Förkortning av gemensamma faktorer i skattade överföringsfunktioner - kort beskrivning av programvaran

Burström, Erik

1973

Document Version:
Förlagets slutgiltiga version

Link to publication

FÖRKORTNING AV GEMENSAMMA FAKTORI SKATTADE ÖVERFÖRINGSFUNKTIONER

KORT BESKRIVNING AV PROGRAMVARAN

E. BURSTRÖM

Rapport 7320(C) Juli 1973
Inst. för Reglerteknik
Lunds Tekniska Högskola

FÖRKORTNING AV GEMENSAMMA FAKTORER I SKATTADE ÖVERFÖRINGSFUNKTIONER.

KORT BESKRIVNING AV PROGRAMVARAN.

E. Burström

INNEHÅLLSFÖRTECKNING

1. INLEDNING.

Problemet att avgöra om skattade överföringsfunktioner har gemensamma faktorer i statistisk mening har undersökts med två olika metoder.

Den första metoden går ut på att beräkna systemets poler och nollställen och en tillhörande kovariansmatris, och därefter görs statistiska hypotestester på så sätt att man testar om vissa kombinationer av poler och nollställen kan anses vara gemensamma. Metoden kallas i fortsättningen TPOL.

Den andra metoden använder Euklides algoritm för polynom, d.v.s. de givna polynomen divideras med varandra till dess att restpolynomet är noll i statistisk mening.

Förutsättningar och analytisk behandling finns beskrivna i [1]. I fortsättningen ges endast en kort beskrivning av programvaran, som realiserar de två metoderna.

Exempel på resultat vid exekvering av programmen ges i [1].

2. PROGRAMVARAN FÖR TPOL.

Problemet löses i det här fallet enl. nedanstående figur.

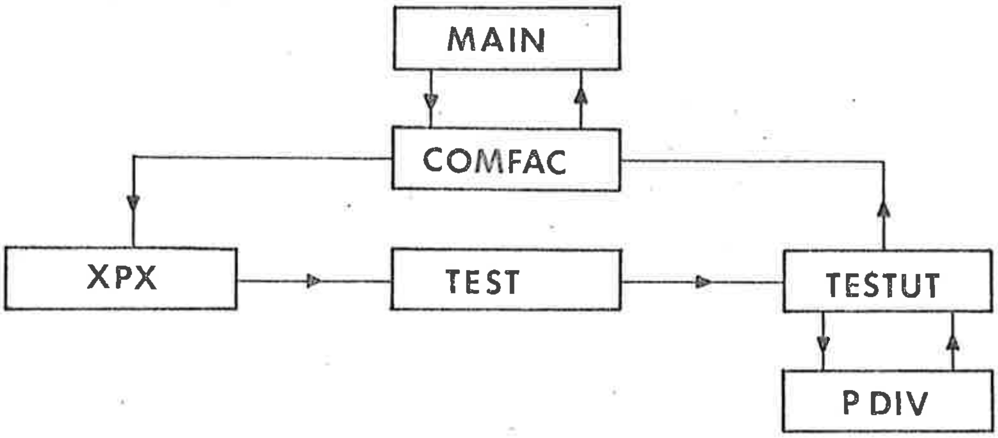

Fig. 2.1.

MAIN. Användarens eget huvudprogram. Här bildas vektorn T, kovariansmatrisen P_T och avgörs om B- eller C-fallet är för handen.

COMFAC. Den överordnade subrutinen som administrerar hela analysen.

XPX. Här beräknas dels vektorn X, vars komponenter utgörs av överföringsfunktionens poler och nollställen, dels matrisen PX, kovariansmatrisen för X. Komplexa rötter separeras i real- och imaginärdel i vektorn X.

TEST. Här görs individuella hypotestester. Varje pol testas individuellt mot varje nollställe, och resultatet, d.v.s. de tester som accepteras, ges i en tabell IQ.

TESTUT. Här görs flervariabla tester. På grundval av resultaten i subrutinen TEST, bestämmes den kombination av så stort antal poler och nollställen som möjligt, som ger en positiv testkvantitet. Om mer än en kombination av samma antal faktorer är möjlig väljs den, som ger lägst testkvantitet. Slutligen beräknas ett nytt estimat av vektorn X under förutsättning att vissa faktorer är gemensamma, d.v.s. lika.

PDIV. Här divideras de givna polynomen med det gemensamma polynomet, som beräknas ur det nya estimatet av vektorn X, som beräknats i subrutinen TESTUT.

En fullständig programlista återfinns i appendix.

3. PROGRAMVARAN FÖR EUKLIDES ALGORITM.

Problemet löses här enligt figur nedan.

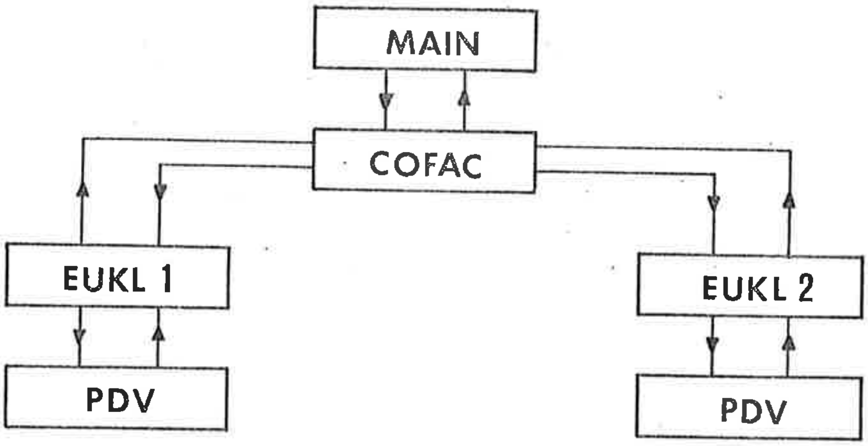

Fig. 3.1.

MAIN. Användarens eget huvudprogram där vektorn T + kovariansmatrisen P_T bildas och avgörs om B- eller C-fallet föreligger.

COFAC. Den överordnade subrutinen. Här modifieras T och P_T för att den fortsatta behandlingen skall bli enhetligare. Därefter anropas EUKL1 och EUKL2, som använder Euklides algoritm i de båda varianterna, se [1].

EUKL1. Här används Euklides algoritm i version 1, vilket innebär att vid varje division skalas de båda ingående polynomen så att högstagradskoefficienterna blir 1. Restpolynomet som erhålles vid de successiva divisionerna testas om det är noll i statistisk mening eller inte. Om så är fallet anropas subrutinen PDV, se nedan, annars fortsätter divisionen på känt sätt.

EUKL2. Här används Euklides algoritm i version 2. Den enda skillnaden mot version 1 är att vid de successiva divisionerna skalas inte polynomen utan divisionerna sker på vanligt sätt, s.k. icke-normaliserad division.

PDV. Här divideras de gemensamma faktorerna bort från de givna polynomen. Divisionen sker på precis samma sätt som i subrutinen PDIV, se kapitel 2.

En fullständig lista återfinns i appendix.

4. REFERENSER.

[1] E. Burström: Förkortning av gemensamma faktorer i skattade överföringsfunktioner. Examensarbete RE 124, 1973, Inst. för Reglerteknik, LTH, Lund.

```
SUBROUTINE COMFAC(T,PT,AM,BM,TT,NA,NB,NMA,NMB,IPR,IERR,IA,IB)

THE SUBROUTINE COMPUTES COMMON FACTORS TO TWO GIVEN POLYNOMIALS, AND
THE GIVEN POLYNOMIALS ARE ABBREVIATED WITH THE COMMON POLYNOMIAL.
AN HYPOTHESIS TEST IS USED IN ORDER TO DECIDE WHETHER THE TWO GIVEN
POLYNOMIALS HAVE COMMON FACTORS OR NOT. THE COEFFICIENTS OF THE TWO
GIVEN POLYNOMIALS SHOULD BE NORMALLY DISTRIBUTED FOR OBTAINING
BEST RESULTS, BECAUSE A CHI-SQUARE TEST WITH THE SIGNIFICANCE LEVEL
OF 5% IS USED.
AUTHOR ERIK BURSTRÖM 1972-12-24

T=(A(1),...,A(NA),B(1),...,B(NB)) VECTOR OF ORDER (NA+NB), (MIN 2, MAX 20
CONTAINING THE COEFFICIENTS OF THE A- RESP. B-POLYNOMIAL.
PT MATRIX OF ORDER (NA+NB)*(NA+NB) THE COVARIANCE MATRIX OF T.
AM VECTOR OF ORDER NMA AT OUTPUT CONTAINING THE NEW ESTIMATED A-COEFFI-
CIENTS.
BM VECTOR OF ORDER NMB AT OUTPUT CONTAINING THE NEW ESTIMATRD B-COEFFI-
CIENTS.
TT STASTISTICAL TEST QUANTITY.
NA NUMBER OF A-COEFFICIENTS (MIN 1, MAX 10).
NB NUMBER OF B-COEFFICIENTS (MIN 1, MAX 10).
NMA NUMBER OF AM-COEFFICIENTS.
NMB NUMBER OF BM-COEFFICIENTS.
IPR  IF IPR=0 NOTHING IS PRINTED.
     IF IPR=1 VECTOR T, POLES, ZEROS AND STATIC GAIN FOR THE ORIGINAL
      SYSTEM, VECTOR X CONSISTING OF POLES AND ZEROS, ALL POSITIVE
      INDIVIDUELL TESTS WHEN TWO FACTORS ARE EQUAL AND CORRESPONDING
      TEST QUANTITIES, MATRIX IQ WHERE EACH ROW MEANS A POSSIBLE
      COMBINATION FOR TWP FACTORS TO BE EQUAL, THE PART OF MATRIX IQ
      THAT REPRESENTS THE BEST MULTIVARIATE TEST, I.E. THE HIGHEST
      NUMBER OF DEGREES OF FREEDOM AND WITHIN THAT NUMBER THE TEST
      WITH SMALLEST TEST QUANTITY, A NEW ESTIMATED VECTOR X, THE NEW A-
      POLYNOMIAL, DEGREE OF A AND POLES, THE NEW B-POLYNOMIAL, DEGREE
      OF B AND ZEROS AND THE STATIC GAIN FOR THE NEW SYSTEM ARE PRINTED.
     IF IPR=2 AS IPR=1 + THE COVARIANCE MATRIX OF T AND THE APPROXIMATE
      COVARIANCE MATRIX OF X, THE COMMON POLYNOMIAL AND THE TWO REST-
      POLYNOMIALS ARE PRINTED.
     IF IPR=3 AS IPR=2 + ALL NEGATIVE INDIVIDUELL TESTS AND ALL MULTI-
      VARIATE TESTS, I.E. THE COMBINATIONS OF FACTORS AND TEST QUANTITIES
      WITH HIGHER AND THE SAME DEGREE OF  FREEDOM AS THE BEST POSITIVE
      TEST ARE PRINTED.
     IF IPR=4 AS IPR=3 + ALL MULTIVARIATE TESTS WITH LOWER DEGREES OF
      FREEDOM THAN THE BEST POSITIVE TEST ARE PRINTED.
IERR IF IERR=0 NORMAL OUTPUT.
     IF IERR=-1 EITHER THE COVARIANCE MATRIX OF X IS NOT POSITIVE DEFINIT
     OR IS IT IMPOSSIBLE TO COMPUTE POLES OR ZEROS TO THE ORIGINAL
     OR THE NEW SYSTEM. IN THE FIRST CASE, THE INTERESTING EIGNVALUES
     AND CORRESPONDING EIGENVECTORS OF PT ARE COMPUTED AND PRINTED.
IA DIMENSION PARAMETER OF PT.
IB IF IB=0 B(0)=0 IS ASSUMED.
   IF IB=1 B(0)=1 IS ASSUMED.

THE FOLLOWING VARIABLE LIES IN A COMMON-BLOCK CALLED COMPOL.

D VECTOR OF ORDER 10 CONTAINING THE COEFFICIENTS OF THE COMMON
POLYNOMIAL.

SUBROUTINE REQUIRED
     XPX
     TEST
     TESTUT
     PDIV
     ROT
```

```
      DESYM
      SOLVS
      EIGS
      PMPY

   DIMENSION T(1),PT(IA,IA),AM(1),BM(1)
   COMMON/COMPOL/D(10)

   DIMENSION X(20),PX(20,20),IQ(100,4)

   IERR=0
   N=NA+NB
   IC=100
   IF(IPR.LE.0)GO TO 5
   PRINT 101
101 FORMAT(1H1,9X,'PRINTOUT FROM COMFAC'/10X,20(1H*))
   PRINT 103
103 FORMAT(/10X,'VECTOR T')
   PRINT 102,(T(I),I=1,N)
102 FORMAT(8X,10G12.5)
  5 CONTINUE

   COMPUTE POLES, ZEROS, STATIC GAIN AND COVARIANCE MATRIX PX FOR THE
   ORIGINAL SYSTEM.

   CALL XPX(T,PT,X,PX,NA,NB,NCA,NCB,IPR,IERR,IA,IB)
   IF(IERR.EQ.-1)RETURN

   COMPUTE ALL INDIVIDUELL TESTS AND MATRIX IQ

   CALL TEST(X,PX,IQ,NA,NB,NCA,NCB,NQ,IPR,IERR,IA,IB,IC)
   IF(IERR.EQ.-1)RETURN

   COMPUTE MULTIVARIATE TESTS, THE NEW A- AND B-POLYNOMIALS AND STATIC
   GAIN FOR THE NEW SYSTEM

   CALL TESTUT(T,X,PX,AM,BM,TT,IQ,NA,NB,NMA,NMB,NQ,IPR,IERR,IA,IB,IC)
   RETURN
   END
```

```
      SUBROUTINE XPX(T,PT,X,PX,NA,NB,NCA,NCB,IPR,IERR,IA,IB)

      SUBROUTINE FOR TRANSFORMING T-VECTOR TO X-VECTOR AND COVARIANCE MATRIX
      PT TO THE APPROXIMATE COVARIANCE MATRIX PX OF VECTOR X.
      NO MULTIPLE POLES OR ZEROS IS ASSUMED.
      THE ELEMENTS OF VECTOR T MUST BE REAL.
      COMPLEX FACTORS ARE SEPARATED INTO REAL AND IMAGINARY PARTS.
      AUTHOR TORSTEN SÖDERSTRÖM 1971-12-24
      REVISED ERIK BURSTRÖM 1972-11-01

      T=(A(1),...,A(NA),B(1),...,B(NB)) VECTOR OF ORDER (NA+NB).
      PT MATRIX OF ORDER (NA+NB)*(NA+NB), THE COVARIANCE MATRIX OF T.
      X VECTOR OF ORDER (NA+NB). THE ELEMENTS OF X ARE NCA/2 REAL PARTS OF
      COMPLEX POLES, NCA/2 POSITIVE IMAGINARY PARTS OF COMPLEX POLES, NA-NCA
      REAL POLES, NCB/2 REAL PARTS OF COMPLEX ZEROS, NCB/2 POSITIVE IMAGINARY
      PARTS OF COMPLEX ZEROS, NB-NCB REAL ZEROS. IF IB=0 X(NA+NB)=T(NA+1).
      PX MATRIX OF ORDER (NA+NB)*(NA+NB) AT RETURN CONTAINING THE APPROXIMATE
      COVARIANCE MATRIX OF X.
      NA NUMBER OF A-COEFFICIENTS (MIN 1, MAX 10).
      NB NUMBER OF B-COEFFICIENTS (MIN 1, MAX 10).
      NCA NUMBER OF COMPLEXVALUED POLES (MIN 0, MAX 10).
      NCB NUMBER OF COMPLEXVALUED ZEROS (MIN 0, MAX 10).
      IPR  IF IPR=0 NOTHING IS PRINTED.
           IF IPR=1 VECTOR T, THE POLES, THE NUMBER OF A-COEFFICIENTS, THE
            NUMBER OF COMPLEX RESP. REAL POLES,THE ZEROS, THE NUMBER OF
            B-COEFFICIENTS, THE NUMBER OF COMPLEX RESP. REAL ZEROS, VECTOR X
            AND STATIC GAIN ARE PRINTED.
           IF IPR=2 AS IPR=1 + THE COVARIANCE MATRIX OF X.
      IERR IF IERR =0 NORMAL RETURN
           IF IERR=-1 ROT HAS FAILED.
      IA DIMENSION PARAMETER OF PT AND PX.
      IB IF IB=0 B(0)=0 IS ASSUMED.
         IF IB=1 B(0)=1 IS ASSUMED.

      SUBROUTINE REQUIRED
           ROT

      DIMENSION T(1),PT(IA,IA),X(1),PX(IA,IA)

      DIMENSION A(10),AR(10),AI(10),B(10),BR(10),BI(10),DXDT(20,20),
     *S(20,20),LINE(22)
      COMPLEX Z(10),ZZ,ZM(10,10)
      DATA STRECK/5H-----/

      N=NA+NB
      NERRA=0
      NERRB0=0
      NERRB1=0
      IERR=0
      IF(IB.EQ.0)G1=T(NA+1)
      IF(IB.EQ.1)G1=1.
      G2=1.
      G=0.
      EPS=1.0E-05
      KCA=0
      KCB=0

      COMPUTE THE POLES, ZEROS AND STATIC GAIN.

      DO 1 I=1,NA
      G2=G2+T(I)
    1 A(I)=T(I)
      CALL ROT(NA,1,A,A,AR,AI,NERRA)
```

```
      IF (NERRA.GT.0) GO TO 8
      DO 54 I=1,NA
   54 IF(ABS(AI(I)).GT.EPS)NCA=NCA+1
      IF (IB.EQ.1) GO TO 51
      NN=NB-1
      IF (NB.EQ.1) GO TO 3
      DO 2 I=1,NN
      G1=G1+T(I+NA+1)
    2 B(I)=T(I+NA+1)/T(NA+1)
      CALL ROT (NN,1,B,B,BR,BI,NERRB0)
      IF (NERRB0.GT.0) GO TO 8
      DO 55 I=1,NN
   55 IF(ABS(BI(I)).GT.EPS)NCB=NCB+1
    3 AK=T(NA+1)
      IF(IPR.EQ.0) GO TO 49
      PRINT 101,AK
  101 FORMAT(/56X,'B(0)=0.',3X,'B(1)= ',G10.5/10X,'ROOTS OF A',36X,'ROOT
     *S OF B'/)
      GO TO 53
   51 DO 52 I=1,NB
      G1=G1+T(I+NA)
   52 B(I)=T(I+NA)
      NN=NB
      CALL ROT (NN,1,B,B,BR,BI,NERRB1)
      IF (NERRB1.GT.0) GO TO 8
      DO 56 I=1,NB
   56 IF(ABS(BI(I)).GT.EPS)NCB=NCB+1
      IF(IPR.EQ.0) GO TO 49
      PRINT 201
  201 FORMAT(/56X,'B(0)=1.'/10X,'ROOTS OF A',36X,'ROOTS OF B'/)
   53 CONTINUE
      N0=MIN0(NA,NN)
      NM=N0+1
      NL=MAX0(NA,NN)
      IF (IB.EQ.0.AND.NB.EQ.1) GO TO 5
      DO 4 I=1,N0
    4 PRINT 102,AR(I),AI(I),BR(I),BI(I)
  102 FORMAT(10X,2G15.7,16X,2G15.7)
    5 IF (NA.EQ.NN) GO TO 7
      IF (NA.GT.NN) GO TO 6
      PRINT 108,((BR(I),BI(I)),I=NM,NL)
  108 FORMAT(56X,2G15.7)
      GO TO 7
    6 PRINT 109,((AR(I),AI(I)),I=NM,NL)
  109 FORMAT(10X,2G15.7)
    7 CONTINUE
      NRA=NA-NCA
      NRB=NN-NCB
      PRINT 103,NA,NN,NCA,NCB,NRA,NRB
  103 FORMAT(/10X,I2,3X,'A-COEFFICIENTS',27X,I2,3X,'B-COEFFICIENTS'/10X,
     *I2,3X,'COMPLEX POLES',28X,I2,3X,'COMPLEX ZEROES'/10X,I2,3X,'REAL P
     *OLES',31X,I2,3X,'REAL ZEROES')
      G=G1/G2
      PRINT 104,G
  104 FORMAT(/10X,'STATIC GAIN',5X,G10.5)
   49 CONTINUE

      COMPUTATION OF THE JACOBIAN.

      DO 10 I=1,NA
      DO 10 J=1,NB
      DXDT(I,J+NA)=0.
   10 DXDT(J+NA,I)=0.
      DO 11 I=1,NA
```

```
      DO 11 J=1,NA
   11 ZM(I,J)=(0.,0.)
      DO 12 I=1,NA
   12 Z(I)=CMPLX(AR(I),AI(I))

      DO 15 I=1,NA
      ZZ=(1.,0.)
      DO 13 J=1,NA
      IF (J.EQ.I) GO TO 13
      ZZ=ZZ*(Z(I)-Z(J))
   13 CONTINUE
      DO 14 J=1,NA
   14 ZM(I,J)=-Z(I)**(NA-J)/ZZ
   15 CONTINUE
      IF (NCA.EQ.0) GO TO 17
      NC=NCA/2
      DO 16 I=1,NC
      DO 16 J=1,NA
      DXDT(2*I-1,J)=(ZM(2*I-1,J)+ZM(2*I,J))/2.
   16 DXDT(2*I  ,J)=(ZM(2*I-1,J)-ZM(2*I,J))/2./(0.,1.)
   17 IF (NCA.EQ.NA) GO TO 19
      I1=NCA+1
      DO 18 I=I1,NA
      DO 18 J=1,NA
   18 DXDT(I,J)=ZM(I,J)
   19 CONTINUE
      IF (IB.EQ.1) GO TO 60

      IF (NB.EQ.1) GO TO 30
      DO 20 I=1,NN
   20 Z(I)=CMPLX(BR(I),BI(I))
      DO 25 I=1,NN
      ZZ=(1.,0.)
      DO 21 J=1,NN
      IF (J.EQ.I) GO TO 21
      ZZ=ZZ*(Z(I)-Z(J))
   21 CONTINUE
      DO 22 J=1,NB
   22 ZM(I,J)=-Z(I)**(NN-J)/ZZ/AK
   25 CONTINUE
      DO 26 I=1,NN
      ZM(I,1)=(0.,0.)
      ZM(NB,I+1)=(0.,0.)
      DO 26 J=1,NN
   26 ZM(I,1)=ZM(I,1)-T(NA+1+J)/T(NA+1)*ZM(I,J+1)
      ZM(NB,1)=(1.,0.)
      IF (NCB.EQ.0) GO TO 28
      NC=NCB/2
      DO 27 I=1,NC
      DO 27 J=1,NB
      DXDT(NA+2*I-1,J+NA)=(ZM(2*I-1,J)+ZM(2*I,J))/2.
   27 DXDT(NA+2*I  ,J+NA)=(ZM(2*I-1,J)-ZM(2*I,J))/2./(0.,1.)
   28 IF (NCB.EQ.NN) GO TO 30
      I1=NCB+1
      DO 29 I=I1,NB
      DO 29 J=1,NB
   29 DXDT(I+NA,J+NA)=ZM(I,J)
   30 DXDT(N,N)=1.
      GO TO 31

   60 DO 62 I=1,NB
   62 Z(I)=CMPLX(BR(I),BI(I))
      DO 65 I=1,NB
      ZZ=(1.,0.)
```

```
    DO 63 J=1,NB
    IF (J.EQ.I) GO TO 63
    ZZ=ZZ*(Z(I)-Z(J))
 63 CONTINUE
    DO 64 J=1,NB
 64 ZM(I,J)=-Z(I)**(NB-J)/ZZ
 65 CONTINUE
    IF (NCB.EQ.0) GO TO 67
    NBC=NCB/2
    DO 66 I=1,NBC
    DO 66 J=1,NB
    DXDT(2*I-1+NA,J+NA)=(ZM(2*I-1,J)+ZM(2*I,J))/2.
 66 DXDT(2*I+NA,J+NA)=(ZM(2*I-1,J)-ZM(2*I,J))/2./(0.,1.)
 67 IF (NCB.EQ.NB) GO TO 69
    I1=NCB+1
    DO 68 I=I1,NB
    DO 68 J=1,NB
 68 DXDT(I+NA,J+NA)=ZM(I,J)
 69 CONTINUE
 31 CONTINUE

    THE COMPUTATION OF X AND PX.

    DO 32 I=1,N
    DO 32 J=1,N
    S(I,J)=0.
    DO 32 K=1,N
 32 S(I,J)=PI(I,K)*DXDT(K,J)+S(I,J)
    DO 33 I=1,N
    DO 33 J=1,N
    PX(I,J)=0.
    DO 33 K=1,N
 33 PX(I,J)=PX(I,J)+DXDT(K,I)*S(K,J)
    IF (NCA.EQ.0) GO TO 35
    NC=NCA/2
    DO 34 I=1,NC
    X(2*I-1)=AR(2*I-1)
 34 X(2*I  )=AI(2*I-1)
 35 IF (NCA.EQ.NA) GO TO 37
    NP=NCA+1
    DO 36 I=NP,NA
 36 X(I)=AR(I)
 37 IF (NCB.EQ.0) GO TO 39
    NC=NCB
    DO 38 I=1,NC
    X(NA+2*I-1)=BR(2*I-1)
 38 X(NA+2*I  )=BI(2*I-1)
 39 IF (NCB.EQ.NN) GO TO 41
    NP=NCB+1
    DO 40 I=NP,NN
 40 X(I+NA)=BR(I)
 41 CONTINUE
    IF(IB.EQ.0) X(N)=AK

    PRINT RESULTS.

    IF(IPR.EQ.0) GO TO 9
    PRINT 105
105 FORMAT(/10X,'VECTOR X'/10X,8(1H-)/10X,'COMPONENT NUMBER')
    IF(IB.EQ.0) N=N-1
    IF(N.GT.10) GO TO 311
    PRINT 130,(I,I=1,N)
130 FORMAT(2X,10I12)
    CALL ENCODE(LINE)
```

```
      CALL FMTX(11)
      DO 302 I=1,N
      CALL FMTA(STRECK,5)
  302 CALL FMTX(7)
      CALL FMTX(121-N*12)
      PRINT 131,(LINE(I),I=1,22)
  131 FORMAT(22A6)
      PRINT 106,(X(I),I=1,N)
  106 FORMAT(8X,10G12.5)
      GO TO 314
  311 CONTINUE
      PRINT 130,(I,I=1,10)
      CALL ENCODE(LINE)
      CALL FMTX(11)
      DO 312 I=1,10
      CALL FMTA(STRECK,5)
  312 CALL FMTX(7)
      CALL FMTX(1)
      PRINT 131,(LINE(I),I=1,22)
      PRINT 106,(X(I),I=1,10)
      PRINT 130,(I,I=11,N)
      CALL ENCODE(LINE)
      CALL FMTX(11)
      DO 313 I=11,N
      CALL FMTA(STRECK,5)
  313 CALL FMTX(7)
      CALL FMTX(121-(N-10)*12)
      PRINT 131,(LINE(I),I=1,22)
      PRINT 106,(X(I),I=11,N)
  314 IF(IPR.LT.2)GO TO 9
      PRINT 212
  212 FORMAT(/10X,'COVARIANCE MATRIX')
      DO 214 I=1,N
  214 PRINT 106,(PX(I,J),J=1,N)
      GO TO 9
    8 IF(NERRA.GT.0)PRINT 210
  210 FORMAT(/10X,'COMPUTATION OF POLES IS IMPOSSIBLE')
      IF(NERRB0.GT.0)PRINT 211
      IF(NERRB1.GT.0)PRINT 211
  211 FORMAT(/10X,'COMPUTATION OF ZEROS IS IMPOSSIBLE')
      IERR=-1
    9 CONTINUE

      RETURN
      END
```

```
      SUBROUTINE TEST(X,PX,IQ,NA,NB,NCA,NCB,NQ,IPR,IERR,IA,IB,IC)

      SUBROUTINE FOR EXAMINING WHICH POLES AND WHICH ZEROS THAT CAN BE CON-
      SIDERED TO BE EQUAL IN STASTISTICAL SENSE.
      EVERY POLE IS TESTED AGAINST EVERY ZERO.
      THE TESTS ARE BASED ON THE ASSUMPTION THAT THE POLES AND THE ZEROS ARE
      NORMAL DISTRIBUTED I.E. A CHI-SQUARE TEST WITH THE SIGNIFICANCE LEVEL
      0.05 IS USED.
      THE RESULTS ARE GIVEN IN THE MATRIX IQ WHERE EVERY ROW MEANS A POSSIBLE
      COMBINATION FOR TWO FACTORS TO BE EQUAL.
      THE ELEMENTS IN A ROW MEANS FROM LEFT TO RIGHT:
      THE REAL PART OF THE POLE,
      THE REAL PART OF THE ZERO,
      THE IMAGINARY PART OF THE POLE,
      THE IMAGINARY PART OF THE ZERO.
      IF ANY OR BOTH ELEMENTS IN THE THIRD OR FORTH COLUMN IS ZERO, THE POLE
      RESP. THE ZERO IS REAL.
      THE NUMBER OF POSSIBLE COMBINATIONS, I.E. THE NUMBER OF ROWS IN THE MATRI
      IQ IS NQ.
      AUTHOR ERIK BURSTRÖM 1972-12-24

      X VECTOR OF ORDER (NA+NB), CONTAINING THE POLES AND THE ZEROS COMPUTED
      IN SUBROUTINE XPX.
      PX THE COVARIANCE MATRIX OF ORDER (NA+NB)*(NA+NB) COMPUTED
      IN SUBROUTINE XPX.
      IQ THE MATRIX OF ORDER (NC*4) CONTAINING ALL POSSIBLE TESTS.
      NA THE NUMBER OF A-COEFFICIENTS (MIN 1, MAX 10). A(0)=1 IS ASSUMED.
      NB THE NUMBER OF B-COEFFICIENTS(MIN 1, MAX 10).
      NCA THE NUMBER OF COMPLEXVALUED POLES COMPUTED IN SUBROUTINE XPX.
      NCB THE NUMBER OF COMPLEXVALUED ZEROS COMPUTED IN SUBROUTINE XPX.
      NQ THE NUMBER OF POSSIBLE INDIVIDUELL TESTS.
      IPR   IF IPR=0 NOTHING IS PRINTED.
            IF IPR=1 ALL POSITIVE TESTS, THE ESTIMATED COMMON VALUES, THE TEST
             QUANTITIES AND THE MATRIX IQ IS PRINTED.
            IF IPR=2 AS IPR=1 + ALL NEGATIVE TESTS.
      IERR IF IERR=0 NORMAL OUTPUT.
            IF IERR=-1 THE COVARIANCE MATRIX PX IS NOT POSITIVE DEFINITE. THE
            EIGENVALUES OF MATRIX PX ARE COMPUTED AND PRINTED.
      IA DIMENSION PARAMETER OF MATRIX PX.
      IB IF IB=0 B(0)=0 IS ASSUMED.
         IF IB=1 B(0)=1 IS ASSUMED.
      IC DIMENSION PARAMETER OF MATRIX IQ.

      SUBROUTINE REQUIRED
            DESYM
            SOLVS
            EIGS

      DIMENSION X(1),PX(IA,IA),IQ(IC,1)

      DIMENSION ST(20),IQQ(100,4),IND(10),S1(2,20),SP(20,2),SPS(2,2),
     *F(4),Y(2),V(2),R(2,2),EV(2),B(2,2)
      IPR1=IPR
      N=NA+NB-1
      IF (IB.EQ.1) N=NA+NB
      TQ=3.84
      TQ1=5.99
      NAP=NA-NCA
      NBP=NB-1-NCB
      IF (IB.EQ.1) NBP=NB-NCB
      NQ=0
      TT=0.
      ID=2
```

```
      EPS IS USED IN DESYM.

      EPS=1.0E-07
      IERR=0
      DO 17 I=1,100
      DO 17 J=1,4
 17   IQ(I,J)=0

      TEST OF COMPLEX FACTORS

      IF (NCA*NCB.EQ.0) GO TO 20

      COMPUTE TEST VECTOR.

      DO 15 I=1,NCA,2
      DO 15 J=1,NCB,2
      TT=0.
      IR=I
      IR1=I+1
      JR=J+NA
      JR1=J+NA+1
      DO 5 K=1,N
      S1(1,K)=0.
  5   S1(2,K)=0.
      S1(1,IR)=1.
      S1(1,JR)=-1.
      S1(2,IR1)=1.
      S1(2,JR1)=-1.
      Y(1)=X(IR)-X(JR)
      Y(2)=X(IR1)-X(JR1)

      COMPUTE TEST QUANTITY.

      DO 8 K=1,N
      DO 8 L=1,2
  7   SP(K,L)=0.
      DO 8 M=1,N
  8   SP(K,L)=SP(K,L)+PX(K,M)*S1(L,M)
      DO 10 K=1,2
      DO 10 L=1,2
  9   SPS(K,L)=0.
      DO 10 M=1,N
 10   SPS(K,L)=SPS(K,L)+S1(K,M)*SP(M,L)
      EPS=10.**-07*SPS(1,1)
      CALL DESYM(SPS,B,2,EPS,IERR,ID)
      IF(IERR.EQ.-1)GO TO 150
      CALL SOLVS(B,Y,V,2,1,ID)
      TT=V(1)*Y(1)+V(2)*Y(2)
      IF(TT.GT.TO1.AND.IPRT.LE.1) GO TO 15
      IF(IPRT.EQ.0) GO TO 14

      PRINT RESULTS.

      PRINT 101,IR,JR
      PRINT 1001,IR1,JR1
101   FORMAT(/10X,'COMPONENTS',I3,3X,'AND',3X,I3,3X,'ARE TESTED',10X,'TH
     *E REAL PARTS OF TWO COMPLEX FACTORS')
1001  FORMAT(/10X,'COMPONENTS',I3,3X,'AND',3X,I3,3X,'ARE TESTED',10X,'TH
     *E IMAGINARY PARTS OF TWO COMPLEX FACTORS')
      PRINT 102,TT
      IF (TT.GT.TO1) GO TO 15
102   FORMAT (10X,'TEST QUANTITY',G12.5)
```

```
      COMPUTE THE ESTIMATED COMMON VALUE.

      DO 12 K=1,N
   11 ST(K)=0.
      DO 12 L=1,2
   12 ST(K)=ST(K)+SP(K,L)*V(L)
      DO 13 K=1,4
   13 F(K)=0.
      F(1)=X(IR)-ST(IR)
      F(2)=X(JR)-ST(JR)
      F(3)=X(IR1)-ST(IR1)
      F(4)=X(JR1)-ST(JR1)
      PRINT 122,F(1),F(3)
  122 FORMAT(/10X,'PREDICTED VALUE OF THE REAL PARTS',5X,G12.5//10X,'PRE
     *DICTED VALUE OF THE IMAGINARY PARTS',G12.5)
   14 CONTINUE
      NQ=NQ+1
      IQ(NQ,1)=IR
      IQ(NQ,2)=JR
      IQ(NQ,3)=IR1
      IQ(NQ,4)=JR1
   15 CONTINUE

      TEST OF REAL FACTORS

   20 IF (NAP*NBP.EQ.0) GO TO 30

      COMPUTE TEST VECTOR. -
      COMPUTE TEST QUANTITY.

      DO 21 I=1,NAP
      DO 21 J=1,NBP
      TT=0.
      IR=NCA+I
      JR=NCB+J+NA
      AN=PX(IR,IR)+PX(JR,JR)-2*PX(IR,JR)
      TT=(X(IR)-X(JR))**2/AN
      IF(TT.GT.TQ.AND.IPRT.LE.1) GO TO 21
      IF(IPRT.EQ.0) GO TO 22

      PRINT RESULTS.

      PRINT 111,IR,JR
  111 FORMAT(/10X,'COMPONENTS',I3,3X,'AND',3X,I3,3X,'ARE TESTED',10X,'TW
     *O REAL FACTORS')
      PRINT 102,TT
      IF (TT.GT.TQ) GO TO 21

      COMPUTE THE ESTIMATED COMMON VALUE.

      FF=X(IR)*PX(JR,JR)+X(JR)*PX(IR,IR)-(X(IR)+X(JR))*PX(IR,JR)
      FF=FF/AN
      PRINT 103,FF
  103 FORMAT (10X,'PREDICTED VALUE',G12.5)
   22 CONTINUE
      NQ=NQ+1
      IQ(NQ,1)=I+NCA
      IQ(NQ,2)=JR
   21 CONTINUE

      TEST OF ONE COMPLEX POLE AND ONE REAL ZERO.

   30 IF (NCA*NBP.EQ.0) GO TO 50
```

```
    COMPUTE TEST VECTOR.

    DO 45 I=1,NCA,2
    DO 45 J=1,NBP
    TT=0.
    IR=I
    IR1=I+1
    JR=J+NA+NCB
    JR1=0
    DO 35 K=1,N
    S1(1,K)=0.
 35 S1(2,K)=0.
    S1(1,IR)=1.
    S1(1,JR)=-1.
    S1(2,IR1)=1.
    Y(1)=X(IR)-X(JR)
    Y(2)=X(IR1)

    COMPUTE TEST QUANTITY.

    DO 38 K=1,N
    DO 38 L=1,2
 37 SP(K,L)=0.
    DO 38 M=1,N
 38 SP(K,L)=SP(K,L)+PX(K,M)*S1(L,M)
    DO 40 K=1,2
    DO 40 L=1,2
 39 SPS(K,L)=0.
    DO 40 M=1,N
 40 SPS(K,L)=SPS(K,L)+S1(K,M)*SP(M,L)
    EPS=10.**-07*SPS(1,1)
    CALL DESYM(SPS,B,2,EPS,IERR,ID)
    IF(IERR.EQ.-1)GO TO 150
    CALL SOLVS(B,Y,V,2,1,ID)
    TT=V(1)*Y(1)+V(2)*Y(2)
    IF(TT.GT.TQ1.AND.IPRT.LE.1) GO TO 45
    IF(IPRT.EQ.0) GO TO 44

    PRINT RESULTS.

    PRINT 123,IR,JR
123 FORMAT(/10X,'COMPONENTS',I3,3X,'AND',3X,I3,3X,'ARE TESTED',10X,'TH
   *E REAL PARTS OF ONE COMPLEX AND ONE REAL FACTOR')
    PRINT 1023,IR1
023 FORMAT(/10X,'COMPONENT',I4,15X,'IS TESTED',11X,'THE IMAGINARY PART
   * OF ONE COMPLEX FACTOR')
    PRINT 102,TT
    IF (TT.GT.TQ1) GO TO 45

    COMPUTE THE ESTIMATED COMMON VALUE.

    DO 42 K=1,N
 41 ST(K)=0.
    DO 42 L=1,2
 42 ST(K)=ST(K)+SP(K,L)*V(L)
    DO 43 K=1,3
 43 F(K)=0.
    F(1)=X(IR)-ST(IR)
    F(2)=X(JR)-ST(JR)
    F(3)=0.
    PRINT 1122,F(1),F(3)
122 FORMAT(/10X,'PREDICTED VALUE OF THE REAL PARTS',4X,G12.5//10X,'PRE
   *DICTED VALUE OF THE IMAGINARY PART',G12.5)
 44 CONTINUE
```

```
      NQ=NQ+1
      IQ(NQ,1)=IR
      IQ(NQ,2)=JR
      IQ(NQ,3)=IR1
   45 CONTINUE
C
C     TEST OF ONE REAL POLE AND ONE COMPLEX ZERO.
C
   50 IF (NAP*NCB.EQ.0) GO TO 70
C
C     COMPUTE TEST VECTOR.
C
      DO 65 I=1,NAP
      DO 65 J=1,NCB,2
      TT=0.
      IR=I+NCA
      IR1=0
      JR=J+NA
      JR1=J+NA+1
      DO 55 K=1,N
      S1(1,K)=0.
   55 S1(2,K)=0.
      S1(1,IR)=1.
      S1(1,JR)=-1.
      S1(2,JR1)=1.
      Y(1)=X(IR)-X(JR)
      Y(2)=X(JR1)
C
C     COMPUTE TEST QUANTITY.
C
      DO 58 K=1,N
      DO 58 L=1,2
   57 SP(K,L)=0.
      DO 58 M=1,N
   58 SP(K,L)=SP(K,L)+PX(K,M)*S1(L,M)
      DO 60 K=1,2
      DO 60 L=1,2
   59 SPS(K,L)=0.
      DO 60 M=1,N
   60 SPS(K,L)=SPS(K,L)+S1(K,M)*SP(M,L)
      EPS=10.**-07*SPS(1,1)
      CALL DESYM(SPS,B,2,EPS,IERR,ID)
      IF(IERR.EQ.-1)GO TO 150
      CALL SOLVS(B,Y,V,2,1,ID)
      TT=V(1)*Y(1)+V(2)*Y(2)
      IF(TT.GT.TQ1.AND.IPRT.LE.1) GO TO 65
      IF(IPRT.EQ.0) GO TO 64
C
C     PRINT RESULTS.
C
      PRINT 123,IR,JR
      PRINT 2023,JR1
 2023 FORMAT(/10X,'COMPONENT',I16,3X,'IS TESTED',11X,'THE IMAGINARY PART
     * OF ONE COMPLEX FACTOR')
      PRINT 102,TT
      IF (TT.GT.TQ1) GO TO 65
C
C     COMPUTE THE ESTIMATED COMMON VALUE.
C
      DO 62 K=1,N
   61 ST(K)=0.
      DO 62 L=1,2
   62 ST(K)=ST(K)+SP(K,L)*V(L)
      DO 63 K=1,3
```

```
 63 F(K)=0.
    F(1)=X(IR)-ST(IR)
    F(2)=X(JR)-ST(JR)
    F(3)=0.
    PRINT 1122,F(1),F(3)
 64 CONTINUE
    NQ=NQ+1
    IQ(NQ,1)=IR
    IQ(NQ,2)=JR
    IQ(NQ,4)=JR1
 65 CONTINUE
C
C      COMPUTE MATRIX IQ
C
 70 CONTINUE
    IF(NQ.EQ.0) GO TO 300
    DO 81 I=1,10
 81 IND(I)=0
    DO 82 I=1,100
    DO 82 J=1,4
 82 IQQ(I,J)=0
    DO 83 I=1,NA
    DO 83 J=1,NQ
    IF (I.NE.IQ(J,1)) GO TO 83
    IND(I)=IND(I)+1
 83 CONTINUE
 84 CONTINUE
    L=0
    K=1
 85 I=0
 86 CONTINUE
    I=I+1
    IF (I.GT.NA) GO TO  97
    IF (IND(I).EQ.0) GO TO 86
    L=I
    DO 90 J=1,NA
    IF (IND(J).EQ.0) GO TO 90
    IF (IND(I).LE.IND(J)) GO TO 90
    L=J
    I=J
 90 CONTINUE
    KK=0
 92 CONTINUE
 93 KK=KK+1
    IF (KK.GT.NQ) GO TO 95
    IF (IQ(KK,1).NE.L) GO TO 93
    IQQ(K,1)=IQ(KK,1)
    IQQ(K,2)=IQ(KK,2)
    IQQ(K,3)=IQ(KK,3)
    IQQ(K,4)=IQ(KK,4)
    K=K+1
    GO TO 92
 95 CONTINUE
    IND(L)=0
 96 GO TO 85
 97 CONTINUE
    DO 98 I=1,100
    DO 98 J=1,4
 98 IQ(I,J)=IQQ(I,J)
    IF(IPRT.EQ.0) GO TO 99
C
C      PRINT RESULTS.
C
    PRINT 130
```

```
  130 FORMAT(/10X,'MATRIX IQ'/10X,'REAL PART OF POLE, REAL PART OF ZERO,
     * IMAGINARY PART OF POLE, IMAGINARY PART OF ZERO')
      IF (NQ.EQ.0) RETURN
      DO 75 I=1,NQ
   75 PRINT 131,(IQ(I,J),J=1,4)
  131 FORMAT(/7X,4I5)
      GO TO 99
  150 PRINT 200
C
C     COMPUTE EIGENVALUES AND EIGENVECTORS.
C
  200 FORMAT (/10X,'DECOMPOSITION IS IMPOSSIBLE')
      PRINT 4000,IR,JR,IR1,JR1
 4000 FORMAT(/10X,'THIS COMBINATION GOES WRONG'/10X,I5,I5,I10,I5)
      PRINT 5000
 5000 FORMAT(/10X,'MATRIX S*P*S-TR')
      DO 5010 I=1,2
 5010 PRINT 5020,(SPS(I,J),J=1,2)
 5020 FORMAT(8X,10G12.7)
      CALL EIGS(SPS,R,EV,2,ID,0)
      PRINT 5030
 5030 FORMAT(/10X,'MATRIX S*P*S-TR AFTER EIGS')
      DO 5040 I=1,2
 5040 PRINT 5020,(SPS(I,J),J=1,2)
      PRINT 5050
 5050 FORMAT(/10X,'EIGENVECTORS')
      DO 5055 I=1,2
 5055 PRINT 5020,(R(I,J),J=1,2)
      PRINT 5060
 5060 FORMAT(/10X,'VECTOR OF EIGENVALUES')
      PRINT 5020,(EV(I),I=1,2)
   99 CONTINUE
      IF(IERR.NE.-1)IERR=0
      RETURN
  300 CONTINUE
      PRINT 1300
 1300 FORMAT(/10X,'NO INDIVIDUELL TESTS ARE POSSIBLE')
      IF(IERR.NE.-1)IERR=0
      RETURN
      END
```

```
      SUBROUTINE TESTUT(T,X,PX,AM,BM,TT1,IQ,NA,NB,NMA,NMB,NQ,IPR,IERR,
     *IA,IB,IC)
C
C     THE SUBROUTINE COMPUTES THE GREATEST COMMON POLYNOMIAL TO TWO GIVEN
C     POLYNOMIALS IN STASTISTICAL SENSE.
C     THE COMMON POLYNOMIAL IS FOUND BY EXAMINING COMBINATIONS OF THE INDI-
C     VIDUELL TESTS COMPUTED IN SUBROUTINE TEST.
C     AUTHOR ERIK BURSTRÖM 1972-12-24
C
C     T VECTOR OF ORDER (NA+NB), CONTAINGING THE A-COEFFICIENTS AND THE
C     B-COEFFICIENTS.
C     X VECTOR OF ORDER (NA+NB), CONTAINING THE POLES AND THE ZEROS COMPUTED
C     IN SUBROUTINE XPX.
C     PX THE COVARIANCE MATRIX OF VECTOR X, OF ORDER (NA+NB)*(NA+NB), COMPUTED
C     IN SUBROUTINE XPX.
C     AM VECTOR CONTAINING THE NEW ESTIMATED A-COEFFICIENTS.
C     BM VECTOR CONTAINING THE NEW B-COEFFICIENTS.
C     TT1 STATISTICAL TEST QUANTITY.
C     IQ MATRIX OF ORDER(NQ*4), CONTAINING ALL POSSIBLE INDIVIDUELL TESTS
C     COMPUTED IN SUBROUTINE TEST.
C     NA THE NUMBER OF A-COEFFICIENTS (MIN 1, MAX 10).
C     NB THE NUMBER OF B-COEFFICIENTS (MIN 0, MAX 10).
C     NMA THE NUMBER OF A-COEFFICIENTS (MIN 1, MAX 10).
C     NMB THE NUMBER OF B-COEFFICIENTS (MIN 1, MAX 10).
C     NQ THE NUMBER OF POSSIBLE INDIVIDUELL TESTS, COMPUTED IN SUBROUTINE TEST.
C     IPR  IF IPR=0 NOTHING IS PRINTED.
C          IF IPR=1 THE COMPONENTS THAT CAN BE ABBREVIATED, THE TEST QUANTITY
C           AND DEGREES OF FREEDOM, THE NEW ESTIMATED X-VECTOR, THE COMPONENTS
C           THAT ARE EQUAL, THE NEW A-POLYNOMIAL, DEGREE OF A AND POLES, THE NE
C            B-POLYNOMIAL, DEGREE OF B AND ZEROS AND STATIC GAIN FOR THE NEW
C           MODEL ARE PRINTED ONCE.
C          IF IPR=2 AS IPR=1 + THE COMMON POLYNOMIAL AND THE TWO REST-POLYNO-
C           MIALS ARE PRINTED ONCE.
C          IF IPR=3 AS IPR=2 + ALL POSSIBLE COMBINATIONS AND TEST QUANTITIES OF
C           ALL TESTS WITH GREATER AND THE SAME DEGREE OF FREEDOM AS THE BEST
C           POSITIVE TEST.
C          IF IPR=4 AS IPR=3 + ALL POSSIBLE COMBINATIONS AND CORRESPONDING
C           TEST QUANTITIES OF ALL DEGREES OF FREEDOM ARE PRINTED.
C     IERR IF IERR=0 NORMAL OUTPUT.
C          IF IERR=-1 THE COVARIANCE MATRIX PX IS NOT POSITIVE DEFINITE. THE
C          EIGENVALUES AND CORRESPONDING EIGENVECTORS OF MATRIX PX ARE COMPUTE
C          AND PRINTED.
C     IA DIMENSION PARAMETER OF MATRIX PX.
C     IB IF IB=0 B(0)=0 IS ASSUMED.
C        IF IB=1 B(0)=1 IS ASSUMED.
C     IC DIMENSION PARAMETER OF MATRIX IQ.
C
C     D VECTOR OF ORDER 10 CONTAINING THE COEFFICIENTS OF THE COMMON
C     POLYNOMIAL.
C
C     SUBROUTINE REQUIRED
C          DESYM
C          SOLVS
C          ROT
C          EIGS
C          PDIV
C          PMPY
C
      DIMENSION T(1),X(1),PX(IA,IA),IQ(IC,1),AM(1),BM(1)
      COMMON/COMPOL/D(10)
C
      DIMENSION MIX(100),MAX(100),LL(100),S(20,20),Y(20),V(20),SP(20,20)
     *,SPS(20,20),B(20,20),ST(20),F(20),MIN(100),LINE(22),R(20,20),
```

```
      *EV(20)
C
       DATA STRECK/5H-----/
       TJITWO(TN)=1.538*TN+3.2
       IF(NQ.EQ.0) RETURN
       IRANK1=0
       IERR=0
C
C      EPS IS USED IN DESYM.
C
       EPS=1.0E-10
       EPS1=1.0E-05
       TT1=10.**30
       ISKR=IPR
       ID=20
C
C      IX IS THE NUMBER OF FACTORS THAT CAN BE ABBREVIATED.
C
       IX=MIN0(NA,NB-1)
       IF (IB.EQ.1) IX=MIN0(NA,NB)
       IF(IX.EQ.0) RETURN
       ITEST=0
       N=NA+NB-1
       IF (IB.EQ.1) N=NA+NB
       DO 2 I=1,100
       LL(I)=0
       MAX(I)=0
       MIN(I)=0
     2 MIX(I)=0
       DO 5 I=1,NQ
       IT=IQ(I,3)+IQ(I,4)
       IF (IT.EQ.0) GO TO 3
       IU=IQ(I,3)*IQ(I,4)
       IF(IU.EQ.0) GO TO 4
       MAX(I)=2
       GO TO 5
     3 MAX(I)=0
       GO TO 5
     4 MAX(I)=1
     5 CONTINUE
     8 CONTINUE
       DO 99 I=1,NQ
       LL(I)=0
    99 MIX(I)=0
       I=0
C
C      COMPUTE A POSSIBLE COMBINATION OF FACTORS.
C
    10 I=I+1
       KM=0
       DO 200 K=1,I
   200 MIX(K)=0
       IF(I.GT.NQ) RETURN
       I1=I
       MIX(I)=1
       IF (MAX(I).EQ.2) GO TO 11
       KM=KM+1
       LL(KM)=I
       IF (KM.EQ.IX) GO TO 20
       IF (KM.LT.IX) GO TO 12
       LL(KM)=0
       MIX(I)=0
       KM=KM-1
       GO TO 12
```

```
11 KM=KM+2
   LL(KM)=I
   IF (KM.EQ.IX) GO TO 20
   IF (KM.LT.IX) GO TO 12
   LL(KM)=0
   MIX(I)=0
   KM=KM-2
12 I=I+1
   IF (I.GT.NQ) GO TO 18
   II=I-1
   DO 15 J=I1,II
   IF (MIX(J).EQ.0) GO TO 15
   ITT=(IQ(I,1)-IQ(J,1))*(IQ(I,2)-IQ(J,2))
   IF (ITT.NE.0) GO TO 15
   ITE=MAX(I)*MAX(J)
   IF (ITE.NE.1) GO TO 12
   IF (IQ(I,1).EQ.IQ(J,1)) GO TO 13
   IF (IQ(I,4).EQ.0) GO TO 12
   GO TO 15
13 IF (IQ(I,3).EQ.0) GO TO 12
15 CONTINUE
   MIX(I)=1
   IF (MAX(I).EQ.2) GO TO 16
   KM=KM+1
   LL(KM)=I
   IF (KM.LT.IX) GO TO 12
   IF (KM.EQ.IX) GO TO 18
   LL(KM)=0
   KM=KM-1
   MIX(I)=0
   GO TO 12
16 KM=KM+2
   LL(KM)=I
   IF (KM.LT.IX) GO TO 12
   IF (KM.EQ.IX) GO TO 18
   LL(KM)=0
   KM=KM-2
   MIX(I)=0
   GO TO 12
18 CONTINUE
   IF(KM.LE.0) GO TO 80
   IF (KM.LT.IX) GO TO 60
C
C
20 CONTINUE
C
C  COMPUTE MATRIX S.
C
   NRA=1
   DO 40 I=1,NQ
   IF (MIX(I).EQ.0) GO TO 40
   IF (MAX(I).EQ.0) GO TO 31
   IF(MAX(I).EQ.1.AND.I.GT.1) GO TO 26
19 CONTINUE
   DO 25 M=1,N
   S(NRA,M)=0.
   S(NRA+1,M)=0.
   IF (M.EQ.IQ(I,1)) GO TO 21
   IF (M.EQ.IQ(I,2)) GO TO 22
   IF (M.EQ.IQ(I,3)) GO TO 23
   IF (M.EQ.IQ(I,4)) GO TO 24
   GO TO 25
21 S(NRA,M)=1.
   GO TO 25
```

```
   22 S(NRA,M)=-1.
      GO TO 25
   23 S(NRA+1,M)=1.
      GO TO 25
   24 S(NRA+1,M)=-1.
   25 CONTINUE
      NRA=NRA+2
      GO TO 40
C
   26 CONTINUE
      I1=I-1
      ICO=0
      DO 30 J=1,I1
      IF(MIX(J).EQ.0.OR.MAX(J).NE.1) GO TO 30
      IF(IABS(IQ(I,1)-IQ(J,1))+IABS(IQ(I,3)-IQ(J,3)).NE.0.AND.IABS(IQ(I,
     *2)-IQ(J,2))+IABS(IQ(I,4)-IQ(J,4)).NE.0) GO TO 30
      DO 29 M=1,N
      IF(M.EQ.IQ(I,1)) GO TO 27
      IF(M.EQ.IQ(I,2)) GO TO 28
      S(NRA,M)=0.
      GO TO 29
   27 S(NRA,M)=1.
      GO TO 29
   28 S(NRA,M)=-1.
   29 CONTINUE
      NRA=NRA+1
      ICO=1
   30 CONTINUE
      IF(ICO.EQ.0) GO TO 19
      GO TO 40
   31 DO 35 M=1,N
      IF (M.EQ.IQ(I,1)) GO TO 32
      IF (M.EQ.IQ(I,2)) GO TO 33
      S(NRA,M)=0.
      GO TO 35
   32 S(NRA,M)=1.
      GO TO 35
   33 S(NRA,M)=-1.
   35 CONTINUE
      NRA=NRA+1
   40 CONTINUE
C
C     COMPUTE TEST VECTOR.
C
      NRA=NRA-1
      DO 46 K=1,NRA
      Y(K)=0.
      DO 46 L=1,N
   46 Y(K)=Y(K)+S(K,L)*X(L)
C
C     COMPUTE TEST QUANTITY.
C
      DO 48 K=1,N
      DO 48 L=1,NRA
      SP(K,L)=0.
      DO 48 M=1,N
   48 SP(K,L)=SP(K,L)+PX(K,M)*S(L,M)
      DO 50 K=1,NRA
      DO 50 L=1,NRA
      SPS(K,L)=0.
      DO 50 M=1,N
   50 SPS(K,L)=SPS(K,L)+S(K,M)*SP(M,L)
      EPS=10.**-07*SPS(1,1)
      CALL DESYM(SPS,B,NRA,EPS,IRANK1,ID)
```

```
      IF (IRANK1.EQ.-1) GO TO 250
      CALL SOLVS(B,Y,V,NRA,1,IO)
      TT=0.
      DO 52 K=1,NRA
   52 TT=TT+V(K)*Y(K)
      TRA=FLOAT(NRA)
      IF(ISKR.LE.2) GO TO 55
C
C     PRINT RESULTS.
C
      PRINT 101
      DO 53 I=1,NQ
      IF(MIX(I).EQ.0) GO TO 53
      PRINT 102,(IQ(I,J),J=1,4)
   53 CONTINUE
      PRINT 103,TT,NRA
   55 CONTINUE
      IF (TT.GT.TJITWO(TRA)) GO TO 60
      IF(TT.GT.TT1) GO TO 60
      ITEST=1
      TT1=TT
C
C     COMPUTE THE NEW ESTIMATED X-VECTOR.
C
      DO 56 I=1,N
      ST(I)=0.
      DO 56 J=1,NRA
   56 ST(I)=ST(I)+SP(I,J)*V(J)
      DO 57 I=1,N
      F(I)=0.
   57 F(I)=X(I)-ST(I)
      DO 58 I=1,N
      IF(ABS(F(I)).LT.EPS1) F(I)=0.
   58 CONTINUE
      DO 121 I=1,NQ
  121 MIN(I)=MIX(I)

      COMPUTE A NEW POSSIBLE COMBINATION OF FACTORS.
C
   60 CONTINUE
      IF (KM.EQ.1) GO TO 64
      IF (KM.EQ.2) GO TO 61
      GO TO 63
   61 I=LL(KM)
      IF (MAX(I).NE.2) GO TO 63
   64 IF (LL(KM).GE.NQ) GO TO 80
   63 CONTINUE
      IF (LL(KM).GE.NQ) GO TO 65
      I=LL(KM)
      LL(KM)=0
      MIX(I)=0
      IF (MAX(I).EQ.2) GO TO 62
      KM=KM-1
      GO TO 71
   62 KM=KM-2
      GO TO 71
   65 CONTINUE
      I=LL(KM)
      MIX(I)=0
      LL(KM)=0
      IF (MAX(I).EQ.2) GO TO 66
      KM=KM-1
      GO TO 67
   66 KM=KM-2
```

```
 67 CONTINUE
    I=LL(KM)
    IF (I.LE.I1) GO TO 10
    MIX(I)=0
    IF (MAX(I).EQ.2) GO TO 167
    KM=KM-1
    GO TO 71
167 KM=KM-2
 71 I=I+1
    IF (I.LE.NQ) GO TO 68
    I=LL(KM)
    MIX(I)=0
    LL(KM)=0
    IF (MAX(I).EQ.2) GO TO 69
    KM=KM-1
    IF (I.LE.I1) GO TO 10
    GO TO 71
 69 KM=KM-2
    IF (I.LE.I1) GO TO 10
    GO TO 71
 68 II=I-1
    DO 75 J=I1,II
    IF (MIX(J).EQ.0) GO TO 75
    ITT=(IQ(I,1)-IQ(J,1))*(IQ(I,2)-IQ(J,2))
    IF (ITT.NE.0) GO TO 75
    ITE=MAX(I)*MAX(J)
    IF (ITE.NE.1) GO TO 71
    IF (IQ(I,1).EQ.IQ(J,1)) GO TO 73
    IF (IQ(I,4).EQ.0) GO TO 71
    GO TO 75
 73 IF (IQ(I,3).EQ.0) GO TO 71
 75 CONTINUE
    MIX(I)=1
    IF (MAX(I).EQ.2) GO TO 76
    KM=KM+1
    LL(KM)=I
    IF (KM.EQ.IX) GO TO 20
    IF (KM.LT.IX) GO TO 78
    LL(KM)=0
    KM=KM-1
    MIX(I)=0
    GO TO 71
 76 KM=KM+2
    LL(KM)=I
    IF (KM.EQ.IX) GO TO 20
    IF (KM.LT.IX) GO TO 78
    LL(KM)=0
    KM=KM-2
    MIX(I)=0
    GO TO 71
 78 CONTINUE
    I=I+1
    IF (I.GT.NQ) GO TO 60
    GO TO 68
 80 IF(ITEST.EQ.1)GO TO 300
 81 CONTINUE
    IX=IX-1
    IF (IX.LE.0) RETURN
    GO TO 8
300 CONTINUE
    DO 122 I=1,NQ
122 MIX(I)=MIN(I)
    IF(ISKR.EQ.0)GO TO 401
C
```

```
C     PRINT RESULTS.
C
      PRINT 101
  101 FORMAT(/10X,'THE FOLLOWING COMPONENTS ARE TESTED'/10X,35(1H-)/10X,
     *'REAL PART OF POLE, REAL PART OF ZERO, IMAGINARY PART OF POLE, IMA
     *GINARY PART OF ZERO')
      DO 54 I=1,100
      IF (MIX(I).EQ.0) GO TO 54
      PRINT 102,(IQ(I,J),J=1,4)
  102 FORMAT(/7X,4I5)
   54 CONTINUE
      PRINT 103,TT1,NRA
  103 FORMAT(/10X,'TEST QUANTITY',G12.5,10X,'DEGREES OF FREEDOM',I3/10X,
     *25(1H-))
      IF(N.GT.10) GO TO 2999
      PRINT 1300
 1300 FORMAT(/10X,'ESTIMATED VECTOR X'/10X,18(1H-)/10X,'COMPONENT NUMBER
     *')
      PRINT 1302,(I,I=1,N)
 1302 FORMAT(/2X,10I12)
      CALL ENCODE(LINE)
      CALL FMTX(11)
      DO 1234 I=1,N
      CALL FMTA(STRECK,5)
 1234 CALL FMTX(7)
      CALL FMTX(121-N*12)
      PRINT 1235,(LINE(I),I=1,22)
 1235 FORMAT(22A6)
      PRINT 1301,(F(I),I=1,N)
 1301 FORMAT(/8X,10G12.5)
      GO TO 399
 2999 CONTINUE
      PRINT 1300
      PRINT 1302,(I,I=1,10)
      CALL ENCODE(LINE)
      CALL FMTX(11)
      DO 610 I=1,10
      CALL FMTA(STRECK,5)
  610 CALL FMTX(7)
      CALL FMTX(1)
      PRINT 1235,(LINE(I),I=1,22)
      PRINT 1301,(F(I),I=1,10)
      PRINT 1302,(I,I=11,N)
      CALL ENCODE(LINE)
      CALL FMTX(11)
      DO 620 I=11,N
      CALL FMTA(STRECK,5)
  620 CALL FMTX(7)
      CALL FMTX(121-(N-10)*12)
      PRINT 1235,(LINE(I),I=1,22)
      PRINT 1301,(F(I),I=11,N)
  399 CONTINUE
      DO 400 I=1,NQ
      IF(MIX(I).EQ.0) GO TO 400
      IF(MAX(I).EQ.2) GO TO 380
      IF(MAX(I).EQ.1) GO TO 370
  360 PRINT 1360,IQ(I,1),IQ(I,2)
 1360 FORMAT(/10X,'COMPONENTS',I3,3X,'AND',3X,I3,3X,'ARE EQUAL'/10X,'ONE
     * REAL POLE AND ONE REAL ZERO CAN BE ABBREVIATED')
      GO TO 400
  370 IF(IQ(I,4).NE.0) GO TO 375
      PRINT 1370,(IQ(I,J),J=1,3)
 1370 FORMAT(/10X,'COMPONENTS',I3,3X,'AND',3X,I3,3X,'ARE EQUAL',10X,'THE
     * REAL PARTS'//10X,'COMPONENT',I4,3X,'IS =0.00000',20X,'THE REMAINI
```

```
     *NG COMPLEX POLE IS REAL')
      PRINT 1371
 1371 FORMAT(/10X,'ONE COMPLEX POLE AND ONE REAL ZERO CAN BE ABBREVIATED
     *')
      GO TO 400
  375 PRINT 1375,IQ(I,1),IQ(I,2),IQ(I,4)
 1375 FORMAT(/10X,'COMPONENTS',I3,3X,'AND',3X,I3,3X,'ARE EQUAL',10X,'THE
     * REAL PARTS'//10X,'COMPONENT',I4,3X,'IS =0.00000',20X,'THE REMAINI
     *NG COMPLEX ZERO IS REAL')
      PRINT 1376
 1376 FORMAT(/10X,'ONE REAL POLE AND ONE COMPLEX ZERO CAN BE ABBREVIATED
     *')
      GO TO 400
  380 PRINT 1380,(IQ(I,J),J=1,4)
 1380 FORMAT(/10X,'COMPONENTS',I3,3X,'AND',3X,I3,3X,'ARE EQUAL',10X,'THE
     * REAL PARTS'//10X,'COMPONENTS',I3,3X,'AND',3X,I3,3X,'ARE EQUAL',10
     *X,'THE IMAGINARY PARTS'//10X,'TWO COMPLEX POLES AND TWO COMPLEX ZE
     *ROS CAN BE ABBREVIATED')
  400 CONTINUE
  401 CONTINUE
C
C     COMPUTE THE NEW A- AND B-POLYNOMIALS.
C
      CALL PDIV(T,X,F,AM,BM,IQ,MIX,MAX,NA,NB,NMA,NMB,NQ,IPR,IERR,IB,IC)
      GO TO 301

      COMPUTE EIGENVALUES AND EIGENVECTORS.

  250 PRINT 260
  260 FORMAT (/10X,'DECOMPOSITION IS IMPOSSIBLE')
      PRINT 4000
 4000 FORMAT(/10X,'MATRIX S')
      DO 5000 I=1,NRA
 5000 PRINT 5100,(S(I,J),J=1,N)
 5100 FORMAT(10X,10G10.5/10X,10G10.5)
      PRINT 5200
 5200 FORMAT(/10X,'MATRIX S*P*S-TR')
      DO 5300 I=1,NRA
 5300 PRINT 7020,(SPS(I,J),J=1,NRA)
      CALL EIGS(SPS,R,EV,NRA,ID,0)
      PRINT 7000
 7000 FORMAT(/10X,'MATRIX SPS-TR AFTER EIGS')
      DO 7010 I=1,NRA
 7010 PRINT 7020,(SPS(I,J),J=1,NRA)
 7020 FORMAT(10X,10G12.5)
      PRINT 7005
 7005 FORMAT(/10X,'MATRIX R')
      DO 7030 I=1,NRA
 7030 PRINT 7020,(R(I,J),J=1,NRA)
      PRINT 7015
 7015 FORMAT(/10X,'VECTOR OF EIGENVALUES')
      PRINT 7020,(EV(I),I=1,NRA)
      IERR=-1
  301 CONTINUE
      IF(ITEST.EQ.0)GO TO 60
      IF(ISKR.GE.4)GO TO 81
      RETURN
      END
```

```
      SUBROUTINE PDIV(T,X,F,AM,BM,IQ,MIX,MAX,NA,NB,NMA,NMB,NQ,IPR,IERR,
     *IB,IC)
C
C     THE SUBROUTINE DIVIDES TWO GIVEN POLYNOMIALS WITH THE GREATEST COMMON
C     POLYNOMIAL. THE COMMON POLYNOMIAL IS COMPUTED IN SUBROUTINE TESTUT.
C     THE POLES, ZEROS AND STATIC GAIN FOR THE NEW SYSTEM IS COMPUTED.
C     THIS SUBROUTINE MUST BE CALLED FROM SUBROUTINE TESTUT.
C     AUTHOR ERIK BURSTRÖM 1972-12-24
C
C     T=(A(1),...,A(NA),B(1),...,B(NB)) VECTOR OF ORDER (NA+NB).
C     X VECTOR OF ORDER (NA+NB) CONTAINING POLES AND ZEROS, COMPUTED IN SUB-
C     ROUTINE XPX.
C     F VECTOR OF ORDER (NA+NB) CONTAINING NEW ESTIMATES OF POLES AND ZEROS
C     ON THE ASSUMPTION THAT SOME FACTORS ARE EQUAL, COMPUTED IN SUB-
C     ROUTINE TESTUT.
C     AM VECTOR OF ORDER NMA (MIN 1, MAX 10) CONTAINING THE NEW A-COEFFICIENTS.
C     BM VECTOR OF ORDER NMB (MIN 1, MAX 10) CONTAINING THE NEW B-COEFFICIENTS.
C     IQ MATRIX CONTAINING ALL POSSIBLE INDIVIDUELL TESTS COMPUTED IN SUB-
C     ROUTINE TEST.
C     MIX VECTOR OF ORDER NQ COMPUTED IN SUBROUTINE TESTUT. IF MIX(I)=1
C     A CERTAIN COMBINATION OF FACTORS WILL BE ABBREVIATED AND IF MIX(I)=0
C     A CERTAIN COMBINATION WILL NOT BE ABBREVIATED.
C     MAX VECTOR OF ORDER NQ COMPUTED IN SUBROUTINE TESTUT. IF MAX(I)=0 TWO
C     REAL FACTORS CAN BE ABBREVIATED, IF MAX(I)=1 ONE REAL AND ONE COMPLEX
C     FACTOR CAN BE ABBREVIATED AND IF MAX(I)=2 TWO COMPLEX FACTORS CAN BE
C     ABBREVIATED. NOTE THAT IF MAX(I)=1 THERE ARE TWO DEGREES OF FREEDOM
C     ALTHOUGH ONLY ONE POLE AND ONE ZERO CAN BE ABBREVIATED.
C     NA NUMBER OF A-COEFFICIENTS.
C     NB NUMBER OF B-COEFFICIENTS.
C     NMA NUMBER OF NEW A-COEFFICIENTS.
C     NMB NUMBER OF NEW B-COEFFICIENTS.
C     NQ NUMBER OF POSSIBLE INDIVIDUELL TESTS COMPUTED IN SUBROUTINE TEST.
C     IPR  IF IPR=0 NOTHING IS PRINTED.
C          IF IPR=1 THE NEW A- AND B-POLYNOMIALS, POLES, ZEROS AND STATIC GAIN
C           ARE PRINTED.
C          IF IPR=2 AS IPR=1 + THE COMMON POLYNOMIAL AND THE REST POLYNOMIALS
C           ARE PRINTED.
C     IERR IF IERR=0 NORMAL OUTPUT
C          IF IERR=-1 COMPUTATION OF POLES OR ZEROS HAS FAILED.
C     IB IF IB=0 B(0)=0 IS ASSUMED.
C        IF IB=1 B(0)=1 IS ASSUMED.
C     IC DIMENSION PARAMETER OF MATRIX IQ.
C
C     D VECTOR OF ORDER 10 CONTAINING THE COEFFICIENTS OF THE COMMON
C     POLYNOMIAL.
C
C     SUBROUTINE REQUIRED
C          ROT
C          PMPY
C
      DIMENSION T(1),X(1),F(1),IQ(IC,1),MIX(1),MAX(1),AM(1),BM(1)
      COMMON/COMPOL/D(10)
C
      DIMENSION R(10),Y(2),YY(3),A(10),A1(10),RE(10),A2(10),AR2(10),
     *AI2(10)
      EPS=1.0E-05
      ISKR=IPR
      IF(IB.EQ.0)G1=0.
      IF(IB.EQ.1)G1=1.
      G2=1.
      G=0.
C
C     COMPUTE THE COMMON POLYNOMIAL.
```

```
C
      DO 1 I=1,10
      AM(I)=0.
      BM(I)=0.
      R(I)=0.
    1 D(I)=0.
      IDD=1
      IR=1
      IDY=2
      IDYY=3
      Y(1)=1.
      YY(1)=1.
      R(1)=1.
      D(1)=1.
C
      DO 15 I=1,NQ
      IF(MIX(I).EQ.0) GO TO 15
      IF(MAX(I).EQ.2) GO TO 10
      I1=IQ(I,1)
      Y(2)=-F(I1)
      CALL PMPY(D,IDD,R,IR,Y,IDY)
      GO TO 12
   10 I1=IQ(I,1)
      I2=IQ(I,3)
      YY(2)=-2*F(I1)
      YY(3)=F(I1)**2+F(I2)**2
      CALL PMPY(D,IDD,R,IR,YY,IDYY)
   12 IR=IDD
      DO 14 J=2,IDD
   14 R(J)=D(J)
   15 CONTINUE
      ID=IDD-1
      IF(ID.LE.0) RETURN
      IF(ISKR.LT.2) GO TO 19

      PRINT RESULTS.

      PRINT 290,ID,R(1)
  290 FORMAT(/10X,'THE COMMON POLYNOMIAL D(Z)',10X,'DEGREE OF D',I5/10X,
     *'D(0)=  ',G10.5)
      IF(IDD.EQ.1) GO TO 19
      DO 291 I=2,IDD
      I0=I-1
  291 PRINT 292,I0,R(I)
  292 FORMAT(/10X,'D(',I1,')=  ',G10.5)
   19 CONTINUE
      KK=0
C
C     COMPUTE THE NEW A- AND B-POLYNOMIALS.
C
      DO 20 I=2,IDD
   20 D(I-1)=D(I)
      D0=1.
   16 KK=KK+1
      A0=1.
      IF(IB.EQ.0.AND.KK.EQ.2) A0=0.
      IF(KK.EQ.1) NA1=NA
      IF(IB.EQ.0.AND.KK.EQ.2) NA1=NB-1
      IF(IB.EQ.1.AND.KK.EQ.2) NA1=NB
      A10=A0/D0
      IF(ID.EQ.NA1) GO TO 80
      IF(KK.EQ.2) GO TO 121
      DO 21 I=1,NA1
   21 A(I)=T(I)
```

```
      GO TO 22
  121 DO 122 I=1,NA1
  122 A(I)=T(I+NA)
   22 CONTINUE
      I3=NA1-ID
      IF(I3.GT.ID) GO TO 31
      A10=A0/D0
      A1(1)=(A(1)-D(1)*A10)/D0
      IF(I3.LE.1) GO TO 26
      DO 24 I=2,I3
      A1(I)=A(I)
      IJ=I-1
      DO 23 J=1,IJ
   23 A1(I)=A1(I)-D(J)*A1(I-J)
      A1(I)=(A1(I)-D(I)*A10)/D0
   24 CONTINUE
   26 CONTINUE
      IF(KK.EQ.1.OR.ABS(A10).GT.EPS) GO TO 39
      A1(I3+1)=A(I3+1)
      DO 27 J=1,I3
   27 A1(I3+1)=A1(I3+1)-D(J)*A1(-J+I3+1)
      A1(I3+1)=(A1(I3+1)-D(I3+1)*A10)/D0
      GO TO 39
   31 CONTINUE
      A10=A0/D0
      A1(1)=(A(1)-D(1)*A10)/D0
      IF(ID.LE.1) GO TO 34
      DO 33 I=2,ID
      A1(I)=A(I)
      IJ=I-1
      DO 32 J=1,IJ
   32 A1(I)=A1(I)-D(J)*A1(I-J)
      A1(I)=(A1(I)-D(I)*A10)/D0
   33 CONTINUE
   34 IL=ID+1
      DO 36 I=IL,I3
      A1(I)=A(I)
      DO 35 J=1,ID
   35 A1(I)=A1(I)-D(J)*A1(I-J)
      A1(I)=A1(I)/D0
   36 CONTINUE
      IF(KK.EQ.1.OR.ABS(A10).GT.EPS) GO TO 39
      A1(I3+1)=A(I3+1)
      DO 37 J=1,ID
   37 A1(I3+1)=A1(I3+1)-D(J)*A1(-J+I3+1)
      A1(I3+1)=A1(I3+1)/D0
   39 CONTINUE
      IF(ABS(A10).LT.EPS) GO TO 51
      DO 42 I=1,I3
   42 A2(I)=A1(I)/A10
      GO TO 53
   51 CONTINUE
      DO 52 I=1,I3
   52 A2(I)=A1(I+1)/A1(1)
   53 CONTINUE
C
C     COMPUTE POLES AND ZEROS.
C
      CALL ROT(I3,1,A2,A2,AR2,AI2,NERR1)
      IF(NERR1.GT.0) GO TO 2000
C
C     PRINT RESULTS.
C
      IF(KK.EQ.1) PRINT 101,I3,A10
```

```
 101 FORMAT(/10X,'THE NEW A-POLYNOMIAL',10X,'DEGREE OF A',I3,10X,'ROOTS
    * OF A'//10X,'A(0)= ',G10.5)
      IF(KK.EQ.1) PRINT 102,((I,A1(I),AR2(I),AI2(I)),I=1,I3)
 102 FORMAT(/10X,'A(',I1,')= ',G10.5,39X,2G10.5)
      IF(KK.EQ.2) PRINT 1201,I3,A10
1201 FORMAT(/10X,'THE NEW B-POLYNOMIAL',10X,'DEGREE OF B',I3,10X,'ROOTS
    * OF B'//10X,'B(0)= ',G10.5)
      IF(KK.EQ.2) PRINT 202,((I,A1(I),AR2(I),AI2(I)),I=1,I3)
 202 FORMAT(/10X,'B(',I1,')= ',G10.5,39X,2G10.5)
      I4=I3+1
      IF(KK.EQ.2.AND.ABS(A10).LT.EPS) PRINT 202,(I4,A1(I4))
      IF(KK.EQ.1) GO TO 61
      IF(KK.EQ.2) GO TO 62
  61 CONTINUE
      DO 54 I=1,I3
      G2=G2+A1(I)
  54 AM(I)=A1(I)
      NMA=I3
      GO TO 200
  62 CONTINUE
      DO 55 I=1,I3
  55 BM(I)=A1(I)
      BM(I4)=A1(I4)
      IF(ABS(A10).LT.EPS)NMB=I3+1
      IF(ABS(A10).GT.EPS)NMB=I3
      DO 71 I=1,NMB
  71 G1=G1+A1(I)
      GO TO 200
  80 CONTINUE
      IF(KK.EQ.1) PRINT 114
 114 FORMAT(/10X,'THE NEW A-POLYNOMIAL IS CONSTANT  1.0')
      NMA=0
      IF(IB.EQ.0.AND.KK.EQ.2) PRINT 113,T(NA+1)
 113 FORMAT(/10X,'THE B-POLYNOMIAL IS A CONSTANT',G10.5)
      NMB=1
      BM(1)=T(NA+1)
      G1=T(NA+1)
      IF(IB.EQ.1.AND.KK.EQ.2) PRINT 115
 115 FORMAT(/10X,'THE NEW B-POLYNOMIAL IS CONSTANT  1.0')
      NMB=0
 200 CONTINUE
      IF(ISKR.LT.2) GO TO 300
C
C     COMPUTE THE REST-POLYNOMIAL.
C
      RE0=A(I3+1)
      IF(ID.EQ.1) GO TO 208
      DO 201 I=1,I3
 201 RE0=RE0-D(I)*A1(-I+I3+1)
      RE0=RE0-D(I3+1)*A10
      I6=NA1-1-2*I3
      DO 204 I=1,I6
      RE(I)=A(I+I3+1)
      DO 203 J=1,I3
 203 RE(I)=RE(I)-D(J+I)*A1(-J+I3+1)
      RE(I)=RE(I)-D(I3+I+1)*A10
 204 CONTINUE
      I7=I6+1
      I8=ID-1
      DO 207 I=I7,I8
      RE(I)=A(I+I3+1)
      I9=ID-I
      DO 205 J=1,I9
 205 RE(I)=RE(I)-D(J+I)*A1(-J+I3+1)
```

```
  207 CONTINUE
  208 CONTINUE
      PRINT 211,I3,REO
  211 FORMAT(/10X,'THE REST-POLYNOMIAL',10X,'DEGREE OF REST',I3//10X,'RE
     *(0)= ',G10.5)
      IF(ID.EQ.1) GO TO 280
      PRINT 212,((I,RE(I)),I=1,I8)
  212 FORMAT(/10X,'RE(',I1,')= ',G10.5)
  280 CONTINUE
      GO TO 300
 2000 IERR=-1
      IF(KK.EQ.1)PRINT 2001
      IF(KK.EQ.2)PRINT 2002
 2001 FORMAT(/10X,'COMPUTATION OF POLES IS IMPOSSIBLE')
 2002 FORMAT(/10X,'COMPUTATION OF ZEROS IS IMPOSSIBLE')
  300 IF(KK.LT.2) GO TO 16
C
C     COMPUTE STATIC GAIN.
C
      G=G1/G2
      PRINT 260,G
  260 FORMAT(/10X,'STATIC GAIN',5X,G10.5)
      RETURN
      END
```

```
      SUBROUTINE COFAC(T,PT,T1,T2,TT1,TT2,NA,NB,NMA1,NMB1,NMA2,NMB2,IPR,
     *IERR,IA,IB)
C
C     THIS SUBROUTINE COMPUTES THE GREATEST COMMON POLYNOMIAL TO TWO GIVEN
C     POLYNOMIALS BY USING THE EUCLIDEAN ALGORITHM. THE ORIGINAL POLY-
C     NOMIALS ARE ABBREVIATED WITH THE COMMON POLYNOMIALS. TO DECIDE
C     WHETHER TWO POLYNOMIALS HAVE COMMON FACTORS OR NOT, THE REST-POLY-
C     NOMIAL COMPUTED AT EACH DIVISION IS TESTED TO BE ZERO ON THE
C     ASSUMPTION THAT THE COEFFICIENTS ARE GAUSSIAN, I.E. A CHI-
C     SQUARE TEST WITH THE SIGNIFICANCE LEVEL OF 5% IS USED.
C     TWO DIFFERENT METHODS ARE USED WHEN THE DIVISIOS ARE DONE. ON ONE
C     HAND THE DIVISOR POLYNOMIAL IS NORMALIZED, I.E. THE HIGHEST DEGREE
C     COEFFICIENT IS 1 (VERSION 1), AND ON THE OTHER HAND THE DIVISOR
C     POLYNOMIAL IS NOT NORMALIZED, I.E. THE DIVISIONS ARE DONE STRAIGHT-
C     FORWARD (VERSION 2).
C     AUTHOR ERIK BURSTRÖM 1972-12-24
C
C     T=(A(1),...,A(NA),B(1),...,B(NB)) VECTOR OF ORDER (NA+NB) CONTAINING
C     THE COEFFICIENTS OF THE ORIGINAL A- RESP. B-POLYNOMIALS.
C     PT MATRIX OF ORDER (NA+NB)*(NA+NB) THE COVARIANCE MATRIX OF T.
C     T1 VECTOR OF ORDER (NMA1+NMB1) AT OUTPUT CONTAINING THE COEFFICIENTS
C     OF THE NEW A- RESP. B-POLYNOMIALS, WHEN VERSION 1 OF THE EUCLIDEAN
C     ALGORITHM IS USED.
C     T2 VECTOR OF ORDER (NMA2+NMB2) AT OUTPUT CONTAINING THE COEFFICIENTS
C     OF THE NEW A- RESP. B-POLYNOMIALS, WHEN VERSION 2 OF THE EUCLIDEAN
C     ALGORITHM IS USED.
C     TT1 STATISTICAL TEST-QUANTITY COMPUTED IN VERSION 1.
C     TT2 STATISTICAL TEST-QUANTITY COMPUTED IN VERSION 2.
C     NA NUMBER OF ORIGINAL A-COEFFICIENTS (MIN 1, MAX 10).
C     NB NUMBER OF ORIGINAL B-COEFFICIENTS (MIN 1, MAX 10). NA AND NB MUST
C     BE EQUAL.
C     NMA1 NUMBER OF ESTIMATED A-COEFFICIENTS IN VERSION 1.
C     NMB1 NUMBER OF ESTIMATED B-COEFFICIENTS IN VERSION 1.
C     NMA2 NUMBER OF ESTIMATED A-COEFFICIENTS IN VERSION 2.
C     NMB2 NUMBER OF ESTIMATED B-COEFFICIENTS IN VERSION 2.
C     IPR  IF IPR=0 NOTHING IS PRINTED.
C          IF IPR=1 VECTOR T, MATRIX PT AND THE STATIC GAIN FOR THE ORIGINAL
C           SYSTEM ARE PRINTED. ALL TEST-QUANTITIES DOWN TO THE FIRST ONE WHICH
C           MEANS A POSITIVE TEST, THE NEW A-POLYNOMIAL, DEGREE OF A AND POLES
C           THE NEW B-POLYNOMIAL, DEGREE OF B AND ZEROS, THE STATIC GAIN FOR
C           THE NEW SYSTEM, THE NEW VECTOR T, NUMBER OF NEW A- RESP. B-COEFFI-
C           CIENTS ARE PRINTED FOR BOTH VERSIONS.
C          IF IPR=2 AS IPR=1 + ALL REST-POLYNOMIALS AND CORRESPONDING COVA-
C           RIANCE MATRICES COMPUTED AT EACH DIVISION AND THE REST-POLY-
C           NOMIALS COMPUTED AT THE FINAL ABBREVIATION ARE PRINTED FOR
C           BOTH VERSIONS.
C     IERR IF IERR=0 NORMAL OUTPUT.
C          IF IERR=-1 EITHER IS MATRIX PT NOT POSITIVE DEFINITE OR IS IT
C          IMPOSSIBLE TO COMPUTE POLES OR ZEROS TO THE NEW SYSTEMS.
C     IA DIMENSON PARAMETER OF PT.
C     IB IF IB=0 B(0)=0 IS ASSUMED.
C        IF IB=1 B(0)=1 IS ASSUMED.
C
C     SUBROUTINE REQUIRED
C          EUKL1
C          EUKL2
C          PDV
C          ROT
C          DESYM
C          SOLVS
C
      DIMENSION T(1),PT(IA,IA),T1(1),T2(1)
C
```

```
      DIMENSION T3(22),PT3(22,22),T4(22),PT4(22,22)
      IF(IPR.GT.0) PRINT 100,NA,NB
100   FORMAT(1H1,9X,'PRINTOUT FROM COFAC',10X,'NA=',I3,3X,'NB=',I3/10X,1
     *9(1H*))
      NA1=NA+1
      IF(IB.EQ.0) NB1=NB
      IF(IB.EQ.1) NB1=NB+1
      N1=NA1+NB1
      IA1=22
      T3(1)=1.
      IF(IB.EQ.1) GO TO 11
      DO 2 I=2,N1
      T3(I)=T(I-1)
      DO 2 J=2,N1
    2 PT3(I,J)=PT(I-1,J-1)
      DO 3 I=1,N1
      PT3(I,1)=0.
    3 PT3(1,I)=0.
      GO TO 21
   11 CONTINUE
      DO 13 I=2,NA1
      T3(I)=T(I-1)
      DO 13 J=2,NA1
   13 PT3(I,J)=PT(I-1,J-1)
      NA2=NA1+1
      T3(NA2)=1.
      DO 14 I=1,N1
      PT3(I,1)=0.
   14 PT3(1,I)=0.
      NA3=NA2+1
      DO 15 I=NA3,N1
      T3(I)=T(I-2)
      DO 15 J=NA3,N1
   15 PT3(I,J)=PT(I-2,J-2)
      DO 16 I=1,N1
      PT3(I,NA2)=0.
   16 PT3(NA2,I)=0.
      DO 17 I=2,NA1
      DO 17 J=1,NB1
      PT3(I,J+NA2)=PT(I-1,J+NA2-2)
   17 PT3(J+NA2,I)=PT(J+NA2-2,I-1)
   21 CONTINUE
      DO 22 I=1,N1
      T4(I)=T3(I)
      DO 22 J=1,N1
   22 PT4(I,J)=PT3(I,J)
      IF(IPR.LE.0) GO TO 24
C
C     PRINT T AND PT.
C
      PRINT 200
  200 FORMAT(/10X,'VECTOR T')
      PRINT 201,(T3(I),I=1,N1)
  201 FORMAT(8X,10G12.5)
      PRINT 202
  202 FORMAT(/10X,'COVARIANCE MATRIX')
      DO 23 I=1,N1
   23 PRINT 201,(PT3(I,J),J=1,N1)
   24 CONTINUE
C
C     COMPUTE THE NEW A- AND B-POLYNOMIALS, VERSION 1.
C
      CALL EUKL1(T3,PT3,T1,TT1,NA1,NB1,NMA1,NMB1,NC1,IPR,IERR,IA1,IB)
      IF(IPR.LE.0) GO TO 28
```

```
      PRINT THE NEW VECTOR T.

      PRINT 150,NMA1,NMB1
  150 FORMAT(/10X,'THE NEW ESTIMATED VECTOR T',10X,'NMA=',I3,3X,'NMB=',
     *I3)
      NN=NMA1+NMB1
      PRINT 160,(T1(I),I=1,NN)
  160 FORMAT(7X,10G12.5)
   28 CONTINUE

      COMPUTE THE NEW A- AND B-POLYNOMIALS, VERSION 2.

      CALL EUKL2(T4,PT4,T2,TT2,NA1,NB1,NMA2,NMB2,NC2,IPR,IERR,IA1,IB)
      IF(IPR.LE.0) RETURN

      PRINT THE NEW VECTOR T.

      NN=NMA2+NMB2
      PRINT 150,NMA2,NMB2
      PRINT 160,(T2(I),I=1,NN)
      RETURN
      END
```

```
      SUBROUTINE EUKL1(X1,PX1,X2,TT,NA1,NB1,NMA,NMB,NC,IPR,IERR,IA1,IB)
C
C     THIS SUBROUTINE COMPUTES COMMON FACTORS TO TWO GIVEN POLYNOMIALS BY
C     USING THE EUCLIDEAN ALGORITHM. THE DIVISOR-POLYNOMIAL IS NORMA-
C     LIZED, I.E. THE HIGHEST DEGREE COEFFICIENT IS 1. TO DECIDE WHETHER
C     TWO POLYNOMIALS HAVE COMMON FACTORS OR NOT, THE REST-POLY-
C     NOMIAL COMPUTED AT EVERY DIVISION IS TESTED TO BE ZERO ON THE ASSUMPTION
C     THAT THE COEFFICIENTS IN THE ORIGINAL POLYNOMIALS ARE GAUSSIAN.
C     IF THE TWO GIVEN POLYNOMIALS HAVE COMMON FACTORS THEY ARE ABBRE-
C     VIATED WITH THE GREATEST COMMON POLYNOMIAL.
C     AUTHOR ERIK BURSTRÖM 1972-12-24
C
C     X1=(1,A(1),....,A(NA),B(0),....,B(NB)) VECTOR OF ORDER (NA1+NB1). IF
C     B(0)=0 B(0) IS OMITTED.
C     PX1 MATRIX OF ORDER (NA1+NB1)*(NA1+NB1) THE COVARIANCE MATRIX
C     OF X1.
C     X2 VECTOR OF ORDER (NMA+NMB) CONTAINING THE NEW ESTIMATED A- RESP. B-
C     POLYNOMIALS.
C     TT STATISTICAL TEST QUANTITY.
C     NA1 NUMBER OF A-COEFFICIENTS (MIN 2, MAX 11).
C     NB1 NUMBER OF B-COEFFICIENTS (MIN 1, MAX 11).
C     NMA NUMBER OF NEW A-COEFFICIENTS.
C     NMB NUMBER OF NEW B-COEFFICIENTS.
C     NC NUMBER OF DEGREES OF FREEDOM.
C     IPR  IF IPR=0 NOTHING IS PRINTED.
C          IF IPR=1 THE STATIC GAIN FOR THE ORIGINAL SYSTEM, THE TEST QUANTITY
C            AND THE NUMBER OF DEGREES OF FREEDOM AT EACH DIVISION DOWN TO
C            THE FIRST POSITIVE TEST, THE NEW A-POLYNOMIAL, DEGREE OF A AND
C            POLES, THE NEW B-POLYNOMIAL, DEGREE OF B AND ZEROS AND THE STATIC
C            GAIN FOR THE NEW SYSTEM ARE PRINTED.
C          IF IPR=2 AS IPR=1 + THE REST-POLYNOMIAL AND CORRESPONDING
C            COVARIANCE MATRIX OBTAINED AT EACH DIVISION AND THE TWO REST-POLY-
C            NOMIALS OBTAINED AT THE FINAL ABBREVIATION ARE PRINTED.
C     IERR IF IERR=0 NORMAL OUTPUT.
C          IF IERR=-1 EITHER IS THE COVARIANCE MATRIX PX1 NOT POSITIVE
C          DEFINITE, OR IS IT IMPOSSIBLE TO COMPUTE POLES OR ZEROS TO
C          THE NEW SYSTEM.
C     IA1 DIMENSION PARAMETER OF X1 AND PX1.
C     IB IF IB=0 B(0)=0 IS ASSUMED.
C        IF IB=1 B(0)=1 IS ASSUMED.
C
C     SUBROUTINE REQUIRED
C          PDV
C          DESYM
C          SOLVS
C          ROT
C
      DIMENSION X1(1),PX1(IA1,IA1),X2(1)
C
      DIMENSION C(10),PC(10,10),Y(10),H1(11,11),H2(11,11),D(11,11),E(11,
     *11),DYDX(22,22),V(10,10),R(10),DVNDA(11),DVNDB(11),DIVI(11),S(21,
     *21),PX2(21,21),R1(10)
      REAL K1,K2,K3
      TJITWO(TN)=1.538*TN+3.2
      IERR=0
      G=0.
      G1=0.
      G2=0.
      G32=0.
      G21=1.
      IF(IB.EQ.0)G22=0.
      IF(IB.EQ.1)G22=1.
      IA2=10
```

```
      NA=NA1
      NB=NB1
      NC=NB1-1
      IF(IB.EQ.0.AND.NB1.EQ.1)GO TO 68
      IF(NC.LE.0) RETURN
      IF(IPR.GT.0) PRINT 600,X1(NA+1)
  600 FORMAT(/10X,'VERSION 1  THE B-POLYNOMIAL IS SCALED  B(0)= ',G10.5/
     *10X,5(2H- ))
      DO 3 I=1,NA1
      G1=G1+X1(I)
    3 BVNDA(I)=X1(I)
      NA2=NA1+1
      N1=NA1+NB1
      DO 4 I=1,NB1
      G2=G2+X1(I+NA1)
    4 BVNDB(I)=X1(I+NA1)

      COMPUTE THE STATIC GAIN.

      G=G2/G1
      IF(IPR.GT.0)PRINT 601,G
  601 FORMAT(/10X,'STATIC GAIN',5X,G10.5)
      DO 5 I=1,11
      DO 5 J=1,11
    5 D(I,J)=0.
      DO 6 I=1,22
      DO 6 J=1,22
    6 DYDX(I,J)=0.
      DO 7 I=1,10
    7 R(I)=0.
      DO 8 I=1,11
      DO 8 J=1,11
      E(I,J)=0.
      IF(I.NE.J) GO TO 8
      E(I,J)=1.
    8 CONTINUE
   18 CONTINUE
      IF(IB.EQ.1.AND.NA.EQ.NB) GO TO 108
C
C     COMPUTE REST-POLYNOMIAL AND CORRESPONDING COVARIANCE MATRIX.
C
      K1=X1(1)/X1(NA+1)
      K2=1./X1(NA+1)*(X1(2)-K1*X1(NA+2))
      NA2=NA
      IF(NC.LE.1) GO TO 12
      N3=NC-1
      DO 11 I=1,N3
   11 C(I)=X1(I+2)-K1*X1(I+NA+2)-K2*X1(I+NA+1)
      C(NC)=X1(NA)-K2*X1(NA+NB)
      GO TO 17
   12 C(1)=X1(3)-K2*X1(5)
   17 CONTINUE
      DO 213 I=1,NC
      DO 212 J=1,NA
  212 H1(I,J)=0.
      DO 213 J=1,NB
  213 H2(I,J)=0.
      DO 13 I=1,N3
      H1(I,I+2)=1.
      H1(I,1)=-1./X1(NA+1)*X1(I+NA+2)+X1(NA+2)/X1(NA+1)**2*X1(I+NA+1)
   13 H1(I,2)=-1./X1(NA+1)*X1(I+NA+1)
      H1(NC,1)=X1(NA+2)/X1(NA+1)**2*X1(NA+NB)
      H1(NC,2)=-1./X1(NA+1)*X1(NA+NB)
      H1(NC,NB)=1.
```

```
      DO 14 I=1,N3
      H2(I,I+2)=-X1(1)/X1(NA+1)
 14   H2(I,1)=X1(1)/X1(NA+1)**2*X1(I+NA+2)+X1(2)/X1(NA+1)**2*X1(I+NA+1)-
     *2.*X1(1)*X1(NA+2)/X1(NA+1)**3*X1(I+NA+1)
      H2(2,1)=-X1(2)/X1(NA+1)+2.*X1(1)*X1(NA+2)/X1(NA+1)**2
      DO 15 I=2,NC
      H2(I,2)=X1(1)*X1(I+NA+1)/X1(NA+1)**2
 15   H2(I,I+1)=-X1(2)/X1(NA+1)+X1(1)*X1(NA+2)/X1(NA+1)**2
      H2(NC,1)=X1(2)/X1(NA+1)**2*X1(NA+NB)-2.*X1(1)*X1(NA+2)/X1(NA+1)**3
     **X1(NA+NB)
      GO TO 19
 108  CONTINUE
      K3=X1(1)/X1(NA+1)
      DO 111 I=1,NC
 111  C(I)=X1(I+1)-K3*X1(I+NA+1)
      DO 112 I=1,NC
      DO 112 J=1,NA1
      H1(I,J)=0.
 112  H2(I,J)=0.
      DO 113 I=1,NC
      H1(I,1)=-1./X1(NA+1)*X1(I+NA+1)
      H1(I,I+1)=1.
      H2(I,1)=X1(1)/X1(NA+1)**2*X1(I+NA+1)
 113  H2(I,I+1)=-X1(1)/X1(NA+1)
      NA2=NA
  19  CONTINUE
      DO 20 I=1,NC
      DO 20 J=1,NA2
  20  DYDX(I,J)=H1(I,J)
      DO 22 I=1,NC
      DO 22 J=1,NB
  22  DYDX(I,J+NA2)=H2(I,J)
      DO 24 I=1,NB
      DO 24 J=1,NA2
  24  DYDX(I+NC,J)=D(I,J)
      DO 26 I=1,NB
      DO 26 J=1,NB
  26  DYDX(I+NC,J+NA2)=E(I,J)
      N4=NB+NC
      N5=NA2+NB
      DO 30 I=1,N4
      DO 30 J=1,N5
      S(I,J)=0.
      DO 30 K=1,N5
  30  S(I,J)=S(I,J)+DYDX(I,K)*PX1(K,J)
      DO 32 I=1,N4
      DO 32 J=1,N4
      PX2(I,J)=0.
      DO 32 K=1,N5
  32  PX2(I,J)=PX2(I,J)+S(I,K)*DYDX(J,K)
      IF(IPR.LE.1) GO TO 33
C
C     PRINT RESULTS.
C
      PRINT 2900
 2900 FORMAT(/10X,'VECTOR C')
      PRINT 3000,(C(I),I=1,NC)
 3000 FORMAT(7X,10G12.5)
      PRINT 2950
 2950 FORMAT(/10X,'COVARIANCE MATRIX')
      DO 3010 I=1,NC
 3010 PRINT 3000,(PX2(I,J),J=1,NC)
  33  CONTINUE
      DO 35 I=1,NC
```

```
      DO 35 J=1,NC
 35   PC(I,J)=PX2(I,J)
      EPS=10.**-07*PC(1,1)

      COMPUTE TEST QUANTITY.

      CALL DESYM(PC,V,NC,EPS,IERR,IA2)
      IF(IERR.EG.-1)GO TO 99
      CALL SOLVS(V,C,Y,NC,1,IA2)
      TT=0.
      DO 36 I=1,NC
 36   TT=TT+Y(I)*C(I)
      IF(IPR.GT.0) PRINT 101,TT,NC
101   FORMAT(/10X,'TEST QUANTITY',G10.5,10X,'DEGREES OF FREEDOM',I5/10X,
     *23(1H-))
      TNC=FLOAT(NC)
      IF(TT.GT.TJITWO(TNC)) GO TO 60
      DO 42 I=1,NB
 42   DIVI(I)=X1(I+NA2)

      COMPUTE THE NEW A-POLYNOMIAL.

      IK=1
      CALL PDV(DVNDA,DIVI,R,NA1,NB,IDR,IPR,IERR,IK)
      DO 44 I=1,IDR
      G21=G21+R(I)
 44   X2(I)=R(I)
C
C     COMPUTE THE NEW B-POLYNOMIAL.
C
      IK=2
      CALL PDV(DVNDB,DIVI,R1,NB1,NB,IDR1,IPR,IERR,IK)
      DO 46 I=1,IDR1
      G22=G22+R1(I)
 46   X2(I+IDR)=R1(I)
      NMA=IDR
      NMB=IDR1
      G32=G22/G21
      IF(IPR.GT.0)PRINT 601,G32
      GO TO 70
C
C     TEST IS READY. THE POLYNOMIALS HAVE COMMON FACTORS.
C
 60   CONTINUE
      IF(NC.LE.1) GO TO 68
      DO 61 I=1,NB
 61   X1(I)=X1(I+NA2)
      DO 62 I=1,NC
 62   X1(I+NB)=C(I)
      DO 64 I=1,NB
      DO 64 J=1,NB
 64   PX1(I,J)=PX2(I+NC,J+NC)
      DO 65 I=1,NC
      DO 65 J=1,NC
 65   PX1(I+NB,J+NB)=PX2(I,J)
      DO 66 I=1,NB
      DO 66 J=1,NC
      PX1(I,J+NB)=PX2(J,I+NC)
 66   PX1(J+NB,I)=PX2(I+NC,J)
      NA=NB
      NB=NC
      NC=NC-1
      IF(IPR.LE.2) GO TO 18
      PRINT 260
```

```
260 FORMAT(/10X,'THE NEW ESTIMATED VECTOR T')
    N7=NA+NB
    PRINT 3000,(X1(I),I=1,N7)
    PRINT 262
262 FORMAT(/10X,'COVARIANCE MATRIX')
    DO 72 I=1,N7
 72 PRINT 3000,(PX1(I,J),J=1,N7)
    GO TO 18


 68 IF(IPR.GT.0) PRINT 160
160 FORMAT(/10X,'NO COMMON FACTORS')
    NM=NA1+NB1
    DO 69 I=1,NM
 69 X2(I)=X1(I)
    NMA=NA1
    NMB=NB1
 70 GO TO 1000


 99 IF(IPR.GT.0) PRINT 199
199 FORMAT(/10X,'DECOMPOSITION IS IMPOSSIBLE')
    NM=NA1+NB1
    DO 79 I=1,NM
 79 X2(I)=X1(I)
    NMA=NA1
    NMB=NB1
    RETURN
C
C
1000 CONTINUE
    RETURN
    END
```

```
      SUBROUTINE EUKL2(X1,PX1,X2,TT,NA1,NB1,NMA,NMB,NC,IPR,IERR,IA1,IB)
C
C     THIS SUBROUTINE COMPUTES COMMON FACTORS TO TWO GIVEN POLYNOMIALS BY
C     USING THE EUCLIDEAN ALGORITHM. THE DIVISOR-POLYNOMIAL IS NOT NORMA-
C     LIZED, I.E. THE SUCCECIVE DIVISIONS ARE DONE STRAIGHTFORWARD.
C     TO DECIDE WHETHER TWO POLYNOMIALS HAVE COMMON FACTORS OR NOT,
C     THE REST-POLYNOMIALS COMPUTED AT EACH DIVISION IS TESTED TO BE ZERO
C     ON THE ASSUMPTION THAT THE COEFFICIENTS OF THE ORIGINAL POLYNOMIALS
C     ARE GAUSSIAN. IF THE ORIGINAL POLYNOMIALS HAVE COMMON FACTORS, THEY
C     ARE ABBREVIATED WITH THE GREATEST COMMON POLYNOMIAL.
C     AUTHOR ERIK BURSTRÖM 1972-12-24
C
C     X1=(1,A(1),....,A(NA),B(0),B(1),....,B(NB)) VECTOR OF ORDER (NA1+NB1)
C     CONTAINING THE COEFFICIENTS OF THE ORIGINAL POLYNOMIALS. IF B(0)=0,
C     B(0) IS OMITTED.
C     PX1 MATRIX OF ORDER (NA1+NB1)*(NA1+NB1) THE COVARIANCE MATRIX
C     OF X1.
C     X2 VECTOR OF ORDER (NMA+NMB) CONTAINING THE COEFFICIENTS OF THE
C     NEW ESTIMATED A- RESP. B-POLYNOMIALS.
C     TT STATISTICAL TEST QUANTITY.
C     NA1 NUMBER OF A-COEFFICIENTS (MIN 2, MAX 10).
C     NB1 NUMBER OF B-COEFFICIENTS (MIN 1, MAX 11).
C     NMA NUMBER OF NEW A-COEFFICIENTS.
C     NMB NUMBER OF NEW B-COEFFICIENTS.
C     NC NUMBER OF DEGREES OF FREEDOM.
C     IPR   IF IPR=0 NOTHING IS PRINTED.
C           IF IPR=1 THE STATIC GAIN FOR THE ORIGINAL SYSTEM, THE TEST QUANTITY
C            AND CORRESPONDING NUMBER OF DEGREES OF FREEDOM AT EACH DIVISION
C            DOWN TO THE FIRST POSITIVE TEST, THE NEW A-POLYNOMIAL, DEGREE OF A
C            AND POLES, THE NEW B-POLYNOMIAL, DEGREE OF B AND ZEROS AND
C            THE STATIC GAIN FOR THE NEW SYSTEM ARE PRINTED.
C           IF IPR=2 AS IPR=1 + THE REST-POLYNOMIAL AND CORRESPONDING
C            COVARIANCE MATRIX OBTAINED AT EACH DIVISION AND THE TWO REST-POLY-
C            NOMIALS OBTAINED AT THE FINAL ABBREVIATION ARE PRINTED.
C     IERR IF IERR=0 NORMAL OUTPUT.
C          IF IERR=-1 EITHER IS THE COVARIANCE MATRIX PX1 NOT POSITIVE
C          DEFINITE OR IS IT IMPOSSIBLE TO COMPUTE POLES OR ZEROS
C          FOR THE NEW SYSTEM.
C     IA1 DIMENSION PARAMETER OF X1 AND PX1
C     IB IF IB =0 B(0)=0 IS ASSUMED.
C        IF IB=1 B(0)=1 IS ASSUMED.
C
C     SUBROUTINE REQUIRED
C           PDV
C           DESYM
C           SOLVS
C           ROT
C
      DIMENSION X1(1),PX1(IA1,IA1),X2(1)
C
      DIMENSION C(10),PC(10,10),Y(10),H1(11,11),H2(11,11),D(11,11),E(11,
     *11),DYDX(22,22),V(10,10),R(10),DVNDA(11),DVNDB(11),DIVI(11),S(21,
     *21),PX2(21,21),R1(10)
      REAL K1,K2,K3
      TJITWO(TN)=1.538*TN+3.2
      IA2=10
      IERR=0
      G=0.
      G1=0.
      G2=0.
      G20=0.
      G21=1.
      IF(IB.EQ.0)G22=0.
```

```
      IF(IB.EQ.1)G22=1.
      NA=NA1
      NB=NB1
      NC=NB1-1
      NC1=NB1-1
      IF(IB.EQ.0.AND.NB1.EQ.1)GO TO 68
      IF(NC.LE.0) RETURN
      IF(IPR.GT.0) PRINT 600,X1(NA+1)
  600 FORMAT(/10X,'VERSION 2  THE B-POLYNOMIAL IS UNSCALED  B(0)= ',G10.
     *5/10X,5(2H- ))

      COMPUTE THE STATIC GAIN FOR THE ORIGINAL SYSTEM.

      DO 3 I=1,NA1
      G1=G1+X1(I)
    3 DVNDA(I)=X1(I)
      DO 4 I=1,NB1
      G2=G2+X1(I+NA1)
    4 DVNDB(I)=X1(I+NA1)
      G=G2/G1
      IF(IPR.GT.0)PRINT 601,G
  601 FORMAT(/10X,'STATIC GAIN',5X,G10.5)
      DO 5 I=1,11
      DO 5 J=1,11
    5 D(I,J)=0.
      DO 6 I=1,22
      DO 6 J=1,22
    6 DYDX(I,J)=0.
      DO 7 I=1,10
    7 R(I)=0.
      DO 8 I=1,11
      DO 8 J=1,11
      E(I,J)=0.
      IF(I.NE.J) GO TO 8
      E(I,J)=1.
    8 CONTINUE
   14 CONTINUE
      IF(IB.EQ.1.AND.NA.EQ.NB) GO TO 108
C
C     COMPUTE THE REST-POLYNOMIAL AND CORRESPONDING COVARIANCE MATRIX.
C
      K1=-X1(NA+1)**2
      K2=X1(2)*X1(NA+1)-X1(1)*X1(NA+2)
      K3=X1(1)*X1(NA+1)
      IF(NC.LE.1) GO TO 51
      NC1=NC-1
      DO 11 I=1,NC1
   11 C(I)=K1*X1(I+2)+K2*X1(I+NA+1)+K3*X1(I+NA+2)
      C(NC)=K1*X1(NC+2)+K2*X1(NA+NB)
      GO TO 27
   51 C(1)=K1*X1(3)+K2*X1(5)
      H1(1,1)=-X1(NA+NB)**2
      H1(1,2)=X1(NA+1)*X1(NA+2)
      H1(1,3)=K1
      H2(1,1)=-2.*X1(NA+1)*X1(3)+X1(2)*X1(NA+2)
      H2(1,2)=-X1(1)*X1(NA+2)+X1(2)*X1(NA+1)-X1(1)*X1(NA+2)
      GO TO 115
   27 CONTINUE
C
C
      NA2=NA
      DO 16 I=1,NC
      DO 15 J=1,NA2
   15 H1(I,J)=0.
```

```
      DO 16 J=1,NB
   16 H2(I,J)=0.
      NC1=NC-1
      DO 17 I=1,NC1
      H1(I,1)=X1(NA+1)*X1(I+NA+2)-X1(NA+2)*X1(I+NA+1)
   17 H1(I,2)=X1(NA+1)*X1(I+NA+1)
      H1(NC,1)=-X1(NA+2)*X1(NA+NB)
      H1(NC,2)=X1(NA+1)*X1(NA+NB)
      DO 19 I=1,NC
   19 H1(I,I+2)=-X1(NA+1)**2


      DO 21 I=1,NC1
      H2(I,I+2)=X1(1)*X1(NA+1)
   21 H2(I,1)=-2.*X1(NA+1)*X1(I+2)+X1(1)*X1(I+NA+2)+X1(2)*X1(I+NA+1)
      H2(NC,1)=-2.*X1(NA+1)*X1(NA)+X1(2)*X1(NA+NB)
      H2(1,2)=-X1(1)*X1(NA+2)+K2
      DO 22 I=2,NC
      H2(I,I+1)=-X1(1)*X1(NA+2)
   22 H2(I,2)=-X1(1)*X1(I+NA+1)
      GO TO 26
  108 CONTINUE
      K1=X1(NA+1)
      K2=-X1(1)
      DO 111 I=1,NC
  111 C(I)=K1*X1(I+1)+K2*X1(I+NA+1)
      DO 112 I=1,NC
      DO 112 J=1,NA
      H1(I,J)=0.
  112 H2(I,J)=0.
      DO 114 I=1,NC
      H1(I,1)=-X1(I+NA+1)
      H2(I,1)=X1(I+1)
      H1(I,I+1)=K1
  114 H2(I,I+1)=K2
  115 CONTINUE
   26 CONTINUE
      NA2=NA
      DO 31 I=1,NC
      DO 31 J=1,NA2
   31 DYDX(I,J)=H1(I,J)
      DO 32 I=1,NC
      DO 32 J=1,NB
   32 DYDX(I,J+NA2)=H2(I,J)
      DO 33 I=1,NB
      DO 33 J=1,NA2
   33 DYDX(I+NC,J)=D(I,J)
      DO 34 I=1,NB
      DO 34 J=1,NB
   34 DYDX(I+NC,J+NA2)=E(I,J)
   35 CONTINUE
      I5=NB+NC
      I6=NA+NB
      DO 36 I=1,I5
      DO 36 J=1,I6
      S(I,J)=0.
      DO 36 K=1,I6
   36 S(I,J)=S(I,J)+DYDX(I,K)*PX1(K,J)
      DO 38 I=1,I5
      DO 38 J=1,I5
      PX2(I,J)=0.
      DO 38 K=1,I6
   38 PX2(I,J)=PX2(I,J)+S(I,K)*DYDX(J,K)
      DO 40 I=1,NC
```

```
      DO 40 J=1,NC
   40 PC(I,J)=PX2(I,J)
      IF(IPR.LE.1) GO TO 39
C
C     PRINT RESULTS.
C
      PRINT 1900
 1900 FORMAT(/10X,'VECTOR C')
      PRINT 2000,(C(I),I=1,NC)
 2000 FORMAT(7X,10G12.5)
      PRINT 1950
 1950 FORMAT(/10X,'COVARIANCE MATRIX')
      DO 41 I=1,NC
   41 PRINT 2000,(PC(I,J),J=1,NC)
   39 CONTINUE
C
C     COMPUET TEST QUANTITY.
C
      EPS=10.**-07*PC(1,1)
      CALL DESYM(PC,V,NC,EPS,IRANK,IA2)
      IF(IRANK.EQ.-1) GO TO 99
      CALL SOLVS(V,C,Y,NC,1,IA2)
      TT=0.
      DO 42 I=1,NC
   42 TT=TT+Y(I)*C(I)
      IF(IPR.GT.0) PRINT 101,TT,NC
  101 FORMAT(/10X,'TEST QUANTITY',G10.5,10X,'DEGREES OF FREEDOM',I5/10X,
     *23(1H-))
      TNC=FLOAT(NC)
      IF(TT.GT.TJITWO(TNC)) GO TO 60
      DO 44 I=1,NB
   44 DIVI(I)=X1(I+NA)
C
C     COMPUTE THE NEW A-POLYNOMIAL.
C
      IK=1
      CALL PDV(DVNDA,DIVI,R,NA1,NB,IDR,IPR,IERR,IK)
      DO 45 I=1,IDR
      G21=G21+R(I)
   45 X2(I)=R(I)
C
C     COMPUTE THE NEW B-POLYNOMIAL.
C
      IK=2
      CALL PDV(DVNDB,DIVI,R1,NB1,NB,IDR1,IPR,IERR,IK)
      DO 46 I=1,IDR1
      G22=G22+R1(I)
   46 X2(I+IDR)=R1(I)
      NMA=IDR
      NMB=IDR1
C
C     COMPUTE THE STATIC GAIN FOR THE NEW SYSTEM.
C
      G20=G22/G21
      IF(IPR.GT.0)PRINT 601,G20
      GO TO 70
C
C     TEST IS READY. THE POLYNOMIALS HAVE COMMON FACTORS.
C
   60 CONTINUE
      IF(NC.LE.1) GO TO 68
      DO 61 I=1,NB
   61 X1(I)=X1(I+NA)
      DO 62 I=1,NC
```

```
 62 X1(I+NB)=C(I)
    DO 64 I=1,NB
    DO 64 J=1,NB
 64 PX1(I,J)=PX2(I+NC,J+NC)
    DO 65 I=1,NC
    DO 65 J=1,NC
 65 PX1(I+NB,J+NB)=PX2(I,J)
    DO 66 I=1,NB
    DO 66 J=1,NC
    PX1(I,J+NB)=PX2(J,I+NC)
 66 PX1(J+NB,I)=PX2(I+NC,J)
    NA=NB
    NB=NC
    NC=NC-1
    IF(IPR.LE.2) GO TO 14
    PRINT 260
260 FORMAT(/10X,'THE NEW ESTIMATED VECTOR T')
    N7=NA+NB
    PRINT 2000,(X1(I),I=1,N7)
    PRINT 262
262 FORMAT(/10X,'THE COVARIANCE MATRIX')
    DO 72 I=1,N7
 72 PRINT 2000,(PX1(I,J),J=1,N7)
    GO TO 14
C
C
 68 IF(IPR.GT.0) PRINT 160
160 FORMAT(/10X,'NO COMMON FACTORS')
    NM=NA1+NB1
    DO 69 I=1,NM
 69 X2(I)=X1(I)
    NMA=NA1
    NMB=NB1
 70 GO TO 100
C
C
 99 IF(IPR.GT.0) PRINT 199
199 FORMAT(/10X,'DECOMPOSITION IS IMPOSSIBLE')
    NM=NA1+NB1
    DO 79 I=1,NM
 79 X2(I)=X1(I)
    NMA=NA1
    NMB=NB1
C
C
100 RETURN
    END
```

TFRT - 7056

FÖRKORTNING AV GEMENSAMMA FAKTORI SKATTADE ÖVERFÖRINGSFUNKTIONER

KORT BESKRIVNING AV PROGRAMVARAN

E. BURSTRÖM

Rapport 7320(C) Juli 1973
Inst. för Reglerteknik
Lunds Tekniska Högskola

FÖRKORTNING AV GEMENSAMMA FAKTORER I SKATTADE ÖVERFÖRINGSFUNKTIONER.

KORT BESKRIVNING AV PROGRAMVARAN.

E. Burström

INNEHÅLLSFÖRTECKNING

1. INLEDNING.

Problemet att avgöra om skattade överföringsfunktioner har gemensamma faktorer i statistisk mening har undersökts med två olika metoder.

Den första metoden går ut på att beräkna systemets poler och nollställen och en tillhörande kovariansmatris, och därefter görs statistiska hypotestester på så sätt att man testar om vissa kombinationer av poler och nollställen kan anses vara gemensamma. Metoden kallas i fortsättningen TPOL.

Den andra metoden använder Euklides algoritm för polynom, d.v.s. de givna polynomen divideras med varandra till dess att restpolynomet är noll i statistisk mening.

Förutsättningar och analytisk behandling finns beskrivna i [1]. I fortsättningen ges endast en kort beskrivning av programvaran, som realiserar de två metoderna.

Exempel på resultat vid exekvering av programmen ges i [1].

2. PROGRAMVARAN FÖR TPOL.

Problemet löses i det här fallet enl. nedanstående figur.

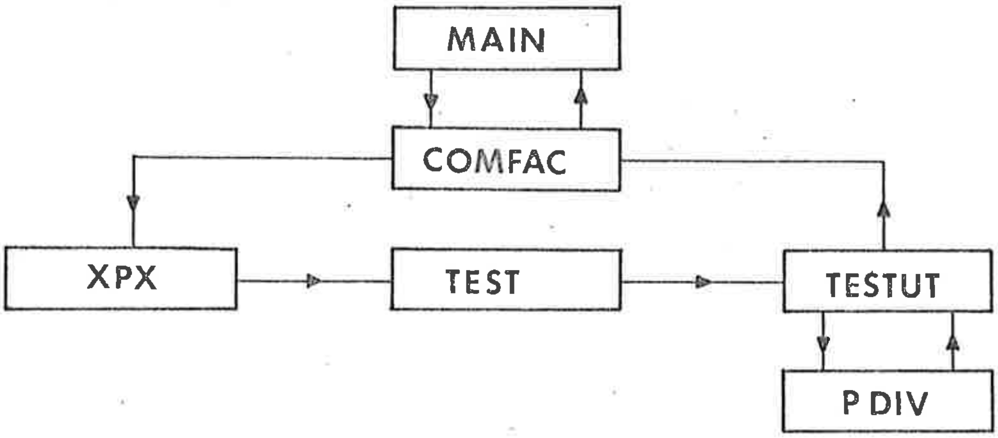

Fig. 2.1.

MAIN. Användarens eget huvudprogram. Här bildas vektorn T, kovariansmatrisen P_T och avgörs om B- eller C-fallet är för handen.

COMFAC. Den överordnade subrutinen som administrerar hela analysen.

XPX. Här beräknas dels vektorn X, vars komponenter utgörs av överföringsfunktionens poler och nollställen, dels matrisen PX, kovariansmatrisen för X. Komplexa rötter separeras i real- och imaginärdel i vektorn X.

TEST. Här görs individuella hypotestester. Varje pol testas individuellt mot varje nollställe, och resultatet, d.v.s. de tester som accepteras, ges i en tabell IQ.

TESTUT. Här görs flervariabla tester. På grundval av resultaten i subrutinen TEST, bestämmes den kombination av så stort antal poler och nollställen som möjligt, som ger en positiv testkvantitet. Om mer än en kombination av samma antal faktorer är möjlig väljs den, som ger lägst testkvantitet. Slutligen beräknas ett nytt estimat av vektorn X under förutsättning att vissa faktorer är gemensamma, d.v.s. lika.

PDIV. Här divideras de givna polynomen med det gemensamma polynomet, som beräknas ur det nya estimatet av vektorn X, som beräknats i subrutinen TESTUT.

En fullständig programlista återfinns i appendix.

3. PROGRAMVARAN FÖR EUKLIDES ALGORITM.

Problemet löses här enligt figur nedan.

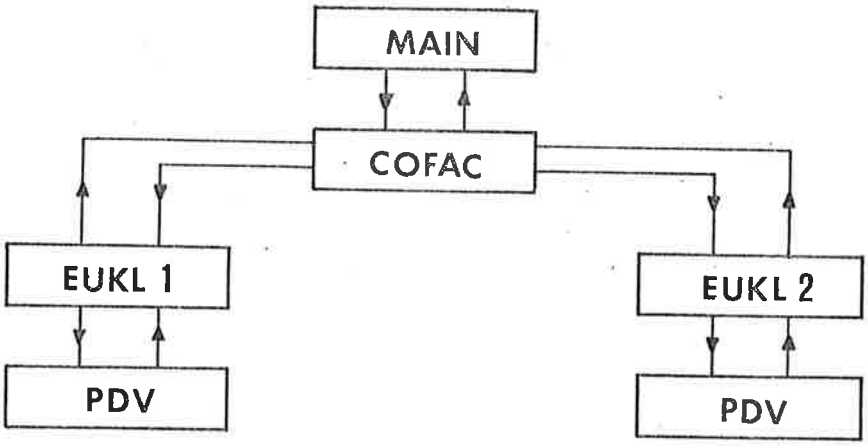

Fig. 3.1.

MAIN. Användarens eget huvudprogram där vektorn T + kovariansmatrisen P_T bildas och avgörs om B- eller C-fallet föreligger.

COFAC. Den överordnade subrutinen. Här modifieras T och P_T för att den fortsatta behandlingen skall bli enhetligare. Därefter anropas EUKL1 och EUKL2, som använder Euklides algoritm i de båda varianterna, se [1].

EUKL1. Här används Euklides algoritm i version 1, vilket innebär att vid varje division skalas de båda ingående polynomen så att högstagradskoefficienterna blir 1. Restpolynomet som erhålles vid de successiva divisionerna testas om det är noll i statistisk mening eller inte. Om så är fallet anropas subrutinen PDV, se nedan, annars fortsätter divisionen på känt sätt.

EUKL2. Här används Euklides algoritm i version 2. Den enda skillnaden mot version 1 är att vid de successiva divisionerna skalas inte polynomen utan divisionerna sker på vanligt sätt, s.k. icke-normaliserad division.

PDV. Här divideras de gemensamma faktorerna bort från de givna polynomen. Divisionen sker på precis samma sätt som i subrutinen PDIV, se kapitel 2.

En fullständig lista återfinns i appendix.

4. REFERENSER.

[1] E. Burström: Förkortning av gemensamma faktorer i skattade överföringsfunktioner. Examensarbete RE 124, 1973, Inst. för Reglerteknik, LTH, Lund.

```
SUBROUTINE COMFAC(T,PT,AM,BM,TT,NA,NB,NMA,NMB,IPR,IERR,IA,IB)

THE SUBROUTINE COMPUTES COMMON FACTORS TO TWO GIVEN POLYNOMIALS, AND
THE GIVEN POLYNOMIALS ARE ABBREVIATED WITH THE COMMON POLYNOMIAL.
AN HYPOTHESIS TEST IS USED IN ORDER TO DECIDE WHETHER THE TWO GIVEN
POLYNOMIALS HAVE COMMON FACTORS OR NOT. THE COEFFICIENTS OF THE TWO
GIVEN POLYNOMIALS SHOULD BE NORMALLY DISTRIBUTED FOR OBTAINING
BEST RESULTS, BECAUSE A CHI-SQUARE TEST WITH THE SIGNIFICANCE LEVEL
OF 5% IS USED.
AUTHOR ERIK BURSTRÖM 1972-12-24

T=(A(1),...,A(NA),B(1),...,B(NB)) VECTOR OF ORDER (NA+NB), (MIN 2, MAX 20
CONTAINING THE COEFFICIENTS OF THE A- RESP. B-POLYNOMIAL.
PT MATRIX OF ORDER (NA+NB)*(NA+NB) THE COVARIANCE MATRIX OF T.
AM VECTOR OF ORDER NMA AT OUTPUT CONTAINING THE NEW ESTIMATED A-COEFFI-
CIENTS.
BM VECTOR OF ORDER NMB AT OUTPUT CONTAINING THE NEW ESTIMATRD B-COEFFI-
CIENTS.
TT STASTISTICAL TEST QUANTITY.
NA NUMBER OF A-COEFFICIENTS (MIN 1, MAX 10).
NB NUMBER OF B-COEFFICIENTS (MIN 1, MAX 10).
NMA NUMBER OF AM-COEFFICIENTS.
NMB NUMBER OF BM-COEFFICIENTS.
IPR  IF IPR=0 NOTHING IS PRINTED.
     IF IPR=1 VECTOR T, POLES, ZEROS AND STATIC GAIN FOR THE ORIGINAL
      SYSTEM, VECTOR X CONSISTING OF POLES AND ZEROS, ALL POSITIVE
      INDIVIDUELL TESTS WHEN TWO FACTORS ARE EQUAL AND CORRESPONDING
      TEST QUANTITIES, MATRIX IQ WHERE EACH ROW MEANS A POSSIBLE
      COMBINATION FOR TWP FACTORS TO BE EQUAL, THE PART OF MATRIX IQ
      THAT REPRESENTS THE BEST MULTIVARIATE TEST, I.E. THE HIGHEST
      NUMBER OF DEGREES OF FREEDOM AND WITHIN THAT NUMBER THE TEST
      WITH SMALLEST TEST QUANTITY, A NEW ESTIMATED VECTOR X, THE NEW A-
      POLYNOMIAL, DEGREE OF A AND POLES, THE NEW B-POLYNOMIAL, DEGREE
      OF B AND ZEROS AND THE STATIC GAIN FOR THE NEW SYSTEM ARE PRINTED.
     IF IPR=2 AS IPR=1 + THE COVARIANCE MATRIX OF T AND THE APPROXIMATE
      COVARIANCE MATRIX OF X, THE COMMON POLYNOMIAL AND THE TWO REST-
      POLYNOMIALS ARE PRINTED.
     IF IPR=3 AS IPR=2 + ALL NEGATIVE INDIVIDUELL TESTS AND ALL MULTI-
      VARIATE TESTS, I.E. THE COMBINATIONS OF FACTORS AND TEST QUANTITIES
      WITH HIGHER AND THE SAME DEGREE OF  FREEDOM AS THE BEST POSITIVE
      TEST ARE PRINTED.
     IF IPR=4 AS IPR=3 + ALL MULTIVARIATE TESTS WITH LOWER DEGREES OF
      FREEDOM THAN THE BEST POSITIVE TEST ARE PRINTED.
IERR IF IERR=0 NORMAL OUTPUT.
     IF IERR=-1 EITHER THE COVARIANCE MATRIX OF X IS NOT POSITIVE DEFINIT
     OR IS IT IMPOSSIBLE TO COMPUTE POLES OR ZEROS TO THE ORIGINAL
     OR THE NEW SYSTEM. IN THE FIRST CASE, THE INTERESTING EIGNVALUES
     AND CORRESPONDING EIGENVECTORS OF PT ARE COMPUTED AND PRINTED.
IA DIMENSION PARAMETER OF PT.
IB IF IB=0 B(0)=0 IS ASSUMED.
   IF IB=1 B(0)=1 IS ASSUMED.

THE FOLLOWING VARIABLE LIES IN A COMMON-BLOCK CALLED COMPOL.

D VECTOR OF ORDER 10 CONTAINING THE COEFFICIENTS OF THE COMMON
POLYNOMIAL.

SUBROUTINE REQUIRED
     XPX
     TEST
     TESTUT
     PDIV
     ROT
```

```
      DESYM
      SOLVS
      EIGS
      PMPY

    DIMENSION T(1),PT(IA,IA),AM(1),BM(1)
    COMMON/COMPOL/D(10)

    DIMENSION X(20),PX(20,20),IQ(100,4)

    IERR=0
    N=NA+NB
    IC=100
    IF(IPR.LE.0)GO TO 5
    PRINT 101
101 FORMAT(1H1,9X,'PRINTOUT FROM COMFAC'/10X,20(1H*))
    PRINT 103
103 FORMAT(/10X,'VECTOR T')
    PRINT 102,(T(I),I=1,N)
102 FORMAT(8X,10G12.5)
  5 CONTINUE

    COMPUTE POLES, ZEROS, STATIC GAIN AND COVARIANCE MATRIX PX FOR THE
    ORIGINAL SYSTEM.

    CALL XPX(T,PT,X,PX,NA,NB,NCA,NCB,IPR,IERR,IA,IB)
    IF(IERR.EQ.-1)RETURN

    COMPUTE ALL INDIVIDUELL TESTS AND MATRIX IQ

    CALL TEST(X,PX,IQ,NA,NB,NCA,NCB,NQ,IPR,IERR,IA,IB,IC)
    IF(IERR.EQ.-1)RETURN

    COMPUTE MULTIVARIATE TESTS, THE NEW A- AND B-POLYNOMIALS AND STATIC
    GAIN FOR THE NEW SYSTEM

    CALL TESTUT(T,X,PX,AM,BM,TT,IQ,NA,NB,NMA,NMB,NQ,IPR,IERR,IA,IB,IC)
    RETURN
    END
```

```
      SUBROUTINE XPX(T,PT,X,PX,NA,NB,NCA,NCB,IPR,IERR,IA,IB)

      SUBROUTINE FOR TRANSFORMING T-VECTOR TO X-VECTOR AND COVARIANCE MATRIX
      PT TO THE APPROXIMATE COVARIANCE MATRIX PX OF VECTOR X.
      NO MULTIPLE POLES OR ZEROS IS ASSUMED.
      THE ELEMENTS OF VECTOR T MUST BE REAL.
      COMPLEX FACTORS ARE SEPARATED INTO REAL AND IMAGINARY PARTS.
      AUTHOR TORSTEN SÖDERSTRÖM 1971-12-24
      REVISED ERIK BURSTRÖM 1972-11-01

      T=(A(1),...,A(NA),B(1),...,B(NB)) VECTOR OF ORDER (NA+NB).
      PT MATRIX OF ORDER (NA+NB)*(NA+NB), THE COVARIANCE MATRIX OF T.
      X VECTOR OF ORDER (NA+NB). THE ELEMENTS OF X ARE NCA/2 REAL PARTS OF
      COMPLEX POLES, NCA/2 POSITIVE IMAGINARY PARTS OF COMPLEX POLES, NA-NCA
      REAL POLES, NCB/2 REAL PARTS OF COMPLEX ZEROS, NCB/2 POSITIVE IMAGINARY
      PARTS OF COMPLEX ZEROS, NB-NCB REAL ZEROS. IF IB=0 X(NA+NB)=T(NA+1).
      PX MATRIX OF ORDER (NA+NB)*(NA+NB) AT RETURN CONTAINING THE APPROXIMATE
      COVARIANCE MATRIX OF X.
      NA NUMBER OF A-COEFFICIENTS (MIN 1, MAX 10).
      NB NUMBER OF B-COEFFICIENTS (MIN 1, MAX 10).
      NCA NUMBER OF COMPLEXVALUED POLES (MIN 0, MAX 10).
      NCB NUMBER OF COMPLEXVALUED ZEROS (MIN 0, MAX 10).
      IPR  IF IPR=0 NOTHING IS PRINTED.
           IF IPR=1 VECTOR T, THE POLES, THE NUMBER OF A-COEFFICIENTS, THE
            NUMBER OF COMPLEX RESP. REAL POLES,THE ZEROS, THE NUMBER OF
            B-COEFFICIENTS, THE NUMBER OF COMPLEX RESP. REAL ZEROS, VECTOR X
            AND STATIC GAIN ARE PRINTED.
           IF IPR=2 AS IPR=1 + THE COVARIANCE MATRIX OF X.
      IERR IF IERR =0 NORMAL RETURN
           IF IERR=-1 ROT HAS FAILED.
      IA DIMENSION PARAMETER OF PT AND PX.
      IB IF IB=0 B(0)=0 IS ASSUMED.
         IF IB=1 B(0)=1 IS ASSUMED.

      SUBROUTINE REQUIRED
           ROT

      DIMENSION T(1),PT(IA,IA),X(1),PX(IA,IA)

      DIMENSION A(10),AR(10),AI(10),B(10),BR(10),BI(10),DXDT(20,20),
     *S(20,20),LINE(22)
      COMPLEX Z(10),ZZ,ZM(10,10)
      DATA STRECK/5H-----/

      N=NA+NB
      NERRA=0
      NERRB0=0
      NERRB1=0
      IERR=0
      IF(IB.EQ.0)G1=T(NA+1)
      IF(IB.EQ.1)G1=1.
      G2=1.
      G=0.
      EPS=1.0E-05
      KCA=0
      KCB=0

      COMPUTE THE POLES, ZEROS AND STATIC GAIN.

      DO 1 I=1,NA
      G2=G2+T(I)
    1 A(I)=T(I)
      CALL ROT(NA,1,A,A,AR,AI,NERRA)
```

```
      IF (NERRA.GT.0) GO TO 8
      DO 54 I=1,NA
   54 IF(ABS(AI(I)).GT.EPS)NCA=NCA+1
      IF (IB.EQ.1) GO TO 51
      NN=NB-1
      IF (NB.EQ.1) GO TO 3
      DO 2 I=1,NN
      G1=G1+T(I+NA+1)
    2 B(I)=T(I+NA+1)/T(NA+1)
      CALL ROT (NN,1,B,B,BR,BI,NERRB0)
      IF (NERRB0.GT.0) GO TO 8
      DO 55 I=1,NN
   55 IF(ABS(BI(I)).GT.EPS)NCB=NCB+1
    3 AK=T(NA+1)
      IF(IPR.EQ.0) GO TO 49
      PRINT 101,AK
  101 FORMAT(/56X,'B(0)=0.',3X,'B(1)= ',G10.5/10X,'ROOTS OF A',36X,'ROOT
     *S OF B'/)
      GO TO 53
   51 DO 52 I=1,NB
      G1=G1+T(I+NA)
   52 B(I)=T(I+NA)
      NN=NB
      CALL ROT (NN,1,B,B,BR,BI,NERRB1)
      IF (NERRB1.GT.0) GO TO 8
      DO 56 I=1,NB
   56 IF(ABS(BI(I)).GT.EPS)NCB=NCB+1
      IF(IPR.EQ.0) GO TO 49
      PRINT 201
  201 FORMAT(/56X,'B(0)=1.'/10X,'ROOTS OF A',36X,'ROOTS OF B'/)
   53 CONTINUE
      N0=MIN0(NA,NN)
      NM=N0+1
      NL=MAX0(NA,NN)
      IF (IB.EQ.0.AND.NB.EQ.1) GO TO 5
      DO 4 I=1,N0
    4 PRINT 102,AR(I),AI(I),BR(I),BI(I)
  102 FORMAT(10X,2G15.7,16X,2G15.7)
    5 IF (NA.EQ.NN) GO TO 7
      IF (NA.GT.NN) GO TO 6
      PRINT 108,((BR(I),BI(I)),I=NM,NL)
  108 FORMAT(56X,2G15.7)
      GO TO 7
    6 PRINT 109,((AR(I),AI(I)),I=NM,NL)
  109 FORMAT(10X,2G15.7)
    7 CONTINUE
      NRA=NA-NCA
      NRB=NN-NCB
      PRINT 103,NA,NN,NCA,NCB,NRA,NRB
  103 FORMAT(/10X,I2,3X,'A-COEFFICIENTS',27X,I2,3X,'B-COEFFICIENTS'/10X,
     *I2,3X,'COMPLEX POLES',28X,I2,3X,'COMPLEX ZEROES'/10X,I2,3X,'REAL P
     *OLES',31X,I2,3X,'REAL ZEROES')
      G=G1/G2
      PRINT 104,G
  104 FORMAT(/10X,'STATIC GAIN',5X,G10.5)
   49 CONTINUE

      COMPUTATION OF THE JACOBIAN.

      DO 10 I=1,NA
      DO 10 J=1,NB
      DXDT(I,J+NA)=0.
   10 DXDT(J+NA,I)=0.
      DO 11 I=1,NA
```

```
      DO 11 J=1,NA
   11 ZM(I,J)=(0.,0.)
      DO 12 I=1,NA
   12 Z(I)=CMPLX(AR(I),AI(I))

      DO 15 I=1,NA
      ZZ=(1.,0.)
      DO 13 J=1,NA
      IF (J.EQ.I) GO TO 13
      ZZ=ZZ*(Z(I)-Z(J))
   13 CONTINUE
      DO 14 J=1,NA
   14 ZM(I,J)=-Z(I)**(NA-J)/ZZ
   15 CONTINUE
      IF (NCA.EQ.0) GO TO 17
      NC=NCA/2
      DO 16 I=1,NC
      DO 16 J=1,NA
      DXDT(2*I-1,J)=(ZM(2*I-1,J)+ZM(2*I,J))/2.
   16 DXDT(2*I  ,J)=(ZM(2*I-1,J)-ZM(2*I,J))/2./(0.,1.)
   17 IF (NCA.EQ.NA) GO TO 19
      I1=NCA+1
      DO 18 I=I1,NA
      DO 18 J=1,NA
   18 DXDT(I,J)=ZM(I,J)
   19 CONTINUE
      IF (IB.EQ.1) GO TO 60

      IF (NB.EQ.1) GO TO 30
      DO 20 I=1,NN
   20 Z(I)=CMPLX(BR(I),BI(I))
      DO 25 I=1,NN
      ZZ=(1.,0.)
      DO 21 J=1,NN
      IF (J.EQ.I) GO TO 21
      ZZ=ZZ*(Z(I)-Z(J))
   21 CONTINUE
      DO 22 J=1,NB
   22 ZM(I,J)=-Z(I)**(NN-J)/ZZ/AK
   25 CONTINUE
      DO 26 I=1,NN
      ZM(I,1)=(0.,0.)
      ZM(NB,I+1)=(0.,0.)
      DO 26 J=1,NN
   26 ZM(I,1)=ZM(I,1)-T(NA+1+J)/T(NA+1)*ZM(I,J+1)
      ZM(NB,1)=(1.,0.)
      IF (NCB.EQ.0) GO TO 28
      NC=NCB/2
      DO 27 I=1,NC
      DO 27 J=1,NB
      DXDT(NA+2*I-1,J+NA)=(ZM(2*I-1,J)+ZM(2*I,J))/2.
   27 DXDT(NA+2*I  ,J+NA)=(ZM(2*I-1,J)-ZM(2*I,J))/2./(0.,1.)
   28 IF (NCB.EQ.NN) GO TO 30
      I1=NCB+1
      DO 29 I=I1,NB
      DO 29 J=1,NB
   29 DXDT(I+NA,J+NA)=ZM(I,J)
   30 DXDT(N,N)=1.
      GO TO 31

   60 DO 62 I=1,NB
   62 Z(I)=CMPLX(BR(I),BI(I))
      DO 65 I=1,NB
      ZZ=(1.,0.)
```

```
      DO 63 J=1,NB
      IF (J.EQ.I) GO TO 63
      ZZ=ZZ*(Z(I)-Z(J))
   63 CONTINUE
      DO 64 J=1,NB
   64 ZM(I,J)=-Z(I)**(NB-J)/ZZ
   65 CONTINUE
      IF (NCB.EQ.0) GO TO 67
      NBC=NCB/2
      DO 66 I=1,NBC
      DO 66 J=1,NB
      DXDT(2*I-1+NA,J+NA)=(ZM(2*I-1,J)+ZM(2*I,J))/2.
   66 DXDT(2*I+NA,J+NA)=(ZM(2*I-1,J)-ZM(2*I,J))/2./(0.,1.)
   67 IF (NCB.EQ.NB) GO TO 69
      I1=NCB+1
      DO 68 I=I1,NB
      DO 68 J=1,NB
   68 DXDT(I+NA,J+NA)=ZM(I,J)
   69 CONTINUE
   31 CONTINUE

      THE COMPUTATION OF X AND PX.

      DO 32 I=1,N
      DO 32 J=1,N
      S(I,J)=0.
      DO 32 K=1,N
   32 S(I,J)=PT(I,K)*DXDT(K,J)+S(I,J)
      DO 33 I=1,N
      DO 33 J=1,N
      PX(I,J)=0.
      DO 33 K=1,N
   33 PX(I,J)=PX(I,J)+DXDT(K,I)*S(K,J)
      IF (NCA.EQ.0) GO TO 35
      NC=NCA/2
      DO 34 I=1,NC
      X(2*I-1)=AR(2*I-1)
   34 X(2*I  )=AI(2*I-1)
   35 IF (NCA.EQ.NA) GO TO 37
      NP=NCA+1
      DO 36 I=NP,NA
   36 X(I)=AR(I)
   37 IF (NCB.EQ.0) GO TO 39
      NC=NCB
      DO 38 I=1,NC
      X(NA+2*I-1)=BR(2*I-1)
   38 X(NA+2*I  )=BI(2*I-1)
   39 IF (NCB.EQ.NN) GO TO 41
      NP=NCB+1
      DO 40 I=NP,NN
   40 X(I+NA)=BR(I)
   41 CONTINUE
      IF(IB.EQ.0) X(N)=AK

      PRINT RESULTS.

      IF(IPR.EQ.0) GO TO 9
      PRINT 105
  105 FORMAT(/10X,'VECTOR X'/10X,8(1H-)/10X,'COMPONENT NUMBER')
      IF(IB.EQ.0) N=N-1
      IF(N.GT.10) GO TO 311
      PRINT 130,(I,I=1,N)
  130 FORMAT(2X,10I12)
      CALL ENCODE(LINE)
```

```
      CALL FMTX(11)
      DO 302 I=1,N
      CALL FMTA(STRECK,5)
302   CALL FMTX(7)
      CALL FMTX(121-N*12)
      PRINT 131,(LINE(I),I=1,22)
131   FORMAT(22A6)
      PRINT 106,(X(I),I=1,N)
106   FORMAT(8X,10G12.5)
      GO TO 314
311   CONTINUE
      PRINT 130,(I,I=1,10)
      CALL ENCODE(LINE)
      CALL FMTX(11)
      DO 312 I=1,10
      CALL FMTA(STRECK,5)
312   CALL FMTX(7)
      CALL FMTX(1)
      PRINT 131,(LINE(I),I=1,22)
      PRINT 106,(X(I),I=1,10)
      PRINT 130,(I,I=11,N)
      CALL ENCODE(LINE)
      CALL FMTX(11)
      DO 313 I=11,N
      CALL FMTA(STRECK,5)
313   CALL FMTX(7)
      CALL FMTX(121-(N-10)*12)
      PRINT 131,(LINE(I),I=1,22)
      PRINT 106,(X(I),I=11,N)
314   IF(IPR.LT.2)GO TO 9
      PRINT 212
212   FORMAT(/10X,'COVARIANCE MATRIX')
      DO 214 I=1,N
214   PRINT 106,(PX(I,J),J=1,N)
      GO TO 9
  8   IF(NERRA.GT.0)PRINT 210
210   FORMAT(/10X,'COMPUTATION OF POLES IS IMPOSSIBLE')
      IF(NERRB0.GT.0)PRINT 211
      IF(NERRB1.GT.0)PRINT 211
211   FORMAT(/10X,'COMPUTATION OF ZEROS IS IMPOSSIBLE')
      IERR=-1
  9   CONTINUE

      RETURN
      END
```

```
      SUBROUTINE TEST(X,PX,IQ,NA,NB,NCA,NCB,NQ,IPR,IERR,IA,IB,IC)

      SUBROUTINE FOR EXAMINING WHICH POLES AND WHICH ZEROS THAT CAN BE CON-
      SIDERED TO BE EQUAL IN STASTISTICAL SENSE.
      EVERY POLE IS TESTED AGAINST EVERY ZERO.
      THE TESTS ARE BASED ON THE ASSUMPTION THAT THE POLES AND THE ZEROS ARE
      NORMAL DISTRIBUTED I.E. A CHI-SQUARE TEST WITH THE SIGNIFICANCE LEVEL
      0.05 IS USED.
      THE RESULTS ARE GIVEN IN THE MATRIX IQ WHERE EVERY ROW MEANS A POSSIBLE
      COMBINATION FOR TWO FACTORS TO BE EQUAL.
      THE ELEMENTS IN A ROW MEANS FROM LEFT TO RIGHT:
      THE REAL PART OF THE POLE,
      THE REAL PART OF THE ZERO,
      THE IMAGINARY PART OF THE POLE,
      THE IMAGINARY PART OF THE ZERO.
      IF ANY OR BOTH ELEMENTS IN THE THIRD OR FORTH COLUMN IS ZERO, THE POLE
      RESP. THE ZERO IS REAL.
      THE NUMBER OF POSSIBLE COMBINATIONS, I.E. THE NUMBER OF ROWS IN THE MATRI
      IQ IS NQ.
      AUTHOR ERIK BURSTRÖM 1972-12-24

      X VECTOR OF ORDER (NA+NB), CONTAINING THE POLES AND THE ZEROS COMPUTED
      IN SUBROUTINE XPX.
      PX THE COVARIANCE MATRIX OF ORDER (NA+NB)*(NA+NB) COMPUTED
      IN SUBROUTINE XPX.
      IQ THE MATRIX OF ORDER (NC*4) CONTAINING ALL POSSIBLE TESTS.
      NA THE NUMBER OF A-COEFFICIENTS (MIN 1, MAX 10). A(0)=1 IS ASSUMED.
      NB THE NUMBER OF B-COEFFICIENTS(MIN 1, MAX 10).
      NCA THE NUMBER OF COMPLEXVALUED POLES COMPUTED IN SUBROUTINE XPX.
      NCB THE NUMBER OF COMPLEXVALUED ZEROS COMPUTED IN SUBROUTINE XPX.
      NQ THE NUMBER OF POSSIBLE INDIVIDUELL TESTS.
      IPR  IF IPR=0 NOTHING IS PRINTED.
           IF IPR=1 ALL POSITIVE TESTS, THE ESTIMATED COMMON VALUES, THE TEST
            QUANTITIES AND THE MATRIX IQ IS PRINTED.
           IF IPR=2 AS IPR=1 + ALL NEGATIVE TESTS.
      IERR IF IERR=0 NORMAL OUTPUT.
           IF IERR=-1 THE COVARIANCE MATRIX PX IS NOT POSITIVE DEFINITE. THE
           EIGENVALUES OF MATRIX PX ARE COMPUTED AND PRINTED.
      IA DIMENSION PARAMETER OF MATRIX PX.
      IB IF IB=0 B(0)=0 IS ASSUMED.
         IF IB=1 B(0)=1 IS ASSUMED.
      IC DIMENSION PARAMETER OF MATRIX IQ.

      SUBROUTINE REQUIRED
           DESYM
           SOLVS
           EIGS

      DIMENSION X(1),PX(IA,IA),IQ(IC,1)

      DIMENSION ST(20),IQQ(100,4),IND(10),S1(2,20),SP(20,2),SPS(2,2),
     *F(4),Y(2),V(2),R(2,2),EV(2),B(2,2)
      IPR1=IPR
      N=NA+NB-1
      IF (IB.EQ.1) N=NA+NB
      TQ=3.84
      TQ1=5.99
      NAP=NA-NCA
      NBP=NB-1-NCB
      IF (IB.EQ.1) NBP=NB-NCB
      NQ=0
      TT=0.
      ID=2
```

```
      EPS IS USED IN DESYM.

      EPS=1.0E-07
      IERR=0
      DO 17 I=1,100
      DO 17 J=1,4
 17   IQ(I,J)=0

      TEST OF COMPLEX FACTORS .

      IF (NCA*NCB.EQ.0) GO TO 20

      COMPUTE TEST VECTOR.

      DO 15 I=1,NCA,2
      DO 15 J=1,NCB,2
      TT=0.
      IR=I
      IR1=I+1
      JR=J+NA
      JR1=J+NA+1
      DO 5 K=1,N
      S1(1,K)=0.
  5   S1(2,K)=0.
      S1(1,IR)=1.
      S1(1,JR)=-1.
      S1(2,IR1)=1.
      S1(2,JR1)=-1.
      Y(1)=X(IR)-X(JR)
      Y(2)=X(IR1)-X(JR1)

      COMPUTE TEST QUANTITY.

      DO 8 K=1,N
      DO 8 L=1,2
  7   SP(K,L)=0.
      DO 8 M=1,N
  8   SP(K,L)=SP(K,L)+PX(K,M)*S1(L,M)
      DO 10 K=1,2
      DO 10 L=1,2
  9   SPS(K,L)=0.
      DO 10 M=1,N
 10   SPS(K,L)=SPS(K,L)+S1(K,M)*SP(M,L)
      EPS=10.**-07*SPS(1,1)
      CALL DESYM(SPS,B,2,EPS,IERR,ID)
      IF(IERR.EQ.-1)GO TO 150
      CALL SOLVS(B,Y,V,2,1,ID)
      TT=V(1)*Y(1)+V(2)*Y(2)
      IF(TT.GT.TQ1.AND.IPRT.LE.1) GO TO 15
      IF(IPRT.EQ.0) GO TO 14

      PRINT RESULTS.

      PRINT 101,IR,JR
      PRINT 1001,IR1,JR1
101   FORMAT(/10X,'COMPONENTS',I3,3X,'AND',3X,I3,3X,'ARE TESTED',10X,'TH
     *E REAL PARTS OF TWO COMPLEX FACTORS')
1001  FORMAT(/10X,'COMPONENTS',I3,3X,'AND',3X,I3,3X,'ARE TESTED',10X,'TH
     *E IMAGINARY PARTS OF TWO COMPLEX FACTORS')
      PRINT 102,TT
      IF (TT.GT.TQ1) GO TO 15
102   FORMAT (10X,'TEST QUANTITY',G12.5)
```

```
      COMPUTE THE ESTIMATED COMMON VALUE.

      DO 12 K=1,N
   11 ST(K)=0.
      DO 12 L=1,2
   12 ST(K)=ST(K)+SP(K,L)*V(L)
      DO 13 K=1,4
   13 F(K)=0.
      F(1)=X(IR)-ST(IR)
      F(2)=X(JR)-ST(JR)
      F(3)=X(IR1)-ST(IR1)
      F(4)=X(JR1)-ST(JR1)
      PRINT 122,F(1),F(3)
  122 FORMAT(/10X,'PREDICTED VALUE OF THE REAL PARTS',5X,G12.5//10X,'PRE
     *DICTED VALUE OF THE IMAGINARY PARTS',G12.5)
   14 CONTINUE
      NQ=NQ+1
      IQ(NQ,1)=IR
      IQ(NQ,2)=JR
      IQ(NQ,3)=IR1
      IQ(NQ,4)=JR1
   15 CONTINUE

      TEST OF REAL FACTORS

   20 IF (NAP*NBP.EQ.0) GO TO 30

      COMPUTE TEST VECTOR. -
      COMPUTE TEST QUANTITY.

      DO 21 I=1,NAP
      DO 21 J=1,NBP
      TT=0.
      IR=NCA+I
      JR=NCB+J+NA
      AN=PX(IR,IR)+PX(JR,JR)-2*PX(IR,JR)
      TT=(X(IR)-X(JR))**2/AN
      IF(TT.GT.TQ.AND.IPRT.LE.1) GO TO 21
      IF(IPRT.EQ.0) GO TO 22

      PRINT RESULTS.

      PRINT 111,IR,JR
  111 FORMAT(/10X,'COMPONENTS',I3,3X,'AND',3X,I3,3X,'ARE TESTED',10X,'TW
     *O REAL FACTORS')
      PRINT 102,TT
      IF (TT.GT.TQ) GO TO 21

      COMPUTE THE ESTIMATED COMMON VALUE.

      FF=X(IR)*PX(JR,JR)+X(JR)*PX(IR,IR)-(X(IR)+X(JR))*PX(IR,JR)
      FF=FF/AN
      PRINT 103,FF
  103 FORMAT (10X,'PREDICTED VALUE',G12.5)
   22 CONTINUE
      NQ=NQ+1
      IQ(NQ,1)=I+NCA
      IQ(NQ,2)=JR
   21 CONTINUE

      TEST OF ONE COMPLEX POLE AND ONE REAL ZERO.

   30 IF (NCA*NBP.EQ.0) GO TO 50
```

```
C     COMPUTE TEST VECTOR.
C
      DO 45 I=1,NCA,2
      DO 45 J=1,NBP
      TT=0.
      IR=I
      IR1=I+1
      JR=J+NA+NCB
      JR1=0
      DO 35 K=1,N
      S1(1,K)=0.
   35 S1(2,K)=0.
      S1(1,IR)=1.
      S1(1,JR)=-1.
      S1(2,IR1)=1.
      Y(1)=X(IR)-X(JR)
      Y(2)=X(IR1)
C
C     COMPUTE TEST QUANTITY.
C
      DO 38 K=1,N
      DO 38 L=1,2
   37 SP(K,L)=0.
      DO 38 M=1,N
   38 SP(K,L)=SP(K,L)+PX(K,M)*S1(L,M)
      DO 40 K=1,2
      DO 40 L=1,2
   39 SPS(K,L)=0.
      DO 40 M=1,N
   40 SPS(K,L)=SPS(K,L)+S1(K,M)*SP(M,L)
      EPS=10.**-07*SPS(1,1)
      CALL DESYM(SPS,B,2,EPS,IERR,ID)
      IF(IERR.EQ.-1)GO TO 150
      CALL SOLVS(B,Y,V,2,1,ID)
      TT=V(1)*Y(1)+V(2)*Y(2)
      IF(TT.GT.TQ1.AND.IPRT.LE.1) GO TO 45
      IF(IPRT.EQ.0) GO TO 44
C
C     PRINT RESULTS.
C
      PRINT 123,IR,JR
  123 FORMAT(/10X,'COMPONENTS',I3,3X,'AND',3X,I3,3X,'ARE TESTED',10X,'TH
     *E REAL PARTS OF ONE COMPLEX AND ONE REAL FACTOR')
      PRINT 1023,IR1
 1023 FORMAT(/10X,'COMPONENT',I4,15X,'IS TESTED',11X,'THE IMAGINARY PART
     * OF ONE COMPLEX FACTOR')
      PRINT 102,TT
      IF (TT.GT.TQ1) GO TO 45
C
C     COMPUTE THE ESTIMATED COMMON VALUE.
C
      DO 42 K=1,N
   41 ST(K)=0.
      DO 42 L=1,2
   42 ST(K)=ST(K)+SP(K,L)*V(L)
      DO 43 K=1,3
   43 F(K)=0.
      F(1)=X(IR)-ST(IR)
      F(2)=X(JR)-ST(JR)
      F(3)=0.
      PRINT 1122,F(1),F(3)
 1122 FORMAT(/10X,'PREDICTED VALUE OF THE REAL PARTS',4X,G12.5//10X,'PRE
     *DICTED VALUE OF THE IMAGINARY PART',G12.5)
   44 CONTINUE
```

```
      NQ=NQ+1
      IQ(NQ,1)=IR
      IQ(NQ,2)=JR
      IQ(NQ,3)=IR1
   45 CONTINUE
C
C     TEST OF ONE REAL POLE AND ONE COMPLEX ZERO.
C
   50 IF (NAP*NCB.EQ.0) GO TO 70
C
C     COMPUTE TEST VECTOR.
C
      DO 65 I=1,NAP
      DO 65 J=1,NCB,2
      TT=0.
      IR=I+NCA
      IR1=0
      JR=J+NA
      JR1=J+NA+1
      DO 55 K=1,N
      S1(1,K)=0.
   55 S1(2,K)=0.
      S1(1,IR)=1.
      S1(1,JR)=-1.
      S1(2,JR1)=1.
      Y(1)=X(IR)-X(JR)
      Y(2)=X(JR1)
C
C     COMPUTE TEST QUANTITY.
C
      DO 58 K=1,N
      DO 58 L=1,2
   57 SP(K,L)=0.
      DO 58 M=1,N
   58 SP(K,L)=SP(K,L)+PX(K,M)*S1(L,M)
      DO 60 K=1,2
      DO 60 L=1,2
   59 SPS(K,L)=0.
      DO 60 M=1,N
   60 SPS(K,L)=SPS(K,L)+S1(K,M)*SP(M,L)
      EPS=10.**-07*SPS(1,1)
      CALL DESYM(SPS,B,2,EPS,IERR,ID)
      IF(IERR.EQ.-1)GO TO 150
      CALL SOLVS(B,Y,V,2,1,ID)
      TT=V(1)*Y(1)+V(2)*Y(2)
      IF(TT.GT.TQ1.AND.IPRT.LE.1) GO TO 65
      IF(IPRT.EQ.0) GO TO 64
C
C     PRINT RESULTS.
C
      PRINT 123,IR,JR
      PRINT 2023,JR1
 2023 FORMAT(/10X,'COMPONENT',I16,3X,'IS TESTED',11X,'THE IMAGINARY PART
     * OF ONE COMPLEX FACTOR')
      PRINT 102,TT
      IF (TT.GT.TQ1) GO TO 65
C
C     COMPUTE THE ESTIMATED COMMON VALUE.
C
      DO 62 K=1,N
   61 ST(K)=0.
      DO 62 L=1,2
   62 ST(K)=ST(K)+SP(K,L)*V(L)
      DO 63 K=1,3
```

```
   63 F(K)=0.
      F(1)=X(IR)-ST(IR)
      F(2)=X(JR)-ST(JR)
      F(3)=0.
      PRINT 1122,F(1),F(3)
   64 CONTINUE
      NQ=NQ+1
      IQ(NQ,1)=IR
      IQ(NQ,2)=JR
      IQ(NQ,4)=JR1
   65 CONTINUE
C
C     COMPUTE MATRIX IQ
C
   70 CONTINUE
      IF(NQ.EQ.0) GO TO 300
      DO 81 I=1,10
   81 IND(I)=0
      DO 82 I=1,100
      DO 82 J=1,4
   82 IQQ(I,J)=0
      DO 83 I=1,NA
      DO 83 J=1,NQ
      IF (I.NE.IQ(J,1)) GO TO 83
      IND(I)=IND(I)+1
   83 CONTINUE
   84 CONTINUE
      L=0
      K=1
   85 I=0
   86 CONTINUE
      I=I+1
      IF (I.GT.NA) GO TO  97
      IF (IND(I).EQ.0) GO TO 86
      L=I
      DO 90 J=1,NA
      IF (IND(J).EQ.0) GO TO 90
      IF (IND(I).LE.IND(J)) GO TO 90
      L=J
      I=J
   90 CONTINUE
      KK=0
   92 CONTINUE
   93 KK=KK+1
      IF (KK.GT.NQ) GO TO 95
      IF (IQ(KK,1).NE.L) GO TO 93
      IQQ(K,1)=IQ(KK,1)
      IQQ(K,2)=IQ(KK,2)
      IQQ(K,3)=IQ(KK,3)
      IQQ(K,4)=IQ(KK,4)
      K=K+1
      GO TO 92
   95 CONTINUE
      IND(L)=0
   96 GO TO 85
   97 CONTINUE
      DO 98 I=1,100
      DO 98 J=1,4
   98 IQ(I,J)=IQQ(I,J)
      IF(IPRT.EQ.0) GO TO 99
C
C     PRINT RESULTS.
C
      PRINT 130
```

```
 130 FORMAT(/10X,'MATRIX IQ'/10X,'REAL PART OF POLE, REAL PART OF ZERO,
    * IMAGINARY PART OF POLE, IMAGINARY PART OF ZERO')
      IF (NQ.EQ.0) RETURN
      DO 75 I=1,NQ
  75  PRINT 131,(IQ(I,J),J=1,4)
 131  FORMAT(/7X,4I5)
      GO TO 99
 150  PRINT 200
C
C     COMPUTE EIGENVALUES AND EIGENVECTORS.
C
 200  FORMAT (/10X,'DECOMPOSITION IS IMPOSSIBLE')
      PRINT 4000,IR,JR,IR1,JR1
4000  FORMAT(/10X,'THIS COMBINATION GOES WRONG'/10X,I5,I5,I10,I5)
      PRINT 5000
5000  FORMAT(/10X,'MATRIX S*P*S-TR')
      DO 5010 I=1,2
5010  PRINT 5020,(SPS(I,J),J=1,2)
5020  FORMAT(8X,10G12.7)
      CALL EIGS(SPS,R,EV,2,ID,0)
      PRINT 5030
5030  FORMAT(/10X,'MATRIX S*P*S-TR AFTER EIGS')
      DO 5040 I=1,2
5040  PRINT 5020,(SPS(I,J),J=1,2)
      PRINT 5050
5050  FORMAT(/10X,'EIGENVECTORS')
      DO 5055 I=1,2
5055  PRINT 5020,(R(I,J),J=1,2)
      PRINT 5060
5060  FORMAT(/10X,'VECTOR OF EIGENVALUES')
      PRINT 5020,(EV(I),I=1,2)
  99  CONTINUE
      IF(IERR.NE.-1)IERR=0
      RETURN
 300  CONTINUE
      PRINT 1300
1300  FORMAT(/10X,'NO INDIVIDUELL TESTS ARE POSSIBLE')
      IF(IERR.NE.-1)IERR=0
      RETURN
      END
```

```
      SUBROUTINE TESTUT(T,X,PX,AM,BM,TT1,IQ,NA,NB,NMA,NMB,NQ,IPR,IERR,
     *IA,IB,IC)
C
C     THE SUBROUTINE COMPUTES THE GREATEST COMMON POLYNOMIAL TO TWO GIVEN
C     POLYNOMIALS IN STASTISTICAL SENSE.
C     THE COMMON POLYNOMIAL IS FOUND BY EXAMINING COMBINATIONS OF THE INDI-
C     VIDUELL TESTS COMPUTED IN SUBROUTINE TEST.
C     AUTHOR ERIK BURSTROM 1972-12-24
C
C     T VECTOR OF ORDER (NA+NB), CONTAINGING THE A-COEFFICIENTS AND THE
C     B-COEFFICIENTS.
C     X VECTOR OF ORDER (NA+NB), CONTAINING THE POLES AND THE ZEROS COMPUTED
C     IN SUBROUTINE XPX.
C     PX THE COVARIANCE MATRIX OF VECTOR X, OF ORDER (NA+NB)*(NA+NB), COMPUTED
C     IN SUBROUTINE XPX.
C     AM VECTOR CONTAINING THE NEW ESTIMATED A-COEFFICIENTS.
C     BM VECTOR CONTAINING THE NEW B-COEFFICIENTS.
C     TT1 STATISTICAL TEST QUANTITY.
C     IQ MATRIX OF ORDER(NQ*4), CONTAINING ALL POSSIBLE INDIVIDUELL TESTS
C     COMPUTED IN SUBROUTINE TEST.
C     NA THE NUMBER OF A-COEFFICIENTS (MIN 1, MAX 10).
C     NB THE NUMBER OF B-COEFFICIENTS (MIN 0, MAX 10).
C     NMA THE NUMBER OF A-COEFFICIENTS (MIN 1, MAX 10).
C     NMB THE NUMBER OF B-COEFFICIENTS (MIN 1, MAX 10).
C     NQ THE NUMBER OF POSSIBLE INDIVIDUELL TESTS, COMPUTED IN SUBROUTINE TEST.
C     IPR  IF IPR=0 NOTHING IS PRINTED.
C          IF IPR=1 THE COMPONENTS THAT CAN BE ABBREVIATED, THE TEST QUANTITY
C           AND DEGREES OF FREEDOM, THE NEW ESTIMATED X-VECTOR, THE COMPONENTS
C           THAT ARE EQUAL, THE NEW A-POLYNOMIAL, DEGREE OF A AND POLES, THE NE
C            B-POLYNOMIAL, DEGREE OF B AND ZEROS AND STATIC GAIN FOR THE NEW
C           MODEL ARE PRINTED ONCE.
C          IF IPR=2 AS IPR=1 + THE COMMON POLYNOMIAL AND THE TWO REST-POLYNO-
C           MIALS ARE PRINTED ONCE.
C          IF IPR=3 AS IPR=2 + ALL POSSIBLE COMBINATIONS AND TEST QUANTITIES OF
C           ALL TESTS WITH GREATER AND THE SAME DEGREE OF FREEDOM AS THE BEST
C           POSITIVE TEST.
C          IF IPR=4 AS IPR=3 + ALL POSSIBLE COMBINATIONS AND CORRESPONDING
C           TEST QUANTITIES OF ALL DEGREES OF FREEDOM ARE PRINTED.
C     IERR IF IERR=0 NORMAL OUTPUT.
C          IF IERR=-1 THE COVARIANCE MATRIX PX IS NOT POSITIVE DEFINITE. THE
C          EIGENVALUES AND CORRESPONDING EIGENVECTORS OF MATRIX PX ARE COMPUTE
C          AND PRINTED.
C     IA DIMENSION PARAMETER OF MATRIX PX.
C     IB IF IB=0 B(0)=0 IS ASSUMED.
C        IF IB=1 B(0)=1 IS ASSUMED.
C     IC DIMENSION PARAMETER OF MATRIX IQ.
C
C     D VECTOR OF ORDER 10 CONTAINING THE COEFFICIENTS OF THE COMMON
C     POLYNOMIAL.
C
C     SUBROUTINE REQUIRED
C          DESYM
C          SOLVS
C          ROT
C          EIGS
C          PDIV
C          PMPY
C
      DIMENSION T(1),X(1),PX(IA,IA),IQ(IC,1),AM(1),BM(1)
      COMMON/COMPOL/D(10)
C
      DIMENSION MIX(100),MAX(100),LL(100),S(20,20),Y(20),V(20),SP(20,20)
     *,SPS(20,20),B(20,20),ST(20),F(20),MIN(100),LINE(22),R(20,20),
```

```
*EV(20)
C
      DATA STRECK/5H-----/
      TJITWO(TN)=1.538*TN+3.2
      IF(NQ.EQ.0) RETURN
      IRANK1=0
      IERR=0
C
C     EPS IS USED IN DESYM.
C
      EPS=1.0E-10
      EPS1=1.0E-05
      TT1=10.**30
      ISKR=IPR
      ID=20
C
C     IX IS THE NUMBER OF FACTORS THAT CAN BE ABBREVIATED.
C
      IX=MINO(NA,NB-1)
      IF (IB.EQ.1) IX=MINO(NA,NB)
      IF(IX.EQ.0) RETURN
      ITEST=0
      N=NA+NB-1
      IF (IB.EQ.1) N=NA+NB
      DO 2 I=1,100
      LL(I)=0
      MAX(I)=0
      MIN(I)=0
    2 MIX(I)=0
      DO 5 I=1,NQ
      IT=IQ(I,3)+IQ(I,4)
      IF (IT.EQ.0) GO TO 3
      IU=IQ(I,3)*IQ(I,4)
      IF(IU.EQ.0) GO TO 4
      MAX(I)=2
      GO TO 5
    3 MAX(I)=0
      GO TO 5
    4 MAX(I)=1
    5 CONTINUE
    8 CONTINUE
      DO 99 I=1,NQ
      LL(I)=0
   99 MIX(I)=0
      I=0
C
C     COMPUTE A POSSIBLE COMBINATION OF FACTORS.
C
   10 I=I+1
      KM=0
      DO 200 K=1,I
  200 MIX(K)=0
      IF(I.GT.NQ) RETURN
      I1=I
      MIX(I)=1
      IF (MAX(I).EQ.2) GO TO 11
      KM=KM+1
      LL(KM)=I
      IF (KM.EQ.IX) GO TO 20
      IF (KM.LT.IX) GO TO 12
      LL(KM)=0
      MIX(I)=0
      KM=KM-1
      GO TO 12
```

```
 11 KM=KM+2
    LL(KM)=I
    IF (KM.EQ.IX) GO TO 20
    IF (KM.LT.IX) GO TO 12
    LL(KM)=0
    MIX(I)=0
    KM=KM-2
 12 I=I+1
    IF (I.GT.NQ) GO TO 18
    II=I-1
    DO 15 J=I1,II
    IF (MIX(J).EQ.0) GO TO 15
    ITT=(IQ(I,1)-IQ(J,1))*(IQ(I,2)-IQ(J,2))
    IF (ITT.NE.0) GO TO 15
    ITE=MAX(I)*MAX(J)
    IF (ITE.NE.1) GO TO 12
    IF (IQ(I,1).EQ.IQ(J,1)) GO TO 13
    IF (IQ(I,4).EQ.0) GO TO 12
    GO TO 15
 13 IF (IQ(I,3).EQ.0) GO TO 12
 15 CONTINUE
    MIX(I)=1
    IF (MAX(I).EQ.2) GO TO 16
    KM=KM+1
    LL(KM)=I
    IF (KM.LT.IX) GO TO 12
    IF (KM.EQ.IX) GO TO 18
    LL(KM)=0
    KM=KM-1
    MIX(I)=0
    GO TO 12
 16 KM=KM+2
    LL(KM)=I
    IF (KM.LT.IX) GO TO 12
    IF (KM.EQ.IX) GO TO 18
    LL(KM)=0
    KM=KM-2
    MIX(I)=0
    GO TO 12
 18 CONTINUE
    IF(KM.LE.0) GO TO 80
    IF (KM.LT.IX) GO TO 60
C
C
 20 CONTINUE
C
C   COMPUTE MATRIX S.
C
    NRA=1
    DO 40 I=1,NQ
    IF (MIX(I).EQ.0) GO TO 40
    IF (MAX(I).EQ.0) GO TO 31
    IF(MAX(I).EQ.1.AND.I.GT.1) GO TO 26
 19 CONTINUE
    DO 25 M=1,N
    S(NRA,M)=0.
    S(NRA+1,M)=0.
    IF (M.EQ.IQ(I,1)) GO TO 21
    IF (M.EQ.IQ(I,2)) GO TO 22
    IF (M.EQ.IQ(I,3)) GO TO 23
    IF (M.EQ.IQ(I,4)) GO TO 24
    GO TO 25
 21 S(NRA,M)=1.
    GO TO 25
```

```
   22 S(NRA,M)=-1.
      GO TO 25
   23 S(NRA+1,M)=1.
      GO TO 25
   24 S(NRA+1,M)=-1.
   25 CONTINUE
      NRA=NRA+2
      GO TO 40
C
   26 CONTINUE
      I1=I-1
      ICO=0
      DO 30 J=1,I1
      IF(MIX(J).EQ.0.OR.MAX(J).NE.1) GO TO 30
      IF(IABS(IQ(I,1)-IQ(J,1))+IABS(IQ(I,3)-IQ(J,3)).NE.0.AND.IABS(IQ(I,
     *2)-IQ(J,2))+IABS(IQ(I,4)-IQ(J,4)).NE.0) GO TO 30
      DO 29 M=1,N
      IF(M.EQ.IQ(I,1)) GO TO 27
      IF(M.EQ.IQ(I,2)) GO TO 28
      S(NRA,M)=0.
      GO TO 29
   27 S(NRA,M)=1.
      GO TO 29
   28 S(NRA,M)=-1.
   29 CONTINUE
      NRA=NRA+1
      ICO=1
   30 CONTINUE
      IF(ICO.EQ.0) GO TO 19
      GO TO 40
   31 DO 35 M=1,N
      IF (M.EQ.IQ(I,1)) GO TO 32
      IF (M.EQ.IQ(I,2)) GO TO 33
      S(NRA,M)=0.
      GO TO 35
   32 S(NRA,M)=1.
      GO TO 35
   33 S(NRA,M)=-1.
   35 CONTINUE
      NRA=NRA+1
   40 CONTINUE
C
C     COMPUTE TEST VECTOR.
C
      NRA=NRA-1
      DO 46 K=1,NRA
      Y(K)=0.
      DO 46 L=1,N
   46 Y(K)=Y(K)+S(K,L)*X(L)
C
C     COMPUTE TEST QUANTITY.
C
      DO 48 K=1,N
      DO 48 L=1,NRA
      SP(K,L)=0.
      DO 48 M=1,N
   48 SP(K,L)=SP(K,L)+PX(K,M)*S(L,M)
      DO 50 K=1,NRA
      DO 50 L=1,NRA
      SPS(K,L)=0.
      DO 50 M=1,N
   50 SPS(K,L)=SPS(K,L)+S(K,M)*SP(M,L)
      EPS=10.**-07*SPS(1,1)
      CALL DESYM(SPS,B,NRA,EPS,IRANK1,ID)
```

```
      IF (IRANK1.EQ.-1) GO TO 250
      CALL SOLVS(B,Y,V,NRA,1,IC)
      TT=0.
      DO 52 K=1,NRA
   52 TT=TT+V(K)*Y(K)
      TRA=FLOAT(NRA)
      IF(ISKR.LE.2) GO TO 55
C
C     PRINT RESULTS.
C
      PRINT 101
      DO 53 I=1,NQ
      IF(MIX(I).EQ.0) GO TO 53
      PRINT 102,(IQ(I,J),J=1,4)
   53 CONTINUE
      PRINT 103,TT,NRA
   55 CONTINUE
      IF (TT.GT.TJITWO(TRA)) GO TO 60
      IF(TT.GT.TT1) GO TO 60
      ITEST=1
      TT1=TT
C
C     COMPUTE THE NEW ESTIMATED X-VECTOR.
C
      DO 56 I=1,N
      ST(I)=0.
      DO 56 J=1,NRA
   56 ST(I)=ST(I)+SP(I,J)*V(J)
      DO 57 I=1,N
      F(I)=0.
   57 F(I)=X(I)-ST(I)
      DO 58 I=1,N
      IF(ABS(F(I)).LT.EPS1) F(I)=0.
   58 CONTINUE
      DO 121 I=1,NQ
  121 MIN(I)=MIX(I)

      COMPUTE A NEW POSSIBLE COMBINATION OF FACTORS.

   60 CONTINUE
      IF (KM.EQ.1) GO TO 64
      IF (KM.EQ.2) GO TO 61
      GO TO 63
   61 I=LL(KM)
      IF (MAX(I).NE.2) GO TO 63
   64 IF (LL(KM).GE.NQ) GO TO 80
   63 CONTINUE
      IF (LL(KM).GE.NQ) GO TO 65
      I=LL(KM)
      LL(KM)=0
      MIX(I)=0
      IF (MAX(I).EQ.2) GO TO 62
      KM=KM-1
      GO TO 71
   62 KM=KM-2
      GO TO 71
   65 CONTINUE
      I=LL(KM)
      MIX(I)=0
      LL(KM)=0
      IF (MAX(I).EQ.2) GO TO 66
      KM=KM-1
      GO TO 67
   66 KM=KM-2
```

```
 67 CONTINUE
    I=LL(KM)
    IF (I.LE.I1) GO TO 10
    MIX(I)=0
    IF (MAX(I).EQ.2) GO TO 167
    KM=KM-1
    GO TO 71
167 KM=KM-2
 71 I=I+1
    IF (I.LE.NQ) GO TO 68
    I=LL(KM)
    MIX(I)=0
    LL(KM)=0
    IF (MAX(I).EQ.2) GO TO 69
    KM=KM-1
    IF (I.LE.I1) GO TO 10
    GO TO 71
 69 KM=KM-2
    IF (I.LE.I1) GO TO 10
    GO TO 71
 68 II=I-1
    DO 75 J=I1,II
    IF (MIX(J).EQ.0) GO TO 75
    ITT=(IQ(I,1)-IQ(J,1))*(IQ(I,2)-IQ(J,2))
    IF (ITT.NE.0) GO TO 75
    ITE=MAX(I)*MAX(J)
    IF (ITE.NE.1) GO TO 71
    IF (IQ(I,1).EQ.IQ(J,1)) GO TO 73
    IF (IQ(I,4).EQ.0) GO TO 71
    GO TO 75
 73 IF (IQ(I,3).EQ.0) GO TO 71
 75 CONTINUE
    MIX(I)=1
    IF (MAX(I).EQ.2) GO TO 76
    KM=KM+1
    LL(KM)=I
    IF (KM.EQ.IX) GO TO 20
    IF (KM.LT.IX) GO TO 78
    LL(KM)=0
    KM=KM-1
    MIX(I)=0
    GO TO 71
 76 KM=KM+2
    LL(KM)=I
    IF (KM.EQ.IX) GO TO 20
    IF (KM.LT.IX) GO TO 78
    LL(KM)=0
    KM=KM-2
    MIX(I)=0
    GO TO 71
 78 CONTINUE
    I=I+1
    IF (I.GT.NQ) GO TO 60
    GO TO 68
 80 IF(ITEST.EQ.1)GO TO 300
 81 CONTINUE
    IX=IX-1
    IF (IX.LE.0) RETURN
    GO TO 8
300 CONTINUE
    DO 122 I=1,NQ
122 MIX(I)=MIN(I)
    IF(ISKR.EQ.0)GO TO 401
C
```

```
C     PRINT RESULTS.
C
      PRINT 101
  101 FORMAT(/10X,'THE FOLLOWING COMPONENTS ARE TESTED'/10X,35(1H-)/10X,
     *'REAL PART OF POLE, REAL PART OF ZERO, IMAGINARY PART OF POLE, IMA
     *GINARY PART OF ZERO')
      DO 54 I=1,100
      IF (MIX(I).EQ.0) GO TO 54
      PRINT 102,(IQ(I,J),J=1,4)
  102 FORMAT(/7X,4I5)
   54 CONTINUE
      PRINT 103,TT1,NRA
  103 FORMAT(/10X,'TEST QUANTITY',G12.5,10X,'DEGREES OF FREEDOM',I3/10X,
     *25(1H-))
      IF(N.GT.10) GO TO 2999
      PRINT 1300
 1300 FORMAT(/10X,'ESTIMATED VECTOR X'/10X,18(1H-)/10X,'COMPONENT NUMBER
     *')
      PRINT 1302,(I,I=1,N)
 1302 FORMAT(/2X,10I12)
      CALL ENCODE(LINE)
      CALL FMTX(11)
      DO 1234 I=1,N
      CALL FMTA(STRECK,5)
 1234 CALL FMTX(7)
      CALL FMTX(121-N*12)
      PRINT 1235,(LINE(I),I=1,22)
 1235 FORMAT(22A6)
      PRINT 1301,(F(I),I=1,N)
 1301 FORMAT(/8X,10G12.5)
      GO TO 399
 2999 CONTINUE
      PRINT 1300
      PRINT 1302,(I,I=1,10)
      CALL ENCODE(LINE)
      CALL FMTX(11)
      DO 610 I=1,10
      CALL FMTA(STRECK,5)
  610 CALL FMTX(7)
      CALL FMTX(1)
      PRINT 1235,(LINE(I),I=1,22)
      PRINT 1301,(F(I),I=1,10)
      PRINT 1302,(I,I=11,N)
      CALL ENCODE(LINE)
      CALL FMTX(11)
      DO 620 I=11,N
      CALL FMTA(STRECK,5)
  620 CALL FMTX(7)
      CALL FMTX(121-(N-10)*12)
      PRINT 1235,(LINE(I),I=1,22)
      PRINT 1301,(F(I),I=11,N)
  399 CONTINUE
      DO 400 I=1,NQ
      IF(MIX(I).EQ.0) GO TO 400
      IF(MAX(I).EQ.2) GO TO 380
      IF(MAX(I).EQ.1) GO TO 370
  360 PRINT 1360,IQ(I,1),IQ(I,2)
 1360 FORMAT(/10X,'COMPONENTS',I3,3X,'AND',3X,I3,3X,'ARE EQUAL'/10X,'ONE
     * REAL POLE AND ONE REAL ZERO CAN BE ABBREVIATED')
      GO TO 400
  370 IF(IQ(I,4).NE.0) GO TO 375
      PRINT 1370,(IQ(I,J),J=1,3)
 1370 FORMAT(/10X,'COMPONENTS',I3,3X,'AND',3X,I3,3X,'ARE EQUAL',10X,'THE
     * REAL PARTS'//10X,'COMPONENT',I4,3X,'IS =0.00000',20X,'THE REMAINI
```

```
     *NG COMPLEX POLE IS REAL')
      PRINT 1371
 1371 FORMAT(/10X,'ONE COMPLEX POLE AND ONE REAL ZERO CAN BE ABBREVIATED
     *')
      GO TO 400
  375 PRINT 1375,IQ(I,1),IQ(I,2),IQ(I,4)
 1375 FORMAT(/10X,'COMPONENTS',I3,3X,'AND',3X,I3,3X,'ARE EQUAL',10X,'THE
     * REAL PARTS'//10X,'COMPONENT',I4,3X,'IS =0.00000',20X,'THE REMAINI
     *NG COMPLEX ZERO IS REAL')
      PRINT 1376
 1376 FORMAT(/10X,'ONE REAL POLE AND ONE COMPLEX ZERO CAN BE ABBREVIATED
     *')
      GO TO 400
  380 PRINT 1380,(IQ(I,J),J=1,4)
 1380 FORMAT(/10X,'COMPONENTS',I3,3X,'AND',3X,I3,3X,'ARE EQUAL',10X,'THE
     * REAL PARTS'//10X,'COMPONENTS',I3,3X,'AND',3X,I3,3X,'ARE EQUAL',10
     *X,'THE IMAGINARY PARTS'//10X,'TWO COMPLEX POLES AND TWO COMPLEX ZE
     *ROS CAN BE ABBREVIATED')
  400 CONTINUE
  401 CONTINUE
C
C     COMPUTE THE NEW A- AND B-POLYNOMIALS.
C
      CALL PDIV(T,X,F,AM,BM,IQ,MIX,MAX,NA,NB,NMA,NMB,NQ,IPR,IERR,IB,IC)
      GO TO 301

      COMPUTE EIGENVALUES AND EIGENVECTORS.

  250 PRINT 260
  260 FORMAT (/10X,'DECOMPOSITION IS IMPOSSIBLE')
      PRINT 4000
 4000 FORMAT(/10X,'MATRIX S')
      DO 5000 I=1,NRA
 5000 PRINT 5100,(S(I,J),J=1,N)
 5100 FORMAT(10X,10G10.5/10X,10G10.5)
      PRINT 5200
 5200 FORMAT(/10X,'MATRIX S*P*S-TR')
      DO 5300 I=1,NRA
 5300 PRINT 7020,(SPS(I,J),J=1,NRA)
      CALL EIGS(SPS,R,EV,NRA,ID,0)
      PRINT 7000
 7000 FORMAT(/10X,'MATRIX SPS-TR AFTER EIGS')
      DO 7010 I=1,NRA
 7010 PRINT 7020,(SPS(I,J),J=1,NRA)
 7020 FORMAT(10X,10G12.5)
      PRINT 7005
 7005 FORMAT(/10X,'MATRIX R')
      DO 7030 I=1,NRA
 7030 PRINT 7020,(R(I,J),J=1,NRA)
      PRINT 7015
 7015 FORMAT(/10X,'VECTOR OF EIGENVALUES')
      PRINT 7020,(EV(I),I=1,NRA)
      IERR=-1
  301 CONTINUE
      IF(ITEST.EQ.0)GO TO 60
      IF(ISKR.GE.4)GO TO 81
      RETURN
      END
```

```
      SUBROUTINE PDIV(T,X,F,AM,BM,IQ,MIX,MAX,NA,NB,NMA,NMB,NQ,IPR,IERR,
     *IB,IC)
C
C     THE SUBROUTINE DIVIDES TWO GIVEN POLYNOMIALS WITH THE GREATEST COMMON
C     POLYNOMIAL. THE COMMON POLYNOMIAL IS COMPUTED IN SUBROUTINE TESTUT.
C     THE POLES, ZEROS AND STATIC GAIN FOR THE NEW SYSTEM IS COMPUTED.
C     THIS SUBROUTINE MUST BE CALLED FROM SUBROUTINE TESTUT.
C     AUTHOR ERIK BURSTRÖM 1972-12-24
C
C     T=(A(1),...,A(NA),B(1),...,B(NB)) VECTOR OF ORDER (NA+NB).
C     X VECTOR OF ORDER (NA+NB) CONTAINING POLES AND ZEROS, COMPUTED IN SUB-
C     ROUTINE XPX.
C     F VECTOR OF ORDER (NA+NB) CONTAINING NEW ESTIMATES OF POLES AND ZEROS
C     ON THE ASSUMPTION THAT SOME FACTORS ARE EQUAL, COMPUTED IN SUB-
C     ROUTINE TESTUT.
C     AM VECTOR OF ORDER NMA (MIN 1, MAX 10) CONTAINING THE NEW A-COEFFICIENTS.
C     BM VECTOR OF ORDER NMB (MIN 1, MAX 10) CONTAINING THE NEW B-COEFFICIENTS.
C     IQ MATRIX CONTAINING ALL POSSIBLE INDIVIDUELL TESTS COMPUTED IN SUB-
C     ROUTINE TEST.
C     MIX VECTOR OF ORDER NQ COMPUTED IN SUBROUTINE TESTUT. IF MIX(I)=1
C     A CERTAIN COMBINATION OF FACTORS WILL BE ABBREVIATED AND IF MIX(I)=0
C     A CERTAIN COMBINATION WILL NOT BE ABBREVIATED.
C     MAX VECTOR OF ORDER NQ COMPUTED IN SUBROUTINE TESTUT. IF MAX(I)=0 TWO
C     REAL FACTORS CAN BE ABBREVIATED, IF MAX(I)=1 ONE REAL AND ONE COMPLEX
C     FACTOR CAN BE ABBREVIATED AND IF MAX(I)=2 TWO COMPLEX FACTORS CAN BE
C     ABBREVIATED. NOTE THAT IF MAX(I)=1 THERE ARE TWO DEGREES OF FREEDOM
C     ALTHOUGH ONLY ONE POLE AND ONE ZERO CAN BE ABBREVIATED.
C     NA NUMBER OF A-COEFFICIENTS.
C     NB NUMBER OF B-COEFFICIENTS.
C     NMA NUMBER OF NEW A-COEFFICIENTS.
C     NMB NUMBER OF NEW B-COEFFICIENTS.
C     NQ NUMBER OF POSSIBLE INDIVIDUELL TESTS COMPUTED IN SUBROUTINE TEST.
C     IPR  IF IPR=0 NOTHING IS PRINTED.
C          IF IPR=1 THE NEW A- AND B-POLYNOMIALS, POLES, ZEROS AND STATIC GAIN
C           ARE PRINTED.
C          IF IPR=2 AS IPR=1 + THE COMMON POLYNOMIAL AND THE REST POLYNOMIALS
C           ARE PRINTED.
C     IERR IF IERR=0 NORMAL OUTPUT
C          IF IERR=-1 COMPUTATION OF POLES OR ZEROS HAS FAILED.
C     IB IF IB=0 B(0)=0 IS ASSUMED.
C        IF IB=1 B(0)=1 IS ASSUMED.
C     IC DIMENSION PARAMETER OF MATRIX IQ.
C
C     D VECTOR OF ORDER 10 CONTAINING THE COEFFICIENTS OF THE COMMON
C     POLYNOMIAL.
C
C     SUBROUTINE REQUIRED
C          ROT
C          PMPY
C
      DIMENSION T(1),X(1),F(1),IQ(IC,1),MIX(1),MAX(1),AM(1),BM(1)
      COMMON/COMPOL/D(10)
C
      DIMENSION R(10),Y(2),YY(3),A(10),A1(10),RE(10),A2(10),AR2(10),
     *AI2(10)
      EPS=1.0E-05
      ISKR=IPR
      IF(IB.EQ.0)G1=0.
      IF(IB.EQ.1)G1=1.
      G2=1.
      G=0.
C
C     COMPUTE THE COMMON POLYNOMIAL.
```

```
C
      DO 1 I=1,10
      AM(I)=0.
      BM(I)=0.
      R(I)=0.
    1 D(I)=0.
      IDD=1
      IR=1
      IDY=2
      IDYY=3
      Y(1)=1.
      YY(1)=1.
      R(1)=1.
      D(1)=1.
C
      DO 15 I=1,NQ
      IF(MIX(I).EQ.0) GO TO 15
      IF(MAX(I).EQ.2) GO TO 10
      I1=IQ(I,1)
      Y(2)=-F(I1)
      CALL PMPY(D,IDD,R,IR,Y,IDY)
      GO TO 12
   10 I1=IQ(I,1)
      I2=IQ(I,3)
      YY(2)=-2*F(I1)
      YY(3)=F(I1)**2+F(I2)**2
      CALL PMPY(D,IDD,R,IR,YY,IDYY)
   12 IR=IDD
      DO 14 J=2,IDD
   14 R(J)=D(J)
   15 CONTINUE
      ID=IDD-1
      IF(ID.LE.0) RETURN
      IF(ISKR.LT.2) GO TO 19

      PRINT RESULTS.

      PRINT 290,ID,R(1)
  290 FORMAT(/10X,'THE COMMON POLYNOMIAL D(Z)',10X,'DEGREE OF D',I5/10X,
     *'D(0)=  ',G10.5)
      IF(IDD.EQ.1) GO TO 19
      DO 291 I=2,IDD
      I0=I-1
  291 PRINT 292,I0,R(I)
  292 FORMAT(/10X,'D(',I1,')=  ',G10.5)
   19 CONTINUE
      KK=0
C
C     COMPUTE THE NEW A- AND B-POLYNOMIALS.
C
      DO 20 I=2,IDD
   20 D(I-1)=D(I)
      D0=1.
   16 KK=KK+1
      A0=1.
      IF(IB.EQ.0.AND.KK.EQ.2) A0=0.
      IF(KK.EQ.1) NA1=NA
      IF(IB.EQ.0.AND.KK.EQ.2) NA1=NB-1
      IF(IB.EQ.1.AND.KK.EQ.2) NA1=NB
      A10=A0/D0
      IF(ID.EQ.NA1) GO TO 80
      IF(KK.EQ.2) GO TO 121
      DO 21 I=1,NA1
   21 A(I)=T(I)
```

```
      GO TO 22
  121 DO 122 I=1,NA1
  122 A(I)=T(I+NA)
   22 CONTINUE
      I3=NA1-ID
      IF(I3.GT.ID) GO TO 31
      A10=A0/D0
      A1(1)=(A(1)-D(1)*A10)/D0
      IF(I3.LE.1) GO TO 26
      DO 24 I=2,I3
      A1(I)=A(I)
      IJ=I-1
      DO 23 J=1,IJ
   23 A1(I)=A1(I)-D(J)*A1(I-J)
      A1(I)=(A1(I)-D(I)*A10)/D0
   24 CONTINUE
   26 CONTINUE
      IF(KK.EQ.1.OR.ABS(A10).GT.EPS) GO TO 39
      A1(I3+1)=A(I3+1)
      DO 27 J=1,I3
   27 A1(I3+1)=A1(I3+1)-D(J)*A1(-J+I3+1)
      A1(I3+1)=(A1(I3+1)-D(I3+1)*A10)/D0
      GO TO 39
   31 CONTINUE
      A10=A0/D0
      A1(1)=(A(1)-D(1)*A10)/D0
      IF(ID.LE.1) GO TO 34
      DO 33 I=2,ID
      A1(I)=A(I)
      IJ=I-1
      DO 32 J=1,IJ
   32 A1(I)=A1(I)-D(J)*A1(I-J)
      A1(I)=(A1(I)-D(I)*A10)/D0
   33 CONTINUE
   34 IL=ID+1
      DO 36 I=IL,I3
      A1(I)=A(I)
      DO 35 J=1,ID
   35 A1(I)=A1(I)-D(J)*A1(I-J)
      A1(I)=A1(I)/D0
   36 CONTINUE
      IF(KK.EQ.1.OR.ABS(A10).GT.EPS) GO TO 39
      A1(I3+1)=A(I3+1)
      DO 37 J=1,ID
   37 A1(I3+1)=A1(I3+1)-D(J)*A1(-J+I3+1)
      A1(I3+1)=A1(I3+1)/D0
   39 CONTINUE
      IF(ABS(A10).LT.EPS) GO TO 51
      DO 42 I=1,I3
   42 A2(I)=A1(I)/A10
      GO TO 53
   51 CONTINUE
      DO 52 I=1,I3
   52 A2(I)=A1(I+1)/A1(1)
   53 CONTINUE
C
C     COMPUTE POLES AND ZEROS.
C
      CALL ROT(I3,1,A2,A2,AR2,AI2,NERR1)
      IF(NERR1.GT.0) GO TO 2000
C
C     PRINT RESULTS.
C
      IF(KK.EQ.1) PRINT 101,I3,A10
```

```
 101 FORMAT(/10X,'THE NEW A-POLYNOMIAL',10X,'DEGREE OF A',I3,10X,'ROOTS
    * OF A'//10X,'A(0)= ',G10.5)
      IF(KK.EQ.1) PRINT 102,((I,A1(I),AR2(I),AI2(I)),I=1,I3)
 102 FORMAT(/10X,'A(',I1,')= ',G10.5,39X,2G10.5)
      IF(KK.EQ.2) PRINT 1201,I3,A10
1201 FORMAT(/10X,'THE NEW B-POLYNOMIAL',10X,'DEGREE OF B',I3,10X,'ROOTS
    * OF B'//10X,'B(0)= ',G10.5)
      IF(KK.EQ.2) PRINT 202,((I,A1(I),AR2(I),AI2(I)),I=1,I3)
 202 FORMAT(/10X,'B(',I1,')= ',G10.5,39X,2G10.5)
      I4=I3+1
      IF(KK.EQ.2.AND.ABS(A10).LT.EPS) PRINT 202,(I4,A1(I4))
      IF(KK.EQ.1) GO TO 61
      IF(KK.EQ.2) GO TO 62
  61 CONTINUE
      DO 54 I=1,I3
      G2=G2+A1(I)
  54 AM(I)=A1(I)
      NMA=I3
      GO TO 200
  62 CONTINUE
      DO 55 I=1,I3
  55 BM(I)=A1(I)
      BM(I4)=A1(I4)
      IF(ABS(A10).LT.EPS)NMB=I3+1
      IF(ABS(A10).GT.EPS)NMB=I3
      DO 71 I=1,NMB
  71 G1=G1+A1(I)
      GO TO 200
  80 CONTINUE
      IF(KK.EQ.1) PRINT 114
 114 FORMAT(/10X,'THE NEW A-POLYNOMIAL IS CONSTANT  1.0')
      NMA=0
      IF(IB.EQ.0.AND.KK.EQ.2) PRINT 113,T(NA+1)
 113 FORMAT(/10X,'THE B-POLYNOMIAL IS A CONSTANT',G10.5)
      NMB=1
      BM(1)=T(NA+1)
      G1=T(NA+1)
      IF(IB.EQ.1.AND.KK.EQ.2) PRINT 115
 115 FORMAT(/10X,'THE NEW B-POLYNOMIAL IS CONSTANT  1.0')
      NMB=0
 200 CONTINUE
      IF(ISKR.LT.2) GO TO 300
C
C     COMPUTE THE REST-POLYNOMIAL.
C
      RE0=A(I3+1)
      IF(ID.EQ.1) GO TO 208
      DO 201 I=1,I3
 201 RE0=RE0-D(I)*A1(-I+I3+1)
      RE0=RE0-D(I3+1)*A10
      I6=NA1-1-2*I3
      DO 204 I=1,I6
      RE(I)=A(I+I3+1)
      DO 203 J=1,I3
 203 RE(I)=RE(I)-D(J+I)*A1(-J+I3+1)
      RE(I)=RE(I)-D(I3+I+1)*A10
 204 CONTINUE
      I7=I6+1
      I8=ID-1
      DO 207 I=I7,I8
      RE(I)=A(I+I3+1)
      I9=ID-I
      DO 205 J=1,I9
 205 RE(I)=RE(I)-D(J+I)*A1(-J+I3+1)
```

```
 207 CONTINUE
 208 CONTINUE
     PRINT 211,I3,REO
 211 FORMAT(/10X,'THE REST-POLYNOMIAL',10X,'DEGREE OF REST',I3//10X,'RE
    *(0)= ',G10.5)
     IF(ID.EQ.1) GO TO 280
     PRINT 212,((I,RE(I)),I=1,I8)
 212 FORMAT(/10X,'RE(',I1,')= ',G10.5)
 280 CONTINUE
     GO TO 300
2000 IERR=-1
     IF(KK.EQ.1)PRINT 2001
     IF(KK.EQ.2)PRINT 2002
2001 FORMAT(/10X,'COMPUTATION OF POLES IS IMPOSSIBLE')
2002 FORMAT(/10X,'COMPUTATION OF ZEROS IS IMPOSSIBLE')
 300 IF(KK.LT.2) GO TO 16
C
C    COMPUTE STATIC GAIN.
C
     G=G1/G2
     PRINT 260,G
 260 FORMAT(/10X,'STATIC GAIN',5X,G10.5)
     RETURN
     END
```

```
      SUBROUTINE COFAC(T,PT,T1,T2,TT1,TT2,NA,NB,NMA1,NMB1,NMA2,NMB2,IPR,
     *IERR,IA,IB)
C
C     THIS SUBROUTINE COMPUTES THE GREATEST COMMON POLYNOMIAL TO TWO GIVEN
C     POLYNOMIALS BY USING THE EUCLIDEAN ALGORITHM. THE ORIGINAL POLY-
C     NOMIALS ARE ABBREVIATED WITH THE COMMON POLYNOMIALS. TO DECIDE
C     WHETHER TWO POLYNOMIALS HAVE COMMON FACTORS OR NOT, THE REST-POLY-
C     NOMIAL COMPUTED AT EACH DIVISION IS TESTED TO BE ZERO ON THE
C     ASSUMPTION THAT THE COEFFICIENTS ARE GAUSSIAN, I.E. A CHI-
C     SQUARE TEST WITH THE SIGNIFICANCE LEVEL OF 5% IS USED.
C     TWO DIFFERENT METHODS ARE USED WHEN THE DIVISIOS ARE DONE. ON ONE
C     HAND THE DIVISOR POLYNOMIAL IS NORMALIZED, I.E. THE HIGHEST DEGREE
C     COEFFICIENT IS 1 (VERSION 1), AND ON THE OTHER HAND THE DIVISOR
C     POLYNOMIAL IS NOT NORMALIZED, I.E. THE DIVISIONS ARE DONE STRAIGHT-
C     FORWARD (VERSION 2).
C     AUTHOR ERIK BURSTRÖM 1972-12-24
C
C     T=(A(1),...,A(NA),B(1),...,B(NB)) VECTOR OF ORDER (NA+NB) CONTAINING
C     THE COEFFICIENTS OF THE ORIGINAL A- RESP. B-POLYNOMIALS.
C     PT MATRIX OF ORDER (NA+NB)*(NA+NB) THE COVARIANCE MATRIX OF T.
C     T1 VECTOR OF ORDER (NMA1+NMB1) AT OUTPUT CONTAINING THE COEFFICIENTS
C     OF THE NEW A- RESP. B-POLYNOMIALS, WHEN VERSION 1 OF THE EUCLIDEAN
C     ALGORITHM IS USED.
C     T2 VECTOR OF ORDER (NMA2+NMB2) AT OUTPUT CONTAINING THE COEFFICIENTS
C     OF THE NEW A- RESP. B-POLYNOMIALS, WHEN VERSION 2 OF THE EUCLIDEAN
C     ALGORITHM IS USED.
C     TT1 STATISTICAL TEST-QUANTITY COMPUTED IN VERSION 1.
C     TT2 STATISTICAL TEST-QUANTITY COMPUTED IN VERSION 2.
C     NA NUMBER OF ORIGINAL A-COEFFICIENTS (MIN 1, MAX 10).
C     NB NUMBER OF ORIGINAL B-COEFFICIENTS (MIN 1, MAX 10). NA AND NB MUST
C     BE EQUAL.
C     NMA1 NUMBER OF ESTIMATED A-COEFFICIENTS IN VERSION 1.
C     NMB1 NUMBER OF ESTIMATED B-COEFFICIENTS IN VERSION 1.
C     NMA2 NUMBER OF ESTIMATED A-COEFFICIENTS IN VERSION 2.
C     NMB2 NUMBER OF ESTIMATED B-COEFFICIENTS IN VERSION 2.
C     IPR  IF IPR=0 NOTHING IS PRINTED.
C          IF IPR=1 VECTOR T, MATRIX PT AND THE STATIC GAIN FOR THE ORIGINAL
C           SYSTEM ARE PRINTED. ALL TEST-QUANTITIES DOWN TO THE FIRST ONE WHICH
C           MEANS A POSITIVE TEST, THE NEW A-POLYNOMIAL, DEGREE OF A AND POLES
C           THE NEW B-POLYNOMIAL, DEGREE OF B AND ZEROS, THE STATIC GAIN FOR
C           THE NEW SYSTEM, THE NEW VECTOR T, NUMBER OF NEW A- RESP. B-COEFFI-
C           CIENTS ARE PRINTED FOR BOTH VERSIONS.
C          IF IPR=2 AS IPR=1 + ALL REST-POLYNOMIALS AND CORRESPONDING COVA-
C           RIANCE MATRICES COMPUTED AT EACH DIVISION AND THE REST-POLY-
C           NOMIALS COMPUTED AT THE FINAL ABBREVIATION ARE PRINTED FOR
C           BOTH VERSIONS.
C     IERR IF IERR=0 NORMAL OUTPUT.
C          IF IERR=-1 EITHER IS MATRIX PT NOT POSITIVE DEFINITE OR IS IT
C          IMPOSSIBLE TO COMPUTE POLES OR ZEROS TO THE NEW SYSTEMS.
C     IA DIMENSON PARAMETER OF PT.
C     IB IF IB=0 B(0)=0 IS ASSUMED.
C        IF IB=1 B(0)=1 IS ASSUMED.
C
C     SUBROUTINE REQUIRED
C          EUKL1
C          EUKL2
C          PDV
C          ROT
C          DESYM
C          SOLVS
C
      DIMENSION T(1),PT(IA,IA),T1(1),T2(1)
C
```

```
      DIMENSION T3(22),PT3(22,22),T4(22),PT4(22,22)
      IF(IPR.GT.0) PRINT 100,NA,NB
  100 FORMAT(1H1,9X,'PRINTOUT FROM COFAC',10X,'NA=',I3,3X,'NB=',I3/10X,1
     *9(1H*))
      NA1=NA+1
      IF(IB.EQ.0) NB1=NB
      IF(IB.EQ.1) NB1=NB+1
      N1=NA1+NB1
      IA1=22
      T3(1)=1.
      IF(IB.EQ.1) GO TO 11
      DO 2 I=2,N1
      T3(I)=T(I-1)
      DO 2 J=2,N1
    2 PT3(I,J)=PT(I-1,J-1)
      DO 3 I=1,N1
      PT3(I,1)=0.
    3 PT3(1,I)=0.
      GO TO 21
   11 CONTINUE
      DO 13 I=2,NA1
      T3(I)=T(I-1)
      DO 13 J=2,NA1
   13 PT3(I,J)=PT(I-1,J-1)
      NA2=NA1+1
      T3(NA2)=1.
      DO 14 I=1,N1
      PT3(I,1)=0.
   14 PT3(1,I)=0.
      NA3=NA2+1
      DO 15 I=NA3,N1
      T3(I)=T(I-2)
      DO 15 J=NA3,N1
   15 PT3(I,J)=PT(I-2,J-2)
      DO 16 I=1,N1
      PT3(I,NA2)=0.
   16 PT3(NA2,I)=0.
      DO 17 I=2,NA1
      DO 17 J=1,NB1
      PT3(I,J+NA2)=PT(I-1,J+NA2-2)
   17 PT3(J+NA2,I)=PT(J+NA2-2,I-1)
   21 CONTINUE
      DO 22 I=1,N1
      T4(I)=T3(I)
      DO 22 J=1,N1
   22 PT4(I,J)=PT3(I,J)
      IF(IPR.LE.0) GO TO 24
C
C     PRINT T AND PT.
C
      PRINT 200
  200 FORMAT(/10X,'VECTOR T')
      PRINT 201,(T3(I),I=1,N1)
  201 FORMAT(8X,10G12.5)
      PRINT 202
  202 FORMAT(/10X,'COVARIANCE MATRIX')
      DO 23 I=1,N1
   23 PRINT 201,(PT3(I,J),J=1,N1)
   24 CONTINUE
C
C     COMPUTE THE NEW A- AND B-POLYNOMIALS, VERSION 1.
C
      CALL EUKL1(T3,PT3,T1,TT1,NA1,NB1,NMA1,NMB1,NC1,IPR,IERR,IA1,IB)
      IF(IPR.LE.0) GO TO 28
```

```
      PRINT THE NEW VECTOR T.

      PRINT 150,NMA1,NMB1
  150 FORMAT(/10X,'THE NEW ESTIMATED VECTOR T',10X,'NMA=',I3,3X,'NMB=',
     *I3)
      NN=NMA1+NMB1
      PRINT 160,(T1(I),I=1,NN)
  160 FORMAT(7X,10G12.5)
   28 CONTINUE

      COMPUTE THE NEW A- AND B-POLYNOMIALS, VERSION 2.

      CALL EUKL2(T4,PT4,T2,TT2,NA1,NB1,NMA2,NMB2,NC2,IPR,IERR,IA1,IB)
      IF(IPR.LE.0) RETURN

      PRINT THE NEW VECTOR T.

      NN=NMA2+NMB2
      PRINT 150,NMA2,NMB2
      PRINT 160,(T2(I),I=1,NN)
      RETURN
      END
```

```
      SUBROUTINE EUKL1(X1,PX1,X2,TT,NA1,NB1,NMA,NMB,NC,IPR,IERR,IA1,IB)
C
C     THIS SUBROUTINE COMPUTES COMMON FACTORS TO TWO GIVEN POLYNOMIALS BY
C     USING THE EUCLIDEAN ALGORITHM. THE DIVISOR-POLYNOMIAL IS NORMA-
C     LIZED, I.E. THE HIGHEST DEGREE COEFFICIENT IS 1. TO DECIDE WHETHER
C     TWO POLYNOMIALS HAVE COMMON FACTORS OR NOT, THE REST-POLY-
C     NOMIAL COMPUTED AT EVERY DIVISION IS TESTED TO BE ZERO ON THE ASSUMPTION
C     THAT THE COEFFICIENTS IN THE ORIGINAL POLYNOMIALS ARE GAUSSIAN.
C     IF THE TWO GIVEN POLYNOMIALS HAVE COMMON FACTORS THEY ARE ABBRE-
C     VIATED WITH THE GREATEST COMMON POLYNOMIAL.
C     AUTHOR ERIK BURSTRÖM 1972-12-24
C
C     X1=(1,A(1),....,A(NA),B(0),....,B(NB)) VECTOR OF ORDER (NA1+NB1). IF
C     B(0)=0 B(0) IS OMITTED.
C     PX1 MATRIX OF ORDER (NA1+NB1)*(NA1+NB1) THE COVARIANCE MATRIX
C     OF X1.
C     X2 VECTOR OF ORDER (NMA+NMB) CONTAINING THE NEW ESTIMATED A- RESP. B-
C     POLYNOMIALS.
C     TT STATISTICAL TEST QUANTITY.
C     NA1 NUMBER OF A-COEFFICIENTS (MIN 2, MAX 11).
C     NB1 NUMBER OF B-COEFFICIENTS (MIN 1, MAX 11).
C     NMA NUMBER OF NEW A-COEFFICIENTS.
C     NMB NUMBER OF NEW B-COEFFICIENTS.
C     NC NUMBER OF DEGREES OF FREEDOM.
C     IPR  IF IPR=0 NOTHING IS PRINTED.
C          IF IPR=1 THE STATIC GAIN FOR THE ORIGINAL SYSTEM, THE TEST QUANTITY
C            AND THE NUMBER OF DEGREES OF FREEDOM AT EACH DIVISION DOWN TO
C            THE FIRST POSITIVE TEST, THE NEW A-POLYNOMIAL, DEGREE OF A AND
C            POLES, THE NEW B-POLYNOMIAL, DEGREE OF B AND ZEROS AND THE STATIC
C            GAIN FOR THE NEW SYSTEM ARE PRINTED.
C          IF IPR=2 AS IPR=1 + THE REST-POLYNOMIAL AND CORRESPONDING
C            COVARIANCE MATRIX OBTAINED AT EACH DIVISION AND THE TWO REST-POLY-
C            NOMIALS OBTAINED AT THE FINAL ABBREVIATION ARE PRINTED.
C     IERR IF IERR=0 NORMAL OUTPUT.
C          IF IERR=-1 EITHER IS THE COVARIANCE MATRIX PX1 NOT POSITIVE
C          DEFINITE, OR IS IT IMPOSSIBLE TO COMPUTE POLES OR ZEROS TO
C          THE NEW SYSTEM.
C     IA1 DIMENSION PARAMETER OF X1 AND PX1.
C     IB IF IB=0 B(0)=0 IS ASSUMED.
C        IF IB=1 B(0)=1 IS ASSUMED.
C
C     SUBROUTINE REQUIRED
C          PDV
C          DESYM
C          SOLVS
C          ROT
C
      DIMENSION X1(1),PX1(IA1,IA1),X2(1)
C
      DIMENSION C(10),PC(10,10),Y(10),H1(11,11),H2(11,11),D(11,11),E(11,
     *11),DYDX(22,22),V(10,10),R(10),DVNDA(11),DVNDB(11),DIVI(11),S(21,
     *21),PX2(21,21),R1(10)
      REAL K1,K2,K3
      TJITWO(TN)=1.538*TN+3.2
      IERR=0
      G=0.
      G1=0.
      G2=0.
      G32=0.
      G21=1.
      IF(IB.EQ.0)G22=0.
      IF(IB.EQ.1)G22=1.
      IA2=10
```

```
      NA=NA1
      NB=NB1
      NC=NB1-1
      IF(IB.EQ.0.AND.NB1.EQ.1)GO TO 68
      IF(NC.LE.0) RETURN
      IF(IPR.GT.0) PRINT 600,X1(NA+1)
600   FORMAT(/10X,'VERSION 1  THE B-POLYNOMIAL IS SCALED  B(0)= ',G10.5/
     *10X,5(2H- ))
      DO 3 I=1,NA1
      G1=G1+X1(I)
    3 DVNDA(I)=X1(I)
      NA2=NA1+1
      N1=NA1+NB1
      DO 4 I=1,NB1
      G2=G2+X1(I+NA1)
    4 DVNDB(I)=X1(I+NA1)

      COMPUTE THE STATIC GAIN.

      G=G2/G1
      IF(IPR.GT.0)PRINT 601,G
 601  FORMAT(/10X,'STATIC GAIN',5X,G10.5)
      DO 5 I=1,11
      DO 5 J=1,11
    5 D(I,J)=0.
      DO 6 I=1,22
      DO 6 J=1,22
    6 DYDX(I,J)=0.
      DO 7 I=1,10
    7 R(I)=0.
      DO 8 I=1,11
      DO 8 J=1,11
      E(I,J)=0.
      IF(I.NE.J) GO TO 8
      E(I,J)=1.
    8 CONTINUE
   18 CONTINUE
      IF(IB.EQ.1.AND.NA.EQ.NB) GO TO 108
C
C     COMPUTE REST-POLYNOMIAL AND CORRESPONDING COVARIANCE MATRIX.
C
      K1=X1(1)/X1(NA+1)
      K2=1./X1(NA+1)*(X1(2)-K1*X1(NA+2))
      NA2=NA
      IF(NC.LE.1) GO TO 12
      N3=NC-1
      DO 11 I=1,N3
   11 C(I)=X1(I+2)-K1*X1(I+NA+2)-K2*X1(I+NA+1)
      C(NC)=X1(NA)-K2*X1(NA+NB)
      GO TO 17
   12 C(1)=X1(3)-K2*X1(5)
   17 CONTINUE
      DO 213 I=1,NC
      DO 212 J=1,NA
  212 H1(I,J)=0.
      DO 213 J=1,NB
  213 H2(I,J)=0.
      DO 13 I=1,N3
      H1(I,I+2)=1.
      H1(I,1)=-1./X1(NA+1)*X1(I+NA+2)+X1(NA+2)/X1(NA+1)**2*X1(I+NA+1)
   13 H1(I,2)=-1./X1(NA+1)*X1(I+NA+1)
      H1(NC,1)=X1(NA+2)/X1(NA+1)**2*X1(NA+NB)
      H1(NC,2)=-1./X1(NA+1)*X1(NA+NB)
      H1(NC,NB)=1.
```

```
      DO 14 I=1,N3
      H2(I,I+2)=-X1(1)/X1(NA+1)
   14 H2(I,1)=X1(1)/X1(NA+1)**2*X1(I+NA+2)+X1(2)/X1(NA+1)**2*X1(I+NA+1)-
     *2.*X1(1)*X1(NA+2)/X1(NA+1)**3*X1(I+NA+1)
      H2(2,1)=-X1(2)/X1(NA+1)+2.*X1(1)*X1(NA+2)/X1(NA+1)**2
      DO 15 I=2,NC
      H2(I,2)=X1(1)*X1(I+NA+1)/X1(NA+1)**2
   15 H2(I,I+1)=-X1(2)/X1(NA+1)+X1(1)*X1(NA+2)/X1(NA+1)**2
      H2(NC,1)=X1(2)/X1(NA+1)**2*X1(NA+NB)-2.*X1(1)*X1(NA+2)/X1(NA+1)**3
     **X1(NA+NB)
      GO TO 19
  108 CONTINUE
      K3=X1(1)/X1(NA+1)
      DO 111 I=1,NC
  111 C(I)=X1(I+1)-K3*X1(I+NA+1)
      DO 112 I=1,NC
      DO 112 J=1,NA1
      H1(I,J)=0.
  112 H2(I,J)=0.
      DO 113 I=1,NC
      H1(I,1)=-1./X1(NA+1)*X1(I+NA+1)
      H1(I,I+1)=1.
      H2(I,1)=X1(1)/X1(NA+1)**2*X1(I+NA+1)
  113 H2(I,I+1)=-X1(1)/X1(NA+1)
      NA2=NA
   19 CONTINUE
      DO 20 I=1,NC
      DO 20 J=1,NA2
   20 DYDX(I,J)=H1(I,J)
      DO 22 I=1,NC
      DO 22 J=1,NB
   22 DYDX(I,J+NA2)=H2(I,J)
      DO 24 I=1,NB
      DO 24 J=1,NA2
   24 DYDX(I+NC,J)=D(I,J)
      DO 26 I=1,NB
      DO 26 J=1,NB
   26 DYDX(I+NC,J+NA2)=E(I,J)
      N4=NB+NC
      N5=NA2+NB
      DO 30 I=1,N4
      DO 30 J=1,N5
      S(I,J)=0.
      DO 30 K=1,N5
   30 S(I,J)=S(I,J)+DYDX(I,K)*PX1(K,J)
      DO 32 I=1,N4
      DO 32 J=1,N4
      PX2(I,J)=0.
      DO 32 K=1,N5
   32 PX2(I,J)=PX2(I,J)+S(I,K)*DYDX(J,K)
      IF(IPR.LE.1) GO TO 33
C
C     PRINT RESULTS.
C
      PRINT 2900
 2900 FORMAT(/10X,'VECTOR C')
      PRINT 3000,(C(I),I=1,NC)
 3000 FORMAT(7X,10G12.5)
      PRINT 2950
 2950 FORMAT(/10X,'COVARIANCE MATRIX')
      DO 3010 I=1,NC
 3010 PRINT 3000,(PX2(I,J),J=1,NC)
   33 CONTINUE
      DO 35 I=1,NC
```

```
      DO 35 J=1,NC
   35 PC(I,J)=PX2(I,J)
      EPS=10.**-07*PC(1,1)

      COMPUTE TEST QUANTITY.

      CALL DESYM(PC,V,NC,EPS,IERR,IA2)
      IF(IERR.EQ.-1)GO TO 99
      CALL SOLVS(V,C,Y,NC,1,IA2)
      TT=0.
      DO 36 I=1,NC
   36 TT=TT+Y(I)*C(I)
      IF(IPR.GT.0) PRINT 101,TT,NC
  101 FORMAT(/10X,'TEST QUANTITY',G10.5,10X,'DEGREES OF FREEDOM',I5/10X,
     *23(1H-))
      TNC=FLOAT(NC)
      IF(TT.GT.TJITWO(TNC)) GO TO 60
      DO 42 I=1,NB
   42 DIVI(I)=X1(I+NA2)

      COMPUTE THE NEW A-POLYNOMIAL.

      IK=1
      CALL PDV(DVNDA,DIVI,R,NA1,NB,IDR,IPR,IERR,IK)
      DO 44 I=1,IDR
      G21=G21+R(I)
   44 X2(I)=R(I)
C
C     COMPUTE THE NEW B-POLYNOMIAL.
C
      IK=2
      CALL PDV(DVNDB,DIVI,R1,NB1,NB,IDR1,IPR,IERR,IK)
      DO 46 I=1,IDR1
      G22=G22+R1(I)
   46 X2(I+IDR)=R1(I)
      NMA=IDR
      NMB=IDR1
      G32=G22/G21
      IF(IPR.GT.0)PRINT 601,G32
      GO TO 70
C
C     TEST IS READY. THE POLYNOMIALS HAVE COMMON FACTORS.
C
   60 CONTINUE
      IF(NC.LE.1) GO TO 68
      DO 61 I=1,NB
   61 X1(I)=X1(I+NA2)
      DO 62 I=1,NC
   62 X1(I+NB)=C(I)
      DO 64 I=1,NB
      DO 64 J=1,NB
   64 PX1(I,J)=PX2(I+NC,J+NC)
      DO 65 I=1,NC
      DO 65 J=1,NC
   65 PX1(I+NB,J+NB)=PX2(I,J)
      DO 66 I=1,NB
      DO 66 J=1,NC
      PX1(I,J+NB)=PX2(J,I+NC)
   66 PX1(J+NB,I)=PX2(I+NC,J)
      NA=NB
      NB=NC
      NC=NC-1
      IF(IPR.LE.2) GO TO 18
      PRINT 260
```

```
260 FORMAT(/10X,'THE NEW ESTIMATED VECTOR T')
    N7=NA+NB
    PRINT 3000,(X1(I),I=1,N7)
    PRINT 262
262 FORMAT(/10X,'COVARIANCE MATRIX')
    DO 72 I=1,N7
 72 PRINT 3000,(PX1(I,J),J=1,N7)
    GO TO 18


 68 IF(IPR.GT.0) PRINT 160
160 FORMAT(/10X,'NO COMMON FACTORS')
    NM=NA1+NB1
    DO 69 I=1,NM
 69 X2(I)=X1(I)
    NMA=NA1
    NMB=NB1
 70 GO TO 1000


 99 IF(IPR.GT.0) PRINT 199
199 FORMAT(/10X,'DECOMPOSITION IS IMPOSSIBLE')
    NM=NA1+NB1
    DO 79 I=1,NM
 79 X2(I)=X1(I)
    NMA=NA1
    NMB=NB1
    RETURN
C
C
1000 CONTINUE
    RETURN
    END
```

```
      SUBROUTINE EUKL2(X1,PX1,X2,TT,NA1,NB1,NMA,NMB,NC,IPR,IERR,IA1,IB)
C
C     THIS SUBROUTINE COMPUTES COMMON FACTORS TO TWO GIVEN POLYNOMIALS BY
C     USING THE EUCLIDEAN ALGORITHM. THE DIVISOR-POLYNOMIAL IS NOT NORMA-
C     LIZED, I.E. THE SUCCECIVE DIVISIONS ARE DONE STRAIGHTFORWARD.
C     TO DECIDE WHETHER TWO POLYNOMIALS HAVE COMMON FACTORS OR NOT,
C     THE REST-POLYNOMIALS COMPUTED AT EACH DIVISION IS TESTED TO BE ZERO
C     ON THE ASSUMPTION THAT THE COEFFICIENTS OF THE ORIGINAL POLYNOMIALS
C     ARE GAUSSIAN. IF THE ORIGINAL POLYNOMIALS HAVE COMMON FACTORS, THEY
C     ARE ABBREVIATED WITH THE GREATEST COMMON POLYNOMIAL.
C     AUTHOR ERIK BURSTRÖM 1972-12-24
C
C     X1=(1,A(1),....,A(NA),B(0),B(1),....,B(NB)) VECTOR OF ORDER (NA1+NB1)
C     CONTAINING THE COEFFICIENTS OF THE ORIGINAL POLYNOMIALS. IF B(0)=0,
C     B(0) IS OMITTED.
C     PX1 MATRIX OF ORDER (NA1+NB1)*(NA1+NB1) THE COVARIANCE MATRIX
C     OF X1.
C     X2 VECTOR OF ORDER (NMA+NMB) CONTAINING THE COEFFICIENTS OF THE
C     NEW ESTIMATED A- RESP. B-POLYNOMIALS.
C     TT STATISTICAL TEST QUANTITY.
C     NA1 NUMBER OF A-COEFFICIENTS (MIN 2, MAX 10).
C     NB1 NUMBER OF B-COEFFICIENTS (MIN 1, MAX 11).
C     NMA NUMBER OF NEW A-COEFFICIENTS.
C     NMB NUMBER OF NEW B-COEFFICIENTS.
C     NC NUMBER OF DEGREES OF FREEDOM.
C     IPR   IF IPR=0 NOTHING IS PRINTED.
C           IF IPR=1 THE STATIC GAIN FOR THE ORIGINAL SYSTEM, THE TEST QUANTITY
C             AND CORRESPONDING NUMBER OF DEGREES OF FREEDOM AT EACH DIVISION
C             DOWN TO THE FIRST POSITIVE TEST, THE NEW A-POLYNOMIAL, DEGREE OF A
C             AND POLES, THE NEW B-POLYNOMIAL, DEGREE OF B AND ZEROS AND
C             THE STATIC GAIN FOR THE NEW SYSTEM ARE PRINTED.
C           IF IPR=2 AS IPR=1 + THE REST-POLYNOMIAL AND CORRESPONDING
C             COVARIANCE MATRIX OBTAINED AT EACH DIVISION AND THE TWO REST-POLY-
C             NOMIALS OBTAINED AT THE FINAL ABBREVIATION ARE PRINTED.
C     IERR IF IERR=0 NORMAL OUTPUT.
C           IF IERR=-1 EITHER IS THE COVARIANCE MATRIX PX1 NOT POSITIVE
C           DEFINITE OR IS IT IMPOSSIBLE TO COMPUTE POLES OR ZEROS
C           FOR THE NEW SYSTEM.
C     IA1 DIMENSION PARAMETER OF X1 AND PX1
C     IB IF IB =0 B(0)=0 IS ASSUMED.
C        IF IB=1 B(0)=1 IS ASSUMED.
C
C     SUBROUTINE REQUIRED
C           PDV
C           DESYM
C           SOLVS
C           ROT
C
      DIMENSION X1(1),PX1(IA1,IA1),X2(1)
C
      DIMENSION C(10),PC(10,10),Y(10),H1(11,11),H2(11,11),D(11,11),E(11,
     *11),DYDX(22,22),V(10,10),R(10),DVNDA(11),DVNDB(11),DIVI(11),S(21,
     *21),PX2(21,21),R1(10)
      REAL K1,K2,K3
      TJITWO(TN)=1.538*TN+3.2
      IA2=10
      IERR=0
      G=0.
      G1=0.
      G2=0.
      G20=0.
      G21=1.
      IF(IB.EQ.0)G22=0.
```

```
      IF(IB.EQ.1)G22=1.
      NA=NA1
      NB=NB1
      NC=NB1-1
      NC1=NB1-1
      IF(IB.EQ.0.AND.NB1.EQ.1)GO TO 68
      IF(NC.LE.0) RETURN
      IF(IPR.GT.0) PRINT 600,X1(NA+1)
  600 FORMAT(/10X,'VERSION 2  THE B-POLYNOMIAL IS UNSCALED  B(0)= ',G10.
     *5/10X,5(2H- ))

      COMPUTE THE STATIC GAIN FOR THE ORIGINAL SYSTEM.

      DO 3 I=1,NA1
      G1=G1+X1(I)
    3 DVNDA(I)=X1(I)
      DO 4 I=1,NB1
      G2=G2+X1(I+NA1)
    4 DVNDB(I)=X1(I+NA1)
      G=G2/G1
      IF(IPR.GT.0)PRINT 601,G
  601 FORMAT(/10X,'STATIC GAIN',5X,G10.5)
      DO 5 I=1,11
      DO 5 J=1,11
    5 D(I,J)=0.
      DO 6 I=1,22
      DO 6 J=1,22
    6 DYDX(I,J)=0.
      DO 7 I=1,10
    7 R(I)=0.
      DO 8 I=1,11
      DO 8 J=1,11
      E(I,J)=0.
      IF(I.NE.J) GO TO 8
      E(I,J)=1.
    8 CONTINUE
   14 CONTINUE
      IF(IB.EQ.1.AND.NA.EQ.NB) GO TO 108
C
C     COMPUTE THE REST-POLYNOMIAL AND CORRESPONDING COVARIANCE MATRIX.
C
      K1=-X1(NA+1)**2
      K2=X1(2)*X1(NA+1)-X1(1)*X1(NA+2)
      K3=X1(1)*X1(NA+1)
      IF(NC.LE.1) GO TO 51
      NC1=NC-1
      DO 11 I=1,NC1
   11 C(I)=K1*X1(I+2)+K2*X1(I+NA+1)+K3*X1(I+NA+2)
      C(NC)=K1*X1(NC+2)+K2*X1(NA+NB)
      GO TO 27
   51 C(1)=K1*X1(3)+K2*X1(5)
      H1(1,1)=-X1(NA+NB)**2
      H1(1,2)=X1(NA+1)*X1(NA+2)
      H1(1,3)=K1
      H2(1,1)=-2.*X1(NA+1)*X1(3)+X1(2)*X1(NA+2)
      H2(1,2)=-X1(1)*X1(NA+2)+X1(2)*X1(NA+1)-X1(1)*X1(NA+2)
      GO TO 115
   27 CONTINUE
C
C
      NA2=NA
      DO 16 I=1,NC
      DO 15 J=1,NA2
   15 H1(I,J)=0.
```

```
      DO 16 J=1,NB
   16 H2(I,J)=0.
      NC1=NC-1
      DO 17 I=1,NC1
      H1(I,1)=X1(NA+1)*X1(I+NA+2)-X1(NA+2)*X1(I+NA+1)
   17 H1(I,2)=X1(NA+1)*X1(I+NA+1)
      H1(NC,1)=-X1(NA+2)*X1(NA+NB)
      H1(NC,2)=X1(NA+1)*X1(NA+NB)
      DO 19 I=1,NC
   19 H1(I,I+2)=-X1(NA+1)**2


      DO 21 I=1,NC1
      H2(I,I+2)=X1(1)*X1(NA+1)
   21 H2(I,1)=-2.*X1(NA+1)*X1(I+2)+X1(1)*X1(I+NA+2)+X1(2)*X1(I+NA+1)
      H2(NC,1)=-2.*X1(NA+1)*X1(NA)+X1(2)*X1(NA+NB)
      H2(1,2)=-X1(1)*X1(NA+2)+K2
      DO 22 I=2,NC
      H2(I,I+1)=-X1(1)*X1(NA+2)
   22 H2(I,2)=-X1(1)*X1(I+NA+1)
      GO TO 26
  108 CONTINUE
      K1=X1(NA+1)
      K2=-X1(1)
      DO 111 I=1,NC
  111 C(I)=K1*X1(I+1)+K2*X1(I+NA+1)
      DO 112 I=1,NC
      DO 112 J=1,NA
      H1(I,J)=0.
  112 H2(I,J)=0.
      DO 114 I=1,NC
      H1(I,1)=-X1(I+NA+1)
      H2(I,1)=X1(I+1)
      H1(I,I+1)=K1
  114 H2(I,I+1)=K2
  115 CONTINUE
   26 CONTINUE
      NA2=NA
      DO 31 I=1,NC
      DO 31 J=1,NA2
   31 DYDX(I,J)=H1(I,J)
      DO 32 I=1,NC
      DO 32 J=1,NB
   32 DYDX(I,J+NA2)=H2(I,J)
      DO 33 I=1,NB
      DO 33 J=1,NA2
   33 DYDX(I+NC,J)=D(I,J)
      DO 34 I=1,NB
      DO 34 J=1,NB
   34 DYDX(I+NC,J+NA2)=E(I,J)
   35 CONTINUE
      I5=NB+NC
      I6=NA+NB
      DO 36 I=1,I5
      DO 36 J=1,I6
      S(I,J)=0.
      DO 36 K=1,I6
   36 S(I,J)=S(I,J)+DYDX(I,K)*PX1(K,J)
      DO 38 I=1,I5
      DO 38 J=1,I5
      PX2(I,J)=0.
      DO 38 K=1,I6
   38 PX2(I,J)=PX2(I,J)+S(I,K)*DYDX(J,K)
      DO 40 I=1,NC
```

```
      DO 40 J=1,NC
   40 PC(I,J)=PX2(I,J)
      IF(IPR.LE.1) GO TO 39

      PRINT RESULTS.

      PRINT 1900
 1900 FORMAT(/10X,'VECTOR C')
      PRINT 2000,(C(I),I=1,NC)
 2000 FORMAT(7X,10G12.5)
      PRINT 1950
 1950 FORMAT(/10X,'COVARIANCE MATRIX')
      DO 41 I=1,NC
   41 PRINT 2000,(PC(I,J),J=1,NC)
   39 CONTINUE

      COMPUET TEST QUANTITY.

      EPS=10.**-07*PC(1,1)
      CALL DESYM(PC,V,NC,EPS,IRANK,IA2)
      IF(IRANK.EQ.-1) GO TO 99
      CALL SOLVS(V,C,Y,NC,1,IA2)
      TT=0.
      DO 42 I=1,NC
   42 TT=TT+Y(I)*C(I)
      IF(IPR.GT.0) PRINT 101,TT,NC
  101 FORMAT(/10X,'TEST QUANTITY',G10.5,10X,'DEGREES OF FREEDOM',I5/10X,
     *23(1H-))
      TNC=FLOAT(NC)
      IF(TT.GT.TJITWO(TNC)) GO TO 60
      DO 44 I=1,NB
   44 DIVI(I)=X1(I+NA)
C
C     COMPUTE THE NEW A-POLYNOMIAL.
C
      IK=1
      CALL PDV(DVNDA,DIVI,R,NA1,NB,IDR,IPR,IERR,IK)
      DO 45 I=1,IDR
      G21=G21+R(I)
   45 X2(I)=R(I)
C
C     COMPUTE THE NEW B-POLYNOMIAL.
C
      IK=2
      CALL PDV(DVNDB,DIVI,R1,NB1,NB,IDR1,IPR,IERR,IK)
      DO 46 I=1,IDR1
      G22=G22+R1(I)
   46 X2(I+IDR)=R1(I)
      NMA=IDR
      NMB=IDR1
C
C     COMPUTE THE STATIC GAIN FOR THE NEW SYSTEM.
C
      G20=G22/G21
      IF(IPR.GT.0)PRINT 601,G20
      GO TO 70
C
C     TEST IS READY. THE POLYNOMIALS HAVE COMMON FACTORS.
C
   60 CONTINUE
      IF(NC.LE.1) GO TO 68
      DO 61 I=1,NB
   61 X1(I)=X1(I+NA)
      DO 62 I=1,NC
```

```
62 X1(I+NB)=C(I)
   DO 64 I=1,NB
   DO 64 J=1,NB
64 PX1(I,J)=PX2(I+NC,J+NC)
   DO 65 I=1,NC
   DO 65 J=1,NC
65 PX1(I+NB,J+NB)=PX2(I,J)
   DO 66 I=1,NB
   DO 66 J=1,NC
   PX1(I,J+NB)=PX2(J,I+NC)
66 PX1(J+NB,I)=PX2(I+NC,J)
   NA=NB
   NB=NC
   NC=NC-1
   IF(IPR.LE.2) GO TO 14
   PRINT 260
260 FORMAT(/10X,'THE NEW ESTIMATED VECTOR T')
   N7=NA+NB
   PRINT 2000,(X1(I),I=1,N7)
   PRINT 262
262 FORMAT(/10X,'THE COVARIANCE MATRIX')
   DO 72 I=1,N7
72 PRINT 2000,(PX1(I,J),J=1,N7)
   GO TO 14
C
C
68 IF(IPR.GT.0) PRINT 160
160 FORMAT(/10X,'NO COMMON FACTORS')
   NM=NA1+NB1
   DO 69 I=1,NM
69 X2(I)=X1(I)
   NMA=NA1
   NMB=NB1
70 GO TO 100
C
C
99 IF(IPR.GT.0) PRINT 199
199 FORMAT(/10X,'DECOMPOSITION IS IMPOSSIBLE')
   NM=NA1+NB1
   DO 79 I=1,NM
79 X2(I)=X1(I)
   NMA=NA1
   NMB=NB1
C
C
100 RETURN
   END
```

```
      SUBROUTINE PDV(DVND,DIVI,R,IDV,IDI,IDR,IPR,IERR,IK)
C
C     THIS SUBROUTINE COMPUTES THE NEW A- RESP. B-POLYNOMIAL BY ABBREVIA-
C     TING THE ORIGINAL POLYNOMIAL WITH THE GREATEST COMMON POLYNOMIAL
C     COMPUTED IN SUBROUTINE EUKL1 OR IN SUBROUTINE EUKL2. THE REST-POLYNOMIAL
C     IS ALSO COMPUTED. THIS SUBROUTINE MUST BE CALLED EITHER FROM SUBROUTINE
C     EUKL1 OR FROM SUBROUTINE EUKL2.
C     AUTHOR ERIK BURSTRÖM 1972-12-24
C
C     DVND VECTOR OF ORDER IDV CONTAINING THE COEFFICIENTS OF THE DIVIDEND
C     POLYNOMIAL.
C     DIVI VECTOR OF ORDER IDV GONTAINING THE COEFFICIENTS OF THE DIVISOR
C     POLYNOMIAL.
C     R VECTOR OF ORDER IDR AT OUTPUT CONTAINGING THE COEFFICIENTS
C     OF THE QUOTE-POLYNOMIAL.
C     IDV NUMBER OF COEFFICIENTS OF THE DIVIDEND POLYNOMIAL.
C     IDI NUMBER OF COEFFICIENTS OF THE DIVISOR POLYNOMAIL.
C     IDR NUMBER OF COEFFICIENTS OF THE QUOTE POLYNOMIAL.
C     IPR  IF IPR=0 NOTHING IS PRINTED.
C          IF IPR=1 THE NEW ESTIMATED QUOTE-POLYNOMIAL, DEGREE AND ROOTS
C           OF IT ARE PRINTED.
C          IF IPR=2 AS IPR=1 + THE REST-POLYNOMIAL.
C     IERR IF IERR=0 NORMAL OUTPUT.
C          IF IERR=-1 COMPUTATON OF THE ROOTS HAS FAILED.
C     IK   IF IK=1 THE NEW A-POLYNOMIAL IS COMPUTED.
C          IF IK=2 THE NEW B-POLYNOMIAL IS COMPUTED.
C
C
C     SUBROUTINE REQUIRED
C          ROT
C
      DIMENSION DVND(1),DIVI(1),R(1)
C
      DIMENSION A(10),A1(10),D(10),RE(10),A2(10),AR2(10),AI2(10)
      EPS=1.0E-05
      NERR1=0
C
C     COMPUTE THE QUOTE-POLYNOMIAL.
C
      IDA=IDV-1
      IF(IK.EQ.2.AND.IB.EQ.0) GO TO 3
      A0=DVND(1)
      DO 2 I=1,IDA
    2 A(I)=DVND(I+1)
      GO TO 6
    3 A0=0.
      DO 5 I=1,IDV
    5 A(I)=DVND(I)
    6 CONTINUE
      IDD=IDI-1
      D0=1.
      DO 4 I=1,IDD
    4 D(I)=DIVI(I+1)/DIVI(1)
      A10=A0/D0
      I3=IDA-IDD
      IF(IDD.EQ.IDA) GO TO 80
      IF(I3.GT.IDD) GO TO 31
      A1(1)=(A(1)-D(1)*A10)/D0
      IF(I3.LE.1) GO TO 26
      DO 24 I=2,I3
      A1(I)=A(I)
      IJ=I-1
      DO 23 J=1,IJ
```

```
 23 A1(I)=A1(I)-D(J)*A1(I-J)
    A1(I)=(A1(I)-D(I)*A10)/D0
 24 CONTINUE
 26 CONTINUE
    IF(IK.EQ.1.OR.ABS(A10).GT.EPS) GO TO 39
    A1(I3+1)=A(I3+1)
    DO 27 J=1,I3
 27 A1(I3+1)=A1(I3+1)-D(J)*A1(-J+I3+1)
    A1(I3+1)=(A1(I3+1)-D(I3+1)*A10)/D0
    GO TO 39
 31 CONTINUE
    A1(1)=(A(1)-D(1)*A10)/D0
    IF(IDD.LE.1) GO TO 34
    DO 33 I=2,IDD
    A1(I)=A(I)
    IJ=I-1
    DO 32 J=1,IJ
 32 A1(I)=A1(I)-D(J)*A1(I-J)
    A1(I)=(A1(I)-D(I)*A10)/D0
 33 CONTINUE
 34 IL=IDD+1
    DO 36 I=IL,I3
    A1(I)=A(I)
    DO 35 J=1,IDD
 35 A1(I)=A1(I)-D(J)*A1(I-J)
    A1(I)=(A1(I)-D(I)*A10)/D0
 36 CONTINUE
    IF(IK.EQ.1.OR.ABS(A10).GT.EPS) GO TO 39
    A1(I3+1)=A(I3+1)
    DO 37 J=1,IDD
 37 A1(I3+1)=A1(I3+1)-D(J)*A1(-J+I3+1)
    A1(I3+1)=A1(I3+1)/D0
 39 CONTINUE
C
C     COMPUTE THE ROOTS.
C
    IF(ABS(A10).LT.EPS) GO TO 55
    DO 51 I=1,I3
 51 A2(I)=A1(I)/A10
    GO TO 57
 55 CONTINUE
    DO 56 I=1,I3
 56 A2(I)=A1(I+1)/A1(1)
 57 CONTINUE
    CALL ROT(I3,1,A2,A2,AR2,AI2,NERR1)
    IF(NERR1.GT.0) GO TO 2000
    IF(IK.EQ.2) GO TO 61
    A20=1.
    IF(IPR.GT.0) PRINT 101,I3,A20
101 FORMAT(/10X,'THE NEW A-POLYNOMIAL',10X,'DEGREE OF A',I3,10X,'ROOTS
   * OF A'//10X,'A(0)= ',G10.5)
    IF(I3.LE.0) RETURN
C
C     PRINT RESULTS.
C
    IF(IPR.GT.0) PRINT 102,((I,A2(I),AR2(I),AI2(I)),I=1,I3)
102 FORMAT(/10X,'A(',I1,')= ',G10.5,37X,2G10.5)
    DO 59 I=1,I3
 59 R(I)=A2(I)
    IDR=I3
    GO TO 41
 61 IF(IPR.GT.0) PRINT 103,I3,A10
103 FORMAT(/10X,'THE NEW B-POLYNOMIAL',10X,'DEGREE OF B',I3,10X,'ROOTS
   * OF B'//10X,'B(0)= ',G10.5)
```

```
      IF(I3.LE.0) RETURN
      IF(IPR.GT.0) PRINT 104,((I,A1(I),AR2(I),AI2(I)),I=1,I3)
      I4=I3+1
      IF(IPR.GT.0.AND.IK.EQ.2.AND.ABS(A10).LT.EPS) PRINT 104,(I4,A1(I4))
  104 FORMAT(/10X,'B(',I1,')= ',G10.5,37X,2G10.5)
      IF(IB.EQ.1) GO TO 63
      DO 62 I=1,I4
   62 R(I)=A1(I)
      IDR=I4
      GO TO 41
   63 CONTINUE
      DO 64 I=1,I3
   64 R(I)=A1(I)
      IDR=I3
   41 IF(IPR.LE.1)GO TO 200

      COMPUTE THE REST-POLYNOMIAL.

      RE0=A(I3+1)
      DO 42 I=1,I3
   42 RE0=RE0-D(I)*A1(-I+I3+1)
      RE0=RE0-D(I3+1)*A10
      IF(IDD.LE.1) GO TO 47
      I6=IDA-1-2*I3
      DO 44 I=1,I6
      RE(I)=A(I+I3+1)
      DO 43 J=1,I3
   43 RE(I)=RE(I)-D(J+I)*A1(-J+I3+1)
      RE(I)=RE(I)-D(I3+I+1)*A10
   44 CONTINUE
      I7=I6+1
      I8=IDD-1
      DO 46 I=I7,I8
      RE(I)=A(I+I3+1)
      DO 45 J=1,I8
   45 RE(I)=RE(I)-D(J+I)*A1(-J+I3+1)
   46 CONTINUE
   47 CONTINUE
      PRINT 111,I8,RE0
  111 FORMAT(/10X,'THE REST POLYNOMIAL',10X,'DEGREE OF REST',I3//10X,'RE
     *(0)= ',G10.5)
      IF(IDD.LE.1) RETURN
      PRINT 112,((I,RE(I)),I=1,I8)
  112 FORMAT(/10X,'RE(',I1,')= ',G10.5)
      GO TO 200
   80 CONTINUE
      IF(IPR.GT.0.AND.IK.EQ.1)PRINT 113,DVND(1)
  113 FORMAT(/10X,'THE A-POLYNOMIAL IS A CONSTANT',G10.5)
      IF(IPR.GT.0.AND.IK.EQ.2) PRINT 114,DVND(1)
  114 FORMAT(/10X,'THE B-POLYNOMIAL IS CONSTANT',G10.5)
      IDR=1
      R(1)=DVND(1)
  200 CONTINUE
      RETURN
 2000 IERR=-1
      IF(IK.EQ.1)PRINT 2001
 2001 FORMAT(/10X,'COMPUTATION OF POLES IS IMPOSSIBLE')
      IF(IK.EQ.2)PRINT 2002
 2002 FORMAT(/10X,'COMPUTATION OF ZEROS IS IMPOSSIBLE')
      RETURN
      END
```