

# DIGITAL SIGNAL PROCESSING

## LECTURE NOTES

**B.TECH**

**III-YEAR & I-SEM(R20)**

**Prepared by:**

**Mr. RAMESH BABU, Assistant Professor**

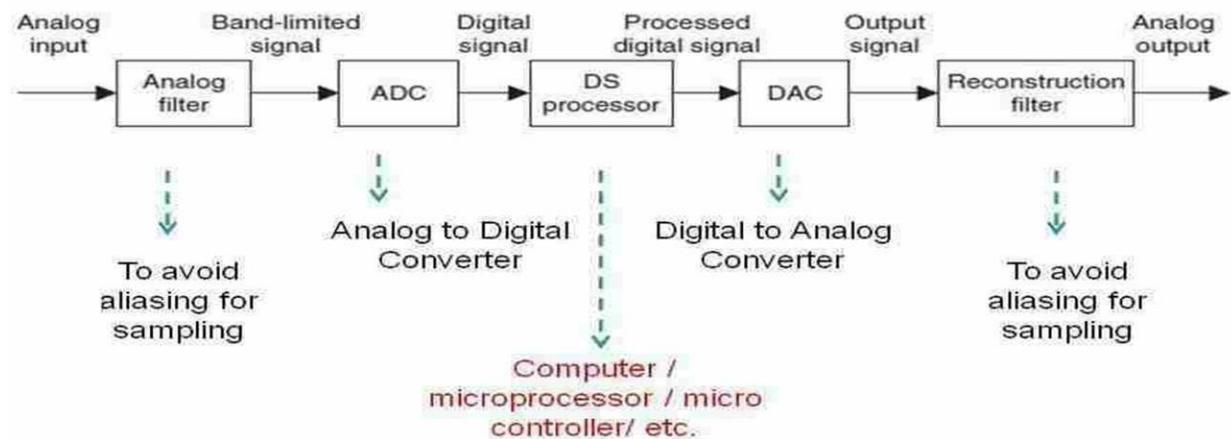
**Department of Electronics and Communication Engineering**

## UNIT -1

### Introduction to Digital Signal Processing

DSP is a technique of performing the mathematical operations on the signals in digital domain. As real time signals are analog in nature we need first convert the analog signal to digital, then we have to process the signal in digital domain and again converting back to analog domain. Thus ADC is required at the input side whereas a DAC is required at the output end. A typical DSP system is as shown in figure 1.1.

- The main function of low pass ant aliasing filter is to band limit the input signal to the folding frequency without distortion.
- It should be noted that even if the signal is band limited, there is always wide-band additive noise which will be folded back to create aliasing.
- When an analog voltage is connected directly to an ADC, the conversion process can be adversely affected if the voltage is changing during the conversion time.
- The quality of conversion process can be improved by using sample and hold circuit



#### Advantages of DSP

- Programmability: software digital signal processes can be quickly modified, in contrast to analog circuits, which must be physically rearranged.
- Versatility: Flexible and easy to upgrade.
- Stability: Less sensitive environmental changes such as electromagnetic interference.

## Need for DSP

Analog signal Processing has the following drawbacks:

- They are sensitive to environmental changes
- Aging
- Uncertain performance in production units
- Variation in performance of units
- Cost of the system will be high
- Scalability

If Digital Signal Processing would have been used we can overcome the above shortcomings of ASP.

## **REVIEW OF DISCRETE TIME SIGNALS AND SYSTEMS**

Anything that carries some information can be called as signals. Some examples are ECG, EEG, ac power, seismic, speech, interest rates of a bank, unemployment rate of a country, temperature, pressure etc.

A **signal** is also defined as any physical quantity that varies with one or more independent variables.

A **discrete time signal** is the one which is not defined at intervals between two successive samples of a signal. It is represented as graphical, functional, tabular representation and sequence.

**Some of the elementary discrete time signals are** unit step, unit impulse, unit ramp, exponential and sinusoidal signals (as you read in signals and systems).

### **Classification of discrete time signals**

#### **Energy and Power signals**

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

If the value of E is finite, then the signal x(n) is called energy signal.

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

If the value of the P is finite, then the signal x(n) is called Power signal.

#### **Periodic and Non periodic signals**

A discrete time signal is said to be periodic if and only if it satisfies the condition  $X(N+n) = x(n)$ , otherwise non periodic

#### **Symmetric (even) and Anti-symmetric (odd) signals**

The signal is said to be even if  $x(-n) = x(n)$  The signal is said to be odd if  $x(-n) = -x(n)$

#### **Causal and non causal signal**

The signal is said to be causal if its value is zero for negative values of 'n'.

Some of the **operations** on discrete time signals are shifting, time reversal, time scaling, signal multiplier, scalar multiplication and signal addition or multiplication.

#### **Discrete time systems**

A discrete time signal is a device or algorithm that operates on discrete time signals and produces another discrete time output.

### **Classification of discrete time systems**

#### **Static and dynamic systems**

A system is said to be static if its output at present time depend on the input at present time only.

### **Causal and non causal systems**

A system is said to be causal if the response of the system depends on present and past values of the input but not on the future inputs.

### **Linear and non linear systems**

A system is said to be linear if the response of the system to the weighted sum of inputs should be equal to the corresponding weighted sum of outputs of the systems. This principle is called superposition principle.

### **Time invariant and time variant systems**

A system is said to be time invariant if the characteristics of the systems do not change with time.

### **Stable and unstable systems**

A system is said to be stable if bounded input produces bounded output only.

## **TIME DOMAIN ANALYSIS OF DISCRETE TIME SIGNALS AND SYSTEMS**

### **Representation of an arbitrary sequence**

Any signal  $x(n)$  can be represented as weighted sum of impulses as given below

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

The response of the system for unit sample input is called impulse response of the system  $h(n)$

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

By time invariant property, we have

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

The above equation is called **convolution sum**.

Some of the properties of convolution are commutative law, associative law and distributive law.

### **Correlation of two sequences**

It is basically used to compare two signals. It is the measure of similarity between two signals. Some of the applications are communication systems, radar, sonar etc.

The cross correlation of two sequences  $x(n)$  and  $y(n)$  is given by

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

One of the important properties of cross correlation is given by

$$r_{xy}(l) = r_{yx}(-l)$$

The auto correlation of the signal  $x(n)$  is given by

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

### Linear time invariant systems characterized by constant coefficient difference equation

The response of the first order difference equation is given by

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

The first part contain initial condition  $y(-1)$  of the system, the second part contains input  $x(n)$  of the system.

The response of the system when it is in relaxed state at  $n=0$  or  $y(-1)=0$  is called **zero state response** of the system or **forced response**.

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

The output of the system at zero input condition  $x(n)=0$  is called **zero input response** of the system or **natural response**.

The impulse response of the system is given by zero state response of the system

$$\begin{aligned} y_{zs}(n) &= \sum_{k=0}^n a^k \delta(n-k) \\ &= a^n \quad n \geq 0 \end{aligned}$$

The total response of the system is equal to sum of natural response and forced responses.

## UNIT-II

### Discrete Fourier Transform

#### Discrete Fourier Series

The Fourier series representation of a continuous-time periodic signal can consist of an infinite number of frequency components, where the frequency spacing between two successive harmonically related frequencies is  $1/Tp$ , and where  $Tp$  is the fundamental period.

Since the frequency range for continuous-time signals extends infinity on both sides it is possible to have signals that contain an infinite number of frequency components.

In contrast, the frequency range for discrete-time signals is unique over the interval. A discrete-time signal of fundamental period  $N$  can consist of frequency components separated by  $2\pi/N$  radians.

Consequently, the Fourier series representation of the discrete-time periodic signal will contain at most  $N$  frequency components. This is the basic difference between the Fourier series representations for continuous-time and discrete-time periodic signals.

The expression for the Fourier coefficients  $c_k$  can be obtained by multiplying both sides of (4.2.1) by the exponential  $e^{-j2\pi ln/N}$  and summing the product from  $n = 0$  to  $n = N - 1$ . Thus

$$\sum_{n=0}^{N-1} x(n)e^{-j2\pi ln/N} = \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} c_k e^{j2\pi(k-l)n/N} \quad (4.2.4)$$

If we perform the summation over  $n$  first, in the right-hand side of (4.2.4), we obtain

$$\sum_{n=0}^{N-1} e^{j2\pi(k-l)n/N} = \begin{cases} N, & k-l = 0, \pm N, \pm 2N, \dots \\ 0, & \text{otherwise} \end{cases} \quad (4.2.5)$$

where we have made use of (4.2.2). Therefore, the right-hand side of (4.2.4) reduces to  $Nc_l$  and hence

$$c_l = \frac{1}{N} \sum_{n=0}^{N-1} x(n)e^{-j2\pi ln/N} \quad l = 0, 1, \dots, N-1 \quad (4.2.6)$$

Thus we have the desired expression for the Fourier coefficients in terms of the signal  $x(n)$ .

### 4.2.3 The Fourier Transform of Discrete-Time Aperiodic Signals

Just as in the case of continuous-time aperiodic energy signals, the frequency analysis of discrete-time aperiodic finite-energy signals involves a Fourier transform of the time-domain signal. Consequently, the development in this section parallels to a large extent, that given in Section 4.1.3.

The Fourier transform of a finite-energy discrete-time signal  $x(n)$  is defined as

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \quad (4.2.23)$$

Physically,  $X(\omega)$  represents the frequency content of the signal  $x(n)$ . In other words,  $X(\omega)$  is a decomposition of  $x(n)$  into its frequency components.

We observe two basic differences between the Fourier transform of a discrete-time finite-energy signal and the Fourier transform of a finite-energy analog signal. First, for continuous-time signals, the Fourier transform, and hence the spectrum of the signal, have a frequency range of  $(-\infty, \infty)$ . In contrast, the frequency range for a discrete-time signal is unique over the frequency interval of  $(-\pi, \pi)$  or, equivalently,  $(0, 2\pi)$ . This property is reflected in the Fourier transform of the signal. Indeed,  $X(\omega)$  is periodic with period  $2\pi$ , that is,

$$\begin{aligned} X(\omega + 2\pi k) &= \sum_{n=-\infty}^{\infty} x(n)e^{-j(\omega+2\pi k)n} \\ &= \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} e^{-j2\pi kn} \\ &= \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} = X(\omega) \end{aligned} \quad (4.2.24)$$

Hence  $X(\omega)$  is periodic with period  $2\pi$ . But this property is just a consequence of the fact that the frequency range for any discrete-time signal is limited to  $(-\pi, \pi)$  or  $(0, 2\pi)$ , and any frequency outside this interval is equivalent to a frequency within the interval.

The second basic difference is also a consequence of the discrete-time nature of the signal. Since the signal is discrete in time, the Fourier transform of the signal involves a summation of terms instead of an integral, as in the case of continuous-time signals.

Since  $X(\omega)$  is a periodic function of the frequency variable  $\omega$ , it has a Fourier series expansion, provided that the conditions for the existence of the Fourier series, described previously, are satisfied. In fact, from the definition of the Fourier transform  $X(\omega)$  of the sequence  $x(n)$ , given by (4.2.23), we observe that  $X(\omega)$  has the form of a Fourier series. The Fourier coefficients in this series expansion are the values of the sequence  $x(n)$ .

To demonstrate this point, let us evaluate the sequence  $x(n)$  from  $X(\omega)$ . First, we multiply both sides (4.2.23) by  $e^{j\omega m}$  and integrate over the interval  $(-\pi, \pi)$ . Thus we have

$$\int_{-\pi}^{\pi} X(\omega) e^{j\omega m} d\omega = \int_{-\pi}^{\pi} \left[ \sum_{n=-\infty}^{\infty} x(n) e^{-j\omega n} \right] e^{j\omega m} d\omega \quad (4.2.25)$$

The integral on the right-hand side of (4.2.25) can be evaluated if we can interchange the order of summation and integration. This interchange can be made if the series

$$X_N(\omega) = \sum_{n=-N}^N x(n) e^{-j\omega n}$$

converges uniformly to  $X(\omega)$  as  $N \rightarrow \infty$ . Uniform convergence means that, for every  $\omega$ ,  $X_N(\omega) \rightarrow X(\omega)$ , as  $N \rightarrow \infty$ . The convergence of the Fourier transform is discussed in more detail in the following section. For the moment, let us assume that the series converges uniformly, so that we can interchange the order of summation and integration in (4.2.25). Then

$$\int_{-\pi}^{\pi} e^{j\omega(m-n)} d\omega = \begin{cases} 2\pi, & m = n \\ 0, & m \neq n \end{cases}$$

Consequently,

$$\sum_{n=-\infty}^{\infty} x(n) \int_{-\pi}^{\pi} e^{j\omega(m-n)} d\omega = \begin{cases} 2\pi x(m), & m = n \\ 0, & m \neq n \end{cases} \quad (4.2.26)$$

By combining (4.2.25) and (4.2.26), we obtain the desired result that

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega) e^{j\omega n} d\omega \quad (4.2.27)$$

If we compare the integral in (4.2.27) with (4.1.9), we note that this is just the expression for the Fourier series coefficient for a function that is periodic with period  $2\pi$ . The only difference between (4.1.9) and (4.2.27) is the sign on the exponent in the integrand, which is a consequence of our definition of the Fourier transform as given by (4.2.23). Therefore, the Fourier transform of the sequence  $x(n)$ , defined by (4.2.23), has the form of a Fourier series expansion.

## FREQUENCY DOMAIN SAMPLING: THE DISCRETE FOURIER TRANSFORM

Before we introduce the DFT, we consider the sampling of the Fourier transform of an aperiodic discrete-time sequence. Thus, we establish the relationship between the sampled Fourier transform and the DFT.

### 5.1.1 Frequency-Domain Sampling and Reconstruction of Discrete-Time Signals

We recall that aperiodic finite-energy signals have continuous spectra. Let us consider such an aperiodic discrete-time signal  $x(n)$  with Fourier transform

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \quad (5.1.1)$$

Suppose that we sample  $X(\omega)$  periodically in frequency at a spacing of  $\delta\omega$  radians between successive samples. Since  $X(\omega)$  is periodic with period  $2\pi$ , only samples in the fundamental frequency range are necessary. For convenience, we take  $N$  equidistant samples in the interval  $0 \leq \omega < 2\pi$  with spacing  $\delta\omega = 2\pi/N$ , as shown in Fig. 5.1. First, we consider the selection of  $N$ , the number of samples in the frequency domain.

If we evaluate (5.1.1) at  $\omega = 2\pi k/N$ , we obtain

$$X\left(\frac{2\pi}{N}k\right) = \sum_{n=-\infty}^{\infty} x(n)e^{-j2\pi kn/N} \quad k = 0, 1, \dots, N-1 \quad (5.1.2)$$

The summation in (5.1.2) can be subdivided into an infinite number of summations, where each sum contains  $N$  terms. Thus

$$\begin{aligned} X\left(\frac{2\pi}{N}k\right) &= \dots + \sum_{n=-N}^{-1} x(n)e^{-j2\pi kn/N} + \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} \\ &\quad + \sum_{n=N}^{2N-1} x(n)e^{-j2\pi kn/N} + \dots \\ &= \sum_{l=-\infty}^{\infty} \sum_{n=lN}^{lN+N-1} x(n)e^{-j2\pi kn/N} \end{aligned}$$

If we change the index in the inner summation from  $n$  to  $n - lN$  and interchange the order of the summation, we obtain the result

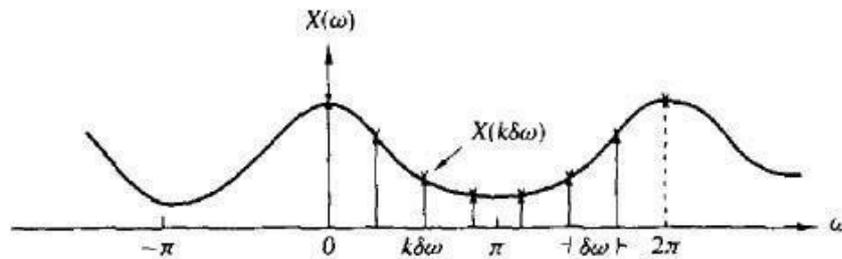
$$X\left(\frac{2\pi}{N}k\right) = \sum_{n=0}^{N-1} \left[ \sum_{l=-\infty}^{\infty} x(n-lN) \right] e^{-j2\pi kn/N} \quad (5.1.3)$$

for  $k = 0, 1, 2, \dots, N-1$ .

The signal

$$x_p(n) = \sum_{l=-\infty}^{\infty} x(n-lN) \quad (5.1.4)$$

obtained by the periodic repetition of  $x(n)$  every  $N$  samples, is clearly periodic with fundamental period  $N$ . Consequently, it can be expanded in a Fourier



**Figure 5.1** Frequency-domain sampling of the Fourier transform.

series as

$$x_p(n) = \sum_{k=0}^{N-1} c_k e^{j2\pi kn/N} \quad n = 0, 1, \dots, N-1 \quad (5.1.5)$$

with Fourier coefficients

$$c_k = \frac{1}{N} \sum_{n=0}^{N-1} x_p(n) e^{-j2\pi kn/N} \quad k = 0, 1, \dots, N-1 \quad (5.1.6)$$

Upon comparing (5.1.3) with (5.1.6), we conclude that

$$c_k = \frac{1}{N} X\left(\frac{2\pi}{N}k\right) \quad k = 0, 1, \dots, N-1 \quad (5.1.7)$$

Therefore,

$$x_p(n) = \frac{1}{N} \sum_{k=0}^{N-1} X\left(\frac{2\pi}{N}k\right) e^{j2\pi kn/N} \quad n = 0, 1, \dots, N-1 \quad (5.1.8)$$

### 5.1.2 The Discrete Fourier Transform (DFT)

The development in the preceding section is concerned with the frequency-domain sampling of an aperiodic finite-energy sequence  $x(n)$ . In general, the equally spaced frequency samples  $X(2\pi k/N)$ ,  $k = 0, 1, \dots, N-1$ , do not uniquely represent the original sequence  $x(n)$  when  $x(n)$  has infinite duration. Instead, the frequency samples  $X(2\pi k/N)$ ,  $k = 0, 1, \dots, N-1$ , correspond to a periodic sequence  $x_p(n)$  of period  $N$ , where  $x_p(n)$  is an aliased version of  $x(n)$ , as indicated by the relation in (5.1.4), that is,

$$x_p(n) = \sum_{l=-\infty}^{\infty} x(n - lN) \quad (5.1.15)$$

When the sequence  $x(n)$  has a finite duration of length  $L \leq N$ , then  $x_p(n)$  is simply a periodic repetition of  $x(n)$ , where  $x_p(n)$  over a single period is

given as

$$x_p(n) = \begin{cases} x(n), & 0 \leq n \leq L-1 \\ 0, & L \leq n \leq N-1 \end{cases} \quad (5.1.16)$$

Consequently, the frequency samples  $X(2\pi k/N)$ ,  $k = 0, 1, \dots, N-1$ , uniquely represent the finite-duration sequence  $x(n)$ . Since  $x(n) \equiv x_p(n)$  over a single period (padded by  $N-L$  zeros), the original finite-duration sequence  $x(n)$  can be obtained from the frequency samples  $\{X(2\pi k/N)\}$  by means of the formula (5.1.8).

It is important to note that *zero padding* does not provide any additional information about the spectrum  $X(\omega)$  of the sequence  $\{x(n)\}$ . The  $L$  equidis-

tant samples of  $X(\omega)$  are sufficient to reconstruct  $X(\omega)$  using the reconstruction formula (5.1.13). However, padding the sequence  $\{x(n)\}$  with  $N-L$  zeros and computing an  $N$ -point DFT results in a "better display" of the Fourier transform  $X(\omega)$ .

**In summary**, a finite-duration sequence  $x(n)$  of length  $L$  [i.e.,  $x(n) = 0$  for  $n < 0$  and  $n \geq L$ ] has a Fourier transform

$$X(\omega) = \sum_{n=0}^{L-1} x(n)e^{-j\omega n} \quad 0 \leq \omega \leq 2\pi \quad (5.1.17)$$

where the upper and lower indices in the summation reflect the fact that  $x(n) = 0$  outside the range  $0 \leq n \leq L-1$ . When we sample  $X(\omega)$  at equally spaced frequencies  $\omega_k = 2\pi k/N$ ,  $k = 0, 1, 2, \dots, N-1$ , where  $N \geq L$ , the resultant samples are

$$X(k) \equiv X\left(\frac{2\pi k}{N}\right) = \sum_{n=0}^{L-1} x(n)e^{-j2\pi kn/N} \quad (5.1.18)$$

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} \quad k = 0, 1, 2, \dots, N-1$$

where for convenience, the upper index in the sum has been increased from  $L-1$  to  $N-1$  since  $x(n) = 0$  for  $n \geq L$ .

The relation in (5.1.18) is a formula for transforming a sequence  $\{x(n)\}$  of length  $L \leq N$  into a sequence of frequency samples  $\{X(k)\}$  of length  $N$ . Since the frequency samples are obtained by evaluating the Fourier transform  $X(\omega)$  at a set of  $N$  (equally spaced) discrete frequencies, the relation in (5.1.18) is called the *discrete Fourier transform* (DFT) of  $x(n)$ . In turn, the relation given by (5.1.19), which allows us to recover the sequence  $x(n)$  from the frequency samples

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi kn/N} \quad n = 0, 1, \dots, N-1 \quad (5.1.19)$$

is called the *inverse DFT* (IDFT). Clearly, when  $x(n)$  has length  $L < N$ , the  $N$ -point IDFT yields  $x(n) = 0$  for  $L \leq n \leq N-1$ . To summarize, the formulas for the DFT and IDFT are

**DFT**

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} \quad k = 0, 1, 2, \dots, N-1 \quad (5.1.18)$$

**IDFT**

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi kn/N} \quad n = 0, 1, 2, \dots, N-1 \quad (5.1.19)$$

### 5.1.3 The DFT as a Linear Transformation

The formulas for the DFT and IDFT given by (5.1.18) and (5.1.19) may be expressed as

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \quad k = 0, 1, \dots, N-1 \quad (5.1.20)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn} \quad n = 0, 1, \dots, N-1 \quad (5.1.21)$$

where, by definition,

$$W_N = e^{-j2\pi/N} \quad (5.1.22)$$

which is an  $N$ th root of unity.

With these definitions, the  $N$ -point DFT may be expressed in matrix form as

$$\mathbf{X}_N = \mathbf{W}_N \mathbf{x}_N \quad (5.1.24)$$

where  $\mathbf{W}_N$  is the matrix of the linear transformation. We observe that  $\mathbf{W}_N$  is a symmetric matrix. If we assume that the inverse of  $\mathbf{W}_N$  exists, then (5.1.24) can be inverted by premultiplying both sides by  $\mathbf{W}_N^{-1}$ . Thus we obtain

$$\mathbf{x}_N = \mathbf{W}_N^{-1} \mathbf{X}_N \quad (5.1.25)$$

**Relationship to the Fourier series coefficients of a periodic sequence.**

A periodic sequence  $\{x_p(n)\}$  with fundamental period  $N$  can be represented in a Fourier series of the form

$$x_p(n) = \sum_{k=0}^{N-1} c_k e^{j2\pi nk/N} \quad -\infty < n < \infty \quad (5.1.29)$$

where the Fourier series coefficients are given by the expression

$$c_k = \frac{1}{N} \sum_{n=0}^{N-1} x_p(n) e^{-j2\pi nk/N} \quad k = 0, 1, \dots, N-1 \quad (5.1.30)$$

If we compare (5.1.29) and (5.1.30) with (5.1.18) and (5.1.19), we observe that the formula for the Fourier series coefficients has the form of a DFT. In fact, if we define a sequence  $x(n) = x_p(n)$ ,  $0 \leq n \leq N-1$ , the DFT of this sequence is simply

$$X(k) = Nc_k \quad (5.1.31)$$

Furthermore, (5.1.29) has the form of an IDFT. Thus the  $N$ -point DFT provides the exact line spectrum of a periodic sequence with fundamental period  $N$ .

**Relationship to the Fourier transform of an aperiodic sequence.** We have already shown that if  $x(n)$  is an aperiodic finite energy sequence with Fourier transform  $X(\omega)$ , which is sampled at  $N$  equally spaced frequencies  $\omega_k = 2\pi k/N$ ,  $k = 0, 1, \dots, N-1$ , the spectral components

$$X(k) = X(\omega)|_{\omega=2\pi k/N} = \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi nk/N} \quad k = 0, 1, \dots, N-1 \quad (5.1.32)$$

are the DFT coefficients of the periodic sequence of period  $N$ , given by

$$x_p(n) = \sum_{l=-\infty}^{\infty} x(n - lN) \quad (5.1.33)$$

Thus  $x_p(n)$  is determined by aliasing  $\{x(n)\}$  over the interval  $0 \leq n \leq N-1$ . The finite-duration sequence

$$\hat{x}(n) = \begin{cases} x_p(n), & 0 \leq n \leq N-1 \\ 0, & \text{otherwise} \end{cases} \quad (5.1.34)$$

bears no resemblance to the original sequence  $\{x(n)\}$ , unless  $x(n)$  is of finite duration and length  $L \leq N$ , in which case

$$x(n) = \hat{x}(n) \quad 0 \leq n \leq N - 1 \quad (5.1.35)$$

Only in this case will the IDFT of  $\{X(k)\}$  yield the original sequence  $\{x(n)\}$ .

**Relationship to the z-transform.** Let us consider a sequence  $x(n)$  having the z-transform

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n} \quad (5.1.36)$$

with a ROC that includes the unit circle. If  $X(z)$  is sampled at the  $N$  equally spaced points on the unit circle  $z_k = e^{j2\pi k/N}$ ,  $0, 1, 2, \dots, N - 1$ , we obtain

$$\begin{aligned} X(k) &\equiv X(z)|_{z=e^{j2\pi k/N}} \quad k = 0, 1, \dots, N - 1 \\ &= \sum_{n=-\infty}^{\infty} x(n)e^{-j2\pi nk/N} \end{aligned} \quad (5.1.37)$$

The expression in (5.1.37) is identical to the Fourier transform  $X(\omega)$  evaluated at the  $N$  equally spaced frequencies  $\omega_k = 2\pi k/N$ ,  $k = 0, 1, \dots, N - 1$ , which is the topic treated in Section 5.1.1.

If the sequence  $x(n)$  has a finite duration of length  $N$  or less, the sequence can be recovered from its  $N$ -point DFT. Hence its z-transform is uniquely determined by its  $N$ -point DFT. Consequently,  $X(z)$  can be expressed as a function of the DFT  $\{X(k)\}$  as follows

$$\begin{aligned} X(z) &= \sum_{n=0}^{N-1} x(n)z^{-n} \\ X(z) &= \sum_{n=0}^{N-1} \left[ \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j2\pi kn/N} \right] z^{-n} \\ X(z) &= \frac{1}{N} \sum_{k=0}^{N-1} X(k) \sum_{n=0}^{N-1} (e^{j2\pi k/N} z^{-1})^n \\ X(z) &= \frac{1 - z^{-N}}{N} \sum_{k=0}^{N-1} \frac{X(k)}{1 - e^{j2\pi k/N} z^{-1}} \end{aligned} \quad (5.1.38)$$

When evaluated on the unit circle, (5.1.38) yields the Fourier transform of the finite-duration sequence in terms of its DFT, in the form

$$X(\omega) = \frac{1 - e^{-j\omega N}}{N} \sum_{k=0}^{N-1} \frac{X(k)}{1 - e^{-j(\omega - 2\pi k/N)}} \quad (5.1.39)$$

**Relationship to the Fourier series coefficients of a continuous-time signal.** Suppose that  $x_a(t)$  is a continuous-time periodic signal with fundamental period  $T_p = 1/F_0$ . The signal can be expressed in a Fourier series

$$x_a(t) = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k F_0 t} \quad (5.1.40)$$

where  $\{c_k\}$  are the Fourier coefficients. If we sample  $x_a(t)$  at a uniform rate  $F_s = N/T_p = 1/T$ , we obtain the discrete-time sequence

$$\begin{aligned} x(n) \equiv x_a(nT) &= \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k F_0 n T} = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k n / N} \\ &= \sum_{k=0}^{N-1} \left[ \sum_{l=-\infty}^{\infty} c_{k-lN} \right] e^{j2\pi k n / N} \end{aligned} \quad (5.1.41)$$

It is clear that (5.1.41) is in the form of an IDFT formula, where

$$X(k) = N \sum_{l=-\infty}^{\infty} c_{k-lN} \equiv N \tilde{c}_k \quad (5.1.42)$$

and

$$\tilde{c}_k = \sum_{l=-\infty}^{\infty} c_{k-lN} \quad (5.1.43)$$

Thus the  $\{\tilde{c}_k\}$  sequence is an aliased version of the sequence  $\{c_k\}$ .

#### PROPERTIES OF DFT:

Property	Time Domain	Frequency Domain
Notation	$x(n), y(n)$	$X(k), Y(k)$
Periodicity	$x(n) \equiv x(n + N)$	$X(k) = X(k + N)$
Linearity	$a_1 x_1(n) + a_2 x_2(n)$	$a_1 X_1(k) + a_2 X_2(k)$
Time reversal	$x(N - n)$	$X(N - k)$
Circular time shift	$x((n - l))_N$	$X(k) e^{-j2\pi k l / N}$
Circular frequency shift	$x(n) e^{j2\pi l n / N}$	$X((k - l))_N$
Complex conjugate	$x^*(n)$	$X^*(N - k)$
Circular convolution	$x_1(n) \textcircled{N} x_2(n)$	$X_1(k) X_2(k)$
Circular correlation	$x(n) \textcircled{N} y^*(-n)$	$X(k) Y^*(k)$
Multiplication of two sequences	$x_1(n) x_2(n)$	$\frac{1}{N} X_1(k) \textcircled{N} X_2(k)$
Parseval's theorem	$\sum_{n=0}^{N-1} x(n) y^*(n)$	$\frac{1}{N} \sum_{k=0}^{N-1} X(k) Y^*(k)$

#### LINEAR FILTERING METHODS BASED ON THE DFT

Since the DFT provides a discrete frequency representation of a finite-duration Sequence in the frequency domain, it is interesting to explore its use as a computational tool for linear system analysis and, especially, for linear filtering. We have already established that a system with frequency response  $H(\omega)$  when excited with an input signal that has a spectrum possesses an output spectrum.

The output sequence  $y(n)$  is determined from its spectrum via the inverse Fourier transform. Computationally, the problem with this frequency domain approach is that are functions of the

continuous variable. As a consequence, the computations cannot be done on a digital computer, since the computer can only store and perform computations on quantities at discrete frequencies. On the other hand, the DFT does lend itself to computation on a digital computer. In the discussion that follows, we describe how the DFT can be used to perform linear filtering in the frequency domain. In particular, we present a computational procedure that serves as an alternative to time-domain convolution.

In fact, the frequency-domain approach based on the DFT, is computationally more efficient than time-domain convolution due to the existence of efficient algorithms for computing the DFT. These algorithms, which are described in Chapter 6, are collectively called fast Fourier transform (FFT) algorithms.

### 5.3.1 Use of the DFT in Linear Filtering

In the preceding section it was demonstrated that the product of two DFTs is equivalent to the circular convolution of the corresponding time-domain sequences. Unfortunately, circular convolution is of no use to us if our objective is to determine the output of a linear filter to a given input sequence. In this case we seek a frequency-domain methodology equivalent to linear convolution.

Suppose that we have a finite-duration sequence  $x(n)$  of length  $L$  which excites an FIR filter of length  $M$ . Without loss of generality, let

$$\begin{aligned} x(n) &= 0, & n < 0 \text{ and } n \geq L \\ h(n) &= 0, & n < 0 \text{ and } n \geq M \end{aligned}$$

where  $h(n)$  is the impulse response of the FIR filter.

The output sequence  $y(n)$  of the FIR filter can be expressed in the time domain as the convolution of  $x(n)$  and  $h(n)$ , that is

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k) \quad (5.3.1)$$

Since  $h(n)$  and  $x(n)$  are finite-duration sequences, their convolution is also finite in duration. In fact, the duration of  $y(n)$  is  $L + M - 1$ .

The frequency-domain equivalent to (5.3.1) is

$$Y(\omega) = X(\omega)H(\omega) \quad (5.3.2)$$

If the sequence  $y(n)$  is to be represented uniquely in the frequency domain by samples of its spectrum  $Y(\omega)$  at a set of discrete frequencies, the number of distinct samples must equal or exceed  $L + M - 1$ . Therefore, a DFT of size  $N \geq L + M - 1$ , is required to represent  $\{y(n)\}$  in the frequency domain.

Now if

$$\begin{aligned} Y(k) &\equiv Y(\omega)|_{\omega=2\pi k/N} & k = 0, 1, \dots, N - 1 \\ &= X(\omega)H(\omega)|_{\omega=2\pi k/N} & k = 0, 1, \dots, N - 1 \end{aligned}$$

then

$$Y(k) = X(k)H(k) \quad k = 0, 1, \dots, N - 1 \quad (5.3.3)$$

where  $\{X(k)\}$  and  $\{H(k)\}$  are the  $N$ -point DFTs of the corresponding sequences  $x(n)$  and  $h(n)$ , respectively. Since the sequences  $x(n)$  and  $h(n)$  have a duration less than  $N$ , we simply pad these sequences with zeros to increase their length to  $N$ . This increase in the size of the sequences does not alter their spectra  $X(\omega)$  and  $H(\omega)$ , which are continuous spectra, since the sequences are aperiodic. However, by sampling their spectra at  $N$  equally spaced points in frequency (computing the  $N$ -point DFTs), we have increased the number of samples that represent these sequences in the frequency domain beyond the minimum number ( $L$  or  $M$ , respectively).

Since the  $N = L + M - 1$ -point DFT of the output sequence  $y(n)$  is sufficient to represent  $y(n)$  in the frequency domain, it follows that the multiplication of the  $N$ -point DFTs  $X(k)$  and  $H(k)$ , according to (5.3.3), followed by the computation of the  $N$ -point IDFT, must yield the sequence  $\{y(n)\}$ . In turn, this implies that the  $N$ -point circular convolution of  $x(n)$  with  $h(n)$  must be equivalent to the linear convolution of  $x(n)$  with  $h(n)$ . In other words, by increasing the length of the sequences  $x(n)$  and  $h(n)$  to  $N$  points (by appending zeros), and then circularly convolving the resulting sequences, we obtain the same result as would have been obtained with linear convolution. Thus with zero padding, the DFT can be used to perform linear filtering.

---

## FAST FOURIER TRANSFORM

In this section we represent several methods for computing dft efficiently. In view of the importance of the DFT in various digital signal processing applications such as linear filtering, correlation analysis and spectrum analysis, its efficient computation is a topic that has received considerably attention by many mathematicians, engineers and scientists. Basically the computation is done using the formula method.

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \quad 0 \leq k \leq N-1$$

where

$$W_N = e^{-j2\pi/N}$$

In general, the data sequence  $x(n)$  is also assumed to be complex valued.

Similarly, the IDFT becomes

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-nk} \quad 0 \leq n \leq N-1$$

We observe that for each value of  $k$ , direct computation of  $X(k)$  involves  $N$  complex multiplications ( $4N$  real multiplications) and  $N-1$  complex additions ( $4N-2$  real additions). Consequently, to compute all  $N$  values of the DFT requires  $N^2$  complex multiplications and  $N^2 - N$  complex additions.

### 6.1.1 Direct Computation of the DFT

For a complex-valued sequence  $x(n)$  of  $N$  points, the DFT may be expressed as

$$X_R(k) = \sum_{n=0}^{N-1} \left[ x_R(n) \cos \frac{2\pi kn}{N} + x_I(n) \sin \frac{2\pi kn}{N} \right] \quad (6.1.6)$$

$$X_I(k) = - \sum_{n=0}^{N-1} \left[ x_R(n) \sin \frac{2\pi kn}{N} - x_I(n) \cos \frac{2\pi kn}{N} \right] \quad (6.1.7)$$

The direct computation of (6.1.6) and (6.1.7) requires:

1.  $2N^2$  evaluations of trigonometric functions.
2.  $4N^2$  real multiplications.
3.  $4N(N-1)$  real additions.
4. A number of indexing and addressing operations.

These operations are typical of DFT computational algorithms. The operations in items 2 and 3 result in the DFT values  $X_R(k)$  and  $X_I(k)$ . The indexing and addressing operations are necessary to fetch the data  $x(n)$ ,  $0 \leq n \leq N-1$ , and the phase factors and to store the results. The variety of DFT algorithms optimize each of these computational processes in a different way.

### Divide-and-Conquer Approach to Computation of the DFT

The development of computationally efficient algorithms for the DFT is made possible if we adopt a divide-and-conquer approach. This approach is based on the decomposition of an  $N$ -point DFT into successively smaller DFT. This basic approach leads to a family of computationally efficient algorithms known collectively as FFT algorithms.

To illustrate the basic notions, let us consider the computation of an  $N$  point DFT, where  $N$  can be factored as a product of two integers, that is,  $N = LM$

### Algorithm 1

1. Store the signal column-wise.
2. Compute the  $M$ -point DFT of each row.
3. Multiply the resulting array by the phase factors  $W_N^{lq}$ .
4. Compute the  $L$ -point DFT of each column
5. Read the resulting array row-wise.

### Algorithm 2

1. Store the signal row-wise.
2. Compute the  $L$ -point DFT at each column.
3. Multiply the resulting array by the factors  $W_N^{pm}$ .
4. Compute the  $M$ -point DFT of each row.
5. Read the resulting array column-wise.

### 6.1.3 Radix-2 FFT Algorithms

Let us consider the computation of the  $N = 2^v$  point DFT by the divide-and-conquer approach specified by (6.1.16) through (6.1.18). We select  $M = N/2$  and  $L = 2$ . This selection results in a split of the  $N$ -point data sequence into two  $N/2$ -point data sequences  $f_1(n)$  and  $f_2(n)$ , corresponding to the even-numbered and odd-numbered samples of  $x(n)$ , respectively, that is,

$$\begin{aligned} f_1(n) &= x(2n) \\ f_2(n) &= x(2n + 1), \quad n = 0, 1, \dots, \frac{N}{2} - 1 \end{aligned} \quad (6.1.23)$$

Thus  $f_1(n)$  and  $f_2(n)$  are obtained by decimating  $x(n)$  by a factor of 2, and hence the resulting FFT algorithm is called a decimation-in-time algorithm.

Now the  $N$ -point DFT can be expressed in terms of the DFTs of the decimated sequences as follows:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \quad k = 0, 1, \dots, N - 1$$

$$\begin{aligned}
&= \sum_{n \text{ even}} x(n)W_N^{kn} + \sum_{n \text{ odd}} x(n)W_N^{kn} & (6.1.24) \\
&= \sum_{m=0}^{(N/2)-1} x(2m)W_N^{2mk} + \sum_{m=0}^{(N/2)-1} x(2m+1)W_N^{k(2m+1)}
\end{aligned}$$

But  $W_N^2 = W_{N/2}$ . With this substitution, (6.1.24) can be expressed as

$$\begin{aligned}
X(k) &= \sum_{m=0}^{(N/2)-1} f_1(m)W_{N/2}^{km} + W_N^k \sum_{m=0}^{(N/2)-1} f_2(m)W_{N/2}^{km} & (6.1.25) \\
&= F_1(k) + W_N^k F_2(k) \quad k = 0, 1, \dots, N-1
\end{aligned}$$

where  $F_1(k)$  and  $F_2(k)$  are the  $N/2$ -point DFTs of the sequences  $f_1(m)$  and  $f_2(m)$ , respectively.

Since  $F_1(k)$  and  $F_2(k)$  are periodic, with period  $N/2$ , we have  $F_1(k + N/2) = F_1(k)$  and  $F_2(k + N/2) = F_2(k)$ . In addition, the factor  $W_N^{k+N/2} = -W_N^k$ . Hence (6.1.25) can be expressed as

$$X(k) = F_1(k) + W_N^k F_2(k) \quad k = 0, 1, \dots, \frac{N}{2} - 1 \quad (6.1.26)$$

$$X\left(k + \frac{N}{2}\right) = F_1(k) - W_N^k F_2(k) \quad k = 0, 1, \dots, \frac{N}{2} - 1 \quad (6.1.27)$$

To be consistent with our previous notation, we may define

$$G_1(k) = F_1(k) \quad k = 0, 1, \dots, \frac{N}{2} - 1$$

$$G_2(k) = W_N^k F_2(k) \quad k = 0, 1, \dots, \frac{N}{2} - 1$$

Then the DFT  $X(k)$  may be expressed as

$$\begin{aligned}
X(k) &= G_1(k) + G_2(k) \quad k = 0, 1, \dots, \frac{N}{2} - 1 \\
X\left(k + \frac{N}{2}\right) &= G_1(k) - G_2(k) \quad k = 0, 1, \dots, \frac{N}{2} - 1 & (6.1.28)
\end{aligned}$$

$N/4$ -point sequences

$$\begin{aligned} v_{11}(n) &= f_1(2n) & n = 0, 1, \dots, \frac{N}{4} - 1 \\ v_{12}(n) &= f_1(2n + 1) & n = 0, 1, \dots, \frac{N}{4} - 1 \end{aligned} \quad (6.1.29)$$

and  $f_2(n)$  would yield

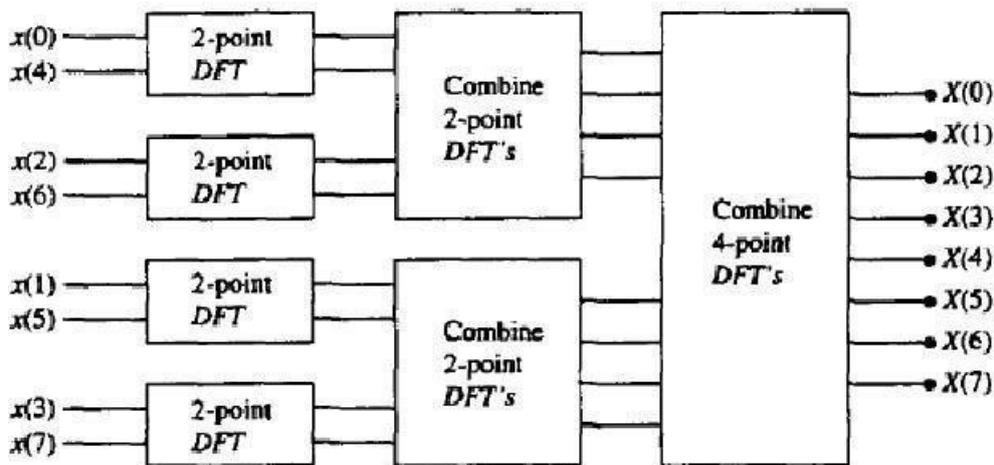
$$\begin{aligned} v_{21}(n) &= f_2(2n) & n = 0, 1, \dots, \frac{N}{4} - 1 \\ v_{22}(n) &= f_2(2n + 1) & n = 0, 1, \dots, \frac{N}{4} - 1 \end{aligned} \quad (6.1.30)$$

By computing  $N/4$ -point DFTs, we would obtain the  $N/2$ -point DFTs  $F_1(k)$  and  $F_2(k)$  from the relations

$$\begin{aligned} F_1(k) &= V_{11}(k) + W_{N/2}^k V_{12}(k) & k = 0, 1, \dots, \frac{N}{4} - 1 \\ F_1\left(k + \frac{N}{4}\right) &= V_{11}(k) - W_{N/2}^k V_{12}(k) & k = 0, 1, \dots, \frac{N}{4} - 1 \end{aligned} \quad (6.1.31)$$

$$\begin{aligned} F_2(k) &= V_{21}(k) + W_{N/2}^k V_{22}(k) & k = 0, 1, \dots, \frac{N}{4} - 1 \\ F_2\left(k + \frac{N}{4}\right) &= V_{21}(k) - W_{N/2}^k V_{22}(k) & k = 0, \dots, \frac{N}{4} - 1 \end{aligned} \quad (6.1.32)$$

where the  $\{V_{ij}(k)\}$  are the  $N/4$ -point DFTs of the sequences  $\{v_{ij}(n)\}$ .



**Figure 6.5** Three stages in the computation of an  $N = 8$ -point DFT.

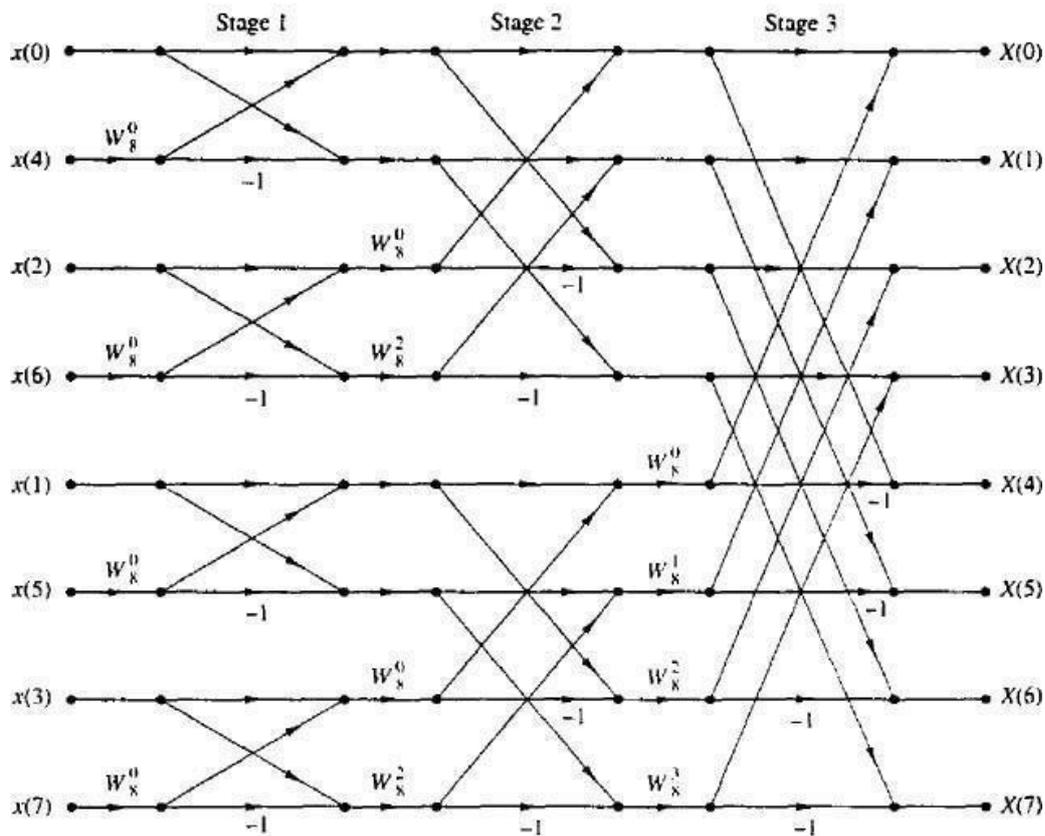


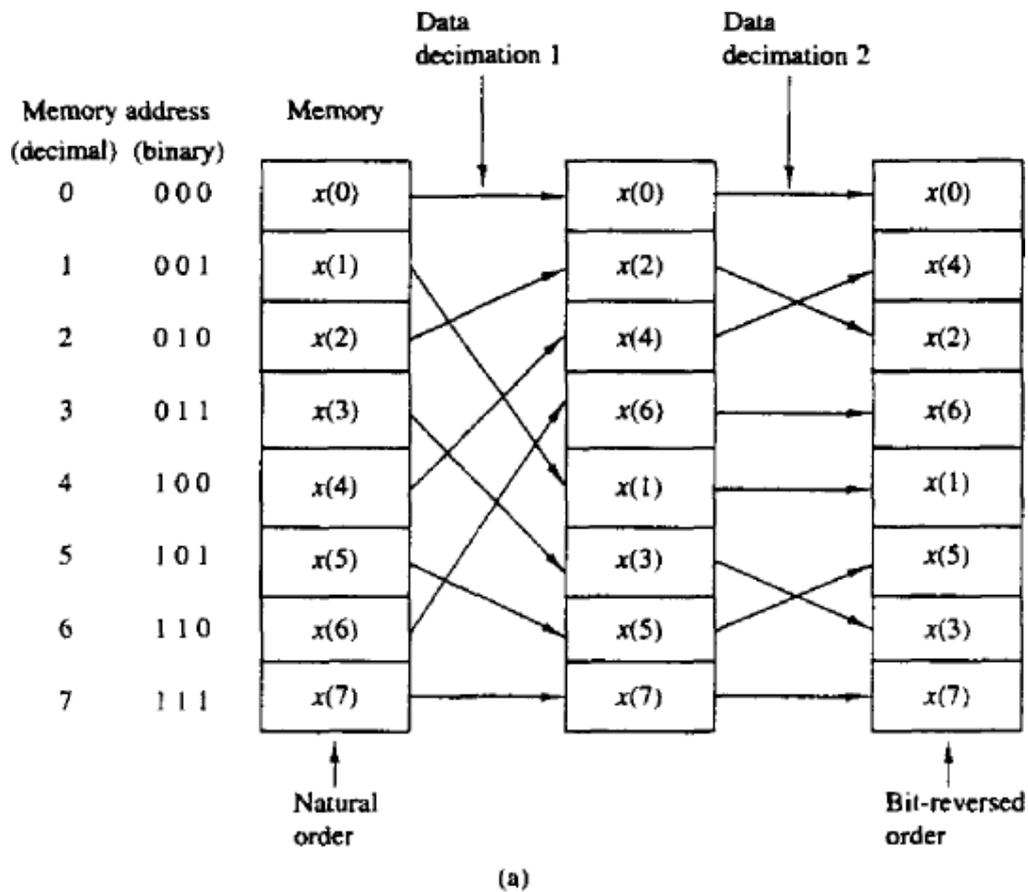
Figure 6.6 Eight-point decimation-in-time FFT algorithm.

Another important radix-2 FFT algorithm, called the decimation-in-frequency algorithm, is obtained by using the divide-and-conquer approach described in Section 6.1.2 with the choice of  $M = 2$  and  $L = N/2$ . This choice of parameters implies a column-wise storage of the input data sequence. To derive the algorithm, we begin by splitting the DFT formula into two summations, one of which involves the sum over the first  $N/2$  data points and the second sum involves the last  $N/2$  data points. Thus we obtain

$$\begin{aligned}
 X(k) &= \sum_{n=0}^{(N/2)-1} x(n)W_N^{kn} + \sum_{n=N/2}^{N-1} x(n)W_N^{kn} \\
 &= \sum_{n=0}^{(N/2)-1} x(n)W_N^{kn} + W_N^{Nk/2} \sum_{n=0}^{(N/2)-1} x\left(n + \frac{N}{2}\right)W_N^{kn}
 \end{aligned} \tag{6.1.33}$$

Since  $W_N^{kN/2} = (-1)^k$ , the expression (6.1.33) can be rewritten as

$$X(k) = \sum_{n=0}^{(N/2)-1} \left[ x(n) + (-1)^k x\left(n + \frac{N}{2}\right) \right] W_N^{kn} \tag{6.1.34}$$



Now, let us split (decimate)  $X(k)$  into the even- and odd-numbered samples. Thus we obtain

$$X(2k) = \sum_{n=0}^{(N/2)-1} \left[ x(n) + x\left(n + \frac{N}{2}\right) \right] W_{N/2}^{kn} \quad k = 0, 1, \dots, \frac{N}{2} - 1 \quad (6.1.35)$$

and

$$X(2k+1) = \sum_{n=0}^{(N/2)-1} \left\{ \left[ x(n) - x\left(n + \frac{N}{2}\right) \right] W_N^n \right\} W_{N/2}^{kn} \quad k = 0, 1, \dots, \frac{N}{2} - 1 \quad (6.1.36)$$

where we have used the fact that  $W_N^2 = W_{N/2}$ .

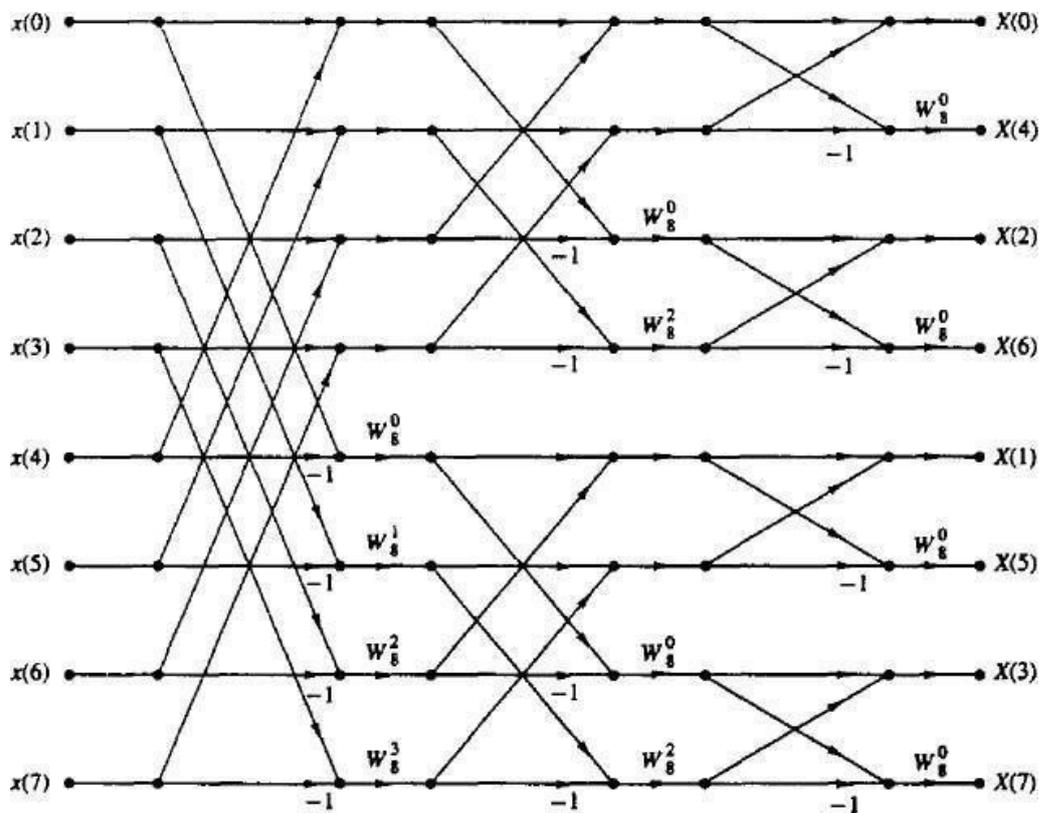
If we define the  $N/2$ -point sequences  $g_1(n)$  and  $g_2(n)$  as

$$\begin{aligned} g_1(n) &= x(n) + x\left(n + \frac{N}{2}\right) \\ g_2(n) &= \left[ x(n) - x\left(n + \frac{N}{2}\right) \right] W_N^n \quad n = 0, 1, 2, \dots, \frac{N}{2} - 1 \end{aligned} \quad (6.1.37)$$

then

$$\begin{aligned} X(2k) &= \sum_{n=0}^{(N/2)-1} g_1(n) W_{N/2}^{kn} \\ X(2k+1) &= \sum_{n=0}^{(N/2)-1} g_2(n) W_{N/2}^{kn} \end{aligned} \quad (6.1.38)$$

We observe from Fig. 6.11, that the input data  $x(n)$  occurs in natural order, but the output DFT occurs in bit-reversed order. We also note that the computations are performed in place. However, it is possible to reconfigure the decimation-in-frequency algorithm so that the input sequence occurs in bit-reversed order while the output DFT occurs in normal order. Furthermore, if we abandon the requirement that the computations be done in place, it is also possible to have both the input data and the output DFT in normal order.



### 6.1.4 Radix-4 FFT Algorithms

When the number of data points  $N$  in the DFT is a power of 4 (i.e.,  $N = 4^v$ ), we can, of course, always use a radix-2 algorithm for the computation. However, for this case, it is more efficient computationally to employ a radix-4 FFT algorithm.

Let us begin by describing a radix-4 decimation-in-time FFT algorithm, which is obtained by selecting  $L = 4$  and  $M = N/4$  in the divide-and-conquer approach described in Section 6.1.2. For this choice of  $L$  and  $M$ , we have  $l, p = 0, 1, 2, 3$ ;  $m, q = 0, 1, \dots, N/4 - 1$ ;  $n = 4m + l$ ; and  $k = (N/4)p + q$ . Thus we split or decimate the  $N$ -point input sequence into four subsequences,  $x(4n)$ ,  $x(4n + 1)$ ,  $x(4n + 2)$ ,  $x(4n + 3)$ ,  $n = 0, 1, \dots, N/4 - 1$ .

By applying (6.1.15) we obtain

$$X(p, q) = \sum_{l=0}^3 \left[ W_N^{lq} F(l, q) \right] W_4^{lp} \quad p = 0, 1, 2, 3 \quad (6.1.39)$$

where  $F(l, q)$  is given by (6.1.16), that is,

$$F(l, q) = \sum_{m=0}^{(N/4)-1} x(l, m) W_{N/4}^{mq} \quad \begin{matrix} l = 0, 1, 2, 3, \\ q = 0, 1, 2, \dots, \frac{N}{4} - 1 \end{matrix} \quad (6.1.40)$$

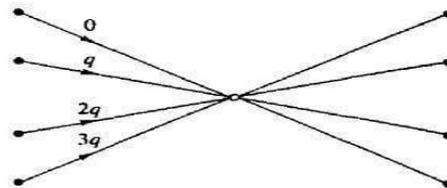
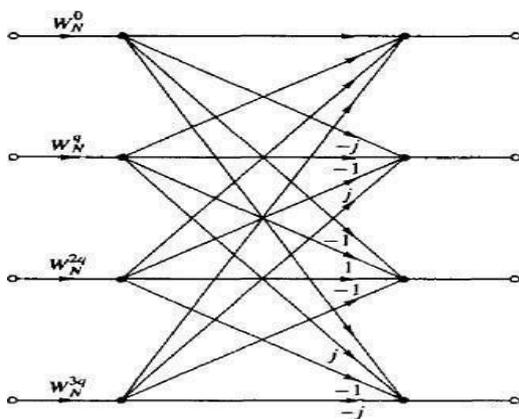
and

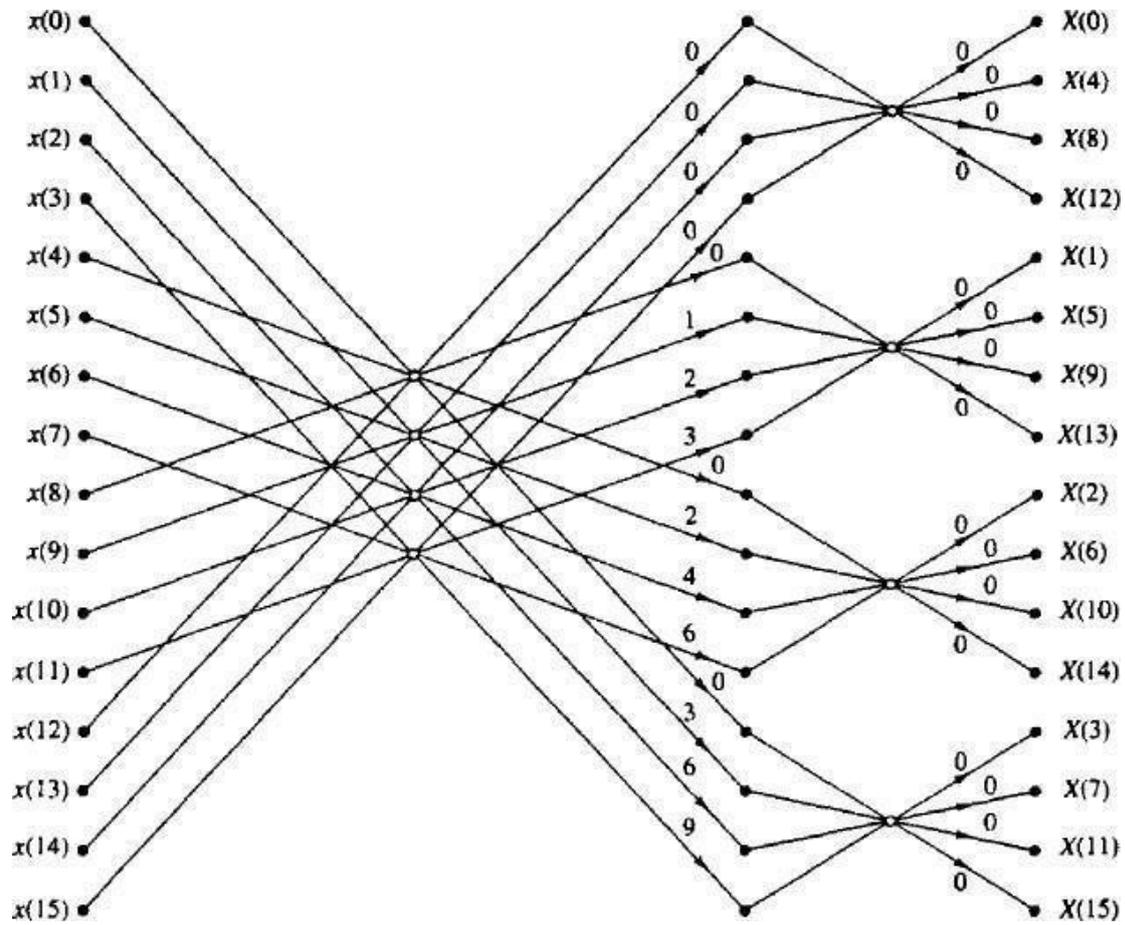
$$x(l, m) = x(4m + l) \quad (6.1.41)$$

$$X(p, q) = X\left(\frac{N}{4}p + q\right) \quad (6.1.42)$$

Thus, the four  $N/4$ -point DFTs obtained from (6.1.40) are combined according to (6.1.39) to yield the  $N$ -point DFT. The expression in (6.1.39) for combining the  $N/4$ -point DFTs defines a radix-4 decimation-in-time butterfly, which can be expressed in matrix form as

$$\begin{bmatrix} X(0, q) \\ X(1, q) \\ X(2, q) \\ X(3, q) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} W_N^0 F(0, q) \\ W_N^q F(1, q) \\ W_N^{2q} F(2, q) \\ W_N^{3q} F(3, q) \end{bmatrix} \quad (6.1.43)$$





$$X(4k) = \sum_{n=0}^{N/4-1} \left[ x(n) + x\left(n + \frac{N}{4}\right) + x\left(n + \frac{N}{2}\right) + x\left(n + \frac{3N}{4}\right) \right] W_N^0 W_{N/4}^{kn}$$

$$X(4k+1) = \sum_{n=0}^{N/4-1} \left[ x(n) - jx\left(n + \frac{N}{4}\right) - x\left(n + \frac{N}{2}\right) + jx\left(n + \frac{3N}{4}\right) \right] W_N^n W_{N/4}^{kn}$$

$$X(4k+2) = \sum_{n=0}^{N/4-1} \left[ x(n) - x\left(n + \frac{N}{4}\right) + x\left(n + \frac{N}{2}\right) - x\left(n + \frac{3N}{4}\right) \right] W_N^{2n} W_{N/4}^{kn}$$

$$X(4k+3) = \sum_{n=0}^{N/4-1} \left[ x(n) + jx\left(n + \frac{N}{4}\right) - x\left(n + \frac{N}{2}\right) - jx\left(n + \frac{3N}{4}\right) \right] W_N^{3n} W_{N/4}^{kn}$$

### 6.1.5 Split-Radix FFT Algorithms

An inspection of the radix-2 decimation-in-frequency flowgraph shown in Fig. 6.11 indicates that the even-numbered points of the DFT can be computed independently of the odd-numbered points. This suggests the possibility of using different computational methods for independent parts of the algorithm with the objective of reducing the number of computations. The split-radix FFT (SRFFT) algorithms exploit this idea by using both a radix-2 and a radix-4 decomposition in the same FFT algorithm.

We illustrate this approach with a decimation-in-frequency SRFFT algorithm due to Duhamel (1986). First, we recall that in the radix-2 decimation-in-frequency FFT algorithm, the even-numbered samples of the  $N$ -point DFT are given as

$$X(2k) = \sum_{n=0}^{N/2-1} \left[ x(n) + x\left(n + \frac{N}{2}\right) \right] W_{N/2}^{nk} \quad k = 0, 1, \dots, \frac{N}{2} - 1 \quad (6.1.55)$$

Note that these DFT points can be obtained from an  $N/2$ -point DFT without any additional multiplications. Consequently, a radix-2 suffices for this computation.

The odd-numbered samples  $\{X(2k+1)\}$  of the DFT require the premultiplication of the input sequence with the twiddle factors  $W_N^n$ . For these samples a radix-4 decomposition produces some computational efficiency because the four-point DFT has the largest multiplication-free butterfly. Indeed, it can be shown that using a radix greater than 4, does not result in a significant reduction in computational complexity.

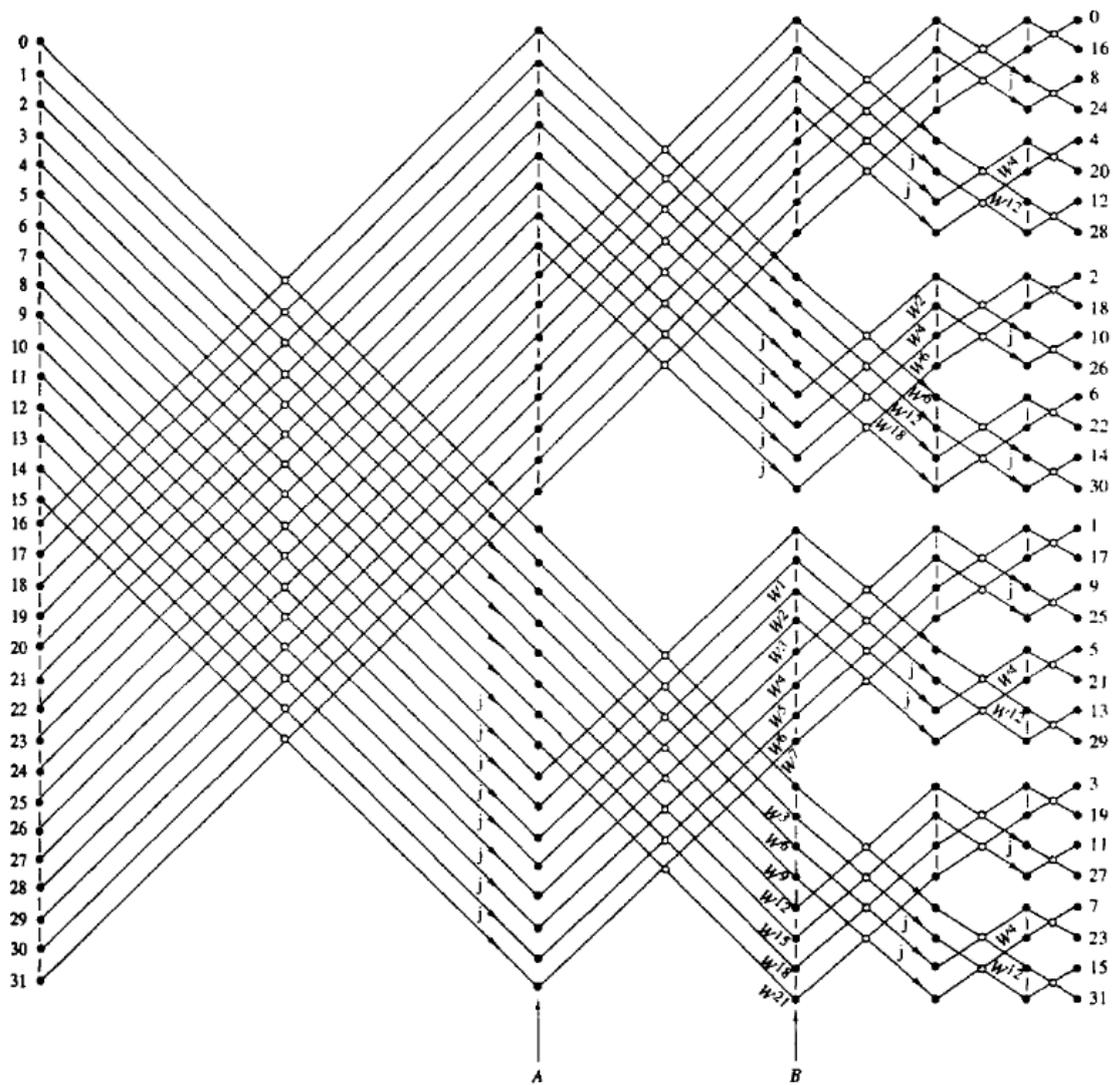
If we use a radix-4 decimation-in-frequency FFT algorithm for the odd-numbered samples of the  $N$ -point DFT, we obtain the following  $N/4$ -point DFTs:

$$X(4k+1) = \sum_{n=0}^{N/4-1} \{ [x(n) - x(n+N/2)] - j[x(n+N/4) - x(n+3N/4)] \} W_N^n W_{N/4}^{kn} \quad (6.1.56)$$

$$X(4k+3) = \sum_{n=0}^{N/4-1} \{ [x(n) - x(n+N/2)] + j[x(n+N/4) - x(n+3N/4)] \} W_N^{3n} W_{N/4}^{kn} \quad (6.1.57)$$

Thus the  $N$ -point DFT is decomposed into one  $N/2$ -point DFT without additional twiddle factors and two  $N/4$ -point DFTs with twiddle factors. The  $N$ -point DFT is obtained by successive use of these decompositions up to the last stage. Thus we obtain a decimation-in-frequency SRFFT algorithm.

Figure 6.15 shows the flow graph for an in-place 32-point decimation-in-frequency SRFFT algorithm. At stage A of the computation for  $N = 32$ , the



An additional factor of 2 savings in storage of twiddle factors can be obtained by introducing a  $90^\circ$  phase offset at the mid point of each twiddle array, which can be removed if necessary at the output of the SRFFT computation. The incorporation of this improvement into the SRFFT results in another algorithm also due to price called the PFFT algorithm.

### Implementation of FFT Algorithms

Now that we have described the basic radix-2 and radix-4 FFT algorithms, let us consider some of the implementation issues. Our remarks apply directly to

radix-2 algorithms, although similar comments may be made about radix-4 and higher-radix algorithms.

Basically, the radix-2 FFT algorithm consists of taking two data points at a time from memory, performing the butterfly computations and returning the resulting numbers to memory. This procedure is repeated many times  $((N \log_2 N)/2$  times) in the computation of an  $N$ -point DFT.

The butterfly computations require the twiddle factors  $\{W_N^k\}$  at various stages in either natural or bit-reversed order. In an efficient implementation of the algorithm, the phase factors are computed once and stored in a table, either in normal order or in bit-reversed order, depending on the specific implementation of the algorithm.

Memory requirement is another factor that must be considered. If the computations are performed in place, the number of memory locations required is  $2N$  since the numbers are complex. However, we can instead double the memory to  $4N$ , thus simplifying the indexing and control operations in the FFT algorithms. In this case we simply alternate in the use of the two sets of memory locations from one stage of the FFT algorithm to the other. Doubling of the memory also allows us to have both the input sequence and the output sequence in normal order.

Finally, we note that the emphasis in our discussion of FFT algorithms was on radix-2, radix-4, and split-radix algorithms. These are by far the most widely used in practice. When the number of data points is not a power of 2 or 4, it is a simple matter to pad the sequence  $x(n)$  with zeros such that  $N = 2^n$  or  $N = 4^n$ .

The measure of complexity for FFT algorithms that we have emphasized is the required number of arithmetic operations (multiplications and additions). Although this is a very important benchmark for computational complexity, there are other issues to be considered in practical implementation of FFT algorithms. These include the architecture of the processor, the available instruction set, the data structures for storing twiddle factors, and other considerations.

For general-purpose computers, where the cost of the numerical operations dominate, radix-2, radix-4, and split-radix FFT algorithms are good candidates. However, in the case of special-purpose digital signal processors, featuring single-cycle multiply-and-accumulate operation, bit-reversed addressing, and a high degree of instruction parallelism, the structural regularity of the algorithm is equally important as arithmetic complexity. Hence for DSP processors, radix-2 or radix-4 decimation-in-frequency FFT algorithms are preferable in terms of speed and accuracy. The irregular structure of the SRFFT may render it less suitable for implementation on digital signal processors. Structural regularity is also important in the implementation of FFT algorithms on vector processors, multiprocessors, and in VLSI. Interprocessor communication is an important consideration in such implementations on parallel processors.

In conclusion, we have presented several important considerations in the implementation of FFT algorithms. Advances in digital signal processing technology, in hardware and software, will continue to influence the choice among FFT algorithms for various practical applications.

## APPLICATIONS OF FFT ALGORITHMS

The FFT algorithms described in the preceding section find application in a variety of areas, including linear filtering, correlation, and spectrum analysis. Basically, the FFT algorithm is used as an efficient means to compute the DFT and the IDFT.

In this section we consider the use of the FFT algorithm in linear filtering and in the computation of the crosscorrelation of two sequences. The use of the FFT in spectrum analysis is considered in Chapter 12. In addition we illustrate how to enhance the efficiency of the FFT algorithm by forming complex-valued sequences from real-valued sequences prior to the computation of the DFT.

### 6.2.1 Efficient Computation of the DFT of Two Real Sequences

The FFT algorithm is designed to perform complex multiplications and additions, even though the input data may be real valued. The basic reason for this situation is

Suppose that  $x_1(n)$  and  $x_2(n)$  are two real-valued sequences of length  $N$ , and let  $x(n)$  be a complex-valued sequence defined as

$$x(n) = x_1(n) + jx_2(n) \quad 0 \leq n \leq N - 1 \quad (6.2.1)$$

The DFT operation is linear and hence the DFT of  $x(n)$  can be expressed as

$$X(k) = X_1(k) + jX_2(k) \quad (6.2.2)$$

The sequences  $x_1(n)$  and  $x_2(n)$  can be expressed in terms of  $x(n)$  as follows:

$$x_1(n) = \frac{x(n) + x^*(n)}{2} \quad (6.2.3)$$

$$x_2(n) = \frac{x(n) - x^*(n)}{2j} \quad (6.2.4)$$

Hence the DFTs of  $x_1(n)$  and  $x_2(n)$  are

$$X_1(k) = \frac{1}{2}\{DFT[x(n)] + DFT[x^*(n)]\} \quad (6.2.5)$$

$$X_2(k) = \frac{1}{2j}\{DFT[x(n)] - DFT[x^*(n)]\} \quad (6.2.6)$$

Recall that the DFT of  $x^*(n)$  is  $X^*(N - k)$ . Therefore,

$$X_1(k) = \frac{1}{2}[X(k) + X^*(N - k)] \quad (6.2.7)$$

$$X_2(k) = \frac{1}{j2}[X(k) - X^*(N - k)] \quad (6.2.8)$$

### 6.2.2 Efficient Computation of the DFT of a $2N$ -Point Real Sequence

Suppose that  $g(n)$  is a real-valued sequence of  $2N$  points. We now demonstrate how to obtain the  $2N$ -point DFT of  $g(n)$  from computation of one  $N$ -point DFT involving complex-valued data. First, we define

$$\begin{aligned} x_1(n) &= g(2n) \\ x_2(n) &= g(2n + 1) \end{aligned} \quad (6.2.9)$$

Let  $x(n)$  be the  $N$ -point complex-valued sequence

$$x(n) = x_1(n) + jx_2(n) \quad (6.2.10)$$

From the results of the preceding section, we have

$$\begin{aligned} X_1(k) &= \frac{1}{2}[X(k) + X^*(N-k)] \\ X_2(k) &= \frac{1}{2j}[X(k) - X^*(N-k)] \end{aligned} \quad (6.2.11)$$

Finally, we must express the  $2N$ -point DFT in terms of the two  $N$ -point DFTs,  $X_1(k)$  and  $X_2(k)$ . To accomplish this, we proceed as in the decimation-in-time FFT algorithm, namely,

$$\begin{aligned} G(k) &= \sum_{n=0}^{N-1} g(2n)W_{2N}^{2nk} + \sum_{n=0}^{N-1} g(2n+1)W_{2N}^{(2n+1)k} \\ &= \sum_{n=0}^{N-1} x_1(n)W_N^{nk} + W_{2N}^k \sum_{n=0}^{N-1} x_2(n)W_N^{nk} \end{aligned}$$

Consequently,

$$\begin{aligned} G(k) &= X_1(k) + W_{2N}^k X_2(k) \quad k = 0, 1, \dots, N-1 \\ G(k+N) &= X_1(k) - W_{2N}^k X_2(k) \quad k = 0, 1, \dots, N-1 \end{aligned} \quad (6.2.12)$$

Thus we have computed the DFT of a  $2N$ -point real sequence from one  $N$ -point DFT and some additional computation as indicated by (6.2.11) and (6.2.12).

### 6.2.3 Use of the FFT Algorithm in Linear Filtering and Correlation

An important application of the FFT algorithm is in FIR linear filtering of long data sequences. In Chapter 5 we described two methods, the overlap-add and the overlap-save methods for filtering a long data sequence with an FIR filter, based on the use of the DFT. In this section we consider the use of these two methods in conjunction with the FFT algorithm for computing the DFT and the IDFT.

Let  $h(n)$ ,  $0 \leq n \leq M-1$ , be the unit sample response of the FIR filter and let  $x(n)$  denote the input data sequence. The block size of the FFT algorithm is  $N$ , where  $N = L + M - 1$  and  $L$  is the number of new data samples being processed by the filter. We assume that for any given value of  $M$ , the number  $L$  of data samples is selected so that  $N$  is a power of 2. For purposes of this discussion, we consider only radix-2 FFT algorithms.

The  $N$ -point DFT of  $h(n)$ , which is padded by  $L-1$  zeros, is denoted as  $H(k)$ . This computation is performed once via the FFT and the resulting  $N$  complex numbers are stored. To be specific we assume that the decimation-in-frequency

FFT algorithm is used to compute  $H(k)$ . This yields  $H(k)$  in bit-reversed order, which is the way it is stored in memory.

In the overlap-save method, the first  $M - 1$  data points of each data block are the last  $M - 1$  data points of the previous data block. Each data block contains  $L$  new data points, such that  $N = L + M - 1$ . The  $N$ -point DFT of each data block is performed by the FFT algorithm. If the decimation-in-frequency algorithm is employed, the input data block requires no shuffling and the values of the DFT occur in bit-reversed order. Since this is exactly the order of  $H(k)$ , we can multiply the DFT of the data, say  $X_m(k)$ , with  $H(k)$  and thus the result

$$Y_m(k) = H(k)X_m(k)$$

is also in bit-reversed order.

The inverse DFT (IDFT) can be computed by use of an FFT algorithm that takes the input in bit-reversed order and produces an output in normal order. Thus there is no need to shuffle any block of data either in computing the DFT or the IDFT.

If the overlap-add method is used to perform the linear filtering, the computational method using the FFT algorithm is basically the same. The only difference is that the  $N$ -point data blocks consist of  $L$  new data points and  $M - 1$  additional zeros. After the IDFT is computed for each data block, the  $N$ -point filtered blocks are overlapped as indicated in Section 5.3.2, and the  $M - 1$  overlapping data points between successive output records are added together.

### 6.3.1 The Goertzel Algorithm

The Goertzel algorithm exploits the periodicity of the phase factors  $\{W_N^k\}$  and allows us to express the computation of the DFT as a linear filtering operation. Since  $W_N^{-kN} = 1$ , we can multiply the DFT by this factor. Thus

$$X(k) = W_N^{-kN} \sum_{m=0}^{N-1} x(m)W_N^{km} = \sum_{m=0}^{N-1} x(m)W_N^{-k(N-m)} \quad (6.3.1)$$

We note that (6.3.1) is in the form of a convolution. Indeed, if we define the sequence  $y_k(n)$  as

$$y_k(n) = \sum_{m=0}^{N-1} x(m)W_N^{-k(n-m)} \quad (6.3.2)$$

then it is clear that  $y_k(n)$  is the convolution of the finite-duration input sequence  $x(n)$  of length  $N$  with a filter that has an impulse response

$$h_k(n) = W_N^{-kn}u(n) \quad (6.3.3)$$

The output of this filter at  $n = N$  yields the value of the DFT at the frequency  $\omega_k = 2\pi k/N$ . That is,

$$X(k) = y_k(n)|_{n=N} \quad (6.3.4)$$

as can be verified by comparing (6.3.1) with (6.3.2).

The filter with impulse response  $h_k(n)$  has the system function

$$H_k(z) = \frac{1}{1 - W_N^{-k}z^{-1}} \quad (6.3.5)$$

This filter has a pole on the unit circle at the frequency  $\omega_k = 2\pi k/N$ . Thus, the entire DFT can be computed by passing the block of input data into a parallel bank of  $N$  single-pole filters (resonators), where each filter has a pole at the corresponding frequency of the DFT.

Instead of performing the computation of the DFT as in (6.3.2), via convolution, we can use the difference equation corresponding to the filter given by (6.3.5) to compute  $y_k(n)$  recursively. Thus we have

$$y_k(n) = W_N^{-k} y_k(n-1) + x(n) \quad y_k(-1) = 0 \quad (6.3.6)$$

The desired output is  $X(k) = y_k(N)$ , for  $k = 0, 1, \dots, N-1$ . To perform this computation, we can compute once and store the phase factors  $W_N^{-k}$ .

The complex multiplications and additions inherent in (6.3.6) can be avoided by combining the pairs of resonators possessing complex-conjugate poles. This leads to two-pole filters with system functions of the form

$$H_k(z) = \frac{1 - W_N^k z^{-1}}{1 - 2 \cos(2\pi k/N) z^{-1} + z^{-2}} \quad (6.3.7)$$

### 6.3.2 The Chirp-z Transform Algorithm

The DFT of an  $N$ -point data sequence  $x(n)$  has been viewed as the  $z$ -transform of  $x(n)$  evaluated at  $N$  equally spaced points on the unit circle. It has also been viewed as  $N$  equally spaced samples of the Fourier transform of the data sequence  $x(n)$ . In this section we consider the evaluation of  $X(z)$  on other contours in the  $z$ -plane, including the unit circle.

Suppose that we wish to compute the values of the  $z$ -transform of  $x(n)$  at a set of points  $\{z_k\}$ . Then,

$$X(z_k) = \sum_{n=0}^{N-1} x(n) z_k^{-n} \quad k = 0, 1, \dots, L-1 \quad (6.3.10)$$

For example, if the contour is a circle of radius  $r$  and the  $z_k$  are  $N$  equally spaced points, then

$$z_k = r e^{j2\pi kn/N} \quad k = 0, 1, 2, \dots, N-1$$

$$X(z_k) = \sum_{n=0}^{N-1} [x(n) r^{-n}] e^{-j2\pi kn/N} \quad k = 0, 1, 2, \dots, N-1 \quad (6.3.11)$$

In this case the FFT algorithm can be applied on the modified sequence  $x(n)r^{-n}$ .

More generally, suppose that the points  $z_k$  in the  $z$ -plane fall on an arc which begins at some point

$$z_0 = r_0 e^{j\theta_0}$$

and spirals either in toward the origin or out away from the origin such that the points  $\{z_k\}$  are defined as

$$z_k = r_0 e^{j\theta_0} (R_0 e^{j\phi_0})^k \quad k = 0, 1, \dots, L-1 \quad (6.3.12)$$

When points  $\{z_k\}$  in (6.3.12) are substituted into the expression for the  $z$ -transform, we obtain

$$X(z_k) = \sum_{n=0}^{N-1} x(n) z_k^{-n}$$

$$= \sum_{n=0}^{N-1} x(n) (r_0 e^{j\theta_0})^{-n} V^{-nk} \quad (6.3.13)$$

where, by definition,

$$V = R_0 e^{j\phi_0} \quad (6.3.14)$$

We can express (6.3.13) in the form of a convolution, by noting that

$$nk = \frac{1}{2}[n^2 + k^2 - (k - n)^2] \quad (6.3.15)$$

Substitution of (6.3.15) into (6.3.13) yields

$$X(z_k) = V^{-k^2/2} \sum_{n=0}^{N-1} [x(n)(r_0 e^{j\phi_0})^{-n} V^{-n^2/2}] V^{(k-n)^2/2} \quad (6.3.16)$$

Let us define a new sequence  $g(n)$  as

$$g(n) = x(n)(r_0 e^{j\phi_0})^{-n} V^{-n^2/2} \quad (6.3.17)$$

Then (6.3.16) can be expressed as

$$X(z_k) = V^{-k^2/2} \sum_{n=0}^{N-1} g(n) V^{(k-n)^2/2} \quad (6.3.18)$$

The summation in (6.3.18) can be interpreted as the convolution of the sequence  $g(n)$  with the impulse response  $h(n)$  of a filter, where

$$h(n) = V^{n^2/2} \quad (6.3.19)$$

Consequently, (6.3.18) may be expressed as

$$\begin{aligned} X(z_k) &= V^{-k^2/2} y(k) \\ &= \frac{y(k)}{h(k)} \quad k = 0, 1, \dots, L-1 \end{aligned} \quad (6.3.20)$$

where  $y(k)$  is the output of the filter

$$y(k) = \sum_{n=0}^{N-1} g(n)h(k-n) \quad k = 0, 1, \dots, L-1 \quad (6.3.21)$$

## QUANTIZATION EFFECTS IN THE COMPUTATION OF THE DFT\*

As we have observed in our previous discussions, the DFT plays an important role in many digital signal processing applications, including FIR filtering, the computation of the correlation between signals, and spectral analysis. For this reason it is important for us to know the effect of quantization errors in its computation. In particular, we shall consider the effect of round-off errors due to the multiplications performed in the DFT with fixed-point arithmetic.

### 6.4.1 Quantization Errors in the Direct Computation of the DFT

Given a finite-duration sequence  $\{x(n)\}$ ,  $0 \leq n \leq N - 1$ , the DFT of  $\{x(n)\}$  is defined as

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn} \quad k = 0, 1, \dots, N - 1 \quad (6.4.1)$$

where  $W_N = e^{-j2\pi/N}$ . We assume that in general,  $\{x(n)\}$  is a complex-valued sequence. We also assume that the real and imaginary components of  $\{x(n)\}$  and  $\{W_N^{kn}\}$  are represented by  $b$  bits. Consequently, the computation of the product  $x(n)W_N^{kn}$  requires four real multiplications. Each real multiplication is rounded from  $2b$  bits to  $b$  bits, and hence there are four quantization errors for each complex-valued multiplication.

In the direct computation of the DFT, there are  $N$  complex-valued multiplications for each point in the DFT. Therefore, the total number of real multiplications in the computation of a single point in the DFT is  $4N$ . Consequently, there are  $4N$  quantization errors.

Let us evaluate the variance of the quantization errors in a fixed-point computation of the DFT. First, we make the following assumptions about the statistical properties of the quantization errors.

1. The quantization errors due to rounding are uniformly distributed random variables in the range  $(-\Delta/2, \Delta/2)$  where  $\Delta = 2^{-b}$ .
2. The  $4N$  quantization errors are mutually uncorrelated.
3. The  $4N$  quantization errors are uncorrelated with the sequence  $\{x(n)\}$ .

Since each of the quantization errors has a variance

$$\sigma_e^2 = \frac{\Delta^2}{12} = \frac{2^{-2b}}{12} \quad (6.4.2)$$

the variance of the quantization errors from the  $4N$  multiplications is

$$\begin{aligned} \sigma_q^2 &= 4N\sigma_e^2 \\ &= \frac{N}{3} \cdot 2^{-2b} \end{aligned} \quad (6.4.3)$$

as

$$\sigma_q^2 = \frac{2^{-2(b-v/2)}}{3} \quad (6.4.4)$$

This expression implies that every fourfold increase in the size  $N$  of the DFT requires an additional bit in computational precision to offset the additional quantization errors.

To prevent overflow, the input sequence to the DFT requires scaling. Clearly, an upper bound on  $|X(k)|$  is

$$|X(k)| \leq \sum_{n=0}^{N-1} |x(n)| \quad (6.4.5)$$

If the dynamic range in addition is  $(-1, 1)$ , then  $|X(k)| < 1$  requires that

$$\sum_{n=0}^{N-1} |x(n)| < 1 \quad (6.4.6)$$

If  $|x(n)|$  is initially scaled such that  $|x(n)| < 1$  for all  $n$ , then each point in the sequence can be divided by  $N$  to ensure that (6.4.6) is satisfied.

The scaling implied by (6.4.6) is extremely severe. For example, suppose that the signal sequence  $\{x(n)\}$  is white and, after scaling, each value  $|x(n)|$  of the sequence is uniformly distributed in the range  $(-1/N, 1/N)$ . Then the variance of the signal sequence is

$$\sigma_x^2 = \frac{(2/N)^2}{12} = \frac{1}{3N^2} \quad (6.4.7)$$

and the variance of the output DFT coefficients  $|X(k)|$  is

$$\begin{aligned} \sigma_X^2 &= N\sigma_x^2 \\ &= \frac{1}{3N} \end{aligned} \quad (6.4.8)$$

Thus the signal-to-noise power ratio is

$$\frac{\sigma_X^2}{\sigma_q^2} = \frac{2^{2b}}{N^2} \quad (6.4.9)$$

We observe that the scaling is responsible for reducing the SNR by  $N$  and the combination of scaling and quantization errors result in a total reduction that is proportional to  $N^2$ . Hence scaling the input sequence  $\{x(n)\}$  to satisfy (6.4.6) imposes a severe penalty on the signal-to-noise ratio in the DFT.

---

## UNIT -III IIR Digital Filters

### IIR FILTER DESIGN

#### DESIGN OF IIR FILTERS FROM ANALOG FILTERS

Just as in the design of FIR filters, there are several methods that can be used to design digital filters having an infinite-duration unit sample response. The techniques described in this section are all based on converting an analog filter into a digital filter. Analog filter design is a mature and well developed field, so it is not surprising that we begin the design of a digital filter in the analog domain and then convert the design into the digital domain.

An analog filter can be described by its system function.

$$H_a(s) = \frac{B(s)}{A(s)} = \frac{\sum_{k=0}^M \beta_k s^k}{\sum_{k=0}^N \alpha_k s^k} \quad (8.3.1)$$

where  $\{\alpha_k\}$  and  $\{\beta_k\}$  are the filter coefficients, or by its impulse response, which is related to  $H_a(s)$  by the Laplace transform

$$H_a(s) = \int_{-\infty}^{\infty} h(t)e^{-st} dt \quad (8.3.2)$$

Alternatively, the analog filter having the rational system function  $H(s)$  given in (8.3.1), can be described by the linear constant-coefficient differential equation

$$\sum_{k=0}^N \alpha_k \frac{d^k y(t)}{dt^k} = \sum_{k=0}^M \beta_k \frac{d^k x(t)}{dt^k} \quad (8.3.3)$$

where  $x(t)$  denotes the input signal and  $y(t)$  denotes the output of the filter.

Each of these three equivalent characterizations of an analog filter leads to alternative methods for converting the filter into the digital domain, as will be described in Sections 8.3.1 through 8.3.4. We recall that an analog linear time-invariant system with system function  $H(s)$  is stable if all its poles lie in the left half of the  $s$ -plane. Consequently, if the conversion technique is to be effective, it should possess the following desirable properties:

1. The  $j\Omega$  axis in the  $s$ -plane should map into the unit circle in the  $z$ -plane. Thus there will be a direct relationship between the two frequency variables in the two domains.

2. The left-half plane (LHP) of the  $s$ -plane should map into the inside of the unit circle in the  $z$ -plane. Thus a stable analog filter will be converted to a stable digital filter.

We mentioned in the preceding section that physically realizable and stable IIR filters cannot have linear phase. Recall that a linear-phase filter must have a system function that satisfies the condition

$$H(z) = \pm z^{-N} H(z^{-1}) \quad (8.3.4)$$

where  $z^{-N}$  represents a delay of  $N$  units of time. But if this were the case, the filter would have a mirror-image pole outside the unit circle for every pole inside the unit circle. Hence the filter would be unstable. Consequently, a causal and stable IIR filter cannot have linear phase.

If the restriction on physical realizability is removed, it is possible to obtain a linear-phase IIR filter, at least in principle. This approach involves performing a time reversal of the input signal  $x(n)$ , passing  $x(-n)$  through a digital filter  $H(z)$ , time-reversing the output of  $H(z)$ , and finally, passing the result through  $H(z)$  again. This signal processing is computationally cumbersome and appears to offer no advantages over linear-phase FIR filters. Consequently, when an application requires a linear-phase filter, it should be an FIR filter.

In the design of IIR filters, we shall specify the desired filter characteristics for the magnitude response only. This does not mean that we consider the phase response unimportant. Since the magnitude and phase characteristics are related, as indicated in Section 8.1, we specify the desired magnitude characteristics and accept the phase response that is obtained from the design methodology.

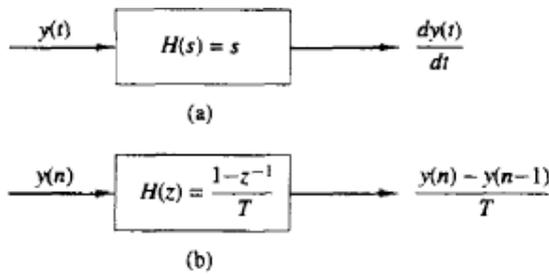
### 8.3.1 IIR Filter Design by Approximation of Derivatives

One of the simplest methods for converting an analog filter into a digital filter is to approximate the differential equation in (8.3.3) by an equivalent difference equation. This approach is often used to solve a linear constant-coefficient differential equation numerically on a digital computer.

For the derivative  $dy(t)/dt$  at time  $t = nT$ , we substitute the *backward difference*  $[y(nT) - y(nT - T)]/T$ . Thus

$$\begin{aligned} \left. \frac{dy(t)}{dt} \right|_{t=nT} &= \frac{y(nT) - y(nT - T)}{T} \\ &= \frac{y(n) - y(n-1)}{T} \end{aligned} \quad (8.3.5)$$

where  $T$  represents the sampling interval and  $y(n) \equiv y(nT)$ . The analog differentiator with output  $dy(t)/dt$  has the system function  $H(s) = s$ , while the digital system that produces the output  $[y(n) - y(n-1)]/T$  has the system function  $H(z) = (1 - z^{-1})/T$ . Consequently, as shown in Fig. 8.29, the frequency-domain



**Figure 8.29** Substitution of the backward difference for the derivative implies the mapping  $s = (1 - z^{-1})/T$ .

equivalent for the relationship in (8.3.5) is

$$s = \frac{1 - z^{-1}}{T} \quad (8.3.6)$$

The second derivative  $d^2y(t)/dt^2$  is replaced by the second difference, which is derived as follows:

$$\begin{aligned} \left. \frac{d^2y(t)}{dt^2} \right|_{t=nT} &= \left. \frac{d}{dt} \left[ \frac{dy(t)}{dt} \right] \right|_{t=nT} \\ &= \frac{[y(nT) - y(nT - T)]/T - [y(nT - T) - y(nT - 2T)]/T}{T} \\ &= \frac{y(n) - 2y(n-1) + y(n-2)}{T^2} \end{aligned} \quad (8.3.7)$$

In the frequency domain, (8.3.7) is equivalent to

$$s^2 = \frac{1 - 2z^{-1} + z^{-2}}{T^2} = \left( \frac{1 - z^{-1}}{T} \right)^2 \quad (8.3.8)$$

It easily follows from the discussion that the substitution for the  $k$ th derivative of  $y(t)$  results in the equivalent frequency-domain relationship

$$s^k = \left( \frac{1 - z^{-1}}{T} \right)^k \quad (8.3.9)$$

Consequently, the system function for the digital IIR filter obtained as a result of the approximation of the derivatives by finite differences is

$$H(z) = H_a(s)|_{s=(1-z^{-1})/T} \quad (8.3.10)$$

where  $H_a(s)$  is the system function of the analog filter characterized by the differential equation given in (8.3.3).

Let us investigate the implications of the mapping from the  $s$ -plane to the  $z$ -plane as given by (8.3.6) or, equivalently,

$$z = \frac{1}{1 - sT} \quad (8.3.11)$$

If we substitute  $s = j\Omega$  in (8.2.11), we find that

$$z = \frac{1}{1 - j\Omega T}$$

$$= \frac{1}{1 + \Omega^2 T^2} + j \frac{\Omega T}{1 + \Omega^2 T^2} \quad (8.3.12)$$

As  $\Omega$  varies from  $-\infty$  to  $\infty$ , the corresponding locus of points in the  $z$ -plane is a circle of radius  $\frac{1}{2}$  and with center at  $z = \frac{1}{2}$ , as illustrated in Fig. 8.30.

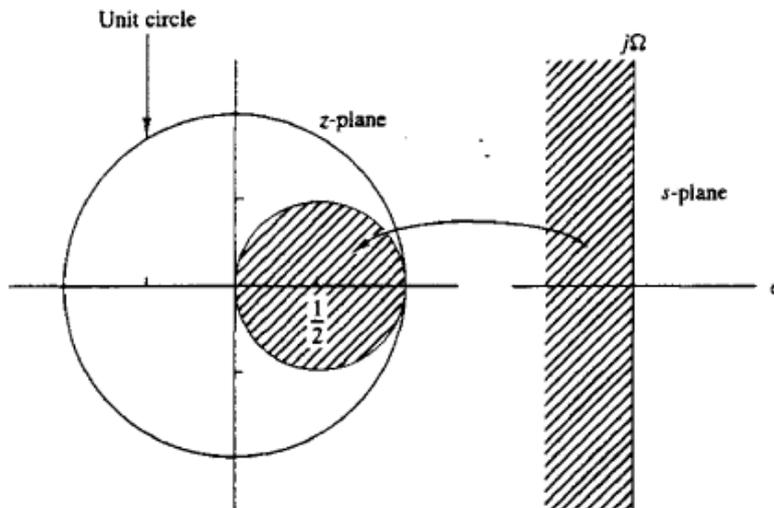
It is easily demonstrated that the mapping in (8.3.11) takes points in the LHP of the  $s$ -plane into corresponding points inside this circle in the  $z$ -plane and points in the RHP of the  $s$ -plane are mapped into points outside this circle. Consequently, this mapping has the desirable property that a stable analog filter is transformed into a stable digital filter. However, the possible location of the poles of the digital filter are confined to relatively small frequencies and, as a consequence, the mapping is restricted to the design of lowpass filters and bandpass filters having relatively small resonant frequencies. It is not possible, for example, to transform a highpass analog filter into a corresponding highpass digital filter.

In an attempt to overcome the limitations in the mapping given above, more complex substitutions for the derivatives have been proposed. In particular, an  $L$ th-order difference of the form

$$\left. \frac{dy(t)}{dt} \right|_{t=nT} = \frac{1}{T} \sum_{k=1}^L \alpha_k \frac{y(nT + kT) - y(nT - kT)}{T} \quad (8.3.13)$$

has been proposed, where  $\{\alpha_k\}$  are a set of parameters that can be selected to optimize the approximation. The resulting mapping between the  $s$ -plane and the  $z$ -plane is now

$$s = \frac{1}{T} \sum_{k=1}^L \alpha_k (z^k - z^{-k}) \quad (8.3.14)$$



**Figure 8.30** The mapping  $s = (1 - z^{-1})/T$  takes LHP in the  $s$ -plane into points inside the circle of radius  $\frac{1}{2}$  and center  $z = \frac{1}{2}$  in the  $z$ -plane.

When  $z = e^{j\omega}$ , we have

$$s = j \frac{2}{T} \sum_{k=1}^L \alpha_k \sin \omega k \quad (8.3.15)$$

which is purely imaginary. Thus

$$\Omega = \frac{2}{T} \sum_{k=1}^L \alpha_k \sin \omega k \quad (8.3.16)$$

is the resulting mapping between the two frequency variables. By proper choice of the coefficients  $\{\alpha_k\}$  it is possible to map the  $j\Omega$ -axis into the unit circle. Furthermore, points in the LHP in  $s$  can be mapped into points inside the unit circle in  $z$ .

Despite achieving the two desirable characteristics with the mapping of (8.3.16), the problem of selecting the set of coefficients  $\{\alpha_k\}$  remains. In general, this is a difficult problem. Since simpler techniques exist for converting analog filters into IIR digital filters, we shall not emphasize the use of the  $L$ th-order difference as a substitute for the derivative.

### 8.3.2 IIR Filter Design by Impulse Invariance

In the impulse invariance method, our objective is to design an IIR filter having a unit sample response  $h(n)$  that is the sampled version of the impulse response of the analog filter. That is,

$$h(n) \equiv h(nT) \quad n = 0, 1, 2, \dots \quad (8.3.17)$$

where  $T$  is the sampling interval.

To examine the implications of (8.3.17), we refer back to Section 4.2.9. Recall that when a continuous time signal  $x_a(t)$  with spectrum  $X_a(F)$  is sampled at a rate  $F_s = 1/T$  samples per second, the spectrum of the sampled signal is the periodic repetition of the scaled spectrum  $F_s X_a(F)$  with period  $F_s$ . Specifically, the relationship is

$$X(f) = F_s \sum_{k=-\infty}^{\infty} X_a[(f - k)F_s] \quad (8.3.18)$$

where  $f = F/F_s$  is the normalized frequency. Aliasing occurs if the sampling rate  $F_s$  is less than twice the highest frequency contained in  $X_a(F)$ .

Expressed in the context of sampling the impulse response of an analog filter with frequency response  $H_a(F)$ , the digital filter with unit sample response  $h(n) \equiv h_a(nT)$  has the frequency response

$$H(f) = F_s \sum_{k=-\infty}^{\infty} H_a[(f - k)F_s] \quad (8.3.19)$$

or, equivalently,

$$H(\omega) = F_s \sum_{k=-\infty}^{\infty} H_a[(\omega - 2\pi k)F_s] \quad (8.3.20)$$

or

$$H(\Omega T) = \frac{1}{T} \sum_{k=-\infty}^{\infty} H_a \left( \Omega - \frac{2\pi k}{T} \right) \quad (8.3.21)$$

Figure 8.31 depicts the frequency response of a lowpass analog filter and the frequency response of the corresponding digital filter.

It is clear that the digital filter with frequency response  $H(\omega)$  has the frequency response characteristics of the corresponding analog filter if the sampling interval  $T$  is selected sufficiently small to completely avoid or at least minimize the effects of aliasing. It is also clear that the impulse invariance method is inappropriate for designing highpass filters due to spectrum aliasing that results from the sampling process.

To investigate the mapping of points between the  $z$ -plane and the  $s$ -plane implied by the sampling process, we rely on a generalization of (8.3.21) which relates the  $z$ -transform of  $h(n)$  to the Laplace transform of  $h_a(t)$ . This relationship is

$$H(z)|_{z=e^{sT}} = \frac{1}{T} \sum_{k=-\infty}^{\infty} H_a \left( s - j \frac{2\pi k}{T} \right) \quad (8.3.22)$$

where

$$H(z) = \sum_{n=0}^{\infty} h(n)z^{-n}$$

$$H(z)|_{z=e^{sT}} = \sum_{n=0}^{\infty} h(n)e^{-sTn} \quad (8.3.23)$$

Note that when  $s = j\Omega$ , (8.3.22) reduces to (8.3.21), where the factor of  $j$  in  $H_a(\Omega)$  is suppressed in our notation.

Let us consider the mapping of points from the  $s$ -plane to the  $z$ -plane implied by the relation

$$z = e^{sT} \quad (8.3.24)$$

If we substitute  $s = \sigma + j\Omega$  and express the complex variable  $z$  in polar form as  $z = re^{j\omega}$ , (8.3.24) becomes

$$re^{j\omega} = e^{\sigma T} e^{j\Omega T}$$

Clearly, we must have

$$r = e^{\sigma T}$$

$$\omega = \Omega T \quad (8.3.25)$$

Consequently,  $\sigma < 0$  implies that  $0 < r < 1$  and  $\sigma > 0$  implies that  $r > 1$ . When  $\sigma = 0$ , we have  $r = 1$ . Therefore, the LHP in  $s$  is mapped inside the unit circle in  $z$  and the RHP in  $s$  is mapped outside the unit circle in  $z$ .

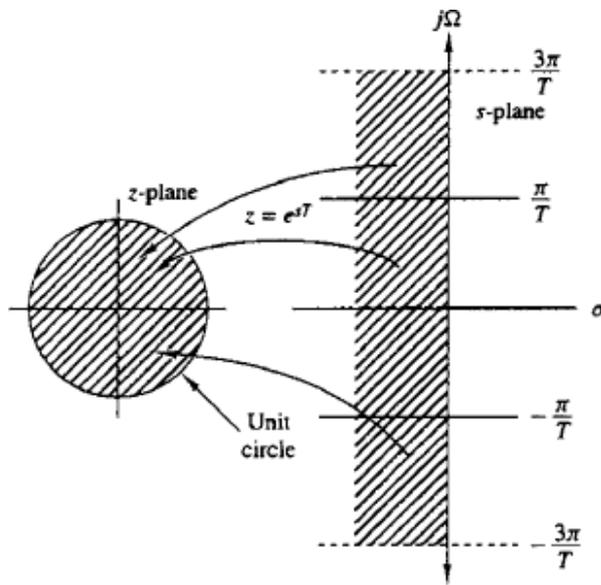
Also, the  $j\Omega$ -axis is mapped into the unit circle in  $z$  as indicated above. However, the mapping of the  $j\Omega$ -axis into the unit circle is not one-to-one. Since  $\omega$  is unique over the range  $(-\pi, \pi)$ , the mapping  $\omega = \Omega T$  implies that the interval  $-\pi/T \leq \Omega \leq \pi/T$  maps into the corresponding values of  $-\pi \leq \omega \leq \pi$ . Furthermore, the frequency interval  $\pi/T \leq \Omega \leq 3\pi/T$  also maps into the interval  $-\pi \leq \omega \leq \pi$  and, in general, so does the interval  $(2k-1)\pi/T \leq \Omega \leq (2k+1)\pi/T$ , when  $k$  is an integer. Thus the mapping from the analog frequency  $\Omega$  to the frequency variable  $\omega$  in the digital domain is many-to-one, which simply reflects the effects of aliasing due to sampling. Figure 8.32 illustrates the mapping from the  $s$ -plane to the  $z$ -plane for the relation in (8.3.24).

To explore further the effect of the impulse invariance design method on the characteristics of the resulting filter, let us express the system function of the analog filter in partial-fraction form. On the assumption that the poles of the analog filter are distinct, we can write

$$H_a(s) = \sum_{k=1}^N \frac{c_k}{s - p_k} \quad (8.3.26)$$

where  $\{p_k\}$  are the poles of the analog filter and  $\{c_k\}$  are the coefficients in the partial-fraction expansion. Consequently,

$$h_a(t) = \sum_{k=1}^N c_k e^{p_k t} \quad t \geq 0 \quad (8.3.27)$$



**Figure 8.32** The mapping of  $z = e^{sT}$  maps strips of width  $2\pi/T$  (for  $\sigma < 0$ ) in the  $s$ -plane into points in the unit circle in the  $z$ -plane.

If we sample  $h_a(t)$  periodically at  $t = nT$ , we have

$$\begin{aligned} h(n) &= h_a(nT) \\ &= \sum_{k=1}^N c_k e^{p_k T n} \end{aligned} \quad (8.3.28)$$

Now, with the substitution of (8.3.28), the system function of the resulting digital IIR filter becomes

$$\begin{aligned} H(z) &= \sum_{n=0}^{\infty} h(n) z^{-n} \\ &= \sum_{n=0}^{\infty} \left( \sum_{k=1}^N c_k e^{p_k T n} \right) z^{-n} \\ &= \sum_{k=1}^N c_k \sum_{n=0}^{\infty} (e^{p_k T} z^{-1})^n \end{aligned} \quad (8.3.29)$$

The inner sum in (8.3.29) converges because  $p_k < 0$  and yields

$$\sum_{n=0}^{\infty} (e^{p_k T} z^{-1})^n = \frac{1}{1 - e^{p_k T} z^{-1}} \quad (8.3.30)$$

Therefore, the system function of the digital filter is

$$H(z) = \sum_{k=1}^N \frac{c_k}{1 - e^{p_k T} z^{-1}} \quad (8.3.31)$$

We observe that the digital filter has poles at

$$z_k = e^{p_k T} \quad k = 1, 2, \dots, N \quad (8.3.32)$$

### 8.3.3 IIR Filter Design by the Bilinear Transformation

The IIR filter design techniques described in the preceding two sections have a severe limitation in that they are appropriate only for lowpass filters and a limited class of bandpass filters.

In this section we describe a mapping from the  $s$ -plane to the  $z$ -plane, called the bilinear transformation, that overcomes the limitation of the other two design

methods described previously. The bilinear transformation is a conformal mapping that transforms the  $j\Omega$ -axis into the unit circle in the  $z$ -plane only once, thus avoiding aliasing of frequency components. Furthermore, all points in the LHP of  $s$  are mapped inside the unit circle in the  $z$ -plane and all points in the RHP of  $s$  are mapped into corresponding points outside the unit circle in the  $z$ -plane.

The bilinear transformation can be linked to the trapezoidal formula for numerical integration. For example, let us consider an analog linear filter with system function

$$H(s) = \frac{b}{s+a} \quad (8.3.33)$$

This system is also characterized by the differential equation

$$\frac{dy(t)}{dt} + ay(t) = bx(t) \quad (8.3.34)$$

Instead of substituting a finite difference for the derivative, suppose that we integrate the derivative and approximate the integral by the trapezoidal formula. Thus

$$y(t) = \int_{t_0}^t y'(\tau) d\tau + y(t_0) \quad (8.3.35)$$

where  $y'(t)$  denotes the derivative of  $y(t)$ . The approximation of the integral in (8.3.35) by the trapezoidal formula at  $t = nT$  and  $t_0 = nT - T$  yields

$$y(nT) = \frac{T}{2} [y'(nT) + y'(nT - T)] + y(nT - T) \quad (8.3.36)$$

Now the differential equation in (8.3.34) evaluated at  $t = nT$  yields

$$y'(nT) = -ay(nT) + bx(nT) \quad (8.3.37)$$

We use (8.3.37) to substitute for the derivative in (8.3.36) and thus obtain a difference equation for the equivalent discrete-time system. With  $y(n) \equiv y(nT)$  and  $x(n) \equiv x(nT)$ , we obtain the result

$$\left(1 + \frac{aT}{2}\right) y(n) - \left(1 - \frac{aT}{2}\right) y(n-1) = \frac{bT}{2} [x(n) + x(n-1)] \quad (8.3.38)$$

The  $z$ -transform of this difference equation is

$$\left(1 + \frac{aT}{2}\right) Y(z) - \left(1 - \frac{aT}{2}\right) z^{-1} Y(z) = \frac{bT}{2} (1 + z^{-1}) X(z)$$

Consequently, the system function of the equivalent digital filter is

$$H(z) = \frac{Y(z)}{X(z)} = \frac{(bT/2)(1 + z^{-1})}{1 + aT/2 - (1 - aT/2)z^{-1}}$$

or, equivalently,

$$H(z) = \frac{b}{\frac{2}{T} \left( \frac{1 - z^{-1}}{1 + z^{-1}} \right) + a} \quad (8.3.39)$$

Clearly, the mapping from the  $s$ -plane to the  $z$ -plane is

$$s = \frac{2}{T} \left( \frac{1 - z^{-1}}{1 + z^{-1}} \right) \quad (8.3.40)$$

This is called the *bilinear transformation*.

Although our derivation of the bilinear transformation was performed for a first-order differential equation, it holds, in general, for an  $N$ th-order differential equation.

To investigate the characteristics of the bilinear transformation, let

$$z = r e^{j\omega}$$

$$s = \sigma + j\Omega$$

Then (8.3.40) can be expressed as

$$s = \frac{2}{T} \frac{z - 1}{z + 1}$$

$$= \frac{2}{T} \frac{r e^{j\omega} - 1}{r e^{j\omega} + 1}$$

$$= \frac{2}{T} \left( \frac{r^2 - 1}{1 + r^2 + 2r \cos \omega} + j \frac{2r \sin \omega}{1 + r^2 + 2r \cos \omega} \right)$$

Consequently,

$$\sigma = \frac{2}{T} \frac{r^2 - 1}{1 + r^2 + 2r \cos \omega} \quad (8.3.41)$$

$$\Omega = \frac{2}{T} \frac{2r \sin \omega}{1 + r^2 + 2r \cos \omega} \quad (8.3.42)$$

First, we note that if  $r < 1$ , then  $\sigma < 0$ , and if  $r > 1$ , then  $\sigma > 0$ . Consequently, the LHP in  $s$  maps into the inside of the unit circle in the  $z$ -plane and the RHP in  $s$  maps into the outside of the unit circle. When  $r = 1$ , then  $\sigma = 0$  and

$$\Omega = \frac{2}{T} \frac{\sin \omega}{1 + \cos \omega}$$

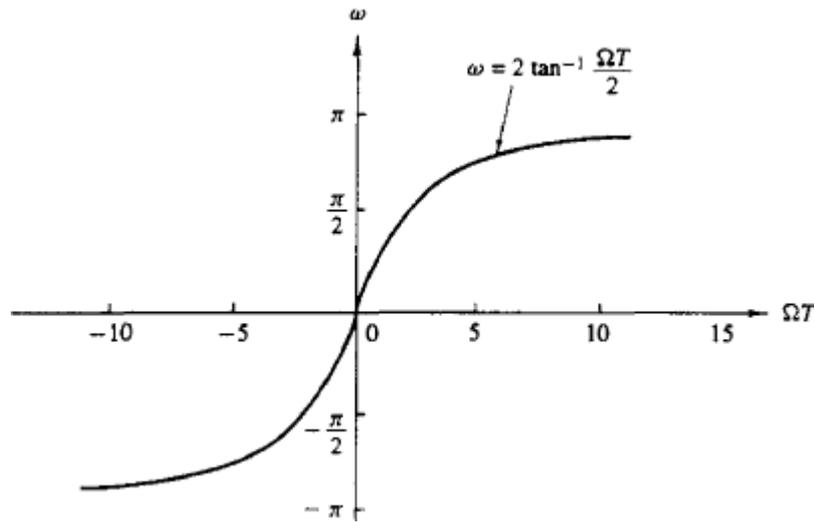
$$= \frac{2}{T} \tan \frac{\omega}{2} \quad (8.3.43)$$

or, equivalently,

$$\omega = 2 \tan^{-1} \frac{\Omega T}{2} \quad (8.3.44)$$

The relationship in (8.3.44) between the frequency variables in the two domains is illustrated in Fig. 8.36. We observe that the entire range in  $\Omega$  is mapped only once into the range  $-\pi \leq \omega \leq \pi$ . However, the mapping is highly nonlinear. We observe a frequency compression or *frequency warping*, as it is usually called, due to the nonlinearity of the arctangent function.

It is also interesting to note that the bilinear transformation maps the point  $s = \infty$  into the point  $z = -1$ . Consequently, the single-pole lowpass filter in



**Chebyshev filters.** There are two types of Chebyshev filters. Type I Chebyshev filters are all-pole filters that exhibit equiripple behavior in the passband and a monotonic characteristic in the stopband. On the other hand, the family of type II Chebyshev filters contains both poles and zeros and exhibits a

monotonic behavior in the passband and an equiripple behavior in the stopband. The zeros of this class of filters lie on the imaginary axis in the  $s$ -plane.

The magnitude squared of the frequency response characteristic of a type I Chebyshev filter is given as

$$|H(\Omega)|^2 = \frac{1}{1 + \epsilon^2 T_N^2(\Omega/\Omega_p)} \quad (8.3.51)$$

where  $\epsilon$  is a parameter of the filter related to the ripple in the passband and  $T_N(x)$  is the  $N$ th-order Chebyshev polynomial defined as

$$T_N(x) = \begin{cases} \cos(N \cos^{-1} x), & |x| \leq 1 \\ \cosh(N \cosh^{-1} x), & |x| > 1 \end{cases} \quad (8.3.52)$$

The Chebyshev polynomials can be generated by the recursive equation

$$T_{N+1}(x) = 2xT_N(x) - T_{N-1}(x) \quad N = 1, 2, \dots \quad (8.3.53)$$

where  $T_0(x) = 1$  and  $T_1(x) = x$ . From (8.3.53) we obtain  $T_2(x) = 2x^2 - 1$ ,  $T_3(x) = 4x^3 - 3x$ , and so on.

Some of the properties of these polynomials are as follows:

1.  $|T_N(x)| \leq 1$  for all  $|x| \leq 1$ .
2.  $T_N(1) = 1$  for all  $N$ .
3. All the roots of the polynomial  $T_N(x)$  occur in the interval  $-1 \leq x \leq 1$ .

The filter parameter  $\epsilon$  is related to the ripple in the passband, as illustrated in Fig. 8.39, for  $N$  odd and  $N$  even. For  $N$  odd,  $T_N(0) = 0$  and hence  $|H(0)|^2 = 1$ . On the other hand, for  $N$  even,  $T_N(0) = 1$  and hence  $|H(0)|^2 = 1/(1 + \epsilon^2)$ . At the band edge frequency  $\Omega = \Omega_p$ , we have  $T_N(1) = 1$ , so that

$$\frac{1}{\sqrt{1 + \epsilon^2}} = 1 - \delta_1$$

or, equivalently,

$$\epsilon^2 = \frac{1}{(1 - \delta_1)^2} - 1 \quad (8.3.54)$$

where  $\delta_1$  is the value of the passband ripple.

The poles of a type I Chebyshev filter lie on an ellipse in the  $s$ -plane with major axis

$$r_1 = \Omega_p \frac{\beta^2 + 1}{2\beta} \quad (8.3.55)$$

and minor axis

$$r_2 = \Omega_p \frac{\beta^2 - 1}{2\beta} \quad (8.3.56)$$

where  $\beta$  is related to  $\epsilon$  according to the equation

$$\beta = \left[ \frac{\sqrt{1 + \epsilon^2} + 1}{\epsilon} \right]^{1/N} \quad (8.3.57)$$

The pole locations are most easily determined for a filter of order  $N$  by first locating the poles for an equivalent  $N$ th-order Butterworth filter that lie on circles of radius  $r_1$  or radius  $r_2$ , as illustrated in Fig. 8.40. If we denote the angular positions of the poles of the Butterworth filter as

$$\phi_k = \frac{\pi}{2} + \frac{(2k + 1)\pi}{2N} \quad k = 0, 1, 2, \dots, N - 1 \quad (8.3.58)$$

then the positions of the poles for the Chebyshev filter lie on the ellipse at the coordinates  $(x_k, y_k)$ ,  $k = 0, 1, \dots, N - 1$ , where

$$\begin{aligned} x_k &= r_2 \cos \phi_k, & k &= 0, 1, \dots, N - 1 \\ y_k &= r_1 \sin \phi_k, & k &= 0, 1, \dots, N - 1 \end{aligned} \quad (8.3.59)$$

A type II Chebyshev filter contains zeros as well as poles. The magnitude squared of its frequency response is given as

$$|H(\Omega)|^2 = \frac{1}{1 + \epsilon^2 [T_N^2(\Omega_s/\Omega_p) / T_N^2(\Omega_s/\Omega)]} \quad (8.3.60)$$

where  $T_N(x)$  is, again, the  $N$ th-order Chebyshev polynomial and  $\Omega_s$  is the stopband frequency as illustrated in Fig. 8.41. The zeros are located on the imaginary axis at the points

$$s_k = j \frac{\Omega_s}{\sin \phi_k} \quad k = 0, 1, \dots, N - 1 \quad (8.3.61)$$

The poles are located at the points  $(v_k, w_k)$ , where

$$v_k = \frac{\Omega_s x_k}{\sqrt{x_k^2 + y_k^2}} \quad k = 0, 1, \dots, N-1 \quad (8.3.62)$$

$$w_k = \frac{\Omega_s y_k}{\sqrt{x_k^2 + y_k^2}} \quad k = 0, 1, \dots, N-1 \quad (8.3.63)$$

where  $\{x_k\}$  and  $\{y_k\}$  are defined in (8.3.59) with  $\beta$  now related to the ripple in the stopband through the equation

$$\beta = \left[ \frac{1 + \sqrt{1 - \delta_2^2}}{\delta_2} \right]^{1/N} \quad (8.3.64)$$

$$N = \frac{\log \left[ \left( \frac{\sqrt{1 - \delta_2^2} + \sqrt{1 - \delta_2^2(1 + \epsilon^2)}}{\epsilon \delta_2} \right) \right]}{\log \left[ (\Omega_s / \Omega_p) + \sqrt{(\Omega_s / \Omega_p)^2 - 1} \right]}$$

$$= \frac{\cosh^{-1}(\delta/\epsilon)}{\cosh^{-1}(\Omega_s / \Omega_p)}$$

where, by definition,  $\delta_2 = 1/\sqrt{1 + \delta^2}$ .

### Frequency Transformations in the Analog Domain

Type of transformation	Transformation	Band edge frequencies of new filter
Lowpass	$s \rightarrow \frac{\Omega_p}{\Omega'_p} s$	$\Omega'_p$
Highpass	$s \rightarrow \frac{\Omega_p \Omega'_p}{s}$	$\Omega'_p$
Bandpass	$s \rightarrow \Omega_p \frac{s^2 + \Omega_l \Omega_u}{s(\Omega_u - \Omega_l)}$	$\Omega_l, \Omega_u$
Bandstop	$s \rightarrow \Omega_p \frac{s(\Omega_u - \Omega_l)}{s^2 + \Omega_u \Omega_l}$	$\Omega_l, \Omega_u$

### Frequency Transformations in the Digital Domain

Type of transformation	Transformation	Parameters
Lowpass	$z^{-1} \rightarrow \frac{z^{-1} - a}{1 - az^{-1}}$	$\omega'_p =$ band edge frequency of new filter $a = \frac{\sin[(\omega_p - \omega'_p)/2]}{\sin[(\omega_p + \omega'_p)/2]}$
Highpass	$z^{-1} \rightarrow -\frac{z^{-1} + a}{1 + az^{-1}}$	$\omega'_p =$ band edge frequency new filter $a = -\frac{\cos[(\omega_p + \omega'_p)/2]}{\cos[(\omega_p - \omega'_p)/2]}$
Bandpass	$z^{-1} \rightarrow -\frac{z^{-2} - a_1z^{-1} + a_2}{a_2z^{-2} - a_1z^{-1} + 1}$	$\omega_l =$ lower band edge frequency $\omega_u =$ upper band edge frequency $a_1 = -2\alpha K / (K + 1)$ $a_2 = (K - 1) / (K + 1)$ $\alpha = \frac{\cos[(\omega_u + \omega_l)/2]}{\cos[(\omega_u - \omega_l)/2]}$ $K = \cot \frac{\omega_u - \omega_l}{2} \tan \frac{\omega_p}{2}$
Bandstop	$z^{-1} \rightarrow -\frac{z^{-2} - a_1z^{-1} + a_2}{a_2z^{-2} - a_1z^{-1} + 1}$	$\omega_l =$ lower band edge frequency $\omega_u =$ upper band edge frequency $a_1 = -2\alpha / (K + 1)$ $a_2 = (1 - K) / (1 + K)$ $\alpha = \frac{\cos[(\omega_u + \omega_l)/2]}{\cos[(\omega_u - \omega_l)/2]}$ $K = \tan \frac{\omega_u - \omega_l}{2} \tan \frac{\omega_p}{2}$

## STRUCTURES FOR IIR SYSTEMS

In this section we consider different IIR systems structures described by the difference equation given by the system function. Just as in the case of FIR systems, there are several types of structures or realizations, including direct-form structures, cascade-form structures, lattice structures, and lattice-ladder structures. In addition, IIR systems lend themselves to a parallel form realization. We begin by describing two direct-form realizations.

### DIRECT FORM STRUCTURES:

The rational system function as given by (7.1.2) that characterizes an IIR system can be viewed as two systems in cascade, that is,

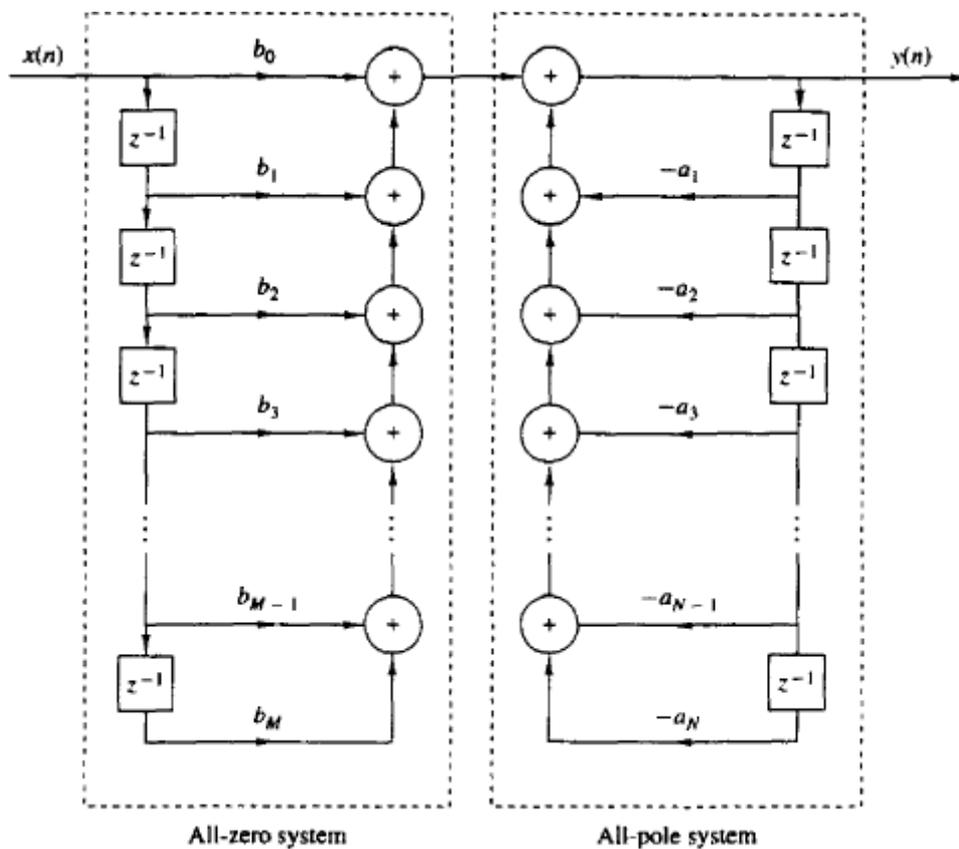
$$H(z) = H_1(z)H_2(z) \quad (7.3.1)$$

where  $H_1(z)$  consists of the zeros of  $H(z)$ , and  $H_2(z)$  consists of the poles of  $H(z)$ ,

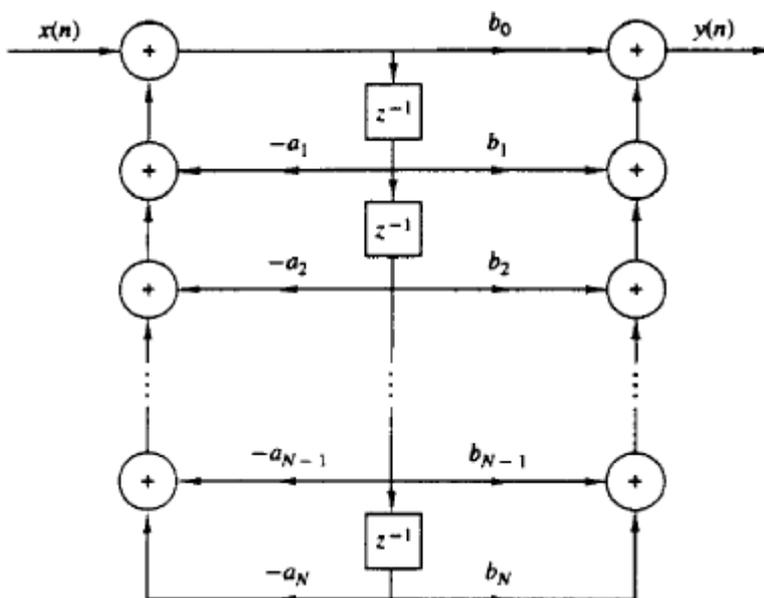
$$H_1(z) = \sum_{k=0}^M b_k z^{-k} \quad (7.3.2)$$

and

$$H_2(z) = \frac{1}{1 + \sum_{k=1}^N a_k z^{-k}} \quad (7.3.3)$$

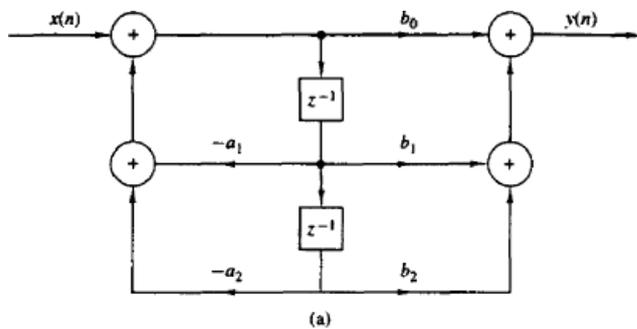


### DIRECT FORM II

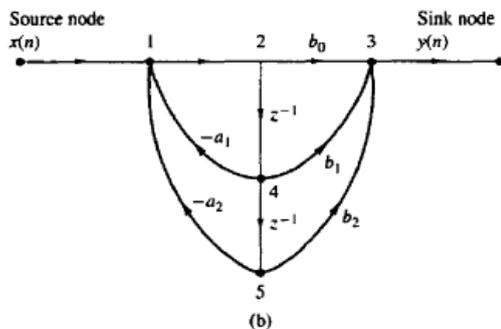


### Signal Flow Graphs and Transposed Structures

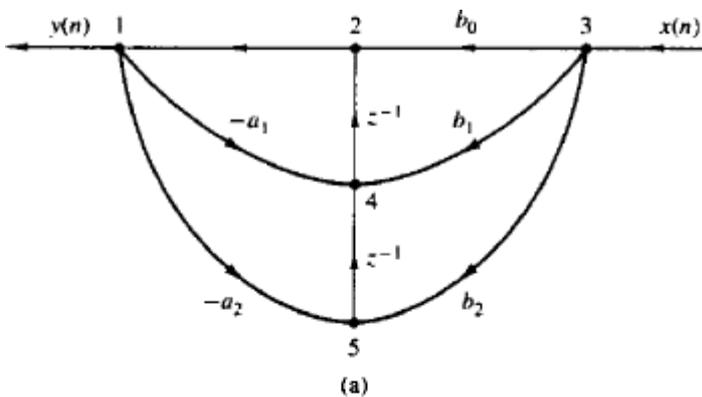
A signal flow graph provides an alternative, N but equivalent, graphical representation to a block diagram structure that we have been using to illustrate various system realizations. The basic elements of a flow graph are branches and nodes. A signal flow graph is basically a set of directed branches that connect at nodes. By definition, the signal out of a branch is equal to the branch gain (system function) times the signal into the branch. Furthermore, the signal at a node of a flow graph is equal to the sum of the signals from all branches connecting to the node.



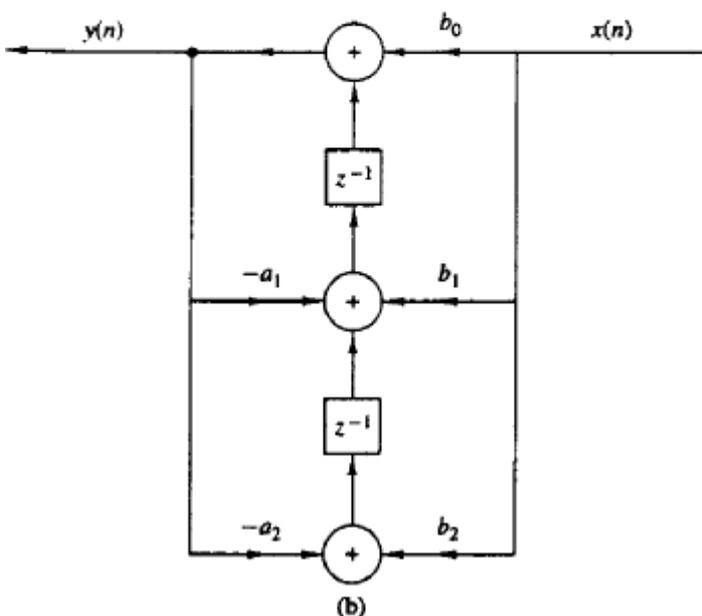
(a)



(b)



(a)



(b)

### Cascade-Form Structures

Let us consider a high-order IIR system with system function given by equation. Without loss of generality we assume that  $N > M$ . The system can be factored into a cascade of second-order subsystems, such that  $H(z)$  can be expressed as

$$H(z) = \prod_{k=1}^K H_k(z)$$

where  $K$  is the integer part of  $(N + 1)/2$ .  $H_k(z)$  has the general form

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

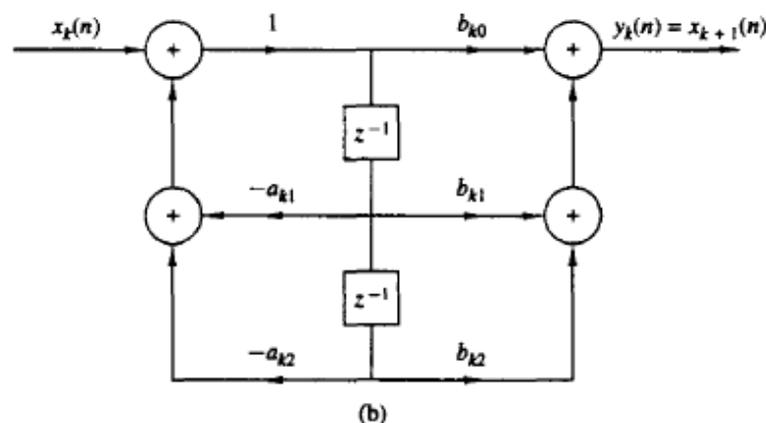
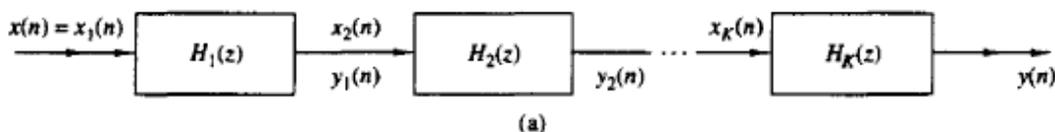
The general form of the cascade structure is illustrated in Fig. 7.19. If we use the direct form II structure for each of the subsystems, the computational algorithm for realizing the IIR system with system function  $H(z)$  is described by the following set of equations.

$$y_0(n) = x(n) \quad (7.3.16)$$

$$w_k(n) = -a_{k1}w_k(n-1) - a_{k2}w_k(n-2) + y_{k-1}(n) \quad k = 1, 2, \dots, K \quad (7.3.17)$$

$$y_k(n) = b_{k0}w_k(n) + b_{k1}w_k(n-1) + b_{k2}w_k(n-2) \quad k = 1, 2, \dots, K \quad (7.3.18)$$

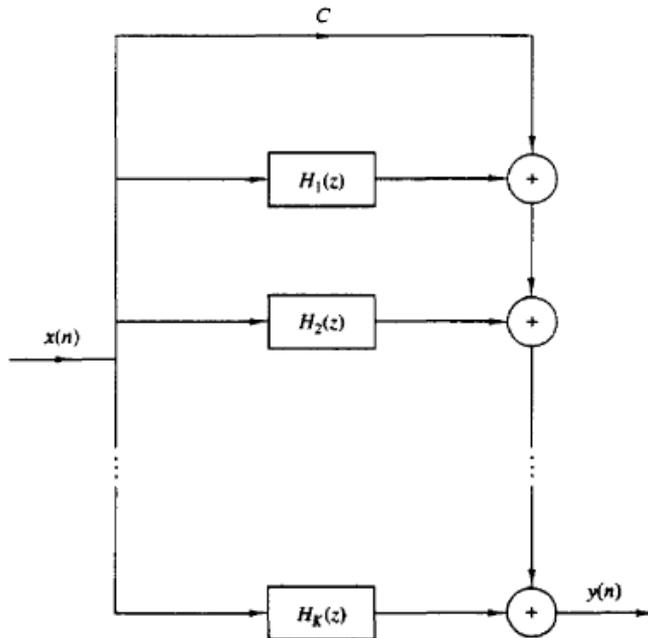
$$y(n) = y_K(n) \quad (7.3.19)$$



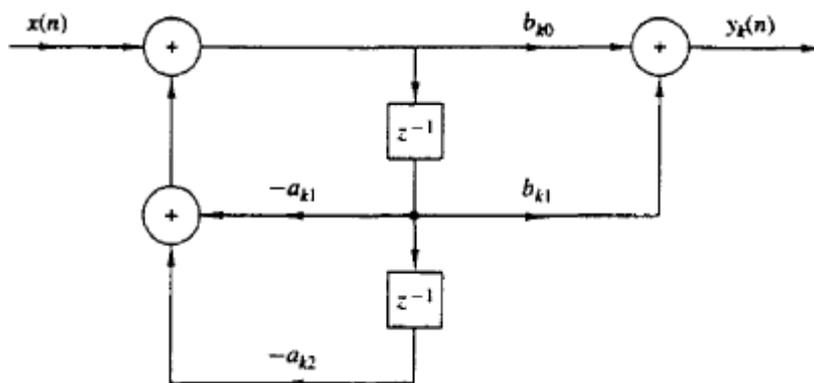
### Parallel-Form Structures

A parallel-form realization of an IIR system can be obtained by performing a partial-fraction expansion of  $H(z)$ . Without loss of generality, we again assume that  $N > M$  and that the poles are distinct. Then, by performing a partial-fraction expansion of  $H(z)$ , we obtain the result

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



The realization of second order form is given by



The general form of parallel form of structure is given by

$$w_k(n) = -a_{k1}w_k(n-1) - a_{k2}w_k(n-2) + x(n) \quad k = 1, 2, \dots, K$$

$$y_k(n) = b_{k0}w_k(n) + b_{k1}w_k(n-1) \quad k = 1, 2, \dots, K$$

$$y(n) = Cx(n) + \sum_{k=1}^K y_k(n)$$

### Lattice and Lattice-Ladder Structures for IIR Systems

In Section 7.2.4 we developed a lattice filter structure that is equivalent to an FIR system. In this section we extend the development to IIR systems.

Let us begin with an all-pole system with system function

$$H(z) = \frac{1}{1 + \sum_{k=1}^N a_N(k)z^{-k}} = \frac{1}{A_N(z)} \quad (7.3.26)$$

The direct form realization of this system is illustrated in Fig. 7.23. The difference equation for this IIR system is

$$y(n) = -\sum_{k=1}^N a_N(k)y(n-k) + x(n) \quad (7.3.27)$$

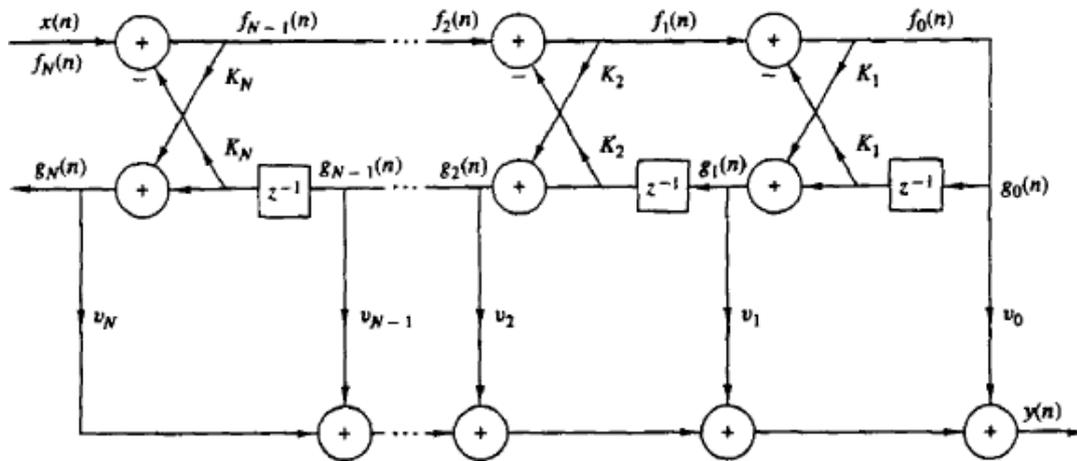
It is interesting to note that if we interchange the roles of input and output [i.e., interchange  $x(n)$  with  $y(n)$  in (7.3.27)], we obtain

$$x(n) = -\sum_{k=1}^N a_N(k)x(n-k) + y(n)$$

or, equivalently,

$$y(n) = x(n) + \sum_{k=1}^N a_N(k)x(n-k) \quad (7.3.28)$$

We note that the equation in (7.3.28) describes an FIR system having the system function  $H(z) = A_N(z)$ , while the system described by the difference equation in (7.3.27) represents an IIR system with system function  $H(z) = 1/A_N(z)$ .



**UNIT – IV**  
**FIR DIGITAL FILTERS**

The transfer function is obtained by taking Z transform of finite sample impulse response. The filters designed by using finite samples of impulse response are called FIR filters.

Some of the advantages of FIR filter are linear phase, both recursive and non recursive, stable and round off noise can be made smaller.

Some of the disadvantages of FIR filters are large amount of processing is required and non integral delay may lead to problems.

**DESIGN OF FIR FILTERS**

An FIR filter of length  $M$  with input  $x(n)$  and output  $y(n)$  is described by the difference equation

$$y(n) = b_0x(n) + b_1x(n-1) + \dots + b_{M-1}x(n-M+1) \\ = \sum_{k=0}^{M-1} b_kx(n-k) \quad (8.2.1)$$

where  $\{b_k\}$  is the set of filter coefficients. Alternatively, we can express the output sequence as the convolution of the unit sample response  $h(n)$  of the system with the input signal. Thus we have

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k) \quad (8.2.2)$$

where the lower and upper limits on the convolution sum reflect the causality and finite-duration characteristics of the filter. Clearly, (8.2.1) and (8.2.2) are identical in form and hence it follows that  $b_k = h(k)$ ,  $k = 0, 1, \dots, M-1$ .

The filter can also be characterized by its system function

$$H(z) = \sum_{k=0}^{M-1} h(k)z^{-k} \quad (8.2.3)$$

which we view as a polynomial of degree  $M-1$  in the variable  $z^{-1}$ . The roots of this polynomial constitute the zeros of the filter.

An FIR filter has linear phase if its unit sample response satisfies the condition

$$h(n) = \pm h(M-1-n) \quad n = 0, 1, \dots, M-1 \quad (8.2.4)$$

When the symmetry and antisymmetry conditions in (8.2.4) are incorporated into (8.2.3), we have

$$H(z) = h(0) + h(1)z^{-1} + h(2)z^{-2} + \dots + h(M-2)z^{-(M-2)} + h(M-1)z^{-(M-1)} \\ = z^{-(M-1)/2} \left\{ h\left(\frac{M-1}{2}\right) + \sum_{n=0}^{(M-3)/2} h(n) [z^{(M-1-2k)/2} \pm z^{-(M-1-2k)/2}] \right\} \quad M \text{ odd} \\ = z^{-(M-1)/2} \sum_{n=0}^{(M/2)-1} h(n) [z^{(M-1-2k)/2} \pm z^{-(M-1-2k)/2}] \quad M \text{ even} \quad (8.2.5)$$

Now, if we substitute  $z^{-1}$  for  $z$  in (8.2.3) and multiply both sides of the resulting equation by  $z^{-(M-1)}$ , we obtain

$$z^{-(M-1)}H(z^{-1}) = \pm H(z) \quad (8.2.6)$$

When  $h(n) = h(M - 1 - n)$ ,  $H(\omega)$  can be expressed as

$$H(\omega) = H_r(\omega)e^{-j\omega(M-1)/2} \quad (8.2.7)$$

where  $H_r(\omega)$  is a real function of  $\omega$  and can be expressed as

$$H_r(\omega) = h\left(\frac{M-1}{2}\right) + 2 \sum_{n=0}^{(M-3)/2} h(n) \cos \omega \left(\frac{M-1}{2} - n\right) \quad M \text{ odd} \quad (8.2.8)$$

$$H_r(\omega) = 2 \sum_{n=0}^{(M/2)-1} h(n) \cos \omega \left(\frac{M-1}{2} - n\right) \quad M \text{ even} \quad (8.2.9)$$

The phase characteristic of the filter for both  $M$  odd and  $M$  even is

$$\Theta(\omega) = \begin{cases} -\omega \left(\frac{M-1}{2}\right), & \text{if } H_r(\omega) > 0 \\ -\omega \left(\frac{M-1}{2}\right) + \pi, & \text{if } H_r(\omega) < 0 \end{cases} \quad (8.2.10)$$

When

$$h(n) = -h(M - 1 - n)$$

the unit sample response is *antisymmetric*. For  $M$  odd, the center point of the antisymmetric  $h(n)$  is  $n = (M - 1)/2$ . Consequently,

$$h\left(\frac{M-1}{2}\right) = 0$$

It is straightforward to show that the frequency response of an FIR filter with an antisymmetric unit sample response can be expressed as

$$H(\omega) = H_r(\omega)e^{j[-\omega(M-1)/2 + \pi/2]} \quad (8.2.11)$$

where

$$H_r(\omega) = 2 \sum_{n=0}^{(M-3)/2} h(n) \sin \omega \left(\frac{M-1}{2} - n\right) \quad M \text{ odd} \quad (8.2.12)$$

$$H_r(\omega) = 2 \sum_{n=0}^{(M/2)-1} h(n) \sin \omega \left(\frac{M-1}{2} - n\right) \quad M \text{ even} \quad (8.2.13)$$

The phase characteristic of the filter for both  $M$  odd and  $M$  even is

$$\Theta(\omega) = \begin{cases} \frac{\pi}{2} - \omega \left(\frac{M-1}{2}\right), & \text{if } H_r(\omega) > 0 \\ \frac{3\pi}{2} - \omega \left(\frac{M-1}{2}\right), & \text{if } H_r(\omega) < 0 \end{cases} \quad (8.2.14)$$

The choice of a symmetric or antisymmetric unit sample response depends on the application. As we shall see later, a symmetric unit sample response is suitable for some applications, while an antisymmetric unit sample response is more suitable for other applications. For example, if  $h(n) = -h(M-1-n)$  and  $M$  is odd, (8.2.12) implies that  $H_r(0) = 0$  and  $H_r(\pi) = 0$ . Consequently, (8.2.12) is not suitable as either a lowpass filter or a highpass filter. Similarly, the antisymmetric unit sample response with  $M$  even also results in  $H_r(0) = 0$ , as can be easily verified from (8.2.13). Consequently, we would not use the antisymmetric condition in the design of a lowpass linear-phase FIR filter. On the other hand, the symmetry condition  $h(n) = h(M-1-n)$  yields a linear-phase FIR filter with a nonzero response at  $\omega = 0$ , if desired, that is,

$$H_r(0) = h\left(\frac{M-1}{2}\right) + 2 \sum_{n=0}^{(M-3)/2} h(n), \quad M \text{ odd} \quad (8.2.15)$$

$$H_r(0) = 2 \sum_{n=0}^{(M/2)-1} h(n), \quad M \text{ even} \quad (8.2.16)$$

### 8.2.2 Design of Linear-Phase FIR Filters Using Windows

In this method we begin with the desired frequency response specification  $H_d(\omega)$  and determine the corresponding unit sample response  $h_d(n)$ . Indeed,  $h_d(n)$  is related to  $H_d(\omega)$  by the Fourier transform relation

$$H_d(\omega) = \sum_{n=0}^{\infty} h_d(n) e^{-j\omega n} \quad (8.2.17)$$

where

$$h_d(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(\omega) e^{j\omega n} d\omega \quad (8.2.18)$$

Thus, given  $H_d(\omega)$ , we can determine the unit sample response  $h_d(n)$  by evaluating the integral in (8.2.18).

In general, the unit sample response  $h_d(n)$  obtained from (8.2.17) is infinite in duration and must be truncated at some point, say at  $n = M - 1$ , to yield an FIR filter of length  $M$ . Truncation of  $h_d(n)$  to a length  $M - 1$  is equivalent to multiplying  $h_d(n)$  by a "rectangular window," defined as

$$w(n) = \begin{cases} 1, & n = 0, 1, \dots, M - 1 \\ 0, & \text{otherwise} \end{cases} \quad (8.2.19)$$

Thus the unit sample response of the FIR filter becomes

$$\begin{aligned} h(n) &= h_d(n)w(n) \\ &= \begin{cases} h_d(n), & n = 0, 1, \dots, M - 1 \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (8.2.20)$$

It is instructive to consider the effect of the window function on the desired frequency response  $H_d(\omega)$ . Recall that multiplication of the window function  $w(n)$  with  $h_d(n)$  is equivalent to convolution of  $H_d(\omega)$  with  $W(\omega)$ , where  $W(\omega)$  is the frequency-domain representation (Fourier transform) of the window function, that is,

$$W(\omega) = \sum_{n=0}^{M-1} w(n)e^{-j\omega n} \quad (8.2.21)$$

Thus the convolution of  $H_d(\omega)$  with  $W(\omega)$  yields the frequency response of the (truncated) FIR filter. That is,

$$H(\omega) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(\nu)W(\omega - \nu)d\nu \quad (8.2.22)$$

The Fourier transform of the rectangular window is

$$\begin{aligned} W(\omega) &= \sum_{n=0}^{M-1} e^{-j\omega n} \\ &= \frac{1 - e^{-j\omega M}}{1 - e^{-j\omega}} = e^{-j\omega(M-1)/2} \frac{\sin(\omega M/2)}{\sin(\omega/2)} \end{aligned} \quad (8.2.23)$$

This window function has a magnitude response

$$|W(\omega)| = \frac{|\sin(\omega M/2)|}{|\sin(\omega/2)|} \quad -\pi \leq \omega \leq \pi \quad (8.2.24)$$

and a piecewise linear phase

$$\Theta(\omega) = \begin{cases} -\omega \left( \frac{M-1}{2} \right), & \text{when } \sin(\omega M/2) \geq 0 \\ -\omega \left( \frac{M-1}{2} \right) + \pi, & \text{when } \sin(\omega M/2) < 0 \end{cases} \quad (8.2.25)$$

The magnitude response of the window function is illustrated in Fig. 8.4 for  $M = 31$  and 61. The width of the main lobe [width is measured to the first zero of  $W(\omega)$ ]

Name of window	Time-domain sequence, $h(n), 0 \leq n \leq M-1$
Bartlett (triangular)	$1 - \frac{2 \left  n - \frac{M-1}{2} \right }{M-1}$
Blackman	$0.42 - 0.5 \cos \frac{2\pi n}{M-1} + 0.08 \cos \frac{4\pi n}{M-1}$
Hamming	$0.54 - 0.46 \cos \frac{2\pi n}{M-1}$
Hanning	$\frac{1}{2} \left( 1 - \cos \frac{2\pi n}{M-1} \right)$
Kaiser	$\frac{I_0 \left[ \alpha \sqrt{\left( \frac{M-1}{2} \right)^2 - \left( n - \frac{M-1}{2} \right)^2} \right]}{I_0 \left[ \alpha \left( \frac{M-1}{2} \right) \right]}$
Lanczos	$\left\{ \frac{\sin \left[ 2\pi \left( n - \frac{M-1}{2} \right) / (M-1) \right]}{2\pi \left( n - \frac{M-1}{2} \right) / \left( \frac{M-1}{2} \right)} \right\}^L$ $L > 0$ $1, \left  n - \frac{M-1}{2} \right  \leq \alpha \frac{M-1}{2} \quad 0 < \alpha < 1$
Tukey	$\frac{1}{2} \left[ 1 + \cos \left( \frac{n - (1+\alpha)(M-1)/2}{(1-\alpha)(M-1)/2} \pi \right) \right]$ $\alpha(M-1)/2 \leq \left  n - \frac{M-1}{2} \right  \leq \frac{M-1}{2}$

### 8.2.3 Design of Linear-Phase FIR Filters by the Frequency-Sampling Method

In the frequency sampling method for FIR filter design, we specify the desired frequency response  $H_d(\omega)$  at a set of equally spaced frequencies, namely

$$\omega_k = \frac{2\pi}{M}(k + \alpha) \quad k = 0, 1, \dots, \frac{M-1}{2} \quad M \text{ odd}$$

$$k = 0, 1, \dots, \frac{M}{2} - 1 \quad M \text{ even} \quad (8.2.30)$$

$$\alpha = 0 \quad \text{or} \quad \frac{1}{2}$$

and solve for the unit sample response  $h(n)$  of the FIR filter from these equally

spaced frequency specifications. To reduce sidelobes, it is desirable to optimize the frequency specification in the transition band of the filter. This optimization can be accomplished numerically on a digital computer by means of linear programming techniques as shown by Rabiner et al. (1970).

In this section we exploit a basic symmetry property of the sampled frequency response function to simplify the computations. Let us begin with the desired frequency response of the FIR filter, which is [for simplicity, we drop the subscript in  $H_d(\omega)$ ],

$$H(\omega) = \sum_{n=0}^{M-1} h(n)e^{-j\omega n} \quad (8.2.31)$$

Suppose that we specify the frequency response of the filter at the frequencies given by (8.2.30). Then from (8.2.31) we obtain

$$H(k + \alpha) \equiv H\left(\frac{2\pi}{M}(k + \alpha)\right)$$

$$H(k + \alpha) \equiv \sum_{n=0}^{M-1} h(n)e^{-j2\pi(k+\alpha)n/M} \quad k = 0, 1, \dots, M - 1 \quad (8.2.32)$$

It is a simple matter to invert (8.2.32) and express  $h(n)$  in terms of  $H(k + \alpha)$ . If we multiply both sides of (8.2.32) by the exponential,  $\exp(j2\pi km/M)$ ,  $m = 0, 1, \dots, M - 1$ , and sum over  $k = 0, 1, \dots, M - 1$ , the right-hand side of (8.2.32) reduces to  $Mh(m) \exp(-j2\pi\alpha m/M)$ . Thus we obtain

$$h(n) = \frac{1}{M} \sum_{k=0}^{M-1} H(k + \alpha)e^{j2\pi(k+\alpha)n/M} \quad n = 0, 1, \dots, M - 1 \quad (8.2.33)$$

The relationship in (8.2.33) allows us to compute the values of the unit sample response  $h(n)$  from the specification of the frequency samples  $H(k + \alpha)$ ,  $k = 0, 1, \dots, M - 1$ . Note that when  $\alpha = 0$ , (8.2.32) reduces to the discrete Fourier transform (DFT) of the sequence  $\{h(n)\}$  and (8.2.33) reduces to the inverse DFT (IDFT).

Since  $\{h(n)\}$  is real, we can easily show that the frequency samples  $\{H(k + \alpha)\}$  satisfy the symmetry condition

$$H(k + \alpha) = H^*(M - k - \alpha) \quad (8.2.34)$$

This symmetry condition, along with the symmetry conditions for  $\{h(n)\}$ , can be used to reduce the frequency specifications from  $M$  points to  $(M + 1)/2$  points for  $M$  odd and  $M/2$  points for  $M$  even. Thus the linear equations for determining  $\{h(n)\}$  from  $\{H(k + \alpha)\}$  are considerably simplified.

In particular, if (8.2.11) is sampled at the frequencies  $\omega_k = 2\pi(k + \alpha)/M$ ,  $k = 0, 1, \dots, M - 1$ , we obtain

$$H(k + \alpha) = H_r\left(\frac{2\pi}{M}(k + \alpha)\right) e^{j[\beta\pi/2 - 2\pi(k+\alpha)(M-1)/2M]} \quad (8.2.35)$$

where  $\beta = 0$  when  $\{h(n)\}$  is symmetric and  $\beta = 1$  when  $\{h(n)\}$  is antisymmetric. A simplification occurs by defining a set of real frequency samples  $\{G(k + m)\}$

$$G(k + \alpha) = (-1)^k H_r \left( \frac{2\pi}{M}(k + \alpha) \right) \quad k = 0, 1, \dots, M - 1 \quad (8.2.36)$$

We use (8.2.36) in (8.2.35) to eliminate  $H_r(\omega_k)$ . Thus we obtain

$$H(k + \alpha) = G(k + \alpha) e^{j\pi k} e^{j[\beta\pi/2 - 2\pi(k + \alpha)(M - 1)/2M]} \quad (8.2.37)$$

Now the symmetry condition for  $H(k + \alpha)$  given in (8.2.34) translates into a corresponding symmetry condition for  $G(k + \alpha)$ , which can be exploited by substituting into (8.2.33), to simplify the expressions for the FIR filter impulse response  $\{h(n)\}$  for the four cases  $\alpha = 0$ ,  $\alpha = \frac{1}{2}$ ,  $\beta = 0$ , and  $\beta = 1$ . The results are summarized in Table 8.3. The detailed derivations are left as exercises for the reader.

#### 8.2.4 Design of Optimum Equiripple Linear-Phase FIR Filters

The window method and the frequency-sampling method are relatively simple techniques for designing linear-phase FIR filters. However, they also possess some minor disadvantages, described in Section 8.2.6, which may render them undesirable for some applications. A major problem is the lack of precise control of the critical frequencies such as  $\omega_p$  and  $\omega_s$ .

The filter design method described in this section is formulated as a Chebyshev approximation problem. It is viewed as an optimum design criterion in the sense that the weighted approximation error between the desired frequency response and the actual frequency response is spread evenly across the passband

and evenly across the stopband of the filter minimizing the maximum error. The resulting filter designs have ripples in both the passband and the stopband.

To describe the design procedure, let us consider the design of a lowpass filter with passband edge frequency  $\omega_p$  and stopband edge frequency  $\omega_s$ . From the general specifications given in Fig. 8.2, in the passband, the filter frequency response satisfies the condition

$$1 - \delta_1 \leq H_r(\omega) \leq 1 + \delta_1 \quad |\omega| \leq \omega_p \quad (8.2.43)$$

Similarly, in the stopband, the filter frequency response is specified to fall between the limits  $\pm\delta_2$ , that is,

$$-\delta_2 \leq H_r(\omega) \leq \delta_2 \quad |\omega| > \omega_s \quad (8.2.44)$$

Thus  $\delta_1$  represents the ripple in the passband and  $\delta_2$  represents the attenuation or ripple in the stopband. The remaining filter parameter is  $M$ , the filter length or the number of filter coefficients.

**Case 1: Symmetric unit sample response  $h(n) = h(M - 1 - n)$  and  $M$  Odd.**  
 In this case, the real-valued frequency response characteristic  $H_r(\omega)$  is

$$H_r(\omega) = h\left(\frac{M-1}{2}\right) + 2 \sum_{n=0}^{(M-3)/2} h(n) \cos \omega \left(\frac{M-1}{2} - n\right) \quad (8.2.45)$$

If we let  $k = (M - 1)/2 - n$  and define a new set of filter parameters  $\{a(k)\}$  as

$$a(k) = \begin{cases} h\left(\frac{M-1}{2}\right), & k = 0 \\ 2h\left(\frac{M-1}{2} - k\right), & k = 1, 2, \dots, \frac{M-1}{2} \end{cases} \quad (8.2.46)$$

then (8.2.45) reduces to the compact form

$$H_r(\omega) = \sum_{k=0}^{(M-1)/2} a(k) \cos \omega k \quad (8.2.47)$$

**Case 2: Symmetric unit sample response  $h(n) = h(M - 1 - n)$  and  $M$  Even.**  
 In this case,  $H_r(\omega)$  is expressed as

$$H_r(\omega) = 2 \sum_{n=0}^{(M/2)-1} h(n) \cos \omega \left(\frac{M-1}{2} - n\right) \quad (8.2.48)$$

Again, we change the summation index from  $n$  to  $k = M/2 - n$  and define a new set of filter parameters  $\{b(k)\}$  as

$$b(k) = 2h\left(\frac{M}{2} - k\right), k = 1, 2, \dots, M/2 \quad (8.2.49)$$

With these substitutions (8.2.48) becomes

$$H_r(\omega) = \sum_{k=1}^{M/2} b(k) \cos \omega \left(k - \frac{1}{2}\right) \quad (8.2.50)$$

In carrying out the optimization, it is convenient to rearrange (8.2.50) further into the form

$$H_r(\omega) = \cos \frac{\omega}{2} \sum_{k=0}^{(M/2)-1} \tilde{b}(k) \cos \omega k \quad (8.2.51)$$

where the coefficients  $\{\tilde{b}(k)\}$  are linearly related to the coefficients  $\{b(k)\}$ . In fact, it can be shown that the relationship is

$$\begin{aligned} \tilde{b}(0) &= \frac{1}{2}b(1) \\ \tilde{b}(k) &= 2b(k) - \tilde{b}(k-1) \quad k = 1, 2, 3, \dots, \frac{M}{2} - 2 \\ \tilde{b}\left(\frac{M}{2} - 1\right) &= 2b\left(\frac{M}{2}\right) \end{aligned} \quad (8.2.52)$$

**Case 3: Antisymmetric unit sample response  $h(n) = -h(M - 1 - n)$  and  $M$  Odd.** The real-valued frequency response characteristic  $H_r(\omega)$  for this case is

$$H_r(\omega) = 2 \sum_{n=0}^{(M-3)/2} h(n) \sin \omega \left( \frac{M-1}{2} - n \right) \quad (8.2.53)$$

If we change the summation in (8.2.53) from  $n$  to  $k = (M - 1)/2 - n$  and define a new set of filter parameters  $\{c(k)\}$  as

$$c(k) = 2h \left( \frac{M-1}{2} - k \right) \quad k = 1, 2, \dots, (M-1)/2 \quad (8.2.54)$$

then (8.2.53) becomes

$$H_r(\omega) = \sum_{k=1}^{(M-1)/2} c(k) \sin \omega k \quad (8.2.55)$$

As in the previous case, it is convenient to rearrange (8.2.55) into the form

$$H_r(\omega) = \sin \omega \sum_{k=0}^{(M-3)/2} \tilde{c}(k) \cos \omega k \quad (8.2.56)$$

**Case 4: Antisymmetric unit sample response  $h(n) = -h(M - 1 - n)$  and  $M$  Even.** In this case, the real-valued frequency response characteristic  $H_r(\omega)$  is

$$H_r(\omega) = 2 \sum_{n=0}^{(M/2)-1} h(n) \sin \omega \left( \frac{M-1}{2} - n \right) \quad (8.2.58)$$

A change in the summation index from  $n$  to  $k = M/2 - n$  combined with a definition of a new set of filter coefficients  $\{d(k)\}$ , related to  $\{h(n)\}$  according to

$$d(k) = 2h \left( \frac{M}{2} - k \right) \quad k = 1, 2, \dots, \frac{M}{2} \quad (8.2.59)$$

results in the expression

$$H_r(\omega) = \sum_{k=1}^{M/2} d(k) \sin \omega \left( k - \frac{1}{2} \right) \quad (8.2.60)$$

As in the previous two cases, we find it convenient to rearrange (8.2.60) into the form

$$H_r(\omega) = \sin \frac{\omega}{2} \sum_{k=0}^{(M/2)-1} \tilde{d}(k) \cos \omega k \quad (8.2.61)$$

Filter type	$Q(\omega)$	$P(\omega)$
$h(n) = h(M - 1 - n)$ $M$ odd (case 1)	1	$\sum_{k=0}^{(M-1)/2} a(k) \cos \omega k$
$h(n) = h(M - 1 - n)$ $M$ even (case 2)	$\cos \frac{\omega}{2}$	$\sum_{k=0}^{(M/2)-1} \tilde{b}(k) \cos \omega k$
$h(n) = -h(M - 1 - n)$ $M$ odd (case 3)	$\sin \omega$	$\sum_{k=0}^{(M-3)/2} \tilde{c}(k) \cos \omega k$
$h(n) = -h(M - 1 - n)$ $M$ even (case 4)	$\sin \frac{\omega}{2}$	$\sum_{k=0}^{(M/2)-1} \tilde{d}(k) \cos \omega k$

## STRUCTURES FOR FIR SYSTEMS

In general, an FIR system is described by the difference equation

$$y(n) = \sum_{k=0}^{M-1} b_k x(n-k) \quad (7.2.1)$$

or, equivalently, by the system function

$$H(z) = \sum_{k=0}^{M-1} b_k z^{-k} \quad (7.2.2)$$

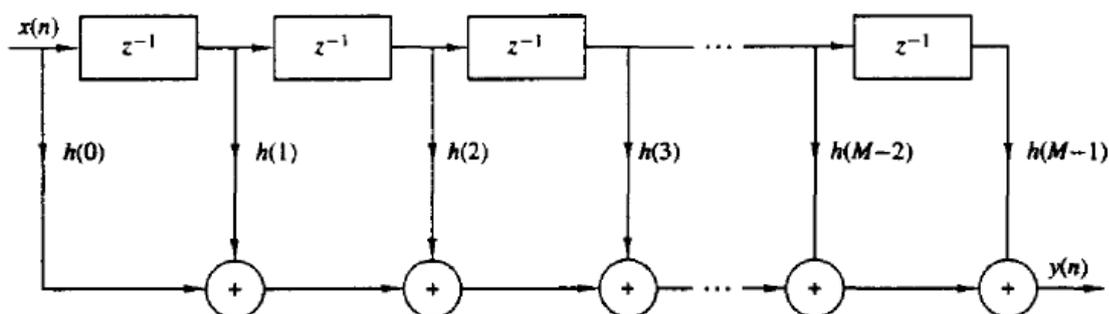
Furthermore, the unit sample response of the FIR system is identical to the coefficients  $\{b_k\}$ , that is,

$$h(n) = \begin{cases} b_n, & 0 \leq n \leq M-1 \\ 0, & \text{otherwise} \end{cases}$$

### Direct-Form Structure

The direct form realization follows immediately from the non recursive difference equation given below

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



### Cascade-Form Structures

The cascaded realization follows naturally system function given by equation. It is simple matter to factor  $H(z)$  into second order FIR system so that

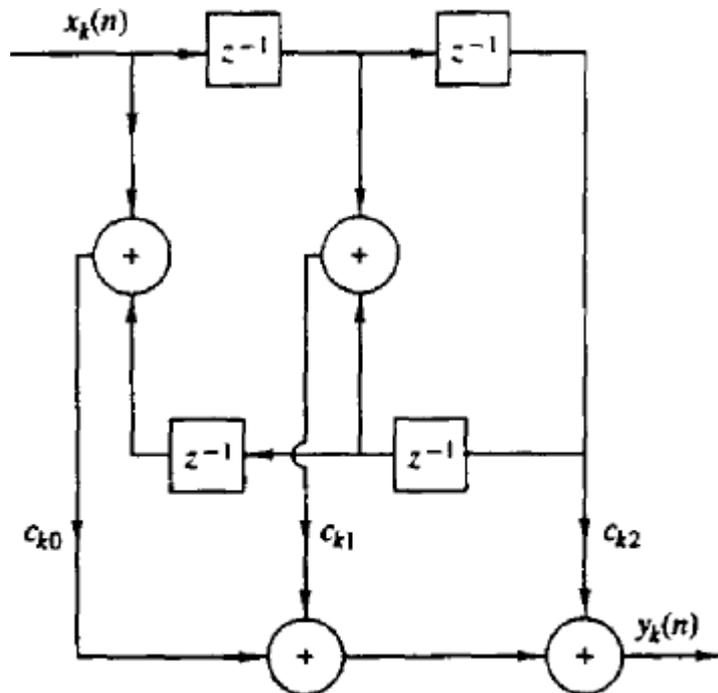
$$H(z) = \prod_{k=1}^K H_k(z)$$

where

$$H_k(z) = b_{k0} + b_{k1}z^{-1} + b_{k2}z^{-2} \quad k = 1, 2, \dots, K$$

$$H_k(z) = c_{k0}(1 - z_k z^{-1})(1 - z_k^* z^{-1})(1 - z^{-1}/z_k)(1 - z^{-1}/z_k^*)$$

$$= c_{k0} + c_{k1}z^{-1} + c_{k2}z^{-2} + c_{k1}z^{-3} + z^{-4}$$



### Frequency-Sampling Structures

The frequency-sampling realization is an alternative structure for an FIR filter in which the parameters that characterize the filter are the values of the desired frequency response instead of the impulse response  $h(n)$ . To derive the frequency sampling structure, we specify the desired frequency response at a set of equally spaced frequencies, namely

$$\omega_k = \frac{2\pi}{M}(k + \alpha) \quad k = 0, 1, \dots, \frac{M-1}{2} \quad M \text{ odd}$$

$$k = 0, 1, \dots, \frac{M}{2} - 1 \quad M \text{ even}$$

$$\alpha = 0 \text{ or } \frac{1}{2}$$

The frequency response of the system is given by

$$H(\omega) = \sum_{n=0}^{M-1} h(n)e^{-j\omega n}$$

$$H(k + \alpha) = H\left(\frac{2\pi}{M}(k + \alpha)\right)$$

$$= \sum_{n=0}^{M-1} h(n)e^{-j2\pi(k+\alpha)n/M} \quad k = 0, 1, \dots, M-1$$

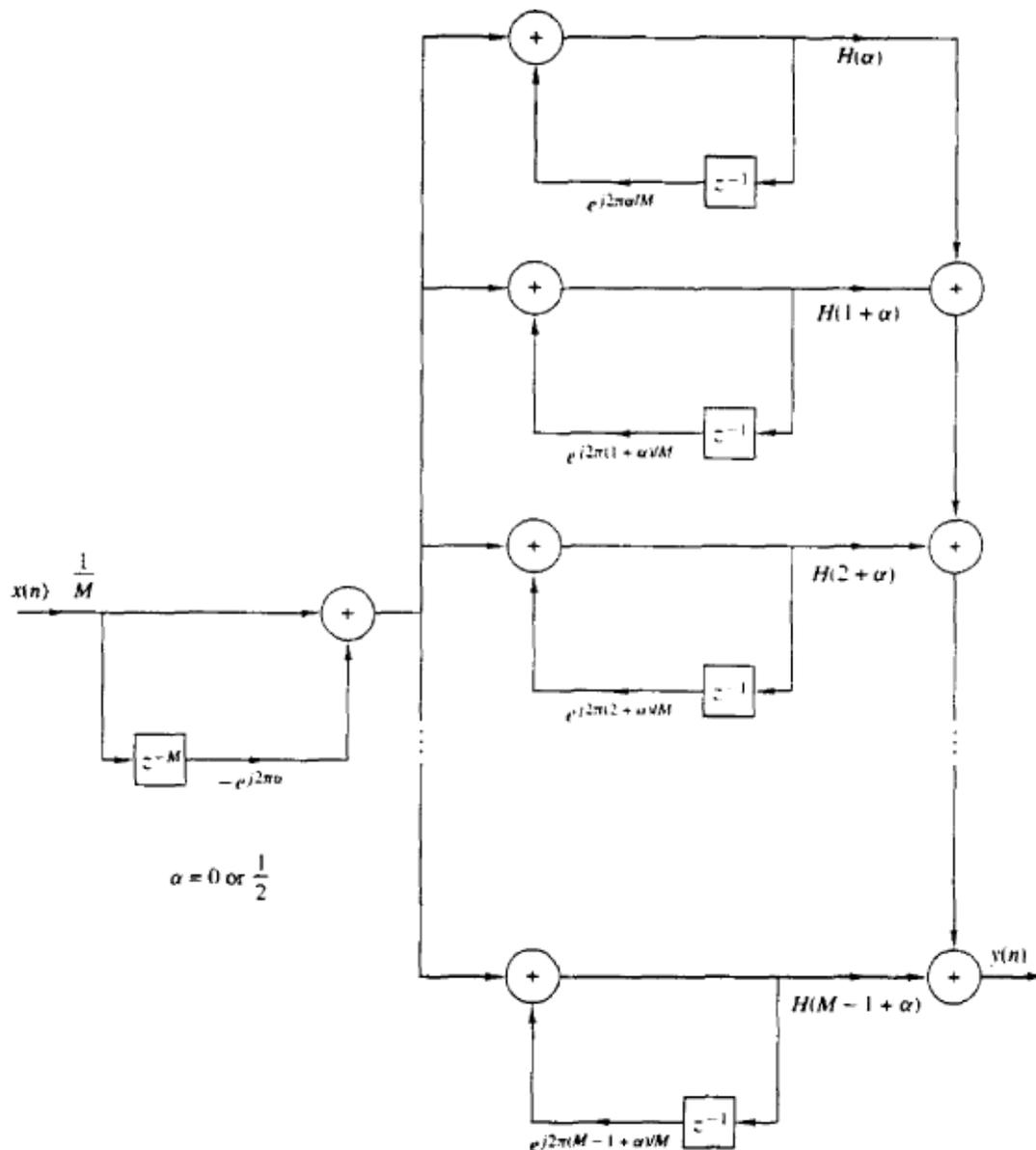
$$h(n) = \frac{1}{M} \sum_{k=0}^{M-1} H(k + \alpha)e^{j2\pi(k+\alpha)n/M} \quad n = 0, 1, \dots, M-1$$

$$H(z) = \sum_{n=0}^{M-1} h(n)z^{-n}$$

$$= \sum_{n=0}^{M-1} \left[ \frac{1}{M} \sum_{k=0}^{M-1} H(k + \alpha)e^{j2\pi(k+\alpha)n/M} \right] z^{-n}$$

$$H(z) = \sum_{k=0}^{M-1} H(k + \alpha) \left[ \frac{1}{M} \sum_{n=0}^{M-1} (e^{j2\pi(k+\alpha)/M} z^{-1})^n \right]$$

$$= \frac{1 - z^{-M} e^{j2\pi\alpha}}{M} \sum_{k=0}^{M-1} \frac{H(k + \alpha)}{1 - e^{j2\pi(k+\alpha)/M} z^{-1}}$$



### Lattice Structure

In this section we introduce another FIR filter structure, called the lattice filter or Lattice realization. Lattice filters are used extensively in digital speech processing and in the implementation of adaptive filters. Let us begin the development by considering a sequence of FIR filters with system functions

$$H_m(z) = A_m(z) \quad m = 0, 1, 2, \dots, M-1 \quad (7.2.17)$$

where, by definition,  $A_m(z)$  is the polynomial

$$A_m(z) = 1 + \sum_{k=1}^m \alpha_m(k)z^{-k} \quad m \geq 1 \quad (7.2.18)$$

and  $A_0(z) = 1$ . The unit sample response of the  $m$ th filter is  $h_m(0) = 1$  and  $h_m(k) = \alpha_m(k)$ ,  $k = 1, 2, \dots, m$ . The subscript  $m$  on the polynomial  $A_m(z)$  denotes the degree of the polynomial. For mathematical convenience, we define  $\alpha_m(0) = 1$ .

If  $\{x(n)\}$  is the input sequence to the filter  $A_m(z)$  and  $\{y(n)\}$  is the output sequence, we have

$$y(n) = x(n) + \sum_{k=1}^m \alpha_m(k)x(n-k) \quad (7.2.19)$$

Next, let us consider an FIR filter for which  $m = 2$ . In this case the output from a direct-form structure is

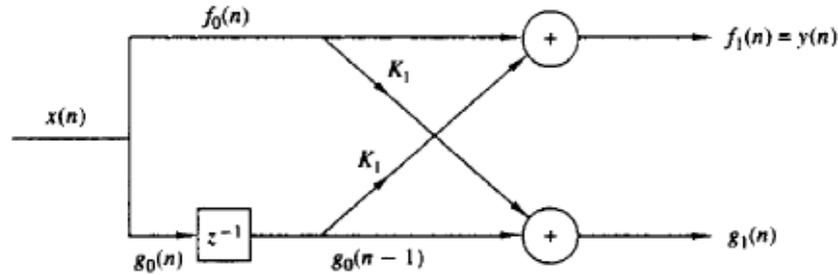
$$y(n) = x(n) + \alpha_2(1)x(n-1) + \alpha_2(2)x(n-2) \quad (7.2.22)$$

By cascading two lattice stages as shown in Fig. 7.10, it is possible to obtain the same output as (7.2.22). Indeed, the output from the first stage is

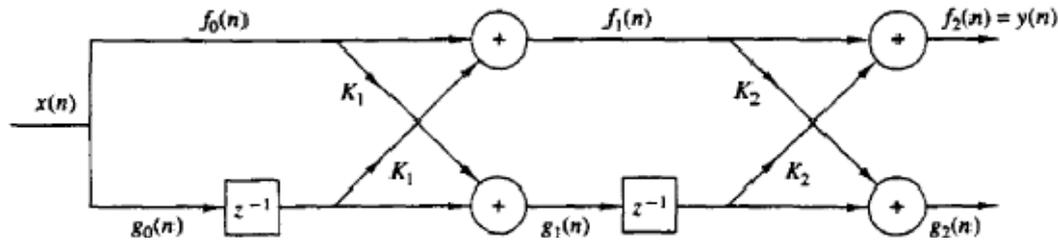
$$\begin{aligned} f_1(n) &= x(n) + K_1x(n-1) \\ g_1(n) &= K_1x(n) + x(n-1) \end{aligned} \quad (7.2.23)$$

The output from the second stage is

$$\begin{aligned} f_2(n) &= f_1(n) + K_2g_1(n-1) \\ g_2(n) &= K_2f_1(n) + g_1(n-1) \end{aligned} \quad (7.2.24)$$



$$\begin{aligned} f_0(n) &= g_0(n) = x(n) \\ f_1(n) &= f_0(n) + K_1g_0(n-1) = x(n) + K_1x(n-1) \\ g_1(n) &= K_1f_0(n) + g_0(n-1) = K_1x(n) + x(n-1) \end{aligned}$$



$$\begin{aligned} f_2(n) &= x(n) + K_1x(n-1) + K_2[K_1x(n-1) + x(n-2)] \\ &= x(n) + K_1(1 + K_2)x(n-1) + K_2x(n-2) \end{aligned}$$

The general form of lattice structure for  $m$  stage is given by

$$f_0(n) = g_0(n) = x(n)$$

$$f_m(n) = f_{m-1}(n) + K_m g_{m-1}(n-1) \quad m = 1, 2, \dots, M-1$$

$$g_m(n) = K_m f_{m-1}(n) + g_{m-1}(n-1) \quad m = 1, 2, \dots, M-1$$

**Conversion of lattice coefficients to direct-form filter coefficients.** The direct-form FIR filter coefficients  $\{\alpha_m(k)\}$  can be obtained from the lattice coefficients  $\{K_i\}$  by using the following relations:

$$A_0(z) = B_0(z) = 1 \quad (7.2.47)$$

$$A_m(z) = A_{m-1}(z) + K_m z^{-1} B_{m-1}(z) \quad m = 1, 2, \dots, M-1 \quad (7.2.48)$$

$$B_m(z) = z^{-m} A_m(z^{-1}) \quad m = 1, 2, \dots, M-1 \quad (7.2.49)$$

### Conversion of direct-form FIR filter coefficients to lattice coefficients.

Suppose that we are given the FIR coefficients for the direct-form realization or, equivalently, the polynomial  $A_m(z)$ , and we wish to determine the corresponding lattice filter parameters  $\{K_i\}$ . For the  $m$ -stage lattice we immediately obtain the parameter  $K_m = \alpha_m(m)$ . To obtain  $K_{m-1}$  we need the polynomials  $A_{m-1}(z)$  since, in general,  $K_m$  is obtained from the polynomial  $A_m(z)$  for  $m = M-1, M-2, \dots, 1$ . Consequently, we need to compute the polynomials  $A_m(z)$  starting from  $m = M-1$  and "stepping down" successively to  $m = 1$ .

$$\begin{aligned} K_m &= \alpha_m(m) \quad \alpha_{m-1}(0) = 1 \\ \alpha_{m-1}(k) &= \frac{\alpha_m(k) - K_m \beta_m(k)}{1 - K_m^2} \\ &= \frac{\alpha_m(k) - \alpha_m(m) \alpha_m(m-k)}{1 - \alpha_m^2(m)} \quad 1 \leq k \leq m-1 \end{aligned}$$

## UNIT – V

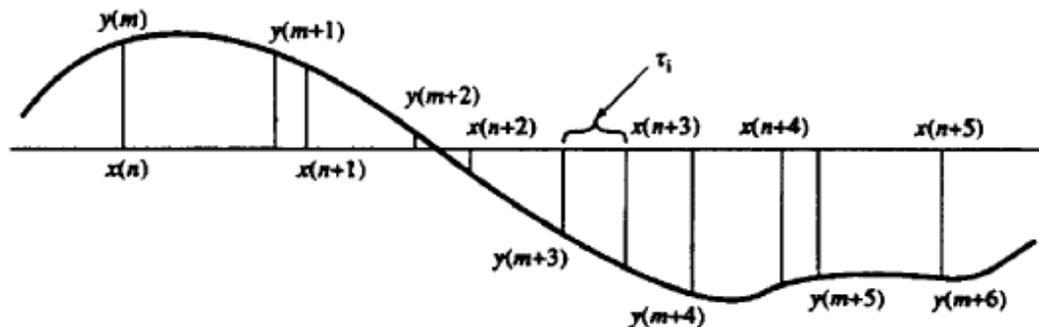
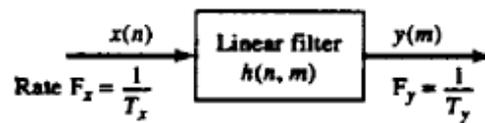
### INTRODUCTION

The process of sampling rate conversion in the digital domain can be viewed as a linear filtering operation, as illustrated in Fig. 10.1(a). The input signal  $x(n]$  is characterized by the sampling rate  $F_x = 1/T_x$  and the output signal  $y(m]$  is characterized by the sampling rate  $F_y = 1/T_y$ , where  $T_x$  and  $T_y$  are the corresponding sampling intervals. In the main part of our treatment, the ratio  $F_y/F_x$  is constrained to be rational,

$$\frac{F_y}{F_x} = \frac{I}{D}$$

where  $D$  and  $I$  are relatively prime integers. We shall show that the linear filter is characterized by a time-variant impulse response, denoted as  $h(n, m]$ . Hence the input  $x(n]$  and the output  $y(m]$  are related by the convolution summation for time-variant systems.

The sampling rate conversion process can also be understood from the point of view of digital resampling of the same analog signal. Let  $x(t]$  be the analog signal that is sampled at the first rate  $F_x$  to generate  $x(n]$ . The goal of rate conversion is to obtain another sequence  $y(m]$  directly from  $x(n]$ , which is equal to the sampled values of  $x(t]$  at a second rate  $F_y$ . As is depicted in Fig. 10.1(b),  $y(m]$  is a time-shifted version of  $x(n]$ . Such a time shift can be



## DECIMATION BY A FACTOR $D$

Let us assume that the signal  $x(n)$  with spectrum  $X(\omega)$  is to be downsampled by an integer factor  $D$ . The spectrum  $X(\omega)$  is assumed to be nonzero in the frequency interval  $0 \leq |\omega| \leq \pi$  or, equivalently,  $|F| \leq F_x/2$ . We know that if we reduce the sampling rate simply by selecting every  $D$ th value of  $x(n)$ , the resulting signal will be an aliased version of  $x(n)$ , with a folding frequency of  $F_x/2D$ . To avoid aliasing, we must first reduce the bandwidth of  $x(n)$  to  $F_{\max} = F_x/2D$  or, equivalently, to  $\omega_{\max} = \pi/D$ . Then we may downsample by  $D$  and thus avoid aliasing.

The decimation process is illustrated in Fig. 10.2. The input sequence  $x(n)$  is passed through a lowpass filter, characterized by the impulse response  $h(n)$  and a frequency response  $H_D(\omega)$ , which ideally satisfies the condition

$$H_D(\omega) = \begin{cases} 1, & |\omega| \leq \pi/D \\ 0, & \text{otherwise} \end{cases} \quad (10.2.1)$$

Thus the filter eliminates the spectrum of  $X(\omega)$  in the range  $\pi/D < \omega < \pi$ . Of course, the implication is that only the frequency components of  $x(n)$  in the range  $|\omega| \leq \pi/D$  are of interest in further processing of the signal.

The output of the filter is a sequence  $v(n)$  given as

$$v(n) = \sum_{k=0}^{\infty} h(k)x(n-k) \quad (10.2.2)$$

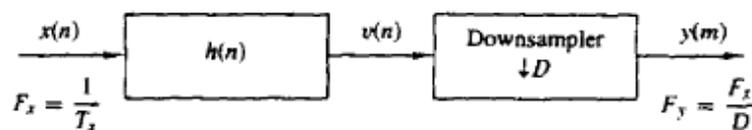


Figure 10.2 Decimation by a factor  $D$ .



which is then downsampled by the factor  $D$  to produce  $y(m)$ . Thus

$$\begin{aligned} y(m) &= v(mD) \\ &= \sum_{k=0}^{\infty} h(k)x(mD - k) \end{aligned} \quad (10.2.3)$$

Although the filtering operation on  $x(n)$  is linear and time invariant, the downsampling operation in combination with the filtering results in a time-variant system. This is easily verified. Given the fact that  $x(n)$  produces  $y(m)$ , we note that  $x(n - n_0)$  does not imply  $y(n - n_0)$  unless  $n_0$  is a multiple of  $D$ . Consequently, the overall linear operation (linear filtering followed by downsampling) on  $x(n)$  is not time invariant.

The frequency-domain characteristics of the output sequence  $y(m)$  can be obtained by relating the spectrum of  $y(m)$  to the spectrum of the input sequence  $x(n)$ . First, it is convenient to define a sequence  $\tilde{v}(n)$  as

$$\tilde{v}(n) = \begin{cases} v(n), & n = 0, \pm D, \pm 2D, \dots \\ 0, & \text{otherwise} \end{cases} \quad (10.2.4)$$

Clearly,  $\tilde{v}(n)$  can be viewed as a sequence obtained by multiplying  $v(n)$  with a periodic train of impulses  $p(n)$ , with period  $D$ , as illustrated in Fig. 10.3. The discrete Fourier series representation of  $p(n)$  is

$$p(n) = \frac{1}{D} \sum_{k=0}^{D-1} e^{j2\pi kn/D} \quad (10.2.5)$$

Hence

$$\tilde{v}(n) = v(n)p(n) \quad (10.2.6)$$

and

$$y(m) = \tilde{v}(mD) = v(mD)p(mD) = v(mD) \quad (10.2.7)$$

Now the  $z$ -transform of the output sequence  $y(m)$  is

$$\begin{aligned} Y(z) &= \sum_{m=-\infty}^{\infty} y(m)z^{-m} \\ &= \sum_{m=-\infty}^{\infty} \tilde{v}(mD)z^{-m} \end{aligned} \quad (10.2.8)$$

where the last step follows from the fact that  $\tilde{v}(m) = 0$ , except at multiples of  $D$ . By making use of the relations in (10.2.5) and (10.2.6) in (10.2.8), we obtain

## INTERPOLATION BY A FACTOR $I$

An increase in the sampling rate by an integer factor of  $I$  can be accomplished by interpolating  $I - 1$  new samples between successive values of the signal. The interpolation process can be accomplished in a variety of ways. We shall describe a process that preserves the spectral shape of the signal sequence  $x(n)$ .

Let  $v(m)$  denote a sequence with a rate  $F_y = IF_x$ , which is obtained from  $x(n)$  by adding  $I - 1$  zeros between successive values of  $x(n)$ . Thus

$$v(m) = \begin{cases} x(m/I), & m = 0, \pm I, \pm 2I, \dots \\ 0, & \text{otherwise} \end{cases} \quad (10.3.1)$$

and its sampling rate is identical to the rate of  $y(m)$ . This sequence has a  $z$ -transform

$$\begin{aligned} V(z) &= \sum_{m=-\infty}^{\infty} v(m)z^{-m} \\ &= \sum_{m=-\infty}^{\infty} x(m/I)z^{-m} \\ &= X(z^I) \end{aligned} \quad (10.3.2)$$

The corresponding spectrum of  $v(m)$  is obtained by evaluating (10.3.2) on the unit circle. Thus

$$V(\omega_y) = X(\omega_y I) \quad (10.3.3)$$

where  $\omega_y$  denotes the frequency variable relative to the new sampling rate  $F_y$  (i.e.,  $\omega_y = 2\pi F/F_y$ ). Now the relationship between sampling rates is  $F_y = IF_x$  and hence, the frequency variables  $\omega_x$  and  $\omega_y$  are related according to the formula

$$\omega_y = \frac{\omega_x}{I} \quad (10.3.4)$$

The spectra  $X(\omega_x)$  and  $V(\omega_y)$  are illustrated in Fig. 10.5. We observe that the sampling rate increase, obtained by the addition of  $I - 1$  zero samples between successive values of  $x(n)$ , results in a signal whose spectrum  $V(\omega_y)$  is an  $I$ -fold periodic repetition of the input signal spectrum  $X(\omega_x)$ .

Since only the frequency components of  $x(n)$  in the range  $0 \leq \omega_y \leq \pi/I$  are unique, the images of  $X(\omega)$  above  $\omega_y = \pi/I$  should be rejected by passing the sequence  $v(m)$  through a lowpass filter with frequency response  $H_I(\omega_y)$  that ideally has the characteristic

$$H_I(\omega_y) = \begin{cases} C, & 0 \leq |\omega_y| \leq \pi/I \\ 0, & \text{otherwise} \end{cases} \quad (10.3.5)$$

where  $C$  is a scale factor required to properly normalize the output sequence  $y(m)$ . Consequently, the output spectrum is

$$Y(\omega_y) = \begin{cases} CX(\omega_y I), & 0 \leq |\omega_y| \leq \pi/I \\ 0, & \text{otherwise} \end{cases} \quad (10.3.6)$$

We conclude that the spectrum  $Y(\omega_y)$ , which is obtained by evaluating (10.2.9) on the unit circle, can be expressed as

$$Y(\omega_y) = \frac{1}{D} \sum_{k=0}^{D-1} H_D \left( \frac{\omega_y - 2\pi k}{D} \right) X \left( \frac{\omega_y - 2\pi k}{D} \right) \quad (10.2.14)$$

With a properly designed filter  $H_D(\omega)$ , the aliasing is eliminated and, consequently, all but the first term in (10.2.14) vanish. Hence

$$\begin{aligned} Y(\omega_y) &= \frac{1}{D} H_D \left( \frac{\omega_y}{D} \right) X \left( \frac{\omega_y}{D} \right) \\ &= \frac{1}{D} X \left( \frac{\omega_y}{D} \right) \end{aligned} \quad (10.2.15)$$

for  $0 \leq |\omega_y| \leq \pi$ . The spectra for the sequences  $x(n)$ ,  $v(n)$ , and  $y(m)$  are illustrated in Fig. 10.4.

ideally has the characteristic

$$H_I(\omega_y) = \begin{cases} C, & 0 \leq |\omega_y| \leq \pi/I \\ 0, & \text{otherwise} \end{cases} \quad (10.3.5)$$

where  $C$  is a scale factor required to properly normalize the output sequence  $y(m)$ . Consequently, the output spectrum is

$$Y(\omega_y) = \begin{cases} CX(\omega_y/I), & 0 \leq |\omega_y| \leq \pi/I \\ 0, & \text{otherwise} \end{cases} \quad (10.3.6)$$

The scale factor  $C$  is selected so that the output  $y(m) = x(m/I)$  for  $m = 0, \pm I, +2I, \dots$ . For mathematical convenience, we select the point  $m = 0$ . Thus

$$\begin{aligned} y(0) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} Y(\omega_y) d\omega_y \\ &= \frac{C}{2\pi} \int_{-\pi/I}^{\pi/I} X(\omega_y/I) d\omega_y \end{aligned} \quad (10.3.7)$$

Since  $\omega_y = \omega_x/I$ , (10.3.7) can be expressed as

$$\begin{aligned} y(0) &= \frac{C}{I} \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega_x) d\omega_x \\ &= \frac{C}{I} x(0) \end{aligned} \quad (10.3.8)$$

Therefore,  $C = I$  is the desired normalization factor.

Finally, we indicate that the output sequence  $y(m)$  can be expressed as a convolution of the sequence  $v(n)$  with the unit sample response  $h(n)$  of the lowpass

filter. Thus

$$y(m) = \sum_{k=-\infty}^{\infty} h(m-k)v(k) \quad (10.3.9)$$

Since  $v(k) = 0$  except at multiples of  $I$ , where  $v(kI) = x(k)$ , (10.3.9) becomes

$$y(m) = \sum_{k=-\infty}^{\infty} h(m-kI)x(k) \quad (10.3.10)$$