

TN 412: Digital Signal Processing

3: Discrete-Time Signals & Systems

Digital Signal Processing

Discrete-Time Signals & Systems

3.1. Analysis of DT Linear Time Invariant Systems

Motivation for the emphasis on the study of LTI

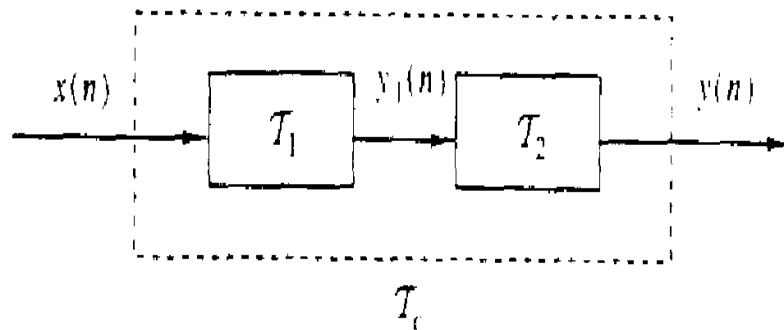
- **There is a large collection of mathematical techniques that can be applied to the analysis of LTI systems.**
- **Many practical systems are either LTI systems or can be approximated by LTI systems.**
- **We demonstrate that such systems are characterized in the time domain simply by their response to a unit impulse sequence**
- **We also demonstrate that any arbitrary input signal can be decomposed and represented as a weighted sum of unit sample sequences.**

Digital Signal Processing

Discrete-Time Signals & Systems

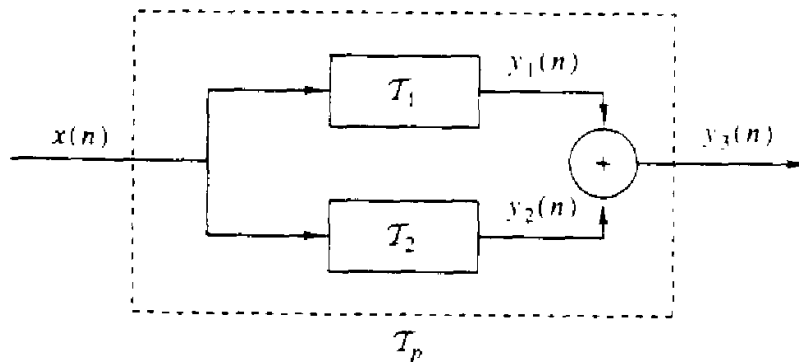
3.3. Analysis of DT Linear Invariant Systems

System Interconnections:



$$y_1(n) = T_1\{x(n)\}$$
$$y(n) = T_2\{y_1(n)\} = T_2\{T_1\{x(n)\}\}$$
$$T_c = T_2 T_1 \neq T_1 T_2$$

Specifically, for **LTI** Systems: $T_2 T_1 = T_1 T_2$



$$y_1(n) = T_1\{x(n)\} \quad y_2(n) = T_2\{x(n)\}$$
$$y_3(n) = y_1(n) + y_2(n) = T_1\{x(n)\} + T_2\{x(n)\}$$
$$y(n) = (T_2 + T_1)x(n)$$

$$T_p = (T_2 + T_1)$$

Digital Signal Processing

Discrete-Time Signals & Systems

3.1. Analysis of DT Linear Invariant Systems

.....•

In this section we will analyze the Linear Time-Invariant (LTI) systems.

3.3.1. System Analysis Techniques:

Two methods are presented in for analyzing the behavior/response of a system to a given input:

- Direct Solution of the Input-Output Equation (or **difference equation**).
- Signal Decomposition (**Convolution**).
 - Decompose the input signal into sum of elementary signals.

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

3.3.1. System Analysis Techniques:

➤ System Input output equation:

The general input output equation for any system:

$$y(n) = F[y(n-1), y(n-2), \dots, y(n-N), x(n), x(n-1), \dots, x(n-M)]$$

where $F[\]$ denotes some functions of the quantities.

We can rewrite the general input output equation as

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

where a_k and b_k
are constant
parameters

If the system is **casual** $L = 0$, so the equation:

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

Note that both **a** and **b**
vary with time

This system is linear , **time-variant** system. This system called Adaptive Linear System.

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

3.3.1. System Analysis Techniques:

Linear Time-Invariant (LTI) Systems:

If a and b are constant over time, then the previous equation further simplifies into the general equation for causal, Linear, Time-**Invariant** (LTI) Systems:

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

Note: a_k and b_k are independent of time (n) or simply constant for all time n .

Input, Output Equation is called **Difference Equation**.
The **order** for the LTI system is N .

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

3.3.1. System Analysis Techniques:

Non-linear Difference Equation:

Difference Equations can also describe non linear systems:

$$y(n) = |x(n)|$$

$$y(n) = 2 \cdot x(n) - 3 \cdot x^2(n)$$

$$y(n) = \sqrt{x(n)} + 3 \cdot \log(x(n))$$

$$y(n) = A \cdot x(n) + B$$

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

3.3.1. System Analysis Techniques:

➤ Signal Decomposition:

Decompose input into weighted basis functions:

$$x(n) = \sum_k c_k \cdot x_k(n)$$

Generally we choose the basis functions and compute C_k .

Decomposition to Impulses:

Consider $x(n) = \{3, 5, 2, 1\}$ Decompose input sequence to sum of unit samples.

$$x_1(n) = \{3, 0, 0, 0\} = 3 \cdot \delta(n)$$

$$x_2(n) = \{0, 5, 0, 0\} = 5 \cdot \delta(n-1)$$

$$x_3(n) = \{0, 0, 2, 0\} = 2 \cdot \delta(n-2)$$

$$x_4(n) = \{0, 0, 0, 1\} = \delta(n-3)$$

$$x(n) = x_1(n) + x_2(n) + x_3(n) + x_4(n)$$

$$x(n) = 3 \cdot \delta(n) + 5 \cdot \delta(n-1) + 2 \cdot \delta(n-2) + \delta(n-3)$$

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

3.3.1. System Analysis Techniques:

Decomposition to Impulses:

We can write the following equation for the previous expression:

$$x(n) = \sum_{k=0}^3 x(k) \cdot \delta(n-k)$$

In general form

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

where $c_k = x(k)$, $x_k(n) = \delta(n-k)$

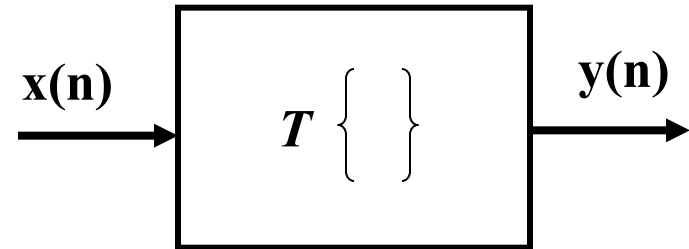
Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Convolution Summation:

Consider the following system:



Recall input can be decomposed as follows:

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

$$\begin{aligned} \text{Therefore: } y(n) = \mathcal{T}[x(n)] &= \mathcal{T}\left[\sum_{k=-\infty}^{\infty} x(k)\delta(n-k)\right] = \sum_{k=-\infty}^{\infty} x(k)\mathcal{T}[\delta(n-k)] \\ &= \sum_{k=-\infty}^{\infty} x(k)h(n, k) \end{aligned}$$

If \mathcal{T} system is Linear:

$$\text{Impulse Response: } y(n, k) \equiv h(n, k) = \mathcal{T}[\delta(n-k)]$$

Useless, infinite set of responses:
(dependent on both n and k)

$$y(n) = \sum_{k=-\infty}^{\infty} x(k) \cdot h_k(n)$$

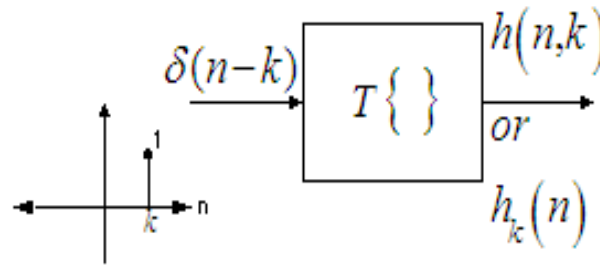
Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Convolution Summation:

Impulse Response



Note – if system is time invariant

$$h(n,k) = T\{\delta(n-k)\} \quad \text{and} \quad h(n-k) = T\{\delta(n-k)\}$$

Therefore
$$h(n,k) = h(n-k)$$

For LTI systems, the convolution sum is as follows:

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

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Convolution Summation Calculation:

Graphical method:

To calculate $y(n_0)$ using the convolution summation, do the following steps:

- ✓ **Folding:** Fold $h(k)$ about $k = 0$ to obtain $h(-k)$.
- ✓ **Shifting:** Shift $h(-k)$ by n_0 to the right (left) if n_0 is positive (negative), to obtain $h(n_0 - k)$
- ✓ **Multiplication:** Multiply $x(k)$ by $h(n_0 - k)$ to obtain the product sequence $x(k)h(n_0 - k)$
- ✓ **Summation:** Sum all the values of the product sequence $x(k)h(n_0 - k)$ to obtain the value of the output at time $n = n_0$

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Graphical Convolution Example

Consider the following input and impulse response

$$x(n) = \{1, 2, 3, 1\}$$

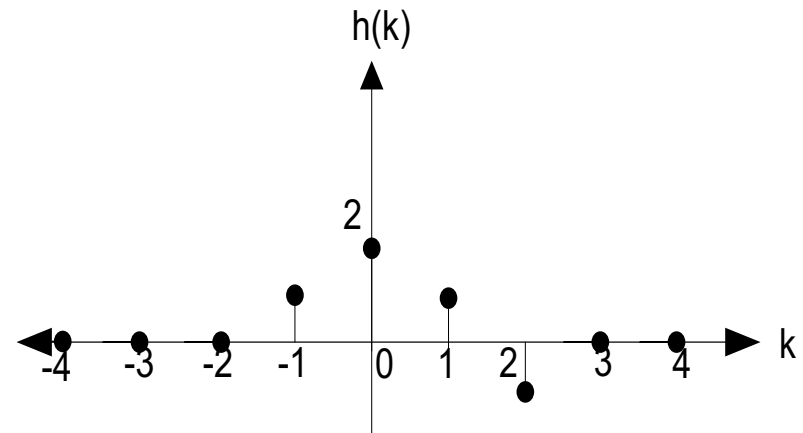
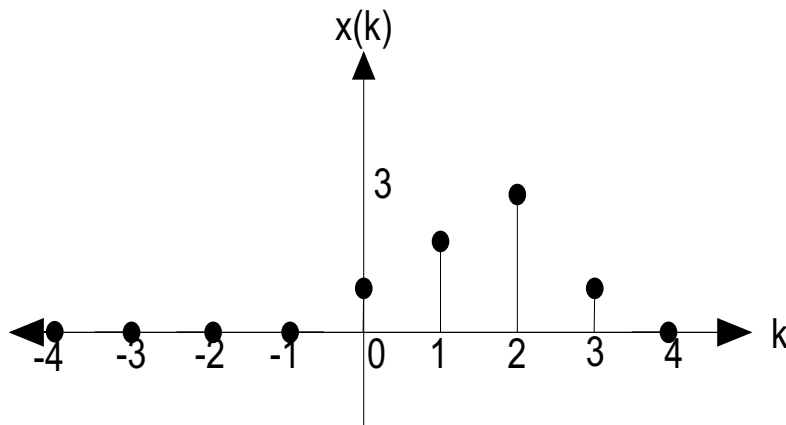


Input Sequence

$$h(n) = \{1, 2, 1, -1\}$$



Impulse Response

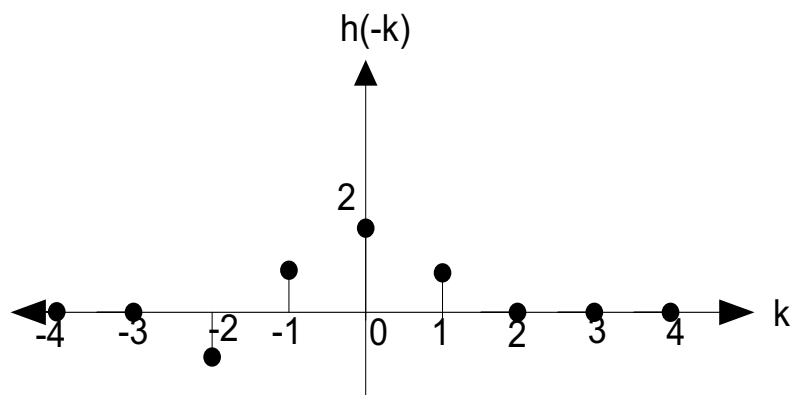


Digital Signal Processing

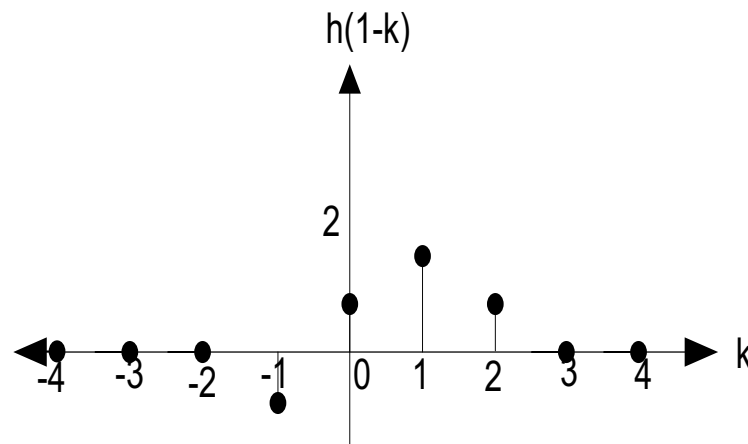
Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Graphical Convolution Example Cont.



Folded Impulse



Shifted Impulse

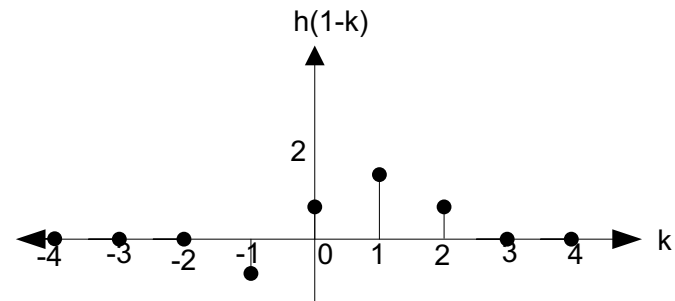
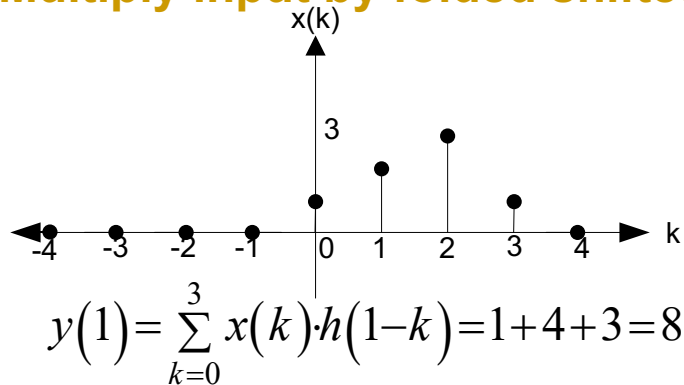
Digital Signal Processing

Discrete-Time Signals & Systems

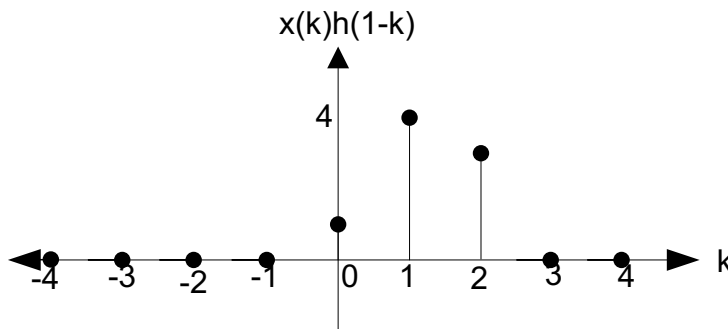
3.3. Analysis of DT Linear Invariant Systems

Graphical Convolution Example Cont.

Multiply input by folded shifted $h(n)$:



Product Sequence – $x(k)h(n-k)$:



Sum of Product Sequence:

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Mathematical Convolution Example

Consider the following input and impulse response

$$x(n) = [5, 2, 4, -1] \quad h(n) = [5, 2, 4, -1]$$

$$y[0] \quad y[1] \quad y[2] \quad y[3] \quad y[4] \quad y[5] \quad y[6]$$

$$25 \quad 20 \quad 44 \quad 6 \quad 12 \quad -8 \quad 1$$

$h(k)$	5	2	4	-1
$x(k)$				
5	25	10	20	-5
2	10	4	8	-2
4	20	8	16	-4
-1	-5	-2	-4	1

$$y(n) = [25, 20, 44, 6, 12, -8, 1]$$

Note: The convolution of two **FINITE-LENGTH** sequences is that if $x(n)$ is of length L_1 and $h(n)$ is of length L_2 , $y(n) = x(n)*h(n)$ will be of length :

$$L = L_1 + L_2 - 1$$

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Convolution Properties

Convolution Symbol:
$$y(n) = \sum_{k=-\infty}^{\infty} x(k) \cdot h(n-k) = x(k) * h(n)$$

Convolution is Commutative:
$$x(k) * h(n) = h(n) * x(k)$$

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

Convolution is Distributive:

$$h(n) * [x_1(n) + x_2(n)] = h(n) * x_1(n) + h(n) * x_2(n)$$

Convolution is Associative:

$$[x_1(n) * x_2(n)] * x_3(n) = x_1(n) * [x_2(n) * x_3(n)]$$

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Useful Geometric Summation Formulas

$$\sum_{n=0}^{N-1} a^n = \frac{1 - a^N}{1 - a}$$

$$\sum_{n=0}^{N-1} na^n = \frac{(N-1)a^{N+1} - Na^N + a}{(1-a)^2}$$

$$\sum_{n=0}^{N-1} n = \frac{1}{2}N(N-1)$$

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

$$\sum_{n=0}^{\infty} na^n = \frac{a}{(1-a)^2} \quad |a| < 1$$

$$\sum_{n=0}^{N-1} n^2 = \frac{1}{6}N(N-1)(2N-1)$$

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Causal Linear Time-Invariant Systems

❖ An LTI system is Causal IFF its impulse response is 0 for negative values of n .

$$h(n) = 0, \text{ for } n < 0$$

❖ Convolution sum for Causal, LTI system reduces to the following:

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

❖ Also, for Causal LTI: $y(n) = \sum_{k=-\infty}^n x(k) \cdot h(n-k)$

❖ If the input is **Causal**, LTI system is **Causal** then the Convolution further simplifies to:

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

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Causal Linear Time-Invariant Systems

If Future Inputs are not needed then Causal LTI system reduces as follows:

For causality, the reference to future inputs must be equal to 0, or simply there must not be future inputs existent in the system's difference equation.

If k is negative then $x(n-k)$ corresponds to future input references

Therefore:

$$\sum_{k=-\infty}^{-1} \overset{\text{future}}{\overset{\text{inputs}}{x(n-k)}} \cdot h(k) = 0$$

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

$$y(n) = \sum_{k=-\infty}^{-1} \overset{\text{future}}{\overset{\text{inputs}}{x(n-k)}} \cdot h(k) + \sum_{k=0}^{\infty} x(n-k) \cdot h(k)$$

Then for the causal system the original equation can be rewritten:

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

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Non-Causal Systems

Non Causal systems will produce an output BEFORE the input is applied.

Consider the following non-causal equation:

$$y(n) = 2 \cdot x(n+1)$$

If the input is applied at $n=0$, then at $n=-1$, we already have an output (before the input is applied):

$$y(-1) = 2 \cdot x(0)$$

The output of course is dependant on a future input (which **does not** make much sense in **real time implementations**).

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Stability of Linear Time-Invariant Systems

❖ BIBO Stable Defined as:

- For all bounded inputs $x(n)$, the outputs $y(n)$ are bounded if the system is BIBO stable:

- $|x(n)| \leq M_x < \infty$ and $|y(n)| \leq M_y < \infty$

- Where $\underline{M_x}$ and $\underline{M_y}$ are finite numbers

❖ Reminder – Triangular or Schwartz

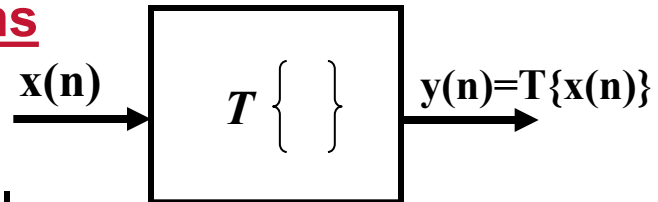
Inequality: $|x_1 + x_2| \leq |x_1| + |x_2|$

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Stability of Linear Time-Invariant Systems

- ❖ Consider the following system: 
- ❖ Note the output is defined as follows:

$$|y(n)| = \left| \sum_{k=-\infty}^{\infty} h(k) \cdot x(n-k) \right| \leq \sum_{k=-\infty}^{\infty} |h(k) \cdot x(n-k)| = \sum_{k=-\infty}^{\infty} |h(k)| \cdot |x(n-k)|$$

- ❖ Assume the input is stable: $|x(n-k)| \leq M_x$
- ❖ The above can then be rewritten as follows:

$$\sum_{k=-\infty}^{\infty} |h(k)| \cdot |x(n-k)| \leq \sum_{k=-\infty}^{\infty} |h(k)| \cdot M_x$$

- ❖ Therefore: $|y(n)| \leq M_x \cdot \sum_{k=-\infty}^{\infty} |h(k)|$

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Stability of Linear Time-Invariant Systems

- ❖ Now, if the system T is stable then the following holds true:

$$|y(n)| \leq M_x \cdot \sum_{k=-\infty}^{\infty} |h(k)| \leq M_y$$

- ❖ Multiplying both sides and further simplifying:

$$\frac{1}{M_x} \cdot M_x \cdot \sum_{k=-\infty}^{\infty} |h(k)| \leq M_y \cdot \frac{1}{M_x}$$

- ❖ Defining new finite integer:

$$M_h \equiv M_y \cdot \frac{1}{M_x}$$

- ❖ Therefore if the system T is stable and the input is bounded, then the following relation must be true:

$$\sum_{k=-\infty}^{\infty} |h(k)| \leq M_h < \infty$$

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Stability of Linear Time-Invariant Systems

- Therefore a stable system's impulse response is absolutely summable:

$$\sum_{k=-\infty}^{\infty} |h(k)| \leq M_h < \infty$$

- That summation may go to **infinity** (the system will be unstable) if any of the following occur:
 - The Summation is Infinite and does not converge
 - The Summation is finite but contains at least one term which is infinite
- If a system is FIR of length N then $h(n)$ is absolutely summable (stable):

- $\sum_{k=0}^N |h(k)| < \infty$ assuming $h(n) \neq \infty$ for all n

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

System with Finite Duration & Infinite Duration Impulse Response:

If impulse response, $h(n)$, is of finite length, then the system is categorized as **finite impulse response – FIR**

➤ Causal FIR System

$$h(n) = 0 \quad n < 0 \text{ and } n \geq M \quad y(n) = \sum_{k=0}^{M-1} h(k)x(n-k)$$

If impulse response, $h(n)$, is of infinite length, then the system is categorized as **infinite impulse response – IIR**

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

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

.....•

- **Recursive** systems use previous outputs to compute present output:

$$y(n) = H \{ y(n-1), \dots, x(n), x(n-1), \dots, x(n+1), \dots \}$$

- ❖ **Non-recursive** systems do NOT use previous outputs to compute present output:

$$y(n) = H \{ x(n), x(n-1), \dots, x(n+1), \dots \}$$

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

.....•

Difference Equations:

- ❖ Convolution summation suggests a means for the realization of the system.
- ❖ In case of FIR systems, such a realization involves additions, multiplications, and a finite number of memory locations and hence FIR can be realized by Convolution summation.
- ❖ If the system IIR, its practical implementations as implied by convolution is clearly impossible, since it requires an infinite number of memory locations, multiplications and additions.
- ❖ IIR systems can be realized by difference equations.
- ❖ IIR systems are useful in a variety of practical applications, including implementation of digital filters and modeling of physical phenomenon and physical systems.

Example:

$$\text{FIR} \quad y(n) = \sum_{k=0}^{M-1} h(k)x(n-k) \qquad \text{IIR} \quad y(n) = \sum_{k=0}^{\infty} h(k)x(n-k)$$

Digital Signal Processing

Discrete-Time Signals & Systems

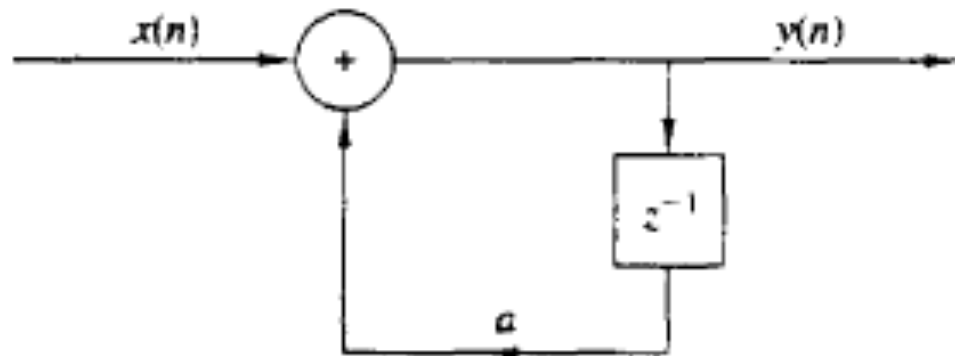
3.3. Analysis of DT Linear Invariant Systems

Difference Equations:

- ❖ System described by constant-coefficient linear difference equations are a subclass of the recursive and non-recursive system.
- ❖ Suppose we have recursive system with an input-output relation :

$$y(n) = ay(n-1) + x(n)$$

- ❖ The system has constant coefficient (independent of time) and can be realized by block diagram as follows:



Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Difference Equations:

LTI System is characterized by a linear Constant-Coefficient difference equation (LCCDE):

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

Order of difference equation is N

Example: $y(n) = \frac{5}{6} \cdot y(n-1) - \frac{1}{6} \cdot y(n-2) + x(n)$

n=0 $y(0) = \frac{5}{6} \cdot y(-1) - \frac{1}{6} \cdot y(-2) + x(0)$

Order: N=2

Mathematical Initial Conditions: $y(-1)$ and $y(-2)$

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

.....•

Difference Equations:

For a given difference equation, there exists as many initial conditions as the order of the difference equation (N initial conditions for a difference equation of order N).

- **Output sequence computed recursively:**

$$n = 1 \quad y(1) = \frac{5}{6} \cdot y(0) - \frac{1}{6} \cdot y(-1) + x(1)$$

$$n = 2 \quad y(2) = \frac{5}{6} \cdot y(1) - \frac{1}{6} \cdot y(0) + x(2)$$

$$n = 3 \quad y(3) = \frac{5}{6} \cdot y(2) - \frac{1}{6} \cdot y(1) + x(3)$$

etc.

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Difference Equations:

Linear Constant-Coefficient difference equation (LCCDE):

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

Total Solution: $y(n) = y_h(n) + y_p(n)$

The procedure for computing the solution of LCCDE is:

$y_h(n)$ is the **homogenous** solution

$y_p(n)$ is the **particular** solution

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Difference Equations:

The Homogenous Solution:

Now set input

$$x(n) = 0$$

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

Assume exponential solution: $y(n) = \lambda^n$ where λ is constant to be determined.

Make substitutions into D.E.: $\sum_{k=0}^N a_k \cdot \lambda^{n-k} = 0$

or

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

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

The Homogenous Solution:

❖ From the previous:

$$\lambda^{n-N} \cdot \left(\lambda^N + a_1 \cdot \lambda^{N-1} + a_2 \cdot \lambda^{N-2} + \dots + a_{N-1} \cdot \lambda + a_N \right) = 0$$

❖ The polynomial in parenthesis is called characteristic polynomial

Has N roots:

❖ In practice, typically coefficients (a_1, a_2, \dots, a_N) are real.

The general form of the homogenous solution is:

$$y_h(n) = C_1 \cdot \lambda_1^n + C_2 \cdot \lambda_2^n + C_3 \cdot \lambda_3^n + \dots + C_N \cdot \lambda_N^n \quad \lambda_1, \lambda_2, \dots, \lambda_N$$

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Recursive Difference Equations Stability:

Assuming N-distinct roots, the homogeneous solution is written as:

$$y_h(n) = C_1 \cdot \lambda_1^n + C_2 \cdot \lambda_2^n + C_3 \cdot \lambda_3^n + \dots C_N \cdot \lambda_N^n$$

$$\text{If } \left. \begin{array}{l} y_h(n) \rightarrow 0 \\ n \rightarrow \infty \end{array} \right\} \begin{array}{l} \text{system is} \\ \text{stable} \end{array}$$

For $y_h(n) \rightarrow 0$ as $n \rightarrow \infty$ then all the roots must satisfy the following relation:

$$|\lambda_i| < 1, \text{ for all } i = 1, 2, \dots, N$$

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Recursive Difference Equations Stability Example:

- Consider: $y(n) = \frac{5}{6} \cdot y(n-1) - \frac{1}{6} \cdot y(n-2) + x(n)$
- Assume an exponential solution: $y(n) = \lambda^n$
 - Where λ is a constant to be determined
- Plug into difference equation: $\frac{1}{6} \cdot \lambda^{n-2} - \frac{5}{6} \cdot \lambda^{n-1} + \lambda^n = 0$
 -
- Note above must = 0 (input = 0)
- Factor:

$$\lambda^{n-2} \cdot \left[\lambda^2 - \frac{5}{6} \cdot \lambda^1 + \frac{1}{6} \right] = 0$$

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Recursive Difference Equations Stability Example:

■ Characteristic Equation: $\lambda^2 - \frac{5}{6} \cdot \lambda + \frac{1}{6} = 0$

■ Eliminate Fractions:

$$6 \cdot \lambda^2 - 5 \cdot \lambda + 1 = 0$$

■ Quadratic Equation:

□ Solution:

$$a \cdot \lambda^2 + b \cdot \lambda + c = 0$$

■ Our Example: $\lambda_{1,2} = \frac{-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$

$$\lambda_{1,2} = \frac{5 \pm \sqrt{25 - 24}}{12} = \frac{5 \pm 1}{12} = \frac{1}{2}, \frac{1}{3}$$

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Recursive Difference Equations Stability Example:

- Two Solutions: $y_1(n) = \left(\frac{1}{2}\right)^n, y_2(n) = \left(\frac{1}{3}\right)^n$
- The Solution is Linear Combination of two solutions:

- $y(n) = C_1 \cdot y_1(n) + C_2 \cdot y_2(n)$
-

$$y(n) = C_1 \cdot \left(\frac{1}{2}\right)^n + C_2 \cdot \left(\frac{1}{3}\right)^n$$

- Note C1 and C2 are arbitrary constants computed from initial conditions

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Recursive Difference Equations Stability Example:

- $y_h(n) =$ homogeneous solution

- If
$$\left. \begin{array}{l} y_h(n) \rightarrow \infty \\ n \rightarrow \infty \end{array} \right\} \begin{array}{l} \text{system is} \\ \text{unstable} \end{array}$$

- If
$$\left. \begin{array}{l} y_h(n) \rightarrow 0 \\ n \rightarrow \infty \end{array} \right\} \begin{array}{l} \text{system is} \\ \text{stable} \end{array}$$

- For \rightarrow
$$y(n) = C_1 \cdot \left(\frac{1}{2}\right)^n + C_2 \cdot \left(\frac{1}{3}\right)^n$$

- System is stable

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

The Particular Solution:

The particular solution $y_p(n)$ is required to satisfy the DE for the **specific input signal** $x(n)$, $n > 0$:

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

For example if $x(n) = a^n u(n)$ the particular solution will be in the form:

$$y_p(n) = C a^n u(n)$$

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

The Particular Solution:

The particular solution to a DE for different several inputs:

Term in $x(n)$	Particular Solution
C	C_1
Cn	$C_1n + C_2$
Ca^n	C_1a^n
$C \cos(n\omega_0)$	$C_1 \cos(n\omega_0) + C_2 \sin(n\omega_0)$
$C \sin(n\omega_0)$	$C_1 \cos(n\omega_0) + C_2 \sin(n\omega_0)$
$Ca^n \cos(n\omega_0)$	$C_1a^n \cos(n\omega_0) + C_2a^n \sin(n\omega_0)$
$C\delta(n)$	None

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Example:

Determine the total solution for $n \geq 0$ of a DT system characterized by the following difference equation:

$$y(n) - 0.25y(n - 2) = x(n)$$

For $x(n) = u(n)$ assuming the initial conditions of $y(-1) = 1$ and $y(-2) = 0$.

Solution:

Particular solution:

For $x(n) = u(n)$

Substitute this
solution into the DE

$$C_1 - 0.25C_1 = 1$$

$$y_p(n) = C_1$$

$$C_1 = \frac{1}{1 - 0.25} = \frac{4}{3}$$

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Example: (cont.)

Homogenous solution:

set $y(n) = \lambda^n$ substitute $\lambda^n - 0.25\lambda^{(n-2)} = 0$

$$\lambda^{(n-2)}(\lambda^2 - 0.25) = 0$$

$$(\lambda - 0.5)(\lambda + 0.5) = 0 \quad \text{then}$$

$$\lambda_1 = 0.5$$

$$\lambda_2 = -0.5$$

$$y_1(n) = (0.5)^n$$

$$y_2(n) = (-0.5)^n$$

$$y_h(n) = A_1 y_1(n) + A_2 y_2(n)$$

The homogenous solution

$$y_h(n) = A_1 (0.5)^n + A_2 (-0.5)^n$$

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Example: (cont.)

Total Solution:

$$y(0) - 0.25y(-2) = x(0) = 1$$

$$y(1) - 0.25y(-1) = x(1) = 1$$

The total solution is:
$$y(n) = \frac{4}{3} + A_1(0.5)^n + A_2(-0.5)^n \quad n \geq 0$$

at $n = 0$ and $n = 1$

$$y(0) = \frac{4}{3} + A_1 + A_2 = 1$$

$$y(1) = \frac{4}{3} + \frac{1}{2}A_1 - \frac{1}{2}A_2 = 1 \quad A_1 = -\frac{1}{2} \quad A_2 = \frac{1}{6}$$

The solution is:
$$y(n) = \frac{4}{3} - (0.5)^{n+1} + \frac{1}{6}(-0.5)^n \quad n \geq 0$$

Digital Signal Processing

Discrete-Time Signals & Systems

3.3. Analysis of DT Linear Invariant Systems

Difference Equations:

Zero-Input & Zero-State Response:

An alternate approach to determining the total solution of DE.

$$y(n) = y_{zi}(n) + y_{zs}(n)$$

$y_{zi}(n)$: zero-input response
 $y_{zs}(n)$: zero-state response

➤ $y_{zi}(n)$ is obtained by solving DE by setting the **input $x(n) = 0$** .

➤ $y_{zs}(n)$ is obtained by solving DE by applying the specified input with **all initial conditions set to zero (0)**.

Digital Signal Processing

Discrete-Time Signals & Systems

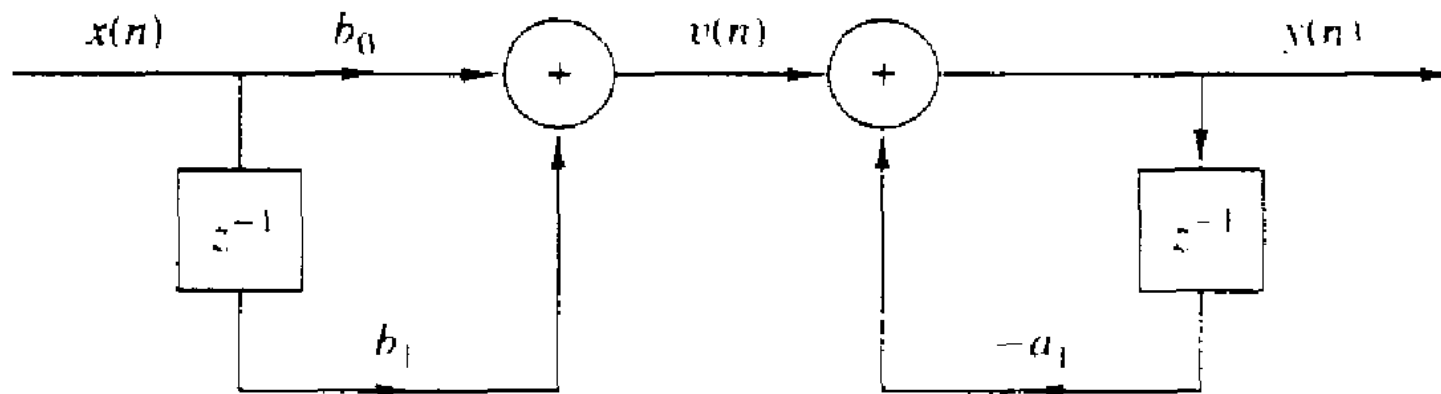
3.4. implementation of DT Systems

Structure for the Realization of LTI Systems:

Here, the **LCCDE structure for the realization** of systems is described, additional structures for these system will introduce in later chapters.

Consider the first-order system:

$$y(n] = -a_1 y(n - 1) + b_0 x(n) + b_1 x(n - 1)$$



Direct Form I Structure

Digital Signal Processing

Discrete-Time Signals & Systems

3.4. implementation of DT Systems

.....•

Structure for the Realization of LTI Systems:

The previous system can be viewed as 2 LTI systems in cascade, the first is a **non-recursive** system described by the equation:

$$v(n) = b_0x(n) + b_1x(n - 1)$$

Whereas the second is a **recursive** system described by the equation:

$$y(n) = -a_1y(n - 1) + v(n)$$

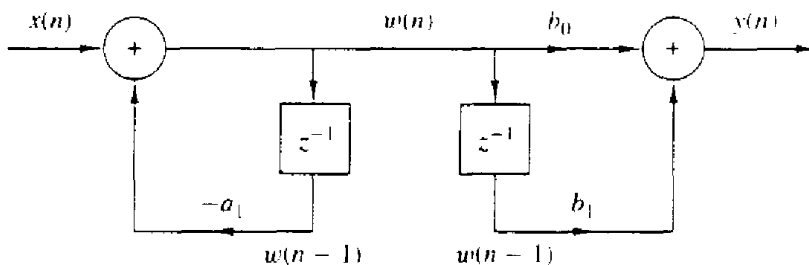
Digital Signal Processing

Discrete-Time Signals & Systems

3.4. implementation of DT Systems

Structure for the Realization of LTI Systems:

If we **interchange** the order of the recursive and non-recursive systems, we obtain an **alternative structure** for the realization of the system described previously:



$$y(n) = b_0 w(n) + b_1 w(n-1)$$

$$w(n) = -a_1 w(n-1) + x(n)$$

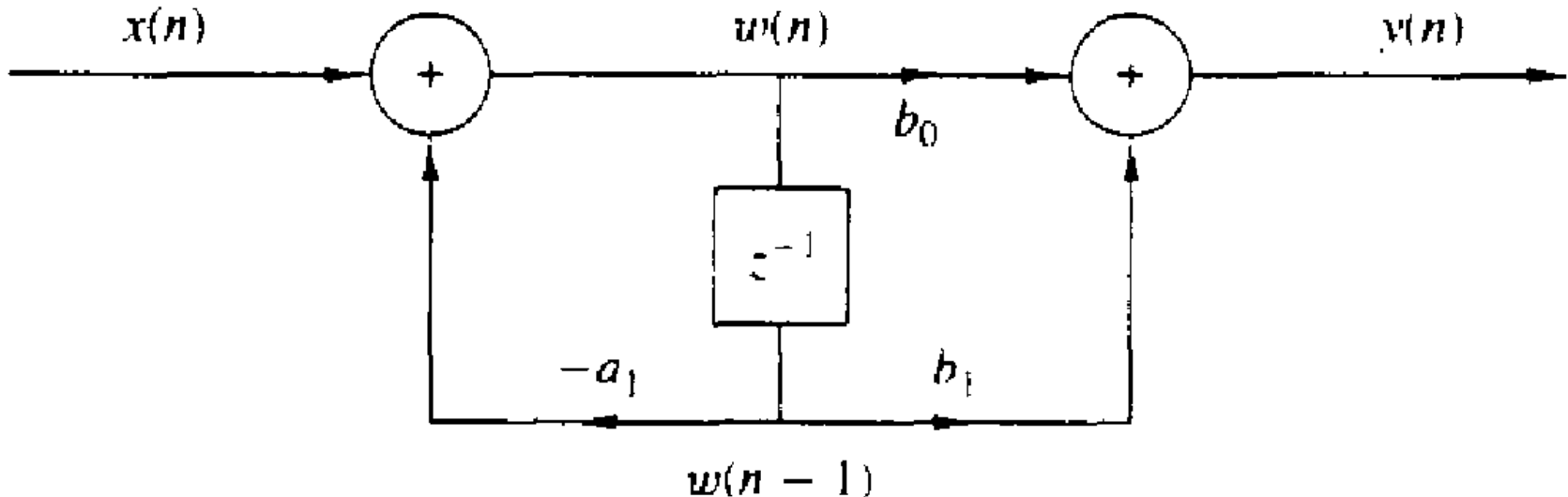
Digital Signal Processing

Discrete-Time Signals & Systems

3.4. implementation of DT Systems

Structure for the Realization of LTI Systems:

Minimizing the **2 delay** in the structure **form I** to **1 delay** in structure **form II**.



Direct Form II Structure

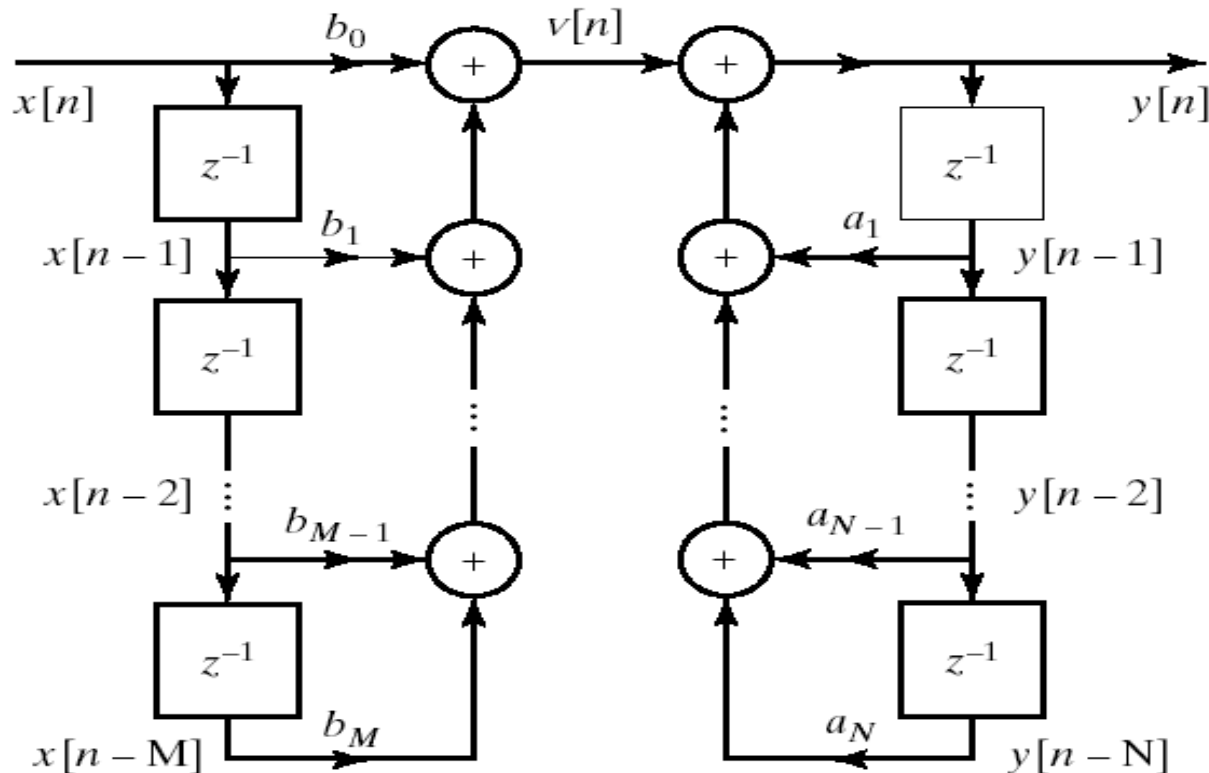
Digital Signal Processing

Discrete-Time Signals & Systems

3.4. implementation of DT Systems

Structure for the Realization of LTI Systems:

In General $\sum_{k=0}^N \hat{a}_k y[n-k] = \sum_{k=0}^M \hat{b}_k x[n-k]$ or $y[n] - \sum_{k=1}^N a_k y[n-k] = \sum_{k=0}^M b_k x[n-k]$



Direct Form I Structure

Digital Signal Processing

Discrete-Time Signals & Systems

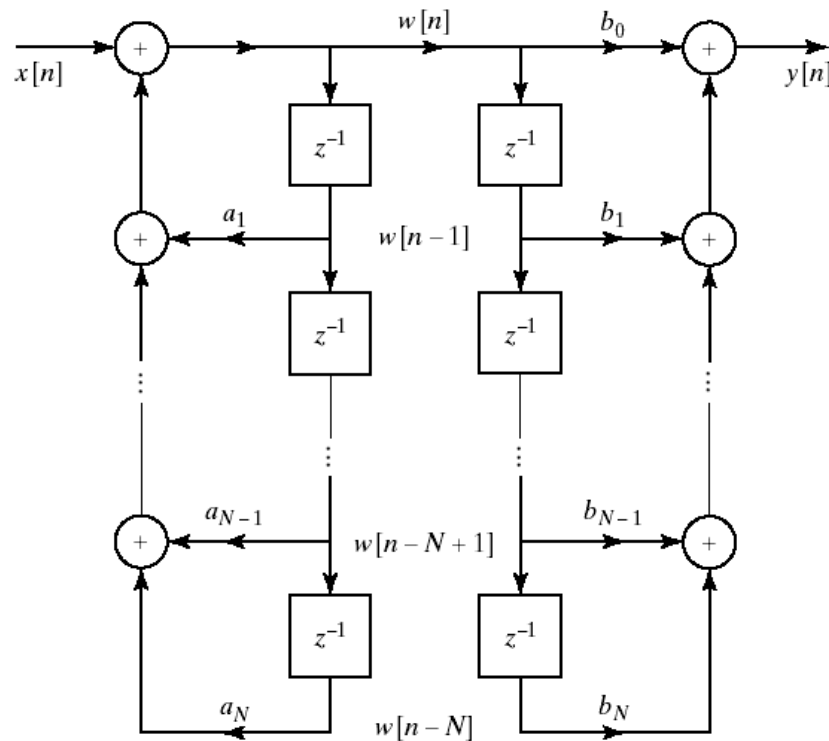
3.4. implementation of DT Systems

Structure for the Realization of LTI Systems:

We can change the order of the cascade systems.

$$w[n] = \sum_{k=1}^N a_k w[n-k] + x[n]$$

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

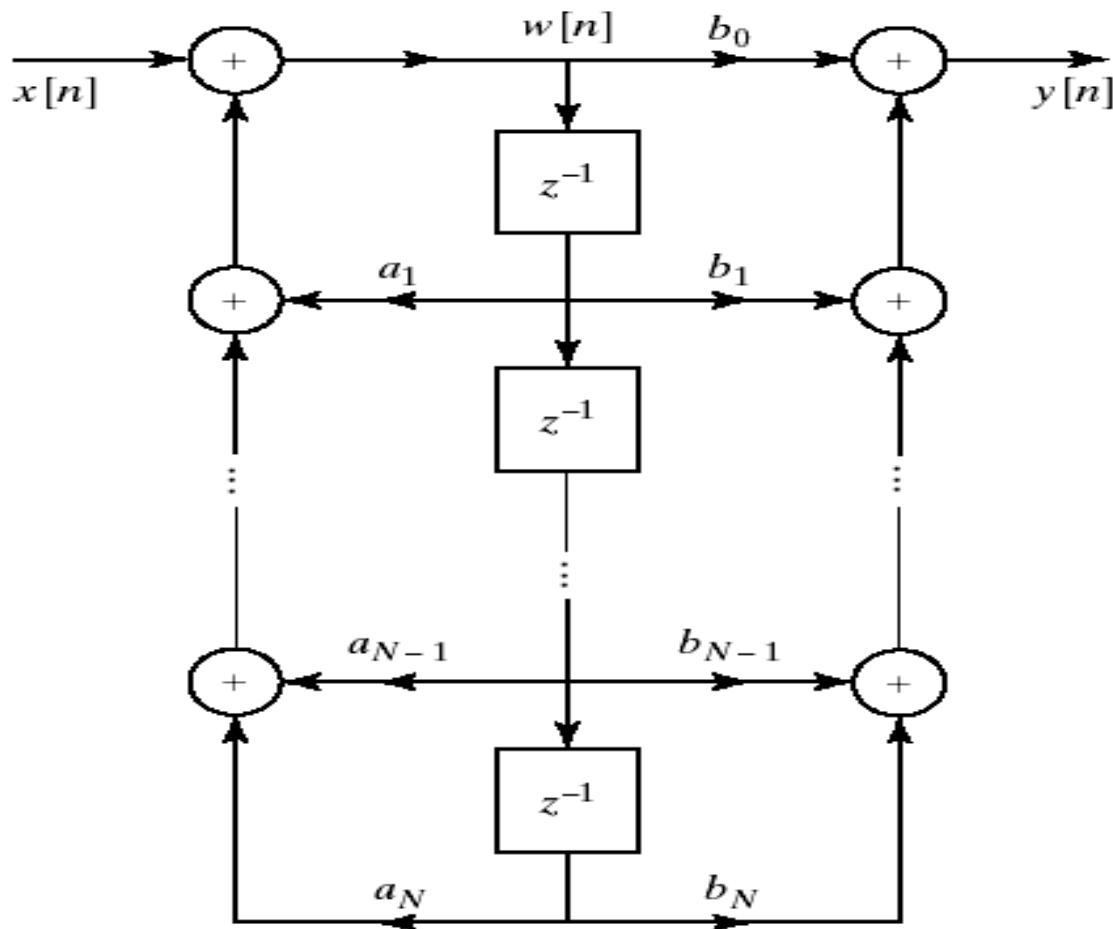


Digital Signal Processing

Discrete-Time Signals & Systems

3.4. implementation of DT Systems

Structure for the Realization of LTI Systems:



Direct Form II Structure



End!!!