ASPECTS OF REAL-TIME DIGITAL SPECTRAL ANALYSIS

by

P.E. WELLSTEAD

A Thesis submitted to the University of Warwick for the degree of Doctor of Philosophy

October, 1970.

PAGE

NUMBERING

AS ORIGINAL

ACKNOWLEDGEMENTS

This thesis is the result of research carried out in the School of Engineering Science, University of Warwick, from April 1968 to September, 1970.

The author is indebted to his supervisor, Dr. M.T.G. Hughes, Senior Lecturer in Automatic Control, for his guidance and help during this period.

Thanks are also due to Mr. J.V. Comfort, for his advice and assistance during the identification experiments, and the members of the Engineering Research Division of British Rail for their permission to use data collected by them.

Finally, the author is grateful to Jo for typing this thesis.

SUMMARY

In the field of control engineering there is a need to study the dynamic behaviour of systems which are subjected to random disturbances. A technique which is of great practical use is to describe the dynamic properties as a function of frequency. This involves determining the frequency content, or spectrum, of the disturbances, and the frequency response function of the system. There are many analogue and digital techniques which are designed for this type of spectral analysis. However, digital computer techniques are often avoided because they are slow, and data must be collected 'off-line'.

A recently discovered computational method, termed the fast-Fourier-transform (FFT), enables digital spectral analysis to be carried-out in a much shorter time than was previously possible. In view of this discovery it was decided to develop digital computer programmes which would overcome the disadvantages of conventional digital spectral analysis. Using these programmes a computer would be connected, via an analogue to digital interface, to the signal source, and would process the data as it entered the computer. In the jargon of computing, the computer would be 'on-line' and analyzing the spectra in 'real-time'.

The first part of the project consisted of an investigation of the FFT when programmed for an on-line digital computer. The results of this investigation showed that a rapid, accurate, and compact FFT could be programmed by using fixed-point arithmetic, and coding in an assembly language. The speed of the transform was sufficient to allow spectral analysis over a frequency range useful in control

applications.

Two on-line computer programmes based upon the FFT were then written; one for 'real-time' spectral analysis of a single record, and another for the 'real-time' estimation of the frequency response function relating two signals. In order that the results of these programmes could be sensibly interpreted, a statistical study was made of the spectral estimators used in the programmes. Arising from this study, several contributions to the field of digital spectral analysis were made. These were :-

1) A more general covariance relationship for cross-spectral estimators.

2) An examination of aliasing in digital spectral estimators.

3) Some theoretical results concerning spectral estimators for closed loop systems with random disturbances inside the loop.

Some experimental work was conducted with the 'real-time' spectral analysis programmes, and it was concluded that the technique is more powerful than conventional digital methods because it is on-line, and can provide estimates with improved resolution and statistical stability. Real-time digital spectral analysis methods also have the advantage that they may be simply and quickly modified to suit specific applications.

CONTENTS

1. INTRODUCTION

This thesis deals with certain problems in the field of digital spectral analysis. In particular it is concerned with a method which enables a digital computer to be used as a real-time spectrum analyser. The spectral estimates discussed in the thesis are suitable for the analysis of random signals, and of linear, time-invariant, systems which are subjected to random disturbances.

The purpose of this chapter is to present some background information, and motivation for the work contained in this thesis. To justify the use of spectral analysis methods on random signals and systems with random inputs, section 1.1. describes some uses of spectral analysis of random data. The spectral analysis methods contained in this thesis are designed for use with the fast-Fourier-transform (FFT). To provide an introduction to these methods, section 1.2. discusses digital spectral analysis methods used before the discovery of the FFT; and section 1.3. describes the effect that the fast-Fourier-transform has had upon spectral calculations. Finally, in sections 1.4, and 1.5. the specific work contained in the thesis is outlined.

1.1. Some Applications of Spectral Analysis

Wherever physical phenomena can be expressed as the reaction of a system to a signal, there is a need to characterise the dynamic behaviour of the system. In many practical cases the system can be assumed to be approximately linear, and time-invariant. This type of system can be completely specified by its dynamic or transient response. The dynamic response of such a system can be presented in two ways :- a) as a function of time, or b) as a function of frequency.

These descriptions are termed respectively the "time-domain" response and the "frequency-domain" response of the system. The two domains are related by the Laplace integral transform, and in a more restricted sense by the Fourier integral transform. For many engineering applications the Laplace transform is too general, and the Fourier transform is used. Since the time and frequency domains are complementary the choice of which system response description to use is largely subjective.

The writer's main interest is in control engineering, and in this field both time-domain and frequency-domain analysis techniques are highly developed. However, for linear, time-invariant control systems, most practical design techniques are formulated in the frequency-domain. Therefore, for practical purposes, presenting system information to the control engineer in terms of frequency is the natural way.

Over the last twenty years random signal theory has found wide applications in control engineering. The main reasons for such developments are :-

a) When system disturbances are random, and therefore cannot be described exactly, feedback control is used. To determine the type of control which is best, statistical descriptions of the signals are used.

b) The high cost of taking expensive control plant off-line has meant that systems must be studied 'in-situ'. Therefore, measurements must be made in the presence of the normal plant operating signals. To differentiate between the system response to the test signal and the operating signal, statistical methods must be used.

The frequency-domain study of a system subjected to random disturbances involves determining the frequency characteristics (or spectra) of the disturbances, and the frequency response of the system. In the linear, time-invariant case a knowledge of the frequency response is sufficient to specify the system. This process of specifying a system is usually termed 'identification', and its purpose is to enable the control engineer to assess and improve the performance of the control system.

Thus it is seen that there is a definite need for experimental procedures to analyse the spectral properties of random signals, and to identify the frequency response of systems in the presence of random disturbances.

1.2. Conventional Digital Spectral Analysis Procedures

We now confine our discussion to digital spectral analysis techniques. Until recently the role of digital methods in spectral analysis has been rather limited. This was because the digital version of the Fourier transform, the discrete Fourier transform, takes a long time to calculate. The discrete Fourier transform of a time series x_i,$(i = 0,1, \ldots\ldots N-1)$ is defined as :-

$$A_k = \frac{1}{N} \sum_{i=0}^{N-1} x_i W_N^{ik} \qquad k = 0,1, \ldots\ldots N-1$$

where $W_N = \exp(-j\frac{2\pi}{N})$

The reason why it takes a long time to calculate the discrete Fourier transform can be seen if the transform is considered as a complex matrix multiplication of an N x N matrix and an N vector. In a computer, multiplication is a lengthy operation, so the speed with which the discrete Fourier transform can be calculated is limited by the time taken to compute N^2 complex multiplications. Even for small values of N this can take a long time. (For example, in a typical computer it might take up to twenty minutes to transform a 500 point complex series.) Clearly then if the engineer can manipulate and present digital data adequately in the discrete time-domain there is no advantage in making a time consuming transformation into the discrete frequency-domain.

In the field of random signal analysis the same difficulty existed, but with the added inconvenience that the discrete estimate of the autospectrum of a random signal is not a consistent estimator. A consistent estimator is one whose variance and bias tend to zero as the number of observations is increased. Unfortunately, the variance of discrete autospectrum estimators are independent of the number of observations.

A discrete autospectrum estimate of a zero mean signal $x(t)$ can be obtained from the discrete Fourier transform as :-

$$C_{xx}(\omega_k) = N\Delta\left\{|A_k|\right\}^2 ; \qquad k = 0,1, \ldots\ldots \frac{N}{2}$$

where; $\omega_k = \frac{2\pi k}{\Delta N}$

Δ = the sampling rate

A_k = the discrete Fourier transform of an N point time series, obtained by sampling $x(t)$ at Δ second intervals

This discrete estimate of the autospectrum consists of a set of estimates of points on the true autospectrum. The true autospectrum $\Gamma_{xx}(\omega)$ of a stationary process $X(t)$ is defined as the Fourier transform of the autocovariance function, $\gamma_{xx}(u)$.

$$\Gamma_{xx}(\omega) = \int_{-\infty}^{+\infty} \gamma_{xx}(u) e^{-j\omega u} du$$

$$\gamma_{xx}(u) = \mathrm{Cov}\left\{X(t), X(t+u)\right\}$$

Where $\mathrm{Cov}\{\ ,\ \}$ is the covariance operator.

The discrete autospectrum estimate is often termed the 'periodogram', and for brevity we will follow this practice. The periodogram is inconsistent because the components of the periodogram are statistically independent, and since the periodogram based upon N observations will have $\frac{N}{2}$ components, each component is effectively based upon only two observations of the random signal. The periodogram thus has a variability which is independent of the number of observations.

The process of obtaining consistent autospectral estimates is termed periodogram smoothing. This entails further lengthly discrete Fourier transforms and provides another argument against the use of digital spectral techniques.

The forms of periodogram smoothing originally used to obtain consistent estimates of the spectrum both used the same premise: By averaging n independent estimates of the same quantity, a reduction in variance of n^{-1} is achieved.

a) Averaging over frequency

This technique was due to J.P. Daniell (ref. 1.1.), and uses the fact that the components of a periodogram are independent. If n neighbouring periodogram elements are averaged, and the true spectrum is 'smooth' over the averaged frequency band, then the result is an improved estimate of the spectrum in that frequency band. To improve the variance further more periodogram elements can be used. The new estimate represents an average over a wider

frequency range, so that the variability of the estimated spectrum is reduced by $\frac{1}{n}$, and the bandwidth of the estimators is increased by n.

b) Averaging over time

This method is usually associated with M.S. Bartlett (ref. 1.1.). The technique used is to average n independent periodograms obtained by observing the same signal. The resultant smoothed periodogram has a variance of $\frac{1}{n}^{th}$ of the raw periodograms. The method is applied to a single record in the following way :- An N point set of observations is sub-divided into n sub-sets of $\frac{N}{n}$ points each. A smoothed periodogram is then obtained by averaging the periodograms of the sub-sets. In a similar way to method a), this reduces the variance by a factor of $\frac{1}{n}$ at the expense of a bandwidth increase proportional to n.

Both smoothing methods reduce the variability of the digital spectral estimates by increasing the bandwidth of the frequency filter through which the true spectrum is observed. In this sense both methods involve spectral estimates which are averages over frequency. Except for differences in the 'shape' of the frequency filter, both methods give similar results. By proper design of spectral estimation experiments the differences caused by these filters can be reduced to negligible proportions. Therefore, the choice between methods a) and b) is based upon the amount of computation involved in each method.

In method a) an N point periodogram is computed, so at least N^2 multiplications are required. However, in method b) there are n periodograms calculated. Each periodogram involves $\frac{N}{n}$ points, so at least N^2/n multiplications are required. Because of this computational advantage method b) is the most popular traditional approach to spectral analysis.

In texts on digital spectral analysis, smoothing by averaging over time rarely appears explicitly. This is because a different computational procedure, which gives the same results, but which is not obviously equivalent has become established as the best way to estimate smoothed spectra. This computational procedure is known as the 'Lagged-Products' or 'Indirect' route; It, became the accepted approach to spectral estimation largely through the book "The Measurement of Power Spectra" by Blackman and Tukey (ref. 1.2.), which was for many years the only book which treated digital spectral analysis from an engineering stand-point.

The 'Lagged-Products' route uses the following procedure:

1) Estimate the autocovariance function,

2) Multiply the estimated autocovariance function by a function $l(t)$. This function is often called a 'window function', and a class of windows which are in wide use satisfy the following general conditions :-

a) $l(0) = 1$

b) $l(t) = l(-t)$

c) $l(t) = 0 \; ; \quad |t| > \frac{N}{n}$

By varying the shape of the window function the user can control the shape of the frequency filter through which the true spectrum is viewed.

3) Finally the smoothed periodogram is calculated by taking the discrete Fourier transform of the 'windowed' estimate of the autocovariance function.

The 'Lagged-Product' route is held to be better than direct use of method b) because an estimate of the autocovariance function is obtained 'en-route'. Method b) also provides autocovariance estimates, but these are based upon records of duration $\frac{N}{n}$, whereas the 'Lagged-Product' estimate is based upon the entire, N point, record.

1.3. The Effect of the FFT on Spectral Analysis Procedures

Until the mid-1960's the 'Lagged-Products' route remained entrenched computationally as the most efficient way of estimating spectra. In 1965 however, the 'Fast-Fourier-Transform' (FFT) algorithm was formulated, and the computational advantage was removed from the 'Lagged-Products' route.

In their paper "An Algorithm for the Machine Calculation of Complex Fourier Series", Cooley and Tukey (ref. 1.3.) showed that by clever programming it is possible to execute the discrete Fourier transform of an N point complex vector (where $N = r^m$) using about N.m complex multiplications instead of N^2. So for example the discrete Fourier transform of a 2048 point block can be transformed in 38 seconds instead of 376 minutes as before. The family of algorithms which enabled this to be done became known collectively as the Fast Fourier Transform.

The advent of the FFT has meant that the transformation between time and frequency domains is no longer the key factor in calculating computation time. Indeed the FFT enables the systems analyst to transform back and forth between the domains with no great computational penalty. This means that data may be handled in whichever domain is the more convenient (ref. 1.4.).

In spectral analysis of random signals, the 'Lagged-Products' method is no longer obviously the more efficient method, and the choice between methods a) and b) is open. Both these

methods are now in common use (ref. 1.5., 1.6.).

The speed advantage of the FFT has also lead to the formulation of spectral analysis methods that would not have been feasible using the direct discrete Fourier transform. Most notable among these is the method of complex demodulates (ref. 1.6.). This technique is really a discrete version of the analogue, heterodyne, spectrum analyzer. Using digital demodulation, the frequency of interest is translated to zero frequency. The demodulated data is then low-pass filtered, this may be done using the FFT method described in reference 1.4. Then by squaring and averaging the filtered series, an estimate of the power near the demodulated frequency is obtained.

To summarise, the FFT has lead to a rebirth of interest in the direct methods of spectral analysis, and has made possible more sophisticated spectral analysis routines. In the general field of data handling the FFT has lead to a greater flexibility in data manipulation.

1.4. Motivation and Content of the Thesis

The work contained in this thesis was initiated by an idea due to Dr. M.T.G. Hughes. The suggestion was, that if the FFT could be efficiently programmed on an 'on-line' digitial computer, it would be possible to estimate spectra in 'real-time'.

In this context the term 'on-line' means that the computer is collecting data directly, and not through some intermediate medium, such as punched paper tape. The term 'real-time' is used in the sense that the computer collects and processes data at a rate determined by data analysis requirements, and not by the limitations of the computer. Therefore, an 'on-line' digital computer analyzing physical data in 'real-time' reads analogue data directly via an analogue/digital interface, and analyses it as it becomes available at the interface. When analyzing random data, some form of averaging of estimates is often required, so that an on-line computer could display current estimates, and update the estimates as more data becomes available. Thus, a properly programmed on-line digital computer would function like, and form a useful alternative to an analogue spectrum analyzer.

Such a computer would be superior to analogue machines for the following reasons :-

a) A programmed computer is more flexible than a special purpose machine, so that 'real-time' digital spectral analysis procedures may be easily tailored to suit specific application. Furthermore, the computer is not committed solely to spectral

analysis, and can be used for other purposes when not required for spectral measurements.

b) Permanent recordings of the analysis results may be produced in a variety of ways through the computer peripherals. If required, addition computations may be done on the results before outputting.

The first phase of the project work was a study of the FFT applied to on-line computers, in particular the programming in assembly code of the fixed-point arithmetic FFT. This aspect of the thesis is described in chapters 2 and 3. Chapter 2 gives a description of the FFT algorithms, and chapter 3 presents the results of the study of the fixed-point, assembly coded, FFT.

The assembly code FFT was then used as the basis of a 'real-time' digital autospectrum estimation programme for an on-line digital computer. A description of this programme, and the statistical properties of the estimator used, is given in chapter 4.

The success of the 'real-time' autospectrum estimation routines led to the development of a programme to estimate (in real-time) frequency response functions from noise-like data. A description of this programme and the properties of the estimators involved is given in chapter 5.

In chapter 6 the problem of estimating the frequency response of transfer functions associated with closed loop systems

is considered. In particular the statistical properties of a frequency response estimator of the forward path transfer function are discussed.

Chapter 7 presents some results of experiments conducted using the 'real-time' autospectrum and frequency response estimation routines. The thesis is concluded with a brief assessment of the real-time spectral analysis method, and a proposal for further work.

1.5. Contribution of the Thesis

This thesis makes five contributions to the field of digital spectral analysis, and frequency response estimation from random data. These contributions are listed below :

a) The first is a practical one, and is the development of 'real-time' digital autospectrum estimation procedures. This involves an application of the 'averaging over time' smoothing method described in section 1.2. At its conception this application was thought to be novel.* However, since then several commercial instruments have appeared on the market, and these use the same technique.

As a direct extension of the auto-spectrum estimation procedures, 'real-time' digital frequency response estimation techniques were developed. To the author's knowledge no published work exists which describes frequency response estimation using these techniques.

b) An r.m.s. round-off error analysis for the fixed-point FFT is developed in chapter 3. This error analysis is an extension of the previous analysis (ref. 1.7.) for sign/magnitude binary arithmetic to the more common two's-complement binary arithmetic.

c) A new expression for the covariance of sample cross-spectral estimators is given in Appendix 2. This expression is more general than existing expressions (for example Jenkins, ref. 1.8.) since it assumes a general window function.

d) In chapter 6 a study is made of the estimation of the frequency response function of the forward path transfer function of a closed loop system with noise in the loop. Approximate

* This differs from the contribution of ref 1-5 in that the author estimates spectra 'on-line' and in 'real-time'. Ref. 15 makes no such suggestion.

variance and bias expressions for the estimators of gain and phase are derived, and confidence intervals are given for the special case of noise-free feedback. To the authors best knowledge these results have not been presented before.

e) In chapter 5 the problem of aliasing in the discrete spectral analysis of noise-like data is approached. An expression for the covariance of aliased sample cross-spectral estimators is given (Appendix 3), and expressions for the bias and variance of aliased frequency response estimates are developed. It is believed that these expressions are original.

REFERENCES

1.1. Bartlett, M.S., "An Introduction to Stochastic Processes", pp. 304 - 331, C.U.P., 1966.

1.2. Blackman, R.B. and Tukey, J.W., "The Measurement of Power Spectra", Dover Press, 1959.

1.3. Cooley, W.J. and Tukey, J.W., "An Algorithm for the Machine Calculation of Complex Fourier Series", Math. Comp., vol. 19, pp. 297 - 301, April 1965.

1.4. Helms, H.D., "Fast Fourier Transform of Method of Computing Difference Equations and Simulating Filters", IEEE. Trans on Audio and Electroacoustics , vol. AU-15, pp. 85 - 90, June 1967.

1.5. Welch, P.D., "The Use of Fast Fourier Transforms for the Estimation of Power Spectra : A Method Based on Timing Averaging Over Short, Modified Periodograms", IEEE. Trans on Audio and Electro-acoustics, vol. AU-15, pp. 70 - 73, June 1967.

1.6. Bingham, C. et al, "Modern Techniques of Power Spectrum Estimation", IEEE Trans. on Audio and Electroacoustics, vol. AU-15, pp. 56 - 66, June 1967.

1.7. Welch, P.D., "A Fixed-Point Fast Fourier Transform Error Analysis", IEEE. Trans. on Audio and Electro-acoustics, vol. AU-17, pp. 151 - 157, June 1969.

1.8. Jenkins, G.M. and Watts, D.G., "Spectral Analysis and its Applications", pp. 414, Holden Day, 1968.

2. THE FAST FOURIER TRANSFORM

2.1. Introduction

The FFT is a method which computes the discrete Fourier transform (DFT) of a complex, N point, time series more rapidly than other algorithms available at the moment. The algorithm obtains its speed advantage from the fact that, if N has many factors, then the number of operations required to execute a DFT can be very much reduced. To be specific, if $N = R^m$ the FFT takes a number of operations, ('operation' meaning a complex multiplication) proportional to N.m to compute the DFT. By comparison, the direct approach takes a number of operations proportional to N^2.

The FFT has the further advantage that it requires less computer store than the direct method, and it computes the DFT with reduced round-off errors.

In this chapter the basic FFT algorithms for radix 2 operation ($N = 2^m$) are described, and their extension to radix R, and mixed radix operation is explained. The use of the FFT to process real data is discussed and the round-off errors are commented upon. Finally reversible "Algol" procedures for the fast-Fourier-transformation of complex, and real, time series are presented.

List of Symbols Used in Chapter 2

A_r	r^{th} component of the discrete Fourier transform of X_k
B_r	r^{th} " " " " " " " " Y_k
B_k^e	k^{th} " " " " " " " " L_k
B_k^o	k^{th} " " " " " " " " P_k
C_r	r^{th} " " " " " " " " Z_k
C_k^e	k^{th} " " " " " " " " V_k
C_k^o	k^{th} " " " " " " " " Q_k
D_r	r^{th} " " " " " " " " a real time series
L_k	k^{th} component of a sub-series of Y_k
m	index used in an FFT
N	number of points in an FFT array
P_k	k^{th} component of a sub-series of Y_k
Q_k	k^{th} " " " " " " Z_k
R	radix used in an FFT
R_r	r^{th} component of a sub-series of A_k
S_r	r^{th} " " " " " " A_k
V_k	k^{th} " " " " " " Z_k
W_N	$\exp(-j\frac{2\pi}{N})$
X_k	k^{th} component of a complex N point time series
Y_k	k^{th} " " " " $\frac{N}{2}$ point sub-series of X_k
Z_k	k^{th} " " " " $\frac{N}{2}$ point sub-series of X_k

2.2. The Fast-Fourier-Transform Algorithm

In this section a brief description of the two principal FFT algorithms is given. These algorithms are known as the decimation-in-time and decimation-in-frequency FFT's. For simplicity the radix 2 forms of the algorithms are given, so it is assumed initially that $N = 2^m$, (where m is an integer).

2.2.1. Decimation-in-time

This is the original form of the FFT used by Cooley & Tukey (ref. 2.1.). It is termed the decimation-in-time FFT because it operates on the time series, decomposing it into successive sets of sub-series, and operates on the sub-series to form the discrete Fourier coefficients.

Suppose we have a complex, N point, time series X_k, and that the DFT of X_k is A_r. Let Y_k and Z_k be sub-series of X_k defined by :-

$$Y_k = X_{2k}$$

$$k = 0,1, \ldots, \frac{N}{2} - 1 \qquad \ldots\ldots\ldots 2.1.$$

$$Z_k = X_{2k+1}$$

Now, let Y_k and Z_k have DFT's B_r and C_r, defined thus :-

$$B_r = \frac{2}{N} \sum_{k=0}^{\frac{N}{2}-1} Y_k \, W_{N/2}^{kr}$$

$$r = 0,1 \ldots\ldots \frac{N}{2} - 1 \quad \ldots\ldots\ldots \quad 2.2.$$

$$C_r = \frac{2}{N} \sum_{k=0}^{\frac{N}{2}-1} Z_k \, W_{N/2}^{kr}$$

Where $W_N = \exp\left(- j\frac{2\pi}{N} \right)$

The DFT of X_k can be written in terms of the sub-series as follows :-

$$A_r = \frac{2}{N} \sum_{k=0}^{\frac{N}{2}-1} \left\{ Y_k \, W_{N/2}^{kr} + Z_k \, W_{N/2}^{kr} \right\} W_N^r$$

$$= \frac{1}{2} \left\{ B_r + W_N^r \, C_r \right\} \quad r = 0,1, \ldots\ldots \frac{N}{2} - 1 \quad \ldots\ldots\ldots \quad 2.3.$$

For $N/2 \leqslant r < N - 1$ it is recalled that B_r and C_r are periodic in $N/2$ so that we can show that :-

$$A_{r + \frac{N}{2}} = \frac{1}{2} \left\{ B_{r + \frac{N}{2}} + W_N^{r + \frac{N}{2}} \, C_{r + \frac{N}{2}} \right\}$$

$$A_{r + \frac{N}{2}} = \frac{1}{2} \left\{ B_r - W_N^r \, C_r \right\} \quad r = 0,1, \ldots\ldots \frac{N}{2} - 1 \quad \ldots\ldots\ldots \quad 2.4.$$

Equations 2.3., and 2.4., are the decimation-in-time algorithm generating equations. They show that the DFT of an N point series can be found by forming a linear combination of the DFT's of the sub-series Y_k and Z_k.

The speed advantage of the FFT can now be illustrated. Assuming 2 times $(N/2)^2$ operations (complex multiplies) were taken to form B_r and C_r. Then only $\frac{N^2}{2} + \frac{N}{2}$ operations are needed to form A_r using equations 2.3., and 2.4., while N^2 operations are needed when direct summation is used.

The full 'decimation-in-time' algorithm is generated from equations 2.3., and 2.4., in the following way. The sub-series Y_k, and Z_k are split to further sub-series P_k, L_k and Q_k, V_k, respectively. These new series are defined by :-

$$\left.\begin{aligned} P_k &= Y_{2k} \\ L_k &= Y_{2k+1} \\ Q_k &= Z_{2k} \\ V_k &= Z_{2k+1} \end{aligned}\right\} k = 0,1 \ldots\ldots \frac{N}{4} - 1$$

Using the equations 2.3., and 2.4., the DFT's B_r, C_r can be expressed in terms of B^e, B^o, C^e, and C^o, the DFT's of the sub-series P_k, L_k, Q_k, and V_k. This process of sub-division can be continued until there are only two elements in each sub-series. The DFT of the complete series can then be found using the DFT of these two element sub-series.

As an example a signal-flow chart of the decimation-in-time algorithm applied to an eight point series is given in fig. 2.1. (This figure is based upon illustrations used in ref. 2.2.). By reference to this flow-chart two more features of the FFT may be shown. Firstly, the data is initially ordered in binary bit-reversed order. (By rearranging the flow-chart the strict Cooley/Tukey algorithm can be obtained, with correctly ordered input data and bit-reversed output data.) The other feature of the FFT which can be obtained from the flow-chart is the 'in-place' computation property.

To illustrate this point consider the in-place calculation of the eight point transform in figure 2.1. The computational sequence is :-

a) Compute B_0^e and B_1^e, and overwrite X_0 and X_4 with the newly computed values.

Compute B_0^o and B_1^o and overwrite X_2 and X_6, and so-on, until all the first stage is calculated.

b) Having completed all the operations in the first stage, we commence calculating the second stage.

Calculate B_0 and B_2, and overwrite B_0^e and B_0^o.

Calculate B_1 and B_3, and overwrite B_1^e and B_1^o and so-on .

c) Finally, having computed the second stage, calculate the final array A_k and overwrite the B's and C's.

Thus the FFT enables the calculation of the DFT with no additional store area being required. The bit-reverse ordering of the data can also be done 'in-place'.

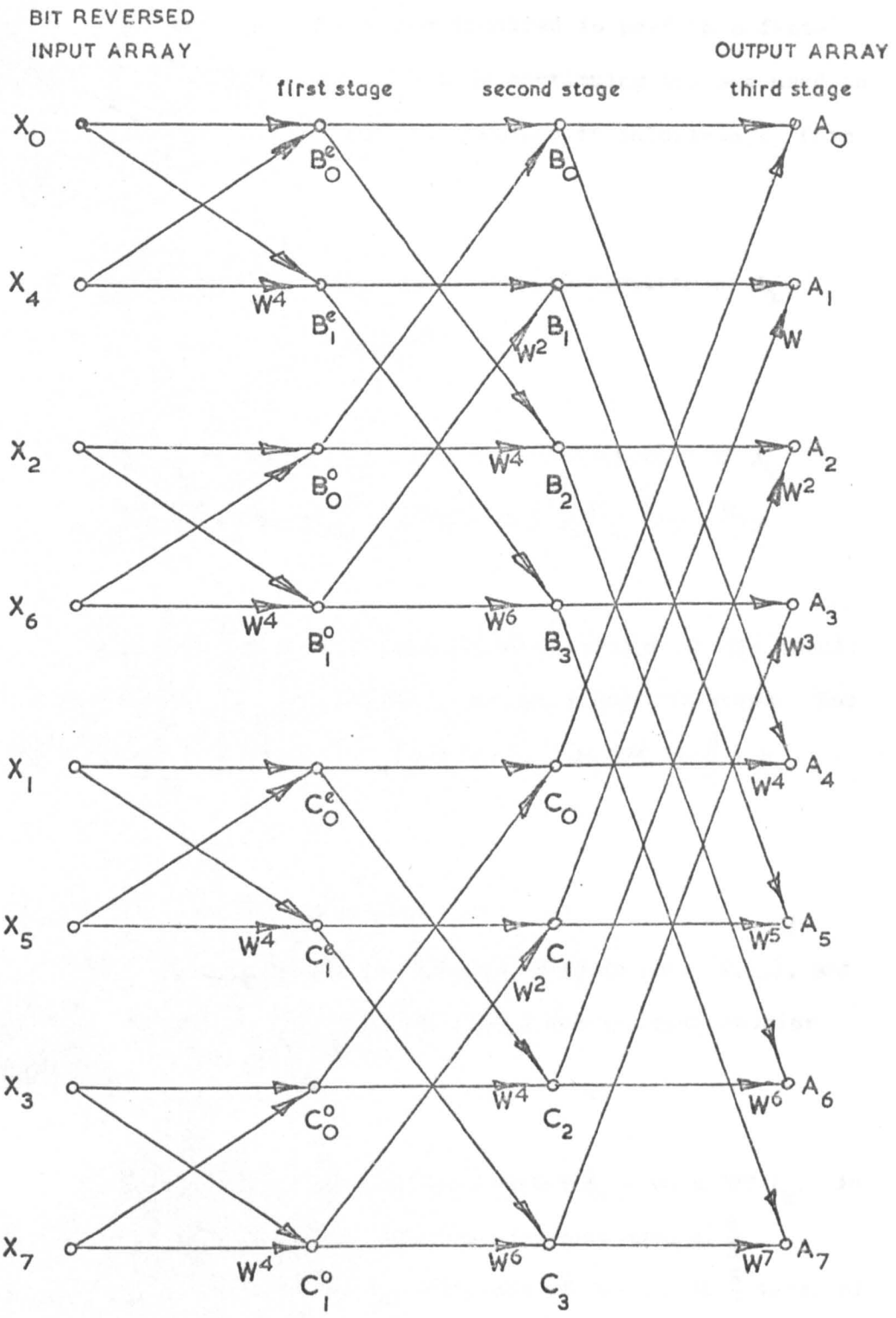


FIGURE 2.1. SIGNAL FLOW CHART FOR AN EIGHT POINT DECIMATION-IN-TIME FFT ALGORITHM.

The number of operations required to perform a fast-Fourier-transformation can be found by continuing the sum used to determine the number of operations required to calculate A_r from B_r and C_r :-

Number of operations using one sub-division of A_k

$$2(N/2)^2 + N/2$$

Number of operations using two sub-division of A_k

$$2(\ 2(N/4)^2 + N/4\ ) + N/2 = 4(N/4)^2 + N/2 + N/2$$

This sum may be extended, until we arrive at the result for a full FFT, which (for $N = 2^m$) involves m sub-divisions. For m sub-division the number of operations by the FFT is $\frac{N}{2} \cdot m$

2.2.2. Decimation-in-Frequency

This form of the FFT was devised by Sande (ref. 2.3.), and as its name suggests operates by subdividing the discrete Fourier coefficients.

Let the complex, N point, time series X_k have a DFT A_k. As before the time series is split into two sequences each of $\frac{N}{2}$ terms. The first sub-series, Y_k, consists of the first $\frac{N}{2}$ terms of X_k, and the other sub-series, Z_k consists of the last $\frac{N}{2}$ terms of X_k.

$$Y_k = X_k$$

$$k = 0,1 \ldots\ldots \frac{N}{2} - 1$$

$$Z_k = X_{k + \frac{N}{2}}$$

The DFT of X_k may be written in terms of the Y_k and Z_k's, thus :-

$$A_r = \frac{1}{N} \sum_{k=0}^{\frac{N}{2} - 1} \left\{ Y_k + Z_k W_N^{\frac{Nr}{2}} \right\} W_N^{kr} \qquad \ldots\ldots\ldots\ldots 2.5.$$

The discrete Fourier coefficients A_k are split into two sub-series. The first, R_r consists of even numbered points and the other, S_r, consists of odd numbered points.

$$R_r = A_{2r}$$

$$r = 0,1, \ldots\ldots \frac{N}{2} - 1$$

$$S_r = A_{2r + 1}$$

Substituting R_r in equation 2.5., gives :-

$$R_r = \frac{1}{N} \sum_{k=0}^{\frac{N}{2} - 1} (Y_k + Z_k) \; W_{N/2}^{rk} \qquad \ldots\ldots\ldots\ldots 2.6.$$

That is, R_r is the mean of the DFT's of Y_k and Z_k.

Similarly for S_r :-

$$S_r = \frac{1}{N} \sum_{k=0}^{\frac{N}{2}-1} (Y_k - Z_k)\, W_{N/2}^{rk}\, W_N^k \qquad \ldots\ldots\ldots\ldots 2.7.$$

Thus S_r is the DFT of the function $(Y_k - Z_k)\, W_N^k$.

Equations 2.6., and 2.7., are the decimation-in-frequency algorithms generating equations. Using these expressions the full decimation-in-frequency FFT can be derived in a similar way to that in which the decimation-in-time FFT was developed in the previous section.

In addition, all the properties of the FFT discussed with reference to decimation-in-time apply to decimation-in-frequency.

2.2.3. Radix R Algorithms

In the previous sections the FFT algorithms for N of the form 2^m were described. Similar algorithms for N of the form R^m (R integer) can be written. If N is not composite of this form, then algorithms to handle $N = \prod_{i=1}^{m} p_i$ can be written, where the p_i are prime. Such FFT's are termed 'mixed-radix' algorithms, and, because of the complex indexing they require, they are usually avoided in favour of the simpler single radix algorithms.

Cooley and Tukey (ref. 2.1.) have shown that the most

efficient FFT has radix 3. However, the binary radices (2,4,8,16,) have other advantages. These are, (i) binary indexing may be used. This is useful in binary arithmetic machines and; (ii) use may be made of the symmetry in the discrete Fourier transforms of binary radix point blocks to further reduce the number of operations involved in the FFT. The larger the binary radix used, the fewer operations are required. Working with large binary radices has two disadvantages. Firstly, the programme instruction store area increases in proportion to the radix, and secondly the usable values of N become widely separated. The latter disadvantage may be overcome by using a mixed binary radix algorithm. For instance, a 4,2 mixed radix transform can transform data blocks of the form 2^m. When $m = 2k$ a pure 4 radix FFT is used, and when $m = 2k + 1$ a radix 4 FFT is used for the first 2k stages and the transform completed with a 2 radix final stage.

2.3. Processing Real Time Series

The FFT algorithms described here can be used to transform (or inverse transform) complex data blocks. In many data processing applications, however, we require the DFT of real time series. The FFT can be used directly to transform such series, by forming the complex series $Z_k = X_k + j0$. This is wasteful of computer time and store space, since by using the conjugate symmetry $A_k = A^*_{N-k}$ of the DFT's of real time series we can write routines to transform two real time series of N points each using one complex N point transform. With a little extra effort it is possible to transform a single real N point series in one complex $N/2$ point FFT.

2.3.1. Transforming Two Real Time Series

If we have two series X_k and Y_k of the same length, then the auxiliary complex time series Z_k is formed. Thus :-

$$Z_k = X_k + j\,Y_k \; ; \; k = 0,1, \ldots\ldots N-1$$

We now show that the DFT's of X_k and Y_k may be obtained from the DFT of Z_k. Let C_r, A_r, and B_r be the DFT's of Z_k, X_k and Y_k respectively.

Then :-

$$C_r = \frac{1}{N} \sum_{k=0}^{N-1} Z_k W^{rk} = \frac{1}{N} \sum_{k=0}^{N-1} (X_k + j\,Y_k) W^{rk} \qquad \ldots\ldots\ldots\ldots 2.8.$$

$$C^*_{N-r} = \frac{1}{N} \sum_{k=0}^{N-1} (X_k - j\, Y_k) W^{rk} \qquad \ldots\ldots\ldots\ldots 2.9.$$

Solving equations 2.8., and 2.9., for A_r and B_r gives

$$B_r = \frac{j}{2} (C^*_{N-r} - C_r)$$

$$r = 0,1, \ldots\ldots \frac{N}{2} \qquad \ldots\ldots\ldots\ldots 2.10.$$

$$A_r = \frac{1}{2} (C^*_{N-r} + C_r)$$

Hence, the N point DFT's of X_k and Y_k can be generated using a single complex N point FFT.

2.3.2. Transforming a Single Real Time Series

A more useful technique than that described in 2.3.1. is now given. This enables us to transform a single real series, X_k, using a complex $\frac{N}{2}$ point FFT.

Defining the auxiliary complex series Z_k :-

$$Z_k = X_{2k} + j\, X_{2k+1}; \; k = 0,1, \ldots\ldots \frac{N}{2} - 1$$

Now, using an $\frac{N}{2}$ point FFT, the DFT of Z_k is obtained

$$A_r = \frac{2}{N} \sum_{k=0}^{\frac{N}{2}-1} (X_{2k} + j\, X_{2k+1})\, W^{kr}_{N/2} \qquad \ldots\ldots\ldots\ldots 2.11.$$

Defining B_r and C_r as the $\frac{N}{2}$ point DFT's of X_{2k} and X_{2k+1} respectively, then from 2.11. and using the result of section 2.3.1.

$$B_r = \frac{1}{2}\left(A_r + A^*_{\frac{N}{2}-r}\right)$$

$$C_r = -\frac{j}{2}\left(A_r - A^*_{\frac{N}{2}-r}\right) \qquad \ldots\ldots\ldots\ldots \text{2.12.}$$

Now refering to section 2.2.1., we can use the algorithm generating equation 2.3., to show that D_r, the DFT of X_k, is given by :-

$$D_r = \frac{1}{2}\left\{B_r + W_N^r C_r\right\} \qquad \ldots\ldots\ldots\ldots \text{2.13.}$$

Note that because of the conjugate symmetry of D_k, all the useful information is contained in the first half of the D_k.

2.4. A Comment on Round-off Errors in the FFT

The FFT has the important advantage that the round-off errors incurred are smaller than those incurred by direct summation. The reduction in errors has often been erroneously coupled with the reduced number of operations required by the FFT. The FFT involves less round-off error because of the way in which the discrete Fourier coefficients are calculated. If a discrete Fourier coefficient is expressed in terms of X_k and W_N using the FFT it may be seen that the FFT calculates each coefficient by a series of nested multiplications.

An analogy may be made between the calculation of the DFT and the evaluation of a polynominal. In this analogy the direct method of calculating a complex Fourier coefficient is compared with the direct evaluation of a polynominal, and the FFT method of calculating a Fourier coefficient is compared with the nested multiplication method of calculation of a polynominal. This analogy may be pursued to obtain an error bound for the FFT using the results of Wilkinson (ref. 2.4.) obtained for the nested product evaluation of a polynominal. However this is not the purpose of the comparison, which is merely to point out that the FFT gains its numerical efficiency by calculating the discrete Fourier coefficients by nested multiplications. The round-off error in the FFT is studied more thoroughly in Chapter 3. In this study the errors incurred by using fixed-point arithmetic are considered. For an error analysis when floating-point arithmetic is used the reader is referred to reference 2.5.

2.5. Computer Programmes for the FFT

"Algol" procedures to execute the FFT have been prepared by the author, and since they are of general interest, listings of them are given in appendix 1. The first two procedures are reversible FFT algorithms operating on the complex arrays REA, IMA of size $N = R^m$. One procedure uses the decimation-in-time algorithm, the other the decimation-in-frequency algorithm.

The remaining procedure is a reversible algorithm for transforming (or inverse-transforming) a single real time series, and is used with either FFT procedure.

The timing of the FFT procedures for various N is compared with the direct method in TABLE 2.1. These figures are for the transformation of complex time series. When ODDEVEN is used to transform N real data points the timing for the N point complex FFT's should be halved.

Execution Time on Elliot 4130

TABLE 2.1.

N	FFT dec-in-time (seconds)	FFT dec-in-freq (seconds)	Direct Method (minutes)
128	1	1	1.45
256	3	4	5.85
512	7	8	22.5
1024	17	17	94.3
2048	38	39	376*
4096	84	85	1505*
8192	184	184	6004*

* These figures were calculated by extrapolating the times for N = 128, 256 and 512, and assuming the execution time is proportional to the square of N.

REFERENCES

2.1. Cooley, J.W., and Tukey, J.W., "An Algorithm for Machine Calculation of Complex Fourier Series" Math. Comp., vol. 19, pp. 297 - 301, April 1965.

2.2. Cochran, W.T., "What is the Fast Fourier Transform?", IEEE.Trans., Vol. AU - 15, No. 2, pp. 45 -55, 1967.

2.3. Gentleman, W.M., and Sande, G., "Fast Fourier Transforms for Fun and Profit", 1966 Fall Joint Comp. Conf., AFIPS Proc., vol. 29, pp. 563 - 578, 1966.

2.4. Wilkinson, J.H., "Rounding Errors in Algebraic Processes", pp. 34 - 38, H.M.S.O., 1963.

2.5. Weinstein, C.J., "Round-off Noise in Floating Point Fast Fourier Transform Computations", IEEE.Trans., Vol. AY - 17, No. 3, pp. 209 - 215, 1969.

3. AN ASSEMBLY CODE STUDY OF THE FFT

3.1. Introduction

The purpose of this project is to investigate real-time spectral analysis techniques, using an on-line digital computer. If it is intended to analyze the signals in real-time we must be able to execute the discrete Fourier transform quickly and accurately. As was shown in the previous chapter, the FFT enables us to do this, and so forms a suitable basis for a real-time spectral analysis system. However, further reductions in computation time can be achieved by coding the FFT in an assembly language, and using fixed-point arithmetic. Therefore, as the key part of our spectral analysis facility we aim at evolving an assembly code FFT programme using fixed-point arithmetic.

In this chapter the results of a micro-programming study of the fixed-point FFT are presented. Section 3.2. deals with the selection of the algorithm type and radix, and outlines the programme structure. In section 3.3. the accuracy and speed of the fixed-point FFT are discussed. It is shown that true fixed-point is inaccurate, and suggests array-scaling as an alternative technique. Some ways of implementing the array-scaled FFT are given, and their speed and accuracy compared.

The assembly code used in this study was the SYMBOL language developed by Scientific Data Systems. All assembly code programmes were implemented using a General Electric Company 90/2 digital computer.

LIST OF SYMBOLS

$A(k)$	the k^{th} component of the discrete Fourier transform of $X_o(v)$.
$\overline{Im(W)}$	the mean value of the imaginary part of the trigonometric coefficients W^k.
$\overline{Im^2(X_p)}$	the mean square value of the imaginary part of the complex array X_p.
$\overline{Im(X_p)}$	the mean value of the imaginary part of the complex array X_p.
K	the mean square modulus of the complex array X_o.
m	the index of the number of points Fourier transformed in a radix 2 FFT.
N	the number of points in a radix 2 FFT.
p	the number of the stage reached in an FFT computation.
$\overline{Re(W)}$	the mean value of the real part of the trigonometric coefficients W^k.
$\overline{Re^2(X_p)}$	the mean square value of the real part of the complex array X_p.
$\overline{Re(X_p)}$	the mean value of the real part of the complex array X_p.
t	the number of binary bits in a computer word.
W	the complex coefficient $\exp(-j\frac{2\pi}{N})$.
$X_m(v)$	the v^{th} component of the complex FFT array after completion of the FFT.
$X_o(v)$	the v^{th} component of the complex FFT array before transformation.

$X_p(v)$ the v^{th} component of the complex FFT array after p stages of an FFT.

ϵ_p the error in the real and imaginary components of the complex array X_p.

ϵ_W the error in the real and imaginary components of the coefficients W^k.

σ_1^2 variance of the error caused by rounding a number to a t bit, two's-complement, binary number.

σ_2^2 variance of the error caused by discarding the least significant bit of a two's-complement binary number.

μ_1 mean value of the error caused by rounding a number to a t bit, two's-complement, binary number.

μ_2 mean value of the error caused by discarding the least significant bit of a t bit, two's-complement, binary number.

3.2. General Description of the FFT assembly code programme

In planning an FFT programme the first decision which must be made is whether to use the decimation-in-time algorithm, or the decimation-in-frequency algorithm. Both algorithms use the same arithmetic structure, the difference between them being in the indexing procedures. The algorithm selection must therefore be based upon the suitability of the indexing procedures to a binary indexing machine. The difference between the two algorithms may be summarised as follows :-

The decimation-in-time FFT requires relatively little indexing, but the indices are altered in bit-reversed fashion. Conversely, the decimation-in-frequency algorithm requires much more indexing, but the indices are altered in a fairly orthodox fashion. The obvious choice for a binary indexing machine is therefore a binary radix decimation-in-time FFT.

In selecting the decimation-in-time FFT we have also restricted the choice of radix to a binary number. If the usable values of N are to be sensibly spaced, we must also stipulate that N take the form 2^m (m, integral). This restricts our choice to either a radix 2 or a mixed, 4 plus 2 radix algorithm. To aid the decision between these two radices both were programmed in assembly code and their performances compared. The 4 plus 2 radix algorithm was found to be slightly the faster (10%), but required about twice as much code space as the radix 2 algorithm. Since the mixed radix algorithm was not dramatically faster, the more compact radix 2 algorithm was selected.

Having justified choosing a radix 2 decimation-in-time FFT to form the basis of the real-time spectral analysis facility, the structure of the assembly code programme for this algorithm is given. For the purposes of this discussion the programme can be conveniently split into four blocks; These blocks are described below :-

1) RESERVED STORE AREAS

These are sections of the computer memory reserved for exclusive use of the FFT algorithm. The store areas are, a) scratch index and work space; and b) scratch parameter block; c) arrays for 'look-up' tables of trigonometrical coefficients and bit-reversed integers; and d) an N point complex array.

Areas a) and b) are sections of the computer scratch memory pad, which is an area of store which can be accessed more quickly than other store areas. Areas c) and d) are arrays associated with the FFT. These arrays vary in size with N, so they are situated at the top of the store and are expanded downward when necessary.

2) SET-UP ROUTINES

When a new value of N is required these routines calculate the new FFT parameters and compile a suitable bit-reversed integer table.

3) FFT ALGORITHM

Operating in the scratch work space this section calculates

the discrete Fourier transform of the contents of the N point complex array.

4) BIT-REVERSED SEQUENCE

After the discrete Fourier transformation this block uses the bit-reversed integer array to rearrange the discrete Fourier coefficients in the correct order.

The programme is coded in such a way that it may be used as a separately loaded subroutine, or as a part of a main programme. The set store areas are in the main programme store reservations. If the FFT programme is loaded as a separate routine, the addresses of the indices and arrays are copied into the FFT at load time. This avoids wasteful indirect indexing between main and sub-programmes.

When the FFT sub-routine is first entered the required value of 'm' must be in the machine 'A' register. The sub-routine compares this new 'm' with the existing m in the parameter block, and if necessary enters the set-up routines and alters the FFT parameters accordingly. The machine then exits from the FFT routine. To forward transform the contents of the complex array the FFT sub-routine is entered with the preset value of m. To inverse transform the contents of the complex array the FFT routine is entered with the negative of the preset value of m.

The maximum usable value of m is set by the length of the trigonometrical table. This table is fed into store at load time

from a paper tape prepared on the University Elliot 4130 computer. Currently used tapes allow a maximum value of N = 256, but this figure can be increased. The ultimate limit on N is set by the size of the computer store.

3.3. Numerical Procedures in the Fixed-Point FFT

The radix 2 FFT operates upon a complex, N point array, $X_o(v), v = 0,1, \ldots\ldots N - 1.$, where N is an integer power of two. In a computer the complex array is represented by two real N - point arrays. Each real array member consists of one computer word, and it is assumed that the binary point is situated to the left of the most significant bit (excluding the sign bit). Therefore, the range of values which can be stored in one word is -1 to +1, and the contents of each complex array point is bounded by the unit square enclosed by the lines 1 + j1, 1 - j1, -1 -j1, and -1 + j1.

The FFT of a 2^m point complex array is executed in 'm' stages. For example, the 8 point FFT described in chapter 2 is executed in three stages, these stages are indicated in figure 2.1. The contents of the complex array at a particular stage are identified by an appropriate suffix. For instance, at the n^{th} stage the array is designated $X_n(v)$. Each stage of the FFT consists of an arithmetic loop and some indexing routines, and it is the purpose of this section to describe the arithmetic loop.

In the loop the complex arithmetic expression in the equation set 3.1. is applied (with suitable changes in indices, v, q, k) to the array. The indices v, q, k are altered in a complex fashion after each application of the set 3.1. Since the purpose of this section is to explain only the numerical properties of the F.F.T loop, the indexing procedures used to alter v, q, k are not described. The procedures of the indexing procedures are discussed in the previous chapter + also in ref. 2.1.

$$X_p(v) = X_{p-1}(v) + X_{p-1}(q)\, W^k$$

$$X_p(q) = X_{p-1}(v) - X_{p-1}(q)\, W^k \qquad \text{..... 3.1.}$$

Where $W = \exp(-j\frac{2\pi}{N})$

Equation set 3.1. is a typical computation pair in the p^{th} loop of the FFT. In this loop the p^{th} complex array is obtained by applying equation set 3.1. to the $p\text{-}1^{th}$ array. The equation set is applied $\frac{N}{2}$ times, so that all N array members are processed. After m stages, the discrete Fourier coefficients, $X_m(v)$ are obtained.

Moving from stage to stage, the modulus of the array members, in a mean square sense, doubles. At each stage therefore there is a possibility of overflow in the newly computed array. If an overflow does occur, then it must be rectified by rescaling the array. Rescaling is accomplished by shifting the contents of the array store words to the right until the overflow is cleared. Each shift to the right is equivalent to dividing by two.

There are two ways of avoiding overflow in the arithmetic loop. These are :-

1) Rescale the array before <u>every</u> loop. This ensures that overflow will not occur, and at the end of the m^{th} loop we have computed :-

$$2^{-m} \sum_{v=0}^{N-1} X_o(v) W^{rk}$$

Which is the correctly scaled discrete Fourier transform of $X_o(v)$. Since the same array scale factor is maintained throughout the transform, this rescaling technique is true fixed-point arithmetic. In general, fixed-point arithmetic is undesirable since overflow will not occur at every stage, and some of the rescalings will be redundant, causing unnecessary inaccuracy.

2) An alternative to true fixed-point is array-scaled arithmetic. Array-scaling is a compromise between the speed and inaccuracy of fixed-point arithmetic, and the slow efficiency of floating-point arithmetic. It is a form of scaling used extensively in computer calculations with matrices (ref. 3.1.). The technique treats all the matrix elements as mantissi of a single floating-point number, and stores the matrix characteristic in a single computer word. If any of the mantissi overflow the entire matrix is rescaled. When applied to the FFT, array scaling operates in the following way :-

To decide whether an overflow will occur in a loop a test is applied to the array members. If the test proves positive, and an overflow can occur, then all the array members are rescaled by the same amount. At the end of the transform we have the complex array $X_m(v)$, which has been rescaled, say l times. The discrete Fourier transform of $X_o(v)$ is then given by :-

$$A(k) = 2^{l-m} X_m(k)$$

3.3.1. Implementing Array Scaling in the FFT

There are several ways in which the FFT can be programmed using array scaling. The different methods arise initially from the different tests which can be applied to the array members to decide whether rescaling of the array is necessary. Secondary differences arise over the choice of when in a loop to rescale the array. There are two principal rescaling criteria. These are, the Modulus test and the Overflow test.

a) The Modulus Test

This technique uses a knowledge of the structure of the arithmetic expressions executed in the FFT loop. By testing the $p-1^{th}$ array, and using a knowledge of equation set 3.1., the possibility of an overflow in the p^{th} loop can be predicted. Consider either of the expressions in equation set 3.1. If the moduli of the $p-1^{th}$ array are controlled to be less than $\frac{1}{2}$, then from 3.1. it can be seen that the moduli of the new array will be less than one, and there will be no overflows in the p^{th} loop.

Once the necessity for a rescaling of the array has been established, there is a choice of when to rescale the data. The alternatives are :-

1) Rescale the array before entering the loop to compute the next array.

2) Enter the loop, and rescale the array members immediately after the new array members are calculated.

The first method is the simpler of the two, because the loop can be entered and the new array calculated with no allowance for overflows. The second method was first suggested by Shively (ref. 3.2.). It has the advantage that rescaling is delayed from the p-1th array until the pth array, thus reducing errors. Its disadvantage lies in the added programming involved in allowing for an overflow in the computation of equation set 3.1.

b) The Overflow Test

This rescaling criterion assumes no knowledge of the equations to be calculated. The loop is entered, and allowing for overflow, the new array is calculated. If an overflow occurs, the offending number is rescaled. The entire array, consisting of some new array members and some old, is then rescaled by the same factor, and computation continued. The rescaling need not be done in any specific order, so the rescaling routine should be no more complex than that required in the Modulus test (type 1). The Overflow test requires more programming than either of the Modulus test transforms since allowance must be made for two overflows in the array calculations.

3.3.2. R.M.S. Error Bound for the Array-Scaled FFT

To compare the three array scaled FFT's, their r.m.s. round-off error bounds were derived. The r.m.s. error criterion was used because it is more useful than the absolute error in assessing the accuracy of random signal measurements. An r.m.s.

error bound will therefore be useful later when we are calculating the measurement noise in the spectral analysis programmes. The error analysis presented here is substantially the same as that used by Welsh (ref. 3.3.). Modification of Welsh's analysis was necessary because his theory was developed for sign/magnitude arithmetic, and the machine used in this project used two's-complement arithmetic.

Errors in Two's-Complement Arithmetic

The operations which cause round-off error in the array scaled FFT are 1) rounding of multiple precision numbers to single precision; and 2) rescaling of single precision numbers before an operation which might otherwise involve an overflow.

1) Rounding

When, in a t bit word machine, a double precision word is rounded to single precision, the least significant word is tested. If it is greater than or equal to $2^{-(t+1)}$, then 2^{-t} is added to the most significant word. If the least significant word is less than $2^{-(t+1)}$, nothing is done. Following Hamming (ref. 3.4.) the binary sequence of errors caused by this rounding may be approximated by a random variable uniformly distributed in the range $-2^{-(t+1)}$ to $+2^{-(t+1)}$. The mean and the variance of the rounding error can therefore be calculated as :-

$$\mu_1 = 0 \qquad \ldots\ldots\ldots 3.2.$$

$$\delta_1^2 = \frac{2^{-2t}}{12}$$

2) Rescaling

When a t bit number is rescaled, it is cycled to the right one place, and the least significant bit discarded. The binary sequence of errors caused by rescaling can be represented by a binary random variable. This random variable may take the values 0 and 2^{-t} with equal probability. Accordingly, the mean and variance of the rescaling error can be found as :-

$$\mu_2 = 2^{-(t+1)}$$
$$\sigma_2^2 = 2^{-2(t+1)} \qquad \ldots\ldots\ldots\ 3.3.$$

Error Analysis

In this section the upper bound on the r.m.s. round-off error for a typical array-scaled radix 2 FFT is calculated. The array scaling technique considered uses the Modulus rescaling criterion, with scaling before the loop. In the analysis it is assumed that the sum computed by the FFT is :-

$$X_m(k) = \sum_{v=0}^{N-1} X_o(v) W^{kv}$$

The correctly scaled discrete Fourier transform is obtained at the end of the transform by dividing by N.

By computing the FFT in this way the magnitude of the members of the complex array will increase at each stage. To accommodate such increases it will be assumed that at each rescaling

the binary point is shifted one place to the right. The magnitude of all the numbers in the complex array is therefore doubled after each rescaling. This applies to rounding and rescaling errors as well, so that after p rescalings the error caused by rounding a double precision number will have zero mean and variance $(4^p)\,\sigma_1^2$. The error caused by the $p+1^{th}$ rescaling of a number will have a mean value $2^p\mu_2$ and a variance of $4^p\sigma_2^2$.

The following result is used in the analysis. This relates the mean square modulus of the complex array at the p^{th} stage with the mean square modulus of the complex array at the $p-1^{th}$ stage.

$$E\left\{ |X_p|^2 \right\} = 2\,E\left\{ |X_{p-1}|^2 \right\}$$

This result is due to Welsh (ref. 3.3.), and it may be obtained by taking the expectation of the squared modulus of equation 3.1. Using this result we can say that if K is the mean square modulus of the input array, then the mean square modulus of the output array is $2^m K$.

To obtain an upper r.m.s. error bound it is assumed that a rescaling is found necessary prior to every loop. It is also assumed that the rounding errors are uncorrelated with each other, and the array members.

Consider a typical real computation at the p^{th} stage of the FFT. From equations 3.1. such a computation is :-

$$Re\left\{X_p(v)\right\} = Re\left\{X_{p-1}(v)\right\} + Re\left\{X_{p-1}(q)\right\} Im(W^k) + Im\left\{X_{p-1}(q)\right\} Re(W^k)$$

To a first order approximation the error in this typical computation can be expressed as a linear function of the errors in the array members and trigonometric coefficients. Using such an approximation the mean square error at the p^{th} stage can be obtained by adding the variances of the error terms, and the square of the expected values of the error terms. This process can be extended to the entire FFT computation by separately calculating the variance and the mean error in the FFT, and combining them to obtain the mean square round-off error.

In the error analysis that follows the error in the real and imaginary components of the p^{th} array, X_p, is denoted ϵ_p. The variance of the error at the p^{th} stage is indicated by $Var(\epsilon_p)$, and the mean error is denoted by $\bar{\epsilon}_p$. The error in the trigonometric coefficients, $Im(W^k)$ and $Re(W^k)$ is denoted ϵ_w. This error is caused by representing the coefficients by a t bit binary number, and has zero mean and a variance $\frac{2^{-2t}}{12}$.

Variance of the Error

The initial array will require rescaling to avoid overflow in the first loop. So, denoting the error in the rescaled initial array by ϵ_o, the variance of the error is :-

$$Var\left\{\epsilon_o\right\} = \sigma_2^2$$

Where σ_2^2 is the round-off error caused by the initial rescaling

The first loop

In this loop the value of W^k in the FFT computation (equations 3.1.) is either ± 1. Therefore the computations reduce to a set of additions and subtractions.

A typical computation is :-

$$\mathrm{Re}\left\{ X_1(v) \right\} = \mathrm{Re}\left\{ X_o(v) \right\} + \mathrm{Re}\left\{ X_o(q) \right\}$$

From which the error in the array after the first loop is :-

$$\mathrm{Var}\left\{ \epsilon_1 \right\} = 2\ \mathrm{Var}\left\{ \epsilon_o \right\} = 2\ \sigma_2^2$$

The second loop

In this loop the value taken by W^k is ± 1 or $\pm j$, so again the FFT computations reduce to a set of additions and subtractions.

$$\mathrm{Var}\left\{ \epsilon_2 \right\} = 2\ \mathrm{Var}\left\{ \epsilon_1 \right\} + 2(4\ \sigma_2^2)$$

The first term in the above equation is the variance transmitted through the FFT calculation. The second term is the variance due to rescaling the data prior to the second loop. The factor of 4 in the rescaling variance is present because the binary point was shifted one bit to the right by the first rescaling.

The third loop

In this loop half the computations are additions and subtractions, and half are proper FFT computations. We estimate the variance of the error in each half of the computations, and

average the results, to obtain an overall error variance for the third loop. The error in the first half of the third loop will be denoted ϵ'_3 and the error in the second half as ϵ''_3.

The first half may be treated in the same way as loops one and two.

$$\text{Var}(\epsilon'_3) = 2\,\text{Var}(\epsilon_2) + 2(4^2\sigma_2^2)$$

A typical computation in the second half is :-

$$\text{Re}\{X_3(v)\} = \text{Re}\{X_2(v)\} + \text{Re}\{X_2(q)\}\,\text{Im}\{W^k\} + \text{Im}\{X_2(q)\}\,\text{Re}\{W^k\} \quad \ldots\ldots\ldots 3.4.$$

The variance of the error in such a computation is given by :-

$$\begin{aligned}\text{Var}\{\epsilon''_3(v)\} \triangleq\ & \text{Var}\{\epsilon_2\} + \text{Var}\{\epsilon_2(\text{Im}\,W^k + \text{Re}\,W^k)\} \\ & + \text{Var}\{\epsilon_w(\text{Re}\,X_2(q) + \text{Im}\,X_2(q))\} \\ & + \text{Variance due to rounding and scaling}\end{aligned} \quad \ldots\ldots\ldots 3.5.$$

Where ϵ_w is the error in the stored values of the trigonometric coefficients. Taking the mean over all of the calculations in the second half of the third loop.

$$\begin{aligned}\text{Var}\{\epsilon''_3\} &\triangleq \text{Var}\{\epsilon_2\} + \text{Var}\{\epsilon_2\}\left[\text{Re}^2\{W^k\} + \text{Im}^2\{W^k\}\right] \\ &\quad + \text{Var}\{\epsilon_w\}\left[\overline{\text{Re}^2\{X_2\}} + \overline{\text{Im}^2\{X_2\}}\right] \\ &\quad + 4^2\sigma_2^2 + 4^3\sigma_1^2 \\ &\triangleq 2\,\text{Var}\{\epsilon_2\} + 2^2K\sigma_1^2 + 4^2\sigma_2^2 + 4^3\sigma_1^2 \quad \ldots\ldots 3.6.\end{aligned}$$

The overall error variance at the end of the third loop is :-

$$\mathrm{Var}\{\epsilon_3\} = 2\,\mathrm{Var}\{\epsilon_2\} + 2K\sigma_1^2 + 4^2\sigma_2^2 + 2.4\sigma_2^2 + 2.4^2\sigma_1^2$$

..... 3.7.

In the fourth loop a quarter of the operations are additions and subtractions, and in subsequent loops fewer still are additions and subtractions. Therefore, to ease the task at hand, it is assumed that in the fourth and subsequent loops all operations are normal FFT computations. By extending the variance figure for the second half of loop three, the variance of the error at the end of the fourth loop is found to be :-

$$\mathrm{Var}\{\epsilon_4\} = 2\,\mathrm{Var}\{\epsilon_3\} + 2^3 K\sigma_1^2 + 4^4\sigma_1^2 + 4^3\sigma_2^2 \quad \ldots\ldots 3.8.$$

The variance at the end of the m^{th} loop can be obtained by extending equation 3.8., and substituting for $\mathrm{Var}\{\epsilon_{m-1}\}$ etc.

This gives the approximate variance expression :—

$$\mathrm{Var}\{\epsilon_m\} \simeq (m-2\tfrac{1}{2})2^{m-1}K\sigma_1^2 + 2^{2m+1}\sigma_1^2 + 2^{2m-1}\sigma_2^2 \quad \ldots\ldots 3.9.$$

Mean of the Error

Denoting the mean error in the p^{th} array by $\bar{\epsilon}_p$, the mean error in the rescaled initial array is :-

$$\bar{\epsilon}_o = \mu_2$$

Where μ_2 is the mean error caused by rescaling.

The first loop consists of a set of additions and sub-

tractions, there being an equal number of each. To find the mean error at the end of the first loop we estimate the mean error in the additions and the mean error in the subtractions, and average the two.

The mean error in computations involving additions is $2\mu_2$, and the mean error in the subtractions is zero. Therefore :-

$$\bar{\epsilon}_1 = \mu_2$$

The second loop can be handled in the same way as loop one. The mean error at the end of the second loop is :-

$$\bar{\epsilon}_2 = \bar{\epsilon}_1 + 2\mu_2$$

In the above equation the first term is the error carried through from the first loop, and the second term is the error introduced by rescaling. The factor of 2 is present because the binary point was shifted by the first rescaling.

Half of the third loop calculations consists of additions and subtractions, the other half consists of normal FFT computations. As before, the mean error in each set of calculations is calculated, and the two values averaged. The mean error in the first half of the calculations is denoted by $\bar{\epsilon}_3'$, and in the second half by $\bar{\epsilon}_3''$.

The mean error in the first half may be found using the methods of loops one and two :-

$$\bar{\epsilon}_3' = \bar{\epsilon}_2 + 2^2\mu_2$$

A typical FFT computation in the second half of the third loop is given in equation 3.4. The mean error for such a computation is given by :-

$$\bar{\epsilon}_3'' = \bar{\epsilon}_2 + 2^2\mu_2 + \bar{\epsilon}_2\left\{\overline{(\mathrm{Im}(W^k)} + \overline{\mathrm{Re}(W^k)}\right\} + \bar{\epsilon}_w\left\{\overline{\mathrm{Re}(X_2)} + \overline{\mathrm{Im}(X_2)}\right\}$$

where $\bar{\epsilon}_w = \mu_1 = 0$, and $\overline{\mathrm{Im}(W^k)} = \overline{\mathrm{Re}(W^k)} = 0$

Therefore the mean error at the end of the third loop is given by :-

$$\bar{\epsilon}_3 = \bar{\epsilon}_2 + 2^2\mu_2$$

The fourth loop and subsequent loops may be treated in the same way, and so on. At the end of the m^{th} loop the mean error is given by :-

$$\bar{\epsilon}_m = \bar{\epsilon}_{m-1} + 2^{m-1}\mu_2 \xrightarrow{(m\to\infty)} 2^m\mu_2 \qquad \ldots\ldots\ldots 3.10.$$

Combining the variance and the mean error, the mean square error at the end of the transform is :-

$$\text{m.s.e.} \doteq (m-2\tfrac{1}{2})2^{m-1}K\,\sigma_1^2 + 2^{2m+1}\sigma_1^2 + 2^{2m-1}\sigma_2^2 + 2^{2m}\mu_2^2 \qquad \ldots\ldots 3.11.$$

The method used by Welsh (ref. 3.3.) can now be used to calculate the ratio :-

$$\frac{\text{mean square error}}{\text{mean square value of the input}} = \frac{\text{m.s.e.}}{\text{m.s.i.}}$$

The mean square modulus of the output array is 2^mK, where K is the mean square modulus of the input array. The mean square value of the input data is $\frac{K}{2}$ and the mean square value of the output data is $2^m.\frac{K}{2}$. Therefore :-

$$\frac{\text{m.s.e.}}{\text{m.s.o.}} \simeq \frac{(m-2\tfrac{1}{2})2^{m-1}K\sigma_1^2+2^{2m+1}\sigma_1^2+2^{2m-1}\sigma_2^2+2^{2m}\mu_2^2}{2^m.\ K/2}$$

Substituting for σ_1^2, σ_2^2 and μ_2, neglecting small terms, and square rooting we obtain an approximate expression for the r.m.s. error due to round-off errors.

$$\frac{\text{r.m.s. error}}{\text{r.m.s.o.}} \simeq \frac{0.74\ 2^{m/2}\ 2^{-t}}{\text{r.m.s.i.}} \text{ (for m large)} \qquad \ldots\ldots\ldots \text{ 3.11.}$$

For a computer using a B bit word, B-1 bits are used to represent the magnitude of a two's-complement number. Therefore, t = B-1 in the above equation.

The r.m.s. error bound given by equation 3.11. assumes that a rescaling of the data is necessary at each stage. In practice it will not be necessary to rescale before every loop unless a strong periodic component is present in the initial data. On the other hand it may be necessary to rescale the data <u>twice</u> before the first loop. This happens whenever the magnitude rescaling criterion is used. If the initial array is correctly scaled it is possible that the maximum initial array modulus is greater than unity. If this is so, then two rescalings are needed to control the modulus to be less than $\frac{1}{2}$.

The approximate r.m.s. error bound in this case is :-

$$\frac{\text{r.m.s.e.}}{\text{r.m.s.o.}} = \frac{2^{\frac{m+2}{2}}\ 2^{-t}\ 0.74}{\text{r.m.s.i.}}$$

The r.m.s. error bounds for the other rescaling criteria are :-

a) The Overflow test

$$\frac{\text{r.m.s.e.}}{\text{r.m.s.o.}} \leq \frac{2^{\frac{m+2}{2}}\, 2^{-t}\, 0.41}{\text{r.m.s.i.}}$$

b) The Modulus test with rescaling in the FFT loop

$$\frac{\text{r.m.s.e.}}{\text{r.m.s.o.}} \leq \frac{2^{\frac{m+2}{2}}\, 2^{-t}\, 0.41}{\text{r.m.s.i.}}$$

3.3.3. Experimental Work

a) Verification of RMS Error Bounds

The theoretical error bounds for the array-scaled FFT were checked by comparing them with the actual rms errors. The actual errors in the FFT were estimated by transforming a signal using array-scaling and comparing the result with the same signal transformed in floating-point. The rms error was then found by assuming the floating-point programme was exact, and calculating the r.m.s. difference between the fixed-point and floating-point transforms. To obtain a consistent estimate of the error each transform was repeated several times with different samples of the test signal. The test signal used was a random number generator generating numbers in the range $-\frac{1}{2}$ to $+\frac{1}{2}$. To ensure that a rescaling would be necessary at each stage of the transform, a strong mean level was added to the signal. The actual and

theoretical r.m.s. errors for various values of m are compared in graphs 3.1., 3.2., and 3.3.

The predicted error bounds all slightly under estimate the maximum r.m.s. errors. Possible reasons for this are :-

a) The approximations made in obtaining the theoretical error bound. In particular, the errors caused by the rounding in the trigonometrical coefficients were neglected.

b) Differences between the theoretical variances and means and the actual variance and means in the rescaling and rounding operations. For instance, Weinstein (ref. 3.5.) found that experimentally determined values of variance due to rounding-off typical FFT sums differed significantly from theoretical values.

A further difference between the actual and theoretical r.m.s. errors was that the experimental error increased with m at a greater rate than that predicted by the theoretical error bound. This behaviour has been commented on by Weinstein (ref. 3.5.). He concluded that correlation between the signal and round-off noise could cause this to happen.

b) Timing of the Transforms

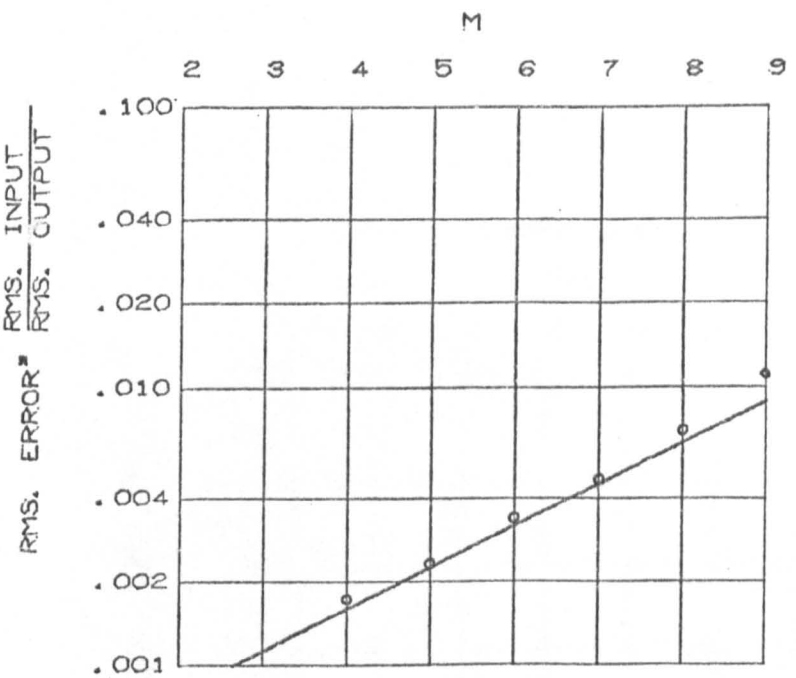
The transform times for the various array scaling techniques were compared by measuring the time taken by each programme to forward Fourier transform a 256 point complex data block. The data transformed was such that all the programmes rescaled the array four times in the course of a transformation.

GRAPHS TO COMPARE THE THEORETICAL AND EXPERIMENTAL UPPER BOUNDS ON RMS ERROR IN THE ARRAY SCALED FFT.

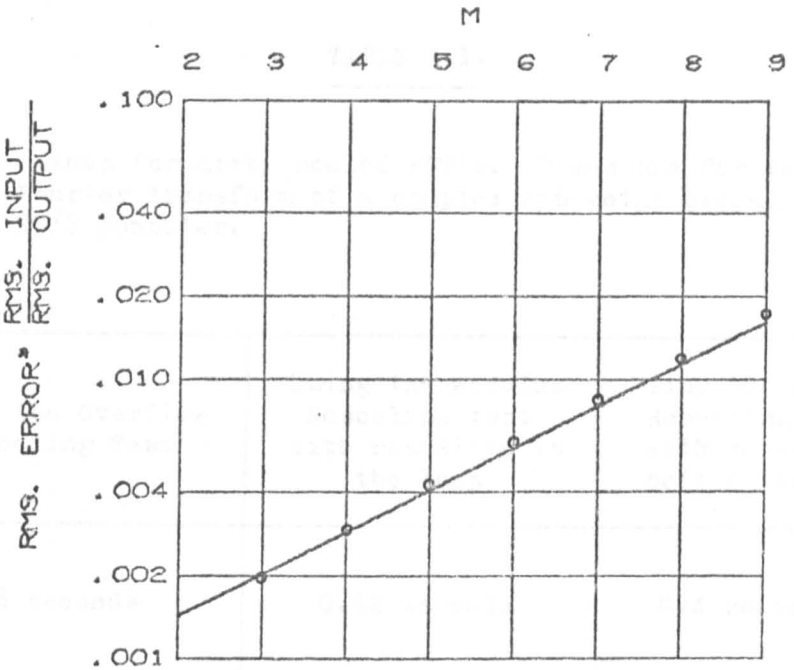
EXPERIMENTAL RESULTS WERE OBTAINED FROM A 12 BIT WORD MACHINE USING TWOS-COMPLEMENT ARITHMETIC
GRAPH 3.1. SHOWS RESULTS OBTAINED USING THE OVERFLOW TEST
GRAPH 3.2. SHOWS RESULTS OBTAINED USING THE MODULUS TEST WITH RESCALING BEFORE THE LOOP
GRAPH 3.3. SHOWS RESULTS OBTAINED USING THE MODULUS TEST WITH RESCALING IN THE LOOP

THE CONTINUOUS LINE SHOWS THE UPPER THEORETICAL RMS ERROR BOUND. THE CIRCLES MARK EXPERIMENTAL POINTS

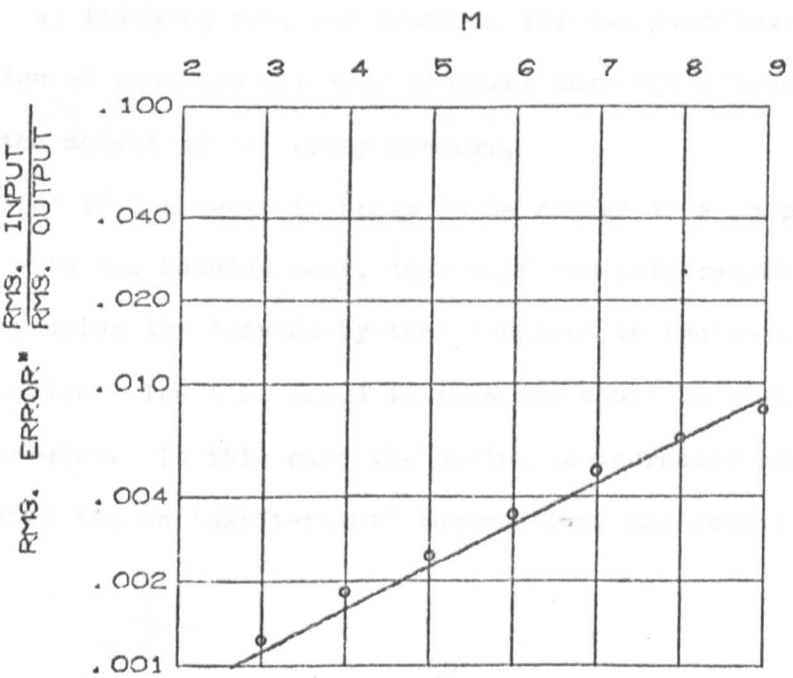
GRAPH 3.1.



GRAPH 3.2.



GRAPH 3.3.



The times taken by the programmes are tabulated in TABLE 3.1.

TABLE 3.1.

Transform times for array scaled FFT's. Times are for the forward discrete Fourier transform of a complex 256 point block, computed on a GEC 90/2 computer.

Using the Overflow Rescaling Test	Using the Modulus Rescaling test with rescaling in the loop	Using the Modulus Rescaling test with rescaling before the loop
0.78 seconds	0.52 seconds	0.4 seconds

From the table it is seen that the overflow rescale test transform is substantially slower than either of the Modulus test transforms. This is because :-

a) Allowing for, and testing, for two overflows in the computation of equation set 3.1. requires more operations than testing the moduli of the array members,

and; b) If a rescale is known to be needed in a loop, as is the case with the Modulus test, then half the data rescalings can be done by using the 'divide-by-two' inherent in two's-complement multiplication. The time saved in this way would be significant in any computer. In this case the saving is increased because the GEC 90/2 has no 'shift-right' instruction, and rescaling must

be done by a long shift left.

The timing experiment was a little unfair on the Overflow test, because it was arranged that each transform rescaled the data the same number of times. In practice it is possible that the Modulus test will cause more rescalings than the Overflow test. The extra rescaling is necessary to control the magnitude of the initial array such that $|X_0(v)| < 1$. However, even allowing for an extra rescaling in the modulus test criterion, the overflow rescaling test is still significantly slower than the modulus test.

3.3.4. Conclusions

The micro-programming of the fixed-point FFT has been discussed, and the choice of a radix 2 decimation-in-time algorithm for a binary arithmetic computer justified.

A comparison of various ways of implementing array scaling in the fixed-point FFT has also been given. The comparison was made by programming the FFT in SYMBOL assembly code using various scaling techniques, and measuring the r.m.s. error in each transform, and the time taken to execute a transformation.

The transforms using the Modulus rescaling criterion were easily the faster. This was because they were easier to programme and could take advantage of the rescaling which is inherent in two's-complement multiplication. The fastest transform

used the Modulus rescaling test with rescaling before the FFT loop. This transform was however also the least accurate.

The Modulus test (with rescaling in the loop) and the Overflow test gave rise to the same upper r.m.s. error bound. However, when transforming data which does not require rescaling at every stage, the overflow test transform will be more accurate. This is because the modulus test transform rescales the data if there is a possibility of an overflow. It is possible therefore that the modulus test might rescale the data unnecessarily. In spite of this disadvantage the Modulus test (with rescaling in the loop) provides the most efficient method in terms of speed and accuracy of implementing array scaling in the FFT.

3.4. Additional Comments

3.4.1. Processing Real Data

The complex fixed-point FFT can be used to transform real data by using the algorithm described in section 2.3. This algorithm shows that with an extra set of additions and subtractions the discrete Fourier transform of two, equal length, real time series can be obtained from a single complex FFT. The accuracy and speed of this technique are now considered.

a) Errors

No additional errors are introduced by the extra addition/subtraction required in the transforming of real time series. This is because the coefficient of $\frac{1}{2}$ in equation 2.10. can be omitted without risk of an overflow by combining the addition/subtraction routine with other subsequent routines. Therefore, the upper rms error bounds in section 3.3.2. apply directly to the fixed-point discrete Fourier transform of two real N point series.

b) Timing

Compared with the time required to execute the FFT, the time taken to execute equations 2.10. is negligible. Therefore, the times given in Table 3.1. are also the times taken to FFT two real time series.

3.4.2. Error Bounds for Data Input from the Analogue/ Digital Converter

The analogue-to-digital converter fitted to the GEC 90/2 computer used in this project is wired so that full scale at the analogue input is equivalent to 0.5 in the computer. Because of this, data input from the analogue to digital converter requires (at most) m-1 rescalings in a 2^m point FFT. Therefore, the r.m.s. error bounds in section 3.3.2. should be divided by $(2)^{\frac{1}{2}}$ when applied to data input via the analogue-to-digital converter.

REFERENCES

3.1. Wilkinson, J.H., "Rounding Errors in Algebraic Processes", pp. 26 - 28, H.M.S.O., 1963.

3.2. Shively, R.R., "A Digital Processor to Generate Spectra in Real Time", 1st. Ann. IEEE Comp. Conf., Digest of Papers, pp. 21 - 24, 1967.

3.3. Welch, P.D., "A Fixed-Point Fast Fourier Transform Error Analysis", IEEE Trans. on Audio and Electroacoustics, Vol. AU-17 No. 2, pp. 151 - 157, 1969.

3.4. Hamming, R.W., "Numerical Methods for Scientists and Engineers", pp. 24 - 40, McGraw-Hill, 1962.

3.5. Weinstein, C.J., "Round-Off Noise in Floating Point Fast Fourier Transform Computation", IEEE. Trans. on Audio and Electroacoustics, Vol. AU-17, No. 3, pp. 209 - 215, 1969.

4. AUTOSPECTRUM ESTIMATION

4.1. Introduction

This chapter deals with the properties of a real-time digital autospectrum estimation technique. Section 4.2. develops the statistical properties of the autospectrum estimator used by the technique, and in section 4.3., a computer routine for the implementation of the digital autospectrum estimation technique is described. Finally, in section 4.4., the properties of the computer routine are discussed.

List of Symbols for Chapter 4

a	expectation of a chi-square variate
a_n, n^{th}	component of a power series expansion of $\frac{1}{q}\left\{ l(t)*l(t)\right\}$
A_k, k^{th}	component of the discrete Fourier transform of $x_i l_i$
$Cxx(\omega)$	the sample autospectrum of $x(t)$
$\overline{C}xx(\omega)$	smoothed estimate of the autospectrum of $x(t)$
$\overline{C}xx(\omega_k)$	smoothed, discrete autospectrum estimate
$\overline{C}xx_A(\omega)$	Aliased smoothed autospectrum estimate
$f(\omega)$	power transfer function of a low-pass filter
$l(t)$	a data window function
l_i	a discrete data window function
$l_H(t)$	the Generalised Hanning data window
$l_R(t)$	the rectangular data window
$L_H(\omega)$	the Fourier transform of $l_H(t)$
$L_R(\omega)$	the Fourier transform of $l_R(t)$
N	the number of points in a time series
q	the integral over time of $l^2(t)$ (discrete or continuous)
t	time
T	duration of a data sample
$U_H(\omega)$	a part of the function $L_R(\omega)$
$x(t)$	a signal
x_i	a time series obtained by sampling $x(t)$
$\overline{x}$	the sample mean of $x(t)$
$X(\omega)$	the Fourier transform of $x(t)l(t)$
$\delta xx(t)$	the auto-covariance of $x(t)$
$\Gamma xx(\omega)$	the autospectrum of $x(t)$
$\triangle$	the sampling period

μ the mean value of $x(t)$

ν the degrees of freedom of a chi-square variate

ω angular frequency

ω_B half power bandwidth of $L_H(\omega)$

ω_c cutoff frequency of a low-pass filter

ω_k the discrete frequency $\frac{2\pi k}{\Delta N}$

ω_p the break-point of the function $U_H(\omega)$

ω_s the sampling frequency

u. a time variable

v. angular frequency.

4.2. Properties of the Autospectral Estimator

The on-line digital computer programmes for autospectrum analysis calculate a smoothed periodogram of a signal by averaging the periodograms of consecutive segments of a data record. The smoothed periodogram so obtained provides a discrete estimate of the signal autospectrum. In this section the statistical properties of this type of discrete autospectrum estimator are discussed. This is done by deriving the properties of an equivalent continuous autospectrum estimator. The properties of the discrete estimator are then obtained by analogy.

4.2.1. Definition of Autospectrum

The autospectrum $\Gamma_{xx}(\omega)$ of a stationary, zero mean, random process $x(t)$ is defined as the Fourier transform of the autocovariance function $\gamma_{xx}(t)$.

Where :-

$$\gamma_{xx}(t) = \lim_{T\to\infty}\left\{\frac{1}{2T}\int_{-T}^{+T} x(u)\,x(u-t)\,du\right\}$$

and :-

$$\Gamma_{xx}(\omega) = \int_{-\infty}^{+\infty} \gamma_{xx}(t)e^{-j\omega t}\,dt$$

4.2.2. Definition of the Sample Autospectrum and the Periodogram

In the on-line digital computer programme for autospectral analysis the discrete Fourier transform of each segment of data is

found using the FFT. Its periodogram is then obtained by calculating the squared modulus of all the Fourier coefficients. In this section the periodogram of a data segment is defined and is related to the sample spectrum which is the continuous equivalent of the periodogram.

a) The Sample Autospectrum

Estimators of the autospectrum of a process x(t) are often based upon the defining equations of the autospectrum, and are therefore expressed as the Fourier transform of the estimated autocovariance. An alternative way of formulating an autospectrum estimator is to express it in terms of the squared modulus of the Fourier transform of the process. When the FFT is used, this direct approach is the more efficient way of computing an autospectrum estimate (ref. 4.1.). For this reason the direct method is used in the real-time autospectrum estimation programmes.

If we have a sample of a zero-mean, gaussian, random process x(t), for $-\frac{T}{2} < t < +\frac{T}{2}$, then the sample autospectrum computed using the direct approach is defined as :-

$$C_{xx}(\omega) = \frac{1}{q} \left| \int_{-\infty}^{+\infty} 1(t)\, x(t)\, e^{-j\omega t}\, dt \right|^2 \qquad \ldots\ldots\ldots\ldots 4.1.$$

In the jargon of spectral analysis the function 1(t) is termed a 'window function', and multiplying the data record by a window function is known as 'linear modification'. The class of window functions used in this thesis satisfies the general conditions :-

$$l(-t) = l(t)$$

$$l(t) = 0 \quad ; \quad |t| > \frac{T}{2} \qquad \ldots\ldots\ldots\ldots 4.2.$$

$$q = \int_{-\infty}^{+\infty} l^2(t)dt$$

In the next section it is shown that the sample autospectrum is a biased estimator, and that only as the sample duration, T, tends to infinity does the bias become zero. However, for finite sample durations, and specific autospectrum shapes, the bias can be minimised by controlling the nature of the bias error. This is the purpose of linear modification, since by varying the shape of the window function we can control the shape of the frequency filter through which the true autospectrum is observed. This point is discussed more fully in sections 4.2.3., and 4.2.6.

b) The Periodogram

The discrete sample autospectrum (the periodogram) used in the real-time digital computer programmes is defined as follows :-

Consider an N point time series x_i (Where, for N even, $i = -\frac{N}{2}, \ldots\ldots 0, \ldots\ldots + \frac{N}{2} - 1$) obtained by sampling a zero-mean, gaussian, random process every Δ seconds. If the time series is multiplied by a weighting sequence l_i which satisfies the conditions :-

$$l_i = l_{-i}$$

$$l_i = 0 \quad ; \quad -\frac{N}{2} > i \geqslant \frac{N}{2} \qquad \ldots\ldots\ldots 4.3.$$

$$\frac{1}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} l_i^2 = q$$

Then the periodogram of x_i can be defined as a discrete version of a sample autospectrum.

$$C_{xx}(\omega_k) = \frac{N\Delta}{q} |A_k|^2$$

$$\text{Where; } A_k = \frac{1}{N} \sum_{i=-\frac{N}{2}}^{+\frac{N}{2}} x_i l_i W_N^{ik} \qquad \ldots\ldots\ldots 4.4.$$

$$\omega_k = \frac{2\pi k}{\Delta N}$$

$$W_N = \exp(-j\frac{2\pi}{N})$$

4.2.3. Statistical Properties of the Sample Autospectrum

The properties of the sample autospectrum are now developed. The corresponding properties of the discrete autospectral estimator are derived in section 4.2.7.

a) The Expected Value

From equation 4.1. the expected value of the sample autospectrum is :-

$$E\left\{Cxx(\omega)\right\} = \frac{1}{q} E\left\{ \int_{-\infty}^{+\infty}\int 1(t_1)1(t_2)\ x(t_1)x(t_2)e^{-j\omega(t_1-t_2)}\, dt_1 dt_2 \right\}$$

$$= \frac{1}{2\pi q} \int_{-\infty}^{+\infty} L^2(v)\ \Gamma xx(v-\omega)\ dv \qquad \ldots\ldots\ldots\ldots\ 4.5.$$

Where $L(\omega)$ is the Fourier transform of $1(t)$, and $\Gamma xx(\omega)$ is the autospectrum of $x(t)$.

The integral in equation 4.5. is the well known convolution integral, and equation 4.5. was obtained using the expressions which relate multiplication and convolution in the time-domain and frequency-domain. These expressions are :-

$$\int_{-\infty}^{+\infty} x(t)\ y(t)\ e^{-j\omega t}\, dt = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(v)\ Y(v-\omega)\ dv$$

$$\int_{-\infty}^{+\infty} \left\{ \int_{-\infty}^{+\infty} x(u)\ y(t-u)\ du \right\} e^{-j\omega t}\, dt = Y(\omega)\ X(\omega)$$

Where ω and v are <u>angular</u> frequencies

The convolution integral is usually represented by a star. Using this notation equation 4.5. is written as :-

$$E\left\{ Cxx(\omega) \right\} = \frac{1}{2\pi q} \left\{ L^2(\omega) * \Gamma xx(\omega) \right\}$$

The constant q was chosen so that when the signal is white, and the bias due to windowing is zero, the expectation of the sample autospectrum estimator is the same as the true auto-

spectrum. For instance, consider the case when x(t) is a white noise process with variance σ_{xx}^2. In this case the autospectrum of x(t) is constant and equal to the variance. Substituting for the autospectrum in equation 4.1., and applying Parseval's Theorem we obtain :-

$$E\left\{Cxx(\omega)\right\} = \frac{\sigma_{xx}^2}{2\pi q} \int_{-\infty}^{+\infty} L^2(\omega)\, d\omega = \sigma_{xx}^2$$

When the true spectrum is not a constant, then the sample autospectrum is a biased estimator. The estimator is biased because the window function l(t) is of finite duration. As the duration of the record (and the window) increases then $\frac{L^2(\omega)}{q}$ tends to a unit impulse. The sample autospectrum is therefore asymptotically unbiased.

Approximate expressions for the bias due to finite record length can be obtained using the method of Jenkins (ref. 4.2.). The bias is given by :-

$$B\left\{Cxx(\omega)\right\} = E\left\{Cxx(\omega)\right\} - \Gamma_{xx}(\omega)$$

$$= \int_{-\infty}^{+\infty} \left\{\frac{l(t)*l(t)}{q} - 1\right\} \gamma_{xx}(t) e^{-j\omega t}\, dt \qquad \ldots\ldots\ldots 4.6.$$

The term l(t)*l(t) denotes the self convolution of the window function. Expanding $\frac{1}{q}\left\{l(t)*l(t)\right\}$ as a power series in the range $-\frac{T}{2}$ to $+\frac{T}{2}$, and assuming $\gamma_{xx}(t)$ is zero for t greater than

$\frac{T}{2}$, we can write :-

$$B\left\{Cxx(\omega)\right\} \simeq \int_{-\infty}^{+\infty}\left\{\frac{a_2 t^2}{T^2} + \frac{a_4 t^4}{T^4} + \ldots\ldots\right\}\gamma_{xx(t)}e^{-j\omega t}dt$$

$$= \frac{a_2}{-T^2}\ \Gamma_{xx}{}^{(2)}(\omega) + \ldots\ldots \frac{a_n\ \Gamma_{xx}{}^{(n)}(\omega)}{(-1)^{\frac{n}{2}}\ T^n}$$

$$\ldots\ldots\ldots\ 4.7.$$

Where $\Gamma_{xx}{}^{(n)}(\omega)$ is the n^{th} derivative of $\Gamma_{xx}(\omega)$ with respect to ω, a_n is the n^{th} coefficient in the power series expansion of $\frac{1}{q}\left\{1(t)*1(t)\right\}$.

For most practical purposes only the term in a_2 need be considered when evaluating the bias in autospectrum estimators. Neglecting all the higher terms in equation 4.7. we can say that the bias due to windowing may be controlled by choosing T such that $\frac{\Gamma_{xx}{}^{(2)}(\omega)}{\Gamma_{xx}(\omega)} \ll \frac{T^2}{a_2}$, and choosing a window $1(t)$ which minimises the term a_2.

b) <u>The Covariance</u>

The covariance of the sample cross-spectrum is derived for the zero mean, gaussian case in appendix 2. From appendix 2 the covariance of the sample autospectrum at the frequencies ω_1 and ω_2 is given by :-

$$\mathrm{Cov}\left\{Cxx(\omega_1)\ Cxx(\omega_2)\right\} = \frac{1}{(2\pi q)^2}\left\{ \Gamma^2_{xx}(\omega_2)\left\{ L(\omega_1-\omega_2)*L(\omega_1-\omega_2)\right\}^2 + \Gamma^2_{xx}(\omega_2)\left\{L(\omega_1+\omega_2)*L(\omega_1+\omega_2)\right\}^2\right\}$$

..... 4.8.

The variance of the sample autospectral estimator at frequency ω, can be obtained by rewriting the above expression in an equivalent form :-

$$\mathrm{Var}\left\{Cxx(\omega)\right\} = \frac{1}{(2\pi q)^2}\left\{ \left\{L^2(\omega) * \Gamma_{xx}(\omega)\right\}^2 + L^2(\omega)\left\{L(\omega) * \Gamma_{xx}(\omega)\right\}^2\right\}$$

..... 4.9.

The spectral windows $L(\omega)$ have low-pass characteristics, and decay rapidly either side of $\omega = 0$. Therefore, to a good approximation :-

$$\mathrm{Var}\left\{Cxx(\omega)\right\} = E^2\left\{Cxx(\omega)\right\} \qquad .....4.10.$$

Except when ω is close to zero, then :-

$$\mathrm{Var}\left\{Cxx(\omega)\right\} \simeq 2\ E^2\left\{Cxx(\omega)\right\}$$

From equation 4.10. we have the important result that the variance of the sample autospectrum does not decrease when the sample duration, T, is increased. To decrease the variability of the estimator, some sort of smoothing is required. The method used to smooth the estimates in the on-line digital computer programmes is to average the periodograms of successive data blocks. The analogous technique in continuous time is to smooth by averaging the sample autospectra of successive data blocks. In the next

section the properites of smoothed autospectral estimates obtained in this way are given.

4.2.4. Properties of the Smoothed Autospectrum Estimator

Let a continuous record of a signal x(t) of duration nT seconds be sub-divided into n segments, where the i^{th} segment is defined by :-

$$x^{(i)}(t) = X(t + iT - \frac{T}{2}) \quad ; \quad -\frac{T}{2} < t < \frac{T}{2}$$

Then, if the i^{th} sample spectrum is defined as :-

$$C^{(i)}_{xx}(\omega) = \frac{1}{q} \left| \int_{-\infty}^{+\infty} x^{(i)}(t) 1(t) e^{-j\omega t} dt \right|^2$$

The smoothed sample autospectrum is :-

$$\bar{C}xx(\omega) = \frac{1}{n} \sum_{i=1}^{n} C^{(i)}_{xx}(\omega) \qquad \ldots\ldots\ldots\ldots 4.11.$$

We now consider the expected value and the variance of the smoothed autospectrum estimator.

a) The Expected Value

Since the expected values of the component sample autospectra are equal, then the expected value of the smoothed autospectrum estimate is equal to the expected value of the sample autospectrum estimators.

b) The Covariance

For n independent samples of the random variable x(t) the variability of the smoothed estimate will be $\frac{1}{n}$ times that of the sample estimator. It can also be shown that if the samples are consecutive windowed segments of the same record, the variability of the smoothed estimate is still reduced by $\frac{1}{n}$. This is contrary to the suggestion of Bartlett (ref. 4.3.), who states that $\gamma_{xx}(t)$ should be zero for $t > T$ if the periodograms of adjacent segments are to be independent.

4.2.4. The Effect of Correcting the Autospectrum Estimator for Non-Zero Mean

The autospectrum of a random signal x(t) has been defined as the Fourier transform of its autocovariance function $\gamma_{xx}(t)$. For a random signal with a non-zero mean μ, the autocovariance function is defined by :-

$$\gamma_{xx}(t) = \lim_{T \to \infty} \left\{ \frac{1}{2T} \int_{-T}^{+T} (x(u) - \mu)(x(t-u) - \mu)\, du \right\}$$

Up to now we have assumed that the random signal x(t) has zero mean. If however, the signal has a significant mean level, it must be removed prior to Fourier transformation.

First consider the effect of correcting the sample autospectrum using the true mean. The expected value of the corrected estimator is :-

$$E\left\{\bar{C}xx(\omega)\right\} = \frac{1}{q}\left\{\frac{1}{2\pi}\left\{ L^2(\omega) * Sxx(\omega)\right\} - L^2(\omega)\mu^2\right\}$$

$$= \frac{1}{2\pi q}\left\{ L^2(\omega) * \bar{C}xx(\omega)\right\}$$

Where $Sxx(\omega)$ is the Fourier transform of the auto-correlation function $Rxx(t)$.

The effect of correcting using the true mean is to subtract an impulse of strength μ^2 from the $\omega = 0$ point of $Sxx(\omega)$. Note that if the mean were not adjusted for, the spectrum around $\omega = 0$ would distort. To correct this distortion the function $\frac{1}{q} L^2(\omega)\mu^2$ would have to be subtracted from $\bar{C}xx(\omega)$.

In the on-line digital computer programmes for auto-spectrum estimation the periodograms cannot be corrected using the true signal mean, since it will not generally be known. Because of this the mean level is compensated for by subtracting the sample mean, $\bar{x}$, from each segment. Where the sample mean is defined as :-

$$\bar{x} = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} x(t)dt$$

Since the sample mean is only an estimate of the true mean, it is to be expected that the statistics of the corrected autospectral estimates will be altered.

Consider the expected value of the mean-corrected sample

autospectrum :-

$$E\left\{Cxx(\omega)\right\} = \frac{1}{q}\int_{-\infty}^{+\infty}\int dt_1 dt_2 l(t_1)l(t_2)\, E\left\{(x(t_1)-\bar{x})(x(t_2)-\bar{x})\right\} e^{-j\omega(t_1-t_2)}$$

$$= \frac{1}{2\pi q}\left\{ L^2(\omega) * \Gamma_{xx}(\omega)\right\} + I_1 \qquad \ldots\ldots\ldots\ldots \text{4.12.}$$

$$I_1 = \frac{L(\omega)}{q}\left\{\frac{L(\omega)}{T}\int_{-T}^{+T}(1-\frac{|t|}{T})\gamma_{xx}(t)dt - \frac{\Gamma_{xx}(\omega)}{\pi}\left\{L(\omega) * \frac{\sin\left\{\frac{\omega T}{2}\right\}}{\frac{\omega T}{2}}\right\}\right\}$$

Where I_1 is the error due to correcting for non-zero mean with the sample mean.

From equation 4.12. the effect of using the sample mean to correct for non-zero mean is to introduce extra bias terms in the region of $\omega = 0$. The size of the additional bias terms depends upon the sample record length, and the extent to which the bias near the origin affects other regions of the auto-spectrum is governed by the shape of the spectral window, $L(\omega)$.

The effect on the variability of the autospectrum estimator of using the sample mean to correct for a non-zero mean level is more difficult to evaluate. However, using the methods of appendix 2 it can be shown that the variance of the mean-corrected autospectrum estimator is approximately given by :-

$$\text{Var}\left\{\bar{C}xx(\omega)\right\}_{\text{corrected}} = \text{Var}\left\{\bar{C}xx(\omega)\right\} + I_2 \qquad \ldots\ldots\ldots\ldots 4.13.$$

$$I_2 = \frac{3}{q^2\pi^2 n}\left\{L(\omega)\left\{L(\omega) * \Gamma_{xx}(\omega)\right\}\right\}^2$$

From equation 4.13. it is seen that the effect on the covariability of correcting for the mean level is to inflate the covariance in the region $\omega = 0$. Like the additional bias, the extent to which the extra covariability affects regions of the spectrum in the vicinity of the origin depends upon the shape of the spectral window $L(\omega)$. However, unlike the extra bias terms, the extra covariability reduces with 'n'.

To show the likely magnitude of these effects consider the following simple example. Let, $x(t)$ be a gaussian white noise process with variance σ^2_{xx}, and let the window function be the simple rectangular function $l_R(t)$, defined as :-

$$l_R(t) = (T)^{-\frac{1}{2}} \qquad |t| < \frac{T}{2}$$

$$l_R(t) = 0 \qquad |t| > \frac{T}{2}$$

$$q_R = 1$$

Substituting for $l_R(t)$ and putting $\Gamma_{xx}(\omega) = \sigma^2_{xx}$ in equations 4.12., the bias error due to correcting with sample mean is :-

$$I_1 = - \sigma_{xx}^2 \left\{ \frac{\sin\left(\frac{\omega T}{2}\right)}{\frac{\omega T}{2}} \right\}^2$$

When $\omega = 0$, $I_1 = - \sigma_{xx}^2$. Therefore at zero frequency the expectation of the sample mean-corrected autospectrum is zero.* As the frequency is increased the magnitude of I_1 decreases rapidly, and at $\omega = \frac{2\pi}{T}$ the error has its first zero. Beyond this frequency the maximum value that I_1 attains is $- \sigma_{xx}^2$ 0.045 at $\omega = \frac{3\pi}{T}$.

Substituting for $l_R(t)$ and $\Gamma_{xx}(\omega)$ in equation 4.13., the additional variance term caused by the sample mean-correction is :-

$$I_2 = 12 \cdot \frac{\sigma_{xx}^4}{n} \left\{ \frac{\sin\left(\frac{\omega T}{2}\right)}{\frac{\omega T}{2}} \right\}^2$$

From this expression it is seen that, at $\omega = 0$ the variability is inflated by a factor of 12 by the sample mean correction. However above this frequency the additional variability, like the bias error, decreases rapidly and is negligible for $\omega > \frac{2\pi}{T}$. Therefore in this simple example we have shown that the errors introduced by correcting the autospectrum estimate using the sample mean are negligible above $\omega \approx \frac{2\pi}{T}$. In discrete autospectrum estimation this means that only the d.c. periodogram component need be discarded.

*This is only strictly true when $L(\omega)$ is a rectangular window function. However, in practice the author has found this to be very nearly true when $L(\omega)$ is a Generalised Hanning Window.

4.2.5. Sampling Distribution and Confidence Intervals for Autospectrum Estimators

The sample spectrum of a normal r.v. $x(t)$ may be written as :-

$$Cxx(\omega) = \frac{1}{q}\left\{\int 1(t_1)x(t_1)\cos\omega t_1 dt_1\right\}^2 + \frac{1}{q}\left\{\int 1(t_2)x(t_2)\sin\omega t_2 dt_2\right\}^2$$

$$= \left[\mathrm{Re}\left\{X(\omega)\right\}\right]^2 + \left[\mathrm{Im}\left\{X(\omega)\right\}\right]^2$$

Where $X(\omega)$ is the Fourier transform of $x(t)1(t)$.

Both $\left[\mathrm{Re}\ X(\omega)\right]^2$, and $\left[\mathrm{Im}\ X(\omega)\right]^2$ are chi-square variates with one degree of freedom. Using the orthogonality of $\sin\omega t_1$, $\cos\omega t_2$, it can be shown that $\left[\mathrm{Re}\ X(\omega)\right]^2$ and $\left[\mathrm{Im}\ X(\omega)\right]^2$ are independent. Hence $Cxx(\omega)$ is distributed as a chi-square variate with two degrees of freedom. In a similar fashion, the smoothed spectrum estimator can be approximated to a chi-square distribution.

If the approximating distribution is a chi-square distribution with ν degrees of freedom and expected value a, the parameter a, and ν can be found by equating the first two moments of the estimator to the corresponding moments of the distribution.

$$E\left\{\bar{C}xx(\omega)\right\} = a\nu$$

$$\mathrm{Var}\left\{\bar{C}xx(\omega)\right\} = 2a^2\nu$$

Therefore :-

$$\nu = \frac{2\,E^2\left\{\bar{C}xx(\omega)\right\}}{\mathrm{Var}\left\{\bar{C}xx(\omega)\right\}} \simeq 2n \qquad \ldots\ldots\ldots\ldots 4.14.$$

$$a = \frac{E\left\{\bar{C}xx(\omega)\right\}}{\nu} \simeq \frac{\Gamma xx(\omega)}{\nu}$$

The quantity $\frac{\nu \bar{C}xx(\omega)}{\Gamma xx(\omega)}$ is approximately chi-square distributed with 2n degrees of freedom. Note that this applies no matter what window function l(t) is used. Confidence intervals for $\Gamma xx(\omega)$ can be constructed using tabulated chi-square distributions.

4.2.6. Linear Modification

In this section the reasons for linearly modifying the data segments are given, and the properties of the window function used in the on-line digital computer programme are discussed.

The expected value of the autospectrum estimator has been shown to be the convolution of $L^2(\omega)$ with the true spectrum $\Gamma xx(\omega)$. Ideally $L^2(\omega)$ should be impulsive, however this is only so in the limiting case when the sample duration tends to infinity. In practice $L^2(\omega)$ will have the characteristics of a low pass filter, that is, it will have a non-zero bandwidth and a finite asymptotic decay rate. The estimated autospectrum is thus a distorted version of the true autospectrum. There are two types of distortion that can be expected, one due to the 'smearing together' of close frequencies, and the other due to the interaction of distant frequencies. The first of these is determined by the bandwidth of the spectral window, and to reduce local distortion a narrow bandwidth is required. The interaction of distant parts of the autospectrum is termed 'leakage', and is the phenomenon whereby a strong periodic component in an autospectrum can swamp low amplitude variations in the autospectrum. To suppress this type of inter-

action a window function with a high asymptotic decay rate is required.

Unfortunately the bandwidth, and asymptotic decay rate of a window cannot (for fixed T) be independently specified. Improving one feature tends to affect the other adversely. Because of this trade-off between bandwidth and asymptotic rate, the choice of window must depend upon the shape of the autospectrum to be measured. In the on-line digital computer programmes the author sought to overcome this problem by using the most sophisticated window that could be rapidly executed in a real-time spectral analysis programme, and to build into the window some control over the trade-off between bandwidth and asymptotic rate. This enables the operator to adjust the balance between bandwidth and asymptotic rate at run-time.

We now restrict our attention to the data window function used in the on-line digital computer programmes. This is the Generalised Hanning window, defined by :-

$$
\begin{aligned}
l_H(t) &= \tfrac{1}{2}\, T^{-\frac{1}{2}} \left\{ 1 - \cos(\omega_p t + \phi) \right\} \quad ; \quad -\frac{T}{2} < t < -\frac{T}{2}(\tfrac{1}{2} - p) \\
&= T^{-\frac{1}{2}} \quad ; \quad -T(\tfrac{1}{2} - p) < t < T(\tfrac{1}{2} - p) \qquad \ldots\ldots\ldots \quad 4.15. \\
&= \tfrac{1}{2}\, T^{-\frac{1}{2}} \left\{ 1 - \cos(\omega_p t - \phi) \right\} \quad ; \quad T(\tfrac{1}{2} - p) < t < \frac{T}{2}
\end{aligned}
$$

where $\omega_p = \frac{2\pi}{T.p}$, and $\phi = \omega_p \cdot \frac{T}{2}$, and T is the duration of the data sample, the parameter p may take any value between 0 and $\frac{1}{2}$.

The Generalised Hanning window was chosen as the most efficient for real-time use because it can operate using the existing FFT trignometric look-up tables. Alternative windows of comparable sophistication (ref. 4.4.) employ power law curves and would thus require separate look-up tables. The possibility of of working without look-up tables was explored but rejected because it seriously wasted computing time.

The Fourier transform of the Generalised Hanning window is :-

$$L_H(\omega) = L_R(\omega, T(1-p)) \cdot U_H(\omega, p) \quad \ldots\ldots\ldots\ldots \quad 4.16.$$

Where, $$L_R(\omega, T(1-p)) = \left\{T(1-p)\right\}^{\frac{1}{2}} \cdot \left\{\frac{\sin\left\{\frac{\omega T}{2}\cdot(1-p)\right\}}{\frac{\omega T}{2}(1-p)}\right\}$$

and, $$U_H(\omega, p) = (1-p)^{\frac{1}{2}} \left\{\frac{\cos\left\{\frac{\omega T}{2}\, p\right\}}{1-\left(\frac{\omega}{\omega_p}\right)^2}\right\}$$

The Fourier transform of the Generalised Hanning window is given as the product of the transform of a rectangular window of duration $T(1-p)$ seconds and the function U_H. This function is fairly constant from $\omega = 0$ to ω_p, its envelope then decays at a rate proportional to ω^{-2}. It therefore controls the asymptotic rate of decay of the Generalised Hanning window, since by varying p between 0 and $\frac{1}{2}$, the breakpoint of U_H can be varied between $\frac{2\pi}{T}$ and infinity.

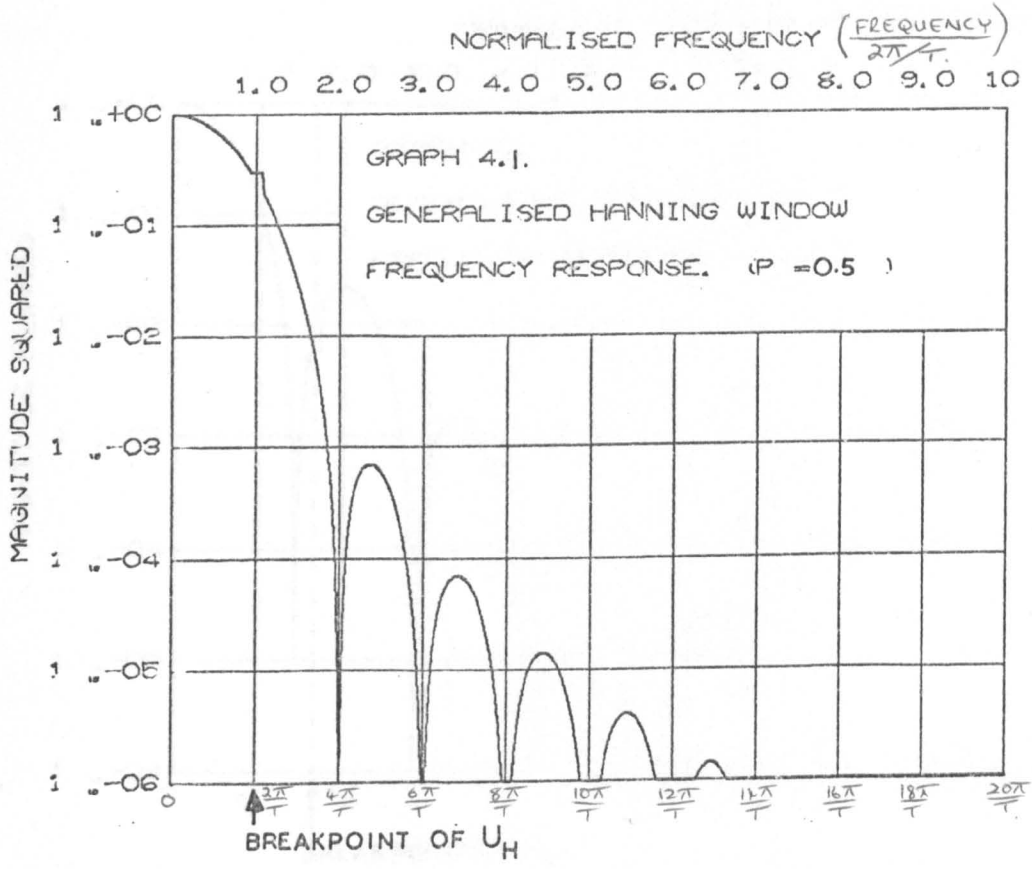
The effect upon the rectangular window of varying p is to alter the bandwidth. When p is small the width of L_R is small, and as p increases so does the width of the rectangular spectral window. Since the bandwidth of L_H is controlled by L_R, varying p results in a trade-off between bandwidth and asymptotic rate. The half-power bandwidth of the Generalised Hanning window is approximately given by :-

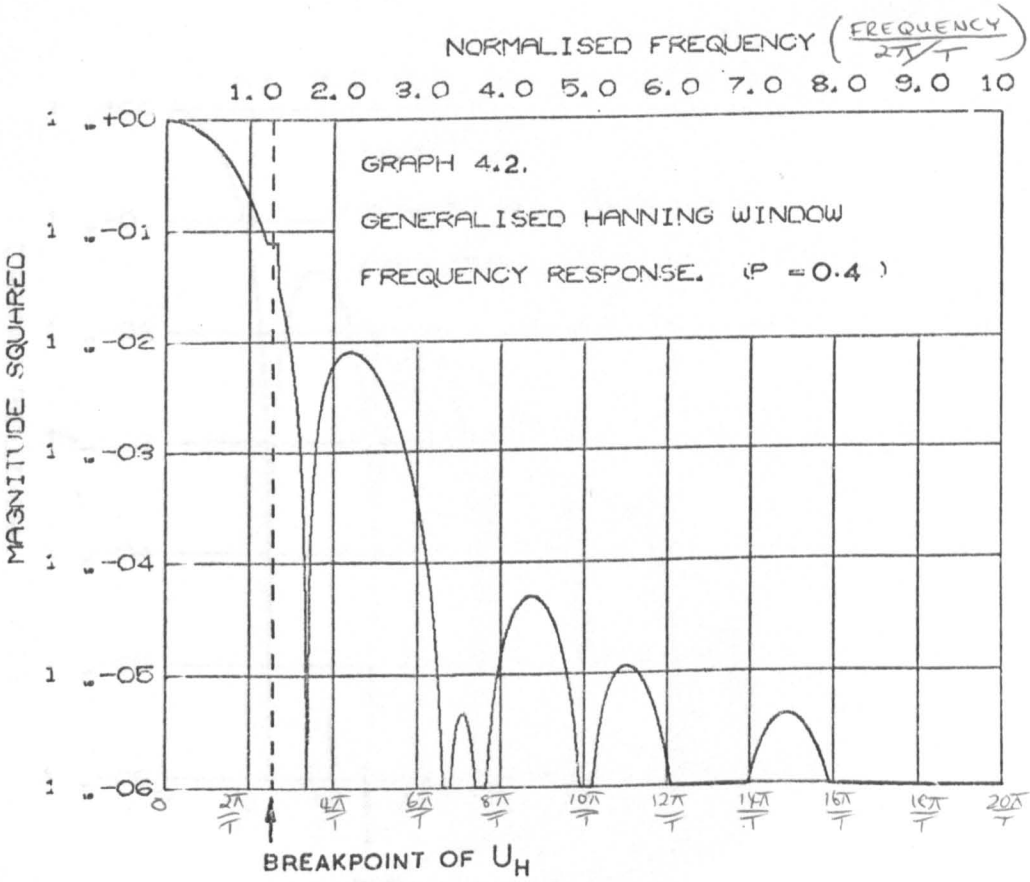
$$\omega_B = \frac{2\pi}{T}(1 + 2p)$$

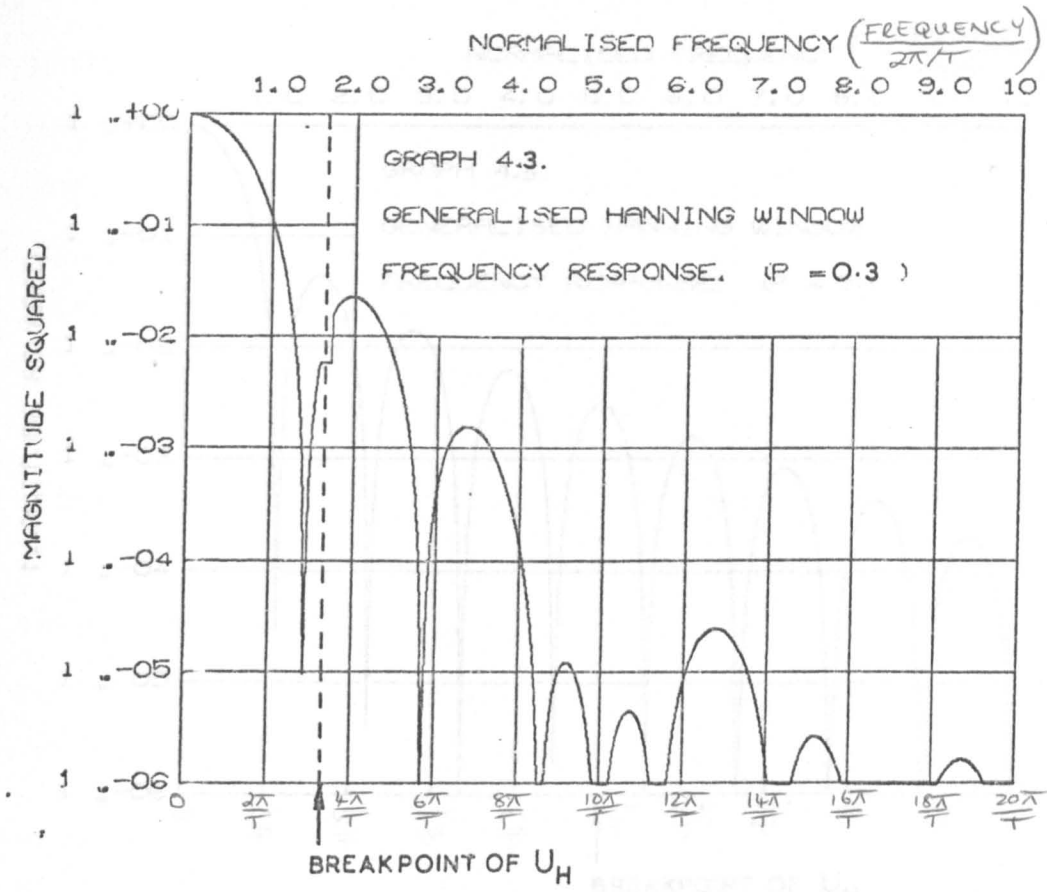
The asymptotic decay rate of the Generalised Hanning window is proportional to ω^{-1} for $\omega < \frac{2\pi}{T.p}$, and proportional to ω^{-3} for $\omega > \frac{2\pi}{T.p}$.

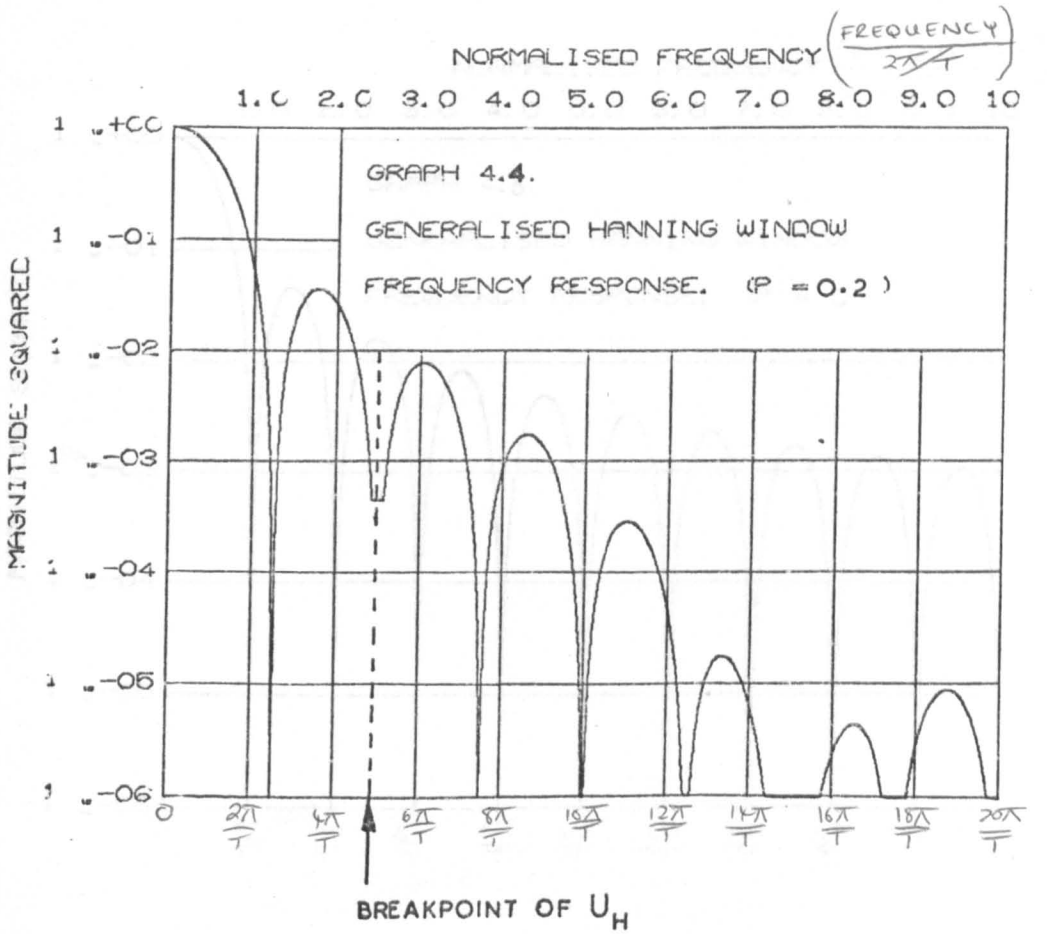
The effect of varying p on the bandwidth and asymptotic rate is illustrated by graphs 4.1. to 4.6. These graphs show the behaviour of $L_H^2(\omega)$ for various values of p. The vertical scales are normalised so that $L_H^2(0) = 1$, and the horizontal scales are calibrated in multiples of $\frac{2\pi}{T}$. The arrows indicate the breakpoints of the function U_H.

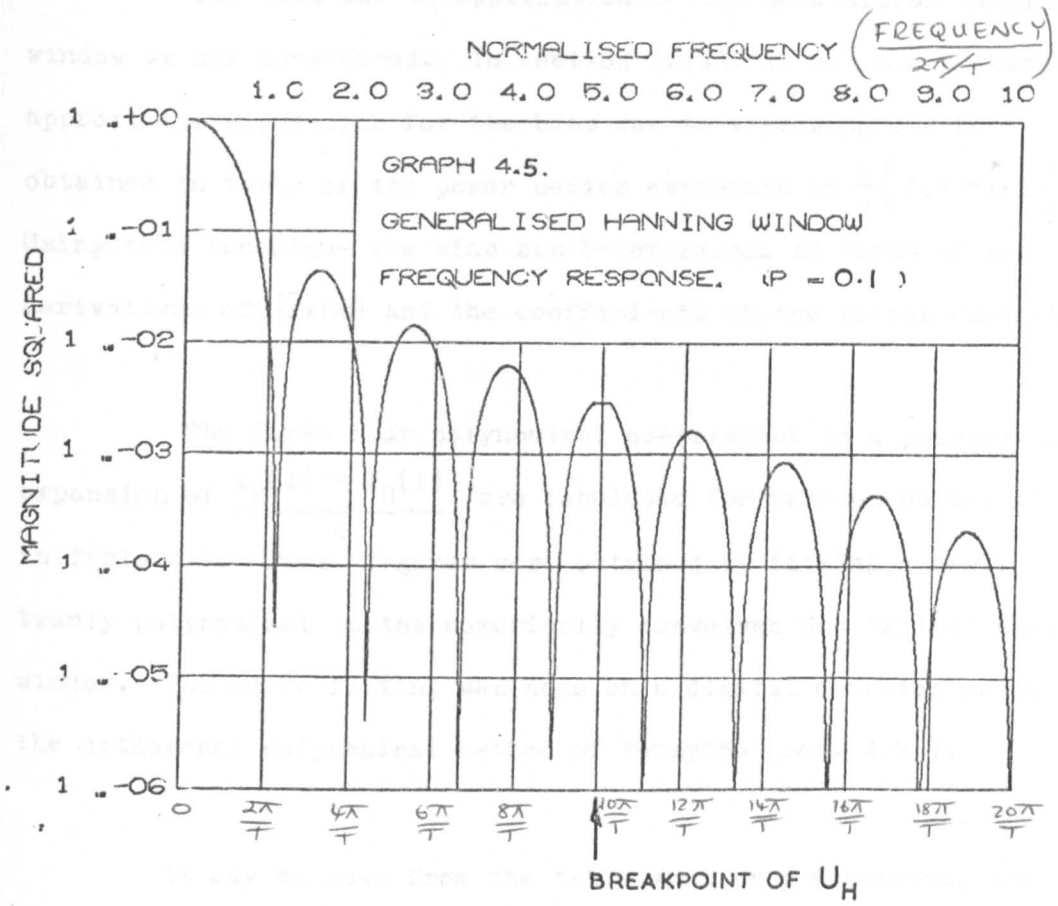
Graph 4.1. (p = 0.5) shows the most extreme use of the Generalised Hanning window when the entire record is modified. Note the wide central lobe, and the rapid decay rate of the minor lobes. Graph 4.6. (p = 0) shows the limiting case of the Generalised Hanning window, when none of the data is modified. In this case the window becomes a rectangular function $L_R^2(\omega, T)$. Note the narrow central lobe, and the slow decay rate of the minor lobes.

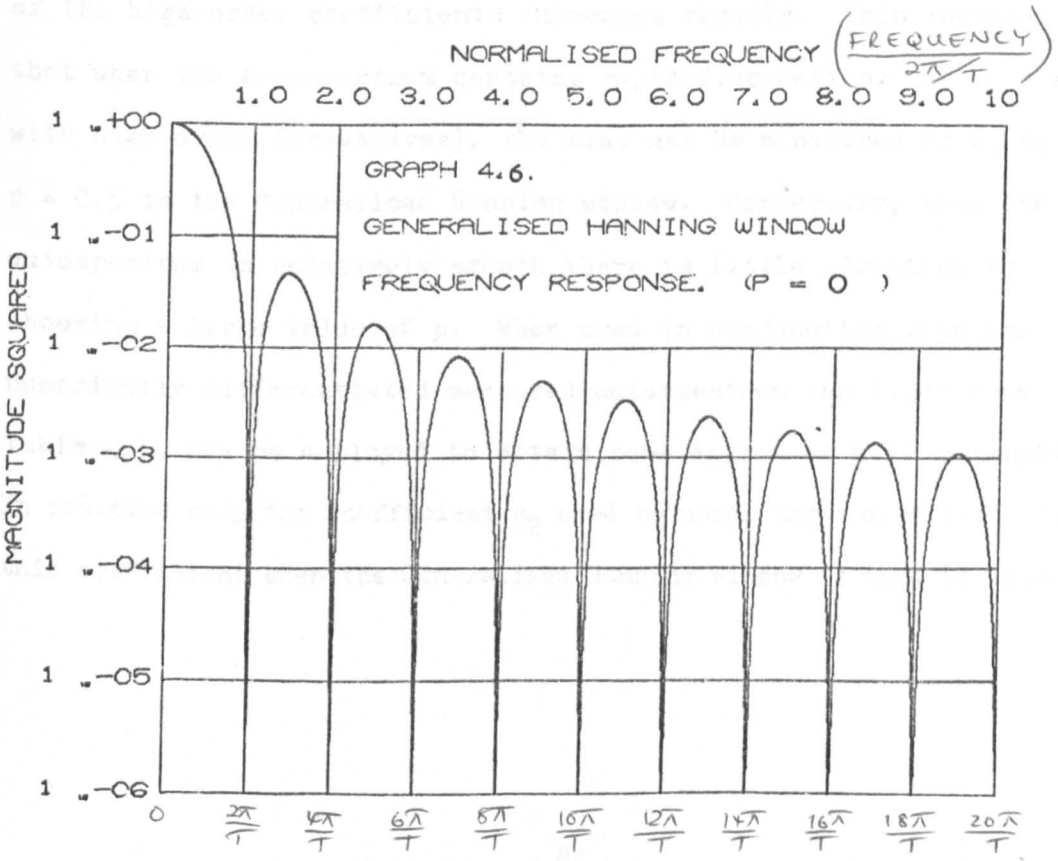

NORMALISED FREQUENCY (FREQUENCY / 2π/T)
1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10
MAGNITUDE SQUARED
1 +00
1 -01
1 -02
1 -03
1 -04
1 -05
1 -06
0
2π/T 4π/T 6π/T 8π/T 10π/T 12π/T 14π/T 16π/T 18π/T 20π/T
BREAKPOINT OF U_H
GRAPH 4.1.
GENERALISED HANNING WINDOW
FREQUENCY RESPONSE. (P = 0.5)


NORMALISED FREQUENCY (FREQUENCY / 2π/T)
1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10
MAGNITUDE SQUARED
1 +00
1 -01
1 -02
1 -03
1 -04
1 -05
1 -06
0
2π/T 4π/T 6π/T 8π/T 10π/T 12π/T 14π/T 16π/T 18π/T 20π/T
BREAKPOINT OF U_H
GRAPH 4.2.
GENERALISED HANNING WINDOW
FREQUENCY RESPONSE. (P = 0.4)


NORMALISED FREQUENCY (FREQUENCY / 2π/T)
1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10
MAGNITUDE SQUARED
1 +00
1 -01
1 -02
1 -03
1 -04
1 -05
1 -06
0
2π/T 4π/T 6π/T 8π/T 10π/T 12π/T 14π/T 16π/T 18π/T 20π/T
GRAPH 4.3.
GENERALISED HANNING WINDOW
FREQUENCY RESPONSE. (P = 0.3)
BREAKPOINT OF U_H


NORMALISED FREQUENCY (FREQUENCY / 2π/T)
1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10
MAGNITUDE SQUARED
1 +00
1 -01
1 -02
1 -03
1 -04
1 -05
1 -06
0
2π/T 4π/T 6π/T 8π/T 10π/T 12π/T 14π/T 16π/T 18π/T 20π/T
GRAPH 4.4.
GENERALISED HANNING WINDOW
FREQUENCY RESPONSE. (P = 0.2)
BREAKPOINT OF U_H


NORMALISED FREQUENCY (FREQUENCY / 2π/T)
1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10
MAGNITUDE SQUARED
1 +00
1 -01
1 -02
1 -03
1 -04
1 -05
1 -06
GRAPH 4.5.
GENERALISED HANNING WINDOW
FREQUENCY RESPONSE. (P = 0.1)
0
2π/T
4π/T
6π/T
8π/T
10π/T
12π/T
14π/T
16π/T
18π/T
20π/T
BREAKPOINT OF U_H


NORMALISED FREQUENCY (FREQUENCY / 2π/T)
1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10
MAGNITUDE SQUARED
1 +00
1 -01
1 -02
1 -03
1 -04
1 -05
1 -06
GRAPH 4.6.
GENERALISED HANNING WINDOW
FREQUENCY RESPONSE. (P = 0)
0
2π/T
4π/T
6π/T
8π/T
10π/T
12π/T
14π/T
16π/T
18π/T
20π/T

The bias due to application of the Generalised Hanning window is now considered. In section 4.2.3. it was shown that an approximate expression for the bias due to windowing can be obtained in terms of the power series expansion of $\frac{1}{q}\left\{1(t)*1(t)\right\}$. Using this technique the bias can be expressed in terms of the derivatives of $\Gamma_{xx}(\omega)$ and the coefficients of the series expansion.

The first four polynominal coefficient in a polynominal expansion of $\frac{1_H(t) * 1_H(t)}{q}$ are tabulated for various values of p in Table 4.1. These figures were obtained by fitting a degree twenty polynominal to the numerically convolved Generalised Hanning window. The curve fitting was done on a digital computer using the orthogonal polynominal method of Forsythe (ref. 4.5.).

It may be seen from the table that as p decreases, the size of the high-order coefficients increases rapidly. This indicates that when the autospectrum contains rapid fluctuations (associated with high order derivatives), the bias may be minimised by using $p = 0.5$ in the Generalised Hanning window. Conversely, when the autospectrum is relatively smooth there is little advantage in choosing a large value of p. When used in conjunction with the numerically differentiated measured autospectrum the figures in Table 4.1. may be employed to obtain some degree of bias compensation. In practice only the coefficient a_2 need be used, and a mean value for this coefficient when the Generalised Hanning window is used is -6.2.

In the real-time autospectrum estimation programme bias compensation should not be necessary since the parameters N and p can be varied on-line to reduce the bias to negligible proportions. However the technique of expressing bias as a function of derivatives of the autospectrum is still extremely useful because it enables us to establish the cause of bias errors. In chapters 5 and 6 for instance the technique is used to establish the possible sources of bias error in frequency response estimators.

TABLE 4.1.

Table of the first four polynominal coefficients obtained by fitting an order twenty orthogonal polynominal to the self convolute of the Generalised Hanning window for various values of p.

p	a_2	a_4	a_6	a_8
0.5	-6.56	20.2	-41.35	71.3
0.4	-6.1	26.3	-113.2	376.7
0.3	-6.25	38.7	-204.9	704
0.2	-6.67.	56.7	-371	1487
0.1	-6.73	64.7	-441.5	1785
0	-6.43	66	-470	1918

In the table a_n is defined by :-

$$\frac{1}{q}\left\{1_H(t) * 1_H(t)\right\} = \sum_{n=0}^{20} a_n t^n$$

and because the Generalised Hanning window is an even function, coefficients of odd powers of t are zero.

4.2.7. Discrete Autospectrum Estimation

4.2.7.(a) Discrete Estimation Formulea

In the previous pages the properties of a continuous autospectrum estimator have been presented. In this section these results are used to obtain the corresponding properties of the equivalent discrete autospectral estimator. This discrete estimator is the one used throughout the real-time digital autospectrum estimation programmes.

Firstly the expressions which relate the discrete Fourier transform and the continuous Fourier transform are given :-

The discrete Fourier transform of an N point time series x_i, modified by the window function l_i, is defined as :-

$$A_k = \frac{1}{N} \sum_{i=-\frac{N}{2}}^{+\frac{N}{2}} x_i l_i W_N^{ik}$$

where, $W_N = \exp(-j\frac{2\pi}{N})$

The Fourier transform of a signal x(t) modified by l(t) is defined as :-

$$X(\omega) = \int_{-\infty}^{+\infty} x(t)l(t)e^{-j\omega t} dt$$

If the time series was obtained by sampling x(t) every Δ seconds, then the discrete, and continuous, Fourier transforms are related by :-

$$A_k = \frac{1}{\Delta N} \sum_{n=-\infty}^{+\infty} X(\omega_{k-nN}) \qquad \ldots\ldots\ldots\ldots 4.17.$$

Where $\omega_{k-nN} = \frac{2\pi(k-nN)}{\Delta N}$

The discrete Fourier transform is thus a discrete aliased, form of continuous Fourier transform.

Now the properties of the discrete autospectral estimators are obtained assuming that the spectrum of x(t) is zero above $\omega = \frac{\pi}{\Delta}$, and that spreading of the spectrum caused by windowing can be neglected. Under these circumstances the aliasing can be ignored, so that from section 4.2.2. the discrete sample autospectrum is :-

$$Cxx(\omega_k) = \frac{N.\Delta}{q} \left\{ |A_k| \right\}^2 \qquad \ldots\ldots\ldots\ldots 4.18.$$

Where $\omega_k = \frac{2\pi k}{\Delta N}$, and $k = 0, \ldots\ldots \frac{N}{2}$

The function $Cxx(\omega_k)$ is termed the 'raw' periodogram of x(t), and is a set of $\frac{N}{2} + 1$ points on the continuous sample autospectrum spaced at frequency intervals of $\frac{2\pi}{\Delta N}$. The smoothed discrete autospectral estimator is obtained by averaging raw periodograms of adjacent segment of time series. This is known as the smoothed periodogram.

The smoothed periodogram is defined by :-

$$Cxx(\omega_k) = \frac{1}{n} \sum_{i=1}^{n} Cxx^{(i)}(\omega_k) \qquad \ldots\ldots\ldots\ldots 4.19.$$

Where $Cxx^{(i)}$ are independent raw periodograms. The expected value of the smoothed periodogram is a discrete form of equation 4.5.

$$E\left\{\overline{C}xx(\omega_k)\right\} = \frac{1}{q}\sum_{v=-\infty}^{+\infty} L^2(\omega_v)\ \Gamma xx(\omega_{v-k})$$

$$= \frac{1}{q}\ L^2(\omega_k) * \Gamma xx(\omega_{v-k}) \quad \ldots\ldots\ldots\ldots 4.20.$$

Here the star indicates the discrete convolution of $L^2(\omega_v)$ and $\Gamma xx(\omega_v)$. The covariance of the smoothed periodogram components at the discrete frequencies ω_k and ω_h can be found by writing equation 4.9. in discrete form. Thus :-

$$Cov\left\{\overline{C}xx(\omega_k),\ \overline{C}xx(\omega_h)\right\}$$

$$= \frac{1}{q^2 n}\left\{ \Gamma^2_{xx}(\omega_h)\ \left\{L(\omega_{k-h}) * L(\omega_{k-h})\right\}^2 + \Gamma^2_{xx}(\omega_h)\left\{L(\omega_{k+h}) * L(\omega_{k+h})\right\}^2 \right\}$$

$$\ldots\ldots\ldots\ldots 4.21.$$

The discrete form of the Generalised Hanning window can be obtained from equation 4.16.

$$L_H(\omega_k) = L_R(\omega_k, N(1-p))\ U_H(\omega_k) \quad \ldots\ldots\ldots\ldots 4.22.$$

where $L_R(\omega_k), N(1-p)\ ) = \left\{\frac{(1-p)}{N\Delta}\right\}^{\frac{1}{2}} \frac{\sin(\pi k(1-p)\ )}{\pi\, k(1-p)}$

and $$U_H(\omega_k) = (1-p)^{\frac{1}{2}} \frac{\cos(\pi kp)}{1-(2pk)^2}$$

The smoothed periodogram is a set of $\frac{N}{2} + 1$ estimates of the autospectrum, spaced at frequency intervals $\frac{2\pi}{\Delta N}$. Since $\Delta.N$ is analogous to T(the sample duration), the calibration of the frequency axes in graphs 4.1. to 4.6. corresponds with the spacing of the members of the periodogram. Two important results which use this fact can be obtained from the graphs.

The first concerns the Generalised Hanning window when p = 0. In this case $L^2_H(\omega_k, N)$ is (from graph 4.6.) identically zero when $\omega = \frac{2\pi k}{\Delta N}$ $(k \neq 0)$. Therefore, from the discrete covariance relationship (equation 4.21.), the members of an unmodified periodogram (raw or smoothed) form a statistically independent set of estimators. The second result concerns the generalised Hanning window when p = 0.5. In this case $L^2(\omega, \frac{N}{2})$ is (from graph 4.1.) identically zero when $\omega = \frac{2\pi k}{\Delta N}$ $(k \neq 0,1)$. Therefore, each members of a fully Hanned periodogram is correlated only with its immediate neighbours.

4.2.7.(b) The Effects of Aliasing

We now consider the effects of aliasing upon the discrete autospectrum estimator. For simplicity it is assumed that the spectrum of x(t) is low-pass, and decays sufficiently fast so that only the first alias need be considered. Therefore from equation 4.17. we can write :-

$$A_k = \frac{1}{\Delta N} \left\{ X(\omega_k) + X^*(\omega_{N-k}) \right\} \quad \ldots\ldots\ldots\ldots \quad 4.23.$$

where, $\omega_k = \frac{2\pi k}{\Delta N}$, and $X^*(\omega)$ denotes the complex conjugate of $X(\omega)$.

From equation 4.23. we now define the aliased Fourier transform of $x(t)1(t)$ as :-

$$X_A(\omega_k) = X(\omega_k) + X^*(\omega_{N-k})$$

From this definition the aliased sample autospectrum is :-

$$Cxx_A(\omega_k) = \frac{1}{q} \left| X_A(\omega_k) \right|^2 \qquad \ldots\ldots\ldots\ldots 4.24.$$

$$= \frac{1}{q} \left| X(\omega_k) + X^*(\omega_{N-k}) \right|^2$$

Now let us consider the expectation and variance of the aliased estimator. From appendix 3 the expected value of the aliased autospectrum estimator is :-

$$E\left\{Cxx_A(\omega_k)\right\} \triangleq \Gamma xx(\omega_k) + \Gamma xx(\omega_{N-k}) \qquad \ldots\ldots\ldots\ldots 4.25.$$

Also from appendix 3, the variance of the aliased autospectrum estimator is given by :-

$$Var\left\{Cxx_A(\omega_k)\right\} \triangleq E^2\left\{Cxx_A(\omega_k)\right\} \qquad \ldots\ldots\ldots 4.26.$$

From the above results we conclude that aliasing does not effect the variability of the autospectral estimator, and that the

results of section 4.2.5. can be applied to aliased autospectral estimators. The undesirable effect of aliasing is to introduce bias errors into the estimated autospectrum. This bias may be controlled by introducing a low-pass filter into the autospectrum measurement scheme. If a low pass filter with power transfer function $f(\omega)$ is introduced between the signal source and the on-line computer, the measured aliased autospectrum will have an expectation :-

$$E\left\{Cxx_A(\omega_k)\right\} = f(\omega_k)\ \Gamma xx(\omega_k) + f(\omega_{N-k})\ \Gamma xx(\omega_{N-k})$$

The measured autospectrum can be compensated by dividing by $f(\omega_k)$. Under these circumstances the bias due to aliasing is given by :-

$$B\left\{Cxx_A(\omega_k)\right\} = \frac{f(\omega_{N-k})}{f(\omega_k)}\ \Gamma xx(\omega_{N-k}) \qquad \ldots\ldots\ldots\ldots 4.27.$$

Thus the bias error due to aliasing can be controlled to be less than ϵ. $\Gamma xx(\omega_{N-k})$ by choosing a filter whose power transfer function satisfies the relationship.

$$\frac{f(\omega_{N-k})}{f(\omega_k)} < \epsilon \qquad \text{when } \omega_k < \omega_r$$

The useful frequency range is 0 to ω_r, and ideally ω_r is as close to the folding frequency as is possible. Several filter functions can be used to control aliasing in this way, and a discussion of some suitable filters is given in the next chapter (section 5.3.3.).

4.3. Real-Time Digital Autospectrum Estimation Using an on-line Digital Computer

4.3.1. Real-Time Autospectrum Estimation by Averaging Raw Periodograms

In the previous section the poor statistical properties of the raw periodogram were mentioned. The raw periodogram is an inconvenient estimator of the autospectrum. Its variance is equal to the square of its expected value, and so does not decrease as the sample size decreases.

A way of overcoming this unfortunate property is to average n independent peridograms of the same signal. The variance of the smoothed periodogram calculated in this way would be $\frac{1}{n}$ times the variance of the raw periodogram. In section 4.2.4. it was shown that this method can be applied to a single record, by segmenting the data into n consecutive lengths and averaging the raw periodograms of the segments. It is now shown that this method can be applied to an on-line digital computer to obtain autospectral estimates in real-time. First the basic method is discussed, then a detailed flowchart is given.

The signal to be analysed is fed into the computer analogue/digital (A/D) interface. The programme, under control of the real-time clock, samples the signal at a uniform rate (the clock frequency f_c Hz) and stores the samples in an 'input array'. When a data segment of size N has been read into the 'input array', the data is transferred to a 'working array', and the input routine reset so that the next signal reading starts filling the 'input array' again. In the description that follows it is understood that when a new signal reading

is due, the real-time clock interrupt stops the computation flow temporarily to insert the new data point in the input array.

After transferring the data to the working array the programme sets about computing (rapidly) the periodogram of the data block. This is done using the fixed-point FFT described in chapter 3 and the routines for transforming real data (also from chapter 3), and conjugate multiplying the Fourier components. The peridogram of the data block is then added into an accumulator array, and the programme goes to an idling loop until the input array is again full when the process is repeated.

To show why this method was not viable before the discovery of the FFT, consider the timing requirements of the algorithm.

The maximum frequency component which can be studied is $fc/2$, i.e. half the maximum clock frequency. This is determined by the time, t(N), taken to compute the periodogram of an N point data block. The time taken to read in an N point data block is N/fc. Clearly this cannot be less than t(N), otherwise the computer will begin calculating the periodogram of the new data block before the periodogram of the previous block is stored. Therefore, N/fc must be greater than t(N) to allow the computer time to compute and store the periodogram of an N point data block, and return to the idling loop.

There is no limit to the lowest sampling frequency, so the useful frequency range which can be analysed using this on-line

technique is :-

$$0 \rightarrow \frac{N}{2.t(N)} \text{ Hz}$$

Using the traditional discrete Fourier transform and floating-point arithmetic from table 2.1. $t(1024) \approx$ 20 minutes, so the useful frequency range would be :

$$0 \rightarrow 0.42 \text{ Hz}$$

Using the FFT in fixed-point $t(1024) \approx 1$ second, so that the useful frequency range is :-

$$0 \rightarrow 500 \text{ Hz}$$

The FFT thus enables digital spectral analysis to be carried out on-line and in real-time over a wide band of sampling frequencies.

The important advantage of this system is the statistical stability of the spectral estimates. Provided that the signal is stationary the computer can continue averaging periodograms until the spectrum accumulator is full. The possible stability of the estimate depends upon the accumulator size. The programme described here can average 8192 periodograms without overflow, giving a possible variance improvement of $^1/8192$. This improvement would not be possible using off-line techniques, since it would require storage space in the computer for a data record of 8192.N points. Even for modest values of N the storage requirements would be very large.

To summarise, the method of averaging independent periodograms can be implemented in real-time on a digital computer. The FFT is used to calculate rapidly the periodogram of consecutive data blocks. The technique was not viable using the traditional DFT because its slowness imposed severe restrictions on the maximum sampling rate. The key advantage of the method is that (practically) arbitrary statistical stability can be attained if the signal is stationary, and if a sufficiently long record is available.

4.3.2. Description of the On-Line Autospectrum Estimation Programme

In this section a real-time on-line autospectrum estimation routine programmed on a GEC 90/2 on-line computer is described. This is done with the aid of the flowchart Fig. 4.1. appended to this chapter. The flow chart consists of three routines, the RTC routine, the analogue input routine, and the main routine.

Mobility between routines is made possible by the 'priority interrupt' system. The (Real Time Clock) routine has the highest priority and when the interrupts are enabled can interrupt any instruction box starred thus :- * . It does this at the clock frequency. The analogue input routine has the second priority and can interrupt any flow-chart block crossed thus :- + . It interrupts whenever the input A/D converter signals the completion of a conversion. This occurs a short time after the RTC routine initiates A/D conversion. After "servicing" an interrupt routine, the computer returns to the point in the programme at which it was interrupted.

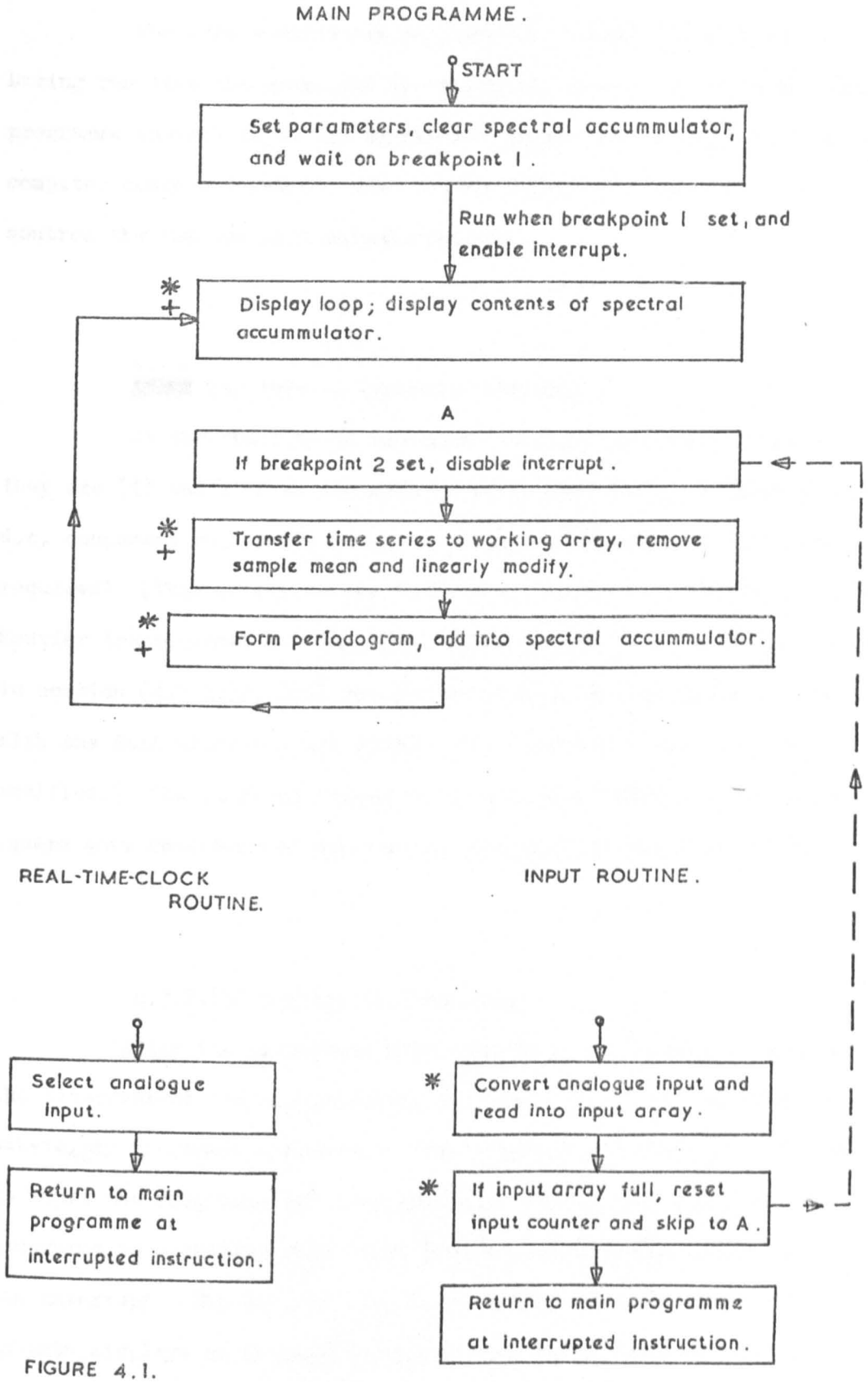


FIGURE 4.1.

SCHEMMATIC FLOW CHART FOR THE REAL TIME AUTOSPECTRUM ESTIMATION ROUTINE.

The main routine has no interrupt associated with it. During run-time the programme is controlled by the RTC routine. Ultimate programme control is in the operator's hands; the 'breakpoints' on the computer console determine when to start/stop the estimation, and control the various data output routines.

4.3.2

[redacted].(a) Initial Parameter Settings

At the start of an experiment various parameters must be set, they are (i) the size of the data block N, (ii) is suppression of the d.c. component required?, (iii) is linear modification of the data required? (This refers to the 'windowing' of the data blocks prior to Fourier transformation. The effects and purpose of this are discussed in section (4.2.6.)), (iv) the parameter p to be used.(This is connected with the data windowing and governs the fraction of the data that is modified.) The sampling frequency is set-up by attaching the output of a square wave generator of appropriate frequency to the external RTC input.

4.3.2.(b) Running the Programme

After the parameters have been set, the programme computes the bit-reversed table appropriate for that value of N, and sets the subsidiary programme parameters. The computer then enters a wait mode. To start the programme BPT 1 is depressed (set), this transfers the programme to a display loop which the RTC, and analogue input routines can interrupt. The display loop is a routine which,via the analogue outputs, displays on an oscilloscope screen the contents of the spectrum

accumulator. This is the equivalent of the idle loop mentioned in 4.3.1., and enables the operator to see the current estimate of the autospectrum while the computer is not calculating the periodogram of a data block.

When the Nth data point of a block has been read by the analogue input routine, the routine resets the input parameters. Instead of returning programme control to the appropriate point in the display loop, the routine diverts control to block A on the flowchart. This part of the programme transfers the newly completed data block to a working array. Here the following operations take place :-

(i) If suppression of the signal mean is required, the sample mean is calculated and subtracted from the working array

(ii) If linear modification is required the data is multiplied by the Generalised Hanning window function.

(iii) The discrete Fourier transform of the data is then obtained using the FFT and a routine for transforming real data.

(iv) The discrete Fourier coefficients are then conjugate multiplied and added into the spectral accumulator.

The programme then returns to the display loop and the updated spectrum appears on the oscilloscope screen.

4.3.2(c) Stopping the Programme and Subsequent Data Handling

When the operator decides that the displayed spectrum is sufficiently smooth, the programme can be stopped. This is done by depressing BPT 2, this causes the RTC interrupt to be disabled. The

programme now rests in the display loop.

The possibility arises that the operator wishes to obtain 'smoother' estimates from the existing estimates. This may be done by averaging adjacent spectral estimates, this increases the statistical stability of the estimates but also increases their bandwidth, so each estimate must be thought of as the average spectral power over an increased bandwidth.

Provision is made for additional smoothing using the Hanning filter weights $\frac{1}{4}$, $\frac{1}{2}$, $\frac{1}{4}$. Depressing BPT 4 convolves the accumulated spectrum with the weights $\frac{1}{4}$, $\frac{1}{2}$, $\frac{1}{4}$. By repeatedly depressing BPT 4 the spectrum can be smoothed as much as desired. Should the operator feel he has smoothed the spectrum too much, the original accumulated spectrum can be recalled.

Permanent records of the accumulated spectrum can be taken by :- (i) photographing the oscilloscope trace, (ii) using the on-line graph plotting facility. (This is triggered by set/resetting SES $\emptyset$.) (iii) punching the data on paper-tape. (This is triggered by set/reset BPT 3.)

The paper tape with the accumulated autospectrum punched on it is then used as data for an off-line "Algol" programme. This programme is coded for the University Elliot 4130 computer, and plots out the estimate of the autospectrum to either logarithmic or linear axes. The programme also prints out the autospectrum estimate.

4.4. Practical Aspects of Real-Time Digital Autospectrum Estimation

4.4.1. Accuracy

The accuracy of the digital autospectrum estimation programmes is determined by the size of the errors involved in collecting N data points and calculating the periodogram. The main error sources are; quantisation error at the analogue-to-digital converter, and round-off error in the calculation of the FFT. For the purpose of assessing the accuracy of the method, these errors can be lumped together, and replaced by an equivalent white noise source n(t). If the measurement noise is independent of the signal, the measured autospectrum of x(t) will have an expected value given by :-

$$\left\{ \Gamma_{xx}(\omega) \right\}_{\text{measured}} = \Gamma_{xx}(\omega) + \sigma_{nn}^2 \quad \ldots\ldots\ldots\ldots \quad 4.28.$$

Where σ_{nn}^2 is the variance of n(t) and $\Gamma_{xx}(\omega)$ is the autospectrum of x(t).

The variance of the error sources is now determined. Consider first the quantisation error: The GEC 90/2 computer used in this project has an eleven bit analogue-to-digital converter. The leading bit is used to indicate the sign of the signal, and the remaining ten bits represent the magnitude of the signal. The GEC 90/2 uses a twelve-bit word, and the output of the analogue-to-digital converter is inserted in the least significant end of the word. Therefore, assuming that the binary point is to the right of the most significant bit, the quantisation error in the analogue-to-digital conversion is uniformly distributed in the range -2^{-12} to $+2^{-12}$, and has a variance

of $\frac{1}{12} 2^{-24}$.

The round-off error in the fast Fourier transform was considered in chapter 3. From section 3.4. of chapter 3, the upper bound on the variance of the round-off error in transforming a 2^m point real data block is $C^2.2^m.2^{-22}$.

The constant C depends upon the type of array-scaling used in the fast Fourier transform. The FFT used in this project employed the modulus rescaling criterion with rescaling in the loop. From chapter 3, the value of C for this FFT is +0.41.

The upper bound on the variance of the measurement noise is therefore :-

$$\sigma_{nn}^2 = C^2 2^m 2^{-22} + \frac{1}{12} .2^{-24}$$

$$\approx C^2 2^m 2^{-22} \qquad \ldots\ldots\ldots\ldots 4.29.$$

Since the digital autospectrum estimates are scaled so that the maximum component lies between 1 and $\frac{1}{2}$, the measurement noise level can be conveniently expressed with respect to the magnitude of the maximum component of the measured autospectrum. In logarithmic terms the upper bound on the measurement noise level is approximately $20 \log_{10}(C\, 2^{(m/2)-11})$dB compared with the maximum autospectrum component.

The above noise figure represents an upper bound on the measurement noise, since it assumes that a rescaling occurs at each

stage of the FFT. If only s rescalings occur, then an approximate measurement noise level can be obtained as $20 \log_{10}(C\ 2^{(s/2)-11})$dB down on the maximum autospectrum component. The value of s may be read from the computer after an autospectrum measurement, and used to determine the approximate measurement noise level.

4.4.2. Limitation on Sampling Frequency and Frequency Resolution

In this section the maximum sampling, and the frequency resolution of the real-time autospectral estimation programmes are given. Firstly we consider the sampling frequency. For an N point block the maximum sampling frequency is determined by the time taken to estimate the periodogram of an N point real time series. Specifically, if it takes t seconds to calculate the periodogram of N data points, the maximum sampling frequency is $f_m = \frac{N}{t}$. The time taken to calculate the periodogram depends upon the type of fixed-point FFT used in the programme, and the ancillary data processing routines, (such as d.c. suppression and data windowing). The fixed point FFT which is used in the project uses the modulus rescaling test with rescaling in the loop. From chapter 3 this FFT can transform a 256-point real time series in 0.27 seconds. An extra 0.015 seconds are required for other computations, and so, assuming that no extra data processing is required, the maximum sampling frequency that can be used is :-

$$f_m = 900 \text{ Hz}$$

If the mean level is suppressed and linear modification is

used, the time required to compute the periodogram is increased to 0.32 seconds. Under these circumstances the maximum sampling frequency is reduced to :-

$$f_m = 800 \text{ Hz}$$

These maximum sampling frequencies can be increased at the expense of accuracy by using the least accurate fixed-point FFT (described in chapter 3). Using this transform the maximum sampling frequency can be increased to 1 KHz.

The frequency resolution is determined by the half-power bandwidth of the data window. In the on-line digital computer programmes the Generalised Hanning window is used. This window is described in section 4.2.6., and from that description the half power bandwidth of the Generalised Hanning window is given by :-

$$\omega_B \approx \frac{\omega_s}{N} (1 + 2p) \qquad \ldots\ldots\ldots\ldots 4.30.$$

Where, ω_s is the angular sampling frequency, and p is a parameter of the Generalised Hanning window. This parameter can theoretically take any value between 0 and $\frac{1}{2}$. However, in the programmes described here it is restricted to be a binary fraction.

As well as affecting the resolution of the Generalised Hanning window, p also controls the asymptotic decay rate. In auto-spectral analysis the decay rate of the Generalised Hanning spectral window is proportional to ω^{-2} for $\omega < \omega_p$ and proportional to ω^{-6} for $\omega > \omega_p$, where $\omega_p = \frac{\omega_s}{Np}$.

Therefore the parameter p should be set at run-time to control the asymptotic decay rate of the Generalised Hanning window. The appropriate data block size is then found by substituting ω_s, ω_B and p in equation 4.30. It is not possible to select any specific data block size because of two restrictions upon N. The first restriction is set by the size of the FFT trignometrical 'look-up' tables used. In the programmes described here the maximum value of N is 256. This limit was set arbitrarily at the time of assembling the autospectrum estimation programmes, and can be easily increased by extending the 'look-up' table. The ultimate limit upon N is set by the size of the computer store. For the GEC 90/2 with an 8K memory it should be possible to have N = 1024.

The second restriction upon N is due to the FFT algorithm which restricts the programme to processing data blocks of size 2^m (m integer). Therefore, the maximum resolution of the real-time digital autospectrum analysis programmes is $\frac{\omega_s}{256}(1 + p)$. This can be doubled by selecting N = 128, and so on.

4.4.3. Restrictions Upon the Useful Frequency Range Set by Aliasing

Aliasing has been shown in section 4.2.7. to introduce bias errors in autospectrum estimation. It was also shown that these errors can be controlled by low-pass filtering the data before measuring the autospectrum. If the low-pass filter has a power transfer function $f(\omega)$, then the bias due to aliasing at frequency ω_k is given by equation 4.27. In chapter 5, some suitable low-pass

filter functions are discussed. The simplest and the most readily available filter function treated in this chapter has the Maximally Flat, Butterworth response. It is shown that using a fourth order Butterworth low-pass filter with cut-off frequency $\omega_c = 0.3\omega_s$, the alias distortion in the frequency range $\omega = 0$ to $0.4\omega_s$ is less than $0.04\ \Gamma_{xx}(\omega_s - \omega)$.

Therefore in practical autospectral measurements if ω_1, is the maximum frequency of interest and the sampling frequency is set at $2.5\omega_1$, then a fourth order Butterworth guard filter ($\omega_c = 0.75\omega_1$) is sufficient to ensure that the bias due to aliasing is less than $0.04\ \Gamma_{xx}(\omega_s - \omega_1)$ in the frequency range $\omega = 0$ to ω_1.

REFERENCES

4.1. Bingham, C., "Modern Techniques of Power Spectrum Estimation ", IEEE Trans., Vol. AU-15, No. 2, June, 1967.

4.2. Jenkins, G.M. and Watts, D.G., "Spectral Analysis and its Application ", Hoden-Day 1968, pp. 247-248

4.3. Bartlett, M.S., "An Introduction to Stochastic Processes", Cambridge University Press, 1966, pp 304-331

4.4. Welch, P.D., "The Use of Fast Fourier Transform for the Estimation of Power Spectra", IEEE Trans, Vol. AU-15, No. 2, June 1967.

4.5. Forsythe, G.E., "Generation and Use of Orthogonal Polynominals for Data-Fitting with a Digital Computer", J. Soc. Ind. Appl. Math., Vol. 5., No. 2, June 1957.

5. FREQUENCY RESPONSE ESTIMATION

5.1. Introduction

This chapter deals with some aspects of frequency response estimation from noise-like data. Section 5.2. describes the properties of a continuous frequency response estimator obtained from finite joint records of a system input and output signals. In section 5.3. the corresponding discrete estimators are presented. Also in this section the problem of suppressing aliasing in discrete frequency response estimation is treated.

Section 5.4. describes the application of these discrete estimators to on-line computers. In particular it is shown that, using the FFT, digital frequency response estimates over a wide frequency range can be obtained in real-time. A real-time frequency response estimation programme is described, and in section 5.5. its accuracy, resolution, and frequency range are discussed.

List of Symbols for Chapter 5

A_k	k^{th} coefficient of the discrete Fourier transform of $x_i l_i$
B_k	" " " " " " " " $y_i l_i$
$C_{xx}(\omega)$	Sample autospectrum of $x(t)$
$\bar{C}_{xx}(\omega)$	smoothed estimate of the autospectrum of $x(t)$
$\bar{C}_{xx}(\omega_k)$	smoothed discrete estimate of the autospectrum of $x(t)$
$\bar{C}_{xx_A}(\omega)$	aliased, smoothed estimate of the autospectrum of $x(t)$
$C_{xy}(\omega)$	sample cross-spectrum of $x(t)$ leading $y(t)$. The modifications to denote, aliased, discrete, and smoothed autospectrum estimates also apply to the cross-spectral estimate.
$f(\omega)$	power transfer function of a filter
$H(\omega)$	the frequency response function of a system
$\hat{H}_A(\omega)$	aliased estimate of $H(\omega)$
$K^2_{xy}(\omega)$	the squared coherency between $x(t)$ and $y(t)$
$\hat{K}^2_{xy_A}(\omega)$	the aliased estimate of $K^2_{xy}(\omega)$
$l(t)$	a data window function
l_i	a discrete data window function
n	number of sample spectra averaged to obtain a smoothed estimate
$n(t)$	a gaussian, zero mean, noise source
P	a probability
$\bar{P}_{xy}(\omega)$	smoothed estimate of the co-spectrum of $x(t)$ leading $y(t)$
$\bar{Q}_{xy}(\omega)$	" " " " quad-spectrum" " " "
$R(\omega_k)$	the power transfer ratio of a filter $\frac{f(\omega_{N-k})}{f(\omega_k)}$
$R_{xx}(\omega_k)$	the autospectral ratio $\frac{\Gamma_{xx}(\omega_{N-k})}{\Gamma_{xx}(\omega_k)}$
T	the duration of a data sample

x_i	a time series obtained by sampling x(t)
x(t)	input signal to a system
$X(\omega)$	the Fourier transform of x(t)1(t)
$X_A(\omega)$	the aliased Fourier transform of x(t)1(t)
y	a real variable
y_i	a time series obtained by sampling y(t)
y(t)	output signal of a system
$Y(\omega)$	the Fourier transform of y(t)1(t)
$Y_A(\omega)$	the aliased Fourier transform of y(t)1(t)
z	a real variable
$\Gamma_{xx}(\omega)$	autospectrum of x(t)
$\Gamma_{xy}(\omega)$	cross-spectrum of x(t) leading y(t)
Δ	a sample period
Δ_H	confidence band associated with $H(\omega)$
$\Delta\phi$	" " " " $\phi(\omega)$
$\Lambda_{xy}(\omega)$	co-spectrum of x(t) leading y(t)
σ^2	a variance
$\phi(\omega)$	the argument of $H(\omega)$
$\hat{\phi}_A(\omega)$	the aliased estimate of $\phi(\omega)$
$\Psi_{xy}(\omega)$	the quad-spectrum of x(t) leading y(t)
ω	angular frequency
ω_c	the angular cutoff frequency of a filter
ω_k	the discrete angular frequency $\frac{2\pi k}{\Delta N}$
ω_s	the angular sampling frequency
q	the integral over time of $\ell^2(t)$ (discrete or continuous)

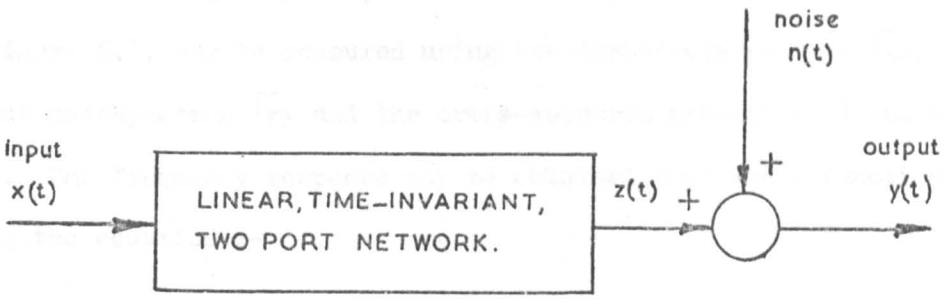


FIGURE 5.1. LINEAR SYSTEM WITH OUTPUT CORRUPTED BY NOISE.

5.2. Statistical Properties of Open Loop Frequency Response Estimators

5.2.1. Introduction

In this section we consider the properties of frequency response estimators associated with open loop system identification. The type of system considered is the two-port, linear, time-invariant network shown in fig. 5.1., whose output is corrupted by noise. The input, $x(t)$ and output $y(t)$ of the system are assumed to be normally distributed random variables, with zero mean levels. The corrupting noise $n(t)$ is also assumed to be a normally distributed random variable, and is uncorrelated with the input signal.

5.2.2. The Estimators

The frequency response function, $H(\omega)$, of the system shown in figure 5.1. may be measured using the input autospectrum Γ_{xx}, the output autospectrum Γ_{yy} and the cross-spectrum between $x(t)$ and $y(t)$, Γ_{xy}. The frequency response may be obtained from these functions using the equation :-

$$H(\omega) = \frac{\Gamma_{xy}(\omega)}{\Gamma_{xx}(\omega)} \qquad \ldots\ldots\ldots\ldots \quad 5.1.$$

Where $\Gamma_{xy}(\omega) = \Lambda_{xy}(\omega) + j\Psi_{xy}(\omega)$

$\Lambda_{xy}(\omega)$ is the real part of $\Gamma_{xy}(\omega)$ and is termed the co-spectrum of $x(t)$ leading $y(t)$

$\Psi_{xy}(\omega)$ is the imaginary part of $\Gamma_{xy}(\omega)$ and is termed the quad-spectrum of $x(t)$ leading $y(t)$

In practice $H(\omega)$ must be measured using estimates of the auto and cross-spectra. When assessing the quality of these estimates it is useful to have a frequency-domain measure of the correlation between $x(t)$ and $y(t)$. This measure is provided by the squared coherency function, defined by :-

$$K^2_{xy}(\omega) = \frac{|\Gamma_{xy}(\omega)|^2}{\Gamma_{xx}(\omega)\,\Gamma_{yy}(\omega)} \qquad \ldots\ldots\ldots\ldots \quad 5.2.$$

The squared coherency function can be rewritten in terms of the input autospectrum and the noise autospectrum. Thus :-

$$K^2_{xy}(\omega) = \frac{1}{1 + \dfrac{\Gamma_{nn}(\omega)}{|H(\omega)|^2 \Gamma_{xx}(\omega)}}$$

$$= \frac{1}{1 + \dfrac{\Gamma_{nn}(\omega)}{\Gamma_{zz}(\omega)}} \qquad \ldots\ldots\ldots 5.3.$$

Where, from figure 5.1., $\Gamma_{zz}(\omega)$ is the autospectrum of the noise-free system output.

From equation 5.3. it is seen that the squared coherency is bounded by 0 and 1. The actual value that the squared coherency takes is governed by ratio of the noise-free output spectrum to the noise spectrum, $\Gamma_{zz}(\omega)/\ \Gamma_{nn}(\omega)$. When the squared coherency is unity, the noise spectrum is zero, and the system output is entirely composed of components due to the input. Conversely, when the squared coherency is zero, the output signal to noise spectrum ratio is zero, and the system output is composed only of the noise n(t). Thus, the squared coherency gives a measure of the correlation between the input and output signals. It is therefore useful in assessing the quality of a frequency response estimate.

Estimators of the frequency response and squared coherency based upon finite samples of x(t) and y(t) are now defined. Let x(t), y(t), $(0 < t < nT)$ be jointly recorded samples of the input and output signals of $H(\omega)$. These samples are each sub-divided into n segments of duration T seconds. The i^{th} segments are defined as :-

$$x^{(i)}(t) = x(t + iT - \frac{T}{2})$$

$$-\frac{T}{2} < t < \frac{T}{2} \qquad \ldots\ldots\ldots 5.4.$$

$$y^{(i)}(t) = y(t + iT - \frac{T}{2})$$

The i^{th} sample auto, and cross-spectra are obtained using the expressions :-

$$C^{(i)}_{xx}(\omega) = \frac{1}{q}\left|\int_{-\infty}^{+\infty} x^{(i)}(t)\, 1(t) e^{-j\omega t}\, dt\right|^2$$

$$C^{(i)}_{yy}(\omega) = \frac{1}{q}\left|\int_{-\infty}^{+\infty} y^{(i)}(t)\, 1(t) e^{-j\omega t}\, dt\right|^2 \qquad \ldots\ldots\ldots 5.5.$$

$$C^{(i)}_{xy}(\omega) = \frac{1}{q}\int_{-\infty}^{+\infty} x^{(i)}(t)\, 1(t) e^{+j\omega t}\, dt. \int_{-\infty}^{+\infty} y^{(i)}(t)\, 1(t) e^{-j\omega t}\, dt$$

Where $1(t)$ is a window function drawn from the class of functions defined by equation 4.2.

The smoothed sample auto and cross-spectra based upon the samples $x(t)$ and $y(t)$ are defined by :-

$$\bar{C}xx(\omega) = \frac{1}{n} \sum_{i=1}^{n} Cxx^{(i)}(\omega)$$

$$\bar{C}yy(\omega) = \frac{1}{n} \sum_{i=1}^{n} Cyy^{(i)}(\omega) \qquad \text{..... 5.6.}$$

$$\bar{C}xy(\omega) = \frac{1}{n} \sum_{i=1}^{n} Cxy^{(i)}(\omega)$$

The smoothed cross-spectral estimator $\bar{C}xy(\omega)$ is complex, and is defined as :-

$$\bar{C}xy(\omega) = \bar{P}xy(\omega) + j\ \bar{Q}xy(\omega)$$

Where $\bar{P}xy(\omega)$ is the real part of $\bar{C}xy(\omega)$, and is the smoothed sample co-spectrum of $x(t)$ leading $y(t)$; and $\bar{Q}xy(\omega)$ is the imaginary part of $\bar{C}xy(\omega)$, and is the smoothed sample quad-spectrum of $x(t)$ leading $y(t)$.

Note that in our definition of cross-spectrum we have used the 'plus' convention, this convention is adhered to throughout this thesis. The alternative convention is to define the cross-spectrum estimate as :-

$$\bar{C}xy(\omega) = \bar{P}xy(\omega) - j\ \bar{Q}xy(\omega)$$

And thence define $\bar{Q}xy(\omega)$ as the sine Fourier transform of the odd part of the cross-covariance estimate, (see for example ref. 5.1.).

The frequency response and coherency estimators associated with these auto-and cross-spectrum estimators are :-

$$\hat{H}(\omega) = \frac{\bar{C}xy(\omega)}{\bar{C}xx(\omega)} \qquad \ldots\ldots\ldots\ldots 5.7.$$

$$\hat{K}^2_{xy}(\omega) = \frac{|\bar{C}xy(\omega)|^2}{\bar{C}xx(\omega)\ \bar{C}yy(\omega)} \qquad \ldots\ldots\ldots\ldots 5.8.$$

The frequency response estimate is usually expressed in terms of its modulus $|\hat{H}(\omega)|$ and argument $\hat{\phi}(\omega)$; thus :-

$$\hat{H}(\omega) = |\hat{H}(\omega)| \exp(j\ \hat{\phi}(\omega))$$

The modulus and argument of $H(\omega)$ are termed the 'gain' and 'phase' respectively, and they are defined in terms of the auto- and cross-spectrum estimators as :-

$$|\hat{H}(\omega)| = \frac{\left\{\bar{P}^2xy(\omega) + \bar{Q}^2xy(\omega)\right\}^{\frac{1}{2}}}{\bar{C}xx(\omega)}$$

$$\hat{\phi}(\omega) = \tan^{-1}\left\{\frac{\bar{Q}xy(\omega)}{\bar{P}xy(\omega)}\right\}$$

5.2.3. Variance and Bias of the Estimators

Approximate expressions for the variance of the estimated frequency response, and squared coherency may be obtained by expanding

the estimators in a Taylor series expansion about their expected values as indicated in appendix 4. Using this technique the variance of $\hat{H}(\omega)$ and $\hat{K}xy(\omega)$ can be expressed approximately in terms of the covariance matrix of the smoothed spectral estimators. Using the results of appendix 2 the covariance matrix of the smoothed estimators of equation 5.6. can be obtained as :-

$$\mathrm{Cov}\begin{bmatrix}\bar{C}xx\\ \bar{C}yy\\ \bar{P}xy\\ \bar{Q}xy\end{bmatrix} \simeq \frac{1}{n}\begin{bmatrix} \Gamma_{xx}^2 & |\Gamma_{xy}|^2 & \frac{\Gamma_{xx}}{2}(\Gamma_{xy}+\Gamma_{yx}) & \frac{\Gamma_{xx}}{j2}(\Gamma_{xy}-\Gamma_{yx}) \\ & \Gamma_{yy}^2 & \frac{\Gamma_{yy}}{2}(\Gamma_{xy}+\Gamma_{yx}) & \frac{\Gamma_{yy}}{2j}(\Gamma_{xy}-\Gamma_{yx}) \\ & & \frac{1}{4}(\Gamma_{xy}^2+\Gamma_{yx}^2+2\Gamma_{xx}\Gamma_{yy}) & \frac{1}{4j}(\Gamma_{xy}^2-\Gamma_{yx}^2) \\ & S & & \frac{1}{4}(2\Gamma_{xx}\Gamma_{yy}-\Gamma_{xy}^2-\Gamma_{yx}^2) \end{bmatrix} \quad \ldots\ldots\ldots\ldots 5.9.$$

In this matrix, S denotes diagonal symmetry. The bias error due to windowing has been neglected, as has the dependence of the auto- and cross-spectra upon ω. This practice is maintained throughout the remainder of the chapter, and the reader is asked to take the dependence of spectral functions upon ω as being implicit. Using the elements of matrix 5.9., the following approximate variance expressions are obtained :-

$$\frac{\mathrm{Var}\left\{|\hat{H}|\right\}}{|H|^2} = \frac{1}{2n}\left\{\frac{1}{K_{xy}^2} - 1\right\}$$

$$\mathrm{Var}\left\{\hat{\phi}\right\} = \frac{1}{2n}\left\{\frac{1}{K_{xy}^2} - 1\right\} \qquad \ldots\ldots\ldots\ldots 5.10.$$

$$\frac{\mathrm{Var}\left\{\hat{K}_{xy}^2\right\}}{K_{xy}^2} = \frac{2}{n}\left\{K_{xy}^2\,(1 - K_{xy}^2)^2\right\}$$

These expressions show that the variability of all estimators depend upon the coherency between the input and output. When the squared coherency is unity, the variances are zero, and (neglecting bias errors) the parameters can be measured exactly. Conversely, as the coherency decreases, the variance of the gain, phase and coherency estimators is increased. The effect of smoothing is shown, since the variance expressions are all inversely proportional to n, the number of sample spectra that are averaged.

Approximate expressions for bias due to windowing may be obtained by expressing the bias of the estimators in terms of the bias in the auto, and cross-spectral estimators, (as shown in appendix 4). Using this approach, and the results of section 4.2.2. on bias errors, the approximate bias expressions for gain, phase and squared coherency are :-

$$B\left\{|\hat{H}|\right\} \doteq \frac{-a_2}{T^2}\left\{ |\ddot{H}| - |H|\,(\dot{\phi})^2 \right\}$$

$$B\left\{\hat{\phi}\right\} \doteq \frac{-a_2}{T^2}\left\{ \ddot{\phi} + \frac{2\dot{\phi}\,|\dot{H}|}{|H|} \right\} \qquad \text{..... 5.11.}$$

$$B\left\{\hat{K}^2_{xy}\right\} \doteq \frac{1}{n}(1 - K^2_{xy}) + \frac{a_2}{T^2}\left\{ \ddot{K}_{xy} K_{xy} - \left\{\dot{\phi} K_{xy}\right\}^2 \right\}$$

Where a_2 is the second term in the power series expansion of $\frac{1}{q}\left\{1(t) * 1(t)\right\}$, T is the sample duration, and the dot notation is used to indicate the first and second derivatives with respect to angular frequency. In the on-line digital computer routines for frequency response estimation the linear data window used to modify the records is the Generalised Hanning window. From TABLE 4.1., the value of a_2 for this window is approximately -6.

The following relevant points can be made concerning these bias expressions.

i) The bias due to finite record lengths is inversely proportional to the square of the sample record length, T. In discrete estimation expressions this is replaced by N the number of data points per periodogram, and the derivatives become appropriate finite differences.

ii) The bias in the gain and coherency estimators contain terms proportional to $\dot{\phi}^2$, the squared first derivative of phase. This implies that if there is a large time delay T′ between x(t) and y(t), then the bias in gain and coherency will be proportional to $\left\{\frac{T'}{T}\right\}^2$. If T′ is of the same order of magnitude as the sample record length, then significant bias errors may occur. In the estimation procedure used

in the on-line computer programmes, it is unlikely that such large delays would be encountered. However, should such delays occur, they can be compensated for by entering new readings of x(t) into an S stage shift register, and storing the shift register output in the input array referred to in the programme description (section 5.4.2.). This compensates for an S.$\triangle$ sec. time delay between x(t) and y(t), where $\triangle$ is the sampling period.

The problem of time delays has been discussed by Akaike (ref. 5.2.) in connection with the 'Lagged-Products' method of spectral analysis. This method employs a data record of fixed duration, which we may call T_R seconds, to obtain spectral estimates. To reduce the variability of the estimates they are effectively, smoothed by averaging n sample estimates obtained from n consecutive segments of the record. Each segment is of duration T seconds, where $T = \frac{T_R}{n}$, and typically T might lie between $\frac{1}{10}^{th}$ and $\frac{1}{100}^{th}$ of T_R. Therefore time delays which are insignificant compared with T_R, may be of comparable magnitude with T, thus causing severe bias errors (ref. 5.3.). This situation could not arise in the on-line computer routines (section 5.4.) because the effective sample duration T is specified initially, and smoothing is accomplished by increasing T_R.

iii) The bias is the squared coherency estimate contains a term inversely proportional to n. The Taylor series approximation to the bias suggests that the bias due to insufficient smoothing is equal to $\frac{1}{n}$. However, experimental studies indicate that the smoothing bias is given by $\frac{1}{n}(1 - K^2_{xy})$ as shown in the bias equation. This result corresponds with the empirical findings of Benignus (ref. 5.4.).

5.2.4. Confidence Statements

In order to make confidence statements about the frequency response functions, we must first find appropriate sampling distributions for the estimators. A comprehensive study of the sampling distributions associated with frequency response estimation has been made by Goodman (ref. 5.5.). Goodman obtained his results by extending the theory of statistical analysis of multivariate real-time series to the analysis of bi-variate complex series. The discrete Fourier transforms of the real-time series x(i) and y(i), which we denote as $X(\omega_k)$ and $Y(\omega_k)$, form a bi-variate set of complex series. The joint distribution of these series can be obtained and used to formulate the distribution of functions associated with the frequency response estimator.

Specifically, Goodman assumes that the complex random variables $X_i(\omega_k)$, $Y_i(\omega_k)$, $(i = 1, \ldots\ldots n)$ form a set of independent estimates of the Fourier transform of the input x(t) and y(t) at a particular frequency ω_k. Then he determines distributions associated with the frequency response estimator $\hat{H}(\omega)$ defined by :-

$$\hat{H}(\omega_k) = \frac{\sum_{i=1}^{n} X_i^*(\omega_k)\, Y_i(\omega_k)}{\sum_{i=1}^{n} X_i^*(\omega_k)\, X_i(\omega_k)} \qquad \ldots\ldots\ldots\ldots \text{5.12.}$$

This is exactly the form of estimator that we use when estimating the frequency response function. Hence, the results of Goodman apply directly to the estimators obtained in this chapter.

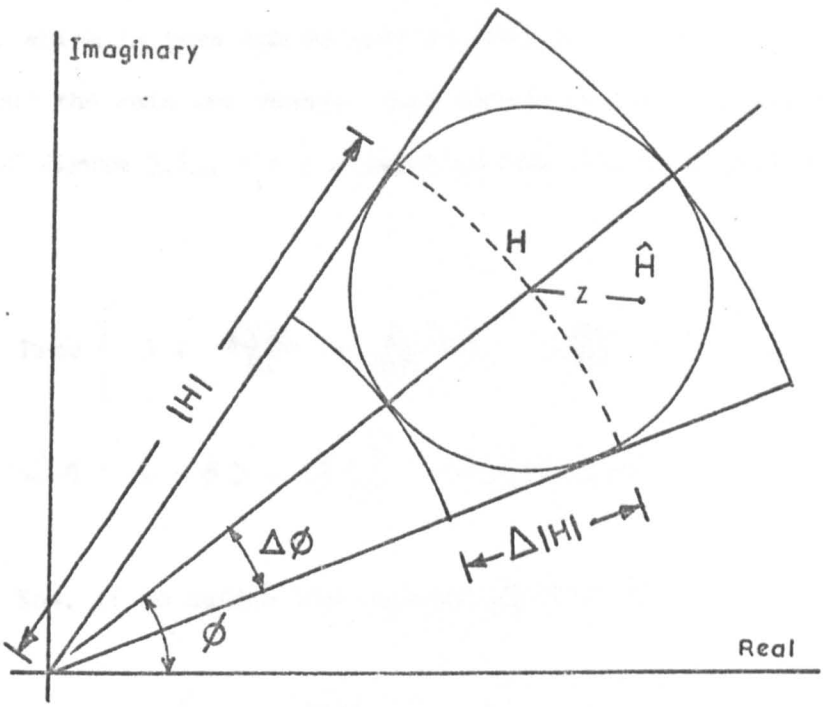


FIGURE 5.2. FREQUENCY RESPONSE ESTIMATE CONFIDENCE REGIONS.

5.2.4(a) Confidence Statements for Gain and Phase

Goodman has shown that the density function of the random variable z associated with the modulus of the complex error $\hat{H} - H$ is given by :-

$$f_Y(y) = \frac{2n(1 - K_{xy}^2)^n y}{\left\{(1 - K_{xy}^2)+y^2\right\}^{n+1}} \qquad \ldots\ldots\ldots\ldots 5.13.$$

Where in this context $y = z.\frac{K_{xy}}{|H|}$, and z is the random variable associated with the modulus of the complex error $\hat{H} - H$.

The cumulative distribution of y is given by :-

$$F_Y(y_1) = 1 - \left\{1 + \frac{y_1^2}{(1-K_{xy}^2)}\right\}^{-n} \qquad \ldots\ldots\ldots\ldots 5.14.$$

This can be used to make a probability statement about $|\hat{H} - H|$, which in turn can be used to make joint probability statements about the gain and phase. From reference 5.4., and using the symbols of figure 5.2., the following probability statement may be made :-

$$\text{Prob}\left\{ 1 + \frac{\Delta|H|}{|H|} > \frac{|\hat{H}|}{|H|} > 1 - \frac{\Delta|H|}{|H|} \right. \quad \ldots\ldots\ldots 5.15.$$

and $$\left. \Delta\phi > \hat{\phi} - \phi > - \Delta\phi \right\} > \text{Prob}\left\{ |z| < \Delta|H| \right\}$$

Now, if we define the probability P as :-

$$P = \text{Prob}\left\{ |z| < \Delta|H| \right\}$$

Then equation 5.14. can be used to show that :-

$$\frac{\Delta|H|}{|H|} = \left\{ \left\{ (1-P)^{-\frac{1}{n}} - 1 \right\} \left\{ \frac{1}{K_{xy}^2} - 1 \right\} \right\}^{\frac{1}{2}} \quad \ldots\ldots\ldots 5.16.$$

$$\Delta\phi = \sin^{-1}\left\{ \frac{\Delta|H|}{|H|} \right\}$$

From expression 5.15. the approximate simultaneous 100 P% confidence intervals for gain H, and phase ϕ are :-

$$|\hat{H}|\left\{ 1 \pm \Delta H \right\} \quad ; \quad \hat{\phi}\left\{ 1 \pm \Delta\phi \right\}$$

Jenkins and Akaike (refs. 5.6., and 5.7.) have also studied the sampling distribution of frequency response estimators, and have shown that the function $C_{xx}\,|\hat{H} - H|^2.(n-1)(C_{nn})^{-1}$ is approximately Fisher distributed with degrees of freedom 2, and 2n-2 respectively.

Thus confidence statements concerning $|\hat{H}|$ and $\hat{\phi}$ can be made using the widely tabulated Fisher distribution.

However, a transformation of the Fisher distribution with degrees of freedom 2, and 2n-2 shows that the distribution of $|\hat{H} - H|$ is the same as Goodman's distribution. The methods are thus equivalent. Use of Goodman's method is advocated since it is simpler than that of Jenkins and Akaike.

5.2.4.(b) Confidence Statements for Squared Coherency

The sampling distribution associated with the squared coherency has been determined by Goodman (ref. 5.5.), and tabulated percentage points are available for $n < 20$ (ref. 5.8.). For larger values of n the cumulative distribution function can be computed. However, as n increases the time required to calculate the cumulative distribution rapidly becomes excessive. Goodman and Enochson (ref. 5.9.) have investigated the sampling distribution of the coherency function, and they show that the lengthy computations associated with the exact distribution can be avoided by using a gaussian approximation to the distribution. This approximation involves transforming the coherency, using the transformation :-

$$z = \frac{1}{2} \ln \left\{ \frac{1 + Kxy}{1 - Kxy} \right\} \qquad \ldots\ldots\ldots\ldots \quad 5.17.$$

Where, for $n > 20$ the transformed variable, z, has, to a close approximation, a normal distribution. The variance and mean of

this distribution are :-

$$\mu = \frac{1}{2} \ln \left\{ \frac{1 + E[Kxy]}{1 - E[Kxy]} \right\} \qquad \ldots\ldots\ldots\ldots 5.18.$$

$$\sigma^2 = \frac{1 - (0.004)^{1.6K_{xy}^2 + 0.22}}{2(n-1)}$$

The variance expression is that quoted by Benigus (ref. 5.4.). In his extensive study of the squared coherency estimator, he shows that using the expressions 5.18. the normal approximation to the distribution of squared coherency holds for the entire range of possible coherency values.

5.3. Discrete Estimation of Frequency Response Functions

5.3.1. Discrete Estimation Formulae

In this section the discrete estimation formulae that are used in the real-time, digital, frequency response estimation programmes are given. The discrete estimators are obtained from the continuous estimators of section 5.2. using the analogy of section 4.2.7.

Let x_i and y_i be N-point real time series obtained by sampling $x(t)$ and $y(t)$ every Δ seconds. The discrete Fourier transforms of the time series (modified by l_i, a window function drawn from the class of functions defined by equation 4.3.) are :-

$$A_k = \frac{1}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}-1} x_i l_i W_N^{ik}$$

$$B_k = \frac{1}{N} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}-1} y_i l_i W_N^{ik} \qquad \text{..... 5.19.}$$

Where $W_N = \exp(-j\frac{2\pi}{N})$

The raw periodograms of the input and output are given by :-

$$C_{xx}(\omega_k) = \frac{\Delta N}{q}\left\{ |A_k|^2 \right\}$$

$$C_{yy}(\omega_k) = \frac{\Delta N}{q}\left\{ |B_k|^2 \right\} \qquad \ldots\ldots\ldots\ldots 5.20.$$

$$C_{xy}(\omega_k) = \frac{\Delta N}{q}\left\{ A_k^* B_k \right\}$$

Where $\omega_k = \frac{2\pi k}{\Delta N}$, $k = 0, \ldots\ldots \frac{N}{2}$

Smoothed discrete estimates are obtained by recording an nN point time series of x(t) and y(t), segmenting each time series into n time series, of N points each, and averaging the periodograms obtained from the segments. The smoothed periodograms obtained in this way are defined by :-

$$\bar{C}_{xx}(\omega_k) = \frac{1}{n}\sum_{i=1}^{n} C_{xx}^{(i)}(\omega_k)$$

$$\bar{C}_{yy}(\omega_k) = \frac{1}{n}\sum_{i=1}^{n} C_{yy}^{(i)}(\omega_k) \qquad \ldots\ldots\ldots\ldots 5.21.$$

$$\bar{C}_{xy}(\omega_k) = \frac{1}{n}\sum_{i=1}^{n} C_{xy}^{(i)}(\omega_k)$$

Where $C_{xx}^{(i)}$, $C_{yy}^{(i)}$, and $C_{xy}^{(i)}$ are the raw periodograms obtained from the i^{th} segment of the nN point series.

The discrete frequency response estimate and squared coherency

estimate obtained from these smoothed periodograms are given by :-

$$\hat{H}(\omega_k) = \frac{\bar{C}xy(\omega_k)}{\bar{C}xx(\omega_k)} \qquad \ldots\ldots\ldots\ldots \text{5.22.}$$

$$\hat{K}^2_{xy}(\omega_k) = \frac{|\bar{C}xy(\omega_k)|^2}{\bar{C}xx(\omega_k)\bar{C}yy(\omega_k)}$$

Where $\omega_k = \frac{2\pi k}{\Delta N}$, $k = 0, \ldots\ldots \frac{N}{2}$

5.3.2. Aliasing

The discrete estimation formulae given above are discrete aliased versions of the continuous estimators given in section 5.2. The effect of aliasing is to distort the discrete estimates so that they are no longer faithful versions of the continuous functions. In this section we consider the nature of this distortion as it affects the expected value and variance of aliased frequency response estimators.

If the continuous Fourier transforms of $x(t)\,1(t)$ and $y(t)1(t)$ are $X(\omega)$ and $Y(\omega)$ respectively, then the discrete Fourier transforms of $x_i 1_i$ and $y_i 1_i$ are related to the continuous transforms by :-

$$A_k = \frac{1}{\Delta N} \sum_{n=-\infty}^{+\infty} X(\omega_{k-nN})$$

$$\ldots\ldots\ldots\ldots \text{5.23.}$$

$$B_k = \frac{1}{\Delta N} \sum_{n=-\infty}^{+\infty} Y(\omega_{k-nN})$$

Where $\omega_k = \frac{2\pi k}{\Delta N}$

Now, if the spectra of x(t) and y(t) are low-pass, then we are justified in assuming that all but the first alias can be neglected. Under this assumption the aliased spectra of l(t)x(t) and l(t)y(t) can be defined as :-

$$X_A(\omega_k) = \Delta N A_k \triangleq X(\omega_k) + X^*(\omega_{N-k}) \quad \ldots\ldots\ldots\ldots 5.24$$

$$Y_A(\omega_k) = \Delta N B_k \triangleq Y(\omega_k) + Y^*(\omega_{N-k})$$

The aliased auto, and cross-spectral estimators obtained from these aliased spectra are :-

$$Cxx_A(\omega_k) = \frac{1}{q} \left| X_A(\omega_k) \right|^2$$

$$Cyy_A(\omega_k) = \frac{1}{q} \left| Y_A(\omega_k) \right|^2 \quad \ldots\ldots\ldots\ldots 5.25$$

$$Cxy_A(\omega_k) = \frac{1}{q} X_A^*(\omega_k)\, Y_A(\omega_k)$$

5.3.2.(a) Bias

From appendix 3, the expected values of the auto- and cross-spectral estimators, (neglecting the effect of windowing) are given by:-

$$E\left\{Cxx_A(\omega_k)\right\} = \Gamma xx(\omega_k) + \Gamma xx(\omega_{N-k})$$

$$E\left\{Cyy_A(\omega_k)\right\} = \Gamma yy(\omega_k) + \Gamma yy(\omega_{N-k}) \qquad \ldots\ldots\ldots\ldots 5.26$$

$$E\left\{Cxy_A(\omega_k)\right\} = \Gamma xy(\omega_k) + \Gamma yx(\omega_{N-k})$$

Using these expressions approximate formulae for the bias error in aliased frequency response and squared coherency estimators can be obtained. These formulae are :-

$$\frac{B\left\{\left|\hat{H}_A(\omega_k)\right|\right\}}{\left|H_A(\omega_k)\right|} \simeq \frac{\Gamma xx(\omega_{N-k})}{\Gamma xx(\omega_k)} \left\{ Re\left\{ \frac{H^*(\omega_{N-k})}{H(\omega_k)} - 1 \right\}\right\}$$

$$B\left\{ \hat{\phi}_A(\omega_k)\right\} \simeq \frac{\Gamma xx(\omega_{N-k})}{\Gamma xx(\omega_k)} \left\{ Im\left\{ \frac{H^*(\omega_{N-k})}{H(\omega_k)} \right\}\right\} \qquad \ldots\ldots\ldots 5.27$$

$$\frac{B\left\{\hat{K}^2_{xy_A}(\omega_k)\right\}}{K^2_{xy_A}(\omega_k)} \simeq \frac{\Gamma xx(\omega_{N-k})}{\Gamma xx(\omega_k)} \left\{ 2Re\left\{ \frac{H^*(\omega_{N-k})}{H(\omega_k)} - 1 \right\}\right\} - \frac{\Gamma yy(\omega_{N-k})}{\Gamma yy(\omega_k)}$$

5.3.2.(b) Variance

From appendix 3, approximate expressions for the variance of the aliased, smoothed, auto and cross-spectral estimators are :-

$$Var\left\{\bar{C}xx_A(\omega_k)\right\} \simeq \frac{1}{n} E^2\left\{\bar{C}xx_A(\omega_k)\right\}$$

$$Var\left\{\bar{C}yy_A(\omega_k)\right\} \simeq \frac{1}{n} E^2\left\{\bar{C}yy_A(\omega_k)\right\} \qquad \ldots\ldots\ldots\ldots 5.28$$

$$Var\left\{\bar{C}xy_A(\omega_k)\right\} \simeq \frac{1}{n}\left\{E^2\left\{\bar{C}xy_A(\omega_k)\right\} + 2\left\{ \Gamma xx(\omega_k)\,\Gamma yy(\omega_{N-k}) - \Gamma xy(\omega_k)\,\Gamma yx(\omega_{N-k})\right\}\right\}$$

The methods described in section 5.2.3. can be used to obtain from equations 5.28. approximate variance expressions for the aliased, smoothed, frequency response estimator. However, the expressions obtained in this way are cumbersome, and do not directly indicate the effect of aliasing. The expressions simplify if we restrict our attention to the band of frequencies in the region of the folding frequency. This is the area we are primarily interested in, since it is in this band of frequencies that the effects of aliasing are most noticeable.

For frequencies near the folding frequency we are justified in making the approximation $\omega_k = \omega_{N-k}$ in equations 5.28. The resultant approximate variance expressions for the aliased gain and phase estimators are :-

$$\frac{\mathrm{Var}\left\{ |\hat{H}_A(\omega)| \right\}}{|H_A(\omega)|^2} \simeq \frac{1}{2n}\left\{ \frac{1}{K^2_{xy_A}(\omega)} - 1 \right\}$$

$$\mathrm{Var}\left\{ \hat{\phi}_A(\omega) \right\} \simeq \frac{1}{2n}\left\{ \tan^2 (\phi(\omega)) \right\} \qquad \ldots\ldots\ldots\ldots 5.29$$

From these relationships the following conclusions can be draw :-

i) The variance of the aliased gain estimate near the folding frequency is determined by the aliased coherency. Now the aliased coherency is given by :-

$$K^2_{xy_A}(\omega) = \frac{\Lambda^2_{xy}(\omega)}{\Gamma_{xx}(\omega)\,\Gamma_{yy}(\omega)} \leqslant K^2_{xy}(\omega) \qquad \ldots\ldots\ldots 5.30$$

Therefore, if the cross-spectrum is not real near the folding frequency, the variance of the aliased gain estimate will be inflated. In the extreme case of an imaginary cross-spectrum, the variance will become infinite.

ii) The variance of the aliased phase estimates near the folding frequency is independent of the coherency. It is governed by the square of the tangent of the phase. So again, if the cross-spectrum near the folding frequency is imaginary, the variance of the aliased phase estimate will become very large.

These results explain the rapid increase in the variability of discrete frequency estimates as the folding frequency is approached.

From the above results we can conclude that if the quadrature-spectrum is zero near the folding frequency, then aliasing will not increase the variability of frequency response estimators. A corresponding comment can be made concerning the bias due to aliasing, since if the substitution $\omega_k = \omega_{N-k}$ is made in equations 5.27., then the condition for zero bias error is that the quadrature-spectrum be zero. Therefore, aliasing errors in frequency response estimates are minimised if the cross-spectrum is adjusted so that it is real near the folding frequency. The cross-spectrum can be adjusted in this way by using the alignment procedure outlined in section 5.2.3. This procedure is not recommended as the sole means of controlling aliasing, but rather as an adjunct to a more reliable method, such as the filtering technique described in the next section.

5.3.3. The Control of Aliasing Errors by Filtering

This section deals with a technique for the reduction and control of aliasing errors in digital frequency response estimation. The method employed is to insert low-pass guard filters at appropriate points in the frequency response measurement scheme. These filters are used to control the shape of the system input and output auto-spectra, and thence to reduce the aliasing errors in the frequency response estimates.

From equations 5.27. the bias errors in the aliased gain, phase and squared coherency estimates at frequency ω_k are proportional to the autospectral ratios $R_{xx}(\omega_k)$ and $R_{yy}(\omega_k)$, defined by :-

$$R_{xx}(\omega_k) = \frac{\Gamma_{xx}(\omega_{N-k})}{\Gamma_{xx}(\omega_k)} \qquad \text{..... 5.31}$$

$$R_{yy}(\omega_k) = \frac{\Gamma_{yy}(\omega_{N-k})}{\Gamma_{yy}(\omega_k)}$$

Therefore, if the autospectrum of the system input x(t) and output y(t) can be controlled so that Rxx(ω) and Ryy(ω) are negligible, then the bias errors in the digital frequency response estimates can be neglected. The autospectra of x(t) and y(t) can be controlled by using appropriate low-pass guard filters. Two possible schemes which employ guard filters are shown in fig. 5.3. Figure 5.3.(a) shows the case in which the system is identified using its normal operating input, and Fig. 5.3.(b) shows the alternative situation where the system is identified by superimposing a noise-like test signal upon the

FIGURE 5.3a. FREQUENCY RESPONSE MEASUREMENT SCHEMES.

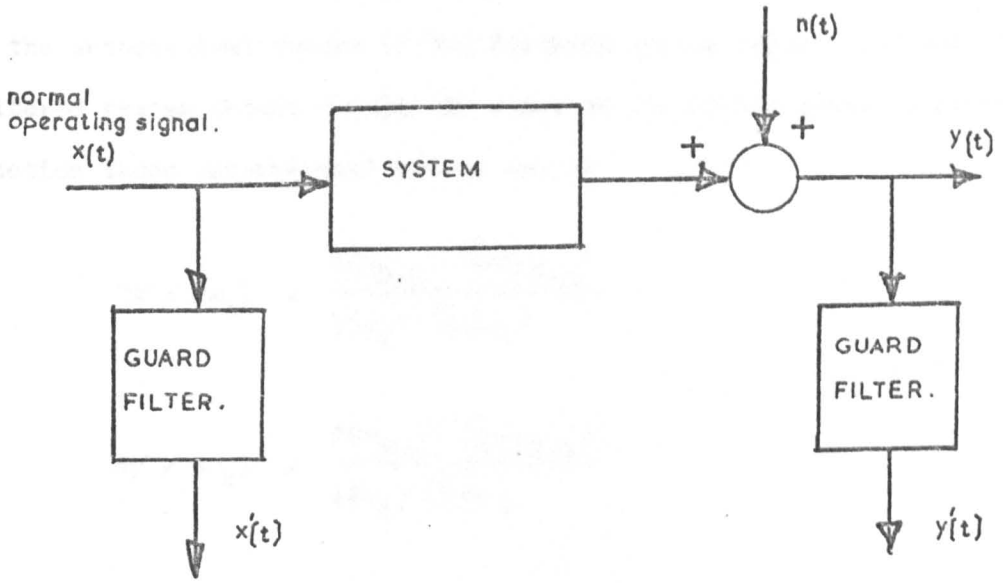
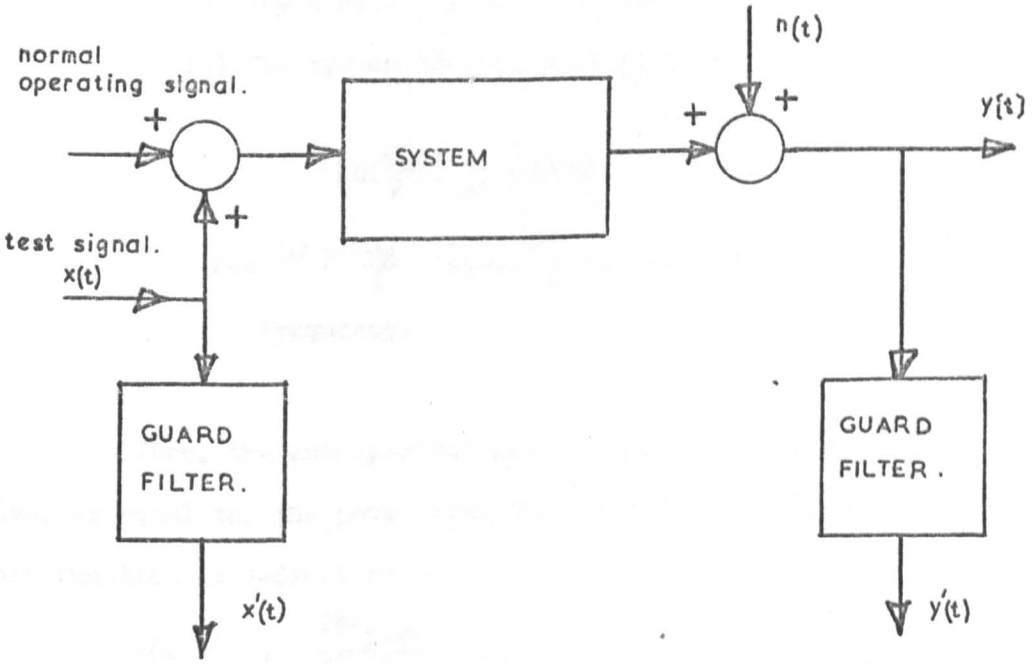


FIGURE 5.3b.



normal operating record.

If the guard filters in fig. 5.3. are identical* and have a power transfer function $f(\omega)$, then the aliasing errors are controlled by the autospectral ratios of the filtered system input $x'(t)$ and the filtered system output $y'(t)$. In terms of the filter power transfer function these autospectral ratios are :-

$$R_{x'x'}(\omega_k) = \frac{f(\omega_{N-k})\ \Gamma_{xx}(\omega_{N-k})}{f(\omega_k)\ \Gamma_{xx}(\omega_k)} \qquad \ldots\ldots\ldots\ldots 5.32$$

$$R_{y'y'}(\omega_k) = \frac{f(\omega_{N-k})\ \Gamma_{yy}(\omega_{N-k})}{f(\omega_k)\ \Gamma_{yy}(\omega_k)}$$

Now, assuming that :-

i) The test signal is a white-noise source,

ii) The corrupting noise is white.

iii) The system $H(\omega)$ is such that :-

$$|H(\tfrac{\omega_s}{2})| \geqslant |H(\omega)|$$

for $\omega > \frac{\omega_s}{2}$, where $\frac{\omega_s}{2}$ is the folding frequency.

Then, the autospectral ratios $Rxx(\omega)$ and $Ryy(\omega)$ will be less than, or equal to, the power transfer ratio $R(\omega)$ of the guard filters. This function is defined by :-

$$R(\omega_k) = \frac{f(\omega_{N-k})}{f(\omega_k)} \qquad \ldots\ldots\ldots\ldots 5.33$$

* Some comments on the effects of non-identical filters in the test schemes of Fig. 5.3 are contained in Appendix 5.

Thus, under these fairly general conditions, the aliasing error can be controlled by proper choice of the guard filter characteristics. Now, the digital frequency response estimates, $\hat{H}(\omega_k)$, are situated at the discrete frequencies $\omega_k = \frac{2\pi k}{\Delta N}$, (where $k = 0,1, \ldots\ldots \frac{N}{2}$). Therefore, to control the aliasing errors, filter characteristics should be chosen which minimise $R(\omega)$ in the range $\omega = 0$ to $\omega = \frac{\pi}{\Delta}$, (where $\frac{\pi}{\Delta} = \frac{\omega_s}{2}$, the folding frequency).

The ideal guard filter has the low-pass, brick-wall, power transfer function, defined by :-

$$f(\omega) = 1 \qquad |\omega| < \frac{\omega_s}{2}$$

$$= 0 \qquad |\omega| > \frac{\omega_s}{2} \qquad \ldots\ldots\ldots\ldots \text{5.34}$$

In this case $R(\omega)$ is zero between zero frequency and the folding frequency, and so the estimates will be unaffected by aliasing. However, the ideal brick-wall filter is physically unrealizeable, and a suitable approximation must be used. There are several well known filter types which embody approximations to the brick-wall function, we will consider three of the best known, these are :-

a) the Butterworth, Maximally Flat, filter

b) the Chebyshev, equal ripple, filter

c) the Optimal 'L' filter

These filters are defined and described in reference 5.10.

When selecting a particular type of filter, two main para-

meters must be specified. These are the half-power, or cutoff frequency, ω_c, and the order of the filter, n. First we consider the factors which determine the selection of an appropriate cutoff frequency.

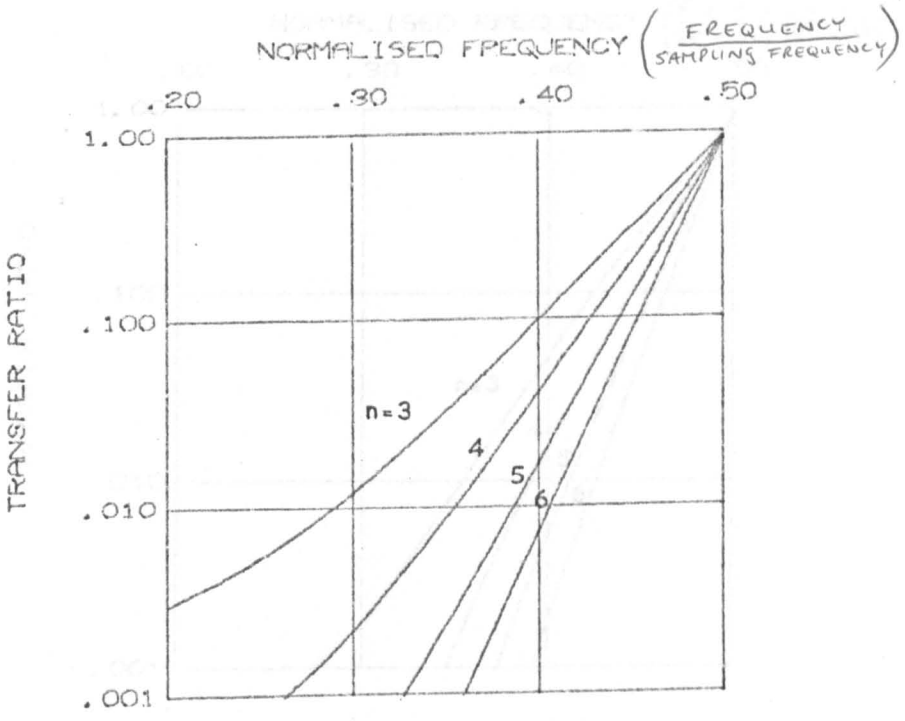
The correct filter cutoff frequency is that frequency which minimises the function $R(\omega)$ in the range 0 to $\frac{\omega_s}{2}$. The optimum choice has been investigated, and for all the above filter types it has been established that $\omega_c = 0$ minimises $R(\omega)$ for all n. In practice, if a cutoff frequency of zero were used, the autospectrum would be highly attenuated for much of the frequency band. In this situation the 'measurement noise' (described in section 5.1.) may seriously impair the accuracy of the frequency response estimates. For this reason, a higher value of cutoff frequency is recommended.

The effect upon $R(\omega)$ of increasing the cutoff frequency is shown for a typical filter in graph 5.2. This shows the power transfer ratio of a third order, Butterworth filter for various ω_c. In this graph the frequency scale is normalised so that the sampling frequency is unity, the 0.5 Hz point on the normalised frequency axis, therefore denotes the folding frequency. From graph 5.2., it is seen that $R(\omega)$ does not increase significantly when the cutoff frequency is increased from zero to $0.3\omega_s$, but as ω_c is increased slightly beyond $0.3\omega_s$ to $0.4\omega_s$, the power transfer ratio increases rapidly. It is therefore concluded that $0.3\omega_s$ is a suitable guard filter cutoff frequency.

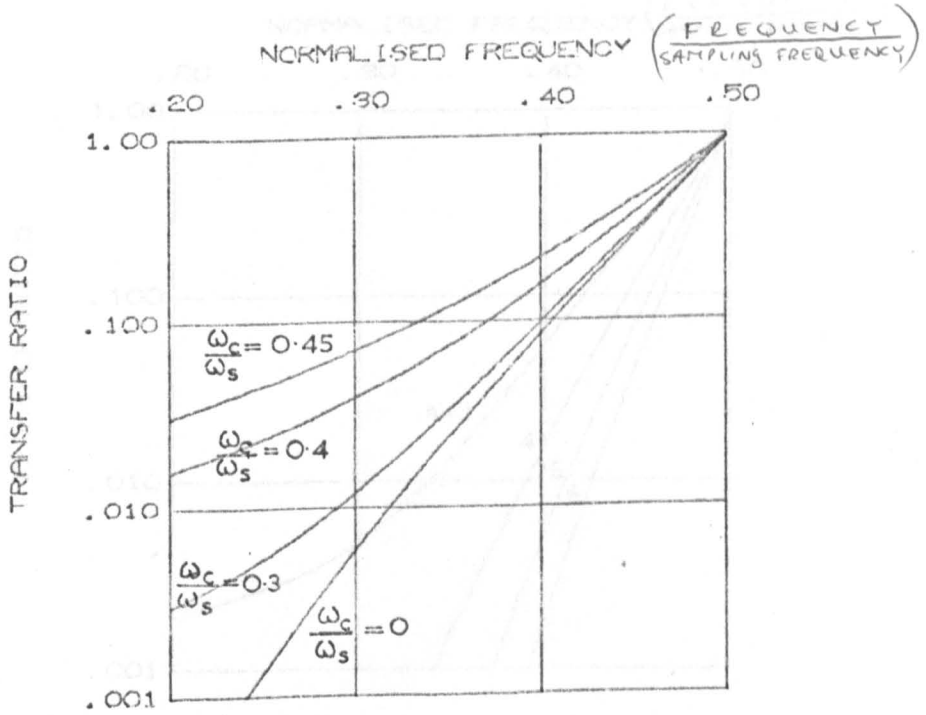
Having selected an appropriate cutoff frequency, the performances of the three filter types listed above may be compared.

GRAPH 5.1.
BUTTERWORTH FILTER TRANSFER RATIO FOR VARIOUS ORDERS

(CUTOFF FREQUENCY ≈ 0.3*WS)

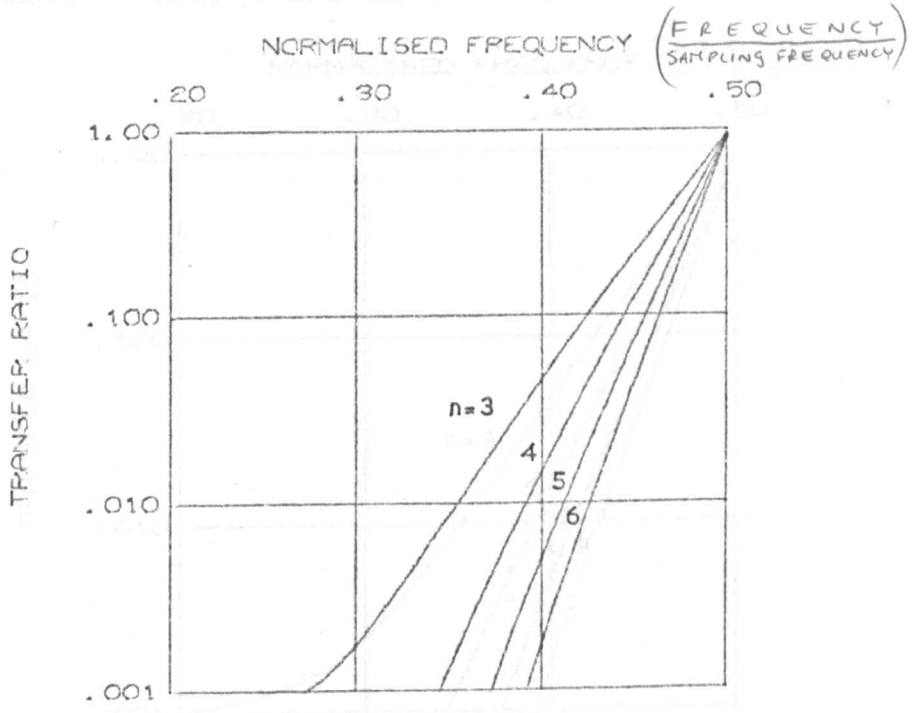


GRAPH 5.2.
THIRD ORDER BUTTERWORTH FILTER TRANSFER RATIO FOR VARIOUS CUTOFF FREQUENCIES.



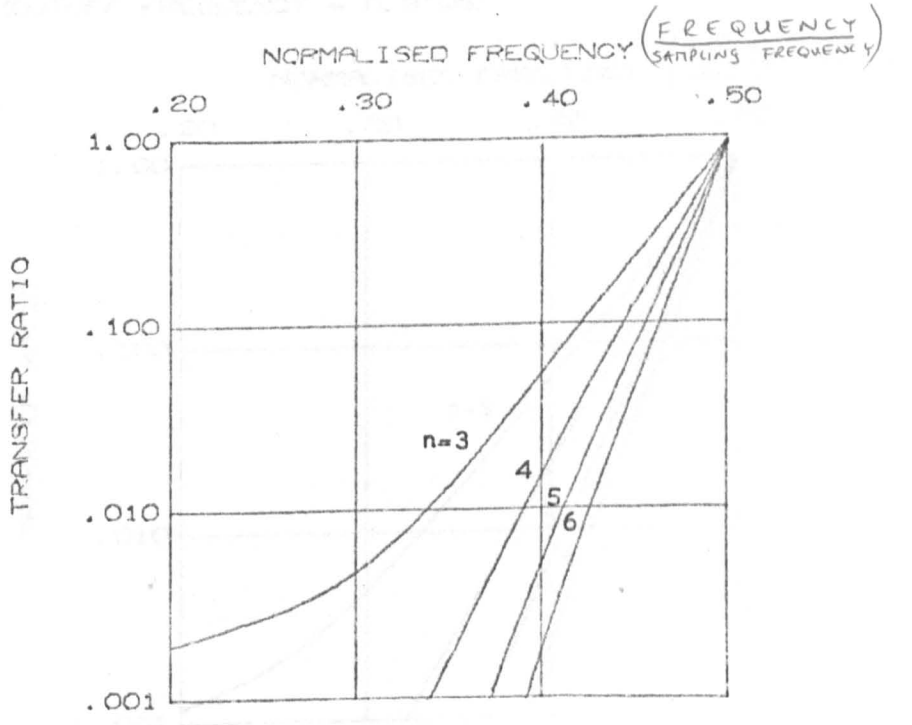
GRAPH 5.3.
CHEBYSHEV FILTER TRANSFER RATIO FOR VARIOUS ORDERS

(CUTOFF FREQ. = 0.3*WS. PASSBAND RIPPLE < 1.5 DB.)



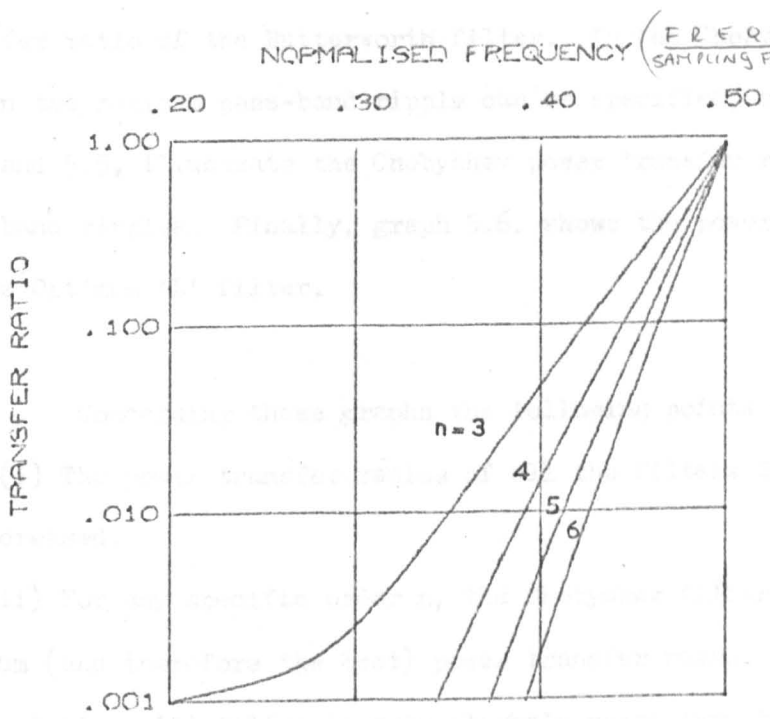
GRAPH 5.4.
CHEBYSHEV FILTER TRANSFER RATIO FOR VARIOUS ORDERS

(CUTOFF FREQ. = 0.3*WS. PASSBAND RIPPLE < 0.5 DB.)



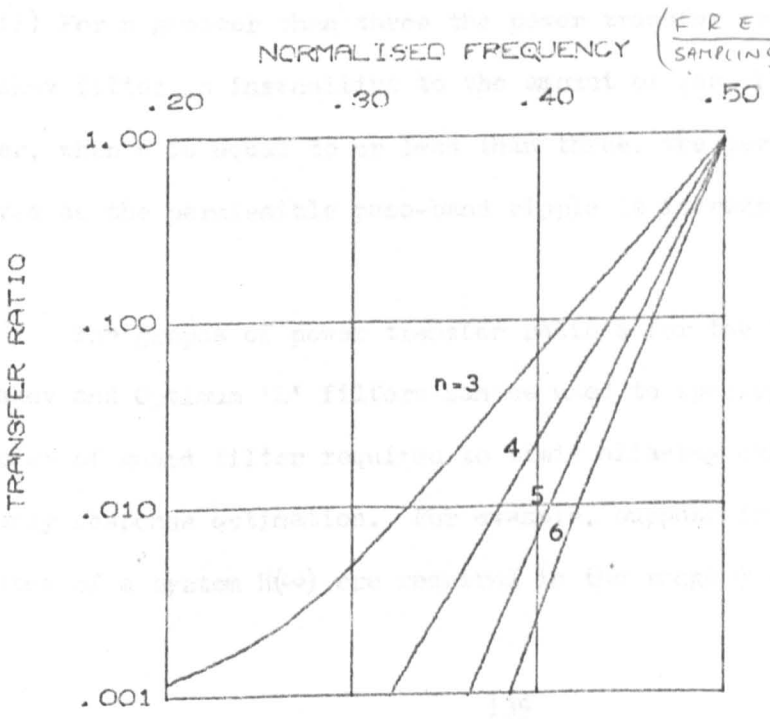
GRAPH 5.5.
CHEBYSHEV FILTER TRANSFER RATIO FOR VARIOUS ORDERS

(CUTOFF FREQ. = 0.3*WS, PASSBAND RIPPLE (1 DB.)



GRAPH 5.6.
OPTIMAL 'L' FILTER TRANSFER RATIO FOR VARIOUS ORDERS

(CUTOFF FREQUENCY = 0.3*WS)



Such a comparison is made in graphs 5.1. to 5.6. These graphs show the power transfer ratios of the Butterworth, Chebyshev, and Optimum 'L' filters, for various Orders, n. Graph 5.1. shows the power transfer ratio of the Butterworth filter. In the Chebyshev filter design the maximum pass-band ripple can be specified, so graphs 5.3, 5.4, and 5.5, illustrate the Chebyshev power transfer ratio for various pass-band ripples. Finally, graph 5.6. shows the power transfer ratio of the Optimum 'L' filter.

Concerning these graphs the following points can be made :-

(i) The power transfer ratios of all the filters decrease as n is increased.

(ii) For any specific order n, the Chebyshev filter provides the minimum (and therefore the best) power transfer ratio. The performance of the Optimum 'L' filter is only slightly worse than the Chebyshev filter, and the Butterworth filter is the least efficient type for suppressing aliasing errors.

(iii) For n greater than three the power transfer ratio of the Chebyshev filter is insensitive to the amount of pass-band ripple. However, when n is equal to or less than three, the performance improves as the permissible pass-band ripple is increased.

The graphs of power transfer ratio's for the Butterworth, Chebyshev and Optimum 'L' filters can be used to specify the type, and order of guard filter required to limit aliasing errors in discrete frequency response estimation. For example, suppose frequency response estimates of a system $H(\omega)$ are required in the range 0 to ω_r, and

fourth order Butterworth filters are available to control aliasing. If a measurement scheme of the type illustrated in figure 5.3. is used, what cutoff frequency and sampling frequency should be employed in order that the bias error in the gain estimate at ω_r will be less than 10% ?

Making the assumptions that the system input $x(t)$ and the corrupting noise have flat autospectra, and that $|H(\omega_r)| > |H(\omega)|$ for $\omega > \omega_r$. Then the worst case bias errors occur at ω_r, and (from equations 5.27. and 5.32.) they are given by :-

$$\frac{B\left\{ |\hat{H}_A(\omega_r)| \right\}}{|H_A(\omega_r)|} \leqslant -2R(\omega_r)$$

$$B\left\{ \hat{\phi}_A(\omega_r) \right\} \leqslant \pm R(\omega_r)$$

$$\frac{B\left\{ \hat{K}^2_{xy_A}(\omega_r) \right\}}{K^2_{xy_A}(\omega_r)} \leqslant -4R(\omega_r)$$

Now from graph 5.1. the power transfer ratio of a fourth order Butterworth filter with cutoff frequency $0.3\omega_s$, is 0.04 at $0.4\omega_s$. If the maximum required frequency ω_r is made equal to $0.4\omega_s$, and a cutoff frequency of $0.3\omega_s$ is used, then the worst case bias errors will be :-

Gain bias ; - 8%

Phase bias ; $\pm$ 0.04 radians, or $\pm$ 2.3 degrees

Squared Coherency bias ; - 16%

In terms of ω_r the sampling frequency is $2.5\omega_r$, and the filter cutoff frequency is $0.75\omega_r$.

Therefore, if fourth order Butterworth filters, with cutoff frequency $0.75\omega_r$, and a sampling frequency of $\omega_s = 2.5\omega_r$ are used in the frequency response measurement scheme, then the bias in the gain will be less than -8% at ω_r. The corresponding worst case errors in phase and squared coherency are $\pm$ 2.3 degrees and -16% respectively. For frequencies below ω_r the errors rapidly become insignificant.

5.4. Real-Time Frequency Response Estimation using an On-Line Digital Computer

5.4.1. General Description

A digital estimate of the frequency response function of a system of the type discussed in 5.2. may be obtained using the smoothed periodogram of the input $x(t)$, and the smoothed cross-periodogram of the input and output $y(t)$. The smoothed periodogram of $x(t)$ is obtained by averaging n raw periodograms, and the smoothed cross-periodogram is obtained in a similar way by averaging raw cross-periodograms. The discrete estimate of the frequency response function is then found by dividing the smoothed cross-periodogram by the smoothed auto-peridogram. This method can be applied to jointly recorded time series of the input and output, by segmenting each series into n segments and calculating the periodograms of each segment. The smoothed discrete frequency response estimate obtained in this way has a variance inversely proportional to n.

It is shown below that this method can be programmed for an on-line digital computer, and that by using the fixed-point FFT, frequency response functions can be estimated in real-time. The input, $x(t)$, and output, $y(t)$, of the system shown in figure 5.1. are fed into the on-line computer via an analogue-to-digital interface. Then, under control of the computer real-time clock (RTC), the computer samples the signals $x(t)$ and $y(t)$ at the RTC frequency, fc. The pairs of data points obtained in this way are stored in a pair of 'input arrays'. When N pairs of data points have been read into the input arrays, they are transferred to a pair of 'working arrays'. The computer then commences to process the data in the working arrays.

This processing is interrupted every RTC period, and a new pair of data readings are collected and stored in the input arrays.

The discrete Fourier transform of the data in the working arrays is obtained using an N point complex FFT as described in chapter 3. The periodogram of the system input and output are formed using the discrete Fourier coefficients. These periodograms are added into arrays in a 'spectral accumulator', the computer then returns to an idling loop. The computer remains here until the input arrays are full again. When this happens the new N pairs of data points are transferred to the working arrays, and the processing starts again. The computer continues in this way, adding periodograms into the spectral accumulator, and the contents of the accumulator become smoother estimates of the auto and cross-spectra. When the periodograms in the spectral accumulator arrays are sufficiently smooth, the periodogram averaging is halted. The frequency response estimate is then calculated.

The usefulness of this real-time estimation procedure is determined by the maximum frequency component of a frequency response function, $H(\omega)$, that can be estimated. This is limited by the maximum RTC frequency which can be used. If $t(N)$ is the time taken to process a pair of N point data blocks. Then, following the reasoning of section 4.3.1., the maximum frequency component that can be studied is $\frac{N}{2}.t(N)$. Most of the processing time is occupied by the discrete Fourier transformation of the data blocks. Now, two real 1024 point data blocks can be transformed in about 2 seconds, using

the fixed-point FFT of chapter 3. Therefore, the maximum frequency component that can be studied in real-time is approximately 250 Hz. Thus, using a properly programmed on-line digital computer, a wide range of systems can be studied.

The principal advantage of on-line frequency response estimation over off-line methods is that the result is available immediately. The on-line method has the additional advantage that extremely stable estimators can be obtained, even in bad measurement conditions. For instance, the programme described in the next section can average, at most, 8192 periodograms. From equation 5.10., this means that provided sufficient input/output data is available, it is possible to estimate frequency response functions with a variance of 1%, when the squared coherency is only $^{1}/_{83}$.

An alternative method which could be used for real-time frequency response estimation is to calculate the frequency response estimate after each new set of periodograms have been added into the spectral accumulator. This method has the advantage that the current estimate of the frequency response function is always available in the computer. However, the division required to calculate the frequency response estimate must, for accuracy, use floating-point arithmetic. This greatly increases the processing-time, and reduces the maximum sampling frequency that can be used. For this reason this method is not recommended.

5.4.2. Programme Description

This section describes the operation of a frequency response estimation routine programmed for an on-line computer. The routine is programmed in assembly code for use on a GEC 90/2 computer.

a) Initial Parameter Setting

At the start of the programme, the computer types a request for various parameters. The parameters are; N the number of data points per block; and p, the Generalised Hanning window parameter. If the input/output signals of the system under test have significant mean levels, these can be suppressed by depressing SES switch Ø on the computer control console. The sampling frequency is set by patching a square wave generator of the desired frequency to the external RTC input.

b) Running the Programme

To start the programme breakpoint 1 on the control console is depressed (set). When this is done, the computer commences averaging periodograms. After each pair of data blocks has been processed, and before the next pair is complete, the computer waits in a display loop. While in this loop, the current estimates of the auto and cross-spectra can be selected for display on an oscilloscope screen. Which spectra is displayed is determined by a set of SES switches as follows :-

SES 1 set, displays the input autospectrum estimate
SES 2 " " " output " "
SES 3 " " " co-spectrum "
SES 4 " " " quad-spectrum "

c) Stopping the Programme

When the estimates appear sufficiently smooth the averaging process can be stopped by setting breakpoint 2. This disconnects the real-time clock and transfers the programme to the display loop. To calculate the frequency response function the breakpoint 3 is set/reset. The computer then divides the estimated cross-spectrum by the estimated autospectrum, and displays the result as follows :-

SES 1 set displays the frequency response in polar form.

SES 2 set displays the real-part of the frequency response.

SES 3 set displays the imaginary-part of the frequency response.

The effect upon the frequency response estimate of further smoothing may be observed by set/resetting SES 5. This convolves the real and imaginary parts of the estimate with the Hanning filter weights $\frac{1}{4}, \frac{1}{2}, \frac{1}{4}$. By repeatedly depressing SES 5, the estimate can be smoothed as much as desired, but with a corresponding loss of frequency resolution. A permanent record of the frequency response estimate can be obtained in one of several ways :- (i) photographing the oscilloscope display, (ii) using the on-line graph-plotter. (This is triggered by set/resetting SES 6) or (iii) punching the contents of the spectral accumulator on to paper tape. (This is triggered by set/resetting breakpoint 4). The paper tape containing the contents of the spectral accumulator can then be used as data for an off-line data handling programme.

d) Restarting the Programme

The variance of the frequency response estimate is inversely

proportional to the coherency between input and output, and the number of periodograms averaged. Therefore, if the coherency is low, it is possible that the auto, and cross-spectrum estimators will appear smooth, but the corresponding frequency response estimate will not. In such a case the operator can restart averaging by set/resetting breakpoint 1. The computer will then inquire, via the typewriter, if the spectral accumulator is to be cleared. If the operator types 'False' the computer will start averaging periodograms again with the contents of the spectral accumulator undisturbed.

e) Resetting the Programme

To reset the programme breakpoint 1 should be reset, and breakpoint 4 set/reset. This returns the programme to the initial setting-up stage, and allows the operator to select new programme parameters.

5.5. Practical Aspects of Real-Time Digital Frequency Response Estimation

5.5.1. Accuracy

Neglecting the noise sources associated with the systems under test, the accuracy of the frequency response estimation routines described in section 5.4. is determined by the errors in the collection and processing of the data. The chief sources of these errors are the round-off error in the fixed-point FFT, and the quantisation error caused by analogue-to-digital conversion.

These errors can be combined to form an equivalent measurement noise source. If $m_i(t)$ is the measurement noise incurred by processing data collected from the input to the system, and $m_o(t)$ is the measurement noise involved in processing data collected from the output of the system, then, assuming that $m_i(t)$ and $m_o(t)$ are independent, and neglecting other bias errors, the measured frequency response has an expectation :-

$$E\left\{ \hat{H}(\omega) \right\}_{\text{measured}} = \frac{H(\omega)}{1 + \frac{\Gamma_{ii}(\omega)}{\Gamma_{xx}(\omega)}} \qquad \ldots\ldots\ldots\ldots \quad 5.35.$$

Where $\Gamma_{ii}(\omega)$ is the autospectrum of the input measurement noise $m_i(t)$.

It is justifiable to assume that the measurement noise is white. Therefore, if the measurement noise has a variance σ_{ii}^2, and the input signal is white with variance σ_{xx}^2, the bias error

introduced by measurement errors is approximately* :-

$$B\left\{H(\omega)\right\} = -\frac{\sigma_{ii}^2}{\sigma_{xx}^2}\left\{H(\omega)\right\} \qquad \ldots\ldots\ldots\ldots 5.36.$$

The frequency response estimation programme described in section 5.5. uses a double precision fixed-point FFT, so that the rounding error is negligible when compared with the quantization error. From section 4.4.1. the quantization error in the GEC 90/2 analogue-to-digital converter has a variance of $\frac{1}{12}\ 2^{-24}$ in the computer scaling. The analogue to digital converter is scaled so that 10 volts signal is equivalent to $\frac{1}{2}$ in the computer. The quantisation error in the GEC 90/2 is therefore $8.4\ 2^{-22}$ volts^2, and the fractional bias error caused by measurement error is :-

$$\frac{B\{H(\omega)\}}{H(\omega)} \simeq -\frac{8.4\ 2^{-22}}{\sigma_{xx}^2}$$

So, for example, if the r.m.s. level of x(t) is greater than 0.5 volt, the mean measurement error will be negligible ($\approx 0.001\%$).

5.5.2. Limitation on Sampling Frequency and Frequency Resolution

The theoretical maximum sampling frequency of the real-time frequency response estimation programme described here is determined by the time taken to process a pair of N-point real data blocks. This processing involves, i) a complex N-point fixed-point FFT, ii) The removal of the sample mean levels, iii) Linear modification of the

*This approximation only holds when the measurement noise spectrum $\Gamma_{ii}(\omega)$ is very much smaller than $\Gamma_{xx}(\omega)$.

data, iv) and forming three periodograms.

Using the fixed-point FFT, which employs rescaling in the FFT loop by the modulus test, the processing time for a pair of 256-point data blocks is approximately 0.62 seconds. Therefore, the maximum theoretical sampling frequency is 410 Hz, and the useful frequency range is 0 to 200 Hz.

The practical maximum sampling frequency of the programmes using the GEC 90/2 computer is set up by the GEC analogue input multiplexer. This takes 8.8ms to switch from reading one analogue input to another. To avoid errors due to fluctuations in the multiplexer reset time, a software delay of 9ms is programmed into the routines. This limits the maximum practical sampling frequency to 55 Hz, giving a useful frequency range of 0 to 25 Hz.

The frequency resolution of the frequency response estimation routines is determined by N, the number of points per block; ω_s, the sampling frequency; and p, the Generalised Hanning window parameter.

From section 4.2.6., the frequency resolution using the Generalised Hanning window is determined by ω_B, its half power bandwidth.

$$\omega_B = \frac{\omega_s}{N}(1 + 2p)$$

As described in section 4.2.6., p is chosen to set the desired asymptotic decay rate of the Generalised Hanning window

frequency response. In theory p may take any value between 0 and $\frac{1}{2}$, but in the computer programmes p is restricted to be a binary fraction ($\frac{1}{2}$, $\frac{1}{4}$, ...).

Having set p, and chosen an appropriate sampling frequency, N can be selected to give the required resolution. In the on-line computer programmes N is restricted by the FFT algorithm to be of the form 2^m(m.integer). The maximum value of N is 256, this limit is set by the length of the trigonometric 'look-up' tables used in the FFT. The choice of 256 as the maximum value was an arbitrary one made during the initial compilation of the programmes, The maximum value of N can be increased by extending the trigonometric table. The ultimate limit is set by the size of the computer store, on the GEC 90/2 computer, with 8K of store, this limit is 512 points.

REFERENCES

5.1. Bendat, J.S. and Piersol, A.G., "Measurement and Analysis of Random Data", Wiley 1966, pp. 81 - 85.

5.2. Akaike, H. and Yamamouchi, Y., "On the Statistical Estimation of Frequency Response Functions", Ann. Inst. Stat. Math. Vol. 14, No. 1, 1962.

5.3. Jenkins, G.M. and Watts, D.G., "Spectral Analysis and it's Applications", Holden-Day 1968, pp. 395-403.

5.4. Benignus, V., "Estimation of the Coherence Spectrum and Its Confidence Interval Using the Fast Fourier Transform", I.E.E.E. Trans., Vol. AU-17, No. 2, June 1967.

5.5. Goodman, N.R., "On the Joint Estimation of the Spectra, Co-Spectra and Quadrature Spectra of a Two-Dimensional Stationary Gaussian Process", Scientific Paper No. 10., New York Univ., 1957.

5.6. Jenkins, G.M. and Watts, D.G., "Spectral Analysis and it's Application, Holden-Day 1968, pp. 430 - 436.

5.7. As ref. 5.1. above

5.8. Alexander, M.J., and Vok, C.A., "Table of the Cumulative Distribution of Sample Multiple Coherence", Research Report 63-67, Rocketdyne Division, North American Aviation Inc., November 1963.

5.9. Enochson, L.D. and Goodman, N.R., "Gaussian Approximation to the Distribution of Sample Coherence", AFFDL TR 65-67, Research and Technology Division, AFSC, Wright-Patterson AFB, Ohio, 1965.

5.10. Kuo, F.F., "Network Analysis and Synthesis", Wiley 1962, pp. 327 - 365.

6. THE ESTIMATION OF FREQUENCY RESPONSE FUNCTIONS ASSOCIATED WITH CLOSED-LOOP SYSTEMS

6.1. Introduction

This chapter presents new theoretical results concerning frequency response estimators associated with closed loop systems. The system considered is shown in fig. 6.1. In this system the frequency response function of the forward path is $H_1(\omega)$, and the feedback path is $H_2(\omega)$. The overall frequency response function is designated $G(\omega)$ and is defined by :-

$$G(\omega) = \frac{H_1(\omega)}{1+H_1H_2(\omega)} \qquad \ldots\ldots\ldots\ldots 6.1.$$

The following assumptions are made concerning the system :-

i) The input signal is a stationary, zero-mean, gaussian noise source.

ii) All transfer functions are linear, time-invariant, two port networks.

iii) There is a stationary, zero-mean, gaussian noise source associated with the forward and the feedback path, and these sources and the input signal are mutually uncorrelated.

iv) The feedback path cannot be broken to make measurements.

The fourth assumption is made because, often we have to make frequency response measurements on closed-loop systems with the system 'in-situ' and operational. Under these conditions the feedback path cannot be broken without endangering the normal operation

FIGURE 6.1. A CLOSED LOOP SYSTEM.

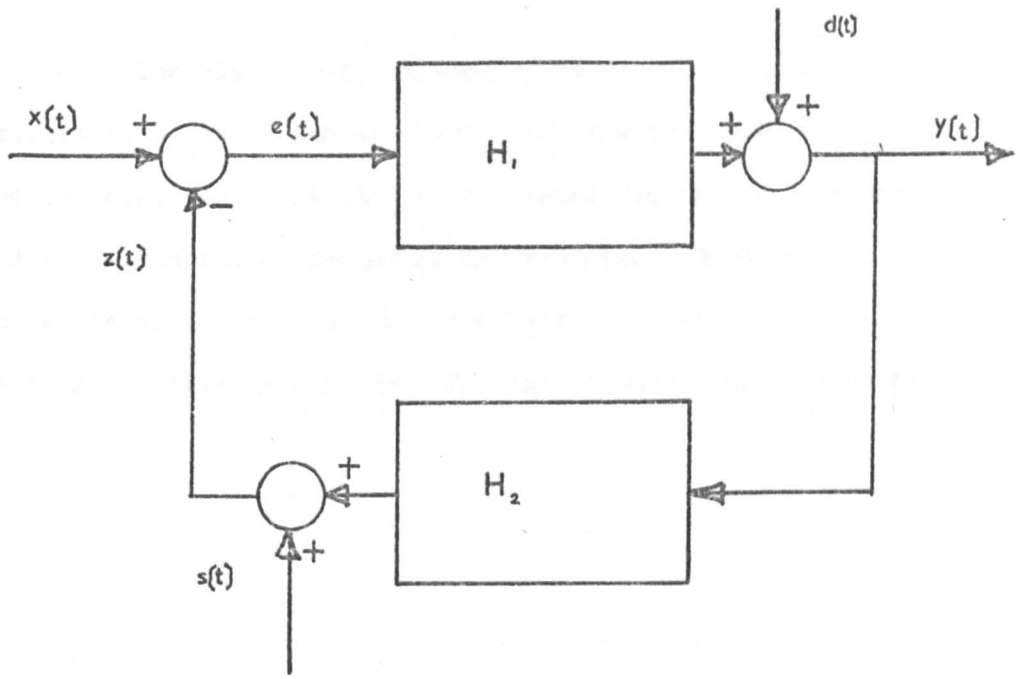
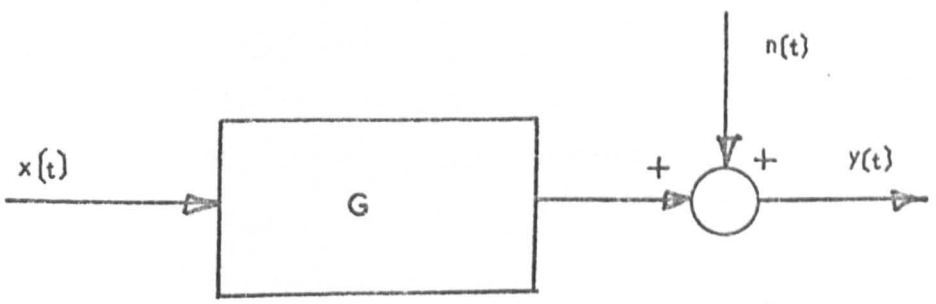


FIGURE 6.2. AN EQUIVALENT OPEN LOOP SYSTEM.



of the system. Unfortunately, 'normal' estimates of $H_1(\omega)$ and $H_2(\omega)$ based upon earlier techniques, and made with the loop closed, yield biased results.

The aim of this chapter is to develop some of the statistical properties of an estimator which overcomes this difficulty. In section 6.2. the shortcomings of 'normal' estimators are described, and a new, unbiased, estimator is presented. Section 6.3. deals with the variance and bias of this new estimator, and in the remaining sections confidence intervals for the estimator are discussed.

List of Symbols in Chapter 6

$\bar{C}_{xx}(\omega)$	smoothed autospectrum estimator
$\bar{C}_{xy}(\omega)$	smoothed cross-spectrum estimator
$d(t)$	the noise source associated with the feedforward path
$e(t)$	the error signal
$E_i(\omega)$	Fourier transform of the i^{th} sample of $e(t)$
$G(\omega)$	the overall Frequency response function
$H_1(\omega)$	forward path frequency response function
$H_2(\omega)$	feedback path frequency response function
$n(t)$	a noise source
n	the number of sample spectra averaged to obtain a smoothed spectral estimate
P	a probability
$\bar{P}_{xy}(\omega)$	smoothed estimate of the co-spectrum of $x(t)$ leading $y(t)$
$\bar{Q}_{xy}(\omega)$	smoothed estimate of the quad-spectrum of $x(t)$ leading $y(t)$
r	radius of a z-plane confidence circle
R	radius of a w-plane confidence circle
$s(t)$	the noise source associated with the feedback path
$S_d(\omega)$	ratio of the noise autospectrum to input signal autospectrum
u	a real variable
u_c	real co-ordinate of the centre of a w-plane confidence circle
u_o	real part of the forward path frequency response function
v	a real variable
v_c	imaginary co-ordinate of the centre of a w-plane confidence circle
v_o	imaginary part of the forward path frequency response function

w	a complex variable
w_c	centre point of a w-plane confidence circle
w_o	the complex point denoting $H_1(\omega)$ in the w-plane
x	a real variable
x_o	the real co-ordinate of $G(\omega)$ in the z-plane
$x(t)$	the input
$X_i(\omega)$	Fourier transform of the i^{th} sample of $x(t)$
y	a real variable
y_o	the imaginary co-ordinate of $G(\omega)$ in the z-plane
$y(t)$	the output
$Y_i(\omega)$	Fourier transform of the i^{th} sample of $y(t)$
z	a complex variable
$z(t)$	the feedback signal
α_1	upper confidence limit of $\text{Arg}\left\{ \hat{H}_1(\omega) - H_1(\omega) \right\}$
α_2'	lower confidence limit on $\text{Arg}\left\{ \hat{H}_1(\omega) - H_1(\omega) \right\}$
β	imaginary part of ρ
γ	real part of ρ
$\Gamma_{xx}(\omega)$	autospectrum of $x(t)$
$\Gamma_{xy}(\omega)$	cross-spectrum of $x(t)$ leading $y(t)$
Δ_1	upper confidence limit on $\lvert\hat{H}_1(\omega) - H_1(\omega)\rvert$
Δ_2	lower confidence limit on $\lvert\hat{H}_1(\omega) - H_1(\omega)\rvert$
$\epsilon(\omega)$	the argument of $\Gamma_{xe}(\omega)$
$\lambda(\omega)$	the argument of $\Gamma_{xy}(\omega)$
ρ	the point $H_2(\omega)^{-1}$ in the z-plane
$\psi(\omega)$	the argument of $H_1(\omega)$
ω	angular frequency

6.2. The Estimators

It is assumed that smoothed auto, and cross-spectral estimators are obtained, in the manner of chapter 5, from continuous, jointly recorded, samples of the input, error and output signal. The relevant, smoothed, spectral estimators are :-

$$\bar{C}xx(\omega) = \frac{1}{n}\sum_{i=1}^{n} Cxx^{(i)}(\omega)$$

$$\bar{C}yy(\omega) = \frac{1}{n}\sum_{i=1}^{n} Cyy^{(i)}(\omega)$$

$$\bar{C}xy(\omega) = \frac{1}{q}\sum_{i=1}^{n} Cxy^{(i)}(\omega)$$

$$\bar{C}xe(\omega) = \frac{1}{n}\sum_{i=1}^{n} Cxe^{(i)}(\omega)$$

As before, these estimates are obtained by sub-dividing each sample into n segments, and averaging the spectra of the segments.

6.2.1. The Overall Transfer Function

An estimate of the overall frequency response function of the system in figure 6.1. may be obtained using the methods of chapter 5. An equivalent two port is given in figure 6.2. In this figure $G(\omega)$ is the overall transfer function of the closed-loop system, and the spectrum of the equivalent noise source $n(t)$ is

given by :-

$$\Gamma_{nn}(\omega) = \left\{ \Gamma_{dd}(\omega) + \Gamma_{ss}(\omega) \left| H_1(\omega) \right|^2 \right\} \left\{ \left| 1 + H_1 H_2(\omega) \right| \right\}^{-2}$$

The appropriate unbiased estimator of the overall transfer function is, from chapter 5, given by :-

$$\hat{G}(\omega) = \frac{\bar{C}xy(\omega)}{\bar{C}xx(\omega)} \qquad \ldots\ldots\ldots\ldots 6.3.$$

6.2.2. The Forward Path Transfer Function

The normal estimator of $H_1(\omega)$ as used in open loop analysis is of little use in estimating the forward path frequency response function. This is because even if the effects of finite records are negligible, it gives a biased estimate. This may be seen if the expectation of the 'normal' estimator $\frac{\bar{C}ey(\omega)}{\bar{C}ee(\omega)}$ is considered. Neglecting the bias caused by finite record lengths, the expected value of the 'normal' estimator is :-

$$E\left\{ \frac{\bar{C}ey(\omega)}{\bar{C}ee(\omega)} \right\} \simeq H_1(\omega) \left\{ \frac{1 + S_s(\omega) - S_d(\omega) . \frac{H_2^*(\omega)}{H_1(\omega)}}{1 + S_s(\omega) + S_d(\omega) \left| H_2(\omega) \right|^2} \right\}$$

where : - $S_s(\omega) = \frac{\Gamma ss(\omega)}{\Gamma xx(\omega)}$, and $S_d(\omega) = \frac{\Gamma dd(\omega)}{\Gamma xx(\omega)}$

Note that the 'normal' estimator will be unbiased only if the noise in the forward path is zero, or if the feedback is disconnected.

To illustrate the nature of the bias associated with this estimator, consider the following example. Suppose that $H_1(\omega) = \frac{1}{1+j\omega}$; the input signal $x(t)$ and the corrupting noise $d(t)$ are white, and are of equal variance; and there is unity (noise-free) feedback. The normal estimation procedure will yield an estimate of $H_1(\omega)$ whose expectation is :-

$$E\left\{\frac{\bar{C}ey(\omega)}{\bar{C}ee(\omega)}\right\} = H_1(\omega)\left\{\frac{1 - \frac{1}{H_1(\omega)}}{2}\right\} = -\frac{j\omega}{2}\left\{H_1(\omega)\right\}$$

Thus it is clear that estimates based on the techniques of chapter 5 are biased in such a way that they may cause false conclusions to be drawn concerning the dynamic structure of the system. For instance, in the above example we would be led to believe that, in addition to a first order lag, there is differential action in the forward path.

The bias associated with normal estimators arises because, due to the feedback, the signal $e(t)$ is correlated with the noise sources. An unbiased estimator of $H_1(\omega)$ may be obtained by introducing the input signal $x(t)$ into the estimator, and using a ratio of cross-spectra as our frequency response function estimator. A suitable unbiased estimator of the forward path transfer function is :-

$$\hat{H}_1(\omega) = \frac{\bar{C}xy}{\bar{C}xe}(\omega) \qquad \ldots\ldots\ldots\ldots 6.4.$$

6.2.3. The Feedback Path Transfer Function

Using similar reasoning, the 'normal' estimator of $H_2(\omega)$ can be shown to be biased. The expected value of the 'normal' estimator $\frac{\bar{C}yz(\omega)}{\bar{C}yy(\omega)}$ is (neglecting the effects of finite duration records) given by :-

$$E\left\{\frac{\bar{C}yz(\omega)}{\bar{C}yy(\omega)}\right\} \simeq H_2(\omega)\left\{\frac{|H_1(\omega)|^2 + S_d(\omega) - \frac{H_1^*(\omega)}{H_2(\omega)} S_s(\omega)}{|H_1(\omega)|^2 + S_d(\omega) + |H_1(\omega)|^2 S_s(\omega)}\right\}$$

Where : - $S_d(\omega) = \frac{\Gamma dd(\omega)}{\Gamma xx(\omega)}$, and $S_s(\omega) = \frac{\Gamma ss(\omega)}{\Gamma xx(\omega)}$

The conditions for zero bias error are that the feed-forward loop be broken, or, that the noise in the feedback path be zero. As in the forward path case the bias error is caused by correlation between the noise sources and the signals in the loop, and an unbiased estimator may be obtained by introducing a reference signal which is uncorrelated with the noise sources. A suitable reference source is the input signal $x(t)$, and an unbiased estimator formed by using $x(t)$ as a reference is :-

$$\hat{H}_2(\omega) = \frac{\bar{C}xz}{\bar{C}xy}(\omega) \qquad \ldots\ldots\ldots\ldots 6.5.$$

In the sections that follow some statistical properties of the forward path transfer function estimate defined by equation 6.4. are given. The properties of the feedback path transfer function estimate are omitted, since they can be calculated by an obvious

rearrangement of the results for the forward path transfer function.

6.3. Variance and Bias of Forward Path Estimators

6.3.1. Variance

The forward path frequency response estimator is defined as :-

$$\hat{H}_1(\omega) = \frac{\bar{P}xy(\omega) + j\,\bar{Q}xy(\omega)}{\bar{P}xe(\omega) + j\,\bar{Q}xe(\omega)}$$

The gain estimator is defined as :-

$$|\hat{H}_1(\omega)| = \frac{\left\{\bar{P}xy^2(\omega) + \bar{Q}xy^2(\omega)\right\}^{\frac{1}{2}}}{\left\{\bar{P}xe^2(\omega) + \bar{Q}xe^2(\omega)\right\}^{\frac{1}{2}}}$$

and the phase estimator is defined as :-

$$\hat{\Psi}(\omega) = \tan^{-1}\left\{\frac{\bar{Q}xy(\omega)}{\bar{P}xy(\omega)}\right\} - \tan^{-1}\left\{\frac{\bar{Q}xe(\omega)}{\bar{P}xe(\omega)}\right\}$$

Therefore, using the methods of section 5.2.3. approximate expressions for the variance of the gain and phase estimators may be obtained in terms of the covariance matrix of $\bar{P}xy$, $\bar{Q}xy$, $\bar{P}xe$, $\bar{Q}xe(\omega)$.

The covariance matrix (obtained using the results of appendix 2) is given in equation 6.7. In this matrix the S denotes diagonal symmetry, and for compactness the dependence upon the frequency has been dropped.

$$\mathrm{Cov}\begin{bmatrix} \bar{P}xy \\ \bar{Q}xy \\ \bar{P}xe \\ \bar{Q}xe \end{bmatrix} = \frac{1}{4n}\begin{bmatrix} 2\Gamma_{xx}\Gamma_{yy} + \Gamma_{xy}^2 + \Gamma_{yx}^2, & j\left\{\Gamma_{yx}^2 - \Gamma_{xy}^2\right\}, & \Gamma_{xe}\Gamma_{xy} + \Gamma_{yx}\Gamma_{ex} + \Gamma_{xx}(\Gamma_{ye} + \Gamma_{ey}), & j\left\{\Gamma_{xx}(\Gamma_{ey} - \Gamma_{ye}) + \Gamma_{yx}\Gamma_{ex} - \Gamma_{xe}\Gamma_{xy}\right\} \\ & 2\Gamma_{xx}\Gamma_{yy} - \Gamma_{xy}^2 - \Gamma_{yx}^2, & j\left\{\Gamma_{ex}\Gamma_{yx} - \Gamma_{xy}\Gamma_{xe} + \Gamma_{xx}(\Gamma_{ye} - \Gamma_{ey})\right\}, & \Gamma_{xx}(\Gamma_{ye} + \Gamma_{ey}) - \Gamma_{xe}\Gamma_{xy} - \Gamma_{yx}\Gamma_{ex} \\ & & 2\Gamma_{xx}\Gamma_{ee} + \Gamma_{xe}^2 + \Gamma_{ex}^2, & j\left\{\Gamma_{ex}^2 - \Gamma_{xe}^2\right\} \\ & S & & \\ & & & 2\Gamma_{xx}\Gamma_{ee} - \Gamma_{xe}^2 - \Gamma_{ex}^2 \end{bmatrix}$$

..... 6.7.

The approximate expressions for the variance of the forward path gain and phase estimates obtained from the covariance matrix are :-

$$\frac{\mathrm{Var}\left\{\left|\hat{H}_1(\omega)\right|\right\}}{\left|H_1(\omega)\right|^2} \simeq \frac{1}{2n}\left\{\frac{S_d(\omega)}{\left|G(\omega)\right|^2}\right\} \qquad \ldots\ldots\ldots\ldots 6.9.$$

$$\mathrm{Var}\left\{\hat{\Psi}(\omega)\right\} \simeq \frac{1}{2n}\left\{\frac{S_d(\omega)}{\left|G(\omega)\right|^2}\right\}$$

Where, $G(\omega)$ is the overall frequency response function; and and, $S_d(\omega) = \dfrac{\Gamma_{dd}(\omega)}{\Gamma_{xx}(\omega)}$.

The above expressions are given in terms of the noise and signal spectra so that the effect that the noise signals have upon the estimators may be directly assessed. To provide a basis for this assessment, the variance figures for the 'open-loop' estimate of $H_1(\omega)$ are now given in terms of the noise and signal spectra. These 'open-loop' variance expressions are those which would apply if the feedback path were opened and the frequency response of $H_1(\omega)$ were estimated using $\dfrac{\bar{C}xy(\omega)}{\bar{C}xx(\omega)}$. Denoting this estimator as $\hat{H}_1(\omega)_{O/L}$ we can write :-

$$\frac{\mathrm{Var}\left\{\left|\hat{H}_1(\omega)_{O/L}\right|\right\}}{\left|H_1(\omega)\right|^2} \simeq \frac{1}{2n}\left\{\frac{S_d(\omega)}{\left|H_1(\omega)\right|^2}\right\}$$

$$\mathrm{Var}\left\{\Psi(\omega)_{O/L}\right\} \simeq \frac{1}{2n}\left\{\frac{S_d(\omega)}{\left|H_1(\omega)\right|^2}\right\}$$

And denoting the 'closed-loop' estimator $\frac{\bar{C}xy(\omega)}{\bar{C}xe(\omega)}$ as $\hat{H}_1(\omega)_{C/L}$, then we can write :-

$$\mathrm{Var}\left\{ \left|\hat{H}_1(\omega)_{C/L}\right| \right\} = \left|1 + H_1H_2(\omega)\right|^2 \mathrm{Var}\left\{\left|\hat{H}_1(\omega)_{O/L}\right|\right\}$$

$$\mathrm{Var}\left\{ \hat{\Psi}(\omega)_{C/L} \right\} = \left|1 + H_1H_2(\omega)\right|^2 \mathrm{Var}\left\{\hat{\Psi}(\omega)_{O/L}\right\}$$

From these expressions it may be seen that the variability of the estimator $\frac{\bar{C}xy(\omega)}{\bar{C}xe(\omega)}$ is $\left|1 + H_1H_2(\omega)\right|^2$ times that which would occur if the loop were opened and the 'normal' estimator were used. Therefore, the variability of the estimator $\frac{\bar{C}xy(\omega)}{\bar{C}xe(\omega)}$ depends upon the nature of the feedback as well as upon the usual factors.

As an example, consider the simple case where the forward path transfer function is $\frac{1}{1 + j\omega}$; the input signal x(t), and the disturbance d(t) are white noise signals with variances $\sigma_{\bar{x}x}^2$ and $\sigma_{\bar{d}d}^2$ respectively, and the feedback transfer function is the real constant B. The variance of the forward path gain estimate at $\omega = 1$ radians/second is from equation 6.9. :-

$$\frac{\mathrm{Var}\left\{ \left|\hat{H}_1(1)\right| \right\}}{\left|H_1(1)\right|^2} \;\underline{\Omega}\; \frac{\sigma_{\bar{d}d}^2\left\{(1 + B)^2 + 1\right\}}{2n\ \sigma_{\bar{x}x}^2}$$

If the constant B were unity, the variance figure would be :-

$$\frac{5}{2n} \cdot \frac{\sigma_{\bar{d}d}^2}{\sigma_{\bar{x}x}^2}$$

If the constant B is doubled, the variance figure also doubles, and to reduce the variability to its previous level we would have to double n, or increase the r.m.s. level of the input signal by a factor of root two.

The approximate variance expressions given in equations 6.9. indicate that the variability of the forward path frequency response estimator is nominally independent of the disturbance in the feedback loop. This interesting result arises because in arriving at equations 6.9. we have neglected all but first order terms in a Taylor series expansion of the gain and phase estimators (see appendix 4). This point may be clarified if we consider the forward path frequency response estimator in terms of the autospectrum of x(t) and the cross-spectra between x(t) and the noise sources s(t) and d(t). The forward path estimator expanded in this way is :-

$$\hat{H}_1(\omega) = H_1(\omega) \left\{ \frac{1 - \frac{\bar{C}xs}{\bar{C}xx}(\omega) + \frac{\bar{C}xd}{H_1\bar{C}xx}(\omega)}{1 - \frac{\bar{C}xs}{\bar{C}xx}(\omega) - \frac{H_2\bar{C}xd}{\bar{C}xx}(\omega)} \right\}$$

Now, when s(t) and d(t) are uncorrelated with x(t), the smoothed cross-spectral estimates $\bar{C}xs(\omega)$ and $\bar{C}xd(\omega)$ have expected values of zero, and variances proportional to $\frac{1}{n}$, so that as n is increased the terms $\frac{\bar{C}xs(\omega)}{\bar{C}xx(\omega)}$ and $\frac{\bar{C}xd(\omega)}{\bar{C}xx(\omega)}$ will become small. Assuming that these terms are much smaller than one, we can make the approximation :-

$$\hat{H}_1(\omega) \simeq H_1(\omega) \left\{ 1 - \frac{\bar{C}xs}{\bar{C}xx}(\omega) + \frac{\bar{C}xd}{H_1\bar{C}xx}(\omega) \right\} \left\{ 1 + \frac{\bar{C}xs}{\bar{C}xx}(\omega) - \frac{H_2\bar{C}xd}{\bar{C}xx}(\omega) \right\}$$

$$\simeq H_1(\omega) \left\{ 1 + \frac{\bar{C}xd}{\bar{C}xx}(\omega) . G(\omega) \right\}$$

Thus we have verified that, to a first order approximation, the variability of the forward path frequency response estimate is independent of s(t).

6.3.2. Bias

The method used to derive expressions for the bias due to data-windowing in the open-loop case can be extended to determine the bias due to data-windowing in the closed loop case. The approximate bias expressions obtained in this way are :-

$$B\left\{ |\hat{H}_1| \right\} \simeq -\frac{a_2}{T^2} \left\{ |\ddot{H}_1| + 2|\dot{H}_1| \frac{|\dot{\Gamma}xe|}{|\Gamma xe|} - |H_1|(\dot{\Psi}\dot{\epsilon} + \dot{\Psi}\dot{\lambda}) \right\} \quad \ldots\ldots 6.9.$$

$$B\left\{ \hat{\Psi} \right\} \simeq -\frac{a_2}{T^2} \left\{ \ddot{\Psi} + 2 \frac{|\dot{H}_1|}{|H_1|} (\dot{\Psi} + \dot{\epsilon}) + \frac{|\dot{\Gamma}xe|}{|\Gamma xe|} \Psi \right\}$$

Where : Ψ = Arg (H_1)

λ = Arg (Γxy)

ϵ = Arg (Γxe)

a_2 = the coefficient of t^2 in the power series of $\frac{1}{q}(1(t) * 1(t))$, and the dot notation is used to indicate derivatives with respect to ω.

Comparing the above expressions with the open-loop expressions for bias due to windowing the following points can be made :-

i) Both gain and phase biases are inflated by additional terms proportional to the first derivative of $|\Gamma_{xe}|$.

ii) The gain bias contains terms proportional to $\dot{\Psi}\dot{\epsilon}$ and $\dot{\Psi}\dot{\lambda}$ Therefore, if large transport delays occur between x(t), e(t) and x(t), y(t) significant bias errors in the gain estimate can be expected. For the reason given in chapter 5 this type of error is not significant in the estimators used by the author. However, should the 'lagged-products' method of spectral estimation be used, care should be taken to align x(t), e(t) and x(t), y(t) prior to estimating the cross-spectra.

6.4. Sampling Distributions Associated with the Forward Path Frequency Response Estimator

In this section the problem of obtaining sampling distributions to suit estimators of the form of equation 6.4., 6.5., is discussed.

6.4.1. The General Case

We can study the general case of estimating forward path frequency response using an approach similar to Goodman's (ref. 6.1.) Such an approach leads us to consider the joint statistical properties of a tri-variate complex normal random series. The work involved in this is complex and considerably more lengthy than Goodman's analysis. For this reason the analysis was not attempted. Instead a special case was considered, which enables direct use of Goodman's results.

6.4.2. A Special Case

Consider the problem of estimating the forward path frequency function $H_1(\omega)$ of figure 6.1. when the feedback noise source is negligible. In this case the problem reduces to the estimation of the spectrum of a two-dimensional stationary gaussian process, and the work of Goodman can be used directly.

Defining the sample spectra of the input, output and error signals as :-

$X_i(\omega)$ = the spectrum of the i^{th} sample of $x(t)$

$Y_i(\omega)$ = the spectrum of the i^{th} sample $y(t)$

$E_i(\omega)$ = the spectrum of the i^{th} sample of $e(t)$

Now, Goodman developed sampling distributions associated with the complex regression coefficient of $Y(\omega)$ on $X(\omega)$, defined by :-

$$\hat{G}(\omega) = \frac{\sum_{i=1}^{n} X_i^* Y_i(\omega)}{\sum_{i=1}^{n} X_i^* X_i(\omega)}$$

And we are interested in the sampling distributions associated with the function :-

$$\frac{\sum_{i=1}^{n} X_i^* Y_i(\omega)}{\sum_{i=1}^{n} X_i^* E_i(\omega)} = \hat{H}_1(\omega)$$

If the noise source in the feedback path is negligible, then these two functions are related by :-

$$\hat{G}(\omega) = \frac{\hat{H}_1(\omega)}{1 + H_2(\omega)\hat{H}_1(\omega)} \qquad \ldots\ldots\ldots\ldots 6.10.$$

Therefore, in the special case of noise-free feedback, the

distributions associated with the estimator of the forward path transfer function can be obtained by a transformation of the distributions associated with overall transfer function estimate.

6.5. Confidence Regions for the Forward Path Frequency Response Estimator

6.5.1. Conformal Mapping of Goodman's Confidence Regions

Confidence statements concerning the forward path frequency response estimator can be obtained by using equation 6.10. to transform the probability density functions given by Goodman in reference 6.1. However, in this section an alternative method is employed. Using this method confidence statements for the forward path frequency response estimator are obtained by a complex conformal mapping of the confidence regions for the overall frequency response estimator. These confidence statements apply to the special case of noise-free feedback.

Define the complex variables z and w as follows :-

$$\begin{aligned} z &= x + jy \\ w &= u + jv \end{aligned} \qquad \ldots\ldots\ldots\ldots 6.11.$$

Let the point x_o, y_o in the z-plane represent the true value of the forward path frequency response function at frequency ω_1.

Defining the complex constant ρ as the inverse complex feedback fraction at frequency ω_1. We can say :-

$$\rho = \frac{1}{H_2(\omega_1)} \qquad \ldots\ldots\ldots\ldots 6.12.$$

$$= \alpha + j\beta$$

Then, at ω_1, the z-plane is mapped into the w-plane by the bilinear conformal transformation :-

$$w = \frac{z}{1 - \frac{z}{\rho}} \qquad \ldots\ldots\ldots\ldots 6.13.$$

From section 5.2.4., we can draw circles of radius r, centre x_o, y_o, in the z-plane and attach a probability P, to the event that the overall frequency response estimate will lie inside that circle. Using the transformation equation 6.13. we can draw corresponding contours in the w-plane, and attach the same probability to the event that the forward path frequency response estimate will lie inside the region mapped by the interior of the w-plane circle.

Using the properties of bilinear transformations (ref. 6.2.) we can show that the circles (r, x_o, y_o) in the z-plane map contours which are circles (R, u_c, v_c) in the w-plane. The radius and centre of the w-plane circles are given by :-

$$R = \frac{|\rho|^2 r}{(\alpha - x_o)^2 + (\beta - y_o)^2 - r^2}$$

$$u_c = \frac{x_o|\rho|^2 - \alpha(x_o^2 + y_o^2 - r^2)}{(\alpha - x_o)^2 + (\beta - y_o)^2 - r^2} \qquad \ldots\ldots\ldots 6.14.$$

$$v_c = \frac{y_o|\rho|^2 - \beta(x_o^2 + y_o^2 - r^2)}{(\alpha - x_o)^2 + (\beta - y_o)^2 - r^2}$$

The true value of the forward path frequency response function $H_1(\omega_1)$ is at u_o, v_o, and is given by :-

$$u_o = \frac{x_o|\rho|^2 - \sigma(x_o^2 + y_o^2)}{(\sigma - x_o)^2 + (\beta - y_o)^2} \qquad \ldots\ldots\ldots\ldots 6.15.$$

$$v_o = \frac{y_o|\rho|^2 - \beta(x_o^2 + y_o^2)}{(\sigma - x_o)^2 + (\beta - y_o)^2}$$

From these equations, the w-plane confidence regions are a set of eccentric circles. For small r the circles are (approximately) centred at u_o, v_o. As r increases the centre points move away from u_o, v_o. However, the circles always contain the u_o, v_o point.

Consider the area mapped in the w-plane by the interior of the z-plane circles. From figure 6.3. :-

a) The contour C_1

The interior of C_1 maps the interior of K_1. This interior-to-interior mapping continues as the radius of the circles is increased, until the contour C_2 is reached.

b) The contour C_2

The contour C_2, which has the point ρ upon it, maps [redacted] into the straight line K_2 given by :-

$$2u(\sigma(x_o^2 + y_o^2 - r^2) - x_o|\rho|^2) + 2v(\beta(x_o^2 + y_o^2 - r^2) - y_o|\rho|^2)$$

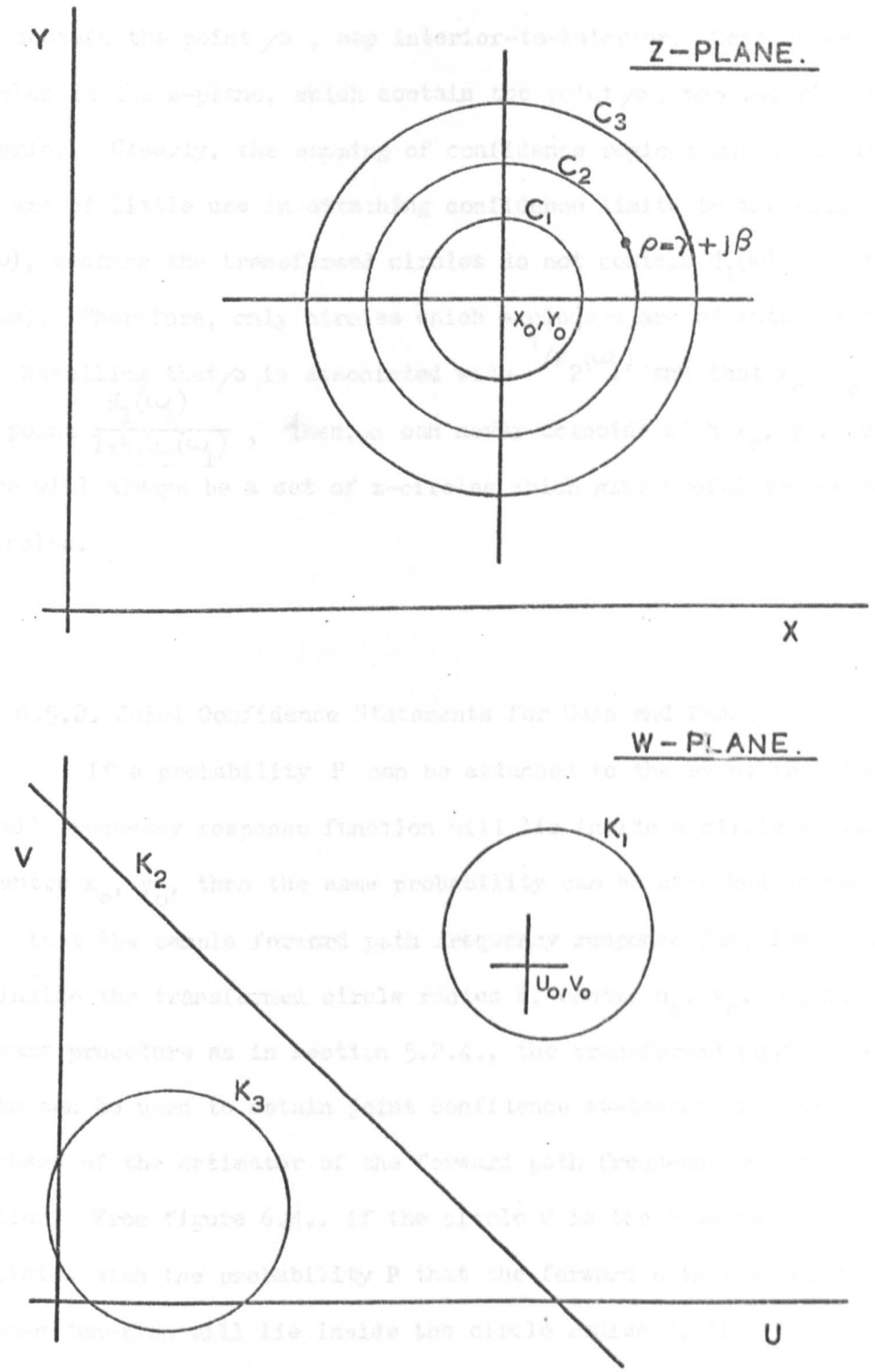
$$= |\rho|^2(r^2 - x_o^2 + y_o^2) \qquad \ldots\ldots\ldots\ldots 6.16.$$

And the interior of C_2 maps into a section of the w-plane.

c) The contour C_3

The interior of C_3 maps, the exterior of the corresponding

FIGURE 6.3. THE MAPPING OF THE Z-PLANE ON TO THE W-PLANE.



contour K_3 in the w-plane. The point u_o, v_o, lies outside K_3.

Summarising, confidence circles in the z-plane, which do not contain the point ρ, map interior-to-interior. Confidence circles in the z-plane, which contain the point ρ, map interior-to-exterior. Clearly, the mapping of confidence regions which contain ρ are of little use in attaching confidence limits to the estimator $\hat{H}_1(\omega)$, because the transformed circles do not contain $H_1(\omega)$ (the true value). Therefore, only circles which exclude ρ are of interest to us. Recalling that ρ is associated with $1/H_2(\omega_1)$ and that x_o, y_o is the point $\frac{H_1(\omega_1)}{1+H_1H_2(\omega_1)}$, then, ρ can never coincide with x_o, y_o, so there will always be a set of z-circles which give useful transformed w-circles.

6.5.2. Joint Confidence Statements for Gain and Phase

If a probability P can be attached to the event that the overall frequency response function will lie inside a circle radius r, centre x_o, y_o, then the same probability can be attached to the event that the sample forward path frequency response function will lie inside the transformed circle radius R, centre u_c, v_c. Using the same procedure as in section 5.2.4., the transformed confidence region can be used to obtain joint confidence statements for the gain and phase of the estimator of the forward path frequency response function. From figure 6.4., if the circle C is the transformed circle associated with the probability P that the forward path frequency response function will lie inside the circle radius R, then the

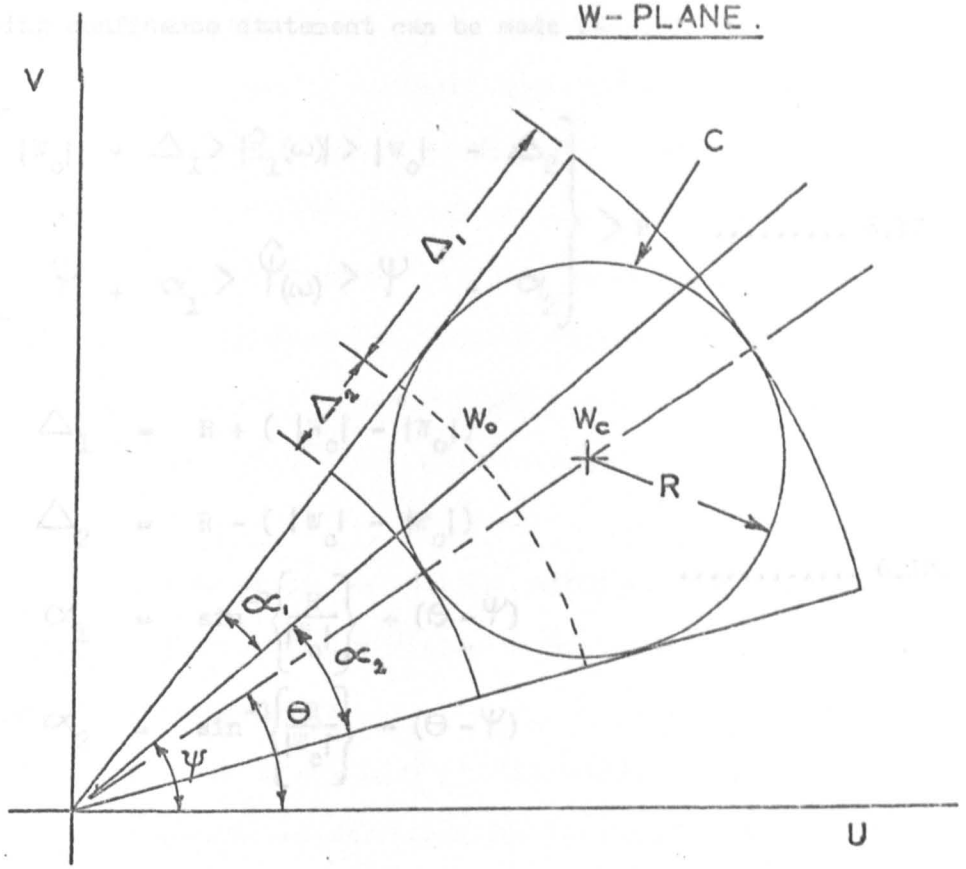


FIGURE 6.4. CONFIDENCE REGIONS FOR FORWARD PATH FREQUENCY RESPONSE FUNCTION ESTIMATES.

following confidence statement can be made :-

$$\text{Prob}\left\{\begin{array}{l} |W_o| + \Delta_1 > |\hat{H}_1(\omega)| > |W_o| - \Delta_2 \\ \\ \Psi + \alpha_1 > \hat{\Psi}(\omega) > \Psi - \alpha_2 \end{array}\right\} > P \qquad \ldots\ldots\ldots 6.17.$$

where
$$\Delta_1 = R + (|W_c| - |W_o|)$$
$$\Delta_2 = R - (|W_c| - |W_o|)$$
$$\alpha_1 = \sin^{-1}\left\{\frac{R}{|W_c|}\right\} + (\Theta - \Psi) \qquad \ldots\ldots\ldots\ldots 6.18.$$
$$\alpha_2 = \sin^{-1}\left\{\frac{R}{|W_c|}\right\} - (\Theta - \Psi)$$

R is defined by equation 6.14. as the radius of the transformed circle, Also :-

$|W_o|$ = the true gain $|H_1(\omega_1)|$

Ψ = the true phase Arg $H_1(\omega_1)$

$$|W_c| = \left\{u_c^2 + v_c^2\right\}^{\frac{1}{2}}$$

From expression 6.17. the approximate 100P% simultaneous confidence intervals for the forward path gain and phase are :-

$$|\hat{H}_1(\omega)| + \Delta_1 > |H_1(\omega)| > H_1(\omega) - \Delta_2$$

$$\hat{\Psi}(\omega) + \alpha_1 > \Psi(\omega) > \Psi(\omega) - \alpha_2$$

As an example of the use of these confidence statements consider the simple case of a closed loop systems with forward path transfer function $\frac{1}{1+j\omega}$, and noise-free unity feedback. Consider the 75% confidence intervals for the overall, and forward path, frequency response functions at $\omega = 1$ radian/second. If $n = 100$ and $K^2_{xy}(1) = 0.5$, then from the previous chapter the 75% confidence intervals for the overall gain and phase can be shown to be :- 0.4496 ± 0.053, and $-26.6^\circ \pm 6.8^\circ$ respectively.

These figures can be used to obtain the 75% confidence bands for the forward path estimator at $\omega = 1$, but first we must determine the appropriate values for x_o, y_o, u_o, v_o, u_c, v_c. The overall frequency response at $\omega = 1$ is $\frac{1}{2+j}$, therefore, $x_o = \frac{2}{5}$, and $y_o = \frac{1}{5}$. The forward path frequency response function at $\omega = 1$ is $\frac{1}{1+j}$, therefore, $u_o = \frac{1}{2}$ and $v_o = -\frac{1}{2}$. Also, because the radius of the confidence circle for the overall transfer function is small, $u_c \approx u_o$ and $v_c \approx v_o$. Since the feedback is unity, then from equation 6.12, $\rho = 1$

Substituting these figures in the expression for R, the radius of the 75% confidence circle for the forward path frequency response is approximately 0.132. Hence, the 75% confidence intervals for the gain and phase of the forward path at $\omega = 1$ are :- 0.7117 ± 0.132, and $-45^\circ \pm 10.7^\circ$, respectively.

6.5.3. Experimental Justification of the Confidence Regions

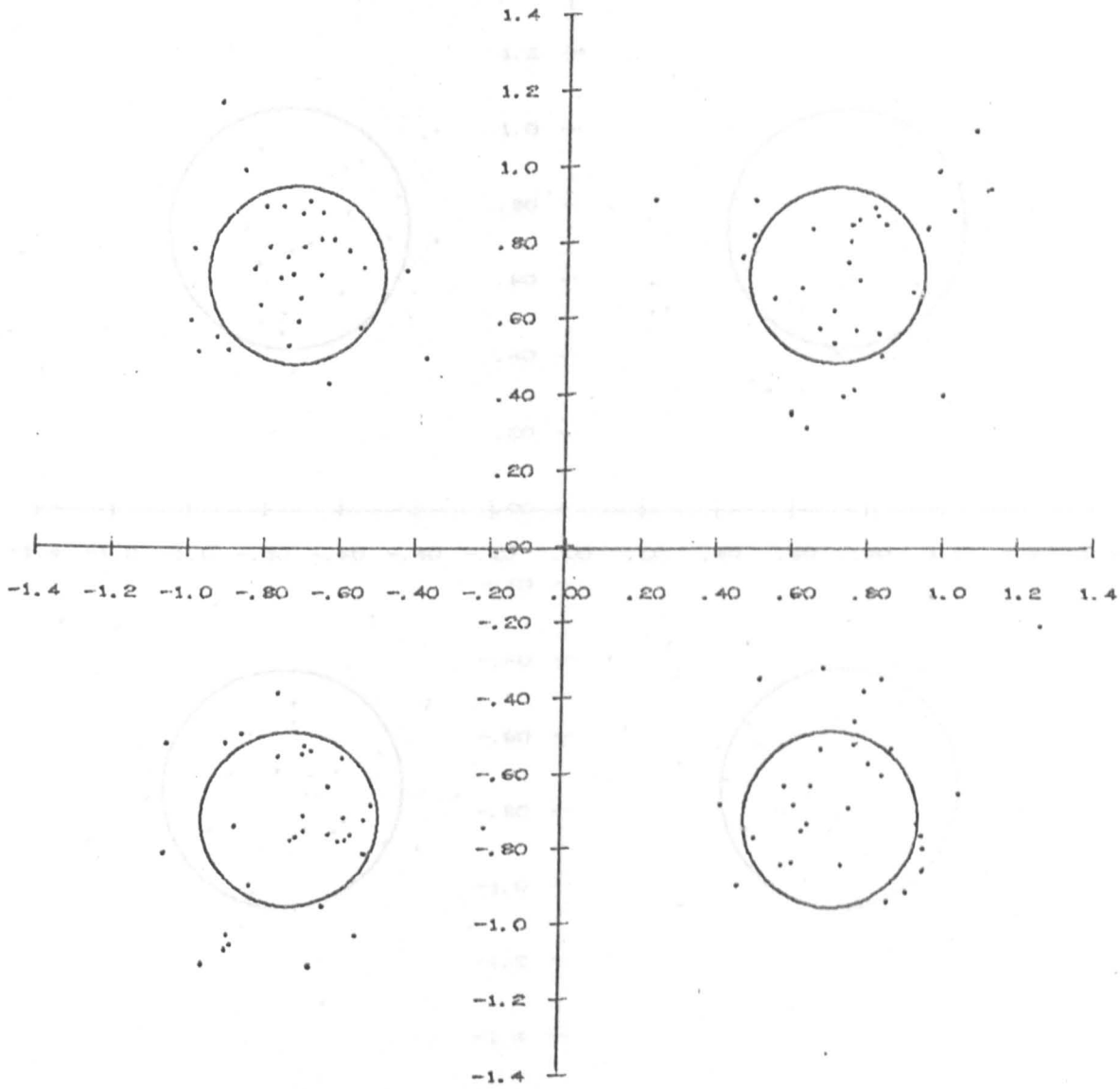
To test the validity of the transformed confidence circles, a typical system was simulated using a digital computer. The system

GRAPH 6.1.

SCATTER DIAGRAM OF FORWARD PATH FREQUENCY

RESPONSE ESTIMATES. CONTOURS CONTAIN 50% CONFIDENCE

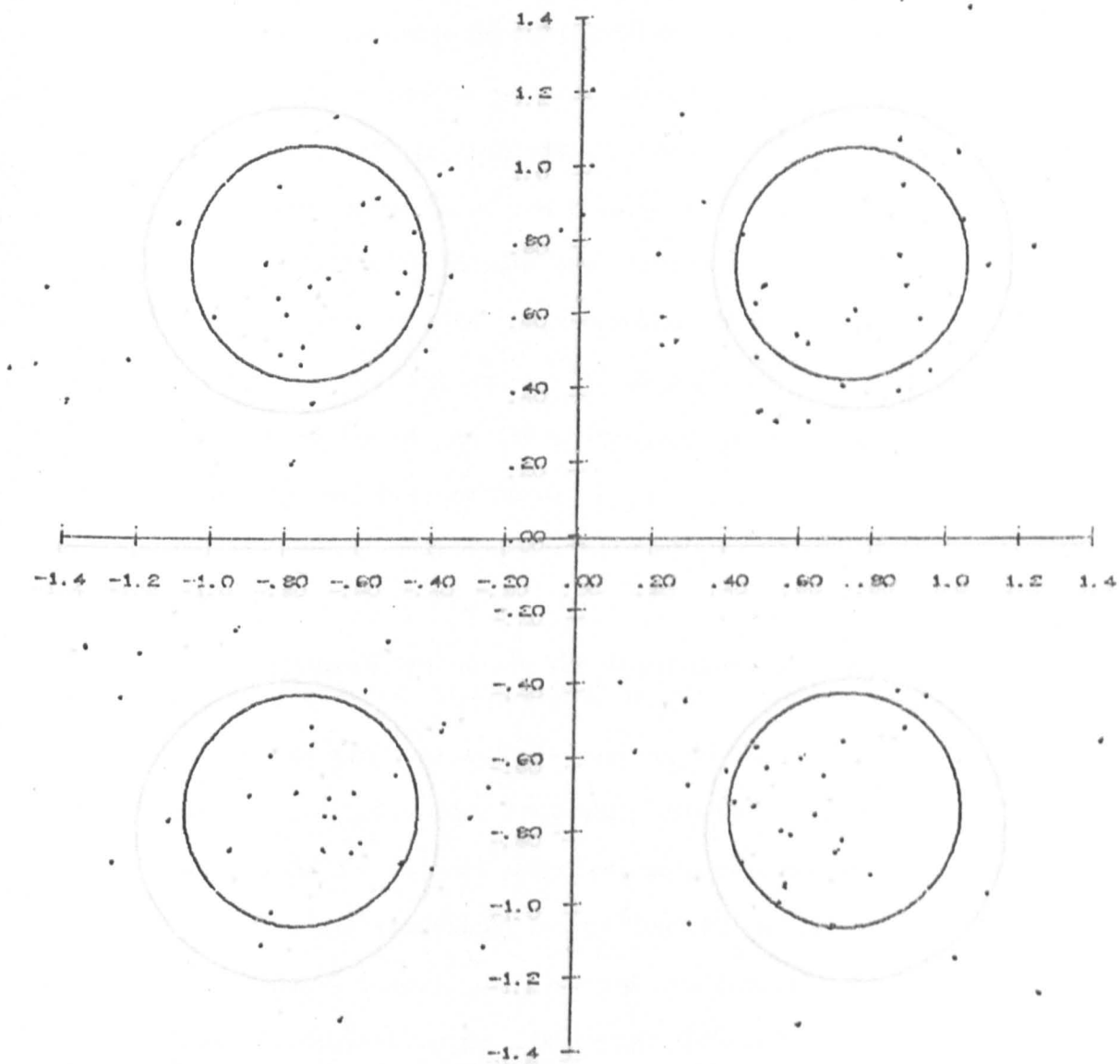
REGIONS FROM TRANSFORMED GOODMANS DISTRIBUTION (B =0.5)



GRAPH 6.2.
SCATTER DIAGRAM OF FORWARD PATH FREQUENCY

RESPONSE ESTIMATES. CONTOURS CONTAIN 50% CONFIDENCE

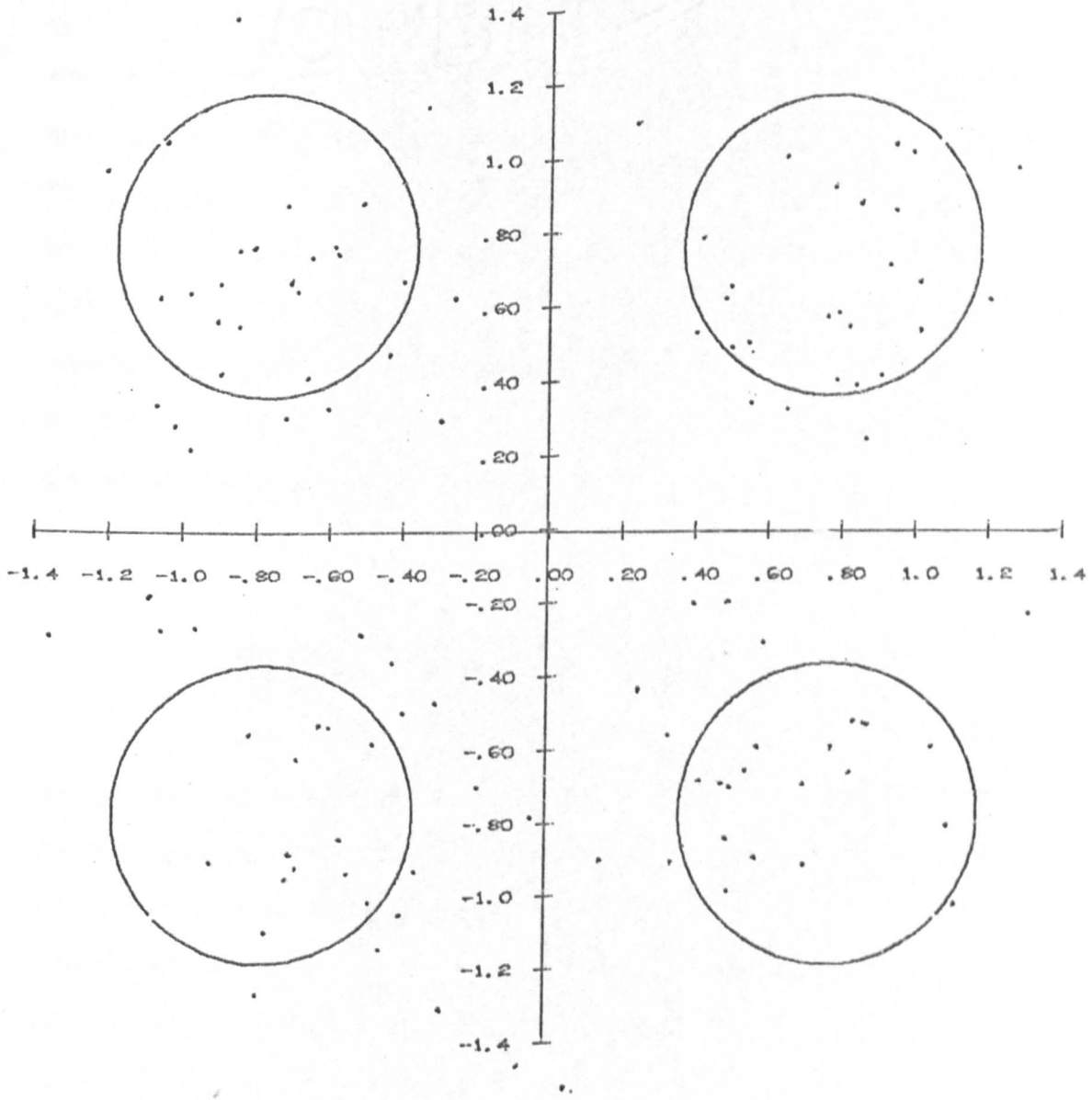
REGIONS FROM TRANSFORMED GOODMANS DISTRIBUTION (B = 1)



GRAPH 6.3.
SCATTER DIAGRAM OF FORWARD PATH FREQUENCY

RESPONSE ESTIMATES. CONTOURS CONTAIN 50% CONFIDENCE

REGIONS FROM TRANSFORMED GOODMANS DISTRIBUTION (B = 1.5)



was a unity forward path transfer function corrupted by noise n(t), and with a feedback transfer function B. The input signal x(t) and the noise n(t) are independent noise sources, but with identical autospectra. The amplitudes of x(t), and n(t) were chosen so that the squared coherency between input and output was 0.75.

One hundred and twenty eight independent estimates of $H_1(\omega)$ were made, each estimate having 10 degrees of freedom. A scattergram of the estimates was then constructed and 50% transformed confidence circles superimposed upon the scattergram. Typical results, for various B, are shown in graphs 6.1. to 6.3. In these graphs the estimates, and confidence circles are plotted in sets of 32 at 90^{o} intervals on the unit circle. This was done to make counting the experimental confidence region population easier. In all cases tried the actual populations of the 50% confidence circles were close to the expected population of sixteen.

6.5.4. Confidence Statements for the General Case when the Feedback Path is Subjected to Random Disturbances

The confidence statements derived in this section apply in the special case of noise-free feedback. However, from section 6.3., the variability of the forward path frequency response estimate is (to a first order approximation) independent of the feedback noise. Therefore, in the absense of more general confidence statements, we are justified in suggesting that, although the confidence statements of equation 6.17. are not strictly valid, they may be used to obtain approximate confidence intervals in the general case of noisy feedback.

This technique is only tentatively suggested, and in order that its validity may be tested a more thorough examination of the variability of the estimator $\frac{\bar{C}xy}{\bar{C}xe}(\omega)$ is proposed in section 8.2.

REFERENCES

6.1. Goodman, N.R., "On the Joint Estimation of the Spectra, Co-Spectrum and Quadrature-Spectrum of a Two-Dimensional Stationary Gaussian Process", Scientific Paper No. 10, New York Univ., 1957.

6.2. Speigel, M.K., "Complex Variables", Schaum 1964, pp. 200-232.

7. EXPERIMENTAL WORK

7.1. Introduction

This chapter presents some experimental results obtained using the real-time digital spectral analysis programmes. The results are presented to illustrate some theoretical points made in chapter 4 and 5, and to show some applications of the real-time spectral technique.

The first two sections are concerned with demonstrating sources of error in the real-time spectral analysis techniques. One section deals with bias due to data windowing, and the other with bias and variance due to aliasing. Normal variability errors are not illustrated. This is because in most cases the programmes can be left running until the variance of the estimates is negligible. In the cases where this is not possible the confidence statements of chapters 4 and 5 may be used.

The last three sections present applications of the real-time digital spectral analysis programmes. The first of these applications is in the autospectral analysis of acoustic data, and the second is in the auto, and cross-spectral analysis of vibration data. Finally, the identification of an internal combustion engine/dynamometer test rig is described.

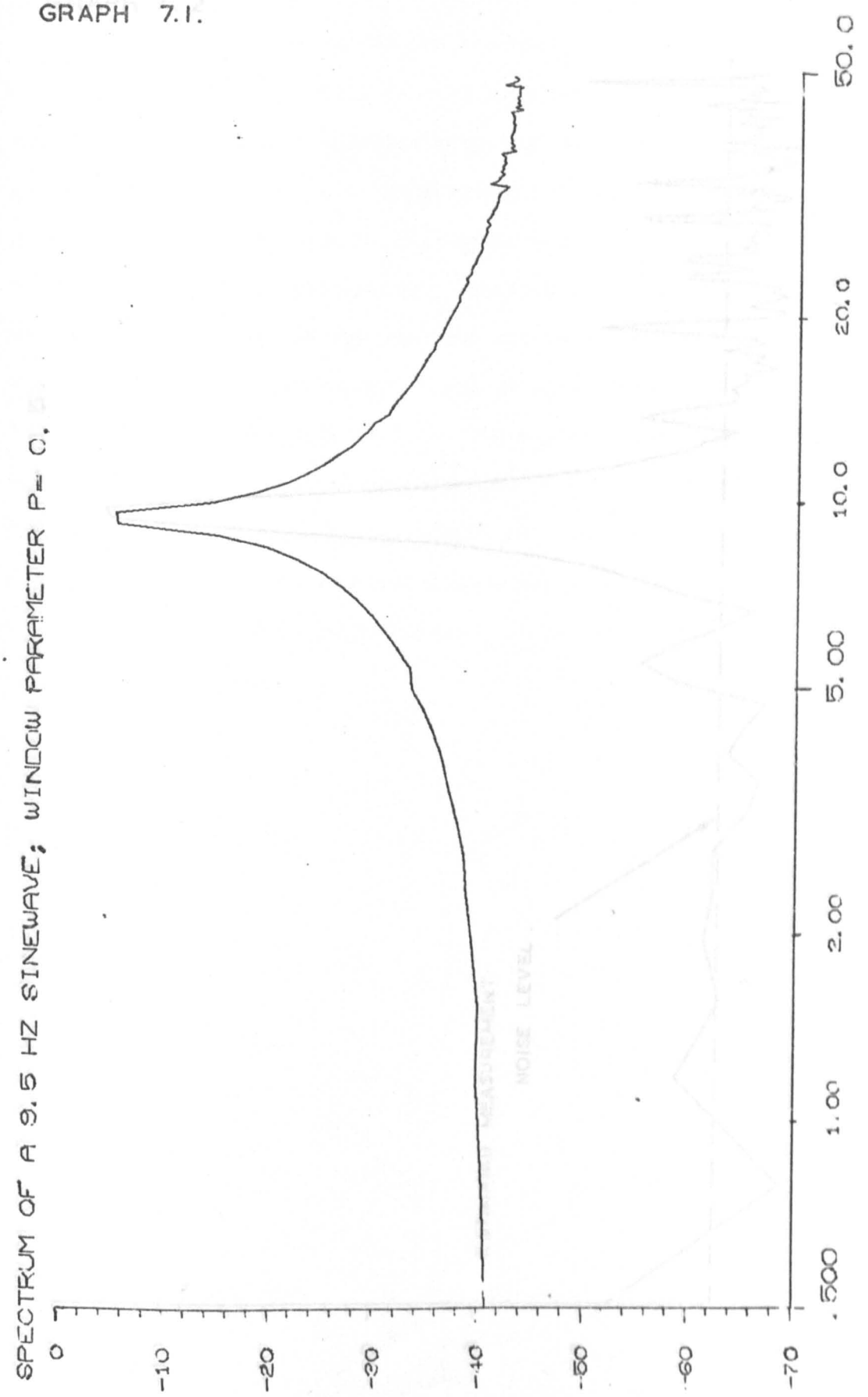
7.2. Bias Errors due to Data Windowing

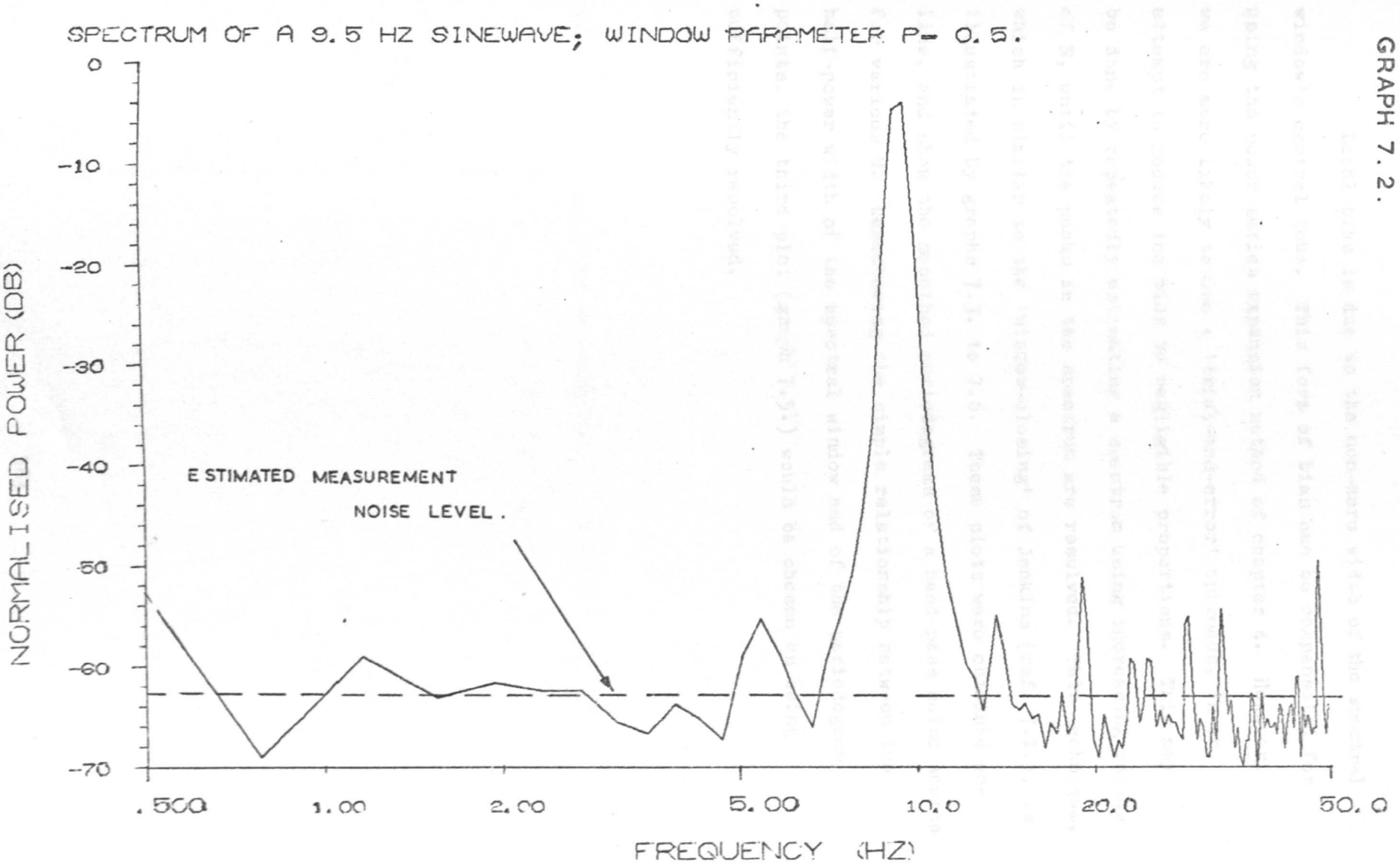
As explained in chapter 4 using a data window, $1(t)$, of finite duration causes two types of bias error. These are local bias error, and long-term bias error (leakage). These errors are now discussed as they occur in the real-time autospectrum estimation programme.

Spectral leakage is the long-term bias which causes spectral components to be swamped by leakage from large peaks in the autospectrum. Graph 7.1. demonstrates leakage as it effects the measured spectrum of a sine-wave. The data window used in this measurement was the Generalised Hanning window with $p = 0$. This window has slowly decaying side lobes (graph 4.6.), and is the same as the spectrum of a simple rectangular data window. The spectrum measured in this way falls away slowly, and so small amplitude peaks near 10 Hz could easily be obscured. Theoretically the spectrum should decay as ω^{-4}, however aliasing prevents it reaching this rate.

The reduction of leakage caused by increasing p is shown by graph 7.2. This shows the spectrum of the same sinusoid, only measured using the Generalised Hanning window (parameter $p = 0.5$.). As can be seen the spectrum falls away very rapidly, and soon becomes lost in the measurement noise. The estimated noise level (shown dotted) was calculated using the expression given in section 4.4.1.

GRAPH 7.1.

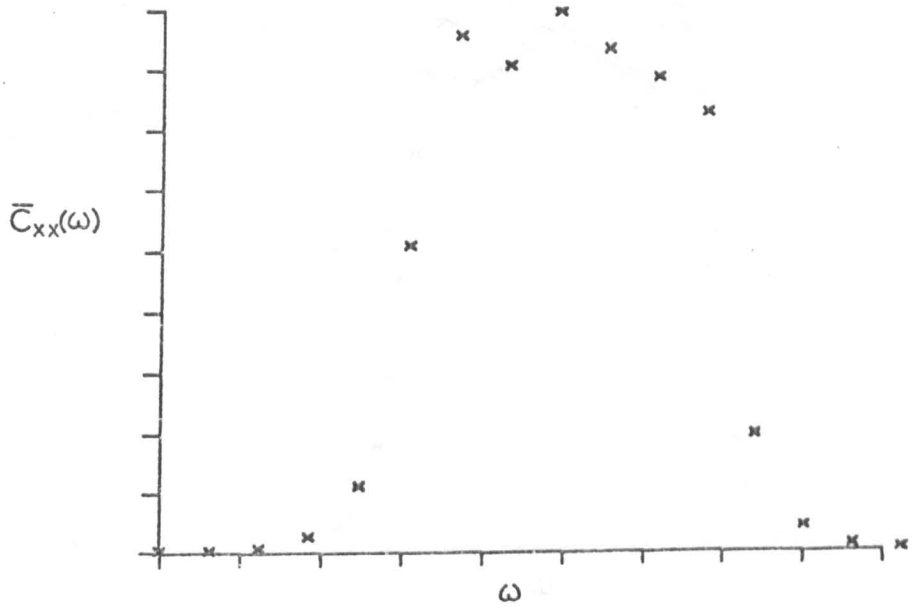

SPECTRUM OF A 9.5 HZ SINEWAVE; WINDOW PARAMETER P= 0.
NORMALISED POWER (DB)
0
-10
-20
-30
-40
-50
-60
-70
.500
1.00
2.00
5.00
10.0
20.0
50.0
FREQUENCY (HZ)


SPECTRUM OF A 9.5 HZ SINEWAVE; WINDOW PARAMETER P= 0.5.
NORMALISED POWER (DB)
0
-10
-20
-30
-40
-50
-60
-70
ESTIMATED MEASUREMENT
NOISE LEVEL.
.500
1.00
2.00
5.00
10.0
20.0
50.0
FREQUENCY (HZ)

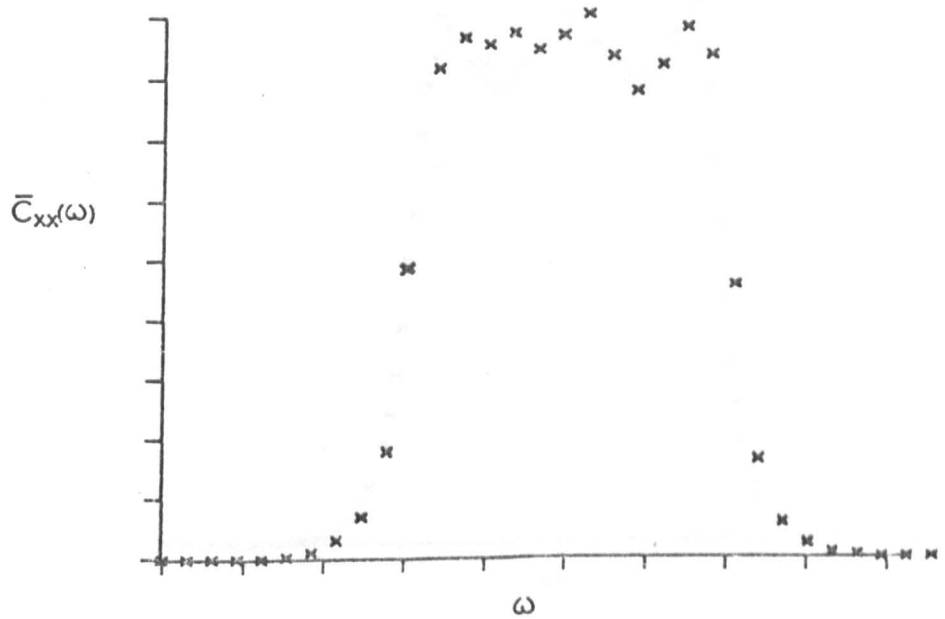
GRAPH 7.2.

Local bias is due to the non-zero width of the spectral window's central lobe. This form of bias can be compensated for using the power series expansion method of chapter 4. However, we are more likely to use a 'trial-and-error' approach, in an attempt to reduce the bias to negligible proportions. This may be done by repeatedly estimating a spectrum using increasing values of N, until the peaks in the spectrum are resolved. This technique, which is similar to the 'window-closing' of Jenkins (ref. 7.1.), is illustrated by graphs 7.3. to 7.6. These plots were obtained on-line, and show the smoothed periodograms of a band-pass noise source for various N. Remembering the simple relationship between the half-power width of the spectral window and of the periodogram points, the third plot (graph 7.5.) would be chosen as being sufficiently resolved.

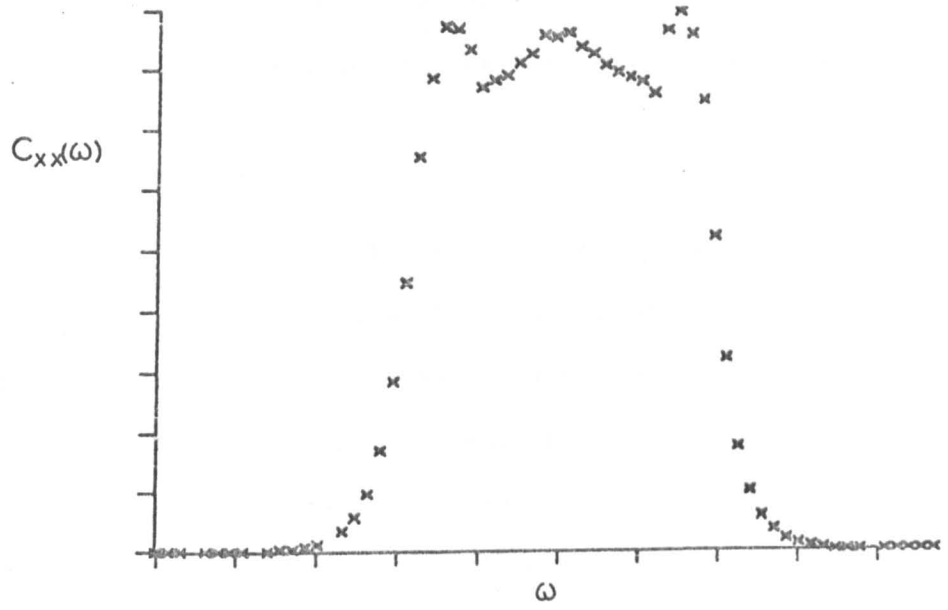
GRAPH 7.3. SMOOTHED PERIODOGRAM. N = 32.



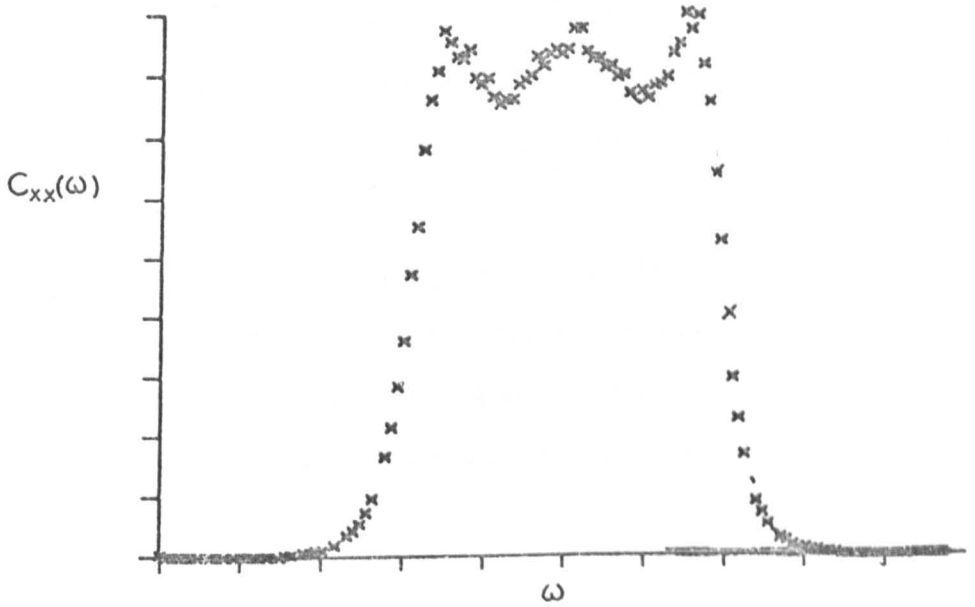
GRAPH 7.4. SMOOTHED PERIODOGRAM. N = 64.



GRAPH 7.5. SMOOTHED PERIODOGRAM. N = 128.



GRAPH 7.6. SMOOTHED PERIODOGRAM. N = 256.



7.3. Aliasing Errors

This section uses the real-time frequency response estimation programme to illustrate the effects of aliasing upon digital frequency response estimates. The results presented here are practical confirmation of the theoretical conclusions drawn in Chapter 5.

The system identified in this section is a time advance. This is physically unrealisable, but may be easily simulated using the time delay inherent in the GEC 90/2 computer interface. The interface analogue to digital converter takes 9ms to change from reading one analogue input to read another. Therefore, there is in the frequency response estimation programme a time delay between sampling the input of the system being studied and the output. To compensate for this delay another delay of equal duration would have to be incorporated in the system under test. Therefore, when the system under test is a unity gain transfer function, the frequency response estimation programme believes it to be a time advance of 9ms. A time advance can thus be simulated by connecting the test signal to the input and output samplers at the computer interface.

In all the results given here the test signal used was a laboratory white noise source. The noise was low-pass filtered using a variable cutoff point, fourth order, Butterworth filter. This filter fulfilled the purpose of the guard filters in figure 5.3.

7.3.1. Bias

Graphs 7.7. and 7.8. show the measured frequency response function and coherency in the absense of noise. The continuous curves are the measured values, and the dotted curves are the theoretical, expected, curves calculated using results of section 5.3.3.

The sampling frequency used for these and all other results, was 50Hz. The maximum frequency scale marking, 25 Hz, is the spectral folding frequency. The filter cutoff frequency used to obtain graphs 7.7, 7.8, was 15 Hz, this corresponds to the figure of 0.3 (f_s) recommended in chapter 5. These estimates are therefore the best that can be expected using this particular guard filter.

The detrimental effect of increasing the guard filter cut-off frequency is shown by graphs 7.9, and 7.10. These show the measured frequency response and squared coherency of the time advance, when the filter cut-off frequency is 0.4 f_s(20 Hz).

In theory the cut-off frequency which minimises alias bias error for all low pass filter approximations is zero. In practice however chosing a low cut-off frequency will cause other error sources to become significant.

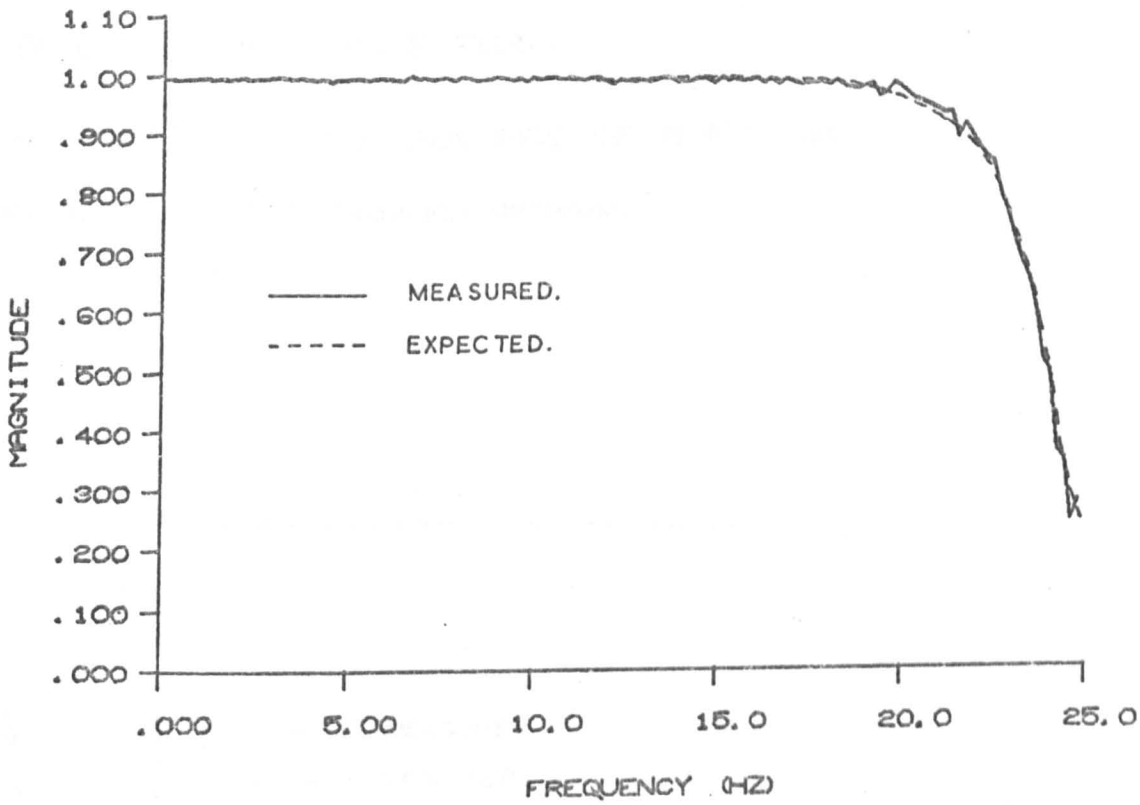
The first of these errors is spectral leakage. If a low cutoff frequency is used the spectra will have most of their power

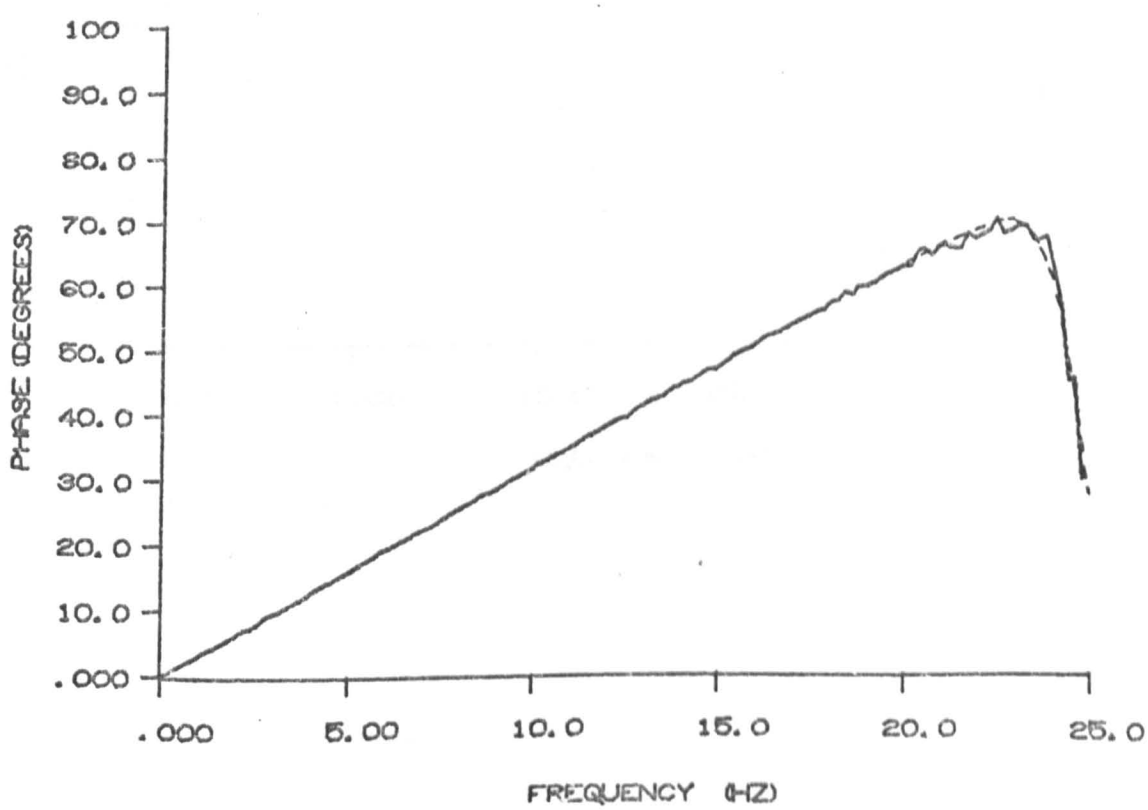
GRAPH 7.7.

FREQUENCY RESPONSE ESTIMATE FOR:-

A SIMULATED TIME ADVANCE OF 9 MS. BIAS DUE TO ALIASING;

GUARD FILTER CUTOFF FREQUENCY OPTIMUM.



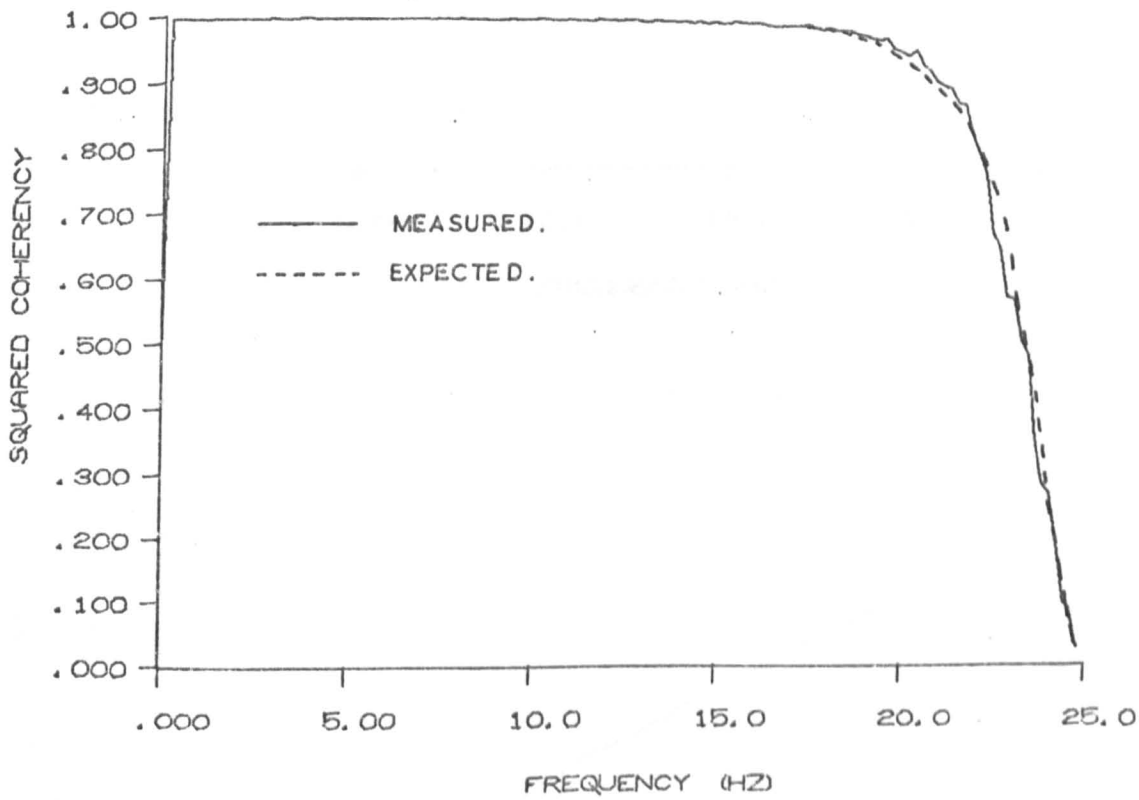


GRAPH 7.8.

COHERENCY ESTIMATE FOR:-

A SIMULATED TIME ADVANCE OF 9 MS. BIAS DUE TO ALIASING;

GUARD FILTER CUTOFF FREQUENCY OPTIMUM.

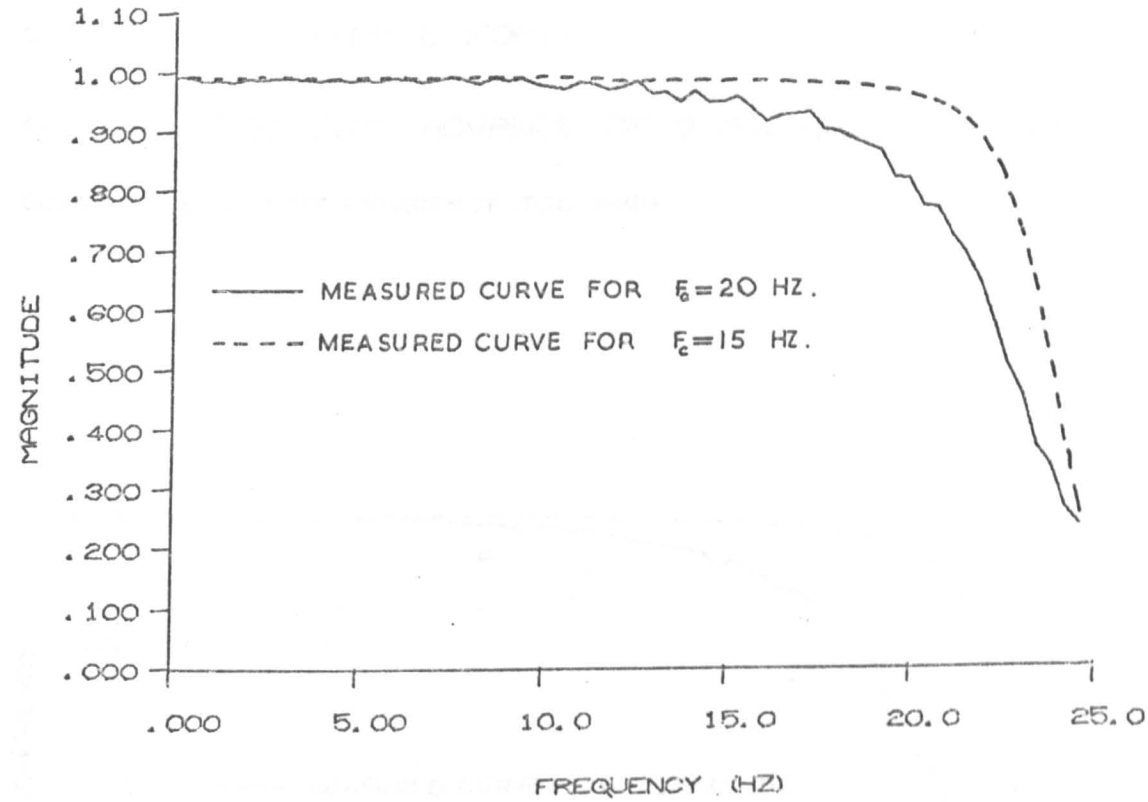


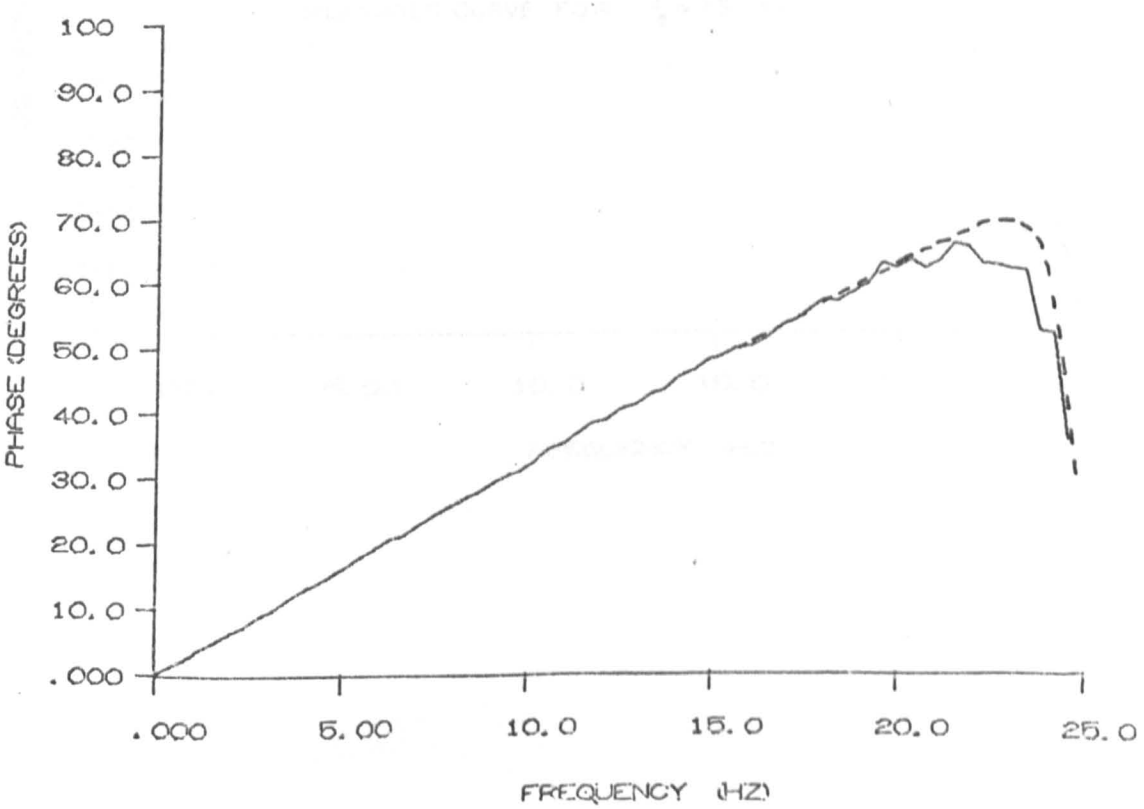
GRAPH 7.9.

FREQUENCY RESPONSE ESTIMATE FOR:-

A SIMULATED TIME ADVANCE OF 9 MS. BIAS DUE TO ALIASING;

GUARD FILTER CUTOFF FREQUENCY TOO HIGH.

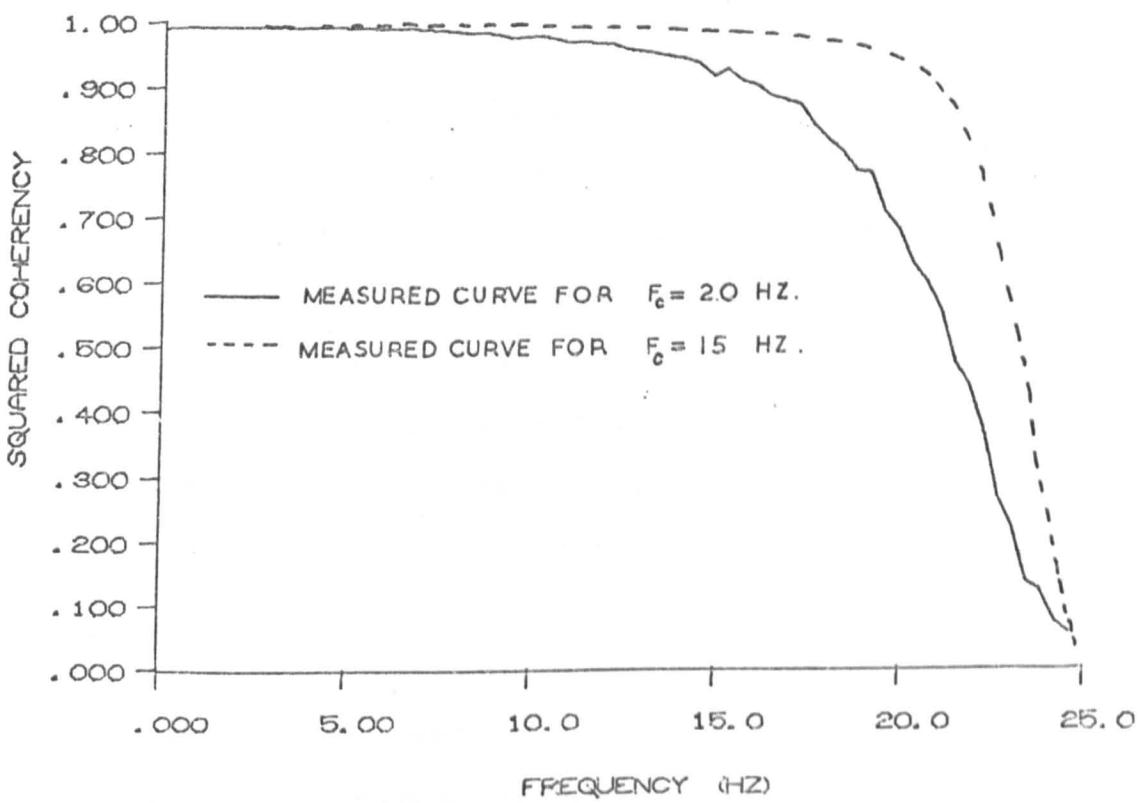




GRAPH 7.10.

COHERENCY ESTIMATE FOR:-

A SIMULATED TIME ADVANCE OF 9 MS. BIAS DUE TO ALIASING; GUARD FILTER CUTOFF FREQUENCY TOO HIGH.



concentrated at low frequencies. If the data window used has a low asymptotic decay rate, then spectral leakage from the low frequency components may overwhelm the true high frequency spectral components. Specifically, this form of error will occur when the asymptotic decay rate of the spectral window is less than that of the guard filter. Graphs 7.11, and 7.12, illustrate such a situation. The cutoff frequency used here was 0.1 f_s(5 Hz), and the Generalised Hanning window with p = 0, was used to modify the data linearly. When p = 0, the spectral window has a low decay rate (see graph 4.6.), and the leakage soon swamps the true spectra. Under these conditions the measured frequency response, and coherency tend to unity as the frequency increases.

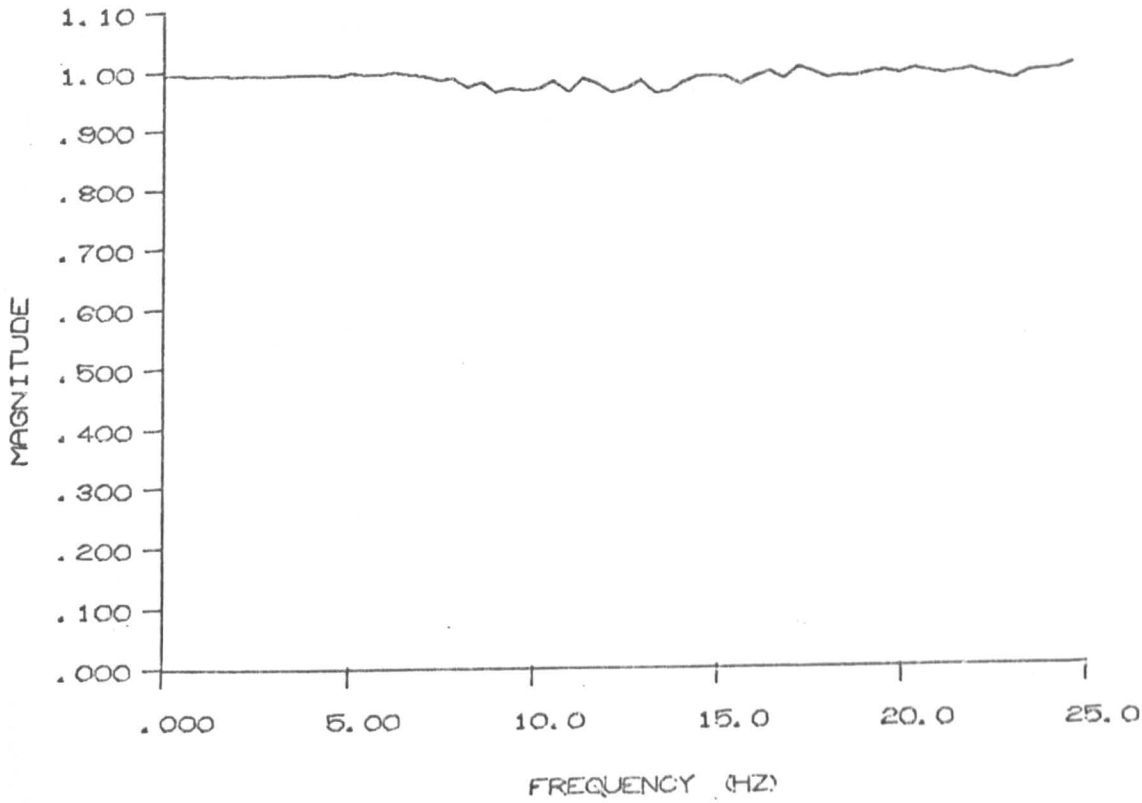
When the spectral leakage is negligible, the measurement noise (described in chapter 5) can distort the estimates. At the higher frequencies the autospectrum of the filtered test signal may become small compared with the measurement noise autospectrum. Under these conditions the expected values of the gain and coherency will tend to zero as frequency increases. The expectation of the phase estimate will be unaltered, but an increase in variability may be expected because of non-zero correlation between measurement noise sources over a finite sample.

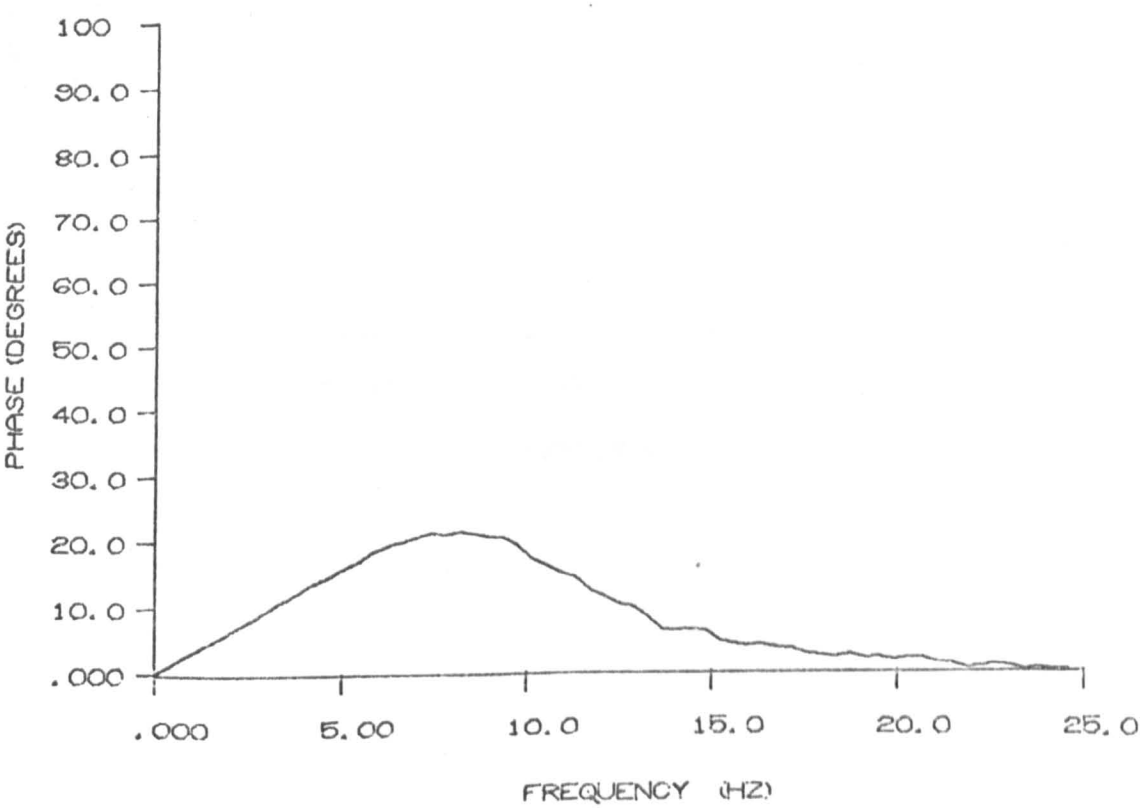
This type of distortion is shown by graphs 7.13, and 7.14. The leakage between estimates was supressed by using p = 0.5 in the Generalised Hanning window. The guard filter cutoff frequency used was 0.1 f_s, (5 Hz).

GRAPH 7.11.

FREQUENCY RESPONSE ESTIMATE FOR:-

A SIMULATED TIME ADVANCE OF 9 MS. BIAS DUE TO LEAKAGE; GUARD FILTER CUTOFF FREQUENCY TOO LOW.

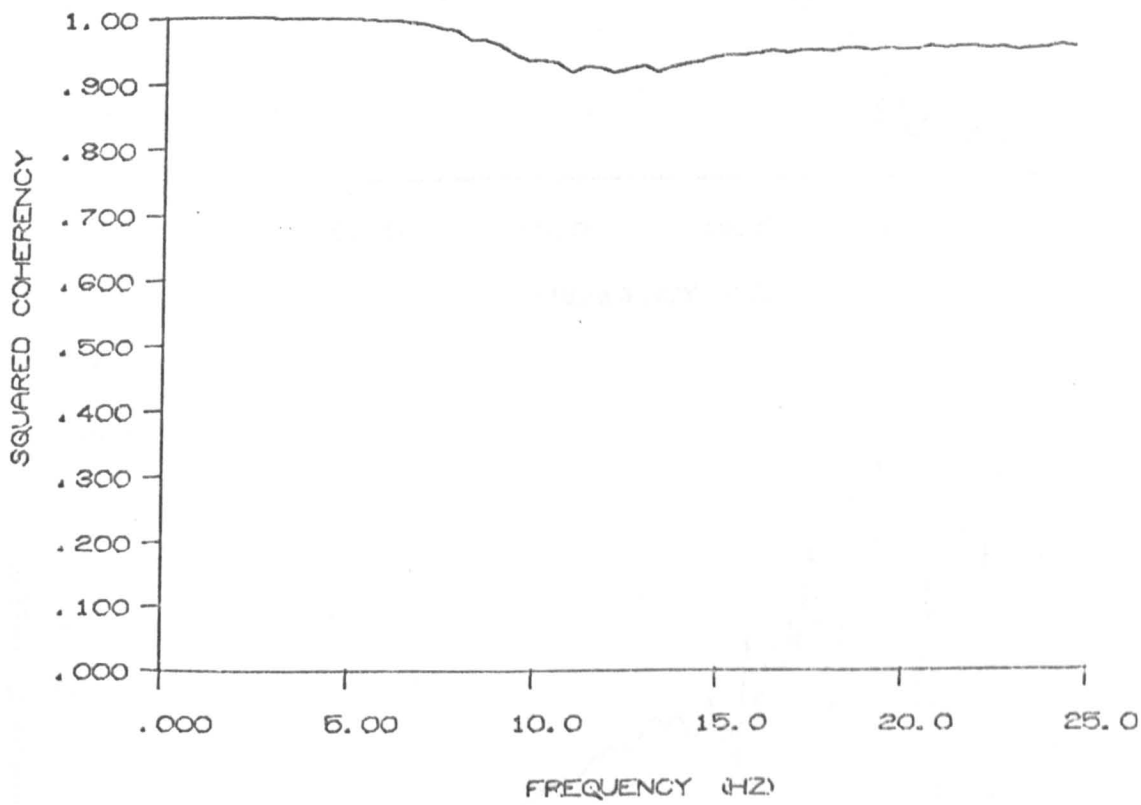




GRAPH 7.12.

COHERENCY ESTIMATE FOR:-

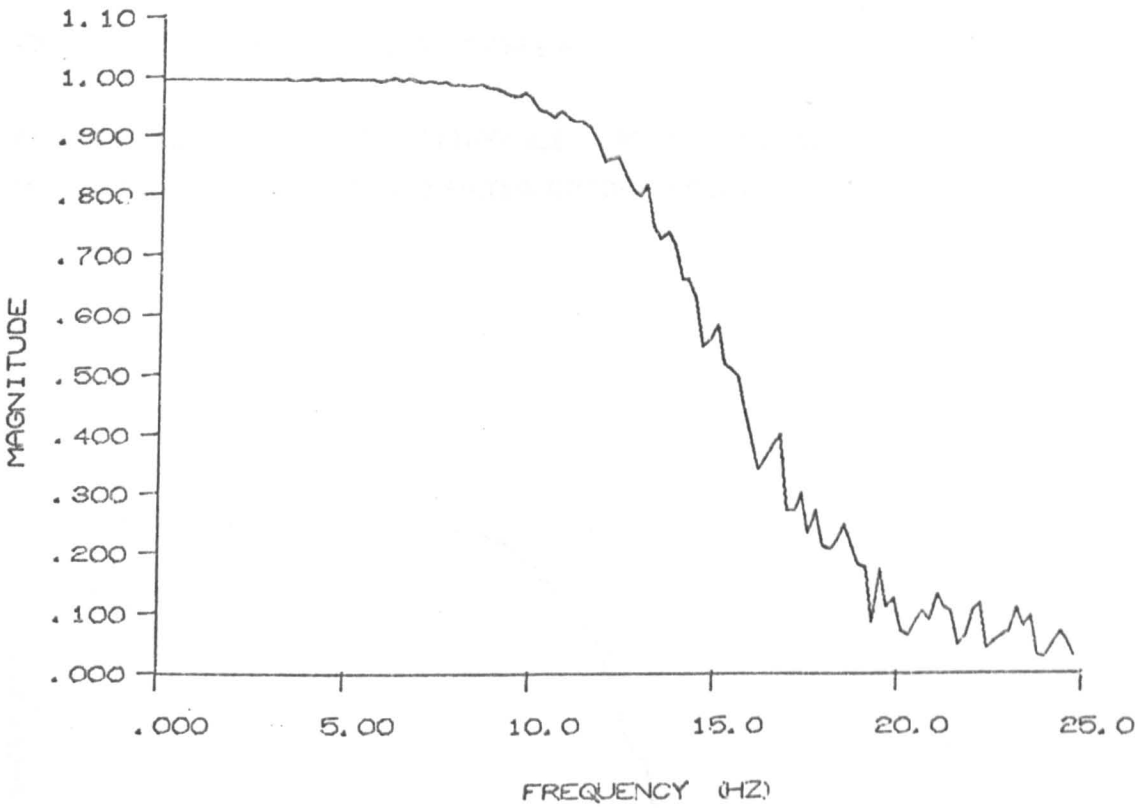
A SIMULATED TIME ADVANCE OF 9 MS. BIAS DUE TO LEAKAGE; GUARD FILTER CUTOFF FREQUENCY TOO LOW.

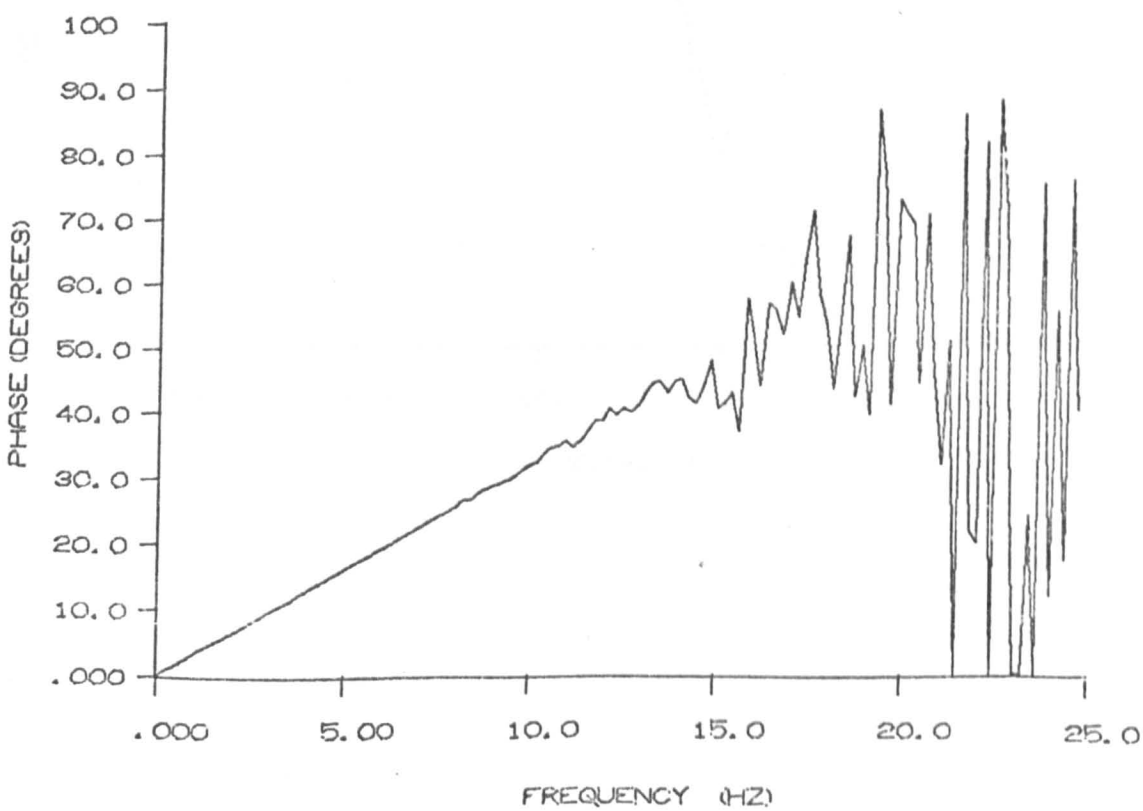


GRAPH 7.13.

FREQUENCY RESPONSE ESTIMATE FOR:-

A SIMULATED TIME ADVANCE OF 9 MS. BIAS DUE TO MEASUREMENT NOISE; GUARD FILTER CUTOFF FREQUENCY TOO LOW.

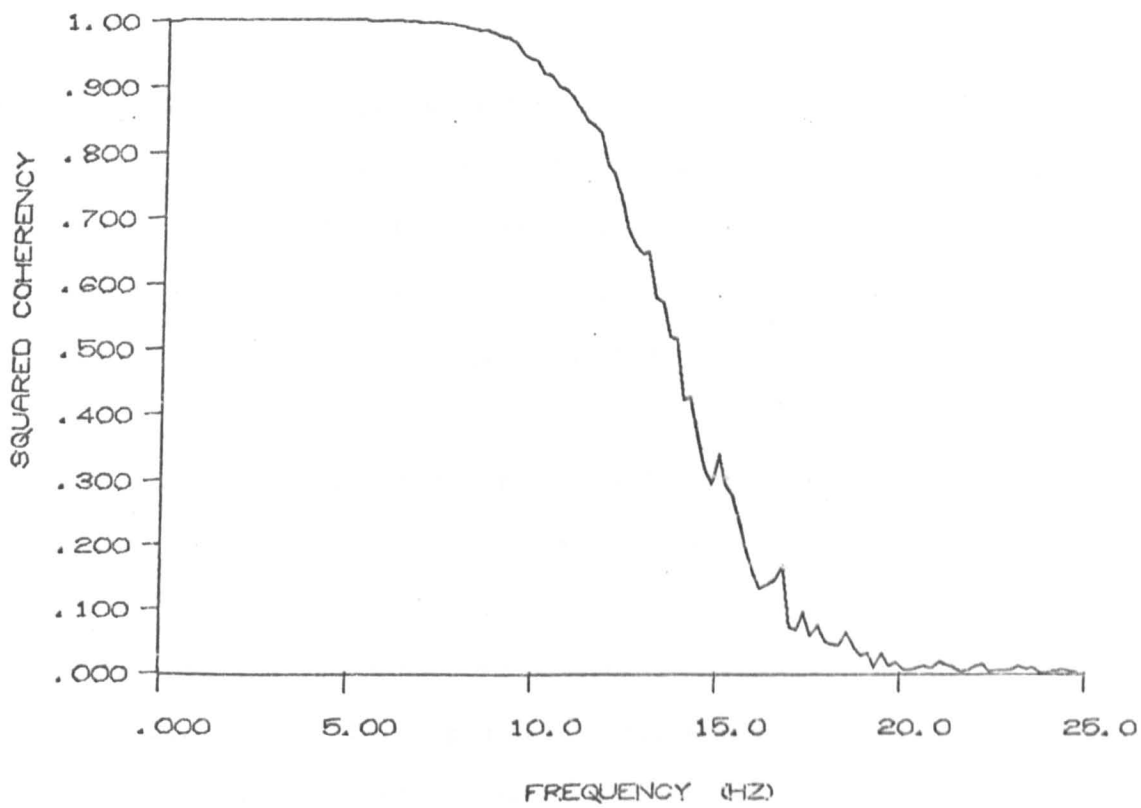




GRAPH 7.14.

COHERENCY ESTIMATE FOR:-

A SIMULATED TIME ADVANCE OF 9 MS. BIAS DUE TO MEASUREMENT NOISE; GUARD FILTER CUTOFF FREQUENCY TOO LOW.



7.3.2. Variance

To indicate the effect of aliasing upon the variance of frequency response estimates, a small noise component was added to the output of the simulated time advance. The noise signal was filtered in the same way as the test signal, so that the coherency between input signal and output signal was constant. In the absence of aliasing, the variance of the gain and phase estimates is constant with frequency.

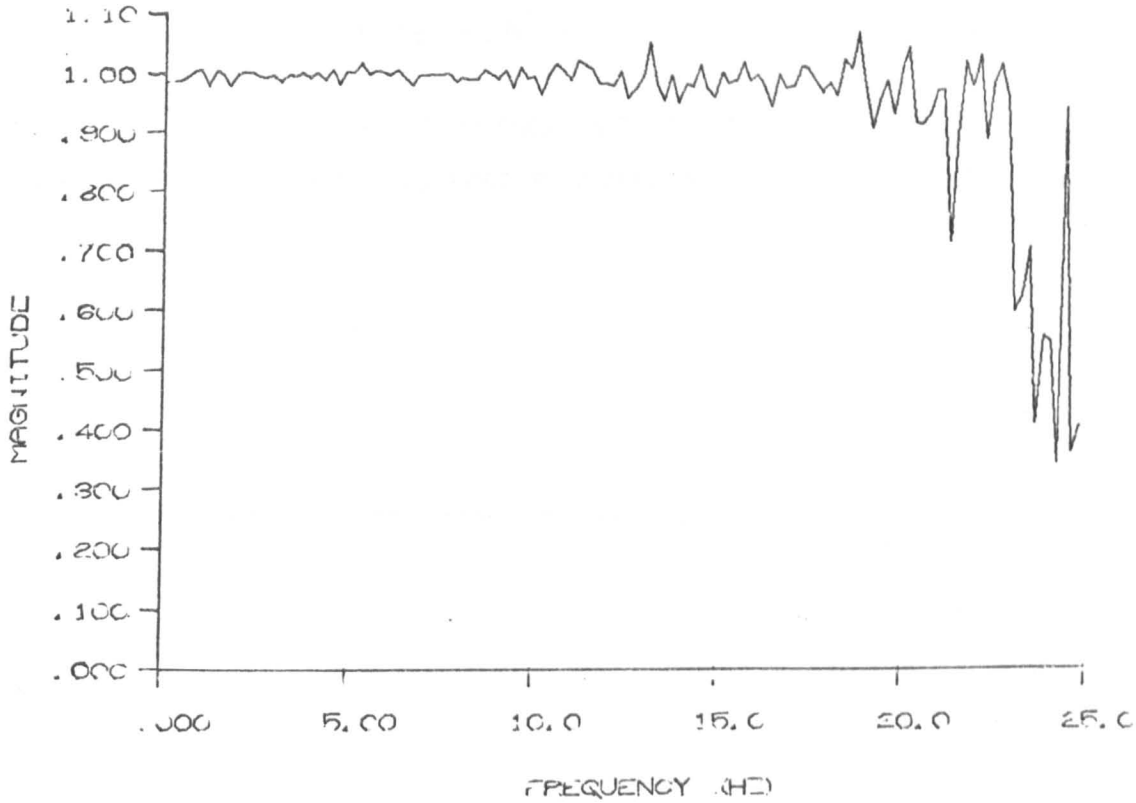
The estimated gain and phase are shown by graphs 7.15. and 7.16. As predicted the variance of the estimates increases greatly as the folding frequency is approached. This occurs because the cross-spectrum near the folding frequency is predominantly imaginary. When the cross-spectrum is real at the folding frequency the variance of the estimates is not inflated. This point is illustrated by graphs 7.17, 7.18. These curves are the estimated gain, phase and coherency of a time delay. The delay was simulated using a P.R.B.S. generator.

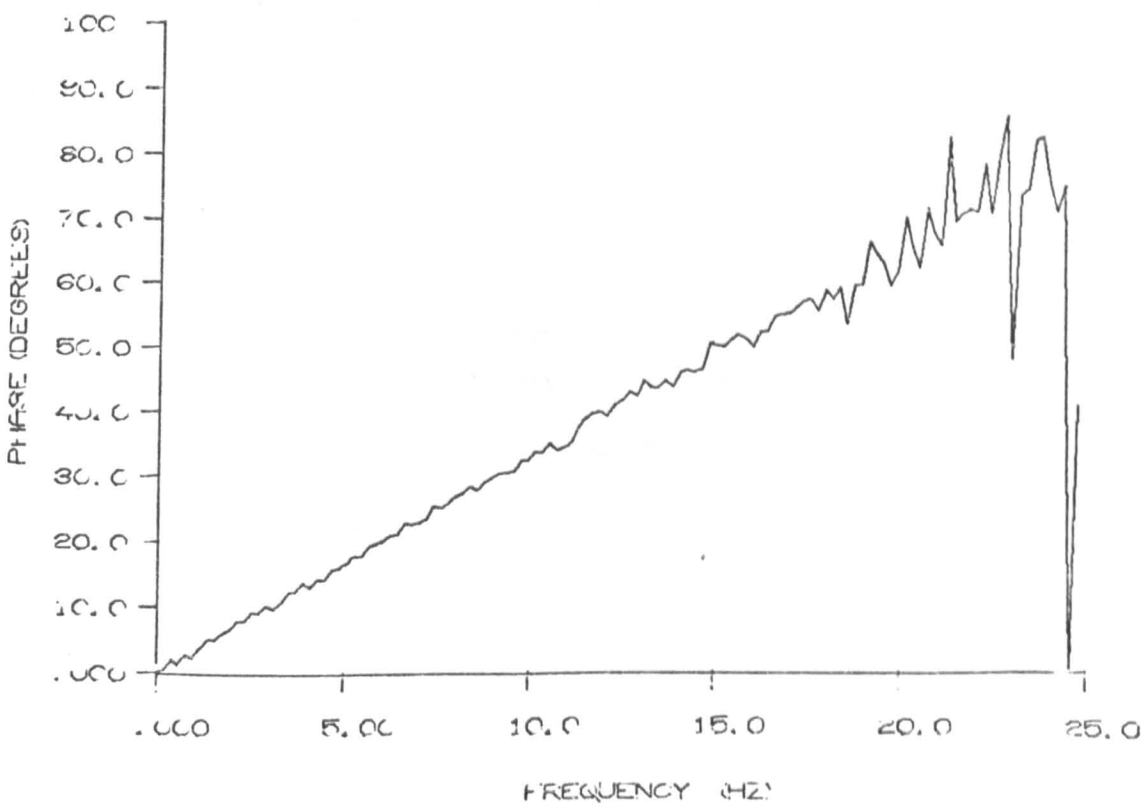
It can be seen from graphs 7.17, and 7.18, that the condition which minimises the variance due to aliasing also minimises the bias. Therefore, the effect of aliasing can be reduced by arranging that the cross-spectra between input and output of a system is real at the folding point. This may be done on-line using the alignment shift register technique described in chapter 5. Alignment can then serve the dual purpose of reducing the effects of aliasing, and reducing possible bias due to large time delays.

GRAPH 7.15.

FREQUENCY RESPONSE ESTIMATE FOR:-

A SIMULATED TIME ADVANCE OF 9 MS. SHOWING THE EFFECT OF ALIASING UPON VARIANCE; CO SPECTRUM ZERO AT THE FOLDING FREQUENCY.

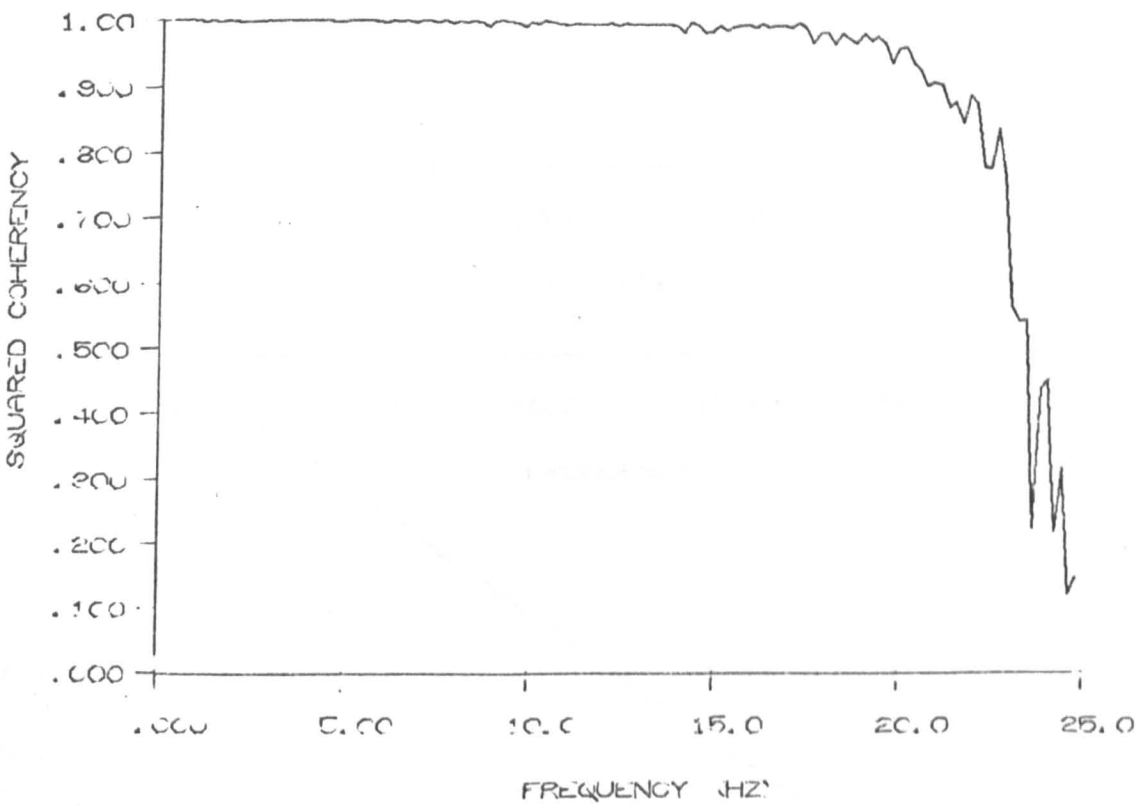




GRAPH 7.16.

COHERENCY ESTIMATE FOR:-

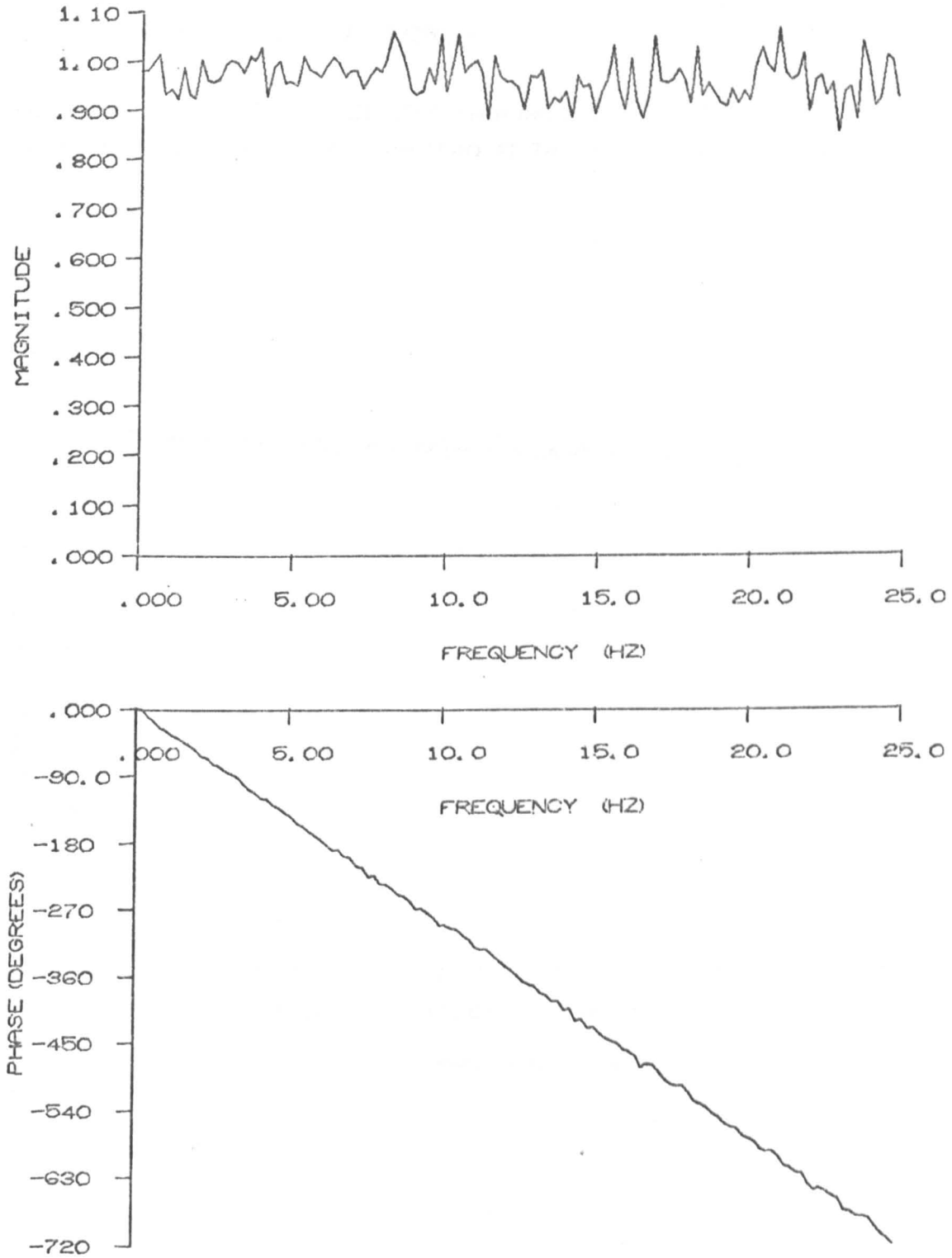
A SIMULATED TIME ADVANCE OF 9 MS. SHOWING THE EFFECT OF ALIASING UPON VARIANCE; CO SPECTRUM ZERO AT THE FOLDING FREQUENCY.



GRAPH 7.17.

FREQUENCY RESPONSE ESTIMATE FOR:-

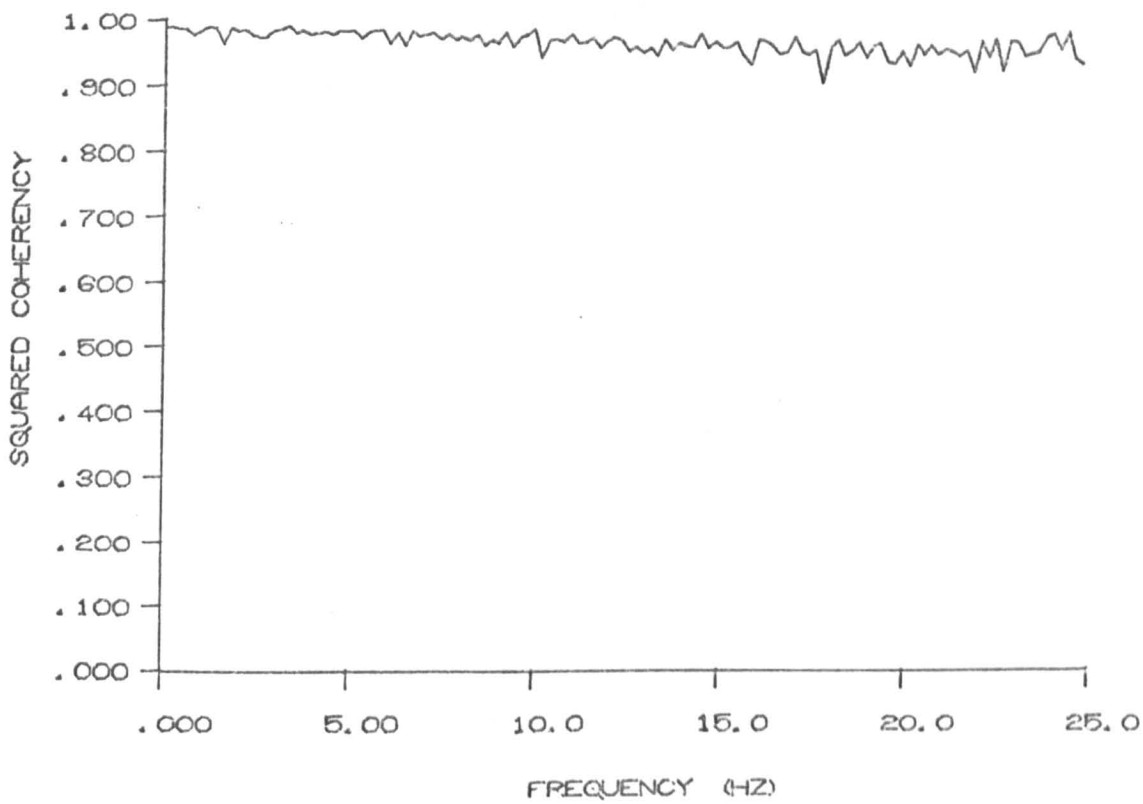
A SIMULATED TIME DELAY SHOWING THE EFFECT OF ALIASING UPON VARIANCE; QUAD SPECTRUM ZERO AT THE FOLDING FREQUENCY.



GRAPH 7.18.

COHERENCY ESTIMATE FOR:-

A SIMULATED TIME DELAY SHOWING THE EFFECT OF ALIASING UPON VARIANCE; QUAD SPECTRUM ZERO AT THE FOLDING FREQUENCY.



Alignment should not be used instead of guard filtering, but in addition to it.

The variance of aliased coherency estimates was not derived in chapter 5, and so the variability of the coherency estimates of graph 7.16, and 7.18, cannot usefully be commented upon. It has however been noticed that the variance of coherency estimates is fairly insensitive to aliasing.

7.4. Autospectral Analysis of Acoustic Noise Inside a Passenger Motor Car

In the quality control of motor car production, an essential task is to ensure that the audible noise inside a typical car does not exceed a certain level. This is done by periodically testing a new car selected at random from the production line. The car is driven at a fixed speed over a test surface, and a tape-recording made of the noise inside. A spectral analysis of the record is then conducted, and the estimated power spectrum is compared with the permissible power level.

Traditionally, analogue spectrum analyzers of the parallel bank filter type are used in this type of acoustic application. Digital techniques have not been favoured because the time series are usually long (requiring large amounts of store), and the digital analysis is 'off-line' (ref. 7.2.). The real-time autospectrum estimation technique overcomes these difficulties. It also has the advantage of presenting the results in a way visually similar to that of the parallel bank analyser. This will appeal to users of this type of analogue spectrum analyser. The spectral analysis of acoustic data is therefore a natural application of the real-time autospectrum estimation programme.

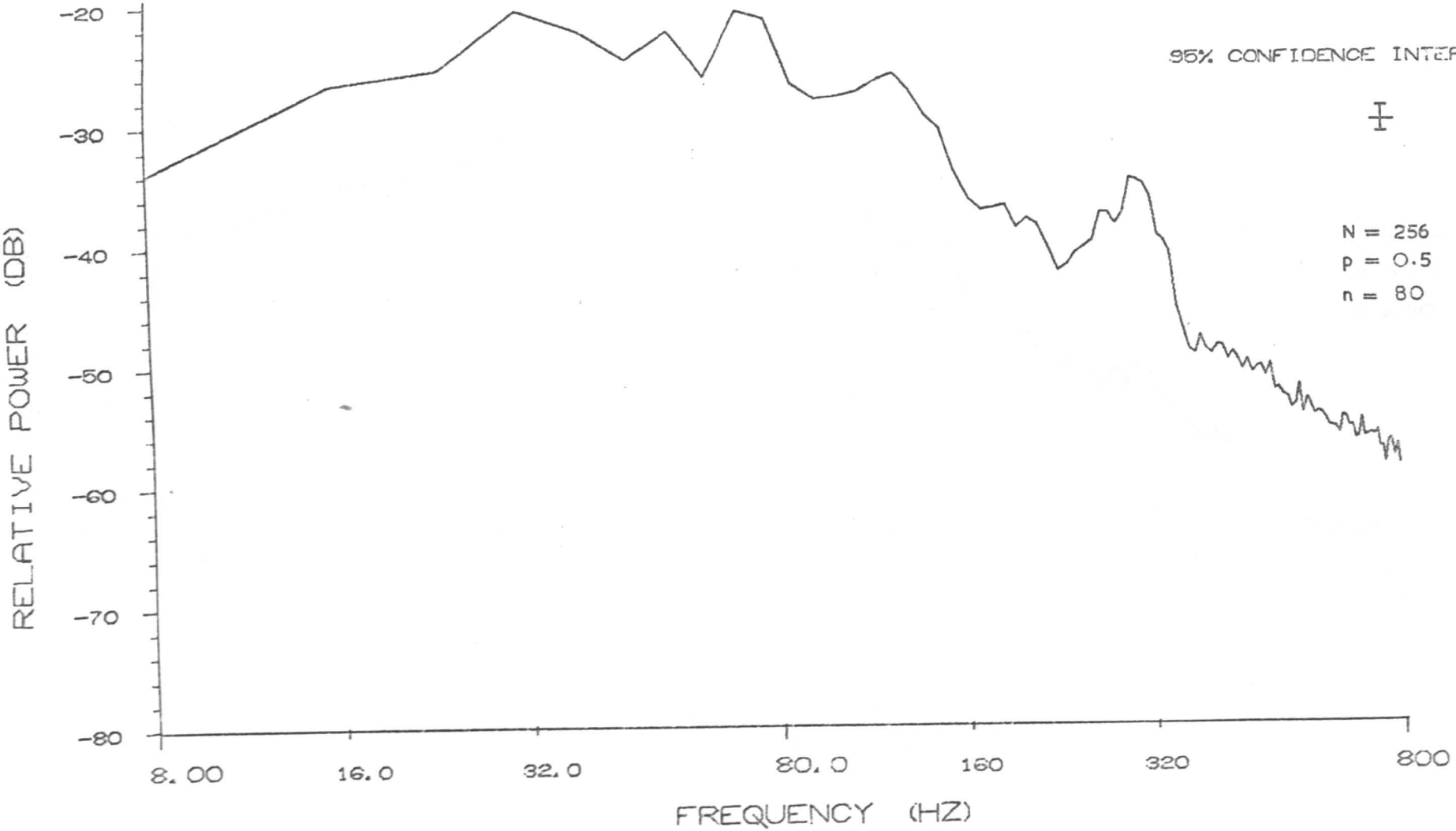
Graphs 7.19. and 7.20., show the digitally estimated auto-spectrum of the audible noise inside a large family saloon car. For convenience, the power is plotted with respect to a reference tone. The noise records were obtained by driving the car at a constant speed of 40 m.p.h. over a standard road surface, and recording the noise detected by microphone inthe front and rear of the car interior. To enable the required frequency range to be covered by the real-time

autospectrum estimation programme, the tape recording was played back at reduced speed. Aliasing was suppressed by low-pass filtering in the manner described in chapter 4. The guard filter had a fourth order Butterworth characteristic, and cutoff at 0.3 f_s. The graphs are plotted up to a frequency of 0.4 f_s, and at this frequency the worst case bias error is $\pm$ 4%.

The autospectrum programme parameters for both graphs were, N = 256 and p = 0.5. The number of periodograms averaged to obtain each curve was 80. The confidence bars are therefore based upon a chi-square distribution with 160 degrees of freedom.

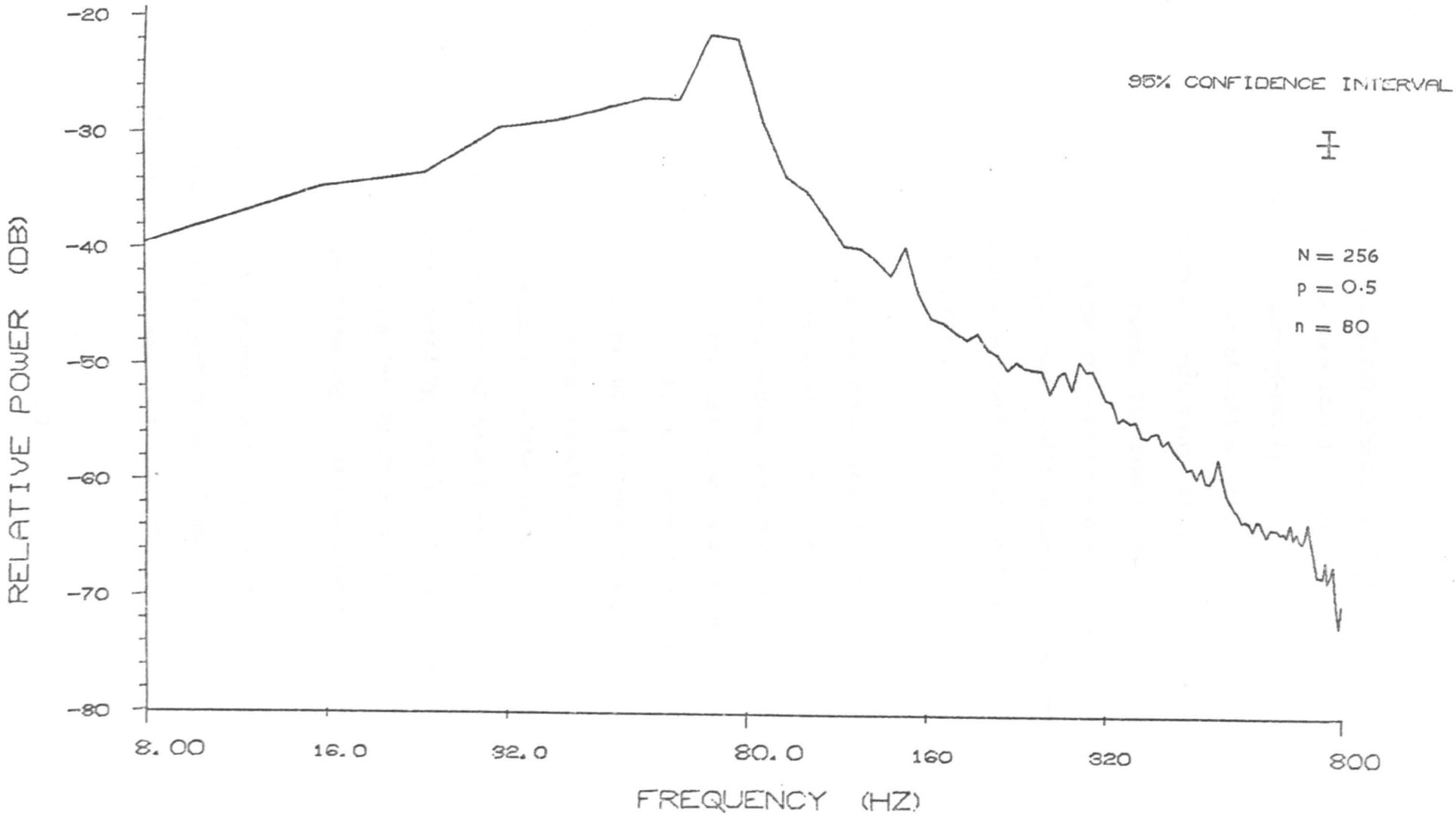
CAR INTERIOR NOISE SPECTRUM, (FRONT MOUNTED MICROPHONE)

GRAPH 7.19.



CAR INTERIOR NOISE SPECTRUM, (REAR MOUNTED MICROPHONE)

GRAPH 7.20.



7.5. Spectral Analysis of a Bogey Suspension System

This section presents some results obtained during real-time spectrum analysis experiments carried-out by the author on behalf of the Engineering Research Division of British Rail. The analysis was conducted upon random data from strain gauges attached to one of the bogeys of a fully laden goods wagon. The data was obtained by tape recording the output of these gauges while the wagon was pulled over a section of jointed track. The track condition was consistent over the section, and the wagon speed constant, so it may be assumed that the strain gauge data is stationary.

The data analysed here was taken from joint records of the outputs of strain gauges attached to a bogey axle box and the bogey bolster (to clarify these terms the reader is referred to fig. 7.1.). The axle box gauge (gauge X) was situated on an axle box on the bogey side section. This gauge indicates directly the vibration of the bogey due to unevenness in the track. The bolster gauge (gauge Y) was attached to the bolster section, which supports the wagon superstructure. Between the bogey side section and the bolster is the suspension system of the bogey. Therefore, the output of gauge Y contains a component which is due to the vibration sensed by gauge X after they have been passed through the suspension. By jointly observing the output of the strain gauges the nature of the bogey suspension may be deduced.

The data from the two gauges can be considered for simplicity as the input and output of a two-port system. Gauge X represents the input to the system from the railway track, gauge Y represents the

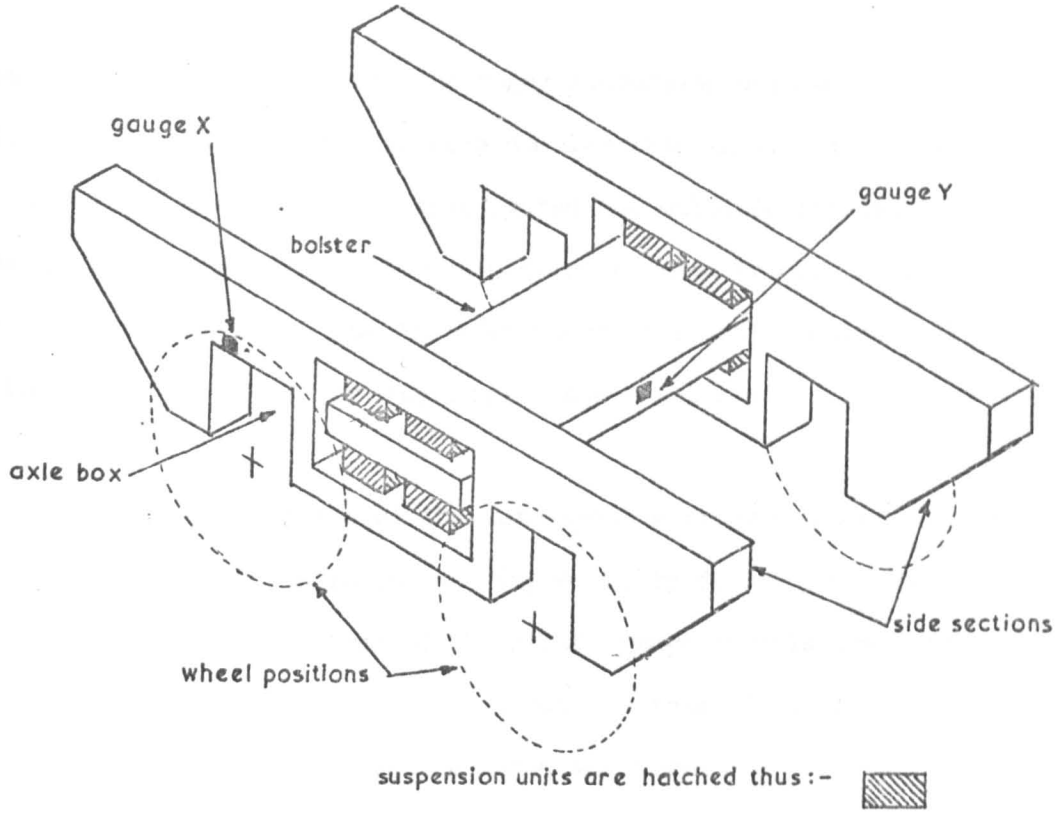
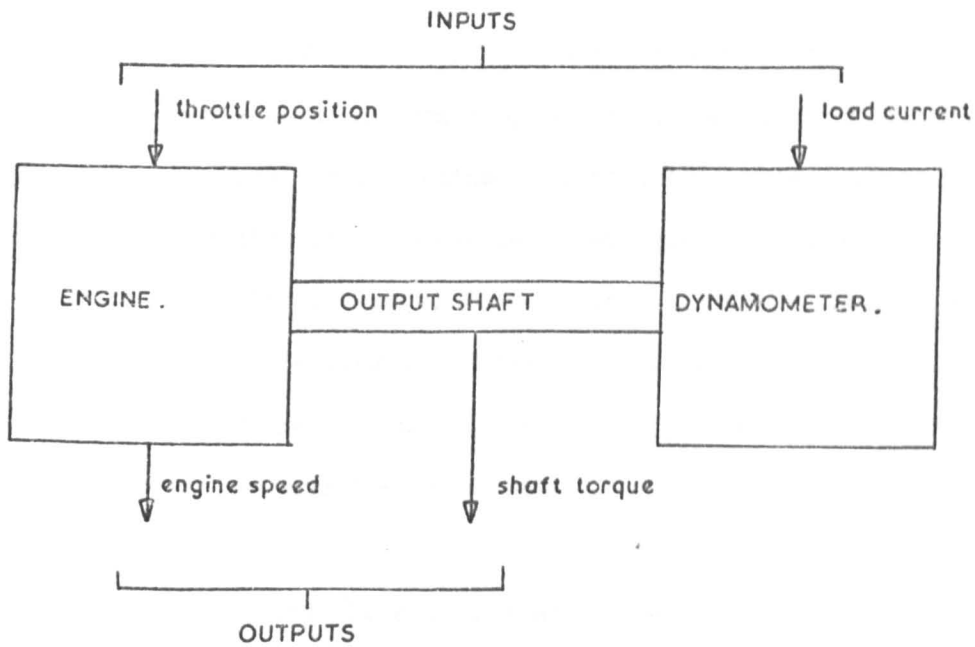


FIGURE 7.1. A RAILWAY GOODS-WAGON BOGEY.

FIGURE 7.2. INPUTS AND OUTPUTS OF THE ENGINE/DYNAMOMETER TEST RIG.



output of the system, with the bogey suspension unit as the system. In reality the situation is more complex than this. Since there are four wheels per bogey, the true system has multiple inputs. Also, because pairs of wheels are drawn over the same track, the inputs are correlated by pairs. The transfer function relating pairs will be a time delay determined by wagon speed and wheel spacing.

The parameters of the autospectrum programme were set at N = 256, p = 0.25. Aliasing was supressed by fourth order Butterworth filters. The autospectrum of the strain gauge signals are shown in graph 7.21. The major peak in the autospectrum of gauge X corresponds with the frequency at which the wagon was drawn over the rail joints. Comparison of the autospectra suggests that the suspension system has a high Q resonance at about 3.0 Hz. Also peaks in the spectra at harmonics of the joint-crossing frequency indicate that the suspension is non-linear.

The cross-spectra between gauge X and gauge Y is shown in graphs 7.22. These curves confirm that a resonance exists near 3.0 Hz. A frequency response estimate based upon this cross-spectrum estimate is given in graph 7.23. This estimate is of little value, since it assumes that the system is a linear two-port, and the actual system is a non-linear, multiple (coherent) input system. However, one useful point emerges from the estimates. They indicate a peak in the phase near 3 Hz. This is known to indicate (ref. 7.3.) the presence of two or more coupled modes near the phase peak.

It is concluded therefore, that the suspension system is

non-linear with two or more high Q modes near 3 Hz. This is consistent with the suspension system being a lightly damped, multiple spring unit. The high degree of non-linearity is probably due to the wagon being fully laden.

GRAPH 7.21.

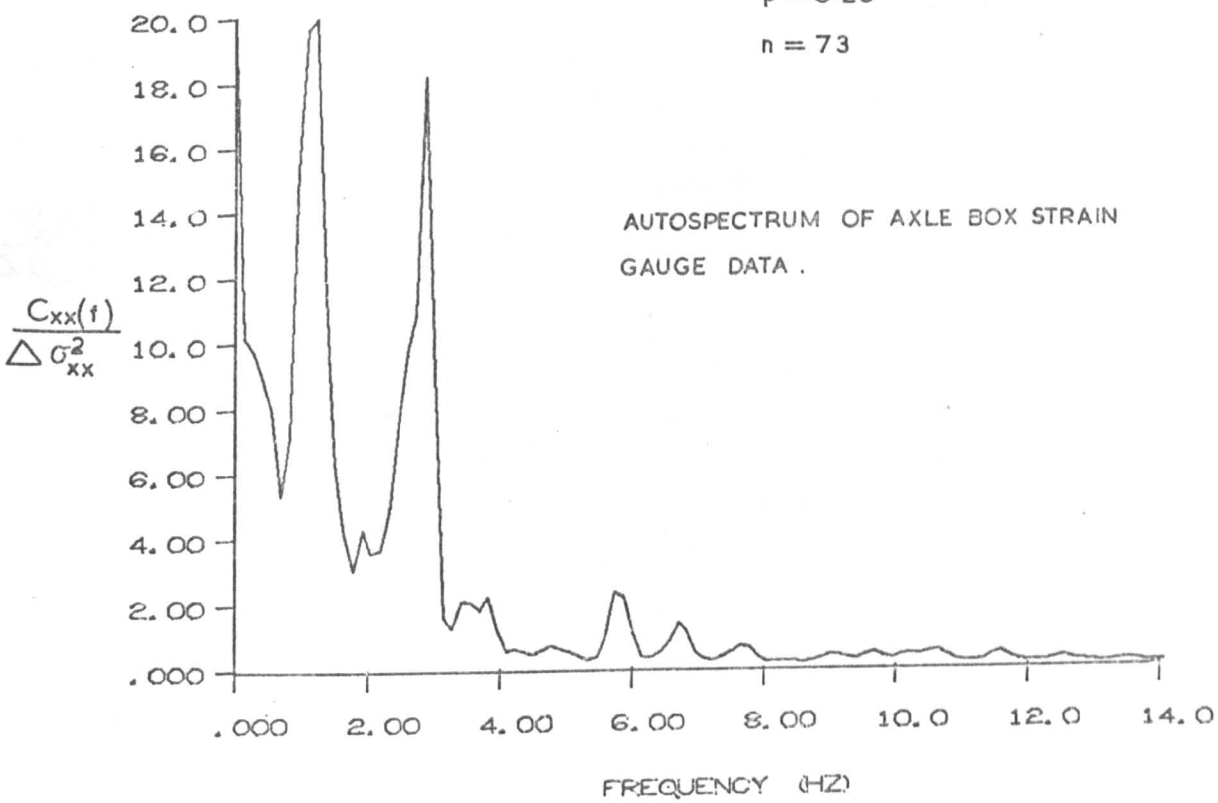
AUTOSPECTRA ESTIMATES FOR:-

BOGEY SUSPENSION SYSTEM

N = 256

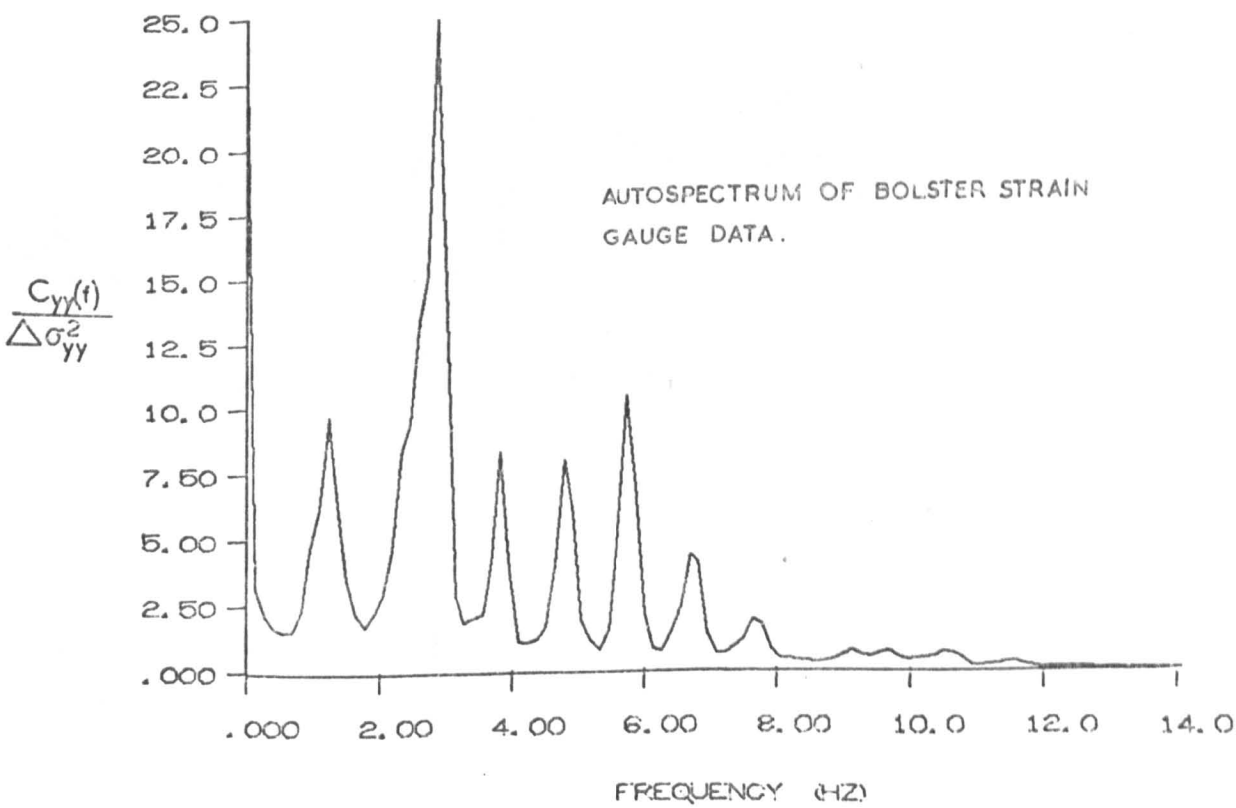
p = 0.25

n = 73



$\frac{C_{xx}(f)}{\Delta \sigma^2_{xx}}$

AUTOSPECTRUM OF AXLE BOX STRAIN GAUGE DATA.



$\frac{C_{yy}(f)}{\Delta \sigma^2_{yy}}$

AUTOSPECTRUM OF BOLSTER STRAIN GAUGE DATA.

GRAPH 7.22.

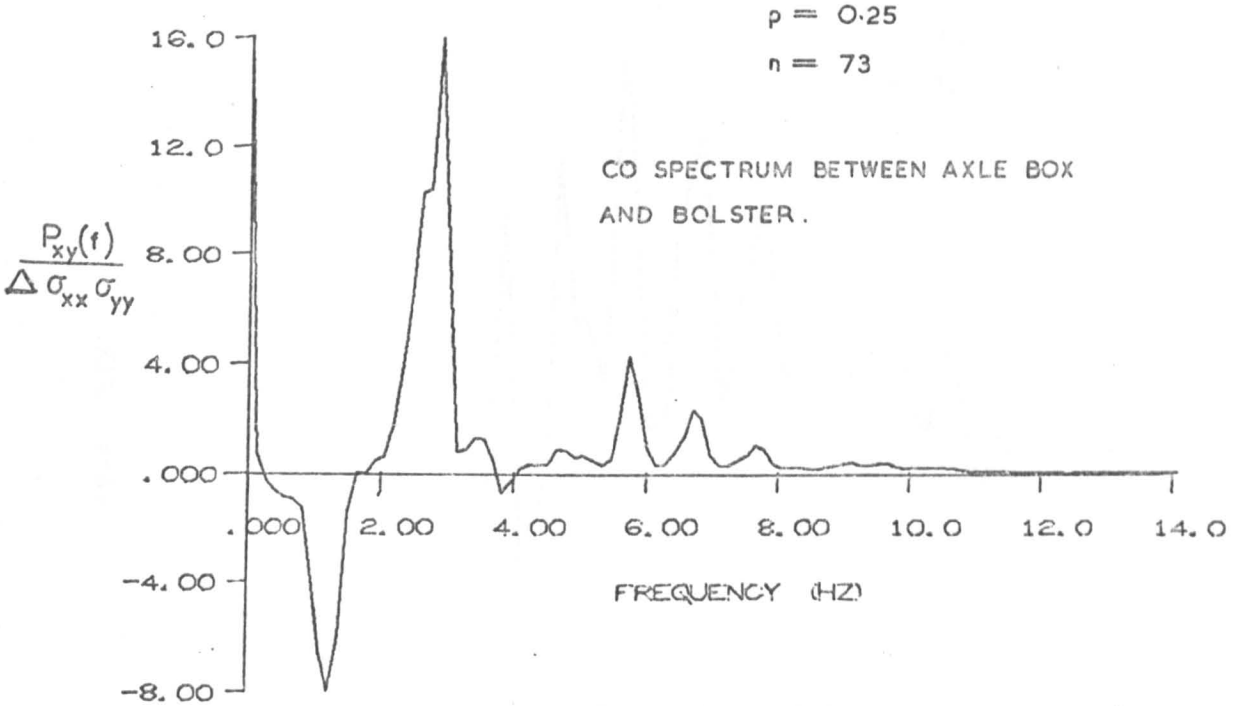
CROSS - SPECTRAL ESTIMATE FOR :-

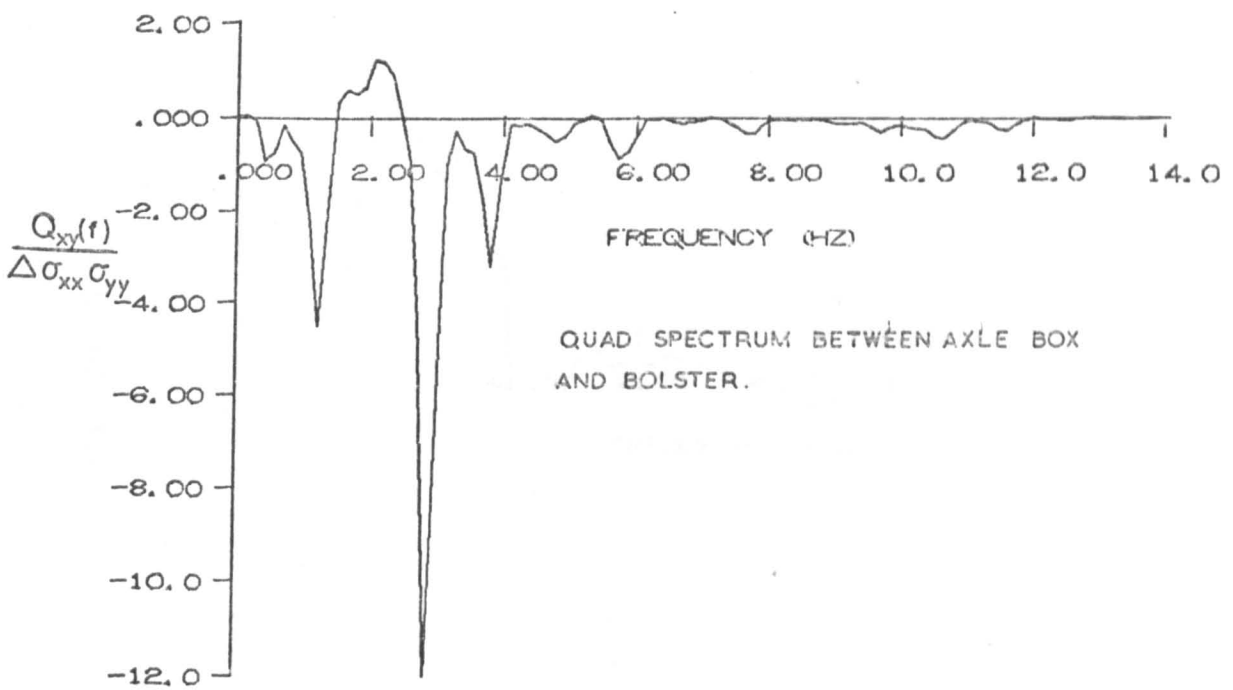
BOGEY SUSPENSION SYSTEM

N = 256

p = 0.25

n = 73

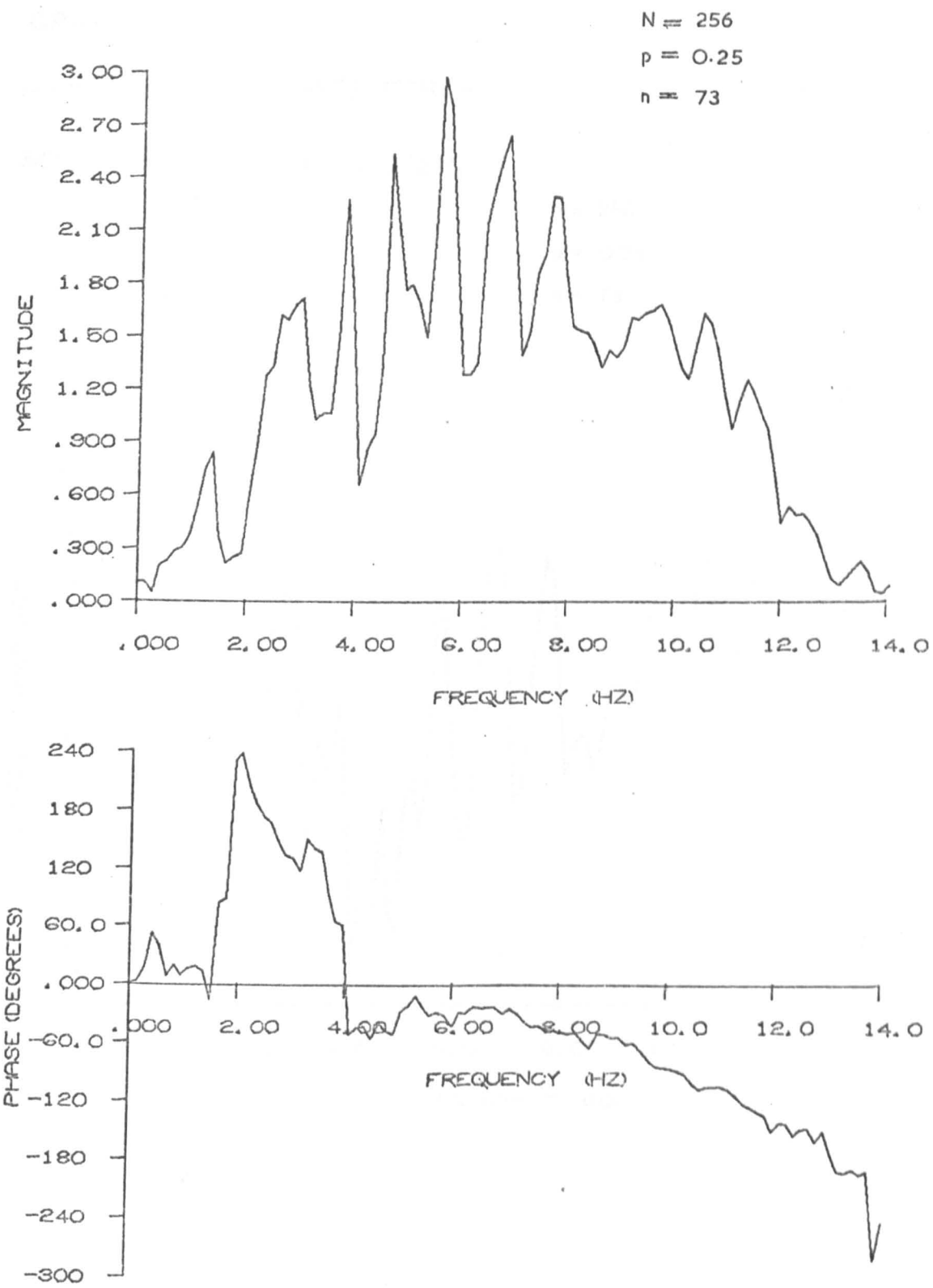




GRAPH 7.23.

FREQUENCY RESPONSE ESTIMATE FOR:-

BOGEY SUSPENSION SYSTEM



GRAPH 7.24.

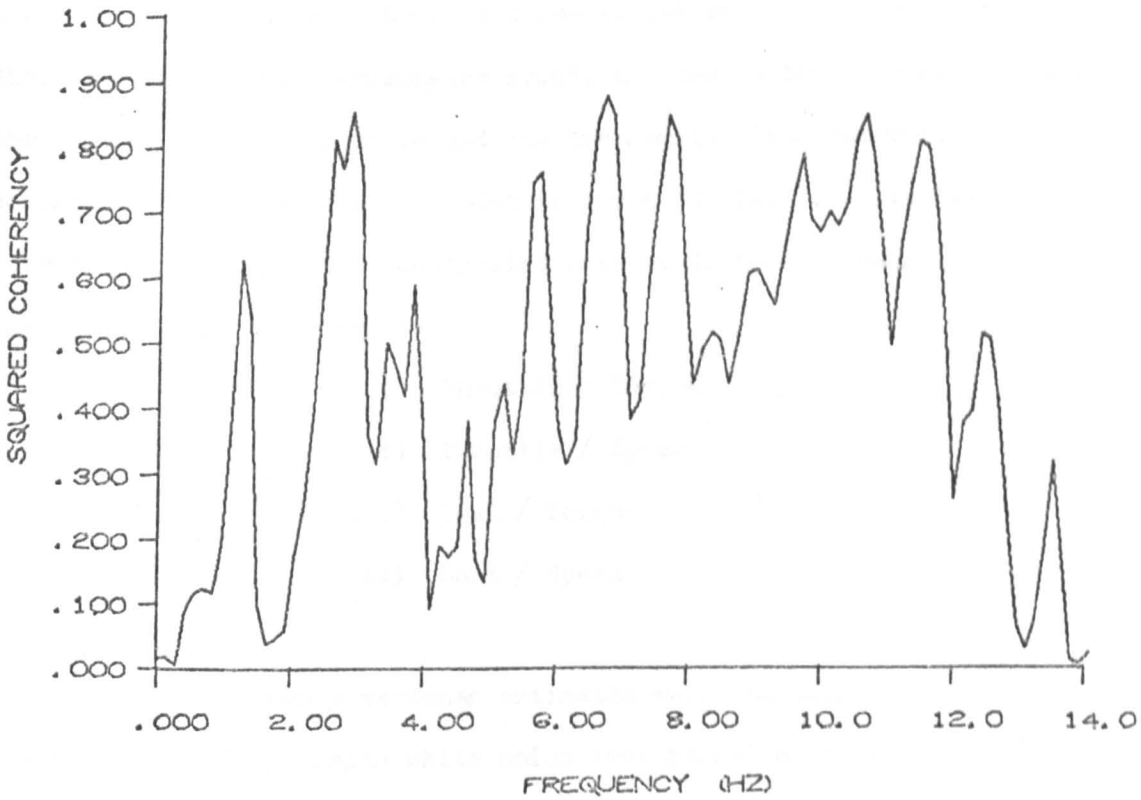
COHERENCY ESTIMATE FOR:-

BOGEY SUSPENSION SYSTEM

N= 256

p= 0.25

n= 73



7.6. Identification of an Internal Combustion Engine/Dynamometer Test Rig

This section presents some results obtained when the real-time frequency response estimation programme was used to identify an i.c. engine/dynamometer test rig. The engine was a Hillman Hunter, 1725cc, petrol engine, loaded with an eddy-current dynamometer. A description of the rig is given by J. Monk and J. Comfort in reference (7.4.), and is amplified in their doctoral theses.

The aim of the identification experiment was to obtain frequency response curves which relate certain control parameters and controlled variables. These curves can then be used to construct a mathematical model of the test rig. The controlled variables measured were the shaft torque and the dynamometer speed, and the control parameters were the throttle butterfly angle and the dynamometer load current. A schematic diagram of the variables is given in fig. 7.2. To fully correlate the control and controlled variables, four transfer functions must be known. These are :-

i) Throttle / Torque

ii) Throttle / Speed

iii) Load / Torque

iv) Load / Speed

The frequency response estimates were obtained by superimposing a small amplitude gaussian white noise test signal on to the normal operating signal of the control parameter. The test signal and the appropriate output signals were then passed through guard filters, and fed to the GEC 90/2 computer. The guard filters were used to suppress

aliasing in the manner described in section 5.3.3. The filters had fourth order Butterworth characteristics, and cutoff frequencies of 0.3 f_s. All results are plotted up to a maximum frequency of 0.4 f_s. Therefore, from chapter 5, the bias due to aliasing is a maximum at 0.4 f_s. The worst case bias values at this frequency are -8% gain error, $\pm$ 0.04 radians phase error, and -16% squared coherency error. The parameters of the real-time frequency response estimation programme were set at N = 256, p = 0.25, throughout the experiment, and in each case the programme was allowed to run until a smooth estimate was obtained.

The results of the experiment are presented in graphs 7.25, to 7.32., the continuous curves are results obtained using the real-time frequency response estimation programme, and the points marked by circles indicate results obtained by sine-wave testing. The estimated frequency response functions are plotted on logarithmic or linear scales. The gain functions shown plotted logarithmically are flat (within 1 dB) below the lowest plotted frequencies.

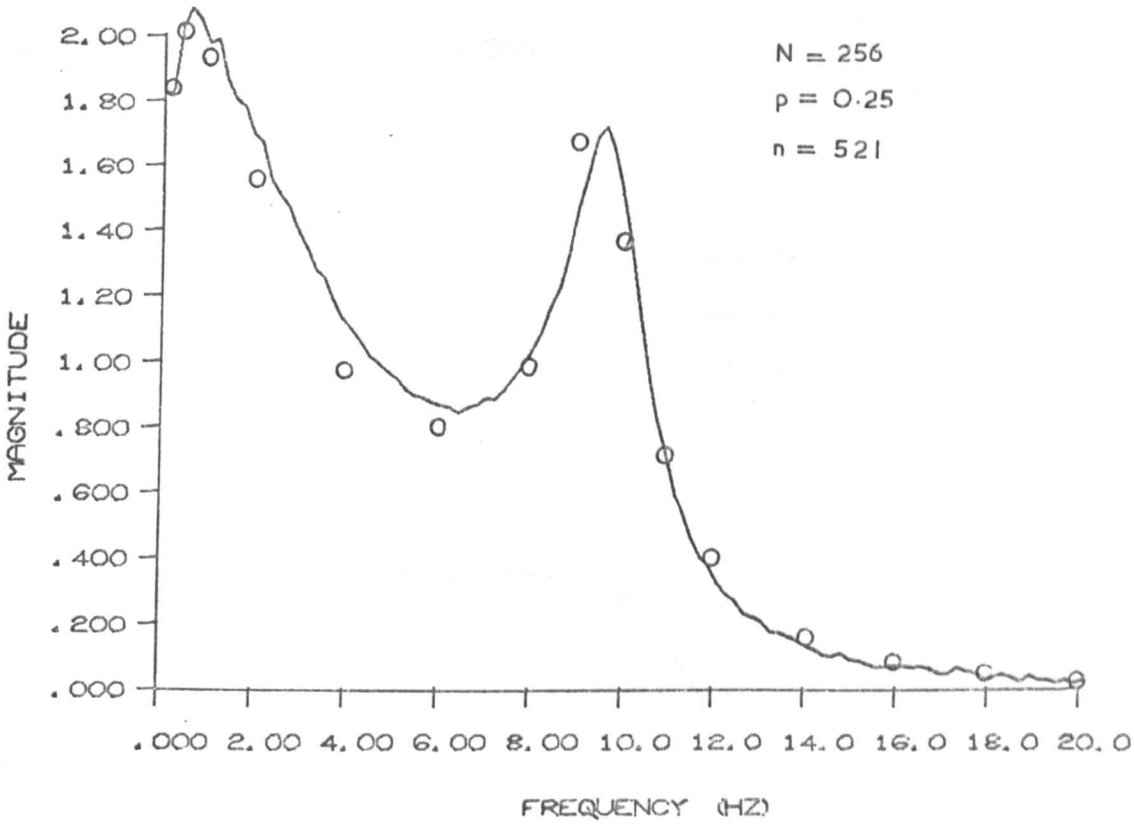
The sine-wave results were obtained at the same operating point as used in the random-signal tests, using a test signal of r.m.s. level equal to that of the noise test signal. The sine-wave results were taken as a verification of the curves obtained using the real-time frequency response estimation programme. This verification was thought necessary because the i.c. engine/dynamometer test rig is non-linear (ref. 7.4.), and it was feared that some distortion in the real-time frequency response estimates might be encountered. The random-signal method and the sine-wave method handle non-linearities differently, so

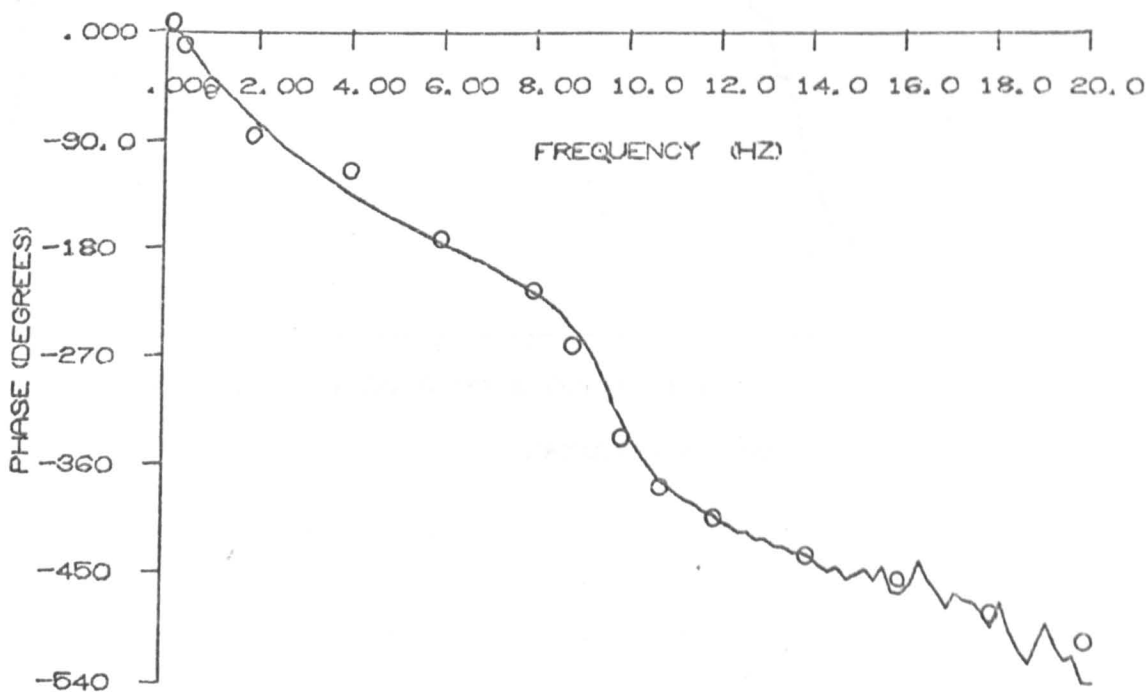
that if non-linear effects were serious, we would see significant differences between the two sets of results. Fortunately the results agree well, and so we can conclude that the real-time estimates are not seriously distorted by non-linearities.

GRAPH 7.25.

FREQUENCY RESPONSE ESTIMATE FOR:-

THROTTLE - TORQUE





GRAPH 7.26.

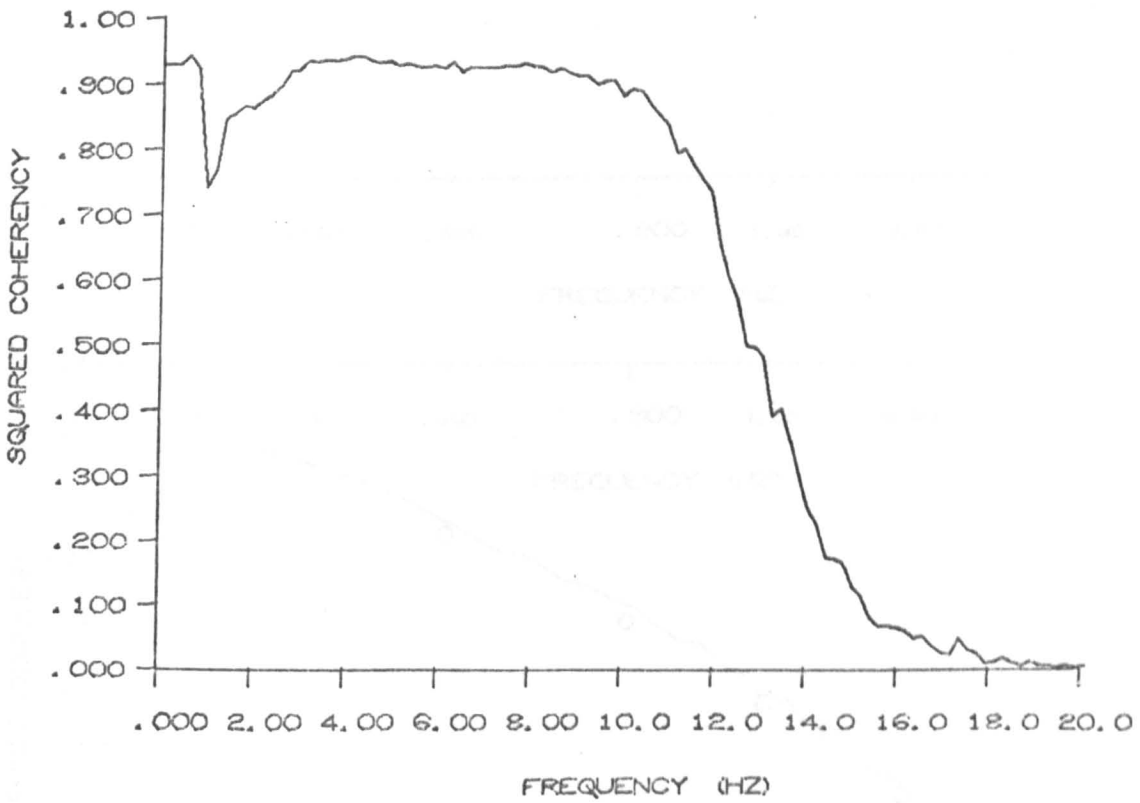
COHERENCY ESTIMATE FOR:-

THROTTLE - TORQUE

N = 256

p = 0.25

n = 521



GRAPH 7.27.

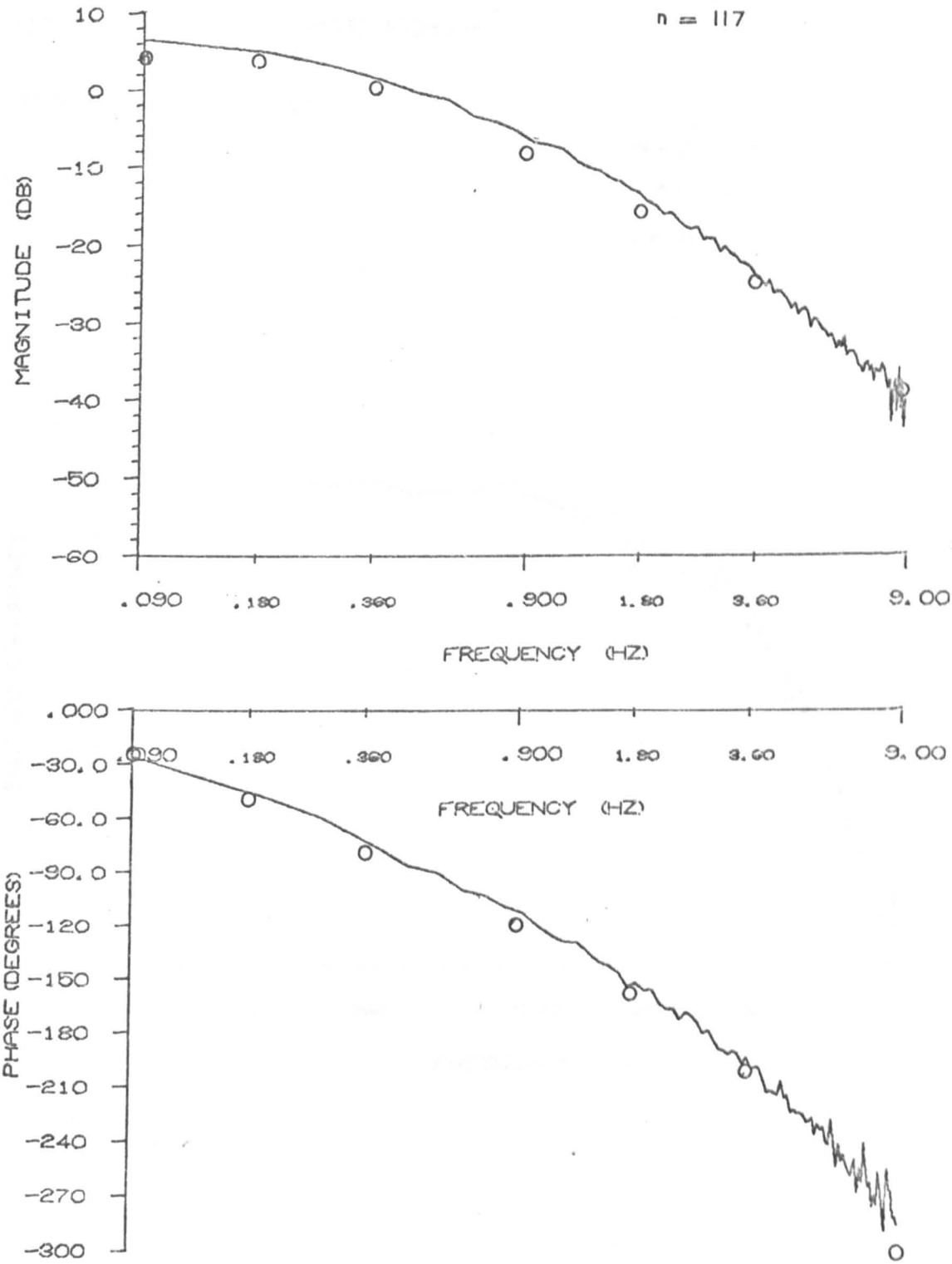
FREQUENCY RESPONSE ESTIMATE FOR:-

THROTTLE - SPEED

N = 256

p = 0·25

n = 117



GRAPH 7.28.

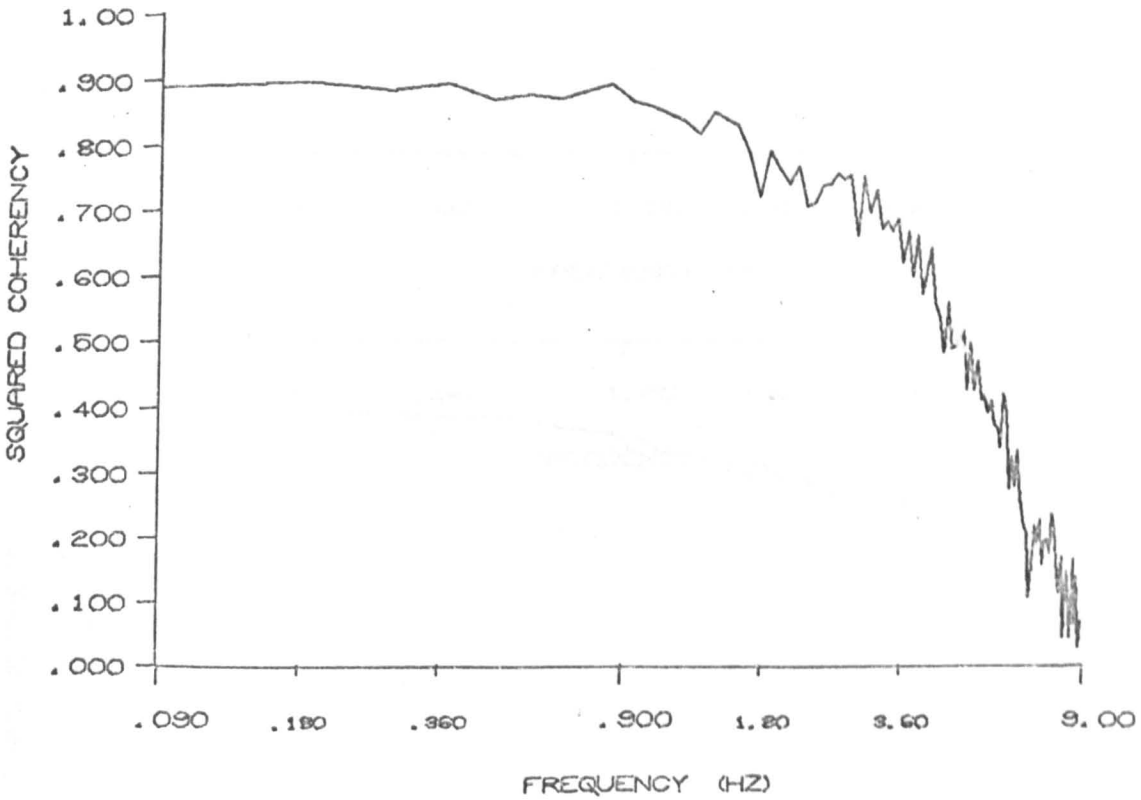
COHERENCY ESTIMATE FOR:-

THROTTLE - SPEED

N = 256

p = 0.25

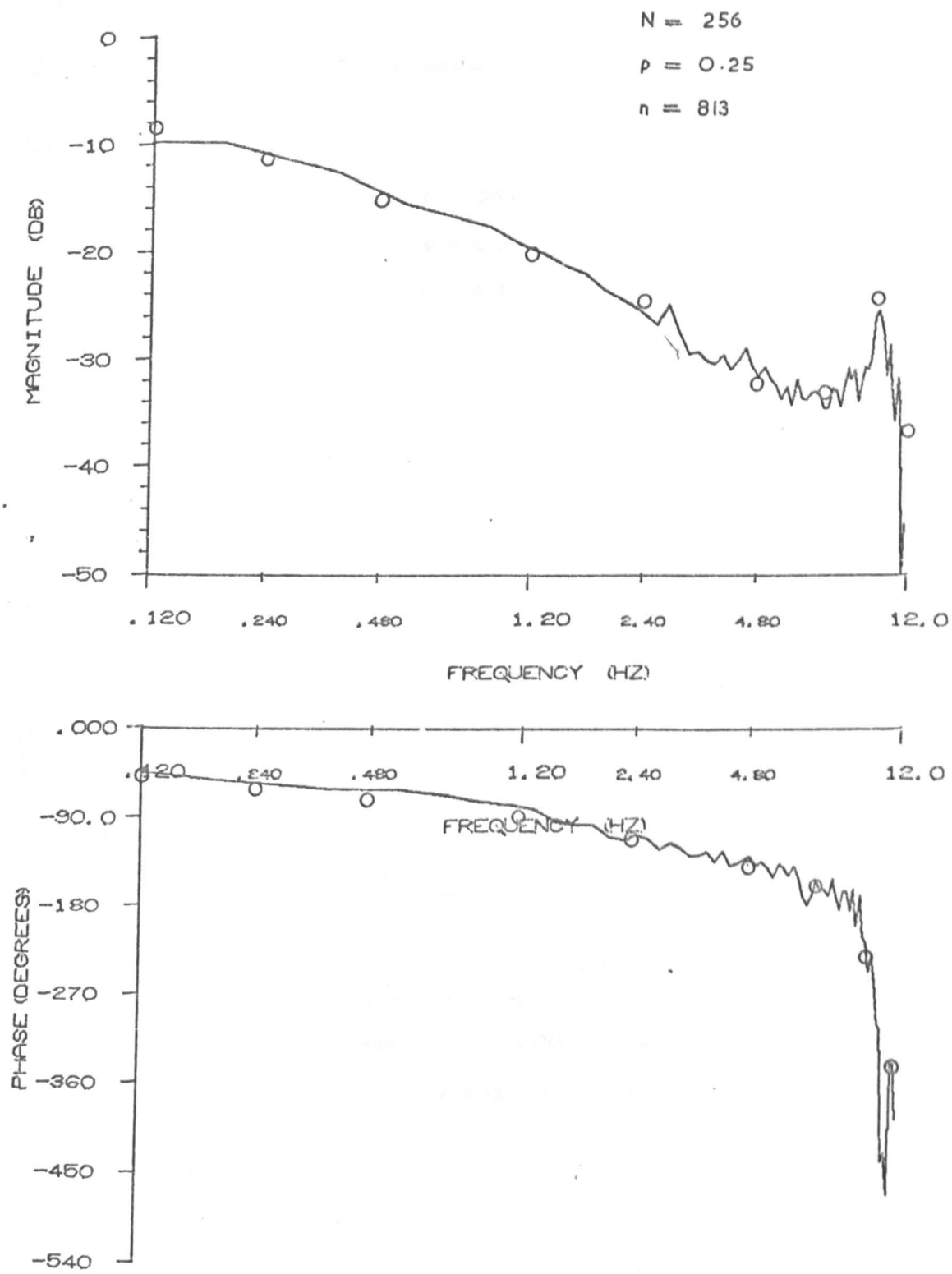
n = 117



GRAPH 7.29.

FREQUENCY RESPONSE ESTIMATE FOR:-

LOAD - TORQUE



GRAPH 7.30.

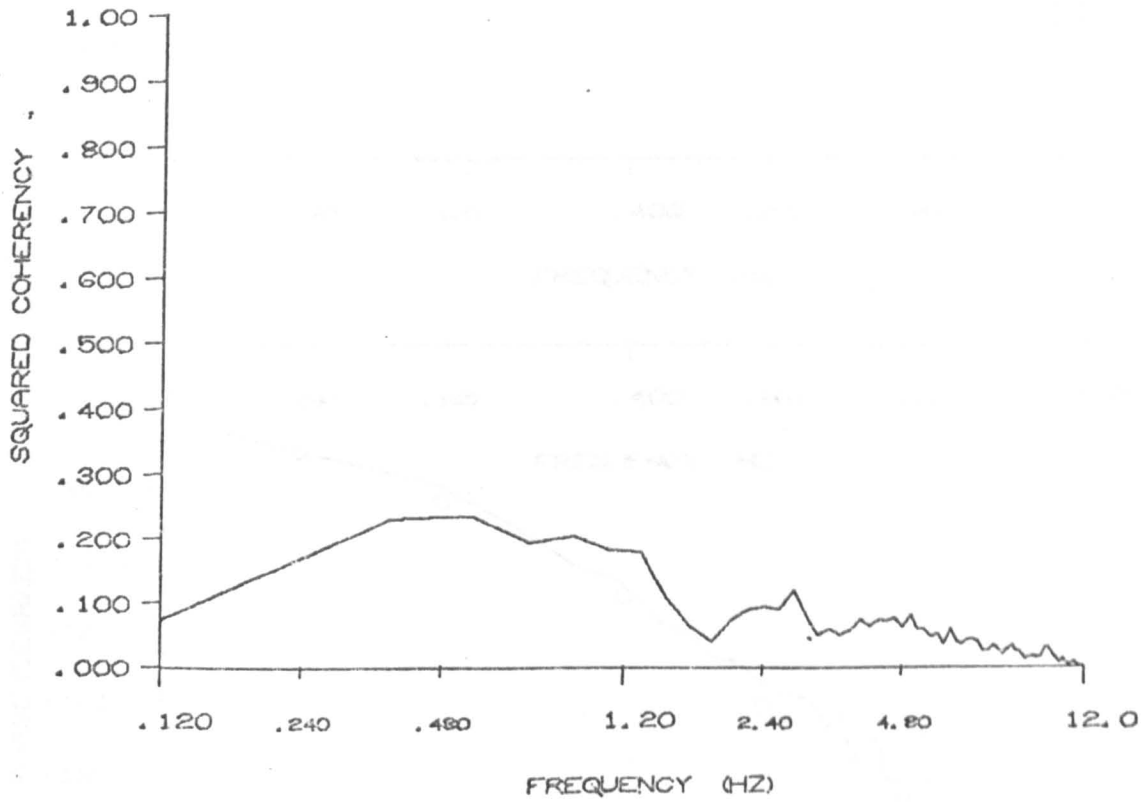
COHERENCY ESTIMATE FOR:-

LOAD - TORQUE

N = 256

p = 0.25

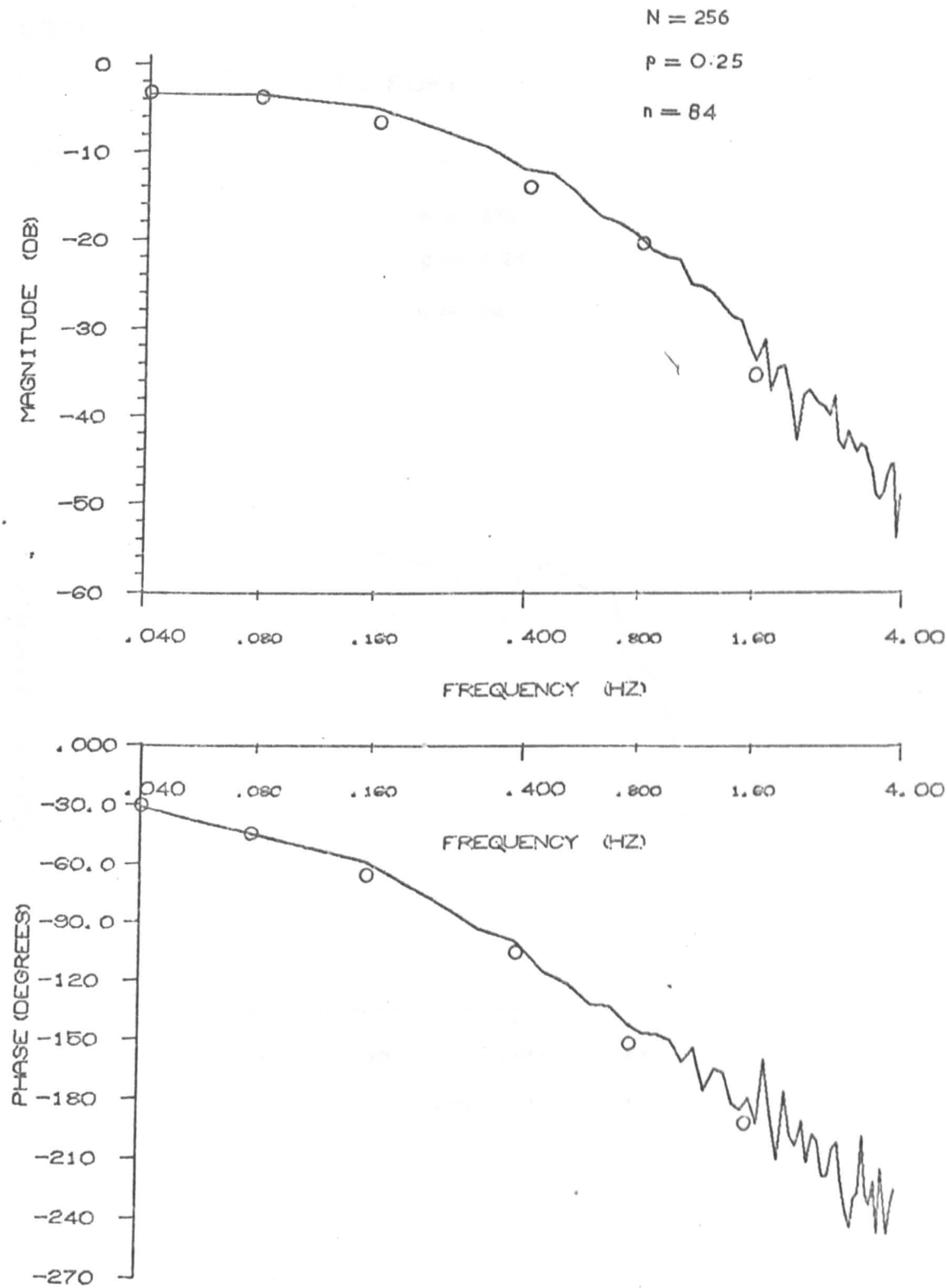
n = 813



GRAPH 7.31.

FREQUENCY RESPONSE ESTIMATE FOR:-

LOAD - SPEED



GRAPH 7.32.

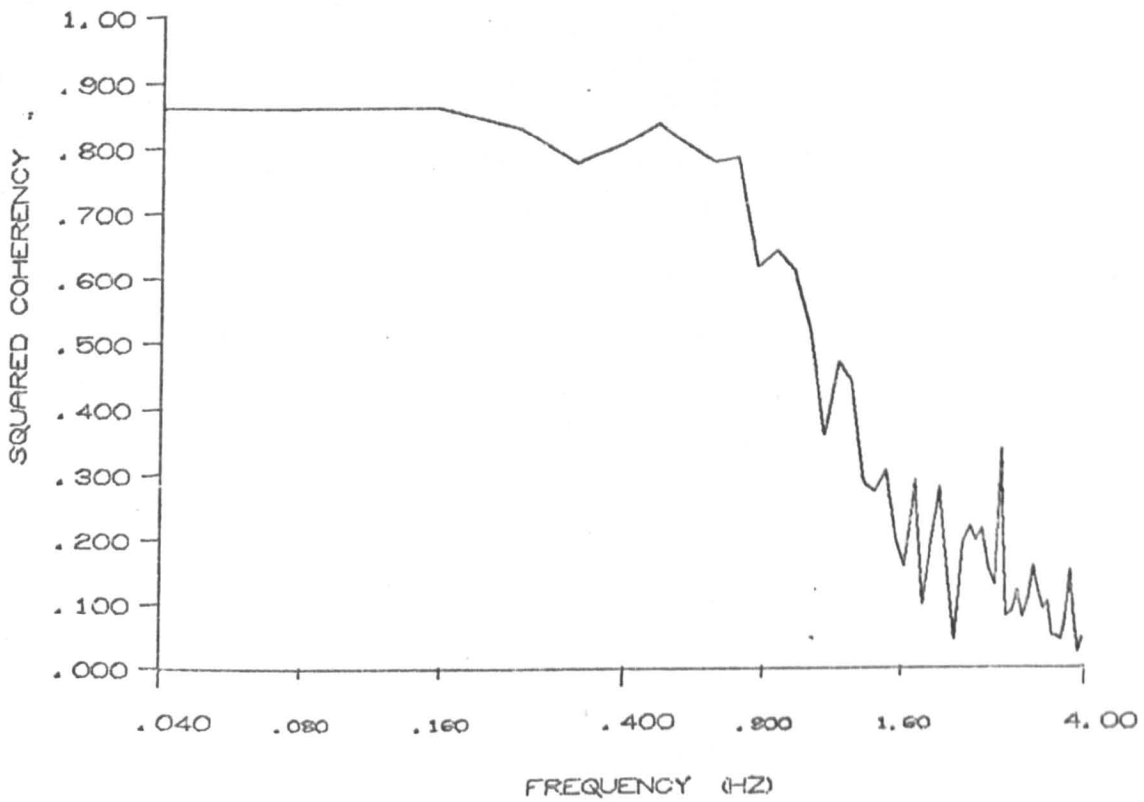
COHERENCY ESTIMATE FOR:-

LOAD - SPEED

N = 256

p = 0.25

n = 84



REFERENCES

7.1. Jenkins, G.M. and Watts, D.G., "Spectral Analysis and its Applications", Holden-Day, 1968, pp 280-282

7.2. Maling, G.C., et al., "Digital Determination of Third Octave and Full-Octave Spectra of Acoustic Noise", Trans. IEEE Vol. AU-15, No. 2, June 1969.

7.3. May, R., "Spectral Analysis of Engineering Systems", Ph.D. Thesis, University of Warwick, 1968.

7.4. Monk, J. and Comfort, J.V., "Mathematical Model of and Internal Combustion Engine and Dynamometer Test Rig", Journal of Inst. M. and C., Vol 3., No. 6, pp T98, 1970

8. CONCLUSIONS

This thesis presents a technique which enables digital spectral analysis to be carried out on-line and in real-time. To demonstrate its validity programmes which implement the technique on a GEC 90/2 on-line digital computer are presented. These programmes allow autospectral and frequency response measurements to be made in real-time up to a maximum frequencies of $\sim$ 500 Hz and $\sim$ 250 Hz respectively. Theoretical support is provided by a description of the statistical properties of the estimators used in the programmes. The frequency response estimation programmes presented here are for the analysis of linear, time-invariant, open-loop, two-port systems. It was recognised that in a large class of measurement situations the system under test is closed-loop, and the loop cannot be opened to make measurements. In such cases normal estimation methods are inadequate, and for this reason a study of frequency response estimators for closed-loop systems is also presented.

8.1. Assessment of the Real-Time Spectral Analysis Technique

The author would at this point like to present his brief assessment of the real-time digital spectral analysis method as compared with alternative techniques. The assessment is presented under a series of headings, each heading being concerned with a desirable feature we would wish to incorporate in a spectral analysis technique.

i) Frequency Resolution and Statistical Stability

These features are required in a spectral analysis method so that reliable spectral estimates with low variability and negligible bias error may be obtained. The real-time spectral analysis method, in common with special purpose machines, can combine the qualities of good resolution and statistical stability. This is a significant advantage over off-line techniques, which generally work from fixed duration records, and sacrifice resolution to obtain stable estimates.

ii) Availability of Results

If the user is to conduct experiments quickly and efficiently, the results of his analyses should be rapidly available. Here again the real-time digital method and on-line special purpose machines have the advantage over off-line techniques, since they present results in real-time.

iii) Frequency Range

In order that a wide range of signal and systems can be studied, a good spectral analysis technique should allow measurements over a wide range of frequencies. In this respect off-line methods and special purpose machines are usually superior, since

the maximum frequency range of the real-time technique is limited by the speed with which the computer can execute the FFT. The programmes described in this thesis have upper frequency limits of ~ 500Hz and ~ 250 Hz for autospectral and frequency response measurements respectively. These limits may be extended if a faster computer is used, and estimated frequency ranges for a typical modern machine are given in section 8.2.

iv) Accuracy

If we are to attach any significance to our results, it is important that the analysis be carried out with as little error as possible. In this respect it is expected that the digital methods would be generally superior to analogue special purpose machines. This is because of the great accuracy that can be achieved with digital arithmetic.

v) Flexibility

If economic use is to be made of a spectral analysis machine it should be easily adaptable to suit a wide variety of problems. In this respect programmed computers have clear advantage over special purpose machinery. This is because it is cheap and easy to alter the coding of a computer programme, while the modification of a special purpose machine might be impractical, or involve the purchase of ancillary equipment.

To summarise, real-time digital spectral methods overcome the major disadvantages of off-line digital methods, and are potentially more accurate and flexible than special purpose machines.

An important feature of spectral estimation methods not mentioned above is the way in which the analysis is conducted. Generally speaking, if we have to obtain an estimate of a continuous curve (such as a frequency response function) we set about it by estimating a set of points on the curve, and then obtain an estimated curve by fitting a curve to the set of points. The estimated points may be obtained in one of two ways :- either we can form estimates of the complete set of points concurrently, or we can obtain estimates point-by-point. These methods are known respectively as 'parallel' and 'serial' estimation.

A parallel estimation procedure is employed in the real-time spectral analysis routines. Its chief advantage is that an overall appreciation of the curves being estimated is quickly obtained, thus enabling the user to react immediately to results, and optimise estimation parameters. This is in direct contrast with the serial mode of estimation where the user must wait until the set of estimates is complete before he can begin to assess the results.

The disadvantages of the parallel estimation procedure applied to real-time spectral analysis is :-

i) The set of estimated points are equally spaced on the frequency axis. This is a disadvantage when small sections of a curve contain sharp peaks or troughs, and so require closely spaced points to described them, while the remainder of the curve is relatively smooth and can adequately be described by widely spaced

points. The equi-spacing of points is also inconvenient when the data is plotted to logarithmic scales, since most of the points then become bunched at the upper end of the scale.

ii) When used to estimate frequency response functions, the parallel mode of the real-time spectral analysis technique is often inefficient. This occurs when the squared coherency is low only over a limited band of frequencies and the estimates are correspondingly unstable in this region. This may mean that much smoothing may have to be used on a set of estimates, which are only unstable over a limited region.

These disadvantages are not shared by serial mode estimation procedures, since the spacing of estimates and the amount of averaging per estimate may be varied from point to point.

From the preceding paragraphs it may be seen that parallel and serial estimation procedures are in some ways complementary and, if used together in a real-time estimation routine, can form a powerful combination. For instance, a parallel estimation procedure can be used to form an initial set of estimates. Then from this set, areas of interest can be selected, and studied in detail using a serial estimation procedure. A proposal for such a real-time spectral analysis facility is given in the following section.

8.2. Proposals for Further Work

In view of the general criticisms of parallel parameter estimation and the deficiences of the GEC 90/2 computer programmes, there is a strong case for the development of a flexible real-time spectral analysis facility on a larger computer. Since the School of Engineering Science is (probably) acquiring an XDS Sigma 5 computer it is expected that the facility will be implemented on this machine.

The proposal is that a digital spectral analysis programme package that contains a parallel and serial estimation procedure be developed. The package will be built around the fixed-point FFT, and will allow estimation of autospectra, cross-spectra or frequency response functions in parallel or serial mode. This will enable the user to obtain real-time estimates using the parallel estimation procedure, select areas of interest, and use the serial estimation procedure to amplify results in those regions.

The parallel spectral estimation technique used will be the one described in this thesis, and it is suggested that the serial technique use the method of complex demodulates outline in the first chapter.

It should be possible to programme this package efficiently on the Sigma 5 computer because :-

i) The Symbol assembler is used, so 90/2 routines can be easily reprogrammed.

ii) Both parallel and serial estimation methods can use the same fixed-point FFT programme block.

iii) The Sigma 5 uses a 32 bit word so that look-up tables and data can be stored two elements per word.

The Sigma 5 is a faster, larger and more accurate machine than the GEC 90/2 computer. The performance of the real-time spectral analysis programme will therefore be improved. It is estimated that the Sigma 5 spectral analysis facility will be able to handle data blocks of (at least) 2048 points, and analyze frequencies up to 3 KHz, (autospectra) and 1.5 KHz (frequency response).

Further theoretical work that is proposed arises from the work contained in chapter 6 on closed-loop systems, and from the serial estimation procedure in the proposed real-time spectral analysis facility.

Firstly, consider the results of chapter 6. Approximate variance expressions for the forward path frequency response function were obtained, and these indicated that disturbances in the feedback loop are of secondary importance in assessing the variability of the forward path frequency response estimators. Confidence statements were developed for the special case of noise-free feedback, and it was suggested that since the feedback noise is a second order effect, (and in the absense of more rigorous results) these confidence statements may be used to assess the

quality of estimates obtained in the general case of noisy feed-forward and feedback paths.

To extend these results the following two items of study are proposed. Firstly, it is suggested that variance expressions which given some indication of the effect of feedback noise be derived. Secondly, it is suggested that the sampling properties of the forward path frequency response estimator be fully developed by considering the probability distributions associated with tri-variate, complex, gaussian process. These results may then be applied to the estimation of closed-loop frequency response functions, and to the more general field of tri-variate spectral analysis. Of particular interest in this field would be the linear, time-invariant, system with two coherent inputs and one output.

The proposed real-time spectral analysis facility, which combines serial and parallel estimation techniques, will require theoretical support. This will be necessary to design and justify an appropriate serial spectral estimation method. The method of complex demodulates was suggested as a suitable serial method because it can be conveniently programmed to operate using the same FFT programme block as the parallel estimation routine. It is therefore suggested that a thorough study of the complex demodulates method be undertaken.

APPENDIX 1: ALGOL PROGRAMMES FOR FOURIER TRANSFORMATION.

FFTT & FFTF

PROCEDURES FFTT AND FFTF EXECUTE THE FAST FOURIER TRANSFORMATION ALGORITHM. THEY OPERATE ON THE N POINT COMPLEX ARRAY REA[I]+JIMA[I]. I:=0,---N-1. THE BLOCK SIZE IS N=R↑M.

CALL:

TO FORWARD TRANSFORM THE N POINT COMPLEX TIME SERIES XR+JXI. CALL FFTT(R,M,N,XR,XI);

TO INVERSE TRANSFORM THE N POINT COMPLEX FOURIER COEFFICIENTS XR+JXI. CALL FFTT(R,M,N,XR,XI);

TIMING:

EXECUTION TIMES FOR TYPICAL N ARE GIVEN IN TABLE 2.1.

```
"PROCEDURE" FFTT(R,M,N,REA,IMA);
"COMMENT" THIS PROCEDURE EXECUTES THE FAST FOURIER TRASFORM USING
      DECIMATION IN TIME;
"VALUE"R,M,N; "INTEGER"R,M,N;
"ARRAY"REA,IMA;
"BEGIN"
"INTEGER"I,L,MLESS1,ILESS2,NLESS1,LTOP,KTOP,JT,JW,RMI,RI1,
K,V,Q,SUM,P,JJ,NN;
"REAL" TPN,A,B,ATR,ATI,BTR,BTI,TEMP;
"INTEGER" "ARRAY" J[0:M-1];
"IF" N<0 "THEN" NN:= -N "ELSE" NN:= N;
MLESS1:=M-1; TPN:= -6.283185307180/N; NLESS1:= NN-1;
"FOR"I:=1"STEP"1"UNTIL"M"DO"
"BEGIN" RMI:=R↑(M-I); RI1:=R↑(I-1);
LTOP:=RI1-1; KTOP:=RMI-1; ILESS2:=I-2;
"FOR"L:=0"STEP"1"UNTIL"LTOP"DO"
"BEGIN" "IF"L=0"THEN"
"BEGIN" "FOR"P:=0 "STEP"1 "UNTIL" ILESS2"DO"
J[P]:=0; JW:=JT:=0; "END"
"ELSE" "BEGIN" P:=ILESS2;
RETURN: J[P]:=J[P]+1;
"IF" J[P]=R "THEN" "BEGIN" J[P]:=0; P:=P-1;
"GOTO" RETURN; "END";
JW:=JT:=0;
"FOR" P:=0 "STEP" 1 "UNTIL" ILESS2 "DO"
"BEGIN" JW:=JW+J[P]*R↑P;
JT:=JT+J[P]*R↑(MLESS1-P); "END"; "END";
A:=COS(RMI*JW*TPN);
B:=SIN(RMI*JW*TPN);
"FOR" K:=KTOP "STEP" -1 "UNTIL" 0 "DO"
"BEGIN" V:=K+JT; Q:=V+RMI;
TEMP:=REA[Q]*A-IMA[Q]*B;
ATR:=REA[V]+TEMP; BTR:=REA[V]-TEMP;
TEMP:=IMA[Q]*A +REA[Q]*B;
ATI:=IMA[V]+TEMP; BTI:=IMA[V]-TEMP;
REA[V]:=ATR;  IMA[V]:=ATI;
REA[Q]:=BTR;  IMA[Q]:=BTI; "END";
"END" L LOOP;
"END" I LOOP;
"FOR" P:=0 "STEP" 1"UNTIL" NLESS1 "DO"
```

```
"BEGIN" K:=P;
BACK: SUM:=0;
"FOR"I:=1 "STEP" 1 "UNTIL" MLESS1 "DO"
"BEGIN" Q:=K"DIV"R;
JJ:=K-Q*R;
SUM:=SUM*R+JJ;
K:=Q; "END";
K:=SUM*R+K;
"IF" K<P "THEN" "GOTO" BACK;
"IF" K "NE" P "THEN"
"BEGIN" ATR:=REA[K]; ATI:=IMA[K];
REA[K]:=REA[P] ; IMA[K]:=IMA[P];
REA[P]:=ATR; IMA[P]:=ATI; "END";
"END" P LOOP;
"IF" 0<N "THEN" "FOR" I:=0 "STEP" 1 "UNTIL" NLESS1 "DO"
"BEGIN" REA[I]:= REA[I]/NN;
        IMA[I]:= IMA[I]/NN;
"END";
"END" OF FFTT;



"PROCEDURE" FFTF(R,M,N,REA,IMA);
"COMMENT" THIS PROCEDURE EXECUTES THE FAST FOURIER TRANSFORM USING
      DECIMATION IN FREQUENCY;
"VALUE" R,M,N; "INTEGER" R,M,N;
"ARRAY" REA,IMA;
"BEGIN"
      "INTEGER" NN,NLESS1,RMI,KTOP,LTOP,MLESS1,KDASH,L,K,KA,KB,OLDRMI,
        P,Q,JJ,SUM;
     "REAL" TPN,C,S,ATR,ATI ,BTR,BTI,TPNIND;
       "IF"N<0 "THEN" NN:=-N
             "ELSE" NN:=N;
      NLESS1 := NN-1;  RMI := NN"DIV" R;
      KTOP := RMI-1;  LTOP :=1;  MLESS1 :=M-1;
      TPN :=-6.283185307180/N;
      "FOR" I:= 1 "STEP" 1 "UNTIL" M"DO"
"BEGIN" KDASH :=0;  TPNIND := TPN*LTOP;
      "FOR" L := 1 "STEP" 1 "UNTIL" LTOP "DO"
"BEGIN"
      "FOR" K := 0 "STEP" 1 "UNTIL" KTOP "DO"
"BEGIN"
      C := COS (K*TPNIND);
      S := SIN (K*TPNIND);
      KA := K+KDASH;  KB := KA+RMI;
      ATR := REA[KA] + REA [KB];
      ATI := IMA[KA] + IMA [KB];
      BTR := (REA[KA] - REA[KB]) *C - (IMA[KA] - IMA[KB])*S;
      BTI := (IMA[KA] - IMA[KB ])*C + (REA[KA] - REA[KB])*S;
      REA[KA] := ATR;  REA[K B] := BTR;
      IMA[KA] := ATI;  IMA[KB] := BTI;
"END" OF KLOOP;
      KDASH := KDASH +  OLDRMI;
"END" OF LOOP;
      LTOP := R*LTOP; OLDRMI := RMI;
      RMI := RMI"DIV" R;  KTOP := RMI -1;
"END" OF I LOOP;
```

```
      "IF" N>0 "THEN"
      "FOR" I := 0 "STEP" 1 "UNTIL" NLESS1 "DO"
"BEGIN"
      REA[I] := REA[ I]/NN;
      IMA [I] := IMA [I]/NN;
"END ";
       "FOR" P := 0 "STEP" 1 "UNTIL" NLESS1 "DO"
"BEGIN"
       K :=P;
      BACK:  SUM :=0;
      "FOR" I := 1 "STEP" 1 "UNTIL" MLESS1 "DO"
"BEGIN"
      Q := K"DIV" R;
      JJ := K-Q*R;
      SUM := SUM*R+JJ;
      K :=Q;
"END";
      K := SUM*R+K;
      "IF" K<P "THEN" "GOTO" BACK;
      "IF" K "NE" P "THEN"
"BEGIN"
      ATR := REA[K];  ATI := IMA[K];
      REA[K] := REA[P];  IMA[K] := IMA[P];
      REA[P] := ATR;  IMA[P] := ATI;
"END";
"END" OF P LOOP;
"END" OF FFTF;
```

ODDEVEN

PROCEDURE ODDEVEN IS USED WITH FFTT OR FFTF (AND A RADIX OF TWO) TO DISCRETE FOURIER TRANSFORM A REAL TIME SERIES X[I]. I:=0,--------N-1.

CALL:

TO FORWARD TRANSFORM THE N POINT REAL TIME SERIES X[I], STORE X[I] IN THE N/2 COMPLEX ARRAY XR+JXI. I:=0,------N/2. SUCH THAT:-

```
XR[I]:=X[2*K]
 XI[I]:=X[2*K+1]        K:= 0,------N/2-1.
XI[N/2],XR[N/2]        UNASSIGNED WORK SPACE.
THEN CALL:-
                FFTF(2,M-1,N"DIV"2,XR,XI);
                 ODDEVEN(N"DIV"2,XR,XI);
```

ARRAY XR,XI NOW CONTAIN THE FIRST N/2+1 CORRECTLY ORDERED DISCRETE FOURIER COEFFICIENTS.

TO INVERSE TRANSFORM THE DISCRETE FOURIER COFFICIENTS IN XR+JXI CALL:-

```
ODDEVEN(-N"DIV"2,XR,XI);
FFTT(2,M-1,-N"DIV"2,XR,XI);
```

THE REAL TIME SERIES IS NOW IN XR+XI IN THE ODD-EVEN ORDER SPECIFIED FOR THE FORWARD TRANSFORM.

```
"PROCEDURE" ODDEVEN(N,REA,IMA);
"COMMENT" THIS PROCEDURE IS USED WITH FFT TO TRANSFORM REAL DATA;
      "VALUE"N; "INTEGER"N;
      "ARRAY" REA,IMA;
"BEGIN"
      "INTEGER" NN,HALFN,K;
      "REAL" ATR,ATI,BTR,BTI,S,C,TPN;
      NN:="IF" N<0 "THEN" -N "ELSE" N;
      HALFN:=NN"DIV"2;
      TPN:=-3.14159265359/NN;
      "IF" N>0 "THEN"
"BEGIN"
      REA[NN]:=REA[0];
      IMA[NN]:=IMA[0];
"END";
      "FOR" K:=0 "STEP" 1 "UNTIL" HALFN "DO"
"BEGIN"
      ATR:=REA[K]+REA[NN-K];
      ATI:=IMA[K]-IMA[NN-K];
      "IF" N>0 "THEN"
"BEGIN"
      BTR:=(IMA[K]+IMA[NN-K])*0.25;
      BTI:=(REA[NN-K]-REA[K])*0.25;
      S:=SIN(TPN*K); ATR:=ATR*0.25;
      C:=COS(TPN*K); ATI:=ATI*0.25;
"END"
      "ELSE"
"BEGIN"
      BTR:=REA[K]-REA[NN-K];
      BTI:=IMA[K]+IMA[NN-K];
      S:=COS(TPN*K);
      C:=SIN(TPN*K);
"END";
      REA[K]:=ATR+C*BTR-S*BTI;
      REA[NN-K]:=ATR-C*BTR+S*BTI;
      IMA[K]:=ATI+S*BTR+C*BTI;
      IMA[NN-K]:=-ATI+S*BTR+C*BTI;
"END";
"END" OF ODDEVEN;
```

APPENDIX 2

The Covariance of Sample Cross-spectral Estimators

Let $x(t)$, $y(t)$, $p(t)$, $z(t)$. $\left(-\frac{T}{2} \leqslant t < \frac{T}{2}\right)$, be samples of gaussian, zero mean, signals. The sample cross-spectral estimators of $x(t)$ leading $y(t)$, and $p(t)$ leading $z(t)$ are :-

$$Cxy(\omega) = \frac{1}{q} \int_{-\infty}^{+\infty} x(t)\, 1(t)\, e^{+j\omega t}\, dt \,.\, \int_{-\infty}^{+\infty} y(t)\, 1(t)\, e^{-j\omega t}\, dt \quad *$$

$$Cpz(\omega) = \frac{1}{q} \int_{-\infty}^{+\infty} p(t)\, 1(t)\, e^{+j\omega t}\, dt \,.\, \int_{-\infty}^{+\infty} z(t)\, 1(t)\, e^{-j\omega t}\, dt \quad \dagger$$

The covariance of these cross-spectral estimators at frequencies ω_1 and ω_2 is given by :-

$$\mathrm{Cov}\left\{Cxy(\omega_1), Cpz(\omega_2)\right\} = E\left\{Cxy(\omega_1), Cpz(\omega_2)\right\} - E\left\{Cxy(\omega_1)\right\} E\left\{Cpz(\omega_2)\right\}$$

Now :-

$$E\left\{Cxy(\omega_1)\; Cpz(\omega_2)\right\} = \frac{1}{q^2} \int_{-\infty}^{+\infty} \int\int\int\int dt_1 dt_2 dt_3 dt_4 \left\{ A \,.\, e^{-j\omega_1(t_2-t_1)-j\omega_2(t_4-t_3)} \right\}$$

..... A.2.1.

where $A = 1(t_1)1(t_2)1(t_3)1(t_4)\; E\left\{x(t_1)y(t_2)p(t_3)z(t_4)\right\}$

and $E\left\{x(t_1)y(t_2)p(t_3)z(t_4)\right\} = \gamma_{xp}(t_3-t_1)\,\gamma_{yz}(t_4-t_2)$

$+\gamma_{xz}(t_4-t_1)\,\gamma_{yp}(t_3-t_2) + \gamma_{xy}(t_2-t_1)\,\gamma_{pz}(t_4-t_3)$ ..... A.2.2.

* See page 212 for a definition of q.

The covariance can be obtained by substituting the three term expansion of A.2.2. into equation A.2.1., and integrating out each term. The first two terms require the use of the following approximation :-

$$L(\omega) * \left\{ L(\omega) . e^{-j\omega t} \right\} \simeq \left\{ L(\omega) * L(\omega) \right\} . e^{-j\omega t}$$

This approximation holds provided $L(\omega)$ is impulse-like compared with the cross-spectra. The covariance expression obtained using this approximation is :-

$$\mathrm{Cov}\left\{ Cxy(\omega_1), Cpz(\omega_2) \right\} =$$

$$= \frac{1}{(2\pi q)^2} \left\{ \left| L(\omega_1 - \omega_2) * L(\omega_1 - \omega_2) \right|^2 . \Gamma_{xz}(\omega_1) . \Gamma_{py}(\omega_1) \right.$$

$$\left. + \left| L(\omega_1 + \omega_2) * L(\omega_1 + \omega_2) \right|^2 . \Gamma_{xp}(\omega_1) . \Gamma_{zy}(\omega_1) \right\} \quad \ldots\ldots\ldots\ldots \text{A.2.3.}$$

When $\omega_1 = \omega_2$, this expression can be written in the modified form :-

$$\mathrm{Cov}\left\{ Cxy(\omega), Cpz(\omega) \right\} = \frac{1}{(2\pi q)^2} \left\{ L^2(\omega) * \Gamma_{xz}(\omega) . L^2(\omega) * \Gamma_{py}(\omega) \right.$$

$$\left. + \left| L(2\omega) * L(2\omega) \right|^2 \Gamma_{xp}(\omega) \Gamma_{zy}(\omega) \right\} \quad \ldots\ldots \text{A.2.4.}$$

The spectral window functions, $L(\omega)$, have low pass characteristics. The class of functions used in this thesis have pass-bands that are of the order of $\frac{1}{T}$ Hz wide, and out-side their pass-bands their responses fall away as $(\omega)^{-n}$. The width of the pass-band and the decay rate are determined by the shape of the data window $l(t)$.

A simplified covariance expression may be obtained by assumming that the cross-spectra $\Gamma_{xz}(\omega)$ and $\Gamma_{py}(\omega)$ are flat over the pass-band of the spectral window function. The expression can be further simplified by noting that for frequencies greater than the window pass-band the second term in equation A.2.4. is much smaller than the first term.

The simplified covariance expression is :-

$$\text{Cov}\left\{C_{xy}(\omega), C_{pz}(\omega)\right\} = \Gamma_{xz}(\omega).\Gamma_{py}(\omega) \qquad \ldots\ldots\ldots\ldots \text{A.2.5.}$$

The covariance expressions A.2.3. and A.2.4. are general, and can be applied to any window functions, $1(t)$ which satisfies the conditions :-

$$1(t) = 1(-t)$$

$$1(t) = 0 \quad ; \quad |t| > \frac{T}{2}$$

$$q = \int_{-\infty}^{+\infty} 1^2(t)dt$$

As an example of the use of the general covariance expression, the covariance relation for a specific window function will be derived. A well known result is that given by Jenkins and Watts on page 415 of "Spectral Analysis and its Application" (Holden Day, 1968). This gives the covariance for cross-spectral estimators obtained using a rectangular data window. This covariance expression may be obtained from the general covariance expression A.2.3. as follows :-

The appropriate data window is the rectangular function $l_R(t)$, defined by :-

$$l_R(t) = (T)^{-\frac{1}{2}} \quad ; \quad |t| < \frac{T}{2}$$

$$l_R(t) = 0 \quad ; \quad |t| > \frac{T}{2}$$

$$q_R = \int_{-\infty}^{+\infty} l_R^2(t)dt = 1$$

The Fourier transform of $l_R(t)$ is denoted $L_R(\omega)$. In expression A.2.3. we require the function $L_R(\omega)*L_R(\omega)$, this may be evaluated indirectly by determining its Fourier transform first. Thus :-

$$L_R(\omega)*L_R(\omega) = 2\pi \int_{-\infty}^{+\infty} l_R(t)\, l_R(t)\, e^{-j\omega t}\, dt = \frac{2\pi}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} e^{-j\omega t}\, dt$$

$$= 2\pi \frac{\sin\left\{\frac{\omega T}{2}\right\}}{\left\{\frac{\omega T}{2}\right\}}$$

The covariance expression for rectangular windowing is thus obtained as :-

$$\mathrm{Cov}\left\{ Cxy(\omega_1), Cpz(\omega_2)\right\} =$$

$$= \left\{ \frac{\sin\left\{\frac{T(\omega_1-\omega_2)}{2}\right\}}{\left\{\frac{T(\omega_1-\omega_2)}{2}\right\}} \right\}^2 \Gamma xz(\omega_1)\ \Gamma py(\omega_1) + \left\{ \frac{\sin\left\{\frac{T(\omega_1+\omega_2)}{2}\right\}}{\left\{\frac{T(\omega_1+\omega_2)}{2}\right\}} \right\}^2 \Gamma xp(\omega_1)\ \Gamma zy(\omega_1)$$

APPENDIX 3

Expectation and Covariance of Aliased Cross-Spectral Estimators

Let $X(\omega)$ be the continuous Fourier transform of the sample $x(t)$ $\left\{-\frac{T}{2} \leqslant t < \frac{T}{2}\right\}$ of a gaussian signal modified by the window function $l(t)$ (defined in Chapter 4). Also let A_k be the discrete Fourier transforms of the time series x_i $\left\{i = -\frac{N}{2}, \ldots\ldots \frac{N}{2} - 1\right\}$ modified by the window function l_i. Then, if the time series is obtained by sampling $x(t)$ every Δ seconds, the discrete and continuous Fourier transforms are related by :-

$$A_k = (\Delta N)^{-1} \sum_{n=-\infty}^{+\infty} X(\omega_{k-nN}) \qquad ; \qquad \omega_k = \frac{2\pi k}{\Delta N}$$

If the spectrum of $x(t)$ is assumed to be low pass, then to a good approximation all but the first alias can be neglected. Under this assumption we can define the aliased spectrum of $x(t)$ as $X_A(\omega_k) = \Delta N\, A_k$. So that :-

$$X_A(\omega_k) = X(\omega_k) + X^*(\omega_{N-k}) \qquad \ldots\ldots\ldots\ldots \text{A.3.1.}$$

The aliased spectra of the gaussian signal sample $y(t)$, $p(t)$, $z(t)$, may be defined in a similar way as :-

$$Y_A(\omega_k) = Y(\omega_k) + Y^*(\omega_{N-k})$$

$$P_A(\omega_k) = P(\omega_k) + P^*(\omega_{N-k})$$

$$Z_A(\omega_k) = Z(\omega_k) + Z^*(\omega_{N-k})$$

And the aliased cross-spectrum of x(t) leading y(t) may be defined as :-

$$Cxy_A(\omega_k) = \frac{1}{q} X_A^*(\omega_k)\ Y_A(\omega_k) \qquad \ldots\ldots\ldots\ldots \text{A.3.2.}$$

Similarly the aliased sample cross-spectrum of p(t) leading z(t) is :-

$$Cpz_A(\omega_k) = \frac{1}{q} P_A^*(\omega_k)\ Z_A(\omega_k) \qquad \ldots\ldots\ldots\ldots \text{A.3.3.}$$

a) The Expected Value

The expected value of the aliased sample cross-spectrum is :-

$$E\left\{Cxy_A(\omega_k)\right\} = \frac{1}{q} E\left\{X_A^*(\omega_k)\ Y_A(\omega_k)\right\}$$

$$\simeq \frac{1}{2\pi q}\left\{L^2(\omega_k) * \Gamma xy(\omega_k) + L^2(\omega_{N-k}) * \Gamma yx(\omega_{N-k})\right\}$$

Assuming the cross-spectrum is smooth over the pass-band of the spectral window, the approximate expected value of the aliased sample cross-spectrum is :-

$$E\left\{Cxy_A(\omega_k)\right\} \simeq \Gamma xy(\omega_k) + \Gamma yx(\omega_{N-k}) \qquad \ldots\ldots\ldots \text{A.3.4.}$$

b) Covariance

The covariance of the aliased sample cross-spectra of x(t) leading y(t), and p(t) leading z(t) is given by :-

$$Cov\left\{Cxy_A(\omega_k), Cpz_A(\omega_{N-k})\right\} = E\left\{Cxy_A(\omega_k).Cpz_A(\omega_{N-k})\right\} - E\left\{Cxy_A(\omega_k)\right\} E\left\{Cpz_A(\omega_{N-k})\right\}$$

This expression can be evaluated in a way similar to that used in appendix 2. Making the usual approximation that the cross-spectra are smooth in the pass-band of the spectral window, the approximate covariance expression is :-

$$\mathrm{Cov}\left\{ \mathrm{Cxy}_A(\omega_k), \mathrm{Cpz}_A(\omega_k) \right\} \simeq \Gamma_{xy}(\omega_k).\Gamma_{py}(\omega_k) + \Gamma_{yp}(\omega_{N-k}).\Gamma_{zx}(\omega_{N-k}) + \Gamma_{xp}(\omega_k).\Gamma_{yz}(\omega_{N-k}) + \Gamma_{px}(\omega_k).\Gamma_{zy}(\omega_{N-k}) \quad \ldots\ldots\ldots\ldots \text{A.3.5.}$$

The covariance expression for smoothed, aliased, cross-spectral estimators, $\bar{C}xy_A(\omega)$, $\bar{C}pz_A(\omega)$, obtained by averaging n independent sample, aliased, cross-spectral estimators can be written directly as :-

$$\mathrm{Cov}\left\{ \bar{C}xy_A(\omega_k), \bar{C}pz_A(\omega_k) \right\} = \frac{1}{n} \mathrm{Cov}\left\{ \mathrm{Cxy}_A(\omega_k), \mathrm{Cpz}_A(\omega_k) \right\} \quad \ldots\ldots\ldots \text{A.3.6.}$$

From equations A.3.4. and A.3.5. the variance expressions for smoothed, aliased, auto- and cross-spectral estimates can be written in terms of their expected values. Thus :-

$$\mathrm{Var}\left\{ \bar{C}xx_A(\omega_k) \right\} \simeq \frac{1}{n} E^2\left\{ \bar{C}xx_A(\omega_k) \right\}$$

$$\mathrm{Var}\left\{ \bar{C}xy_A(\omega_k) \right\} \simeq \frac{1}{n} \left\{ E^2\left\{ \bar{C}xy_A(\omega_k) \right\} + 2\,\Gamma_{xx}(\omega_k)\,\Gamma_{yy}(\omega_{N-k}) - \Gamma_{xy}(\omega_k)\,\Gamma_{yx}(\omega_{N-k}) \right\} \quad \ldots\ldots\ldots\ldots \text{A.3.7.}$$

From these expression we conclude that aliasing will

modify the variability of cross-spectral estimators, but not auto-spectral estimators. The extent to which the variance of cross-spectral estimators is altered can be judged by considering the behaviour of expression A.3.7. as the folding frequency, $\frac{\omega_s}{2}$ is approached. Near the folding frequency we are justified in making the approximation $\omega_k = \omega_{N-k}$. Substituting for ω_{N-k} in A.3.7. gives :-

$$\mathrm{Var}\left\{\bar{C}xy_A(\omega_k)\right\} \doteq \frac{1}{n}\left\{E^2\left\{\bar{C}xy_A(\omega_k)\right\} + 2\Gamma_{xx}(\omega_k)\,\Gamma_{yy}(\omega_k)\left\{1-K^2_{xy}(\omega_k)\right\}\right\} \qquad \text{.....A.3.8.}$$

Therefore, since the squared coherency is bounded by 0 and 1, the variability of cross-spectral estimates is generally increased by aliasing.

APPENDIX 4

Bias and Variance of Non-Linear Functions of Random Variables

Suppose $f(x_1,x_2, \ldots x_n)$ is a non-linear function of the random variables $X_1,X_2, \ldots X_n$. Approximate expressions for the bias and variance of $f(x_1,x_2, \ldots x_n)$ in terms of the bias and covariances of the random variables may be obtained from a Taylor series expansion. To a first order approximation, $f(x_1,x_2, \ldots x_n)$ in the region of $p_1,p_2, \ldots p_n$ is given by :-

$$f(x_1,x_2, \ldots x_n) \simeq f(p_1,p_2 \ldots p_n) + \sum_{i=1}^{n} \left(\frac{\delta f}{\delta x_i}\right)_p (x_i-p_i) \qquad \ldots\ldots\ldots \text{A.4.1.}$$

a) Bias

If the random variables $x_1,x_2 \ldots x_n$ are associated with estimators of the parameters $p_1,p_2, \ldots p_n$, then the bias in the estimator associated with $f(x_1,x_2, \ldots x_n)$ is approximately :-

$$B\left\{f(x_1,x_2, \ldots x_n)\right\} \simeq \sum_{i=1}^{n} \left(\frac{\delta f}{\delta x_i}\right)_p B\left\{x_i\right\} \qquad \ldots\ldots\ldots \text{A.4.2.}$$

b) Variance

If the expected value of $x_1,x_2, \ldots x_n$ is denoted $\mu_1,\mu_2, \ldots \mu_n$, then the first order approximation about the mean is :-

$$f(x_1,x_2, \ldots x_n) - f(\mu_1,\mu_2, \ldots \mu_n) \simeq \sum_{i=1}^{n} \left(\frac{\delta f}{\delta x_i}\right)_\mu (x_i-\mu_i)$$

And the variance is approximately :-

$$\mathrm{Var}\left\{ f(x_1, x_2, \ldots\ldots x_n) \right\} \underline{\Omega} \sum_{i=1}^{n} \sum_{i=1}^{n} \mathrm{Cov}(x_j, x_i) \left(\frac{\partial f}{\partial x_i} \right)_{\mu} \left(\frac{\partial f}{\partial x_j} \right)_{\mu}$$

..... A.4.3.

APPENDIX 5

Errors in Frequency Response Measurements caused by Non-Identical Guard Filters

In section 5.3.3. a technique for the suppression of aliasing errors using a pair of identical guard filters is described. In this appendix we consider the errors that occur when the guard filters do not have identical frequency responses. This situation may be caused by variations in filter component values.

Let us assume that in figure 5.3. the frequency response function of the filter for $x(t)$ is $g_1(\omega)$ and that the frequency response function of the filter for $y(t)$ is $g_2(\omega)$, then neglecting errors caused by windowing and aliasing, the expected value of the frequency response between $x'(t)$ and $y'(t)$ is :-

$$E\{\hat{H}(\omega)\} = \left\{\frac{g_2(\omega)}{g_1(\omega)}\right\} . H(\omega) \qquad \ldots\ldots\ldots \text{A.5.1.}$$

From this equation it can be seen that the fractional bias error in the gain estimate is :-

$$\frac{B\{|\hat{H}(\omega)|\}}{|H(\omega)|} = \frac{|g_2(\omega)|}{|g_1(\omega)|} - 1 \qquad \ldots\ldots\ldots \text{A.5.2.}$$

and also that the bias error in the phase estimate is :-

$$B\{\hat{\phi}(\omega)\} = \text{Arg}\left\{\frac{g_2(\omega)}{g_1(\omega)}\right\} \qquad \ldots\ldots\ldots \text{A.5.3.}$$

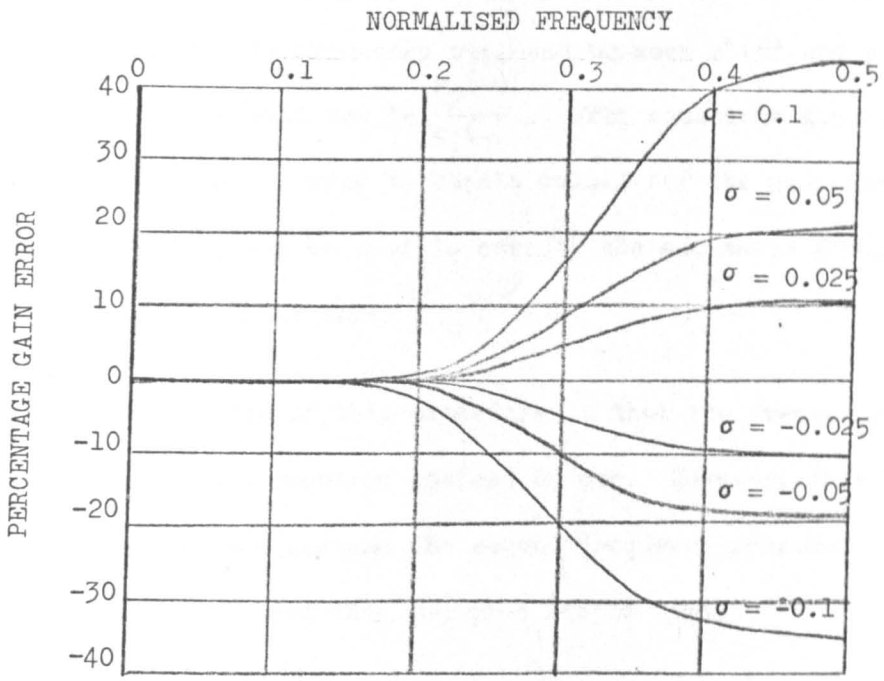
The following simple example illustrates the likely magnitude of these errors. Assume that the guard filters in figure 5.3. have fourth order Butterworth characteristics, but that their cutoff frequencies are slightly different. The guard filter for the input signal x(t) has a cutoff frequency of $0.3f_s$ HZ (the value recommended in section 5.3.3.), and the guard filter for the output signal y(t) has a cutoff frequency of $0.3f_s(1 + \sigma)$HZ (where σ is a small constant). The errors associated with various values of σ have been plotted using equations A.5.2. and A.5.3. and are shown in graphs A.5.1. and A.5.2. The frequency scale on these graphs is normalised so that the sampling frequency is unity and the 0.5 HZ point on the normalised frequency axis therefore denotes the folding frequency.

It can be seen from the graphs that even for small differences between the two guard filters the errors in the gain and phase estimates can be large. It is therefore important that the guard filters have identical responses. In some cases, however, it may be impractical to obtain or construct matched guard filters and in these cases the use of the following procedure is advocated.

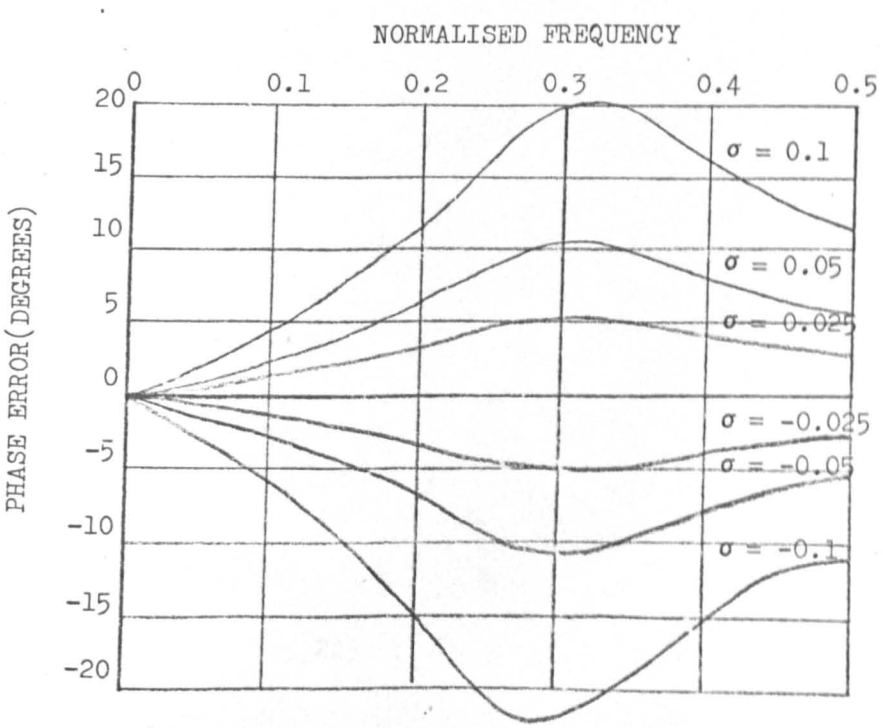
1) Match the guard filters as closely as possible and then estimate the frequency response between x'(t) and y'(t) using one of the measurement schemes shown in figure 5.3. Neglecting windowing and aliasing errors this estimate will have an expectation given by equation A.5.1.

TYPICAL ERRORS IN FREQUENCY RESPONSE ESTIMATES CAUSED BY GUARD FILTERS WITH UNEQUAL CUTOFF FREQUENCIES. (FOURTH ORDER BUTTERWORTH FILTERS; FOR THE INPUT FILTER $f_c = 0.3f_s$, AND FOR THE OUTPUT FILTER $f_c = 0.3f_s(1 + \sigma)$).

GRAPH A.5.1. PERCENTAGE GAIN ERROR



GRAPH A.5.2. PHASE ERROR



2) Disconnect the system under test and replace it with a short-circuit.

3) Estimate the frequency response between x'(t) and y'(t) again. Its expectation will now be $\frac{g_2(\omega)}{g_1(\omega)}$. From equations A.5.2. and A.5.3. this estimate can be used to obtain values for the gain and phase error and these in turn can be used to correct the estimated frequency response of the system under test.

The disadvantage of this procedure is that two frequency response measurements are required instead of one. However, this is not a serious disadvantage because the second frequency response measurement is noise-free and thus occupies little time.