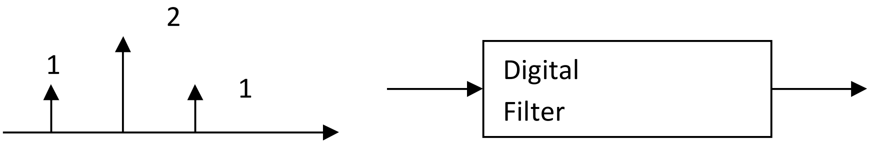
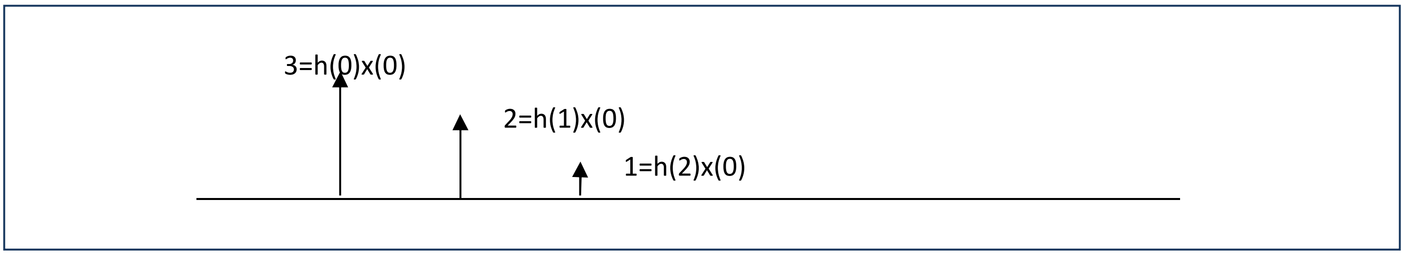
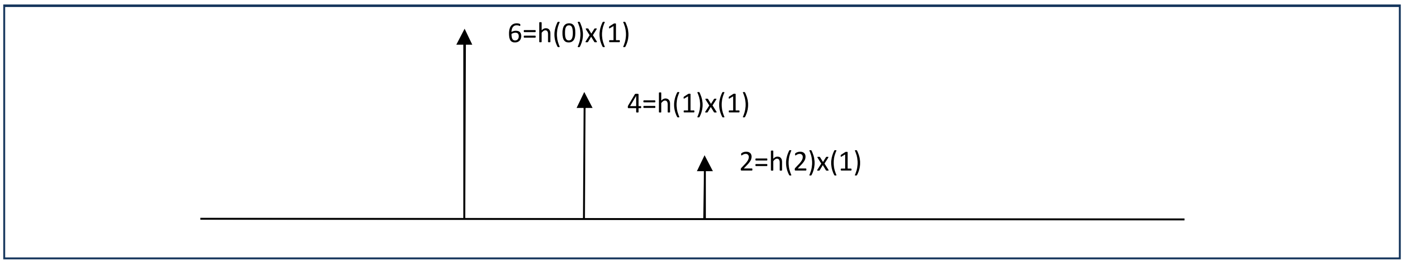
Digital convolution

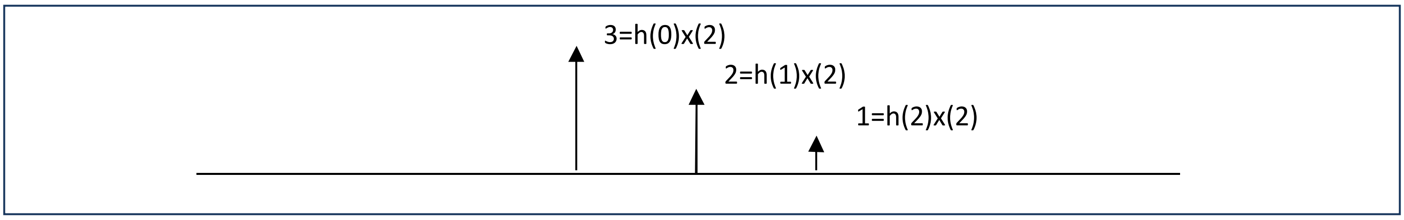
Let a filters impulse response be defined as *h(n).* For example, let *h(n)=[3,2,1].* If an *impulse* is applied to this filter, then its output will be *[3,2,1].* If an *impulse* of amplitude *5* is applied to this filter, then its output is *[15,10,5].* Now lets consider what happens when the signal x(n)=[1,2,1] is applied to the filter input.

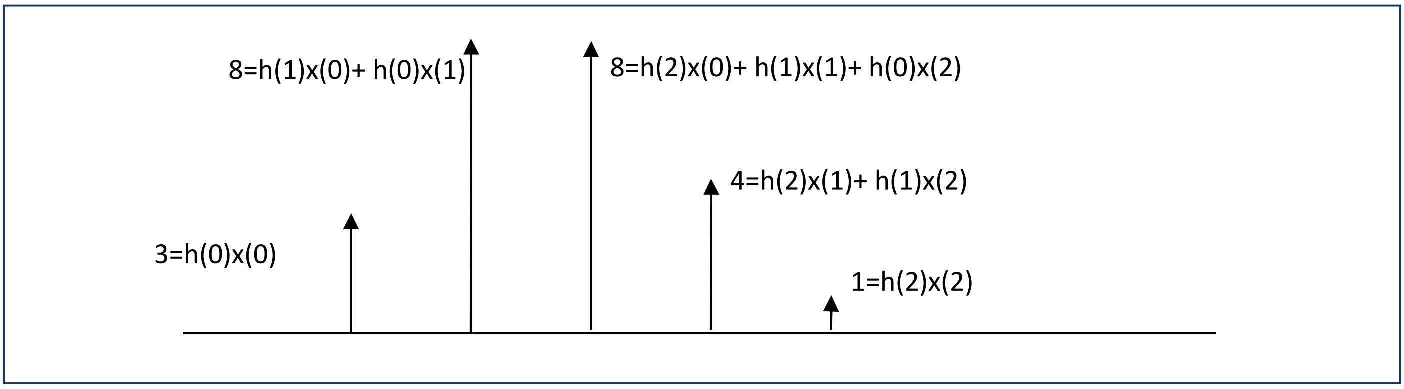


The first impulse *x(0)=1* produces the output *[3,2,1].* The next impulse produces *[6,4,2]* except that it is one sample period later, and finally the third impulse produces *[3,2,1]* again but delayed by 2 sample periods. The final output is obtained by combining the filters response to the three inputs.









This process can now be generalised into what is referred to as the convolution equation. This states that the output *y(n)* of the filter is:

$$y(n) = \sum_{m=-\infty}^{m=\infty} h(m)x(n-m)$$
$$y(2) = \sum_{m=0}^{m=\infty} h(m)x(2-m)$$
$$y(2) = h(0)x(2) + h(1)x(1) + h(2)x(0)$$

Note that the convolution sum is symmetrical in that it can also be expressed as:

$$y(n) = \sum_{m=-\infty}^{m=\infty} x(m)h(n-m)$$
$$y(2) = \sum_{m=0}^{m=\infty} x(m)h(2-m)$$
$$y(2) = h(2)x(0) + h(1)x(1) + h(0)x(2)$$

This is exactly the same, except that it is written the other way around. So we count up with 1 value and down with the other, for example:

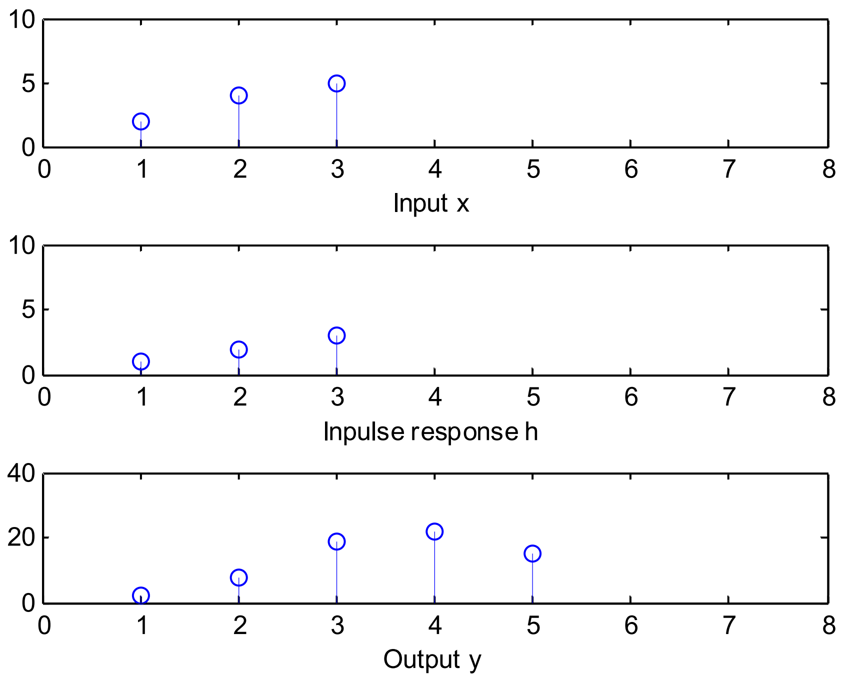
$$y(10) = h(0)x(10) + h(1)x(9) + h(2)x(8) + \ldots h(10)x(0)$$

Problem

A system has an impulse response h(n)=[1,3,4]. The signal x(n)=[1 2 3] is applied to the input of this system. Determine using a digital convolution technique, the output from the system up to n=8.

The Matlab code below produces convolution by 1st principles:

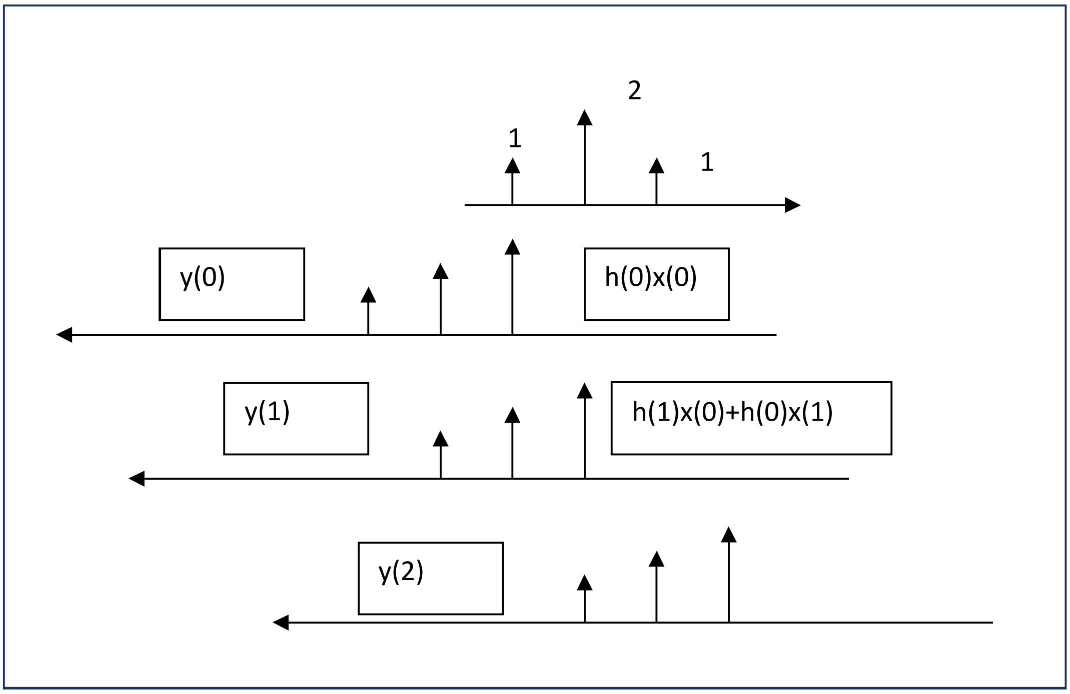
```
xx=[]; hh=[];
x=[2 4 5]
h=[1 2 3]
N=length(x)+length(h)-1
padx=N-length(x); padx=[zeros(1,padx)];
padh=N-length(h); padh=[zeros(1,padh)];
xx=[x padx]
hh=[h padh]
y=zeros(N,1)
for n=0:N-1
    for m=0:n
        y(n+1)=y(n+1)+hh(m+1)*xx(n-m+1);
    end
end
y
subplot(3,1,1), stem(x); axis ([0,8,0,10]); xlabel 'Input x'
subplot(3,1,2), stem(h); axis ([0,8,0,10]); xlabel 'Inpulse response h'
subplot(3,1,3), stem(y); axis ([0,8,0,40]); xlabel 'Output y'
```



Above uses definition of convolution with an offset of 1 for Matlab indices. The input x and h values are padded out with zeros to convolve up to the sum of the 2 sequence lengths minus 1. Check this with the Matlab conv(x,h) function.

Fold & Shift Technique

A convenient way to visualise the convolution process is to use a folding and shifting process. First we time reverse h(m) and slide the function through x(m). This is a graphical way of implementing the convolution equation. See the process below.



Now by multiplying and adding the elements in each column, the convolution sum is obtained. We can check a typical convolution example in Matlab, as follows:

```
» a=[1,2,1]
» b=[3,2,1]
» c=conv(a,b)
c = 3    8    8    4    1
```