THE INFLUENCE OF FILTER SELECTION ON DETECTION PROBABILITY FOR RECEIVERS USING SQUARE-LAW DETECTION, A GENERAL APPROACH

by

CONNIE ADAMS

B.A., Ottawa University, 1975

A MASTER'S THESIS

Submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1982

Approved by:

Donald R. Hummels
Major Professor

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

This document is concerned with the development of numerical methods for finding the probability of detection for a very general square-law detector system. The system consists of a bandpass pre-filter followed by a square-law envelope detector and a low-pass video filter.

This problem has been dealt with previously by Kac and Siegert [1] and by Marcum [2]. The results found by Kac and Siegert require eigenvalues and eigenvectors of an integral equation. The final form for the probability of detection does not allow for easy solution of the eigenvalue problem, i.e., there is not a general solution to finding the eigenvalues, and the method for doing so must be found on a case-by-case basis. Marcum's development deals with specific input signals and is not readily adaptable to various filter transfer functions or input signal configurations.

The development found in this paper incorporates a series representation of noise [Yaglom, 3] that allows for matrix formulation of noise only or signal plus noise cases. The matrix representation affords solution of the eigenvalue problem, and leads to numerical solutions for the probability of detection for the signal plus noise case, or the probability of false alarm for the noise only case.

The results of this development are general in the sense that they are applicable to arbitrary pre- and post-filter transfer functions as well as to arbitrary input signal formats, and the results may be used for systems where the ratio of RF bandwidth to video bandwidth is large. Thus, the results of the development found here are useful in solving some traditionally difficult detection problems.

II. A MODEL FOR THE SQUARE-LAW RECEIVER

The Physical System

A simplified block diagram of the receiver under consideration is shown in Figure 1. It consists of a bandpass pre-filter having transfer function H(f) followed by a square-law envelope detector and low-pass filter with transfer function G(f). The input to the receiver is a signal of interest, s(t), plus a stationary bandpass Gaussian noise process, n(t). All of the results presented herein are for the case where n(t) has a flat power spectrum given by

$$S_n(f) = \begin{cases} N_o/2 \ , & f_c - B_n \leq |f| \leq f_c + B_n \\ 0 \ , & \text{elsewhere} \end{cases} \tag{1}$$

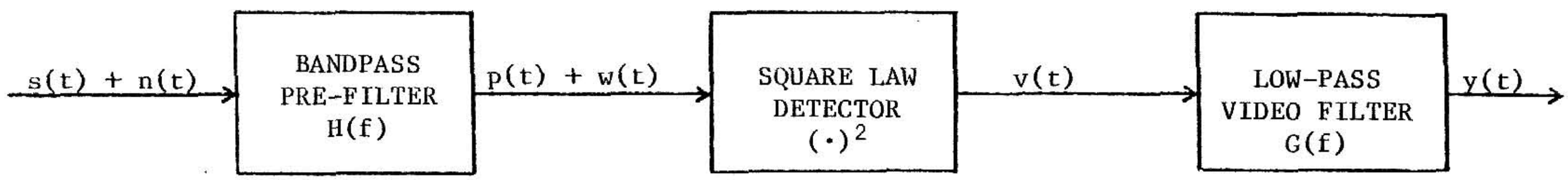
where f_c denotes the center frequency of the pre-filter and $2B_n$ is the bandwidth of the noise process, typically somewhat wider than the bandwidth of the pre-filter so that for practical purposes the input noise appears to be white. There are no restrictions on the pre-filter or post-filter other than that of linearity.

An Equivalent Low-Pass Model

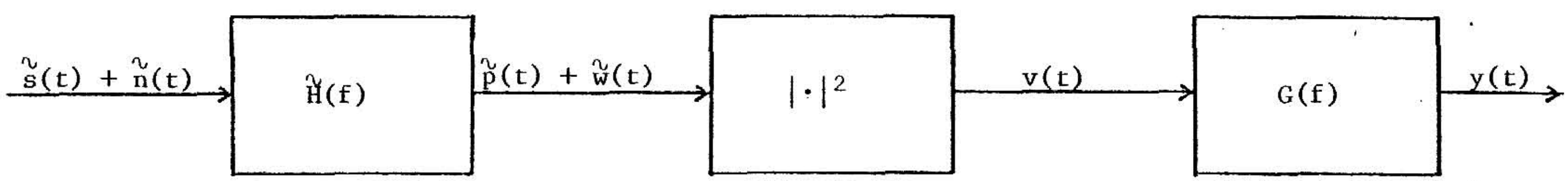
The equivalent low-pass model shown in Figure 1 is developed in the usual way [4] with a signal and its complex envelope related by

$$s(t) = \mathrm{Re}\left\{\tilde{s}(t)\ e^{j\omega_c t}\right\} \tag{2}$$

where $\tilde{s}(t)$ is the complex envelope of s(t) and $\omega_c = 2\pi f_c$ is the center frequency or carrier frequency of the signal and is taken as the center frequency of the bandpass pre-filter for convenience. The complex enve-



a) Physical System



b) Equivalent Low-Pass Model

Figure 1. Block Diagram of Receiver Model

lope, $\tilde{n}(t)$, of the input noise process is a stationary complex Gaussian process. The power spectrum of $\tilde{n}(t)$ is related to that of $n(t)$ by the equation

$$S_n(f) = \frac{1}{4} S_{\tilde{n}}(f - f_c) + \frac{1}{4} S_{\tilde{n}}(-f - f_c) \; . \tag{3}$$

Thus the power spectrum for $\tilde{n}(t)$ used here is

$$S_{\tilde{n}}(f) = \begin{cases} 2N_o \; , & |f| \leq B_n \\ 0 \; , & |f| > B_n \end{cases} \tag{4}$$

The sum of signal and noise envelopes, $\tilde{s}(t) + \tilde{n}(t)$, serves as the input to the low-pass equivalent of the pre-filter. The transfer function of the pre-filter and that of its low-pass equivalent, $\tilde{H}(f)$, are related via

$$H(f) = \tilde{H}(f - f_c) + \tilde{H}^*(-f - f_c) \; . \tag{5}$$

The complex envelope of the signal at the pre-filter output is a sum, $\tilde{p}(t) + \tilde{w}(t)$, of filtered signal and noise envelopes respectively. The action of the square law detector is to produce a voltage $v(t)$ proportional to the magnitude squared of this complex envelope, i.e.,

$$v(t) = |\tilde{p}(t) + \tilde{w}(t)|^2 \, . \tag{6}$$

The output $y(t)$ is obtained by passing $v(t)$ through a low pass filter with transfer function $G(f)$.

A Matrix Formulation for the Output

The determination of the probability distribution for the output $y(t)$ depends upon the development of a matrix representation for $y(t)$. We begin by expanding (6) as

$$v(t) = |\tilde{p}(t)|^2 + 2\mathrm{Re}\{\tilde{p}(t)\ \tilde{w}^*(t)\} + |\tilde{w}(t)|^2 \qquad (7)$$

where * denotes the complex conjugate.

If the subscripts r and i are used to denote real and imaginary parts respectively, v(t) may be further expanded to yield

$$\begin{aligned} v(t) &= \tilde{p}_r^2(t) + 2\tilde{p}_r(t)\ \tilde{w}_r(t) + \tilde{w}_r^2(t) \\ &\quad + \tilde{p}_i^2(t) + 2\tilde{p}_i(t)\ \tilde{w}_i(t) + \tilde{w}_i^2(t). \end{aligned} \qquad (8)$$

At this point a noise model described in Appendix A is introduced. Our objective is to find tractable expansions for the noise terms in (8). Let the complex envelope of the input noise be represented by

$$\tilde{n}(t) = \sum_{k=1}^{K} (a_k e^{j\lambda_k t} + b_k e^{-j\lambda_k t}) \qquad (9)$$

where the a_k and b_k are statistically independent complex Gaussian random variables and the λ_k are a set of frequencies selected in accordance with the procedure outlined in Appendix A. The nature of the model is such that the real and imaginary parts of the a_k and b_k are zero-mean and statistically independent with variances

$$E\{a_{r_k}^2\} = E\{a_{i_k}^2\} = R_k/2 \qquad (10)$$

and

$$E\{b_{r_k}^2\} = E\{b_{i_k}^2\} = R_k/2 \qquad (11)$$

It is shown in Appendix A that the pairs of frequencies and variances (λ_k, R_k) may be determined from a tabulated Gauss quadrature rule (GQR) with respect to the unit weighting function on the interval [-1,1].

This feature makes the model particularly attractive for numerical work. If the GQR pairs obtained from tables are designated as (γ_k, ν_k), then λ_k and R_k are found from

$$\lambda_k = 2\pi B_n \nu_k \tag{12}$$

and

$$R_k = 2N_o B_n \gamma_k \quad . \tag{13}$$

Further expansion of (8) requires expressions for the real processes $\tilde{w}_r(t)$ and $\tilde{w}_i(t)$. An expression for the complex process $\tilde{w}(t)$ is readily obtained from (9) as

$$\tilde{w}(t) = \sum_{k=1}^{K} \left[a_k\tilde{H}(\lambda_k)e^{j\lambda_k t} + b_k\tilde{H}(-\lambda_k)e^{-j\lambda_k t}\right] \tag{14}$$

The real part of $\tilde{w}(t)$ is then given by

$$\tilde{w}_r(t) = \mathrm{Re}\left\{\sum_{k=1}^{K} (a_k\tilde{H}(\lambda_k)e^{j\lambda_k t} + b_k\tilde{H}^*(\lambda_k)e^{-j\lambda_k t})\right\} \tag{15}$$

where in writing (15) we have assumed that H(f) is symetrical about f_c so that $\tilde{H}(-\lambda) = \tilde{H}^*(\lambda)$. Observing that the real part of a sum is the sum of the real parts, we have

$$\tilde{w}_r(t) = \sum_{k=1}^{K} \mathrm{Re}\left\{a_k\tilde{H}(\lambda_k)e^{j\lambda_k t} + b_k\tilde{H}^*(\lambda_k)e^{-j\lambda_k t}\right\} \tag{16}$$

One may verify after some manipulation that the indicated real part in (16) may be written in the form

$$\mathrm{Re}\left\{a_k \tilde{H}(\lambda_k) e^{j\lambda_k t} + b_k \tilde{H}^*(\lambda_k) e^{-j\lambda_k t}\right\}$$

$$= (a_{r_k} + b_{r_k}) \frac{\tilde{H}(\lambda_k) e^{j\lambda_k t} + \tilde{H}^*(\lambda_k) e^{-j\lambda_k t}}{2}$$

$$+ (b_{i_k} - a_{i_k}) \frac{\tilde{H}(\lambda_k) e^{j\lambda_k t} - \tilde{H}^*(\lambda_k) e^{-j\lambda_k t}}{2j} \quad . \tag{17}$$

Define a new set of random variables using

$$c_k \triangleq \begin{cases} a_{r_k} + b_{r_k} & , \ k = 1, \ldots , K \\ b_{i_{k-K}} - a_{i_{k-K}} & , \ k = K+1, \ldots , 2K . \end{cases} \tag{18}$$

The c_k are real Gaussian random variables with zero-means and variances given by

$$E\{c_k^2\} = \begin{cases} R_k & , \ k = 1, \ldots , K \\ R_{k-K} & , \ k = K+1, \ldots , 2K . \end{cases} \tag{19}$$

It can be shown that they are also statistically independent.

Now define the real parameter h_k as

$$h_k = \begin{cases} \dfrac{\tilde{H}(\lambda_k) e^{j\lambda_k t} + \tilde{H}^*(\lambda_k) e^{-j\lambda_k t}}{2} & , \ k = 1, \ldots , K \\ \dfrac{\tilde{H}(\lambda_{k-K}) e^{j\lambda_{k-K} t} + \tilde{H}^*(\lambda_{k-K}) e^{-j\lambda_{k-K} t}}{2j} & , \ k = K+1, \ldots, 2K. \end{cases} \tag{20}$$

The real part of $\tilde{w}(t)$ may now be written as

$$\tilde{w}_r(t) = \sum_{k=1}^{2K} c_k h_k \tag{21}$$

and the square of this quantity is readily obtained as

$$\tilde{w}_r^2(t) = \sum_{k=1}^{2K} \sum_{\ell=1}^{2K} c_k\, h_k\, h_\ell\, c_\ell \ . \tag{22}$$

Equations (21) and (22) may be conveniently arranged in matrix notation by defining the vectors

$$c^T = (c_1,\ c_2,\ \ldots,\ c_{2K}) \ , \tag{23}$$

$$h^T = (h_1,\ h_2,\ \ldots,\ h_{2K}) \tag{24}$$

and the real symetric matrix

$$H = \begin{bmatrix} h_1h_1 & h_1h_2 & \cdots & h_1h_{2K} \\ h_2h_1 & h_2h_2 & & \\ \vdots & & \ddots & \\ h_{2K}h_1 & \cdots & & h_{2K}h_{2K} \end{bmatrix} \ . \tag{25}$$

The resulting matrix forms are:

$$\tilde{w}_r(t) = h^T c \tag{26}$$

$$\tilde{w}_r^2(t) = c^T H c \ . \tag{27}$$

Note that all of the system properties and time dependence are imbedded in the vector h and matrix H and that c is a Gaussian random vector with elements that are statistically independent.

Following the same procedure yields similar forms for $\tilde{w}_i(t)$ and $\tilde{w}_i^2(t)$. The details are omitted here, but one may verify the not too surprising result that system properties and time dependence contained in h and H turn out to be the same for this case as for the development leading to (26) and (27). It is necessary; however, to define a new random vector d with elements d_k given by

$$d_k = \begin{cases} a_{i_k} + b_{i_k} & , k = 1, \ldots, K \\ a_{r_{k-K}} - b_{r_{k-K}} & , k = K+1, \ldots, 2K . \end{cases} \tag{28}$$

The d_k are real Gaussian random variables with zero-means and variances given by

$$E\{d_k^2\} = \begin{cases} R_k & , k = 1, \ldots, K \\ R_{k-K} & , k = K+1, \ldots, 2K . \end{cases} \tag{29}$$

Furthermore they are statistically independent and it may be shown that the vector d so defined is independent of the vector c. The resulting matrix forms for $\tilde{w}_i(t)$ and $\tilde{w}_i^2(t)$ are:

$$\tilde{w}_i(t) = h^T d \tag{30}$$

$$\tilde{w}_i^2(t) = d^T H d \tag{31}$$

A matrix form of the square-law device output, v(t), is now obtained by substitution of (26), (27), (30), and (31) into (8), Viz.

$$\begin{aligned} v(t) &= \tilde{p}_r^2(t) + 2\tilde{p}_r(t)\ h^T c + c^T H c \\ &+ \tilde{p}_i^2(t) + 2\tilde{p}_i(t)\ h^T d + d^T H d \quad . \end{aligned} \tag{32}$$

This signal is subsequently filtered to give the system output y(t). An expression for y(t) may be found by convolving (32) with the impulse response g(t) of the low-pass filter. It is helpful to define the new set of variables,

$$q_r = \tilde{p}_r^2(t) \star g(t) \tag{33}$$

$$q_i = \tilde{p}_i^2(t) \star g(t) \tag{34}$$

$$z_r^T = (2\tilde{p}_r(t)h^T) \star g(t) \tag{35}$$

$$z_i^T = (2\tilde{p}_i(t)h^T) \star g(t) \tag{36}$$

$$P = H \star g(t) \tag{37}$$

where ★ denotes convolution. The operations implied in Equations (33)-(37) are tedious but not particularly difficult. The details are contained in Appendix B. These transformations allow the system output to be written as

$$\begin{aligned} y(t) &= q_r + z_r^T c + c^T Pc \\ &\quad + q_i + z_i^T d + d^T Pd \ . \end{aligned} \tag{38}$$

In writing (38), the time dependence has been suppressed for convenience in subsequent discussions; but it should be kept in mind that q_r and q_i are scalar functions of time, z_r^T and z_i^T are vector functions of time and P is a time varying matrix.

III. DETERMINATION OF THE PROBABILITY DENSITY FUNCTION OF AN OUTPUT SAMPLE

The major objective of this work is to present procedures for determining the statistical properties of the output of a very general square-law receiver. An important step is to determine the probability density function of a sample of the output process. The approach used is to find the characteristic function and then obtain the density function via the Fourier transform.

The Characteristic Function of y(t)

As a preliminary step, we consider the problem of finding the characterisitic function of a portion of y(t). The methods used are similar to those used by Kwon and Shehadeh [5] in an analysis of noncoherent FSK systems. Specifically we seek $M_{y_r}(\nu)$ where

$$y_r = q_r + z_r^T c + c^T P c \quad . \tag{39}$$

First decompose the random vector c as

$$c = Dv \tag{40}$$

so that D is a diagonal matrix and v is a Gaussian random vector with components that are zero-mean, statistically independent and have unit variance. This is achieved by choosing the diagonal elements of D as

$$d_{kk} = \begin{cases} 1/\sqrt{R_k} & , \; k = 1, \ldots, K \\ 1/\sqrt{R_{k-K}} & , \; k = K+1, \ldots, 2K \; . \end{cases} \tag{41}$$

Substitution for c in (39) yields

$$y_r = q_r + z_r^T Dv + v^T DPDv \quad . \tag{42}$$

Let the orthonormal eigenvectors of DPD be arranged as the columns of a matrix M and define the Gaussian vector u by the transformation

$$u = M^T v. \tag{43}$$

Because the columns of M are orthonormal, it is not difficult to show that the components of u are uncorrelated, zero-mean and have unit variance. Furthermore, M is an orthogonal matrix with the property $M^T M = I$ so that we have

$$v = Mu. \tag{44}$$

Substitution for v in (42) yields

$$y_r = q_r + z_r^T\, DMu + u^T M^T DPDMu\ . \tag{45}$$

Observing that $M^T DPDM$ is just a diagonalizing transformation of the matrix DPD, we define the diagonal matrix

$$D_\alpha \triangleq M^T DPDM. \tag{46}$$

It is a simple exercise to construct D_α since its diagonal elements are just the eigenvalues of DPD. It is also useful to define the vector

$$r_r^T \triangleq z_r^T DM\ . \tag{47}$$

Substitution of (47) and (46) into (45) leads to

$$y_r = q_r + r_r^T u + u^T D_\alpha u. \tag{48}$$

A scalar form for (48) is more convenient in subsequent manipulations. We have

$$y_r = q_r + \sum_{k=1}^{2K} r_{r_k} u_k + \sum_{k=1}^{2K} \alpha_k u_k^2 \tag{49}$$

where $\{\alpha_k\}$ are the eigenvalues of DPD. Rearranging (49) as a sum of squares leads to

$$y_r = q_r - \sum_{k=1}^{2K} \frac{r_{r_k}^2}{4\alpha_k} + \sum_{k=1}^{2K} \alpha_k \left(u_k + \frac{r_{r_k}}{2\alpha_k}\right)^2 \quad . \tag{50}$$

Now let

$$Q_r \triangleq q_r - \sum_{k=1}^{2K} \frac{r_{r_k}^2}{4\alpha_k} \tag{51}$$

and

$$R_{r_k} \triangleq \frac{r_{r_k}}{2\alpha_k} \quad . \tag{52}$$

Substitution yields

$$y_r = Q_r + \sum_{k=1}^{2K} \alpha_k (u_k + R_{r_k})^2 \tag{53}$$

where the only random terms are the u_k which are statistically independent zero-mean, unit-variance Gaussian random variables.

The characteristic function of y_r is obtained by forming

$$\begin{aligned} M_{y_r}(\nu) &= E\left\{e^{j\nu y_r}\right\} \\ &= E\left\{e^{j\nu Q_r} e^{j\nu \sum_{k=1}^{2K} \alpha_k (u_k + R_{r_k})^2}\right\} \\ &= e^{jQ_r \nu} E\left\{\prod_{k=1}^{2K} e^{j\nu\alpha_k (u_k + R_{r_k})^2}\right\} \end{aligned} \tag{54}$$

Since the u_k are statistically independent and Gaussian, we may write

$$M_{y_r}(\nu) = e^{jQ_r\nu} \prod_{k=1}^{2K} E\left\{e^{j\nu\alpha_k(u_k + R_{r_k})^2}\right\} , \qquad (55)$$

where

$$E\left\{e^{j\nu\alpha_k(u_k + R_{r_k})^2}\right\}$$

$$= \int_{-\infty}^{\infty} e^{j\nu\alpha_k(u_k + R_{r_k})^2} \frac{1}{\sqrt{2\pi}} e^{-\frac{u_k^2}{2}} du_k$$

$$= \frac{\exp\left\{j \dfrac{\alpha_k R_{r_k}^2 \nu}{1-j2\alpha_k\nu}\right\}}{(1-j2\,\alpha_k\nu)^{1/2}} . \qquad (56)$$

The characteristic function of y_r thus becomes

$$M_{y_r}(\nu) = e^{jQ_r\nu} \prod_{k=1}^{2K} \frac{\exp\left\{\dfrac{j\alpha_k R_{r_k}^2 \nu}{1-j2\alpha_k\nu}\right\}}{(1-j2\alpha_k\nu)^{1/2}} . \qquad (57)$$

The same procedures may be used to find the characteristic function of the remaining terms in y(t),

$$y_i = q_i + z_i^T d + d^T P d , \qquad (58)$$

with the result

$$M_{y_i}(\nu) = e^{jQ_i\nu} \prod_{k=1}^{2K} \frac{\exp\left\{\dfrac{j\alpha_k R_{i_k}^2 \nu}{1-j2\alpha_k\nu}\right\}}{(1-j\,2\alpha_k\nu)^{1/2}} \qquad (59)$$

where

$$Q_i \triangleq q_i - \sum_{k=1}^{2K} \frac{r_{i_k}^2}{4\alpha_k} \tag{60}$$

and

$$R_{i_k} \triangleq \frac{r_{i_k}}{2\alpha_k} \quad . \tag{61}$$

Finally, observing that $y = y_r + y_i$ is the sum of independent random variables, c and d are independent random vectors hence y_r and y_i are also independent, we may find the characteristic function of y from

$$M_y(\nu) = M_{y_r}(\nu)\ M_{y_i}(\nu)$$

which becomes

$$M_y(\nu) = e^{jQ\nu} \prod_{k=1}^{2K} \frac{\exp\left\{\dfrac{jR_k^2\ \alpha_k \nu}{1-j2\alpha_k\nu}\right\}}{(1-j\ 2\alpha_k\ \nu)} , \tag{62}$$

where

$$Q = Q_r + Q_i \tag{63}$$

and

$$R_k^2 = R_{r_k}^2 + R_{i_k}^2 \quad . \tag{64}$$

Probability Density for the Case of Noise Only

Since the density function and the characteristic function of a random variable are Fourier transform pairs, the problem now reduces to finding the transform of $M_y(\nu)$ as given in (62), i.e., we want to evaluate

$$p(y) = \int_{-\infty}^{\infty} M_y(\nu)\ e^{-j\nu y}\ \frac{d\nu}{2\pi} \ . \tag{65}$$

A considerable reduction in the required effort is possible for the case of noise only. The simplification results from observing that when the signal is set to zero, we find that

$$Q = 0$$

and

$$R_k = 0, \text{ all } k.$$

The characteristic function for the case of noise only thus becomes

$$M_y(\nu) = \prod_{k=1}^{2K} \frac{1}{1-j2\,\alpha_k\,\nu} \quad . \tag{66}$$

When the eigenvalues are distinct, $M_y(\nu)$ may be expanded in partial fractions to yield

$$M_y(\nu) = \sum_{k=1}^{2K} \frac{K_k}{(1-j2\alpha_k\nu)} \tag{67}$$

where

$$K_k = (1-j2\,\alpha_k\,\nu)\,M_y(\nu)\Big|_{j\nu = \frac{1}{2\alpha_k}} \tag{68}$$

The Fourier transform of $M_y(\nu)$ is now readily found by the method of residues with the result

$$p(y) = \sum_{k=1}^{2K} \frac{K_k}{2\,\alpha_k}\, e^{-\frac{y}{2\alpha_k}} \, , \; y > 0 \tag{69}$$

where

$$K_k = \prod_{\substack{i=1 \\ i\neq k}}^{2K} \frac{1}{1 - \frac{\alpha_i}{\alpha_k}} \quad . \tag{70}$$

Probability Density for the Case of Signal Plus Noise

Solutions for the case of signal plus noise require numerical procedures. Since density functions are known to be real, it is only necessary to deal with the real part of the transform integral

$$p(y) = \int_{-\infty}^{\infty} e^{-j\nu y}\, e^{jQ\nu} \prod_{k=1}^{2K} \frac{\exp\left\{\dfrac{jR_k^2 \alpha_k \nu}{1-j2\alpha_k \nu}\right\}}{1-j\,2\alpha_k\,\nu} \frac{d\nu}{2\pi} \quad . \tag{71}$$

The real part of (71) may be shown to be

$$p(y) = \int_{-\infty}^{\infty} \frac{\exp\left\{-\dfrac{1}{2} \displaystyle\sum_{k=1}^{2K} \frac{(r_{r_k}^2 + r_{i_k}^2)\,\nu^2}{(1 + 4\,\alpha_k^2\,\nu^2)}\right\}}{\displaystyle\prod_{k=1}^{2K} (1 + 4\,\alpha_k^2\,\nu^2)^{1/2}}$$

$$\cdot \cos\left[q\nu - y\nu - \sum_{k=1}^{2K} \left[\frac{(r_{r_k}^2 + r_{i_k}^2)\,\alpha_k\,\nu^3}{(1 + 4\,\alpha_k^2\,\nu^2)} - \tan^{-1}(2\alpha_k\,\nu)\right]\right] \frac{d\nu}{2\pi} \tag{72}$$

where $q \triangleq q_r + q_i$ is the output signal voltage at the instant of interest. The integrand in (72) has been found to be an even function and is well behaved when y is close to q so that numerical solutions yield good accuracy for values of y within a few standard deviations of the mean. It has not been possible to achieve useful results for values of y in the tails of the distribution.

Some solutions for p(y) are graphed in Figures 2-6. The input signal is the same for all cases shown, a 100 ns rectangular RF pulse in the center of the pre-filter passband. The density functions are given for a sample taken at a time corresponding to the peak of the output signal waveform. Various filter configurations were considered. In every case, the post-filter bandwidth was less than the pre-filter bandwidth but not enough less to justify a Gaussian assumption.

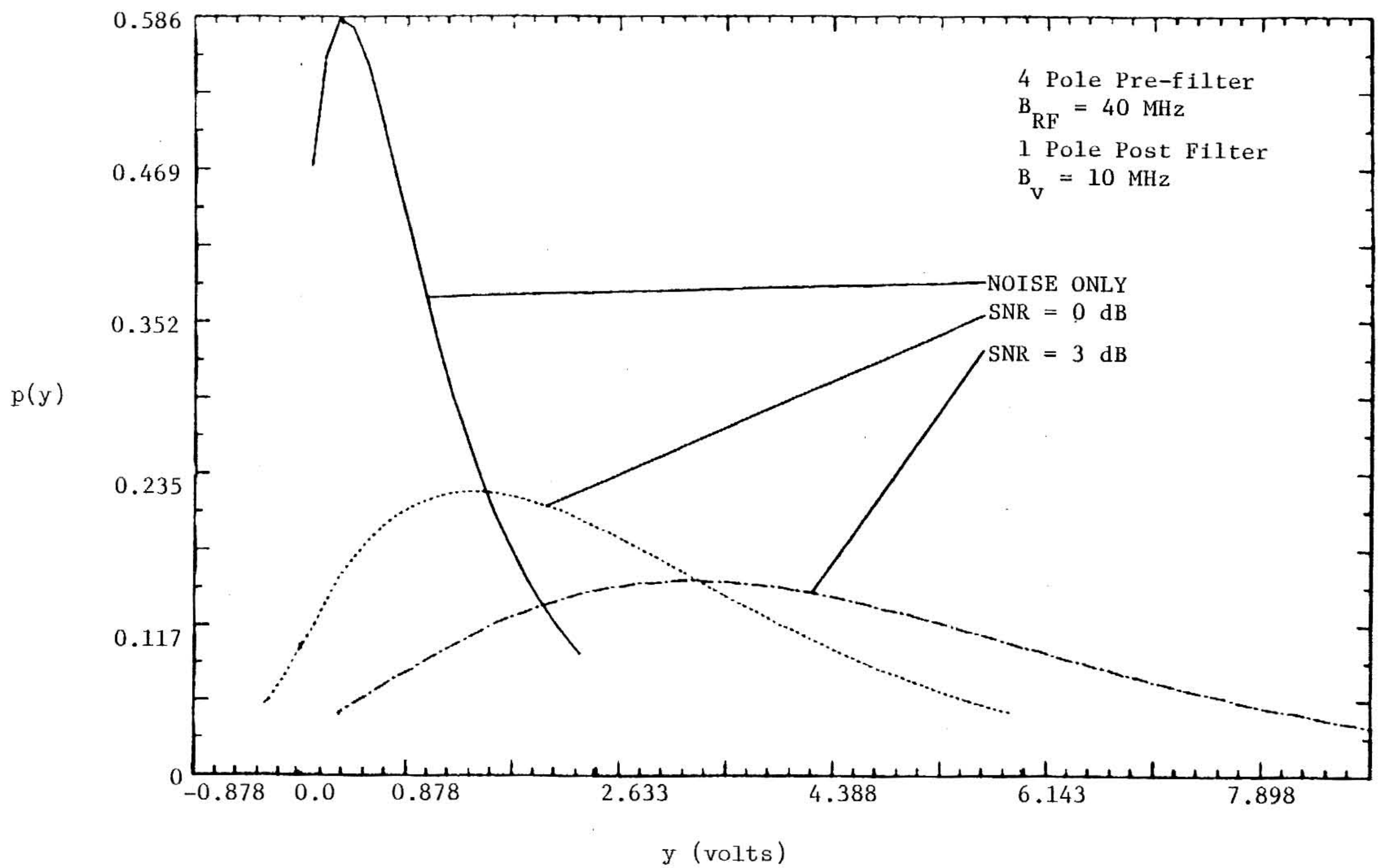


Figure 2. Probability Density Functions for Butterworth Pre-filter, Bandwidth Ratio = 4

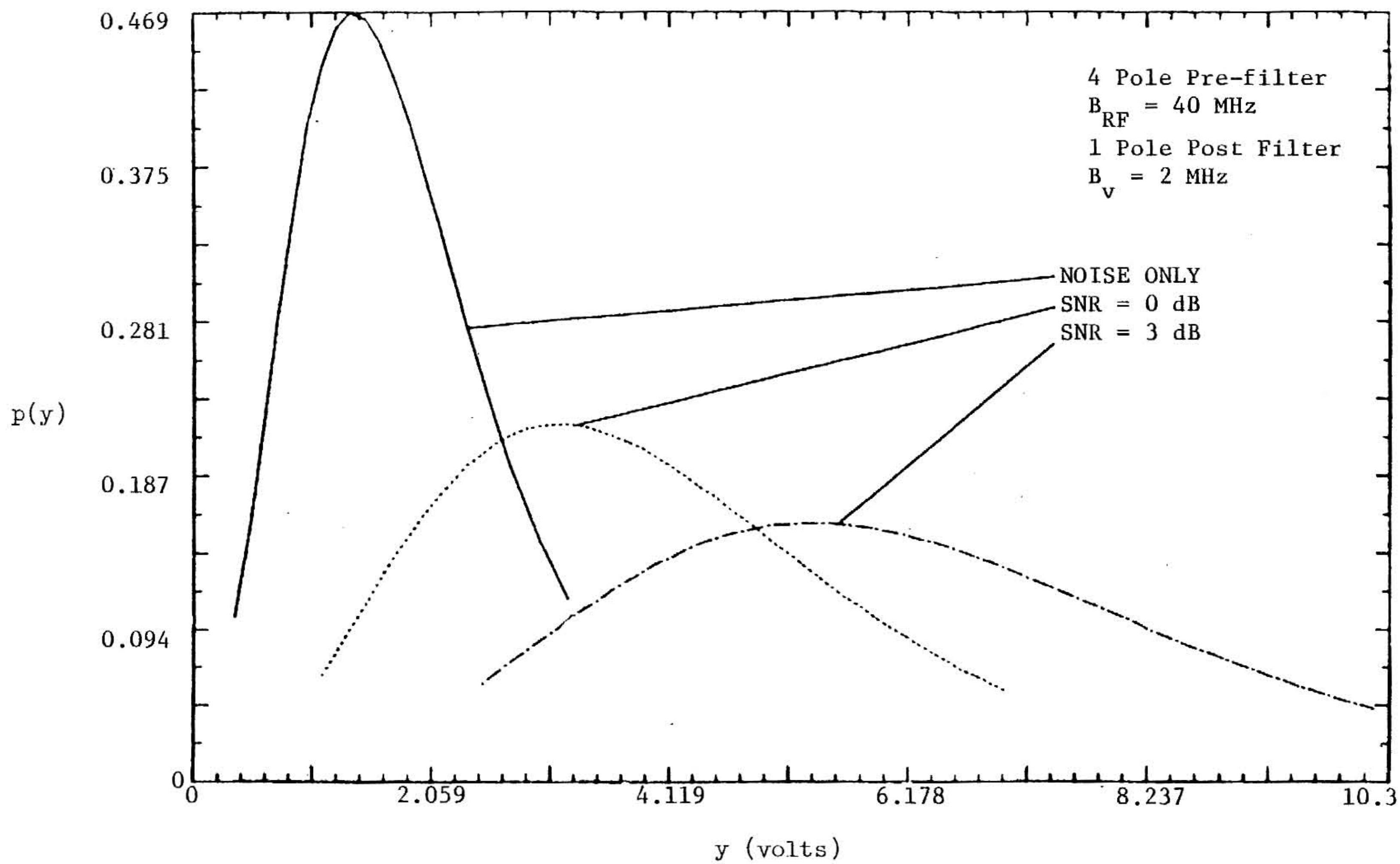


Figure 3. Probability Density Functions for Butterworth Pre-filter, Bandwidth Ratio = 20

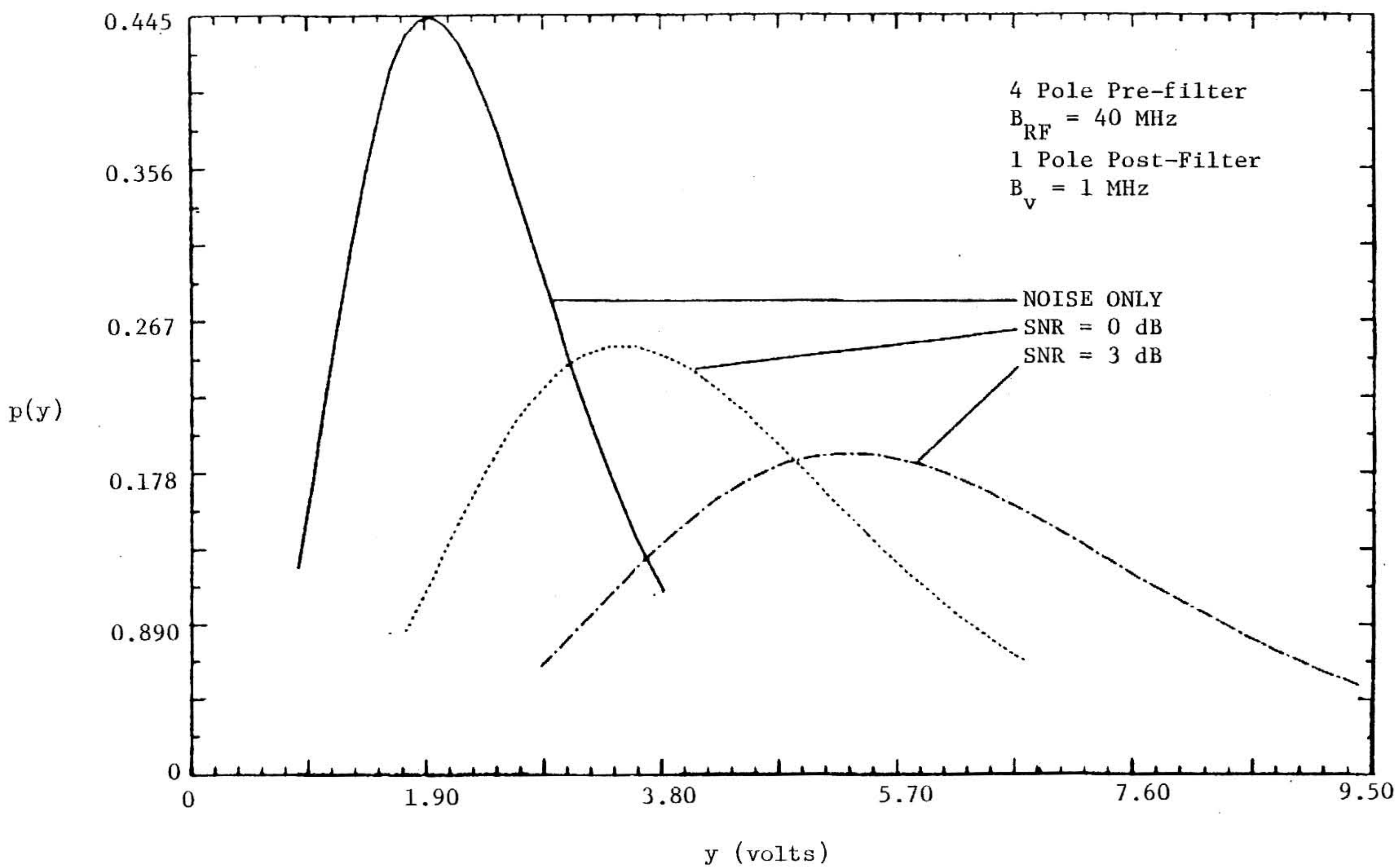


Figure 4. Probability Density Functions for Butterworth Pre-filter, Bandwidth Ratio = 40

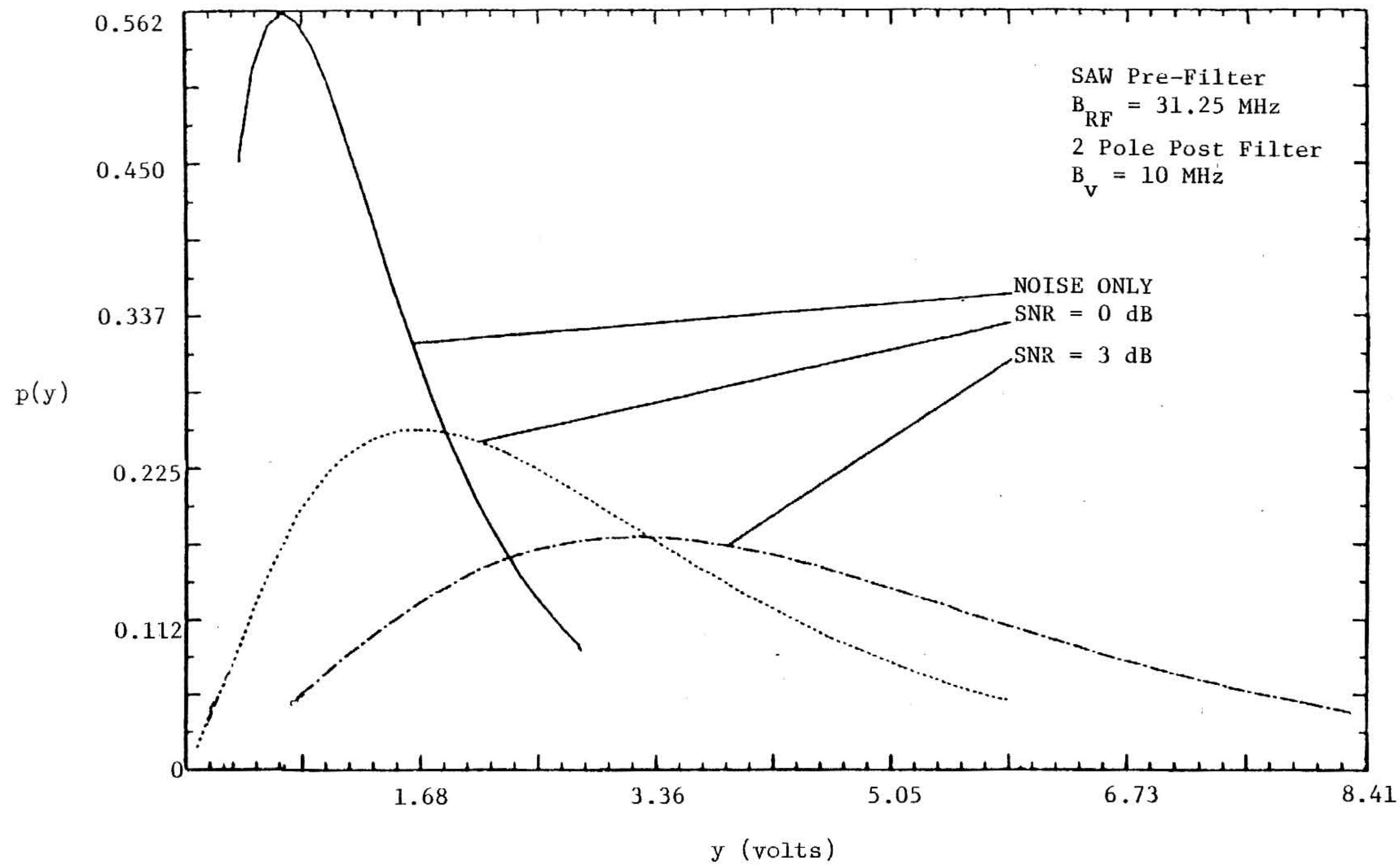


Figure 5. Probability Density Functions for Surface Acoustic Wave (SAW) Pre-filter, Bandwidth Ratio = 3.125

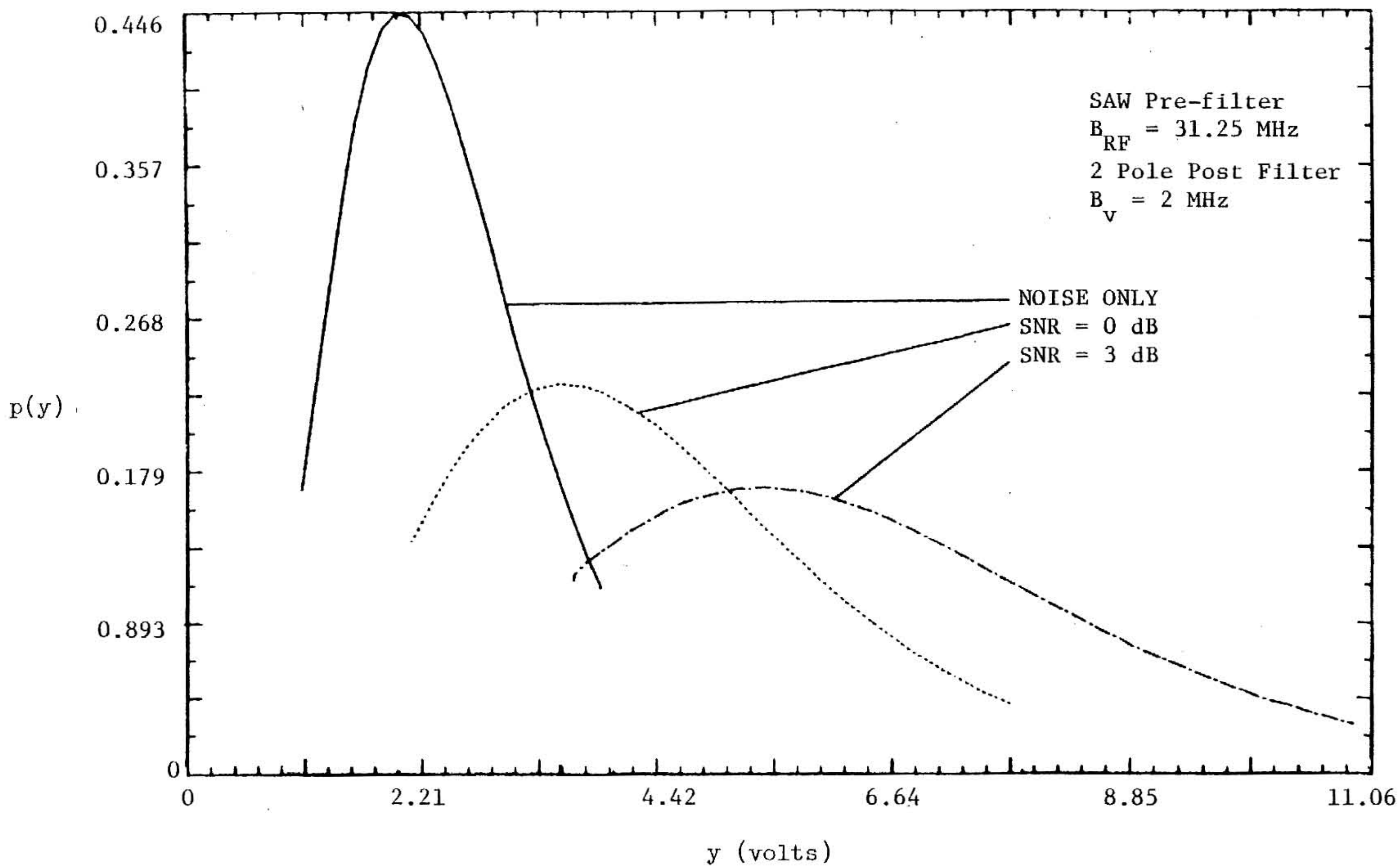


Figure 6. Probability Density Functions for Surface Acoustic Wave (SAW) Pre-filter, Bandwidth Ratio = 15.625

The Mean and Variance of a Sample of y(t)

It is possible to compute the mean and variance of a sample of the output process y(t) in terms of the eigenvalues and eigenvectors of the matrix DPD. A sample $y = y_r + y_i$ may be written, with the aid of Equation (49) for y_r and the implied counter part for y_i, as

$$
\begin{aligned}
y = q_r &+ \sum_{k=1}^{2K} r_{r_k} u_{r_k} + \sum_{k=1}^{2K} \alpha_k u_{r_k}^2 \\
&+ q_i + \sum_{k=1}^{2K} r_{i_k} u_{i_k} + \sum_{k=1}^{2K} \alpha_k u_{i_k}^2 \, .
\end{aligned}
\tag{73}
$$

Recalling that the u_{r_k} and u_{i_k} are statistically independent Gaussian random variables with zero-mean and unit variance, the mean of y is readily found to be

$$\bar{y} = q_r + q_i + 2 \sum_{k=1}^{2K} \alpha_k$$

or

$$\bar{y} = q + 2 \sum_{k=1}^{2K} \alpha_k \, . \tag{74}$$

An expression for the variance, σ_y^2, is found by forming

$$
\begin{aligned}
\sigma_y^2 &= E\left\{(y-\bar{y})^2\right\} \\
&= E\left\{\left(\sum_{k=1}^{2K} \left[\alpha_k (u_{r_k}^2 + u_{i_k}^2) + r_{r_k} u_{r_k} + r_{i_k} u_{i_k} - 2\alpha_k\right]\right)^2\right\}
\end{aligned}
\tag{75}
$$

Expanding the indicated square leads to

$$
\begin{aligned}
\sigma_y^2 = \sum_{k=1}^{2K} \sum_{\ell=1}^{2K} E\Big\{&(\alpha_k (u_{r_k}^2 + u_{i_k}^2) + r_{r_k} u_{r_k} + r_{i_k} u_{i_k} - 2\alpha_k) \\
&\cdot (\alpha_\ell (u_{r_\ell}^2 + u_{i_\ell}^2) + r_{r_\ell} u_{r_\ell} + r_{i_\ell} u_{i_\ell} - 2\alpha_\ell)\Big\}
\end{aligned}
\tag{76}
$$

If $\ell \neq k$, the indicated expectation becomes zero for any choice of ℓ and k due to the fact that the random variables are independent, zero-mean, and unit variance. Consequently, (76) reduces to

$$\sigma_y^2 = \sum_{k=1}^{2K} E\left\{(\alpha_k(u_{r_k}^2 + u_{i_k}^2) + r_{r_k}u_{r_k} + r_{i_k}u_{i_k} - 2\alpha_k)^2\right\} \tag{77}$$

After squaring and averaging term by term, the variance is found to be

$$\sigma_y^2 = 4\sum_{k=1}^{2K} \alpha_k^2 + \sum_{k=1}^{2K} (r_{r_k}^2 + r_{i_k}^2) \; . \tag{78}$$

It is worth observing that in the case of noise only, the eigenvalues α_k remain the same but the terms r_{r_k} and r_{i_k} are identically zero for all k whenever the signal component is set to zero. Thus the variance of an output sample for the case of noise only reduces to

$$\sigma_y^2 = 4\sum_{k=1}^{2K} \alpha_k^2 \quad , \text{ noise only.} \tag{79}$$

The output process is known to be stationary for the case of noise only, hence the eigenvalues, α_k, are not expected to change with time even through the elements of the P matrix are time-varying. This property has been verified. For the case of signal plus noise, the output process is frequently nonstationary. In this case the variance changes with time and all of the time dependence is embedded in the parameters r_{r_k} and r_{i_k}.

IV. CALCULATION OF DETECTION PROBABILITY

The most commonly used indicator of receiver performance is the received signal power required to yield a specified probability of detection when the receiver has been set up to operate at a specified false-alarm rate. The purpose of the following arguments is to extend the results of the previous section to yield relationships which may be used to compute this performance measure for very general receiver configurations. The discussion begins with the problem of determining the threshold which results in the desired false-alarm rate.

Setting the Threshold

The false-alarm rate (FAR) is the average number of times per second that the threshold is exceeded when the receiver has noise only at the input. If the bandwidth of low-pass output filter is $B_{\ell p}$ Hz, there are approximately $2B_{\ell p}$ independent opportunities per second for the output noise to exceed threshold. The probability of false alarm, P_f, for a given noise sample is then related to the false-alarm rate by

$$P_f = \frac{FAR}{2B_{\ell p}} \quad . \tag{80}$$

The probability of false alarm is, in turn, related to the threshold voltage V_t by

$$P_f = \int_{V_t}^{\infty} p(y)\, dy \quad , \tag{81}$$

where p(y) is the probability density function for a sample of the receiver output for the case of noise only at the input. This density was determined in a previous section to be

$$p(y) = \sum_{k=1}^{2K} \frac{K_k}{2\alpha_k} e^{-\frac{y}{2\alpha_k}}.$$

Substituting for p(y) in (81) and integrating yields

$$P_f = \sum_{k=1}^{2K} K_k e^{-\frac{V_t}{2\alpha_k}} \tag{82}$$

which may be readily solved by numerical methods to find the threshold V_t to yield a specified P_f or corresponding FAR.

Calculating P_d

The probability of detection, denoted by P_d, of an output signal is equal to the probability of the output signal exceeding the detector threshold voltage. That is,

$$P_d(V_t) \triangleq P[y \geq V_t]$$

or

$$P_d(V_t) = 1 - P[y < V_t] \tag{83}$$

where V_t is the threshold voltage of the detector. Equation (83) is based on the assumption that pre-filter and post-filter bandwidths are such that only one independent sample of the pulse occurs.

Recall from (72) that

$$p(y) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{\exp\left\{ -\frac{1}{2} \sum_{k=1}^{2K} \left[\frac{(r_{r_k}^2 + r_{i_k}^2)\, \nu^2}{(1+4\, \alpha_k^2\, \nu^2)} \right] \right\}}{\prod_{k=1}^{2K} (1 + 4\, \alpha_k^2\, \nu^2)^{1/2}}$$

$$\cdot \cos\left[q\nu - y\nu - \prod_{k=1}^{2K} \left\{\frac{(r_{r_k}^2 + r_{i_k}^2)\alpha_k \nu^3}{(1 + 4\alpha_k^2 \nu^2)} - \tan^{-1}(2\alpha_k \nu)\right\}\right] d\nu$$

Letting

$$e(\nu) = \frac{\exp\left\{-\frac{1}{2}\sum_{k=1}^{2K}\left[\frac{(r_{r_k}^2 + r_{i_k}^2)\nu^2}{(1 + 4\alpha_k^2\nu^2)}\right]\right\}}{\prod_{k=1}^{2K}(1 + 4\alpha_k^2\nu^2)^{1/2}} \tag{84}$$

and

$$g(\nu) = -q\nu + \sum_{k=1}^{2K}\left[\frac{(r_{r_k}^2 + r_{i_k}^2)\alpha_k \nu^3}{(1 + 4\alpha_k^2\nu^2)} - \tan^{-1}(2\alpha_k\nu)\right] , \tag{85}$$

Equation (72) becomes

$$p(y) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e(\nu)\cos(y\nu + g(\nu))\, d\nu . \tag{86}$$

Using the rectangular rule of numerical integration, p(y) can be expressed as

$$p(y) = \frac{1}{2\pi}\sum_{\ell=-N}^{N} e(\nu_\ell)\cos(y\nu_\ell + g(\nu_\ell))\,\Delta\nu \tag{87}$$

To find $P[y < V_t]$, p(y) is integrated over the appropriate limits. Specifically,

$$P[y < V_t] = \int_{-\infty}^{V_t} p(y)\, dy \tag{88}$$

$$= \int_{-\infty}^{0} p(y)\, dy + \int_{0}^{V_t} p(y)\, dy . \tag{89}$$

Solving the first integral in (89),

$$\int_{-\infty}^{0} p(y)\ dy = \frac{1}{2\pi}\int_{-\infty}^{\infty} \sum_{\ell=-N}^{N} e(\nu_\ell)\ \cos(y\nu_\ell + g(\nu_\ell))\ \Delta\nu\ dy \qquad (90)$$

$$= \sum_{\ell=-N}^{N} e(\nu_\ell)\left[\int_{-\infty}^{\infty} \frac{\cos\ (y\nu_\ell + g(\nu_\ell))}{2\pi}\ dy\right]\ \Delta\nu \qquad (91)$$

It is shown in Appendix C that

$$\int_{-\infty}^{\infty} \frac{\cos(y\nu_\ell + g(\nu_\ell))}{2\pi}\ dy = \frac{1}{2}\ \delta(\nu_\ell) + \frac{\sin(y\nu_\ell + g(\nu_\ell))}{\pi\nu_\ell}\ . \qquad (92)$$

Therefore

$$\int_{-\infty}^{0} p(y)\ dy = \sum_{\ell=-N}^{N} \frac{1}{2}\ e(\nu_\ell)\ \delta(\nu_\ell)\ \Delta\nu$$

$$+ \frac{1}{2\pi} \sum_{\ell=-N}^{N} \left\{\frac{e(\nu_\ell)\ \sin(g(\nu_\ell))\Delta\nu}{\nu_\ell}\right\} \qquad (93)$$

The expression

$$\sum_{\ell=-N}^{N} \frac{1}{2}\ e(\nu_\ell)\ \delta(\nu_\ell)\ \Delta\nu \qquad (94)$$

represents the rectangular rule integration form of

$$\int_{-\infty}^{\infty} \frac{1}{2}\ e(\nu)\ \delta(\nu)d\nu \qquad (95)$$

which can be shown to be $\frac{1}{2}$. Thus

$$\int_{-\infty}^{0} p(y)\ dy = \frac{1}{2} + \frac{1}{2\pi} \sum_{\ell=-N}^{N}\left\{\frac{e(\nu_\ell)\ \sin(g(\nu_\ell))\Delta\nu}{\nu_\ell}\right\}\ . \qquad (96)$$

Now finding $\int_0^{V_t} p(y)\, dy$,

$$\int_0^{V_t} p(y)\, dy = \int_0^{V_t} \frac{1}{2\pi} \sum_{\ell=-N}^{N} e(\nu_\ell) \cos(y\nu_\ell + g(\nu_\ell))\Delta\nu\, dy \tag{97}$$

$$= \frac{1}{2} \sum_{\ell=-N}^{N} \left\{\frac{e(\nu_\ell)[\sin(V_t\nu_\ell + g(\nu_\ell)) - \sin(g(\nu_\ell))]\Delta\nu}{\nu_\ell}\right\} \tag{98}$$

Substituting (96) and (98) into (89), we have

$$P[y<V_t] = \frac{1}{2} + \frac{1}{2\pi} \sum_{\ell=-N}^{N} \left\{\frac{e(\nu_\ell) \sin(V_t\nu_\ell + g(\nu_\ell))\Delta\nu}{\nu_\ell}\right\} \quad . \tag{99}$$

Recognizing that the expression in braces is an even function, (99) can be rewritten as

$$P[y<V_t] = \frac{1}{2} + \frac{1}{\pi} \sum_{\ell=1}^{N} \left\{\frac{e(\nu_\ell) \sin(V_t\nu_\ell + g(\nu_\ell))\Delta\nu}{\nu_\ell}\right\} \quad . \tag{100}$$

Now substituting (100) into (83), we find

$$P_d(V_t) = \frac{1}{2} - \frac{1}{\pi} \sum_{\ell=1}^{N} \left\{\frac{e(\nu_\ell) \sin(V_t\nu_\ell + g(\nu_\ell))}{\nu_\ell}\right\} \Delta\nu \ , \tag{101}$$

and replacing $e(\nu_\ell)$ and $g(\nu_\ell)$ with the expressions defined in (84) and (85), the final form of P_d is found to be

$$P_d(V_t) = \frac{1}{2} - \frac{1}{\pi} \sum_{\ell=1}^{N} \left\{ \frac{\left[- \frac{1}{2} \sum_{k=1}^{2K} \frac{(r_{r_k}^2 + r_{i_k}^2)\nu_\ell^2}{(1 + 4\, \alpha_k^2\, \nu_\ell^2)} \right]}{\nu_\ell \cdot \prod_{k=1}^{2K} (1+4\, \alpha_k^2\, \nu_\ell^2)^{1/2}} \right.$$

$$\cdot \sin\left[V_t \nu_\ell - q\nu_\ell + \sum_{k=1}^{2K} \left\{\frac{(r_{r_k}^2 + r_{i_k}^2)\ \alpha_k \nu_\ell^3}{(1 + 4\ \alpha_k^2\ \nu_\ell^2)} - \tan^{-1}(2\alpha_k \nu_\ell)\right\}\right] \Delta\nu\Bigg\} \tag{102}$$

V. PROBABILITY OF DETECTION CURVES FOR SOME SPECIFIC CASES

The numerical solution for probability of detection, Equation (102), developed in a previous section was used to find probability of detection curves for several cases of interest. The results are presented in Figures 7, 8 and 9. In all cases, a 100 nanosecond pulse is used as the input signal to the system. Results are based on a single sample of the output signal with the time of sampling corresponding to the time of the output signal peak. Also, a system noise figure of 2 dB was used for all cases.

Figure 7 shows the probability of detection curves for cases of a surface acoustic wave (SAW) RF pre-filter with a bandwidth of 31.25 MHz. A 1-pole Butterworth filter is used as the post filter. The three curves presented correspond to post filter bandwidths of 2 MHz, 4 MHz, and 10 MHz, or pre- to post filter bandwidth ratios of 15.625, 7.81, and 3.125.

As a check on the accuracy of the numerical methods developed in this work, a comparison was done with results presented by Skolnik [6] and is shown in Figure 8. The results found by Skolnik are based on a linear envelope detector. However, it is generally accepted that the detector law plays a minor role in the probability of detection. The comparison presented is based on a 4-pole Butterworth pre-filter with a bandwidth of 40 MHz, and a 1-pole post filter with a bandwidth of 20 MHz, or a pre- to post filter bandwidth ratio of 2. The results found by the numerical methods and those of Skolnik are in close agreement.

Figure 9 shows a comparison of the probability of detection curves found with and without a Gaussian assumption for the case of a 4-pole Butterworth pre-filter of 40 MHz bandwidth and a 1-pole Butterworth post

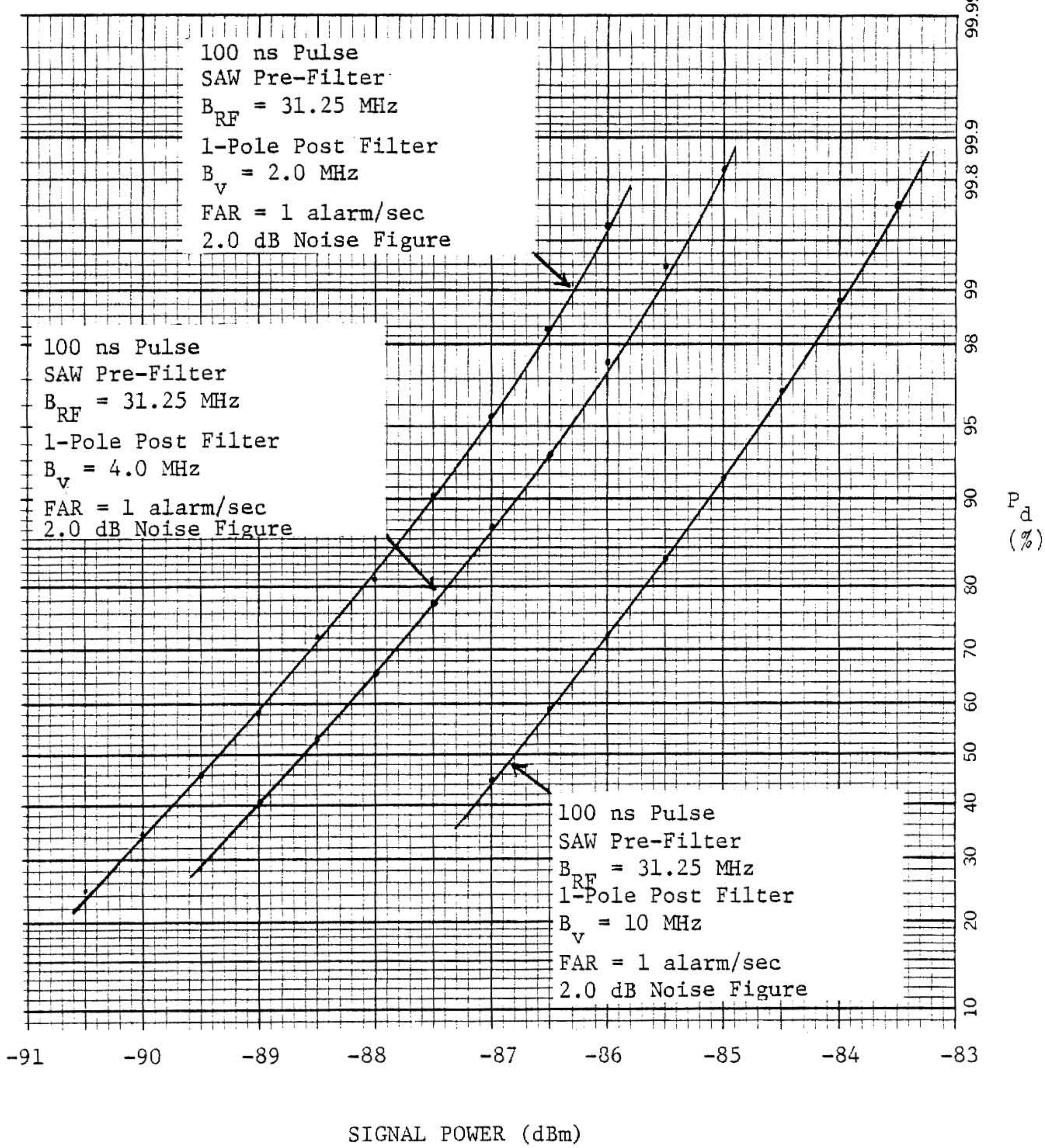


Figure 7. Probability of Detection Curves for Three Surface Acoustic Wave (SAW) Pre-Filter Cases

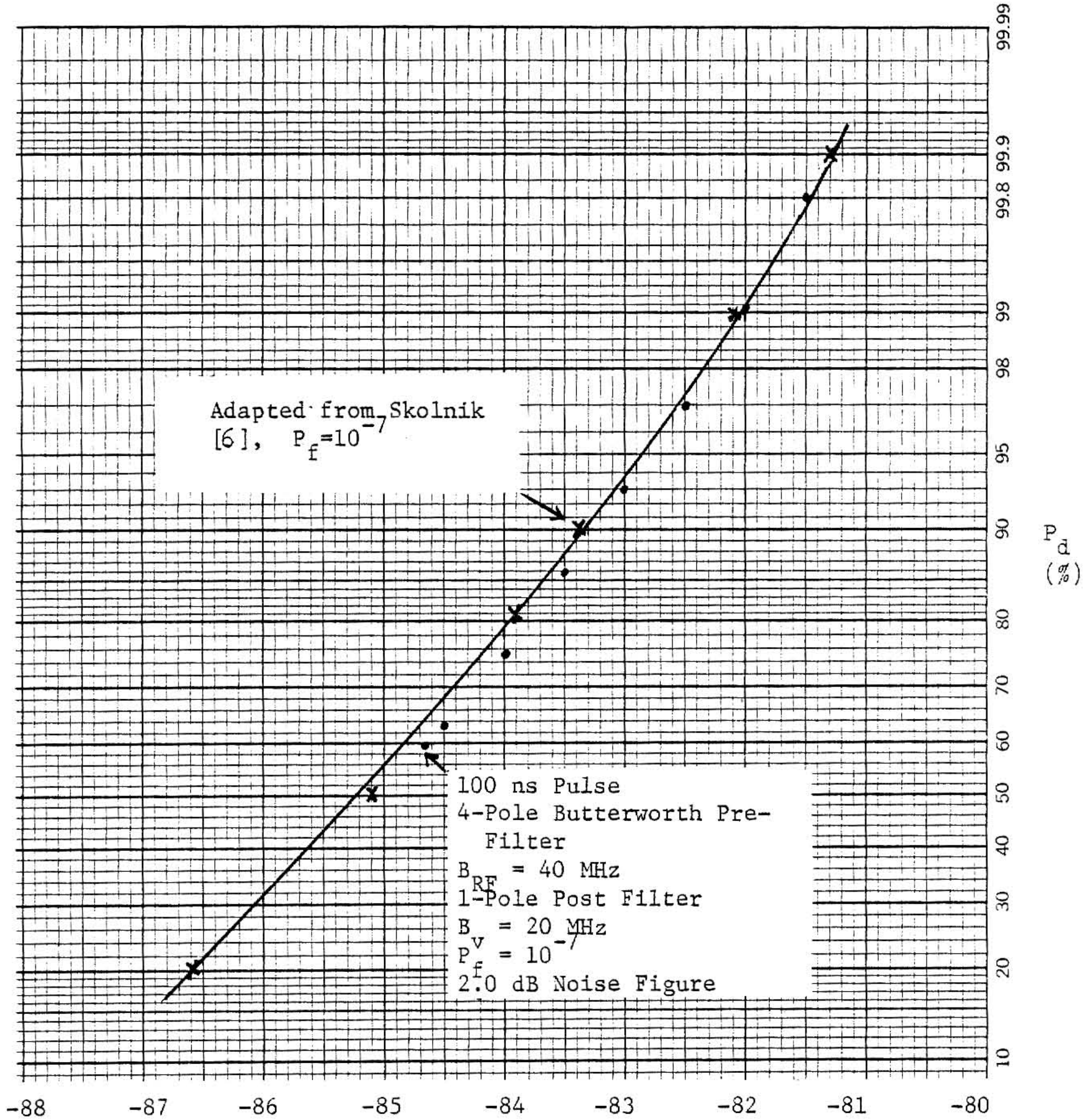


Figure 8. Probability of Detection Using Numerical Techniques Compared to a Previous Result

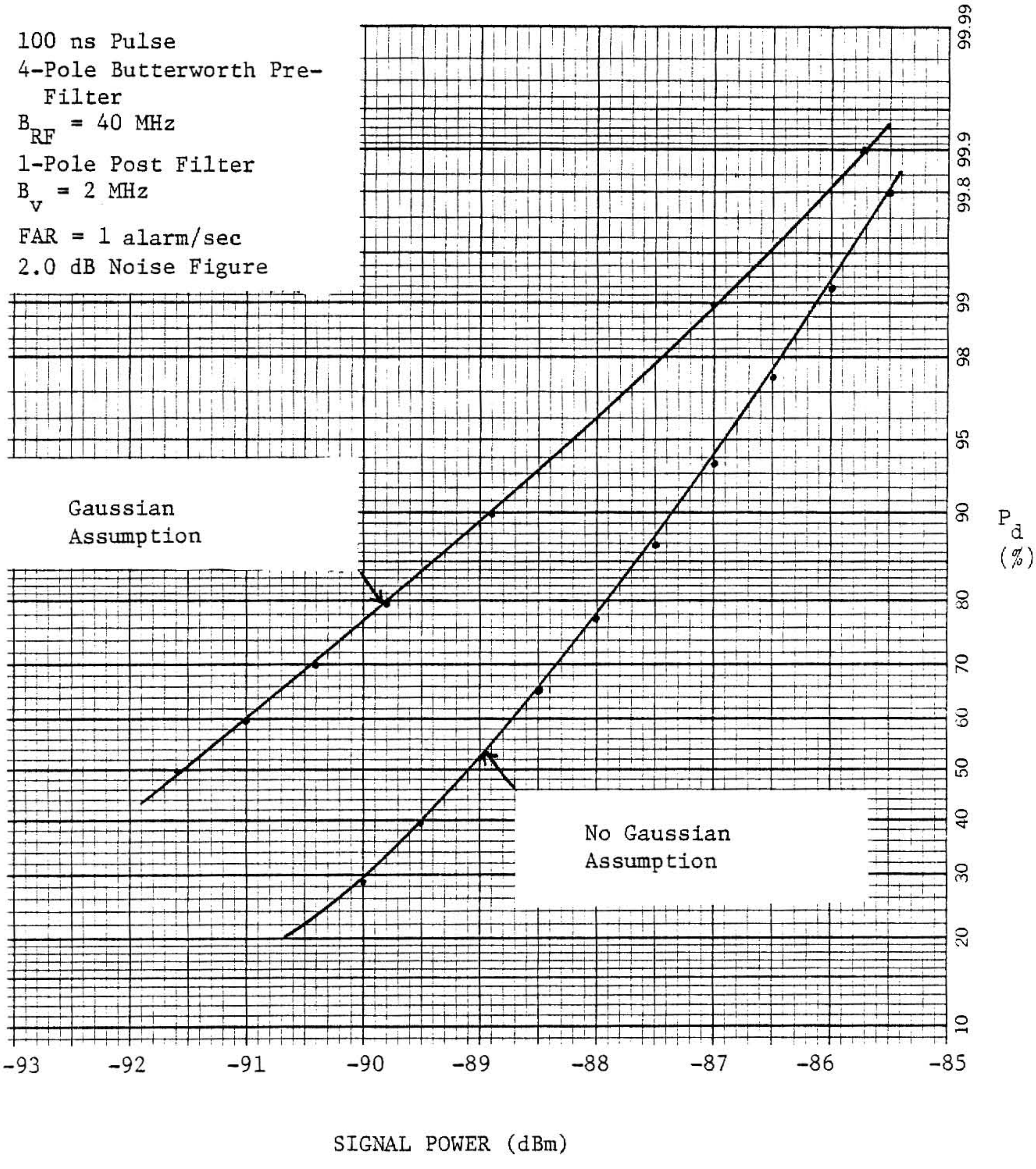


Figure 9. Probability of Detection With and Without a Gaussian Assumption

filter with a bandwidth of 2 MHz, or a pre- to post filter bandwidth ratio of 20. It will be noticed that as signal power increases, the probability of detection curves for the results with a Gaussian assumption and those without a Gaussian assumption come closer together. This is not unexpected since the signal dependent noise terms begin to dominate at high signal power, and these noise terms are Gaussian in nature.

VI. CONCLUSIONS

A method for evaluating the probability of detection for a general square-law detector has been given. The main results appear in Equations (72) and (102), the probability density function and probability of detection, respectively. The solutions of these equations are easily found using numerical methods.

The results of the development presented are directly applicable to any input signal and to any pre- and post filter configuration. The procedure given is not without its limitations, however. As the pre-filter to post filter bandwidth becomes very large (greater than 40), the matrices used become sparse and the accuracy of the results becomes suspect. Also, the accuracy of the probability of detection on the tails of p(y) are not accurate due to the nature of p(y).

The probability density functions and the probability of detection curves for several filter configurations have been presented. All results are based on a single sample of the signal plus noise. Further development must be done to find the density function and probability of detection for the multiple sample case.

APPENDIX A

THE NOISE MODEL

The objective of this appendix is to outline the derivation of a noise model which has proven effective in numerical procedures. The approach uses a series expansion described in Yaglom [3] to approximate the complex envelope of a bandpass random process.

The Equivalent Low-Pass Model

Let the noise signal of interest be a stationary bandpass Gaussian process, $n(t)$. Procedures for modeling $n(t)$ in terms of a low-pass complex envelope $\tilde{n}(t)$ are available elsewhere [4], therefore only a few necessary results are summarized here. The bandpass signal and its complex envelope are related by

$$n(t) = \text{Re}\ \{\tilde{n}(t)e^{j\omega_c t}\} \tag{A1}$$

where ω_c is the center frequency of the system. The complex envelope $\tilde{n}(t)$ is a complex low-pass stationary Gaussian random process. If the correlation function of $\tilde{n}(t)$ is defined as

$$R_{\tilde{n}}(\tau) \triangleq E\{n^*(t)n(t+\tau)\}\ , \tag{A2}$$

then it can be shown that the power spectrum of $n(t)$ is related to that of $\tilde{n}(t)$ by

$$S_n(f) = \frac{1}{4}\, S_{\tilde{n}}(f - f_c) + \frac{1}{4}\, S_{\tilde{n}}(-f - f_c) \tag{A3}$$

where $S_{\tilde{n}}(f)$ and $R_{\tilde{n}}(\tau)$ are Fourier transform pairs. The complex process $\tilde{n}(t)$ may be written in terms of the real processes $\tilde{n}_r(t)$ and $\tilde{n}_i(t)$, i.e.,

$$\tilde{n}(t) = \tilde{n}_r(t) + j\ \tilde{n}_i(t) \tag{A4}$$

where $\tilde{n}_r(t)$ and $\tilde{n}_i(t)$ are statistically independent Gaussian processes with identical correlation functions

$$R_{\tilde{n}_r}(\tau) = R_{\tilde{n}_i}(\tau) = \frac{1}{2} R_{\tilde{n}}(\tau). \tag{A5}$$

An Approximation for $\tilde{n}(t)$

It has been shown (see Yaglom [3]) that every stationary random process with a correlation function of the form

$$R_{\tilde{n}}(\tau) = \sum_{k=-\infty}^{\infty} R_k e^{j\lambda_k \tau} \tag{A6}$$

may be expressed as a series,

$$\tilde{n}(t) = \sum_{k=-\infty}^{\infty} n_k e^{j\lambda_k t} \tag{A7}$$

where the n_k are complex random variables with

$$E\{n_k\} = 0 \tag{A8}$$

and

$$E\{n_k n_\ell^*\} = \begin{cases} R_k, & \ell=k \\ 0, & \ell\neq k \end{cases} \tag{A9}$$

The λ_k are an appropriately selected set of frequencies. For the case of interest here, $\tilde{n}(t)$ is a Gaussian process and hence the n_k are necessarily Gaussian random variables. Condition (A8) implies they are also statistically independent. It may be verified that a sufficient condition for satisfying requirements (A4) and (A5) is to further stipulate that the n_k have real and imaginary parts n_{r_k} and n_{i_k} which are statistically independent with zero-means and equal variances

$$E\{n_{r_k}^2\} = E\{n_{i_k}^2\} = \frac{R_k}{2} \quad . \tag{A10}$$

A suitable finite term approximation to (A7) may be obtained if the parameters R_k and λ_k can be specified in such a way that the correlation function $R_{\tilde{n}}(\tau)$ is accurately represented over an appropriate range of τ. The key is to use a Gauss quadrature procedure. A specific case will serve to illustrate.

Let the bandpass process n(t) have a power density $N_o/2$ over a bandwidth B_n centered on f_c Hz. Then the complex envelope $\tilde{n}(t)$ will have the power spectrum

$$S_{\tilde{n}}(f) = \begin{cases} 2\ N_o \ , & |f| \le B_n \\ 0 \ , & |f| > B_n \end{cases} \quad . \tag{A11}$$

The correlation function of $\tilde{n}(t)$ is found by taking the inverse Fourier transfer of $S_{\tilde{n}}(f)$, i.e.,

$$R_{\tilde{n}}(\tau) = \int_{-B_n}^{B_n} 2\ N_o \cos(2\pi f\tau)\ df \ . \tag{A12}$$

Making the change of variables $f = B_n \nu$ allows (A12) to be written as

$$R_{\tilde{n}}(\tau) = 2\ N_o\ B_n \int_{-1}^{1} \cos(2\pi B_n \nu\tau)\ d\nu \ . \tag{A13}$$

Let the pairs $\{(\gamma_k, \nu_k)\}_{k=-K}^{K}$ be a Gauss quadrature rule (GQR) with respect to the unit weighting function on [-1, 1]. A good approximation to the integral in (A13) which improves with increasing K is

$$R_{\tilde{n}}(\tau) \cong 2\, N_o\, B_n \sum_{k=-K}^{K} \gamma_k \cos(2\pi B_n \nu_k \tau) . \tag{A14}$$

Now define

$$R_k \triangleq 2\, N_o\, B_n\, \gamma_k \tag{A15}$$

and

$$\lambda_k \triangleq 2\pi\, B_n\, \nu_k \quad . \tag{A16}$$

Substitution into (A14) yields the form

$$R_{\tilde{n}}(\tau) \cong \sum_{k=-K}^{K} R_k \cos(\lambda_k \tau) \ . \tag{A17}$$

A convenient symmetry of the GQR employed here is that for every pair (γ_k, ν_k) in the rule, the pair $(\gamma_k, -\nu_k)$ is also in the rule. Thus an exactly equivalent representation of (A17) is

$$R_{\tilde{n}}(\tau) \cong \sum_{k=-K}^{K} R_k e^{j\lambda_k \tau} \quad . \tag{A18}$$

It is routine to show that (A18) is the correlation function of the approximating process

$$\tilde{n}(t) \cong \sum_{k=-K}^{K} n_k e^{j\lambda_k t} \ , \tag{A19}$$

hence we conclude that if the pairs $\{(R_k, \lambda_k)\}$ are selected using the GQR procedures just described, (A19) will be a good approximation for K sufficiently large. The nature of the approximation (A18) is such that it compares very well with the actual correlation function with only a few terms when τ is small. The memory of the system under analysis thus dictates the number of terms required for a useful representation.

There are times when it is convenient to rearrange (A19) slightly by taking advantage of the symmetry of the GQR. Let the rule be restricted to have an even number of terms. Then (A19) may be written as

$$\tilde{n}(t) = \sum_{k=1}^{K} (a_k e^{j\lambda_k t} + b_k e^{-j\lambda_k t}) \qquad \text{(A20)}$$

where

$$a_k \triangleq n_k \quad , \quad k = 1, 2, \ldots, K$$

and

$$b_k \triangleq n_k \quad , \quad k = -1, -2, \ldots, -K.$$

The a_k and b_k are observed to be statically independent complex random variables with equal variances

$$E\{|a_k|^2\} = E\{|b_k|^2\} = R_k, \quad k = 1, 2, \ldots, K. \qquad \text{(A21)}$$

APPENDIX B

COMPUTATIONAL CONSIDERATIONS

The purpose of this Appendix is to give procedures for carrying out the computations implied in Equations (33)-(37).

The Output Signal

The signal component of the output is

$$q(t) = q_r(t) + q_i(t)$$

$$= |\tilde{p}(t)|^2 \star g(t) \tag{B1}$$

where

$$\tilde{p}(t) = \tilde{s}(t) \star h(t) \tag{B2}$$

It is convenient for numerical purposes to represent the complex envelope, $\tilde{s}(t)$, of the input signal in the series form

$$\tilde{s}(t) = \sum_{n=-N}^{N} s_n e^{j\omega_n t} , \tag{B3}$$

where N is taken sufficiently large to yield a good representation. A series representation for $\tilde{p}(t)$ is then readily found as

$$\tilde{p}(t) = \sum_n s_n \tilde{H}(\omega_n) e^{j\omega_n t} . \tag{B4}$$

Carrying out the operations implied in (B1) yields the output signal as

$$q(t) = \sum_n \sum_m p_n p_m^* G(\omega_n - \omega_m) e^{j(\omega_n - \omega_m)t} . \tag{B5}$$

The Signal Cross Noise Vectors, z_r^T and z_i^T

The vectors z_r^T and z_i^T are real and since the vector h^T and impulse response g(t) are also real, it follows from Equations (35) and (36) that the elements of these vectors are given by

$$z_{r_k} = \mathrm{Re}\{(2\tilde{p}(t)\, h_k) \star g(t)\} \tag{B6}$$

and

$$z_{i_k} = \mathrm{Im}\{(2\tilde{p}(t)h_k) \star g(t)\}\ . \tag{B7}$$

If $\tilde{p}(t)$ is in series form as in (B4), the quantity in braces may be found with the aid of (20) to be

$$\left[2\tilde{p}(t)h_k(t)\right]\star g(t) = \begin{cases} \sum\limits_n s_n[\tilde{H}(\lambda_k)\tilde{H}(\omega_n)G(\omega_n+\lambda_k)e^{j(\omega_n+\lambda_k)t} \\ \qquad + \tilde{H}^*(\lambda_k)\tilde{H}(\omega_n)G(\omega_n-\lambda_k)e^{j(\omega_n-\lambda_k)t}]\ , \\ \qquad\qquad k = 1, 2, \ldots, K \\ \sum\limits_n -js_n[\tilde{H}(\lambda_{k-K})\tilde{H}(\omega_n)G(\omega_n+\lambda_{k-K})e^{j(\omega_n+\lambda_{k-K})t} \\ \qquad - \tilde{H}^*(\lambda_{k-K})\tilde{H}(\omega_n)G(\omega_n-\lambda_{k-K})e^{j(\omega_n-\lambda_{k-K})t}]\ , \\ \qquad\qquad k = K+1, \ldots, 2K \end{cases} \tag{B8}$$

Numerical evaluation of detection probability and probability density requires elements of the vectors r_r^T and r_i^T. Once the vectors z_r^T and z_i^T are formed using (B6), (B7), and (B8); r_r^T and r_i^T are obtained from

$$r_r^T = z_r^T\ DM \tag{B9}$$

and

$$r_i^T = z_i^T\ DM\ . \tag{B10}$$

The Noise Matrix, P

As indicated in Equation (37), the matrix P is found by convolving the matrix H with the impulse response, g(t), of the output filter. Actually, the exponential nature of the elements of H makes it more convenient to perform this operation in the frequency domain. The elements of P may be found by straightforward manipulation to be

$$P_{k\ell} = \frac{1}{2}\,\mathrm{Re}\left\{\tilde{H}(\lambda_k)\tilde{H}(\lambda_\ell)G(\lambda_k + \lambda_\ell)e^{j(\lambda_k + \lambda_\ell)t} + \tilde{H}(\lambda_k)\tilde{H}^*(\lambda_\ell)G(\lambda_k - \lambda_\ell)e^{j(\lambda_k - \lambda_\ell)t}\right\} \tag{B11}$$

for $1 \le k \le K$, $1 \le \ell \le K$

and

$$P_{k\ell} = \frac{1}{2}\,\mathrm{Im}\left\{\tilde{H}(\lambda_k)\tilde{H}(\lambda_{\ell-k})G(\lambda_k + \lambda_{\ell-k})e^{j(\lambda_k + \lambda_{\ell-K})t} - \tilde{H}(\lambda_k)\tilde{H}^*(\lambda_{\ell-k})G(\lambda_k - \lambda_{\ell-K})e^{j(\lambda_k - \lambda_{\ell-K})t}\right\} \tag{B12}$$

for $1 \le k \le K$, $K < \ell \le 2K$

with $P_{\ell k} = P_{k\ell}$ for $1 \le \ell \le K$, $K < k \le 2K$

and finally

$$P_{k\ell} = -\frac{1}{2}\,\mathrm{Re}\left\{\tilde{H}(\lambda_k)\tilde{H}(\lambda_\ell)G(\lambda_k + \lambda_\ell)e^{j(\lambda_k + \lambda_\ell)t} - \tilde{H}(\lambda_k)\tilde{H}^*(\lambda_\ell)G(\lambda_k - \lambda_\ell)e^{j(\lambda_k - \lambda_\ell)t}\right\} \tag{B13}$$

for $K < k \le 2K$, $K < \ell \le 2K$.

APPENDIX C

SOLUTION OF A USEFUL INTEGRAL

The purpose of this Appendix is to find a closed form of

$$\int_{-\infty}^{0} \frac{\cos(\omega t + \phi)}{2\pi}\, d\omega .$$

$$\int_{-\infty}^{0} \frac{\cos(\omega t + \phi)}{2\pi}\, d\omega = \int_{-\infty}^{-\phi/t} \frac{\cos(\omega t + \phi)}{2\pi}\, d\omega + \int_{-\phi/t}^{0} \frac{\cos(\omega t + \phi)}{2\pi}\, d\omega \;. \qquad \text{(C1)}$$

Let $(\omega + \phi/t) = x$ and $d\omega = dx$. When $\omega = -\phi/t$, $x = 0$ and when $\omega = -\infty$, $x = -\infty$. Then

$$\int_{-\infty}^{-\phi/t} \frac{\cos(\omega t + \phi)}{2\pi}\, d\omega = \int_{-\infty}^{0} \frac{\cos xt}{2\pi}\, dx = \frac{1}{2}\ \delta(t) \qquad . \qquad \text{(C2)}$$

Substituting (C2) into (C1), we get

$$\int_{-\infty}^{0} \frac{\cos(\omega t + \phi)}{2\pi}\, d\omega = \frac{1}{2}\ \delta(t) + \int_{-\phi/t}^{0} \frac{\cos(\omega t + \phi)}{2\pi}\, d\omega \qquad . \qquad \text{(C3)}$$

The remaining integral on the right-hand side is straightforward, and the result is

$$\int_{-\infty}^{0} \frac{\cos(\omega t + \phi)}{2\pi}\, d\omega = \frac{1}{2}\ \delta(t) + \frac{\sin\phi}{2\pi t} \qquad \text{(C4)}$$

APPENDIX D

COMPUTER PROGRAMS

The computer programs used for the numerical solution to probability of detection or the probability density function, along with some of the required subroutines, are presented. All routines are double precision.

The routine SQUARE LAW DETECTOR, PART I does the matrix manipulations necessary to find the eigenvalues and eigenvectors. One can choose to find the probability density function for noise only in which case $\sigma_N^2 = 1$ is used. This choice will find intermediate results that can be used by SQUARE LAW DETECTOR, PART II, which allows for finding the probability density functions for signal plus noise cases. The other choice is for SQUARE LAW DETECTOR, PART I to obtain intermediate results which can be used by SQUARE LAW DETECTOR, PART II to find probability of detection for signal plus noise. In this case, the false alarm rate (FAR) or probability of false alarm is an input parameter to SQUARE LAW DETECTOR, PART I.

The calculations were broken into two programs due to memory size limitations of the computer used (Data General Nova 4). The programs require a great deal of memory to store the necessary matrices and to complete the matrix manipulations. Thus, at the termination of SQUARE LAW DETECTOR, PART I, the input parameters and only those vectors, matrices and other results necessary for completion of probability density functions or probabilities of detection are stored on disk, which can then be retrieved by SQUARE LAW DETECTOR, PART II.

The routines BUTTERWORTH RESPONSE FUNCTION, SAW RESPONSE FUNCTION, and FILTER are all function subroutines. BUTTERWORTH RESPONSE FUNCTION returns the complex response of a 1, 2, 3, or 4 pole Butterworth filter at a specified input frequency to filter 3 dB bandwidth ratio. SAW RESPONSE FUNCTION returns the complex response of a surface acoustic wave (SAW) filter at the specified input frequency to filter 3 dB bandwidth ratio. The routine FILTER will return the complex response of either a Butterworth or a SAW filter as described.

The routines as presented are configured to handle a rectangular input signal, SAW or Butterworth pre-filter and a Butterworth post filter. The programs could be easily adapted to other configurations by modifying the expressions for SN and SM (the series expressions for the low pass equivalent of the signal input) found in SQUARE LAW DETECTOR, PART II, and by substituting the desired filter response functions for BUTTERWORTH RESPONSE FUNCTION, SAW RESPONSE FUNCTION, and FILTER.

```
C*****************************************************************************
C
C       SQUARE LAW DETECTOR PROBLEM, PART I
C
C       DATA GENERAL FORTRAN 5 SOURCE FILENAME:  CA2R5.FR
C
C       PROGRAMMED BY CONNIE ADAMS
C
C       DEPARTMENT OF ELECTRICAL ENGINEERING    KANSAS STATE UNIVERSITY
C
C       FINAL REVISION DATE -- FEB 07, 1982
C
C*****************************************************************************
C
C       PURPOSE
C
C               This routine gives the output of a square law detector with a
C               pre-detection filter and a post-detection filter.
C
C       ROUTINE(S) CALLED BY THIS ROUTINE
C
C               BUTTER          finds response of a Butterworth filter
C               CLOSEP          KSU LIBRARY SUBROUTINE--closes the lineprinter
C               FILTER          finds response of a specified filter
C               OPENP           KSU LIBRARY SUBROUTINE--opens the lineprinter
C               RDATA           KSU LIBRARY SUBROUTINE--reads a file from disk
C               RS              EISPAK SUBROUTINE--finds eigenvalues and
C                                       eigenvectors (developed by ARGONNE
C                                       NATIONAL LABORATORY)
C               SINCDP          KSU LIBRARY FUNCTION SUBROUTINE--finds SIN(X)/X
C               WAIT            SYSTEM SUBROUTINE
C               WDATA           KSU LIBRARY SUBROUTINE--writes a file to disk
C               WNAME           KSU LIBRARY SUBROUTINE--checks disk for existing
C                                       file
C
C*****************************************************************************
C
        DIMENSION NAME(13),  SPDF(100)
        DOUBLE PRECISION T, WC1, WC2, WX, YMAX, YINC, GQR(76),
     1     LAMB1, LAMB2, HK, THETAK, HL, THETAL, LAMBS, GLAMBS,
     2     GLAMBD, GS, PHIS, GD, LAMBD, PHID, P(30,30), VAL(30),
     3     VEC(30,30), FV1(30), FV2(30), VAR, KK(30), Y, PDF(100), TEMP2,
     4     PI, MEAN, YMIN, TEMP3, NO, FF, TAU, PFA, TV, FAR, F, G,
     5     PFTV, COMP, UB, LB, TEMP1, G1
        DOUBLE PRECISION COMPLEX BUTTER, FILTER
C
C
1       FORMAT ('<012><015>', 'DISK FILE CONTAINING GQR DATA ? ')
2       FORMAT (S23)
3       FORMAT (5X, 'NPAIR = ', I2)
6       FORMAT (5X, 'NPOLE (PRE-FILTER)= ', 3X, I4 )
7       FORMAT (5X, 'T = ', F10.5, ' NANOSECONDS')
12      FORMAT ('<012><015>', 'RUN PROGRAM AGAIN ? ')
13      FORMAT (S1)
14      FORMAT (5X, 'NPAIR = ', 3X, F10.5)
18      FORMAT ('<012><015>')
```

```
21      FORMAT (5X, 'PRE FILTER CUT-OFF = ', F10.5, ' MHz'),
22      FORMAT (5X, 'POST FILTER CUT-OFF = ', F10.5, ' MHz'),
25      FORMAT ('<012><015><012><015>', 'PROBABILITY DENSITY FUNCTION', '<012>
     1          <015>')
27      FORMAT (5X, 'REFERENCE BANDWIDTH = ', F10.5, ' MHz')
28      FORMAT (5X, 'NOISE FIGURE = ', F10.5,  ' dB')
29      FORMAT ('OUTPUT PROBABILITY DENSITY FUNCTION TO DISK ? ')
30      FORMAT ('<012><015>PROBABILITY DENSITY FUNCTION STORED ON DISK FILE:  ',
     1           2X, S23)
31      FORMAT ('<012><015><012><015>', 5X, 'VARIANCE = ', F10.5)
36      FORMAT ('<012><015>', 5X, 'MEAN = ', 3X, F12.6)
38      FORMAT ('<012><015>', 5X, 'GAIN FOR UNITY THRESHOLD = ', F10.5, ' dB')
43      FORMAT (5X, 'NPOLE (POST-FILTER) = ', 3X, I4)
44      FORMAT (5X, 'SAW PRE-FILTER')
45      FORMAT (5X, 'BUTTERWORTH PRE-FILTER')
46      FORMAT (5X, 'FUNDAMENTAL FREQUENCY = ', F10.5, ' MHz')
47      FORMAT (5X, 'RECEIVER GAIN = ', F10.5, ' dB')
48      FORMAT (5X, PULSE WIDTH = ', F10.5, ' NANOSECONDS')
49      FORMAT (5X, 'NUMBER OF POSITIVE FOURIER COEFFICIENTS = ', I2)
50      FORMAT ('<012><015>', 'USER INPUT INFORMATION STORED IN DISK FILE:  ',
     &    S23)
51      FORMAT ('<012><015>', 'EIGENVALUES STORED IN DISK FILE:  ', S23)
52      FORMAT ('<012><015>', 'EIGENVECTORS STORED IN DISK FILE:  ', S23)
53      FORMAT ('<012><015>', 5X, 'REFERENCE BANDWIDTH = ', F12.6, ' MHz')
54      FORMAT (5X,'(FALSE ALARM RATE = ', F12.6, 2X, ' ALARMS/SEC)')
55      FORMAT (5X, 'NO = ', F12.6)
56      FORMAT ('FIND PROBABILITY DENSITY FUNCTION (NOISE ONLY) ? ')
57      FORMAT ('<012><015>', 5X, 'DESIRED PROBABILITY OF FALSE ALARM = ', F10.5)
58      FORMAT ('<012><015>', 5X, 'ACTUAL PROBABILITY OF FALSE ALARM = ', F12.6)
60      FORMAT (5X, '(ACTUAL FALSE ALARM RATE = ', F12.6, 'ALARMS/SEC)')
61      FORMAT ('<012><015>', 5X, 'GAIN FOR UNITY VARIANCE (NOISE ONLY) = ',
     &          F12.6, ' dB')
62      FORMAT ('USER PROVIDED PARAMETERS<012><015>')
63      FORMAT ('RESULTS<012><015>')
     &    ' dB')
C
C******************************************************************************
C
        ITTO = 10
        ITTI = 11
        ICHAN = 12
        PI = 3.141593
C
C       GET USER PROVIDED INFORMATION
C
100     CONTINUE
        TYPE
        ACCEPT 'NPAIR = ? ', NPAIR
125     CONTINUE
        ACCEPT 'PRE-FILTER: BUTTERWORTH = 1, SAW = 0   ', N
        IF (N.NE.1.AND.N.NE.0)  TYPE 'TRY AGAIN'
        IF (N.NE.1.AND.N.NE.0)  GO TO 125
        TYPE
        IF (N.EQ.0)  N1 = 0
        IF (N.EQ.0)  GO TO 130
        TYPE
```

```
      ACCEPT 'NUMBER OF POLES IN PRE-FILTER ? ', N1
      TYPE
130   CONTINUE
      ACCEPT 'NUMBER OF POLES IN POST-FILTER ? ', N2
      TYPE
      ACCEPT 'T = ? ', T
      NPAIRD2 = NPAIR/2
      TYPE
      ACCEPT 'CUT-OFF OF PRE-FILTER (HZ) = ', WC1
      TYPE
      ACCEPT 'CUT-OFF OF POST-FILTER (HZ) = ', WC2
      TYPE
      ACCEPT 'REFERENCE BANDWIDTH (WX) = ', WX
      TYPE
      ACCEPT 'NOISE FIGURE (dB) = ? ', F
      F = 10 ** (F/10.0)
      TYPE
      ACCEPT 'FUNDAMENTAL FREQUENCY = ? ', FF
      TYPE
      ACCEPT 'PULSE WIDTH = ? ', TAU
      TYPE
      ACCEPT 'NUMBER OF POSITIVE FOURIER COEFFICIENTS = ? ', NFC
132   CONTINUE
      ACCEPT 'FIND GAIN FOR TV = 1 OR FOR VAR = 1 (0 OR 1) ? ', NOP
      IF (NOP.NE.0.AND.NOP.NE.1)  GO TO 132
      IF (NOP.EQ.1)  GO TO 145
135   CONTINUE
      ACCEPT 'INPUT PFA OR FAR (0 OR 1) ? ', I1
      IF (I1.NE.1.AND.I1.NE.0)  TYPE 'TRY AGAIN'
      IF (I1.NE.1.AND.I1.NE.0)  GO TO 135
      IF (I1.EQ.1)  GO TO 140
      ACCEPT 'PROBABILITY OF FALSE ALARM = ? ', PFA
      FAR = 2.0 * WC2 * PFA
      GO TO 145
140   CONTINUE
      ACCEPT 'FALSE ALARM RATE = ? ', FAR
      PFA = FAR / (2.0 * WC2)
145   CONTINUE
      G = 10.0 ** 10
      NO = 4.0 * (10.0 ** -21) * G * F
C
C***********************************************************************************
C
C     READ GAUSS QUADRATURE TABLE FROM DISK FILE
C
150   CONTINUE
      WRITE (ITTO, 1)
      READ (ITTI, 2)  NAME(1)
      CALL RDATA (0, 'D', 2*NPAIR, NAME, GQR)
C
C
C***********************************************************************************
C
C     FIND P MATRIX
C
C     FIND UPPER LEFT SECTION
```

```
C
      INC = NPAIR + NPAIRD2 + 1
      DO 1300 K = 1, NPAIRD2
      LAMB1 = GQR(INC-K) * WX / WC1
      HK = DCABS(FILTER(N,N1,LAMB1))
      THETAK = DATAN2(DIMAG(FILTER(N,N1,LAMB1)),DREAL(FILTER(N,N1,LAMB1)))
              DO 1200 L = 1, NPAIRD2
              LAMB1 = GQR(INC-L) * WX / WC1
              HL = DCABS(FILTER(N,N1,LAMB1))
              THETAL = DATAN2(DIMAG(FILTER(N,N1,LAMB1)),
     &                DREAL(FILTER(N,N1,LAMB1)))
              LAMBS = (GQR(INC-K) + GQR(INC-L)) * WX / WC2
              GLAMBS = (GQR(INC-K) + GQR(INC-L)) * WX
              GLAMBD = (GQR(INC-K) - GQR(INC-L)) * WX
              GS = DCABS(BUTTER(N2,LAMBS))
              PHIS = DATAN2(DIMAG(BUTTER(N2,LAMBS)),DREAL(BUTTER(N2,LAMBS)))
              LAMBD = (GQR(INC-K) - GQR(INC-L)) * WX / WC2
              GD = DCABS(BUTTER(N2,LAMBD))
              PHID = DATAN2(DIMAG(BUTTER(N2,LAMBD)),DREAL(BUTTER(N2,LAMBD)))
              P(K,L) = 0.5 * HK * HL * GS * DCOS(GLAMBS * 2.0 * PI * T
     2          + THETAK + THETAL + PHIS) + 0.5 * HK * HL * GD * DCOS(GLAMBD
     3          * 2.0 * PI * T + THETAK - THETAL + PHID)
1200          CONTINUE
1300  CONTINUE
C
C     FIND UPPER RIGHT SECTION
C
      DO 1500 K = 1, NPAIRD2
      LAMB1 = GQR(INC-K) * WX / WC1
      HK = DCABS(FILTER(N,N1,LAMB1))
      THETAK = DATAN2(DIMAG(FILTER(N,N1,LAMB1)),DREAL(FILTER(N,N1,LAMB1)))
              DO 1400 I = NPAIRD2+1, NPAIR
              L = I - NPAIRD2
              LAMB1 = GQR(INC-L) * WX / WC1
              HL = DCABS(FILTER(N,N1,LAMB1))
              THETAL = DATAN2(DIMAG(FILTER(N,N1,LAMB1)),
     &                DREAL(FILTER(N,N1,LAMB1)))
              LAMBS = (GQR(INC-K) + GQR(INC-L)) * WX / WC2
              GLAMBS = (GQR(INC-K) + GQR(INC-L)) * WX
              GLAMBD = (GQR(INC-K) - GQR(INC-L)) * WX
              LAMBD = (GQR(INC-K) - GQR(INC-L)) * WX / WC2
              GS = DCABS(BUTTER(N2,LAMBS))
              PHIS = DATAN2(DIMAG(BUTTER(N2,LAMBS)),DREAL(BUTTER(N2,LAMBS)))
              GD = DCABS(BUTTER(N2,LAMBD))
              PHID = DATAN2(DIMAG(BUTTER(N2,LAMBD)),DREAL(BUTTER(N2,LAMBD)))
              P(K,I) = 0.5 * HK * HL * GS * DSIN(GLAMBS * 2.0 * PI * T
     1          + THETAK + THETAL + PHIS) - 0.5 * HK * HL * GD * DSIN(GLAMBD *
     2          2.0 * PI * T + THETAK - THETAL + PHID)
1400          CONTINUE
1500  CONTINUE
C
C     FIND LOWER LEFT SECTION
C
      DO 1700 K = NPAIRD2+1, NPAIR
              DO 1600 L = 1, NPAIRD2
              P(K,L) = P(L,K)
```

```
1600            CONTINUE
1700      CONTINUE
C
C         FIND LOWER RIGHT SECTION
C
          DO 1900 J = NPAIRD2+1, NPAIR
          K = J - NPAIRD2
          LAMB1 = GQR(INC-K) * WX / WC1
          HK = DCABS(FILTER(N,N1,LAMB1))
          THETAK = DATAN2(DIMAG(FILTER(N,N1,LAMB1)),DREAL(FILTER(N,N1,LAMB1)))
                DO 1800 I = NPAIRD2+1, NPAIR
                L = I - NPAIRD2
                LAMB1 = GQR(INC-L) * WX / WC2
                HL = DCABS(FILTER(N,N1,LAMB1))
                THETAL = DATAN2(DIMAG(FILTER(N,N1,LAMB1)),
     &                  DREAL(FILTER(N,N1,LAMB1)))
                LAMBS = (GQR(INC-K) + GQR(INC-L)) * WX / WC2
                GLAMBS = (GQR(INC-K) + GQR(INC-L)) * WX
                GLAMBD = (GQR(INC-K) - GQR(INC-L)) * WX
                LAMBD = (GQR(INC-K) - GQR(INC-L)) * WX / WC2
                GS = CABS(BUTTER(N2,LAMBS))
                PHIS = DATAN2(DIMAG(BUTTER(N2,LAMBS)),DREAL(BUTTER(N2,LAMBS)))
                GD = CABS(BUTTER(N2,LAMBD))
                PHID = DATAN2(DIMAG(BUTTER(N2,LAMBD)),DREAL(BUTTER(N2,LAMBD)))
                P(J,I) = -0.5 * HK * HL * GS * DCOS(GLAMBS * 2.0 * PI * T +
     1            THETAK + THETAL + PHIS) + 0.5 * HK * HL * GD * DCOS(GLAMBD
     2            * 2.0 * PI * T + THETAK - THETAL + PHID)
1800            CONTINUE
1900      CONTINUE
C
C*****************************************************************************
C
C         FORM DPD MATRIX
C
          DO 2140 K = 1, NPAIR
          IF (K.LE.NPAIRD2) LAMB1 = SQRT(2.0*NO*WX*GQR(NPAIRD2+1-K))
          IF (K.GT.NPAIRD2) LAMB1 = SQRT(2.0*NO*WX*GQR(NPAIR+1-K))
                DO 2130 L = 1, NPAIR
                IF (L.LE.NPAIRD2) LAMB2 = SQRT(2.0*NO*WX*GQR(NPAIRD2+1-L))
                IF (L.GT.NPAIRD2) LAMB2 = SQRT(2.0*NO*WX*GQR(NPAIR+1-L))
                P(K,L) = LAMB1 * P(K,L) * LAMB2
2130            CONTINUE
2140      CONTINUE
C
C*****************************************************************************
C
C         FIND EIGENVALUES AND EIGENVECTORS OF DPD MATRIX
C
          CALL RS(30, NPAIR, P, VAL, 1, VEC, FV1, FV2, IERR)
          IF (IERR.EQ.0)  GO TO 2200
          TYPE
          TYPE 'ERROR ON RETURN FROM EISPAK, ERROR CODE = ', IERR
          STOP
2200      CONTINUE
C
C****************************************************************************
```

```
C
C       FIND MEAN AND VARIANCE
C
        VAR = 0.0
        MEAN = 0.0
                DO 2250 K = 1, NPAIR
                TEMP2 = 4.0 * VAL(K) ** 2
                VAR = VAR + TEMP2
                MEAN = MEAN + 2.0 * VAL(K)
2250            CONTINUE
C
C*******************************************************************************
C
C       FIND KK
C
        DO 2400 K = 1, NPAIR
        KK(K) = 1.0
                DO 2350 I = 1, NPAIR
                IF (I.EQ.K.OR.VAL(K).EQ.0.0)  GO TO 2350
                KK(K) = KK(K) / (1.0 - (VAL(I)/VAL(K)))
                IF (KK(K).EQ.0.0)  GO TO 2375
2350            CONTINUE
2375    CONTINUE
2400    CONTINUE
        IF (NOP.EQ.1)  GO TO 3235
C
C*******************************************************************************
C
C       FIND THRESHOLD FOR THE GIVEN FALSE ALARM RATE, IF DESIRED
C
        COMP = PFA * 1.0D-3
        LB = 0.0
        UB = 1.0
        TV = UB
3220    CONTINUE
        PFTV = 0.0
                DO 3225 K = 1, NPAIR
                TEMP1 = 0.0
                TEMP2 = 0.0
                IF (VAL(K).GT.0.0)  TEMP1 = TV / (2.0 * VAL(K))
                TEMP2 = KK(K) * DEXP(-TEMP1)
                PFTV = PFTV + TEMP2
3225            CONTINUE
        IF (DABS(PFTV-PFA).LE.COMP)  GO TO 3230
        IF (PFTV.LT.PFA)  UB = TV
        IF (PFTV.LT.PFA)  TV = (TV + LB) / 2.0
        IF (PFTV.GT.PFA)  LB = TV
        IF (PFTV.GT.PFA)  TV = (UB + TV) / 2.0
        GO TO 3220
3230    CONTINUE
        G1 = G
        G = G / TV
        TV = 1.0
        NO = 4.0 * 10.0 ** -21 * G * F
C
C*******************************************************************************
```

```
C
C       FIND GAIN FOR VAR = 1, IF DESIRED
C
        IF (NOP.EQ.0)  GO TO 3240
3235    CONTINUE
        G1 = G
        G = G / DSQRT(VAR)
        NO = 4.0 * (10.0 ** -21) * G * F
        TV = 1.0
C
C******************************************************************************
C
C       ADJUST EIGENVALUES FOR CALCULATED RECEIVER GAIN
C
3240    CONTINUE
        DO 3250 K = 1, NPAIR
        VAL(K) = G / G1  * VAL(K)
3250    CONTINUE
C
C******************************************************************************
C
C       FIND MEAN AND VARIANCE FOR CALCULATED RECEIVER GAIN
C
        VAR = 0.0
        MEAN = 0.0
                DO 3260 K = 1, NPAIR
                TEMP2 = 4.0 * (VAL(K) ** 2)
                VAR = VAR + TEMP2
                MEAN = MEAN + 2.0 * VAL(K)
3260            CONTINUE
C
C******************************************************************************
C
C       OUTPUT USER PROVIDED INFORMATION AND CALCULATED DATA
C
3262    CONTINUE
        CALL WAIT
        CALL OPENP (ICHAN, 'SQUARE LAW DETECTOR PROBLEM')
        WRITE (ICHAN, 62)
        IF (N.EQ.0)  WRITE (ICHAN, 44)
        IF (N.EQ.1)  WRITE (ICHAN, 45)
        WRITE (ICHAN, 3)  NPAIR
        IF (N.EQ.1)  WRITE (ICHAN, 6)  N1
        WRITE (ICHAN, 43)  N2
        WRITE (ICHAN, 21)  WC1/1.0D6
        WRITE (ICHAN, 22)  WC2/1.0D6
        WRITE (ICHAN, 27)  WX/1.0D6
        WRITE (ICHAN, 46)  FF/1.0D6
        WRITE (ICHAN, 48)  TAU/1.0D-9
        WRITE (ICHAN, 49)  NFC
        WRITE (ICHAN, 7)   T/1.0D-9
        WRITE (ICHAN, 28)  10.0 * DLOG10(F)
        WRITE (ICHAN, 63)
        WRITE (ICHAN, 55)  NO
        IF (NOP.EQ.0)  WRITE (ICHAN, 38)  10 * DLOG10(G)
        IF (NOP.EQ.1)  WRITE (ICHAN, 61)  10 * DLOG10(G)
```

```
      WRITE (ICHAN, 31)  VAR
      WRITE (ICHAN, 36)  MEAN
      IF (NOP.EQ.0)  WRITE (ICHAN, 57)  PFA
      IF (NOP.EQ.0)  WRITE (ICHAN, 54)  FAR
      IF (NOP.EQ.0)  WRITE (ICHAN, 58)  PFTV
      IF (NOP.EQ.0)  WRITE (ICHAN, 60)  PFTV * 2.0 * WC2
      IF (NOP.EQ.0)  GO TO 3295
C
C*****************************************************************************
C
C     FIND PROBABILITY DENSITY FUNCTION, IF DESIRED
C
      WRITE (ITTO, 56)
      READ (ITTI, 13)  IANS
      IF (IANS.NE.'Y<0>')  GO TO 3295
      WRITE (ICHAN, 25)
      YMAX = MEAN + 1.5 * DSQRT(VAR)
      YMIN = MEAN - 1.5 * DSQRT(VAR)
      IF (YMIN.GT.1.0D-4)  GO TO 3270
      YMIN = MEAN - 1.0 * DSQRT(VAR)
      IF (YMIN.GT.1.0D-4)  GO TO 3270
      YMIN = MEAN - 0.5 * DSQRT(VAR)
      IF (YMIN.GT.1.0D-4)  GO TO 3270
      YMIN = MEAN
      IF (YMIN.EQ.0.0)  YMIN = YMIN + 0.01
3270  CONTINUE
      Y = YMIN
      YINC = 0.125 * DSQRT(VAR)
      I = 1
3280  CONTINUE
      PDF(I) = 0.0
              DO 3290 K = 1, NPAIR
              IF (VAL(K).LE.0.0)  GO TO 3290
              TEMP2 = EXP(-Y/(2.0*VAL(K)))
              IF (Y/(2.0*VAL(K)).GE.165.0)  TEMP2 = 0.0
              TEMP3 = KK(K)/(2.0*VAL(K))
              TEMP2 = TEMP2 * TEMP3
              PDF(I) = PDF(I) + TEMP2
              SPDF(I) = SNGL(PDF(I))
3290          CONTINUE
      WRITE (ICHAN, 26)  Y, PDF(I)
      Y = Y + YINC
      I = I + 1
      IF (Y.LE.YMAX)  GO TO 3280
C
C*****************************************************************************
C
C     OUTPUT PROBABILITY DENSITY FUNCTION TO DISK IF DESIRED
C
      WRITE (ITTO, 29)
      READ (ITTI, 13)  IANS
      IF (IANS.NE.'Y<0>')  GO TO 3225
      LENGTH = I - 1
      CALL WNAME ('FILE FOR PROBABILITY DENSITY FUNCTION ? ', NAME)
      CALL WDATA (0, 'R', LENGTH, NAME, SPDF)
      WRITE (ICHAN, 30)  NAME(1)
```

```
C
C******************************************************************************
C
C       OUTPUT DATA TO DISK
C
3295    CONTINUE
        GQR(61) = DFLOAT(N1)
        GQR(62) = DFLOAT(N2)
        GQR(63) = T
        GQR(64) = DFLOAT(NPAIR)
        GQR(65) = WC1
        GQR(66) = WC2
        GQR(67) = WX
        GQR(68) = NO
        GQR(69) = DFLOAT(N)
        GQR(70) = FF
        GQR(71) = TV
        GQR(72) = TAU
        GQR(73) = DFLOAT(NFC)
        GQR(74) = 10.0 * DLOG10(G)
        GQR(75) = 10.0 * DLOG10(F)
        GQR(76) = DFLOAT(NOP)
C
C       OUTPUT GQR VECTOR, USER PROVIDED PARAMETERS AND SOME RESULTS
C
        CALL WNAME ('FILE FOR GQR & USER DATA ? ', NAME)
        CALL WDATA(0, 'D', 76, NAME, GQR)
        WRITE (ICHAN, 50)  NAME(1)
C
C       OUTPUT EIGENVALUES AND EIGENVECTORS
C
        CALL WNAME ('FILE FOR EIGENVALUES ? ', NAME)
        CALL WDATA (0, 'D', NPAIR, NAME, VAL)
        WRITE (ICHAN, 51)  NAME(1)
        CALL WNAME ('FILE FOR EIGENVECTORS ? ', NAME)
        CALL WDATA (0, 'D', NPAIR * NPAIR, NAME, VEC)
        WRITE (ICHAN, 52)  NAME(1)
C
C******************************************************************************
C
C       CLOSE THE LINEPRINTER AND ASK IF PROGRAM IS TO BE RERUN
C
        CALL CLOSEP(ICHAN)
        WRITE (ITTO, 12)
        READ (ITTI, 13)  IANS
        IF (IANS.EQ.'Y<0>')  GO TO 100
        STOP
        END
```

```
C**********************************************************************
C
C       SQUARE LAW DETECTOR PROBLEM, PARTII
C
C       DATA GENERAL SOURCE FILENAME:  CA2P2R.FR
C
C       PROGRAMMED BY CONNIE ADAMS
C
C       DEPARTMENT OF ELECTRICAL ENGINEERING, KANSAS STATE UNIVERSITY
C
C       FINAL REVISION DATE -- FEB 07, 1982
C
C**********************************************************************
C
C       PURPOSE
C
C               This is the second part of the SQUARE LAW DETECTOR PROBLEM
C               program.
C
C       ROUTINE(S) CALLED BY THIS ROUTINE
C
C               BUTTER          finds response of a Butterworth filter
C               FILTER          finds response of a specified filter
C               SINCDP          KSU LIBRARY FUNCTION SUBROUTINE--finds SIN(X)/X
C               RDATA           KSU LIBRARY SUBROUTINE--reads a file from disk
C               WAIT            SYSTEM SUBROUTINE
C               WDATA           KSU LIBRARY SUBROUTINE--writes a file to disk
C               WNAME           KSU LIBRARY SUBROUTINE--checks disk for existing
C                                       file
C
C**********************************************************************
C
        DIMENSION NAME(13), PYSP(100)
        DOUBLE PRECISION T, WC1, WC2, WX, NO, GQR(76), VAL(30), VEC(30,30),
     &          PI, FF, FN, TAU, A, RR(30), RI(30), ZR, ZI, FM,
     &          SINCDP, LAMB1, RSUM, X1, X2, MEAN, VAR, TV, SS,
     &          X3, NU, PDET, COMP, Y, PY(100), G, F, PS, PSDBM,
     &          TEMP1, TEMP2, SNRDB, SNR

        DOUBLE PRECISION COMPLEX BUTTER, FILTER, HHG, EXP1, EXP2, HCHG,
     &          TEMP, FN, PM, Q, QT, PHG(30), SN, SM
C
C
1       FORMAT (S23)
2       FORMAT (5X, 'NPOLE (PRE-FILTER) = ', 3X, I4)
3       FORMAT (5X, 'T = ', 3X, F10.5, ' NANOSECONDS')
4       FORMAT ('<012><015>', 'RUN PROGRAM AGAIN? ')
5       FORMAT (S1)
6       FORMAT (5X, 'NPAIR = ', 3X, I2)
7       FORMAT ('<012><015>')
8       FORMAT (5X, 'PRE FILTER CUT-OFF = ', F10.5, ' MHz')
9       FORMAT (5X, 'POST FILTER CUT-OFF = ', F10.5, ' MHz')
10      FORMAT (5X, 'REFERENCE BANDWIDTH = ', F10.5, ' MHz')
11      FORMAT (5X, 'NO = ', G10.5)
12      FORMAT (5X, 'NPOLE (POST-FILTER) = ', 3X, I4)
13      FORMAT (5X, 'SAW PRE-FILTER')
```

```
14      FORMAT (5X, 'BUTTERWORTH PRE-FILTER')
15      FORMAT (5X, 'FUNDAMENTAL FREQUENCY = ', F10.5, ' MHZ')
16      FORMAT ('<012><015><012><015>', 5X, 'SIGNAL POWER = ', F10.5, 'dBm')
17      FORMAT (5X, 'PULSE WIDTH = ', F9.5, ' NANOSECONDS')
18      FORMAT (5X, 'NUMBER OF POSITIVE FOURIER COEFFICIENTS = ', I2)
21      FORMAT ('NAME OF DISK FILE CONTAINING USER INPUT DATA AND GQR? ')
22      FORMAT ('NAME OF DISK FILE CONTAINING EIGENVALUES? ')
23      FORMAT ('NAME OF DISK FILE CONTAINING EIGENVECTORS? ')
24      FORMAT (5X, 'Q/A**2 = (', G15.8, ',', G15.8, ')')
25      FORMAT ('RR(',I2,')/A = ', G15.8, 10X, 'RI(',I2,')/A = ', G15.8)
26      FORMAT ('<012><15>', 5X,'Q(T)/A**2 = (',G15.8,',',G15.8,')')
27      FORMAT (5X, 'MEAN = ', F10.5)
28      FORMAT (5X, 'VARIANCE = ', F10.5)
29      FORMAT ('OUTPUT PROBABILITY DENSITY FUNCTION TO DISK ? ')
31      FORMAT (5X, 'THRESHHOLD VOLTAGE = ', 3X, F10.5)
32      FORMAT (5X, 'INTEGRATION STEP SIZE = ', 3X, F10.5)
33      FORMAT (5X, 'PDF(', G15.8, ') = ', G15.8)
34      FORMAT ('<012><015>', 5X, 'PROBABILITY OF DETECTION = ', G15.8)
35      FORMAT (5X, 'COMPARISON VALUE = ', F10.5)
36      FORMAT (5X, 'PY(', G15.8, ') = ', G15.8)
37      FORMAT ('FIND PY(Y) ? ')
38      FORMAT (5X, 'RECEIVER GAIN = ', F10.5, 2X, 'dB')
39      FORMAT (5X, 'NOISE FIGURE = ', F10.5, 2X, 'dB')
40      FORMAT (5X, 'SIGNAL POWER AT THRESHOLD VOLTAGE = ', F10.5, 2X, 'dBm',
     &          '<012><015>')
42      FORMAT ('FIND PROBABILITY OF DETECTION FOR NEW SIGNAL POWER ? ')
43      FORMAT (5X, 'FALSE ALARM RATE = ', F10.5, 2X, 'ALARMS/SEC')
44      FORMAT ('FIND PDF FOR A NEW SNR ? ', Z)
45      FORMAT ('<012><015>',5X,'SIGNAL-TO-NOISE RATIO = ',F10.5,'dBm')
46      FORMAT ('<012><015>', 5X, 'PDF STORED ON DISK FILE :  ', S23)
47      FORMAT ('USER PROVIDED PARAMETERS<012><015>')
48      FORMAT ('PARAMETERS FROM PART I<012><015>')
49      FORMAT ('RESULTS<012><015>')
C
C*****************************************************************
C
        ITTO = 10
        ITTI = 11
        ICHAN = 12
        PI = 3.141593
C
C       READ GQR AND USER DEFINED DATA
C
100     CONTINUE
        TYPE
        WRITE (ITTO, 21)
        READ (ITTI, 1) NAME(1)
        CALL RDATA (0, 'D', 76, NAME, GQR)
        N1 = IFIX(SNGL(GQR(61)))
        N2 = IFIX(SNGL(GQR(62)))
        T = GQR(63)
        NPAIR = IFIX(SNGL(GQR(64)))
        NPAIRD2 = NPAIR/2
        WC1 = GQR(65)
        WC2 = GQR(66)
        WX = GQR(67)
```

```
      NO = GQR(68)
      N = IFIX(SNGL(GQR(69)))
      FF = GQR(70)
      TV = GQR(71)
      TAU = GQR(72)
      NFC = IFIX(SNGL(GQR(73)))
      G = GQR(74)
      F = GQR(75)
      NOP = IFIX(SNGL(GQR(76)))
C
C     READ EIGENVALUES AND EIGENVECTORS FROM DISK
C
      WRITE (ITTO, 22)
      READ (ITTI, 1)  NAME(1)
      CALL RDATA(0, 'D', NPAIR, NAME, VAL)
      WRITE (ITTO, 23)
      READ (ITTI, 1)  NAME(1)
      CALL RDATA (0, 'D', NPAIR * NPAIR, NAME, VEC)
      TYPE
      ACCEPT 'STEPSIZE FOR INTEGRATION = ? ', SS
      ACCEPT 'LOWEST VALUES OF PDF TO BE INCLUDED IN INTEGRATION = ?', COMP
      TYPE
C
C*****************************************************************
C
C     FIND RR, RI AND Q
C
      G = 10.0 ** (G/10.0)
      PSDBM = 10.0 * DLOG10(TV / (2.0 * G)) + 30.0
      TYPE 'SIGNAL POWER CORRESPONDING TO THRESHOLD = ', PSDBM
C
C     FINDING RR AND RI
C
      DO 300 K = 1, NPAIR
      IF (K.LE.NPAIRD2)  INC = NPAIR + NPAIRD2 + 1
      IF (K.GT.NPAIRD2)  INC = 2 * NPAIR + 1
      PHG(K) = (0.0, 0.0)
      LAMB1 = GQR(INC - K) * WX
              DO 200 I = 1, 2*NFC + 1
              FN = FF * DFLOAT(I-NFC-1)
              HHG = FILTER(N,N1,LAMB1/WC1) * FILTER(N,N1,FN/WC1) *
     &           BUTTER(N2, ((FN + LAMB1)/WC2))
              X1 = DCOS(2.0 * PI * (FN + LAMB1) * T)
              X2 = DSIN(2.0 * PI * (FN + LAMB1) * T)
              EXP1 = DCMPLX(X1, X2)
              HCHG = DCONJG(FILTER(N,N1,LAMB1/WC1)) * FILTER(N,N1,FN/WC1) *
     &           BUTTER(N2,(FN - LAMB1)/WC2)
              X1 = DCOS(2.0 * PI * (FN - LAMB1) * T)
              X2 = DSIN(2.0 * PI * (FN - LAMB1) * T)
              EXP2 = DCMPLX(X1, X2)
              SN = DCMPLX((TAU * FF * SINCDP(PI*FN*TAU)), 0.0)
              IF (K.LE.NPAIRD2)  TEMP = SN * (HHG * EXP1 + HCHG * EXP2)
              IF (K.GT.NPAIRD2)  TEMP = (0.0, -1.0) * SN * (HHG * EXP1 -
     &           HCHG * EXP2)
              PHG(K) = PHG(K) + TEMP
200           CONTINUE
```

```
300     CONTINUE
        DO 450 K = 1, NPAIR
        RR(K) = 0.0
        RI(K) = 0.0
                DO 400 L = 1, NPAIR
                IF (L.LE.NPAIRD2)  LAMB1 = DSQRT(2.0 * NO * WX * GQR(NPAIRD2
     &             + 1 - L))
                IF (L.GT.NPAIRD2)  LAMB1 = DSQRT(2.0 * NO * WX * GQR(NPAIR +
     &             1 - L))
                ZR = DREAL(PHG(L))
                ZI = DIMAG(PHG(L))
                RR(K) = RR(K) + ZR * LAMB1 * VEC(L,K)
                RI(K) = RI(K) + ZI * LAMB1 * VEC(L,K)
400             CONTINUE
450     CONTINUE
C
C       FINDING Q
C
        QT = (0.0, 0.0)
                DO 800 I = 1, 2 * NFC + 1
                FN = FF * DFLOAT(I-NFC-1)
                SN = DCMPLX((TAU * FF * SINCDP(PI*FN*TAU)), 0.0)
                PN = SN * FILTER(N, N1, FN / WC1)
                        DO 700 J = 1, 2 * NFC + 1
                        FM = FF * DFLOAT(J-NFC-1)
                        SM = DCMPLX((TAU * FF * SINCDP(PI*FM*TAU)), 0.0)
                        PM = SM * FILTER(N, N1, FM / WC1)
                        X1 = DCOS(2.0 * PI * (FN - FM) * T)
                        X2 = DSIN(2.0 * PI * (FN - FM) * T)
                        EXP1 = DCMPLX(X1, X2)
                        QT = QT + PN * DCONJG(PM) * BUTTER(N2, (FN-FM)/
     &                     WC2) * EXP1
700                     CONTINUE
800             CONTINUE
        RSUM = 0.0
                DO 900 K = 1, NPAIR
                RSUM = RSUM + (RR(K) ** 2 + RI(K) ** 2)/(4.0 * VAL(K))
900             CONTINUE
        Q = QT - RSUM
C
C
C**********************************************************************
C
C       FIND MEAN AND VARIANCE
C
925     CONTINUE
        IF (NOP.EQ.0)  ACCEPT 'SIGNAL POWER (dBm) FOR WHICH Pd DESIRED?', PSDBM
        IF (NOP.EQ.0)  PS = 10.0 ** (PSDBM/10.0 - 3.0)
        IF (NOP.EQ.0)  A = DSQRT(2.0 * G * PS)
        IF (NOP.EQ.1)  ACCEPT 'SIGNAL-TO-NOISE RATIO (dB) = ? ', SNRDB
        IF (NOP.EQ.1)  SNR = 10.0 ** (SNRDB/10.0)
        IF (NOP.EQ.1)  A = 2.0 * DSQRT(NO * WC1 * SNR)
        MEAN = DCABS(Q * A ** 2)
        VAR = 0.0
                DO 950 K = 1, NPAIR
                TEMP1 = (8.0 * VAL(K) ** 2 + (RR(K) * A) ** 2
```

```
     &                  + (RI(K) * A) ** 2) / (4.0 * VAL(K))
                MEAN = MEAN + TEMP1
                TEMP2 = 4.0 * VAL(K) ** 2 + (RR(K) * A) ** 2 +
     &                  (RI(K) * A) ** 2
                VAR = VAR + TEMP2
950             CONTINUE
C
C**********************************************************************
C
C       FIND PROBABILITY OF DETECTION (SIGNAL PLUS NOISE CASE)
C
        IF (NOP.EQ.1)  GO TO 1240
        NU = 0.0
        RSUM = 0.0
                DO 975 K = 1, NPAIR
                RSUM = RSUM + VAL(K)
975             CONTINUE
        PDET = (((TV - A ** 2 * DREAL(QT))/2.0 - RSUM) / PI) * SS
        NU = NU + SS
        IFLAG = 0.0
1000    CONTINUE
        RSUM = 0.0
                DO 1100 K = 1, NPAIR
                TEMP1 = ((RR(K) * A) ** 2 + (RI(K) * A) ** 2) /
     &                  (1.0 + 4.0 * VAL(K) ** 2 * NU ** 2)
                RSUM = RSUM + TEMP1
1100            CONTINUE
        RSUM = -0.5 * RSUM * NU ** 2
        X1 = DEXP(RSUM)/(2.0 * PI)
        LAMB1 = 1.0
        RSUM = 0.0
                DO 1200 K = 1, NPAIR
                TEMP1 = (((RR(K) * A) ** 2 + (RI(K) * A) ** 2) * VAL(K) * NU
     &                  **3 / (1.0 + 4.0 * VAL(K) ** 2 * NU ** 2) - DATAN(2.0 *
     &                  VAL(K) * NU))
                RSUM = RSUM + TEMP1
                TEMP2 = 4.0 * VAL(K) ** 2 * NU ** 2
                LAMB1 = LAMB1 * DSQRT(1.0 + TEMP2)
1200            CONTINUE
        LAMB1 = LAMB1 * NU
        X2 = DSIN((TV - DREAL(QT) * A**2) * NU + RSUM)
        X3 = X1 * X2 / LAMB1
        IF (DABS(X3).LT.COMP)  IFLAG = IFLAG + 1
        IF (DABS(X3).GE.COMP)  IFLAG = 0
        IF (IFLAG.GE.(IFIX(5.0 / SS)))  X3 = X3 * 0.5
        PDET = PDET + X3 * SS
        NU = NU + SS
        IF (IFLAG.LT.(IFIX(5.0 / SS)))  GO TO 1000
        PDET = 2.0 * PDET
        IF (NOP.EQ.0)  GO TO 1420
C
C****************************************************************************
C
C       FIND PY(Y) FOR A FEW VALUES OF Y, IF DESIRED
C
1240    CONTINUE
```

```
      I = 1
      Y = MEAN - 1.5 * DSQRT(VAR)
 1250 CONTINUE
      IFLAG = 0
      PY(I) = 0.0
      NU = 0.0
 1275 CONTINUE
      RSUM = 0.0
              DO 1300 K = 1, NPAIR
              RSUM = RSUM + ((RR(K) * A) ** 2 + (RI(K) * A) ** 2) /
     &                (1.0 + 4.0 * VAL(K) ** 2 * NU ** 2)
 1300         CONTINUE
      RSUM = -0.5 * RSUM * NU ** 2
      X1 = DEXP(RSUM)/(2.0 * PI)
      LAMB1 = 1.0
      RSUM = 0.0
              DO 1350 K = 1, NPAIR
              RSUM = RSUM + (((RR(K) * A) ** 2 + (RI(K) * A) ** 2)
     &                * VAL(K) * NU ** 3 / (1.0 + 4.0 * VAL(K) ** 2 *
     &                NU ** 2) - DATAN(2.0 * VAL(K) * NU))
              LAMB1 = LAMB1 * DSQRT(1.0 + 4.0 * VAL(K) ** 2 * NU ** 2)
 1350         CONTINUE
      X2 = DCOS((DREAL(QT) * A ** 2 - Y) * NU - RSUM)
      X3 = X1 * X2 / LAMB1
      IF (DABS(X3).LT.COMP)  IFLAG = IFLAG + 1
      IF (DABS(X3).GE.COMP)  IFLAG = 0
      IF (IFLAG.GE.(IFIX(5.0 / SS))) X3 = X3 * 0.5
      PY(I) = PY(I) + 2.0 * X3 * SS
      NU = NU + SS
      IF (IFLAG.LT.(IFIX(5.0 / SS))) GO TO 1275
      I = I + 1
      Y = Y + 0.125 * DSQRT(VAR)
      IF (Y.LE.(MEAN + 1.5 * DSQRT(VAR))) GO TO 1250
C
C**************************************************************************
C
C     OUTPUT USER DEFINED INFORMATION AND RESULTS
C
 1420 CONTINUE
      CALL WAIT
      CALL OPENP (ICHAN, 'SQUARE LAW DETECTOR PROBLEM, PART II')
      IF (N.EQ.0)  WRITE (ICHAN, 13)
      IF (N.EQ.1)  WRITE (ICHAN, 14)
      WRITE (ICHAN, 6)  NPAIR
      IF (N.EQ.1)  WRITE (ICHAN, 2)  N1
      WRITE (ICHAN, 12)  N2
      WRITE (ICHAN, 8)  WC1/1.0D6
      WRITE (ICHAN, 9)  WC2/1.0D6
      WRITE (ICHAN, 10)  WX/1.0D6
      WRITE (ICHAN, 11)  NO
      WRITE (ICHAN, 15)  FF/1.0D6
      WRITE (ICHAN, 17)  TAU/1.0D-9
      WRITE (ICHAN, 18)  NFC
      WRITE (ICHAN, 3)  T/1.0D-9
      IF (NOP.EQ.0)  WRITE (ICHAN, 31)  TV
      WRITE (ICHAN, 32)  SS
```

```
      WRITE (ICHAN, 35)  COMP
      WRITE (ICHAN, 38)  10.0 * DLOG10(G)
      WRITE (ICHAN, 39)  F
      WRITE (ICHAN, 26)  DREAL(QT), DIMAG(QT)
      WRITE (ICHAN, 24)  DREAL(Q), DIMAG(Q)
      WRITE (ICHAN, 27)  MEAN
      WRITE (ICHAN, 28)  VAR
      IF (NOP.EQ.1)  GO TO 1430
      WRITE (ICHAN, 16)  PSDBM
      WRITE (ICHAN, 34)  0.5 - PDET
      CALL CLOSEP(ICHAN)
      WRITE (ITTO, 42)
      READ (ITTI, 5)  IANS
      IF (IANS.EQ.'Y<0>')  GO TO 925
      GO TO 1460
1430  CONTINUE
      WRITE (ICHAN, 45)  SNRDB
      WRITE (ICHAN, 7)
      WRITE (ICHAN, 7)
      Y = MEAN - 1.5 * DSQRT(VAR)
            DO 1440 K = 1, I-1
            WRITE (ICHAN, 36)  Y, PY(K)
            PYSP(K) = SNGL(PY(K))
            Y = Y + 0.125 * DSQRT(VAR)
1440        CONTINUE
C
C     OUTPUT PY(Y) TO DISK IF DESIRED
C
      WRITE (ITTO, 29)
      READ (ITTI, 5)  IANS
      IF (IANS.NE.'Y<0>')  GO TO 1450
      CALL WNAME('FILE FOR PDF ? ', NAME)
      CALL WDATA (0, 'R', I-1, NAME, PYSP)
      WRITE (ICHAN, 46)  NAME(1)
1450  CONTINUE
      CALL CLOSEP(ICHAN)
      WRITE (ITTO, 44)
      READ (ITTI, 5)  IANS
      IF (IANS.EQ.'Y<0>')  GO TO 925
C
C*******************************************************************************
C
C     ASK IF PROGRAM IS TO BE RERUN
C
1460  CONTINUE
      WRITE (ITTO, 4)
      READ (ITTI, 5)  IANS
      IF (IANS.EQ.'Y<0>')  GO TO 100
      STOP
      END
```

```
C***********************************************************************
C
C       BUTTERWORTH RESPONSE FUNCTION (DOUBLE PRECISION VERSION)
C
C       DATA GENERAL SOURCE FILENAME:  BUTTER.FR
C
C       PROGRAMMED BY FRED RATCLIFFE
C
C       DEPARTMENT OF ELECTRICAL ENGINEERING, KANSAS STATE UNIVERSITY
C
C       FINAL REVISION DATE -- APRIL 15, 1981
C
C***********************************************************************
C
C       CALLING SEQUENCE
C
C               COMPLEX RESPONSE = BUTTER (NPOLE, WRATIO)
C
C       PURPOSE
C
C               This routine calculates the response of a Butterworth
C               filter at a specified frequency.
C
C       ROUTINE(S) CALLED BY THIS ROUTINE
C
C               NONE
C
C       ARGUMENT(S) REQUIRED FROM THE CALLING ROUTINE
C
C               NPOLE   number of filter poles (1-4)
C
C               WRATIO  ratio of frequency at which response is desired
C                       to 3db cutoff frequency of filter (W / Wcutoff)
C
C       ARGUMENT(S) SUPPLIED  TO  THE CALLING ROUTINE
C
C               NONE
C
C***********************************************************************
C
C       NOTE 1: This routine makes no checks on the validity
C               of the data supplied by the calling routine.
C
C       NOTE 2: The name BUTTER must be declared complex
C               by the calling routine.
C
C***********************************************************************
C
        DOUBLE PRECISION COMPLEX FUNCTION BUTTER (NPOLE, WRATIO)
        DOUBLE PRECISION WRATIO, ANGLE, TEMP
C
C * * * * * * * * * *
C
C       CALCULATE NUMERATOR ANGLE
C
```

```
      GO TO (100, 200, 300, 400)  NPOLE
100   CONTINUE
C
      ANGLE = - DATAN (WRATIO)
      GO TO 600
200   CONTINUE
C
      ANGLE = - DATAN2 (1.414 * WRATIO, 1.0 - WRATIO ** 2)
      GO TO 600
300   CONTINUE
C
      ANGLE = - DATAN (WRATIO) - DATAN2 (WRATIO, 1.0 - WRATIO ** 2)
      GO TO 600
400   CONTINUE
C
      ANGLE = - DATAN2 (0.765 * WRATIO, 1.0 - WRATIO ** 2)
     %        - DATAN2 (1.848 * WRATIO, 1.0 - WRATIO ** 2)
600   CONTINUE
C
C * * * * * * * * * *
C
C     CALCULATE COMPLEX RESPONSE
C
      TEMP = DSQRT (1.0 + WRATIO ** (2 * NPOLE))
      BUTTER = DCMPLX (DCOS (ANGLE) / TEMP, DSIN (ANGLE) / TEMP)
      RETURN
      END
```

```
C***********************************************************************
C
C       SAW RESPONSE FUNCTION (DOUBLE PRECISION VERSION)
C
C       DATA GENERAL FORTRAN 5 SOURCE FILENAME:  SAW.FR
C
C       PROGRAMMED BY FRED RATCLIFFE
C
C       DEPARTMENT OF ELECTRICAL ENGINEERING, KANSAS STATE UNIVERSITY
C
C       FINAL REVISION DATE -- JUNE 29, 1981
C
C***********************************************************************
C
C       CALLING SEQUENCE
C
C               COMPLEX RESPONSE = SAW (WRATIO)
C
C       PURPOSE
C
C               This routine calculates the response of a SAW
C               filter at a specified frequency.
C
C       ROUTINE(S) CALLED BY THIS ROUTINE
C
C               SINC            KSU LIBRARY FUNCTION SUBROUTINE--finds SIN(X)/X
C
C       ARGUMENT(S) REQUIRED FROM THE CALLING ROUTINE
C
C               WRATIO  ratio of frequency at which response is desired
C                       to 3db cutoff frequency of filter (W / Wcutoff)
C
C       ARGUMENT(S) SUPPLIED  TO  THE CALLING ROUTINE
C
C               NONE
C
C***********************************************************************
C
C       NOTE 1: This routine makes no checks on the validity
C               of the data supplied by the calling routine.
C
C       NOTE 2: The name SAW must be declared double precision complex
C               by the calling routine.
C
C***********************************************************************
C
        DOUBLE PRECISION COMPLEX FUNCTION SAW (WRATIO)
        DOUBLE PRECISION CONSTANT, PI, ANGLE, SINCDP, WRATIO
C
C       INITIALIZATION
C
        CONSTANT = 0.566
        PI = 3.141593
        ANGLE = PI * WRATIO * CONSTANT
C
```

```
C       CALCULATE RESPONSE
C
        SAW = SINCDP (0.4 * ANGLE) * (SINCDP (ANGLE) + (0.23 / 0.54) *
     &      (SINCDP (ANGLE + PI) + SINCDP (ANGLE - PI)))
        RETURN
        END
```

```
C***********************************************************************
C
C       FILTER (DOUBLE PRECISION VERSION)
C
C       DATA GENERAL FORTRAN 5 SOURCE FILENAME:  FILTER.FR
C
C       PROGRAMMED BY CONNIE ADAMS
C
C       DEPARTMENT OF ELECTRICAL ENGINEERING    KANSAS STATE UNIVERSITY
C
C       FINAL REVISION DATE -- JUNE 29, 1981
C
C***********************************************************************
C
C       CALLING SEQUENCE
C
C               COMPLEX RESPONSE = FILTER (NF, NPOLE, WRATIO)
C
C       PURPOSE
C
C               This routine determines the response of either a SAW
C               filter or a BUTTERWORTH filter.
C
C       ROUTINE(S) CALLED BY THIS ROUTINE
C
C               SAW               finds response of a SAW filter
C               BUTTER            finds response of a Butterworth filter
C
C       ARGUMENT(S) REQUIRED FROM THE CALLING ROUTINE
C
C               NF      an integer indicating the type of filter used
C                       0 = SAW filter
C                       1 = BUTTERWORTH filter

C
C       ARGUMENT(S) SUPPLIED  TO  THE CALLING ROUTINE
C
C               NONE
C
C***********************************************************************
C
C       NOTE 1: This subroutine makes no checks on the validity
C               of the data supplied by the calling routine.
C
C       NOTE 2: Argument(s) supplied by the calling routine are
C               not modified by this subroutine.
C
C***********************************************************************
C
        DOUBLE PRECISION COMPLEX FUNCTION FILTER (NF, NPOLE, WRATIO)
        DOUBLE PRECISION COMPLEX BUTTER, SAW
        DOUBLE PRECISION WRATIO
C
C       FIND FILTER RESPONSE
C
```

```
IF (NF.EQ.1)  FILTER = BUTTER(NPOLE, WRATIO)
IF (NF.EQ.0)  FILTER = SAW(WRATIO)
RETURN
END
```

REFERENCES

[1] M. Kac and A. J. F. Siegert, "On the Theory of Noise in Radio Receivers with Square Law Detectors", Journal of Applied Physics, Vol. 18, No. 4, pp. 383-397, 1947.

[2] J. I. Marcum, "A Statistical Theory of Target Detection by Pulsed Radar," RAND Research Memo RM-754, December, 1947. (Reissued in Transactions of IRE Professional Group on Information Theory, Vol. IT-6, No. 2, pp. 59-144, April, 1960).

[3] A. M. Yoglom, An Introduction to the Theory of Stationary Random Functions, translated by R. A. Silverman, Prentice-Hall, Inc., New Jersy, 1962.

[4] M. Schwartz, W. R. Bennett, and S. Stein, Communication Systems and Techniques, McGraw-Hill, Inc., New York, 1966.

[5] S. T. Kwon and N. M. Shehadeh, "Analysis of Incoherent Frequency-Shift Keyed Systems," IEEE Transactions on Communications, Vol. COM-23, No. 11, pp. 1331-1339, November, 1975.

[6] M. I. Skolnik, Introduction to Radar Systems, 2nd Edition, McGraw-Hill Book Company, New York, 1962.

ACKNOWLEDGMENTS

I would like to express my appreciation to Dr. Donald R. Hummels for his help in the development and preparation of this document; to Fred Ratcliffe for the use of his subroutines and for his help in programming matters; to my parents, Jack and Pat Adams, for their constant encouragement; and to my husband, Tom Crice, for his extreme patience and understanding throughout the entire process of my studies.

THE INFLUENCE OF FILTER SELECTION ON DETECTION PROBABILITY FOR RECEIVERS USING SQUARE-LAW DETECTION, A GENERAL APPROACH

by

CONNIE ADAMS

B.A., Ottawa University, 1975

AN ABSTRACT OF A MASTER'S THESIS

Submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1982

ABSTRACT

This document involves the development of numerical methods for finding the probability of detection for a square-law detector system comprised of an RF pre-filter, a square-law envelope detector, and a video post filter. The input to the system is an arbitrary signal plus noise with the filter configuration also being arbitrary. All results are based on a single sample of the output signal plus noise with the sample time corresponding to the signal peak.

Computer programs are presented which are capable of finding probability density functions and probabilities of detection for a system with a rectangular input signal, a surface acoustic wave (SAW) or Butterworth pre-filter, and a Butterworth post filter. The programs yield good accuracy provided the pre- to post filter bandwidth ratio does not exceed 40 and the probability of detection is sought for a system output value that does not fall in the tails of the probability density function.

Probability density functions and probability of detection curves are presented for some specific cases of filter configuration and input signal shape. Results for a classical case are compared to previous work.