CHAPTER 3

DIGITAL SIGNAL PROCESSING INSTRUMENTATION

3.1 Signals and Signal Processing:

Signals play an important role in our daily life. Examples of signals that we encounter frequently are speech, music, picture, and video signals. A signal is a function of independent variables such as time, distance, position, temperature, pressure. For example, speech and music signals represent air pressure as a function of time at a point in a space. A black and white picture is a representation of light intensity as function two spatial coordinates. The video signal in television consists of sequence of images, called frames, and is function of three variables; two spatial coordinates and time.

Most signals we encounter are generated by natural means. However, a signal can also be generated synthetically or by computer simulation. A signal carries information, and the objective of signal processing is to extract the information carried by the signal. The method of information extraction depends on the type of signal and the nature of information being carried by the signal. Thus, roughly speaking, signal processing is concerned with the mathematical representation of the signal and the algorithmic operation carried out on it to extract the information present. The representation of the signal can be in terms of basis functions in the domain of the original independent variable(s) or it can be in terms of basis functions in a transformed domain. Likewise, the information extraction process may be carried out in the original domain of the signal or in a transformed domain.

Depending on the nature of the independent variables and the value of the function defining the signal, various types of signals can be defined. For example, independent variable can be continuous or discrete. Likewise, the signal can either continuous or discrete function of the independent variables. Moreover, the signal can be either a real valued function or a complex-valued function.
A signal can be generated by a single source or by a multiple sources. In the former case, it is a scalar signal and in the latter case it is a vector signal, often called multi channel signal.

A one-dimensional (1-D) signal is a function of a single independent variable. A two dimensional (2-D) signal is a function of two independent variables. A multidimensional (M-D) signal is a function of more than two variables.

The speech signal is an example of a 1-D signal where the independent variable is time. An image signal, such as a photograph, is an example of a 2-D signal where the two independent variables are spatial variables. Each frame of black-and-white signal is a 2-D image signal that is a function of two discrete spatial variables, with each frame occurring sequentially at discrete instants of time. Hence, the black-and-white video signal must be considered as an example of three dimensional (3-D) signal where the three independent variables are the two spatial variables and time. A colour video signal is a three-channel signal composed of three 3-D signals representing the three primary colours: red, green, and blue (RGB). For transmission purpose, the RGB television signal is transformed into another type of three-channel signal composed of a luminance component and two chrominance components.

The value of the signal at specific values of the independent variable(s) is called its amplitude. The variation of the amplitude as a function of the independent variable(s) is called waveform.

For a 1-D signal, the independent variable is usually labeled as time. If the independent variable is continuous, the signal is called continuous-time signal. If the independent variable is discrete, the signal is called discrete-time signal. A -time signal is defined at every instant of time. On the other hand, a discrete-time signal is defined at the discrete instants of time, and hence, it is a sequence numbers.

A continuous time signal with a continuous amplitude is usually called an analog signal. A speech signal is an example of an analog signal. Analog signals are commonly encountered in our daily life and are usually generated by natural means. A discrete-time signal with discrete-valued
amplitudes represented by a finite number of digits is referred to as a digital signal. An example of a
digital signal is the digitized music signal stored in a CD-ROM disk. A discrete-time signal with
continuous-valued amplitudes is called a sampled-data signal. This last type of signal occurs in
switched-capacitor (SC) circuits. A digital signal is thus a quantized sampled-data signal. Finally, a
continuous-time signal with discrete-valued amplitudes has been referred to as a quantized boxcar
signal. Figure 3.1 illustrates the four types of signals.

The functional dependence of a signal in its mathematical representation is often explicitly
shown. For a continuous-time 1-D signal, the continuous independent variable is usually denoted by
\( t \), whereas for a discrete-time 1-D signal, the discrete independent variable is usually denoted by \( n \).
For example, \( u(t) \) represents a continuous-time 1-D signal and \( \{v[n]\} \) represents a discrete-time 1-D
signal. Each member, \( v[n] \), of a discrete-time signal is called a sampling many applications, a
discrete-time signal is generated from a parent continuous-time signal by sampling the latter at
uniform intervals of time. If the discrete instants of time at which a discrete-time signal is
defined are uniformly spaced, the independent discrete variable \( n \) can be normalized to assume integer
values.

In the case of continuous-time 2-D signal, the two independent variables are the spatial
coordinates, which are usually denoted by \( x \) and \( y \). For example, the intensity of a black-and-white
image can be expressed as \( u(x,y) \). On the other hand, a digitized image in 2-D discrete-time signal,
and its two independent variables are discretized spatial variables often denoted by \( m \) and \( n \). Hence, a
digitized image can be represented as \( u(x,y,t) \) where \( x \) and \( y \) denote the two spatial variables and \( t \)
denotes the temporal variable time. A colour video signal is a vector signal composed of three
signals representing the three primary colours: red, green, and blue.

\[
\begin{align*}
\text{red} & : r(x,y,t) \\
u(x,y,t) & = g(x,y,t) \\
\text{blue} & : b(x,y,t)
\end{align*}
\]
Fig 3.1 (a) A continuous-time signal, (b) a digital signal, (c) a sampled-data signal, and (d) a quantized boxcar signal.
There is another classification of signals that depends on certainty by which the signal can be described. A signal that can be uniquely determined by a well-defined process such as a mathematical expression or rule, or table look-up, is called a deterministic signal. A signal that is generated in a random fashion cannot be predicted ahead of time is called random signal. However, practical discrete-time systems employ finite word lengths for the storing of signals and the implementation of signal processing algorithms, it is necessary to develop tools for the analysis of finite length effects on the performance of discrete-time systems. To this end it has been found convenient to represent certain pertinent signals as random signals and employ statistical techniques for their analysis.

Some typical signal processing operations are reviewed in the following sections:

3.1.1 Typical Signal Processing Operations:

Various types of signal processing operations are employed in practice. In the case of analog signals, most signal processing operations are usually carried out in the time-domain, whereas, in the case of discrete-time signals, both time-domain and frequency domain operations are employed. In either case, the desired operations are implemented by a combination of some elementary operations. These operations are also usually implemented in real-time or near real-time, even though, in certain applications, they may be implemented off-line.

The mathematical foundations on which the signal analysis is being performed is briefly illustrated in the following sections - 3.2 Digital Signal Processing and 3.3 FFT Techniques.

3.2.1 INTRODUCTION TO DIGITAL SIGNAL PROCESSING [1]

Digital signal processing is an area of science and engineering that has developed rapidly over the past four decades. The rapid development has been a result of the significant advances in digital computer technology and integrated-circuit fabrication. The digital computers and associated digital hardware has reached many users and is less expensive due to mass manufacturing of modern computers and necessary hardware. From expensive and limited special usage of computers, a few decades ago, the present personal computers are available at low affordable price (even to common users in the form of Pcs). The rapid developments in the integrated -circuit technology, starting with medium scale integration (MSI comprising of a few thousand transistor equivalents) to present day
Ultra Very Large Scale Integration (Ultra VLSI comprising a few billion thousand transistor equivalents) that are produced in large quantities at very low cost are, smaller faster, and reliable. The digital computers prepared using these devices are relatively fast digital circuits that highly sophisticated digital systems capable of performing complex digital signal processing tasks. Using equally reliable operating systems and software packages, the analog systems have been completely replaced digital computer systems. The present work is perhaps a direct example of what is narrated above. However some sections of scientific community are still not ready to adopt the new digital systems. It is sufficient to state that many of the signal processing tasks that were conventionally performed by analog means are realized today by less expensive and more reliable digital computer firmware (the combination computer hardware and software).

However, signals with extremely wide bandwidths of real-time processing DSP is not a replacement. For such applications analog or optical signal processing is the only possible solution. Where necessary digital circuits of required speed are available, DSP are preferred. DSPs apart being cheaper and more reliable, have other advantages of flexibility. The digital processing hardware are usually programmable, and the programs can be easily altered as per to user requirements. Through software, one can more easily modify the signal processing functions to suit the performance of the hardware. Higher precision is possible using the digital hardware (and the associated software) compared to analog circuits and analog signal processing systems. Due to these advantages there has been an explosive growth in the field of DSP and associated applications.

3.2.2. Frequency Analysis of Signals and DSPs:

The Fourier transform is one of the several mathematical tools that is useful in the analysis and design of the so called LTI (Linear Time-Invariant) systems. These signal basically represent an equivalent signal of known wave shape and distribution (sinusoidal or a complex exponent). With such a decomposition, a signal is said to be represented in the frequency domain.

Most signals of practical interest [1] can be decomposed into a sum of sinusoidal signal components. For the class of periodic signals, such a decomposition is often called as a Fourier series. For the class of finite energy signals, the decomposition is called the Fourier transform. These decompositions are of extreme importance. The LTI of sinusoidal signal input is a sinusoid of the same frequency of different amplitude and phase. The linearity property of the LTI
systems implies that a linear sum of sinusoidal components, which produces a similar linear sum of sinusoidal components at the output (which may differ in amplitudes and phases of the input sinusoids). This characteristic behavior of LTI systems renders the sinusoidal decomposition. Many other decompositions of signals are possible [2]. The class of sinusoidal (or complex exponential) signals possess this desirable property in passing through an LTI system.

3.2.3 Tools Used in DSP Systems:

The tools required for frequency analysis of continuous periodic signals are briefly dealt. The best examples of periodic signals encountered in practice are: square waves, rectangular waves, triangular waves, sinusoids and complex exponents. The basic mathematical representation of any periodic signals is the well known Fourier series (a linear weighted sum harmonically related sinusoids or complex exponents). Jean Baptise Joseph Fourier (1738-1830), a French mathematician used trigonometric series expansions in describing the phenomenon of heat conduction and temperature distribution through bodies. The mathematical techniques so developed found application in a variety of problems (encompassing the fields of optics, vibrations, mechanical systems, system theory, and electromagnetics). This tool has been used for analysing functions encountered in DSP. A linear combination of harmonically related complex exponents of the form

\[ x(t) = \sum_{k = -\infty}^{\infty} c_k e^{j2\pi kf_0 t} \]  

represents a periodic signal with fundamental period \( T_p = 1/f_0 \). Hence we can think of exponential signals \( \{ e^{j2\pi kf_0 t} \quad k = 0, \pm 1, \pm 2, \pm 3 \ldots \} \) as the basic “building blocks” from which we can construct periodic signals of various types by proper fundamental frequency and coefficients \( \{c_k\} \). \( f_0 \) determines the fundamental period of \( x(t) \) and the coefficients \( \{c_k\} \) specify the shape of waveform. Suppose that we are given a periodic signal \( x(t) \) with period \( T_p \) we can represent the periodic signal by the series, called a Fourier series, where the fundamental frequency \( f_0 \) is selected to be the reciprocal of the given period \( T_p \). To determine the expression for the coefficients \( \{c_k\} \), we first multiply both sides of (3.2) by the complex exponential

\[ e^{j2\pi kf_0 t} \]
where \( l \) is an integer and then integrate both sides of resulting equation over a single period say from 0 to \( T_p \), are more generally, from \( t_0 \) to \( t_0 + T_p \), where \( t_0 \) is an arbitrary but mathematically convenient value [3]. Thus we obtain

\[
\int_{t_0}^{t_0 + T_p} x(t) e^{-j2\pi \frac{t}{T_p}} dt = \int_{t_0}^{t_0 + T_p} x(t) e^{-j2\pi \frac{k}{T_p} \cdot \frac{t}{T_p}} dt = \sum_{k=-\infty}^{\infty} c_k e^{-j2\pi k \cdot \frac{t_0}{T_p}} \cdots \tag{3.3}
\]

To evaluate the integral on the right-hand side of (3.3), we interchange the order of the summation and integration and combine two exponentials. Hence

\[
\sum_{k=-\infty}^{\infty} c_k \int_{t_0}^{t_0 + T_p} e^{j2\pi \frac{k}{T_p} \cdot \frac{t}{T_p}} dt = \sum_{k=-\infty}^{\infty} c_k \left[ e^{j2\pi \frac{k}{T_p} \cdot \frac{T_p}{T_p}} / e^{j2\pi \frac{k}{T_p} \cdot \frac{t_0}{T_p}} \right] \cdots \tag{3.4}
\]

for \( k \neq 1 \), the right-hand side of (3.4) evaluated at the lower and upper limits, \( t_0 \) and \( t_0 + T_p \), respectively, yields zero. On the other hand, if \( k = 1 \), we have

\[
\int_{t_0}^{t_0 + T_p} e^{j2\pi \frac{t}{T_p}} dt = \left[ \frac{t}{T_p} \right]_{t_0}^{t_0 + T_p}
\]

consequently, (3.3) reduces to

\[
\int_{t_0}^{t_0 + T_p} x(t) e^{-j2\pi \frac{t}{T_p}} dt = c_1 T_p \]

and therefore the expression for Fourier coefficients in terms of given periodic signal becomes

\[
c_1 \cdot \frac{1}{T_p} \int_{t_0}^{t_0 + T_p} x(t) e^{-j2\pi \frac{t}{T_p}} dt
\]

since \( t_0 \) is arbitrary, this integral can be evaluated over any interval of length \( T_p \), i.e., over any interval equal to the period of signal \( x(t) \). Consequently, the integral for Fourier series coefficients will be written as

\[
c_1 = \frac{1}{T_p} \int_{0}^{T_p} x(t) e^{-j2\pi \frac{t}{T_p}} dt \quad \cdots \tag{3.5}
\]
an important issue that arises in the representation of the period signal $x(t)$ by the Fourier series is whether or not the series converges to $x(t)$ for every value of $t$, i.e., if the signal $x(t)$ and its Fourier series representation

$$
\sum_{k = -\infty}^{\infty} c_k e^{j2\pi k t / T_0} \ldots \quad (3.6)
$$

are equal at every value at $t$. The so-called Dirichlet conditions guarantee that the eqn (3.6) will be equal to $x(t)$, except the values of $t$ for which $x(t)$ is discontinuous. At these values of $t$, it converges to the mid point of a the discontinuity.

$$
\int |x(t)| \, dt < \infty \quad \ldots \quad (3.7)
$$

$T_p$

all periodic signals of practical interest satisfies this conditions.

The weaker condition that signal has finite energy in one period,

$$
\int |x(t)|^2 \, dt < \infty \quad \ldots \quad (3.8)
$$

$T_p$

guarantees that the energy in the difference signal attains a value equal to zero through

$$
e(t) = x(t) - \sum_{k = -\infty}^{\infty} c_k e^{j2\pi k t / T_0} \ldots
$$

Although $x(t)$ and its Fourier series may not be equal for all values of $t$, eqn (3.7) implies eqn (3.8) but not vice versa, eqns (3.7), (3.8) and Dirichlet conditions are sufficient but not necessary conditions. This implies that the signals have Fourier series representation but do not fully satisfy these conditions.

3.2.4 POWER DENSITY SPECTRUM OF PERIODIC SIGNALS

A periodic signal has infinite energy and a finite average power, which is given as

$$
p_x = 1/T_p \int_{T_p} |x(t)|^2 \, dt < \infty \quad \ldots \quad (3.9)
$$
if we take the complex conjugate of (3.2) and substitute for \( x(t) \) in (3.9) we obtain

\[
p_x = \frac{1}{T_p} \int_{T_p}^\infty x(t) \sum_{k=-\infty}^{\infty} c_k e^{-j2\pi kF_0 t} dt = \sum_{k=-\infty}^{\infty} c_k^* \left[ \frac{1}{T_p} \int_{T_p}^\infty x(t) e^{-j2\pi kF_0 t} dt \right] \ldots (3.10)
\]

therefore, we have established the relation

\[
p_x = \frac{1}{T_p} \int_{T_p}^\infty |x(t)|^2 dt = \sum_{k=-\infty}^{\infty} |c_k|^2 \ldots (3.11)
\]

which is called Parseval's relation for signal. To illustrate the physical meaning of eqn (3.11) suppose that \( x_k(t) \) consists of single complex exponential of the form

\[x(t) = c_k e^{j2\pi kF_0 t}\]

it can be shown that all Fourier series coefficients except \( c_k \) are zero consequently, the average power in the signal is

\[p_x = |c_k|^2\]

It is obvious that \( |c_k|^2 \) represents the power in the \( k \)th harmonic component [4] of the signal hence the total average power in the periodic signal is simply the sum of the average power in all harmonics.

If we plot the \( |c_k|^2 \) as function of the frequencies \( kF_0 \), \( k = 0, \pm 1, \pm 2, \pm 3 \ldots \) the diagram that we obtain shows how the power periodic signal is distributed among the various frequency components. Figure 3.2 illustrated below is known as “power density spectrum” of the periodic signal \( x(t) \). Since the power in a periodic signal exits only at discrete values of frequencies (i.e., \( F = 0, \pm F_0, \pm 2F_0, \pm 3F_0, \ldots \)), the signal is said to have line spectrum. The spacing between two consecutive spectral line is equal to the reciprocal of the fundamental period \( T_p \), whereas the shape of the spectrum (known as the power distribution of the signal), depends on the time-domain characteristics of the signal. A typical power density spectrum is shown below:
Fig 3.2 - Frequency versus $|c_k|^2$, the modulus of the square Fourier coefficients (Power Spectral Density), a useful graph to identify the nature of the time distribution of signal.

The Fourier series coefficients $\{c_k\}$ are complex valued, that is, they can be represented as $c_k = |C_k| e^{i\theta_k}$ where $\theta_k = \angle c_k$. Instead of plotting the power density spectrum, we can plot the magnitude voltage spectrum $|C_k|$ and the phase spectrum $\theta_k$ as functions of frequency. Clearly, the power spectral density in the periodic signal is simply the square of the magnitude spectrum[5]. The phase information is totally destroyed (or does not appear) in the power spectral density. If the periodic signal is real valued, the Fourier series coefficients $C_k$ satisfy the condition [6]

$$C_{-k} = C_k^*$$

Consequently, $|C_k|^2 = |C_k^*|^2$. Hence the power spectrum is a symmetric function of frequency. This condition also means that the magnitude spectrum is symmetric (even function) about the origin and the phase spectrum is an odd function. As a consequence of the symmetry, it is sufficient to specify that the spectrum of a real [7] periodic signal for positive frequencies only. The total average power can be expressed as

$$P_x = c_0^2 + 2 \sum_{k=-\infty}^{\infty} |c_k|^2$$

...(3.12)
= a_0^2 + \frac{1}{2} \sum_{k=-\infty}^{\infty} (a_k^2 + b_k^2) \quad \ldots \quad (3.13)

which follows directly from the relationships among \{a_k\}, \{b_k\} and \{c_k\} coefficients in the Fourier series expressions.

3.2.5 ENERGY DENSITY SPECTRUM OF APERIODIC SIGNAL

The energy density spectrum of an aperiodic signal can also be evaluated. Let \(x(t)\) be any finite signal with Fourier transform \(X(F)\). Its energy is

\[
E_x = \int_{-\infty}^{\infty} |x(t)|^2 \, dt
\]

which in turn be expressed in terms \(X(f)\) as follows

\[
E_x = \int_{-\infty}^{\infty} x(t)x^*(t) \, dt
\]

\[
E_x = \int_{-\infty}^{\infty} |x(t)|^2 \, dt
\]

which is known as Parseval's relation for aperiodic, finite energy signals and expresses the principle of conservation of energy in the time and frequency domains. The spectrum \(X(F)\) of signal is in general, complex valued \([8]\) consequently, it is usually expressed in the polar form:

\[
X(F) = |x(F)| e^{i\Theta(F)}
\]

where \(X(F)\) is the magnitude spectrum and \(\Theta(F) = <X(F)\)

on the other hand, the quantity

\[
S_{xx}(F) = |x(F)|^2
\]

which is in the integrand, represent the distribution of energy in the signal as a function of frequency. Hence \(S_{xx}(F)\) is called energy density spectrum of \(X(t)\). The integral of \(S_{xx}(F)\) over all frequencies gives the total energy in the signal.

It can be easily shown that if the signal \(x(t)\) is real, then

\[
|x(-F)| = |x(F)| \quad \ldots \quad (3.14)
\]

\[
< x(-F) = - < x(F) \quad \ldots \quad (3.15)
\]

By combining eqn (3.14) & (3.15) we obtain
\( S_{xx}(-F) = S_{xx}(F) \) \hspace{1cm} (3.16)
in other words, the energy density spectrum of a real signal has even symmetry.

3.2.6 SAMPLING THEOREM REVISTED

To process a continuous time signal using signal processing technique, it is necessary to convert the signal into a sequence of numbers. This is usually done by sampling the analog signal, say \( X_a(t) \), periodically every \( T \) seconds to produce [9] a discrete time signal \( x(n) \) given by

\[
X(n) = X_a(nT) \quad -\infty < n < \infty \] \hspace{1cm} (3.17)

The relationship equ.(3.17) describes the sampling process in the time domain the sampling frequency \( F_s = 1/T \) must be selected large enough such that the sampling does not cause any loss of information. We investigate the sampling process by finding the relationship between the spectra of signals \( X_a(t) \) and \( X(n) \). If \( X_a(t) \) in an aperiodic signal with finite energy, its spectrum is given by the Fourier transform relation

\[
X_a(F) = \int X_a(t) e^{-j2\pi F t} dt \] \hspace{1cm} (3.18)

where as the signal \( X_a(t) \) can be recovered from its spectrum by the inverse Fourier transform

\[
X_a(t) = \int X_a(F) e^{j2\pi F t} dF \] \hspace{1cm} (3.19)

Note that utilization of all frequency components in the infinite range \(-\infty < F < \infty \) is necessary to recover the signal \( X_a(t) \) if the signal \( X_a(t) \) is not band limited.

The spectrum of discrete-time signal \( x(n) \) is obtained by sampling \( X_a(a) \) and its Fourier transform is equal to

\[
X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{j\omega n} \] \hspace{1cm} (3.20)
or, equivalently,

\[
X(f) = \sum_{n=-\infty}^{\infty} x(n)e^{j2\pi f n} \] \hspace{1cm} (3.21)

The sequence \( x(n) \) can be recovered [10] from its spectrum \( X(\omega) \) or \( X(f) \) by the inverse transform

\[
x(n) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{j\omega n} d\omega \] \hspace{1cm} (3.22)
\[ \int_{-\frac{1}{2}}^{\frac{1}{2}} X(f) e^{j2\pi fn} df \]

In order to determine the relationship between the spectra discrete time signal and the analog signal, we note that periodic sampling imposes a relationship between the independent variable \( t \) and \( n \) in the signals \( x_a(t) \) and \( x(n) \) respectively, that is

\[ t = nT = n/F, \quad \ldots \quad (3.24) \]

This relationship in the domain implies a corresponding relationship between frequency variables \( F \) and \( f \) in \( X_a(F) \) and \( X(f) \) respectively. Substitution of eqn (3.24) into eqn (3.19) yields

\[ x(n) = x_a(nT) = \int X_a(F) e^{j2\pi fn/F_s} dF \quad \ldots \quad (3.25) \]

On comparing eqn (3.23) with eqn (3.25) we obtain

\[ \int_{-\infty}^{\infty} X(f) e^{j2\pi fn} df = \int X_a(F) e^{j2\pi nF/F_s} dF \quad \ldots \quad (3.26) \]

The periodic signal sampling imposes a relationship between frequency variables \( F \) and \( f \) of the corresponding analog and discrete-time signals respectively. That is

\[ f = F / F_s, \quad \ldots \quad (3.27) \]

with the aid of (3.27), we can make simple change variable in eqn (3.26) and obtain the result

\[ \frac{F_s}{2} \int_{1/F_s}^{F_s/2} x(F/F_s) e^{j2\pi fnF/F_s} dF = \int_{-\infty}^{\infty} X_a(F) e^{j2\pi nF/F_s} dF \quad \ldots \quad (3.28) \]

we now turn our attention to the integral on the right-hand side of eqn (3.28). The integration range of this integral can be divided into an infinite number of intervals of width \( F_s \). Thus the integral over the finite range can be expressed as sum of integrals,

\[ \int_{-\infty}^{\infty} X_a(F) e^{j2\pi nF/F_s} dF = \sum_{k=-\infty}^{\infty} \int_{(k + 1/2)F_s}^{(k + 1/2)F_s} X_a(F) e^{j2\pi nF/F_s} dF \quad \ldots \quad (3.29) \]
we observe that $X_a(F)$ is in the frequency interval $(K-\frac{1}{2})F_s$ to $(K+\frac{1}{2})F_s$ and is identical to $X_a(F-kF_s)$ in the interval $-F_s/2$ to $F_s/2$. Consequently,

$$
\sum_{k=-\infty}^{\infty} \int_{-F_s/2}^{F_s/2} X_a(F) e^{j2\pi kF/F_s} dF = \int_{-\infty}^{\infty} \left[ \sum_{k=-\infty}^{\infty} X_a(F-kF_s) \right] e^{j2\pi kF/F_s} dF \ldots (3.30)
$$

where we had used the periodicity of the exponential, namely, $e^{j2\pi (F+kF_s)/F_s} = e^{j2\pi F/F_s}$.

This is the desired relationship between the spectrum $X(F/F_s)$ or $X(f)$ of the discrete time signal and the spectrum $X_a(F)$ of the analog signal [11]. The right hand side of eqn (3.31) or eqn (3.32) consists of a periodic repetition of the scaled spectrum $F_sX_a(F)$ with period $F_s$. This periodicity is necessary because of spectrum $X(f)$ or $X(F/F_s)$ of the discrete time signal is periodic with period $f_p = 1$ or $F_p = F_s$.

For example, suppose that the spectrum of a band limited analog signal is as shown in Fig 3.3a. The spectrum is zero for $|F| > B$. Now, if the sampling frequency $F_s$ is selected to be greater than $2B$, the spectrum $X(F/F_s)$ of the discrete time signal will appear as shown in Fig 3.3b. Thus if the sampling frequency $F_s$ is selected such that $F_s \geq 2B$ is the Nyquist rate, then

$$
X(F/F_s) = F_sX_a(F) \quad |F| \leq F_s/2 \ldots (3.31)
$$

In this case there is no aliasing and therefore, the spectrum of the discrete time signal is identical (with in the scale factor $F_s$) to the spectrum of the analog signal, with in the fundamental frequency range $|F| \leq F_s/2$ or $|F| \leq F_s/2$ or $|f| \leq 1/2$.

On the other hand, if the sampling frequency $F_s$ is selected such that $F_s < 2B$, the periodic continuation of $X_a(F)$ results in spectral overlap, as illustrated in the Fig 3.3c and 3.3d. Thus the spectrum $X(F/F_s)$ of the discrete time signal contains aliased frequency components of the analog signal spectrum $X_a(F)$. The end result is that the aliasing which occurs prevents us from recovering the original signal $X_a(t)$ from the samples.

It is now possible to reconstruct the original analog signal from the samples $X(n)$ by considering

$$
X_a(F) = \begin{cases} \frac{1}{F_s} X(F/F_s), & |F| \leq F_s/2 = 0 \\ 0, & |F| > F_s/2 \end{cases} \ldots (3.32)
$$

and the Fourier transform relationship equ (3.21) yields
Fig 3.3 Sampling of an analog bandlimited signal and aliasing of spectral components.
The reconstruction of the original signal is hence possible and useful.

\[ X(F/F_s) = \sum X(n) e^{j2\pi Fn/F_s} \] ...

(3.33)

3.3 FUNDAMENTAL OF FAST FOURIER TRANSFORM (FFT)

The frequency analysis on a time domain signal function is performed using the well known methods of Discrete Fourier Transform (DFT). DFT is a powerful computational tool. It plays an important role in many applications of digital signal processing including linear filtering, correlation analysis, and spectrum analysis. A major reason for its importance is the existence of efficient algorithms for computing the DFT.

A few details of the algorithms for evaluating the DFT are presented. Two different approaches are described. One is divide and conquer approach in which a DFT of size N is computed (when the size N is a power of 2 or 4. The second approach is based on the formulation of the DFT as a linear filtering operation on the data. This second approach leads to two algorithms, the Goertzel algorithm and the chirp-z transform algorithm for computing the DFT via linear filtering of the data sequence.

3.3.1 COMPUTATION METHODS OF THE DFT AND FFT ALGORITHMS

Efficient methods of computing DFT are presented here. In view of the importance of the DFT in various digital signal processing applications (linear filtering, correlation analysis and spectrum analysis) this topic has received considerable attention by many mathematicians, engineers and applied scientists[5]. The computational problem for the DFT is to compute the sequence \( \{X(k)\} \) of N complex numbers using given sequence of data \( \{x(n)\} \) of length N, which on mathematical formulation turns out as

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

(3.34)

Where \( W_N = e^{-j2\pi / n} \) ...

(3.35)

in general, the data sequence \( x(n) \) is also assumed to be complex valued. Similarly, the IDFT becomes

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

(3.36).
Since the DFT and IDFT involve basically the same type of computations, our discussion of efficient computational algorithms for the DFT also applies to the IDFT. For each value of $X(k)$, direct computation of $X(k)$ involves $N$ complex multiplications ($4N$ real multiplications) and $N - 1$ complex additions ($4N - 2$ real additions). Consequently, to compute all $N$ values of the DFT requires $N^2$ complex multiplications and $N^2 - N$ complex additions.

Direct computation of the DFT is basically inefficient primarily because it does not exploit the symmetry and periodicity properties of the phase factor $W_N$. These two properties are:

Symmetry property: $W_N^{k+N/2} = -W_N^k$ ... (3.37)

Periodicity property: $W_N^{k-n} = W_N^k$ ... (3.38)

The computationally efficient algorithms described in this section, known collectively as Fast Fourier Transform (FFT) algorithms, exploit these two basic properties.

### 3.3.2 DIRECT COMPUTATION OF THE DFT

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

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

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

The direct computation of eqn(3.39) and (3.40) requires:

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

These operations [12] are typical of DFT computational algorithms. The operations in items 2 and 3 result in the DFT values $X_R(k)$ and $X_I(K)$. The indexing and addressing operations are necessary to fetch the data $x(n)$, $0 \leq n \leq N-1$, and the diphase factors and to store the results. The variety of DFT algorithms optimize each of these computational processes in a different way.

### 3.3.3 DIVIDE-AND-CONQUER APPROACH TO COMPUTATION OF THE DFT:
The development of computationally efficient algorithms for the DFT is made possible if we adopt a divided-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 computationally efficient algorithms known collectively as FFT algorithms.

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

\[ N = LM \quad \ldots (3.40a) \]

The assumption that N is not a prime number is not restrictive, since we can pad any sequence with zeros to ensure a factorization of the form (3.40a). Now the sequence \( x(n), 0 \leq n \leq N - 1 \), can be stored in either a one-dimensional array indexed by \( n \) or as a two-dimensional array indexed by \( l \) and \( m \), where \( 0 \leq l \leq L-1 \) and \( 0 \leq m \leq M-1 \) as illustrated in Table 3.1.

Table 3.1 Two Dimensional Data Array For Storing the Sequence \( x(n), 0 \leq n \leq N-1 \)

<table>
<thead>
<tr>
<th>( n )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>\ldots</th>
<th>( N-1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( X(0) )</td>
<td>( X(1) )</td>
<td>( X(2) )</td>
<td>\ldots</td>
<td>( X(N-1) )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>row index</th>
<th>( m )</th>
<th>column index</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>( x(0,0) )</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>( x(1,0) )</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>( x(2,0) )</td>
</tr>
<tr>
<td>( L-1 )</td>
<td>( L-1,0 )</td>
<td>( x(L-1,1) )</td>
</tr>
</tbody>
</table>
Note that $l$ is the row index and $m$ is the column index. Thus the sequenced $x(n)$ can be stored in a rectangular array in a variety of ways, each of which depends on the mapping of index $n$ to the indexes $(l,m)$.

For example, suppose that we select the mapping

$$n = Ml + m \quad \ldots \quad (3.41)$$

This leads to an arrangement in which the first row consists of the first $M$ elements of $x(n)$, the second row consists of the next $M$ elements of $x(n)$, and so on, as illustrated in Table 3.1. On the other hand, the mapping

$$n = l + mL \quad \ldots \quad (3.42)$$

stores the first $L$ elements of $x(n)$ in the first column, the next $L$ elements in the second column, and so on, as illustrated in Table 3.1 as a two dimensional array.

A similar arrangement can be used to store the computed DFT values. In particular, the mapping is from the index $k$ to a pair of indices $(p,q)$, where $0 \leq p \leq L - 1$ and $0 \leq q \leq M - 1$. If we select the mapping

$$k = Mp + q \quad \ldots \quad (3.43)$$

the DFT is stored on a row-wise basis, where the first row contains the first $M$ elements of the DFT $X(k)$, the second row contains the next set of $M$ elements, and so on. On the other hand, the mapping

$$k = qL + p \quad \ldots \quad (3.44)$$

results in a column-wise storage of $X(k)$, where the first $L$ elements are stored in the first column, the second set of $L$ elements are stored in the second column, and so on.

Now suppose that $x(n)$ is mapped into the rectangular array $x(l,m)$ and $X(k)$ is mapped into a corresponding rectangular array $X(p,q)$. Then the DFT can be expressed as a double sum over the elements of the rectangular array multiplied by the corresponding phase factors. To be specific let us adopt a column-wise mapping for $x(n)$ given by (3.42) and the row-wise mapping for the DFT given by equ(3.44). Then

$$X(p,q) = \sum_{m=0}^{M-1} \sum_{l=0}^{L-1} x(l,m) W_N^{(Mp + q)(Ml + 1)} \quad \ldots \quad (3.45)$$

But
\[ W_{N}^{(\text{M} p q(x, M x, L))} = W_{N}^{M p q} W_{N}^{M L q} W_{N}^{M p L} W_{N}^{L q} \]  \hspace{1cm} (3.46)

However \( W_{N}^{M p q} = 1 \), \( W_{N}^{M L q} = W_{N L}^{M q} = W_{N}^{M q} \)

and \( W_{N}^{M p L} = W_{N M}^{L p} = W_{L}^{L p} \) \hspace{1cm} (3.47)

with these simplifications eqn(3.45) can be expressed as

\[
X(p, q) = \sum_{l=0}^{M-1} \left\{ W_{N}^{l q} \left[ \sum_{m=0}^{M-1} x(l, m) W_{M}^{m q} \right] \right\} W_{L}^{l p} \hspace{1cm} (3.48)
\]

The expression in eqn(3.48) involves computation of DFTs of length \( M \) and length \( L \). To elaborate, let us subdivide the computation into three steps

1. First, we compute the \( M \) - DFTs

\[
F(l, q) = \sum_{m=0}^{M-1} x(l, m) W_{M}^{m q}, \hspace{0.5cm} 0 \leq q \leq M-1 \]

for each other rows \( l = 0, 1, 2 \ldots L-1 \).

2. Second we compute, a new rectangular array \( G(l, q) \) defined as

\[
G(l, q) = W_{N}^{k q} F(l, q), \hspace{0.5cm} 0 \leq l \leq L-1 \hspace{1cm} 0 \leq q \leq M-1 \]

\hspace{1cm} (3.50)

3. Finally we compute the \( L \) - DFTs

\[
X(p, q) = \sum_{l=0}^{L-1} G(l, q) W_{L}^{l p}, \hspace{1cm} (3.51)
\]

for each column \( q = 0, 1, \ldots M-1 \), of the array \( G(l, q) \)

On the surface it may appear that the computational procedure outlined above is more complex than the direct computation of the DFT. However, let us evaluate the computational complexity of (3.48). The first step involves the computation of \( L \) DFTs, each of \( M \) points. Hence
this step requires \( LM^2 \) complex multiplications and \( LM(M-1) \) complex additions. The second step requires \( LM^2 \) complex multiplications. Finally, the third step in the computation requires \( ML^2 \) complex multiplications and \( ML(M-1) \) complex additions. Therefore, the computational complexity is

\[
\begin{align*}
\text{Complex multiplications:} & \quad N(M + L + 1) \\
\text{Complex additions:} & \quad N(M + L - 2) \quad \ldots \quad (3.52)
\end{align*}
\]

where \( N = ML \). Thus the number of multiplications has been reduced from \( N^2 \) to \( N(M + L + 1) \) and the number of additions has been reduced from \( N(N-1) \) to \( N(M+L-2) \).

For example, suppose that \( N=1000 \) and we select \( L=2 \) and \( M=500 \). Then, instead of having to perform \( 10^6 \) complex multiplications via direct computation of the DFT, this approach leads to 503,000 complex multiplications. This represents a reduction by approximately a factor of 2. The number of additions is also reduced by about a factor of 2.

When \( N \) is a highly composite number, that is, \( N \) can be factored into a product of prime numbers of the form

\[
N = r_1 r_2 \ldots r_v \quad \ldots \quad (3.53)
\]

then the decomposition above can be repeated \( (v-1) \) more times. This procedure results in smaller DFTs, which in turn, leads to a more efficient computational algorithm.

In effect, the first segmentation of the sequence \( x(n) \) into a rectangular array of \( Md \) columns with \( L \) elements in each column resulted in DFTs of sizes \( L \) and \( M \). Further decomposition of the data in effect involves the segmentation of each row (or column) into smaller rectangular array which result in smaller DFTs. This procedure terminates when \( N \) is factored into its prime factors.

To summarize, the algorithm that we have introduced involves following computations:

Algorithm 1

1. Store the signal column -wise.
2. Compute the \( M \)-point DFT of each row.
3. Multiply the resulting array by the face factors \( W_{k}^{l} \).
4. Compute the \( L \)-point DFT of each column.
5. Read the resulting array row-wise.

An additional algorithm with a similar computational structure can be obtained if the input signal is stored row-wise and the resulting transformation is column-wise. In this case we select as
\[ n = M l + m \]
\[ k = q L + p \]  ... (3.54)

This choice of indices leads to the formula for the DFT in the form

\[
X(p,q) = \sum_{m=0}^{N-1} \sum_{l=0}^{L-1} x(l,m) W_N^{pm} W_L^{pl} W_N^{qm} \]  ... (3.55)

\[
X(p,q) = \sum_{m=0}^{N-1} W_M^{qm} \left( \sum_{l=0}^{L-1} x(l,m) W_L^{pl} \right) W_N^{qm} \]  ... (3.56)

Thus we obtain a second algorithm.

**Algorithm 2**

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

The two algorithms given above have the same complexity. However, they differ in the arrangement of the computations. In the following sections we exploit the divide-and-conquer approach to derive fast algorithms when the size of the DFT is restricted to be a power of 2 or a power of 4.

### 3.3.4 APPLICATIONS OF FFT ALGORITHMS

The FFT algorithms described in the preceding section [12] find application in a variety of areas, including linear filtering, correlation, and spectrum analysis. Basically, the FFT algorithm is used as an efficient means to compute the DFT and the IDFT.

In this section we consider the use of the FFT algorithm in linear filtering and in the computation of the cross correlation of two sequences. We illustrate how to enhance the efficiency of the FFT algorithm by forming complex-valued sequences from real-valued sequences prior to the computation of the DFT.

**Efficient Computation of the DFT of two Real Sequences**
The FFT algorithm is designed to perform complex multiplications and additions, even though the input data may be real valued. The basic reason for this situation is that the phase factors are complex and hence, after the first stage of the algorithm, all variables are basically complex-valued.

In view of the fact that the algorithm can handle complex-valued input sequences, we can exploit this capability in the computation of the DFT of two real-valued sequences.

Suppose that \( x_1(n) \) and \( x_2(n) \) are two real-valued sequences of length \( N \), and let \( x(n) \) be a complex-valued sequence defined as
\[
x(n) = x_1(n) + jx_2(n) \quad 0 \leq n \leq N-1 \quad \text{... (3.57)}
\]
The DFT operation is linear and hence the DFT of \( x(n) \) can be expressed as
\[
X(k) = X_1(k) + jX_2(k) \quad \text{... (3.58)}
\]
The sequences \( x_1(n) \) and \( x_2(n) \) can be expressed in terms of \( x(n) \) as follows:
\[
x_1(n) = (x(n) + x(n)^*)/2
\]
\[
x_2(n) = (x(n) - x(n)^*)/2 \quad \text{... (3.59)}
\]

hence the DFTs of \( x_1(n) \) and \( x_2(n) \) are
\[
X_1(k) = \frac{1}{2} \{ \text{DFT}[x(n)] + \text{DFT}[x^*(n)] \}
\]
\[
X_2(k) = \frac{1}{2} \{ \text{DFT}[x(n)] - \text{DFT}[x^*(n)] \} \quad \text{... (3.60)}
\]

OR
\[
X_1(k) = \frac{1}{2} \{ x(k) + [x^*(N-k)] \}
\]
\[
X_2(k) = \frac{1}{2} \{ x(k) - [x^*(N-k)] \} \quad \text{... (3.61)}
\]

Thus, by performing a single DFT on the complex-valued sequence \( x(n) \), we have obtained the DFT of the two real sequences with only a small amount of additional computation that is involved in computing \( X_1(k) \) and \( X_2(k) \) from \( X(k) \) by use of eqn (3.50) and (3.51).

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

Suppose that \( g(n) \) is a real-valued sequence of 2\( N \) points. We now demonstrate how to obtain the 2\( N \)-point DFT of \( g(n) \) from computation of one \( N \)-point DFT involving complex-valued data. First, we define
\[
x_1(n) = g(2n)
\]
\[
x_2(n) = g(2n+1) \quad \text{... (3.62)}
\]
Fig 3.4 (a) Input signal, (b) output of a lowpass filter with a cutoff at 80 Hz, (c) output of a highpass filter with a cutoff at 150 Hz, (d) output of a bandpass filter with cutoffs at 80 Hz and 150 Hz, and (e) output of a bandstop filter with cutoffs at 80 Hz and 150 Hz.
Thus we have subdivided the 2N-point real sequence into two N-point real sequences. Now we can apply the method described in the preceding section.

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

Thus we have computed the DFT of a 2N-point real sequence from one N-point DFT and some additional computation as indicated above. The FFT method is hence useful to evaluate the properties of variable signals even when signals are measured for a relatively short duration of time.

3.3.6 Filtering:

One of the most widely used complex signal processing operations is filtering. Filtering is used to pass certain frequency components in a signal through the system without any distortion and to block other frequency components. The system implementing this operation is called filter. Range of frequencies that is allowed to pass through the filter is called passband, the range of frequencies that is blocked by the filter is called the stopband. Various types of filters can be defined, depending on the nature of the filtering operation. In most cases, the filtering operation for analog signals is linear and is described by the convolution integral

$$y(t) = \int_{-\infty}^{\infty} y(t-\tau) x(\tau) d\tau$$  \hspace{1cm} (3.63)

where $x(t)$ is the input signal and $y(t)$ is the output of the filter characterized by an impulse response $h(t)$.

A lowpass filter passes all low-frequency components below a certain specification $f_c$, called cutoff frequency, and blocks all high frequency components above $f_c$. A highpass filter passes all high-frequency components above a certain cutoff frequency $f_c$ and blocks all low-frequency components below $f_c$. A bandpass filter passes all frequency components between two cutoff frequencies $f_{c1}$ and $f_{c2}$, and blocks all frequency components below the frequency $f_{c1}$ and above the frequency $f_{c2}$. A bandstop filter blocks all frequency components between two cutoff frequencies $f_{c1}$ and $f_{c2}$, and passes all frequency components below the frequency $f_{c1}$ and above the frequency $f_{c2}$.

Figure 3.4(a) shows a signal composed of three sinusoidal components of frequencies 50Hz, 110Hz
and 210 Hz, respectively. Fig 3.4(b) to (e) show the results of the above four types of filtering operations with appropriately chosen cutoff frequencies.

A stopband filter designed to block a single frequency component is called notch filter. A multiband filter has more than one passband and more than one stopband. A comb filter is designed to block frequencies that are integral multiples of a low frequency.

A signal may get corrupted unintentionally by an interfering signal called interference or noise. In many applications the desired signal occupies a low-frequency band from dc to some frequency \( f_L \) Hz, and it is corrupted by a high-frequency components above \( f_H \) Hz with \( f_H > f_L \). In such cases, the desired signal can be recovered from the noise corrupted signal by passing the latter through a low pass filter with a cutoff frequency \( f_c \) where \( f_L < f_c < f_{\text{in}} \). A common source of noise is power lines radiating electric fields. The noise generated by power lines radiating sinusoidal signal corrupting the desired signal and can be removed by passing the corrupted signal through a notch filter with a notch frequency at 60 Hz.

### 3.3.7 Butterworth Approximation:

The magnitude-square response of an analog lowpass Butterworth filter \( H_4(s) \) of \( N \)th order is given by

\[
| H_a(j\Omega)|^2 = \frac{1}{1 + (\Omega / \Omega_c)^{2N}} \quad (3.64)
\]

It can be easily shown that the first \( 2N-1 \) derivatives of \( |H_a(j\Omega)|^2 \) at \( \Omega = 0 \) are equal to zero, and as a result, the Butterworth filter is said to have a maximally flat magnitude at \( \Omega = 0 \). The gain of the Butterworth filter in dB is given by

\[
\zeta(\Omega) = 10 \log_{10} |H_a(j\Omega)|^2 \text{ dB} \quad (3.65)
\]

At dc, i.e., at \( \Omega = 0 \), the gain in dB is equal to zero, and at \( \Omega = \Omega_c \), the gain is...
\[ \zeta(\Omega_c) = 10 \log_{10}(1/2) = -3.0103 = -3 \text{ dB} \quad \ldots \ (3.66) \]

and therefore, \( \Omega_c \) is often called the 3-dB cutoff frequency. Since the derivative of the companding squared-magnitude response, or equivalently, of the magnitude response is always negative for positive values of \( \Omega \), the magnitude response, is monotonically decreasing with increasing \( \Omega \). For \( \Omega \gg \Omega_c \), the squared-magnitude function can be approximated by

\[ |H(j\Omega)|^2 \approx 1/(\Omega/\Omega_c)^{2N} \quad \ldots \ (3.67) \]

The gain \( \zeta(\Omega_2) \) in dB at \( \Omega_2 = 2\Omega_1 \), with \( \Omega_1 \gg \Omega_c \) is given by

\[ \zeta(\Omega_2) = -20 \log_{10}(\Omega_2/\Omega_c)^{2N} = \zeta(\Omega_1) - 6N \text{ dB} \quad \ldots \ (3.68) \]

where \( \zeta(\Omega_1) \) is given by the gain in dB at \( \Omega_1 \). As a result the gain roll-off per octave in the stopband decreases by 6 dB, or equivalently, by 20 dB per decade of an increase of the filter order by one. In other words, the passband and stopband behaviors of the magnitude response improve with a corresponding decrease in the transition band as the filter order \( N \) increases. A plot of the magnitude response of the normalized Butterworth lowpass filter with \( \Omega_c = 1 \) for some typical values of \( N \) is shown in fig 3.5.

The two parameters completely characterizing a Butterworth filter are therefore the 3-dB cutoff frequency \( \Omega_c \) and the order \( N \). These are determined from specified passband edge \( \Omega_p \), the minimum passband magnitude \( 1/\sqrt{1 + \varepsilon^2} \) the stopband edge \( \Omega_s \), and the maximum stopband ripple \( 1/A \). Therefore can we get

\[ |H(j\Omega_p)|^2 = 1/\left\{1 + (\Omega_p/\Omega_c)^{2N}\right\} = 1/(1+\varepsilon^2) \quad \ldots \ (3.69) \]

\[ |H(j\Omega_s)|^2 = 1/\left\{1 + (\Omega_s/\Omega_c)^{2N}\right\} = 1/A^2 \quad \ldots \ (3.70) \]

Solving the above we arrive at the expression for the order \( N \) as
\[ N = \log_{10}(\frac{(A^2 - 1)/\varepsilon^2}{2\log_{10}(\Omega_p/\Omega_s)}) = \frac{\log_{10}(1/k)}{\log_{10}(1/k)} \]

Since the order \( N \) of the filter must be an integer, the value of \( N \) computed using the above expression is rounded up to the next higher integer. This value of \( N \) can be used next in either Eqn (3.63) or Eqn (3.64) to solve for the 3-dB cutoff frequency \( \Omega_c \). If it is used in Eqn (3.63), the passband specification is met exactly, whereas the stopband specification is exceeded. On the other hand, if it is used in Eqn (3.64), the stopband specification is met exactly, whereas the passband specification is exceeded.

The expression for the transfer function of the Butterworth lowpass filter is given by

\[ H_s(s) = \frac{C}{\{D_N(s)\}} = \Omega_c^N/\{s^N + \sum_{i=0}^{N-1} d_i s^i\} = \Omega_c^N/\{\prod_{i=1}^{N} (s-p_i)\} \quad (3.71) \]

where

\[ p_i = \Omega_c e^{i\pi(N+1-2i)/2N}, \quad i = 1, 2, \ldots, N. \quad (3.72) \]

The denominator \( D_N(s) \) of Eqn (3.65) is known as the Butterworth polynomial of order \( N \) and is easy to compute. These polynomials have been tabulated for easy design reference. The analog lowpass Butterworth filters can be readily designed using MATLAB.

3.3.8 Chebyshev Approximation:

In this case, the approximation error, defined as the difference between the ideal brickwall characteristic and the actual response, is minimized over a prescribed band frequencies. In fact the magnitude error is equiripple in the band. There are two types of Chebyshev transfer functions. In the Type 1 approximation, the magnitude characteristic is equiripple in the passband and monotonic in the stopband, whereas in the Type 2 approximation, the magnitude response is monotonic in the passband and equiripple in the stopband.

3.3.9 Type 1 Chebyshev Approximation:

The type 1 Chebyshev transfer function \( H_s(s) \) has a magnitude response given by
Fig 3.5 Typical Butterworth lowpass filter responses.

Fig 3.6 Typical Type 1 Chebyshev lowpass filter responses with 1 dB passband ripple.
where $T_N(\Omega)$ is the Chebyshev polynomial of order $N$:

\begin{align*}
T_N &= \cos(N \cos^2 \Omega), \quad \text{if } |\Omega| \leq 1, \quad \text{(3.74)} \\
T_N &= \cosh(N \cosh^{-1} \Omega), \quad \text{if } |\Omega| > 1.
\end{align*}

The above polynomial can also be derived via a recurrence relation given by

\begin{equation}
T_r(\Omega) = 2 \Omega T_{r-1}(\Omega) - T_{r-2}(\Omega), \quad \text{for } r \geq 2, \quad \text{(3.75)}
\end{equation}

with $T_0(\Omega) = 1$ and $T_1(\Omega) = \Omega$.

Typical plots of the magnitude response of the type 1 Chebyshev lowpass filter are shown in Fig 3.6 for three different values of filter order $N$ with the same passband ripple. From these plots it is seen that the square-magnitude response is equiripple between $\Omega=0$ and $\Omega=1$, and it decreases monotonically for all $\Omega > 1$.

3.3.10 Digital Filter Design:

An important step in development of a digital filter is the determination of a realizable transfer function $G(z)$ approximating the given frequency response specifications. If an IIR filter is desired, it is also necessary to ensure that $G(z)$ is stable. The process of deriving the transfer function $G(z)$ is called digital filter design. After $G(z)$ has been obtained, the next step is to realize it in the form of a suitable filter structure.

3.4.1 Preliminary Conditions:

There are two major issues that need to be answered before one can develop the digital transfer function $G(z)$. The first and foremost issue is the development of a reasonable filter frequency response specification from the requirements of the overall system in which the digital filter is to be employed. The second issue is to determine whether a finite-impulse response (FIR) or an infinite-impulse response (IIR) digital filter is to be designed.
3.4.2 Digital Filter Specifications:

As in the case of the analog filter, either the magnitude or the phase (delay) response is specified for the design of a digital filter for most applications. In some situations, the unit sample response or the step response may be specified. In most practical applications, the problem of interest is the development of a realizable approximation to a given magnitude response specification. There are four basic types of filters, namely, ideal lowpass filter, ideal highpass filter, ideal bandpass filter and ideal bandstop filter, whose magnitude responses are shown in Fig 3.7. Since the impulse response corresponding to each of these is noncasual and of infinite length, these ideal filters are not realizable. One way to develop a realizable approximation to these filters would be to truncate the impulse response. The magnitude response of FIR lowpass filter obtained by truncating the impulse response of the ideal lowpass filter does not have sharp transition from passband to stopband but, rather, exhibits a gradual "roll-off".

Thus, as in the case of the analog filter design problem, the magnitude response specifications of a digital filter in the passband and in the stopband are given with some acceptable tolerances. In addition, a transition band is specified between the passband and the stopband to permit the magnitude to drop off smoothly. For example, the magnitude \(|G(e^{j\omega})|\) of a lowpass filter may be given as shown in Fig 3.8. As indicated in the figure, in the passband defined by \(0 \leq \omega \leq \omega_p\), we require that the magnitude approximates unity with an error of \(\pm \delta_p\), i.e.,

\[
1 - \delta_p \leq |G(e^{j\omega})| \leq 1 + \delta_p, \quad \text{for } |\omega| \leq \omega_p
\]  

(3.76)

In the stopband, defined by \(\omega_s\), we require the magnitude approximate zero with an error of \(s\), i.e.,

\[
|G(e^{j\omega})| \leq \delta_s \quad \text{for } \omega_s \leq |\omega| \leq \pi
\]  

(3.77)

The frequencies \(\omega_p\) and \(\omega_s\) are respectively, called the passband edge frequency and the stopband edge frequency. The limits of the tolerances in the passband and stopband, \(\delta_p\) and \(\delta_s\) are usually called the peak ripple values. Note that the frequency response \(G(e^{j\omega})\) of a digital filter is a periodic function of \(\omega\), and the magnitude response of a real coefficient digital filter is an even function of \(\omega\). As a result, the digital filter specifications are given only for the range \(0 \leq |\omega| \leq \pi\).

Often, the specifications of the digital filter are given in terms of the loss function,

\(\zeta(\omega) = -20\log_{10}|G(e^{j\omega})|,\) in dB. Here the peak passband ripple \(\alpha_p\) and the minimum stopband attenuation \(\alpha_s\) are given in dB, i.e., the loss specifications of a digital filter are given by
\[ \alpha_p = -20 \log_{10} (1 - \delta_p) \text{ dB}, \quad (3.78) \]

\[ \alpha_s = -20 \log_{10} (\delta_s) \text{ dB}. \quad (3.79) \]

As in the case of an analog lowpass filter, the specifications for a digital lowpass filter may alternatively be given in terms of its magnitude response, as in Fig 3.8a. Here the maximum value of the magnitude in the passband is assumed to be unity, and the maximum passband deviation, denoted as \( 1 / \sqrt{1+e^2} \), is given by the maximum value of the magnitude in the passband. The maximum stopband magnitude is given by \( 1/A \).

For the normalized specification, the maximum value of the gain function or the minimum value of the loss function is therefore 0 dB. The quantity \( \alpha_{\text{max}} \) given by

\[ \alpha_{\text{max}} = 20 \log_{10} (\sqrt{1+e^2}) \text{ dB} \quad (3.80) \]

is called the maximum attenuation. For \( \delta_p << 1 \), as is typically the case, it can be shown that

\[ \alpha_{\text{max}} = 20 \log_{10} (1 - 2 \delta_p) \equiv 2 \alpha_p \quad (3.81) \]

The passband and stopband frequencies, in most applications, are specified in Hz, along with the sampling rate of the digital filter. Since all filter design techniques are developed in terms of normalized angular frequencies \( \Omega_p \) and \( \Omega_s \), the specified critical frequencies need to be normalized first before a specific filter design algorithm can be applied. Let \( F_T \) denote the sampling frequency in Hz, and \( F_p \) and \( F_s \) denote, respectively, the passband and stopband edge frequencies in Hz. Then the normalized angular edge frequencies in radians are given by

\[ \omega_p = \frac{\Omega_p}{F_T} = \frac{2 \pi F_p}{F_T} = 2 \pi F_T \quad (3.82) \]

\[ \omega_s = \frac{\Omega_s}{F_T} = \frac{2 \pi F_s}{F_T} = 2 \pi F_T \quad (3.83) \]

### 3.4.3 Selection of the Filter Type:

The second issue of interest is the selection of the digital filter type, i.e., whether an IIR or an FIR digital filter is to be employed. There are several advantages to using an FIR filter, since it can
Fig 3.7 Four types of ideal filters: (a) ideal lowpass filter, (b) ideal highpass filter, (c) ideal bandpass filter, and (d) ideal bandstop filter.

Fig 3.8 Magnitude responses of FIR filters: (a) first-order lowpass filter $H_{LP}(e^{j\omega})$ of Eq (4.64), (b) cascade of 3 sections of $H_0(c)$, and (c) first-order highpass filter $H_{HP}(e^{j\omega})$ of Eq (4.67).
be designed with exact linear phase and the filter structure is always stable with quantized filter coefficients. However, in most cases, the order NFIR of an FIR filter is considerably higher than the order NIIR of an equivalent IIR filter meeting the same magnitude specifications. In general, the implementation of the FIR filter requires approximately NFIR multiplications per output sample, whereas the IIR filter requires 2NIIR+1 multiplications per output sample. In the former case, if the FIR filter is designed with the linear phase, then the number of multiplications per output sample reduces to approximately (NFIR+1)/2. Likewise, most IIR filter design result in transfer functions with zeros on the unit circle, and the cascade realization of an IIR filter order NIIR with zeros on the unit circle requires [(3NIIR+3)/2] multiplications per output sample. It has been shown by Rabiner and Gold[13] that for most practical filter specifications, the ratio NFIR/NIIR is typically of the order of tens or more, and as a result, the IIR filter usually is computationally more efficient. However, if the group delay of the IIR filter is equalized by cascading it with an allpass equalizer, then the savings in the computation may no longer be significant[13]. In many applications, the linearity of the phase response of the digital filter is not an issue, making the IIR filter preferable because of the lower computational requirements.

3.4.4 Basic Approaches to Digital Filter Design:

In the case of an IIR filter design, the most common practice is to convert the digital filter specifications to analog lowpass prototype filter specifications, to determine the analog lowpass filter transfer function Ha(s) meeting these specifications, and then to transform it into the desired digital filter function G(z). This approach has been widely used for many reasons:
(a) Analog approximation techniques are highly advanced
(b) They usually yield closed-form solutions
(c) Extensive tables are available for analog filter design
(d) Many applications require the digital simulation of analog filters.

In the sequel, we denote an analog transfer function as

$$H_a(s) = \frac{P_a(s)}{D_a(s)}$$

(3.84)
where the subscript “a” specifically indicates the analog domain. The digital transfer function derived from $H_a(s)$ is denoted by

$$G(z) = \frac{P(z)}{D(z)} \quad (3.85)$$

The basic idea behind the conversion of an analog prototype transfer function $H_a(s)$ to a digital IIR transfer function $G(z)$ is to apply a mapping from $s$-domain to the $z$-domain so that the essential properties of the analog frequency response are preserved. This implies that the mapping function should be such that:

(a) The imaginary (j) axis in the $s$-plane be mapped onto the unit circle of the $z$-plane

(b) A stable analog transfer function be transformed into a stable digital transfer function.

Unlike the IIR digital filter design, the FIR filter design does not have any connection with the design of analog filters. The design of FIR filters is therefore based on a direct approximation of the specified magnitude response, with the often added requirement that the phase response be linear. A casual FIR transfer function $H(z)$ of length $N$ is a polynomial in $z^{-1}$ of degree $N-1$,:

$$H(z) = \sum_{n=0}^{N-1} h[n] z^{-n} \quad (3.86)$$

The corresponding frequency response is given by

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h[n] e^{-j\omega n} \quad (3.87)$$

It is the fact that any finite duration sequence $x[n]$ of length $N$ is completely characterized by $N$ samples of its Fourier transform $X(e^{j\omega})$. As a result, the design of an FIR filter of length $N$ may be accomplished by finding either the impulse response sequence $\{h[n]\}$ or $N$ samples of its frequency response $H(e^{j\omega})$. Also, to ensure a linear phase design, the condition $h[N-1-n] = h[n]$ or $-h[n]$ must be satisfied. Two direct approaches to the design of FIR filters are the truncated Fourier series approach and the frequency sampling approach.
REFERENCES OF CHAPTER 3:


