C.P. No. 1225

C.P. No. 1225

MINISTRY OF DEFENCE (PROCUREMENT EXECUTIVE)

AERONAUTICAL RESEARCH COUNCIL

CURRENT PAPERS

Some Experiences with "On-Line" Spectral Analysis Using a Small Digital Computer

By

J. B. Roberts and D. Surry

with an Appendix by

R. F. Johnson

LONDON: HER MAJESTY'S STATIONERY OFFICE
1972
Price £1 35 net

AERONAUTICAL RESEARCH COUNCIL

C P. No 1225

CORRECTION

1 Cover page i and title page

After Author's name insert 'Environmental Unit, N P L '

2 F*ig III 2 should follow* F*ig III 1 immediately before the detachable* Abstract Cards instead of following Fig 2 22

LONDON HER MAJESTY'S STATIONERY OFFICE

October 1972

CP No. 1225*

March 1971

SOME EXPERIENCES WITH "ON-LINE" SPECTRAL ANALYSIS USING A SMALL DIGITAL COMPUTER

by

J.B. Roberts and D. Surry
with an Appendix by
R.F. Johnson

Summary

The capabilities of an "on-line" spectral analysis system, developed at NPL, are described. Two approaches to the digital generation of spectral estimates, the correlation technique and the FFT technique, have been adopted and a comprehensive appraisal of these methods is given. A full description of the physical and performance characteristics of the hardware involved is included.

*Replaces A.R.C.33 097

CONTENTS

NOTATION

$a_x(n)$, $a_y(n)$	Fourier coefficients association with $x(j)$ and $y(j)$ respectively.
B	statistical bandwidth (in Hz)
$\underset{\sim}{c}(n)$	Fourier transform of $\underset{\sim}{Z}(j)$
$C_{xy}(f)$	co-spectrum
E{ }	expectation, or ensemble averaging operation
f	frequency (in Hz)
$G_{xy}(f)$	cross-spectral density function (auto-spectral if $x = y$)
i	$\sqrt{-1}$
j,k,n,r,p	integer indices
L	number of samples in one block of data (in FFT method)
m	maximum number of lag intervals
m_x, m_y	mean values of $\underset{\sim}{X}(t)$ and $\underset{\sim}{Y}(t)$, respectively
M	block size index $(L = 2^M)$
n_B	number of blocks
n_E	number of final spectral estimates
N	total number of samples in one channel of data
NDF	number of degrees of freedom
q	number of Hanned periodogram estimates per final spectral estimate (in FFT method)
$Q_{xy}(f)$	quad-spectrum
r_s	sampling rate (in numbers/sec)
r_m	maximum allowable sampling rate (in correlation method)
$R_{xy}(\tau)$	cross-correlation function (auto-correlation function if $x = y$)
t	time (in secs)
T	total sampling time
T(m)	correlation program cycle time = $1/r_m$
w(j)	data window weightings

$\underset{\sim}{x}(t)$, $\underset{\sim}{y}(t)$	random processes with zero mean
$x(t)$, $y(t)$	realisations of $\underset{\sim}{x}(t)$ and $\underset{\sim}{y}(t)$, respectively
$x(j)$, $y(j)$	equi-spaced samples of $x(t)$ and $y(t)$ respectively
$\underset{\sim}{X}(t)$, $\underset{\sim}{Y}(t)$	random processes with non-zero mean
$X(t)$, $Y(t)$	realisations of $\underset{\sim}{X}(t)$ and $\underset{\sim}{Y}(t)$, respectively
$\underset{\sim}{z}(j)$	complex random process (= $\underset{\sim}{x}(j) + i\,\underset{\sim}{y}(j)$)
$\underset{\sim}{Z}_x(f)$, $\underset{\sim}{Z}_y(f)$	complex random processes, related to $\underset{\sim}{x}(t)$ and $\underset{\sim}{y}(t)$
$\alpha(f)$	spectral window
$\hat{\sigma}_x$	estimate of the sample variance of x
τ	time delay
$\Delta\tau$	sampling interval (or lag interval)
μ	ratio of statistical bandwidth to spacing between final estimates

Supercripts

$\hat{}$	estimators
H, $\hat{\hat{}}$	Hanned estimators

Abbreviations

ADC	analogue to digital converter
AFD	auto-spectral FFT program using floating point arithmetic and DAP subroutines
AFF	auto-spectral FFT program using floating point arithmetic and FORTRAN subroutines
AID	auto-spectral FFT program using integer arithmetic and DAP subroutines
AIS	analogue input system (ADC plus sample and hold/multiplexer system
CID(E)	cross-spectral FFT program using integer arithmetic and DAP subroutines
CORDI	DAP program for mode I of direct correlation technique (I = 1,2 or 3)
CORF	FORTRAN program for mode 1 of direct calculation technique
DAP	Honeywell assembly language
DMC	direct multiplex control
FFT	Fast Fourier transform
HSA	high speed arithmetic
NDF	number of degrees of freedom

1. INTRODUCTION

Power and cross-spectral analysis has become an increasingly popular tool over recent years. In particular, it has found many applications in the fields of turbulence and structural vibration. This report serves to document the current capabilities of a system which has been developed in response to these problems. It includes a description of the physical and performance characteristics of the hardware involved in operating on-line to remote (about 350 metres) experiments. Furthermore, it describes and compares in detail two alternative methods of obtaining spectral estimates which have been developed on the computer used in the system. The first uses the traditional correlation approach whilst the second applies fast Fourier transform (FFT) techniques directly to the data.

There are many means available for spectral data analysis. Some of the choices open to the experimentalist are shown in figure 1.1. Inevitably, the more sophisticated systems tend to be of higher cost but require less time to produce more accurate results. For instance, the introduction of an analogue tape recorder reduces the bandwidth requirements (and hence the cost of the following analysis system) due to its ability to change the the frequency range of the data. However, this slows the turn-around time of the system and increases the noise level. The most flexible method of handling the data is to digitise it and manipulate it in a computer. However, again, the interface between the analogue to digital converter (ADC) and the computer can vary. One can choose either to use an intermediate storage device, or the ADC can interface directly with the computer memory. The former method is most suitable for a large computer operating on a batch processing basis, whereas the latter method currently requires operator access to the computer. This generally implies a small capacity device. The special purpose black boxes, which are currently commercially available, serve to reduce the large data records to relatively small time-lagged correlation records. These can then be transformed, either by another special purpose device or by a straightforward computer operation. It is important that such correlation points are sufficiently accurate* to

*Accuracy here implies two different aspects. Firstly, the number of decades range of power will be inherently limited by the quantization noise (see reference 1) arising from the resolution of the digitisation. Secondly, the power estimates are sensitive to any noise superimposed on the correlation estimates. Devices with analogue stores for the correlation estimates are particularly prone to this problem.

provide acceptable spectral estimates. The evolution of spectral analysis systems seems to be towards a special purpose digital analyser in which the sampling and computation are accomplished under conditions optimized for these specialized computations. Such systems are just now becoming commercially available.

The system discussed in this report was designed to exploit an available small computer (namely a Honeywell DDP 516) equipped with a multi-channel ADC. The system is intended to provide spectral analysis of experimental data originating in laboratories some distance from the computer facility. This requires the use of data links over a distance of approximately 350 metres. In this case the links are cables carrying the data in analogue form. A brief description of this part of the system - essentially that of the hardware other than the computer - is given in section 3, together with some performance characteristics.

The software approach to the data analysis includes two diverse approaches to spectral computation. The first computes lagged product correlations in real time - i.e. all the arithmetic is accomplished between successive samples. The second method samples blocks of data and uses FFT techniques to develop spectral estimates during pauses between successive block sampling. Spectral estimates are derived directly without producing intermediate correlation functions. Both of these methods are described and compared in section 2.

Necessarily, the absolute performance characteristics of the computations are peculiar to the machine used, so for the record, the pertinent characteristics of the computer employed in this work are given in section 3.

2. SOFTWARE

2.1 Real time correlation technique

Consider two channels of data, $X(t)$ and $Y(t)$, each of duration $0 < t < T$. If the conditions of the experiment are steady, we can regard $\underset{\sim}{X}(t)$ and $\underset{\sim}{Y}(t)$ as realisations, or sample functions, of two stationary random processes $X(t)$ and $Y(t)$, respectively. We can write

$$\begin{aligned}\underset{\sim}{X}(t) &= m_x + \underset{\sim}{x}(t)\\ \underset{\sim}{Y}(t) &= m_y + \underset{\sim}{y}(t)\end{aligned} \qquad \ldots (2.1)$$

where $m_x = E\{X(t)\}$ and $m_y = E\{Y(t)\}$*. The processes $\underset{\sim}{x}(t)$ and $\underset{\sim}{y}(t)$

*$E\{\cdot\}$ denotes the "expectation", or "ensemble averaging operator".

have zero mean and their cross-correlation function is defined as

$$R_{xy}(\tau) \quad = \quad E\{\underset{\sim}{x}(t)\underset{\sim}{y}(t+\tau)\} \qquad \ldots (2.2)$$

The physically realisable one-sided cross-spectral density of $\underset{\sim}{x}(t)$ and $\underset{\sim}{y}(t)$, $G_{xy}(f)$, and $R_{xy}(\tau)$ are a Fourier transform pair. Following reference 1, we have

$$G_{xy}(f) \quad = \quad C_{xy}(f)-iQ_{xy}(f) \quad = \quad 2\int_{-\infty}^{\infty} R_{xy}(\tau)e^{-i2\pi f\tau}d\tau \qquad \ldots (2.3)$$

$C_{xy}(f)$ and $Q_{xy}(f)$ are, respectively, the real and imaginary parts of the cross-spectrum, otherwise known as the co- and quad-spectra. We note that the above reduces to the relation between power spectra and auto-correlation functions when $R_{xy}(\tau) = R_{xx}(\tau)$, ($R_{xx}(\tau)$ must be an even function).

In practice we have to deal with finite records sampled at equi-spaced intervals. Consider the set of 2N data values, $x(j)$, $y(j)$, $j = 0, 1, \ldots., N - 1$. We can obtain a sample of the unbiased correlation estimator

$$\left.\begin{aligned}\hat{R}_{xy}(r) &= \frac{1}{N-r}\sum_{n=1}^{N-r} \underset{\sim}{x}(n)\ \underset{\sim}{y}(n+r)\\ &= \frac{1}{N-r}\sum_{n=r+1}^{N} \underset{\sim}{x}(n-r)\ \underset{\sim}{y}(n)\\ &\qquad\qquad (r=0,1,\ldots.m)\end{aligned}\right\} \qquad \ldots (2.4)$$

without difficulty, and we note that negative time delays are most conveniently handled by using the fact that $R_{x\underset{\sim}{y}}(-r) = R_{yx}(r)$. A discrete form of equation (2.3) can then be applied to $R_{xy}(r)$ to yield an estimator for $G_{xy}(f)$. The appropriate formula is

$$\left.\begin{aligned}\hat{C}_{xy}(k) &= 2\left[\hat{A}(0)+2\sum_{r=1}^{m-1}\hat{A}(r)\cos\left(\frac{\pi rk}{m}\right)+\hat{A}(m)\cos(\pi k)\right]\\ \hat{Q}_{xy}(k) &= 2\left[2\sum_{r=1}^{m-1}\hat{B}(r)\sin\left(\frac{\pi rk}{m}\right)+\hat{B}(m)\sin(\pi k)\right]\\ &\qquad\qquad (k=0,1,\ldots\ldots.,n)\end{aligned}\right\} \ldots (2.5)$$

where

$$\hat{A}(r) = \tfrac{1}{2}\left[\hat{R}_{xy}(r)+\hat{R}_{yx}(r)\right]$$
$$\hat{B}(r) = \tfrac{1}{2}\left[\hat{R}_{xy}(r)-\hat{R}_{yx}(r)\right]$$

Here $\hat{G}_{xy}(k)$ is an estimator of the cross-spectrum at frequency

$$f_k = \frac{k}{2m}\,(\mathrm{Hz}) \qquad \ldots (2.6)$$

Note in the above and in the following argument it has been convenient to take the value of $\Delta\tau$ as unity. This simply affects the frequency scaling - i.e. it defines the Nyquist or folding frequency, $f_n = 0.5$. Multiplication of the frequencies of $1/\Delta\tau$ and the spectral values by $\Delta\tau$ will restore them to their scaled values.

It is fairly easy to show that (e.g. see reference 2)

$$E\{\hat{G}_{xy}(k)\} = \int_0^\infty G_{xy}(f)\{\alpha(f_k-f)+\alpha(f_k+f)\}df \qquad \ldots (2.7)$$

The "filter operator", or "spectral window", $\alpha(f)$, has a (sin x/x) form, as shown in figure 2.1(a) and has the property that

$$\int_{-\infty}^{\infty} \alpha(f)df = 1 \qquad \ldots (2.8)$$

To reduce undesirable "leakage", it is beneficial to smooth the $\hat{G}_{xy}(k)$ estimates, using the well-known Hanning technique. "Hanned" estimators are found from

$$\left.\begin{aligned}
\hat{\hat{G}}_{xy}(0) &= 0.5\hat{G}_{xy}(0)+0.5\hat{G}_{xy}(1)\\
\hat{\hat{G}}_{xy}(k) &= 0.25\hat{G}_{xy}(k-1)+0.5\hat{G}_{xy}(k)+0.25\hat{G}_{xy}(k+1)\\
&\qquad\qquad (k=1,2,\ldots\ldots,m-1)\\
\hat{\hat{G}}_{xy}(m) &= 0.5\hat{G}_{xy}(m-1)+0.5\hat{G}_{xy}(m)
\end{aligned}\right\} \qquad \ldots (2.9)$$

This operation has the effect of modifying the spectral window, to the shape shown in figure 2.1(b). It will be noted that the side lobes are greatly reduced and in fact fall off like f^{-3}, rather than as f^{-1}.

A convenient measure of the variability of the $\hat{\hat{G}}_{xy}(k)$ estimator is given by its number of degrees of freedom (NDF). It can be shown that, (see reference 1), approximately

$$\mathrm{NDF} = \frac{2N}{m} \qquad \ldots (2.10)$$

when m is small compared with N. A large value of N implies small statistical uncertainty. It would seem advisable, therefore, to have

$$m << N$$

However, the statistical bandwidth of the spectral window, B (see reference 1), shown in figure 2.1(b) is approximately

$$B = \frac{1.3}{m} \qquad \ldots (2.11)$$

and it can be shown that (see reference 1) the bias error, defined as

$$\mathrm{bias}\{\hat{G}_{xy}(k)\} = G_{xy}(k)-E\{\hat{G}_{xy}(k)\} \qquad \ldots (2.12)$$

is approximately

$$\frac{B^2}{24}\left[\frac{d^2}{df^2} G_{xy}(f)\right]_{f=f_k} \qquad \ldots (2.13)$$

Thus high frequency resolution and small bias error can only be achieved if m is large. Clearly, the choice of a suitable value of m is a matter of compromise.

The basis of the real time correlation approach is that for any sample of one process, x(n) say, the products contributing to (2.4) require only samples of the other process, y(t), from y(n) to y(n-m). Hence, if all the relevant arithmetic can be done between successive samples, then only the most recent m + 1 samples of each data channel need be stored at any one time. This store is updated every time a new sample is taken.

The program now in use (CORD) was designed to operate in 3 modes. The first mode (CORD1) provides auto-correlations of one channel ($R_{xx}(r)$, r = 0, ..., m) Mode 2 (CORD2) provides both sides of the cross-correlation of two channels ($R_{xy}(r)$, $R_{yx}(r)$, r = 0, ..., m). Mode 3 (CORD3) provides both auto-correlations and complete cross-correlation of two channels ($R_{xx}(r)$, $R_{yy}(r)$, $R_{xy}(r)$, $R_{yx}(r)$, r = 0, ... m). For each mode the real time sampling rate (r_s) has a maximum (r_m) determined by the computation time required between samples. For example, in its simplest form, mode 1 would be required to perform the following operations between each sample. Firstly the array of $m + 1$ samples is updated by discarding the oldest sample and replacing it by the most recent, say $x(n)$. (If the array is thought of as circular, this merely rotates the present right around the array once every $m + 1$ samples). Then $m + 1$ products of the form $x(n - r)x(n)$ are formed and each is added to the appropriate running store. Lastly, $x(n)$ is added to a running store, $\Sigma x(n)$, which is used for mean removal. (The data variance is obtained from the value of $R(0)$). Modes 2 and 3 require more inter-sample computation and hence have a lower r_m. In practice, because of these restrictions on r_m, mode 1 was actually written to form a one-sided cross-correlation between two channels. If the two channels are identical this reduces to the auto-correlation, but it also provides the option of a fast cross-correlation mode whereby the two sides of the cross-correlation function are carried out sequentially rather than simultaneously, as in modes 2 and 3. This modification causes a slight increase in computation time for mode 1 due to the additionally required running sums, $\Sigma y(n)$, $\Sigma x^2(n)$ and $\Sigma y^2(n)$. An illustrative FORTRAN version of mode 1 (CORF) is shown in Appendix 1.

At the completion of sampling, the running sums are translated into final estimates corrected for mean by the identities

$$\hat{m}_x = \frac{1}{N}\sum_{n=1}^{N} x(n); \quad \hat{m}_y = \frac{1}{N}\sum_{n=1}^{N} y(n)$$

$$\hat{R}_{xy}(r) = \frac{1}{N}\left[\sum_{n=1}^{N} x(n-r)y(n)\right] - \hat{m}_x\hat{m}_y$$

$$\hat{\sigma}_x^{\,2} = \frac{1}{N-1}\left\{\sum_{n=1}^{N} x^2(n) - N\hat{m}_x^{\,2}\right\}$$

$$\hat{\sigma}_y^{\,2} = \frac{1}{N-1}\left\{\sum_{n=1}^{N} y^2(n) - N\hat{m}_y^{\,2}\right\}$$

The divisor in the equation for $\hat{R}_{xy}(r)$ is N rather than $N-r$ because the program cycles through the first $m + 1$ samples to fill the arrays before starting the intersample computation of the running sums.

To provide the fastest possible sampling rates for a given m, the final program for the intersample calculations was written in DAP-16* by Richard Young.† To further improve the speed, integer arithmetic was employed on data, truncated to 8 bits - i.e., restricted in range to $\pm 2^7$. Multiplications were performed and the running sums were stored in double precision integer words, which for the Honeywell means allowable integers up to $\pm 2^{30}$. Hence, if all products are of maximum size, the allowable number of such products is $2^{16} = 65,536$. This requirement (for no overflow) limits the maximum number of samples per record that can be taken. However, for all practical purposes, allowing a conservative crest value of two for the data, the maximum running sum (the mean square) would not overflow for less than 260,000 samples. This does not usually apply a restriction to an analysis.

The performance curves for the three modes of CORD, as written in DAP, are shown in figure 2.2 as functions of the number of lags. For comparison, a curve is also shown for the FORTRAN floating point version (CORF) of Appendix 1. It can be seen that the DAP programs reduce the cycle time by a factor of 17 for 2 lags and 20 for 200 lags. An exactly parallel FORTRAN version to the DAP program cannot be developed because of the lack of a FORTRAN double precision integer command. However, various other FORTRAN approaches have been made to try and improve the version of Appendix 1 but with little success. For instance, a FORTRAN program has been written which reduces the data to 8 bits by dividing by 256, and then utilizes all integer multiplications. The running sums are stored as floating point words. The speed of this program is amplitude dependent but is up to twice as fast as that of Appendix 1 for low amplitude inputs. The only significant improvement that appears feasible could be achieved by setting the ADC to take only 8 bits (which is already possible), and then interfacing it to hand over these 8 bits in the least significant part of the computer's integer word, rather than in the most significant part, which is the present practice. This would avoid the necessity of the initial integer division by 256.

---

*DAP 16 is the assembly language used on the Honeywell DDP 516.

†Richard Young did this work while at NPL. His present address is Micro Consultants, Newbury.

Obviously, the lagged product method can be extended to any sampling rate simply by completing the sampling before undertaking the computation of lagged products. Since the number of computations required is similar, regardless of the approach, and since the ADC sampling time is a small part of the program's cycle time, an approximation to the total time required in the block sampling method can be derived from the boundaries of the real time method shown in figure 2.2. These boundaries occur when all the intersample time is required for the m computations. Thus, the order of calculation time required for the block method is simply $N\ T(m)$ where $T(m)$ is the cycle time established from figure 2.2 as the reciprocal of r_m. Of course, the total program time will require in addition the sampling time for N samples. This would have to be less than $N\ T(m)$ for this approach to be required.

Since we were primarily interested in the limitations of this method in terms of maximum allowable sampling rate (i.e., the results of figure 2.2), no attempt was made to optimise the transform by using FFT techniques. Instead, an existing FORTRAN subroutine (reference 3) was employed, which accepts two one-sided correlation arrays (either $R_{xx}(r)$ and $R_{yy}(r)$ or $R_{xy}(r)$ and $R_{yx}(r)$) and outputs the relevant Hanned spectral estimates. It is hence very inefficient for mode 1 operation. For illustration, the transformation time for one or two auto-correlations on the DDP-516 is given approximately by $0.0045m^2$ seconds. The results of section 2.2 suggest that an FFT approach would reduce this to less than 5 seconds for anything up to 250 lags.* Thus, in the comparison of the two approaches in section 2.3, the transform time is assumed negligible.

2.2 Direct FFT technique

As in the previous section, consider two processes, $\underset{\sim}{x}(t)$ and $\underset{\sim}{y}(t)$. These can be represented in the form

$$\begin{aligned} \underset{\sim}{x}(t) &= \int_{-\infty}^{\infty} e^{-i2\pi ft} d\underset{\sim}{Z}_x^*(f) \\ \underset{\sim}{y}(t) &= \int_{-\infty}^{\infty} e^{i2\pi ft}\ d\underset{\sim}{Z}_y(f) \end{aligned} \qquad \ldots (2.14)$$

where $\underset{\sim}{Z}_x(f)$ and $\underset{\sim}{Z}_y(f)$ are complex random processes non-differentiable

---

*For this purpose the array of correlation estimates to be transformed must contain 2^Q elements, where Q is a positive integer. If this is not so, the array must be "padded" with zeros.

with respect to f (e.g. see reference 4). The correlation function $R_{xy}(\tau)$ then becomes, from equation (2.2),

$$R_{xy}(\tau) \;=\; E\left\{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-i2\pi f_1 t}\, dZ^*_{\underset{\sim}{x}}(f_1)\, e^{i2\pi f_2(t+\tau)}\, dZ_{\underset{\sim}{y}}(f_2)\right\}$$

and since, for stationary processes (see reference 4)

$$E\{dZ^*_{\underset{\sim}{x}}(f_1)\,dZ_{\underset{\sim}{y}}(f_2)\} \;=\; 0$$

when the corresponding intervals df_1, df_2 are non-overlapping, the above equation reduces to

$$R_{xy}(\tau) \;=\; \int_{-\infty}^{\infty} e^{i2\pi f\tau} E\{dZ^*_{\underset{\sim}{x}}(f)\,dZ_{\underset{\sim}{y}}(f)\} \qquad \ldots (2.15)$$

On comparing this result with equation (2.3) we see that

$$\begin{aligned} G_{xy}(f) &= C_{zy}(f) - iQ_{xy}(f) \\ &= \frac{d}{df}\, E\{dZ^*_{\underset{\sim}{x}}(f)\,dZ_{\underset{\sim}{y}}(f)\} \qquad \ldots (2.16) \end{aligned}$$

This simple result suggests a direct approach to spectral estimation. Instead of estimating the correlation function, and then implementing the Fourier transform of equation (2.6), we can obtain estimates of $G_{xy}(f)$ from the Fourier transforms of the data records. This approach is the basis of the classic "periodogram" technique, developed at the turn of the century for detecting periodicities in random data. Until recently the digital implementation of this direct method was severely restricted by the very large computation time required to obtain complete Fourier transforms of long records. However, due to the rediscovery in 1965 of a very fast and efficient algorithm for implementing Fourier transforms, known as the Fast Fourier Transform (FFT), the direct approach can now be, in many situations, considerably faster than the conventional correlation method, described in the previous section.

In practice we have, as before, to deal with finite records sampled at equi-spaced intervals, $x(j)$; $y(j)$, $j = 0,1,\ldots,N-1$. The appropriate discrete form of equation (2.14) is then

$$\underset{\sim}{x}(j) = \sum_{n=0}^{N-1} \underset{\sim}{a}_x(n) e^{\frac{i2\pi jn}{N}}$$

$$(j=0,1,\ldots\ldots,N-1) \qquad \ldots (2.17)$$

and the inverse transform is

$$\underset{\sim}{a}_x(n) = \frac{1}{N} \sum_{j=0}^{N-1} \underset{\sim}{x}(j) e^{-\frac{i2\pi jn}{N}}$$

$$(n=0,1,\ldots\ldots,N-1) \qquad \ldots (2.18)$$

(similar formulae apply for $\underset{\sim}{y}(j)$). By an argument similar to that leading to equation (2.16), (see reference 5) an estimator of $G_{xy}(f)$ at frequency

$$f_k = \frac{k}{N} \quad (k=0,1,\ldots,N-1) \qquad \ldots (2.19)$$

is

$$\hat{G}^1_{xy}(k) = N \underset{\sim}{a}_x(k) \underset{\sim}{a}^*_y(k) \qquad \ldots (2.20)$$

In fact, since

$$\underset{\sim}{a}(n) = \underset{\sim}{a}^*(N-n) \qquad \ldots (2.21)$$

and hence

$$\hat{G}_{xy}(k) = \hat{G}_{xy}(N-k) \qquad \ldots (2.22)$$

we need only calculate $\hat{G}_{xy}(k)$ from $k=0$ to $k=N/2$.

It is important to consider the bias and variability of the estimator $\hat{G}_{xy}(k)$ given by equation (2.20). As before, equation (2.7) applies, where here (see reference 5)

$$\alpha(f) \simeq N \frac{\sin^2(\pi f N)}{(\pi f N)^2} \qquad \ldots (2.23)$$

where N is large. This spectral window, therefore, has a $(\frac{\sin x}{x})^2$ shape, as shown in Figure 2.3. From this we find that the statistical band-width is

$$B = \frac{1.5}{N} \text{ (Hz)} \qquad \ldots (2.24)$$

and the number of degrees of freedom is, in this case given by

$$\text{NDF} = 2 \qquad \text{... (2.25)}$$

The variability of the classic periodogram estimator is thus very high and is independent of N. An increase of N produces more periodogram estimators, but does not increase the accuracy of each estimate.

It is a simple matter to smooth the periodogram, by averaging over prescribed frequency intervals. Thus, if we take estimators from $k = p$ to $k = p + q$ we can form a smoothed estimator

$$\hat{G}^2_{xy}\left(p + \frac{q}{2}\right) = \frac{1}{q}\sum_{k=p}^{p+q-1} \hat{G}^1(k) \qquad \text{... (2.26)}$$

This new estimator has a band-width of approximately

$$B = \frac{q}{N}\,(\text{Hz}) \qquad \text{... (2.27)}$$

when q is large (the exact shape of the spectral window resulting from this smoothing operation is discussed in section 2.3), and, for all q, we have

$$\text{NDF} = 2q \qquad \text{... (2.28)}$$

Thus, we can improve the accuracy of the spectral estimates obtained by the direct method if we make both q and N fairly large.

Using the FFT algorithm we can perform the transform given by equation (2.18) when N is some power of 2. In this method the original $x(j)$ array is replaced by the $a_x(n)$ array without the use of extra storage space in the computer for intermediate calculations. Thus, to transform N data points we need only the core space required to store these points. However, when using a small computer, this does imply a practical restriction on N. It is possible to execute the transform for large N on a machine with a small core store by using an ancillary memory, such as a disc, and transferring intermediate stages in the calculation from the core to the disc and back again. An easier and faster solution is to perform the transform on fairly small blocks of data. Suppose

$$N = n_B L \qquad \ldots (2.29)$$

where

$$L = 2^M \qquad \ldots (2.30)$$

and M is an integer. By performing an FFT operation on each block (containing L numbers) we generate n_B estimators

$$\hat{G}^1_{xy}(k) = N \underset{\sim}{a}_x(k) \underset{\sim}{a}^*_y(k)$$

$$(k=0,1,\ldots.,L/2) \qquad \ldots (2.31)$$

for each value of k. If these are simply averaged, we have new estimators

$$\hat{G}^3_{xy}(k) = \sum_{j=1}^{n_B} \hat{G}^1_{xy}(k)$$

$$(k=0,1,\ldots.,L/2) \qquad \ldots (2.32)$$

For these estimators we have

$$B = \frac{1.5}{L} \text{ (Hz)} \qquad \ldots (2.33)$$

and

$$NDF = 2n_B \qquad \ldots (2.34)$$

In practice, it is usually best to combine block averaging with averaging over frequency to obtain estimators

$$\hat{G}^4_{xy}\left(p + \frac{q}{2}\right) = \frac{1}{q} \sum_{k=p}^{p+q-1} \hat{G}^3_{xy}(k) \qquad \ldots (2.35)$$

We then have (for large q)

$$B = \frac{q}{L} \text{ (Hz)} \qquad \ldots (2.36)$$

and

$$NDF = 2\, qn_B \qquad \ldots (2.37)$$

Since the FFT algorithm for computing (2.18) requires $N \log_2 N$ operations, where "operation" means a complex multiplication and addition, the total number of operations required for transforming N numbers, in blocks of size L, is clearly

$$\frac{N}{L} . L \log_2 L = M.N \qquad \ldots (2.38)$$

Thus, if n_E is the number of final estimates required it would seem advisable to choose the smallest value of L for which

$$L > 2 n_E$$

i.e., to use the smallest block size possible. However, due to the introduction of repetitive operations which are necessary to implement the block averaging method, and which absorb a certain amount of computer time in addition to that used by the FFT algorithm, it turns out in practice that the optimum value of L is not necessarily its smallest possible value, for a given n_E. For this reason $\hat{G}^4_{xy}(k)$ estimators are generally more efficient than $\hat{G}^3_{xy}(k)$ estimators.

As in the case of the lagged-producted method, described in the previous section, it is desirable to reduce the "leakage" effect due to the side lobes of the spectral window (see section 2.3). A very simple method is to "Hann" the raw Fourier coefficients; this produces new, smoothed coefficients (see reference 6).

$$a^H_x(n) = -0.25a_x(n-1)+0.5a_x(n)-0.25a_x(n+1) \qquad \ldots (2.39)$$

and similarly for $a^H_y(n)$. One can then produce $\hat{G}^4_{xy}(k)$ estimators, as before. Exactly the same result can be achieved by applying the data window

$$w(j) = \tfrac{1}{2} \left(1 - \cos \frac{2\pi j}{L} \right) \qquad \ldots (2.40)$$

$$(j=0,1,\ldots, L-1)$$

to each data block; i.e., forming the new data values

$$x^H(j) = w(j)x(j) \qquad \ldots (2.41)$$

and similarly for $y(j)$. In either case, the final spectral estimates must be corrected by multiplying by the factor 8/3. The precise effect of this smoothing operation on the shape of the spectral window will be discussed in section 2.3. We note here, however, that this operation has the effect of reducing the statistical accuracy of the final spectral estimates.

When discussing the implementation of FFT it is convenient to consider the complex array

$$\underset{\sim}{z}(j) = \underset{\sim}{x}(j)+i\underset{\sim}{y}(j) \qquad (j=0,1,\ldots\ldots,N-1) \qquad \ldots (2.42)$$

and the complex Fourier transform

$$\underset{\sim}{c}(n) = \frac{1}{N}\sum_{j=0}^{N-1} \underset{\sim}{z}(j)e^{-\frac{i2\pi jn}{N}} \qquad (n=0,1,\ldots\ldots,N-1) \qquad \ldots (2.43)$$

From $\underset{\sim}{c}(n)$ we can deduce $\underset{\sim}{a}_x(n)$ and $\underset{\sim}{a}_y(n)$, using the relationships

$$\underset{\sim}{a}_x(n) = \frac{\underset{\sim}{c}(n)+\underset{\sim}{c}^*(N-n)}{2}$$

$$\underset{\sim}{a}_y(n) = \frac{\underset{\sim}{c}(n)-\underset{\sim}{c}^*(N-n)}{2} \qquad \ldots (2.44)$$

It is clear that the discrete, complex Fourier transform (DCFT) given by equation (2.43) is a kind of nucleus around which we can organise our calculations. A FORTRAN subroutine for computing a DCFT, using the FFT algorithm, is shown in Appendix 2*. This routine, called FAST4, uses floating point arithmetic throughout and calls another FORTRAN subroutine, REV (see Appendix 2), which is used to "unscramble" the order of the Fourier coefficients (see reference 6).

*This subroutine, and other FFT routines mentioned in this section, are due to R.W. Herring (see reference 7).

When performing spectral calculations on a single channel of data, $x(j)$, $j = 0,1,\ldots, N-1$, it is of course possible to use the DCFT program, FAST4, by setting $y(j)$, $j = 0,1,\ldots,N-1$, to zero. However, this is wasteful in time and storage space. A much better approach is to employ intermediate steps in a DCFT of size N/2 to transform the N $x(j)$ data points. A suitable FORTRAN subroutine for this purpose, called REAL4, is shown in Appendix 2. This subroutine calls FAST4 and REV, and performs a floating point transform on N data points from a single channel.

Having described the basic FFT method of spectral estimation, we will now pass on to describe the performance of auto-spectral programs which are founded on this method, and which are currently in use. A suitable measure of the efficiency of these programs is the time, per 1000 data points, necessary to obtain a given number, n_E, of final spectral estimates. We will then discuss the performance of a cross-spectral program currently in use.

The first version of the auto-spectral program (AFF) uses floating point arithmetic throughout and calls the subroutines listed in Appendix 2. The maximum value of M possible is here 10, and the maximum value of n_E is 512 (these values are determined by the size of the computer store). We see from figure 2.4 that $\hat{G}^4_{xx}(k)$ estimates can be obtained at a rate of 10 seconds per 1000 numbers, and that this rate is almost independent of n_E, when the block size factor M = 10. We will discuss the effect of varying M later.

Some increase in efficiency can be achieved by translating the three subroutines given in Appendix 2 into the assembly language DAP. Version 2 of the auto-spectral program (AFD) uses floating point arithmetic throughout and calls DAP versions of FAST4, REAL4 and REV. The maximum values of M and n_E are, again, 10 and 512 respectively. A 20% reduction in computer time is achieved by this means, as shown in figure 2.4.

A further, and far more significant increase in efficiency is achieved by the use of integer (or fixed-point) arithmetic, rather than floating-point arithmetic, wherever possible. Version 3 of the auto-spectral program (AID) uses integer arithmetic completely up to and including the stage where the Hanning operation of equation (2.39) is performed, and calls integer arithmetic, DAP versions of FAST4, REAL4 and REV. These FFT routines scale the Fourier coefficients so that the maximum coefficient lies just within the range ±32,768 (i.e. 16 bits); thus the spectral estimates from the AID program are accurate over about 6 decades of power.

The multiplication of equation (2.31) is achieved by storing the result of an integer multiplication in a double precision store and then transforming to floating point. The final $\hat{G}^{4}_{xx}(k)$ estimates are thus obtained in floating point form. A further advantage of using integer arithmetic is that the core store needed for the FFT transform is halved so that M can be increased to 11 in the AID program. Figure 2.4 shows that the AID program is up to 10 times faster than its equivalent FORTRAN version, AFF, for the same value of M(10).

The times shown in figure 2.4 were obtained with the inputs to the ADC grounded. However, because of the nature of the shifting operations necessary in the DAP, integer version of FAST4, the speed of the AID program is to some extent dependent on the magnitude of the data. Figure 2.5 shows that when a 1 volt R.M.S. sine-wave is applied to the ADC (which results in about 10 shifting operations each time the FFT routine is called) the AID program is slightly slower than in the case where the inputs are grounded (no shifting operations). Since this effect is small, we will, for convenience, in future discussion consider times relating to the case where the inputs are grounded.

The reason for choosing to Hann the Fourier coefficients, rather than apply a data window, in the AID program, is clearly shown by figure 2.6. The very simple operations (shifting and addition) necessary to implement equation (2.39) in integer arithmetic are much faster than the equivalent data window operations (L multiplications of 16 bit numbers). It is interesting to note, however, that the data window approach is faster if one uses floating-point arithmetic throughout, as figure 2.7 shows.

Figure 2.8 shows the effect of block size on the speed of the AID auto-spectral programme. We have already pointed out that the number of FFT operations is a minimum when the block size L is a minimum. However, when the block size is very small a relatively large amount of computer time is absorbed in repetitive operations, such as accessing arrays, which are necessary to implement the FFT routines. This is especially true when n_E is large. As figure 2.8 shows, the program is fastest when $M = 11$, its maximum permissible value. A simple extrapolation of these results indicates that very little advantage, if any, is to be gained by using a larger block size. Indeed, as equation (2.38) suggests, as the block size is increased the speed of the program will eventually fall and the optimum value of M is thus probably 11.

Finally we turn to the performance of the cross-spectral program currently in use (CID). This program uses integer arithmetic as far as possible, as in the case of AID, and produces floating point estimates $\hat{G}^4_{xy}(k)$. The maximum possible values of M and n_E in the standard program is strongly dependent on n_E and that the best value of M to use is 10. An extended version of this program (CIDE) is available, (maximum $n_E = 400$) which was made possible by re-allocating arrays in the core. As figure 2.9 shows, the extended version is somewhat slower than CID, due to the increased time needed for array access.

2.3 Comparison

In this section we will compare the performance of the real time correlation technique described in section 2.1 with that of the direct FFT technique discussed in section 2.2. Speed and accuracy comparisons will be given in detail, but, initially, we will compare the spectral windows which are characteristic of the two, very different, techniques.

2.3.1 Spectral windows

A consideration of spectral windows is inevitable when one attempts to assess the statistical accuracy of spectral estimates obtained from a digital technique. From the appropriate spectral window, $\alpha(f)$, one can readily deduce the statistical band width, B (see reference 1), and hence the bias error of the spectral estimates (see equation (2.13)). Further, if B is large in comparison with 1/N we have the simple result (see reference 1) that

$$NDF = 2BN$$

so that the variability of spectral estimates is also directly related to $\alpha(f)$. A knowledge of the shape of the spectral window is of vital importance if one is concerned with the "leakage" effects which can arise when the spectrum changes rapidly with frequency.

For the periodogram, which is the basis of the FFT technique, the spectral window is given by equation (2.23) (see figure 2.3). This window is replotted in figure 2.10, on a log-linear basis. As equation (2.23) indicates, the lobes of this window fall off only as f^{-2}. Figure 2.11 shows the spectral window which results when the simple "Hanning" operation, given by equation (2.39) is applied to the Fourier coefficients. The side lobes are here clearly much reduced and in fact fall off as f^{-6}, rather than f^{-2}.

The effect of adding together a number (q) of Hanned periodogram estimates to produce one final spectral estimate is to produce a more rectangular window shape. Figure 2.12 shows the spectral window which results when $q = 7$, a typical value. This window is also plotted linearly in figure 2.13 where it is compared with (a) the "ideal", rectangular window, and (b) the window pertaining to estimates obtained from the real time correlation technique.

Finally it is interesting to compare the statistical band widths, B, of the spectral windows. In figure 2.14 we have plotted the ratio

$$\mu = \frac{\text{Statistical Band width}}{\text{Spacing between final estimates}} = 2\,B\,n_E$$

against q, for the spectral estimates derived from the FFT method. For large q we find that μ is approximately unity, in contrast to the value of μ of 2.6 (see equation (2.11)) which applies in the case of the real time correlation method.

2.3.2 Speed

The information given in sections 2.1 and 2.2 enables one to make a direct comparison of the speed of the two methods in any given real time analysis. We will consider the case of auto-spectral estimation initially.

In figure 2.15 we have compared the speed of the correlation progıam (CORD1) with that of the FFT integer auto-spectral program (AID). Contours of total time (i.e., computation time + sampling time) per 1000 numbers have been plotted on an $r_s:n_E$ plane. The operating range of the correlation method lies to the left of the broken boundary line in this figure. In any application one can estimate the total time required by locating the "operating point" in figure 2.15, and then interpolating the contours of total time. For example, if 50 estimates are required and one wishes to sample 50,000 numbers at the rate of 200 samples/sec., then we find from figure 2.15 that CORD1 will take a total time of 250 seconds whereas AID will take about 300 seconds.

It will be noted from figure 2.13 that, in the region of the operating plane where CORD1 can be used, it is always faster than AID. The difference in speed is very small, however, except in the area where n_E is small and r_s is high. As mentioned in section 2.1, the operating range of the correlation program could be extended by sampling before any lagged product computation is undertaken. The appropriate time contours for this block sampling mode of operation are shown as dotted lines in figure 2.15. Clearly, in the block sampling mode, the correlation technique is considerably slower than the FFT technique when n_E and r_s are large.

Figure 2.16 shows a similar comparison for auto-spectral estimation, but in this case we have compared the all FORTRAN version of the correlation program (CORF) with the all FORTRAN FFT auto-spectral program (AFF). Here the operating range of the correlation method is very restricted and it is only appreciably faster than the FFT method when n_E is very small.

In figure 2.17 the speed of the correlation program in mode 3 (CORD3) has been compared with that of the FFT integer cross-spectral program (CID). Here again the correlation method offers some noticeable saving in overall computer time in the case where n_E is small and r_s is fairly high.

It is perhaps worth pointing out that if one records the data on a tape-recorder with variable record and replay speeds, prior to analysis, then one has a choice of operating points in figures 2.15 to 2.17. In particular, if data is sampled at a low rate, then by using a high ratio of replay to record speed one can take advantage of the capabilities of the FFT programs and substantially reduce the overall computer time necessary for the analysis of a given number of data points.

2.3.3 Accuracy

To fully test the accuracy of the programs we have described it has been found very convenient to use versions which accept a paper tape input, rather than an input from the ADC. In this way one can test the various programs which have been described on one prescribed and repeatable set of data points.

By operating on normally-distributed random numbers it is possible to simulate, exactly, equi-spaced samples of the response of a simple linear oscillator to ideal white noise excitation (see reference 8). 40,000 such samples have been generated for the case where the damping factor of the oscillator is 0.1 and the ratio

$$\frac{\text{aliasing frequency}}{\text{natural frequency of the oscillator}} = 25$$

The samples, which have been stored on paper tape in 8 bit binary code, have the theoretical spectrum shown in figure 2.18. Also shown in this figure is the level of the 8 bit quantisation noise appropriate for these samples together with the results of analysing the samples using the correlation program, CORD1, with $n_E = 200$. The agreement between the theoretical spectrum and the spectral estimates is excellent, the estimates only departing from the theoretical curve when the quantisation level is reached. In this case we have (see equation (2.10)) NDF = 400.

Figure 2.19 shows the result of analysing the test tape, using the FFT integer program, AID. Here a block size, L, of 2048 was chosen and $n_E = 512$. Again an excellent agreement between the final estimates and the theoretical curve is obtained. Here we chose $n_B = 19$ and thus, from equation (2.37), NDF = 76.

To check the accuracy of the integer arithmetic in AID, the test tape was analysed by both AID and the floating point version, AFD, using identical values of L = 1024 and $n_E = 200$. This gives NDF = 152. Figure 2.20 shows the result of analysing the test tape with the AID program; once again a good agreement with theory is obtained. The results of the AFD analysis shown in figure 2.21, prove that the use of integer arithmetic produces negligible errors over at least 5 decades of power.

Finally, it is interesting to examine the role of the Hanning operation in the FFT method. Figure 2.22 shows the spectral estimates obtained from the AID program, modified in such a way that the Hanning operation is eliminated. Here L = 2048, $n_E = 512$ and here NDF = 76, as in figure 2.20. The errors due to leakage are clear, the spectral estimates levelling off well above the quantisation level for 8 bits, at high frequencies.

Similar accuracy comparisons have been carried out to check the cross-spectral programs. None are presented here since they do not add any significantly different information.

3. HARDWARE

Figure 3.1 shows a block diagram of the overall hardware system described below. It consists primarily of the line system,[†] the ADC and the computer and its peripherals. These subsystems are described below and the performance of the system is then discussed. More detailed descriptions of the computer system and its use as a data processor are given in reference 9.

[†] Although two lines only are shown, in fact, several lines are available which run to various wind tunnel facilities.

3.1 Relevant computer specifications

The Honeywell DDP-516 computer used in this work is equipped with a core store of 2^{14} (approximately 16K) words. The word size is 16 bits which corresponds to a single precision integer. Two words are required for a floating point number. The machine has the optional fast hardware for high speed integer arithmetic (HSA). Typical machine times are its memory cycle time of 0.96μsecs and its addition time for two integers of 1.92μsecs.

3.2 Relevant analogue input system specifications

The Micro Consultants Modular 15 ADC unit provides simultaneous sample and hold capability for up to 16 channels. The maximum resolution of a sample on this particular instrument is 15 bits (i.e. integers in the range $\pm 2^{14}$). The data is transferred to the computer boosted by a factor of two as a number in the range ±2 volts. Thus the bit size corresponds to approximately 122μvolts. The sampling rate on this instrument is to a large extent a function of the manner in which it is initially set up. Furthermore, it depends on whether the ADC operates under so-called "hardware" or "software" control.* The latter, requiring the use of an external clock, is the mode used by the present set of programs. Also optional on the ADC is the nominal settling time per bit which can be taken as one or two microseconds. The one microsecond setting is usually used as its accuracy is consistent with the noise levels introduced by the sampling and hold amplifiers. The maximum scanning rate,† then, is a function of the number of channels sampled in each scan, and the resolution selected. An external "startscan" clock is used to select rates less than the maximum. The minimum conversion time per channel varies from 20μsec for a 15 bit conversion to 13μsec for an 8 bit conversion; in addition, there is a settling time of about 14μsec for the parallel sample and hold amplifiers at the start of each scan. Maximum allowable sampling rates, as functions of

---

*The added restriction to the sampling rate under software control can be avoided by different logic in the ADC.

†The sampling rates can be described reasonably accurately by a simple model of the sampling process given as: $T = D+S_s+NnS_b$ where T = time per scan of N channels at n bits, D is the software delay (= 0 for hardware control), S_s is the settling time per scan, and S_b is the settling time per bit (nominally 1 or 2μsecs). For the instrument used here, these parameters are currently given as: D = 40.4μsecs, S_s = 14.0μsecs, S_b (1μsec) = 1.09μsecs, S_b (2μsec) = 2.76μsecs. Note that the maximum allowable sampling rate = 1/T.

resolution, were measured directly under various conditions for single and dual channel operation and are shown in figures 3.2(a) and (b). Figure 3.2(a) shows the sampling rates allowable when under software control. These are the relevant ones for the current programs. Figure 3.2(b) is added for completeness, and shows similar curves for hardware control.

3.3 Description and performance of the lines

The system currently in use was developed to fulfil an immediate requirement, and as such does not represent an optimized or final solution. Nevertheless the performance obtained is remarkably good. A typical data channel has been illustrated in block diagram form in figure 3.1. The experimental data is amplified to ±10 volt range and is applied to the lines through simple line drivers. These line drivers are convenient amplifiers which can provide sufficient current to drive the primarily capacitive load. The line systems are of considerable interest in themselves. Hence a more detailed description of their physical set-up and the model of them used to minimize the system resonance is included as Appendix III. For the present purpose the lines can be considered simply as twin conductor shielded cables. At the end of each cable the two conductors are connected to the two inputs of a unity gain differential amplifier which acts as a line receiver. At this point the data is attenuated and filtered to provide an ADC input which is within the specified ±2 volt range and which contains negligible aliasing power. In practice the filters must be at least fourth order, and are set for convenience with their natural frequencies at the Nyquist or folding frequency. (i.e. half of the sampling rate). This gives acceptable spectral estimates with negligible aliasing up to 60% of the folding frequency for turbulence type spectra. Increasing the filter order to eight increases the acceptability limit to nearly 80% of the folding frequency. The filters currently in use are Krohn-Hite model 3100 fourth-order band-pass filters modified to give a flat response down to one Hz.[†] This modification also includes the incorporation of an output gain of nominally ½ in order to allow the filter to operate at its design input level of about 4 volts. Matching of the two channels is accomplished by applying the same random input to both channels and using the integer cross-spectral program described previously to measure the

[†] The low frequency response characteristics of the filters are shown as an example in Section 4.4.

phase difference between the two channels over the range required. By manually changing the natural frequency setting of one of the filters, a minimum phase difference condition can rapidly be obtained. The accuracy obtainable is typically better than ±0.5 degrees over the range 1 Hz to 60% of the Nyquist frequency. This includes the complete system from line drivers through to the ADC. This performance is shown as an example in section 4.3. The initial line system (consisting of the heavy duty amplifier driving into a line with a differential amplifier receiver) displayed a frequency response with a substantial peak at approximately 40 kHz. This is caused by the capacitive loading (approximately 0.03 μ farads) on the line driver amplifiers (see Appendix III for details). By the use of a variable series resistor (VR in figure III.1) a best compromise of flat response to as high a frequency as possible was obtained. The resulting amplitude response obtained is shown as curves A and B of figure 3.3. As can be seen, the two line systems still show a marked resonance; however, it was found that further reduction of these peaks led to a drop in the response at 10 kHz. Hence the performance shown was accepted. Over the range 0 to 10 kHz, the amplitude response is flat to within ±0.5%. There is little doubt that improved components could extend the flat response to much higher frequencies if required.

The system noise level has been assessed for a variety of configurations to determine the characteristics of the noise in comparison with noise levels of expected data sources, the ADC, and that associated with the quantization/spectral analysis process itself. In terms of overall rms noise levels, the following levels as seen by the ADC are typical.

1) typical signal level for ±2 V ADC range = 0.5 V rms
2) ADC quantization noise (see later) = 3.52×10^{-5} V rms
3) ADC noise when grounded at the input to the sample and hold amplifiers = 2.5×10^{-4} V rms
4) filter plus ADC noise (filter input shorted, range set to be 3 db down at 2.5 kHz) = 3.0×10^{-4} V rms
5) total system line noise with line driver inputs grounded as seen through 1 Hz to 10 kHz filter range (1 Hz to 2.5 kHz is essentially the same) = 6.0×10^{-4} V rms
6) total wide band noise (line drivers grounded with no attenuators or filters) = 1.5×10^{-3} V rms

These quoted rms noise values are only repeatable to within about ±50% due to non-stationary effects such as particular interference sources being on or off. The values given above are conservative. It can be seen that for the frequency range of interest here, it appears that the signal to noise ratio is of the order of 10^3 in overall amplitude and 10^6 in overall power.

Further to the above, it is of particular interest to examine the spectral distribution of the noise to determine more precisely whether noise problems exist over any part of the analysis range. The configurations in the list below are of interest and are shown plotted in figures 3.4 and 3.5. For convenience they are all considered for data sampled at 5 kHz; in the total analysis range (up to 2.5 kHz) the spectral estimates can be expected to suffer negligible gain or alaiasing problems up to 1.5 kHz.

1) Two comparative input spectra of standard deviation $\sigma = 0.5$ V rms following the form $G(f) = \frac{2\sigma^2}{\pi f_n} \left(\frac{1}{1+(f/f_n)^2}\right)$, where the 3db down points at $f = f_n$ have been taken at 1 Hz (1A) and 10 Hz (1B).

2A) ADC quantization noise level arising from the 122 microvolt resolution of the quantizer. This noise level is given in reference 1 as having a mean square of $(\text{bit size})^2/12 = 1.24 \times 10^{-9}$. Since it can also be assumed to be effectively white noise, its spectral level is given as $1.24 \times 10^{-9}/2500 = 4.96 \times 10^{-13}$ volts^2/Hz.

2B) Quantization noise associated with the correlation program which operates on 8 bits (a resolution of 15.6 mv.). By a similar argument to that of 2A, this level is given as 8.15×10^{-9} volts^2/Hz.

3) ADC sample and hold amplifier noise level. (By nature it is unfiltered and hence the complete mean square appears distributed over the range of interest).

4) Complete system (as per figure 3.2) noise spectrum with line drivers grounded. This corresponds to the case of data naturally originating in the ±10 V range. A best case (4A) and a worst case (4B) so far recorded are both shown.

5) Noise spectrum with no attenuators in use other than the 0.5 gain of the filters. (Line drivers' inputs grounded). This corresponds to data originating and being transmitted down the lines in the range ±4 V.

6) Complete system noise spectrum with line driver inputs driven by an amplifier with a gain of nominally 2.5. The amplifier input is grounded. This corresponds to data originating in the ±4 V range which is then boosted for line transmission.

7) A typical tape recorder noise spectrum obtained by recording and replaying ground at 15 ips on an Ampex FR 1300 tape recorder. The output is filtered but no part of the line system is included. This implies ideal amplification, transmission, and attenuation, since the tape recorder output is naturally in the ±2 volt range. (The filter used here was an eighth order low pass with unity gain).

Again it should be noted that the data corresponding to runs 4, 5 and 6 above tends to be very variable in terms of long period stability(e.g. different days), although runs are reasonably repeatable if performed consecutively. The above set (4A, 5 and 6) are such consecutive data and hence should be consistent; however, a variability of the order of 100%, as shown for 4, might be expected in the spectral estimates on different days.

In Figures 3.4 and 3.5 the above examples are referred to by number. The following observations can be made. Curve 4 represents typical system noise. Curves 1 are spectra covering a rather large number of decades in the experimental range. It should be noted that frequently spectra may be obtained which drop off more sharply than those shown - for example see section 4.2. In such cases, more decades of power may be obtained before running into noise problems. It can be seen that the highest noise levels are at low frequency where experimental spectra are also typically high. At all frequencies, the noise level is not appreciably different to the ADC sample and hold noise. These results indicate that the system allows a spectral signal to noise range of the order of 8 decades which is about all that can be obtained using the integer arithmetic programs. Curves 2 and 3A indicate that under the present operating conditions using the sample and hold capability, the accuracy of the digitization could be reduced by two bits (i.e. reduce the resolution to 488 microvolts) without significantly altering the base noise level.

Curve 3B shows the quantization noise obtained when using the correlation approach, or when using 8 bit binary paper tape as a data input. This level is more consistent with tape recorder noise levels (curve 7). Accuracy would be restricted at high frequency if a spectrum of the type shown, covering more than six decades, were analysed.

The noise spectrum from the tape recorder is typical of many instrumentation noise levels and can be seen to be more restrictive than line noise. It is also noteworthy that the strong spectral peaks in evidence reduce the available signal to noise ratio considerably more than the overall noise level would indicate. However, the seven decades of "clean" power available will usually be more than adequate for most experiments. Lower overall tape noise can be obtained by using faster tape speeds. In this case 60 ips gives the lowest signal to noise ratio. Similar spectra for this case however, show that it is primarily low frequencies at which the noise level is most significant, there is negligible advantage in using the higher tape speed. Obviously, if one's basic instrumentation noise is significantly less than the tape recorder's, then direct on-line analysis should be employed.

Curves 5 and 6 in figure 3.5 illustrate that there is a small advantage to be gained in boosting the signal for transmission. If the pre-amplification and attenuation were ideal for case 6, then all pickup induced in the lines should be reduced by a factor of 2.5^2. The small differences seen show that the addition of these real components themselves add some noise. The high frequency bump in the attenuated case is not fully understood, but may be due to the amplifier used. Since in most cases, amplifiers are in permanent use for a variety of reasons, it is felt that the best way to minimize external effects is to make use of ±10 volt line signals. Furthermore, this has been a some what conservative view due to the additional attenuation already incorporated in the filters.

This section has shown that the present system is capable of providing a high standard of accuracy. However, it should be noted that there are at least two areas which could be significantly improved both in standard and convenience. The first is provision of wider range, low noise line drivers. The second is replacement of the present analogue filters by programmable digital filters. This would assure absolute matching of the two input channels, with less set-up time.

4. SOME TYPICAL RESULTS

The spectral analysis system as developed at NPL is primarily intended for the analysis of turbulence and structural response. The following examples illustrate some typical useage of the programs to date. They are either related to testing parts of the system - the lines and filters - or arise in turbulent flow fields which are currently being examined.

Unless otherwise indicated, the programs used are the integer FFT versions with the maximum value of L allowable.

4.1 Line Distortion

To indicate the dynamic range of the system and to determine whether any significant nonlinearities exist for the transmission line, an analogue sine wave originating from a Hewlett Packard precision oscillator (a model 745A a.c. calibrator) was examined. The auto spectra of a 500 Hz signal was obtained for three configurations: the sine wave was applied a) directly to the ADC to provide a reference, b) to the system without the filters to check for line distortion alone, and c) to the complete line system. In all cases, amplitudes were taken such as to provide the ADC with about one volt rms. The results are shown in Figures 4.1a, b and c. Case a provides the lowest base noise level, and shows the oscillator does have some slight distortion showing up at 1500 Hz. The 50 Hz bump is presumably pickup in the connections used, since it does not appear in all the cases. From figure 4.1b we see that without the filters, the lines themselves have negligible distortion. The noise level is increased, but it should be noted that since no filters were used in either a or b, the total wide band noise appears distributed over the analysis range. Note also that the ADC noise is considerably below the levels shown. Its level is at about 10^{-10}. Figure 4.1c shows that introduction of the filters causes measurable distortion although even here the largest harmonic is more than five decades down from the peak (about 55 dB). In all cases, nearly eight decades of power have been resolved. Note that the filter side lobes can be clearly seen.

4.2 Some typical turbulence spectra

In a current experiment, there is particular interest in that part of the turbulence spectrum associated with the dissipation process at high wave number ($k = 2\pi f/U$, where k = wave number and U = mean speed). For these experiments a low mean speed is used, and the resulting hot wire signals descend into the equipment noise level at relatively low frequency. Since it is that part of the spectrum just before the signals enter the noise level that is important, it was of interest to examine the feasibility of intermediate tape recording (without any further complications such as prewhitening the data). Figure 4.2 shows a comparison of two analyses - one made "on-line", the second made using an Ampex FR1300 tape recorder for intermediate storage (recording at 15 ips). Clearly, the FR1300 tape recorder only results in a small loss of information at the high frequency end in this very restrictive experiment. Since from section 3, we know that the submerged noise levels are relatively high over the last decade or so of power, we might expect a more severe test to be that of dual channel coherence. This is illustrated in section 4.5. Also shown in figure 4.2 is the spectrum obtained for a case similar in all respects but with no turbulence producing grid. This indicates some background wind tunnel turbulence and, at high frequency, the DISA anemometer model A01 noise level. Finally, the system noise level was found with the DISA anemometer replaced with a short circuit and the appropriate scaling factors incorporated. Its value was approximately 10^{-13}. Clearly, the system's noise level is compatible with the hot wire equipment's noise level for this experiment.

4.3 Phase matching of filter characteristics

As mentioned in section 3.3, for cross-spectral analysis both channels must be filtered to prevent aliasing, and for convenience of data handling, this filtering must not introduce any significant phase errors into the processes. This is because phase errors lead to distortion of the relative magnitudes of the real and imaginary cross spectra, and hence are not so easily handled as gains. In order to phase match the two filters, the same wide band noise signal is applied to both lines with the filters nominally set at the required cut-off. Then, cross-spectra over the required range are formed, for about 10,000 samples. The results give the phase difference as a function of frequency. If there is a significant phase difference in evidence ($>1^\circ$) over the first 70% of the folding frequency, then the cut-off frequency of one of the filters is altered slightly and the test repeated until the required matching is obtained. Two typical results are shown in figure 4.3. They represent the cases of 250 and 2500 Hz nominal cut-off frequencies. In both cases, the order of accuracy of the matching is better than $\pm 0.5^\circ$ over the required range.

4.4 Filter characteristics

It was of interest to derive the absolute gain and phase characteristics of the two filters in use over the low frequency part of the range. For this purpose, wide band noise was first filtered by a Krohn-Hite model 3340 8th order filter, tuned to 100 Hz low pass cut-off. The resulting signal was then passed through one of the system filters. Cross-spectra of the input and output of the system filter yielded the gain and phase characteristic of that filter over approximately 0.1 to 70 Hz. The results, for both system filters, are shown in figure 4.4. It can be seen that the gain has dropped by about 2% at 1 Hz and at this point the phase is matched to better than 0.5°. The gain and phase characteristics are obtained without problems of variability simply because the same Fourier components are being sampled at the output as were applied to the input, and hence their ratio is independent of their individual variability (neglecting finite record effects.) Naturally, in all these results, the measured coherence (indicating the linearity of the filters) was essentially 1.00.

4.5 Coherence measurements

Returning to the turbulence example of section 4.2, the experiment aims to provide coherence estimates for two laterally displaced hot wire sensors in turbulent flow. Here, coherence is defined as.

$$\gamma_{xy}^{2}(f) = \frac{|G_{xy}(f)|^{2}}{G_{xx}(f)G_{yy}(f)}$$

The quantity $\gamma_{xy}(f)$ can in this case be interpreted as a narrow band cross-correlation function. It should be noted that if x and y are two data channels with superimposed noise, then the coherence may be increased if the noise is correlated, or it may be decreased if the noise is uncorrelated and simply contributes to the power spectra. The effect of tape recording the signals is shown in figure 4.5. Since the signals enter the hot wire anemometer noise level at about 1000 Hz it can be seen that the tape recorder does lead to satisfactory answers over most of the important range. When the tape recorder results do diverge appreciably from the on-line values, it is towards higher values, indicating that the tape noise is correlated.

REFERENCES

No.	Author(s)	Title, etc.
1	J. S. Bendat, and A. G. Piersol	Measurement and Analysis of Random Data. J. Wiley and Sons, New York, 1966.
2	R. B. Blackman, and J. W. Tukey	The Measurement of Power Spectra. Dover Pub. Inc., New York, 1959.
3	L. D. Reid.	The Measurement of Human Pilot Dynamics in a Pursuit-Plus-Disturbance Tracking Task. University of Toronto, Institute of Aerospace Studies, Report No.138. April, 1969.
4	A. Papoulis.	Probability, Random Variables and Stochastic Processes, McGraw-Hill, New York 1965.
5	J. W. Cooley, P. A. W. Lewis, and P. D. Welch	The Application of the Fast Fourier Transform Algorithm to the Estimation of Spectra and Cross-Spectra. J. Sound and Vibration, Vol.12, No.3, 1970, pp. 339-352.
6	C. Bingham, M. D. Godfrey, and J. W. Tukey	Modern Techniques of Power Spectrum Analysis. IEEE Trans. Audio Electroacoust. AU-15, p.56, 1967.
7	R. W. Herring	A Fast Fourier Subroutine Package, National Radio Propagation Laboratory, Communications Research Centre, Ottawa, X16-0342-007.
8	J. B. Roberts	Estimation of the Probability of First-Passage Failure for a Linear Oscillator. J. Sound and Vibration, vol.10, No.1, 1969, pp. 42-61.
9	R. J. North, R. F. Johnson, and J. Bridgewater	Experience in the Introduction and use of a Small Computer in an Aerodynamic Research Laboratory. NPL Aero Note 1096. A.R.C.32 398. September, 1970.

ACKNOWLEDGEMENTS

The authors would like to express their thanks to a number of people who have directly contributed to the work presented in this report. Mr. Roy Johnson was responsible for the design and installation of the line system hardware, and the high performance which was achieved was to a large extent due to his many helpful suggestions. Mr. Richard Young was responsible for writing the DAP version of CORD and numerous small DAP subroutines used in the integer method. Mr. Brian Smith and Mr. Peter Ponsford both provided a great deal of assistance in developing and "debugging" the line system hardware. Without the willing assistance of the above, the project would have been considerably more difficult.

Appendix 1. Fortran program for real time correlation technique.

```
C       FORTRAN PROGRAM FOR DEVELOPING AUTO OR ONE SIDED CROSS,RXY
C       ALL FLOATING POINT ARITHMETIC
        DIMENSION IDATA(2),X(200),Y(200),RXY(200),RYX(1),RXX(1),RYY(1)
111     CONTINUE
        WRITE(1,101)
101     FORMAT(3H M=)
        READ(1,102)M
102     FORMAT(I4)
        WRITE(1,103)
103     FORMAT(7H NLAGS=)
        READ(1,102)NLAGS
C       TOTAL NO OF SAMPLED POINTS PER CHANNEL=M*NLAGS
        WRITE(1,104)
104     FORMAT(6H MODE=)
        READ(1,107)ISW
C       THIS VERSION ONLY PERFORMS MODE 1:RYX,RXX,RYY ARE DUMMY
        WRITE(1,105)
105     FORMAT(7H CLOCK=)
        READ(1,102)ICLOCK
110     RYX(1)=0.
        RXX(1)=0.
        RYY(1)=0.
        XTOT=0.
        YTOT=0.
        XVAR=0.
        YVAR=0.
C       THIS LOOP FILLS ARRAY BEFORE COMPUTING OF RXY BEGINS
        DO 1 I=1,NLAGS
        RXY(I)=0.
        CALL INPUT(IDATA,2,0,1,1)
C       INPUT SAMPLES CHANNELS 0 AND 1 ON NEXT EXTERNAL PULSE COMMAND
        X(I)=IDATA(1)
1       Y(I)=IDATA(2)
        DO 2 J=1,M
        DO 2 I=1,NLAGS
        CALL INPUT(IDATA,2,0,1,1)
        X(I)=IDATA(1)
        Y(I)=IDATA(2)
        XTOT=XTOT+X(I)
        YTOT=YTOT+Y(I)
        XVAR=XVAR+X(I)**2
        YVAR=YVAR+Y(I)**2
        DO 3 K=1,I
        II=I-K+1
3       RXY(K)=RXY(K)+X(II)*Y(I)
        KK=I+1
        DO 2 K=KK,NLAGS
        JJ=NLAGS+I+1-K
2       RXY(K)=RXY(K)+X(JJ)*Y(I)
        CALL XCALC(M,NLAGS,ISW,ICLOCK,XTOT,YTOT,XVAR,YVAR,RXY,RYX,RXX,RYY)
C       XCALC CORRECTS FOR MEAN,INTRODUCES CALIBRATIONS,AND TRANSFORMS
        WRITE(1,106)
106     FORMAT(28H REPEAT(0),CHANGE(1),END(2)=)
        READ(1,107)ID
107     FORMAT(I1)
        IF(ID-1)110,111,112
112     CALL EXIT
        END
```

APPENDIX II

FORTRAN subroutines for direct FFT technique

The FORTRAN versions of the FFT algorithm, listed in this appendix, consist of three separate subroutines. The first of these, named FAST4, executes a complex FFT on the (assumed) complex input data, and replaces them with their complex spectrum in the same area of memory. The second subroutine, named REV, sets the input data in the appropriate order to processing by FAST4. The third subroutine, named REAL4, executes a complex FFT on real input data by manipulating intermediate results from FAST4, and returnsthe transformed data in the same area of memory that held the input data.

FFT on complex data

To perform a complex FFT on complex data the calling statement is

```
CALL FAST4 (M,X,W,ISIGN)
```

where:

(a) M contains$\log_2$L in integer format where L is the number of complex data points.

(b) X is the first location of the array containing the complex floating-point data. The data are assumed to be stored with their real and imaginary parts in successive array locations, the usual manner in which FORTRAN handles complex data. Hence the array X must be sufficiently large to contain 2L real values. The transform is executed"in place", i.e., the results, the complex coefficients, are returned unscaled in natural order in the array X, with real and imaginary values in successive locations. (It should be noted that a call to REV is contained in FAST4).

(c) W is the first location of an array sufficiently large to contain at least L/4 real floating-point values. This array is used by FAST4 to store the first quarter of a cosine wave of period L.

(d) ISIGN determines the effect of a call to FAST4. To initialise FAST4, ISIGN is set to zero. The cosine table W is then set according to the value of L for the call but the FFT is not executed. Subsequent calls to FAST4 use a value for ISIGN of +1, or -1, the sign of the contents of ISIGN determining the sign of the argument of the complex exponential in the transform.

FFT on real data

To perform a complex FFT on real data the calling statement is

```
CALL REAL (M,X,W,ISIGN)
```

A call to FAST4, and hence an implicit call to REV, is contained in REAL4.

Here:

(a) M contains $\log_2 L$ with L is the number of real data points.

(b) X is the first location of the array containing the real floating-point data. This array must be large enough to contain at least L data points. The results of the transform, the cosine and sine coefficients are returned unscaled in the array X.

(c) W is the first location of an array sufficiently large to contain at least L/4 real floating-point values.

(d) ISIGN has the same function as before.

(a) Subroutine Fast 4.

```
      SUBROUTINE FAST4(M,X,W,SGN)
      DIMENSION X(1),W(1)
      INTEGER SGN
      DATA ML/0/
      IF (M .GE. 14) STOP
      IF (IABS(SGN)-1) 1,3,41
    1 IF (M .EQ. ML) RETURN
      NTWID4 = 2**(M-2)
      NTWID2 = NTWID4+NTWID4
      PHASE = 3.1415926536/FLOAT(NTWID2)
      DO 2 I = 1,NTWID4
    2 W(I) = COS(FLOAT(I)*PHASE)
      ML = M
      IF (SGN .EQ. 0) RETURN
    3 IF (M .GT. ML) GO TO 1
      N2 = 2**(M-1)
      N = N2+N2
      NT2 = N+N
      I = 1
      IF (SGN .GT. 0) GO TO 33
    4 CALL REV(M,X)
      IRAT = NTWID2
      IDISP = 2
    5 IREPL = IDISP+IDISP
      JLIM = IDISP-2
      J= 0
      DO 10 IJ = 2,NT2,IREPL
      IJD = IJ+IDISP
      T1 = X(IJ-1)+X(IJD-1)
      T2 = X(IJ)+X(IJD)
      X(IJD-1) = X(IJ-1)-X(IJD-1)
      X(IJD) = X(IJ)-X(IJD)
      X(IJ-1) = T1
      X(IJ) = T2
   10 CONTINUE
   11 IF (J .EQ. JLIM) GO TO 31
      J=J+2
      IND = J*IRAT/2
      IF (IND .NE. NTWID4) GO TO 21
      DO 20 I= 2,NT2,IREPL
      IJ = I+J
      IJD = IJ+IDISP
      T1 = X(IJ-1)+X(IJD)
      T2 = X(IJ)-X(IJD-1)
      T3 = X(IJ-1)-X(IJD)
      X(IJD) = X(IJ)+X(IJD-1)
      X(IJD-1) = T3
      X(IJ-1) = T1
      X(IJ) = T2
   20 CONTINUE
      GO TO 11
```

Subroutine Fast 4 cont'd.

```
21 IF (IND .LT. NTWID4) GO TO 22
   IND = IND-NTWID4
   WI = W(IND)
   IND = NTWID4-IND
   WR = -W(IND)
   GO TO 23
22 WR = W(IND)
   IND = NTWID4-IND
   WI = W(IND)
23 DO 30 I = 2,NT2,IREPL
   IJ = I+J
   IJD = IJ+IDISP
   T3 = X(IJD-1)*WR+X(IJD)*WI
   T4 = X(IJD)*WR-X(IJD-1)*WI
   T1 = X(IJ-1)+T3
   T2 = X(IJ)+T4
   X(IJD-1) = X(IJ-1)-T3
   X(IJD) = X(IJ)-T4
   X(IJ-1) = T1
   X(IJ) = T2
30 CONTINUE
   GO TO 11
31 IF (IDISP.EQ. N) GO TO 32
   IRAT = IRAT/2
   IDISP = IDISP+IDISP
   GO TO 5
32 IF (SGN .LT. 0) RETURN
   I = 2
33 DO 40 J = 2,NT2,2
40 X(J) = -X(J)
   GO TO (4,42),I
41 SGN = ML
42 RETURN
   END
```

(b) Subroutine Rev

```
      SUBROUTINE REV(M,X)
      INTEGER C
      DIMENSION X(1),C(15)
      DATA C(1)/16384/,C(2)/8192/,C(3)/4096/,C(4)/2048/,C(5)/1024/,
     .C(6)/512/,C(7)/256/,C(8)/128/,C(9)/64/,C(10)/32/,C(11)/16/,
     .C(12)/8/,C(13)/4/,C(14)/2/,C(15)/1/
      L=15-M
      LIM=C(L-1)-4
      JJ=0
      DO 10 J=2,LIM,2
      K=L
   11 IF(JJ.LT.C(K)) GO TO 12
      JJ=JJ-C(K)
      K=K+1
      GO TO 11
   12 JJ=JJ+C(K)
      IF(JJ.LE.J) GO TO 10
      T1=X(J+1)
      T2=X(J+2)
      X(J+1)=X(JJ+1)
      X(J+2)=X(JJ+2)
      X(JJ+1)=T1
      X(JJ+2)=T2
   10 CONTINUE
      RETURN
      END
```

(c) Subroutine Real 4

```
      SUBROUTINE REAL4(M,X,W,SGN)
      DIMENSION X(1),W(1)
      INTEGER SGN
      ML=2
      CALL FAST4(M,DUM,DUM,ML)
      IF (IABS(SGN)-1) 23,1,21
    1 I=1
      IF (ML-M) 24,2,2
    2 N4=2**(M-2)
      JLIM=N4-1
      N=4*N4
      NP2=N+2
      NTWID4=2**(ML-2)
      IRAT=NTWID4/N4
      CALL FAST4(M-1,X,W,-1)
      T=X(1)+X(2)
      X(2)=X(1)-X(2)
      X(1)=T
      DO 10 J=1,JLIM
      K=J*IRAT
      WR=W(K)
      K=NTWID4-K
      WI=W(K)
      K=J+J
      KP=NP2-K
      AKRL=X(K+1)+X(KP-1)
      AKIM=X(K+2)-X(KP)
      BKRL=X(K+2)+X(KP)
      BKIM=-X(K+1)+X(KP-1)
      T1=WR*BKRL+WI*BKIM
      T2=WR*BKIM-WI*BKRL
      X(K+1)=AKRL+T1
      X(K+2)=AKIM+T2
      X(KP-1)=AKRL-T1
      X(KP)=-AKIM+T2
   10 CONTINUE
      X(K+3)=X(K+3)+X(K+3)
      X(K+4)=-X(K+4)-X(K+4)
      IF(SGN.LT.0) RETURN
      DO 20 J=4,N,2
   20 X(J)=-X(J)
      RETURN
   21 SGN=ML
   22 RETURN
   23 I=2
   24 CALL FAST4(M,DUM,W,0)
      ML=M
      GO TO (2,22),I
      END
```

APPENDIX III

Further Details on the Line System

by R.F. Johnson

The history of the development of the lines is recorded below in order to simplify any future projects of this type. Also included below is an analysis of the response of operational amplifiers subjected to capacitive loading of the type presented by such long lines.

III.1 Further physical details of the lines

A number of different cables were investigated for use as analogue transmission lines. It was soon apparent that a twin twisted cable with an overall shield was best for the prevention of noise pickup. Two forms of shielding are available, a woven copper braid and a conducting plastic sheath. Tests on specimen lengths of these (DEF 2C with the braid and type EHE 2PM from Souriau Lectropon with the graphite-impregnated plastic sheath) showed that the metallic braided cable was superior to pickup at frequencies above 50 KHz but that over the range 0 - 10 KHz the conducting plastic sheathed cable was superior. The latter also had the advantage that voltages introduced by cable movements (microphoney) were very much lower, although this effect would not be serious in practice because of the high line capacitance and the low source impedance of the line driving amplifier.

kHz

Because of the underground installation of cables would have taken considerable time and the overhead crossing of a raodway was not permitted, use had to be made of some existing coaxial cables. A typical line system is shown in figure III.1; where the first 100m and the last 150m of the line are screened pairs but the central portion is made up from two 75 ohm coaxial cables. This central section is load in underground ducting in close proximity to power and control cables and it is probable that most of the low frequency noise originates in this section. The twin shielded cables are at an average height of 5m and well away from power cables.

Considerable radio-frequency pickup occurs on the lines due to local broadcasting stations. The dominant frequencies are 200 KHz and 960 KHz, which are greatly attenuated by the line receiver and lower pass filter. Care must be taken, however, that these high frequencies do not cause anomalous operation of the line receiver, for example by the amplifier acting as a detector. To obviate possible trouble, the line receiver has r.f. filters in the input lines and a good high frequency performance. A 60dB common mode rejection ratio is achieved at 200 KHz, from 0 to 100 KHz the ratio is >100 dB.

III.2 Response of operational amplifiers with capacitative loading

The idealised open-loop amplitude and phase response for the line driver amplifier are shown in figure III.2 by the full lines. It is seen that the gain falls at 6dB per octave at frequencies above the break point T_1 and that the phase response is also that of a first order network.

If feedback is applied to this amplifier, the closed-loop gain, G, is given by

$$G = \frac{A}{1+\beta A}$$

where A is the open loop voltage gain

and β is the feedback factor

Now the open-loop amplitude response for the idealised amplifier is

$$A = \frac{A_o}{1+i\omega T_1}$$

where A_o is the zero frequency voltage gain, hence

$$|G| = \left| \frac{A_o}{(1+\beta A_o) + i\omega T_1} \right| = \left| \frac{\dfrac{A_o}{1+\beta A}}{1 + i \left(\dfrac{1}{1+\beta A_o} \right) .\omega T_1} \right|$$

This shows that at low frequencies the gain is approximately 1/β and that the first break point occurs at a new-time constant $\frac{T_1}{1+\beta A_o}$. The system is unconditionally stable since the denominator is essentially positive.

The addition of a capacitative load, C_L, modifies the open loop response as shown by the broken lines in figure III.2. A new break point, T_2 given by $T_2 = C_L R_o$ where R_o is the open loop output resistance.

The modified open loop response is

$$A_c = \frac{A_o}{(1+i\omega T_1)(1+i\omega T_2)}$$

Hence $$|G| = \left| \frac{A_o}{(1+i\omega T_1)(1+i\omega T_2)+\beta A_o} \right|$$

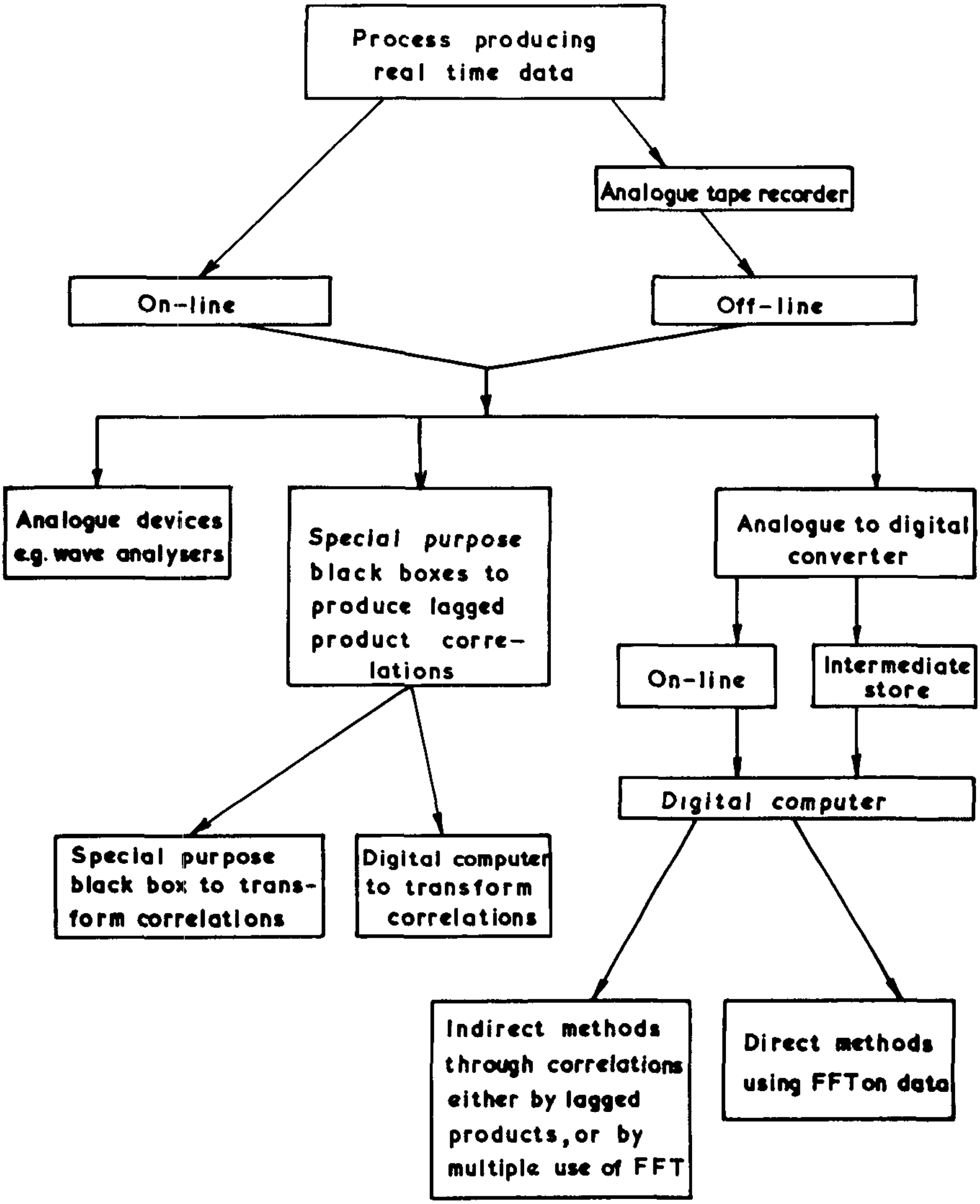
$$= \left| \frac{A_o}{1+\beta A_o - \omega^2 T_1 T_2 + \imath\omega(T_1 + T_2)} \right|$$

$$= \left| \frac{\frac{A_o}{1+\beta A_o}}{1 - \omega^2 \cdot \frac{T_1 \ T_2}{1+\beta A_o} + i \left(\frac{1}{1+\beta A_o} \right) \omega \ (T_1 \ + T_2)} \right|$$

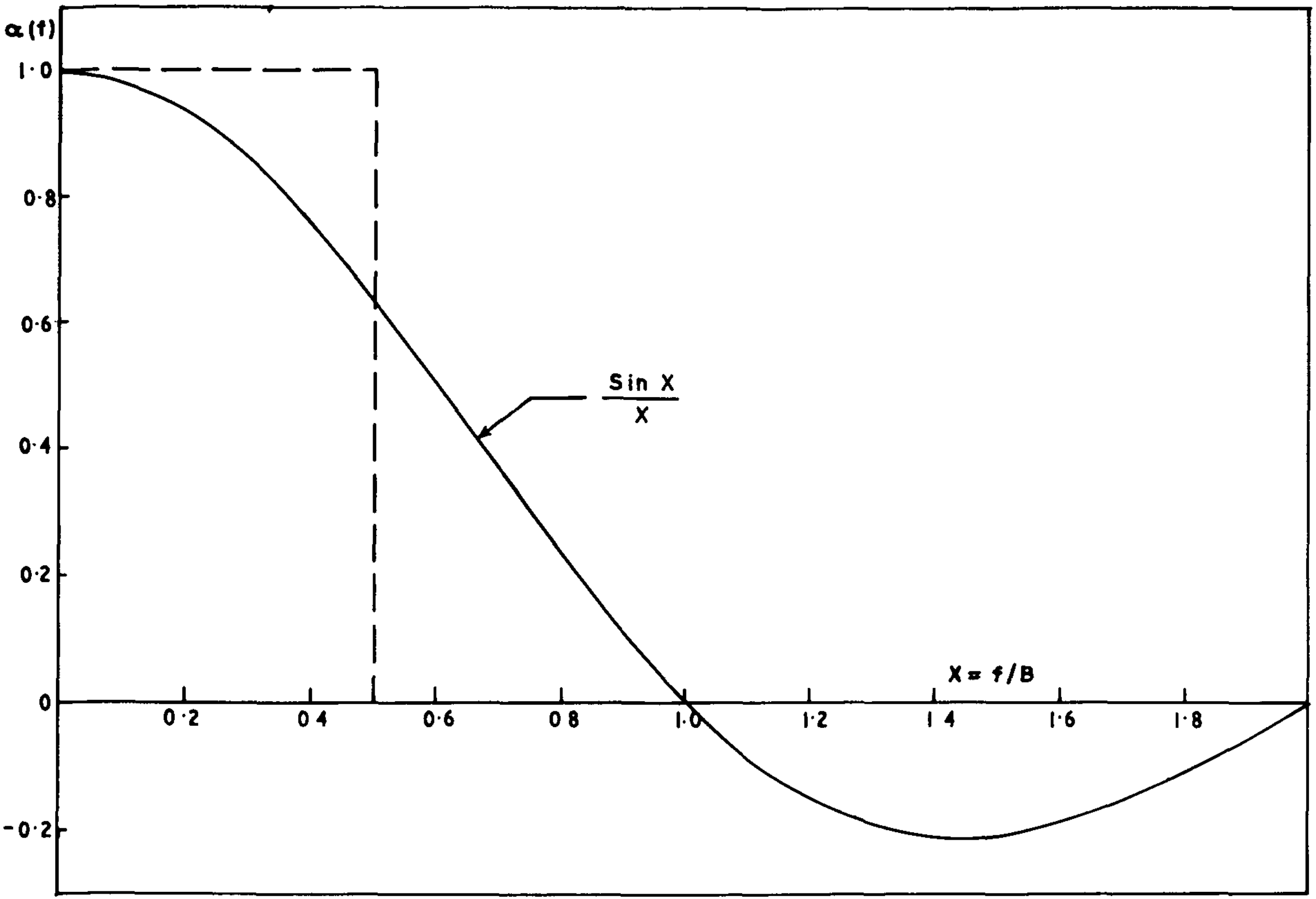
Again the low frequency gain approximates to 1/β. However at higher frequencies, the denominator can become negative hence causing gain peaking with the characteristic response of a second order system. This may be seen qualitatively from the open loop phase response, since if the phase is below 90° a component of positive feedback is introduced.

III.3 Since the tests in this report were carried out with the temporary line drivers, an improved version has been designed for permanent use. These provide a frequency response flat within ±2% up to 80 KHz.

FIG. 1.1

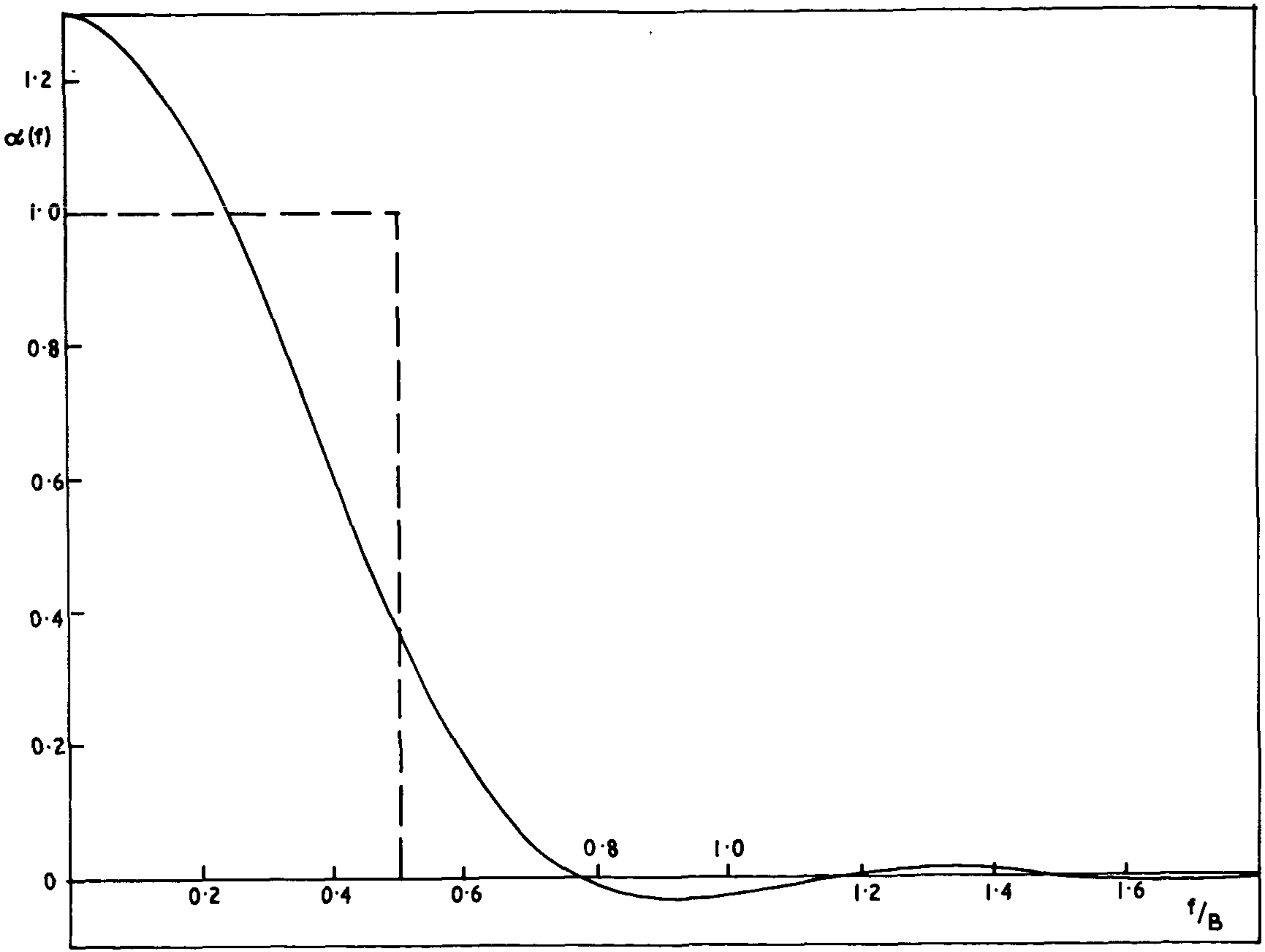

Process producing real time data
Analogue tape recorder
On-line
Off-line
Analogue devices e.g. wave analysers
Special purpose black boxes to produce lagged product correlations
Analogue to digital converter
On-line
Intermediate store
Digital computer
Special purpose black box to transform correlations
Digital computer to transform correlations
Indirect methods through correlations either by lagged products, or by multiple use of FFT
Direct methods using FFT on data

A variety of paths to the production of spectral estimates



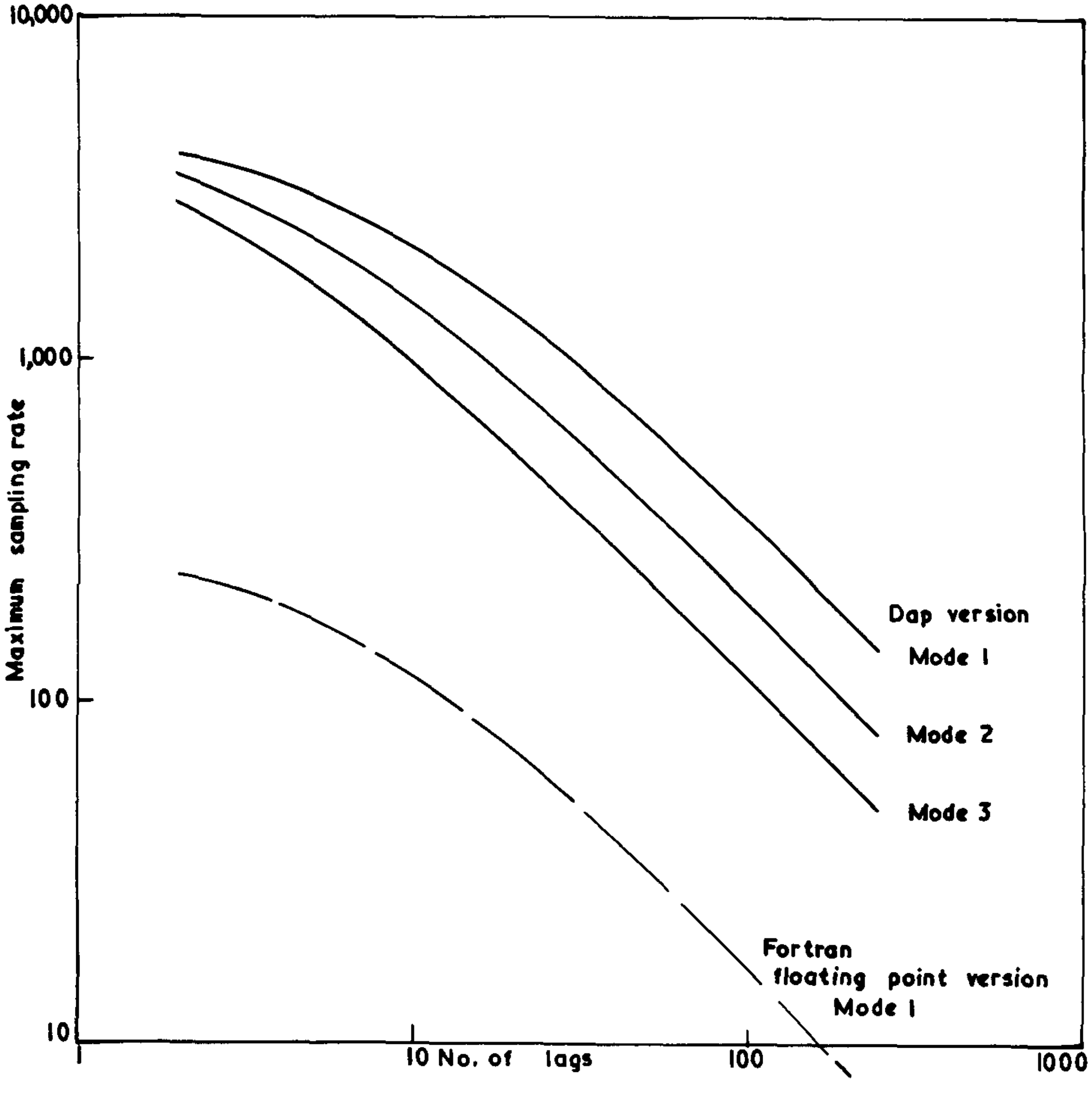
The spectral window for the real time correlation technique. No Hanning.

FIG. 2.1(b)



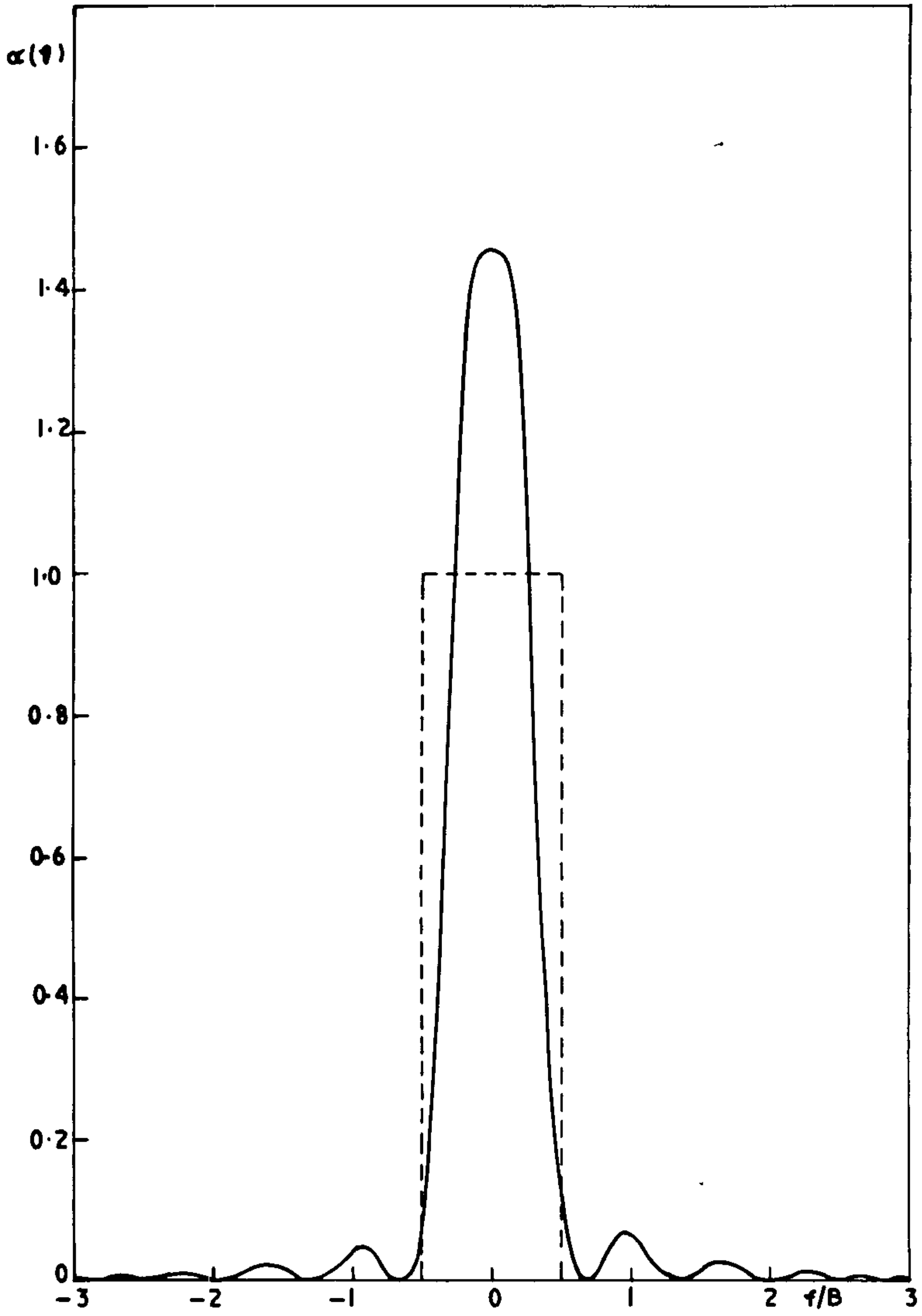
The spectral window for the real time correlation technique, with Hanning.

FIG. 2.2



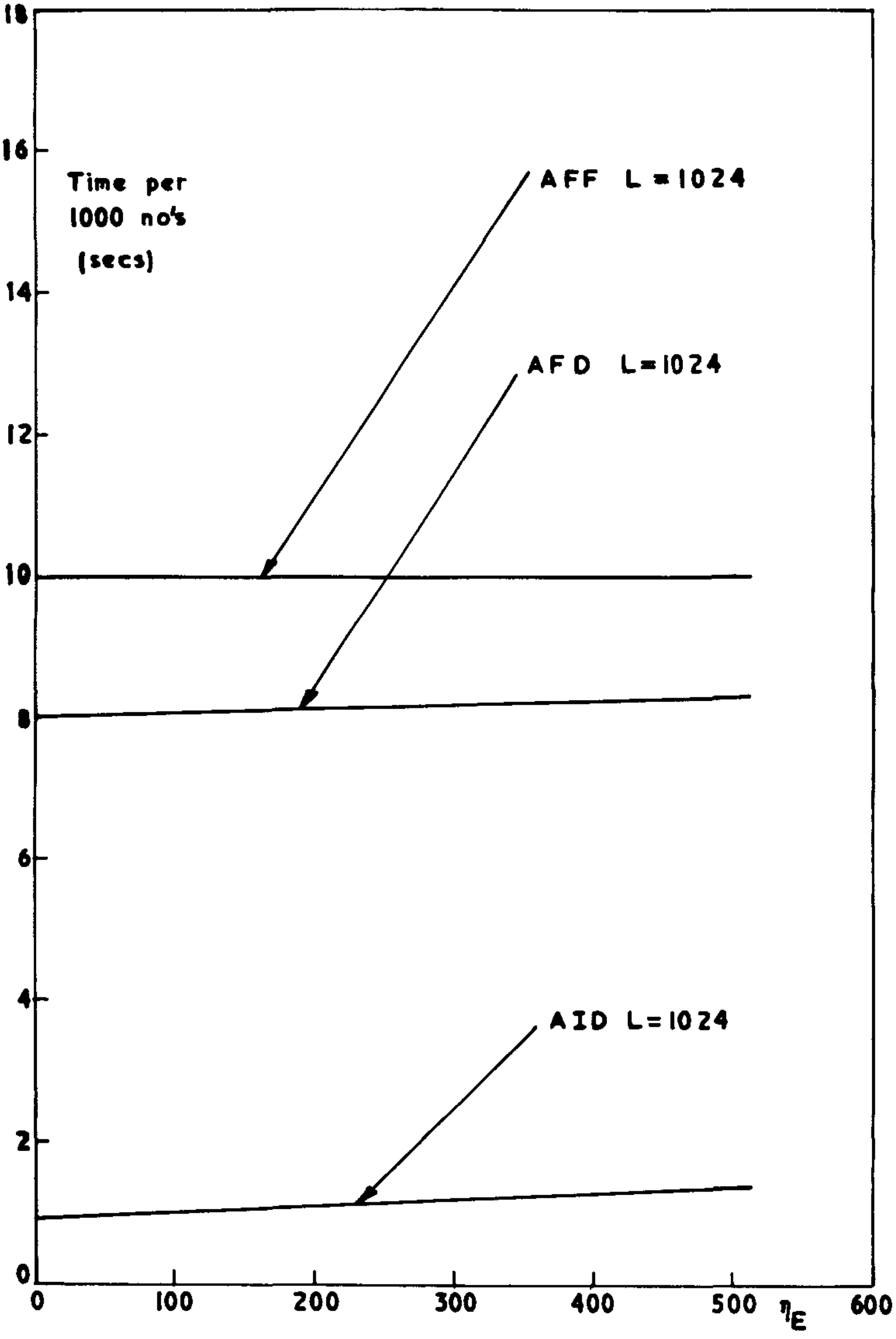
Maximum allowable sampling rate for real time correlation technique

FIG. 2.3



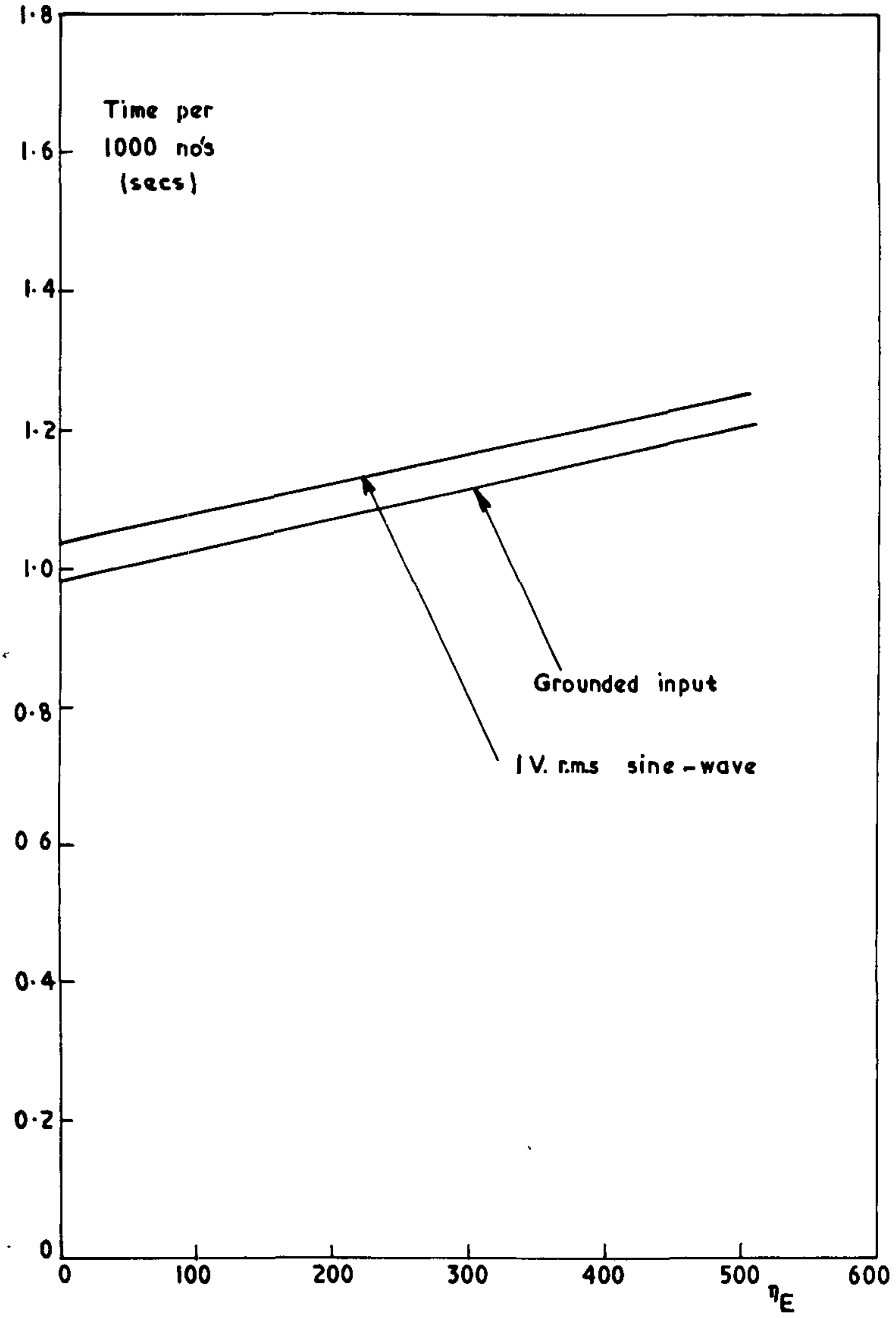
The spectral window for the periodogram

FIG. 2.4



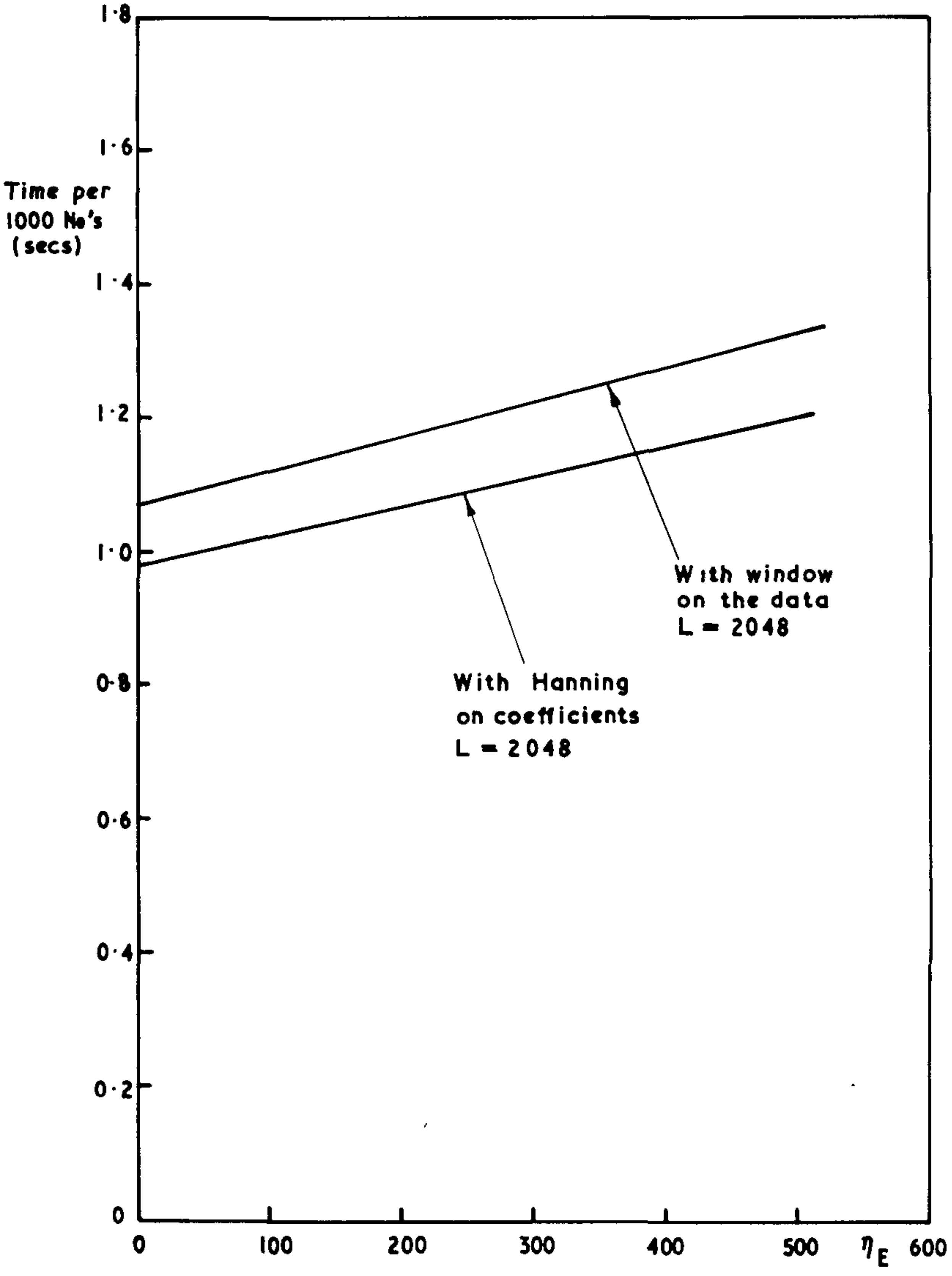
Variation of the computation rates of the programs AID., AFD and AFF with the number of final estimates.

FIG. 2.5



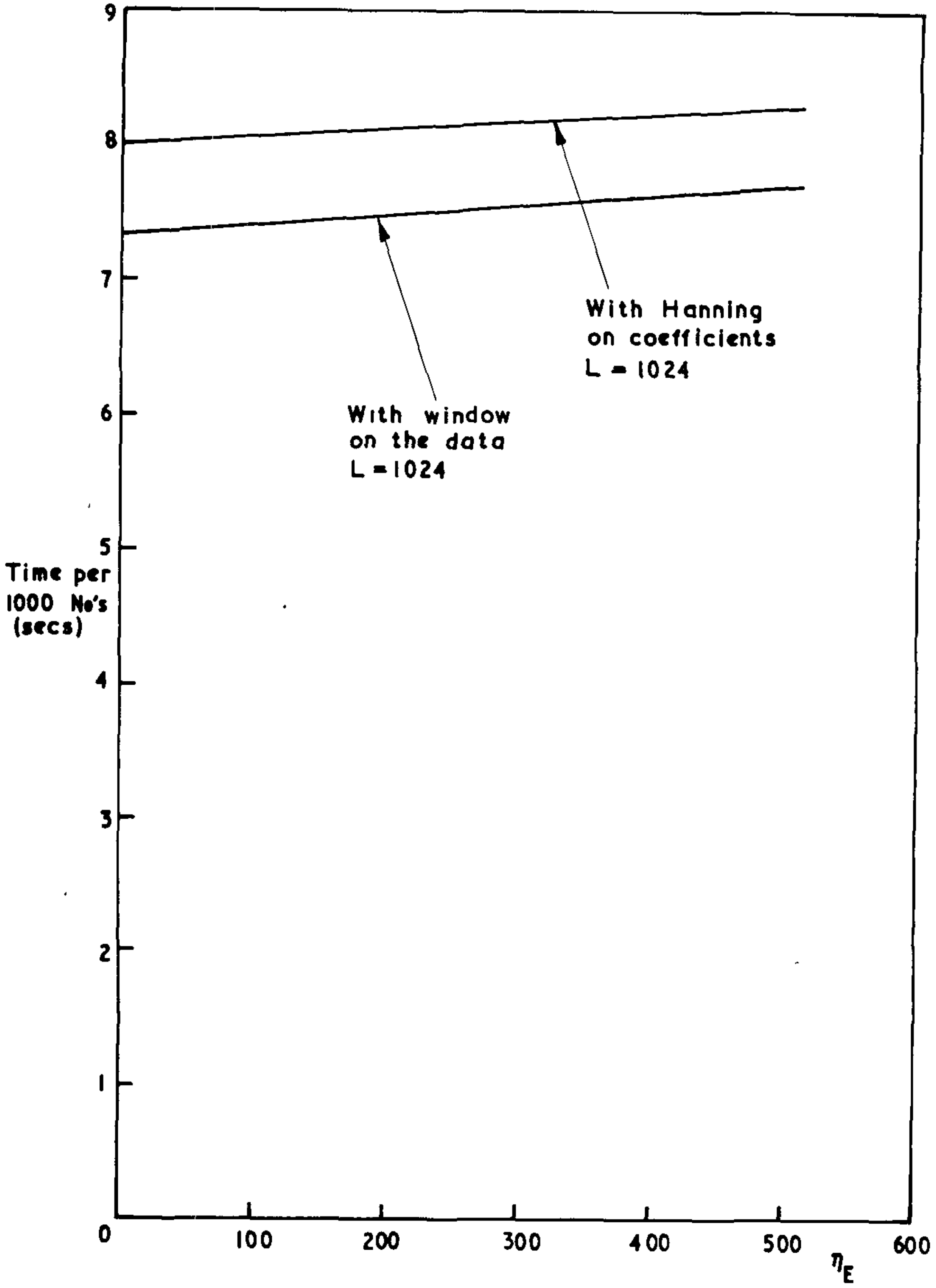
Variation of the computation rate of the AID programs with the number of final estimates. Effect of input scale (M = 10)

FIG. 2 . 6



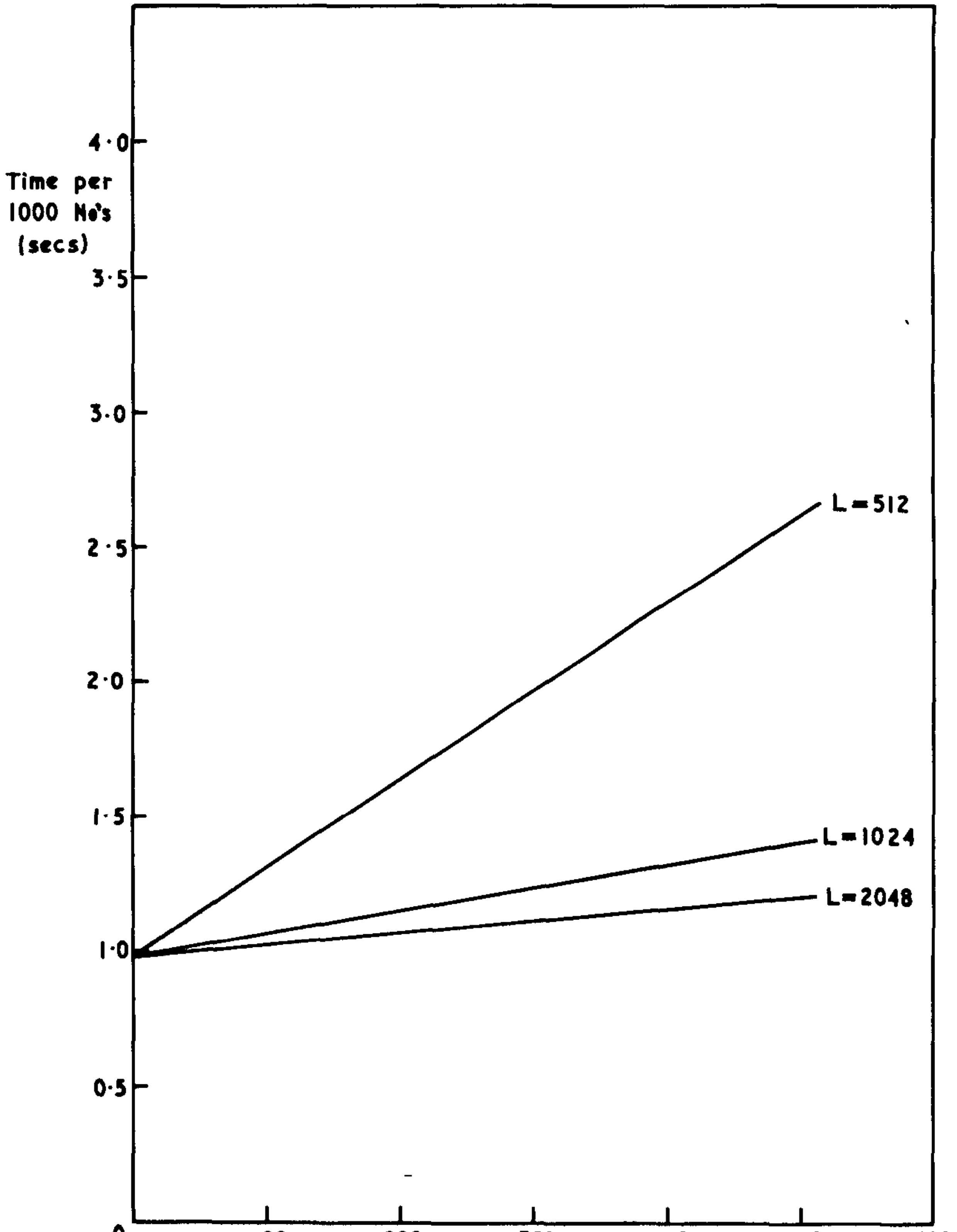
Variation of the computation rate of the AID program with the number of final estimates. Two smoothing methods

FIG 2.7



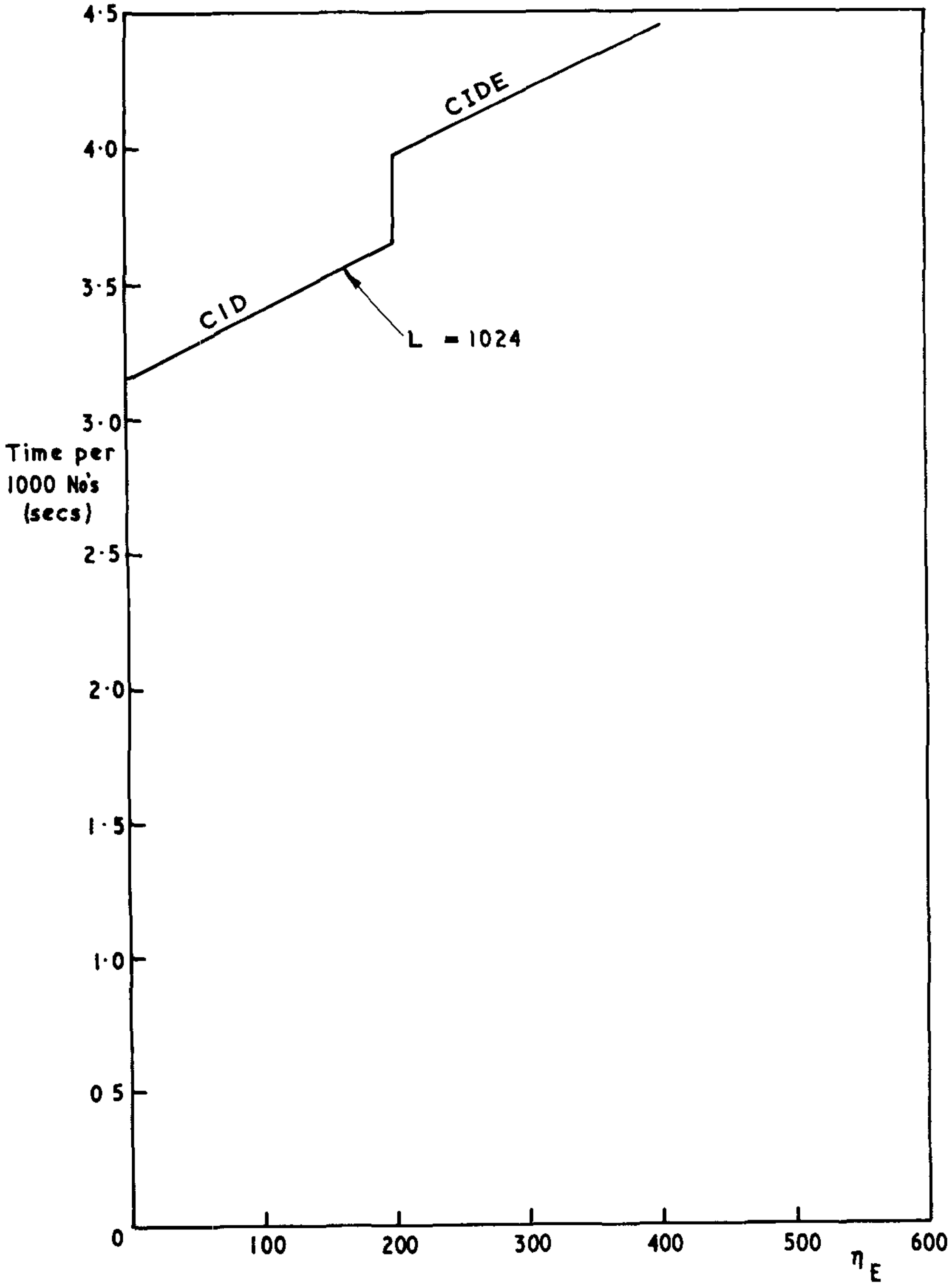
Variation of the computation rate of the AFD program with the number of final estimates. Two smoothing methods

FIG. 2.8



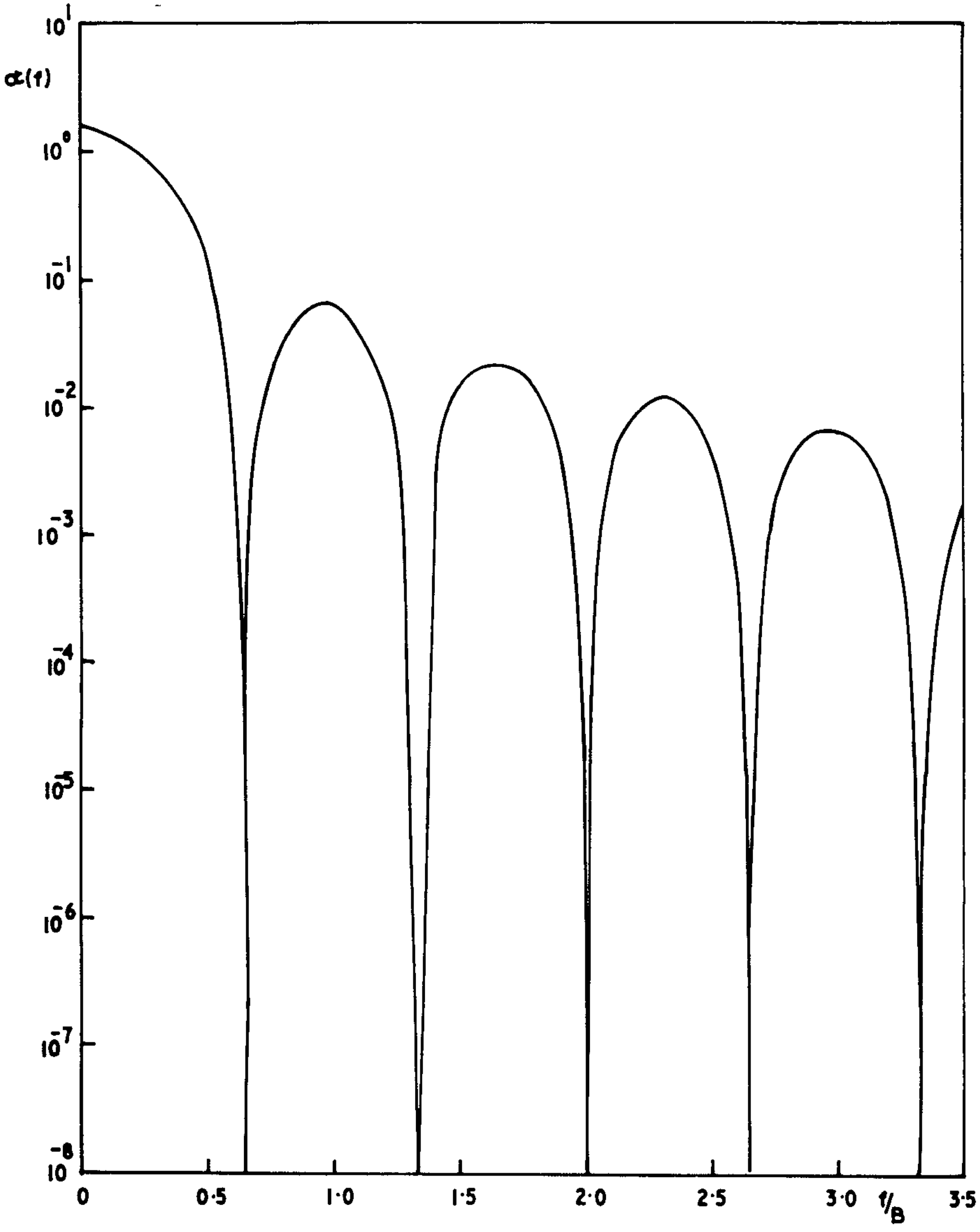
Variation of the computation rate of the AID program with the number of final estimates, for various block sizes.

FIG 2.9



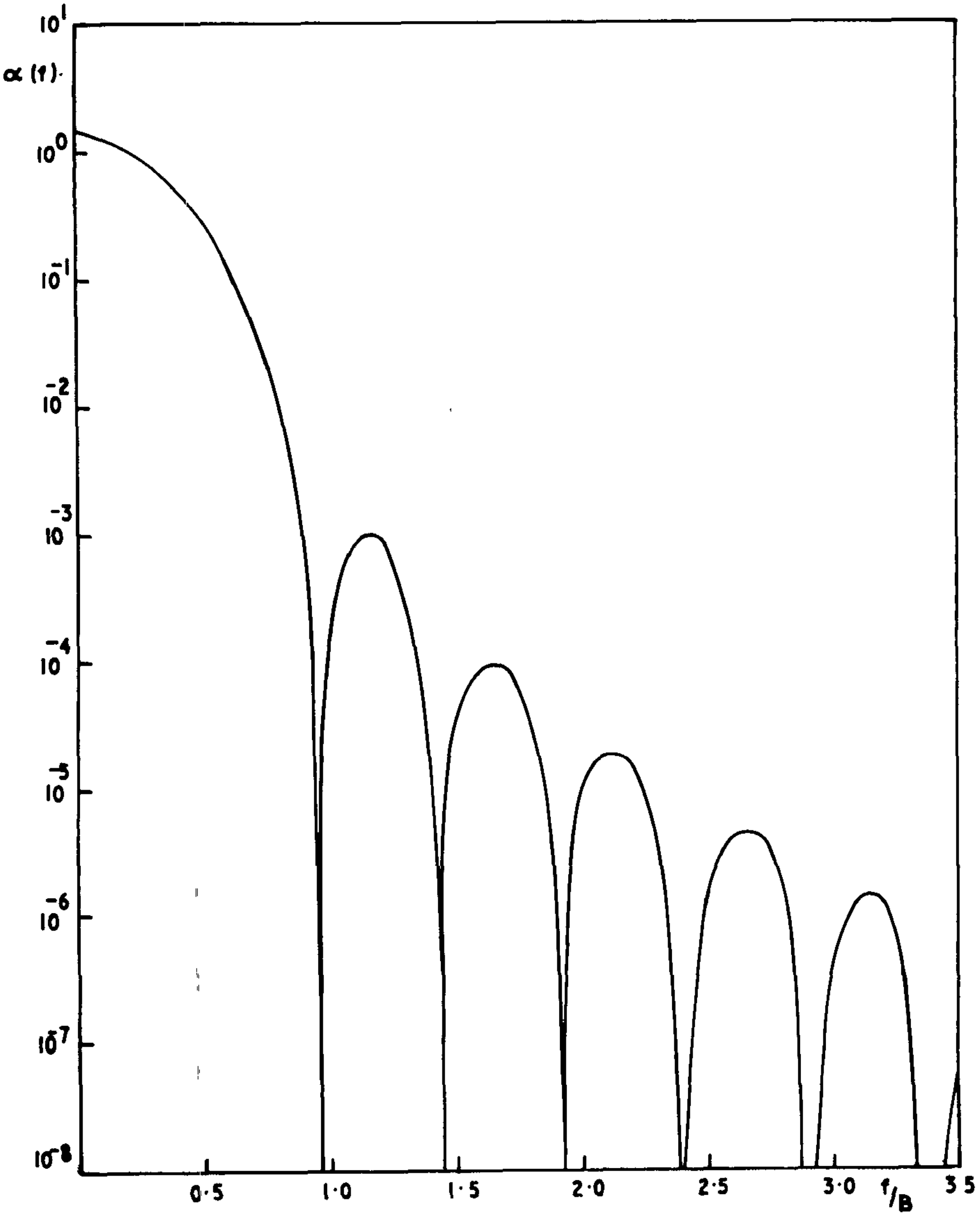
Variation of the computation rate of the CID and CIDE programs with the number of final estimates.

FIG. 2 10



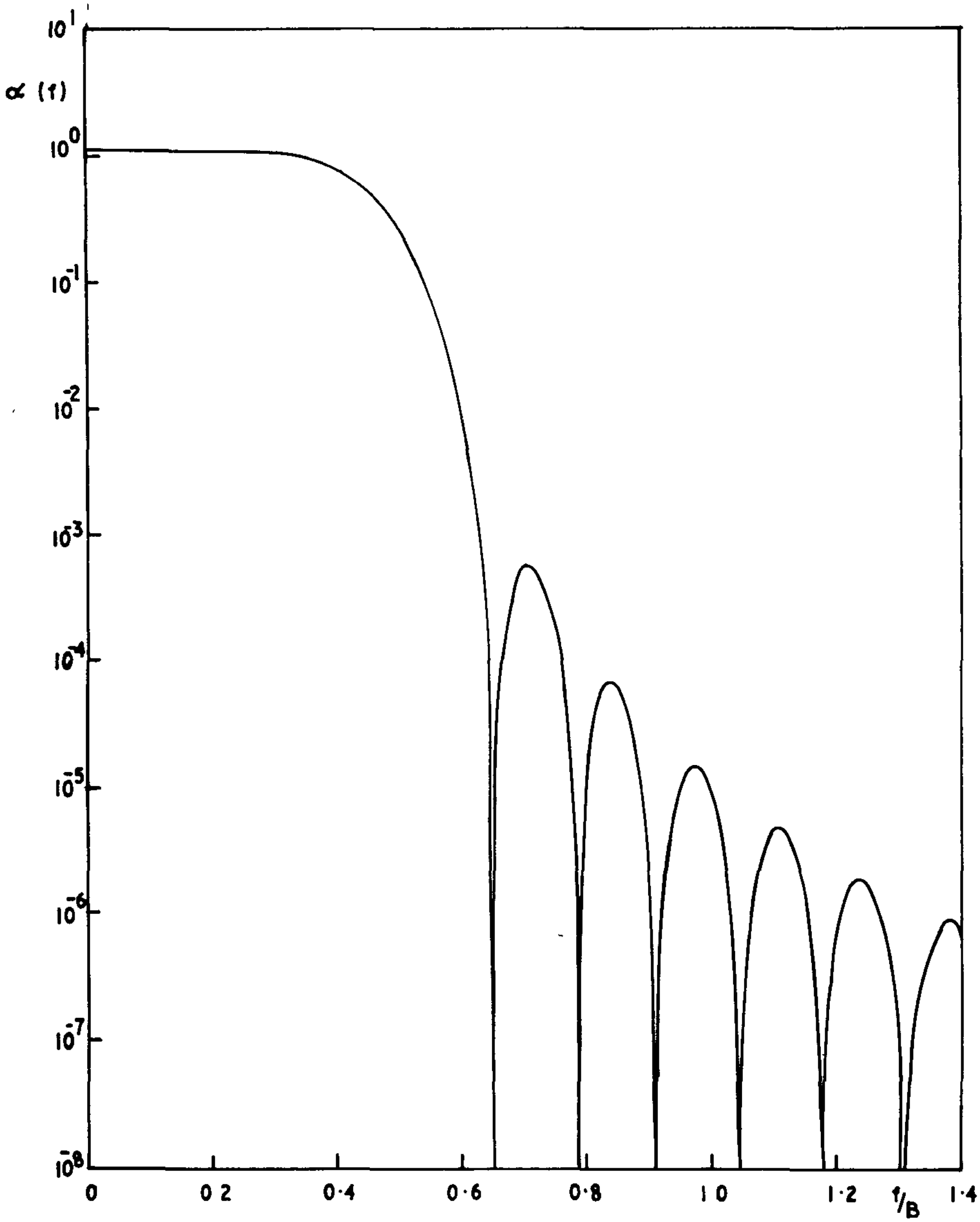
The spectral window for the periodogram

FIG. 2.11



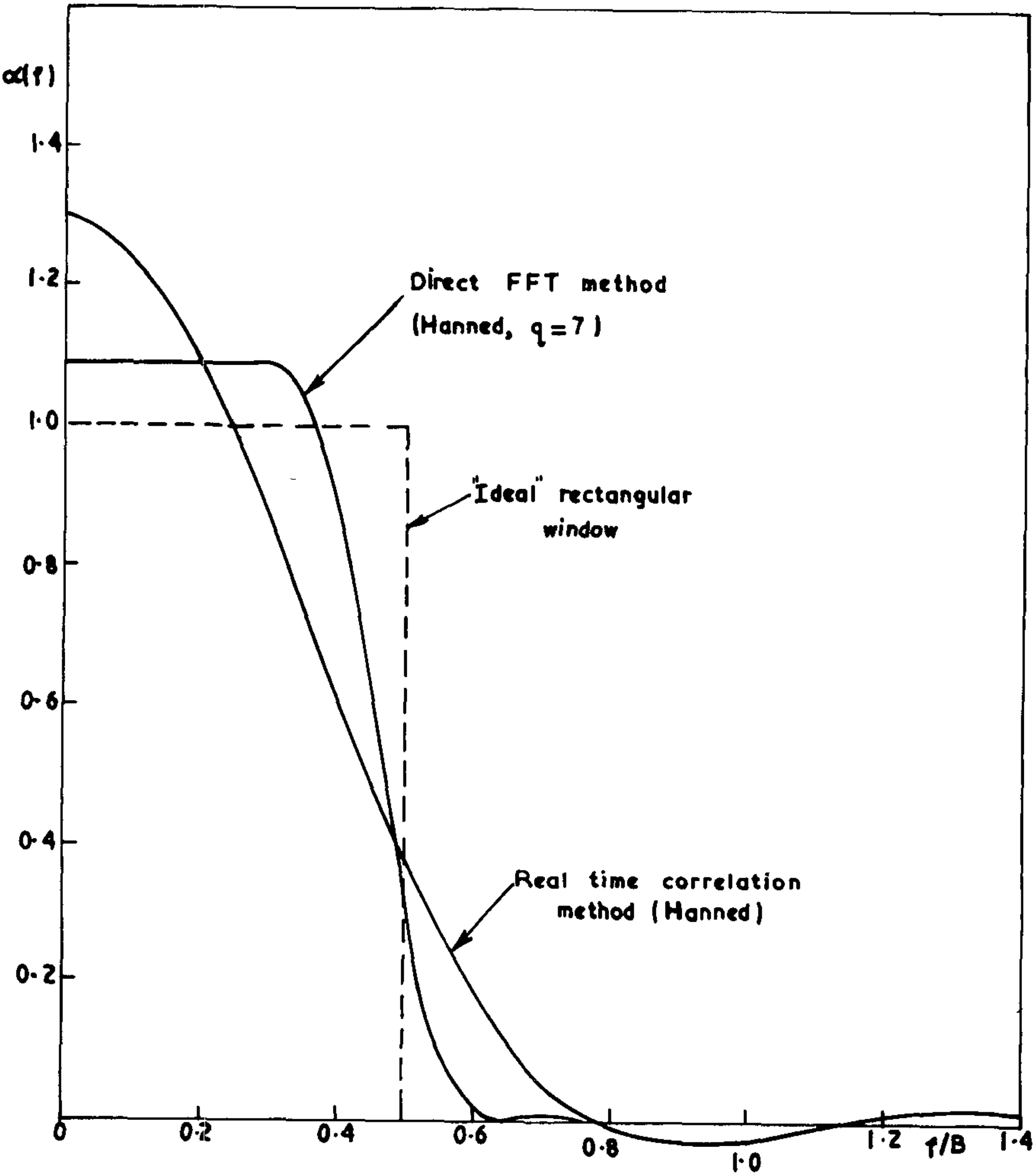
The spectral window for the Hanned periodogram

FIG. 2.12



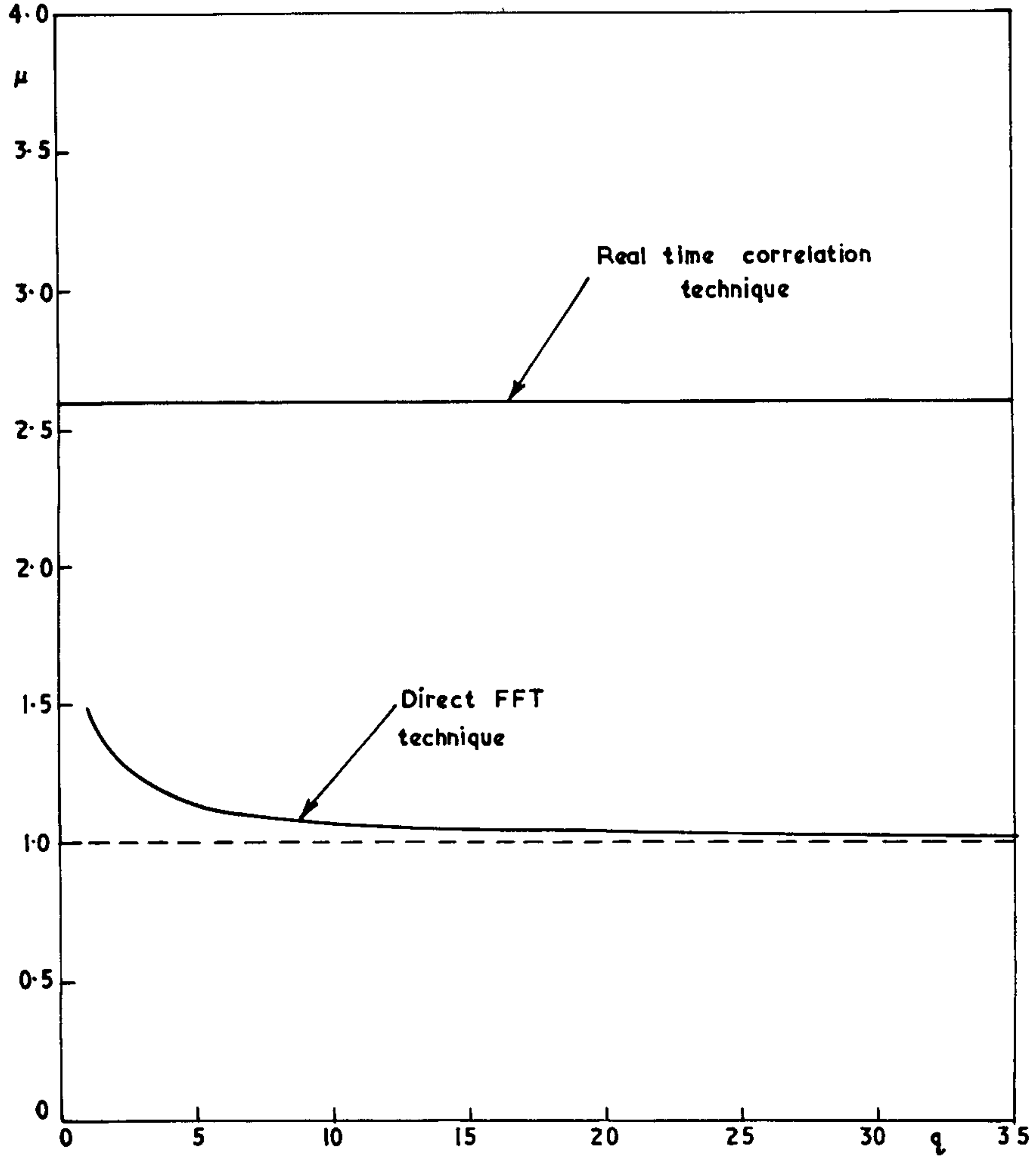
Typical spectral window for spectral estimates obtained from the direct FFT method (q = 7)

FIG. 2.13



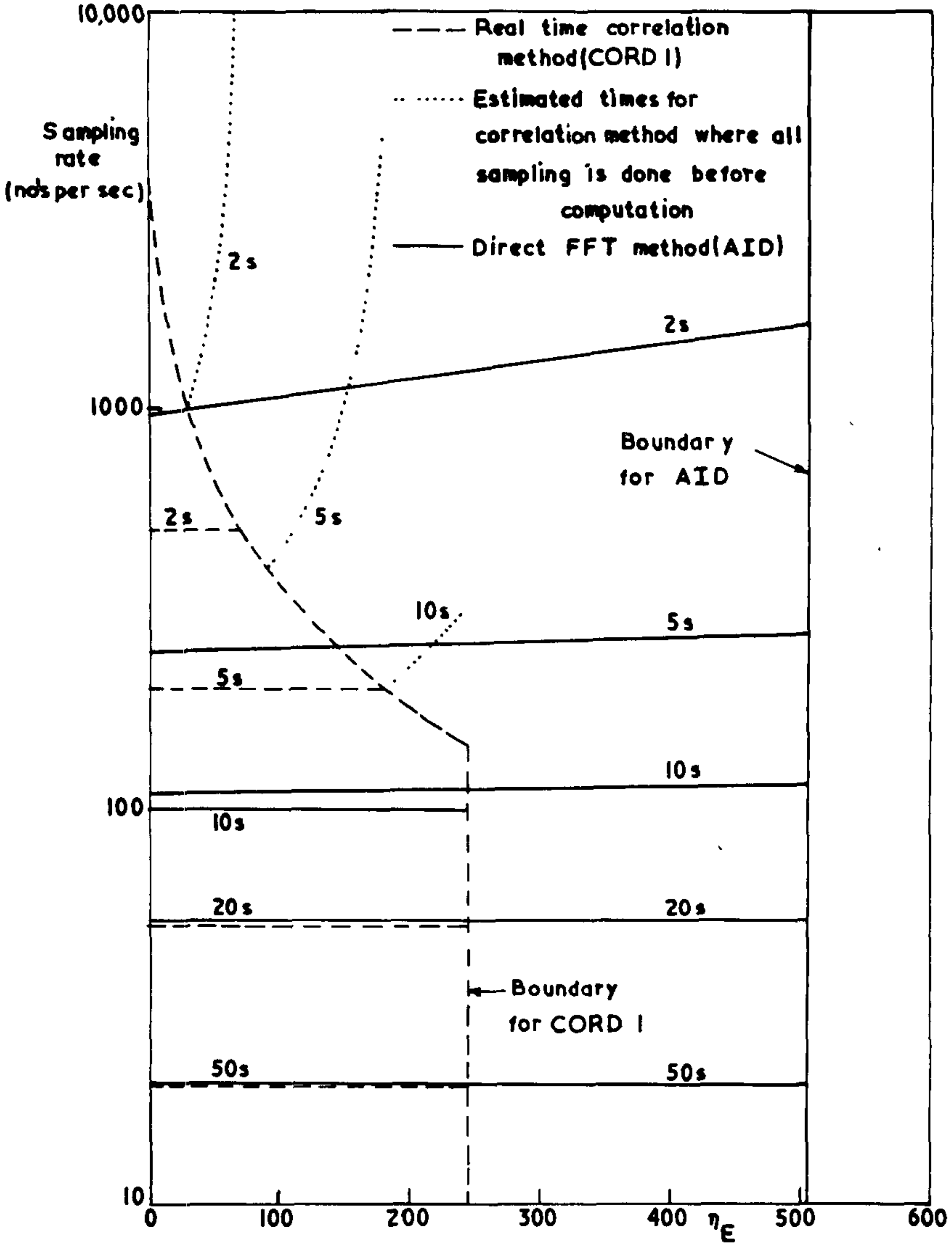
A comparison of the spectal windows obtained from the direct FFT method and the real time correlation method

FIG. 2.14



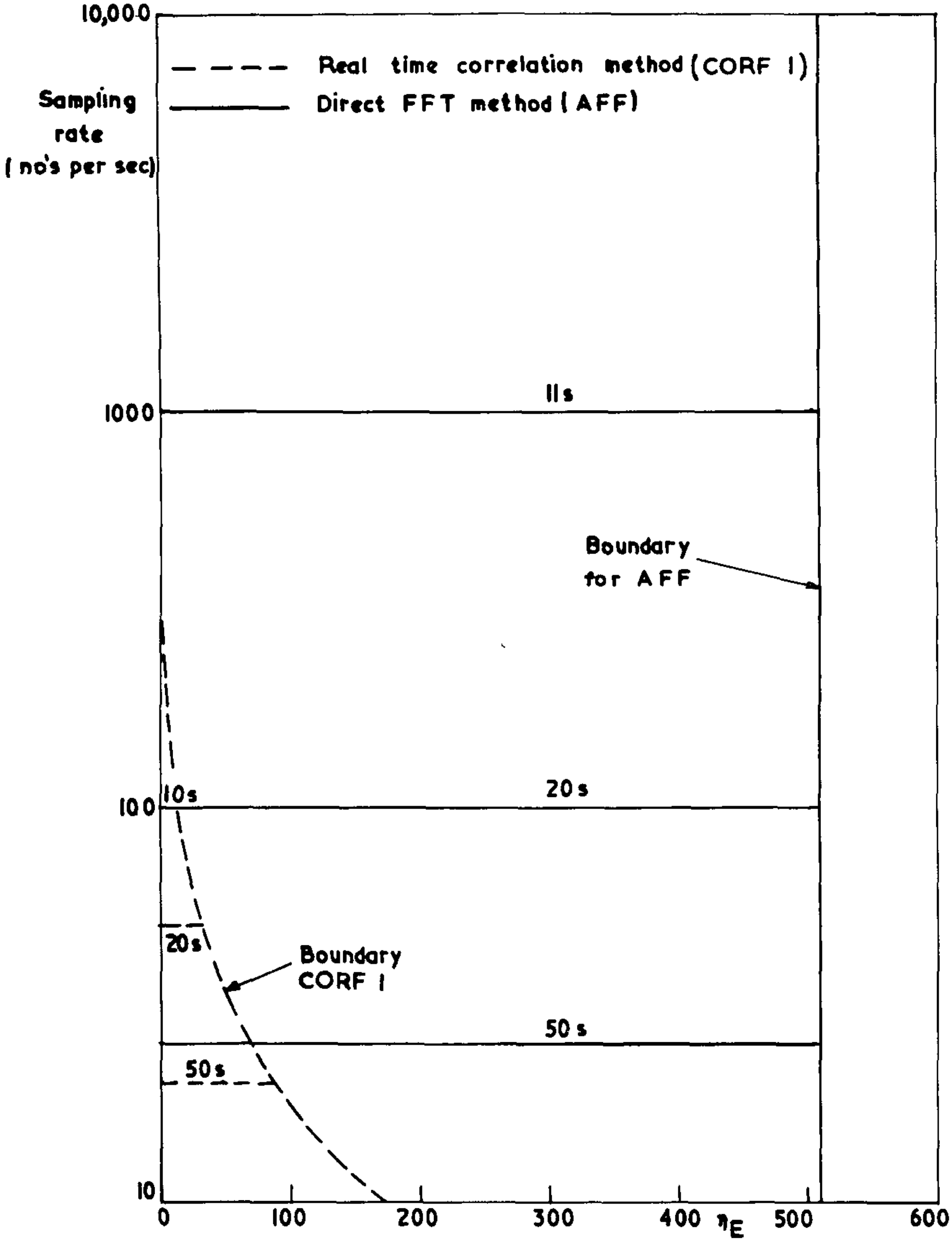
Variation of the statistical bandwidth with the number of periodogram estimates per final estimate

FIG. 2.15



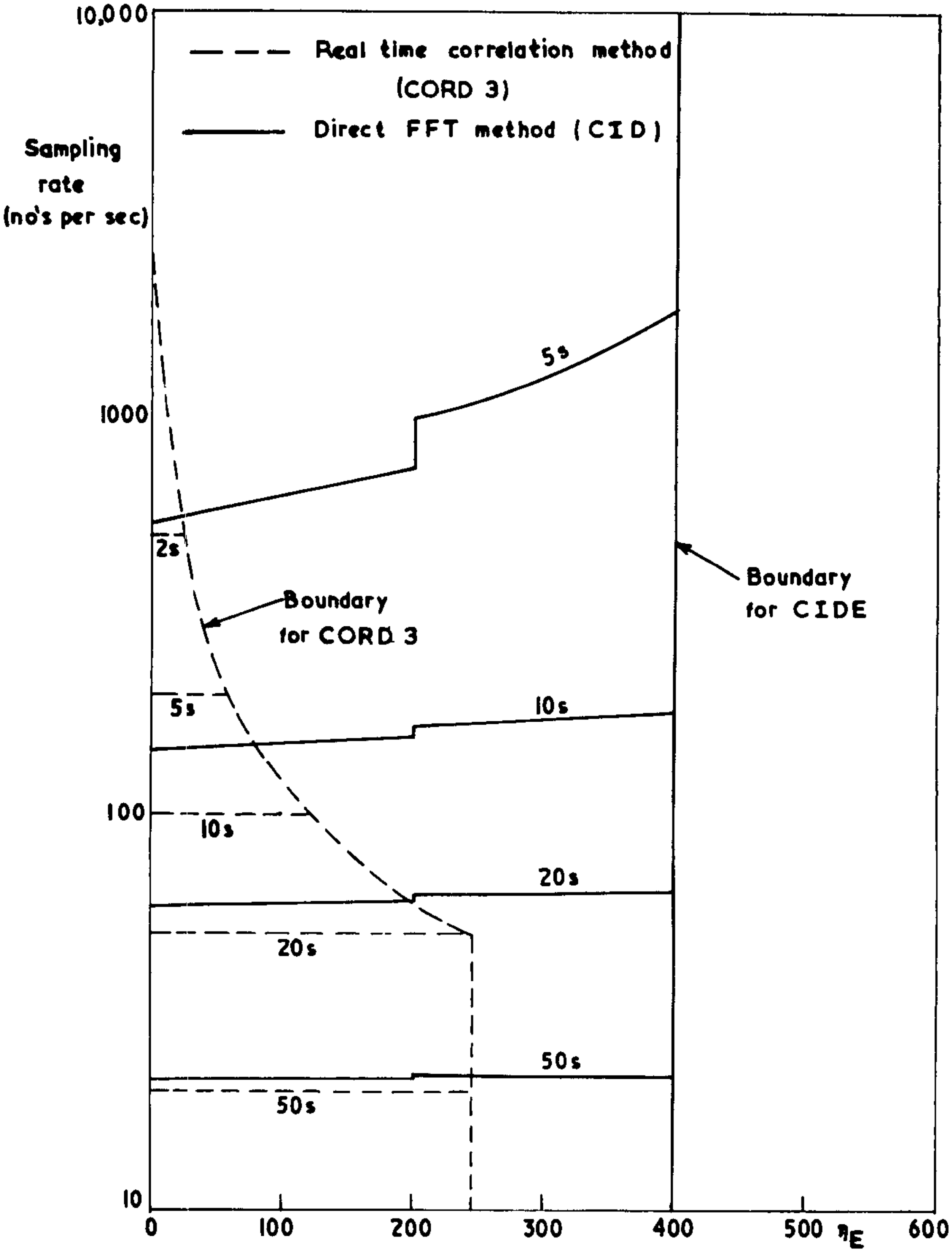
Contours of total computer time per 1000 no's on the sampling rate / final number of estimates plane

FIG. 2.16



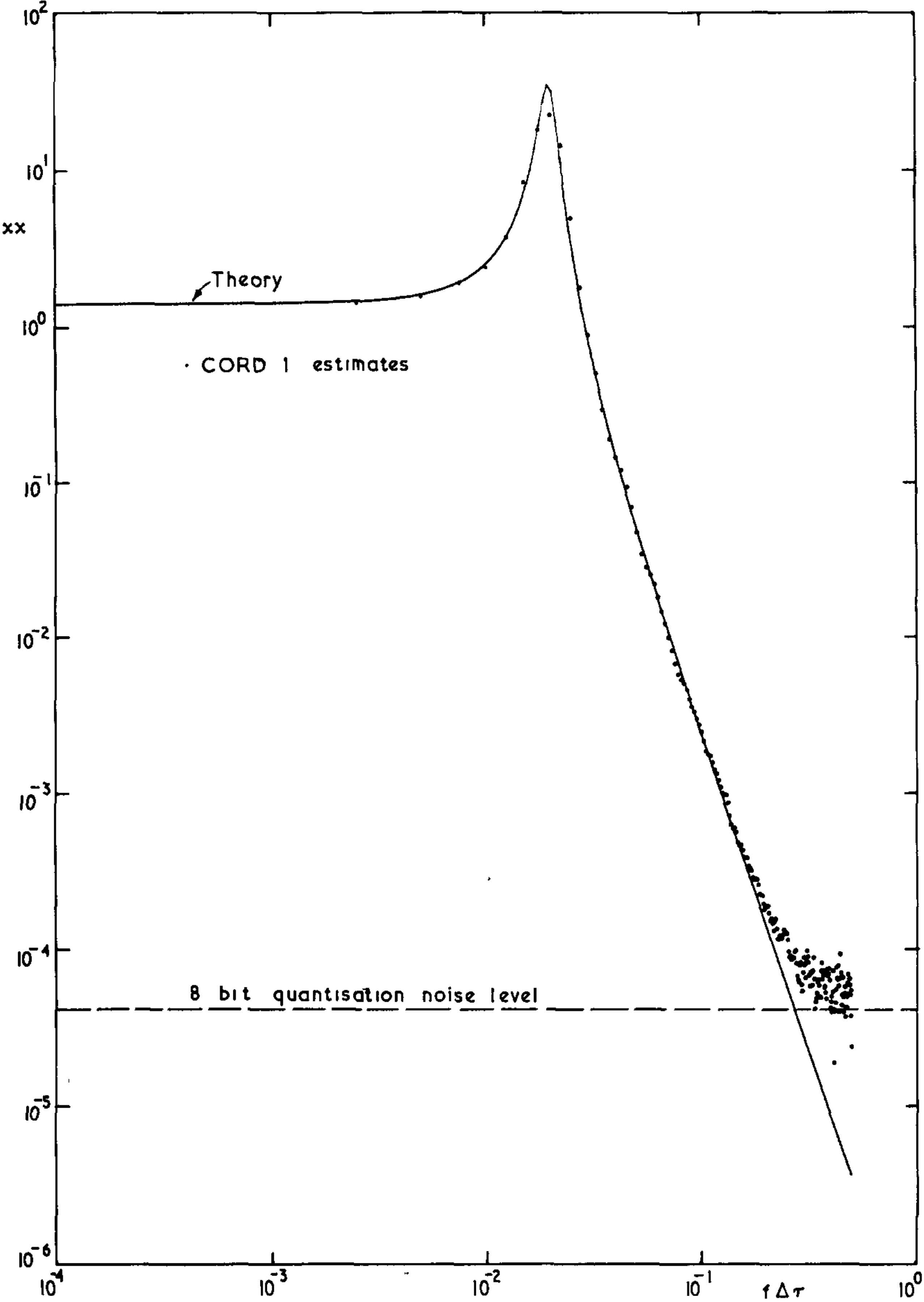
Contours of total computer time per 1000 no's on the sampling rate /final number of estimates plane

FIG. 2.17



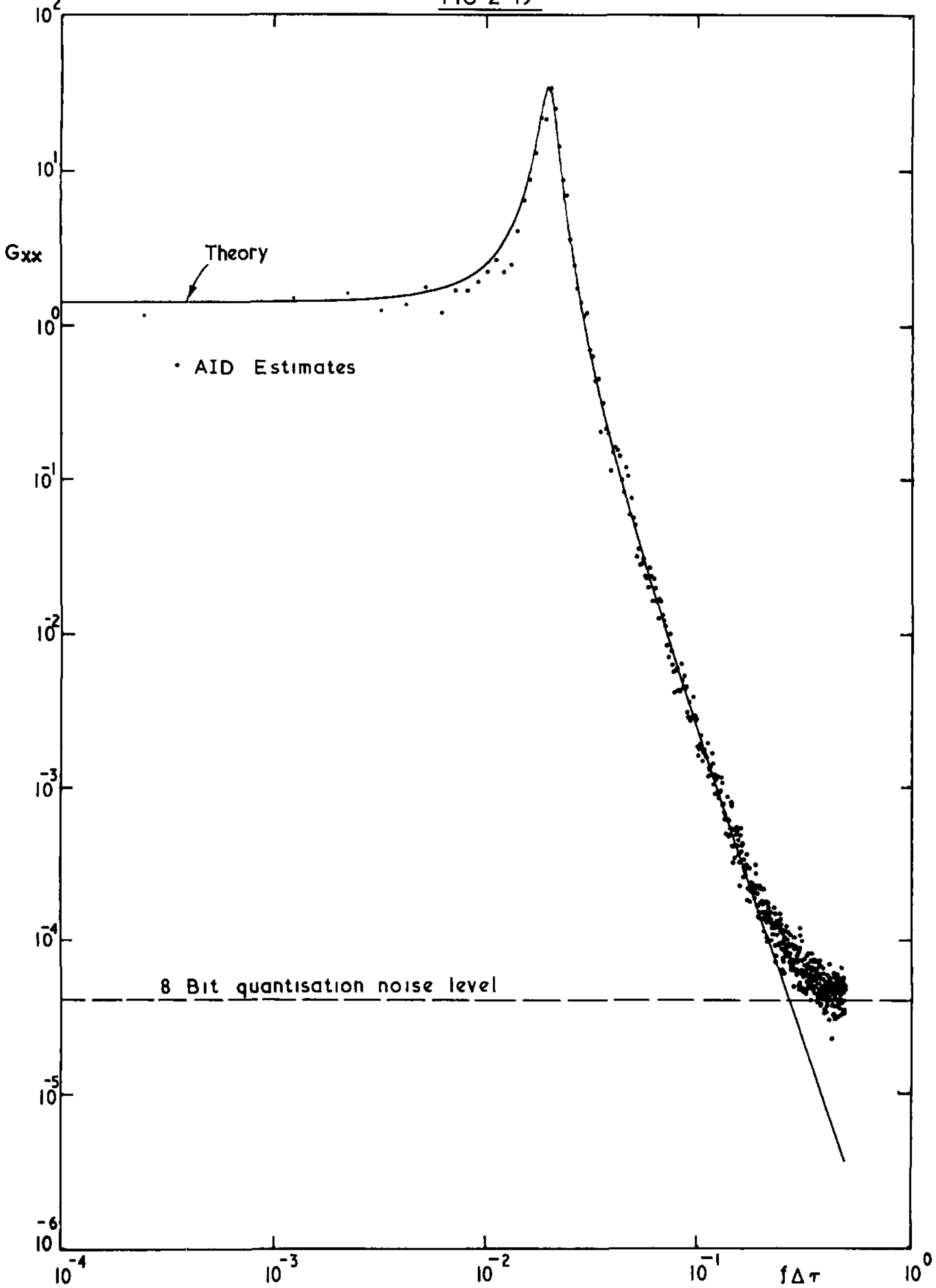
Contours of total time per 1000 no's on the sampling rate / final number of estimates plane

FIG 2 18



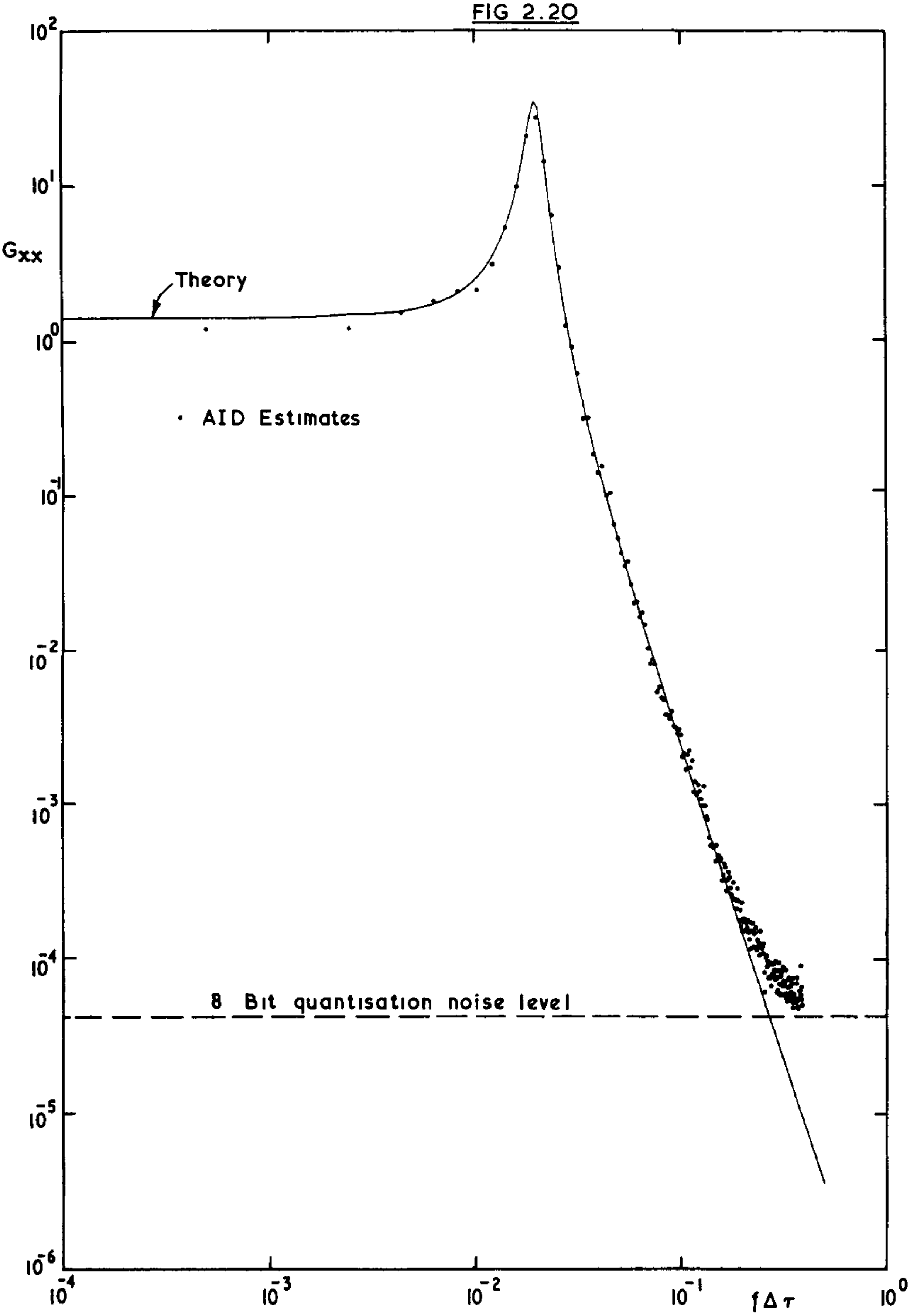
Comparison of the estimates obtained from CORD 1 with theory $\eta_E = 200$

FIG 2 19



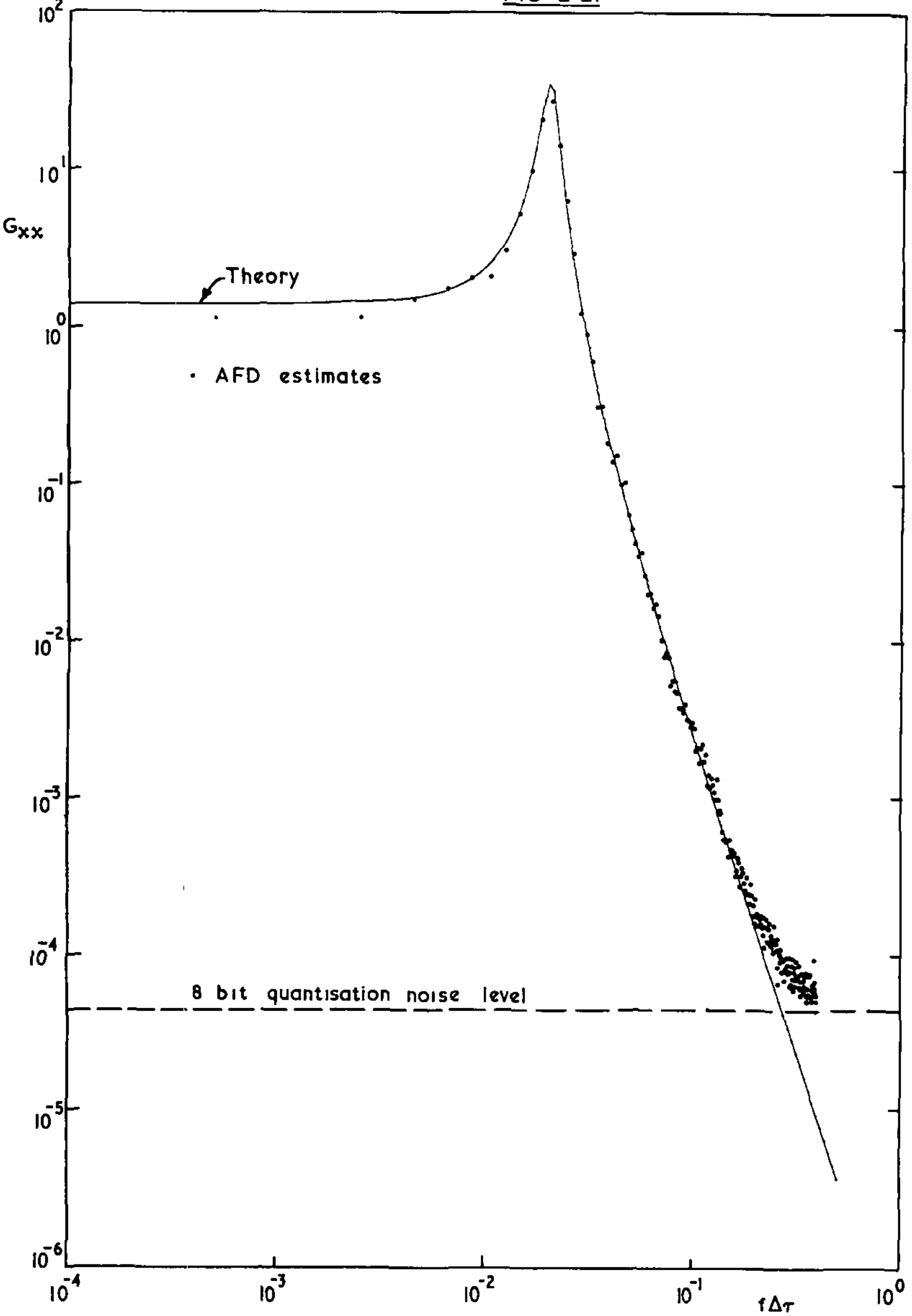
Comparison of the estimates obtained from AID with theory L = 2048, η_E = 512

FIG 2.20



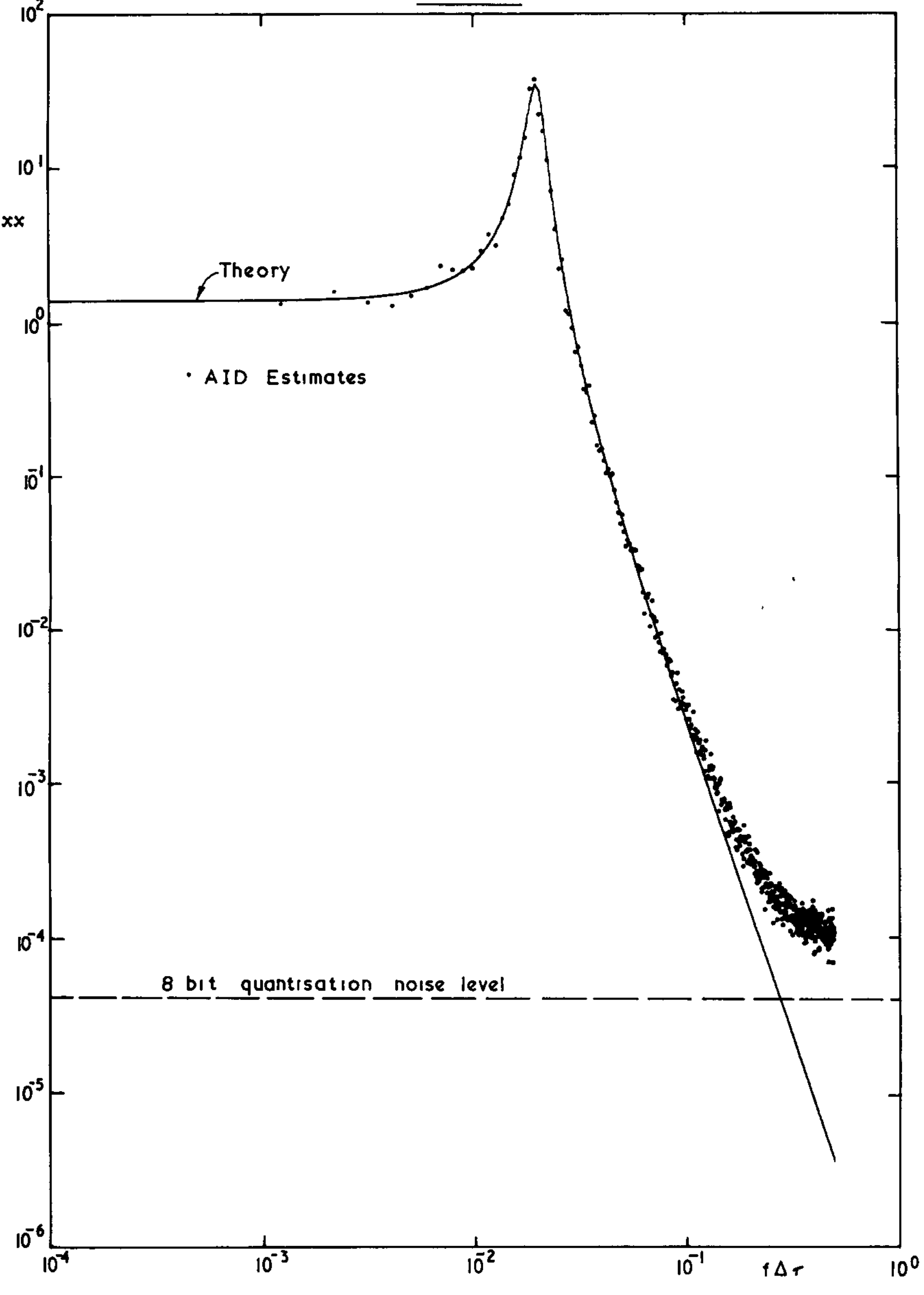
Comparison of the estimates obtained from AID with theory $L = 1024, \eta_E = 200$

FIG 2 21



Comparison of the estimates obtained from AFD with theory $L = 1024$, $\eta_E = 200$

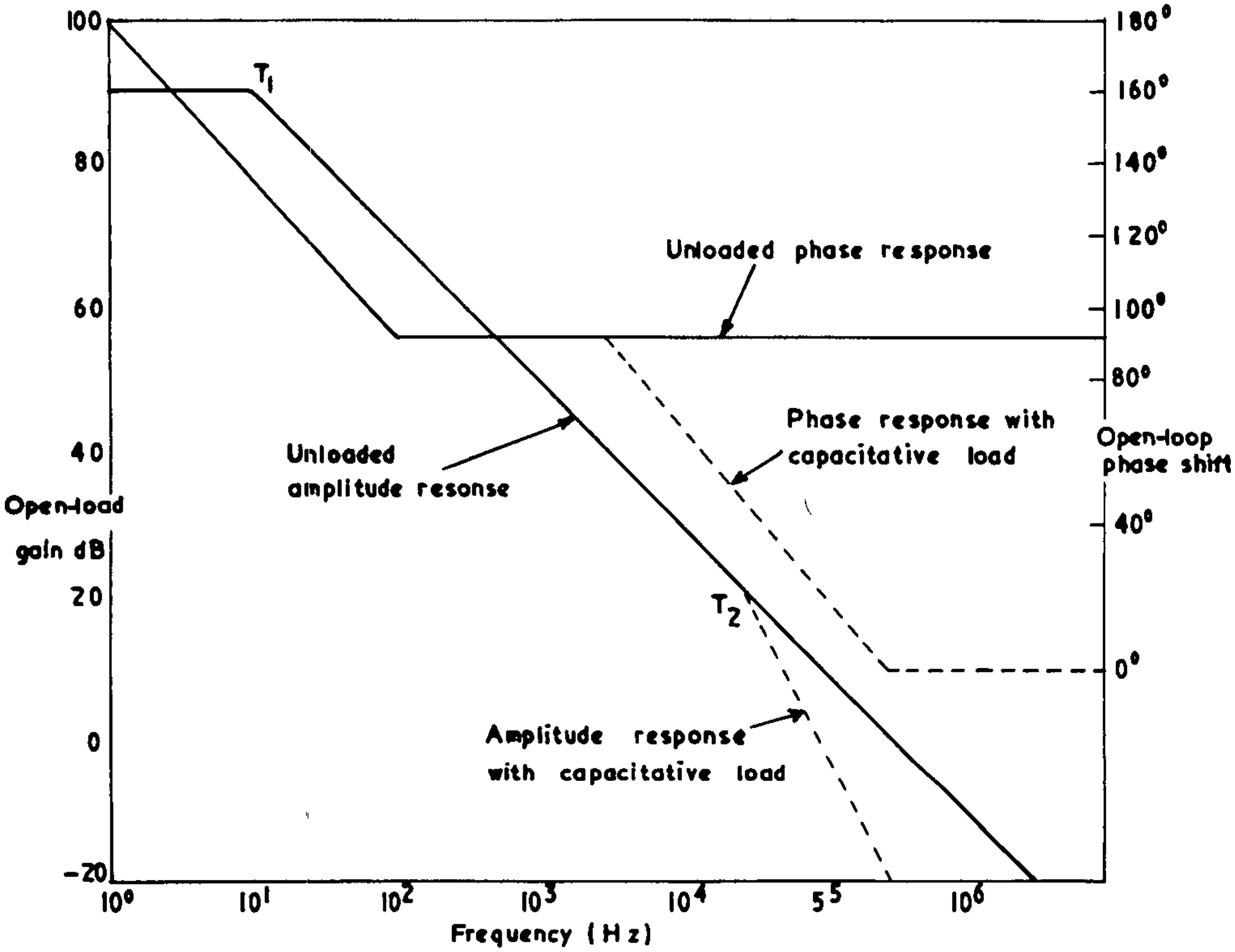
FIG 2 22



Comparison of the estimates obtained from AID (Without Hanning) with theory

$L = 2048$, $\eta_E = 512$

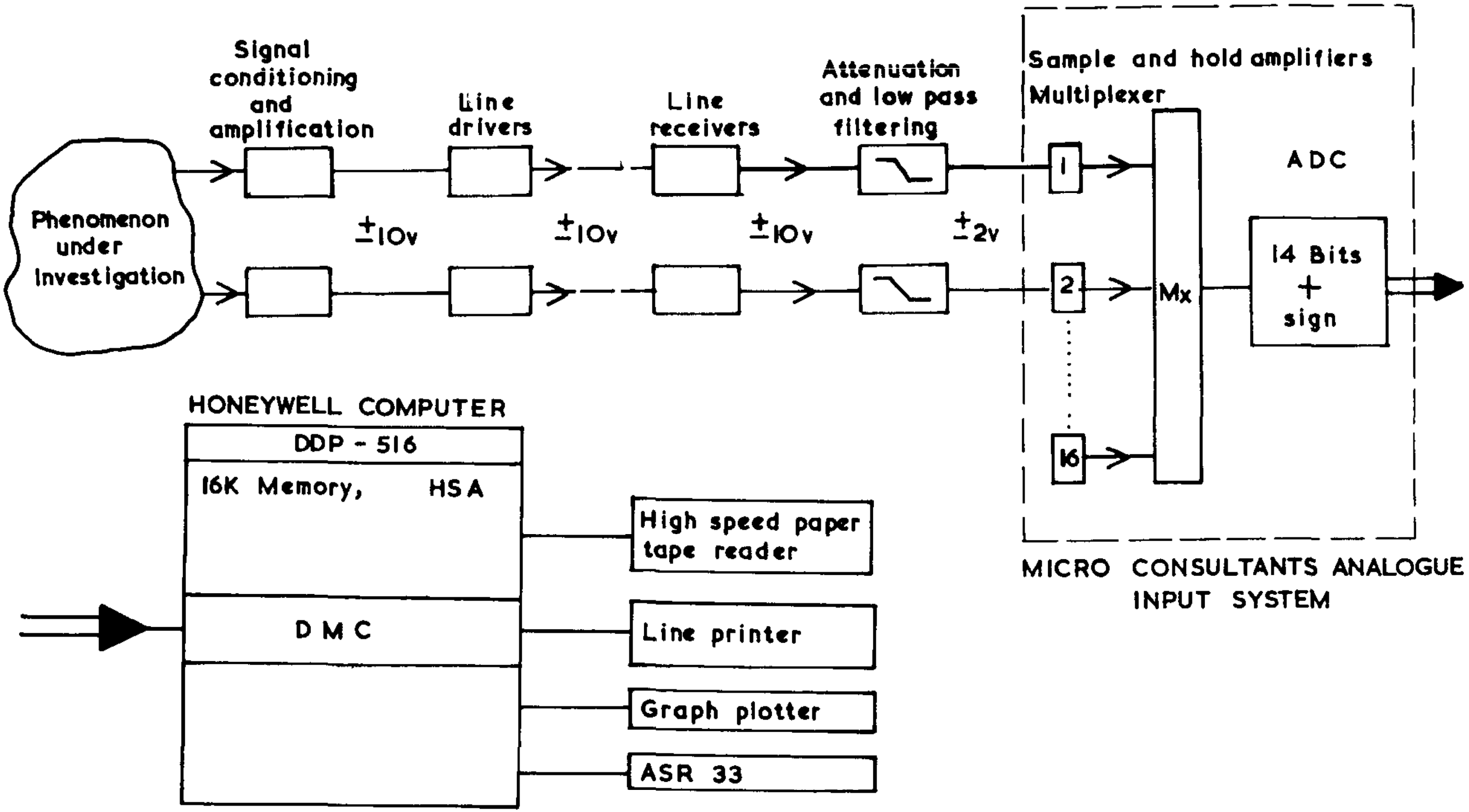
FIG.III.2



Idealised open-loop amplitude and phase responses showing effect of capacitative load

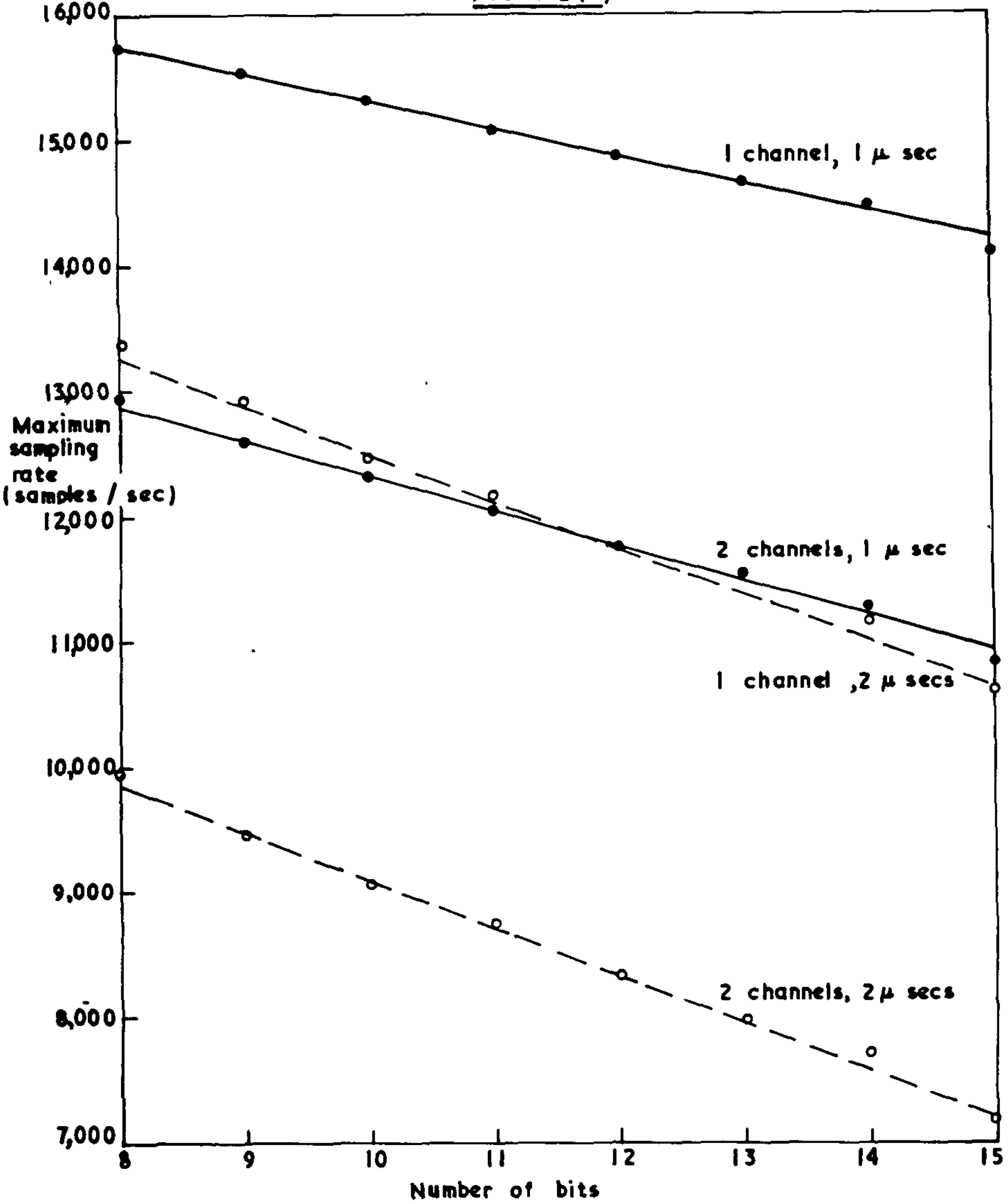
NPL AERO REPORT 1328

FIG 3.1


Signal conditioning and amplification
Line drivers
Line receivers
Attenuation and low pass filtering
Sample and hold amplifiers
Multiplexer
ADC
Phenomenon under Investigation
±10v
±10v
±10v
±2v
1
2
16
Mx
14 Bits + sign
MICRO CONSULTANTS ANALOGUE INPUT SYSTEM
HONEYWELL COMPUTER
DDP - 516
16K Memory, HSA
DMC
High speed paper tape reader
Line printer
Graph plotter
ASR 33

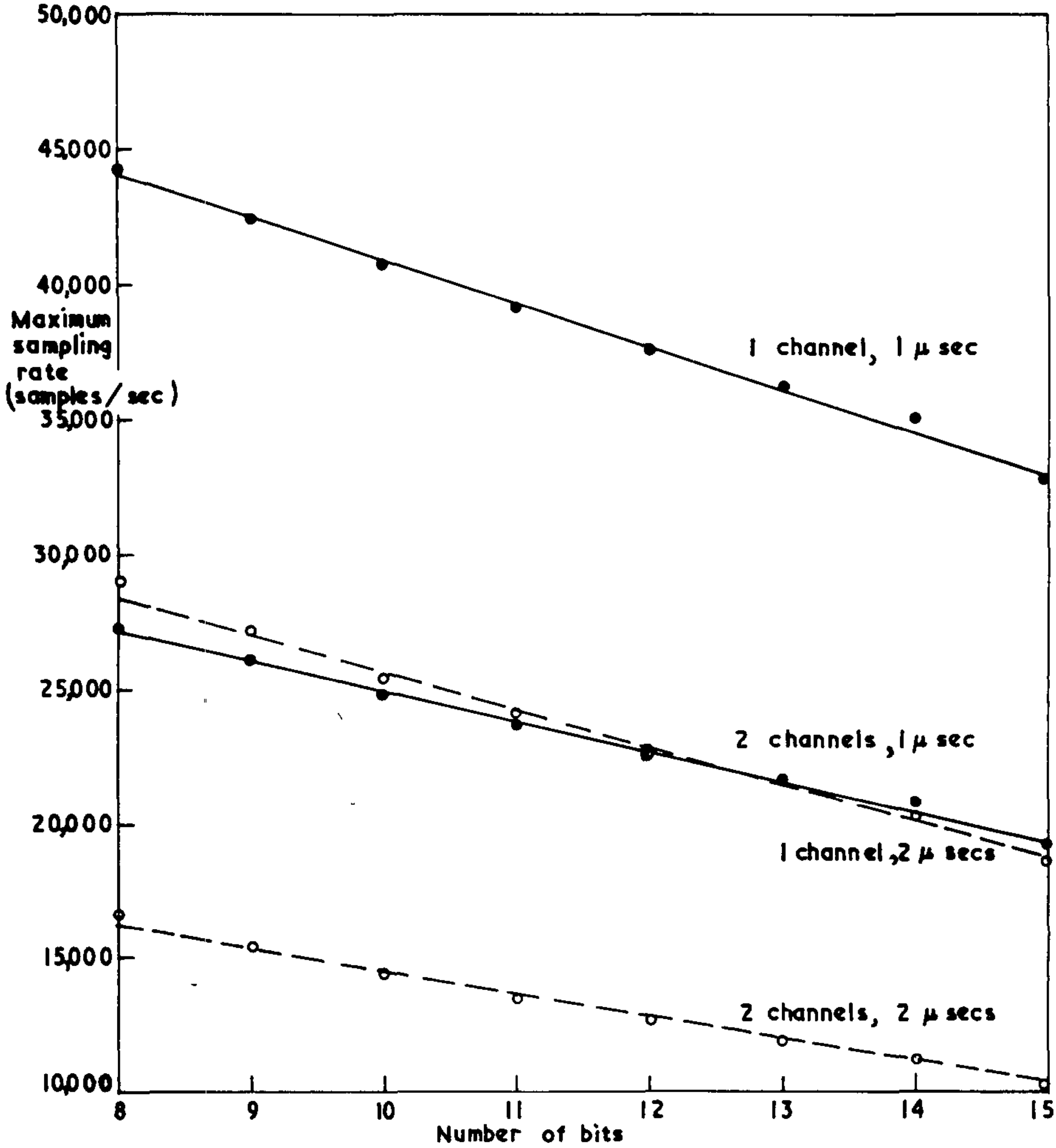
Block diagram of the overall system

FIG. 3.2 (a)

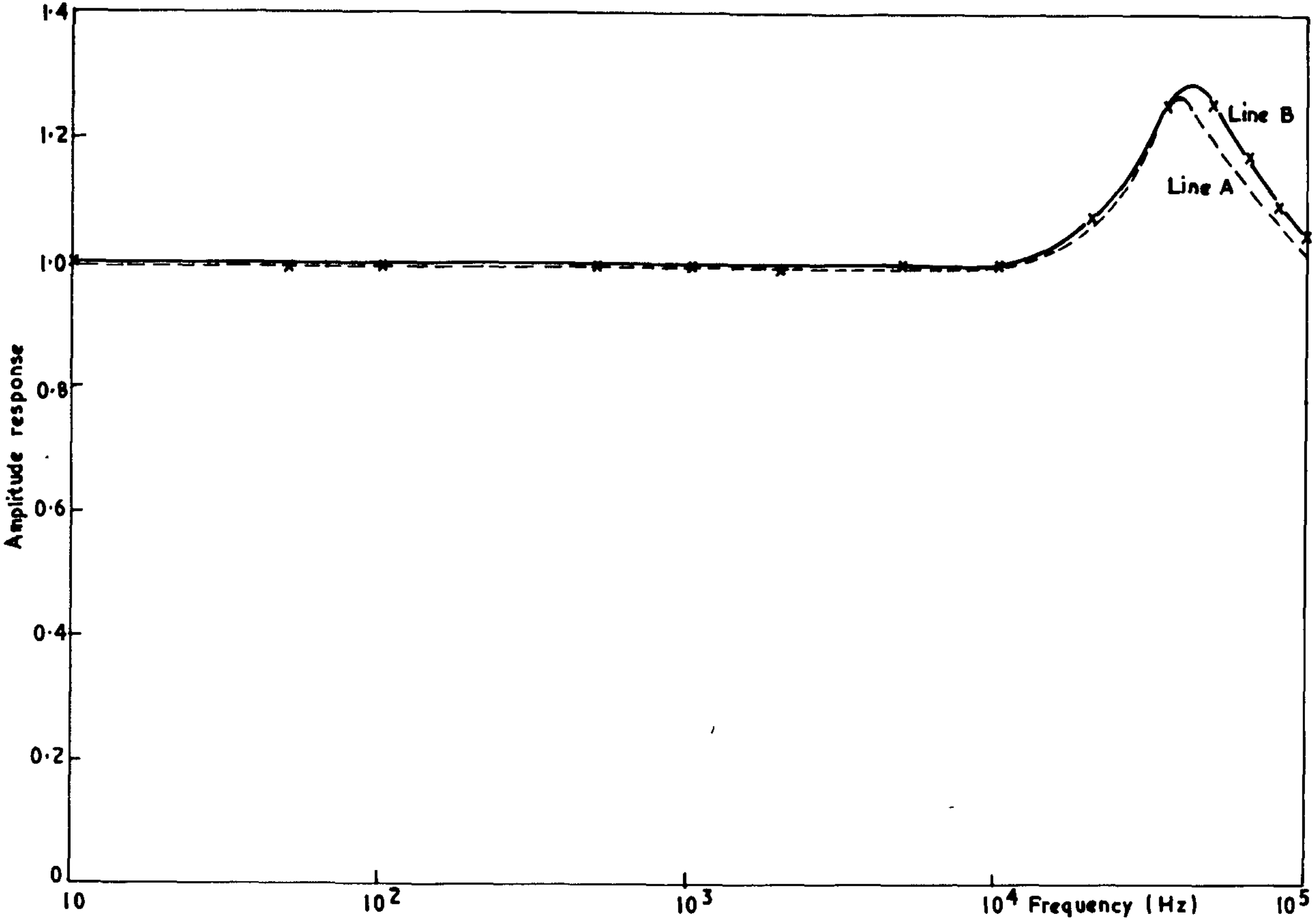


Maximum allowable sampling rates for the A I S under software control

FIG. 3.2 (b)

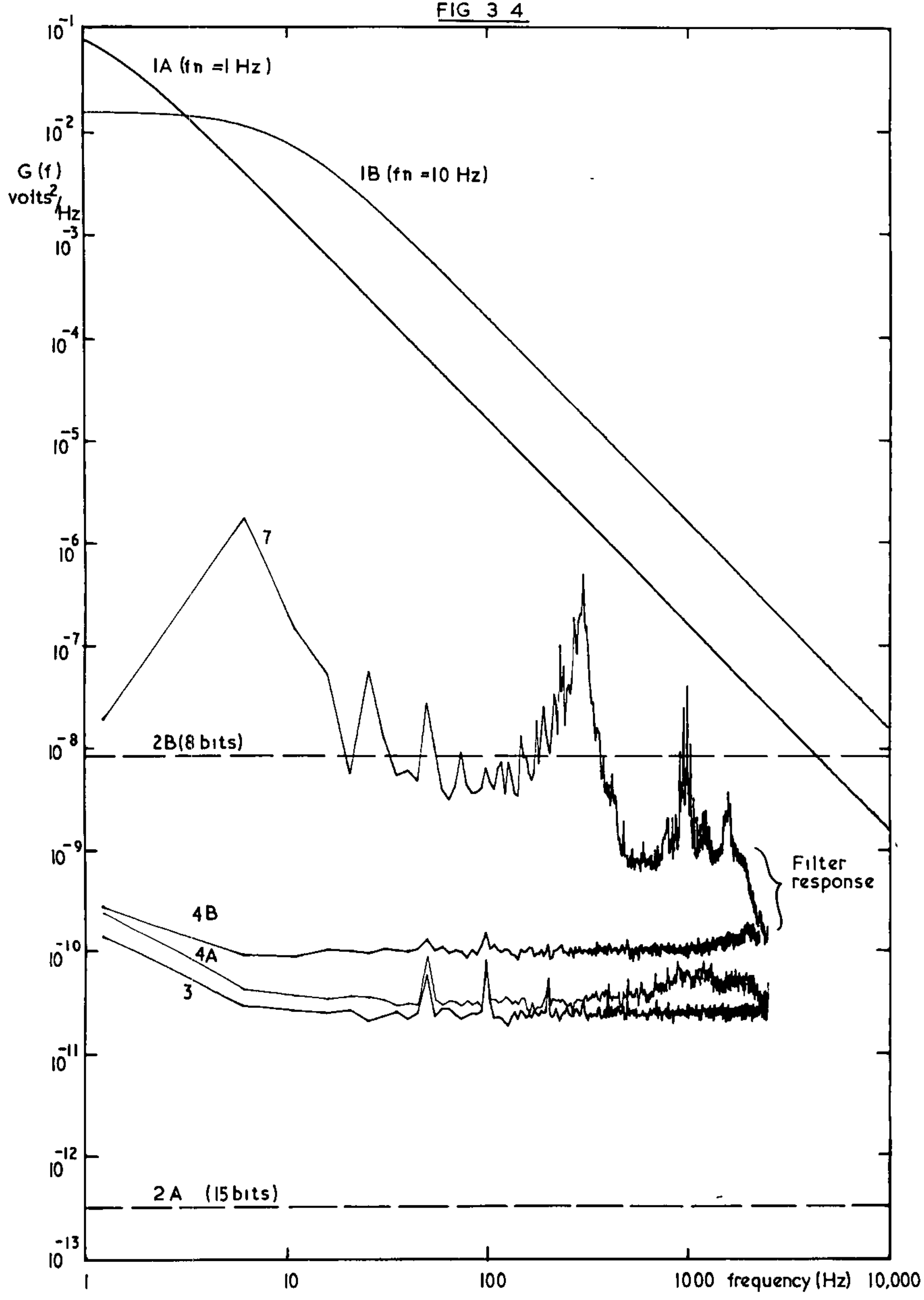


Maximum allowable sampling rates for the AIS under hardware control

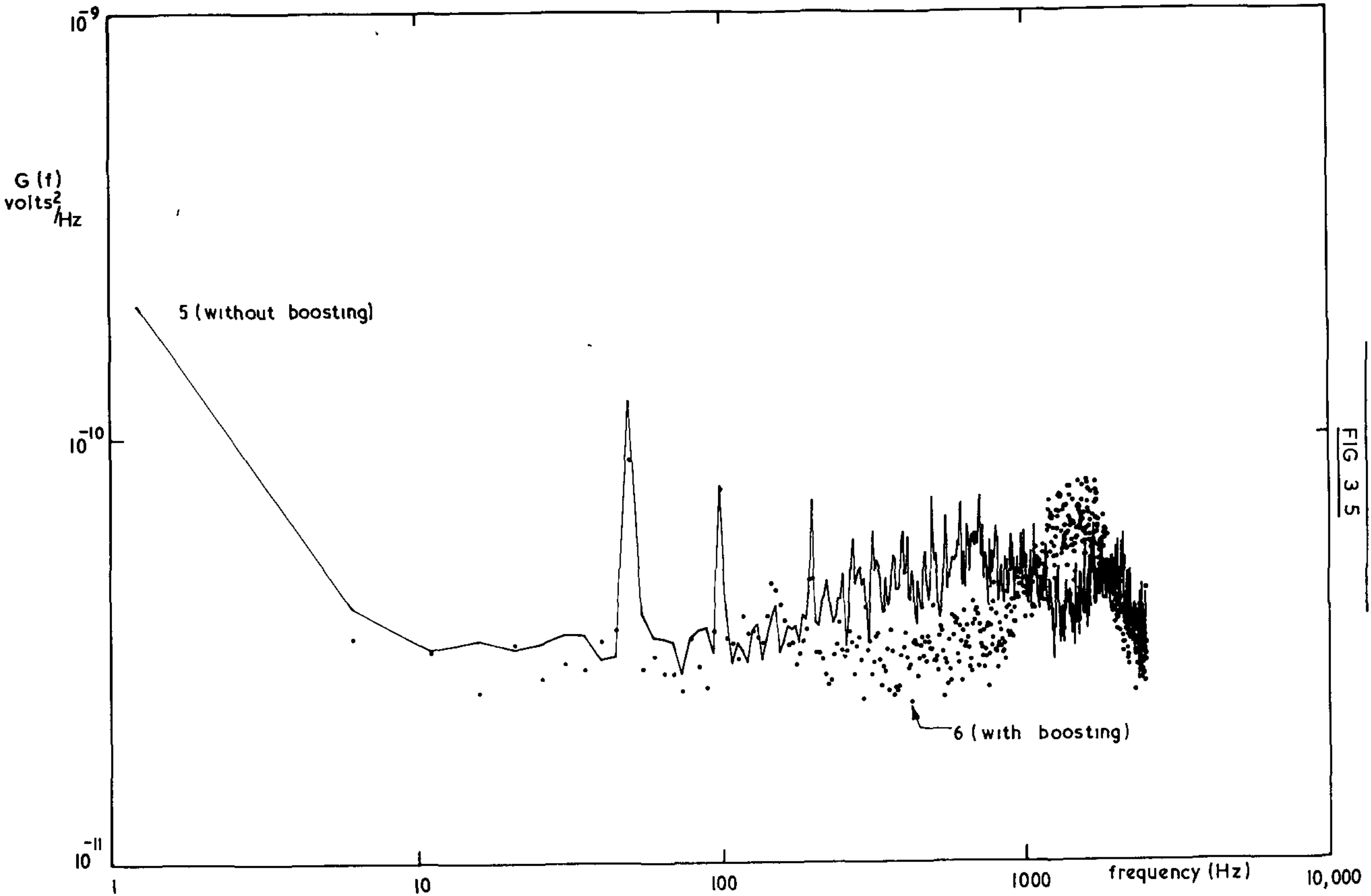


Amplitude response of the lines in use (3' x 3' tunnel to computer)

FIG 3 4

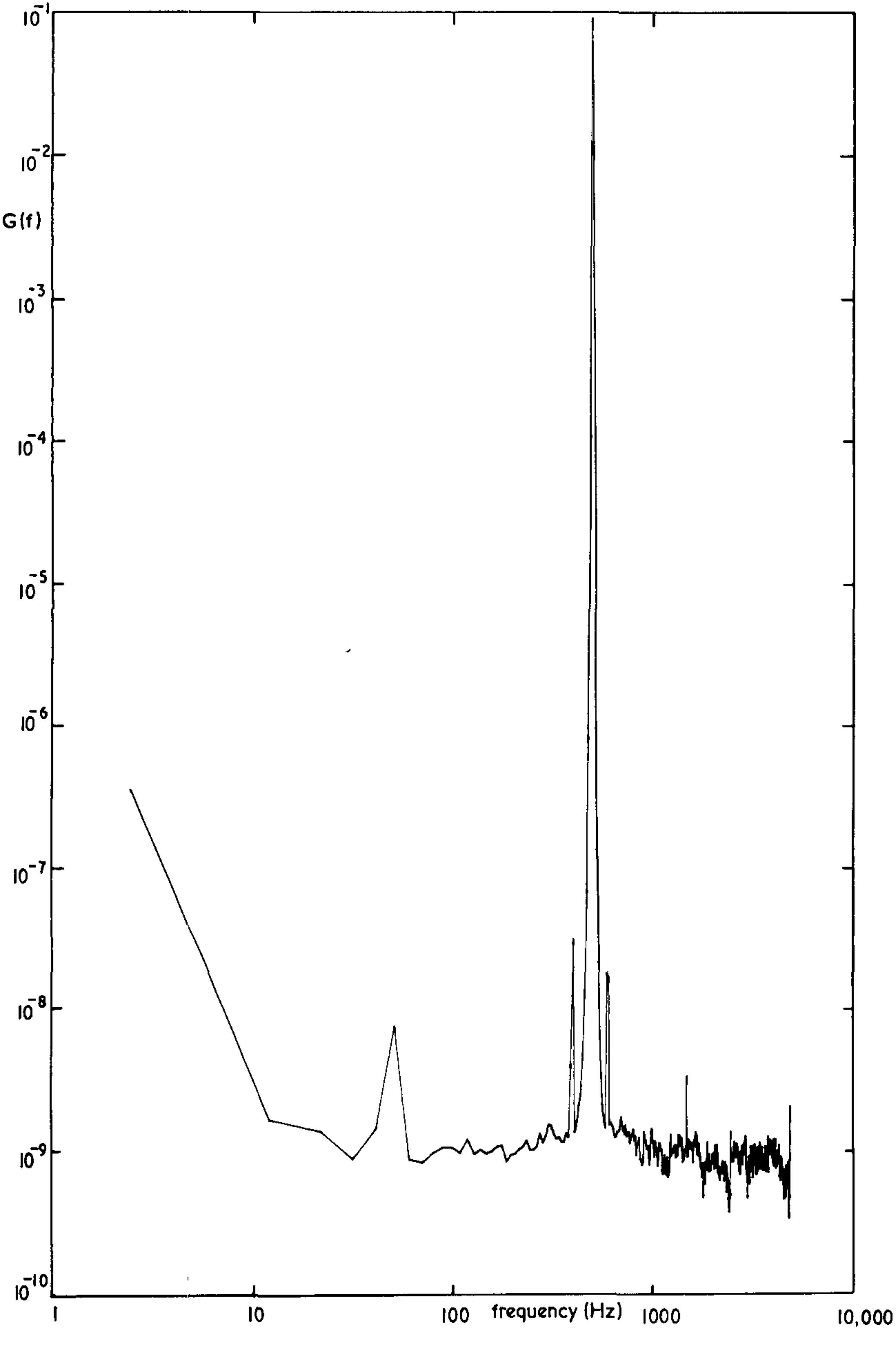


A comparison of expected data levels and system noise spectra under various conditions



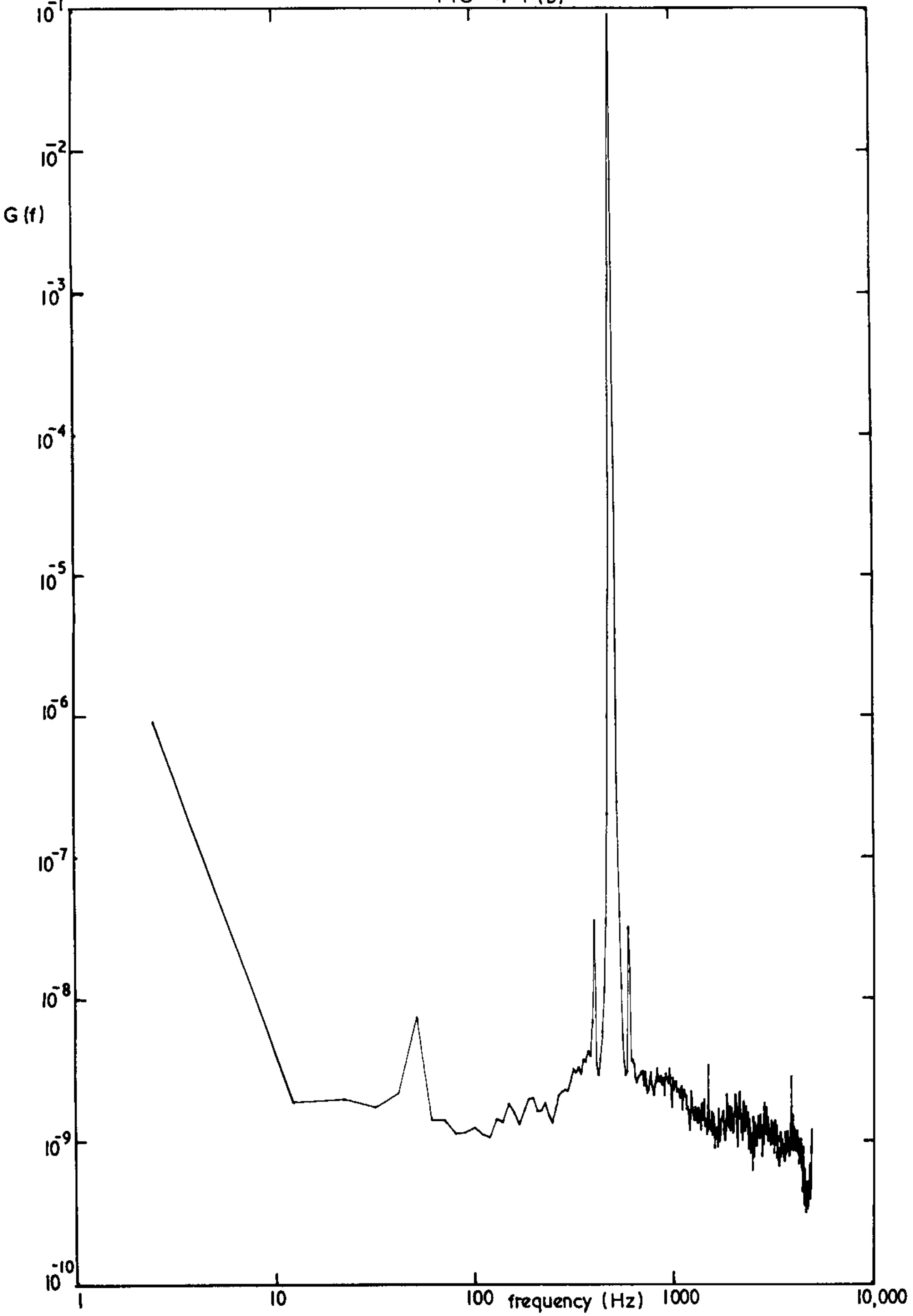
Effect of boosting signal levels on system noise

FIG 4 1 (a)



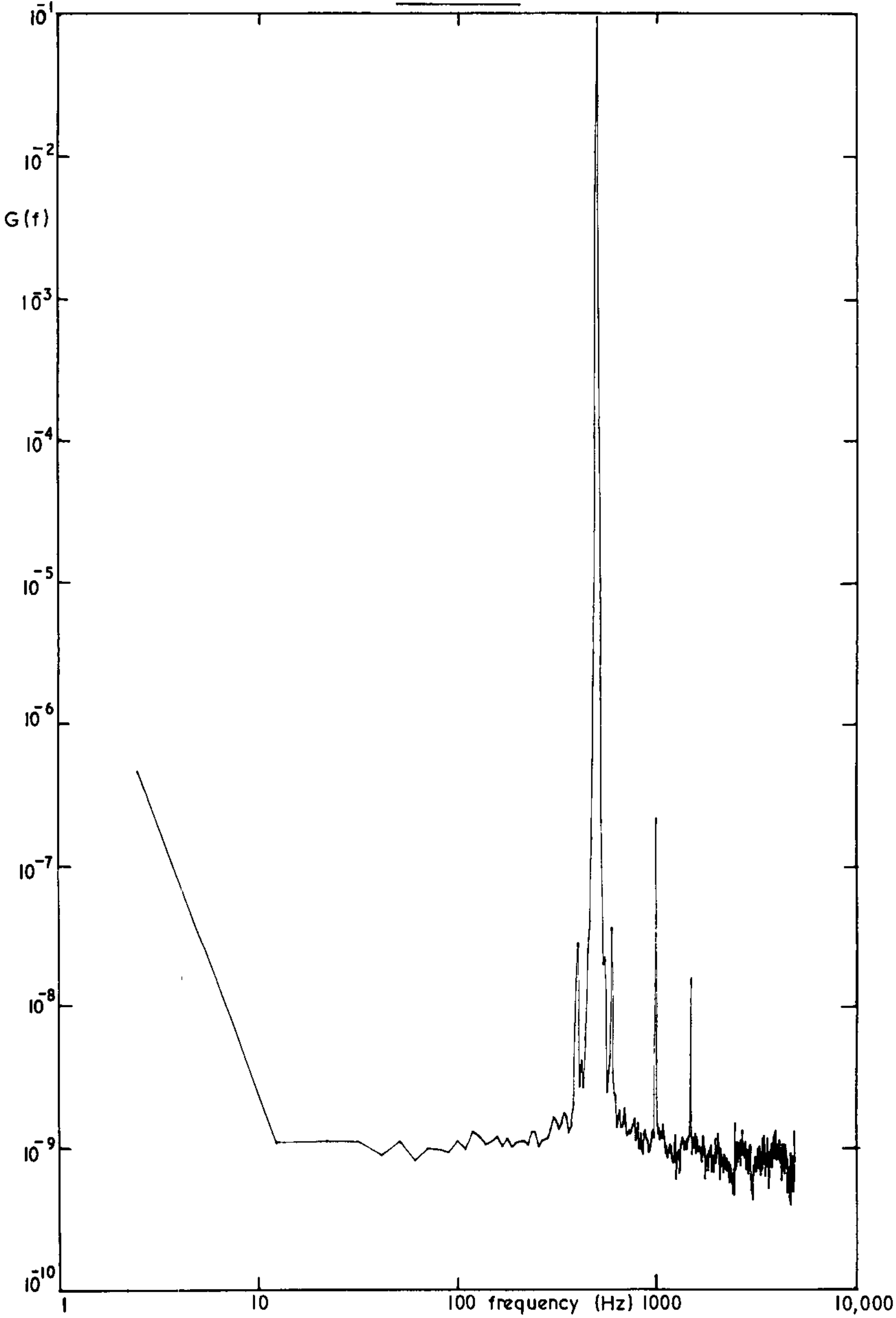
Power spectrum of a 500 Hz sine wave connected directly into the AIS

FIG 4 1 (b)



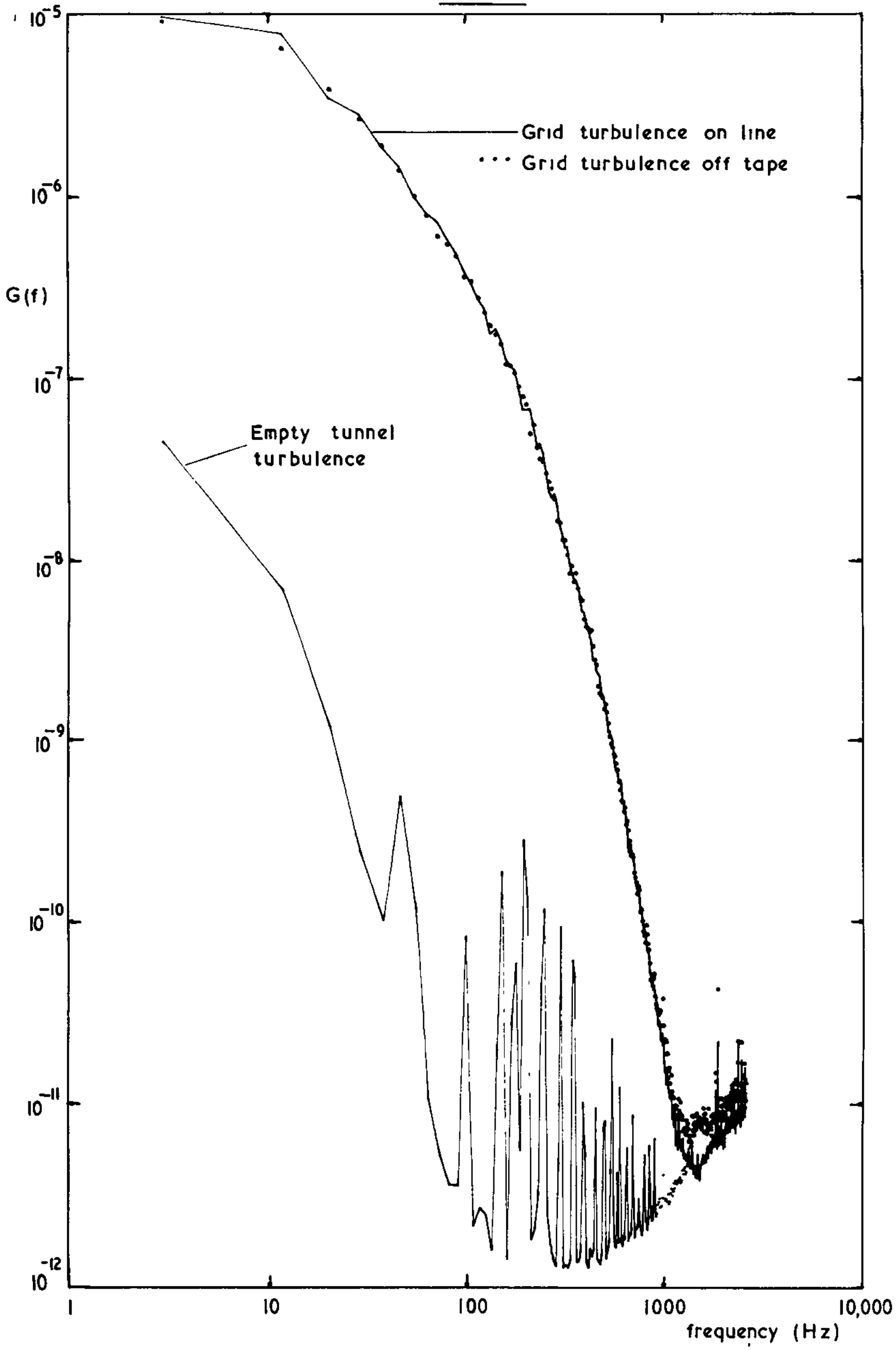
Power spectrum of a 500 Hz sine wave connected to the AIS through the lines only

FIG 4 1(c)



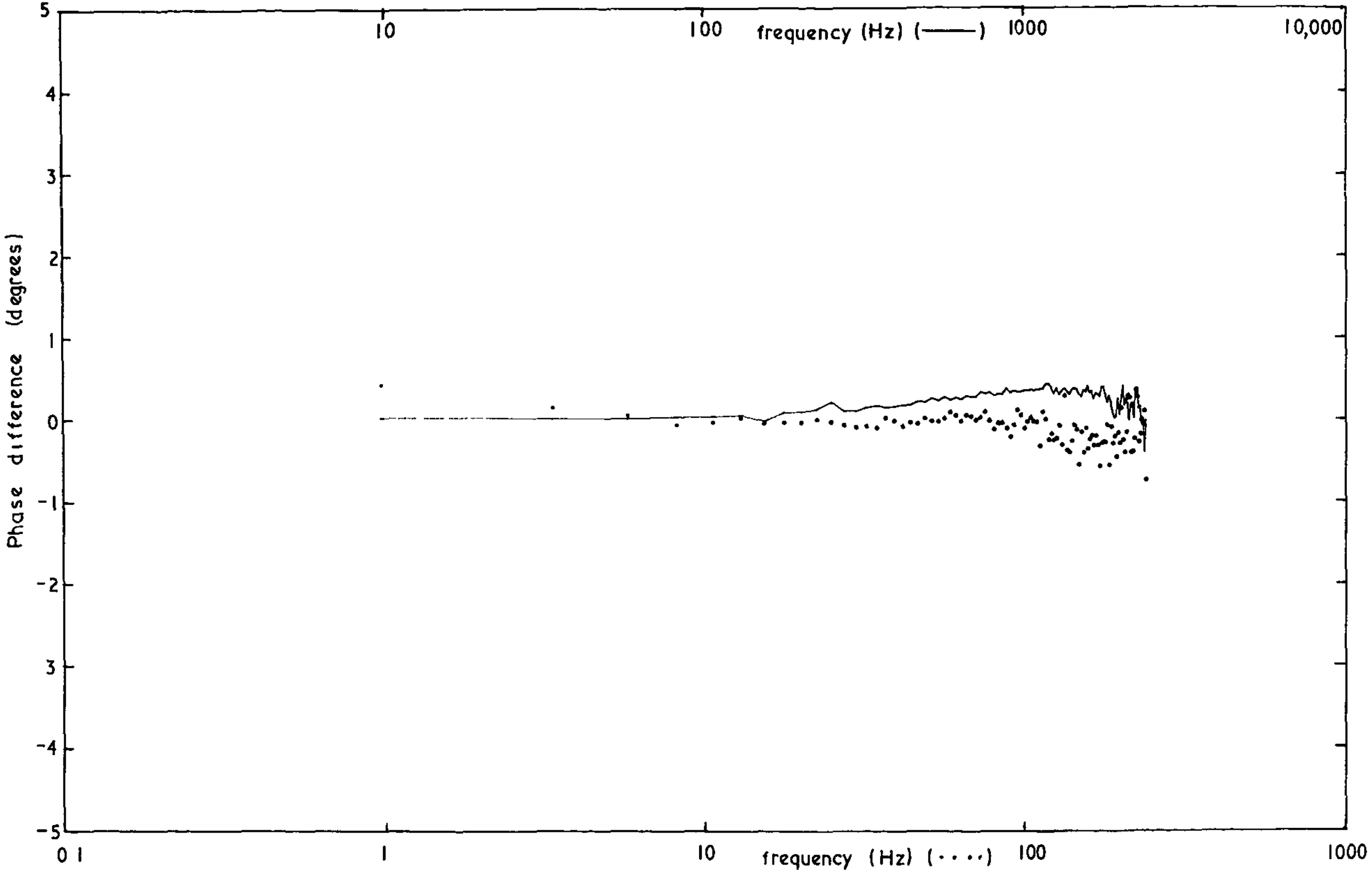
Power spectrum of a 500 Hz sine wave connected to the AIS through the entire system

FIG 4 2

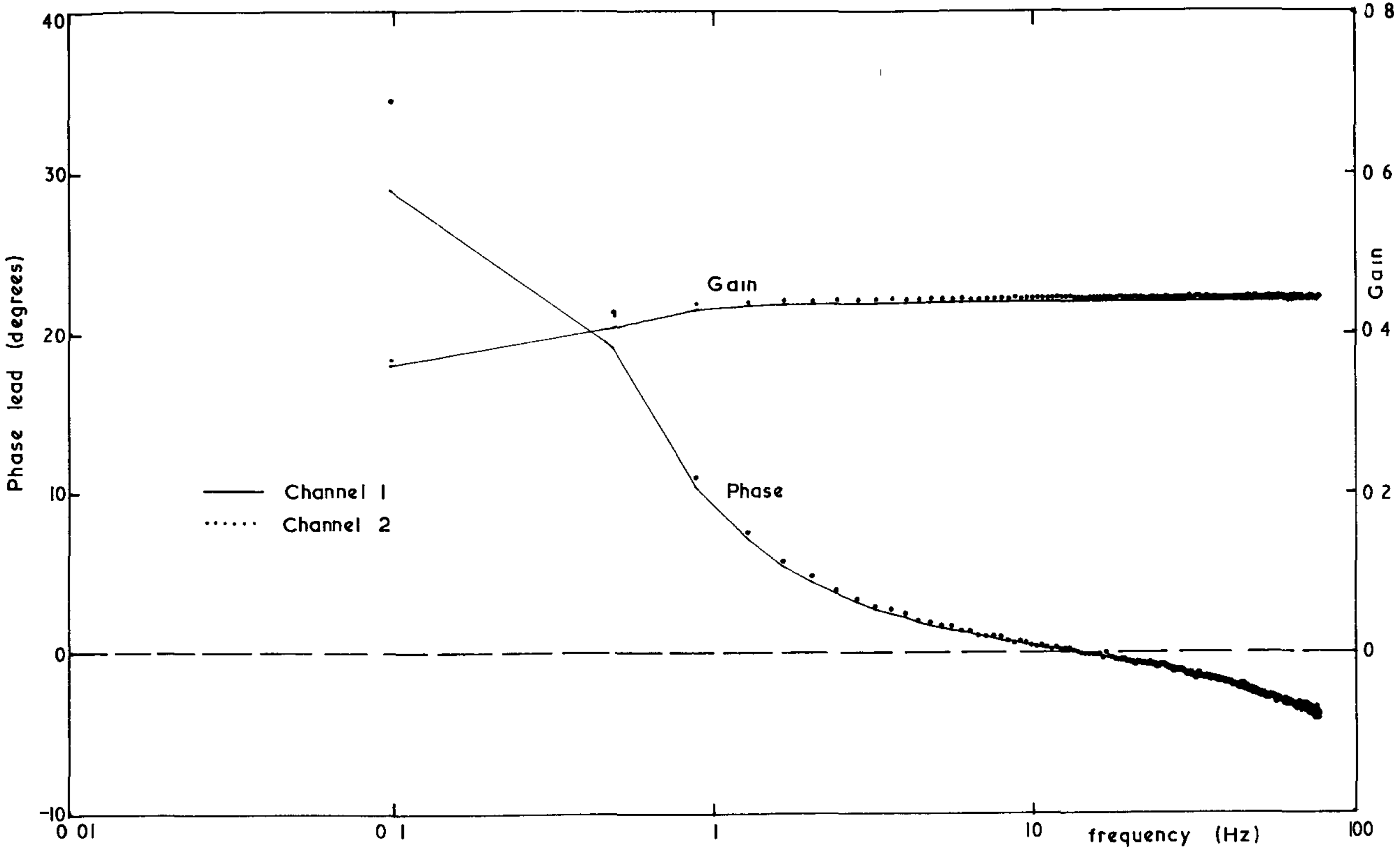


Some typical turbulence spectra

FIG 4 3

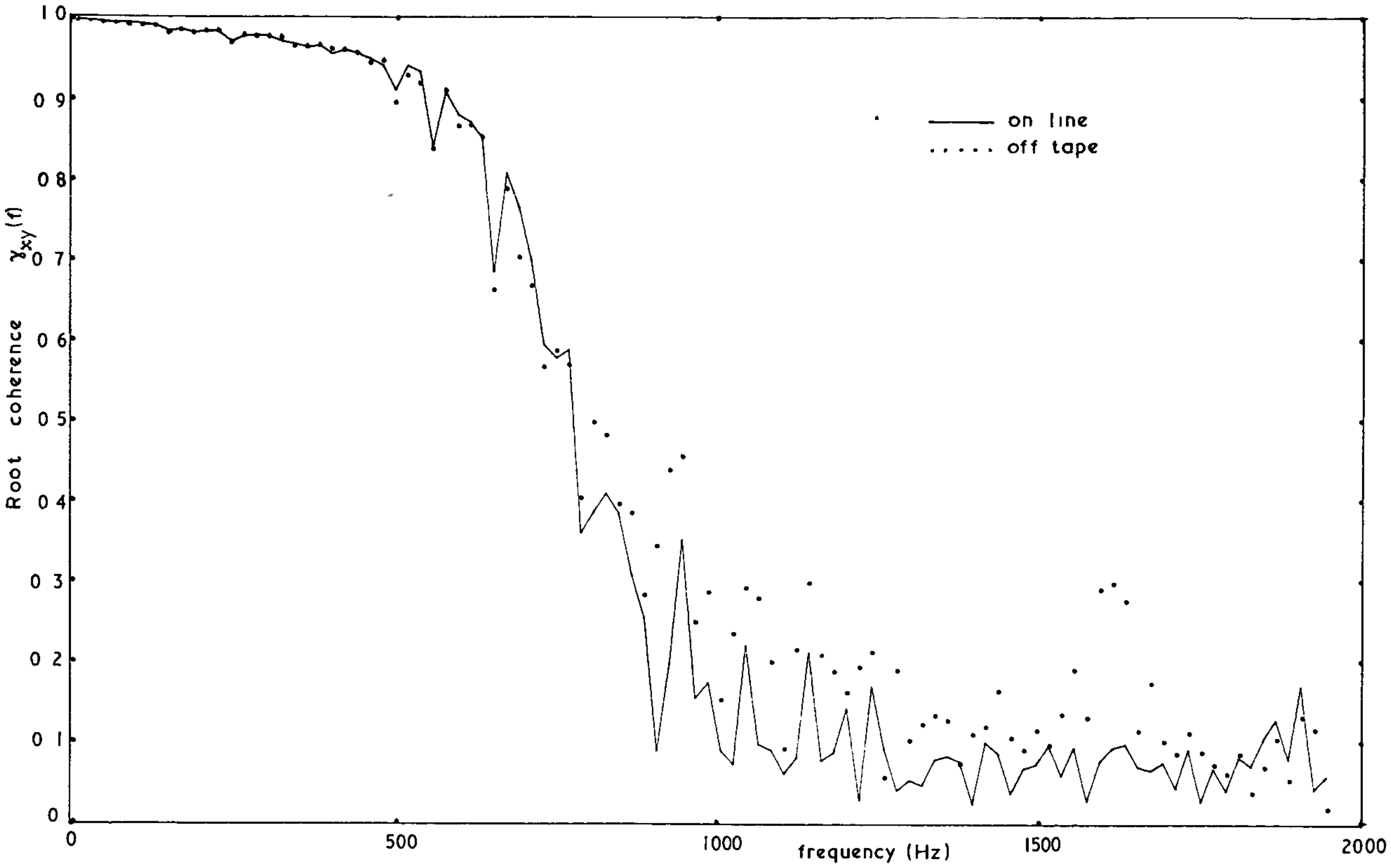


Phase matching of the filter characteristics



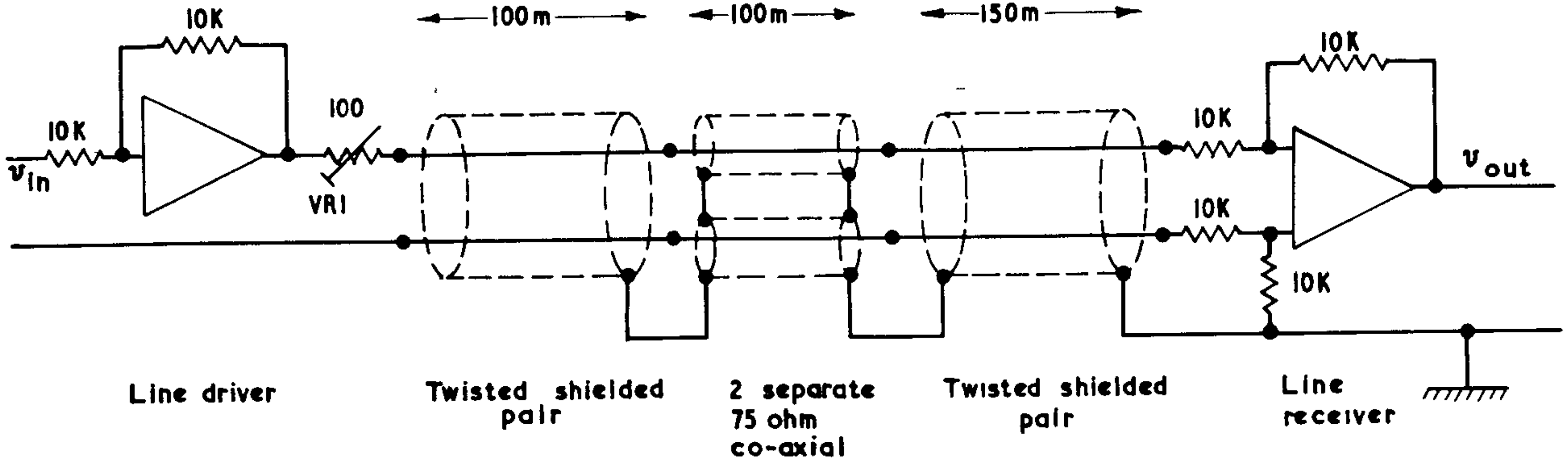
Gain and phase characteristics of the filters

FIG 4 5



Coherence measurements on line and off tape

FIG. III.1



A Typical line system

A.R.C. C.P. No.1225
March, 1971

J. B. Roberts and
D. Surry - with an Appendix by R. F. Johnson

SOME EXPERIENCES WITH "ON-LINE" SPECTRAL ANALYSIS

USING A SMALL DIGITAL COMPUTER

The capabilities of an "on-line" spectral analysis system, developed at NPL, are described. Two approaches to the digital generation of spectral estimates, the correlation technique and the FFT technique, have been adopted and a comprehensive appraisal of these methods is given. A full description of the physical and performance characteristics of the hardware involved is included.

A.R.C. C.P. No.1225
March, 1971

J. B. Roberts and
D. Surry - with an Appendix by R. F. Johnson

SOME EXPERIENCES WITH "ON-LINE" SPECTRAL ANALYSIS

USING A SMALL DIGITAL COMPUTER

The capabilities of an "on-line" spectral analysis system, developed at NPL, are described. Two approaches to the digital generation of spectral estimates, the correlation technique and the FFT technique, have been adopted and a comprehensive appraisal of these methods is given. A full description of the physical and performance characteristics of the hardware involved is included.

A.R.C. C.P. No.1225
March, 1971

J. B. Roberts and
D. Surry - with an Appendix by R. F. Johnson

SOME EXPERIENCES WITH "ON-LINE" SPECTRAL ANALYSIS

USING A SMALL DIGITAL COMPUTER

The capabilities of an "on-line" spectral analysis system, developed at NPL, are described. Two approaches to the digital generation of spectral estimates, the correlation technique and the FFT technique, have been adopted and a comprehensive appraisal of these methods is given. A full description of the physical and performance characteristics of the hardware involved is included.

DETACHABLE ABSTRACT CARDS

R3141/S02109 K4 8/72 P

C.P. No. 1225

HER MAJESTY'S STATIONERY OFFICE

Government Bookshops

49 High Holborn, London WC1V 6HB
13a Castle Street, Edinburgh EH2 3AR
109 St Mary Street, Cardiff CF1 1JW
Brazennose Street, Manchester M60 8AS
50 Fairfax Street, Bristol BS1 3DE
258 Broad Street, Birmingham B1 2HE
80 Chichester Street, Belfast BT1 4JY

Government publications are also available through booksellers

C.P. No. 1225

SBN 11 470783 9