PENGOLAHAN SINYAL DIGITAL

Adhi Harmoko Saputro

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia

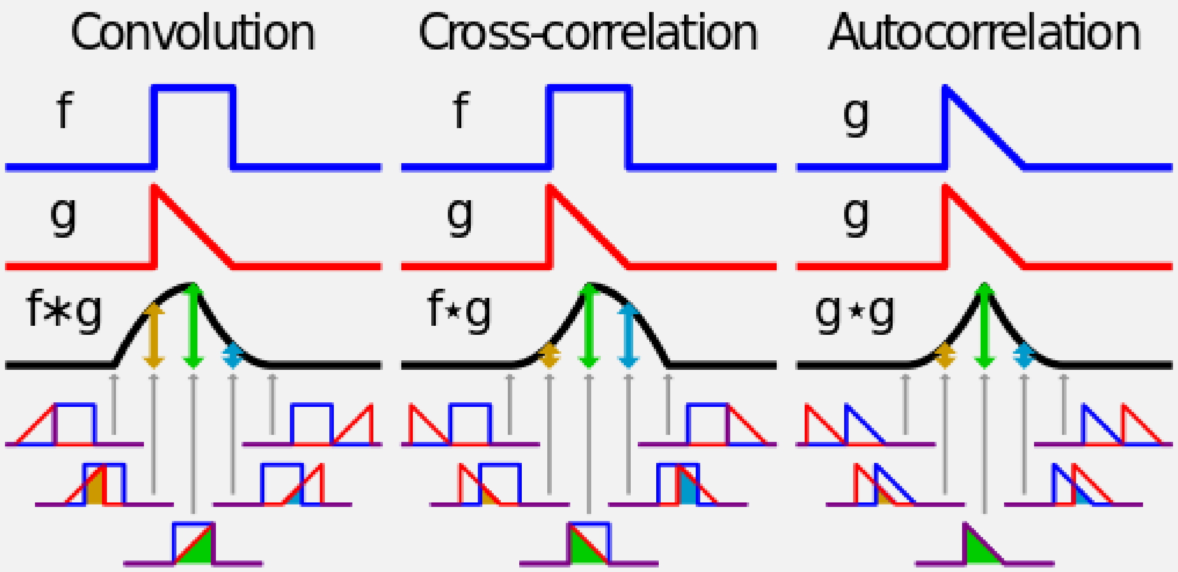
CONVOLUTION

Adhi Harmoko Saputro

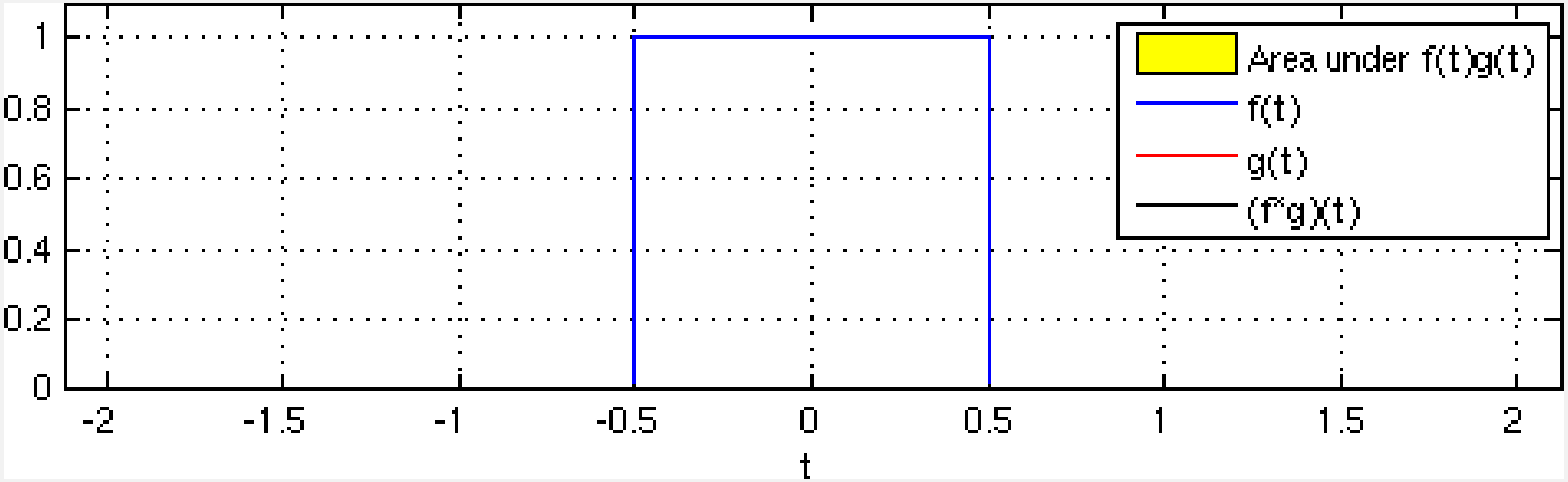
UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia

CONVOLUTION

- Convolution is a mathematical operation on two functions (f and g)
- Produces a third function, that is typically viewed as a modified version of one of the original functions, giving the integral of the pointwise multiplication of the two functions as a function of the amount that one of the original functions is translated.



CONVOLUTION



CONVOLUTION

- The impulse response of an LTI system is given by $h[n]$

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

- The mathematical operation is called a *linear convolution sum* and is denoted by

$$y[n] \triangleq x[n] * h[n]$$

- If the sequences are mathematical functions (of finite or infinite duration), then we can analytically evaluate for all n to obtain a functional form of $y[n]$.

CONVOLUTION

- The "*n*" dependency of *y*[*n*] deserves some care: for each value of "*n*" the convolution sum must be computed ***separately*** over all values of a dummy variable "*m*". So, for each "*n*"
  1. Rename the independent variable as **m**. You now have **x[m]** and **h[m]**. Flip **h[m]** over the origin. This is **h[-m]**
  2. Shift **h[-m]** as far left as possible to a point "**n**", where the two signals barely touch. This is **h[n-m]**
  3. Multiply the two signals and sum over all values of **m**. This is the convolution sum for the specific "**n**" picked above.
  4. Shift / move **h[-m]** to the right by one sample, and obtain a new **h[n-m]**. Multiply and sum over all **m**.
  5. Repeat 2~4 until **h[n-m]** no longer overlaps with **x[m]**, i.e., shifted out of the **x[m]** zone.

CONVOLUTION

- The "*n*" dependency of *y*[*n*] deserves some care: for each value of "*n*" the convolution sum must be computed ***separately*** over all values of a dummy variable "*m*". So, for each "*n*"
  1. Rename the independent variable as **m**. You now have **x[m]** and **h[m]**. Flip **h[m]** over the origin. This is **h[-m]**
  2. Shift **h[-m]** as far left as possible to a point "**n**", where the two signals barely touch. This is **h[n-m]**
  3. Multiply the two signals and sum over all values of **m**. This is the convolution sum for the specific "**n**" picked above.
  4. Shift / move **h[-m]** to the right by one sample, and obtain a new **h[n-m]**. Multiply and sum over all **m**.
  5. Repeat 2~4 until **h[n-m]** no longer overlaps with **x[m]**, i.e., shifted out of the **x[m]** zone.

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

USEFUL EXPRESSIONS

- The following expressions are often useful in calculating convolutions of analytical discrete signals

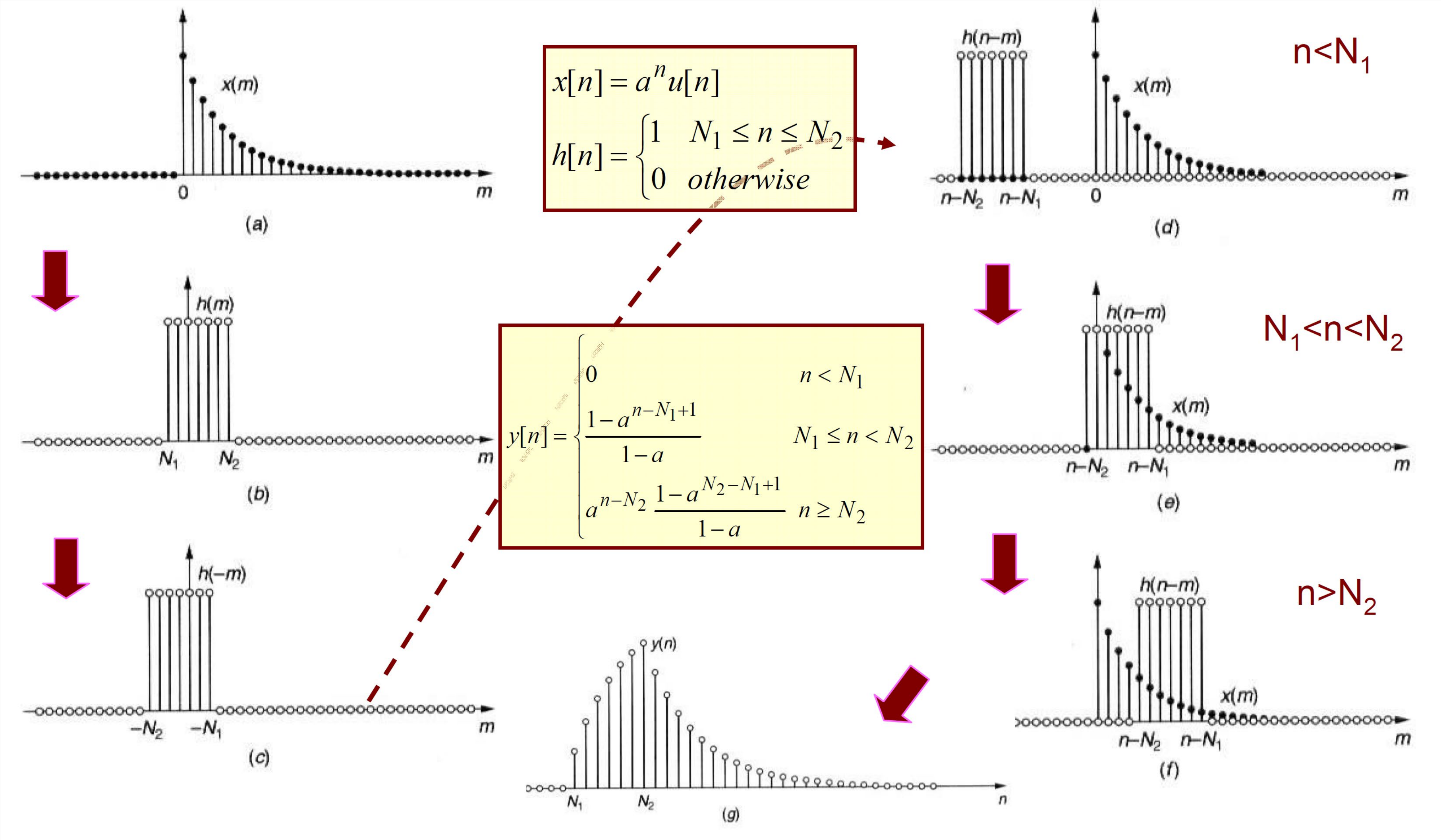
$$\sum_{n=0}^{\infty} a^n = \frac{1}{1-a}, \quad |a| < 1$$

$$\sum_{n=k}^{\infty} a^n = \frac{a^k}{1-a}, \quad |a| < 1$$

$$\sum_{n=m}^{N} a^n = \frac{a^m - a^{N+1}}{1-a}, \quad a \neq 1$$

$$\sum_{n=0}^{N-1} a^n = \begin{cases} \dfrac{1-a^N}{1-a}, & |a| \neq 1 \\ N, & \text{a=1} \end{cases}$$

CONVOLUTION EXAMPLE



EXAMPLE 1

- Let the rectangular pulse $x(n) = u(n) - u(n - 10)$ be an input to an LTI system with impulse response

$$h(n) = (0.9)^n u(n)$$

- Determine the output $y(n)$.

EXAMPLE 1

- From the convolution equation

$$y(n) = x(n) * h(n)$$

$$y(n) = \sum_{k=0}^{9} (1)(0.9)^{(n-k)} u(n-k) = (0.9)^{n} \sum_{k=0}^{9} (0.9)^{-k} u(n-k)$$

- The sum in equation is almost a geometric series sum except that the term $u(n-k)$ takes different values depending on *n* and *k*.
- There are three possible conditions under which $u(n-k)$ can be evaluated.

EXAMPLE 1

- Case 1
  - $n < 0$
  - Then $u(n - k) = 0$, $0 \le k \le 9$

$$y(n) = 0$$

EXAMPLE 1

- Case 2
  - The nonzero values of $x(n)$ and $h(n)$ *do not overlap*.
  - $0 \le n < 9$: Then $u(n - k) = 1$, $0 \le k \le n$.

$$y(n) = (0.9)^n \sum_{k=0}^{n} (0.9)^{-k} = (0.9)^n \sum_{k=0}^{n} \left[(0.9)^{-1}\right]^k$$

$$= (0.9)^n \frac{1-(0.9)^{-(n+1)}}{1-(0.9)^{-1}} = 10\left[1-(0.9)^{(n+1)}\right]$$

EXAMPLE 1

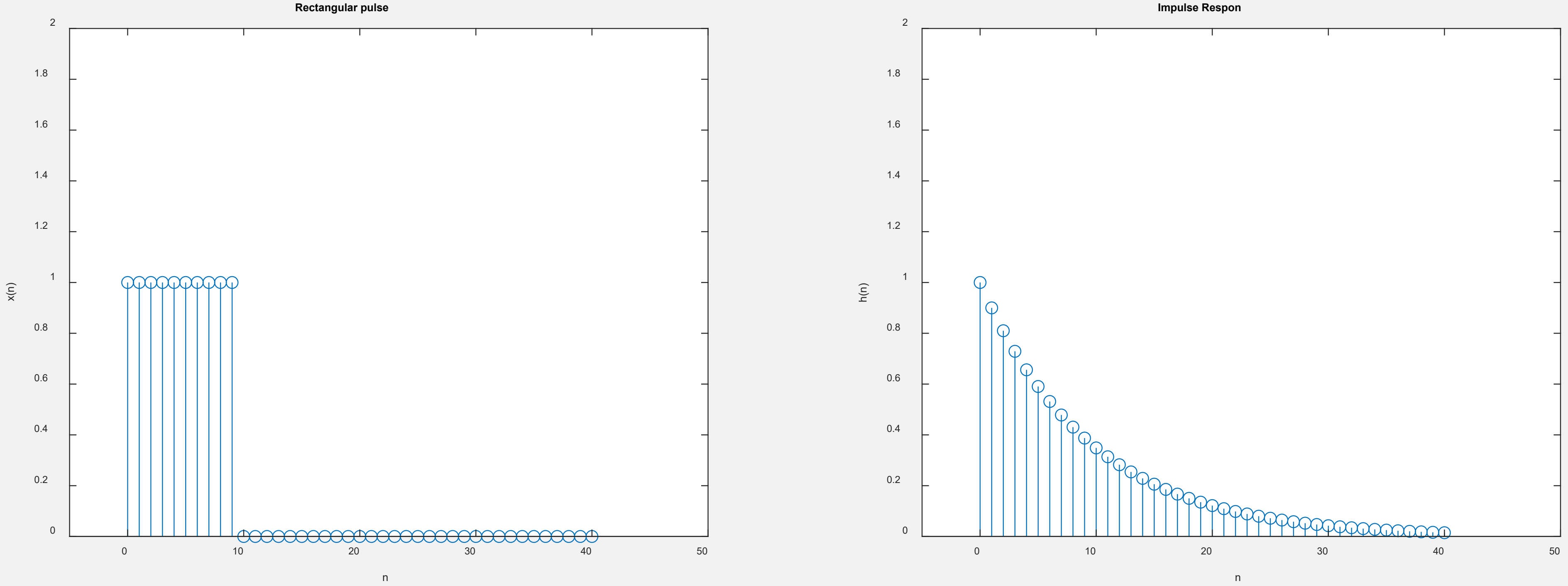
- Case 3
  - The impulse response $h(n)$ *partially overlaps* the input $x(n)$.
  - $n \geq 9$: Then $u(n - k) = 1,\ 0 \leq k \leq 9$

$$y(n) = (0.9)^n \sum_{k=0}^{9} (0.9)^{-k}$$

$$= (0.9)^n \frac{1-(0.9)^{-10}}{1-(0.9)^{-1}} = 10(0.9)^{n-9}\left[1-(0.9)^{10}\right]$$

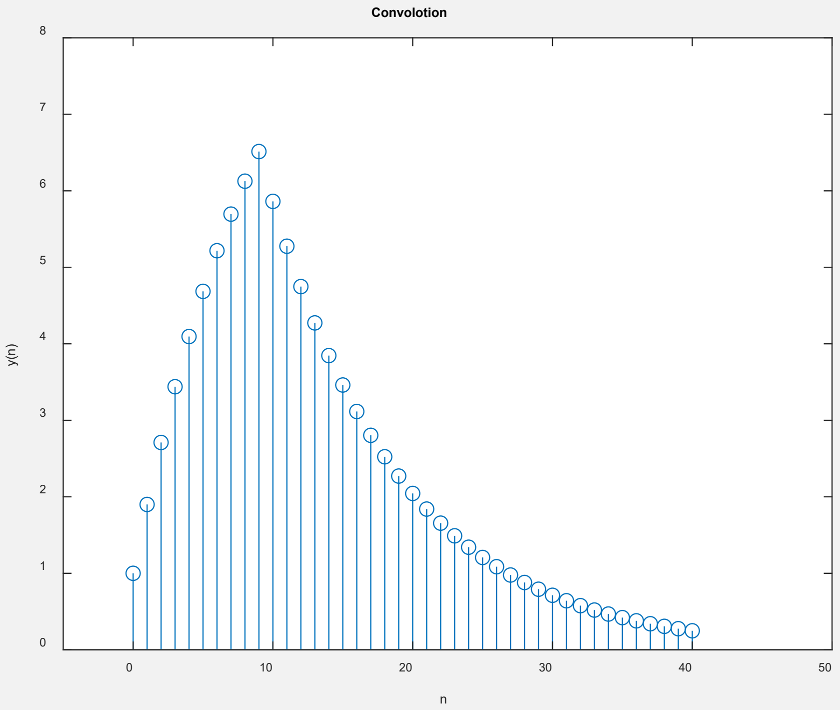
EXAMPLE 1

- The input sequence and the impulse response



EXAMPLE 1

- The output sequence



EXAMPLE 2

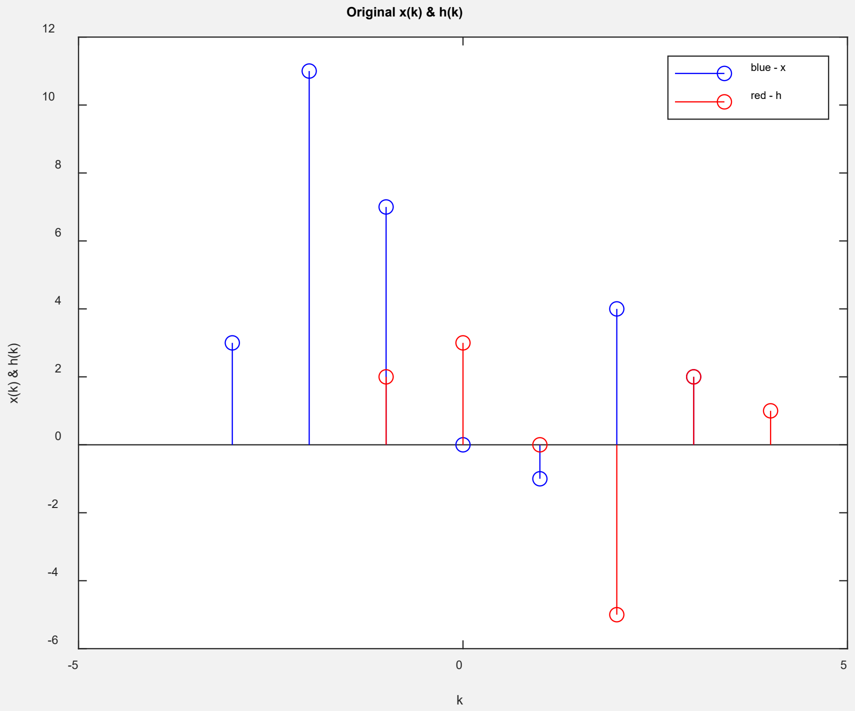
- Given the following two sequences

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

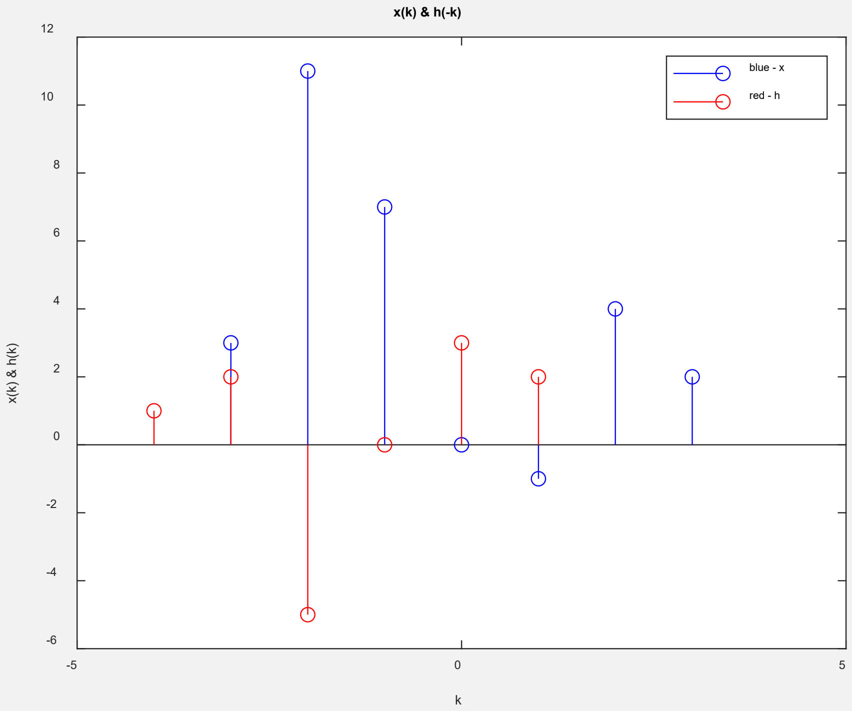
$$h(n) = \Big[2, \underset{\uparrow}{3}, 0, -5, 2, 1 \Big], \quad -1 \le n \le 4$$

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

EXAMPLE 2

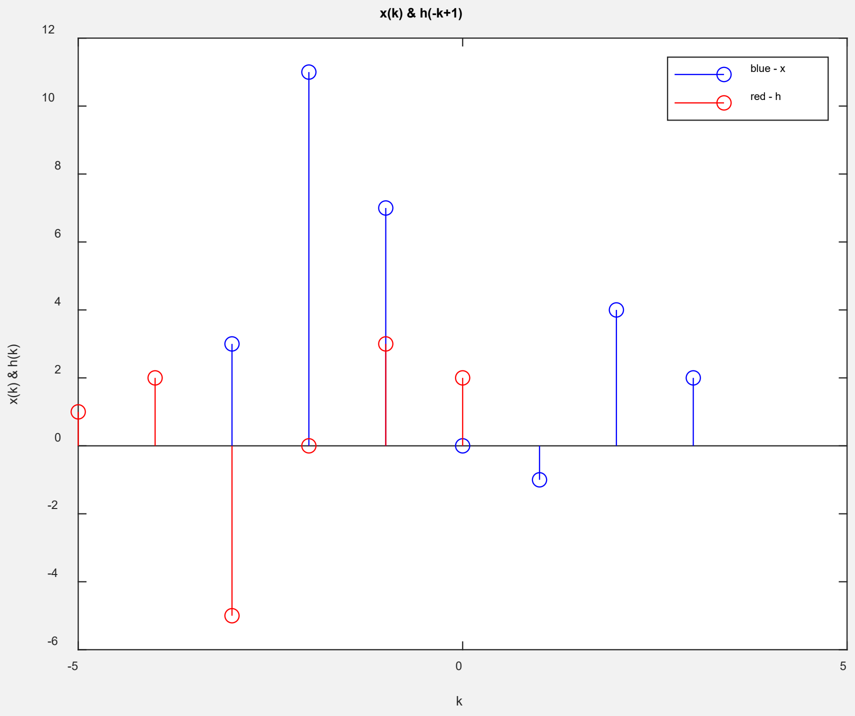


The original sequences



x(k) and h(−k), the folded version of h(k)

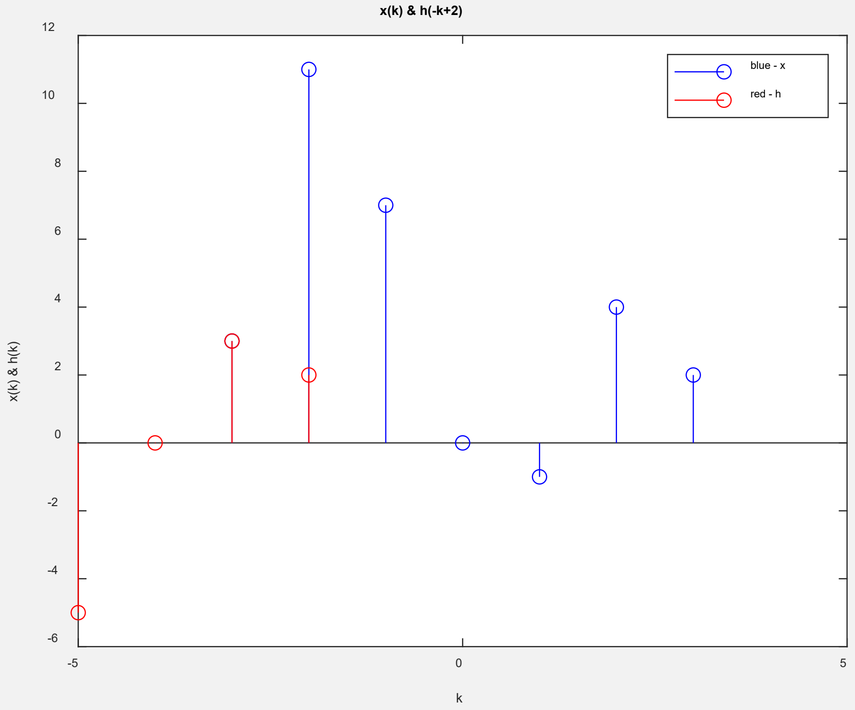
EXAMPLE 2



x(k) and h(−1−k), the folded-and-shifted by −1 version of h(k).

$$\sum_k x(k)h(-1-k) = 3\times(-5)+11\times 0+7\times 3+0\times 2 = 6 = y(-1)$$

EXAMPLE 2



x(k) and h(2 − k), the folded-and-shifted-by-2 version of h(k)

$$\sum_k x(k)h(2-k)=11\times1+7\times2+0\times(-5)+(-1)\times0+4\times3+2\times2=41=y(2)$$

EXAMPLE 2

- Similar graphical calculations can be done for other remaining values of $y(n)$.
- The beginning point (first nonzero sample) of $y(n)$ is given by

$$n = -3 + (-1) = -4$$

- The end point (the last nonzero sample) is given by

$$n = 3 + 4 = 7$$

- The complete output is given by

$$y(n) = \left\{6, 31, 47, 6, -\underset{\uparrow}{5}1, -5, 41, 18, -22, -3, 8, 2\right\}$$

MATLAB CONVOLUTION

- 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 = 0 and is invoked by

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

EXAMPLE 3

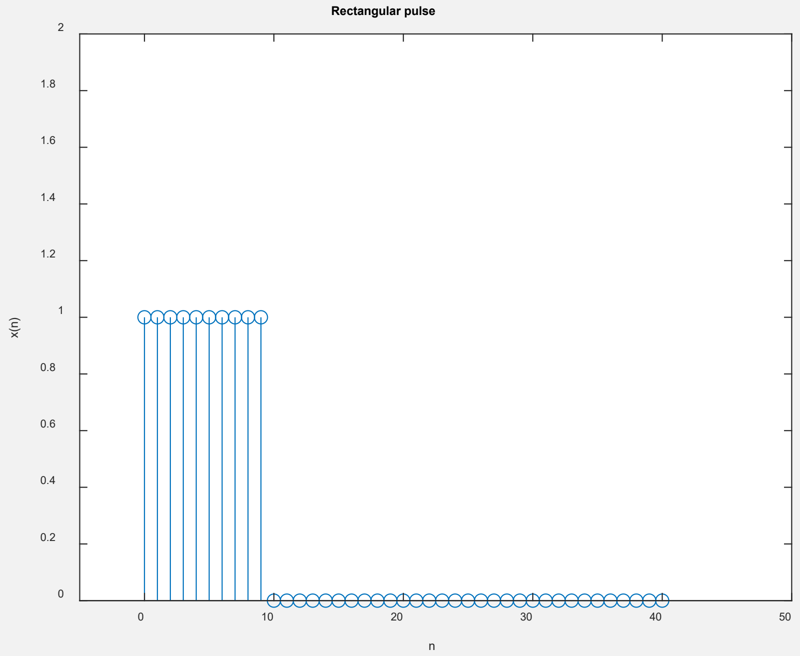
- Let the rectangular pulse $x(n) = u(n) - u(n - 10)$ be an input to an LTI system with impulse response

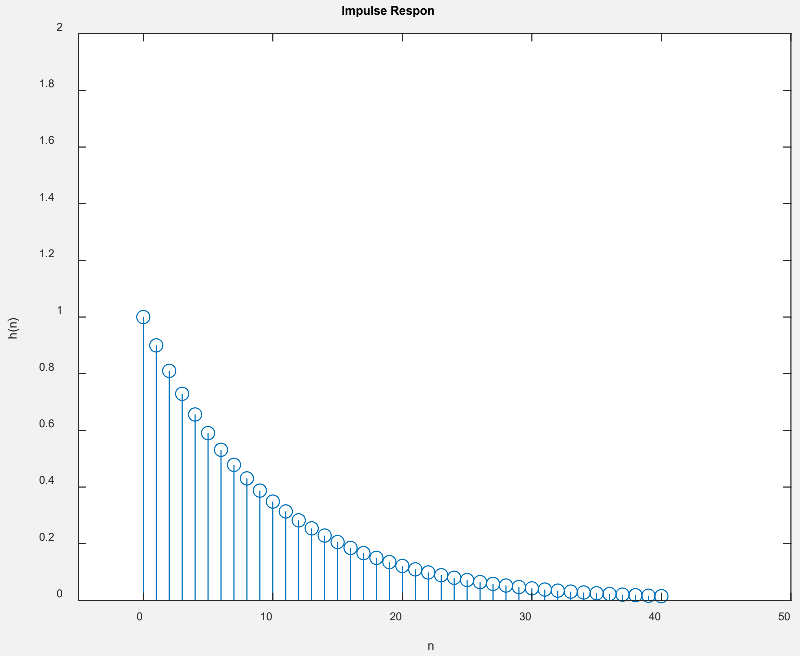
$$h(n) = (0.9)^n u(n)$$

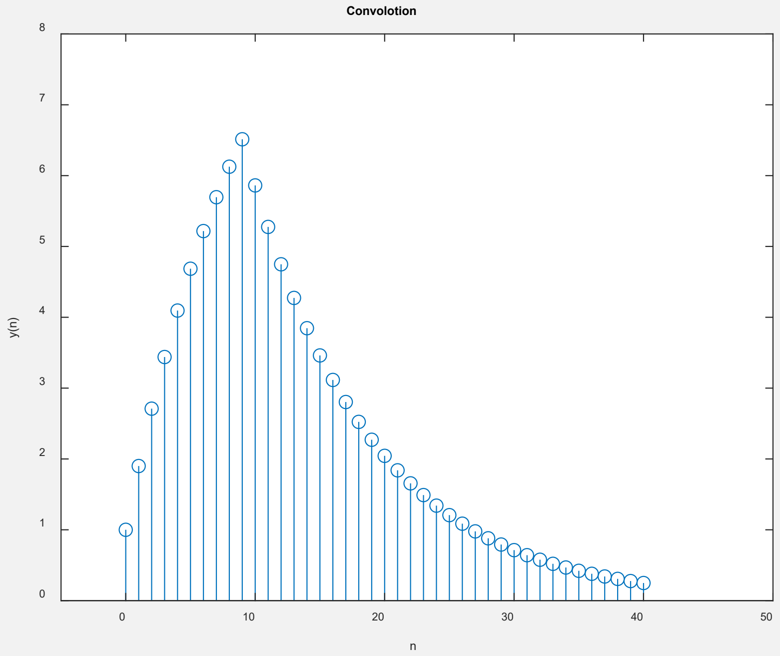
- Determine the output $y(n)$.

```
[x1,n] = stepseq(0,0,40);
[x2,n] = stepseq(10,0,40);
x = x1-x2;
h = (0.9.^n).*stepseq(0,0,40);
y = conv(x,h);
subplot(311); stem(n,x);
subplot(312); stem(n,h);
subplot(313); stem(n,y(1:length(n)));
```

EXAMPLE 3


Rectangular pulse
x(n)
n
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50


Impulse Respon
h(n)
n
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50


Convolotion
y(n)
n
8
7
6
5
4
3
2
1
0
0
10
20
30
40
50

MATLAB CONVOLUTION

- A simple modification of the conv function, which performs the convolution of arbitrary support sequences

```
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);
```

EXAMPLE 4

- Given the following two sequences

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

$$h(n) = \left[2, \underset{\uparrow}{3}, 0, -5, 2, 1\right], \quad -1 \le n \le 4$$

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

EXAMPLE 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)
y =
6 31 47 6 -51 -5 41 18 -22 -3 8 2
ny =
-4 -3 -2 -1 0 1 2 3 4 5 6 7
```

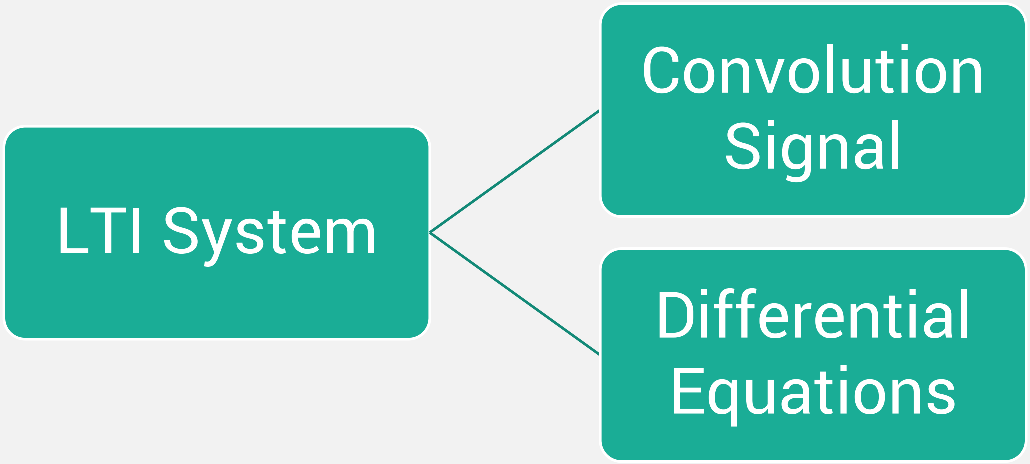
$$y(n) = \left\{6, 31, 47, 6, \underset{\uparrow}{-51}, -5, 41, 18, -22, -3, 8, 2\right\}$$

DIFFERENTIAL EQUATIONS

Adhi Harmoko Saputro

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia

LTI SYSTEM



$$y[n] \triangleq x[n] * h[n]$$

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

a recursive approach for computing the current output, given the input values and previously computed output values

LINEAR CONSTANT-COEFFICIENT DIFFERENCE EQUATIONS

- In general, an N^{th}-order linear constant coefficient difference equation has the form

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

- The output is not uniquely specified for a given input
  - The initial conditions are required
  - Linearity, time invariance, and causality depend on the initial conditions
  - If initial conditions are assumed to be zero, system is linear, time invariant, and causal

LINEAR CONSTANT-COEFFICIENT DIFFERENCE EQUATIONS

- Example
  - Moving Average

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

  - Difference Equation Representation

$$\sum_{k=0}^{0} a_k y[n-k] = \sum_{k=0}^{3} b_k x[n-k] \quad \text{where} \quad a_k = b_k = 1$$

RECURSIVE SOLUTION

- In the discrete-time case, there are an alternative to find the Differential Equations

$$y[n]=\frac{1}{a_o}\left(\sum_{k=0}^{M}b_x x[n-k]-\sum_{k=1}^{N}a_k y[n-k]\right)$$

- This is often how digital filters are implemented on a computer or a digital signal processor board
- The response of a differential equation
    - The differential equation is discretized at a given sampling rate to obtain a difference equation
    - The response of the difference equation is computed recursively

GENERAL SOLUTION

- A general solution can be expressed as the sum of a homogeneous response (natural response), and a particular solution (forced response) of the system:

$$y[n] = y_h[n] + y_p[n]$$

- The concept of *initial rest* of the LTI causal system described by the difference equation here means that $x[n] = 0$ implies $y[n] = 0$

GENERAL SOLUTION

- The *homogeneous part* of the solution

$$y_h[n] = \sum_{k=1}^{N} c_k z_k^n$$

z_k, $k = 1, . . . , N$ are N roots (also called *natural frequencies*) of the characteristic equation

$$\sum_{k=1}^{N} a_k z^k = 0$$

- important in determining the stability of systems

GENERAL SOLUTION

- The *particular part* of the solution is determined from the right-hand side

$x[n]$	$y_p[n]$
A	K
$A.M^n$	$K.M^n$
$A.n^M$	$K_0n^M + K_1n^{M-1} + K_2n^{M-2} + \ldots + K_M$
$A^n.n^M$	$A^n (K_0n^M + K_1n^{M-1} + K_2n^{M-2} + \ldots + K_M)$

EXAMPLE 5

- Consider the difference equation

$$y[n] - 3y[n-1] - 4y[n-2] = 0$$

- Find system response $y[n]$ for $y[-1] = 5$ and $y[-2] = 0$!

EXAMPLE 5

- Solution: Assumption → homogen solution

$$y_h[n] = z^n$$

- Subsitution to original equation

$$z^n - 3z^{n-1} - 4z^{n-2} = 0$$

$$z^{n-2}(z^2 - 3z - 4) = 0$$

$$(z + 1)(z - 4) = 0$$

$$z = -1 \text{ and } z = 4$$

EXAMPLE 5

- Solution

$$y[n] = y_h[n] + y_p[n]$$

$$y[n] = a_1(-1)^n + a_2(4)^n$$

- Finding a_1 and a_2 to fulfil the solution

EXAMPLE 5

- Finding a_1 and a_2 to fulfil the solution

$$y[n] - 3y[n-1] - 4y[n-2] = 0$$

$$y[n] = 3y[n-1] + 4y[n-2]$$

$$y[0] = 3y[-1] + 4y[-2]$$

$$y[1] = 3y[0] + 4y[-1]$$

$$= 3y\{3y[-1] + 4y[-2]\} + 4y[-1]$$

$$= 13y[-1] + 12y[-2]$$

$$y_h[n] = a_1(-1)^n + a_2(4)^n$$

$$y_h[0] = a_1 + a_2$$

$$y_h[1] = -a_1 + 4a_2$$

EXAMPLE 5

- Input the initial value

$$y[0] = 15$$
$$y[1] = 65$$

$$15 = a_1 + a_2 \qquad a_1 = -1$$
$$65 = -a_1 + 4a_2 \qquad a_2 = 16$$

- Solution

$$y[\mathrm{n}] = -1(-1)^n + 16(4)^n \quad n \geq 0$$

EXAMPLE 5

- Consider the difference equation

$$y[n]-\frac{5}{6}y[n-1]+\frac{1}{6}y[n-2]=3x[n]-2x[n-1]$$

- Find the recursive form of the difference equation
- Assuming initial rest and that the input is an impulse,
  - $y[-2]=y[-1]=0$

EXAMPLE 5

$$y[n] = \frac{5}{6} y[n-1] - \frac{1}{6} y[n-2] + 3x[n] - 2x[n-1]$$

- the recursion can be started

$$y[0] = \frac{5}{6} y[-1] - \frac{1}{6} y[-2] + 3x[0] - 2x[-1]$$
$$= \frac{5}{6}(0) - \frac{1}{6}(0) + 3(1) - 2(0) = 3$$

$$y[1] = \frac{5}{6} y[0] - \frac{1}{6} y[-1] + 3x[1] - 2x[0]$$
$$= \frac{5}{6}(3) - \frac{1}{6}(0) + 3(0) - 2(1) = \frac{1}{2}$$

EXAMPLE 5

$$
\begin{aligned}
y[2] &= \frac{5}{6}y[1]-\frac{1}{6}y[0]+3x[2]-2x[1] \\
&= \frac{5}{6}\left(\frac{1}{2}\right)-\frac{1}{6}(3)+3(0)-2(0)=-\frac{1}{12}
\end{aligned}
$$

MATLAB IMPLEMANTATION

- A function called filter is available to solve difference equations numerically, given the input and the difference equation coefficients.

- In its simplest form this function is invoked by

```
y = filter(b,a,x)
```

  - where
    - b = [b0, b1, ..., bM]; a = [a0, a1, ..., aN] are the coefficient arrays

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

MATLAB IMPLEMANTATION

- To compute and plot impulse response, MATLAB provides the function impz.
- When invoked by

```
h = impz(b,a,n)
```

- Computes samples of the impulse response of the filter at the sample indices given in n with numerator coefficients in b and denominator coefficients in a.

EXAMPLE 6

- Given the following difference equation

$$y(n) - y(n-1) + 0.9y(n-2) = x(n);$$

- Calculate and plot the impulse response $h(n)$ at $n = -20, \ldots, 100$.
- Calculate and plot the unit step response $s(n)$ at $n = -20, \ldots, 100$.
- Is the system specified by $h(n)$ stable?

EXAMPLE 6

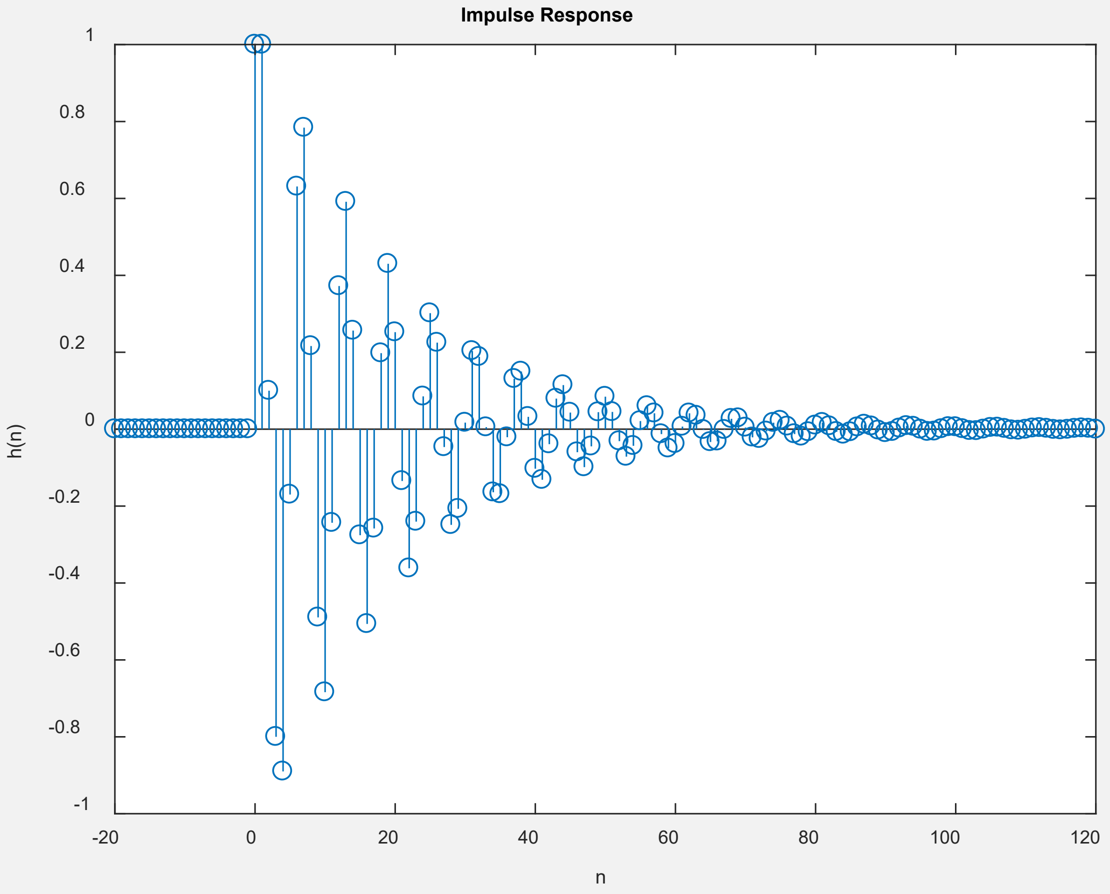
- From the given difference equation the coefficient arrays are
  - b = [1]; a=[1, -1, 0.9];

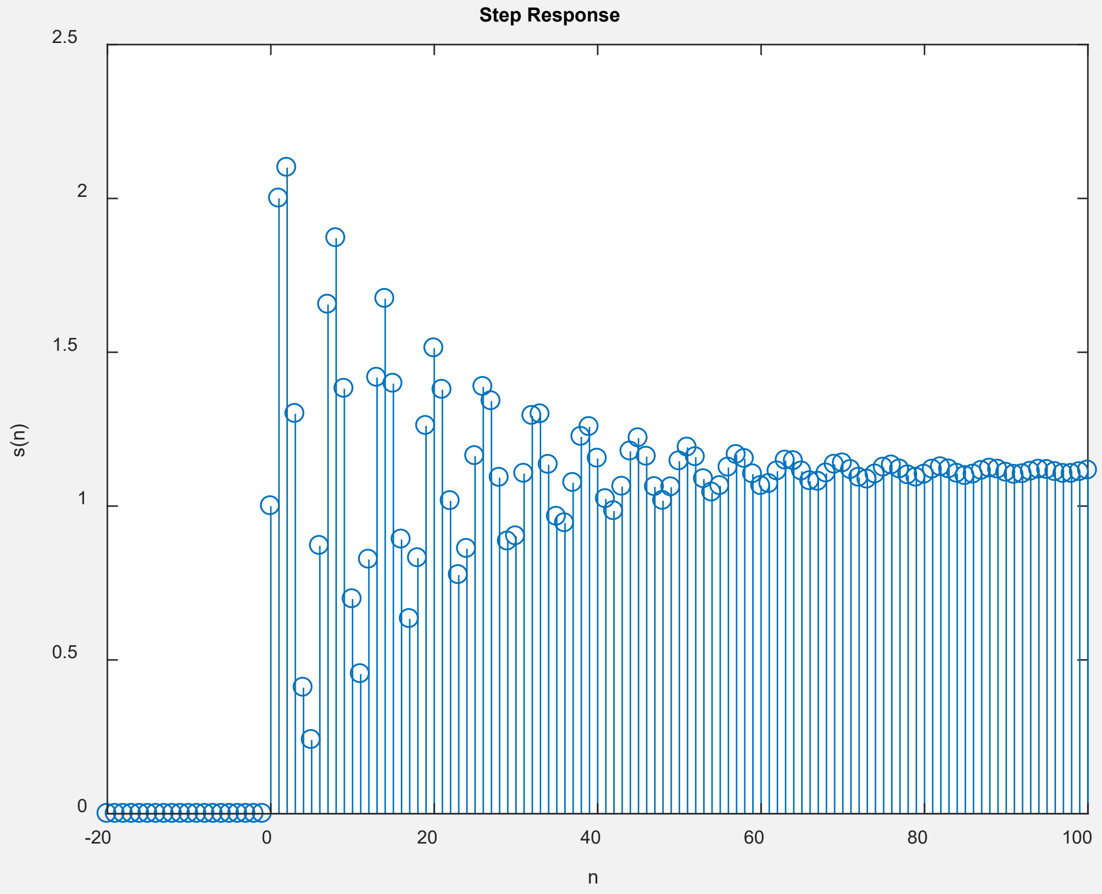
```
>> b = [1]; a = [1, -1, 0.9]; n = [-20:100];
>> h = impz(b,a,n);
>> subplot(2,1,1); stem(n,h);
>> title('Impulse Response');
>> xlabel('n'); ylabel('h(n)')
```

EXAMPLE 6

```
>> n=-20:100;
>> x = (n-0) >= 0; % x = stepseq(0,-20,100);
>> s = filter(b,a,x);
>> subplot(2,1,2); stem(n,s)
>> title('Step Response');
>> xlabel('n'); ylabel('s(n)')
```

EXAMPLE 6


Impulse Response
h(n)
n
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-20
0
20
40
60
80
100
120


Step Response
s(n)
n
2.5
2
1.5
1
0.5
0
-20
0
20
40
60
80
100

EXAMPLE 6

- To determine the stability of the system, we have to determine $h(n)$ for all n.
- Although we have not described a method to solve the difference equation
- we can use the plot of the impulse response to observe that $h(n)$ is practically zero for $n > 120$. Hence the sum

$$\sum \left| h[n] \right|$$

```
>> sum(abs(h))
ans = 14.8785
```

which implies that the system is stable.

EXAMPLE 7

- Let us consider the convolution given in previous example. The input sequence is of finite duration $x(n) = u(n) - u(n-10)$
- while the impulse response is of infinite duration

$$h(n) = (0.9)^n u(n)$$

- Determine $y(n) = x(n) * h(n)$.

EXAMPLE 7

- If the LTI system, given by the impulse response *h*(*n*), can be described by a difference equation, then *y*(*n*) can be obtained from the filter function.
- From the *h*(*n*) expression

$$(0.9)h[n-1]=(0.9)(0.9)^{n-1}u[n-1]=(0.9)^{n}u[n-1]$$

$$\begin{aligned} h[n]-(0.9)h[n-1] &= (0.9)^{n}u[n]-(0.9)^{n}u[n-1] \\ &= (0.9)^{n}\left(u[n]-u[n-1]\right)=(0.9)^{n}\delta[n] \\ &= \delta[n] \end{aligned}$$

EXAMPLE 7

- The last step follows from the fact that $\delta(n)$ is nonzero only at $n = 0$.
- By definition $h(n)$ is the output of an LTI system when the input is $\delta(n)$.
- Hence substituting $x(n)$ for $\delta(n)$ and $y(n)$ for $h(n)$, the difference equation is

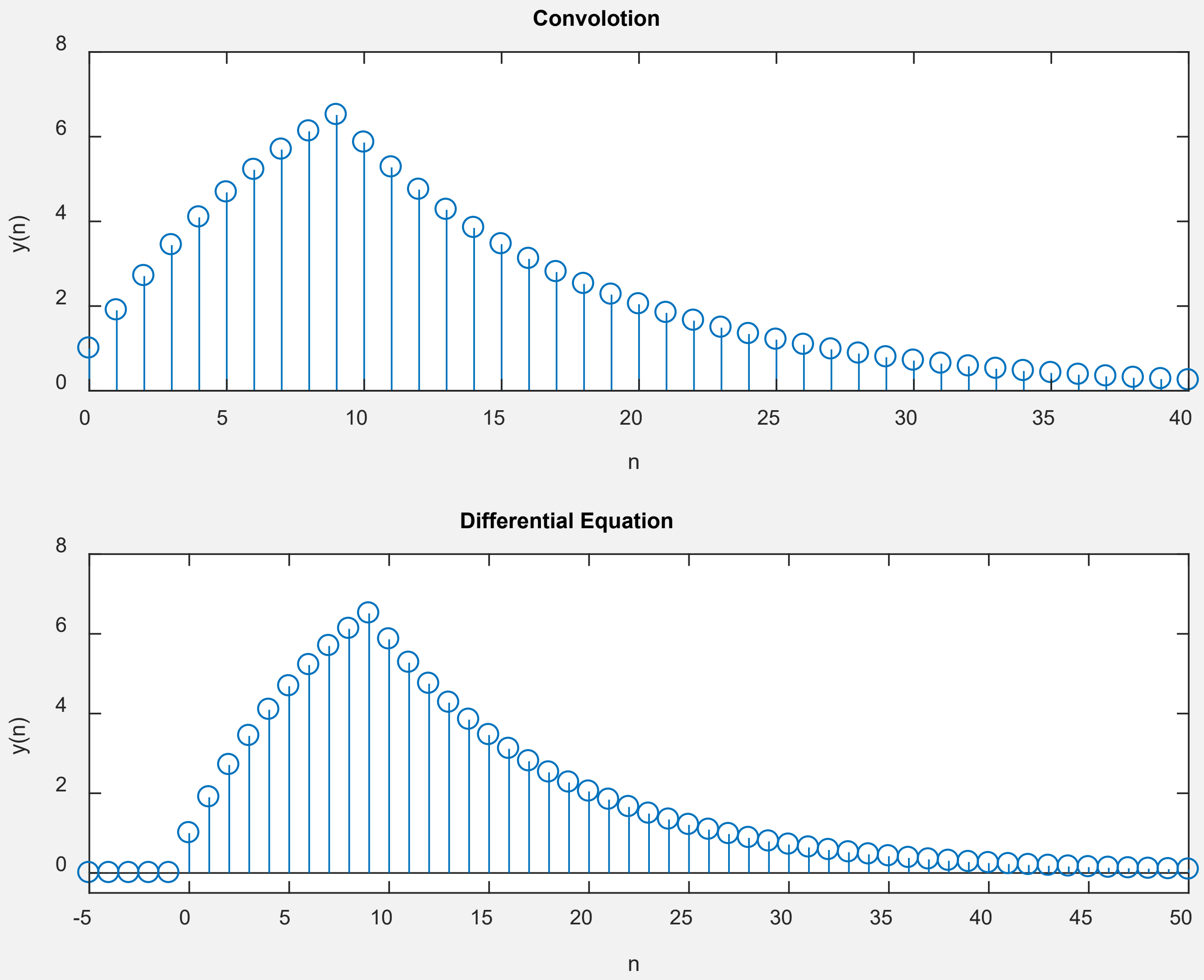
$$y[n]-(0.9)\,y[n-1]=x[n]$$

EXAMPLE 7

- MATLAB’s filter function can be used to compute the convolution indirectly.

```
>> b = [1]; a = [1,-0.9];
>> n = -5:50;
>> x = stepseq(0,-5,50) - stepseq(10,-5,50);
>> y = filter(b,a,x);
>> subplot(2,1,2); stem(n,y);
>> title('Output sequence')
>> xlabel('n'); ylabel('y(n)'); axis([-5,50,-0.5,8])
```

EXAMPLE 7



EXAMPLE 7

```
err = y1(1:40) - y2(1:40);
mean(err);

disp(['Error ' num2str(mean(err))]);
```

DIGITAL FILTERS

Adhi Harmoko Saputro

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia

DIGITAL FILTERS

- Two types of digital filters
  - ***FIR filter*** If the unit impulse response of an LTI system is of finite duration, then the system is called a *finite-duration impulse response* (or FIR) filter.

$$y[n] = \sum_{m=0}^{M} b_m x[n-m]$$

    - $h(n) = 0$ for $n < n1$ and for $n > n2$

  - ***IIR filter*** If the impulse response of an LTI system is of infinite duration, then the system is called an *infinite-duration impulse response* (or IIR) filter.

$$\sum_{k=0}^{N} a_k y[n-k] = x[n]$$

MATLAB IMPLEMENTATION

- FIR filters are represented either as impulse response values $\{h(n)\}$ or as difference equation coefficients $\{b_m\}$ and $\{a_0 = 1\}$.
    - conv(x,h)
    - filter(b,1,x)

- IIR filters are described by thedifference equation coefficients $\{b_m\}$ and $\{a_k\}$
    - filter(b,a,x)

TERIMA KASIH

Adhi Harmoko Saputro

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia