

## 2

# Harmonic Analysis

## 2.1 Introduction

The voltage and current waveforms at points of connection of nonlinear devices can either be obtained from appropriate transducers or calculated for a given operating condition, from knowledge of the devices' nonlinear characteristics. In 1822 J.B.J. Fourier [1] postulated that any continuous function repetitive in an interval  $T$  can be represented by the summation of a d.c. component, a fundamental sinusoidal component and a series of higher-order sinusoidal components (called *harmonics*) at frequencies which are integer multiples of the fundamental frequency.

Harmonic analysis is then the process of calculating the magnitudes and phases of the fundamental and higher-order harmonics of the periodic waveform. The resulting series, known as the Fourier series, establishes a relationship between a time-domain function and that function in the frequency domain.

The Fourier series of a general periodic waveform is derived in the first part of this chapter and its characteristics discussed with reference to simple waveforms.

More generally, the Fourier transform and its inverse are used to map any function in the interval from  $-\infty$  to  $\infty$ , in either the time or frequency domain. The Fourier series therefore represents the special case of the Fourier transform applied to a periodic signal.

In practice, data is often available in the form of a sampled time function, represented by a time series of amplitudes, separated by fixed time intervals of limited duration. When dealing with such data, a modification of the Fourier transform, the discrete Fourier transform (DFT), is used. The implementation of the DFT by means of the so-called Fast Fourier transform (FFT) forms the basis of most modern spectral and harmonic analysis systems.

The voltage and current waveforms captured from the power system, however, may contain transient or time-varying components. Even stationary signals when viewed from limited data (due to finite sampling) will introduce errors in the frequency spectrum of the signal. A variety of techniques have been developed to derive the frequency spectrum under those conditions. The chapter ends with a brief review of these alternative techniques.

## 2.2 Fourier Series and Coefficients [2,3]

The Fourier series of a periodic function  $x(t)$  has the expression

$$x(t) = a_0 + \sum_{n=1}^{\infty} \left( a_n \cos \left( \frac{2\pi nt}{T} \right) + b_n \sin \left( \frac{2\pi nt}{T} \right) \right) \quad (2.1)$$

This constitutes a frequency-domain representation of the periodic function.

In this expression  $a_0$  is the average value of the function  $x(t)$ , while  $a_n$  and  $b_n$ , the coefficients of the series, are the rectangular components of the  $n$ th harmonic. The corresponding  $n$ th harmonic vector is

$$A_n \angle \phi_n = a_n + jb_n \quad (2.2)$$

with magnitude

$$A_n = \sqrt{a_n^2 + b_n^2}$$

and phase angle

$$\phi_n = \tan^{-1} \left( \frac{b_n}{a_n} \right)$$

For a given function  $x(t)$ , the constant coefficient,  $a_0$ , can be derived by integrating both sides of equation (2.1) from  $-T/2$  to  $T/2$  (over a period  $T$ ):

$$\int_{-T/2}^{T/2} x(t) dt = \int_{-T/2}^{T/2} \left[ a_0 + \sum_{n=1}^{\infty} a_n \cos \left( \frac{2\pi nt}{T} \right) + b_n \sin \left( \frac{2\pi nt}{T} \right) \right] dt \quad (2.3)$$

The Fourier series of the right-hand side can be integrated term by term, giving

$$\begin{aligned} \int_{-T/2}^{T/2} x(t) dt &= a_0 \int_{-T/2}^{T/2} dt + \sum_{n=1}^{\infty} \left[ a_n \int_{-T/2}^{T/2} \cos \left( \frac{2\pi nt}{T} \right) dt \right. \\ &\quad \left. + b_n \int_{-T/2}^{T/2} \sin \left( \frac{2\pi nt}{T} \right) dt \right] \end{aligned} \quad (2.4)$$

The first term on the right-hand side equals  $Ta_0$ , while the other integrals are zero. Hence, the constant coefficient of the Fourier series is given by

$$a_0 = 1/T \int_{-T/2}^{T/2} x(t) dt \quad (2.5)$$

which is the area under the curve of  $x(t)$  from  $-T/2$  to  $T/2$ , divided by the period of the waveform,  $T$ .

The  $a_n$  coefficients can be determined by multiplying equation (2.1) by  $\cos(2\pi mt/T)$ , where  $m$  is any fixed positive integer, and integrating between  $-T/2$  and  $T/2$ , as previously:

$$\begin{aligned}
 \int_{-T/2}^{T/2} x(t) \cos\left(\frac{2\pi mt}{T}\right) dt &= \int_{-T/2}^{T/2} \left[ a_0 + \sum_{n=1}^{\infty} \left[ a_n \cos\left(\frac{2\pi nt}{T}\right) + b_n \sin\left(\frac{2\pi nt}{T}\right) \right] \right] \cos\left(\frac{2\pi mt}{T}\right) dt \\
 &= a_0 \int_{-T/2}^{T/2} \cos\left(\frac{2\pi mt}{T}\right) dt + \sum_{n=1}^{\infty} \left[ a_n \int_{-T/2}^{T/2} \cos\left(\frac{2\pi nt}{T}\right) \right. \\
 &\quad \times \cos\left(\frac{2\pi mt}{T}\right) dt \\
 &\quad \left. + b_n \int_{-T/2}^{T/2} \sin\left(\frac{2\pi nt}{T}\right) \cos\left(\frac{2\pi mt}{T}\right) dt \right] \quad (2.6)
 \end{aligned}$$

The first term on the right-hand side is zero, as are all the terms in  $b_n$  since  $\sin(2\pi nt/T)$  and  $\cos(2\pi mt/T)$  are orthogonal functions for all  $n$  and  $m$ .

Similarly, the terms in  $a_n$  are zero, being orthogonal, unless  $m = n$ . In this case, equation (2.6) becomes

$$\begin{aligned}
 \int_{-T/2}^{T/2} x(t) \cos\left(\frac{2\pi mt}{T}\right) dt &= a_n \int_{-T/2}^{T/2} \cos\left(\frac{2\pi nt}{T}\right) dt \\
 &= \frac{a_n}{2} \int_{-T/2}^{T/2} \cos\left(\frac{4\pi nt}{T}\right) dt + \frac{a_n}{2} \int_{-T/2}^{T/2} dt \quad (2.7)
 \end{aligned}$$

The first term on the right-hand side is zero while the second term equals  $a_n T/2$ . Hence, the coefficients  $a_n$  can be obtained from

$$a_n = \frac{2}{T} \int_{-T/2}^{T/2} x(t) \cos\left(\frac{2\pi nt}{T}\right) dt \quad \text{for } n = 1 \rightarrow \infty \quad (2.8)$$

To determine the coefficients  $b_n$ , equation (2.1) is multiplied by  $\sin(2\pi mt/T)$  and, by a similar argument to the above,

$$b_n = \frac{2}{T} \int_{-T/2}^{T/2} x(t) \sin\left(\frac{2\pi nt}{T}\right) dt \quad \text{for } n = 1 \rightarrow \infty \quad (2.9)$$

It should be noted that because of the periodicity of the integrands in equations (2.5), (2.8) and (2.9), the interval of integration can be taken more generally as  $t$  and  $t + T$ .

If the function  $x(t)$  is piecewise continuous (i.e. has a finite number of vertical jumps) in the interval of integration, the integrals exist and Fourier coefficients can be calculated for this function.

Equations (2.5), (2.8) and (2.9) are often expressed in terms of the angular frequency as follows:

$$a_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} x(\omega t) d(\omega t) \quad (2.10)$$

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} x(\omega t) \cos(n\omega t) d(\omega t) \quad (2.11)$$

$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} x(\omega t) \sin(n\omega t) d(\omega t) \quad (2.12)$$

so that

$$x(t) = a_0 + \sum_{n=1}^{\infty} [a_n \cos(n\omega t) + b_n \sin(n\omega t)] \quad (2.13)$$

### 2.3 Simplifications Resulting From Waveform Symmetry [2,3]

Equations (2.5), (2.8) and (2.9), the general formulas for the Fourier coefficients, can be represented as the sum of two separate integrals:

$$a_n = \frac{2}{T} \int_0^{T/2} x(t) \cos\left(\frac{2\pi nt}{T}\right) dt + \frac{2}{T} \int_{-T/2}^0 x(t) \cos\left(\frac{2\pi nt}{T}\right) dt \quad (2.14)$$

$$b_n = \frac{2}{T} \int_0^{T/2} x(t) \sin\left(\frac{2\pi nt}{T}\right) dt + \frac{2}{T} \int_{-T/2}^0 x(t) \sin\left(\frac{2\pi nt}{T}\right) dt \quad (2.15)$$

Replacing  $t$  by  $-t$  in the second integral of equation (2.14), and changing the limits produces

$$\begin{aligned} a_n &= \frac{2}{T} \int_0^{T/2} x(t) \cos\left(\frac{2\pi nt}{T}\right) dt + \frac{2}{T} \int_{+T/2}^0 x(-t) \cos\left(\frac{-2\pi nt}{T}\right) d(-t) \\ &= \frac{2}{T} \int_0^{T/2} [x(t) + x(-t)] \cos\left(\frac{2\pi nt}{T}\right) dt \end{aligned} \quad (2.16)$$

Similarly,

$$b_n = \frac{2}{T} \int_0^{T/2} [x(t) - x(-t)] \sin\left(\frac{2\pi nt}{T}\right) dt \quad (2.17)$$

**Odd Symmetry** The waveform has odd symmetry if  $x(t) = -x(-t)$ . Then the  $a_n$  terms become zero for all  $n$ , while

$$b_n = \frac{4}{T} \int_0^{T/2} x(t) \sin\left(\frac{2\pi nt}{T}\right) dt \quad (2.18)$$

The Fourier series for an odd function will, therefore, contain only sine terms.

**Even Symmetry** The waveform has even symmetry if  $x(t) = x(-t)$ . In this case  $b_n = 0$  for all  $n$  and

$$a_n = \frac{4}{T} \int_0^{T/2} x(t) \cos\left(\frac{2\pi nt}{T}\right) dt \quad (2.19)$$

The Fourier series for an even function will, therefore, contain only cosine terms.

Certain waveforms may be odd or even depending on the time reference position selected. For instance, the square wave of Figure 2.1, drawn as an odd function, can be transformed into an even function by shifting the origin (vertical axis) by  $T/2$ .

**Halfwave Symmetry** A function  $x(t)$  has halfwave symmetry if

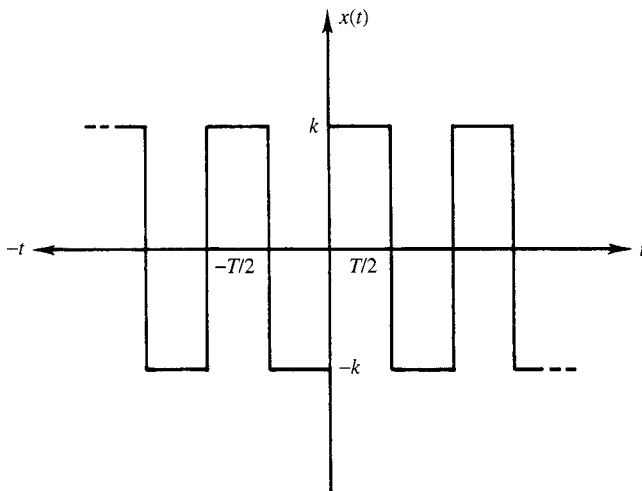
$$x(t) = -x(t + T/2) \quad (2.20)$$

i.e. the shape of the waveform over a period  $t + T/2$  to  $t + T$  is the negative of the shape of the waveform over the period  $t$  to  $t + T/2$ . Consequently, the square wave function of Figure 2.1 has halfwave symmetry.

Using equation (2.8) and replacing  $(t)$  by  $(t + T/2)$  in the interval  $(-T/2, 0)$

$$\begin{aligned} a_n &= \frac{2}{T} \int_0^{T/2} x(t) \cos\left(\frac{2\pi nt}{T}\right) dt + \frac{2}{T} \int_{-T/2+T/2}^{0+T/2} x(t + T/2) \cos\left(\frac{2\pi n(t + T/2)}{T}\right) dt \\ &= \frac{2}{T} \int_0^{T/2} x(t) \left[ \cos\left(\frac{2\pi nt}{T}\right) - \cos\left(\frac{2\pi nt}{T} + n\pi\right) \right] dt \end{aligned} \quad (2.21)$$

since by definition  $x(t) = -x(t + T/2)$ .



**Figure 2.1** Square wave function

If  $n$  is an odd integer then

$$\cos\left(\frac{2\pi nt}{T} + n\pi\right) = -\cos\left(\frac{2\pi nt}{T}\right)$$

and

$$a_n = \frac{4}{T} \int_0^{T/2} x(t) \cos\left(\frac{2\pi nt}{T}\right) dt \quad (2.22)$$

However, if  $n$  is an even integer then

$$\cos\left(\frac{2\pi nt}{T} + n\pi\right) = \cos\left(\frac{2\pi nt}{T}\right)$$

and

$$a_n = 0.$$

Similarly,

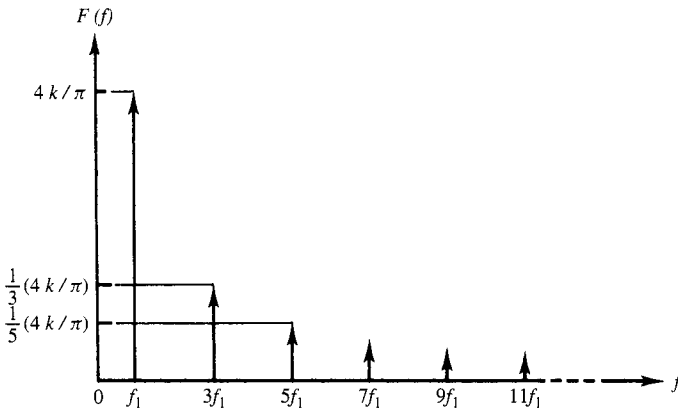
$$\begin{aligned} b_n &= \frac{4}{T} \int_0^{T/2} x(t) \sin\left(\frac{2\pi nt}{T}\right) dt \quad \text{for } n \text{ odd} \\ &= 0 \quad \text{for } n \text{ even} \end{aligned} \quad (2.23)$$

Thus, waveforms which have halfwave symmetry contain only odd order harmonics.

The square wave of Figure 2.1 is an odd function with halfwave symmetry. Consequently, only the  $b_n$  coefficients and odd harmonics will exist. The expression for the coefficients taking into account these conditions is

$$b_n = \frac{8}{T} \int_0^{T/4} x(t) \sin\left(\frac{2\pi nt}{T}\right) dt \quad (2.24)$$

which can be represented by a line spectrum of amplitudes inversely proportional to the harmonic order, as shown in Figure 2.2.



**Figure 2.2** Line spectrum representation of a square wave

## 2.4 Complex Form of the Fourier Series

The representation of the frequency components as rotating vectors in the complex plane gives a geometrical interpretation of the relationship between waveforms in the time and frequency domains.

A uniformly rotating vector  $A/2e^{j\phi}(X(f_n))$  has a constant magnitude  $A/2$ , and a phase angle  $\phi$ , which is time varying according to

$$\phi = 2\pi fT + \theta \quad (2.25)$$

where  $\theta$  is the initial phase angle when  $t = 0$ .

A second vector  $A/2e^{-j\phi}(X(-f_n))$  with magnitude  $A/2$  and phase angle  $-\phi$  will rotate in the opposite direction to  $A/2e^{j\phi}(X(f_n))$ . This negative rate of change of phase angle can be considered as a negative frequency.

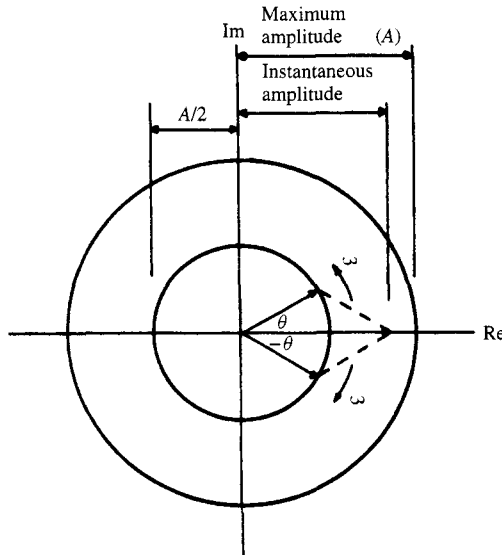
The sum of the two vectors will always lie along the real axis, the magnitude oscillating between  $A$  and  $-A$  according to

$$\frac{A}{2}e^{j\phi} + \frac{A}{2}e^{-j\phi} = A \cos \phi \quad (2.26)$$

Thus, each harmonic component of a real valued signal can be represented by two half-amplitude contra-rotating vectors as shown in Figure 2.3, such that

$$X(f_n) = X^*(-f_n) \quad (2.27)$$

where  $X^*(-f_n)$  is the complex conjugate of  $X(-f_n)$ .



**Figure 2.3** Contra-rotating vector pair producing a varying amplitude (pulsating) vector

The sine and cosine terms of equations (2.11) and (2.12) may, therefore, be solved into positive- and negative-frequency terms using the trigonometric identities

$$\cos(n\omega t) = \frac{e^{jn\omega t} + e^{-jn\omega t}}{2} \quad (2.28)$$

$$\sin(n\omega t) = \frac{e^{jn\omega t} - e^{-jn\omega t}}{2j} \quad (2.28a)$$

Substituting into equation (2.13) and simplifying yields

$$x(t) = \sum c_n e^{jn\omega t} \quad (2.29)$$

where

$$\begin{aligned} c_n &= \frac{1}{2}(a_n - jb_n), & n > 0 \\ c_{-n} &= c_n \\ c_0 &= a_0 \end{aligned}$$

The  $c_n$  terms can also be obtained by complex integration:

$$c_n = \frac{1}{\pi} \int_{-\pi}^{\pi} x(\omega t) e^{-jn\omega t} d(\omega t) \quad (2.30)$$

$$c_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} x(\omega t) d(\omega t) \quad (2.31)$$

If the time domain signal  $x(t)$  contains a component rotating at a single frequency  $nf$ , then multiplication by the unit vector  $e^{-j2\pi nft}$ , which rotates at a frequency  $-nf$ , annuls the rotation of the component, such that the integration over a complete period has a finite value. All components at other frequencies will continue to rotate after multiplication by  $e^{-j2\pi nft}$ , and will thus integrate to zero.

The Fourier series is most generally used to approximate a periodic function by truncation of the series. In this case, the truncated Fourier series is the best trigonometric series expression of the function, in the sense that it minimises the square error between the function and the truncated series. The number of terms required depends upon the magnitude of repeated derivatives of the function to be approximated. Repeatedly differentiating equation (2.30) by parts, it can readily be shown that

$$c_n = \frac{1}{2\pi} \frac{1}{n^{m+1}} \int_{-\pi}^{\pi} f^{(m+1)}(\omega t) d(\omega t) \quad (2.32)$$

Consequently, the Fourier series for repeatedly differentiated functions will converge faster than that for functions with low-order discontinuous derivatives.

The complex Fourier series expansion is compatible with the FFT, the method of choice for converting time-domain data samples into a Nyquist rate-limited frequency



spectrum. The trigonometric Fourier expression can also be written as a series of phase-shifted sine terms by substituting

$$a_n \cos n\omega t + b_n \sin n\omega t = d_n \sin(n\omega t + \psi_n) \quad (2.33)$$

into equation (2.13), where

$$\begin{aligned} d_n &= \sqrt{a_n^2 + b_n^2} \\ \psi_n &= \tan^{-1} \frac{b_n}{a_n} \end{aligned} \quad (2.34)$$

Finally, the phase-shifted sine terms can be represented as peak value phasors by setting

$$\Psi_n = d_n e^{j\psi_n} \quad (2.35)$$

so that

$$\begin{aligned} d_n \sin(n\omega t + \psi_n) &= I\{\Psi_n e^{jn\omega t}\} \\ &= |\Psi_n| \sin(n\omega t + \angle\Psi_n) \end{aligned} \quad (2.36)$$

The harmonic phasor Fourier series is, therefore,

$$f(t) = \sum_{n=0}^{\infty} I\{\Psi_n e^{jn\omega t}\} \quad (2.37)$$

which does not contain negative-frequency components. Note that the d.c. term becomes

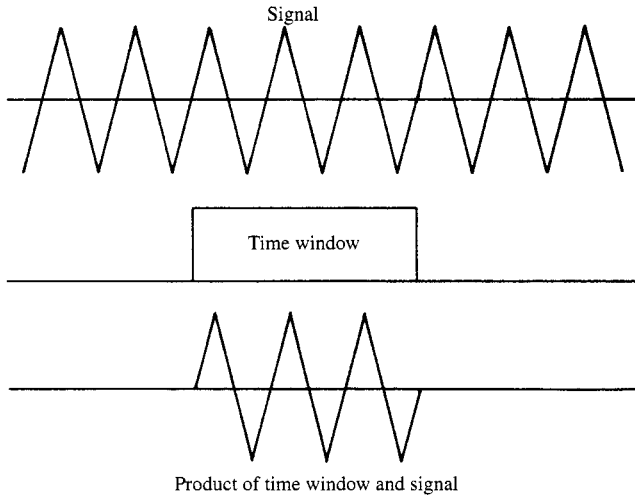
$$\begin{aligned} \Psi_0 &= \frac{a_0}{2} e^{j\pi/2} \\ &= j \frac{a_0}{2} \end{aligned} \quad (2.38)$$

In practice, the upper limit of the summation is set to  $n_h$ , the highest harmonic order of interest.

## 2.5 Convolution of Harmonic Phasors

Viewing a continuous function through a sampling period of interval  $T$  seconds is equivalent to multiplying the signal in the time domain by a rectangular pulse of length  $T$  (Figure 2.4). This corresponds to the convolution in the frequency domain of their respective frequency spectra.

In general, the point-by-point multiplication of two time-domain waveforms is expressed in the harmonic domain by a discrete convolution of their Fourier series. When two harmonic phasors of different frequencies are convolved, the results are harmonic phasors at sum and difference harmonics. This is best explained by multiplying the corresponding sinusoids using the trigonometric identity for the product of sine waves,



**Figure 2.4** Influence of viewing a continuous function through a rectangular time window

and then converting back to phasor form. Given two phasors,  $A_k$  and  $B_m$ , of harmonic orders  $k$  and  $m$ , the trigonometric identity for their time-domain multiplication is:

$$|A_k| \sin(k\omega t + \angle A_k) |B_m| \sin(m\omega t + \angle B_m) = \frac{1}{2} |A_k| |B_m| \left[ \sin \left( (k-m)\omega t + \angle A_k - \angle B_m + \frac{\pi}{2} \right) - \sin \left( (k+m)\omega t + \angle A_k + \angle B_m + \frac{\pi}{2} \right) \right] \quad (2.39)$$

Converting to phasor form:

$$\begin{aligned} A_k \otimes B_m &= \frac{1}{2} |A_k| |B_m| \left[ e^{j(\angle A_k - \angle B_m + \pi/2)} |_{(k-m)} - e^{j(\angle A_k + \angle B_m + \pi/2)} |_{(k+m)} \right] \\ &= \frac{1}{2} \left[ (|A_k| e^{j\angle A_k} |B_m| e^{-j\angle B_m} e^{j\pi/2}) |_{(k-m)} - (|A_k| e^{j\angle A_k} |B_m| e^{j\angle B_m} e^{j\pi/2}) |_{(k+m)} \right] \\ &= \frac{1}{2} j [(A_k B_m^*)_{k-m} - (A_k B_m)_{k+m}] \end{aligned} \quad (2.40)$$

If  $k$  is less than  $m$ , a negative harmonic can be avoided by conjugating the difference term. This leads to the overall equation:

$$A_k \otimes B_m = \begin{cases} \frac{1}{2} j (A_k B_m^*)_{(k-m)} - \frac{1}{2} j (A_k B_m)_{(k+m)} & \text{if } k \geq m \\ \frac{1}{2} j (A_k B_m^*)_{(m-k)}^* - \frac{1}{2} j (A_k B_m)_{(k+m)} & \text{otherwise} \end{cases} \quad (2.41)$$

The multiplication of two non-sinusoidal periodic waveforms leads to a discrete convolution of their harmonic phasor Fourier series:

$$\begin{aligned} f_a(t) f_b(t) &= \sum_{k=0}^{n_h} |A_k| \sin(k\omega t + \angle A_k) \sum_{m=0}^{n_h} |B_m| \sin(m\omega t + \angle B_m) \\ &= \sum_{k=0}^{n_h} \sum_{m=0}^{n_h} |A_k| \sin(k\omega t + \angle A_k) |B_m| \sin(m\omega t + \angle B_m) \end{aligned} \quad (2.42)$$

Rewriting this in terms of phasors yields

$$\mathbf{F}_A \otimes \mathbf{F}_B = \sum_{k=0}^{n_h} \sum_{m=0}^{n_h} A_k \otimes B_m \quad (2.43)$$

Equation (2.43) generates harmonic phasors of order up to  $2n_h$ , due to the sum terms. Substituting the equation for the convolution of two phasors, equation (2.41), into (2.43) and solving for the  $l$ th order component yields

$$(A \otimes B)_l = \frac{1}{2}j \left[ \sum_{k=1}^{n_h} A_k B_{k-l}^* + \sum_{k=1}^{n_h} (A_k B_{k+l}^*)^* - \sum_{k=0}^{n_h} A_k B_{l-k}^* \right], l > 0 \quad (2.44)$$

$$(A \otimes B)_l = \frac{1}{2}j \left[ -A_0 B_0 + \sum_{k=0}^{n_h} A_k B_k^* \right], l = 0 \quad (2.45)$$

The convolution equations are non-analytic in the complex plane but are differentiable by decomposing into two real valued components (typically rectangular).

If negative frequencies are retained, the convolution is just the multiplication of two series:

$$\begin{aligned} f_{a(t)} f_{b(t)} &= \sum_{n=-n_h}^{n_h} c_{an} e^{jn\omega t} \sum_{l=-n_h}^{n_h} c_{bl} e^{jl\omega t} \\ &= \sum_{n=-n_h}^{n_h} \sum_{l=-n_h}^{n_h} c_{an} c_{bl} e^{j(l+n)\omega t} \end{aligned} \quad (2.46)$$

In practice, the discrete convolution can be evaluated faster using FFT methods.

## 2.6 The Fourier Transform [3,4]

Fourier analysis, when applied to a continuous, periodic signal in the time domain, yields a series of discrete frequency components in the frequency domain.

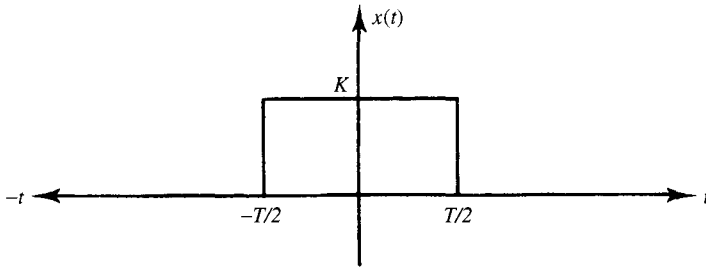
By allowing the integration period to extend to infinity, the spacing between the harmonic frequencies,  $\omega$ , tends to zero and the Fourier coefficients,  $c_n$ , of equation (2.30) become a continuous function, such that

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi f T} dt \quad (2.47)$$

The expression for the time domain function  $x(t)$ , which is also continuous and of infinite duration, in terms of  $X(f)$  is then

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi f T} df \quad (2.48)$$

$X(f)$  is known as the spectral density function of  $x(t)$ .



**Figure 2.5** Rectangular function

Equations (2.47) and (2.48) form the Fourier transform pair. Equation (2.47) is referred to as the 'forward transform' and equation (2.48) as the 'reverse' or 'inverse transform'.

In general,  $X(f)$  is complex and can be written as

$$X(f) = \text{Re } X(f) + j \text{Im } X(f) \quad (2.49)$$

The real part of  $X(f)$  is obtained from

$$\text{Re } X(f) = \frac{1}{2}[X(f) + X(-f)] = \int_{-\infty}^{\infty} x(t) \cos 2\pi f t \, dt \quad (2.50)$$

Similarly, for the imaginary part of  $X(f)$

$$\text{Im } X(f) = \frac{1}{2}j[X(f) - X(-f)] = - \int_{-\infty}^{\infty} x(t) \sin 2\pi f T \, dt \quad (2.51)$$

The amplitude spectrum of the frequency signal is obtained from

$$|X(f)| = [(\text{Re } X(f))^2 + (\text{Im } X(f))^2]^{\frac{1}{2}} \quad (2.52)$$

The phase spectrum is

$$\phi(f) = \tan^{-1} \left[ \frac{\text{Im } X(f)}{\text{Re } X(f)} \right] \quad (2.53)$$

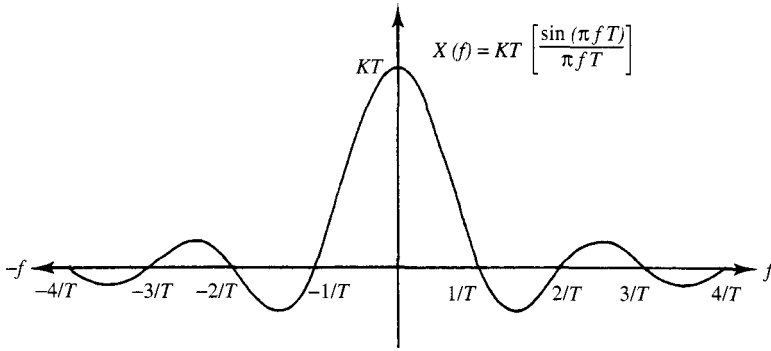
Using equations (2.49) to (2.53), the inverse Fourier transform can be expressed in terms of the magnitude and phase spectra components:

$$x(t) = \int_{-\infty}^{\infty} |X(f)| \cos[2\pi f T - \phi(f)] \, df \quad (2.54)$$

As an example, let us consider a rectangular function such as Figure 2.5, defined by

$$\begin{aligned} x(t) &= K \text{ for } |t| \leq T/2 \\ &= 0 \text{ for } |t| > T/2 \end{aligned}$$

i.e. the function is continuous over all  $t$  but is zero outside the limits  $(-T/2, T/2)$ .



**Figure 2.6** The sinc function,  $\sin(\pi f T)/(\pi f T)$

Its Fourier transform is

$$\begin{aligned}
 X(f) &= \int_{-\infty}^{\infty} x(t) e^{-j2\pi f T} dt \\
 &= \int_{-T/2}^{T/2} K e^{-j2\pi f T} dt \\
 &= \frac{-K}{\pi f} \cdot \frac{1}{2j} [e^{-j\pi f T} - e^{j\pi f T}]
 \end{aligned} \tag{2.55}$$

and using the identity

$$\sin \phi = \frac{1}{2j} (e^{j\phi} - e^{-j\phi})$$

yields the following expression for the Fourier transform:

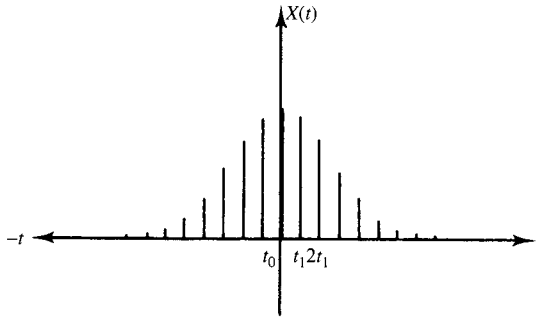
$$\begin{aligned}
 X(f) &= \frac{K}{\pi f} \sin(\pi f T) \\
 &= K T \left[ \frac{\sin(\pi f T)}{\pi f T} \right]
 \end{aligned} \tag{2.56}$$

The term in brackets, known as the sinc function, is shown in Figure 2.6.

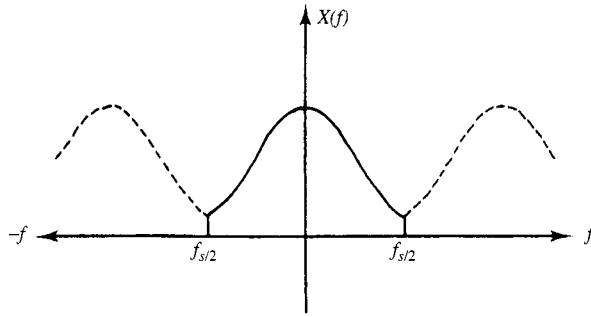
While the function is continuous, it has zero value at the points  $f = n/T$  for  $n = \pm 1, \pm 2, \dots$  and the side lobes decrease in magnitude as  $1/T$ . This should be compared to the Fourier series of a periodic square wave, which has discrete frequencies at odd harmonics. The interval  $1/T$  is the effective bandwidth of the signal.

## 2.7 Sampled Time Functions [4,5]

With an increase in the digital processing of data, functions are often recorded by samples in the time domain. Thus, the signal can be represented as in Figure 2.7,



**Figure 2.7** Sampled time-domain function



**Figure 2.8** Frequency spectrum for discrete time-domain function

where  $f_s = 1/t_1$  is the frequency of the sampling. In this case, the Fourier transform of the signal is expressed as the summation of the discrete signal where each sample is multiplied by  $e^{-j2\pi f n t_1}$ :

$$X(f) = \sum_{n=-\infty}^{\infty} x(n t_1) e^{-j2\pi f n t_1} \quad (2.57)$$

The frequency domain spectrum, shown in Figure 2.8, is periodic and continuous. The inverse Fourier transform is thus

$$x(t) = 1/f_s \int_{-f_s/2}^{f_s/2} X(f) e^{j2\pi f n t_1} df \quad (2.58)$$

## 2.8 Discrete Fourier Transform (DFT) [4,5]

In the case where the frequency domain spectrum is a sampled function, as well as the time domain function, we obtain a Fourier transform pair made up of discrete components:

$$X(f_k) = 1/N \sum_{n=0}^{N-1} x(t_n) e^{-j2\pi k n / N} \quad (2.59)$$

and

$$x(t_n) = \sum_{k=0}^{N-1} X(f_k) e^{j2\pi kn/N} \quad (2.60)$$

Both the time domain function and the frequency domain spectrum are assumed periodic as in Figure 2.9, with a total of  $N$  samples per period. It is in this discrete form that the Fourier transform is most suited to numerical evaluation by digital computation.

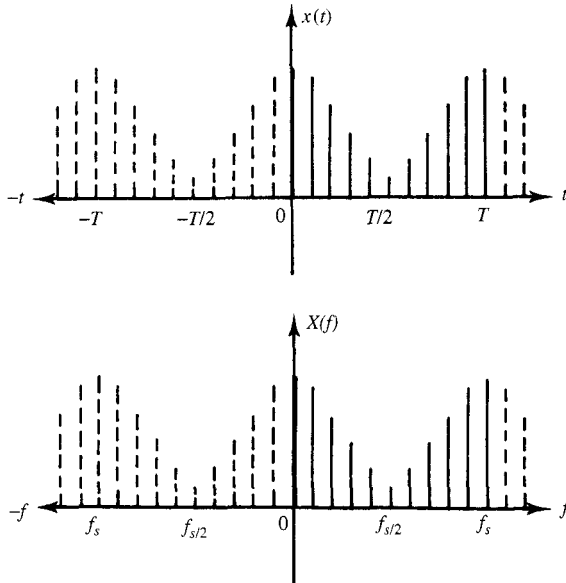
Consider equation (2.59) rewritten as

$$X(f_k) = 1/N \sum_{n=0}^{N-1} x(t_n) W^{kn} \quad (2.61)$$

where  $W = e^{-j2\pi/N}$ .

Over all the frequency components, equation (2.61) becomes a matrix equation.

$$\begin{bmatrix} X(f_0) \\ X(f_1) \\ \vdots \\ X(f_k) \\ \vdots \\ X(f_{N-1}) \end{bmatrix} = 1/N \begin{bmatrix} 1 & 1 & \dots & 1 & \dots & 1 \\ 1 & W & \dots & W^k & \dots & W^{N-1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & W^k & \dots & W^{k^2} & \dots & W^{k(N-1)} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & W^{N-1} & \dots & W^{(N-1)k} & \dots & W^{(N-1)^2} \end{bmatrix} \cdot \begin{bmatrix} x(t_0) \\ x(t_1) \\ \vdots \\ x(t_k) \\ \vdots \\ x(t_{N-1}) \end{bmatrix} \quad (2.62)$$



**Figure 2.9** Discrete time- and frequency-domain function

or in a condensed form

$$[X(f_k)] = 1/N[W^{kn}][x(t_n)] \quad (2.63)$$

In these equations,  $[X(f_k)]$  is a vector representing the  $N$  components of the function in the frequency domain, while  $[x(t_n)]$  is a vector representing the  $N$  samples of the function in the time domain.

Calculation of the  $N$  frequency components from the  $N$  time samples, therefore, requires a total of  $N^2$  complex multiplications to implement in the above form.

Each element in the matrix  $[W^{kn}]$  represents a unit vector with a clockwise rotation of  $2n\pi/N$  ( $n = 0, 1, 2, \dots, (N-1)$ ) introduced between successive components. Depending on the value of  $N$ , a number of these elements are the same.

For example, if  $N = 8$  then

$$\begin{aligned} W &= e^{-j2\pi/8} \\ &= \cos \frac{\pi}{4} - j \sin \frac{\pi}{4} \end{aligned}$$

As a consequence

$$\begin{aligned} W^0 &= -W^4 = 1 \\ W^1 &= -W^5 = (1/\sqrt{2} - j1/\sqrt{2}) \\ W^2 &= -W^6 = -j \\ W^3 &= -W^7 = -(1/\sqrt{2} + j1/\sqrt{2}) \end{aligned}$$

These can also be thought of as unit vectors rotated through  $\pm 0^\circ$ ,  $\pm 45^\circ$ ,  $\pm 90^\circ$  and  $\pm 135^\circ$ , respectively.

Further,  $W^8$  is a complete rotation and hence equal to 1. The value of the elements of  $W^{kn}$  for  $kn > 8$  can thus be obtained by subtracting full rotations, to leave only a fraction of a rotation, the values for which are shown above. For example, if  $k = 5$  and  $n = 6$ , then  $kn = 30$  and  $W^{30} = W^{3 \times 8 + 6} = W^6 = j$ .

Thus, there are only four unique absolute values of  $W^{kn}$  and the matrix  $[W^{kn}]$ , for the case  $N = 8$ , becomes

1	1	1	1	1	1	1	1
1	$W$	$-j$	$W^3$	$-1$	$-W$	$j$	$-W^3$
1	$-j$	$-1$	$j$	1	$-j$	$-1$	$j$
1	$W^3$	$j$	$W$	$-1$	$-W^3$	$-j$	$-W$
1	$-1$	1	$-1$	1	$-1$	1	$-1$
1	$-W$	$-j$	$-W^3$	$-1$	$W$	$j$	$W^3$
1	$j$	$-1$	$-j$	1	$j$	$-1$	$-j$
1	$-W^3$	$j$	$-W$	$-1$	$W^3$	$-j$	$W$



It can be observed that the d.c. component of the frequency spectrum,  $X(f_0)$ , obtained by the algebraic addition of all the time domain samples, divided by the number of samples, is the average value of all the samples.

Subsequent rows show that each time sample is weighted by a rotation dependent on the row number. Thus, for  $X(f_1)$  each successive time sample is rotated by  $1/N$  of a revolution; for  $X(f_2)$  each sample is rotated by  $2/N$  revolutions, and so on.

## 2.9 The Nyquist Frequency and Aliasing [4]

With regard to equation (2.62) for the DFT and the matrix  $[W^{kn}]$ , it can be observed that for the rows  $N/2$  to  $N$ , the rotations applied to each time sample are the negative of those in rows  $N/2$  to 1. Frequency components above  $k = N/2$  can be considered as negative frequencies, since the unit vector is being rotated through increments greater than  $\pi$  between successive components. In the example of  $N = 8$ , the elements of row 3 are successively rotated through  $-\pi/2$ . The elements of row 7 are similarly rotated through  $-3\pi/2$ , or in negative frequency form through  $\pi/2$ . More generally, a rotation through

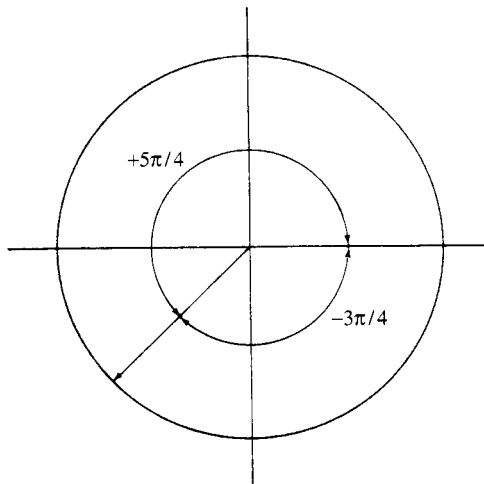
$$2\pi(N/2 + p)/N \text{ radians}$$

for  $p = 1, 2, 3 \dots, (N/2 - 1)$  [with  $N$  even]

corresponds to a negative rotation of

$$-2\pi(N/2 - p)/N \text{ radians}$$

Hence,  $-X(k)$  corresponds to  $X(N - k)$  for  $k = 1$  to  $N/2$  as shown by Figure 2.10.



**Figure 2.10** Correspondence of positive and negative angles

This is an interpretation of the sampling theorem, which states that the sampling frequency must be at least twice the highest frequency contained in the original signal for a correct transfer of information to the sampled system.

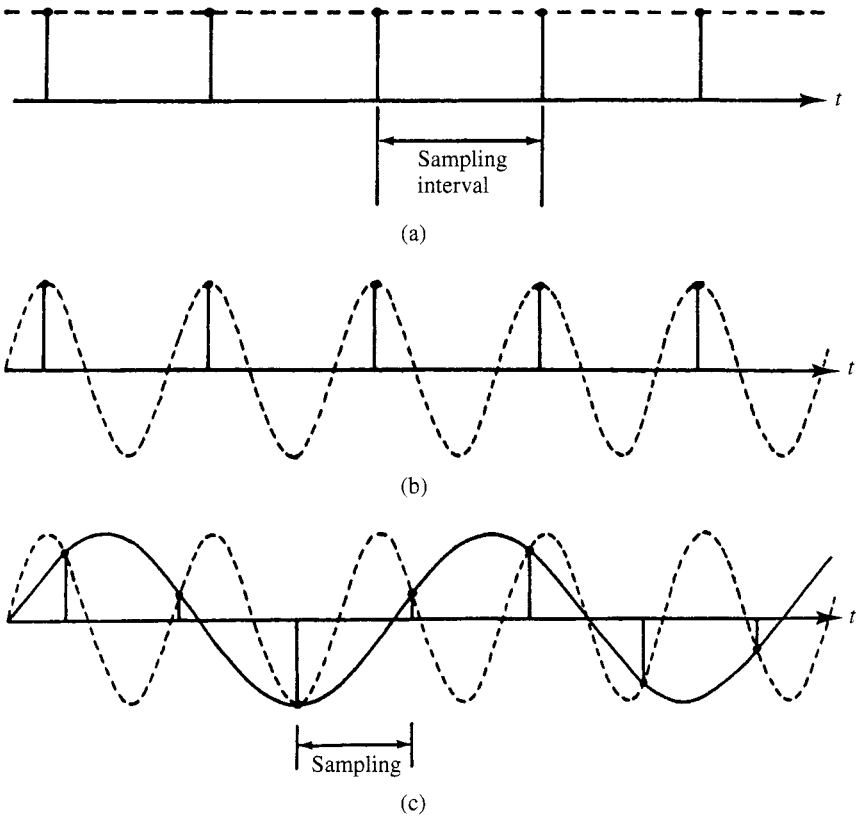
The frequency component at half the sampling frequency is referred to as the Nyquist frequency.

The representation of frequencies above the Nyquist frequency as negative frequencies means that if the sampling rate is less than twice the highest frequency present in the sampled waveform, then these higher frequency components can mimic components below the Nyquist frequency, introducing error into the analysis.

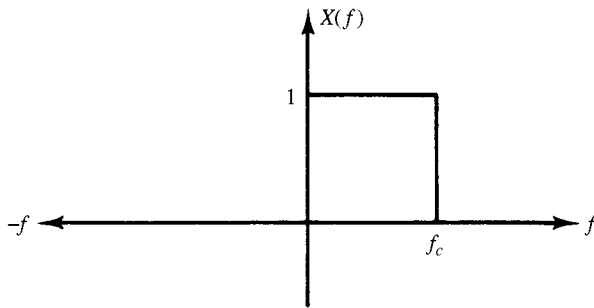
It is possible for high-frequency components to complete many revolutions between samplings; however, since they are only sampled at discrete points in time, this information is lost.

This misinterpretation of frequencies above the Nyquist frequency, as lower frequencies, is called *aliasing* and is illustrated in Figure 2.11.

To prevent aliasing it is necessary to pass the time domain signal through a band-limited low-pass filter, the ideal characteristic of which is shown in Figure 2.12, with a cut-off frequency,  $f_c$ , equal to the Nyquist frequency.



**Figure 2.11** The effect of aliasing: (a)  $x(t) = k$ ; (b)  $x(t) = k \cos 2\pi nft$ . For (a) and (b) both signals are interpreted as being d.c. In (c) the sampling can represent two different signals with frequencies above and below the Nyquist or sampling rate



**Figure 2.12** Frequency domain characteristics of an ideal low-pass filter with cut-off frequency  $f_c$

Thus, if sampling is undertaken on the filtered signal and the DFT applied, the frequency spectrum has no aliasing effect and is an accurate representation of the frequencies in the original signal that are below the Nyquist frequency. However, information on those frequencies above the Nyquist frequency is lost due to the filtering process.

## 2.10 Fast Fourier Transform (FFT) [4–7]

For large values of  $N$ , the computational time and cost of executing the  $N^2$  complex multiplications of the DFT can become prohibitive.

Instead, a calculation procedure known as the FFT, which takes advantage of the similarity of many of the elements in the matrix  $[W^{kn}]$ , produces the same frequency components using only  $N/2 \log_2 N$  multiplications to execute the solution of equation (2.63). Thus, for the case  $N = 1024 = 2^{10}$ , there is a saving in computation time by a factor of over 200. This is achieved by factorising the  $[W^{kn}]$  matrix of equation (2.63) into  $\log_2 N$  individual or factor matrices such that there are only two non-zero elements in each row of these matrices, one of which is always unity. Thus, when multiplying by any factor matrix only  $N$  operations are required.

The reduction in the number of multiplications required, to  $(N/2)\log_2 N$ , is obtained by recognising that

$$\begin{aligned} W^{N/2} &= -W^0 \\ W^{(N+2)/2} &= -W^1 \text{ etc.} \end{aligned}$$

To obtain the factor matrices, it is first necessary to re-order the rows of the full matrix. If rows are denoted by a binary representation, then the re-ordering is by bit reversal.

For the example where  $N = 8$ , row 5, represented as 100 in binary (row 1 is 000), now becomes row 2, or 001 in binary. Thus, rows 2 and 5 are interchanged. Similarly, rows 4 and 7, represented as 011 and 110, respectively, are also interchanged. Rows 1, 3, 6 and 8 have binary representations which are symmetrical with respect to bit reversal and hence remain unchanged.

The corresponding matrix is now

1	1	1	1	1	1	1	1
1	-1	1	-1	1	-1	1	-1
1	-j	-1	j	1	-j	-1	j
1	j	-1	-j	1	j	-1	-j
1	W	-j	W <sup>3</sup>	-1	-W	j	-W <sup>3</sup>
1	-W	-j	-W <sup>3</sup>	-1	W	j	W <sup>3</sup>
1	W <sup>3</sup>	j	W	-1	-W <sup>3</sup>	-j	-W
1	-W <sup>3</sup>	j	-W	-1	W <sup>3</sup>	-j	W

This new matrix can be separated into log<sub>2</sub>8 (=3) factor matrices:

1	1						
1	-1						
		1	-j				
		1	j				
				1	W		
				1	-W		
						1	W <sup>3</sup>
						1	-W <sup>3</sup>

1		1					
	1		1				
1		-1					
	1		-1				
				1		-j	
					1		-j
				1		j	
					1		j

1				1			
	1				1		
		1				1	
			1				1
1				-1			
	1				-1		
		1				-1	
			1				-1

As previously stated, each factor matrix has only two non-zero elements per row, the first of which is unity.

The re-ordering of the  $[W^{kn}]$  matrix results in a frequency spectrum which is also re-ordered. To obtain the natural order of frequencies, it is necessary to reverse the previous bit reversal.

In practice, a mathematical algorithm implicitly giving factor matrix operations is used for the solution of an FFT [8].

Using  $N = 2^m$ , it is possible to represent  $n$  and  $k$  by  $m$  bit binary numbers such that:

$$n = n_{m-1}2^{m-1} + n_{m-2}2^{m-2} + \cdots + 4n_2 + 2n_1 + n_0 \quad (2.64)$$

$$k = k_{m-1}2^{m-1} + k_{m-2}2^{m-2} + \cdots + 4k_2 + 2k_1 + k_0 \quad (2.65)$$

where

$$n_i = 0, 1 \text{ and } k_i = 0, 1$$

For  $N = 8$ :

$$n = 4n_2 + 2n_1 + n_0$$

and

$$k = 4k_2 + 2k_1 + k_0$$

where  $n_2, n_1, n_0$  and  $k_2, k_1, k_0$  are binary bits ( $n_2, k_2$  most significant and  $n_0, k_0$  least significant).

Equation (2.61) can now be rewritten as:

$$X(k_2, k_1, k_0) = \sum_{n_2=0}^1 \sum_{n_1=0}^1 \sum_{n_0=0}^1 \frac{1}{N} x(n_2, n_1, n_0) W \quad (2.66)$$

Defining  $n$  and  $k$  in this way enables the computation of equation (2.61) to be performed in three independent stages computing in turn:

$$A_1(k_0, n_1, n_0) = \sum_{n_2=0}^1 \frac{1}{N} x(n_2, n_1, n_0) W^{4k_0 n_2} \quad (2.67)$$

$$A_2(k_0, k_1, n_0) = \sum_{n_1=0}^1 A_1(k_0, n_1, n_0) W^{2(k_0+2k_1)n_1} \quad (2.68)$$

$$A_3(k_0, k_1, k_2) = \sum_{n_0=0}^1 A_2(k_0, k_1, n_0) W^{(k_0+2k_1+4k_2)n_0} \quad (2.69)$$

From equation (2.69) it is seen that the  $A_3$  coefficients contain the required  $X(k)$  coefficients but in reverse binary order:

Order of  $A_3$  in binary form is  $k_0 k_1 k_2$

Order of  $X(k)$  in binary form is  $k_2 k_1 k_0$

Hence

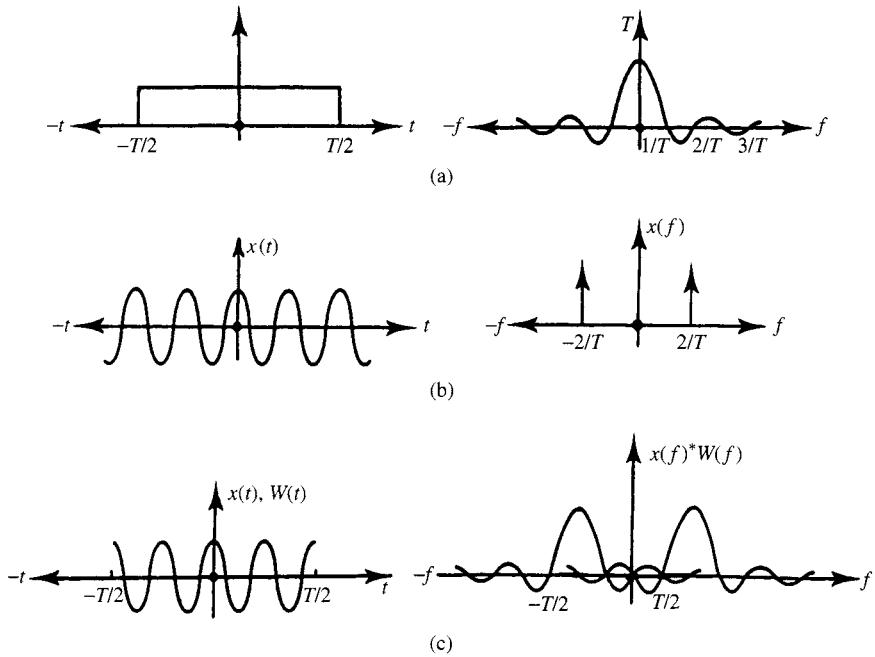
	Binary		Reversed	
$A_3(3)$	$= A_3(011)$	$=$	$X(110)$	$= X(6)$
$A_3(4)$	$= A_3(100)$	$=$	$X(001)$	$= X(1)$
$A_3(5)$	$= A_3(101)$	$=$	$X(101)$	$= X(5)$

## 2.11 Window Functions [9]

In any practical measurement of a time domain signal, it is normal to limit the time duration over which the signal is observed. This process is known as *windowing* and is particularly useful for the measurement of non-stationary signals, which may be divided into short segments of a quasi-stationary nature with an implied infinite periodicity. Furthermore, in the digital analysis of waveforms, only a finite number of samples of the signal is recorded on which a spectral analysis is made. Thus even stationary signals are viewed from limited time data and this can introduce errors in the frequency spectrum of the signal.

The effect of windowing can best be seen by defining a time domain function which lies within finite time limits. Outside of these, the function is zero. The simplest window function is the rectangular window of Figure 2.13(a). The frequency spectrum of this function, obtained in Section 2.6, is also included.

The application of a window function has the effect of multiplying each point of a time domain signal by the corresponding time point of the window function. Thus, within a rectangular window, the signal is just itself, but outside of this the signal is completely attenuated although a periodicity of the signal within the window is implied outside the defined window. This time domain multiplication has its equivalent in the frequency domain as the convolution of the spectra of the window function and the



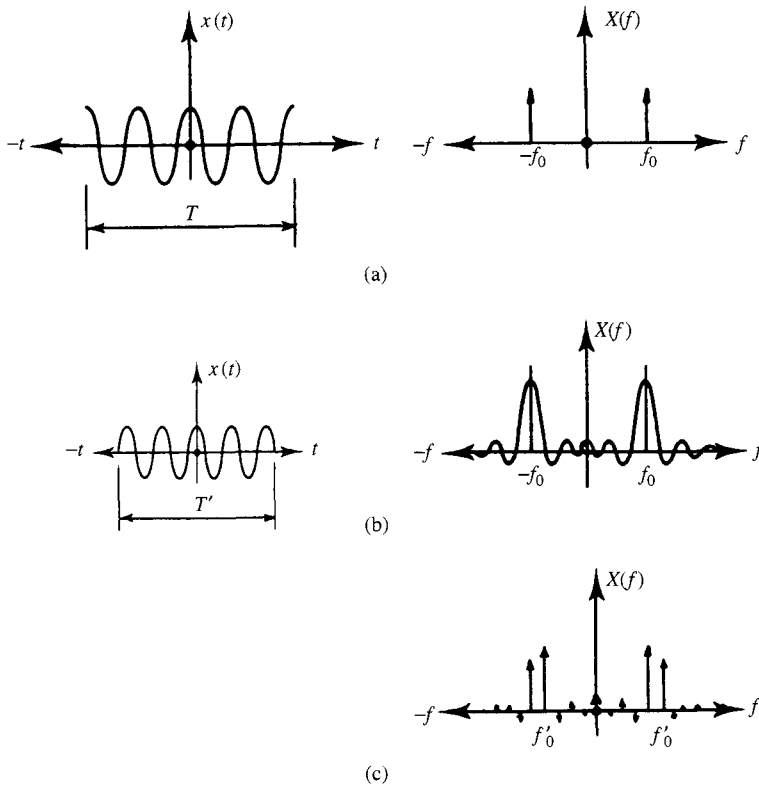
**Figure 2.13** Infinite periodic function processed with a rectangular window function: (a) rectangular window function and frequency spectrum; (b) periodic function  $x(t) = A \cos(4\pi t/T)$  and frequency spectrum; (c) infinite periodic function viewed through a rectangular time window

signal. This is illustrated for an infinite periodic function and a rectangular window function in Figure 2.13(b) and (c). It can be observed that there is significant power in the frequencies of the side lobes about the fundamental frequency, which is not present in the infinite fundamental frequency waveform.

“In this sample case where the signal  $x(t)$  is of fundamental frequency  $f_1$  only, with a sampled frequency spectrum, the spectral components of the function  $x(t)$ .  $W(t)$  are integer multiples of  $f_1$  and lie at the zero crossings of  $X(f)$ . Hence the only spectral component which contributes is that being evaluated, i.e. the component at  $f_1$ .”

However, it is highly likely that waveforms will be made up of many frequency components, not necessarily integer multiples of the fundamental window frequency  $f_1$ . Consequently, discontinuities will exist between the function at the start and finish of the window, which will introduce uncertainty in the identification of the periodic components present, since Fourier analysis assumes periodicity of functions and continuity at the boundaries. The resulting error is known as spectral leakage and is the non-periodic noise contributing to each of the periodic spectral components present.

As an illustration of spectral leakage with respect to a single-frequency periodic waveform Fig. 14(a) shows the case where the time window is an exact multiple of the waveform period, 14(b) the phase discontinuity for a worst case sample and 14(c) when viewed as a discrete frequency spectrum. Besides the main frequency component there are high side lobes and d.c.



**Figure 2.14** Infinite periodic signal viewed during different duration time windows with periodicity implied: (a) the time window is an exact multiple of the period of the waveform; (b) the time window is a  $(2n + 1)/2$  multiple of the period of the waveform; (c) case (b) viewed as a discrete frequency spectrum

### 2.11.1 The Picket Fence

The combination of the DFT and window function produces a response equivalent to filtering the time domain signal through a series of filters with centre frequencies at integer multiples of  $1/T$ , where  $T$  is the sampling period. The filter characteristic and the associated leakage are determined by the particular window function chosen. The resulting spectrum can therefore be considered as the true spectrum viewed through a picket fence with only frequencies at points corresponding to the gaps in the fence being visible.

When the signal being analysed is not one of these discrete, orthogonal frequencies, then, because of the non-ideal nature of the DFT filter, it will be seen by more than one such filter, but at a reduced level in each. The effect can be reduced by adding a number of zeros, usually equivalent to the original record length, to the data to be analysed. This is called zero padding. This effective increase in the sampling period  $T$  introduces extra DFT filters at points between the original filters. The bandwidth of the individual filters still depends upon the original sample period and is therefore unchanged.



### 2.11.2 Spectral Leakage Reduction

The effect of spectral leakage can be reduced by changing the form of the window function. In particular, if the magnitude of the window function is reduced towards zero at the boundaries, any discontinuity in the original waveform is weighted to a very small value and thus the signal is effectively continuous at the boundaries. This implies a more periodic waveform, which has a more discrete frequency spectrum.

A number of window functions are shown in Figure 2.15, along with their Fourier transforms, which give a measure of the attenuation of the side lobes that give rise to spectral leakage.

### 2.11.3 Choice of Window Function

The objective in choosing a window to minimise spectral leakage is to obtain a main-lobe width which is as narrow as possible, so that it only includes the spectral component of interest, with minimal side-lobe levels to reduce the contribution from interfering spectral components. These two specifications are interrelated for realisable windows and a compromise is made between main-lobe width compression and side-lobe level reduction.

The rectangular window function, defined by

$$W(t) = \begin{cases} 1 & \text{for } -T/2 < t < T/2 \\ 0 & \text{otherwise} \end{cases} \quad (2.70)$$

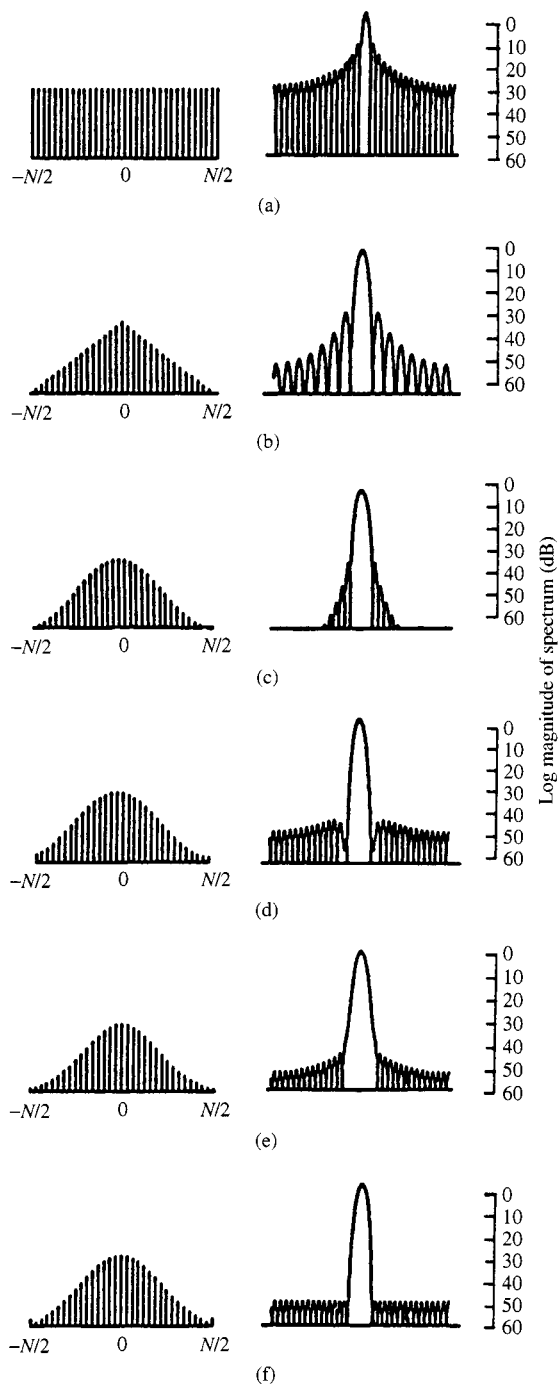
has a noise or effective bandwidth of  $1/T$ , where  $T$  is the window length; the side-lobe levels are large ( $-13$  dB from the main lobe for the first side lobe), and their rate of decay with frequency is slow (being 20 dB per decade). This means that when evaluating the fundamental component of a signal, interfering spectral components near to it will be weighted heavily, contributing greater interference to the fundamental component than for the other windows illustrated in Figure 2.15.

However, as mentioned previously, there is one situation where the rectangular window ideally results in zero spectral leakage and high spectral resolution. This situation occurs when the duration of the rectangular window is equal to an integer multiple of the period of a periodic signal. When the rectangular window spans exactly one period, the zeros in the spectrum of the window coincide with all the harmonics excepting one. This results in no spectral leakage under ideal conditions. Consequently, spectrum analysers often incorporate the rectangular window function facility for the analysis of periodic waveforms, to which the duration of the window can be matched. This is achieved through the use of a phase-locked loop. This frequency matching gives the greatest resolution of the periodic frequency.

The triangular window, defined by

$$W(t) = \begin{cases} 1 + 2t/T & \text{for } -T/2 < t < 0 \\ 1 - 2t/T & \text{for } 0 < t < T/2 \\ 0 & \text{otherwise} \end{cases} \quad (2.71)$$

is a simple modification of the rectangular window, where the amplitude of the multiplying window is reduced linearly to zero from the window centre. The reduction of



**Figure 2.15** Fourier transform pairs of common window functions (a) rectangular; (b) triangular; (c) cosine squared (Hanning); (d) Hamming; (e) Gaussian; (f) Dolph-Chebyshev

the side-lobe is readily seen in Figure 2.15(b), but this is at the expense of main-lobe width and a consequent reduction in frequency resolution.

An international standard window function often incorporated into spectrum analysers is the cosine-squared or Hanning window, defined by

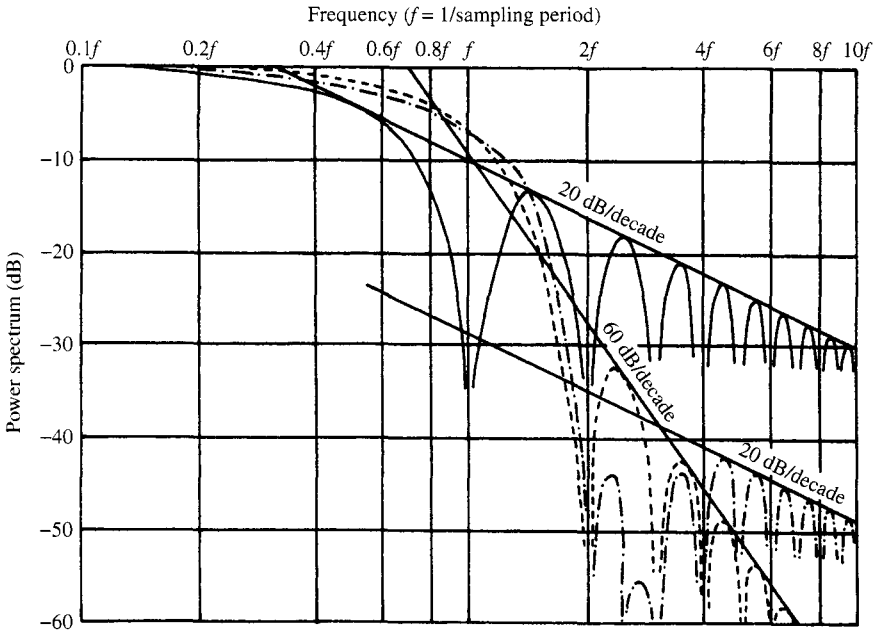
$$W(t) = \frac{1}{2} \left( 1 - \cos \frac{2\pi t}{T} \right) \quad \text{for } -\frac{T}{2} < t < \frac{T}{2} \quad (2.72)$$

and in which it is the power term that is cosine squared. This function is easily generated from sinusoidal signals and in FFT analysers a table of cosine values can be utilised for generating the window. The main lobe noise bandwidth is greater than that for the rectangular window, being  $1.5T$ ; however, the highest side lobe is at  $-32$  dB and the side-lobe fall-off rate is 60 dB per decade, thus reducing the effect of spectral leakage. This is illustrated in Figure 2.16, where the Hanning window is compared to the rectangular window for side-lobe level reduction.

By mounting the Hanning window on a small rectangular pedestal (but limiting the maximum of the function to unity), the Hamming window is obtained. This is described in its amplitude form as

$$W(t) = 0.54 - 0.46 \cos \frac{2\pi t}{T} \quad \text{for } -\frac{T}{2} < t < \frac{T}{2} \quad (2.73)$$

The second side lobe of the rectangular function coincides with the first side lobe of the Hanning function, and since these are in opposite phase, they can be scaled to cancel



**Figure 2.16** Power spectrum versus log(frequency) for selected window functions. (—), Rectangular window; (---), Hanning window; (-·-·-), Hamming window

each other. As a consequence, the highest side-lobe level is  $-42$  dB (Figure 2.16). The remaining side lobes are dominated by the rectangular function and have a fall-off rate of 20 dB per decade. A slight improvement in main-lobe noise bandwidth (to  $1.4/T$ ) is observed also.

The ideal window, which has a single main lobe and no side lobes, is in the form of a Gaussian function:

$$W(t) = \exp(-t^2/2\sigma^2) \quad (2.74)$$

The Gaussian function has the property of transforming, by the Fourier transform, to another Gaussian function. On a decibel scale, its shape is that of an inverted parabola, with a characteristic which becomes successively steeper. Theoretically, the Gaussian function is defined between infinite time limits. For practical use, the function is truncated at three times the half-amplitude width, which is 7.06 times the standard deviation. As a consequence, side lobes are established in the power spectrum but these are of the order of  $-44$  dB down. The main-lobe noise bandwidth is wider than the previous windows, being  $1.9/T$ .

The final window, presented in Figure 2.15(f), is the Dolph-Chebyshev function, the discrete form of which is defined as

$$W(r) = \frac{(-1)^r \cos[N \cos^{-1}[\beta \cos(\pi r/N)]]}{\cosh[n \cosh^{-1}(\beta)]} \quad \text{for } 0 < r < N - 1 \quad (2.75)$$

where  $r$  is an integer,  $N$  is the number of discrete samples of the window function,

$$\beta = \cosh \left[ \frac{1}{N} \cosh^{-1}(10^\alpha) \right]$$

and the inverse hyperbolic cosine is defined by

$$\cosh^{-1} x = \begin{cases} \pi/2 - \tan^{-1}[x/\sqrt{(1-x^2)}] & \text{for } |x| < 1.0 \\ \ln[x + \sqrt{(x^2-1)}] & \text{for } |x| > 1.0 \end{cases}$$

This function provides the narrowest possible main-lobe width for a given specified side-lobe level, which is constant on a decibel scale. The side-lobe levels are controlled by the parameter in equation (2.75). With  $\alpha = 4.0$  the side lobes are at  $-80$  dB (0.01%) with respect to the main lobe.

### 2.11.4 Main-Lobe Width Reduction

In obtaining low side-lobe levels to reduce spectral leakage, in all the window functions previously mentioned, there has been a sacrifice of main-lobe bandwidth. It is possible that in harmonic analysis, when evaluating, for example, the fundamental of a waveform, the resolution is such that the d.c. component and the second and third harmonics are included within the main lobe. This causes considerable interference in the individual harmonic evaluations and restricts the identification of spectral leakage effects to higher order frequencies and noise, outside the main lobe.

However, with the ability to change the side-lobe level with the Dolph-Chebyshev window, an algorithm presents itself to effectively reduce the main-lobe bandwidth.

Consider the complex fundamental Fourier component of the waveform, multiplied by the Dolph-Chebyshev window function, i.e.  $W(r) \cdot x(r)_{r=1}^N$ , obtained by the application of the DFT (using the FFT technique):

$$X_1 = W_0 C_1 + W_1 C_2 + W_2 C_3 + W_{-2} C_{-1} + W_{-3} C_{-2} + W_{-4} C_{-3} + W_{-1} C_0 \\ + \sum_{n=3}^{N/2} W_n C_{n+1} + \sum_{n=4}^{N/2-1} W_{-n+1} C_{-n} \quad (2.76)$$

where  $W$  are the discrete window coefficients in the frequency domain and  $C_n = C_{-n}$  are the complex periodic Fourier coefficients of harmonic order  $n$ .

By pre-processing the waveform, the d.c. component can be removed, thereby eliminating  $C_0$  in equation (2.76). In addition, since the last two terms of this equation include all the higher-order harmonics and noise, but are weighted with window coefficients of the order of 0.01%, they can also be neglected. Equation (2.76) can therefore be reduced to:

$$X_1 = C_1(W_0 + W_{-2}) + C_2(W_1 + W_{-3}) + C_3(W_2 + W_{-4}) \quad (2.77)$$

Application of the DFT to the windowed discrete time-domain waveform yields a value for  $X_1$ . The window coefficients in the frequency domain are known by the defining equation (2.73) and hence only the three harmonic terms  $C_1, C_2, C_3$  are known.

The use of three windows, each with a different  $\alpha$  parameter, and the application of the DFT three times to the same waveform, produces three simultaneous equations of the same form as equation (2.77). The solution of these leads directly to the values of  $C_1, C_2$  and  $C_3$ , i.e. the fundamental component and second and third harmonics of the original waveform.

As an illustration of the effectiveness of this algorithm, consider the function defined by

$$x(t) = C_1 \cos(2\pi f_1 t + \phi_1) + \sum_{n=2}^7 C_n \cos(n 2\pi f_1 + \phi_n) \quad (2.78)$$

where  $C_1 = 1.0$  and  $C_n = 0.2$  for  $n = 2$  to  $7$ , for which 32 samples were available.

With window filtering the error introduced by the higher harmonics in the identification of  $C_1$  was limited to 0.16%. However, when a non-periodic component of frequency  $5.65 f_1$  and magnitude 0.2 was introduced into  $x(t)$ , the error in identifying  $C_1$  without window filtering was 2%. With the Dolph-Chebyshev window and using  $\alpha$  values of 3.2, 3.5 and 3.8, the error was reduced to 0.36%.

### 2.11.5 Application to Inter-Harmonic Analysis [10]

If frequencies not harmonically related to the sampling period are present or the waveform is not periodic over the sampling interval, errors are encountered due to spectral leakage.

Section 2.11 has shown that the method normally used to minimise spectral leakage and obtain accurate magnitude and frequency information involves the use of windows. Windowing functions weight the waveform to be processed by the FFT in such a way as to taper the ends of the sample to near zero.

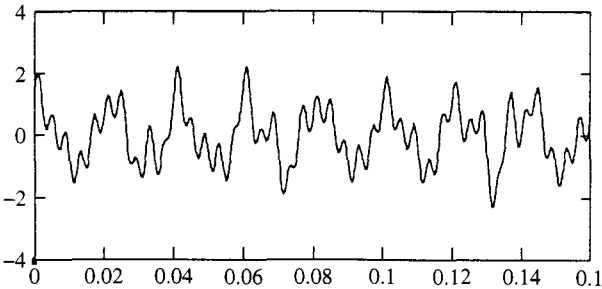
Figure 2.17 illustrates a waveform containing the six steady-state frequency components shown in Table 2.1. The resulting waveform is not periodic and appears even asymmetric depending on the observation interval.

Applying the Hanning window to the waveform of Figure 2.17 produces the waveform shown in Figure 2.18 and spectrum shown in Figure 2.19.

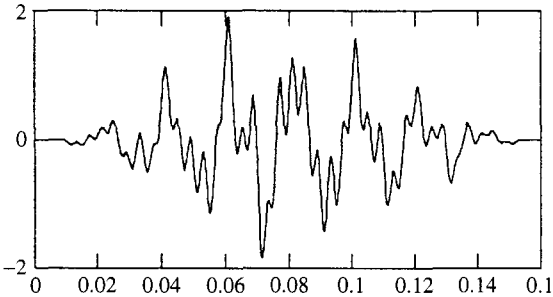
Even with the use of windowing functions, closely spaced inter-harmonic frequencies are hard to determine due to the resolution of the FFT as determined by the original

**Table 2.1** Frequency components of example system

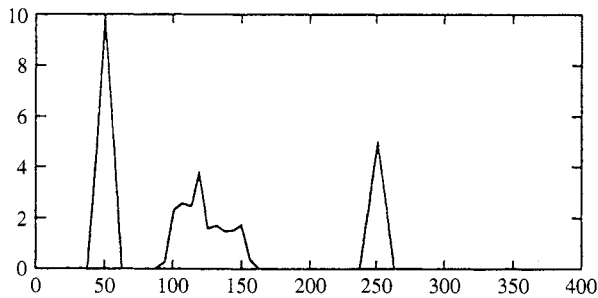
Frequency	Magnitude
50	1.0
104	0.3
117	0.4
134	0.2
147	0.2
250	0.5



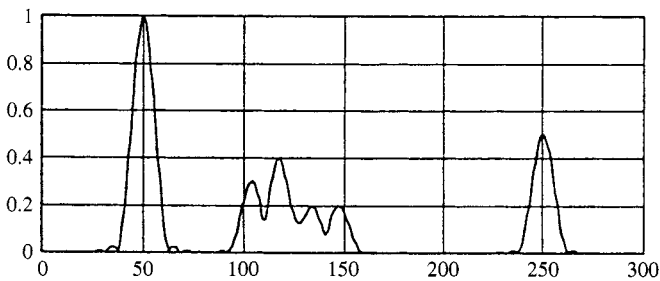
**Figure 2.17** Waveform with harmonic and inter-harmonic components



**Figure 2.18** Result of Hanning window



**Figure 2.19** FFT spectrum of Figure 2.17



**Figure 2.20** Result of FFT analysis of Figure 2.19 with four-fold zero padding

sampling period, which is 8 cycles of 50 Hz in this case. The use of the zero padding technique can result in a much more accurate determination of the actual inter-harmonic frequency component magnitudes and frequencies. Figure 2.20 shows the results of applying a four-fold zero padding before performing the FFT.

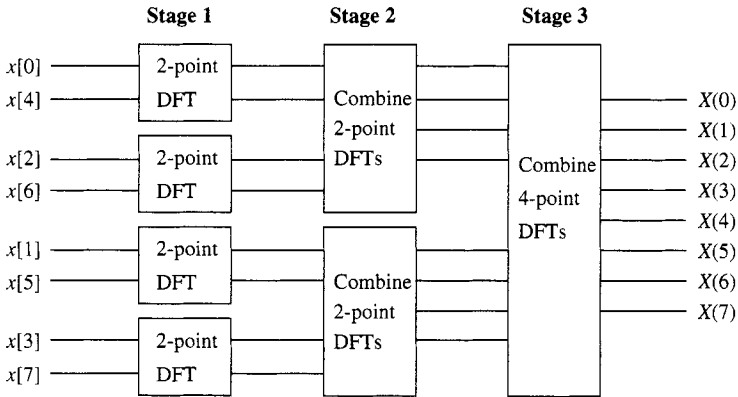
Note that the resolution has been improved enough to accurately determine the magnitude and frequency of each component even though the sampled waveform was not periodic and some of the inter-harmonic components are very close to each other.

## 2.12 Efficiency of FFT Algorithms

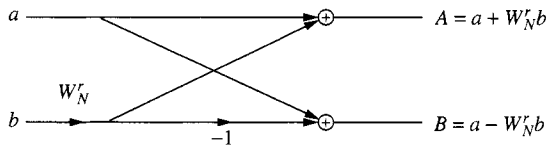
### 2.12.1 The Radix-2 FFT

The complex radix-2 [11] is the standard FFT version and is usually available in DSP libraries. A number of alternative algorithms, such as the higher radix, mixed radix and split radix, have been developed but the radix-2 is still widely used. It relies on a decomposition of the set of  $N$  inputs into successively smaller sets on which the DFT is computed until sets of length 2 remain. Figure 2.21 represents this approach, which requires that the number of input points  $N$  is a power of two ( $N = 2^v$ ). The decomposition in this way can then be performed  $v$  times.

Each of the 2-point DFTs consists of a *butterfly* computation as depicted in Figure 2.22. It involves the two complex numbers  $a$  and  $b$ , where  $b$  is multiplied by the complex phase factor and then the product is added and subtracted from  $a$ .



**Figure 2.21** Decomposition of an 8-point DFT into 2-point DFTs by the radix-2 FFT algorithm



**Figure 2.22** Butterfly computation with twiddle factor  $W_N^r$

One butterfly hence requires one complex multiplication and two complex additions. Each of the stages consists of  $N/2$  butterflies. Thus to compute the radix-2 FFT,  $\nu N/2$  butterflies are required. The total number of real operations is  $5\nu N$ , consisting of  $2\nu N$  multiplications and  $3\nu N$  additions.

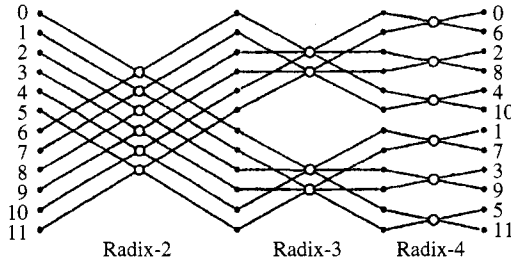
### 2.12.2 Mixed-Radix FFT

The sampling rate of measurement equipment (and step-length for time-domain simulation) is not often chosen to result in  $2^n$  sample points for one period of the fundamental frequencies of 50 Hz or 60 Hz. With the increase in computer power, the mixed-radix FFT is useful and practical for power system studies, giving flexibility in transform data size and sampling rate, and eliminating the leakage problem that often occurs with radix-2 FFT.

FFT algorithms are not limited to the radix-2 family of algorithm's only. Reference [12] describes an improvement of these algorithms where radix-2 and radix-4 routines are mixed to produce a more efficient FFT. Single-radix FFTs such as the radix-3 and the radix-5 require 3, 9, 27, 81, 243, ... data points and 5, 25, 125, 625, ... data points, respectively. Algorithms based on routines higher than radix-10 require too many points for efficient computation.

More flexibility in the selection of data size is therefore provided by the mixed-radix FFT. For example, a 2000-point DFT can be achieved using a mixed-radix FFT





**Figure 2.23** 12-point  $2 \times 3 \times 2$  FFT flow chart

to perform the DFT in  $2 \times 2 \times 2 \times 2 \times 5 \times 5 \times 5$  or  $2 \times 10 \times 10 \times 10$ . Single-radix FFTs for radices 2, 3, 4, 5, 6, ... can be stacked in any order to accommodate the desired number of data points. Figure 2.23 gives a simple flow chart showing a 12-point ( $2 \times 3 \times 2$ ) FFT.

### 2.12.3 Real-Valued FFTs

FFT algorithms are designed to perform complex multiplications and additions but the input sequence may be real. This can be exploited to compute the DFT of two real-valued sequences of length  $N$  by one  $N$ -point FFT. This is based on the linearity of the DFT and on the fact that the spectrum of a real-valued sequence has complex conjugate symmetry, i.e.

$$X(k) = X^*(N - k), \quad k = 1 \dots \frac{N}{2} - 1 \quad (2.79)$$

while  $X(k)$  and  $X(N/2)$  are real. A complex-valued sequence  $x[n]$  defined by two real-valued sequences  $x_1[n]$  and  $x_2[n]$  such that

$$x[n] = x_1[n] + jx_2[n] \quad (2.80)$$

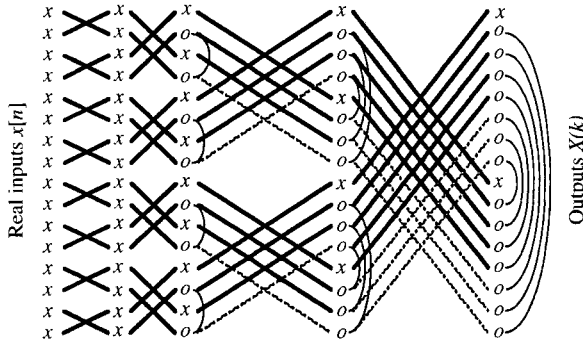
has a DFT that, due to the linearity of the transform, may be expressed as

$$X(k) = X_1(k) + jX_2(k) \quad (2.81)$$

The DFTs of the original sequences are then given by

$$\begin{aligned} X_1(k) &= \frac{1}{2}[X(k) + X^*(N - k)] \\ X_2(k) &= \frac{1}{(2j)}[X(k) - X^*(N - k)] \end{aligned} \quad (2.82)$$

The extra amount of computation needed to recover the DFTs according to equation (2.82) is small. Since  $X_1(k)$  and  $X_2(k)$  represent the DFTs of real sequences, they must have complex conjugate symmetry (equation (2.79)) and only the values for  $k = 0 \dots N/2$  need to be computed. For  $k = 0$  or  $k = N/2$  the result will be real. Thus for two length  $N$  real-valued sequences, the total number of computations amounts to



**Figure 2.24** Radix-2 FFT of length 16 with real inputs. Real values are indicated by  $x$ , complex ones by  $o$ . The complex values connected by arcs are conjugates of each other. The solid lines represent the butterflies that need to be computed. Reproduced from [13]

one  $N$ -point FFT plus  $2N - 4$  extra additions, which essentially means that the number of operations compared to the standard FFT is halved by this technique.

An even more efficient technique [13] is based on the fact that complex conjugate symmetry not only exists at the output of the FFT algorithm (coefficients  $X(k)$ ) but also at every stage. This is shown in Figure 2.24 for a 16-point FFT with real-valued inputs. Thus the complex conjugate values need not be calculated (saving five butterflies in Figure 2.24).

Although the concept can be applied to any kind of FFT algorithm, the *split-radix* FFT algorithm is recommended because it requires fewer operations than a radix-2 FFT, a radix-4 FFT or any higher radix FFT. It is called ‘split-radix’ because an  $N$ -point DFT is broken up into a length  $N/2$  DFT over the inputs with even indices and two length  $N/4$  DFTs over the inputs with odd indices. This scheme is then iterated through all stages of the transform. A corresponding real-valued inverse split-radix FFT exists (in this case the outputs being real-valued).

The computational complexity of the real-valued split-radix FFT is reduced to:

$$\begin{aligned}
 \#_{\text{mul}} &= \frac{2}{3}vN - \frac{19}{9}N + 3 + \frac{(-1)^v}{9} \\
 \#_{\text{add}} &= \frac{4}{3}vN - \frac{17}{9}N + 3 - \frac{(-1)^v}{9} \\
 \#_{\text{total}} &= 2vN - 4N + 6
 \end{aligned} \tag{2.83}$$

where  $2^v = N$ .

#### 2.12.4 Partial FFTs

FFT algorithms generally compute  $N$  frequency samples for a length  $N$  input sequence. If only a narrow band of the spectrum is of interest, i.e. fewer than the  $N$  outputs are needed, there are methods to reduce the complexity of the FFT accordingly.

One way is to compute the desired DFT points directly using digital filters, which resonate at the corresponding frequencies [14]. Another well-known technique is *FFT pruning* [15] where the branches in the tree-like FFT flow graph (see Figure 2.24) that lead to unwanted outputs are removed. Instead of a full set of  $N$  points, a smaller subset of  $L$  points, which must occur in a sequence, can be calculated.

A more flexible and efficient method is *transform decomposition* (TD) [16], where  $S$  output points, which need not be in a sequence, are computed by decomposing the  $N$ -point DFT into  $Q$   $P$ -point DFTs ( $N = PQ$ ). Subsequently, each of the  $P$ -point DFTs is computed and recombination (multiplication by the phase factors and summation) leads to the desired  $S$  output points. This process is shown in the block diagram of Figure 2.25 for an 8-point DFT. Any FFT algorithm may be used and therefore the real-valued split-radix algorithm described in the previous section is the best choice.

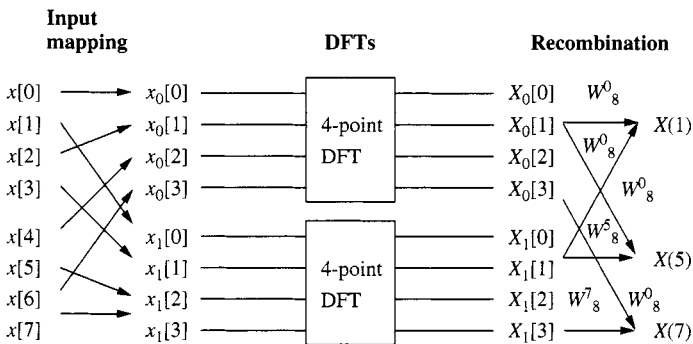
The overhead for the transform decomposition amounts to:

$$\begin{aligned}\#_{\text{mul}} &= 4QS \\ \#_{\text{add}} &= 4QS - 2S \\ \#_{\text{total}} &= 8QS - 2S\end{aligned}\tag{2.84}$$

To obtain the total number of operations for the TD, the computations for the  $Q$  length  $P$  split-radix have to be added. These follow from equation (2.83), RFFTs where the operation counts of an  $N$ -point split-radix RFFT are given, and the total becomes:

$$\#_{\text{total}} = 2N[\log_2 P - 2 + (3 + 4S)/P] - 2S\tag{2.85}$$

Table 2.2 lists the number of operations required by each of the FFT algorithms discussed in this section to compute the DFT for a number of inputs and outputs. The savings achieved by the TD are more than 75%.



**Figure 2.25** Transform decomposition of an 8-point DFT ( $Q = 2$ ,  $P = 4$ ). Only  $S = 3$  output points are computed

**Table 2.2** Number of operations (real multiplications and additions) required for the DFT computation by the discussed FFT algorithms. The last column indicates the savings of the transform decomposition over a standard FFT

Number of DFTs		Number of operations			Reduction (%)
Inputs $N$	Outputs $S$ (0–5 Hz)	Standard FFT	Split-radix RFFT	TD + split-radix RFFT	
512	9	23 040	7 174	4 302	81.3
1024	17	51 200	16 390	10 430	79.6
2048	33	112 640	36 870	24 734	78.0
4096	65	245 760	81 926	57 438	76.6
8192	129	532 480	180 230	131 038	75.4

### 2.13 Alternative Transforms

Traditionally, the Fourier transform has almost exclusively been used in power as well as most other engineering fields, therefore this book concentrates on assessment techniques based on Fourier analysis.

Three principal alternatives have been discussed at great length in recent literature with reference to potential power system applications. These are the Walsh, Hartley and Wavelets transforms.

As in the Fourier case, the Walsh transform [17] constitutes a set of orthogonal functions. It is extremely simple conceptually, as it involves only square wave components, but for accurate results it needs a large number of terms for the processing of power system waveforms. Moreover, it does not benefit from the differential-to-phasor transformation, and handling differentiation and integration, common operations in power systems.

The Hartley transform [17], also using the orthogonal principle, is expressed as

$$F(v) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) \text{cas}(vt) dt \quad (2.86)$$

where

$$\text{cas}(vt) = \cos(vt) + \sin(vt)$$

and  $v$  is identical to the  $\omega$  of Fourier, i.e. radians/second.

Moreover, its inverse, i.e.

$$f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} F(v) \text{cas}(vt) dt \quad (2.87)$$

has exactly the same form, and this leads to simpler software. An important difference between the Fourier and Hartley transforms is that the latter is all real and thus requires only one half of the memory for storage (i.e. one real quantity as compared with one complex quantity of Fourier's). Also, the convolution operation requires only one real multiplication as compared with four multiplications in the Fourier domain.

However, the Fourier transform is widely spread throughout the power system field and permits a very convenient assessment of magnitude and phase information. The latter is not always required, however, and in such cases the efficiency of the Hartley transform may be sufficiently attractive for it to be used as an alternative to the traditional philosophy.

Owing to the interest generated in the potential applications of wavelets, this transform is given special consideration in the next section.

### 2.13.1 The Wavelet Transform

The WT, originally derived to process seismic signals, provides a fast and effective way of analysing non-stationary voltage and current waveforms. As in the Fourier case, the Wavelet transform decomposes a signal into its frequency components. Unlike the Fourier transform, the wavelet can tailor the frequency resolution, a useful property in the characterisation of the source of a transient.

The ability of wavelets to focus on short time intervals for high-frequency components and long intervals for low-frequency components improves the analysis of signals with localised impulses and oscillations, particularly in the presence of a fundamental and low-order harmonics.

A wavelet is the product of an oscillatory function and a decay function. A *mother* wavelet is expressed as [18]

$$g(t) = e^{-at^2} e^{j\omega t} \quad (2.88)$$

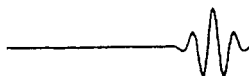
An example of a mother wavelet is shown in Figure 2.26.

A variety of wavelets originating from a mother wavelet can be used to approximate any given function. These wavelets are derived by scaling and shifting (in time) mother wavelets as shown in Figure 2.27, and can be expressed as

$$g'(a, b, t) = \frac{1}{\sqrt{a}} g\left(\frac{t-b}{a}\right) \quad (2.89)$$



**Figure 2.26** A sample mother wavelet



**Figure 2.27** A sample daughter wavelet

The WT of a continuous signal  $x(t)$  is defined as

$$\text{WT}(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) g\left(\frac{t-b}{a}\right) dt \quad (2.90)$$

The time extent of the wavelet  $g(t - b/a)$  is expanded or contracted in time depending on whether  $a > 1$  or  $a < 1$ . A value of  $a > 1$  ( $a < 1$ ) expands (contracts)  $g(t)$  in time and decreases (increases) the frequency of the oscillations in  $g(t - b/a)$ . Hence, as  $a$  is ranged over some interval, usually beginning with unity and increasing, the input is analysed by an increasingly dilated function that is becoming less and less focused in time.

The WT has a digitally implementable counterpart, the discrete wavelet transform (DWT).

In DWT, the scale and translation variables are discretised but not the independent variable of the original signal. It is to be noted that the two variables  $a$  and  $b$  are continuous in the continuous transform. However, in the reconstruction process, the independent variable will be broken down into small segments for ease of computer implementations. A DWT gives a number of wavelet coefficients depending upon the integer number of the discretisation step in scale and translation, denoted by  $m$  and  $n$ , respectively. So any wavelet coefficient can be described by two integers,  $m$  and  $n$ . If  $a_0$  and  $b_0$  are the segmentation step sizes for the scale and translation, respectively, the scale and translation in terms of these parameters will be  $a = a_0^m$  and  $b = nb_0 a_0^m$ .

In terms of the new parameters  $a_0$ ,  $b_0$ ,  $m$  and  $n$ , equation (2.89) becomes

$$g'(m, n, t) = \frac{1}{\sqrt{a_0^m}} g\left(\frac{t - nb_0 a_0^m}{a_0^m}\right) \quad (2.91)$$

or

$$g'(m, n, t) = \frac{1}{\sqrt{a_0^m}} g(ta_0 - nb_0) \quad (2.92)$$

and the discrete wavelet coefficients are given by

$$\text{DWT}(m, n) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{a_0^m}} f(t) g(a_0^{-m} t - nb_0) dt \quad (2.93)$$

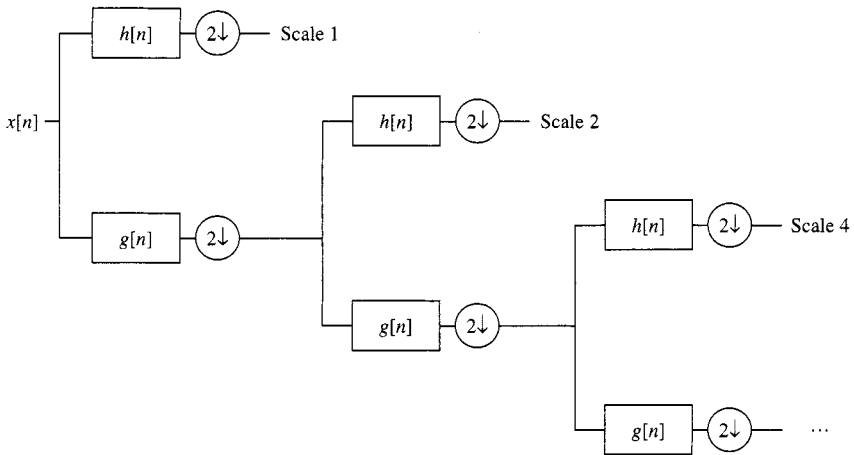
Although the transformation is over continuous time, the wavelets representation is discrete and the discrete wavelet coefficients represent the correlation between the original signal and wavelets for different combinations of  $m$  and  $n$ .

The inverse DWT is given by

$$f(t) = K \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} W_g f(m, n) \frac{1}{\sqrt{a_0^m}} g(a_0^{-m} t - nb_0) \quad (2.94)$$

where  $K = (A + B)/2$ , and  $A$  and  $B$  are the frame bounds (maximum values of  $a$  and  $b$ ).

Wavelet analysis is normally implemented using multi-resolution signal decomposition (MSD). High- and low-pass equivalent filters,  $h$  and  $g$  respectively, are



**Figure 2.28** Multi-resolution signal decomposition implementation of wavelet analysis

formed from the analysing wavelet. The digital signal to be analysed is then decomposed (filtered) into smoothed and detailed versions at successive scales, as shown in Figure 2.28, where  $(2\downarrow)$  represents a down-sampling by half.

Scale 1 in Figure 2.28 contains information from the Nyquist frequency (half the sampling frequency) to one quarter the sampling frequency; scale 2 contains information from one quarter to one eighth of the sampling frequency, and so on.

**Choice of Analysing Wavelets** The decomposition can be halted at any scale, with the final smoothed output containing the information of all of the remaining scales, i.e. scales 8, 16, 32, ... if it is halted at scale 4, one of the desirable properties of MSD.

The choice of mother wavelet is different for each problem at hand and can have a significant effect on the results obtained. Orthogonal wavelets ensure that the signal can be reconstructed from its transform coefficients [19]. Wavelets with symmetric filter coefficients generate linear phase shift, and some wavelets have better time localisation than others.

The wavelet family derived by Daubechies [20] covers the field of orthonormal wavelets. This family is very large and includes members ranging from highly localised to highly smooth. For short and fast transient disturbances, Daub4 and Daub6 wavelets are the best choice, while for slow transient disturbances, Daub8 and Daub10 are particularly good.

However, the selection of an appropriate mother wavelet without knowledge of the types of transient disturbances (which is always the case) is a formidable task. A more user-friendly solution [21] utilises one type of mother wavelet in the whole course of detection and localisation for all types of disturbances.

In doing so, higher-scale signal decomposition is needed. At the lowest scale, i.e. scale 1, the mother wavelet is most localised in time and oscillates most rapidly within a very short period of time. As the wavelet goes to higher scales, the analysing wavelets become less localised in time and oscillate less due to the dilation nature of the wavelet transform analysis. As a result of higher-scale signal decomposition, fast and short

transient disturbances will be detected at lower scales, whereas slow and long transient disturbances will be detected at higher scales.

**Example of Application [21]** Figure 2.29(a) shows a sequence of voltage disturbances. To remove the *noise* present in the waveform, squared wavelet transform coefficients (SWTC) are used at scales  $m = 1, 2, 3$  and 4 (shown in Figure 2.29(b), (c), (d) and (e), respectively); these are analysed using the Daub4 wavelet.

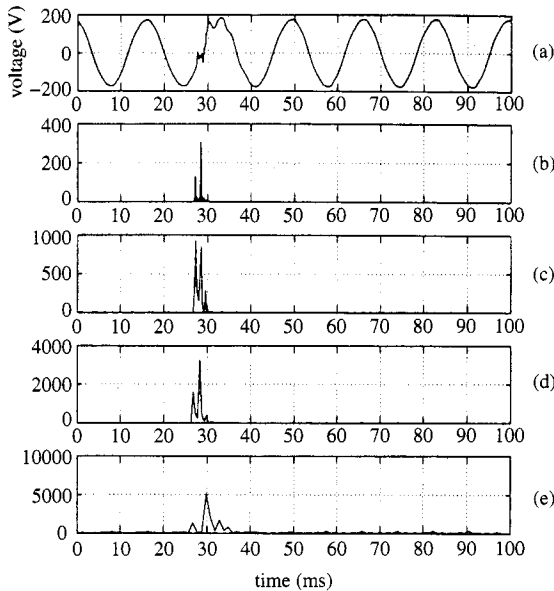
Figure 2.29(a) contains a very rapid oscillation disturbance (high frequency) before time 30 ms, followed by a slow oscillation disturbance (low frequency) after time 30 ms. The SWTCs at scales 1, 2 and 3 catch these rapid oscillations, while scale 4 catches the slow oscillating disturbance which occurred after time 30 ms. Note that the high SWTCs persist at the same temporal location over scales 1, 2 and 4.

It must be pointed out that the same technique can be used to detect other forms of waveform distortion (like notches and harmonics) and other types of disturbance such as momentary interruptions, sags and surges.

However, rigorous uniqueness search criteria must be developed for each disturbance for the wavelet transform to be accepted as a reliable tool for the automatic classification of power quality disturbances.

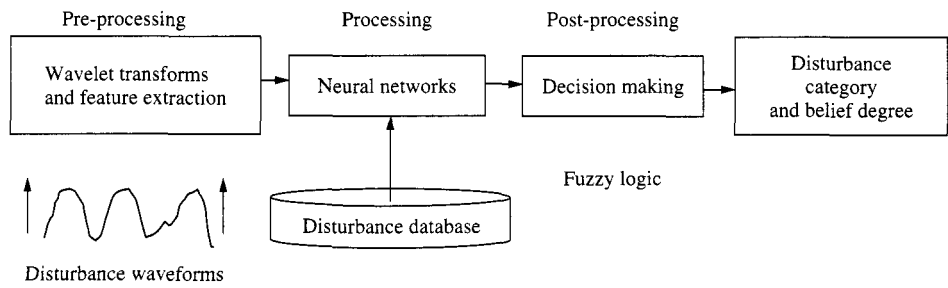
### 2.13.2 Automation of Disturbance Recognition

Some proposals are being made to automate the process of disturbance recognition [22,23] to improve the speed, reliability and ease of data collection and storage.



**Figure 2.29** Wavefault disturbance detection using Daub4: (a) the voltage disturbance signal; (b), (c), (d) and (e) the squared WTCs at scales 1, 2, 3 and 4, respectively. Copyright 1996





**Figure 2.30** Block diagram of the automatic disturbance recognition system

Such a scheme involves the three separate stages illustrated in Figure 2.30. These are a pre-processing stage to extract the disturbance information from the generated power signal, a main processing stage to carry out pattern recognition on the disturbance data, and a post-processing stage to group the output data and form decisions on the possible nature and cause of the disturbance.

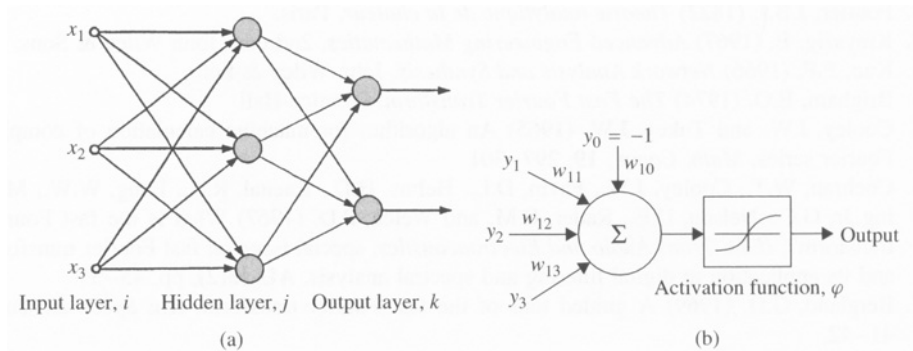
The WT described in Section 2.13.1 is an obvious candidate to extract the disturbance information owing to its greater precision and speed over Fourier methods. A collection of standard libraries of wavelets can be developed to fit specific types of disturbance or transient.

Artificial neural networks can be used in the main processing stage to perform pattern recognition. The neural network can be trained to classify the preliminary information extracted in the pre-processing stage.

The most commonly used type of neural network for pattern recognition is the multi-layered perceptron. These are constructed as shown in Figure 2.31 and are usually trained using the recursive error back-propagation algorithm or a modification thereof [24].

The output of the network is

$$y_k(p) = \varphi_k \left( \sum_j w_{kj}(p) \cdot y_j(p) \right) \tag{2.95}$$



**Figure 2.31** (a) Multi-layered perceptron; (b) Configuration of an individual neuron

Finally, fuzzy logic [25,26] is well suited in the post-processing stage to make decisions on the disturbance category. It is simple and fast to compute.

Fuzzy rules must be derived in order to take the information provided by the neural networks and produce a 'belief' in each disturbance category. These rules take the form of IF ... THEN relations based on human knowledge of the problem.

The output of the disturbance recognition system is produced as one or a list of disturbance categories with an associated degree of belief. A list of disturbance categories with belief degree is necessary, as pattern recognition systems are inexact by nature. The system should, however, produce high belief degrees only for disturbance categories that are likely causes.

## 2.14 Discussion

The Fourier transform is still the most widely utilised signal processing tool in power system harmonic analysis. For efficient computation its use has been enhanced by the development of a variety of algorithms under the general heading of the Fast Fourier transform. The development of effective windows, essential to the analysis of non-stationary signals, as well as inter-harmonic information, is another important addition to Fourier assessment.

The main alternatives to the Fourier transform under consideration in power system analysis are the Walsh, Hartley and wavelets transforms. Although harmonic analysis implies steady-state conditions, the information recorded from actual measurements is not necessarily of that type and thus the wavelet approach, although initially designed for the processing of transients, can also be used to extract harmonic content from practical power system recordings.

Finally, with the greater availability of computing power it is possible to make more extensive use of computer-demanding heuristic techniques, such as neural networks and fuzzy logic, in the processing of power system signals, although at the point of writing these are mainly confined to the technical literature.

## 2.15 References

1. Fourier, J.B.J. (1822) *Theorie analytique de la chaleur*, Paris.
2. Kreyszig, E. (1967) *Advanced Engineering Mathematics*, 2nd edn. John Wiley & Sons.
3. Kuo, F.F. (1966) *Network Analysis and Synthesis*. John Wiley & Sons.
4. Brigham, E.O. (1974) *The Fast Fourier Transform*. Prentice-Hall.
5. Cooley, J.W. and Tukey, J.W. (1965) An algorithm for machine calculation of complex Fourier series, *Math. Comp.*, **19**, 297–301.
6. Cochran, W.T., Cooley, J.W., Favon, D.L., Helms, H.D., Kaenal, R.A., Lang, W.W., Maling Jr, G.C., Nelson, D.E., Rader, C.M. and Welch, P.D. (1967) What is the fast Fourier transform?, *IEEE Trans Audio and Electroacoustics*, special issue on fast Fourier transform and its application to digital filtering and spectral analysis, **AU-15**(2), pp. 45–55.
7. Bergland, G.D. (1969) A guided tour of the fast Fourier transform. *IEE Spectrum*, July, 41–42.
8. Bergland, G.D. (1968) A fast Fourier transform algorithm for real-values series, *Numerical Analysis*, **11**(10), 703–10.

9. Harris, F.J. (1978) On the use of windows for harmonic analysis with the discrete Fourier transform, *Proc. IEEE*, **66**, 51–84.
10. IEEE Task Force on Interharmonics and CIGRE WG 36.05/CIGRE 2-CCO2.
11. Lu, I.D. and Lee, P. (1994) Use of mixed radix FFT in electric power system studies, *IEEE Trans. Power Delivery*, **9**(3), 1276–80.
12. Sorensen, H.V., Heiderman, M.T. and Burrus, C.S. (1986) On computing the split-radix FFT, *IEEE Trans. Acoustics, Speech Signal Process.*, **ASSP-34** 152–6.
13. Sorensen, H.V., Jones, D.L., Heiderman, M.T. and Burrus, C.S. (1987) Real-valued FFT algorithms, *IEEE Trans. Acoustics, Speech Signal Process.*, **35**(6), 849–64.
14. Soertzel, G. (1958) An algorithm for the evaluation of finite trigonometric series, *Am. Math. Monthly*, **65**(1), 34–5.
15. Markel, J.D. (1971) FFT pruning, *IEEE Trans. Audio Electroacoustics*, **19**(4), 305–11.
16. Sorensen, H.V. and Burrus, C.S. (1993) Efficient computation of the DFT with only a subset of input or output points, *IEEE Trans. Signal Process.*, **41**(3), 1184–1200.
17. Heydt, G.T. (1991) *Electric Power Quality*, Stars in a Circle Publications, West LaFayette.
18. Ribeiro, P.F., Haque, T., Pillay P. and Bhattacharjee, A. (1994) Application of wavelets to determine motor drive performance during power systems switching transients, *Power Quality Assessment*, Amsterdam.
19. Chui, C.K. (1992) *An Introduction To Wavelets*, Academic Press, pp. 6–18.
20. Daubechies, I. (1988) Orthonormal bases of compactly supported wavelets, *Comm. Pure Appl. Math.*, **41**, 909–96.
21. Santoso, S., Powers, E.J., Grady W.M. and Hofmann, P. (1996) Power quality assessment via wavelet transform analysis, *IEEE Trans. Power Delivery*, **11**(2), 924–30.
22. Ringrose, M. and Negnevitsky, M. (1998) *Automated disturbance recognition in power systems*, Australasian Universities Power Engineering Conference (AUPEC'98), Hobart, pp. 593–7.
23. Ribeiro, P.F. and Celio, R. (1994) Advanced techniques for voltage quality analysis: unnecessary sophistication or indispensable tools, paper A-206. *Power Quality Assessment*, Amsterdam.
24. Haykin, S. (1994) *Neural Networks: A Comprehensive Foundation*. Macmillan, pp. 138–229.
25. Zadeh, L. (1965) Fuzzy sets, *Inform. Control*, **8**(3), 338–54.
26. Tanaka, K. (1997) *An Introduction to Fuzzy Logic for Practical Applications*. Springer.