# Department of Electronics & Communication Engineering

## Course File

# DIGITAL SIGNAL PROCESSING

# (18PC0EC16)

**III B Tech – II Semester [Branch: ECE]**

**Academic Year: 2022-2023**

**Guru Nanak Institutions Technical Campus (Autonomous)**
**School of Engineering and Technology**
**Ibrahimpatnam, R R District – 501 506 (T.S.)**

**GURU NANAK INSTITUTIONS TECHNICAL CAMPUS (AUTONOMOUS)**

**HYDERABAD**

| **III Year B.Tech. ECE. II-Sem** | **L** | **T** | **P** | **C** |
|---|---|---|---|---|
| | 3 | 1 | 0 | 4 |

**(18PC0EC16) DIGITAL SIGNAL PROCESSING**

**UNIT I: INTRODUCTION TO DISCRETE TIME SIGNALS AND SYSTEMS**

Discrete time signals: Sequences; representation of signals on orthogonal basis; Sampling and reconstruction of signals; Discrete systems attributes, Application of DSP, Z-Transform, Analysis of LSI systems, frequency Analysis, Inverse Systems

**UNIT II: DISCRETE FOURIER TRANSFORM AND FAST FOURIER TRANSFORM**

Discrete Fourier Transforms: Properties of DFT, Linear Convolution of Sequences using DFT, Computation of Over-Lap Add Method, Over-Lap Save Method.

Fast Fourier Transforms: Fast Fourier Transforms (FFT) – Radix-2 Decimation-in-Time and Decimation-in-Frequency FFT Algorithms, Inverse FFT, and FFT with General Radix-N.

**UNIT III: FIR DIGITAL FILTERS**

Characteristics of FIR Digital Filters, Frequency Response, Design of FIR Digital filters: Fourier Method, Window method, Park-McClellan's method.

**UNIT IV: IIR FILTERS**

Design of IIR analog filters – Butterworth and Chebyshev, Elliptic Approximations, Design of IIR Digital Filters from Analog Filters by using Impulse Invariant Techniques, Bilinear Transformation Method, Spectral Transformation: Low pass, Band pass, Band stop and High pass filters. Comparison of FIR and IIR filters.

**UNIT V: FINITE WORD LENGTH EFFECTS AND MULTIRATE SIGNAL PROCESSING**

Effect of finite register length in FIR filter design. Parametric and non-parametric spectral estimation. Introduction to multirate signal processing, Decimation, Interpolation, Sampling Rate Conversion

# UNIT I: INTRODUCTION TO DIGITAL SIGNAL PROCESSING

## 1.1 INTRODUCTION

Signals constitute an important part of our daily life. Anything that carries some information is called a signal. A signal is defined as a single-valued function of one or more independent variables which contain some information. A signal is also defined as a physical quantity that varies with time, space or any other independent variable. A signal may be represented in time domain or frequency domain. Human speech is a familiar example of a signal. Electric current and voltage are also examples of signals. A signal can be a function of one or more independent variables. A signal may be a function of time, temperature, position, pressure, distance etc. If a signal depends on only one independent variable, it is called a one- dimensional signal, and if a signal depends on two independent variables, it is called a two- dimensional signal.

A system is defined as an entity that acts on an input signal and transforms it into an output signal. A system is also defined as a set of elements or fundamental blocks which are connected together and produces an output in response to an input signal. It is a cause-and- effect relation between two or more signals. The actual physical structure of the system determines the exact relation between the input $x(n)$ and the output $y(n)$, and specifies the output for every input. Systems may be single-input and single-output systems or multi-input and multi-output systems.

Signal processing is a method of extracting information from the signal which in turn depends on the type of signal and the nature of information it carries. Thus signal processing is concerned with representing signals in the mathematical terms and extracting information by carrying out algorithmic operations on the signal. Digital signal processing has many advantages over analog signal processing. Some of these are as follows:

Digital circuits do not depend on precise values of digital signals for their operation. Digital circuits are less sensitive to changes in component values. They are also less sensitive to variations in temperature, ageing and other external parameters.

In a digital processor, the signals and system coefficients are represented as binary words. This enables one to choose any accuracy by increasing or decreasing the number of bits in the binary word.

Digital processing of a signal facilitates the sharing of a single processor among a number of signals by time sharing. This reduces the processing cost per signal.

Digital implementation of a system allows easy adjustment of the processor characteristics during processing.

Linear phase characteristics can be achieved only with digital filters. Also multirate processing is possible only in the digital domain. Digital circuits can be connected in cascade without any loading problems, whereas this cannot be easily done with analog circuits.

Storage of digital data is very easy. Signals can be stored on various storage media such as magnetic tapes, disks and optical disks without any loss. On the other hand, stored analog signals deteriorate rapidly as time progresses and cannot be recovered in their original form.

Digital processing is more suited for processing very low frequency signals such as seismic signals.

Though the advantages are many, there are some drawbacks associated with processing a signal in digital domain. Digital processing needs ‘pre’ and ‘post’ processing devices like analog-to-digital and digital-to-analog converters and associated reconstruction filters. This increases the complexity of the digital system. Also, digital techniques suffer from frequency limitations. Digital systems are constructed using active devices which consume power whereas analog processing algorithms can be implemented using passive devices which do not consume power. Moreover, active devices are less reliable than passive components. But the advantages of digital processing techniques outweigh the disadvantages in many applications. Also the cost of DSP hardware is decreasing continuously. Consequently, the applications of digital signal processing are increasing rapidly.

The digital signal processor may be a large programmable digital computer or a small microprocessor programmed to perform the desired operations on the input signal. It may also be a hardwired digital processor configured to perform a specified set of operations on the input signal.

DSP has many applications. Some of them are: Speech processing, Communication, Biomedical, Consumer electronics, Seismology and Image processing.

The block diagram of a DSP system is shown in Figure 1.1.

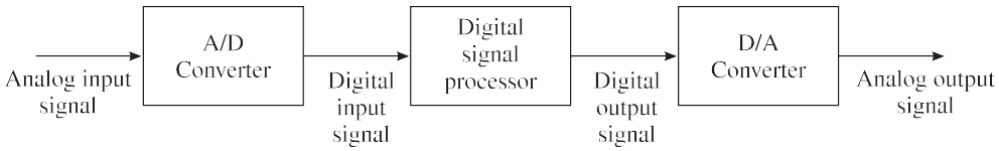



Figure 1.1 Block diagram of a digital signal processing system.

In this book we discuss only about discrete one-dimensional signals and consider only single-input and single-output discrete-time systems. In this chapter, we discuss about various basic discrete-time signals available, various operations on discrete-time signals and classification of discrete-time signals and discrete-time systems.

### *1.2 REPRESENTATION OF DISCRETE-TIME SIGNALS*

Discrete-time signals are signals which are defined only at discrete instants of time. For those signals, the amplitude between the two time instants is just not defined. For discrete- time signal the independent variable is time *n*, and it is represented by x (n).

There are following four ways of representing discrete-time signals:

1. Graphical representation
2. Functional representation
3. Tabular representation
4. Sequence representation

### 1.2.1 Graphical Representation

Consider a single x (n) with values

X (-2) = -3, x(-1) = 2, x(0) = 0, x(1) = 3, x(2) = 1 and x(3) = 2

This discrete-time single can be represented graphically as shown in Figure 1.2

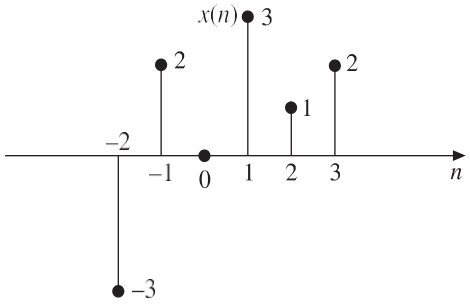



Figure 1.2 Graphical representation of discrete-time signal

### *1.2.2 Functional Representation*

In this, the amplitude of the signal is written against the values of *n.* The signal given in section 1.2.1 can be represented using the functional representation as follows:

$$x(n) = \begin{cases} -3 & \text{for } n = -2 \\ 2 & \text{for } n = -1 \\ 0 & \text{for } n = 0 \\ 3 & \text{for } n = 1 \\ 1 & \text{for } n = 2 \\ 2 & \text{for } n = 3 \end{cases}$$

Another example is:

$$X(n) = 2^n u(n)$$

Or $$x(n) = \begin{cases} 2^n \ for \ n \geq 0 \\ 0 \ \ for \ \ n < 0 \end{cases}$$

### 1.2.3 Tabular Representation

In this, the sampling instant *n* and the magnitude of the signal at the sampling instant are represented in the tabular form. The signal given in section 1.2.1 can be represented in tabular form as follows:

| *n* | 2 | 1 | 0 | 1 | 2 | 3 |
|---|---|---|---|---|---|---|
| *x* (*n*) | 3 | 2 | 0 | 3 | 1 | 2 |

### *1.2.4 Sequence Representation*

A finite duration sequence given in section 1.2.1 can be represented as follows:

$$X(n) = \left\{ \underset{\uparrow}{-3,2,0,3,1,2} \right\}$$

Another example is:

$$X(n) = \left\{ \underset{\uparrow}{\cdots 2,3,0,1,-2 \cdots} \right\}$$

The arrow mark ↑ denotes the $n = 0$ term. When no arrow is indicated, the first term corresponds to $n = 0$.

So a finite duration sequence, that satisfies the condition $x(n) = 0$ for $n < 0$ can be represented as:

$$x(n) = \{3, 5, 2, 1, 4, 7\}$$

### Sum and product of discrete-time sequences

The sum of two discrete-time sequences is obtained by adding the corresponding elements of sequences

$$\{C_n\} = \{a_n\} + \{b_n\} \rightarrow C_n = a_n + b_n$$

The product of two discrete-time sequences is obtained by multiplying the corresponding elements of the sequences.

$$\{C_n\} = \{a_n\}\{b_n\} \rightarrow C_n = a_n b_n$$

The multiplication of a sequence by a constant *k* is obtained by multiplying each element of the sequence by that constant.

$$\{C_n\} = k\{a_n\} \rightarrow C_n = ka_n$$

## *1.3 ELEMENTARY DISCRETE-TIME SIGNALS*

There are several elementary signals which play vital role in the study of signals and systems. These elementary signals serve as basic building blocks for the construction of more complex signals. Infact, these elementary signals may be used to model a large number of physical signals, which occur in nature. These elementary signals are also called standard signals.

The standard discrete-time signals are as follows:

1. Unit step sequence
2. Unit ramp sequence
3. Unit parabolic sequence
4. Unit impulse sequence
5. Sinusoidal sequence
6. Real exponential sequence
7. Complex exponential sequence

### 1.3.1 Unit Step Sequence

The step sequence is an important signal used for analysis of many discrete-time systems. It exists only for positive time and is zero for negative time. It is equivalent to applying a signal whose amplitude suddenly changes and remains constant at the sampling instants forever after application. In between the discrete instants it is zero. If a step function has unity magnitude, then it is called unit step function.

The usefulness of the unit-step function lies in the fact that if we want a sequence to start at $n = 0$, so that it may have a value of zero for $n < 0$, we only need to multiply the given sequence with unit step function $u(n)$.

The discrete-time unit step sequence $u(n)$ is defined as:

$$U(n) = \begin{cases} 1 \; for \; n \geq 0 \\ 0 \; for \; n < 0 \end{cases}$$

The shifted version of the discrete-time unit step sequence $u(n-k)$ is defined as:

$$U(n-k) = \begin{cases} 1 \; for \; n \geq k \\ 0 \; for \; n < k \end{cases}$$

It is zero if the argument $(n-k) < 0$ and equal to 1 if the argument $(n-k) \geq 0$.

The graphical representation of $u(n)$ and $u(n-k)$ is shown in Figure 1.3[(a) and (b)].


Figure 1.3 Discrete-time (a) Unit step function (b) Shifted unit step function

#### *1.3.2 Unit Ramp Sequence*

The discrete-time unit ramp sequence $r(n)$ is that sequence which starts at $n = 0$ and increases linearly with time and is defined as:

$$r(n) = \begin{cases} n \; for \; n \geq 0 \\ 0 \; for \; n < 0 \end{cases}$$

or $$r(n) = nu(n)$$

It starts at $n = 0$ and increases linearly with $n$.

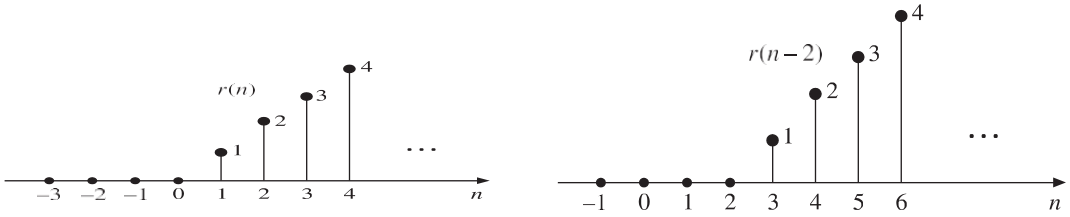
The shifted version of the discrete-time unit ramp sequence $r(n-k)$ is defined as:

$$R(n-k) = \begin{cases} n-k \; for \; n \geq k \\ 0 \quad for \; n < k \end{cases}$$

Or $$r(n-k) = (n-k)\, u(n-k)$$

The graphical representation of $r(n)$ and $r(n-2)$ is shown in Figure 1.4[(a) and (b)].

Figure 1.4 Discrete-time (a) Unit ramp sequence (b) Shifted ramp sequence.




### *1.3.3 Unit Parabolic Sequence*

The discrete-time unit parabolic sequence $p(n)$ is defined as:

$$P(n) = \begin{cases} \frac{n^2}{2} \; for \; n \geq 0 \\ 0 \; for \; n < 0 \end{cases}$$

Or $$P(n) = \frac{N^2}{2} u(n)$$

The shifted version of the discrete-time unit parabolic sequence $p(n-k)$ is defined as:

$$P(n-k) = \begin{cases} \frac{(n-k)^2}{2} \; for \; n \geq k \\ 0 \quad for \; n < k \end{cases}$$

Or $$p(n-k) = \frac{(n-k)^2}{2} u(n-k)$$

The graphical representation of $p(n)$ and $p(n-3)$ is shown in Figure 1.5[(a) and (b)].

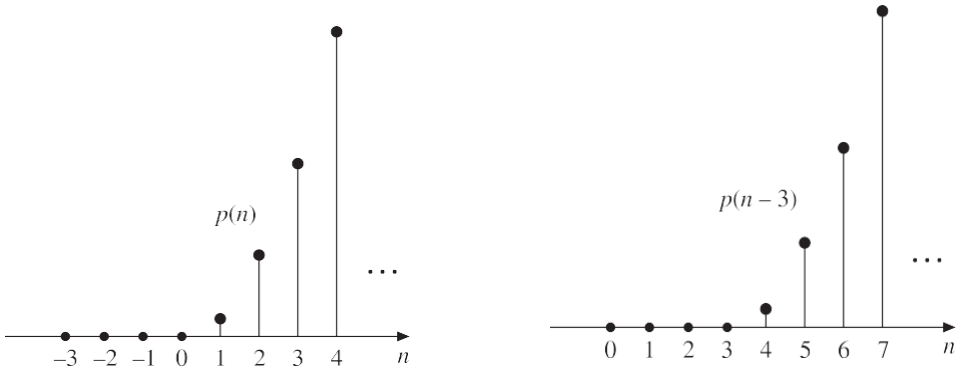



Figure 1.5 Discrete-time (a) Parabolic sequence (b) Shifted parabolic sequence.

## *1.3.4 Unit Impulse Function or Unit Sample Sequence*

The discrete-time unit impulse function $(n)$, also called unit sample sequence, is defined as:

$$\delta(n) = \begin{cases} 1 \; for \; n = 0 \\ 0 \; for \; n \neq 0 \end{cases}$$

This means that the unit sample sequence is a signal that is zero everywhere, except at $n = 0$, where its value is unity. It is the most widely used elementary signal used for the analysis of signals and systems.

The shifted unit impulse function $(n-k)$ is defined as:

$$\delta(n-k) = \begin{cases} 1 \; for \; n = k \\ 0 \; for \; n \neq k \end{cases}$$

The graphical representation of $(n)$ and $(n-k)$ is shown in Figure 1.6[(a) and (b)].

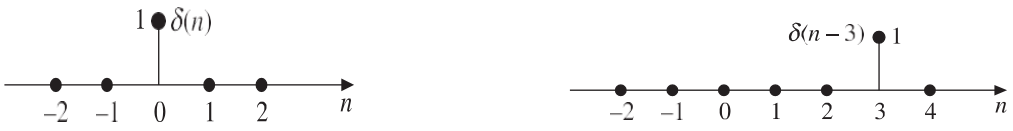



Figure 1.6 Discrete-time (a) Unit sample sequence (b) Delayed unit sample sequence.

**Properties of discrete-time unit sample sequence**

1. $\delta(n) = u(n) - u(n-1)$
2. $\delta(n-k) = \begin{cases} 1 \; for \; n = k \\ 0 \; for \; n \neq k \end{cases}$
3. $X(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k)$
4. $\sum_{n=-\infty}^{\infty} x(n)\delta(n-n_0) = x(n_0)$

**Relation Between The Unit Sample Sequence And The Unit Step Sequence**

The unit sample sequence $\delta(n)$ and the unit step sequence $u(n)$ are related as:

$$U(n) = \sum_{m=0}^{n} \delta(m), \; \delta(n) = u(n) - u(n-1)$$

## Sinusoidal Sequence

The discrete-time sinusoidal sequence is given by

$$X(n) = A \sin(\omega n + \emptyset)$$

Where $A$ is the amplitude, is angular frequency, is phase angle in radians and $n$ is an integer.

The period of the discrete-time sinusoidal sequence is:

$$N = \frac{2\pi}{\omega} m$$

Where $N$ and $m$ are integers.

All continuous-time sinusoidal signals are periodic, but discrete-time sinusoidal sequences may or may not be periodic depending on the value of.

For a discrete-time signal to be periodic, the angular frequency must be a rational multiple of 2. The graphical representation of a discrete-time sinusoidal signal is shown in Figure 1.7.

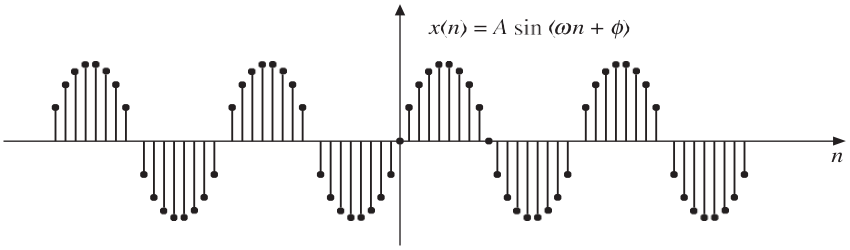



Figure 1.7 Discrete-time sinusoidal signal

### 1.3.6 Real Exponential Sequence

The discrete-time real exponential sequence $a^n$ is defined as:

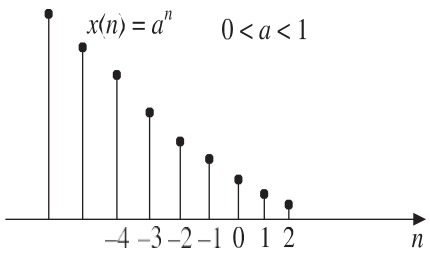
$$X(n) = a^n \text{ for all } n$$

Figure 1.8 illustrates different types of discrete-time exponential signals.

When $a > 1$, the sequence grows exponentially as shown in Figure 1.8(a).

When $0 < a < 1$, the sequence decays exponentially as shown in Figure 1.8(b).

When $a < 0$, the sequence takes alternating signs as shown in Figure 1.8[(c) and




(d)].

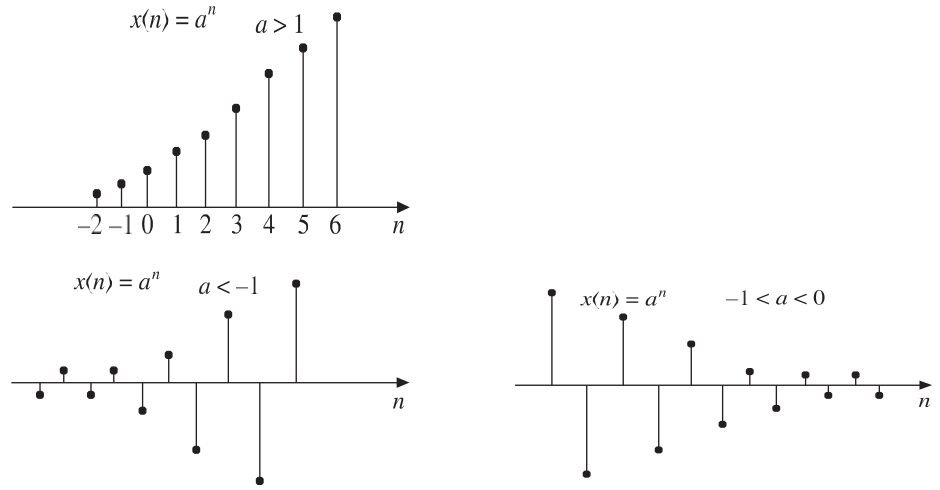



**Figure 1.8 Discrete-time exponential signal an for (a) a > 1 (b) 0 < a < 1 (c) a < -1 (d) -1 < a < 0.**

### *1.3.7 Complex Exponential Sequence*

The discrete-time complex exponential sequence is defined as:

$$X(n) = a^n e^{j(\omega_0 n + \emptyset)}$$

$$= an \cos(\omega_0 n + \emptyset) + ja^n \sin(\omega_0 n = \emptyset)$$

For $|a| = 1$, the real and imaginary parts of complex exponential sequence are sinusoidal.

For $|a| > 1$, the amplitude of the sinusoidal sequence exponentially grows as shown in Figure 1.9(a).

For $|a| < 1$, the amplitude of the sinusoidal sequence exponentially decays as shown in Figure 1.9(b).

**EXAMPLE 1.1** Find the following summations:

(a) $\sum_{n=-\infty}^{\infty} e^{3n}\,\delta\,(n-3)$ (b) $\sum_{n=-\infty}^{\infty} \delta\,(n-2)\cos 3n$

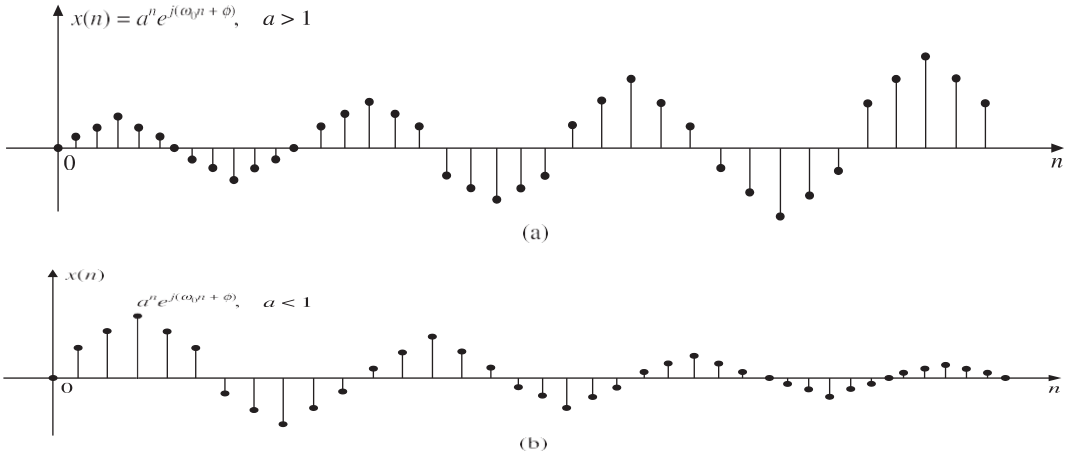



**Figure 1.9** complex exponential sequence $x(n) = a^n e^{j(\omega_0 n + \emptyset)}$ for (a) $a > 1$ (b) $a < 1$.

(c) $\sum_{n=-\infty}^{\infty} n^2\,\delta\,(n+4)$ (d) $\sum_{n=-\infty}^{\infty} \delta\ (n-2)\,e^{n^2}$

(e) $\sum_{n=0}^{\infty} \delta\,(n+1)\ 4^n$

**Solution:**

**(a) Given** $\sum_{n=-\infty}^{\infty} \boldsymbol{\delta^{3n}}\ \boldsymbol{\delta(n-3)}$

We know that $\delta(n-3) = \begin{cases} 1 & for\ n = 3 \\ 0 & elsewhere \end{cases}$

$$\sum_{n=-\infty}^{\infty} e^{3n}\,\delta\,(n-3) = [e^{3n}]_{n=3} = e^{9}$$

(a) Given $\sum_{n=-\infty}^{\infty} \delta(n-2)\cos 3n$

We know that $\delta(n-2) = \begin{cases} 1 & for\ n = 2 \\ 0 & elsewhere \end{cases}$

$$\sum_{n=-\infty}^{\infty} \delta(n-2)cos3n = [\cos 3n]_{n=2} = \cos 6$$

(b) Given $\sum_{n=-\infty}^{\infty} n^2\ \delta(n=4)$

We know that $\delta(n=4) = \begin{cases} 1 & for\ n = -4 \\ 0 & elsewhere \end{cases}$

$$\sum_{n=n-\infty}^{\infty} n^2\delta(n=4) = [n^2]_{n=-4} = 16$$

(c) Given $\sum_{n=-\infty}^{\infty} \delta(n-2)e^{n^2}$

We know that $\delta(n-2) = \begin{cases} 1\ for\ n = 2 \\ 0\ elsewhere \end{cases}$

$$\sum_{n=-\infty}^{\infty} \delta(n-2)e^{n^2} = [e^{n^2}]_{n=2} = e^{2^2} = e^{4}$$

(d) Given $\sum_{n=0}^{\infty} \delta(n=1)4^n$

We know that $\delta(n=1) = \begin{cases} 1\ for\ n = -1 \\ 0\ for\ n \neq -1 \end{cases}$

$$\sum_{n=0}^{\infty} \delta(n+1)4^n = 0$$

**1.4 BASIC OPERATIONS ON SEQUENCES**

When we process a sequence, this sequence may undergo several manipulations involving the independent variable or the amplitude of the signal.

The basic operations on sequences are as follows:

1. Time shifting
2. Time reversal
3. Time scaling
4. Amplitude scaling
5. Signal addition
6. Signal multiplication

The first three operations correspond to transformation in independent variable *n* of a signal. The last three operations correspond to transformation on amplitude of a signal.

## 1.4.1 Time Shifting

The time shifting of a signal may result in time delay or time advance. The time shifting operation of a discrete-time signal $x(n)$ can be represented by

$$y(n) = x(n - k)$$

This shows that the signal $y(n)$ can be obtained by time shifting the signal $x(n)$ by $k$ units. If $k$ is positive, it is delay and the shift is to the right, and if $k$ is negative, it is advance and the shift is to the left.

An arbitrary signal $x(n)$ is shown in Figure 1.10(a). $x(n-3)$ which is obtained by shifting $x(n)$ to the right by 3 units (i.e. delay $x(n)$ by 3 units) is shown in Figure 1.10(b). $x(n+2)$ which is obtained by shifting $x(n)$ to the left by 2 units (i.e. advancing $x(n)$ by 2 units) is shown in Figure 1.10(c).

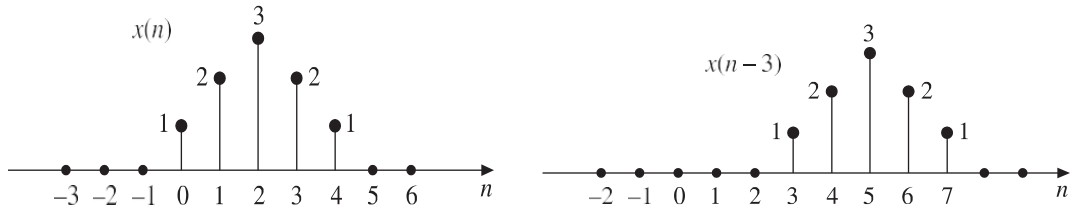



Figure 1.10 (a) Sequence x(n) (b) x(n – 3) (c) x(n + 2).

### *1.4.2 Time Reversal*

The time reversal also called time folding of a discrete-time signal $x(n)$ can be obtained by foldingthe sequence about $n = 0$. The time reversed signal is the reflection of the original signal. It is obtained by replacing the independent variable $n$ by $-n$. Figure 1.11(a) shows an arbitrary discrete-time signal $x(n)$, and its time reversed version $x(-n)$ is shown in Figure 1.11(b).

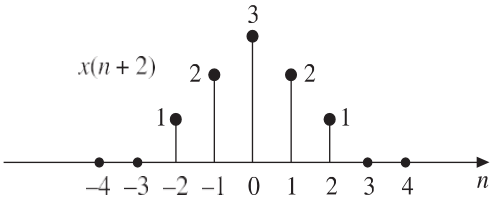



Figure 1.11[(c) and (d)] shows the delayed and advanced versions of reversed signal $x(-n)$.

The signal $x(-n+3)$ is obtained by delaying (shifting to the right) the time reversed signal $x(-n)$ by 3 units of time. The signal $x(-n-3)$ is obtained by advancing (shifting to the left) the time reversed signal $x(-n)$ by 3 units of time.

Figure 1.12 shows other examples for time reversal of signals

**EXAMPLE 1.2** Sketch the following signals:

(a) U(n+2) u(-n+3) (b) x(n) = u(n+4) – u(n-2)

**Solutions:**

**(a) Given x(n)=u(n+2) u(-n+3)**

The signal $u(n+2)\,u(-n+3)$ can be obtained by first drawing the signal $u(n+2)$ as shown in Figure 1.13(a), then drawing $u(-n+3)$ as shown in Figure 1.13(b),

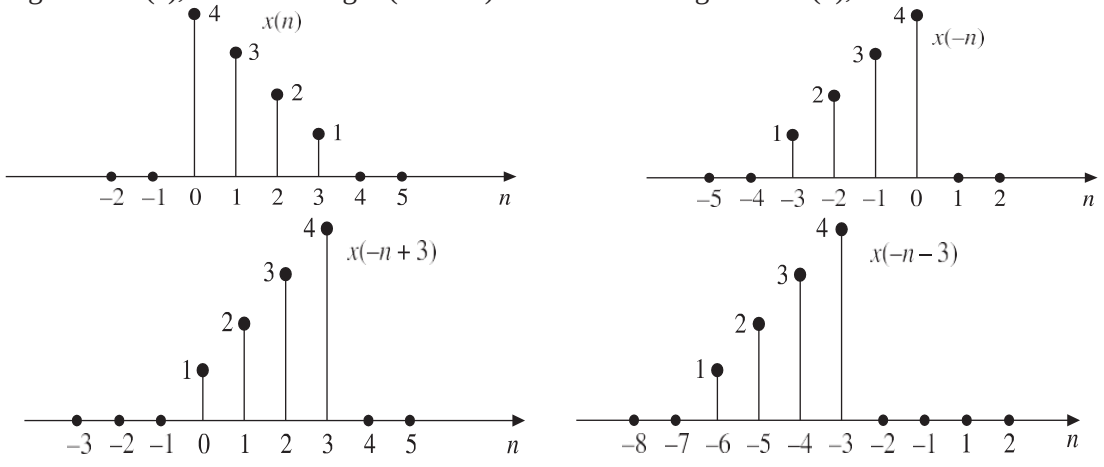



**Figure 1.11** (a) Original signal x(n) (b) Time reversed signal x(-n) (c) Time reversed and delayed

signal x(-n+3) (d) Time reversed and advanced signal x(-n-3).

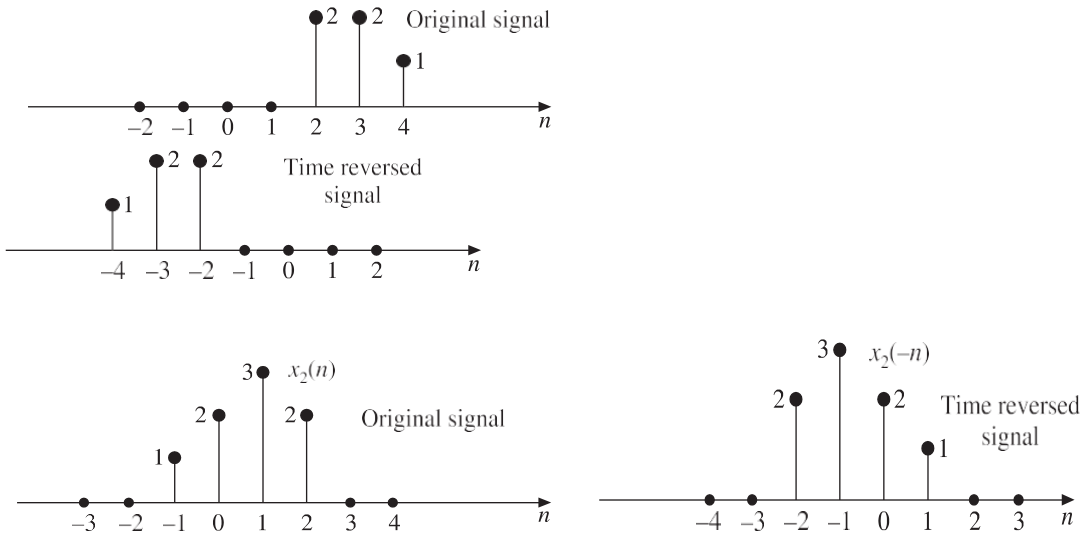



**Figure 1.12** Time reversal operations.

and then multiplying these sequences element by element to obtain $u(n + 2)\, u(-n + 3)$ as shown in Figure 1.13(c).

$x(n) = 0$ for $n < -2$ and $n > 3$; $x(n) = 1$ for $-2 < n < 3$

(a) Given $x(n) = u(n + 4) - u(n - 2)$

The signal $u(n + 4) - u(n - 2)$ can be obtained by first plotting $u(n + 4)$ as shown in Figure 1.14(a), then plotting $u(n - 2)$ as shown in Figure 1.14(b), and then subtracting each element of $u(n - 2)$ from the corresponding element of $u(n + 4)$ to obtain the result shown in Figure 1.14(c).

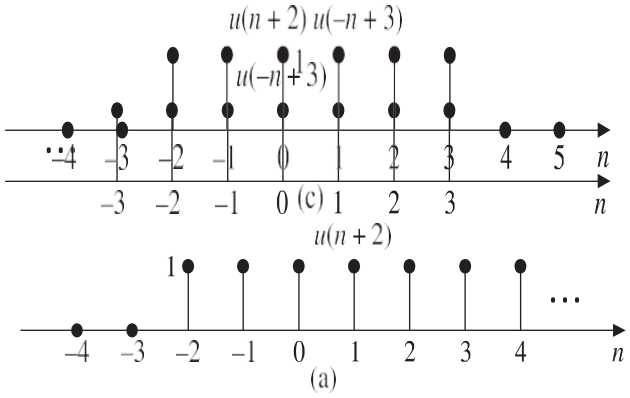
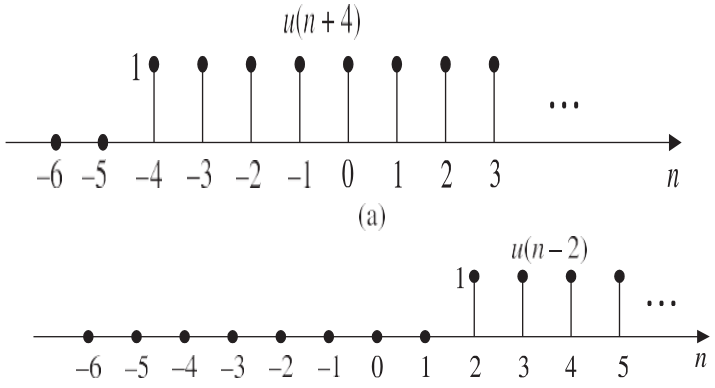



**Figure 1.13** Plots of (a) u(n + 2) (b) u(–n + 3) (c) u(n + 2) u(–n + 3).



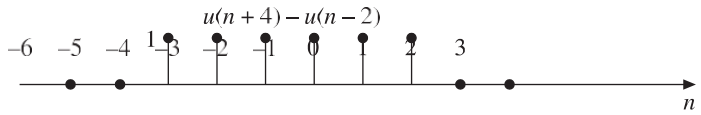



**Figure 1.14** Plots of (a) u(n + 4) (b) u(n – 2) (c) u(n + 4) – u(n – 2).

### *1.4.3 Amplitude Scaling*

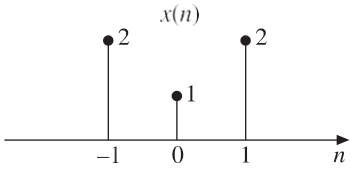
The amplitude scaling of a discrete-time signal can be represented by

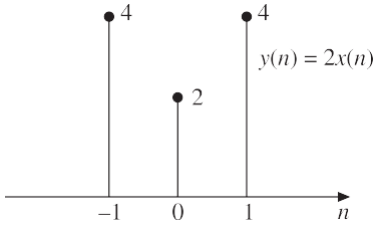
$$y(n) = ax(n)$$

where $a$ is a constant.

The amplitude of $y(n)$ at any instant is equal to $a$ times the amplitude of $x(n)$ at that instant. If $a > 1$, it is amplification and if $a < 1$, it is attenuation. Hence the amplitude is rescaled. Hence the name amplitude scaling.

Figure 1.15(a) shows a signal $x(n)$ and Figure 1.15(b) shows a scaled signal $y(n) = 2x(n)$.




### *1.4.1 Time Scaling*

Time scaling may be time expansion or time compression. The time scaling of a discrete- time signal $x(n)$ can be accomplished by replacing $n$ by $an$ in it. Mathematically, it can be expressed as:

$$y(n) = x(an)$$

When $a > 1$, it is time compression and when $a < 1$, it is time expansion.

Let $x(n)$ be a sequence as shown in Figure 1.16(a). If $a = 2$, $y(n) = x(2n)$. Then

$$y(0) = x(0) = 1$$
$$y(–1) = x(–2) = 3$$
$$y(–2) = x(–4) = 0$$
$$y(1) = x(2) = 3$$
$$y(2) = x(4) = 0$$

and so on.

So to plot $x(2n)$ we have to skip odd numbered samples in $x(n)$.

We can plot the time scaled signal $y(n) = x(2n)$ as shown in Figure 1.16(b). Here the signal is

compressed by 2.

If $a = (1/2)$, $y(n) = x(n/2)$, then

$$y(0) = x(0) = 1$$
$$y(2) = x(1) = 2$$
$$y(4) = x(2) = 3$$
$$y(6) = x(3) = 4$$
$$y(8) = x(4) = 0$$
$$y(–2) = x(–1) = 2$$
$$y(–4) = x(–2) = 3$$
$$y(–6) = x(–3) = 4$$
$$y(–8) = x(– 4) = 0$$

We can plot $y(n) = x(n/2)$ as shown in Figure 1.16(c). Here the signal is expanded by 2. All odd components in $x(n/2)$ are zero because $x(n)$ does not have any value in between the sampling instants.

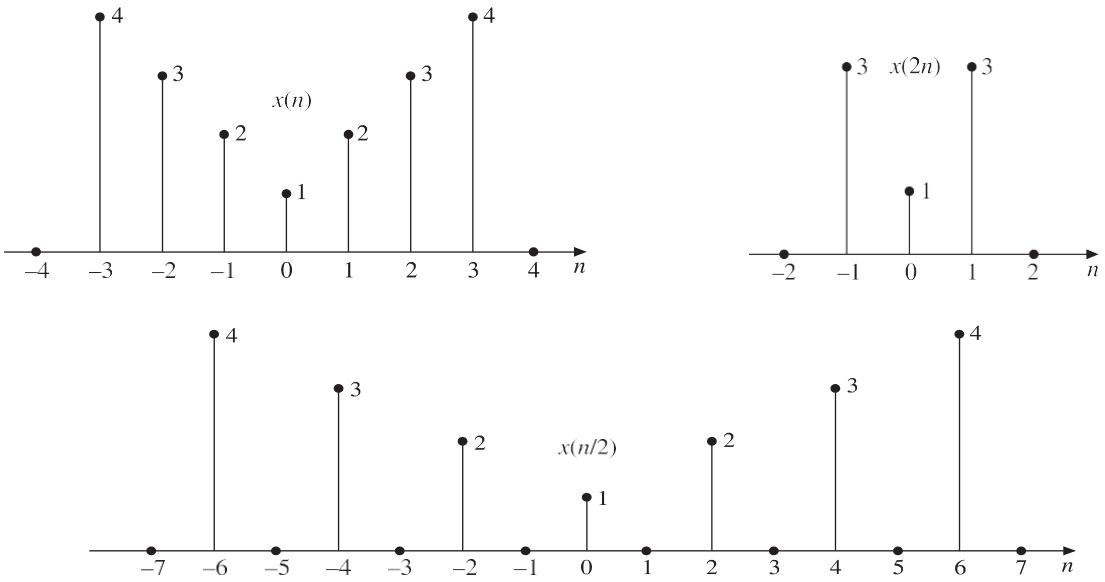



**Figure 1.16** Discrete-time scaling (a) Plot of x(n) (b) Plot of x(2n) (c) Plot of x(n/2)

Time scaling is very useful when data is to be fed at some rate and is to be taken out at a different rate.

## *1.45 Signal Addition*

In discrete-time domain, the sum of two signals $x_1(n)$ and $x_2(n)$ can be obtained by adding the corresponding sample values and the subtraction of $x_2(n)$ from $x_1(n)$ can be obtained by subtracting each sample of $x_2(n)$ from the corresponding sample of $x_1(n)$ as illustrated below.

If $x_1(n) = \{1, 2, 3, 1, 5\}$ and $x_2(n) = \{2, 3, 4, 1, -2\}$

Then $x_1(n) + x_2(n) = \{1 + 2, 2 + 3, 3 + 4, 1 + 1, 5 - 2\} = \{3, 5, 7, 2, 3\}$

and $x_1(n) - x_2(n) = \{1 - 2, 2 - 3, 3 - 4, 1 - 1, 5 + 2\} = \{-1, -1, -1, 0, 7\}$

### *1.4.6 Signal multiplication*

The multiplication of two discrete-time sequences can be performed by multiplying their values at the sampling instants as shown below.

If $x_1(n) = \{1, -3, 2, 4, 1.5\}$ and $x_2(n) = \{2, -1, 3, 1.5, 2\}$

Then $x_1(n)\, x_2(n) = \{1 \times 2, -3 \times -1, 2 \times 3, 4 \times 1.5, 1.5 \times 2\}$

$= \{2, 3, 6, 6, 3\}$

**EXAMPLE 1.3** Express the signals shown in Figure 1.17 as the sum of singular functions.

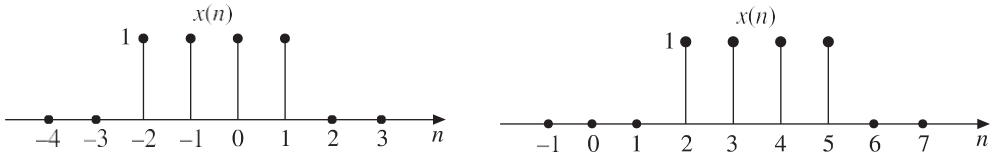



**Figure 1.17** Waveforms for Example 1.3

Solution:

(a) The given signal shown in Figure 1.17(a) is:

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

$$x(n) = \begin{cases} 0 & for\ n \le -3 \\ 1 & for -2 \le n \le 1 \\ 0 & for\ n \ge 2 \end{cases}$$

$\therefore$ $x(n) = u(n+2) - u(n-2)$

(b) The signal shown in Figure 1.17(b) is:

$$x(n) = \delta(n-2) + \delta(n-3) + \delta(n-4) + \delta(n-5)$$

$$x(n) = \begin{cases} 0 & for\ n \le 1 \\ 1 & for\ 2 \le n \le 5 \\ 0 & for\ n \ge 6 \end{cases}$$

$$\therefore \quad x(n) = u(n-2) - u(n-6)$$

## 1.4 CLASSIFICATION OF DISCRETE-TIME SIGNALS

The signals can be classified based on their nature and characteristics in the time domain. They are broadly classified as: (i) continuous-time signals and (ii) discrete-time signals.

The signals that are defined for every instant of time are known as continuous-time signals. The continuous-time signals are also called analog signals. They are denoted by $x(t)$. They are continuous in amplitude as well as in time. Most of the signals available are continuous-time signals.

The signals that are defined only at discrete instants of time are known as discrete-time signals. The discrete-time signals are continuous in amplitude, but discrete in time. For discrete- time signals, the amplitude between two time instants is just not defined. For discrete-time signals, the independent variable is time $n$. Since they are defined only at discrete instants of time, they are denoted by a sequence $x(nT)$ or simply by $x(n)$ where $n$ is an integer.

Figure 1.18 shows the graphical representation of discrete-time signals. The discrete- time signals may be inherently discrete or may be discrete versions of the continuous-time signals.

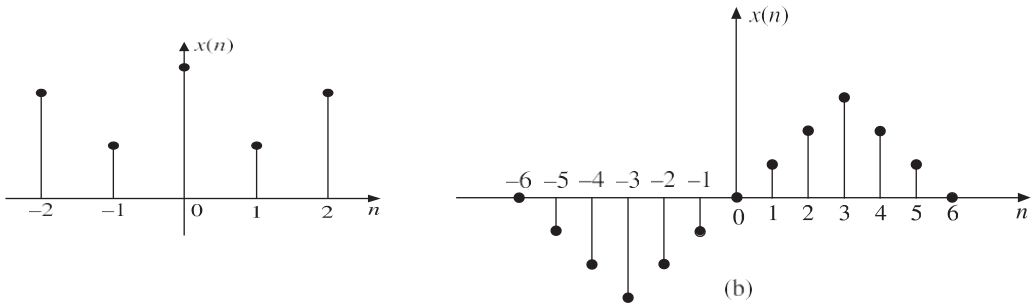



**Figure 1.18** Discrete-time signals

Both continuous-time and discrete-time signals are further classified as follows:

1. Deterministic and random signals
2. Periodic and non-periodic signals
3. Energy and power signals
4. Causal and non-causal signals
5. Even and odd signals

### 1.5.1 Deterministic and Random Signals

A signal exhibiting no uncertainty of its magnitude and phase at any given instant of time is called deterministic signal. A deterministic signal can be completely represented by mathematical equation at any time and its nature and amplitude at any time can be predicted.

*Examples:* Sinusoidal sequence $x(n) = \cos n$, Exponential sequence $x(n) = e^{j\ n}$, ramp sequence $x(n) = n$.

A signal characterized by uncertainty about its occurrence is called a non-deterministic or random signal. A random signal cannot be represented by any mathematical equation. The behavior of such a signal is probabilistic in nature and can be analyzed only stochastically. The pattern of such a signal is quite irregular. Its amplitude and phase at any time instant cannot be predicted in advance. A typical example of a non-deterministic signal is thermal noise.

### 1.5.2 Periodic and Non-periodic Sequences

A signal which has a definite pattern and repeats itself at regular intervals of time is called a periodic signal, and a signal which does not repeat at regular intervals of time is called a non-periodic or aperiodic signal.

A discrete-time signal $x(n)$ is said to be periodic if it satisfies the condition $x(n) = x(n + N)$ for all integers $n$.

The smallest value of $N$ which satisfies the above condition is known as fundamental period.

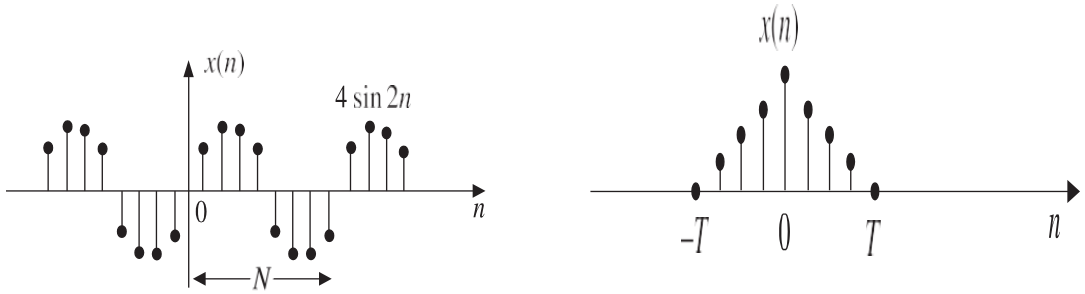
If the above condition is not satisfied even for one value of $n$, then the discrete-time signal is aperiodic. Sometimes aperiodic signals are said to have a period equal to infinity.

The angular frequency is given by

$$\omega = \frac{2\pi}{N}$$

Fundamental period $N = \frac{2\pi}{\omega}$

The sum of two discrete-time periodic sequence is always periodic.




some examples of discrete-time periodic/non-periodic signals are shown in Figure 1.19.

**Figure 1.19** Example of discrete-time: (a) Periodic and (b) Non-periodic signals

**EXAMPLE 1.4** Show that the complex exponential sequence $x(n) = e^{j\ 0n}$ is periodic only if $_0/2$ is a rational number.

***Solution:*** Given $x(n) = e^{j\omega_0 n}$

X (n) will be periodic if $x(n + N) = x(n)$

i.e. $e^{j[\omega_0(n=N0]} = e^{j\omega_0 n}$

i.e. $e^{j\omega_0 N}\ e^{j\omega_0 n} = e^{j\omega_0 n}$

This is possible only if $e^{j\ 0N} = 1$

This is true only if $\omega_0 N = 2\pi k$

Where k is an integer $\frac{\omega_0}{2\pi} = \frac{k}{N}$

### *1.5.3 Energy Signals And Power Signals*

Signals may also be classified as energy signals and power signals. However there are some signals which can neither be classified as energy signals nor power signals.

The total energy $E$ of a discrete-time signal $x(n)$ is defined as:

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

and the average power $P$ of a discrete-time signal $x(n)$ is defined as:

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

or $P = \frac{1}{N}\sum_{n=0}^{n-1} |x(n)|^2$ for a digital signal with $x(n) = 0$ for $n < 0$.

A signal is said to be an energy signal if and only if its total energy $E$ over the interval $(-\infty, \infty)$ is finite (i.e., $0 < E < \infty$). For an energy signal, average power $P = 0$. Non-periodic signals which are defined over a finite time (also called time limited signals) are the examples of energy signals. Since the energy of a periodic signal is always either zero or infinite, any periodic signal cannot be an energy signal.

A signal is said to be a power signal, if its average power $P$ is finite (i.e., $0 < P < \infty$). For a power signal, total energy $E = \infty$. Periodic signals are the examples of power signals. Every bounded and periodic signal is a power signal. But it is true that a power signal is not necessarily a bounded and periodic signal.

Both energy and power signals are mutually exclusive, i.e. no signal can be both energy signal and power signal.

The signals that do not satisfy the above properties are neither energy signals nor power signals. For example, $x(n) = u(n)$, $x(n) = nu(n)$, $x(n) = n^2u(n)$.

These are signals for which neither $P$ nor $E$ are finite. If the signals contain infinite energy and zero power or infinite energy and infinite power, they are neither energy nor power signals.

If the signal amplitude becomes zero as $|n| \rightarrow \infty$, it is an energy signal, and if the signal amplitude does not become zero as $|n| \rightarrow \infty$, it is a power signal.

### *Causal and Non-causal Signals*

A discrete-time signal $x(n)$ is said to be causal if $x(n) = 0$ for $n < 0$, otherwise the signal is non-causal. A discrete-time signal $x(n)$ is said to be anti-causal if $x(n) = 0$ for $n > 0$.

A causal signal does not exist for negative time and an anti-causal signal does not exist for positive time. A signal which exists in positive as well as negative time is called a non-casual signal.

$u(n)$ is a causal signal and $u(-n)$ an anti-causal signal, whereas $x(n) = 1$ for $-2 \leq n \leq 3$ is a non-causal signal.

### *Even and Odd Signals*

Any signal $x(n)$ can be expressed as sum of even and odd components. That is

$$x(n) = x_e(n) + x_o(n)$$

where $x_e(n)$ is even components and $x_o(n)$ is odd components of the signal.

## *Even (syMMetric) signal*

A discrete-time signal $x(n)$ is said to be an even (symmetric) signal if it satisfies the condition:

$$x(n) = x(-n) \quad \text{for all } n$$

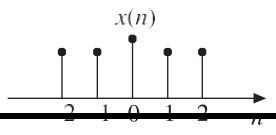
Even signals are symmetrical about the vertical axis or time origin. Hence they are also called symmetric signals: cosine sequence is an example of an even signal. Some even signals are shown in Figure 1.20(a). An even signal is identical to its reflection about the origin. For an even signal $x_0(n) = 0$.

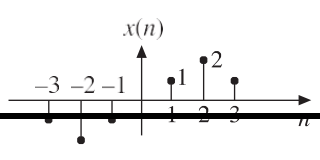
## *Odd (anti-syMMetric) signal*

A discrete-time signal $x(n)$ is said to be an odd (anti-symmetric) signal if it satisfies the condition:

$$x(-n) = -x(n) \quad \text{for all } n$$

Odd signals are anti-symmetrical about the vertical axis. Hence they are called anti- symmetric signals. Sinusoidal sequence is an example of an odd signal. For an odd signal $x_e(n) = 0$. Some odd signals are shown in Figure 1.20(b).



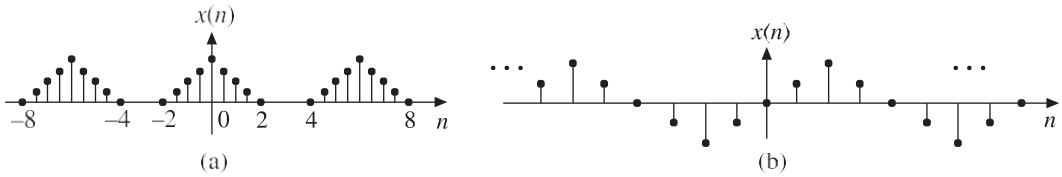



Figure 1.20 (a) Even sequences (b) Odd sequences.

Thus, the product of two even signals or of two odd signals is an even signal, and the product of even and odd signals is an odd signal.

*Every signal need not be either purely even signal or purely odd signal, but every signal can be decomposed into sum of even and odd parts.*

### *CLASSIFICATIOK OF DISCRETE-TImE SYSTEmS*

A system is defined as an entity that acts on an input signal and transforms it into an output signal. A system may also be defined as a set of elements or functional blocks which are connected together and produces an output in response to an input signal. The response or output of the system depends on the transfer function of the system. It is a cause and effect relation between two or more signals.

As signals, systems are also broadly classified into continuous-time and discrete-time systems. A continuous-time system is one which transforms continuous-time input signals into continuous-time output signals, whereas a discrete-time system is one which transforms discrete-time input signals into discrete-time output signals.

For example microprocessors, semiconductor memories, shift registers, etc. are discrete-time systems.

A discrete-time system is represented by a block diagram as shown in Figure 1.22. An arrow entering the box is the input signal (also called excitation, source or driving function) and an arrow leaving the box is an output signal (also called response). Generally, the input is denoted by $x(n)$ and the output is denoted by $y(n)$.

The relation between the input $x(n)$ and the output $y(n)$ of a system has the form:

$$y(n) = \text{Operation on } x(n)$$

Mathematically,

$$y(n) = T[x(n)]$$

which represents that $x(n)$ is transformed to $y(n)$. In other words, $y(n)$ is the transformed version of $x(n)$.

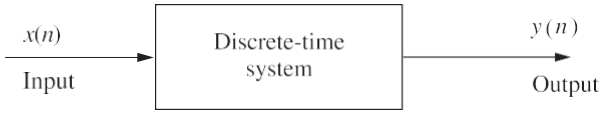



Figure 1.22 Block diagram of discrete-time system.

Both continuous-time and discrete-time systems are further classified as follows:

1. Static (memoryless) and dynamic (memory) systems
2. Causal and non-causal systems
3. Linear and non-linear systems
4. Time-invariant and time varying systems
5. Stable and unstable systems.
6. Invertible and non-invertible systems
7. FIR and IIR systems

### *Static and Dynamic Systems*

A system is said to be static or memoryless if the response is due to present input alone, i.e., for a static or memoryless system, the output at any instant $n$ depends only on the input applied at that instant $n$ but not on the past or future values of input or past values of output.

For example, the systems defined below are static or memoryless systems.

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

$$y(n) = 2x^2(n)$$

In contrast, a system is said to be dynamic or memory system if the response depends upon past or future inputs or past outputs. A summer or accumulator, a delay element is a discrete-time system with memory.

For example, the systems defined below are dynamic or memory systems.

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

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

$$y(n) + 4y(n-1) + 4y(n-2) = x(n)$$

Any discrete-time system described by a difference equation is a dynamic system.

A purely resistive electrical circuit is a static system, whereas an electric circuit having inductors and/or capacitors is a dynamic system.

A discrete-time LTI system is memoryless (static) if its impulse response $h(n)$ is zero for $n \neq 0$. If the impulse response is not identically zero for $n \neq 0$, then the system is called dynamic system or system with memory.

**EXAMPLE 1.12** Find whether the following systems are dynamic or not:

(a) $y(n) = x(n + 2)$ (b) $y(n) = x^2(n)$

(c) $y(n) = x(n - 2) + x(n)$

*Solution:*

(a) Given $y(n) = x(n + 2)$

The output depends on the future value of input. Therefore, the system is dynamic.

(b) Given $y(n) = x^2(n)$

The output depends on the present value of input alone. Therefore, the system is static.

(c) Given $y(n) = x(n - 2) + x(n)$

The system is described by a difference equation. Therefore, the system is dynamic.

### *Causal and Non-causal Systems*

A system is said to be causal (or non-anticipative) if the output of the system at any instant $n$ depends only on the present and past values of the input but not on future inputs, i.e., for a causal system, the impulse response or output does not begin before the input function is applied, i.e., a causal system is non anticipatory.

Causal systems are real time systems. They are physically realizable.

The impulse response of a causal system is zero for $n < 0$, since $(n)$ exists only at $n = 0$,

i.e. $h(n) = 0 \quad \text{for} \quad n < 0$

The examples for causal systems are:

$$y(n) = nx(n)$$

$$y(n) = x(n - 2) + x(n - 1) + x(n)$$

A system is said to be non-causal (anticipative) if the output of the system at any instant $n$ depends on future inputs. They are anticipatory systems. They produce an output even before the input is given. They do not exist in real time. They are not physically realizable.

A delay element is a causal system, whereas an image processing system is a non-causal system.

The examples for non-causal systems are:

$$y(n) = x(n) + x(2n)$$

$$y(n) = x^2(n) + 2x(n + 2)$$

**EXAMPLE 1.13** Check whether the following systems are causal or not:

(a) $y(n)\ x(n)\ x(n\ 2)$ (b) $y(n) = x(2n)$
(c) $y(n) = \sin[x(n)]$ (d) $y(n) = x(-n)$

*Solution:*

(a) Given $y(n)\ x(n)\ x(n\ 2)$

For $n = -2$ $y(\ 2)\ x(\ 2)\ x(\ 4)$

For $n = 0$ $y(0)\ x(0)\ x(\ 2)$

For $n = 2$ $y(2)\ x(2)\ x(0)$

For all values of $n$, the output depends only on the present and past inputs. Therefore, the system is causal.

(a) Given $y(n)\quad x(2n)$

For $n = -2$ $y(\ 2)\quad x(\ 4)$

For $n = 0$ $y(0)\quad x(0)$

For $n = 2$ $y(2)\quad x(4)$

For positive values of $n$, the output depends on the future values of input. Therefore, the system is non-causal.

(a) Given $y(n)\ \sin\ [x(n)]$

For $n = -2$ $y(\ 2)\ \sin\ [x(\ 2)]$

For $n = 0$ $y(0)\quad \sin\ [x(0)]$

For $n = 2$ $y(2)\quad \sin\ [x(2)]$

For all values of $n$, the output depends only on the present value of input. Therefore, the system is causal.

(d) Given $y(n) = x(-n)$

For $n = -2$ $y(\ 2)\ x(2)$

For $n = 0$ $y(0)\ x(0)$

For $n = 2$ $y(2)\ x(\ 2)$

For negative values of $n$, the output depends on the future values of input. Therefore, the system is non-causal.

### *Linear and Non-linear Systems*

A system which obeys the principle of superposition and principle of homogeneity is called a linear system and a system which does not obey the principle of superposition and homogeneity is called a non-linear system.

Homogeneity property means a system which produces an output $y(n)$ for an input $x(n)$ must produce an output $ay(n)$ for an input $ax(n)$.

Superposition property means a system which produces an output $y_1(n)$ for an input $x_1(n)$ and an output $y_2(n)$ for an input $x_2(n)$ must produce an output $y_1(n) + y_2(n)$ for an input $x_1(n) + x_2(n)$.

Combining them we can say that a system is linear if an arbitrary input $x_1(n)$ produces an output $y_1(n)$ and an arbitrary input $x_2(n)$ produces an output $y_2(n)$, then the weighted sum of inputs $ax_1(n) + bx_2(n)$ where $a$ and $b$ are constants produces an output $ay_1(n) + by_2(n)$ which is the sum of weighted outputs.

$$T(ax_1(n) + bx_2(n)] = aT[x_1(n)] + bT[x_2(n)]$$

Simply we can say that a system is linear if the output due to weighted sum of inputs is equal to the weighted sum of outputs.

In general, if the describing equation contains square or higher order terms of input and/or output and/or product of input/output and its difference or a constant, the system will definitely be non-linear.

### *Shift-invariant and Shift-varying Systems*

Time-invariance is the property of a system which makes the behaviour of the system independent of time. This means that the behaviour of the system does not depend on the time at which the input is applied. For discrete-time systems, the time invariance property is called shift invariance.

A system is said to be shift-invariant if its input/output characteristics do not change with time, i.e., if a time shift in the input results in a corresponding time shift in the output as shown in Figure 1.23, i.e.

If $T[x(n)] = y(n)$

Then $T[x(n-k)] = y(n-k)$

A system not satisfying the above requirements is called a time-varying system (or shift-varying system). A time-invariant system is also called a fixed system.

The time-invariance property of the given discrete-time system can be tested as follows:

Let $x(n)$ be the input and let $x(n-k)$ be the input delayed by $k$ units.

$y(n) = T[x(n)]$ be the output for the input $x(n)$.

### *Stable and Unstable Systems*

A bounded signal is a signal whose magnitude is always a finite value, i.e. $|x(n)| \leq M$, where $M$ is a positive real finite number. For example a sinewave is a bounded signal. A system is said to be bounded-input, bounded-output (BIBO) stable, if and only if every bounded input produces a bounded output. The output of such a system does not diverge or does not grow unreasonably large.

Let the input signal $x(n)$ be bounded (finite), i.e.,

$$|x(n)| \leq M_x \quad \text{for all } n$$

where $M_x$ is a positive real number. If

$$|y(n)| \leq M_y \leq \infty$$

i.e. if the output $y(n)$ is also bounded, then the system is BIBO stable. Otherwise, the system is unstable. That is, we say that a system is unstable even if one bounded input produces an unbounded output.

It is very important to know about the stability of the system. Stability indicates the usefulness of the system. The stability can be found from the impulse response of the system which is nothing but the output of the system for a unit impulse input. If the impulse response is absolutely summable for a discrete-time system, then the system is stable.

## *BIBO stability criterion*

The necessary and sufficient condition for a discrete-time system to be BIBO stable is given by the expression:

$$\sum_{n=-\infty}^{\infty} |h(n)| < \infty$$

where $h(n)$ is the impulse response of the system. This is called BIBO stability criterion.

*Proof:* Consider a linear time-invariant system with $x(n)$ as input and $y(n)$ as output. The input and output of the system are related by the convolution integral.

**SOLUTION OF DIFFERENCE EQUATIONS USING Z-TRANSFORMS.**

To solve the difference equation, first it is converted into algebraic equation by taking its Z-transform. The solution is obtained in *z*-domain and the time domain solution is obtained by taking its inverse Z-transform. The system response has two components. The source free response and the forced response. The response of the system due to input alone when the initial conditions are neglected is called the forced response of the system. It is also called the steady state response of the system. It represents the component of the response due to the driving force. The response of the system due to initial conditions alone when the input is neglected is called the free or natural response of the system. It is also called the transient response of the system. It represents the component of the response when the driving function is made zero. The response due to input and initial conditions considered simultaneously is called the total response of the system. For a stable system, the source free component always decays with time. In fact a stable system is one whose source free component decays with time. For this reason the source free component is also designated as the transient component and the component due to source is called the steady state component. When input is a unit impulse input, the response is called the impulse response of the system and when the input is a unit step input, the response is called the step response of the system.

**EXAMPLE 1** A linear shift invariant system is described by the difference equation

$$y(n) - \frac{3}{4}y(n-1) + \frac{1}{8}y(n-2) = x(n) + x(n-1)$$

with $y(-1) = 0$ and $y(-2) = -1$.
Find (a) the natural response of the system (b) the forced response of the system for a step input and (c) the frequency response of the system.

***Solution:***

(a) The natural response is the response due to initial conditions only. So make $x(n) = 0$. Then the difference equation becomes

$$y(n) - \frac{3}{4}y(n-1) + \frac{1}{8}y(n-2) = 0$$

Taking Z-transform on both sides, we have

$$Y(z) - \frac{3}{4}\left[z^{-1}Y(z) + y(-1)\right] + \frac{1}{8}\left[z^{-2}Y(z) + z^{-1}y(-1) + y(-2)\right] = 0$$

i.e.
$$Y(z)\left(1 - \frac{3}{4}z^{-1} + \frac{1}{8}z^{-2}\right) - \frac{1}{8} = 0$$

$$\therefore\ Y(z) = \frac{1/8}{1 - (3/4)z^{-1} + (1/8)z^{-2}} = \frac{1/8z^2}{z^2 - (3/4)z + (1/8)} = \frac{1/8z^2}{[z - (1/2)][z - (1/4)]}$$

The partial fraction expansion of $Y(z)/z$ gives

$$\frac{Y(z)}{z} = \frac{(1/8)z}{[z - (1/2)][z - (1/4)]} = \frac{A}{z - (1/2)} + \frac{B}{z - (1/4)} = \frac{1/4}{z - (1/2)} - \frac{1/8}{z - (1/4)}$$

$$Y(z) = \frac{1}{4}\frac{z}{z - (1/2)} - \frac{1}{8}\frac{z}{z - (1/4)}$$

Taking inverse Z-transform on both sides, we get the natural response as:

$$y(n) = \frac{1}{4}\left(\frac{1}{2}\right)^n u(n) - \frac{1}{8}\left(\frac{1}{4}\right)^n u(n)$$

(a) To find the forced response due to a step input, put $x(n) = u(n)$. So we have

$$y(n) - \frac{3}{4}y(n-1) + \frac{1}{8}y(n-2) = u(n) + u(n-1)$$

We know that the forced response is due to input alone. So for forced response, the initial conditions are neglected. Taking Z-transform on both sides of the above equation and neglecting the initial conditions, we have

$$y(n) - \frac{3}{4}y(n-1) + \frac{1}{8}y(n-2) = u(n) + u(n-1)$$

We know that the forced response is due to input alone. So for forced response, the initial conditions are neglected. Taking Z-transform on both sides of the above equation and neglecting the initial conditions, we have

$$Y(z) - \frac{3}{}z^{-1}Y(z) + \frac{1}{}z^{-2}Y(z) = U(z) + z^{-1}U(z) = 4\ 8$$

i.e. $$Y(z)\left(1-\frac{3}{4}z^{-1}+\frac{1}{8}z^{-2}\right)=\frac{z+1}{z-1}$$

$$\therefore\quad Y(z)=\frac{z+1}{(z-1)[1-(3/4)z^{-1}+(1/8)z^{-2})}=\frac{z^2(z+1)}{(z-1)[z^2-(3/4)z+(1/8)]}$$

$$=\frac{z^2(z+1)}{(z-1)[z-(1/2)][z-(1/4)]}$$

Taking partial fractions of $Y(z)/z$, we have

$$\therefore\quad \frac{Y(z)}{z}=\frac{z(z+1)}{(z-1)[z-(1/2)][z-(1/4)]}=\frac{A}{z-1}+\frac{B}{z-(1/2)}+\frac{C}{z-(1/4)}$$

$$=\frac{16/3}{z-1}-\frac{6}{z-(1/2)}+\frac{5/3}{z-(1/4)}$$

or $$Y(z)=\frac{16}{3}\left(\frac{z}{z-1}\right)-6\left[\frac{z}{z-(1/2)}\right]+\frac{5}{3}\left[\frac{z}{z-(1/4)}\right]$$

Taking the inverse Z-transform on both sides, we have the forced response for a step input.

$$y(n)=\frac{16}{3}u(n)-6\left(\frac{1}{2}\right)^n u(n)+\frac{5}{3}\left(\frac{1}{4}\right)^n u(n)$$

© The frequency response of the system $H(\ )$ is obtained by putting $z = e^{jw}$ in $H(z)$.

$$H(z)=\frac{Y(z)}{X(z)}=\frac{z(z+1)}{z^2-(3/4)z+(1/8)}$$

$$H(\omega)=\frac{e^{j\omega}(e^{j\omega}+1)}{(e^{j\omega})^2-(3/4)e^{j\omega}+(1/8)}$$

**EXAMPLE 2** (a) Determine the free response of the system described by the difference equation $y(n)-\frac{5}{6}y(n-1)+\frac{1}{6}y(n-2)=x(n)$ with $y(-1) = 1$ and $y(-2) = 0$

(b) Determine the forced response for an input

**Solution:**

(a) The free response, also called the natural response or transient response is the response due to initial conditions only [i.e. make $x(n) = 0$]. So, the difference equation is:

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

Taking Z-transform on both sides, we get

$$Y(z)-\frac{5}{6}[z^{-1}Y(z)+y(-1)]+\frac{1}{6}[z^{-2}Y(z)+z^{-1}y(-1)+y(-2)]=0$$

$$Y(z)\left(1-\frac{5}{6}z^{-1}+\frac{1}{6}z^{-2}\right)-\frac{5}{6}+\frac{1}{6}z^{-1}=0$$

$$\therefore \quad Y(z)=\frac{(5/6)-(1/6)z^{-1}}{1-(5/6)z^{-1}+(1/6)z^{-2}}=\frac{5/6[z-(1/5)]z}{z^2-(5/6)z+(1/6)}=\frac{(5/6)z\,[z-(1/5)]}{[z-(1/2)][z-(1/3)]}$$

Taking partial fractions of $Y(z)/z$, we have

$$\frac{Y(z)}{z}=\frac{5/6\,[z-(1/5)]}{[z-(1/2)][z-(1/3)]}=\frac{A}{z-(1/2)}+\frac{B}{z-(1/3)}=\frac{3/2}{z-(1/2)}-\frac{2/3}{z-(1/3)}$$

$$\therefore \quad Y(z)=\frac{3}{2}\frac{z}{z-(1/2)}-\frac{2}{3}\frac{z}{z-(1/3)}$$

Taking inverse Z-transform on both sides, we get the free response of the system as:

(a) To determine the forced response, i.e. the steady state response, the initial conditions are to be neglected.

The given difference equation is:

$$y(n)-\frac{5}{6}y(n-1)+\frac{1}{6}y(n-2)=x(n)=\left(\frac{1}{4}\right)^n u(n)$$

Taking Z-transform on both sides and neglecting the initial conditions, we have

$$Y(z)-\frac{5}{6}z^{-1}Y(z)+\frac{1}{6}z^{-2}Y(z)=\frac{z}{z-(1/4)}$$

i.e., $$Y(z)\left(1-\frac{5}{6}z^{-1}+\frac{1}{6}z^{-2}\right)=\frac{z}{z-(1/4)}$$

$$\therefore \quad Y(z)=\frac{z}{z-(1/4)}\frac{1}{1-(5/6)z^{-1}+(1/6)z^{-2}}=\frac{z^3}{[z-(1/4)][z-(1/2)][z-(1/3)]}$$

Partial fraction expansion of $Y(z)/z$ gives

$$\frac{Y(z)}{z}=\frac{z^2}{[z-(1/4)][z-(1/3)][z-(1/2)]}=\frac{A}{z-(1/4)}+\frac{B}{z-(1/3)}+\frac{C}{z-(1/2)}$$

$$=\frac{3}{z-(1/4)}-\frac{8}{z-(1/3)}+\frac{6}{z-(1/2)}$$

Multiplying both sides by $z$, we get

$$Y(z)=3\frac{z}{z-(1/4)}-8\frac{z}{z-(1/3)}+6\frac{z}{z-(1/2)}$$

Taking inverse Z-transform on both sides, the forced response of the system is:

$$y(n) = 3\left(\frac{1}{4}\right)^n u(n) - 8\left(\frac{1}{3}\right)^n u(n) + 6\left(\frac{1}{2}\right)^n u(n)$$

**EXAMPLE 3** Find the impulse and step response of the system

$$y(n) = 2x(n) - 3x(n-1) + x(n-2) - 4x(n-3)$$

***Solution:*** For impulse response, $x(n) = \delta(n)$

The impulse response of the system is:

$$y(n) = 2\delta(n) - 3\delta(n-1) + \delta(n-2) - 4\delta(n-3)$$

For step response, $x(n) = u(n)$

The step response of the system is:

$$y(n) = 2u(n) - 3u(n\ 1) + u(n\ 2) - 4u(n\ 3)$$

**EXAMPLE 4** Solve the following difference equation

$$y(n) + 2y(n-1) = x(n)$$

with $x(n) = (1/3)^n u(n)$ and the initial condition $y(-1) = 1$.

***Solution:*** The solution of the difference equation considering the initial condition and input simultaneously gives the total response of the system.

The given difference equation is:

$$y(n) + 2y(n-1) = x(n) = \left(\frac{1}{3}\right)^n u(n) \text{ with } y(-1) = 1$$

Taking Z-transform on both sides, we get

$$Y(z) + 2[z^{-1}Y(z) + y(-1)] = X(z) = \frac{1}{1-(1/3)z^{-1}}$$

Substituting the initial conditions, we have

$$Y(z)\,(1+2z^{-1}) = -2(1) + \frac{1}{1-(1/3)\,z^{-1}}$$

$$\therefore \quad Y(z) = \frac{-2}{1+2z^{-1}} + \frac{1}{[1-(1/3)\,z^{-1}][1+2z^{-1}]}$$

$$= \frac{-2z}{z+2} + \frac{z^2}{[z-(1/3)]\,(z+2)}$$

$$\text{Let} \quad Y_1(z) = \frac{z^2}{[z-(1/3)]\,(z+2)}$$

Taking partial fractions of $Y_1(z)/z$, we have

$$\frac{Y_1(z)}{z} = \frac{z}{[z-(1/3)](z+2)} = \frac{A}{z-(1/3)} + \frac{B}{z+2} = \frac{1/7}{z-(1/3)} + \frac{6/7}{z+2}$$

Multiplying both sides by $z$, we have

$$Y_1(z) = \frac{1}{7}\frac{z}{z-(1/3)} + \frac{6}{7}\frac{z}{z+2}$$

$$\therefore \quad Y(z) = -\frac{2z}{z+2} + \frac{6}{7}\frac{z}{z+2} + \frac{1}{7}\frac{z}{z-(1/3)} = -\frac{8}{7}\frac{z}{z+2} + \frac{1}{7}\frac{z}{z-(1/3)}$$

Taking inverse Z-transform on both sides, the solution of the difference equation is:

$$y(n) = -\frac{8}{7}(-2)^n u(n) + \frac{1}{7}\left(\frac{1}{3}\right)^n u(n)$$

**EXAMPLE 5** Solve the following difference equation using unilateral Z-transform. with initial conditions

$$y(n) - \frac{7}{12}y(n-1) + \frac{1}{12}y(n-2) = x(n) \text{ for } n \geq 0$$

$$y(-1) = 2,\ y(-2) = 4 \text{ and } x(n) = \left(\frac{1}{5}\right)^n u(n)$$

***Solution:*** The solution of the difference equation gives the total response of the system (i.e., the sum of the natural (free) response and the forced response)

The given difference equation is:

$$y(n) - \frac{7}{12}y(n-1) + \frac{1}{12}y(n-2) = x(n) = \left(\frac{1}{5}\right)^n u(n)$$

with initial conditions $y(-1) = 2$ and $y(-2) = 4$. Taking Z-transform on both sides, we have

$$Y(z) - \frac{7}{12}[z^{-1}Y(z) + y(-1)] + \frac{1}{12}[z^{-2}Y(z) + z^{-1}y(-1) + y(-2)] = \frac{1}{1-(1/5)z^{-1}}$$

i.e. $$Y(z)\left(1 - \frac{7}{12}z^{-1} + \frac{1}{12}z^{-2}\right) = \frac{7}{12}(2) - \frac{1}{12}(2z^{-1}) - \frac{1}{12}(4) + \frac{1}{1-(1/5)z^{-1}}$$

i.e. $$Y(z)\left(1 - \frac{7}{12}z^{-1} + \frac{1}{12}z^{-2}\right) = \frac{5}{6}\left(1 - \frac{1}{5}z^{-1}\right) + \frac{1}{1-(1/5)z^{-1}}$$

$$\therefore \quad Y(z) = \frac{(5/6)[1-(1/5)z^{-1}]}{[1-(7/12)z^{-1}+(1/12)z^{-2}]} + \frac{1}{[1-(1/5)z^{-1}][1-(7/12)z^{-1}+(1/12)z^{-2}]}$$

$$= \frac{(5/6)[z-(1/5)]z}{[z-(1/4)][z-(1/3)]} + \frac{z^3}{[z-(1/5)][z-(1/4)][z-(1/3)]}$$

$$= \frac{z[(11/6)z^2-(1/3)z+(1/30)]}{[z-(1/5)][z-(1/4)][z-(1/3)]}$$

Taking partial fractions of $Y(z)/z$, we have

$$\frac{Y(z)}{z} = \frac{A}{z-(1/5)} + \frac{B}{z-(1/4)} + \frac{C}{z-(1/3)} = \frac{6}{5}\frac{1}{z-(1/5)} + \frac{1}{8}\frac{1}{z-(1/4)} + \frac{100}{27}\frac{1}{z-(1/3)}$$

Multiplying both sides by $z$, we have

$$Y(z) = \frac{6}{5}\frac{z}{z-(1/5)} + \frac{1}{8}\frac{z}{z-(1/4)} + \frac{102}{27}\frac{z}{z-(1/3)}$$

Taking inverse Z-transform on both sides, the solution of the difference equation is:

$$y(n) = \frac{6}{5}\left(\frac{1}{5}\right)^n u(n) + \frac{1}{8}\left(\frac{1}{4}\right)^n u(n) + \frac{102}{27}\left(\frac{1}{3}\right)^n u(n)$$

**EXAMPLE 6** Using Z-transform determine the response of the LTI system described by $y(n) - 2r\cos y(n-1) + r^2 y(n-2) = x(n)$ to an excitation $x(n) = a^n u(n)$.

***Solution:*** Taking Z-transform on both sides of the difference equation, we have

$$Y(z) - 2r\cos\theta\,[z^{-1}Y(z) + y(-1)] + r^2[z^{-2}Y(z) + z^{-1}y(-1) + y(-2)] = X(z)$$

$$\text{i.e.} \qquad Y(z)\,[1 - 2r\cos\theta\, z^{-1} + r^2 z^{-2}] = \frac{z}{z-a}$$

$$\therefore \qquad Y(z) = \frac{z^3}{(z-a)\,(z-re^{j\theta})\,(z-re^{-j\theta})}$$

$$= \frac{a^2}{a^2 - 2ar\cos\theta + r^2}\frac{z}{z-a} + \frac{r^2 e^{j2\theta}}{(re^{j\theta}-a)(re^{j\theta}-re^{-j\theta})}\frac{z}{z-re^{j\theta}}$$

$$+ \frac{r^2 e^{-j2\theta}}{(re^{-j\theta}-a)(re^{-j\theta}-re^{j\theta})}\frac{z}{z-re^{-j\theta}}$$

$$\therefore \qquad y(n) = \frac{a^2}{a^2 - 2ar\cos\theta + r^2}a^n u(n) + \frac{r^{n+1}}{\sin\theta}\left[\frac{r\sin(n+1)\theta - a\sin(n+2)\theta}{a^2 - 2ar\cos\theta + r^2}\right]u(n)$$

**EXAMPLE 7** Determine the step response of an LTI system whose impulse response $h(n)$ is given by $h(n) = a^{-n}u(-n);\ 0 < a < 1$ .

***Solution:*** The impulse response is

$$h(n) = a^{-n}u(-n);\ 0 < a < 1$$

$$H(z) = \frac{1}{1-az} = -\frac{1}{a}\frac{1}{z-(1/a)}$$

We have to find the step response

$$x(n) = u(n) \quad \text{and} \quad H(z) = \frac{z}{z-1}$$

The step response of the system is given by

$$y(n) = x(n) * h(n)$$

$$Y(z) = X(z)\,H(z) = \left(-\frac{1}{a}\right)\frac{z}{z-1}\,\frac{1}{z-(1/a)} = \frac{1}{1-a}\left[\frac{z}{z-1} - \frac{z}{z-(1/a)}\right]$$

So the step response is

$$y(n) = \frac{1}{1-a}\left[u(n) - \left(\frac{1}{a}\right)^n u(n)\right]$$

**EXAMPLE 8** The step response of an LTI system is

***Solution:*** We have $s(n) = h(n) * u(n)$

$$\therefore \qquad S(z) = H(z)U(z) = H(z)\,\frac{z}{z-1}$$

Given $$s(n) = \left(\frac{1}{3}\right)^{n-2} u(n+2)$$

$$S(z) = \sum_{n=-\infty}^{\infty}\left(\frac{1}{3}\right)^{n-2} u(n+2)\,z^{-n} = 3^2\sum_{n=-2}^{\infty}\left(\frac{1}{3z}\right)^n$$

$$= 3^2\,\frac{\left(\frac{1}{3z}\right)^{-2}}{1-\frac{1}{3z}} = \frac{3^4 z^2}{1-\frac{1}{3}z^{-1}} = \frac{81z^3}{\left(z-\frac{1}{3}\right)}$$

The system function $H(z)$ is

$$H(z) = S(z)\,\frac{z-1}{z} = \frac{81z^3}{\left(z-\frac{1}{3}\right)}\,\frac{z-1}{z} = \frac{81z^2(z-1)}{\left(z-\frac{1}{3}\right)} = \frac{81z^3}{z-\frac{1}{3}} - \frac{81z^2}{z-\frac{1}{3}} = 81z^2\,\frac{z}{z-\frac{1}{3}} - 81z\,\frac{z}{z-\frac{1}{3}}$$

The impulse response of the system is

$$h(n) = 81\left(\frac{1}{3}\right)^{n+2} u(n+2) - 81\left(\frac{1}{3}\right)^{n+1} u(n+1) = 9\left(\frac{1}{3}\right)^n u(n+2) - 27\left(\frac{1}{3}\right)^n u(n+1)$$

# UNIT II

## Discrete Fourier Transforms

**INTRODUCTION** :The DFT of a discrete-time signal $x(n)$ is a finite duration discrete frequency sequence. The DFT sequence is denoted by $X(k)$. The DFT is obtained by sampling one period of the Fourier transform $X(W)$ of the signal $x(n)$ at a finite number of frequency points. This sampling is conventionally performed at $N$ equally spaced points in the period $0 \leq w \leq 2w$ or at $w_k = 2\pi k/N$; $0 \leq k \leq N-1$. We can say that DFT is used for transforming discrete-time sequence $x(n)$ of finite length into discrete frequency sequence $X(k)$ of finite length. The DFT is important for two reasons. First it allows us to determine the frequency content of a signal, that is to perform spectral analysis. The second application of the DFT is to perform filtering operation in the frequency domain. Let $x(n)$ be a discrete-time sequence with Fourier transform $X(W)$, then the DFT of $x(n)$ denoted by $X(k)$ is defined as

$$X(k) = X(\omega)\big|_{\omega = (2\pi k/N)}; \quad \text{for } k = 0, 1, 2, \ldots, N-1$$

The DFT of $x(n)$ is a sequence consisting of $N$ samples of $X(k)$. The DFT sequence starts at $k = 0$, corresponding to $w = 0$, but does not include $k = N$ corresponding to $w = 2\pi$ (since the sample at $w = 0$ is same as the sample at $w = 2\pi$). Generally, the DFT is defined as

**DFT** The $N$-point DFT of a finite duration sequence $x(n)$ of length $L$, where $N \geq L$ is defined as:

$$\text{DFT}\{x(n)\} = X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N} = \sum_{n=0}^{N-1} x(n) W_N^{nk}; \quad \text{for } k = 0, 1, 2, \ldots, N-1$$

**IDFT** The Inverse Discrete Fourier transform (IDFT) of the sequence $X(k)$ of length $N$ is defined as:

$$\text{IDFT}\{X(k)\} = x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi nk/N} = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-nk}; \quad \text{for } n = 0, 1, 2, \ldots, N-1$$

where $W_N = e^{-j(2\pi/N)}$ is called the twiddle factor.

The $N$-point DFT pair $x(n)$ and $X(k)$ is denoted as:

$$x(n) \xleftrightarrow[N]{\text{DFT}} X(k)$$

**EXAMPLE 2.1** (a) Find the 4-point DFT of $x(n) = \{1, -1, 2, -2\}$ directly.
(b) Find the IDFT of $X(k) = \{4, 2, 0, 4\}$ directly.

***Solution:***

(a) Given sequence is $x(n) = \{1, -1, 2, -2\}$. Here the DFT $X(k)$ to be found is $N$ =4-point and length of the sequence $L = 4$. So no padding of zeros is required. We know that the DFT $\{x(n)\}$ is given by

$$X(k)=\sum_{n=0}^{N-1}x(n)\,W_N^{nk}=\sum_{n=0}^{N-1}x(n)\,e^{-j(2\pi/N)nk}=\sum_{n=0}^{3}x(n)\,e^{-j(\pi/2)nk},\quad k=0,1,2,3$$

$$X(0)=\sum_{n=0}^{3}x(n)\,e^{0}=x(0)+x(1)+x(2)+x(3)=1-1+2-2=0$$

$$X(1)=\sum_{n=0}^{3}x(n)e^{-j(\pi/2)n}=x(0)+x(1)\,e^{-(j\pi/2)}+x(2)\,e^{-j\pi}+x(3)\,e^{-j(3\pi/2)}$$

$$=1+(-1)(0-j)+2(-1-j0)-2(0+j)$$

$$=-1-j$$

$$X(2)=\sum_{n=0}^{3}x(n)e^{-j\pi n}=x(0)+x(1)\,e^{-j\pi}+x(2)\,e^{-j2\pi}+x(3)\,e^{-j3\pi}$$

$$=1-1(-1-j0)+2(1-j0)-2(-1-j0)=6$$

$$X(3)=\sum_{n=0}^{3}x(n)e^{-j(3\pi/2)n}=x(0)+x(1)\,e^{-j(3\pi/2)}+x(2)\,e^{-j3\pi}+x(3)\,e^{-j(9\pi/2)}$$

$$=1-1(0+j)+2(-1-j0)-2(0-j)=-1+j$$

$$X(k)=\{0,\ -1-j,\ 6,\ -1+j\}$$

(b) Given DFT is $X(k) = \{4, 2, 0, 4\}$. The IDFT of $X(k)$, i.e. $x(n)$ is given by

$$x(n)=\frac{1}{N}\sum_{k=0}^{N-1}X(k)\,W_N^{-nk}=\frac{1}{N}\sum_{k=0}^{N-1}X(k)\,e^{j(2\pi/N)nk}$$

i.e. $$x(n)=\frac{1}{4}\sum_{k=0}^{3}X(k)e^{j(\pi/2)nk}$$

$$\therefore\quad x(0)=\frac{1}{4}\sum_{k=0}^{3}X(k)e^{0}=\frac{1}{4}[X(0)+X(1)+X(2)+X(3)]$$

$$=\frac{1}{4}[4+2+0+4]=2.5$$

$$x(1)=\frac{1}{4}\sum_{k=0}^{3}X(k)\,e^{j(\pi/2)k}=\frac{1}{4}\left[X(0)+X(1)\,e^{j(\pi/2)}+X(2)\,e^{j\pi}+X(3)e^{j(3\pi/2)}\right]$$

$$=\frac{1}{4}[4+2(0+j)+0+4(0-j)]=1-j0.5$$

$$x(2)=\frac{1}{4}\sum_{k=0}^{3}X(k)\,e^{j\pi k}=\frac{1}{4}\left[X(0)+X(1)\,e^{j\pi}+X(2)\,e^{j2\pi}+X(3)e^{j3\pi}\right]$$

$$=\frac{1}{4}[4+2(-1+j0)+0+4(-1+j0)]=-0.5$$

$$x(3)=\frac{1}{4}\sum_{k=0}^{3}X(k)\,e^{j(3\pi/2)k}=\frac{1}{4}\left[X(0)+X(1)\,e^{j(3\pi/2)}+X(2)\,e^{j3\pi}+X(3)e^{j(9\pi/2)}\right]$$

$$=\frac{1}{4}[4+2(0-j)+0+4(0+j)]=1+j0.5$$

$$x_3(n)=\{2.5,\ 1-j0.5,\ -0.5,\ 1+j0.5\}$$

**EXAMPLE 2.2** (a) Find the 4-point DFT of $x(n) = \{1, -2, 3, 2\}$.
(b) Find the IDFT of $X(k) = \{1, 0, 1, 0\}$.

***Solution:***

(a) Given $x(n) = \{1, -2, 3, 2\}$.

Here $N = 4$, $L = 4$. The DFT of $x(n)$ is $X(k)$.

$$\therefore \quad X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} = \sum_{n=0}^{3} x(n) e^{-j(2\pi/4)nk} = \sum_{n=0}^{3} x(n) e^{-j(\pi/2)nk}, \quad k = 0, 1, 2, 3$$

$$X(0) = \sum_{n=0}^{3} x(n) e^{0} = x(0) + x(1) + x(2) + x(3) = 1 - 2 + 3 + 2 = 4$$

$$X(1) = \sum_{n=0}^{3} x(n) e^{-j(\pi/2)n} = x(0) + x(1) e^{-j(\pi/2)} + x(2) e^{-j\pi} + x(3) e^{-j(3\pi/2)}$$

$$= 1 - 2(0 - j) + 3(-1 - j0) + 2(0 + j) = -2 + j4$$

$$X(2) = \sum_{n=0}^{3} x(n) e^{-j\pi n} = x(0) + x(1) e^{-j\pi} + x(2) e^{-j2\pi} + x(3) e^{-j3\pi}$$

$$= 1 - 2(-1 - j0) + 3(1 - j0) + 2(-1 - j0) = 4$$

$$X(3) = \sum_{n=0}^{3} x(n) e^{-j(3\pi/2)n} = x(0) + x(1) e^{-j(3\pi/2)} + x(2) e^{-j3\pi} + x(3) e^{-j(9\pi/2)}$$

$$= 1 - 2(0 + j) + 3(-1 - j0) + 2(0 - j) = -2 - j4$$

$$\therefore \quad X(k) = \{4, -2 + j4, 4, -2 - j4\}$$

(b) Given $X(k) = \{1, 0, 1, 0\}$

Let the IDFT of $X(k)$ be $x(n)$.

$$\therefore \quad x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-nk} = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j(2\pi/N)nk}$$

$$x(0) = \frac{1}{4} \sum_{k=0}^{3} X(k) e^{0} = \frac{1}{4} [X(0) + X(1) + X(2) + X(3)] = \frac{1}{4} [1 + 0 + 1 + 0] = 0.5$$

$$x(1) = \frac{1}{4} \sum_{k=0}^{3} X(k) e^{j(\pi/2)k} = \frac{1}{4} \left[ X(0) + X(1) e^{j(\pi/2)} + X(2) e^{j\pi} + X(3) e^{j(3\pi/2)} \right]$$

$$= \frac{1}{4} [1 + 0 + e^{j\pi} + 0] = \frac{1}{4} [1 + 0 - 1 + 0] = 0$$

$$x(2) = \frac{1}{4} \sum_{k=0}^{3} X(k) e^{j\pi k} = \frac{1}{4} \left[ X(0) + X(1) e^{j\pi} + X(2) e^{j2\pi} + X(3) e^{j3\pi} \right]$$

$$= \frac{1}{4} [1 + 0 + e^{j2\pi} + 0] = \frac{1}{4} [1 + 0 + 1 + 0] = 0.5$$

$$x(3) = \frac{1}{4} \sum_{k=0}^{3} X(k) e^{j(3\pi/2)k} = \frac{1}{4} \left[ X(0) + X(1) e^{j(3\pi/2)} + X(2) e^{j3\pi} + X(3) e^{j(9\pi/2)} \right]$$

$$= \frac{1}{4} [1 + 0 + e^{j3\pi} + 0] = \frac{1}{4} [1 + 0 - 1 + 0] = 0$$

The IDFT of $X(k) = \{1, 0, 1, 0\}$ is $x(n) = \{0.5, 0, 0.5, 0\}$.

**EXAMPLE 2.3** Compute the DFT of the 3-point sequence $x(n) = \{2, 1, 2\}$. Using the same sequence, compute the 6-point DFT and compare the two DFTs.

***Solution:*** The given 3-point sequence is $x(n) = \{2, 1, 2\}$, $N = 3$.

$$\text{DFT}\, x(n) = X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} = \sum_{n=0}^{2} x(n) e^{-j(2\pi/3)nk}, \quad k = 0, 1, 2$$

$$= x(0) + x(1) e^{-j(2\pi/3)k} + x(2) e^{-j(4\pi/3)k}$$

$$= 2 + \left( \cos\frac{2\pi}{3}k - j \sin\frac{2\pi}{3}k \right) + 2\left( \cos\frac{4\pi}{3}k - j \sin\frac{4\pi}{3}k \right)$$

When $k = 0$, $X(k) = X(0) = 2 + 1 + 2 = 5$

When $k = 1$,
$$X(k) = X(1) = 2 + \left( \cos\frac{2\pi}{3} - j \sin\frac{2\pi}{3} \right) + 2\left( \cos\frac{4\pi}{3} - j \sin\frac{4\pi}{3} \right)$$

$$= 2 + (-0.5 - j0.866) + 2(-0.5 + j0.866)$$

$$= 0.5 + j0.866$$

When $k = 2$,
$$X(k) = X(2) = 2 + \left( \cos\frac{4\pi}{3} - j \sin\frac{4\pi}{3} \right) + 2\left( \cos\frac{8\pi}{3} - j \sin\frac{8\pi}{3} \right)$$

$$= 2 + (-0.5 + j0.866) + 2(-0.5 - j0.866)$$

$$= 0.5 - j0.866$$

$\therefore$ 3-point DFT of $x(n) = X(k) = \{5, 0.5 + j0.866, 0.5 - j0.866\}$

To compute the 6-point DFT, convert the 3-point sequence $x(n)$ into 6-point sequence by padding with zeros.

$$x(n) = \{2, 1, 2, 0, 0, 0\}, \quad N = 6$$

$$\text{DFT}\{x(n)\} = X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} = \sum_{n=0}^{5} x(n) e^{-j(2\pi/N)nk}, \quad k = 0, 1, 2, 3, 4, 5$$

$$= x(0) + x(1) e^{-j(2\pi/6)k} + x(2) e^{-j(4\pi/6)k} + x(3) e^{-j(6\pi/6)k} + x(4) e^{-j(8\pi/6)k}$$

$$+ x(5) e^{-j(10\pi/6)k}$$

$$= 2 + e^{-j(\pi/3)k} + 2e^{-j(2\pi/3)k}$$

When $k = 0$, $X(0) = 2 + 1 + 2 = 5$

When $k = 1$,
$$X(1) = 2 + e^{-j(\pi/3)} + 2e^{-j(2\pi/3)}$$
$$= 2 + (0.5 - j0.866) + 2(-0.5 - j0.866) = 1.5 - j2.598$$

When $k = 2$,
$$X(2) = 2 + e^{-j(2\pi/3)} + 2e^{-j(4\pi/3)}$$
$$= 2 + (-0.5 - j0.866) + 2(-0.5 + j0.866) = 0.5 + j0.866$$

When $k = 3$,
$$X(3) = x(0) + x(1)e^{-j(3\pi/3)} + x(2)e^{-j(6\pi/3)}$$
$$= 2 + (\cos\pi - j\sin\pi) + 2(\cos 2\pi - j\sin 2\pi)$$
$$= 2 - 1 + 2 = 3$$

When $k = 4$,
$$X(4) = x(0) + x(1)e^{-j(4\pi/3)} + x(2)e^{-j(8\pi/3)}$$
$$= 2 + \left(\cos\frac{4\pi}{3} - j\sin\frac{4\pi}{3}\right) + 2\left(\cos\frac{8\pi}{3} - j\sin\frac{8\pi}{3}\right)$$
$$= 2 + (-0.5 + j0.866) + 2(-0.5 - j0.866)$$
$$= 0.5 - j0.866$$

When $k = 5$,
$$X(5) = x(0) + x(1)e^{-j(5\pi/3)} + x(2)e^{-j(10\pi/3)}$$
$$= 2 + \left(\cos\frac{5\pi}{3} - j\sin\frac{5\pi}{3}\right) + 2\left(\cos\frac{10\pi}{3} - j\sin\frac{10\pi}{3}\right)$$
$$= 2 + (0.5 - j0.866) + 2(-0.5 + j0.866) = 1.5 + j0.866$$

Tabulating the above 3-point and 6-point DFTs, we have

| DFT | $X(0)$ | $X(1)$ | $X(2)$ | $X(3)$ | $X(4)$ | $X(5)$ |
|---|---|---|---|---|---|---|
| 3-point | 5 | $0.5 + j0.866$ | $0.5 - j0.866$ | – | – | – |
| 6-point | 5 | $1.5 - j2.598$ | $0.5 + j0.866$ | 3 | $0.5 - j0.866$ | $1.5 + j0.866$ |

## MATRIX FORMULATION OF THE DFT AND IDFT

If we let $W_N = e^{-j(2\pi/N)}$, the defining relations for the DFT and IDFT may be written as:

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}, \qquad k = 0, 1, \ldots, N-1$$

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)W_N^{-nk}, \qquad n = 0, 1, 2, \ldots, N-1$$

The first set of $N$ DFT equations in $N$ unknowns may be expressed in matrix form as:

$$\mathbf{X} = \mathbf{W_N x}$$

Here $\mathbf{X}$ and $\mathbf{x}$ are $N \times 1$ matrices, and $\mathbf{W_N}$ is an $N \times N$ square matrix called the DFT matrix. The full matrix form is described by

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ \vdots \\ X(N-1) \end{bmatrix} = \begin{bmatrix} W_N^0 & W_N^0 & W_N^0 & \cdots & W_N^0 \\ W_N^0 & W_N^1 & W_N^2 & \cdots & W_N^{(N-1)} \\ W_N^0 & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ W_N^0 & W_N^{(N-1)} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(N-1) \end{bmatrix}$$

# THE IDFT FROM THE MATRIX FORM

The matrix **x** may be expressed in terms of the inverse of $\mathbf{W_N}$ as:

$$\mathbf{x} = \mathbf{W_N^{-1}}\ \mathbf{x}$$

$\mathbf{W_N}$ is called the IDFT matrix. We may also obtain **x** directly from the IDFT relation in matrix form, where the change of index from $n$ to $k$ and the change in the sign of the exponent in $e^{j(2\pi/N)nk}$ lead to the conjugate transpose of $\mathbf{W_N}$. We then have

$$\mathbf{x} = \frac{1}{\mathbf{N}}\left[\mathbf{W_N^*}\right]^T \mathbf{x}$$

**EXAMPLE 2.4** Find the DFT of the sequence $x(n) = \{1, 2, 1, 0\}$
***Solution:*** The DFT $X(k)$ of the given sequence $x(n) = \{1, 2, 1, 0\}$ may be obtained by solving the matrix product as follows. Here $N = 4$.

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} W_N^0 & W_N^0 & W_N^0 & W_N^0 \\ W_N^0 & W_N^1 & W_N^2 & W_N^3 \\ W_N^0 & W_N^2 & W_N^4 & W_N^6 \\ W_N^0 & W_N^3 & W_N^6 & W_N^9 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 4 \\ -j2 \\ 0 \\ j2 \end{bmatrix}$$

The result is DFT $\{x(n)\} = X(k) = \{4, -j2, 0, j2\}$.

**EXAMPLE 2.5** Find the DFT of $x(n) = \{1, –1, 2, –2\}$.
***Solution:*** The DFT, $X(k)$ of the given sequence $x(n) = \{1, –1, 2, –2\}$ can be determined using matrix as shown below.

$$X(k) = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^4 & W_4^6 \\ W_4^0 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 2 \\ -2 \end{bmatrix} = \begin{bmatrix} 0 \\ -1-j \\ 6 \\ -1+j \end{bmatrix}$$

$\therefore$ DFT $\{x(n)\} = X(k) = \{0, -1-j, 6, -1+j\}$

**EXAMPLE 2.6.** Find the 4-point DFT of $x(n) = \{1, –2, 3, 2\}$.
***Solution:*** Given $x(n) = \{1, –2, 3, 2\}$, the 4-point DFT$\{x(n)\} = X(k)$ is determined using matrix as shown below.

$$\text{DFT}\{x(n)\} = X(k) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ -2 \\ 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 4 \\ -2+j4 \\ 4 \\ -2-j4 \end{bmatrix}$$

DFT $\{x(n)\} = X(k) = \{4, -2+j4, 4, -2-j4\}$

**EXAMPLE 2.6** Find the IDFT of $X(k)=\{4, -j2, 0, j2\}$ using DFT.
***Solution:*** Given $X(k) = \{4, -j2, 0, j2\}$ _ $X^*(k) = \{4, j2, 0, -j2\}$
The IDFT of $X(k)$ is determined using matrix as shown below.
To find IDFT of $X(k)$ first find $X^*(k)$, then find DFT of $X^*(k)$, then take conjugate of DFT $\{X^*(k)\}$ and divide by $N$.

$$\text{DFT}\{X^*(k)\} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 4 \\ j2 \\ 0 \\ -j2 \end{bmatrix} = \begin{bmatrix} 4 \\ 8 \\ 4 \\ 0 \end{bmatrix}$$

$$\therefore \quad \text{IDFT}[X(k)] = x(n) = \frac{1}{4}[4, 8, 4, 0]^* = \frac{1}{4}[4, 8, 4, 0] = [1, 2, 1, 0]$$

**EXAMPLE 2.7**Find the IDFT of $X(k) = \{4, 2, 0, 4\}$ using DFT.
***Solution:*** Given $X(k) = \{4, 2, 0, 4\}$
$X^*(k) = \{4, 2, 0, 4\}$
The IDFT of $X(k)$ is determined using matrix as shown below.
To find IDFT of $X(k)$, first find $X^*(k)$, then find DFT of $X^*(k)$, then take conjugate of DFT $\{X^*(k)\}$ and divide by $N$

$$\text{DFT}[X^*(k)] = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ 0 \\ 4 \end{bmatrix} = \begin{bmatrix} 10 \\ 4 + j2 \\ -2 \\ 4 - j2 \end{bmatrix}$$

$$\therefore \quad \text{IDFT}\{X(k)\} = x(n) = \frac{1}{4}[10, 4 + j2, -2, 4 - j2]^* = \{2.5, 1 - j0.5, -0.5, 1 + j0.5\}$$

**EXAMPLE2.8** Find the IDFT of $X(k) = \{1, 0, 1, 0\}$.
***Solution:*** Given $X(k) = \{1, 0, 1, 0\}$, the IDFT of $X(k)$, i.e. $x(n)$ is determined using matrix as shown below.

$$X^*(k) = \{1, 0, 1, 0\}^* = \{1, 0, 1, 0\}$$

$$\text{DFT}\{X^*(k)\} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 2 \\ 0 \end{bmatrix}$$

$$\therefore \quad \text{IDFT}\{X(k)\} = x(n) = \frac{1}{4}[\text{DFT}\{X^*(k)\}]^* = \frac{1}{4}\{2, 0, 2, 0\} = \{0.5, 0, 0.5, 0\}$$

## PROPERTIES OF DFT

Like the Fourier and Z-transforms, the DFT has several important properties that are used to process the finite duration sequences. Some of those properties are discussed as follows

### Periodicity:

If a sequence $x(n)$ is periodic with periodicity of $N$ samples, then $N$-point DFT of the sequence, $X(k)$ is also periodic with periodicity of $N$ samples.
Hence, if $x(n)$ and $X(k)$ are an $N$-point DFT pair, then

$$x(n+N) = x(n) \qquad \text{for all } n$$
$$X(k+N) = X(k) \qquad \text{for all } k$$

*Proof:* By definition of DFT, the $(k + N)$th coefficient of $X(k)$ is given by

$$X(k+N) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi n(k+N)/N} = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N} e^{-j2\pi nN/N}$$

But $e^{-j2\pi n} = 1$ for all $n$ (Here $n$ is an integer)

$$\therefore \qquad X(k+N) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N} = X(k)$$

## Linearity

If $x_1(n)$ and $x_2(n)$ are two finite duration sequences and if

$$\text{DFT } \{x_1(n)\} = X_1(k)$$

and

$$\text{DFT } \{x_2(n)\} = X_2(k)$$

Then for any real valued or complex valued constants $a$ and $b$,

$$\text{DFT}\{ax_1(n) + bx_2(n)\} = aX_1(k) + bX_2(k)$$

*Proof:*

$$\text{DFT}\{ax_1(n) + bx_2(n)\} = \sum_{n=0}^{N-1} [ax_1(n) + bx_2(n)]\, e^{-j2\pi nk/N}$$

$$= a\sum_{n=0}^{N-1} x_1(n) e^{-j2\pi nk/N} + b\sum_{n=0}^{N-1} x_2(n) e^{-j2\pi nk/N}$$

$$= aX_1(k) + bX_2(k)$$

## DFT of Even and Odd Sequences

The DFT of an even sequence is purely real, and the DFT of an odd sequence is purely imaginary. Therefore, DFT can be evaluated using cosine and sine transforms for even and odd sequences respectively.

For even sequence, $X(k) = \sum_{n=0}^{N-1} x(n) \cos\left(\frac{2\pi nk}{N}\right)$

For odd sequence, $X(k) = \sum_{n=0}^{N-1} x(n) \sin\left(\frac{2\pi nk}{N}\right)$

## Time Reversal of the Sequence

The time reversal of an $N$-point sequence $x(n)$ is obtained by wrapping the sequence $x(n)$ around the circle in the clockwise direction. It is denoted as $x[(-n), \text{mod } N]$ and

$$x[(-n), \text{ mod } N] = x(N - n), \qquad 0 \le n \le N - 1$$

If DFT $\{x(n)\} = X(k)$, then

$$\text{DFT}\{x(-n), \text{mod } N\} = \text{DFT}\{x(N-n)\}$$

$$= X[(-k), \text{mod } N] = X(N-k)$$

*Proof:* $\text{DFT}\{x(N-n)\} = \sum_{n=0}^{N-1} x(N-n) e^{-j2\pi nk/N}$

Changing index from $n$ to $m$, where $m = N - n$, we have $n = N - m$.

Now,
$$\text{DFT}\{x(N-n)\} = \sum_{m=0}^{N-1} x(m)e^{-j2\pi k(N-m)/N}$$
$$= \sum_{m=0}^{N-1} x(m)e^{-j(2\pi/N)kN}e^{j(2\pi/N)km}$$
$$= \sum_{m=0}^{N-1} x(m)e^{j(2\pi/N)km}$$
$$= \sum_{m=0}^{N-1} x(m)e^{j(2\pi/N)km}e^{-j2\pi m}$$
$$= \sum_{m=0}^{N-1} x(m)e^{-j2\pi m[(N-k)/N]} = X(N-k)$$

## Circular Frequency Shift

If
$$\text{DFT}\{x(n)\} = X(k)$$
Then,
$$\text{DFT}\{x(n)\ e^{j2\pi ln/N}\} = X\{(k-l), (\text{mod } N)\}$$

*Proof:*
$$\text{DFT}\{x(n)e^{j2\pi ln/N}\} = \sum_{n=0}^{N-1} x(n)e^{j2\pi ln/N}e^{-j2\pi kn/N}$$
$$= \sum_{n=0}^{N-1} x(n)e^{-j2\pi n(k-l)/N}$$
$$= \sum_{n=0}^{N-1} x(n)e^{-j2\pi n(N+k-l)/N}$$
$$= X(N+k-l) = X[(k-l), (\text{mod } N)]$$

## Complex Conjugate Property

If $\text{DFT}\ \{x(n)\} = X(k)$

Then $\text{DFT}\ \{x^*(n)\} = X^*(N-k) = X^*[(-k), \text{mod}\ N]$

*Proof:*

$$\text{DFT}\{x^*(n)\} = \sum_{n=0}^{N-1} x^*(n) e^{-j2\pi kn/N}$$

$$= \left[\sum_{n=0}^{N-1} x(n) e^{j2\pi nk/N}\right]^* = \left[\sum_{n=0}^{N-1} x(n) e^{-j2\pi n(N-k)/N}\right]^* = X^*(N-k)$$

$$\text{DFT}\{x^*(N-n)\} = X^*(k)$$

*Proof:*

$$\text{IDFT}\{X^*(k)\} = \frac{1}{N}\sum_{k=0}^{N-1} X^*(k) e^{j2\pi kn/N}$$

$$= \frac{1}{N}\left[\sum_{k=0}^{N-1} X(k) e^{-j2\pi kn/N}\right]^* = \frac{1}{N}\left[\sum_{k=0}^{N-1} X(k) e^{j2\pi k(N-n)/N}\right]^* = x^*(N-n)$$

## DFT of Delayed Sequence (Circular time shift of a sequence)

Let $x(n)$ be a discrete sequence, and $x'(n)$ be a delayed or shifted sequence of $x(n)$ by $n_0$ units of time.

If $\text{DFT}\ \{x(n)\} = X(k)$

Then, $\text{DFT}\ \{x'(n)\} = \text{DFT}\{x[(n-n_0), \text{mod}\ N]\} = X(k)\, e^{-j2\pi n_0 k/N}$

*Proof:* By the definition of IDFT,

$$\text{IDFT}\ \{X(k)\} = x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k) e^{j\frac{2\pi}{N}nk}$$

Replacing $n$ by $n-n_0$, we have

$$x(n-n_0) = \frac{1}{N}\sum_{k=0}^{N-1} X(k) e^{j\frac{2\pi}{N}(n-n_0)k}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1}\left[X(k)\, e^{-j\frac{2\pi}{N}n_0 k}\right] e^{j\frac{2\pi}{N}nk}$$

$$= \text{IDFT}\left[X(k)\, e^{-j\frac{2\pi}{N}n_0 k}\right]$$

On taking DFT on both sides, we get

$$\text{DFT}\ [x(n-n_0)] = X(k) e^{-j\frac{2\pi}{N}kn_0}$$

## DFT of Real Valued Sequences

Let $x(n)$ be a real sequence. By definition of DFT,

$$\text{DFT}\ \{x(n)\} = X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}kn}$$

$$= \sum_{n=0}^{N-1} x(n)\left( \cos\frac{2\pi}{N}nk - j\sin\frac{2\pi}{N}nk \right)$$

$$= \sum_{n=0}^{N-1} x(n)\cos\frac{2\pi}{N}nk - j\sum_{n=0}^{N-1} x(n)\sin\frac{2\pi}{N}nk$$

Also $X(k) = X_R(k) + jX_I(k)$

Therefore, we can say

Real part $$X_R(k) = \sum_{n=0}^{N-1} x(n)\cos\left(\frac{2\pi}{N}nk\right), \quad \text{for } 0 \le k \le N-1$$

Imaginary part $$X_I(k) = -\sum_{n=0}^{N-1} x(n)\sin\left(\frac{2\pi}{N}nk\right), \quad \text{for } 0 \le k \le N-1$$

When $x(n)$ is real, then $X(k)$ will have the following features:

(a) $X(k)$ has complex conjugate symmetry, i.e. $X(k) = X^*(N-k)$
(b) Real component is even function, i.e. $X_R(k) = X_R(N-k)$
(c) Imaginary component is odd function, i.e. $X_I(k) = -X_I(N-k)$
(d) Magnitude function is even function, i.e. $|X(k)| = |X(N-k)|$
(e) Phase function is odd function, i.e. $\angle X(k) = -\angle X(N-k)$
(f) If $x(n) = x(-n)$ (even sequence), then $X(k)$ is purely real.
(g) If $x(n) = -x(-n)$ (odd sequence), then $X(k)$ is purely imaginary.

## Multiplication of Two Sequences

If $\text{DFT}\ [x_1(n)] = X_1(k)$

and $\text{DFT}\ [x_2(n)] = X_2(k)$

Then $$\text{DFT}[x_1(n)\,x_2(n)] = \frac{1}{N}[X_1(k) \oplus X_2(k)]$$

## Circular Convolution of Two Sequences

The convolution property of DFT says that, the multiplication of DFTs of two sequences is equivalent to the DFT of the circular convolution of the two sequences.

Let DFT $[x_1(n)] = X_1(k)$ and DFT $[x_2(n)] = X_2(k)$, then by the convolution property $X_1(k)X_2(k) = \text{DFT}\{x_1(n) \oplus x_2(n)\}$.

*Proof:* Let $x_1(n)$ and $x_2(n)$ be two finite duration sequences of length $N$. The $N$-point DFTs of the two sequences are:

$$X_1(k) = \sum_{n=0}^{N-1} x_1(n) e^{-j\frac{2\pi}{N}nk}, \quad k = 0, 1, ..., N-1$$

$$X_2(k) = \sum_{l=0}^{N-1} x_2(l) e^{-j\frac{2\pi}{N}lk}, \quad k = 0, 1, ..., N-1$$

On multiplying the above two DFTs, we obtain the result as another DFT, say, $X_3(k)$. Now, $X_3(k)$ will be $N$-point DFT of a sequence $x_3(m)$.

$$\therefore \qquad X_3(k) = X_1(k)X_2(k) \quad \text{and} \quad \text{IDFT}\{X_3(k)\} = x_3(m)$$

By the definition of IDFT,

$$x_3(m) = \frac{1}{N}\sum_{k=0}^{N-1} X_3(k) e^{j\frac{2\pi}{N}mk}, \quad m = 0, 1, 2, ..., N-1$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} X_1(k) X_2(k) e^{j\frac{2\pi}{N}mk}$$

Using the above equations for $X_1(k)$ and $X_2(k)$, the equation for $x_3(m)$ is:

$$x_3(m) = \frac{1}{N}\sum_{k=0}^{N-1}\sum_{n=0}^{N-1} x_1(n) e^{-j\frac{2\pi}{N}nk} \sum_{l=0}^{N-1} x_2(l) e^{-j\frac{2\pi}{N}lk} e^{j\frac{2\pi}{N}mk}$$

$$= \frac{1}{N}\sum_{n=0}^{N-1} x_1(n) \sum_{l=0}^{N-1} x_2(l) \sum_{k=0}^{N-1} e^{j\frac{2\pi}{N}k(m-n-l)}$$

Let $m - n - l = PN$ where $P$ is an integer.

$$\therefore \qquad e^{j\frac{2\pi}{N}k(m-n-l)} = e^{j\frac{2\pi}{N}kPN} = e^{j2\pi kP} = (e^{j2\pi P})^k$$

We know that

$$\sum_{k=0}^{N-1} e^{j\frac{2\pi k}{N}(m-n-l)} = \sum_{k=0}^{N-1} (e^{j2\pi P})^k = \sum_{k=0}^{N-1} 1^k = \sum_{k=0}^{N-1} 1 = N$$

Therefore, the above equation for $x_3(m)$ can be written as:

$$x_3(m) = \frac{1}{N} \sum_{n=0}^{N-1} x_1(n) \sum_{l=0}^{N-1} x_2(l) N = \sum_{n=0}^{N-1} x_1(n) \sum_{l=0}^{N-1} x_2(l)$$

If $x_2(l)$ is a periodic sequence with periodicity of $N$ samples, then $x_2(l \pm PN) = x_2(l)$

Here $$m - n - l = PN$$

$\therefore$ $$l = m - n - PN$$

$\therefore$ $$x_2(l) = x_2(m - n - PN) = x_2(m - n) = x_2[(m - n), \text{ mod } N]$$

Therefore, $x_3(m)$ can be

$$x_3(m) = \sum_{n=0}^{N-1} x_1(n) \sum_{n=0}^{N-1} x_2(m-n) = \sum_{n=0}^{N-1} x_1(n)\, x_2(m-n)$$

Replacing $m$ by $n$ and $n$ by $k$, we have $x_3(n) = \sum_{k=0}^{N-1} x_1(k)\, x_2(n-k)$

*Note:* For simplicity, $x_2[(m - n), \text{ mod } N]$ is represented as $x_2(m - n)$.

The equation for $x_2(l)$ is in the form of convolution sum. Since the equation for $x_2(l)$ involves the index $[(m - n), \text{ mod } N]$, it is called circular convolution.

Hence, we conclude that multiplication of the DFTs of two sequences is equivalent to the DFT of the circular convolution of the two sequences.

$$X_1(k)\, X_2(k) = \text{DFT}\, \{x_1(n) \oplus x_2(n)\}$$

## Parseval's Theorem

Parseval's theorem says that the DFT is an energy-conserving transformation and allows us to find the signal energy either from the signal or its spectrum. This implies that the sum of squares of the signal samples is related to the sum of squares of the magnitude of the DFT samples.

If $$\text{DFT}\ \{x_1(n)\} = X_1(k)$$

and $$\text{DFT}\ \{x_2(n)\} = X_2(k)$$

Then $$\sum_{n=0}^{N-1} x_1(n)\, x_2^*(n) = \frac{1}{N} \sum_{k=0}^{N-1} X_1(k)\, X_2^*(k)$$

## Circular Correlation

For complex valued sequences $x(n)$ and $y(n)$,

If $$\text{DFT}\ \{x(n)\} = X(k)$$

and $$\text{DFT}\ \{y(n)\} = Y(k)$$

Then $$\text{DFT}\ \{r_{xy}(l)\} = \text{DFT}\left[\sum_{n=0}^{N-1} x(n)\, y^*((n-l), \text{mod } N)\right] = X(k)\, Y^*(k)$$

where, $r_{xy}(l)$ is the circular cross correlation sequence. The properties of DFT are summarized in Table 6.3.

# Linear Convolution using DFT

The DFT supports only circular convolution. When two numbers of $N$-point sequence are circularly convolved, it produces another $N$-point sequence. For circular convolution, one of the sequence should be periodically extended. Also the resultant sequence is periodic with period $N$. The linear convolution of two sequences of length $N_1$ and $N_2$ produces an output sequence of length $N_1 + N_2 - 1$. To perform linear convolution using DFT, both the sequences should be converted to $N_1 + N_2 - 1$ sequences by padding with zeros. Then take $N_1 + N_2 - 1$-point DFT of both the sequences and determine the product of their DFTs. The resultant sequence is given by the IDFT of the product of DFTs. [Actually the response is given by the circular convolution of the $N_1 + N_2 - 1$ sequences]. Let $x(n)$ be an $N_1$-point sequence and $h(n)$ be an $N_2$-point sequence. The linear convolution of $x(n)$ and $h(n)$ produces a sequence $y(n)$ of length $N_1 + N_2 - 1$. So pad $x(n)$ with $N_2 - 1$ zeros and $h(n)$ with $N_1 - 1$ zeros and make both of them of length $N_1 + N_2 - 1$. Let $X(k)$ be an $N_1 + N_2 - 1$-point DFT of $x(n)$, and $H(k)$ be an $N_1 + N_2 - 1$-point DFT of $h(n)$. Now, the sequence $y(n)$ is given by the inverse DFT of the product $X(k)$ $H(k)$.

$$y(n) = \text{IDFT}\{X(k)H(k)\}$$

This technique of convolving two finite duration sequences using DFT techniques is called fast convolution. The convolution of two sequences by convolution sum formula. This technique of convolving two finite duration sequences using DFT techniques is called fast convolution. The convolution of two sequences by convolution sum formula.

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

is called direct convolution or slow convolution. The term fast is used because the DFT can be evaluated rapidly and efficiently using any of a large class of algorithms called Fast Fourier Transform (FFT). In a practical sense, the size of DFTs need not be restricted to $N_1 + N_2 - 1$-point transforms.
Any number $L$ can be used for the transform size subject to the restriction $L \_ (N_1 + N_2 - 1)$. If
$L > (N_1 + N_2 - 1)$, then $y(n)$ will have zero valued samples at the end of the period.

**EXAMPLE 2.1** Find the linear convolution of the sequences $x(n)$ and $h(n)$ using DFT.

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

***Solution:*** Let $y(n)$ be the linear convolution of $x(n)$ and $h(n)$. $x(n)$ and $h(n)$ are of length 2 each. So the linear convolution of $x(n)$ and $h(n)$ will produce a 3 sample sequence ($2 + 2 - 1 = 3$). To avoid time aliasing, we convert the 2 sample input sequences into 3 sample sequences by padding with zeros.

$x(n) = \{1, 2, 0\}$ and $h(n) = \{2, 1, 0\}$

By the definition of $N$-point DFT, the 3-point DFT of $x(n)$ is:

$$X(k) = \sum_{n=0}^{2} x(n)\, e^{-j\frac{2\pi}{3}kn} = x(0)e^{0} + x(1)e^{-j\frac{2\pi}{3}k} + x(2)e^{-j\frac{4\pi}{3}k} = 1 + 2\, e^{-j\frac{2\pi}{3}k}$$

When $k = 0$, $X(0) = 1 + 2e^{0} = 3$

When $k = 1$, $X(1) = 1 + 2e^{-j\frac{2\pi}{3}} = 1 + 2(-0.5 - j0.866) = -j1.732$

When $k = 2$, $X(2) = 1 + 2e^{-j\frac{4\pi}{3}} = 1 + 2(-0.5 + j0.866) = j1.732$

By the definition of $N$-point DFT, the 3-point DFT of $h(n)$ is:

$$H(k) = \sum_{n=0}^{2} h(n)\, e^{-j\frac{2\pi}{3}nk} = h(0)e^{0} + h(1)e^{-j\frac{2\pi}{3}k} + h(2)e^{-j\frac{4\pi}{3}k} = 2 + e^{-j\frac{2\pi}{3}k}$$

When $k = 0$, $H(0) = 2 + 1 = 3$

When $k = 1$, $H(1) = 2 + e^{-j\frac{2\pi}{3}} = 2 + (-0.5 - j0.866) = 1.5 - j0.866$

When $k = 2$, $H(2) = 2 + e^{-j\frac{4\pi}{3}} = 2 + (-0.5 + j0.866) = 1.5 + j0.866$

Let $Y(k) = X(k)H(k)$ for $k = 0, 1, 2$

When $k = 0$, $Y(0) = X(0)H(0) = (3)(3) = 9$

When $k = 1$, $Y(1) = X(1)H(1) = (-j1.732)(1.5 - j0.866) = -1.5 - j2.598$

When $k = 2$, $Y(2) = X(2)H(2) = (j1.732)(1.5 + j0.866) = -1.5 + j2.598$

$\therefore$ $Y(k) = \{9,\ -1.5 - j2.598, -1.5 + j2.598\}$

The sequence $y(n)$ is obtained from IDFT of $Y(k)$. By definition of IDFT,

$$y(n) = \frac{1}{N}\sum_{k=0}^{N-1} Y(k)\, e^{j\frac{2\pi}{N}nk}; \quad \text{for} \quad n = 0, 1, 2, \ldots, N-1$$

$$y(n) = \frac{1}{3}\sum_{k=0}^{2} Y(k)e^{j\frac{2\pi}{3}nk} = \frac{1}{3}\left[Y(0)e^{0} + Y(1)e^{j\frac{2\pi}{3}n} + Y(2)e^{j\frac{4\pi}{3}n}\right] \quad \text{for} \quad n = 0, 1, 2$$

$$\text{When } n = 0,\ y(0) = \frac{1}{3}[Y(0) + Y(1) + Y(2)]$$

$$= \frac{1}{3}[9 + (-1.5 - j2.598) + (-1.5 + j2.598)]$$

$$= \frac{1}{3}[6] = 2$$

$$\text{When } n = 1,\ y(1) = \frac{1}{3}\left[Y(0) + Y(1)\, e^{j\frac{2\pi}{3}} + Y(2) e^{j\frac{4\pi}{3}}\right]$$

$$= \frac{1}{3}[9 + (-1.5 - j2.598)(-0.5 + j0.866) + (-1.5 + j2.598)(-0.5 - j0.866)]$$

$$= \frac{1}{3}[9 + 0.75 + 2.25 + 0.75 + 2.25] = 5$$

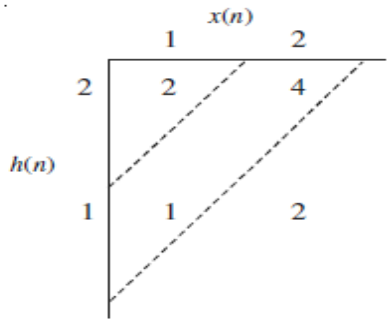
$$\text{When } n = 2,\ y(2) = \frac{1}{3}\left[Y(0) + Y(1)\, e^{j\frac{4\pi}{3}} + Y(2) e^{j\frac{8\pi}{3}}\right]$$

$$= \frac{1}{3}[9 + (-1.5 - j2.598)(-0.5 + j0.866) + (-1.5 + j2.598)(-0.5 - j0.866)]$$

$$= \frac{1}{3}[9 + 0.75 - 2.25 + 0.75 - 2.25] = 2$$

$$\therefore \quad y(n) = \{2, 5, 2\}$$

The linear convolution of $x(n) = \{1, 2\}$ and $h(n) = \{2, 1\}$ is obtained using the tabular method as shown below.




From the above table, $y(n) = \{2, 1 + 4, 2\} = \{2, 5, 2\}$.

**EXAMPLE 2.2** Find the linear convolution of the sequences $x(n)$ and $h(n)$ using DFT.
$x(n) = \{1, 0, 2\}$, $h(n) = \{1, 1\}$

***Solution:*** Let $y(n)$ be the linear convolution of $x(n)$ and $h(n)$. $x(n)$ is of length 3 and $h(n)$ is of length 2. So the linear convolution of $x(n)$ and $h(n)$ will produce a 4-sample sequence $(3 + 2 - 1 = 4)$. To avoid time aliasing, we convert the 2-sample and 3-sample sequences into 4-sample sequences by padding with zeros.

$x(n) = \{1, 0, 2, 0\}$ and $h(n) = \{1, 1, 0, 0\}$

By the definition of $N$-point DFT, the 4-point DFT of $x(n)$ is:

$$X(k) = \sum_{n=0}^{3} x(n)\, e^{-j\frac{2\pi}{4}kn} = x(0)e^{0} + x(1)e^{-j\frac{\pi}{2}k} + x(2)e^{-j\pi k} + x(3)\, e^{-j\frac{3\pi}{2}k}$$

$$= 1 + 2e^{-j\pi k} \qquad k = 0, 1, 2, 3$$

When $k = 0$, $X(0) = 1 + 2e^{0} = 1 + 2 = 3$

When $k = 1$, $X(1) = 1 + 2e^{-j\pi} = 1 + 2(-1) = -1$

When $k = 2$, $X(2) = 1 + 2e^{-j2\pi} = 1 + 2(1) = 3$

When $k = 3$, $X(3) = 1 + 2e^{-j3\pi} = 1 + 2(-1) = -1$

$\therefore$ $X(k) = \{3, -1, 3, -1\}$

By the definition of $N$-point DFT, the 4-point DFT of $h(n)$ is:

$$H(k) = \sum_{n=0}^{3} h(n)\, e^{-j\frac{2\pi}{4}nk} = h(0)e^{0} + h(1)e^{-j\frac{\pi}{2}k} + h(2)e^{-j\pi k} + h(3)\, e^{-j\frac{3\pi}{2}k}$$

$$= 1 + e^{-j\frac{\pi}{2}k} \qquad k = 0, 1, 2, 3$$

When $k = 0$, $H(0) = 1 + 1 = 2$

When $k = 1$, $H(1) = 1 + e^{-j\frac{\pi}{2}} = 1 - j$

When $k = 2$, $H(2) = 1 + e^{-j\pi} = 1 - 1 = 0$

When $k = 3$, $H(3) = 1 + e^{-j\frac{3\pi}{2}} = 1 + j$

$\therefore$ $H(k) = \{2, 1 - j, 0, 1 + j\}$

Let $Y(k) = X(k)H(k)$ for $k = 0, 1, 2$

$\therefore$ $Y(k) = X(k)\, H(k) = \{3, -1, 3, -1\}\{2, 1 - j, 0, 1 + j\} = \{6, -1 + j, 0, -1 - j\}$

The sequence $y(n)$ is obtained from IDFT of $Y(k)$.

By definition of IDFT,

$$y(n) = \frac{1}{N} \sum_{k=0}^{N-1} Y(k)\, e^{j\frac{2\pi}{N}nk}, \quad \text{for} \quad n = 0, 1, 2, 3$$

$$y(n) = \frac{1}{4} \sum_{k=0}^{3} Y(k)\, e^{j\frac{\pi}{2}nk}$$

$$= \frac{1}{4}\left[ Y(0)e^{0} + Y(1)e^{j\frac{\pi}{2}n} + Y(2)e^{j\pi n} + Y(3)e^{j\frac{3\pi}{2}n} \right], \quad \text{for} \quad n = 0, 1, 2, 3$$

$$y(n) = \frac{1}{4}[6 + (-1 + j)\, e^{j\frac{\pi}{2}n} + (-1 - j)\, e^{j\frac{3\pi}{2}n}$$

$$\text{When } n = 0,\ y(0) = \frac{1}{4}[6 + (-1 + j) + (-1 - j)] = 1$$

$$\text{When } n = 1,\ y(1) = \frac{1}{4}\left[6 + (-1 + j)e^{j\frac{\pi}{2}} + (-1 - j)e^{j\frac{3\pi}{2}}\right]$$

$$= \frac{1}{4}[6 + (-1 + j)(j) + (-1 - j)(-j)]$$

$$= \frac{1}{4}[6 - j - 1 + j - 1] = 1$$

$$\text{When } n = 2,\ y(2) = \frac{1}{4}[6 + (-1 + j)e^{j\pi} + (-1 - j)e^{j3\pi}]$$

$$= \frac{1}{4}[6 + (-1 + j)(-1) + (-1 - j)(-1)]$$

$$= \frac{1}{4}[6 + 1 - j + 1 - j] = 2$$

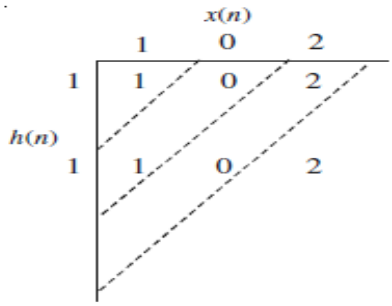
$$\text{When } n = 3,\ y(3) = \frac{1}{4}\left[6 + (-1 + j)e^{j\frac{3\pi}{2}} + (-1 - j)e^{j\frac{9\pi}{2}}\right]$$

$$= \frac{1}{4}[6 + (-1 + j)(-j) + (-1 - j)(j)]$$

$$= \frac{1}{4}[6 + j + 1 - j + 1] = 2$$

Therefore, the linear convolution of $x(n)$ and $h(n)$ is:
$y(n) = x(n) * h(n) = \{1, 1, 2, 2\}$

The linear convolution of $x(n) = \{1, 0, 2\}$ and $h(n) = \{1, 1\}$ is obtained using the tabular method as shown below.




From the above table, $y(n) = \{1, 1, 2, 2\}$.

**OVERLAP-ADD METHOD :**

In overlap-add method, the longer sequence $x(n)$ of length $L$ is split into $m$ number of smaller sequences of length $N$ equal to the size of the smaller sequence $h(n)$. (If required zero padding may be done to $L$ so that $L = mN$). The linear convolution of each section (of length $N$) of longer sequence with the smaller sequence of length $N$ is performed. This gives an output sequence of length $2N - 1$.
In t his method, the last $N - 1$ samples of each output sequence overlaps with the first $N - 1$ samples of next section. While combining the output sequences of the various sectioned convolutions, the corresponding samples of overlapped regions are added and the samples of non-overlapped regions are retained as such. If the linear convolution is to be performed by DFT (or FFT), since DFT supports only circular convolution and not linear convolution directly, we have to pad each section of the longer sequence (of length $N$) and also the smaller sequence (of length $N$) with $N - 1$ zeros before computing the circular convolution of each section with the smaller sequence. The steps for this fast convolution by overlap-add method are as follows:
***Step* 1**: $N - 1$ zeros are padded at the end of the impulse response sequence $h(n)$ which isof length $N$ and a sequence of length $2N - 1$ is obtained. Then the $2N - 1$ point FFT is performed and the output values are stored.
***Step* 2**: Split the data, i.e. $x(n)$ into $m$ blocks each of length $N$ and pad $N - 1$ zeros to each block to make them $2N - 1$ sequence blocks and find the FFT of each block.

***Step* 3**: The stored frequency response of the filter, i.e. the FFT output sequence obtained in Step 1 is multiplied by the FFT output sequence of each of the selected block in Step 2.
***Step* 4**: A $2N - 1$ point inverse FFT is performed on each product sequence obtained in Step 3.
***Step* 5**: The first $(N - 1)$ IFFT values obtained in Step 4 for each block, overlapped with the last $N - 1$ values of the previous block. Therefore, add the overlapping values and keep the non-overlapping values as they are. The result is the linear convolution of $x(n)$ and $h(n)$.

## OVERLAP-SAVE METHOD

In overlap-save method, the results of linear convolution of the various sections are obtained using circular convolution. Let $x(n)$ be a longer sequence of length $L$ and $h(n)$ be a smaller sequence of length $N$. The regular convolution of sequences of length $L$ and $N$ has $L + N - 1$ samples. If $L > N$, we have to zero pad the second sequence $h(n)$ to length $L$. So their linear convolution will have $2L - 1$ samples. Its first $N - 1$ samples are contaminated by

wraparound and the rest corresponds to the regular convolution. To understand this let $L = 12$ and $N = 5$. If we pad $N$ by 7 zeros, their regular convolution has 23 (or $2L - 1$) samples with 7 trailing zeros ($L - N = 7$). For periodic convolution, 11 samples ($L - 1 = 11$) are wrapped around. Since the last 7 (or $L - N$) are zeros only, first four samples $(2L - 1) - (L) - (L - N) = N - 1 = 5 - 1 = 4$ of the periodic convolution are contaminated by wraparound. This idea is the basis of overlap-save method. First, we add $N - 1$ leading zeros to the longer sequence $x(n)$ and section it into $k$ overlapping (by $N - 1$) segments of length $M$. Typically we choose $M = 2N$. Next, we zero pad $h(n)$ (with trailing zeros) to length $M$, and find the periodic convolution of $h(n)$ with each section of $x(n)$. Finally, we discard the first $N - 1$ (contaminated) samples from each convolution and glue (concatenate) the results to give the required convolution.
***Step* 1**: $N$ zeros are padded at the end of the impulse response $h(n)$ which is of length $N$ and a sequence of length $M = 2N$ is obtained. Then the $2N$ point FFT is performed and the output values are stored.
***Step* 2**: A $2N$ point FFT on each selected data block is performed. Here each data block begins with the last $N - 1$ values in the previous data block, except the first data block which begins with $N - 1$ zeros.
***Step* 3**: The stored frequency response of the filter, i.e. the FFT output sequence obtained in Step 1 is multiplied by the FFT output sequence of each of the selected blocks obtained in Step 2.
***Step* 4**: A $2N$ point inverse FFT is performed on each of the product sequences obtained in Step 3.
***Step* 5**: The first $N - 1$ values from the output of each block are discarded and the remaining values are stored. That gives the response $y(n)$.
In either of the above two methods, the FFT of the shorter sequence need be found only once, stored, and reused for all subsequent partial convolutions. Both methods allow online implementation if we can tolerate a small processing delay that equals the time required for each section of the long sequence to arrive at the processor

# Fast Fourier Transform

## 2.2 INTRODUCTION

The $N$-point DFT of a sequence $x(n)$ converts the time domain $N$-point sequence $x(n)$ to a frequency domain $N$-point sequence $X(k)$. The direct computation of an $N$-point DFT requires $N \times N$ complex multiplications and $N(N - 1)$ complex additions. Many methods were developed for reducing the number of calculations involved. The most popular of these is the Fast Fourier Transform (FFT), a method developed by Cooley and Turkey. The FFT may be defined as an algorithm (or a method) for computing the DFT efficiently (with reduced number of calculations). The computational efficiency is achieved by adopting a divide and conquer approach. This approach is based on the decomposition of an $N$-point DFT into

successively smaller DFTs and then combining them to give the total transform. Based on this basic approach, a family of computational algorithms were developed and they are collectively known as FFT algorithms. Basically there are two FFT algorithms; Decimation- in-time (DIT) FFT algorithm and Decimation-in-frequency (DIF) FFT algorithm. In this chapter, we discuss DIT FFT and DIF FFT algorithms and the computation of DFT by these methods.

# FAST FOURIER TRANSFORM

The DFT of a sequence $x(n)$ of length $N$ is expressed by a complex-valued sequence $X(k)$ as

$$X(K) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N}, K = 0,1,2,....N-1 \text{ where}$$

Let $W_N$ be the complex valued phase factor, which is an $N^{th}$ root of unity given by

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

Thus,

$X(k)$ becomes,

$$X(K) = \sum_{n=0}^{N-1} x(n) W_N{}^{nk}, K = 0,1,2,....N-1$$

Similarly, IDFT is written as

$$x(n) = \sum_{n=0}^{N-1} X(K) W_N{}^{-nk}, n = 0,1,2,....N-1$$

From the above equations for $X(k)$ and $x(n)$, it is clear that for each value of $k$, the direct computation of $X(k)$ involves $N$ complex multiplications ($4N$ real multiplications) and $N - 1$ complex additions ($4N - 2$ real additions). Therefore, to compute all $N$ values of DFT, $N^2$ complex multiplications and $N(N - 1)$ complex additions are required. In fact the DFT and IDFT involve the same type of computations.

If $x(n)$ is a complex-valued sequence, then the $N$-point DFT given in equation for $X(k)$ can be expressed as

$$X(k) = X_R(k) + jX_I(k)$$

The direct computation of the DFT needs $2N^2$ evaluations of trigonometric functions, $4N^2$ real multiplications and $4N(N-1)$ real additions. Also this is primarily inefficient as it cannot exploit the symmetry and periodicity properties of the phase factor $W_N$, which are

Symmetry property $W_N^{k+N/2} = -W_N^K$

Periodicity property $W_N^{k+N} = W_N^K$

FFT algorithm exploits the two symmetry properties and so is an efficient algorithm for DFT computation.

By adopting a divide and conquer approach, a computationally efficient algorithm can be developed. This approach depends on the decomposition of an $N$-point DFT into successively smaller size DFTs. An $N$-point sequence, if $N$ can be expressed as $N = r_1 r_2 r_3, ..., r_m$. where $r_1 = r_2 = r_3 = ... = r_m$, then $N = r^m$, can be decimated into $r$-point sequences. For each $r$-point sequence, $r$-point DFT can be computed. Hence the DFT is of size $r$. The number $r$ is called the radix of the FFT algorithm and the number $m$ indicates the number of stages in computation. From the results of $r$-point DFT, the $r^2$-point DFTs are computed. From the results of $r^2$-point DFTs, the $r^3$-point DFTs are computed and so on, until we get $r^m$-point DFT. If $r = 2$, it is called radix-2 FFT.

## DECIMATION IN TIME (DIT) RADIX-2 FFT

In Decimation in time (DIT) algorithm, the time domain sequence $x(n)$ is decimated and smaller point DFTs are computed and they are combined to get the result of $N$-point DFT.

In general, we can say that, in DIT algorithm the $N$-point DFT can be realized from two numbers of $N/2$-point DFTs, the $N/2$-point DFT can be realized from two numbers of $N/4$-point DFTs, and so on.

In DIT radix-2 FFT, the $N$-point time domain sequence is decimated into 2-point sequences and the 2-point DFT for each decimated sequence is computed. From the results of 2-point DFTs, the 4-point DFTs, from the results of 4-point DFTs, the 8-point DFTs and so on are computed until we get $N$-point DFT.

For performing radix-2 FFT, the value of $r$ should be such that, $N = 2^m$. Here, the decimation can be performed $m$ times, where $m = \log_2 N$. In direct computation of $N$-point DFT, the total number of complex additions are $N(N-1)$ and the total number of complex multiplications are $N^2$. In radix-2 FFT, the total number of complex additions are reduced to $N \log_2 N$ and the total number of complex multiplications are reduced to $(N/2) \log_2 N$.

Let $x(n)$ be an $N$-sample sequence, where $N$ is a power of 2. Decimate or break this sequence into two sequences $f_1(n)$ and $f_2(n)$ of length $N/2$, one composed of the even indexed values of $x(n)$ and the other of odd indexed values of $x(n)$.

Given sequence $x(n) : x(0), x(1), x(2), .....x(\frac{N}{2}-1)....x(N-1)$

Even indexed sequence $f_1(n) = x(2n) : x(0), x(2), x(4), .....x(N-2)$

Odd indexed sequence $f_2(n) = x(2n+1) : x(1), x(3), x(5), .....x(N-1)$

We know that the transform X(k) of the N-point sequence x(n) is given by

$$X(K) = \sum_{n=0}^{N-1} x(n) W_N^{nk}, K = 0,1,2,....N-1$$

Breaking the sum into two parts, one for the even and one for the odd indexed values, gives

$$X(K) = \sum_{n=0}^{N/2-1} x(n)W_N{}^{nk} + \sum_{n=N/2}^{N-1} x(n)W_N{}^{nk}, K = 0,1,2,....N-1.$$

$$X(K) = \sum_{n=even}^{N/2-1} x(n)W_N{}^{nk} + W_N{}^{nk} \sum_{n=odd}^{N-1} x(n)W_N{}^{nk}$$

When n is replaced by 2n, the even numbered samples are selected and when n is replaced by 2n + 1, the odd numbered samples are selected. Hence,

$$X(K) = \sum_{n=0}^{N/2-1} x(2n)W_N{}^{2nk} + \sum_{n=0}^{N/2-1} x(2n+1)W_N{}^{(2n+1)k}$$

Rearranging each part of X(k) into (N/2)-point transforms using

$$W_N^{2nk} = (W_N^2)^{nk} = \left[ e^{-j\frac{2\pi}{N}} \right]^{2nk} = W_{N/2}^{nk} \text{ and } W_N^{(2n+1)k} = (W_N^k)W_{N/2}^{nk}$$

We can write

$$X(K) = \sum_{n=0}^{N/2-1} f_1(n)W_{N/2}^{nk} + W_N{}^k \sum_{n=0}^{N/2-1} f_2(n)W_{N/2}^{nk}$$

By definition of DFT, the N/2-point DFT of f1(n) and f2(n) is given by

$$F_1(K) = \sum_{n=0}^{N/2-1} f_1(n)W_{N/2}^{nk} \;\&\; F_2(K) + \sum_{n=0}^{N/2-1} f_2(n)W_{N/2}^{nk}$$

$$X(k) = F_1(K) + W_N{}^k F_2(K), ...k = 0,1,2,3,....N-1$$

The implementation of this equation for X(k) is shown in the following Figure . This first step in the decomposition breaks the N-point transform into two (N/2)-point transforms and the k WN provides the N-point combining algebra. The DFT of a sequence is periodic with period given by the number of points of DFT. Hence, F1(k) and F2(k) will be periodic with period N/2.

$$F_1(k + N/2) = F_1(K), \&F_2(k + N/2) = F_2(K)$$

$$F_1(k + N/2) = F_1(K), \&F_2(k + N/2) = F_2(K)$$

In addition, the phase factor $W_N^{(k+N/2)} = -(W_N^k)$

Therefore, for k ≥ N/2, X(k) is given by

$$X(K) = F_1(k - N/2) - W_N{}^k F_2(K - N/2)$$

The implementation using the periodicity property is also shown in following Figure

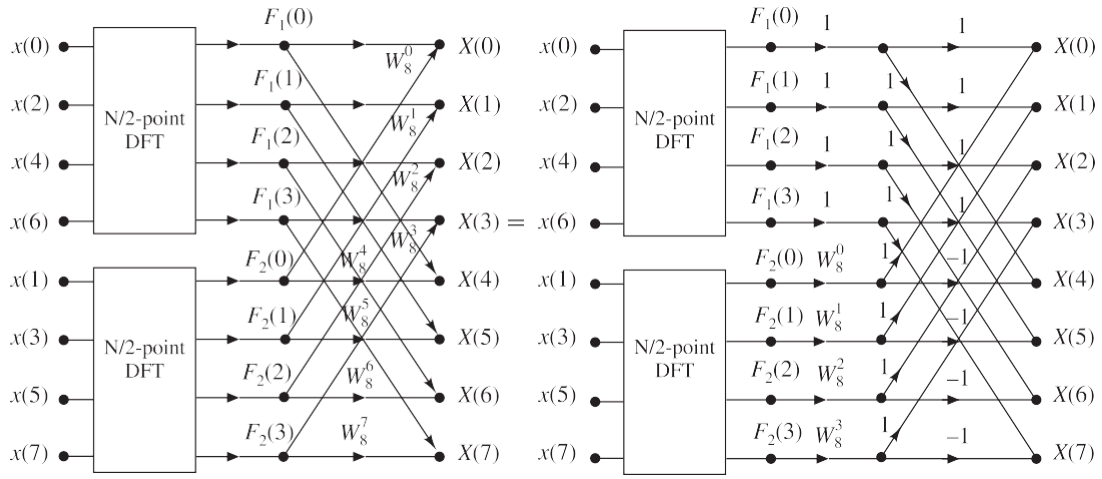



**Figure 2.1** Illustration of flow graph of the first stage DIT FFT algorithm for $N = 8$.

Having performed the decimation in time once, we can repeat the process for each of the

sequences $f_1(n)$ and $f_2(n)$. Thus $f_1(n)$ would result in two ($N$/4)-point sequences and $f_2(n)$ would result in another two ($N$/4)-point sequences.

## THE 8-POINT DFT USIKG RADIX-2 DIT FFT

The computation of 8-point DFT using radix-2 FFT involves three stages of computation. Here $N = 8 = 2^3$, therefore, $r = 2$ and $m = 3$. The given 8-point sequence is decimated into four 2-point sequences. For each 2-point sequence, the two point DFT is computed. From the results of four 2-point DFTs, two 4-point DFTs are obtained and from the results of two 4-point DFTs, the 8-point DFT is obtained.

Let the given 8-sample sequence $x(n)$ be {$x(0)$, $x(1)$, $x(2)$, $x(3)$, $x(4)$, $x(5)$, $x(6)$, $x(7)$}. The 8-samples should be decimated into sequences of two samples. Before decimation they are arranged in bit reversed order as shown in Table 2.1.

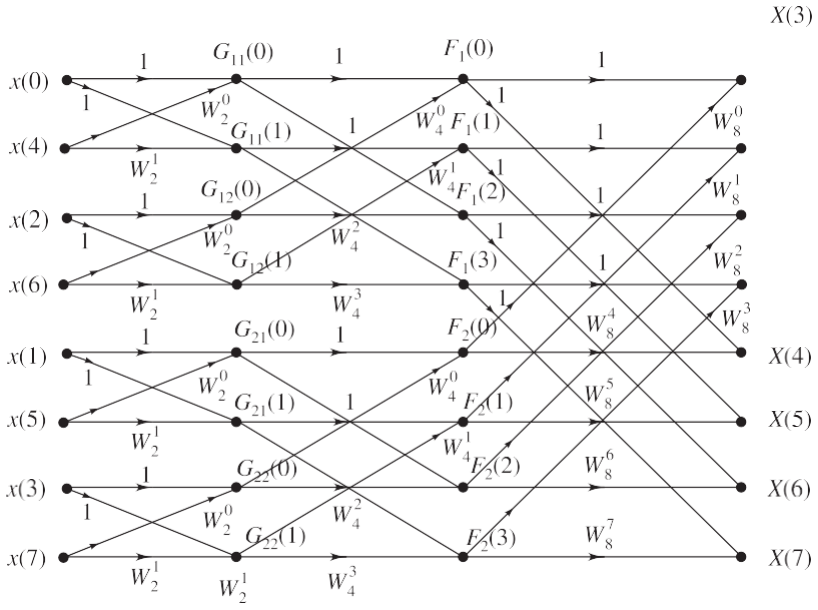



**Figure 2.4** Illustration of complete flow graph obtained by combining all the three stages for $N = 8$.

**TABLE 2.1** Normal and bit reversed order for $N = 8$.

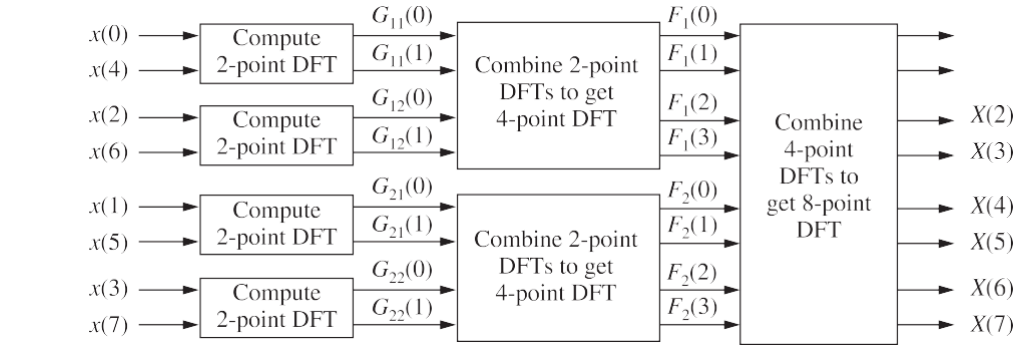
| *Normal order* | | *Bit reversed order* | |
|---|---|---|---|
| $x(0)$ | $x(000)$ | $x(0)$ | $x(000)$ |
| $x(1)$ | $x(001)$ | $x(4)$ | $x(100)$ |
| $x(2)$ | $x(010)$ | $x(2)$ | $x(010)$ |
| $x(3)$ | $x(011)$ | $x(6)$ | $x(110)$ |
| $x(4)$ | $x(100)$ | $x(1)$ | $x(001)$ |
| $x(5)$ | $x(101)$ | $x(5)$ | $x(101)$ |
| $x(6)$ | $x(110)$ | $x(3)$ | $x(011)$ |
| $x(7)$ | $x(111)$ | $x(7)$ | $x(111)$ |

The $x(n)$ in bit reversed order is decimated into 4 numbers of 2-point sequences as shown below.

(i) $x(0)$ and $x(4)$
(ii) $x(2)$ and $x(6)$
(iii) $x(1)$ and $x(5)$
(iv) $x(3)$ and $x(7)$

Using the decimated sequences as input, the 8-point DFT is computed. Figure 7.5 shows the three stages of computation of an 8-point DFT.

The computation of 8-point DFT of an 8-point sequence in detail is given below. The 8-point sequence is decimated into 4-point sequences and 2-point sequences as shown below.



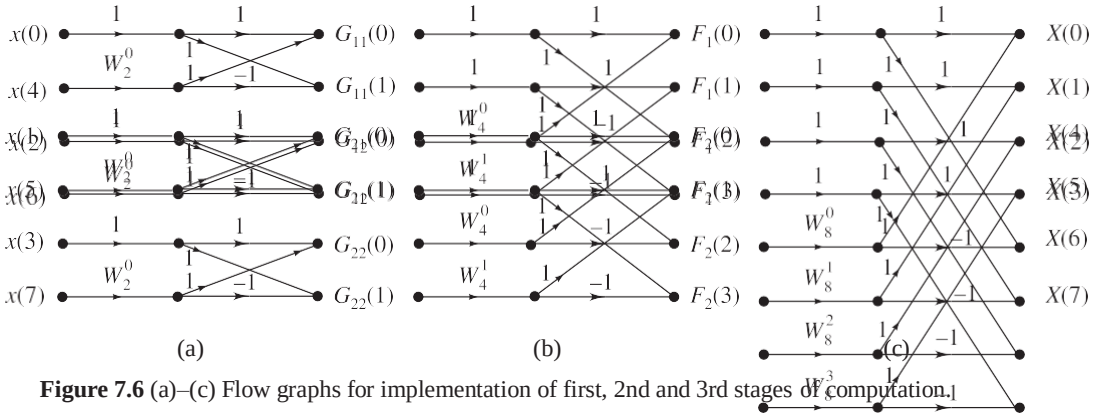




**Figure 7.6** (a)–(c) Flow graphs for implementation of first, 2nd and 3rd stages of computation.

## Butterfly Diagram

Observing the basic computations performed at each stage, we can arrive at the following conclusions:

(i) In each computation, two complex numbers $a$ and $b$ are considered.
(ii) The complex number $b$ is multiplied by a phase factor $W_N^k$.
(iii) The product $bW_N^k$ is added to the complex number $a$ to form a new complex number $A$.
(i) The product $bW^k{}_N$ is subtracted from complex number $a$ to form new complex number $B$.

The above basic computation can be expressed by a signal flow graph shown in Figure 7.7. The signal flow graph is also called butterfly diagram since it resembles a butterfly.

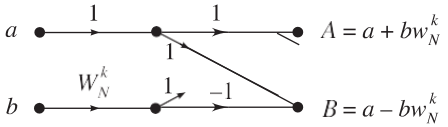



**Figure 7.7** Basic butterfFy diagram or fFow graph of radix–2 DI† FF†.

The complete flow graph for 8-point DIT FFT considering periodicity drawn in a way to remember easily is shown in Figure 7.8. In radix-2 FFT, $N/2$ butterflies per stage are required to represent the computational process. In the butterfly diagram for 8-point DFT shown in Figure 7.8, for symmetry, $W_2^0$, $W_4^0$ and $W_8^0$ are shown on the graph eventhough they are unity. The subscript 2 indicates that it is the first stage of computation. Similarly, subscripts 4 and 8 indicate the second and third stages of computation.

$X(0)$

$X(1)$

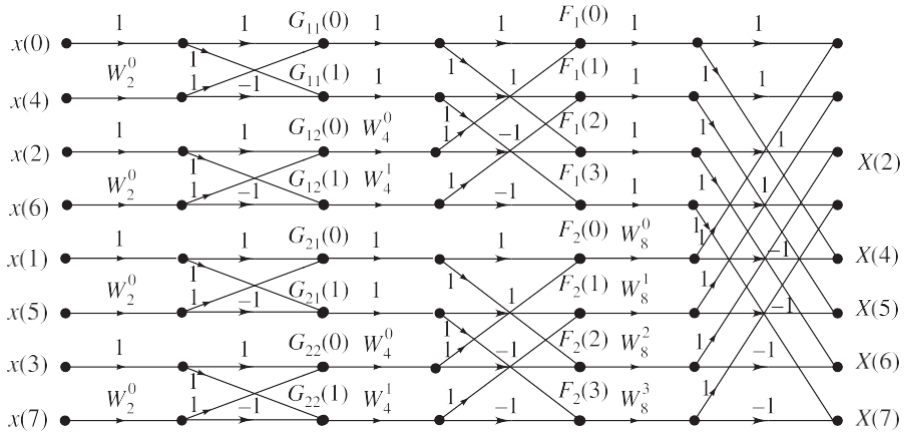



**Figure 7.8** †he signal flow graph or butterfly diagram for 8–point radix–2 DIT FFT.

# DECIMATION IN FREQUENCY (DIF) RADIX-2 FFT

In decimation in frequency algorithm, the frequency domain sequence $X(k)$ is decimated. In this algorithm, the $N$-point time domain sequence is converted to two numbers of $N/2$-point sequences. Then each $N/2$-point sequence is converted to two numbers of $N/4$-point sequences. This process is continued until we get $N/2$ numbers of 2-point sequences. Finally, the 2-point DFT of each 2-point sequence is computed. The 2-point DFTs of $N/2$ numbers of 2-point sequences will give $N$-samples, which is the $N$-point DFT of the time domain sequence. Here the equations for $N/2$-point sequences, $N/4$-point sequences, etc., are obtained by decimation of frequency domain sequences. Hence this method is called DIF.

To derive the decimation-in-frequency form of the FFT algorithm for $N$, a power of 2, we can first divide the given input sequence $x(n) = \{x(0), x(1), x(2), x(3), x(4), x(5), x(6), x(7)$ into the first half and last half of the points so that its DFT $X(k)$ is

$$X(K) = \sum_{n=0}^{N-1} x(n) W_N^{nk} = \sum_{n=0}^{N/2-1} x(n) W_N^{nk} + W_N^{nk} \sum_{n=N/2}^{N-1} x(n) W_N^{nk}$$

$$= \sum_{n=0}^{N/2-1} x(n) W_N^{nk} + W_N^{nk} \sum_{n=N/2}^{N/2-1} x(n+N/2) W_N^{(n+N/2)k}$$

It is important to observe that while the above equation for $X(k)$ contains two summations over $N/2$-points, each of these summations is not an $N/2$-point DFT, since $W_N^{nk}$ rather than $W^{nkN/2}$

$$X(K) = \sum_{n=0}^{N/2-1} x(n) W_N^{nk} + W_N^{(N/2)k} \sum_{n=0}^{N/2-1} x(n+N/2) W_N^{nk}$$

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

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

Let us split $X(k)$ into even and odd numbered samples. For even values of $k$, the $X(k)$ can be written as

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

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

For odd values of k, the X(k) can be written as

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

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

The above equations for $X(2k)$ and $X(2k + 1)$ can be recognized as $N/2$-point DFTs. $X(2k)$ is the DFT of the sum of first half and last half of the input sequence, i.e. of $\{x(n) + x(n + N/2)\}$ and $X(2k + 1)$ is the DFT of the product $W_N^n$ with the difference of first half and last half of the input, i.e. of $\{x(n) - x(n + N/2)\}W_N^n$.

If we define new time domain sequences, $u_1(n)$ and $u_2(n)$ consisting of $N/2$-samples, such that

$$u_1(n) = x(n) + x\left(n + \frac{N}{2}\right); \quad \text{for } n = 0, 1, 2, \ldots, \frac{N}{2} - 1$$

and

$$u_2(n) = \left[ x(n) - x\left(n + \frac{N}{2}\right) \right] W_N^n; \quad \text{for } n = 0, 1, 2, \ldots, \frac{N}{2} - 1$$

then the DFTs $U_1(k) = X(2k)$ and $U_2(k) = X(2k + 1)$ can be computed by first forming the sequences $u_1(n)$ and $u_2(n)$, then computing the $N/2$-point DFTs of these two sequences to obtain the even numbered output points and odd numbered output points respectively. The procedure suggested above is illustrated in Figure 7.9 for the case of an 8-point sequence.

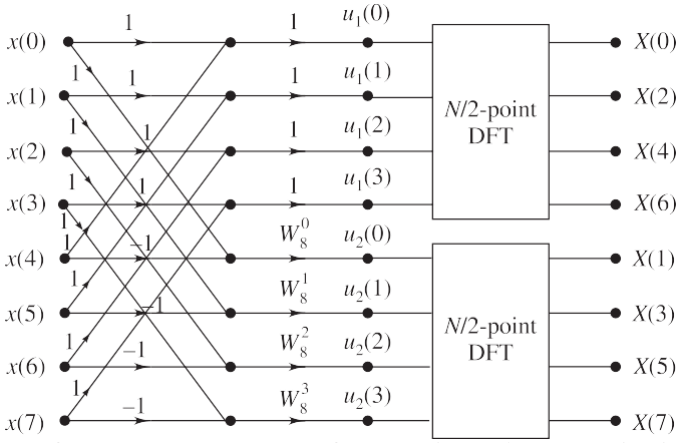



**Figure 7.9** FFow graph of the DIF decomposition of an $N$–point DF† computation into two $N/2$–point DF† computations $N = 8$.

Now each of the $N/2$-point frequency domain sequences, $U_1(k)$ and $U_2(k)$ can be decimated into two numbers of $N/4$-point sequences and four numbers of new $N/4$-point sequences can be obtained from them.

Let the new sequences be $v_{11}(n)$, $v_{12}(n)$, $v_{21}(n)$, $v_{22}(n)$. On similar lines as discussed above, we can get

$$v_{11}(n) = u_1(n) + u_1(n+2); \quad \text{for } n = 0, 1, 2, \ldots, \frac{N}{4} - 1$$

$$v_{12}(n) = [u_1(n) - u_1(n+2)]W_{N/2}^n; \quad \text{for } n = 0, 1, 2, \ldots, \frac{N}{4} - 1$$

$$v_{21}(n) = u_2(n) + u_2(n+2); \quad \text{for } n = 0, 1, 2, \ldots, \frac{N}{4} - 1$$

$$v_{22}(n) = [u_2(n) - u_2(n+2)]W_{N/2}^n; \quad \text{for } n = 0, 1, 2, \ldots, \frac{N}{4} - 1$$

This process is continued till we get only 2-point sequences. The DFT of those 2-point sequences is the DFT of $x(n)$, i.e. $X(k)$ in bit reversed order.

The third stage of computation for $N = 8$ is shown in Figure 7.11.

The entire process of decimation involves $m$ stages of decimation where $m = \log_2 N$. The computation of the $N$-point DFT via the DIF FFT algorithm requires $(N/2)\ \log_2 N$ complex multiplications and $(N - 1)\ \log_2 N$ complex additions (i.e. total number of computations remains same in both DIF and DIT).

Observing the basic calculations, each stage involves $N/2$ butterflies of the type shown in Figure 7.12.

The butterfly computation involves the following operations:

(i) In each computation two complex numbers $a$ and $b$ are considered.

(ii) The sum of the two complex numbers is computed which forms a new complex number $A$.

(iii) Subtract the complex number $b$ from $a$ to get the term $(a - b)$. The difference term $(a - b)$ is multiplied with the phase factor or twiddle factor $W_N^n$ to form a new complex number $B$.

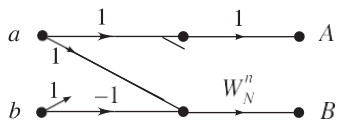



**Figure 7.12** Basic butterfFy diagram for DIF FI†.

The signal flow graph or butterfly diagram of all the three stages together is shown in Figure 7.13.

## TKE 8-POINT DFT USIKG RADIX-2 DIF FFT

The DIF computations for an 8-sample sequence are given below in detail.

Let $x(n) = \{x(0), x(1), x(2), x(3), x(4), x(5), x(6), x(7)\}$ be the given 8-sample sequence.

## First stage of COMPUTATION

In the first stage of computation, two numbers of 4-point sequences $u_1(n)$ and $u_2(n)$ are obtained from the given 8-point sequence $x(n)$ as shown below.

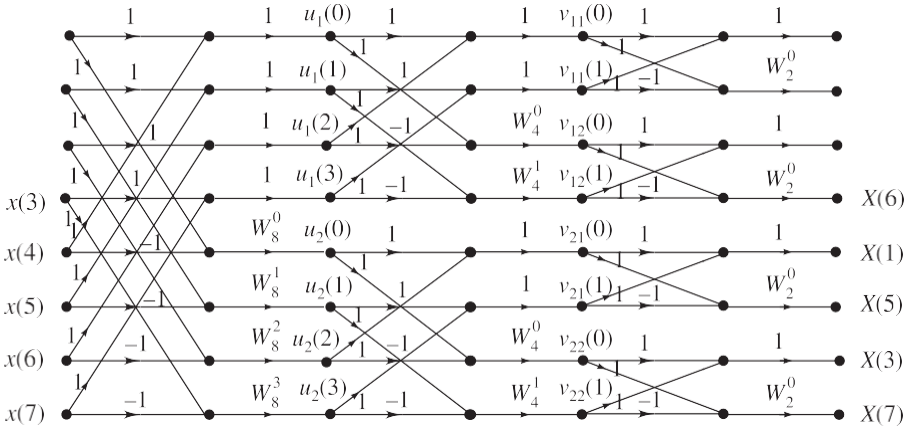



**Figure 7.13** SignaF fFow graph or butterfFy diagram for the 8–point radix–2 DIF FF† aFgorithm.

## Seconb stage of CONPUTATION

In the second stage of computation, four numbers of 2-point sequences $v_{11}(n)$, $v_{12}(n)$ and $v_{21}(n)$, $v_{22}(n)$ are obtained form the two 4-point sequences $u_1(n)$ and $u_2(n)$ obtained in stage one

## Thirb stage of CONPUTATION

In the third stage of computation, the 2-point DFTs of the 2-point sequences obtained in the second stage . The computation of 2-point DFTs is done by the butterfly operation shown in Figure 7.14(c).

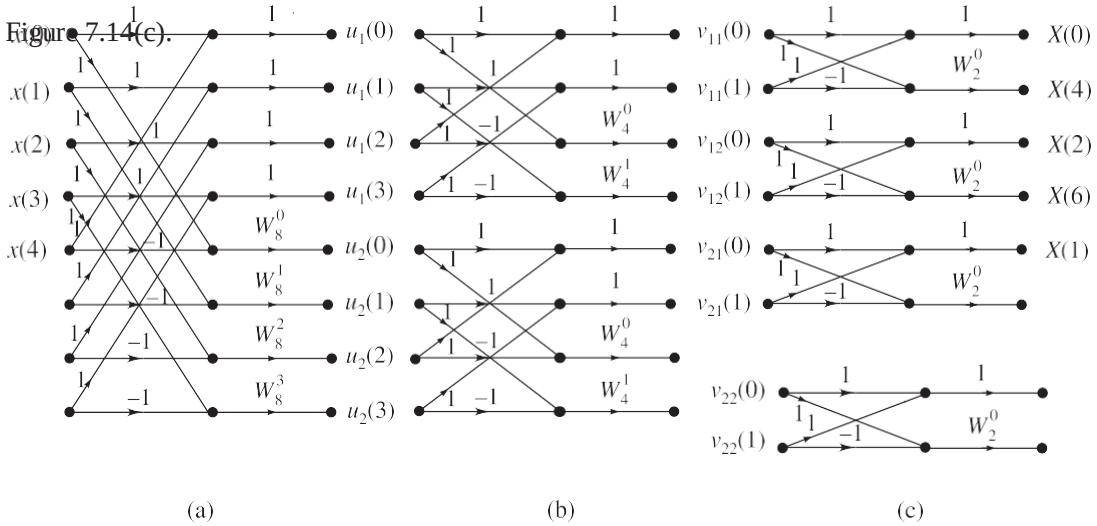



**Figure 7.14** (a)–(c) †he first, second and third stages of computation of 8–point DF† by Radix–2 DIF FF†.

## CONPARISON of DlT (DECIMATION-IN-TINE) and DlF (DECINATION-IN-FREQUENCY) ALGORITHMS

*Difference between DIT and DIF*

1. In DIT, the input is bit reversed while the output is in normal order. For DIF, the reverse is true, i.e. the input is in normal order, while the output is bit reversed. However, both DIT and DIF can go from normal to shuffled data or vice versa.
2. Considering the butterfly diagram, in DIT, the complex multiplication takes place before the add subtract operation, while in DIF, the complex multiplication takes place after the add subtract operation.

*Similarities*

1. Both algorithms require the same number of operations to compute DFT.
2. Both algorithms require bit reversal at some place during computation.

## 7.6.f Computation of IDFT through FFT

The IDFT of an $N$-point sequence $\{X(k)\}$; $k = 0, 1, ..., N-1$ is defined as

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\, e^{j\frac{2\pi}{N}nk} = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\, W_N^{-nk}$$

Taking the conjugate of the above equation for $x(n)$, we get

$$x^*(n) = \left[\frac{1}{N}\sum_{k=0}^{N-1} X(k)\, W_N^{-nk}\right]^* = \frac{1}{N}\sum_{k=0}^{N-1} X^*(k)\, W_N^{nk}$$

Taking the conjugate of the above equation for $x^*(n)$, we get

$$x(n) = \frac{1}{N}\left[\sum_{k=0}^{N-1} X^*(k)\, W_N^{nk}\right]^*$$

The term inside the square brackets in the above equation for $x(n)$ is same as the DFT computation of a sequence $X^*(k)$ and may be computed using any FFT algorithm. So we can say that the IDFT of $X(k)$ can be obtained by finding the DFT of $X^*(k)$, taking the conjugate of that DFT and dividing by $N$. Hence, to compute the IDFT of $X(k)$ the following procedure can be followed

1. Take conjugate of $X(k)$, i.e. determine $X^*(k)$.
2. Compute the $N$-point DFT of $X^*(k)$ using radix-2 FFT.
3. Take conjugate of the output sequence of FFT.
4. Divide the sequence obtained in step-3 by $N$.

The resultant sequence is $x(n)$.Thus, a single FFT algorithm serves the evaluation of both direct and inverse DFTs.

**EXAMPLE 1** Draw the butterfly line diagram for 8-point FFT calculation and briefly explain. Use decimation-in-time algorithm.

***Solution:*** The butterfly line diagram for 8-point DIT FFT algorithm is shown in following Figure

***Solution:*** For 8-point DIT FFT

1. The input sequence $x(n) = \{x(0), x(1), x(2), x(3), x(4), x(5), x(6), x(7)\}$,
2. bit reversed order, of input as i.e. as $x_r(n) = \{x(0), x(4), x(2), x(6), x(1), x(5), x(3), x(7)\}$. Since $N = 2^m = 2^3$, the 8-point DFT computation
3. Radix-2 FFT involves 3 stages of computation, each stage involving 4 butterflies. The output $X(k)$ will be in normal order.
4. In the first stage, four 2-point DFTs are computed. In the second stage they are combined into two 4-point DFTs. In the third stage, the two 4-point DFTs are combined into one 8-point DFT.
5. The 8-point FFT calculation requires $8 \log_2 8 = 24$ complex additions and $(8/2) \log_2 8 = 12$ complex multiplications.

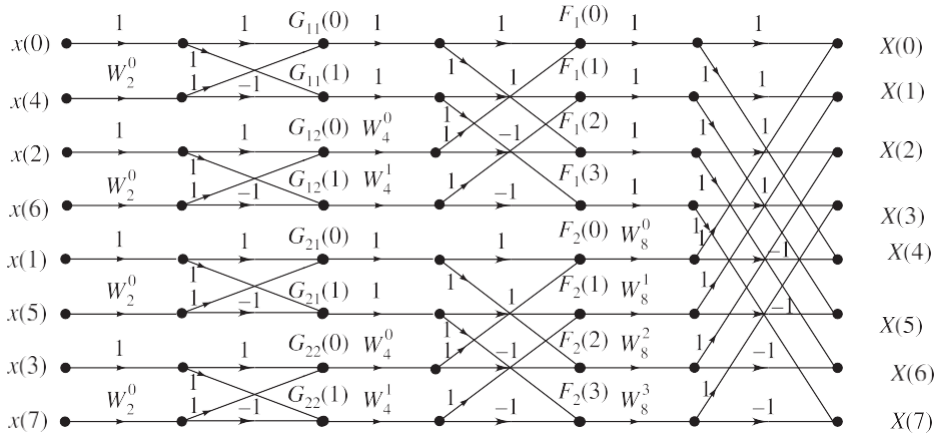



**Figure :** Butterfly Fine diagram for 8–point DIT FFT algorithm for $N = 8$.

**EXAMPLE 2** Implement the decimation-in-frequency FFT algorithm of $N$-point DFT where $N = 8$. Also explain the steps involved in this algorithm.

***Solution:*** The 8-point radix-2 DIF FFT algorithm

1. It involves 3 stages of computation. The input to the first stage is the input time sequence $x(n)$ in normal order. The output of first stage is the input to the second stage and the output of second stage is the input to the third stage. The output of third stage is the 8-point DFT in bit reversed order.

2. In DIF algorithm, the frequency domain sequence $X(k)$ is decimated.

3. In this algorithm, the $N$-point time domain sequence is converted to two numbers of $N/2$-point sequences. Then each $N/2$-point sequence is converted to two numbers of $N/4$-point sequences. Thus, we get 4 numbers of $N/4$, i.e. 2-point sequences.

4. Finally, the 2-point DFT of each 2-point sequence is computed. The 2-point DFTs of $N/2$ number of 2-point sequences will give $N$-samples which is the $N$-point DFT of the time domain sequence. The implementation of the 8-point radix-2 DIF FFT algorithm is shown in Figure 7.16.

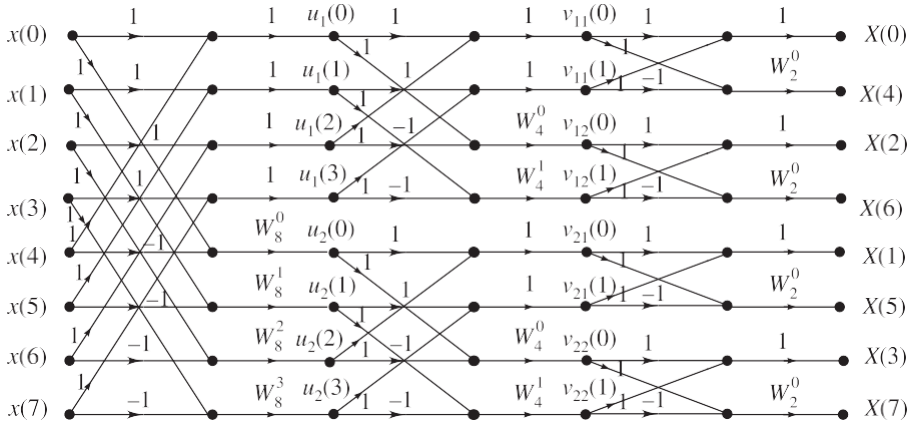



**Figure 7.16** Butterfly Fine diagram for 8–point radix–2 DIF FFT algorithm.

**EXAMPLE 7.4** What is FFT? Calculate the number of multiplications needed in the calculation of DFT using FFT algorithm with 32-point sequence.

***Solution:*** The FFT, i.e. Fast Fourier transform is a method (or algorithm) for computing the DFT with reduced number of calculations. The computational efficiency is achieved by adopting a divide and conquer approach. This approach is based on the decomposition of an *N*-point DFT into successively smaller DFTs. This basic approach leads to a family of efficient computational algorithms known as FFT algorithms. Basically there are two FFT algorithms. (i) DIT FFT algorithm and (ii) DIF FFT algorithm. If the length of the sequence $N = 2^m$, 2 indicates the radix and *m* indicates the number of stages in the computation. In radix-2 FFT, the *N*-point sequence is decimated into two *N*/2-point sequences, each *N*/2-point sequence is decimated into two *N*/4-point sequences and so on till we get two point sequences. The DFTs of two point sequences are computed and DFTs of two 2-point sequences are combined into DFT of one 4-point sequence, DFTs of two 4-point sequences are combined into DFT of one 8-point sequence and so on till we get the *N*-point DFT.

The number of multiplications needed in the computation of DFT using FFT algorithm with $N = 32$-point sequence is $= \frac{N}{2}\log_2 N = \frac{32}{2}\log_2 2^5 = 80$.

The number of complex additions $= N \log_2 N = 32 \log_2 32 = 32 \log_2 2^5 = 160$

**EXAMPLE 7.5** Explain the inverse FFT algorithm to compute inverse DFT of a 8-point DFT. Draw the flow graph for the same.

***Solution:*** The IDFT of an 8-point sequence $\{X(k), k = 0, 1, 2, ..., 7\}$ is defined as

$$x(n) = \frac{1}{8}\sum_{k=0}^{7} X(k)\, W_8^{-nk}, \quad n = 0, 1, 2, \ldots, 7$$

Taking the conjugate of the above equation for $x(n)$, we have

$$x^*(n) = \frac{1}{8}\left[\sum_{k=0}^{7} X^*(k) W_8^{nk}\right]$$

Taking the conjugate of the above equation for $x^*(n)$ we have

$$x(n) = \frac{1}{8}\left[\sum_{k=0}^{7} X^*(k) W_8^{nk}\right]^*$$

The term inside the square brackets in the RHS of the above expression for $x(n)$ is the 8-point DFT of $X^*(k)$. Hence, in order to compute the IDFT of $X(k)$ the following procedure can be followed:

1. Given $X(k)$, take conjugate of $X(k)$ i.e. determine $X^*(k)$.
2. Compute the DFT of $X^*(k)$ using radix-2 DIT or DIF FFT, [This gives $8x^*(n)$]
3. Take conjugate of output sequence of FFT. This gives $8x(n)$.
4. Divide the sequence obtained in step 3 by 8. The resultant sequence is $x(n)$.

The flow graph for computation of $N = 8$-point IDFT using DIT FFT algorithm is shown in Figure 7.18.

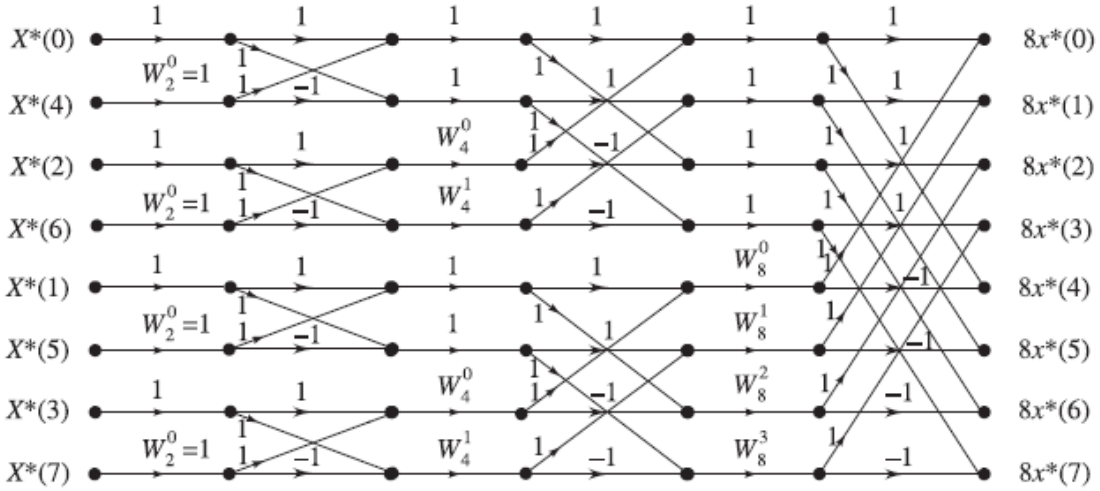



**Figure 7.18** Computation of 8–point DF† of $X^*(k)$ by radix–2, DI† FF†.

From Figure 7.18, we get the 8-point DFT of $X^*(k)$ by DIT FFT as

$$8x^*(n) = \{8x^*(0), 8x^*(1), 8x^*(2), 8x^*(3), 8x^*(4), 8x^*(5), 8x^*(6), 8x^*(7)\}$$

$$x(n) = \frac{1}{8}\{8x^*(0), 8x^*(1), 8x^*(2), 8x^*(3), 8x^*(4), 8x^*(5), 8x^*(6), 8x^*(7)\}^*$$

**EXAMPLE 7.11** Compute the DFT of the sequence $x(n) = \{1, 0, 0, 0, 0, 0, 0, 0\}$ (a) directly, (b) by FFT.

***Solution:*** (a) Direct computation of DFT
The given sequence is $x(n) = \{1, 0, 0, 0, 0, 0, 0, 0\}$. We have to compute 8-point DFT. So $N = 8$.

$$\text{DFT}\{x(n)\} = X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2}{N}nk} = \sum_{n=0}^{N-1} x(n)\, W_N^{nk} = \sum_{n=0}^{7} x(n)\, W_8^{nk}$$

$$= x(0)W_8^0 + x(1)W_8^1 + x(2)W_8^2 + x(3)W_8^3 + x(4)W_8^4 + x(5)W_8^5 + x(6)W_8^6 + x(7)W_8^7$$

$$= (1)(1) + (0)(W_8^1) + (0)W_8^2 + (0)W_8^3 + (0)W_8^4 + (0)W_8^5 + (0)W_8^6 + (0)W_8^7 = 1$$

$X(k) = 1$ for all $k$

$X(0) = 1, X(1) = 1, X(2) = 1, X(3) = 1, X(4) = 1, X(5) = 1, X(6) = 1, X(7) = 1$

$X(k) = \{1, 1, 1, 1, 1, 1, 1, 1\}$

(b) Computation by FFT. Here $N = 8 = 2^3$

The computation of 8-point DFT of $x(n) = \{1, 0, 0, 0, 0, 0, 0, 0\}$ by radix-2 DIT FFT algorithm is shown in Figure 7.31. $x(n)$ in bit reverse order is

$$x_r(n) = \{x(0), x(4), x(2), x(6), x(1), x(5), x(3), x(7)\}$$

$$= \{1, 0, 0, 0, 0, 0, 0, 0\}$$

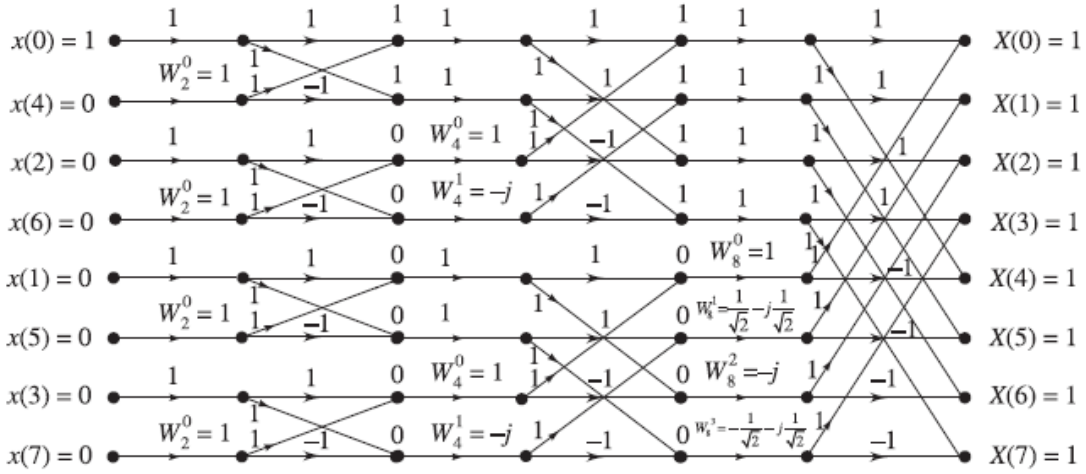
For DIT FFT input is in bit reversed order and output is in normal order.

From Figure 7.31, the 8-point DFT of the given $x(n)$ is $X(k) = \{1, 1, 1, 1, 1, 1, 1, 1\}$

**EXAMPLE 7.12** An 8-point sequence is given by $x(n) = \{2, 2, 2, 2, 1, 1, 1, 1\}$. Compute the 8-point DFT of $x(n)$ by

(a) Radix-2 DIT FFT algorithm
(b) Radix-2 DIF FFT algorithm

Also sketch the magnitude and phase spectrum.




***Solution:*** (a) 8-point DFT by Radix-2 DIT FFT algorithm

The given sequence is $x(n) = \{x(0), x(1), x(2), x(3), x(4), x(5), x(6), x(7)\}$

$$= \{2, 2, 2, 2, 1, 1, 1, 1\}$$

The given sequence in bit reversed order is

$$x_r(n) = \{x(0), x(4), x(2), x(6), x(1), x(5), x(3), x(7)\}$$

$$= \{2, 1, 2, 1, 2, 1, 2, 1\}$$

For DIT FFT, the input is in bit reversed order and the output is in normal order. The computation of 8-point DFT of $x(n)$, i.e. $X(k)$ by Radix-2 DIT FFT algorithm is shown in Figure 7.32.

From Figure 7.32, we get the 8-point DFT of $x(n)$ as

$$X(k) = \{12, 1 - j2.414, 0, 1 - j0.414, 0, 1 + j0.414, 0, 1 + j2.414\}$$

(b) 8-point DFT by radix-2 DIF FFT algorithm
For DIF FFT, the input is in normal order and the output is in bit reversed order. The computation of DFT by radix-2 DIF FFT algorithm is shown in Figure 7.33.

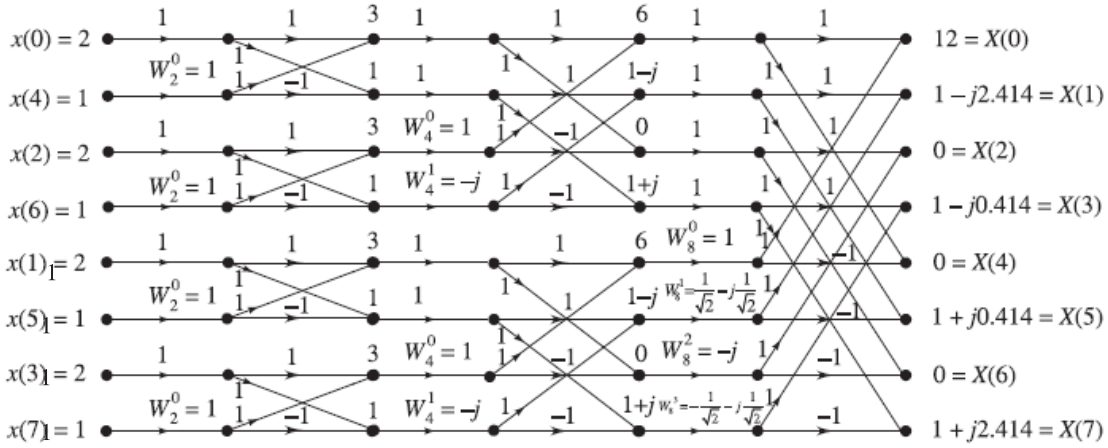



**Figure 7.33** Computation of 8–point DF† of $x(n)$ by radix–2 DIF FF† aFgorithm.

From Figure 7.33, we observe that the 8-point DFT in bit reversed order is

$$X_r(k) = \{X(0), X(4), X(2), X(6), X(1), X(5), X(3), X(7)\}$$
$$= \{12, 0, 0, 0, 1 - j2.414, 1 + j0.414, 1 - j0.414, 1 + j2.414\}$$

The 8-point DFT in normal order is

$$X(k) = \{X(0), X(1), X(2), X(3), X(4), X(5), X(6), X(7)\}$$
$$= \{12, 1 - j2.414, 0, 1 - j0.414, 0, 1 + j0.414, 0, 1 + j2.414\}$$

### *Magnitude and Phase Spectrum*

Each element of the sequence $X(k)$ is a complex number and they are expressed in rectangular coordinates. If they are converted to polar coordinates, then the magnitude and phase of each element can be obtained.

The magnitude spectrum is the plot of the magnitude of each sample of $X(k)$ as a function of $k$. The phase spectrum is the plot of phase of each sample of $X(k)$ as a function of $k$. When $N$-point DFT is performed on a sequence $x(n)$ then the DFT sequence $X(k)$ will have a periodicity of $N$. Hence, in this example, the magnitude and phase spectrum will have a periodicity of 8 as shown below.

$$X(k) = \{12, 1 - j2.414, 0, 1 - j0.414, 0, 1 + j0.414, 0, 1 + j2.414\}$$

$$= \{12\angle 0, 2.61\angle -67.5, 0\angle 0, 1.08\angle -22.5, 0\angle 0, 1.08\angle 22.5, 0\angle 0, 2.61\angle 67.5\}$$

$$= \{12\angle 0, 2.61\angle -0.37\pi, 0\angle 0, 1.08\angle -0.12\pi, 0\angle 0, 1.08\angle 0.12\pi, 0\angle 0, 2.61\angle 0.37\pi\}$$

$$|X(k)| = \{12, 2.61, 0, 1.08, 0, 1.08, 0, 2.61\}$$

$$\angle X(k) = \{0, -0.37\pi, 0, -0.12\pi, 0, 0.12\pi, 0, 0.37\pi\}$$

The magnitude and phase spectrum are shown in Figures 7.34(a) and (b).

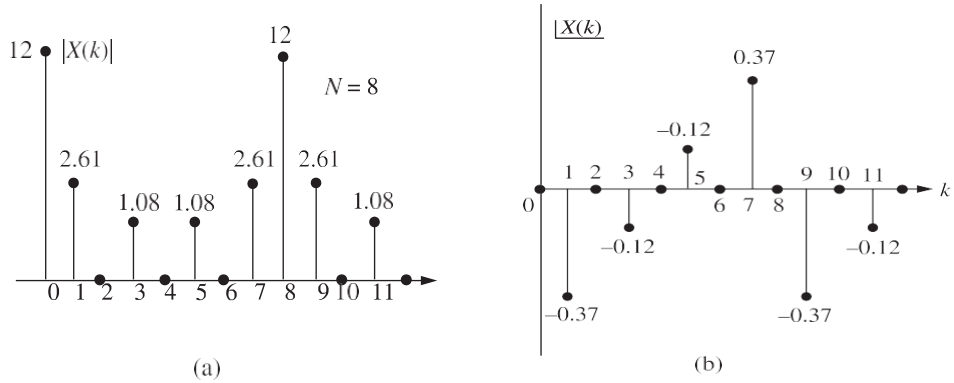



**Figure 7.34** (a) Magnitude spectrum, (b) Phase spectrum.

**EXAMPLE 7.13** Find the 8-point DFT by radix-2 DIT FFT algorithm.

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

***Solution:*** The given sequence is $x(n) = \{x(0), x(1), x(2), x(3), x(4), x(5), x(6), x(7)\}$

$$\overset{k}{=} \{2, 1, 2, 1, 2, 1, 2, 1\}$$

For DIT FFT computation, the input sequence must be in bit reversed order and the output sequence will be in normal order.

$x(n)$ in bit reverse order is

$$x_r(n) = \{x(0), x(4), x(2), x(6), x(1), x(5), x(3), x(7)\}$$

$$= \{2, 2, 2, 2, 1, 1, 1, 1\}$$

The computation of 8-point DFT of $x(n)$ by radix-2 DIT FFT algorithm is shown in Figure 7.35.

From Figure 7.35, we get the 8-point DFT of $x(n)$ as $X(k) = \{12, 0, 0, 0, 4, 0, 0, 0\}$

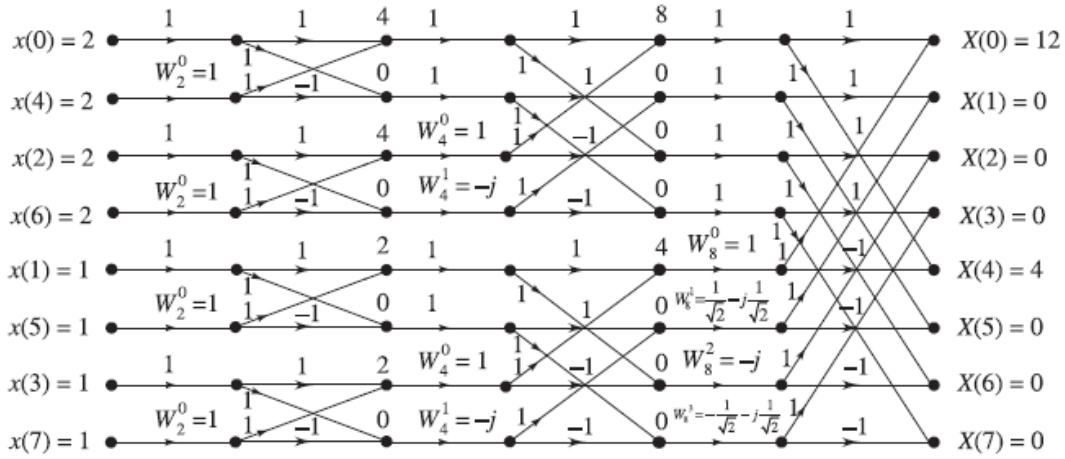


**Figure 7.35** Computation of 8–point DF† of $x(n)$ by radix–2, DI† FF†.

**EXAMPLE 7.14** Compute the DFT for the sequence $x(n) = \{1, 1, 1, 1, 1, 1, 1, 1\}$.

***Solution:*** The given sequence is $x(n) = \{x(0), x(1), x(2), x(3), x(4), x(5), x(6), x(7)\}$

$$= \{1, 1, 1, 1, 1, 1, 1, 1\}$$

The computation of 8-point DFT of $x(n)$, i.e. $X(k)$ by radix-2, DIT FFT algorithm is shown in Figure 7.36.

The given sequence in bit reversed order is

$$x_r(n) = \{x(0), x(4), x(2), x(6), x(1), x(5), x(3), x(7)\}$$

$$= \{1, 1, 1, 1, 1, 1, 1, 1\}$$

For DIT FFT, the input is in bit reversed order and output is in normal order.

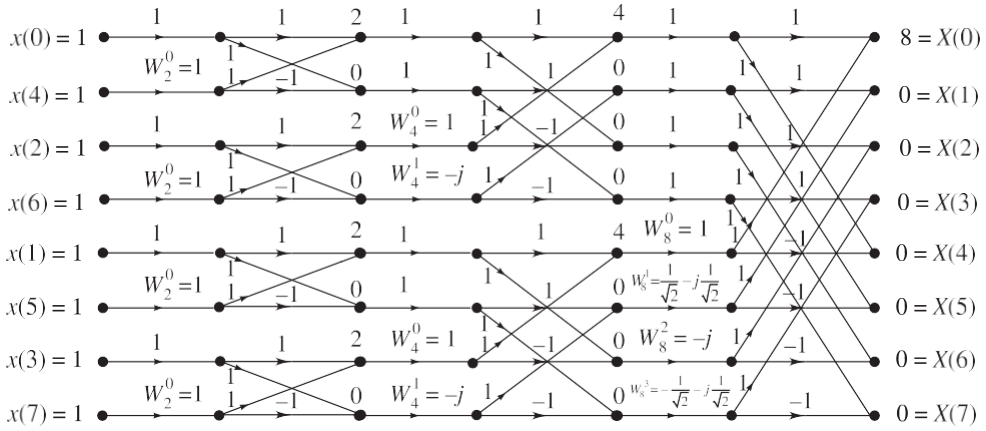



**Figure 7.36** Computation of 8–point DF† of $x(n)$ by radix–2, DI† FF†.

From Figure 7.36, we get the 8-point DFT of $x(n)$ as $X(k) = \{8, 0, 0, 0, 0, 0, 0, 0\}$.

**EXAMPLE 7.15** Given a sequence $x(n) = \{1, 2, 3, 4, 4, 3, 2, 1\}$, determine $X(k)$ using DIT FFT algorithm.

***Solution:*** The given sequence is $x(n) = \{x(0), x(1), x(2), x(3), x(4), x(5), x(6), x(7)\}$

$$= \{1, 2, 3, 4, 4, 3, 2, 1\}$$

The computation of 8-point DFT of $x(n)$, i.e. $X(k)$ by radix-2, DIT FFT algorithm is shown in Figure 7.37. For DIT FFT, the input is in bit reversed order and the output is in normal order.

The given sequence in bit reverse order is

$$x_r(n) = \{x(0), x(4), x(2), x(6), x(1), x(5), x(3), x(7)\} = \{1, 4, 3, 2, 2, 3, 4, 1\}$$

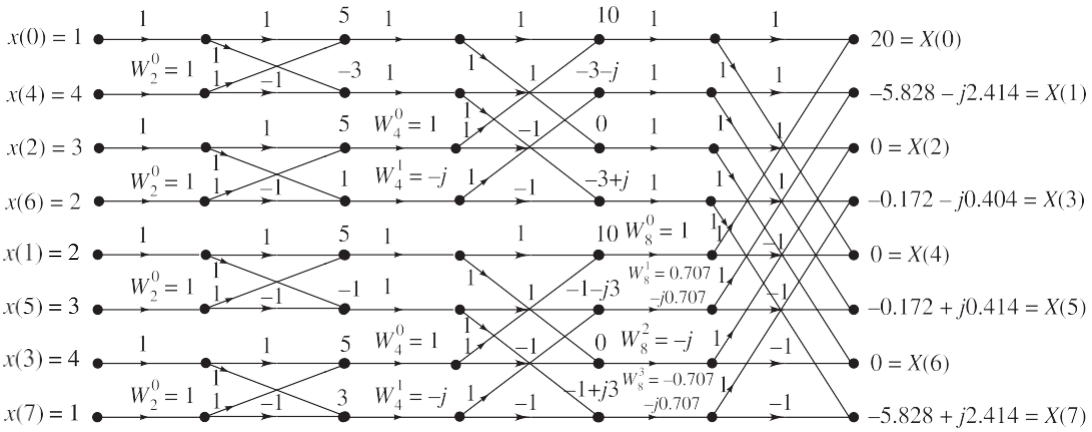



**Figure 7.37** Computation of 8–point DF† of $x(n)$ by radix–2, DI† FF†.

From Figure 7.37, we get the 8-point DFT of $x(n)$ as

$X(k) = \{20, \square\ 5.828 \square\ j2.414, \square\ 0.172 \square\ j0.414, \square\ 0.172 + j0.414, \square\ 5.828 + j2.414\}$
0, 0, 0,

**EXAMPLE 7.16** Given a sequence $x(n) = \{0, 1, 2, 3, 4, 5, 6, 7\}$, determine $X(k)$ using DIT FFT algorithm.

***Solution:*** The given sequence is $x(n) = \{x(0), x(1), x(2), x(3), x(4), x(5), x(6), x(7)\}$

$$= \{0, 1, 2, 3, 4, 5, 6, 7\}$$

The computation of 8-point DFT of $x(n)$, i.e. $X(k)$ by radix-2, DIT FFT algorithm is shown in Figure 7.38. For DIT FFT, the input is in bit reversed order and output is in normal order.

The given sequence in bit reverse order is

$$x_r(n) = \{x(0), x(4), x(2), x(6), x(1), x(5), x(3), x(7)\}$$

$$= \{0, 4, 2, 6, 1, 5, 3, 7\}$$

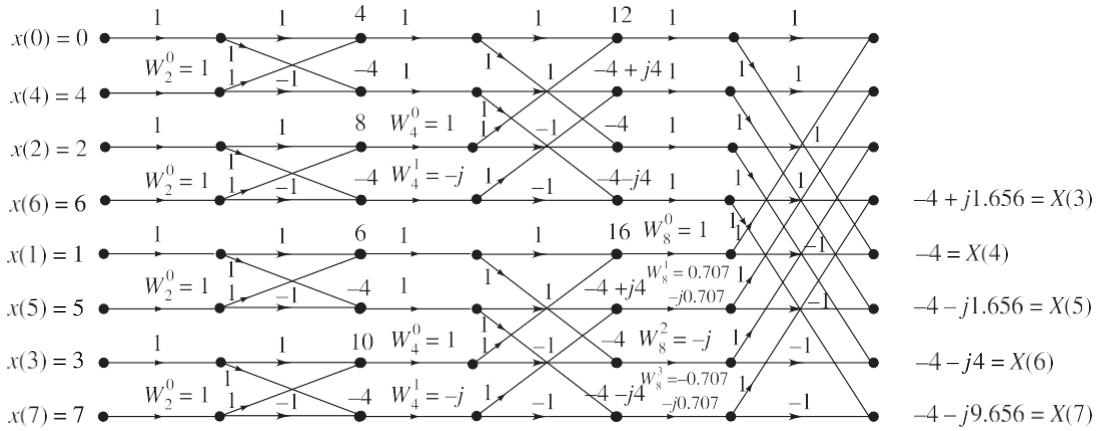



**Figure 7.38** Computation of 8–point DF† of $x(n)$ by radix–2, DI† FF†.

From Figure 7.38, we get the 8-point DFT of $x(n)$ as

$X(k) = \{28, -4 + j9.656, -4 + j4, -4 + j1.656, -4, -4 - j1.656, -4 - j4, -4 - j9.656\}$

**EXAMPLE 7.18** Find the IDFT of the sequence

$$X(k) = \{4, 1 - j2.414, 0, 1 - j0.414, 0, 1 + j0.414, 0, 1 + j2.414\}$$

using DIF algorithm.

***Solution:*** The IDFT $x(n)$ of the given 8-point sequence $X(k)$ can be obtained by finding $X^*(k)$, the conjugate of $X(k)$, finding the 8-point DFT of $X^*(k)$, using DIF algorithm to get

$8x^*(n)$, taking the conjugate of that to get $8x(n)$ and then dividing the result by 8 to get $x(n)$. For DIF algorithm, input $X^*(k)$ must be in normal order. The output will be in bit reversed order for the given $X(k)$.

$$X^*(k) = \{4, 1 + j2.414, 0, 1 + j0.414, 0, 1 - j0.414, 0, 1 - j2.414\}$$

The DFT of $X^*(k)$ using radix-2, DIF FFT algorithm is computed as shown in Figure 7.42.

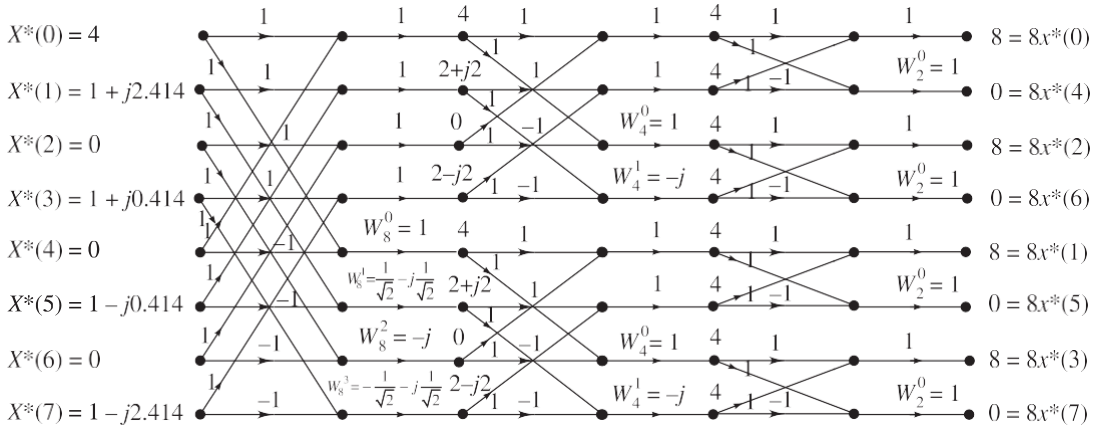



**Figure 7.42** Computation of 8–point DF† of $X^*(k)$ by radix–2 DIF FF†.

From the DIF FFT algorithm of Figure 7.42, we get

$$8x_r^*(n) = \{8, 0, 8, 0, 8, 0, 8, 0\}$$

$$8x_r(n) = \{8, 0, 8, 0, 8, 0, 8, 0\}^* = \{8, 0, 8, 0, 8, 0, 8, 0\}$$

$$x(n) = \frac{1}{8}\{8, 8, 8, 8, 0, 0, 0, 0\} = \{1, 1, 1, 1, 0, 0, 0, 0\}$$

**EXAMPLE 7.19** Compute the IDFT of the sequence

$$X(k) = \{7, -0.707 - j0.707, -j, 0.707 - j0.707, 1, 0.707 + j0.707, j, -0.707 + j0.707\}$$

using DIT algorithm.

***Solution:*** The IDFT $x(n)$ of the given sequence $X(k)$ can be obtained by finding $X^*(k)$, the conjugate of $X(k)$, finding the 8-point DFT of $X^*(k)$ using radix-2 DIT FFT algorithm to get $8x^*(n)$, taking the conjugate of that to get $8x(n)$ and then dividing by 8 to get $x(n)$. For DIT FFT, the input $X^*(k)$ must be in bit reverse order. The output $8x^*(n)$ will be in normal order. For the given $X(k)$.

$$X^*(k) = \{7, -0.707 + j0.707, j, 0.707 + j0.707, 1, 0.707 - j0.707, -j, -0.707 - j0.707\}$$

$X^*(k)$ in bit reverse order is

$$X_r^*(k) = \{7, 1, j, -j, -0.707 + j0.707, -j0.707, 0.707 + j0.707, -0.707 - j0.707\}$$

The 8-point DFT of $X^*(k)$ using radix-2, DIT FFT algorithm is computed as shown in Figure 7.43.

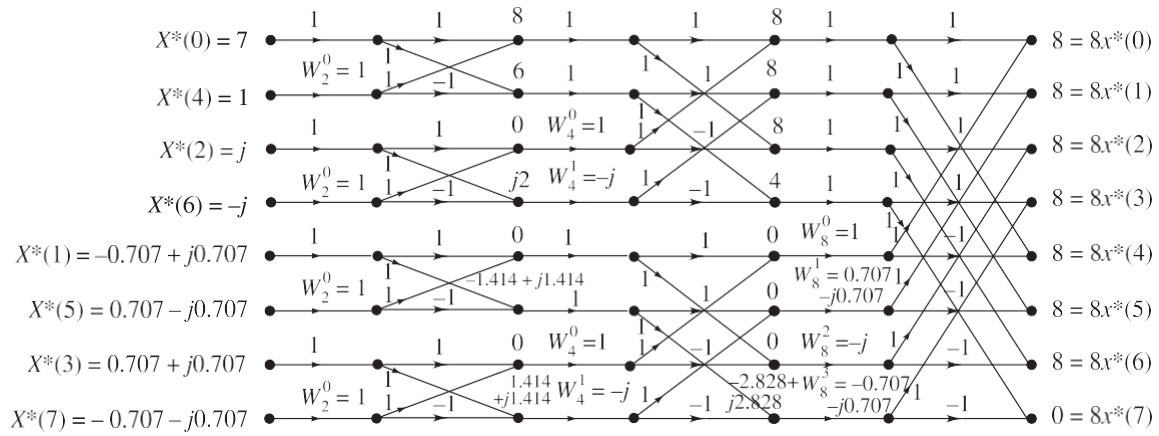



**Figure 7.43** Computation of 8–point DF† of $X^*(k)$ by radix–2, DI† FF†.

From the DIT FFT algorithm of Figure 7.43, we have

$$8x^*(n) = \{8, 8, 8, 8, 8, 8, 8, 0\}$$

$$8x(n) = \{8, 8, 8, 8, 8, 8, 8, 0\}$$

$$x(n) = \{1, 1, 1, 1, 1, 1, 1, 0\}$$

**EXAMPLE 7.20** Compute the IDFT of the square wave sequence $X(k) = \{12, 0, 0, 0, 4, 0, 0, 0\}$ using DIF algorithm.

***Solution:*** The IDFT $x(n)$ of the given sequence $X(k)$ can be obtained by finding $X^*(k)$, the conjugate of $X(k)$, finding the 8-point DFT of $X^*(k)$ using DIF algorithm to get $8x^*(n)$ taking the conjugate of that to get $8x(n)$ and then dividing the result by 8 to get $x(n)$. For DIF algorithm, the input $X^*(k)$ must be in normal order and the output $8x^*(n)$ will be in bit reversed order.

For the given $X(k)$

$$X^*(k) = \{12, 0, 0, 0, 4, 0, 0, 0\}$$

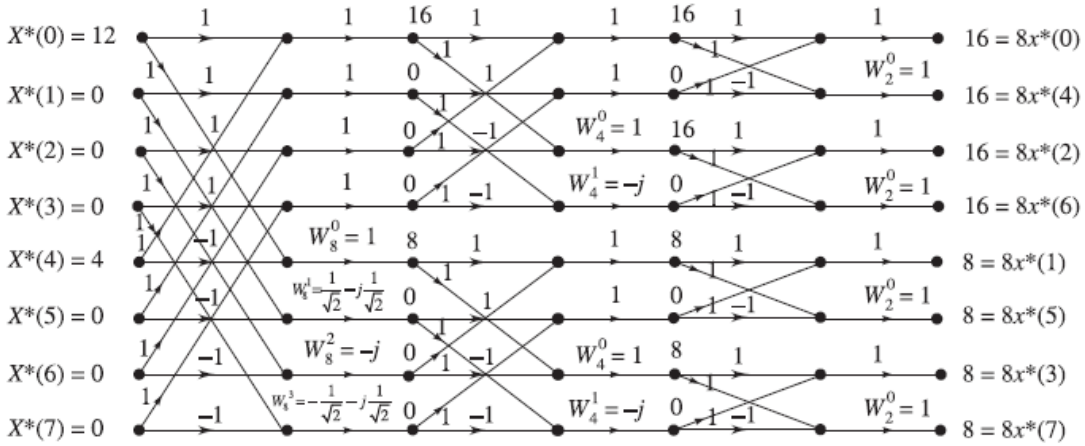
The 8-point DFT of $X^*(k)$ using radix-2, DIF FFT algorithm is computed as shown in Figure 7.44.

From Figure 7.44, we have

$$8x^*(n) = \{16, 16, 16, 16, 8, 8, 8, 8\}$$

$$8x_r(n) = \{16, 16, 16, 16, 8, 8, 8, 8\}^* = \{16, 16, 16, 16, 8, 8, 8, 8\}$$

$$x(n) = \frac{1}{8}\{16, 8, 16, 8, 16, 8, 16, 8\} = \{2, 1, 2, 1, 2, 1, 2, 1\}$$


X*(0) = 12
X*(1) = 0
X*(2) = 0
X*(3) = 0
X*(4) = 4
X*(5) = 0
X*(6) = 0
X*(7) = 0
16 = 8x*(0)
16 = 8x*(4)
16 = 8x*(2)
16 = 8x*(6)
8 = 8x*(1)
8 = 8x*(5)
8 = 8x*(3)
8 = 8x*(7)
W_2^0 = 1
W_4^0 = 1
W_4^1 = −j
W_8^0 = 1
W_8^1 = 1/√2 − j 1/√2
W_8^2 = −j
W_8^3 = −1/√2 − j 1/√2

# UNIT - III

# FIR DIGITAL FILTERS

## *INTRODUCTION*

A filter is a frequency selective system. Digital filters are classified as finite duration unit impulse response (FIR) filters or infinite duration unit impulse response (IIR) filters, depending on the form of the unit impulse response of the system. In the FIR system, the impulse response sequence is of finite duration, i.e., it has a finite number of non-zero terms. The IIR system has an infinite number of non-zero terms, i.e., its impulse response sequence is of infinite duration. IIR filters are usually implemented using recursive structures (feedback-poles and zeros) and FIR filters are usually implemented using non-recursive structures (no feedback-only zeros). The response of the FIR filter depends only on the present and past input samples, whereas for the IIR filter, the present response is a function of the present and past values of the excitation as well as past values of the response.

**Advantages of FIR filter over IIR filters:**

1. FIR filters are always stable.
2. FIR filters with exactly linear phase can easily be designed.
3. FIR filters can be realized in both recursive and non-recursive structures.
4. FIR filters are free of limit cycle oscillations, when implemented on a finite word length digital system.
5. Excellent design methods are available for various kinds of FIR filters.

**Disadvantages of FIR filters:**

1. The implementation of narrow transition band FIR filters is very costly, as it requires considerably more arithmetic operations and hardware components such as multipliers, adders and delay elements.
2. Memory requirement and execution time are very high.

FIR filters are employed in filtering problems where linear phase characteristics within the pass band of the filter are required. If this is not required, either an FIR or an IIR filter may be employed. An IIR filter has lesser number of side lobes in the stop band than an FIR filter with the same number of parameters. For this reason if some phase distortion is tolerable, an IIR filter is preferable. Also, the implementation of an IIR filter involves fewer parameters, less memory requirements and lower computational complexity.

## *Characteristics of Fir Filters with Linear Phase*

The transfer function of a FIR causal filter is given by

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

where $h(n)$ is the impulse response of the filter. The frequency response [Fourier transform of $h(n)$] is given by

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

which is periodic in frequency with period $2\pi$, i.e.,

$$H(\omega) = H(\omega + 2k\pi), \quad k = 0, 1, 2, \ldots$$

Since $H(\omega)$ is complex it can be expressed as

$$H(\omega) = \pm |H(\omega)| e^{j\theta(\omega)}$$

where $H(\omega)$ is the magnitude response and $\theta(\omega)$ is the phase response

We define the phase delay $\tau_p$ and group delay $\tau_g$ of a filter as:

$$\tau_p = -\frac{\theta(\omega)}{\omega} \text{ and } \tau_g = -\frac{d\theta(\omega)}{d\omega}$$

For FIR filters with linear phase, we can define

$$\theta(\omega) = -\alpha\omega \qquad -\pi \le \omega \le \pi$$

Where $\alpha$ is constant phase delay in samples

$$\tau_g = -\frac{d\theta(\omega)}{d\omega} = -\frac{d}{d\omega}(-\alpha\omega) = \alpha \text{ and } \tau_p = -\frac{\theta(\omega)}{\omega} = \frac{\alpha\omega}{\omega} = \alpha$$

i.e. $\tau_p = \tau_g = \alpha$ which means that $\alpha$ is independent of frequency.

We have

$$\sum_{n=0}^{N-1} h(n)\, e^{-j n} = \pm \left| H(\phi) \right| e^{\, j\, \theta(\omega)}$$

i.e.

$$\sum_{n=0}^{N-1} h(n)[\cos\omega n - j \sin\omega n] = \pm |H(\omega)| [\cos\theta(\omega) + j \sin\theta(\omega)]$$

This gives us

$$\sum_{n=0}^{N-1} h(n) \cos\omega n = \pm |H(\omega)| \cos\theta(\omega)$$

$$-\sum_{n=0}^{N-1} h(n) \sin\omega n = \pm |H(\omega)| \sin\theta(\omega)$$

Therefore,

$$\frac{\sum_{n=0}^{N-1} h(n) \sin\omega n}{\sum_{n=0}^{N-1} h(n) \cos\omega n} = \frac{\sin\theta(\omega)}{\cos\theta(\omega)} = \frac{\sin\alpha\omega}{\cos\alpha\omega}$$

i.e.

$$\sum_{n=0}^{N-1} h(n) [\sin\omega n \cos\alpha\omega - \cos\omega n \sin\alpha\omega] = 0$$

i.e.

$$\sum_{n=0}^{N-1} h(n) \sin(\alpha - n)\omega = 0$$

This will be zero when

$$h(n) = h(N-1-n) \text{ and } \alpha = \frac{N-1}{2}, \quad \text{for } 0 \le n \le N-1$$

This shows that FIR filters will have constant phase and group delays when the impulse response is symmetrical about $\alpha = (N-1)/2$.

The impulse response satisfying the symmetry condition $h(n) = h(N-1-n)$ for odd and even values of $N$ is shown in Figure 1. When $N = 9$, the centre of symmetry of the sequence occurs at the fourth sample and when $N = 8$, the filter delay is $3\frac{1}{2}$ samples.

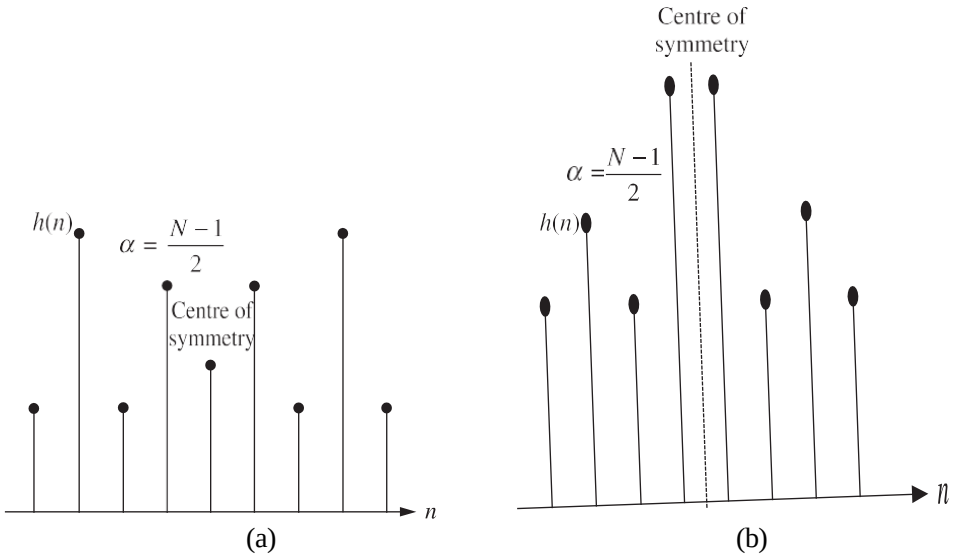



**Figure 1** Impulse response sequence of symmetrical sequences for (a) $N$ odd (b) $N$ even.

If only constant group delay is required and not the phase delay, we can write

$$\theta(\omega) = \beta - \alpha\omega$$

Now, we have

$$H(\omega) = \pm|H(\omega)|e^{j(\beta - \alpha\omega)}$$

$$\sum_{n=0}^{N-1} h(n)e^{-j\omega n} = \pm|H(\omega)|e^{j(\beta - \alpha\omega)}$$

$$\sum_{n=0}^{N-1} h(n)[\cos\omega n - j\sin\omega n] = \pm|H(\omega)|[\cos(\beta - \alpha\omega) + j\sin(\beta - \alpha\omega)]$$

This gives

$$\sum_{n=0}^{N-1} h(n)\cos\omega n = \pm\left|H(\omega)\right|\cos(\beta-\alpha\omega)$$

$$-\sum_{n=0}^{N-1} h(n)\sin\omega n = \pm\left|H(\omega)\right|\sin(\beta-\alpha\omega)$$

$$-\frac{\sum_{n=0}^{N-1} h(n)\sin\omega n}{\sum_{n=0}^{N-1} h(n)\cos\omega n} = \frac{\sin(\beta-\alpha\omega)}{\cos(\beta-\alpha\omega)}$$

Cross multiplying and rearranging, we get

$$\sum_{n=0}^{N-1} h(n)\left[\cos\omega n\sin(\beta-\alpha\omega)+\sin\omega n\cos(\beta-\alpha\omega)\right]=0$$

$$\sum_{n=0}^{N-1} h(n)\sin\left[\beta-(\alpha-n)\omega\right]=0$$

If $\beta = \pi/2$, the above equation can be written as:

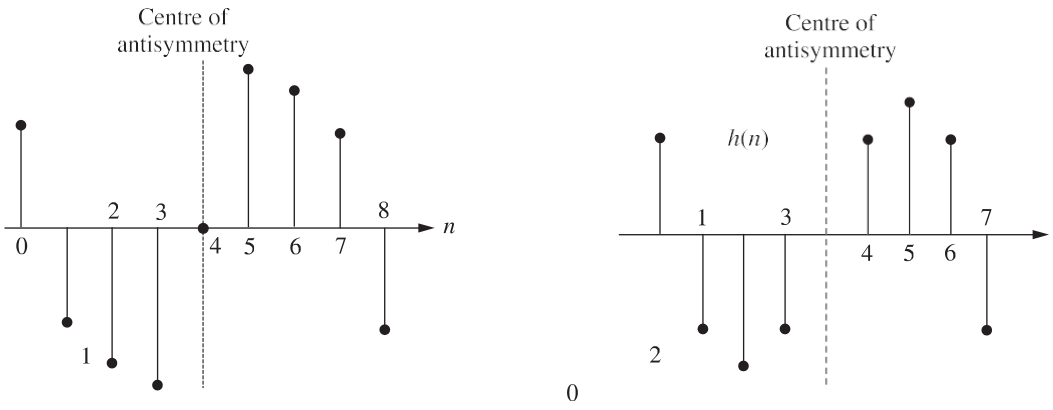
$$\sum_{n=0}^{N-1} h(n)\cos(\alpha-n)\,\omega = 0$$

This equation will be satisfied when

$$h(n) = -h(N-1-n) \text{ and } \alpha = \frac{N-1}{2}$$

This shows that FIR filters have constant group delay $\tau_g$ and not constant phase delay when the impulse response is antisymmetrical about $\alpha = (N-1)/2$.

The impulse response satisfying the antisymmetry condition is shown in Figure 2. When $N = 9$, the centre of antisymmetry occurs at fourth sample and when $N = 8$, the centre of antisymmetry occurs at $3\frac{1}{2}$ samples. From Figure 2, we find that $h[(N-1)/2] = 0$ for antisymmetric odd sequence.




( a ) ( b )

**Figure 2** Impulse response sequence of antisymmetric sequences for (a) *N* odd (b) *N* even.

**EXAMPLE 1** The length of an FIR filter is 7. If this filter has a linear phase, show that

$$\sum_{n=0}^{N-1} h(n)\sin(\alpha - n)\omega = 0$$

is satisfied

***Solution:*** The length of the filter is 7. Therefore, for linear phase,

$$\alpha = \frac{N-1}{2} = \frac{7-1}{2} = 3$$

The condition for symmetry when $N$ is odd, is $h(n) = h(N-1-n)$.
Therefore, the filter coefficients are $h(0) = h(6)$, $h(1) = h(5)$, $h(2) = h(4)$ and $h(3)$.

$$\sum_{n=0}^{N-1} h(n)\sin(\alpha - n)\omega = \sum_{n=0}^{6} h(n)\sin(3-n)\omega$$

$$= h(0)\sin 3\omega + h(1)\sin 2\omega + h(2)\sin\omega + h(3)\sin 0 + h(4)\sin(-\omega) + h(5)\sin(-2\omega) + h(6)\sin(-3\omega)$$

$$= 0$$

Hence, the equation $\sum_{n=0}^{N-1} h(n)\sin(\alpha - n)\omega = 0$ is satisfied.

**EXAMPLE 2**

The following transfer function characterizes an FIR filter ($N = 9$).
Determine the magnitude response and show that the phase and group delays are constant.

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

***Solution:*** The transfer function of the filter is given by

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

$$= h(0) + h(1)z^{-1} + h(2)z^{-2} + h(3)z^{-3} + h(4)z^{-4} + h(5)z^{-5} + h(6)z^{-6} + h(7)z^{-7} + h(8)z^{-8}$$

The phase delay $\alpha = \frac{N-1}{2} = \frac{9-1}{2} = 4$.

$$H(z) = z^{-4}\,[h(0)\,z^{4} + h(1)\,z^{3} + h(2)\,z^{2} + h(3)\,z^{1} + h(4)\,z^{0} + h(5)\,z^{-1} + h(6)\,z^{-2} + h(7)\,z^{-3} + h(8)\,z^{-4}]$$

Since $h(n) = h(N-1-n)$

$$H(z) = z^{-4}[h(0)\,(z^{4} + z^{-4}) + h(1)\,(z^{3} + z^{-3}) + h(2)\,(z^{2} + z^{-2}) + h(3)\,(z + z^{-1}) + h(4)]$$

The frequency response is obtained by replacing $z$ with $e^{j}$.

$$H(\omega) = e^{-j4\omega}[h(0)[e^{j4\omega} + e^{-j4\omega}] + h(1)[e^{j3\omega} + e^{-j3\omega}] + h(2)[e^{j2\omega} + e^{-j2\omega}] + h(3)[e^{j\omega} + e^{-j\omega}] + h(4)]$$

$$= e^{-j4\omega}\left[h(4) + 2\sum_{n=0}^{3} h(n)\cos(4-n)\omega\right]$$

$$= e^{-j4\omega}|H(\omega)|$$

where $|H(\omega)|$ is the magnitude response and $\theta(\omega) = -5\omega$ is the phase response. The phase delay $\tau_p$ and group delay $\tau_g$ are given by

$$\tau_p = -\frac{\theta(\omega)}{\omega} = 5 \text{ and } \tau_g = \frac{d(\theta(\omega))}{d\omega} = -\frac{d(-5\omega)}{d\omega} = 5$$

Thus, the phase delay and the group delay are the same and are constants.

## *Design Techniques for FIR FIilters*

The well known methods of designing FIR filters are as follows:

1. Fourier series method
2. Window method
3. Frequency sampling method
4. Optimum filter design

In Fourier series method, the desired frequency response $H_d(\omega)$ is converted to a Fourier series representation by replacing by $2\pi fT$, where $T$ is the sampling time. Then using this expression, the Fourier coefficients are evaluated by taking inverse Fourier transform of $H_d(\omega)$, which is the desired impulse response of the filter $h_d(n)$. The Z-transform of $h_d(n)$ gives $H_d(z)$ which is the transfer function of the desired filter. The $H_d(z)$ obtained from $H_d(n)$ will be a transfer function of unrealizable non causal digital filter of infinite duration. A finite duration impulse response $h(n)$ can be obtained by truncating the infinite duration impulse response $h_d(n)$ to $N$-samples. Now, take $Z$-transform of $h(n)$ to get $H(z)$. This $H(z)$ corresponds to a non-causal filter. So multiply this $H(z)$ by $z^{-(N-1)/2}$ to get the transfer function of realizable causal filter of finite duration.

In window method, we begin with the desired frequency response specification $H_d(\omega)$ and determine the corresponding unit sample response $h_d(n)$. The $h_d(n)$ is given by the inverse Fourier transform of $H_d(\omega)$. The unit sample response $h_d(n)$ will be an infinite sequence and must be truncated at some point, say, at $n = N - 1$ to yield an FIR filter of length $N$. The truncation is achieved by multiplying $h_d(n)$ by a window sequence $w(n)$. The resultant sequence will be of length $N$ and can be denoted by $h(n)$. The Z-transform of $h(n)$ will give the filter transfer function $H(z)$. There have been many windows proposed like Rectangular window, Triangular window, Hanning window, Hamming window, Blackman wndow and Kaiser window that approximate the desired characteristics.

In frequency sampling method of filter design, we begin with the desired frequency response specification $H_d(\omega)$, and it is sampled at $N$-points to generate a sequence $\tilde{H}(k)$ which corresponds to the DFT coefficients. The $N$-point IDFT of the sequence $\tilde{H}(k)$ gives the impulse response of the filter $h(n)$. The Z-transform of $h(n)$ gives the transfer function $H(z)$ of the filter.

In optimum filter design method, the weighted approximation error between the desired frequency response and the actual frequency response is spread evenly across the pass band and evenly across the stop band of the filter. This results in the reduction of maximum error. The resulting filter have ripples in both the pass band and the stop band. This concept of

design is called optimum equiripple design criterion.

The various steps in designing FIR filters are as follows:

1. Choose an ideal(desired) frequency response, $H_d(\omega)$.
2. Take inverse Fourier transform of $H_d(\omega)$ to get $h_d(n)$ or sample $H_d(\omega)$ at finite number of points ($N$-points) to get $\tilde{H}(k)$.
3. If $h_d(n)$ is determined, then convert the infinite duration $h_d(n)$ to a finite duration $h(n)$ (usually $h(n)$ is an $N$-point sequence) or if $\tilde{H}(k)$ is determined, then take $N$-point inverse DFT to get $h(n)$.
4. Take $Z$-transform of $h(n)$ to get $H(z)$, where $H(z)$ is the transfer function of the digital filter.
5. Choose a suitable structure and realize the filter.

## Design OF FIR Filters using Windows

The procedure for designing FIR filter using windows is:

1. Choose the desired frequency response of the filter $H_d(\omega)$.
2. Take inverse Fourier transform of $H_d(\omega)$ to obtain the desired impulse response $h_d(n)$.
3. Choose a window sequence $w(n)$ and multiply $h_d(n)$ by $w(n)$ to convert the infinite duration impulse response to a finite duration impulse response $h(n)$.
4. The transfer function $H(z)$ of the filter is obtained by taking Z-transform of $h(n)$.

### *Rectangular Window*

The weighting function (window function) for an $N$-point rectangular window is given by

$$w_R(n) = \begin{cases} 1, & -\dfrac{(N-1)}{2} \le n \le \left(\dfrac{N-1}{2}\right) \\ 0, & \text{elsewhere} \end{cases} \quad \text{or} \quad w_R(n) = \begin{cases} 1, & 0 \le n \le (N-1) \\ 0, & \text{elsewhere} \end{cases}$$

The spectrum (frequency response) of rectangular window $W_R(\omega)$ is given by the Fourier transform of $w_R(n)$.

$$W_R(\omega) = \sum_{n=-(N-1)/2}^{(N-1)/2} e^{-j\omega n} = \sum_{n=0}^{N-1} e^{-j\omega\left(n-\frac{N-1}{2}\right)}$$

$$= \sum_{n=0}^{N-1} e^{-j\omega n} e^{j\omega\frac{N-1}{2}} = e^{j\omega\frac{N-1}{2}} \sum_{n=0}^{N-1} e^{-j\omega n}$$

$$= e^{j\omega\left(\frac{N-1}{2}\right)}\left[\frac{1 - e^{-j\omega N}}{1 - e^{-j\omega}}\right]$$

$$= e^{j\frac{\omega N}{2}} e^{-j\frac{\omega}{2}} \frac{e^{-j\frac{\omega N}{2}}\left[e^{j\frac{\omega N}{2}} - e^{-j\frac{\omega N}{2}}\right]}{e^{-j\frac{\omega}{2}}\left[e^{j\frac{\omega}{2}} - e^{-j\frac{\omega}{2}}\right]}$$

$$= \frac{e^{j\frac{\omega N}{2}} - e^{-j\frac{\omega N}{2}}}{e^{j\frac{\omega}{2}} - e^{-j\frac{\omega}{2}}} = \frac{\sin\frac{\omega N}{2}}{\sin\frac{\omega}{2}}$$

The frequency spectrum for $N$ = 31 is shown in Figure 3. The spectrum $W_R(\omega)$ has two features that are important. They are the width of the main lobe and the side lobe amplitude. The frequency response is real and its zero occurs when $\omega = 2k/N$ where $k$ is an integer. The response for between $-2\pi/N$ and $2\pi/N$ is called the main lobe and the other lobes are called side lobes. For rectangular window the width of main lobe is $4\pi/N$. The first side lobe will be 13 dB down the peak of the main lobe and the roll off will be at 20 dB/decade. As the window is made longer, the main lobe becomes narrower and higher, and the side lobes become more concentrated around $\omega = 0$, but the amplitude of side lobes is unaffected. So increase in length does not reduce the amplitude of ripples, but increases the frequency when rectangular window is used.

If we design a low-pass filter using rectangular window, we find that the frequency response differs from the desired frequency response in many ways. It does not follow quick transitions in the desired response. The desired response of a low-pass filter changes abruptly from pass band to stop band, but the actual frequency response changes slowly. This region of gradual change is called filter's transition region, which is due to the convolution of the desired response with the window response's main lobe. The width of the transition region depends on the width of the main lobe. As the filter length $N$ increases, the main lobe becomes narrower decreasing the width of the transition region.

The convolution of the desired response and the window response's side lobes gives rise to the ripples in both the pass band and stop band. The amplitude of the ripples is dictated by the amplitude of the side lobes. This effect, where maximum ripple occurs just before and just after the transition band, is known as Gibb's phenomenon.

The Gibbs phenomenon can be reduced by using a less abrupt truncation of filter coefficients. This can be achieved by using a window function that tapers smoothly towards zero at both ends.

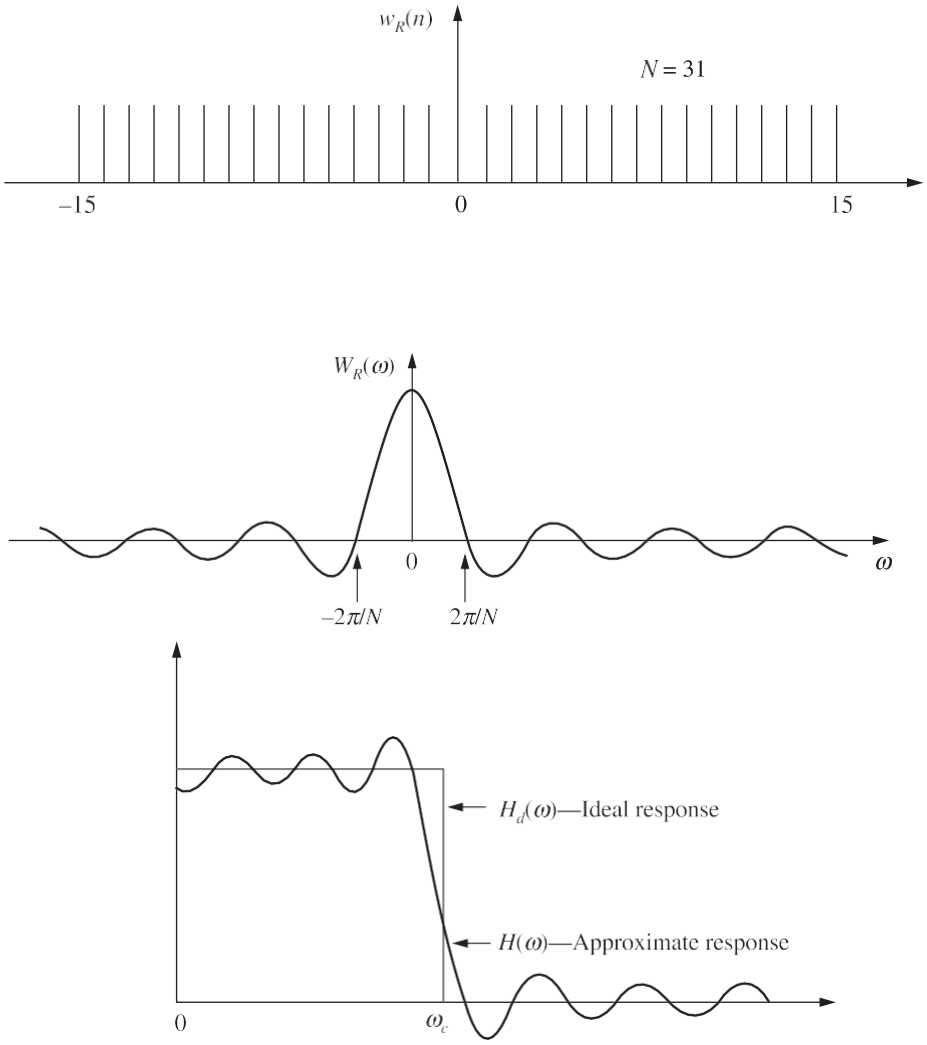



**Figure 3** (a) Rectangular window sequence, (b) Magnitude response of rectangular window, (c) Magnitude response of Now-pass filter approximated using rectangular window.

### *Triangular or Bartlett Kindow*

The triangular window has been chosen such that it has tapered sequences from the middle on either side. The window function $w_T(n)$ is defined as

$$w_T(n) = \begin{cases} 1 - \dfrac{2|n|}{N-1}, & \text{for } -\left(\dfrac{N-1}{2}\right) \le n \le \left(\dfrac{N-1}{2}\right) \\ 0 \quad , & \text{otherwise} \end{cases}$$

$$w_T(n) = \begin{cases} 1 - \dfrac{2|n-(N-1)/2|}{N-1}, & 0 \le n \le N-1 \\ 0 \quad , & \text{otherwise} \end{cases}$$

In magnitude response of triangular window, the side lobe level is smaller than that of the rectangular window being reduced from –13 dB to –25 dB. However, the main lobe width is now 8 /*N* or twice that of the rectangular window.

The triangular window produces a smooth magnitude response in both pass band and stop band, but it has the following disadvantages when compared to magnitude response

obtained by using rectangular window:

1. The transition region is more.
2. The attenuation in stop band is less.

Because of these characteristics, the triangular window is not usually a good choice

### *Raised Cosine Window*

The raised cosine window multiplies the central Fourier coefficients by approximately unity and smoothly truncates the Fourier coefficients toward the ends of the filter. The smoother ends and broader middle section produces less distortion of $h_d(n)$ around $n = 0$. It is also called generalized Hamming window.

The window sequence is of the form:

### *Hanning Window*

The Hanning window function is given by

$$w_H(n) = \begin{cases} \alpha + (1-\alpha)\cos\left(\dfrac{2\pi n}{N-1}\right), & \text{for } -\left(\dfrac{N-1}{2}\right) \le n \le \left(\dfrac{N-1}{2}\right) \\ 0, & \text{elsewhere} \end{cases}$$

The width of main lobe is 8 /*N*, i.e., twice that of rectangular window which results in doubling of the transition region of the filter. The peak of the first side lobe is –32 dB relative to the maximum value. This results in smaller ripples in both pass band and stop band of the low-pass filter designed using Hanning window. The minimum stop band attenuation of the filter is 44 dB. At higher frequencies the stop band attenuation is even greater. When compared to triangular window, the main lobe width is same, but the magnitude of the side lobe is reduced, hence the Hanning window is preferable to triangular window.

### *Hamming Window*

The Hamming window function is given by

$$w_H(n) = \begin{cases} 0.54 + 0.46\cos\left(\dfrac{2\pi n}{N-1}\right), & \text{for } -\left(\dfrac{N-1}{2}\right) \le n \le \left(\dfrac{N-1}{2}\right) \\ 0 \quad , & \text{otherwise} \end{cases}$$

$$w_H(n) = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2n\pi}{N-1}\right), & 0 \le n \le N-1 \\ 0 \quad , & \text{otherwise} \end{cases}$$

In the magnitude response for $N = 31$, the magnitude of the first side lobe is down about 41dB from the main lobe peak, an improvement of 10 dB relative to the Hanning window. But this improvement is achieved at the expense of the side lobe magnitudes at higher frequencies, which are almost constant with frequency. The width of the main lobe is 8 /*N*. In the magnitude response of low-pass filter designed using Hamming window, the first side lobe peak is –51 dB, which is –7 dB lesser with respect to the Hanning window filter. However, at higher frequencies, the stop band attenuation is low when compared to that of Hanning window. Because the Hamming window generates lesser oscillations in the side lobes than the Hanning window for the same main lobe width, the Hamming window is generally preferred.

### *Blackman Window*

The Blackman window function is another type of cosine window and given by the equation

$$w_B(n) = \begin{cases} 0.42 + 0.5\cos\dfrac{2\pi n}{N-1} + 0.08\cos\dfrac{4\pi n}{N-1}, & \text{for } -\left(\dfrac{N-1}{2}\right) \le n \le \left(\dfrac{N-1}{2}\right) \\ 0, & \text{otherwise} \end{cases}$$

$$w_B(n) = \begin{cases} 0.42 - 0.5\cos\dfrac{2n\pi}{N-1} + 0.08\cos\dfrac{4n\pi}{N-1}, & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

In the magnitude response, the width of the main lobe is $12\pi/N$, which is highest among windows. The peak of the first side lobe is at –58 dB and the side lobe magnitude decreases with frequency. This desirable feature is achieved at the expense of increased main lobe width. However, the main lobe width can be reduced by increasing the value of N. The side lobe attenuation of a low-pass filter using Blackman window is –78 dB.

Table 1 gives the important frequency domain characteristics of some window functions.

**TABLE 1** Frequency domain characteristics of some window functions.

| *Type of window* | *Approximate transition width of main lobe* | *Minimum stop band attenuation* (dB) | *Peak of first side lobe* (dB) |
|---|---|---|---|
| Rectangular | $4\pi/N$ | –21 | –13 |
| Bartlett | $8\pi/N$ | –25 | –25 |
| Hanning | $8\pi/N$ | – 44 | –31 |
| Hamming | $8\pi/N$ | –51 | – 41 |
| Blackmann | $12\pi/N$ | –78 | –58 |

**EXAMPLE 3**

Design an ideal low-pass filter with $N = 11$ with a frequency response

$$H_d(e^{j\omega}) = \begin{cases} 1, & \text{for } -\dfrac{\pi}{2} \le \omega \le \dfrac{\pi}{2} \\ 0, & \text{for } \dfrac{\pi}{2} \le |\omega| \le \pi \end{cases}$$

***Solution:*** For the given desired frequency response,

$$H_d(\omega) = \begin{cases} 1, & \text{for } -\dfrac{\pi}{2} \le \omega \le \dfrac{\pi}{2} \\ 0, & \text{for } \dfrac{\pi}{2} \le |\omega| \le \pi \end{cases}$$

The filter coefficients are given by

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

$$= \frac{1}{2\pi}\int_{-\pi/2}^{\pi/2} (1)\, e^{j\omega n}\, d\omega \;= \frac{1}{2\pi}\left[\frac{e^{j\omega n}}{jn}\right]_{\frac{-\pi}{2}}^{\frac{\pi}{2}}$$

$$= \frac{1}{n\pi}\left[\frac{e^{j\frac{n\pi}{2}} - e^{-j\frac{n\pi}{2}}}{2j}\right]$$

$$= \frac{1}{n\pi}\sin\frac{n\pi}{2} \quad \text{for } n \neq 0$$

$h_d(n) = \frac{1}{2}$ for $n = 0$ [using L'Hospital rule]

$h_d(0) = \frac{1}{2}$, $\quad h_d(1) = \frac{1}{\pi}\sin\frac{\pi}{2} = \frac{1}{\pi} = h_d(-1)$

$h_d(2) = \frac{1}{2\pi}\sin\pi = 0 = h_d(-2)$, $\quad h_d(3) = \frac{1}{3\pi}\sin\frac{3\pi}{2} = -\frac{1}{3\pi} = h_d(-3)$

$h_d(4) = \frac{1}{4\pi}\sin 2\pi = 0 = h_d(-4)$, $\quad h_d(5) = \frac{1}{5\pi}\sin\frac{5\pi}{2} = \frac{1}{5\pi} = h_d(-5)$

Assuming the window function,

$$w(n) = \begin{cases} 1, & \text{for } -5 \le n \le 5 \\ 0, & \text{otherwise} \end{cases}$$

$$h(n) = h_d(n) \cdot w(n) = h_d(n)$$

We have

Therefore, the designed filter coefficients are given as

$$h(0) = \frac{1}{2}, h(1) = \frac{1}{\pi} = h(-1), \quad h(2) = 0 = h(-2), \quad h(3) = -\frac{1}{3\pi} = h(-3),$$

$$h(4) = 0 = h(-4), \quad h(5) = \frac{1}{5\pi} = h(-5)$$

The above coefficients correspond to a non-causal filter which is not realizable.

The realizable digital filter transfer function $H(z)$ is given by

$$H(z) = z^{-(N-1)/2}\left[h(0) + \sum_{n=1}^{(N-1)/2} h(n)\,[z^{-n} + z^{n}]\right] = z^{-5}\left[h(0) + \sum_{n=1}^{5} h(n)[z^{-n} + z^{n}]\right]$$

$$= z^{-5}\left[h(0) + h(1)[z + z^{-1}] + h(3)[z^3 + z^{-3}] + h(5)[z^5 + z^{-5}]\right]$$

$$= h(5) + h(3)z^{-2} + h(1)z^{-4} + h(0)z^{-5} + h(1)z^{-6} + h(3)z^{-8} + h(5)z^{-10}$$

$$= \frac{1}{5\pi} - \frac{1}{3\pi}z^{-2} + \frac{1}{\pi}z^{-4} + \frac{1}{2}z^{-5} + \frac{1}{\pi}z^{-6} - \frac{1}{3\pi}z^{-8} + \frac{1}{5\pi}z^{-10}$$

Therefore, the coefficients of the realizable digital filter are:

$$h(0) = \frac{1}{5\pi} = h(10),\ h(1) = 0 = h(9),\ h(2) = -\frac{1}{3\pi} = h(8),$$

$$h(3) = 0 = h(7),\ h(4) = \frac{1}{\pi} = h(6),\ h(5) = \frac{1}{2}$$

## Designing Optimum FIR Filter using the Park-McClellan Algorithm

The Parks-McClellan algorithm, or Remez Exchange, uses an iterative technique based on an error criterion to design FIR filter coefficients. Parks-McClellan algorithm is used to design optimum, linear-phase, FIR filter coefficients. Filters that design with the Parks-McClellan algorithm are optimal because they minimize the maximum error between the actual magnitude response of the filter and the ideal magnitude response of the filter.

Designing optimum FIR filters reduces adverse effects at the cut-off frequencies. Designing optimum FIR filters also offers more control over the approximation errors in different frequency bands than other FIR filter design techniques, such as designing FIR filters by windowing, which provides no control over the approximation errors in different frequency bands.

Optimum FIR filters that can design using the Parks-McClellan algorithm have the following characteristics:

- A magnitude response with the weighted ripple evenly distributed over the passband and stopband
- A sharp transition band

FIR filter design using the Parks-McClellan algorithm have an optimal response. However, the design process is complex and requires a large amount of time and computational resources than designing FIR filters by windowing.

## Designing Equi ripple FIR Filter using the Park-McClellan Algorithm

Parks-McClellan algorithm can be used to design equiripple FIR filters. Equiripple design equally weights the passband and stopband ripple and produces filters with a linear phase characteristic.
Specify the following filter characteristics to design an equi ripple FIR filter:

- Cut-off frequency
- Number of taps
- Filter type, such as lowpass, high pass, bandpass, or band stop
- Pass frequency
- Stop frequency

The cut-off frequency for equi ripple filters specifies the edge of the passband, the stopband, or both. The ripple in the passband and stopband of equi ripple filters causes the following magnitude responses:

- Passband—a magnitude response greater than or equal to 1
- Stopband—a magnitude response less than or equal to the stopband attenuation

For example, to specify a lowpass filter, the passband cut-off frequency is the highest frequency for which the passband conditions are true. Similarly, the stopband cut-off frequency is the lowest frequency for which the stopband conditions are true.

## Park-McClellan Algorithm

One of the best "catch-all" algorithms used to determine the filter coefficients is the Parks-McClellan algorithm. Once the specifications are obtained (cutoff frequency, attenuation, band of filter), they can be supplied as parameters to the function, and the output of the function will be the coefficients for the filter. The program works by spreading out the error over the entire frequency response. So, an equal amount of "minimized" error will be present in the passband and stopband ripple. Also, the Parks-McClellan algorithm isn't limited to the types of filters discussed earlier (low-pass, high-pass). It can have as many bands as are desired, and the error in each band can be weighted. This facilitates building filters of arbitrary frequency response. To design the filter, first calculate the order of the filter with the following equations:

$$\hat{M} = \frac{-20\log_{10}\sqrt{A\delta_1\delta_2} - 13}{14.6\Delta f}; \qquad \Delta f = \frac{W_s - W_p}{2\pi}$$

where M is the order, $w_s$ and $w_p$ are the passband and stopband frequencies, and $\delta_1$ and $\delta_2$ are the ripple on the passband and stopband.

$\delta_1$ and $\delta_2$ are calculated from the desired passband ripple and stopband attenuation with the following formulas.

$$\delta_1 = 10^{Ap\,/\,20} - 1 and \delta_2 = 10^{-As\,/\,20}$$

# UNIT - IV

# IIR DIGITAL FILTERS

## Introduction

Filters are of two types—FIR and IIR. The types of filters which make use of feedback connection to get the desired filter implementation are known as recursive filters. Their impulse response is of infinite duration. So they are called IIR filters. The type of filters which do not employ any kind of feedback connection are known as non-recursive filters. Their impulse response is of finite duration. So they are called FIR filters. IIR filters are designed by considering all the infinite samples of the impulse response. The impulse response is obtained by taking inverse Fourier transform of ideal frequency response. There are several techniques available for the design of digital filters having an infinite duration unit impulse response. The popular methods for such filter design uses the technique of first designing the digital filter in analog domain and then transforming the analog filter into an equivalent digital filter because the analog filter design techniques are well developed. Various methods of transforming an analog filter into a digital filter and methods of designing digital filters are discussed.

## Requirements for transformation

The system function describing an analog filter may be written as

$$H_a(s) = \frac{Y(s)}{X(s)} = \frac{\sum_{k=0}^{M} b_k s^k}{\sum_{k=}^{N} a_k s^k}$$

where $\{a_k\}$ and $\{b_k\}$ are filter coefficients.

The impulse response of these filter coefficients is related to $H_a(s)$ by the Laplace transform.

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

The analog filter having the rational system function $H_a(s)$ is expressed by a linear constant coefficient differential equation.

$$\sum_{k=0}^{N} a_k \frac{d^k y(t)}{dt^k} = \sum_{k=0}^{M} b_k \frac{d^k x(t)}{dt^k}$$

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

The above three equivalent characterizations of an analog filter leads to three alternative methods for transforming the analog filter into digital domain. The restriction on the design is that the filters should be realizable and stable.

For stability and causality of analog filter, the analog transfer function should satisfy the following requirements:

1. The $H_a(s)$ should be a rational function of $s$, and the coefficients of $s$ should be real.
2. The poles should lie on the left half of $s$-plane.
3. The number of zeros should be less than or equal to the number of poles.

For stability and causality of digital filter, the digital transfer function should satisfy the following requirements:

1. The $H(z)$ should be a rational function of $z$ and the coefficients of $z$ should be real.
2. The poles should lie inside the unit circle in $z$-plane.
3. The number of zeros should be less than or equal to the number of poles.

We know that the analog filter with transfer function $H_a(s)$ is stable if all its poles lie in the left half of the $s$-plane. Consequently for the conversion technique to be effective, it should possess the following desirable properties:

1. The imaginary 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 of the s-plane should map into the interior of the unit circle centered at the origin in $z$-plane. Thus, a stable analog filter will be converted to a stable digital filter.

The physically realizable and stable IIR filter cannot have a linear phase. For a filter to have a linear phase, the condition to be satisfied is $h(n) = h(N - 1 - n)$ where $N$ is the length of the filter and the filter would have a mirror image pole outside the unit circle for every pole inside the unit circle. This results in an unstable filter. As a result, a causal and stable IIR filter cannot have linear phase. In the design of IIR filters, only the desired magnitude response is specified and the phase response that is obtained from the design methodology is accepted.

The comparison of digital and analog filters is given below.

**TABLE 1** Comparison of Digital and Analog Filters

| *Digital filter* | *Analog filter* |
|---|---|
| 1. It operates on digital samples (or sampled version) of the signal. | 1. It operates on analog signals (or actual signals). |
| 2. It is governed (or defined) by linear difference equations. | 2. It is governed (or defined) by linear differential equations. |
| 3. It consists of adders, multipliers, and delay elements implemented in digital logic (either in hardware or software or both). | 3. It consists of electrical components like resistors, capacitors, and inductors. |
| 4. In digital filters, the filter coefficients are designed to satisfy the desired frequency response. | 4. In analog filters, the approximation problem is solved to satisfy the desired frequency response. |

### *Advantages of digital filters*

1. The values of resistors, capacitors and inductors used in analog filters change with temperature. Since the digital filters do not have these components, they have high thermal stability.
2. In digital filters, the precision of the filter depends on the length (or size) of the registers used to store the filter coefficients. Hence by increasing the register bit length (in hardware) the performance characteristics of the filter like accuracy, dynamic range, stability and frequency response tolerance, can be enhanced.
3. The digital filters are programmable. Hence the filter coefficients can be changed any time to implement adaptive features.
4. A single filter can be used to process multiple signals by using the techniques of multiplexing.

### *Disadvantages of digital filters*

1. The bandwidth of the discrete signal is limited by the sampling frequency. The bandwidth of real discrete signal is half the sampling frequency.
2. The performance of the digital filter depends on the hardware (i.e., depends on the bit length of the registers in the hardware) used to implement the filter.

### *Features of IIR filters*

1. The physically realizable IIR filters do not have linear phase.
2. The IIR filter specifications include the desired characteristics for the magnitude response only.

## DESIGN OF IIR FILTER BY BILINEAR TRSFORMATION METHOD

IIR filter can be designed using (a) approximation of derivatives method and (b) Impulse invariant transformation method. However the IIR filter design using these methods is appropriate only for the design of low-pass filters and band pass filters whose resonant frequencies are small. These techniques are not suitable for high-pass or band rejects filters. The limitation is overcome in the mapping technique called the **bilinear transformation.** This transformation is a one-to-one mapping from the *s*-domain to the *z*-domain. That is, the bilinear transformation is a conformal mapping that transforms the imaginary axis of *s*-plane into the unit circle in the *z*-plane only once, thus avoiding aliasing of frequency components. In this mapping, all points in the left half of *s*-plane are mapped inside the unit circle in the *z*-plane, and all points in the right half of *s*-plane are mapped outside the unit circle in the *z*-plane. So the transformation of a stable analog filter results in stable digital filter. The bilinear transformation can be obtained by using the trapezoidal formula for the numerical integration.

Let the system function of the analog filter be $H_a(s) = \frac{b}{s+a}$

The differential equation describing the above analog filter can be obtained as:

$$H_a(s) = \frac{Y(s)}{X(s)} = \frac{b}{s+a}$$

or $$sY(s) + aY(s) = bX(s)$$

Taking inverse Laplace transform on both sides, we get

$$\frac{dy(t)}{dt} + a\,y(t) = bx(t)$$

Integrating the above equation between the limits $(nT - T)$ and $nT$, we have

$$\int_{nT-T}^{nT} \frac{dy(t)}{dt}\,dt + a\int_{nT-T}^{nT} y(t)\,dt = b\int_{nT-T}^{T} x(t)\,dt$$

The trapezoidal rule for numeric integration is expressed as:

$$\int_{nT-T}^{nT} a(t)\,dt = \frac{T}{2}[a(nT) + a(nT-T)]$$

Therefore, we get

$$y(nT) - y(nT-T) + a\frac{T}{2}y(nT) + a\frac{T}{2}y(nT-T) = b\frac{T}{2}x(nT) + b\frac{T}{2}x(nT-T)$$

Taking $z$-transform, we get

$$Y(z)[1 - z^{-1}] + a\frac{T}{2}[1 + z^{-1}]\,Y(z) = b\frac{T}{2}[1 + z^{-1}]\,X(z)$$

Therefore, the system function of the digital filter is:

$$\frac{Y(z)}{X(z)} = H(z) = \frac{b}{\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} + a}$$

Comparing this with the analog filter system function $H_a(s)$ we get

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

Rearranging, we can get

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

This is the relation between analog and digital poles in bilinear transformation. So to convert an analog filter function into an equivalent digital filter function, just put

$$s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} \text{ in } H_a(s)$$

The general characteristic of the mapping $z = e^{sT}$ may be obtained by putting $s = \sigma + j\Omega$ and expressing the complex variable $z$ in the polar form as $z = re^{j\omega}$ in the above equation for $s$.

Thus,

$$s = \frac{2}{T}\left(\frac{z-1}{z+1}\right) = \frac{2}{T}\left(\frac{re^{j\omega} - 1}{re^{j\omega} + 1}\right)$$

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

Since $s = \sigma + j\Omega$, we get

$$\sigma = \frac{2}{T}\left[\frac{r^2-1}{1+r^2+2r\cos\omega}\right]$$

And

$$\Omega = \frac{2}{T}\left[\frac{2r\sin\omega}{1+r^2+2r\cos\omega}\right]$$

From the above equation for , we observe that if $r < 1$ then $\sigma < 0$ and if $r > 1$, then $\sigma > 0$, and if $r = 1$, then $\sigma = 0$. Hence the left half of the *s*-plane maps into points inside the unit circle in the *z*-plane, the right half of the *s*-plane maps into points outside the unit circle in the *z*-plane and the imaginary axis of *s*-plane maps into the unit circle in the *z*-plane. This transformation results in a stable digital system.

### *Relation between analog and digital frequencies*

On the imaginary axis of *s*-plane $\sigma = 0$ and correspondingly in the *z*-plane $r = 1$.

$$\Omega = \frac{2}{T}\left(\frac{2\sin\omega}{1+1+2\cos\omega}\right) = \frac{2}{T}\left(\frac{\sin\omega}{1+\cos\omega}\right)$$

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

The relation between analog and digital frequencies is:

$$\Omega = \frac{2}{T}\tan\frac{\omega}{2}$$

or equivalently, we have $= 2\tan\frac{\Omega T}{2}$.

The above relation between analog and digital frequencies shows that the entire range in $\Omega$ is mapped only once into the range $-\pi \le \omega \le \pi$. The entire negative imaginary axis in the *s*-plane (from $\Omega = -\infty$ to 0) is mapped into the lower half of the unit circle in *z*-plane (from $\omega = -\pi$ to 0) and the entire positive imaginary axis in the *s*-plane (from $\Omega = \alpha$ to 0) is mapped into the upper half of unit circle in *z*-plane (from $\omega = 0$ to $+\pi$).

But as seen in Figure 1, the mapping is non-linear and the lower frequencies in analog domain are expanded in the digital domain, whereas the higher frequencies are

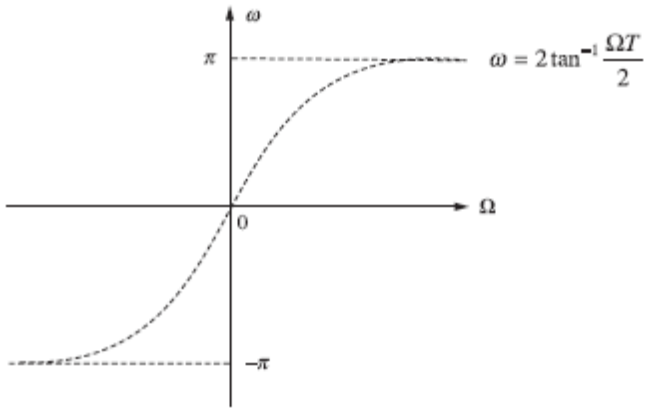



**Figure 1 M**apping between Ω and ω in bilinear transformation.

compressed. This is due to the nonlinearity of the arctangent function and usually known as frequency warping.

The effect of warping on the magnitude response can be explained by considering an analog filter with a number of passbands as shown in Figure 2(a). The corresponding digital filter will have same number of passbands, but with disproportionate bandwidth, as shown in Figure 2(a).

In designing digital filter using bilinear transformation, the effect of warping on amplitude response can be eliminated by prewarping the analog filter. In this method, the specified digital frequencies are converted to analog equivalent using the equation $\Omega = \frac{2}{T}\tan\frac{\omega}{2}$. These analog frequencies are called prewarp frequencies. Using the prewarp

frequencies, the analog filter transfer function is designed, and then it is transformed to digital filter transfer function.

This effect of warping on the phase response can be explained by considering an analog filter with linear phase response as shown in Figure 2(b). The phase response of corresponding digital filter will be nonlinear.

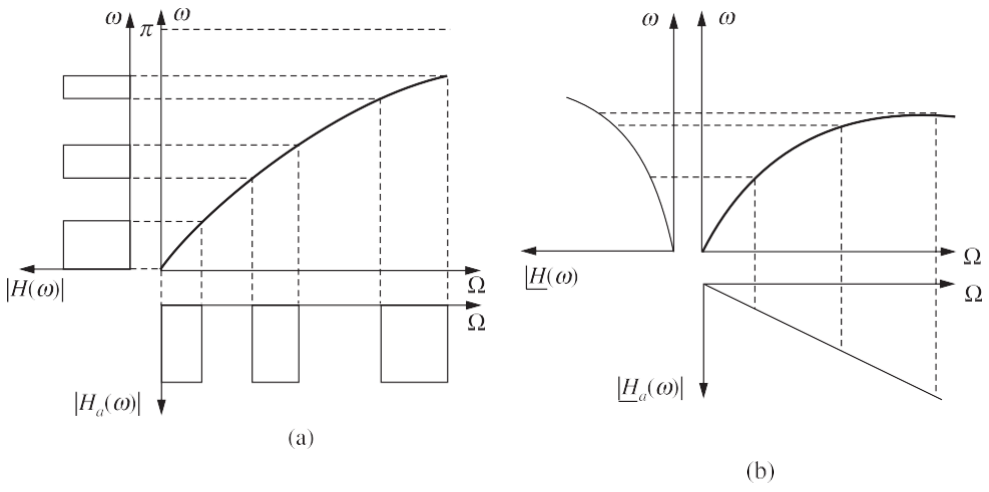



**Figure 2 T**he warping effect on (a) magnitude response and (b) phase response.

It can be stated that the bilinear transformation preserves the magnitude response of an analog filter only if the specification requires piecewise constant magnitude, but the phase response of the analog filter is not preserved. Therefore, the bilinear transformation can be used only to design digital filters with prescribed magnitude response with piecewise constant values. A linear phase analog filter cannot be transformed into a linear phase digital filter

using the bilinear transformation.

**EXAMPLE 1**

Convert the following analog filter with transfer function

$$H_a(s) = \frac{s + 0.1}{(s + 0.1)^2 + 9}$$

into a digital IIR filter by using bilinear transformation. The digital IIR filter is having a resonant frequency of $\omega_r = \pi/2$.

***Solution:*** From the transfer function, we observe that $\Omega_c = 3$. The sampling period $T$ can be determined using the equation:

$$\Omega_c = \frac{2}{T} \tan \frac{\omega_r}{2}$$

$$T = \frac{2}{\Omega_c} \tan \frac{\omega_r}{2} = \frac{2}{3} \tan \frac{\pi/2}{2} = 0.6666 \text{ s}$$

Using the bilinear transformation, the digital filter system function is:

$$H(z) = H_a(s)\Big|_{s=\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}} = H_a(s)\Big|_{s=3\frac{1-z^{-1}}{1+z^{-1}}}$$

$$H(z) = \frac{s + 0.1}{(s + 0.1)^2 + 9}\Bigg|_{s=3\frac{1-z^{-1}}{1+z^{-1}}}$$

$$= \frac{3\dfrac{1-z^{-1}}{1+z^{-1}} + 0.1}{\left[3\dfrac{1-z^{-1}}{1+z^{-1}} + 0.1\right]^2 + 9}$$

$$= \frac{\left[3(1-z^{-1}) + 0.1(1+z^{-1})\right][1+z^{-1}]}{\left[3(1-z^{-1}) + 0.1(1+z^{-1})\right]^2 + 9(1+z^{-1})^2}$$

$$= \frac{3.1 + 0.2z^{-1} - 2.9z^{-2}}{18.61 + 0.02z^{-1} + 17.41z^{-2}}$$

**EXAMPLE 2**

Convert the analog filter with system function

$$H_a(s) = \frac{s + 0.5}{(s + 0.5)^2 + 16}$$

into a digital IIR filter using the bilinear transformation. The digital filter should have a resonant frequency of $\omega_r = \pi/2$.

***Solution:*** From the system function, we observe that $\Omega_c = 4$. The sampling period $T$ can be

determined using the equation $\Omega = \frac{2}{T}\tan\frac{\omega}{2}$

$$\Omega_c = \frac{2}{T}\tan\frac{\omega_r}{2}$$

i.e. $$T = \frac{2}{\Omega_c}\tan\frac{\omega_r}{2} = \frac{2}{4}\tan\frac{\pi}{4} = 0.5 \text{ s}$$

Using the bilinear transformation, the digital filter system function is:

$$H(z) = H(s)\Big|_{s=\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)} = H(s)\Big|_{s=4\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}$$

$$H(z) = \frac{s+0.5}{(s+0.5)^2+16}\Big|_{s=4\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}$$

$$= \frac{4\left(\frac{1-z^{-1}}{1+z^{-1}}\right)+0.5}{\left[4\left(\frac{1-z^{-1}}{1+z^{-1}}\right)+0.5\right]^2+16}$$

$$= \frac{\left[4(1-z^{-1})+0.5(1+z^{-1})\right][1+z^{-1}]}{\left[4(1-z^{-1})+0.5(1+z^{-1})\right]^2+16[1+z^{-1}]^2}$$

$$= \frac{4.5+z^{-1}-3.5z^{-2}}{36.25+0.5z^{-1}+28.25z^{-2}}$$

**EXAMPLE 3**

Apply the bilinear transformation to

$$H_a(s) = \frac{4}{(s+3)(s+4)}$$

with $T = 0.5$ s and find $H(z)$.

***Solution:*** Given that

$$H_a(s) = \frac{4}{(s+3)(s+4)}$$

and $T = 0.5$ s

To obtain $H(z)$ using the bilinear transformation in $H_a(s)$ , replace $s$ by

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

$$H(z) = \frac{4}{(s+3)(s+4)}\bigg|_{s=\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}} = \frac{4}{(s+3)(s+4)}\bigg|_{s=4\frac{1-z^{-1}}{1+z^{-1}}}$$

$$= \frac{4}{\left[4\left(\frac{1-z^{-1}}{1+z^{-1}}\right)+3\right]\left[4\left(\frac{1-z^{-1}}{1+z^{-1}}\right)+4\right]}$$

$$= \frac{4}{\left[\frac{4-4z^{-1}+3+3z^{-1}}{1+z^{-1}}\right]\left[\frac{4-4z^{-1}+4+4z^{-1}}{1+z^{-1}}\right]}$$

$$= \frac{4(1+z^{-1})^2}{(7-z^{-1})\,8}$$

$$= \frac{1}{2}\,\frac{(1+z^{-1})^2}{(7-z^{-1})}$$

**EXAMPLE 4**

Obtain $H(z)$ from $H_a(s)$ when $T = 1$ s and

$$H_a(s) = \frac{3s}{s^2+0.5s+2}$$

using the bilinear transformation.

***Solution:*** Given

$$H_a(s) = \frac{3s}{s^2+0.5s+2}$$ and $T = 1$ s.

To get $H(z)$ using the bilinear transformation, put

$$s = \frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) \text{ in } H_a(s).$$

$$H(z) = H_a(s)\bigg|_{s=\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)} = \frac{3s}{s^2+0.5s+2}\bigg|_{s=\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}$$

$$= \frac{3\times 2\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}{\left[2\left(\frac{1-z^{-1}}{1+z^{-1}}\right)\right]^2 + 0.5\left[2\left(\frac{1-z^{-1}}{1+z^{-1}}\right)\right]+2}$$

$$= \frac{6\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}{\frac{4(1-z^{-1})^2+(1-z^{-1})(1+z^{-1})+2(1+z^{-1})^2}{(1+z^{-1})^2}}$$

$$= \frac{6(1+z^{-1})}{4(1-2z^{-1}+z^{-2})+(1-z^{-2})+2(1+2z^{-1}+z^{-2})}$$

$$= \frac{6+6z^{-1}}{7-4z^{-1}+5z^{-2}}$$

**EXAMPLE 5**

Using the bilinear transformation, obtain $H(z)$ from $H_a(s)$ when $T$ = 1s and $H_a(s) = \dfrac{s^3}{(s+1)(s^2+2s+2)}$

***Solution:*** Given that

$$H_a(s) = \frac{s^3}{(s+1)(s^2+2s+2)} \text{ and } T = 1 \text{ s.}$$

To obtain $H(z)$ using the bilinear transformation,

$$\text{put } s = \frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) \text{ in } H_a(s).$$

Given $T$ = 1 s,

$$H(z) = H_a(s)\Big|_{s=\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)} = \frac{s^3}{(s+1)(s^2+2s+2)}\Bigg|_{s=2\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}$$

$$= \frac{\left[2\frac{(1-z^{-1})}{(1+z^{-1})}\right]^3}{\left[2\frac{(1-z^{-1})}{1+z^{-1}}+1\right]\left\{\left[2\frac{(1-z^{-1})}{1+z^{-1}}\right]^2+2\left[2\frac{(1-z^{-1})}{1+z^{-1}}\right]+2\right\}}$$

$$= \frac{8(1-z^{-1})^3}{\left[2(1-z^{-1})+(1+z^{-1})\right]\left[4(1-z^{-1})^2+4(1-z^{-1})(1+z^{-1})+2(1+z^{-1})^2\right]}$$

$$= \frac{8(1-z^{-1})^3}{(3-z^{-1})[10-4z^{-1}+2z^{-2}]}$$

$$= \frac{4(1-z^{-1})^3}{(3-z^{-1})(5-2z^{-1}+2z^{-2})}$$

$$= 4\frac{(1-3z^{-1}+3z^{-2}-z^{-3})}{15-11z^{-1}+8z^{-2}-2z^{-3}}$$

**EXAMPLE 6**

A digital filter with a 3 dB bandwidth of 0.4 is to be designed from the analog filter whose system response is:

$$H(s) = \frac{\Omega_c}{s + 2\Omega_c}$$

Use the bilinear transformation and obtain $H(z)$.

***Solution:*** We know that $\Omega_c = \frac{2}{T} \tan \frac{\omega_c}{2}$

Here the 3 dB bandwidth $\omega_c = 0.4$

$$\Omega_c = \frac{2}{T} \tan \frac{0.4\pi}{2} = \frac{1.453}{T}$$

The system response of the digital filter is given by

$$H(z) = H_a(s)\Big|_{s=\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}$$

$$= \frac{\Omega_c}{\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) + 2\Omega_c} = \frac{\frac{1.453}{T}}{\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) + 2\left(\frac{1.453}{T}\right)}$$

$$= \frac{1.453\,(1+z^{-1})}{2(1-z^{-1}) + 2(1+z^{-1})\,1.453}$$

$$= \frac{1+z^{-1}}{3.376 - 0.624z^{-1}}$$

**EXAMPLE 7**

The normalized transfer function of an analog filter is given by

$$H(s_n) = \frac{1}{s_n^2 + 1.6s_n + 1}$$

Convert the analog filter to a digital filter with a cutoff frequency of 0.6, using the bilinear transformation.

***Solution:*** The prewarping of analog filter has to be performed to preserve the magnitude response. For this the analog cutoff frequency is determined using the bilinear transformation, and the analog transfer function is unnormalized using this analog cutoff frequency. Then the analog transfer function is converted to digital transfer function using the bilinear transformation.

Given that, digital cutoff frequency, $\omega_c = 0.6\,\pi$ rad/s. Let $T = 1$s.

In the bilinear transformation,

Analog cutoff frequency

$$\Omega_c = \frac{2}{T} \tan \frac{\omega_c}{2} = 2\tan \frac{0.6\,\pi}{2} = 2.753 \text{ rad/s.}$$

Normalized analog transfer function

$$H_a(s_n) = \frac{1}{s_n^2 + 1.6s_n + 1}$$

The analog transfer function is unnormalized by replacing $s_n$ by $s/\Omega_c$. Therefore, unnormalized analog filter transfer function is given by

$$H_a(s) = \frac{1}{\left(\frac{s}{\Omega_c}\right)^2 + 1.6\left(\frac{s}{\Omega_c}\right) + 1} = \frac{1}{\left(\frac{s}{2.753}\right)^2 + 1.6\left(\frac{s}{2.753}\right) + 1}$$

$$= \frac{2.753^2}{s^2 + 1.6 \times 2.753 s + 2.753^2} = \frac{7.579}{s^2 + 4.404 s + 7.579}$$

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

The digital filter system function $H(z)$ is obtained by substituting $H_a(s)$. Here $T = 1$. Therefore, the digital filter transfer function is:

$$H(z) = \frac{7.579}{\left[2\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right)\right]^2 + 4.404\left[2\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right)\right] + 7.579}$$

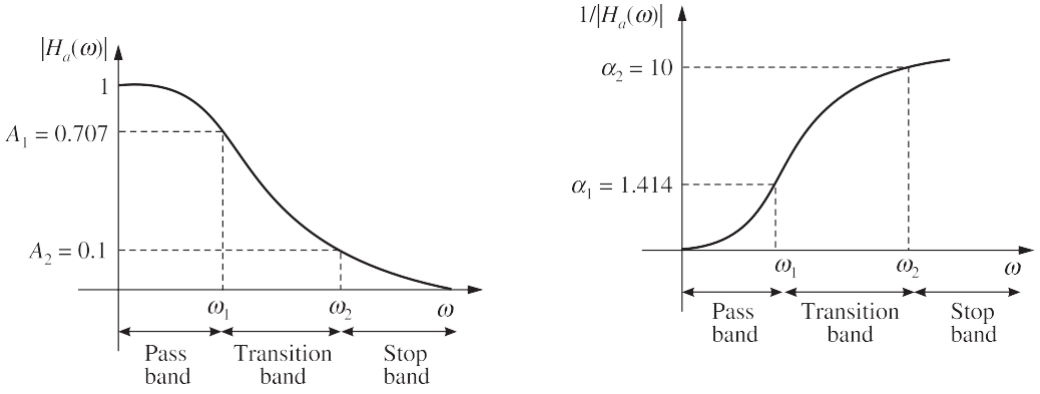
$$= \frac{7.579(1 + z^{-1})^2}{4(1 - 2z^{-1} + z^{-2}) + 4.404(1 + z^{-1})\,2(1 - z^{-1}) + 7.579(1 + z^{-1})^2}$$

$$= \frac{7.579[1 + 2z^{-1} + z^{-2}]}{20.387 + 7.158z^{-1} + 2.771z^{-2}}$$

$$= \frac{0.371 + 0.742z^{-1} + 0.371z^{-2}}{1 + 0.351z^{-1} + 0.136z^{-2}}$$

## SPECIFICATIONS OF THE LOW-PASS FILTER

The magnitude response of low-pass filter in terms of gain and attenuation are shown in Figure 1.




(a ) (b)

**Figure 3 M**agnitude response of low–pass filter (a) Gain vs ω and (b) Attenuation vs ω.

Let $\omega_1$ = Passband frequency in rad/s.
$\omega_2$ = Stopband frequency in rad/s.

Let the gain at the passband frequency $\omega_1$ be $A_1$ and the gain at the stopband frequency $\omega_2$ be $A_2$, i.e.

$$A_1 = |H(\omega)|_{\omega=\omega_1} \quad \text{and} \quad A_2 = |H(\omega)|_{\omega=\omega_2}$$

The filter may be expressed in terms of the gain or attenuation at the edge frequencies. Let $\alpha_1$ be the attenuation at the passband edge frequency $\omega_1$, and $\alpha_2$ be the attenuation at the stopband edge frequency $\omega_2$.

$$\alpha_1 = \frac{1}{A_1} = \frac{1}{|H(\omega)|_{\omega=\omega_1}} \quad \text{and} \quad \alpha_2 = \frac{1}{A_2} = \frac{1}{|H(\omega)|_{\omega=\omega_2}}$$

The maximum value of normalized gain is unity, so $A_1$ and $A_2$ are less than 1 and $\alpha_1$ and $\alpha_2$ are greater than 1. In Figure 1, $A_1$ is assumed as $1/\sqrt{2}$ and $A_2$ is assumed as 0.1. Hence $\alpha_1 = \sqrt{2} = 1.41$ and $\alpha_2 = 1/0.1 = 10$.

Another popular unit that is used for filter specification is dB. When the gain is expressed in dB, it will be a negative dB. When the attenuation is expressed in dB, it will be a positive dB.

Let $k_1$ = Gain in dB at a passband frequency $\omega_1$

$k_2$ = Gain in dB at a stopband frequency $\omega_2$

The gain can be converted into normal values as follows:

| | |
|---|---|
| $20 \log A_1 = k_1$ | $20 \log A_2 = k_2$ |
| $\log A_1 = k_1/20$ | $\log A_2 = k_2/20$ |
| $A_1 = 10^{k_1/20}$ | $A_2 = 10^{k_2/20}$ |

When expressed in dB, the gain and attenuation will have only change in sign because $\log \alpha = \log(1/A) = -\log A$. (Hence when dB is positive it is attenuation and when dB is negative it is gain).

When $A_1 = 0.707$, $k_1 = 20 \log(0.707) = -3.0116 = -3$ dB

When $A_2 = 0.1$, $k_2 = 20 \log(0.1) = -20$ dB

The magnitude response of low-pass filter in terms of dB-attenuation is shown in Figure 4.

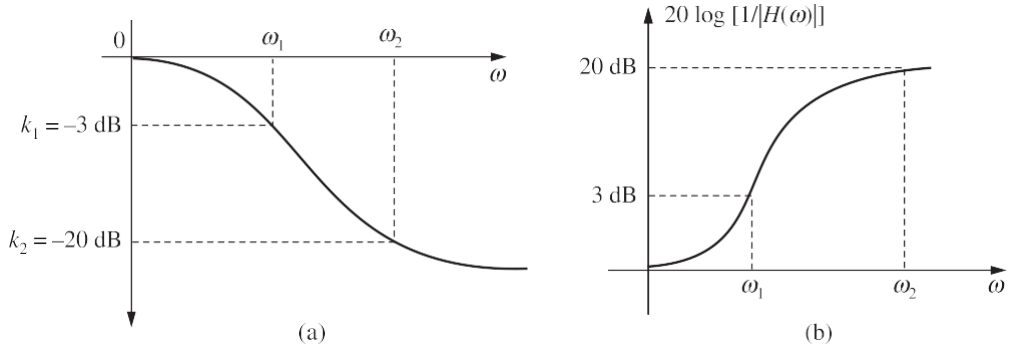



**Figure 4 M**agnitude response of low–pass filter (a) dB–Gain vs $\omega$ and (b) dB–attenuation vs $\omega$.

Sometimes the specifications are given in terms of passband ripple $\delta_p$ and stopband ripple $\delta_s$. In this case, the dB gain and attenuation can be estimated as follows:

$$k_1 = 20 \log (1 - \delta_p) \qquad \alpha_1 = -20 \log (1 - \delta_p)$$
$$k_2 = 20 \log \delta_s \qquad \alpha_2 = -20 \log \delta_s$$

If the ripples are specified in dB, then the minimum passband ripple is equal to $k_1$ and the negative of maximum passband attenuation is equal to $k_2$.

## DESIGN OF LOW-PASS DIGITAL BUTTERWORTH FILTER

The popular methods of designing IIR digital filter involves the design of equivalent analog filter and then converting the analog filter to digital filter. Hence to design a Butterworth IIR digital filter, first an analog Butterworth filter transfer function is determined using the given specifications. Then the analog filter transfer function is converted to a digital filter transfer function using either impulse invariant transformation or bilinear transformation

### *Analog Butterworth filter*

The analog Butterworth filter is designed by approximating the ideal frequency response using an error function. The error function is selected such that the magnitude is maximally flat in the passband and monotonically decreasing in the stopband. (Strictly speaking the magnitude is maximally flat at the origin, i.e., at $\Omega$ = 0, and monotonically decreasing with increasing $\Omega$ ).

The magnitude response of low-pass filter obtained by this approximation is given by

$$|H_a(\omega)|^2 = \frac{1}{1 + \left(\frac{\Omega}{\Omega_c}\right)^{2N}}$$

where $\Omega_c$ is the 3 dB cutoff frequency and $N$ is the order of the filter.

### *Frequency response of the Butterworth filter*

The frequency response of Butterworth filter depends on the order $N$. The magnitude response for different values of $N$ are shown in Figure 5. From Figure 5, it can be observed that the approximated magnitude response approaches the ideal response as the value of $N$ increases. However, the phase response of the Butterworth filter becomes more nonlinear with increasing $N$.

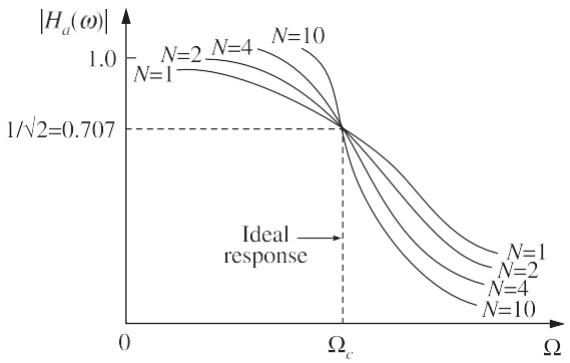



**Figure 5 M**agnitude response of Butterworth low–pass filter for various values of *N*.

### *Order of the filter*

Since the frequency response of the filter depends on its order $N$, the order $N$ has to be estimated to satisfy the given specifications.

Usually the specifications of the filter are given in terms of gain $A$ or attenuation at a passband or stopband frequency as given below:

$$A_1 \le |H(\omega)| \le 1, \qquad 0 \le \omega \le \omega_1$$
$$|H(\omega)| \le A_2, \qquad \omega_2 \le \omega \le \pi$$

The order of the filter is determined as given below.

Let $\Omega_1$ and $\Omega_2$ be the analog filter edge frequencies corresponding to digital frequencies $\omega_1$ and $\omega_2$. The values of $\Omega_1$ and $\Omega_2$ are obtained using the bilinear transformation or impulse invariant transformation.

$$A_1^2 \le \frac{1}{1+\left(\frac{\Omega_1}{\Omega_c}\right)^{2N}} \le 1$$

$$\frac{1}{1+\left(\frac{\Omega_2}{\Omega_c}\right)^{2N}} \le A_2^2$$

These two equations can be written in the form

$$\left(\frac{\Omega_1}{\Omega_c}\right)^{2N} \le \frac{1}{A_1^2} - 1$$

And

$$\left(\frac{\Omega_2}{\Omega_c}\right)^{2N} \ge \frac{1}{A_2^2} - 1$$

Assuming equality we can obtain the filter order $N$ and the 3 dB cutoff frequency $\Omega_c$. Dividing the first equation by the second, we have

$$\left(\frac{\Omega_1}{\Omega_2}\right)^{2N} = \frac{\frac{1}{A_1^2} - 1}{\frac{1}{A_2^2} - 1}$$

From this equation, the order of the filter $N$ is obtained approximately as

$$N = \frac{1}{2} \frac{\log\left\{\left(\frac{1}{A_2^2} - 1\right) \Big/ \left(\frac{1}{A_1^2} - 1\right)\right\}}{\log \frac{\Omega_2}{\Omega_1}}$$

If $N$ is not an integer, the value of $N$ is chosen to be the next nearest integer. Also we can get

$$\Omega_c = \frac{\Omega_1}{\left[\frac{1}{A_1^2} - 1\right]^{1/2N}}$$

when parameters $A_1$ and $A_2$ are given in dB.

$A_1$ in dB is given by

$$A_1 \text{ dB} = -20 \log A_1$$

i.e.

$$\log A_1 = -\frac{A_1 \text{ dB}}{20}$$

or

$$A_1 = 10^{-\frac{A_1 \text{ dB}}{20}}$$

i.e.

$$\frac{1}{A_1^2} - 1 = \frac{1}{\left(10^{-\frac{A_1 \text{ dB}}{20}}\right)^2} - 1$$

$$\frac{1}{A_1^2} - 1 = 10^{0.1A_1 \text{ dB}} - 1$$

Similarly

$$\frac{1}{A_2^2} - 1 = 10^{0.1A_2 \text{ dB}} - 1$$

$$N = \frac{1}{2} \frac{\log\{[\frac{1}{A_2^2} - 1]/[\frac{1}{A_1^2} - 1]\}}{\log\left(\frac{\Omega_2}{\Omega_1}\right)} = \frac{1}{2} \frac{\log\left(\frac{10^{0.1A_2 \text{dB}} - 1}{10^{0.1A_1 \text{dB}} - 1}\right)}{\log\left(\frac{\Omega_2}{\Omega_1}\right)}$$

and $\Omega_c$ is given by

$$\Omega_c = \frac{\Omega_1}{(10^{0.1A_1 \text{dB}} - 1)^{1/2N}} \quad \text{or} \quad \Omega_c = \frac{\Omega_2}{(10^{0.1A_2 \text{dB}} - 1)^{1/2N}}$$

In fact,

$$\Omega_c = \frac{1}{2}\left[\frac{\Omega_1}{\left[10^{0.1A_1 \text{dB}} - 1\right]^{1/2N}} + \frac{\Omega_2}{\left[10^{0.1A_2 \text{dB}} - 1\right]^{1/2N}}\right]$$

### *Butterworth low-pass filter transfer function*

The unnormalized transfer function of the Butterworth filter is usually written in factored form as:

$$H_a(s) = \prod_{k=1}^{N/2} \frac{\Omega_c^2}{s^2 + b_k \Omega_c s + \Omega_c^2} \quad \text{(when } N \text{ is even)}$$

or

$$H_a(s) = \frac{\Omega_c}{s + \Omega_c} \prod_{k=1}^{\frac{N-1}{2}} \frac{\Omega_c^2}{s^2 + b_k \Omega_c s + \Omega_c^2} \quad \text{(when } N \text{ is odd)}$$

Where

$$b_k = 2 \sin\left[\frac{(2k-1)\pi}{2N}\right]$$

If $s/\Omega_c$ (where $\Omega_c$ is the 3 dB cutoff frequency of the low-pass filter) is replaced by $s_n$, then the normalized Butterworth filter transfer function is given by

$$H_a(s) = \prod_{k=1}^{N/2} \frac{1}{s_n^2 + b_k s_n + 1} \quad \text{(when } N \text{ is even)}$$

or

$$H_a(s) = \frac{1}{s_n + 1} \prod_{k=1}^{\frac{N-1}{2}} \frac{1}{s_n^2 + b_k s_n + 1} \quad \text{(when } N \text{ is odd)}$$

where $b_k = 2 \sin\left[\frac{(2k-1)\pi}{2N}\right]$

***Design proceure for low-pass digital Butterworth IIR filter***

The low-pass digital Butterworth filter is designed as per the following steps:

Let $A_1$ = Gain at a passband frequency $\omega_1$

$A_2$ = Gain at a stopband frequency $\omega_2$

$\Omega_1$ = Analog frequency corresponding to $\omega_1$

$\Omega_2$ = Analog frequency corresponding to $\omega_2$

**Step 1** Choose the type of transformation, i.e., either bilinear or impulse invariant transformation.

**Step 2** Calculate the ratio of analog edge frequencies $\Omega_2/\Omega_1$.

For bilinear transformation

$$\Omega_1 = \frac{2}{T} \tan\frac{\omega_1}{2}, \Omega_2 = \frac{2}{T} \tan\frac{\omega_2}{2} \quad \therefore \frac{\Omega_2}{\Omega_1} = \frac{\tan \omega_2/2}{\tan \omega_1/2}$$

For impulse invariant transformation,

$$\Omega_1 = \frac{\omega_1}{T}, \quad \Omega_2 = \frac{\omega_2}{T} \quad \therefore \frac{\Omega_2}{\Omega_1} = \frac{\omega_2}{\omega_1}$$

**Step 3** Decide the order $N$ of the filter. The order $N$ should be such that

$$N \geq \frac{1}{2} \frac{\log\left\{\left[\frac{1}{A_2^2} - 1\right] \Big/ \left[\frac{1}{A_1^2} - 1\right]\right\}}{\log \frac{\Omega_2}{\Omega_1}}$$

Choose $N$ such that it is an integer just greater than or equal to the value obtained above.

**Step 4** Calculate the analog cutoff frequency

$$\Omega_c = \frac{\Omega_1}{\left[\frac{1}{A_1^2} - 1\right]^{1/2N}}$$

**Step 5** Determine the transfer function of the analog filter.
Let $H_a(s)$ be the transfer function of the analog filter. When the order $N$ is even, for unity dc gain filter, $H_a(s)$ is given by

$$H_a(s) = \prod_{k=1}^{N/2} \frac{\Omega_c^2}{s^2 + b_k \Omega_c s + \Omega_c^2}$$

When the order $N$ is odd, for unity dc gain filter, $H_a(s)$ is given by

$$H_a(s) = \frac{\Omega_c}{s + \Omega_c} \prod_{k=1}^{\frac{N-1}{2}} \frac{\Omega_c^2}{s^2 + b_k \Omega_c s + \Omega_c^2}$$

The coefficient $b_k$ is given by

$$b_k = 2 \sin\left[\frac{(2k-1)\pi}{2N}\right]$$

For normalized case, $\Omega_c$ = 1 rad/s

**Step 6** Using the chosen transformation, transform the analog filter transfer function $H_a(s)$ to digital filter transfer function $H(z)$.

**Step 7** Realize the digital filter transfer function $H(z)$ by a suitable structure.

***Poles of normalized Butterworth filter***

The Butterworth low-pass filter has a magnitude squared response given by

$$\left|H_a(\omega)\right|^2 = \frac{1}{1 + \left(\frac{\Omega}{\Omega_c}\right)^{2N}}$$

We know that the frequency response $H_a(\Omega)$ of an analog filter is obtained by substituting $s = j\Omega$ in the analog transfer function $H_a(s)$. Hence the system transfer function is obtained by replacing $\Omega$ by $(s/j)$ in the above equation.

$$H_a(s)\, H_a(-s) = \frac{1}{1 + \left(\frac{s}{j\Omega_c}\right)^{2N}} = \frac{1}{1 + \left(\frac{s^2}{j^2\Omega_c^2}\right)^{2N}}$$

In the above equation, when $s/\Omega_c$ is replaced by $Sn$ (i.e. $\Omega_c$ = 1 rad/s), the transfer function is called normalized transfer function.

$$H_a(s_n)H_a(-s_n) = \frac{}{1+(-s_n^2)N}$$

The transfer function of the above equation will have $2N$ poles which are given by the roots of the denominator polynomial. It can be shown that the poles of the transfer function symmetrically lie on a unit circle in $s$-plane with angular spacing of $\pi/N$.

For a stable and causal filter the poles should lie on the left half of the $s$-plane. Hence the desired filter transfer function is formed by choosing the $N$-number of left half poles. When $N$ is even, all the poles are complex and exist in conjugate pairs. When $N$ is odd, one of the pole is real and all other poles are complex and exist as conjugate pairs. Therefore, the transfer function of Butterworth filters will be a product of second order factors.

The poles of the Butterworth polynomial lie on a circle, whose radius is $\omega_c$. To determine the number of poles of the Butterworth filter and the angle between them we use the following rules.

- Number of Butterworth poles = $2N$
- Angle between any two poles = 360°/(2N)

If the order of the filter $N$ is even, then the location of the first pole is at $\theta/2$ w.r.t. the positive real axis, with the angle measured in the counter-clockwise direction. The location of the subsequent poles are respectively, at

$$\left(\frac{\theta}{2}+\theta\right), \left(\frac{\theta}{2}+2\theta\right), \left(\frac{\theta}{2}+3\theta\right), \ldots, \left(360-\frac{\theta}{2}\right)$$

If the order of the filter $N$ is odd, then the location of the first pole is on the $X$-axis. The location of subsequent poles are at θ , 2θ , ..., (360 –θ ) with the angle measured in the counter-clockwise direction.

If $\phi$ is the angle of a valid pole w.r.t. the $X$-axis, then the pole and its conjugate are located at $[\omega_c(\cos\phi \pm j \sin\phi)]$.

### *Properties of Butterworth filters*

1. The Butterworth filters are all pole designs (i.e. the zeros of the filters exist at œ).
2. The filter order $N$ completely specifies the filter.
3. The magnitude response approaches the ideal response as the value of $N$ increases.
4. The magnitude is maximally flat at the origin.
5. The magnitude is monotonically decreasing function of Ω.
6. At the cutoff frequency $\omega_c$, the magnitude of normalized Butterworth filter is $1/\sqrt{2}$. Hence the dB magnitude at the cutoff frequency will be 3 dB less than the maximum value.

**EXAMPLE 8**

Design a Butterworth digital filter using the bilinear transformation. The specifications of the desired low-pass filter are:

$$0.9 \le |H(\omega)| \le 1; \quad 0 \le \omega \le \frac{\pi}{2}$$

$$|H(\omega)| \le 0.2; \quad \frac{3\pi}{4} \le \omega \le \pi$$

with $T$ = 1 s

***Solution:*** The Butterworth digital filter is designed as per the following steps. From the given specification, we have

$$A_1 = 0.9 \text{ and } \omega_1 = \frac{\pi}{2}$$

$$A_2 = 0.2 \text{ and } \omega_2 = \frac{3\pi}{4} \qquad \text{and } T = 1 \text{ s}$$

**Step 1** Choice of the type of transformation
Here the bilinear transformation is already specified.

**Step 2** Determination of the ratio of the analog filter's edge frequencies, $\Omega_2/\Omega_1$

$$\Omega_2 = \frac{2}{T}\tan\frac{\omega_2}{2} = \frac{2}{1}\tan\left[\frac{(3\pi/4)}{2}\right] = 2\tan\frac{3\pi}{8} = 4.828$$

$$\Omega_1 = \frac{2}{T}\tan\frac{\omega_1}{2} = \frac{2}{1}\tan\left[\frac{(\pi/2)}{2}\right] = 2\tan\frac{\pi}{4} = 2$$

$$\frac{\Omega_2}{\Omega_1} = \frac{4.828}{2} = 2.414$$

**Step 3** Determination of the order of the filter $N$

$$N \geq \frac{1}{2}\frac{\log\left\{\left[\frac{1}{A_2^2}-1\right]\Big/\left[\frac{1}{A_1^2}-1\right]\right\}}{\log\frac{\Omega_2}{\Omega_1}}$$

$$\geq \frac{1}{2}\frac{\log\left\{\left[\frac{1}{(0.2)^2}-1\right]\Big/\left[\frac{1}{(0.9)^2}-1\right]\right\}}{\log 1.207}$$

$$\geq \frac{1}{2}\frac{\log\{24/0.2345\}}{\log 2.414} \geq 2.626$$

Since $N \geq 2.626$, choose $N = 3$.

**Step 4** Determination of the analog cutoff frequency $\Omega_c$ (i.e., –3 dB frequency)

$$\Omega_c = \frac{\Omega_1}{\left[\frac{1}{A_1^2}-1\right]^{1/2N}} = \frac{2}{\left[\frac{1}{0.9^2}-1\right]^{1/2\times 3}} = 2.5467$$

**Step 5** Determination of the transfer function of the analog Butterworth filter $H_a(s)$

For odd $N$, we have

$$H_a(s) = \frac{\Omega_c}{s+\Omega_c}\prod_{k=1}^{\frac{N-1}{2}}\frac{\Omega_c^2}{s^2 + b_k\Omega_c s + \Omega_c^2}$$

where

$$b_k = 2\sin\left[\frac{(2k-1)\pi}{2N}\right]$$

For $N = 3$, we have

$$H_a(s) = \frac{\Omega_c}{s + \Omega_c} \frac{\Omega_c^2}{s^2 + b_1 \Omega_c s}$$

where

$$b_1 = 2 \sin\left[\frac{(2\times 1 - 1)\pi}{2\times 3}\right] = 2 \sin \frac{\pi}{6} = 1$$

$$H_a(s) = \left(\frac{2.5467}{s + 2.5467}\right)\left(\frac{(2.5467)^2}{s^2 + 1(2.5467)\, s + (2.5467)^2}\right)$$

**Step 6** Conversion of $H_a(s)$ into $H(z)$

Since bilinear transformation is to be used, the digital filter transfer function is

$$H(z) = H_a(s)\Big|_{s=\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)} = H_a(s)\Big|_{s=2\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}$$

$$H(z) = \left(\frac{2.5467}{2\left(\frac{1-z^{-1}}{1+z^{-1}}\right) + 2.5467}\right)\left[\frac{(2.5467)^2}{\left[2\left(\frac{1-z^{-1}}{1+z^{-1}}\right)\right]^2 + 2.5467\left[2\frac{1-z^{-1}}{1+z^{-1}}\right] + (2.5467)^2}\right]$$

$$= \frac{0.2332(1 + z^{-1})^3}{1 + 0.4394 z^{-1} + 0.3845 z^{-2} + 0.0416 z^{-3}}$$

**EXAMPLE 9**

Design a low-pass Butterworth digital filter to give response of 3 dB or less for frequencies upto 2 kHz and an attenuation of 20 dB or more beyond 4 kHz. Use the bilinear transformation technique and obtain $H(z)$ of the desired filter.

***Solution:*** The specifications of the desired filter are given in terms of dB attenuation and frequency in Hz. First the gain is to be expressed as a numerical value and frequency in rad/s.

Here attenuation at passband frequency ($\omega_1$) = 3 dB

Therefore, gain at passband edge frequency ( $\omega_1$) is $k_1 = -3$ dB

$$A_1 = 10^{k_1/20} = 10^{-3/20} = 0.707 = \frac{1}{\sqrt{2}}$$

Attenuation at stopband frequency ($\omega_2$) = 20 dB

Therefore, gain at stopband edge frequency ($\omega_2$) is $k_2 = -20$ dB

$$A_2 = 10^{k_2/20} = 10^{-20/20} = 0.1$$

Passband edge frequency = 2 kHz,

Stopband edge frequency = 4 kHz,

The design is performed as given below.

Let the sampling frequency be 10000 Hz.

$$\text{Normalized } \omega_1 = 2\pi \frac{f_1}{f_s} = 2\pi \frac{2000}{10000} = 0.4$$

$$\text{Normalized } \omega_2 = 2\pi \frac{f_2}{f_s} = 2\pi \frac{4000}{10000} = 0.8$$

**Step 1** Bilinear transformation is chosen

**Step 2** Ratio of analog filter edge frequencies $\Omega_2/\Omega_1$

$$\Omega_1 = \frac{2}{T}\tan\frac{\omega_1}{2} = \frac{2}{T}\tan\frac{0.4\pi}{2} = 14530.8 \text{ rad/s}$$

$$\Omega_2 = \frac{2}{T}\tan\frac{\omega_2}{2} = \frac{2}{T}\tan\frac{0.8\pi}{2} = 61553.6 \text{ rad/s}$$

$$\frac{\Omega_2}{\Omega_1} = \frac{\tan\frac{\omega_2}{2}}{\tan\frac{\omega_1}{2}} = \frac{\tan 0.4\pi}{\tan 0.2\pi} = 4.236$$

**Step 3** Order of the filter

$$N \geq \frac{1}{2}\frac{\log\left\{\left[\frac{1}{A_2^2}-1\right]\Big/\left[\frac{1}{A_1^2}-1\right]\right\}}{\log\frac{\Omega_2}{\Omega_1}}$$

$$\geq \frac{1}{2}\frac{\log\left\{\left[\frac{1}{(0.1)^2}-1\right]\Big/\left[\frac{1}{(1/\sqrt{2})^2}-1\right]\right\}}{\log 4.236}$$

$$\geq \frac{1}{2}\frac{\log[99/1]}{\log 4.236} \geq 1.59$$

$$N = 2$$

**Step 4** Analog cutoff frequency $\Omega_c$

$$\Omega_c = \frac{\Omega_1}{\left[\frac{1}{A_1^2}-1\right]^{1/2N}} = \frac{1.4530}{\left[\frac{1}{(1/\sqrt{2})^2}-1\right]^{1/2\times 2}} = 1.4530$$

Unnormalized

$$\Omega_c = f_s \times 1.4530 = 14530 \text{ rad/s}$$

**Step 5** Transfer function $H_a(s)$

$$\text{For } N = 2,\ H_a(s) = \frac{\Omega_c^2}{s^2 + b_1\Omega_c s + \Omega_c^2}$$

$$\text{where } b_1 = 2\sin\left[\frac{(2\times 1-1)\pi}{2\times 2}\right] = 2\sin\frac{\pi}{4} = 1.414$$

$$\therefore \quad H_a(s) = \frac{(14530)^2}{s^2 + 1.414 \times 14530\, s + (14530)^2}$$

$$= \frac{2.1112 \times 10^8}{s^2 + 20545.42\, s + 2.1112 \times 10^8}$$

**Step 6** Conversion of $H_a(s)$ into $H(z)$

$$H(z) = H_a(s)\Big|_{s=\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)} = H_a(s)\Big|_{s=20000\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}$$

$$H(z) = \frac{2.112 \times 10^8}{\left[20 \times 10^3\left(\frac{1-z^{-1}}{1+z^{-1}}\right)\right]^2 + 2.0545 \times 10^4 \times 20 \times 10^3\left[\frac{1-z^{-1}}{1+z^{-1}}\right] + 2.112 \times 10^8}$$

$$= \frac{0.528}{2.5552 - 0.946z^{-1} + 0.5008z^{-2}}$$

**EXAMPLE 10**

Design a low-pass Butterworth filter using the bilinear transformation method for satisfying the following constraints:

Passband: 0–400 Hz
Passband ripple: 2 dB
Sampling frequency: 10 kHz
Stopband: 2.1– 4 kHz
Stopband attenuation: 20 dB

***Solution:*** Given

$\alpha_1$ = 2 dB, $k_1$ = –2 dB and A1= $10^{k_1/20} = 10^{-2/20} = 0.794$

$\alpha_2$ = 20 dB, $k_2$ = –20 dB $A_2 = 10^{k_2/20} = 10^{-20/20} = 0.1$

**Step 1** Type of transformation
Bilinear transformation is already specified.

**Step 2** Ratio of analog edge frequencies $\Omega_2/\Omega_1$.
Here $f_s$ = 10 kHz
Passband edge frequency $f_1$ = 400 Hz
Stopband edge frequency $f_2$ = 2.1 kHz
Normalizing the frequencies, we have

$$\omega_1 = 2\pi\frac{f_1}{f_s} = 2\pi \times \frac{400}{10000} = 0.25 \text{ rad}$$

$$\omega_2 = 2\pi\frac{f_2}{f_s} = 2\pi \times \frac{2100}{10000} = 1.319 \text{ rad}$$

Therefore, the analog filter edge frequencies are:

$$\Omega_1 = \frac{2}{T}\tan\frac{\omega_1}{2} = 2 \times 10000 \tan\frac{0.25}{2} = 2513.102 \text{ rad/s}$$

and

$$\Omega_2 = \frac{2}{T}\tan\frac{\omega_2}{2} = 2 \times 10000 \tan\frac{1.319}{2} = 15{,}506.08 \text{ rad/s}$$

$$\frac{\Omega_2}{\Omega_1} = \frac{15506.08}{2513.102} = 6.1703$$

**Step 3** Order of the filter $N$

$$N \geq \frac{1}{2} \frac{\log\left\{\left[\frac{1}{A_2^2}-1\right] \Big/ \left[\frac{1}{A_1^2}-1\right]\right\}}{\log\left(\frac{\Omega_2}{\Omega_1}\right)} \quad \text{or} \quad N \geq \frac{1}{2} \frac{\log\left\{\frac{10^{0.1A_2\,\text{dB}}-1}{10^{0.1A_1\,\text{dB}}-1}\right\}}{\log(6.1703)}$$

$$\text{i.e.} \quad N \geq \frac{1}{2} \frac{\log\left\{\left\{\left[\frac{1}{(0.1)^2}-1\right] \Big/ \left[\frac{1}{(0.794)^2}-1\right]\right\}\right\}}{\log(6.1703)} \quad \text{or} \quad N \geq \frac{1}{2} \frac{\log\left\{\frac{10^{0.1\times 20\,\text{dB}}-1}{10^{0.1\times 2\,\text{dB}}-1}\right\}}{\log(6.1703)}$$

$$\text{i.e.} \quad N \geq 1.409 \approx 2 \quad \text{or} \quad N \geq 1.410 \approx 2$$

**Step 4** The cutoff frequency $\Omega_c$

$$\Omega_c = \frac{\Omega_2}{\left[\frac{1}{A_2^2}-1\right]^{1/2N}} \quad \text{or} \quad \Omega_c = \frac{\Omega_2}{[10^{0.1A_2\text{dB}}-1]^{1/2N}}$$

i.e.

$$\Omega_c = \frac{15506.08}{\left[\frac{1}{(0.1)^2}-1\right]^{1/2\times 2}} = 4915.7 \quad \text{or} \quad \Omega_c = \frac{15506.08}{[10^{0.1\times 20}-1]^{1/2\times 2}} = 4915.788 \text{ rad/s}$$

**Step 5** The system function $H_a(s)$

$$H_a(s) = \frac{\Omega_c^2}{s^2 + b_1\Omega_c s + \Omega_c^2} \quad \text{where} \quad b_1 = 2\sin\frac{(2\times 1-1)\pi}{2\times 2} = 1.414$$

$$= \frac{(4915.788)^2}{s^2 + 1.414 \times 4915.788\, s + (4915.788)^2}$$

$$= \frac{2.416 \times 10^7}{s^2 + 6950.92\, s + 2.416 \times 10^7}$$

**Step 6** Digital transfer function $H(z)$

$$H(z) = H_a(s)\Big|_{s=\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}$$

$$= \frac{2.416\times 10^7}{\left[\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)\right]^2 + 6950.92 \times \frac{2}{T}\left[\frac{1-z^{-1}}{1+z^{-1}}\right] + 2.416\times 10^7}$$

$$= \frac{2.416\times 10^7}{\left[20000\times\left(\frac{1-z^{-1}}{1+z^{-1}}\right)\right]^2 + 6950.92\times 20000\left(\frac{1-z^{-1}}{1+z^{-1}}\right) + 2.416\times 10^7}$$

$$= \frac{0.042 + 0.085z^{-1} + 0.042z^{-2}}{1 - 1.335z^{-1} + 0.506z^{-2}}$$

The poles are given by $P_k = \pm(\Omega_c)\left[e^{\frac{j(2k+N+1)\pi}{2N}}\right],\ k = 0, 1 < N$

$$P_0 = \pm(\Omega_c)e^{j\left(\frac{3\pi}{4}\right)} = 4.915.788(-0.707 + j0.707) = -3475.6 + j3475.46$$

$$P_1 = \pm(\Omega_c)e^{j\left(\frac{5\pi}{4}\right)} = -3475.6 - j3475.6$$

**EXAMPLE 11**

A digital low-pass filter is required to meet the following specifications.

| | |
|---|---|
| Passband attenuation ≤ 1 dB | Passband edge = 4 kHz |
| Stopband attenuation ≥ 40 dB | Stopband edge = 8 kHz |
| Sampling rate = 24 kHz | |

The filter is to be designed by performing the bilinear transformation on an analog system function. Design the Butterworth filter.

***Solution:*** Given $\alpha_1$ = 1 dB, $k_1$ = –1 dB and $A_1 = 10^{k1/20} = 10^{-1/20} = 0.8912$

$\alpha_2$ = 40 dB, $k_2$ = – 40 dB and $A_2 = 10^{k_2/20} = 10^{-40/20} = 0.01$

Since $f_s$ = 24 kHz, normalized angular frequencies are:

$$f_1 = 4\,\text{kHz}, \qquad \omega_1 = \frac{2\pi f_1}{f_s} = 2\pi \times \frac{4000}{24000} = 1.047\ \text{rad/s}$$

$$f_2 = 8\,\text{kHz}, \qquad \omega_2 = \frac{2\pi f_2}{f_s} = 2\pi \times \frac{8000}{24000} = 2.094\ \text{rad/s}$$

The Butterworth filter is designed as follows:

**Step 1** Type of transformation
Bilinear transformation is already specified.

**Step 2** Ratio of analog edge frequencies, $\Omega_2/\Omega_1$

$$\Omega_1 = \frac{2}{T}\tan\frac{\omega_1}{2} = 2 \times 24000 \tan\frac{1.047}{2} = 27706.49\,\text{rad/s}$$

$$\Omega_2 = \frac{2}{T}\tan\frac{\omega_2}{2} = 2 \times 24000 \tan\frac{2.094}{2} = 83100.52\,\text{rad/s}$$

$$\therefore \quad \frac{\Omega_2}{\Omega_1} = \frac{83000.52}{27706.49} = 2.9957$$

**Step 3** Order of the filter N

$$N \geq \frac{1}{2}\frac{\log\left\{\left[\frac{1}{A_2^2}-1\right]\Big/\left[\frac{1}{A_1^2}-1\right]\right\}}{\log(\Omega_2/\Omega_1)} \quad \text{or} \quad N \geq \frac{1}{2}\frac{\log\left(\frac{10^{0.1A_2\,\text{dB}}-1}{10^{0.1A_1\,\text{dB}}-1}\right)}{\log(\Omega_2/\Omega_1)}$$

$$\geq \frac{1}{2}\frac{\log\left\{\left[\frac{1}{(0.01)^2}-1\right]\Big/\left[\frac{1}{(0.8912)^2}-1\right]\right\}}{\log(2.9957)} \quad \text{or} \quad N \geq \frac{1}{2}\frac{\log\left(\frac{10^{0.1\times40}-1}{10^{0.1\times1}-1}\right)}{\log(2.9957)}$$

$$\geq \frac{1}{2}\frac{\log\{9999/0.2590\}}{\log(2.9957)} \quad \text{or} \quad N \geq \frac{1}{2}\frac{\log(9999/0.2589)}{\log(2.9957)}$$

$$\geq \frac{1}{2}\frac{4.586}{0.476} \quad \text{or} \quad N \geq \frac{1}{2}\frac{4.586}{0.476}$$

$$\geq 4.8 \approx 5 \quad \text{or} \quad N \geq 4.8 \approx 5$$

**Step 4** The cutoff frequency $\Omega_c$

$$\Omega_c = \frac{\Omega_1}{\left[\frac{1}{A_1^2} - 1\right]^{1/2N}} \quad \text{or} \quad \Omega_c = \frac{\Omega_1}{[10^{0.1A_1\,\text{dB}} - 1]^{1/2N}}$$

$$= \frac{27706.49}{\left[\frac{1}{(0.8912)^2} - 1\right]^{1/2\times5}} \quad \text{or} \quad \Omega_c = \frac{27706.49}{\left[10^{0.1\times1} - 1\right]^{1/2\times5}}$$

$$= 31{,}715 \text{ rad/s} \quad \text{or} \quad \Omega_c = 31{,}715 \text{ rad/s}$$

**Step 5** Analog filter transfer function $H_a(s)$

$$\text{For } N = 5,\ H_a(s) = \left(\frac{\Omega_c}{s+\Omega_c}\right)\left(\frac{\Omega_c^2}{s^2 + b_1\Omega_c s + \Omega_c^2}\right)\left(\frac{\Omega_c^2}{s^2 + b_2\Omega_c s + \Omega_c^2}\right)$$

where

$$b_1 = 2\sin\left(\frac{(2\times1-1)\pi}{2N}\right) = 2\sin\frac{\pi}{10} = 0.618$$

$$b_2 = 2\sin\left(\frac{(2\times2-1)\pi}{10}\right) = 2\sin\frac{3\pi}{10} = 1.618$$

$$\therefore \quad H_a(s) = \left(\frac{31708}{s+31708}\right)\left(\frac{(31708)^2}{s^2 + 0.618\times31708 s + (31708)^2}\right)$$
$$\left(\frac{(31708)^2}{s^2 + 1.618\times31708 s + (31708)^2}\right)$$

**Step 6** Digital filter function $H(z)$

Using the bilinear transformation, we have

$$H(z) = H_a(s)\Big|_{s=\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)} = H_a(s)\Big|_{s=48000\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}$$

$$= \left[\left(\frac{31708}{s+31708}\right)\left(\frac{(31708)^2}{s^2 + 0.618\times31708 s + (31708)^2}\right)\right.$$
$$\left.\left(\frac{(31708)^2}{s^2 + 1.618\times31708 s + (31708)^2}\right)\right]\Big|_{s=48000\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}$$

**EXAMPLE 12**

Design a digital IIR low-pass filter with passband edge at 1000 Hz and stopband edge at 1500 Hz for a sampling frequency of 5000 Hz. The filter is to have a passband ripple of 0.5 dB and a stopband ripple below 30 dB. Design a Butterworth filter using the bilinear transformation.

***Solution:*** Given $f_s$ = 5000 Hz, the normalized frequencies are given as:

$$\text{Passband edge } f_1 = 1000 \text{ Hz}, \quad \therefore \quad \omega_1 = 2\pi\frac{f_1}{f_s} = 2\pi\times\frac{1000}{5000} = 0.4\pi \text{ rad/s}$$

$$\text{Stopband edge } f_2 = 1500 \text{ Hz}, \quad \therefore \quad \omega_2 = 2\pi\frac{f_2}{f_s} = 2\pi\times\frac{1500}{5000} = 0.6\pi \text{ rad/s}$$

$$\alpha_1 = 0.5 \text{ dB}, \quad \therefore k_1 = -0.5 \text{ dB} \quad \text{and} \quad A_1 = 10^{k_1/20} = 10^{-0.5/20} = 0.9446$$

$$\alpha_2 = 30 \text{ dB}, \quad \therefore k_2 = -30 \text{ dB} \quad \text{and} \quad A_2 = 10^{k_2/20} = 10^{-30/20} = 0.0316$$

The Butterworth filter is designed as follows:

**Step 1** Type of transformation.
Bilinear transformation is to be used.

**Step 2** Ratio of analog filter edge frequencies, $\Omega_2/\Omega_1$

$$\Omega_1 = \frac{2}{T}\tan\frac{\omega_1}{2} = 2\times 5000\tan\frac{0.4\pi}{2} = 7265.425 \text{ rad/s}$$

$$\Omega_2 = \frac{2}{T}\tan\frac{\omega_2}{2} = 2\times 5000\tan\frac{0.6\pi}{2} = 13763.819 \text{ rad/s}$$

$$\frac{\Omega_2}{\Omega_1} = \frac{13763.819}{7265.425} = 1.8944$$

**Step 3** Order of the filter $N$

$$N \geq \frac{1}{2}\frac{\log\left\{\left[\frac{1}{A_2^2}-1\right]\Big/\left[\frac{1}{A_1^2}-1\right]\right\}}{\log(\Omega_2/\Omega_1)}$$

$$\geq \frac{1}{2}\frac{\log\left\{\left[\frac{1}{(0.0316)^2}-1\right]\Big/\left[\frac{1}{(0.9446)^2}-1\right]\right\}}{\log(1.844)}$$

$$\geq \frac{1}{2}\frac{\log\{1000.44/0.1207\}}{\log(1.844)}$$

$$\geq 7.35 \approx 8$$

**Step 4** The cutoff frequency

$$\Omega_c = \frac{\Omega_1}{\left[\frac{1}{A_1^2}-1\right]^{1/2N}} = \frac{7265.425}{\left[\frac{1}{0.9446^2}-1\right]^{1/2\times 8}} = 8292 \text{ rad/s}$$

**Step 5** The system function $H_a(s)$

$$\therefore \quad b_1 = 2\sin\left[\frac{\pi}{16}\right] = 0.390 \qquad b_2 = 2\sin\left[\frac{3\pi}{16}\right] = 1.111$$

$$b_3 = 2\sin\left[\frac{5\pi}{16}\right] = 1.662 \qquad b_4 = 2\sin\left[\frac{7\pi}{16}\right] = 1.961$$

$$H_a(s) = \left(\frac{(8292)^2}{s^2+0.39\times 8292\,s+(8292)^2}\right)\left(\frac{(8292)^2}{s^2+1.111\times 8292\,s+(8292)^2}\right)$$
$$\left(\frac{(8292)^2}{s^2+1.662\times 8292\,s+(8292)^2}\right)\left(\frac{(8292)^2}{s^2+1.961\times 8292\,s+(8292)^2}\right)$$

$$H_a(s) = \prod_{k=1}^{N/2}\frac{\Omega_c^2}{s^2+b_k\Omega_c s+\Omega_c^2}$$

where $b_k = 2\sin\left[\frac{(2k-1)\pi}{2N}\right]$

**Step 6** Digital filter function $H(z)$
Using the bilinear transformation, we have

$$H(z) = H_a(s)\Big|_{s=\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)} = H_a(s)\Big|_{s=10000\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}$$

$$H(z) = \left[\frac{\left(\frac{(8292)^2}{s^2+3233.8s+(8292)^2}\right)\left(\frac{(8292)^2}{s^2+9212.4s+(8292)^2}\right)}{\left(\frac{(8292)^2}{s^2+13781.3s+(8292)^2}\right)\left(\frac{(8292)^2}{s^2+16260.6s+(8292)^2}\right)}\right]_{s=10000\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}$$

**EXAMPLE 13**

Determine the order of a Butterworth low-pass filter satisfying the following specifications:

$f_p = 0.10$ Hz, $\alpha_p = 0.5$ dB

$f_s = 0.15$ Hz, $\alpha_s = 15$ dB; $f = 1$ Hz

***Solution:*** Given

$f_p = 0.10$ Hz, $\omega_p = \omega_1 = 2\pi f_p = 2\pi(0.1) = 0.2\pi$

$f_s = 0.15$ Hz, $\omega_s = \omega_2 = 2\pi f_s = 2\pi(0.15) = 0.30\pi$

$\alpha_p = \alpha_1 = 0.5$ dB, $k_1 = -0.5$ dB, so $A_1 = 10^{k_1/20} = 10^{-0.5/20} = 0.944$

$\alpha_s = \alpha_2 = 15$ dB, $k_2 = -15$ dB, so $A_2 = 10^{k_2/20} = 10^{-15/20} = 0.177$

$f = 1$ Hz, $\therefore T = \frac{1}{f} = \frac{1}{1} = 1$s.

1. The type of transformation is not specified. Let us use bilinear transformation.

2. $$\frac{\Omega_2}{\Omega_1} = \frac{\frac{2}{T}\tan\frac{\omega_2}{2}}{\frac{2}{T}\tan\frac{\omega_1}{2}} = \frac{\frac{2}{1}\tan\frac{0.3\pi}{2}}{\frac{2}{1}\tan\frac{0.2\pi}{2}} = \frac{1.019}{0.649} = 1.57$$

$$N \geq \frac{1}{2}\frac{\log\left\{\left[\frac{1}{A_2^2}-1\right]\Big/\left[\frac{1}{A_1^2}-1\right]\right\}}{\log\left(\frac{\Omega_2}{\Omega_1}\right)}$$

3. $$\geq \frac{1}{2}\frac{\log\left\{\left[\frac{1}{0.177^2}-1\right]\Big/\left[\frac{1}{0.944^2}-1\right]\right\}}{\log(1.57)}$$

$\geq 6.16 \approx 7$

So the order of the low-pass Butterworth filter is $N = 7$.

## DESIGN OF LOW-PASS CHEBYSHEV FILTER

For designing a Chebyshev IIR digital filter, first an analog filter is designed using the given specifications. Then the analog filter transfer function is transformed to digital filter transfer function by using either impulse invariant transformation or bilinear transformation.

The analog Chebyshev filter is designed by approximating the ideal frequency response using an error function. There are two types of Chebyshev approximations.

In type-1 approximation, the error function is selected such that the magnitude response is equiripple in the passband and monotonic in the stopband.

In type-2 approximation, the error function is selected such that the magnitude function is monotonic in the passband and equiripple in the stopband. The type-2 magnitude response is also called inverse Chebyshev response. The type-1 design is discussed.

The magnitude response of type-1 Chebyshev low-pass filter is given by

$$|H_a(\Omega)|^2 = \frac{1}{1+\varepsilon^2 c_N^2\left(\frac{\Omega}{\Omega_c}\right)}$$

Where ε is attenuation constant given by

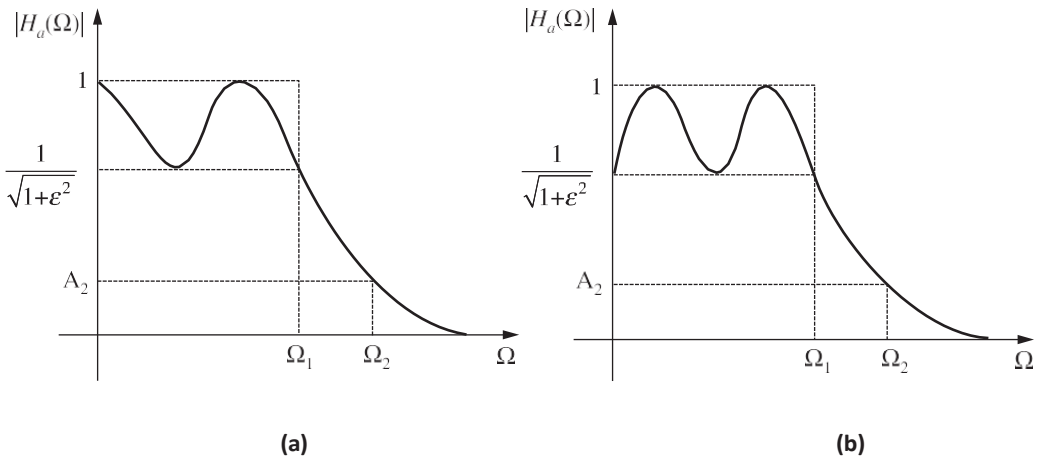
$$\varepsilon = \left[\frac{1}{A_1^2} - 1\right]^{\frac{1}{2}}$$

$A_1$ is the gain at the passband edge frequency ω1 and Chebyshev polynomial of the first kind of degree N given by

$$c_N(x) = \{\cos(N\cos^{-1}x), \quad \text{for } x \le 1$$
$$= \{\cos(N\cosh^{-1}x), \quad \text{for } x \ge 1$$

and $\Omega_c$ is the 3 dB cutoff frequency.

The frequency response of Chebyshev filter depends on order *N*. The approximated response approaches the ideal response as the order *N* increases. The phase response of the Chebyshev filter is more nonlinear than that of the Butterworth filter for a given filter length *N*. The magnitude response of type-1 Chebyshev filter is shown in Figure 6.




**(a)** **(b)**

**Figure 6 M**agnitude response of type–I Chebyshev filter.

The design parameters of the Chebyshev filter are obtained by considering the low-pass filter with the desired specifications as given below.

$$A_1 \le |H(\omega)| \le 1 \qquad 0 \le \omega \le \omega_1$$
$$|H(\omega)| \le A_2 \qquad \omega_2 \le \omega \le \pi$$

The corresponding analog magnitude response is to be obtained in the design process. We have

$$A_1^2 \le \frac{1}{1+\varepsilon^2 c_N^2(\Omega_1/\Omega_2)} \le 1$$

$$\frac{1}{1+\varepsilon^2 c_N^2(\Omega_1/\Omega_2)} \le A_2^2$$

Assuming $\Omega_c = \Omega_1$, we will have $c_N(\Omega_1/\Omega_c) = c_N(1) = 1$.

. Therefore, from the above inequality involving $A^2$ , we get

$$A_1^2 \le \frac{1}{1+\varepsilon^2}$$

Assuming equality in the above equation, the expression for ε is

$$\varepsilon = \left[\frac{1}{A_1^2} - 1\right]^{\frac{1}{2}}$$

The order of the analog filter, *N* can be determined from the inequality for $A_2^2$

Assuming $\Omega_c = \Omega_1$,

$$c_N(\Omega_2/\Omega_1) \ge \frac{1}{\varepsilon}\left[\frac{1}{A_2^2} - 1\right]^{\frac{1}{2}}$$

Since $\Omega_2 > \Omega_1$,

$$\cosh[N\cosh^{-1}(\Omega_2/\Omega_1)] \ge \frac{1}{\varepsilon}\left[\frac{1}{A_2^2} - 1\right]^{\frac{1}{2}}$$

or

$$N \ge \frac{\cosh^{-1}\left\{\frac{1}{\varepsilon}\left[\frac{1}{A_2^2} - 1\right]^{\frac{1}{2}}\right\}}{\cosh^{-1}(\Omega_2/\Omega_1)}$$

Choose *N* to be the next nearest integer to the value given above. The values of $\Omega_2$ and $\Omega_1$ are determined from $\omega_1$ and $\omega_2$ using either impulse invariant transformation or bilinear transformation.

The transfer function of Chebyshev filters are usually written in the factored form as given below.

When *N* is even,

$$H_a(s) = \frac{B_0\Omega_c}{s+\Omega_c}\prod_{k=1}^{\frac{N-1}{2}}\frac{B_k\Omega_c^2}{s^2 + b_k\Omega_c s + c_k\Omega_c^2}$$

When *N* is odd, where

$$H_a(s) = \prod_{k=1}^{\frac{N}{2}}\frac{B_k\Omega_c^2}{s^2 + b_k\Omega_c s + c_k\Omega_c^2}$$

$$b_k = 2\,y_N \sin\left(\frac{(2k-1)\pi}{2N}\right)$$

$$c_k = y_N^2 + \cos^2\left(\frac{(2k-1)\pi}{2N}\right)$$

$$c_0 = y_N$$

$$y_N = \frac{1}{2}\left\{\left[\left(\frac{1}{\varepsilon^2}+1\right)^{\frac{1}{2}} + \frac{1}{\varepsilon}\right]^{\frac{1}{N}} - \left[\left(\frac{1}{\varepsilon^2}+1\right)^{\frac{1}{2}} + \frac{1}{\varepsilon}\right]^{\frac{-1}{N}}\right\}$$

For even values of $N$ and unity dc gain filter, the parameter $B_k$ are evaluated using the equation:

$$H_a(s)\Big|_{s=0} = \frac{1}{[1+\varepsilon^2]^{1/2}}$$

For odd values of $N$ and unity dc gain filter, the parameter $B_k$ are evaluated using the equation:

$$H_a(s)\Big|_{s=0} = 1$$

### *Poles of a NORMALIZED Chebyshev filter*

The transfer function of the analog system can be obtained from the equation for the magnitude squared response as:

$$H_a(s)H_a(-s) = \frac{1}{1+\varepsilon^2 c_N^2\left(\frac{s/j}{\Omega_c}\right)}$$

For the normalized transfer function, let us replace $s/\Omega_c$ by $s_n$.

$$H_a(s_n)H_a(-s_n) = \frac{1}{1+\varepsilon^2 c_N^2(-js_n)}$$

The normalized poles in the $s$-domain can be obtained by equating the denominator of the above equation to zero, i.e., $1+\varepsilon^2 c_N^2(-js_n)$ to zero.

The

The solution to the above expression gives us the $2N$ poles of the filter given by

$$s_n = -\sin x \sinh y + j\cos x \cosh y = \sigma_n + j\Omega_n$$

where $n = 1, 2, \ldots, (N+1)/2$ for $N$ odd
$= 1, 2, \ldots, N/2$ for $N$ even

And $$x = \frac{(2n-1)\pi}{2N} \qquad n = 1,2, \ldots, N$$

$$y = \pm\frac{1}{N}\sinh^{-1}\left(\frac{1}{\varepsilon}\right) \qquad n = 1, 2, \ldots, N$$

The unnormalized poles, $s'_n$ can be obtained from the normalized poles as shown below.

$$s'_n = s_n \Omega_c$$

The normalized poles lie on an ellipse in *s*-plane. Since for a stable filter all the poles should lie in the left half of *s*-plane, only the *N* poles on the ellipse which are in the left half of *s*-plane are considered.

For *N* even, all the poles are complex and exist in conjugate pairs. For *N* odd, one pole is real and all other poles are complex and occur in conjugate pairs.

### *Design procedure for low-pass digital Chebyshev IIR filter*

The low-pass Chebyshev IIR digital filter is designed following the steps given below.

**Step 1** Choose the type of transformation.
(Bilinear or impulse invariant transformation)

**Step 2** Calculate the attenuation constant .

$$\varepsilon = \left[\frac{1}{A_1^2} - 1\right]^{\frac{1}{2}}$$

**Step 3** Calculate the ratio of analog edge frequencies $\Omega_2/\Omega_1$. .
For bilinear transformation,

$$\frac{\Omega_2}{\Omega_1} = \frac{\frac{2}{T}\tan\frac{\omega_2}{2}}{\frac{2}{T}\tan\frac{\omega_1}{2}} = \frac{\tan\frac{\omega_2}{2}}{\tan\frac{\omega_1}{2}}$$

**Step 4** Decide the order of the filter *N* such that

$$N \geq \frac{\cosh^{-1}\left\{\frac{1}{\varepsilon}\left[\frac{1}{A_2^2} - 1\right]\right\}}{\cosh^{-1}\left\{\frac{\Omega_2}{\Omega_1}\right\}}$$

**Step 5** Calculate the analog cutoff frequency $\Omega_c$.
For bilinear transformation,

$$\Omega_c = \frac{\Omega_1}{\left[\frac{1}{A_1^2} - 1\right]^{1/2N}} = \frac{\frac{2}{T}\tan\frac{\omega_1}{2}}{\left[\frac{1}{A_1^2} - 1\right]^{1/2N}}$$

**Step 6** Determine the analog transfer function $H_a(s)$ of the filter.
When the order *N* is even, $H_a(s)$ is given by

$$H_a(s) = \prod_{k=1}^{N/2} \frac{B_k \Omega_c^2}{s^2 + b_k \Omega_c s + c_k \Omega_c^2}$$

When the order *N* is odd, $H_a(s)$ is given by

$$H_a(s) = \frac{B_0 \Omega_c}{s + c_0 \Omega_c} \prod_{k=1}^{\frac{N-1}{2}} \frac{B_k \Omega_c^2}{s^2 + b_k \Omega_c s + c_k \Omega_c^2}$$

where

$$b_k = 2y_N \sin\left(\frac{(2k-1)\pi}{2N}\right)$$

$$c_k = y_N^2 + \cos^2 \frac{(2k-1)\pi}{2N}$$

$$c_0 = y_N$$

$$y_N = \frac{1}{2}\left\{\left[\left(\frac{1}{\varepsilon^2}+1\right)^{\frac{1}{2}} + \frac{1}{\varepsilon}\right]^{\frac{1}{N}} - \left[\left(\frac{1}{\varepsilon^2}+1\right)^{\frac{1}{2}} + \frac{1}{\varepsilon}\right]^{\frac{-1}{N}}\right\}$$

For even values of $N$ and unity dc gain filter, find $B_k$'s such that

$$H_a(0) = \frac{1}{(1+\varepsilon^2)^{1/2}}$$

For odd values of $N$ and unity dc gain filter, find $B_k$'s such that

$$\prod_{k=0}^{\frac{N-1}{2}} \frac{B_k}{c_k} = 1$$

(It is normal practice to take $B_0 = B_1 = B_2 = ... = B_k$)

**Step 7** Using the chosen transformation, transform $H_a(s)$ to $H(z)$, where $H(z)$ is the transfer function of the digital filter.

[The high-pass, band pass and band stop filters are obtained from low-pass filter design by frequency transformation].

### *Properties of Chebyshev filters* (*Type* 1)

1. The magnitude response is equiripple in the passband and monotonic inmn the stopband.
2. The chebyshev type-1 filters are all pole designs.
3. The normalized magnitude function has a value of $1/\sqrt{1+\varepsilon^2}$ at the cutoff frequency $\Omega_c$.
4. The magnitude response approaches the ideal response as the value of $N$ increases.

**EXAMPLE 14**

Design a Chebyshev IIR digital low-pass filter to satisfy the constraints.

$$0.707 \le |H(\omega)| \le 1, \qquad 0 \le \omega \le 0.2\pi$$

$$|H(\omega)| \leq 0.1, \qquad 0.5\pi \leq \omega \leq \pi$$

using bilinear transformation and assuming $T = 1$ s.

***Solution:*** Given

$$A_1 = 0.707, \qquad \omega_1 = 0.2\pi$$

$$A_2 = 0.1, \qquad \omega_2 = 0.5\pi$$

$T = 1$ s and bilinear transformation is to be used. The low-pass Chebyshev IIR digital filter is designed as follows:

**Step 1** Type of transformation
Here bilinear transformation is to be used.

**Step 2** Attenuation constant $\varepsilon$.

$$\varepsilon = \left[\frac{1}{A_1^2} - 1\right]^{\frac{1}{2}} = \left[\frac{1}{0.707^2} - 1\right]^{\frac{1}{2}} = 1$$

**Step 3** Ratio of analog edge frequencies, $\Omega_2/\Omega_1$.
Since bilinear transformation is to be used,

$$\frac{\Omega_2}{\Omega_1} = \frac{\frac{2}{T}\tan\frac{\omega_2}{2}}{\frac{2}{T}\tan\frac{\omega_1}{2}} = \frac{\tan\frac{0.5\pi}{2}}{\tan\frac{0.2\pi}{2}} = \frac{2}{0.6498} = 3.0779$$

**Step 4** Order of the filter $N$

$$N \geq \frac{\cosh^{-1}\left\{\frac{1}{\varepsilon}\left[\frac{1}{A_2^2} - 1\right]^{\frac{1}{2}}\right\}}{\cosh^{-1}\left\{\frac{\Omega_2}{\Omega_1}\right\}} \geq \frac{\cosh^{-1}\left\{\frac{1}{1}\left[\frac{1}{0.1^2} - 1\right]^{0.5}\right\}}{\cosh^{-1}\{3.0779\}} \geq 1.669 \approx 2.$$

**Step 5** Analog cutoff frequency $\Omega_c$

$$\Omega_c = \frac{\Omega_1}{\left[\frac{1}{A_1^2} - 1\right]^{1/2N}} = \frac{\frac{2}{T}\tan\frac{\omega_1}{2}}{\left[\frac{1}{A_1^2} - 1\right]^{1/2N}} = \frac{0.6498}{\left[\frac{1}{0.7077} - 1\right]^{\frac{1}{4}}} = 0.6498$$

**Step 6** Analog filter transfer function $H_a(s)$

$$H_a(s) = \prod_{k=1}^{N/2} \frac{B_k \Omega_c^2}{s^2 + b_k \Omega_c s + c_k \Omega_c^2} = \frac{B_1 \Omega_c^2}{s^2 + b_1 \Omega_c s + c_1 \Omega_c^2}$$

$$y_N = \frac{1}{2}\left\{\left[\left(\frac{1}{\varepsilon^2}+1\right)^{\frac{1}{2}} + \frac{1}{\varepsilon}\right]^{\frac{1}{N}} - \left[\left(\frac{1}{\varepsilon^2}+1\right)^{\frac{1}{2}} + \frac{1}{\varepsilon}\right]^{\frac{-1}{N}}\right\}$$

$$= \frac{1}{2}\left\{\left[\left(\frac{1}{1^2}+1\right)^{\frac{1}{2}} + \frac{1}{1}\right]^{\frac{1}{2}} - \left[\left(\frac{1}{1^2}+1\right)^{\frac{1}{2}} + \frac{1}{1}\right]^{\frac{-1}{2}}\right\}$$

$$= \frac{1}{2}\left\{[2.414]^{\frac{1}{2}} - [2.414]^{\frac{-1}{2}}\right\} = 0.455$$

$$b_1 = 2y_N \sin\left[\frac{(2k-1)\pi}{2N}\right] = 2 \times 0.455 \ \sin\left[\frac{(2\times 1-1)\pi}{2\times 2}\right] = 0.6435$$

$$c_1 = y_N^2 + \cos^2\left[\frac{(2k-1)\pi}{2N}\right] = (0.455)^2 \ + \cos^2\left[\frac{(2\times 1-1)\pi}{2\times 2}\right] = 0.707$$

For $N$ even,

$$\prod_{k=1}^{\frac{N}{2}} \frac{B_k}{c_k} = \frac{A}{(1+\varepsilon^2)^{0.5}} = 0.707$$

That is $B_1 = c_1$ x 0.707 = 0.707 x 0.707 = 0.5.
Therefore, the system function is:

$$H_a(s) = \frac{0.5(0.6498)^2}{s^2 + (0.6435)(0.6498)s + (0.707)(0.6498)^2}$$

On simplifying, we get

$$H_a(s) = \frac{0.2111}{s^2 + 0.4181s + 0.2985}$$

**Step 7** Digital filter transfer function $H(z)$

$$H(z) = H_a(s)\Big|_{s=\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}} = \frac{0.2111}{s^2 + 0.4181s + 0.2985}\Bigg|_{s=2\frac{1-z^{-1}}{1+z^{-1}}}$$

$$= \frac{0.2111}{\left[2\left(\frac{1-z^{-1}}{1+z^{-1}}\right)\right]^2 + 0.4181\left[2\left(\frac{1-z^{-1}}{1+z^{-1}}\right)\right] + 0.2985}$$

$$= \frac{0.2111(1+z^{-1})^2}{5.1347 - 7.403z^{-1} + 3.463z^{-2}}$$

$$= \frac{0.0411(1+z^{-1})^2}{1 - 1.441z^{-1} + 0.6744z^{-2}}$$

**EXAMPLE 15**

Determine the system function $H(z)$ of the lowest order Chebyshev IIR digital filter with the following specifications:

3 dB ripple in passband $0 \le \omega \le 0.2\pi$

25 dB attenuation in stopband $0.45\pi \le \omega \le \pi$

***Solution:*** Given

$\alpha_1 = 3$ dB, $k_1 = -3$dB and hence $A_1 = 10^{k_1/20} = 10^{-3/20} = 0.707$

$\alpha_2 = 25$ dB, $k_2 = -25$dB and hence $A_2 = 10^{k_2/20} = 10^{-25/20} = 0.0562$

$\omega_1 = 0.2\pi$ and $\omega_2 = 0.45\pi$

Let $T = 1$ and bilinear transformation is used

Attenuation constant

$$\varepsilon = \left[\frac{1}{A_1^2} - 1\right]^{\frac{1}{2}} = \left[\frac{1}{0.707^2} - 1\right] = 1$$

Ratio of analog frequencies

$$\frac{\Omega_2}{\Omega_1} = \frac{\frac{2}{T}\tan\frac{\omega_2}{2}}{\frac{2}{T}\tan\frac{\omega_1}{2}} = \frac{\tan\frac{0.45\pi}{2}}{\tan\frac{0.2\pi}{2}} = 2.628$$

Order of filter

$$N \ge \frac{\cosh^{-1}\left\{\frac{1}{\varepsilon}\left[\frac{1}{A_2^2} - 1\right]^{\frac{1}{2}}\right\}}{\cosh^{-1}\left\{\frac{\Omega_2}{\Omega_1}\right\}}$$

$$\ge \frac{\cosh^{-1}\left\{\frac{1}{1}\left[\frac{1}{0.0562^2} - 1\right]^{\frac{1}{2}}\right\}}{\cosh^{-1}(2.628)}$$

$$\ge \frac{3.569}{1.621} \ge 2.20 \approx 3$$

Analog cutoff frequency

$$\Omega_c = \frac{\Omega_1}{\left[\frac{1}{A_1^2} - 1\right]^{1/2N}} = \frac{\frac{2}{T}\tan\frac{\omega_1}{2}}{\left[\frac{1}{0.707^2} - 1\right]^{1/6}} = 1.708$$

Analog filter transfer function for $N = 3$.

$$H_a(s) = \frac{B_0\Omega_c}{s + c_0\Omega_c} \frac{B_1\Omega_c^2}{s^2 + b_1\Omega_c s + c_1\Omega_c^2}$$

$$y_N = \frac{1}{2}\left\{\left[\left(\frac{1}{\varepsilon^2} + 1\right)^{\frac{1}{2}} + \frac{1}{\varepsilon}\right]^{\frac{1}{N}} - \left[\left(\frac{1}{\varepsilon^2} + 1\right)^{\frac{1}{2}} + \frac{1}{\varepsilon}\right]^{\frac{-1}{N}}\right\}$$

$$y_N = \frac{1}{2}\left\{\left[\left(\frac{1}{1^2} + 1\right)^{\frac{1}{2}} + \frac{1}{1}\right]^{\frac{1}{3}} - \left[\left(\frac{1}{1^2} + 1\right)^{\frac{1}{2}} + \frac{1}{1}\right]^{\frac{-1}{3}}\right\} = 0.5959$$

$$c_0 = y_N = 0.5959$$

$$b_1 = 2y_N \sin\left[\frac{(2\times 1 - 1)\,\pi}{2N}\right] = 2 \times 0.5959 \sin\frac{\pi}{6} = 0.5959$$

$$c_1 = y_N^2 + \cos^2\frac{(2\times 1 - 1)\pi}{2N} = 0.5959^2 + \cos^2\frac{\pi}{6} = 1.105$$

*For N* odd

$$\prod_{k=0}^{(N-1)/2} \frac{B_k}{c_k} = 1$$

$$\therefore \quad B_0 = c_0 = 0.5959, \quad B_1 = c_1 = 1.105$$

$$H_a(s) = \left(\frac{0.5959 \times 1.708}{s + 0.5959 \times 1.708}\right)\left(\frac{1.105(1.708)^2}{s^2 + 0.5959 \times 1.708 s + 1.105(1.708)^2}\right)$$

$$= \left(\frac{1.01}{s + 1.01}\right)\left(\frac{3.223}{s^2 + 1.01 s + 3.223}\right)$$

Using bilinear transformation, $H(z)$ is given by

$$H(z) = H_a(s)\Big|_{s=\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}} = \left(\frac{1.01}{s+1.01}\right)\left(\frac{3.223}{s^2 + 0.1s + 3.223}\right)\Bigg|_{s=2\,\frac{1-z^{-1}}{1+z^{-1}}}$$

$$= \frac{3.25}{\left[2\left(\frac{1-z^{-1}}{1+z^{-1}}\right) + 1.01\right]\left[2\left(\frac{1-z^{-1}}{1+z^{-1}}\right)^2 + 0.1 \times 2\left(\frac{1-z^{-1}}{1+z^{-1}}\right) + 3.223\right]}$$

$$= \frac{(3.25)\,(1 + z^{-1})^3}{7.423 - 1.554\,z^{-1} + 7.023\,z^{-2}}$$

**EXAMPLE 16**

The specification of the desired low-pass filter is:

$$0.9 \le |H(\omega)| \le 1.0; \qquad 0 \le \omega \le 0.3\ \pi$$

$$|H(\omega)| \le 0.15; \qquad 0.5\pi \le \omega \le \pi$$

Design a Chebyshev digital filter using the bilinear transformation.

***Solution:*** Given

$$A_1 = 0.9, \quad \omega_1 = 0.3\,\pi$$

$$A_2 = 0.15, \quad \omega_2 = 0.5\,\pi$$

The Chebyshev filter is designed as per the following steps:

**Step 1** The bilinear transformation is used.

**Step 2** Attenuation constant $\varepsilon$

$$\varepsilon = \left[\frac{1}{A_1^2} - 1\right]^{1/2} = \left[\frac{1}{(0.9)^2} - 1\right]^{1/2} = 0.484$$

**Step 3** Ratio of analog edge frequencies $\Omega_2/\Omega_1$

$$\frac{\Omega_2}{\Omega_1} = \frac{\frac{2}{T}\tan\frac{\omega_2}{2}}{\frac{2}{T}\tan\frac{\omega_1}{2}} = \frac{\tan 0.25\pi}{\tan 0.15\pi} = 1.962$$

**Step 4** Order of the filter $N$

$$N \geq \frac{\cosh^{-1}\left[\frac{1}{\varepsilon}\left(\frac{1}{A_2^2} - 1\right)^{\frac{1}{2}}\right]}{\cosh^{-1}\left(\frac{\Omega_2}{\Omega_1}\right)} \geq \frac{\cosh^{-1}\left[\frac{1}{0.484}\left(\frac{1}{0.15^2} - 1\right)^{\frac{1}{2}}\right]}{\cosh^{-1} 1.962}$$

$$\geq \frac{\cosh^{-1} 13.618}{\cosh^{-1} 1.962} \geq 2.55 = 3$$

So order of the filter is $N = 3$. Let $T = 1$ s.

**Step 5** Analog cutoff frequency $\Omega_c$

$$\Omega_c = \frac{\Omega_1}{\left[\frac{1}{A_1^2} - 1\right]^{\frac{1}{2N}}} = \frac{\frac{2}{T}\tan\frac{\omega_1}{2}}{\left[\frac{1}{A_1^2} - 1\right]^{\frac{1}{2N}}} = \frac{1.019}{\left[\frac{1}{0.92} - 1\right]^{1/6}} = 1.13 \text{ rad/s}$$

**Step 6** Analog transfer function $H_a(s)$

For $N = 3$, $H_a(s) = \frac{B_0\Omega_c}{s + c_0\Omega_c} \quad \frac{B_1\Omega_c^2}{s^2 + b_1\Omega_c s + c_1\Omega_c^2}$

$$y_N = \frac{1}{2}\left\{\left[\left(\frac{1}{\varepsilon^2} + 1\right)^{\frac{1}{2}} + \frac{1}{\varepsilon}\right]^{\frac{1}{N}} - \left[\left(\frac{1}{\varepsilon^2} + 1\right)^{\frac{1}{2}} + \frac{1}{\varepsilon}\right]^{\frac{-1}{N}}\right\}$$

$$= \frac{1}{2}\left\{\left[\left(\frac{1}{(0.484)^2} + 1\right)^{\frac{1}{2}} + \frac{1}{0.484}\right]^{\frac{1}{3}} - \left[\left(\frac{1}{(0.484)^2} + 1\right)^{\frac{1}{2}} + \frac{1}{0.484}\right]^{\frac{-1}{3}}\right\}$$

$$= \frac{1}{2}\{1.634 - 0.612\} = 0.511$$

$$c_0 = y_N = 0.511$$

$$c_k = y_N^2 + \cos^2\frac{(2k-1)\pi}{2N}$$

When $k = 1$,

$$c_1 = y_N^2 + \cos^2\left(\frac{\pi}{6}\right) = (0.511)^2 + 0.75 = 1.011$$

$$b_k = 2y_N \sin\frac{(2k-1)\pi}{2N}$$

When $k = 1$, $b_1 = y_N + \sin\left(\frac{\pi}{6}\right) = 2 \times 0.511\left(\frac{1}{2}\right) = 0.511$

$$\therefore \quad H_a(s) = \left(\frac{B_0(1.13)}{s + 0.511 \times 1.13}\right)\left(\frac{B_1(1.13)^2}{s^2 + 0.511 \times 1.13s + 1.011(1.13)^2}\right)$$

When $s = 0$

$$H_a(s) = H_a(0) = \frac{B_0 B_1 (1.442)}{(0.511)(1.13)(1.011)(1.13)^2} = 1.935\, B_0 B_1$$

,

$$\therefore \quad H_a(s) = \left(\frac{B_0(1.13)}{s + 0.511 \times 1.13}\right)\left(\frac{B_1(1.13)^2}{s^2 + 0.511 \times 1.13s + 1.011(1.13)^2}\right)$$

Let $H_a(0) = 1$, $\quad 1.935\, B_0 B_1 = 1$

Let $B_0 = B_1$, $\quad B_0^2 = \dfrac{1}{1.935} = 0.516$ or $\quad B_0 = 0.718$

$$B_0 = B_1 = 0.86$$

$$H_a(s) = \frac{0.516\ (1.442)}{(s + 0.577)\ (s^2 + 0.577s + 1.29)}$$

$$= \frac{0.744}{(s + 0.577)\ (s^2 + 0.577s + 1.29)}$$

**Step 7** Digital transfer function

$$H(z) = H_a(s)\Big|_{s=\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}} = H_a(s)\Big|_{s=2\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}$$

$$= \frac{0.744}{(s + 0.577)(s^2 + 0.577s + 1.29)}\Bigg|_{s=2\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}$$

$$= \frac{0.744}{\left(2\dfrac{1-z^{-1}}{1+z^{-1}} + 0.577\right)\left(\left(2\dfrac{1-z^{-1}}{1+z^{-1}}\right)^2 + 0.577 \times 2\dfrac{1-z^{-1}}{1+z^{-1}} + 1.29\right)}$$

$$= \frac{0.744(1 + z^{-1})^3}{(2.577 - 1.423z^{-1})(6.83 - 5.42z^{-1} + 3.75)}$$

**EXAMPLE 17**

Determine the system function of the lowest order Chebyshev digital filter that meets the following specifications.

2 dB ripple in the passband $0 \leq \omega \leq 0.25\,\pi$

Atleast 50 dB attenuation in stopband $0.4\pi \leq \omega \leq \pi$

***Solution:*** Given

Ripple in passband = 2 dB, i.e. $k_1 = -2$ dB $\quad A_1 = 10^{k1/20} = 10^{-2/20} = 0.794$

Attenuation in stopband = 50 dB, i.e. $k_2 = -50$ dB $\quad A_2 = 10^{k2/20} = 10^{-50/20} = 0.0031$

$$A_1 = 0.794, \quad \omega_1 = 0.25\pi$$
$$A_2 = 0.003, \quad \omega_2 = 0.4\pi$$

The Chebyshev filter is designed as per the following steps:

**Step 1** Type of transformation

Let us choose bilinear transformation.

**Step 2** Attenuation constant $\varepsilon$

$$\varepsilon = \left[\frac{1}{A_1^2} - 1\right]^{1/2} = \left[\frac{1}{0.794^2} - 1\right]^{1/2} = 0.765$$

**Step 3** Ratio of analog edge frequencies, $\Omega_2/\Omega_1$

$$\frac{\Omega_2}{\Omega_1} = \frac{\frac{2}{T}\tan\frac{\omega_2}{2}}{\frac{2}{T}\tan\frac{\omega_1}{2}} = \frac{\tan 0.4\pi/2}{\tan 0.25\pi/2} = \frac{1.453}{0.828} = 1.754$$

**Step 4** Order of the filter $N$

$$N \geq \frac{\cosh^{-1}\left[\frac{1}{\varepsilon}\left(\frac{1}{A_2^2} - 1\right)^{\frac{1}{2}}\right]}{\cosh^{-1}\left(\frac{\Omega_2}{\Omega_1}\right)}$$

$$\geq \frac{\cosh^{-1}\left[\frac{1}{0.765}\left(\frac{1}{(0.0031)^2} - 1\right)^{\frac{1}{2}}\right]}{\cosh^{-1} 1.754}$$

$$\geq \frac{6.718}{1.161} \geq 5.786 \approx 6$$

$$N = 6$$

**Step 5** Analog cutoff frequency $\Omega_c$

$$\Omega_c = \frac{\Omega_1}{\left[\frac{1}{A_1^2} - 1\right]^{\frac{1}{2N}}} = \frac{\frac{2}{T}\tan\frac{\omega_1}{2}}{\left[\frac{1}{A_1^2} - 1\right]^{\frac{1}{2N}}} = \frac{0.828}{\left[\frac{1}{0.794^2} - 1\right]^{1/12}} = 0.866 \text{ rad/s}$$

**Step 6** Analog transfer function $H_a(s)$

$$\text{For } N=6,\ H_a(s)=\left(\frac{B_1\Omega_c^2}{s^2+b_1\Omega_c s+c_1\Omega_c^2}\right)\left(\frac{B_2\Omega_c^2}{s^2+b_2\Omega_c s+c_2\Omega_c^2}\right)\left(\frac{B_3\Omega_c^2}{s^2+b_3\Omega_c s+c_3\Omega_c^2}\right)$$

$$y_N=\frac{1}{2}\left\{\left[\left(\frac{1}{\varepsilon^2}+1\right)^{\frac{1}{2}}+\frac{1}{\varepsilon}\right]^{\frac{1}{N}}-\left[\left(\frac{1}{\varepsilon^2}+1\right)^{\frac{1}{2}}+\frac{1}{\varepsilon}\right]^{\frac{-1}{N}}\right\}$$

$$=\frac{1}{2}\left\{\left[\left(\frac{1}{(0.765)^2}+1\right)^{\frac{1}{2}}+\frac{1}{0.765}\right]^{\frac{1}{6}}-\left[\left(\frac{1}{(0.756)^2}+1\right)^{\frac{1}{2}}+\frac{1}{0.765}\right]^{\frac{-1}{6}}\right\}$$

$$=\frac{1}{2}\{1.197-0.83\}=0.183$$

$\therefore\quad c_0=y_N=0.183$

$$c_k=y_N^2+\cos^2\frac{(2k-1)\pi}{2N}$$

$$c_1=y_N^2+\cos^2\frac{(2\times1-1)\pi}{2\times6}=(0.183)^2+\cos^2\left(\frac{\pi}{12}\right)=0.9664$$

$$b_1=2y_N\sin\frac{(2\times1-1)\pi}{2\times6}=2\times0.183\sin\left(\frac{\pi}{12}\right)=0.094$$

$$c_2=y_N^2+\cos^2\frac{(2\times2-1)\pi}{2\times6}=(0.183)^2+\cos^2\left(\frac{3\pi}{12}\right)=0.5334$$

$$b_2=2y_N\sin\frac{(2\times2-1)\pi}{2\times6}=2\times0.183\sin\left(\frac{3\pi}{12}\right)=0.258$$

$$c_3=y_N^2+\cos^2\frac{(2\times3-1)\pi}{2\times6}=0.1$$

$$b_3=2y_N\sin\frac{(2\times3-1)\pi}{2\times6}=0.353$$

Let $B_1 = B_2 = B_3$ and let $H_a(0) = 1$

$$\therefore\quad \frac{B_1B_2B_3\Omega_c^6}{c_1c_2c_3\Omega_c^6}=1$$

$$\therefore\quad B_1=B_2=B_3=(c_1c_2c_3)^{\frac{1}{3}}=(0.964\times0.533\times0.1)^{\frac{1}{3}}=0.371$$

$$\therefore\quad H_a(s)=\left(\frac{0.371\times(0.866)^2}{s^2+0.094\times0.866s+0.966\times(0.866)^2}\right)\left(\frac{0.371\times(0.866)^2}{s^2+0.258\times0.866s+0.533\times(0.866)^2}\right)\left(\frac{0.371\times(0.866)^2}{s^2+0.353\times0.866s+0.1\times(0.866)^2}\right)$$

$$=\left(\frac{0.278}{s^2+0.018s+0.724}\right)\left(\frac{0.278}{s^2+0.223s+0.399}\right)\left(\frac{0.278}{s^2+0.305s+0.074}\right)$$

**Step 7** Digital filter transfer function $H(z)$ taking $T$ = 1s.

$$H(z) = H_a(s)\Big|_{s=\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)} = H_a(s)\Big|_{s=2\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}$$

$$H(z) = \left[\frac{0.278}{\left\{\left\{2\left(\frac{1-z^{-1}}{1+z^{-1}}\right)^2 + 0.081 \times 2\left(\frac{1-z^{-1}}{1+z^{-1}}\right) + 0.724\right\}\right\}}\right]$$

$$\left[\frac{0.278}{\left[2\left(\frac{1-z^{-1}}{1+z^{-1}}\right)\right]^2 + 0.223\left[2\left(\frac{1-z^{-1}}{1+z^{-1}}\right)\right] + 0.399}\right]$$

$$\left[\frac{0.278}{\left[2\left(\frac{1-z^{-1}}{1+z^{-1}}\right)\right]^2 + 0.305 \times 2\left(\frac{1-z^{-1}}{1+z^{-1}}\right) + 0.074}\right]$$

$$= \left[\frac{0.278(1+z^{-1})^2}{4.886 - 6.552z^{-1} + 4.562z^{-2}}\right]\left[\frac{0.278(1+z^{-1})^2}{4.845 - 7.202z^{-1} + 3.953z^{-2}}\right]$$

$$\left[\frac{0.278(1+z^{-1})^2}{4.684 - 7.852z^{-1} + 3.464z^{-2}}\right]$$

**EXAMPLE 18**

Determine the lowest order of Chebyshev filter that meets the following specifications:

(i) 1 dB ripple in the passband $0 \le |\omega| \le 0.3\pi$

(ii) Atleast 60 dB attenuation in the stopband $0.35\pi \le |\omega| \le \pi$

Use the bilinear transformation.

***Solution:*** Given $\omega_1 = 0.3$ , $\omega_2 = 0.35$

1 dB ripple, so $\alpha_1 = 1$ dB or $k_1 = -1$ dB $A_1 = 10^{k_1/20} = 10^{-1/20} = 0.891$

60 dB attenuation, so $\alpha_2 = 60$ dB or $k_2 = -60$ dB

**Step 1** Bilinear transformation is to be used.

**Step 2** Attenuation constant

$$\varepsilon = \left[\frac{1}{A_1^2} - 1\right]^{\frac{1}{2}} = \left[\frac{1}{(0.891)^2} - 1\right]^{\frac{1}{2}} = 0.509$$

**Step 3** Ratio of analog edge frequencies

$$\frac{\Omega_2}{\Omega_1} = \frac{\frac{2}{T}\tan\frac{\omega_2}{2}}{\frac{2}{T}\tan\frac{\omega_1}{2}} = \frac{\tan\frac{0.35\pi}{2}}{\tan\frac{0.3\pi}{2}} = 1.2$$

**Step 4** Order of the filter

$$N \geq \frac{\cosh^{-1}\left[\frac{1}{\varepsilon}\left(\frac{1}{A_2^2}-1\right)^{\frac{1}{2}}\right]}{\cosh^{-1}\left(\frac{\Omega_2}{\Omega_1}\right)} \geq \frac{\cosh^{-1}\left[\frac{1}{0.509}\left(\frac{1}{0.001^2}-1\right)^{\frac{1}{2}}\right]}{\cosh^{-1}(1.2)}$$

$$\geq 13.338 \approx 14$$

So the lowest order of the filter is $N = 14$.

# UNIT- 5: Multirate Digital Signal Processing

## 5.0. INTRODUCTION

Discrete-time systems may be single-rate systems or multi-rate systems. The systems that use single sampling rate from A/D converter to D/A converter are known as single-rate systems and the discrete-time systems that process data at more than one sampling rate are known as multi-rate systems. In digital audio, the different sampling rates used are 32 kHz for broadcasting, 44.1 kHz for compact disc and 48 kHz for audio tape. In digital video, the sampling rates for composite video signals are 14.3181818 MHz and 17.734475 MHz for NTSC and PAL respectively. But the sampling rates for digital component of video signals are 13.5 MHz and 6.75 MHz for luminance and colour difference signal. Different sampling rates can be obtained using an up sampler and down sampler. The basic operations in multirate processing to achieve this are decimation and interpolation. Decimation is for reducing the sampling rate and interpolation is for increasing the sampling rate. There are many cases where multi-rate signal processing is used. Few of them are as follows:

1. In high quality data acquisition and storage systems
2. In audio signal processing
3. In video
4. In speech processing
5. In transmultiplexers
6. For narrow band filtering

The various advantages of multirate signal processing are as follows:

1. Computational requirements are less.
2. Storage for filter coefficients is less.
3. Finite arithmetic effects are less.
4. Filter order required in multirate application is low.
5. Sensitivity to filter coefficient lengths is less.

While designing multi-rate systems, effects of aliasing for decimation and pseudoimages for interpolators should be avoided.

## 5.1 SAMPLING

A continuous-time signal x(t) can be converted into a discrete-time signal x(nT) by sampling it at regular intervals of time with sampling period T. The sampled signal x(nT) is given by

$$x(nT) = x(t)\big|_{t=nT}, \quad -\infty < n < \infty$$

A sampling process can also be interpreted as a modulation or multiplication process.

### SANPLING THEOREM

Sampling theorem states that a band limited signal x(t) having finite energy, which has no spectral components higher than $f_h$ hertz can be completely reconstructed from its samples taken at the rate of $2f_h$ or more samples per second.

The sampling rate of $2f_h$ samples per second is the Nyquist rate and its reciprocal $1/2f_h$ is the Nyquist period.

## 5.3 DOWN SAMPLING

Reducing the sampling rate of a discrete-time signal is called down sampling. The sampling rate of the discrete-time signal can be reduced by a factor D by taking every $D^{th}$ value of the signal. Mathematically, down sampling is represented by and the symbol for the down sampler is shown in Figure 5.1.

$$y(n) = x(Dn)$$


**Figure 5.1** A down sampler.

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

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

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

x(2n) is obtained by keeping every second sample of x(n) and x(3n) is obtained by keeping every 3rd sample of x(n) and removing other samples.

If the input signal x(n) is not band limited, then there will be overlapping of spectra at the output of the down sampler. This overlapping of spectra is called aliasing which is undesirable. This aliasing problem can be eliminated by band limiting the input signal by inserting a low-pass filter called anti-aliasing filter before the down sampler. The anti-aliasing filter and the down sampler together is called decimator. The decimator is also known as sub sampler, down sampler or under sampler. Decimation (sampling rate compression) is the process of decreasing the sampling rate by an integer factor D by keeping every Dth sample and removing D – 1 in between samples.

Figure 10.2 shows the signal x(n) and its down sampled versions by a factor of 2 and 3.

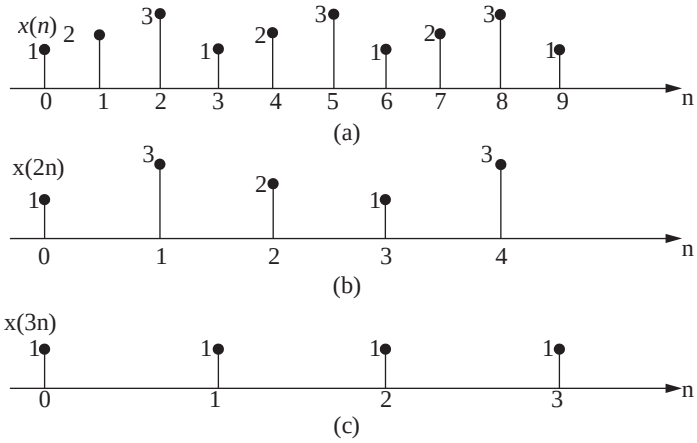



**Figure 5.2** Plots of (a) x(n), (b) x(2n) and (c) x(3n).

The block diagram of the decimator is shown in Figure 5.3. The decimator comprises two blocks such as anti-aliasing filter and down sampler. Here the anti-aliasing filter is a low-pass filter to band limit the input signal so that aliasing problem is eliminated and the down sampler is used to reduce the sampling rate by keeping every Dth sample and removing D – 1 in between samples.

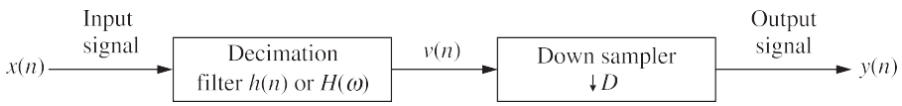



**Figure 5.3** Block diagram of decimator.

## SPECTRUM of Down SANPLED signal

Let T be sampling period of input signal x(n), and let F be its sampling rate or frequency. When the signal is down sampled by D, let T ' be its new sampling period and F' be its sampling frequency, then

$$\frac{T'}{T} = D$$

$$F' = \frac{1}{T'} = \frac{1}{TD} = \frac{F}{D}$$

Let us derive the spectrum of a down sampled signal x(Dn) and compare it with the spectrum of input signal x(n). The Z-transform of the signal x(n) is given by

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

The down sampled signal y(n) is obtained by multiplying the sequence x(n) with a periodic train of impulses p(n) with a period D and then leaving out the D – 1 zeros between each pair of samples. The periodic train of impulses is given by

$$p(n) = \begin{cases} 1, & n = 0,\ \pm D,\ \pm 2D, \ldots \\ 0, & \text{otherwise} \end{cases}$$

The discrete Fourier series representation of the signal p(n) is given by

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

Multiplying the sequence x(n) with p(n) yields

$$x'(n) = x(n)p(n)$$

$$x'(n) = \begin{cases} x(n), & n = 0,\ \pm D,\ \pm 2D, \ldots \\ 0, & \text{otherwise} \end{cases}$$

If we leave D – 1 zeros between each pair of samples, we get the output of down sampler

$$y(n) = x'(nD) = x(nD)\ p(nD)$$
$$= x(nD)$$

The Z-transform of the output sequence is given by

$$Y(z) = \sum_{n=-\infty}^{\infty} y(n) z^{-n}$$

$$= \sum_{n=-\infty}^{\infty} x'(nD) z^{-n}$$

$$= \sum_{n=-\infty}^{\infty} x'(n) z^{-n/D}$$

where $x'(n) = 0$ except at multiple of $D$. Since $x'(n) = x(n)\, p(n)$, we get

$$Y(z) = \sum_{n=-\infty}^{\infty} x(n)\ p(n) z^{-n/D}$$

$$= \sum_{n=-\infty}^{\infty} x(n) \left[ \frac{1}{D} \sum_{k=0}^{D-1} e^{j2\pi kn/D} \right] z^{-n/D}$$

$$= \frac{1}{D} \sum_{k=0}^{D-1} \sum_{n=-\infty}^{\infty} x(n) \, (e^{-j2\pi k/D} z^{1/D})^{-n}$$

$$= \frac{1}{D} \sum_{k=0}^{D-1} X[e^{-j2\pi k/D} z^{1/D}]$$

Substituting $z = e^{jw}$ , we get the frequency response

$$Y(\omega) = \frac{1}{D} \sum_{k=0}^{D-1} X(e^{-j2\pi k/D} e^{j\omega/D}) = \frac{1}{D} \sum_{k=0}^{D-1} X(e^{j(\omega - 2\pi k)/D})$$

$$Y(\omega) = \frac{1}{D} \sum_{k=0}^{D-1} X\left[ \frac{(\omega - 2\pi k)}{D} \right]$$

From the above relation we find that if the Fourier transform of the input signal x(n) of a down sampler is X(W), then the Fourier transform Y(W) of the output signal y(n) is a sum of D uniformly shifted and stretched versions of X(W) scaled by a factor 1/D.

If the spectrum of the original signal X(W) is band limited to = π/d, as shown in Figure 5.4(a), the spectrum being periodic with period 2 , the spectrum of the down sampled signal Y(W) is the sum of all the uniformly shifted and stretched versions of X(W) scaled by a factor 1/D as shown in Figure 5.4(b). In every interval of 2 in addition to the original spectrum we find D – 1 equally spaced replica.

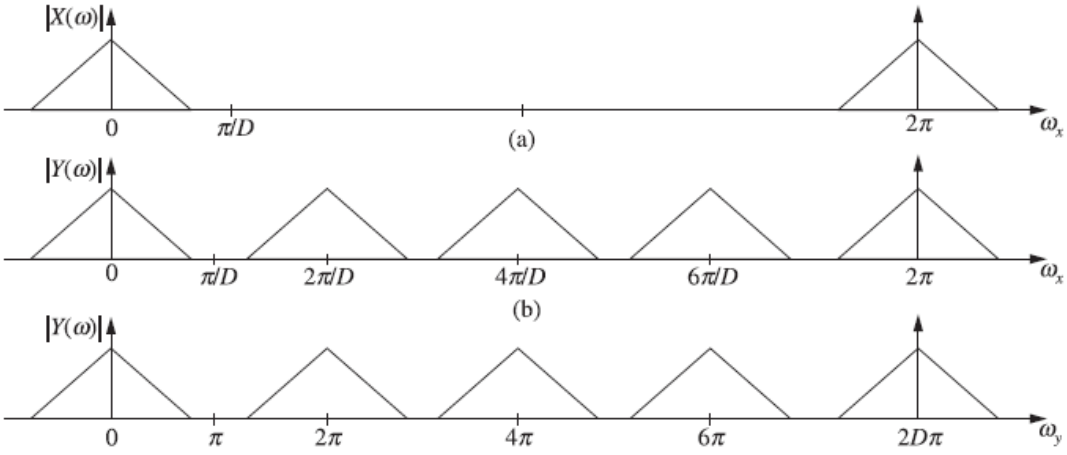



**Figure 5.4** Spectrum of (a) input, (b) output, and (c) normalized output.

## Aliasing effect and Anti-aliasing filter

From Figure 5.5, we can find that the spectrum obtained after down sampling will overlap if the original spectrum is not band limited to w=π/D. This overlapping of spectra is called aliasing. Therefore, aliasing due to down sampling a signal by a factor of D is absent if and only if the signal x(n) is band limited to ±π/D. If the signal x(n) is not band limited to ± π/D, then a low-pass filter with a cutoff frequency π/D is used prior to down sampling. This low-pass filter which is connected before the down sampler to prevent the effect of aliasing by band limiting the input signal is called the anti-aliasing filter.

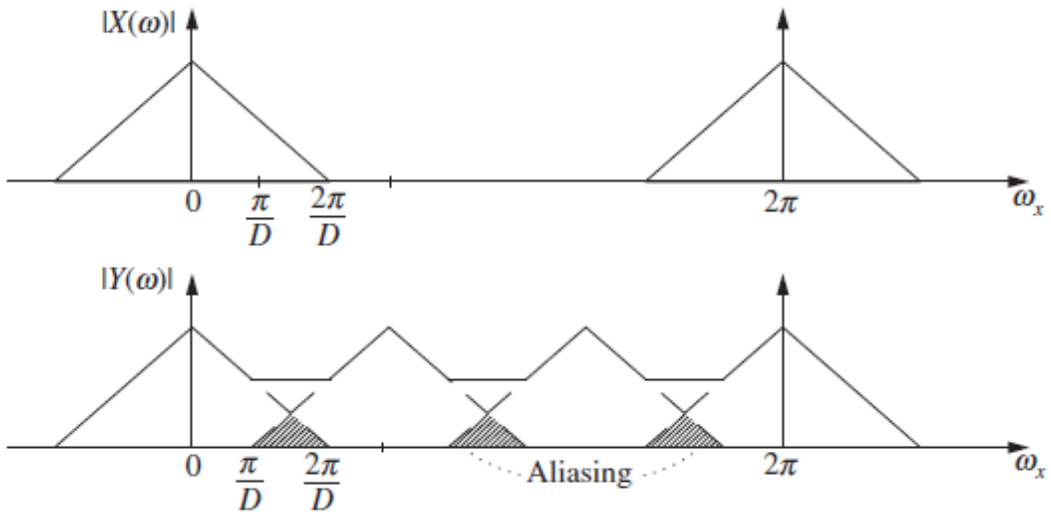



**Figure 5.5** (a) Input spectrum, (b) aliased output spectrum.

The signal obtained after filtering is given by

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

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

For example, consider a factor of D down sampler, then

$$Y(\omega) = \frac{1}{2}\sum_{k=0}^{1} X\left(\frac{\omega - 2\pi k}{2}\right)$$

$$= \frac{1}{2}\left[X\left(\frac{\omega}{2}\right) + X\left(\frac{\omega - 2\pi}{2}\right)\right]$$

$$= \frac{1}{2}\left[X\left(\frac{\omega}{2}\right) + X\left(-\frac{\omega}{2}\right)\right]$$

The second term X(–W /2) is simply obtained by shifting the first term X(W ) to the right by an amount of $2\pi$

## 5.4 UP SAMPLING

Increasing the sampling rate of a discrete-time signal is called up sampling. The sampling rate of a discrete-time signal can be increased by a factor I by placing I – 1 equally spaced zeros between each pair of samples.

Mathematically, up sampling is represented by

$$y(n) = \begin{cases} x\left(\frac{n}{I}\right), & n = 0, \pm I, \pm 2I, \ldots \\ 0, & \text{otherwise} \end{cases}$$

and the symbol for up sampler is shown in Figure 5.6.

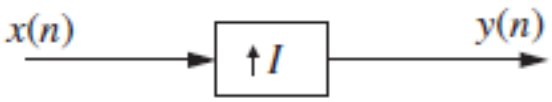



Figure 5.6 Up sampler

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

Then, $y(n) = x\left(\frac{n}{2}\right)$ $\{1, 0, 2, 0, 3, 0, 1, 0, 2, 0, 3, 0, \ldots\}$ for an up-sampling factor of $I = 2$.

and $y(n) = x\left(\frac{n}{3}\right)$ $\{1, 0, 0, 2, 0, 0, 3, 0, 0, 1, \ldots\}$ for an up-sampling factor of $I = 3$.

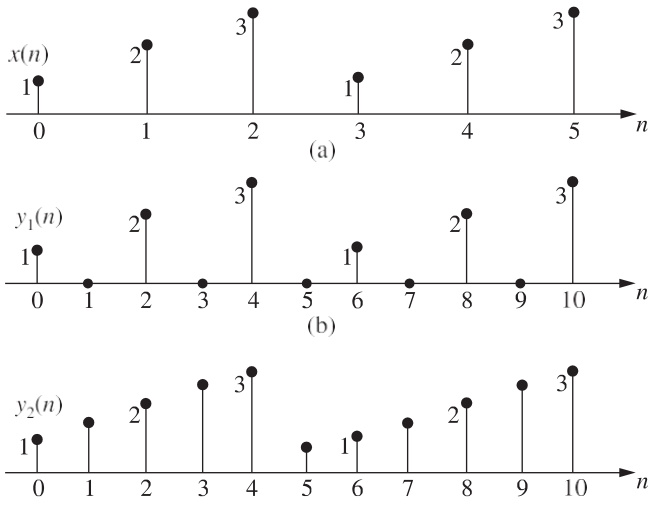
Usually an anti-imaging filter is to be kept after the up sampler to remove the unwanted images developed due to up sampling. The anti-imaging filter and the up sampler together is called interpolator. Interpolation is the process of increasing the sampling rate by an integer factor I by interpolating I – 1 new samples between successive values of the signal.

Figure 5.7 shows the signal x(n) and its two-fold up-sampled signal $y_1(n)$ and the interpolated signal $y_2(n)$.

The block diagram of the interpolator is shown in Figure 5.8. The interpolator comprises two blocks such as up sampler and anti-imaging filter. Here up sampler is used to increase the sampling rate by introducing zeros between successive input samples and the interpolation filter, also known as anti-imaging filter, is used to remove the unwanted images that are yielded by up sampling.

### Expression for output of interpolator

Let I be an integer interpolating factor of the signal. Let T be sampling period and F = 1/T be the sampling frequency (sampling rate) of the input signal. After up sampling, let T ' be the new sampling period and F' be the new sampling frequency, then



(c)

**Figure 5.7** (a) Input signal x(n), (b) Output of 2 fold up sampler $y_1(n) = x(n/2)$, (c) Output of interpolator $y_2(n) = x(n/2)$.

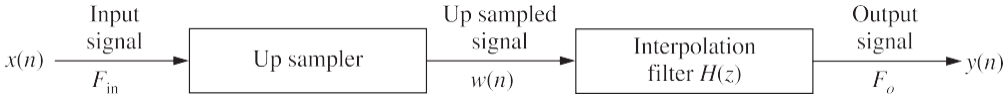



**Figure 5.8** Block diagram of an interpolator.

$$\frac{T'}{T} = \frac{1}{I}$$

The sampling rate is given by

$$F' = \frac{1}{T'} = \frac{I}{T} = IF$$

Let $w(n)$ be the signal obtained by interpolating $I-1$ samples between each pair of samples of $x(n)$.

$$w(n) = \begin{cases} x\left(\frac{n}{I}\right), & n = 0, \pm I, \pm 2I, \ldots \\ 0, & \text{otherwise} \end{cases}$$

The Z-transform of the signal $w(n)$ is given by

$$W(z) = \sum_{n=-\infty}^{\infty} w(n) z^{-n} = \sum_{n=-\infty}^{\infty} x\left(\frac{n}{I}\right) z^{-n}$$

$$= \sum_{n=-\infty}^{\infty} x(n)\, z^{-nI}$$

$$= X(z^I)$$

When considered over the unit circle $z = e^{j\omega'}$.

$$W(e^{j\omega'}) = X(e^{j\omega' I}) \text{ i.e. } W(\omega') = X(I\omega')$$

where $W = 2\pi fT$. The spectra of the signal w(n) contains the images of base band placed at the harmonics of the sampling frequency ±2 /I, ±4 /I. To remove the images an anti-imaging filter is used. The ideal characteristics of low-pass filter is given by

$$H(e^{j\omega'}) = \begin{cases} G, & |\omega'| \le 2\pi fT'/2 = \pi/I \\ 0, & \text{otherwise} \end{cases}$$

$$Y(e^{j\omega'}) = H(e^{j\omega'}) X(e^{j\omega' I})$$

$$= \begin{cases} GX(e^{j\omega' I}), & |\omega'| \le \pi/I \\ 0, & \text{otherwise} \end{cases}$$

The output signal $y(n)$ is given by

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

$$= \sum_{k=-\infty}^{\infty} h(n-k)\, x(k/I), \qquad k/I \text{ is an integer}$$

## ANTI-IMAGING Filter

The low-pass filter placed after the up sampler to remove the images created due to up sampling is called the anti-imaging filter.

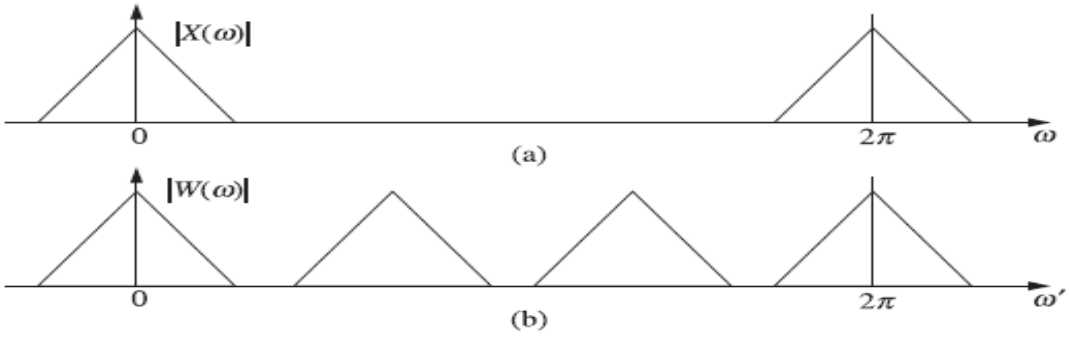



**Figure 5.9** Spectrum of (a) X( ) and (b) X(3 ).

**EXAMPLE 10.1** Show that the up sampler and down sampler are time-variant systems.

**Solution:** Consider a factor of I up sampler defined by

$$y(n) = x\left(\frac{n}{I}\right)$$

The output due to delayed input is given by

$$y(n, k) = x\left(\frac{n}{I} - k\right)$$

The delayed output is given by

$$y(n-k) = x\left(\frac{n-k}{I}\right)$$

Therefore, y(n, k) s y(n – k)

So the up sampler is a time-variant system.

Consider a factor of D down sampler defined by

y(n) = x(Dn)

The output due to delayed input is given by

y(n, k) = x(Dn – k)

The delayed output is given by

y(n – k) = x[D(n – k)]

Therefore, y(n, k) s y(n – k)

So the down sampler is a time-variant system.

**EXAMPLE 5.2** Consider a signal x(n) = u(n).

(i) Obtain a signal with a decimation factor 3.

(ii) Obtain a signal with an interpolation factor 3.

**Solution:** Given that x(n) = u(n) is the unit step sequence and is defined as:

$$u(n) = \begin{cases} 1, & \text{for } n \geq 0 \\ 0, & \text{elsewhere} \end{cases}$$

The graphical representation of unit step sequence is shown in Figure 5.10(a).

(i) Signal with a decimation factor 3.
The decimated signal is given by

$$y(n) = x(Dn) = x(3n)$$

It is obtained by considering only every third sample of x(n). The output signal y(n) is shown in Figure 5.10(b).

(ii) Signal with interpolation factor 3.
The interpolated signal is given by

$$y(n) = x\left(\frac{n}{I}\right) = x\left(\frac{n}{3}\right)$$

The output signal y(n) is shown in Figure 5.10(c). It is obtained by inserting two zeros between two consecutive samples.

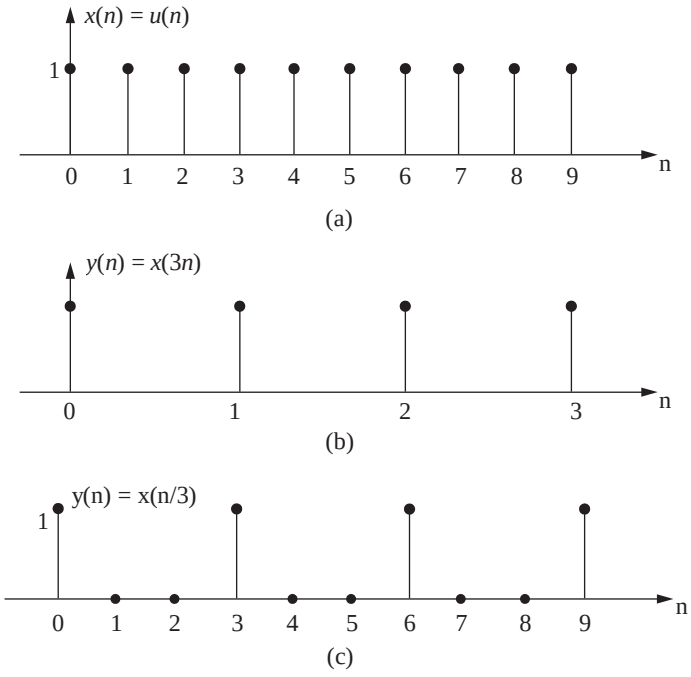



**Figure 5.10** Plots of (a) x(n)=u(n), (b) x(3n) and (c) x(n/3).

**EXAMPLE 5.3** Consider a ramp sequence and sketch its interpolated and decimated versions with a factor of 3.

**Solution:** The ramp sequence is denoted as r(n) and defined as

$$r(n) = \begin{cases} nu(n), & \text{for } n \geq 0 \\ 0, & \text{elsewhere} \end{cases}$$

The graphical representation of unit ramp signal is shown in Figure 5.11(a). The

decimated signal is given by

$$y(n) = r(Dn) = r(3n)$$

The output signal y(n) = r(3n) is shown in Figure 5.11(b). It is obtained by skipping 2 samples between every two successive sampling instants.

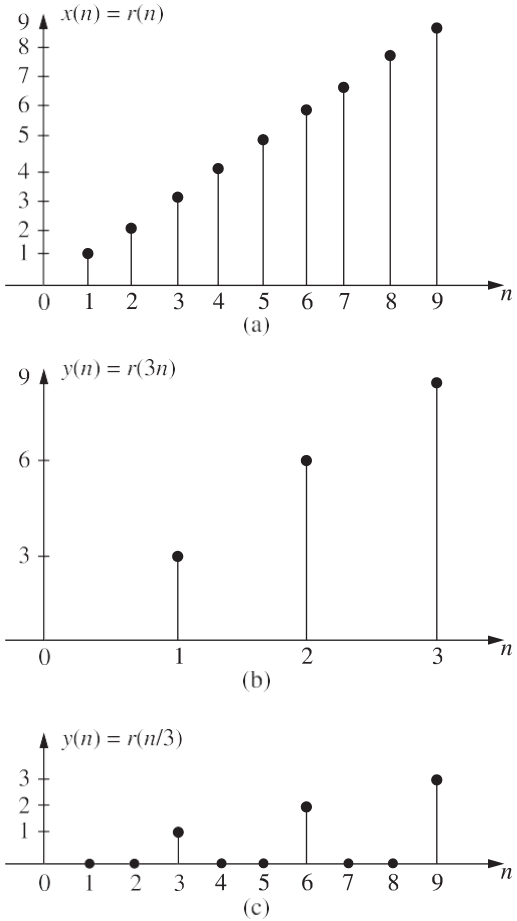



**Figure 5.11** Plots of (a) r(n) = nu(n), (b) y(n) = r(3n) and (c) y(n) = r(n/3).

**EXAMPLE 5.4** Consider a signal x(n) = sin nu(n).

(i) Obtain a signal with a decimation factor 2.

(ii) Obtain a signal with an interpolation factor 2.

**Solution:** The given signal is x(n) = sin n u(n). It is as shown in Figure 5.12(a).

(i) Signal with decimation factor 2. The signal x(n) with a decimation factor 2 is given

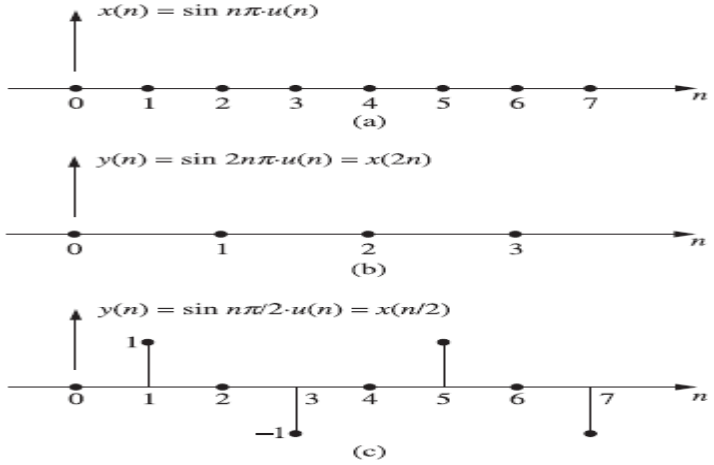



**Figure 5.12** Plots of (a) x(n) = sin n u(n), (b) y(n) = sin 2n u(n) and (c) y(n) = sin (n /2)u(n).

## 5.5 SAMPLIKG RATE CONVERSION

In some applications sampling rate conversion by a non-integer factor may be required. For example transferring data from a compact disc at a rate of 44.1 kHz to a digital audio tape at 48 kHz. Here the sampling rate conversion factor is 48/44.1, which is a non-integer.

A sampling rate conversion by a factor I/D can be achieved by first performing interpolation by factor I and then performing decimation by factor D. Figure 5.19(a) shows the cascade configuration of interpolator and decimator. The anti-imaging filter $H_u(z)$ and the anti-aliasing filter $H_d(z)$ are operated at the sampling rate, hence can be replaced by a simple low-pass filter with transfer function H(z) as shown in Figure 5.19(b), where the low-pass

$$\omega_c = \min\left[\frac{\pi}{I}, \frac{\pi}{D}\right].$$

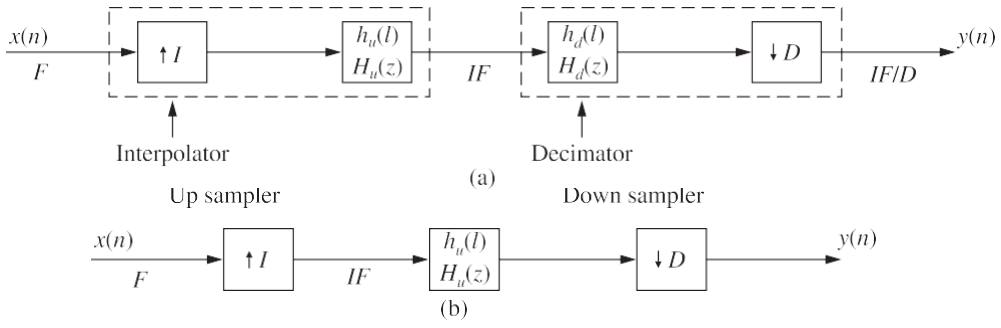



**Figure 5.19** Cascading of sample rate converters.

**EXAMPLE 5.7** Considering an example

$$x(n) = \{1, 3, 2, 5, 4, -1, -2, 6, -3, 7, 8, 9, ...\}$$

show that a cascade of D down sampler and I up sampler is interchangeable only when D and I are co-prime.

**Solution:** Given x(n) = {1, 3, 2, 5, 4, –1, –2, 6, –3, 7, 8, 9, ...}

(i) Let D = 2 and I = 3. Here D and I are co-prime.

$$x_d(n) = \{1, 2, 4, -2, -3, 8, \ldots\}$$

$$y_1(n) = \{1, 0, 0, 2, 0, 0, 4, 0, 0, -2, 0, 0, -3, 0, 0, 8, \ldots\}$$

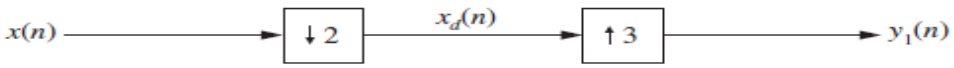



**Figure 5.20** Cascading of D = 2 and l = 3.

Interchanging the cascading as shown in Figure 5.22, we have

$$x_u(n) = \{1, 0, 0, 3, 0, 0, 2, 0, 0, 5, 0, 0, 4, 0, 0, -1, 0, 0, -2, 0, 0, 6, 0, 0, -3, 0, 0, 7, 0, 0, 8, \ldots)$$

$$y_2(n) = \{1, 0, 0, 2, 0, 0, 4, 0, 0, -2, 0, 0, -3, 0, 0, 8, \ldots\}$$

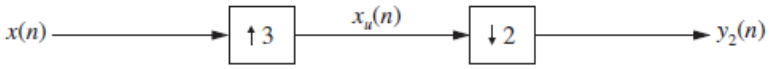



**Figure 5.21** Cascading of l = 3 and D = 2.

Now $y_1(n) = y_2(n)$. This shows that the cascade of an $I$ up sampler and a $D$ down sampler are interchangeable when $I$ and $D$ are co-prime.

(ii) Let $D = 2$ and $I = 4$. Here $D$ and $I$ are not co-prime.
For the cascading shown in Figure 10.23, we have

$$x_d(n) = \{1, 2, 4, -2, -3, 8, \ldots\}$$

$$y_3(n) = \{1, 0, 0, 0, 2, 0, 0, 0, 4, 0, 0, 0, -2, 0, 0, 0, -3, 0, 0, 0, -8, \ldots\}$$

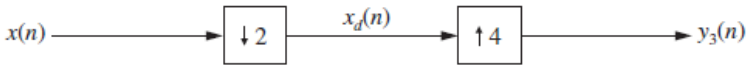



**Figure 5.22** Cascading of D = 2 and l = 4.

Interchanging the cascading as shown in Figure 5.23, we have

$$x_u(n) = \{1, 0, 0, 0, 3, 0, 0, 0, 2, 0, 0, 0, 5, 0, 0, 0, 4, 0, 0, 0, -1, \ldots\}$$

$$y_4(n) = \{1, 0, 3, 0, 2, 0, 5, 0, 4, 0, -1, \ldots\}$$

Now, $y_3(n) \neq y_4(n)$.

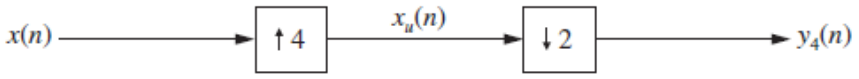



**Figure 5.23** Cascading of l = 4 and D = 2.

This shows that the cascading of up sampler and down sampler is not interchangeable when D and I are not co-prime, i.e., when D and I have a common factor.

# 5 Applications of multi-rate digital signal processing

Here we consider two applications of multi-rate digital signal processing.

1. **Implementation of a narrow band low-pass filter.** A narrow band low-pass filter is characterized by a narrow pass band and a narrow transition band. It requires a very large number of coefficients. Due to high value of N, it is susceptive to finite word length effects. In addition, the number of computations and memory locations required are very high. To overcome these problems multi- rate approach is used in implementing a narrow band low-pass filter. Figure 10.67 shows the cascading stage of a decimator and interpolator. The filters $h_1(n)$ and $h_2(n)$ in the decimator and interpolator are low-pass filters. The input signal is first passed through a low-pass filter. The sampling frequency F of the input sequence x(n) is first reduced by a factor D and then raised by the same factor D and then again low-pass filtering is performed.

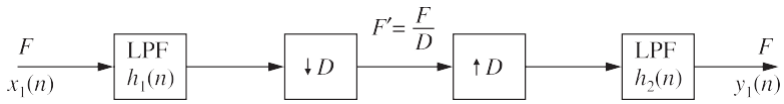



**Figure 5.24** A narrow band pass filter.

To meet the desired specifications of a narrow band LPF, the filters $h_1(n)$ and $h_2(n)$ should be identical with the same pass band ripple $_p/2$ and the same stop band ripple $_s$.

2. **Filter banks.** Filter banks are usually classified into two types:
(i) Analysis filter bank and (ii) Synthesis filter bank

## Analysis filter bank

The D-channel analysis filter bank is shown in Figure 10.68. It consists of D sub-filters. All the sub-filters are equally spaced in frequency and each have the same bandwidth. The spectrum of the input signal lies in the range $0 \leq \leq$ . The filter bank splits the signal into a number of sub-bands each having a bandwidth $\pi$ /D. The filter $H_0(z)$ is a low-pass filter, $H_1(z)$ to $H_{D-2}(z)$ are band pass and $H_{D-1}(z)$ is high-pass. As the spectrum of the signal is band limited to $\pi$ /D, the sampling rate can be reduced by a factor D. The down sampling moves all the pass band signals to a base band $0 \leq w \leq \pi$ /D.

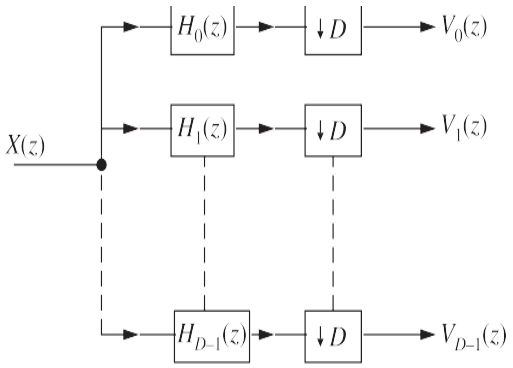



**Figure** 5.25 **Analysis filter bank**

## Synthesis filter bank bank

The D-channel synthesis filter bank shown in Figure 10.69 is dual of the analysis filter bank. In this case, each $V_d(z)$ is fed to an up sampler. The up-sampling process produces the signal $V_d(z^D)$. These signals are applied to filters $G_d(z)$ and finally added to get the output signal $\hat{X}(z)$. The filters $G_0(z)$ to $G_{D-1}(z)$ have the same characteristics as the analysis filters $H_0(z)$ to $H_{D-1}(z)$.

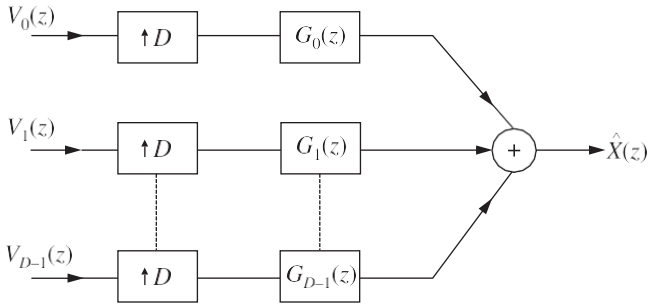



**Figure 5.26** Synthesis filter bank.

## Sub-band coding filter bank

By combining the analysis filter bank of Figure 5.25 and the synthesis filter bank of Figure 5.27, we can obtain a D-channel sub-band coding filter bank shown in Figure5.27. The analysis filter bank splits the broad band input signal x(n) into D non-overlapping frequency band signals $X_0(z)$, $X_1(z)$, ..., $X_{D-1}(z)$ of equal bandwidth. These outputs are coded and transmitted. The synthesis filter bank is used to reconstruct output signal $\hat{X}(z)$ which should approximate the original signal. Sub-band coding is very much used in speech signal processing.

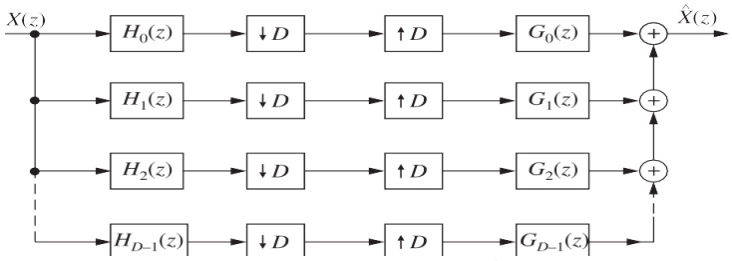



**Figure 5.27** Sub–band coding filter bank.

# FINITE WORD LENGTH EFFECTS IN DIGITAL FILTER

**Finite Word length Effects:**

- In the design of FIR Filters, The filter coefficients are determined by the system transfer functions. These filter co-efficient are quantized/truncated while implementing DSP System because of finite length registers.
- Only Finite numbers of bits are used to perform arithmetic operations. Typical word length is 16 bits, 24 bits, 32 bits etc.
- This finite word length introduces an error which can affect the performance of the DSP system.
- The main errors are
  1. Input quantization error
  2. Co-efficient quantization error
  3. Overflow & round off error (Product Quantization error)
- The effect of error introduced by a signal process depend upon number of factors including the
  1. Type of arithmetic
  2. Quality of input signal
  3. Type of algorithm implemented

**1. Input quantization error**

- The conversion of continuous-time input signal into digital value produces an error which is known as input quantization error.
- This error arises due to the representation of the input signal by a fixed number of digits in A/D conversion process.

**2. Co-efficient quantization error**

- The filter coefficients are compared to infinite precision. If they are quantized the frequency response of the resulting filter may differ from the desired frequency response.
  i.e poles of the desired filter may change leading to instability.

3. **Product Quantization error**

- It arises at the output of the multiplier
- When a 'b' bit data is multiplied with another 'b' bit coefficient the product ('2b' bits) should be stored in 'b' bits register. The multiplier Output must be rounded or truncated to 'b' bits. This known as overflow and round off error.

**********

**Types of number representation:**

There are two common forms that are used to represent the numbers in a digital or any other digital hardware.

1. Fixed point representation
2. Floating point representation

*** Explain the various formulas of the fixed point representation of binary numbers.**

**1. Fixed point representation**

- In the fixed point arithmetic, the position of the binary point is fixed. The bit to the right represents the fractional part of the number and to those to the left represents the integer part.

- For example, the binary number 01.1100 has the value 1.75 in decimal.

  $(0*2^{1)} + (1*2^{0}) + (1*2^{-1}) + (1*2^{-2}) + (0*2^{-3}) = 1.75$

  In general, we can represent the fixed point number 'N' to any desired accuracy by the series

$$N = \sum_{i=n_i}^{n_2} C_i r^i$$

  Where, r is called as radix.

- If r=10, the representation is known as decimal representation having numbers from 0 to 9. In this representation the number

$$30.285 = \sum_{i=-3}^{1_2} C_i 10^i$$

$$= (3*10^{1}) + (0*10^{0}) + (2*10^{-1}) + (8*10^{-2}) + (5*10^{-3})$$

- If r=2, the representation is known as binary representation with two numbers 0 to 1.
- For example, the binary number

$$110.010 = (1*2^{2}) + (1*2^{1}) + (0*2^{0}) + (0*2^{-1}) + (1*2^{-2}) + (0*2^{-3}) = 6.25$$

**Examples:**

**Convert the decimal number 30.275 to binary form**

| | | |
|---|---|---|
| 2 | 30 | |
| 2 | 15 | --0 |
| 2 | 7 | --1 |
| 2 | 3 | --1 |
| | 1 | --1 |

| | | |
|---|---|---|
| 0.275 * 2 | → 0.55 | → 0 |
| 0.55 * 2 | → 1.10 | → 1 |
| 0.10 * 2 | → 0.20 | → 0 |
| 0.20 * 2 | → 0.40 | → 0 |
| 0.40 * 2 | → 0.80 | → 0 |
| 0.80 * 2 | → 1.60 | → 1 |
| 0.60 * 2 | → 1.20 | → 1 |
| 0.20 * 2 | → 0.40 | → 0 |

**$(30.275)_{10}$ = $(11110.01000110)_2$**

In fixed point arithmetic =, the negative numbers are represented by 3 forms.

1. Sign-magnitude form
2. One's complement form
3. Two's complement form

**1.1 Sign-magnitude form:**

- Here an additional bit called sign bit is added as MSB.
  - If this bit is zero → It is a positive number
  - If this bit is one → It is a positive number
- For example
  - 1.75 is represented as 01.110000.
  - -1.75 is represented as 11.110000

**1.2 One's complement form:**

- Here the positive number is represented same as that in sign magnitude form.
- But the negative number is obtained by complementing all the bits of the positive number
- For eg: the decimal number -0.875 can be represented as
  - $(0.875)_{10}=(0.111000)_2$
  - $(-0.875)_{10}=(1.000111)_2$

0.111000
↓ ↓↓↓↓↓↓ (Complement each bit)
1.000111

**1.3 Two's complement form:**

- Here the positive numbers are represented as same in sign magnitude and one's complement form.
- The negative numbers are obtained by complementing all the bits of the positive number and adding one to the least significant bit

$(0.875)_{10}=(0.111000)_2$

↓ ↓↓↓↓↓↓ (Complement each bit)

1.000111

\+ 1

1.001000

$(-0.875)_{10}=(1.001000)_2$

**Examples:**

- **Find the sign magnitude, 1's complement, 2's complement for the given numbers.**

**1.** $-\frac{7}{32}$

**2.** $-\frac{7}{8}$

**3.** $+\frac{7}{8}$

**1.** $-\frac{7}{32}$

0.21875 * 2 ➔0.43750 ➔0
0.43750 * 2 ➔0.87500 ➔0
0.87500* 2 ➔ 1.750000 ➔1
0.75* 2 ➔1.50 ➔1
0.50* 2 ➔1.00 ➔1

$-\frac{7}{32}=(-0.21875)_{10}=(1.00111)_2$

Sign magnitude form = 1.00111
1's complement form = 1.11000
2's complement form = 1.11001

**2.** $-\frac{7}{8}$

0.875 * 2 ➔1.75 ➔1
0.750 * 2 ➔1.500 ➔1
0.500 * 2 ➔1.000 ➔1

$-\frac{7}{8}=(-0.875)_{10}=(0.111)_2$

Sign magnitude form = 0.111
1's complement form = 1.000
2's complement form = 1.001

**3.** $+\frac{7}{8}$

Sign magnitude form = 0.111
1's complement form = 0.111
2's complement form = 0.111

**Addition of two fixed point numbers:**

- **Add $(0.5)_{10}$ + $(0.125)_{10}$**

$(0.5)_{10}$ = $(0.100)_2$

$(0.125)_{10}$ = $(0.001)_2$

$(0.101)_2 = (0.625)_{10}$

- Addition of two fixed point numbers causes an overflow.

  For example

  $(0.100)_2$

  $(0.101)_2$

  $(1.001)_2 = (-0.125)_{10}$ in sign magnitude form

**Subtraction of two fixed point numbers:**

- Subtraction of two numbers can be easily performed easily by using two's complement representation.
- **Subtract 0.25 from 0.5**

  0.25 * 2 ➔0.50 ➔0 Sign magnitude form = $(0.010)_2$

  0.50 * 2 ➔1.00 ➔1 1's complement form = $(1.101)_2$

  0.00 * 2 ➔0.00 ➔0 2's complement form = $(1.110)_2$

  $(0.5)_{10}$ = $(0.100)_2$

  $-(0.25)_{10}$ = $(1.110)_2$ ➔Two's complement of -0.25

  $(10.010)_2$

  Here the carry is generated after the addition. Neglect the carry bit to get the result in decimal.

  $(0.010)_2 = (0.25)_{10}$

- **Subtract 0.5 from 0.25**

  0.5 * 2 ➔1.00 ➔1 Sign magnitude form = $(0.100)_2$

  0.00 * 2 ➔0.00 ➔0 1's complement form = $(1.011)_2$

  0.00 * 2 ➔0.00 ➔0 2's complement form = $(1.100)_2$

  $(0.25)_{10}$ = $(0.010)_2$

  $-(0.5)_{10}$ = $(1.100)_2$

  $(1.110)_2$

  Here the carry is not generated after the addition. So the result is negative.

**Multiplication in fixed point arithmetic:**

- Here the sign magnitude components are separated.
- The magnitudes of the numbers are multiplied. Then the sign of the product is determined and applied to the result.
- In the fixed point arithmetic, multiplication of two fractions results in a fraction.
- For multiplications with fractions, overflow can never occur.
- Eg:

  $(0.1001)_2 * (0.0011)_2$ = $(0.00011011)_2$

**2. Floating point representation**

- Here, a number 'x' is represented by

  $X=M.r^e$

  Where, M ➔ Mantissa which requires a sign bit for representing positive number and negative numbers.

  R ➔ base (or) radix

  e ➔ exponent which require an additional and it may be either positive or negative.

  - For eg, 278 can be represented in floating point representation.

    $278 = \frac{278 \text{ X } 1000}{1000} = 0.278*10_3$

    0.278 ➔ Mantissa (M)

    10 ➔ base (or) radix (r)

    3 ➔ exponents (e)

- Similarly, to represent a binary floating point number
  $X=M.2^e$ in which the fractional part of a number should fall (or) lie in the range of 1/2 to 1.

  $5 = \frac{5 \text{ X } 8}{8}$ $0.625 \text{ X} 2^3$

  Mantissa (M) = 0.625

  Base (or) radix (r) = 2

  Exponent (e) = 3
- Some decimal numbers and their floating point representations are given below:

  4.5 → $0.5625 \text{ X } 2^3$ $=0.1001 \text{ X } 2^{011}$

  1.5 → $0.75 \text{ X } 2^1$ $=0.1100 \text{ X } 2^{001}$

  6.5 → $0.8125 \text{ X } 2^3$ $=0.1100 \text{ X } 2^{011}$

  0.625 → $0.625 \text{ X } 2^0$ $=0.1010 \text{ X } 2^{000}$
- Negative floating point numbers are generally represented by considering the mantissa as a fixed point number. The sign of the floating point number is obtained from the first bit of mantissa.
- To represent floating point in multiplication

  Consider $X_1 = M_1 r^{e_1}$

  $X_2 = M_2 r^{e_2}$

  $X_1 X_2 = (M_1 * M_2) r^{(e_1+e_2)}$

**Example**

Given $X_1 = 3.5*10^{-12}$, $X_2 = 4.75*10^6$. Find the product $X_1 X_2$

$X=(3.5 \text{ X } 4.75)\ 10^{(-12+6)}$

$= (16.625)10^{-6}$ → in decimal

In binary: $(1.5)_{10} \text{ X } (1.25)_{10}$ = $(2^1 0.75) \text{ X } (2^1 0.625)$

= $2^{001} \text{ X } 0.1100 \text{ X } 2^{001} \text{ X } 0.1010$

= $2^{010} \text{ X } 0.01111$

**Addition and subtraction:**

- Here the exponent of a smaller number is adjusted until it matches the exponent of a larger number.
- Then, the mantissa are added or subtracted
- The resulting representation is rescaled so that its mantissa lies in the range 0.5 to 1.
- Eg: **Add $(3.0)_{10}$ & $(0.125)_{10}$**

  $(3.0)_{10}$ $=2^{010} \text{ X } 0.1100 = r^{e_1} \text{ x } M_1$

  $(0.125)_{10}$ $=2^{000} \text{ X } 0.0010 = r^{e_2} \text{ x } M_2$

  Now adjust $e_2$ Such that $e_1=e_2$

  $(0.125)_{10}$ $=2^{010} \text{ X } 0.0000100$

  Addition → $2^{010} (0.110000 + 0.0000100)$ → $2^{010} \text{ X } 0.110010$

  Subraction → $2^{010} \text{ X } 1.001101$

**Compare floating point with fixed point arithmetic.**

| Sl.No | Fixed point arithmetic | Floating point arithmetic |
|---|---|---|
| 1 | Fast operation | Slow operation |
| 2 | Relatively economical | More expensive because of costlier hardware |
| 3 | Small dynamic range | Increased Dynamic range |
| 4 | Round off errors occurs only for addition | Round off errors can occur with addition and multiplication |
| 5 | Overflow occur in addition | Overflow does not arise |
| 6 | Used in small computers | Used in large general purpose computers. |

**Quantization:**

***Discuss the various methods of quantization.**

***Derive the expression for rounding and truncation errors**

**** Discuss in detail about Quantization error that occurs due to finite word length of registers.**

The common methods of quantization are

1. Truncation
2. Rounding

**1. Truncation**

- The abrupt termination of given number having a large string of bits (or)
- Truncation is a process of discarding all bits less significant than the LSB that is retained.
- Suppose if we truncate the following binary number from 8 bits to 4 bits, we obtain
  - 0.00110011 to 0.0011
    (8 bits) (4 bits)
  - 1.01001001 to 1.0100
    (8 bits) (4 bits)
- When we truncate the number, the signal value is approximated by the highest quantization level that is not greater than the signal.

**2. Rounding (or) Round off**

- Rounding is the process of reducing the size of a binary number to finite word size of 'b' bits such that the rounded b-bit number is closest to the original unquantised number.

**Error Due to truncation and rounding:**

- While storing (or) computation on a number we face registers length problems. Hence given number is quantized to truncation (or) round off.
  i.e. Number of bits in the original number is reduced register length.

**Truncation error in sign magnitude form:**

- Consider a 5 bit number which has value of
  $0.11001_2 \rightarrow (0.7815)_{10}$
- This 5 bit number is truncated to a 4 bit number
  $0.1100_2 \rightarrow (0.75)_{10}$
  i.e. 5 bit number $\rightarrow$0.11001 has 'l' bits
  4 bit number $\rightarrow$0.1100 has 'b' bits
- Truncation error, $e_t$ = 0.1100 – 0.11001
  = -0.00001 $\rightarrow (-0.03125)_{10}$
- Here original length is 'l' bits. (l=5). The truncated length is 'b' bits.
- The truncation error, $e_t$ = $2^{-b}$-$2^{-l}$
  = -($2^{-l}$-$2^{-b}$)
  $e_t$ = -($2^{-5}$-$2^{-4}$) = $-2^{-1}$
- The truncation error for a positive number is
  $-\left(2^{-b}-2^{-l}\right) \le e_t \le 0$ $\rightarrow$ Non causal
- The truncation error for a negative number is
  $0 \le e_t \le \left(2^{-b}-2^{-l}\right)$ $\rightarrow$ Causal

**Truncation error in two's complement:**

- For a positive number, the truncation results in a smaller number and hence remains same as in the case of sign magnitude form.
- For a negative number, the truncation produces negative error in two's complement

$$-\left(2^{-b}-2^{-l}\right) \le e_t \le \left(2^{-b}-2^{-l}\right)$$

**Round off error (Error due to rounding):**

- Let us consider a number with original length as '5' bits and round off length as '4' bits.

$$0.11001 \xrightarrow{\text{Round off to}} 0.1101$$

- Now error due to rounding $e_r = \frac{2^{-b} - 2^{-l}}{2}$

  Where b→Number of bits to the right of binary point after rounding
  L→Number of bits to the right of binary point before rounding

- Rounding off error for positive Number:

$$-\frac{2^{-b} - 2^{-l}}{2} \le e_r \le 0$$

- Rounding off error for negative Number:

$$0 \le e_r \le \frac{2^{-b} - 2^{-l}}{2}$$

- For two's complement

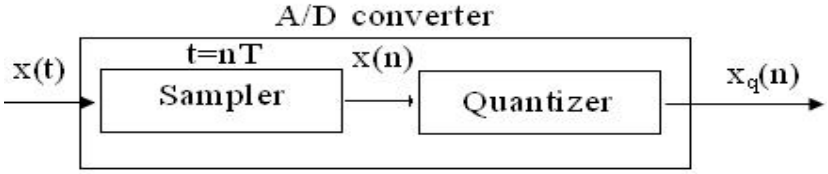
$$-\frac{2^{-b} - 2^{-l}}{2} \le e_r \le \frac{2^{-b} - 2^{-l}}{2}$$

**********

**Quantization Noise:**

***Derive the expression for signal to quantization noise ratio**

***What is called Quantization Noise? Derive the expression for quantization noise power.**




- The analog signal is converted into digital signal by ADC
- At first, the signal x(t) is sampled at regular intervals t=nT, where n=0,1,2… to create sequence x(n). This is done by a sampler.
- Then the numeric equivalent of each sample x(n) is expressed by a finite number of bits giving the sequence $x_q(n)$
- The difference signal e(n)= $x_q(n)$- x(n) is called quantization noise (or) A/D conversion noise.
- Let us assume a sinusoidal signal varying between +1 & -1 having a dynamic range 2
- ADC employs (b+1) bits including sign bit. In this case, the number of levels available for quantizing x(n) is $2^{b+1}$.
- The interval between the successive levels is

$$q = \frac{2}{2^{b+1}} = 2^{-b}$$

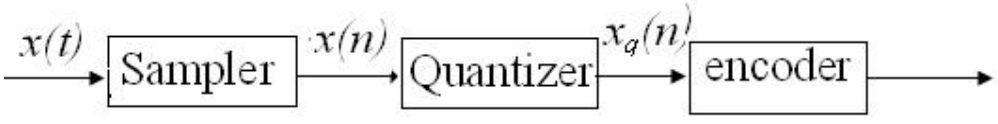
  Where q → quantization step size
  If b=3 bits, then $q=2^{-3}=0.125$

**********

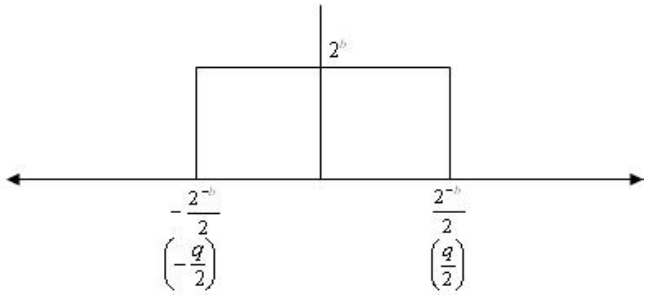
**Quantization Noise power:**

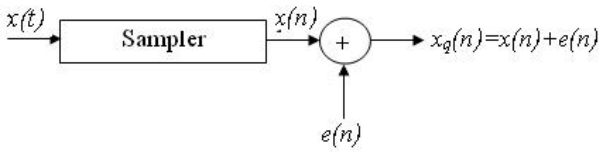
**<u>Input Quantization error:</u>**

***Derive the equation for quantization noise power (or) Steady state Input Noise Power.**



Probability density function for round off error in A/D conversion is




If rounding is used for quantization, which is bounded by $-\frac{q}{2} \le e(n) \le \frac{q}{2}$, then the error lies between $-\frac{q}{2}$ to $\frac{q}{2}$ with equal probability, where q→ quantization step size.

Properties of analog to digital conversion error, e(n):

1. The error sequence e(n) is a sample sequence of a stationary random process.
2. The error sequence is uncorrelated with x(n) and other signals in the system.
3. The error is a white noise process with uniform amplitude probability distribution over the range of quantization error.

The variance of e(n) is given by

$$\pm_e^2 = E\left[e^2(n)\right] - E^2\left[e(n)\right] \text{ ----------->(1)}$$

Where $E\left[e^2(n)\right]$→Average of $e^2(n)$

$E\left[e(n)\right]$→Mean value of e(n).

For rounding, e(n) lies between $-\frac{q}{2}$ and $\frac{q}{2}$ with equal probability

$$E\left[e^2(n)\right] = \int_{-\infty}^{\infty} e^2(n)p(e)de \text{ ----------->(2)}$$

$$p(e) = \frac{1}{q}, -\frac{q}{2} \le e(n) \le \frac{q}{2} \text{ ----------->(3)}$$

Substituting (3) in (2)

$$E\left[e^2(n)\right] = \int_{-\frac{q}{2}}^{\frac{q}{2}} e^2(n)\frac{1}{q}de$$

$$E\left[e^2(n)\right] = \frac{1}{q}\int_{-\frac{q}{2}}^{\frac{q}{2}} e^2(n)de \text{ ----------->(4)}$$

$$E\left[e(n)\right] = 0$$

$$E^2\left[e(n)\right] = 0 \text{ ----------->(5)}$$

Substituting (4) and (5) in (1)

$$\pm_e^2 = \frac{1}{q}\int_{-\frac{q}{2}}^{\frac{q}{2}} e^2(n)de - 0$$

$$= \frac{1}{q}\left[\frac{e^3}{3}\right]_{-\frac{q}{2}}^{\frac{q}{2}}$$

$$= \frac{1}{3q}\left[\left(\frac{q}{2}\right)^3 - \left(-\frac{q}{2}\right)^3\right]$$

$$= \frac{1}{3q}\left[\left(\frac{q^3}{8}\right) - \left(-\frac{q^3}{8}\right)\right]$$

$$= \frac{1}{3q}\left[\left(\frac{q^3}{8}\right) + \left(\frac{q^3}{8}\right)\right]$$

$$= \frac{1}{3q}\left[\frac{2q^3}{8}\right]$$

$$\pm_e^2 = \frac{q^2}{12} \text{ ----------->(6)}$$

In general, $\frac{1}{2^b} = 2^{-b} = q$ ----------->(7)

$$\pm_e^2 = \frac{\left(2^{-b}\right)^2}{12}$$

$$\pm_e^2 = \frac{2^{-2b}}{12} \text{ ----------->(8)}$$

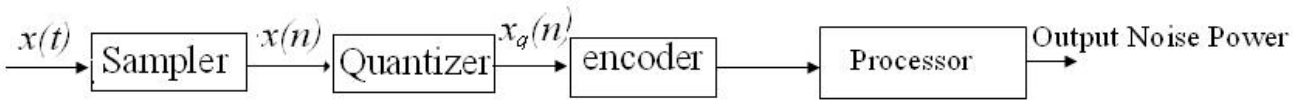
Equation (8) is known as the steady state noise power due to input quantization.

$q = \frac{R}{2^b}$ → in two's complement representation.

$q = \frac{R}{2^b - 1}$ → in sign magnitude (or) one's complement representation.

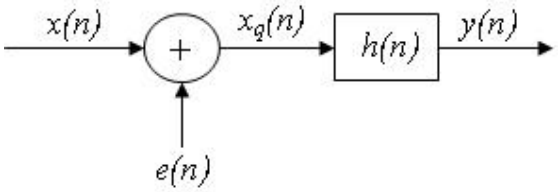
R → Range of analog signal to be quantized.

**Steady state Output Noise power:**




After quantization, we have noise power $\pm_e^2$ as input noise power. Therefore, Output noise power of system is given by

$$\pm_{eo}^2 = \pm_e^2\left[\sum_{n=0}^{\infty} h^2(n)\right] \text{ ----------->(9)}$$




where h(n) → impulse response of the system.

Let error E(n) be output noise power due to quantization

Error $E(n) = e(n) * h(n)$

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

The variance of error E(n) is called output noise power, $\pm_e^2$.

By using Parseval's theorem,

$$\pm_{eo}^2 = \pm_e^2 \sum_{n=0}^{\infty} h^2(n)$$

$$= \pm_e^2 \frac{1}{2\pi j} \oint H(Z) H(Z^{-1}) \frac{dZ}{z}$$

Where the closed contour integration is evaluated using the method of residue by taking only the poles that lie inside the unit circle.

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

Z transform of $h^2(n)$ = Z[$h^2(n)$] $= \sum_{n=0}^{\infty} h^2(n) z^{-n} = \sum_{n=0}^{\infty} h(n)h(n) z^{-n}$ ----------->(10)

By Inverse Z transform, $h(n) = \frac{1}{2\pi j} \oint H(Z) Z^{n-1} dZ$ ----------->(11)

Substituting (11) in (10)

$$\sum_{n=0}^{\infty} h^2(n) z^{-n} = \sum_{n=0}^{\infty} \frac{1}{2\pi j} \oint H(Z) Z^{n-1} dZ \; h(n) z^{-n}$$

$$= \frac{1}{2\pi j} \oint H(Z) \left[ \sum_{n=0}^{\infty} h(n) Z^{-1} \right] dZ$$

$$\sum_{n=0}^{\infty} h^2(n) = \frac{1}{2\pi j} \oint H(Z) \left[ \sum_{n=0}^{\infty} h(n) Z^{-1} \right] \frac{dZ}{Z^{-n}}$$

$$= \frac{1}{2\pi j} \oint H(Z) \left[ \sum_{n=0}^{\infty} h(n) \left(Z^{-n}\right)^{-1} Z^{-1} dZ \right]$$

$$\sum_{n=0}^{\infty} h^2(n) = \frac{1}{2\pi j} \oint H(Z) H(Z^{-1}) \frac{dZ}{Z}$$ ----------->(12)

Substituting (12) in (9)

$$\pm_{eo}^2 = \pm_e^2 \left[ \frac{1}{2\pi j} \oint H(Z) H(Z^{-1}) Z^{-1} dZ \right]$$

**Problem:**

**The output signal of an A/D converter is passed through a first order low pass filter, with transfer function given by**

$H(z) = \frac{(1-a)z}{z-a}$ *for* $0 < a < 1$. **Find the steady state output noise power due to quantization at the output of the digital filter. [Nov/Dec-2015]**

Solution:

$$\pm_e^2 = \pm_e^2 \frac{1}{2\pi j} \oint_c H(z) H(z^{-1}) z^{-1} dz$$

*Given* $H(z) = \frac{(1-a)z}{(z-a)}$ $H(z^{-1}) = \frac{(1-a)z^{-1}}{(z^{-1}-a)}$

*Substituting H(z) and H(z$^{-1}$) in equation (1), we have*

$$\pm_e^2 = \frac{\pm_e^2}{2\pi j} \oint_c \frac{(1-a)z}{(z-a)} \frac{(1-a)z^{-1}}{(z^{-1}-a)} z^{-1} dz = \frac{\pm_e^2}{2\pi j} \oint_c \frac{(1-a)^2}{(z-a)} \frac{dz}{(z^{-1}-a)} z^{-1}$$

$$= \pm_e^2 \left[ residue\ of\ H(z)\,H(z^{-1})z^{-1} at\ z = a\ + residue\ of\ H(z)\,H(z^{-1})z^{-1} at\ z = \frac{1}{a} \right]$$

$$= \pm_e^2 \left[ (z-a)\frac{(1-a)^2 z^{-1}}{(z-a)(z^{-1}-a)} + 0 \right]$$

$$= \pm_e^2 \left[ \frac{(1-a)^2}{(z^{-1}-a)} \right] = \pm_e^2 \left[ \frac{(1-a)}{(1+a)} \right]$$

*Where,* $\pm_e^2 = \frac{2^{-2b}}{12}$

**********

**Find the steady state variance of the noise in the output due to quantization of input for the first order filter.**

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

**Solution:**

The impulse response for the above filter is given by $h(n) = a^n u(n)$

$$\pm_\epsilon^2 = \pm_e^2 \sum_{k=0}^{\sim} h^2(n)$$

$$= \pm_e^2 \sum_{k=0}^{\sim} a^{2n}$$

$$= \pm_e^2 \left[ 1 + a^2 + a^4 + ....\infty \right]$$

$$= \pm_e^2 \frac{1}{1-a^2}$$

$$= \frac{2^{-2b}}{12}\left[ \frac{1}{1-a^2} \right] \qquad (or)$$

Taking Z-transform on both sides we have

$$Y(z) = az^{-1}Y(z) + X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1-az^{-1}} = \frac{z}{z-a}$$

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

We know

$$\pm_\epsilon^2 = \pm_e^2 \frac{1}{2\Re j}\int_c H(z)H(z^{-1})z^{-1}dz$$

*Substituting* $H(z)$ *and* $H(z^{-1})$ *values in the above equation we get*

$$\pm_\epsilon^2 = \pm_e^2 \frac{1}{2\Re j}\int_c \frac{z}{z-a}\frac{z^{-1}}{z^{-1}-a}z^{-1}dz$$

$$\pm_\epsilon^2 = \pm_e^2 \frac{1}{2\Re j}\int_c \frac{z^{-1}}{(z-a)(z^{-1}-a)}dz$$

$$= \pm_e^2 \left[ \begin{array}{l} residue\ of\ \dfrac{z^{-1}}{(z-a)(z^{-1}-a)}\ \ at\ z = a \\ +residue\ of\ \dfrac{z^{-1}}{(z-a)(z^{-1}-a)}\ \ at\ z = 1/a \end{array} \right]$$

$$= \pm_e^2 \left[ (z-a)\frac{z^{-1}}{(z-a)(z^{-1}-a)}\bigg|_{z=a} \right]$$

$$= \pm_e^2 \frac{a^{-1}}{a^{-1} - a} = \pm_e^2 \frac{1}{1 - a^2}$$

**********

**The output of the A/D converter is applied to a digital filter with the system function**

$$H(Z) = \frac{0.45Z}{Z - 0.72}$$

**Find the output noise power of the digital filter, when the input signal is quantized to 7 bits.**

Given:

$$H(Z) = \frac{0.45Z}{Z - 0.72}$$

**Solution:**

$$H(Z)H(Z^{-1})Z^{-1} = \frac{0.45Z}{Z - 0.72} \times \frac{0.45Z^{-1}}{Z^{-1} - 0.72} \times Z^{-1}$$

$$= \frac{0.45^2 Z^{-1}}{(Z - 0.72)\left(\frac{1}{Z} - 0.72\right)}$$

$$= \frac{0.2025Z^{-1}}{(Z - 0.72)\left(\frac{1 - 0.72Z}{Z}\right)}$$

$$= \frac{0.2025Z^{-1}Z}{(Z - 0.72)\left(Z - \frac{1}{0.72}\right)}$$

$$= \frac{-0.28125}{(Z - 0.72)(Z - 1.3889)}$$

Now the poles of $H(Z)H(Z^{-1})Z^{-1}$ are $p_1=0.72$ , $p_2=1.3889$

Output noise power due to input quantization

$$\pm_{eo}^2 = \pm_e^2 \left[ \frac{1}{2®j} \oint H(Z)H(Z^{-1})Z^{-1} dZ \right]$$

$$= \pm_e^2 \sum_{i=1}^{N} \operatorname{Re} s\left[H(Z)H(Z^{-1})Z^{-1}\right]_{z=p_i}$$

$$= \pm_e^2 \sum_{i=1}^{N} \operatorname{Re} s\left[H(Z)H(Z^{-1})Z^{-1}\right]_{z=p_i}$$

Where $p_1, p_2, \ldots . p_n$ are the poles of $H(Z)H(Z^{-1})\ Z^{-1}$ that lies inside the unit circle in z-plane.

$$\pm_{eo}^2 = \pm_e^2 \times (Z - 0.72) \times \left. \frac{-0.28125}{(Z - 0.72)(Z - 1.3889)} \right|_{Z=0.72}$$

$$= \pm_e^2 \times \frac{-0.28125}{0.72 - 1.3889}$$

$$= 0.4205 \pm_e^2$$

**********

**Consider the transfer function** $H(z) = H_1(z)H_2(z)$ **where** $H_1(z) = \frac{1}{1 - a_1 z^{-1}}$ *and* $H_2(z) = \frac{1}{1 - a_2 z^{-1}}$

**Find the output round off noise power. Assume** $\sim_1 = 0.5 \; and \; \sim_2 = 0.6$ **and find output round off noise power.**

**Solution:**

The round off noise model for $H(z) = H_1(z)H_2(z)$ is given by,

From the realization we can find that the noise transfer function seen by noise source $e_1(n)$ is H(z), where,

$$H(z) = \frac{1}{(1-a_1 z^{-1})(1-a_2 z^{-1})} \text{ ----------(1)}$$

*Whereas, the noise transfer function seen by* $e_2(n)$ *is,*

$$H_2(z) = \frac{1}{(1-a_2 z^{-1})} \text{ ----------(2)}$$

The total steady state noise variance can be obtained, we have

$$\pm_0^2 = \pm_{01}^2 + \pm_{02}^2 \text{ ----------(3)}$$

$$\pm_{01}^2 = \frac{1}{2\pi j}\oint_c H(z)H(z^{-1})z^{-1}dz$$

$$= \pm_e^2 \frac{1}{2\pi j}\oint_c \frac{1}{1-a_1 z^{-1}}\frac{1}{1-a_2 z^{-1}}\frac{1}{1-a_1 z}\frac{1}{1-a_2 z} z^{-1}dz$$

$$= \pm_e^2 \left[\sum of \; residue \; of \; H(z)H(z^{-1})z^{-1} \; at \; po1es \; z = a_1, z = a_2, z = \frac{1}{a_1} \; and \; z = \frac{1}{a_2}\right]$$

If a1 and a2 are less than the poles $z=1/a_1$ and $z=1/a_2$ lies outside of the circle $|z| = 1$. So, the residue of H(z) $H(z^{-1})$ $z^{-1}$ at $z=1/a_1$ and $z=1/a_2$ are zero. Consequently we have,

$$\pm_{01}^2 = \left[\sum of \; residue \; of \; H(z)H(z^{-1})z^{-1} \; at \; po1es \; z = a_1, z = a_2\right]$$

$$= \left[(z-a_1)\frac{z^{-1}}{(1-a_1 z^{-1})(1-a_2 z^{-1})(1-a_1 z)(1-a_2 z)}\bigg|_{z=a_1} + (z-a_2)\frac{z^{-1}}{(1-a_1 z^{-1})(1-a_2 z^{-1})(1-a_1 z)(1-a_2 z)}\bigg|_{z=a_2}\right]$$

$$= \pm_e^2 \left[\frac{1}{\left(1-\frac{a_2}{a_1}\right)(1-a_2^2)(1-a_1 a_2)} + \frac{1}{\left(1-\frac{a_2}{a_1}\right)(1-a_1 a_2)(1-a_2^2)}\right]$$

$$\pm_{01}^2 = \pm_e^2 \left[\frac{a_1}{a_1-a_2}\cdot\frac{1}{1-a_1^2}\cdot\frac{1}{1-a_1 a_2} + \frac{a_2}{a_2-a_1}\cdot\frac{1}{1-a_2^2}\cdot\frac{1}{1-a_1 a_2}\right] \text{ ----------(4)}$$

In the same way,

$$\pm_{02}^2 = \frac{\pm_e^2}{2\pi j}\oint_c H_2(z)H_2(z^{-1})z^{-1}dz$$

$$= \frac{\pm_e^2}{2\pi j}\oint_c \frac{1}{1-a_2 z^{-1}}\frac{1}{1-a_2 z} z^{-1}dz$$

$$= \pm_e^2 \left[(z-a_2)\frac{z^{-1}}{(1-a_2 z^{-1})(1-a_2 z)}\bigg|_{z=a_2}\right]$$

$$= \pm_e^2 \left[(z-a_2 z^{-1})\frac{z^{-1}}{(1-a_2 z^{-1})(1-a_2 z)}\bigg|_{z=a_2}\right]$$

$$= \pm_e^2 \left[\frac{1}{1-a_2^2}\right] \text{ ----------(5)}$$

$$\pm_0^2 = \pm_e^2\left[\frac{1}{1-a_2^2} + \frac{a_1}{a_1-a_2}\cdot\frac{1}{1-a_1^2}\cdot\frac{1}{1-a_1a_2} + \frac{a_2}{a_2-a_1}\cdot\frac{1}{1-a_2^2}\cdot\frac{1}{1-a_1a_2}\right]$$

$$= \pm_e^2\left[\frac{1}{1-a_2^2} + \frac{a_1(1-a_2^2)-a_2^2(1-a_1^2)}{(1-a_1^2)(1-a_2^2)(1-a_1a_2)(a_1-a_2)}\right]$$

$$= \pm_e^2\left[\frac{1}{1-a_2^2} + \frac{(a_1-a_2)(1+a_1a_2)}{(1-a_1^2)(1-a_2^2)(1-a_1a_2)(a_1-a_2)}\right]$$

$$= \frac{2^{-2b}}{12}\left[\frac{1}{1-a_2^2} + \frac{(1+a_1a_2)}{(1-a_1^2)(1-a_2^2)(1-a_1a_2)}\right]$$

The steady state noise power for $a_1 = 0.5, a_2 = 0.6$ is given by

$$= \frac{2^{-2b}}{12}\left[\frac{1}{1-(0.6)^2} + \frac{1+(0.5)(0.6)}{(1-(0.5)^2)(1-(0.6)^2)(1-0.6(0.5))}\right]$$

$$= \frac{2^{-2b}}{12}(5.4315)$$

**********

**Draw the quantization noise model for a second order system $H(z) = \frac{1}{1-2r\cos z^{-1} + r^2z^{-2}}$ and find the steady state output noise variance.**

**Solution:**

Given:

$$H(z) = \frac{1}{1-2r\cos z^{-1} + r^2z^{-2}}$$

**The quantization noise model is,**

**we know,** $\pm_0^2 = \pm_{01}^2 + \pm_{02}^2$

Both noise sources see the same transfer function

$$H(z) = \frac{1}{1-2r\cos z^{-1} + r^2z^{-2}}$$

The impulse response of the transfer function is given by

$$h(n) = r^n\frac{\sin(n+1)}{\sin}u(n)$$

Now the steady state output noise variance is,

$\pm_0^2 = \pm_{01}^2 + \pm_{02}^2$

**But** $\pm_{01}^2 = \pm_{02}^2 = \pm_e^2\sum_{n=-\infty}^{\infty}h^2(n)$, *which gives us*

$$\pm_0^2 = 2.\frac{2^{-2b}}{12}\sum_{n=0}^{\infty} r^{2n}\frac{\sin^2(n+1)\bar{}}{\sin^2\bar{}}$$

$$= 2.\frac{2^{-2b}}{12}\frac{1}{2\sin^2\bar{}}\sum_{n=0}^{\infty} r^{2n}\left[1-\cos 2(n+1)\bar{}\right] \qquad \therefore cos2\bar{} = 1 - 2sin^2\bar{}$$

$$= \frac{2^{-2b}}{6}\frac{1}{2\sin^2\bar{}}\left[\sum_{n=0}^{\infty} r^{2n} - \sum_{n=0}^{\infty} r^{2n}\cos 2(n+1)\bar{}\right]$$

$$= \frac{2^{-2b}}{6}\frac{1}{2\sin^2\bar{}}\left[\frac{1}{1-r^2} - \frac{1}{2}\left(\sum_{n=0}^{\infty} r^{2n}e^{j2(n+1)\bar{}} + \sum_{n=0}^{\infty} r^{2n}e^{-j2(n+1)\bar{}}\right)\right]$$

$$= \frac{2^{-2b}}{6}\frac{1}{2\sin^2\bar{}}\left[\frac{1}{1-r^2} - \frac{1}{2}\left(\frac{e^{j2\bar{}}}{1-r^2e^{2j\bar{}}} + \frac{e^{-j2\bar{}}}{1-r^2e^{-2j\bar{}}}\right)\right]$$

$$= \frac{2^{-2b}}{6}\frac{1}{2\sin^2\bar{}}\left[\frac{1}{1-r^2} - \frac{\cos 2\bar{} - r^2}{1-2r^2\cos 2\bar{} + r^4}\right]$$

$$= \frac{2^{-2b}}{6}\frac{1}{2\sin^2\bar{}}\left[\frac{(1+r)^2(1-\cos 2\bar{})}{(1-r^2)(1-2r^2\cos 2\bar{} + r^4)}\right]$$

$$= \frac{2^{-2b}}{6}\frac{(1+r)^2}{(1-r^2)(1-2r^2\cos 2\bar{} + r^4)}$$

**********

**Co-efficient quantization error**

- We know that the IIR Filter is characterized by the system function

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

- After quantizing ,

$$[H(Z)]_q = \frac{\sum_{k=0}^{M} [b_k]_q z^{-k}}{1+\sum_{k=1}^{N} [a_k]_q z^{-k}}$$

Where $[a_k]_q = a_k + \Delta a_k$

$[b_k]_q = b_k + \Delta b_k$

- The quantization of filter coefficients alters the positions of the poles and zeros in z-plane.
  1. If the poles of desired filter lie close to the unit circle, then the quantized filter poles may lie outside the unit circle leading into instability of filter.
  2. Deviation in poles and zeros also lead to deviation in frequency response.

**********

**Consider a second order IIR filter with $H(z) = \frac{1.0}{(1-0.5z^{-1})(1-0.45z^{-1})}$ find the effect on quantization on pole locations of the given system function in direct form and in cascade form. Take b=3bits. [Apr/May-10] [Nov/Dec-11]**

**Solution:**

Given that,

$$H(z) = \frac{1.0}{(1-0.5z^{-1})(1-0.45z^{-1})}$$

$$H(z) = \frac{1}{z^{-1}(z-0.5z^{-1})z^{-1}(z-0.5)}$$

$$= \frac{z^2}{(z-0.5)(z-0.45)}$$

The roots of the denominator of H(z) are the original poles of H(z). let the original poles of H(z) be $p_1$ and $p_2$.

Here $p_1$=0.5 and $p_2$=0.45

**Direct form I:**

$$H(z) = \frac{1.0}{(1-0.5z^{-1})(1-0.45z^{-1})}$$

$$H(z) = \frac{1}{1-0.5z^{-1}-0.45z^{-1}+0.225z-2}$$

$$= \frac{1}{1-0.95z^{-1}+0.225z^{-2}}$$

Let us quantize the coefficients by truncation.

$.95_{10}$ —Convert to Binary→ $.1111_2$ —Truncate to 3-bits→ $.111_2$ —Convert to decimal→ $.875_{10}$

$.225_{10}$ —Convert to Binary→ $.0011_2$ —Truncate to 3-bits→ $.001_2$ —Convert to decimal→ $.125_{10}$

Let $\overline{H}(z)$ be the transfer function of the IIR system after quantizing the coefficients.

$$\overline{H}(z) = \frac{1}{1-0.875z^{-1}+0.125z^{-2}}$$

$$let\ \overline{H}(z) = \frac{Y(z)}{X(z)} = \frac{1}{1-0.875z^{-1}+0.125z^{-2}}$$

On cross multiplying the above equation we get,

$$Y(z) - 0.875z^{-1}Y(z) + 0.125z^{-2}Y(z) = X(z)$$

$$Y(z) = X(z) + 0.875z^{-1}Y(z) - 0.125z^{-2}Y(z)$$

**<u>Cascade form</u>:**

Given that

$$H(z) = \frac{1.0}{(1-0.5z^{-1})(1-0.45z^{-1})}$$

In cascade realization the system can be realized as cascade of first order sections.

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

Where,

$$H_1(z) = \frac{1}{1-0.5z^{-1}}\ and\ \ H_2(z) = \frac{1}{1-0.45z^{-1}}$$

Let us quantize the coefficients of $H_1(z)$ and $H_2(z)$ by truncation.

$.5_{10}$ —Convert to Binary→ $.1000_2$ —Truncate to 3-bits→ $.100_2$ —Convert to decimal→ $.5_{10}$

$.45_{10}$ —Convert to Binary→ $.0111_2$ —Convert to 3-bits→ $.011_2$ —Convert to decimal→ $.375_{10}$

let , $\overline{H_1}(z)$ and $\overline{H_2}(z)$ be the transfer function of the first-order sections after quantizing the coefficients.

$\overline{H_1}(z) = \frac{1}{1-0.5z^{-1}}$

$\overline{H_2}(z) = \frac{1}{1-0.375z^{-1}}$

$let, \overline{H_1}(z) = \frac{Y_1(z)}{X(z)} = \frac{1}{1-0.5z^{-1}}$

$Y_1(z) - 0.5z^{-1}Y_1(z) = X(z)$

$Y_1(z) = X(z) + 0.5z^{-1}Y_1(z)$

$\text{let}, \overline{H_2}(z) = \frac{Y(z)}{Y_1(z)} = \frac{1}{1-0.375z^{-1}}$

*on cross multiplying the above equation we get*,

$Y(z) - 0.375z^{-1}Y(z) = Y_1(z)$

$Y(z) = Y_1(z) + 0.375z^{-1}Y(z)$

**********

**Round off effects and overflow in digital filter:**

***Explain in detail about round off effects in digital filters.**

- The presence of one or more quantizer in the realization of a digital filter results in a non-linear device. i.e. recursive digital filter may exhibit undesirable oscillations in its output
- In the finite arithmetic operations, some registers may overflow if the input signal level becomes large.
- These overflow represents non-linear distortion leading to limit cycle oscillations
- There are two types of limit cycle oscillations which includes
  1. Zero input limit cycle oscillations (Low amplitude compared to overflow limit cycle oscillations)
  2. Over flow limit cycle oscillations.

Zero input limit cycle oscillations

- The arithmetic operations produces oscillations even when the input is zero or some non zero constant values. Such oscillations are called zero input limit cycle oscillations.

Overflow limit cycle oscillations

- The limit cycle occurs due to the overflow of adder is known as overflow limit cycle oscillations.

Dead Band:

The limit cycle occurs as a result of quantization effect in multiplication. The amplitude of the output during a limit cycle is confined to a range of values called the dead band of the filter**.**

$$|y(n-1)| \le \frac{2^{-b}}{(1-|a|)}$$

Consider a first order filter

$$y(n) = ay(n-1) + x(n); \qquad n > 0$$

After rounding the product

$$y_q(n) = Q[a * y(n-1)] + x(n);$$

The round off error

$$-\frac{2^{-b}}{2} \le e_r \le \frac{2^{-b}}{2}$$

where, $e_r$ → difference between the quantized value and the actual value.

$$Q[ay(n-1) - ay(n-1)] \le \frac{2^{-b}}{2}$$

The dead band of the filter for the limit cycle oscillations are

$$Q[ay(n-1)] = \begin{cases} y(n-1) & a > 0 \\ -y(n-1) & a < 0 \end{cases}$$

$$|y(n-1)| - a|y(n-1)| \le \frac{2^{-b}}{2}$$

$$y(n-1)\left(1-|a|\right) \le \frac{2^{-b}}{2}$$

Dead band of the filter, $|y(n-1)| \le \dfrac{\frac{2^{-b}}{2}}{\left(1-|a|\right)}$

**********

**Problem: Consider a 1st order FIR system equation** $y(n) = x(n) + ay(n-1)$ **with**

$$x(n) = \begin{cases} 0.875 & , \ n = 0 \\ 0 & , \ otherwise \end{cases}$$

**Find the limit cycle effect and the dead band. Assume b=4 and a=0.95. (Nov/Dec-12)(Nov/Dec-15) [May/June-2016]**

**Solution:**

**Given:**

$$x(n) = \begin{cases} 0.875 & , \ n = 0 \\ 0 & , \ otherwise \end{cases}$$

$$Dead\ band = \frac{2^{-b}}{2\left(1-|a|\right)} = \frac{2^{-4}}{2\left(1-|0.95|\right)} = 0.625$$

$$y(n) = x(n) + 0.95y(n-1)$$

| ***n*** | ***x(n)*** | $y(n-1)$ | $ay(n-1)$ | $Q[ay(n-1)]$ *(round off to 4-bits)* | $y(n) = x(n) + Q[ay(n-1)]$ |
|---|---|---|---|---|---|
| 0 | 0.875 | 0 | 0 | 0.0000 | y(0)=0.875 |
| 1 | 0 | 0.875 | $0.875 * 0.95$<br>$= (0.83125)_{10}$<br>$= (0.11010)_2$ | $= (0.1101)_2$<br>$= 0.8125$ | y(1)=0.8125 |
| 2 | 0 | 0.8125 | $0.8125 * 0.95$<br>$= (0.77187)_{10}$<br>$= (0.110001)_2$ | $= (0.1100)_2$<br>$= 0.75$ | $y(2) = 0.75$ |
| 3 | 0 | 0.75 | $0.75 * 0.95$<br>$= (0.7125)_{10}$<br>$= (0.1011011)_2$ | $= (0.1011)_2$<br>$= 0.6875$ | $y(3) = 0.6875$ |
| 4 | 0 | 0.6875 | $0.6875 * 0.95$<br>$= (0.653125)_{10}$<br>$= (0.101001)_2$ | $= (0.1010)_2$<br>$= 0.625$ | $y(4) = 0.625$ |
| 5 | 0 | 0.625 | $0.625 * 0.95$<br>$= (0.59375)_{10}$<br>$= (0.10011)_2$ | $= (0.1010)_2$<br>$= 0.625$ | $y(5) = 0.625$ |
| 6 | 0 | 0.625 | $0.625 * 0.95$<br>$= (0.59375)_{10}$<br>$= (0.10011)_2$ | $= (0.1010)_2$<br>$= 0.625$ | $y(6) = 0.625$ |

**Conclusion:**

The dead band of the filter is 0.625. When $n \geq 5$ the output remains constant at 0.625 causing limit cycle oscillations.

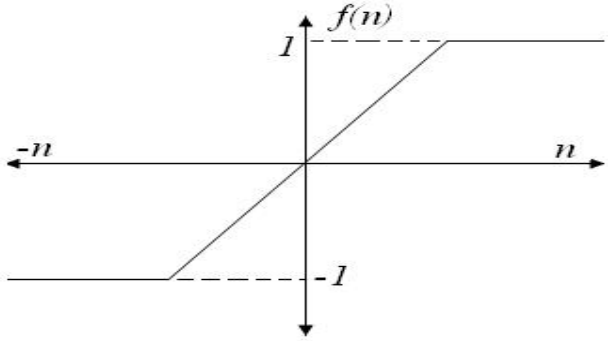
**********

**Overflow Limit cycle oscillations:**

***What are called overflow oscillations? How it can be prevented?**

- We know that the limit cycle oscillation is caused by rounding the result of multiplication.
- The limit cycle occurs due to the overflow of adder is known as overflow limit cycle oscillations.\
- Several types of limit cycle oscillations are caused by addition, which makes the filter output oscilate between maximum and minimum amplitudes.
- Let us consider 2 positive numbers $n_1$ & $n_2$
  $n_1$=0.111→7/8
  $n_2$=0.110→6/8
  $n_1$ + n2=1.101→-5/8 in sign magnitude form.
  The sum is wrongly interpreted as a negative number.
- The transfer characteristics of an saturation adder is shown in fig below

  where n → The input to the adder
  f(n) → The corresponding output




**Saturation adder transfer characteristics**

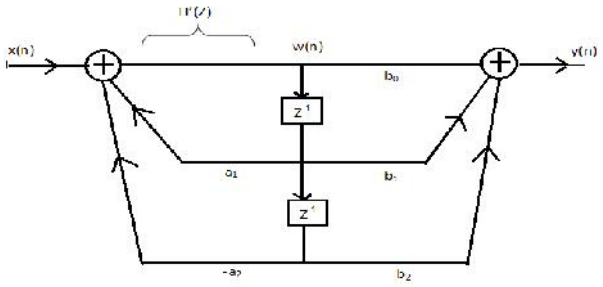
- From the transfer characteristics, we find that when overflow occurs, the sum of adder is set equal to the maximum value.

**********

**Signal Scaling:**

**Explain how reduction of round-off errors is achieved in digital filters.**

- Saturation arithmetic eliminates limit cycles due to overflow, but it causes undeniable signal distortion due to the non linearity of the clipper.
- In order to limit the amount of non linear distortion, it is important to scale input signal and unit sample response between input and any internal summing node in the system to avoid overflow.




**Realization of a second order IIR Filter**

- Let us consider a second order IIR filter as shown in the above figure. Here a scale factor $S_0$ is introduced between the input x(n) and the adder 1 to prevent overflow at the output of adder 1.

- Now the overall input-output transfer function is

Now the transfer function

$$H(z) = S_0 \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

$$= S_0 \frac{N(z)}{D(z)}$$

From figure

$$H'(z) = \frac{W(z)}{X(z)} = \frac{S_0}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{S_0}{D(z)}$$

$$W(z) = \frac{S_0 X(z)}{D(z)} = S_0 \, \mathrm{S}(z) \, \mathrm{x(Z)}$$

$$\text{Where } S(z) = \frac{1}{D(z)}$$

*we have*

$$w(n) = \frac{S_0}{2\pi} \int S(e^{j\omega}) X(e^{j\omega}) (e^{jn\omega}) \, \mathrm{d}\omega$$

$$\mathrm{w(n)}^2 = \frac{S_0^2}{2\pi^2} \left| \int S(e^{j\omega}) X(e^{j\omega}) (e^{jn\omega}) \, \mathrm{d}\omega \right|^2$$

Using Schwartz inequality

$$\mathrm{w(n)}^2 \le S_0^2 \left[ \int_{2\pi} \left| S(e^{j\omega}) \right|^2 \mathrm{d}\omega \right] \left[ \int_{2\pi} \left| X(e^{j\omega}) \right|^2 \mathrm{d}\omega \right]$$

Applying parsevals theorem

$$\mathrm{w(n)}^2 \le S_0^2 \sum_{n=0}^{\infty} x^2(n) \frac{1}{2\pi} \int_{2\pi} \left| S(e^{j\omega}) \right|^2 \mathrm{d}\omega$$

*if* $z = e^{j\omega}$ *then* $dz = j e^{j\omega} \, d\omega$

*which gives*

$$d\omega = \frac{dz}{jz}$$

By substituting all values

$$\mathrm{w(n)}^2 \le S_0^2 \sum_{n=0}^{\infty} x^2(n) \frac{1}{2\pi j} \int_c \left| S(z) \right|^2 \mathrm{z}^{-1} \mathrm{d}z$$

$$\mathrm{w(n)}^2 \le S_0^2 \sum_{n=0}^{\infty} x^2(n) \frac{1}{2\pi j} \int_c S(z) S(z^{-1}) \, \mathrm{z}^{-1} \mathrm{d}z$$

$$w^2(n) \le \sum_{n=0}^{\infty} x^2(n) \quad when$$

$$S_0^2 \frac{1}{2\pi j} \int_c S(z) S(z^{-1}) \, \mathrm{d}z = 1$$

Which gives us,

$$S_0^2 = \frac{1}{\frac{1}{2\pi j}\int_c S(z)S(z^{-1})z^{-1}dz}$$

$$= \frac{1}{\frac{1}{2\pi j}\int_c \frac{z^{-1}dz}{D(z)D(z^{-1})}}$$

$$S_0^2 = \frac{1}{I}$$

Where I=

$$\frac{1}{2\pi j}\int_c \frac{z^{-1}dz}{D(z)D(z^{-1})}$$

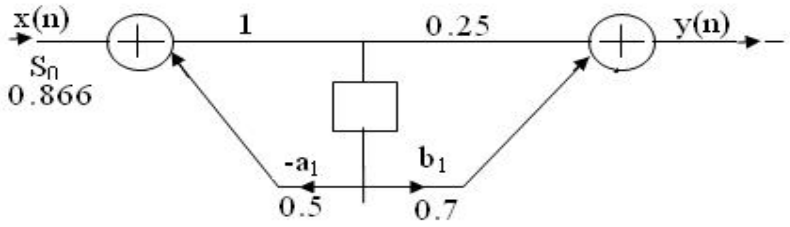
Note:

- Because of the process of scaling, the overflow is eliminated. Here so is the scaling factor for the first stage.
- Scaling factor for the second stage = $S_{01}$ and it is given by $S_{01}^2 = \frac{1}{S_0^2 I_2}$

  Where $I_2 = \frac{1}{2\pi j}\oint_c \frac{H_1(Z)H_1(Z^{-1})Z^{-1}}{D_2(Z)D_2(Z^{-1})}dZ$

**********

**For the given transfer function, $H(Z) = \frac{0.25 + 0.7Z^{-1}}{1 - 0.5Z^{-1}}$, find scaling factor so as to avoid overflow in the adder '1' of the filter.**




**Given:**

$D(Z) = 1-0.5Z^{-1}$

$D(Z^{-1}) = 1-0.5\,Z$

**Solution:**

$$I = \frac{1}{2\pi j}\oint_c \frac{1}{D(Z)D(Z^{-1})}\frac{dZ}{Z}$$

$$= \frac{1}{2\pi j}\oint_c \frac{1}{(1-0.5Z^{-1})(1-0.5Z)}\frac{dZ}{Z}$$

$$= \frac{1}{2\pi j}\oint_c \frac{Z}{(Z-0.5)}\frac{1}{(1-0.5Z)}\frac{dZ}{Z}$$

Residue of $\left.\frac{Z}{(Z-0.5)}\frac{1}{(1-0.5Z)}\right|_{Z=0.5} + 0$

I=1.3333

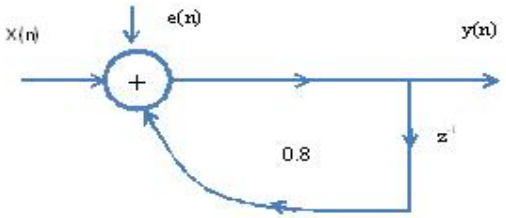
$$S_0 = \frac{1}{\sqrt{I}}$$

$$S_0 = \frac{1}{\sqrt{1.333}}$$

$= 0.866$

**********

**Consider the recursive filter shown in fig. The input x(n) has a range of values of ±100V, represented by 8 bits. Compute the variance of output due to A/D conversion process. (6)**




**Solution:**

Given the range is ±100V

The difference equation of the system is given by $y(n) = 0.8\,y(n-1) + x(n)$, whose impulse response h(n) can be obtained as

$h(n) = (0.8)^n u(n)$

$$quantization\ step\ size = \frac{range\ of\ the\ signal}{No.of\ quantization\ levels}$$

$$= \frac{200}{2^8}$$

$$= 0.78125$$

*Variance of the error signal*

$$\pm_e^2 = \frac{q^2}{12}$$

$$= \frac{(0.78125)^2}{12}$$

$$\pm_e^2 = 0.05086$$

*Variance of output*

$$\pm_{\phi}^2 = \pm_e^2 \sum_{n=0}^{\infty} h^2(n)$$

$$= (0.05086)\sum_{n=0}^{\infty} (0.8)^{2n}$$

$$= \frac{0.05086}{1-(0.8)^2} = 0.14128$$

**********

**The input to the system y(n)=0.999y(n-1)+x(n) is applied to an ADC. What is the power produced by the quantization noise at the output of the filter if the input is quantized to a) 8 bits b)16 bits. May-07**

Solution:

$y(n)=0.999y(n-1)+x(n)$

Taking z-transform on both sides

$Y(z)=0.999z^{-1}Y(z)+X(z)$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1-0.999z^{-1}}$$

$$H(z)H(z^{-1})z^{-1} = (\frac{z}{z-0.999})(\frac{z^{-1}}{z^{-1}-0.999})z^{-1}$$

$$= \frac{z^{-1}}{(z-0.999)(-0.999)(z-\frac{1}{0.999})}$$

$$= \frac{-0.001}{(z-0.999)(z-0.001)}$$

$$\left.\begin{matrix} output\ noise\ power\ due \\ to\ input\ quantization \end{matrix}\right\} \pm_{eoi}^{2} = \pm_{e}^{2} \frac{1}{2\pi j}\int_{c} H(z)H(z^{-1})z^{-1}dz$$

$$= \pm_{e}^{2} \sum_{i=1}^{N} Res\left[H(z)H(z^{-1})z^{-1}\right]\Big|_{z=p_i}$$

$$= \pm_{e}^{2} \sum_{i=1}^{N} \left[(z=p_i)H(z)H(z^{-1})z^{-1}\right]\Big|_{z=p_i}$$

Where $p_1, p_2, \ldots\ldots p_N$ are poles of $H(z)H(z^{-1})z^{-1}$, that lies inside the unit circle in z-plane.

$$\sigma_{eoi^2} = \sigma_{e^2}(z-0.999)(\frac{0.001}{(z-0.999)(z-0.001)})\Big|_{z=0.999}$$

$$= \sigma_{e^2} 500.25$$

a) b+1=8 bits(Assuming including sign bit)

$$\pm_{\varepsilon}^{2} = \frac{2^{2(7)}}{12}(500.25) = 2.544\times10^{-3}$$

b) b+1=16 bits

$$\pm_{\varepsilon}^{2} = \frac{2^{2(15)}}{12}(500.25) = 3.882\times10^{-8}$$

**********

**Find the effect of coefficient quantization on pole locations of the given second order IIR system, when it is realized in direct form I and in cascade form. Assume a word length of 4 bits through truncation.**

$$H(z) = \frac{1}{1-0.9z^{-1}+0.2z^{-2}}$$

**Solution:**

**<u>Direct form I</u>**

Let b=4 bits including a sign bit

$(0.9)_{10} = (0.111001...)_2$

*Integer part*

$$\frac{0.9\times 2}{1.8} \mapsto 1 \downarrow$$

$$\frac{0.8\times 2}{1.6} \mapsto 1$$

$$\frac{0.6\times 2}{1.2} \mapsto 1$$

$$\frac{0.2\times 2}{0.4} \mapsto 0$$

$$\frac{0.4\times 2}{0.8} \mapsto 0$$

$$\frac{0.8\times 2}{1.6} \mapsto 1$$

After truncation we get

$(0.111)_2=(0.875)_{10}$

$(0.2)_{10}=(0.00110\ldots)_2$

$$(0.2)_{10} = \frac{0.2\times 2}{0.4} \mapsto 0 \downarrow$$

$$\frac{0.4\times 2}{0.8} \mapsto 0$$

$$\frac{0.8\times 2}{1.6} \mapsto 1$$

$$\frac{0.6\times 2}{1.2} \mapsto 1$$

$$\frac{0.2\times 2}{0.4} \mapsto 0$$

After truncation we get

$(0.001)_2=(0.125)_{10}$

The system function after coefficient quantization is

$$H(z) = \frac{1}{1-0.875z^{-1}+0.125z^{-2}}$$

Now the pole locations are given by

$z_1=0.695$

$z_2=0.178$

If we compare the Poles of H(z) and $\overline{H}(z)$ we can observe that the poles of $\overline{H}(z)$ deviate very much from the original poles.

**Cascade form**

$$H(z) = \frac{1}{1-0.5z^{-1}(1-0.4z^{-1})}$$

$$(0.5)_{10} = (0.1000)_2$$

After truncation we get

$(0.100)_2=(0.5)_{10}$
After truncation we get
$(0.011)_2=(0.375)_{10}$

$$(0.4)_{10}=\frac{0.4\times 2}{0.8}$$
$$\mapsto \quad 0 \quad \downarrow$$
$$\frac{0.8\times 2}{1.6}$$
$$\mapsto \quad 0$$
$$\frac{0.6\times 2}{1.2}$$
$$\mapsto \quad 1$$
$$\frac{0.2\times 2}{0.4}$$
$$\mapsto \quad 1$$
$$\frac{0.4\times 2}{0.8}$$
$$\mapsto \quad 0$$

$(0.4)_{10}=(0.01100\ldots)_2$
The system function after coefficient quantization is

$$H(z)=\frac{1}{(1-0.5z^{-1})(1-0.375z^{-1})}$$

The pole locations are given by

$z_1=0.5$
$z_2=0.375$
on comparing the poles of the cascade system with original poles we can say that one of the poles is same and other pole is very close to original pole.
**********

**A LTI system is characterized by the difference equation y(n)=0.68y(n-1)+0.5x(n).**
**The input signal x(n) has a range of -5V to +5V, represented by 8-bits. Find the quantization step size, variance of the error signal and variance of the quantization noise at the output.**
**Solution:**
**Given**
Range R=-5V to +5V = 5-(-5) =10
Size of binary, B= 8 bits (including sign bit)
Quantization step size,

$$q=\frac{R}{2^8}=\frac{10}{2^8}=0.0390625$$

$$\text{var}\,iance\,of\ error\ signal,\pm_e^2=\frac{q^2}{12}=\frac{0.0390625^2}{12}=1.27116*10^{-4}$$

The difference equation governing the LTI system is
Y (n) =0.68y (n-1) +0.15x (n)
On taking z transform of above equation we get

$$Y(z) = 0.68z^{-1}Y(z) + 0.15X(z)$$

$$Y(z) - 0.68z^{-1}Y(z) = 0.15X(z)$$

$$Y(z)[1 - 0.68z^{-1}] = 0.15X(z)$$

$$\frac{Y(z)}{X(z)} = \frac{0.15}{1 - 0.68z^{-1}}$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{0.15}{1 - 0.68z^{-1}}$$

$$H(z)H(z^{-1})z^{-1} = \frac{0.15}{1 - 0.68z^{-1}} * \frac{0.15}{1 - 0.68z} * z^{-1}$$

$$H(z)H(z^{-1})z^{-1} = \frac{0.225z^{-1}}{\left(1 - \frac{0.68}{z}\right)(-0.68)\left(z - \frac{1}{0.68}\right)}$$

$$H(z)H(z^{-1})z^{-1} = \frac{-0.0331z^{-1}}{\left(\frac{z - 0.68}{z}\right)(z - 1.4706)} = \frac{-0.0331z^{-1}}{(z - 0.68)(z - 1.4706)}$$

Now, poles of H (z) H ($z^{-1}$) $z^{-1}$ are $p_1$=0.68, $p_2$=1.4706
Here, $p_1$=0.68 is the only pole that lies inside the unit circle in z-plane
Variance of the input quantization noise at the output.

$$\pm_{eoi}^2 = \pm_e^2 \frac{1}{2\circledR j} \int_c H(z)H(z^{-1})z^{-1}dz$$

$$\pm_{eoi}^2 = \pm_e^2 \sum_{i=1}^{N} \left[ \text{Res}\, H(z)H(z^{-1})z^{-1} \right]\Big|_{z=pi}$$

$$\pm_{eoi}^2 = \pm_e^2 \sum_{i=1}^{N} \left[ (z - p_i)H(z)H(z^{-1})z^{-1} \right]\Big|_{z=pi}$$

$$\pm_{eoi}^2 = \pm_e^2 (z - 0.68) * \frac{-0.0331}{(z - 0.68)(z - 1.4706)}\Big|_{z=0.68}$$

$$\pm_{eoi}^2 = \pm_e^2 * \frac{-0.0331}{(0.68 - 1.4706)}\Big| = 0.0419 \pm_e^2$$

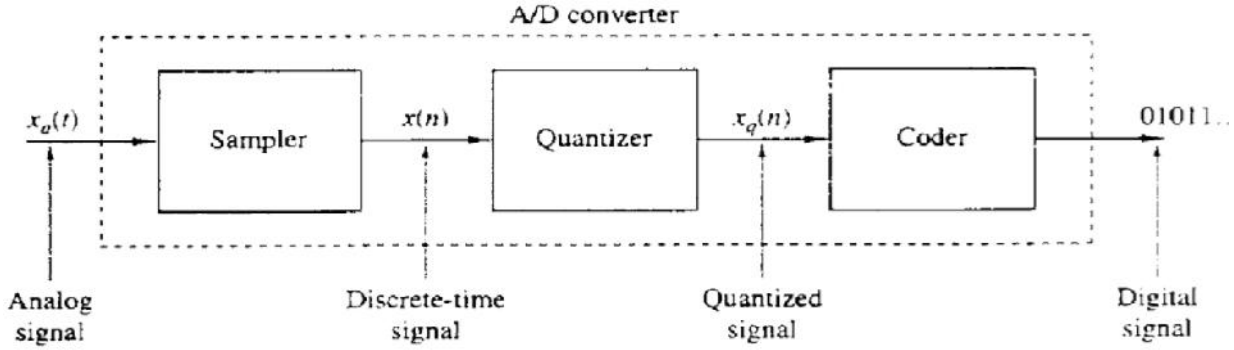
$$\pm_{eoi}^2 = 0.0419 * 1.2716 * 10^{-4}$$

$$\pm_{eoi}^2 = 5.328 * 10^{-6}$$

**Analog to digital conversion:**

## 10. Explain the ADC and DAC in detail.

A/D conversion has three process.




**Basic parts of an analog-to digital (A/D) converter**

1. Sampling

- Sampling is the conversion of a continuous- tome signal into a discrete-time signal obtained by taking the samples of continuous-time signal at discrete instants.
- Thus if $x_a(t)$ is the input to the sampler, the output is $x_a(nT) \equiv x(n)$, where T is called the sampling interval.

2. Quantisation

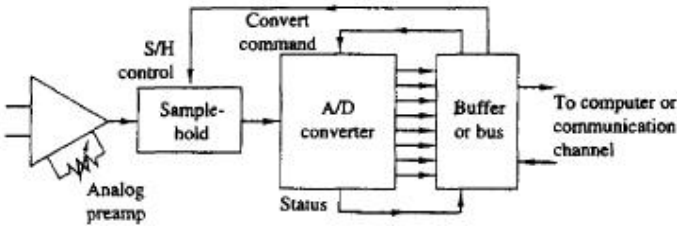
- The process of converting a discrete-time continuous amplitude signal into digital signal is called quantization.
- The value of each signal sample is represented by a value selected from a finite set of possible values.
- The difference between the unquantised sample x(n) and the quantized output $x_q(n)$ is called the quantization error or quantization noise.

$$e_q(n) = x_q(n) - x(n)$$

- To eliminate the excess bits either discard them by the process of truncation or discard them by rounding the resulting number by the process of rounding.
- The values allowed in the digital signals are called the quantization levels
- The distance $\Delta$ between two successive quantization levels is called the quantization step size or resolution.
- The quality of the output of the A/D converter is measured by the signal-to-quantization noise ratio.

3. Coding

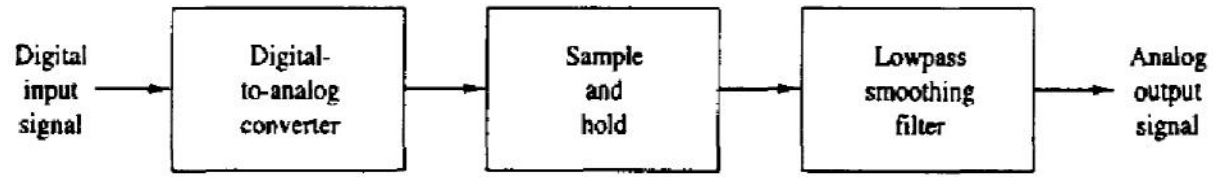
- In the coding process, each discrete value $x_q(n)$ is represented by a b-bit binary sequence.




**Block diagram of basic elements of an A/D Converter**

**Digital to analog conversion:**

- To convert a digital signal into an analog signal, digital to analog converters are used.



**Basic operations in converting a digital signal into an analog signal**

- The D/A converter accepts, at its input, electrical signals that corresponds to a binary word, and produces an output voltage or current that is proportional to the value of the binary word.
- The task of D/A converter is to interpolate between samples.
- The sampling theorem specifies the optimum interpolation for a band limited signal.
- The simplest D/A converter is the zero order hold which holds constant value of sample until the next one is received.
- Additional improvement can be obtained by using linear interpolation to connect successive samples with straight line segment.
- Better interpolation can be achieved y using more sophisticated higher order interpolation techniques.
- Suboptimum interpolation techniques result in passing frequencies above the folding frequency. Such frequency components are undesirable and are removed by passing the output of the interpolator through a proper analog filter which is called as post filter or smoothing filter.
- Thus D/A conversion usually involve a suboptimum interpolator followed by a post filter.

**********