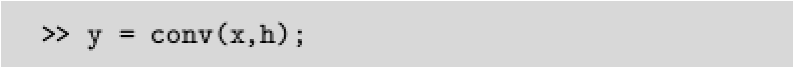
Session 4

CONVOLUTION

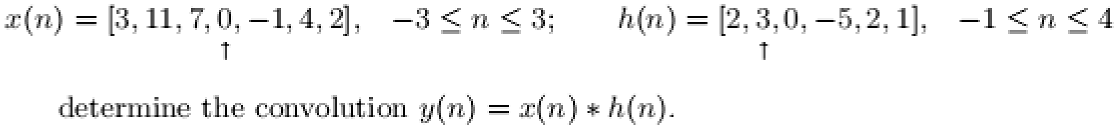
We introduced the convolution operation to describe the response of an LTI system.

If arbitrary sequences are of infinite duration, then MATLAB cannot be used directly to compute the convolution. MATLAB does provide a built-in function called conv that computes the convolution between two finite-duration sequences. The conv function assumes that the two sequences begin at n=0and is invoked by



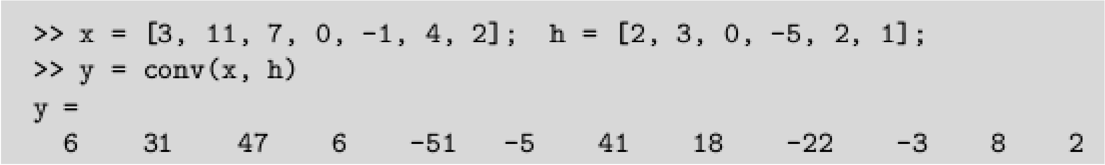
```
>> y = conv(x,h);
```

Example:



$$x(n) = [3, 11, 7, \underset{\uparrow}{0}, -1, 4, 2], \quad -3 \le n \le 3; \qquad h(n) = [2, \underset{\uparrow}{3}, 0, -5, 2, 1], \quad -1 \le n \le 4$$

determine the convolution $y(n) = x(n) * h(n)$.



```
>> x = [3, 11, 7, 0, -1, 4, 2];  h = [2, 3, 0, -5, 2, 1];
>> y = conv(x, h)
y =
   6    31    47     6   -51    -5    41    18   -22    -3     8     2
```

obtain the correct y(n)values. However, the conv function neither

provides nor accepts any timing information if the sequences have arbitrary support. What is needed is a beginning point and an end point of

y(n). Given finite duration x(n) and h(n), it is easy to determine these

points.

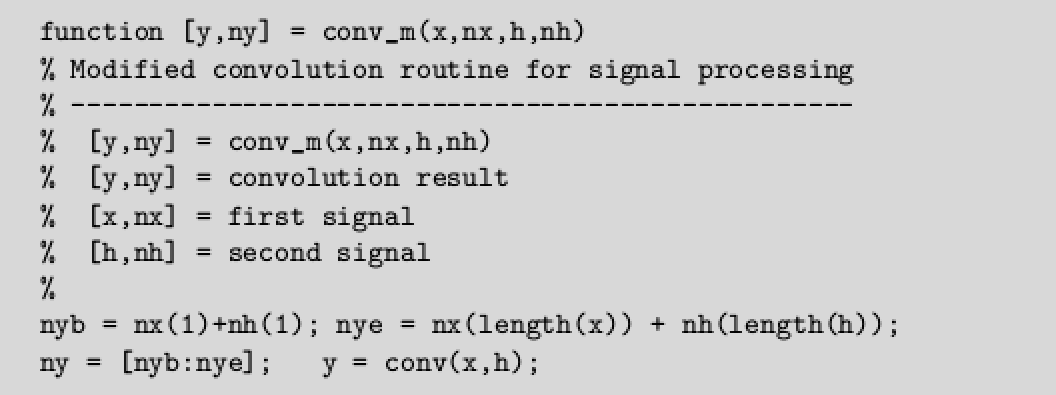
Let

$$\{x(n);\ n_{xb} \leq n \leq n_{xe}\} \qquad \text{and} \qquad \{h(n);\ n_{hb} \leq n \leq n_{he}\}$$

Be two finite-duration sequences. Then the beginning and end points of y(n) are

$$n_{yb} = n_{xb} + n_{hb} \quad \text{and} \quad n_{ye} = n_{xe} + n_{he}$$

A simple modification of the conv function, called conv_m, which performs the convolution of arbitrary support sequences can now be designed.



```
function [y,ny] = conv_m(x,nx,h,nh)
% Modified convolution routine for signal processing
% --------------------------------------------------
%  [y,ny] = conv_m(x,nx,h,nh)
%  [y,ny] = convolution result
%  [x,nx] = first signal
%  [h,nh] = second signal
%
nyb = nx(1)+nh(1); nye = nx(length(x)) + nh(length(h));
ny = [nyb:nye];   y = conv(x,h);
```