Systems

Prof. Seungchul Lee
Industrial AI Lab.

Most slides from
(edX) Discrete Time Signals and Systems by Prof. Richard G. Baraniuk

Systems

- A discrete-time system H is a transformation (a rule or formula) that maps a discrete-time input signal x into a discrete-time output signal y

$$x \longrightarrow \boxed{H} \longrightarrow y$$

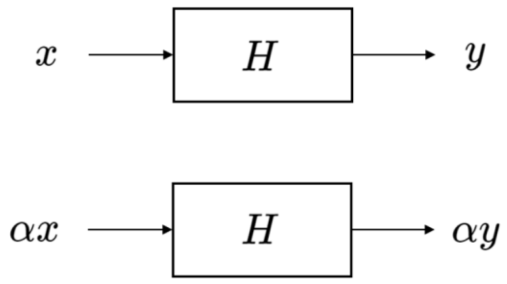
Example: Systems

Identity	$y[n] = x[n]$	$\forall n$
Scaling	$y[n] = 2\,x[n]$	$\forall n$
Offset	$y[n] = x[n]\ +\ 2$	$\forall n$
Square signal	$y[n] = (x[n])^2$	$\forall n$
Shift	$y[n] = x[n+2]$	$\forall n$
Decimate	$y[n] = x[2n]$	$\forall n$
Square time	$y[n] = x[n^2]$	$\forall n$

Linear Systems

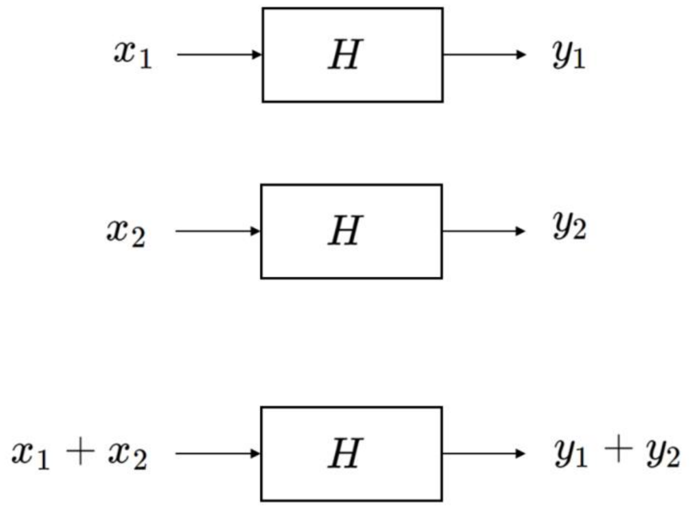
- A system H is linear if it satisfies the following two properties:
  - Scaling:

$$H\{\alpha x\} = \alpha H\{x\} \quad \forall\, \alpha \in \mathbb{C}$$



  - Additivity:

If $y_1 = H\{x_1\}$ and $y_2 = H\{x_2\}$ then $H\{x_1 + x_2\} = y_1 + y_2$



Linear Systems and Matrix Multiplication

- Matrix multiplication (aka linear combination) is a fundamental signal processing system
- Matrix multiplications are linear systems

$$y = Hx$$

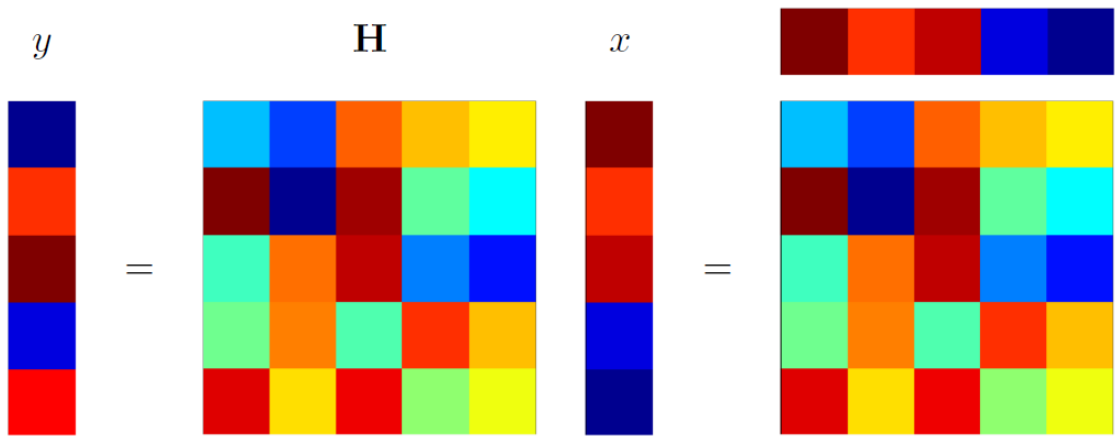
$$y[n] = \sum_m [H]_{n,m} x[m] = \sum_m h_{n,m} x[m]$$

where $h_{n,m} = [H]_{n,m}$ represents the row-n, column-m entry of the matrix H

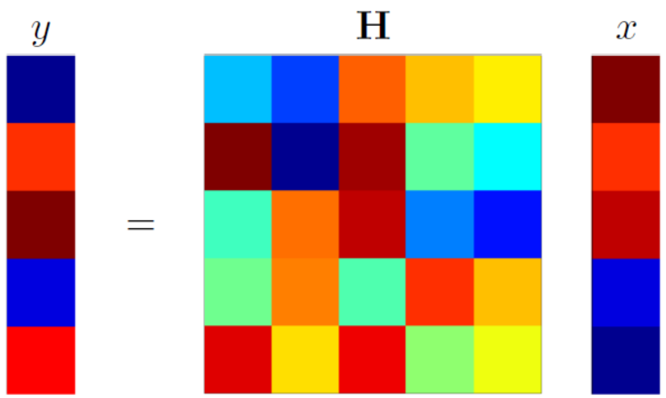
- All linear systems can be expressed as matrix multiplications

Matrix Multiplication and Linear Systems in Pictures

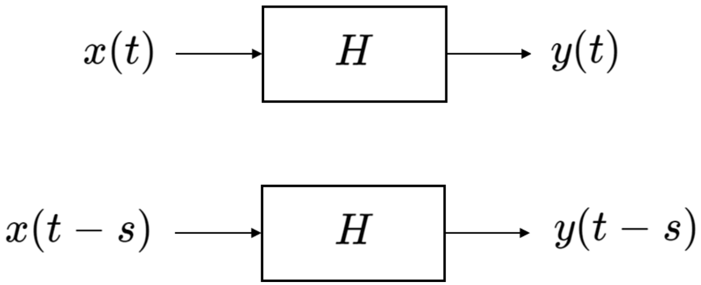
- System output as a linear combination of columns



- System output as a sequence of inner products



Time-Invariant Systems



- For infinite-length signals
  - A system H processing infinite-length signals is time-invariant (shift-invariant) if a time shift of the input signal creates a corresponding time shift in the output signal

- For finite-length signals
  - A system H processing infinite-length signals is time-invariant (shift-invariant) if a circular time shift of the input signal creates a corresponding circular time shift in the output signal

Linear Time-Invariant (LTI) Systems

- A system H is linear time-invariant (LTI) if it is both linear and time-invariant
- We will only consider Linear Time-Invariant (LTI) systems.

Identity	$y[n] = x[n]$	$\forall n$	Linear	Time Invariant
Scaling	$y[n] = 2\,x[n]$	$\forall n$	Linear	Time Invariant
Offset	$y[n] = x[n]\ +\ 2$	$\forall n$	Non Linear	Time Invariant
Square signal	$y[n] = (x[n])^2$	$\forall n$	Non Linear	Time Invariant
Shift	$y[n] = x[n+2]$	$\forall n$	Linear	Time Invariant
Decimate	$y[n] = x[2n]$	$\forall n$	Linear	Time Variant
Square time	$y[n] = x[n^2]$	$\forall n$	Linear	Time Variant

Matrix Multiplication and LTI Systems (Infinite-Length Signals)

- When the linear system is also shift invariant, H has a special structure
- Linear system for infinite-length signals can be expressed as

$$y[n] = H\{x[n]\} = \sum_{m=-\infty}^{\infty} h_{n,m} x[m], \quad -\infty < n < \infty$$

- Enforcing time invariance implies that for all $q \in \mathbb{Z}$

$$H\{x[n-q]\} = \sum_{m=-\infty}^{\infty} h_{n,m} x[m-q] = y[n-q]$$

- Change of variables: $n' = n - q$ and $m' = m - q$

$$H\{x[n']\} = \sum_{m=-\infty}^{\infty} h_{n'+q,m'+q} x[m'] = y[n']$$

Matrix Multiplication and LTI Systems (Infinite-Length Signals)

$$y[n] = H\{x[n]\} = \sum_{m=-\infty}^{\infty} h_{n,m} x[m], \quad -\infty < n < \infty$$

$$H\{x[n']\} = \sum_{m=-\infty}^{\infty} h_{n'+q,m'+q} x[m'] = y[n']$$

- We see that for an LTI system

$$h_{n,m} = h_{n+q,m+q}$$

$$H = \begin{bmatrix} & \vdots & \vdots & \vdots & \\ \cdots & h_{-1,-1} & h_{-1,0} & h_{-1,1} & \cdots \\ \cdots & h_{0,-1} & h_{0,0} & h_{0,1} & \cdots \\ \cdots & h_{1,-1} & h_{1,0} & h_{1,1} & \cdots \\ & \vdots & \vdots & \vdots & \end{bmatrix} = \begin{bmatrix} & \vdots & \vdots & \vdots & \\ \cdots & h_{0,0} & h_{-1,0} & h_{-2,0} & \cdots \\ \cdots & h_{1,0} & h_{0,0} & h_{-1,0} & \cdots \\ \cdots & h_{2,0} & h_{1,0} & h_{0,0} & \cdots \\ & \vdots & \vdots & \vdots & \end{bmatrix}$$

- Entries on the matrix diagonals are the same - Toeplitz matrix

Matrix Multiplication and LTI Systems (Infinite-Length Signals)

- All of the entries in a Toeplitz matrix can be expressed in terms of the entries of the
  - 0-th column

$$h[n] = h_{n,0}$$

  - Time-reversed 0-th row

$$h[m] = h_{0,-m}$$

$$\begin{bmatrix} & \vdots & \vdots & \vdots & \\ \cdots & h_{0,0} & h_{-1,0} & h_{-2,0} & \cdots \\ \cdots & h_{1,0} & h_{0,0} & h_{-1,0} & \cdots \\ \cdots & h_{2,0} & h_{1,0} & h_{0,0} & \cdots \\ & \vdots & \vdots & \vdots & \end{bmatrix} = \begin{bmatrix} & \vdots & \vdots & \vdots & \\ \cdots & h[0] & h[-1] & h[-2] & \cdots \\ \cdots & h[1] & h[0] & h[-1] & \cdots \\ \cdots & h[2] & h[1] & h[0] & \cdots \\ & \vdots & \vdots & \vdots & \end{bmatrix}$$

- Row-n, column-m entry of the matrix

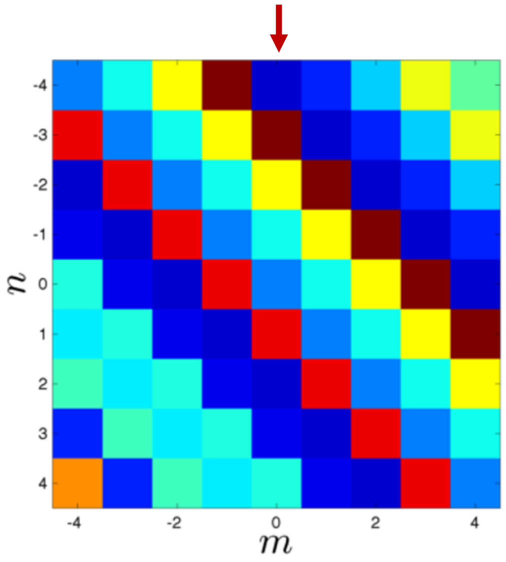
$$[H]_{n,m} = h_{n,m} = h[n-m]$$

Matrix Multiplication and LTI Systems (Infinite-Length Signals)

- Note the diagonals !

$$\begin{bmatrix} & \vdots & \vdots & \vdots & \\ \cdots & h_{0,0} & h_{-1,0} & h_{-2,0} & \cdots \\ \cdots & h_{1,0} & h_{0,0} & h_{-1,0} & \cdots \\ \cdots & h_{2,0} & h_{1,0} & h_{0,0} & \cdots \\ & \vdots & \vdots & \vdots & \end{bmatrix} = \begin{bmatrix} & \vdots & \vdots & \vdots & \\ \cdots & h[0] & h[-1] & h[-2] & \cdots \\ \cdots & h[1] & h[0] & h[-1] & \cdots \\ \cdots & h[2] & h[1] & h[0] & \cdots \\ & \vdots & \vdots & \vdots & \end{bmatrix}$$

$\mathbf{H} =$



$h =$

Matrix Multiplication and LTI Systems (Finite-Length Signals)

- Linear system for signals of length N can be expressed as

$$y[n] = H\{x[n]\} = \sum_{m=0}^{N} h_{n,m} x[m], \quad 0 \le n \le N-1$$

- Enforcing time invariance implies that for all $q \in \mathbb{Z}$

$$H\{x[(n-q)_N]\} = \sum_{m=0}^{N-1} h_{n,m} x[(m-q)_N] = y[(n-q)_N]$$

- Change of variables: $n' = n - q$ and $m' = m - q$

$$H\{x[(n')_N]\} = \sum_{m=-q}^{N-1-q} h_{(n'+q)_N,(m'+q)_N} x[(m')_N] = y[(n')_N]$$

Matrix Multiplication and LTI Systems (Finite-Length Signals)

- We see that for an LTI system

$$h_{n,m} = h_{(n+q)_N,(m+q)_N}$$

$$H = \begin{bmatrix} h_{0,0} & h_{0,1} & h_{0,2} & \cdots & h_{0,N-1} \\ h_{1,0} & h_{1,1} & h_{1,2} & \cdots & h_{1,N-1} \\ h_{2,0} & h_{2,1} & h_{2,2} & \cdots & h_{2,N-1} \\ \vdots & \vdots & \vdots & & \vdots \\ h_{N-1,0} & h_{N-1,1} & h_{N-1,2} & \cdots & h_{N-1,N-1} \end{bmatrix} = \begin{bmatrix} h_{0,0} & h_{N-1,0} & h_{N-2,0} & \cdots & h_{1,0} \\ h_{1,0} & h_{0,0} & h_{N-1,0} & \cdots & h_{2,0} \\ h_{2,0} & h_{1,0} & h_{0,0} & \cdots & h_{3,0} \\ \vdots & \vdots & \vdots & & \vdots \\ h_{N-1,0} & h_{N-2,0} & h_{N-3,0} & \cdots & h_{0,0} \end{bmatrix}$$

- Entries on the matrix diagonals are the same + circular wraparound - Circulent matrix

Matrix Multiplication and LTI Systems (Finite-Length Signals)

- All of the entries in a circulent matrix can be expressed in terms of the entries of the
  - 0-th column

$$h[n] = h_{n,0}$$

  - Time-reversed 0-th row

$$h[m] = h_{0,(-m)_N}$$

$$\begin{bmatrix} h_{0,0} & h_{N-1,0} & h_{N-2,0} & \cdots & h_{1,0} \\ h_{1,0} & h_{0,0} & h_{N-1,0} & \cdots & h_{2,0} \\ h_{2,0} & h_{1,0} & h_{0,0} & \cdots & h_{3,0} \\ \vdots & \vdots & \vdots & & \vdots \\ h_{N-1,0} & h_{N-2,0} & h_{N-3,0} & \cdots & h_{0,0} \end{bmatrix} = \begin{bmatrix} h[0] & h[N-1] & h[N-2] & \cdots & h[1] \\ h[1] & h[0] & h[N-1] & \cdots & h[2] \\ h[2] & h[1] & h[0] & \cdots & h[3] \\ \vdots & \vdots & \vdots & & \vdots \\ h[N-1] & h[N-2] & h[N-3] & \cdots & h[0] \end{bmatrix}$$

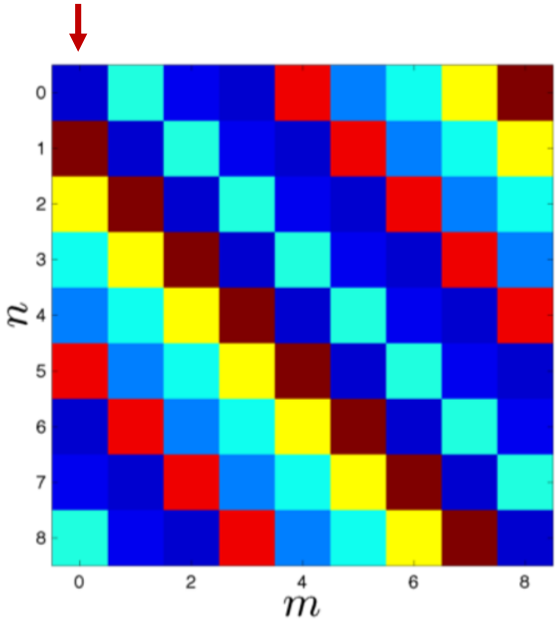
- Row-n, column-m entry of the matrix

$$[H]_{n,m} = h_{n,m} = h[(n-m)_N]$$

Matrix Multiplication and LTI Systems (Finite-Length Signals)

- Note the diagonals and circulent shifts !

$$\begin{bmatrix} h_{0,0} & h_{N-1,0} & h_{N-2,0} & \cdots & h_{1,0} \\ h_{1,0} & h_{0,0} & h_{N-1,0} & \cdots & h_{2,0} \\ h_{2,0} & h_{1,0} & h_{0,0} & \cdots & h_{3,0} \\ \vdots & \vdots & \vdots & & \vdots \\ h_{N-1,0} & h_{N-2,0} & h_{N-3,0} & \cdots & h_{0,0} \end{bmatrix} = \begin{bmatrix} h[0] & h[N-1] & h[N-2] & \cdots & h[1] \\ h[1] & h[0] & h[N-1] & \cdots & h[2] \\ h[2] & h[1] & h[0] & \cdots & h[3] \\ \vdots & \vdots & \vdots & & \vdots \\ h[N-1] & h[N-2] & h[N-3] & \cdots & h[0] \end{bmatrix}$$

$\mathbf{H} =$ 

$h =$ 

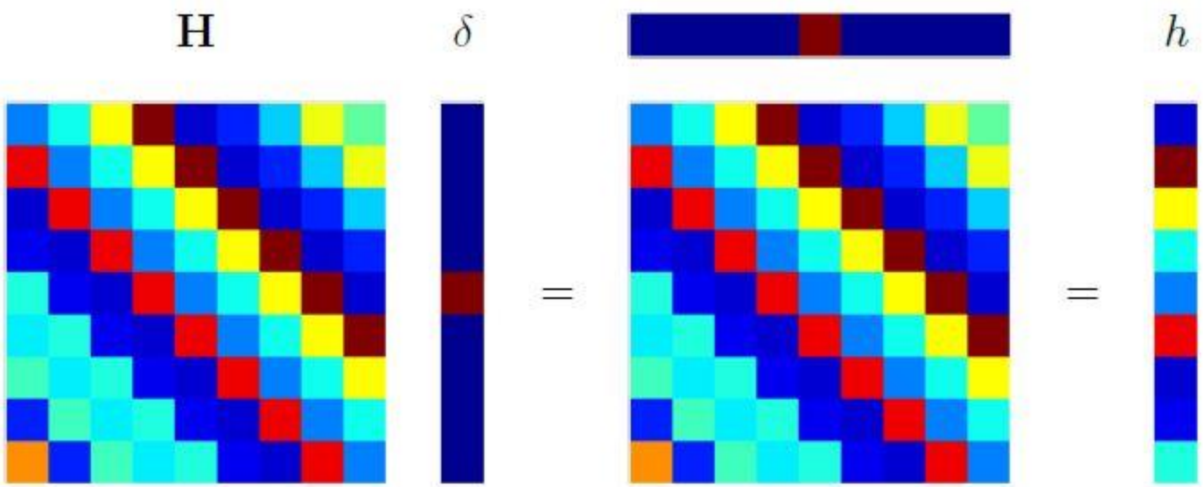
Impulse Response

Impulse Response

- We will Illustrate it with an infinite-length signal

- The 0-th column of the matrix H has a special interpretation
- Compute the output when the input is a delta function (impulse)

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & \text{otherwise} \end{cases}$$



- This suggests that we call h the impulse response of the system
- Output of system to delta function (impulse) is h. So, We call h the impulse response of the system

Impulse Response

- From h, we can build matrix H
  - Columns/rows of H are the shifted versions of the 0-th column/row
  - h contains all the information of the LTI system

- LTI systems are Toeplitz matrices (infinite-length signals)
  - Entries on the matrix diagonals are the same

$$y[n] = \sum_{-\infty}^{\infty} h[n-m]x[m]$$

- Let the input $\delta[n]$ and compute $y[n]$

$$\sum_{-\infty}^{\infty} h[n-m]\delta[m] = h[n]$$

- The impulse response characterizes an LTI system (that is, carries all of the information contained in matrix H)

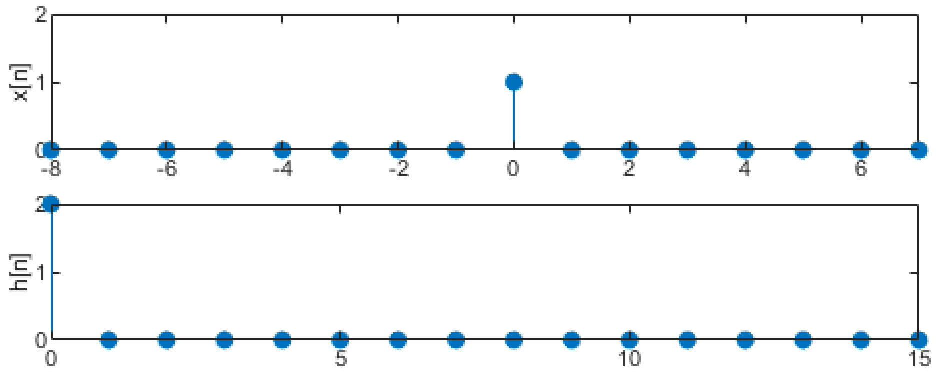
Examples (Infinite-Length Signals)

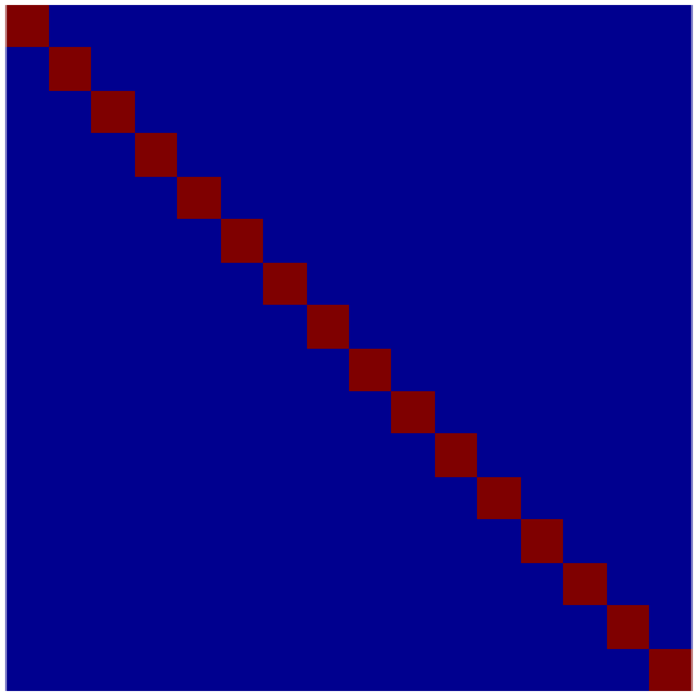
- Impulse response of the scaling system

$$H = \begin{bmatrix} & \vdots & \vdots & \vdots & \\ \cdots & h[0] & h[-1] & h[-2] & \cdots \\ \cdots & h[1] & h[0] & h[-1] & \cdots \\ \cdots & h[2] & h[1] & h[0] & \cdots \\ & \vdots & \vdots & \vdots & \end{bmatrix}$$

$$h[n] = 2\delta[n]$$

$$H_{n,m} = h[n-m]$$



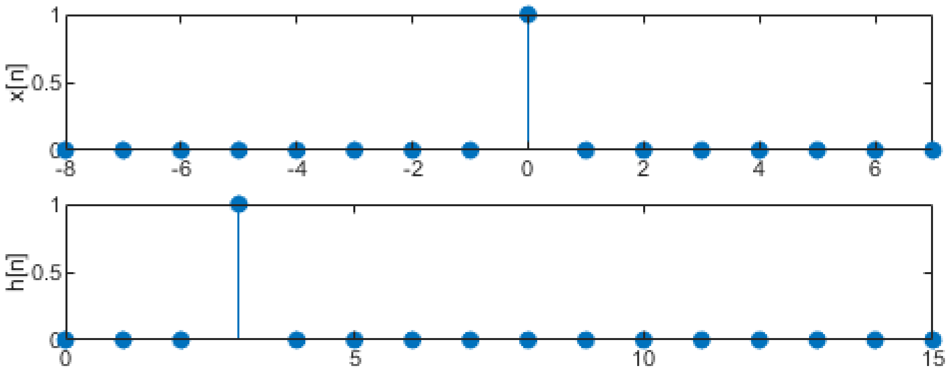


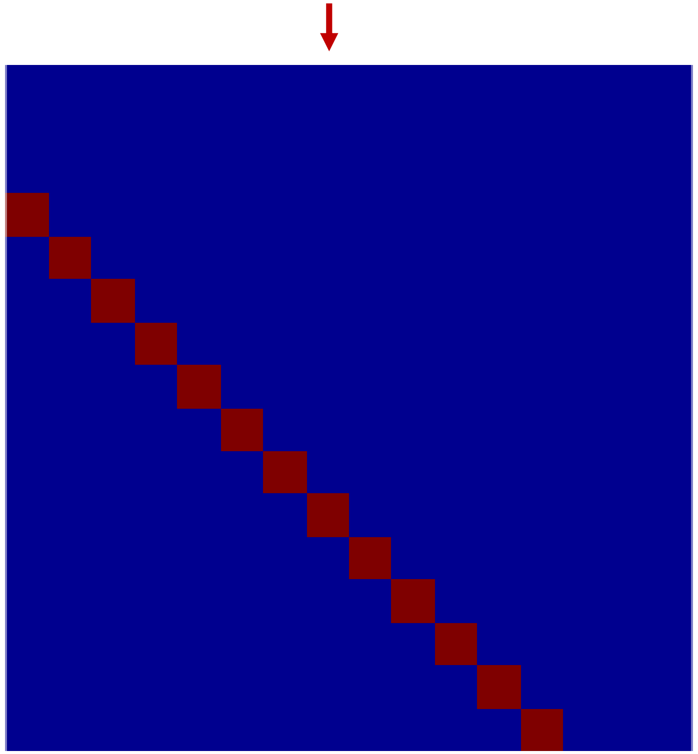
Examples (Infinite-Length Signals)

- Impulse response of the shift system

$$h[n] = \delta[n-3]$$

$$H_{n,m} = h[n-m]$$



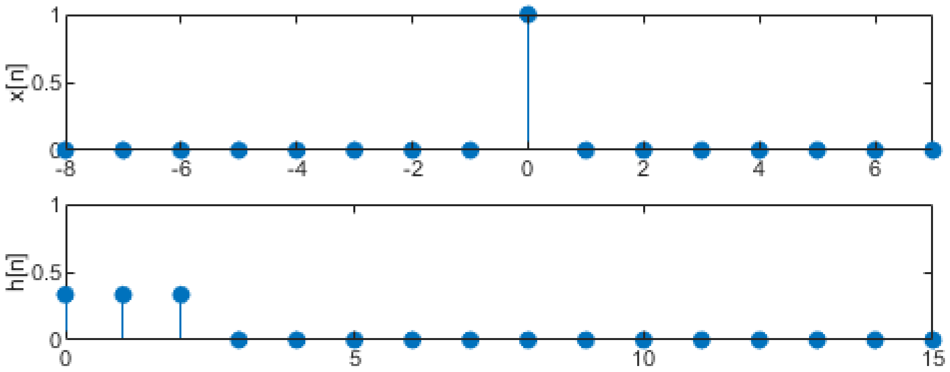


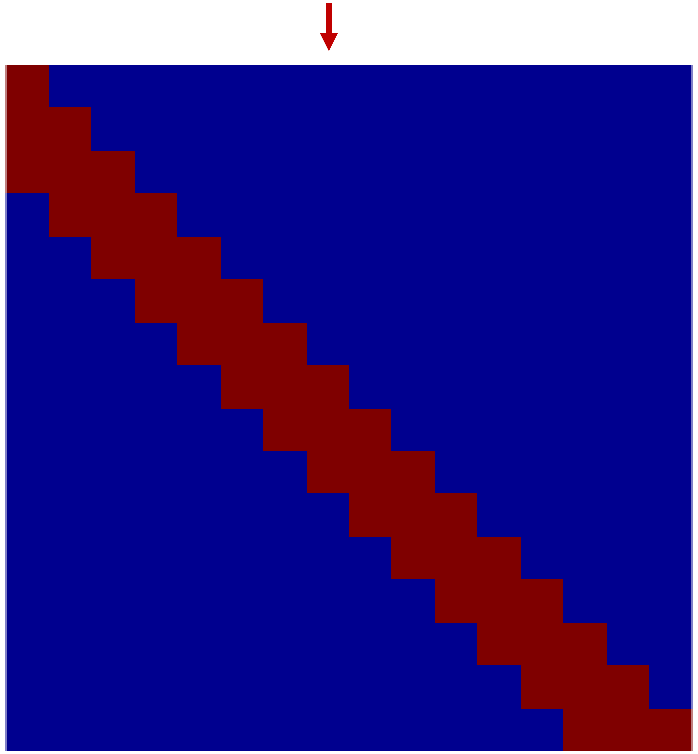
Examples (Infinite-Length Signals)

- Impulse response of the moving average system

$$h[n] = \frac{1}{3}(\delta[n] + \delta[n-1] + \delta[n-2])$$

$$H_{n,m} = h[n-m] = \frac{1}{3}(\delta[n-m] + \delta[n-m-1] + \delta[n-m-2])$$





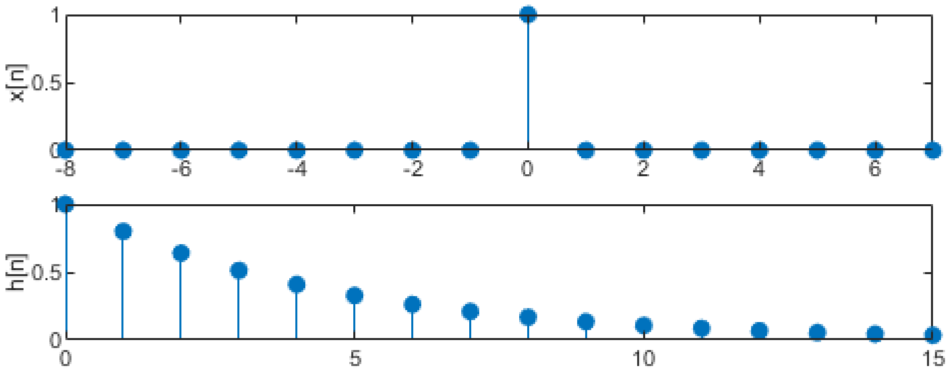
Examples (Infinite-Length Signals)

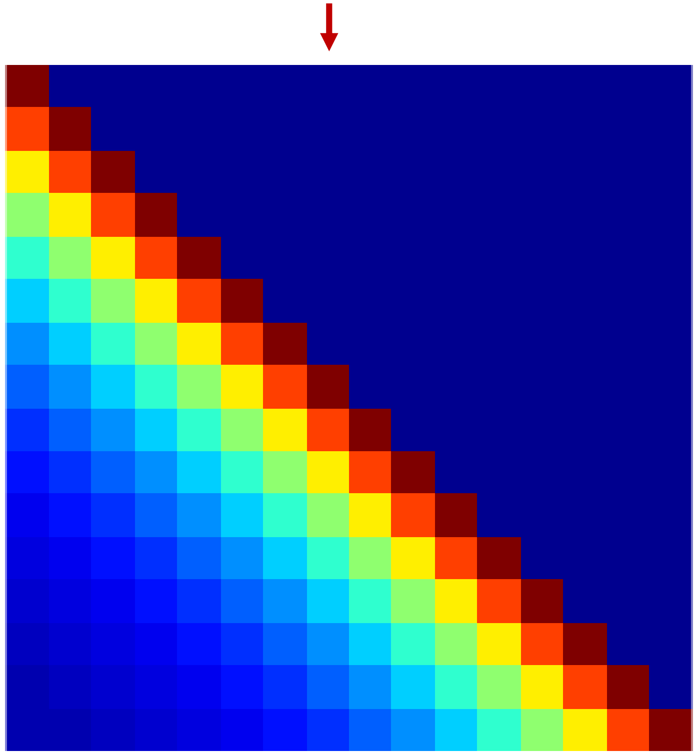
- Impulse response of the recursive average system

$$y[n] = x[n] + \alpha y[n-1]$$

$$h[n] = \alpha^n u[n]$$

$$H_{n,m} = h[n-m] = \alpha^{n-m} u[n-m]$$

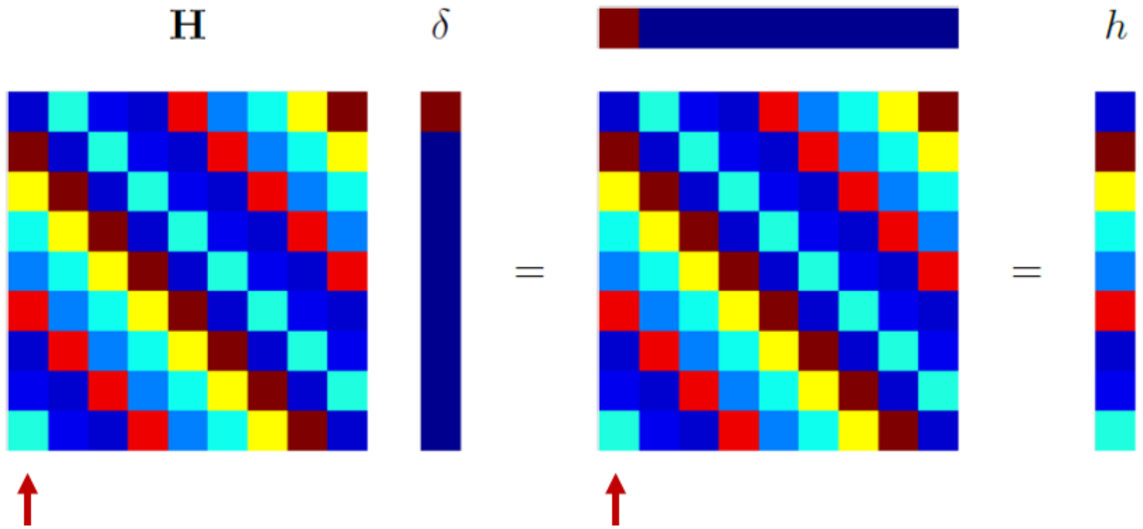




Examples (Finite-Length Signals)

- Entries on the matrix diagonals are the same + circular wraparound

$$H = \begin{bmatrix} h[0] & h[N-1] & h[N-2] & \cdots & h[1] \\ h[1] & h[0] & h[N-1] & \cdots & h[2] \\ h[2] & h[1] & h[0] & \cdots & h[3] \\ \vdots & \vdots & \vdots & & \vdots \\ h[N-1] & h[N-2] & h[N-3] & \cdots & h[0] \end{bmatrix}$$

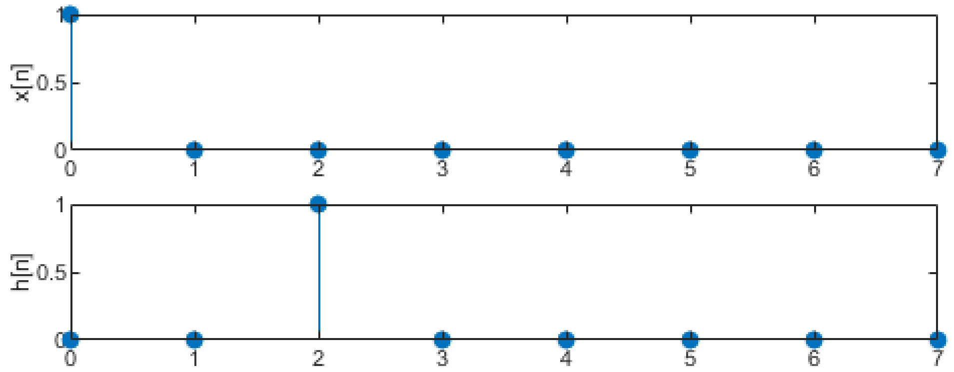


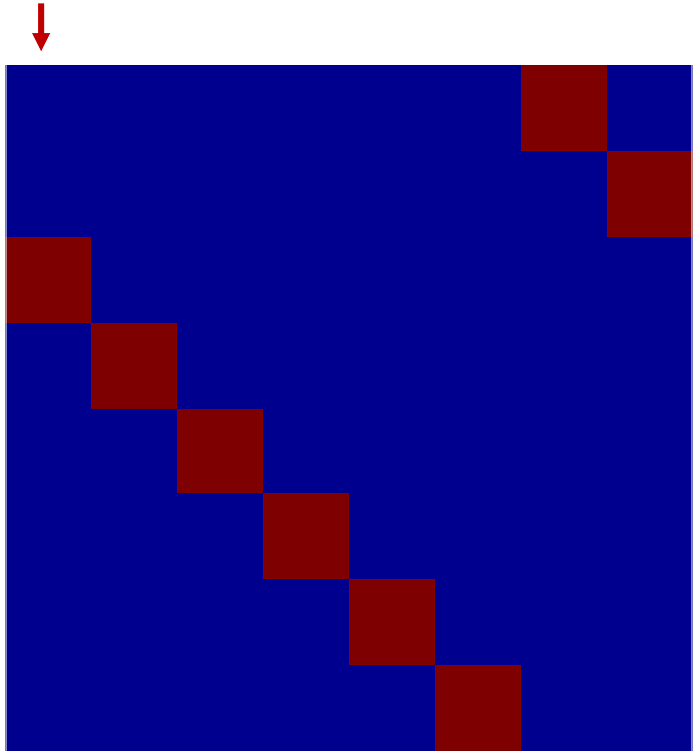
Examples (Finite-Length Signals)

- Impulse response of the shift system

$$h[n] = \delta[n-2]$$

$$H_{n,m} = h[(n-m)_N]$$



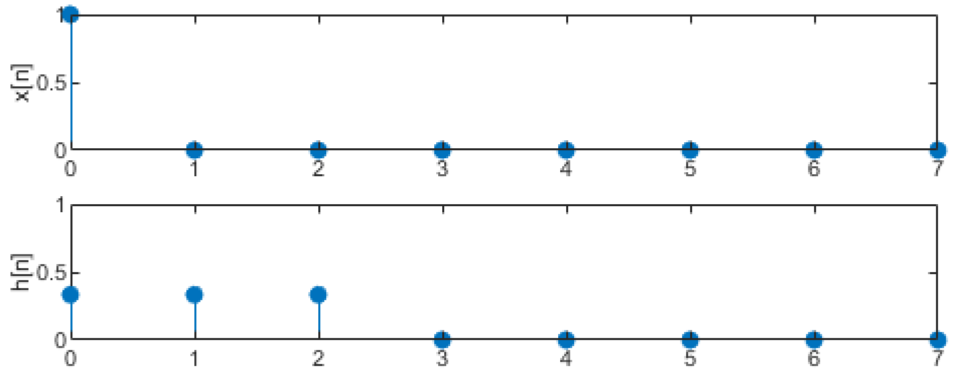


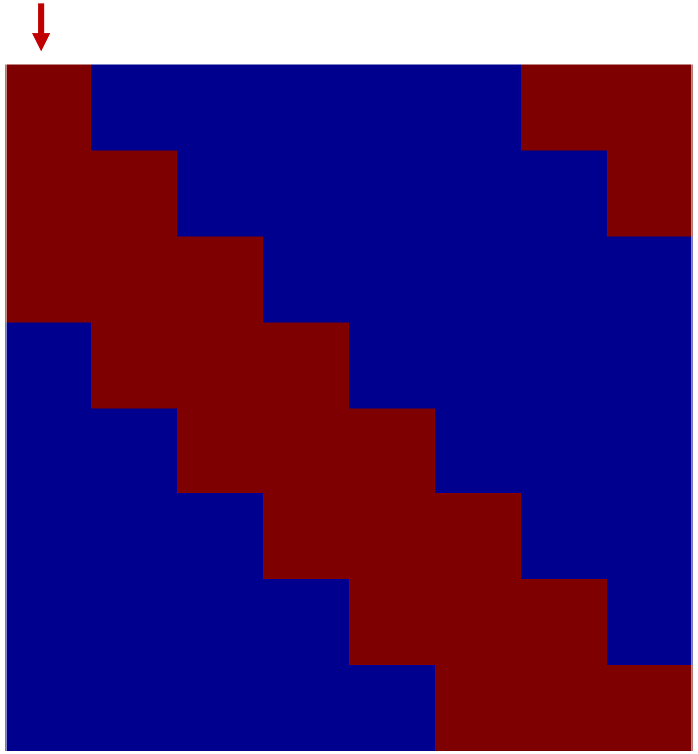
Examples (Finite-Length Signals)

- Impulse response of the moving average system

$$h[n] = \frac{1}{3}(\delta[n] + \delta[n-1] + \delta[n-2])$$

$$H_{n,m} = h[(n-m)_N]$$





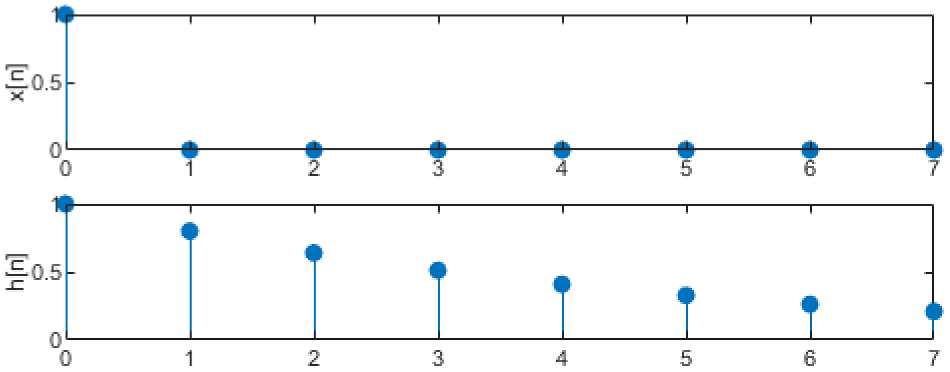
Examples (Finite-Length Signals)

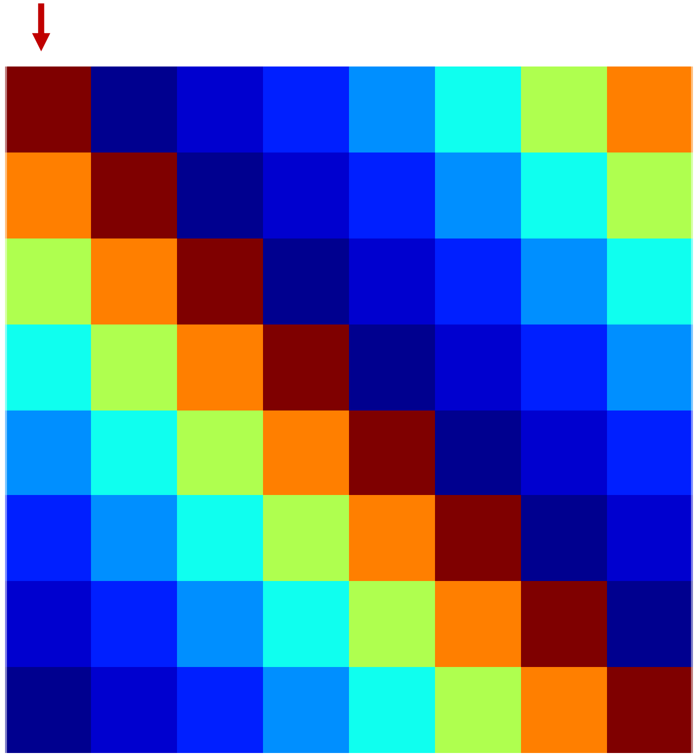
- Impulse response of the recursive average system

$$y[n] = x[n] + \alpha y[n-1]$$

$$h[n] = \alpha^n u[n]$$

$$H_{n,m} = h[(n-m)_N]$$





Time Response to Arbitrary Input: Convolution

Convolution

- Convolution is defined as the integral of the product of the two signals after one is reversed and shifted

- Output $y[n]$ came out by convolution of input $x[n]$ and system $h[n]$
  - Time reverse the impulse response h and shift it n time steps to the right (delay)
  - Compute the inner product between the shifted impulse response and the input vector x

$$\begin{bmatrix} & \vdots & \vdots & \vdots & \\ \cdots & h[0] & h[-1] & h[-2] & \cdots \\ \cdots & h[1] & h[0] & h[-1] & \cdots \\ \cdots & h[2] & h[1] & h[0] & \cdots \\ & \vdots & \vdots & \vdots & \end{bmatrix}$$

$$y = Hx$$

$$y[n] = \sum_{m=-\infty}^{\infty} h[n-m]\, x[m] = x[n] * h[n]$$

Convolution For Infinite-Length Signals

$$\begin{bmatrix} & \vdots & \vdots & \vdots & \\ \cdots & h[0] & h[-1] & h[-2] & \cdots \\ \cdots & h[1] & h[0] & h[-1] & \cdots \\ \cdots & h[2] & h[1] & h[0] & \cdots \\ & \vdots & \vdots & \vdots & \end{bmatrix}$$

- Toeplitz Matrices

$$y[n] = x[n] * h[n] = \sum_{m=-\infty}^{\infty} h[n-m]\, x[m]$$

$$= \cdots + \; h[n+2]\, x[-2] + h[n+1]\, x[-1] + h[n]\, x[0] + h[n-1]\, x[1] + h[n-2]\, x[2] \; + \cdots$$

- It is an inner product of h vectors and x

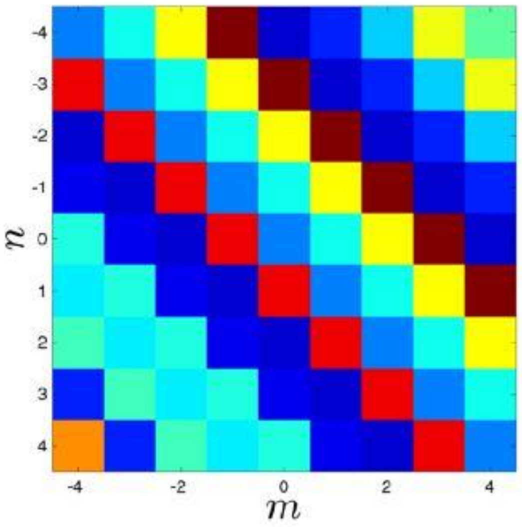
$$y[n] = \begin{bmatrix} \cdots & | & | & | & | & | & \cdots \\ \cdots & h[n+2] & h[n+1] & h[n] & h[n-1] & h[n-2] & \cdots \\ \cdots & | & | & | & | & | & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ x[-2] \\ x[-1] \\ x[0] \\ x[1] \\ x[2] \\ \vdots \end{bmatrix} = Hx$$

Convolution For Infinite-Length Signals

- Convolution is product of matrix H and x

$$y[n] = \begin{bmatrix} \cdots & | & | & | & | & | & \cdots \\ \cdots & h[n+2] & h[n+1] & h[n] & h[n-1] & h[n-2] & \cdots \\ \cdots & | & | & | & | & | & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ x[-2] \\ x[-1] \\ x[0] \\ x[1] \\ x[2] \\ \vdots \end{bmatrix}$$

$\mathbf{H} =$

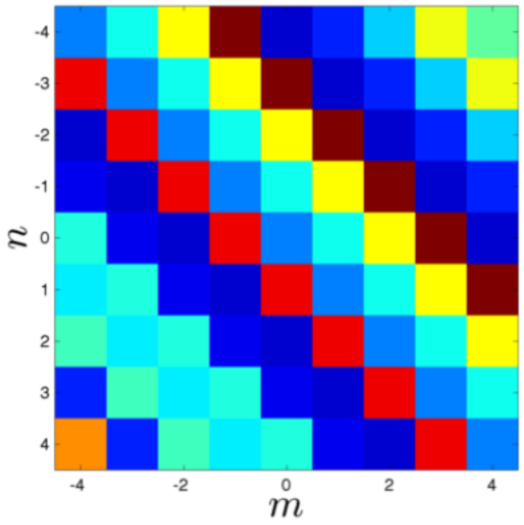


Convolution using Toeplitz Matrix

- LTI systems are Toeplitz matrices (infinite-length signals)
  - Entries on the matrix diagonals are the same

```
x = [3, 11, 7, 0, -1, 4, 2]';
h = [2, 3, 0, -5, 2, 1]';

hc = [h,zeros([length(x)-1,1])];
hr = [h(1),zeros([1,length(x)-1])];
H = toeplitz(hc,hr)

y = H*x
```

$\mathbf{H} =$

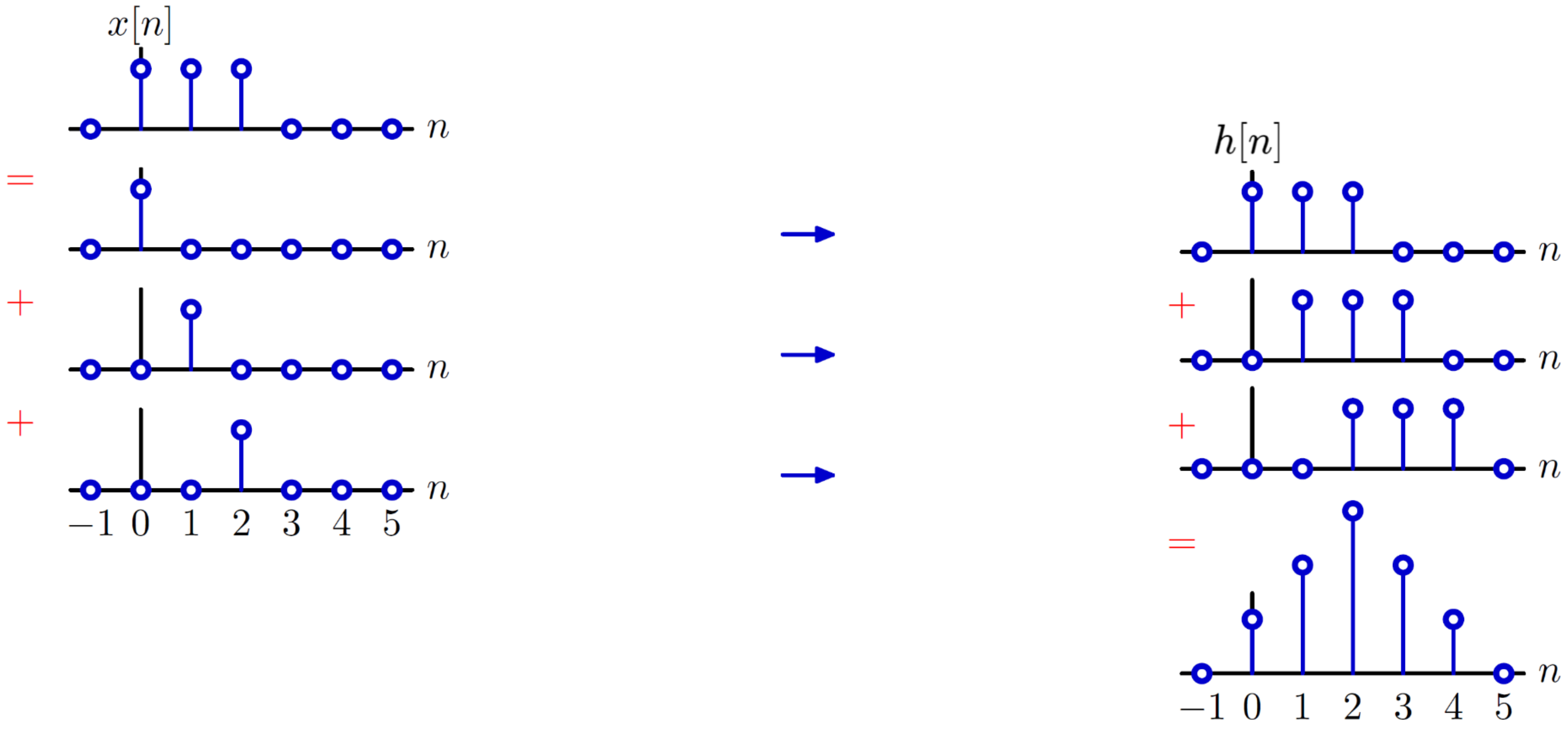


$h =$

Superposition (Linear) and Time-Invariant

$$y[n] = \sum_{m=-\infty}^{\infty} h[n-m]\, x[m] = x[n] * h[n]$$

- Think about convolution in time
  - Break input into additive parts and sum the responses to the parts

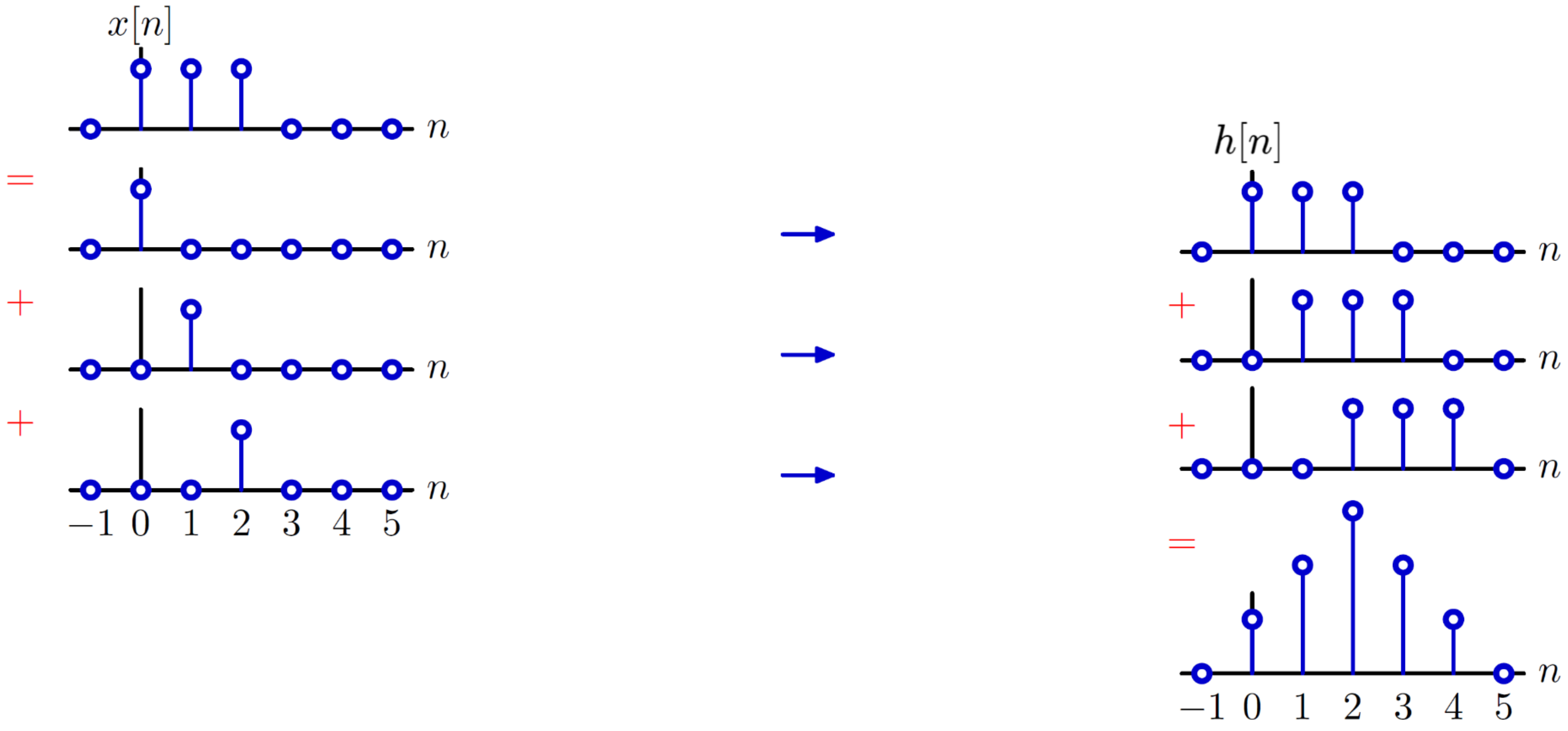


Source: Prof. Denny Freeman at MIT

Superposition (Linear) and Time-Invariant

$$y[n] = \sum_{m=-\infty}^{\infty} h[n-m]\, x[m] = x[n] * h[n]$$

- Think about convolution in time
  - You are standing at time n



Source: Prof. Denny Freeman at MIT

Convolution

- If a system is linear and time-invariant (LTI) then its output is the sum of weighted and shifted unit-sample responses.

$$\delta[n] \longrightarrow \boxed{\text{system}} \longrightarrow h[n]$$

$$\delta[n-k] \longrightarrow \boxed{\text{system}} \longrightarrow h[n-k]$$

$$x[k]\delta[n-k] \longrightarrow \boxed{\text{system}} \longrightarrow x[k]h[n-k]$$

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \longrightarrow \boxed{\text{system}} \longrightarrow y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

Source: Prof. Denny Freeman at MIT

1D Convolution

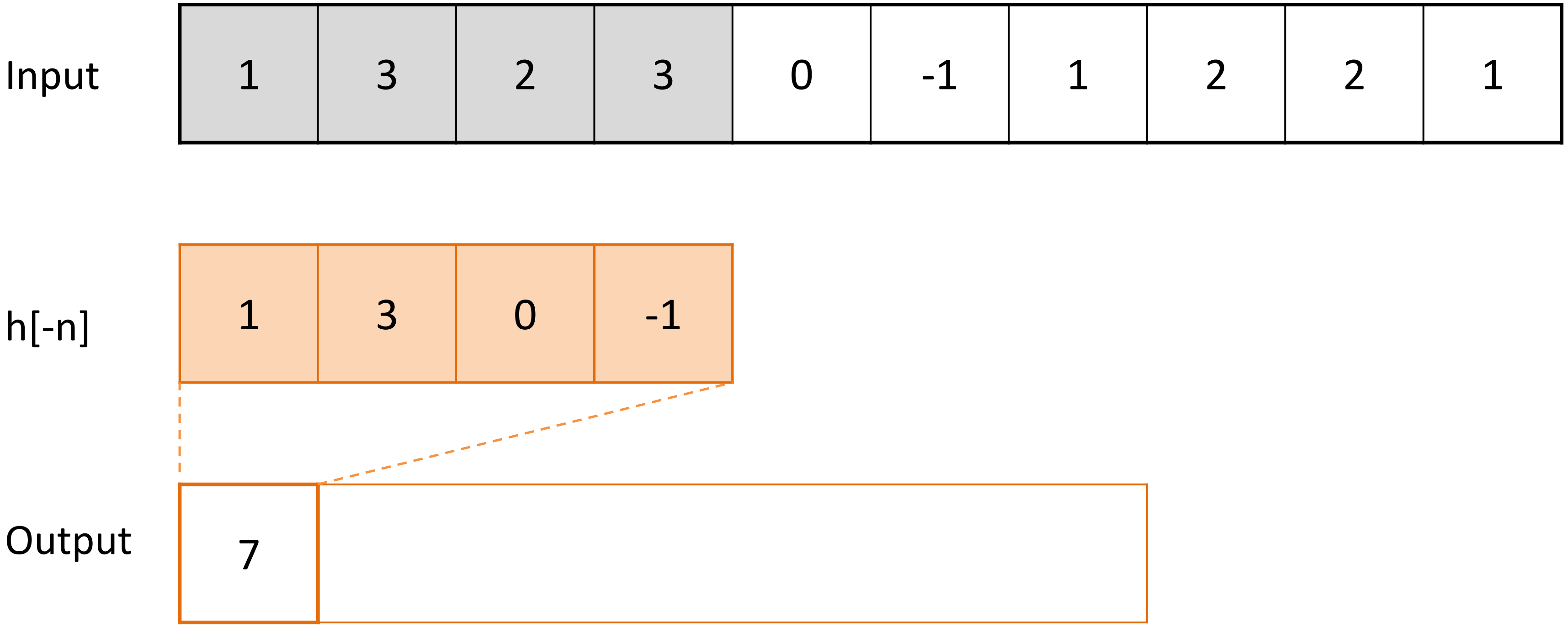
Input	1	3	2	3	0	-1	1	2	2	1

1D Convolution

Input	1	3	2	3	0	-1	1	2	2	1
h[-n]	1	3	0	-1						
Output										

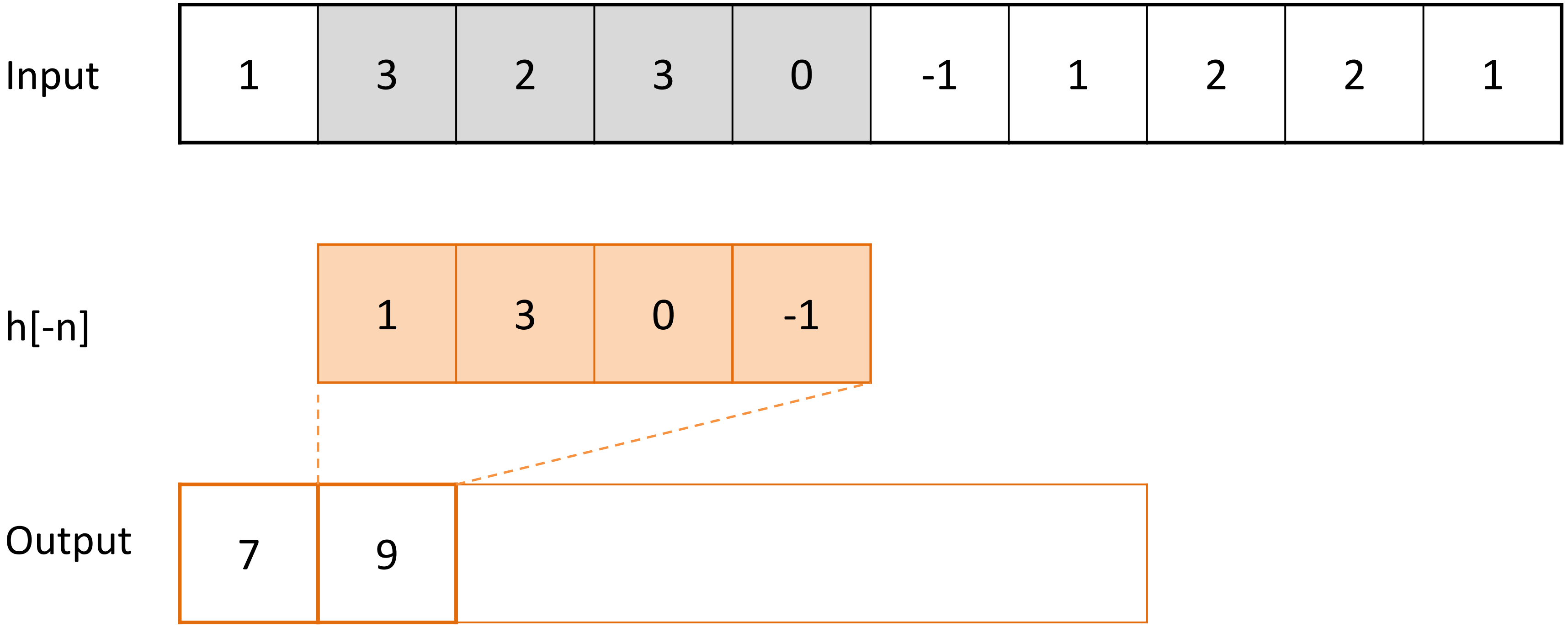
Source: Dr. Francois Fleuret at EPFL

1D Convolution

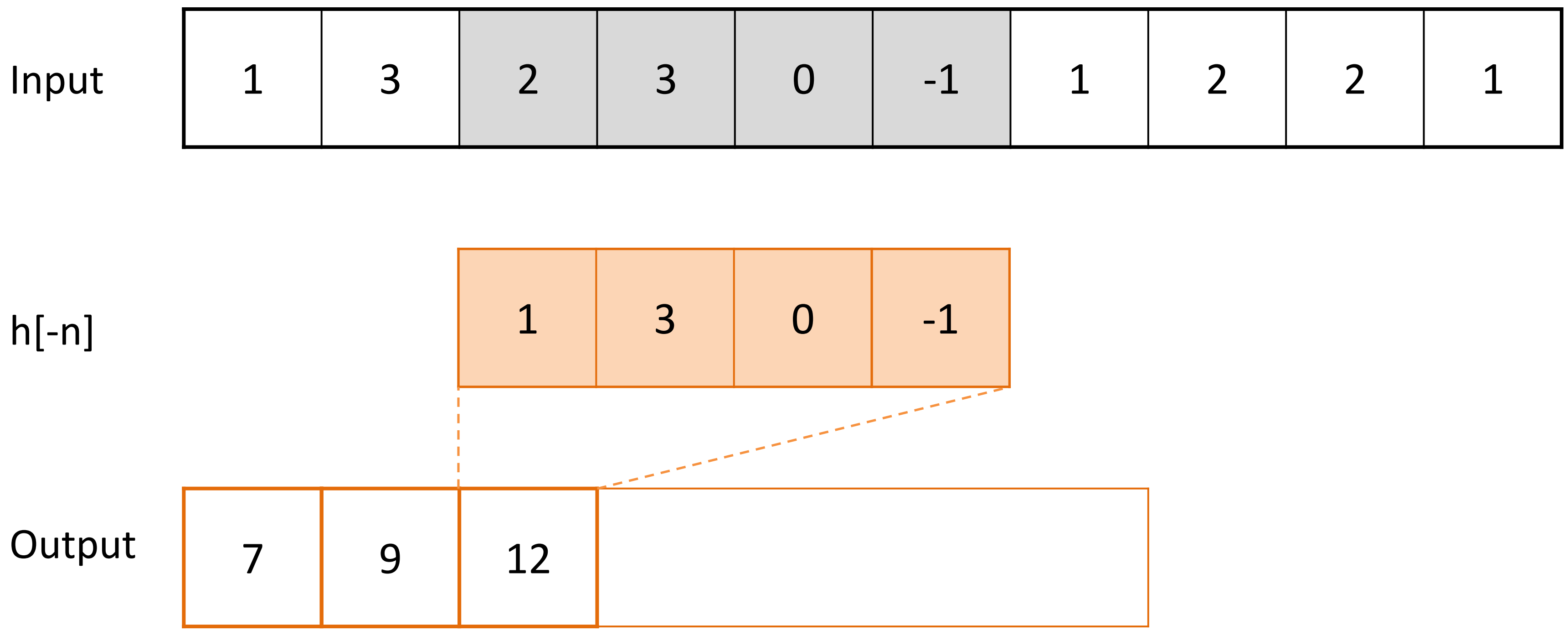


Source: Dr. Francois Fleuret at EPFL

1D Convolution



1D Convolution



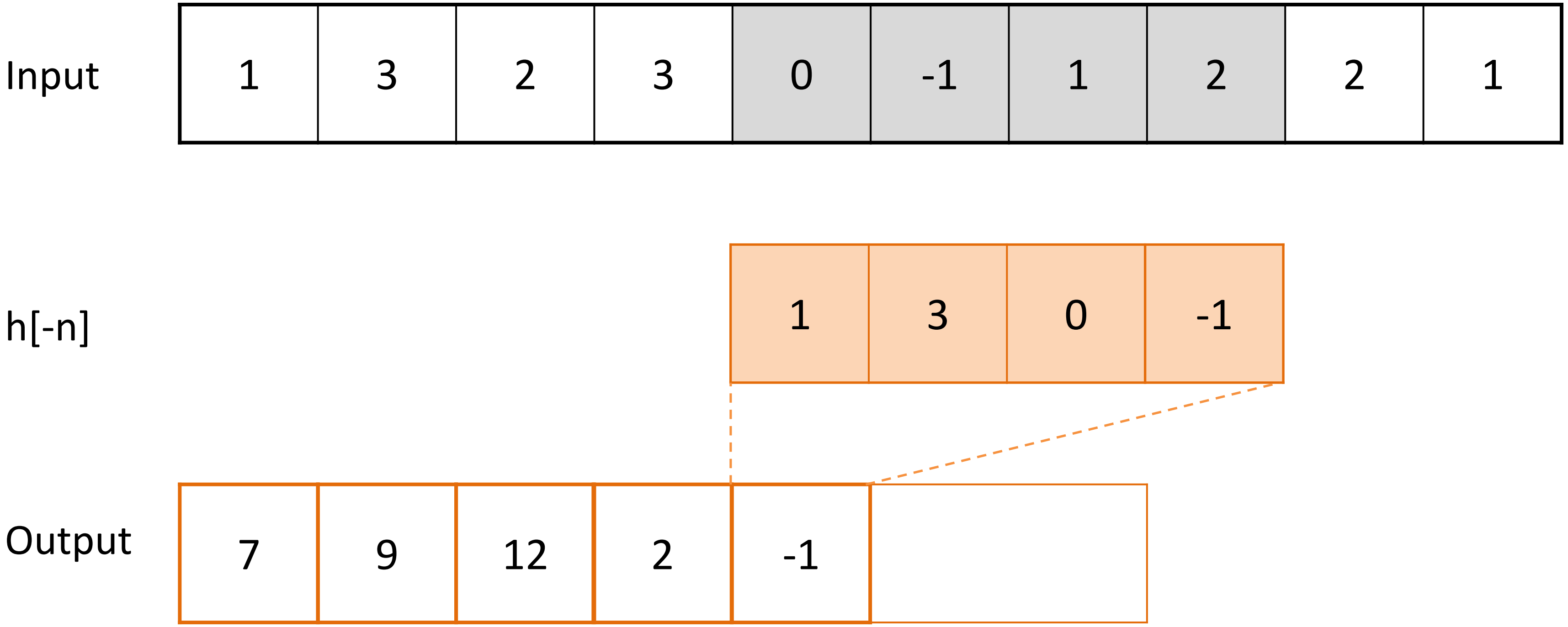
Source: Dr. Francois Fleuret at EPFL

1D Convolution

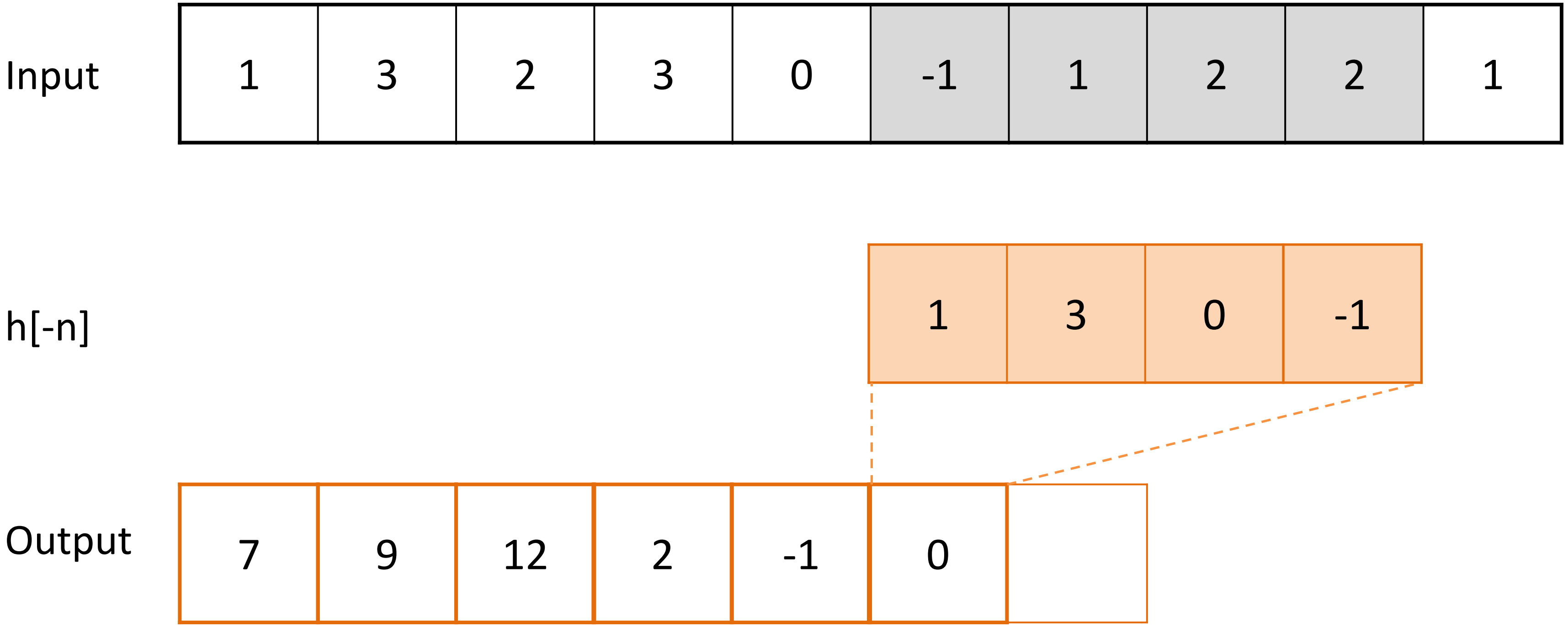
Input	1	3	2	3	0	-1	1	2	2	1
h[-n]				1	3	0	-1			
Output	7	9	12	2						

Source: Dr. Francois Fleuret at EPFL

1D Convolution

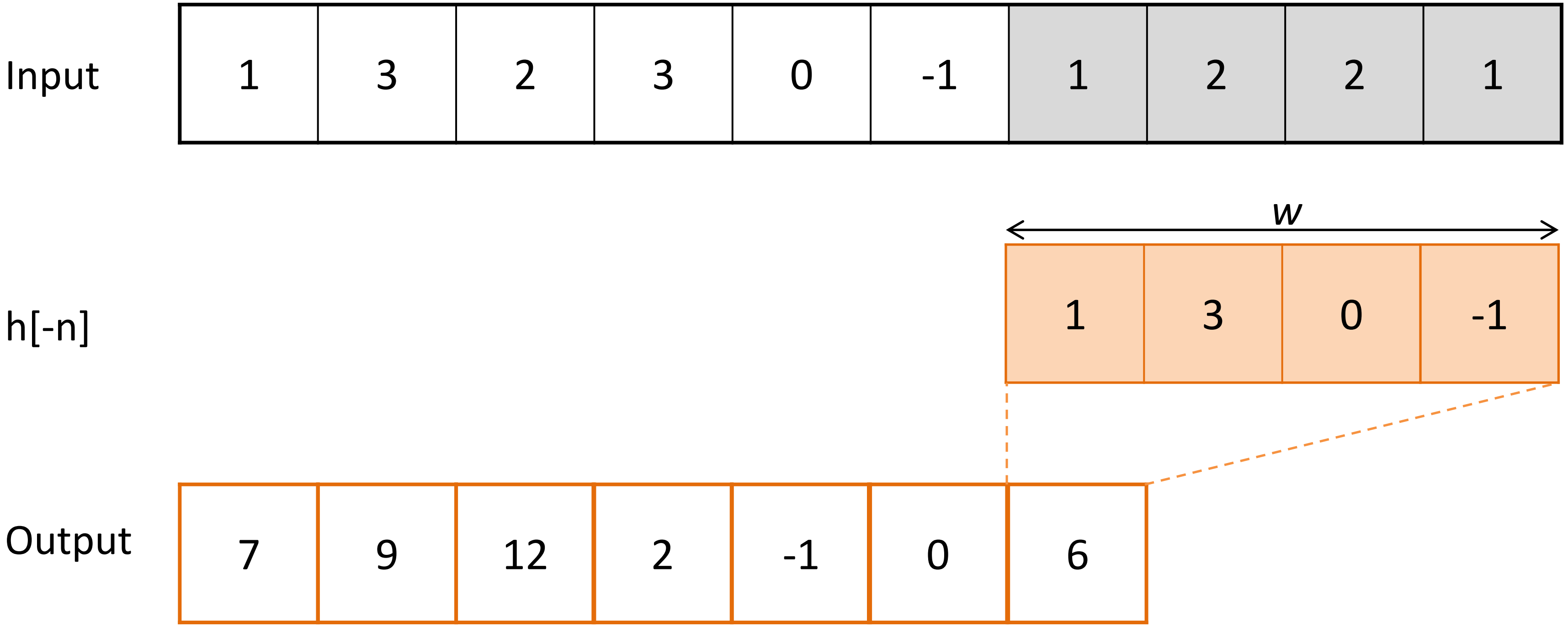


1D Convolution



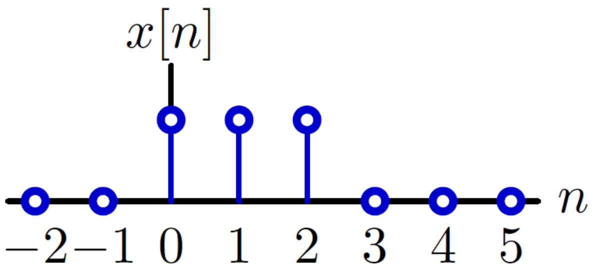
Source: Dr. Francois Fleuret at EPFL

1D Convolution

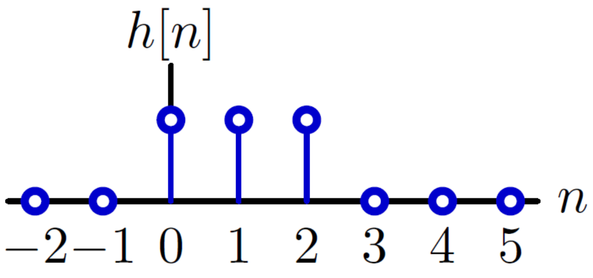


Structure of Convolution

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$



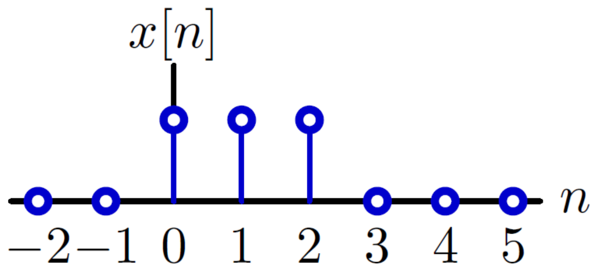
$*$



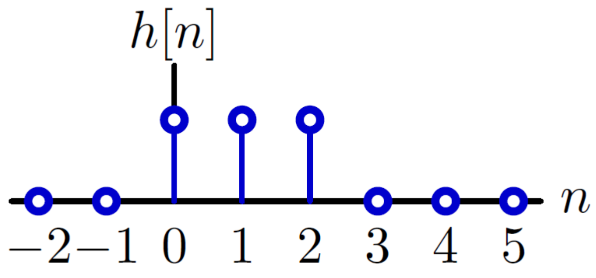
Source: Prof. Denny Freeman at MIT

Structure of Convolution

$$y[0] = \sum_{k=-\infty}^{\infty} x[k]h[0-k]$$

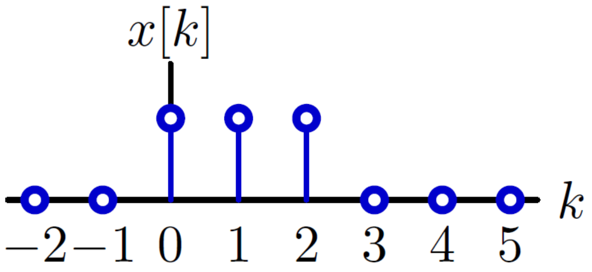


$*$

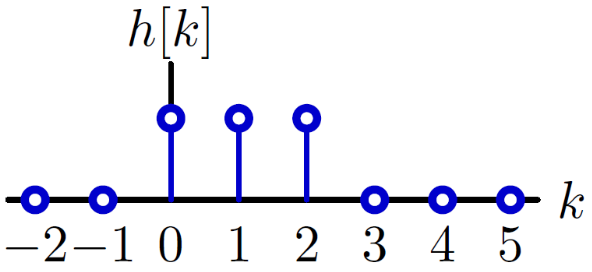


Structure of Convolution

$$y[0] = \sum_{k=-\infty}^{\infty} x[k]h[0-k]$$



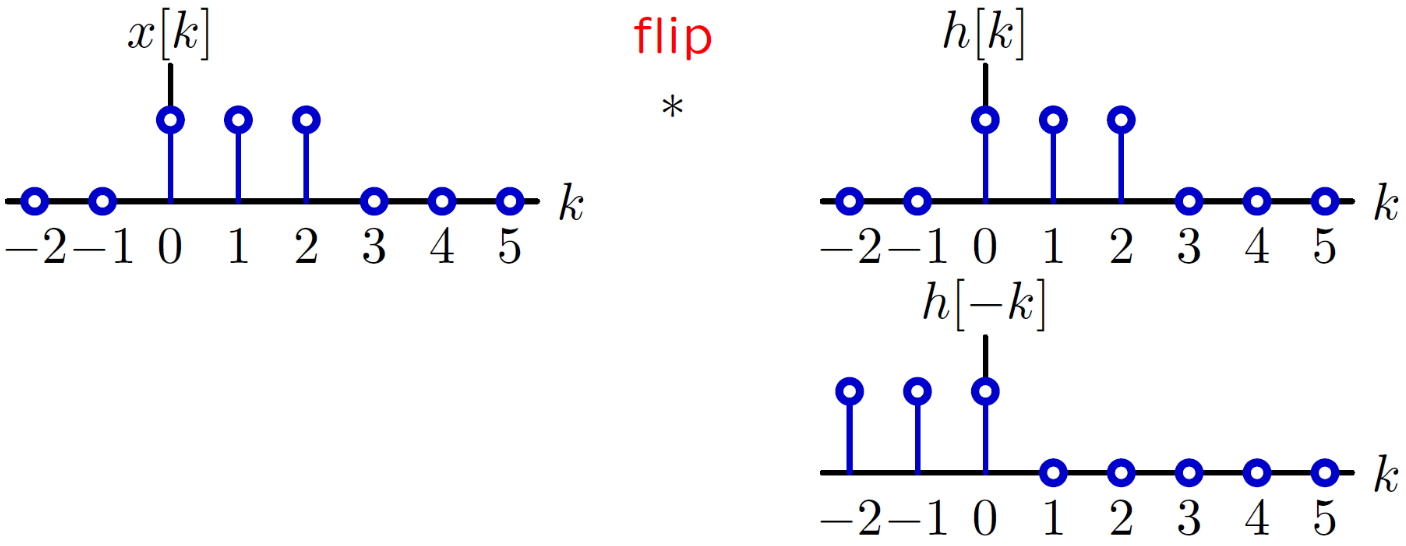
$*$



Source: Prof. Denny Freeman at MIT

Structure of Convolution

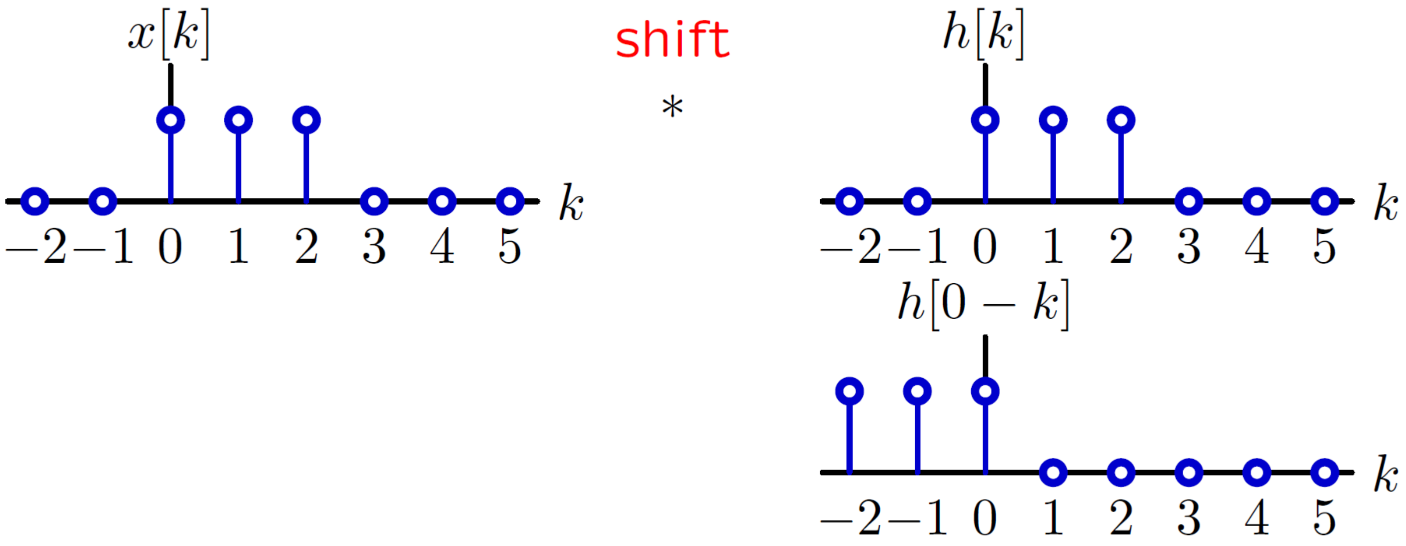
$$y[0] = \sum_{k=-\infty}^{\infty} x[k]h[0 - k]$$



Source: Prof. Denny Freeman at MIT

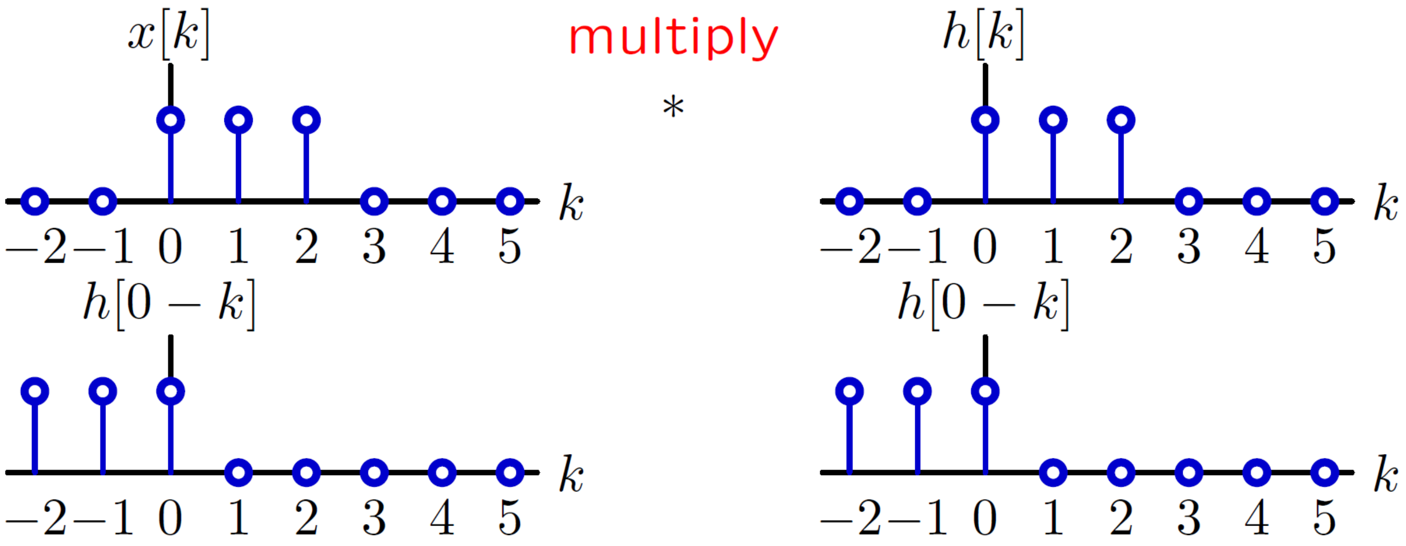
Structure of Convolution

$$y[0] = \sum_{k=-\infty}^{\infty} x[k]h[0-k]$$



Structure of Convolution

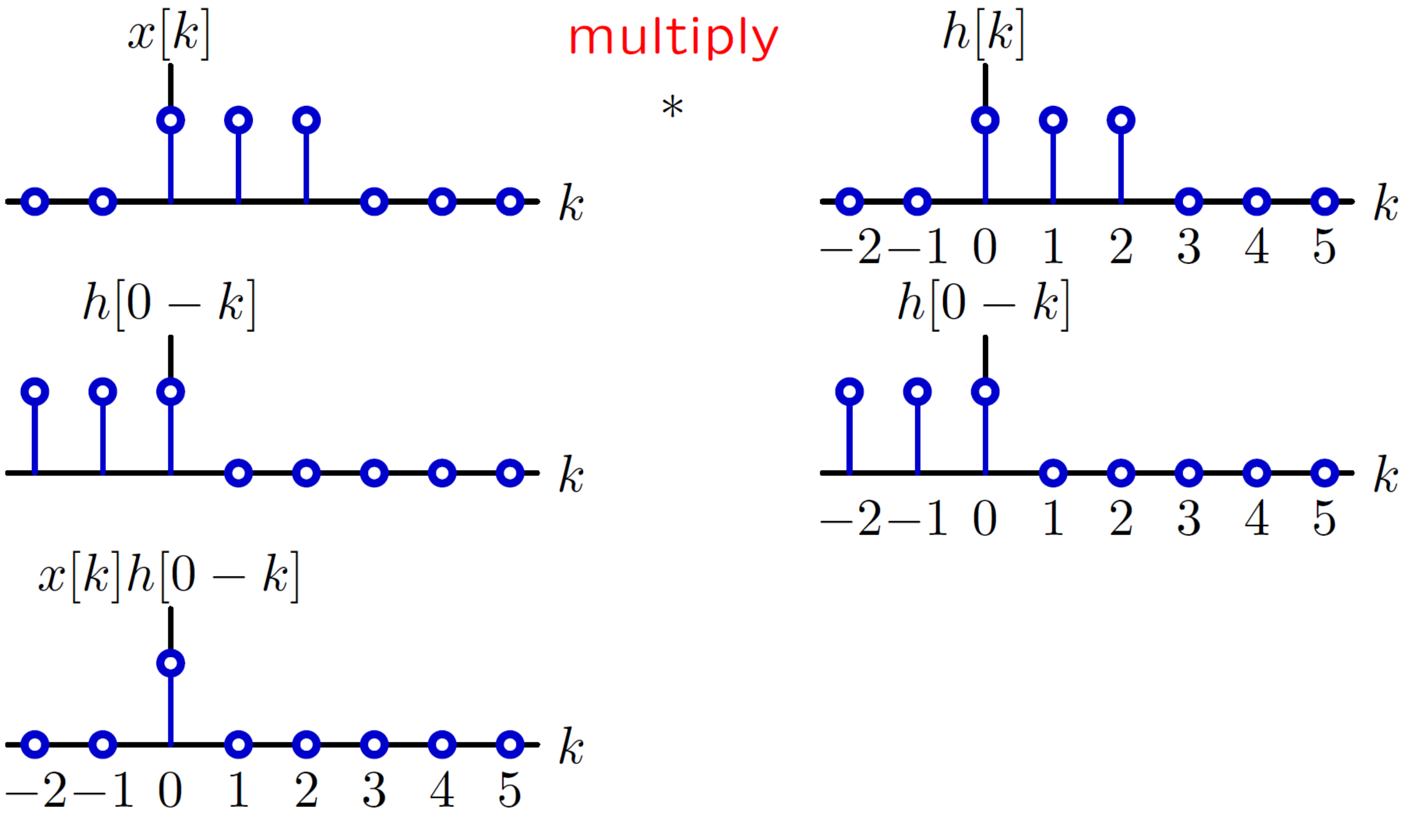
$$y[0] = \sum_{k=-\infty}^{\infty} x[k]h[0-k]$$



Source: Prof. Denny Freeman at MIT

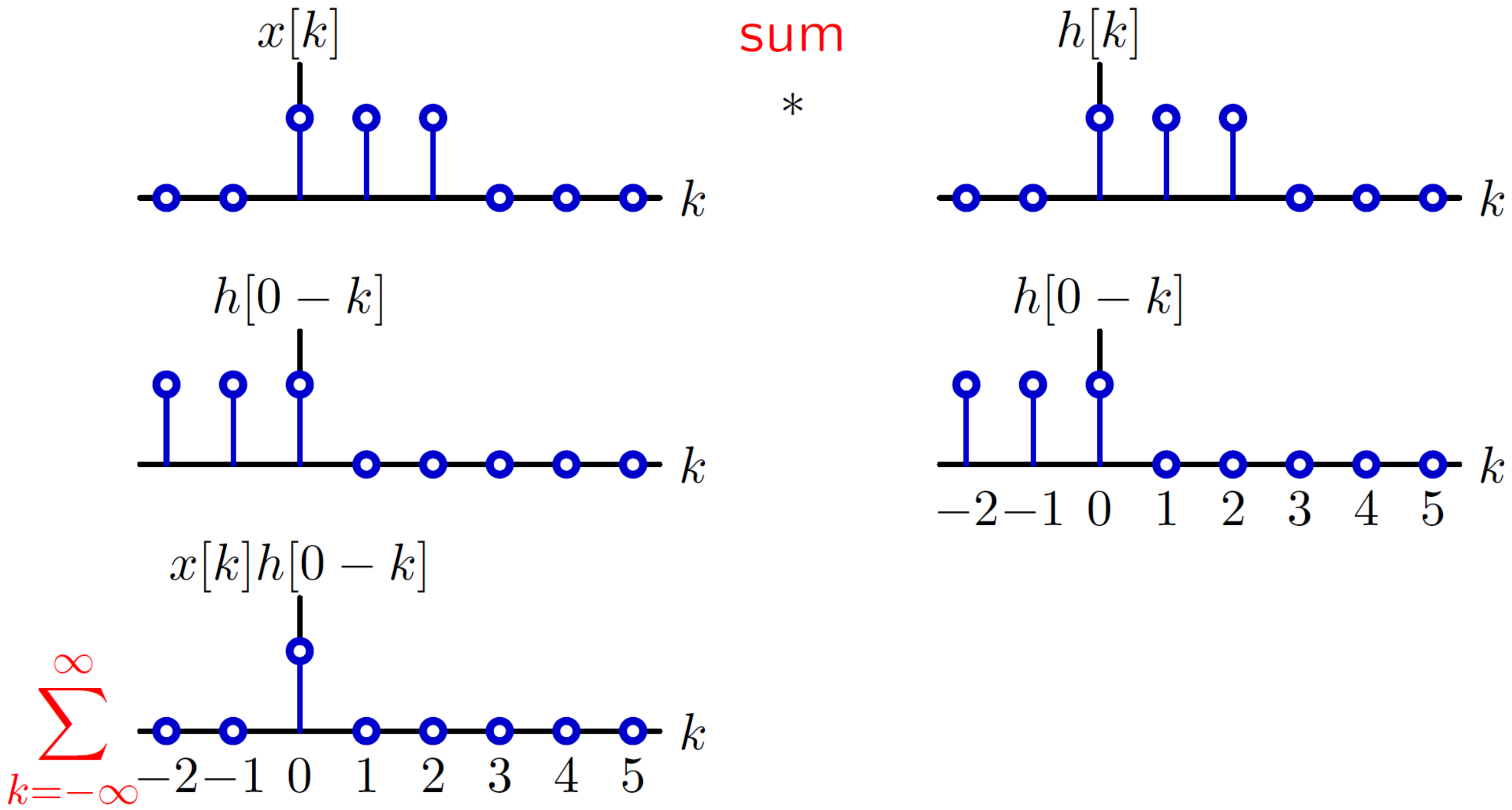
Structure of Convolution

$$y[0] = \sum_{k=-\infty}^{\infty} x[k]h[0-k]$$



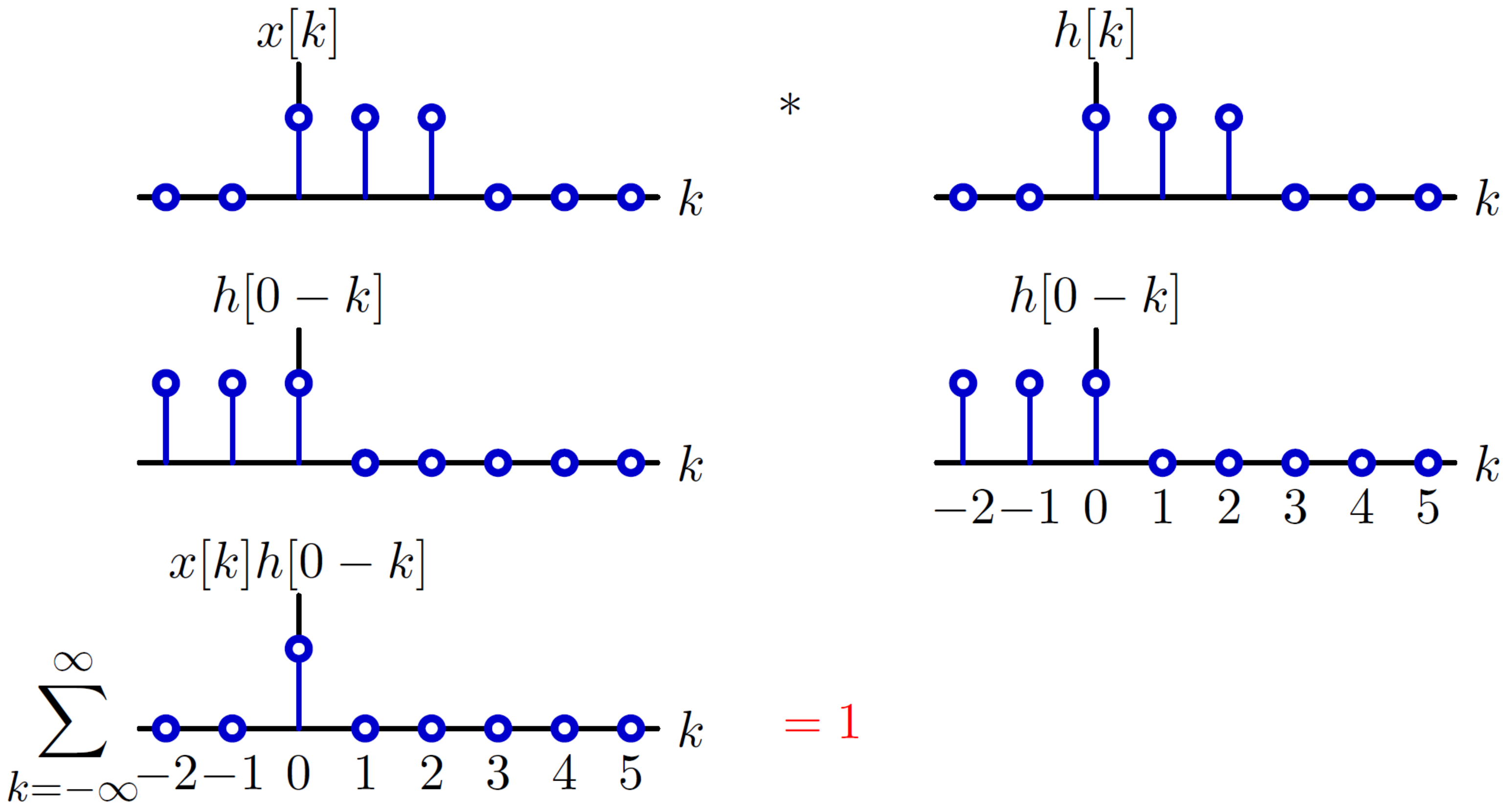
Structure of Convolution

$$y[0] = \sum_{k=-\infty}^{\infty} x[k]h[0-k]$$



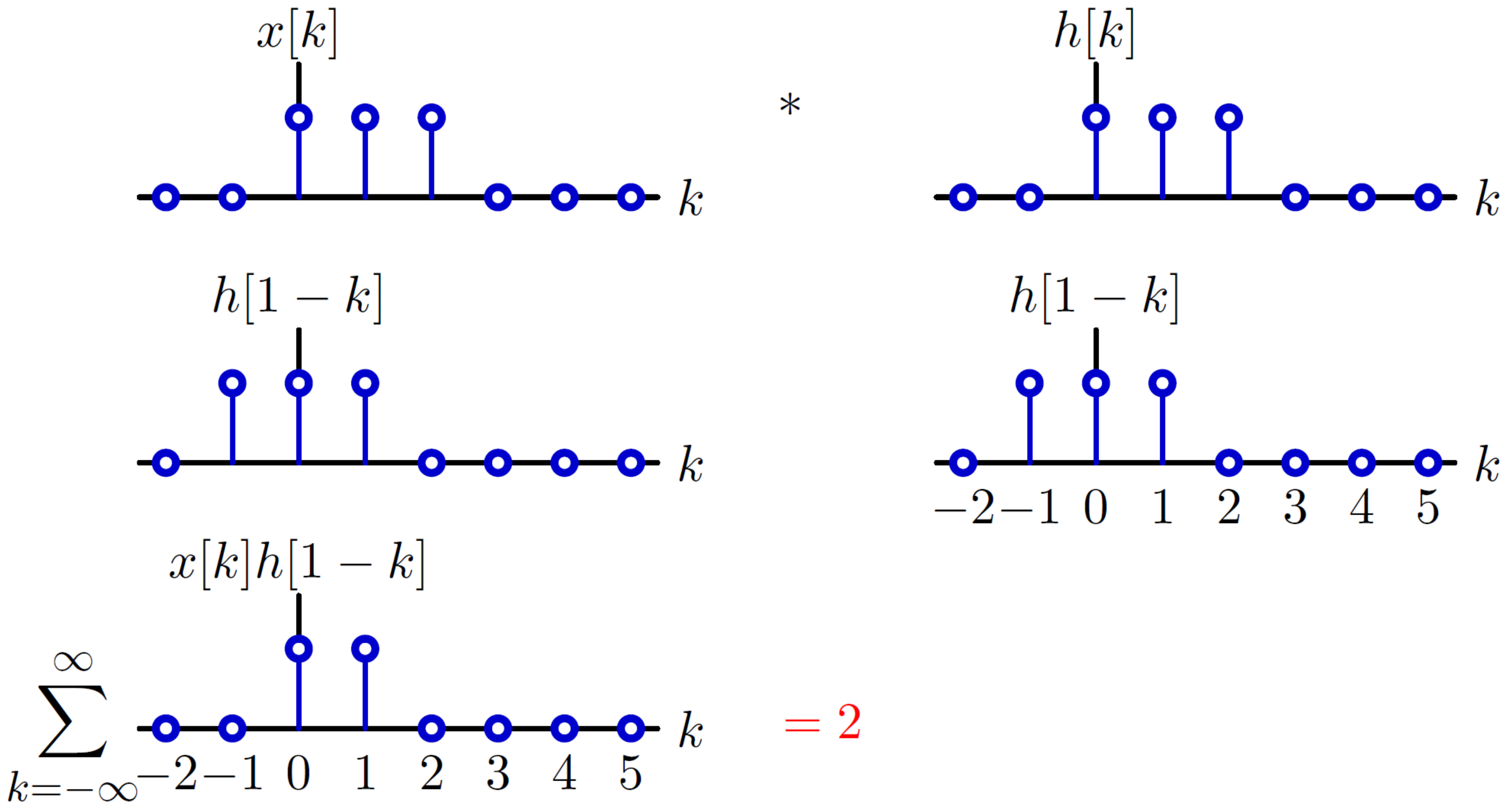
Structure of Convolution

$$y[0] = \sum_{k=-\infty}^{\infty} x[k]h[0-k]$$



Structure of Convolution

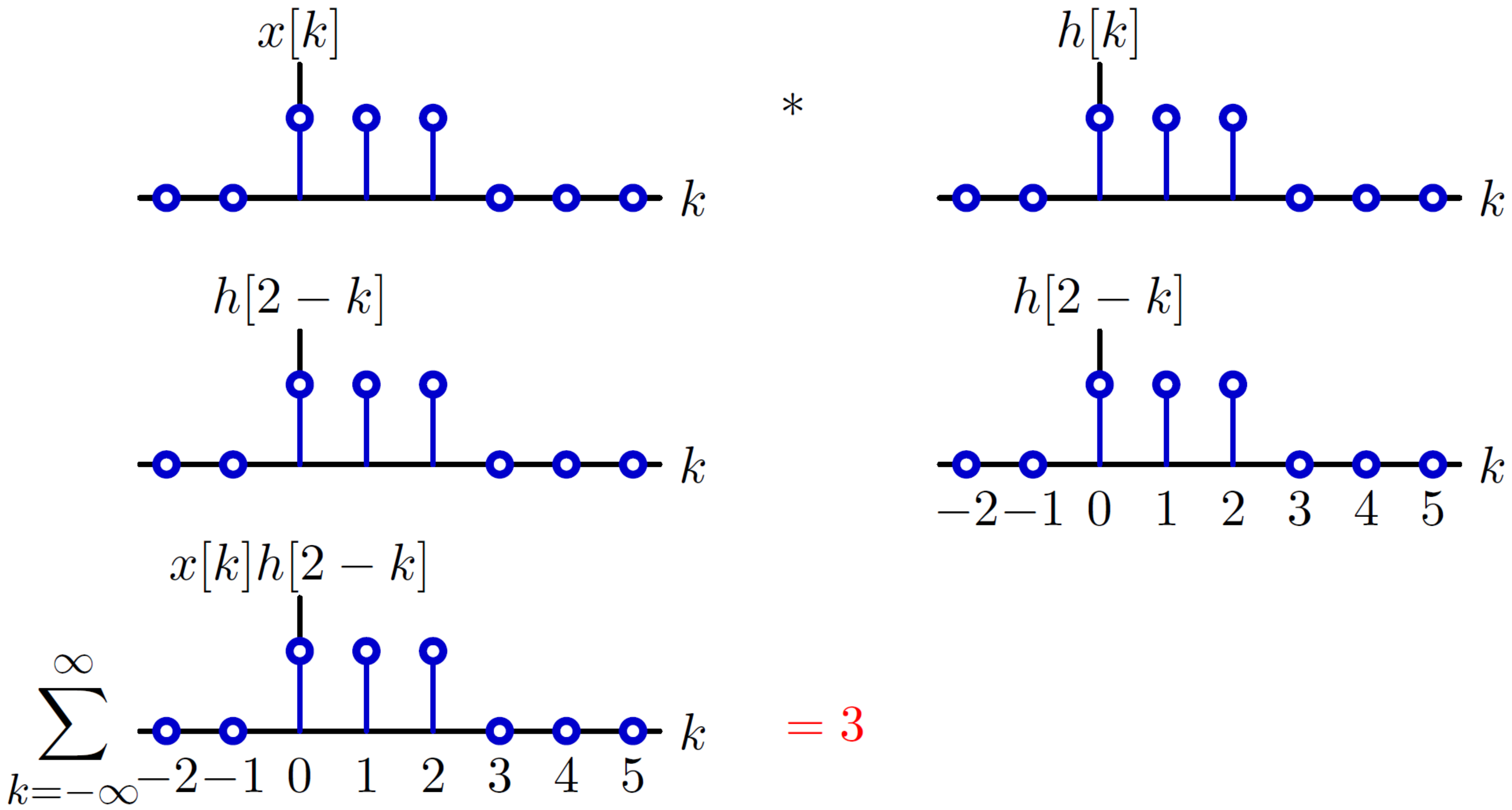
$$y[{\color{red}1}] = \sum_{k=-\infty}^{\infty} x[k]h[1-k]$$



Source: Prof. Denny Freeman at MIT

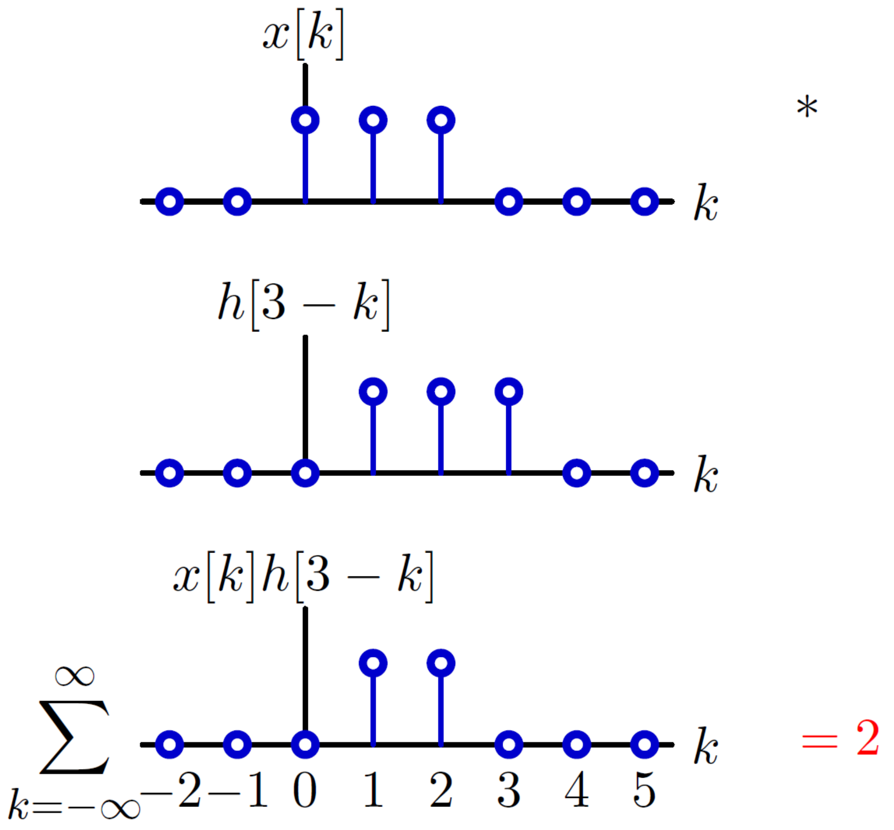
Structure of Convolution

$$y[\textcolor{red}{2}] = \sum_{k=-\infty}^{\infty} x[k]h[2-k]$$



Structure of Convolution

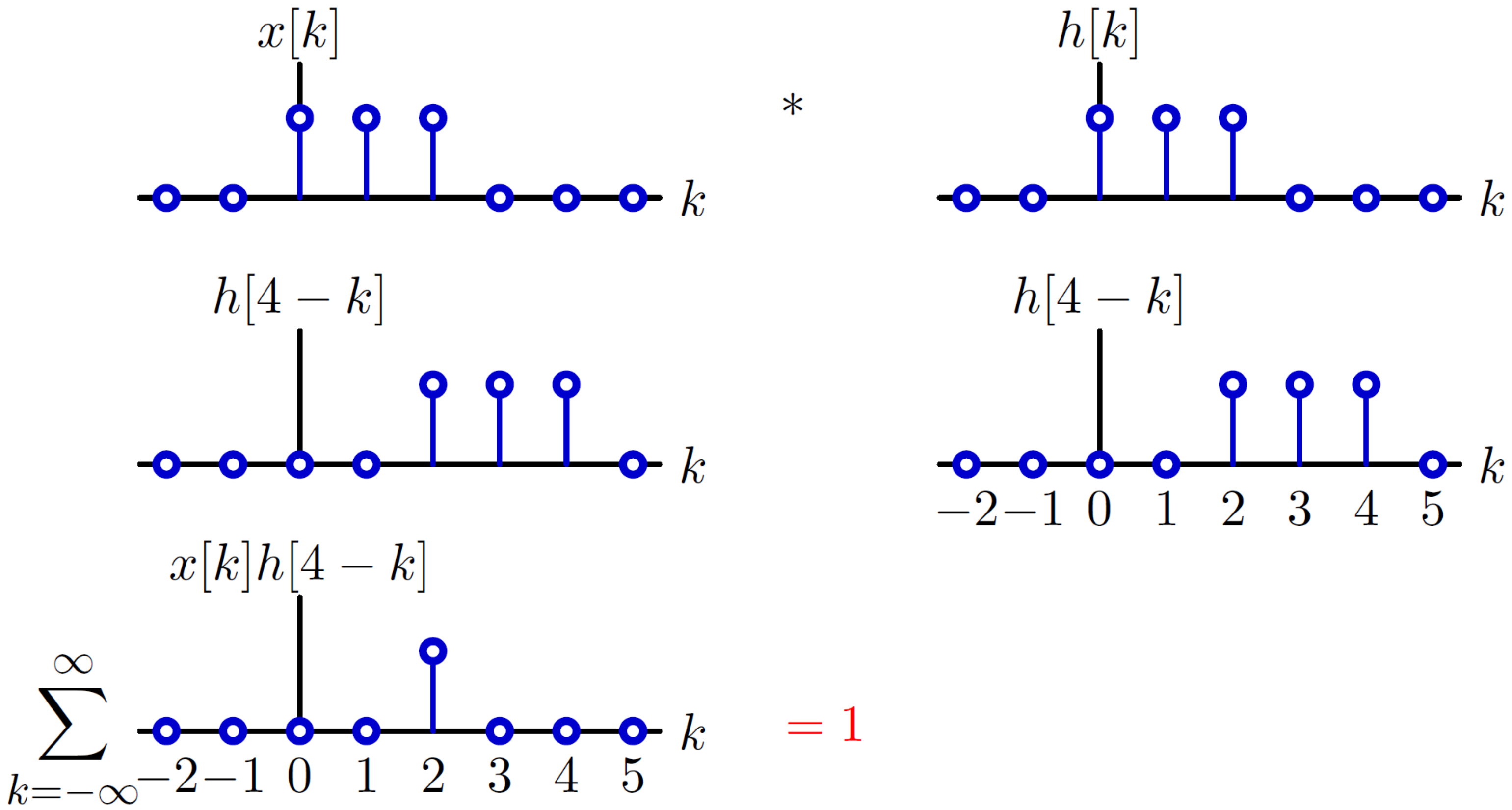
$$y[3] = \sum_{k=-\infty}^{\infty} x[k]h[3-k]$$



Source: Prof. Denny Freeman at MIT

Structure of Convolution

$$y[4] = \sum_{k=-\infty}^{\infty} x[k]h[4-k]$$

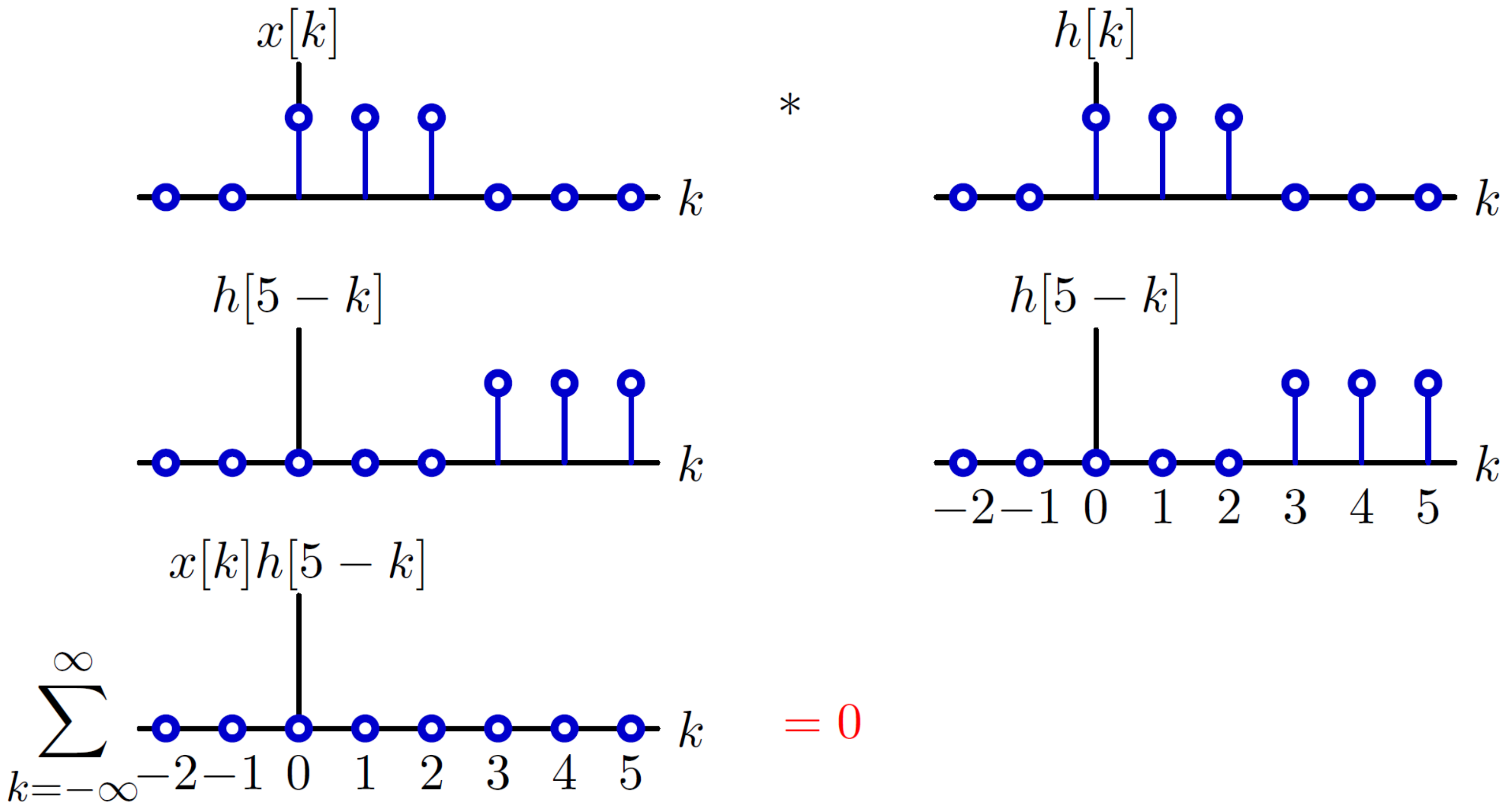


$= 1$

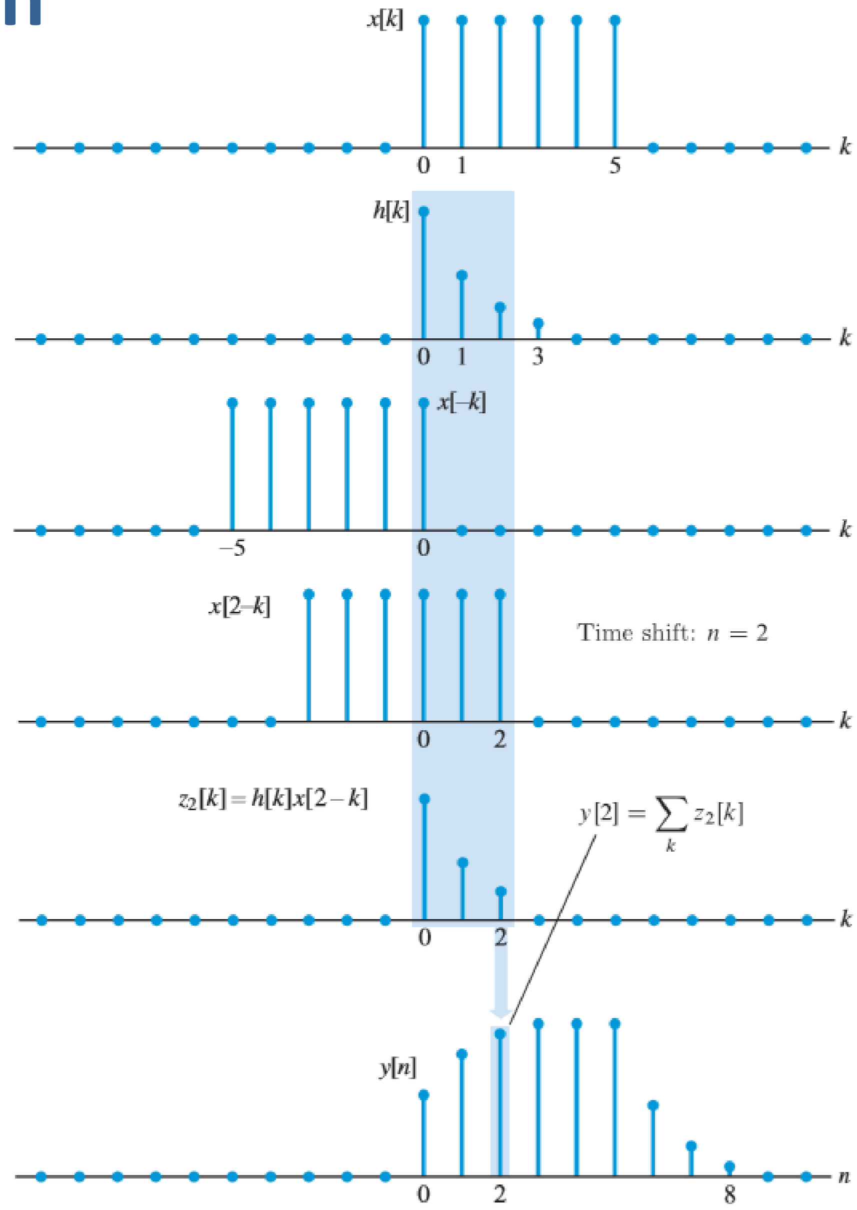
Source: Prof. Denny Freeman at MIT

Structure of Convolution

$$y[5] = \sum_{k=-\infty}^{\infty} x[k]h[5-k]$$



Graphical Illustration



Source: Applied Digital Signal Processing, Theory and Practice

Discrete-Time Convolution: Summary

- Representing an LTI system by a single signal

$$x[n] \longrightarrow \boxed{h[n]} \longrightarrow y[n]$$

- Unit-impulse response $h[n]$ is a complete description of an LTI system

- Given $h[n]$, one can compute the response to any arbitrary input signal $x[n]$

$$y[n] = (x * h)[n] \equiv \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

Convolution: Commutative

$$y[n] = \sum_{m=-\infty}^{\infty} h[n-m]\, x[m] = x[n] * h[n]$$

- Convolution is commutative

$$x * h = h * x$$

$$y[n] = \sum_{m=-\infty}^{\infty} h[n-m]\, x[m] = x[n] * h[n]$$

$$= \sum_{k=-\infty}^{\infty} h[k]\, x[n-k] = \sum_{k=-\infty}^{\infty} x[n-k]\, h[k] = h[n] * x[n]$$

$$(k = n - m \quad \Longrightarrow \quad m = n - k)$$

→ Signal = System

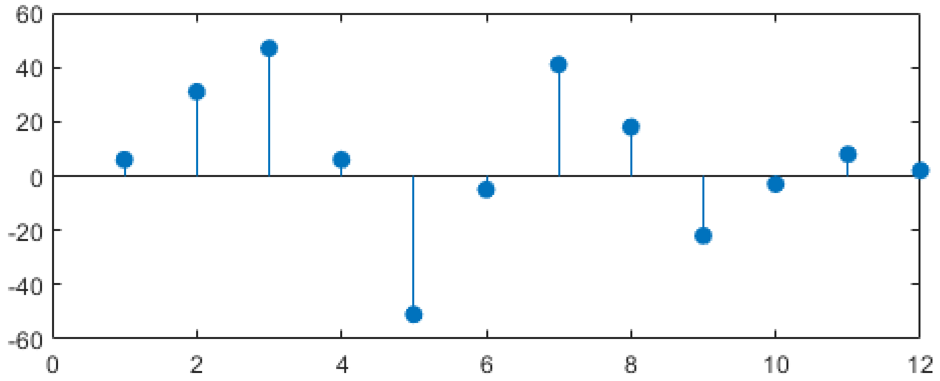
Convolution in MATLAB

- For finite-length signals

$$x[n] = \{3, 11, 7, \underset{\uparrow}{0}, -1, 4, 2\} \qquad -3 \le n \le 3$$

$$h[n] = \{2, \underset{\uparrow}{3}, 0, -5, 2, 1\} \qquad -1 \le n \le 4$$

```
%% matlab command 'conv'
x = [3,11,7,0,-1,4,2];
h = [2,3,0,-5,2,1];
y = conv(x,h);

stem(y,'filled'), ylim([-60 60])

% this is not correct
```



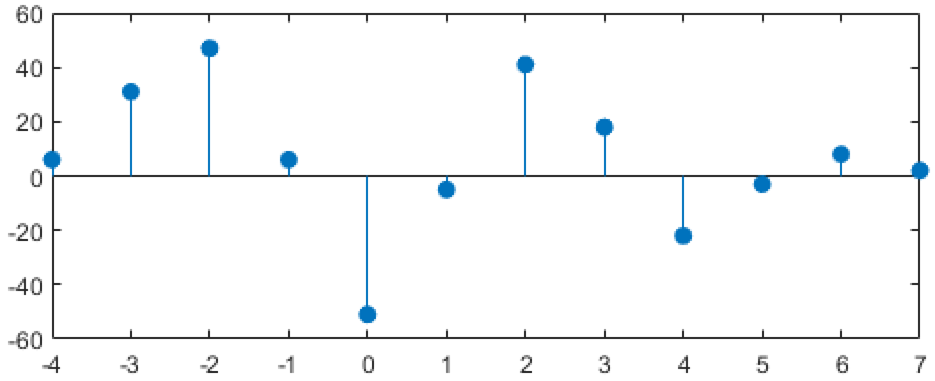
Convolution in MATLAB

- For finite-length signals

$$x[n] = \{3, 11, 7, \underset{\uparrow}{0}, -1, 4, 2\} \qquad -3 \le n \le 3$$

$$h[n] = \{2, \underset{\uparrow}{3}, 0, -5, 2, 1\} \qquad -1 \le n \le 4$$

```
x = [3,11,7,0,-1,4,2];  nx = [-3:3];
h = [2,3,0,-5,2,1];     nh = [-1:4];

[y,ny] = conv_m(x,nx,h,nh);

stem(ny,y,'filled'), axis tight, ylim([-60 60])

% this is what we want
```



Convolution Function

- You have to include conv_m function file in the path

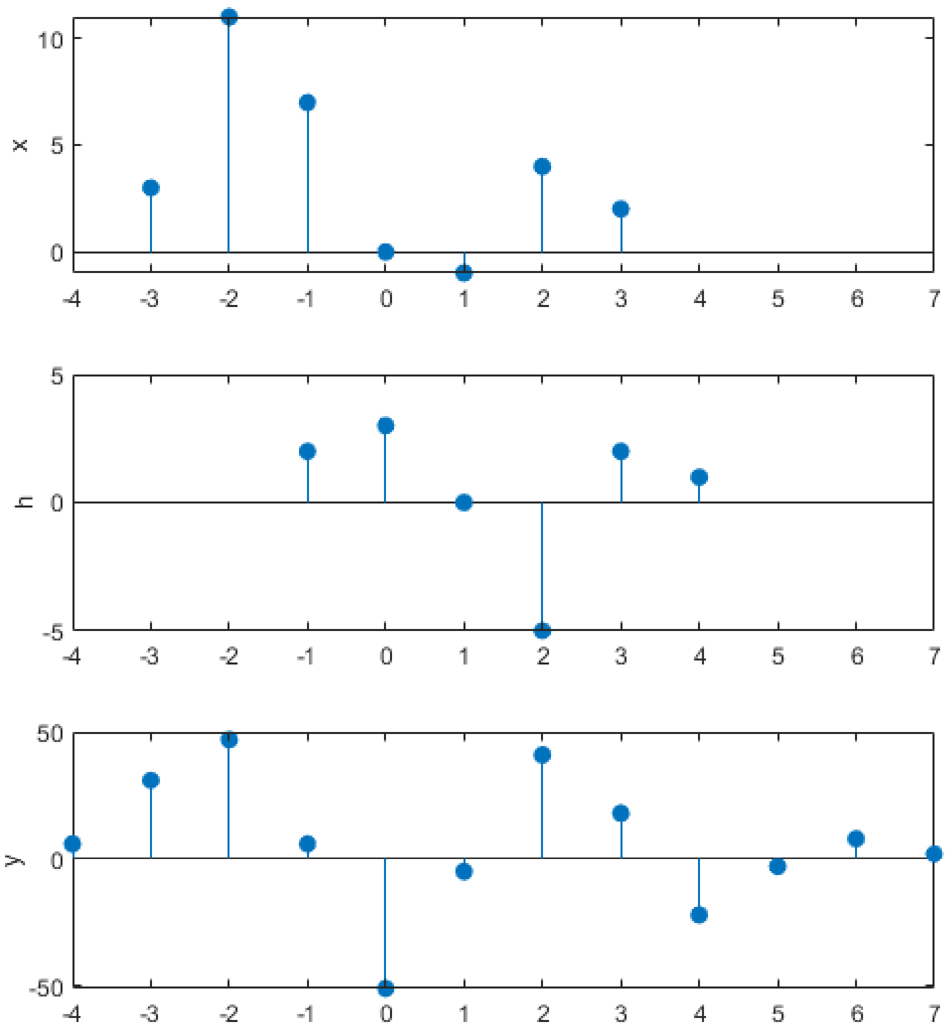
```
function [y,ny] = conv_m(x,nx,h,nh)

% Modified convolution routine for signal processing
% [y,ny] = conv_m(x,nx,h,nh)
%  y = convolution result
% ny = support of y
%  x = first signal on support nx
% nx = support of x
%  h = second signal on support nh
% nh = support of h

nyb = nx(1) + nh(1);
nye = nx(length(x)) + nh(length(h));
ny = [nyb:nye];
y = conv(x,h);
```

Convolution in MATLAB

```
subplot(3,1,1), stem(nx,x,'filled');
subplot(3,1,2), stem(nh,h,'filled');
subplot(3,1,3), stem(ny,y,'filled');
```

$$x[n] = \{3, 11, 7, \underset{\uparrow}{0}, -1, 4, 2\} \qquad -3 \le n \le 3$$

$$h[n] = \{2, \underset{\uparrow}{3}, 0, -5, 2, 1\} \qquad -1 \le n \le 4$$

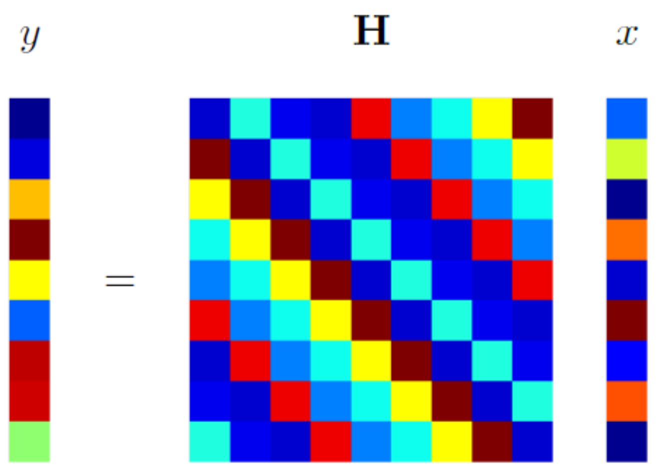


Convolution For Finite-Length Signals

- Circular convolution

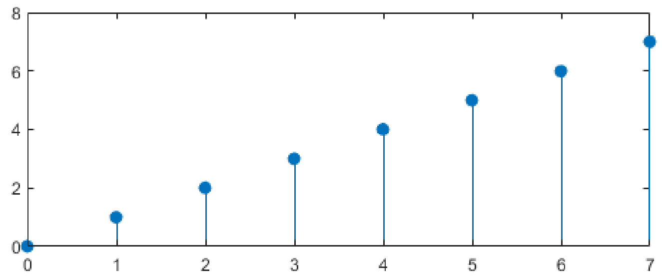
$$y[n] = x[n] \circledast h[n] = \sum_{m=0}^{N-1} h\left[(n-m)_N\right]\, x[m]$$

  – Circularly time reverse the impulse response h and circularly shift it n time steps to the right (delay)
  – Compute the inner product between the shifted impulse response and the input vector x



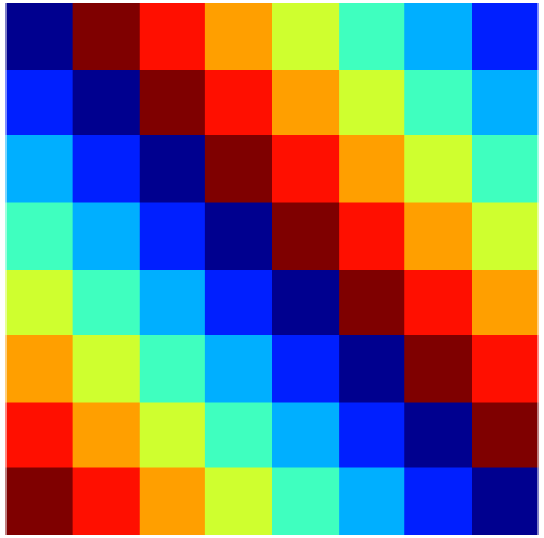
Circular Convolution in MATLAB

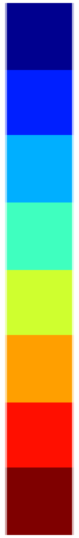
```
N = 8;
n = 0:N-1;

h = [0 1 2 3 4 5 6 7]';

stem(n,h,'filled');
```




```
H = zeros(N,N);
for k = 0:N-1
    H(:,k+1) = circshift(h,k);
end
```

```
H =

     0     7     6     5     4     3     2     1
     1     0     7     6     5     4     3     2
     2     1     0     7     6     5     4     3
     3     2     1     0     7     6     5     4
     4     3     2     1     0     7     6     5
     5     4     3     2     1     0     7     6
     6     5     4     3     2     1     0     7
     7     6     5     4     3     2     1     0
```

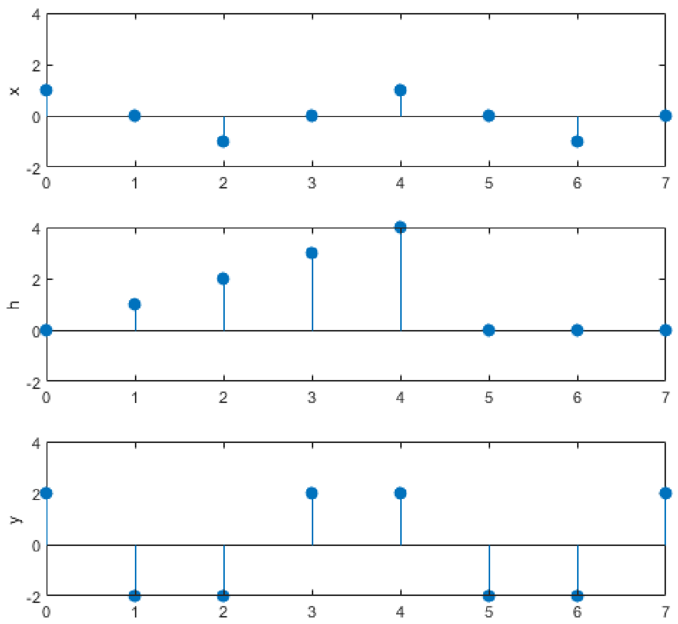


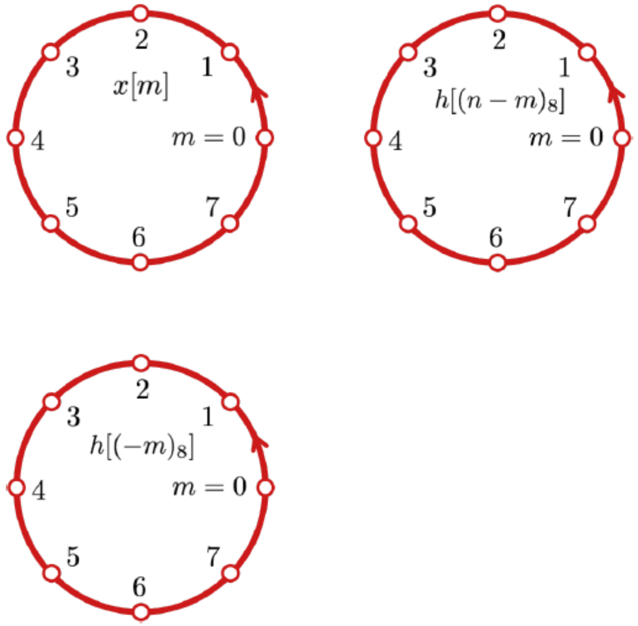


```
x = [1,1,0,0,0,0,0,0]';
y1 = H*x
```

Convolution For Finite-Length Signals

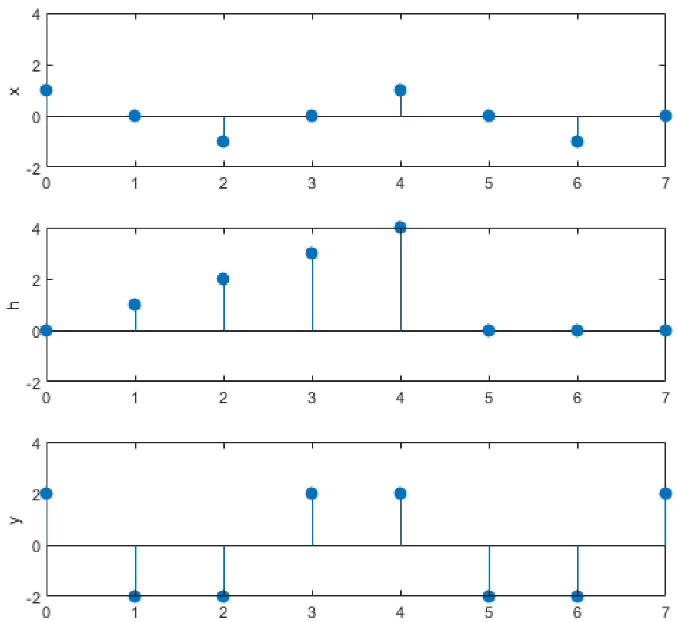
- Think about circular convolution in time
  - <u>Circularly</u> time reverse the impulse response h and <u>circularly</u> shift it n time steps to the right (delay)
  - Compute the inner product between the shifted impulse response and the input vector x

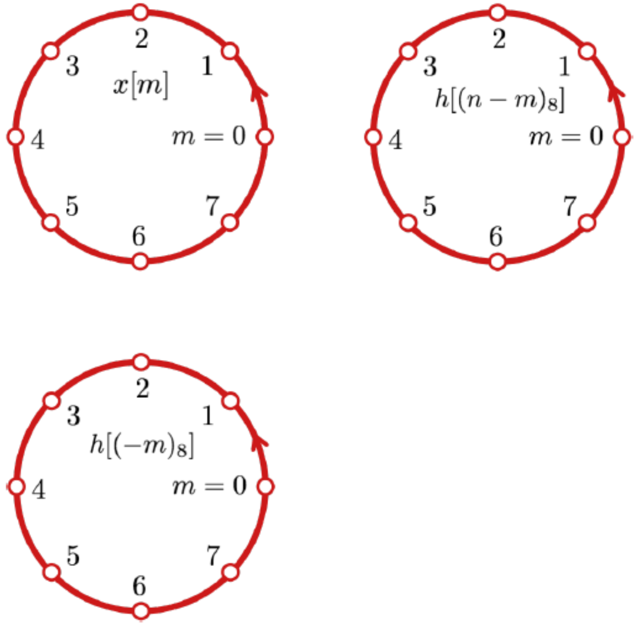




Convolution For Finite-Length Signals

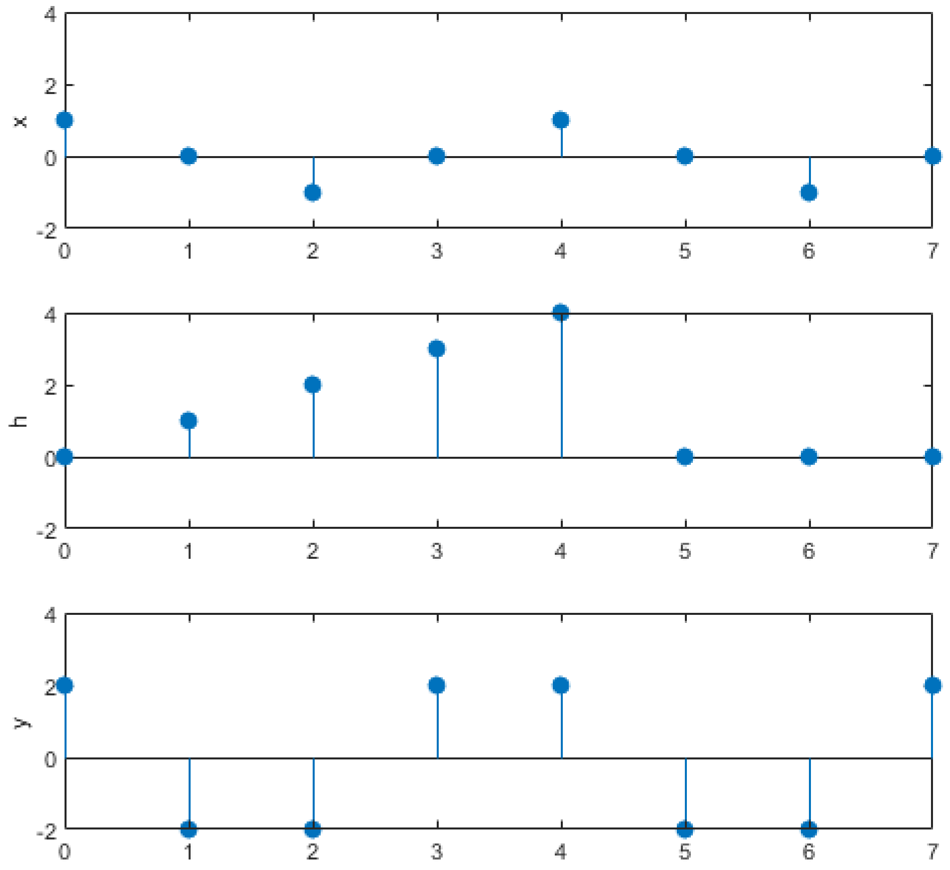
- Think about circular convolution in time
  - <u>Circularly</u> time reverse the impulse response h and <u>circularly</u> shift it n time steps to the right (delay)
  - Compute the inner product between the shifted impulse response and the input vector x

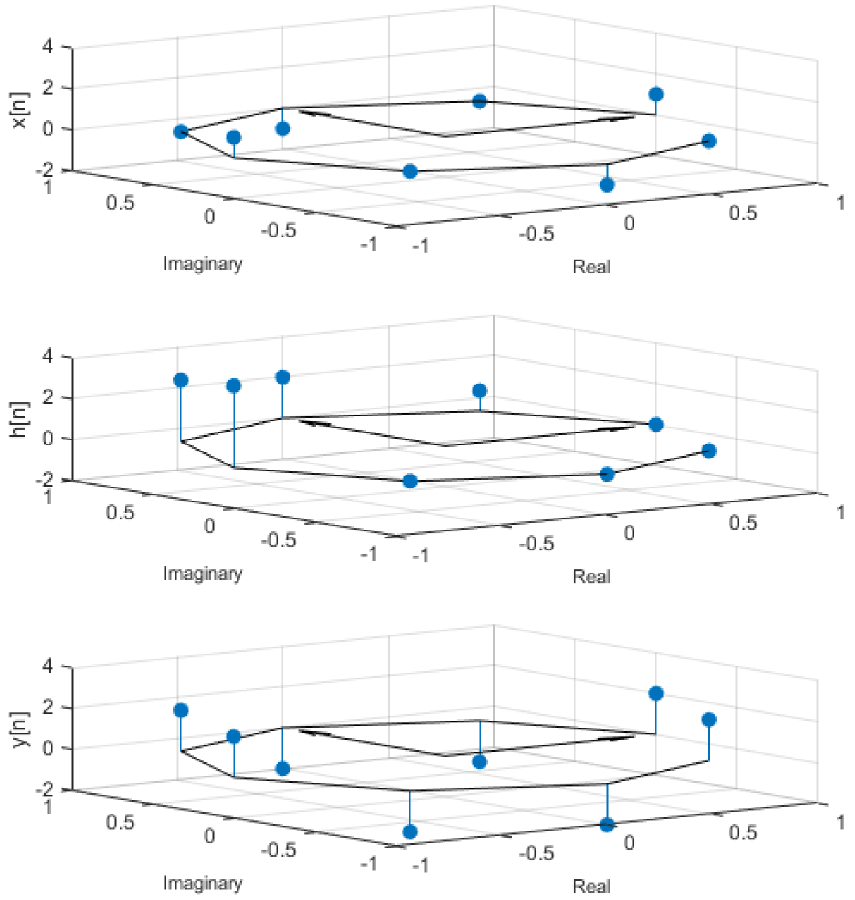




Circular Convolution in MATLAB

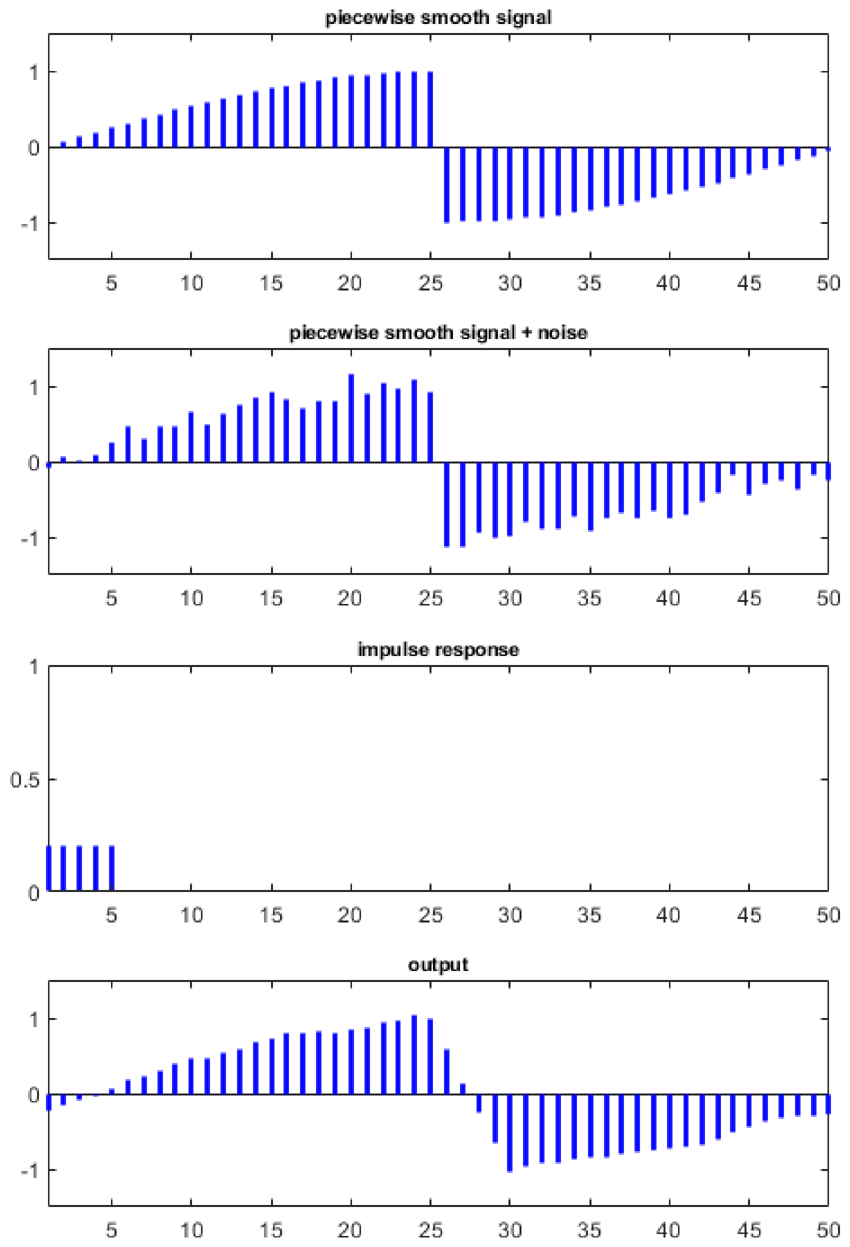
```
N = 8;
xn = 0:N-1;
x = [1 0 -1 0 1 0 -1 0];
h = [0 1 2 3 4 0 0 0];
y = cconv(x,h,N);
```





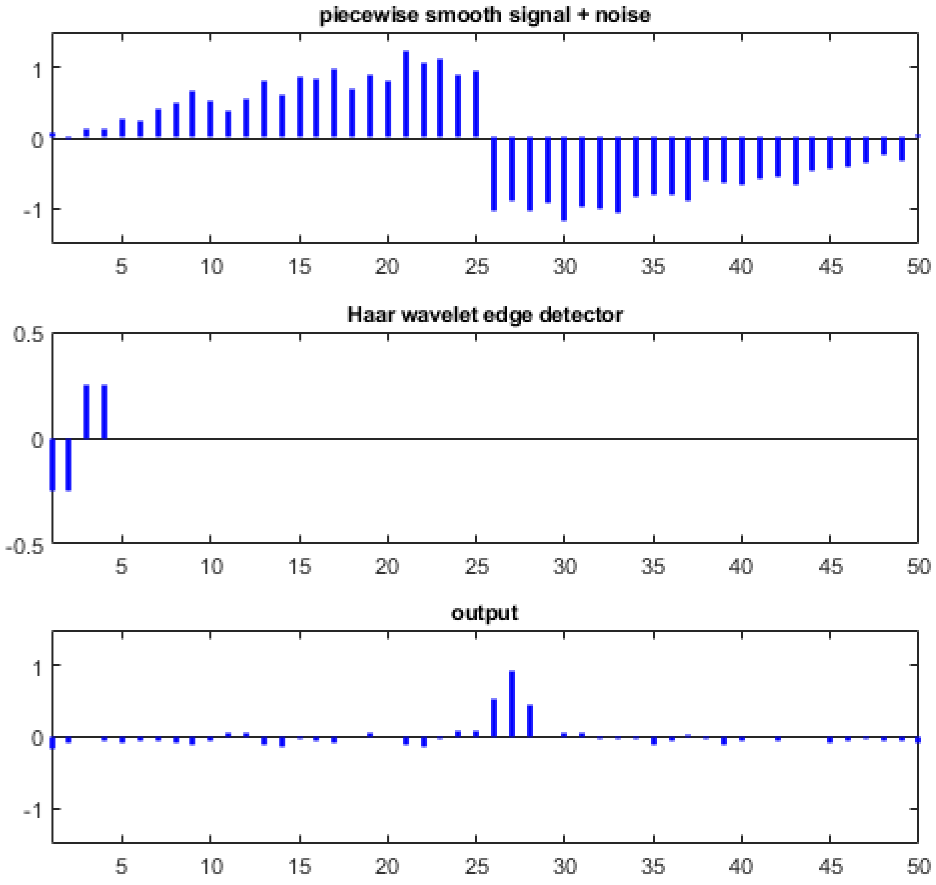
Convolution Examples

De-noising

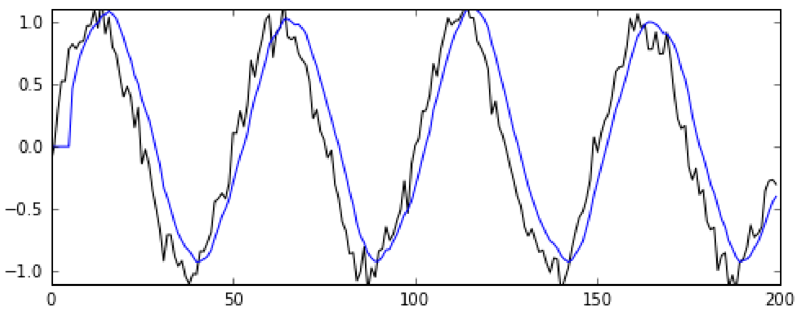


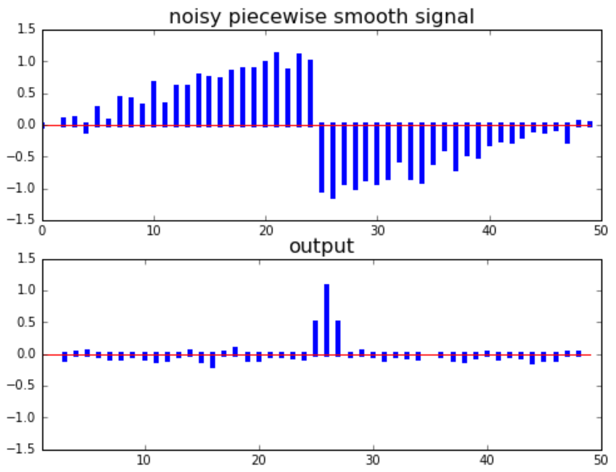
piecewise smooth signal
piecewise smooth signal + noise
impulse response
output

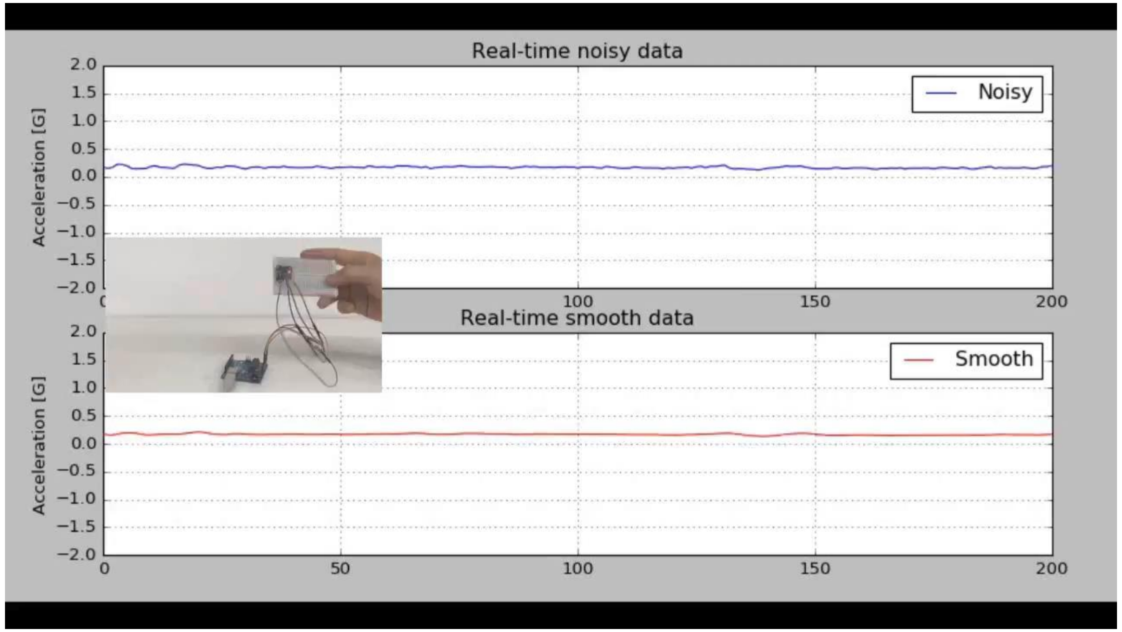
Edge Detection

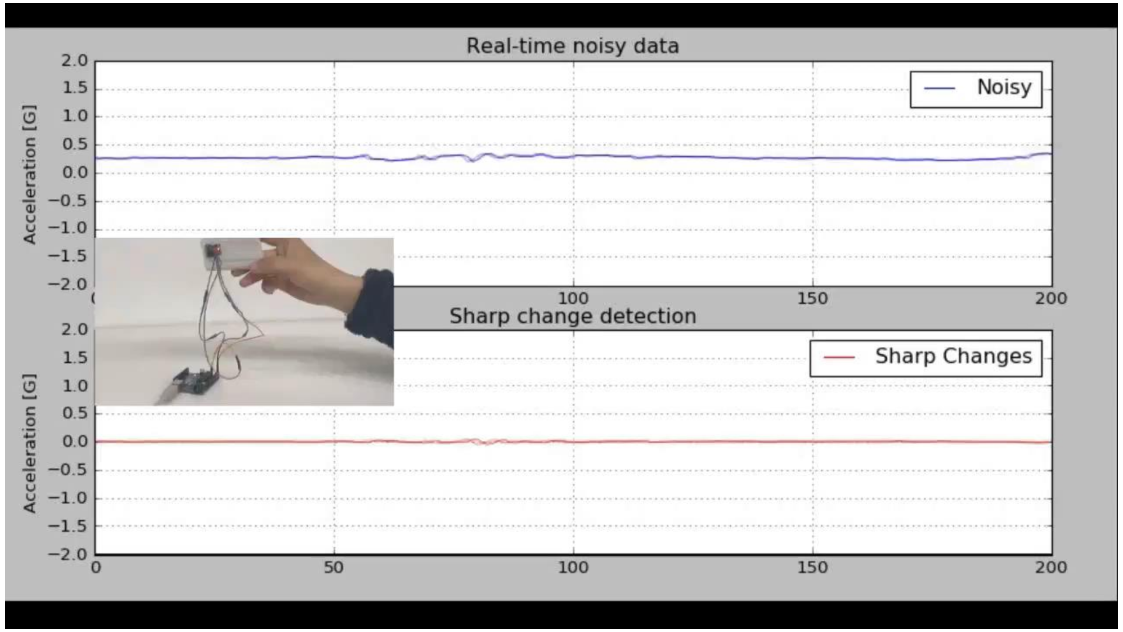


Smoothing and Detection of Abrupt Changes



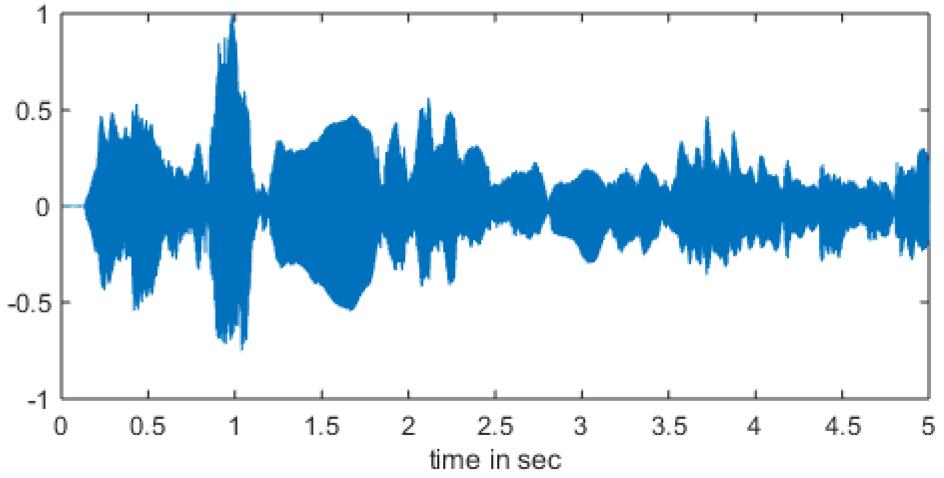






Example: Convolution on Audio

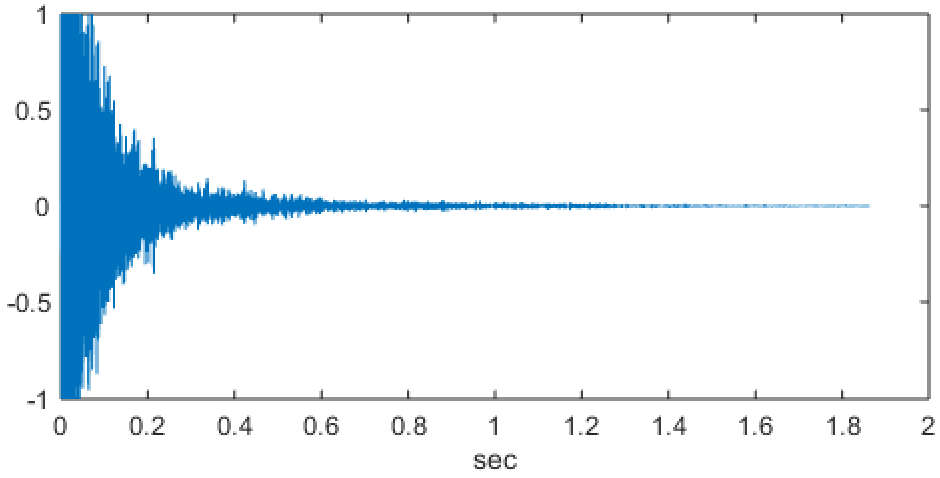
```
[x, Fs] = audioread([pwd,'\data_files\violin_origional.wav']);

x = x/max(x);   % normalize
sound(x, Fs);   % play a wave file with sampling rate Fs
```



Source: Prof. Allen Downey at Olin

Example: Convolution on Audio

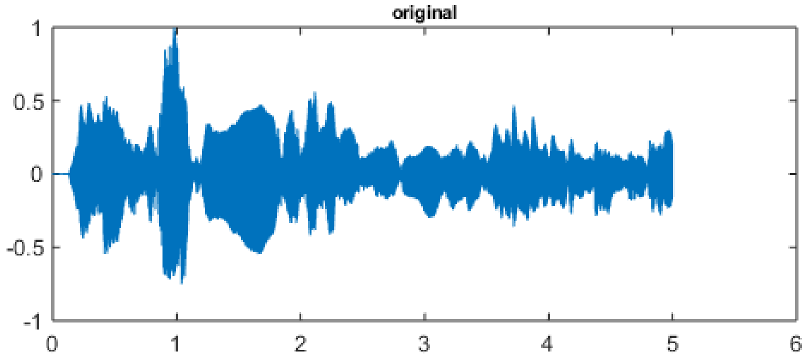
```
% impulse response in a closed room (by gunshot)

[h, Fs] = audioread([pwd,'\data_files\gunshot.wav']);
h = h(:,1); % stereo to mono

sound(h, Fs);
plot((1:length(h))/Fs, h), xlabel('sec')
```

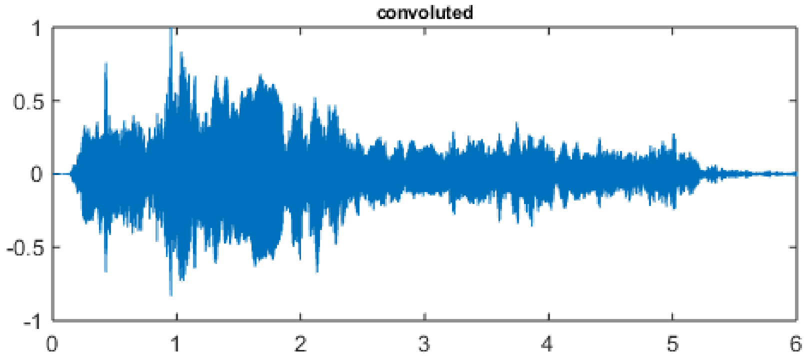


Source: Prof. Allen Downey at Olin

Example: Convolution on Audio

```
% image how the music played in a closed room sounds like

y = cconv(x,h);
y = y/max(y);

sound(y, Fs);
```





Source: Prof. Allen Downey at Olin