**NATIONAL INSTITUTE OF TECHNOLOGY CALICUT**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

****

**EC 4091: DIGITAL SIGNAL PROCESSING LABORATORY**

EXPERIMENT No. 2

**Submitted by:**

B.Veda Sree - B150734EC

B.Sravani - B150783EC

Ch.Priyanka - B150837EC

Ch.Vamsi Narayana Reddy - B150530EC

**Aim:** To write programs in matlab to

1. Find convolution of two sequences and verify with in-built function.

2. Find the response of an LTI system to step, rectangular,exponential and sinusoidal inputs.

3. Implement block convolution using

a. Overlap and add method and

b. Overlap and save method

**THEORY:**

**CONVOLUTION:** Convolution is defined as the integral of the product of the two functions after one is reversed and shifted. The convolution of f and g is written f\*g, using an [asterisk](https://en.wikipedia.org/wiki/Asterisk) or star. As such, it is a particular kind of [integral transform](https://en.wikipedia.org/wiki/Integral_transform).

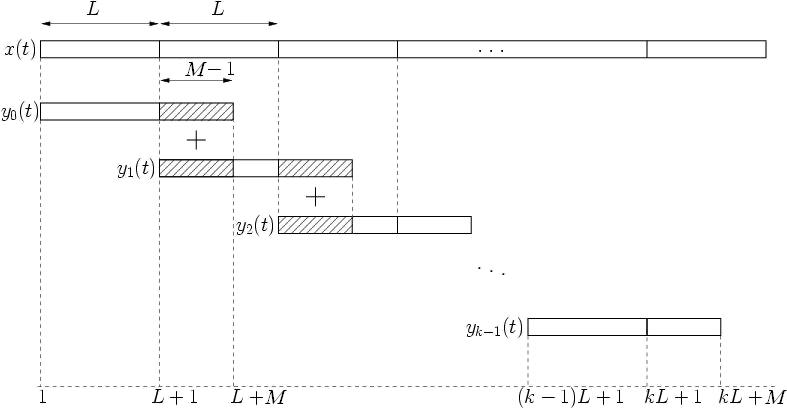
The most important application of the convolution integral is that it can be used to generate the output of an LTI system from the input function and the impulse response of the system.

In continuous domain, if x(t) is the input signal, y(t) the output signal and h(t) the impulse response of the system,

In discrete domain, if x[n] is the input signal, y[n] is the output and h[n] the impulse response of the system,

For longer sequences of input signal x[n], especially in real time processing, convolution is done with segments of input; and the results are combined in a special manner to generate the output. There are two methods of doing this: Overlap-add and overlap-save.

**Overlap Add Method**

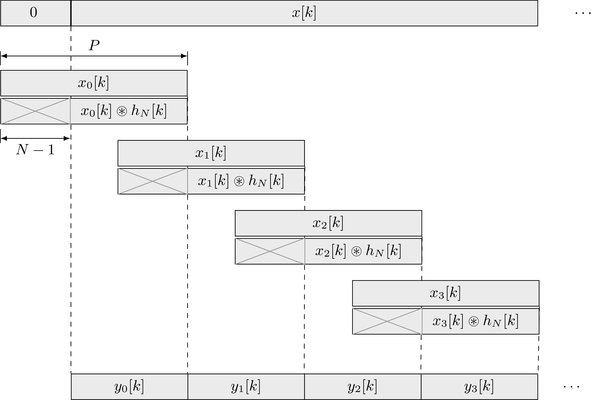
****

**Fig 2.1 Overlap Add method**

Consider a long input sequence of length *l* being convoluted with a filter of length *m*. In the overlap-add method, small segments of input (length L) are taken at a time to convolute with the filter function (length M). The last M-1samples of each segment’s output (length L+M-1) is added with the first M-1 samples of the next segment’s output. The last such output segment is left as such: hence the total length of the output sequence remains the same as *l+m-1*.

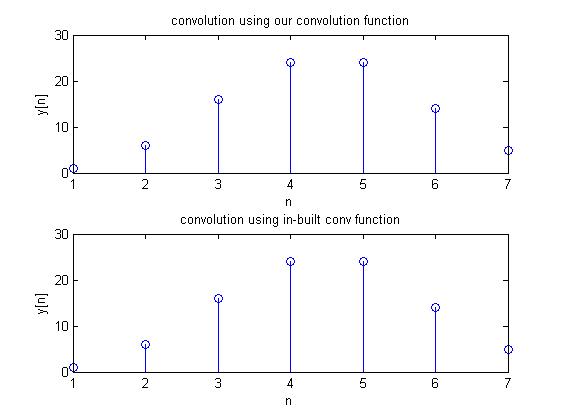
**Overlap Save Method**

Again consider a long input sequence of length *l* being convoluted with a filter of length *m*. In the overlap save method, the input sequence is first appended with M-1 zeros prior to the start of the input sequence, making its total length *l+m-1*. Then, as above, segments of length L are taken as input to convolve with filter of length M. From the thus-generated output segments, discard the first M-1 samples each time, and simply combine the remaining part of the output to generate the final long output sequence .



**Fig 2.2 Overlap save method**

a)



a) Write a program to perform convolution of two sequences given by

Y[n]=

Create any two arbitrary sequences x[n] and h[n] of short lengths and compute the result using your program. Use the “conv” function in Matlab to verify your result.

**MATLAB CODE:**

clear all

close all

x=input('enter input sequence\n');

h=input('enter impulse response sequence\n');

display('convolution using our convolution function');

display(convolution(x,h));

display('convolution using in-built conv function');

display(conv(x,h));

subplot(2,1,1),stem(convolution(x,h));

title('convolution using our convolution function');

xlabel('n');

ylabel('y[n]');

subplot(2,1,2),stem(conv(x,h));

title('convolution using in-built conv function');

xlabel('n');

ylabel('y[n]');

**convolution function:**

function res = convolution(x,h)

l = length(x);

m = length(h);

x = [x,zeros(1,m-1)];

res = zeros(1,l+m-1);

for i = 1:m-1

for j = 1:i

res(i) = res(i) + h(j)\*x(i-j+1);

end

end

for i= m:l+m-1

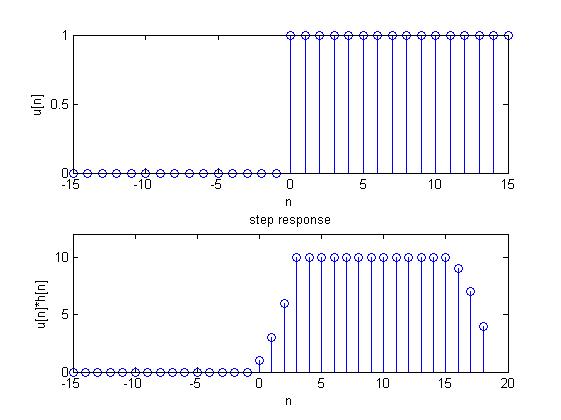
for j = 1:m

res(i) = res(i) + h(j)\*x(i-j+1);

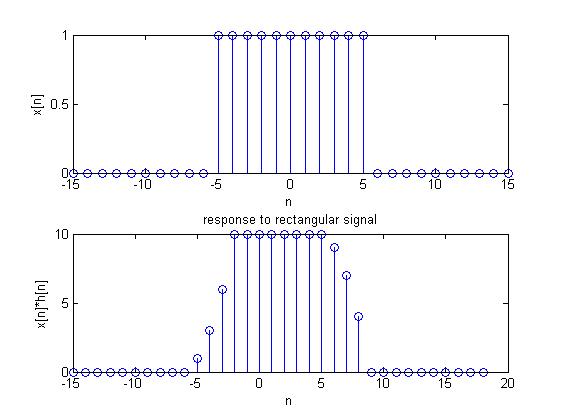
end

end

b) Step input



Rectangular input



b)Use the above program (you can define it as a function) to find the step response of an LTI system given the impulse response(Take h(n)={1,2,3,4}). Also compute the response of the same system to a rectangular pulse, sinusoidal and exponential signal.

**MATLAB CODE:**

%%step response

n=-15:15;

step=(n>=0);

h=[1 2 3 4];

StepResponse=convolution(step,h);

figure,subplot(211),stem(n,step),

ylabel('u[n]'),xlabel('n');

subplot(212),stem((-15:18),StepResponse)

title('step response')

ylabel('u[n]\*h[n]'),xlabel('n');

ylim([0,12]);

%% rectangular

n=-15:15;

rect=(n>=-5 & n<=5);

h=[1 2 3 4];

res=convolution(rect,h);

figure,subplot(211),stem(n,rect),

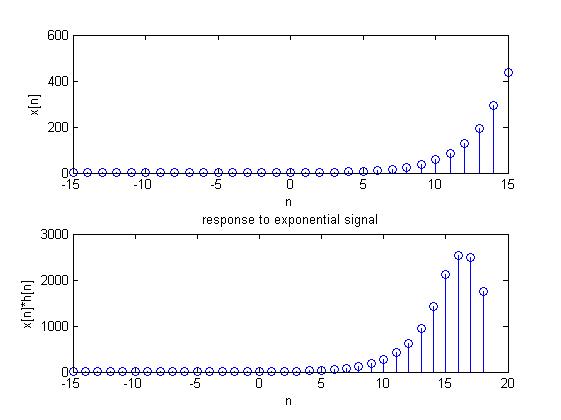
ylabel('x[n]'),xlabel('n');

subplot(212),stem((-15:18),res)

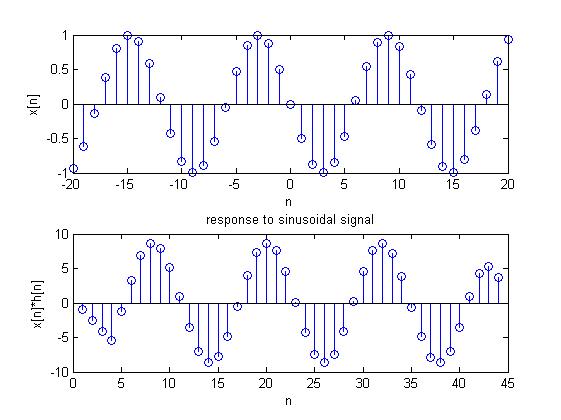
title('response to rectangular signal')

ylabel('x[n]\*h[n]'),xlabel('n');

Exponential input



Sinusoidal signal



%% exponential

n=-15:15;

exp=(1.5).^n;

h=[1 2 3 4];

res=convolution(exp,h);

figure,subplot(211),stem(n,exp),

ylabel('x[n]'),xlabel('n');

subplot(212),stem((-15:18),res)

title('response to exponential signal')

ylabel('x[n]\*h[n]'),xlabel('n');

%% sinusoidal

n=0:30;

sine=sin(2\*pi\*n);

h=[1 2 3 4];

res=convolution(sine,h);

figure,subplot(211),stem(n,sine),

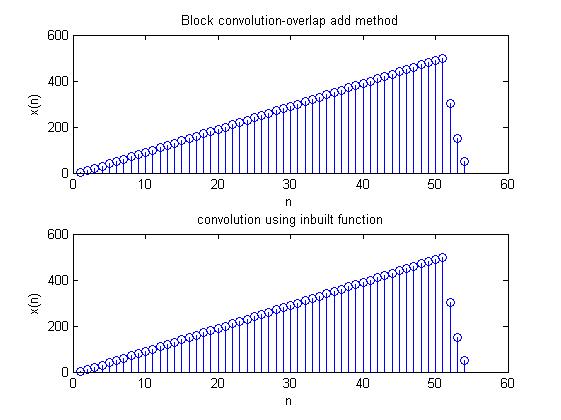
ylabel('x[n]'),xlabel('n');

subplot(212),stem((0:33),res)

title('response to sinusoidal signal')

ylabel('x[n]\*h[n]'),xlabel('n');

c)



**c)** Write a program to implement block convolution using

(a)Overlap Add method and

(b)Overlap Save method.

Demonstrate your results using appropriate sequences.

**MATLAB CODE:**

%**% overlap and add method**

x=(1:51);

h=[4 3 2 1];

L=3;

y=0;

M=length(h);

res = zeros(1,length(x)+M-1);

for v = 1: length(x)/L

xtemp = x(((v-1)\*L+1):v\*L);

ytemp = convolution(xtemp,h);

res(((v-1)\*L+1):(v\*L)+M-1) = res(((v-1)\*L+1):(v\*L)+M-1) + ytemp;

end

subplot(211), stem(res)

title('Block convolution-overlap add method')

xlabel('n');

ylabel('x(n)');

subplot(212), stem(conv(x,h))

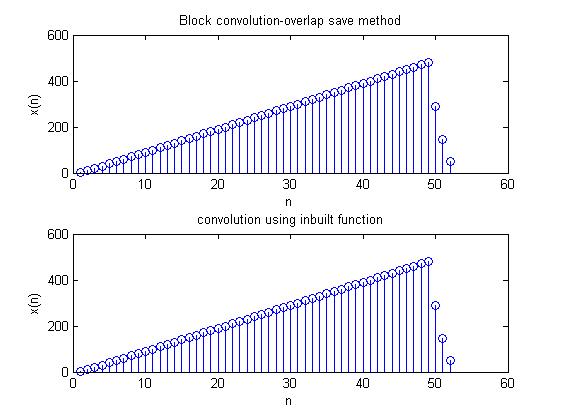
title('convolution using inbuilt function')

xlabel('n');

ylabel('x(n)');

fprintf('result')

disp(res);

****

%**Overlap and save**

clc;

clear var;

x=(1:49);

h=[4 3 2 1];

N=10;

m=length(h);

l=N-(m-1);

y=zeros(1,length(h)+length(x)-1);

pre=zeros(1,m-1);

for i=1:(length(x))/l

xb=[pre,x((i-1)\*l+1:l\*i)];

yb=conv(xb,h);

y((i-1)\*l+1:l\*i)=yb(m:m+l-1);

pre=xb(l+m-1-(m-1)+1:l+m-1);

end

n=length(xb)+length(h)-1;

y(length(x)+1:length(x)+m-1)=yb(n-(m-1)+1:n);

w=conv(x,h);

figure;

subplot(211), stem(y)

title('Block convolution-overlap save method')

xlabel('n');

ylabel('x(n)');

subplot(212), stem(w)

title('convolution using inbuilt function')

xlabel('n');

ylabel('x(n)');

**OBSERVATIONS AND INFERENCES**

Large sequences of data will require a sizeable amount of memory to hold the values for convolution in the traditional method. This also increases delay in the system. Block convolution is implemented to forego this. It takes only small sequences at a time.

**RESULT**

Convolution algorithm was written and the result was verified with that of the in-built MATLAB function, conv. A series of inputs (rectangular pulse, exponential signal, sinusoidal signal) was convoluted with a given LTI system response h[n] and the output was verified with conv function. Block convolution was implemented using overlap and add and overlap and save methods and verified using conv function.