Lecture - 19

Digital Filters-IIR Filter Structures

DIGITAL FILTERS

Filters are a basic component of all signal processing and telecommunication systems. The primary functions of a filter are one or more of the followings: (a) to confine a signal into a prescribed frequency band or channel for example as in anti-aliasing filter or a radio/tv channel selector, (b) to decompose a signal into two or more sub-band signals for sub-band signal processing, for example in music coding, (c) to modify the frequency spectrum of a signal, for example in audio graphic equalizers, and (d) to model the input-output relation of a system such as a mobile communication channel, voice production, musical instruments, telephone line echo, and room acoustics.

In this chapter we introduce the general form of the equation for a linear time-invariant filter and consider the various methods of description of a filter in time and frequency domains. We study different filter forms and structures and the design of low-pass filters, band-pass filters, band-stop filters and filter banks. We consider several applications of filters such as in audio graphic equalizers, noise reduction filters in Dolby systems, image deblurring, and image edge emphasis.

4.1 Introduction

Filters are widely employed in signal processing and communication systems in applications such as channel equalization, noise reduction, radar, audio processing, video processing, biomedical signal processing, and analysis of economic and financial data. For example in a radio receiver band-pass filters, or tuners, are used to extract the signals from a radio channel. In an audio graphic equalizer the input signal is filtered into a number of sub-band signals and the gain for each sub-band can be varied manually with a set of controls to change the perceived audio sensation. In a Dolby system pre-filtering and post-filtering are used to minimize the effect of noise. In hi-fi audio a compensating filter may be included in the preamplifier to compensate for the non-ideal frequency-response characteristics of the speakers. Filters are also used to create perceptual audio-visual effects for music, films and in broadcast studios.

The primary functions of filters are one of the followings:

(a) To confine a signal into a prescribed frequency band as in low-pass, high-pass, and band-pass filters.

(b) To decompose a signal into two or more sub-bands as in filter-banks, graphic equalizers, sub-band coders, frequency multiplexers.

(c) To modify the frequency spectrum of a signal as in telephone channel equalization and audio graphic equalizers.

(d) To model the input-output relationship of a system such as telecommunication channels, human vocal tract, and music synthesizers.

Depending on the form of the filter equation and the structure of implementation, filters may be broadly classified into the following classes:

(a) Linear filters versus nonlinear filters.

(b) Time-invariant filters versus time-varying filters.

(c) Adaptive filters versus non-adaptive filters.

(d) Recursive versus non-recursive filters.

(e) Direct-form, cascade-form, parallel-form and lattice structures.

In this chapter we are mainly concerned with linear time-invariant (LTI) filters. These are a class of filters whose output is a linear combination of the input and whose coefficients do not vary with time. Time-varying and adaptive filters are considered in later chapters.

4.1.1 Alternative Methods for Description of Filters

Filters can be described using the following time or frequency domain methods:

(a) *Time domain input-output relationship*. As described in section 4.2 a difference equation is used to describe the output of a discrete-time filter in terms of a weighted combination of the input and previous output samples. For example a first-order filter may have the following difference equation

$$y(m) = a\, y(m-1) + x(m) \tag{4.1}$$

where $x(m)$ is the filter input, $y(m)$ is the filter output and a is the filter coefficient.

(b) Impulse Response. A filter can be described in terms of its response to an impulse input. For example the response of the filter of Eq. (4.1) to a discrete-time impulse input at m=0 is

$$y(m) = a^m \qquad m=0,\ 1,\ 2,\ \ldots \tag{4.2}$$

$y(m) = a^m = 1, a, a^2, a^3,\ a^4, \ldots$ for m=0,1,2,3, 4 ... and it is assumed y(-1)=0.

Impulse response is useful because: (i) any signal can be viewed as the sum of a number of shifted and scaled impulses, hence the response a linear filter to a signal is the sum of the responses to all the impulses that constitute the signal, (ii) an impulse input contains all frequencies with equal energy, and hence it excites a filter at all frequencies and (iii) impulse response and frequency response are Fourier transform pairs.

(c) *Transfer Function, Poles and Zeros*. The transfer function of a digital filter $H(z)$ is the ratio of the z-transforms of the filter output and input given by

$$H(z) = \frac{Y(z)}{X(z)} \tag{4.3}$$

For example the transfer function of the filter of Eq. (4.1) is given by

$$H(z) = \frac{1}{1 - a\, z^{-1}} \tag{4.4}$$

A useful method of gaining insight into the behavior of a filter is the pole-zero description of a filter. As described in Sec. X poles and zeros are the roots of the denominator and numerator of the transfer function respectively.

(d) *Frequency Response.* The frequency response of a filter describes how the filter alters the magnitude and phase of the input signal frequencies. The frequency response of a filter can be obtained by taking the Fourier transform of the impulse response of the filter, or by simple substitution of the frequency variable $e^{j\omega}$ for the z variable $z = e^{j\omega}$ in the z-transfer function as

$$H(z = e^{j\omega}) = \frac{Y(e^{j\omega})}{X(e^{j\omega})} \tag{4.5}$$

The frequency response of a filter is a complex variable and can be described in terms of the filter magnitude response and the phase response of the filter.

4.2 Linear Time-Invariant Digital Filters

Linear time-invariant (LTI) filters are a class of filters whose output is a linear combination of the input signal samples and whose coefficients do not vary with time. The *linear* property entails that the filter response to a weighted sum of a number of signals, is the weighted sum of the filter responses to the individual signals. This is the *principle of superposition*. The term *time-invariant* implies that the filter coefficients and hence its frequency response is fixed and does not vary with time.

In the time domain the input-output relationship of a discrete-time linear filter is given by the following linear difference equation:

$$\boxed{y(m) = \sum_{k=1}^{N} a_k y(m-k) + \sum_{k=0}^{M} b_k x(m-k) \qquad (4.6)}$$

where $\{a_k, b_k\}$ are the filter coefficients, and the output $y(m)$ is a linear combination of the previous N output samples $[y(m-1),\ldots, y(m-N)]$, the present input sample $x(m)$ and the previous M input samples $[x(m-1),\ldots, x(m-M)]$. The characteristic of a filter is completely determined by its coefficients $\{a_k, b_k\}$.

For a time-invariant filter the coefficients $\{a_k, b_k\}$ are constants calculated to obtain a specified frequency response.

The filter transfer function, obtained by taking the z-transform of the difference equation (4.6), is given by:

$$\boxed{H(z)=\frac{\sum_{k=0}^{M} b_k z^{-k}}{1-\sum_{k=1}^{N} a_k z^{-k}}} \tag{4.7}$$

The frequency response of this filter can be obtained from Eq. (4.7) by substituting the frequency variable $e^{j\omega}$ for the z variable, $z = e^{j\omega}$, as

$$H(e^{j\omega})=\frac{\sum_{k=0}^{M} b_k e^{-j\omega k}}{1-\sum_{k=1}^{N} a_k e^{-j\omega k}} \tag{4.8}$$

Since from Fourier transform a signal is a weighted combination of a number of sine waves, it follows, from superposition principle, that in frequency domain linear filtering can be viewed as linear combination of the frequency constituents of the input multiplied by the frequency response of the signal.

Filter Order – The order of a discrete-time filter is the highest discrete-time *delay* used in the input-output equation of the filter. For Example, in Equations (4.6 or 4.7) the filter order is the larger of the values of N or M. For continuous-time filters the filter order is the order of the highest differential term used in the input-output equation of the filter.

4.3 Recursive and non-Recursive Filters

Fig. 4.1 shows a block diagram implementation of the linear time-invariant filter Eq. (4.1). The transfer function of the filter in Eq. (4.7) is the ratio of two polynomials in the variable z and may be written in a cascade form as

$$H(z)=H_1(z)H_2(z) \tag{4.9}$$

where $H_1(z)$ is the transfer function of a feed-forward, *all-zero*, filter given by

$$H_1(z)=\sum_{k=0}^{M} b_k z^{-k} \tag{4.10}$$

and $H_2(z)$ is the transfer function of a feedback, *all-pole*, recursive filter given by

$$H_2(z)=\frac{1}{1-\sum_{k=1}^{N} a_k z^{-k}} \tag{4.11}$$

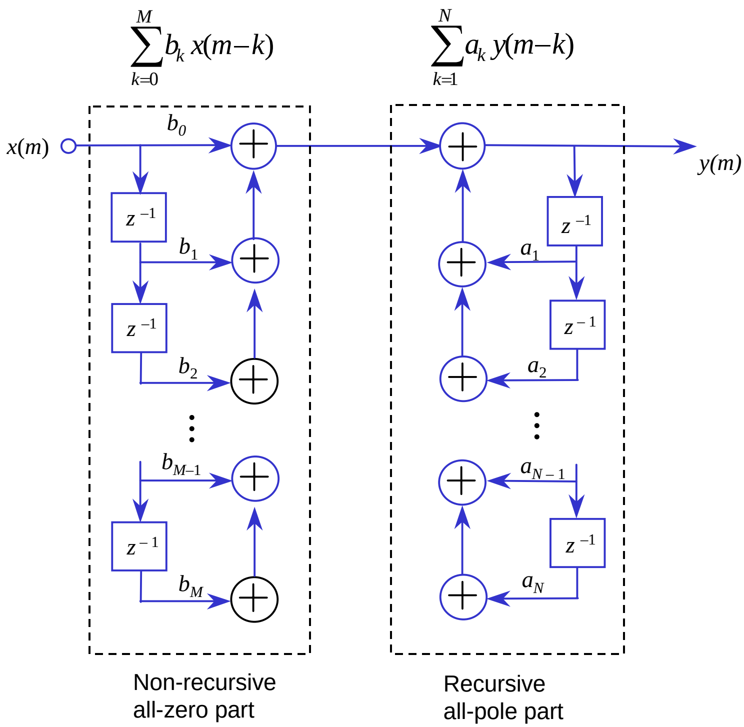


Figure 4.1 Illustration of a direct-form pole-zero IIR filter showing the output is composed of the sum of two vector products: a weighted combination of the input samples $[b_0, \ldots, b_M][x(m), \ldots, x(M\text{-}1)]^T$ plus a weighted combination of the output feedback $[a_1, \ldots, a_N][y(m\text{-}1), \ldots, y(m\text{-}N)]^T$. T denotes transpose.

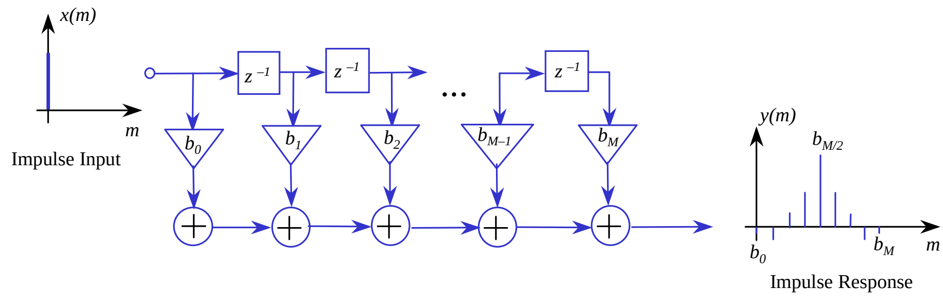


Figure 4.2 Direct-form Finite Impulse Response (FIR) filter.

4.3.1 Non-Recursive or Finite Impulse Response (FIR) Filters

A non-recursive filter has no feedback and its input-output relation is given by

$$y(m)=\sum_{k=0}^{M} b_k\, x(m-k) \tag{4.12}$$

As shown in Fig 4.2 the output $y(m)$ of a non-recursive filter is a function only of the input signal $x(m)$. The response of such a filter to an impulse consists of a finite sequence of M+1 samples, where M is the filter order. Hence, the filter is known as a *Finite-Duration Impulse Response* (FIR) filter. Other names for a non-recursive filter include all-zero filter, feed-forward filter or moving average (MA) filter a term usually used in statistical signal processing literature.

4.3.2 Recursive or Infinite Impulse Response (IIR) Filters

A recursive filter has feedback from output to input, and in general its output is a function of the previous output samples and the present and past input samples as described by the following equation

$$y(m)=\sum_{k=1}^{N} a_k\, y(m-k)+\sum_{k=0}^{M} b_k\, x(m-k) \tag{4.13}$$

Fig 4.1 shows a direct form implementation of Eq. (4.13). In theory, when a recursive filter is excited by an impulse, the output persists forever. Thus a recursive filter is also known as an *Infinite Duration Impulse Response* (IIR) filter. Other names for an IIR filter include feedback filters, pole-zero filters and auto-regressive-moving-average (ARMA) filter a term usually used in statistical signal processing literature.

A discrete-time IIR filter has a z-domain transfer function that is the ratio of two z-transform polynomials as expressed in Eq. (4.7); it has a number of poles corresponding to the roots of the denominator polynomial and it may also have a number of zeros corresponding to the roots of the numerator polynomial.

The main difference between IIR filters and FIR filters is that an IIR filter is more compact in that it can usually achieve a prescribed frequency response with a smaller number of coefficients than an FIR filter. A smaller number of filter coefficients imply less storage requirements and faster calculation and a higher throughput. Therefore, generally IIR filters are more efficient in memory and computational requirements than FIR filters. However, it must be noted that an FIR filter is always stable, whereas an IIR filter can become unstable (for example if the poles of the IIR filter are outside the unit circle) and care must be taken in design of IIR filters to ensure stability.

Fig. 4.3 shows a particular case of an IIR filter when the output is a function of N previous output samples and the present input sample given by

$$y(m)=\sum_{k=1}^{N} a_k y(m-k)+g\,x(m) \tag{4.14}$$

The transfer function of this filter is given by

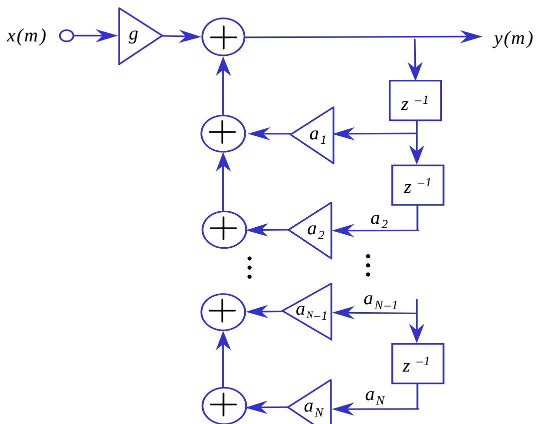


Figure 4.3 Direct-form all-pole IIR filter.

1.1 Introduction :

We know that IIR stands for infinite impulse response. Generally IIR systems are recursive type. A recursive system means; feedback connection is present from output side to the input side. For the realization of IIR systems present, past, future samples of input and past values of output are required. For the realization of IIR systems, following structures are used.

1. Direct form structure
2. Cascade form structure
3. Parallel form structure

1.2 Direct Form Structure for IIR Systems :

Direct form structure is again divided into two types :

A. Direct form-I structure

B. Direct form-II structure.

The general difference equation for discrete time LTI system is given as,

$$H(Z) = \frac{\sum_{k=0}^{M} b_k Z^{-k}}{1 + \sum_{k=1}^{N} a_k Z^{-k}} \qquad ...(1)$$

Now let $H_1(Z)$ = Numerator term of Equation (1)

$$\therefore \quad H_1(Z) = \sum_{k=0}^{M} b_k Z^{-k} \qquad ...(2)$$

And let, $$H_2(Z) = \frac{1}{1 + \sum_{k=1}^{N} a_k Z^{-k}} \qquad ...(3)$$

Putting Equations (2) and (3) in Equation (1) we get,

$$H(Z) = H_1(Z) \cdot H_2(Z) \qquad ...(4)$$

This equation shows that $H(Z)$ can be represented as the cascade connection of $H_1(Z)$ and $H_2(Z)$ as shown in Fig. I-1(a).

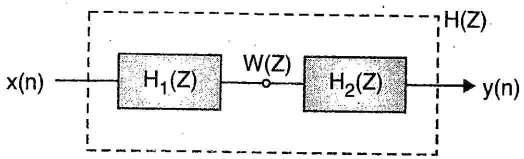


Fig. I-1(a) : Cascade connection of $H_1(Z)$ and $H_2(Z)$

We know that $H_1(Z)$ is the transfer function of the numerator term. Numerator always contain zeros of the system. So $H_1(Z)$ is called as all zero system. Now $H_2(Z)$ is transfer function of denominator. Denominator always contain poles of the system. So $H_2(Z)$ is called as all pole system.

(A) Direct form-I structure :

The direct form-I structure is obtained by cascading (connecting in series), the structure for $H_1(Z)$ and $H_2(Z)$. First we will draw the direct form structure for $H_1(Z)$.

Direct form structure for $H_1(Z)$:

Recall the equation of $H_1(Z)$ (Equation (2)). It is,

$$H_1(Z) = \sum_{k=0}^{M} b_k Z^{-k} \qquad ...(5)$$

But we know that, $$H(Z) = \frac{\text{Output } [Y(Z)]}{\text{Input } [X(Z)]}$$

To avoid the confusion we will write,

$$H_1(Z) = \frac{W(Z)}{X(Z)};$$ where $W(Z)$ is output of first stage.

$$\frac{W(Z)}{X(Z)} = \sum_{k=0}^{M} b_k Z^{-k} \qquad ...(6)$$

$$\therefore \quad \frac{W(Z)}{X(Z)} = b_0 Z^0 + b_1 Z^{-1} + b_2 Z^{-2} + \ldots.. + b_M Z^{-M}$$

But $Z^0 = 1$

$$\therefore \quad W(Z) = b_0 X(Z) + b_1 Z^{-1} X(Z) + b_2 Z^{-2} X(Z) + \ldots + b_M Z^{-M} X(Z) \qquad ...(7)$$

Taking inverse Z-transform (IZT) of Equation (7) we get,

$$y_1(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) + \ldots + b_M x(n-M) \qquad ...(8)$$

Here b_0, b_1, b_2 ... b_M are the coefficients. The direct form realization of Equation (8) is shown in Fig. I-1(b).

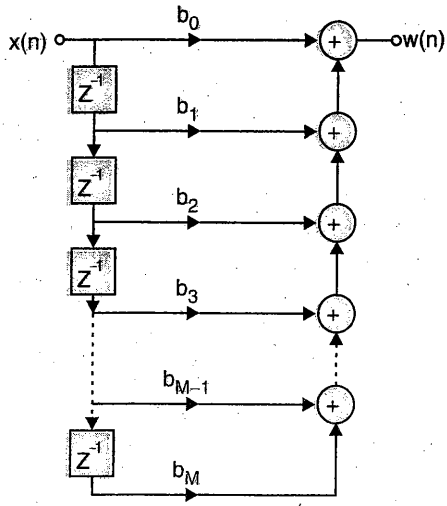


Fig. I-1(b) : Direct form realization of $H_1(Z)$ (All zero system)

Direct form structure for $H_2(Z)$:

Recall the equation for $H_2(Z)$. It is,

$$H_2(Z) = \frac{1}{1 + \sum_{k=1}^{N} a_k Z^{-k}} \qquad ...(9)$$

We have,

$$H_2(Z) = \frac{\text{Output } [Y(Z)]}{\text{Input } [X(Z)]} \qquad ...(10)$$

Now $H_2(Z)$ represents second stage of Fig. I-1(a). Input of second stage is the output of first stage. Thus input of $H_2(Z)$ is $W(Z)$ and output of $H_2(Z)$ is the output of overall system which is $Y(Z)$. Thus Equation (10) becomes,

$$H_2(Z) = \frac{Y(Z)}{W(Z)}$$

Putting this value in Equation (9) we get,

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

$$\therefore \quad Y(Z)\left[1 + \sum_{k=1}^{N} a_k Z^{-k}\right] = W(Z)$$

$$\therefore \quad Y(Z) + Y(Z)\left[\sum_{k=1}^{N} a_k Z^{-k}\right] = W(Z)$$

$$\therefore \quad Y(Z) = -\left[\sum_{k=1}^{N} a_k Z^{-k}\right] Y(Z) + W(Z) \qquad \text{...(11)}$$

Expanding the summation we get,

$$Y(Z) = -\left[a_1 Z^{-1} + a_2 Z^{-2} + \dots + a_N Z^{-N}\right] Y(Z) + W(Z)$$

$$\therefore \quad Y(Z) = -a_1 Z^{-1} Y(Z) - a_2 Z^{-2} Y(Z) \dots - a_N Z^{-N} Y(Z) + W(Z) \qquad \text{...(12)}$$

Taking IZT of Equation (12) we get,

$$y(n) = -a_1 y(n-1) - a_2 y(n-2) \dots - a_N y(n-N) + w_n \qquad \text{...(13)}$$

Here $-a_1, -a_2 \dots -a_N$ are the coefficients.

The direct form implementation of Equation (13) is shown in Fig. I-1(c).

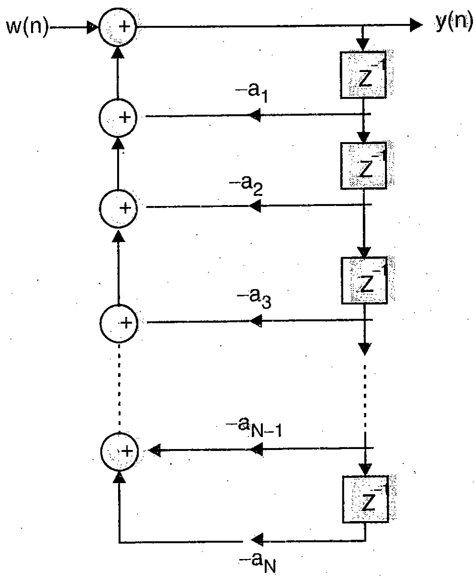


Fig. I-1(c) : Direct form realization of $H_2(Z)$ (All pole system)

Here all R.H.S. terms of Equation (13) except $w(n)$ are the delayed output terms. Thus this is a feedback (recursive) connection.

Now **direct form - I** structure is obtained by cascading $H_1(Z)$ and $H_2(Z)$. Thus direct form-I realization of IIR system is obtained by connecting Fig. I-1(b) and Fig. I-1(c) in series. It is shown in Fig. I-1(d).

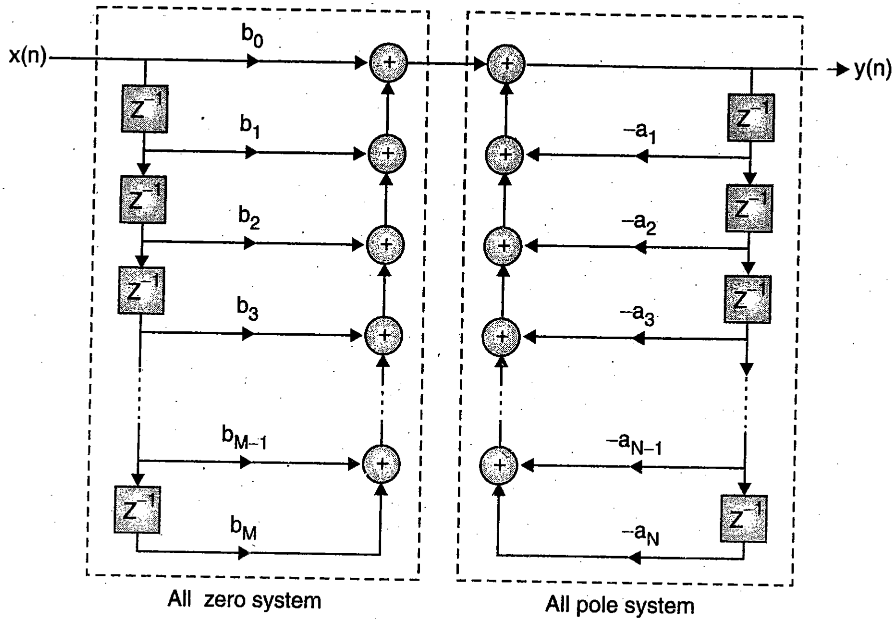


Fig. I-1(d) : Direct form-I realization of IIR system

Computation complexity :

The computational complexity of direct form-I structure is as follows :

1. Number of multiplications = $M + N + 1$
2. Number of additions = $M + N$
3. Number of memory locations = $M + N + 1$.

(B) Direct form-II structure :

Observe Fig. I-1(d). Here, first block represents all zero system and second block represents all pole system. Thus, in direct form- I structure ; zeros of $H(Z)$ are realised first and the poles of $H(Z)$ are realized second.

Now we are studying these structures for LTI (linear time invariant) systems. Since the systems are linear we can interchange the positions of $H_1(Z)$ and $H_2(Z)$. This will give us the direct form-II structure. Thus in direct form-II structure; poles are realised first and zeros second.

We have,
$$H(Z) = H_1(Z) \cdot H_2(Z) \quad \text{...(1)}$$

We will interchange the equations of $H_1(Z)$ and $H_2(Z)$.

$$\therefore \quad H_1(Z) = \frac{1}{1 + \sum_{k=1}^{N} a_k Z^{-k}} \quad \text{...(2)}$$

$$\text{and} \quad H_2(Z) = \sum_{k=0}^{M} b_k Z^{-k} \quad \text{...(3)}$$

The cascade connection of $H(Z) = H_1(Z) \cdot H_2(Z)$ is shown in Fig. I-2(a).

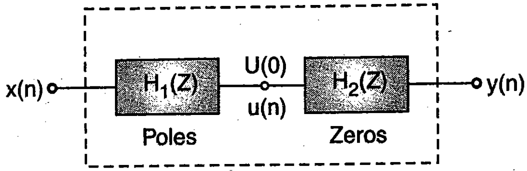


Fig. I-2(a) : Decomposition of direct form-II realization

Here $U(Z)$ represents the output of first block.

(1) All pole system :

We can write, $$H_1(Z) = \frac{\text{Output}}{\text{Input}} = \frac{U(Z)}{X(Z)} \qquad \text{...(4)}$$

Comparing Equations (2) and (3) we get,

$$\frac{U(Z)}{X(Z)} = \frac{1}{1 + \sum_{k=1}^{N} a_k Z^{-k}}$$

$$\therefore \quad U(Z)\left[1 + \sum_{k=1}^{N} a_k Z^{-k}\right] = X(Z)$$

$$\therefore \quad U(Z) + U(Z)\left[\sum_{k=1}^{N} a_k Z^{-k}\right] = X(Z)$$

$$\therefore \quad U(Z) = X(Z) - U(Z)\left[\sum_{k=1}^{N} a_k Z^{-k}\right]$$

Expanding the summation we get,

$$U(Z) = X(Z) - a_1 Z^{-1} U(Z) - a_2 Z^{-2} U(Z) \ldots - a_N Z^{-N} U(Z) \qquad \text{...(5)}$$

Taking IZT of Equation (5) we get,

$$u(n) = x(n) - a_1 x(n-1) - a_2 x(n-2) \ldots - a_N x(n-N) \qquad \text{...(6)}$$

The direct form implementation of Equation (6) is shown in Fig. I-2(b).

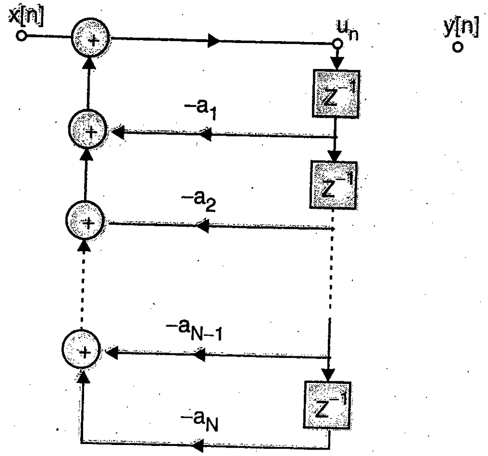


Fig. I-2(b) : Direct form realization of $H_1(Z)$ (All pole system)

(2) All zero system :

As shown in Fig. I-2(a), the output of second stage is the overall output of the system which is $Y(Z)$. While input to this stage is the output of first stage, which is $U(Z)$. Thus we can write,

$$H_2(Z) = \frac{\text{Output}}{\text{Input}} = \frac{Y(Z)}{U(Z)} \quad \text{...(7)}$$

Comparing Equations (3) and (7) we get,

$$\frac{Y(Z)}{U(Z)} = \sum_{k=0}^{M} b_k Z^{-k} \quad \text{...(8)}$$

$$\therefore \quad Y(Z) = U(Z)\left[\sum_{k=0}^{M} b_k Z^{-k}\right]$$

Expanding the summation we get,

$$Y(Z) = U(Z)\left[b_0 Z^0 + b_1 Z^{-1} + b_2 Z^{-2} + \dots + b_M Z^{-M}\right]$$

Here $Z^0 = 1$; multiplying by $U(Z)$ we get,

$$Y(Z) = b_0 U(Z) + b_1 Z^{-1} U(Z) + b_2 Z^{-2} U(Z) + \dots + b_M Z^{-M} U(Z) \quad \text{...(9)}$$

Taking IZT of Equation (9) we get,

$$y(n) = b_0 u(n) + b_1 u(n-1) + b_2 u(n-2) + \dots + b_m u(n-M) \quad \text{...(10)}$$

The direct form implementation of Equation (10) is shown in Fig. I-2(c).

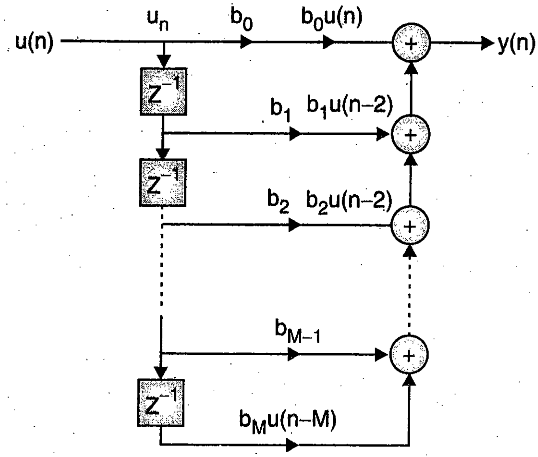


Fig. I-2(c) : Direct form realization of $H_2(Z)$ (All zero system)

Now direct form-II structure is obtained by cascading $H_1(Z)$ and $H_2(Z)$. That means it is obtained by connecting Fig. I-2(b) and Fig. I-2(c) in series. This is shown in Fig. I-2(d).

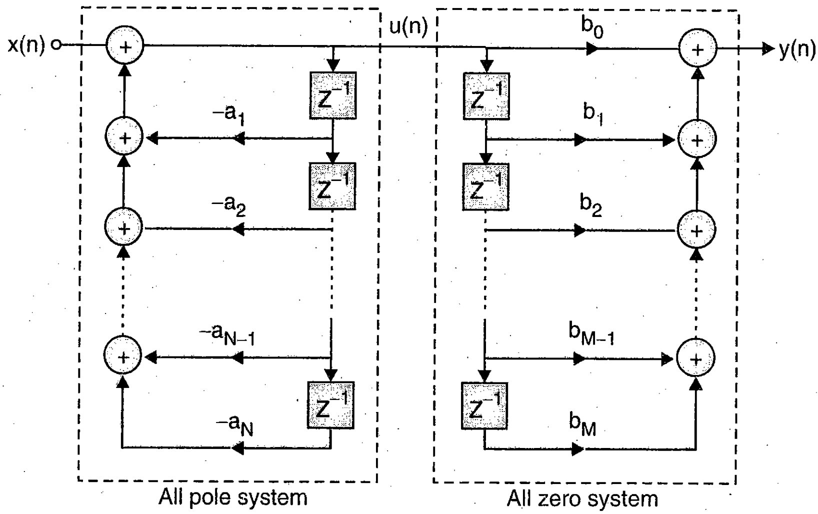


Fig. I-2(d) : Direct form-II realization of IIR system

In Fig. I-2(d), separate delay elements are used. Generally common delay elements are used in direct form-II structure. By using common delays, direct form II structure is drawn as shown in Fig. I-2(e).

Note that here we have considered N = M.

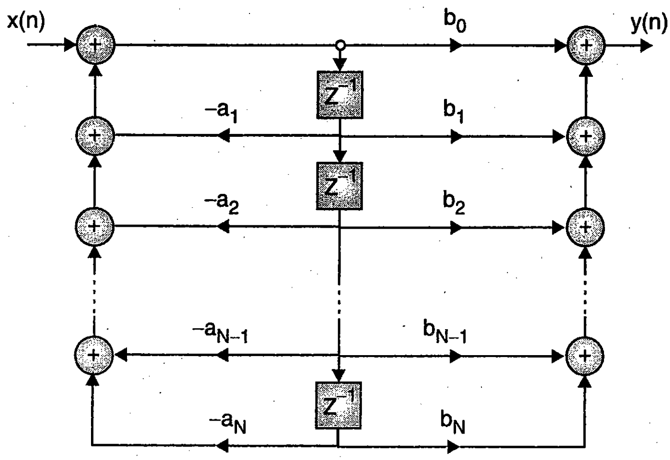


Fig. I-2(e) : Direct form-II realization using common delay elements

Computation complexity :

The computational complexity for direct form-II realization of IIR is as follows :

1. Number of multiplications = $M + N + 1$.
2. Number of additions = $M + N$.
3. The number of delay elements are reduced in direct form-II structure, compared to direct form-I structure. That means the memory locations are reduced. The memory locations required for direct form-II structure are $\{M, N\}$.

Why these structure are called as direct form structures ?

The direct form-I and direct form-II structures are obtained directly from the corresponding transfer functions without any rearrangements. So these structure is called as direct form structures.

Advantage : The only advantage of direct form realization is its implementation which is easy. We prefer direct form-II structure compared to direct form-I structure; because less memory locations are required.

Disadvantage : Both direct form structures are sensitive to the effects of quantization errors in the coefficients. So in the practical applications ; these structures are not preferred.

1.3 Cascade Form Structures :

To obtain the cascade form realization; the numerator and denominator of given transfer function $H(Z)$ is factored into the product of second order terms.

Then the total transfer function $H(Z)$ is expressed as,

$$H(Z) = H_1(Z) \cdot H_2(Z) \ldots\ldots H_k(Z) \qquad \ldots(1)$$

Here $H_1(Z)$, $H_2(Z)$ ... $H_k(Z)$ are second order polynomials. Then each subtransfer function $(H_1, H_2 \ldots \text{etc.})$ can be realised using direct form-I or direct form-II structures. The total transfer function is obtained by connecting all second order subsystems in series as shown in Fig. I-3(a).

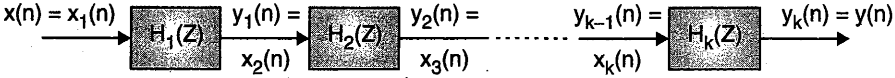


Fig. I-3(a) : Cascade form realization

Now we have the general difference equation for discrete time LTI system given by,

$$H(Z) = \frac{\sum_{k=0}^{M} b_k Z^{-k}}{1 + \sum_{k=1}^{N} a_k Z^{-k}} \qquad \text{...(2)}$$

Since $H_1(Z)$, $H_2(Z)$ are second order polynomials ; we can write the second order differential equation for $H_k(Z)$ by putting M = N = 2 in Equation (2).

$$\therefore \quad H_k(Z) = \frac{\sum_{k=0}^{2} b_k Z^{-k}}{1 + \sum_{k=1}^{2} a_k Z^{-k}}$$

$$\therefore \quad H_k(Z) = \frac{b_0 Z^0 + b_1 Z^{-1} + b_2 Z^{-2}}{1 + a_1 Z^{-1} + a_2 Z^{-2}}$$

But $\quad Z^0 = 1$

$$\therefore \quad H_k(Z) = \frac{b_0 + b_1 Z^{-1} + b_2 Z^{-2}}{1 + a_1 Z^{-1} + a_2 Z^{-2}} \qquad \text{...(3)}$$

We can obtain direct form-II structure of Equation (3), similar to Fig. I-2(e). Now, Fig. I-2(e) shows the direct form-II realization for N = M ; that means for M stages. In this case we have second order equation. So by using Fig. I-2(e) we can draw the direct form structure for Equation (3) by putting M = N = 2. This structure is shown in Fig. I-3(b).

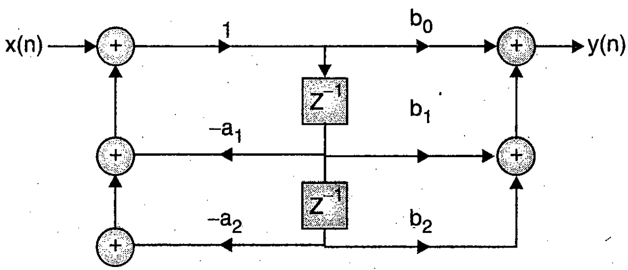


Fig. I-3(b) : Direct form-II realization of second order subsystem

So all such subsystems should be connected in series to obtain the cascade form realization of IIR system.

1.4 Parallel Form Structure :

We have the general difference equation for IIR systems given by,

$$H(Z) = \frac{\sum_{k=0}^{M} b_k Z^{-k}}{1 + \sum_{k=1}^{N} a_k Z^{-k}} \qquad ...(1)$$

By using partial fraction expansion we can express overall transfer function $H(Z)$ as,

$$H(Z) = C + H_1(Z) + H_2(Z) + \ldots + H_k(Z) \qquad ...(2)$$

Here 'C' is constant and $H_1(Z)$, $H_2(Z)$... $H_k(Z)$ are second order subsystems. The general block schematic of parallel form realization structure for IIR system is as shown in Fig. I-4.

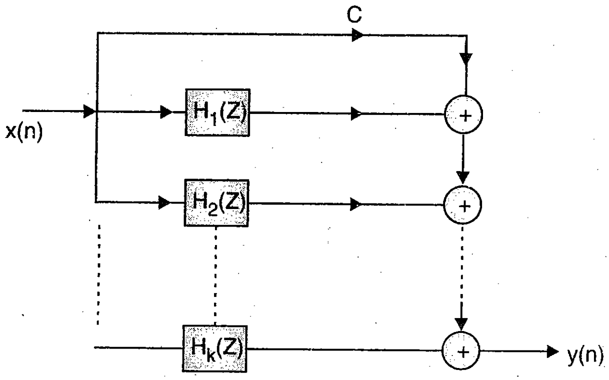


Fig. I-4 : Parallel form realization for IIR system

Here $H_1(Z)$, $H_2(Z)$...etc. can be realised by using direct form-I or direct form-II structures. Then all these structures are connected in parallel as shown in Fig. I-4 to obtain the parallel form realization for IIR system.

Applications :

The parallel form realization is generally used for high speed filtering applications. Since this is a parallel connection ; the processing of filtering operation is performed parallely.

1.5 Representation of Structures using Signal Flow Graphs :

Basically a signal flow graph is graphical representation of the block diagram structure. Both the signal flow graph and the block diagram structure provide the same information.
For example, we will consider the second order subsystem. We know that the transfer function of

second order subsystem is given by,

$$H(Z) = \frac{b_0 + b_1 Z^{-1} + b_2 Z^{-2}}{1 + a_1 Z^{-1} + a_2 Z^{-2}} \quad ...(1)$$

We have already drawn the direct form-II realization of second order subsystem (Fig. H-7(b)). For the reference it is again drawn as shown in Fig. I-5(a) and its signal flow graph representation is shown in Fig. I-5(b).

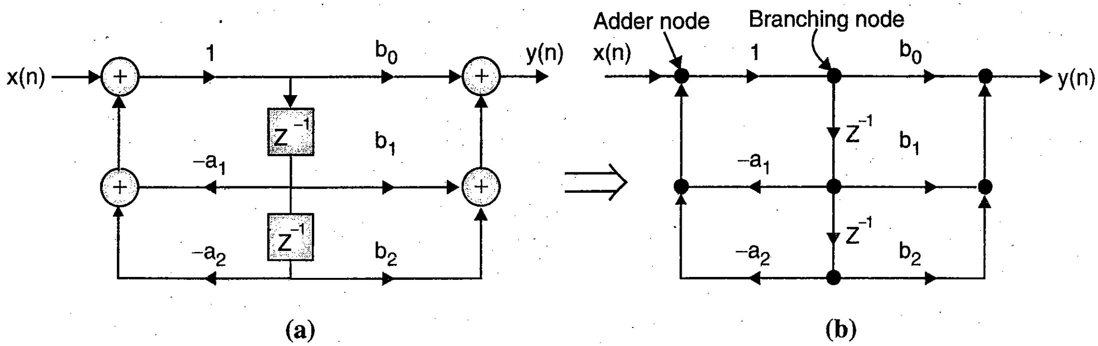


Fig. I-5 : Direct form-II realization and its signal flow graph

How to draw a signal flow graph ?

From the given block diagram realization; it is simple to draw the signal flow graph. The procedure is as follows :

1. Replace all adders by adder nodes.
2. Whenever there are different branches ; draw the branching node.
3. Keep the directions of arrows and the corresponding coefficients as it is.
4. Replace every delay element by simple transmittance branch. For that branch, write Z^{-1} to indicate the delay operation.

There is 1:1 correspondence between the signal flow graph and block diagram representation. And the signal flow graph representation is much simple compared to block diagram representation.

1.6 Transposed Structures :

If two digital filter structures have the same transfer function then they are called as equivalent structures. By using the transpose operation, we can obtain equivalent structure from a given realization structure.

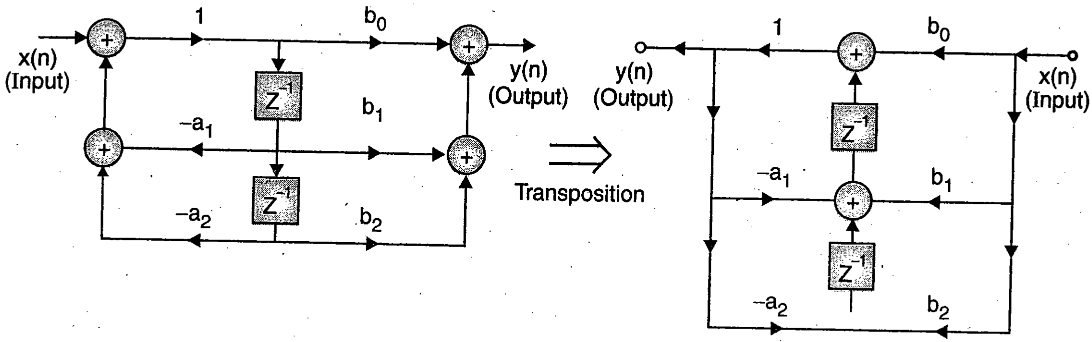
Transposition or flow graph reversal theorem :

If we reverse the directions of all branch transmittances and interchange input and output in the flow graph (or given structure) then the system transfer function remains unchanged. It is called as transposition theorem.

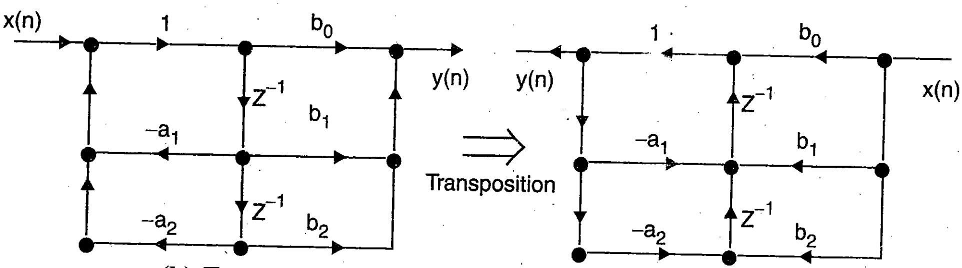
Procedure to obtain transposed structure and transposed flow graph :

1. Reverse all signal flow graph directions.
2. Change branching nodes into adders and vice-versa.
3. Interchange input and output.

The transposed structure for second order subsystem is shown in Fig. I-6(a) and the corresponding transposed flow graph is shown in Fig. I-6(b).



(a) Transposed structure for second order subsystem



(b) Transposed flow graph for second order subsystem

Fig. I-6

1.7 Feedback in IIR Systems :

We have studied that, in case of IIR systems, feedback connection is present. That means output signal is fed back to the input side. This is also called as a loop. Generally this feedback connection is required to generate infinitely long impulse responses.

First we will consider a system where feedback is not used. Consider the difference equation,

$$y(n) = x(n) + x(n-1) + x(n-2) \quad ...(1)$$

The block diagram representing Equation (1) is shown in Fig. I-7(a).

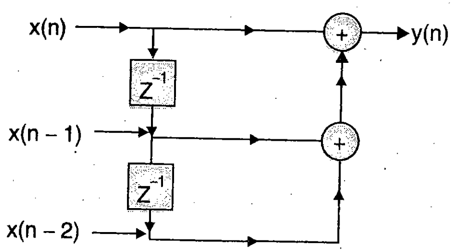


Fig. I-7(a)

As shown in Fig. I-7(a), a feedback path is not present. All the signals simply travels in the forward direction towards output. In this case an impulse response is no longer than the total number of delays in the system. Thus we can conclude that, if loops (feedback connections) are not present then the system function has only zeros (except for pole at z = 0). To prove this, we will take Z transform of Equation (1),

$$Y(Z) = X(Z) + Z^{-1}X(Z) + Z^{-2}X(Z)$$

$$\therefore \quad Y(Z) = X(Z)[1 + Z^{-1} + Z^{-2}]$$

$$\therefore \quad \frac{Y(Z)}{X(Z)} = H(Z) = 1 + Z^{-1} + Z^{-2}$$

Multiplying and dividing by Z^2 we get,

$$\therefore \quad H(Z) = \frac{Z^2 + Z + 1}{Z^2} \qquad ...(2)$$

Equation (2) shows that the system has only zeros except the poles at origin. The poles are not present at any other locations.

Now consider a system with feedback connection. Let us consider a difference equation,

$$y(n) = 2y(n-1) + x(n) \qquad ...(3)$$

Equation (3) indicates that, the delayed output, $y(n-1)$ is multiplied by constant value '2' and it is added with input $x(n)$. The block diagram representation and signal flow graph is shown in Fig. I-7(b).

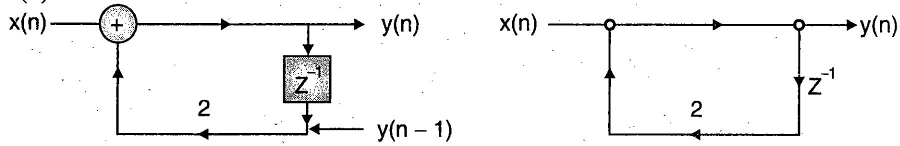


Fig. I-7(b)

If input is an impulse sequence then the single input sample continually recirculates in the feedback loop. Every time an amplitude increases due to multiplication by constant 2.

Now taking Z transform of both sides of Equation (3) we get,

$$Y(Z) = 2Z^{-1}Y(Z) + X(Z)$$

$$\therefore \quad Y(Z) - 2Z^{-1}Y(Z) = X(Z)$$

$$\therefore \quad Y(Z)[1 - 2Z^{-1}] = X(Z)$$

$$\therefore \quad \frac{Y(Z)}{X(Z)} = H(Z) = \frac{1}{1 - 2Z^{-1}}$$

$$\therefore \quad H(Z) = \frac{Z}{Z-2} \qquad ...(4)$$

Impulse response h(n) is obtained by taking inverse Z transform of Equation (4).

$$\therefore \quad h(n) = 2^n u(n) \qquad ...(5)$$

Equation (5) indicates that, because of the feedback connection an infinitely long impulse response is obtained.

Equation (4) indicates that the system function has a pole other than origin. This indicates the presence of feedback loop. But a network with feedback connection will not produce infinitely long impulse response if the poles of system function cancels with zeros.