REVIEW 6 FIR FILTERS & LTI SYSTEMS

Modified TLH

Lecture Chapter 5

FIR Filtering Intro

LECTURE OBJECTIVES

- INTRODUCE FILTERING IDEA
  - Weighted Average
  - Running Average
- FINITE IMPULSE RESPONSE FILTERS
  - **FIR** Filters
  - Show how to **compute** the output y[n] from the input signal, x[n]

The Running (Moving) Average Filter

- A three-sample *causal* moving average filter is a special case of (5.1)

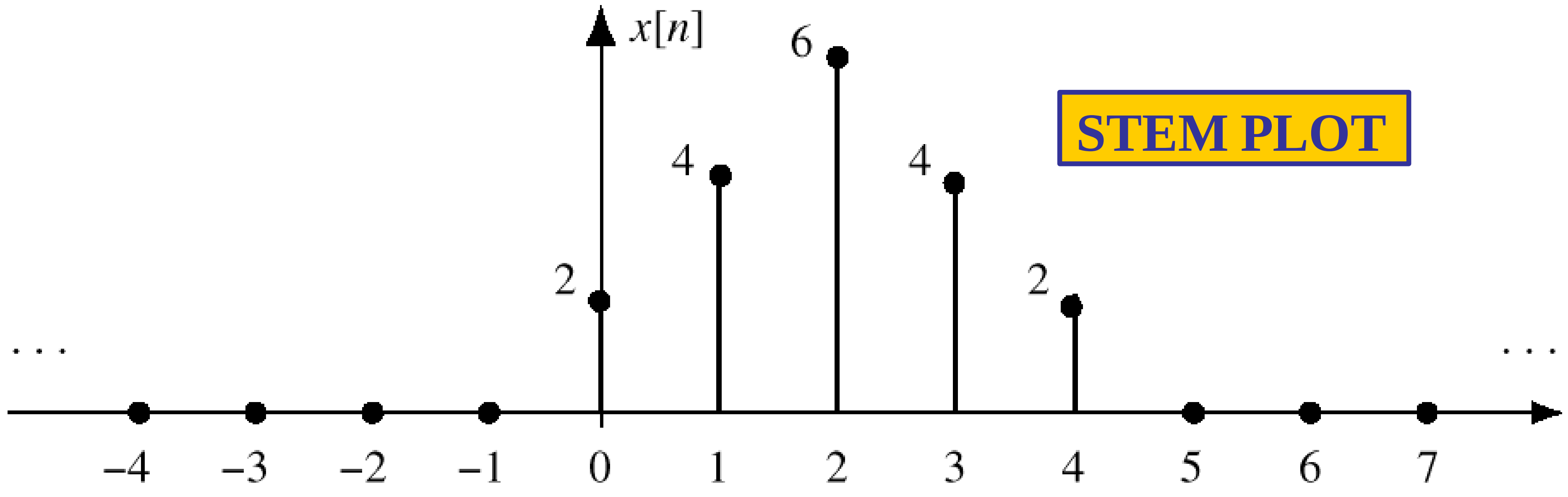
$$y[n] = \frac{1}{3}(x[n] + x[n-1] + x[n-2]), \qquad (5.4)$$

which uses no future input values to compute the present output

From ECE 2601 Chapter 5
Causal is From The Past

DISCRETE-TIME SIGNAL

- x[n] is a LIST of NUMBERS
  - INDEXED by "*n*"



3-pt AVERAGER

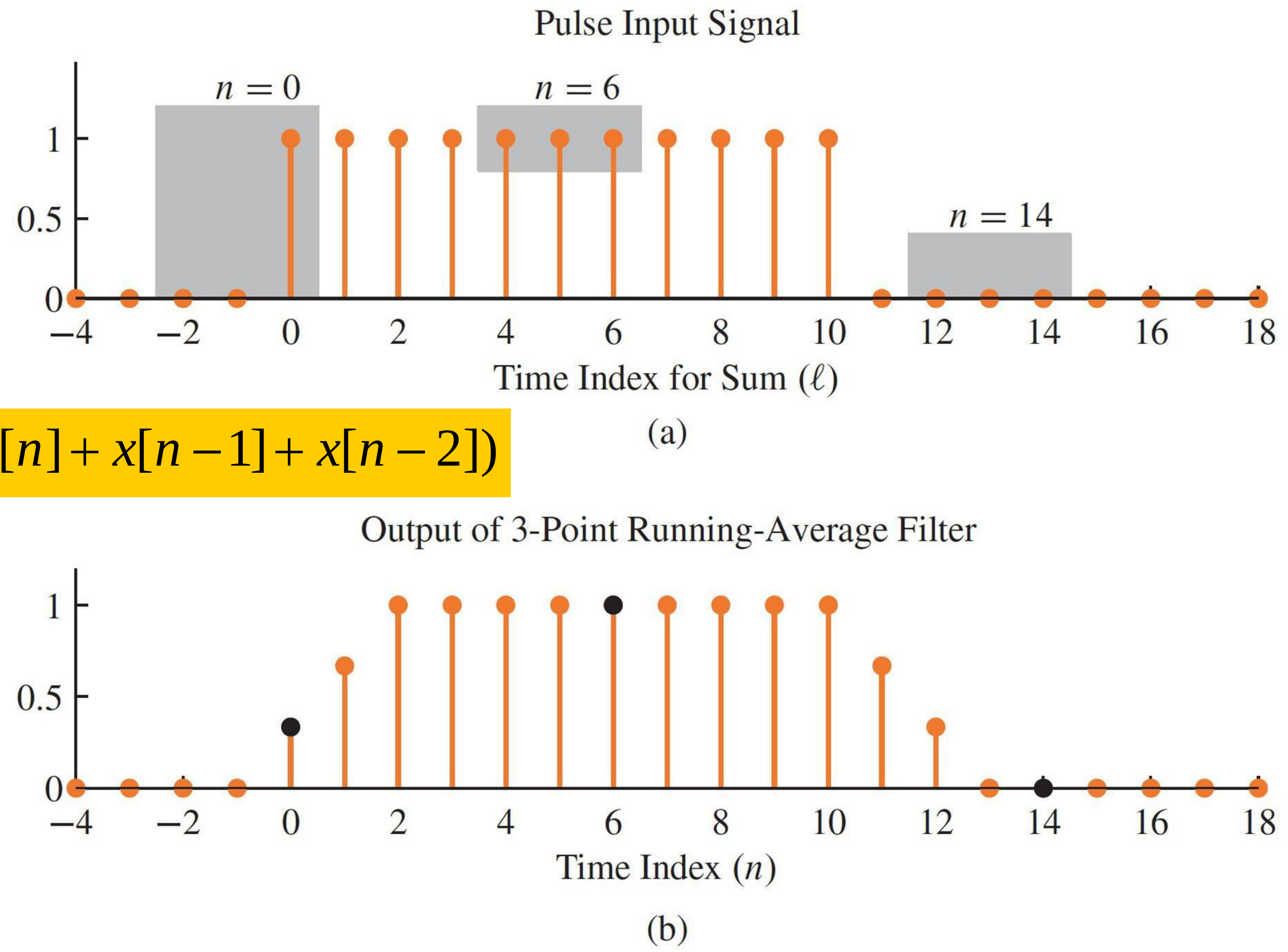
- Uses “PAST” VALUES of x[n]
  - IMPORTANT IF “n” represents REAL TIME
    - WHEN x[n] & y[n] ARE STREAMS

$$y[n] = \tfrac{1}{3}(x[n] + x[n-1] + x[n-2])$$

n	$n < -2$	-2	-1	0	1	2	3	4	5	6	7	$n > 7$
$x[n]$	0	0	0	2	4	**6**	**4**	**2**	0	0	0	0
$y[n]$	0	0	0	$\frac{2}{3}$	2	4	$\frac{14}{3}$	**4**	2	$\frac{2}{3}$	0	0

Aug 2016

CAUSAL 3-pt AVERAGER



$$y[n] = \tfrac{1}{3}(x[n] + x[n-1] + x[n-2])$$

Finite Impulse Response

- Each output value y[n] is the some of a FINITE number of weighted values of the input sequence x[n]

- The FIR filter can be represented in various ways:
- By a difference Equation Page 150
- By the Impulse Response Page 158
- By the Convolution Sum Page 162

GENERAL CAUSAL FIR FILTER

- FILTER COEFFICIENTS $\{b_k\}$
  - DEFINE THE FILTER

NOTE: Index k = 0, 1,2,…

$$y[n] = \sum_{k=0}^{M} b_k x[n-k]$$

- For example, $\boxed{b_k = \{3, -1, 2, 1\}}$

$$y[n] = \sum_{k=0}^{3} b_k x[n-k]$$

DIFFERENCE EQUATION

$$= 3x[n] - x[n-1] + 2x[n-2] + x[n-3]$$

GENERAL CAUSAL FIR FILTER

- FILTER COEFFICIENTS $\{b_k\}$

$$y[n] = \sum_{k=0}^{M} b_k x[n-k]$$

- FILTER **ORDER** is M
- FILTER **"LENGTH"** is L = M+1
  - NUMBER of FILTER COEFFS is L

SPECIAL INPUT SIGNALS

- x[n] = SINUSOID

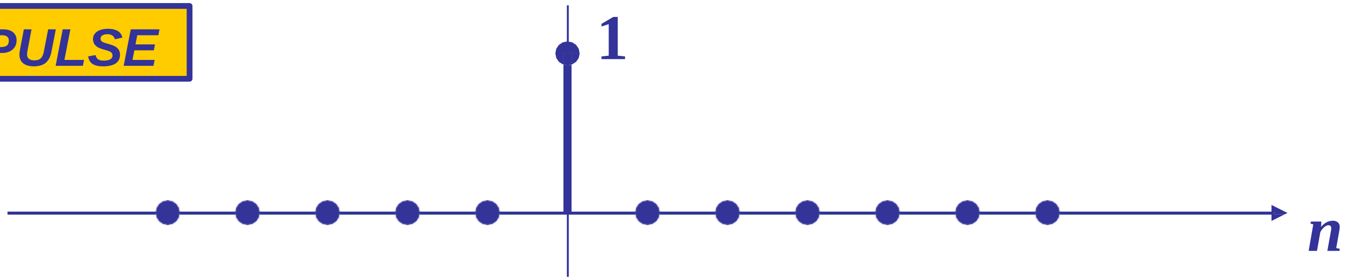
FREQUENCY RESPONSE (LATER)

- x[n] has only one NON-ZERO VALUE

Test Signal

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

UNIT-IMPULSE



UNIT IMPULSE SIGNAL δ[n]

n	$\ldots$	-2	-1	0	1	2	3	4	5	6	$\ldots$
$\delta[n]$	0	0	0	$\boxed{1}$	0	0	0	0	0	0	0
$\delta[n-3]$	0	0	0	0	0	0	$\boxed{1}$	0	0	0	0

$n = 3$

δ[n] is NON-ZERO When its argument is equal to ZERO

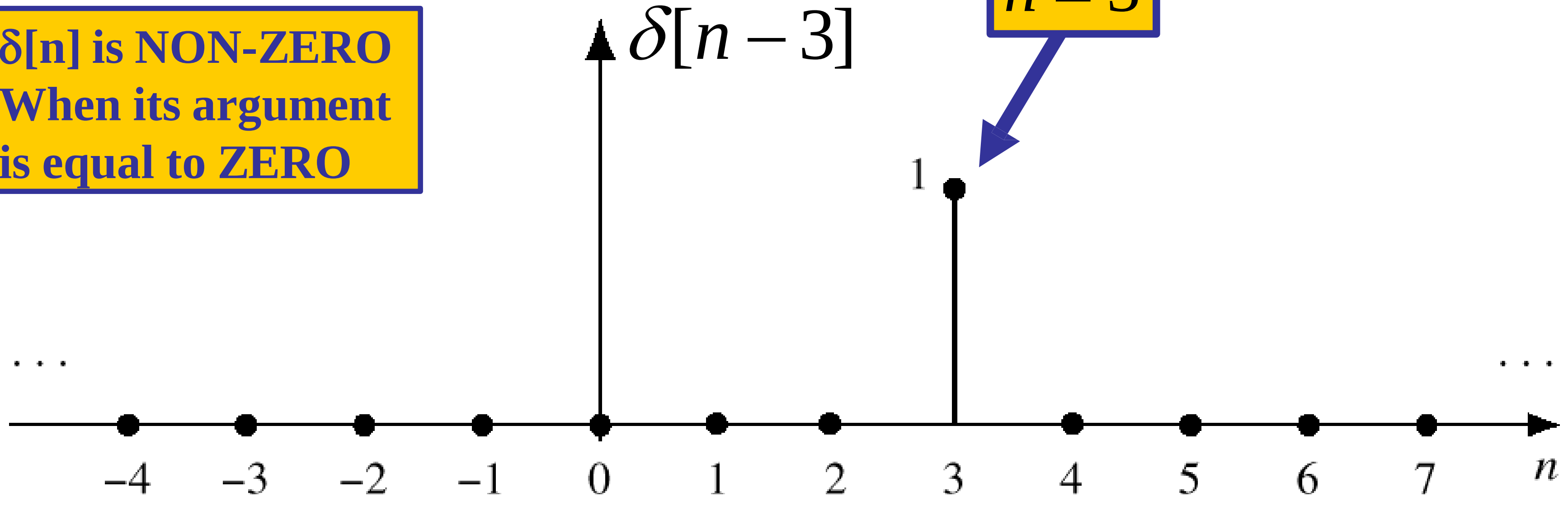
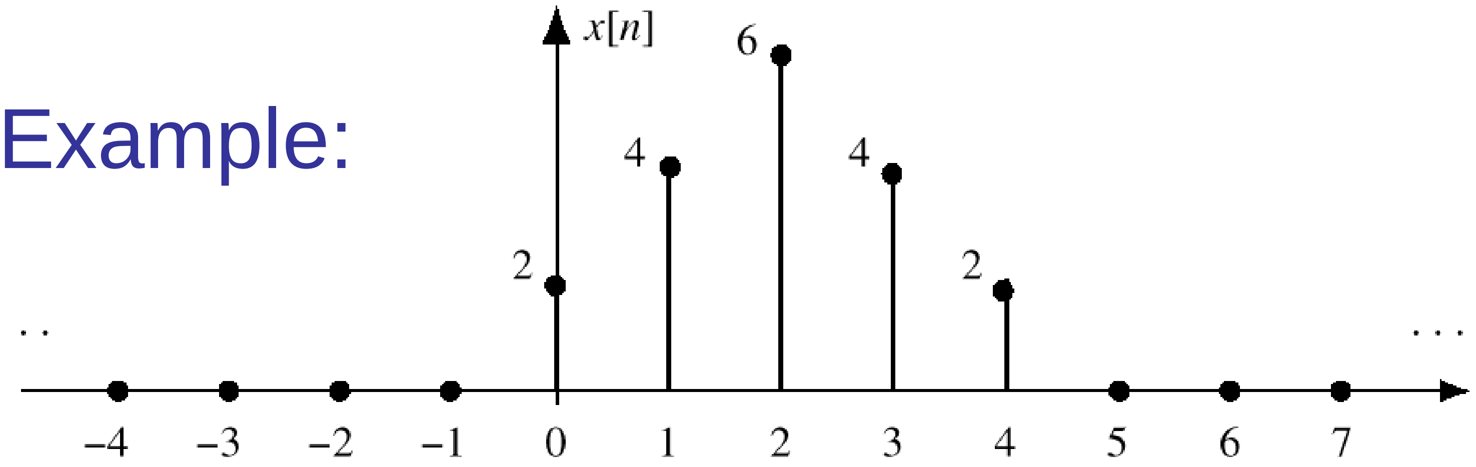


Figure 5.7 Shifted impulse sequence, $\delta[n-3]$.

Sequence Representation

Example:



$$x[n=0]=x[0]=2 \qquad x[n=1]=x[1]=4$$

$$x[n=2]=x[2]=6 \qquad x[n=3]=x[3]=4$$

$$x[n]=\cdots+\ \mathbf{0}\ \delta[n+1]+\boxed{\mathbf{2}}\,\delta[n]+\ \mathbf{4}\ \delta[n-1]$$
$$+\ \mathbf{6}\ \delta[n-2]+\boxed{\mathbf{4}}\,\delta[n-3]+\cdots$$

GIVES US A MATH FORM

UNIT IMPULSE RESPONSE

- FIR filter description usually given in terms of coefficients b_k

$$\boxed{y[n] = \sum_{k=0}^{M} b_k x[n-k]}$$

- Can we describe the filter using a **SIGNAL** instead?
- What happens if input is a unit impulse?

Example: 4-pt AVERAGER

- CAUSAL SYSTEM: USE PAST VALUES

$$y[n] = \tfrac{1}{4}(x[n] + x[n-1] + x[n-2] + x[n-3])$$

- INPUT = UNIT IMPULSE SIGNAL = $\delta[n]$

$$x[n] = \delta[n]$$
$$y[n] = \frac{1}{4}\delta[n] + \frac{1}{4}\delta[n-1] + \frac{1}{4}\delta[n-2] + \frac{1}{4}\delta[n-3]$$

- OUTPUT is called **"IMPULSE RESPONSE"**
  - Denoted $h[n]=y[n]$ when $x[n]=\delta[n]$

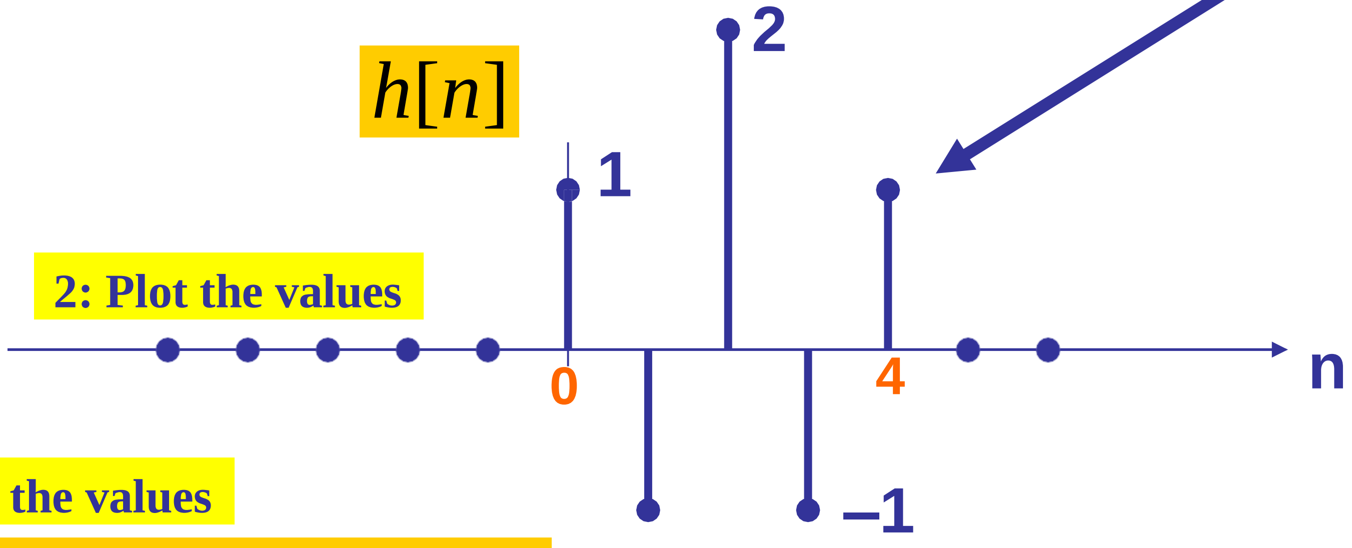
3 Ways to Represent the FIR filter

1 Use SHIFTED IMPULSES to write $h[n]$

$$h[n] = \delta[n] - \delta[n-1] + 2\delta[n-2] - \delta[n-3] + \delta[n-4]$$

$h[n]$

2: Plot the values



3: List the values

$$b_k = \{\, 1, -1, 2, -1, 1 \,\}$$

True for any signal, x[n]

MODIFIED TLH

DSP First, 2/e

Lecture 12

Convolution

Linearity & Time-Invariance

OVERVIEW

- IMPULSE RESPONSE, $h[n]$
  - FIR case: same as $\{b_k\}$
- CONVOLUTION
  - GENERAL: $y[n] = h[n] * x[n]$
  - GENERAL CLASS of SYSTEMS
  - LINEAR and TIME-INVARIANT
- ALL **LTI** systems have $\boldsymbol{h}[\boldsymbol{n}]$ & use convolution

LTI: Convolution Sum

- **Output = Convolution of x[n] & h[n]**
  - NOTATION:
  - FIR case:

$$y[n] = h[n] * x[n]$$

FINITE LIMITS

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

Same as $\mathbf{b_k}$

FINITE LIMITS

GENERAL FIR FILTER

- FILTER COEFFICIENTS $\{b_k\}$
  - DEFINE THE FILTER

$$y[n] = \sum_{k=0}^{M} b_k x[n-k]$$

  - For example,

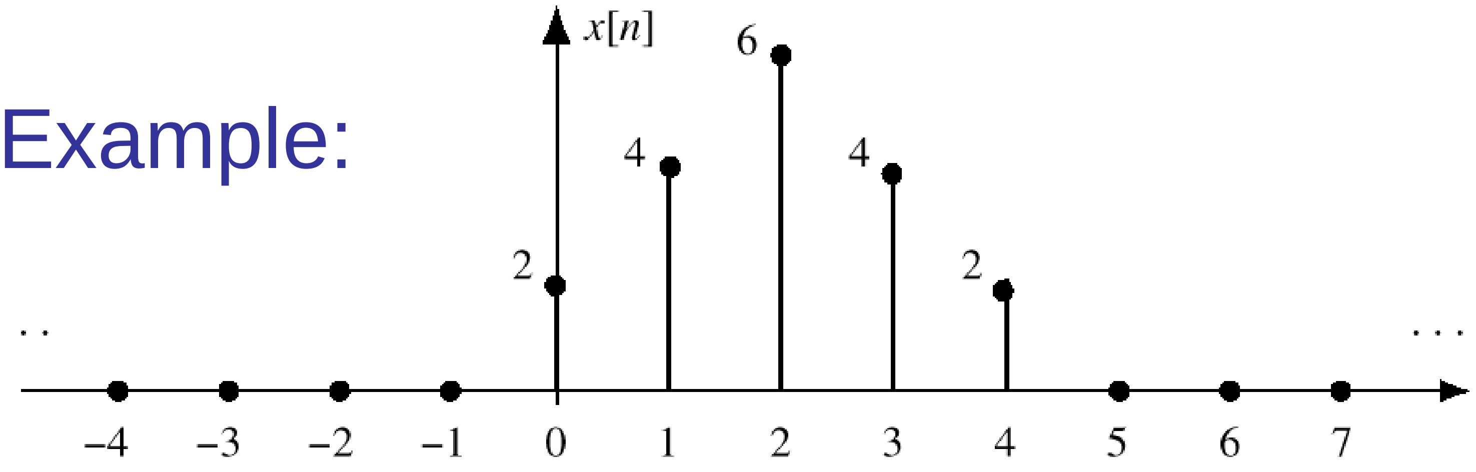
$$\boxed{b_k = \{3, -1, 2, 1\}}$$

$$y[n] = \sum_{k=0}^{3} b_k x[n-k]$$

$$= 3x[n] - x[n-1] + 2x[n-2] + x[n-3]$$

Sequence Representation

Example:



$$x[n=0]=x[0]=2 \qquad x[n=1]=x[1]=4$$

$$x[n=2]=x[2]=6 \qquad x[n=3]=x[3]=4$$

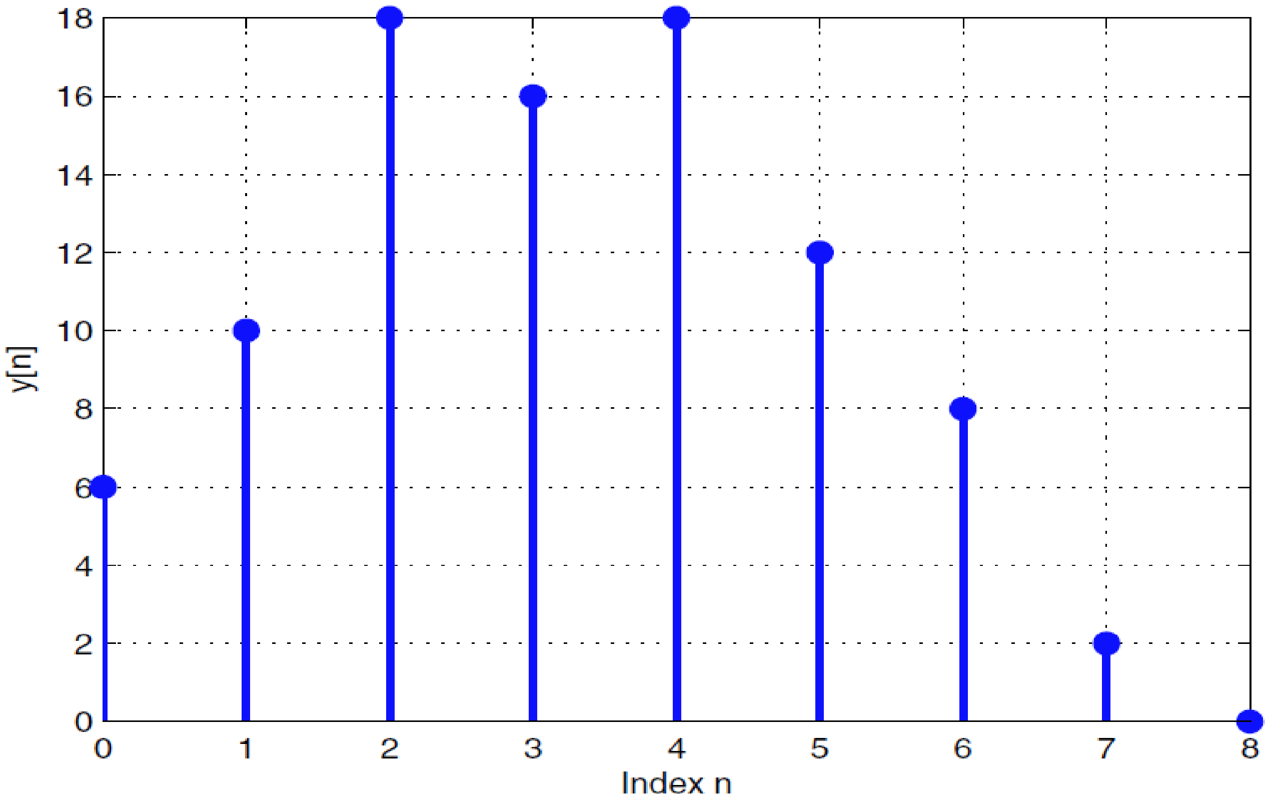
$$x[n]=\cdots+\ \mathbf{0}\ \delta[n+1]+\ \mathbf{2}\ \delta[n]+\ \mathbf{4}\ \delta[n-1]$$
$$+\ \mathbf{6}\ \delta[n-2]+\ \mathbf{4}\ \delta[n-3]+\cdots$$

GIVES US A MATH FORM

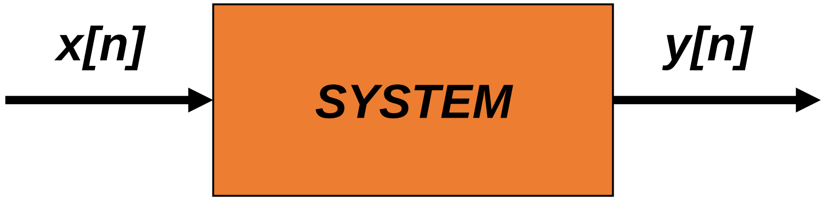
- We can check the answers using MATLAB's filter function

```
>> n = 0:8;
>> x = [2 4 6 4 2 0 0 0 0];
>> h = [3 -1 2 1];
>> y = filter(h,1,x);
>> y

y = 6    10    18    16    18    12    8    2    0
```



SYSTEM PROPERTIES



- MATHEMATICAL DESCRIPTION
- TIME-INVARIANCE
- LINEARITY
- CAUSALITY
  - “No output prior to input”

TIME-INVARIANCE

- IDEA:
  - “Time-Shifting the input will cause the same time-shift in the output”

- EQUIVALENTLY,
  - We can prove that
    - The time origin (n=0) is picked arbitrary

LINEAR SYSTEM

- LINEARITY = Two Properties
- SCALING
  - “Doubling x[n] will double y[n]”
- SUPERPOSITION:
  - “Adding two inputs gives an output that is the sum of the individual outputs”

LTI SYSTEMS

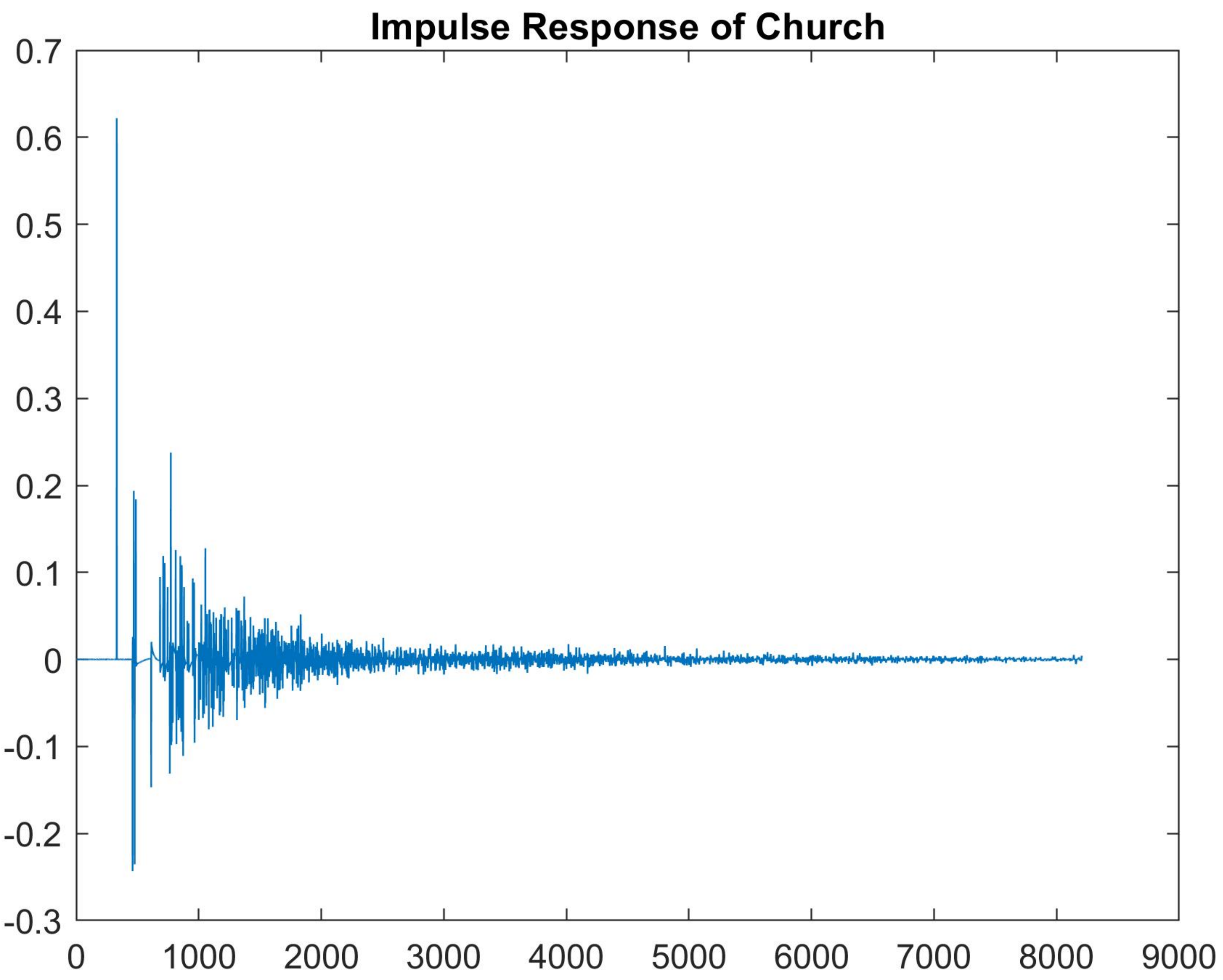
$$y[n] = h[n] * x[n]$$

- LTI: **L**inear & **T**ime-**I**nvariant
- COMPLETELY CHARACTERIZED by:
  - **IMPULSE RESPONSE** h[n]
  - **CONVOLUTION**:
    - The “rule” defining the system can ALWAYS be re-written as convolution
- FIR Example: h[n] is same as b_k

PRACTICAL EXAMPLE OF CONVOLUTION TO CHANGE YOUR ENVIRONMENT

```
%  DorranInChurch_2021_TLHdemo.m
clc, clear all, clf
church = audioread('church.wav')
audioinfo('church.wav')
% NumChannels:1 SampleRate:16000;TotalSamples:8206
%     Duration: 0.5129  sec
Fschurch= 16000;    % Samples/sec
%  Plot
figure(1), plot(church), title('Impulse Response of 
Church')
sound(church, 16000)
%
churchlen= length(church); % churchlen = 8206 points
ts=1/Fschurch    % ts = 6.2500e-05 sec
t_duration = length(church)/Fschurch % t_duration 
=0.5129
```

IMPULSE RESPONSE OF CHURCH


Impulse Response of Church
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0
1000
2000
3000
4000
5000
6000
7000
8000
9000

```
disp('Start speaking for 10 seconds.')
record_voice = audiorecorder(16000, 16, 1);
%disp('Start speaking for 10 seconds.')
recordblocking(record_voice,10);
disp('End of Recording.');
pause(2)
p = play(record_voice);    % listen to complete recording
pause(10)
mySpeech = getaudiodata(record_voice, 'int16'); % get data
as int16 array
%
disp('Speaking in Church')
pause(2)
output=conv(mySpeech,church);
soundsc(output, 16000);
```

SPEAK UP