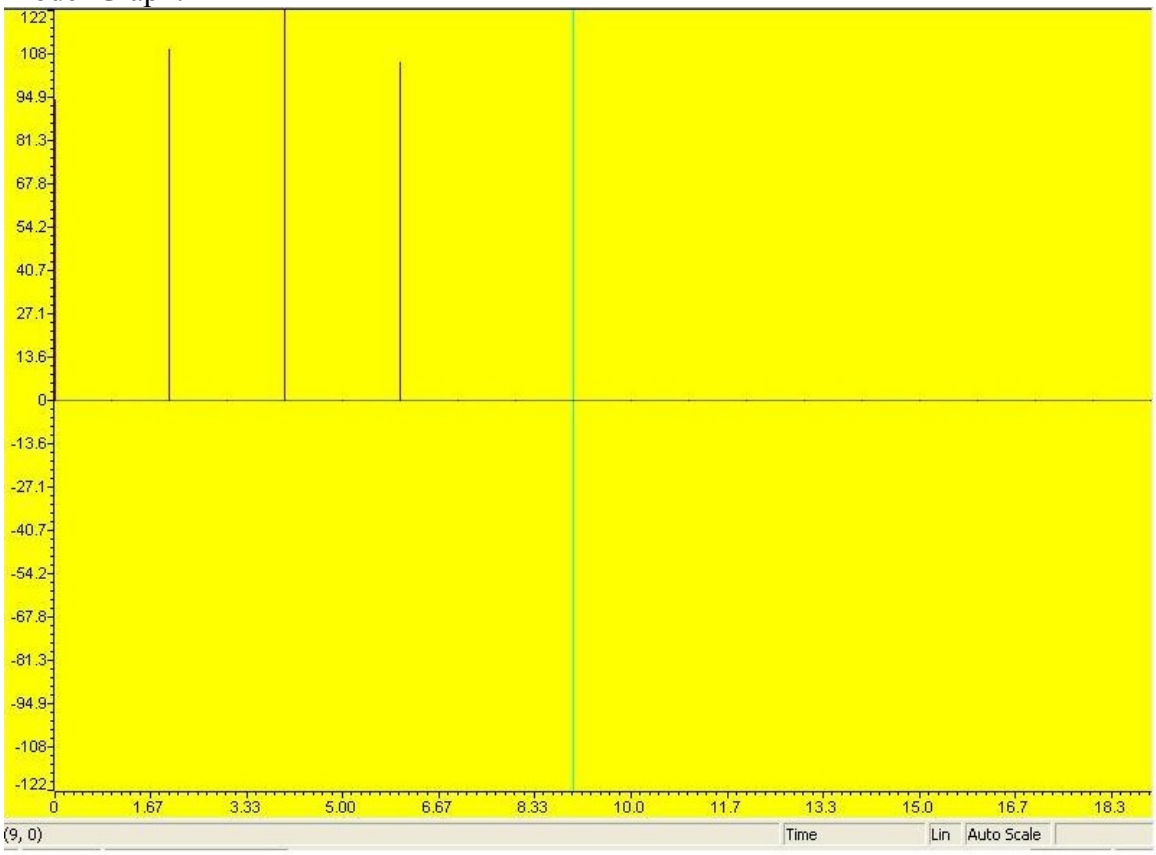
Enter the first sequence

5
6
7
Enter the second sequence
7
8
5
4

OUTPUT ;- the circular convolution is

94 110 122 106

Model Graph:-



%circular convolution program

```
clc;
clear all;
close all;
disp('circular convolution program');

x=input('enter i/p x(n):');
m=length(x);
h=input('enter i/p sequence h(n)');
n=length(h);

subplot(2,2,1), stem(x);
title('i/p sequencce x(n)is:');
xlabel('---->n');
ylabel('---->x(n)');grid;

subplot(2,2,2), stem(h);
title('i/p sequencce h(n)is:');
xlabel('---->n');
ylabel('---->h(n)');grid;

disp('circular convolution of x(n) & h(n) is y(n):');
if(m-n~=0)
   if(m>n)
      h=[h,zeros(1,m-n)];
      n=m;
   end
      x=[x,zeros(1,n-m)];
      m=n;
end

y=zeros(1,n);
y(1)=0;
a(1)=h(1);

for j=2:n
   a(j)=h(n-j+2);
end

%ciruclar conv
for i=1:n
   y(1)=y(1)+x(i)*a(i);
   end

for k=2:n
   y(k)=0;
   % circular shift
   for j=2:n
      x2(j)=a(j-1);
```

```
    end
    x2(1)=a(n);

    for i=1:n
        if(i<n+1)
            a(i)=x2(i);
        y(k)=y(k)+x(i)*a(i);
        end
    end
end
y
subplot(2,2,[3,4]),stem(y);
title('convolution of x(n) & h(n) is:');
xlabel('---->n');
ylabel('---->y(n)');grid;
```

Result :