# LECTURE NOTES

# ON

# DIGITAL SIGNAL PROCESSING

**III B.TECH II SEMESTER (JNTUK – R 13)**

**FACULTY : B.V.S.RENUKA DEVI (Asst.Prof) / Dr. K. SRINIVASA RAO (Assoc. Prof)**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATIONS ENGINEERING**

**GVP COLLEGE OF ENGINEERING FOR WOMEN**

**MADHURAWADA, VISAKHAPATNAM-48**

## UNIT III – REALIZATION OF DIGITAL FILTERS

**Syllabus**

Review of Z-transforms, Applications of Z – transforms, solution of difference equations – digital filters, Block diagram representation of linear constant-coefficient difference equations, Basic structures of IIR systems, Transposed forms, Basic structures of FIR systems, System function.

**Review of Z – Transforms**

Z – Transform expresses a discrete – time sequence as a linear combination of discrete time complex exponentials.
The Z – Transform of a general DT signal x[n] is defined as

$$X(z) \triangleq \sum_{n=-\infty}^{\infty} x[n]z^{-n}$$

$$, z \text{ is a complex variable} = re^{j\omega}$$

r =magnitude of z
ω=angle of z

The range of values of z for which X(z) converges is called the Region Of Convergence (ROC) of the Z – Transform.
Specification of z-Transform requires both the algebraic expression for the ZT and the ROC.

**Properties of Z – Transform**

| Property | Time Domain | z-Domain | ROC |
|---|---|---|---|
| Notation: | $x(n)$ | $X(z)$ | ROC: $r_2 < \|z\| < r_1$ |
| | $x_1(n)$ | $X_1(z)$ | $ROC_1$ |
| | $x_2(n)$ | $X_1(z)$ | $ROC_2$ |
| Linearity: | $a_1x_1(n) + a_2x_2(n)$ | $a_1X_1(z) + a_2X_2(z)$ | At least $ROC_1 \cap ROC_2$ |
| Time shifting: | $x(n-k)$ | $z^{-k}X(z)$ | ROC, except $z = 0$ (if $k > 0$) and $z = \infty$ (if $k < 0$) |
| z-Scaling: | $a^n x(n)$ | $X(a^{-1}z)$ | $\|a\|r_2 < \|z\| < \|a\|r_1$ |
| Time reversal | $x(-n)$ | $X(z^{-1})$ | $\frac{1}{r_1} < \|z\| < \frac{1}{r_2}$ |
| Conjugation: | $x^*(n)$ | $X^*(z^*)$ | ROC |
| z-Differentiation: | $n\,x(n)$ | $-z\frac{dX(z)}{dz}$ | $r_2 < \|z\| < r_1$ |
| Convolution: | $x_1(n) * x_2(n)$ | $X_1(z)X_2(z)$ | At least $ROC_1 \cap ROC_2$ |

**solution of difference equations – digital filters**

the convolution sum description of an LTI Discrete – Time System can be used to implement the LTI system.

The output of a Linear Time – Invariant system is given by the convolution sum as

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

Taking Z – Transform on both sides and using the convolution property of the Z – Transform, we have

$$Y(z) = X(z)H(z)$$

$$or, H(z) = \frac{Y(z)}{X(z)}$$

H(z) is called the **System Function or Transfer Function.**

An LTI system can also be represented by a Linear Constant Coefficient Difference Equation (LCCDE) as

$$\sum_{k=0}^{N-1} a_k y[n-k] = \sum_{k=0}^{M-1} b_k x[n-k] \;, a_0 = 1$$

Taking Z – Transform on both sides, and applying time – shifting property of Z – Transform, we have

$$Y(z) \sum_{k=0}^{N-1} a_k z^{-k} = X(z) \sum_{k=0}^{M-1} b_k z^{-k}$$

$$or, H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M-1} b_k z^{-k}}{1 + \sum_{k=1}^{N-1} a_k z^{-k}}$$

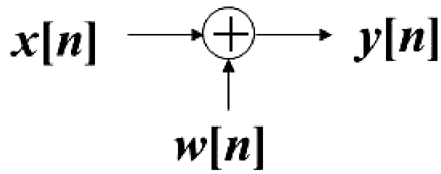
The above equation denotes the system function of an IIR system. It has both poles and zeros.

If $a_k = 0 \; \forall k,$ then, H(z) represents an FIR filter. It has only zeros and no poles.

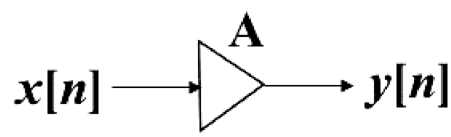
**Block diagram representation of linear constant-coefficient difference equations**

A structural representation using interconnected basic building blocks is the first step in hardware or software implementation of an LTI system.

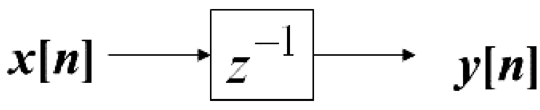
**Basic Building Blocks :**



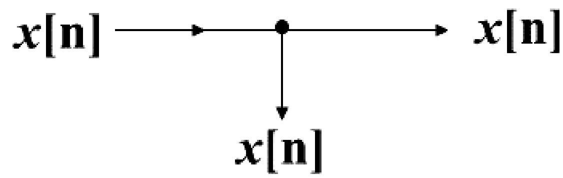

**Adder**




**Multiplier**




**Unit delay**




**Pick-off node**

A realization is **canonic** if the realization uses minimum number of delay units. Two realizations are equivalent if they have the same transfer function.

**Transpose operation** generates an equivalent structure from a given realization by the following steps:

- Interchange input and output nodes
- Reverse all the paths
- Replace pick – off nodes with adders and vice – versa.

## <u>FIR Filter Structures</u>

An FIR filter has a system function given by

$$H(z) = \sum_{k=0}^{M} b_k z^{-k} = b_0 + b_1 z^{-1} + \cdots\cdots + b_M z^{-M}$$

From this equation, the impulse response h[n] can be written as

$$h[n] = \begin{cases} b_n, 0 \le n \le M \\ 0, \quad otherwise \end{cases}$$

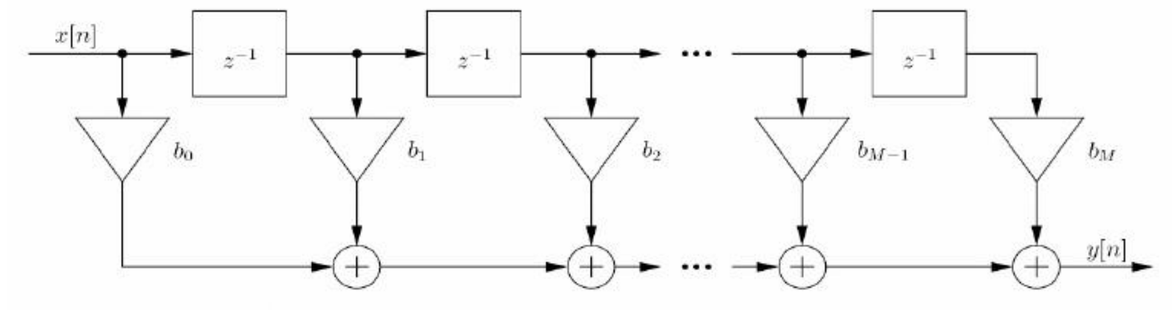
the difference equation representation is given by

$$y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + \cdots \cdots + b_M x[n-M]$$

The **length** of the filter is M+1, and the **order** of the filter is M.

1. **<u>Direct – Form/Transversal/Tapped Delay Line Structure</u>**

From the above equation, the direct form structure is as follows

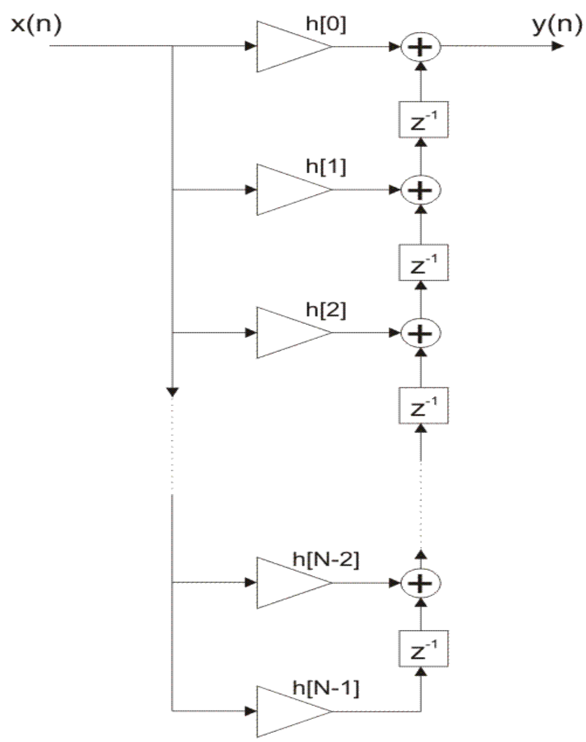



Here, the filter has an order of M, it requires M delays/memory locations, M+1 Multipliers, and M adders.

Structures in which the multiplier coefficients are directly available as coefficients of H(z) are called **Direct – Form Structures.**

2. **<u>Transposed version of direct form</u>**

By applying the steps to obtain the transposed form to the above direct – form structure, we obtain the following transposed structure.



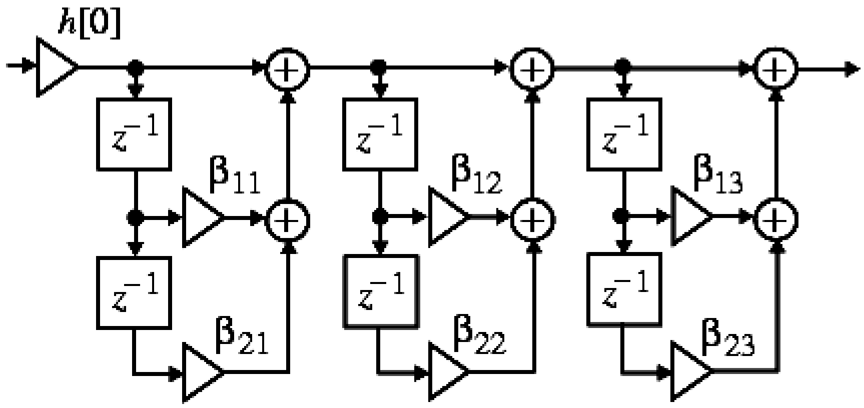

3. **<u>Cascade form structure</u>**

To obtain the cascade structure, H(z) is factorized in terms of second – order factors and first – order factors.

$$H(z) = b_0 + b_1 z^{-1} + \cdots\cdots + b_{M-1} z^{-(M-1)}$$

$$H(z) = \begin{cases} b_0 \left\{ \prod_{k=1}^{(M-1)/2} (1 + B_{1k} z^{-1} + B_{2k} z^{-2}) \right\} \; for\ M\ odd \\ b_0 \left\{ (1 + b_{10} z^{-1}) \prod_{k=1}^{(M-2)/2} (1 + B_{1k} z^{-1} + B_{2k} z^{-2}) \right\} \; for\ M\ even \end{cases}$$

For example, for M=7 (order =6), the cascade structure would be




Where h[0]=$b_0$

**Basic Structures for IIR Systems**

The convolution sum description of an LTI discrete – time system can, in principle, be used to implement the system. However, for an IIR system, this approach is not practical, since the impulse response is infinite in length. So, we use the input – output relation to obtain the realization.

The system function of an IIR filter is

$$H(z) = \frac{\sum_{k=0}^{M-1} b_k z^{-k}}{1 + \sum_{k=1}^{N-1} a_k z^{-k}} \;\; , N \geq M$$

Order = N-1

The corresponding difference equation is

$$y[n] = -\sum_{k=1}^{N-1} a_k y[n-k] + \sum_{k=0}^{M-1} b_k x[n-k]$$

**Direct Form – I Structure**

The transfer function H(z) of the IIR system is divided into two parts connected in cascade, with the first part $H_1(z)$ containing only the zeroes, and the second part $H_2(z)$ containing only the poles.

$$H(z) = H_1(z) H_2(z)$$

Where

$$H_1(z) = \sum_{k=0}^{M-1} b_k z^{-k}$$

And

$$H_2(z) = \frac{1}{1 + \sum_{k=1}^{N-1} a_k z^{-k}}$$

These equations can be rewritten as

$$H_1(z) = \frac{W(z)}{X(z)} = \sum_{k=0}^{M} b_k z^{-k}$$

or in time domain

$$w[n] = b_0 x[n] + b_1 x[n-1] + \cdots\cdots + b_M x[n-M]$$

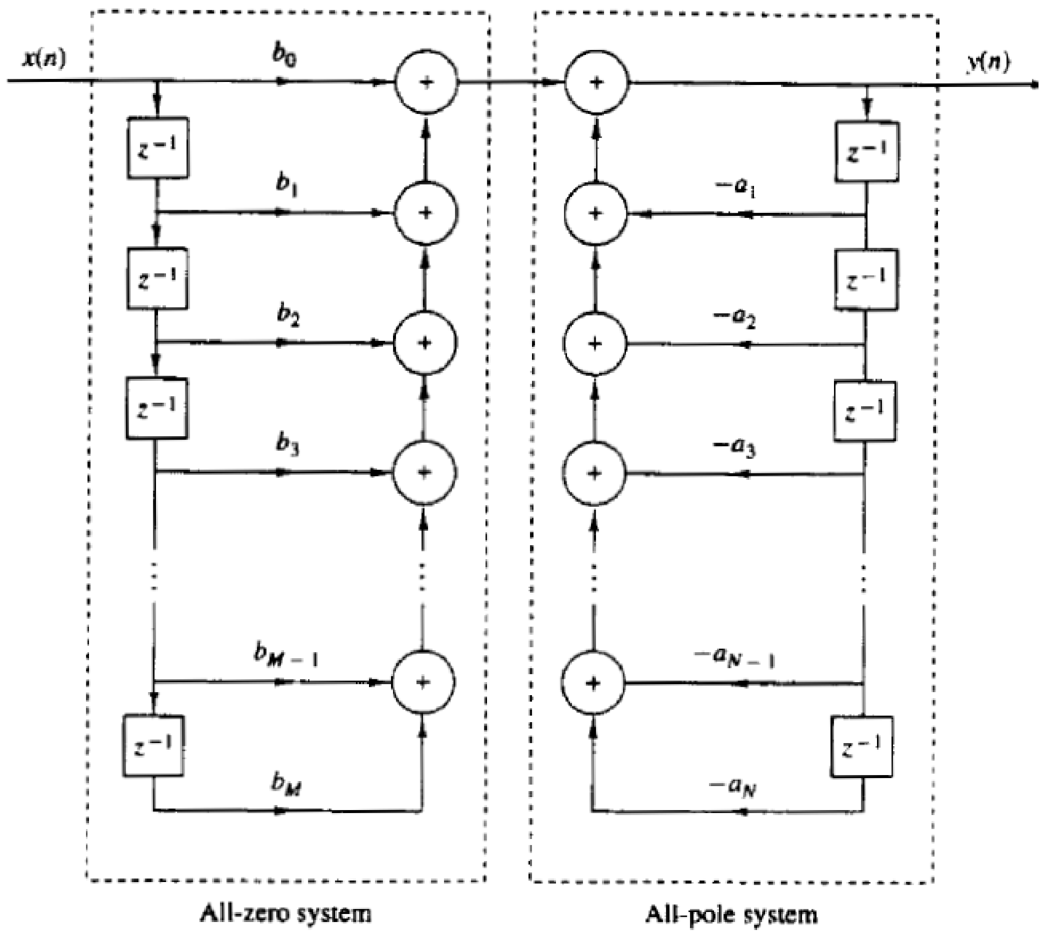
And

$$H_2(z) = \frac{Y(z)}{W(z)} = \frac{1}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

Or, in time domain,

$$y[n] = w[n] - a_1 y[n-1] - a_2 y[n-2] - \cdots\cdots - a_N y[n-N]$$

Realizing the above two equations for $H_1(z)$ and $H_2(z)$ using basic building blocks and connecting them in cascade, we obtain the **Direct Form – I structure** as follows




This realization requires M+N memory units, M+N+1 multipliers and M+N adders.

**Direct Form – II Structure**

Since, in a cascade arrangement, the order of the systems is not important, the all – pole system $H_2(z)$ and the all - zero system $H_1(z)$ can be interchanged .i.e.,

$$H_1(z) = \frac{V(z)}{X(z)} = \frac{1}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

Or, in time domain,

$$v[n] = x[n] - a_1 v[n-1] - a_2 v[n-2] - \cdots\cdots - a_N v[n-N]$$

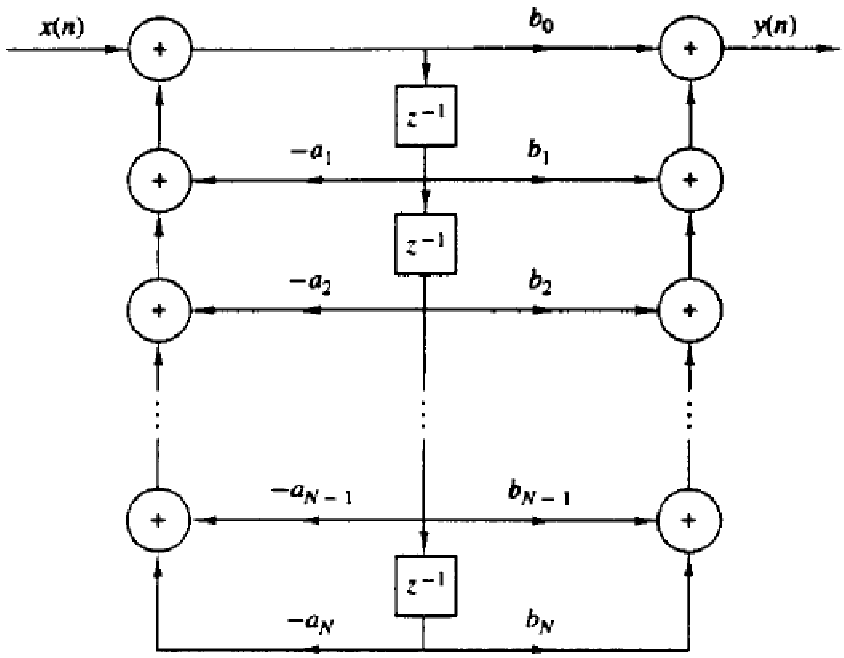
And

$$H_1(z) = \frac{Y(z)}{V(z)} = \sum_{k=0}^{M} b_k z^{-k}$$

or in time domain

$$y[n] = b_0 v[n] + b_1 v[n-1] + \cdots\cdots + b_M v[n-M]$$

Both the time domain equations involve the delayed versions of v[n], and hence require a single set of delay elements. This results in the **Direct – Form II structure** as follows




This structure requires M+N+1 multipliers, M+N additions and maximum{M, N} delays. Since this structure minimizes the number of delays, it is called **canonic.**

**<u>Cascade – Form Structures</u>**

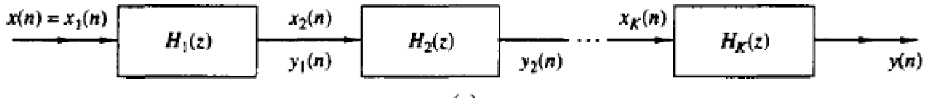
The system function of an IIR filter is given by

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} \quad , N \geq M$$

The system is factored into a cascade of second – order subsystems, such that H(z) can be expressed as

$$H(z) = \prod_{i=1}^{K} H_i(z) \, , K = integral\ part\ of\ \frac{N+1}{2}$$

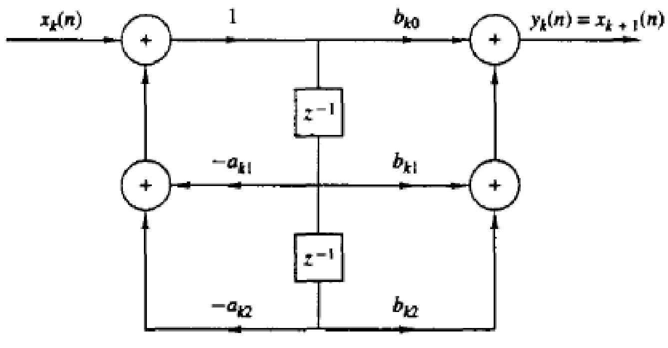
If N is odd, one of the subsystems is of first order.




In the above equation, each of the subsystems $H_i(z)$ has the general form

$$H_i(z) = \frac{b_{i0} + b_{i1} z^{-1} + b_{i2} z^{-2}}{1 + a_{i1} z^{-1} + a_{i2} z^{-2}}$$

Each of the second – order subsystem can be realized either in Direct Form – I, or Direct – Form II or transposed form.




Since there are many ways of pairing poles and zeroes, a variety of cascade realizations are possible.

**Parallel – Form Structures**

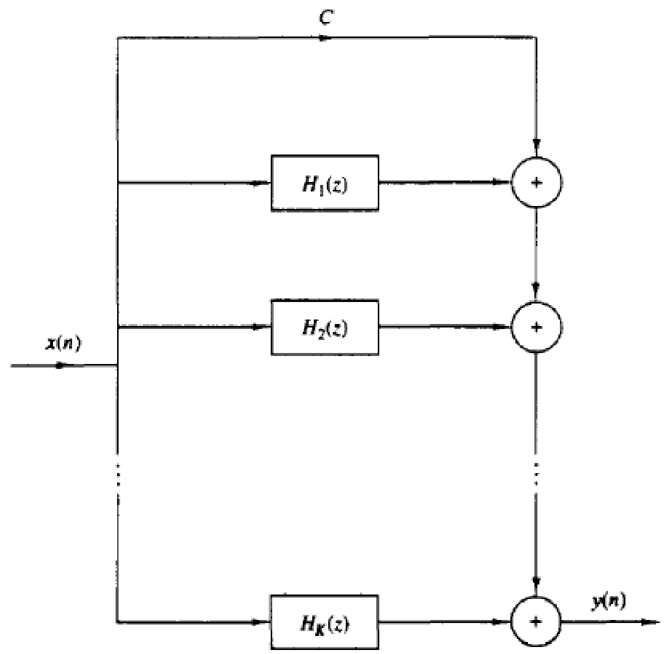
A parallel form realization of an IIR system can be obtained by performing partial – fraction expansion of H(z) .i.e.,

$$H(z) = C + \sum_{k=1}^{N} \frac{A_k}{1 - p_k z^{-1}}$$

If some poles are complex valued, pairs of complex conjugate poles are combined to get second order subsystems.

$$H(z) = C + \sum_{k=1}^{N} H_k(z), \text{ where}$$

$$H_k(z) = \frac{b_{k0} + b_{k1} z^{-1}}{1 + a_{k1} z^{-1} + a_{k2} z^{-2}}$$




Each of the subsystems can be realized either in Direct Form – I, or Direct – Form II or transposed form.