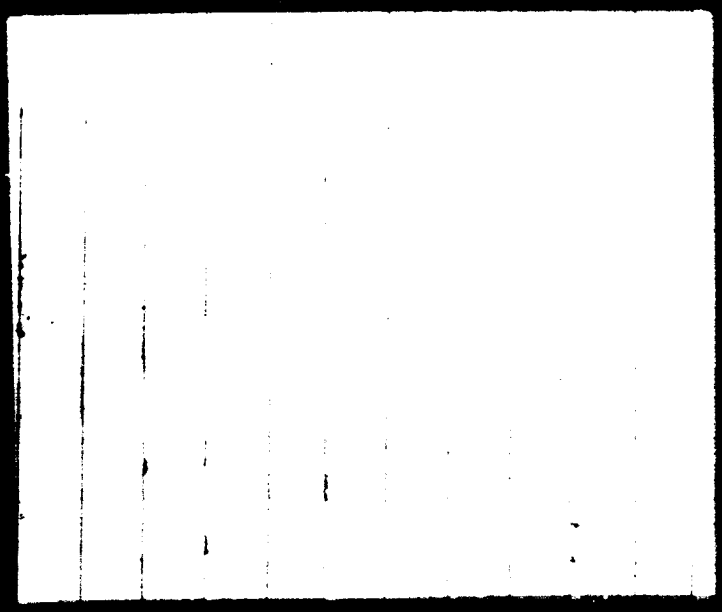

Signal-to-Noise Ratio Estimation in Digital Computer Simulation of Lowpass and Bandpass Systems with Applications to Analog and Digital Communications

by

W. H. Tranter

and

M. D. Turner

FINAL REPORT - VOLUME III

Prepared For

National Aeronautics and Space Administration

Under

Contract No. NAS 9-14848

University of Missouri-Rolla

July 1977

Technical Report CSR-77-6

ABSTRACT

Digital computer simulation has become an important technique for the evaluation of a wide range of systems. A general method, suitable for use in digital computer simulations, is often needed to measure the performance of these systems. A useful criterion for performance evaluation is the mean-square error between waveforms at two points in a system or between the waveform at one point and a theoretical reference. Signal-to-noise ratio, a useful parameter in many cases, can be defined for arbitrary waveforms using measurements of power and mean-square error. Other system parameters, such as gain and delay, can be given general definitions through the use of a minimum mean-square error criterion. Techniques are developed to estimate power, gain, delay, signal-to-noise ratio, and mean-square error in digital computer simulations of low-pass and bandpass systems. The techniques are applied to analog and digital communications. The signal-to-noise ratio estimates are shown to be maximum likelihood estimates in additive white Gaussian noise. The methods are seen to be especially useful for digital communication systems where the mapping from the signal-to-noise ratio to the error probability can be obtained. Simulation results show the techniques developed to be accurate and quite versatile in evaluating the performance of many systems through digital computer simulation.

ACKNOWLEDGEMENTS

The authors wish to express their sincere appreciation to the National Aeronautics and Space Administration - Lyndon B. Johnson Space Center for their support of this work. In particular, Mr. C. T. Dawson and Mr. T. W. Eggleston have contributed significantly to this effort through the many enlightening discussions which have been held at NASA - JSC.

TABLE OF CONTENTS

LIST OF FIGURES

FIG. NO. PAGE

LIST OF TABLES

I. INTRODUCTION

Digital computer simulation is becoming an increasingly important tool in the analysis of many systems. In order to evaluate systems through the use of computer simulation, a general quantitative measurement of performance that can be applied to a broad range of systems is needed. In many cases, an appropriate measurement of performance is the mean-square error between waveforms which appear at different points within a system, or between the waveform at one point and some theoretical reference waveform. In cases where the mean-square error itself is not an appropriate figure of merit, other useful measures of performance may often be derived from the mean-square error.

One quantity commonly desired in evaluating many systems is the signal-to-noise ratio. By defining noise as the mean-square error, it is possible to use measurements of power and mean-square error to obtain signal-to-noise ratios for arbitrary waveforms. This procedure is complicated by the fact that many systems contain unknown gain and delays which need to be estimated in order to calculate the mean-square error. These parameters, also, can be defined through determination of estimates which minimize the mean-square error or maximize the signal-to-noise ratio. One area in which measurements of power, gain, delay, and the signal-to-noise ratio are particularly needed is in the evaluation

of analog and digital communication systems through digital simulation.

The widespread use of complex digital communication systems, and the advancement of computer technology in recent years has resulted in the increased use of digital computer simulation for predicting the performance of digital communication systems. Monte Carlo techniques have been applied to a wide variety of systems. Observations of the simulated system under various operating conditions are used to predict the performance of the actual system.

The most common method for evaluating the performance of a digital communication system through computer simulation is to determine the symbol error probability by direct error counting. However, this technique has the disadvantage of requiring prohibitively large amounts of computing time to measure typically encountered error probabilities. For example, on the average, to produce ten errors at an error probability of 10^{-6} requires that 10^{7} information symbols be simulated. Extremal statistics and related techniques can generally extend the practical range of the error counting method by approximately an order of magnitude. However, estimation of error probabilities below about 10^{-4} or 10^{-5} still require too much computer time to be economically feasible in most cases.

When the mapping from the signal-to-noise ratio to the error probability is either analytically known or can be experimentally determined, an alternative approach is

to estimate the signal-to-noise ratio and map this to the symbol error probability. In the work which follows, a method is developed to accurately measure the signal-to-noise ratio in a digital computer simulation of a communication system. The method is quite general and may be applied to a wide variety of systems. Both digital and analog systems can be evaluated. The use of direct simulation is not restricted to systems with lowpass signals. Simulation of a bandpass system may utilize either conversion to an equivalent baseband system, in which lowpass signals are processed, or a direct simulation of the system, which retains signals having bandpass spectra. The latter approach is commonly referred to as direct RF simulation, even when the signals are not radio frequency.

In the work which follows, computer routines are developed for implementing signal-to-noise ratio estimators for the lowpass case and for the bandpass case where direct RF simulation of a bandpass system has been utilized. The lowpass estimator forms a minimum mean-square error estimate of the signal, allowing for any unknown amplitude scaling and time delay within the system. In the bandpass case, both group delay and phase delay of the signal may be arbitrary. The FFT provides a practical method for estimating delay in the lowpass case. For direct simulation of bandpass systems the FFT is essential for independently determining group and phase delay to form the minimum mean-square signal estimate. An option is provided for applying a data window to smooth the estimates if desired.

To evaluate the accuracy, reliability, and versatility of the signal-to-noise ratio estimation routines, a variety of tests are performed. First simple filtered noise tests are used to verify the accuracy and determine the useful range of the estimators. Then digital computer simulation and performance evaluations are conducted for a number of systems including FM systems with phase-locked loop demodulation, coherent ASK and FSK, noncoherent FSK, and Rayleigh fading channels, including diversity transmission. The simulations incorporate additive Gaussian noise, and signal-to-noise ratio estimates are obtained to beyond 50 dB in some cases using simulations of approximately 10^3 information symbols. The simulations were obtained using the IBM 370 computers at the University of Missouri and a Univac 1110 computer at NASA-Johnson Space Center in Houston, Texas.

Results obtained from the estimators are excellent. In all the various systems which were simulated, accurate results were obtained over the ranges of the signal-to-noise ratio that were considered. Of course, the necessity of knowing the mapping from the signal-to-noise ratio to the symbol error probability is sometimes a handicap in dealing with digital systems. However, in many cases this mapping can be determined using analysis or approximately determined using bounding techniques. Sometimes, however, the signal-to-noise ratio is a satisfactory figure of merit, and determination of the symbol error probability is unnecessary.

The methods developed appear to be quite versatile and promise to provide a very useful tool for evaluating the performance of a wide variety of practical systems through the use of computer simulation.

II. REVIEW OF SIMULATION TECHNIQUES

Computer simulation of communication systems is becoming a popular technique in the study of systems which are too complicated for an exact mathematical analysis. Simulation offers an economical alternative to the fabrication and testing of all the variations of the system being studied. In order to take advantage of these computer methods, several schemes have been developed for evaluating the performance of systems through the use of computer simulation.

Most communication systems require some modulation technique which generates a bandpass signaling scheme. The most direct method of accurately simulating such systems is to generate simulations based upon these bandpass signals. This method has been used successfully by several investigators [1] - [6].* An alternative approach is to mathematically convert the bandpass signals to equivalent lowpass (baseband) signals. The baseband technique avoids some problems encountered in directly simulating the bandpass systems and often lends itself to simpler analysis. For these reasons, the baseband methods have found somewhat wider acceptance than direct RF simulation of bandpass systems and appear more often in the literature [7] - [12].

*All numbers shown in brackets refer to corresponding numbers in the list of references.

One advantage of baseband simulation is that often less samples need to be used for a given number of information symbols. This results from the fact that the highest frequencies being sampled are those in lowpass signals. For direct RF simulation of the bandpass signals, the sampling frequency must be sufficiently high to include the carrier frequency and usually an upper sideband of modulation. The number of samples necessary for a given simulation time interval is generally at least five to ten times less with the baseband method. Another advantage is that the simulation models are sometimes easier to program. For example, a bank of filters written for use in an exclusively baseband simulation may often include only lowpass filters. In the direct RF simulations, the usually more complex bandpass filters are necessary. Problems such as adjacent channel interference can be eliminated in baseband methods, where no adjacent frequency channels exist. This can be an advantage if such an idealized system is to be simulated.

More often, however, this inability to directly account for interference is a disadvantage. Many times in an FSK system or in a frequency multiplexed system, the problem of interference between adjacent frequency channels is an important effect which must be included in the simulation. The direct RF simulation can simply model the system as it actually exists, simulating all channels simultaneously at the appropriate carrier frequencies. The baseband simulation is not capable of this approach, so more elaborate techniques are required.

One simulation method used by Hedderly and Lundquist [13] to account for the interference effects of frequency diversity is to compute the spectrum of power from adjacent channels which would fall into the channel of interest and then to increase the additive channel noise to include the additional "noise" power created by interference. Another technique used by Bello and Ehrman [14] involves the use of short tone bursts of the desired frequency channel adjacent in time to tone bursts of two interfering frequencies. The channel is used in rapid time diversity to account for the intersymbol interference effects in an FSK system. These methods have been applied in RF simulations with some success, but the direct modeling of all the frequency channels is certainly more intuitively appealing.

Another disadvantage of the baseband technique is that it is usually necessary to model inherently bandpass systems using a complex envelope representation. Hence, either two real number simulation paths or a complex number simulation must be used to represent the general system.

In the work which follows, the direct RF simulation approach for modeling bandpass systems was most often used. It was felt that the baseband simulations often idealized and oversimplified some of the various effects which could not be directly modeled. Also, intuitive insight into a system seems to be more easily applied with direct RF modeling. As explained later, the accuracy of the method chosen to evaluate the performance of the systems being simulated is a function of the total number of samples used and not

of the number of modulation symbols involved. Hence the increased number of samples required by the direct RF simulation technique to simulate a given number of modulation symbols compared to baseband methods is not a serious problem.

Once a simulation, baseband or RF, has been completed and data collected, some method must be used to evaluate the system performance. In certain limited applications, only qualitative information is required. Simply plotting output waveforms and inspecting them visually for distortion is sometimes acceptable in these cases [15]. However, in most cases a quantitative measurement of system performance is required. For digital communication systems, the basic measure of performance is usually taken to be the probability that a given information symbol is in error. Several approaches have been applied to directly or indirectly estimate this quantity. The most straightforward method is simply to perform the simulation, count the errors, and utilize the law of large numbers to directly estimate the probability of error. Other methods use extremal statistics and related techniques to estimate the probability of error. Still another possibility is to measure some other quantity such as the signal-to-noise ratio and estimate the probability of error indirectly.

By far the most often used technique in the estimation of P_E (the probability of symbol error) for digital communication systems has been direct error counting. This

approach has been used by many researchers and is reported in the literature [16] - [18]. This approach works very well where it is practical. However, actual systems typically display error probabilities ranging from 10^{-5} to 10^{-7} or even less. To obtain a reliable estimate of the error probability, it is desirable to utilize at least enough data to include ten errors. For a simulation of a system exhibiting an error probability of 10^{-6}, this would require 10^{7} symbols to be processed. This is completely impractical from an economic point of view. For example, Leon and Kitahara [19] report that for a particular QPSK (quadrature phase shift keyed) simulation performed on a CDC 6500 computer, the expected cost of generating a single error with an error probability of 10^{-6} would be approximately $4000. They obtained P_E estimates down to about 10^{-3} with direct counting methods. Similar results were reported by Leon and others [20] for PSK (phase shift keyed) data links. Bello and Crystal [21] were able to use direct error counts for P_E to about 10^{-4}. Their system was one which was difficult to analyze for low signal-to-noise ratios. For high signal-to-noise ratios, analysis was mathematically tractable and bounds were developed for P_E below 10^{-4}. Joining the bounds with the computer simulation results yielded curves over a wide range of P_E. Because of the large amounts of computer time required, the direct error count method seems to be impractical below error probabilities of about 10^{-3} in most cases, or 10^{-4} in the most

efficient simulations. This is not adequate for most applications.

In an effort to estimate low error probabilities using less data, several researchers have applied extremal statistics to the problem [22] - [26]. In this technique an exponential function is used to approximate the tail of the density function for the errors by adjusting parameters in a general exponential form. The parameters are computed using the statistics of the maxima (extremes) of groups of independent samples collected from the simulation [27]. Smaller values of P_E are then computed from this easily integrable exponential approximation to the density function. This general technique may be applied to a wide variety of density functions which behave as a decaying exponential on the tails. Specific tests are available to determine whether a set of data points do in fact satisfy this type of behavior [28]. One disadvantage of the extremal statistics method is that it requires a learning period when the signal is known so that the parameters may be computed as a function of only the unknown noise distribution. A method to eliminate the learning period and estimate parameters based upon noisy samples has been proposed by Milstein [29]. Schwartz [30] and other investigators have concluded that extremal statistics may be used to reduce by an order of magnitude in many cases the data that are required for satisfactory P_E estimates. Attempts to extend these methods beyond about two orders of

magnitude often result in P_E estimates which are in error by almost an order of magnitude, however. It appears that extremal statistics may extend the capabilities of the error counting techniques from a typical 10^{-3} limit to about 10^{-4}, or at best 10^{-5}. This may be sufficient in some applications, but it is still far short of the 10^{-6} or 10^{-7} range which is often desired.

Another variation of error counting has been tried with some success by a number of researchers. This method, described by Gooding [31], is based upon measuring what is termed a "pseudoerror" probability and mapping this quantity to P_E. To implement this method, thresholds are established for the demodulated signals other than the usual decision thresholds used to decode the received signal. These new thresholds define a region similar to that which corresponds to an erasure in the classical binary erasure channel. All demodulated symbols which fall into this highly uncertain region within the new thresholds are termed pseudoerrors. The symbols may still be decoded as usual; there is no need to actually erase them. By monitoring these pseudoerrors and applying knowledge of the distribution function which generates them, P_E may be estimated. A significant advantage of this technique is that it requires no knowledge of the input data stream. The thresholds are generally established to be symmetric with respect to the ideal signal points in decision space, so that the pseudoerror probability is independent of the stream of input data and even of

its statistics. This method does, however, require that the distribution function for the noise be known in order that proper thresholds may be established and the mapping to the error probability computed. The method has been applied to PSK and QPSK systems [32] - [34]. It appears to yield results slightly more reliable than those obtained by extremal statistics methods. However, it should be remembered that the extremal statistics approach does not require that the distribution function be precisely known. The error probability estimates obtained using the pseudoerror technique are usually accurate to within a factor of three for values of P_E as small as two orders of magnitude less than those which are practical with direct error counting. This method is not usually capable of measuring P_E much below 10^{-5} in computer simulation work, and Leon and Kitahara [35] note that this technique is not economically feasible for what they term practical 10^{-7} error probabilities.

Another similar technique has been suggested by Weinstein [36], which involves counting the decoded symbols which fall above some threshold. This method also is reported to be capable of reliably estimating P_E down to values between one and two orders of magnitude smaller than is possible with a direct error count. The technique is based on linearizing the tail of the distribution function of the noise with a log-log transformation.

An indirect method for estimating the error probability is to first estimate the signal-to-noise ratio and map this

into P_E. This method has the disadvantage that the distribution function for the noise must be known, or somehow the mapping from the signal-to-noise ratio to the probability of error must be obtained. The estimation of P_E through estimation of the signal-to-noise ratio has one striking advantage over all the other techniques mentioned. For any signal-to-noise ratio that can be reliably estimated where the mapping to P_E is known, reliable estimates for the error probability can be obtained. That is, there is no reason why P_E cannot be measured for any range of interest. For most digital systems, if signal-to-noise ratios as high as 30 or 40 dB can be measured, probability of error estimates are possible far below the typical 10^{-7} range.

A method used for estimating the signal-to-noise ratio in order to monitor the error rate of a PCM system is given by Gagliardi and Thomas [37]. The method presented is only applicable to a system or a simulation with no unknown delay, however. No implementation is suggested for use with a computer simulation. The decoded data stream at the output is assumed to be correct in order to serve as a reference. If the method is to be applied exclusively to systems with very low error probabilities, this might be acceptable. It would seem more general and more accurate to use a known input data stream if possible. This does, however, introduce the problem of finding an unknown delay between input and output, a problem not treated in [38].

Another method for signal-to-noise ratio estimation is suggested by Nahi and Gagliardi [39]. For the case where both signal and noise are Gaussian processes with non-identical correlation functions known to within a constant, the signal-to-noise ratio may be estimated using a hard-limiter. The method is based on known properties of the correlation function of the output of a hard-limiter. This estimator is clearly very specialized, and its reliability has not been analyzed. It is not a suitable estimator in general and is mentioned only for completeness.

The signal-to-noise ratio estimation technique for finding P_E would seem to be the most promising method for small error probabilities, if some method is available for accurately measuring the signal-to-noise ratio within a digital simulation. Indeed for P_E below 10^{-5}, it seems to be the only practical method, since all other techniques are not economically feasible because of the large amounts of computer time they require. It has been suggested [40] that no single technique for P_E estimation should be intended for a very wide dynamic range of error probabilities and that different methods should be applied on different ranges. Whether this is true or not, the missing range for P_E estimation through simulation seems to be exactly the range most needed for computer simulation of practical systems, the range where the error probability is below 10^{-5}.

The method used in this work for predicting the performance of digital communication systems through computer

simulation is estimation of the signal-to-noise ratio, which is mapped to the error probability. Thus the mapping from the signal-to-noise ratio to P_E must be known, or at least approximated. Also, a representation of the signal must be known to serve as a reference. However, it is assumed that the system may incorporate any unknown amplitude scaling and any unknown linear phase characteristic in processing the signal, so that the exact form of the signal at the point of measurement may be quite different from the form of the reference. None of the other techniques described above require knowledge of both the signal form and the mapping from the signal-to-noise ratio to P_E. However, each method requires that either the signal or the noise distribution function be known. The big advantage of the signal-to-noise ratio method is that practical systems operating with practical error probabilities can be analyzed through computer simulations. No other technique is capable of this. And it is worthy of note that in this process for estimating P_E, an estimate for the signal-to-noise ratio is always obtained. The signal-to-noise ratio itself is often a useful parameter in evaluating communication systems without reference to an error probability.

III. DESCRIPTION OF THE PROBLEM

If the SNR (signal-to-noise ratio) is to be adopted as a parameter for evaluating the performance of all systems to be analyzed, a general definition for this quantity must be developed. For most practical communication systems, the amplitude response and phase characteristic of the system are not precisely known. Therefore, at any arbitrary point in a system where an estimate for the SNR is desired, it is often difficult to completely specify the signal. **Thus, before the SNR can be estimated, a rigorous definition for the signal must be devised.**

It is usually possible to provide an input to a system so that the signal component of the waveform at the test point is known except for changes imposed within the system by an unknown scale factor and linear phase characteristic. A linear system is usually considered to be distortionless if it imposes only amplitude scaling and a time delay upon an input signal [41]. It seems appropriate then to allow for any arbitrary scale factor and delay to be present in the signal component at the point where the SNR is to be estimated. For lowpass systems, this signal model is adequate.

In such a case, Shepertycki [42] has suggested an error measurement scheme for telemetry systems. In his method of error measurement, the mean-square error is computed between the waveform to be analyzed and a reference

signal. The amplitude scaling and time delay of the reference are adjusted to minimize the mean-square error, and this error is used as a figure of merit for the system. Simpson and Houts [43] have proposed that the delay value which minimizes the mean-square error be defined as the average time delay for a linear system. Later, the same investigators [44] suggested that the minimum mean-square error technique can be used in the analysis of waveform distortion in linear systems. It is also suggested that random noise could be included in this analysis.

When a bandpass system is being analyzed using direct RF simulation, both group delay and phase delay are involved. A more general definition for the signal component is necessary for this more complicated case. In bandpass systems involving modulation, the information is carried in the complex envelope of the signal. To avoid corruption of the information, it is sufficient that this envelope remain undistorted; delay of the carrier itself is not important in preserving the information content of the signal. It is possible to model phase delay as a time delay of the carrier, and group delay as a time delay of the complex envelope. Therefore, a definition of the signal in this case should allow for any arbitrary scale factor, delay of the envelope (group delay), and delay of the carrier (phase delay) to be present at the point of measurement.

In keeping with this reasoning, a definition was developed for the signal component of any arbitrary bandpass

waveform to be analyzed. It is assumed that a reference signal is available which differs from the signal component only by an unknown amplitude scale factor and unknown group and phase delays. For lowpass systems the group and phase delays are equal, and only the scale factor and a simple time shift are unknown. To define the signal at the point of measurement, the mean-square error between the reference signal and the waveform to be analyzed is observed. The form of the reference after amplitude scaling and appropriate time delays have been imposed upon it to minimize the mean-square error is defined to be the signal component of the waveform under test. All other components of the waveform are grouped together and are termed noise. Thus errors resulting from both random noise and system distortion terms are included. With this definition, it is possible to define the SNR for any system where an appropriate reference signal can be obtained. In digital computer simulation, such a reference waveform is almost always available.

Once the reference and noisy waveforms have been obtained, the problem of estimating the SNR involves finding a method to determine the gain and delay values which yield the minimum mean-square error. It will be shown in later sections that the appropriate gain and delay values can be obtained using the cross-correlation function between the noisy waveform and the reference. For bandpass systems, the appropriate value of group delay will be seen to be the value

for which the envelope of the cross-correlation function is maximized. For a lowpass system the peak of the cross-correlation function itself indicates the average system delay. In either case the cross-correlation function must be found. For large sets of data, the computer time required to directly generate this function is prohibitive. By utilizing the FFT, the cross-correlation function can be computed quickly. Hilbert transform methods provide a way to obtain the envelope of this function when it is required. Scanning these functions for peak values provides a delay estimate. However, it was experimentally determined that the errors introduced in the successive transformations make the values obtained for the cross-correlation function unacceptable for estimating the SNR, except in the very noisy cases where the SNR is quite low. To obtain a good SNR estimate, a more accurate method is required for the final computations.

In the lowpass case, the signal component of the waveform may be obtained by merely time shifting and amplitude scaling the reference signal. A relatively simple method was found for accurately estimating the SNR in this case. When the system is bandpass, arbitrary group and phase delays are both involved; simple time shifting will not account for these effects. No simple scheme was found for deriving the signal component by performing a trivial operation on the reference for this case. The most direct methods seem to involve either a complicated filtering operation or successive Fourier transformations. The

errors introduced in these methods defeat the purpose, which is to obtain a highly accurate representation of the signal. A method was found to measure the SNR in direct RF simulations of bandpass systems, but it is more complex than the lowpass case. Therefore, lowpass and bandpass systems are treated separately in the following development of the estimators.

A question that is considered is that of how accurate and reliable the estimates obtained for the SNR are. A partial answer is obtained by simply using the estimators on a variety of systems where the SNR can be estimated by another means for comparison. Consideration is also given to the statistical properties of the estimators. It is helpful to compare these estimators to maximum likelihood estimators. Confidence intervals to indicate the reliability of the estimates are developed where practical, also.

After the SNR estimators have been developed and implemented, they are tested on a wide variety of systems. Both analog and digital systems are simulated and evaluated using the SNR estimators. The systems are necessarily chosen to be suitable for theoretical analysis, since the estimators themselves are being evaluated. For digital systems the desired parameter for evaluating performance is usually the symbol error probability. Therefore, in the testing programs, digital systems are simulated where the mapping from the SNR to P_E is known. For P_E ranges where it is practical, a logical test for digital systems is to compare the error probability estimate computed from the measured SNR

to the error probability obtained by direct error count. Obviously this testing is restricted to large error probabilities, since the direct counting technique is impractical elsewhere. However, it provides an excellent test for the SNR estimators where it can be applied. Where the SNR is large, it is only possible to directly test the estimation of the SNR itself, using simple systems where gain and phase information are known, and the true SNR can be computed. For this range it is not practical to make direct comparisons with the error probability, since there exists no feasible alternative to the SNR estimators for determining P_E at low error probabilities--the very reason for developing these estimators in the first place.

IV. AN SNR ESTIMATOR FOR COMPUTER SIMULATIONS OF LOWPASS SYSTEMS

A. THEORETICAL DEVELOPMENT

In analyzing a communication system by computer simulation, or in evaluating the simulation itself, quantities of interest are the mean-square error and the signal-to-noise ratio at various points in the simulation. Measurements of system gain, system delay, and power are also important in the analysis of many systems. Subroutine SNRMSE is a computer routine which was developed to estimate these quantities in digital computer simulations of wide-sense stationary lowpass systems.

In computing all these quantities except power, two sets of data are required: a reference array and an array of measurement data to be analyzed. For the computation of the error, the measurement data is compared to an amplitude scaled and time-shifted representation of the reference data. A lowpass system which exhibits only amplitude scaling and a time delay is considered to be distortionless or ideal. Optimal estimates for system gain and system delay are defined to be those values of the gain and delay which minimize the mean-square error between the measurement data and the reference data.

The reference signal will be denoted by x(t), and the measurement data by y(t). Since the ideal system is assumed to impose only amplitude scaling and a time shift

on the reference, the ideal signal at any point may be represented by $Ax(t-\tau)$, where A is the assumed gain of the system and τ is the assumed delay. Then, letting an overbar denote a time average, the mean-square error is given by

$$E(A,\tau) = \overline{[y(t) - Ax(t-\tau)]^2} \qquad (1)$$

Figure 1 shows the test configuration which is assumed for computing the mean-square error. Expanding (1),

$$E(A,\tau) = \overline{y^2(t) - 2Ax(t-\tau)\, y(t) + A^2x^2(t-\tau)}$$

$$E(A,\tau) = \overline{y^2(t)} - \overline{2Ax(t-\tau)\, y(t)} + \overline{A^2x^2(t-\tau)}.$$

And, assuming a stationary system,

$$E(A,\tau) = P_y - 2AR_{xy}(\tau) + A^2P_x \qquad (2)$$

where

P_y = power in the measurement data y(t)

P_x = power in the reference data x(t)

$R_{xy}(\tau)$ = cross-correlation function between the reference data and the measurement data.

It is clear that P_y and A^2P_x are positive quantities. Hence, for any value of A, $E(A,\tau)$ is minimized when $[AR_{xy}(\tau)]$ is maximized. Assuming system gain A is positive, the optimal estimate for system delay must therefore be the value of delay for which $R_{xy}(\tau)$ is maximized.

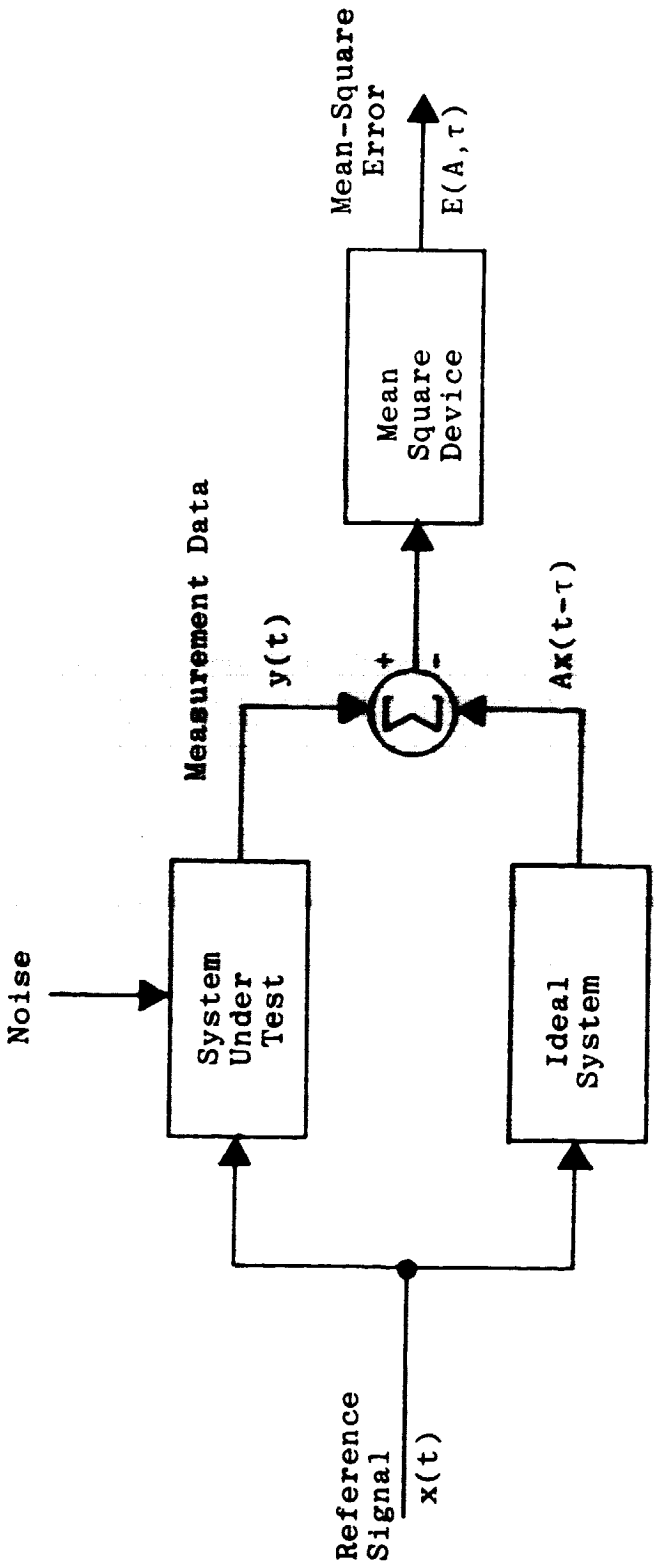

Figure 1. Test Configuration for Mean-Square Error

Denoting this value of delay by τ_m, the mean-square error becomes

$$E(A,\tau_m) = P_y - 2AR_{xy}(\tau_m) + A^2P_x.$$

With τ_m fixed, the expression for $E(A,\tau_m)$ is differentiated with respect to A and set equal to zero to yield the optimal estimate for system gain. This estimate is denoted by A_m so that

$$\left.\frac{dE(A,\tau_m)}{dA}\right|_{A=A_m} = -2R_{xy}(\tau_m) + 2A_mP_x = 0.$$

Thus

$$A_m = \frac{R_{xy}(\tau_m)}{P_x} . \tag{3}$$

From (2) and (3), the resulting mean-square error is given by

$$E(A_m,\tau_m) = P_y - \frac{R_{xy}^2(\tau_m)}{P_x} . \tag{4}$$

From the definitions of A_m and τ_m, the estimate of the signal at the point where the measurement data is taken must be $A_mx(t-\tau_m)$. Thus the signal power at that point in the system is estimated to be

$$S = \overline{[A_mx(t-\tau_m)]^2} = A_m^2P_x \tag{5}$$

or

$$S = \frac{R_{xy}^2(\tau_m)}{P_x} \quad . \tag{6}$$

In obtaining the SNR estimate, all components of y(t) which are not included in the signal estimate are termed noise. That is, the SNR is defined as the ratio of the signal power to the noise power at the point where the measurement data is taken; and noise will include any distortion or roundoff errors present in addition to the additive random noise. Then the SNR may be expressed by

$$SNR = \frac{S}{E(A_m, \tau_m)} = \frac{\left[\frac{R_{xy}^2(\tau_m)}{P_x}\right]}{\left[P_y - \frac{R_{xy}^2(\tau_m)}{P_x}\right]}$$

or

$$SNR = \frac{R_{xy}^2(\tau_m)}{P_x P_y - R_{xy}^2(\tau_m)} \quad . \tag{7}$$

If the normalized cross-correlation is defined by

$$\rho = \frac{\overline{x(t-\tau)y(t)}}{\sqrt{\overline{x^2(t)}} \cdot \sqrt{\overline{y^2(t)}}} = \frac{R_{xy}(\tau)}{\sqrt{P_x P_y}}$$

then $-1 \leq \rho \leq 1$. It is possible to express the SNR in the form

$$SNR = \frac{\rho^2}{1 - \rho^2} \quad . \tag{8}$$

Similarly, if the error is normalized by P_y,

$$E_N = 1 - \frac{R_{xy}^2(\tau_m)}{P_x P_y} \tag{9}$$

or

$$E_N = 1 - \rho^2. \tag{10}$$

Estimates of the various system parameters in terms of P_x, P_y, and $R_{xy}(\tau_m)$ are given in Table I.

Equation (8) illustrates that the SNR estimate is accurate if the normalized cross-correlation ρ is accurately determined. It is important to note that ρ is independent of the power contained in the reference and measurement data. Therefore, very accurate estimates can be obtained for the SNR, even in the presence of large errors in the individual measurements for P_x, P_y, and $R_{xy}(\tau)$. As long as the same data are used to compute P_x and P_y as were used in determining $R_{xy}(\tau)$, errors in the measurements often tend to be cancelled when $R_{xy}(\tau)$ is divided by $\sqrt{P_x P_y}$. This effect is used to advantage in the SNR estimation routine.

When the SNR is to be estimated in a digital computer simulation, the values of $R_{xy}(\tau_m)$, P_x, and P_y must be estimated from a finite set of sample values. If a set of N samples are to be used, equation (7) becomes

TABLE I.

PARAMETER ESTIMATES FOR LOWPASS SYSTEMS

IN TERMS OF P_x, P_y, τ_m, AND $R_{xy}(\tau_m)$

Parameter	Estimate
Gain	$A_m = \frac{R_{xy}(\tau_m)}{P_x}$
Delay	τ_m (As defined in text)
Signal power at the point of measurement	$S = \frac{R_{xy}^2(\tau_m)}{P_x}$
Mean-square error	$E(A_m,\tau_m)=P_y - \frac{R_{xy}^2(\tau_m)}{P_x}$
Normalized mean-square error*	$E_N = 1 - \frac{R_{xy}^2(\tau_m)}{P_x P_y}$
	$E_N = 1 - \rho^2$
Signal-to-noise ratio*	$SNR = \frac{R_{xy}^2(\tau_m)}{P_x P_y - R_{xy}^2(\tau_m)}$
	$SNR = \frac{\rho^2}{1 - \rho^2}$

*$\rho = \frac{R_{xy}(\tau_m)}{\sqrt{P_x P_y}}$

$$(SNR) = \frac{\left[\frac{1}{N}\sum_{i=1}^{N} x(t_i-\tau_m)y(t_i)\right]^2}{\left[\frac{1}{N}\sum_{i=1}^{N} x^2(t_i-\tau_m)\right]\left[\frac{1}{N}\sum_{i=1}^{N} y^2(t_i)\right]-\left[\frac{1}{N}\sum_{i=1}^{N} x(t_i-\tau_m)y(t_i)\right]^2} \qquad (11)$$

where the subscript denotes a time sample.

It is possible to determine whether this SNR estimate is a maximum likelihood estimate for some additive noise channels. Consider the case where the measurement data are given by

$$y(t) = A_m x(t-\tau_m) + n'(t). \qquad (12)$$

The estimates described earlier for gain and delay will always give a representation of y(t) in this form where $n'(t)$ represents all components of y(t) which are orthogonal to $x(t-\tau_m)$. This may be seen by noting that from equations (1) and (12)

$$\overline{n'^2(t)} = \overline{[y(t) - A_m x(t-\tau)]^2} = E(A_m, \tau_m) \qquad (13)$$

and also from (12),

$$\overline{y^2(t)} = \overline{[A_m x(t-\tau_m)]^2} + \overline{n'^2(t)} + 2A_m\overline{x(t-\tau_m)n'(t)}.$$

Substituting (5) and (13) into the above expression yields

$$P_y = S + E(A_m,\tau_m) + 2A_m\overline{x(t-\tau_m)\, n'(t)}. \qquad (14)$$

However, combining (4) and (6) shows that

$$P_y = S + E(A_m, \tau_m). \tag{15}$$

Thus, comparing (14) and (15),

$$\overline{A_m x(t-\tau_m) n'(t)} = 0.$$

In fact, an alternative derivation of the SNR estimator can be easily accomplished by using the orthogonality principle applied to linear minimum mean-square estimation. Thus, the representation of y(t) in (12) is valid where $n'(t)$ represents all components of y(t) orthogonal to $x(t-\tau_m)$. This representation is useful in determining whether the estimate given by (11), based on minimizing the mean-square error, is also the maximum likelihood (ML) estimate for the SNR.

To find the ML estimate for the SNR it is first necessary to obtain the joint probability density function for samples from y(t) as a function of gain and delay. For the additive white Gaussian noise channel

$$y(t_i) = Ax(t_i-\tau) + n(t_i) \tag{16}$$

where the subscript i denotes a time sample. The noise samples are independent and zero mean. Thus the samples of y(t), denoted $y(t_i)$, are Gaussian with means given by $Ax(t_i-\tau)$. These samples are also independent. Denoting

the density function for $y(t_i)$ by $f_i(y_i|A,\tau)$, the joint probability density function for N samples from y(t) is given by

$$f_{\underline{y}}(\underline{\bar{y}}\,|A,\tau) = \prod_{i=1}^{N} f_i(y_i|A,\tau) \quad . \tag{17}$$

The values of A and τ for which this likelihood function is maximized are the ML estimates for gain and delay. If the noise variance is denoted by σ_n^2, the likelihood function becomes

$$f_{\underline{y}}(\underline{\bar{y}}|A,\tau) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left[-\left(\frac{y(t_i)-Ax(t_i-\tau)}{\sqrt{2}\,\sigma_n}\right)^2\right]$$

or

$$f_{\underline{y}}(\underline{\bar{y}}|A,\tau)=(2\pi\sigma_n^2)^{\frac{-N}{2}} \exp\left\{-\left[\frac{1}{2\sigma_n^2}\sum_{i=1}^{N}\left(y^2(t_i) + A^2x^2(t_i-\tau) - 2Ax(t_i-\tau)y(t_i)\right)\right]\right\}.$$

The log-likelihood function is

$$-\frac{1}{2\sigma_n^2}\sum_{i=1}^{N}\left[y^2(t_i) + A^2x^2(t_i-\tau) - 2Ax(t_i-\tau)y(t_i)\right]$$

$$-\frac{N}{2}\ln(2\pi\sigma_n^2)$$

or

$$\frac{1}{2\sigma_n^2}\left[2A\sum_{i=1}^{N}x(t_i-\tau)y(t_i)-A^2\sum_{i=1}^{N}x^2(t_i-\tau)-\sum_{i=1}^{N}y^2(t_i)\right]$$

$$-\frac{N}{2}\ln(2\pi\sigma_n^2).$$

Since $x(t)$ is assumed to be wide-sense stationary, the second summation term is not a function of τ. Thus it is clear that, assuming positive gain, the log-likelihood function is maximized as a function of τ when the term $\sum_{i=1}^{N}x(t_i-\tau)y(t_i)$ is maximized. From (7) and (11), the discrete form for $R_{xy}(\tau)$ is $\frac{1}{N}\cdot\sum_{i=1}^{N}x(t_i-\tau)y(t_i)$. Therefore, the delay value required to maximize the log-likelihood function is τ_m. Hence the delay estimate given by the minimum mean-square error technique is the ML delay estimate for the additive white Gaussian noise channel.

It can be shown that the ML estimate of the ratio of two parameters is simply the ratio of the ML estimates [45]. Therefore, in order to estimate the SNR it is only necessary to form ML estimates for the signal power and the noise power and compute the ratio. Denoting the power in the reference by P_x and noting that the discrete form for P_x in (11) is $\frac{1}{N}\sum_{i=1}^{N}x^2(t_i-\tau_m)$, the log-likelihood function may be expressed as

$$\frac{1}{2\sigma_n^2}\left[2A\sqrt{P_x}\sum_{i=1}^{N}\left(\frac{x(t_i-\tau_m)}{\sqrt{P_x}}\right)y(t_i) - NA^2P_x - \sum_{i=1}^{N} y^2(t_i)\right]$$

$$- \frac{N}{2}\ln(2\pi\sigma_n^2).$$

The power contained in the signal component of y(t) is given by A^2P_x. Defining $S_x = A^2P_x$ and rewriting the log-likelihood function yields

$$\frac{1}{2\sigma_n^2}\left[2\sqrt{S_x}\sum_{i=1}^{N}\left(\frac{x(t_i-\tau_m)}{\sqrt{P_x}}\right)y(t_i) - NS_x - \sum_{i=1}^{N} y^2(t_i)\right]$$

$$- \frac{N}{2}\ln(2\pi\sigma_n^2)$$

again assuming that A is positive.

The ML estimates for S_x and (σ_n^2) can be found by setting the partial derivatives with respect to S_x and (σ_n^2) equal to zero and solving. Then the ML estimate for the SNR is the ratio of these estimates.

For the signal power, the partial derivative of the log-likelihood function yields

$$\frac{1}{\sqrt{\hat{S}_x}}\sum_{i=1}^{N}\left(\frac{x(t_i-\tau_m)}{\sqrt{P_x}}\right)y(t_i) - N = 0$$

or

$$\hat{S}_x = \frac{1}{P_x}\left[\frac{1}{N}\sum_{i=1}^{N} x(t_i-\tau_m)y(t_i)\right]^2. \tag{18}$$

For the noise power, (σ_n^2), the partial derivative of the log-likelihood function yields

$$-\frac{1}{2(\hat{\sigma}_n^2)^2}\left[2\sqrt{\hat{S}_x}\sum_{i=1}^{N}\left(\frac{x(t_i-\tau_m)}{\sqrt{P_x}}\right)y(t_i) - N\hat{S}_x - \sum_{i=1}^{N} y^2(t_i)\right]$$

$$-\frac{N}{2(\hat{\sigma}_n^2)} = 0$$

and using (18)

$$\frac{1}{(\hat{\sigma}_n^2)}\left[2N\hat{S}_x - N\hat{S}_x - \sum_{i=1}^{N} y^2(t_i)\right] = -N$$

or

$$\hat{\sigma}_n^2 = \frac{1}{N}\sum_{i=1}^{N} y^2(t_i) - \hat{S}_x. \tag{19}$$

Then, expressing $(\hat{SNR})$ as the ratio $\hat{S}_x/\hat{\sigma}_n^2$ yields

$$(\hat{SNR}) = \frac{\frac{1}{P_x}\left[\frac{1}{N}\cdot\sum_{i=1}^{N} x(t_i-\tau_m)y(t_i)\right]^2}{\frac{1}{N}\sum_{i=1}^{N} y^2(t_i) - \hat{S}_x}.$$

Using (18) to replace $\hat{S}_x$ and using the summation which represents the discrete formulation for P_x results in

$$(\hat{SNR}) = \frac{\left[\frac{1}{N}\sum_{i=1}^{N} x(t_i-\tau_m)y(t_i)\right]^2}{\left[\frac{1}{N}\sum_{i=1}^{N} x^2(t_i-\tau_m)\right]\left[\frac{1}{N}\sum_{i=1}^{N} y^2(t_i)\right] - \left[\frac{1}{N}\sum_{i=1}^{N} x(t_i-\tau_m)y(t_i)\right]^2} \tag{20}$$

which is identical to (11). Thus the SNR estimator developed using the minimum mean-square error criterion yields the ML estimate of the SNR for the white additive Gaussian noise channel.

If confidence intervals are to be found, it is necessary to determine the probability density function of the SNR estimate. In the additive white Gaussian noise case, the required density function can be found unless the SNR is low. Except for low values of the SNR, it has been experimentally determined that delay is estimated in almost all cases with sufficient accuracy that the error in the SNR estimate generated by inaccurately determining delay is small compared to other error sources. When delay errors of significant size do occur, they are usually of sufficient magnitude to be quite obvious. Therefore, it is assumed that τ, the system delay, has been correctly determined. In order to determine the density function for the SNR estimate, first denote the term in brackets in (18) as

$$G = \frac{1}{N} \sum_{i=1}^{N} x(t_i - \tau) y(t_i).$$

For the additive Gaussian noise channel described by (16), each $y(t_i)$ is Gaussian distributed; therefore, this weighted sum of $y(t_i)$ samples must also be Gaussian. The mean and variance of G are easily obtained. Replacing τ in (16) by τ_m and computing the expected value of G yields

$$E\{G\} = E\left[\frac{1}{N}\sum_{i=1}^{N} x(t_i-\tau_m)[Ax(t_i-\tau_m) + n(t_i)]\right]$$

$$= \frac{A}{N}\sum_{i=1}^{N} x^2(t_i-\tau_m) + \frac{1}{N}\sum_{i=1}^{N} x(t_i-\tau_m)E\{n(t_i)\}$$

$$= \frac{A}{N}\sum_{i=1}^{N} x^2(t_i-\tau_m) + 0$$

$$= AP_x.$$

Also, the variance is easily found as follows:

$$E\{[G-E(G)]^2\} = E\left[\left(\frac{1}{N}\sum_{i=1}^{N} x(t_i-\tau_m)[Ax(t_i-\tau_m)+n(t_i)]\right.\right.$$

$$\left.\left. - \frac{A}{N}\sum_{i=1}^{N} x^2(t_i-\tau_m)\right)^2\right]$$

$$= E\left[\left(\frac{1}{N}\sum_{i=1}^{N} x(t_i-\tau_m)\, n(t_i)\right)^2\right]$$

$$= \frac{1}{N^2}\, E\left[\sum_{i=1}^{N}\sum_{j=1}^{N} x(t_i-\tau)x(t_j-\tau)n(t_i)n(t_j)\right].$$

Since $E\{n(t_i)n(t_j)\} = 0$ for $i \neq j$,

$$E\{[G-E(G)]^2\} = \frac{1}{N^2}\sum_{i=1}^{N} x^2(t_i-\tau)\,\sigma_n^2$$

$$= \frac{\sigma_n^2}{N}\left[\frac{1}{N}\sum_{i=1}^{N} x^2(t_i-\tau)\right]$$

$$= \frac{\sigma_n^2}{N}\, P_x.$$

It is useful to define a normalized statistic with unity variance. Therefore, let

$$G' = \frac{\sqrt{N}}{\sqrt{P_x}\,\sigma_n} G$$

and note

$$E\{G'\} = \frac{\sqrt{N}\,\sqrt{P_x}\,A}{\sigma_n}$$

and

$$\sigma^2_{G'} = 1.$$

It then follows that (G'^2) is a non-central chi-square random variable with non-centrality parameter [46]

$$\lambda = E\{G'\}^2 = \frac{NP_xA^2}{\sigma_n^2}$$

or

$$\lambda = N(SNR). \tag{21}$$

Note that $(\hat{SNR})$ can be written as

$$(\hat{SNR}) = \frac{\hat{S}_x}{(\hat{\sigma_n^2})} \tag{22}$$

$$= \frac{G^2}{P_x(\hat{\sigma_n^2})}$$

$$= \frac{\left(\frac{P_x \sigma_n^2}{N}\right) G'^2}{P_x(\hat{\sigma}_n^2)}$$

$$= \frac{G'^2}{\left[\frac{N}{\sigma_n^2}\right](\hat{\sigma}_n^2)} \cdot \qquad (23)$$

Also, from (22) and (23),

$$G'^2 = \left[\frac{\sigma_n^2}{N}\right] \hat{S}_x. \qquad (24)$$

Now, from (18) and (19) it is clear that $\hat{S}_x$ and $(\hat{\sigma}_n^2)$ represent the square of a weighted sample mean and the sample variance, respectively, of a Gaussian population. The weighted sample mean and the sample variance of a set of samples from a Gaussian population are independent [47]. Hence, the distribution for $(\hat{SNR})$ is the ratio of two independent random variables. Using the form in (23), the numerator has been shown to have a non-central chi-square density. The denominator, which is just a scale factor times the sample variance, is a central chi-square statistic with (N-1) degrees of freedom [48].

It is well known that the ratio of two central chi-square random variables leads to an F-distribution; the ratio of a non-central chi-square random variable to a central chi-square random variable follows a non-central F-distribution [49]. If the chi-square statistics are

normalized by their respective numbers of degrees of freedom, a new random variable is defined using (23) by

$$F' = \frac{(G'^{,2})/m}{\left[\frac{N}{\sigma_n^2}\right](\hat{\sigma}_n^2)/n} \tag{25}$$

where

m = number of degrees of freedom of the non-central chi-square numerator

n = number of degrees of freedom of the central chi-square denominator.

This ratio of normalized chi-square random variables has a non-central F density given [50], [51] by

$$f_F(F'|m,n,\lambda) = \sum_{i=0}^{\infty} e^{-\frac{\lambda}{2}} (\tfrac{\lambda}{2})^i (\tfrac{1}{i!})\, f_F(F'|m+2i,n)$$

with

$$f_F(F'|k,n) = \frac{\Gamma(\frac{k+n}{2})}{\Gamma(\frac{k}{2})\Gamma(\frac{n}{2})} (\tfrac{k}{n})^{\frac{k}{2}} \frac{(F')^{\frac{k-2}{2}}}{\left[1 + \frac{k}{n} F'\right]^{\frac{k+n}{2}}} \quad \text{for } F'>0$$

$$= 0 \quad \text{for } F'<0 \tag{26}$$

where λ = non-centrality parameter of the numerator of (25); it is given by (21). Finally, note that from (25)

$$F' = \frac{G'^{,2}}{\left[\frac{N}{\sigma_n^2}\right](\hat{\sigma}_n^2)} (\tfrac{n}{m})$$

so that using (23)

$$F' = \frac{n}{m} (\hat{SNR}) \tag{27}$$

which provides the relationship between F' and $(\hat{SNR})$.

Equations (26) and (27) determine the density function for $(\hat{SNR})$. For ease of computation, several approximations to the non-central F cumulative distribution function are available. One method involves approximating the non-central F-distribution with a central F-distribution, which is a widely tabulated function [52]. Another approximation given in the same reference makes use of the standard normal distribution function. This approximation for the cumulative distribution function is given by

$$F_{F'}(F' | m,n,\lambda) \simeq P(x_1)$$

where

$$x_1 = \frac{\left[\frac{mF'}{m+\lambda}\right]^{\frac{1}{3}} \left[1 - \frac{2}{9n}\right] - \left[1 - \frac{2(m+2\lambda)}{9(m+\lambda)^2}\right]}{\left[\frac{2}{9}\left(\frac{m+2\lambda}{(m+\lambda)^2}\right) + \frac{2}{9n}\left(\frac{mF'}{m+\lambda}\right)^{\frac{2}{3}}\right]^{\frac{1}{2}}}$$

and $P(\cdot)$ is defined by

$$P(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}} \, dt. \tag{28}$$

Substituting the values m=1, n=N-1, and λ=N(SNR) from (21) into this expression yields an approximation for

the cumulative distribution function, $F(\hat{SNR})$, in terms of the number of data points used, N, and the true SNR.

$$F(\hat{SNR}) \approx P\left\{\frac{\left[\frac{(N-1)(\hat{SNR})}{1+N(SNR)}\right]^{\frac{1}{3}}\left[1-\frac{2}{9(N-1)}\right]-\left[1-\frac{2+4N(SNR)}{9(1+N(SNR))^2}\right]}{\left[\frac{2+4N(SNR)}{9(1+N(SNR))^2}+\frac{2}{9(N-1)}\left[\frac{(N-1)(\hat{SNR})}{1+N(SNR)}\right]^{\frac{2}{3}}\right]^{\frac{1}{2}}}\right\}. \tag{29}$$

Using (29) with $P(\cdot)$ defined in (28), confidence intervals can be constructed. Ninety-nine percent confidence interval curves have been computed for SNR from 0 to 20 dB assuming N=1000, N=5000, and N=50,000 samples used for the $(\hat{SNR})$ estimate. Plots of the confidence interval curves for the white Gaussian noise channel are shown for SNR values from 0 to 10 dB in Figure 2 and for SNR values from 10 to 20 dB in Figure 3. For the values of N considered, the widths of the confidence intervals vary less than 0.01 dB for SNR values from 20 dB to 40 dB. Hence the results at 20 dB may be extended to include most higher values of the SNR that might be needed.

B. METHOD OF IMPLEMENTATION

In order to form the desired estimates for the SNR and the normalized mean-square error, it is first necessary to find estimates for P_x, P_y, and $R_{xy}(\tau_m)$. Perhaps the most straightforward method for obtaining these quantities is shown in Figure 4. The estimates for P_x and P_y are

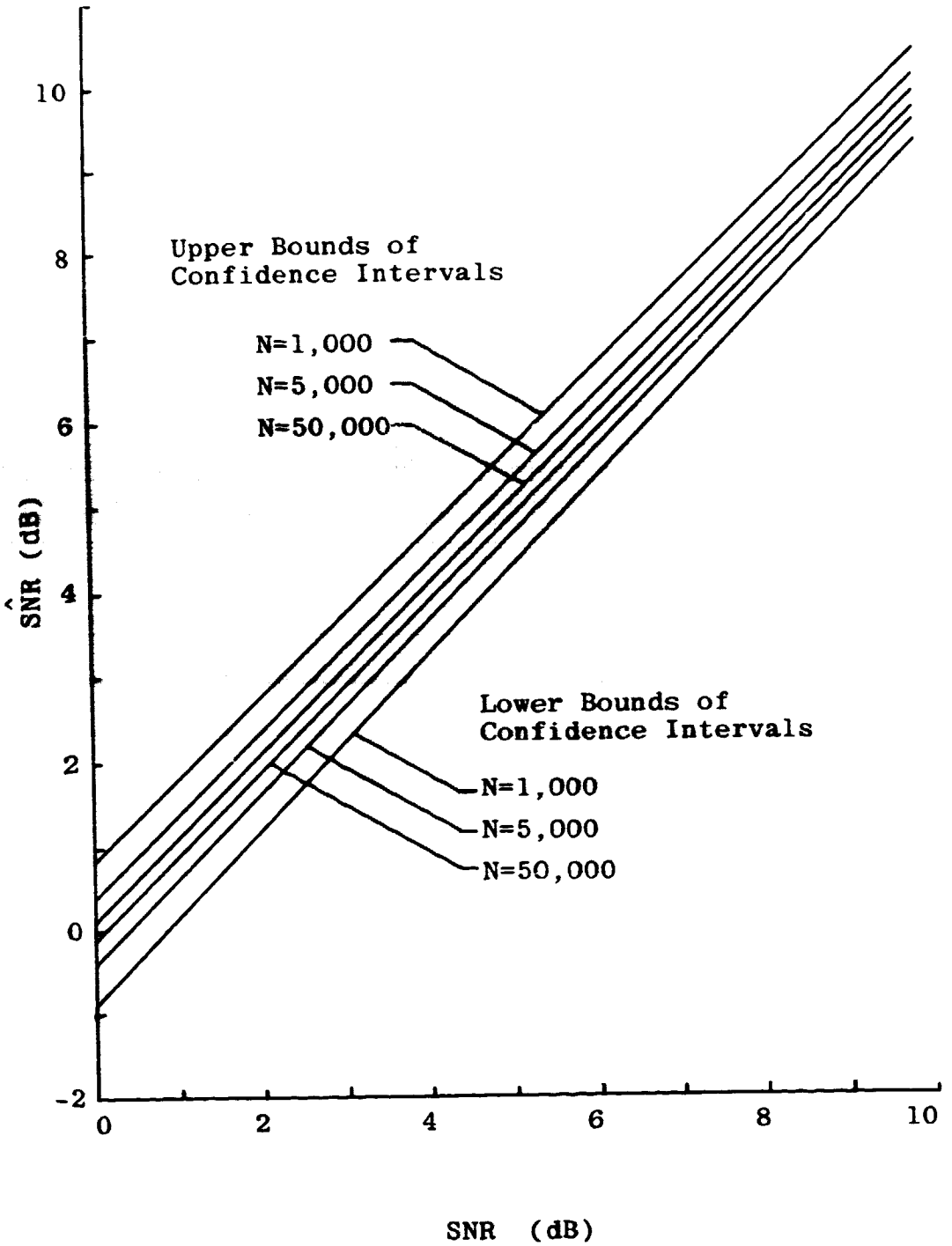

Figure 2. Ninety-nine Percent Conf dence Interval Curves, SNR = 0 to 10 dB

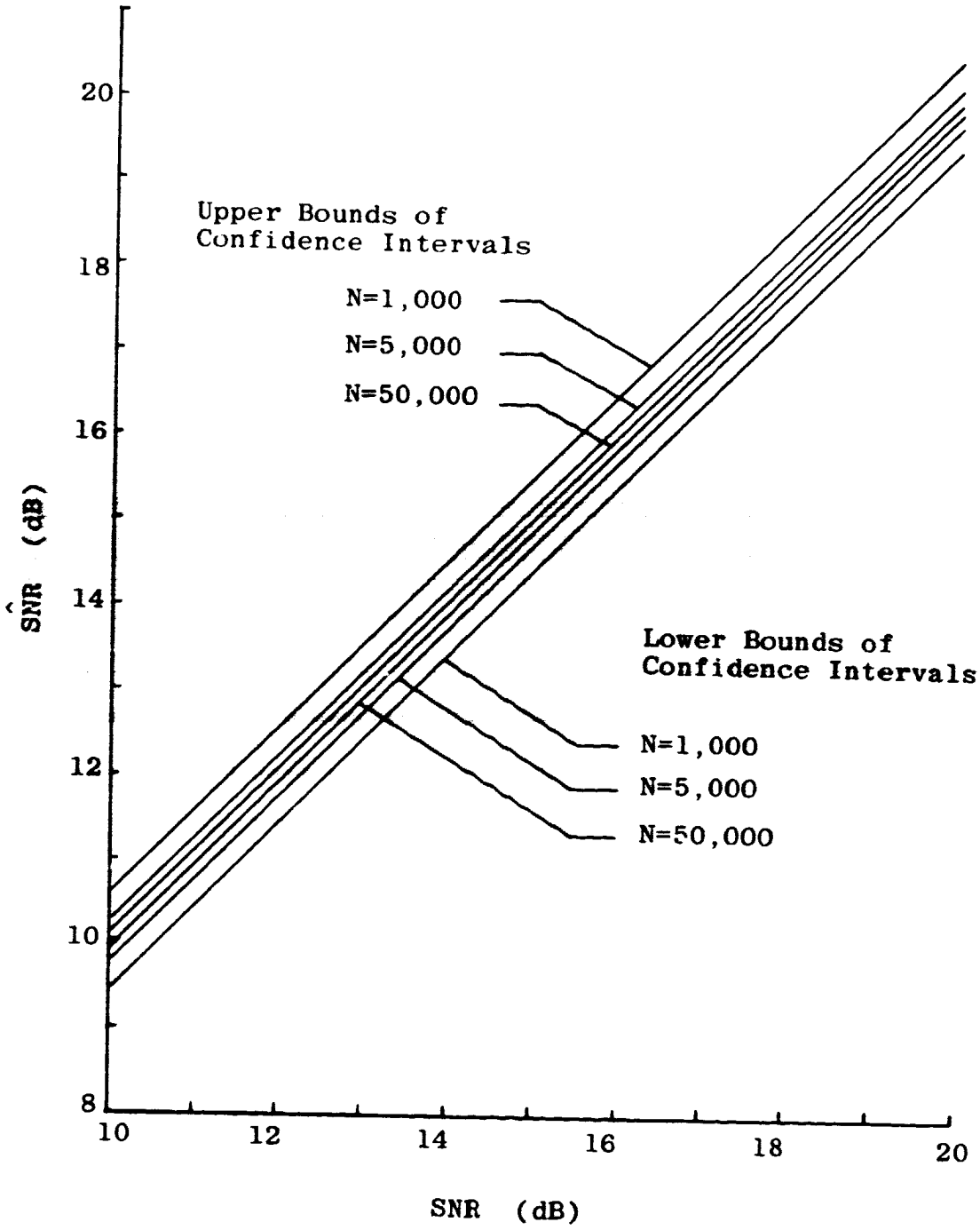

Figure 3. Ninety-nine Percent Confidence Interval Curves, SNR = 10 to 20 dB

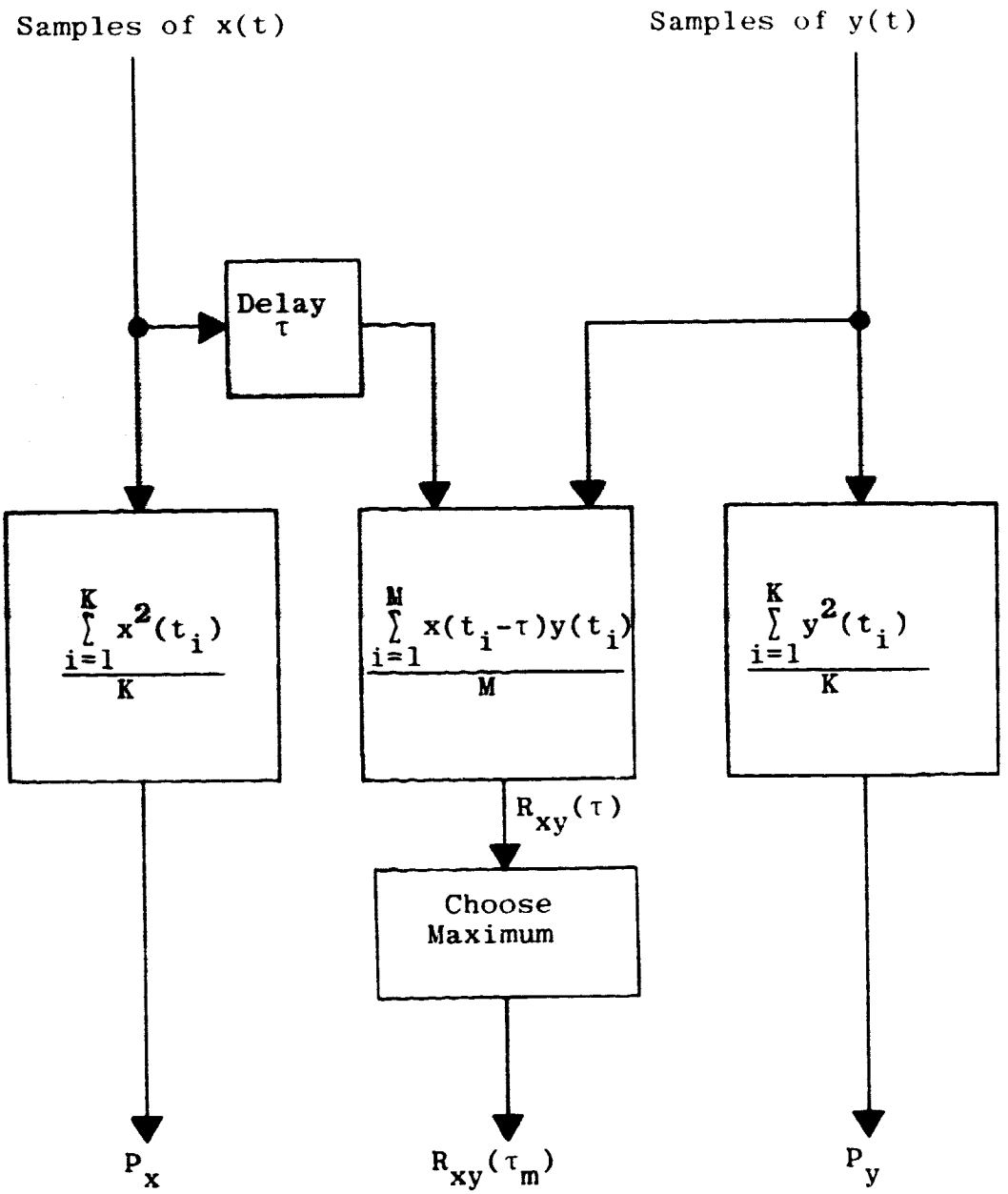

Figure 4. Direct Computation of P_x, P_y, and $R_{xy}(\tau_m)$

found by simply averaging the squared sample values. The cross-correlation estimate is found by delaying the samples from x(t), forming the products of samples $x(t_i-\tau)y(t_i)$ and averaging over the available sample pairs for any given delay τ. Then $R_{xy}(\tau_m)$ is found by computing $R_{xy}(\tau)$ for all values of τ where data are available and picking $R_{xy}(\tau_m)$ to be the maximum value computed. This provides the estimate for τ_m, also. It is clear from Table I that the values for P_x, P_y, and $R_{xy}(\tau_m)$ can be used to form all the desired estimates.

For estimating P_x and P_y, this method is easily applied. Unfortunately, however, this direct approach for estimating $R_{xy}(\tau_m)$ is extremely time consuming for large sets of data. The entire summation of products must be performed for each value of τ. A faster approach is to utilize the fast Fourier transform (FFT) algorithm to find $R_{xy}(\tau)$ by applying frequency domain techniques, and this is the procedure used here.

The first step implemented in the computer routine is to find the approximate value of system delay using the FFT. The necessary cross-correlation function, $R_{xy}(\tau)$, is the inverse Fourier transform of the cross-power spectral density, $S_{xy}(f)$. Thus

$$R_{xy}(\tau) = F^{-1}[S_{xy}(f)]$$

$$= F^{-1}[X^*(f)Y(f)] \qquad (30)$$

where the asterisk denotes the complex conjugate, and X(f) and Y(f) are the Fourier transforms of x(t) and y(t), respectively. Hence, it is possible to find the cross-correlation function between two sets of data x(t) and y(t) by finding their Fourier transforms X(f) and Y(f), conjugating X(f), forming the product $X^*(f)Y(f)$, and finally inverse Fourier transforming to obtain $R_{xy}(\tau)$. As explained earlier, the value of τ for which $R_{xy}(\tau)$ is maximized is the value which the FFT indicates will minimize the mean-square error. Figure 4 can be easily modified to utilize the FFT for finding $R_{xy}(\tau_m)$; the resulting configuration is shown in Figure 5.

It is convenient to be able to specify the size of the array to be Fourier transformed independent of the total amount of data collected. Often it is helpful to utilize relatively short transform lengths. One reason is that errors due to roundoff increase as the size of the transform is increased [53]. Perhaps more important, the amount of memory required to be in the main storage of the computer is governed chiefly by the size of the transforms used. The approach chosen allows the FFT size to be specified as any value less than or equal to the total number of data values available. Of course, many algorithms for performing the FFT impose the additional restriction that the number of points transformed be a power of two [54].

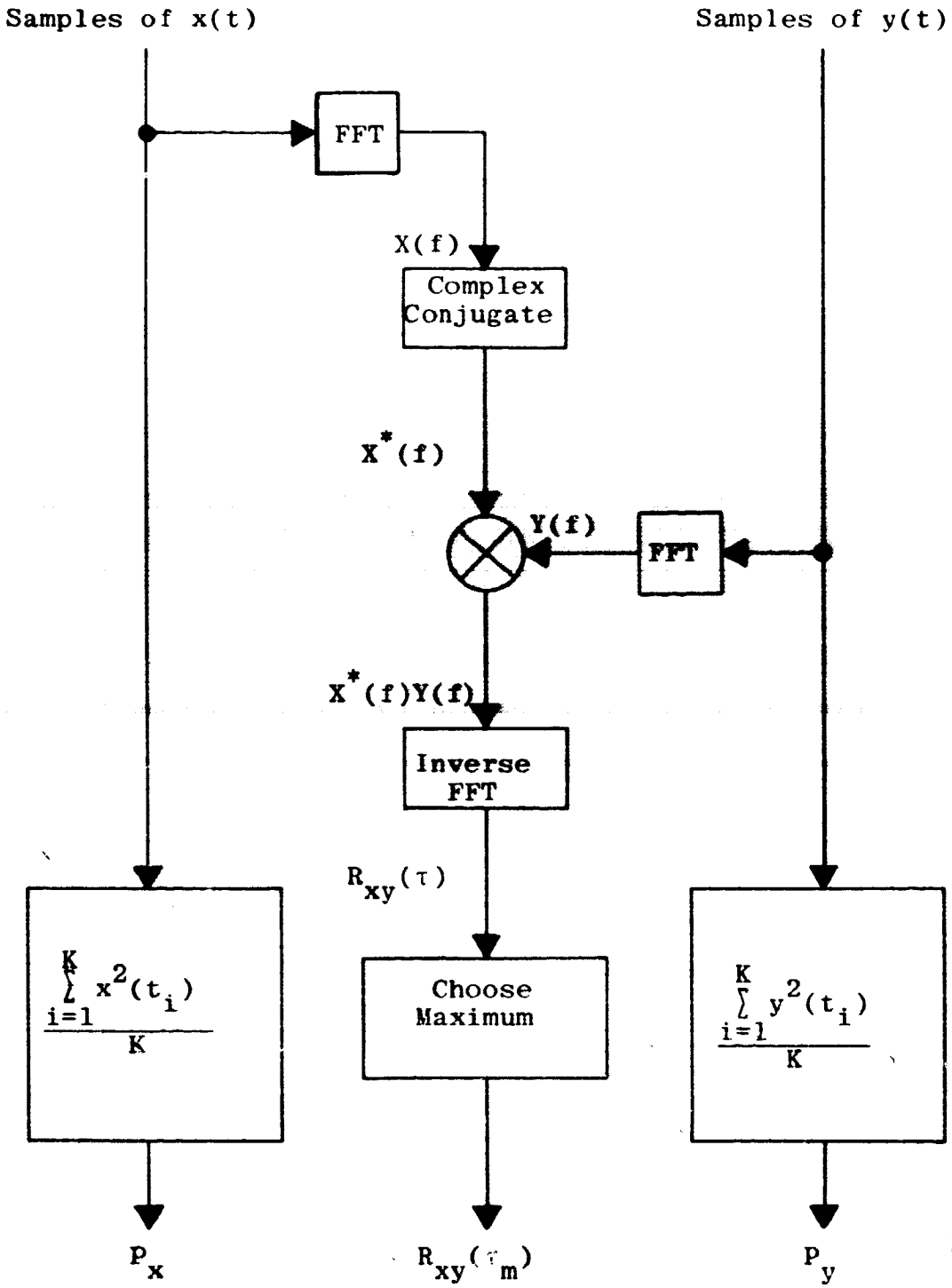

Figure 5. Computation of P_x, P_y, and $R_{xy}(\tau)$ Utilizing the FFT

The X and Y arrays contain the sample values from x(t) and y(t), respectively. Before Fourier transforming the data, the X and Y arrays, which are of length K, are divided into smaller blocks of length N. This is illustrated in Figure 6. Because of the assumed wide-sense stationarity of x(t) and y(t), it is possible to estimate $R_{xy}(\tau)$ by selecting a short block of data from the X array and correlating it with the appropriate samples from the Y array for all values of τ. The technique used to find $R_{xy}(\tau)$ is to transform blocks of data of length N one at a time and form $R_{xy}(\tau)$ as a number of separate segments. The block of $\frac{N}{2}$ samples taken from the beginning of the X array is padded with zeroes to form a block N samples long. This serves as a reference block. Then the first block of N samples is taken from the Y array. The reference block and the first measurement data block are Fourier transformed, and a segment of $R_{xy}(\tau)$ is found using the method described. Of the N values which result, the first $\frac{N}{2}$ values are valid values of $R_{xy}(\tau)$. The last $\frac{N}{2}$ values are extraneous data and are discarded. The last values fail to be valid because of the periodicity assumed in using the FFT algorithm. This cyclic property of the FFT is also the reason for padding the reference block with zeroes.

In order to find the next $\frac{N}{2}$ values of $R_{xy}(\tau)$, a new block of N samples is taken from y(t) which overlaps exactly one half the first block. This is also shown in Figure 6. These samples are processed just as for the

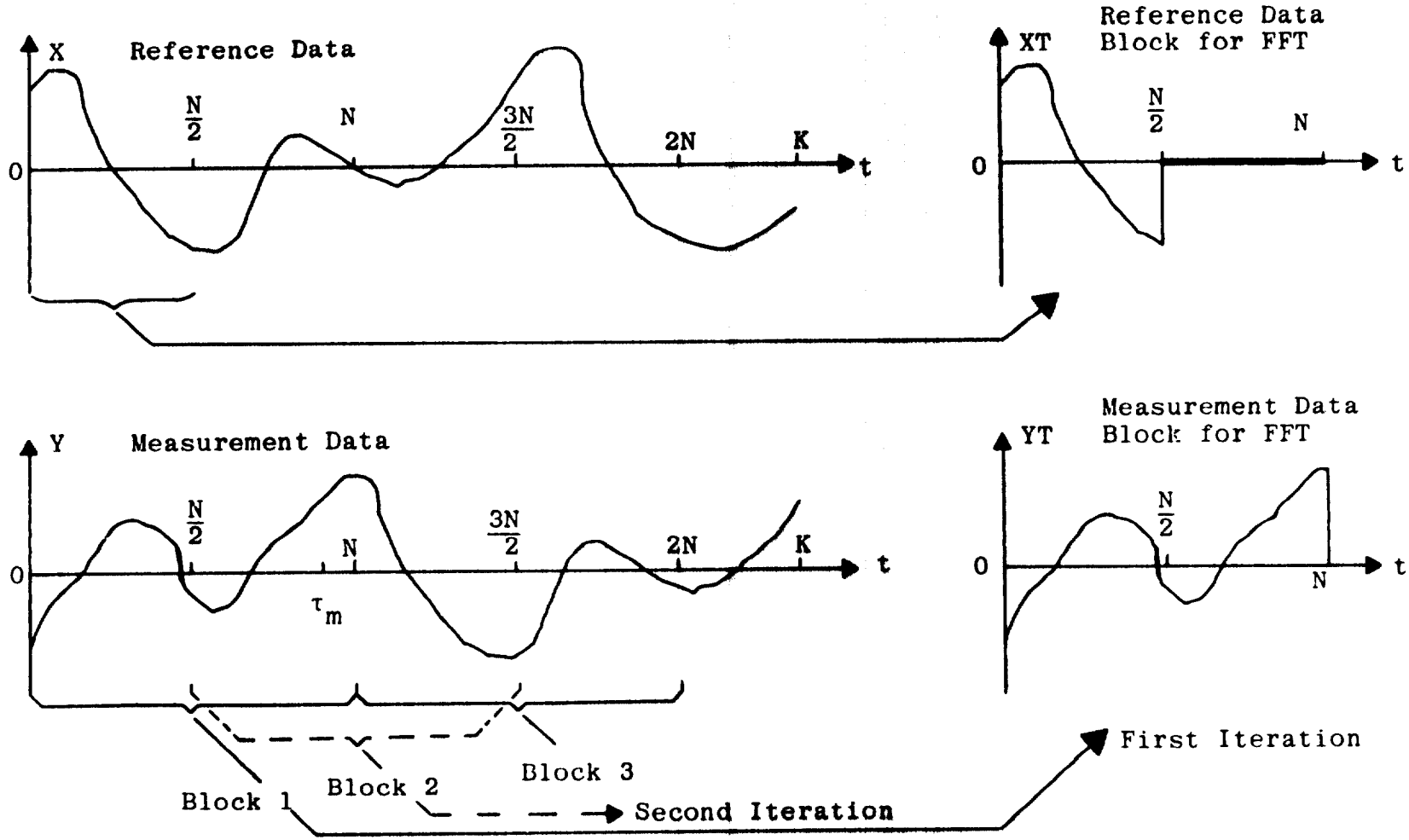

Figure 6. Manipulation of Data Blocks for Computing $R_{xy}(\tau)$

first block. The process is continued as long as there are data points in the Y array to transform. In this way $R_{xy}(\tau)$ is found for all delay values, except for the very large values where there is insufficient data to perform the last FFT. The $R_{xy}(\tau)$ array is then scanned to find the delay value for which it is a maximum. Finding $R_{xy}(\tau)$ in this way allows the length of the transformed arrays to be independent of the total amount of data to be analyzed, except that total data length serves as an upper bound on transform length.

Once a delay value has been found using the FFT, a direct time domain approach is used to compute several values of $R_{xy}(\tau)$ in the vicinity of the delay indicated by the FFT. There are two major reasons for doing this. One is that the FFT is affected by cumulative roundoff errors to a greater extent than is the direct (time domain) method. Another is the fact that each value of the correlation function computed with the FFT is obtained using only $\frac{N}{2}$ data values. The time domain method utilizes all the data available from the current delay value to the end of the data stream. Additionally, the time domain calculation includes normalization by the power in the samples being used at each iteration of the delay. This corrects for small changes in the power contained in the array as single samples are added and deleted from the $R_{xy}(\tau)$ calculation. After the delay is found, the other quantities follow easily. Provision is also made to use

a Hamming data window, if desired, to smooth the final estimates for $R_{xy}(\tau)$, P_x, and P_y.

A block diagram of the complete routine is shown in Figure 7. Input data for the routine consists of the X array of reference data, the Y array of measurement data, and various control parameters which indicate such things as whether to window the data, what the FFT block length is to be, etc. As shown in the diagram, the reference block is selected from the X array and loaded into a work array XT for transforming. A block of measurement data is selected from the Y array and loaded into the work array YT for transforming. Both XT and YT are transformed by the FFT, and the product $X^*(f)Y(f)$ is inverse transformed to yield $R_{xy}(\tau)$. This process is repeated for each block of measurement data, and the delay value where the FFT indicates a peak in $R_{xy}(\tau)$ is found. Then iterations are performed for a few values of τ around this peak, and the direct time domain method with double precision arithmetic is used to compute

$$\rho = \frac{R_{xy}(\tau)}{\sqrt{P_x P_y}} \tag{31}$$

for each iteration, where

$$R_{xy}(\tau) = \frac{1}{K-\tau} \sum_{i=1}^{K-\tau} X(i)Y(i+\tau) \tag{32}$$

and

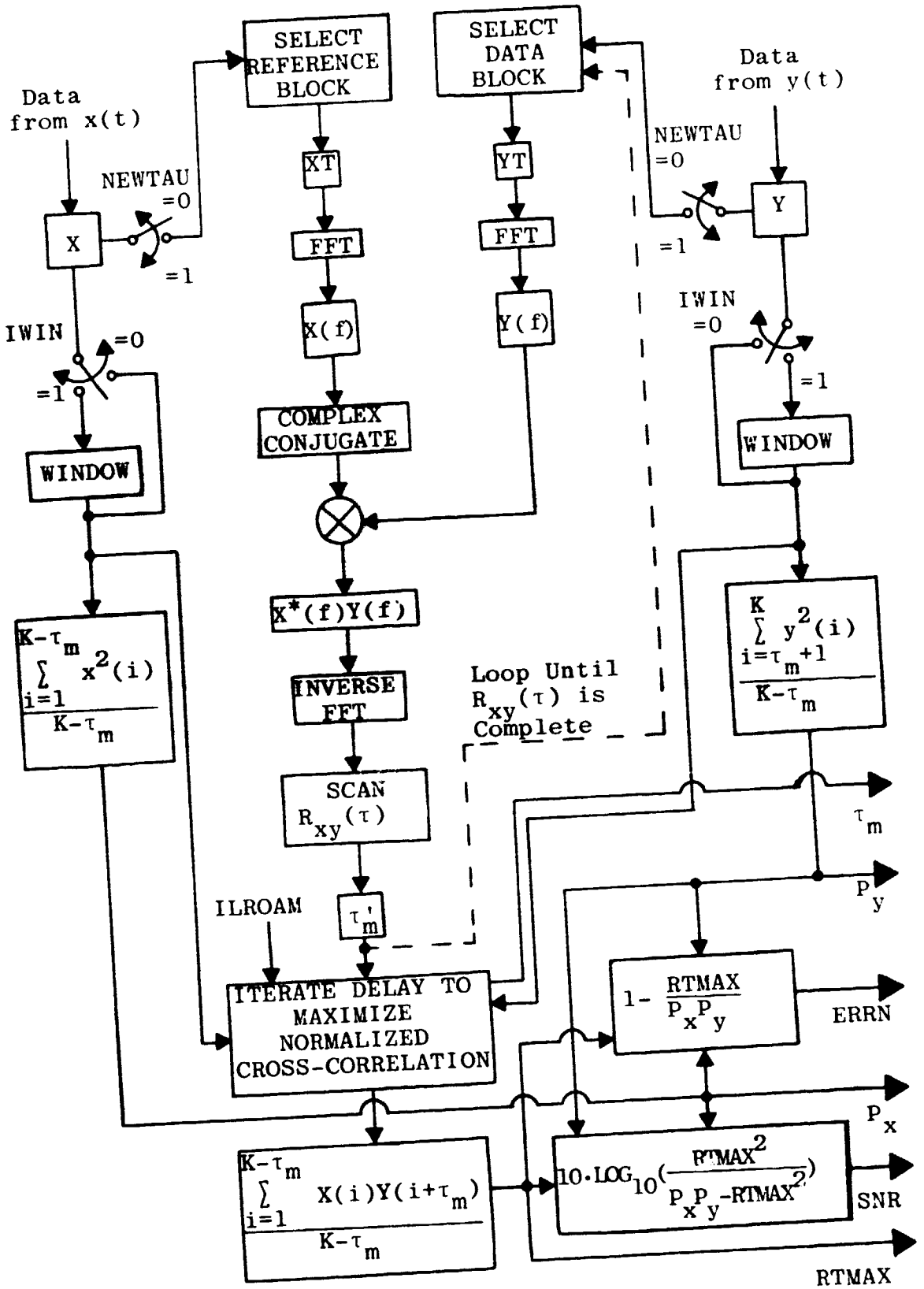

Figure 7. Block Diagram of SNRMSE

$$P_x = \frac{1}{K-\tau} \sum_{i=1}^{K-\tau} X^2(i) \qquad (33)$$

and

$$P_y = \frac{1}{K-\tau} \sum_{i=\tau+1}^{K} Y^2(i). \qquad (34)$$

In the above expressions, K is the total number of data samples in the X or Y array, and i is used to denote the sample index in the array. The number of iterations performed about the initial estimate for τ is specified by an input parameter, ILROAM. The peak of this normalized cross-correlation defines τ_m. The value of P_y is computed directly as the average of the squares of the samples in Y from (τ_m+1) through the end of the data. The value of P_x is similarly computed using the first $(K-\tau_m)$ samples in X. The Hamming window is applied to the data, if specified, in doing the final computations. These same sets of data samples are used to compute the cross-correlation peak, $R_{xy}(\tau_m)$. It is important in calculating P_x and P_y to use only those samples which were used in finding $R_{xy}(\tau_m)$, since the estimate for system gain should be the same in measuring $R_{xy}(\tau_m)$ as it is in measuring P_y. In computing the SNR, for example, using

$$SNR = \frac{R_{xy}^2(\tau_m)}{P_x P_y - R_{xy}^2(\tau_m)}$$

the quantities $(P_x P_y)$ and $R^2_{xy}(\tau_m)$ are identical in the first three, four, or even five digits for a high SNR. Thus it is critical to have identical estimates of gain implicit in these two terms. In order to make the mean-square error measurements independent of the effects of system gain and power in the reference data, a normalized mean-square error, ERRN, is computed using

$$ERRN = \frac{E(A, \tau_m)}{P_y} = 1 - \frac{R^2_{xy}(\tau_m)}{P_x P_y} \quad . \tag{35}$$

Double precision arithmetic is used for computing all final output values. A listing of the computer subroutine SNRMSE is given in Appendix A.

There are several things which must be considered in using the computer routine for SNR estimation if accurate results are to be obtained. Two obvious considerations concern the total number of data points used. Clearly, more data points will usually provide more accurate results. Also, the data stream must be long enough to make certain that the system delay value falls well within the total length of the Y data array being analyzed. Choosing a size for the FFT block involves more subtle considerations. Errors in the FFT algorithm increase as the block length of the transform increases, which would indicate smaller block lengths should be best However, the use of small block lengths forces the routine to select the preliminary delay value (the

value indicated by FFT) based upon short data segments. Hence, the routine is more likely to miss τ_m completely if the block length is small. One key assumption in the development of the routine is that the data is stationary from block to block. If the block length is chosen to be very short, this assumption may fail to be valid. The routine tests the mean and variance of each block to be transformed and generates a warning statement if the data does not appear approximately stationary. A brief description of these stationarity tests is given in Appendix A. These considerations indicate that very short FFT block lengths are also undesirable. Some typical values of block length which have been used with success are 2048, 4096, and 8192. All these values of block length seem to give excellent results. It does not appear that the largest block lengths used significantly degrade accuracy through round-off errors. Block lengths less than 2048, however, have occasionally been found to miss the initial delay estimate by large amounts. Results of tests using various block lengths are given in the section on applications to analog systems.

It has also been seen that the window function provides no noticeable improvement in the accuracy of the routine with most of the systems tested. For this reason, windowing was seldom used in the final testing of

the lowpass SNR estimator. It will be seen that this is not the case, however, for the bandpass SNR estimator.

V. AN SNR ESTIMATOR FOR COMPUTER SIMULATIONS OF BANDPASS SYSTEMS

A. THEORETICAL DEVELOPMENT

The problem of estimating the SNR in simulations of bandpass systems is complicated by the fact that simple time-shifting and amplitude scaling of the reference will not yield a replica of the signal component of the measurement data, as it does for lowpass systems. Estimates for both the group and phase delays must be simultaneously determined. This problem is solved by computing all the required quantities in the frequency domain. Subroutine SNRBPS is a computer routine which was developed to estimate the SNR and other parameters in simulations of wide-sense stationary bandpass systems.

As in the lowpass case, a reference signal x(t) and a measurement signal y(t) are required. The estimate for the signal component of y(t), denoted by z(t), is defined to be identical to the reference signal except for some unknown amplitude scaling and unknown group and phase delays. To form a minimum mean-square estimate of the signal, it is necessary to define z(t) such that the error

$$E = \overline{[y(t)-z(t)]^2} \tag{36}$$

is minimized, where the overbar denotes a time average. This implies that the power on the given interval in the difference signal [y(t)-z(t)] is to be minimized.

As a direct result of Parseval's theorem [55], it is clear that this is accomplished if power computed from the spectrum of [y(t)-z(t)] on the interval is minimized. The FFT may be used to compute the desired spectrum, and the error can be expressed by

$$E = \sum_{i=1}^{N} \left| Y(f_i) - Z(f_i) \right|^2$$

where $Y(f_i)$ and $Z(f_i)$ denote the ith samples from the Fourier transforms of y(t) and z(t), and N is used to denote the number of points transformed. For convenience, it will be assumed that N is even; if not, it can easily be made so by deleting one of the original data points. No significant accuracy is lost since only rather large data sets (N>500) are likely to be used here. If a binary radix FFT is used, this criterion is automatically satisfied already. The symmetry involved in the FFT algorithm makes it clear that the summation over the last $\frac{N}{2}$ complex frequency samples must be identical to that over the first $\frac{N}{2}$ samples, except for errors within the algorithm itself such as roundoff or truncation. This analysis will neglect such errors. In the actual implementation it will be explained later that all samples are utilized in an effort to minimize roundoff errors. For now, however, it will be expedient to consider minimizing the error computed from only $\frac{N}{2}$ samples. Hence the error expression to be minimized can be

$$\frac{1}{2} E = \sum_{i=1}^{\frac{N}{2}} \left| Y(f_i) - Z(f_i) \right|^2$$

which represents one half the total error power.

Consider some algebraic manipulations on the error expression as follows:

$$\frac{1}{2} E = \sum_{i=1}^{\frac{N}{2}} [Y(f_i) - Z(f_i)][Y(f_i) - Z(f_i)]^*$$

$$= \sum_{i=1}^{\frac{N}{2}} [Y(f_i)Y^*(f_i) + Z(f_i)Z^*(f_i) - Z^*(f_i)Y(f_i) - Z(f_i)Y^*(f_i)]$$

$$= \sum_{i=1}^{\frac{N}{2}} \{|Y(f_i)|^2 + |Z(f_i)|^2 - [Z^*(f_i)Y(f_i) + (Z^*(f_i)Y(f_i))^*]\}$$

$$= \sum_{i=1}^{\frac{N}{2}} |Y(f_i)|^2 + \sum_{i=1}^{\frac{N}{2}} |Z(f_i)|^2 - \sum_{i=1}^{\frac{N}{2}} \{2\mathrm{Re}[Z^*(f_i)Y(f_i)]\}$$

where Re[·] denotes the real part of a complex quantity. Thus, the error expression to be minimized can be written

$$\frac{1}{2} E = \sum_{i=1}^{\frac{N}{2}} \left| Y(f_i) \right|^2 + \sum_{i=1}^{\frac{N}{2}} \left| Z(f_i) \right|^2 - 2\mathrm{Re}\left[\sum_{i=1}^{\frac{N}{2}} Z^*(f_i)Y(f_i) \right].$$

Since $Z(f_i)$ is not directly available, it is necessary to substitute a representation of $Z(f_i)$ in terms of the reference $X(f_i)$. From the definition of z(t) given earlier, it is clear that $Z(f_i)$ should differ from $X(f_i)$ only by the frequency domain equivalents of group delay, phase delay, and gain. It follows from the definitions of group and phase delay [56] that, for the assumed linear phase characteristic, phase delay of the signal estimate z(t) can be adjusted by adding a constant phase angle to all frequency components. Also, the group delay of the estimate for the signal can be adjusted by adding a phase angle to each component which is directly proportional to the frequency of that component. Gain is adjusted by a scale factor, just as in the time domain. Therefore, the expression for $Z^*(f_i)$ for positive frequencies (the first $\frac{N}{2}$ samples from the FFT) can be written

$$Z^*(f_i) = AX^*(f_i)e^{j(af_i+b)} \tag{37}$$

where A, a, and b are real constants which can be chosen to minimize error. For negative frequencies the sign of b is reversed. Adjusting the parameter b is equivalent to adjusting the phase delay estimate in z(t). Varying the parameter a provides for adjustment of the group delay estimate.

Substituting (37) into the error expression to be minimized yields

$$\frac{1}{2}E = \sum_{i=1}^{\frac{N}{2}} \left|Y(f_i)\right|^2 + \sum_{i=1}^{\frac{N}{2}} \left|AX(f_i)e^{-j(af_i+b)}\right|^2$$

$$- 2\mathrm{Re}\left[\sum_{i=1}^{\frac{N}{2}} Y(f_i)AX^*(f_i)e^{j(af_i+b)}\right].$$

If E_2 is used to denote the minimized error expression,

$$E_2 = \underset{A,a,b}{\mathrm{MIN}}\left\{\sum_{i=1}^{\frac{N}{2}} \left|AX(f_i)e^{-j(af_i+b)}\right|^2 + \sum_{i=1}^{\frac{N}{2}} |Y(f_i)|^2 \right.$$

$$\left. - 2\mathrm{Re}\left[\sum_{i=1}^{\frac{N}{2}} Y(f_i)AX^*(f_i)e^{j(af_i+b)}\right]\right\}$$

or

$$E_2 = \underset{A,a,b}{\mathrm{MIN}}\left\{A^2 \sum_{i=1}^{\frac{N}{2}} |X(f_i)|^2 + \sum_{i=1}^{\frac{N}{2}} |Y(f_i)|^2 \right.$$

$$\left. - 2A\mathrm{Re}\left[\sum_{i=1}^{\frac{N}{2}} Y(f_i)X^*(f_i)e^{jaf_i}\, e^{jb}\right]\right\}. \qquad (38)$$

It should be remembered that E_2 equals one half the total mean-square error.

Clearly the first two terms in (38) are positive and independent of a and b. Thus, assuming that A is a positive scale factor, the optimal choices for a and b must be such that

$$\mathrm{Re}\left[\sum_{i=1}^{\frac{N}{2}} Y(f_i)X^*(f_i)\, e^{jaf_i}\, e^{jb}\right]$$

is maximized. Therefore, consider the expression

$$\underset{a,b}{\mathrm{MAX}}\left\{\mathrm{Re}\left[e^{jb}\left(\sum_{i=1}^{\frac{N}{2}} Y(f_i)X^*(f_i)e^{jaf_i}\right)\right]\right\}. \qquad (39)$$

Note that once the summation has been performed, the resulting sum will be some complex number, say C(a). Then (39) becomes

$$\underset{a,b}{\mathrm{MAX}}\{\mathrm{Re}[e^{jb}C(a)]\}.$$

If the angle associated with C(a) is denoted by θ, then $C(a) = |C(a)|e^{j\theta}$. Multiplication of C(a) by e^{jb} yields

$$e^{jb}C(a) = e^{j(b+\theta)}|C(a)|.$$

The vector represertation for C(a) and $e^{jb}\cdot C(a)$ are illustrated in Figure 8.

The angle of $[e^{jb}C(a)]$ may be adjusted to any desired value by varying b, and the magnitude, $|e^{jb}C(a)| = |C(a)|$, is independent of this adjustment. It is

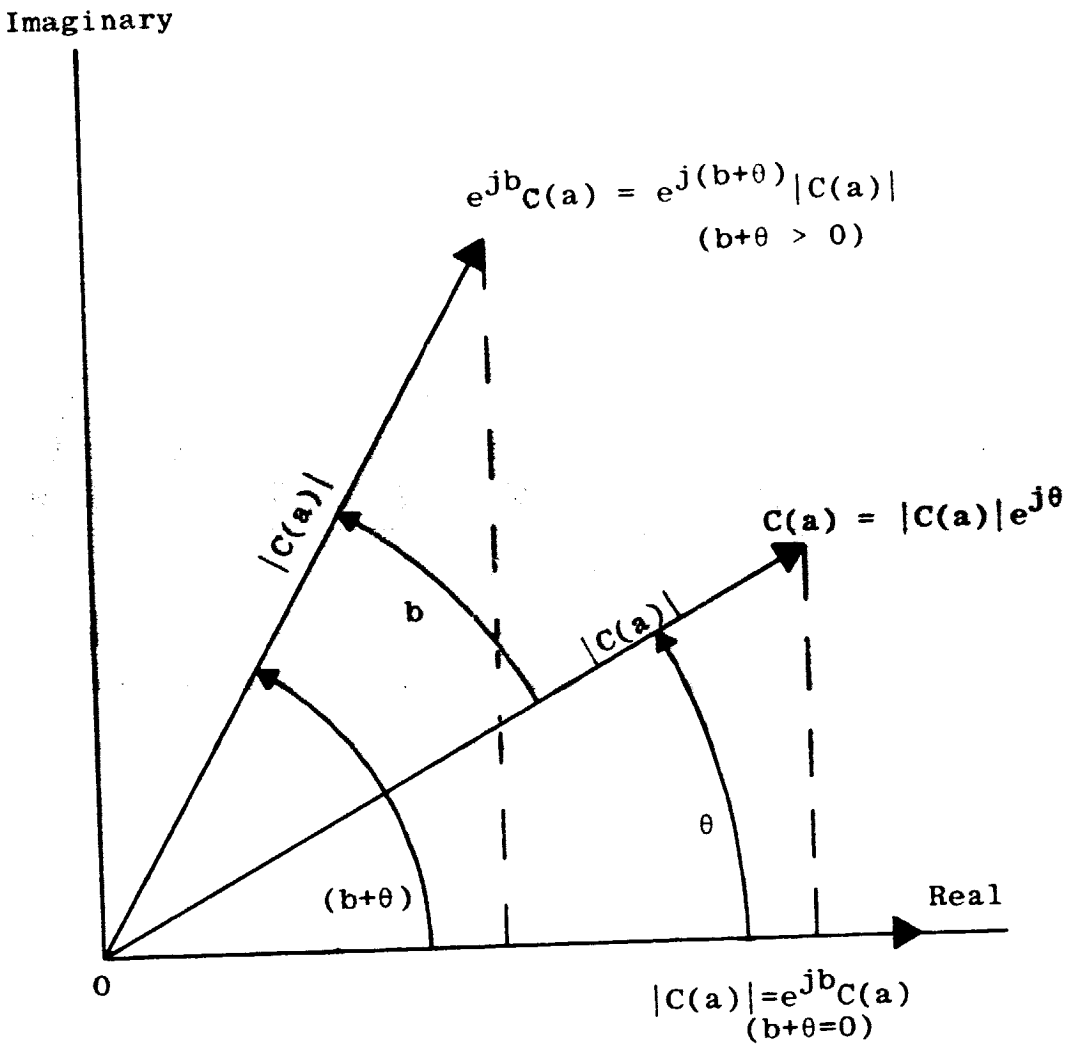

Figure 8. Estimation of the Group Delay Parameter, b

clear from the diagram that the real part of $[e^{jb}C(a)]$ is maximized when $b = -\theta$ so that $[e^{jb}C(a)]$ is real and positive. For this choice of b,

$$\mathrm{Re}[e^{jb}C(a)] = |C(a)|.$$

Therefore, the maximization over b always leads to

$$\underset{a,b}{\mathrm{MAX}}\{\mathrm{Re}[e^{jb}C(a)]\} = \underset{a}{\mathrm{MAX}}|C(a)|. \qquad (40)$$

Although this development is valid, it should probably be mentioned that the derivation of equation (40) is simplified somewhat by the decision which was made to utilize only $\frac{N}{2}$ samples in the theoretical development. If all the samples are used, the summation over N samples must be broken into two sums over the first and last $\frac{N}{2}$ samples, and discussions about the symmetry of the FFT are required. After some development, however, a criterion for choosing b can be obtained that is the same as (40).

Using the result given in (40), expression (39) may be written more simply as

$$\underset{a}{\mathrm{MAX}}\left|\sum_{i=1}^{\frac{N}{2}} Y(f_i)X^*(f_i)e^{jaf_i}\right|. \qquad (41)$$

With $a = 2\pi\tau$ the preceding expression may be rewritten as

$$\underset{\tau}{\text{MAX}}\left|\sum_{i=1}^{\frac{N}{2}} X^{*}(f_i)Y(f_i)e^{j2\pi f_i\tau}\right|$$

or

$$\underset{\tau}{\text{MAX}}\left|\sum_{i=1}^{\frac{N}{2}} S_{xy}(f_i)e^{j2\pi f_i\tau}\right| \qquad (42)$$

where $S_{xy}(f_i)$ is the ith sample of the cross-spectral density $[X^{*}(f)Y(f)]$. Let τ_m denote the choice of τ which maximizes (42). The the value of τ_m is the group delay estimate.

Once τ_m has been found, it is useful to define

$$R_2 = \left|\sum_{i=1}^{\frac{N}{2}} S_{xy}(f_i)e^{j2\pi f_i\tau_m}\right|. \qquad (43)$$

Then, (38) and (43) can be used to obtain

$$E_2 = \underset{A}{\text{MIN}}\left[A^2\sum_{i=1}^{\frac{N}{2}}|X(f_i)|^2+\sum_{i=1}^{\frac{N}{2}}|Y(f_i)|^2-2AR_2\right]. \qquad (44)$$

Denoting

$$\tfrac{1}{2}\text{ power in X} = P_X = \sum_{i=1}^{\frac{N}{2}}|X(f_i)|^2$$

and

$$\tfrac{1}{2}\text{ power in Y} = P_Y = \sum_{i=1}^{\frac{N}{2}}|Y(f_i)|^2$$

provides the error expression

$$E_2 = \underset{A}{MIN}[A^2P_X+P_Y-2AR_2].$$

To find the value of A that minimizes E_2, the derivative with respect to A is computed and set equal to zero. This yields

$$\frac{dE_2}{dA} = \frac{d}{dA}\,[A^2P_X+P_Y-2AR_2]=0.$$

Using A_m to denote the optimal choice for A,

$$2A_mP_X-2R_2=0.$$

Thus,

$$A_m = \frac{R_2}{P_X} = \text{System Gain.} \qquad (45)$$

Substitution of (45) into the error expression yields

$$E_2 = A_m^2\,P_X+P_Y-2A_mR_2$$

$$= P_Y - \frac{R_2^2}{P_X}$$

or

$$E_2 = P_Y-A_m^2P_X. \qquad (46)$$

The signal power in the measurement data is clearly

$$S = A_m^2(2P_X). \tag{47}$$

From (46) and (47), and remembering that E_2 represents one half the total error power, the signal-to-noise ratio is given by

$$SNR = \frac{S}{2E_2} = \frac{\left(\frac{S}{2}\right)}{E_2} = \frac{A_m^2 P_X}{P_Y - A_m^2 P_X}$$

or in terms of R_2, P_X, and P_Y,

$$SNR = \frac{R_2^2}{P_X P_Y - R_2^2} \tag{48}$$

where

$$P_X = \sum_{i=1}^{\frac{N}{2}} |X(f_i)|^2$$

$$P_Y = \sum_{i=1}^{\frac{N}{2}} |Y(f_i)|^2$$

and

$$R_2 = \left| \sum_{i=1}^{\frac{N}{2}} X^*(f_i)Y(f_i)\, e^{j2\pi f_i \tau_m} \right|.$$

Estimates for various system parameters that are available in terms of these quantities are tabulated for reference in Table II.

As in the lowpass case, it can be shown that the bandpass SNR estimator provides the ML estimate for the additive white Gaussian noise channel. In this case a frequency domain approach is used. It is necessary to determine the joint density of the frequency domain samples, $Y(f_i)$, obtained by the FFT, in terms of the gain A, the group delay parameter a, and the phase delay parameter b.

First note that for the additive white Gaussian noise channel $y(t) = z(t) + n(t)$. Therefore, the FFT yields

$$Y(f_i) = Z(f_i) + N(f_i) \tag{49}$$

where $Y(f_i)$ denotes the ith frequency component of the waveform under test, $y(t)$, and $Z(f_i)$ and $N(f_i)$ represent estimates for the signal and noise components, respectively. The discrete transformation of the noise may be written

$$N(f_i) = \frac{1}{N} \sum_{k=0}^{N-1} n(t_k)\, e^{-j2\pi f_i k/N}$$

or

$$N(f_i) = \frac{1}{N} \sum_{k=0}^{N-1} n(t_k)\cos\left(2\pi f_i \frac{k}{N}\right)$$

$$- j \frac{1}{N} \sum_{k=0}^{N-1} n(t_k)\sin\left(2\pi f_i \frac{k}{N}\right)$$

TABLE II

PARAMETER ESTIMATES FOR BANDPASS SYSTEMS

IN TERMS OF P_X, P_Y, τ_m, AND R_2

Parameter	Estimate
Gain	$A_m = \frac{R_2}{P_X}$
Group Delay	τ_m (as defined in text)
Signal power at the point of measurement	$S = \frac{2R_2^2}{P_X}$
Mean-square Error	$E = 2\left[P_Y - \frac{R_2^2}{P_X}\right]$
Normalized Mean-square Error	$E_N = 1 - \frac{R_2^2}{P_X P_Y}$
Signal-to-Noise Ratio	$SNR = \frac{R_2^2}{P_X P_Y - R_2^2}$

which is

$$N(f_i) = N_R(f_i) - j\, N_I(f_i) \tag{50}$$

where

$$N_R(f_i) = \frac{1}{N} \sum_{k=0}^{N-1} n(t_k)\cos(2\pi f_i \frac{k}{N})$$

and

$$N_I(f_i) = \frac{1}{N} \sum_{k=0}^{N-1} n(t_k)\sin(2\pi f_i \frac{k}{N}).$$

The parameter N represents the total number of points being transformed.

Since $N_R(f_i)$ and $N_I(f_i)$ are linear combinations of Gaussian random variables, they must themselves be Gaussian distributed. Also, they represent the direct and quadrature components of the ith spectral component and are uncorrelated [57]. Thus for any given i, $N_R(f_i)$ and $N_I(f_i)$ are independent Gaussian random variables. Further, since spectral components of different (non-overlapping) frequencies are uncorrelated, it follows for any ergodic process that $N_R(f_i)$ and $N_R(f_j)$ must be uncorrelated for any $i \neq j$ where the FFT gives a valid representation of the spectral components in the frequency domain [58]. The same argument holds for $N_I(f_i)$ and $N_I(f_j)$. The FFT provides a valid representation of spectral components for frequencies up to one half the sampling frequency, if properly applied. This corresponds to the first half of all the computed frequency domain samples.

For real signals, which are the only ones considered here, the latter $\frac{N}{2}$ complex samples produced by the FFT are the complex conjugates of the first $\frac{N}{2}$ samples [59]. This deterministic symmetry makes it clear that no additional information is gained in using the second half of the samples. Therefore, the analysis continues based upon only $\frac{N}{2}$ complex samples.

Uncorrelated Gaussian random variables are independent. Therefore, the first half of all the noise samples computed by the FFT, $N_R(f_i)$ and $N_I(f_i)$, are independent.

It is useful to extend the R and I subscript notation to include $Y(f_i)$ and $Z(f_i)$ so that

$$Y_R(f_i) = Z_R(f_i) + N_R(f_i) \qquad \text{(real components)}$$

and

$$Y_I(f_i) = Z_I(f_i) + N_I(f_i) \qquad \text{(imaginary components).}$$

Then, the joint density of the first $\frac{N}{2}$ real sample values and $\frac{N}{2}$ imaginary sample values can be written as

$$p_{\bar{Y}}(\bar{Y}|A,a,b) = \prod_{i=1}^{\frac{N}{2}} p_i(Y_R(f_i)|A,a,b)p_i(Y_I(f_i)|A,a,b)$$

where $p_i(Y_R(f_i)|A,a,b)$ and $p_i(Y_I(f_i)|A,a,b)$ are the Gaussian densities for the ith real and imaginary sample

values, respectively. Since $N(f_i)$ is zero mean, the means of $Y_R(f_i)$ and $Y_I(f_i)$ are given by $Z_R(f_i)$ and $Z_I(f_i)$. Under the white noise assumption, the noise variance will be the same for each $N_R(f_i)$ and $N_I(f_i)$ and will be denoted by σ_N^2. Admittedly, this is not generally the case, since filtering is usually present which scales noise differently at different frequencies. However, an exact analysis would be system dependent and a general solution does not appear possible.

From (37) it is seen that

$$Z_R(f_i) = A[X_R(f_i)\cos(af_i+b)+X_I(f_i)\sin(af_i+b)]$$
$$\text{and} \qquad (51)$$
$$Z_I(f_i) = A[X_I(f_i)\cos(af_i+b)-X_R(f_i)\sin(af_i+b)].$$

Using these results and assumptions, the joint density becomes

$$p_{\bar{Y}}(\bar{Y}|A,a,b) = \prod_{i=1}^{\frac{N}{2}} \frac{1}{2\pi\sigma_N^2}$$
$$\cdot\exp\left\{-\left[\frac{Y_R(f_i)-A[X_R(f_i)\cos(af_i+b)+X_I(f_i)\sin(af_i+b)]}{\sqrt{2}\,\sigma_N}\right]^2\right.$$
$$\left.-\left[\frac{Y_I(f_i)-A[X_I(f_i)\cos(af_i+b)-X_R(f_i)\sin(af_i+b)]}{\sqrt{2}\,\sigma_N}\right]^2\right\}$$

This density, which is the likelihood function for A, a, and b can be rewritten after some algebraic manipulation as

$$P_{\bar{Y}}(\bar{Y}|A,a,b) =$$

$$(2\pi\sigma_N^2)^{-\frac{N}{2}} \exp\left\{\frac{\left[-\sum_{i=1}^{\frac{N}{2}}\left[Y_R^2(f_i)+Y_I^2(f_i)\right]-A^2\sum_{i=1}^{\frac{N}{2}}\left[X_R^2(f_i)+X_I^2(f_i)\right]\right]}{2\sigma_N^2}\right.$$
$$\left.\frac{+2A\mathrm{Re}\left[\sum_{i=1}^{\frac{N}{2}} X^*(f_i)Y(f_i)e^{j(af_i+b)}\right]}{2\sigma_N^2}\right\} \quad (52)$$

The log-likelihood function is

$$\frac{1}{2\sigma_N^2}\left\{\begin{array}{l} 2A\mathrm{Re}\left[\sum_{i=1}^{\frac{N}{2}} X^*(f_i)Y(f_i)e^{j(af_i+b)}\right] \\ -\sum_{i=1}^{\frac{N}{2}}\left[Y_R^2(f_i)+Y_I^2(f_i)\right]-A^2\sum_{i=1}^{\frac{N}{2}}\left[X_R^2(f_i)+X_I^2(f_i)\right] \end{array}\right\} -\frac{N}{2}\ln(2\pi\sigma_N^2) \quad (53)$$

By definition, ML estimates for the parameters a and b are those values which maximize this function. Assuming that A is any positive scale factor and σ_N^2 is any positive value, the optimal choice for a and b must be such that the expression

$$\mathrm{Re}\left[\sum_{i=1}^{\frac{N}{2}} X^{*}(f_i)Y(f_i)e^{j(af_i+b)}\right] \tag{54}$$

is maximized. Note that this is true regardless of the values of A and σ_N^2, which will be variables in determining signal and noise estimates later. This leads immediately to the criterion expressed in equation (39), and thus it is seen that the estimates formed for parameters a and b are ML estimates. Parameters a and b are directly proportional to the group and phase delay estimates, respectively. Thus the group and phase delay estimates obtained with the minimum mean-square error criterion are ML estimates for the additive white Gaussian noise case.

Once a and b have been chosen, expression (54) becomes a constant. Denoting this constant by R_1, the log-likelihood function becomes

$$\frac{1}{2\sigma_N^2}\left[2AR_1 - \sum_{i=1}^{\frac{N}{2}} [Y_R^2(f_i)+Y_I^2(f_i)] - A^2 \sum_{i=1}^{\frac{N}{2}} [X_R^2(f_i)+X_I^2(f_i)]\right]$$

$$- \frac{N}{2} \ln(2\pi\sigma_N^2).$$

At this point it is convenient to drop the subscripted form of X and Y. Rewriting in terms of the original complex form yields the log likelihood function

$$\frac{1}{2\sigma_N^2}\left[2AR_1-\sum_{i=1}^{\frac{N}{2}}|Y(f_i)|^2-A^2\sum_{i=1}^{\frac{N}{2}}|X(f_i)|^2\right]-\frac{N}{2}\ln(2\pi\sigma_N^2).$$

From (43) and its earlier equivalent form, (39), and the definition of R_1, expression (54) maximized over a and b, it is seen that $R_1 = R_2$. Using this result and the definitions for P_X and P_Y given following (44), the log-likelihood function can be written

$$\frac{1}{2\sigma_N^2}\left[2AR_2-P_Y-A^2P_X\right]-\frac{N}{2}\ln(2\pi\sigma_N^2).$$

Just as in the lowpass case, the ML estimate for the SNR is formed as the ratio of the ML estimates for the signal power and for the noise power. As a convenience in using the notation developed for the bandpass case, the estimates will actually be formed for one half the signal power and one half the noise power; obviously the ratio is unchanged. One half the signal power is given by A^2P_X. Defining $S_2=A^2P_X$, remembering that A is assumed to be positive, and rewriting the log-likelihood function yields

$$\frac{1}{2\sigma_N^2}\left[2\sqrt{S_2}\,\frac{R_2}{\sqrt{P_X}}-P_Y-S_2\right]-\frac{N}{2}\ln(2\pi\sigma_N^2).$$

One half the noise power is given by $(N\,\sigma_N^2)$ so that the log-likelihood function can be written

$$\frac{N}{2(N\sigma_N^2)}\left[\frac{2R_2}{\sqrt{P_X}}\sqrt{S_2} - P_Y - S_2\right] - \frac{N}{2}\ln\left[\frac{2\pi}{N}(N\,\sigma_N^2)\right]. \qquad (55)$$

The ML estimates for S_2 and $(N\,\sigma_N^2)$ can be found by setting the partial derivatives with respect to S_2 and $(N\,\sigma_N^2)$ equal to zero and solving. For the signal power, the partial derivative of (55) yields

$$\frac{N}{2(N\,\sigma_N^2)}\left[\frac{R_2}{\sqrt{P_X}}\cdot\frac{1}{\sqrt{\hat{S}_2}} - 1\right] = 0$$

or

$$\hat{S}_2 = \frac{R_2^2}{P_X}\,. \qquad (56)$$

For the noise power, the partial derivative of (55) yields

$$-\left(\frac{N}{2}\right)\left[\frac{2R_2}{\sqrt{P_X}}\sqrt{\hat{S}_2} - P_Y - \hat{S}_2\right]\frac{1}{(N\,\hat{\sigma}_N^2)^2} - \frac{N}{2}\cdot\frac{1}{(N\,\hat{\sigma}_N^2)} = 0.$$

Substituting from (56) and simplifying,

$$\left[\frac{2R_2^2}{P_X} - P_Y - \frac{R_2^2}{P_X}\right]\cdot\frac{1}{(N\,\hat{\sigma}_N^2)} + 1 = 0$$

or

$$(N\ \hat{\sigma}_N^2) = P_Y - \frac{R_2^2}{P_X} \quad . \tag{57}$$

As explained in the development for the lowpass case, the ML estimate for the SNR is simply the ratio of the ML estimates for the signal power and the noise power. Thus

$$\hat{SNR} = \frac{2\hat{S}_2}{2(N\ \hat{\sigma}_N^2)} = \frac{\hat{S}_2}{(N\ \hat{\sigma}_N^2)} = \frac{\left(\frac{R_2^2}{P_X}\right)}{P_Y - \left(\frac{R_2^2}{P_X}\right)}$$

or

$$\hat{SNR} = \frac{R_2^2}{P_X P_Y - R_2^2} \quad . \tag{58}$$

The estimate given by (58) is precisely that of (48). Thus for the white additive Gaussian noise case, the estimator developed is the ML estimator.

Unfortunately the development of confidence intervals for the bandpass case does not appear to be mathematically feasible. The computation of the quantity R_2^2 involves the sum of the squares of two Gaussian random variables which are not easily shown to be independent. This may be seen by considering

$$R_2^2 = \left| \sum_{i=1}^{\frac{N}{2}} X^*(f_i) Y(f_i) e^{j2\pi f_i \tau m} \right|^2$$

$$= \left| \sum_{i=1}^{\frac{N}{2}} [X_R(f_i) - jX_I(f_i)][Y_R(f_i) + jY_I(f_i)][\cos(2\pi f_i \tau_m) + j \sin(2\pi f_i \tau_m)] \right|^2$$

$$= \left\{ \sum_{i=1}^{\frac{N}{2}} \left[Y_R(f_i)[X_R(f_i)\cos(2\pi f_i \tau_m) + X_I(f_i)\sin(2\pi f_i \tau_m)] - Y_i(f_i)[X_R(f_i)\sin(2\pi f_i \tau_m) - X_I(f_i)\cos(2\pi f_i \tau_m)] \right] \right\}^2$$

$$+ \left\{ \sum_{i=1}^{\frac{N}{2}} \left[Y_R(f_i)[X_R(f_i)\sin(2\pi f_i \tau_m) - X_I(f_i)\cos(2\pi f_i \tau_m)] + Y_I(f_i)[X_R(f_i)\cos(2\pi f_i \tau_m) + X_I(f_i)\sin(2\pi f_i \tau_m)] \right] \right\}^2 .$$

An inspection of the various terms within braces indicates that it may be possible to show the individual terms within the first squared quantity to be independent of the individual terms within the second squared quantity. However, this is insufficient to guarantee independence of the complete quantities. If independence were assumed in order to continue the analysis, the distribution of R_2^2 would be non-central chi-square. It does not appear that R_2^2 is the square of a weighted sample mean of a Gaussian population. This means that the denominator of (58) cannot easily be shown to be the sample variance

of a Gaussian population. Hence the distribution for the denominator of (58) does not follow easily, as was the case with the lowpass estimator. Finally, it is not clear that the numerator and denominator in equation (58) are independent. Thus, determination of the distribution function for the bandpass SNR estimator appears to be a very formidable problem. It may be possible to form an approximation to the distribution function, however. The estimate has been shown to be the ML estimate for a white additive Gaussian noise channel. It can be shown that under reasonably general conditions, the ML estimate is asymptotically Gaussian as the number of samples used approaches infinity [60]. Thus, one approach might be to develop a Gaussian approximation, especially since the cases of interest generally involve at least 1000 samples. The difficulty with obtaining the approximation arises in attempting to determine the variance of $\hat{(SNR)}$, which is necessary for forming the Gaussian density. This does not appear to be a simple matter. Therefore, confidence intervals were not derived in this work for the bandpass estimator.

Confidence intervals would be convenient in order to precisely define the reliability of the estimator, at least for the additive white Gaussian noise channel. A deeper statistical study of the reliability of the estimators is an area in which further work could certainly be done. However, a good case for assuming adequate

reliability of the bandpass estimation routine can be made based upon empirical data. Results of tests of this estimator appear later in the section on applications to analog systems and throughout the section on applications to digital systems. Excellent reliability is evident in these results.

B. METHOD OF IMPLEMENTATION

In order to estimate the SNR and other parameters of interest in simulations of bandpass systems, it is convenient to first find estimates for P_X, P_Y, and R_2. As can be seen from Table II, a variety of system parameters, including the SNR, can be computed in terms of these quantities.

From the definitions given for P_X, P_Y, and R_2 following (48), it is evident that the computation of the required quantities is done after the FFT has been applied to the reference data, x(t), and the measurement data, y(t). A conceptually direct method for computing P_X, P_Y, and R_2 is shown in Figure 9. However, as in the lowpass case, the most straightforward method of implementation is not practical because it requires that the complete summation over all complex samples be repeated for every possible value of τ in order to find τ_m and thus determine R_2. This operation requires a great deal more computer time than is practical using the number of samples generated in a typical simulation. In order to make the SNR estimation routine practical, some method must be found for rapidly determing τ_m.

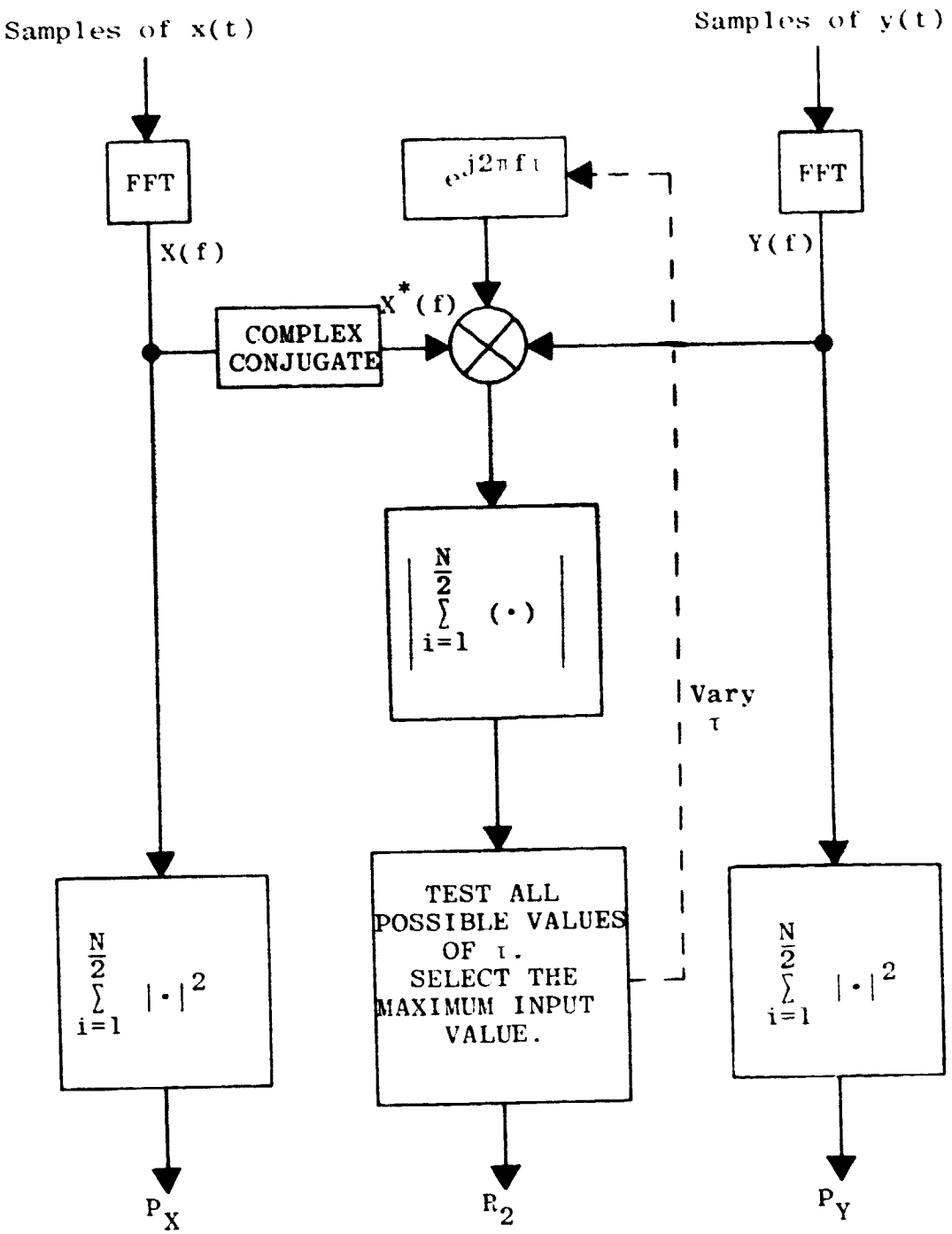

Figure 9. Direct Computation of P_X, P_Y, and R_2

Consider the quantity to be maximized in locating τ_m. From (42) and (43)

$$R_2 = \underset{\tau}{\text{MAX}} \left| \sum_{i=1}^{\frac{N}{2}} S_{xy}(f_i) e^{j2\pi f_i \tau} \right|$$

where $S_{xy}(f_i) = |X^*(f_i)Y(f_i)|$ and τ_m is the value of τ for which maximization is achieved. Consider the complex quantity generated by the summation and define

$$R_A(\tau) = \sum_{i=1}^{\frac{N}{2}} S_{xy}(f_i)\, e^{j2\pi f_i \tau}.$$

Also, make the definition

$$S_A(f_i) = S_{xy}(f_i) \qquad \text{for } 1 \le i \le \frac{N}{2}$$

$$= 0 \qquad \text{for } (\frac{N}{2} + 1) \le i \le N.$$

This yields

$$R_A(\tau) = \sum_{i=1}^{N} S_A(f_i)\, e^{j2\pi f_i \tau}. \tag{59}$$

Let τ be chosen to be some value of time equal to an integral multiple of the time interval between samples so that

$$\tau = kT, \qquad 0 \leq k \leq N-1$$

where T is the time between samples. With this restriction $R_A(\tau)$ is, by definition [61], the inverse discrete Fourier transform of $S_A(f)$. The frequency domain samples for f_i with $(\frac{N}{2} + 1) \leq i \leq N$ can be viewed as samples from the negative frequency range [62]. Thus, $R_A(\tau)$ may be viewed as the inverse Fourier transform of $S_{xy}(f)$ with the negative frequency components set to zero. Remembering that $S_{xy}(f)$ is the Fourier transform of $R_{xy}(\tau)$, it is seen that $R_A(\tau)$ is the function which has a spectrum identical to $R_{xy}(\tau)$ for positive frequencies and equal to zero for negative frequencies. It follows immediately from the definition of analytic signals [63] that $R_A(\tau)$ is proportional to the analytic signal corresponding to the real "signal" $R_{xy}(\tau)$. Then

$$R_2 = \underset{\tau}{\text{MAX}} \; |R_A(\tau)|$$

and τ_m is the value of τ for which the magnitude of the analytic signal corresponding to $R_{xy}(\tau)$ is maximized. It is possible to generate this analytic signal quite rapidly with the FFT. This approach provides the necessary technique for estimating τ_m quickly.

It is possible to obtain the desired analytic signal directly from x(t) and y(t) using the FFT. This can be accomplished by forming $[2X^*(f)Y(f)]$ and setting

the last $\frac{N}{2}$ sample values (corresponding to negative frequency samples) to zero before inverse transforming. An alternative approach is to compute $R_{xy}(\tau)$, which is the real component of the analytic signal, just as in the lowpass estimator and to find the imaginary component of the analytic signal by computing the Hilbert transform of $R_{xy}(\tau)$. Computation of the imaginary component may be done with a discrete Hilbert transform algorithm or with the FFT, using the simple frequency domain equivalent of the Hilbert transform [64], [65]. Since the software to compute and store $R_{xy}(\tau)$ had been written and thoroughly tested in the lowpass SNR estimator, the latter approach for computing the analytic signal was used, and the Hilbert transform was performed with the FFT.

Instead of computing the Hilbert transform and determining the analytic signal for the entire range of $R_{xy}(\tau)$, a preliminary scan of $R_{xy}(\tau)$ is performed, and the peak value of $R_{xy}(\tau)$ is found. It is assumed that although the maximum magnitude of the analytic signal does not necessarily correspond to the peak of $R_{xy}(\tau)$, it is highly probable, using the slowly varying envelope concept, that the two maxima occur in the same neighborhood. Also, it was experimentally determined that approximately twenty well behaved artificial sample values added at each end of the FFT transform block

helps to avoid any problems with errors resulting from the discontinuities which may exist there.

The twenty artificial values at the beginning of the transform block were obtained by taking the first twenty true sample values in reverse order. For example, suppose the artificial samples are denoted by YT(i), i=1,...,20, and the true data values follow, beginning with YT(21), YT(22), and so on. The values of the artificial samples are assigned using YT(21-i) = YT(20+i), for i = 1, . . ., 20. A similar procedure is utilized to develop the artificial sample values at the end of the transform block using the last twenty true data values. Experimental testing of this scheme showed that good results were obtained for a variety of signals, so the method was adopted for use in SNRBPS.

Therefore, in the implementation used to determine group delay, the assumption is made that the maximum magnitude of the analytic signal and the peak value of $R_{xy}(\tau)$ are separated by a delay value corresponding to no more than $\pm$ ($\frac{N}{2}$ - 20) samples, where N is the number of samples chosen for the FFT block size. Under this assumption only one block of N samples needs to be processed by the Hilbert transformation procedure, since the analytic signal is required only on the interval extending $\pm$ ($\frac{N}{2}$ - 20) samples from the delay value where the peak of $R_{xy}(\tau)$ is located.

This assumption saves computation time and avoids the problems associated with end effects that result in attempting to Hilbert transform many separate blocks of data to produce a longer transformed array. If the signals involved are not narrowband, it is possible that the peak of $R_{xy}(\tau)$ could be sufficiently removed from the desired delay value that the assumption of a $\pm$ ($\frac{N}{2}$ - 20) sample range could result in an error. This never occurred in any of the applications of the bandpass SNR estimation routine, however, and the assumption appears to be justified by the resulting simplifications and reduction in computer time requirements. The possibility of error due to this assumption could, of course, be eliminated by Hilbert transforming the entire $R_{xy}(\tau)$ array or by computing the analytic signal for the entire $R_{xy}(\tau)$ array directly as mentioned earlier, if it were felt to be necessary.

As with the lowpass SNR estimation routine, the elaborate processing performed to initially estimate delay serves only to determine a starting point for the final search. The successive Fourier transformations used to obtain $R_{xy}(\tau)$, and to implement the Hilbert transform, reduce the accuracy of the initial R_2 estimate. Also, as in the lowpass case, the original $R_{xy}(\tau)$ computation for any given τ value is based upon only $\frac{N}{2}$ samples from the x(t) and y(t) arrays. As a result, the initial estimate obtained for R_2 is unsuitable for forming an accurate estimate of the SNR.

A number of iterations are performed using a more direct and accurate computation to form trial values for R_2, denoted R_2', in the vicinity of the initial delay estimate. The number of points to be checked in the final iterations is specified by an input parameter to the routine. Each of these computations is performed by first shifting the x(t) data to account for the delay value being considered and thus the exponential factor in the defining sum for R_2. The delayed data are denoted by $x_d(t)$. Then the FFT is applied to data blocks from $x_d(t) = x(t-\tau)$ and y(t) to form $X_d(f_i) = X(f_i)e^{-j2\pi f_i \tau}$ and $Y(f_i)$, respectively. Finally, complex conjugates are computed, multiplication of the sample pairs is accomplished, and the summation is performed. The magnitude of the resulting sum is computed at each iteration and is normalized by $\sqrt{P_x P_y}$, where P_x and P_y are the powers computed from the same frequency domain samples that were used in computing R_2'. The peak of this normalized cross-spectral power computation defines τ_m and R_2 for use in the final SNR estimate.

The reason for shifting the x(t) data to account for the $e^{-j2\pi f\tau}$ factor before transforming instead of including such a factor in the frequency domain calculations for R_2 is to avoid problems which occur because of the periodicity inherent in the FFT. Performing the frequency domain equivalent of time-shifting with a finite block length yields errors equivalent in the time domain to

overlapping the beginning of the block of data from one array with the end of a data block from another array in a cyclic fashion. This problem can be overcome by padding one array with zeroes before transforming as was done in initially computing $R_{xy}(\tau)$ in both SNRMSE and SNRBPS. This still leaves some problems with high frequency components generated by the discontinuities which exist after abruptly setting the waveform to zero at some point. Unlike the initial $R_{xy}(\tau)$ estimation, the result of the final R_2 computation is used directly in the estimation of the SNR. Therefore, maximum possible accuracy is necessary. Time-shifting the array before transforming avoids most of the problems. Although slightly more complicated in concept, this approach is simpler in implementation and more accurate.

Actually, the basic quantities which are computed in the final implementation are $(2P_X)$, $(2P_Y)$, and $(2R_2)$. These computations result from the fact that P_X, P_Y, and R_2 have been defined as summations over only the first half of the complex frequency domain samples computed by each application of the FFT. The symmetry which theoretically occurs when the FFT is applied to real data indicates that including the second half of the complex samples would yield identical results to those obtained using only the first half. In practice, this is not quite true because of roundoff errors which occur within the algorithm due to finite register length. It seems

unlikely that roundoff errors which appear in the second half of the samples computed by the FFT would be identical to those in the first half. Also, it is logical to assume that the mean-square value of the error introduced by roundoff is roughly the same for samples in both halves of the computed array. If these assumptions are true, the effects of roundoff error should be reduced by utilizing all the complex frequency domain samples, instead of only half of them. The additional samples are readily available and the additional time required to utilize them is not a significant factor in the overall computer time requirements of the estimation routine. Therefore, in an effort to minimize the effects of roundoff errors, all samples computed by the FFT are utilized. Summations over all the samples produce the quantities $(2P_X)$, $(2P_Y)$, and $(2R_2)$.

In performing the final computations, a data window is applied before Fourier transforming the data to smooth the frequency domain estimates, if specified using an input parameter to the routine. A variety of data windows were tested for this purpose. It was found that the application of data windows often improves the SNR estimates in cases where the SNR is relatively high. The Hanning window appeared to yield the best results and was chosen for use in the SNR estimation routine. Some results of the tests of the various data windows are tabulated in Appendix B.

A block diagram for the bandpass SNR estimation routine, SNRBPS, is given in Figures 10 and 11. Figure 10 gives an overall view of the routine. Unless the group delay is known and is input to the routine, the first step required is to find an initial estimate for τ_m, denoted by τ_m'. Then x(t) is delayed and blocks of data from $x_d(t)$ and y(t) are transformed, windowing if specified. The power in x, P_x, is computed as shown, averaging over all the blocks of data processed. The total power, P_x, is twice the value of P_X defined by (48), and the power in y, P_y, is similarly found. Using delay values in the vicinity of τ_m', computations of $(2R_2')$ normalized by $\sqrt{P_x P_y}$ are performed for the number of iterations specified by input parameter ILROAM. The maximum value obtained in these computations is found, and RTMAX is defined as twice R_2 to simplify the notation. Finally, computations are done for estimating the SNR and for ERRN, the normalized mean-square error. All the final computations except the FFT are done using double precision arithmetic.

Figure 11 provides a view of the method used to form the initial group delay estimate, τ_m'. The process involves first finding the peak value of the cross-correlation function $R_{xy}(\tau)$. This process is identical to that described in the discussion of the lowpass SNR estimator. Once the delay value associated with this peak, τ_{MAX}, is found, a block of data centered about τ_{MAX} is taken from $R_{xy}(\tau)$. On this interval the Hilbert

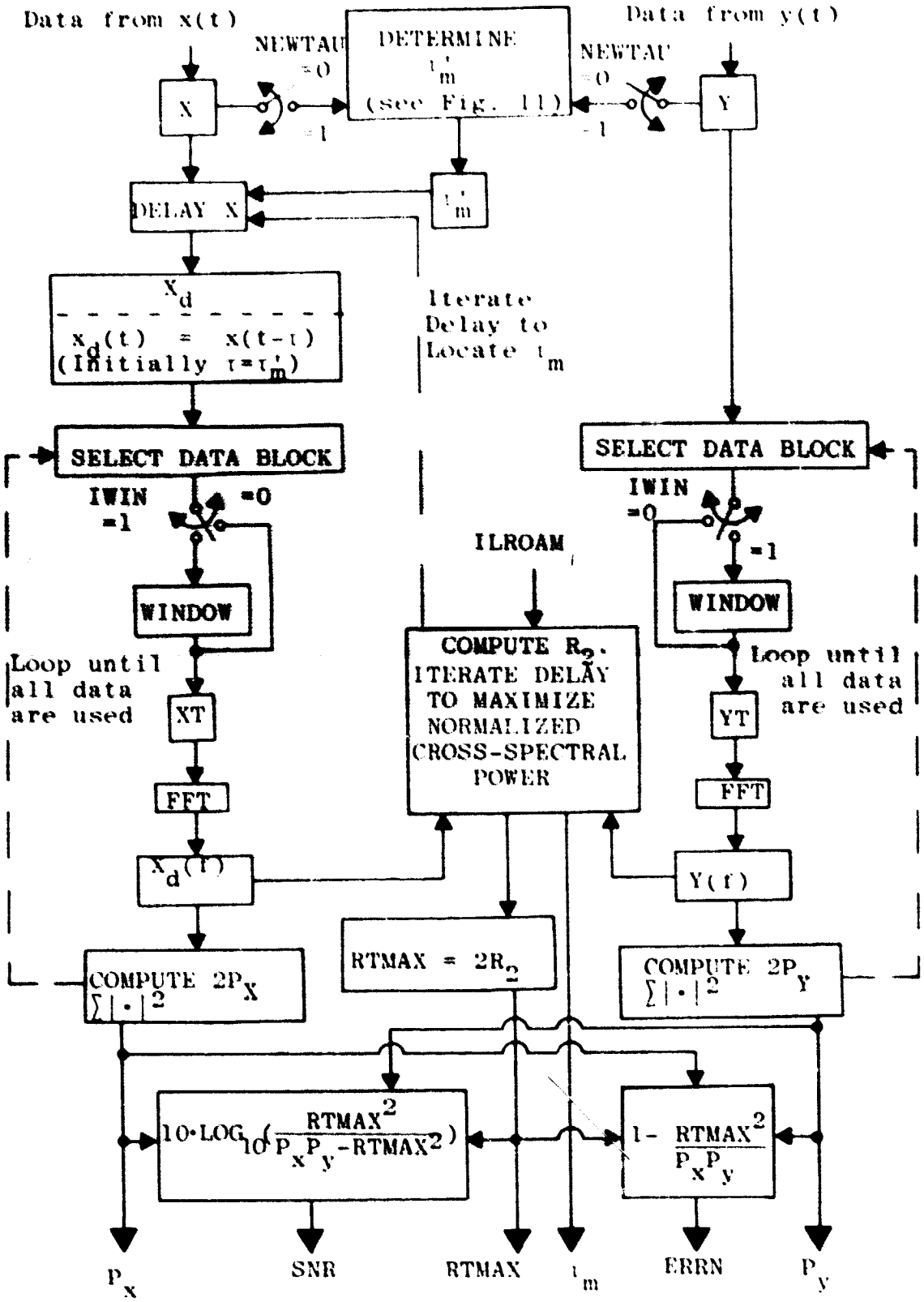

Figure 10. Block Diagram of SNRBPS

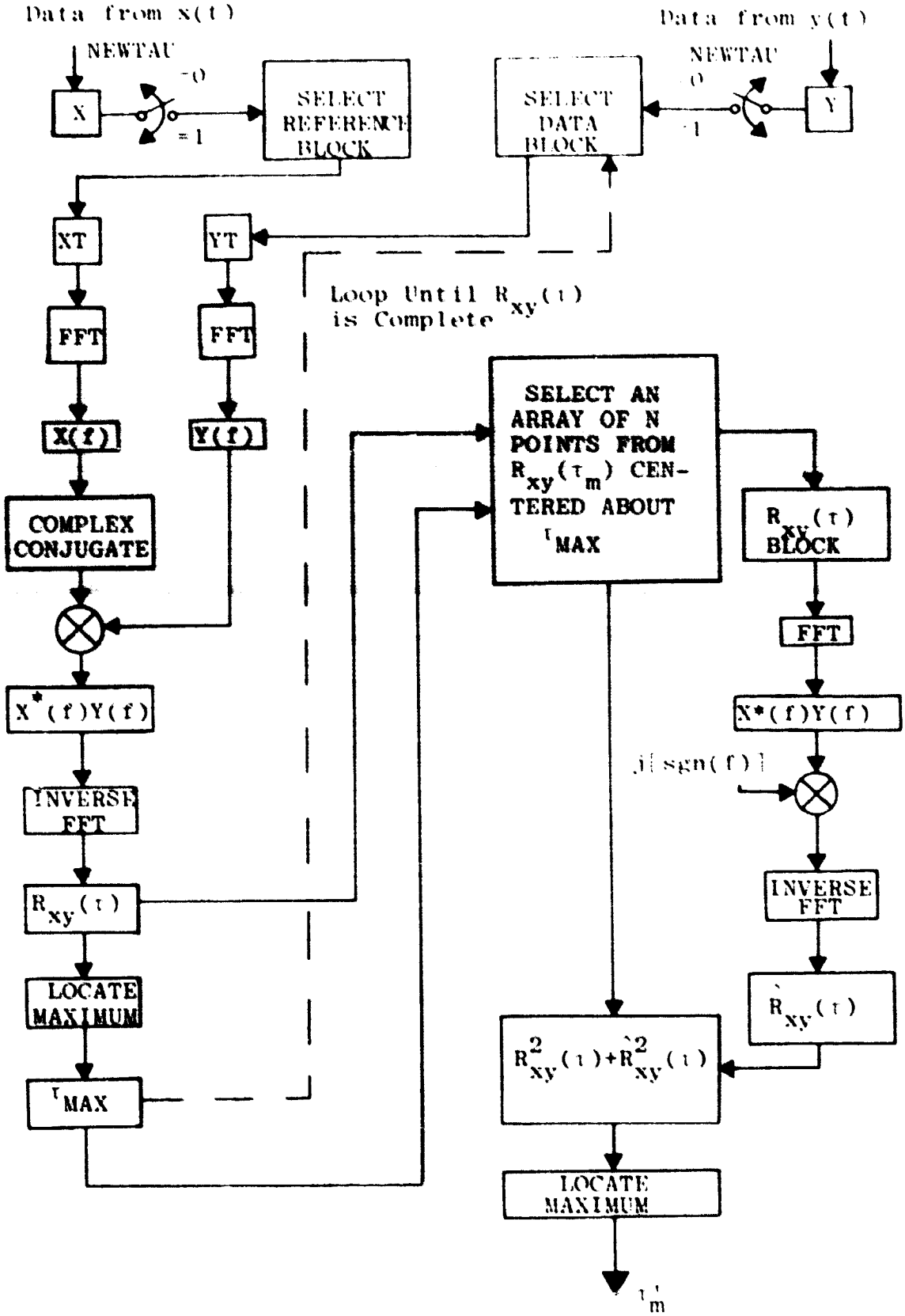

Figure 11. Algorithm to Find Initial Delay Estimate

transform is computed using the FFT. Then the square of the magnitude of the analytic signal corresponding to $R_{xy}(\tau)$ is computed for the interval. The delay value corresponding to the peak of the squared magnitude is the initial estimate for group delay, τ_m. This value is then used as a starting point for the delay iterations performed in the final computations. A listing of the computer subroutine SNRBPS is given in Appendix C.

All the considerations mentioned in the discussion of the lowpass SNR estimation routine concerning the total number of data points used and the choice of the FFT block length apply as well to the routine for SNR estimation in simulations of bandpass systems. Larger sets of data generally result in more accurate estimates for the SNR. Errors within the FFT increase as block size increases. However, the use of very short blocks can cause problems with the stationarity assumptions with the initial group delay estimate. Results of tests using various block lengths are given in the section on applications to analog systems.

VI. APPLICATION TO ANALOG SYSTEMS

In order to study the accuracy, reliability, and range of the SNR estimation routines, computer simulations were performed for a number of analog systems. Initially, test signals were combined with filtered noise, and the SNR estimation routines were applied to both lowpass and bandpass systems. Additive bandlimited white Gaussian noise was employed, and delays were programmed into the data arrays in many cases to check the ability of the routines to correctly estimate the values of these delays. The test signals used included simple sinusoidal signals and more complex waveforms with Gaussian distributed amplitudes. Tests were then made on a more complicated simulation of an FM communication system with phase-locked loop demodulation.

In all the simulations performed for both analog and digital communication systems, filter models were required. Therefore, a general lowpass and bandpass Butterworth digital filter with variable order and critical frequencies was programmed for use in the simulations which follow. Some preliminary tests of the digital filter, as described in the following two paragraphs, were necessary before all of the results could be properly evaluated.

To accurately determine the theoretical values of the SNR in the simulations, it is sometimes necessary to

compute the noise power at the output of a digital filter using measurements taken at the input. In order to perform this computation, the noise-equivalent bandwidth of the particular filter involved must be found. In the simulations of digital communication systems discussed later, the error probability estimates can be properly formed from the SNR estimates only if the noise-equivalent bandwidths of the filters can be accurately determined. Therefore, tests were performed to measure the noise-equivalent bandwidth of the general filter for a range of filter parameters.

For many analog filters the noise-equivalent bandwidths are well-known, and tables of these values may be used to estimate the noise-equivalent bandwidth of the digital filter. However, in many cases, especially where a critical frequency of the filter exceeds ten percent of the sampling frequency, the digital filter noise-equivalent bandwidth departs significantly from that of the corresponding analog filter. This is particularly true when the filter order is low. The noise-equivalent bandwidth, even for a given digital filter, will vary somewhat depending upon the details of the implementation used. Different software packages may compute the filter coefficients in a variety of ways, resulting in different errors. In addition, the different register lengths of various machines will also create some variation in the noise-equivalent bandwidth from computer to

computer. Thus, in order to determine very accurately the noise-equivalent bandwidth for a given digital filter implemented on a particular machine, the bandwidth should be experimentally measured on the computer where it is to be used. The noise-equivalent bandwidth of the general filter used in all the simulations performed on the IBM 370 was tabulated for a wide range of filter parameters using a technique for experimentally determining the bandwidth. Appendix D explains the method used to perform the necessary calculations. A brief table is included to give an indication of the results which were obtained. A complete tabulation of the rather extensive results is not given, since it would not necessarily be applicable to a different software implementation, or even to the same software used on another computer with a different register length.

In order to test the lowpass and bandpass SNR estimators in estimations of a simple analog system, various delay values were programmed into test signals. Then, filtered Gaussian noise was added to these delayed signals. Both lowpass and bandpass systems were simulated, and the two routines for SNR estimation were applied to measure the SNR and delay values. The signals themselves were not passed through filters in these tests because unknown delay and distortion would be imposed by the filters. This might yield interesting results, but it would make evaluation of the SNR estimation routines impossible

because the theoretical values for the delay and the SNR would then be unknown. These initial tests were intended to check the accuracy and the useful operating range for the estimators. The tests were not intended to measure the delays and accuracy of the filter models.

A block diagram of the configuration used to perform these simple simulations is shown in Figure 12. The test signals used included waveforms with Gaussian amplitude distributions having lowpass and bandpass spectra for use with the corresponding SNR estimation routines. Sinusoidal signals were also used with the lowpass routine. Sinusoidal modulation of a sinusoid was included in the tests of the bandpass routine. The additive Gaussian noise was filtered so that the noise bandwidth was somewhat greater than that of the signal. Lowpass and bandpass noise spectra were usually chosen to correspond with the type of estimator being used, although this is certainly not a necessary requirement. From Figure 12 it is clear that the signal component of the measurement data is identically a delayed version of the signal source. Thus, even though the test signal is bandpass, subroutine SNRMSE may be used because the group and phase delays are equal. The delayed signals were generated in some cases by time-shifting either the signal or, in the bandpass case, the modulation. In other cases, such as with sinusoids, delays were easily programmed into the signal generator itself.

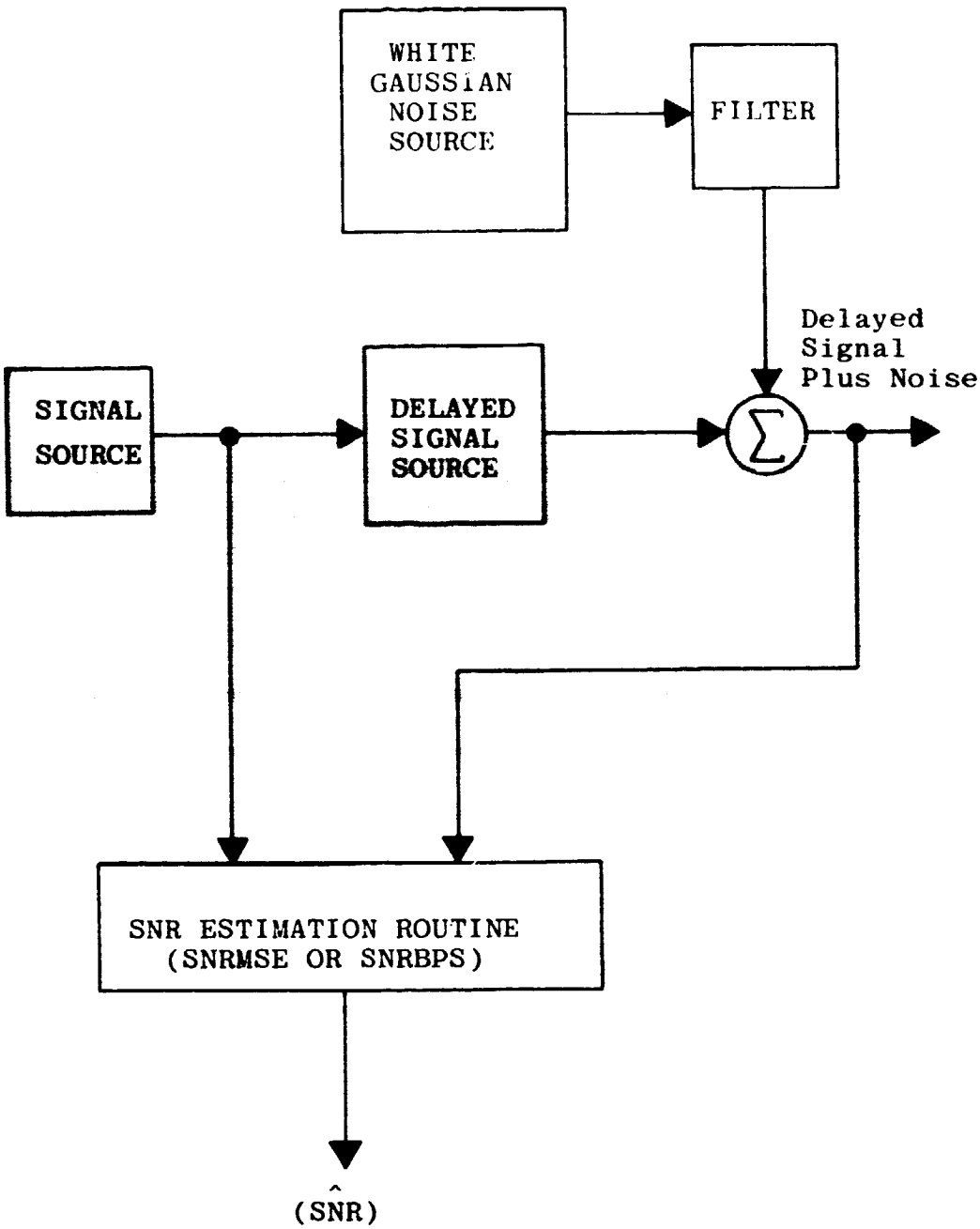

Figure 12. Configuration for Filtered Noise Tests

The results of these tests are given in Table III and Table IV. It should be noted that the SNR results are accurate to within one dB in all cases. For the SNRMSE tests the delay estimates are all correct. Because group delay estimates in SNRBPS are determined from the magnitude of the complex envelope, delay estimates may appear to correspond to either positive or negative values of gain. However, a negative gain may alternately be considered as positive gain with a phase delay of 180 degrees. For a signal such as the sinusoidally modulated test signal, theoretically correct delay values occur every half cycle of the modulation. There were only two errors in group delay estimation, and clearly the SNR estimates were not seriously degraded.

In order to test the lowpass estimator further and to apply the routine to a simulation of a more complex analog system, a computer simulation was performed for an FM communication system. The simulation models required included an FM modulator model and phase-locked loop (PLL) models for demodulation, in addition to the filter package used in the previous tests. Two PLL models were utilized, one of which incorporated a hard-limiter at the input, and one of which did not. Transient response tests and phase error variance tests were performed to verify the accuracy of both PLL models. These results, which are not reproduced here, indicated satisfactory performance of the models [66].

TABLE III

SIMULATION TEST RESULTS WITH SNRMSE

Run	Computer	Test Signal	Total Samples	FFT Length	(dB) SNR	$\widehat{SNR}$	τ	$\hat{\tau}_m$
1	IBM 370	SINUSOIDAL	6144	4096	20	19.5	0	0
2	IBM 370	SINUSOIDAL	6144	4096	30	29.5	0	0
3	IBM 370	SINUSOIDAL	12288	4096	40	39.7	0	0
4	IBM 370	SINUSOIDAL	12288	4096	60	59.7	0	0
5	IBM 370	RANDOM	6144	4096	20	19.3	0	0
6	IBM 370	RANDOM	6144	4096	40	39.1	0	0
7	IBM 370	RANDOM	6144	4096	60	59.1	0	0
8	IBM 370	RANDOM	12288	8192	30	30.4	1027	1027
9	IBM 370	RANDOM	12288	8192	40	40.4	1027	1027
10	IBM 370	RANDOM	12288	8192	50	50.4	1027	1027
11*	IBM 370	RANDOM	12288	8192	50	49.4	1027	1027
12**	UNIVAC 1110	SINUSOIDAL	6001	4096	20	20.9	0	0
13	UNIVAC 1110	SINUSOIDAL	6001	2048	35	34.3	0	0
14	UNIVAC 1110	SINUSOIDAL	6001	2048	50	49.3	0	0
15***	UNIVAC 1110	SINUSOIDAL	6001	2048	35	34.1	1000	3000

*Run 11 test signal is from a bandpass Gaussian process. All other random signals are lowpass Gaussian.

**Due to sinusoidal periodicity, any multiple of 2000 is correct delay on runs 12-14

***Due to sinusoidal periodicity, 1000 + (any multiple of 2000) is correct delay on run 15.

TABLE IV

SIMULATION TEST RESULTS WITH SNRBPS (IBM 370)

Run	Modulation for Test Signal	Total Samples	FFT Length	(dB) SNR	$\hat{SNR}$	τ	τ_m
1	SINUSOIDAL	10,000	4096	20.2	20.1	0	0
2	SINUSOIDAL	10,000	4096	20.2	20.2	163	163
3*	SINUSOIDAL	5,000	2048	10.4	10.4	163	63
4	SINUSOIDAL	5,000	2048	30.4	30.5	163	163
5	SINUSOIDAL	10,000	8192	9.9	9.8	163	163
6	SINUSOIDAL	20,000	8192	29.9	29.8	163	163
7	RANDOM	5,000	4096	10.6	9.9	163	163
8	RANDOM	10,000	4096	10.5	11.4	163	164
9	RANDOM	20,00	4096	10.0	10.9	163	163
10	RANDOM	10,000	4096	0.5	1.2	0	4
11	RANDOM	10,000	4096	9.0	9.1	0	1
12	RANDOM	10,000	4096	20.9	21.2	0	0
13	RANDOM	10,000	4096	30.1	30.4	0	0
14	RANDOM	10,000	4096	41.2	41.4	0	0
15	RANDOM	10,000	4096	49.4	49.8	0	0

*One half cycle of the sinusoidal modulation is 100 samples. Therefore, due to periodicity, a group delay estimate of (163 $\pm$ multiple of 100) yields a theoretically correct value of R_2 and hence a valid SNR estimate.

In the FM communication system simulation, several different combinations of system parameters were used. Simulations were performed using 10 Hz sinusoidal modulation and a 200 Hz carrier with deviation ratios of one and five. Actually, the frequency scale utilized was arbitrary. For example, if 10 MHz modulation and a 200 MHz carrier were used, the simulation would remain the same except for scale factors. The frequencies used were chosen mainly for convenience in determining various system parameters. Two different sets of seeds were used for the noise generator. Then, a sample function from a lowpass Gaussian random process, f_{3dB} = 30 Hz, was used for the modulating signal with a deviation ratio of one. Once again two different seed sets were utilized for generating noise. Every simulation was performed using both of the PLL models. In each case the system was simulated for a total of one second with a sampling frequency of 10 KHz. Data were collected for the last 0.75 second. Therefore, results are based upon 7500 samples in each case. A block diagram of the system under test is shown with the SNR estimator SNRMSE in Figure 13.

The simulations were performed on two computer systems, an IBM 370 system at the University of Missouri and a UNIVAC 1108/1110 system at NASA-Johnson Space Center. All of the cases described above were tested using the IBM 370 system, and some of the tests were duplicated on the UNIVAC 1108/1110 system. Specifically,

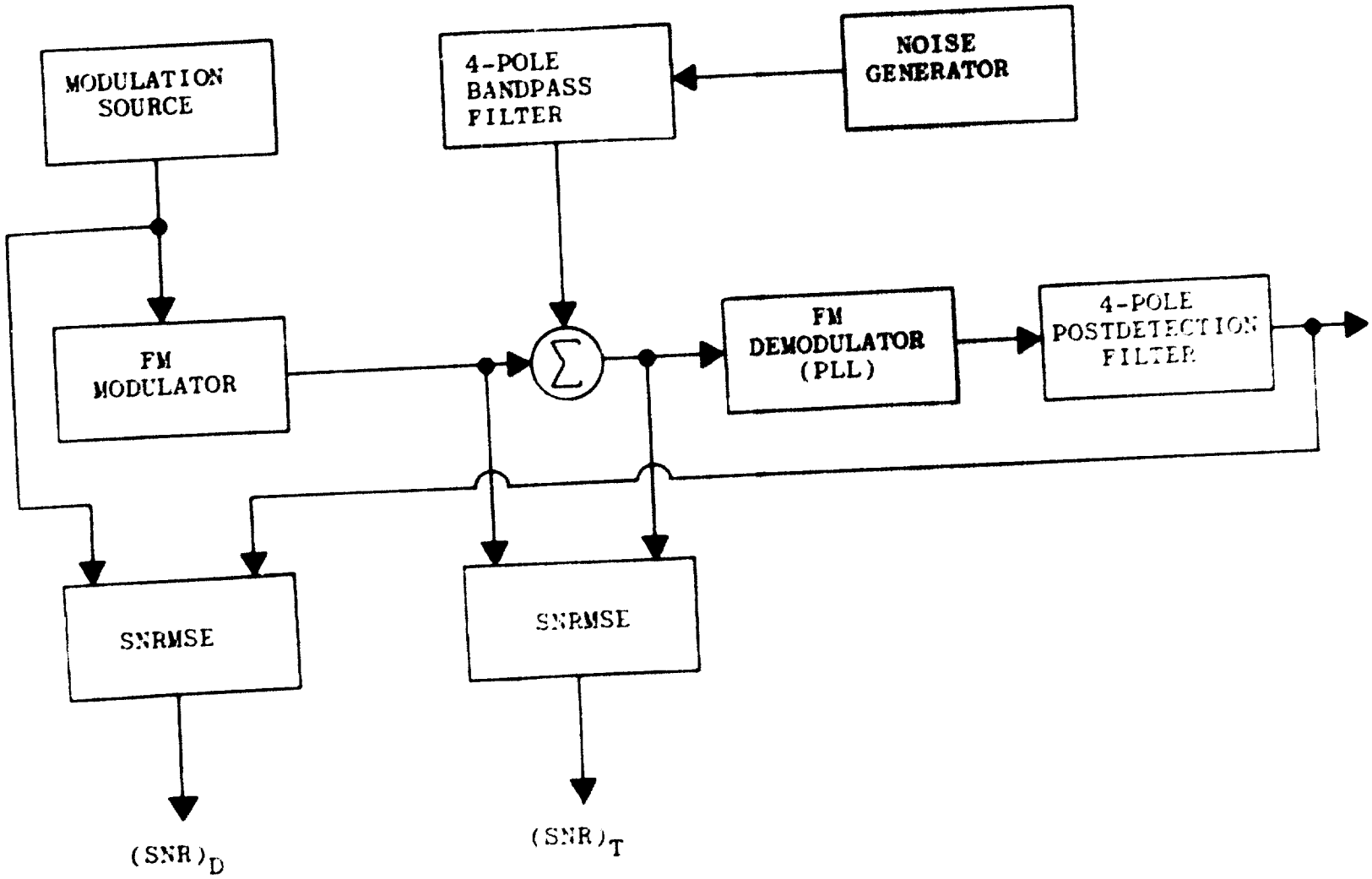

Figure 13. FM System Simulation with SNRMSE

the simulations using sinusoidal modulation and a deviation ratio of five were performed on both machines using both PLL models. Tests were made using PLL natural frequencies of $f_n = 40$ and $f_n = 100$.

In all tests, the lowpass SNR estimation routine, SNRMSE, was used to measure the signal-to-noise ratios at the input to the PLL demodulator, $(SNR)_T$, and at the output of the postdetection filter, $(SNR)_D$. Although the signal at the input to the PLL was bandpass, it may be seen from Figure 13 that there is no delay involved between the reference and the measurement data. Therefore, the routine developed for lowpass systems may be applied. Predetection signal-to-noise ratio, $(SNR)_T$, was varied over a range of values to include the region of threshold. These values and the resulting values of postdetection signal-to-noise ratio, $(SNR)_D$, were tabulated. This data appears in Appendix E.

The theoretical value of $(SNR)_D$ which results from a given value of $(SNR)_T$ for an FM system using a conventional discriminator with sinusoidal modulation is given by [67]*

$$(SNR)_D = \left[\frac{2W^3}{3B_T f_d^2} \cdot \frac{1}{(SNR)_T} + \frac{2WB_T}{\sqrt{3} f_d^2} \operatorname{erfc}\left(\sqrt{(SNR)_T} \right) + \frac{8W}{\pi f_d} e^{-(SNR)_T} \right]^{-1}$$

*This expression differs from that given by Ziemer and Tranter because a noise term is included here which is neglected in the reference to obtain a simple approximation. There are also notational changes.

where

W = Postdetection filter noise-equivalent bandwidth

B_T = Predetection filter noise-equivalent bandwidth

and

f_d = Peak frequency deviation.

For the case of Gaussian modulation, the corresponding expression is [68]*

$$(SNR)_D = \left[\frac{W^3}{3B_T f_d^2 \sigma_m^2} \cdot \overline{(SNR)_T} + \frac{WB_T}{f_d^2 \sigma_m^2 \sqrt{3}} \operatorname{erfc}[\sqrt{(SNR)_T}] + \frac{2\sqrt{2W}}{\sqrt{\pi} f_d \sigma_m} e^{-(SNR)_T} \right]^{-1}$$

where σ_m^2 represents the variance of the modulating signal.

Theoretically, the PLL and the conventional discriminator performances should be essentially the same for high values of the SNR. As the SNR is reduced, however, the PLL should extend to a lower value of $(SNR)_T$ before exhibiting a threshold. Exactly this behavior can be seen in the curves of $(SNR)_D$ versus $(SNR)_T$ which were plotted in Figure 14 through Figure 21. The threshold extension is more pronounced at the higher deviation ratio, just as it should be. This may be seen by comparing Figures 14 through 17 (deviation ratio = 5) with Figures 18 and 19 (deviation ratio = 1). Figures 14 through 17 also show the effect of decreasing the loop natural

*This expression includes a noise term which is neglected to obtain a simple approximation by Taub and Schilling. Again there are notational changes.

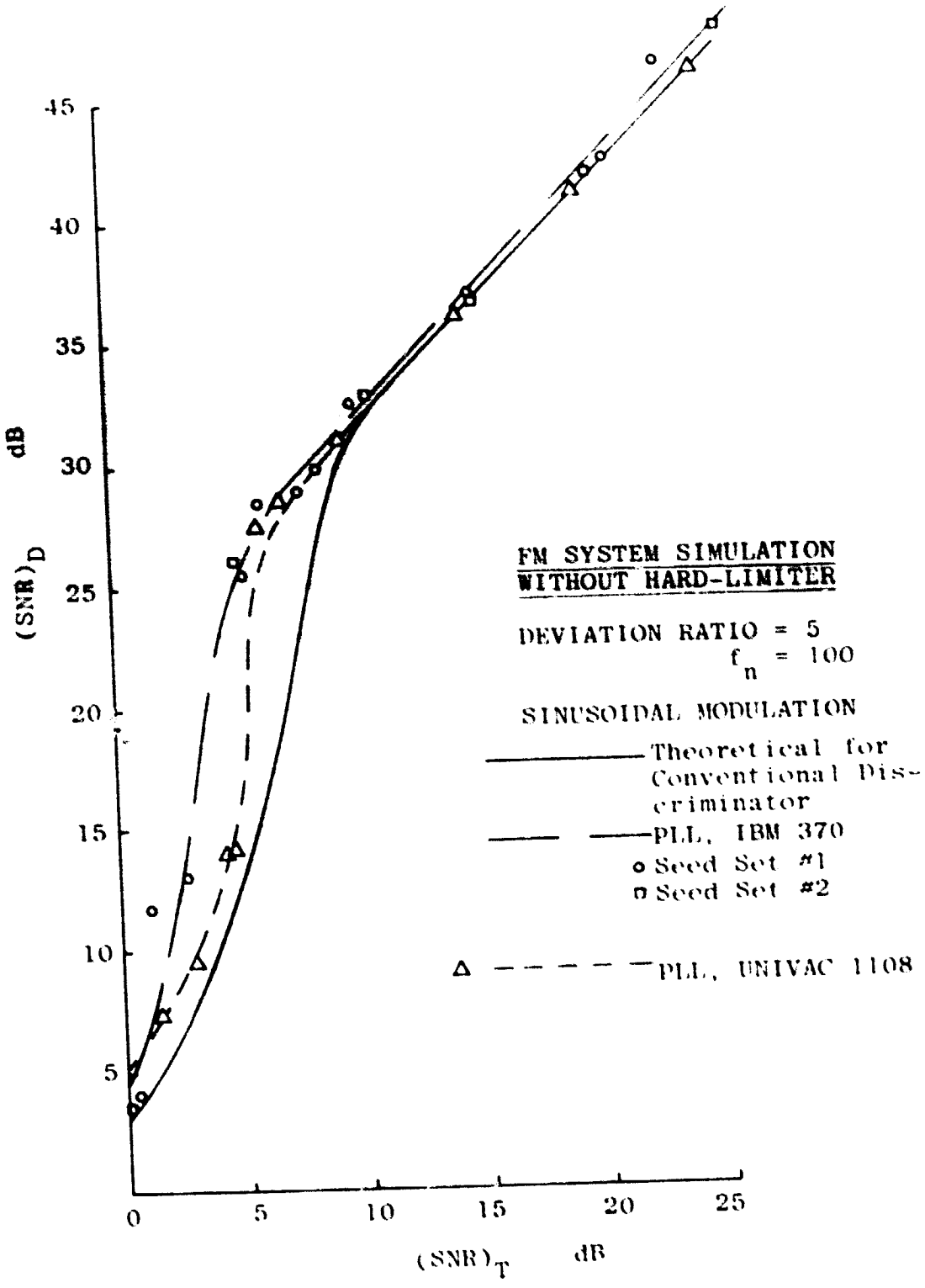

Figure 14. $(SNR)_D$ Versus $(SNR)_T$, Case 1

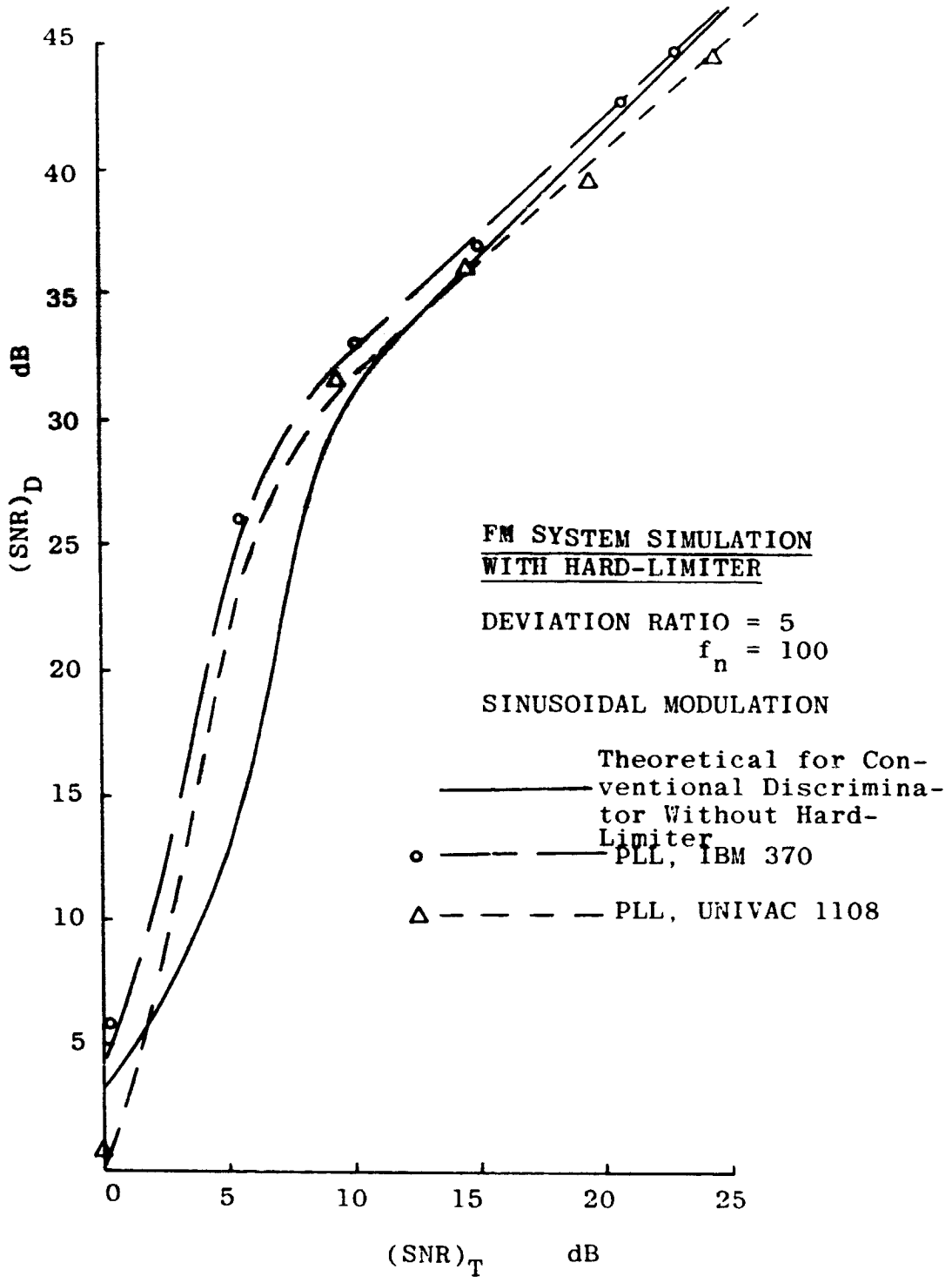

Figure 15. $(SNR)_D$ Versus $(SNR)_T$, Case 2

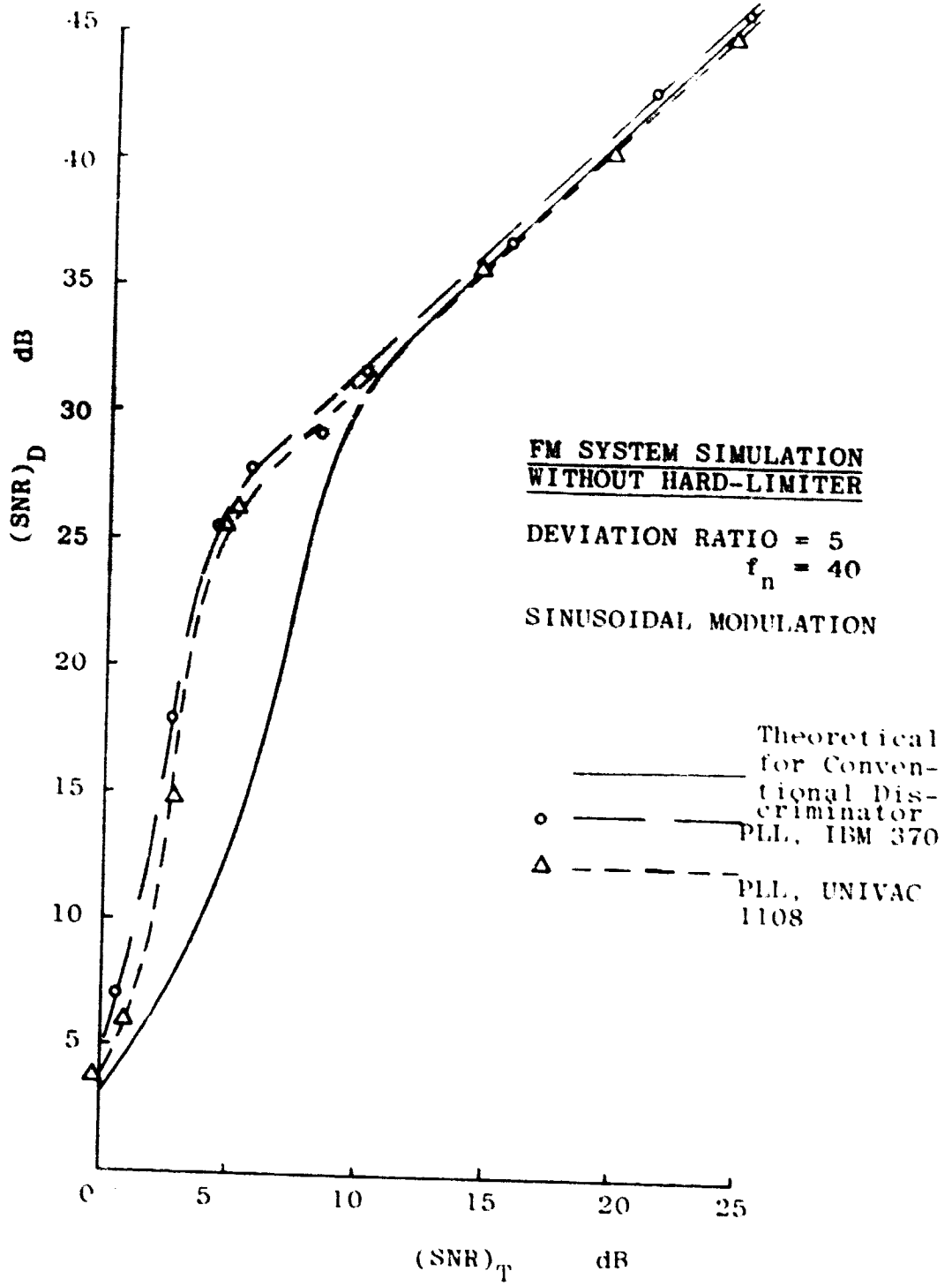

Figure 16. $(SNR)_D$ Versus $(SNR)_T$, Case 3

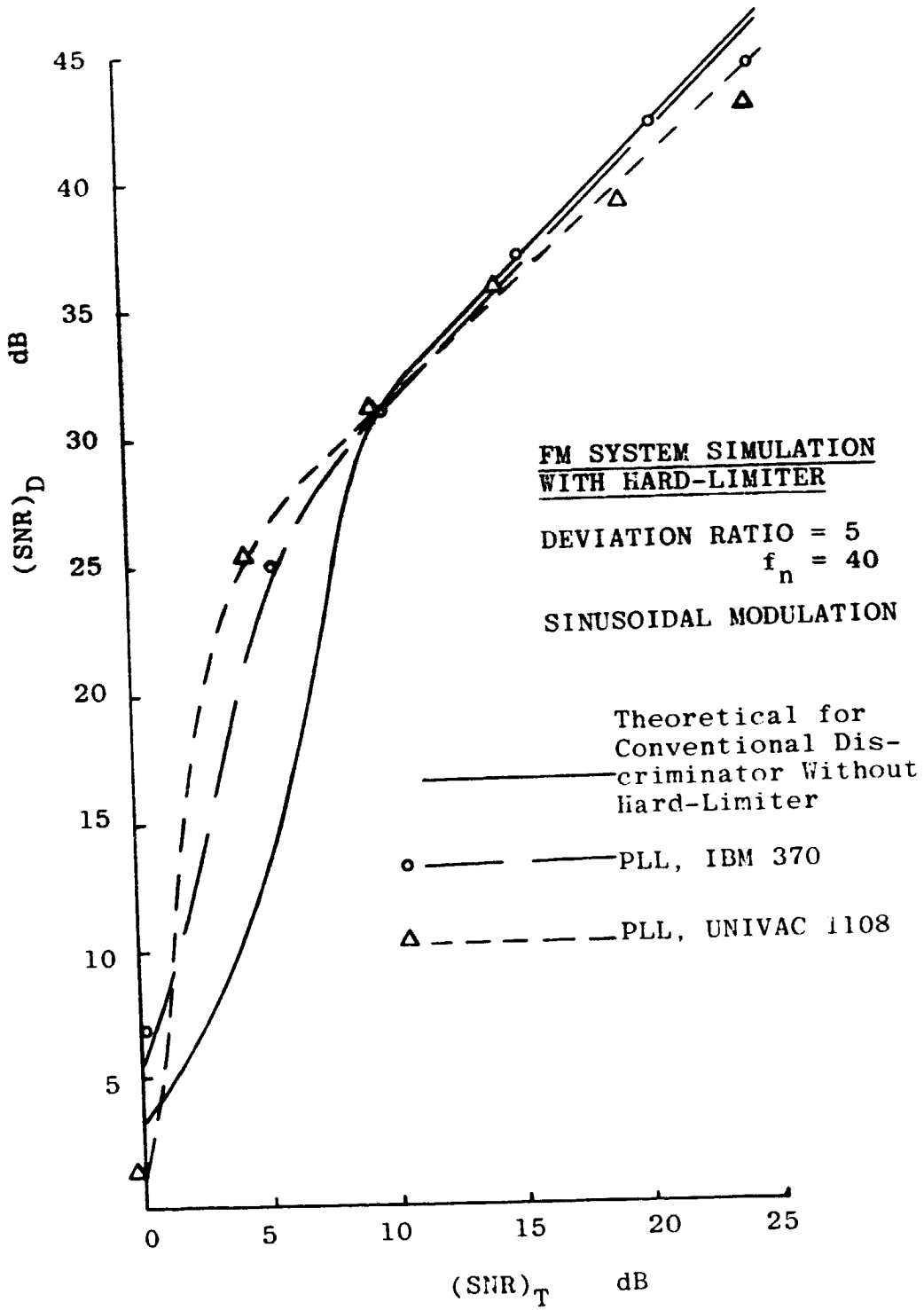

Figure 17. $(SNR)_D$ Versus $(SNR)_T$, Case 4

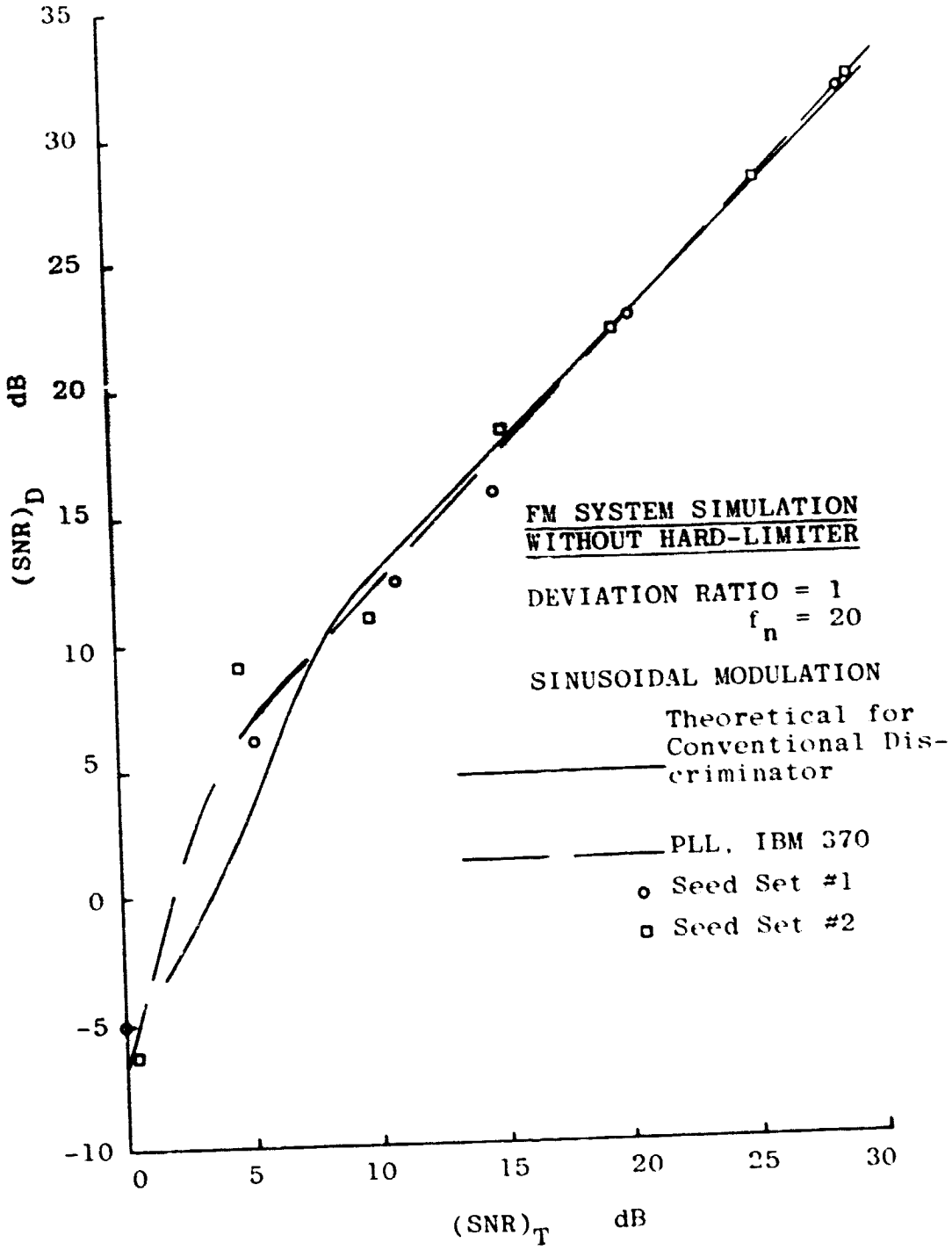

Figure 18. $(SNR)_D$ Versus $(SNR)_T$, Case 5

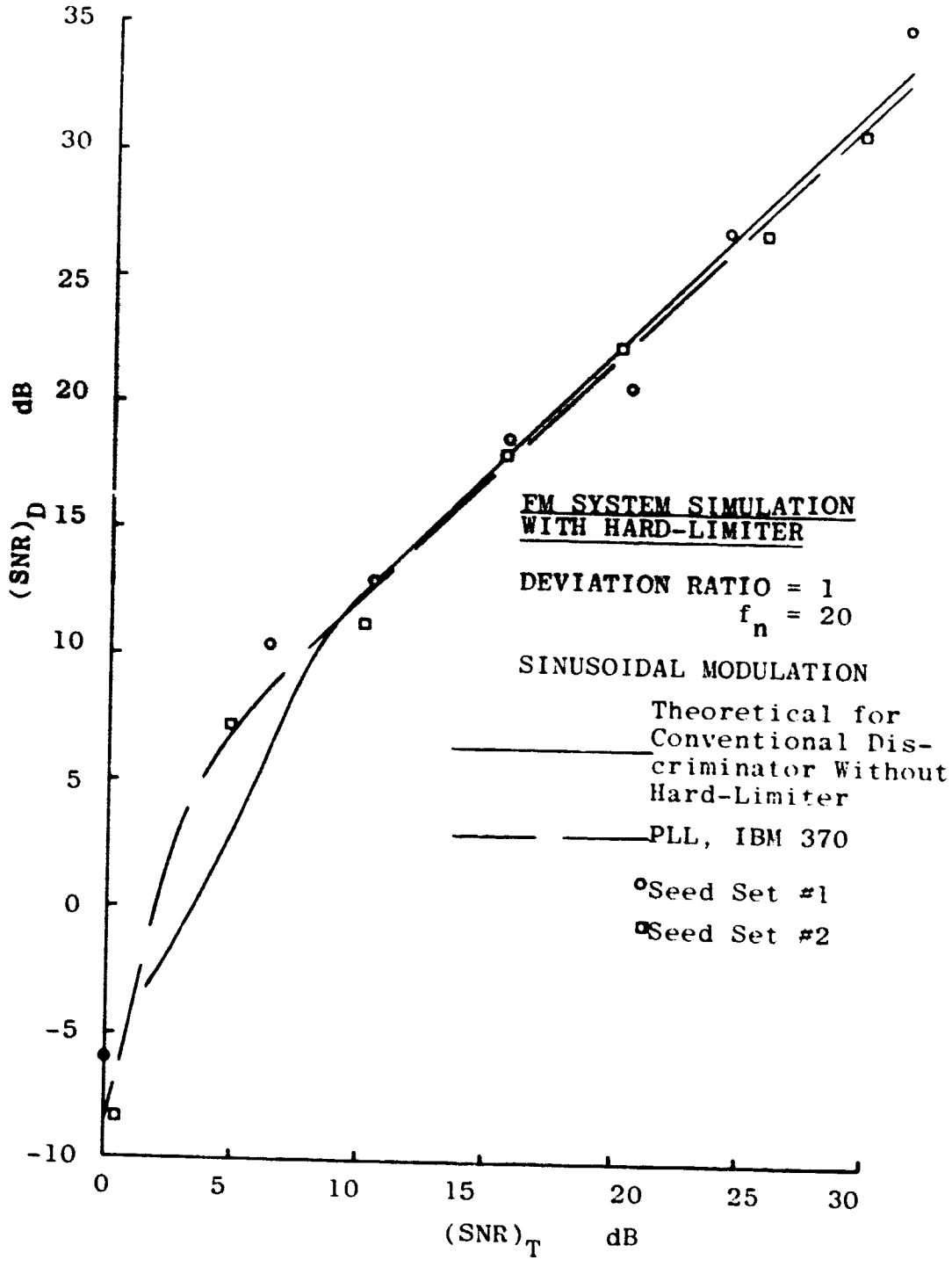

Figure 19. $(SNR)_D$ Versus $(SNR)_T$, Case 6

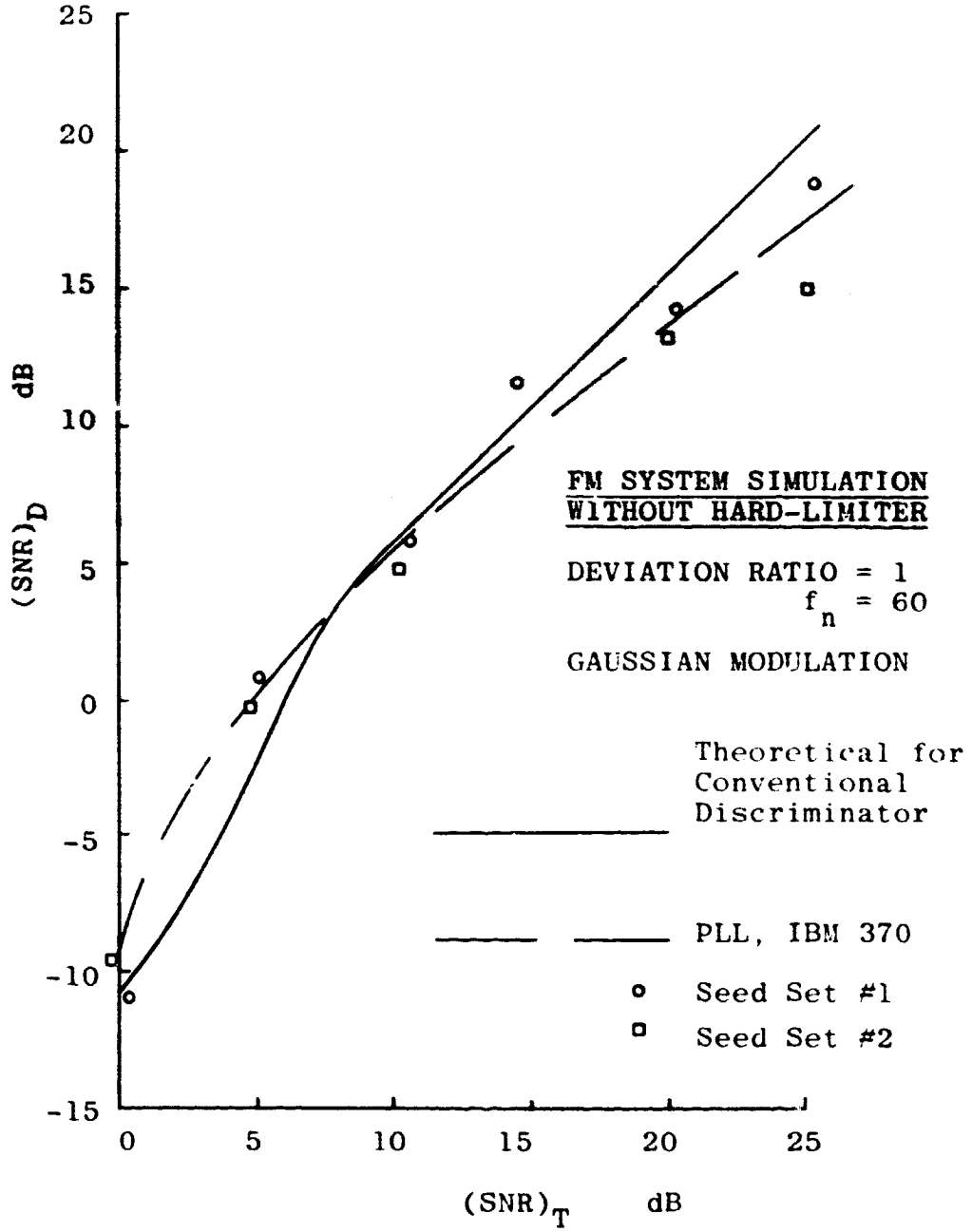

Figure 20. $(SNR)_D$ Versus $(SNR)_T$, Case 7

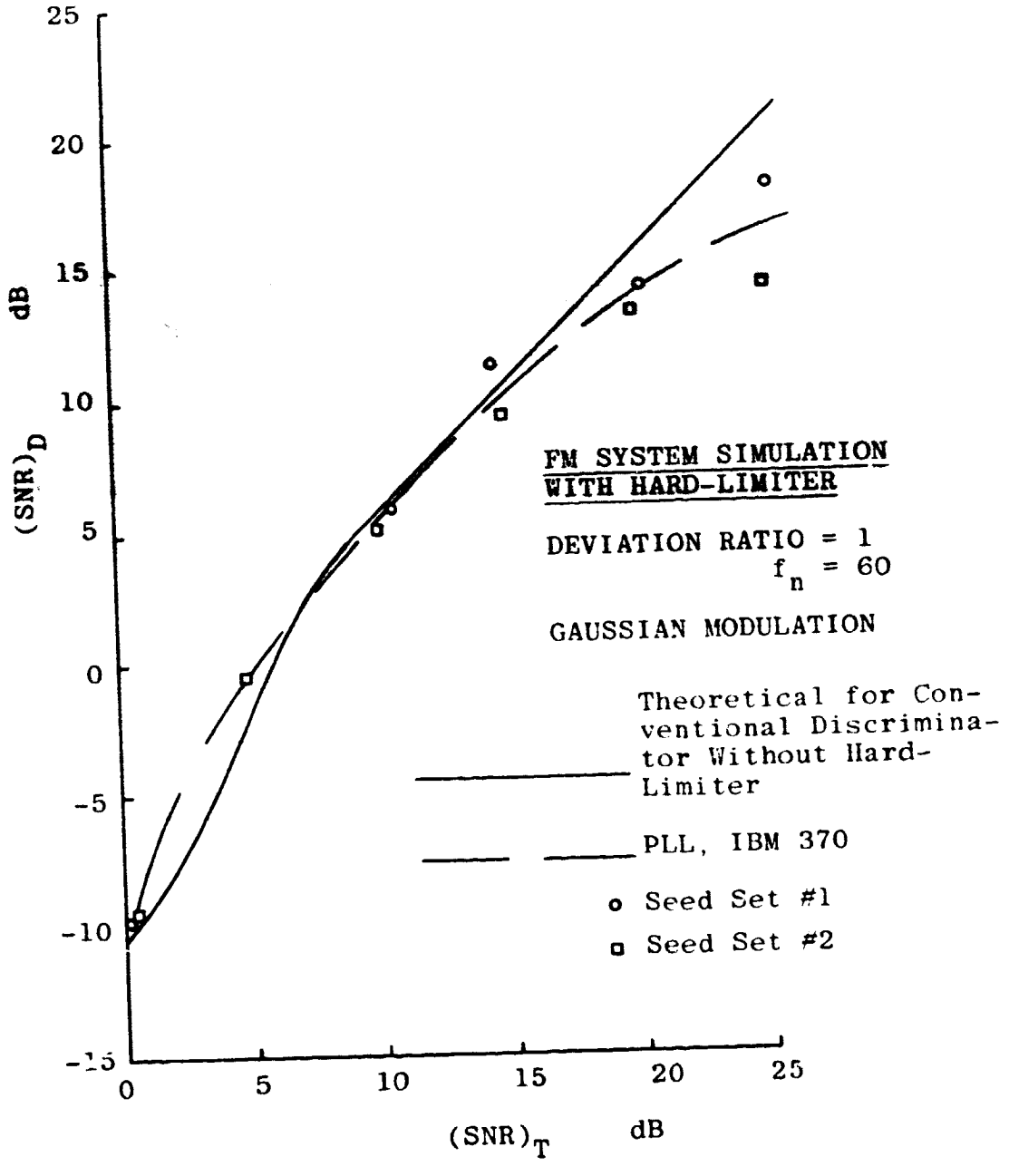

Figure 21. $(SNR)_D$ Versus $(SNR)_T$, Case 8

frequency, f_n. As f_n is reduced, the passband of the PLL is reduced. Less noise power is passed by the loop, and the threshold is extended more. This may be seen by comparing Figures 14 and 15 (f_n = 100) with Figures 16 and 17 (f_n = 40). Continuing to reduce the value of f_n, however, would eventually prevent the PLL from tracking the signal accurately because of insufficient demodulator bandwidth.

The results obtained using Gaussian modulation are shown in Figures 20 and 21. The measurements of $(SNR)_D$ appear to be low for large values of $(SNR)_T$. This reduction in accuracy was partially caused by distortion of the lowpass random signal in the postdetection filter, since a small part of the signal power fell outside the passband of this filter. There were also some problems with obtaining stationary signals due to the very low frequency components in the modulation. Even so, the results appear to be quite good below $(SNR)_T$ = 20 dB. The accuracy of the simulation should be improved by increasing the order of the filter used to generate the lowpass modulating signal, so that its spectrum would roll off more sharply. Reducing the amplitude of the lowest frequency components of the modulation should reduce the problems with stationarity and improve the results, also.

The simulation results are in good agreement with the theoretical results for a true FM system. It is important to note that subroutine SNRMSE has provided

accurate SNR estimates for values of the SNR from below 0 dB to over 45 dB. These results indicate that the SNR estimation routine is valid and accurate. They also demonstrate the successful application of the SNR estimator to a simulation of an analog communication system.

VII. APPLICATION TO DIGITAL SYSTEMS

One of the major reasons for developing the SNR estimators was to facilitate the evaluation of digital communication systems using computer simulation. Thus, to test the routines, a number of digital systems were simulated, and their performances were evaluated using SNR estimation. These results were then compared with the theoretical performances. The systems simulated include coherent ASK and FSK systems and a noncoherent FSK system, all with simple additive Gaussian noise channels. Additionally, a noncoherent FSK system was tested in a Rayleigh fading environment, and in another simulation diversity transmission was utilized to improve performance in the fading channel. In all cases the systems being simulated were chosen so that the mapping from the SNR to P_E could be analytically determined. This was done so that P_E could be estimated using the SNR estimate for comparison with a direct count of the errors produced by the system.

In order to provide a somewhat realistic simulation of the actual system in each case, additive white Gaussian noise was combined with the signal in the channel, and the signal-plus-noise was filtered by a predetection filter. This procedure allowed unknown group and phase delays to be introduced by the predetection filter, and the SNR estimation routine for bandpass systems was

required. Some early attempts to use the lowpass version of the estimator showed SNRMSE to be completely inadequate with such systems as, of course, would be expected.

There is one theoretical problem that results when filtering the signals in this manner which should be mentioned. The theoretical mapping from the SNR to P_E is generally developed under the assumption that the only errors present in the noisy waveform are those produced by the additive random noise. The signal is usually assumed to pass through the predetection filter without distortion. In practice, the filter generally distorts the signal to some degree. The amplitude response of the filter is not perfectly flat over the passband, and the phase characteristic is not perfectly linear. Additionally, there are usually some spectral components of any practical signal which fall outside the passband, and intersymbol interference can become significant. In order to justify neglecting errors in the mapping from the SNR to P_E due to these effects, a simulation was performed for each system with the Gaussian noise set to zero. The SNR estimator measured errors in the system due to all sources other than the Gaussian noise. It was determined in each case that the Gaussian noise error would exceed all other errors by at least ten dB (and usually much more), even for the lowest values of Gaussian noise power to be considered. Hence, the assumption that the total noise power is essentially the result of only the additive

Gaussian noise is justified, and the theoretical mapping from the SNR to P_E is applicable. To further substantiate these arguments, additional simulations were performed for the coherent ASK and FSK systems in which the Gaussian noise was first bandlimited by a filter identical to the predetection filter before adding it to the signal. Then the predetection filter was removed from the signal path altogether. The results of these simulations are tabulated and plotted with the other results for the coherent ASK and FSK systems. It can be seen that there is no noticeable difference in the accuracy obtained with the two configurations.

Block diagrams of the simulations used for the coherent ASK system are shown in Figures 22 and 23. The modulation source utilized to provide input symbols to the binary ASK modulator was a PN sequence generator producing pseudo-noise symbols in a sequence with a period of 1023 symbols. The ASK modulator performed on-off keying of a carrier. In one case, white Gaussian noise was added, and the noisy waveform was passed through a bandpass predetection filter. In the other case, the same filter was applied to the noise before it was added to the signal, and the signal was not passed through a filter. In both cases, the output was provided to a coherent ASK demodulator and to a PLL which tracked the carrier component and provided a phase reference for the coherent demodulator. The demodulated symbols were compared with the original modulation symbols, and the errors which occurred in operating

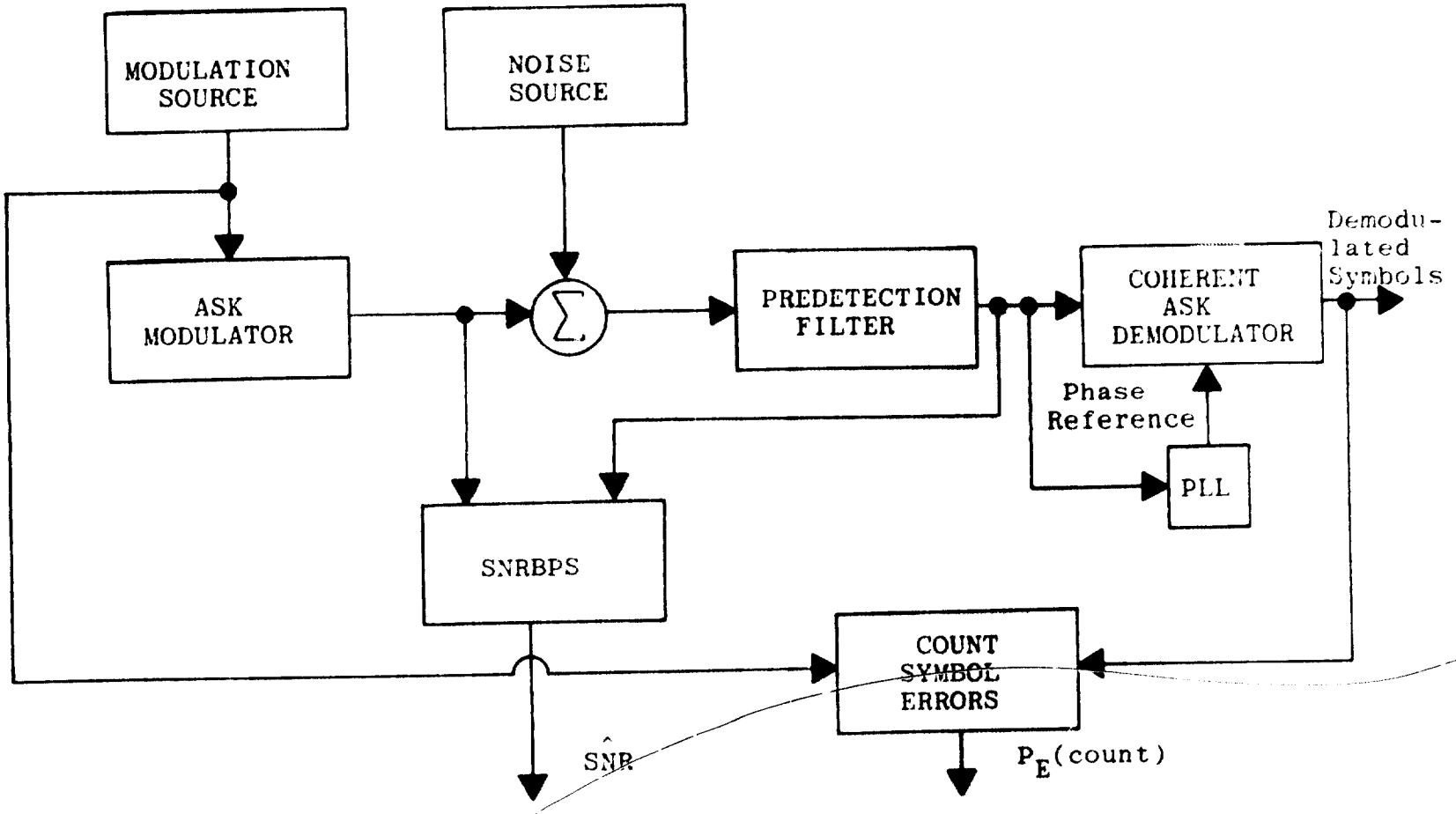

Figure 22. Simulation of the Coherent ASK System Using a Predetection Filter

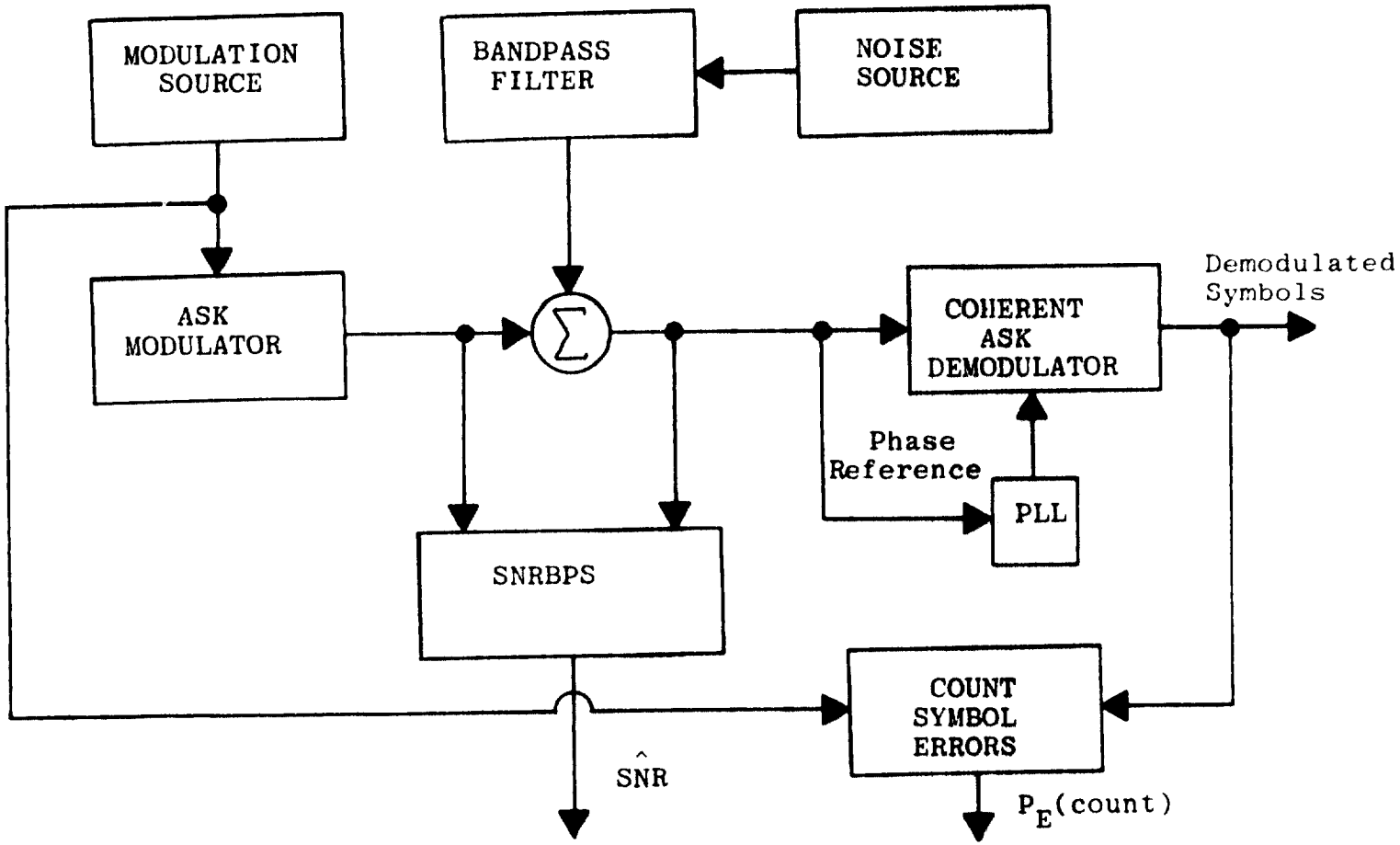

Figure 23. Simulation of the Coherent ASK System Without a Predetection Filter

the system were counted. Dividing the number of errors by the total number of symbols processed yields an estimate for the probability of error obtained by a direct count, $P_E(\text{count})$.

Data were collected at the input to the demodulator where an estimate of the SNR was desired. Data were also collected at the modulator output to serve as the reference signal. Subroutine SNRBPS formed the SNR estimate for the waveform at the demodulator input. From this estimate, $\hat{SNR}$, an estimate can be obtained for the ratio of the average energy per transmitted symbol, E_s, to the noise power spectral density, N_o. Then the well known mapping from E_s/N_o to P_E for a coherent binary ASK system operating in additive Gaussian noise, [69]

$$P_E = \frac{1}{2} \operatorname{erfc} \sqrt{\frac{1}{2} \frac{E_s}{N_o}} \tag{60}$$

where

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-\alpha^2} d\alpha$$

can be applied.

In order to find E_s/N_o from the SNR, denote the time duration of an information symbol by T_s and the average power present in the signal at the point of measurement by P_s. Then the total average energy per symbol is given by

$$E_s = P_s T_s .$$

Denote the noise-equivalent bandwidth of the filter by B_n. Then, the total noise power present at the point of measurement is given by $N_o B_n$. Thus, the SNR at the point of measurement can be written

$$SNR = \frac{P_s}{N_o B_n} = \frac{E_s}{N_o T_s B_n} .$$

Therefore

$$\frac{E_s}{N_o} = (SNR)(T_s B_n)$$

Hence, the ratio E_s/N_o is related to the SNR by a proportionality constant which is the time-bandwidth product of the system. The relation for P_E becomes

$$P_E = \frac{1}{2} \operatorname{erfc} \sqrt{\frac{1}{2}(SNR)(T_s B_n)} . \qquad (61)$$

Inherent in the theoretical development of this mapping is the assumption of a perfect phase reference in the coherent detector. Although the phase reference provided by the PLL is not perfect, the phase error variance was measured for each simulation and was found to remain below 0.03 radian^2 throughout. Hence phase errors may reasonably be assumed to be quite small, and the mapping based upon a perfect phase reference should yield a very

close approximation to the correct mapping for the system being simulated.

Simulations were performed for this system over a range of SNR values where a direct error count was feasible. The various system parameters utilized are tabulated with the results of the simulations in Appendix E. A plot of the results is shown in Figure 24. Points are plotted showing the relationship between the estimated SNR and the error probability obtained by a direct error count. Every data point represents the results of a different simulation. For each point, the value of the abcissa is the SNR estimate scaled by the time-bandwidth product, $(\hat{SNR})(T_s B_n)$, expressed in dB, and the value of the ordinate is the probability of error computed by a direct error count for that same simulation. The number of symbol errors that occurred in each case is shown next to the point in order to reflect the reliability of the P_E estimate obtained by direct error counting. The curve showing the theoretical mapping from the SNR to P_E is plotted for comparison.

The distance along the ordinate between each point and the curve is the error in estimating the probability of error for that simulation that would result if P_E were determined by estimating the SNR and mapping this to an estimate for P_E. Inspection of the plot shows good agreement between the P_E estimates which are found using the

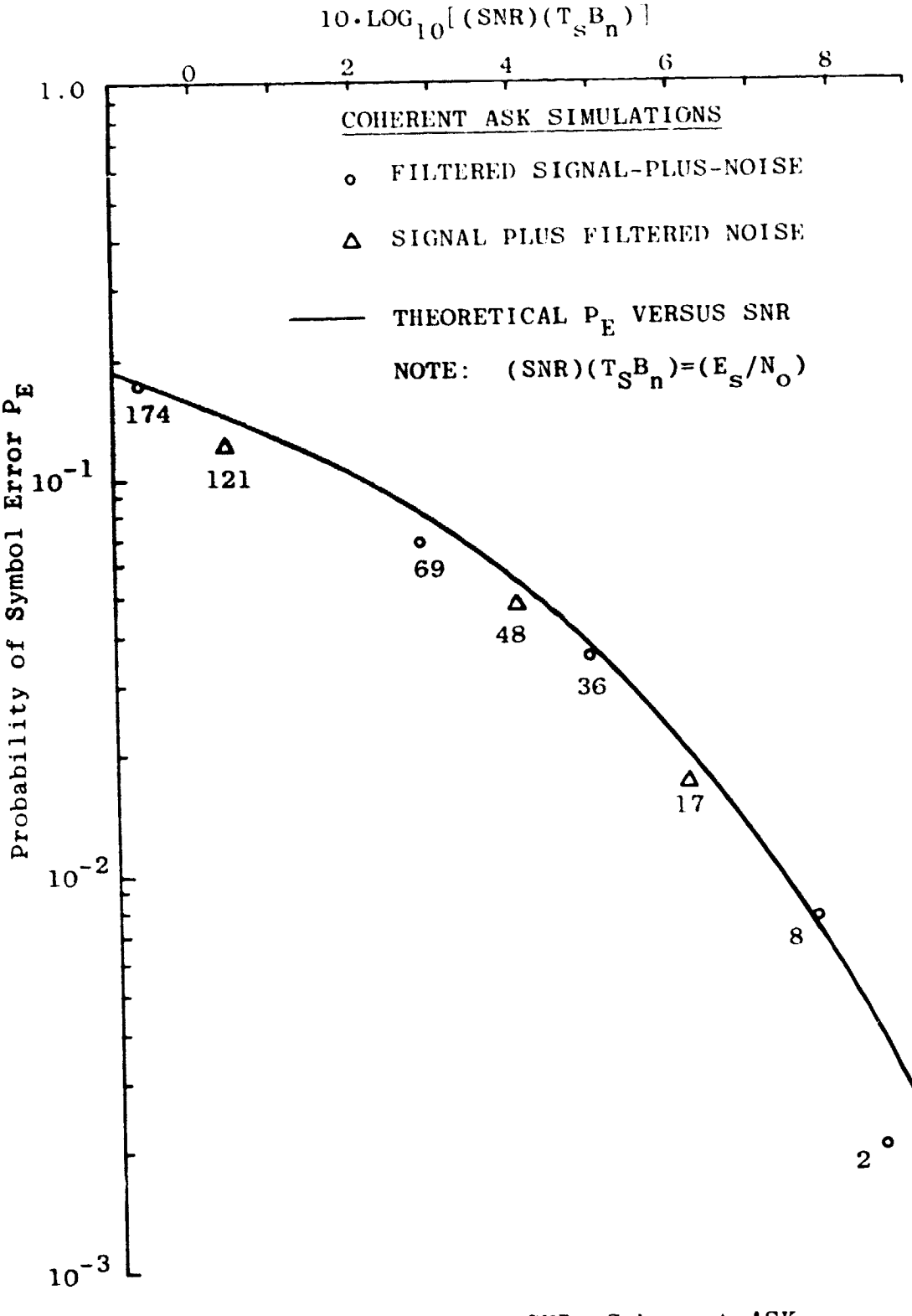

Figure 24. P_E Versus SNR, Coherent ASK

SNR estimation routines and the results obtained by direct error counting. In all cases where more than five errors were made, so that the error count may be considered meaningful, the values obtained for the probability of error by counting and through SNR estimation differ by less than a factor of 1.3. This level of accuracy is more than adequate in most applications where an estimate of P_E is required.

An important point about the data required to produce the estimate for P_E using the two techniques should be made here. Each ASK simulation processed a total of 30,700 samples to simulate the 1023 information symbols used. All these data were used in finding the probability of error by direct error count. The SNR estimate was produced using only 20,000 of these samples. Applying the mapping from the SNR to P_E for the system provided an estimate for P_E. Thus the simulation time required for this case could be reduced by more than one third with no change whatsoever in the P_E estimates obtained through SNR estimation. Further, all the tests which have been performed to check the accuracy of the SNR estimator indicate that SNR estimates could have been obtained with only 10,000 samples, or even less, which would probably differ from the estimates actually obtained by no more than a few tenths of a dB. It seems likely that good P_E estimates for this system could have been obtained through SNR estimation for one third or less the cost in computer time which was

used. Similar results should be possible for other systems which can be accurately simulated, providing the mapping from the SNR to P_E is known. In the cases of the lower error probabilities obtained, such a reduction in simulation time would provide an insufficient number of errors to yield a reliable estimate of the error probability using a direct error count. This sort of reasoning was experimentally verified in the FSK simulations which are discussed next. The P_E estimates obtained through SNR estimation in those (FSK) simulations are formed using well under one half the data used for the direct error counts. The agreement between the results and the theory is even better for the FSK simulations than it is for the ASK case.

The importance of these arguments is obvious when the extension to estimates of P_E in ranges several orders of magnitude smaller are considered. For the same number of samples, the reliability of the SNR estimate is almost unchanged as the SNR is increased approximately ten dB in most systems to obtain this extension. This is seen in the tests of the estimators and in the confidence intervals which were derived in earlier sections. For these higher values of the SNR, a given error in $\hat{SNR}$ generates an error in the estimate for P_E larger than at low values of the SNR. A brief study of this effect is given in Appendix F. The filtered noise test results obtained in the previous section show that the SNR

estimates are usually accurate to within a few tenths of one dB. In most digital systems, these measurements would lead to P_E estimates well within one order of magnitude of the theoretical value. This accuracy is sufficiently adequate to provide a useful estimate of P_E in most applications. The computer time required to estimate P_E using direct error counts in this range usually exceeds practical values by a few orders of magnitude, since, for a given value of P_E, the expected number of errors generated is directly proportional to the simulation time used.

Thus, the SNR estimation technique for determining P_E is directly verified, at least over the range of P_E shown in the plots. There appears to be no reason why the method should not also give good results for the lower ranges of P_E typical in most practical systems.

A similar simulation scheme was used to model a binary coherent FSK communication system. Block diagrams for the FSK system are given in Figures 25 and 26. The configurations are similar to those for the ASK system except for minor changes. The ASK modulator and demodulator are replaced by an FSK modulator and demodulator. Also, two phase-locked loops are required instead of one to supply phase references to the coherent demodulator. Again, a PN sequence is used as a modulation source.

A single predetection filter is used with sufficient bandwidth to cover both frequency channels. The use of

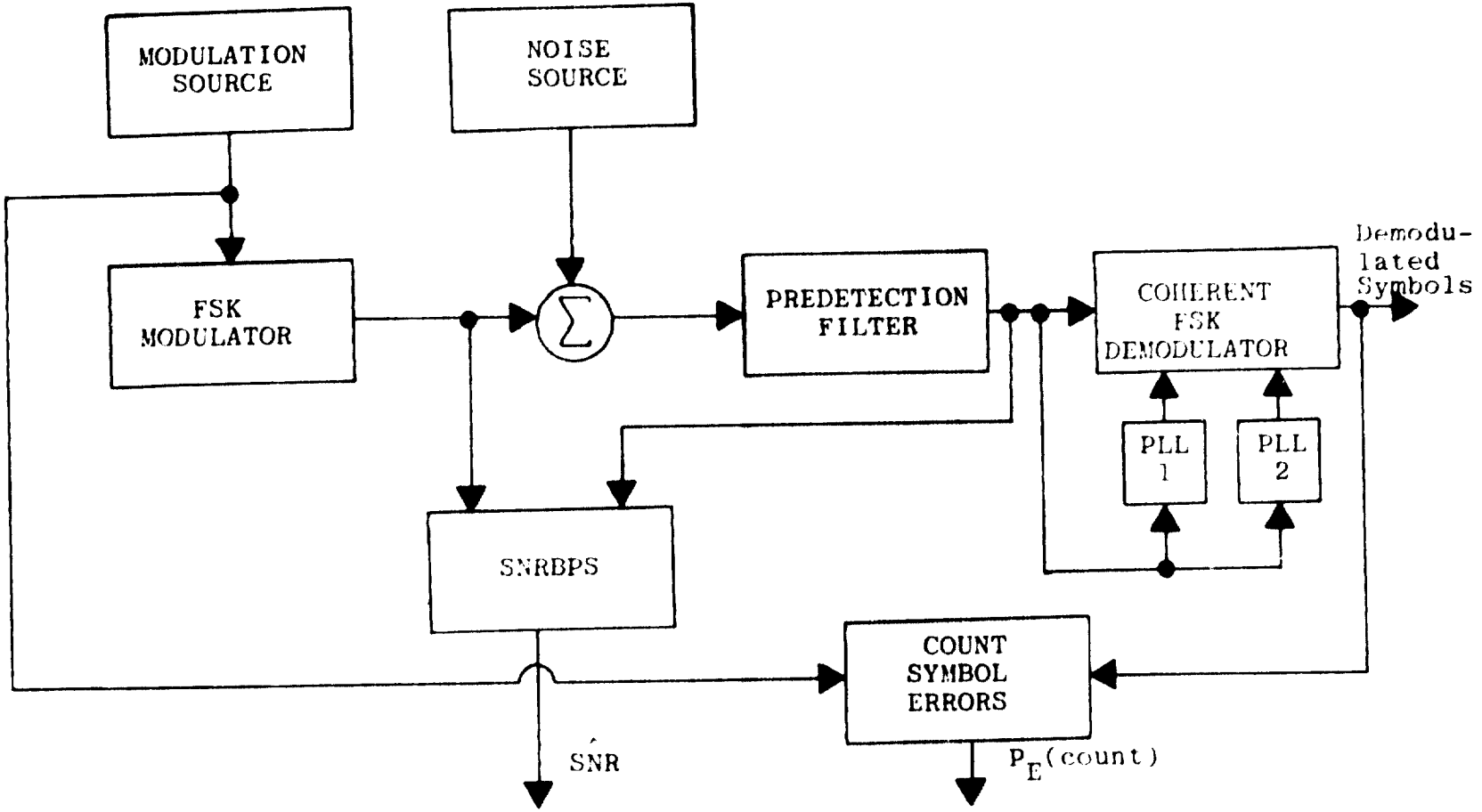

Figure 25. Simulation of the Coherent FSK System Using a Predetection Filter

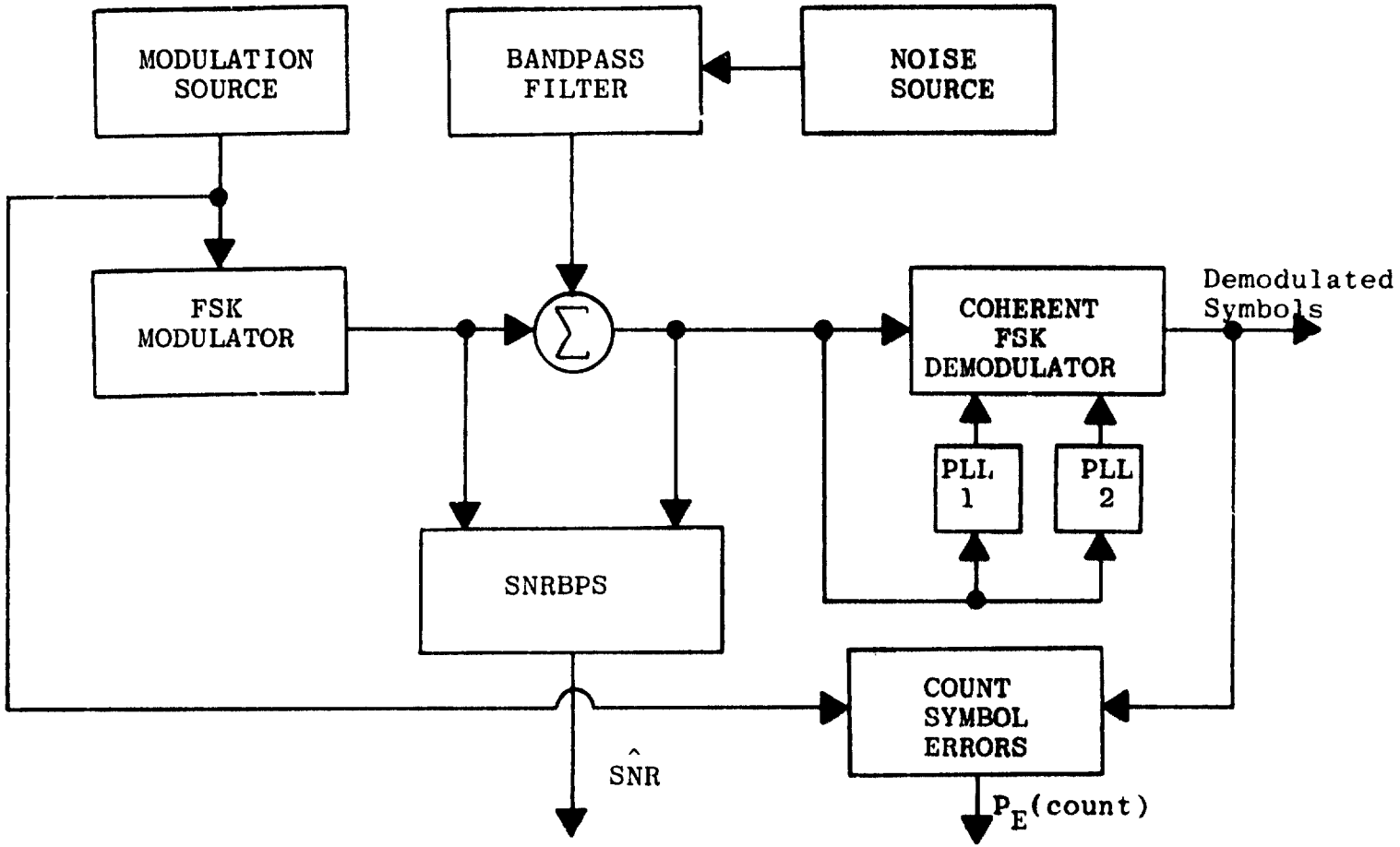

Figure 26. Simulation of the Coherent FSK System Without a Predetection Filter

two filters with narrower bandwidths, one for each frequency channel, would yield a system with performance superior to that of the one simulated, since the SNR for each channel would be improved. However, this change would have no significant effect on the validity of the tests of the SNR estimator. It would introduce the problem of matching the possibly different delays in the two filters to obtain the synchronization necessary at the end of each symbol time when a decision must be made about which symbol was sent. Also, the equivalent configuration which avoids filtering the signal, as shown in Figure 26, could not be so directly simulated using the version with two filters. Therefore, for convenience, the single filter was chosen.

Simulations were performed over a range of SNR values as was done for the ASK case. System parameters and results are given in Appendix E. A plot is drawn just as before to show the relationship of the SNR estimates, the probability of error determined by direct error count, and the theoretical mapping between the SNR and P_E. The theoretical mapping is identical to that given by (61) for the coherent ASK system [10]. These results, shown in Figure 27, indicate excellent accuracy in the SNR estimator. In this case the phase error variance remained below 0.04 radian^2, indicating nearly perfect coherence. The values obtained by the two methods for P_E again differ by less than a factor of 1.3, except in cases where less than five errors were counted.

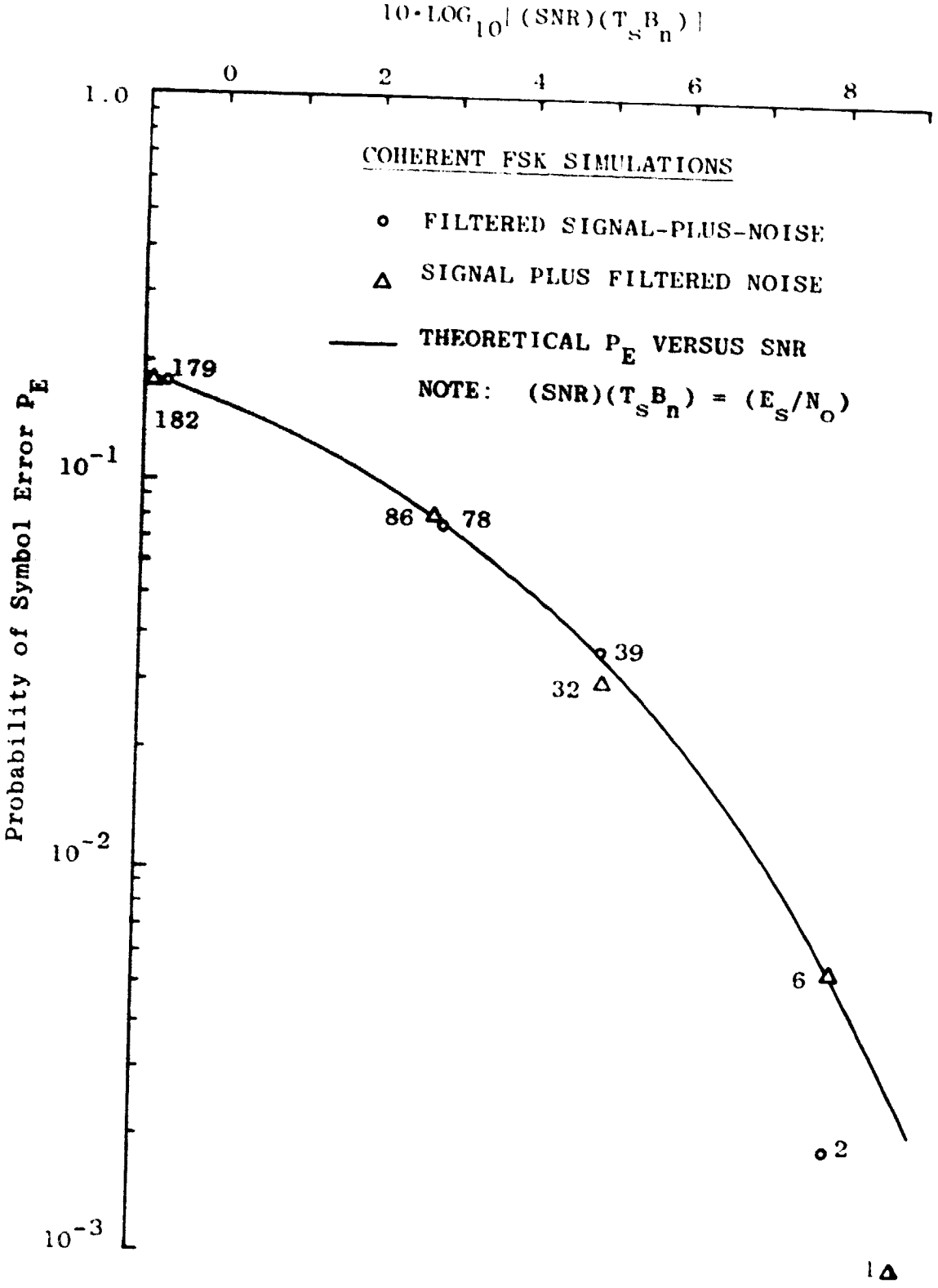

Figure 27. P_E Versus SNR, Coherent FSK

Discussion of the accuracy of the methods developed to evaluate the performance of digital communication systems is most directly accomplished by considering the error in estimating the probability of error. For example, in the ASK and FSK systems discussed, the variation between the probability of error obtained by direct error counting and the P_E estimate obtained through SNR estimation was noted to be less than a factor of 1.3 in all cases where more than five errors were counted. But the accuracy of the SNR estimator is most easily described in terms of the error in the SNR estimate, not in terms of the error generated in the P_E estimate. It is useful to have a set of curves available which maps the error in the SNR estimate to the error in the P_E estimate. Such a mapping is obviously system dependent and cannot be obtained in general. However, the case of coherent systems operating in additive Gaussian noise is quite common, and it is helpful to derive curves for these systems. In Appendix F curves are obtained which cover some of these systems, and a convenient way of expressing the error in estimating P_E is suggested. The data obtained from the coherent ASK and FSK simulations is plotted with the curves to show the area in which these simulations were operating.

The next system which was simulated is a noncoherent FSK system. The simulation is the same as that for the coherent FSK system except for the form of the demodulator

and the fact that the phase-locked loops are not required. A block diagram is shown in Figure 28. As before, a PN sequence serves as the modulation for the system. After white Gaussian noise has been added, the noisy waveform is input to a predetection filter. This filter is followed by the noncoherent demodulator, which consists of two bandpass filters and envelope detectors together with a decision mechanism for decoding each symbol. Also included in the demodulator is a delay in one of the two signal paths, which serves to match the slightly different group delays of the two bandpass filters.

It is worth mentioning that the group delays imposed by these filters were determined experimentally using the group delay estimation feature of SNRBPS. These delays must be found by some technique so that the signals along the two paths are synchronized before decoding each symbol. The routine chiefly intended for SNR estimation in simulations of bandpass systems provides a convenient means for making such delay measurements.

The predetection filter is included mainly for convenience in measuring the SNR over the portion of the spectrum where the system is operating. It is possible to compute this quantity through wideband measurements of noise power taken at the noise generator output before filtering by applying knowledge of the noise spectrum generated. However, it was preferred to directly measure the power in the frequency band occupied by the signal.

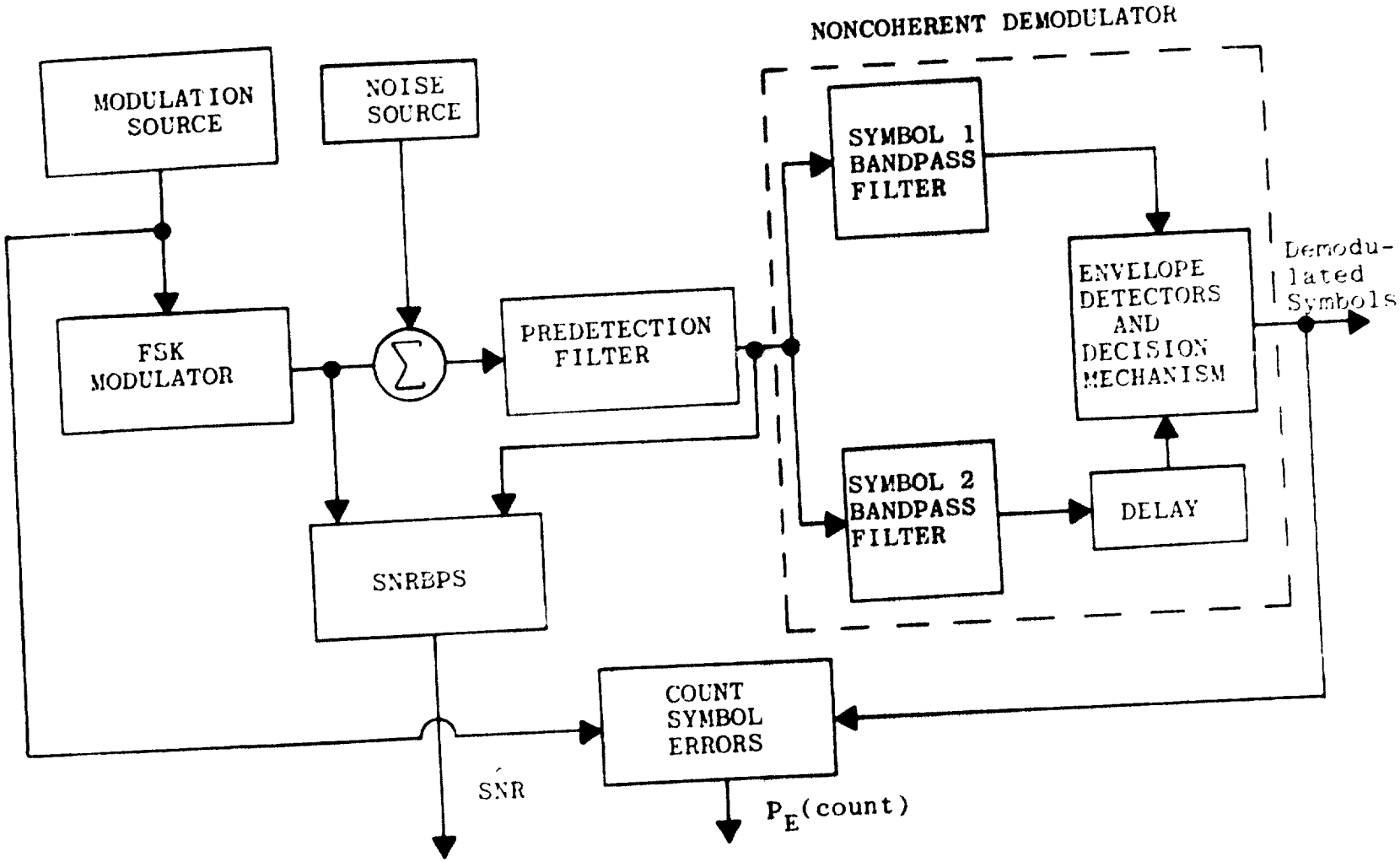

Figure 28. Simulation of the Noncoherent FSK System

Assuming the predetection filter passband includes the passbands of the individual filters which follow, as it must for proper operation, the performance of the system is essentially the same whether or not the predetection filter is used. Without it, however, the signal-to-noise ratios would have to be measured at the outputs of the demodulator bandpass filters. This presents a problem in determining a reference signal because each filter passes frequencies corresponding to only one of the two information symbols. In order to measure the SNR at the output of one of the demodulator filters, a special reference signal would have to be generated corresponding to the theoretical output of that filter. This might be possible, but the predetection filter approach appeared to be simpler. Since the quantity to be determined is E_s/N_o, the calculation is performed just as in every other case, multiplying the SNR by the time-bandwidth of the filter involved, in this case the predetection filter. Once E_s/N_o is found, the expression [71]

$$P_E = \frac{1}{2} e^{-\frac{E_s}{2N_o}}$$

can be used to estimate P_E.

The results of simulations performed for the system are tabulated in Appendix E, and results are illustrated in Figure 29. Once again, the results appear to be in rather good agreement with theory, and the SNR estimator is shown to be capable of producing good estimates for P_E.

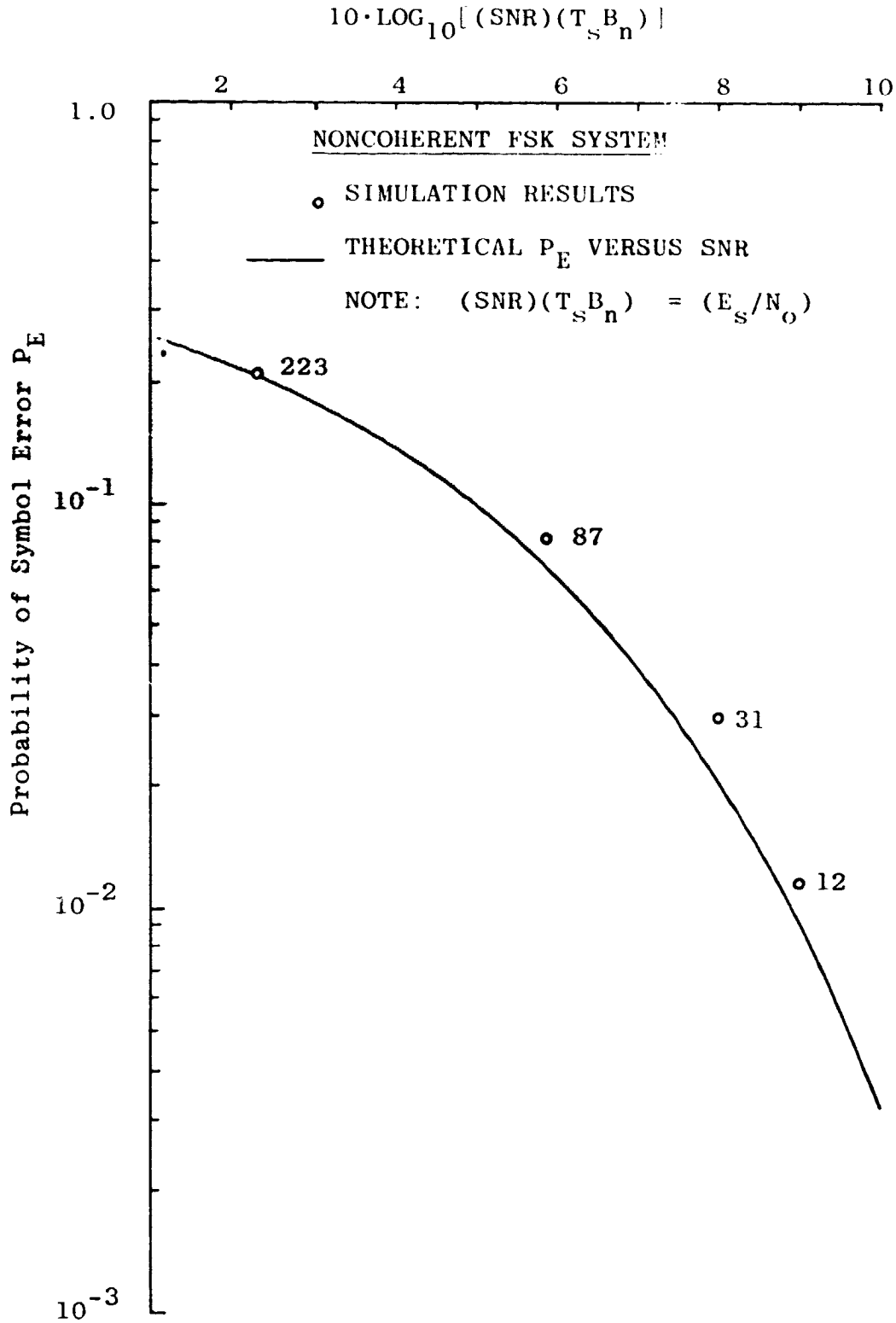

Figure 29. P_E Versus SNR, Noncoherent FSK

Values for the probability of error derived using the two methods differ by less than a factor of 1.5 in all cases for simulations of this system.

In the next set of simulations, the same noncoherent FSK system was used. However, the channel model included a multiplier for the signal which simulated Rayleigh fading. A block diagram for the system is shown in Figure 30. The tap for the reference signal for SNRBPS is taken at the output of the random gain multiplier. The SNR value which is required to compute P_E for the system is based upon the average symbol energy after the Rayleigh fading. If the reference for SNRBPS were taken at the output of the FSK modulator, all error produced by the variations in signal amplitude due to fading would be included as noise in the SNR estimate.

The Rayleigh random variable used to simulate fading was generated as the square root of the sum of the squares of two independent, equal variance Gaussian random variables. In order to achieve an intuitively appealing model for the fading, it was desired that the fade durations be long enough so that the signal amplitude would remain more or less constant over the width of a single information symbol. Yet the duration was required to be sufficiently short that several fades would be included in a single FFT block size suitable for use in the SNR estimation routine. Otherwise, the stationarity assumptions required of the signal for proper operation of SNRBPS would be

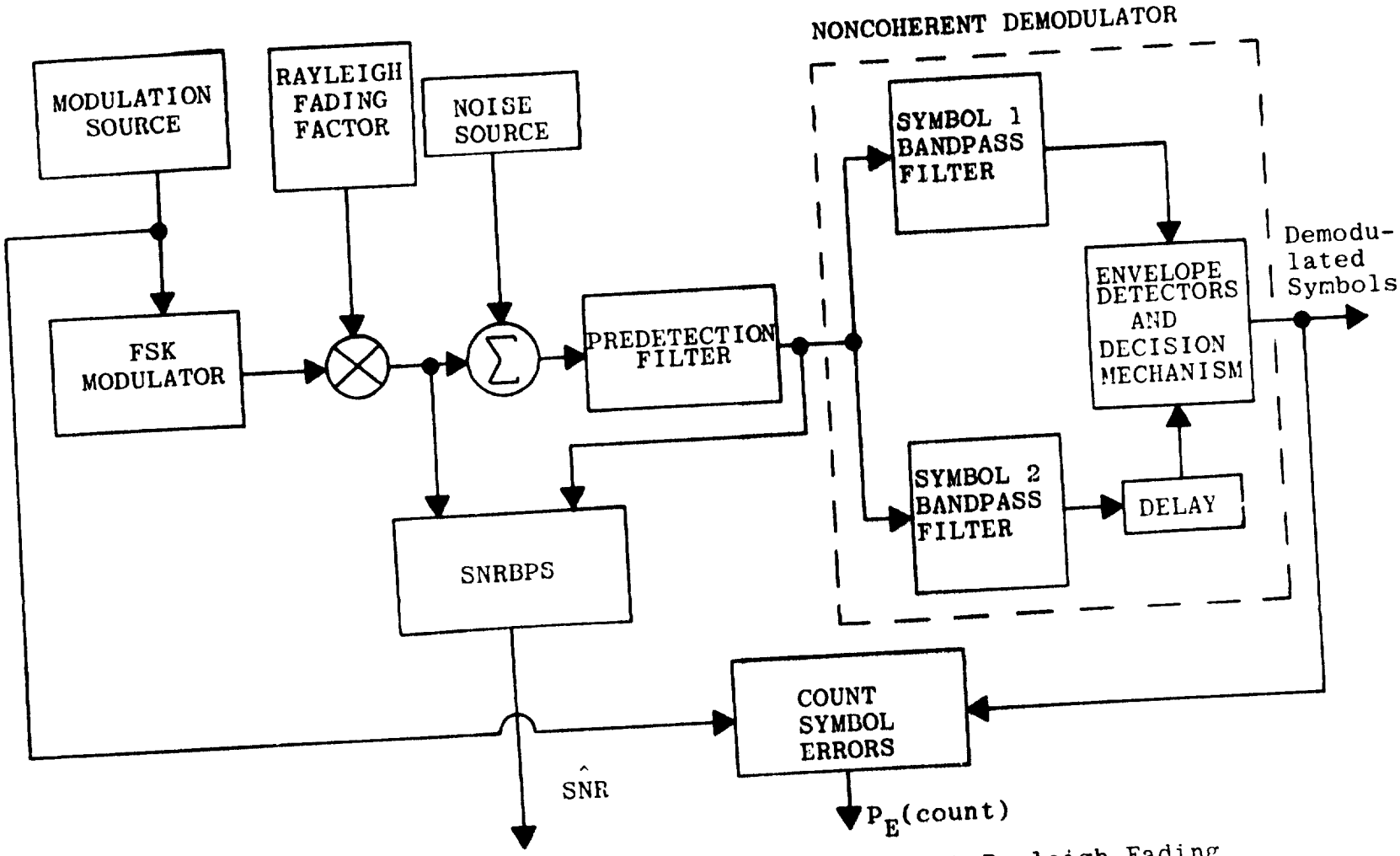

Figure 30. Simulation of the Noncoherent FSK System with Rayleigh Fading

violated. The adjustment of fade duration was performed by controlling the spectral content of the independent Gaussian random variables utilized. Independent Gaussian, equal variance random variables were used as inputs to two identical bandpass filters. Since the two filters are linear systems, the outputs remain Gaussian. Thus two independent Gaussian random variables of equal variance were available at the filter outputs at each sample time to form a Rayleigh distributed random variable. The approximate duration of the fades was controlled by identical adjustment of the two filters. The desired fade duration was achieved by experimenting with the filters and observing the results. Statistical tests were performed to verify the independence of the Gaussian random variables between the two filter outputs and to verify that the distribution of the final random variable generated was a good approximation to a Rayleigh distribution.

The value of P_E for the Rayleigh fading channel is given by [72]

$$P_E = \frac{1}{2 + \frac{\bar{E}_s}{N_o}}$$

where $\bar{E}_s$ is the time-averaged symbol energy present at the demodulator input and N_o is the power spectral density of the white Gaussian noise. This mapping was used to obtain estimates of P_E from the SNR estimates. A large

number of fades were included in the data stream used to compute the SNR, so that an accurate value could be obtained for the time-averaged symbol energy. A table of the results of these simulations is given in Appendix E. A plot similar to those described for the other systems is given in Figure 31. The P_E values obtained by SNR estimation and by direct error count differ in all cases by less than a factor of 1.2. Once again the SNR estimation routine appears to be extremely accurate.

In the final simulations performed for digital communication systems, a diversity transmission system was simulated to combat the Rayleigh fading channel. It was assumed that a number of independent Rayleigh channels having identical statistics were to be utilized. Spatial diversity was actually simulated, although the results would theoretically be the same if the system employed time or frequency diversity, as long as the assumptions of independence and identical statistics remain valid.

The modulation scheme employed was noncoherent FSK. The system configuration for each spatial channel was identical to that of the single channel system just described in the previous simulations except for the decision mechanism in the demodulator. In order to decode a symbol in the diversity system demodulator, first the outputs of the two envelope detectors in each binary channel are squared. Then the squared outputs corresponding to the same symbol in each channel are summed over all the channels.

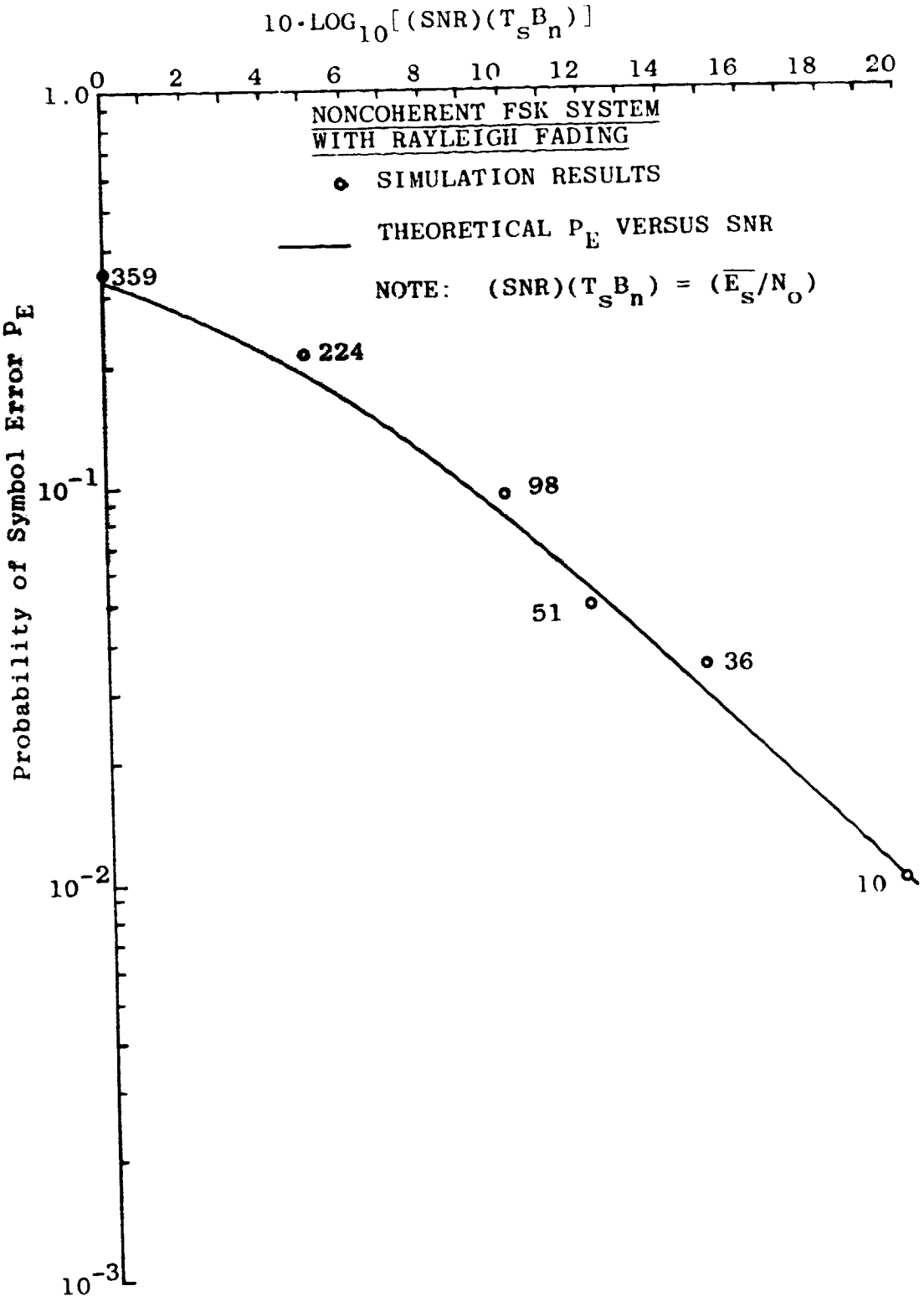

Figure 31. P_E Versus SNR, Rayleigh Fading Channel

Finally the larger of these two sums is chosen to define the decoded output symbol. The structure of this system is illustrated in Figure 32.

The effects of diversity transmission can be seen by assuming a given amount of energy is available for each modulation symbol. Holding this total symbol energy constant, system performance is observed as the energy is divided among an increasing number of independent channels. This procedure was followed in the simulations performed. The value of P_E in this system is given by [73]

$$P_E = p^L \sum_{j=0}^{L-1} \binom{L+j-1}{j} (1-p)^j$$

where

$$p = \frac{1}{2 + \left[\frac{\bar{E}_s}{LN_o}\right]}$$

L = the order of diversity

and $\bar{E}_s$ and N_o are as defined earlier, but remembering that $\bar{E}_s$ represents the total energy summed over all the channels.

In these simulations the ratio $\bar{E}_s/N_o$ was to be fixed at 15 dB. The SNR was estimated for only one channel. It was assumed that the other channels were operating at the same SNR. Each order of diversity from one through

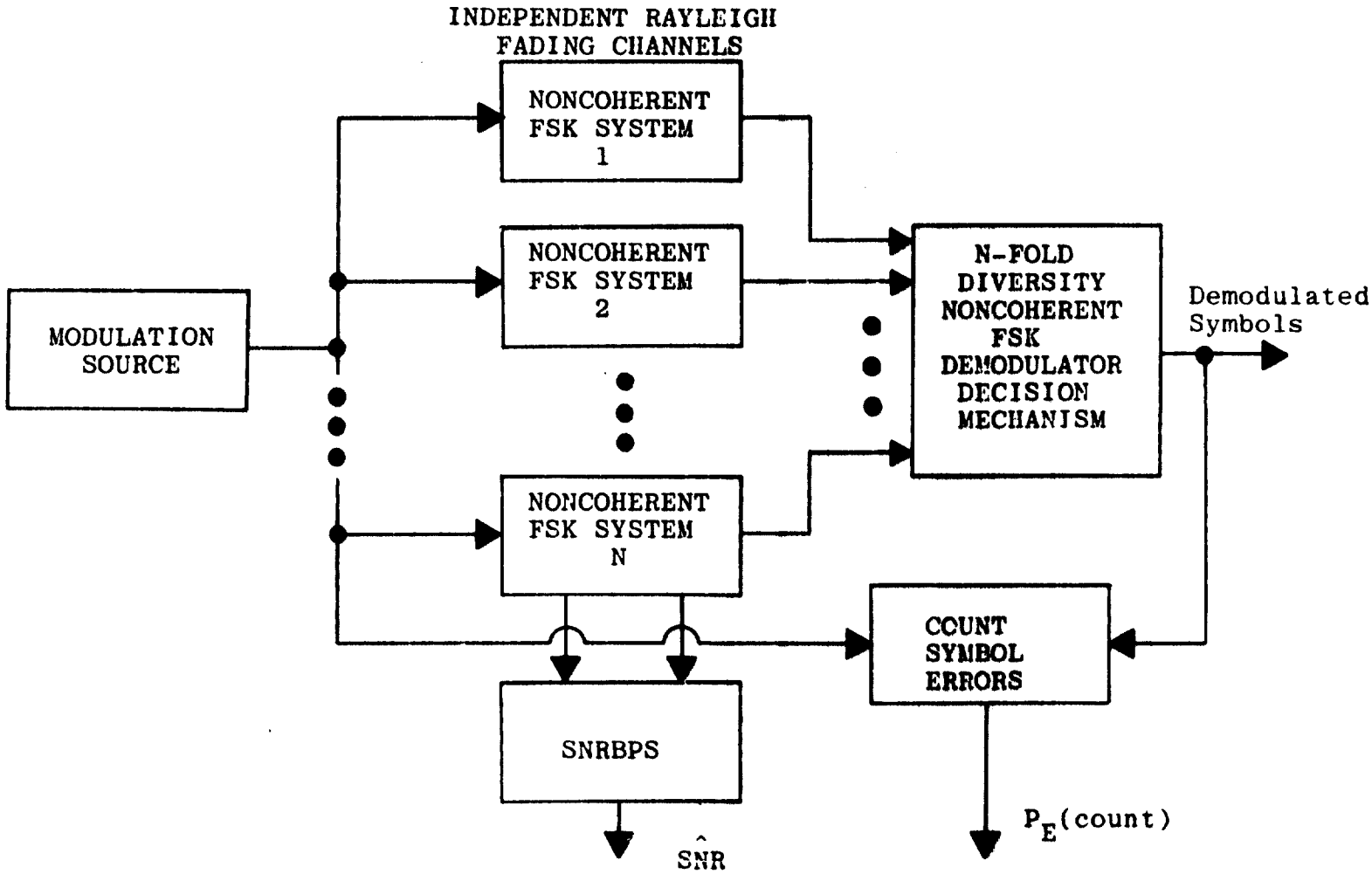

Figure 32. Simulation of the Diversity Transmission System

four was simulated. In each case data were collected from one channel to estimate the SNR, and a computation was performed to obtain $\bar{E}_s/N_o$. The value was held close to 15 dB for each order of diversity by using the SNR estimator to check performance. The number of errors which occurred in each case was used to obtain the probability of error. The results are tabulated in Appendix E. A plot was constructed showing the value obtained for the probability of error at each order of diversity. A curve was drawn indicating the theoretical value of P_E for each order of diversity, assuming that $\bar{E}_s/N_o$ is 15 dB. Of course, the curve is only defined for a practical system at integer values of diversity. The plot is given in Figure 33. The results again are quite good. For diversities of orders less than four, the values of P_E miss the theoretical curve by less than a factor of 1.1. In the fourth order diversity system, the factor is about 1.7.

In this case the SNR estimation routine was not utilized in order to measure an unknown but fixed simulation. Rather, the routine was used to measure the SNR of a simulation to maintain the desired SNR specification for the system. Adjustments of the noise were checked using SNRBPS until the desired SNR was achieved. The SNR estimation routine appears to have been accurately monitoring the SNR, since the number of errors that occurred are in good agreement with the theory. This

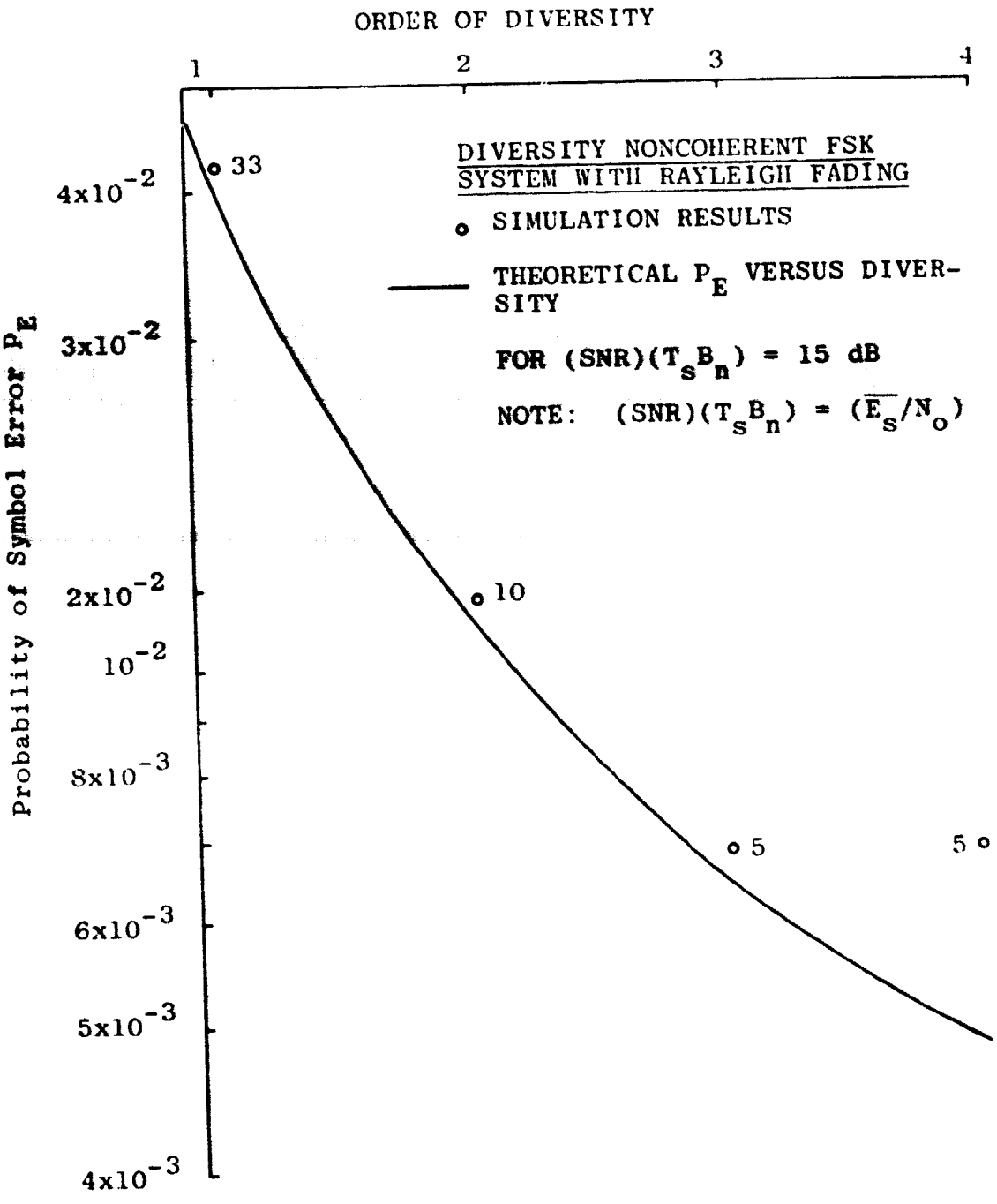

Figure 33. P_E Versus SNR, Diversity Transmission

particular application demonstrates another way the SNR estimation routines may be helpful in evaluating the performance of a communication system through computer simulation.

VIII. CONCLUSION

As digital computer simulation has begun to play a major role in the analysis of many systems, it has been seen that a general criterion for performance evaluation, which is suitable for use in digital simulation, is needed. In many cases a useful criterion is the mean-square error between a measured waveform and some reference. In other cases some other figure of merit for a system can be derived using the mean-square error. In conjunction with measurements of power, the mean-square error can be used to define signal-to-noise ratios for arbitrary waveforms, where the error is considered to be noise. Other parameters of general interest, such as system gain and system delay, can be defined by using measurements of the mean-square error or the signal-to-noise ratio.

In this work, methods were developed for estimating all these quantities in simulations of lowpass and bandpass systems. The technique requires that a reference waveform be available which differs from the ideal signal component of the waveform at the point of measurement only by an unknown scale factor and, in the lowpass case, by an unknown time delay. In the bandpass case, both group and phase delays can be arbitrary. The method can be applied to digital simulations of any wide-sense stationary system where this reference is available. The measurements of power, gain, delay, and signal-to-noise ratio find many

applications in simulations of both analog and digital communication systems.

In order to obtain these estimates, a method was developed to determine the amplitude scaling and the group and phase delays that must be imposed on the reference to form a signal estimate which minimizes the mean-square error with respect to the measurement waveform. Two software implementations were developed. One is for the case where group and phase delay are equal, which means the signal estimate may be formed by simply amplitude scaling and time-shifting the reference. This case is usually associated with lowpass linear systems. The other implementation, which allows for arbitrary group and phase delays, reflects the processing generally encountered in bandpass linear systems. Both of the SNR estimators were proven to be ML estimators in systems with additive white Gaussian noise. Curves showing confidence intervals were developed for the lowpass SNR estimator in this case. These curves indicate that excellent SNR estimates can usually be formed using data sets of a practical size.

Development of confidence intervals for the bandpass estimator proved to be too complicated for simple mathematical analysis. Further work could be done in deriving such curves. It would be helpful to have a more thorough statistical study of the estimators. It may be possible to determine whether the estimators are ML for channel models other than the additive white Gaussian channel, and

confidence intervals would be useful for these channels, also.

The SNR estimators were first tested in applications to analog systems. Results obtained from simulations of a filtered noise system and a more complex FM communication system indicate that accurate estimates can be obtained for a wide range of signal-to-noise ratios. In some cases, the SNR estimator is useful in checking the validity of the simulation itself. For example, in the FM system which was simulated, the SNR measurements helped to verify the proper operation of the PLL demodulator simulation model.

A survey of the techniques currently available for evaluating the performance of digital communication systems through the use of computer simulation revealed that the methods generally in use are not suitable for evaluation of most practical systems. The parameter most often desired in evaluating these systems is the probability of symbol error, P_E. The estimation of P_E based upon counting the errors which occur, or extensions of error counting techniques, require too much computer time to be feasible for values of P_E less than about 10^{-5}, which are typical in most systems.

In many cases a mapping from P_E to the SNR at some point in the system can be found, or at least approximated, using either mathematical analysis or empirical data. In such systems, P_E may be estimated through the measurement

of the SNR. In some cases the SNR itself is a useful parameter for evaluating performance.

The technique developed for SNR estimation was applied to simulations of a variety of digital communication systems. Estimates of P_E obtained through SNR measurements were compared with results obtained by direct error counting. The tests directly verified that accurate estimates for P_E can be formed using the SNR estimators over the range of the SNR associated with error probabilities where error counting is feasible. Since measurements can be accurately made for a much higher range of the SNR, all indications are that accurate estimates for P_E can be found over ranges of P_E where the error counting methods cannot be applied successfully.

In short, the results of all tests show that the SNR estimation methods developed accomplish their task quite well. Accurate measurements have been obtained for a wide range of signal-to-noise ratios when applied to simulations of both digital and analog communication systems. The accurate estimation of P_E has been directly verified for error probabilities greater than 10^{-3} through simulation. In almost all cases the probability of error was estimated to within a factor of 1.5. It seems accuracy similar to this should be expected for the much smaller values of P_E common in most practical systems. The accuracy of the P_E estimate would be decreased slightly in many cases becuase, as the SNR increases, a given error

in the SNR estimate usually produces larger errors in the estimate for P_E.

The method is economically feasible. The amount of computer time required to determine an SNR estimate for producing a P_E estimate of 10^{-6} is essentially the same as for producing a P_E estimate of 10^{-3}, since the requirement on computer time is only a function of the number of samples being processed. The various measurements for delay, gain, and correlation which are generated in the process are often helpful, also.

The techniques developed seem to offer a versatile tool for the accurate quantitative evaluation of many different systems through the use of computer simulation.

LIST OF REFERENCES

1. W. H. Tranter and C. T. Dawson, "SYSTID--A Flexible Tool for the Analysis of Communications," Conference Record, 1972 IEEE National Telecommunications Conference, Houston, 1972, pp. 38C-1 - 38C-7.

2. L. B. Milstein and Mark C. Austin, "Performance of Noncoherent FSK and AM-FSK Systems with Postdetection Filtering," IEEE Trans. on Communications, Vol. COM-23, Nov. 1975, pp. 1300-1306.

3. S. Lebowitz and L. Palmer, "Simulation Results for Intersymbol Interference Loss; QPSK Transmission Through Several Filter Types," 1973 IEEE National Telecommunications Conference, Atlanta, Nov. 1973, pp. 32D-1 - 32D-7.

4. T. C. Huang and F. J. Ricci, "Hybrid Computer Simulation and Performance Assessment of a Partial Response Signaling System," 1973 IEEE National Telecommunications Conference, Atlanta, Nov. 1973, pp. 4E-1 - 4E-8.

5. L. Davisson and L. Milstein, "On the Performance of Digital Communications Systems with Bandpass Limiters--Part II," IEEE Trans. on Communications, Vol. COM-20, Oct. 1972, pp. 975-980.

6. P. Bello and L. Ehrman, "Performance of an Energy Detection FSK Digital Modem for Troposcatter Links," IEEE Trans. on Communication Tech., Vol. COM-17, April 1969, pp. 192-200.

7. Huang and Ricci, loc. cit.

8. B. J. Leon and others, "A Bit Error Rate Monitor for Digital PSK Links," IEEE Trans. on Communications, Vol. COM-23, April 1975, pp. 518-525.

9. B. J. Leon and R. T. Kitahara, "Monitor Simulation of the NEC TRP QPSK Receiver," 1973 IEEE National Telecommunications Conference, Atlanta, Nov. 1973, pp. 4D-1 - 4D-8.

10. P. Bello and T. Crystal, "A Class of Efficient High-Speed Digital Modems for Troposcatter Links," IEEE Trans. on Communication Tech., Vol. COM-17, April 1969, pp. 162-183.

11. P. K. Chaterjee and V. R. Rau, "Digital-Computer-Simulation Results of Delta-Modulation Systems," Proceedings of IEE, Vol. 120, Nov. 1973, pp. 1379-1382.

12. R. E. Lawrence and H. Kaufman, "The Kalman Filter for the Equalization of a Digital Communications Channel," IEEE Trans. on Communication Tech., Vol. COM-19, Dec. 1971, pp. 1137-1141.

13. D. L. Hedderly and L. Lundquist, "Computer Simulation of a Digital Satellite Communications Link," IEEE Trans. on Communications, Vol. COM-21, Apr. 1973, pp. 321-325.

14. Bello and Ehrman, loc. cit.

15. S. Singh and others, "Digital Single-Sideband Modulation," IEEE Trans. on Communications, Vol. COM-21, Mar. 1973, pp. 255-262.

16. Milstein and Austin, loc. cit.

17. Huang and Ricci, loc. cit.

18. Bello and Crystal, loc. cit.

19. Leon and Kitahara, loc. cit.

20. Leon and others, loc. cit.

21. Bello and Crystal, loc. cit.

22. E. J. Gumbel, Statistics of Extremes. New York: Columbia University Press, 1958.

23. E. C. Posner, "The Application of Extreme Value Theory to Error Free Communication," Technometrics, Vol. 7, Nov. 1965, pp. 517-529.

24. M. C. Jeruchim, "On the Estimation of Error Probability Using Generalized Extreme-Value Theory," IEEE Trans. on Information Theory, Vol. IT-22, Jan. 1976, pp. 108-110.

25. M. Schwartz, "Extremal Statistics in Computer Simulation of Digital Communication Systems," Proceedings of the National Electronics Conference, Vol. 25, 1969, pp. 438-440.

26. L. B. Milstein, "On the Application of Extreme-Value Theory to Bit Error Rate Monitoring," Proceedings of the 1976 International Conference on Communications, Philadelphia, June 1976.

27. Schwartz, loc. cit.

28. Posner, loc. cit.

29. Milstein, loc. cit.

30. Schwartz, loc. cit.

31. D. J. Gooding, "Performance Monitor Techniques for Digital Receivers Based on Extrapolation of Error Rate," IEEE Trans. on Communication Tech., Vol. COM-16, June 1968, pp. 380-387.

32. Leon, loc. cit.

33. Leon and Kitahara, loc. cit.

34. J. L. Hammond and others, "Extrapolation Monitors for Digital PSK Links," 1973 IEEE National Telecommunications Conference, Atlanta, Nov. 1973, pp. 4C-1 - 4C-8.

35. Leon and Kitahara, loc. cit.

36. S. B. Weinstein, "Estimation of Small Probabilities by Linearization of the Tail of a Probability Distribution Function," IEEE Trans. on Communication Tech., Vol. COM-19, Dec. 1971, pp. 1149-1155.

37. R. M. Gagliardi and C. M. Thomas, "PCM Data Reliability Monitoring Through Estimation of Signal-to-Noise Ratio," IEEE Trans. on Communication Tech., Vol. COM-16, June 1968, pp. 479-486.

38. Ibid.

39. N. E. Nahi and R. M. Gagliardi, "Use of Limiters for Estimating SNR," IEEE Trans. on Information Theory, Vol. IT-13, Jan. 1967, pp. 127-128.

40. D. Smith, "Performance Assessment of Digital Transmission Systems," 1973 IEEE National Telecommunications Conference, Atlanta, Nov. 1973, pp. 4F-1 - 4F-6.

41. R. C. Houts and R. S. Simpson, "Analysis of Waveform Distortion in Linear Systems," IEEE Trans. on Education, Vol. E-11, June 1968, pp. 122-125.

42. T. H. Shepertycki, "Telemetry Error Measurements Using Pseudo-Random Signals," IEEE Trans. on Space Electronics and Telemetry, Vol. SET-10, Sept. 1964, pp. 111-115.

43. R. S. Simpson and R. C. Houts, "A Definition of Average Time Delay for a Linear System," Proceedings of the IEEE (Letters), Vol. 55, Oct. 1967, pp. 1733-1734.

44. Houts and Simpson, loc. cit.

45. Gagliardi and Thomas, loc. cit.

46. H. Scheffe, The Analysis of Variance. New York: John Wiley & Sons, Inc., 1959, pp. 41-52.

47. Gagliardi and Thomas, loc. cit.

48. A. M. Mood and others, Introduction to the Theory of Statistics. New York: McGraw-Hill, 1974, pp. 241-246.

49. Scheffe, loc. cit.

50. Mood and others, op. cit., pp. 246-249.

51. M. Abramowitz and I. A. Stegun, eds., Handbook of Mathematical Functions. New York: Dover Publications, Inc., 1970, pp. 946-947.

52. Ibid., pp. 931-948.

53. A. V. Oppenheim and C. J. Weinstein, "Effects of Finite Register Length in Digital Filtering and the Fast Fourier Transform," Proceedings of the IEEE, Vol. 60, Aug. 1972, pp. 122-125.

54. G. D. Bergland, "A Guided Tour of the Fast Fourier Transform," IEEE Spectrum, Vol. 6, July 1969, pp. 41-52.

55. H. Taub and D. L. Schilling, Principles of Communication Systems. New York: McGraw-Hill, 1971, pp. 25-26.

56. W. D. Stanley, Digital Signal Processing. Reston, Virginia: Reston Publishing Co., 1975, pp. 117-121.

57. Taub and Schilling, op. cit., pp. 242-256.

58. Ibid.

59. Bergland, loc. cit.

60. H. Cramér, Mathematical Methods of Statistics. Princeton: Princeton University Press, 1946, pp. 500-504.

61. Stanley, op. cit., pp. 239-249.

62. Bergland, loc. cit.

63. A. Papoulis, Probability, Random Variables, and Stochastic Processes. New York: McGraw-Hill, 1965, pp. 355, 357.

64. Ibid.

65. B. Gold, A. V. Oppenheim, and C. M. Rader, "Theory and Implementation of the Discrete Hilbert Transform," Proc. Symp. Comput. Process. Commun., 1970, pp. 235-250.

66. W. H. Tranter and M. D. Turner, Phase-Lock Loop Models for SYSTID Simulations, University of Missouri Technical Report CSR-77-3, March 1977.

67. R. E. Ziemer and W. H. Tranter, Principles of Communications. Boston: Houghton Mifflin Co., 1976, pp. 277-291.

68. Taub and Schilling, op. cit., pp. 295-339.

69. Ziemer and Tranter, op. cit., pp. 319-320.

70. Ibid., p. 323.

71. Ibid., pp. 326-331.

72. J. M. Wozencraft and I. M. Jacobs, Principles of Communication Engineering. New York: John Wiley & Sons, Inc., 1965, pp. 527-533.

73. Ibid., pp. 533-545.

74. Taub and Schilling, op. cit., pp. 25-26.

75. Ziemer and Tranter, op. cit., pp. 236-237.

76. Ibid.

77. Ibid., pp. 321-322.

78. Ibid., pp. 333-336.

BIBLIOGRAPHY

Books:

Abramowitz, M. and Stegun, I. A., eds., Handbook of Mathematical Functions. New York: Dover Publications, Inc., 1970.

Cramer, H., Mathematical Methods of Statistics. Princeton University Press, 1946.

Gold, B., Oppenheim, A. V., and Rader, C. M., "Theory and Implementation of the Discrete Hilbert Transform," Proc. Symp. Comput. Process Commun., 1970.

Gumbel, E. J., Statistics of Extremes. New York: Columbia University Press, 1958.

Hammond, J. L. and others, "Extrapolation Monitors for Digital PSK Links," 1973 IEEE National Telecommunications Conference, Atlanta, Nov. 1973.

Huang, T. C. and Ricci, F. J., "Hybrid Computer Simulation and Performance Assessment of a Partial Response Signaling System," 1973 IEEE National Telecommunications Conference, Atlanta, Nov. 1973.

Lebowitz, S. and Palmer, L., "Simulation Results for Intersymbol Interference; QPSK Transmission Through Several Filter Types," 1973 IEEE National Telecommunications Conference, Atlanta, Nov. 1973.

Leon, B. J. and Kitahara, R. T., "Monitor Simulation of the NEC TRP QPSK Receiver," 1973 IEEE National Telecommunications Conference, Atlanta, Nov. 1973.

Milstein, L. B., "On the Application of Extreme-Value Theory to Bit Error Rate Monitoring," Proceedings of the 1976 International Conference on Communications, Philadelphia, June 1976.

Mood, A. M. and others, Introduction to the Theory of Statistics. New York: McGraw-Hill, 1974.

Papoulis, A., Probability, Random Variables, and Stochastic Processes. New York: McGraw-Hill, 1965.

Scheffe, H., The Analysis of Variance. New York: John Wiley & Sons, Inc., 1959.

Schwartz, M., "Extremal Statistics in Computer Simulation of Digital Communication Systems," Proceedings of the National Electronics Conference, Vol. 25, 1969.

Stanley, W. D., Digital Signal Processing. Reston, Virginia: Reston Publishing Co., 1975.

Taub, H. and Schilling, D. L., Principles of Communication Systems. New York: McGraw-Hill, 1971.

Tranter, W. H. and Dawson, C. T., "SYSTID--A Flexible Tool for the Analysis of Communications," Conference Record, 1972 IEEE National Telecommunications Conference, Houston, 1972.

Tranter, W. H. and Turner, M. D., Phase-Lock Loop Models for SYSTID Simulations, University of Missouri Technical Report CSR-77-3, March 1977.

Wozencraft, J. M. and Jacobs, I. M., Principles of Communication Engineering. New York: John Wiley & Sons, Inc., 1965.

Ziemer, R. E. and Tranter, W. H., Principles of Communications. Boston: Houghton Mifflin Co., 1976.

Periodicals:

Bello, P. and Crystal, T., "A Class of Efficient High-Speed Digital Modems for Troposcatter Links," IEEE Trans. on Communication Tech., Vol. COM-17, April 1969.

Bello, P. and Ehrman, L., "Performance of an Energy Detection FSK Digital Modem for Troposcatter Links," IEEE Trans. on Communication Tech., Vol. COM-17, April 1969.

Bergland, G. D., "A Guided Tour of the Fast Fourier Transform," IEEE Spectrum, Vol. 6, July 1969.

Chatterjee, P. K. and Rau, V. R., "Digital-Computer-Simulation Results of Delta-Modulation Systems," Proceedings of IEEE, Vol. 120, Nov. 1973.

Davisson, L. and Milstein, L., "On the Performance of Digital Communications Systems with Bandpass Limiters--Part II," IEEE Trans. on Communications, Vol. COM-20, Oct. 1972.

Gagliardi, R. M. and Thomas C. M., "PCM Data Reliability Monitoring Through Estimation of Signal-to-Noise Ratio," IEEE Trans. on Communication Tech., Vol. COM-16, June 1968.

Gooding, D. J., "Performance Monitor Techniques for Digital Receivers Based on Extrapolation of Error Rate," IEEE Trans. on Communication Tech., Vol. COM-16, June 1968.

Hedderly, D. J. and Lundquist, L., "Computer Simulation of a Digital Satellite Communications Link," IEEE Trans. on Communications, Vol. COM-21, Apr. 1973.

Houts, R. C. and Simpson, R. S., "Analysis of Waveform Distortion in Linear Systems," IEEE Trans. on Education, Vol. E-11, June 1968.

Jeruchim, M. C., "On the Estimation of Error Probability Using Generalized Extreme-Value Theory," IEEE Trans. on Information Theory, Vol. IT-22, Jan. 1976.

Lawrence, R. E. and Kaufman H., "The Kalman Filter for the Equalization of a Digital Communications Channel," IEEE Trans. on Communication Tech., Vol. COM-19, Dec. 1971.

Leon, B. J. and others, "A Bit Error Rate Monitor for Digital PSK Links," IEEE Trans. on Communications, Vol. COM-23, April 1975.

Milstein, L. B. and Austin, Mark C., "Performance of Noncoherent FSK and AM-FSK Systems with Postdetection Theory," IEEE Trans. on Communications, Vol. COM-23, Nov. 1975.

Nahi, N. E. and Gagliardi, R. M., "Use of Limiters for Estimating SNR," IEEE Trans. on Information Theory, Vol. IT-13, Jan. 1967.

Oppenheim, A. V. and Weinstein, C. J., "Effects of Finite Register Length in Digital Filtering and the Fast Fourier Transform," Proceedings of the IEEE, Vol. 60, Aug. 1972.

Posner, E. C., "The Application of Extreme Value Theory to Error Free Communication," Technometrics, Vol. 7, Nov. 1965.

Shepertycki, T. H., "Telemetry Error Measurements Using Pseudo-Random Signals," IEEE Trans. on Space Electronics and Telemetry, Vol. SET-10, Sept. 1964.

Simpson, R. S. and Houts, R. C., "A Definition of Average Time Delay for a Linear System," Proceedings of the IEEE (Letters), Vol. 55, Oct. 1967.

Singh, S. and others, "Digital Single-Sideband Modulation," IEEE Trans. on Communications, Vol. COM-21, Mar. 1973.

Smith, D., "Performance Assessment of Digital Transmission Systems," 1973 National Telecommunications Conference, Atlanta, Nov. 1973.

Weinstein, S. B., "Estimation of Small Probabilities by Linearization of the Tail of a Probability Distribution Function," IEEE Trans. on Communication Tech., Vol. COM-19, Dec. 1971.

APPENDICES

APPENDIX A

SUBROUTINE SNRMSE

Subroutine SNRMSE is a FORTRAN IV implementation of one of the SNR estimators developed for use in computer simulation. It is applicable to systems where the group and phase delays of the measurement data with respect to the reference data are equal. These are usually, but not necessarily, lowpass linear systems. For proper estimation of delay, the data should be wide-sense stationary.

The measurement data set of K samples is input to the routine in the Y array, and the X array contains the reference data. Included in the argument list are the various work arrays and dimension values required and input parameters to specify various processing options. These inputs include the choice of the FFT block size to be used, whether an initial delay estimate is to be determined by the routine or specified in the argument list, the choice of the number of iterations to be performed in choosing the final delay estimate, whether to apply a data window, and whether the cross-correlation function is to be printed, plotted, both, or neither. All the arguments of the subroutine are listed with explanations in the comments at the beginning of the program listing which is included.

The execution of SNRMSE does not destroy any of the input data. Thus multiple calls of the routine or successive processing of the data set by other routines is

possible. The routine provides a printout of estimates for power in X, power in Y, system gain, system delay, the peak of the cross-correlation function, normalized mean-square error, and the SNR estimate expressed in dB. Most of these values are also returned in the argument list so that they may be used in any successive processing. Also printed are a list of the values of the input parameters in effect upon execution of the routine, the normalized cross-correlation values used to select the final delay estimate, and the means and variances of the FFT data blocks which were computed in testing stationarity.

If any stationarity tests were failed, a warning message is printed to that effect. The tests are made by computing the mean and variance of each FFT data block and comparing these values with those of the entire X or Y data set. If a value outside a certain experimentally chosen range is found, the warning is generated. To aid in locating the problem which generated this warning, immediately following the warning an output of the form IFLAG = I will be printed, where I is a signed integer. The magnitude of I indicates the number of the FFT measurement data block which caused the warning, except that a value of 10,000 denotes the reference block. A minus sign indicates a variance test was failed. Otherwise, a test of the mean was failed. If more than one test was failed, only the last failure which was detected is

indicated by IFLAG. A warning statement does not necessarily mean that the results are in error, only that the stationarity is questionable. This often causes the delay estimate, and hence the other results, to be erroneous. If the correct delay value has been found in spite of the nonstationarity, then the results are all valid.

The subroutines CORR, GRAPH, and DATAPT, which are also listed, are called by SNRMSE and require no attention by the user, except that they must be available to SNRMSE. The subroutine CORR is used to obtain the cross-correlation function, and GRAPH and DATAPT are used to graph this function if specified. Also, the FFT routine HARM must be available for use in SNRMSE.

One parameter which needs some explanation is ISIGNR. This input specifies whether the cross-correlation peak to be found in determining delay is to be positive, negative, or the largest absolute value computed. For most cases the user should specify ISIGNR = 0, indicating that the absolute value peak is to be used, and the routine will execute properly. Only for a special class of signals does the user need to specify whether a positive or negative peak is required. If the signals being analyzed possess the proper symmetry such that the cross-correlation function theoretically has equal magnitude positive and negative peaks, then the user must specify which sign is to be used in order to obtain correct delay estimates.

This would be the case for signals with half-wave symmetry. It should be noted that in such cases the estimates for the SNR, power, and error are valid regardless of the specification chosen for ISIGNR. Only if the user is concerned with delay or gain estimates in a system with theoretically equal magnitude positive and negative cross-correlation peaks, does ISIGNR need to be specified as 1 or 0. In such a case, a 1 should be used if the system gain is positive, and a -1 should be used if the system gain is negative.

```
      SUBROUTINE SNRMSE(X,M1,Y,K,XT,NT,YT,IWIN,NEWTAU,ITAU,PX,PY,RTMAX,
     1SNR,ERRN,INV,S,ND,RXY,ISIGNR,ISAVER,ILROAM)
C*****************************************************************C
C                                                                 C
C                        SUBROUTINE SNRMSE                        C
C                                                                 C
C      ROUTINE TO MEASURE SIGNAL-TO-'NOISE' RATIO, NORMALIZED     C
C     MEAN SQUARE ERROR, DELAY, AND POWER FOR A REFERENCE ARRAY X C
C     AND AN ARRAY OF MEASUREMENT DATA Y                          C
C                                                                 C
C     SNRMSE REQUIRES SUBROUTINE CORR IF NEWTAU = 1 IS SPECIFIED  C
C                                                                 C
C                                                                 C
C                  *****INPUT PARAMETERS*****                     C
C                                                                 C
C     X - THE ARRAY OF REFERENCE DATA                             C
C     M1 - DETERMINES SIZE OF REFERENCE BLOCK TO BE TRANSFORMED IN C
C          FINDING DELAY.  N=2**M1 EQUALS NUMBER OF POINTS USED   C
C          FOR FFT                                                C
C     Y - THE ARRAY OF MEASUREMENT DATA                           C
C     K - THE NUMBER OF DATA POINTS IN X AND Y ARRAYS.  DIMENSION OF C
C         X,Y,AND RXY                                             C
C     XT - WORK ARRAY FOR REFERENCE TRANSFORM                     C
C     NT - DIMENSION OF XT AND YT.  NT=2*2**M1                    C
C     YT - WORK ARRAY FOR MEASUREMENT BLOCK TRANSFORMS            C
C     IWIN - SPECIFIES WHETHER TO WINDOW DATA IN COMPUTING POWERS C
C            AND FINAL CROSSCORRELATION                           C
C              0 - DO NOT WINDOW DATA                             C
C              1 - WINDOW DATA                                    C
C     NEWTAU - SPECIFIES WHETHER TO COMPUTE A NEW DELAY VALUE OR  C
C              TO USE THE INPUT VALUE OF ITAU                     C
C                0 - DO NOT COMPUTE A NEW DELAY VALUE.  USE THE   C
C                    INPUT VALUE ASSIGNED TO ITAU                 C
C                1 - COMPUTE A NEW DELAY VALUE                    C
```

```
C      INV - WORK ARRAY FOR HARM FFT. DIMENSION IS NT/8                    C
C      S - WORK ARRAY FOR HARM FFT. DIMENSION IS NT/8                      C
C      ND - DIMENSION OF INV AND S.  ND = NT/8                             C
C      RXY - ARRAY TO STORE CROSSCORRELATION FUNCTION.  DIMENSION IS K     C
C      ISIGNR - DETERMINES WHETHER TO USE POSITIVE PEAK, NEGATIVE PEAK,    C
C               OR ABSOLUTE VALUE PEAK OF CROSSCORRELATION FUNCTION IN     C
C               FINDING DELAY.                                            C
C                 -1 - USE NEGATIVE PEAK                                  C
C                  0 - USE ABSOLUTE VALUE PEAK                            C
C                  1 - USE POSITIVE PEAK                                  C
C      ISAVER - DETERMINES WHETHER OR NOT TO PRINT AND/OR PLOT THE         C
C               CROSSCORRELATION FUNCTION                                 C
C                  0 - DO NOT PRINT OR PLOT RXY                           C
C                  1 - PRINT RXY                                          C
C                  2 - PLOT RXY ON THE PRINTER                            C
C                  3 - PRINT AND PLOT RXY                                 C
C      ILROAM - DETERMINES HOW MANY POINTS TO SEARCH FOR THE PEAK          C
C              CROSSCORRELATION.  ANY ODD INTEGER GREATER THAN  OR        C
C              EQUAL TO 1 IS VALID, BUT LARGE NUMBERS (SAY,               C
C              GREATER THAN ABOUT 7 OR 9) WILL INCREASE RUN TIME          C
C              SIGNIFICANTLY                                              C
C      ITAU - IF NEWTAU IS SPECIFIED 0, THE INPUT VALUE OF ITAU            C
C             WILL BE USED FOR DELAY VALUE IN COMPUTING ALL OUTPUTS       C
C             (SEE ITAU BELOW, ALSO)                                      C
C                                                                         C
C                   *****OUTPUT VALUES*****                               C
C                                                                         C
C      ITAU - NUMBER OF STEPSIZE DELAYS TO PEAK OF NORMALIZED              C
C             CROSSCORRELATION FUNCTION IF NEWTAU IS 1.  IF               C
C             NEWTAU IS 0, OUTPUT ITAU EQUALS INPUT ITAU.                 C
C      PX - POWER IN X ARRAY                                              C
C      PY - POWER IN Y ARRAY                                              C
C      RTMAX - PEAK VALUE OF CROSSCORRELATION FUNCTION                    C
C      SNR - SIGNAL TO NOISE RATIO                                        C
```

```
C      ERRN - NORMALIZED MEAN SQUARE ERROR                                    C
C                                                                             C
C*************************************************************************C
      REAL*8 XYS,XYSM,XYSN,XYSNM,XSS,YSS,XSST,YSST
      DIMENSION M(3),X(K),Y(K),XT(NT),YT(NT),INV(ND),S(ND),RXY(K)
      IFLAG = 0
      IROAM = ILROAM
C
C     BRANCH IF THE PREVIOUSLY COMPUTED VALUE OF ITAU IS TO BE USED
C     AS THE CURRENT DELAY VALUE
C
      IF (NEWTAU.NE.0) GO TO 2
      JTAU = ITAU+1
      GO TO 30
C
C     INITIALIZE PARAMETERS
C
    2 M(1) = M1
      M(2) = 0
      M(3) = 0
      N = 2**M1
      LL = N/2
      XM = 0.
      X2 = 0.
      WRITE (6,209) K,N,IWIN,NEWTAU,ISIGNR,ILROAM,ISAVER
C
C     COMPUTE MEAN AND VARIANCE OF REFERENCE BLOCK
C
      DO 5 I=1,LL
      X2 = X2+X(I)**2
    5 XM = XM+X(I)
      XM = XM/FLOAT(LL)
      XVAR = (X2/FLOAT(LL))-XM**2
C
```

```
C       COMPUTE MEAN AND VARIANCE OF ENTIRE X AND Y ARRAYS
C
        XMN = 0.
        YMN = 0.
        X2 = 0.
        Y2 = 0.
        DO 8 I=1,K
        XMN = XMN + X(I)
        YMN = YMN + Y(I)
        X2 = X2+X(I)**2
      8 Y2 = Y2 + Y(I)**2
        XMN = XMN/FLOAT(K)
        YMN = YMN/FLOAT(K)
        XVAR2 = X2/FLOAT(K)-XMN**2
        YVAR2 = Y2/FLOAT(K)-YMN**2
C
C       SET UP STATIONARITY TEST PARAMETERS
C
        XVTEST = XVAR2/2.
        XMTEST = SQRT(XVAR2)/2.
        YVTEST = YVAR2/2.
        YMTEST = SQRT(YVAR2)/2.
C
C       TEST STATIONARITY OF REFERENCE BLOCK
C
        IF (ABS(XMN-XM).GT.XMTEST) IFLAG = 10000
        IF (ABS(XVAR2-XVAR).GT.XVTEST) IFLAG = -10000
        WRITE (6,206) XMN,XVAR2,YMN,YVAR2
        WRITE (6,200) XM,XVAR
C
C       PAD WITH ZEROES AND FFT THE REFERENCE BLOCK
C
        DO 10 I=1,LL
        XT(3*I) = 0.
```

```
      XT(4*I) = 0.
      J = 2*I
      XT(J) = 0.
   10 XT(J-1) = X(I)
      CALL HARM(XT,M,INV,S,-1,IF)
C
C     COMPUTE CROSSCORRELATION FUNCTION AND LOCATE ITS PEAK
C
      NBLOCK = K/LL-1
      DO 20 I=1,NBLOCK
      CALL CORR(Y,K,XT,NT,LL,YT,I,JTAU,INV,S,ND,M,RXY,ISIGNR,ISAVER,YM,
     1YVAR)
C
C     TEST STATIONARITY OF DATA BLOCK FROM Y ARRAY
C
      IF(ABS(YMN-YM).GT.YMTEST) IFLAG = 1
      IF (ABS(YVAR2-YVAR).GT.YVTEST) IFLAG = -1
      WRITE (6,201) I,YM,YVAR
   20 CONTINUE
C
C     SET BOUNDARIES FOR SEARCH FOR LOCATION OF PEAK CORRELATION
C     AROUND THE FFT INDICATED VALUE
C
   30 IBEGIN = JTAU-(IROAM-1)/2
      IF (IBEGIN.LT.1) IBEGIN = 1
      IQUIT = IBEGIN+IROAM-1
      IF (IQUIT.GT.K) IQUIT = K
      IF (IQUIT.EQ.K) IBEGIN = IQUIT - IROAM + 1
      IF (IBEGIN.LT.1) IBEGIN = 1
      XYSNM = 0.0D0
      WRITE (6,205)
C
C     PERFORM DOUBLE PRECISION COMPUTATIONS OF POWER IN X ARRAY,
C     POWER IN Y ARRAY, CROSSCORRELATION FUNCTION, AND NORMALIZED
```

```
C      CROSSCORRELATION FUNCTION FOR THE SPECIFIED SEARCH RANGE
C
       DO 48 KTAU=IBEGIN,IQUIT
       I=0
       XSST = 0.0D0
       YSST = 0.0D0
       XYS = 0.0D0
       LTAU = KTAU-1
       KL = K-KTAU+1
       DO 44 J=KTAU,K
       I = I+1
       XSST = XSST + (DBLE(X(I)))**2
       YSST = YSST + (DBLE(Y(J)))**2
   44  XYS = XYS+DBLE(X(I))*DBLE(Y(J))
       XYSN = XYS/DSQRT(XSST*YSST)
       RTM = SNGL(XYSN)
C
C      LOCATE PEAK OF NORMALIZED CROSSCORRELATION FUNCTION
C
       IF(XYSN.LE.XYSNM.AND.ISIGNR.GT.0) GO TO 48
       IF(XYSN.GE.XYSNM.AND.ISIGNR.LT.0) GO TO 48
       IF(DABS(XYSN).LE.DABS(XYSNM).AND.ISIGNR.EQ.0) GO TO 48
       XYSNM = XYSN
       ITAU = LTAU
       XSS = XSST
       YSS = YSST
       XYSM = XYS
   48  WRITE(6,207) LTAU,RTM
       NPTS = K-ITAU
       WRITE (6,210) NPTS
       KTAU = ITAU+1
       KL = K-KTAU+1
C
C      IF WINDOW IS SPECIFIED, RECOMPUTE POWERS AND CROSSCORRELATION
```

```
C       VALUE USING THE HAMMING WINDOW.
C
        IF (IWIN.EQ.0) GO TO 42
        I = 0
        XSS = 0.0D0
        YSS = 0.0D0
        XYSM = 0.0D0
        DO 49 J=KTAU,K
        I = I+1
        W = 0.8566035 - 0.7296993*COS((6.2831853*(FLOAT(I)-.5))/FLOAT(KL))
        XSS = XSS+(DBLE(W*X(I)))**2
        YSS = YSS+(DBLE(W*Y(J)))**2
   49   XYSM = XYSM+DBLE(W*X(I))*DBLE(W*Y(J))
   42   RTMAX = SNGL(XYSM/DFLOAT(KL))
        L = K-ITAU
C
C       COMPUTE AND PRINT NUMBER OF DELAYS, POWER IN X, POWER IN Y,
C       CROSSCORRELATION FUNCTION VALUE, SIGNAL TO NOISE RATIO, GAIN,
C       AND THE NORMALIZED MEAN SQUARE ERROR, COMPUTED USING THE DELAY
C       INDICATED BY THE PEAK OF THE NORMALIZED CROSSCORRELATION FUNCTION
C
        PX = SNGL(XSS/DFLOAT(L))
        PY = SNGL(YSS/DFLOAT(L))
        GAIN = SNGL((XYSM*DFLOAT(L))/(XSS*DFLOAT(KL)))
        XYS = XYSM*XYSM
        ERRN = SNGL(1.0D0-XYS/(XSS*YSS))
        XYS = XYS/(XSS*YSS-XYS)
        SNR = 10.0*SNGL(DLOG10(XYS))
        IF (IFLAG.EQ.0) GO TO 45
C
C       PRINT WARNING IF ANY STATIONARITY TESTS WERE FAILED
C
        WRITE (6,208) IFLAG
   45   WRITE (6,202) ITAU,PX,PY,RTMAX,SNR,ERRN,GAIN
```

```
      IF (ISAVER.EQ.0) GO TO 50
      IF (ISAVER.NE.2.AND.ISAVER.NE.3) GO TO 46
C
C     GRAPH CROSSCORRELATION FUNCTION IF SPECIFIED.
C
      J = ITAU + 1
      I3 = 5
      I1 = MOD(J,I3)
      ICHK = J-200*I3
      IF (ICHK.GT.I1) I1 = ICHK
      I2 = I1+2000
      ICHK = NBLOCK*LL
      IF (I2.GT.ICHK) I2 = ICHK
      CALL GRAPH(RXY,K,I1,I2,I3,101,0.,0,1.,0.)
   46 CONTINUE
      IF (ISAVER.NE.1.AND.ISAVER.NE.3) GO TO 50
      LSTOP = NBLOCK*LL
      J = 0
C
C     PRINT ENTIRE CROSSCORRELATION FUNCTION IF SPECIFIED
C
      WRITE (6,203) J,(RXY(I),I,I=1,LSTOP)
      WRITE (6,204)
   50 RETURN
  200 FORMAT(///,' REFERENCE MEAN =',E19.7,/,' REFERENCE VARIANCE =',
     1E15.7,///)
  201 FORMAT(' MEASUREMENT DATA, BLOCK NUMBER',I4,' -     MEAN =',
     1E19.7,/,T43,'VARIANCE =',E15.7)
  202 FORMAT(//,'0TAU   =',I17,/,'0PX    =',E17.7,/,'0PY    =',E17.7,/,
     1'0RTMAX =',E17.7,/,'0SNR   =',F17.7,/,'0ERRN  =',E17.7,/,
     2'0GAIN  =',E17.7)
  203 FORMAT(1H1,5(7X,'TAU',5X,'RXY(TAU)',2X),///,
     1(1X,I10,E15.7,I10,E15.7,I10,E15.7,I10,E15.7,I10,E15.7))
  204 FORMAT('0IGNORE THE LAST OUTPUT VALUE OF TAU.  NO VALID RXY(TAU) H
```

```
   1AS BEEN COMPUTED FOR IT.')
205 FORMAT(///)
206 FORMAT(1H ,'MEAN OF ENTIRE X ARRAY =',E19.7,/,' VARIANCE OF ENTIRE
   1 X ARRAY =',E15.7,/,' MEAN OF ENTIRE Y ARRAY =',E19.7,/,1X,
   2'VARIANCE OF ENTIRE Y ARRAY =',E15.7)
207 FORMAT(' AT TAU =',I8,'  NORMALIZED RXY(TAU) =',E15.7)
208 FORMAT(///,   '  **>>> WARNING - DATA DOES NOT APPEAR STATIONARY. <
   1<<**',/,' REFERENCE LENGTH MAY BE TOO SHORT FOR RESULTS TO BE VALI
   2D.',/,T20,'IFLAG =',I8)
209 FORMAT('1RESULTS FROM SNRMSE',/,'*_______ ____ ______',//,
   1' * PARAMETERS IN EFFECT *',//,' K      =',T16,I9,/,
   2' BLOCK LENGTH =',I9,/,' IWIN   =',T16,I9,/,' NEWTAU =',T16,I9,/,
   3' ISIGNR =',T16,I9,/,' ILROAM =',T16,I9,/,' ISAVER =',T16,I9,
   4///,7X,'* ANALYSIS *',//)
210 FORMAT(///,' NUMBER OF SAMPLES USED TO COMPUTE FINAL RESULTS =',
   1I8)
    END
```

```
      SUBROUTINE CORR(Y,KY,XT,NT,L,YT,II,JTAU,INV,S,ND,M,RXY,ISIGNR,
     1ISAVER,YM,YVAR)
C
CC    ROUTINE TO PERFORM CORRELATIONS USING FFT FOR USE IN SNRMSE
C
      DIMENSION M(3),Y(KY),XT(NT),YT(NT),INV(ND),S(ND),RXY(KY)
      IF (II.NE.1) GO TO 10
      N = 2*L
      RTMAX = 0.
      JTAU = 1
C
CC    TEST STATIONARITY OF EACH DATA BLOCK FROM Y ARRAY
C
   10 KK = (II-1)*L
      K = KK+1
      YM = 0.
      Y2 = 0.
      DO 15 I=1,N
      K = K+1
      Y2 = Y2+Y(K)**2
   15 YM = YM+Y(K)
      YM = YM/FLOAT(N)
      YVAR = Y2/FLOAT(N) - YM**2
      K = KK+1
C
CC    FFT EACH DATA BLOCK FROM Y ARRAY
C
      DO 20 I=1,N
      K = K+1
      J = 2*I
      YT(J) = 0.
   20 YT(J-1) = Y(K)
      CALL HARM(YT,M,INV,S,-2,IF)
C
```

```
C       FORM CROSS SPECTRAL DENSITY OF REFERENCE BLOCK WITH DATA
C       BLOCKS FROM Y ARRAY
C
        DO 30 I=1,N
        J = 2*I
        K = J-1
        TR = YT(K)
        YT(K) = XT(K)*YT(K)+XT(J)*YT(J)
   30   YT(J) = XT(K)*YT(J)-XT(J)*TR
C
C       COMPUTE INVERSE FFT OF CROSS SPECTRAL DENSITY TO OBTAIN A SEGMENT
C       OF THE CROSSCORRELATION FUNCTION BETWEEN X AND Y ARRAYS
C
        CALL HARM(YT,M,INV,S,2,IF)
C
C       FIND THE PEAK OF THE CROSS CORRELATION FUNCTION.
C       BRANCH DEPENDING UPON WHETHER A POSITIVE PEAK, A NEGATIVE PEAK,
C       OR AN ABSOLUTE VALUE PEAK IS SPECIFIED.
C
        DO 40 I=1,L
        J = 2*I-1
C
C       SAVE CROSSCORRELATION FUNCTION TO BE PRINTED LATER IF SPECIFIED
C
        IF (ISAVER.EQ.0) GO TO 32
        RXY(I+KK) = YT(J)
   32   CONTINUE
        IF (ISIGNR) 34,36,38
   34   IF(YT(J).GE.RTMAX) GO TO 40
        RTMAX = YT(J)
        JTAU = I+KK
        GO TO 40
   36   IF (ABS(YT(J)).LE.RTMAX) GO TO 40
        RTMAX = ABS(YT(J))
```

```
      JTAU = 1+KK
      GO TO 40
   38 IF(YT(J).LE.RTMAX) GO TO 40
      RTMAX = YT(J)
      JTAU = 1+KK
   40 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GRAPH(X,N,INDXLO,INDXHI,INCR,IAXIS,AXIS,ISCALE,XMAX,
     1XMIN)
      DIMENSION VECTOR(101),ABCISA(107),X(N)
      DATA DASH,BAR,STAR,BLANK,PLUS/1H-,1H|,1H*,1H ,1H+/
      DATA A,B,C,EYE,S/1HA,1HB,1HC,1HI,1HS/
      IF (ISCALE.EQ.1) GO TO 20
      XMX = -1.0E70
      XMN = 1.0E70
      IF (IAXIS.GT.100) XMX = AXIS
      IF (IAXIS.GT.100) XMN = AXIS
      DO 10 I=INDXLO,INDXHI,INCR
      IF (X(I).GT.XMX) XMX = X(I)
   10 IF (X(I).LT.XMN) XMN = X(I)
      GO TO 30
   20 XMX = XMAX
      XMN = XMIN
   30 RANGE = XMX-XMN
      WRITE(6,200) XMN,XMX,RANGE
      DO 40 I=2,100
   40 VECTOR(I) = DASH
      DO 50 I=1,101,10
   50 VECTOR(I) = PLUS
      Y = (X(INDXLO)-XMN)/RANGE
      IPOINT = INT(Y*100.+1.5)
      IF (IPOINT.GE.1.AND.IPOINT.LE.101) VECTOR(IPOINT) = STAR
   55 WRITE (6,201) VECTOR
      ILO = INDXLO + INCR
      IF(ILO.GE.INDXHI) GO TO 65
      IF(IAXIS.GE.0) GO TO 70
      DO 60 I=ILO,INDXHI,INCR
      CALL DATAPT(VECTOR,X,N,I,XMN,RANGE,IPOINT)
      IF (IPOINT.GE.1.AND.IPOINT.LE.101) VECTOR(IPOINT) = STAR
   60 WRITE (6,202) VECTOR
   65 RETURN
```

```
   70 IAXPT = IAXIS+1
      IF (IAXIS.LE.100) GO TO 80
      Y = (AXIS-XMIN)/RANGE
      IAXPT = INT(Y*100.+1.5)
   80 J = 0
      DO 90 I=ILO,INDXHI,INCR
      J = J+1
      CALL DATAPT(VECTOR,X,N,I,XMN,RANGE,IPOINT)
      IF(IAXPT.GE.1.AND.IAXPT.LE.101)VECTOR(IAXPT) = BAR
      IF (IPOINT.GE.1.AND.IPOINT.LE.101) VECTOR(IPOINT) = STAR
      K = MOD(J,10)
      IF (K.EQ.0) GO TO 85
      WRITE (6,202) VECTOR
      GO TO 90
   85 VECTOR(IAXPT) = PLUS
      WRITE(6,203) VECTOR,J
   90 CONTINUE
      IF(IAXPT.LT.1.OR.IAXPT.GT.101) RETURN
      DO 95 I=1,107
   95 ABCISA(I) = BLANK
      ABCISA(IAXPT) = A
      ABCISA(IAXPT+1) = B
      ABCISA(IAXPT+2) = C
      ABCISA(IAXPT+3) = EYE
      ABCISA(IAXPT+4) = S
      ABCISA(IAXPT+5) = S
      ABCISA(IAXPT+6) = A
      WRITE (6,204) ABCISA
      RETURN
  200 FORMAT(1H1,T10,'**GRAPH SCALING**',//,'0MINIMUM VALUE OF ORDINATE
     1=',E15.7,/,'0MAXIMUM VALUE OF ORDINATE =',E15.7,/,
     2'0DYNAMIC RANGE OF ORDINATE VALUES SHOWN =',E15.7)
  201 FORMAT(1H1,T56,'ORDINATE',//,T10,'0',T29,'2 0',T49,'4 0',T69,'6 0'
     1,T89,'8 0',T109,'100',T114,'NORMALIZED SCALES',//,T10,101A1,9X,'0'
```

```
   2)
202 FORMAT(T10,101A1)
203 FORMAT(T10,101A1,I10)
204 FORMAT(//,T7,107A1)
    END

    SUBROUTINE DATAPT(VECTOR,X,N,I,XMN,RANGE,IPOINT)
    DIMENSION VECTOR(101),X(N)
    DATA BLANK/1H /
    DO 10 J=1,101
 10 VECTOR(J) = BLANK
    Y = (X(I)-XMN)/RANGE
    IPOINT = INT(Y*100.+1.5)
    RETURN
    END
```

APPENDIX B

COMPARISON OF DATA WINDOWS FOR USE IN SNRBPS

Initial testing of the Hamming data window for use in SNRBPS showed that a significant improvement in the accuracy of results could be realized compared to processing without applying the window function. These results prompted further testing of a variety of data windows.

The tests were made using a double sideband signal, specifically a product of sinusoids. Amplitude scaling and delays were programmed into the measurement signal, and additive Gaussian noise was applied. Tests were made for SNR values of 17.7 and 36.9 dB and for a very large value, which was too great to be measured accurately by the routine. Results are tabulated in Table V. Seven windows, in addition to rectangular, were tested.

The results show all windows used work well for the lowest value of the SNR. At 36.9 dB, many window functions yield very poor results. Two Kaiser windows, the Hanning window, and the triangular window all remain within one dB of the correct SNR value. For the extremely high SNR test (on the order of 70 dB), results are not tabulated, since no accurate measurements were obtained. However, the Hanning window provided an estimate in excess of 60 dB, which was more than 5 dB greater than any other window. On the basis of these results, the Hanning window was tentatively chosen for

TABLE V

RESULTS OF TESTS OF WINDOW FUNCTIONS

Window Function	SNR Estimates	
	True SNR = 17.7 dB	True SNR = 36.9 dB
Rectangular	16.4	22.1
Blackman	17.4	30.0
Kaiser, $\theta = \pi$	17.5	30.8
Kaiser, $\theta = 3\pi/2$	17.7	36.3
Kaiser, $\theta = 2\pi$	17.9	37.6
Hamming	17.7	34.8
Hanning	17.7	37.6
Triangular	17.8	37.4

use in SNRBPS. Subsequent testing verified the overall accuracy and reliability of estimates formed using this window on many different signals. Therefore, the Hanning window was permanently installed in subroutine SNRBPS for the data window processing option.

APPENDIX C

SUBROUTINE SNRBPS

Subroutine SNRBPS is a FORTRAN IV implementation of the second SNR estimator developed for use in computer simulations of communication systems. It is applicable to systems where the group and phase delays of the measurement data with respect to the reference data are arbitrary. These arbitrary delays are associated with bandpass systems. The routine will yield valid results only with signals which carry all information in the envelope of the waveform. In such cases only the envelope of the signal is required to be distortionless to retain all the information; delay of the carrier itself is unimportant. The routine should be applied to systems having some carrier frequency displaced from zero. For proper estimation of group delay, the data should be wide-sense stationary.

Input to the routine is essentially the same as for SNRMSE. The delay included in the argument list refers to group delay. There is an additional input parameter for SNRBPS which can be used to limit the maximum delay value to be considered in estimating group delay for the system. All the arguments for the subroutine are listed with explanations in the comments at the beginning of the program listing which is given.

Like SNRMSE, the execution of SNRBPS does not destroy any input data, so successive processing of the data sets

is possible. Outputs provided by SNRBPS are identical to those of SNRMSE with three exceptions. The delay value given denotes group delay. (No phase delay estimate is produced.) The peak of the envelope of the cross-correlation function is provided, instead of the peak of the cross-correlation function itself. Finally, magnitudes of normalized cross-spectral power estimates are printed instead of normalized cross-correlation values, since these power measurements are used in estimating group delay.

Stationarity tests are applied and warnings generated exactly as in the lowpass routine. The subroutine CORR2 is listed following SNRBPS. It is called by SNRBPS to compute the cross-correlation function, and it requires no attention from the user. In addition to CORR2, subroutine SNRBPS requires GRAPH and DATAPT, which have already been listed with the lowpass routine, and the FFT routine HARM. Actually, subroutines GRAPH and DATAPT are only used in either routine when a plot of the cross-correlation function is requested. Many FORTRAN compilers, however, require that the routines (or a dummy routine named GRAPH) be available in order to properly compile SNRBPS, which contains a calling statement for GRAPH. Similarly, CORR and CORR2 are only executed in their respective routines if a delay estimate is to be found by the routines in forming the SNR estimate.

The input parameter ISIGNR indicates whether a positive, negative, or an absolute value peak is to be found for the

cross-correlation function to select the general region where a search is performed to determine group delay. The discussion given in Appendix A for SNRMSE explains the effect of ISIGNR on the search of the cross-correlation function. However, system gain is always considered to be positive in the bandpass case. The phase delay may be changed by 180^{o} to provide sign changes. Therefore, in normal use, the user should always specify ISIGNR = 0, indicating the absolute value peak is to be found.

In some cases, signals may be analyzed where it is known that theoretically equal magnitude positive and negative cross-correlation peaks exist and that either peak would lead to a theoretically correct group delay estimate. A simple example of this is the sinusoidal modulation of a sinusoid used as a test signal for SNRBPS in the section on application to analog systems. It is preferable to use the smallest theoretically correct delay value available, since this procedure yields estimates based on the largest number of samples. If the user knows a positive cross-correlation peak will occur for the smallest theoretically correct delay, it is helpful to specify ISIGNR = 1 to increase the likelihood of selecting this delay. Similarly, ISIGNR = -1 can be used if a negative peak is known to occur for the smallest theoretically correct delay. This procedure is occasionally useful, but it is entirely optional. The user may specify ISIGNR = 0 in all cases and obtian valid results. The option simply allows the user to obtain more reliable estimates in a few special cases.

```
      SUBROUTINE SNRBPS(X,M1,Y,K,XT,NT,YT,IWIN,NEWTAU,ITAU,PX,PY,RTMAX,
     1SNR,ERRN,INV,S,ND,RXY,ISIGNR,ISAVER,ILROAM,MAXTAU)
C*****************************************************************C
C                                                                 C
C                        SUBROUTINE SNRBPS                        C
C                                                                 C
C                                                                 C
C     ROUTINE TO MEASURE SIGNAL-TO-'NOISE' RATIO, NORMALIZED      C
C     MEAN SQUARE ERROR, DELAY, AND POWER FOR A REFERENCE ARRAY X C
C     AND AN ARRAY OF MEASUREMENT DATA Y                          C
C      SNRBPS IS SUITABLE FOR BANDPASS SYSTEMS ONLY.              C
C                                                                 C
C     SNRBPS REQUIRES SUBROUTINE CORR2 IF NEWTAU = 1 IS SPECIFIED C
C                                                                 C
C                                                                 C
C                    *****INPUT PARAMETER$*****                  C
C                                                                 C
C                                                                 C
C     X - THE ARRAY OF REFERENCE DATA                             C
C     M1 - DETERMINES SIZE OF REFERENCE BLOCK TO BE TRANSFORMED IN C
C          FINDING DELAY.  N=2**M1 EQUALS NUMBER OF POINTS USED   C
C          FOR FFT                                                C
C     Y - THE ARRAY OF MEASUREMENT DATA                           C
C     K - THE NUMBER OF DATA POINTS IN X AND Y ARRAYS.  DIMENSION OF C
C         X,Y,AND RXY                                             C
C     XT - WORK ARRAY FOR REFERENCE TRANSFORMS                    C
C     NT - DIMENSION OF XT AND YT.  NT=2*2**M1                    C
C     YT - WORK ARRAY FOR MEASUREMENT BLOCK TRANSFORMS            C
C     IWIN - SPECIFIES WHETHER TO WINDOW DATA IN COMPUTING POWERS C
C               0 - DO NOT WINDOW DATA                            C
C               1 - WINDOW DATA                                   C
C     NEWTAU - SPECIFIES WHETHER TO COMPUTE A NEW DELAY VALUE OR  C
C              TO USE THE INPUT VALUE OF ITAU FOR INITIAL DELAY   C
C              ESTIMATE                                           C
C                0 - DO NOT COMPUTE A NEW DELAY VALUE.  USE THE   C
C                    INPUT VALUE ASSIGNED TO ITAU                 C
C                                                                 C
```

```
C                1 - COMPUTE A NEW DELAY VALUE
C      INV - WORK ARRAY FOR HARM FFT. DIMENSION IS NT/8
C      S - WORK ARRAY FOR HARM FFT. DIMENSION IS NT/8
C      ND - DIMENSION OF INV AND S.  ND = NT/8
C      RXY - ARRAY TO STORE CROSSCORRELATION FUNCTION.  DIMENSION IS K
C      ISIGNR - DETERMINES WHETHER TO USE POSITIVE PEAK, NEGATIVE PEAK,
C               OR ABSOLUTE VALUE PEAK OF CROSSCORRELATION FUNCTION IN
C               FINDING INITIAL DELAY ESTIMATE.
C                 -1 - USE NEGATIVE PEAK
C                  0 - USE ABSOLUTE VALUE PEAK
C                  1 - USE POSITIVE PEAK
C      ISAVER - DETERMINES WHETHER OR NOT TO PRINT AND/OR PLOT THE
C               CROSSCORRELATION FUNCTION
C                  0 - DO NOT PRINT OR PLOT RXY
C                  1 - PRINT RXY
C                  2 - PLOT RXY ON THE PRINTER
C                  3 - PRINT AND PLOT RXY
C      ILROAM - DETERMINES HOW MANY POINTS TO SEARCH FOR THE PEAK
C               CROSSCORRELATION.  ANY ODD INTEGER GREATER THAN OR
C               EQUAL TO 1 IS VALID, BUT LARGE NUMBERS (SAY,
C               GREATER THAN ABOUT 7 OR 9) WILL INCREASE RUN TIME
C               SIGNIFICANTLY
C      ITAU - IF NEWTAU IS SPECIFIED 0, THE INPUT VALUE OF ITAU
C             WILL BE USED FOR INITIAL DELAY ESTIMATE.
C             (SEE ITAU BELOW, ALSO)
C      MAXTAU - SPECIFIES MAXIMUM DELAY VALUE TO BE CONSIDERED AS AN
C               INITIAL DELAY ESTIMATE.  A NEGATIVE INPUT VALUE IS
C               USED TO IMPLY NO RESTRICTION.
C
C                    *****OUTPUT VALUES*****
C
C      ITAU - NUMBER OF STEPSIZE DELAYS TO PEAK OF NORMALIZED
C             CROSSCORRELATION FUNCTION IF NEWTAU IS 1.  IF
C             NEWTAU IS 0, OUTPUT ITAU APPROXIMATELY EQUALS INPUT ITAU.
```

```
C             (ILROAM GREATER THAN 0 MAY LEAD TO A NEW DELAY ESTIMATE.)
C      PX - POWER IN X ARRAY
C      PY - POWER IN Y ARRAY
C      RTMAX - PEAK VALUE OF MAGNITUDE OF CROSS-SPECTRAL POWER
C      SNR - SIGNAL TO NOISE RATIO
C      ERRN - NORMALIZED MEAN SQUARE ERROR
C
C*****************************************************************************
      COMPLEX*16 XC,YC,XYC,XCYC
      REAL*8 XSST,YSST,XYST,XSS,YSS,XYS,XYSN,XYM
      DIMENSION M(3),X(K),Y(K),XT(NT),YT(NT),INV(ND),S(ND),RXY(K)
      IFLAG = 0
      IROAM = ILROAM
      W = 1.0
C
C     INITIALIZE PARAMETERS
C
      M(1) = M1
      M(2) = 0
      M(3) = 0
      CALL HARM(XT,M,INV,S,0,IF)
      N = 2**M1
      LL = N/2
C
C     BRANCH IF THE PREVIOUSLY COMPUTED VALUE OF ITAU IS TO BE USED
C     AS THE CURRENT DELAY VALUE
C
      IF (NEWTAU.NE.0) GO TO 2
      JTAU = ITAU+1
      GO TO 40
    2 XM = 0.
      X2 = 0.
      WRITE (6,209) K,N,IWIN,NEWTAU,ISIGNR,ILROAM,ISAVER,MAXTAU
C
```

```
C       COMPUTE MEAN AND VARIANCE OF REFERENCE BLOCK
C
        DO 5 I=1,LL
        X2 = X2+X(I)**2
      5 XM = XM+X(I)
        XM = XM/FLOAT(LL)
        XVAR = (X2/FLOAT(LL))-XM**2
C
C       COMPUTE MEAN AND VARIANCE OF ENTIRE X AND Y ARRAYS
C
        XMN = 0.
        YMN = 0.
        X2 = 0.
        Y2 = 0.
        DO 8 I=1,K
        XMN = XMN + X(I)
        YMN = YMN + Y(I)
        X2 = X2+X(I)**2
      8 Y2 = Y2 + Y(I)**2
        XMN = XMN/FLOAT(K)
        YMN = YMN/FLOAT(K)
        XVAR2 = X2/FLOAT(K)-XMN**2
        YVAR2 = Y2/FLOAT(K)-YMN**2
C
C       SET UP STATIONARITY TEST PARAMETERS
C
        XVTEST = XVAR2/2.
        XMTEST = SQRT(XVAR2)/2.
        YVTEST = YVAR2/2.
        YMTEST = SQRT(YVAR2)/2.
C
C       TEST STATIONARITY OF REFERENCE BLOCK
C
        IF (ABS(XMN-XM).GT.XMTEST) IFLAG = 10000
```

```
      IF (ABS(XVAR2-XVAR).GT.XVTEST) IFLAG = -10000
      WRITE (6,206) XMN,XVAR2,YMN,YVAR2
      WRITE (6,200) XM,XVAR
C
C     PAD WITH ZEROES AND FFT THE REFERENCE BLOCK
C
      DO 10 I=1,LL
      XT(3*I) = 0.
      XT(4*I) = 0.
      J = 2*I
      XT(J) = 0.
   10 XT(J-1) = X(I)
      CALL HARM(XT,M,INV,S,-2,IF)
C
C     COMPUTE CROSSCORRELATION FUNCTION AND LOCATE ITS PEAK
C
      NBLOCK = K/LL-1
C
C     BRANCH AND PRINT MESSAGE IF FFT BLOCK LENGTH IS TOO LONG FOR
C     DATA SET
C
      IF(NBLOCK.LE.1) GO TO 72
      IMTHRU = 0
      DO 20 I=1,NBLOCK
      CALL CORR2(Y,K,XT,NT,LL,YT,I,JTAU,INV,S,ND,M,RXY,ISIGNR,ISAVER,YM,
     1YVAR,MAXTAU,IMTHRU)
C
C     TEST STATIONARITY OF DATA BLOCK FROM Y ARRAY
C
      IF(ABS(YMN-YM).GT.YMTEST) IFLAG = 1
      IF (ABS(YVAR2-YVAR).GT.YVTEST) IFLAG = -1
      WRITE (6,201) I,YM,YVAR
C
C     BRANCH OUT OF LOOP IF SEARCH HAS EXCEEDED MAXTAU
```

```
C
      IF(IMTHRU.NE.0) GO TO 22
   20 CONTINUE
C
C     COMPUTE RANGE TO BE SEARCHED FOR THE PEAK OF THE ENVELOPE OF
C     THE CROSSCORRELATION FUNCTION.  ONE FFT BLOCK MINUS 80 SAMPLES
C     IS USED; LAST 40 SAMPLES AT EACH END ARE ARTIFICIALLY CHOSEN TO
C     AVOID UNDESIREABLE END EFFECTS.
C
   22 INDEX1 = JTAU-LL+20
      INDEX1 = MAX0(INDEX1,1)
      INDEX2 = INDEX1+N-41
      IND = NBLOCK*LL
      IF(IND.LT.INDEX2) INDEX1 = IND-N+41
      INDEX2 = MIN0(INDEX2,IND)
      J = 41
C
C     LOAD PROPER PORTION OF RXY INTO XT
C
      DO 30 I=INDEX1,INDEX2
      JJ = J+1
      XT(J) = RXY(I)
      XT(JJ) = 0.0
   30 J = J+2
      NFOLD = NT-80
      DO 32 I=1,39
      II = 80-I
      XT(I) = XT(II)
      J = NFOLD+I
      JJ = NT-I
   32 XT(JJ) = XT(J)
      XT(40) = 0.0
      XT(NT) = 0.0
C
```

```
C     COMPUTE HILBERT TRANSFORM FOR CROSSCORRELATION FUNCTION ON THE
C     INTERVAL BEING SEARCHED.  HILBERT TRANSFORM IS IMPLEMENTED USING
C     THE FFT.
C
      CALL HARM(XT,M,INV,S,-2,IF)
      DO 34 I=2,N,2
      TI = XT(I)
      II = I-1
      XT(I) = -XT(II)
      XT(II) = TI
      J = I+N
      JJ = J-1
      TI = XT(J)
      XT(J) = XT(JJ)
   34 XT(JJ) = -TI
C
C     FIND INITIAL ESTIMATE FOR GROUP DELAY (PEAK OF CROSSCORRELATION
C     FUNCTION ENVELOPE)
C
      CALL HARM(XT,M,INV,S,2,IF)
      J = 41
      ENVMAX = 0.0
      DO 36 I=INDEX1,INDEX2
C
C     ENVELOPE(FUNCTION) = SQRT(FUNCTION**2+HILBERT(FUNCTION)**2)
C
      ENV = SQRT(RXY(I)**2+XT(J)**2)
      IF (ENV.LE.ENVMAX) GO TO 36
      ENVMAX = ENV
      JTAU = I
   36 J = J+2
C
C     SET BOUNDARIES FOR FINAL SEARCH FOR GROUP DELAY
C
```

```
   40 IBEGIN = JTAU-(IROAM-1)/2
      IF (IBEGIN.LT.1) IBEGIN = 1
      IQUIT = IBEGIN+IROAM-1
      IQUT = IQUIT+N
C
C     BRANCH AND PRINT MESSAGE IF DELAY VALUE IS SO LARGE THAT THERE
C     IS INSUFFICIENT DATA TO FFT FOR FINAL SEARCH RANGE
C
      IF(IQUT.GT.K) GO TO 70
      WRITE (6,205)
      KPTS = K-JTAU+1
      NBLK = KPTS/N
C
C     IF MORE DATA THAN ONE FOURTH OF AN FFT BLOCK IS LEFT OVER AFTER
C     COMPUTING NUMBER OF FULL SIZE BLOCKS TO BE ANALYZED, ADD AN
C     OVERLAPPING LAST BLOCK TO USE ALL THE DATA.
C
      LASTPT = N*NBLK
      LEFT = KPTS-LASTPT
      LL2 = LL/2
      IF(LEFT.LE.LL2) GO TO 44
      NBLK = NBLK+1
      LASTPT = KPTS
C
C     INITIALIZE FOR FINAL SEARCH
C
   44 XYS = 0.0
      XSS = 0.0
      YSS = 0.0
      XYM = 0.0
C
C     ITERATE THROUGH RANGE SELECTED FOR GROUP DELAY SEARCH
C
      DO 60 KTAU = IBEGIN,IQUIT
```

```
      XSST = 0.0D0
      YSST = 0.0D0
      XYC = DCMPLX(0.D0,0.D0)
C
C     PERFORM COMPUTATIONS AT EACH DELAY
C
      DO 58 LOOP=1,NBLK
C
C     COMPUTE INDICES OF ARRAYS FOR EACH FFT BLOCK
C
      ILOY = KTAU+(LOOP-1)*N
      IHIY = ILOY+N-1
      ILOX = ILOY-KTAU+1
      IF(IHIY.LE.K) GO TO 52
      IHIY = K
      ILOY = IHIY-N+1
      ILOX = ILOY-KTAU+1
   52 IX = ILOX
      J = 1
      DO 54 IY = ILOY,IHIY
      JJ = J+1
C
C     WINDOW IF SPECIFIED
C
      IF(IWIN.EQ.0) GO TO 53
      JW = JJ/2
      ARGC = 6.2831853*FLOAT(JW-1)/FLOAT(N)
      W = 0.8164966*(1.-COS(ARGC))
   53 XT(J) = X(IX)*W
      YT(J) = Y(IY)*W
      XT(JJ) = 0.0
      YT(JJ) = 0.0
      IX = IX+1
   54 J=J+2
```

```
C
C      PERFORM FFT
C
       CALL HARM(XT,M,INV,S,-2,IF)
       CALL HARM(YT,M,INV,S,-2,IF)
C
C      COMPUTE POWER IN REFERENCE, POWER IN MEASUREMENT DATA, AND
C      MAGNITUDE OF CROSS-SPECTRAL POWER AT EACH DELAY VALUE USING
C      DOUBLE PRECISION ARITHMETIC, EXCEPT IN THE FFT
C
       DO 57 J=2,NT,2
       HALF = FLOAT(N-J)+0.5
       I=J-1
       XC = DCMPLX(DBLE(XT(I)),DBLE(XT(J)))
       YC = DCMPLX(DBLE(YT(I)),DBLE(YT(J)))
       XSST = XSST + XC*DCONJG(XC)
       YSST = YSST + YC*DCONJG(YC)
       IF (HALF.LT.0.) GO TO 56
       XCYC = XC*DCONJG(YC)
       GO TO 57
   56  XCYC = DCONJG(XC)*YC
   57  XYC = XYC + XCYC
   58  CONTINUE
       XYST = XYC*DCONJG(XYC)
       XYSN = DSQRT(XYST/(XSST*YSST))
       LTAU = KTAU-1
C
C      LOCATE PEAK MAGNITUDE OF NORMALIZED CROSS-SPECTRAL POWER
C
       IF(XYSN.LE.XYM.AND.ISIGNR.GT.0) GO TO 59
       IF(XYSN.GE.XYM.AND.ISIGNR.LT.0) GO TO 59
       IF(DABS(XYSN).LE.DABS(XYM).AND.ISIGNR.EQ.0) GO TO 59
       XYM = XYSN
       XYS = XYST
```

```
      XSS = XSST
      YSS = YSST
      ITAU = LTAU
   59 WRITE (6,207) LTAU,XYSN
   60 CONTINUE
      WRITE (6,210) LASTPT
C
C     COMPUTE AND PRINT NUMBER OF DELAYS, POWER IN X, POWER IN Y,
C     CROSSCORRELATION FUNCTION VALUE, SIGNAL TO NOISE RATIO, GAIN,
C     AND THE NORMALIZED MEAN SQUARE ERROR, COMPUTED USING THE DELAY
C     INDICATED BY THE PEAK MAGNITUDE OF THE NORMALIZED CROSS-SPECTRAL
C     POWER
C
      RTMAX = SNGL(DSQRT(XYS)/DFLOAT(NBLK))
      PX = SNGL(XSS/DFLOAT(NBLK))
      PY = SNGL(YSS/DFLOAT(NBLK))
      GAIN = SNGL(DSQRT(XYS)/XSS)
      ERRN = SNGL(1.0D0-XYS/(XSS*YSS))
      XYS = XYS/(XSS*YSS-XYS)
      SNR = 10.0*SNGL(DLOG10(XYS))
      IF (IFLAG.EQ.0) GO TO 45
C
C     PRINT WARNING IF ANY STATIONARITY TESTS WERE FAILED
C
      WRITE (6,208) IFLAG
   45 WRITE (6,202) ITAU,PX,PY,RTMAX,SNR,ERRN,GAIN
      IF (ISAVER.EQ.0) GO TO 50
      IF (ISAVER.NE.2.AND.ISAVER.NE.3) GO TO 46
C
C     GRAPH CROSSCORRELATION FUNCTION IF SPECIFIED.
C
      J = ITAU + 1
      I3 = 5
      I1 = MOD(J,I3)
```

```
      ICHK = J-200*13
      IF (ICHK.GT.11) 11 = ICHK
      12 = 11+2000
      ICHK = NBLOCK*LL
      IF (12.GT.ICHK) 12 = ICHK
      CALL GRAPH(RXY,K,11,12,13,101,0.,0,1.,0.)
   46 CONTINUE
      IF (ISAVER.NE.1.AND.ISAVER.NE.3) GO TO 50
      LSTOP = NBLOCK*LL
      J = 0
C
C     PRINT ENTIRE CROSSCORRELATION FUNCTION IF SPECIFIED
C
      WRITE (6,203) J,(RXY(I),I,I=1,LSTOP)
      WRITE (6,204)
   50 RETURN
C
C     ERROR MESSAGES AND RETURNS
C
   70 WRITE(6,211) JTAU,IROAM,N,K
      GO TO 50
   72 WRITE(6,212) K,N
      GO TO 50
  200 FORMAT(///,' REFERENCE MEAN =',E19.7,/,' REFERENCE VARIANCE =',
     1E15.7,///)
  201 FORMAT(' MEASUREMENT DATA, BLOCK NUMBER',I4,' -     MEAN =',
     1E19.7,/,T43,'VARIANCE =',E15.7)
  202 FORMAT(//,'0TAU    =',I17,/,'0PX     =',E17.7,/,'0PY     =',E17.7,/,
     1'0RTMAX =',E17.7,/,'0SNR    =',F17.7,/,'0ERRN   =',E17.7,/,
     2'0GAIN  =',E17.7)
  203 FORMAT(1H1,5(7X,'TAU',5X,'RXY(TAU)',2X),///,
     1(1X,I10,E15.7,I10,E15.7,I10,E15.7,I10,E15.7,I10,E15.7))
  204 FORMAT('0IGNORE THE LAST OUTPUT VALUE OF TAU.  NO VALID RXY(TAU) H
     1AS BEEN COMPUTED FOR IT.')
```

```
205 FORMAT(///)
206 FORMAT(1H ,'MEAN OF ENTIRE X ARRAY =',E19.7,/,' VARIANCE OF ENTIRE
   1 X ARRAY =',E15.7,/,' MEAN OF ENTIRE Y ARRAY =',E19.7,/,1X,
   2'VARIANCE OF ENTIRE Y ARRAY =',E15.7)
207 FORMAT(' AT TAU =',I8,'  NORMALIZED CROSS-SPECTRAL PWR =',D17.9)
208 FORMAT(///,  '  **>>> WARNING - DATA DOES NOT APPEAR STATIONARY. <
   1<<**',/,' REFERENCE LENGTH MAY BE TOO SHORT FOR RESULTS TO BE VALI
   2D.',/,T20,'IFLAG =',I8)
209 FORMAT('1RESULTS FROM SNRBPS',/,'+_____ ____ _____',//,
   1' * PARAMETERS IN EFFECT *',//,' K      =',T16,I9,/,
   2' BLOCK LENGTH =',I9,/,' IWIN   =',T16,I9,/,' NEWTAU =',T16,I9,/,
   3' ISIGNR =',T16,I9,/,' ILROAM =',T16,I9,/,' ISAVER =',T16,I9,
   4/,' MAXTAU =',T16,I9,///,7X,'* ANALYSIS *',//)
210 FORMAT(///,' NUMBER OF SAMPLES USED TO COMPUTE FINAL RESULTS =',
   1I8)
211 FORMAT(1H1,'***** THE ROUGH ESTIMATE FOR DELAY, JTAU =',I7,
   1', AND THE SPECIFIED VALUE OF ILROAM =',I7,/,'0PREVENT THE USE OF
   2EVEN ONE DATA BLOCK OF LENGTH N=',I7,' FROM BEING ANALYZED OVER TH
   3E SPECIFIED RANGE OF DELAYS.',/,'0TO CONTINUE ANALYSIS, RERUN REDU
   4CING THE VALUE OF N OR OF ILROAM SO THAT (JTAU+N+ILROAM/2) IS LESS
   5 THAN K=',I7,' . *****')
212 FORMAT('0ERROR:  FFT BLOCK LENGTH MUST BE CHOSEN NOT MORE THAN ONE
   1 HALF',/,' THE LENGTH OF THE TOTAL DATA SET IN X OR Y IF DELAY IS
   2TO BE',/,' FOUND.  FOR THIS RUN K =',I10,' AND FFT LENGTH =',I10)
    END
```

```
      SUBROUTINE CORR2(Y,KY,XT,NT,L,YT,II,JTAU,INV,S,ND,M,RXY,ISIGNR,
     1ISAVER,YM,YVAR,MAXTAU,IMTHRU)
C
C     ROUTINE TO PERFORM CORRELATIONS USING FFT FOR USE IN SNRBPS
C
      DIMENSION M(3),Y(KY),XT(NT),YT(NT),INV(ND),S(ND),RXY(KY)
      IF (II.NE.1) GO TO 10
      N = 2*L
      RTMAX = 0.
      JTAU = 1
C
C     TEST STATIONARITY OF EACH DATA BLOCK FROM Y ARRAY
C
   10 KK = (II-1)*L
      K = KK+1
      YM = 0.
      Y2 = 0.
      DO 15 I=1,N
      K = K+1
      Y2 = Y2+Y(K)**2
   15 YM = YM+Y(K)
      YM = YM/FLOAT(N)
      YVAR = Y2/FLOAT(N) - YM**2
      K = KK+1
C
C     FFT EACH DATA BLOCK FROM Y ARRAY
C
      DO 20 I=1,N
      K = K+1
      J = 2*I
      YT(J) = 0.
   20 YT(J-1) = Y(K)
      CALL HARM(YT,M,INV,S,-2,IF)
C
```

```
C       FORM CROSS SPECTRAL DENSITY OF REFERENCE BLOCK WITH DATA
C       BLOCKS FROM Y ARRAY
C
        DO 30 I=1,N
        J = 2*I
        K = J-1
        TR = YT(K)
        YT(K) = XT(K)*YT(K)+XT(J)*YT(J)
   30   YT(J) = XT(K)*YT(J)-XT(J)*TR
C
C       COMPUTE INVERSE FFT OF CROSS SPECTRAL DENSITY TO OBTAIN A SEGMENT
C       OF THE CROSSCORRELATION FUNCTION BETWEEN X AND Y ARRAYS
C
        CALL HARM(YT,M,INV,S,2,IF)
C
C       FIND THE PEAK OF THE CROSS CORRELATION FUNCTION.
C       BRANCH DEPENDING UPON WHETHER A POSITIVE PEAK, A NEGATIVE PEAK,
C       OR AN ABSOLUTE VALUE PEAK IS SPECIFIED.
C
        DO 40 I=1,L
C
C       IF SEARCH IS TO EXCEED MAXTAU, BRANCH OUT OF LOOP AND SET FLAG.
C
        MITHRU = I+KK-1
        IF(MITHRU.GT.MAXTAU.AND.MAXTAU.GE.0) GO TO 50
        J = 2*I-1
        RXY(I+KK) = YT(J)
   32   CONTINUE
        IF (ISIGNR) 34,36,38
   34   IF(YT(J).GE.RTMAX) GO TO 40
        RTMAX = YT(J)
        JTAU = I+KK
        GO TO 40
   36   IF (ABS(YT(J)).LE.RTMAX) GO TO 40
```

```
      RTMAX = ABS(YT(J))
      JTAU = 1+KK
      GO TO 40
   38 IF(YT(J).LE.RTMAX) GO TO 40
      RTMAX = YT(J)
      JTAU = 1+KK
   40 CONTINUE
      RETURN
   50 IMTHRU = 1
      RETURN
      END
```

APPENDIX D

NOISE-EQUIVALENT BANDWIDTHS FOR THE DIGITAL FILTER

For the purpose of computer simulation of communication systems, a general lowpass and bandpass bilinear-Z digital filter was designed with variable order and critical frequencies. For various filter orders and a range of values of the filter critical frequencies, noise-equivalent bandwidths of the filter were measured using a time domain technique. A comparison was made between the noise-equivalent bandwidth of the lowpass digital filter and that of the corresponding lowpass analog filter.

The filter was designed by first determining the transfer function of the Butterworth analog filter with the desired critical frequencies. Then standard bilinear-Z techniques were applied to obtain the corresponding digital filter. The filter was scaled so that H_{MAX}, the maximum magnitude of the filter transfer function, was equal to unity.

The method used to measure the noise-equivalent bandwidth of the filter is based upon Parseval's theorem, which states that [74]

$$\int_{-\infty}^{\infty} |H(f)|^2 df = \int_{-\infty}^{\infty} h^2(t) dt$$

where H(f) is the transfer function of the filter and h(t) is the impulse response. Since the filter is

causal, $h(t) = 0$ for $t < 0$, and

$$\int_{-\infty}^{\infty} |H(f)|^2 df = \int_{0}^{\infty} h^2(t)dt. \qquad (62)$$

By definition of B_N, the noise-equivalent bandwidth, [75]

$$B_N = \frac{1}{H_{MAX}^2} \int_{0}^{\infty} |H(f)|^2 df = \frac{1}{2H_{MAX}^2} \int_{-\infty}^{\infty} |H(f)|^2 \, df.$$

Since H_{MAX} has been scaled to unity,

$$B_N = \frac{1}{2} \int_{-\infty}^{\infty} |H(f)|^2 df. \qquad (63)$$

Substitution of (62) into (63) yields

$$B_N = \frac{1}{2} \int_{0}^{\infty} h^2(t)dt$$

for the noise-equivalent bandwidth.

To measure the noise-equivalent bandwidth, a pulse of unit area was applied to the digital filter. Rectangular integration was used by the computer to obtain

$$B_{ND} \simeq \frac{1}{2} \sum_{n=0}^{K} h^2(nT)T$$

where T denotes the time between samples. The integer K must be large enough to include all significant terms. The value of K required depends upon how rapidly the summation converges.

Noise-equivalent bandwidths were computed for both lowpass and bandpass versions having up to 20 poles. A wide range of critical frequencies were covered, and a rather extensive table was compiled for use with the computer simulations of communication systems. The table is not reproduced here, since it is only accurate for this particular software package applied to the IBM 370.

In order to study the relationship between the noise-equivalent bandwidths for the analog and digital filters, an analysis was performed for several orders of the low-pass filter. The qualitative conclusions are easily extended to include the bandpass filter.

The noise-equivalent bandwidth of the analog lowpass Butterworth filter can easily be shown to be [76]

$$B_{NA} = \frac{\pi f_{3dB}}{2N\sin(\pi/2N)}$$

where f_{3dB} denotes the 3dB frequency of the filter, and N is the filter order.

Using the expressions for B_{ND} and B_{NA}, a comparison was made of the noise-equivalent bandwidths of the low-pass analog filter and the corresponding lowpass digital filter. For filter orders 1 through 9, tests were performed over the range $0.0015 \leq R \leq 0.25$, where R is the 3 dB frequency of the filter normalized by the sampling frequency. The sampling frequency, f_s, was chosen to be unity for convenience. Thus, the normalized 3dB frequency, R, becomes equal to f_{3dB}; and f_{3dB} is directly applicable to both digital and analog filters.

Results are given in Table VI for filter orders 1, 2, 4, and 8. Curves are shown in Figure 34. It can be seen that the noise-equivalent bandwidths of the digital filter are less than those of the corresponding analog filters throughout this range. However, the difference is small for small values of R. The difference becomes significant for large values of R and is clearly more pronounced for filters of low order.

This behavior can be explained by remembering that the frequency response of the digital filter approaches zero as the frequency approaches $f_s/2$. For frequencies which are very small with respect fo f_s, the response curves of the analog and digital filters are nearly identical. For frequencies near one half the sampling frequency, however, the digital filter frequency response begins to diverge downward from the analog response. If f_{3dB} is chosen so that R is small, the frequency responses of both filters are very low over the range where this divergence occurs, and the effect is small. Hence, B_{ND} is approximately equal to B_{NA} for very small values of R. If f_{3dB} is chosen so that R is large, the frequency response of the digital filter for frequencies above f_{3dB} rolls off very rapidly. Thus, it immediately diverges from the analog frequency response, which rolls off at the rate of 20N dB/decade. It follows that B_{ND} and B_{NA} are significantly different for large values of R. A filter of high order has a steeper frequency response curve than a low order filter. Therefore, a

TABLE VI

COMPARISON OF

ANALOG AND DIGITAL NOISE-EQUIVALENT BANDWIDTHS

$R = \frac{f_{3dB}}{f_s}$	B_{ND}/B_{NA}			
	N=1	N=2	N=4	N=8
0.0015	0.995	1.000	1.000	unstable
0.0050	0.985	1.000	1.000	unstable
0.0150	0.956	0.999	1.000	1.000
0.0500	0.870	0.988	0.998	1.000
0.1500	0.716	0.939	0.987	0.997
0.2500	0.637	0.900	0.975	0.994

All results are rounded to the nearest thousandth.

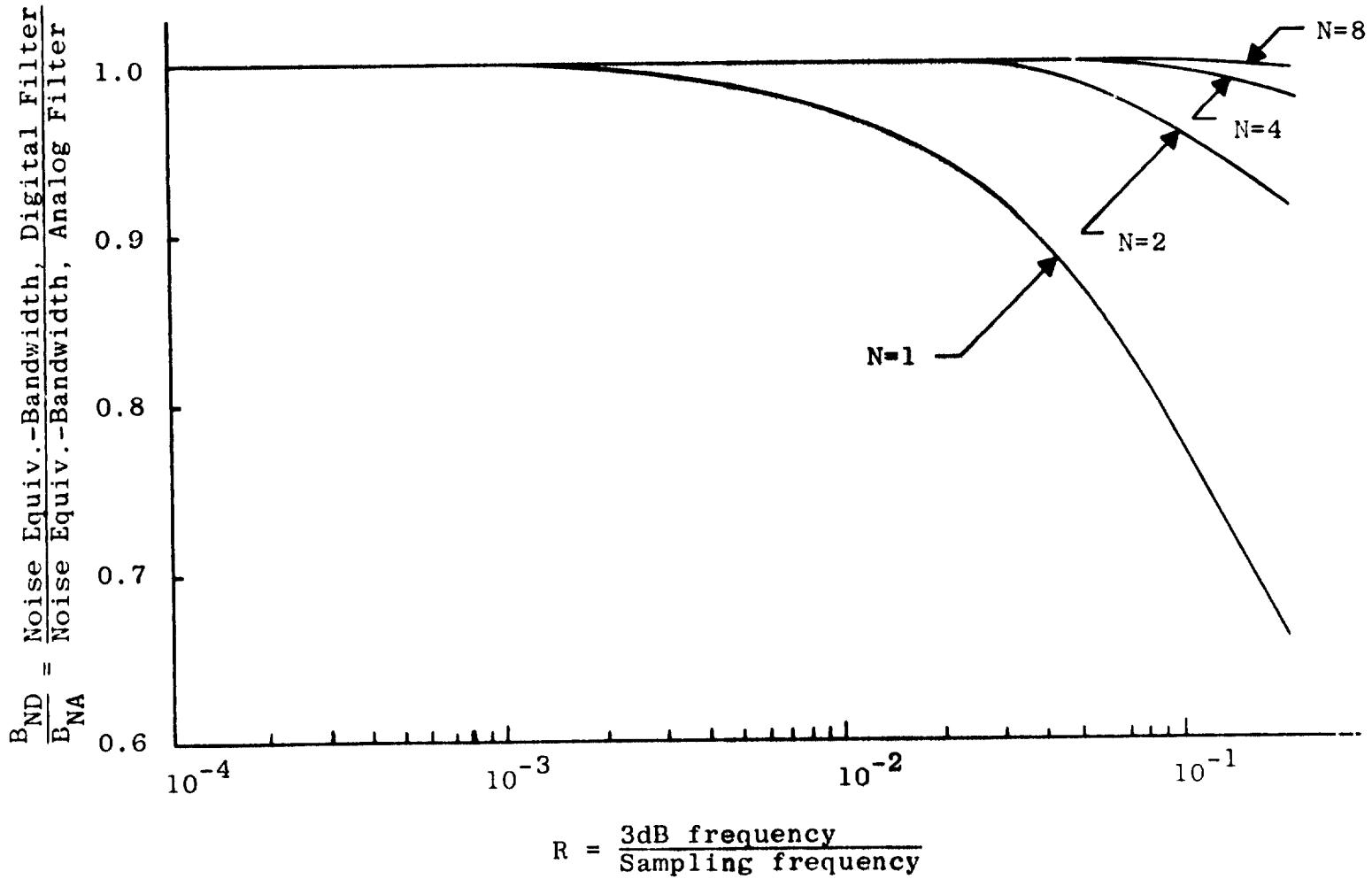

Figure 34. Comparison of Noise-Equivalent Bandwidths of Analog and Digital Lowpass Filters

greater value of R is required to show the difference between the values of B_{ND} and B_{NA} for a filter of high order. Exactly these effects can be seen in the plots.

For values of R greater than about 0.25, accurate results for B_{ND} cannot be obtained using the techniques described because the integration unavoidably becomes coarse. Fixing the sampling frequency fixes T, which is the step size for the integration. For values of R between 0.25 and 0.5, the response, h(t), becomes large for t slightly greater than zero and decreases very rapidly to terms of negligible magnitude. Thus, the sum essentially converges after only a few terms, and the rectangular integration becomes inaccurate. This is not usually a problem, since the digital filter is most often designed to have critical frequencies less than one-fourth the sampling frequency.

It is apparent that if an accurate value is required for the noise-equivalent bandwidth of a digital filter, it is usually necessary to measure that bandwidth. Only for high order and high sampling rates do the well known analog noise-equivalent bandwidths yield accurate approximations for the digital filter. Hence in the simulations, whenever a filter noise-equivalent bandwidth was required, the tabulated measurements were either utilized directly or used indirectly to interpolate to the required values of the filter parameters.

APPENDIX E

DATA FOR COMMUNICATION SYSTEM SIMULATIONS

This appendix contains tables of data collected in performing simulations of both analog and digital communication systems.

Tables VII through XII contain data for the various cases of the FM communication system simulations. Each table is labeled with the modulation type, the value of the deviation ratio and the PLL natural frequency used, as well as whether a hard-limiter was utilized.

Tables XIII through XVII contain the data which were obtained from the various digital communication system simulations. In each table, the particular system which generated the data is given at the top, followed by a group of parameters for that simulation. Data concerning the SNR and the error probabilities appear last in each case.

TABLE VII

FM SIMULATION, CASE 1

SINUSOIDAL MODULATION WITH DEVIATION RATIO = 5

f_n = 100, MODEL WITHOUT HARD-LIMITER

RUN NO.	IBM 370 - UMR			UNIVAC 1108 - NASA		
	$(SNR)_T$ DB	$(SNR)_D$ DB	SEED SET	RUN NO.	$(SNR)_T$ DB	$(SNR)_D$ DB
1	0.53	3.97	1	1	-0.03	-0.04
2	1.28	11.87	1	2	1.45	7.37
3	2.64	13.10	1	3	3.27	9.37
4	5.30	25.42	1	4	4.47	13.80
5	6.08	28.48	1	5	4.97	13.85
6	7.77	28.71	1	6	5.98	27.31
7	8.46	29.77	1	7	6.98	28.47
8	10.04	32.43	1	8	9.49	31.06
9	14.98	37.13	1	9	14.50	36.06
10	20.76	42.41	1	10	19.50	41.08
11	22.69	46.46	1	11	24.50	45.90
12	-0.11	3.56	2			
13	4.90	25.96	2			
14	10.45	32.63	2			
15	15.27	36.40	2			
16	19.98	41.82	2			
17	25.45	47.92	2			

TABLE VIII

FM SIMULATION, CASE 2

SINUSOIDAL MODULATION WITH DEVIATION RATIO =5

f_n = 100, MODEL WITH HARD-LIMITER

IBM 370 - UMR				UNIVAC 1108 - NASA			
RUN NO.	$(SNR)_T$ DB	$(SNR)_D$ DB	f_n	RUN NO.	$(SNR)_T$ DB	$(SNR)_D$ DB	f_n
1	0.37	5.71	100	1	-0.03	0.67	100
2	5.30	26.18	100	2	9.50	31.53	100
3	10.04	33.28	100	3	14.51	36.02	100
4	14.98	37.13	100	4	19.52	39.56	100
5	20.76	42.76	100	5	24.52	44.41	100
6	22.69	44.87	100				

TABLE IX

FM SIMULATION, CASE 3

SINUSOIDAL MODULATION WITH DEVIATION RATIO =5

f_n = 40, MODEL WITHOUT HARD-LIMITER

IBM 370 - UMR			UNIVAC 1108 - NASA		
RUN NO.	$(SNR)_T$ DB	$(SNR)_D$ DB	RUN NO.	$(SNR)_T$ DB	$(SNR)_D$ DB
1	0.58	7.14	1	-0.27	3.85
2	2.64	17.93	2	1.25	6.00
3	4.32	25.52	3	2.96	14.73
4	5.57	27.75	4	4.46	25.67
5	8.34	29.19	5	4.97	26.26
6	9.91	31.80	6	9.49	30.90
7	15.44	37.11	7	14.51	35.78
8	20.94	43.21	8	19.50	40.60
9	24.53	46.31	9	24.53	45.11

TABLE X

FM SIMULATION, CASE 4

SINUSOIDAL MODULATION WITH DEVIATION RATIO = 5

f_n = 40, MODEL WITH HARD-LIMITER

IBM 370 - UMR				UNIVAC 1108 - NASA			
RUN NO.	$(SNR)_T$ DB	$(SNR)_D$ DB	f_n	RUN NO.	$(SNR)_T$ DB	$(SNR)_D$ DB	f_n
1	0.27	6.78	40	1	- .26	1.28	40
2	5.52	25.00	40	2	.47	25.33	40
3	9.91	31.17	40	3	9.50	31.18	40
4	15.44	37.22	40	4	14.51	35.76	40
5	20.68	42.30	40	5	19.52	38.94	40
6	24.53	44.51	40	6	24.53	43.07	40

TABLE XI

FM SIMULATION, CASES 5 AND 6

SINUSOIDAL MODULATION WITH DEVIATION RATIO = 1

f_n = 20, RESULTS BOTH WITH AND WITHOUT HARD-LIMITER

RUN ON IBM 370 - UMR

WITHOUT HARD-LIMITER - CASE 5				WITH HARD-LIMITER - CASE 6			
RUN NO.	$(SNR)_T$ DB	$(SNR)_D$ DB	SEED SET	RUN NO.	$(SNR)_T$ DB	$(SNR)_D$ DB	SEED SET
1	0.05	-4.86	1	15	0.06	-5.97	1
2	5.51	6.37	1	16	6.23	10.65	1
3	11.23	12.53	1	17	10.24	13.14	1
4	14.96	15.80	1	18	15.53	18.84	1
5	20.53	22.92	1	19	20.35	20.82	1
6	25.47	28.63	1	20	23.97	27.12	1
7	28.97	31.94	1	21	31.03	35.49	1
8	0.608	-6.36	2	22	0.608	-8.37	2
9	4.70	9.21	2	23	4.70	7.34	2
10	9.98	11.15	2	24	10.06	11.35	2
11	15.57	18.44	2	25	15.57	18.14	2
12	19.82	22.56	2	26	19.82	22.47	2
13	25.64	28.64	2	27	25.64	27.17	2
14	29.30	32.59	2	28	29.30	31.31	2

TABLE XII

FM SIMULATION, CASES 7 AND 8

RANDOM GAUSSIAN MODULATION WITH DEVIATION RATIO = 1

f_n = 60, RESULTS BOTH WITH AND WITHOUT HARD-LIMITER

RUN ON IBM 370 - UMR

WITHOUT HARD-LIMITER - CASE 7				WITH HARD-LIMITER - CASE 8			
RUN NO.	$(SNR)_T$ DB	$(SNR)_D$ DB	SEED SET	RUN NO.	$(SNR)_T$ DB	$(SNR)_D$ DB	SEED SET
1	0.29	-10.88	1	13	0.29	-9.77	1
2	5.02	1.06	1	14	4.82	-15.38	1
3	10.60	6.07	1	15	10.60	6.03	1
4	14.37	11.79	1	16	14.37	11.64	1
5*	20.17	14.68	1	17*	20.18	14.41	1
6*	25.16	19.26	1	18*	25.16	18.45	1
7*	-0.42	-9.65	2	19*	0.67	-9.59	2
8	4.77	-0.12	2	20	4.98	-0.48	2
9	10.03	5.10	2	21	10.03	5.18	2
10*	14.94	-19.17	2	22*	14.87	9.52	2
11	19.87	13.43	2	23	19.87	13.54	2
12	24.97	15.32	2	24	24.97	14.56	2

*Flagged by SNRMSE for nonstationarity

NOTE: All points except runs 10 and 14 were plotted, whether flagged or not.

TABLE XIII

COHERENT ASK SIMULATION

Carrier Frequency	- 1 MHz
Time-Bandwidth Product (4-Pole Filter)	- 2.244
Sampling Frequency	- 10 MHz
Number of Symbols Simulated	- 1022
Number of Samples Processed	- 30,700
Number of Samples Used to Obtain $\hat{SNR}$	- 20,000

Run No.	$\hat{SNR}$ (dB)	$\hat{P}_E(\hat{SNR})$	P_E(count)	Number of Errors	Phase Error Var.(Rad^2)
1	-4.21	0.1781	0.1703	174	0.0054
2	-0.69	0.0833	0.0675	69	0.0082
3	1.37	0.0397	0.0352	36	0.0122
4	4.17	0.00774	0.00783	8	0.0171
5	4.97	0.00398	0.00196	2	0.0172
6	-3.15	0.1485	0.1184	121	0.00910
7	0.46	0.0572	0.0470	48	0.0196
8	2.61	0.0216	0.0166	17	0.0249
9	5.60	0.00217	--	0	0.0219
10	6.48	0.000797	--	0	0.0205

NOTE: Runs 1-5 utilize filtered signal-plus-noise

Runs 6-10 add filtered noise to the signal

TABLE XIV

COHERENT FSK SIMULATION

Frequency for Symbol 1	- 1.00 MHz
Frequency for Symbol 2	- 1.67 MHz
Time-Bandwidth Product (4-Pole Filter)	- 4.458
Sampling Frequency	- 15 MHz
Number of Symbols Simulated	- 1023
Number of Samples Processed	- 46,100
Number of Samples Used to Obtain $\hat{SNR}$	- 20,000

Run No.	$\hat{SNR}$ (dB)	$\hat{P}_E(\hat{SNR})$	P_E(count)	Number of Errors	Phase Error Var. (Rad^2)	
1	-7.20	0.1782	0.1750	179	0.033	0.030
2	-3.60	0.0816	0.0763	78	0.015	0.015
3	-1.47	0.0373	0.0381	39	0.010	0.011
4	1.50	0.00605	0.00196	2	0.006	0.008
5	2.37	0.00277	0.000978	1	0.005	0.008
6	-7.41	0.1842	0.1779	182	0.036	0.036
7	-3.71	0.0842	0.0841	86	0.015	0.015
8	-1.52	0.0382	0.0313	32	0.009	0.009
9	1.51	0.00600	0.00599	6	0.004	0.004
10	2.40	0.00269	0.00269	1	0.004	0.004

NOTE: Runs 1-5 utilize filtered signal-plus-noise

Runs 6-10 add filtered noise to the signal

TABLE XV

NONCOHERENT FSK SIMULATION
WITHOUT RAYLEIGH FADING

Frequency for Symbol 1	**- 1.00 MHz**
Frequency for Symbol 2	**- 1.67 MHz**
Time-Bandwidth Product (Predetection 4-Pole Filter)	**- 6.390**
Sampling Frequency	**- 15 MHz**
Number of Symbols Simulated	**- 1023**
Number of Samples Processed	**- 46,100**
Number of Samples Used to Obtain SNR	**- 20,000**

Run No.	$\hat{SNR}$ (dB)	$\hat{P}_E(\hat{SNR})$	P_E(count)	Number of Errors
1	-5.76	0.2141	0.2180	223
2	-2.16	0.0717	0.0850	87
3	-0.02	0.0208	0.0303	31
4	0.94	0.0095	0.0117	12

TABLE XVI

NONCOHERENT FSK SIMULATION IN

SINGLE RAYLEIGH FADING CHANNEL

All system and simulation parameters are identical to those given in Table XV

Run No.	$\hat{SNR}$ (dB)	$\hat{P}_E(\hat{SNR})$	P_E(count)	Number of Errors
1	-8.00	0.3319	0.3509	359
2	-2.98	0.1917	0.2190	224
3	2.02	0.0822	0.0958	98
4	4.06	0.0547	0.0499	51
5	6.97	0.0296	0.0352	36
6	11.79	0.0102	0.0978	10

TABLE XVII

DIVERSITY TRANSMISSION IN

THE RAYLEIGH FADING CHANNEL

All system and simulation parameters are identical to those given in Table XV. Channel noise was adjusted to maintain $(\bar{E}_s/N_o) \approx 15$ dB

Run No.	Order of Diversity	SNR (dB) Per Channel	Theo.P_E at 15 dB	P_E(count)	Number of Errors
1	1	6.97	0.0297	0.0323	33
2	2	4.06	0.00910	0.00978	10
3	3	2.89	0.00448	0.00489	5
4	4	1.83	0.00283	0.00489	5

APPENDIX F

THE EFFECT OF SNR ESTIMATION ERRORS ON P_E ESTIMATES

When the SNR (signal-to-noise ratio) is estimated at a point in a particular simulation, errors in the estimate generally occur. The error, ε, in the SNR estimate, $\hat{SNR}$, can be defined by

$$\hat{SNR} = (1+\varepsilon)(SNR) \tag{64}$$

so that $\varepsilon=0$ corresponds to an exactly correct estimate. Many times in applications to digital systems, the probability of symbol error, P_E, is estimated using $\hat{SNR}$. Thus, the error in $\hat{SNR}$ produces an error in the symbol error probability estimate, $\hat{P}_E$, and it is useful to evaluate the error in $\hat{P}_E$ resulting from the error in $\hat{SNR}$. The sensitivity of $\hat{P}_E$ to the measurement error, ε, is dependent upon the mapping from $\hat{SNR}$ to $\hat{P}_E$ for the particular system involved.

A number of coherent systems operating in additive Gaussian noise environments can be studied using a general mapping of the form

$$P_E = \frac{1}{2}\,\mathrm{erfc}(\sqrt{z}) \tag{65}$$

where

$$\mathrm{erfc}\,(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-\alpha^2}\,d\alpha.$$

The quantity z represents the product of a system dependent constant, β, with the ratio of the average energy per

transmitted symbol, E_s, to the noise power spectral density, N_o. Therefore, (65) may be written

$$P_E = \frac{1}{2} \operatorname{erfc} \sqrt{\beta \cdot \left(\frac{E_s}{N_o}\right)}.$$

It was shown in the section on application to digital systems that

$$\frac{E_s}{N_o} = (SNR)(T_s B_n)$$

where $(T_s B_n)$ is the time-bandwidth product for the system. Therefore the general mapping may be written

$$P_E = \frac{1}{2} \operatorname{erfc} \sqrt{\beta (SNR)(T_s B_n)} .$$

This relation may be applied to many communication systems. If $\beta = \frac{1}{2}$ is used, the mapping is valid for a coherent binary ASK or FSK system. With $\beta = 1$, the mapping applies to a phase-reversal keying system [77]. The mapping can also be applied to a quadriphase phase-shift keying system [78].

Since many systems may be represented by this general mapping, it is helpful to investigate the effect of errors in $\hat{SNR}$ on the resulting error probability estimate obtained using

$$\hat{P}_E = \frac{1}{2} \operatorname{erfc} \sqrt{\beta (\hat{SNR})(T_s B_n)}.$$

Substituting from (64) yields

$$P_E = \frac{1}{2} \operatorname{erfc} \sqrt{\beta(SNR)(T_s B_n)(1+\varepsilon)}.$$

Noting

$$z = \beta \frac{E_s}{N_o} = \beta(SNR)(T_s B_n)$$

yields

$$\hat{P}_E = \frac{1}{2} \operatorname{erfc} \sqrt{z(1+\varepsilon)} \qquad (66)$$

which gives a general form for analysis. Plots of (66) are given in Figure 35 for values of ε from -0.5 to 0.5. The curve with ε=0 represents the mapping when no error is present. The distance along the ordinate between this curve and any other curve is the error in $\hat{P}_E$ which corresponds to the value of ε specified. It can be seen that for large vlaues of z, the curves diverge, and the slopes of the curves increase. This leads to larger errors in $\hat{P}_E$ for a given value of ε as z is increased. Figure 35 is not particularly convenient for studying the error in $\hat{P}_E$, so another representation is helpful.

In many applications P_E estimates are typically specified by giving the order of magnitude. For example $P_E = 0.871 \times 10^{-7}$ would usually be simply referred to as $P_E \approx 10^{-7}$. Often the order of magnitude is an adequate specification for P_E and for specifying error in $\hat{P}_E$. Percentage error measurements are not convenient to use for specifying the error in $\hat{P}_E$ in most cases. An appropriate way to specify the error in $\hat{P}_E$ is provided by defining

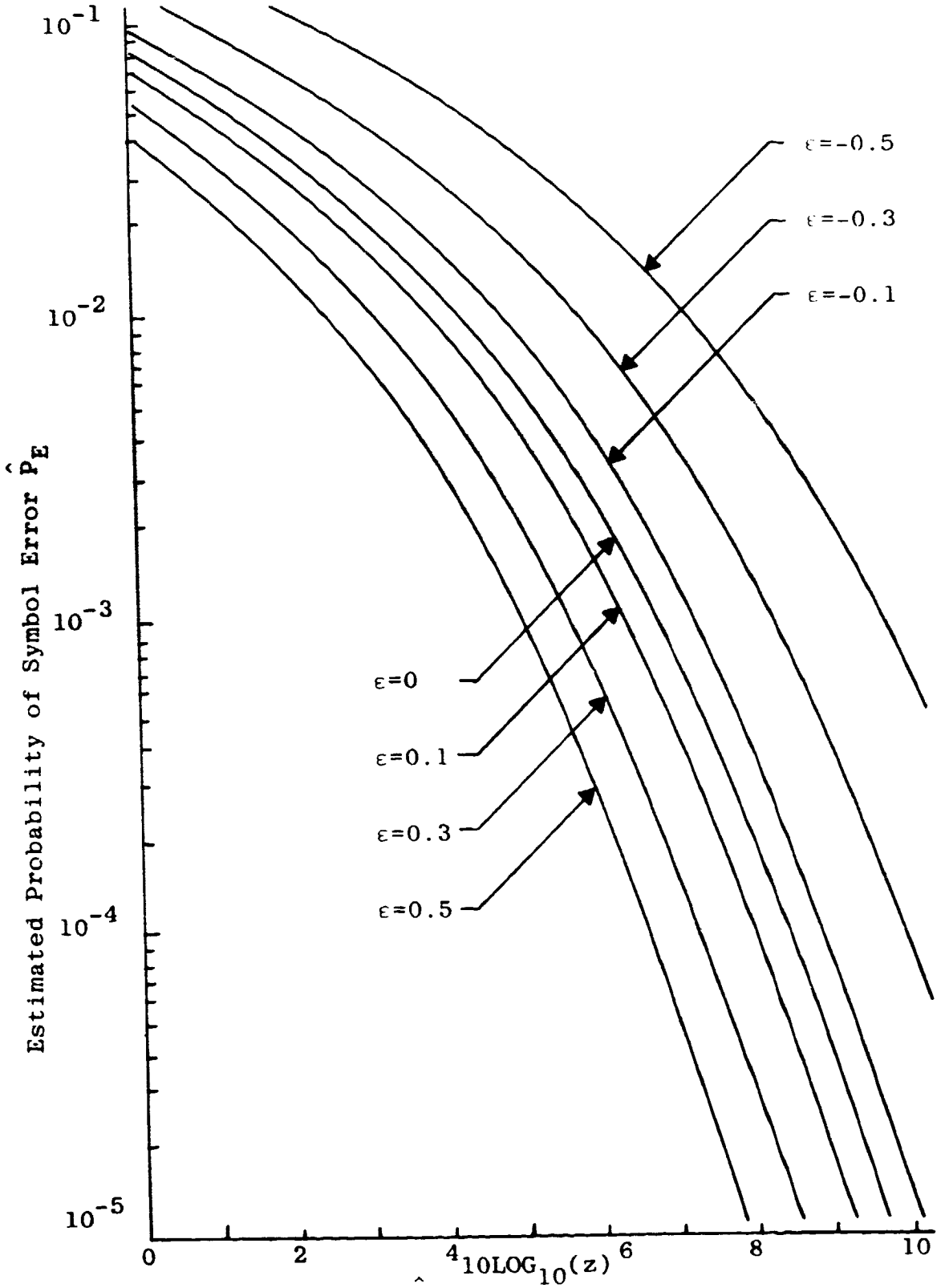

Figure 35. Sensitivity of $\hat{P}_E$ Error to Errors in SNR Estimation

$$\gamma = \mathrm{LOG}_{10}\left(\frac{\hat{P}_E}{P_E}\right).$$

Using this expression, $\gamma=0$ corresponds to $\hat{P}_E = P_E$, indicating zero error. Additionally, $(\hat{P}_E/P_E) = 10$ yields $\gamma = 1$, $(\hat{P}_E/P_E) = 1/10$ yields $\gamma = -1$, and $(\hat{P}_E/P_E) = 1/100$ yields $\gamma = -2$, and so on. Thus, γ is the order of magnitude error in the estimate, as desired.

A plot of γ versus $10\cdot\mathrm{LOG}_{10}(z)$ is given in Figure 36. This plot illustrates the sensitivity of the error in $\hat{P}_E$ to the error in the estimate for z. The divergence of the curves for increasing values of z indicates how a given value of ε produces larger errors in $\hat{P}_E$ for larger values of SNR. The box shown in the figure encompasses the region where the coherent ASK and FSK simulations were performed. This region is shown in Figure 37. On this plot the results obtained in the coherent ASK and FSK simulations are shown. The direct error counts are used to define the probability of symbol error, P_E, and the estimate for the symbol error probability computed using SNR estimation defines $\hat{P}_E$. For each point where less than five errors were counted, the number of errors is shown next to the point. The data for the simulations are tabulated in Appendix E.

It can be seen that excellent results were obtained. In all cases where more than five errors were counted, so

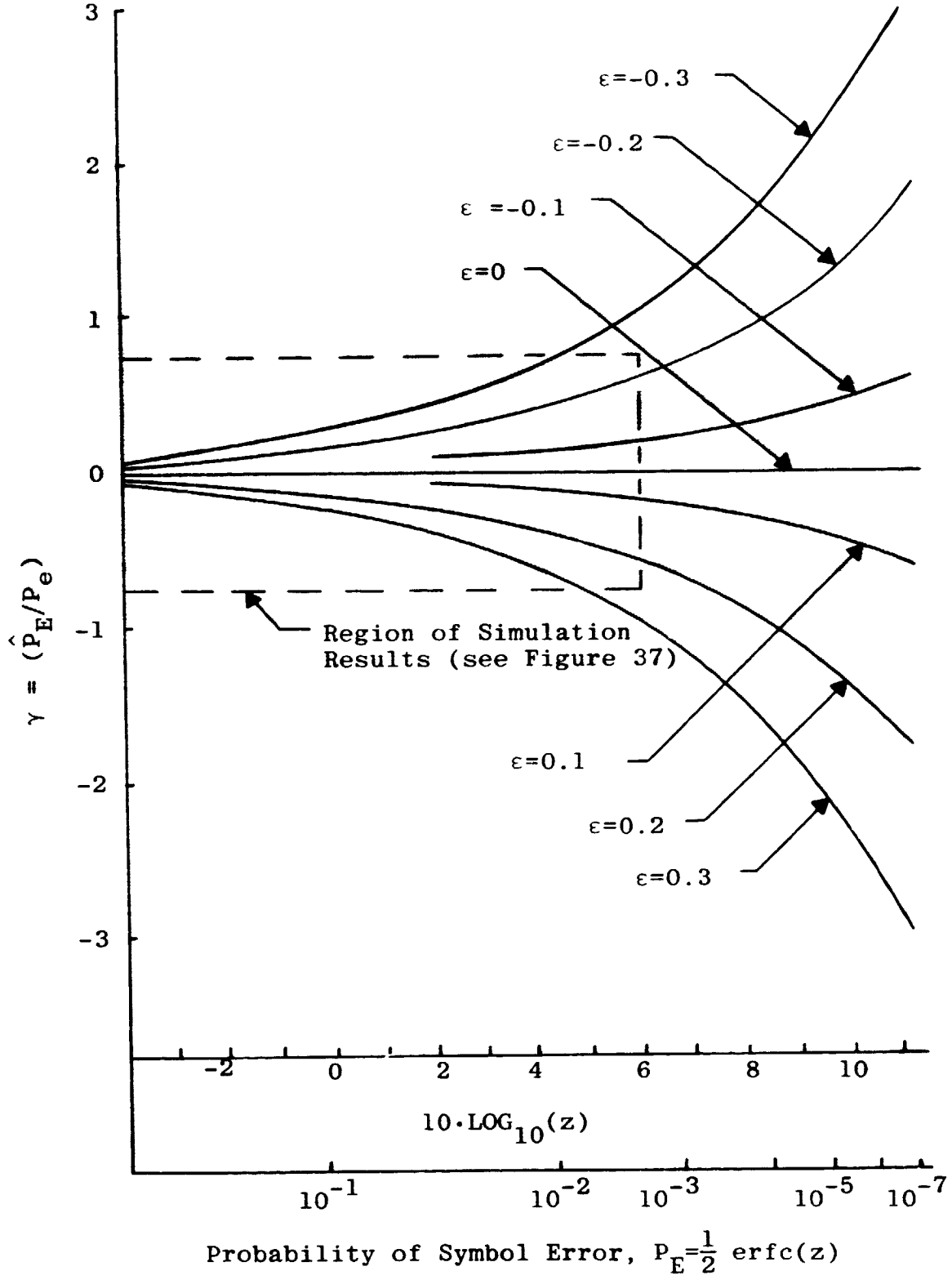

Figure 36. γ Versus P_E and $10 \cdot LOG_{10}(z)$

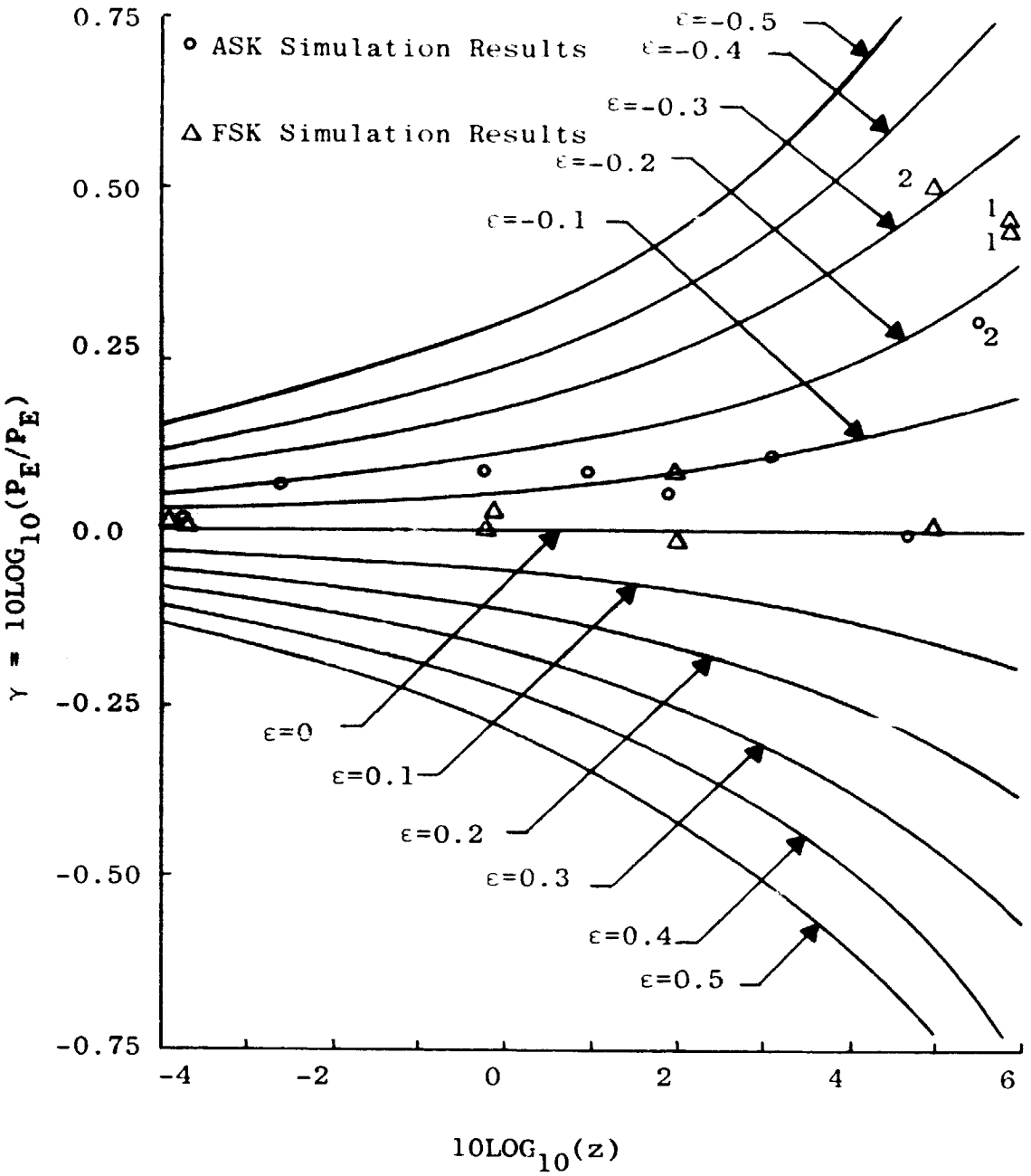

Figure 37. γ Versus $10 \cdot LOG_{10}(z)$ with Simulation Results

that the error count is useful in estimating the error probability, the points remain between $\varepsilon = -0.2$ and $\varepsilon = 0.1$ and correspond to very small values of γ. Most points remain within or nearly within the $\varepsilon = \pm 0.1$ range. Simulation results and the confidence intervals derived earlier show that the reliability of the SNR estimator is such that for higher values of the SNR, the values of ε obtained would usually be approximately the same as shown here. For errors in the range $\varepsilon = \pm 0.1$ and values of P_E as low as 10^{-7}, Figure 36 shows that γ remains between plus and minus one. Thus, P_E estimates should usually be expected within one order of magnitude of the correct vlaue, even for P_E values as low as 10^{-7}, when the SNR estimation routine is applied to a system with this mapping from the SNR to P_E. It should be remembered, however, that for a system with a different mapping from the SNR to P_E, different curves would result.

END

DATE

FILMED

JUL 12 1978