Chapter 2 : Discrete-Time Signal and Systems

- There are some alternative ***representations of a discrete –time signal***:

+ ***Functional***

$$X(n) = \begin{cases} 1, for\ n = 1,3 \\ 4, for\ n = 2 \\ 0, elsewhere \end{cases} \qquad (2.1.1)$$

+ ***Tabular***

n	$\cdots$	-2	-1	0	1	2	3	4	5	$\cdots$
$X(n)$	$\cdots$	0	0	0	1	4	1	0	0	$\cdots$

+ ***Sequence***

$$X(n) = \left\{ \cdots 0, \underset{\uparrow}{0}, 1, 4, 1, 0, 0 \cdots \right\} \qquad (2.1.2)$$

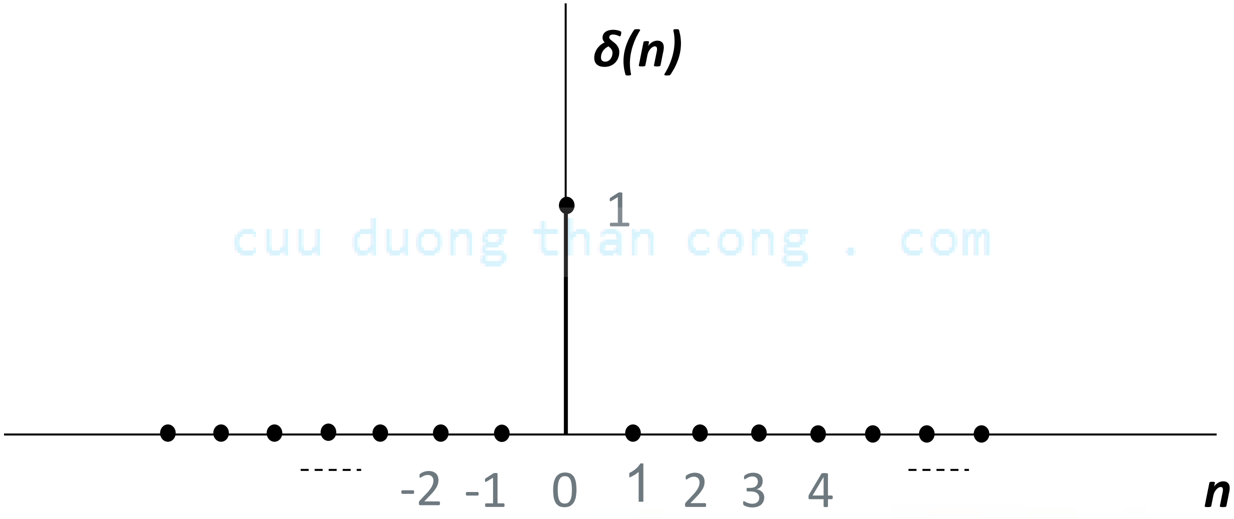
Symbol ↑ is represented *n = 0*

2.1.1 Some Elementary Discrete-Time Signals

+ *The **unit sample sequence***:

$$\delta(n) \equiv \begin{cases} 1, & for\ n = 0 \\ 0, & for\ n \neq 0 \end{cases} \qquad (2.1.6)$$

Figure 2.2: Graphical representation of the unit signal

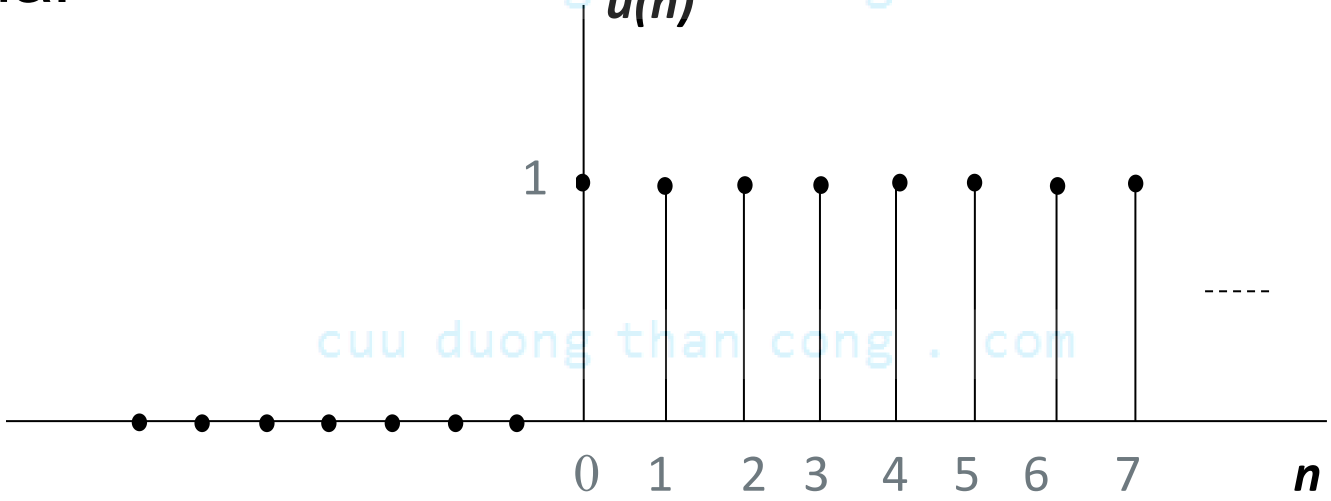


2.1.1 Some Elementary Discrete-Time Signals

+ *The* ***unit step signal****:*

$$u(n) \equiv \begin{cases} 1, & \text{for } n \geq 0 \\ 0, & \text{for } n < 0 \end{cases} \qquad (2.1.7)$$

Figure 2.3: Graphical representation of the unit step signal

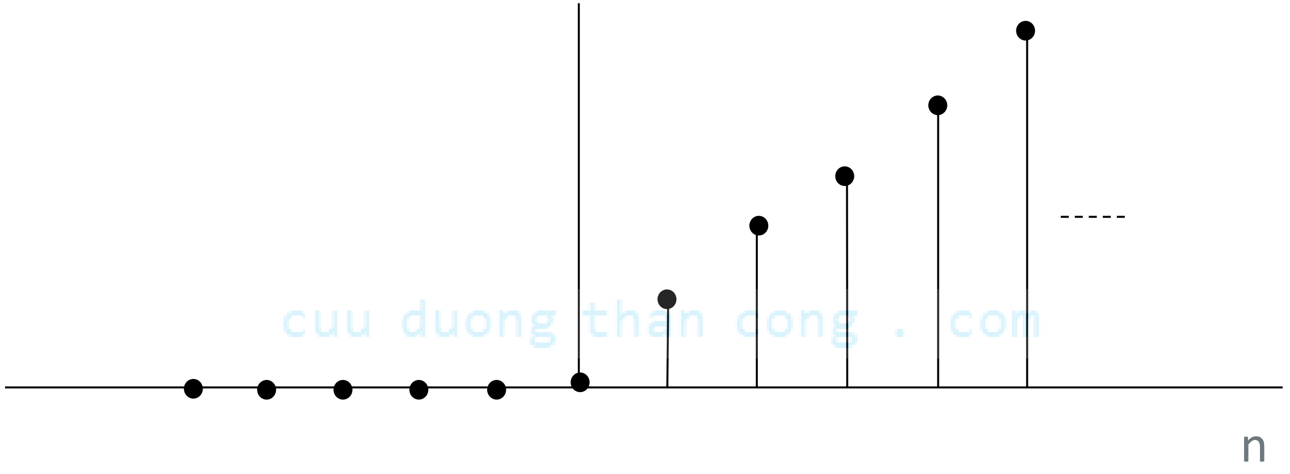


2.1.1 Some Elementary Discrete-Time Signals

+ The ***unit ramp signal***

$$u_r(n) \equiv \begin{cases} n, & for\ n \geq 0 \\ 0, & for\ n < 0 \end{cases} \qquad (2.1.8)$$

Figure 2.4: Graphical representation of the unit ramp signal



2.1.1 Some Elementary Discrete-Time Signals

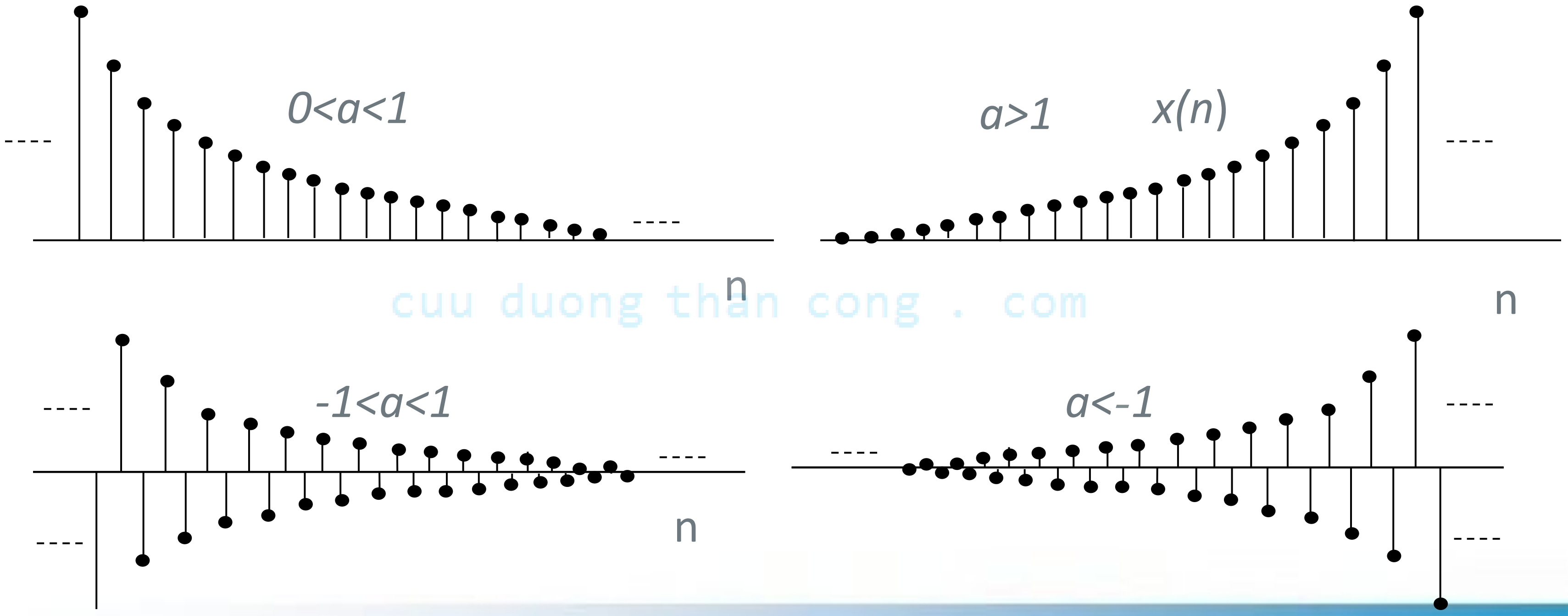
+ The ***exponential signal***:

$$X(n) = a^{n} \quad for\ all\ n \qquad (2.1.9)$$

$$X(n) = r^{n} e^{j\theta n} \qquad (2.1.10)$$

$$= r^{n}(\cos\theta n + j sin\theta n)$$

Figure 2.5 Graphical representation of exponential signal



2.1.2 Classification of Discrete-Time signals

+ The energy *E* of a signal *X(n)* is defined as

$$E \equiv \sum_{n=-\infty}^{\infty} |x(n)|^2 \qquad (2.1.15)$$

If *0< E < ∞* then *x(n)* called ***an energy signal***

+ The ***average power*** of a discrete-time signal *x(n)* is defined as

$$P = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x(n)|^2 \qquad (2.1.16)$$

+ If *-N ≤ n ≤ N* as

$$E_N \equiv \sum_{n=-N}^{N} |x(n)|^2 \qquad (2.1.17)$$

2.1.2 Classification of Discrete-Time signals

+ The we can express the signal energy *E* as

$$E \equiv \lim_{N\to\infty} E_N \qquad (2.1.18)$$

and

$$P \equiv \lim_{N\to\infty} \frac{1}{2N+1} E_N \qquad (2.1.19)$$

+ If *P* is finite the signal is called a ***power signal***.

- **Periodic signals** and **aperiodic signals**.

+ Signal *x(n)* is *periodic* with period *N* (*N>0*)
if and only if *X(n+N) = X(n)* for all *n* (2.1.20)

then *N* is called the ***fundamental period***

+ If there is no value of *N* that satisfies (2.1.20), the signal is called ***nonperiodic*** or ***aperiodic***.

2.1.2 Classification of Discrete-Time signals

+ If *X(n)* is a periodic signal with fundamental period *N* and takes on finite values, its power is given by

$$P = \frac{1}{N}\sum_{n=0}^{N-1} |x(n)|^2 \qquad (2.1.23)$$

Consequently, ***periodic signals*** are ***power signals***.

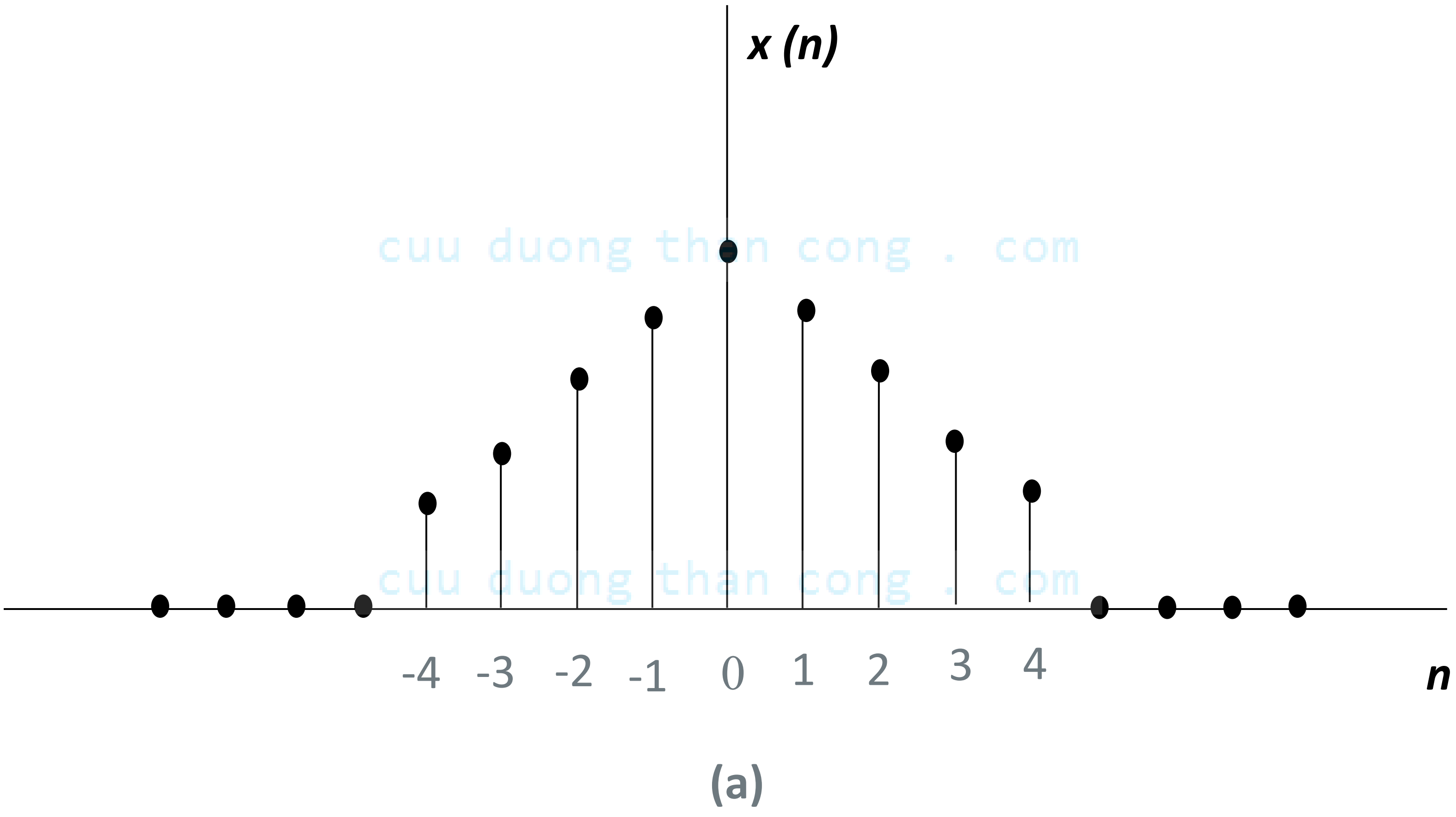
- **Symmetric** (**even**) and **antisymmetric** (**odd**) signal.

+ A real-valued signal *X(n)* is called ***symmetric*** (***even***) if *X(-n) = X(n)* *(2.1.24)*

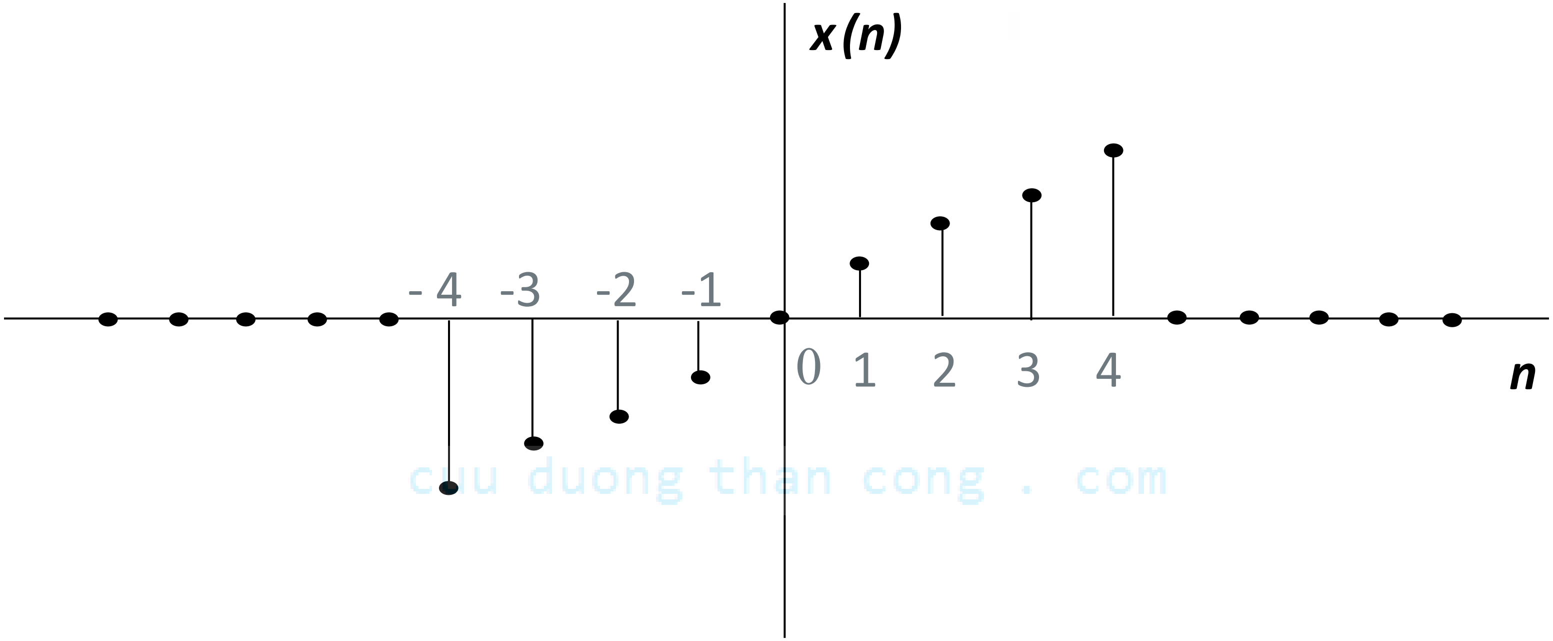
+ A signal *X(n)* is called ***antisymmetric*** (***odd***) if *X(-n) = - X(n)* *(2.1.25)*

2.1.2 Classification of Discrete-Time signals

Figure 2.8 Example of even (a) and odd (b) signal



2.1.2 Classification of Discrete-Time signals



(b)

$$X_e(n) = \frac{1}{2}[x(n) + x(-n)] \qquad (2.1.26)$$

$$X_0(n) = \frac{1}{2}[x(n) - x(-n)] \qquad (2.1.27)$$

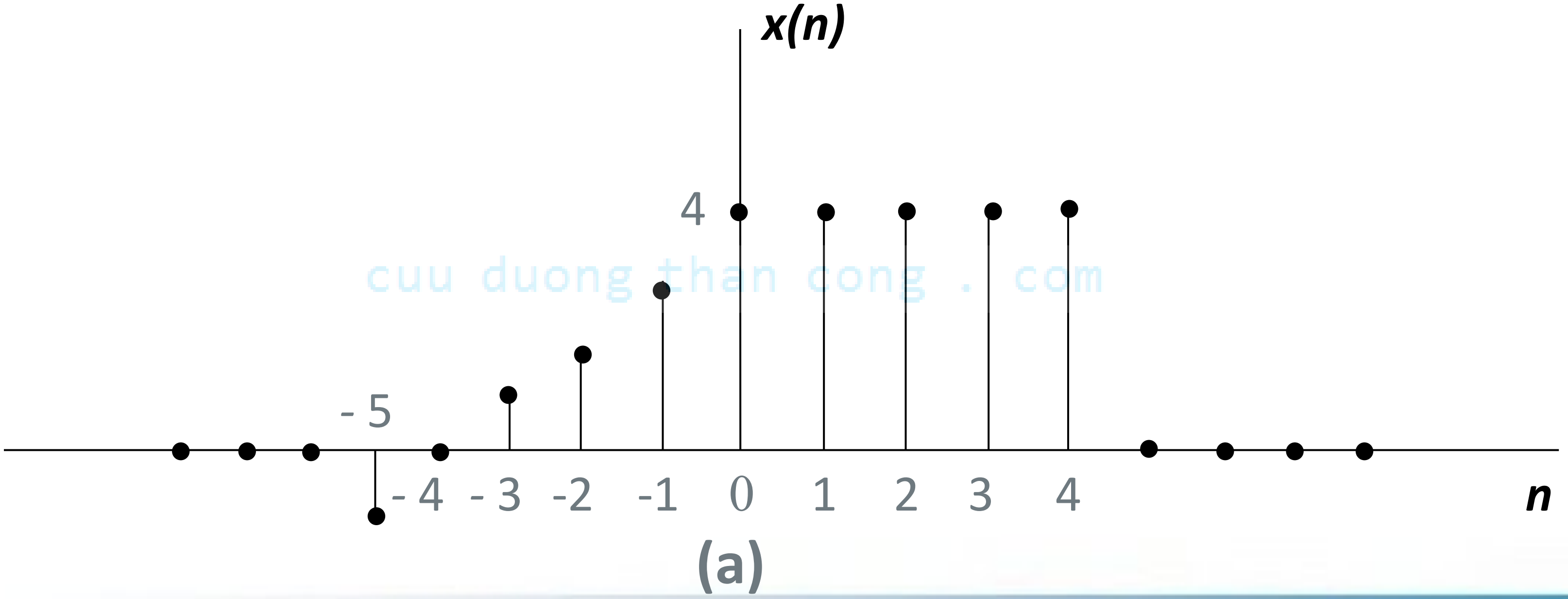
$$\rightarrow \qquad X(n) = X_e(n) + X_0(n) \qquad (2.1.28)$$

2.1.3 Simple Manipulations of Discrete-Time Signal

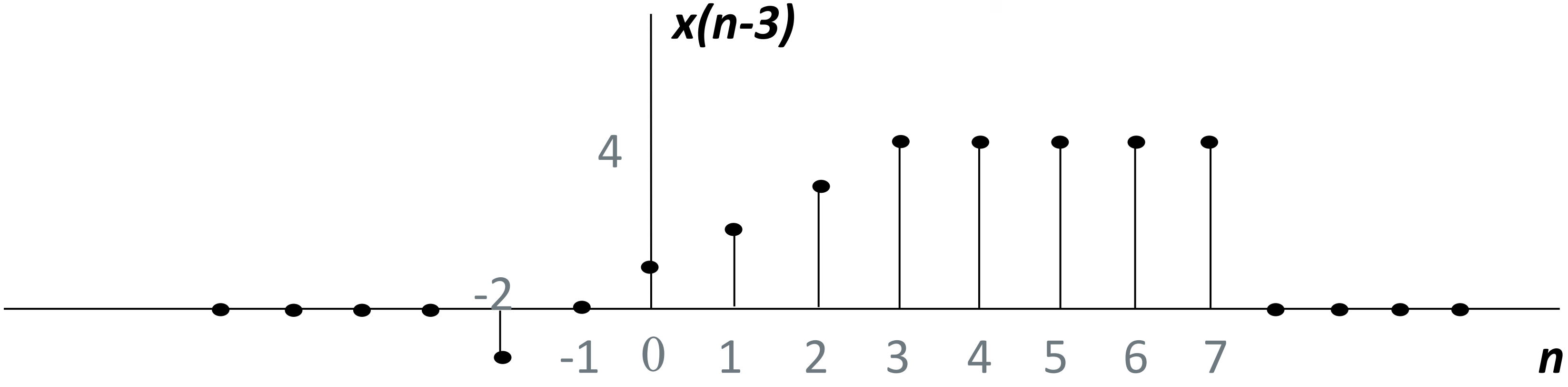
Transformation of the independent variable (time)

A signal *X(n)* may be shifted in time by replacing independent variable *n* by *n - k,* where *k* is an integer

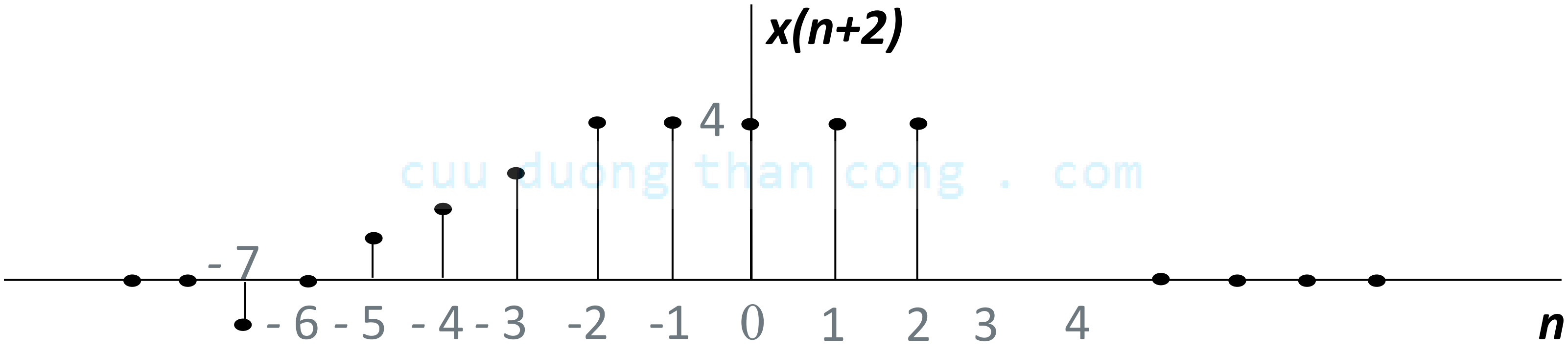
Figure 2.9 Graphical representation of a signal, and its delayed and advanced versions.



2.1.3 Simple Manipulations of Discrete-Time Signal



(b)



(c)

2.1.3 Simple Manipulations of Discrete-Time Signal

To replace the independent variable *n* by *–n* is a ***folding*** or a ***reflection*** of the signal about the time origin *n = 0*

+ ***Time-delay*** operation by *TD* and the *folding operation* by *FD :* $TD_K\,[x(n)] = x(n-k) \quad k>0$

$$FD\,[x(n)] = x(-n) \qquad (2.1.29)$$

Now $TD_K\{\,FD[x(n)]\,\} = TD_K\,[x(-n)] = x(-n-k) \qquad (2.1.30)$

Whereas $FD\,\{TD_K\,[x(n)]\,\} = FD\,[x(n-k)] = x(-n-k) \quad (2.1.31)$

+ Replacing *n* by *μn,* is as ***time scaling*** or ***down-sampling***. *μ* is an integer

2.1.3 Simple Manipulations of Discrete-Time Signal

Addition, multiplication, and scaling of sequences

+ ***Amplitude scaling*** of signal by a constant A is accomplished by multiplying the value of every signal by A

$$y(n) = Ax(n) \qquad -\infty < n < \infty$$

+ The ***sum*** of two signal $X_1(n)$ and $X_2(n)$

$$y(n) = x_1(n) + x_2(n) \qquad -\infty < n < \infty$$

+ The ***product*** of two signal

$$y(n) = x_1(n)x_2(n) \qquad -\infty < n < \infty$$

2.2 Discrete-Time Systems

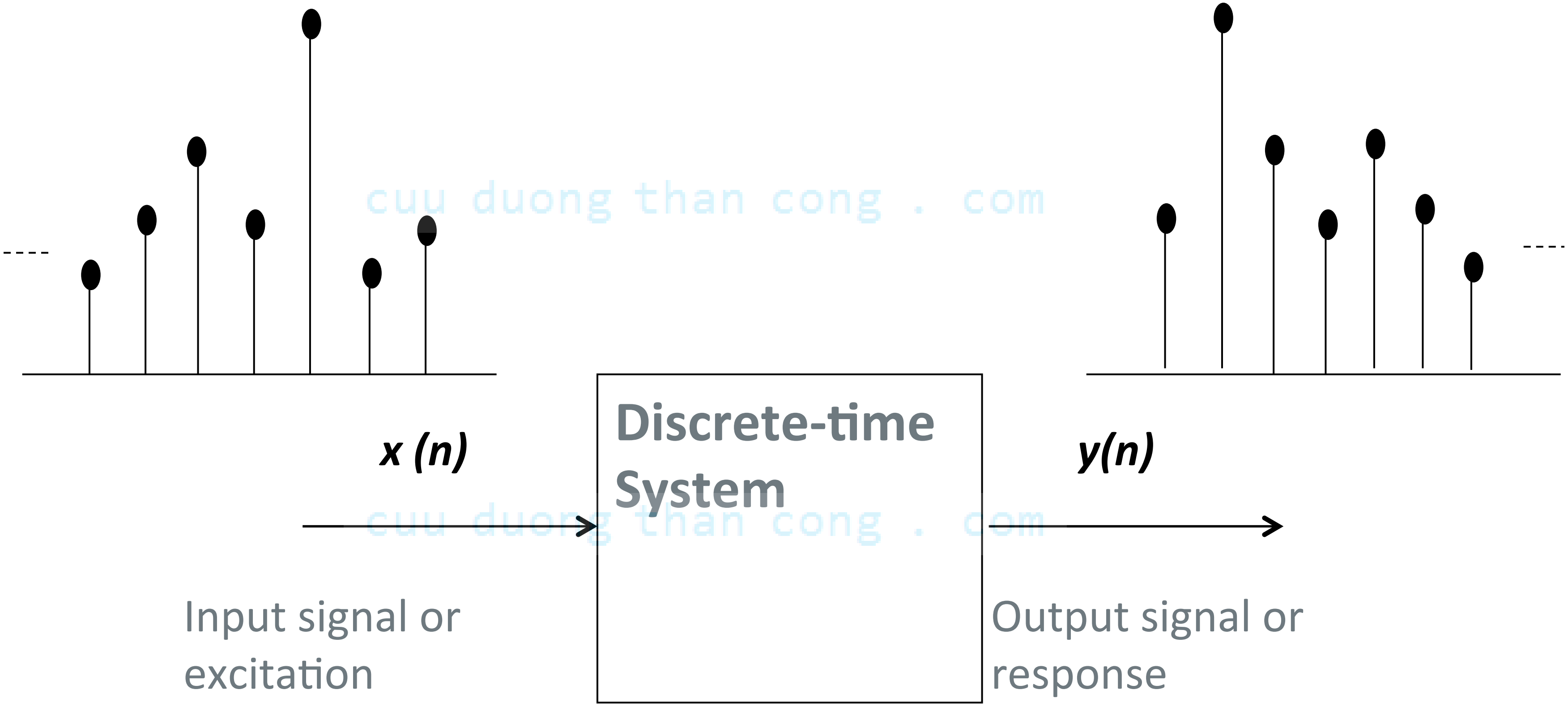
Discrete-Time Systems is a ***device*** or ***algorithm*** that operates on a discrete-time signal, called the ***input*** or ***excitation***, to produce another discrete-time signal called the ***output*** or ***response*** of the system.

$$y(n) \equiv \text{ح} [x(n)] \qquad (2.2.1)$$

to denotes the ***transformation*** by the ح .)system on *x(n)* to produce *y(n*

2.2.1 Input-Output Description of Systems

Figure 2.12 Block diagram representation of a discrete-time system



x (n)

Discrete-time System

y(n)

Input signal or excitation

Output signal or response

2.2.1 Input-Output Description of Systems

The general ***input- output relationship*** in (2.2.1), alternatively.

$$X(n) \overset{\tau}{\rightarrow} y(n) \qquad (2.2.2)$$

+ which simply means that *y(n)* is the system ح to the excitation *x(n)*

+ The additional information required to determine *y(n)* for $n \geq n_0$ is the ***initial condition*** $y(n_0 - 1)$

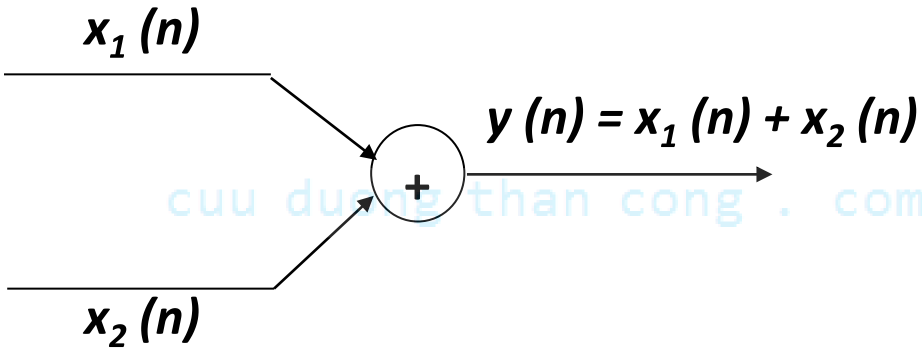
+ If the initial condition is $y(n_0 - 1) = 0$, thus system is ***initially relaxed***.

+ Block Diagram Representation of Discrete-Time System in fig 2.12

2.2.2 Block Diagram Representation of Discrete-Time Systems

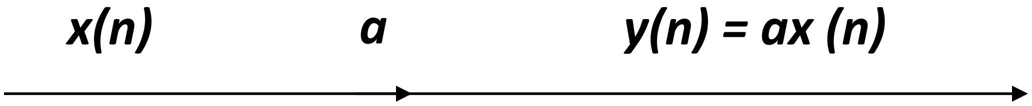
+ **An adder**

Figure 2.13 Graphical representation of an adder



+ **A constant multiplier**

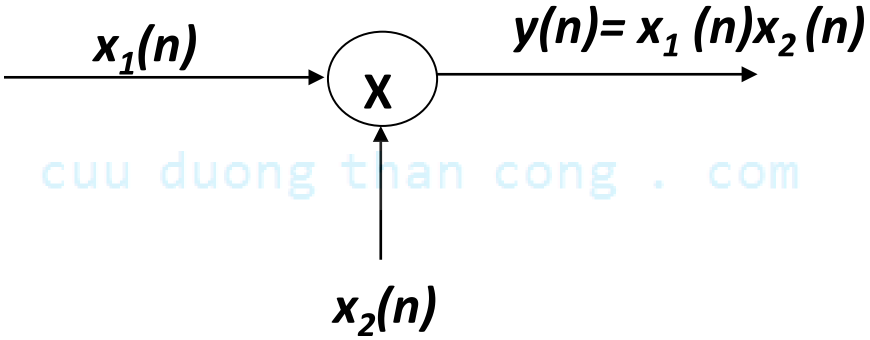
Figure 2.14 Graphical representation of a constant multiplier.



2.2.2 Block Diagram Representation of Discrete-Time Systems

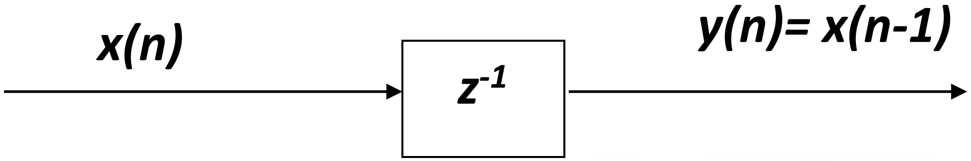
+ A signal multiplier

Figure 2.15 Graphical representation of a signal multiplier



+ A unit delay element

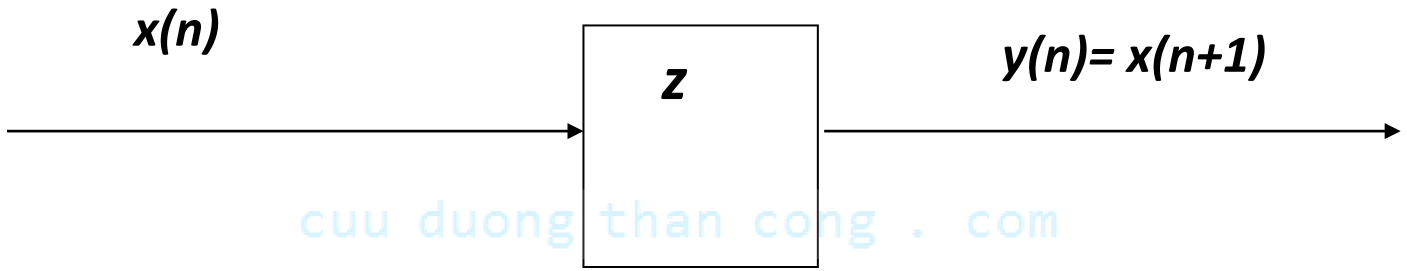
Figure 2.16 Graphical representation of unit delay element



2.2.2 Block Diagram Representation of Discrete-Time Systems

+ **A unit advance element**

Figure 2.17 Graphical representation of unit advance element



2.2.3 Classification of Discrete-Time Systems

Static versus dynamic systems

+ A discrete-time system is called ***static*** or ***memoryless*** if its output at any instant *n* depends at most on the input sample at the same time, but not on past or future samples of input.

+ In other case, the system is said to be ***dynamic*** or to have ***memory*** of duration *N*.

2.2.3 Classification of Discrete-Time Systems

+ System are ***static*** or ***memoryless***

$$y(n) = ax(n) \qquad (2.2.7)$$

$$y(n) = nx(n) + bx^3(n) \qquad (2.2.8)$$

+ System are ***dynamic***:

$$y(n) = x(n) + 3x(n-1) \qquad (2.2.9)$$

$$y(n) = \sum_{k=0}^{n} x(n-k) \qquad (2.2.10)$$

$$y(n) = \sum_{k=0}^{\infty} x(n-k) \qquad (2.2.11)$$

2.2.3 Classification of Discrete-Time Systems

Time-Invariant versus time-variant system

A system is called ***time invariant*** if its input-output characteristics do not change with time.

A ***relaxed system*** $\mathcal{T}$ is *time invariant* or ***shift invariant*** if and only if. $x(n) \xrightarrow{\mathcal{T}} y(n)$

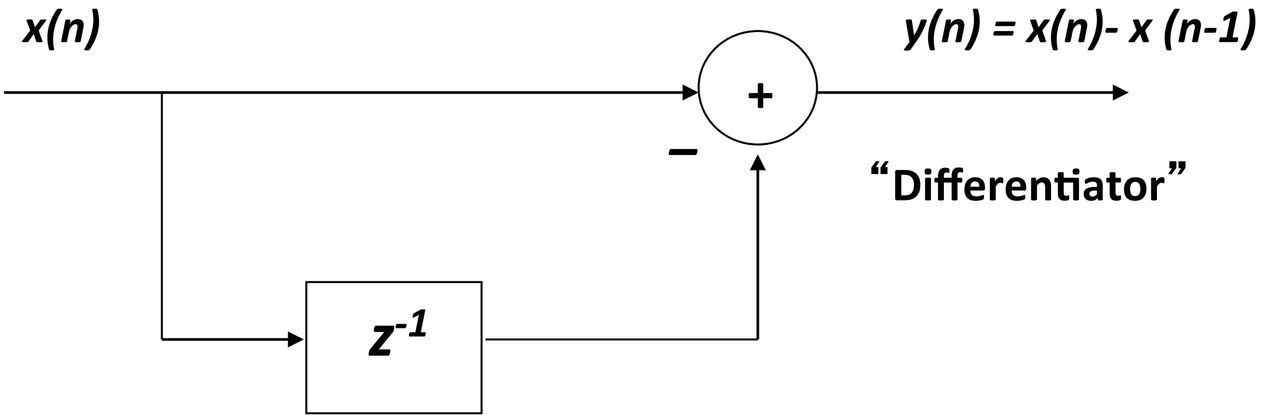
implies that

$$x(n-k) \xrightarrow{\mathcal{T}} y(n-k) \qquad (2.2.14)$$

for every input signal *x(n)* and every time shift *k.*

Figure 2. 19 Examples of a time-invariant (a) and some time-variant systems (b) - (d)

2.2.3 Classification of Discrete-Time Systems



x(n)

y(n) = x(n)- x (n-1)

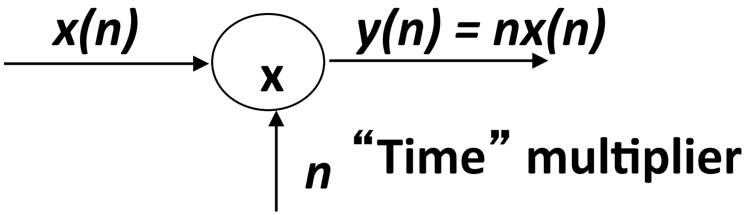
+

–

z^{-1}

"Differentiator"

(a)



x(n)

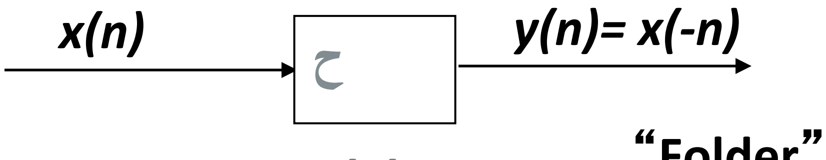
x

y(n) = nx(n)

n

"Time" multiplier

(b)



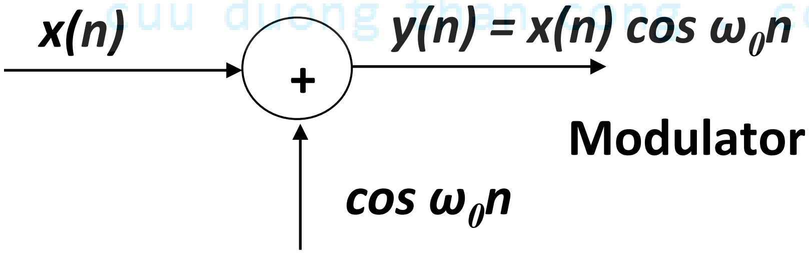
x(n)

ح

y(n)= x(-n)

"Folder"

(c)



x(n)

+

y(n) = x(n) cos $\omega_0 n$

Modulator

cos $\omega_0 n$

(d)

2.2.3 Classification of Discrete-Time Systems

Linear versus nonlinear systems

A linear system is one that satisfies the ***superposition principle***.

+ A signal is ***linear*** if and only if

$$\tau[a_1x_1(n)+a_2x_2(n)] = a_1\tau[x_1(n)]+a_2\tau[x_2(n)] \qquad (2.2.26)$$

for any arbitrary input sequences $x_1(n)$ and $x_2(n)$, and any arbitrary constants a_1 and a_2.

2.2.3 Classification of Discrete-Time Systems

Suppose that $a_2 = 0$, then (2.2.26) reduces to

$$\tau[a_1x_1(n)] = a_1\tau[x_1(n)] = a_1y_1(n) \qquad (2.2.27)$$

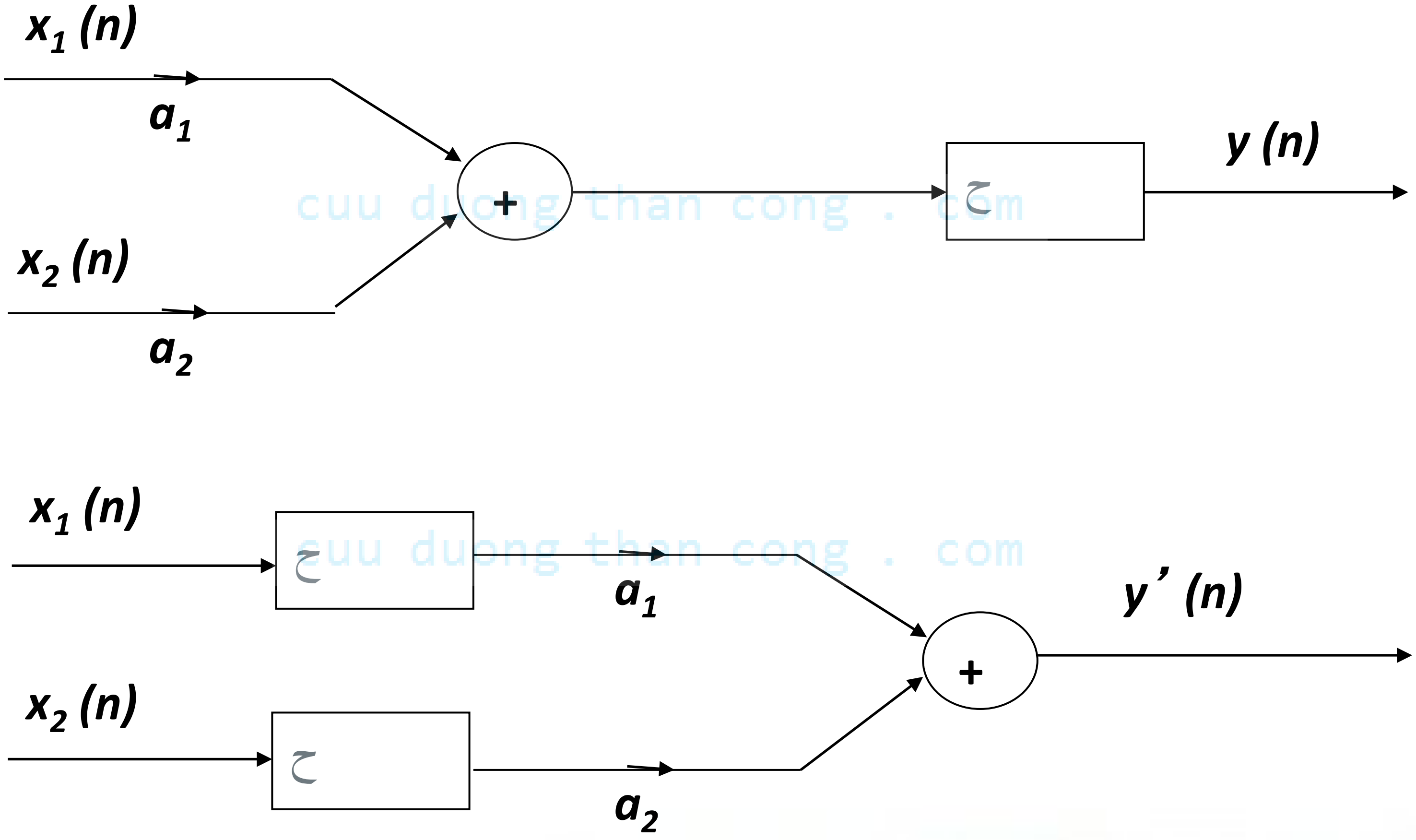
Where $y_1(n) = \tau[x_1(n)]$

Suppose that $a_1 = a_2 = 1$ in (2.2.26), then

$$\tau[x_1(n) + x_2(n)] = \tau[x_1(n)] + \tau[x_2(n)]$$
$$= y_1(n) + y_2(n) \qquad (2.2.28)$$

2.2.3 Classification of Discrete-Time Systems

Figure 2.20 Graphical representation of the superposition principle ح is linear if and only if *y(n)* = *y'(n)*.



2.2.3 Classification of Discrete-Time Systems

Causal versus noncausal system

A system is causal if the output of the system at any time *n*, depends only on ***present*** and ***past inputs***.

$$y(n) = F[x(n), x(n-1), x(n-2), \ldots] \qquad (2.2.29)$$

where *F [.]* is some arbitrary function.

If a system does not satisfy this definition, it is called ***noncausal***.

2.2.3 Classification of Discrete-Time Systems

Stable versus unstable systems

A arbitrary relaxed system is said to be ***bounded input- bounded output (BIBO) stable*** if and only if every bounded input produces a bounded output.

$|x(n)| \leq M_x < \infty, \quad |y(n)| \leq M_y < \infty \quad (2.2.45)$

where M_X, M_y - are finite numbers.

If for some bounded input sequence *x(n)*, the output is ***unbounded***, the system is classified as ***unstable***.

2.2.4 Interconnection of Discrete-time Systems

These interconnects are illustrated in fig.2.21

$y_1(n) =$ 1ح $[x(n)]$ (2.2.46)

$y(n) =$ 2ح $[y_1(n)] =$ 1ح{ 2ح$[x(n)]$} (2.2.47)

)248.2(1ح 2ح ≡ cح

- In the ***cascade interconnection***:

If 1ح and 2ح are linear and time invariant, then

a/ 1ح is time invariant

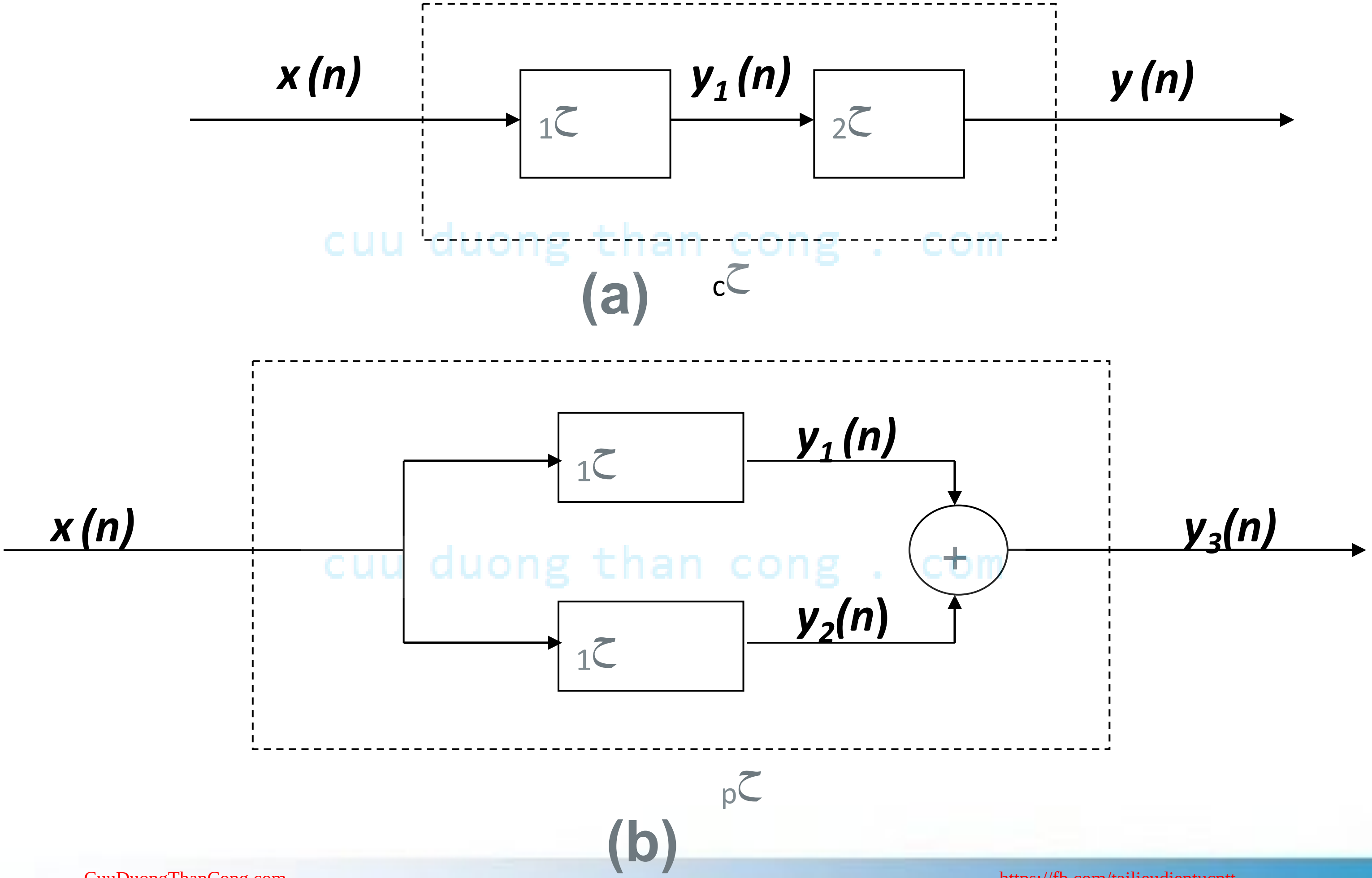
b/ ح = 2ح1ح = 2ح1حc is time invariant

- In the ***parallel interconnection:***

$y_3(n) =$ حp $[\, x(n)]$ where حp = 2ح + 1ح

2.2.4 Interconnection of Discrete-time Systems

Figure 2.21 *Cascade* (a) and *parallel* (b) interconnections of systems.



(a)

(b)

2.3 Analysis of Discrete-Time linear time-invariant (LTI) Systems

Systems are characterized in the time domain simply by their ***response*** to a ***unit sample sequence***.

Any arbitrary ***input signal*** can be decomposed and represented as a ***weighted sum*** of ***unit sample sequences***.

2.3.1 Techniques for the analysis of linear systems

+ Fist method (***a different equation***)

For an LTI system, the general form of input-output relationship is

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k) \qquad (2.3.1)$$

where *{ a_k}* and *{ b_k}* are constant parameters.

The solution of (2.3.1) is the subject of Section 2.4.

2.3.1 Technical for the Analysis of linear Systems

+ Second method ***(analyzing the behavior)***

$$x(n) = \sum_k c_k x_k(n) \qquad (2.3.2)$$

$$\boldsymbol{y_k(n) \equiv \text{ح} [\, x_k(n) \,]} \qquad \textbf{(2.3.3)}$$

$$y(n) = \tau[x(n)] = \tau\left[\sum_k c_k x_k(n)\right]$$

$$= \sum_k c_k \tau[x_k(n)] \quad = \sum_k c_k y_k(n) \qquad (2.3.4)$$

2.3.2 Resolution of a Discrete-Time signal into impulses

We select the elementary signals $x_k(n)$ to be

$$\boldsymbol{x_k(n) = \delta(n-k)} \qquad (3.3.7)$$

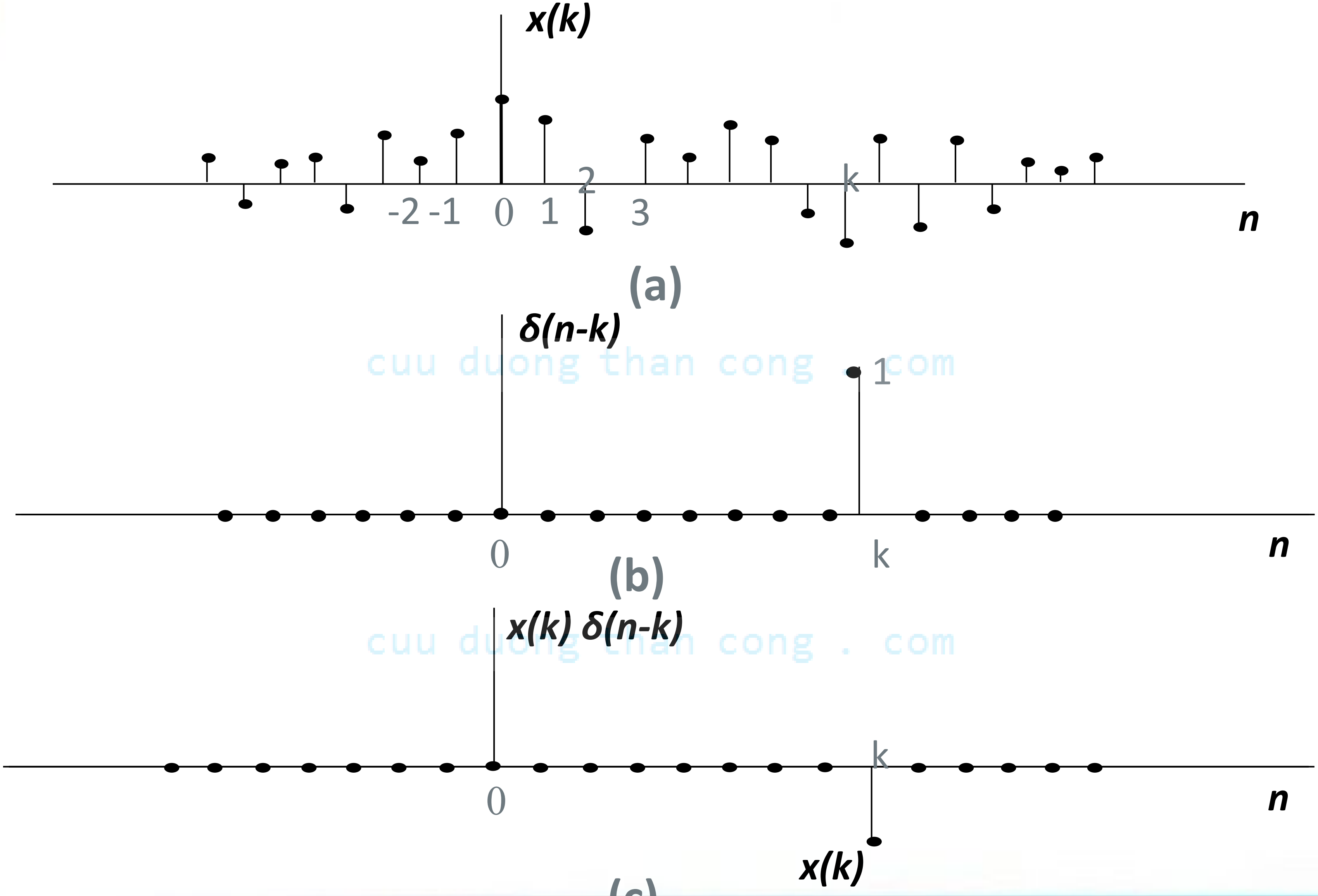
since only $\delta(n-k) = 1$ when $n = k$, then

$$\boldsymbol{x(n)\,\delta(n-k) = x(k)\,\delta(n-k)} \qquad (3.3.8)$$

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k) \qquad (3.3.10)$$

Figure 2.22: Multiplication of a signal *x(n)* with a shifted unit sample sequence

2.3.2 Resolution of a Discrete-Time Signal into Impulse



(a)

(b)

(c)

2.3.3 Response of LTI systems to Arbitrary Inputs: convolution sum

The response *y(n,k)* of the system to the input unit sample sequence at *n=k* by the special symbol ***h(n.k)***

$$\boldsymbol{y(n,k) \equiv h(n,k) = \zeta\,[\,\delta(n-k)]} \qquad (3.3.11)$$

where $-\infty < k < \infty$

$$y(n) = \tau[x(n)] = \tau\left[\sum_{k=-\infty}^{\infty} x(k)\delta(n-k)\right]$$

$$= \sum_{k=-\infty}^{\infty} x(k)\tau[\delta(n-k)] \qquad (2.3.14)$$

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

(2.3.14) is known as the ***superposition summation***

2.3.3 Response of LTI systems to Arbitrary Inputs: convolution sum

If the response of the LTI system to the unit sample sequence *δ(n - k)* is denoted as *h(n)* , that is.

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) \qquad (2.3.17)$$

The formula in (2.3.17) is called a ***convolution sum***

The response at $n = n_0$ is given as

$$y(n_0) = \sum_{k=-\infty}^{\infty} x(k)h(n_0-k) \qquad (2.3.18)$$

2.3.3 Response of LTI systems to Arbitrary Inputs: convolution sum

The ***process of computing the convolution*** between *x(k)* and *h(k)* involves the following four steps.

1. *Folding*. Fold *h(k)* about *k = 0* to obtain *h(-k)*
2. *Shifting*. Shift *h(-k)* by n_0 to obtain $h(n_0 - k)$
3. *Multiplication*. Multiply *x(k)* by $h(n_0 – k)$ to obtain $v_{n0}(k) \equiv x(k)\, h(n_0 - k)$
4. *Summation*. sum all the values of the product sequence $v_{n0}(k)$

2.3.3 Response of LTI systems to Arbitrary Inputs: convolution sum

Example 2.3.2

The impulse response of a linear time-invariant system is $h(n) = \{1,\underset{\uparrow}{2},1,-1\}$ (2.3.19)

Determine the response of the system to the input signal $x(n) = \{\underset{\uparrow}{1},2,1,-1\}$ (2.3.20)

Solution: at $n = 0$

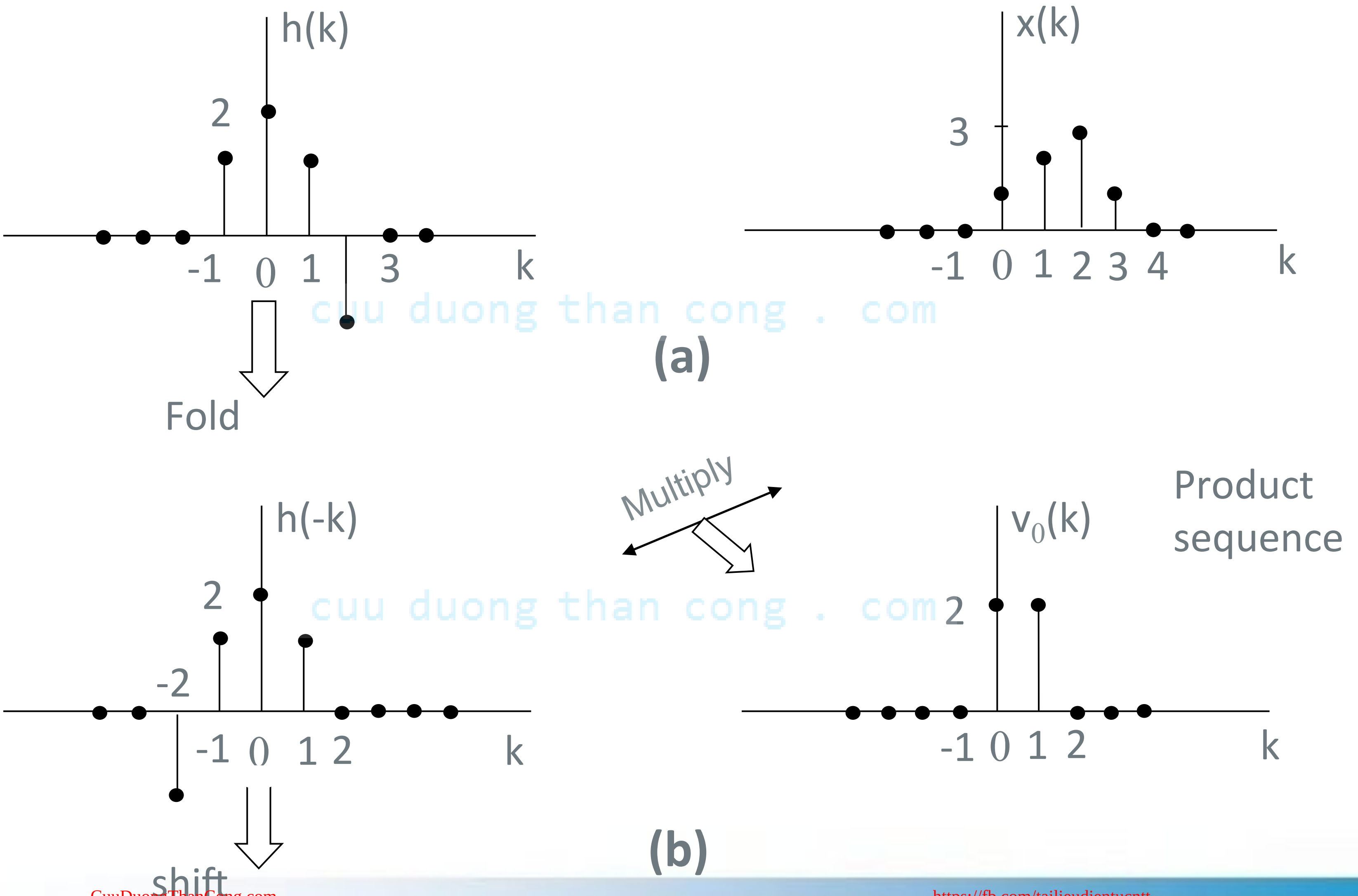
$$y(0) = \sum_{k=-\infty}^{\infty} x(k)h(-k) \qquad (2.3.21)$$

$$v_0(k) \equiv x(k)h(-k) \qquad (2.3.22)$$

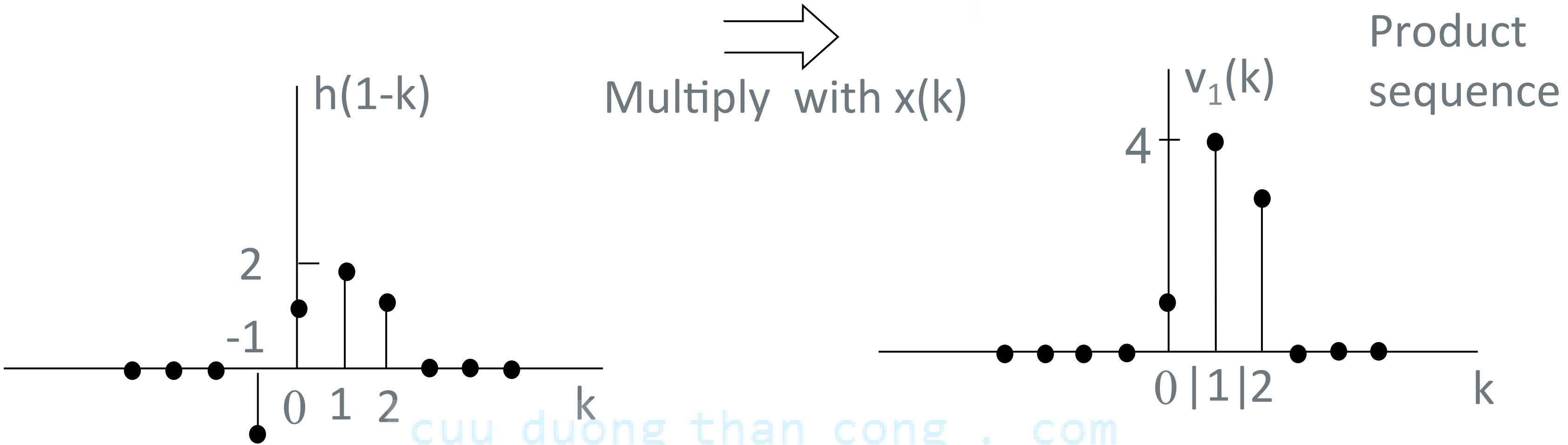
$$y(0) = \sum_{k=-\infty}^{\infty} v_0(k) = 4$$

2.3.3 Response of LTI systems to Arbitrary Inputs: convolution sum

Figure 2.24 Graphical computation of convolution.



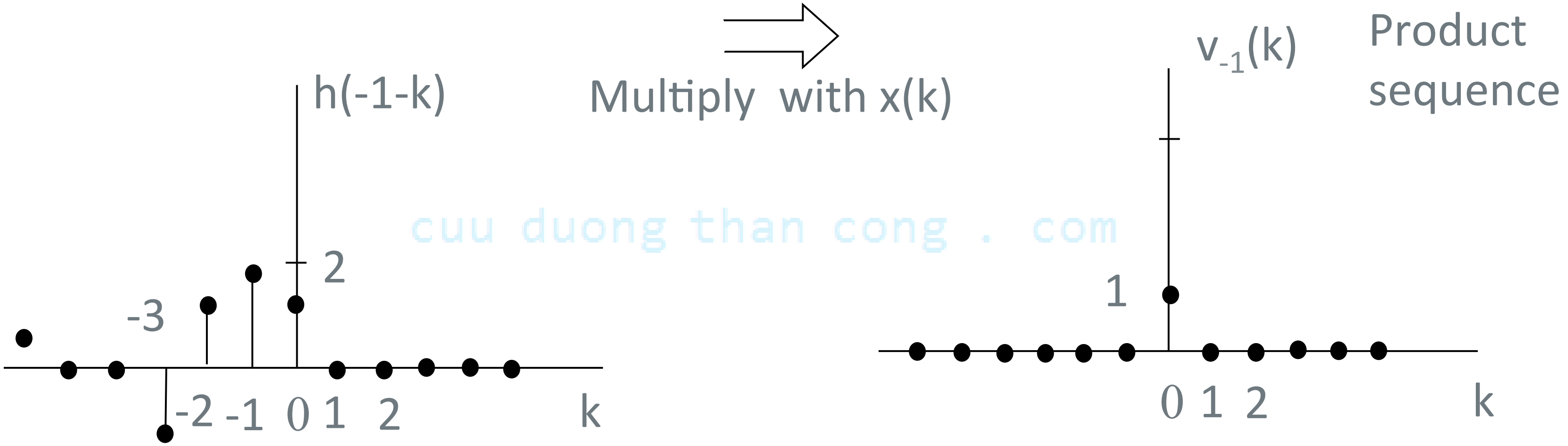
2.3.3 Response of LTI systems to Arbitrary Inputs: convolution sum



Multiply with x(k)

Product sequence

(c)

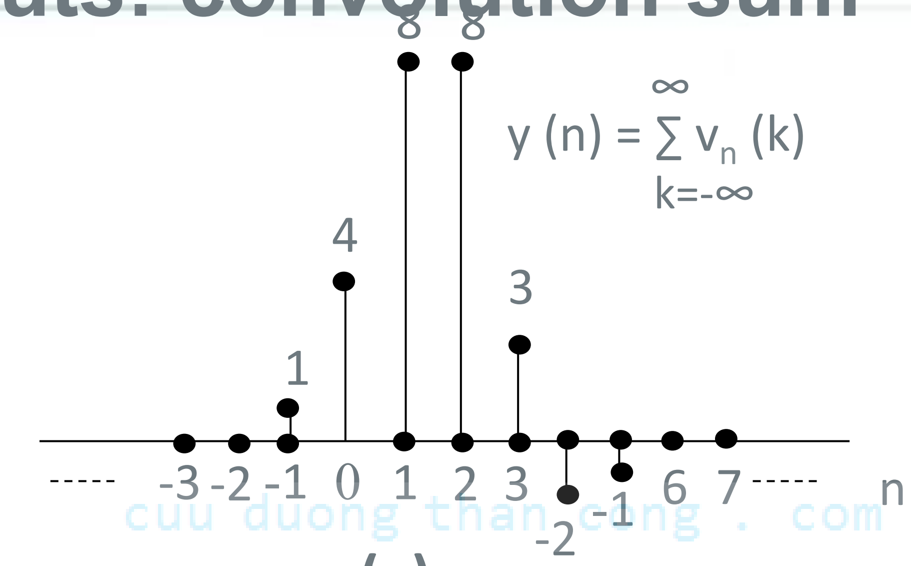


Multiply with x(k)

Product sequence

(d)

2.3.3 Response of LTI systems to Arbitrary Inputs: convolution sum



$$y(n) = \sum_{k=-\infty}^{\infty} v_n(k)$$

(e)

$$n=1 \qquad y(1)=\sum_{k=-\infty}^{\infty} x(k)h(1-k) = 8 \qquad (2.3.23)$$

$$n=-1 \qquad y(-1)=\sum_{k=-\infty}^{\infty} x(k)h(-1-k) = 1 \qquad (2.3.24)$$

$$y(n) = \{\ldots,0,0,1,\underset{\uparrow}{4},8,8,3,-2,-1,0,0,\ldots\} \qquad (2..3.26)$$

2.3.4 Properties of convolution and the interconnection of LTI system

An other form of the convolution formula

$$y(n) = \sum_{k=-\infty}^{\infty} x(n-k)h(k) \qquad (2.3.28)$$

Using an ***asterisk*** to denote the convolution operation. Thus,

$$y(n) = x(n) * h(n) \equiv \sum_{k=-\infty}^{\infty} h(k)h(n-k) \qquad (2.3.31)$$

$$y(n) = h(n) * x(n) \equiv \sum_{k=-\infty}^{\infty} h(k)x(n-k) \qquad (2.3.32)$$

2.3.4 Properties of convolution and the interconnection of LTI system

+ **Commutative law**

$$x(n) * h(n) = h(n) * x(n) \qquad (2.3.33)$$

+ **Associative law**

$$[x(n) * h_1(n)] * h_2(n) = x(n) * [h_1(n) * h_2(n)] \qquad (2.3.34)$$

+ **Distributive law**

$$x(n) * [h_1(n) + h_2(n)] = x(n) * h_1(n) + x(n) * h_2(n) \ (2.3.36)$$

2.3.5 Causal linear time-invariant systems

An LTI system is ***causal*** if and only if ***its impulse response*** is ***zero*** for ***negative values*** of *n*.

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) \qquad (2.3.39)$$

$$y(n) = \sum_{k=-\infty}^{n} x(k)h(n-k) \qquad (2.3.40)$$

The ***response*** of a causal system to a causal input sequence is ***causal***, since

$y(n) = 0$ for $n < 0$.

2.3.6 Stability of linear time-invariant systems

If *x(n)* or *y(n)* is bounded, there exists a ***constant*** $\boldsymbol{M_X}$ or $\boldsymbol{M_y}$, such that

$$|y(n)| < M_x < \infty \qquad |y(n)| < M_y < \infty$$

for all *n*

A linear time-invariant system is ***stable*** if its impulse response is ***absolutely summable***.

$$S_h \equiv \sum_{k=-\infty}^{\infty} |h(k)| < \infty \qquad (2.3.43)$$

2.3.7 System with Finite-duration and infinite-duration impulse response

Linear time-invariant system is subdivided into two types

+ *A **finite-duration impulse response*** (***FIR***) system has an impulse response that is ***zero outside*** of some finite time-interval.

If *h(n) = 0* with *n < 0* and *n* ≥ *M*

then

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k)$$

+ And those that have *an **infinite-duration impulse response*** (***IIR***).

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k)$$

2.4 Discrete-time system described by difference equations

The convolution summation

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k) \qquad (2.4.1)$$

In the case of ***FIR*** systems, realization involves ***additions***, ***multiplications***, and a *finite number* of ***memory locations***.

If the system is ***IIR***, which is more conveniently described by ***difference equations.***

This family of IIR systems is very useful in a variety of practical applications.

2.4.1 Recursive and Nonrecursive Discrete-time systems

Suppose that the ***cumulative average***

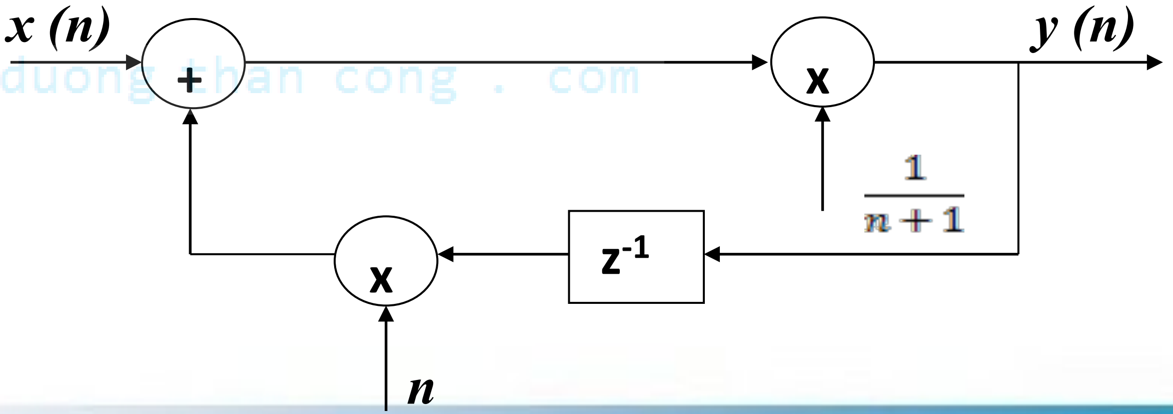
$$y(n) = \frac{1}{n+1}\sum_{k=0}^{n} x(k) \qquad (2.4.2)$$

and hence

$$y(n) = \frac{1}{n+1} y(n-1) + \frac{1}{n+1} x(n) \qquad (2.4.3)$$

n = 0,1,… 0 ≤ k ≤ n.

Figure 2.28 Realization of a recursive cumulative averaging system.



2.4.1Recursive and Nonrecursive Discrete-time systems

+ The causal linear time-invariant FIR system described by the convolution formula in following are ***nonrecursive***.

$$y(n) = \sum_{k=0}^{M} h(k)x(n-k)$$

$$= h(0)x(n) + h(1)x(n-1) + \cdots + h(M)x(n-M)$$

$$= F[x(n)x(n-1),\dots,x(n-M)]$$

where *F[.]* is simply a linear weighted sum of present and past inputs and the impulse response values *h(n).*

2.4.2 Linear Time-Invariant Systems Characterized by Constant-Coefficient Difference Equations

Suppose that

$$y(n) = a\,y(n-1) + x(n) \qquad (2.4.7)$$

Thus

$$y(n) = a^{n+1}y(-1) + \sum_{k=0}^{n} a^k x(n-k) \quad n \geq 0 \qquad (2.4.8)$$

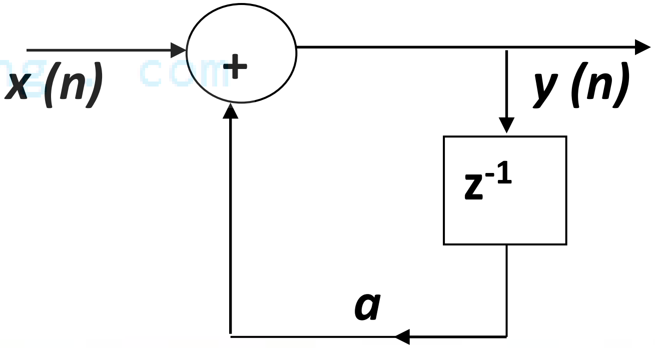


Figure 2.31: Block diagram realization of a simple recursive system.

2.4.2 Linear Time-Invariant Systems Characterized by Constant-Coefficient Difference Equations

+ The system is at zero state and its corresponding output is called ***the zero state response*** or ***forced response*** and is denoted by ***$y_{zs}(n)$***

$$y_{zs}(n) = \sum_{k=0}^{n} a^k x(n-k) \qquad n \geq 0 \qquad (2.4.9)$$

+ The output of the system with zero input is called ***the zero-input response*** or ***natural response***, and is dented by ***$y_{zi}(n)$***

$$y_{zi}(n) = a^{n+1} y(-1), \qquad n \geq 0 \qquad (2.4.11)$$

+ The ***total response*** of the system can be expressed as ***$y(n) = y_{zi}(n) + y_{zs}(n)$***

2.4.2 Linear Time-Invariant Systems Characterized by Constant-Coefficient Difference Equations

+ The *general* ***form*** of ***recursive system*** described by linear constant-coefficient difference equations.

$$y(n) = -\sum_{K=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k) \qquad (2.4.12)$$

or, equivalently,

$$\sum_{k=0}^{N} a_k y(n-k) = \sum_{k=0}^{M} b_k x(n-k), \quad a_0 \equiv 1 \quad (2.4.13)$$

The Integer ***N*** is called *the* ***order*** of ***difference equation***.

2.4.2 Linear Time-Invariant Systems Characterized by Constant-Coefficient Difference Equations

+ A system is ***linear*** if it satisfies the following three requirements:

1. The total response is equal to the sum of the *zero input* and the *zero-state responses*.
2. The principle of superposition applies to the zero-state response (*zero-state linear*).
3. The principle of superposition applies to the zero-input response *(zero-input linear*).

+ ***The recursive system*** described by a *linear constant-coefficient difference equation* is ***linear and time invariant.***

2.4.3 Solution of Linear constant-coefficient Difference Equations.

+ *The **direct solution method*** is to determine the output $y(n), n \geq 0$, of the system given a specific input $x(n), n \geq 0$, and set of initial conditions.

Total solution is the sum of two parts:

$$y(n) = y_h(n) + y_p(n)$$

$y_h(n)$ – the ***homogeneous*** or ***complementary solution***.

$y_p(n)$ – the ***particular solution***.

2.4.3 Solution of Linear constant-coefficient Difference Equations

+ **The homogeneous solution of a difference equation.**

To Assume that *x(n) = 0*, from (2.4.13), we obtain the homogeneous difference equation

$$\sum_{k=0}^{N} a_k y(n-k) = 0 \qquad (2.4.14)$$

We assume that the solution is in the form of an exponential

$$y_h(n) = \lambda^n \qquad (2.4.15)$$

2.4.3 Solution of Linear constant-coefficient Difference Equations

$$\sum_{k=0}^{N} a_k \lambda^{n-k} = 0$$

or

$$\lambda^{n-N} (\lambda^N + a_1\lambda^{N-1} + a_2\lambda^{N-2} + \ldots + a_{N-1}\lambda + a_N) = 0 \qquad (2.4.16)$$

The polynomial in parentheses is called the ***characteristic polynomial*** of the system.

It has N roots, which we denote as $\lambda_{1,}\ \lambda_{2,}\ \ldots\ldots,\lambda_N$.

If the roots are distinct, that is.

$$y_h(n) = C_1\lambda^n{}_1 + C_2\lambda^n{}_2 + \ldots + C_N\lambda^n{}_N \qquad (2.4.17)$$

Where c_1 , $c_{2}, \ldots, c_N$ are weighting coefficients.

2.4.3 Solution of Linear constant-coefficient Difference Equations

The particular solution of the difference equation *$y_p(n)$*.

+ $y_p(n)$ is required to satisfy the difference equation (2.4.13)

$$\sum_{k=0}^{N} a_k y_p(n-k) = \sum_{k=0}^{M} b_k x(n-k) \qquad a_0 = 1 \qquad (2.4.25)$$

+ Our assumed form for the particular solution takes the ***basic form*** of the particular signal *$x(n)$*.

2.4.3 Solution of Linear constant-coefficient Difference Equations

Table 2.1 General form of the particular solution for several types of input signals.

Input signal, $x(n)$	*Particular solution,* $y_p\,(n)$
A (constant)	K
AM^n	KM^n
An^M	$K_0 n^M + K_1 n^{M-1} + \ldots + K_M$
$A^n n^M$	$A^n\,(K_0 n^M + K_1 n^{M-1} + \ldots + K_M)$
$\left\{ \begin{array}{l} A \cos \omega_0 n \;\; A \\ \sin \omega_0 n \end{array} \right\}$	$K_1 \cos \omega_0 n + K_2 \sin \omega_0 n$

2.4.3 Solution of Linear constant-coefficient Difference Equations

The total solution of the difference equation

As the linearity property of the linear constant-coefficient difference equation, we have

$$y(n) = y_h(n) + y_p(n)$$

The resultant sum *y(n)* contains the constant parameters $\{ C_i \}$ embodied in the homogeneous solution component $y_h(n)$.

These constants can be determined to satisfy the initial conditions

2.4.4 The impulse response of the linear Time-Invariant recursive system

In the simple first-order rescusive system, the zero-state response given in (2.4.7)

$$y_{zs}(n) = \sum_{k=0}^{n} a^k x(n-k) \qquad (2.4.37)$$

with $X(n) = \delta(n)$

then

$$y_{zs}(n) = \sum_{k=0}^{n} a^k \delta(n-k) = a^n, \qquad n \geq 0$$

The impulse response of the first-order recursive system described by (2.4.7) is

$$\boldsymbol{h(n) = a^n u(n)} \qquad (2.4.38)$$

2.4.4 The impulse response of the linear Time-Invariant recursive system

In the general case of an arbitrary, linear time-invariant recursive system, we have

$$y_{zs}(n) = \sum_{k=0}^{n} h(k)x(n-k), \qquad n \geq 0 \qquad (2.4.39)$$

when the input is an impulse, reduces to

$y_{zs}(n) = h(n)$ (2.4.40)

since $x(n) = 0$ for $n > 0$, that is,

$y_p(n) = 0$

2.5 Implementation of discrete-time systems

+ Let us consider the first-order system.

$$y(n) = - a_1 y(n-1) + b_0 x(n) + b_1 x(n-1) \quad (2.5.1)$$

+ Nonrecursive system described by the equation.

$$v(n) = b_0 x(n) + b_1 x(n-1) \qquad (2.5.2)$$

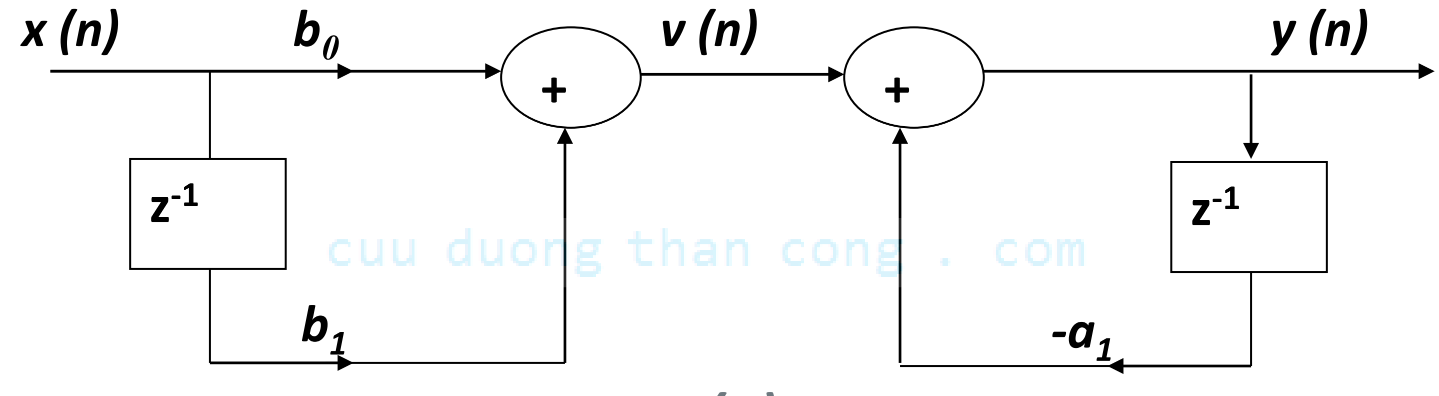
+ Recursive system described by the equation.

$$y\ (n) = - a_1 y(n-1) + v(n) \qquad (2.5.3)$$

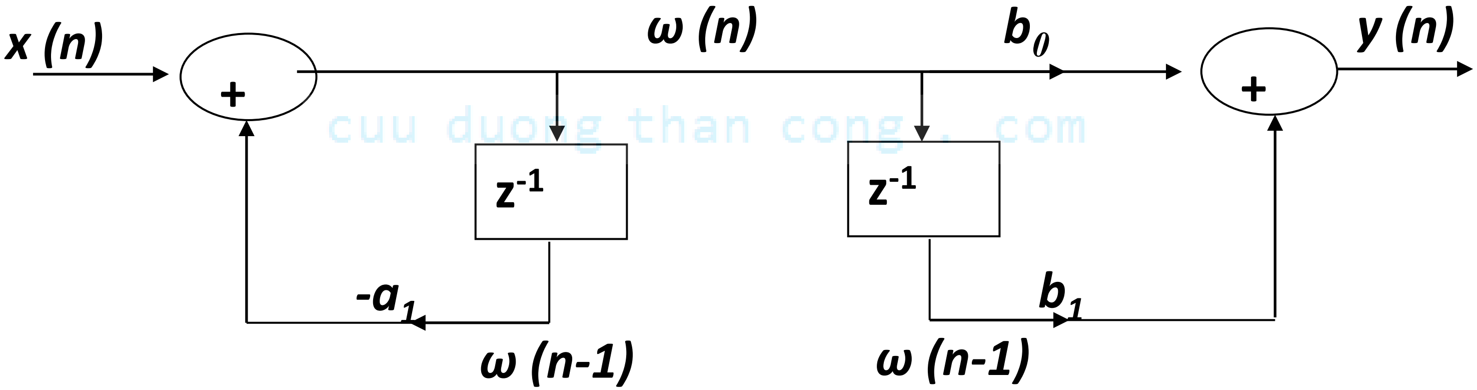
It is called a ***direct form I structure*** in *Fig 2.32(a)*.

2.5.1 Structures for the Realization of Linear Time-invariant System

Figure 2.32 Steps in converting from *the direct from I realization* in (a) to *the direct form II realization* in (c)

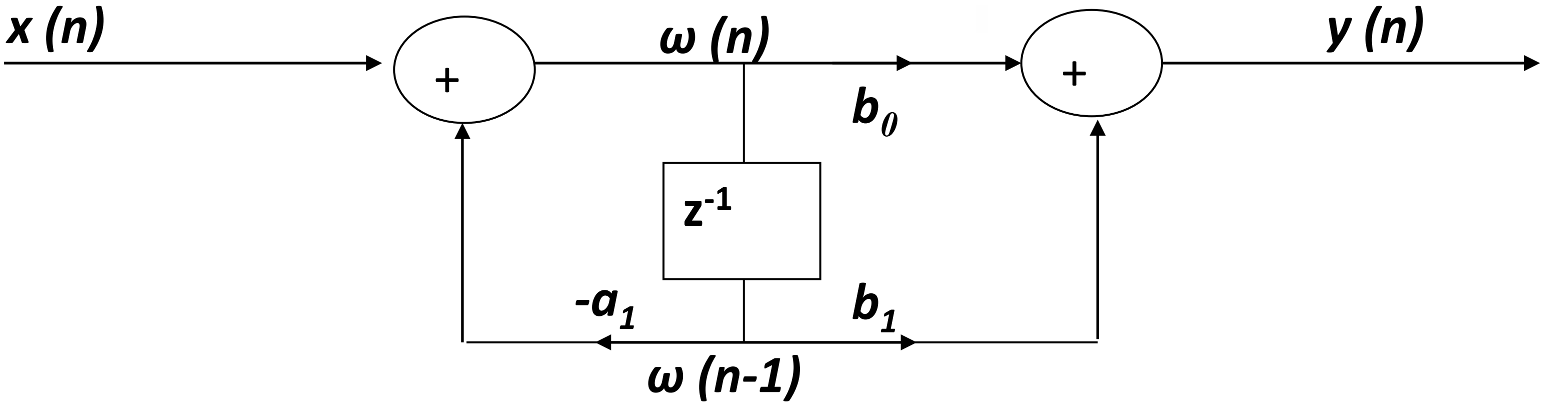


(a)



(b)

2.5.1 Structures for the Realization of Linear Time-invariant System



(c)

From this figure we obtain the two difference equations

$$w(n) = - a_1w(n\text{-}1) + x(n) \qquad (2.5.4)$$

$$y(n) = b_0w(n) + b_1w(n\text{-}1) \qquad (2.5.5)$$

These two elements *w(n-1)* can be merged in to one a delay, as show in figure 2.32 c.

It is called ***the direct form II structure.***

2.5.1 Structures for the Realization of Linear Time-invariant System

+ The difference equation.

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k) \qquad (2.5.6)$$

+ This structure requires *M+N* delays and *N+M+1* multiplications.

Nonrecursive system.

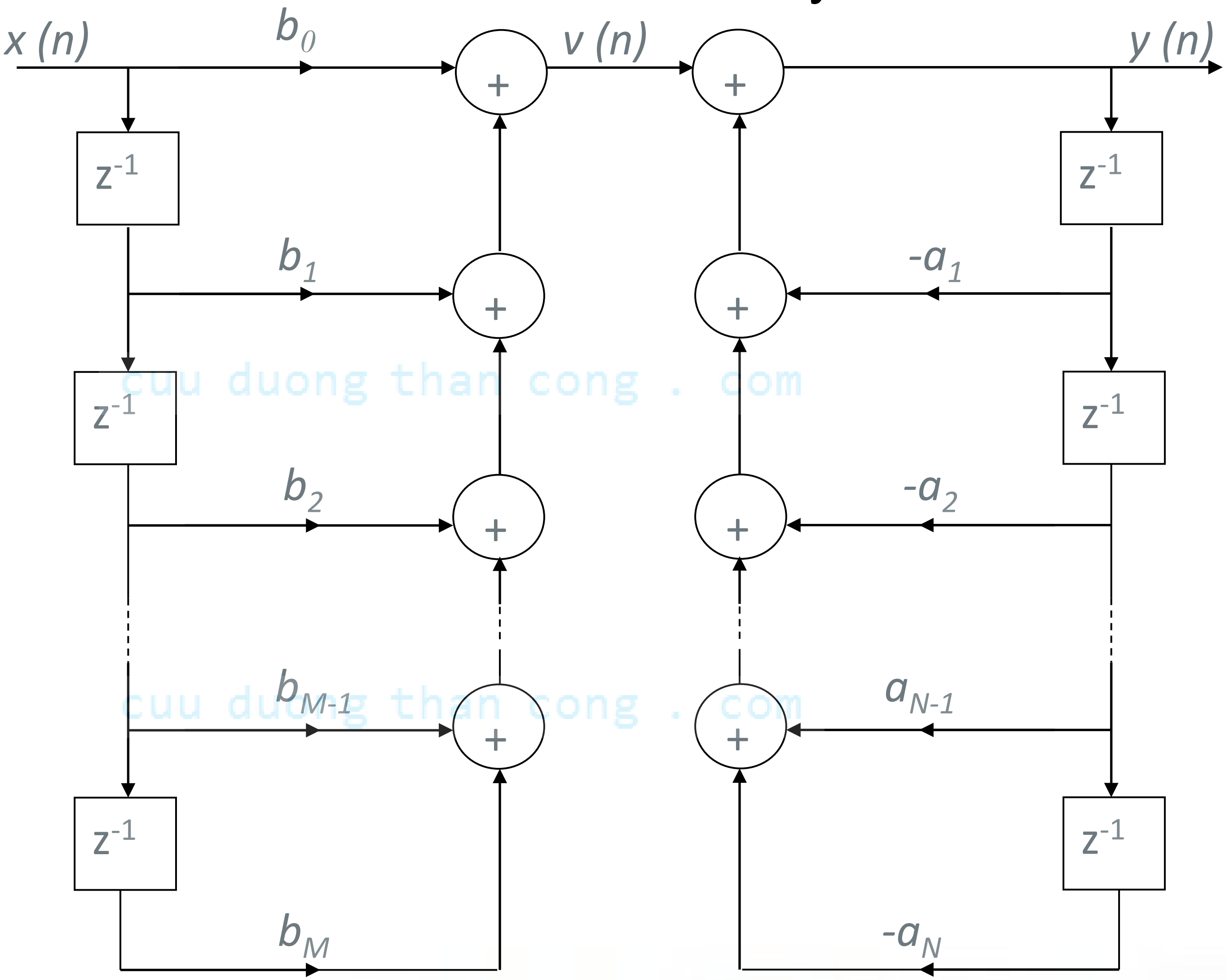
$$v(n) = \sum_{k=0}^{M} b_k x(n-k) \qquad (2.5.7)$$

Recursive system

$$y(n) = -\sum_{k=0}^{M} a_k y(n-k) + v(n) \qquad (2.5.8)$$

2.5.1 Structures for the Realization of Linear Time-invariant System

Figure 2.33 *Direct form I structure* of the system described by (2.5.6)



2.5.1 Structures for the Realization of Linear Time-invariant System

We obtain the ***direct form II structure*** show in Figure 2.34 for *N* > *M*. It is the cascade of a recursive system.

$$w(n) = -\sum_{k=1}^{N} a_k w(n-k) + x(n) \qquad (2.5.9)$$

followed by a ***nonrecursive system***

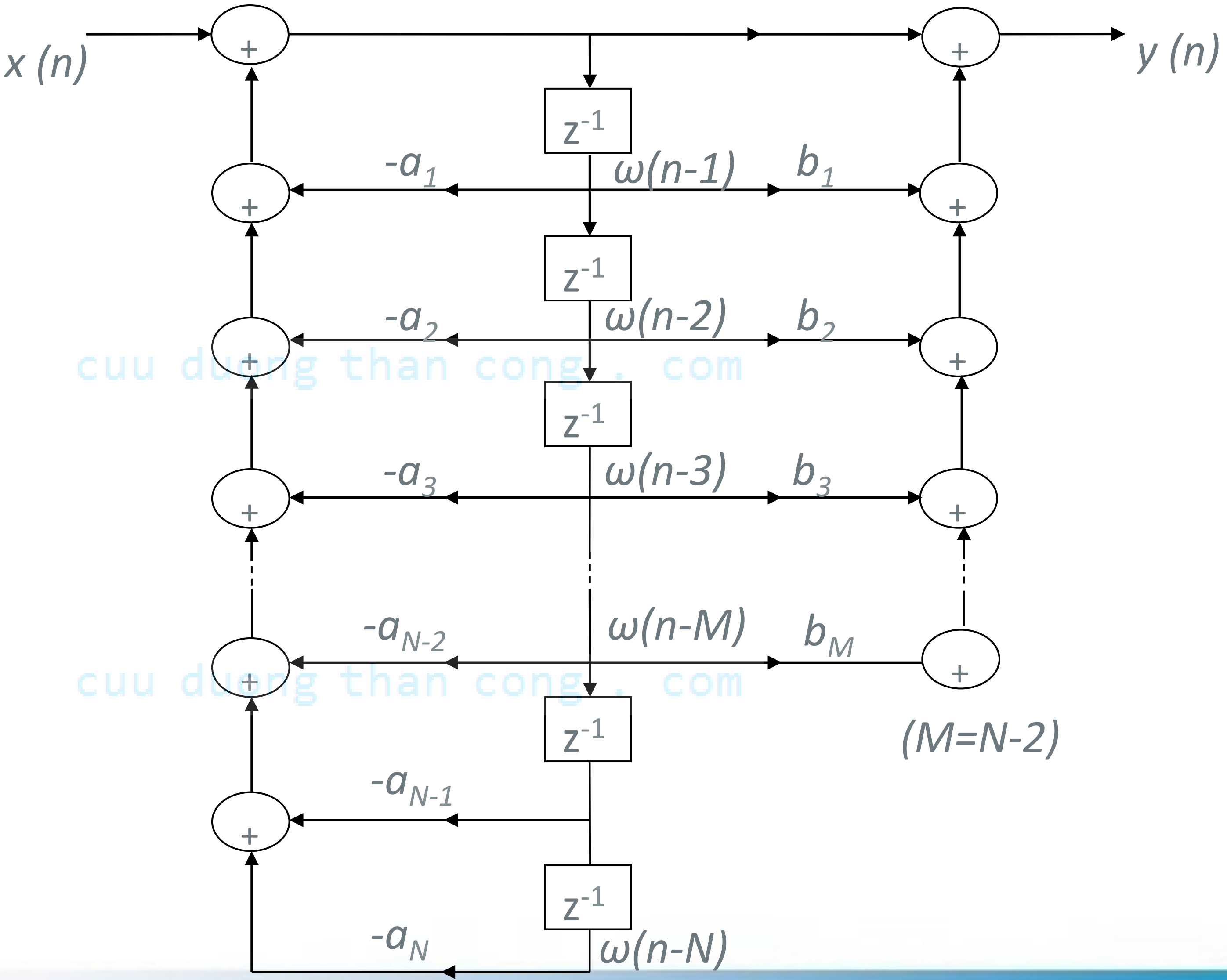
$$y(n) = \sum_{k=0}^{M} b_k w(n-k) \qquad (2.5.10)$$

If *N* ≥ *M*, this structure requires a number of delays equal to the ***order N*** of the system.

If *M* > *N*, the requires ***memory*** is specified by ***M***.

2.5.1 Structures for the Realization of Linear Time-invariant System

Figure 2.34 *Direct form II structure* of the system described by (2.5.6)



2.5.1 Structures for the Realization of Linear Time-invariant System

The direct form II structure requires *M + N +1* multiplications and *max{ M, N }* delays.

It' s sometimes called ***a canonicform*** .

A special case of (2.5.6), If $a_k = 0, k = 1, \ldots, N$, then

$$y(n) = \sum_{k=0}^{M} b_k x(n-k) \qquad (2.5.11)$$

Which is a nonrecursive linear time-invariant system.

2.5.1 Structures for the Realization of Linear Time-invariant System

The system output is basically a *weighted moving average* of the input signal, or a *moving average (MA) system*.

+ A system is an FIR system with an impulse response *h(k)* equal to the coefficients b_k, that is,

$$h(k) = \begin{cases} b_k, & 0 \le k \le M \\ 0, & otherwise \end{cases}$$

+ from (2.5.6) and set *M=0*,

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + b_0 x(n) \qquad (2.5.13)$$

2.5.1 Structures for the Realization of Linear Time-invariant System

The ***most general second-order system*** is described by the difference equation

$$y(n) = - a_1y(n-1)-a_2y(n-2)+b_0x(n)+b_1x(n-1)+b_2x(n-2) \quad (2.5.14)$$

If $N = 2, M = 2$ shown in fig 2.35a

If $a_1 = a_2 = 0$ shown in fig.2.35b

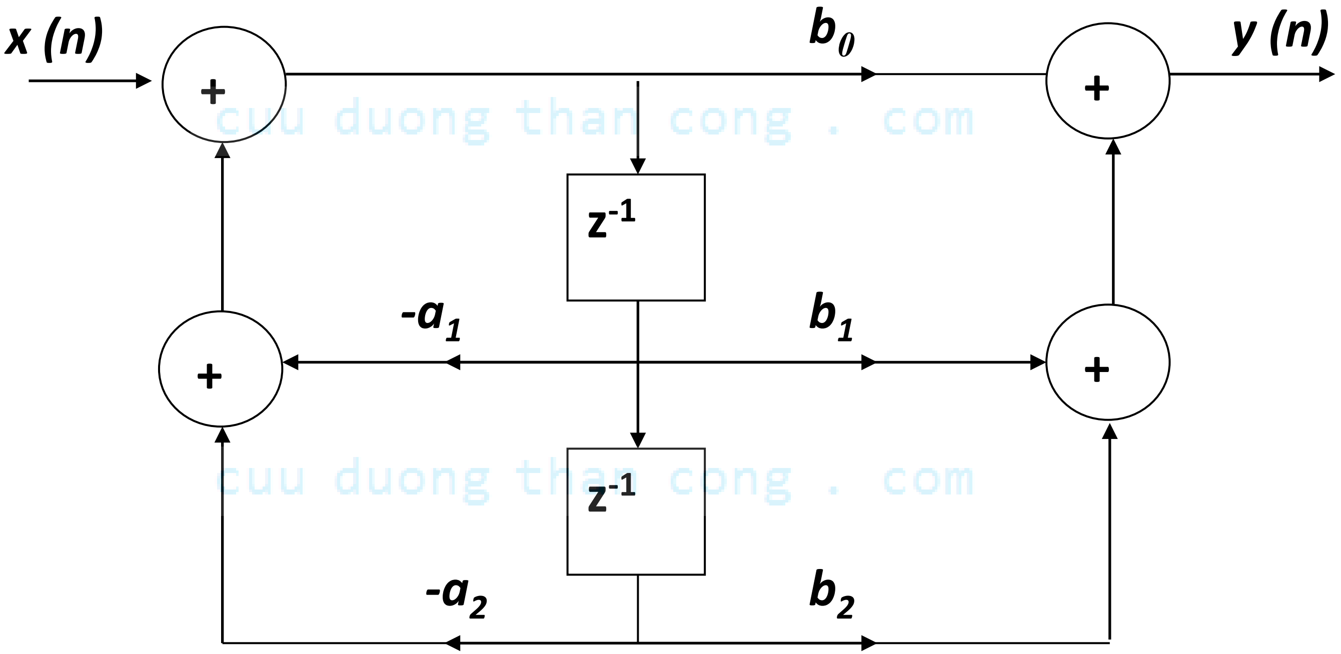
$$y(n) = b_0x(n) + b_1x(n - 1) + b_2x(n - 2) \quad (2.5.15)$$

+ If $b_1 = b_2 = 0$ shown in fig.2.35c

$$y(n) = -a_1y(n - 1) + a_2y(n - 2) + b_0x(n) \quad (2.5.16)$$

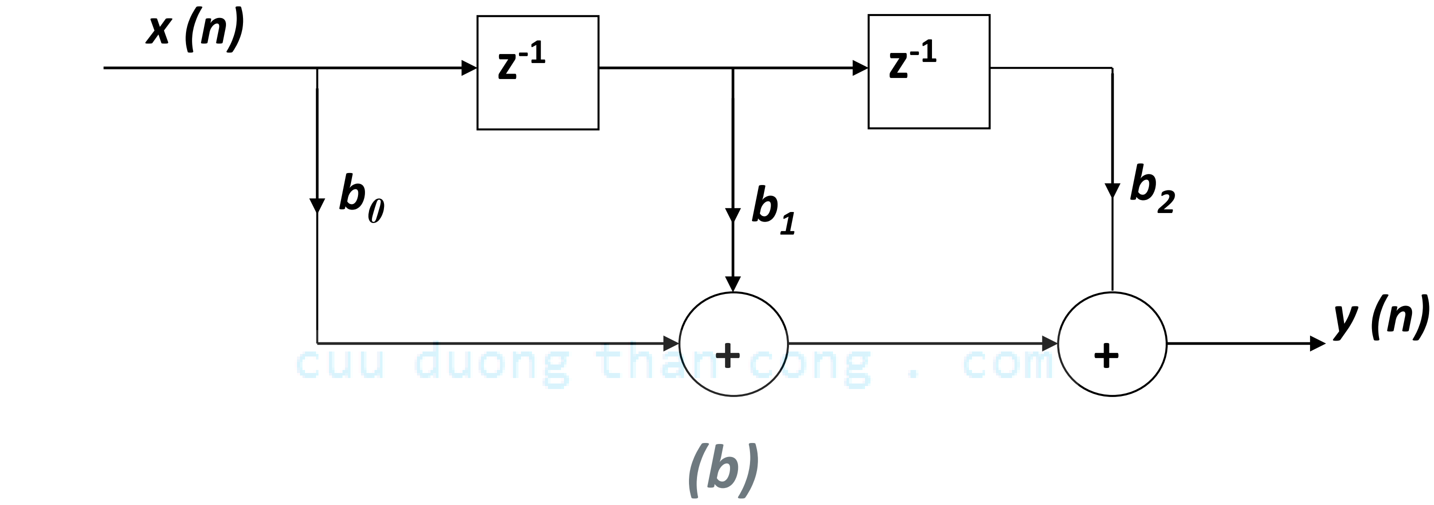
2.5.1 Structures for the Realization of Linear Time-invariant System

Figure 2.35 Structures for the realization of second-order system : (a) general second-order system; (b) FIR system; (c) "purely recursive system".

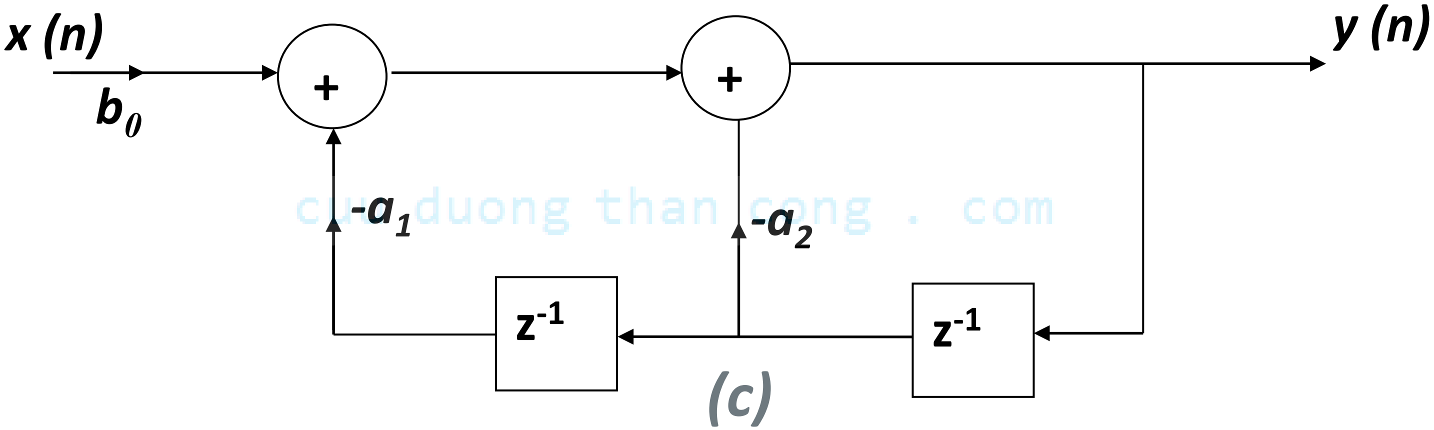


(a)

2.5.1 Structures for the Realization of Linear Time-invariant System



(b)



(c)

2.5.2 Recursive and nonrecursive realizations of FIR Systems

A causal recursive system is described by an input-output equation of the form

$y(n) = F[\, y(n-1), \ldots, y(n-N), x(n), \ldots, x(n-N)]$

(2.5.17)

For *a linear time-invariant system* specifically, by the difference equation

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k) \qquad (2.5.18)$$

2.5.2 Recursive and nonrecursive realizations of FIR Systems

Causal nonrecursive system don’t depend on past values the output

$y(n) = F[\, x(n), x(n-1), x(n), \ldots.., x(n-M)]$

(2.5.19)

The nonrecursive and FIR system specifically with $a_k = 0$ for $k = 1,2,\ldots N.$

$$y(n) = \sum_{k=0}^{M} b_k x(n-k) \qquad (2.5.20)$$

2.5.2 Recursive and nonrecursive realizations of FIR Systems

Suppose that we have an FIR system of the form:

$$y(n) = \frac{1}{M+1}\sum_{k=0}^{M} x(n-k) \qquad (2.5.21)$$

for computing the *moving average* of a signal *x(n)*.

With impulse response

$$h(n) = \frac{1}{M+1} \qquad 0 \le n \le M$$

$$y(n) = \frac{1}{M+1}\sum_{k=0}^{M} x(n-1-k) + \frac{1}{M+1}[x(n) - x(n-1-M)]$$

$$= \quad y(n-1) + \frac{1}{M+1}[X(n) - x(n-1-M)] \qquad (2.5.22)$$

2.5.2 Recursive and nonrecursive realizations of FIR Systems

Figure 2.36 Nonrecursive realization of an FIR moving average system.

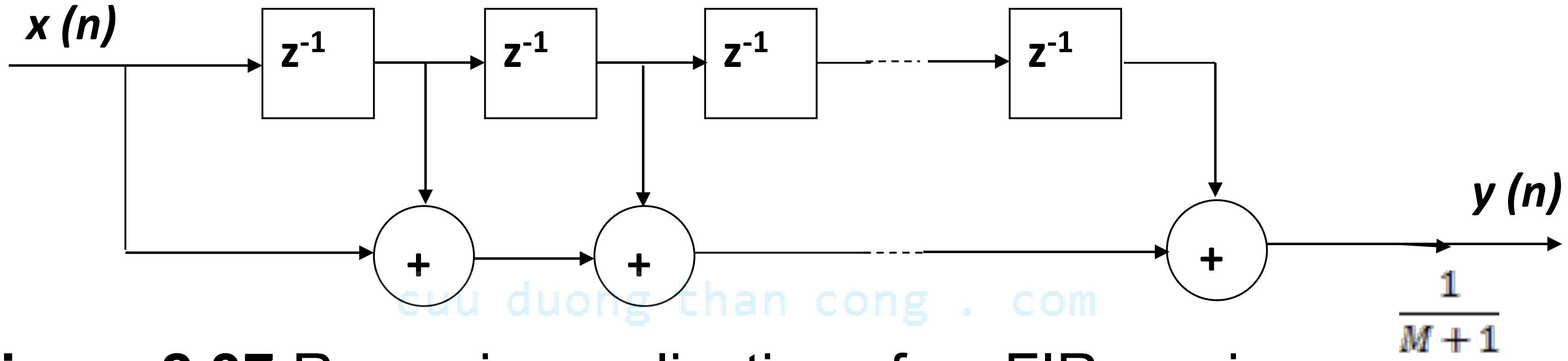
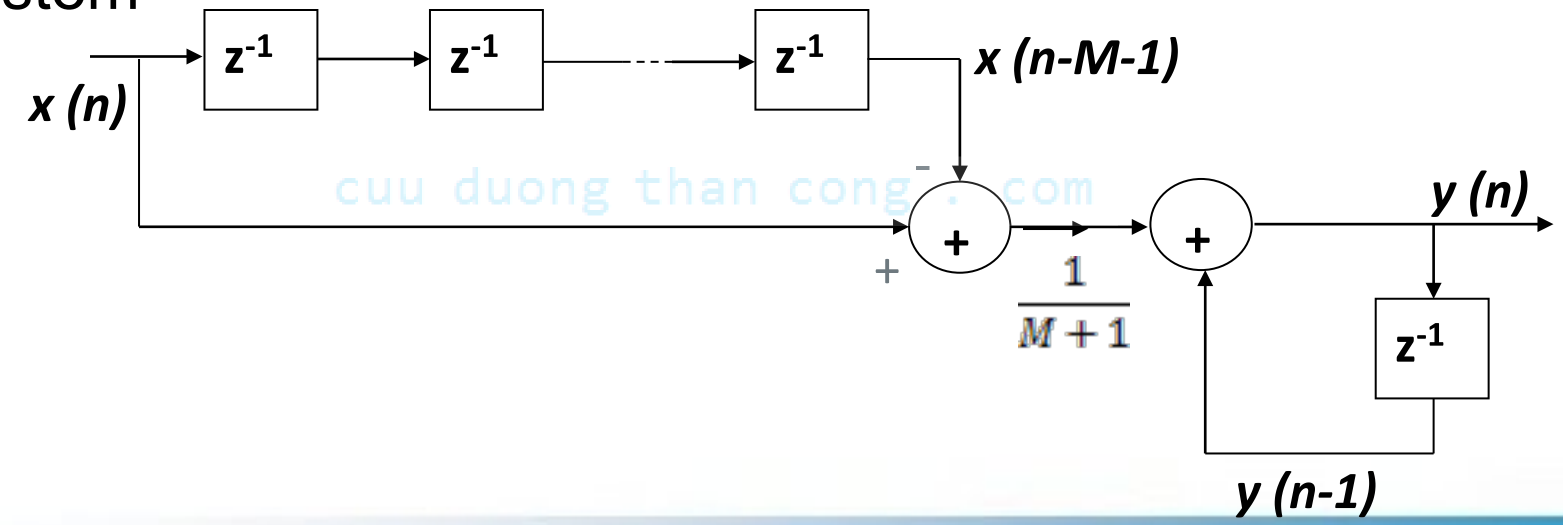


Figure 2.37 Recursive realization of an FIR moving average system



2.6 Correlation of discrete- time signals

+ The ***correlation*** between the two signal is to measure the degree to which the two signal are *similar* and *to extract* some information .

+ For example – we can represent the received signal sequence form Fig.2.38 as

$$y(n) = \alpha x(n - D) + w(n) \qquad (2.6.1)$$

where:

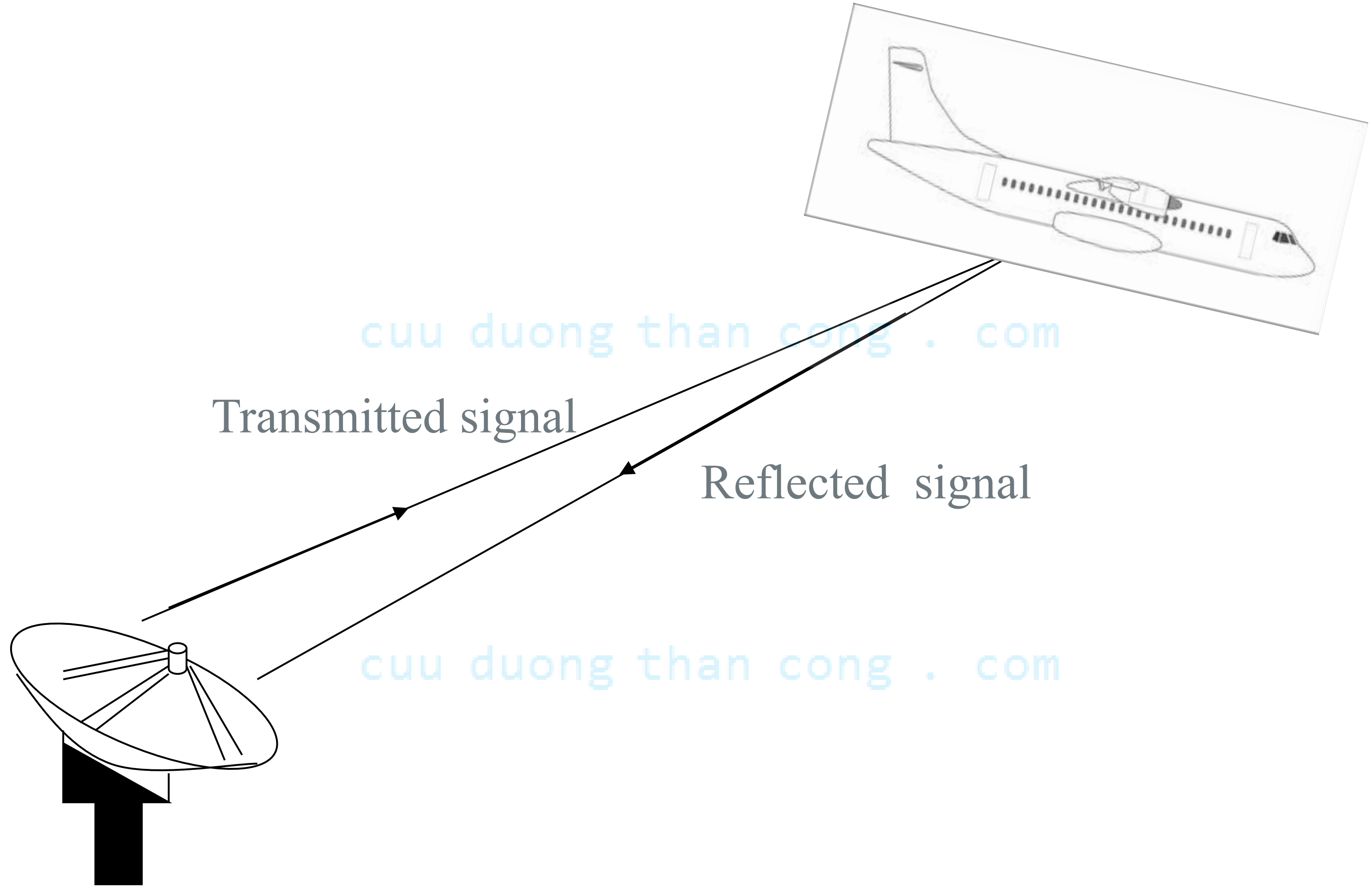
α is some attenuation factor representing the signal loss involved in the round-trip transmission of the system $x(n)$.

D is the round-trip delay, an integer multiple

$w(n)$ represents the additive noise.

2.6 Correlation of discrete- time signals

Figure 2.38 Radar target detection



2.6 Correlation of discrete- time signals

The problem is to compare *y(n)* and *x(n)* to determine

1/ If a target is present .

2/ Time delay *D* and compute the distance to the target.

Correlation provides us with a means for extracting this important in formation from *y(n)*.

2.6.1 Crosscorrelation and Autocorrelation Sequences

The *Crosscorrelation* of *x(n)* and *y(n)* is a sequence $r_{xy}(l)$, which is defined as

$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} x(n)y(n-l), \qquad l = 0, \pm 1, \pm 2, \ldots \qquad (2.6.3)$$

$$or \quad r_{xy}(l) = \sum_{n=-\infty}^{\infty} x(n+l)y(n), \quad l = 0, \pm 1, \pm 2, \ldots \qquad (2.6.4)$$

The index *l* is the (time) shift (or *lag*) parameter

xy indicate the sequences being correlated.

2.6.1 Cross correlation and Autocorrelation Sequences

If we reverse the roles of *x(n)* and *y(n)*:

$$r_{yx}(l) = \sum_{n=-\infty}^{\infty} y(n)x(n-l) \qquad (2.6.5)$$

or

$$r_{yx}(l) = \sum_{n=-\infty} x(n+l)y(n) \qquad (2.6.6)$$

By comparing (2.6.3) with (2.6.6) or (2.6.4) with (2.6.5), we conclude that.

$$r_{xy}\,(l) = r_{yx}\,(-l) \qquad (2.6.7)$$

+ The *convolution* of *x(n)* with *y(-n)* yields the cross correlation r_{xy} (l), that is

$$r_{xy}\,(l) = x\,(l)^{*}\, r_{yx}\,(-l) \qquad (2.6.8)$$

2.6.1 Cross correlation and Autocorrelation Sequences

In the special case where *y(n)* = *x(n)* we have the ***autocorrelation*** of *x(n)*,

$$r_{xx}(l) = \sum_{n=-\infty}^{\infty} x(n)x(n-l) \qquad (2.6.9)$$

or

$$r_{xx}(l) = \sum_{n=-\infty}^{\infty} x(n+l)x(n) \qquad (2.6.10)$$

2.6.1 Crosscorrelation and Autocorrelation Sequences

If *x(n)* and *y(n)* are causal sequence of length ***N***, the ***crosscorrelation***, ***autocorrelation*** sequence may be expressed as

$$r_{xy}(l) = \sum_{n=l}^{N-|k|-l} x(n)y(n-l) \qquad (2.6.11)$$

and

$$r_{xx}(l) = \sum_{n=i}^{N-|k|-l} x(n)x(n-l) \qquad (2.6.12)$$

where *i = l*, *k = 0* for *l* ≥ *0*, and *i = 0, k = l* for *l* < *0*.

2.6.2 - Properties of the Autocorrelation and Crosscorreletion Sequences

Assume we have two sequences *x(n)* and *y(n)* with finite energy from which we form the linear combination

$$y(n) = ax(n) + by(n - l)$$

The ***energy*** in this signal is

$$\sum_{n=-\infty}^{\infty}[ax(n)+by(n-l)]^2 = a^2\sum_{n=-\infty}^{\infty}x^2(n)+b^2\sum_{n=-\infty}^{\infty}y^2(n-l)$$

$$+2ab\sum_{n=-\infty}^{\infty}x(n)y(n-l) \qquad (2.6.13)$$

$$= a^2r_{xx}(0) + b^2r_{yy}(0) + 2abr_{xy}(l)$$

2.6.2 - Properties of the Autocorrelation and Crosscorreletion Sequences

Which are the *energies* of *x(n)* and *y(n)*,

$a^2 r_{xx}(0) + b^2 r_{yy}(0) + 2ab r_{xy}(l) \geq 0$ (2.6.14)

$$r_{xx}(0)\left(\frac{a}{b}\right)^2 + 2r_{yy}(l)\left(\frac{a}{b}\right) + r_{yy}(0) \geq 0$$

$$|r_{xy}(l)| \leq \sqrt{r_{xx}(0) r_{yy}(0)} = \sqrt{E_x E_y}$$ (2.6.15)

If $y(n) = x(n)$, then

$| r_{xx}(l)| \leq r_{xx}(0) = E_x$ (2.6.16)

Autocorrelation sequences of signal attains its ***maximum value*** at zero lag.

2.6.2 Properties of the Autocorrelation and Crosscorreletion Sequences

The *normalized autocorrelation sequences* is

$$\rho_{xx}(l) = \frac{r_{xx}(l)}{r_{xx}(0)} \qquad (2.6.17)$$

The *normalized crosscorrelation sequences*

$$\rho_{xy}(l) = \frac{r_{xy}(l)}{\sqrt{r_{xx}(0)r_{yy}(0)}} \qquad (2.6.18)$$

The crosscorrelation sequences satisfies the property

$r_{xy}(l) = r_{yx}(-l)$ with $y(n) = x(n)$,

then

$r_{xx}(l) = r_{xx}(-l)$ (2.6.19)

2.6.3 Correlation of Periodic Sequences

Let *x(n)* and *y(n)* be two *power signals*. Their *crosscorrelation* sequences is defined as

$$r_{xy}(l) = \lim_{M\to\infty} \frac{1}{2M+1} \sum_{n=-M}^{M} x(n)y(n-l) \quad (2.6.22)$$

If   *x(n) = y(n)*, the *autocorrelation* sequences:

$$r_{xx}(l) = \lim_{M\to\infty} \frac{1}{2M+1} \sum_{n=-M}^{M} x(n)x(n-l) \quad (2.6.23)$$

2.6.3 Correlation of Periodic Sequences

If *x(n)* and *y(n)* are two periodic sequences, each with period *N*, so that.

$$r_{xy}(l) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)y(n-l) \qquad (2.6.24)$$

$$r_{xx}(l) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)x(n-l) \qquad (2.6.25)$$

For example *y(n) = x(n) – w(n)* 2.6.26)

x(n) - is a periodic sequences *N* ;

w(n) - is an additive random interference.

2.6.3 Correlation of Periodic Sequences

Suppose that, we observe *M* samples of *y(n)* *0 ≤ n ≤ M-1* where *M > N*,

$$r_{yy}(l) = \frac{1}{M}\sum_{n=0}^{M-1} y(n)y(n-l) \qquad (2.6.27)$$

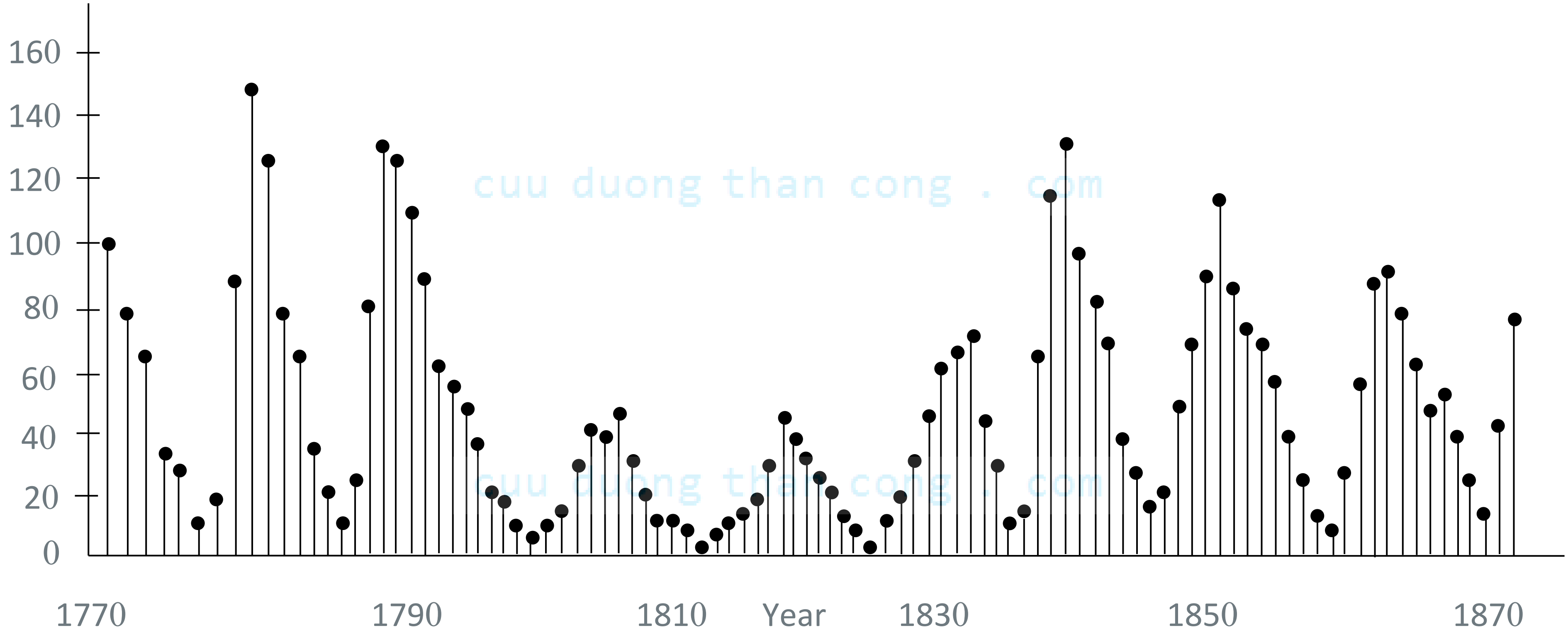
$$r_{yy}(l) = \frac{1}{M}\sum_{n=0}^{M-1} [x(n)+w(n)][x(n-l)+w(n-l)$$

$$= r_{xx}(l) + r_{xw}(l) + r_{wx}(l) + r_{ww}(l) \qquad (2.6.28)$$

+ Only r_{xx} *(l)* has large peaks for *l > 0*, to allows us to ***detect*** the *presence* of the periodic signal *x(n)* buried in the interference *w(n)* and to identify its period.

2.6.3 Correlation of Periodic Sequences

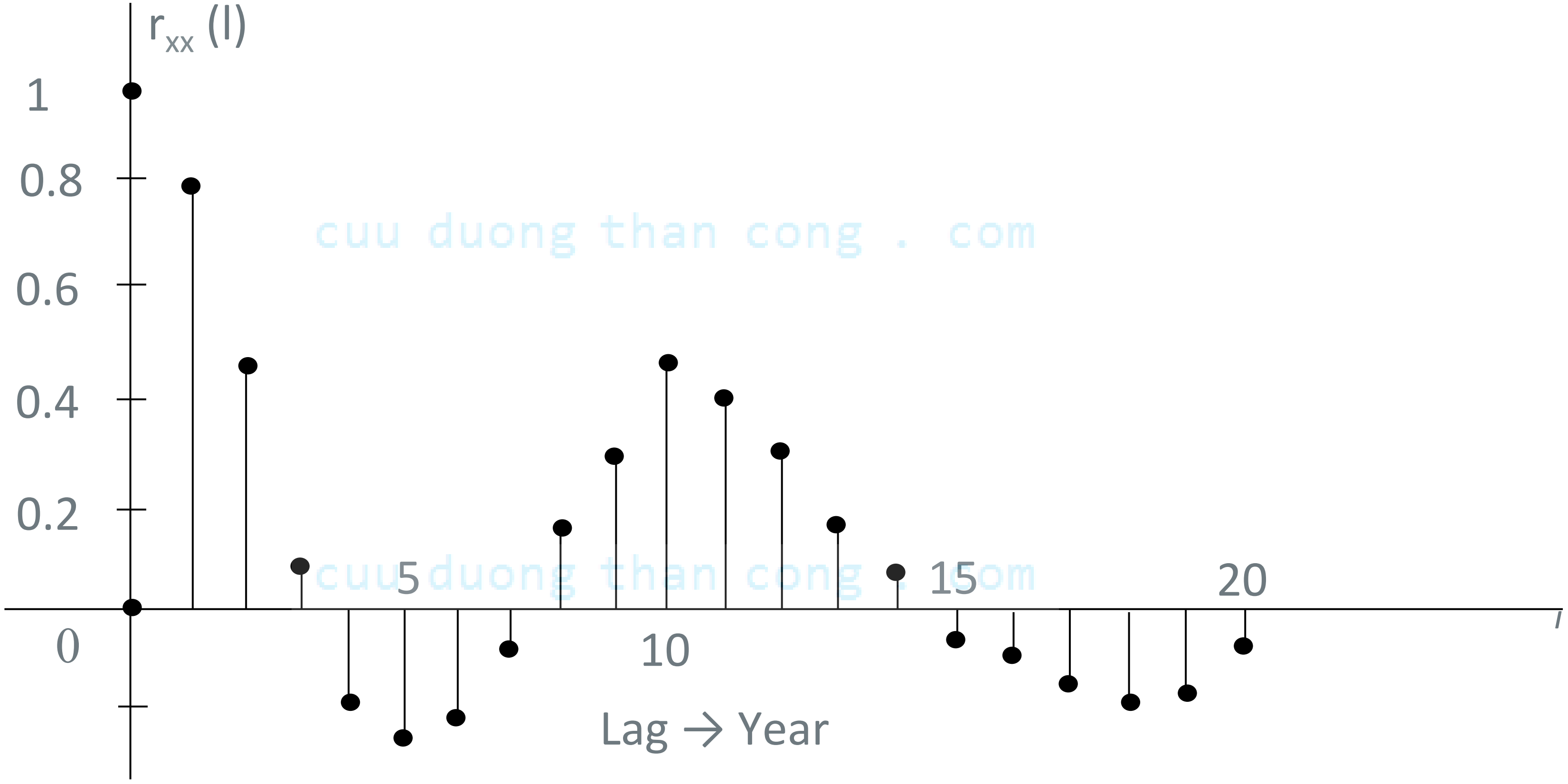
+ An example illustrates the use of *autocorrelation* to identify a hidden periodicity in an observed physical signal in fig.2.40



(a)

2.6.3 Correlation of Periodic Sequences

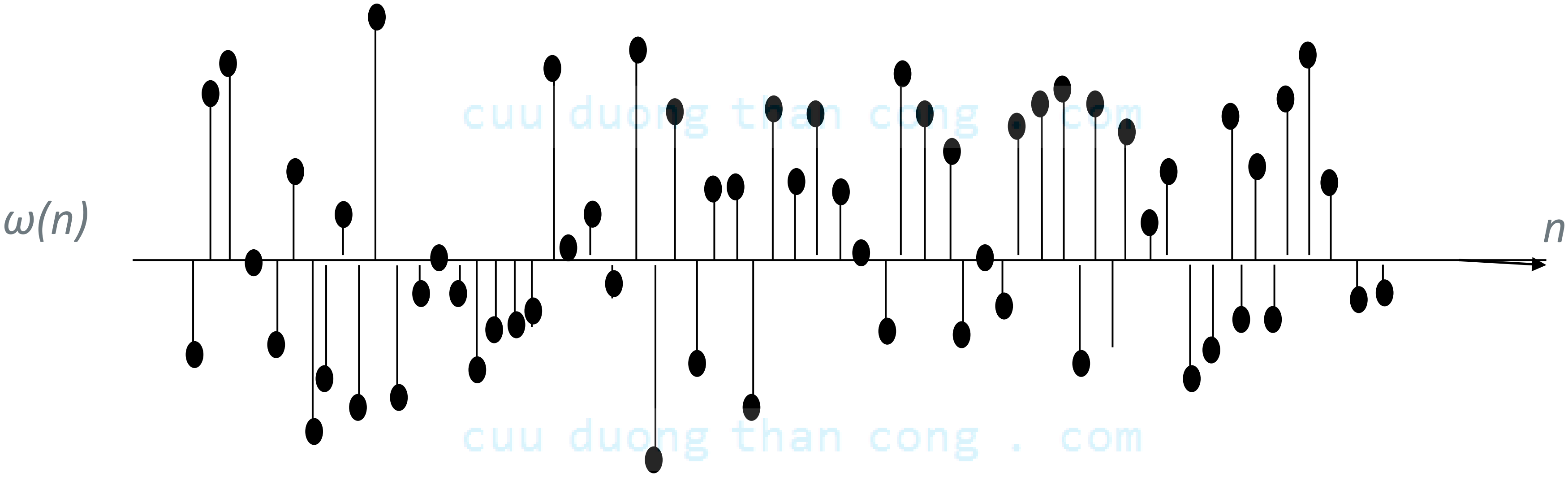
Figure 2.40: Identification of periodicity in the Wőlfer sunspot numbers: (a) annual Wőlfer sunspot numbers; (b) autocorrclation sequence.



(b)

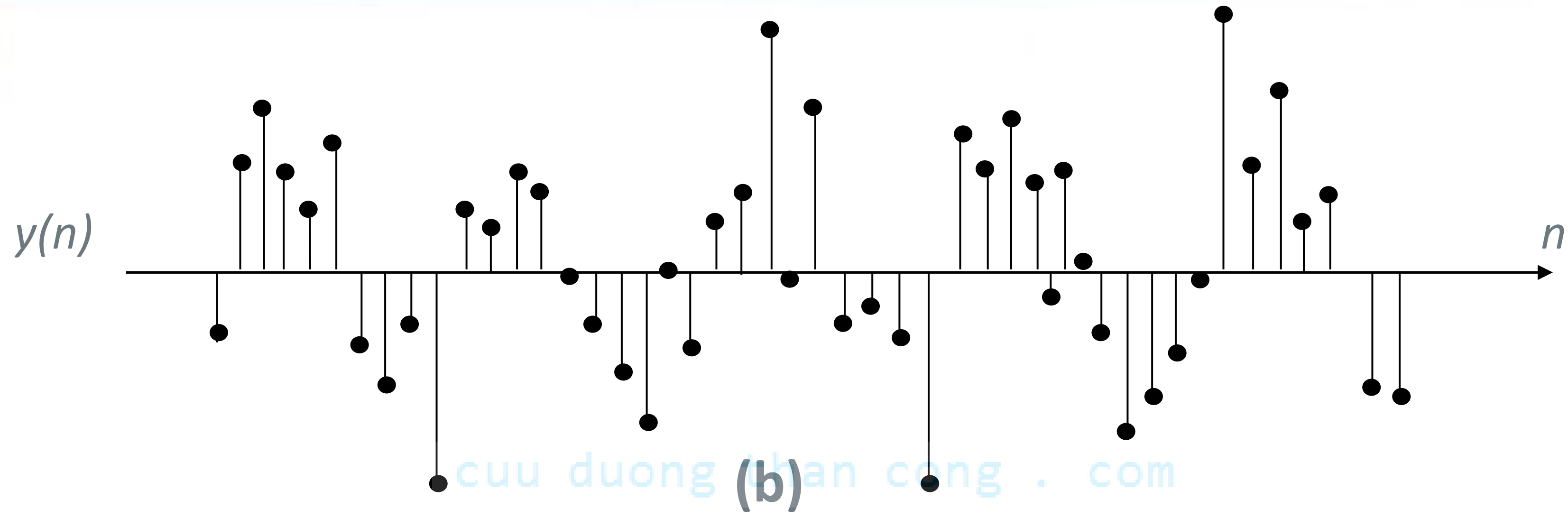
2.6.3 Correlation of Periodic Sequences

Figure 2.41. Use of autocorrelation to detect the presence of a *periodic signal* corrupted by noise.

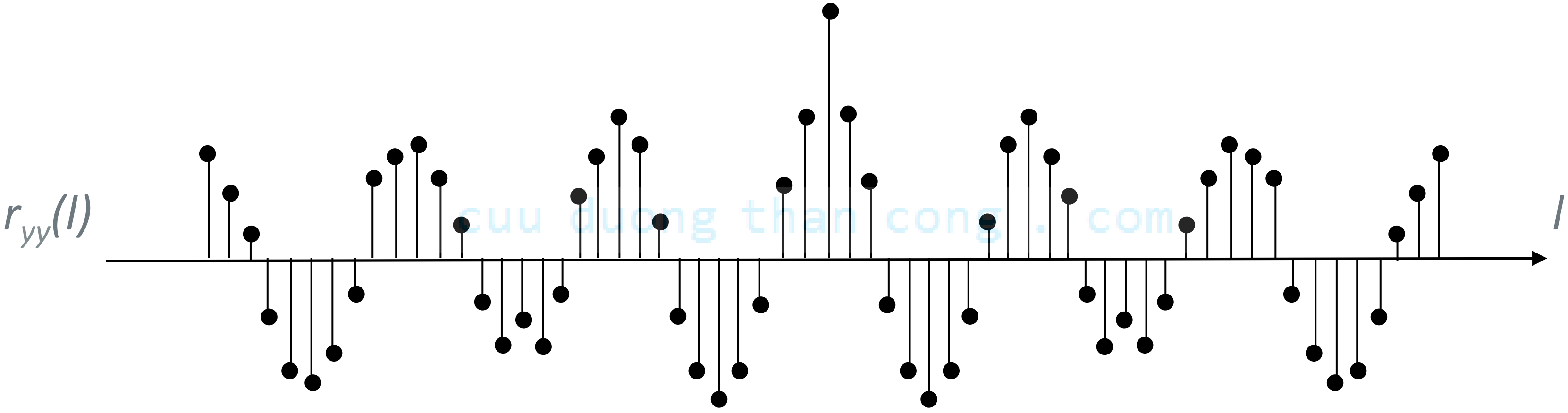


(a)

2.6.3 Correlation of Periodic Sequences



(b)



(c)

2.6.4 Computation of correlation sequences

To compute the *crosscorrelation sequences* of two finite duration signal

x(n) $0 \le n \le N\text{-}1$ and *y(n)* $0 \le n \le M\text{-}1$

$$r_{xy}(l) = \begin{cases} \displaystyle\sum_{n=l}^{M-l+1} x(n)y(n-l), & 0 \le l \le N-M \\ \displaystyle\sum_{n=l}^{N-l} x(n)y(n-l), & N-M < l \le N-l \end{cases} \qquad (2.6.30)$$

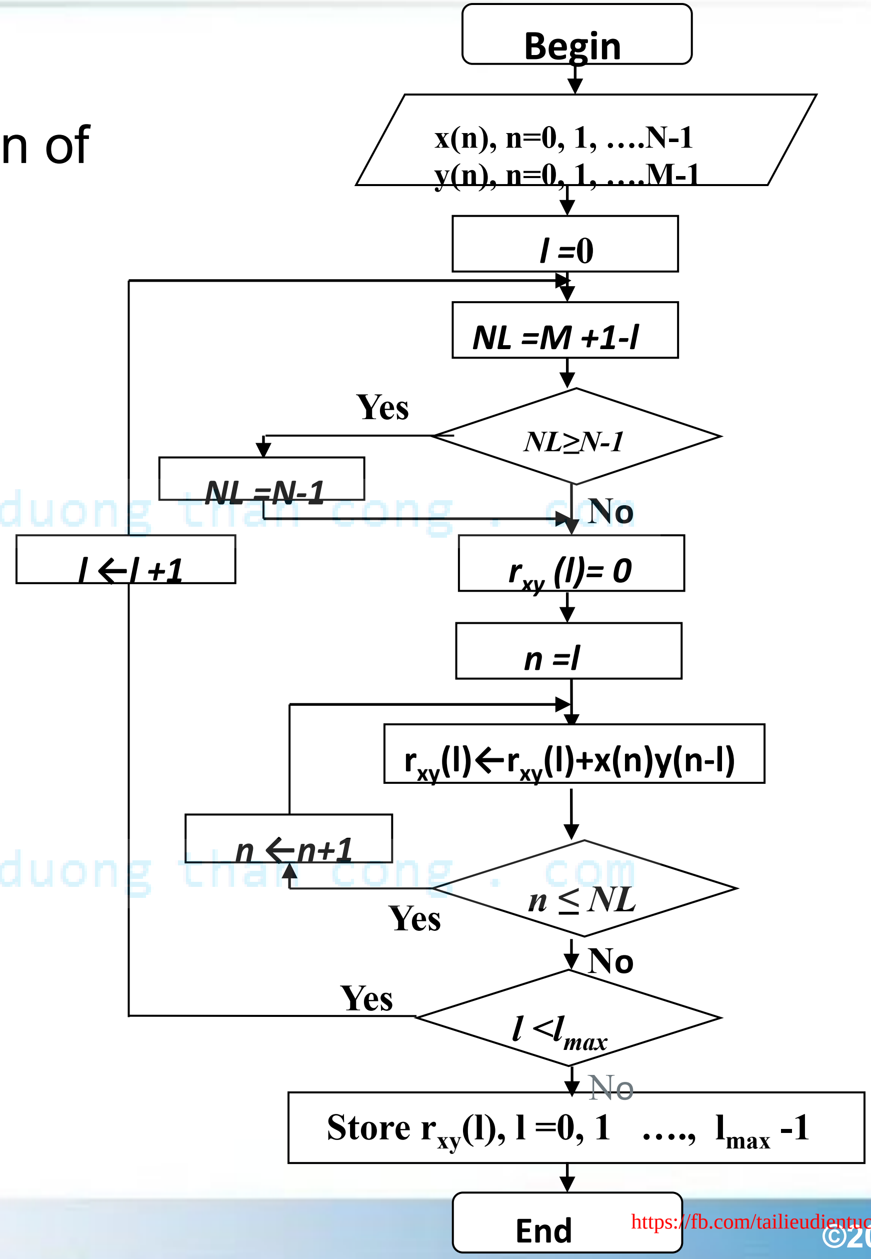
if *M* > *N*, then

$$r_{xy}(l) = \sum_{n=l}^{N-l} x(n)y(n-l), \; 0 \le l \le N-1 \quad (2.6.31)$$

Means of the following simple logarithm is illustrated in the flowchart in Fig 2.43

2.6.4 Computation of correlation sequences

Figure 2.43. Flowchart for software implementation of crosscorrelation



2.6.5 Input – Output Correlation Sequences

X(n) with $r_{xx}(l)$ is applied to an LTI system with impulse response *h(n)*.

$$r_{yx}(l) = h(l) * r_{xx}(l) \qquad (2.6.32)$$

Using (2.6.8) with *x(n) = y(n)* and properties of convolution.

$$r_{yy}(l) = y(l) * y(-l) = [h(l) *x(-l)] * [x(l) * x(-l)]$$

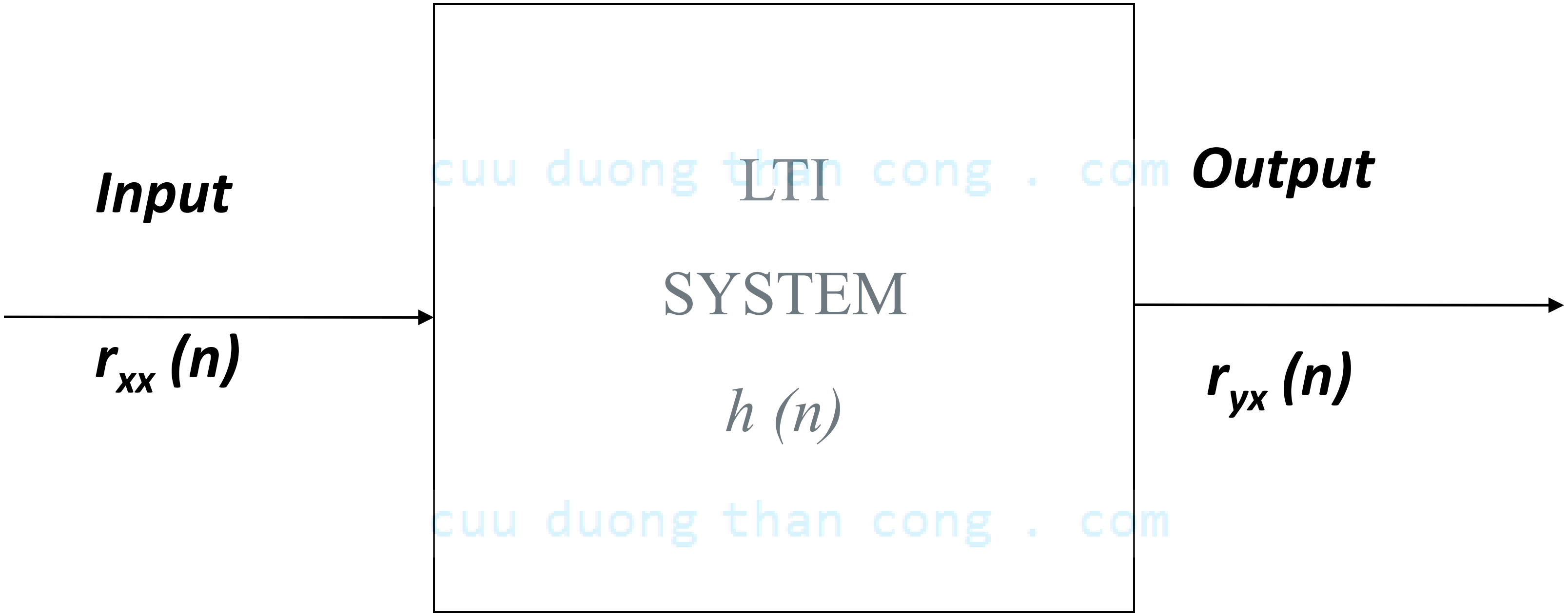
$$= r_{hh}(l) * r_{xx}(l) \qquad (2.6.33)$$

The autocorrelation $r_{hh}(l)$ of the impulse response *h(n)* exits if the system is stable. For *l = 0* we obtain

$$r_{yy}(0) = \sum_{k=-\infty}^{\infty} r_{hh}(k) r_{xx}(k) \qquad (2.6.34)$$

2.6.5 Input – Output Correlation Sequences

Figure 2.44 Input-output relation for crosscorrelation $r_{yx}(n)$



Problems 2.1, 2.2, 2.7, 2.18, 2.27, 2.31, 2.32, 2.37, 2.41, 2.42, 2.44, 2.59, 2.60