Experiment 1) Discrete time convolution

Tutorial:

The convolution sum expresses the relation between the input signal of a linear time invariant system and its output. The expression of the discrete time convolution is:

$$y(n) = \sum_{m=-\infty}^{\infty} h(m)\,x(n-m)$$

We can think about the convolution sum as a weighted sum of shifted versions of input vector. The weights of the different versions are determined by the impulse response of the system

y[n]= ……+h[-1] x[n+1] +h[0] x[n]+h[1] x[n-1] +……

So, to get the result of convolution, you have to add different versions of the signal multiplied by a number that depends on the system.

Assume x[n] starts from a_x, and ends at b_x

Assume h[n] starts from a_h, and ends at b_h

Then y[n] starts at a_x+a_h and ends at b_x+b_h

In MATLAB, we treat the signal and its time axis independently.

The complete algorithm to do the convolution is:

1- Define nx, x , nh, h

2- ny ”the time index of the result” as a vector:

starting from nx(1)+nh(1) and ends at nx(end)+nh(end)

3- Initialize the output vector y to all zeros, its length should equal to

length(x)+length(h)-1

4- Loop over all elements of the vector h, at each iteration add a shifted version of the signal to the old y

5- End

Students experiment:

1- An outline of the code required to do the previous algorithm follows. You are required to complete the code, and stem the result of convolution between the two following vectors:

nx=[-3 -2 -1];

nh=[-6 -5 -4 -3 -2 -1];

x=[1 2 3];

h=[9 8 5 32 5 3];

The outline is:

M=length(x);

N=length(h);

ny= ……

y=zeros(1,…..);

for u=1:……

x1=h(u)*[zeros(1,u-1) x zeros(1,……)];

y=……+x1;

End

2- The convolution can be understood in another way, namely as inversion for one signal, shifting it, multiplying it by the other signal and adding the result of multiplication to get the result of one sample of the convolution output. The code follows implements this vision. You are required to complete the code, describe how this code implements this vision of convolution.

```
N=length(x);M=length(h);
ymat=zeros(1,….);
for n=1:……
    for k=max(1,n-M+1):min(n,…….)
    ymat(n)=…..+x(k)*h(n-k+1);
    end
end % please note that ymat(1) is the y[0] of the analytical solution
```

Experiment 2) The (conv) command:

Students experiment:

The MATLAB conv command implements the discrete time convolution, see the MATLAB help and try to use it to verify your results in the previous part

Experiment 3) Filters:

Tutorial:

A discrete time filter is just a discrete time system with certain impulse response. The impulse response of the filter determines its characteristics; either low pass, high pass … etc. The relationship between the input and the output of the filter can be also expressed in another form called representation using finite difference equations:

$a(1)*y(n) = b(1)*x(n) + b(2)*x(n-1) + ... + b(n_b+1)*x(n-n_b)- a(2)*y(n-1) - ... - a(n_a+1)*y(n-n_a)$

Any system that can be expressed as a finite difference equation is an LTI systems , we can prove time invariance by entering new input for the system called x2(n) equals to x1(n-N) "delayed version of the original input, we will find

(First input): (1)

$a(1)*y1(n) = b(1)*x1(n) + b(2)*x1(n-1) + ... + b(n_b+1)*x1(n-n_b)- a(2)*y1(n-1) - ... - a(n_a+1)*y1(n-n_a)$

(Second input=delayed version of input) "$x_2(n)=x_1(n-N)$" (2)

$a(1)*y2(n) = b(1)*x2(n) + b(2)*x2(n-1) + ... + b(n_b+1)*x2(n-n_b) - a(2)*y2(n-1) - ... - a(n_a+1)*y2(n-n_a)$

$a(1)*y2(n) = b(1)*x1(n-N) + b(2)*x1(n-1-N) + ... + b(n_b+1)*x1(n-n_b-N)- a(2)*y2(n-1) - ... - a(n_a+1)*y2(n-n_a)$

By delaying the output of the first input by N "means replace every n by n-N in the first equation, we will find that $y_2(n)=y_1(n-N)$, this means that the system is time invariant.

If we want to prove linearity, we will enter $x_1(n)$ on the system, get $y_1(n)$, then enter $x_2(n)$ to get $y_2(n)$, then enter $x_3(n)=a*x_1(n)+b*x_2(n)$, if the output y3=a*y1(n)+b*y2(n), then the system is Linear, this can be easily verified if we multiplied eq (1) by a, eq(2) by b and add the two equations, we will find that we got new input a*x1(n)+b*x2(n) with new output $a*y_1(n)+b*y_2(n)$

If all the a coefficients equals to zero except a(1), then we have a finite impulse response system. An example of a famous FIR that acts as a low pass filter is the moving average filter.

Y[n]=(x[n]+x[n-1+x[n-2]])/3;

The output of the system at any time instant equals to the average of the previous sample, the current sample and the future sample. Here we have b(1)=b(2)=b(3)=1/3

"*Moving average filter of order 3*"

If any of the a's other than a(1) doesn't equal to zero, then we have infinite impulse response system, the current output depends on all the previous inputs till the output moment. An example of this system is

Y[n]=.5 y[n-1]+x[n].

The previous inputs affects the current output through the term y[n-1].

To know the output of a system described by a difference equation, MATLAB has the (filter) command.

Students experiment:

Apply the input x=ones(1,10) to the following systems:

a) y[n]=x[n]+.5x[n-1]+2x[n-2]

b) y[n]-.8y[n-1]=x[n]

c) y[n]-.8y[n-1]=x[n-1]

- What's the expected steady state output for the first system?
- What's the relationship between the output of the 2nd and 3rd systems
- Design a moving average filter of order five. Use it to filter the following input

x=[1 1 1 1 1 1 1 2 1 1 1 1]

Comment on your results.

d) y[n]=x[n]-x[n-1]

Apply an input = [1 1 1 1 1 2 1 1 1 1 1] to the system, determine its output

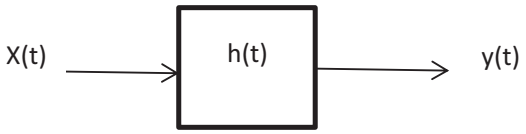
Comment on your results

System representation

3.1 Time domain (Impulse response):

- The system is represented by its impulse response, and the relation between the input and the output of the system is as follows:

  y(t) = x(t) * h(t)



- Matlab performs only discrete convolution:

  Z = x*y

  Z[n] = x[n] * y[n]

  $= \sum x[k]y[n-k]$

  To implement continuous convolution, we must approximate it into discrete one:

  Z(t) = x(t) * y(t)

  $= \int_{-\infty}^{\infty} x(\tau)y(t-\tau)d\tau$

  $\cong \sum x(\tau)y(n-\tau)\Delta\tau$

  $\cong Ts$. discrete convolution (x,y)

3.2 Matlab function of convolution:

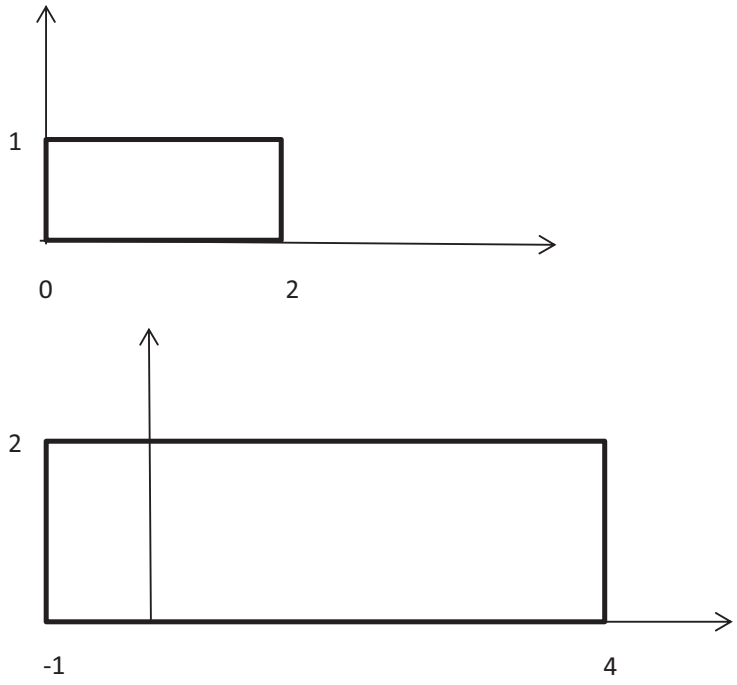
For continuous signals,

Z = $\frac{1}{fs}$ conv(x,y)

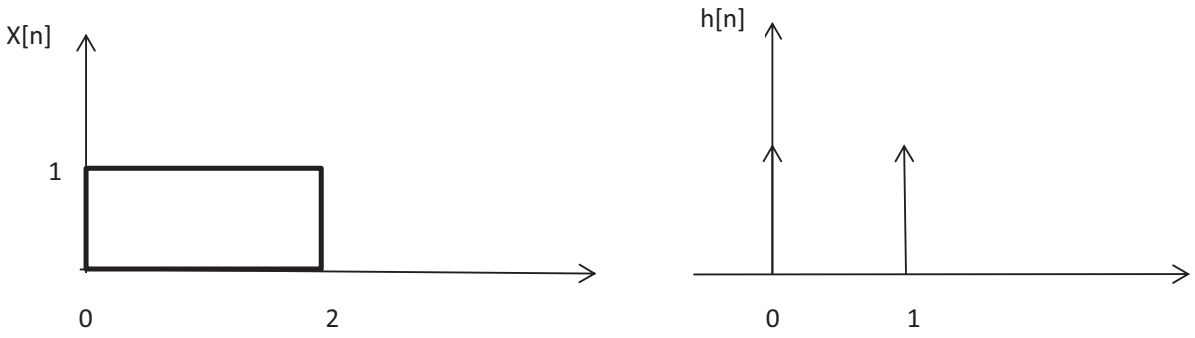
- Start time for the output of the convolution (z) = start time of (x) + start time of (y)
- End time for the output of the convolution (z) = End time of (x) + End time of (y)
- Length of the output (z) = length of (x) + length of (y) - 1

3.5 Exercises

1. For sampling frequency, fs 1000, find the output of the convolution between the two rectangular signals:



2. Find the output of the system h[n] for the input x[n], and find the transfer function.



(1) Compute and plot the convolution between the following pair of sequences:

(a) $x_1[n] = \left[\overset{\frown}{1}, 2,4\right], h_1[n] = \left[1,1,\overset{\frown}{1},1,1\right].$

(b) $x_2[n] = \left[\overset{\frown}{0}, 1, -2,3, -4\right], h_2[n] = \left[0.5,1,\overset{\frown}{2},1,0.5\right].$

(c) $x_3[n] = \left[\overset{\frown}{1}, 2,3,4\right], h_3[n] = \left[\overset{\frown}{4}, 3,2,1\right].$

(d) $x_4[n] = \left[\overset{\frown}{1}, 2,3,4\right], h_4[n] = \left[\overset{\frown}{1}, 2,3,4\right].$

Hint: that conv command has no timing information so you must set start and end of x , h and the output to plot it correctly