INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION
INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
I.C.T.P., P.O. BOX 586, 34100 TRIESTE, ITALY, CABLE: CENTRATOM TRIESTE

UNITED NATIONS INDUSTRIAL DEVELOPMENT ORGANIZATION
INTERNATIONAL CENTRE FOR SCIENCE AND HIGH TECHNOLOGY

H4.SMR/537-10

SECOND COLLEGE ON THEORETICAL AND EXPERIMENTAL RADIOPROPAGATION PHYSICS
(7 January - 1 February 1991)

Co-sponsered by ICTP, URSI, ICSU
and with the participation of ICS

Digital system performance

Algorithms and software

A. D. Spaulding
Institute for Telecommunication Sciences
U.S. Department of Commerce
Boulder, Colorado

DIGITAL SYSTEM PERFORMANCE

ALGORITHMS AND SOFTWARE

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

DIGITAL SYSTEM PERFORMANCE SOFTWARE UTILIZING NOISE MEASUREMENT DATA

A. D. Spaulding*

This report summarizes techniques that use the measured instantaneous envelope statistics of arbitrary noise or interference processes to calculate the degradation these processes cause to digital communication systems. Computer implementation of the techniques are also given. The computer algorithms are designed for the data obtained from a general purpose noise measurement device, termed the DM-4 (for "distribution meter-model number 4") recently developed by NTIA/ITS. For illustration and for comparison with theoretical results, two noise examples are employed, one for "narrowband" interference, and one for "broadband" interference. These examples are taken from the noise models recently developed by Middleton.

Key Words: computer algorithms; digital system performance; non-Gaussian noise; system performance software.

1. INTRODUCTION

Most currently used receiving systems are those which are optimum in Gaussian noise. Unfortunately, the actual interference environment is almost never Gaussian in character, but usually quite different, being impulsive in nature. By "impulsive" we mean only that there are significant probabilities of quite large instantaneous values of noise, which is a more general definition in that we can, and do, have both broadband (the usual definition, e.g., automotive ignition noise) and narrowband (e.g., various combinations of interfering signals) "impulsive" processes. Recently there have been receiving systems designed to match this actual interference (e.g., Spaulding and Middleton, 1977; Middleton, 1979). However, it is the purpose of this short report to provide computer programs that will use noise measurements to calculate the performance of "normal" digital systems in arbitrary noise or interference (including, of course, Gaussian noise). By "normal" systems we mean those that are optimum in Gaussian noise, and, therefore, suboptimum in any other kind of noise or interference. The "normal" digital systems are "matched filter" or "correlation" systems. The common digital systems covered here are:

*The author is with the Institute for Telecommunication Sciences, National Telecommunications and Information Administration, U.S. Department of Commerce, Boulder, Colorado 80303.

1. Binary non-coherent frequency shift keying (NCFSK);
2. Binary differentially coherent phase shift keying (DCPSK);
3. Binary coherent phase shift keying (antipodal or CPSK);
4. Binary coherent frequency shift keying (othogonal or CFSK); and
5. Binary coherent ON-OFF keying.

In addition, the coherent signal detection system (Neyman-Person detection) is included. The performance of other systems, such as M level systems and minimal shift keying systems, can usually be obtained by appropriate extensions of the techniques summarized here. However, these extensions are not always straightforward.

Recently, a general purpose noise measurement device, termed the DM-4 (for "distribution meter-model number 4") was developed by NTIA/ITS (Matheson, 1980, DM-4 operation and maintenance manual, NTIA-TM 80-50), and the software presented here is designed to work specifically with the DM-4 measurements although no actual DM-4 measurements are used. For any received noise process, Z(t), we denote the probability density function (pdf) of the instantanecus amplitude by $p_Z(z)$. Denote the envelope of this received noise process by R(t) and pdf of the envelope by $p_R(r)$. The DM-4 measures the amplitude probability distribution (APD) of the envelope or Prob$[R > R_0]$, which we will denote by $P_R(r)$. Note that

$$p_R(r) = - \frac{d}{dr} P_R(r). \quad (1)$$

The DM-4 can also measure the average crossing rate characteristic of the received noise envelope, but these measurements are not considered here. While the DM-4 measures the APD in terms of the actual levels exceeded, referred to the input of the receiving system via calibration (e.g., dBm), we normally require the $P_R(r)$ or $p_Z(z)$ in normalized form so that the mean noise power is equal to 1. When we consider our desired signals, then the mean signal power is also the signal-to-noise ratio. For example, for the signal $\sqrt{2S}$ Cos $(\omega_0 t)$, S is the signal power and also the signal-to-noise ratio. The DM-4 measures the APD at 31 calibrated levels, i.e., measures Prob$[R > R_0]$ for 31 values of R_0. It uses a maximum sampling rate of 20 MHz, which means it can measure the output waveforms from systems of about 10 MHz bandwidth (IF) or less. The DM-4 is designed to work with the detected logarithmic output of modern spectrum

analyzers and EMI meters, so that the 31 DM-4 levels are equally spaced in voltage, corresponding to 31 levels equally spaced in dB when referred to the receiving system input. For our purposes here we only need the calibrated 31 levels (not necessarily equally spaced) and the Prob$[R > R_0]$ for each of these levels as the input data to the system performance algorithms.

For illustration and for comparison with theoretical results we make use of recently developed noise models. The models were developed for ITS by Middleton (1977, 1980) and Spaulding (1977). Two examples are selected, one for "narrowband" interference, termed Class A, and one for "broadband" interference, termed Class B. These examples of noise, from actual measurements (not DM-4 however), are presented in the next section (Section 2) and are used then to simulate DM-4 "measurements."

Section 3 presents the system performance algorithms, sample performance calculations, comparison with theoretical results when possible, and discussion of the algorithms. An Appendix then contains the actual computer software listings in FORTRAN.

2. THE IMPULSIVE NOISE MODEL, TEST EXAMPLES

Recent work by Middleton has led to the development of a physical-statistical model for radio noise and interference. This model has been used to develop optimum detection algorithms for a wide range of communications problems (Spaulding and Middleton, 1977). It is this model which we will use here to simulate DM-4 "measurements." The Middleton model is the only one proposed to date in which the parameters of the model are determined <u>explicitly</u> by the underlying physical mechanisms (e.g., source density, beam-patterns, propagation conditions, and emission waveforms). It is also the first model which treats narrowband interference processes (termed Class A), as well as the traditional broadband processes (Class B). The model is also canonical in nature in that the mathematical forms do not change with changing physical conditions. For a large number of comparisons of the model with measurements and for the details of the derivation of the model, see Middleton (1974, 1976, 1977, 1978a, 1978b) and Spaulding (1977). We only summarize the results of the model which we need here.

For the class A model, the expression for the pdf of the received noise signal, Z(t), is

$$P_Z(z) = e^{-A} \sum_{m=0}^{\infty} \frac{A^m}{m!\sqrt{2\pi\sigma_m^2}} e^{-z^2/2\sigma_m^2} , \quad (2)$$

where

$$\sigma_m^2 = \frac{m/A + \Gamma'}{1 + \Gamma'} , \quad (3)$$

and for the envelope, R(t),

$$P[R > R_0] = e^{-A} \sum_{m=0}^{\infty} \frac{A^m}{m!} e^{-R_0^2/\sigma_m^2} . \quad (4)$$

The Class A model has two parameters, A and Γ'. A is termed the impulsive index, and as A becomes larger (~ 10), the noise approaches Gaussian (still narrowband) and Γ' is the ratio of the energy in the Gaussian portion of the noise to the energy in the non-Gaussian component. In the above, the rms value of Z is equal to 1, i.e., the process is already normalized.

For our sample DM-4 "measurement" of Class A noise, Figure 1 gives a measured Class A distribution and the appropriate model parameters are A=0.35 and $\Gamma'=0.5 \times 10^{-3}$. The first program, APDA, given in the Appendix, simply generates our "measurement" data. We compute $P_R(r)$ at 31 levels, starting with -59 dB (see Table 1) with the levels 3 dB apart. We further assume that the measurements after $P_R(r) = 10^{-6}$ are zero. This gives us some zero "measurements" that are likely from actual DM-4 measurements. The dynamic range covered, therefore, is 90 dB, from -59 to 31 dB. If we were actually measuring the APD of Figure 1, we would probably adjust the levels, so that the significant portion of the distribution (-50 to 20 dB, say) was more accurately covered. The above procedure (3 dB spacing), however, will be a better test of the system performance algorithms, but we want to keep in mind that accurracy can be improved by proper adjustment of the measurement levels. Once we have determined our 31 values of $P_R(r)$, we then assign these values to arbitrary (unnormalized) levels (3 dB apart) to check the normalization portion of the algorithms.

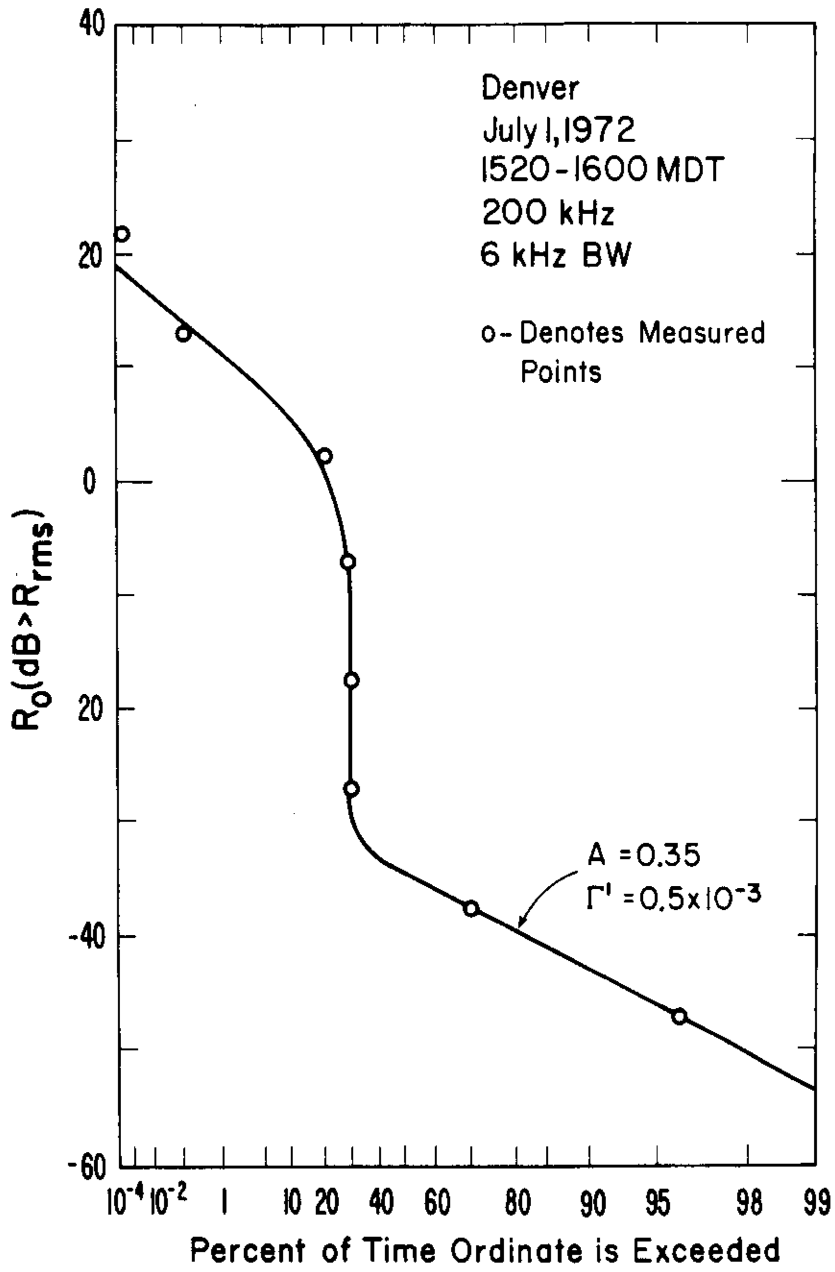

Figure 1. Example of Class A noise.

For the Class B model the pdf of the received instantaneous amplitude is:

$$p_Z(z) = \frac{e^{-z^2/\Omega}}{\pi\sqrt{\Omega}} \sum_{m=0}^{\infty} \frac{(-1)^m}{m!} A_\alpha^m \, \Gamma\left(\frac{m\alpha + 1}{2}\right) \, {}_1F_1\left(\frac{-m\alpha}{2} ; \frac{1}{2} ; \frac{z^2}{\Omega}\right) , \quad (5)$$

$$-\infty \le z \le \infty$$

where, ${}_1F_1$ is a confluent hypergeometric function (Abramowitz and Stegun, 1964). The model has three parameters, α, A_α, and Ω. [A more detailed and complete model involving additional parameters has been developed, but (5) above is quite sufficient for our purposes.] The parameters α and A_α are intimately involved in the physical processes causing the interference. Again, definitions and details are contained in the references. The parameter Ω is a normalizing parameter. In the references, the normalization is $\Omega=1$, which normalizes the process to the energy contained in the Gaussian positon of the noise. Here, we use a value of Ω which normalizes the process (z values) to the measured energy in the process. We cannot normalize to the energy computed from the model, since for (5), the second moment (or any moment) does not exist (i.e., is infinite). This is a typical problem with most such models for broadband impulsive noise. While the more complete model removes this problem, use of (5) will not limit us here. The result corresponding to (5) for the APD is:

$$P(R > R_0) = e^{-R_0/\Omega} \left[1 - \frac{R_0^2}{\Omega} \sum_{m=1}^{\infty} \frac{(-1)^m}{m!} A_\alpha^m \, \Gamma\left(1 + \frac{m\alpha}{2}\right) \, {}_1F_1\left(1 - \frac{m\alpha}{2} , 2; \frac{R_0^2}{\Omega}\right) \right] \quad (6)$$

$$0 \le R_0 < \infty$$

On Figure 2, the parameter Ω was calculated with the assumption that $P_R(r)$ is zero for values of $R_0 > 40$ dB. The program APDB, listed in the Appendix, is used to generate "measurement" from the APD of Figure 2. As before, a 90 dB dynamic range (here, -40 dB to 50 dB) is covered in 3-dB steps. Also, once the 31 values of $P_R(r)$ are obtained, we assign arbitrary 3-dB step values for the corresponding R_0 values to simulate actual measurements. The Class A example of Figure 1 is from Spaulding and Middleton (1977) and the Class B example of Figure 2 is from Evans and Griffiths (1974). Table 1 shows the outputs of AFDA and APDB, and these then become our example "measurements."

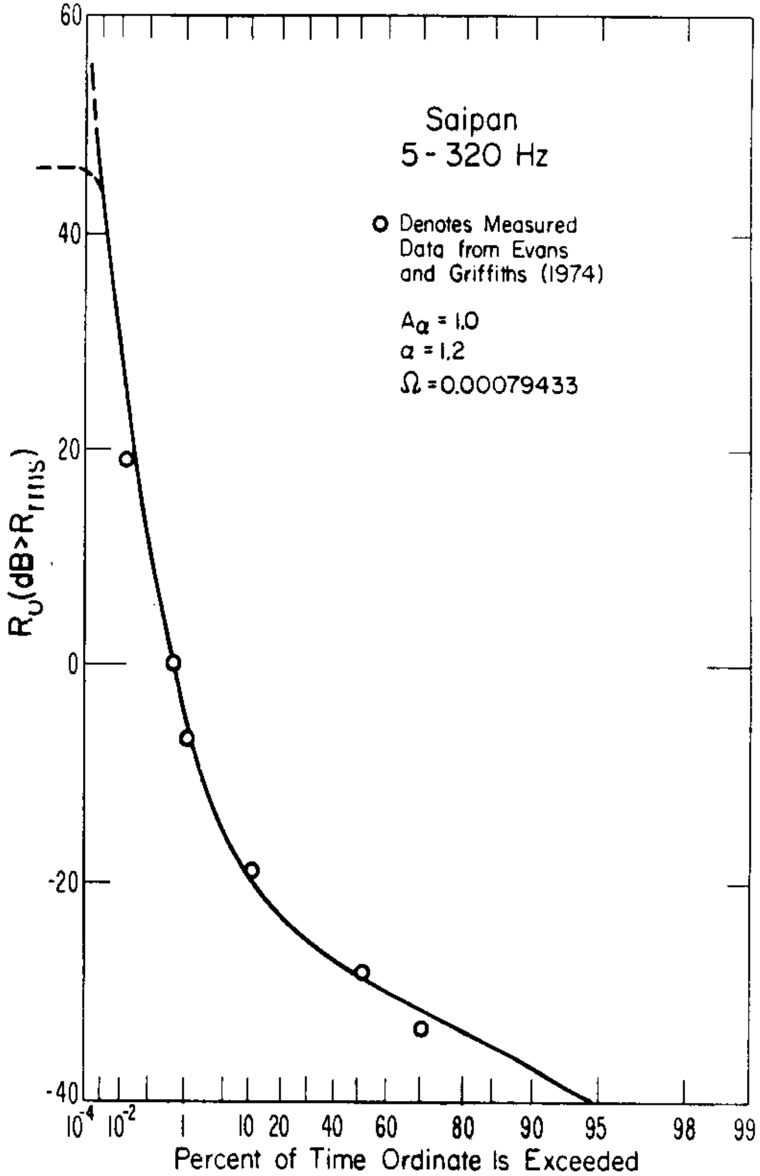

Figure 2. Example of Class B noise.

Table 1. Class A and Class B Noise Data from Figures 1 and 2

Class A

R_0(dB)	Prob $[R > R_0]$
-5.90000E+01	9.98227E-01
-5.60000E+01	9.96467E-01
-5.30000E+01	9.92968E-01
-5.00000E+01	9.86038E-01
-4.70000E+01	9.72418E-01
-4.40000E+01	9.46038E-01
-4.10000E+01	8.96434E-01
-3.80000E+01	8.08473E-01
-3.50000E+01	6.69554E-01
-3.20000E+01	4.94632E-01
-2.90000E+01	3.51942E-01
-2.60000E+01	2.99700E-01
-2.30000E+01	2.94870E-01
-2.00000E+01	2.94368E-01
-1.70000E+01	2.93432E-01
-1.40000E+01	2.91573E-01
-1.10000E+01	2.87901E-01
-8.00000E+00	2.80719E-01
-5.00000E+00	2.66942E-01
-2.00000E+00	2.41529E-01
1.00000E+00	1.98120E-01
4.00000E+00	1.34301E-01
7.00000E+00	6.37192E-02
1.00000E+01	1.67017E-02
1.30000E+01	2.11605E-03
1.60000E+01	1.04690E-04
1.90000E+01	1.07174E-06
2.20000E+01	0.
2.50000E+01	0.
2.80000E+01	0.
3.10000E+01	0.

Class B

R_0(dB)	Prob $[R > R_0]$
-4.00000E+01	9.43528E-01
-3.70000E+01	8.91427E-01
-3.40000E+01	7.98434E-01
-3.10000E+01	6.49084E-01
-2.80000E+01	4.51644E-01
-2.50000E+01	2.63073E-01
-2.20000E+01	1.44504E-01
-1.90000E+01	8.57505E-02
-1.60000E+01	5.38743E-02
-1.30000E+01	3.46832E-02
-1.00000E+01	2.25913E-02
-7.00000E+00	1.48055E-02
-4.00000E+00	9.73559E-03
-1.00000E+00	6.41403E-03
2.00000E+00	4.23040E-03
5.00000E+00	2.79201E-03
8.00000E+00	1.84343E-03
1.10000E+01	1.21743E-03
1.40000E+01	8.04130E-04
1.70000E+01	5.31191E-04
2.00000E+01	3.50915E-04
2.30000E+01	2.31830E-04
2.60000E+01	1.53161E-04
2.90000E+01	1.01190E-04
3.20000E+01	6.68539E-05
3.50000E+01	4.41694E-05
3.80000E+01	2.91821E-05
4.10000E+01	1.92803E-05
4.40000E+01	1.27383E-05
4.70000E+01	0.
5.00000E+01	0.

Finally, white Gaussian noise is a special case, and the performance of digital systems in white Gaussian noise has been treated in great detail. Here, we will occasionally refer to the well-known results of system performance in white Gaussian noise for comparison. The pdf for the instantaneous amplitude for Gaussian noise (mean noise power = 1) is:

$$p_Z(z) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2} , \tag{7}$$

and for the corresponding envelope

$$p_R(r) = r e^{-r^2/2} , \text{ and} \tag{8}$$

$$P_R(r) = e^{-r^2/2} . \tag{9}$$

3. SYSTEM PERFORMANCE CALCULATIONS

In this Section we want to present the results which are most advantageous for our use. We want to develop system performance algorithms which do not require particularly sophisticated numerical analysis techniques and which can be used on small scale computers.

We start with the simplest. For arbitrary additive interference which is independent from an integration period (bit length) to the next and which has uniformly distributed phase, Montgomery (1954) has shown that the probability of binary bit error, P_e, for NCFSK (non-coherent frequency shift keying) is given by:

$$P_e = \frac{1}{2} \text{ Prob [noise envelope > rms signal level]}. \tag{10}$$

While Montgomery's result for NCFSK is in terms of the noise and signal envelopes at the input to an ideal discriminator, it has been shown (White, 1966) that the result is also applicable to most common FSK receivers (bandpass-filter discriminator receivers and matched-filter envelope detection receivers). Using (10), then, the performance for NCFSK can be obtained instantly from the APD measurements, once the APD has been normalized to its rms level. For example, if the two NCFSK waveforms are given by

$$S_1(t) = \sqrt{2S}\ \cos(\omega_1 t + \phi), \text{ and}$$
$$S_2(t) = \sqrt{2S}\ \cos(\omega_2 t + \phi), \tag{11}$$

where ϕ is the unknown (uniformly distributed) phase (i.e., incoherent signaling), ω_1 and ω_2 are the two frequencies, and S is the signal power, using (9) and (10), performance in Gaussian noise is, therefore,

$$P_e = \frac{1}{2}\, e^{-S/2} \, . \tag{12}$$

For Class A noise, using (4) and (10),

$$P_e = \frac{1}{2}\, e^{-A} \sum_{m=0}^{\infty} \frac{A^m}{m!}\, e^{-S/2\sigma_m^2} \, . \tag{13}$$

The above, of course, is for the binary symmetric channel. That is, $S_1(t)$ and $S_2(t)$ are equally probable. In short, the performance can be obtained by inspection from the normalized APD. If, for example, the probability that $R_0=1$ (0 dB) is exceeded is P_0, then, for the signaling set given by (11), p_e for a SNR of 2(3 db) is $P_0/2$, and so on, for any SNR. Note the 3 dB "shift" for NCFSK. No algorithm is given in the Appendix for NCFSK, since all that is required is a normalized APD, and the normalization procedure is included in other system performance algorithms. Also, in any case, the APD measurement device, DM-4, would normally present the measurements in normalized form, although we do not make that assumption in this report in order to maintain as much generality as possible.

In the bi-phase, DCPSK (differentially coherent phase shift keying) system, the receiver compares the phase ϕ of a noisy signal with a reference phase $\bar{\phi}$, to decide whether the corresponding pure signal relative phase ψ was 0 or π ($\psi = 0$, corresponding to the signal $\sqrt{2S}\cos\omega_0 t$, is selected if $|\phi-\bar{\phi}|<\pi/2$, and $\psi=\pi$, corresponding to $-\sqrt{2S}\cos(\omega_0 t)$, otherwise.) The reference phase is obtained from the previously received signals; usually it is just the phase of the previous signal. Thus the analysis of this system is complicated by the fact that both

ϕ and $\bar{\phi}$ are affected by noise. This system also has adjacent symbol dependency, and, therefore, the occurrence of paired errors and other error groupings cannot be obtained easily, even with independent noise. Halton and Spaulding (1966) have given results for this system, including the occurrence of various error groupings. However, it can be shown that for binary DCPSK, the elemental probability of error, P_e, is the same as for NCFSK, with 3 dB less signal energy required. That is, for a given P_e, DCPSK requires 3 dB less SNR than does NCFSK for arbitrary additive interference that is independent from one bit time to the next. [For a geometrical derivation of this result see Arthurs and Dym (1962).] For example, therefore, for Gaussian noise for binary DCPSK;

$$P_e = \frac{1}{2}\, e^{-S} . \tag{14}$$

The performance of DCPSK can be obtained directly from the APD of the additive interference. If, for example, the probability that $R_0 = 1$ (0 dB) is exceeded is P_0, then for the above signaling set, P_e for a SNR of 1 (0 dB), is $P_0/2$, and so on for any SNR. Figure 3 shows P_e versus SNR for the noise of figure 1 for both NCFSK and DCPSK, while Figure 4 shows P_e versus SNR for the Class B noise of Figure 2 for these two systems. Performance for Gaussian noise is also shown for reference.

We next consider coherent binary systems. The performance of these systems can be obtained from the pdf of the additive interference envelope by means of the result:

$$P_e = \frac{1}{\pi} \int_K^{\infty} p_R(r)\, \cos^{-1}\left(\frac{K}{r}\right) dr \, . \tag{15}$$

For the derivation of this result see Spaulding (1964), and for various other approaches which led to (15), see Arthurs and Dym (1962). For antipotal signaling (CPSK, coherent phase shift keying), the binary signal set is,

$$S_1(t) = \sqrt{2S}\cos(\omega_0 t), \text{ and}$$
$$S_2(t) = -\sqrt{2S}\cos(\omega_0 t), \tag{16}$$

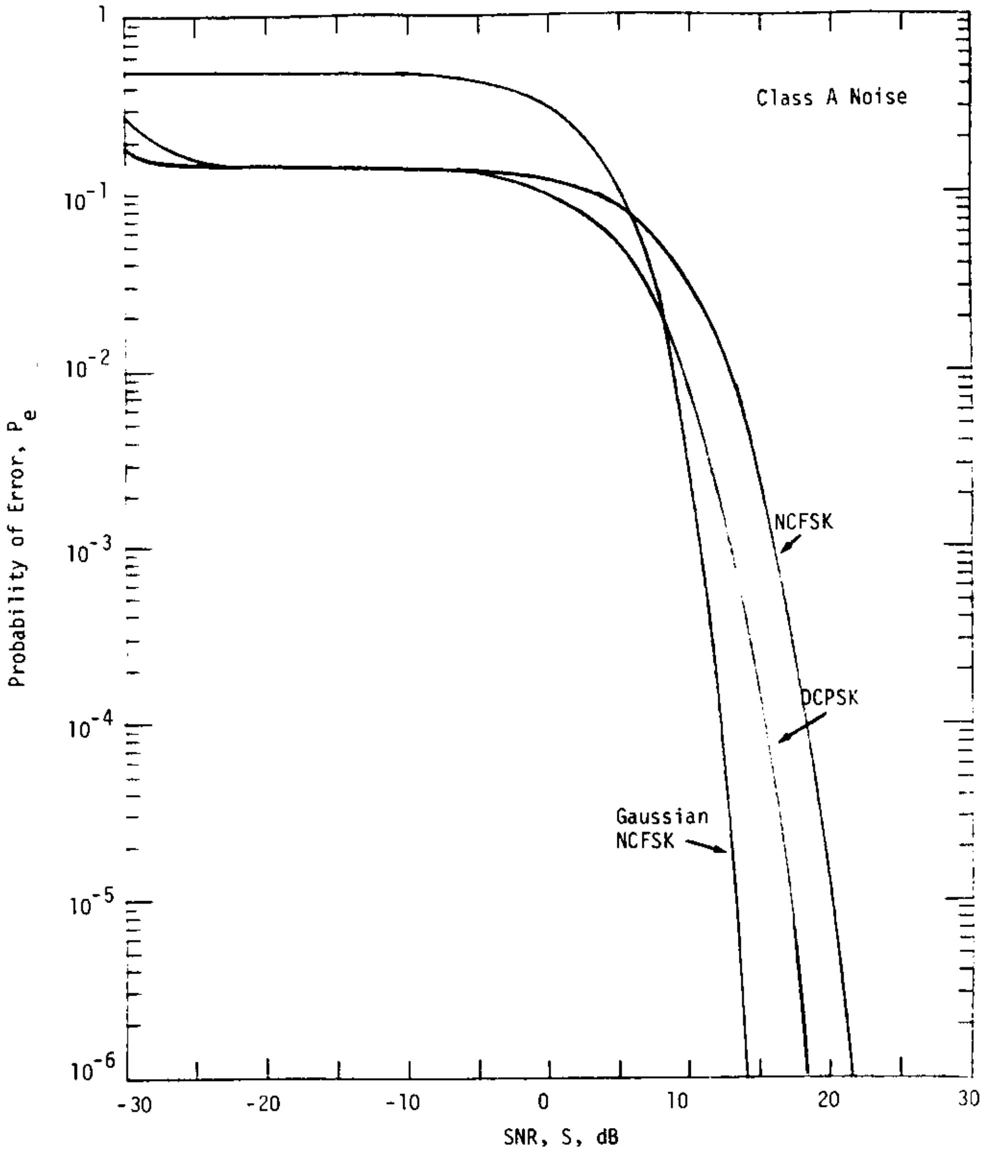

Figure 3. Systems performance for NCFSK and DCPSK for the Class A noise example of Figure 1.

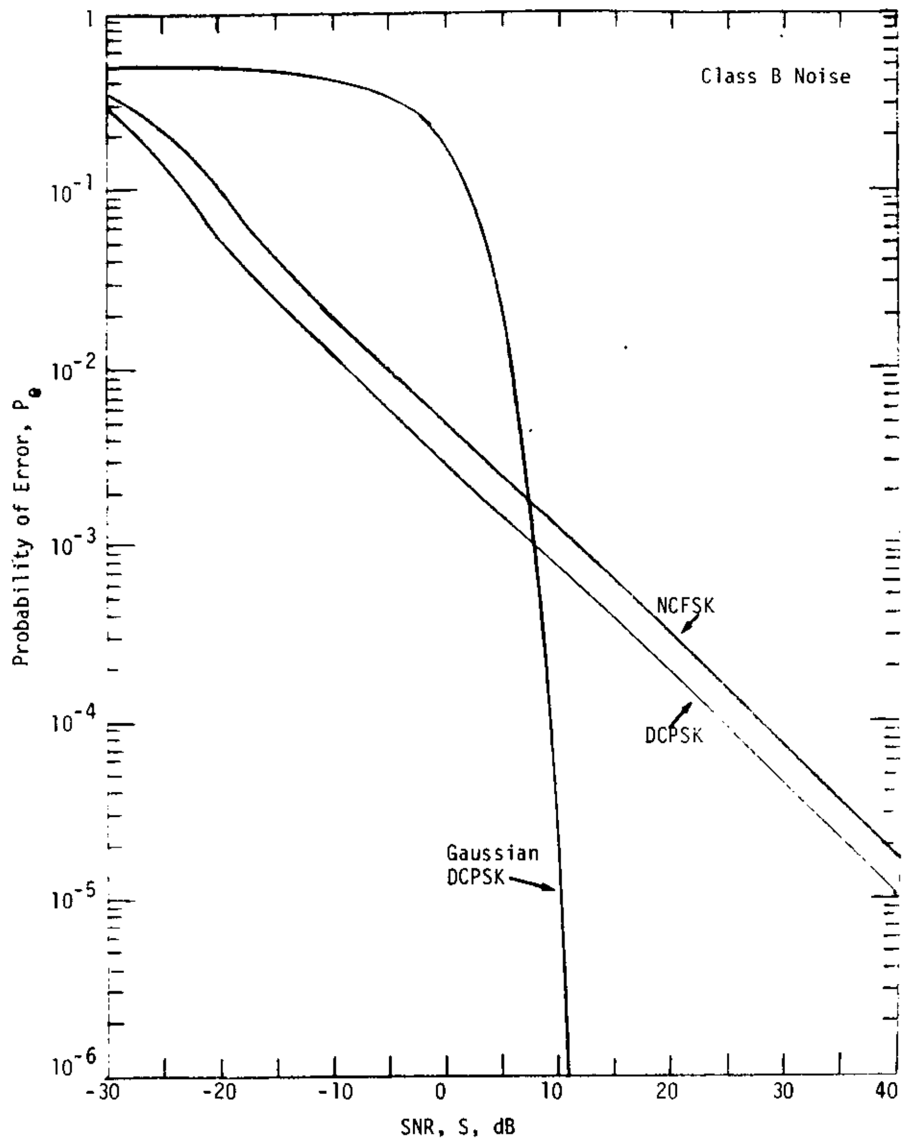

Figure 4. System performance for NCFSK and DCPSK for the Class B noise example of Figure 2.

and in (15), $K = \sqrt{S}$.

For coherent, orthogonal signaling, the signal set is,

$$S_1(t) = \sqrt{2S} \cos(\omega_0 t), \text{ and}$$
$$S_2(t) = \sqrt{2S} \sin(\omega_0 t) \quad , \tag{17}$$

and in (15), $K = \sqrt{S/2}$.

For ON-OFF coherent signaling,

$$S_1(t) = \sqrt{2S} \cos(\omega_0 t), \text{ and}$$
$$S_2(t) = 0 \quad , \tag{18}$$

and in (15), $K = \sqrt{S/4}$, where we use the convention that the SNR is based on the average signal power of the two signals, $S_1(t)$ and $S_2(t)$. This average is, of course, S/2, for a symmetric channel.

The performance of a coherent Neyman-Pearson signal detection system can also be obtained via the integral in (15). We have two hypotheses:

$$H_0: \; X(t) = Z(t) + S(t), \text{ and}$$
$$H_1: \; X(t) = Z(t). \tag{19}$$

The received waveform, X(t), is composed of noise plus the completely known signal to be detected (H_0), or it is composed of noise alone (H_1). The Neyman-Pearson detector, which is optimum if Z(t) is Gaussian, decides between H_0 and H_1 by presetting a probability of false alarm (deciding H_0, when H_1 is true) α. Performance is then given by the probability of detection (deciding H_0 when H_0 is true), P_D, and the probability of a miss (deciding H_1 when H_0 is true), P_M, and $P_D = 1 - P_M$. Performance for additive interference is given by (15), where

$$K = \sqrt{S} - \sqrt{2}\, \text{erfc}^{-1}(2\alpha), \tag{20}$$

for a desired signal of power S. The complimentary error function is given by

$$\text{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt. \tag{21}$$

Use of the K given in (20) in (15) gives the probability of a miss, P_M. This is

$$P_M = \frac{1}{\pi} \int_K^\infty p_R(r) \cos^{-1}\left(\frac{K}{r}\right) dr, \tag{22}$$

and $P_D = 1 - P_M$, K given by (20).

Table 2 below gives $\text{erfc}^{-1}(2\alpha)$ for various probabilities of false alarm, α.

Table 2. $\text{Erfc}^{-1}(2\alpha)$ for Various α

α	$\text{Erfc}^{-1}(2\alpha)$
10^{-2}	1.645
10^{-3}	2.185
10^{-4}	2.630
10^{-5}	3.015

For the above coherent systems in Gaussian noise,

$$P_e = \frac{1}{2} \text{erfc}(K), \tag{23}$$

where $K = \sqrt{S}$ for antipodal signaling, $K = \sqrt{S/2}$ orthogonal signaling, and $K = \sqrt{S/4}$ for ON-OFF signaling. For the signal detection system in Gaussian noise,

$$P_M = \frac{1}{2} \text{erfc}(K), \tag{24}$$

where $K = \sqrt{S} - \sqrt{2}\, \text{erfc}^{-1}(2\alpha)$.

Likewise, for Class A noise, we can obtain,

$$P_e = \frac{e^{-A}}{2} \sum_{m=0}^{\infty} \frac{A^m}{m!} \operatorname{erfc}(K/\sigma_m) \quad . \tag{25}$$

Equation (25) gives the P_m when the K given by (20) is used.

It now remains to develop efficient computer algorithms based on (15). The result (15) uses the pdf of the interference envelope, and the measurements are of the APD. Actually, the measurements at 31 levels of the APD also give an equally valid estimate of the pdf as well. Also if we attempt to modify (15) to a form that uses the APD directly, i.e., uses $P_R(r)$ rather than $p_R(r)$, we obtain computational complexities. For example, (15) can be transformed to:

$$P_e = \frac{1}{\pi}\int_K^{\infty} P_R(r) \frac{K}{r^2\sqrt{1-K^2/r^2}} \, dr, \tag{26}$$

or

$$P_e = \frac{1}{\pi}\int_0^{1} P_R\left(\frac{K}{r}\right) \frac{1}{\sqrt{1-r^2}} \, dr. \tag{27}$$

Both (26) and (27) are improper integrals, and while this creates no problem analytically, very sophisticated numerical integration routines are required in order to obtain any accuracy for P_e, especially when $P_R(r)$ is given only in sampled data form. It turns out that it is much better to use (15) "directly" along with $p_R(r)$ estimated from the measured $P_R(r)$.

The main algorithm presented in the Appendix is called SYSAPD. This program takes the 31 measured APD data points, normalizes the APD to its rms level, obtains the pdf, and then evaluates the integral (15) for the appropriate K. The program SYSAPD uses Gauss-Laguerre quadratures to evaluate (15) (Kopal, 1961). The Gauss-Laguerre quadrature formula is

$$\int_0^{\infty} e^{-x} f(x)dx \quad \sum_{j=1}^{n} H_j f(z_j) \quad , \tag{28}$$

the points at which the integrand must be evaluated, y_j, and the corresponding weight, H_j, are obtained via the Laguerre polynominals. The program SYSAPD used a fifteenth order quadrature [(n=15 in (28)]. The above means that the integral (15) is put in the form

$$P_e = \frac{1}{\pi}\int_0^{\infty} \left[p_R(y+K) \, e^y \, \operatorname{Cos}^{-1}\left(\frac{K}{y+K}\right)\right] e^{-y} dy \quad , \tag{29}$$

for evaluation.

Consider first the Class A "measurement" data of Table 1. Table 3 gives P_e versus SNR for CPSK obtained from the program SYSAPD which uses (29). Note that in using the data (see program listing in the Appendix) arbitrary 3 dB levels are used. For Class A noise, (25) gives the "correct" theoretical performance. The program SYSCOR computes (25) and Table 3 also gives these results so that the approximation from the "measurements" can be compared with the "true" answer. Another program that is given in the Appendix is SYSGL. This program uses (29), but $p_R(r)$ is obtained from the Class A model mathematical expression (4) rather than from the corresponding "measurement" data. The P_e versus SNR for CPSK from this program is also given on Table 3. This shows the accuracy of the integration routine when these results are compared with the "true" results. It also indicates the accuracy of the normalization, pdf determination, and interpolation techniques used in SYSAPD. Finally, Figure 5 shows the results of Table 3 along with the standard performance in Gaussian noise (23) for further comparison.

The above results are for the Class A example. For the Class B case, the simple Gauss-Laguerre quadrature used above does not give sufficient accuracy when the Class B "measurements" are used in the program SYSAPD or when the corresponding mathematical model for Class B noise is used with program SYSGL. [The result of using the Class B example in SYSAPD (or in SYSGL) is shown by the dashed curve on Figure 6.] Because of this, a different integration routine must be used. This is given by program SYSWR, which used Weddle's Rule (Kopal, 1961) to perform the integrations. This integration routine uses (15) directly and, of course, is somewhat more sophisticated than the Gauss-Laguerre quadrature used previously, but it is still appropriate for small scale computers. For the Class B case, we have no "theoretical" results to use to check the accuracy of the integrations performed by SYSWP.

Table 3. CPSK System Performance for Class A Noise

SNR (dB)	P_e, SYSCOR	P_e, SYSGL	P_e, SYSAPD
-3.00000E+01	1.60710E-01	1.44710E-01	1.39511E-01
-2.75000E+01	1.46402E-01	1.43726E-01	1.38669E-01
-2.50000E+01	1.42528E-01	1.42416E-01	1.37559E-01
-2.25000E+01	1.40644E-01	1.40669E-01	1.36052E-01
-2.00000E+01	1.38309E-01	1.38342E-01	1.34258E-01
-1.75000E+01	1.35202E-01	1.35244E-01	1.30981E-01
-1.50000E+01	1.31073E-01	1.31125E-01	1.26556E-01
-1.25000E+01	1.25602E-01	1.25664E-01	1.20784E-01
-1.00000E+01	1.18386E-01	1.19458E-01	1.13917E-01
-7.50000E+00	1.08950E-01	1.09034E-01	1.04749E-01
-5.00000E+00	9.67394E-02	9.68926E-02	9.24013E-02
-2.50000E+00	8.15682E-02	8.16687E-02	7.69463E-02
0.	6.33627E-02	6.34648E-02	5.82915E-02
2.50000E+00	4.33409E-02	4.34341E-02	4.15365E-02
5.00000E+00	2.42702E-02	2.43404E-02	2.21766E-02
7.50000E+00	9.99877E-03	1.00367E-02	9.42682E-03
1.00000E+01	2.69063E-03	2.70273E-03	2.68102E-03
1.25000E+01	4.46870E-04	4.48876E-04	4.70352E-04
1.50000E+01	4.05121E-05	4.07002E-05	4.63872E-05
1.75000E+01	1.39231E-06	1.39909E-06	2.07974E-06
2.00000E+01	1.32516E-08	1.33143E-08	5.27832E-09
2.25000E+01		2.09713E-11	0.

Table 4. CSPK System Performance for Class B Noise

SNR (dB)	P_e, SYSWR Model	P_e, SYSWR Data
-30.0	1.69496E-01	1.73082E-01
-25.0	7.04404E-02	7.50442E-02
-20.0	2.90375E-02	3.07851E-02
-15.0	1.36975E-02	1.43575E-02
-10.0	6.72071E-03	7.02565E-03
-5.0	3.34009E-03	3.49568E-03
0.0	1.66805E-03	1.73837E-03
5.0	8.34675E-04	8.67205E-04
10.0	4.18022E-04	4.33207E-04
15.0	2.09434E-04	2.15463E-04
20.0	1.04948E-04	1.06017E-04
25.0	5.25944E-05	5.07207E-05
30.0	2.63586E-05	2.44030E-05
35.0	1.32103E-05	1.05207E-05
40.0	6.62078E-06	3.62328E-06

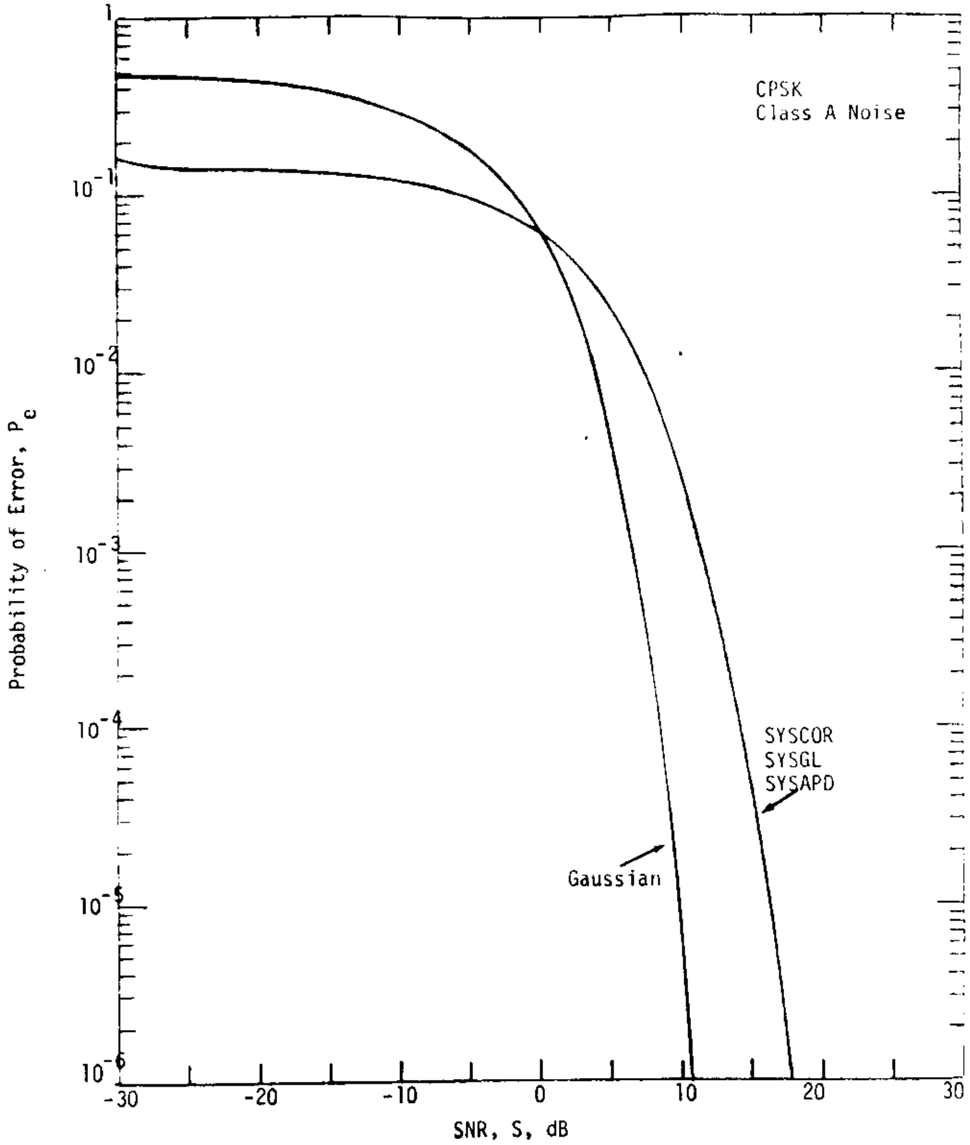

Figure 5. System performance for CPSK for the Class A noise example of Figure 1.

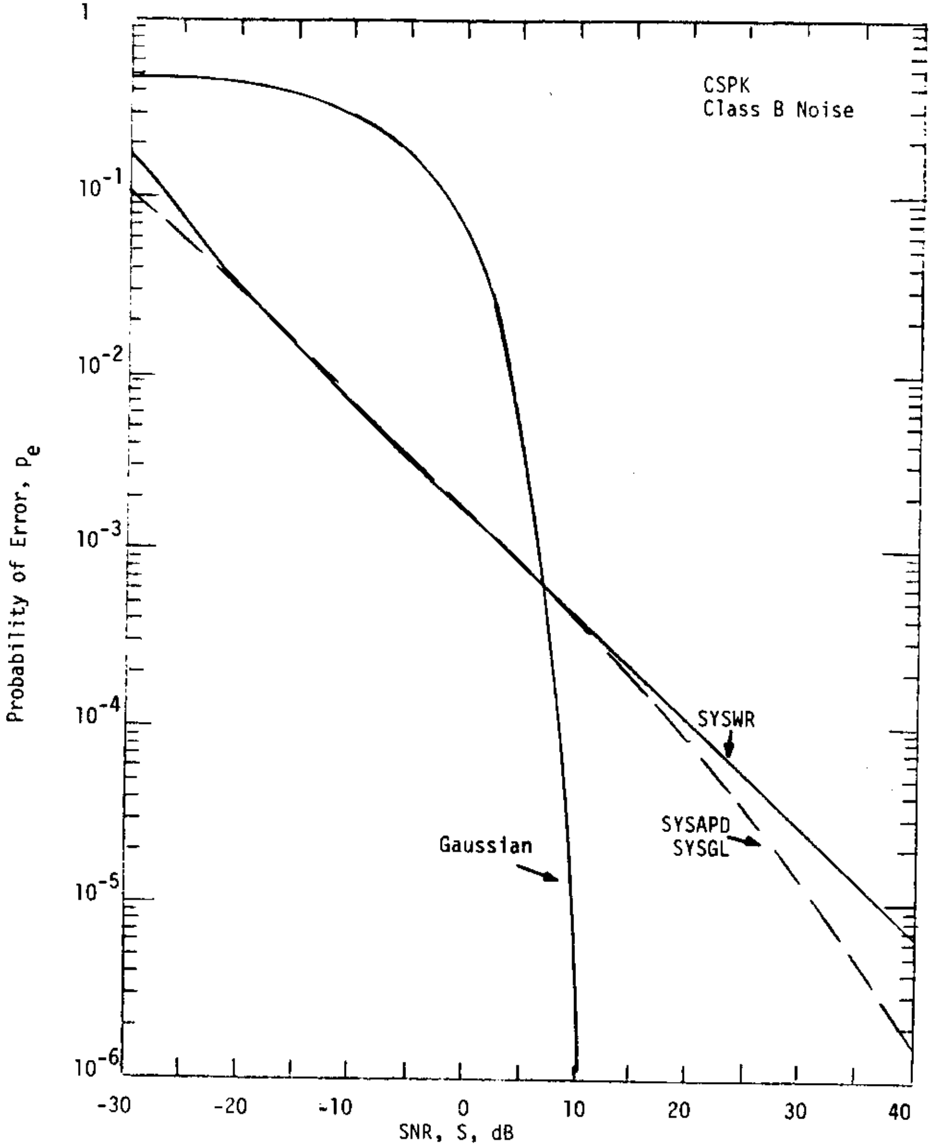

Figure 6. System performance for the Class B noise example of Figure 2.

The accuracy was checked by using another integration routine, appropriate only for large computers, in which the desired accuracy can be specified in conjunction with the Class B mathematical model. Program SYSWP was found to give very good accuracy for all signal-to-noise ratios. Table 4 and Figure 6 show the results of the use of SYSWR with the "measurements" of Table 1. Two subroutines, both termed FUN1, are given. One for the "measurement" data and program SYSWR and one for the mathematical Class B model for use with SYSWR.

In order to evaluate the integral (15) the pdf of the noise envelope is very easily obtained from the APD for Class A noise given in (4). However, obtaining the pdf for the envelope of Class B noise corresponding to the APD given by (6) is somewhat more involved. By differentiating (6) we eventually obtain the following, which has been put in a form suitable for numerical computation:

$$P_R(r) = \frac{2r}{\Omega} e^{-r^2/\Omega} \left\{ \sum_{n=0}^{\infty} \frac{(-1)^n A_\alpha^n}{n!} \Gamma\left(1 + \frac{n\alpha}{2}\right) \left[{}_1F_1\left(1 - \frac{n\alpha}{2}\,;\, 2;\, \frac{r^2}{\Omega}\right) - \frac{r^2}{2\Omega}\left(1 + \frac{n\alpha}{2}\right) {}_1F_1\left(1 - \frac{n\alpha}{2}\,;\, 3;\, \frac{r^2}{\Omega}\right)\right]\right\} . \tag{30}$$

The Appendix lists the appropriate programs and all the required subroutines used in the above example calculations.

4. CONCLUSIONS

This report has developed simple computer algorithms which use measurements of the APD of an interfering waveform (or a corresponding mathematical model) to determine performance of various "normal" digital data systems. As can be seen from the examples above, we obtain very good estimates of system performance using SYSAPD and DM-4 Class A simulated noise measurements and by using SYSWR and DM-4 Class B simulated noise measurements. Of course, the Class A measurements can also be used with SYSWR. The algorithms developed are for binary digital systems (and the coherent Neyman-Person signal detection system), however, the performance of other systems (e.g., M level systems, and minimal shift keying systems) can usually be obtained by appropriate extensions of the techniques developed here.

5. REFERENCES

Abramowitz, M., and I.A. Stegun (1964), Handbook of mathematical Functions, National Bureau of Standards Applied Mathematics Series, No. 55, Section 13, 503. (U.S. Government Printing Office, Washington, D.C. 20402).

Arthurs, E., and H. Dym (1962), On the optimum detection of digital signals in the presence of white Gaussian noise: a geometrical interpretation and a study of three basic data transmission systems, IRE Trans. Commun. Sys. CS-10, No. 4, pp. 336-372.

Evans, T.T., and A.S. Griffiths (1974), Design of a sanguine noise processor based upon world-wide extremely low frequency (ELF) recordings, IEEE Trans. Commun. COM-22, No. 4, pp. 528-539.

Halton, J.H., and A.D. Spaulding (1966), Error rates in differentially coherent phase systems in non-Gaussian noise, IEEE Trans. Commun. Techrol. COM-14, No. 5, pp. 594-601.

Kopal, Z. (1961), Numerical Analysis, (John Wiley and Sons, New York, New York).

Middleton, D. (1974), Statistical-physical models of man-made radio noise, Part I: First order probability models of the instantaneous amplitude, Office of Telecommunications Report OT 74-36 (NTIA Access. No. COM75-10864PAS).

Middleton, D. (1976), Statistical-physical models of man-made and natural radio noise, Part II: First-order probability models of the envelope and phase, Office of Telecommunications Report OT 76-86 (NTIS Access. No. PB253-949).

Middleton, D. (1977), Statistical-physical models of electromagnetic interference, IEEE Trans EMC EMC-19, No. 3, pp. 106-127.

Middleton, D. (1978a), Statistical-physical models of man-made and natural radio noise, Part III: First-order probability models of the instantaneous amplitude of Class B interference, National Telecommunications and Information Administration Contractor Report 78-1 (NTIS Access. No. PB284862/AS).

Middleton, D. (1978b), Statistical-physical models of man-made and natural radio noise, Part IV: Determination of the first-order parameters of Class A and Class B interference, National Telecommunications and Information Administration Contractor Report 78-2 (NTIS Access. No. PB288467).

Middleton, D. (1979), Canonical non-Gaussian noise models: Their implications for measurements and for prediction of receiver performance, IEEE Trans. EMC EMC-21, No. 3, pp, 209-220.

Middleton, D. (1980), A tutorial review of the new EMI models and their effects on receiver performance, National Telecommunications and Information Administration Contractor Report 80-7 (NTIS Access. No. PB80-211188).

Montgomery, G.F. (1954), Comparison on amplitude and angle modulation for narrowband communications of binary-coded messages in fluctuation noise, Proc. IRE 42, pp. 447-454.

Spaulding, A.D. (1964), Determination of error rates for narrow-band communication of binary-coded messages in atmospheric radio noise, Proc. IEEE 52, No. 2, pp. 220-221.

Spaulding, A.D. (1977), Stochastic modeling of the electromagnetic interference environment, Conference Record - International Conference on Communications, ICC '77, Chicago, 42.2-114-123 (IEEE Catalog No. 77CH 1209-6C SCB).

Spaulding, A.D., and D. Middleton (1977), Optimum reception in an impulsive interference environment - Part I: Coherent detection; Part II: Incoherent reception, IEEE Trans. Commun. COM-25, pp. 910-934.

White, H.E. (1966), Atmospheric noise FSK error probabilities for an envelope - detector receiver, IEEE Trans. Commun. Technol. COM-14, No. 2, pp. 288-289.

APPENDIX

Program Listings

In this Appendix we simply list the computer programs (in Fortran 4) used for the sample calculations given in the report and required for similar calculations. The programs are essentially self-explanatory via the comment statements, but some further explanation may be helpful.

The first two programs, APDA and APDB, compute the APD "measurement data" from the Middleton Class A and Class B models. The output is given in Table 1. The program APDB requires Subroutine CONHYP and FUNCTION GAMMA which are also given.

The next set of programs are for the Class A example. The program SYSCOR computes the theoretical performance and requires FUNCTION CERF. The program SYSGL computes performance via Gauss-Laguerre integration but obtains the required pdf of the envelope values directly from the Class A model and requires SUBROUTINE FUN. Finally, one of the main results is program SYSAPD which computes performance from measured APD values. Sections of this program normalize the APD and estimate the pdf from the measured APD. These routines are useful in their own right for various purposes. The outputs of these programs are contained in Table 3.

The final set of programs are for the Class B example, but can also be used for any noise example. The program SYSWR computes performance via Weddle's Rule integration. Two FUN1 subroutines are given for use with SYSWR. One calculates the envelope pdf from the mathematical model given by (30) and the other one calculates the envelope pdf from the measured APD. The outputs of SYSWR are given in Table 4.

In some of the programs IRAY and SYSTEMC are used. This is to suppress an exponent underflow error message for the particular computer used (CYBER 170/750) and is not, in general, required.

```
      PROGRAM APDA(INPUT,OUTPUT)
C     PROGRAM USED TO OBTAIN APD VALUES FROM CLASS A MODEL.
C     EQUATION 4 OF TEXT.
      DIMENSION IRAY(6)
      DATA IRAY/-1,-1,-1,0,-1,-1/
      CALL SYSTEMC(115,IRAY)
      PRINT 6
    6 FORMAT(1H1)
      A=0.35
      GAM=0.5E-3
      DO 40 I=1,31
      RDB=-59.+3.*(I-1.)
      R=10.**(RDB/20.)
      SM=0. $ FJ=1.
      DO 20 JJ=1,26
      J=JJ-1
      IF(J.NE.0) FJ=FJ*J
      SIGSQ=(J/A+GAM)/(1.+GAM)
      T=((A**J)FJ)*EXP(-R*R/SIGSQ)
      SM=SM+T
   20 CONTINUE
      P=SM*EXP(-A)
      IF(P.LT.1.0E-6) GO TO 10
      GO TO 15
   10 P=0.
   15 PRINT 7,RDB,P
   40 CONTINUE
    7 FORMAT(10X,2(1PE12.5,3X))
      END
```

```
      PROGRAM APDB(INPUT,OUTPUT)
C     PROGRAM USED TO OBTAIN APD VALUES FROM CLASS B MODEL.
C     EQUATION 6 OF TEXT.
      DIMENSION IRAY(6)
      DATA IRAY/-1,-1,-1,0,-1,-1/
      CALL SYSTEMC(115,IRAY)
      PRINT 6
    6 FORMAT(1H1)
      AA=1.
      ALPHA=1.2
      OMEGA=0.00079433
      DO 40 I=1,31
      RDB=-40.+3.*(I-1.)
      IF(RDB.GT.45.) GO TO 10
      R=10.**(RDB/20.)
      ZN=R*R/OMEGA
      SM=0. $ FN=1. $ SM1=0.
      DO 20 N=1,25
      FN=FN*N
      CALL CONHYP(1.-N*ALPHA/2.,2.,ZN,S,IOVFLW)
      T=(((-AA)**N)/FN)*GAMMA(1.+N*ALPHA/2.)*S
      IF(IOVFLW.NE.1) GO TO 14
      SM1=SM1+T
      GO TO 20
   14 SM=SM+T
   20 CONTINUE
      FP=0.
      IF(ZN.LT.675.) FP=EXP(-ZN)
      P=FP-ZN*(FP*SM+SM1)
      GO TO 15
   10 P=0.
   15 PRINT 7, RDB,P
   40 CONTINUE
    7 FORMAT(10X,2(1PE12.5,3X))
      END
```



```
      SUBROUTINE CONHYP(A,B,X,S,IOVFLW)
C.....COMPUTES 1F1(A,B,X) FOR REAL A,B,X
C.....IF X GREATER THAN 741. AN OVERFLOW WILL OCCUR. SEE
C.....COMMENTS BELOW.
      S=1.  $ Y=1.
      IOVFLW=0
      KUNDEF=0
      IF(A.GT.0.)GO TO 101
      K=-A
      ENA=-K-1
      VA=A-ENA
      IF(VA.EQ.1..OR.VA.EQ.0.)GO TO 110
  101 IF(B.GT.0.)GO TO 130
      J=-B
      ENB=-J-1
      VB=B-ENB
      IF(VB.EQ.1..OR.VB.EQ.0.)120,130
  110 KUNDEF=1
      GO TO 101
  120 IF(KUNDEF.EQ.1)PRINT1000,A,B
      IF(KUNDEF.NE.1)PRINT1001,B
      RETURN
  130 IF(KUNDEF.EQ.1)GO TO 10
    5 IF(X.GE.100.) GO TO 60
    6 IF(X.GE.10.) GO TO 10
      NN=100
      GO TO 15
   10 NN=300
   15 IF(KUNDEF.EQ.1) NN=-A+1
      DO 20 N=1,NN
      D=N*((B+N-1.0)**2.)
      Y=(A+N-1.0)*(Y/D)
      Y=Y*(B+N-1.0)
      Y=Y*X
      IF(S.EQ.(S+Y))GO TO 50
      S=S+Y
   20 CONTINUE
   50 RETURN
C.....APPROXIMATES 1F1(A,B,X) FOR REAL A,B,X BY USING THE
C.....ASYMPTOTIC EXPANSION. SEE PAGE 1073, INTRODUCTION
C.....TO STATISTICAL COMMUNICATIONS THEORY, MIDDLETON.
C.....IF X.GE.675. AN OVERFLOW WILL OCCUR FROM EXP.
C.....TO AVOID THIS, THE VARIABLE IOVFLW IS SET TO 1 AND
C.....THE FUNCTION VALUE IS CALCULATED WITHOUT THE EXP(X) FACTOR.
C.....SO THAT THE VALUE RETURNED IS S/EXP(X)
   60 NN=20
      DO 100 N=1,NN
      Y=Y*(B-A+N-1.)*(N-A)
      Y=Y/(N*X)
      IF(S.EQ.(S+Y))GO TO 150
      S=S+Y
  100 CONTINUE
  150 S=S*(GAMMA(B)/GAMMA(A))*(X**(A-B))
      IF(X.LT.675.)GO TO 190
      IOVFLW=1
      GO TO 200
  190 S=S*EXP(X)
```

```
 200 RETURN
1000 FORMAT(//,1X,* CANNOT EVALUATE EXPRESSION SINCE BOTH*,
    1* A AND B ARE NEGATIVE INTEGERS OR ZERO, A=*,F10.2,* , B=*,
    2F10.2,//)
1001 FORMAT(//,1X,* BAD VALUE FOR B GIVES INFINITE RESULT FOR S*,
    1 *, B=*,F10.2,//)
     END
```

```
      FUNCTION GAMMA(X)
C     RETURNS THE GAMMA FUNCTION FOR REAL ARGUMENT.
C     NOTE. THE GAMMA FUNCTION IS NOT DEFINED FOR A NEGATIVE INTEGER OR ZER
C     INPUT
C        X = THE REAL ARGUMENT.
C     OUTPUT.
C        GAMMA(X) = THE GAMMA FUNCTION OF ARGUMENT X.
   75 FORMAT(66H GAMMA FUNCTION OF A NEGATIVE INTEGER, OR OF ZERO, IS NO
     1T DEFINED.)
    5 IF(X) 10,80,15
   10 N=-X
      EN=-N-1
      V=X-EN
      IF(V.EQ.1.)80,20
   15 N=X
      EN=N
      V=X-EN
   20 GAMMA=1.+V*(.422784337+V*(.4118402518+V*(.08157821878+V*
     1(.07423790761+V*(-.0002109074673+V*(.01097369584+V*(-.002466747981
     2+V*(.001539768105-V*(.0003442342046-V*.00006771057117))))))))
      IF(EN-2.) 37,25,30
   25 RETURN
   30 N=N-1
      DO 35 I=2,N
      FI=I
   35 GAMMA=GAMMA*(FI+V)
      RETURN
   37 N=2.-EN
      DO 40 I=1,N
      FI=2-I
   40 GAMMA=GAMMA/(FI+V)
      RETURN
   80 PRINT 75
      CALL EXIT
      END
```



```
      PROGRAM SYSCOR(INPUT,OUTPUT)
C     THIS PROGRAM COMPUTES THE PROBABILITY OF ERROR FOR BINARY
C     CPSK, CFSK, AND COHERENT ON-OFF IN CLASS A NOISE, THAT IS,
C     MIDDLETON"S CLASS A MODEL.
C     HERE, THE ERFC FUNCTION IS TERMED CERF, TO BYPASS THE
C     SYSTEMS INTERNAL ERFC ROUTINE.
C     FOR CPSK, AK=SQRT(S)
C     FOR CFSK, AK=SQRT(S/2.)
C     FOR ON-OFF, AK=SQRT(S/4.)
      A=0.35
      GAM=0.5E-3
      PRINT 6
    6 FORMAT(1H1)
      DO 40 J=1,29
      SDB=-30.+2.5*(J-1)
      S=10.**(SDB/10.)
      AK=SQRT(S)
      SUM=0. $ FK=1.
      DO 20 KK=1,26
      K=KK-1
      IF(K.NE.0) FK=FK*K
      SIGSQ=(K/A+GAM)/(1.+GAM)
      SIG=SQRT(SIGSQ)
      T=((A**K)/FK)*CERF(AK/SIG)
      SUM=SUM+T
   20 CONTINUE
      PE=EXP(-A)*SUM/2.
      IF(PE.LT.1.E-9) CALL EXIT
      PRINT 8, SDB,PE
   40 CONTINUE
    8 FORMAT(10X,2(1PE12.5,3X))
      END
```

```
      FUNCTION CERF(X)
C     SEE APPROXIMATIONS FOR DIGITAL COMPUTERS
C     BY C. HASTINGS, PRINCETON U. PRESS, 1955,
C     PAGE 169. ALSO IN ABRAMOWITZ AND STEGUN.
      E=1.0/(1.0+0.3275911*X)
      S=(((((0.940646070*E)-1.287822453)*E+1.259695130)*E-0.252128668)*E
     1+0.225836846)*E
      XSQ=X**2
      EXPFX=0.0
      IF(XSQ.LT.709.0)EXPFX=EXP(-XSQ)
      CERF=S*EXPFX*1.128379167
      RETURN
      END
```

```
      PROGRAM SYSGL(INPUT,OUTPUT)
C     THIS PROGRAM USES GAUSS-LAQUERRE QUADRATURES ALONG WITH
C     CLASS A NOISE MODEL TO COMPUTE SYSTEM PERFORMANCE.
C     SEE PROGRAM SYSAPD FOR FURTHER DETAILS.
      DIMENSION IRAY(6),Z(15),H(15)
      DATA IRAY/-1,-1,-1,0,-1,-1/
      DATA Z/0.09330781,0.49269174,1.21559541,2.26994952,3.66762272,
     15.42533663,7.56591623,10.12022857,13.13028248,16.65440771,
     220.77647890,25.62389423,31.40751917,38.53068331,48.02608557/
      DATA H/.21823489,.34221018,.26302758,.12642582,.40206865E-1,
     1.85638778E-2,.12124361E-2,.11167439E-3,.64599268E-5,.22263169E-6,
     2.42274304E-8,.39218973E-10,.14565153E-12,.14830271E-15,
     3.16005949E-19/
      CALL SYSTEMC(115,IRAY)
      PRINT 6
    6 FORMAT(1H1)
      DO 50 N=1,29
      SDB=-30.+2.5*(N-1)
      S=10.**(SDB/10.)
      AK=S**0.5
      SUM=0.
      DO 30 K=1,15
      CALL FUN(Z(K),AK,F)
      SUM=SUM+F*H(K)
   30 CONTINUE
      PE=SUM/3.141592654
      PRINT8, SDB,PE
      IF(PE.LT.1.E-9) CALL EXIT
   50 CONTINUE
    8 FORMAT(10X,2(1PE12.5,3X))
      END


      SUBROUTINE FUN(X,P,Y)
C     THIS FUNCTION ROUTINE IS FOR CLASS A NOISE FOR USE
C     WITH PROGRAM SYSGL.
      A=0.35
      GAM=0.5E-3
      V=X+P
      SM=0. $ FJ=1.
      DO 20 JJ=1,26
      J=JJ-1
      IF(J.NE.0) FJ=FJ*J
      SIGSQ=(J/A+GAM)/(1.+GAM)
      T=(2.*V/SIGSQ)*((A**J)/FJ)*EXP(-V*V/SIGSQ)
      SM=SM+T
   20 CONTINUE
      Y=SM*EXP(X)*ACOS(P/V)*EXP(-A)
      RETURN
      END
```



```
      PROGRAM SYSAPD(INPUT,OUTPUT)
C     THIS PROGRAM MAKES DIRECT USE OF THE MEASURED APD DATA,31 LEVELS,
C     AND ESTIMATES THE PDF OF THE ENVELOPE FOR USE IN THE GENERAL
C     COHERENT SYSTEM PERFORMANCE ALGORITHM. THE PROB OF BIT ERROR
C     FOR BINARY CPSK,CFSK,AND COHERENT ON-OFF IS ESTIMATED. THE
C     SIGNAL POWER IS GIVEN BY S, AND THE NOISE IS
C     NORMALIZED SO THAT THE MEAN NOISE POWER IS UNITY. THEN
C     THE SIGNAL-TO-NOISE RATIO IS ALSO GIVEN BY S. SIGNAL-TO-NOISE
C     RATIOS FROM 40DB TO -30DB ARE COVERED IN 2.5DB STEPS. THE
C     PROGRAM USES GAUSS-LAQUERRE QUADRATURES TO EVALATE THE INTEGRALS.
C     FOR CPSK, AK=SQRT(S)
C     FOR CFSK, AK=SQRT(S/2.)
C     FOR OFF-ON, AK=SQRT(S/4.)
C     NOTE, BEFORE THIS PROGRAM IS EFFECTIVE, THE ENTIRE APD MUST
C     BE COVERED BY THE 31 (OR LESS) LEVELS. THAT IS, P SHOULD
C     RANGE FROM ABOUT 0.95 OR HIGHER DOWN TO ABOUT 1.0E-5 OR SMALLER.
      DIMENSION CL(31),SL(31),P(31),CX(31),SX(31),SP(31)
      DIMENSION Z(15),H(15)
      DATA CL/1.,4.,7.,10.,13.,16.,19.,22.,25.,28.,31.,34.,37.,
     140.,43.,46.,49.,52.,55.,58.,61.,64.,67.,70.,73.,76.,79.,
     282.,85.,88.,91./
      DATA P/.9982,.9965,.9930,.9860,.9724,.9460,.8964,.8085,.6696,
     1.4946,.3519,.2997,.2949,.2944,.2934,.2916,.2879,.2807,.2669,
     2.2415,.1981,.1343,.06372,.01670,.002116,.0001047,.000001072,
     3.0,.0,.0,.0/
C     CL IS THE APD LEVELS IN DB AND P IS THE PROBABILITIES AT
C     THESE LEVELS.
      DATA Z/0.09330781,0.49269174,1.21559541,2.26994952,3.66762272,
     15.42533663,7.56591623,10.12022857,13.13028248,16.65440771,
     220.77647890,25.62389423,31.40751917,38.53068331,48.02608557/
      DATA H/.21823489,.34221018,.26302758,.12642582,.40206865E-1,
     1.85638778E-2,.12124361E-2,.11167439E-3,.64599268E-5,.22263169E-6,
     2.42274304E-8,.39218973E-10,.14565153E-12,.14830271E-15,
     3.16005949E-19/
C     H AND Z ARE AS GIVEN IN EQ. 28 OF TEXT.
      CX(1)=1.5*CL(1)-0.5*CL(2)
      SX(1)=10.**(CX(1)/20.)
      DO 20 I=2,31
      CX(I)=(CL(I-1)+CL(I))/2.
      SX(I)=10.**(CX(I)/20.)
   20 CONTINUE
C     COMPUTE RMS
      SUM=SX(1)*SX(1)*(1.-P(1))
      DO 30 J=2,31
      T=SX(J)*SX(J)*(P(J-1)-P(J))
      SUM=SUM+T
   30 CONTINUE
      RMS=(SUM+10.**(CL(31)/20.)*P(31))**0.5
C     NORMALIZE TO RMS LEVEL AND COMPUTE PDF
      CL(1)=CL(1)-20.*ALOG10(RMS)
      SL(1)=10.**(CL(1)/20.)
      SX(1)=SX(1)/RMS
      CX(1)=20.*ALOG10(SX(1))
      SP(1)=(1.-P(1))/SL(1)
```

```
      DO 40 K=2,31
      CL(K)=CL(K)-20.*ALOG10(RMS)
      SL(K)=10.**(CL(K)/20.)
      SX(K)=SX(K)/RMS
      CX(K)=20.*ALOG10(SX(K))
      SP(K)=(P(K-1)-P(K))/(SL(K)-SL(K-1))
   40 CONTINUE
      PRINT 6
    6 FORMAT(1H1)
      DO 80 N=1,29
      SDB=-30.+2.5*(N-1)
      S=10.**(SDB/10.)
      AK=SQRT(S)
      SUM=0. $ MM=2
      DO 70 L=1,15
      V=Z(L)+AK
      IF(V.LT.SX(1).OR.V.GT.SX(31)) 15,16
   16 DO 17 M=MM,31
      IF(V.LE.SX(M).AND.V.GT.SX(M-1)) GO TO 18
   17 CONTINUE
   18 IF(SP(M-1).EQ.0..OR.SP(M).EQ.0.) GO TO 15
      YDB=(20.*ALOG10(V)-CX(M-1))/(CX(M)-CX(M-1))
      YDB=YDB*(20.*ALOG10(SP(M))-20.*ALOG10(SP(M-1)))
      YDB= YDB+20.*ALOG10(SP(M-1))
      Y=10.**(YDB/20.)
      MM=M
      GO TO 19
   15 Y=0.
   19 T=Y*EXP(Z(L))*ACOS(AK/V)
      SUM=SUM+T*H(L)
   70 CONTINUE
      PE=SUM/3.141592654
      PRINT 8, SDB,PE
      IF(PE.LT.1.E-9) CALL EXIT
   80 CONTINUE
    8 FORMAT(10X,2(1PE12.5,3X))
      END
```

```
      PROGRAM SYSWR(INPUT,OUTPUT)
C     THIS PROGRAM USES WEDDLES RULE FOR THE
C     INTEGRATIONS REQUIRED IN THE DETERMINATION OF
C     SYSTEM PERFORMANCE.
C     DIFFERENT FUN1 SUBROUTINES ARE USED FOR
C     DIFFERENT NOISE MODELS AND/OR NOISE MEASUREMENTS.
      DIMENSION Z(7)
      COMMON/QQQ/AA,ALPHA,OMEGA,AK
      PRINT 5
    5 FORMAT(1H1)
      AA=1.0
      ALPHA=1.[illegible]
      OMEGA=0.00079433
      AK=1.
      CALL FUN1(1.,ZZ)
      DO 60 J=1,15
      SDB=-30.+5.*(J-1)
      S=10.**(SDB/10.)
      AK=S**0.5
      SUM=0.
      DO 50 I=1,15
      AKDB=20.*ALOG10(AK)
      ADB=AKDB+4.*(I-1)
      BDB=AKDB+4.*I
      A=10.**(ADB/20.)
      B=10.**(BDB/20.)
      DX=(B-A)/6.
      DO 40 K=1,7
      X=A+(K-1)*DX
      CALL FUN2(X,Z(K))
   40 CONTINUE
      SS=0.3*DX*(Z(1)+5.*Z(2)+Z(3)+6.*Z(4)+Z(5)+5.*Z(6)+Z(7))
      IF(SS.EQ.0.)  GO TO 55
      SUM=SUM+SS
   50 CONTINUE
   55 PE=SUM/3.141592654
      PRINT 8, SDB, PE
      IF(PE.LT.1.E-7) CALL EXIT
   60 CONTINUE
    8 FORMAT(10X,F5.1,2X,1PE12.5)
      END
```

```
      SUBROUTINE FUN1(X,Y)
C     THIS SUBROUTINE IS FOR USE WITH PROGRAM SYSWR.
C     IT MAKES USE OF MEASURED APD DATA. SEE PROGRAM
C     SYSAPD FOR FURTHUR DETAILS.
      DIMENSION CL(31),SL(31),P(31),CX(31),SX(31),SP(31)
      COMMON/Q/AK
      DATA CL/1.,4.,7.,10.,13.,16.,19.,22.,25.,28.,31.,34.,37.,
     140.,43.,46.,49.,52.,55.,58.,61.,64.,67.,70.,73.,76.,79.,
     282.,85.,88.,91./
      DATA P/.9435,.8914,.7984,.6491,.4516,.2631,.1445,.08575,.05387,
     1.03468,.02259,.01481,.009736,.006414,.004230,.002792,.001843,
     2.001217,.0008041,.0005312,.0003509,.0002318,.0001532,.0001012,
     3.00006685,.00004417,.00002918,.00001928,.00001274,0.,0./
C     CL IS THE APD LEVELS IN DB AND P IS THE PROBABILITIES AT
C     THESE LEVELS.
      CX(1)=1.5*CL(1)-0.5*CL(2)
      SX(1)=10.**(CX(1)/20.)
      DO 20 I=2,31
      CX(I)=(CL(I-1)+CL(I))/2.
      SX(I)=10.**(CX(I)/20.)
   20 CONTINUE
C     COMPUTE RMS
      SUM=SX(1)*SX(1)*(1.-P(1))
      DO 30 J=2,31
      T=SX(J)*SX(J)*(P(J-1)-P(J))
      SUM=SUM+T
   30 CONTINUE
      RMS=(SUM+10.**(CL(31)/20.)*P(31))**0.5
C     NORMALIZE TO RMS LEVEL AND COMPUTE PDF
      CL(1)=CL(1)-20.*ALOG10(RMS)
      SL(1)=10.**(CL(1)/20.)
      SX(1)=SX(1)/RMS
      CX(1)=20.*ALOG10(SX(1))
      SP(1)=(1.-P(1))/SL(1)
      DO 40 K=2,31
      CL(K)=CL(K)-20.*ALOG10(RMS)
      SL(K)=10.**(CL(K)/20.)
      SX(K)=SX(K)/RMS
      CX(K)=20.*ALOG10(SX(K))
      SP(K)=(P(K-1)-P(K))/(SL(K)-SL(K-1))
   40 CONTINUE
      MM=M
      ENTRY FUN2
      V=X
      IF(V.LT.SX(1).OR.V.GT.SX(31)) 15,16
   16 DO 17 M=2,31
      IF(V.LE.SX(M).AND.V.GT.SX(M-1)) GO TO 18
   17 CONTINUE
   18 IF(SP(M-1).EQ.0..OR.SP(M).EQ.0.) GO TO 15
      YDB=(20.*ALOG10(V)-CX(M-1))/(CX(M)-CX(M-1))
      YDB=YDB*(20.*ALOG10(SP(M))-20.*ALOG10(SP(M-1)))
      YDB= YDB+20.*ALOG10(SP(M-1))
      PDF=10.**(YDB/20.)
      GO TO 19
   15 PDF=0.
   19 Y=PDF*ACOS(AK/V)
      RETURN
      END
```



```
      SUBROUTINE FUN1(X,Y)
      THIS FUNCTION ROUTINE IS FOR CLASS B NOISE FOR USE
      WITH PROGRAM SYSWR.
      COMMON/QQQ/AA,ALPHA,OMEGA,AK
      ENTRY FUN2
      V=X
      ZN=V*V/OMEGA
      SM=0. $ FN=1. $ SM1=0.
      DO 20 NN=1,26
      N=NN-1
      IF(N.NE.0) FN=FN*N
      CALL CONHYP(1.-N*ALPHA/2.,2.,ZN,S,IOVFLW)
      CALL CONHYP(1.-N*ALPHA/2.,3.,ZN,SS,IOVFLW)
      SSS=(ZN/2.)*(1.+N*ALPHA/2.)*SS
      T=(((-AA)**N)/FN)*GAMMA(1.+N*ALPHA/2.)*(S-SSS)
      IF(IOVFLW.NE.1) GO TO 14
      SM1=SM1+T
      GO TO 20
   14 SM=SM+T
   20 CONTINUE
      FP=0.
      IF(ZN.LT.675.) FP=EXP(-ZN)
      PDF=(2.*V/OMEGA)*(FP*SM+SM1)
      Y=PDF*ACOS(AK/V)
      RETURN
      END
```