Development of a Computer Program
to Simulate a Noncoherent FSK System
in the Presence of Multipath Fading

by

LOREN D. BAREISS, JR.

B.S., Kansas State University, 1977

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1979

Approved by

Donald R. Hummels

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ILLEGIBLE

THE FOLLOWING DOCUMENT (S) IS ILLEGIBLE DUE TO THE PRINTING ON THE ORIGINAL BEING CUT OFF

ILLEGIBLE

I. INTRODUCTION

In studying the problem of a noncoherent frequency shift keyed (FSK) system operating in the presence of multipath fading, one finds there is substantial literature available dealing with the subject.

One of the earliest and surely most important is that of Chadwick (1). In this paper, the results of the work by Glenn (2) and Boyd (3) on noncoherent FSK detection were used and their technique was applied to the problem of multipath interference. This technique relies on the sampling theorem to evaluate the output of an integrate and dump stage in the form of a summation. Variations on this method can be found in papers by Austin (4), Austin and Milstein (5) and Schuchman (6). All of these rely on the approximation of the postdetection filter by sampling and summing techniques.

Probably the most useful work can be found in a paper by Kwon and Shehadeh (7). Unlike the aforementioned authors, they have used the sinusoidal series expansion technique for the representation of a band-limited Gaussian process as developed by Yaglum (8).

Figure 1 shows a diagram of a typical system. The received signal has both direct and reflected components. The most general case must assume the receiver and/or the transmitter are mobile units. The relationship between the reflected and direct signals is thus a random process. Because the characteristics of the direct signal are known, those of the reflected signal are random variables. In addition, the noise in the channel can be assumed to be white Gaussian noise. Because of these factors, signal processing techniques, as well as communication theory, are basic to the study of the problem.

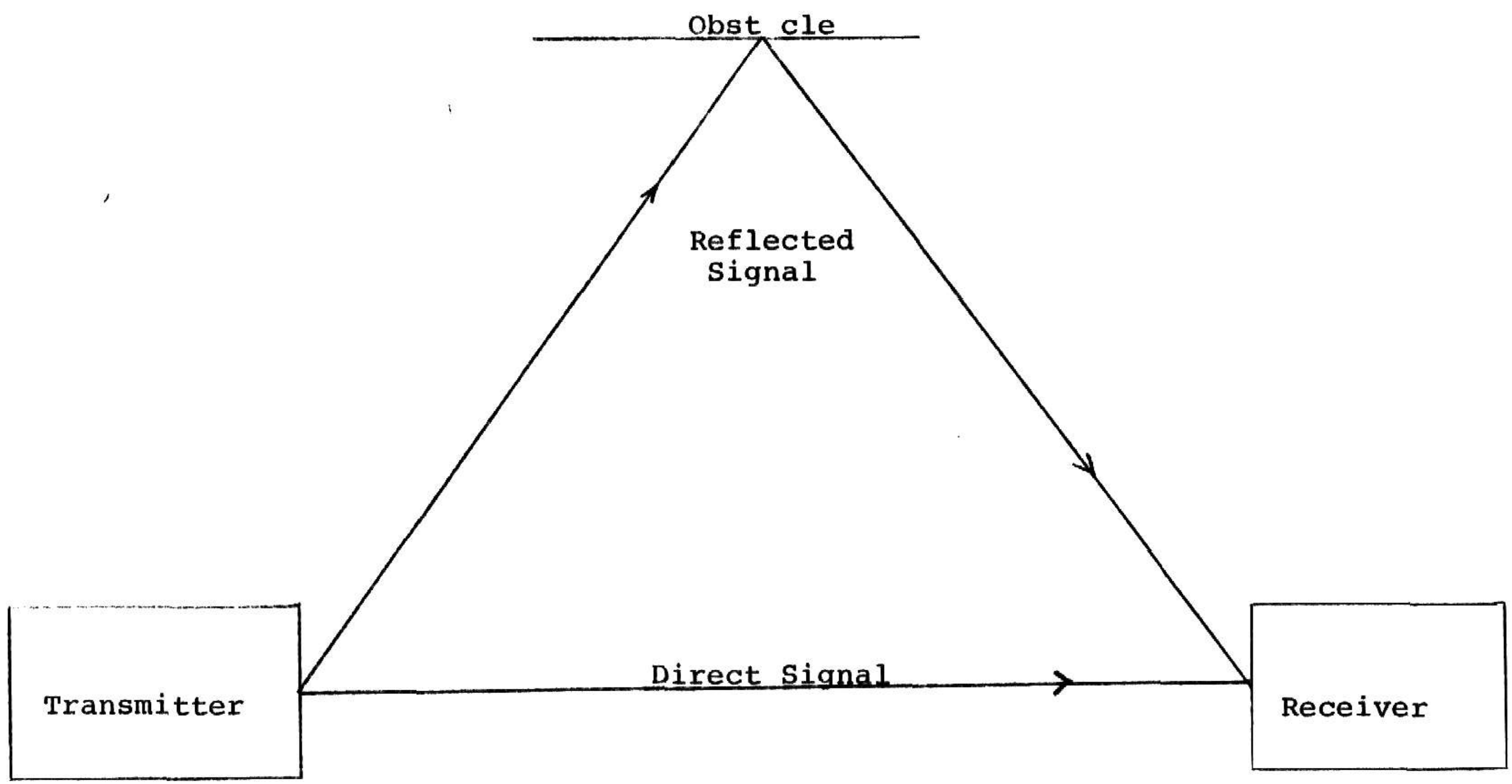

Figure 1. Diagram of System

The major purpose of this paper is to show the development of a package of computer programs to generate the values for probability of error as calculated from the equations derived by Kwon and Shehadeh.

This report begins with a review of the derivation with additional comments where appropriate to clarify some points due to the conciseness of the reference. It is then shown that the equations derived in an earlier paper by Kwon and Shehadeh (9) associated with the noncoherent detection of FSK are a special case of this derivation.

With this, the programs were developed to incorporate the results of both cases (7), (9) and to simulate either system at the user's discretion.

Some considerations in the development which imposed major constraints on the structure and logical flow of the programs were memory available, excessive run time and round-off error. The implications of these factors and the techniques used to overcome them are discussed in detail as their effects appear in the generation of the programs.

II. DERIVATION OF EQUATIONS FOR SYSTEM WITH MULTIPATH FADING

The following section is an explanation and clarification of the derivation by Kwon and Shehadeh (7). In analyzing the problem one must begin with a diagram of the receiver, Figure 2. Although this is by no means neither the only system nor the optimum one, it is one that is commonly used for the detection of FSK signals.

The received signal is the sum of the direct signal, the reflected signal and the noise.

$$r(t) = S_d(t) + S_r(t) + N(t) \qquad 0 \leq t \leq T$$

where

$$S_d(t) = A\cos\,(\omega_i t + \Delta\omega t) \qquad i = 0,1$$

$$S_r(t) = R(t)\cos\,[\omega t + \phi(t)] \qquad (1)$$

$R(t)$ and $\phi(t)$ have Rayleigh and uniform distributions, respectively, and are assumed to be statistically independent.

When setting up the model one has two extreme conditions for fading. One is that of fast fading. This is the case when the fading bandwidth is much larger than the information bandwidth. The other condition is called slow fading and occurs when the fading bandwidth is much smaller than that of the information signal. In the general model both fast and slow fading exist simultaneously. In the case of slow fading, all the reflected signal is passed through the band pass filter. This results in maximum fading degradation. We will analyze the system for this worst case.

The time delay of the reflected signal with respect to the direct signal will in general be a random variable. We will consider the two

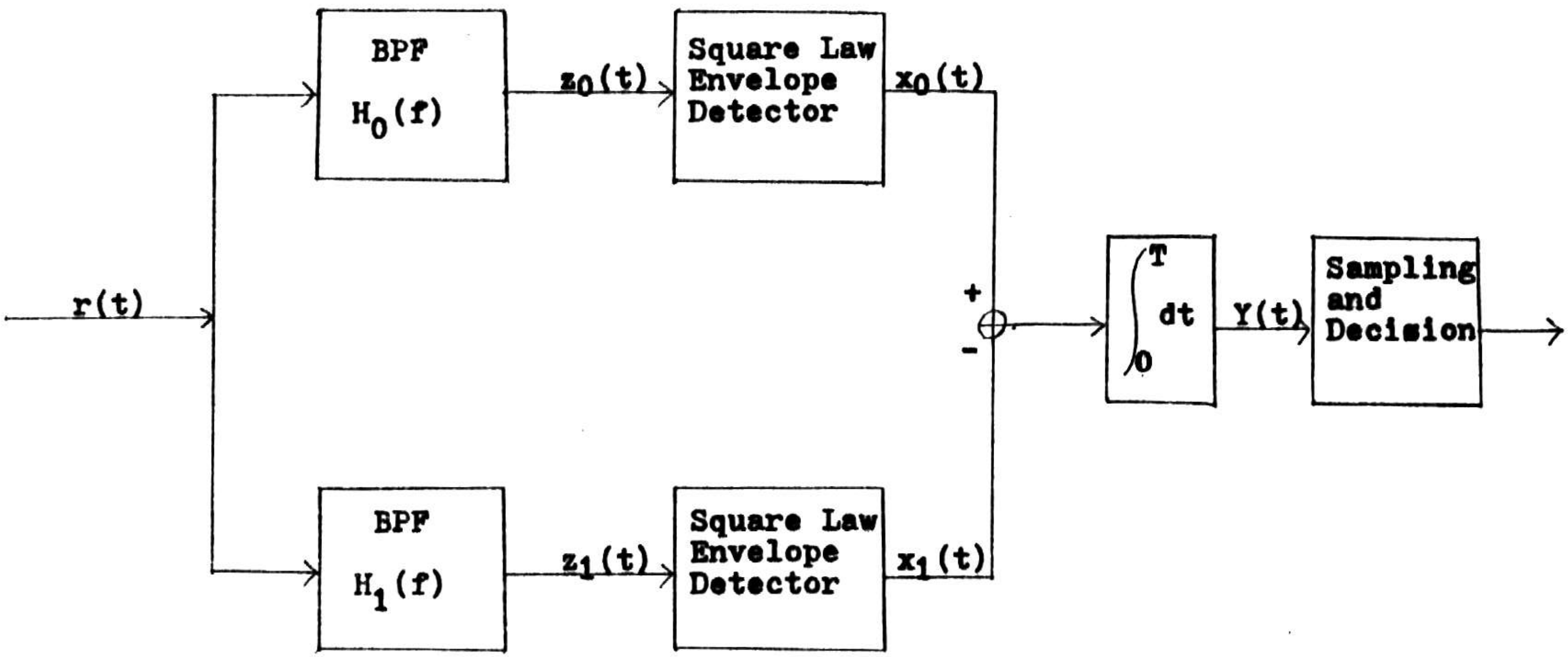

Figure 2. Block Diagram of FSK Receiver.

specific cases which give the best and worst performances.

1. Direct and reflected signals both at the frequency f_o over the entire bit interval,
2. Direct signal is at frequency f_o whole the reflected signal is at frequency f_1 over the entire bit interval.

Since any system will be a combination of the two cases the performance will fall somewhere between the best and worst case. We can assume the cases occur with equal probability, .5. Then we can analyze the system under Case 1. From these results we can then generalize the results for Case 2, which we average to obtain the final result. Under these assumptions: slow fading and Case 1;

$$Z_0(t) = S_{d0}(t) + S_{r0}(t) + N_0(t)$$

$$Z_0(t) = N_1(t)$$

where

$$S_{d0}(t) = A\cos \omega_0 t$$

$$S_{r0}(t) = y_{c0}(t)\cos \omega_0 t + y_{s0}(t)\sin \omega_0 t$$

$$N_0(t) = N_{c0}(t)\cos \omega_0 t + N_{s0}(t)\sin \omega_0 t$$

$$N_1(t) = N_{c1}(t)\cos \omega_1 t + N_{s1}(t)\sin \omega_1 t \tag{2}$$

Thus, the output of the integrate and dump stage can be written as

$$Y(t) = d_0 - d_1 = (d_{01} + d_{02}) - (d_{11} + d_{12}) \tag{3}$$

where

$$d_{01} = \int_0^T [A + N_{c0}(t) + y_{c0}(t)]^2\, dt$$

$$d_{02} = \int_0^T [N_{s0}(t) + y_{s0}(t)]^2\, dt$$

$$d_{11} = \int_0^T N_{c1}^2(t)\ dt$$

$$d_{12} = \int_0^T N_{s1}^2(t)\ dt \tag{4}$$

The decision is made to accept hypothesis H_0 if $Y(t) > 0$. The probability of error can be expressed as

$$\begin{aligned} P(E) &= P[d_0 < d_1 | H_0] \\ &= \int_0^\infty p_{d_0}(\alpha) \left(\int_0^\alpha p_{d_1}(\beta)\ d\beta \right) d\alpha \end{aligned} \tag{5}$$

The next step is to derive these probability density functions p_{d_0} and p_{d_1}.

A band limited signal with Gaussian power spectral density function has a correlation function that can be written as

$$\begin{aligned} R_n(t) &= E[n(t)\ n(s)] \qquad \tau = t-s \\ &= \int_{-\infty}^{\infty} S_n(f) \cos 2\pi\ f\tau\ df \\ &= \lim_{m\to\infty} \sum_{i=1}^{m} h_i \cos 2\pi\ f_i\ \tau \end{aligned} \tag{6}$$

when $\{h_i, f_i\}$ are the weights and abscissas with respect to the weight function $S_n(f)$.

Yaglom has shown (8) that a narrow band Gaussian process $n(t)$ with zero mean and correlation function given by Eq. 6 can be expanded as

$$n(t) = \lim_{m\to\infty} \sum_{i=1}^{m} n_{ci} \cos 2\pi\ f_i t + n_{si} \sin 2\pi\ f_i t \tag{7}$$

where n_{ci} and n_{si} are independent Gaussian random variables with zero means and variances h_i. The correlation function of $n(t)$, which is an ideal band-limited Gaussian process with zero mean and power spectral density $S_n(f)$ equal to N_0 over the frequency interval $[-B, B]$, may be expressed as

$$R_n(t) = \int_{-B}^{B} N_0 \cos 2\pi f\tau \, df = 2BN_0 \lim_{m\to\infty} \sum_{i=1}^{m} h_i \cos 2\pi Bz_i\tau \tag{8}$$

where h_i and z_i are the weights and abscissas of a GQR with respect to the unit weight function over the interval [-1,1]. From Eqs. 6, 7, and 8 we can write n_{c0}, y_{c0} as

$$n_{c0}(t) = \sigma \lim_{m\to\infty} \sum_{i=1}^{m} n_{ci} \cos 2\pi Bz_i t + n_{si} \sin 2\pi Bz_i t$$

$$y_{c0}(t) = \Psi \lim_{m\to\infty} \sum_{i=1}^{m} y_{ci} \cos 2\pi Brz_i t + y_{si} \sin 2\pi Brz_i t \tag{9}$$

where $\sigma = (2BN_0)^{\frac{1}{2}}$ and $\Psi = (A^2/2\gamma^2)^{\frac{1}{2}}$. γ^2 is the ratio of the direct to the reflected power. n_{ci}, n_{si}, y_{ci}, y_{si} are independent Gaussian random variables with zero means and variances h_i. Substituting Eq. 9 in Eq. 4 reveals d_{01} is the integral of the product of trucated summations. These are much easier to manipulate as matrix products.

$$d_{01} = \lim_{m\to\infty} 2BN_0T \int_0^1 \left(A^2/\sigma^2 + 2A/\sigma \, C_0^T X_0 + X_0^T C_0^T C_0 X_0 \right) dt \tag{10}$$

X_0 is a column vector with elements x_{0i} being independent Gaussian random variables with zero means and variances

$$\sigma^2_{xi} = \sigma^2_{xi+m} = h_i$$

$$\sigma^2_{xi+2m} = \sigma^2_{xi+3m} = h_i \Psi^2/\sigma^2 = \text{SNR } h_i/\gamma^2 \; 2BT \tag{11}$$

The elements of $\tilde{c}_0$ are

$$\tilde{c}_{0i} = \cos 2\pi BTz_i t$$

$$\tilde{c}_{0i+m} = \sin 2\pi BTz_i t$$

$$\tilde{c}_{0i+2m} = \cos 2\pi B_r Tz_i t$$

$$\tilde{c}_{0i+3m} = \sin 2\pi B_r Tz_i t \tag{12}$$

if $C_0^T = \int_0^1 \tilde{C}_0^T\,dt$ and $F_0 = \int_0^1 \tilde{C}_0^T\tilde{C}_0\,dt$ the elements of F_0 are

$$f_{0i,j} = \int_0^1 \cos 2\pi BTz_i t \cos 2\pi BTz_j t$$

$$f_{0i,j+m} = \int_0^1 \cos 2\pi BTz_i t \sin 2\pi BTz_j t$$

$$f_{0i,j+2m} = \int_0^1 \cos 2\pi BTz_i t \cos 2\pi B_r Tz_j t$$

$$f_{0i,j+3m} = \int_0^1 \cos 2\pi BTz_i t \sin 2\pi B_r Tz_j t$$

$$f_{0i+m,j+m} = \int_0^1 \sin 2\pi BTz_i t \sin 2\pi B_r Tz_j t$$

$$f_{0i+m,j+2m} = \int_0^1 \sin 2\pi BTz_i t \cos 2\pi B_r Tz_j t\,dt$$

$$f_{0i+m,j+3m} = \int_0^1 \sin 2\pi BTz_i t \sin 2\pi B_r Tz_j t\,dt$$

$$f_{0i+2m,j+2m} = \int_0^1 \cos 2\pi B_r Tz_i t \cos 2\pi B_r Tz_j t\,dt$$

$$f_{0i+2m,j+3m} = \int_0^1 \cos 2\pi B_r Tz_i t \sin 2\pi B_r Tz_j t\,dt$$

$$f_{0i+3m,j+3m} = \int_0^1 \sin 2\pi B_r Tz_i t \sin 2\pi B_r Tz_j t\,dt \tag{13}$$

and the elements of C_0^T are

$$c_{0i} = \int_0^1 \cos 2\pi BTz_i t\,dt$$

$$c_{0i+m} = \int_0^1 \sin 2\pi BTz_i t\,dt$$

$$c_{0i+2m} = \int_0^1 \cos 2\pi B_r Tz_i t\,dt$$

$$c_{0i+3m} = \int_0^1 \sin 2\pi B_r Tz_i t\,dt \tag{14}$$

Making these substitutions leads to

$$d_{0i} = \lim_{m\to\infty} 2BN_0T[A^2/\sigma^2 + 2A/\sigma\; C_0^T X_0 + X_0^T F_0 X_0] \tag{15}$$

At this point it is convenient to make a sequence of substitutions that simplify the calculations needed to find the characteristic function of d_{01}. First, let $D_0V_0 = X_0$, where the elements v_{0i} of V_0 are independent Gaussian random variables with zero means and unit variances and D_0 is a diagonal matrix with elements

$$d_{0i} = \sigma_i \qquad i = 1,2,\ldots 4m \tag{16}$$

Next, let $U_0 = M_0^TV_0$, where M_0 is formed by ordering the eigenvectors of $D_0F_0D_0$ in columns. Then the u_{0i} of U_0 are independent Gaussian random variables with zero means and unit variances. Making these substitutions yields

$$d_{01} = \lim_{m\to\infty} 2BN_0T[A^2/\sigma^2 + \frac{2AC_0^T}{\sigma} C_0M_0U_0 + U_0^TM_0^TD_0F_0D_0M_0U_0] \tag{17}$$

It is obvious by looking at Eq. 17 that $M_0^TD_0F_0D_0M_0$ is a similarity transformation of $D_0F_0D_0$. Since M_0 is the eigenvector matrix of $D_0F_0D_0$ this is the Karhunen Loeve transform of $D_0F_0D_0$ yielding the diagonal matrix which we denote D_λ. The elements λ_{0i} of D_λ are the eigenvalues of $D_0F_0D_0$. By making the substitution $R_0^T = C_0^TD_0M_0$

$$d_{01} = \lim_{m\to\infty} 2BN_0T\left[\frac{A^2}{\sigma^2} + \frac{2AR_0^TU_0}{\sigma} + U_0^TD_\lambda U_0\right] \tag{18}$$

Rewriting these matrix products as summations

$$d_{01} = \lim_{m\to\infty} \sum_{i+1}^{4m} \lambda_{0i}(u_{0i} + r_{0i}A^2/\lambda_{0i})^2 +$$

$$A^2/\sigma^2(1 - \sum_{n=1}^{4m} r_{0n}^2/\lambda_{0n}) \tag{19}$$

The last term $A^2/\sigma^2\ (1 - \sum_{n=1}^{4m} r_{0n}^2/\lambda_{0n})$ can be shown to be zero using the properties of the orthogonal functions which are the solutions of the equation

$$\lambda_i\phi_i(t) = \int_0^T K_n(t,s)\ \phi_i(s)\ ds \tag{20}$$

where $K_n(t,s)$ is the covariance function of the process.

$$d_{01} = \lim_{m\to\infty} \sum_{i=1}^{4m} \lambda_{0i}(u_{0i} + r_{0i}A^2/\lambda_{0i})^2 \tag{21}$$

The characteristic function of d_{01} can be written as

$$M_{d_{01}}(v) = E[\exp(jvd_{01})] \tag{22}$$

$$= E\left[\exp\left(\lim_{m\to\infty}\left[\sum_{i=1}^{4m} jv\ \lambda_{0i}(\mu_{0i} + v_{0i}\ A^2/\lambda_{0i})^2\right]\right)\right] \tag{23}$$

$$= \lim_{m\to\infty} \int_{-\infty}^{\infty} \exp\left\{\sum_{i=1}^{4m} jv\ \lambda_{0i}(\mu_{0i} + r_{0i}\ A^2/\lambda_{0i})^2\right\}dd_{01} \tag{24}$$

$$= \lim_{m\to\infty} \int_{-\infty}^{\infty} \prod_{i=1}^{4m} \exp(j_v\ \lambda_{0i}(\mu_{0i} + r_{0i}\ A^2/\lambda_{0i})^2)dd_{01} \tag{25}$$

$$= \lim_{m\to\infty} \prod_{i=1}^{4m} \frac{\exp[jv\ \lambda_{0i}r_{0i}^2A^2/\lambda_{0i}^2\sigma^2(1-j2\lambda_{0i}v)]}{(1-2j\lambda_{0i}v)^{\frac{1}{2}}} \tag{26}$$

The characteristic functions of d_{02}, d_{11}, d_{12} can be obtained by setting A=0 in Eq. 21. Since d_{01}, d_{02}, d_{11} and d_{12} are independent random variables the characteristic functions of d_0 and d_1 are

$$M_{d_0}(v) = \lim_{m} \prod_{i=1}^{4m} \frac{\exp[jv\ \lambda_{0i}r_{0i}^2A^2/\lambda_{0i}^2\sigma^2(1-2j\lambda_{0i}v)]}{(1-2j\lambda_{0i}v)} \tag{27}$$

$$M_{d_1}(v) = E[\exp(jvd_1)]$$

$$= \lim_{m\to\infty} \sum_{i=1}^{4m} \frac{K_i}{1-2j\lambda_{1i}v} \tag{28}$$

where

$$K_{1i} = \prod_{\substack{j=1 \\ i \neq j}}^{2m} 1/(1-\lambda_{ij}/\lambda_{1i}) \tag{29}$$

$\{\lambda_{1i}\}$ are the eigenvalues of $D_1F_1D_1$ where D_1 is a diagonal matrix with elements

$$d_{1i} = d_{0i} \qquad i = 1,2,\ldots 2m$$

F_1 has elements

$$f_{1i,j} = f_{0i,j} \qquad i,j = 1,2,\ldots 2m$$

C_1 has elements

$$c_{1i} = c_{0i} \qquad i = 1,2,\ldots 2m$$

The elements of R_1 are

$$r_{1i} = r_{0i} \qquad i = 1,2,\ldots 2m \tag{30}$$

The probability density functions of $M_{d_0}(v)$ and $M_{d_1}(v)$, $p_{d_0}(\alpha)$ and $p_{d_1}(\beta)$ can be found as the transforms of Eq. 27 and Eq. 28, respectively. Putting these into Eq. 5

$$P_1(E) = \int_0^\infty p_{d_0}(\alpha) \left(\int_0^\alpha \lim_{m\to\infty} \sum_{i=1}^{2m} \frac{K_{1i}}{1-2j\lambda_{1i}\beta} \, d\beta \right) d\alpha \tag{31}$$

$$= \int_0^\infty p_{d_0}(\alpha) \lim_{m\to\infty} \sum_{i=1}^{2m} K_{1i} \exp(-\alpha/2\lambda_{1i}) \, d\alpha \tag{32}$$

$$= \lim_{m\to\infty} \sum_{i=1}^{2m} K_{1i} \int_0^\infty p_{d_0}(\alpha) \exp(-\alpha/2\lambda_{1i}) \, d\alpha \tag{33}$$

Since $p_{d_0}(\alpha) = 0$ for $\alpha<0$ this can be written as

$$P_1(E) = \lim_{m\to\infty} \sum_{i=1}^{2m} K_{1i} \int_{-\infty}^{\infty} p_{d_0}(\alpha) \exp(-\alpha/2\lambda_{1i})\, d\alpha \tag{34}$$

This is the inverse transform of $p_{d_0}(v)$ at $jv = -1/2\lambda_{1i}$ or

$$M_{d_0}(v)\,|\,jv = -1/2\lambda_{1i} \tag{35}$$

so

$$P_1(E) = \lim_{m\to\infty} \sum_{i=1}^{2m} K_{1i} \prod_{n=1}^{4m} \frac{\exp[-r_{0n}^2 A^2/2\lambda_{1i}\lambda_{0n}\sigma^2(1 + \lambda_{0n}/\lambda_{1i})]}{1 + \lambda_{0n}/\lambda_{1i}} \tag{36}$$

But A^2/σ^2 = SNR/BT, thus

$$P_1(E) = \lim_{m\to\infty} \sum_{i=1}^{2m} K_{1i} \prod_{n=1}^{4m} \frac{\exp[-r_{0n}^2 \mathrm{SNR}/2\mathrm{BT}\lambda_{0n}(\lambda_{1i} + \lambda_{0n})]}{1 + \lambda_{0n}/\lambda_{1i}} \tag{37}$$

With this result for Case 1, which is the case of both direct and reflected signal at frequency f_0 over the entire bit interval, we can obtain the result for Case 2, which is direct signal at frequency f_0 while reflected signal is at frequency f_1, through a similar procedure yielding;

$$P_2(E) = \lim_{m\to\infty} \sum_{i=1}^{4m} K_{0i} \prod_{n=1}^{2m} \frac{\exp[-r_{1n}^2 \mathrm{SNR}/2\mathrm{BT}\lambda_{1n}(\lambda_{0i} + \lambda_{1n})]}{1 + \lambda_{1n}/\lambda_{0i}} \tag{38}$$

The average probability of error can be expressed as

$$P(E) = \tfrac{1}{2}[P_1(E) + P_2(E)] \tag{39}$$

With this, we now have an expression for the probability of error for a noncoherent FSK system in the presence of multipath fading as a function of SNR, BT, direct to reflected power ratio and fading bandwidth. Kwon and Shehadeh claim this approximation is as much as .5 dB better than that of Chadwick for the combination of SNR $\geq$ 14 dB, γ^2 = 10 dB, BT = .5, B_r = B.

III. EXTENSION TO THE CASE OF NONCOHERENT DETECTION OF FSK SIGNALS

It can be shown that the derivation presented in an earlier paper by Kwon and Shehadeh (9) on noncoherent detection of FSK is a special case of the equations just derived. In the average FSK system, multipath fading is not a concern and analysis of the system need only take into account the uncertainty in the received signal due to the noise in the channel. The approach used to derive the equations for probability of error in this case is very similar to that just presented.

The received signal is the sum of the information term and the noise. The result in Eq. 4 in this case looks like

$$d_{01} = \int_0^T [A + n_{c0}(t)]^2 \, dt$$

$$d_{02} = \int_0^T n_{s0}^2(t) \, dt$$

$$d_{11} = \int_0^T n_{c1}^2(t) \, dt$$

$$d_{12} = \int_0^T n_{s1}^2(t) \, dt \tag{40}$$

The same matrix multiplication technique can be used so that

$$d_{01} = \lim_{m\to\infty} 2BN_0T[A^2/\sigma^2 + 2A/\sigma \, C^TDMU + U^TM^TDFDMU] \tag{41}$$

but C^T has elements

$$c_i = \int_0^1 \cos 2\pi BTz_it \, dt \qquad i = 1,2,\ldots m$$

$$c_{i+m} = \int_0^1 \sin 2\pi BTz_it \, dt \tag{42}$$

F has the elements

$$f_{i,j} = \int_0^1 \cos 2\pi BTz_i t \cos 2\pi BTz_j t \, dt$$

$$f_{i,j+m} = \int_0^1 \cos 2\pi BTz_i t \sin 2\pi BTz_j t \, dt$$

$$f_{i+m,j+m} = \int_0^1 \sin 2\pi BTz_i t \sin 2\pi BTz_j t \, dt \tag{43}$$

D is a diagonal matrix with elements

$$d_i = d_{i+m} = (h_i)^{\frac{1}{2}} \qquad i = 1,2,\ldots m \tag{44}$$

Again, Eq. 41 can be written as

$$d_{01} = \lim_{m\to\infty} 2BN_0T[A^2/\sigma^2 + 2A/\sigma\, R^TU + U^TD\,U] \tag{45}$$

Where $R^T = C^TDM$ and D_λ is a diagonal matrix with elements λ_i being the eigenvalues of DFD.

$$d_{01} = \lim_{m\to\infty} \sum_{i=1}^{2m} \lambda_i(u_i + r_iA^2/\lambda_i)^2 \tag{46}$$

The characteristic function of d_{01} is

$$M_{d_{01}}(v) = \lim_{m\to\infty} \prod_{i=1}^{2m} \frac{\exp[jv\lambda_i r_i^2A^2/\lambda_i^2\sigma^2(1-2j\lambda_iv)]}{(1-j2\lambda_iv)^{\frac{1}{2}}} \tag{47}$$

The characteristic equations for d_{02}, d_{11}, d_{12} can be found by setting $A = 0$ in Eq. 45. Since d_{01}, d_{02}, d_{11} and d_{12} are independent random variables the characteristic functions of d_0 and d_1 are

$$M_{d_0}(v) = \lim_{m\to\infty} \prod_{i=1}^{2m} \frac{\exp[jvr_i^2A^2/\lambda_i\sigma^2(1-j2\lambda_iv)]}{(1-j2\lambda_iv)} \tag{48}$$

$$M_{d_1}(v) = \lim_{m\to\infty} \sum_{i=1}^{2m} \frac{K_i}{1-2j\lambda_i v} \tag{49}$$

where

$$K_i = (1-2j\lambda_i v)M_{d_1}(v)\,|\,jv = -1/2\lambda_i$$

From the transform of $M_{d_1}(v)$ the probability density functions are

$$p_{d_1}(\beta) = \lim_{m\to\infty} \sum_{i=1}^{2m} K_i/2\lambda_i \exp(-\beta/2\lambda_i) \tag{50}$$

Thus, the probability of error is

$$P(E) = \int_0^\infty p_{d_0}(\alpha) \int_0^\alpha p_{d_1}(\beta)\, d\beta \quad d\alpha \tag{51}$$

$$= \lim_{m\to\infty} \sum_{i=1}^{2m} K_i \int_0^\infty p_{d_0}(\alpha) \exp(-\alpha/2\lambda_i)\, d\alpha \tag{52}$$

$$= \lim_{m\to\infty} \sum_{i=1}^{2m} K_i \int_{-\infty}^\infty p_{d_0}(\alpha) \exp(-\alpha/2\lambda_i)\, d\alpha \tag{53}$$

$$= M_{d_0}(v)\,|\,jv = -1/2\lambda_i \tag{54}$$

$$= \lim_{m\to\infty} \sum_{i=1}^{2m} K_i \prod_{n=1}^{2m} \frac{\exp[-r_n^2 SNR/2BT(\lambda_i + \lambda_n)\lambda_n]}{1 + \lambda_n/\lambda_i} \tag{55}$$

This is very similar to Eqs. 37 and 38 and would be the result if Eq. 37 were set equal to Eq. 38. In this case, there are some interesting characteristics of the eigenvalues. They approach zero rapidly for $i > (2BT + 1)$ and therefore, the K_i approach zero rapidly for $i > (2BT + 1)$. Another property is that $\sum_{i=1}^{2m} \lambda_i = 1$.

Kwon and Shehadeh claim that this technique results in a marked improvement in the approximation as compared to the sampling and summing technique. It shows an improvement of 1 dB for BT = 2.5 and .3 dB for BT = 2 dB at high SNR.

IV. DEVELOPMENT OF THE COMPUTER PROGRAM

Before beginning the development of the program, it is necessary to make some comments concerning the computer used. It was a Data General Nova 1200 which has 64K of core memory and one disk drive. The language in which it was written was FORTRAN.

The first step is to define the problem. What was needed was a program that would input values for SNR, BT, B_r/B, and γ^2 and return with a value for the probability of error in accordance with Eq. 39. Since the final result would be most convenient as a relationship between a series of values for SNR and the corresponding values for P(E), SNR could be increased by a known amount each time through a DO loop which calculates P(E). In order to take advantage of the graphics capabilities of the terminal used, these values for P(E) had to be written out to disk. With this sketchy outline, the next step is to decide what values are necessary to calculate the final result.

Besides the above-mentioned parameters, a value for m, which is the factor setting the number of terms to be incorporated, needed to be input. Also, the GQR coefficients corresponding to this m must be input. The only way available to get the GQR coefficients in this case was to look them up in a reference (10). This required a decision to be made on the value of m needed because the simplest way to get these coefficients into the program was to incorporate them as part of the program code. Discussion of the value chosen for m will be left for a later discussion when its importance becomes more apparent.

The next step is to set up the matrices: D_0, D_1, F_0, F_1, C_0, C_1, according to the corresponding equations. These matrices are multiplied to get $D_0E_0D_0$ and $D_1F_1D_1$, whose eigenvalues and eigenvector matrices are then computed. With these values R_0, R_1, K_0, and K_1 are generated. This now gives all the values needed to calculate P(E). When committing these steps to code, one almost immediately runs into the problem of lack of memory. Kwon and Shehadeh suggest an adequate value for m to be 10. A look at the matrices needed and their sizes yielded a total of more than 12K floating point numbers to be stored. With this system each single precision number takes 4 bytes of memory to store it. This means more than 48K of memory would be needed to store the matrix elements. Obviously, there is not enough memory available to accommodate such a program.

There were 4 methods available to alleviate this problem. The first and most obvious was to reduce the size of the matrices, meaning, make m smaller. A tentative value of $m = 6$ was chosen with the stipulation that too much deviation from known results would require m to be increased.

Another technique employed was that of matrix compression. This can be used on the matrices: D_0, D_1, F_0, and F_1. Because D_0 and D_1 are diagonal matrices, only the diagonal need be stored in vector form. By definition, F_0 and F_1 are symmetric non-negative definite, so only the upper half plus the diagonal of each requires storage.

A third method, although very useful, often results in ambiguous coding. This is the elimination of unique intermediate matrix storage. Also, where possible, final results can be stored in one of the original matrices. This technique was used as sparingly as possible in order to obtain useful and understandable code.

The final method for decreasing the memory needed for storage was not directly related to matrix storage, but more related to code storage.

The initial estimate of code length was about 40K bytes. With this much code in core it was evident that no matrix manipulation could be accomplished. Upon examining the equations, it was found that those needed to set up the F_0 matrix constituted a large part of the code. It was decided that these equations should be written in a separate program to generate F_0 and store it on disk. This disk file would then be read into the main program. This routine must be run previous to the execution of the main routine.

Because of the parallel calculations for terms involving limits of 4m and those of 2m, it was evident a subroutine should be incorporated. This subroutine would take the matrices set up in the main routine, do all the matrix calculations and return with the values for r_i, λ_i, K_i for each case.

Upon the first execution of the program, it was found that the calculated eigenvalues were accurate to the point at which they became less than about 10^{-9}. It was decided that, due to the round-off error of the system, the packaged routine EIGEN would not be able to calculate those eigenvalues less than 10^{-9}. The proposed solution was to set up the program for double-precision accuracy. Because of the peculiarities of the machine, all the values and calculations in the supporting software had to be changed to double-precision. This doubled the amount of memory needed to store the matrices and increased the run-time by a factor of 8. With this change to double-precision, those eigenvalues as small as 10^{-18} could be found accurately. Those eigenvalues smaller than 10^{-18} do not affect the accuracy of the final result due to the fact that the associated r_i is less than 10^{-16} and the corresponding K_i is less than 10^{-150}. Because of the computations involving these eigenvalues, they could not be set to zero. Therefore, it was necessary to set them to values approximating what they would be if they could be calculated.

The subroutine to find these eigenvalues and the corresponding eigenvectors is by far the slowest link in the program. To calculate one probability of error value, it took about 14 minutes. About 13 minutes of this time was spent in this subroutine.

An equation for the run-time for single precision is

$$t = 10\ N^3\ (u + 2v)$$

where the dimensions of the matrix are N x N, u is the multiplication time of the machine, v is the addition time.

For the double precision, $N_d = 2N$, so

$$t_d = 80\ N^3\ (u + 2v)$$

It was fortunate that $m = 6$ was chosen instead of any value higher. If $m = 10$ had been used, $t_{d,10} = 65$ minutes. It would have taken more than ten hours to accumulate enough data to plot one curve.

Because of this excessive run-time, the program was set up to calculate P(E) for ten values of SNR ranging from 0 to 18 dB. When plotted, this resulted in a fairly smooth curve.

Curves for this case are shown in Figure 3. It should be noted that the shape of the curve is dependent more on γ^2 than any of the other parameters. When these curves are compared to those presented by Kwon and Shehadeh, it is extremely difficult to find any deviation even though $m = 6$ was used instead of $m = 10$. Table 1 shows the values of $\hat{L}$, K and r. It can be seen that $i = 17$ is the point at which the

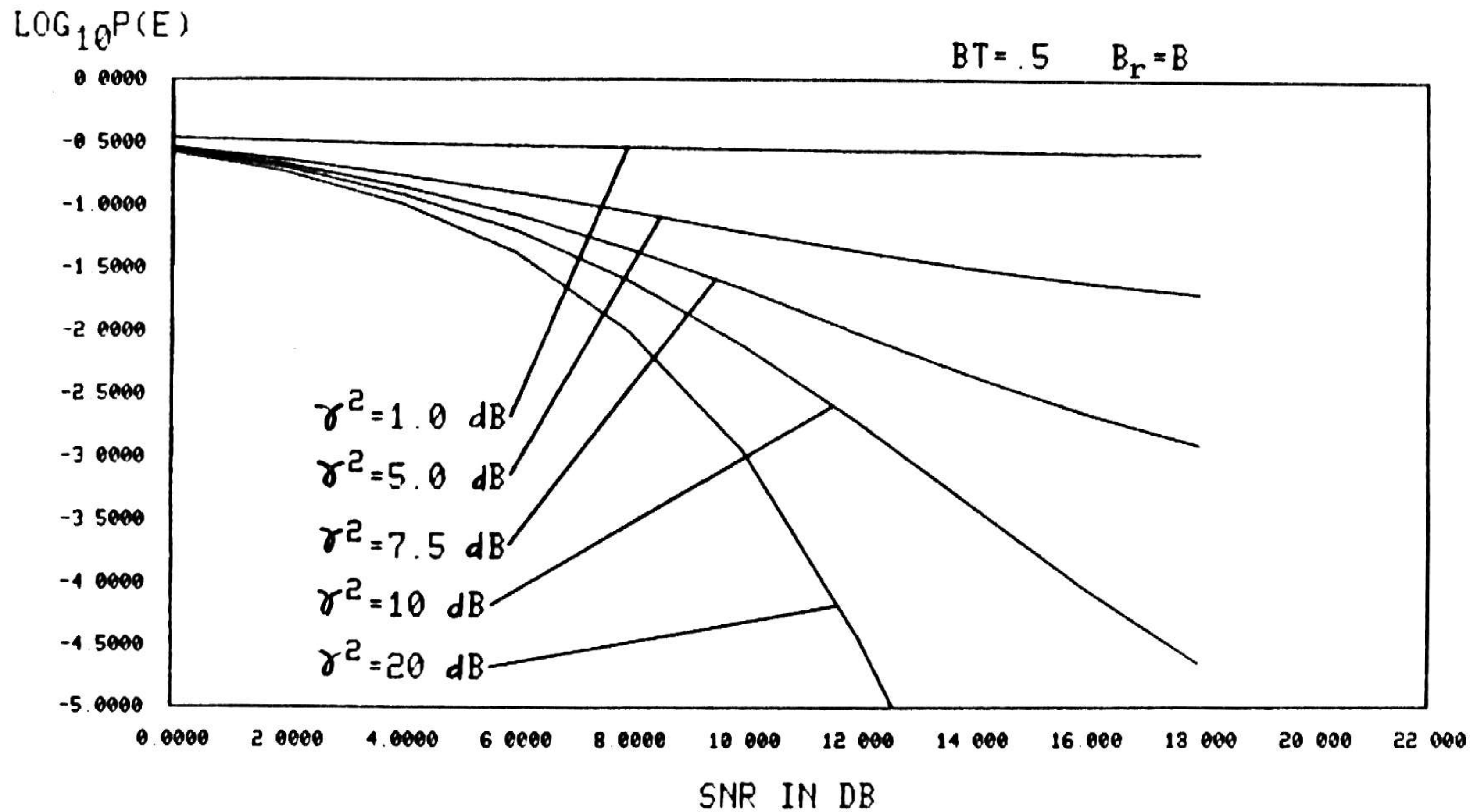

FIGURE 3. P(E) VERSUS SNR FOR MULTIPATH FADING ENVIRONMENT

TABLE 1

Numerical Values for Multipath System

$\lambda_{01} = 0.79158715$	$r_{01} = 0.88389645$	$K_{01} = 1.40474904$
$\lambda_{02} = 0.20659897$	$r_{02} = -0.10191454\times10^{-3}$	$K_{02} = -0.40857538$
$\lambda_{03} = 0.11755455\times10^{-1}$	$r_{03} = -0.93135978\times10^{-2}$	$K_{03} = 0.15786580\times10^{-1}$
$\lambda_{04} = 0.10387424\times10^{-1}$	$r_{04} = -0.80359465\times10^{-2}$	$K_{04} = -0.12028302\times10^{-1}$
$\lambda_{05} = 0.34853788\times10^{-2}$	$r_{05} = 0.83323205\times10^{-4}$	$K_{05} = 0.77924669\times10^{-4}$
$\lambda_{06} = 0.22887496\times10^{-2}$	$r_{06} = 0.50199167\times10^{-3}$	$K_{06} = -0.99106078\times10^{-5}$
$\lambda_{07} = 0.10917050\times10^{-2}$	$r_{07} = -0.38613699\times10^{-3}$	$K_{07} = 0.57170029\times10^{-7}$
$\lambda_{08} = 0.27014795\times10^{-3}$	$r_{08} = 0.17224013\times10^{-4}$	$K_{08} = -0.22395217\times10^{-11}$
$\lambda_{09} = 0.11808980\times10^{-3}$	$r_{09} = 0.75815993\times10^{-4}$	$K_{09} = 0.29942669\times10^{-14}$
$\lambda_{010} = 0.47029827\times10^{-4}$	$r_{010} = 0.23585613\times10^{-4}$	$K_{010} = -0.45319098\times10^{-18}$
$\lambda_{011} = 0.80336030\times10^{-6}$	$r_{011} = 0.57733104\times10^{-6}$	$K_{011} = 0.44443817\times10^{-36}$
$\lambda_{012} = 0.52422517\times10^{-8}$	$r_{012} = 0.49319542\times10^{-8}$	$K_{012} = -0.39482904\times10^{-60}$
$\lambda_{013} = 0.25806665\times10^{-11}$	$r_{013} = 0.25987506\times10^{-11}$	$K_{013} = 0.00000000$
$\lambda_{014} = 0.98170922\times10^{-14}$	$r_{014} = -0.48780051\times10^{-14}$	$K_{014} = 0.00000000$
$\lambda_{015} = 0.25221692\times10^{-16}$	$r_{015} = -0.74562616\times10^{-16}$	$K_{015} = 0.00000000$
$\lambda_{016} = 0.26044510\times10^{-18}$	$r_{016} = 0.19565402\times10^{-16}$	$K_{016} = 0.00000000$
$\lambda_{017} = 0.10000000\times10^{-17}$	$r_{017} = 0.10000000\times10^{-19}$	$K_{017} = 0.00000000$
$\lambda_{018} = 0.10000000\times10^{-19}$	$r_{018} = 0.10000000\times10^{-19}$	$K_{018} = 0.00000000$
$\lambda_{019} = 0.10000000\text{x1-}^{-21}$	$r_{019} = 0.10000000\times10^{-19}$	$K_{019} = 0.00000000$
$\lambda_{020} = 0.10000000\times10^{-23}$	$r_{020} = 0.10000000\times10^{-19}$	$K_{020} = 0.00000000$
$\lambda_{021} = 0.10000000\times10^{-25}$	$r_{021} = 0.10000000\times10^{-19}$	$K_{021} = 0.00000000$
$\lambda_{022} = 0.10000000\times10^{-27}$	$r_{022} = 0.10000000\times10^{-19}$	$K_{022} = 0.00000000$
$\lambda_{023} = 0.10000000\times10^{-29}$	$r_{023} = 0.10000000\times10^{-19}$	$K_{023} = 0.00000000$
$\lambda_{024} = 0.10000000\times10^{-31}$	$r_{024} = 0.10000000\times10^{-19}$	$K_{024} = 0.00000000$
$\lambda_{11} = 0.78336878$	$r_{11} = 0.87951834$	$K_{11} = 1.37487670$
$\lambda_{12} = 0.20503983$	$r_{12} = -0.10615016\times10^{-16}$	$K_{12} = -0.37575883$
$\lambda_{13} = 0.11373989\times10^{-1}$	$r_{13} = -0.11937155\times10^{-1}$	$K_{13} = 0.88214019\times10^{-3}$
$\lambda_{14} = 0.21521849\times10^{-3}$	$r_{14} = -0.22768245\times10^{-16}$	$K_{14} = -0.56259515\times10^{-8}$
$\lambda_{15} = 0.21572193\times10^{-5}$	$r_{15} = 0.23875466\times10^{-5}$	$K_{15} = 0.56002102\times10^{-16}$
$\lambda_{16} = 0.13610926\times10^{-7}$	$r_{16} = 0.29798619\times10^{-16}$	$K_{16} = -0.55670195\times10^{-27}$
$\lambda_{17} = 0.59188319\times10^{-10}$	$r_{17} = -0.66188564\times10^{-10}$	$K_{17} = 0.37525724\times10^{-41}$
$\lambda_{18} = 0.18829990\times10^{-12}$	$r_{18} = 0.11827968\times10^{-16}$	$K_{18} = -0.12353444\times10^{-58}$
$\lambda_{19} = 0.45931491\times10^{-15}$	$r_{19} = -0.55245521\times10^{-15}$	$K_{19} = 0.00000000$
$\lambda_{110} = 0.75844397\times10^{-18}$	$r_{110} = 0.10137290\times10^{-16}$	$K_{110} = 0.00000000$
$\lambda_{111} = 0.10000000\times10^{-21}$	$r_{111} = 0.10000000\times10^{-19}$	$K_{111} = 0.00000000$
$\lambda_{112} = 0.10000000\times10^{-23}$	$r_{112} = 0.10000000\times10^{-19}$	$K_{112} = 0.00000000$

eigenvalues become less than 10^{-18}. Table 2 shows values of P(E) for BT = .5, γ^2 = 20 dB, B_r = B and SNR ranging from 0 to 18 dB.

The noncoherent detection of FSK without regard to fading is the case most generally encountered and studied. Because this is a special case of the system with fading the capability to generate the probability of error for this special case is built into the program. When queried by the computer, by replying "Paper = 1," the user can simulate the system described by the equation derived in the first paper by Kwon and Shehadeh (9). Because it is only necessary to find the eigenvalues of $D_1F_1D_1$ once for this case and the fact that D_1 is a vector, only 2m element long, the run-time for this case is reduced to only a few minutes.

Probability of error curves for this case are shown in Figure 4. As one would expect, by decreasing the bandwidth of the message, and also correspondingly decreasing the bandwidth of the filter $H_1(f)$, the probability of error increases much more rapidly for increasing signal to noise ratio.

Table 3 shows the values of λ_i, r_i, K_i and P(E), for BT = .5. These values correspond to those presented by Kwon and Shehadeh as closely as the seventh decimal place for most values of i.

TABLE 2

Values of Probability of Error for Multipath System

SNR = 0 dB	PE = 0.26931934
SNR = 2 dB	PE = 0.18730122
SNR = 4 dB	PE = 0.10534453
SNR = 6 dB	PE = $0.42480663 \times 10^{-1}$
SNR = 8 dB	PE = $0.10246072 \times 10^{-1}$
SNR = 10 dB	PE = $0.11352198 \times 10^{-2}$
SNR = 12 dB	PE = $0.39953080 \times 10^{-4}$
SNR = 14 dB	PE = $0.27605609 \times 10^{-6}$
SNR = 16 dB	PE = $0.21927799 \times 10^{-9}$
SNR = 18 dB	PE = $0.12854767 \times 10^{-13}$

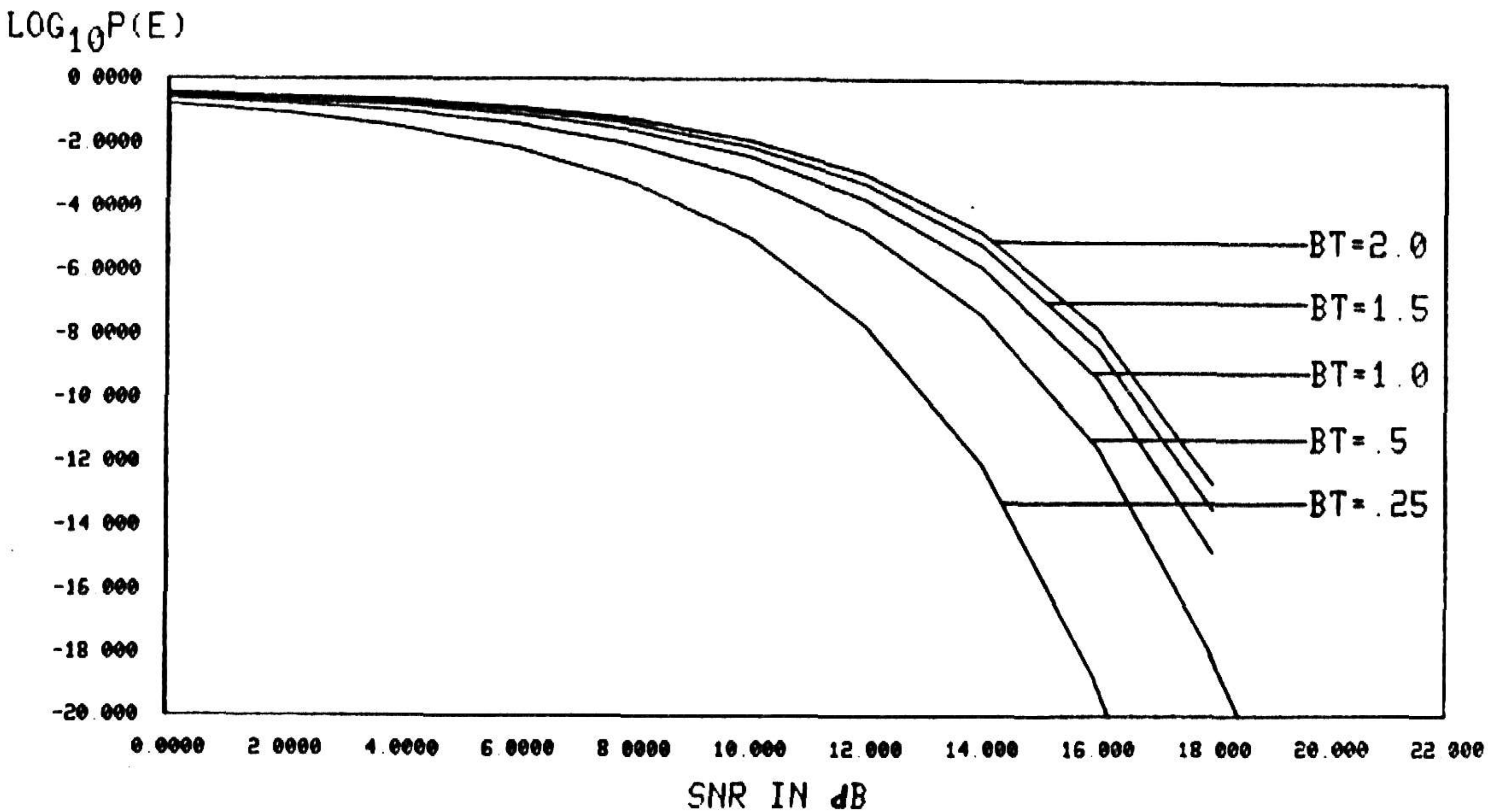

FIGURE 4. P(E) VERSUS SNR FOR NONCOHERENT FSK DETECTION

TABLE 3

Numerical Values for Noncoherent Detection of FSK

$\lambda_1 = 0.49052313$	$r_1 = 0.66225975$	$K_1 = 5.79117616$
$\lambda_2 = 0.37481009$	$r_2 = -0.82440644\text{x}10^{-16}$	$K_2 = -4.9687792$
$\lambda_3 = 0.12179650$	$r_3 = 0.11324031$	$K_3 = 0.17770627$
$\lambda_4 = 0.12323273\text{x}10^{-1}$	$r_4 = -0.60091693\text{x}10^{-18}$	$K_4 = -0.10319623\text{x}10^{-3}$
$\lambda_5 = 0.53302987\text{x}10^{-3}$	$r_5 = -0.56155333\text{x}10^{-3}$	$K_5 = 0.31614452\text{x}10^{-9}$
$\lambda_6 = 0.13707566\text{x}10^{-4}$	$r_6 = -0.17685836\text{x}10^{-16}$	$K_6 = -0.34424461\text{x}10^{-17}$
$\lambda_7 = 0.24013907\text{x}10^{-6}$	$r_7 = -0.26259339\text{x}10^{-6}$	$K_7 = 0.98118466\text{x}10^{-28}$
$\lambda_8 = 0.30666013\text{x}10^{-8}$	$r_8 = 0.14425804\text{x}10^{-16}$	$K_8 = -0.53896183\text{x}10^{-41}$
$\lambda_9 = 0.29856769\text{x}10^{-10}$	$r_9 = -0.33083612\text{x}10^{-10}$	$K_9 = 0.43286208\text{x}10^{-57}$
$\lambda_{10} = 0.22915534\text{x}10^{-12}$	$r_{10} = -0.10278427\text{x}10^{-16}$	$K_{10} = -0.39860462\text{x}10^{-76}$
$\lambda_{11} = 0.14131188\text{x}10^{-14}$	$r_{11} = 0.15640835\text{x}10^{-14}$	$K_{11} = 0.00000000$
$\lambda_{12} = 0.71359684\text{x}10^{-17}$	$r_{12} = -0.11361755\text{x}10^{-16}$	$K_{12} = 0.00000000$

SNR = 0 dB	PE = 0.3255364397
SNR = 2 dB	PE = 0.2510333997
SNR = 4 dB	PE = 0.1645001572
SNR = 6 dB	PE = $0.8239602481\text{x}10^{-1}$
SNR = 8 dB	PE = $0.2649048905\text{x}10^{-1}$
SNR = 10 dB	PE = $0.4117330880\text{x}10^{-2}$
SNR = 12 dB	PE = $0.1970329406\text{x}10^{-3}$
SNR = 14 dB	PE = $0.1431725860\text{x}10^{-5}$
SNR = 16 dB	PE = $0.5263848726\text{x}10^{-9}$
SNR = 18 dB	PE = $0.1764359267\text{x}10^{-14}$

V. CONCLUSION

In this paper, a mathematical model for a receiver used to detect noncoherent FSK signals in the presence of multipath fading is derived This derivation relies on the fact that a bandlimited Guassion process can be expanded in a sinusoidal series. This expansion avoids the problem of evaluating the integral of (20).

With this derivation, a package of computer programs was developed to calculate the numerical values for the probability of error. The curves generated by these programs very closely approximate those presented by Kwon and Shehadeh (7) and (9). This method of calculation shows much improvement over that used by other studying the problem.

APPENDIX

```
C
C       *****************************************************************
C
C
C       PROGRAM PROBERR
C
C       PURPOSE
C           THIS PROGRAM FINDS THE PROBABILITY OF ERROR AS A
C           FUNCTION OF SNR FOR FSK IN THE PRESENCE OF
C           MULTIPATH FADING THIS PROGRAM WAS ADAPTED FROM
C           THE DERIVATION PRESENTED IN A PAPER BY S Y KWON AND
C           N M SHEHADEH:  NONCOHERNET DETECTION OF FSK SIGNALS
C           IN THE PRESENCE OF MULTIPATH FADING, IEEE TRANSACTIONS ON
C           COMMUNICATIONS, JAN., 1978.  THIS PROGRAM CAN ALSO
C           BE USED TO FIND THE PROBABILITY OF ERROR FOR THE CASE
C           OF NO MULTIPATH FADING AS DESCRIBED IN AN EARLIER PAPER
C           BY KWON AND SHEHADEH ANALYSIS OF INCOHERENT FSK SYSTEMS,
C           IEEE TRANSACTIONS ON COMMINICATIONS, NOV., 1975
C
C       DEFINITIONS OF PARAMETERS
C           BT IS THE PRODUCT OF THE MESSAGE BANDWIDTH
C           AND THE BIT TIME
C           BR/B IS THE RATIO OF THE BANDWIDTH OF THE
C           REFLECTED SIGNAL TO THAT OF THE DIRECT SIGNAL
C           GAMMA SQUARED IS THE RATIO OF REFLECTED POWER
C           TO DIRECT POWER
C           N IS THE NUMBER OF DATA POINTS DESIRED
C
C       REMARKS
C           THE PROGRAM FMATRIX MUST BE RUN BEFORE
C           EXECUTION OF THIS PROGRAM TO GENERATE
C           THE DISK FILE DFDA.FP UNLESS THE
C           SUBROUTINE VERSION FMTRX HAS BEEN
C           INCORPORATED
C
C       SUBROUTINES REQUIRED
C           SUBROUTINE MATRIX MUST BE SUPPLIED
C           SUBROUTINE FMTRX IS OPTIONAL
C
C       METHOD
C           EVALUATION IS DONE BY MEANS OF A SERIES EXPANSION OF
C           THE CHARACTERISTIC EQUATION FOR THE OUTPUT OF THE
C           INTEGRATE AND DUMP STAGE ON THE INTERVAL [0,T].
C           THEN BY USE OF A GAUSS QUADRATURE RULE WITH
C           RESPECT TO THE UNIT WEIGHT FUNCTION OVER [-1,1]
C           ONE CAN WRITE THESE CHARACTERISTIC EQUATIONS IN
C           MATRIX FORM.  THESE EQUATIONS CAN THEN BE SOLVED
C           YIELDING THE PROBABILITY OF ERROR.
C
C
C       AUTHOR:  LOREN BAREISS, MARCH, 1979
C
C       *****************************************************************
C
```

```
      DOUBLE PRECISION KA(24),KB(12),RA(24),RB(12),
     @ DFDA(300),DFDB(78),LA(24),LB(12),CA(24),CB(12),
     @ DA(24),DB(12),R(24),H(6),Z(6),PE(10)
      DOUBLE PRECISION GAMS,GAMSDB,KBT,KBRT,BT,BRT,SNR,
     @ PI,BRB,SNRDB,POE,PTE,PROD
      REAL LPE(10)
      INTEGER PAPER
1     FORMAT(3X,'SNRDB=',G12.6,2X,'BT=',G12.6,2X,'BR/B=',G8.2,
     @'GAMSDB=',G8.2)
2     FORMAT(3X,'PE=',G17.10,2X,'LPE=',G11.4)
3     FORMAT(3X)
4     FORMAT(3X,'OUTPUT CHANNEL CODE')
5     FORMAT(3X,'CRT CONSOLE: NC=10, TELETYPE: NC=0')
6     FORMAT(3X,'PAPER #',I1)
7     FORMAT(2X,'PAPER=1, NO MULTIPATH FADING')
8     FORMAT(2X,'PAPER=2, MULTIPATH FADING')
9     FORMAT(1X,'NUMBER OF DATA POINTS DESIRED, 1-10?   ',Z)
      CALL OPEN(0,'$TTO1',3,IERR)
      WRITE(10,4)
      WRITE(10,5)
      ACCEPT'NC=?  ',NC
      WRITE(10,7)
      WRITE(10,8)
      ACCEPT'WHICH PAPER?    ',PAPER
C
C     INPUT VALUES OF BT, BR/B, GAMSDB, N
C
      ACCEPT'BT=?  ',BT
      IF(PAPER.EQ.1)GO TO 11
      ACCEPT'BR/B=?  ',BRB
      ACCEPT'GAMMA SQUARED IN DB=?  ',GAMSDB
11    CONTINUE
      WRITE(10,9)
      ACCEPT N
C
C     ASSIGN VALUES OF H(I) AND Z(I).  THESE VALUES ARE THE WEIGHTS
C     AND ABCISSAS OF A GAUSS QUADRATURE RULE WITH RESPECT TO
C     THE UNIT WEIGHT FUNCTION OVER THE INTERVAL [-1,1].
C
      H(1)=0.249147045813402785D0
      H(2)=0.233492536538354809D0
      H(3)=0.203167426723065922D0
      H(4)=0.160078328543346226D0
      H(5)=0.106939325995318431D0
      H(6)=0.471753363865118272D-1
      Z(1)=0.125233408511468915D0
      Z(2)=0.367831498998180194D0
      Z(3)=0.587317954286617447D0
      Z(4)=0.769902674194304687D0
      Z(5)=0.904117256370474857D0
      Z(6)=0.981560634246719251D0
```

```
C
C       INITIONALIZATION
C
        M=6
        SNRDB=-2.0D0
        BRT=BRB*BT
        MT4=4*M
        MT2=2*M
        GAMS=10.0D0**(GAMSDB/10.0D0)
        PI=3.14159265358979323 8D0
        KBT=2.0D0*PI*BT
        KBRT=KBT*BRB
C
C       SET UP CA AND CB MATRICES  THESE VALUES COME FROM
C       EQUATION 14
C
        DO 10 I=1,M
        CA(I)=(DSIN(KBT*Z(I)))/(KBT*Z(I))
        CA(I+M)=(1.0D0-DCOS(KBT*Z(I)))/(KBT*Z(I))
        CA(I+2*M)=(DSIN(KBRT*Z(I)))/(KBRT*Z(I))
        CA(I+3*M)=(1.0D0-DCOS(KBRT*Z(I)))/(KBRT*Z(I))
        CB(I)=CA(I)
        CB(I+M)=CA(I+M)
10      CONTINUE
C
C       GENERATE VALUES FOR SNR
C
        DO 180 IK=1,N
        SNRDB=SNRDB+2.0D0
        SNR=10.0D0**(SNRDB/10.0D0)
        IF(IK.EQ.1)GO TO 20
        IF(PAPER.EQ.1)GO TO 140
20      CONTINUE
C
C       READ FA MATRIX FROM DISK UNDER THE NAME DFDA
C       FA IS SET UP IN THE PROGRAM FMATRIX FROM EQUATION 13
C       THIS CAN BE AVOIDED BY USING THE SUBROUTINE
C       FMTRX, REMOVING THE C FROM THE FOLLOWING
C       STATEMENT AND DELETING STATEMENTS 30,40,50
C       CALL FMTRX(BT,BRB,DFDA)
C
30      CALL OPEN(1,'DFDA.FP',1,IERR,300*8)
40      CALL READR(1,0,DFDA,1,IERR)
50      CALL CLOSE(1,IERR)
C
C       THE FB MATRIX IS MADE FROM THE UPPER LEFT
C       QUADRANT OF THE FA MATRIX. CALL IT DFDB
C
        NR=0
        DO 60 I=1,MT2
        DO 60 J=1,I
        NR=NR+1
        DFDB(NR)=DFDA(NR)
60      CONTINUE
```

```
C
C       SET UP DA AND DB ACCORDING TO EQUATION 16. DA AND DB
C       ARE DIAGONAL MATRICES WRITTEN IN STANDARD VECTOR FORM.
C
        DO 70 I=1,M
        DA(I)=DSQRT(H(I))
        DA(I+M)=DA(I)
        DB(I)=DA(I)
        DB(I+M)=DA(I)
        DA(I+2*M)=DSQRT(H(I)*SNR/(2*GAMS*BT))
        DA(I+3*M)=DA(I+2*M)
70      CONTINUE
        IF(PAPER.EQ.1)GO TO 80
C
C       FIND THE EIGENVALUES AND EIGENVECTORS OF THE PRODUCT
C       OF DA*DFDA*DA. CALL THE EIGENVALUES LA AND USE
C       THESE EIGENVALUES TO FIND THE KA'S. MULTIPLY THE
C       NORMALIZED EIGENVECTOR MATRIX BY THE PRODUCT OF
C       CA AND DA TO FIND THE RA'S ACCORDING TO EQUATION 18
C
C
C
        CALL MATRIX(DA,DFDA,CA,RA,LA,KA,MT4)
80      CONTINUE
C
C
C       FIND THE EIGENVALUES AND EIGENVECTORS OF THE
C       PRODUCT OF DB*DFDB*DB. CALL THE EIGENVALUES LB AND
C       USE THESE EIGENVALUES TO FIND THE KB'S. MULTIPLY
C       THE NORMALIZED EIGENVECTOR MATRIX BY THE PRODUCT
C       OF CB AND DB TO FIND THE RB'S ACCORDING TO EQUATION 18
C
C
C
        CALL MATRIX(DB,DFDB,CB,RB,LB,KB,MT2)
C
C
C       FIND PROBABILITY OF ERROR PE FOR THE CASE OF MULTIPATH
C       FADING THIS COMES FROM EQUATIONS 36,37 AND 38
C
        IF(PAPER.EQ.1)GO TO 130
        POE=0.0D0
        DO 100 I=1,MT2
        PROD=1.0D0
            DO 90 J=1,MT4
            PROD=PROD*((DEXP(((-(RA(J)**2))*SNR)/(2.0D0*BT*(LA(J)+LB(I))
     @      *LA(J))))/(1.0D0+LA(J)/LB(I)))
90          CONTINUE
        POE=POE+KB(I)*PROD
100     CONTINUE
```

```
      PTE=0.0D0
      DO 120 I=1,MT4
      PROD=1.0D0
          DO 110 J=1,MT2
          PROD=PROD*((DEXP(((-(RB(J)**2))*SNR)/(2.0D0*BT*(LA(I)+LB(J))
     @    *LB(J))))/(1.0D0+LB(J)/LA(I)))
110       CONTINUE
      PTE=PTE+KA(I)*PROD
120   CONTINUE
      PE(IK)=POE/2.0D0+PTE/2.0D0
130   CONTINUE
140   CONTINUE
C
C     FIND PROBABILTIY OF ERROR PE FOR THE CASE OF NO FADING
C     USING EQUATION 55
C
      IF(PAPER.EQ.2)GO TO 170
      POE=0.0D0
      DO 160 I=1,MT2
      PROD=1.0D0
          DO 150 J=1,MT2
          PROD=PROD*((DEXP(((-(RB(J)**2))*SNR)/(2.0D0*BT*(LB(I)+
     @    LB(J))*LB(J))))/(1.0D0+LB(J)/LB(I)))
150           CONTINUE
      POE=POE+KB(I)*PROD
160   CONTINUE
      PE(IK)=POE
170   CONTINUE
      LPE(IK)=SNGL(DLOG10(PE(IK)))
C
C     OUTPUT VALUES FOR SNR AND PE TO DESIGNATED CHANNEL, NC
C
      WRITE(NC,1) SNRDB,BT,BRB,GAMSDB
      WRITE(NC,2) PE(IK),LPE(IK)
      WRITE(NC,3)
      WRITE(NC,3)
180   CONTINUE
C
C     WRITE LOG10(PE) TO DISK
C
      CALL OPEN(1,'LPE.FP',3,IERRO,10*4)
      CALL WRITR(1,0,LPE,1,IERRO)
      CALL CLOSE(1,IERRO)
      CALL CLOSE(0,IERR)
      STOP
      END
```

```
C
C       ************************************************************
C
C       PROGRAM FMATRIX
C
C       PURPOSE
C           THIS PROGRAM SETS UP THE FA MATRIX USED
C           IN THE PROGRAM PROBERR
C           EACH MATRIX ELEMENT IS CALCULATED FROM A CLOSED
C           INTEGRAL OVER [0,1] OF A COMBINATION OF SINE
C           AND COSINE ARGUMENTS ACCORDING TO EQUATION 13
C
C       DEFINITION OF PARAMETERS
C           BT IS THE PRODUCT OF THE MESSAGE BANDWIDTH
C           AND THE BIT TIME
C           BR/B IS THE RATIO OF THE BANDWIDTH OF THE
C           REFLECTED SIGNAL TO THAT OF THE DIRECT SIGNAL
C
C       SUBROUTINES REQUIRED
C           NONE
C
C       REMARKS
C           THIS PROGRAM CAN BE MADE A SUBROUTINE BY
C           REMOVING THE C FROM THE NEXT LINE
C       SUBROUTINE FMTRX(BT,BRB,DFDA)
C           ALSO YOU MAY REMOVE STATEMENTS 1 THROUGH 8
C
C
C       AUTHOR:  LOREN BAREISS, MARCH, 1979
C
C
C       ************************************************************
C
        DOUBLE PRECISION FA(24,24),Z(6),DFDA(300)
        DOUBLE PRECISION KBT,KBRT,BT,BRB,PI
C
C       INPUT VALUES OF BT,BR/B,M
C
1       ACCEPT"BT=?",BT
2       ACCEPT"BR/B=?",BRB
C
C       ASSIGN VALUES OF Z(I).  THESE VALUES ARE THE
C       ABCISSAS OF A GAUSS QUADRATURE RULE WITH RESPECT TO
C       THE UNIT WEIGHT FUNCTION OVER THE INTERVAL [-1,1].
C
        Z(1)=.1252334085114689150O
        Z(2)=.367831498998180194D0
        Z(3)=.587317954286617447D0
        Z(4)=.769902674194304687D0
        Z(5)=.904117256370474857D0
        Z(6)=.981560634246719251D0
```

```
C
C       ****************************************************************
C
C       SUBROUTINE MATRIX
C
C       PURPOSE
C           TO FIND THE EIGENVALUES AND THE NORMALIZED
C           EIGENVECTOR MATRIX OF THE PRODUCT OF
C           THE INPUT MATRICES D AND DFD AND TO
C           MULTIPLY THE NORMALIZED EIGENVECTOR MATRIX
C           BY THE PRODUCT OF THE INPUT MATRICES C AND D.
C
C       USAGE
C           CALL MATRIX(D,DFD,C,R,L,K,N)
C
C       DESCRIPTIONS OF PARAMETERS
C           D, DFD AND C ARE THE INPUT MATRICES
C           R, L AND K ARE THE OUTPUT VECTORS
C           N IS THE LENGTH OF THESE VECTORS
C
C       REMARKS
C           THE ORIGINAL MATRIX DFD IS DESTROYED
C           IN COMPUTATION
C
C       SUBROUTINES REQUIRED
C           SUBROUTINE DEIG MUST BE SUPPLIED
C
C       AUTHOR:  LOREN BAREISS, MARCH, 1979
C
C       ****************************************************************
C
C
        SUBROUTINE MATRIX(D,DFD,C,R,L,K,N)
        DOUBLE PRECISION D(1),DFD(1),M(576),C(1),L(1),R(1),CD(24),K(1)
        DOUBLE PRECISION XJ,PROD
C
C       FIND THE PRODUCT OF D AND DFD AND THEN POSTMULTIPLY BY
C       D AND CALL THE RESULT DFD
C
        KO=0
        DO 20 J=1,N
        DO 10 I=1,J
        KO=KO+1
        DFD(KO)=D(I)*DFD(KO)*D(J)
10      CONTINUE
20      CONTINUE
C
C       FIND THE EIGENVALUES AND EIGENVECTORS OF DFD AND CALL
C       THE EIGENVALUES L AND THE EIGENVECTOR MATRIX M
C
        CALL DEIG(DFD,M,N)
```

```
C
C       PICK OUT EIGENVALUES
C
        KR=0
        LM=0
        DO 30 I=1,N
        LM=LM+I
        KR=KR+1
        L(KR)=DFD(LM)
30      CONTINUE
C
C       FIND THE PRODUCT OF C AND D AND CALL IT CD
C
        DO 40 I=1,N
        CD(I)=C(I)*D(I)
40      CONTINUE
C
C       FIND THE PRODUCT OF CD AND M AND CALL IT R
C
        KR=0
        DO 60 I=1,N
        XJ=0.0D0
                DO 50 J=1,N
                KR=KR+1
                XJ=XJ+CD(J)*M(KR)
50              CONTINUE
        R(I)=XJ
60      CONTINUE
C
C       FIXUP FOR ERRANT EIGENVALUES
C
        DO 90 I=1,N
        IF(L(I).GT.1.0D-20)GO TO 90
        IF(N.EQ.12)GO TO 70
        L(I)=1.0D-18*((1.0D-2)**(I-17))
        GO TO 80
70      CONTINUE
        L(I)=1.0D-18*((1.0D-2)**(I-9))
80      CONTINUE
        R(I)=1.0D-20
90      CONTINUE
C
C       USE THESE L'S TO FIND THE K'S THROUGH A HEAVISIDE EXPANSION
C
        DO 110 I=1,N
        PROD=1.0D0
                DO 100 II=1,N
                IF(II.EQ.I)GO TO 100
                PROD=PROD*(1.0D0/(1.0D0-L(II)/L(I)))
100             CONTINUE
        K(I)=PROD
110     CONTINUE
        RETURN
        END
```

```
C
C       INITIALIZATION
C
        PI=3.141592653589793238D0
        KBT=2.0D0*PI*BT
        KBRT=KBT*BRB
        M=6
C
C       SET UP FA MATRIX
C
C
C       THERE ARE 4 COMBINATIONS OF THE PARAMETERS
C       BR/B AND Z NEEDED TO FIND THE FA MATRIX.
C       CASE1--BR/B.EQ.1,Z(J).EQ.Z(I)
C       CASE2--BR/B.EQ.1,Z(I).NE.Z(J)
C       CASE3--BR/B.NE.1,Z(I).EQ.Z(J)
C       CASE4--BR/B.NE.1,Z(I).NE.Z(J)
C
C
C       LOOP1 TAKES CARE OF CASES 2,4
C
        DO 20 I=1,M
        DO 10 J=1,M
        IF(I.EQ.J)GO TO 10
        FA(I,J)=(DSIN(KBT*(Z(I)-Z(J))))/(2.0D0*KBT*(Z(I)-Z(J)))
     @+(DSIN(KBT*(Z(I)+Z(J))))/(2.0D0*KBT*(Z(I)+Z(J)))
        FA(I,J+M)=(1.0D0-DCOS(KBT*(Z(I)+Z(J))))/(2.0D0*KBT*(Z(I)+Z(J)))
     @-(1.0D0-DCOS(KBT*(Z(I)-Z(J))))/(2.0D0*KBT*(Z(I)-Z(J)))
        FA(I,J+2*M)=(DSIN(KBT*Z(I)-KBRT*Z(J)))/(2.0D0*(KBT*Z(I)
     @-KBRT*Z(J)))+(DSIN(KBT*Z(I)+KBRT*Z(J)))/(2.0D0*(KBT*Z(I)
     @+KBRT*Z(J)))
        FA(I+M,J+2*M)=(1.0D0-DCOS(KBT*Z(I)-KBRT*Z(J)))/
     @(2.0D0*(KBT*Z(I)-KBRT*Z(J)))+(1.0D0-DCOS(KBT*Z(I)
     @+KBRT*Z(J)))/(2.0D0*KBT*Z(I)+KBRT*Z(J))
        FA(I+M,J+M)=(DSIN(KBT*(Z(I)-Z(J))))/(2.0D0*KBT*(Z(I)-Z(J)))
     @-(DSIN(KBT*(Z(I)+Z(J))))/(2.0D0*KBT*(Z(I)+Z(J)))
        FA(I,J+3*M)=(1.0D0-DCOS(KBT*Z(I)+KBRT*Z(J)))
     @/(2.0D0*(KBT*Z(I)+KBRT*Z(J)))-(1.0D0-DCOS(KBT*Z(I)-KBRT*Z(J)))
     @/(2.0D0*(KBT*Z(I)-KBRT*Z(J)))
        FA(I+M,J+3*M)=(DSIN(KBT*Z(I)-KBRT*Z(J)))
     @/(2.0D0*(KBT*Z(I)-KBRT*Z(J)))-(DSIN(KBT*Z(I)+KBRT*Z(J)))
     @/(2.0D0*(KBT*Z(I)+KBRT*Z(J)))
        FA(I+2*M,J+2*M)=(DSIN(KBRT*(Z(I)-Z(J))))/(2.0D0*KBRT*(Z(I)-Z(J)))
     @+(DSIN(KBRT*(Z(I)+Z(J))))/(2.0D0*KBRT*(Z(I)+Z(J)))
        FA(I+2*M,J+3*M)=(1.0D0-DCOS(KBRT*(Z(J)+Z(I))))
     @/(2.0D0*KBRT*(Z(J)+Z(I)))-(1.0D0-DCOS(KBRT*(Z(J)-Z(I))))
     @/(2.0D0*KBRT*(Z(J)-Z(I)))
        FA(I+3*M,J+3*M)=(DSIN(KBRT*(Z(I)-Z(J))))/(2.0D0*KBRT*(Z(I)-Z(J)))
     @-(DSIN(KBRT*(Z(I)+Z(J))))/(2.0D0*KBRT*(Z(I)+Z(J)))
10      CONTINUE
20      CONTINUE
```

```
C
C       LOOP2 TAKES CARE OF CASE 1
C
        IF(BRB.NE.1.0D0)GO TO 40
        DO 30 I=1,M
        DO 30 J=1,M
        IF(I.NE.J)GO TO 30
        FA(I,J)=.5D0+(DSIN(2.0D0*KBT*Z(I)))/(4.0D0*KBT*Z(I))
        FA(I,J+M)=((DSIN(KBT*Z(I)))**2)/(2.0D0*KBT*Z(I))
        FA(I,J+2*M)=.5D0+(DSIN(2.0D0*KBT*Z(I)))/(4.0D0*KBT*Z(I))
        FA(I,J+3*M)=((DSIN(KBT*Z(I)))**2)/(2.0D0*KBT*Z(I))
        FA(I+M,J+2*M)=((DSIN(KBT*Z(I)))**2)/(2.0D0*KBT*Z(I))
        FA(I+M,J+3*M)=.5D0-(DSIN(2.0D0*KBT*Z(I)))/(4.0D0*KBT*Z(I))
        FA(I+M,J+M)=.5D0-(DSIN(2.0D0*KBT*Z(I)))/(4.0D0*KBT*Z(I))
        FA(I+2*M,J+2*M)=.5D0+(DSIN(2.0D0*KBT*Z(I)))/(4.0D0*KBT*Z(I))
        FA(I+2*M,J+3*M)=((DSIN(KBT*Z(I)))**2)/(2.0D0*KBT*Z(I))
        FA(I+3*M,J+3*M)=.5D0-(DSIN(2.0D0*KBT*Z(I)))/(4.0D0*KBT*Z(I))
30      CONTINUE
40      CONTINUE
C
C       LOOP3 TAKES CARE OF CASE3
C
        IF(BRB.EQ.1.0D0)GO TO 60
        DO 50 I=1,M
        DO 50 J=1,M
        IF(I.NE.J)GO TO 50
        FA(I,J)=.5D0+(DSIN(2.0D0*KBT*Z(I)))/(4.0D0*KBT*Z(I))
        FA(I,J+M)=((DSIN(KBT*Z(I)))**2)/(2.0D0*KBT*Z(I))
        FA(I,J+2*M)=(DSIN(Z(I)*(KBT-KBRT)))/(2.0D0*Z(I)*(KBT-KBRT))+
     @(DSIN(Z(I)*(KBT+KBRT)))/(2.0D0*Z(I)*(KBT+KBRT))
        FA(I,J+3*M)=(1.0D0-DCOS(Z(I)*(KBT+KBRT)))/(2.0D0*Z(I)*(KBT
     @+KBT))-(1.0D0-DCOS(Z(I)*(KBT-KBRT)))/(2.0D0*Z(I)*(KBT-KBRT))
        FA(I+M,J+M)=.5D0-(DSIN(2.0D0*KBT*Z(I)))/(4.0D0*KBT*Z(I))
        FA(I+M,J+2*M)=(1.0D0-DCOS(Z(I)*(KBRT+KBT)))/
     @(2.0D0*Z(I)*(KBRT+KBT))-(1.0D0-DCOS(Z(I)*(KBRT-KBT)))/
     @(2.0D0*Z(I)*(KBRT-KBT))
        FA(I+M,J+3*M)=(DSIN(Z(I)*(KBT-KBRT)))/(2.0D0*Z(I)*(KBT-KBRT))-
     @(DSIN(Z(I)*(KBT+KBRT)))/(2.0D0*Z(I)*(KBT+KBRT))
        FA(I+2*M,J+2*M)=.5D0+(DSIN(2.0D0*KBRT*Z(I)))/(4.0D0*KBRT*Z(I))
        FA(I+2*M,J+3*M)=((DSIN(KBRT*Z(I)))**2)/(2.0D0*KBRT*Z(I))
        FA(I+3*M,J+3*M)=.5D0-(DSIN(2.0D0*KBRT*Z(I)))/(4.0D0*KBRT*Z(I))
50      CONTINUE
60      CONTINUE
C
C       PUT FA INTO THE CUSTOMARY VECTOR FORM FOR A
C       SYMMETRIC MATRIX AND CALL IT DFDA
C
        MT4=4*M
        NR=0
        DO 70 J=1,MT4
        DO 70 I=1,J
        NR=NR+1
        DFDA(NR)=FA(I,J)
70      CONTINUE
```

```
C3
C4       WRITE DFDA TO DISC
C5
6        CALL OPEN(0,'DFDA.FP',3,IERRO,300*8)
7        CALL WRITR(0,0,DFDA,1,IERRO)
8        CALL CLOSE(0,IERRO)
         STOP
         END
```

```
C
C     ******************************************************************
C
C        SUBROUTINE DEIG
C
C        PURPOSE
C           COMPUTE THE DOUBLE PRECISION EIGENVALUES AND EIGENVECTORS
C           OF A REAL SYMMETRIC MATRIX
C
C        USAGE
C           CALL DEIG(A,R,N)
C
C        DESCRIPTION OF PARAMETERS
C           A - ORIGINAL MATRIX (SYMMETRIC), DESTROYED IN COMPUTATION.
C               RESULTANT EIGENVALUES ARE DEVELOPED IN DIAGONAL OF
C               MATRIX A IN DESCENDING ORDER.
C           R - RESULTANT MATRIX OF EIGENVECTORS (STORED COLUMNWISE,
C               IN SAME SEQUENCE AS EIGENVALUES)
C           N - ORDER OF MATRICES A AND R
C
C        REMARKS
C           ORIGINAL MATRIX A MUST BE REAL SYMMETRIC
C           MATRIX A CANNOT BE IN THE SAME LOCATION AS MATRIX R
C
C        SUBROUTINES REQUIRED
C           NONE
C
C        METHOD
C           DIAGONALIZATION METHOD ORIGINATED BY JACOBI AND ADAPTED
C           BY VON NEUMANN FOR LARGE COMPUTERS AS FOUND IN 'MATHEMATICAL
C           METHODS FOR DIGITAL COMPUTERS', EDITED BY A. RALSTON AND
C           H.S. WILF, JOHN WILEY AND SONS, NEW YORK, 1962, CHAPTER 7
C
C     ******************************************************************
C
      SUBROUTINE DEIG(A,R,N)
      DIMENSION A(1),R(1)
      DOUBLE PRECISION A,R,ANORM,ANRMX,THR,X,Y,SINX,SINX2,COSX,
     @                 COSX2,SINCS,RANGE
C
C        GENERATE IDENTITY MATRIX
C
      MV=0
    5 RANGE=1.0D-25
      IF(MV-1) 10,25,10
   10 IQ=-N
      DO 20 J=1,N
      IQ=IQ+N
      DO 20 I=1,N
      IJ=IQ+I
      R(IJ)=0.0D0
      IF(I-J) 20,15,20
   15 R(IJ)=1.0D0
   20 CONTINUE
```

```
C
C        COMPUTE INITIAL AND FINAL NORMS (ANORM AND ANORMX)
C
   25 ANORM=0.0D0
      DO 35 I=1,N
      DO 35 J=I,N
      IF(I-J) 30,35,30
   30 IA=I+(J*J-J)/2
      ANORM=ANORM+A(IA)*A(IA)
   35 CONTINUE
      IF(ANORM) 165,165,40
   40 ANORM=1.414D0*DSQRT(ANORM)
      ANRMX=ANORM*RANGE/DFLOAT(N)
C
C        INITIALIZE INDICATORS AND COMPUTE THRESHOLD, THR
C
      IND=0
      THR=ANORM
   45 THR=THR/DFLOAT(N)
   50 L=1
   55 M=L+1
C
C        COMPUTE SIN AND COS
C
   60 MQ=(M*M-M)/2
      LQ=(L*L-L)/2
      LM=L+MQ
   62 IF(DABS(A(LM))-THR) 130,65,65
   65 IND=1
      LL=L+LQ
      MM=M+MQ
      X=0.5D0*(A(LL)-A(MM))
   68 Y=-A(LM)/DSQRT(A(LM)*A(LM)+X*X)
      IF(X) 70,75,75
   70 Y=-Y
   75 SINX=Y/DSQRT(2.0D0*(1.0D0+(DSQRT(1.0D0-Y*Y))))
      SINX2=SINX*SINX
   78 COSX=DSQRT(1.0D0-SINX2)
      COSX2=COSX*COSX
      SINCS =SINX*COSX
C
C        ROTATE L AND M COLUMNS
C
      ILQ=N*(L-1)
      IMQ=N*(M-1)
      DO 125 I=1,N
      IQ=(I*I-I)/2
      IF(I-L) 80,115,80
   80 IF(I-M) 85,115,90
   85 IM=I+MQ
      GO TO 95
   90 IM=M+IQ
   95 IF(I-L) 100,105,105
  100 IL=I+LQ
```

```
      GO TO 110
  105 IL=L+IQ
  110 X=A(IL)*COSX-A(IM)*SINX
      A(IM)=A(IL)*SINX+A(IM)*COSX
      A(IL)=X
  115 IF(MV-1) 120,125,120
  120 ILR=ILQ+I
      IMR=IMQ+I
      X=R(ILR)*COSX-R(IMR)*SINX
      R(IMR)=R(ILR)*SINX+R(IMR)*COSX
      R(ILR)=X
  125 CONTINUE
      X=2.0D0*A(LM)*SINCS
      Y=A(LL)*COSX2+A(MM)*SINX2-X
      X=A(LL)*SINX2+A(MM)*COSX2+X
      A(LM)=(A(LL)-A(MM))*SINCS+A(LM)*(COSX2-SINX2)
      A(LL)=Y
      A(MM)=X
C
C         TESTS FOR COMPLETION
C
C         TEST FOR M = LAST COLUMN
C
  130 IF(M-N) 135,140,135
  135 M=M+1
      GO TO 60
C
C         TEST FOR L = SECOND FROM LAST COLUMN
C
  140 IF(L-(N-1)) 145,150,145
  145 L=L+1
      GO TO 55
  150 IF(IND-1) 160,155,160
  155 IND=0
      GO TO 50
C
C         COMPARE THRESHOLD WITH FINAL NORM
C
  160 IF(THR-ANRMX) 165,165,45
C
C         SORT EIGENVALUES AND EIGENVECORS
C
  165 IQ=-N
      DO 185 I=1,N
      IQ=IQ+N
      LL=I+(I*I-I)/2
      JQ=N*(I-2)
      DO 185 J=I,N
      JQ=JQ+N
      MM=J+(J*J-J)/2
      IF(A(LL)-A(MM)) 170,185,185
  170 X=A(LL)
      A(LL)=A(MM)
      A(MM)=X
```

```
      IF(MV-1) 175,185,175
  175 DO 180 K=1,N
      ILR=IQ+K
      IMR=JQ+K
      X=R(ILR)
      R(ILR)=R(IMR)
  180 R(IMR)=X
  185 CONTINUE
      RETURN
      END
```

BIBLIOGRAPHY

1. H. D. Chadwick, "The error probability of a wide band FSK receiver in the prasence of multipath fading," IEEE Trans. Commun. Tech., Vol. COM-19, pp. 699-707, Oct. 1971.

2. A. B. Glenn, "Analysis of noncoherent FSK systems with large ratios of frequency uncertainties to information rates," RCA Rev., Vol. 27, pp. 272-314, June 1966.

3. D. W. Boyd, "Approximate analysis of a frequency-shift keyed system with uncertainty in the carrier frequency," Jet. Propul. Lab., SPS 37-41, Vol. 4, pp. 233-239, Oct. 1966.

4. M. C. Austin, "Wide-band frequency-shift keyed received performance in the presence of intersymbol interference," IEEE Trans. Commun., Vol. COM-23, pp. 453-458, April 1975.

5. L. B. Milstein and M. C. Austin, "Performance of noncoherent FSK and AM-FSK systems with postdetection filtering," IEEE Trans. Commun., Vol. COM-23, pp. 1300-1306, Nov. 1975.

6. L. Schuchman, "Wide-band Detection of FSK transmissions in a Three Components Two Path Channel," IEEE Trans. Commun. Tech., Vol. COM-18, pp. 319-332, Aug. 1970.

7. S. Y. Kwon and N. M. Shehadeh, "Analysis of Incoherent Frequency-Shift Keyed Systems," IEEE Trans. Commun., Vol. COM-23, pp. 1331-1339, Nov. 1975.

8. A. M. Yaglom, An Introduction to the Theory of Stationary Random Functions. Englewood Cliffs, N.J.: Prentice-Hall, 1962, pp. 31-36.

9. S. Y. Kwon and N. M. Shehadeh, "Noncoherent detection of FSK Signal in the Presence of Multipath Fading," IEEE Trans. Commun. Vol. COM-26, pp. 164-168, Jan. 1978.

10. A. H. Stroud and D. Secrest, Gaussion Quadrature Formulas. Prentice-Hall, 1966, p. 100.

Development of a Computer Program
to Simulate a Noncoherent FSK System
in the Presence of Multipath Fading

by

LOREN D. BAREISS, JR.

B.S., Kansas State University, 1977

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1979

ABSTRACT

The derivation of a mathematical model for a noncoherent frequency shift keyed system by Kwon and Shehadeh is used to develop a computer program to generate numerical values for the probability of error. These values of probability of error are plotted as a function of the system parameters BT, signal to noise ratio, fading bandwidth and direct to reflected signal power ratio.