CHAPTER 5

TOOLS USED IN THE PRESENT RESEARCH WORK

5.1 INTRODUCTION TO THE TOOLS USED:

In the present work the analog noise signals generated due to the low frequency noise are processed using several tools. The measuring systems can be broadly classified into the following categories:

1. Wide-band Low-noise Pre-amplifiers specially fabricated for amplification of noise signals comprise of three stages using IC . The construction of the amplifier and its performance evaluation is undertaken jointly by the present investigator and a team of three who are working for special studies of noise generation in thin films and other physical devices. The circuit is presented in Aannexure 1, along with necessary details.

2. The amplified output of the noise signal is analysed using a 12 bit A/D converter supplied by Keonics India Ltd, Bangalore. Due to the availability of good amplifiers in the 'sound recording card' of the multi-media kit of Pentium(Pro) based Zenith PC, we found that this card serves as a good measuring equipment. As a matter of fact the A/D converter is found to introduce extraneous noise in the frequency region of 5 to 6 KHz characteristic of the present converter. The Sound card has been found to be better for 8-bit, 16-bit or 32-bit signal records. The input characteristics of the card can be altered to have wide frequency response, if required. However since our interest is well within the stipulated frequency response region of the card we have not altered the circuitry.

3. The digital outputs recorded using the sound-recorder through WINDOWS 98 software are available in the digital form. The digital data records are amenable to any other software. In the present study the digital noise reoords were analysed using a special software MATLAB 5. The methods of analysis use (a) digital filters (b) fast fourier transform (FFT) analysis and (c) special graphic design available with the MATLAB 5 software. Since these software facilities are commonly available in most of the laboratories, no attempt has been made to elaborately describe the algorithms and programs associated with these special facilities. However, some of the programs specially written for the present program in the form of matlab programs (*.m files) are presented in Annexure 2. The programs presented in Annexure 2 are for the purpose of (a) Digitizing data (b) Time-Signal representation (c) Power Spectral Density Peaks and (d) FFT plots.

The mathematical foundations on which the signal analysis is being performed is briefly illustrated in the following sections - 5.2 Digital Signal Processing and 5.3 FFT Techniques.

5.2.1 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 an 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.

5.2.2. 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 (same 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.

5.2.3 Tools Used in DSP Systems:

The tools required for frequency analysis of continuos 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

$$x(t) = \sum_{k=-\infty}^{\infty}\!\!\!\!\!\!\text{Ó}\;\; c_k e^{j2\eth kf_0 t} \qquad \ldots \qquad (5.1)$$

represents a periodic signal with fundamental period $T_p = 1/f_0$. hence we can think of exponential signals $\{ e^{j2\eth kf_0 t} \quad k = 0, \pm 1, \pm 2, \pm 3 \ldots \}$ as the basic "building blocks" from which we can construct periodic signals of various types by proper of the 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 (5.1) by the complex exponential

$$e^{j2\eth kf_{\bullet} lt}$$

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 a arbitrary but mathematically convenient value [3]. Thus we obtain

$$\int_{t0}^{t0+Tp} x(t)\, e^{-j2\Pi lfot}\, dt = \int_{t0}^{t0+Tp} x(t)\, e^{-j2\Pi lfot} \left(\sum_{k=-\infty}^{\infty} c_k e^{j2\Pi kfot}\right) dt \qquad \ldots \qquad (5.2)$$

To evaluate the integral on the right-hand side of (5.2), we interchange the order of the summation and integration and combine two exponentials. Hence

$$\sum_{=-\infty}^{\infty} c_k \int_{t_0}^{t_0+T_p} e^{j2\Pi(k-l)fot}\, dt = \sum_{k=-\infty}^{\infty} c_k \left[e^{j2\Pi(k-l)fot} \;/\; j2\Pi(k-l)fo \right]_{t_0}^{t_0+T_p} \qquad \ldots \qquad (5.3)$$

for $k \neq l$, the right-hand side of (5.3) evaluated at the lower and upper limits , t_o and t_oDTp, respectively, yields zero. On the other hand, if k = l , we have·

$$\int_{t_0}^{t_0+T_p} dt \qquad = \qquad [\, t \,]_{t_0}^{t_0+T_p}$$

consequently, (5.2) reduces to

$$\int_{t_0}^{t_0+T_p} x(t)\, e^{-j2\Pi lfot}\, dt \quad = \quad c_l\, T_p$$

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

$$c_l = 1/T_p \int_{t_0}^{t_0+T_p} x(t)\, e^{-j2\Pi lfot}\, dt$$

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

$$c_l = 1/T_p \int_{0}^{T_p} x(t)\, e^{-j2\Pi lfot}\, dt \qquad \ldots \qquad (5.4)$$

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

$$\underset{k=-\infty}{\overset{\infty}{\acute{O}}} c_k\, e^{j2\text{Ð}kf_0 t} \qquad \ldots \qquad (5.5)$$

are equal at every value at t. The so called *Dirichlet conditions* guarantee that the eqn 5.5 will b e equal to x(t), except the values of t for which x(t) is discontinuous. At these values of 't ' converges to the mid point of a the discontinuity.

$$\int |\, x(t)\, |\, dt < \infty \qquad \ldots \qquad (5.6)$$

all periodic signals of practical interest satisfies this conditions.

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

$$\int |\, x(t(\ )\, |^2\, dt < \infty' \qquad \ldots \qquad (5.7)$$

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

$$e(t) = x(t) \; - \; \underset{k=-\infty}{\overset{\infty}{\acute{O}}} c_k\, e^{j2\text{Ð}kf_0 t}$$

Although x(t) and its fourier series may not be equal for all values of t, eqn 5.6 implies eqn 5.7 but not vice versa, eqns 5.6, 5.7 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.

5.2.3 POWER DENSITY SPECTRUM OF PERIODIC SIGNALS

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

$$p_x = 1/T_p \int_{T_p} | x(t) |^2 dt < \infty \qquad .. \qquad (5.8)$$

if we take the complex conjugate of (1) and substitute for x(t) in (8) we obtain

$$p_x = 1/T_p \int_{T_p} x(t) \underset{k=-\infty}{\overset{\infty}{\acute{O}}} c_k e^{-j2\eth kf_0 t} dt = \underset{k=-\infty}{\overset{\infty}{\acute{O}}} c_k \left[1/T_p \int_{T_p} x(t)\ e^{-j2\eth kf_0 t} dt \right] \qquad ...(5.9)$$

$$= \underset{k=-\infty}{\overset{\infty}{\acute{O}}} | c_k |^2$$

therefore, we have established the relation

$$p_x = 1/T_p \int_{T_p} | x(t) |^2 dt < \infty = \underset{k=-\infty}{\overset{\infty}{\acute{O}}} | c_k |^2 \qquad ... \qquad (5.10)$$

which is called *Parseval's relation* for signal. To illustrate the physical meaning of eqn 5.10, suppose that xk (t) consists of single complex exponential of the form

$$x(t) = c_k e^{j2\eth kf_0 t}$$

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

$$p_x = \quad | c_k |^2$$

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_o, $k = 0, \pm 1, \pm 2, \pm 3$ ....
the diagram that we obtain shows how the power periodic signal is distributed among the various frequency components. Fig 5.1 illustrated below is known as "power density spectrum" of the

$$p_x = c^2_0 + 2 \overset{\infty}{\underset{k=-\infty}{\acute{O}}} | c_k |^2 \qquad \ldots \qquad (5.11)$$

$$= a^2_0 + 1/2 \overset{\infty}{\underset{k=-\infty}{\acute{O}}} (a^2_k + b^2_k) \qquad (5.12)$$

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

5.2.4 Energy Density spectrum of Aperioidic signals

The energy density spectrum of a 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 parsevals 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, complexed valued [8] consequently, it is usually expressed in the polar form:

$$X(F) = |x(F)| \, e^{j\Theta(F)}$$

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

on the other hand, the quantity $\qquad S_{xx}(F) = \, . \, |x(F)|^2$

which is in the integrand, represent the distribution of energy in the signal as a fiinction 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

$$|x(-F)| = |x(F)| \quad \dots \quad (5.13)$$

$$< x(-F) = - < x(F) \quad \dots \quad (5.14)$$

By combining eqn 5.13 & 5.14 we obtain

$$Sxx(-F) = Sxx(F) \quad \dots \quad (5.15)$$

in other words ,the energy density spectrum of a real signal has even symmetry.

5.2.5 The Sampling Theorem Revisited

To process a continuous time signal using signal processing technique, it is necessary to convert the signal in to 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

$$X(n)=X_a(nT) \qquad -\infty < n < \infty \quad \dots \quad (5.16)$$

The relationship 5.16 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

$$X_a(F) = \int - X_a(t)\, e^{-j2\Pi Ft} dt \quad \dots \quad (5.17)$$

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

$$X_a(t) == \int - X_a(F)\, e^{-j2\Pi Ft} dF \quad \dots \quad (5.18)$$

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:.

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \quad \dots \quad (5.19)$$

or, equivalently,

$$X(f) = \sum_{n=-\infty}^{\infty} x(n)e^{-j2\Pi fn} \quad \dots \quad (5.20)$$

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

$$x(n) = 1/2\Pi \int_{-\Pi}^{\Pi} X(\omega)e^{-j\omega n}\, d\omega \qquad \ldots \qquad (5.21)$$

$$= \int_{-1/2}^{1/2} X(f)e^{j2\Pi fn}\, df \qquad \ldots \qquad (5.22)$$

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

$$t = nT = n/F_s \qquad \ldots \qquad (5.23)$$

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 5.23 into eqn 5.18 yields

$$x(n) \equiv x_a(n\,T) = \int_{-\infty}^{\infty} X_a(F)\, e^{j2\Pi nF/F_s}\, dF \qquad \ldots \qquad (5.24)$$

On comparing eqn 5.22 with eqn 5.24 we obtain

$$\int_{-1/2}^{1/2} X(f)e^{j2\Pi fn} df = \int_{-\infty}^{\infty} X_a(F)\, e^{j2nnF/F_s}\, dF \qquad \ldots \qquad (5.25)$$

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

$$f = F / F_s \qquad \ldots \qquad (5.26)$$

with the aid of (26), we can make simple change variable in eqn 5.25 and obtain the result

$$1/F_s \int_{F_s/2}^{F_s/2} x\,(F/F_s)\; e^{j2\Pi nF/F_s}\; dF = \int_{-\infty}^{\infty} X_a(F)\, e^{j2\Pi nF/F_s}\; dF \qquad \ldots \qquad (5.27)$$

we now turn our attention to the integral on the right-hand side of eqn 27. 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,

$$\sum \int_{\infty}^{\infty} X_a(F)\, e^{j2\Pi nF/Fs}\, dF = \sum_{k=-\infty}^{+\infty} \int_{(k-1/2)F_s}^{(k+1/2)F_s} X_a(F)\, e^{j2\Pi nF/Fs}\, dF \quad \dots \quad (5.28)$$

we observe that Xa(F) is in the frequency interval (K--1/2)Fs to (K+1/2)Fs and is identical to Xa(F-kFs) In the interval -Fs/2 to Fs/2. consequently,

$$\sum_{k=-\infty}^{+\infty} \int_{(k-1/2)F_s}^{(k+1/2)F_s} X_a(F)\, e^{j2\Pi nF/Fs}\, dF = \int_{-F_s/2}^{+F_s/2} \left[\sum_{k=\infty}^{\infty} x_a(F-kF_s) \right] e^{j2\Pi nF/F_s}\, dF \quad \dots \quad (5.29)$$

where we had used the periodicity of the exponential, namely, $e^{j2\Pi n(F+kFs/)/Fs} = e^{j2\Pi nF7Fs}$

This is the desired relationship between the spectrum X(F/Fs)or X(f) of the discrete time signal and the spectrum Xa(F) of the analog signal [11]. the right hand side of eqn 5.30 or eqn 5.31 consists of a periodic repetition of the scaled spectrum FsXa(F) with period Fs. This periodicity is necessary because of spectrum X(f) or X(F/Fs) of the discrete time signal is periodic with period fp =1 or Fp=Fs.

For example, suppose that the spectrum of a band limited analog signal is as shown in fig. The spectrum is zero for $|F| \geq B$. Now,if the sampling frequency Fs is selected to be greeter than 2B,the spectrum X(F/Fs)of the discrete time signal will appear as shown in fig .thus if the sampling frequency Fs is selected such that $Fs \geq 2B$ is the Nquist rate, then

$$X(F/Fs) = FsXa(F) \qquad |F| \leq Fs/2 \quad \dots \quad (5.30)$$

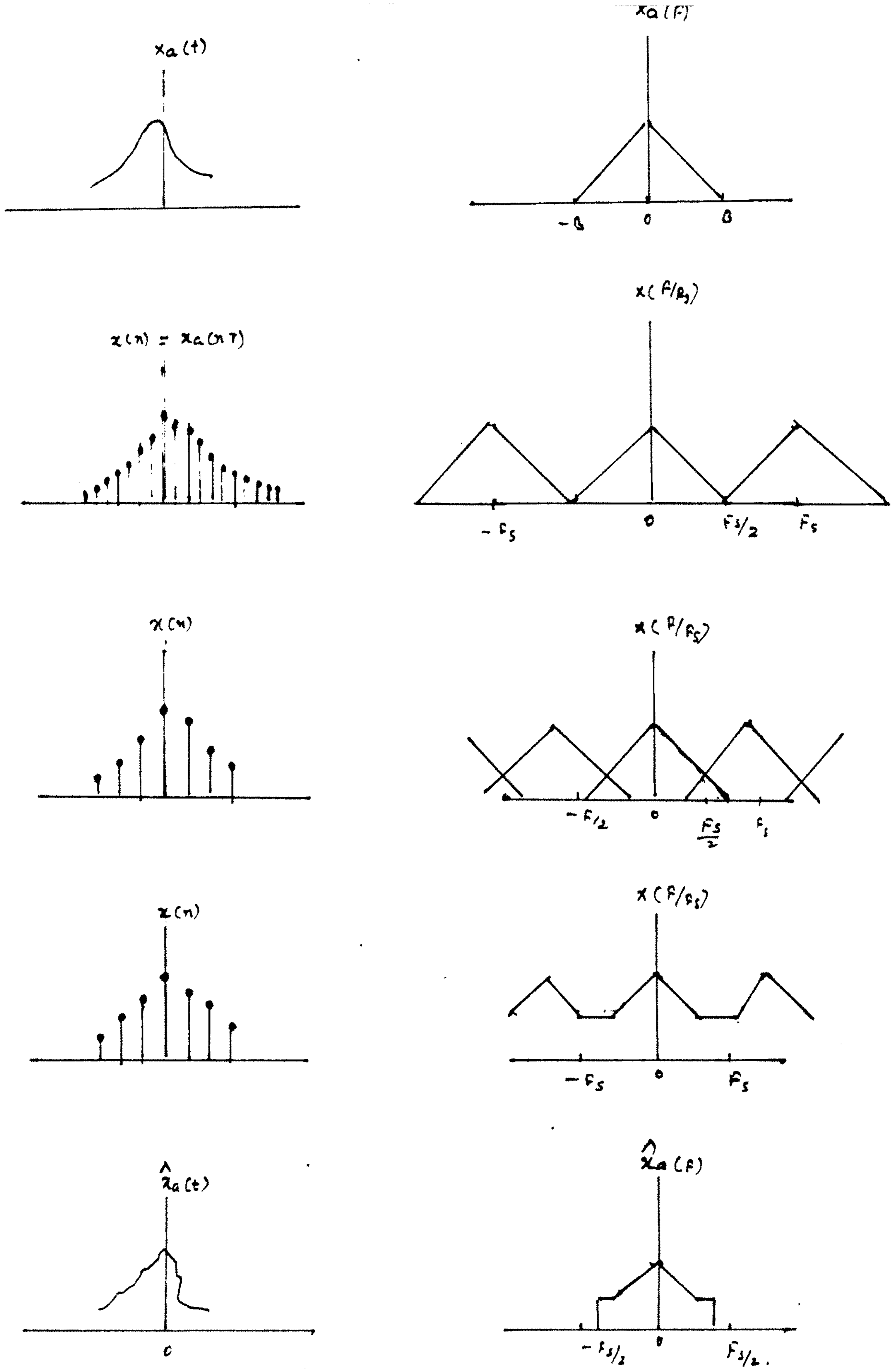
in this case there is no aliasing and therefore, the spectrum of the discrete time signal is identical (with in the scale factor Fs) to the spectrum of the analog signal, with in the fundamental frequency range $|F| \leq Fs/2$ or $|F| \leq Fs/2$or $|f| \leq 1/2$.

On the other hand, if the sampling frequency Fs is selected such that Fs<2B, the periodic continuation ofXa(F) results in spectral overlap, as illustrated in the figure. Thus the spectrum X(F/Fs) of the discrete time signal contains aliased frequency components of the analog signal spectrum Xa(F) .the end result is that the aliasing which occurs prevents us from recovering the original signal Xa(t) from the samples.

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

$$Xa(F) = \{1/Fs\ X(F/Fs), \quad |F| \leq Fs/2 = 0 \qquad |F| > Fs/2 \quad \dots \quad (5.31)$$

and the fourier transform relationship 5.20 yields


$x_a(t)$
$X_a(F)$
$-B$
0
B
$x(n) = x_a(nT)$
$X(F/F_s)$
$-F_s$
0
$F_s/2$
F_s
$x(n)$
$X(F/F_s)$
$-F/2$
0
$\frac{F_s}{2}$
F_s
$x(n)$
$X(F/F_s)$
$-F_s$
0
F_s
$\hat{x}_a(t)$
0
$\hat{X}_a(F)$
$-F_s/2$
0
$F_s/2$.

$$X(F/Fs) = \sum X(n)\, e^{-j2\Pi Fn/Fs} \quad \ldots \quad (5.32)$$

The reconstruction of the original signal is hence possible and useful.

5.3 Fundamentals of Fast Fourier Transfor (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 DFTare presented. Two different approaches are described. One is devide-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.

5.3.2 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 computed 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

$$X(k) = \sum_{n=0}^{N-1} x(n)\, W^{kn}_{N} \qquad 0 \le k \le N-1 \quad \ldots \quad (5.33)$$

Where $W_N = e^{-j2\pi/n}$ ... (5.34)

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

$$X(n) = 1/N \sum_{n=0}^{N-1} x(k)\, W^{-nk}_{N} \qquad 0 \le k \le N-1 \quad \ldots \quad (5.35).$$

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 k, direct computation of X(k) involves N complex multiplication's (4N real multiplications) and N - 1 complex additions (4N - 2real 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 : $W_N^{k+N/2} = -W_N^{k}$ ... (5.36)

Periodicity property : $W_N^{k+n} = W_N^{k}$ ... (5.37)

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

5.3.3 DIRECT COMPUTATION OF THE DFT

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

$$X_R(k) = \sum_{n=0}^{N-1} [\, x_R(n) \cos(2\pi kn/N) + x_I(n) \sin(2\pi kn/N) \,] \quad \ldots \quad (5.38)$$

$$X_R(k) = -\sum_{n=0}^{N-1} [\, x_R(n) \sin(2\pi kn/N) - x_I(n) \cos(2\pi kn/N) \,] \quad \ldots \quad (5.39).$$

The direct computation of (5.38) and (5.39) requires:

1.2N^2 evaluations of trigonometric functions.

2. 4N^2 real multiplications.

3. 4N(N - 1) real additions.

4. A number of indexing and addressing operations.

These operations [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 indexiing and addressing operations are necessary to fetch the data x(n), 0 < = n < = N -1, and the dphase factors and to store the results. The variety of DFT algorithms optimize each of these computational processes in a different way.

5.3.4 *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 cimputationally 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. 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 (6.1.8). Now the sequence x(n), 0< = n< = 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< = l< = L-1 and 0< = m< = M-1 as illustrated in Table 6.1.

Table 6.1 Two Dimensional Data Array For Storing the Sequence x(n), 0<n<N-1

n	0	1	2	—	N-1
	X(0)	X(1)	X(2)	... ...	X(N-1)

row index l \ m column index	0	1	2		M-1
1	x(0,0)	x(0,1)	x(0,2)		x(0,M-1)
2	x(1,0)	x(1,1)	x(1,2)		x(1,M-1)
3	x(2,0)	x(2,1)	x(2,2)		x(2,M-1)
L-1	x(L-1,0)	x(L-1,1)	x(L-1,2)		x(L-1,M-1)

Note that l is the row index and m is the column index. Thus the sequence 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

$$n = M l + m \quad \ldots \quad (5.40)$$

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 6.1. On the other hand, the mapping

$$n = l + mL \quad \ldots \quad (5.41)$$

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 6.1 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 $0< = p < = L - 1$ and $0< \ = q < = M - 1$. If we select the mapping

$$k = Mp + q \quad \ldots \quad (5.42)$$

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

$$k = qL + p \quad . \quad \ldots \quad (5.43)$$

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 (6.1.10) and the row-wise mapping for the DFT given by(5.43). Then

$$X(p,q) = \sum_{m=0}^{N-1} \sum_{l=0}^{l-1} x_K(l,m)\ W_N^{(Mp+q)\,(mL + l)} \quad \ldots \quad (5.44)$$

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 96.1.15). The first step involves the computation of L DFTs, each of M points. Hence this step requires LM2 complex multiplications and LM(M-1) complex additions. The second step requires LM2 complex multiplications. Finally, the third step in the computation requires ML2 complex multiplications and ML(M-1) complex additions. Therefore, the computational complexity is

$$\text{Complex multiplications: } N(M + L + 1)$$

$$\text{Complex additions: } N(M + L - 2) \quad \ldots \quad (5.45)$$

where N = ML. Thus the number of multiplications has been reduced from N2 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 ethat 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

$$N=r1r2...rv \quad \ldots \quad (5.46)$$

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^{lq}_{n}.
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

$$n = M\ l+m$$

$$k = q\,L+p \qquad \ldots \qquad (5.47)$$

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

$$X(p,q) = \sum_{m=0}^{N-1} \sum_{l=0}^{l-1} x(l,m)\; W_N^{pm}\, W_L^{pl}\, W_L^{qm} \qquad \ldots \qquad (5.44)$$

$$X(p,q) = \sum_{m=0}^{N-1} W_M^{qm} \Big[\sum_{l=0}^{l-1} x(l,m)\, W_L^{pl} \Big] W_N^{pm} \qquad \ldots \qquad (5.45)$$

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^{pm}.
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 rrestricted toi be a power of 2 or a power of 4.

5.3.5 APPLICATIONS OF FFT ALGORITHMS

The FFT algorithms desdcribed 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 crosscorrelation of two sequences. The use of the FFT in spectrum analysis is considered In addition we illustrate how to enhance the efficiency of the FFT algorithm by forming complex-valued sequences from real-valued sequences prior to the computation of the DFT.

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 x1 (n) and x2(n) are two real-valued sequences oif length N, and let x(n) be a complex-valued sequence defined as .

$x(n) = x1(n)+jx_2(n) \quad 0<= n<=N-1$ ... (5.46)

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

$X(k) = X_1(k)+jX_2(k)$ ... (5.47)

The sequences x1(n) and x2(n) can be expressed in terms of x(n) as follows:

$x_1(n) =(x(n)+ x(n)^*)/2$

$x_2(n) =(x(n)- x(n)^*)/2$ ... (5.48)

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

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

$X_2(k) = 1/2\{DFT[x(n)]-DFT[x^*(n)]\}$ ... (5.49)

OR $X_1(k) = 1/2\{x(k)]+[x^*(N-k)]$

$X_2(k) = 1/2\{x(k)]-[x^*(N-k)]$ ... (5.50)

Thus, by performing a single DFT on the complex-valued sequence x(n), we have obtained the DFT of the two reral sequences with only a small amount of additional computation that is involved in computing X1(k) andX2(k) from X(k) by use of 5.49 and 5.50.

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

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

$x_1(n) = g(2n)$

$x_2(n) = g(2n+1)$ ... (5.51)

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.

CHAPTER 5 - REFERENCES

1 John G Proakis and Dimitris G Manolakis, "Digital Signal Processing - Principles, Algorithms, and Applications", Prentice Hall of India, New Delhi(1997).
2 Blahut RE, "Fast Algorithms for Digital Signal Processing", Addison-Wesley, Reading(1985).
3 Bluestein LI, "A Linear Filtering Approach to Computation of the Discrete Fourier Transform", IEEE Trans Audio and Electroacoustics, 18, pp451-455(1970).
4 Brigham EO, "The Fast Fourier Transform and its Applications", Prentice Hall, Englewood(1988).
5 Bracewell RN, "The Fourier Transform and its Applications", McGraw Hill, New York(1978).
6 Childers DG (ed),"Modern Spectrum Analysis", IEEE Press, New York(1978).
7 Crochiere RE and Rabinee IR, "Multirate Digital Signal Processing", Prentice & Hall, EC(1983).
8 Cooley JW, Lewis P and Eelch P.D,"The Fast Fourier Transform and its Applications", IEE Trans Education, 12, pp27-34(1969).
9. Constantinides AG,"Spectral Transformation for Digital Filters", Pro IEEE, 117, p1585-1590(1979).
10 Brown JL(Jr),"First Order Sampling of Bandpass Signals-A New Approach", IEEE Trans Information Theory, 26, pp613-615(1980).
11 Bomar BW, "New Second-Order State-Space Structures for Realizing Low Roundoff Noise Digital Filters". IEEE Trans Acoustics, Speech and Signal Processing, 33, pp106-110(1985).
12 Price R, "Mirror FFT and Phase FFT Algoriths", Raytheon Research Div (1990).