A Computer Program for Simulation of Multivariable Self-Tuning Regulators

Borisson, Ulf

1976

Document Version:
Publisher's PDF, also known as Version of record

A COMPUTER PROGRAM FOR SIMULATION OF MULTIVARIABLE SELF-TUNING REGULATORS

U. BORISSON

Report 7615(C) March 1976
Department of Automatic Control
Lund Institute of Technology

A COMPUTER PROGRAM FOR SIMULATION OF MULTIVARIABLE SELF-TUNING REGULATORS

This report contains a listing of a simple computer program for simulation of multivariable self-tuning regulators. The results of the simulations are printed on line-printer, and they are also written in a file which can be plotted directly by IDPAC.

This version of the program is intended just for internal use. The program is started up on PDP15 as follows:

```
§ A NON 4,5,7,15,16
§ GLOAD
← STUREM,INDAM,PRIM,SYSM,STURM1,FILES
```

```
C       PROGRAM STUREM
C
C       SIMULATION OF SELF TUNING REGULATORS FOR MULTIVARIABLE SYSTEMS
C
C       AUTHOR: ULF BORISSON 1973-11-20
C
C       SUBROUTINES REQUIRED
C               INDAM
C               PRIM
C               SYSM
C               STURM1
C               FILES  (in the program library)
C
        COMMON YM(4,7),YSM(4,7),UM(4,10),EM(4,5),PM(16,16,4),TM(16,4),
       2 NA(4),NB(4),BO(4),KSYS(4),KREG(4),ASYS(4,4,4),BSYS(4,4,4),
       3 CSYS(4,4,4),N,NP,
       4 EVEC(4),COV(4,4),ILOG,MNOD,MACO,MAC,NABMAX,PDIAG,YSUM,
       5 RL,ULIM,NSAMP,IP1,IP2,ICONT,ICOUN,
       6 IP,IY,IU,IE,IPM,ITM,IABC,ICONS,JCONS(9,4),
       7 FNAMA,FNAMB,W(20),BUFF(32),KOD(10),ITAPE,IPOIN
C
        IP=4
        IY=7
        IU=10
        IE=5
        IPM=16
        ITM=16
        IABC=4
C
C       IN THIS VERSION IP AND IU ARE USED IN STURM1
C
10      CALL INDAM(1)
20      CALL PRIM(1)
C
        IF (ITAPE .EQ. 0) GO TO 30
        KOD(2)=2*NABMAX+1
        KOD(7)=1
C
        CALL FILES('ENTER',3,KOD,FNAMA,'BIN',LOG)
        CALL FILDAT('WRITE',3,KOD,5)
        IF (ITAPE.LE.1) GO TO 30
        KO2DT2=4
        KOD(2)=KO2DT2
        CALL FILES('ENTER',4,KOD,FNAMB,'BIN',LOG)
        CALL FILDAT('WRITE',4,KOD,5)
C
C       SET ARRAYS TO ZERO
C
30      YM(1,1)=0.
        CALL RMOVE(YM(1,1),1,YM(2,1),1,IP*IY-1)
        CALL RMOVE(YM(1,1),1,YSM(1,1),1,IP*IY)
        CALL RMOVE(YM(1,1),1,UM(1,1),1,IP*IU)
        CALL RMOVE(YM(1,1),1,EM(1,1),1,IP*IE)
        CALL RMOVE(YM(1,1),1,EVEC(1),1,IP)
C
        ILOG=0
        MNOD=0
        ICOUN=0
        YSUM=0.
C
        MAC=MACO
C
C       REGULATOR LOOP
```

```
C
90      DO 500 ILOOP=1,NSAMP
        ICOUN=ICOUN+1
C
C       SIMULATE THE SYSTEM
C
        CALL SYSM
C
C       NO CONTROL IF PDIAG IS NEGATIVE
C
        IF (PDIAG) 162,98,98
C
98      DO 100 I=1,NP
100     YSM(I,1)=YM(I,1)/BO(I)
C
        CALL STURM1(YSM,UM,TM,PM,RL,NP,NA,NB,KREG,IPM)
C
C       MAKE SOME COVARIANCE MATRIX ENTRIES ZERO TO GET THE DESIRED
C       CONSTANT REGULATOR PARAMETERS
C
        IF (ICONS.EQ.0) GO TO 116
        DO 108 J=1,NP
        I1=JCONS(1,J)
        DO 108 I=1,I1
        IPOS=JCONS(I+1,J)
        DO 106 J1=1,IPM
106     PM(IPOS,J1,J)=0.
        DO 108 J1=1,IPM
108     PM(J1,IPOS,J)=0.
C
116     IF (ULIM) 162,162,120
120     DO 160 I=1,NP
        IF (UM(I,1)-ULIM) 140,140,130
130     UM(I,1)=ULIM
        GO TO 160
140     IF (UM(I,1)+ULIM) 150,160,160
150     UM(I,1)=-ULIM
160     CONTINUE
C
C       COMPUTE THE LOSS FUNCTION
C
162     DO 164 I=1,NP
164     YSUM=YSUM+YM(I,1)*YM(I,1)
C
170     CALL PRIM(2)
        IF (ITAPE.EQ.0) GO TO 500
        IF (MOD(ICOUN,IPOIN)) 500,174,500
174     DO 180 I=1,NABMAX
180     W(I)=TM(I,1)
        DO 182 I=1,NABMAX
        J=I+NABMAX
182     W(J)=TM(I,2)
        J=J+1
        W(J)=YSUM
        CALL FILDAT('WRITE',3,W,2*NABMAX+1)
        IF (ITAPE.GE.2) GO TO 184
        GO TO 500
C
C
C       FOR TEMPORARY USE. DIVIDE PARAMETERS BY TM(NA+1)-MATRIX
C
184     L=NA(1)*NP
        DET=TM(L+1,1)*TM(L+2,2)-TM(L+1,2)*TM(L+2,1)
C
```

```
      BI11=TM(L+2,2)/DET
      BI22=TM(L+1,1)/DET
      BI12=-TM(L+2,1)/DET
      BI21=-TM(L+1,2)/DET
C
      W(1)=BI11*TM(1,1)+BI12*TM(1,2)
      W(2)=BI11*TM(2,1)+BI12*TM(2,2)
      W(3)=BI21*TM(1,1)+BI22*TM(1,2)
      W(4)=BI21*TM(2,1)+BI22*TM(2,2)
C
      CALL FILDAT('WRITE',4,W,KO2DT2)
C
      IF (ICOUN-IP1) 200,200,190
190   IF (MOD(ICOUN,IP2)) 500,200,500
200   WRITE(6,1000) (W(I),I=1,4)
1000  FORMAT(' REG PAR: '4F6.2)
C
C     END IF TEMPORARY CODE
C
500   CONTINUE
C
C     ASK FOR DESIRED CONTINUATION
C
      CALL INDAM(2)
      GO TO (550,600,550,550,550,550,999,999,550),ICONT
550   IF (ITAPE.EQ.0) GO TO 560
      CALL FILES('CLOSE',3,KOD,FNAMA,'BIN',LOG)
C
      IF (ITAPE.LE.1) GO TO 560
      CALL FILES('CLOSE',4,KOD,FNAMB,'BIN',LOG)
560   GO TO (10,600,20,20,20,20,999,999,999),ICONT
C
600   CALL PRIM(3)
      GO TO 90
C
999   STOP
      END
```

```
      SUBROUTINE INDAM(IN)
C
C     AUTHOR: ULF BORISSON 1973-11-20
C
C     IN-TYPE OF OPERATION
C
C              1: INITIALIZATION
C              2: CONTINUATION
C
      COMMON YM(4,7),YSM(4,7),UM(4,10),EM(4,5),PM(16,16,4),TM(16,4),
     2 NA(4),NB(4),B0(4),KSYS(4),KREG(4),ASYS(4,4,4),BSYS(4,4,4),
     3 CSYS(4,4,4),N,NP,
     4 EVEC(4),COV(4,4),ILOG,MNOD,MACO,MAC,NABMAX,PDIAG,YSUM,
     5 RL,ULIM,NSAMP,IP1,IP2,ICONT,ICOUN,
     6 IP,IY,IU,IE,IPM,ITM,IABC,ICONS,JCONS(9,4),
     7 FNAMA,FNAMB,W(20),BUFF(32),KOD(10),ITAPE,IPOIN
C
      LV=10
C
      GO TO (10,300),IN
C
10    WRITE(LV,1010)
      IC=1
      N=RTTFF(IC)
      NP=RTTFF(IC)
C
      WRITE(LV,1020)
      IC=1
      DO 20 I=1,NP
20    KSYS(I)=RTTFF(IC)
C
      WRITE(LV,1030)
      IC=1
      DO 30 I=1,NP
30    KREG(I)=RTTFF(IC)
C
      WRITE(LV,1040)
      DO 40 K=1,N
      IC=1
      DO 40 J=1,NP
      DO 40 I=1,NP
40    ASYS(J,I,K)=RTTFF(IC)
C
      WRITE(LV,1050)
      DO 50 K=1,N
      IC=1
      DO 50 J=1,NP
      DO 50 I=1,NP
50    BSYS(J,I,K)=RTTFF(IC)
C
      WRITE(LV,1060)
      DO 60 K=1,N
      IC=1
      DO 60 J=1,NP
      DO 60 I=1,NP
60    CSYS(J,I,K)=RTTFF(IC)
C
58    WRITE(LV,1061)
      IC=1
      ITAPE=RTTFF(IC)
      IF (ITAPE.EQ.0) GO TO 61
C
      WRITE(LV,1065)
```

```
      CALL RLINE(9,8,BUFF,1)
      I=1
130   CALL RABC(I,FNAMA,ABIND)
      IF (ABIND.EQ.2 .OR.ABIND .EQ.4) GO TO 130
      IF (ITAPE.LE.1) GO TO 134
      CALL RLINE(9,8,BUFF,1)
      I=1
132   CALL RABC(I,FNAMB,ABIND)
      IF (ABIND.EQ.2 .OR. ABIND.EQ.4) GO TO 132
C
134   WRITE(LV,1066)
      IC=1
      IPOIN=RTTFF(IC)
      WRITE(LV,1067)
      IC=1
      KOD(1)=RTTFF(IC)
C
61    WRITE(LV,1070)
      IC=1
      MTYP=RTTFF(IC)
      COV(1,1)=0.
      CALL RMOVE(COV(1,1),1,COV(2,1),1,IP*IP-1)
      IF (MTYP-1) 66,62,66
62    DO 63 I=1,NP
63    COV(I,I)=1.
      GO TO 68
C
66    IC=1
      DO 67 J=1,NP
      DO 67 I=1,NP
67    COV(J,I)=RTTFF(IC)
C
68    WRITE(LV,1075)
      IC=1
      MACO=RTTFF(IC)
C
69    WRITE(LV,1080)
      IC=1
      DO 70 I=1,NP
70    NA(I)=RTTFF(IC)
      WRITE(LV,1090)
      IC=1
      DO 80 I=1,NP
80    NB(I)=RTTFF(IC)
C
      NABMAX=0
      DO 86 I=1,NP
      NAB=NA(I)+NB(I)
      IF (NAB-NABMAX) 86,86,84
84    NABMAX=NAB
86    CONTINUE
      NABMAX=NABMAX*NP
C
      WRITE(LV,1100)
      IC=1
      DO 90 I=1,NP
90    BO(I)=RTTFF(IC)
C
140   WRITE(LV,1110)
      IC=1
      PDIAG=RTTFF(IC)
C
C     INITIALIZE COVARIANCE MATRIX
C
```

```
      PM(1,1,1)=PDIAG
      PM(2,1,1)=0.
      CALL RMOVE(PM(2,1,1),1,PM(3,1,1),1,IPM-1)
      CALL RMOVE(PM(1,1,1),1,PM(2,2,1),1,IPM*IPM-IPM-1)
      CALL RMOVE(PM(1,1,1),1,PM(1,1,2),1,IPM*IPM*(IP-1))
C
C
      WRITE(LV,1112)
      IC=1
      ICONS=RTTFF(IC)
      IF (ICONS.EQ.0) GO TO 120
      WRITE(LV,1113)
      DO 110 J=1,NP
      IC=1
110   JCONS(1,J)=RTTFF(IC)
      WRITE(LV,1114)
      DO 114 J=1,NP
      IF (JCONS(1,J).EQ.0) GO TO 114
      I1=JCONS(1,J)
      DO 112 I=1,I1
      IC=1
      JCONS(I+1,J)=RTTFF(IC)
      IPOS=JCONS(I+1,J)
112   PM(IPOS,IPOS,J)=0.
114   CONTINUE
C
120   WRITE(LV,1120)
      IC=1
      ULIM=RTTFF(IC)
C
      WRITE(LV,1125)
      IC=1
      INITI=RTTFF(IC)
      IF (INITI) 96,96,92
92    DO 93 J=1,NP
      WRITE(LV,1126)
      IC=1
      DO 93 I=1,NABMAX
93    TM(I,J)=RTTFF(IC)
      GO TO 100
96    TM(1,1)=0.
      CALL RMOVE(TM(1,1),1,TM(2,1),1,ITM*IP-1)
C
100   WRITE(LV,1130)
      IC=1
      NSAMP=RTTFF(IC)
      RL=RTTFF(IC)
C
      WRITE(LV,1140)
      IC=1
      IP1=RTTFF(IC)
      IP2=RTTFF(IC)
      GO TO 999
C
300   WRITE(LV,1300)
      IC=1
      ICONT=RTTFF(IC)
      GO TO (999,100,61,69,140,58,999,999,999),ICONT
C
999   RETURN
C
1010  FORMAT(1H1,'ORDER OF SYSTEM',/
      1' NUMBER OF OUTPUTS (MUST BE THE SAME AS NUMBER OF INPUTS)')
1020  FORMAT(' NUMBER OF TIME DELAYS OF SYSTEM, KSYS(I)')
```

```
1030    FORMAT(' NUMBER OF TIME DELAYS OF REGULATOR, KREG(I)')
1040    FORMAT(' A-MATRIX POLYNOMIAL')
1050    FORMAT(' B-MATRIX POLYNOMIAL')
1060    FORMAT(' C-MATRIX POLYNOMIAL')
1061    FORMAT(' ITAPE'/
        15X,'0: NO DATA SAVED ON TAPE'/
        15X,'1: DATA SAVED ON DT1'/
        15X,'2: DATA SAVED ON DT1 AND DT2')
1065    FORMAT(' FILE NAME')
1066    FORMAT(' IPOIN  (EVERY IPOIN:TH POINT IS SAVED ON TAPE)')
1067    FORMAT(' NUMBER OF RECORDS IN THE FILE')
1070    FORMAT(' COVARIANCE MATRIX FOR THE NOISE'/
        15X,'1: UNIT MATRIX'/
        15X,'2: SPECIAL MATRIX  (TO BE TYPED IN ROWS)')
1075    FORMAT(' ODD INTEGER FOR MNODI')
1080    FORMAT(' REGULATOR:'/
        15X,'NUMBER OF ALFA PARAMETERS')
1090    FORMAT(5X,'NUMBER OF BETA PARAMETERS')
1100    FORMAT(' B0(I)')
1110    FORMAT(' PDIAG'/' (PDIAG<0: NO CONTROL)')
1112    FORMAT(' ICONS (NUMBER OF CONSTANT REGULATOR PARAMETERS)')
1113    FORMAT(' NUMBER OF CONSTANT PARAMETERS IN ROW 1, 2, ...,NP')
1114    FORMAT(' TYPE ICONS POSITIONS OF CONSTANT REGULATOR '/
        1' PARAMETERS IN T-VECTOR FOR THE ROWS 1, 2, ... ,NP')
1120    FORMAT(' LIMIT ON THE CONTROL SIGNAL'/
        15X,'<0 OR =0: NO LIMIT'/
        15X,'>0: LIMIT')
1125    FORMAT(' INITIAL VALUES OF THE REGULATOR PARAMETERS'/
        15X,'0: ZEROES'/
        15X,'1: SPECIAL')
1126    FORMAT(' ELEMENTS OF NEXT COLUMN')
1130    FORMAT(' NUMBER OF SAMPLE POINTS'/' RL')
1140    FORMAT(' IP1'/
        1' IP2')
1300    FORMAT(1H1,' CONTINUATION'/
        15X,'1: NEW INITIALIZATION'/
        15X,'2: CONTINUE THE SIMULATION'/
        15X,'3: CONTINUE FROM COVARIANCE MATRIX'/
        15X,'4: CONTINUE FROM REGULATOR'/
        15X,'5: CONTINUE FROM PDIAG'/
        15X,'6: CONTINUE FROM ITAPE'/
        15X,/
        15X,'9: STOP')
        END
```

```
      SUBROUTINE PRIM(IPRI)
C
C     IPRI - TYPE OF OPERATION
C
C              1: PRINT INITIAL VALUES
C              2: PRINT SIMULATED DATA
C
C     AUTHOR: ULF BORISSON 1973-11-20
C
      COMMON YM(4,7),YSM(4,7),UM(4,10),EM(4,5),PM(16,16,4),TM(16,4),
     2 NA(4),NB(4),B0(4),KSYS(4),KREG(4),ASYS(4,4,4),BSYS(4,4,4),
     3 CSYS(4,4,4),N,NP,
     4 EVEC(4),COV(4,4),ILOG,MNOD,MAC0,MAC,NABMAX,PDIAG,YSUM,
     5 RL,ULIM,NSAMP,IP1,IP2,ICONT,ICOUN,
     6 IP,IY,IU,IE,IPM,ITM,IABC,ICONS,JCONS(9,4),
     7 FNAMA,FNAMB,W(20),BUFF(32),KOD(10),ITAPE,IPOIN
C
      IF (MNOD-1) 6,4,6
4     WRITE(6,1005)
C
6     GO TO (10,200,300),IPRI
C
C     PRINT INITIAL VALUES
C
10    WRITE(6,1010) N,NP
      WRITE(6,1020) (KSYS(I),I=1,NP)
      WRITE(6,1030) (KREG(I),I=1,NP)
C
      WRITE(6,1040)
      DO 20 K=1,N
      WRITE(6,1050)
      DO 20 J=1,NP
20    WRITE(6,1060) (ASYS(J,I,K),I=1,NP)
C
      WRITE(6,1070)
      DO 30 K=1,N
      WRITE(6,1050)
      DO 30 J=1,NP
30    WRITE(6,1060) (BSYS(J,I,K),I=1,NP)
C
      WRITE(6,1080)
      DO 40 K=1,N
      WRITE(6,1050)
      DO 40 J=1,NP
40    WRITE(6,1060) (CSYS(J,I,K),I=1,NP)
C
      WRITE(6,1062) ITAPE
      IF (ITAPE.EQ.0) GO TO 44
      WRITE(6,1064) FNAMA,FNAMB
      WRITE(6,1066) IPOIN,KOD(1)
C
44    WRITE(6,1090) MAC0
      WRITE(6,1095)
      DO 45 J=1,NP
45    WRITE(6,1060) (COV(J,I),I=1,NP)
      WRITE(6,1100) (NA(I),I=1,NP)
      WRITE(6,1110) (NB(I),I=1,NP)
      WRITE(6,1120) (B0(I),I=1,NP)
      WRITE(6,1130) PDIAG
      WRITE(6,1132) ICONS
      IF (ICONS.EQ.0) GO TO 48
      WRITE(6,1134) (JCONS(1,J),J=1,NP)
48    WRITE(6,1140) RL
```

```
      WRITE(6,1150) ULIM
      WRITE(6,1160) NSAMP
      WRITE(6,1170)
      DO 50 J=1,NABMAX
 50   WRITE(6,1510) (TM(J,I),I=1,NP)
 C
      GO TO 999
 C
 C     PRINT SIMULATED DATA
C
 200  IF (ICOUN-IP1) 250,250,210
 210  IF (MOD(ICOUN,IP2)) 999,250,999
 250  WRITE(6,1500) ICOUN,YSUM, (YM(I,1),UM(I,1),I=1,NP)
      DO 260 I=1,NABMAX
 260  WRITE(6,1510) (TM(I,J),J=1,NP)
      GO TO 999
 C
 300  WRITE(6,1140) RL
      GO TO 999
 C
 999  RETURN
 C
 1005 FORMAT(' DECOMPOSITION OF COV IN SUBROUTINE MNODI HAS FAILED')
 1010 FORMAT(1H1,'ORDER OF SYSTEM: ',I1/
      1' NUMBER OF OUTPUTS (INPUTS): ',I2)
 1020 FORMAT(' TIME DELAYS OF SYSTEM: ',8I7)
 1030 FORMAT(' TIME DELAYS OF REGULATOR: ',8I7)
 1040 FORMAT(//' PARAMETERS OF THE SIMULATED SYSTEM'/
      1' A-MATRIX POLYNOMIAL:')
 1050 FORMAT(1X)
 1060 FORMAT(8F10.3)
 1062 FORMAT(' ITAPE: ',I1)
 1064 FORMAT(' FILE NAMES: ',A5,5X,A5)
 1066 FORMAT(' IPOIN: ',I4,5X,'KOD(1)=',I5)
 1070 FORMAT(/' B-MATRIX POLYNOMIAL')
 1080 FORMAT(/' C-MATRIX POLYNOMIAL')
 1090 FORMAT(/' ODD INTEGER FOR MCNODI: ',I4)
 1095 FORMAT(' COVARIANCE MATRIX FOR NOISE')
 1100 FORMAT(' NUMBER OF ALFA PARAMETERS: ',8I7)
 1110 FORMAT(' NUMBER OF BETA PARAMETERS: ',8I7)
 1120 FORMAT(' B0(I):',8F10.3)
 1130 FORMAT(' INITIAL VALUE OF DIAGONAL ELEMENT: ',F10,3)
 1132 FORMAT(' TOTAL NUMBER OF CONSTANT REGULATOR PARAMETERS:
      1 ',I2)
 1134 FORMAT(' NUMBER OF CONSTANT PARAMETERS IN EACH ROW  ',8I5)
 1140 FORMAT(' EXPONENTIAL WEIGHTING FACTOR:',F10.7)
 1150 FORMAT(' LIMIT ON CONTROL SIGNAL: ',F10.3)
 1160 FORMAT(' NUMBER OF SAMPLE POINTS: ',I6///)
 1170 FORMAT(' INITIAL VALUES OF REGULATOR PARAMETERS')
 C
 C
 1500 FORMAT(1X,I5,2X,F10.0,4X,5(F6.2,1X,F6.2,4X)/
      11X,6(F6.2,1X,F6.2,4X))
 1510 FORMAT(8F11.3)
      END
```

```
      SUBROUTINE SYSM
C
C     SIMULATES A MULTIVARIABLE SYSTEM
C
C     Y(I,T)+A(I,J,1)*Y(I,T-1)+...+A(I,J,N)*Y(I,T-N)=
C     =B(I,J,1)*U(I,T-K-1)+...+B(I,J,N)*U(I,T-K-N)+
C     +E(I,T)+C(I,J,1)*E(I,T-1)+...+C(I,J,N)*E(I,T-N)
C
C     YM IS ORGANIZED AS YSM IN STURM1
C
C     UM IS ORGANIZED AS IN STURM1
C
C     EM-MATRIX OF PROCESS DISTURBANCES ORGANIZED AS FOLLOWS
C        EM(I,1)=E(I,T-1), EM(I,2)=E(I,T-2), ETC.,    I=1,...,NP
C        RETURNED AS EM(I,1)=E(I,T), EM(I,2)=E(I,T-1), ETC,
C
C     AUTHOR: ULF BORISSON 1973-11-20
C
      COMMON YM(4,7),YSM(4,7),UM(4,10),EM(4,5),PM(16,16,4),TM(16,4),
     2 NA(4),NB(4),B0(4),KSYS(4),KREG(4),ASYS(4,4,4),BSYS(4,4,4),
     3 CSYS(4,4,4),N,NP,
     4 EVEC(4),COV(4,4),ILOG,MNOD,MAC0,MAC,NABMAX,PDIAG,YSUM,
     5 RL,ULIM,NSAMP,IP1,IP2,ICONT,ICOUN,
     6 IP,IY,IU,IE,IPM,ITM,IABC,ICONS,JCONS(9,4),
     7 FNAMA,FNAMB,W(20),BUFF(32),KOD(10),ITAPE,IPOIN
C
      CALL RMOVE(EM(IP,IE-1),-1,EM(IP,IE),-1,IP*(IE-1))
      CALL MNODI(EVEC,COV,IP,IP,MAC,ILOG,MNOD)
      CALL RMOVE(EVEC(1),1,EM(1,1),1,NP)
      CALL RMOVE(YM(IP,IY-1),-1,YM(IP,IY),-1,IP*(IY-1))
C
      DO 100 I=1,NP
      YM(I,1)=EM(I,1)
      DO 100 J=1,N
100   YM(I,1)=YM(I,1)-SCAPRO(ASYS(I,1,J),IABC,YM(1,J+1),1,NP)+
     1 SCAPRO(BSYS(I,1,J),IABC,UM(1,J+KSYS(I)),1,NP)
     1 +SCAPRO(CSYS(I,1,J),IABC,EM(1,J+1),1,NP)
C
      RETURN
      END
```

```
      SUBROUTINE STURM1(YSM,UM,TM,PM,RL,NP,NA,NB,K,IPDIM)
C
C     SELF TUNING REGULATOR FOR MULTIVARIABLE SYSTEMS
C
C     THIS VERSION HAS NO EXTRA TIME DELAY IN THE REGULATOR
C
C     AUTHOR: ULF BORISSON 1973-11-20
C
C     THE ALGORITHM IS BASED ON THE MODEL
C
C    Y(T)+A(I,J,1)*Y(T-K-1)+...+A(I,J,NA(I))*Y(T-K-1-NA(I))=
C    =B0(I)*(U(T-K-1)+B(I,J,1)*U(T-K-2)+...+B(I,J,NB(I))*U(T-K-1-
C    -NB(I))+E(T)
C
C     YSM-MATRIX OF SCALED PROCESS OUTPUTS ORGANIZED AS FOLLOWS
C         YSM(I,1)=Y(I,T), YSM(I,2)=Y(I,T-1),ETC., I=1,...,NP
C         RETURNED AS YSM(I,1)=YSM(I,T),YSM(I,2)=Y(I,T),
C         YSM(I,3)=Y(I,2),YSM(I,4)=Y(I,3),ETC.
C
C     UM-MATRIX OF PROCESS INPUTS ORGANIZED AS FOLLOWS
C        UM(I,1)=U(I,T-1),UM(I,2)=U(I,T-2),ETC., I=1,...,NP
C        RETURNED AS UM(I,1)=U(I,T), UM(I,2)=U(I,T-2), ETC.
C
C     TM-MATRIX OF REGULATOR PARAMETERS ORGANIZED AS FOLLOWS
C        ALFA(1,1,1)    ...    ALFA(NP,1,1)
C        ALFA(1,NP,1)   ...    ALFA(NP,NP,1)
C           .                     .
C        ALFA(1,1,NA(1)).      ALFA(NP,1,NA(NP))
C        ALFA(1,NP,NA(1)).     ALFA(NP,NP,NA(NP))
C        BETA(1,1,1)    ...    BETA(NP,1,1)
C        BETA(1,NP,1)   ...    BETA(NP,NP,1)
C           .                     .
C        BETA(1,1,NB(1)).      BETA(NP,1,NB(NP))
C        BETA(1,NP,NB(1)).     BETA(NP,NP,NB(NP))
C
      DIMENSION YSM(1,1),UM(1,1),TM(1,1),PM(1,1,1),NA(1),NB(1),K(1)
C
      COMMON/SLASK/ DUM(112),T(16),P(16,16),FI(64),DUM8(64)
C
C     ORGANIZE DATA FOR THE IDENTIFICATION ROUTINE
C
      DO 50 I=1,NP
      KI=K(I)
      NAI=NA(I)
      YIN=YSM(I,1)-UM(I,KI+1)
      J=0
      DO 10 J1=1,NAI
      DO 10 J2=1,NP
      J=J+1
      L=KI+1+J1
10    FI(J)=-YSM(J2,L)
C
      IF (NB(I))  40,40,20
20    NBI=NB(I)
      DO 30 J1=1,NBI
      DO 30 J2=1,NP
      J=J+1
      L=KI+1+J1
30    FI(J)=UM(J2,L)
C
40    NPA=(NA(I)+NB(I))*NP
      CALL RMOVE(TM(1,I),1,T(1),1,NPA)
      CALL RMOVE(PM(1,1,I),1,P(1,1),1,256)
```

```
C
      CALL RTLSID(T,P,FI,YIN,NPA,IPDIM,RL,RES,DENOM)
C           CALL RMOVE(T(1),1,TM(1,I),1,NPA)
50    CALL RMOVE(P(1,1),1,PM(1,1,I),1,256)
C
C     REORGANIZE DATA FOR NEXT STEP
C
C
C     FOR SIMPLICITY THE WHOLE ARRAYS ARE SHIFTED IN THIS VERSION
C
      CALL RMOVE(YSM(4,6),-1,YSM(4,7),-1,24)
      CALL RMOVE (UM(4,9),-1,UM(4,10),-1,36)
C
C     COMPUTE CONTROL SIGNAL
C
      DO 100 I=1,NP
      UM(I,1)=0.
      J=0
      NAI=NA(I)
      DO 70 J1=1,NAI
      DO 70 J2=1,NP
      J=J+1
70    UM(I,1)=UM(I,1)+TM(J,I)*YSM(J2,J1+1)
      IF (NB(I)) 100,100,80
80    NBI=NB(I)
      DO 90 J1=1,NBI
      DO 90 J2=1,NP
      J=J+1
90    UM(I,1)=UM(I,1)-TM(J,I)*UM(J2,J1+1)
100   CONTINUE
      RETURN
      END
```