The program PDVPDS computes the probability of detection compared to the optimum probability of detection results of Figures 7.23a and 7.23b, and the program PEVPES computes the probability of error versus the optimum probability of error characteristics, Figure 7.24.

Finally, the programs WOA and WOB compute the pdf, evaluated at zero, for Class A and Class B noise, Figures 7.25 and 7.26.

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

```
      PROGRAM NORMB(INPUT,OUTPUT)
C     THIS PROGRAM COMPUTES THE NORMALIZATION FACTOR OMEGA
C     FOR THE TRUNCATED CLASS B MODEL SO THAT THE ENVELOPE RMS
C     VALUE IS EQUAL TO 1. SATURATION(TRUNCATION)IS ASSUMED
C     TO BE AT 80DB(ON ORIGINAL NORMALIZED TO GAUSS POWER
C     SCALE)OR AT P=1.E-6, WHICHEVER OCCURS FIRST.
C     NOTE. THIS PROGRAM NORMALIZES TO THE ENVELOPE RMS. THE REAL
C     NOISE POWER IS ONE HALF THE ENVELOPE POWER. FOR COMPUTATIONS
C     WHICH USE THE INSTANTANEOUS AMPLITUDE PDF, THE PROPER
C     NORMALIZATION IS OBTAINED BY USING 2.*OMEGA, OMEGA BEING
C     THE NORMALIZATION PARAMETER OBTAINED HERE.
      DIMENSION IRAY(6),AALPHA(9),AAA(6)
      DATA IRAY/-1,-1,-1,0,-1,-1/
      DATA AALPHA/0.2,0.4,0.6,0.8,1.0,1.2,1.4,1.6,1.8/
      DATA AAA/0.001,0.01,0.1,0.5,1.0,2.0/
      PRINT 6
    6 FORMAT(1H1)
      DO 80 I=1,9
      ALPHA=AALPHA(I)
      DO 70 J=1,6
      AA=AAA(J)
      SS=0.
      PE2=1.
      DO 60 K=1,25
      E=10.**((-22.+K*4.)/20.)
      PE1=PE2
      SS1=0. $ FN=1. $ SS2=0.
      E2=10.**((-20.+K*4.)/20.)
      DO 40 N=1,25
      FN=FN*N
      CALL CONHYP(1.-N*ALPHA/2.,2.,E2*E2,S,IOVFLW)
      T=(((-AA)**N)/FN)*GAMMA(1.+N*ALPHA/2.)*S
      IF(IOVFLW.NE.1) GO TO 24
      SS2=SS2+T
      GO TO 40
   24 SS1=SS1+T
   40 CONTINUE
      FP=0.
      IF(E2*E2.LT.675.) FP=EXP(-E2*E2)
      PE2=FP-E2*E2*(FP*SS1+SS2)
      SS=SS+E*E*(PE1-PE2)
      IF(PE2.LE.1.E-6) GO TO 64
   60 CONTINUE
   64 RMSS=SS+PE2*E2
      RMS=SQRT(RMSS)
      RMSDB=20.*ALOG10(RMS)
      OMEGA=1./RMSS
      PRINT 7, ALPHA, AA, RMSDB, OMEGA
   70 CONTINUE
   80 CONTINUE
    7 FORMAT(5X,3(1PE9.2,2X),2X,1PE12.5)
      END
```

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

```
  150 S=S*(GAMMA(B)/GAMMA(A))*(X**(A-B))
      IF(X.LT.675.)GO TO 190
      IOVFLW=1
      GO TO 200
  190 S=S*EXP(X)
  200 RETURN
 1000 FORMAT(//,1X,* CANNOT EVALUATE EXPRESSION SINCE BOTH*,
     1* A AND B ARE NEGATIVE INTEGERS OR ZERO, A=*,F10.2,* , B=*,
     2F10.2,//)
 1001 FORMAT(//,1X,* BAD VALUE FOR B GIVES INFINITE RESULT FOR S*,
     1 *, B=*,F10.2,//)
      END
```

```
      FUNCTION GAMMA(X)
C     RETURNS THE GAMMA FUNCTION FOR REAL ARGUMENT.
C     NOTE. THE GAMMA FUNCTION IS NOT DEFINED FOR A NEGATIVE INTEGER OR ZER
C     INPUT
C       X = THE REAL ARGUMENT.
C     OUTPUT.
C       GAMMA(X) = THE GAMMA FUNCTION OF ARGUMENT X.
   75 FORMAT(66H GAMMA FUNCTION OF A NEGATIVE INTEGER, OR OF ZERO, IS NO
     1T DEFINED.)
    5 IF(X) 10,80,15
   10 N=-X
      EN=-N-1
      V=X-EN
      IF(V.EQ.1.)80,20
   15 N=X
      EN=N
      V=X-EN
   20 GAMMA=1.+V*(.422784337+V*(.4118402518+V*(.08157821878+V*
     1(.07423790761+V*(-.0002109074673+V*(.01097369584+V*(-.002466747981
     2+V*(.001539768105-V*(.0003442342046-V*.000067710571171)))))))))
      IF(EN-2.) 37,25,30
   25 RETURN
   30 N=N-1
      DO 35 I=2,N
      FI=I
   35 GAMMA=GAMMA*(FI+V)
      RETURN
   37 N=2.-EN
      DO 40 I=1,N
      FI=2-I
   40 GAMMA=GAMMA/(FI+V)
      RETURN
   80 PRINT 75
      CALL EXIT
      END
```

```
      PROGRAM LOBDNA(INPUT,OUTPUT)
C     THIS PROGRAM COMPUTES THE LOBD NON-LINEARITY FOR CLASS A
C     NOISE FOR BOTH THE CANONICAL AND QUASICANONICAL MODEL.
C     THE PARAMETERS AA,GAM,ALPHAO,U(FOR MU),SGAM,AND GO ARE INPUTS.
C     THE DERIVED PARAMETERS,DSQ,ALPHA,AND GAMH ARE COMPUTED.
C     PC IS THE CANONICAL NON-LINEARITY AND PQC IS THE
C     QUASICANONICAL NON-LINEARITY.
      DIMENSION IRAY(6),AA(4),GAM(4)
      DATA IRAY/-1,-1,-1,0,-1,-1/
      DATA AA/0.35,0.1,0.01,0.35/
      DATA GAM/5.E-4,0.001,5.E-4,0.1/
      CALL SYSTEMC(115,IRAY)
    6 FORMAT(1H1)
      ALPHAO=0.01
      U=0.
      SGAM=2.
      GO=1.
      PI=3.1415926
      SRPI=SQRT(PI)
      ALPHA=(2.-U)/SGAM
      DSQ=ALPHA/((2.-ALPHA)*(ALPHAO**((2.-ALPHA)*SGAM)))
      D=SQRT(DSQ)
      DO 90 I=1,4
      GAMM=GAM(I)
      GAMH=GAM(I)*DSQ
      A=AA(I)
      PRINT 6
      PRINT 7, GAMM, GAMH, A
      DO 80 J=1,51
      ZDB=-62.+J*2.
      Z=10.**(ZDB/20.)
      ZSQ=Z*Z
      FM=1.
      SIGCO=GAMM/(1.+GAMM)
      SIGQO=GAMH/(1.+GAMM)
      SUMCB=EXP(-ZSQ/(2.*SIGCO))/SQRT(2.*PI*SIGCO)
      SUMCT=SUMCB*Z/SIGCO
      SUMQB=D*EXP(-ZSQ*DSQ/(2.*SIGQO))/SQRT(2.*PI*SIGQO)
      SUMQT=SUMQB*DSQ*Z/SIGQO
      DO 70 M=1,15
      FM=FM*M
      SIGC=(M/A+GAMM)/(1.+GAMM)
      SIGQ=(M/A+GAMH)/(1.+GAMM)
      TCB=(A**M/FM)*EXP(-ZSQ/(2.*SIGC))/SQRT(2.*PI*SIGC)
      TCT=TCB*Z/SIGC
      F=((A*GO)**M)/FM
      TQB=F*D*EXP(-ZSQ*DSQ/(2.*SIGQ))/SQRT(2.*PI*SIGQ)
      TQT=TQB*DSQ*Z/SIGQ
      FN=1. $ SB=0. $ ST=0.
      F1=PI/(2.*GAMMA((3.+ALPHA)/2.)*((M+GAMM*A*DSQ)**(ALPHA/2.)))
```

```
      DO 60 N=1,M
      FN=FN*N
      FMN=1.
      KK=M-N
      IF(KK.EQ.0) GO TO 21
      DO 20 K=1,KK
      FMN=FMN*K
   20 CONTINUE
   21 CMN=FM/(FN*FMN)
      F2=GAMMA((1.+N*ALPHA)/2.)*F1**N
      ARG=ZSQ*DSQ/(2.*SIGQ)
      CALL CONHYP(-0.5*N*ALPHA,0.5,ARG,BS,IOVFLW)
      IF(IOVFLW.EQ.1) GO TO 31
      S=((-1.)**N)*CMN*EXP(-ARG)*F2*BS
      GO TO 32
   31 S=((-1.)**N)*CMN*F2*BS
   32 SB=SB+S
      CALL CONHYP(-0.5*N*ALPHA,1.5,ARG,TS,IOVFLW)
      IF(IOVFLW.EQ.1) GO TO 33
      SS=((-1.)**N)*CMN*(1.+N*ALPHA)*EXP(-ARG)*F2*TS
      GO TO 34
   33 SS=((-1.)**N)*CMN*(1.+N*ALPHA)*F2*TS
   34 ST=ST+SS
   60 CONTINUE
      H=D/SQRT(2.*PI*PI*SIGQ)
      HT=H*Z*DSQ/SIGQ
      SB=F*H*SB
      ST=F*HT*ST
      SUMCB=SUMCB+TCB
      SUMCT=SUMCT+TCT
      SUMQB=SUMQB+TQB+SB
      SUMQT=SUMQT+TQT+ST
   70 CONTINUE
      PC=SUMCT/SUMCB
      PQC=SUMQT/SUMQB
      PRINT 8, ZDB, PC, PQC
   80 CONTINUE
   90 CONTINUE
    7 FORMAT(2X,3(1PE12.5,2X),/)
    8 FORMAT(5X,F5.1,2X,1PE12.5,2X,1PE12.5)
      END
```

```
      PROGRAM LOBDNB(INPUT,OUTPUT)
C     THIS PROGRAM COMPUTES THE LOBD NON-LINEARITY FOR
C     CLASS B NOISE. THE INPUT PARAMETERS ARE ALPHA,AA,AND OMEGA.
C     THE NON-LINEARITY VALUES ARE GIVEN BY ZZ.
      DIMENSION IRAY(6),AALPHA(3),AAA(3),OOMEGA(9)
      DATA IRAY/-1,-1,-1,0,-1,-1/
      DATA AALPHA/0.8,1.0,1.2/
      DATA AAA/0.2,1.0,2.0/
      DATA OOMEGA/2.0087E-4,4.0202E-5,2.0115E-5,9.9889E-4,
     11.9996E-4,1.0000E-4,5.1565E-3,1.0357E-3,5.1816E-4/
      CALL SYSTEMC(115,IRAY)
      DO 80 I=1,3
      ALPHA=AALPHA(I)
      DO 70 J=1,3
      AA=AAA(J)
      OMEGA=2.*OOMEGA(3*(I-1)+J)
C     TO NORMALIZE TO REAL NOISE RMS.
      PRINT 6
      PRINT 7, ALPHA, AA, OMEGA
      DO 60 K=1,25
      ZDB=-65.+K*5.
      Z=10.**(ZDB/20.)
      ZN=Z*Z/OMEGA
      SUM=0. $ FN=1. $ SUM1=0.
      DO 20 N=1,26
      FN=FN*N
      CALL CONHYP(-N*ALPHA/2.,1.5,ZN,S,IOVFLW)
      T=(((-AA)**N)/FN)*GAMMA(.5+N*ALPHA/2.)*S*(1.+N*ALPHA)
      IF(IOVFLW.NE.1) GO TO 14
      SUM1=SUM1+T
      GO TO 20
   14 SUM=SUM+T
   20 CONTINUE
      FP=0.
      IF(ZN.LT.675.) FP=EXP(-ZN)
      TOP=FP*(SUM+1.7724539)+SUM1
      SUMM=0. $ FM=1. $ SUMM1=0.
      DO 40 M=1,26
      FM=FM*M
      CALL CONHYP(-M*ALPHA/2.,0.5,ZN,S,IOVFLW)
      TT=(((-AA)**M)/FM)*GAMMA(.5+M*ALPHA/2.)*S
      IF(IOVFLW.NE.1) GO TO 34
      SUMM1=SUMM1+TT
      GO TO 40
   34 SUMM=SUMM+TT
   40 CONTINUE
      FPP=0.
      IF(ZN.LT.675.) FPP=EXP(-ZN)
      BOT=FPP*(SUMM+1.7724539)+SUMM1
      ZZ=2.*(Z/OMEGA)*TOP/BOT
      PRINT 8, ZDB, ZZ
   60 CONTINUE
   70 CONTINUE
   80 CONTINUE
    6 FORMAT(1H1)
    7 FORMAT(2X,2(F4.1,2X),1PE12.5,/)
    8 FORMAT(5X,F5.1,1PE12.5)
      END
```

```
      PROGRAM PC1(INPUT,OUTPUT)
C     TO COMPUTE THE GENERAL PERFORMANCE CURVES,
C     FIGURES 7.3 AND 7.4, EQUATION 7.13.
      DIMENSION ALPHA(9)
      DATA ALPHA/1.E-1,1.E-2,1.E-3,1.E-4,1.E-5,1.E-6,1.E-8,
     11.E-10,1.E-12/
      PRINT 6
    6 FORMAT(1H1)
      DO 20 I=1,25
      SIGDB=-22.+2.*I
      SIG=10.**(SIGDB/20.)
      PE=0.5*CERF(SIG/(2.*SQRT(2.)))
      PRINT 7, SIGDB, PE
   20 CONTINUE
    7 FORMAT(10X,F5.1,2X,1PE12.5)
      PRINT 6
      DO 60 J=1,9
      T1=ERFIN(1.0-2.*ALPHA(J))
      DO 40 K=1,25
      SIGDB=-22.+2.*K
      SIG=10.**(SIGDB/20.)
      T=SIG/SQRT(2.)-T1
      IF(T.LE.0.) GO TO 30
      PD=1.-0.5*CERF(T)
      GO TO 35
   30 T=-T
      PD=0.5*CERF(T)
   35 PRINT 7, SIGDB, PD
   40 CONTINUE
      PRINT 8
   60 CONTINUE
    8 FORMAT(//)
      END
```

```
      PROGRAM PC3(INPUT,OUTPUT)
C     TO COMPUTE THE RESULTS GIVEN ON FIG 7.6, EQ. 7.14.
      DIMENSION PE(13)
      DATA PE/0.4,0.2,0.1,5.E-2,1.E-2,5.E-3,1.E-3,5.E-4,1.E-4,
     15.E-5,1.E-5,5.E-6,1.E-6/
      PRINT 6
    6 FORMAT(1H1)
      DO 10 I=1,13
      C=ERFIN(1.-2.*PE(I))
      CDB=10.*ALOG10(C)
      PRINT 7, PE(I), C, CDB
   10 CONTINUE
    7 FORMAT(10X,3(1PE12.5,2X))
      END
```

```
      PROGRAM PC2(INPUT,OUTPUT)
C     TO COMPUTE THE PROBABILISTIC CONTROLS ON DETECTION,
C     FIGURE 7.5, EQUATION 7.14.
      DIMENSION ALPHA(9), PDET(16), P1(9), P2(16)
      DATA ALPHA/1.E-1,1.E-2,1.E-3,1.E-4,1.E-5,1.E-6,
     1 1.E-8,1.E-10,1.E-12/
      DATA PDET/0.02,0.04,0.06,0.08,0.1,0.2,0.3,0.4,0.5,0.6,
     10.7,0.8,0.9,0.95,0.98,0.99/
      PRINT 6
    6 FORMAT(1H1)
      DO 10 I=1,9
      P1(I)=ERFIN(1.-2.*ALPHA(I))
   10 CONTINUE
      DO 20 J=1,16
      P2(J)=ERFIN(ABS(2.*PDET(J)-1.))
      IF(2.*PDET(J)-1..LE.0.)  P2(J)=-P2(J)
   20 CONTINUE
      DO 40 K=1,9
      PC=P1(K)
      DO 30 L=1,16
      PDB=0.
      P=PC+P2(L)
      IF(P.LE.0.) GO TO 29
      PDB=10.*ALOG10(P)
   29 PRINT 7, PC, P2(L), P, PDB
   30 CONTINUE
      PRINT 6
   40 CONTINUE
    7 FORMAT(10X,4(1PE12.5,2X))
      END


      FUNCTION CERF(X)
C     SEE APPROXIMATIONS FOR DIGITAL COMPUTERS
C     BY C. HASTINGS, PRINCETON U. PRESS, 1955,
C     PAGE 169. ALSO IN ABRAMOWITZ AND STEGUN.
C     NOTE, VALID ONLY FOR X.GE.0..
      E=1.0/(1.0+0.3275911*X)
      S=(((((0.940646070*E)-1.287822453)*E+1.259695130)*E-0.252128668)*E
     1+0.225836846)*E
      XSQ=X**2
      EXPFX=0.0
      IF(XSQ.LT.709.0)EXPFX=EXP(-XSQ)
      CERF=S*EXPFX*1.128379167
      RETURN
      END

      FUNCTION ERFIN(Q)
C     COMPUTES THE INVERSE ERROR FUNCTION, USING
C     26.2.23 OF ABRAMOWITZ AND STEGUN.
      P=(1.-Q)/2.
      T=SQRT(ALOG(1./(P*P)))
      X1=2.515517+0.802853*T+0.010328*T*T
      X2=1.0+1.432788*T+0.189269*T*T+0.001308*T*T*T
      XP=T-X1/X2
      ERFIN=XP/(SQRT(2.))
      RETURN
      END
```

```
      PROGRAM PARA(INPUT,OUTPUT)
C     THIS PROGRAM COMPUTES L(2),L(4),L(2,2) AND L(6)
C     FOR VARIOUS COMBINATIONS OF CLASS A NOISE
C     PARAMETERS.
C     THE COHERENT PROCESSING GAIN, PER SAMPLE, IS L(2).
C     THE INCOHERENT PROCESSING GAIN, PER SAMPLE, IS
C     IS A FUNCTION OF L(2),L(4) AND THE SIGNAL PARAMETER QN.
C     THE RATIO OF THE COHERENT AND INCOHERENT PROCESSING
C     GAINS IS ALSO CALCULATED ALONG WITH XO AND YO.
      COMMON/QQQ/A,GAM
      DIMENSION IRAY(6),AA(10),G(7),QN(2)
      DIMENSION PIINC(2),PIDB(2),RATIO(2),RADB(2),YO(2)
      DIMENSION Z1(7),Z2(7),Z3(7),Z4(7)
      DATA AA/1.E-4,1.E-2,0.1,0.5,1.0,2.0,3.0,4.0,5.0,10.0/
      DATA G/1.E-8,1.E-7,1.E-6,1.E-5,1.E-4,1.E-3,1.E-2/
      DATA QN/1.,10./
      DATA IRAY/-1,-1,-1,0,-1,-1/
      CALL SYSTEMC(115,IRAY)
      PRINT 6
    6 FORMAT(1H1)
      DO 80 L=1,10
      A=AA(L)
      DO 70 L1=1,7
      GAM=G(L1)
      SUM1=0.$SUM2=0.$SUM3=0.$SUM4=0.
      DO 50 I=1,36
      BDB=-160.+5.*(I-1)
      CDB=-160.+5.*I
      B=10.**(BDB/20.)
      C=10.**(CDB/20.)
      IF(I.EQ.1) B=0.
      DX=(C-B)/6.
      DO 40 J=1,7
      X=B+(J-1)*DX
      CALL FUN1(X,Z1(J),Z3(J))
      CALL FUN2(X,Z2(J),Z4(J))
   40 CONTINUE
      S1=0.3*DX*(Z1(1)+5.*Z1(2)+Z1(3)+6.*Z1(4)+Z1(5)+5.*Z1(6)+Z1(7))
      S2=0.3*DX*(Z2(1)+5.*Z2(2)+Z2(3)+6.*Z2(4)+Z2(5)+5.*Z2(6)+Z2(7))
      S3=0.3*DX*(Z3(1)+5.*Z3(2)+Z3(3)+6.*Z3(4)+Z3(5)+5.*Z3(6)+Z3(7))
      S4=0.3*DX*(Z4(1)+5.*Z4(2)+Z4(3)+6.*Z4(4)+Z4(5)+5.*Z4(6)+Z4(7))
      SUM1=SUM1+S1
      SUM2=SUM2+S2
      SUM3=SUM3+S3
      SUM4=SUM4+S4
   50 CONTINUE
```

```
      SUM1=2.*SUM1
C     SUM1 IS L(2)
      SUM2=2.*SUM2
C     SUM2 IS L(4)
      SL4DB=10.*ALOG10(SUM2)
      SUM3=2.*SUM3
C     SUM3 IS L(2,2)
      SL22DB=10.*ALOG10(SUM3)
      SUM4=2.*SUM4
C     SUM4 IS L(6)
      SL6DB=10.*ALOG10(ABS(SUM4))
      SL2=SUM1
      SL2DB=10.*ALOG10(SL2)
      X0=SL2/(SUM3/2.-SL2*SL2)
      DO 60 K=1,2
      F=2.*SL2*SL2/SUM2
      PIINC(K)=(SUM2/8.)*(1.+F*(ON(K)-1.))
      PIDB(K)=10.*ALOG10(PIINC(K))
      RATIO(K)=PIINC(K)/SL2
      RADB(K)=10.*ALOG10(RATIO(K))
   60 CONTINUE
      Y0(1)=SUM2/ABS(SUM4/2.+6.*SL2*SUM3)
      Y0(2)=1./ABS(3.*SUM3/SL2+2.*SL2)
      PRINT 7, A, GAM
      PRINT 8, SL2, SUM2, SUM3, SUM4
      PRINT 9, X0, Y0(1), Y0(2)
      PRINT 10, PIINC(1), PIINC(2), RATIO(1), RATIO(2)
      PRINT 11, SL2DB, PIDB(1), PIDB(2), RADB(1), RADB(2)
      PRINT 12, SL4DB, SL22DB, SL6DB
   70 CONTINUE
   80 CONTINUE
    7 FORMAT(2X,2(1PE10.3,2X))
    8 FORMAT(5X,4(1PE10.3,2X))
    9 FORMAT(5X,3(1PE10.3,2X))
   10 FORMAT(5X,4(1PE10.3,2X))
   11 FORMAT(5X,5(1PE10.3,2X))
   12 FORMAT(5X,3(1PE10.3,2X),/)
      END
```

```
      SUBROUTINE FUN1(X,Y,YY)
C     THIS SUBROUTINE COMPUTES THE L(2) AND L(2,2) INTEGRANDS
C     FOR CLASS A NOISE.
      COMMON/QQQ/A,GAM
      PI=3.141592654
      SM1=0. $ SM2=0. $ FM=1.
      DO 10 MM=1,26
      M=MM-1
      IF(M.NE.0) FM=FM*M
      SIGSQ=(M/A+GAM)/(1.+GAM)
      T1=((A**M)/FM)*EXP(-X*X/(2.*SIGSQ))/((2.*PI*SIGSQ)**0.5)
      T2=T1/SIGSQ
      SM1=SM1+T1
      SM2=SM2+T2
   10 CONTINUE
      TEMP=X*X*SM2*SM2/SM1
      Y=EXP(-A)*TEMP
C     Y IS THE L(2) INTEGRAND.
      YY=2.*EXP(-A)*TEMP*TEMP/SM1
C     YY IS THE L(2,2) INTEGRAND
      RETURN
      END



      SUBROUTINE FUN2(X,Y,YY)
C     THIS SUBROUTINE COMPUTES THE L(4) AND L(6) INTEGRANDS
C     FOR CLASS A CANONICAL NOISE.
      COMMON/QQQ/A,GAM
      PI=3.141592654
      SM1=0. $ SM2=0. $ SM3=0. $ FM=1.
      DO 10 MM=1,26
      M=MM-1
      IF(M.NE.0) FM=FM*M
      SIGSQ=(M/A+GAM)/(1.+GAM)
      T1=((A**M)/FM)*EXP(-X*X/(2.*SIGSQ))/(SQRT(2.*PI*SIGSQ))
      T2=T1/SIGSQ
      T3=T2/SIGSQ
      SM1=SM1+T1
      SM2=SM2+T2
      SM3=SM3+T3
   10 CONTINUE
      G=X*X*SM3-SM2
      TEMP=G*G/SM1
      Y=EXP(-A)*TEMP
C     Y IS THE L(4) INTEGRAND.
      YY=EXP(-A)*TEMP*G/SM1
C     YY IS THE L(6) INTEGRAND.
      RETURN
      END
```

```
      PROGRAM PARB(INPUT,OUTPUT)
C     THIS PROGRAM COMPUTES L(2),L(4),L(2,2),AND L(6)
C     FOR VARIOUS COMBINATIONS OF CLASS B NOISE
C     PARAMETERS, ALPHA,AA,AND OMEGA.
C     THE COHERENT PROCESSING GAIN, PER SAMPLE, IS L(2).
C     THE INCOHERENT PROCESSING GAIN, PER SAMPLE, IS A
C     FUNCTION OF L(2), L(4) AND THE SIGNAL PARAMETER QN.
C     THE RATIO OF THE COHERENT AND INCOHERENT PROCESSING
C     GAINS IS ALSO CALCULATED ALONG WITH THE BOUNDS XO AND YO.
      COMMON/QQQ/ALPHA,AA,OMEGA
      DIMENSION IRAY(6),AALPHA(9),AAA(6),OOMEGA(54),QN(2)
      DIMENSION PIINC(2),PIDB(2),RATIO(2),RADB(2),YO(2)
      DIMENSION Z1(7),Z2(7),Z3(7),Z4(7)
      DATA AALPHA/0.2,0.4,0.6,0.8,1.0,1.2,1.4,1.6,1.8/
      DATA AAA/0.001,0.01,0.1,0.5,1.0,2.0/
      DATA OOMEGA/6.551E-4,6.564E-5,6.659E-6,1.419E-6,7.683E-7,
     14.502E-7,2.057E-3,2.062E-4,2.067E-5,4.176E-6,2.115E-6,
     I1.085E-6,8.519E-3,8.588E-4,8.598E-5,1.723E-5,8.631E-6,
     34.333E-6,1.078E-1,4.001E-3,4.016E-4,8.037E-5,4.020E-5,
     42.012E-5,7.396E-1,3.069E-2,1.996E-3,3.998E-4,2.000E-4,
     51.000E-4,9.335E-1,3.080E-1,1.026E-2,2.069E-3,1.036E-3,
     65.182E-4,9.572E-1,7.708E-1,1.121E-1,1.438E-2,5.507E-3,
     72.761E-3,9.618E-1,9.099E-1,4.697E-1,1.119E-1,4.993E-2,
     82.133E-2,9.636E-1,9.456E-1,7.584E-1,3.656E-1,2.255E-1,0.0/
      DATA QN/1.,10./
      DATA IRAY/-1,-1,-1,0,-1,-1/
      CALL SYSTEMC(115,IRAY)
      PRINT 6
    6 FORMAT(1H1)
      DO 80 L=1,9
      ALPHA=AALPHA(L)
      DO 70 L1=1,5
      AA=AAA(L1)
      OMEGA=2.0*OOMEGA(L1+6*(L-1))
C     TO NORMALIZE TO REAL NOISE RMS.
      IF(OMEGA.EQ.0.) GO TO 70
      SUM1=0.$SUM2=0.$SUM3=0.$SUM4=0.
      DO 50 I=1,42
      BDB=-160.+5.*(I-1)
      CDB=-160.+5.*I
      B=10.**(BDB/20.)
      C=10.**(CDB/20.)
      IF(I.EQ.1) B=0.
      DX=(C-B)/6.
      DO 40 J=1,7
      X=B+(J-1)*DX
      CALL FUN(X,Z1(J),Z2(J),Z3(J),Z4(J))
   40 CONTINUE
```

```
      S1=0.3*DX*(Z1(1)+5.*Z1(2)+Z1(3)+6.*Z1(4)+Z1(5)+5.*Z1(6)+Z1(7))
      S2=0.3*DX*(Z2(1)+5.*Z2(2)+Z2(3)+6.*Z2(4)+Z2(5)+5.*Z2(6)+Z2(7))
      S3=0.3*DX*(Z3(1)+5.*Z3(2)+Z3(3)+6.*Z3(4)+Z3(5)+5.*Z3(6)+Z3(7))
      S4=0.3*DX*(Z4(1)+5.*Z4(2)+Z4(3)+6.*Z4(4)+Z4(5)+5.*Z4(6)+Z4(7))
      SUM1=SUM1+S1
      SUM2=SUM2+S2
      SUM3=SUM3+S3
      SUM4=SUM4+S4
   50 CONTINUE
      SUM1=2.*SUM1
C     SUM1 IS L(2)
      SUM2=2.*SUM2
C     SUM2 IS L(4)
      SL4DB=10.*ALOG10(SUM2)
      SUM3=2.*SUM3
C     SUM3 IS L(2,2)
      SL22DB=10.*ALOG10(SUM3)
      SUM4=2.*SUM4
C     SUM4 IS L(6)
      SL6DB=10.*ALOG10(ABS(SUM4))
      SL2=SUM1
      SL2DB=10.*ALOG10(SL2)
      XO=SL2/(SUM3/2.-SL2*SL2)
      DO 60 K=1,2
      F=2.*SL2*SL2/SUM2
      PIINC(K)=(SUM2/8.)*(1.+F*(QN(K)-1.))
      PIDB(K)=10.*ALOG10(PIINC(K))
      RATIO(K)=PIINC(K)/SL2
      RADB(K)=10.*ALOG10(RATIO(K))
   60 CONTINUE
      YO(1)=SUM2/ABS(SUM4/2.+6.*SL2*SUM3)
      YO(2)=1./ABS(3.*SUM3/SL2+2.*SL2)
      PRINT 7, ALPHA, AA, OMEGA
      PRINT 8, SL2, SUM2, SUM3, SUM4
      PRINT 9, XO, YO(1), YO(2)
      PRINT 10, PIINC(1), PIINC(2), RATIO(1), RATIO(2)
      PRINT 11, SL2DB, PIDB(1), PIDB(2), RADB(1), RADB(2)
      PRINT 12, SL4DB, SL22DB, SL6DB
   70 CONTINUE
   80 CONTINUE
    7 FORMAT(2X,3(1PE10.3,3X))
    8 FORMAT(5X,4(1PE10.3,2X))
    9 FORMAT(5X,3(1PE10.3,2X))
   10 FORMAT(5X,4(1PE10.3,2X))
   11 FORMAT(5X,5(1PE10.3,2X))
   12 FORMAT(5X,3(1PE10.3,2X),/)
      END
```

```
      SUBROUTINE FUN(X,Y,YY,YYY,YYYY)
C     THIS SUBROUTINE COMPUTES THE L(2)(Y),THE L(4)(YY),
C     THE L(2,2)(YYY),AND THE L(6)(YYYY) INTEGRANDS FOR
C     CLASS B NOISE.
      COMMON/QQQ/ ALPHA,AA,OMEGA
      PI=3.141592654
      ZN=X*X/OMEGA
      FM=1.
      SUM1=SUM2=SUM3=0.
      SUM4=SUM5=SUM6=0.
      DO 20 M=1,26
      FM=FM*M
      CALL CONHYP(-M*ALPHA/2.,0.5,ZN,S,IOVFLW)
      CALL CONHYP(-M*ALPHA/2.,1.5,ZN,SS,IOVLLW)
      CALL CONHYP(-M*ALPHA/2.,2.5,ZN,SSS,IOVFLW)
      T=(((-AA)**M)/FM)*GAMMA(0.5+M*ALPHA/2.)
      T1=T*S
      T2=T*(1.+M*ALPHA)*SS
      T3=T*(1.+M*ALPHA)*(1.+M*ALPHA/3.)*SSS
      IF(IOVFLW.NE.1) GO TO 15
      SUM4=SUM4+T1
      SUM5=SUM5+T2
      SUM6=SUM6+T3
      GO TO 20
   15 SUM1=SUM1+T1
      SUM2=SUM2+T2
      SUM3=SUM3+T3
   20 CONTINUE
      FP=0.
      IF(ZN.LT.675.) FP=EXP(-ZN)
      PT=FP*(SUM1+SQRT(PI))+SUM4
      P1T=FP*(SUM2+SQRT(PI))+SUM5
      P11T=FP*(SUM3+SQRT(PI))+SUM6
      P=PT/(PI*SQRT(OMEGA))
      P1=-2.*X*P1T/(PI*OMEGA**1.5)
      P11=4.*X*X*P11T/(PI*OMEGA**2.5)-2.*P1T/(PI*OMEGA**1.5)
C     P IS PDF OF X
C     P1 IS PDF PRIME
C     P11 IS PDF PRIME PRIME (2ND DERIVITIVE)
      Y=P1*P1/P
      YY=P11*P11/P
      YYY=(2.*P1**4.)/(P**3.)
      YYYY=(P11**3.)/(P*P)
      RETURN
      END
```

```
      PROGRAM PDVPDS(INPUT,OUTPUT)
C     PROGRAM COMPUTES THE CANONICAL PERFORMANCE RESULTS,EQUATION
C     6.50, PROBABILITY OF DETECTION VERSUS OPTIMUM PROBABILITY
C     OF DETECTION, AS FUNCTION OF DEGRADATION FACTOR PHID,
C     FOR A GIVEN FALSE ALARM PROBILITY, ALPHAF.
      DIMENSION ALPHA(2), PHI(7), PDSS(18)
      DATA ALPHA/1.0E-3,1.0E-6/
      DATA PHI/1.,0.5,0.1,0.05,0.01,0.005,0.001/
      DATA PDSS/1.0E-6,5.0E-6,1.0E-5,5.0E-5,1.0E-4,5.0E-4,0.001,
     10.005,0.01,0.05,.1,.5,.6,.7,.8,.9,.95,.98/
      PRINT 6
    6 FORMAT(1H1)
      DO 60 I=1,2
      ALPHAF=ALPHA(I)
      DO 50 J=1,7
      PHID=PHI(J)
      PRINT 7, ALPHAF, PHID
      DO 40 K=1,18
      PDS=PDSS(K)
      T1=ERFIN(1.-2.*ALPHAF)
      T2=ERFIN(ABS(2.*PDS-1.))
      IF(2.*PDS-1..LE.0.) T2=-T2
      T3=SQRT(PHID)*(T2+T1)
      T=T3-T1
      IF(T.LE.0.) GO TO 30
      PD=1.0-0.5*CERF(T)
      GO TO 35
   30 T=-T
      PD=0.5*CERF(T)
   35 PRINT 8, PDS, PD
   40 CONTINUE
      PRINT 9
   50 CONTINUE
   60 CONTINUE
    7 FORMAT(5X,2(1PE10.3,3X),/)
    8 FORMAT(8X,2(1PE10.3,3X))
    9 FORMAT(//)
      END
```

```
      PROGRAM PEVPES(INPUT,OUTPUT)
C     PROGRAM COMPUTES THE CANONICAL PERFORMANCE RESULTS EQUATION
C     6.51, PROBABILITY OF BINARY BIT ERROR VERSUS OPTIMUM
C     PROBABILITY OF ERROR, AS A FUNCTION OF DEGRADATION FACTOR PHID.
      DIMENSION PHI(10),PESS(17)
      DATA PHI/1.0,.9,.8,.7,.6,.5,.4,.3,.2,.1/
      DATA PESS/1.E-6,2.E-6,5.E-6,1.E-5,2.E-5,5.E-5,1.E-4,2.E-4,
     15.E-4,.001,.002,.005,.01,.02,.05,.1,.2/
      PRINT 6
    6 FORMAT(1H1)
      DO 60 I=1,10
      PHID=PHI(I)
      PRINT 7, PHID
      DO 40 J=1,17
      PES=PESS(J)
      PE=0.5*CERF(PHID*ERFIN(1.-2.*PES))
      PRINT 8, PES, PE
   40 CONTINUE
      PRINT 9
   60 CONTINUE
    7 FORMAT(5X,1PE10.3,/)
    8 FORMAT(8X,2(1PE10.3,3X))
    9 FORMAT(//)
      END

      PROGRAM WOA(INPUT,OUTPUT)
C     PROGRAM COMPUTES THE PDF EVALUATED AT ZERO
C     FOR CLASS A NOISE.
      DIMENSION AA(10),G(7)
      DATA AA/1.E-4,1.E-2,0.1,0.5,1.0,2.,3.,4.,5.,10./
      DATA G/1.E-8,1.E-7,1.E-6,1.E-5,1.E-4,1.E-3,1.E-2/
      PI=3.141592654
      PRINT 6
    6 FORMAT(1H1)
      DO 80 J=1,10
      A=AA(J)
       PRINT 7, A
      DO 70 K=1,7
      GAM=G(K)
      SUM=0. $ FN=1.
      DO 60 NN=1,30
      N=NN-1
      IF(N.NE.0) FN=FN*N
      T=((A**N)/FN)*SQRT((1.+GAM)/(N/A+GAM))
      SUM=SUM+T
   60 CONTINUE
      WO=EXP(-A)*SUM/SQRT(2.*PI)
      ARE=4.*WO*WO
      AREDB=10.*ALOG10(ARE)
      PRINT 8, GAM,WO,ARE,AREDB
   70 CONTINUE
      PRINT 9
   80 CONTINUE
    7 FORMAT(5X,1PE10.3,/)
    8 FORMAT(8X,4(1PE10.3,3X))
    9 FORMAT(//)
      END
```

```
      PROGRAM WOB(INPUT,OUTPUT)
C     PROGRAM COMPUTES THE PDF EVALUATED AT ZERO
C     FOR CLASS B NOISE.
      DIMENSION AALPHA(9),AAA(6),OOMEGA(54)
      PI=3.141592654
      DATA AALPHA/0.2,0.4,0.6,0.8,1.0,1.2,1.4,1.6,1.8/
      DATA AAA/0.001,0.01,0.1,0.5,1.0,2.0/
      DATA OOMEGA/6.551E-4,6.564E-5,6.659E-6,1.419E-6,7.683E-7,
     14.502E-7,2.057E-3,2.062E-4,2.067E-5,4.176E-6,2.115E-6,
     I1.085E-6,8.519E-3,8.588E-4,8.598E-5,1.723E-5,8.631E-6,
     34.333E-6,1.078E-1,4.001E-3,4.016E-4,8.037E-5,4.020E-5,
     42.012E-5,7.396E-1,3.069E-2,1.996E-3,3.998E-4,2.000E-4,
     51.000E-4,9.335E-1,3.080E-1,1.026E-2,2.069E-3,1.036E-3,
     65.182E-4,9.572E-1,7.708E-1,1.121E-1,1.438E-2,5.507E-3,
     72.761E-3,9.618E-1,9.099E-1,4.697E-1,1.119E-1,4.993E-2,
     82.133E-2,9.636E-1,9.456E-1,7.584E-1,3.656E-1,2.255E-1,0.0/
      PRINT 6
    6 FORMAT(1H1)
      DO 80 J=1,5
      AA=AAA(J)
      PRINT 7, AA
      DO 70 K=1,9
      ALPHA=AALPHA(K)
      OMEGA=2.0*OOMEGA(6*(K-1)+J)
C     TO NORMALIZE TO REAL NOISE RMS.
      SUM=0. $ FN=1.
      DO 60 NN=1,30
      N=NN-1
      IF(N.NE.0) FN=FN*N
      T=(((-AA)**N)/FN)*GAMMA(N*ALPHA/2.+0.5)
      SUM=SUM+T
   60 CONTINUE
      WO=SUM/(PI*SQRT(OMEGA))
      ARE=4.*WO*WO
      AREDB=10.*ALOG10(ARE)
      PRINT 8, ALPHA,OMEGA,WO,ARE,AREDB
   70 CONTINUE
      PRINT 9
   80 CONTINUE
    7 FORMAT(5X,1PE10.3,/)
    8 FORMAT(8X,5(1PE10.3,3X))
    9 FORMAT(//)
      END
```

FORM **NTIA-29**
(4-80)

U.S. DEPARTMENT OF COMMERCE
NAT'L. TELECOMMUNICATIONS AND INFORMATION ADMINISTRATION

BIBLIOGRAPHIC DATA SHEET

	1. PUBLICATION NO.	2. Gov't Accession No.	3. Recipient's Accession No.
	NTIA Report 83-120		

4. TITLE AND SUBTITLE
Optimum reception in nongaussian electromagnetic interference environments: II. Optimum and suboptimum threshold signal detection in Class A and B noise

5. Publication Date
May 1983

6. Performing Organization Code
NTIA/ITS.S3

7. AUTHOR(S)
David Middleton and A. D. Spaulding

9. Project/Task/Work Unit No.
910 1518

8. PERFORMING ORGANIZATION NAME AND ADDRESS
NTIA/ITS.S3
325 Broadway
Boulder, CO 80303

10. Contract/Grant No.

11. Sponsoring Organization Name and Address
NTIA/DoD
U.S. DEPARTMENT OF COMMERCE BLDG.
Washington, DC

12. Type of Report and Period Covered

13.

14. SUPPLEMENTARY NOTES

15. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

The general theory of optimum and suboptimum detection of threshold (i.e., weak) signals in highly nongaussian interference environments is developed. Both signal processing algorithms and performance measures are obtained canonically, and specifically for both narrowband and broadband interference environments. Suboptimum systems are also treated and a number of numerical examples are included which illustrate the determination of performance and performance comparisons. An extensive set of appendices contains many of the analytic details developed and presented here as well as the required computational algorithms.

16. Key Words *(Alphabetical order, separated by semicolons)*

Threshold signal detection; optimum threshold detection algorithms; performance measures; performance comparisons; electromagnetic interference environments (EMI); suboptimum detectors; locally optimum and asymptotically optimum algorithms; Class A, B noise; correlation detectors; clipper-correlators; error probabilities; minimum detectable signals; processing gain; bias, EMI scenarios; composite threshold detection algorithms; on-off binary signal detection; non-gaussian noise and interference.

17. AVAILABILITY STATEMENT	18. Security Class. *(This report)*	20. Number of pages
☒ X UNLIMITED.	UNCLASSIFIED	346
☐ FOR OFFICIAL DISTRIBUTION.	19. Security Class. *(This page)* UNCLASSIFIED	21. Price: