

Signals and Systems

Lecture 11

Time Domain Models of Systems

Outline

➤ **Computation of the convolution sum in time domain (Continue).**

- ✓ **Graphical Method.**
 - **Graphical Method 1**
 - **Graphical Method 2**
- ✓ **Using Matlab.**

Graphical Method 1

The digital convolution algorithm to find $y[n]$ is to follow these steps (note that n is some fixed value; it will range over all values for which $y[n]$ are not zero):

1. Folding (reverse) one of the digital signal about the origin, $k = 0$

$$h[k] \xrightarrow{\text{folding}} h[-k].$$

2. Shift it along the time axis by one sample n to the right if n is a positive.

$$h[-k] \xrightarrow{\text{shift by } n} h[n-k]$$

3. Multiply the corresponding values of the two digital signals (multiply $x[k]$ by $h[n-k]$) to obtain the product sequence $v_n[k] = x[k]h[n-k]$.
4. Sum the products from step 3 (all the values of the product sequence $v_n[k]$) to get one point of the digital convolution at n .
5. Repeat **steps 2-4** to obtain the digital convolution at all times, n where the digital signals overlap.

The convolution of a M -length sequence with a N -length sequence leads to a sequence of length $M + N - 1$

To compute the convolution $y[n] = x[n] * h[n]$ of the signals $x[n] = \{1, 0, 3, -3\}$ where $-1 \leq n \leq 2$ and $h[n] = \{-2, 4, 1\}$ where $0 \leq n \leq 2$ as shown in figure 4-11 using the graphical Method 1.

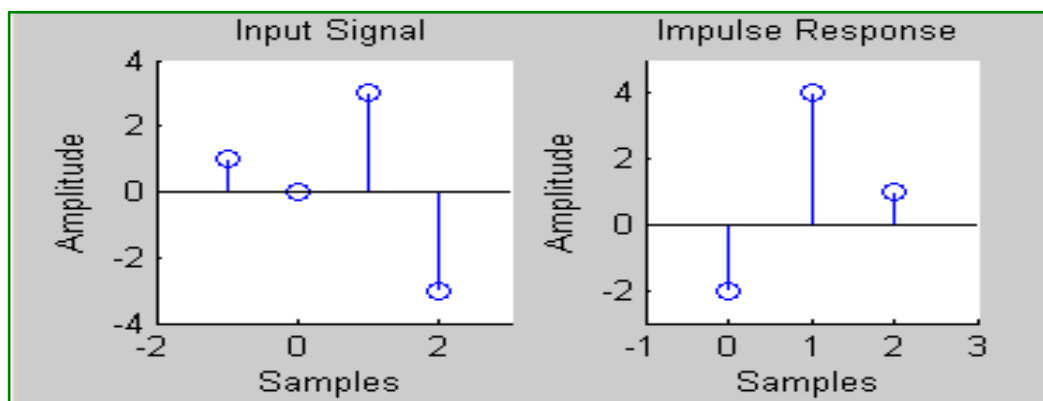
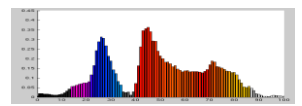


Figure 4-11



Dr. Qadri Hamarsheh

Solution: According to the steps of the graphical method 1, we obtain, the following figures:

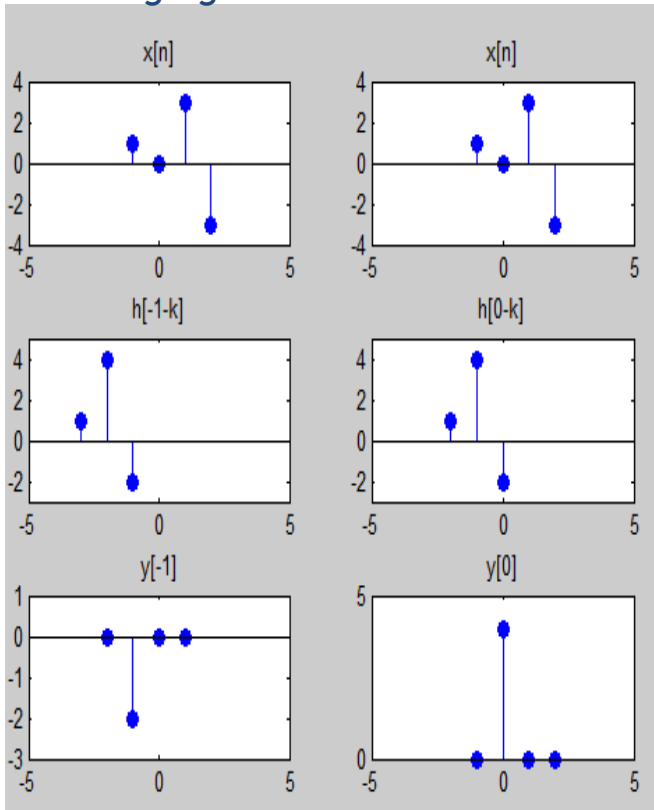


Figure 4-12

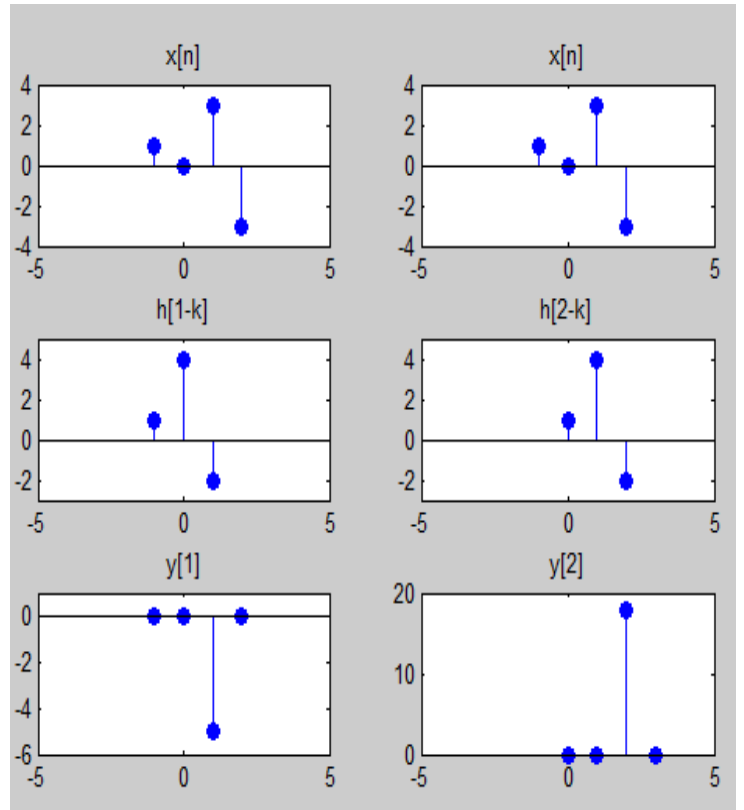


Figure 4-13

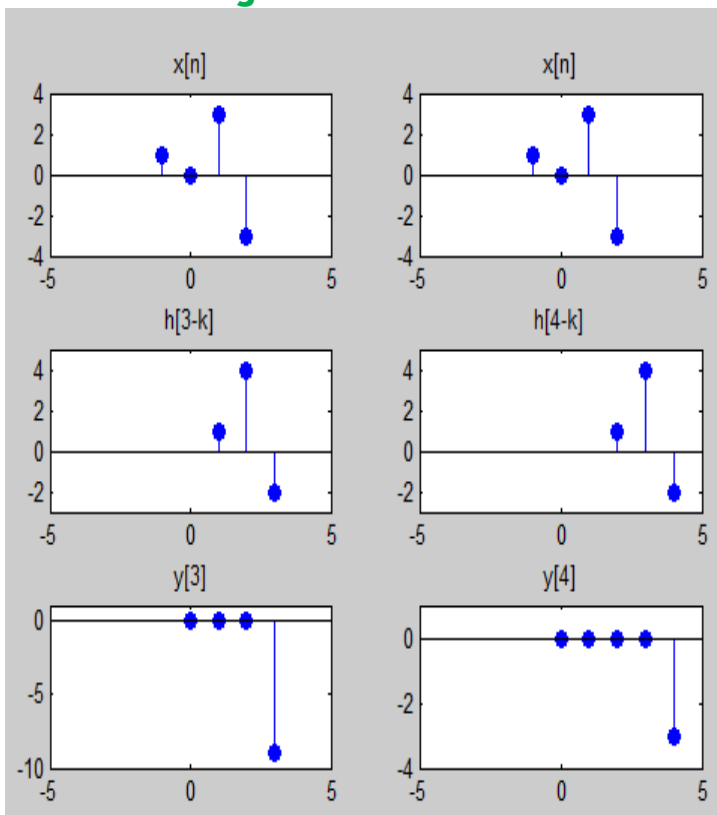


Figure 4-14

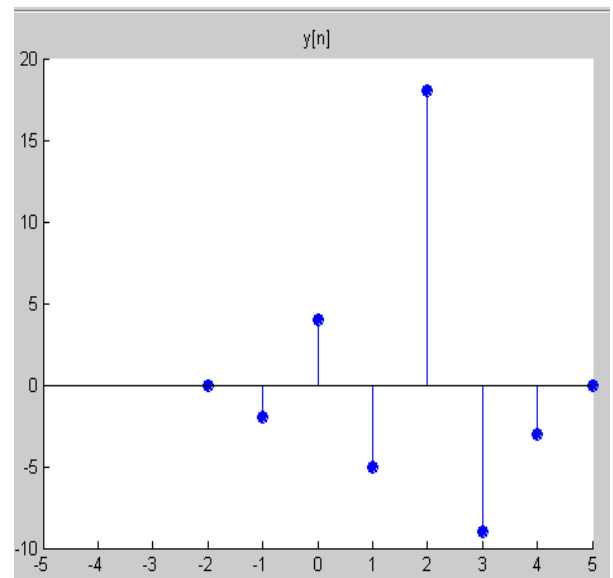
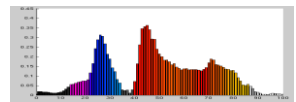


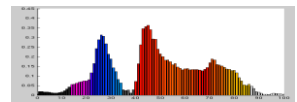
Figure 4-15



<pre> %Input Signal nx=[-1 0 1 2]; x=[1 0 3 -3]; subplot(3,2,1); stem(nx, x, 'filled'); title('x[n]'); axis ([-5 5 -4 4]); %Impulse Response h[-1-k] nh=[-3 -2 -1]; h=[1 4 -2]; subplot(3,2,3); stem(nh,h,'filled'); title('h[-1-k]'); axis ([-5 5 -3 5]); %displacement i =-1 %Intermediate output y[-1] ny=[-2 -1 0 1]; y=[0 -2 0 0]; subplot(3,2,5); stem(ny,y,'filled'); title('y[-1]'); axis ([-5 5 -3 1]); </pre>	<pre> %Input Signal nx=[-1 0 1 2]; x=[1 0 3 -3]; subplot(3,2,2); stem(nx, x, 'filled'); title('x[n]'); axis ([-5 5 -4 4]); %Impulse Response h[0-k] nh=[-2 -1 0]; h=[1 4 -2]; subplot(3,2,4); stem(nh,h,'filled'); title('h[0-k]'); axis ([-5 5 -3 5]); %displacement i=0 %Intermediate output y[0] ny=[-1 0 1 2]; y=[0 4 0 0]; subplot(3,2,6); stem(ny,y,'filled'); title('y[0]'); axis ([-5 5 0 5]); </pre>
<p>Matlab Code for Figure 4-12</p>	

<pre> %Input Signal nx=[-1 0 1 2]; x=[1 0 3 -3]; subplot(3,2,1); stem(nx,x,'filled'); title('x[n]'); axis ([-5 5 -4 4]); %Impulse Response h[1-k] nh=[-1 0 1]; h=[1 4 -2]; subplot(3,2,3); stem(nh,h,'filled'); title('h[1-k]'); axis ([-5 5 -3 5]); %displacement i =1 %Intermediate output y[1] ny=[-1 0 1 2]; y=[0 0 -5 0]; subplot(3,2,5); stem(ny,y,'filled'); title('y[1]'); axis ([-5 5 -6 1]); </pre>	<pre> %Input Signal nx=[-1 0 1 2]; x=[1 0 3 -3]; subplot(3,2,2); stem(nx,x,'filled'); title('x[n]'); axis ([-5 5 -4 4]); %Impulse Response h[2-k] nh=[0 1 2]; h=[1 4 -2]; subplot(3,2,4); stem(nh,h,'filled'); title('h[2-k]'); axis ([-5 5 -3 5]); %displacement i=2 %Intermediate output y[2] ny=[0 1 2 3]; y=[0 0 18 0]; subplot(3,2,6); stem(ny,y,'filled'); title('y[2]'); axis ([-5 5 0 20]); </pre>
<p>Matlab Code for Figure 4-13</p>	

<pre> %Input Signal nx=[-1 0 1 2]; </pre>	<pre> %Input Signal nx=[-1 0 1 2]; </pre>
---	---



<pre> x=[1 0 3 -3]; subplot(3,2,1); stem(nx,x,'filled'); title('x[n]'); axis ([-5 5 -4 4]); %Impulse Response h[3-k] nh=[1 2 3]; h=[1 4 -2]; subplot(3,2,3); stem(nh,h,'filled'); title('h[3-k]'); axis ([-5 5 -3 5]); %displacement i =3 %Intermediate output y[3] ny=[0 1 2 3 0]; y=[0 0 0 -9 0]; subplot(3,2,5); stem(ny,y,'filled'); title('y[3]'); axis ([-5 5 -10 1]); </pre>	<pre> x=[1 0 3 -3]; subplot(3,2,2); stem(nx,x,'filled'); title('x[n]'); axis ([-5 5 -4 4]); %Impulse Response h[4-k] nh=[2 3 4]; h=[1 4 -2]; subplot(3,2,4); stem(nh,h,'filled'); title('h[4-k]'); axis ([-5 5 -3 5]); %displacement i=4 %Intermediate output y[4] ny=[0 1 2 3 4]; y=[0 0 0 0 -3]; subplot(3,2,6); stem(ny,y,'filled'); title('y[4]'); axis ([-5 5 -4 1]); </pre>
<p>Matlab Code for Figure 4-14</p>	

<pre> %Output Signal y[n] ny=[-2 -1 0 1 2 3 4 5]; y=[0 -2 4 -5 18 -9 -3 0]; stem(ny,y,'filled'); title('y[n]'); axis ([-5 5 -10 20]); </pre>
<p>Matlab Code for Figure 4-15</p>

Graphical Method 2:

As you know that the convolution of two signals can be defined as the following

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] = \dots + h[-2]x[n+2] + h[-1]x[n-1] + h[0]x[n] + h[1]x[n-1] + h[2]x[n-2] + \dots$$

For causal systems, it will be reduced to

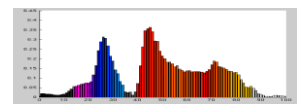
$$y[n] = h[0]x[n] + h[1]x[n-1] + h[2]x[n-2] + \dots$$

Each term in the above equation means that each sample of the impulse response is multiplying by the shifted version of the input signal (shifted value equal to the location of that sample).

The graphical method 2 used the shift and add algorithm to compute the convolution in the discrete case.

Algorithm steps:

1. Compute new sequences that are based on shifting and scaling of $x[n]$ based on the data contained in $h[n]$.



Dr. Qadri Hamarsheh

2. Add these sequences together to obtain $y[n]$.

To compute the convolution $y[n]=x[n]*h[n]$ of the signals $x[n]=\{1, 0, 3, -3\}$ where $-1 \leq n \leq 2$ and $h[n]=\{-2, 4, 1\}$ where $0 \leq n \leq 2$ as shown in figure 4-11 using the graphical Method 2.

Solution:

In this case, $h[n]$ has three nonzero data points, so we need to create these shifted and scaled copies of $x[n]$

❖ $n = 0$

First point $\Rightarrow h[0] = -2$, this tells us that we need to compute $-2x[n]$, we can do that by multiplying each point of $x[n]$ by -2 .

❖ $n = 1$

Second point $\Rightarrow h[1] = 4$, this tells us that we need to compute $4x[n-1]$, we can do that by shifting each data point $x[n]$ one position to the right, and we then multiply by 4.

❖ $n = 2$

Third point $\Rightarrow h[2] = 1$, this tells us that we need to compute $1x[n-2]$, we can do that by shifting each data point $x[n]$ two position to the right, and we then multiply by 1.

To obtain $y[n]$, we simply add up all the terms. We obtain the resulting sequence as shown in figure 4-16 and can be calculated as the following:

$$y[-2] = 0; \quad y[-1] = -2; \quad y[0] = 4; \quad y[1] = -5$$

$$y[2] = 18; \quad y[3] = -9; \quad y[4] = -3; \quad y[5] = 0$$

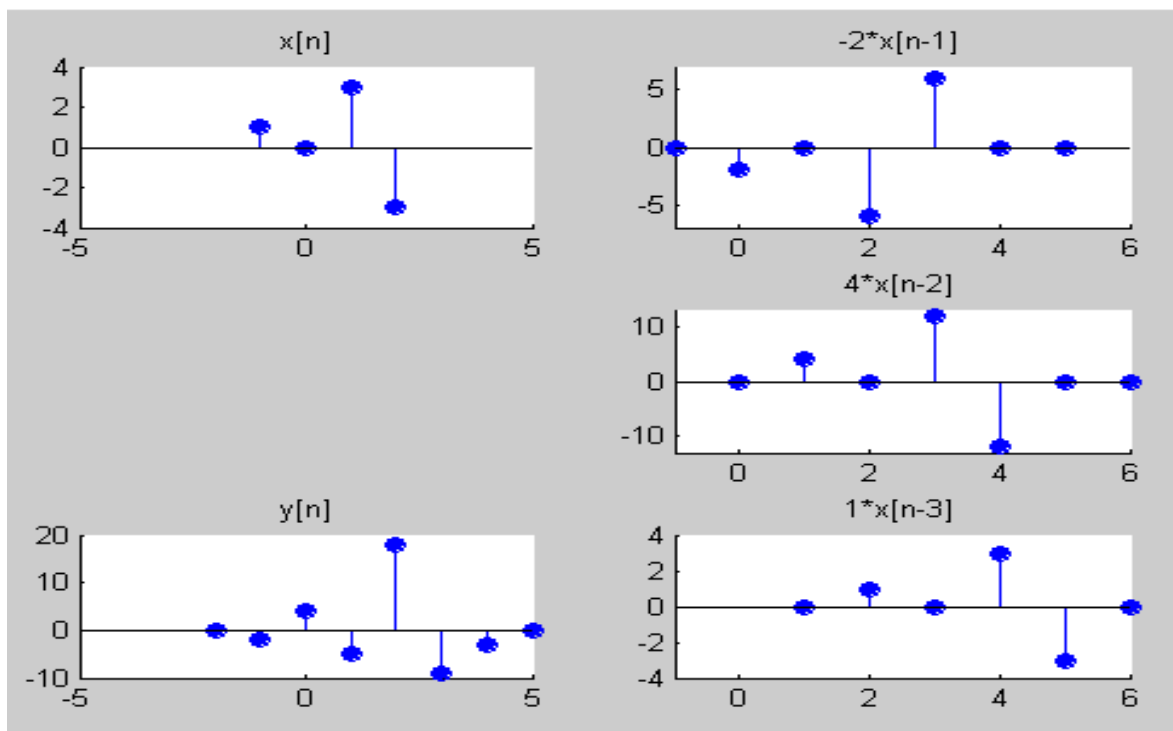
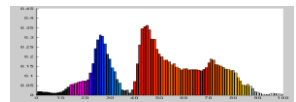


Figure 4-16

<pre>%Input Signal nx=[-1 0 1 2];</pre>	<pre>%copy of 1*x[n-3] nc=[1 2 3 4 5 6];</pre>
---	--



<pre> x=[1 0 3 -3]; subplot(3,2,1); stem(nx,x,'filled'); title('x[n]'); axis ([-5 5 -4 4]); %copy of -2*x[n-1] nc=[-1 0 1 2 3 4 5]; xc=[0 -2 0 -6 6 0 0]; subplot(3,2,2); stem(nc,xc,'filled'); title('-2*x[n-1]'); axis ([-1 6 -7 7]); %copy of 4*x[n-2] nc=[0 1 2 3 4 5 6]; xc=[0 4 0 12 -12 0 0]; subplot(3,2,4); stem(nc,xc,'filled'); title('4*x[n-2]'); axis ([-1 6 -13 13]); </pre>	<pre> xc=[0 1 0 3 -3 0]; subplot(3,2,6); stem(nc,xc,'filled'); title('1*x[n-3]'); axis ([-1 6 -4 4]); %Output Signal y[n] ny=[-2 -1 0 1 2 3 4 5]; y=[0 -2 4 -5 18 -9 -3 0]; subplot(3,2,5); stem(ny,y,'filled'); title('y[n]'); axis ([-5 5 -10 20]); </pre>
<p>Matlab Code for Figure 4-15</p>	

Convolution Using Matlab

Matlab function **conv(x,h)** performs a 1-D convolution of vectors x and h . The result is a vector with

$$Y_Length = length(x) + length(h) - 1$$

Imagine vector x as stationary and the flipped version of h is slid from left to right. Note that

$$conv(x,h) = conv(h,x)$$

Examples:

1. if $x = [1\ 2\ 3\ 4]$ and $h = [5\ 6]$, then:

$$conv(x,h) = 5\ 16\ 27\ 38\ 24$$

$$conv(h,x) = 5\ 16\ 27\ 38\ 24$$

2. To compute the convolution $y[n] = x[n] * h[n]$ of the signals $x[n] = \{1, 0, 3, -3\}$ where $-1 \leq n \leq 2$ and $h[n] = \{-2, 4, 1\}$ where $0 \leq n \leq 2$ using the Matlab program that computes and plots the output.

Solution:

```

n=-1:4;
x=[1 0 3 -3];
h=[-2 4 1];
y = conv(x,h);
stem(n,y,'filled');
xlabel('Samples');
ylabel('Amplitude');
title('Output Signal');

```

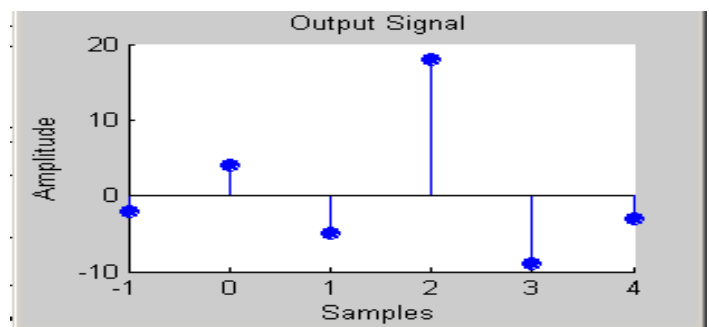
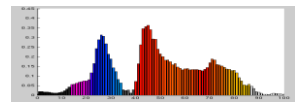


Figure 4-17



Dr. Qadri Hamarsheh

3. Suppose that $p[n]=1$, for $3 \leq n \leq 12$, $p[n]=0$ for all others, compute the convolution $p[n]*p[n]$.

Solution:

```
p=[0 0 0 ones(1,10) zeros(1,7)];
% 20 point of signal p
n=0:19;
%correspond to n=0 to 19
y = conv(p, p);
n=0:38;
stem(n,y,'filled');
```

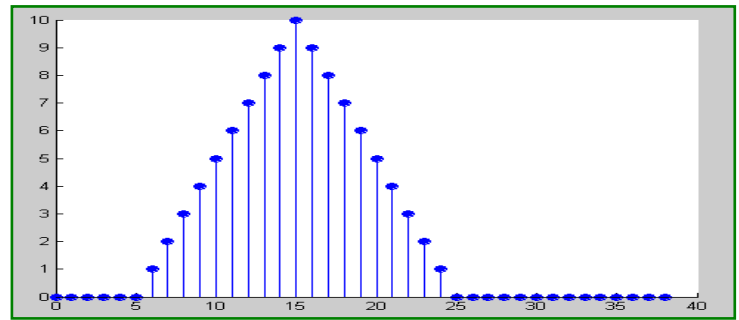


Figure 4-18

4. Suppose that the unit pulse response $h[n]=5\sin(\frac{2}{3}n)$ for $n \geq 0$ and the input $x[n]=3\cos(\frac{3}{5}n + 0.2\pi)$ for $n \geq 0$. Compute the response $y[n]$ for $n = 0,1,\dots,30$

```
n=0:30;
%the required output points
h = 5*sin(2*n/3);
x = 3*cos(3*n/5 + 0.2*pi);
y = conv(x,h);
stem(n,y(1:length(n)),'filled');
```

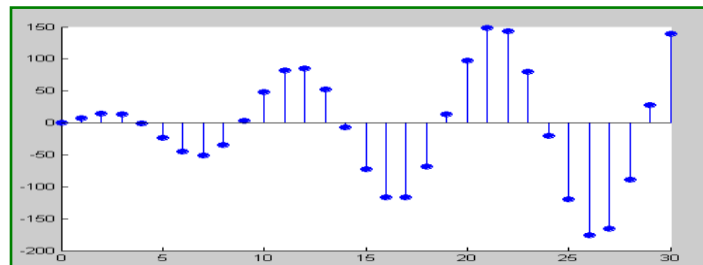


Figure 4-19

5. Computation the convolution between two signals interactively

```
x = input('Input the signal values: ');
h = input ('Input the impulse response values: ');
xlen = length(x); hlen = length(h); ylen = xlen+hlen -1;
y = conv(x,h);
n=0:ylen-1;
stem(n,y,'filled');
xlabel('Output Samples'); ylabel('Values');
title('Convolution between signals'); axis auto
```

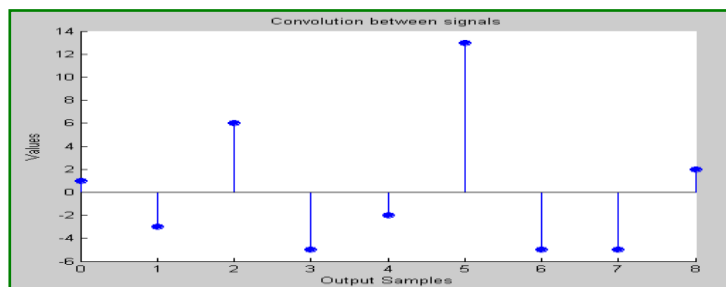


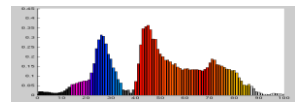
Figure 4-20

The run of the program will be:

Input signal values: [1 -1 3 0 -3 1]

Input the impulse response values: [1 -2 1 2]

The output is shown in figure 4-20



Dr. Qadri Hamarsheh

We cannot compute the convolution for noncausal systems using matlab

If the first nonzero value of a finite-length sequence $x[n]$ occurs at index $n=i$ and has a value $x[i]=v$, then the index of the first nonzero value in the convolution $y[n]=x[n]*x[n]$ the index is $n=i+i$, and the value is $v \times v$.