NUWC-NPT Technical Report 11,262
12 December 2000

Saddlepoint Approximation and First-Order Correction Term to the Joint Probability Density Function of M Quadratic and Linear Forms in K Gaussian Random Variables with Arbitrary Means and Covariances

Albert H. Nuttall
Surface Undersea Warfare Directorate

NAVSEA
NEWPORT
Undersea Warfare Center Division

Naval Undersea Warfare Center Division
Newport, Rhode Island

Approved for public release; distribution is unlimited.

PREFACE

The work described in this report was funded by Project No. A101401, "Automatic Signal Classification," principal investigator Stephen G. Greineder (Code 2121). The sponsoring activity is Office of Naval Research, program manager John Tague (Code 321US).

The technical reviewer for this report was Phillip L. Ainsleigh (Code 2122).

Reviewed and Approved: 12 December 2000

Patricia J. Dean
Director, Surface Undersea Warfare

REPORT DOCUMENTATION PAGE			*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE 12 December 2000	3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE

Saddlepoint Approximation and First-Order Correction Term to the Joint Probability Density Function of M Quadratic and Linear Forms in K Gaussian Random Variables with Arbitrary Means and Covariances

5. FUNDING NUMBERS

6. AUTHOR(S)

Albert H. Nuttall

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Undersea Warfare Center Division
1176 Howell Street
Newport, RI 02841-1708

8. PERFORMING ORGANIZATION REPORT NUMBER

TR 11,262

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5160

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Let w be a Kx1 Gaussian random vector with arbitrary Kx1 mean vector r and KxK covariance matrix R. The general quadratic and linear forms of interest are the M random scalars

z(m) = w′ P(m) w + p(m)′ w +q(m) for m=1:M,

where KxK matrix P(m), Kx1 vector p(m), and scalar q(m) contain arbitrary constants for m=1:M. The joint probability density function (PDF) of Mx1 random vector z = [z(1) ... z(M)]′ at an arbitrary point in M-dimensional space is desired.

An exact expression for the joint moment generating function (MGF) of random vector z is derived. The inability (analytic and numerical) to perform the M-dimensional inverse Laplace transform back to the PDF domain requires use of the saddlepoint appproximation (SPA) to obtain useful numerical values for the desired PDF of z. A first-order correction term to the SPA is also employed for more accuracy, which requires fourth-order partial derivatives of the joint cumulant generating function (CGF).

Derivation of the fourth-order partial derivatives of the CGF involves some interesting and useful matrix manipulations which are fully developed. Two MATLAB programs for the entire SPA procedure (with correction term) are presented in this report.

14. SUBJECT TERMS			15. NUMBER OF PAGES
Saddlepoint Approximation	Quadratic Forms	Correction Term	63
Cumulant Generating Function	Probability Density Function	Moment Generating Function	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

TABLE OF CONTENTS

LIST OF ABBREVIATIONS, ACRONYMS, AND SYMBOLS

$a(n)$	Autocorrelation estimate, equation (2)
B_m	m-th matrix in shortcut notation, equation (59)
$B(\lambda,m)$	m-th K×K auxiliary matrix, m=1:M, equation (55)
$\{c(k)\}$	Constants, k=1:K
$c(m)$	m-th condensed constants, equation (16)
$c(n)$	Cyclic correlation estimate, equation (6)
$C(m)$	m-th condensed K×K matrix, equation (16)
C_m	m-th contour of integration, equation (27)
CGF	Cumulant generating function, equation (25)
$CGF_{1,2}$	First two parts of CGF $\chi(\lambda)$, equation (39)
c_t	First-order correction term, equation (37)
c_4, c_{3a}, c_{3b}	Three components of correction term, equation (35)
$D(\lambda)$	K×K matrix function, equation (19)
det	Determinant, equation (25)
$E\{t\}$	Expectation of random variable **t**
$E(\lambda)$	K×K eigenvalue matrix of $D(\lambda)$, equation (48)
$\{e_k(\lambda)\}$	Eigenvalues of matrix $E(\lambda)$, k=1:K, equation (48)
FFT	Fast Fourier transform
g	Normalized Gaussian random vector, equation (14)
$\{g(k)\}$	Components of random vector **g**, equation (14)
$G(\lambda)$	M×1 Gradient vector, equation (32)
$H(\lambda)$	M×M Hessian matrix, equation (31)
H	M×M Hessian matrix at saddlepoint, equation (34)
K	Number of Gaussian random variables
M	Number of quadratic and linear forms

LIST OF ABBREVIATIONS, ACRONYMS, AND SYMBOLS (Cont'd)

MGF	Moment generating function
n	Delay, equation (2)
PDF	Probability density function
$p(g)$	Probability density of Gaussian $\mathbf{g}$, equation (22)
$p(m)$	m-th K×1 vector in quadratic form, equation (13)
$P(m)$	m-th K×K matrix in quadratic form, equation (13)
$\tilde{P}(m;k,l)$	Auxiliary constants, equation (11)
$p(z)$	Probability density function at z, equation (27)
$p_0(z)$	Saddlepoint approximation, equation (30)
$p_1(z)$	First-order saddlepoint approximation, equation (36)
$p_a(z)$	Rational saddlepoint approximation, equation (38)
$p_e(z)$	Exponential saddlepoint approximation, equation (38)
QAL	Quadratic and linear form
$q(m)$	m-th constant in quadratic form, equation (13)
$q(\lambda)$	K×K inverse to matrix $Q(\lambda)$, equation (53)
$Q(\lambda)$	K×K matrix function, equation (24)
quadlinspa	MATLAB program, appendix D
quadspa	MATLAB program, appendix E
r	K×1 mean vector of random vector $\mathbf{w}$
R	K×K covariance matrix of random vector $\mathbf{w}$
RV	Random variable or random vector
S	Cholesky decomposition of matrix R, equation (16)
SOCT	Second-order correction term, equation (38)
SP	Saddlepoint, equation (29)
SPA	Saddlepoint approximation

LIST OF ABBREVIATIONS, ACRONYMS, AND SYMBOLS (Cont'd)

SPA0	Zeroth-order saddlepoint approximation, equation (30)
SPA1	First-order saddlepoint approximation, equation (36)
$t(\lambda)$	Kx1 vector function, equation (20)
tr	trace, equations (51) and (C-1)
T	MxM inverse matrix of H, equation (34)
$u(\lambda)$	scalar function, equation (21)
$\mathbf{u},\mathbf{v}$	Random sequences, equation (7)
$v(m)$	m-th condensed Kx1 vector, equation (16)
$V(\lambda)$	Eigenvector matrix of $D(\lambda)$, equation (48)
$\mathbf{w}$	Gaussian Kx1 random vector
$\{\mathbf{w}(k)\}$	Components of random vector $\mathbf{w}$, k=1:K
$\mathbf{x},\mathbf{y}$	Random sequences, equation (9)
z	Mx1 vector, field point of interest
$\{z(m)\}$	Components of vector z, m=1:M
$\mathbf{z}$	Quadratic and linear form or Mx1 random vector
$\{\mathbf{z}(m)\}$	Components of random vector $\mathbf{z}$, m=1:M
$\chi(\lambda)$	Cumulant generating function, equation (25)
χ_{jklm}	Partial derivatives of χ at SP, equation (33)
$\chi_{1,2,3}$	Three parts of CGF $\chi(\lambda)$, equation (39)
∂	Partial derivative
δ_{jk}	Kronecker delta
λ	Mx1 vector in MGF domain
$\{\lambda(m)\}$	Components of vector λ, m=1:M
$\hat{\lambda} = \hat{\lambda}(z)$	Saddlepoint, equation (29)

LIST OF ABBREVIATIONS, ACRONYMS, AND SYMBOLS (Cont'd)

$\mu(\lambda)$	Moment generating function, equation (18)
-1 (super)	Inverse
′ (prime)	Transpose
· (dot)	Derivative

SADDLEPOINT APPROXIMATION AND FIRST-ORDER CORRECTION TERM TO THE JOINT PROBABILITY DENSITY FUNCTION OF M QUADRATIC AND LINEAR FORMS IN K GAUSSIAN RANDOM VARIABLES WITH ARBITRARY MEANS AND COVARIANCES

INTRODUCTION

When K normalized Gaussian random variables (RVs) $\{g(k)\}$ are squared and summed, the resultant z is called a chi-squared variate with K degrees of freedom, and the probability density function (PDF) of RV z is available in a closed form involving an exponential. If constants $\{c(k)\}$ are added to each of the RVs $\{g(k)\}$ before squaring and summation, the PDF of the resultant z is called a noncentral chi-squared variate with K degrees of freedom, and is again available in a closed form, this time involving a Bessel function and an exponential. However, virtually any additional complexity beyond this case results in a RV z for which the corresponding PDF is analytically intractable.

However, in these one-dimensional cases of RV z, the moment generating function (MGF) of z is frequently available in closed form, and a numerical technique involving fast Fourier transforms (FFTs) can be efficiently employed to get numerous quick and accurate values for the PDF, as well as the exceedance distribution function, at arbitrary points of interest, whether near the mean of RV z or on the tails of the distribution of z

(references 1 through 5). Thus, the one-dimensional statistical problem involving quadratic forms of Gaussian RVs is in good shape numerically.

The situation in M dimensions is much more difficult. Even if the joint M-dimensional MGF of a random vector (RV), denoted by column vector $\mathbf{z} = [z(1) \ldots z(M)]'$, is available in closed form, its inverse M-dimensional Laplace or Fourier transform back into the PDF domain cannot be accomplished analytically, except in the simplest of cases. Also, numerical evaluation of the pertinent M-dimensional integral for the joint PDF cannot be done accurately for M greater than four or so. These conditions force acceptance of an approximation to the M-dimensional PDF of RV $\mathbf{z}$; also, they force the effort to be concentrated on the evaluation of the joint PDF at very few points in M-dimensional PDF space, due to the extensive numerical effort and execution time involved in the accurate evaluation of multiple integrals.

The M-dimensional PDF approximation adopted here is that obtained via the saddlepoint (SP) method, with a first-order correction term (reference 6, page 180). The saddlepoint approximation (SPA) is accurate on the tails of the joint PDF, as well as near the mean of the distribution. For its evaluation, the SPA requires the calculation of some partial derivatives of the joint MGF up through fourth-order; evaluation of these derivatives will consume much of the effort in this report.

EXAMPLE PROBLEMS

CORRELATION ESTIMATES

Let $\mathbf{w} = [\mathbf{w}(1) \ldots \mathbf{w}(K)]'$ be a K×1 real Gaussian RV with K×1 mean vector r and K×K positive-definite covariance matrix R; that is,

$$E\{\mathbf{w}\} = r \ , \quad E\{(\mathbf{w} - r)(\mathbf{w} - r)'\} = R \ , \tag{1}$$

and E{ } denotes an expectation. An autocorrelation estimate of sequence $\mathbf{w}$ at delay n is available according to

$$\mathbf{a}(n) = \sum_{k=n+1}^{K} \mathbf{w}(k)\ \mathbf{w}(k - n) \quad \text{for } n=0:K-1 \ . \tag{2}$$

Suppose, for example, that only the correlation estimates at delays n = 0, 1, 3, and 7 are of interest; that is, M = 4 and RV $\mathbf{z}$ has components

$$\mathbf{z}(1) = \mathbf{a}(0), \quad \mathbf{z}(2) = \mathbf{a}(1), \quad \mathbf{z}(3) = \mathbf{a}(3), \quad \mathbf{z}(4) = \mathbf{a}(7) \ . \tag{3}$$

The problem of interest is to obtain the joint PDF of RV $\mathbf{z}$ for arbitrary sample size K and statistics r and R.

The quantities in equations (2) and (3) can be written as quadratic forms

$$\mathbf{z}(m) = \mathbf{w}'\ P(m)\ \mathbf{w} \quad \text{for } m=1:M \ , \tag{4}$$

where, for example, K×K matrices

$$P(1) = \begin{bmatrix} 1 & 0 & 0 & \cdots & \\ 0 & 1 & 0 & & \\ 0 & 0 & 1 & & \\ \vdots & & & & \\ & & & & \end{bmatrix}, \qquad P(2) = \frac{1}{2}\begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & \\ 1 & 0 & 1 & 0 & & \\ 0 & 1 & 0 & 1 & & \\ \vdots & & & & & \\ & & & & & \end{bmatrix}. \tag{5}$$

Matrix P(2) is nonzero only on the super- and sub-diagonals numbered 1; matrix P(3) is nonzero only on super- and sub-diagonals 3; and matrix P(4) is nonzero only on super- and sub-diagonals 7.

If the sample mean is subtracted from data sequence **w** prior to calculation of correlation estimates (2), the quadratic forms for RV z in equation (4) still hold, but the elements of the matrices {P(m)} for m=1:M in equation (5) are changed. For example, the j,k element of matrix P(1) is now $\delta_{jk} - 1/K$ instead of δ_{jk}, where δ_{jk} is the Kronecker delta; the remaining matrices {P(m)} are more complicated, but each element in matrix P(m) can be evaluated by means of a single sum.

If the correlation estimates are to be unbiased, additional scale factors are required in equations (2) or (3). Again, the quadratic forms in equation (4) are appropriate, but the elements of matrices {P(m)} require additional calculations involving the particular scale factors adopted.

Equation (2) involves an aperiodic correlation of data **w**. The extension to cyclic correlation estimates {c(n)} can also be formulated in terms of quadratic forms (4). Consider the cyclic correlation estimate at delay n = 1:

$$\mathbf{z}(2) = \mathbf{c}(1) = \mathbf{a}(1) + \mathbf{w}(1)\ \mathbf{w}(K)\ , \tag{6}$$

where the added term represents wraparound. This RV immediately fits equation (4) if the two corner elements (upper right and lower left) in matrix P(2) (equation (5)) are changed from 0 to 1. Instead, for delay n = 2, **c**(2) requires four changes in its P matrix; namely, from 0 to 1 of the four elements immediately bordering the two corner elements. This procedure extends to any delay n; the corresponding P matrix for cyclic correlation estimate **c**(n) will have K 1s on the n-th super- and sub-diagonals and their wraparound extensions.

Cross-correlation estimates from two different-length data sequences **u** and **v** can be written in the form

$$\mathbf{z}(m) = \mathbf{u}'\ A(m)\ \mathbf{v} \quad \text{for } m=1{:}M\ , \tag{7}$$

where RV **u** is J×1, RV **v** is K×1, and matrix A(m) is J×K. By defining augmented (J+K)×1 RV **w** as [**u**′ **v**′]′, equation (7) may be reformulated as

$$\mathbf{z}(m) = \mathbf{w}'\ P(m)\ \mathbf{w} \quad \text{for } m=1{:}M\ , \tag{8}$$

where matrix P(m) is (J+K)×(J+K) for m=1:M. Thus, cross-correlation estimates obtained from two different sequences can also be expressed as quadratic forms of a concatenated sequence.

FILTERED SQUARED DATA

Suppose data $\mathbf{w}$ in equation (1) is processed as follows:

$$\mathbf{x} = \mathrm{A}\ \mathbf{w}\ , \quad \mathbf{y}(n) = \mathbf{x}(n)^2 \quad \text{for } n=1:N\ , \quad \mathbf{z} = \mathrm{B}\ \mathbf{y}\ , \tag{9}$$

where matrix A is N×K, matrix B is M×N, and $\mathbf{y} = [\mathbf{y}(1)\ \ldots\ \mathbf{y}(N)]'$. Thus, N×1 RV $\mathbf{x}$ is a filtered (linearly transformed) version of data $\mathbf{w}$, which is then squared and subjected to additional filtering, resulting in M×1 RV $\mathbf{z}$. The problem is to determine the joint M-dimensional PDF of RV $\mathbf{z}$.

By combining the operations in equation (9), the component RVs $\{\mathbf{z}(m)\}$ of $\mathbf{z}$ can be expressed as

$$\mathbf{z}(m) = \sum_{k,l=1}^{K} \tilde{P}(m;k,l)\ \mathbf{w}(k)\ \mathbf{w}(l) \quad \text{for } m=1:M\ , \tag{10}$$

where constants

$$\tilde{P}(m;k,l) = \sum_{n=1}^{N} B(m,n)\ A(n,k)\ A(n,l) \quad \text{for } m=1:M;\ k,l=1:K. \tag{11}$$

Thus, the RVs $\{\mathbf{z}(m)\}$ in equations (9) and (10) can again be expressed as quadratic forms (4), where K×K matrices

$$P(m) = [\tilde{P}(m;k,l);\ k,l=1:K] \quad \text{for } m=1:M \tag{12}$$

in terms of its elements calculated in equation (11). That is, the classical filter-square-filter operation is basically a problem in finding the joint PDF of several statistically dependent quadratic forms.

PROBLEM FORMULATION

QUADRATIC AND LINEAR FORMS OF INTEREST

The formulations above all resulted in purely quadratic forms for RV $\mathbf{z} = [z(1) \ldots z(M)]'$. More generally, interest here will be concentrated on the M quadratic and linear (QAL) forms

$$z(m) = \mathbf{w}' \, P(m) \, \mathbf{w} + p(m)' \, \mathbf{w} + q(m) \quad \text{for } m=1:M \,, \tag{13}$$

where RV $\mathbf{w}$ is K×1, matrix P(m) is K×K, vector p(m) is K×1, and scalar q(m) is 1×1. Also, every matrix P(m) for m=1:M is symmetric without loss of generality (see appendix A).

RV $\mathbf{w}$ is presumed to have joint Gaussian statistics with K×1 mean vector r and K×K covariance matrix R, as in equation (1). Thus, equation (13) exhibits the most general second-order forms in correlated Gaussian RVs with arbitrary statistics. Since all M components of RV $\mathbf{z}$ in equation (13) utilize the same K×1 RV $\mathbf{w}$ but in different combinations, these M components {z(m)} are statistically dependent on each other, in addition to being non-Gaussian. These complications are what force the need to resort to an approximation for the desired joint PDF of RV $\mathbf{z}$.

There are five types of input information required to completely specify the QAL problem posed in equation (13). They are: the K×1 mean vector r of K×1 RV $\mathbf{w}$, the K×K covariance matrix R of K×1 RV $\mathbf{w}$, the M matrices {P(m)} of size K×K, the M vectors {p(m)} of size K×1, and the M scalars {q(m)} of size 1×1.

CONDENSATION OF QUADRATIC AND LINEAR PROBLEM

The $K \times 1$ RV $\mathbf{w}$ can be expressed in terms of a set of K normalized RVs $\mathbf{g} = [g(1) \ldots g(K)]'$, which have zero mean and an identity covariance matrix, according to

$$\mathbf{w} = \mathbf{r} + \mathbf{S}' \mathbf{g} \ , \quad E\{\mathbf{g}\} = 0 \ , \quad E\{\mathbf{g}\ \mathbf{g}'\} = \mathbf{I} \ , \tag{14}$$

where $R = S' S$. For example, $K \times K$ matrix S can be the Cholesky decomposition of positive-definite covariance matrix R. Then, by substitution of equation (14) into equation (13), there follows

$$z(m) = \mathbf{g}' \ C(m)\ \mathbf{g} + v(m)'\ \mathbf{g} + c(m) \quad \text{for } m=1:M \ , \tag{15}$$

where

$$\left.\begin{array}{lll} C(m) = S\ P(m)\ S' & \text{(symmetric} & K \times K) \\ v(m) = S[p(m) + 2\ P(m)\ r] & & (K \times 1) \\ c(m) = q(m) + p(m)'\ r + r'\ P(m)\ r & & (1 \times 1) \end{array}\right\} \text{ for } m=1:M \ . \tag{16}$$

Now, RV $\mathbf{z}$ in equation (15) depends only on the three types of fundamental quantities given in equation (16). This condensation or pre-processing of the input information will prove very useful later when the desired joint statistics (M-dimensional PDF) of RV $\mathbf{z}$ are derived.

If mean vector $r = 0$ and all vectors $p(m) = 0$ for $m=1:M$, then all vectors $v(m) = 0$ for $m=1:M$, and equation (15) reduces to $z(m) = \mathbf{g}'\ C(m)\ \mathbf{g} + q(m)$ for $m=1:M$. This is called the purely quadratic case; its SPA and first-order correction term to the joint PDF of $\mathbf{z}$ is much simpler than for the general QAL problem.

MOMENT GENERATING FUNCTION OF QUADRATIC AND LINEAR FORMS

Let M×1 vector λ have components

$$\lambda = [\lambda(1) \ \ldots \ \lambda(M)]' \ . \tag{17}$$

The joint MGF of M-dimensional RV z in equation (15) is defined as

$$\mu(\lambda) = E\{\exp(\lambda' \ z)\} = E\left\{\exp\left[\sum_{m=1}^{M} \lambda(m) \ z(m)\right]\right\} =$$

$$= E\{\exp[g' \ D(\lambda) \ g + t(\lambda)' \ g + u(\lambda)]\} \ , \tag{18}$$

where

$$D(\lambda) = \sum_{m=1}^{M} \lambda(m) \ C(m) \qquad (\text{symmetric} \quad K\times K) \ , \tag{19}$$

$$t(\lambda) = \sum_{m=1}^{M} \lambda(m) \ v(m) \qquad (K\times 1) \ , \tag{20}$$

and

$$u(\lambda) = \sum_{m=1}^{M} \lambda(m) \ c(m) \qquad (1\times 1) \ . \tag{21}$$

The constant quantities $\{C(m)\}$, $\{v(m)\}$, and $\{c(m)\}$ were defined in equation (16) for m=1:M. It should be observed that matrix functions $D(\lambda)$, $t(\lambda)$, and $u(\lambda)$ are linear in the components $\{\lambda(m)\}$ of M×1 vector λ. The problem of interest now is to evaluate the K-dimensional statistical average in equation (18) in order to determine the M-dimensional MGF $\mu(\lambda)$ of RV z in closed form.

Recall from equation (14) that Kx1 RVs **w** and **g** are related by a linear transformation. Since RV **w** was presumed to have Gaussian statistics, RV **g** must also have Gaussian statistics. In fact, from equation (14), Gaussian RV **g** has a zero mean vector and an identity covariance matrix. The joint PDF of RV **g** is then

$$p(g) = (2\pi)^{-K/2} \exp(- g' g/2) \quad \text{for all } g \, , \tag{22}$$

where $g = [g(1) \ldots g(K)]'$ is a K-dimensional field point. The pertinent K-fold integral representation of equation (18) is

$$\mu(\lambda) = (2\pi)^{-K/2} \int dg \exp[-.5\, g' \, Q(\lambda)\, g + t(\lambda)' \, g + u(\lambda)] =$$

$$= \frac{\exp\left[\frac{1}{2}\, t(\lambda)' \, Q(\lambda)^{-1} \, t(\lambda) + u(\lambda)\right]}{[\det(Q(\lambda))]^{\frac{1}{2}}} \, , \tag{23}$$

where symmetric KxK matrix $Q(\lambda)$ is defined as

$$Q(\lambda) = I - 2\, D(\lambda) \, . \tag{24}$$

The K-fold integral in equation (23) for the joint MGF $\mu(\lambda)$ converges only if all K of the eigenvalues of matrix $Q(\lambda)$ are positive. Equivalently, all K of the eigenvalues of matrix $D(\lambda)$, defined in equation (19), must be less than 1/2. This eigenvalue restriction establishes a boundary on allowed values of vector λ in the M-dimensional λ plane; in particular, the origin, $\lambda = 0$, is always an allowed point, that is, a point at which the joint MGF (23) exists.

Although equation (23) is a closed-form expression for the joint MGF of RV $\mathbf{z}$ in equation (15), it contains numerous branch points and overlapping essential singularities in the complex λ plane, which make it impossible to obtain the corresponding joint M-dimensional PDF analytically. Furthermore, for M large, it is not possible to perform a numerical M-dimensional FFT. Thus, it is necessary to resort to the SPA for the desired PDF in M dimensions.

The corresponding joint cumulant generating function (CGF) to joint MGF (23) is

$$\chi(\lambda) = \log \mu(\lambda) =$$

$$= -\frac{1}{2} \log \det(Q(\lambda)) + \frac{1}{2} t(\lambda)' \, Q(\lambda)^{-1} \, t(\lambda) + u(\lambda) \ . \tag{25}$$

In order to utilize the SPA and its first-order correction term, partial derivatives of joint CGF $\chi(\lambda)$, with respect to its M components $\{\lambda(m)\}$ up through the fourth order, must be determined. One important feature of these derivations is that K×K symmetric matrix

$$Q(\lambda) = I - 2 \, D(\lambda) = I - 2 \sum_{m=1}^{M} \lambda(m) \, C(m) \tag{26}$$

is linear in components $\{\lambda(m)\}$. Here, equations (24) and (19) were used.

M-DIMENSIONAL SADDLEPOINT APPROXIMATION

M-DIMENSIONAL SADDLEPOINT IN λ DOMAIN

Suppose that the joint PDF of M×1 RV z defined in equation (15) is desired to be evaluated at a general M×1 field point $z = [z(1) \ldots z(M)]'$. This PDF value is given in terms of the joint MGF $\mu(\lambda)$ by the M-dimensional integral

$$p(z) = \frac{1}{(i2\pi)^M} \int_{C_1} \cdots \int_{C_M} d\lambda(1) \quad \ldots \; d\lambda(M) \quad \exp[-\, \lambda' \; z] \; \mu(\lambda) =$$

$$= \frac{1}{(i2\pi)^M} \int_{C_1} \cdots \int_{C_M} d\lambda(1) \quad \ldots \; d\lambda(M) \quad \exp[\chi(\lambda) - \lambda' \; z] \; , \quad (27)$$

where contour C_m in the complex $\lambda(m)$ plane goes from $-i\infty$ to $+i\infty$ and stays within the analytic boundary of the joint MGF $\mu(\lambda)$. The SPA consists of locating these contours so that they pass through the M-dimensional SP of the integrand of equation (27), and then approximating the integrand values on the contours by a Gaussian M-dimensional mountain in the neighborhood of the peak at the SP. Finally, this Gaussian approximation is extended to all values on the contours of integration, for which the modified M-dimensional integral is capable of evaluation in closed form.

In order to determine the SP in the λ plane, it is necessary to find the location of the minimum of the real quantity

$$\chi(\lambda) - \lambda' \; z = \chi(\lambda) - \sum_{m=1}^{M} \lambda(m) \; z(m) \qquad (28)$$

in equation (27) for a real λ vector. Alternatively, this SP location is found by solving the M simultaneous nonlinear real equations

$$\frac{\partial \chi(\lambda)}{\partial \lambda(m)} = z(m) \quad \text{for } m=1:M \; ; \quad \{\lambda(m)\} \text{ real} . \tag{29}$$

The real solution $\hat{\lambda} = \hat{\lambda}(z) = [\hat{\lambda}(1) \; ... \; \hat{\lambda}(M)]'$ is a function of the particular M-dimensional field point z of interest. If this field point is changed, the M nonlinear equations (29) must be re-solved for the new SP.

M-DIMENSIONAL SADDLEPOINT APPROXIMATION TO PDF OF z

When the M-fold integration procedure above is carried out, the resulting SPA to the joint PDF is (reference 6)

$$p(z) \cong \frac{\exp[\chi(\hat{\lambda}) - \hat{\lambda}' \, z]}{(2\pi)^{M/2} \; [\det(H(\hat{\lambda}))]^{\frac{1}{2}}} \equiv p_0(z) \; , \tag{30}$$

where $H(\lambda)$ is the M×M symmetric Hessian matrix of second-order partial derivatives of the joint CGF:

$$H(\lambda) = \left[\frac{\partial^2 \chi(\lambda)}{\partial \lambda(m) \;\; \partial \lambda(\underline{m})} \right] \; , \; m,\underline{m}=1:M \; . \tag{31}$$

The function $p_0(z)$ is denoted as SPA0, meaning the zeroth-order SPA to the joint PDF p(z); this nomenclature distinguishes it from some further approximations to the joint PDF p(z) that will employ a first-order correction term.

It is useful to define a Gradient vector as

$$G(\lambda) = \left[\frac{\partial \chi(\lambda)}{\partial \lambda(1)} \cdots \frac{\partial \chi(\lambda)}{\partial \lambda(M)}\right]' \; ; \qquad (32)$$

then, SP equation (29) can be succinctly expressed as $G(\hat{\lambda}) = z$. If the solution for the M-dimensional SP location $\hat{\lambda} = \hat{\lambda}(z)$ is obtained by using the Newton-Raphson search procedure, then both the Mx1 Gradient vector $G(\lambda)$ and the M×M Hessian matrix $H(\lambda)$ will be required during the complete search procedure. This necessitates the evaluation of first- and second-order partial derivatives of the joint CGF, as indicated in equations (31) and (32).

FIRST-ORDER CORRECTION TERM TO THE SP APPROXIMATION

For integers j,k,l,m=1:M, define the following quantities, which are evaluated at the SP $\hat{\lambda} = \hat{\lambda}(z)$, once it has been determined for a given field point z:

$$\chi_m \equiv \left.\frac{\partial \chi(\lambda)}{\partial \lambda(m)}\right|_{\hat{\lambda}} , \quad \chi_{lm} \equiv \left.\frac{\partial^2 \chi(\lambda)}{\partial \lambda(l)\; \partial \lambda(m)}\right|_{\hat{\lambda}} , \qquad (33)$$

$$\chi_{klm} \equiv \left.\frac{\partial^3 \chi(\lambda)}{\partial \lambda(k)\; \partial \lambda(l)\; \partial \lambda(m)}\right|_{\hat{\lambda}} , \quad \chi_{jklm} \equiv \left.\frac{\partial^4 \chi(\lambda)}{\partial \lambda(j)\; \partial \lambda(k)\; \partial \lambda(l)\; \partial \lambda(m)}\right|_{\hat{\lambda}} .$$

The latter two quantities do not need to be evaluated during the search for the SP, but only need to be evaluated after the search has been completed. Also, define the two symmetric M×M matrices

$$H = [X_{lm}] , \qquad T = H^{-1} = [T_{lm}] . \tag{34}$$

Finally, define the three constants

$$c_4 = \frac{1}{8} \sum_{jklm} X_{jklm} \, T_{jk} \, T_{lm} ,$$

$$c_{3a} = - \frac{1}{8} \sum_{klm} \sum_{\underline{klm}} X_{klm} \, X_{\underline{klm}} \, T_{kl} \, T_{m\underline{k}} \, T_{\underline{lm}} ,$$

and

$$c_{3b} = - \frac{1}{12} \sum_{klm} \sum_{\underline{klm}} X_{klm} \, X_{\underline{klm}} \, T_{k\underline{k}} \, T_{l\underline{l}} \, T_{m\underline{m}} , \tag{35}$$

where the sums all run from 1 to M.

The first-order correction to the SPA0 given in equation (30) can now be expressed in the form (reference 6, page 180)

$$p_1(z) \equiv p_0(z) \, [1 + c_t] , \tag{36}$$

where the total first-order correction term is defined as

$$c_t = c_4 + c_{3a} + c_{3b} . \tag{37}$$

The joint PDF approximation $p_1(z)$ in equation (36) is denoted as SPA1, meaning the first-order SPA. Its computation requires determination of the SP location, as well as third- and fourth-order information about the partial derivatives of the joint CGF $\chi(\lambda)$ at the SP $\hat{\lambda} = \hat{\lambda}(z)$.

MODIFIED SADDLEPOINT APPROXIMATIONS

Consider the following three modified SPAs:

$$p_1(z) \equiv p_0(z)\,[1 + c_t] = p_0(z)\,[1 + c_t + 0\,c_{t2} + 0\,c_{t3} + \cdots]\,,$$

$$p_e(z) \equiv p_0(z)\,\exp(c_t) = p_0(z)\,[1 + c_t + \frac{1}{2}\,c_t^2 + \frac{1}{6}\,c_t^3 + \cdots]\,,$$

and

$$p_a(z) \equiv p_0(z)\,\frac{1 + c_t/2}{1 - c_t/2} = p_0(z)\,[1 + c_t + \frac{1}{2}\,c_t^2 + \frac{1}{4}\,c_t^3 + \cdots]\,. \tag{38}$$

Approximation $p_1(z)$ defined in equation (36) tacitly employs a zero coefficient for the second-order correction term (SOCT) c_{t2}. This coefficient is most certainly incorrect because there definitely is a nonzero SOCT; however, this SOCT is not known. Furthermore, the SOCT c_{t2} would require knowledge of the fifth- and sixth-order partial derivatives of the joint CGF $\chi(\lambda)$ at the SP. Since there are M^6 sixth-order partial derivatives, a problem arises in execution time and storage when attempting to calculate these latter quantities.

To circumvent the lack of knowledge and computational limitations, approximation $p_e(z)$ in equation (38) has been suggested (reference 6, page 180) because it injects the SOCT $c_t^{\,2}/2$ instead of zero. Again, this term is most certainly incorrect; however, it may give a better approximation to the true joint PDF $p(z)$ than either of the approximations $p_0(z)$ or $p_1(z)$.

The third approximation, $p_a(z)$, in equation (38) uses, instead, a rational function in c_t, which has the same power series expansion as $\exp(c_t)$ through second order. As c_t increases, approximation $p_a(z)$ becomes greater than $p_e(z)$ and would tend to infinity if c_t approaches 2; however, by this time, c_t could no longer be considered a correction term, but in fact, a dominant contributor.

The author has conducted some numerical comparisons of the three approximations in equation (38) for some cases where the exact joint PDF p(z) can be determined. These results indicate that both $p_e(z)$ and $p_a(z)$ generally yield worthwhile improvements relative to $p_1(z)$, which, in turn, yields worthwhile improvements compared to $p_0(z)$. The choice between $p_e(z)$ and $p_a(z)$ varies from example to example.

EVALUATION OF PARTIAL DERIVATIVES OF JOINT CGF $\chi(\lambda)$

Evaluation of the various SPAs depends heavily upon the ability to obtain the partial derivatives of the joint CGF $\chi(\lambda)$ of RV z; see equations (30) through (33). This joint CGF is repeated from equations (25), (26), (20), and (21):

$$\chi(\lambda) = -\frac{1}{2} \log \det(Q(\lambda)) + \frac{1}{2} t(\lambda)' \, Q(\lambda)^{-1} \, t(\lambda) + u(\lambda) \; , \tag{39}$$

where

$$Q(\lambda) = I - 2 \, D(\lambda) = I - 2 \sum_{m=1}^{M} \lambda(m) \, C(m) \qquad (K\times K) \; , \tag{40}$$

$$t(\lambda) = \sum_{m=1}^{M} \lambda(m) \, v(m) \qquad (K\times 1) \; , \tag{41}$$

$$u(\lambda) = \sum_{m=1}^{M} \lambda(m) \, c(m) \qquad (1\times 1) \; . \tag{42}$$

A POSSIBLE APPROACH TO THE PARTIAL DERIVATIVES

From equation (25), joint CGF $\chi(\lambda) = \log \mu(\lambda)$; therefore,

$$\frac{\partial \chi(\lambda)}{\partial \lambda(m)} = \frac{1}{\mu(\lambda)} \frac{\partial \mu(\lambda)}{\partial \lambda(m)} \; . \tag{43}$$

Also, from equation (18),

$$\mu(\lambda) = E\{\exp[\lambda(1) \, z(1) + \cdots + \lambda(M) \, z(M)]\} \; . \tag{44}$$

There follows immediately

$$\frac{\partial \mu(\lambda)}{\partial \lambda(m)} = E\{z(m) \, \exp[\lambda(1) \, z(1) + \cdots \lambda(M) \, z(M)]\} \; , \tag{45}$$

and

$$\frac{\partial^2 \mu(\lambda)}{\partial\lambda(1)\ \partial\lambda(m)} = E\{z(1)\ z(m)\ \exp[\lambda(1)\, z(1) + \cdots \lambda(M)\ z(M)]\}\ . \quad (46)$$

Recall from defining equation (13) and condensed version (15) that RV z is quadratic in Gaussian RVs; therefore, the arguments of the exponentials in equations (45) and (46) are quadratic in Gaussian RVs. Similarly, the leading multiplying factor z(m) in equation (45) is quadratic in Gaussian RVs. Therefore, the multiple integral representing expectation (45) can certainly be evaluated in closed form. In a similar fashion, the leading factor z(1) z(m) in equation (46) is quartic in Gaussian RVs, meaning that it too can be evaluated in closed form. The same conclusion holds for all the higher order partial derivatives of joint CGF $\chi(\lambda)$, although the integral evaluations will be considerably more tedious to carry out.

The significance of this observation is that all of the required information for obtaining the various SPAs is obtainable, somehow, in closed form. The best route for getting this information may not be by means of expectations (45) and (46), but at least it is now known that the desired information is obtainable. (An example of this route to the partial derivatives of the joint CGF is given in appendix B; some interesting results on partial derivatives of eigenvalues are also provided.) However, the alternative technique described below is much more efficient and more readily supplies the required higher order joint CGF partial derivatives needed for the various SPAs.

FIRST-ORDER PARTIAL DERIVATIVES OF CGF1

The joint CGF $\chi(\lambda)$ is given in equation (39). It is composed of three additive parts, to be labeled $\chi_1(\lambda)$, $\chi_2(\lambda)$, and $\chi_3(\lambda)$. The partial derivative of the third part, $\chi_3(\lambda) = u(\lambda)$, with respect to $\lambda(m)$, is simply $c(m)$, as seen from equation (42). The higher order derivatives of $\chi_3(\lambda)$ are all zero because $\{c(m)\}$ are constants (see equation (16)).

The immediate interest in this subsection is in the first part, CGF1, of the complete joint CGF $\chi(\lambda)$, namely,

$$\chi_1(\lambda) = -\frac{1}{2} \log \det Q(\lambda) \ , \tag{47}$$

where the symmetric K×K matrix $Q(\lambda)$ is given by equation (40). In order to streamline the following derivations, a number of useful matrix properties were collected in appendix C and will be referred to, as necessary.

If symmetric K×K matrix $D(\lambda)$ is expanded in its eigen-decomposition, the result is

$$D(\lambda) = \sum_{m=1}^{M} \lambda(m)\ C(m) = V(\lambda)\ E(\lambda)\ V(\lambda)' \ , \quad V(\lambda)\ V(\lambda)' = I \ , \tag{48}$$

where K×K matrix $V(\lambda)$ is the set of eigenvectors, and diagonal K×K matrix $E(\lambda)$ is the set of eigenvalues $\{e_k(\lambda)\}$, k=1:K. There follows

$$Q(\lambda) = I - 2\ D(\lambda) = V(\lambda)\ [I - 2\ E(\lambda)]\ V(\lambda)' \ , \tag{49}$$

and

$$\det Q(\lambda) = \det[I - 2\ E(\lambda)] = \prod_{k=1}^{K} [1 - 2\ e_k(\lambda)] \tag{50}$$

because matrix V has a unity determinant. Equation (47) yields

$$\chi_1(\lambda) = -\frac{1}{2} \log \det Q(\lambda) = -\frac{1}{2} \sum_{k=1}^{K} \log[1 - 2\ e_k(\lambda)] =$$

$$= \frac{1}{2} \sum_{k=1}^{K} \sum_{p=1}^{\infty} \frac{2^p}{p}\ e_k(\lambda)^p = \frac{1}{2} \sum_{p=1}^{\infty} \frac{2^p}{p} \sum_{k=1}^{K} e_k(\lambda)^p = \frac{1}{2} \sum_{p=1}^{\infty} \frac{2^p}{p} \operatorname{tr}\{D(\lambda)^p\}$$

$$= \frac{1}{2} \operatorname{tr}\left\{\sum_{p=1}^{\infty} \frac{2^p}{p}\ D(\lambda)^p\right\} = \frac{1}{2} \operatorname{tr}\{-\log[I - 2\ D(\lambda)]\}\ , \tag{51}$$

where equations (C-6) and (C-12) were used. Now, by using equations (C-18), (48), and (49), there follows the desired first-order partial derivative

$$\frac{\partial \chi_1(\lambda)}{\partial \lambda(m)} = \frac{1}{2} \operatorname{tr}\left\{[I - 2\ D(\lambda)]^{-1}\ 2\ \frac{\partial D(\lambda)}{\partial \lambda(m)}\right\} = \operatorname{tr}\{Q(\lambda)^{-1}\ C(m)\} \tag{52}$$

for m=1:M. The symmetric K×K matrices {C(m)} are given in equation (16).

SECOND-ORDER PARTIAL DERIVATIVES OF CGF1

For convenience of notation, let symmetric K×K matrix

$$q(\lambda) = Q(\lambda)^{-1}\ , \quad Q(\lambda) = I - 2\ D(\lambda) = \sum_{m=1}^{M} \lambda(m)\ C(m)\ . \tag{53}$$

Then, by using equations (C-15), (48), and (49), there follows

$$\frac{\partial q(\lambda)}{\partial \lambda(m)} = - q(\lambda) \frac{\partial Q(\lambda)}{\partial \lambda(m)} q(\lambda) = 2\ q(\lambda)\ C(m)\ q(\lambda) \quad \text{for } m=1:M. \tag{54}$$

This enables equation (52) to be written in the compact form

$$\frac{\partial \chi_1(\lambda)}{\partial \lambda(m)} = \mathrm{tr}\{q(\lambda)\ C(m)\} = \mathrm{tr}\{B(\lambda,m)\} \quad \text{for } m=1:M\ , \tag{55}$$

where nonsymmetric K×K matrices $\{B(\lambda,m)\}$, $m=1:M$, are introduced for future use. The desired second-order partial derivative now follows from equations (54) and (55) as

$$\frac{\partial^2 \chi_1(\lambda)}{\partial \lambda(l)\ \partial \lambda(m)} = 2\ \mathrm{tr}\{q(\lambda)\ C(l)\ q(\lambda)\ C(m)\} = 2\ \mathrm{tr}\{B(\lambda,l)\ B(\lambda,m)\}$$

$$\text{for } l,m=1:M\ . \tag{56}$$

THIRD- AND FOURTH-ORDER PARTIAL DERIVATIVES OF CGF1

By using equations (56) and (54), it immediately follows that, for $k,l,m=1:M$,

$$\frac{\partial^3 \chi_1(\lambda)}{\partial \lambda(k)\ \partial \lambda(l)\ \partial \lambda(m)} = 8\ \mathrm{tr}\{B(\lambda,k)\ B(\lambda,l)\ B(\lambda,m)\} = 8\ \mathrm{tr}\{B_k\ B_l\ B_m\}\ , \tag{57}$$

where the shortcut notation $B(\lambda,m) = B_m$ has been introduced. With this notation, the final quantity of interest is

$$\frac{\partial^4 \chi_1(\lambda)}{\partial \lambda(j)\ \partial \lambda(k)\ \partial \lambda(l)\ \partial \lambda(m)} = 16\ \mathrm{tr}\{B_j\ B_k\ B_l\ B_m +$$

$$+ B_j\ B_k\ B_m\ B_l + B_j\ B_l\ B_k\ B_m\} \quad \text{for } j,k,l,m=1:M\ . \tag{58}$$

A summary of the notation is given by

$$B_m = B(\lambda,m) = q(\lambda)\ C(m) = Q(\lambda)^{-1}\ C(m) = [I - 2\ D(\lambda)]^{-1}\ C(m) \qquad (59)$$

for m=1:M, with

$$D(\lambda) = \sum_{m=1}^{M} \lambda(m)\ C(m)\ . \qquad (60)$$

In writing expressions (57) and (58), advantage has been taken of symmetries of some expressions involved in matrices $\{B_m\}$; this allowed equation (57) to be condensed into a single trace. However, the three traces remaining in equation (58) are all different in general, and no further reduction is possible in the number of terms that must be calculated.

FIRST- AND SECOND-ORDER PARTIAL DERIVATIVES OF CGF2

The interest is now centered on the second part, CGF2, of the complete joint CGF $\chi(\lambda)$ in equation (39), namely,

$$\chi_2(\lambda) = \frac{1}{2}\ t(\lambda)'\ q(t)\ t(\lambda)\ , \qquad t(\lambda) = \sum_{m=1}^{M} \lambda(m)\ v(m)\ . \qquad (61)$$

The pertinent partial derivatives required are given by equation (54) and

$$\frac{\partial t(\lambda)}{\partial \lambda(m)} = v(m) \quad \text{for } m=1:M\ . \qquad (62)$$

Application to equation (61) yields the first-order partial derivative of CGF2 in the form

$$\frac{\partial \chi_2(\lambda)}{\partial \lambda(m)} = v(m)' \; q \; t + t' \; B_m \; q \; t \quad \text{for } m=1:M \; , \tag{63}$$

using an obvious shorthand notation. The corresponding second-order partial derivative is obtained by repeated applications of the above rules:

$$\frac{\partial^2 \chi_2(\lambda)}{\partial \lambda(1) \; \partial \lambda(m)} = v(1)' \; q \; v(m) + 2 \; v(1)' \; B_m \; q \; t \; +$$

$$+ \; 2 \; v(m)' \; B_1 \; q \; t + 4 \; t' \; B_1 \; B_m \; q \; t \quad \text{for } 1,m=1:M \; . \tag{64}$$

It should be noted that the original CGF2, namely quadratic form $\chi_2(\lambda)$ in equation (61), began and ended with K×1 vector t; this is called a (t-t) type of term. On the other hand, the first-order partial derivative in equation (63) involved an additional type of quadratic form starting with K×1 vector v(m) and ending with vector t; this is called a (v-t) type of term. Finally, the second-order partial derivative in equation (64) involved still another type of quadratic form, beginning and ending with two v vectors; this is called a (v-v) type of term. At this point, steady state is reached; that is, no more additional <u>types</u> of terms are generated by taking additional higher order partial derivatives of equation (64). However, the numbers of each type of term do increase, and the complexity of each term also increases.

THIRD-ORDER PARTIAL DERIVATIVES OF CGF2

Upon taking the next partial derivative of equation (64) and combining like terms, the result is

$$\frac{\partial^3 \chi_2(\lambda)}{\partial\lambda(k)\ \partial\lambda(l)\ \partial\lambda(m)} =$$

$$\begin{array}{cccccccccc}
2\,v'_k\,q & C_l\,q & v_m + 4 & v'_k\,q & C_l\,q & C_m\,q\,t + 8\,t'\,q & C_k\,q & C_l\,q & C_m\,q\,t + & \\
k & m & l & k & m & l & k & m & l & \\
l & k & m & l & k & m & l & k & m & \\
 & & & l & m & k & & & & \\
 & & & m & k & l & & & & \\
 & & & m & l & k & & & & (65)
\end{array}$$

for k,l,m=1:M, where only the subscripts have been indicated after the first line. There are 3! = 6 terms of type (v-t), but only three terms of types (v-v) and (t-t). The reason for this is that the transposes of half of the (v-v) and (t-t) (scalar) terms can be shown to be equal to those displayed in equation (65); these common values have been combined and the appropriate scale factor adjusted. All the quadratic forms remaining in equation (65) are different in general; no further reduction in the number or types of terms is possible.

FOURTH-ORDER PARTIAL DERIVATIVES OF CGF2

The fourth-order partial derivative of CGF2, namely $\chi_2(\lambda)$, also contains only the (v-v), (v-t), and (t-t) terms. In particular, for j,k,l,m=1:M, the (v-v) terms are

$$4 \; v_j' \; q \; C_k \; q \; C_l \; q \; v_m \tag{66}$$

and 12 permutations of its subscripts. The (v-t) terms are

$$8 \; v_j' \; q \; C_k \; q \; C_l \; q \; C_m \; q \; t \tag{67}$$

and 24 permutations of its subscripts. The (t-t) terms are

$$8 \; t' \; q \; C_j \; q \; C_k \; q \; C_l \; q \; C_m \; q \; t \tag{68}$$

and 12 permutations of its subscripts.

There are 4! = 24 terms of type (v-t), but only 12 terms of types (v-v) and (t-t). The reason is identical to that cited under equation (65), namely, the equality of some transposes of scalar quantities involving (v-v) or (t-t) terms. The twelve quadratic forms remaining in equations (66) through (68) are different in general; no further reduction in the number or types of terms is possible.

It should be noted in equations (66) through (68) that a large number of matrix multiplications are involved, especially in equation (68), where 11 terms are involved. However, by starting at one end of equation (68), for example, all the successive multiplications involve a vector with a matrix, which is considerably faster than for full $K \times K$ matrices. Alternatively, the $\{B_m\}$ matrices in equation (59) can be computed once and stored for repeated use in the operations above. The danger with this latter approach is the possibility of very large storage requirements, especially for large M and/or K.

The totality of partial derivatives required to compute the first-order correction term c_t to the SPA in equations (33) through (38) is given in equations (55) through (58) and equations (63) through (68). These results have been combined in a MATLAB program listed in appendix D and entitled quadlinspa, denoting the SPA for the M general quadratic and linear forms of equation (13). In the special case of purely quadratic forms (see bottom of page 8), an alternative MATLAB program, listed in appendix E and entitled quadspa, has been written. Both programs compute the three SPAs in equation (38) as well as the standard SPA0 in equation (30).

SUMMARY

The saddlepoint approximation to the joint M-dimensional probability density function for M arbitrary quadratic and linear forms in K Gaussian random variables with arbitrary means and covariance matrix has been derived. Also, the first-order correction term to the standard saddlepoint approximation in M dimensions has been determined and used to form several different possible approximations to the desired joint probability density function.

The determination of the M-dimensional saddlepoint location, and the standard saddlepoint approximation itself, require evaluation of first- and second-order partial derivatives of the joint cumulant generating function at arbitrary points in M-dimensional space; these quantities have been derived in closed form. Also, the third- and fourth-order partial derivatives of the joint cumulant generating function have been derived for use in calculating the first-order correction term to the saddlepoint approximation in M dimensions. All these results have been combined in two MATLAB programs; namely, quadlinspa, which handles the quadratic and linear case, and quadspa, which handles the purely quadratic case.

Sometimes, interest is centered on the square roots of the quadratic and linear random variables $\{z(m)\}$ when all of these quantities are nonnegative. For example, in some signal

processing applications, the square roots represent envelope or amplitude quantities of interest, while the $\{z(m)\}$ are power quantities. Letting $\mathbf{u}(m) = \mathbf{z}(m)^{1/2}$ for m=1:M, the joint probability density function p_2 of the random variables $\{\mathbf{u}(m)\}$ at M-dimensional field point $u = [u_1, \ldots, u_M]'$ is given by

$$p_2(u_1, \cdots, u_M) = p\left(u_1^2, \cdots, u_M^2\right) 2^M u_1 \cdots u_M \tag{69}$$

for $u_m > 0$ for m=1:M. Thus, if joint probability density function p_2 is to be determined at arguments $u_1, \ldots, u_M$, the joint probability density function p of random vector $\mathbf{z}$ must be evaluated at arguments $u_1^2, \ldots, u_M^2$; this serves as the field point z, namely, $z = [u_1^2, \ldots, u_M^2]'$, for the procedures detailed above in this report. More generally, this procedure can be extended to nonlinear transformations $\mathbf{u}(m) = f_m(\mathbf{z})$ for m=1:M, provided that the right-hand sides $\{f_m(\mathbf{z})\}$ do not generate imaginary numbers for some values of random vector $\mathbf{z}$.

APPENDIX A — A PROPERTY OF QUADRATIC FORMS

Let v be a K×1 vector and let C be a K×K matrix, not necessarily symmetric. The symmetric and anti(skew)-symmetric matrices of C are defined as

$$C_s = \frac{1}{2}(C + C') , \quad C_a = \frac{1}{2}(C - C') . \qquad \text{(A-1)}$$

It immediately follows that $C = C_s + C_a$ and $C'_a = -C_a$.

Now, consider the quadratic form

$$f = v' \, C \, v = v' \, (C_s + C_a) \, v = v' \, C_s \, v \quad \text{for any } v . \qquad \text{(A-2)}$$

The term involving C_a is zero, as can be seen by taking its transpose and using the property $C'_a = -C_a$; thus, only the symmetric part of matrix C is active in quadratic form f. Therefore, when a quadratic form such as $v' \, C \, v$ is encountered, the matrix C may be presumed symmetric without loss of generality.

APPENDIX B — DIRECT EVALUATION OF EXPECTATIONS

FIRST-ORDER PARTIAL DERIVATIVES OF CGF

The first-order (FO) partial derivative (PD) of joint MGF $\mu(\lambda)$ is given by the expectation in equation (45). Also, RV $\mathbf{z}(m)$ is given in equations (14) and (15) as

$$\mathbf{z}(m) = \mathbf{g}' \, C(m) \, \mathbf{g} \quad \text{for } m=1:M \;, \tag{B-1}$$

where consideration is limited here to the purely quadratic case; see bottom of page 8. Combining these results leads to FO PD

$$\frac{\partial\mu(\lambda)}{\partial\lambda(m)} = E\{\mathbf{g}' \, C(m) \, \mathbf{g} \, \exp[\mathbf{g}' \, D(\lambda) \, \mathbf{g}]\} \quad \text{for } m=1:M \;, \tag{B-2}$$

where symmetric matrix $D(\lambda)$ is given in equation (19) as

$$D(\lambda) = \sum_{m=1}^{M} \lambda(m) \, C(m) \;. \tag{B-3}$$

Perform the same eigen-decomposition on matrix $D(\lambda)$ as in equation (48); namely, $D(\lambda) = V(\lambda) \, E(\lambda) \, V(\lambda)' = V \, E(\lambda) \, V'$, and define the linearly transformed Kx1 zero-mean Gaussian RV

$$\mathbf{y} = V' \, \mathbf{g} \quad \text{with} \quad E\{\mathbf{y}\} = 0 \;, \quad E\{\mathbf{y} \, \mathbf{y}'\} = E\{V' \, \mathbf{g} \, \mathbf{g}' \, V\} = I \;. \tag{B-4}$$

Then, using $\mathbf{g} = V \, \mathbf{y}$, the FO PD in equation (B-2) becomes

$$\frac{\partial\mu(\lambda)}{\partial\lambda(m)} = E\{\mathbf{y}' \, V' \, C(m) \, V \, \mathbf{y} \, \exp(\mathbf{y}' \, V' \, D(\lambda) \, V \, \mathbf{y}\} =$$

$$= E\{\mathbf{y}' \, F(\lambda,m) \, \mathbf{y} \, \exp(\mathbf{y}' \, E(\lambda) \, \mathbf{y})\} \quad \text{for } m=1:M \;, \tag{B-5}$$

where $E(\lambda) = \text{diag}\{e_k(\lambda)\}$, and symmetric KxK matrix

$$F(\lambda,m) \equiv V' \, C(m) \, V = V(\lambda)' \, C(m) \, V(\lambda) \quad \text{for } m=1:M \, . \tag{B-6}$$

Define the elements of this K×K F matrix as

$$F(\lambda,m) = [f(\lambda,m;k,\underline{k})] \quad \text{for } k,\underline{k}=1:K \, ; \quad m=1:M \, , \tag{B-7}$$

and let the components of RV $\mathbf{y}$ in equation (B-4) be denoted as

$$\mathbf{y} = [\mathbf{y}(1) \cdots \mathbf{y}(K)]' \, . \tag{B-8}$$

Then, the FO PD in equation (B-5) becomes, for m=1:M,

$$\frac{\partial\mu(\lambda)}{\partial\lambda(m)} = E\left\{\sum_{k,\underline{k}=1}^{K} f(\lambda,m;k,\underline{k}) \; \mathbf{y}(k) \; \mathbf{y}(\underline{k}) \; \exp\left(\sum_{p=1}^{K} e_p(\lambda) \; \mathbf{y}(p)^2\right)\right\} =$$

$$= \sum_{k,\underline{k}=1}^{K} f(\lambda,m;k,\underline{k}) \; E\left\{\mathbf{y}(k) \; \mathbf{y}(\underline{k}) \; \exp\left(\sum_{p=1}^{K} e_p(\lambda) \; \mathbf{y}(p)^2\right)\right\} \, . \tag{B-9}$$

If $k \neq \underline{k}$, the expectation in equation (B-9) is zero. Therefore,

$$\frac{\partial\mu(\lambda)}{\partial\lambda(m)} = \sum_{k=1}^{K} f(\lambda,m;k,k) \; E\left\{\mathbf{y}(k)^2 \; \exp\left(\sum_{p=1}^{K} e_p(\lambda) \; \mathbf{y}(p)^2\right)\right\} \, . \tag{B-10}$$

The k-th average in equation (B-10) is, with the help of the statistics of Gaussian RV $\mathbf{y}$ in equation (B-4),

$$(1 - 2\, e_k)^{-3/2} \prod_{\substack{p=1 \\ p\neq k}}^{K} (1 - 2\, e_p)^{-\frac{1}{2}} = (1 - 2\, e_k)^{-1} \prod_{p=1}^{K} (1 - 2\, e_p)^{-\frac{1}{2}} =$$

$$= (1 - 2\, e_k)^{-1} \, \mu(\lambda) \quad \text{for } k=1:K \, . \tag{B-11}$$

This last result for joint MGF $\mu(\lambda)$ follows from equations (23) and (50) in the purely quadratic case. The use of equation

(B-11) in equation (B-10) yields

$$\frac{\partial \mu(\lambda)}{\partial \lambda(m)} = \mu(\lambda) \sum_{k=1}^{K} f(\lambda,m;k,k) \, [1 - 2\, e_k(\lambda)]^{-1} \quad \text{for } m=1:M \; . \tag{B-12}$$

But, since $\chi(\lambda) = \log \mu(\lambda)$, there immediately follows

$$\frac{\partial \chi(\lambda)}{\partial \lambda(m)} = \sum_{k=1}^{K} \frac{f(\lambda,m;k,k)}{1 - 2\, e_k(\lambda)} \quad \text{for } m=1:M \; . \tag{B-13}$$

The elements $\{f(\lambda,m,k,\underline{k})\}$ of matrix $F(\lambda,m)$ are given in equations (B-6) and (B-7). If the K×K eigenvector matrix $V(\lambda)$ in equations (B-4) through (B-6) is expressed in terms of its K×1 column vectors $\{V_k(\lambda)\}$, k=1:K, according to $V(\lambda) = [V_1(\lambda) \cdots V_K(\lambda)]$, then equations (B-6) and (B-7) yield

$$f(\lambda,m,k,k) = V_k(\lambda)' \, C(m) \, V_k(\lambda) \quad \text{for } m=1:M \; , \; k=1:K \; , \tag{B-14}$$

which avoids the calculation of the entire K×K $F(\lambda,m)$ matrix for each m.

The result in equation (B-13) can be manipulated into a familiar form:

$$\frac{\partial \chi(\lambda)}{\partial \lambda(m)} = \operatorname{tr}\{[I - 2\, E(\lambda)]^{-1} F(\lambda,m)\} = \operatorname{tr}\{[I - 2\, E(\lambda)]^{-1} V' \, C(m) \, V\}$$

$$= \operatorname{tr}\{V \, [I - 2\, E(\lambda)]^{-1} V' \, C(m)\} = \operatorname{tr}\{Q(\lambda)^{-1} C(m)\} \; , \tag{B-15}$$

where equations (B-6), (B-7), and (49) were used. This latter result in equation (B-15) agrees with equation (52) because the second part of the joint CGF, $\chi_2(\lambda)$, is zero in this purely quadratic case.

SECOND-ORDER PARTIAL DERIVATIVES OF CGF

This presentation will be somewhat abbreviated, since the details are similar to those above. Using shorthand notation

$$\mu_m(\lambda) = \frac{\partial \mu(\lambda)}{\partial \lambda(m)} , \quad \chi_m(\lambda) = \frac{\partial \chi(\lambda)}{\partial \lambda(m)} = \frac{\mu_m(\lambda)}{\mu(\lambda)} . \tag{B-16}$$

There follows, for the second-order PDs of the joint CGF,

$$\chi_{m\underline{m}}(\lambda) = \frac{\mu_{m\underline{m}}(\lambda)}{\mu(\lambda)} - \chi_m(\lambda)\ \chi_{\underline{m}}(\lambda) . \tag{B-17}$$

From equation (46), the pertinent quantity is

$$\mu_{m\underline{m}}(\lambda) = E\left\{z(m)\ z(\underline{m})\ \exp\left(\sum_{p=1}^{M} \lambda(p)\ z(p)\right)\right\} =$$

$$= \sum_{k\underline{k}} \sum_{l\underline{l}} f(m;k,\underline{k})\ f(\underline{m};l,\underline{l})\ E\left\{y_k\ y_{\underline{k}}\ y_l\ y_{\underline{l}}\ \exp\left(\sum_{p=1}^{K} e_p\ y_p^2\right)\right\} . \tag{B-18}$$

Only two cases yield nonzero averages in equation (B-18). In the first case, $k = \underline{k} = l = \underline{l}$, the statistical average becomes

$$E\left\{y_k^4\ \exp\left(\sum_{p=1}^{K} e_p\ y_p^2\right)\right\} = E\left(y_k^4\ \exp\left[e_k\ y_k^2\right]\right) \prod_{\substack{p=1\\p\neq k}}^{K} E\left(\exp\left[e_p\ y_p^2\right]\right) =$$

$$= \frac{3}{[1 - 2\ e_k]^{5/2}} \prod_{\substack{p=1\\p\neq k}}^{K} [1 - 2\ e_k]^{-½} = \frac{3\ \mu(\lambda)}{[1 - 2\ e_k]^2} \quad \text{for } k=1:K. \tag{B-19}$$

For convergence of these integrals, it is necessary that all the eigenvalues $e_k = e_k(\lambda) < ½$ for $k=1:K$. This case contributes the following term to $\mu_{m\underline{m}}(\lambda)$:

$$3\ \mu(\lambda) \sum_{k=1}^{K} \frac{f(m;k,k)\ f(\underline{m};k,k)}{[1 - 2\ e_k]^2} \quad \text{for } m,\underline{m}=1:M\ . \tag{B-20}$$

The second case consists of three subcases:

$$\begin{aligned} k &= \underline{k} \neq l = \underline{l}\ , \\ k &= l \neq \underline{k} = \underline{l}\ , \\ k &= \underline{l} \neq \underline{k} = l\ . \end{aligned} \tag{B-21}$$

The pertinent average for the first subcase is

$$E\left\{y_k^2\ y_l^2\ \exp\left(\sum_{p=1}^{K} e_p\ y_p^2\right)\right\} = \frac{\mu(\lambda)}{[1 - 2\ e_k][1 - 2\ e_l]}\ . \tag{B-22}$$

This first subcase contributes the term

$$\mu(\lambda) \sum_{k \neq l}^{K} \frac{f(m;k,k)\ f(\underline{m};l,l)}{[1 - 2\ e_k][1 - 2\ e_l]} = \tag{B-23}$$

$$= \mu(\lambda)\ \left\{\left(\sum_{k=1}^{K} \frac{f(m;k,k)}{1 - 2\ e_k}\right) \left(\sum_{k=1}^{K} \frac{f(\underline{m};k,k)}{1 - 2\ e_k}\right) - \sum_{k=1}^{K} \frac{f(m;k,k)\ f(\underline{m};k,k)}{[1 - 2\ e_k]^2}\right\}\ .$$

The other two subcases each contribute the following term to $\mu_{m\underline{m}}(\lambda)$:

$$\mu(\lambda) \sum_{k \neq \underline{k}}^{K} \frac{f(m;k,\underline{k})\ f(\underline{m};k,\underline{k})}{[1 - 2\ e_k][1 - 2\ e_{\underline{k}}]} = \tag{B-24}$$

$$= \mu(\lambda)\ \left\{\sum_{k,\underline{k}=1}^{K} \frac{f(m;k,\underline{k})\ f(\underline{m};k,\underline{k})}{[1 - 2\ e_k][1 - 2\ e_{\underline{k}}]} - \sum_{k=1}^{K} \frac{f(m;k,k)\ f(\underline{m};k,k)}{[1 - 2\ e_k]^2}\right\}\ .$$

When all the terms are combined according to equation (B-17), a number of cancellations occur, yielding, for $m,\underline{m}=1:M$,

$$\chi_{m\underline{m}}(\lambda) = 2 \sum_{k,\underline{k}=1}^{K} \frac{f(\lambda,m;k,\underline{k})\ f(\lambda,\underline{m};k,\underline{k})}{[1 - 2\ e_k(\lambda)][1 - 2\ e_{\underline{k}}(\lambda)]}\ , \tag{B-25}$$

where the λ dependence has been reintroduced, and equation (B-13) was used.

Equation (B-25) may be manipulated as follows:

$$\chi_{m\underline{m}}(\lambda) = 2\ \mathrm{tr}\{(I - 2\ E)^{-1}\ F(\lambda,m)\ (I - 2\ E)^{-1}\ F(\lambda,\underline{m})\} =$$

$$= 2\ \mathrm{tr}\{(I- 2\ E)^{-1}\ V'\ C(m)\ V\ (I - 2\ E)^{-1}\ V'\ C(\underline{m})\ V\}\ , \tag{B-26}$$

upon use of equations (B-6) and (B-7). Then, upon movement of the trailing matrix V to the front of the trace, there follows

$$\chi_{m\underline{m}}(\lambda) = 2\ \mathrm{tr}\{Q(\lambda)^{-1}\ C(m)\ Q(\lambda)^{-1}\ C(\underline{m})\} \quad \text{for } m,\underline{m}=1:M\ , \tag{B-27}$$

where equation (49) was used. This result is identical to equation (56) in this purely quadratic case.

RELATED EIGENVALUE PROPERTIES

Suppose matrix D is K×K and symmetric. Its eigen-decomposition is $D = V\ E\ V'$ or $D\ V_k = e_k\ V_k$ for k=1:K. Then, $V_k'\ D = e_k\ V_k'$ and $V_k'\ V_l = \delta_{kl}$. Equivalently, $e_k = V_k'\ D\ V_k$. Now, suppose that matrix D is a function of scalar x. Then,

$$\frac{d}{dx}e_k = \frac{d}{dx}V_k' \; D \; V_k + V_k' \frac{d}{dx}D \; V_k + V_k' \; D \frac{d}{dx}V_k =$$

$$= e_k\left(\frac{d}{dx}V_k' \; V_k + V_k' \frac{d}{dx}V_k\right) + V_k' \frac{d}{dx}D \; V_k =$$

$$= e_k \frac{d}{dx}\left(V_k' \; V_k\right) + V_k' \frac{d}{dx}D \; V_k \; . \qquad \text{(B-28)}$$

But, since $V_k' \; V_k = 1$ for all x, it follows that

$$\frac{d}{dx}e_k(x) = V_k(x)' \frac{d}{dx}D(x) \; V_k(x) \quad \text{for } k=1:K \; . \qquad \text{(B-29)}$$

Relation (B-29) is true for any matrix D(x). Now, let $\lambda = [\lambda(1) \cdots \lambda(M)]'$ and

$$D = D(\lambda) = \sum_{m=1}^{M} \lambda(m) \; C(m) \; , \quad K\times K \; C(m) \text{ constant} \; , \qquad \text{(B-30)}$$

and interpret x as $\lambda(m)$. Then, $D(\lambda) \; V_k(\lambda) = e_k(\lambda) \; V_k(\lambda)$ and

$$\frac{\partial e_k(\lambda)}{\partial \lambda(m)} = V_k(\lambda)' \frac{\partial D(\lambda)}{\partial \lambda(m)} V_k(\lambda) = V_k(\lambda)' \; C(m) \; V_k(\lambda) \qquad \text{(B-31)}$$

for k=1:K, m=1:M. This is a useful relation for the PD of an eigenvalue. The quantity on the right-hand side of equation (B-31) is identical to the quantity in equation (B-14).

Upon multiplication of equation (B-31) by $\lambda(m)$, summation over m, and use of equation (B-30), there follows

$$\sum_{m=1}^{M} \lambda(m) \frac{\partial e_k(\lambda)}{\partial \lambda(m)} = V_k(\lambda)' \; D(\lambda) \; V_k(\lambda) = e_k(\lambda) \quad \text{for } k=1:K \; . \qquad \text{(B-32)}$$

This result gives an expansion of an eigenvalue in terms of its PDs weighted by the $\{\lambda(m)\}$.

A second PD on equation (B-31) yields

$$\frac{\partial^2 e_k(\lambda)}{\partial\lambda(m)\ \partial\lambda(n)} = \frac{\partial V_k(\lambda)'}{\partial\lambda(n)}\ C(m)\ V_k(\lambda) + V_k(\lambda)'\ C(m)\ \frac{\partial V_k(\lambda)}{\partial\lambda(n)} \ . \quad \text{(B-33)}$$

Multiplication by $\lambda(m)$ and summation over m now leads to

$$\sum_{m=1}^{M} \lambda(m)\ \frac{\partial^2 e_k(\lambda)}{\partial\lambda(m)\ \partial\lambda(n)} = \frac{\partial V_k(\lambda)'}{\partial\lambda(n)}\ D(\lambda)\ V_k(\lambda) + V_k(\lambda)'\ D(\lambda)\ \frac{\partial V_k(\lambda)}{\partial\lambda(n)}$$

$$= \frac{\partial V_k(\lambda)'}{\partial\lambda(n)}\ V_k(\lambda)\ e_k(\lambda) + e_k(\lambda)\ V_k(\lambda)'\ \frac{\partial V_k(\lambda)}{\partial\lambda(n)} =$$

$$= e_k(\lambda)\ \frac{\partial}{\partial\lambda(n)}\left[V_k(\lambda)'\ V_k(\lambda)\right] = 0 \quad \text{for } k=1:K\ ,\ n=1:M\ . \quad \text{(B-34)}$$

Thus, the sum of second-order PDs of the eigenvalues, weighted by the $\{\lambda(m)\}$, is always identically zero. Relations (B-31), (B-32), and (B-34) have been checked numerically.

APPENDIX C — MATRIX PROPERTIES

TRACE PROPERTY

Let A be a K×K matrix, not necessarily symmetric. The trace of matrix A, in terms of its elements {A(k,l)}, is

$$\mathrm{tr}(A) = \sum_{k=1}^{K} A(k,k) \ . \tag{C-1}$$

Then, the trace of a product of two K×K matrices follows as

$$\mathrm{tr}(A\ B) = \sum_{k,l=1}^{K} A(k,l)\ B(l,k) = \mathrm{tr}(B\ A) \ . \tag{C-2}$$

It immediately follows that the trace of a product of several K×K matrices, such as C D E F G, can be rearranged, for example, as

$$\mathrm{tr}(C\ D\ E\ F\ G) = \mathrm{tr}(E\ F\ G\ C\ D) \ , \tag{C-3}$$

simply by identifying A as C D and identifying B as E F G. In fact, any cyclic rearrangement of the matrices is allowed without changing the value of the trace. However, if the cyclic pattern is changed (for example, by switching the locations of matrices F and G), the trace is modified.

EIGENVALUE PROPERTY

Let matrix A be K×K and have eigen-decomposition

$$A = V\ E\ V^{-1}\ ,\quad E = \mathrm{diag}\{e_k\}\ ,\ k=1:K\ . \tag{C-4}$$

Then, $A^2 = A\ A = V\ E\ V^{-1}\ V\ E\ V^{-1} = V\ E^2\ V^{-1}$, which can be immediately generalized to

$$A^p = V\ E^p\ V^{-1} \text{ for integer } p\ . \tag{C-5}$$

There follows

$$\mathrm{tr}(A^p) = \mathrm{tr}(V\ E^p\ V^{-1}) = \mathrm{tr}(E^p\ V^{-1}\ V) = \mathrm{tr}(E^p) = \sum_{k=1}^{K} e_k^p\ . \tag{C-6}$$

Thus, the trace of the p-th power of A is equal to the sum of the p-th powers of all the eigenvalues of matrix A.

USEFUL MATRIX PROPERTIES

Let A be a K×K matrix as in equation (C-4). For scalar a, the identity

$$(1 - a)^{-1} = \frac{1}{1 - a} = 1 + a + a^2 + a^3 + \cdots \quad \text{if } |a| < 1\ . \tag{C-7}$$

By using equation (C-4) and $I - A = V\ (I - E)\ V^{-1}$, this generalizes to the matrix relation

$$(I - A)^{-1} = V\ (I - E)^{-1}\ V^{-1} = V\ \mathrm{diag}\{(1 - e_k)^{-1}\}\ V^{-1} =$$

$$= V\ \mathrm{diag}\{1 + e_k + e_k^2 + \cdots\}\ V^{-1} = V\ (I + E + E^2 + \cdots\}\ V^{-1} =$$

$$= I + A + A^2 + A^3 + \cdots \quad \text{if } |e_k| < 1 \text{ for } k=1:K\ . \tag{C-8}$$

That is,

$$\sum_{n=0}^{\infty} A^n = (I - A)^{-1} \quad \text{if } |\mathrm{eig}(A)| < 1 \; , \tag{C-9}$$

where eig(A) denotes all the eigenvalues of matrix A.

A function f(A) of matrix A is defined according to

$$f(A) \equiv V\, f(E)\, V^{-1} \quad \text{where} \quad A = V\, E\, V^{-1} \; . \tag{C-10}$$

Therefore, the relation $-\log(1 - a) = a + a^2/2 + a^3/3 + \cdots$ for scalar a generalizes to

$$-\log(I - A) = -V \log(I - E)\, V^{-1} = -V\, \mathrm{diag}\{\log(1 - e_k)\}\, V^{-1} =$$

$$= -V\, \mathrm{diag}\left(-e_k - \tfrac{1}{2}e_k^2 - \tfrac{1}{3}e_k^3 - \cdots\right)V^{-1} = -V\left(-E - \tfrac{1}{2}E^2 - \tfrac{1}{3}E^3 - \cdots\right)V^{-1}$$

$$= A + \tfrac{1}{2}A^2 + \tfrac{1}{3}A^3 + \cdots \quad \text{if } |e_k| < 1 \text{ for } k=1:K \; . \tag{C-11}$$

That is,

$$\sum_{n=1}^{\infty} \frac{1}{n}A^n = -\log(I - A) \quad \text{if } |\mathrm{eig}(A)| < 1 \; . \tag{C-12}$$

Now, let A(x) be a K×K matrix which is a function of x. Represent the derivative with respect to x by the symbol $\dot{A}(x)$. Then, by use of the chain rule, the derivative of the p-th power of A(x) contains p terms:

$$\frac{d}{dx} A(x)^p = \frac{d}{dx}\Big(A(x) \cdots A(x)\Big) = \dot{A}\, A^{p-1} + A\, \dot{A}\, A^{p-2} + \cdots + A^{p-1}\, \dot{A} \; . \tag{C-13}$$

Then, by using the trace properties in equations (C-2) and (C-3), there follows

$$\mathrm{tr}\left(\frac{d}{dx} A(x)^p\right) = p \; \mathrm{tr}\left(A(x)^{p-1} \dot{A}(x)\right) . \tag{C-14}$$

If matrix B(x) is the inverse of matrix A(x), the derivative of matrix B(x) may be found as follows:

$$B(x) = A(x)^{-1} , \quad B(x) \; A(x) = I , \quad \dot{B}(x) \; A(x) + B(x) \; \dot{A}(x) = 0 ,$$

$$\dot{B}(x) = - B(x) \; \dot{A}(x) \; A(x)^{-1} = - B(x) \; \dot{A}(x) \; B(x) . \tag{C-15}$$

That is, the derivative of the inverse of a matrix A(x) is given by the negative derivative of the matrix A(x), which is then pre- and post-multiplied by the inverse matrix B(x).

The final needed matrix property involves the derivative of equation (C-11). There follows

$$\frac{d}{dx}\left[-\log(I - A)\right] = \dot{A} + \frac{1}{2}\left(\dot{A} \; A + A \; \dot{A}\right) + \frac{1}{3}\left(\dot{A} \; A^2 + A \; \dot{A} \; A + A^2 \; \dot{A}\right) + \cdots \tag{C-16}$$

from which is obtained, by using equation (C-8),

$$\mathrm{tr}\left(\frac{d}{dx}\left[-\log(I - A)\right]\right) = \mathrm{tr}\left(\dot{A} + A \; \dot{A} + A^2 \; \dot{A} + \cdots\right) = \mathrm{tr}\left((I - A)^{-1} \; \dot{A}\right). \tag{C-17}$$

Finally, interchanging the trace and derivative,

$$\frac{d}{dx}\left(\mathrm{tr}\left[-\log[I - A(x)]\right]\right) = \mathrm{tr}\left([I - A(x)]^{-1} \; \dot{A}(x)\right) . \tag{C-18}$$

This holds for all A(x), except if one or more eig(A(x)) = 1.

APPENDIX D – MATLAB PROGRAM quadlinspa

```
clear all       % SPA to joint PDF of M quadratic and linear forms.
M=4;            % Number of quadratic and linear forms.
K=64;           % Number of Gaussian random variables.
tol=1e-7;       % Tolerance in saddlepoint search.
kkmax=100;      % Maximum number of search trials.
f=.499;         % Proximity to boundary at .5

randn('state',0)          % INPUT INFORMATION
A=randn(K,K);             % Positive-definite covariance
R=A*A';                   % matrix, R, of K Gaussian RVs.
r=randn(K,1);             % Mean vector, r, of K Gaussian RVs.
P=zeros(K,K,M);
for m=1:M
A=randn(K,K);
P(:,:,m)=(A+A')*.5;  % Symmetric quadratic terms, P
end
p=randn(K,M);             % Linear terms, p
q=randn(M,1);             % Constant terms, q

z=zeros(M,1);             % SPECIFY FIELD POINT z
S=chol(R);                % KxK
g=randn(K,1);             % Kx1, N(0,1)
w=r+S'*g;                 % Kx1, N(r,R)
for m=1:M
z(m)=(P(:,:,m)*w+p(:,m))'*w+q(m);
end                       % Mx1, field point z

C=zeros(K,K,M);           % PRE-COMPUTATION OF MATRICES
v=zeros(K,M);
c=zeros(M,1);
S=chol(R);                          % KxK
for m=1:M
A=P(:,:,m);                         % KxK
C(:,:,m)=S*A*S';                    % KxK
a=A*r;                              % Kx1
v(:,m)=S*a;                         % Kx1
c(m)=r'*a;                          % 1x1
end
v=S*p+2*v;                          % KxM
c=q+p'*r+c;                         % Mx1

tic                        % SEARCH FOR SADDLEPOINT
L=zeros(M,1);
```

```
B=zeros(K,K,M);
G=zeros(M,1);
H=zeros(M,M);
vt=v';                              % MxK
kk=0;
K2=K*K;
znorm=sqrt(z'*z);
err=z-G;                            % Mx1
while(sqrt(err'*err)/znorm)>tol
    t=v*L;                          % Kx1
    P=reshape(C,K2,M);
    DL=reshape(P*L,K,K);            % KxK
    e=eig(DL);                      % Kx1
    em=max(e);
    if em>=.5
        L=L*(f/em);                 % Mx1
        DL=DL*(f/em);               % KxK
        eigmax=[em kk]
    end
    Q=eye(K)-2*DL;                  % KxK
    qt=Q\t;                         % Kx1
    for m=1:M
        B1=C(:,:,m);                % KxK
        A=Q\B1;                     % KxK
        B(:,:,m)=A;
        G(m)=trace(A)+qt'*B1*qt;
    end
    G=G+vt*qt+c;             % Mx1 Gradient vector
    for m1=1:M
        B1=B(:,:,m1);               % KxK
        tb=2*t'*B1+v(:,m1)';        % 1xK
        for m2=m1:M
            B2=B(:,:,m2);           % KxK
            ts=B1(:)'*reshape(B2',K2,1)...
            +(tb*B2+v(:,m2)'*B1)*qt;
            H(m1,m2)=ts;
            H(m2,m1)=ts;
        end
    end
    H=H*2+vt/Q*v;            % MxM Hessian matrix
    err=z-G;                        % Mx1
    dL=H\err;                       % Mx1
    fr=.3;                   % fraction: [0 1)
```

```
    ff=1-fr^(kk+1);
    L=L+dL*ff;                          % Mx1
    kk=kk+1;
    if kk>kkmax, break, end
end % while
disp(['kk = 'int2str(kk)])

L=L+dL*(1-ff);  % saddlepoint       % Mx1
u=c'*L;                             % 1x1
t=v*L;                              % Kx1
P=reshape(C,K2,M);
DL=reshape(P*L,K,K);                % KxK
e=eig(DL);
if(max(e)>f)
    disp(['eigmax is greater than f = 'num2str(f)])
    keyboard
end
Q=eye(K)-2*DL;                      % KxK
qt=Q\t;                             % Kx1
br=.5*(t'*qt)+u;                    % 1x1
mgf0=1/sqrt(prod(1-2*e));
mgf=mgf0*exp(br);
cgf=log(mgf0)+br;

for m=1:M
    B1=C(:,:,m);                    % KxK
    A=Q\B1;                         % KxK
    B(:,:,m)=A;
    G(m)=trace(A)+qt'*B1*qt;
end
G=G+vt*qt+c;             % Mx1 Gradient vector.
err=z-G;                 % Error in gradient of CGF.
reg=sqrt(err'*err)/znorm;
disp(['rel_err_grad = 'num2str(reg)])
t1=toc;
disp(['t1(sec) = 'num2str(t1)])

tic
BB=zeros(K,K,M,M);
for m1=1:M
    B1=B(:,:,m1);                   % KxK
    tb=2*t'*B1+v(:,m1)';            % 1xK
    for m2=1:M
```

```
        B2=B(:,:,m2);                % KxK
        A=B1*B2;                     % KxK
        BB(:,:,m1,m2)=A;
        if(m1<=m2)
            ts=trace(A)...           % 1x1
            +(tb*B2+v(:,m2)'*B1)*qt;
            H(m1,m2)=ts;
            H(m2,m1)=ts;
        end
    end
end
qv=Q\v;                              % KxM
H=H*2+vt*qv;          % MxM Hessian matrix

den=sqrt((2*pi)^M*det(H));
pdf0=mgf*exp(-z'*L)/den;             % SPA0

T=zeros(M,M,M);
for m1=1:M
for m2=m1:M
A=reshape(BB(:,:,m1,m2)',1,K2);
for m3=m2:M
T(m1,m2,m3)=A*reshape(B(:,:,m3),K2,1);
end, end, end
for m1=1:M
for m2=1:M
for m3=1:M
s=sort([m1 m2 m3]);
T(m1,m2,m3)=T(s(1),s(2),s(3));
end, end, end
T=T*8;        % MxMxM; Third-Order Partial Derivatives

T1=zeros(M,M,M);
T2=zeros(M,M,M);
T3=zeros(M,M,M);
for m=1:M
A2=qv'*C(:,:,m)*qv;                  % MxM
T1(m,:,:)=A2;
T2(:,m,:)=A2;
T3(:,:,m)=A2;
end
Ta=(T1+T2+T3)*2;                     % MxMxM; TO PDs
```

```
for m1=1:M
for m2=1:M
A1=vt*((BB(:,:,m1,m2)+BB(:,:,m2,m1))*qt);
T1(:,m1,m2)=A1;                    % Mx1
T2(m2,:,m1)=A1;
T3(m1,m2,:)=A1;
end, end
Tb=(T1+T2+T3)*4;                   % MxMxM; TO PDs

for m1=1:M
for m2=1:M
B1=t'*BB(:,:,m1,m2);               % 1xK
for m3=1:M
a=B1*B(:,:,m3)*qt;                 % 1x1
T1(m1,m2,m3)=a;
T2(m3,m1,m2)=a;
T3(m2,m3,m1)=a;
end, end, end
Tc=(T1+T2+T3)*8;                   % MxMxM; TO PDs

T=T+Ta+Tb+Tc; % MxMxM; Third-Order Partial Derivatives

F=zeros(M,M,M,M);
for m1=1:M
for m2=m1:M
A=reshape(BB(:,:,m1,m2)',1,K2);
for m3=m2:M
B1=reshape(BB(:,:,m1,m3)',1,K2);
for m4=m3:M
F(m1,m2,m3,m4)=A*reshape(BB(:,:,m3,m4)...
+BB(:,:,m4,m3),K2,1)...
+B1*reshape(BB(:,:,m2,m4),K2,1);
end, end, end, end
for m1=1:M
for m2=1:M
for m3=1:M
for m4=1:M
s=sort([m1 m2 m3 m4]);
F(m1,m2,m3,m4)=F(s(1),s(2),s(3),s(4));
end, end, end, end
F=F*16; % MxMxMxM; Fourth-Order Partial Derivatives

T1=zeros(M,M,M,M);
```

```
T2=T1; T3=T1; T4=T1; T5=T1; T6=T1; T7=T1;
T8=T1; T9=T1; T10=T1; T11=T1; T12=T1;

for m1=1:M
for m2=1:M
A2=vt*(BB(:,:,m1,m2)+BB(:,:,m2,m1))*qv;
T1(m1,m2,:,:)=A2;                    % MxM
T2(m1,:,m2,:)=A2;
T3(m1,:,:,m2)=A2;
T4(:,m1,m2,:)=A2;
T5(:,m1,:,m2)=A2;
T6(:,:,m1,m2)=A2;
end, end
Ta=(T1+T2+T3+T4+T5+T6)*4;            % MxMxMxM; FO PDs

for m1=1:M
for m2=1:M
B1=vt*(BB(:,:,m1,m2)+BB(:,:,m2,m1)); % MxK
for m3=1:M
A1=B1*(B(:,:,m3)*qt);                % Mx1
T1(:,m1,m2,m3)=A1;
T2(m3,:,m1,m2)=A1;
T3(m2,m3,:,m1)=A1;
T4(m1,m2,m3,:)=A1;
T5(:,m1,m3,m2)=A1;
T6(m2,:,m1,m3)=A1;
T7(m3,m2,:,m1)=A1;
T8(m1,m3,m2,:)=A1;
T9(m3,m2,m1,:)=A1;
T10(m2,m1,:,m3)=A1;
T11(m1,:,m3,m2)=A1;
T12(:,m3,m2,m1)=A1;
end, end, end
Tb=(T1+T2+T3+T4+T5+T6+...
T7+T8+T9+T10+T11+T12)*8;             % MxMxMxM; FO PDs

for m1=1:M
for m2=1:M
B1=t'*(BB(:,:,m1,m2)+BB(:,:,m2,m1)); % 1xK
for m3=1:M
for m4=1:M
a=B1*BB(:,:,m3,m4)*qt;               % 1x1
T1(m1,m2,m3,m4)=a;
```

```
T2(m4,m1,m2,m3)=a;
T3(m3,m4,m1,m2)=a;
T4(m2,m3,m4,m1)=a;
T5(m1,m2,m4,m3)=a;
T6(m3,m1,m2,m4)=a;
T7(m4,m3,m1,m2)=a;
T8(m2,m4,m3,m1)=a;
T9(m1,m3,m2,m4)=a;
T10(m4,m1,m3,m2)=a;
T11(m2,m4,m1,m3)=a;
T12(m3,m2,m4,m1)=a;
end, end, end, end
Tc=(T1+T2+T3+T4+T5+T6+...
T7+T8+T9+T10+T11+T12)*8;            % MxMxMxM; FO PDs

F=F+Ta+Tb+Tc; % MxMxMxM; Fourth-Order Partial Derivatives

                    % CALCULATE CORRECTION TERMS
A2=zeros(M,M);
M2=M*M;
Hi=inv(H);                          % MxM
Hr=Hi(:)';                          % 1*M2
for m1=1:M
for m2=1:M
A2(m1,m2)=Hr*reshape(F(:,:,m1,m2),M2,1);
end, end
c4=Hr*A2(:)/8;

A1=zeros(M,1);
for m=1:M
A1(m)=Hr*reshape(T(:,:,m),M2,1);
end
c3a=-A1'*Hi*A1/8;

A3=zeros(M,M,M);
for m1=1:M
B2=Hi(:,m1)';                       % Mx1
for m2=1:M
B3=Hi(:,m2);                        % Mx1
for m3=1:M
A3(m1,m2,m3)=B2*T(:,:,m3)*B3;
end, end, end
B2=zeros(M,M);
```

```
for m1=1:M
B3=reshape(T(:,:,m1),1,M2);
for m2=1:M
B2(m1,m2)=B3*reshape(A3(:,:,m2),M2,1);
end, end
c3b=-Hr*B2(:)/12;

ct=c4+c3a+c3b;        % FIRST-ORDER CORRECTION TERM
disp('c4  c3a  c3b  ct =')
disp([c4 c3a c3b ct])
pdf1=pdf0*(1+ct);
pdfe=pdf0*exp(ct);
pdfb=pdf0*(1+ct/2)/(1-ct/2);
t2=toc;
disp(['t2(sec) = 'num2str(t2)])
disp(['pdf0 = 'num2str(pdf0)])
disp(['pdf1 = 'num2str(pdf1)])
disp(['pdfe = 'num2str(pdfe)])
disp(['pdfb = 'num2str(pdfb)])
```

APPENDIX E – MATLAB PROGRAM quadspa

```
clear all     % SPA to joint PDF of M quadratic forms.
M=4;          % Number of quadratic forms.
K=7;          % Number of Gaussian random variables.
tol=1e-7;     % Tolerance in saddlepoint search.
kkmax=100;    % Maximum number of search trials.
f=.499;       % Proximity to boundary at .5

randn('state',0)       % INPUT INFORMATION R AND P
A=randn(K,K);          % Positive-definite covariance
R=A*A';                % matrix, R, of K Gaussian RVs.
P=zeros(K,K,M);
for m=1:M
A=randn(K,K);
P(:,:,m)=(A+A')*.5;  % Symmetric quadratic terms, P
end

z=zeros(M,1);          % SPECIFY FIELD POINT z
S=chol(R);             % KxK
g=randn(K,1);          % Kx1, N(0,1)
w=S'*g;                % Kx1, N(0,R)
for m=1:M
z(m)=w'*P(:,:,m)*w;
end                    % Mx1, field point z

C=zeros(K,K,M);        % PRE-COMPUTATION OF MATRIX C
S=chol(R);                          % KxK
for m=1:M
C(:,:,m)=S*P(:,:,m)*S';             % KxK
end

tic                    % SEARCH FOR SADDLEPOINT
L=zeros(M,1);
B=zeros(K,K,M);
G=zeros(M,1);
H=zeros(M,M);
kk=0;
K2=K*K;
znorm=sqrt(z'*z);
err=z-G;                            % Mx1
while(sqrt(err'*err)/znorm)>tol
    P=reshape(C,K2,M);
    DL=reshape(P*L,K,K);            % KxK
    e=eig(DL);                      % Kx1
```

```
        em=max(e);
        if em>=.5
            L=L*(f/em);                    % Mx1
            DL=DL*(f/em);                  % KxK
            eigmax=[em kk]
        end
        Q=eye(K)-2*DL;                     % KxK
        for m=1:M
            A=Q\C(:,:,m);                  % KxK
            B(:,:,m)=A;
            G(m)=trace(A);   % Mx1 Gradient vector
        end
        for m1=1:M
            A=reshape(B(:,:,m1)',1,K2);
            for m2=m1:M
                ts=A*reshape(B(:,:,m2),K2,1);
                H(m1,m2)=ts;
                H(m2,m1)=ts;
            end
        end
        H=H*2;            % MxM Hessian matrix
        err=z-G;                           % Mx1
        dL=H\err;                          % Mx1
        fr=.6;            % fraction: [0 1)
        ff=1-fr^(kk+1);
        L=L+dL*ff;                         % Mx1
        kk=kk+1;
        if kk>kkmax, break, end
    end % while
    disp(['kk = 'int2str(kk)])

    L=L+dL*(1-ff);  % saddlepoint      % Mx1
    P=reshape(C,K2,M);
    DL=reshape(P*L,K,K);                   % KxK
    e=eig(DL);
    if(max(e)>f)
        disp(['eigmax is greater than f = 'num2str(f)])
        keyboard
    end
    Q=eye(K)-2*DL;                         % KxK
    mgf=1/sqrt(prod(1-2*e));
    cgf=log(mgf);
```

```
for m=1:M
    A=Q\C(:,:,m);                  % KxK
    B(:,:,m)=A;
    G(m)=trace(A);   % Mx1 Gradient vector
end
err=z-G;              % Error in gradient of cgf
reg=sqrt(err'*err)/znorm;
disp(['rel_err_grad = 'num2str(reg)])
t1=toc;
disp(['t1(sec) = 'num2str(t1)])

tic
BB=zeros(K,K,M,M);
for m1=1:M
    B1=B(:,:,m1);                  % KxK
    for m2=1:M
        A=B1*B(:,:,m2);            % KxK
        BB(:,:,m1,m2)=A;
        if(m1<=m2)
            ts=trace(A);           % 1x1
            H(m1,m2)=ts;
            H(m2,m1)=ts;
        end
    end
end
H=H*2;                  % MxM Hessian matrix

den=sqrt((2*pi)^M*det(H));
pdf0=mgf*exp(-z'*L)/den;          % SPA0

T=zeros(M,M,M);
for m1=1:M
for m2=m1:M
A=reshape(BB(:,:,m1,m2)',1,K2);
for m3=m2:M
T(m1,m2,m3)=A*reshape(B(:,:,m3),K2,1);
end, end, end
for m1=1:M
for m2=1:M
for m3=1:M
s=sort([m1 m2 m3]);
T(m1,m2,m3)=T(s(1),s(2),s(3));
end, end, end
```

```
T=T*8; % MxMxM; Third-Order Partial Derivatives

F=zeros(M,M,M,M);
for m1=1:M
for m2=m1:M
A=reshape(BB(:,:,m1,m2)',1,K2);
for m3=m2:M
B1=reshape(BB(:,:,m1,m3)',1,K2);
for m4=m3:M
F(m1,m2,m3,m4)=A*reshape(BB(:,:,m3,m4)...
+BB(:,:,m4,m3),K2,1)...
+B1*reshape(BB(:,:,m2,m4),K2,1);
end, end, end, end
for m1=1:M
for m2=1:M
for m3=1:M
for m4=1:M
s=sort([m1 m2 m3 m4]);
F(m1,m2,m3,m4)=F(s(1),s(2),s(3),s(4));
end, end, end, end
F=F*16;% MxMxMxM; Fourth-Order Partial Derivatives

             % CALCULATE CORRECTION TERMS
A2=zeros(M,M);
M2=M*M;
Hi=inv(H);                        % MxM
Hr=Hi(:)';                        % 1xM2
for m1=1:M
for m2=1:M
A2(m1,m2)=Hr*reshape(F(:,:,m1,m2),M2,1);
end, end
c4=Hr*A2(:)/8;

A1=zeros(M,1);
for m=1:M
A1(m)=Hr*reshape(T(:,:,m),M2,1);
end
c3a=-A1'*Hi*A1/8;

A3=zeros(M,M,M);
for m1=1:M
B2=Hi(:,m1)';                     % Mx1
for m2=1:M
```

```
B3=Hi(:,m2);                        % Mx1
for m3=1:M
A3(m1,m2,m3)=B2*T(:,:,m3)*B3;
end, end, end
B2=zeros(M,M);
for m1=1:M
B3=reshape(T(:,:,m1),1,M2);
for m2=1:M
B2(m1,m2)=B3*reshape(A3(:,:,m2),M2,1);
end, end
c3b=-Hr*B2(:)/12;

ct=c4+c3a+c3b;  % FIRST-ORDER CORRECTION TERM
disp('c4  c3a  c3b  ct =')
disp([c4 c3a c3b ct])
pdf1=pdf0*(1+ct);
pdfe=pdf0*exp(ct);
pdfb=pdf0*(1+ct/2)/(1-ct/2);
t2=toc;
disp(['t2(sec) = 'num2str(t2)])
disp(['pdf0 = 'num2str(pdf0)])
disp(['pdf1 = 'num2str(pdf1)])
disp(['pdfe = 'num2str(pdfe)])
disp(['pdfb = 'num2str(pdfb)])
```

REFERENCES

1. M. G. Kendall and A. Stuart, **The Advanced Theory of Statistics, Volume 1, Distribution Theory**, Hafner Publishing Company, New York, NY, 1969. See equation (4.14).

2. A. H. Nuttall, "Numerical Evaluation of Cumulative Probability Distribution Functions Directly from Characteristic Functions," NUSL Report Number 1032, Naval Underwater Systems Center, New London, CT, 11 August 1969.

3. A. H. Nuttall, "Numerical Evaluation of Cumulative Probability Distribution Functions Directly from Characteristic Functions," **Proceedings of the Institute of Electrical and Electronics Engineers**, volume 57, number 11, pages 2071 - 2072, November 1969.

4. A. H. Nuttall, "Accurate Efficient Evaluation of Cumulative or Exceedance Probability Distributions Directly from Characteristic Functions," NUSC Technical Report 7023, Naval Underwater Systems Center, New London, CT, 1 October 1983.

5. A. H. Nuttall, "Evaluation of Small Tail Probabilities Directly from the Characteristic Function," NUWC-NPT Technical Report 10,840, Naval Undersea Warfare Center Division, Newport, RI, 15 September 1997.

6. P. M. McCullagh, **Tensor Methods in Statistics**, Chapman and Hall, England, 1987.

INITIAL DISTRIBUTION LIST

Addressee	No. of Copies
Center for Naval Analyses, VA	1
Coast Guard Academy, CT	
J. Wolcin	1
Commander Submarine Force, U.S. Pacific Fleet, HI	
W. Mosa, CSP N72	1
Defense Technical Information Center	2
Griffiss Air Force Base, NY	
Documents Library	1
J. Michels	1
Hanscom Air Force Base, MA	
M. Rangaswamy	1
National Radio Astronomy Observatory, VA	
F. Schwab	1
National Security Agency, MD	
J. Maar	1
National Technical Information Service, VA	10
Naval Environmental Prediction Research Facility, CA	1
Naval Intelligence Command, DC	1
Naval Oceanographic and Atmospheric Research Laboratory, MS	
B. Adams	1
R. Fiddler	1
E. Franchi	1
R. Wagstaff	1
Naval Oceanographic Office, MS	1
Naval Personnel Research and Development Center, CA	1
Naval Postgraduate School, CA	
Superintendent	1
C. Therrien	1
Naval Research Laboratory, DC	
W. Gabriel	1
E. Wald	1
N. Yen	1
Naval Surface Warfare Center, FL	
E. Linsenmeyer	1
D. Skinner	1
Naval Surface Warfare Center, VA	
J. Gray	1
Naval Technical Intelligence Center, DC	
Commanding Officer	1
D. Rothenberger	1
Naval Undersea Warfare Center, FL	
Officer in Charge	1
Naval Weapons Center, CA	1
Office of the Chief of Naval Research, VA	
ONR 321 (D. Johnson)	1
ONR 321US (J. Tague)	1
ONR 322 (R. Tipper)	1
ONR 334 (P. Abraham)	1

INITIAL DISTRIBUTION LIST (Cont'd)

Addressee	No. of Copies
Office of Naval Research	
ONR 31 (R. R. Junker)	1
ONR 311 (A. M. van Tilborg)	1
ONR 312 (M. N. Yoder)	1
ONR 313 (N. L. Gerr)	1
ONR 32 (S. E. Ramberg	1
ONR 321 (F. Herr)	1
ONR 33 (S. G. Lekoudis)	1
ONR 334 (A. J. Tucker)	1
ONR 342 (W. S. Vaughan)	1
ONR 343 (R. Cole)	1
ONR 362 (M. Sponder)	1
Naval Sea Systems Command (SEA-93, ASTO), VA	
J. Thompson, A. Hommel, R. Zarnich	3
U.S. Air Force, Maxwell Air Force Base, AL	
Air University Library	1
Vandenberg Air Force Base, CA	
CAPT R. Leonard	1
Brown University, RI	
Documents Library	1
Catholic University of America, DC	
J. McCoy	1
Drexel University, PA	
S. Kesler	1
Duke University, NC	
J. Krolik	1
Harvard University, MA	
Gordon McKay Library	1
Johns Hopkins University, Applied Physics Laboratory, MD	
H. M. South	1
T. N. Stewart	1
Lawrence Livermore National Laboratory, CA	
L. Ng	1
Los Alamos National Laboratory, NM	1
Marine Biological Laboratory, MA	
Library	1
Massachusetts Institute of Technology, MA	
Barker Engineering Library	1
Massachusetts Institute of Technology, Lincoln Laboratory, MA	
V. Premus	1
J. Ward	1
Northeastern University, MA	
C. Nikias	1
Pennsylvania State University, Applied Research Laboratory, PA	
R. Hettche	1
E. Liszka	1
F. Symons	1
Princeton University, NJ	
S. Schwartz	1
Rutgers University, NJ	
S. Orfanidis	1

INITIAL DISTRIBUTION LIST (Cont'd)

Addressee	No. of Copies
San Diego State University, CA	
F. Harris	1
Scripps Institution of Oceanography, Marine Physical Laboratory, CA	
Director	1
Syracuse University, NY	
D. Weiner	1
Engineering Societies Information Center, Kansas City, MO	
Linda Hall Library-East	1
University of Colorado, CO	
L. Scharf	1
University of Connecticut, CT	
Wilbur Cross Library	1
C. Knapp	1
P. Willett	1
University of Florida, FL	
D. Childers	1
University of Hartford	
Science and Engineering Library	1
University of Illinois, IL	
D. Jones	1
University of Illinois at Chicago, IL	
A. Nehorai	1
University of Massachusetts, MA	
C. Chen	
University of Massachusetts, No. Dartmouth, MA	
J. Buck	1
University of Michigan, MI	
Communications and Signal Processing Laboratory	1
W. Williams	1
University of Minnesota, MN	
M. Kaveh	1
University of Rhode Island, RI	
Library	1
G. Boudreaux-Bartels	1
S. Kay	1
D. Tufts	1
University of Rochester, NY	
E. Titlebaum	1
University of Southern California, CA	
W. Lindsey	1
A. Polydoros	1
University of Texas, TX	
Applied Research Laboratory	1
C. Penrod	1
University of Washington, WA	
Applied Physics Laboratory	1
D. Lytle	1
J. Ritcey	1
R. Spindel	1

INITIAL DISTRIBUTION LIST (Cont'd)

Addressee	No. of Copies
Villanova University, PA	
M. Amin	1
Woods Hole Oceanographic Institution, MA	
Director	1
T. Stanton	1
Yale University, CT	
Kline Science Library	1
P. Schultheiss	1
Analysis and Technology, CT	
Library	1
Analysis and Technology, VA	
D. Clark	1
Atlantic Aerospace Electronics Corp.	
R. Stahl	1
Bell Communications Research, NJ	
D. Sunday	1
Berkeley Research, CA	
S. McDonald	1
Bolt, Beranek, and Newman, CT	
P. Cable	1
Bolt, Beranek, and Newman, MA	
H. Gish	1
DSR, Inc., VA	
M. Bozek-Kuzmicki	1
EG&G Services, CT	
J. Pratt	1
Engineering Technology Center	
D. Lerro	1
General Electric, NJ	
H. Urkowitz	1
Harris Scientific Services, NY	
B. Harris	1
Hughes Defense Communications, IN	
R. Kenefic	1
Kildare Corporation, CT	
R. Mellen	1
Lincom Corporation, MA	
T. Schonhoff	1
Lockheed Martin, Undersea Systems, VA	
M. Flicker	1
Lockheed Martin, Ocean Sensor Systems, NY	
R. Schumacher	1
Marconi Aerospace Defense Systems, TX	
R. D. Wallace	1
MITRE Corporation, VA	
S. Pawlukiewicz	1
R. Bethel	1
Neural Technology, Inc., SC	
E. A. Tagliarini	1
Orincon Corporation, VA	
H. Cox	1

INITIAL DISTRIBUTION LIST (Cont'd)

Addressee	No. of Copies
Philips Research Laboratory, Netherlands	
A. J. E. M. Janssen	1
Planning Systems, Inc., CA	
W. Marsh	1
Prometheus, RI	
M. Barrett	1
J. Byrnes	1
Raytheon, RI	
R. Conner	1
S. Reese	1
Schlumberger-Doll Research, CT	
R. Shenoy	1
Science Applications International Corporation, CA	
C. Katz	1
Science Applications International Corporation, VA	
P. Mikhalevsky	1
Toyon Research, CA	
M. Van Blaricum	1
Tracor, TX	
T. Leih	1
TRW, VA	
R. Prager	1
G. Maher	1
Westinghouse Electric, MA	
R. Kennedy	1
Westinghouse Electric, Annapolis, MD	
H. Newman	1
Westinghouse Electric, Baltimore, MD	
R. Park	1
K. Harvel, Austin, TX	1