# OPTIMAL INSTÄLLNING AV REGULATORER

PER-ARNE HEDLUND

RE-173 Mars 1976
Inst. för Reglerteknik
Lunds Tekniska Högskola

OPTIMAL INSTÄLLNING AV REGULATORER

Per-Arne Hedlund

Handledare: Torkel Glad

Innehållsförteckning

Abstract

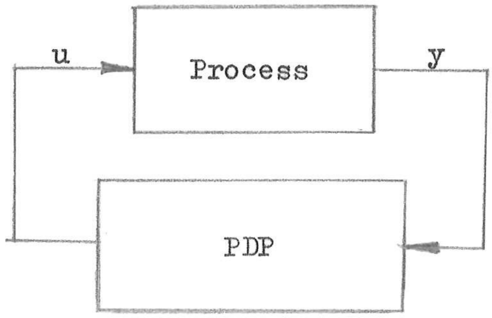
At manual tuning of a regulator you make an experiment, for exemple a stepresponse and then you vary the regulatorparameters in order to improve the performance of the system. The aim has been to make this adjustingprocedure automatic. Programdevelopments and experiments have been made at the processor PDP-15 with the realtime system RSX-15 Plus. Programming language: FORTRAN.

Sammanfattning

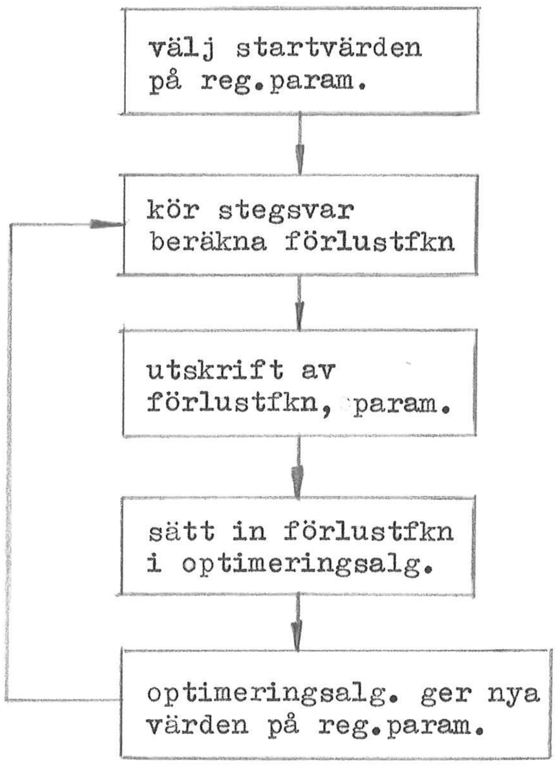
Vid manuell tuning av en regulator gör man ett experiment t.ex. stegsvar och gör sedan förändringar av regulatorparametrarna för att förbättra systemets uppförande. Uppgiften har varit att automatisera denna inställningsprocedur med hjälp av dator. Programutveckling och experiment har skett på processdatorn PDP-15 med reelltidssystemet RSX-15 Plus. Programmeringsspråk: FORTRAN.

## 1. Systembeskrivning

Optimeringen av regulatorparametrarna sköts helt automatiskt av processdatorn.




Funktionen framgår av följande flödesplan:




Optimeringsprogrammet består av tre huvudprogram med subrutiner: START, REGUL (PIDREG, BBREG), REGO (NUFLE2, PRINT (HESS)). Optimering kan ske för PI-, PD-, PID- och "Bang-Bang"-regulatorer samt för alla kombinationer av dessa.

## 1.1 Program START

Tilldelning av startvärden sker genom vektorn USER i blocket IOCOM.
Ex. MCR>SET USER 7 0.5

| | | | |
|---|---|---|---|
| USER 1 | N | | antal variabler |
| USER 2 | G1 | | |
| USER 3 | TI1 | A1 | |
| USER 4 | TD1 | K1 | |
| USER 5 | GD1 | DZ1 | regulatorparametrar |
| USER 6 | G2 | K2 | se listning av PIDREG resp. BBREG |
| USER 7 | TI2 | DZ2 | |
| USER 8 | TD2 | A2 | |
| USER 9 | GD2 | | |
| USER 10 | IREG(1) | | regulatortyper PI: 1 PD: 2 PID: 3 BB: 4 |
| USER 11 | IREG(2) | | en regulator - IREG(2): 0 |
| USER 12 | TSAMP | | samplingsperiod |
| USER 13 | NSAMP | | antal samplingar |
| USER 14 | UR | | steg |
| USER 15 | ALFA | | P-I-D: 0 BB: 1 konstanter för modifiering |
| USER 16 | FIV | | servo: 0 bom: 0,1 av förlustfkn |
| USER 19 | DFN | | se listning |
| USER 20 | HH | | av NUFLE2 |
| USER 21 | DELMAX(1) | | |
| USER 22 | DELMAX(2) | | |
| USER 23 | DELMAX(3) | | |
| USER 24 | DELMAX(4) | | max. deländring av parametervärdet |
| USER 25 | DELMAX(5) | | se NUFLE2 |
| USER 26 | DELMAX(6) | | |
| USER 27 | DELMAX(7) | | |
| USER 28 | DELMAX(8) | | |

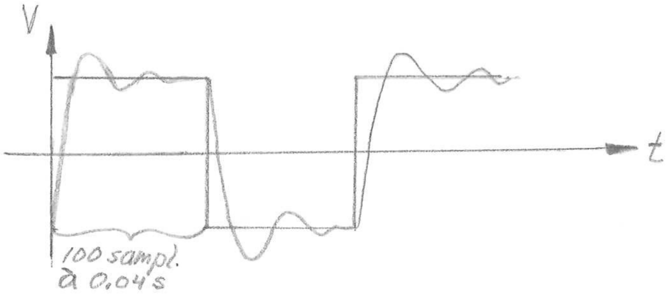
Request name sätts till REGUL vilket innebär att programmet ALIO (utför logisk och analog in/utmatning mellan datorn och datorns interface) vid varje exekvering (sampling) aktiverar programmet REGUL (logisk insignal 7 eller 16 sann).

```
TDV>TKB

OPT
>SZ,EXM
TN
>START
DP
>2ØØ
PAR
>TDV
SCP
>IOCOM(72ØØØ,4ØØ),RTCOM(75ØØØ,1ØØØ)
RC
>START,.ER#
L&S
>
START   2ØØ2Ø-21Ø22 Ø1ØØ3
STOP    21Ø23-21Ø36 ØØØ14
.FPP    21Ø37-21445 ØØ4Ø7
OTSER   21446-21716 ØØ251
SPMSG   21717-22Ø42 ØØ124
GOTO    22Ø43-22Ø7Ø ØØØ26
.DA     22Ø71-22171 ØØ1Ø1
.BP     22172-222Ø1 ØØØ1Ø
EXIF.1  222Ø2-222Ø4 ØØØØ3
.FP     222Ø5-222Ø6 ØØØØ2

CORE REQ'D
        2ØØØØ-222Ø6 Ø22Ø7
```

## 1.2 Program REGUL

Programmet REGUL sköter regleringen av processen. REGUL aktiveras varje samplingsintervall av ALIO och kallar på regulatorerna (subrutinerna PIDREG och BBREG).

En stegstörning sker och förlustfunktionen beräknas under ett




antal samplingsintervall, varefter programmet REGO aktiveras. Inkopplingen av REGO styrs från datorns interface genom logisk insignal 14. REGUL måste (liksom ALIO) fixeras i kärnminnet då programmet exekverar var 40:e ms. Regul har högre prioritet än REGO för att inte beräkningar och tidskrävande utskrifter skall påverka regleringen.

```
TDV>TKB

OPT
>SZ,EXM
TN
>REGUL
DP
>15Ø
PAR
>PIPUS
SCB
>IOCOM(72ØØØ,4ØØ),RTCOM(75ØØØ,1ØØØ)
RC
>REGUL,PIDREG,BBREG
L&S
>
REGUL    13ØØØ-13526 ØØ527
PIDREG   13527-13777 ØØ251
BBREG    14ØØØ-14157 ØØ16Ø
.BC      1416Ø-143Ø5 ØØ126
STOP     143Ø6-14321 ØØØ14
.FPP     14322-1473Ø ØØ4Ø7
SPMSG    14731-15Ø54 ØØ124
GETBIT   15Ø55-151Ø2 ØØØ26
REQF.2   151Ø3-15131 ØØØ27
.DA      15132-15232 ØØ1Ø1
FTS.3    15233-15322 ØØØ7Ø
.BP      15323-15332 ØØØ1Ø
EXIF.1   15333-15335 ØØØØ3
.FP      15336-15337 ØØØØ2

CORE REQ'D
         13ØØØ-15337 Ø234Ø
```

## 1.3 Program REGO

Programmet sköter optimeringen av regulatorparametrarna och resultatutskrifterna genom subrutinerna NUFLE2 resp. PRINT. Dessutom sker en kontroll av att de nya parametervärdena inte passerar noll, dvs byter tecken, vilket skulle få ödesdigra konsekvenser för regleringen av processen.

### 1.3.1 Subrutinen NUFLE2

Algoritmen beräknar nya värden på regulatorparametrarna utifrån de förändringar i förlustfunktionen som parametervariationerna ger. Algoritmen beräknar således minimum av en funktion med tillgång endast till funktionsvärdena.

Givet $f(x)$ där $x=(x_1, x_2, x_3)$

Gradienten beräknas

$\tilde{x}=(x_1+h, x_2, x_3)$ $\quad f(\tilde{x})$

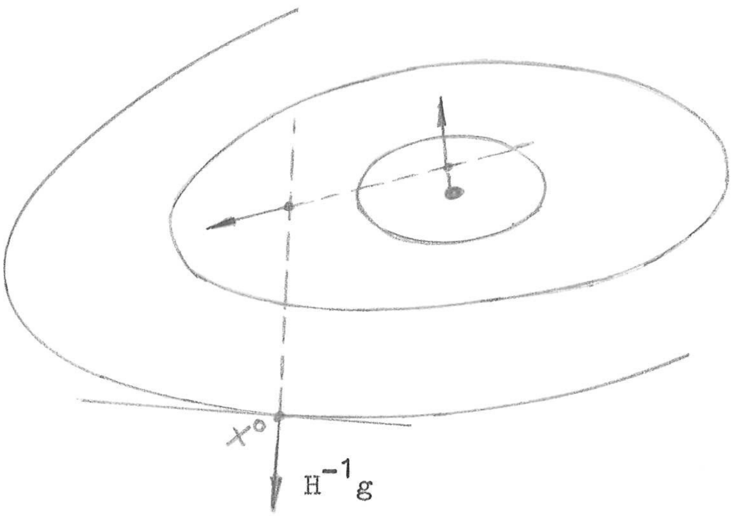
$g_1=(f(\tilde{x})-f(x))/h \approx \frac{\delta f}{\delta x_1}$ $\quad g_2= \approx \frac{\delta f}{\delta x_2}$ $\quad g_3= \approx \frac{\delta f}{\delta x_3}$

Algoritmen bildar sökriktningen

$s= -H^{-1}g$ där H är Hessianen (approx. av 2:a deriv.).

Därefter sker beräkning av min. $f(x)$ i sökriktningen.

I min.punkten bildas en ny sökriktning osv.




(se Jacoby, Kowalik, Pizzo: Iterative methods for nonlinear optimization problems sid 143 - 145)

## 1.3.2 Subrutin PRINT

Rutinen ger utskrift av process (DC-servo,Bom), regulatortyp, parametrar,gradienter och Hessianen.

Hessianen överförs från produktform till enkel matris med subrutinen HESS (se listning av HESS).

```
TDV>TKB

OPT
>SZ,EXM
TN
>REGO
DP
>200
PAR
>TDV
SCB
>RTCOM(75000,1000)
RC
>REGO,NUFLE2,PRINT,HESS
L&S
>
REGO     20020-20354 00335
NUFLE2   20355-25201 04625
PRINT    25202-27624 02423
HESS     30020-30421 00402
BCDIO    30422-34220 03577
ABS      27625-27637 00013
.BC      27640-27765 00126
STOP     34221-34234 00014
FIOPS    34235-35132 00676
INTEAE   35133-35263 00131
.FPP     35264-35672 00407
OTSER    35673-36143 00251
SPMSG    36144-36267 00124
DOTSS    36270-36406 00117
GOTO     36407-36434 00026
.DA      36435-36535 00101
.BP      27766-27775 00010
EXIF.1   36536-36540 00003
.FP      27776-27777 00002

CORE REQ'D
         20000-36540 16541
```

## 2. Exempel

Optimeringsprogrammet har körts på DC-servo och Bom.

Lämpliga värden på USER-parametrarna:

TSAMP=0.04
NSAMP=200
UR=1

DFN=-0.5
HH=0.05

### 2.1 DC-servo

Bang-Bang regulator (A=0.9)

Startvärden: K=1 DZ=0.1 Delmax(1)=0.2 Delmax(2)=0.2

$K_{opt}$=0.9595 $DZ_{opt}$=0.3801

Diagram 1 visar förlustfunktionen som fkn av antalet exekveringar av optimeringsalgoritmen.

Diagram 2 — stegsvar för startvärden

Diagram 3 — stegsvar för optimala värden

PI-regulator

Diagram 4 — stegsvar för G=1 TI=1

Diagram 5 — stegsvar för $G_{opt}$=1.1450 $TI_{opt}$=2.2998
(Då servot är integrerande kan ett stort värde på TI användas, RI≈0)

PID-regulator

Diagram 6 — stegsvar för G=1 TI=1 TD=1 GD=1

Diagram 7 — stegsvar för $G_{opt}$=1.7365 $TI_{opt}$=1.3644
$TD_{opt}$=0.0852 $GD_{opt}$=0.8875

Om startvärdena skiljer sig väsentligt från de optimala kan algoritmen finna ett lokalt minimum som ej är optimalt. Om DELMAX då är litet kan det hända att algoritmen ej kan bryta sig ur denna omgivning. Variera startvärdena eller öka DELMAX.

För PI- och PID-reg. har genomgående använts DELMAX(I)=0.4

## 2.2 Bom

PD — PD (läge — vinkel)

Lämpliga startvärden:

| | reg.1(läge) | reg.2(vinkel) |
|---|---|---|
| G | -0.5 | 1.2 |
| TI | $10^6$ | $10^6$ |
| TD | 0.9 | 0.25 |
| GD | 30 | 30 |

DELMAX bör ej vara större än 0.1 för G1 och TD2 (DELMAX(1) resp. DELMAX(5)) då större förändringar av dessa parametrar påverkar regleringen så pass kraftigt att kulan ramlar av bommen.

| | reg.1(läge) | reg.2(vinkel) |
|---|---|---|
| $G_{opt}$ | -0.8390 | 1.4388 |
| $TI_{opt}$ | $10^6$ | $10^6$ |
| $TD_{opt}$ | 1.0465 | 0.3159 |
| $GD_{opt}$ | 30.0068 | 30.0311 |

Kommentar

Vid rimligt val av startvärden och få parametrar finner algoritmen snabbt en miniminivå på förlustfunktionen (se diagram 1,diagram 8). En svag variation av de optimala parametervärdena erhålls dock pga. friktion, glapp m.m. i systemet.

Som regel bör alternativa startvärden prövas då algoritmen kan finna lokala minimipunkter som ej är optimala.

OPTIMERING AV REGULATORPARAMETRAR (se diagram 1)
*********************************

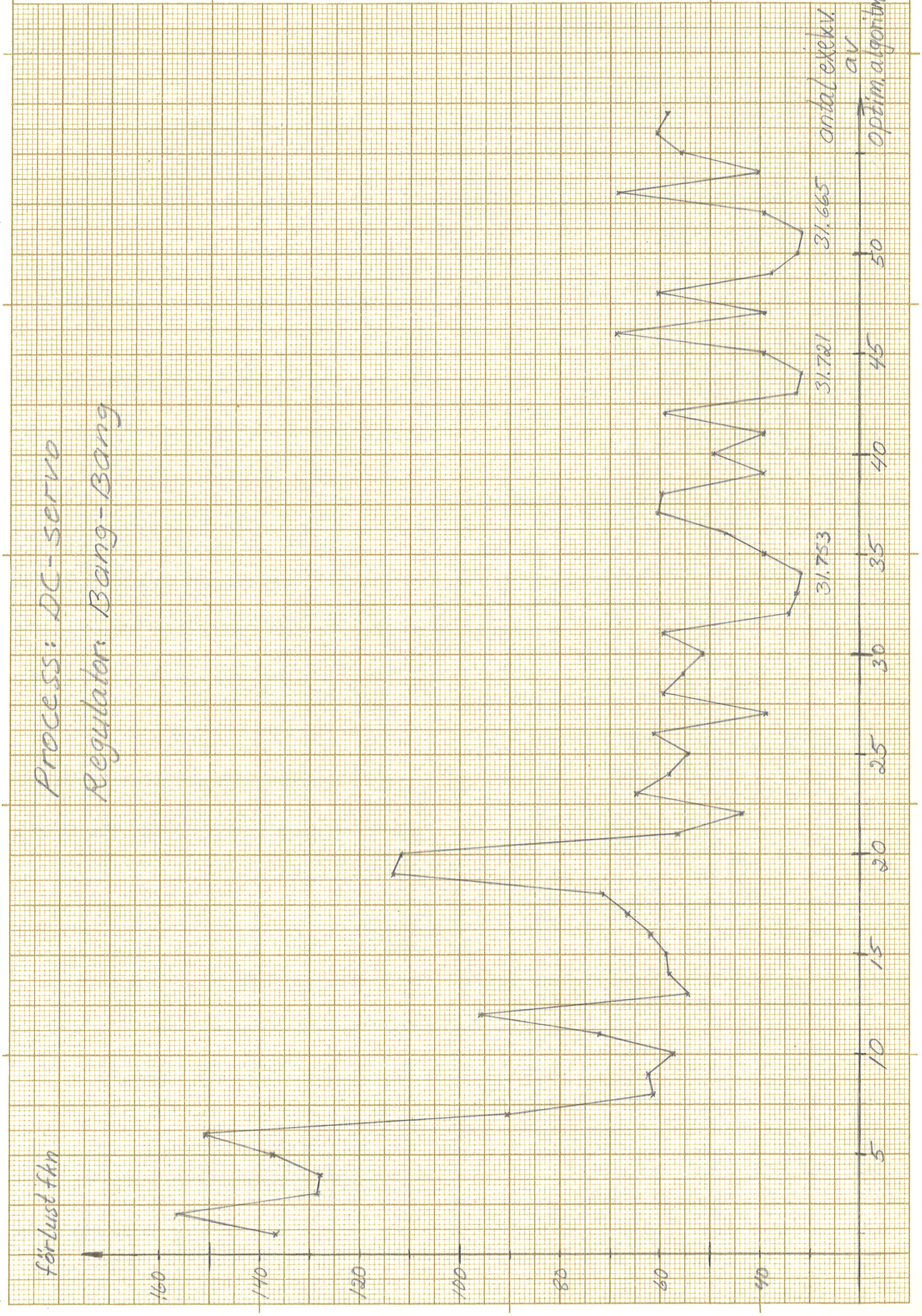
PROCESS: DC-SERVO REGULATOR: B-B (LAGE) A= 0.90

```
  ALOSS          K1           DZ1

 136.780       1.0000       0,1000
 156.707       1.0000       0,1000
 128.504       1.0500       0,1000
 127.958       0,9500       0,1000
 137.342       1.0000       0,1500
 150.714       1.0000       0,0500
GRADIENT:     44.5201    -131,6881
             HESSIAN MATRIX
               1,0000       0,0000
               0,0000       1,0000
  90,630       0,9662       0,2000
  61,626       0,9324       0,3000
  62,275       0,9824       0,3000
  57,303       0,8824       0,3000
  72,191       0,9324       0,3500
  96,047       0,9324       0,2500
GRADIENT:     59,6565    -246,0415
             HESSIAN MATRIX
               1,0000       0,0000
               0,0000       1,0000
  54,434       0,9081       0,4000
  58,151       0,8839       0,5000
  58,842       0,9581       0,4000
  61,918       0,8581       0,4000
  66,364       0,9081       0,4500
  71,288       0,9081       0,3500
GRADIENT:    -11,6545     -49,2344
             HESSIAN MATRIX
             238,4770    -655,2882
            -655,2882    1809,1862
 113,314       1,1081       0,4726
 111,745       1,0081       0,4363
GRADIENT:    -11,6545     -49,2344
             HESSIAN MATRIX
             238,4770    -655,2882
            -655,2882    1809,1862
  56,300       0,9181       0,4036
  43,697       0,9131       0,4018
  64,807       0,9631       0,4018
  58,161       0,8631       0,4018
  54,359       0,9131       0,4518
  61,319       0,9131       0,3518
GRADIENT:     59,7051     -41,2825
             HESSIAN MATRIX
           13955,3711     871,4693
             871,4693    1979,1197
  38,916       0,9074       0,4252
  59,288       0,9017       0,4486
  55,252       0,9574       0,4252
  51,575       0,8574       0,4252
  59,280       0,9074       0,4752
  34,076       0,9074       0,3752
GRADIENT:     61,7149     356,9735
             HESSIAN MATRIX
```

```
                     13981.6663      3516,8975
                      3516.8975     17892.7314
         32.735          0.9080         0,4051
34 -->   31.753          0.9087         0,3851
         39.215          0.9587         0,3851
         46.959          0.8587         0,3851
         60.377          0.9087         0,4351
         59.695          0.9087         0,3351
      GRADIENT:       -101.1626         5,5641
                         HESSIAN MATRIX
                     15355.5044      4542,8793
                      4542.8793      8895,9690
         39.590          0.9167         0,3804
         59.509          0.9127         0,3827
      GRADIENT:       -101.1626         5,5641
                         HESSIAN MATRIX
                     15355.5044      4542,8793
                      4542.8793      8895,9690
         39.299          0.9091         0,3848
     RESTART
         59.092          0,9091         0,3848
         32.694          0.9087         0,3851
44 -->   31.721          0.9587         0,3851
         39.212          0.8587         0,3851
         68.737          0.9087         0,4351
         39.170          0.9087         0,3351
      GRADIENT:        -50.0434       291.1633
                         HESSIAN MATRIX
                         1.0000         0,0000
                         0.0000         1,0000
         60.359          0.9259         0,2851
         37.712          0.9173         0,3351
      GRADIENT:        -50.0434       291.1633
                         HESSIAN MATRIX
                         1.0000         0,0000
                         0.0000         1,0000
         32.356          0.9095         0,3801
51 -->   31.665          0,9595         0,3801
         39.210          0,8595         0,3801
         68.470          0.9095         0,4301
         40.251          0.9095         0,3301
      GRADIENT:        -50.4348       278.9326
                         HESSIAN MATRIX
                         3.5309        78,8845
                        78.8845      2459,6967
         55.687          0.9208         0,3797
         60.512          0.9152         0,3799
      GRADIENT:        -50.4348       278.9326
                         HESSIAN MATRIX
                         3.5309        78,8845
                        78.8845      2459,6967
         58.778          0.9101         0,3800
     RESTART
         33.185          0.9101         0,3800
         33.241          0.9095         0,3801
         32.279          0.9595         0,3801
         39.324          0.8595         0,3801
         67.784          0.9095         0,4301
         49.237          0.9095         0,3301
      GRADIENT:        -45.4261       189.9918
                         HESSIAN MATRIX
                         1.0000         0,0000
                         0.0000         1,0000
         86.947          0.9334         0,2801
```

Förlustfunktion som fkn av antalet exekveringar av optimeringsalgoritm

Diagram 1



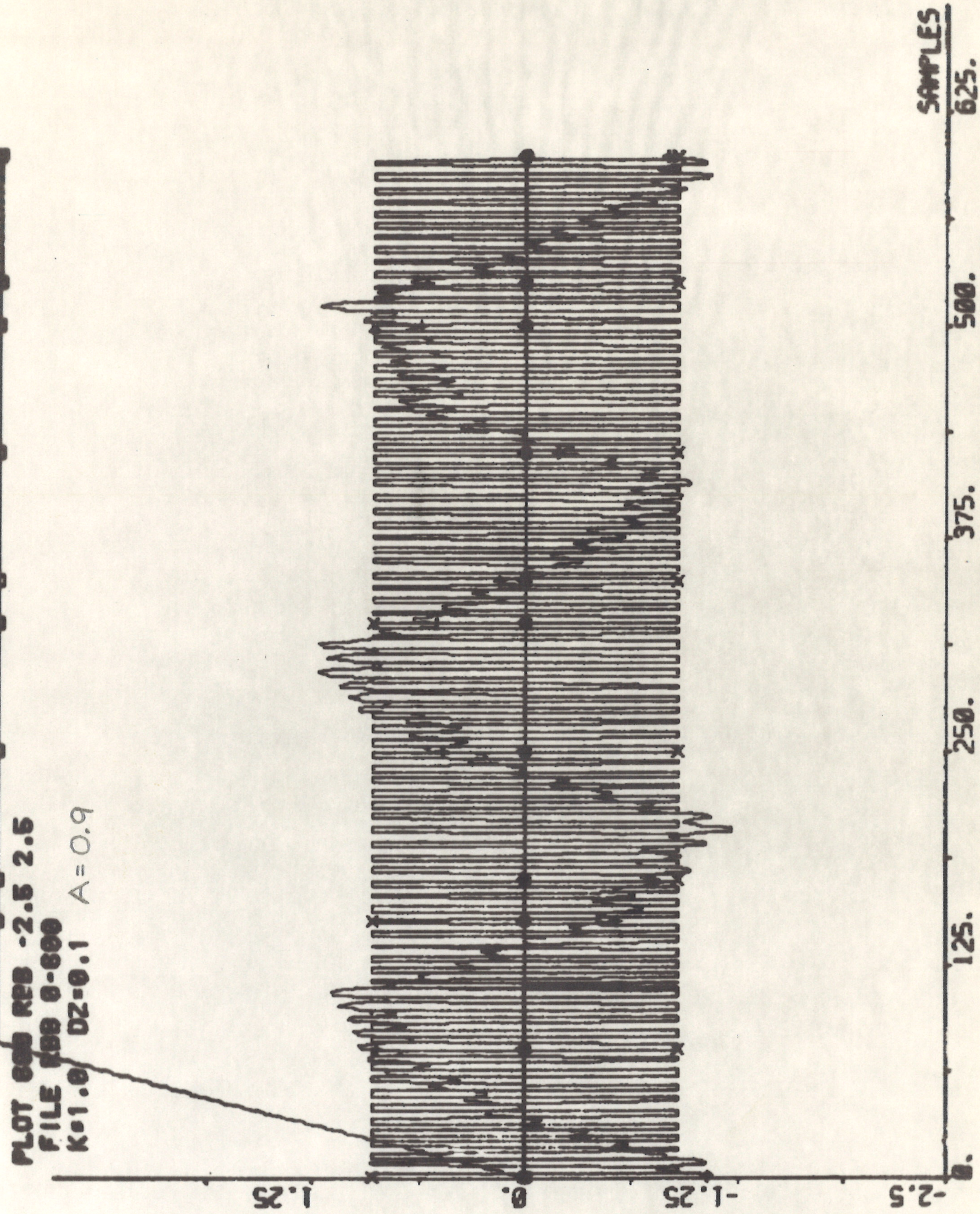



Diagram 2. Stegsvar för servo med Bang-Bang reg. (ref.)

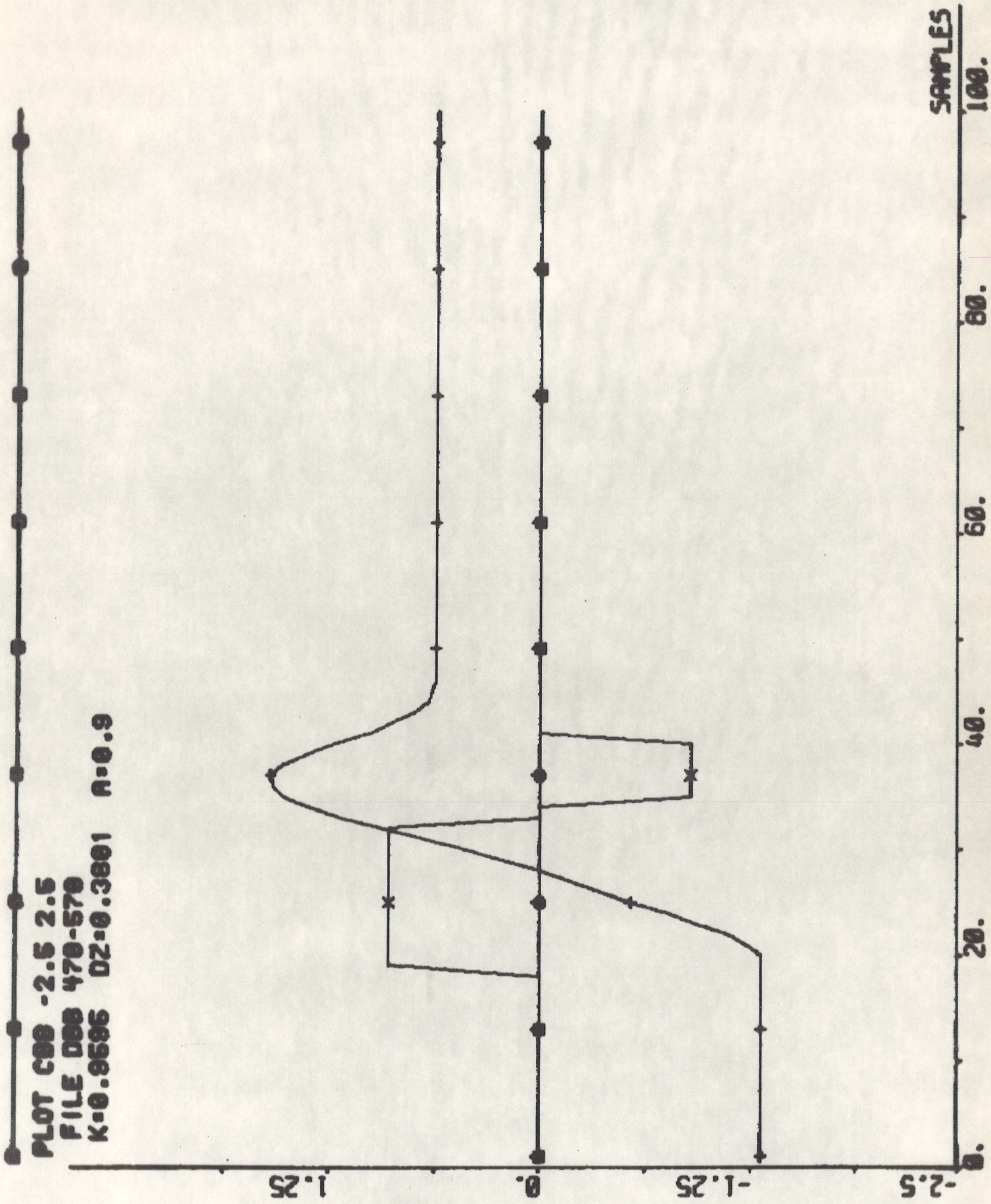



Diagram 3. Stegsvar för servo med Bang-Bang reg. (opt.)

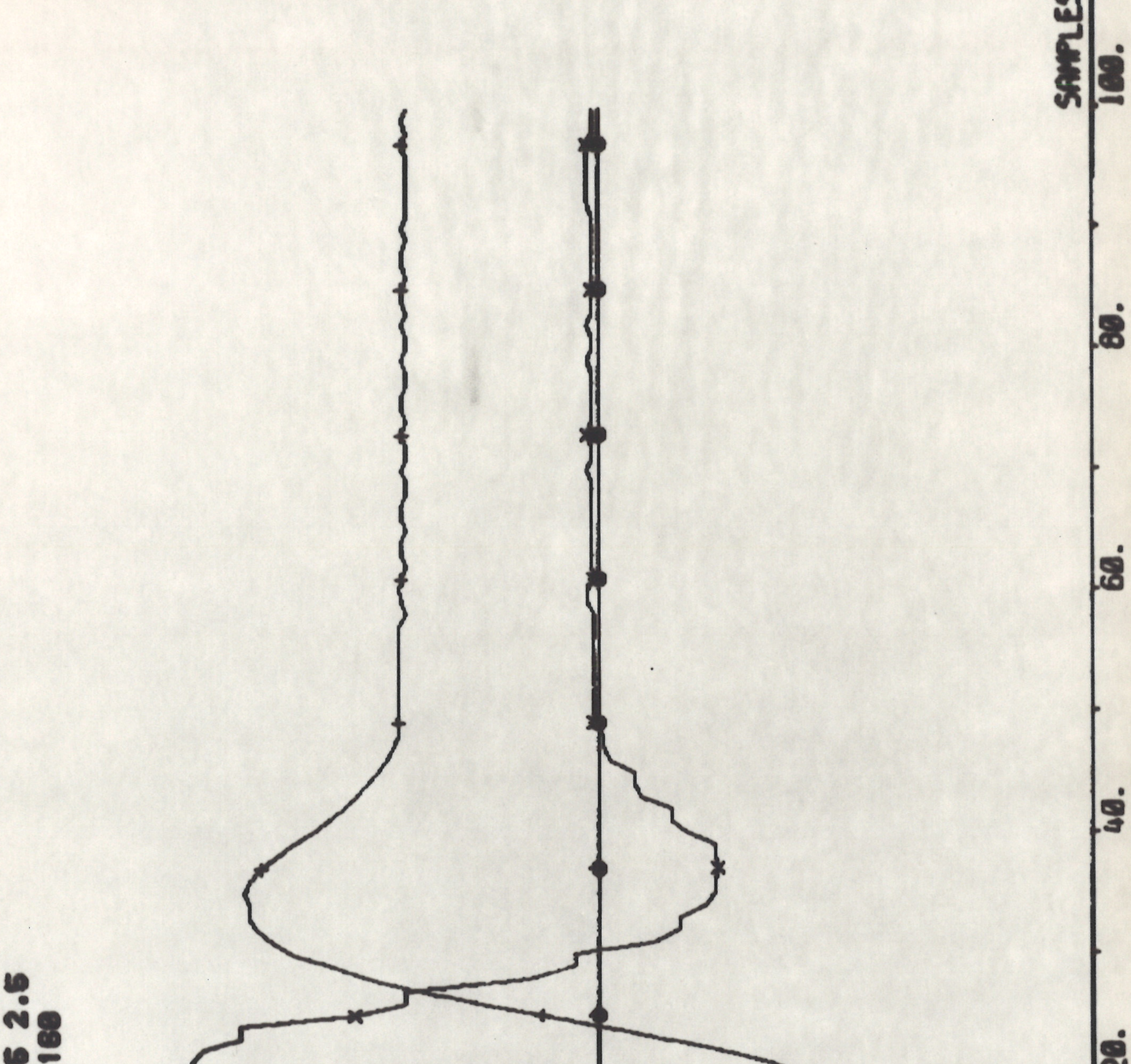



Diagram 4. Stegsvar för servo med PI-reg. (ref.)

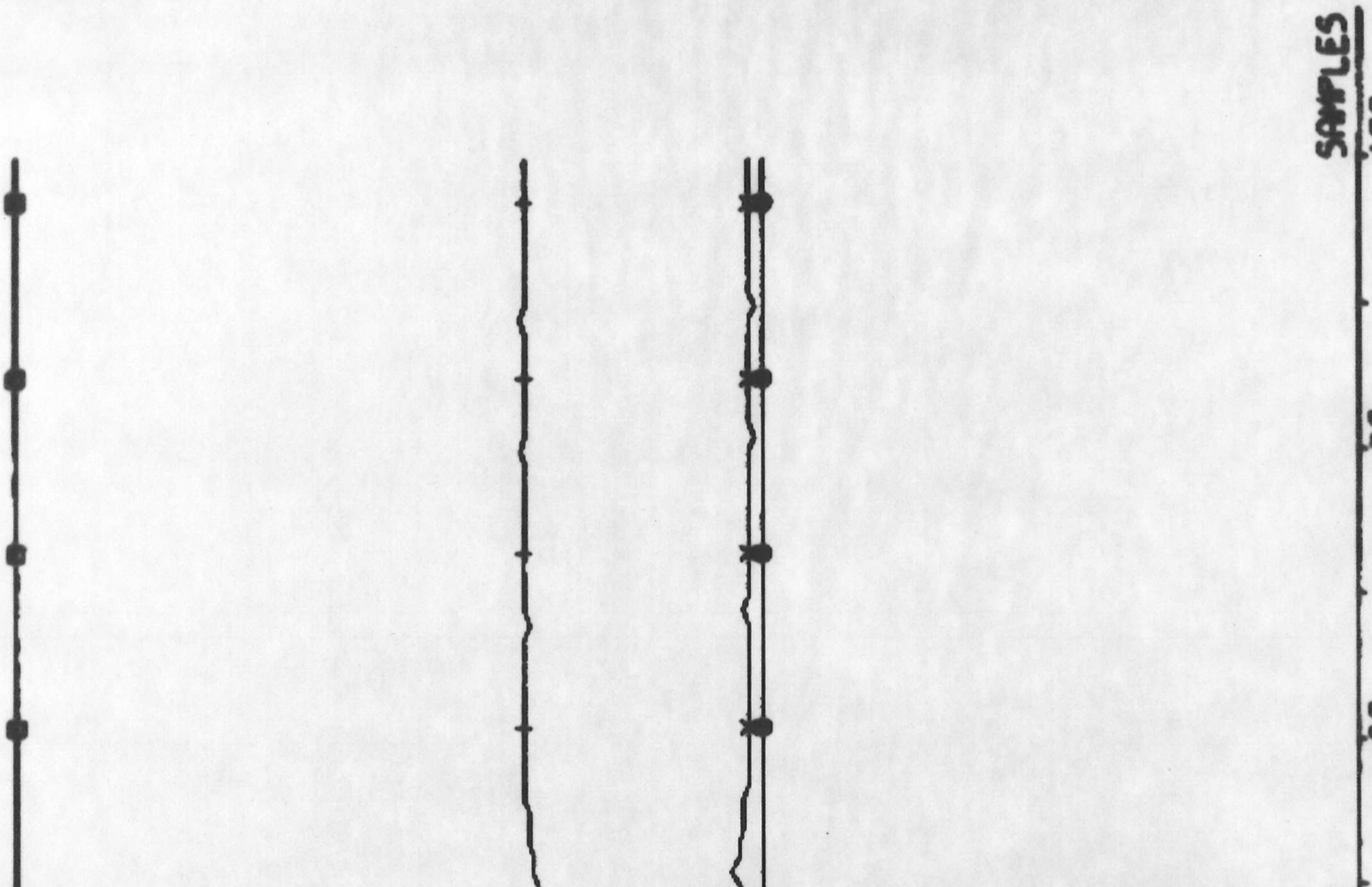



Diagram 5. Stegsvar för servo med PI-reg. (opt.)

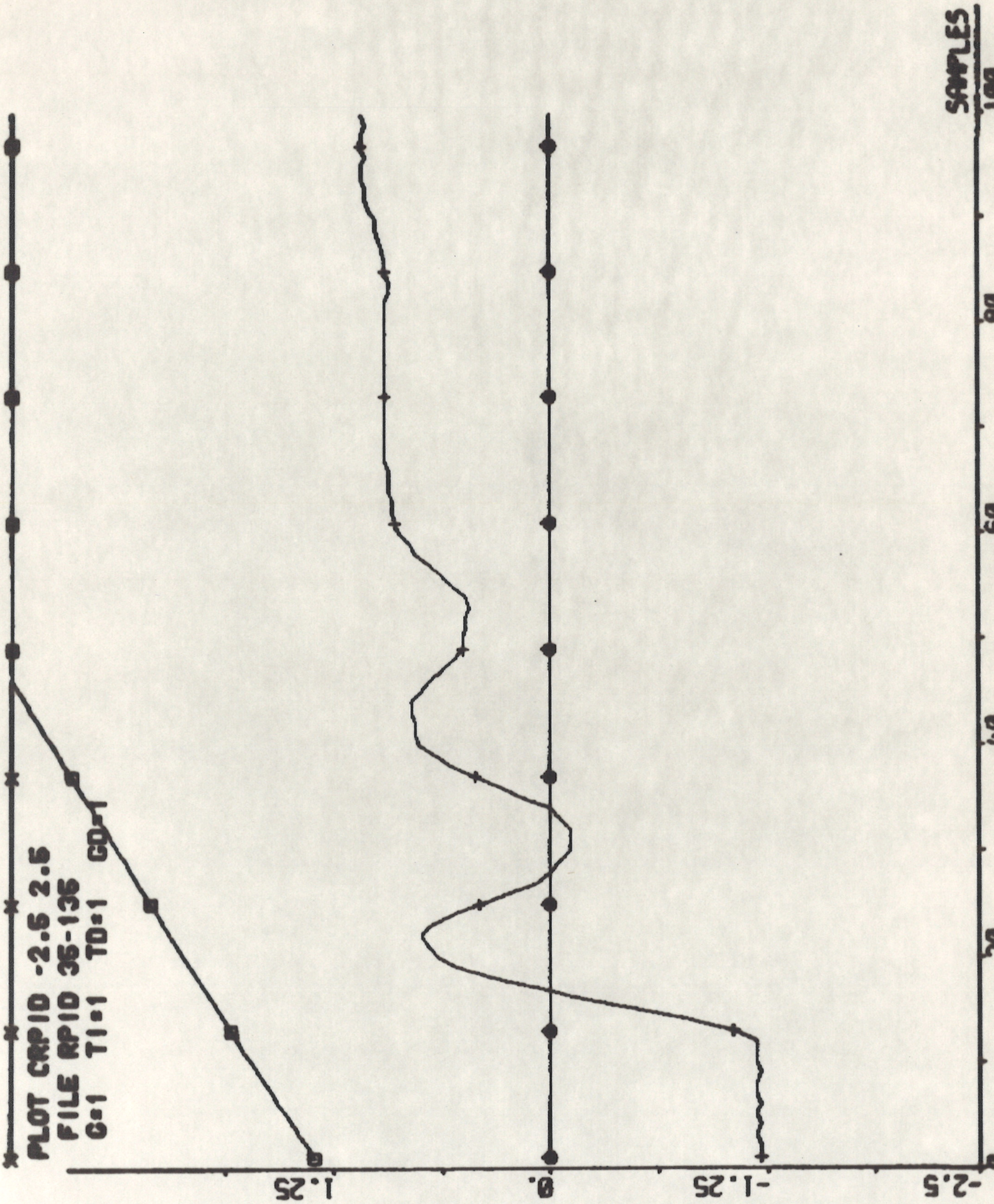



Diagram 6. Stegsvar för servo med PID-reg. (ref.)

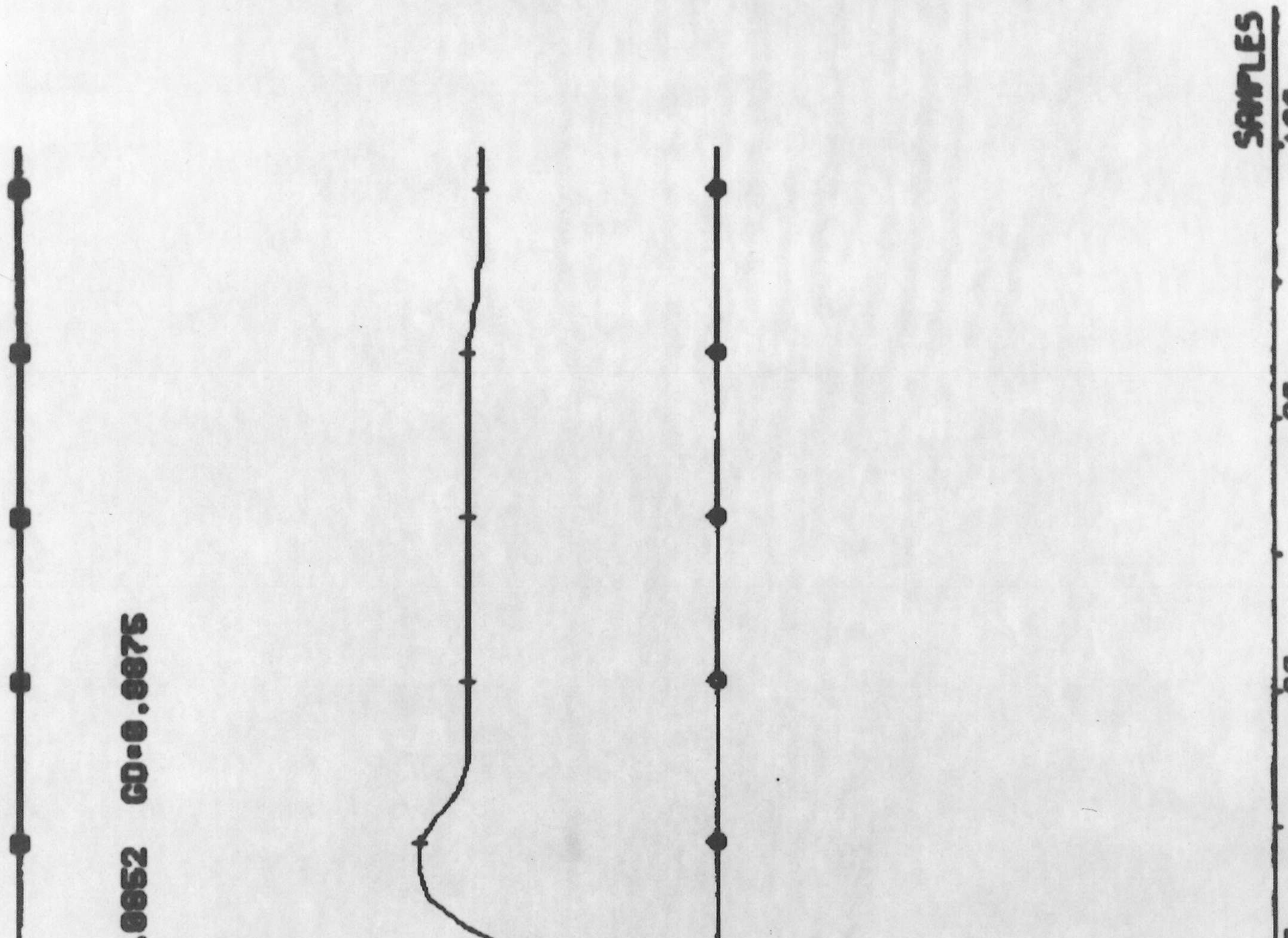
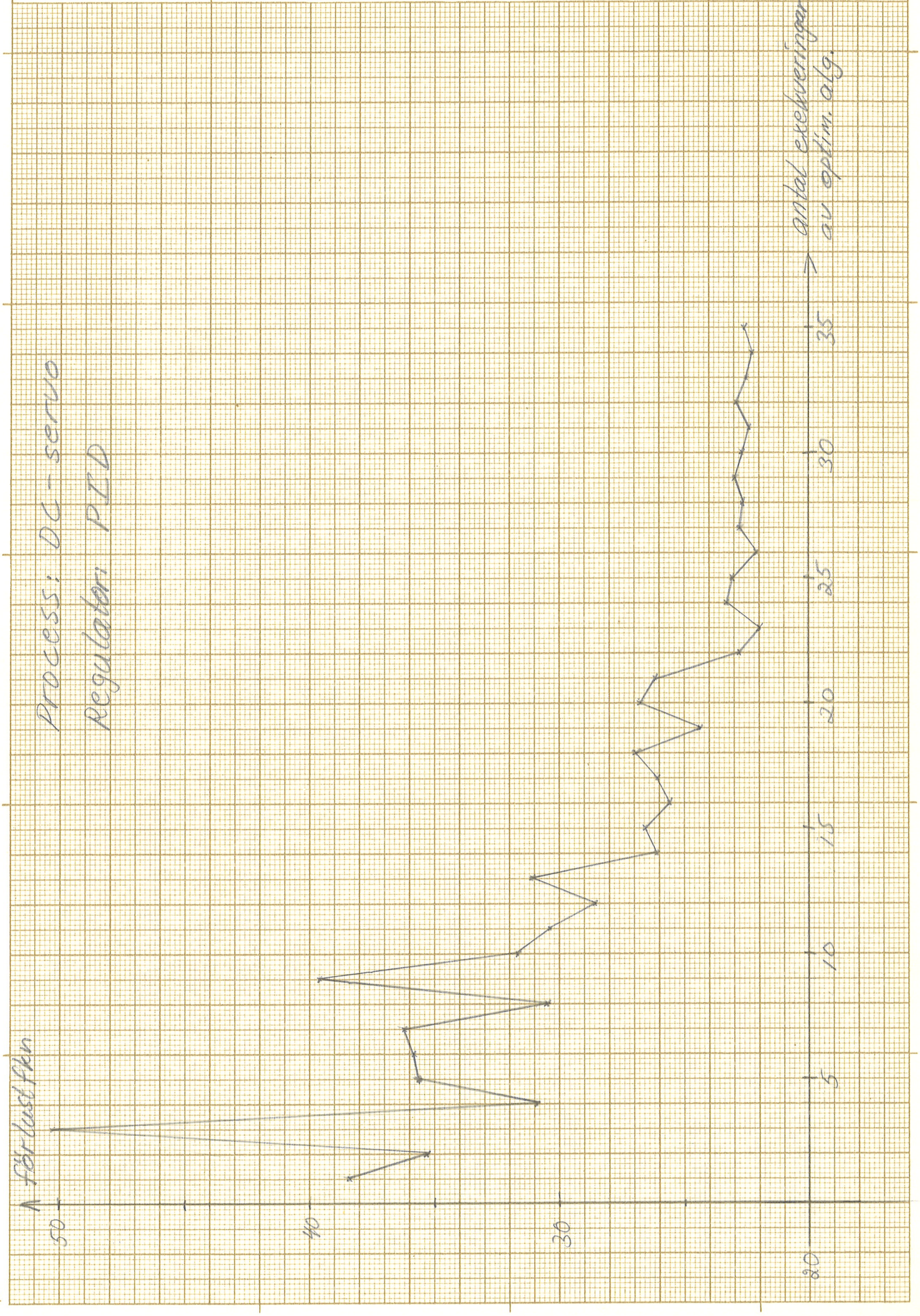



Diagram 7. Stegsvar för servo med PID-reg. (opt.)

Förlustfunktion som fkn av antalet exekveringar av optimeringsalgoritm

Diagram 8



## Appendix

Programlistningar

```
C       TASK START
C
C       ALLOTS CONSTANTS AND VARIABLES THEIR INITIAL VALUES.
C       SETS REQUEST NAME TO 'REGUL'.
C
        DIMENSION USER(36),X(8),XM(8),DELMAX(8),G(8),RP(8),IREG(2)
        COMMON /IOCOM/ICOM(256)
        COMMON /RTCOM/IBUFF(512)
        EQUIVALENCE
        1(ICOM(122),REQNM),(ICOM(185),USER(1)),
        1(IBUFF(1),IND),(IBUFF(2),N),(IBUFF(3),MODE),(IBUFF(4),IDIFF),
        2(IBUFF(46),X(1)),(IBUFF(62),G(1)),(IBUFF(78),XM(1)),
        3(IBUFF(114),ALOSS),(IBUFF(118),DFN),(IBUFF(120),HH),
        4(IBUFF(122),EPS),(IBUFF(204),DELMAX(1)),
        5(IBUFF(361),NSAMP),(IBUFF(362),MSAMP),(IBUFF(363),UR),
        6(IBUFF(365),Y01),(IBUFF(367),RI1),(IBUFF(369),D1),
        7(IBUFF(371),E01),(IBUFF(373),Y02),(IBUFF(375),RI2),
        8(IBUFF(377),D2),(IBUFF(379),E02),(IBUFF(381),IREG(1)),
        9(IBUFF(384),RP(1)),
        1(IBUFF(400),A1),(IBUFF(402),A2),(IBUFF(404),TSAMP),
        2(IBUFF(406),ALFA),(IBUFF(408),FIV),(IBUFF(410),IST)
C
        REQNM=5HREGUL
        ALOSS=0.
        Y01=0.
        RI1=0.
        D1=0.
        Y02=0.
        RI2=0.
        D2=0.
        E01=0.
        E02=0.
        IND=1
        MODE=1
        IDIFF=2
        EPS=0.001
C
        N=USER(1)
        A1=USER(3)
        A2=USER(8)
        IREG(1)=USER(10)
        IREG(2)=USER(11)
        TSAMP=USER(12)
        NSAMP=USER(13)
        MSAMP=USER(13)/2.
        UR=USER(14)
        ALFA=USER(15)
        FIV=USER(16)
        DFN=USER(19)
        HH=USER(20)
        DO 1 I=1,N
1       DELMAX(I)=USER(20+I)
C
        J=0
        M=1
        IF(IREG(1).EQ.4) M=3
        DO 2 I=M,8
        J=J+1
        G(J)=0.
        RP(J)=USER(1+I)
2       XM(J)=1.
C
        MREG=1
```

```
      IX=1
      IRP=1
3     X(IX)=RP(IRP)
      IX=IX+1
      IRP=IRP+1
      IF(IX.GT.N) GO TO 7
      GO TO (4,4,5,6,3,4,3,3),IRP
4     IF(IREG(MREG).EQ.2) IRP=IRP+1
      GO TO 3
5     IF(IREG(1).EQ.1) IRP=IRP+2
      GO TO 3
6     IF(IREG(1).EQ.4.AND.IREG(2).EQ.2) IRP=IRP+1
      MREG=MREG+1
      GO TO 3
C
7     IST=0
C
      CALL EXIT
      END
```

```
C        TASK REGUL
C
C        PERFORMS PID- AND "BANG-BANG" CONTROL.
C        ESTIMATES LOSS-FUNCTION.
C        ACTIVATES TASK REGO.
C
C        RUNS IN PARTITION PIPUS
C
C        TASK FIXED IN CORE
C
C        DEFAULT PRIORITY 150
C
C        SUBROUTINE REQUIRED
C           PIDREG
C           BBREG
C
         LOGICAL OPT
         DIMENSION USER(36),FLAI(16),FLAO(8),RP(8),RPO(8),IREG(2)
         COMMON /IOCOM/ICOM(256)
         COMMON /RTCOM/IBUFF(512)
         EQUIVALENCE
        1(ICOM(26),LIWRD),
        2(ICOM(137),FLAI(1),Y1),(ICOM(139),Y2),
        3(ICOM(169),FLAO(1),U1),(ICOM(171),U2),
        4(ICOM(185),USER(1)),
        1(IBUFF(114),ALOSS),
        2(IBUFF(361),NSAMP),(IBUFF(362),MSAMP),(IBUFF(363),UR),
        3(IBUFF(365),YO1),(IBUFF(367),RI1),(IBUFF(369),D1),
        4(IBUFF(371),EO1),(IBUFF(373),YO2),(IBUFF(375),RI2),
        5(IBUFF(377),D2),(IBUFF(379),EO2),(IBUFF(381),IREG(1)),
        6(IBUFF(384),RP(1)),(IBUFF(400),A1),
        7(IBUFF(402),A2),(IBUFF(404),TSAMP),(IBUFF(406),ALFA),
        8(IBUFF(408),FIV),(IBUFF(411),ALOSO),(IBUFF(413),RPO(1))
C
         ALOSS=ALOSS+(Y1-UR)**2+ALFA*U1**2+FIV*(Y2**2+ALFA*U2**2)
         IF(NSAMP.NE.MSAMP) GO TO 2
C
C        CHECK IF THE OPTIMIZATION ROUTINE SHOULD BE CALLED
         CALL GETBIT(LIWRD,14,OPT)
         IF(.NOT.OPT) GO TO 1
C        SAVE DATA FOR SUBROUTINE PRINT
         ALOSO=ALOSS
         DO 10 I=1,8
10       RPO(I)=RP(I)
         CALL REQST(4HREGO,200)
C
1        UR=-UR
C
2        IFI=1
         IF(IREG(1).EQ.4) GO TO 4
         CALL PIDREG(RP,IFI,TSAMP,UR,Y1,U1,YO1,RI1,D1)
         IF(IREG(2).EQ.0) GO TO 6
         IFI=5
         IF(IREG(2).EQ.4) GO TO 5
3        CALL PIDREG(RP,IFI,TSAMP,U1,Y2,U2,YO2,RI2,D2)
         GO TO 6
4        CALL BBREG(RP,IFI,TSAMP,UR,Y1,U1,EO1,A1)
         IF(IREG(2).EQ.0) GO TO 6
         IFI=3
         IF(IREG(2).NE.4) GO TO 3
5        CALL BBREG(RP,IFI,TSAMP,U1,Y2,U2,EO2,A2)
6        NSAMP=NSAMP-1
         IF(NSAMP.LE.0) GO TO 7
```

```
C
      CALL EXIT
C
7     ALOSS=0.
      NSAMP=USER(13)
      UR=USER(14)
C
      CALL EXIT
      END
```

```
C        PID-REGULATOR
C
         SUBROUTINE PIDREG(RP,IFI,TSAMP,UR,Y,U,YO,RI,D)
C
C        INTEGRATING TERM RI,DERIVATIVE TERM D AND OLD PROCESS OUTPUT
C        ARE UPDATED.
C
C        RP         A REAL ARRAY IN WHICH THE REGULATOR-PARAMETERS ARE
C                   STORED.
C        IFI        INDICATES THE POSITION OF THE FIRST PARAMETER (G) IN RP.
C        G          GAIN
C        TI         INTEGRATING TIME CONSTANT
C        TD         DERIVATING TIME CONSTANT
C        GD         DERIVATING FILTER CONSTANT
C        TSAMP      SAMPLING PERIOD
C        UR         INPUT REFERENCE VALUE
C        Y          OUTPUT SIGNAL FROM PROCESS
C        U          INPUT SIGNAL TO PROCESS
C        YO         OLD VALUE OF PROCESS OUTPUT
C        RI         INTEGRATING TERM (INTERNAL VARIABLE)
C        D          DERIVATIVE TERM (INTERNAL VARIABLE)
C
         DIMENSION RP(1)
C
         G=RP(IFI)
         TI=RP(IFI+1)
         TD=RP(IFI+2)
         GD=RP(IFI+3)
C
         E=UR-Y
         ALFA=G*TD/TSAMP
         BETA=TD/(TD+GD*TSAMP)
C
         P=G*E
         RI=RI+G*TSAMP/TI*E
         D=BETA*D+ALFA*(1.-BETA)*(YO-Y)
         U=P+RI+D
C
         YO=Y
C
         RETURN
         END
```

```
C       "BANG-BANG"-REGULATOR
C
        SUBROUTINE BBREG(RP,IFI,TSAMP,UR,Y,U,EO,A)
C
C       OLD ERROR IS UPDATED.
C
C       RP       A REAL ARRAY IN WHICH THE REGULATOR-PARAMETERS ARE
C                STORED.
C       IFI      INDICATES THE POSITION OF THE FIRST PARAMETER (K) IN RP.
C       K        SWITCHING LINE CONSTANT
C       DZ       DEATH ZONE CONSTANT
C       TSAMP    SAMPLING PERIOD
C       UR       INPUT REFERENCE VALUE
C       Y        OUTPUT SIGNAL FROM PROCESS
C       U        INPUT SIGNAL TO PROCESS
C       EO       OLD VALUE OF ERROR (INTERNAL VARIABLE)
C       A        "BANG-BANG" CONSTANT (+A OR -A)
C
        DIMENSION RP(1)
C
        K=RP(IFI)
        DZ=RP(IFI+1)
C
        E=Y-UR
        EP=(E-EO)/TSAMP
        ALFA=E+K*EP
C
        U=0.
        IF(ALFA.GT.DZ) U=-A
        IF(ALFA.LT.-DZ) U=A
C
        EO=E
C
        RETURN
        END
```

```
C       TASK REGO
C
C       PERFORMS OPTIMIZATION OF REGULATOR-PARAMETERS
C
C       RUNS IN PARTITION TDV
C
C       DEFAULT PRIORITY 200
C
C       SUBROUTINE REQUIRED
C          NUFLE2
C          PRINT
C
        DIMENSION P(8),RP(8),IREG(2)
        COMMON /RTCOM/IBUFF(512)
        EQUIVALENCE
        1(IBUFF(2),N),(IBUFF(30),P(1)),
        2(IBUFF(381),IREG(1)),(IBUFF(384),RP(1))
C
C       CALL THE OPTIMIZATION ROUTINE
        CALL NUFLE2
C
C       PRINT DATA ON LINE-PRINTER
        CALL PRINT
C
C       PREVENT PARAMETERS FROM PASSING ZERO
        MREG=1
        IP=1
        IRP=1
1       IF(P(IP)/RP(IRP).GT.0.) GO TO 2
        P(IP)=0.1
        IF(RP(IRP).LT.0.) P(IP)=-0.1
2       RP(IRP)=P(IP)
        IP=IP+1
        IRP=IRP+1
        IF(IP.GT.N) GO TO 6
        GO TO (3,3,4,5,1,3,1,1),IRP
3       IF(IREG(MREG).EQ.2) IRP=IRP+1
        GO TO 1
4       IF(IREG(1).EQ.1) IRP=IRP+2
        GO TO 1
5       IF(IREG(1).EQ.4.AND.IREG(2).EQ.2) IRP=IRP+1
        MREG=MREG+1
        GO TO 1
6       CONTINUE
C
        CALL EXIT
        END
```

```
      SUBROUTINE NUFLE2
C
C        ADAPTION OF NUFLET FOR RSX
C
C     ROUTINE FOR FINDING THE MINIMUM OF A FUNCTION F(X), WHEN
C     ONLY FUNCTION VALUES ARE AVAILABLE
C     AUTHOR, T. GLAD 1973-07-01.
C     REVISED, C. KALLSTROM 1973-08-10.
C     REVISED (ADAPTED FOR SIMNON), T. GLAD 1974-07-06
C     REVISED (ADAPTED FOR RSX), T. GLAD 1974-11-16
C     REFERENCE:   R. FLETCHER,  FORTRAN SUBROUTINES FOR MINIMIZATION
C     BY QUASI-NEWTON METHODS,  REPORT AERE-R7125, HARWELL
C
C     THE FIELD IBUFF IS DIVIDED ACCORDING TO THE EQUIVALENCE
C     STATEMENT BELOW
C
C     ALOSS  VALUE OF LOSS FUNCTION
C     P      NEW PARAMETER VALUE
C     IND    INDICATES TO WHICH POINT IN NUFLE1 A RETURN SHOULD
C            BE MADE
C     X      A REAL ARRAY OF N ELEMENTS IN WHICH THE CURRENT ESTIMATE
C            OF THE SOLUTION IS STORED. AN INITIAL APPROXIMATION MUST
C            BE SET IN X ON ENTRY TO NUFLET AND THE BEST ESTIMATE
C            OBTAINED WILL BE RETURNED ON EXIT
C     N      THE NUMBER OF VARIABLES (MIN 2, NO MAX)
C     F      A REAL NUMBER IN WHICH THE BEST VALUE OF F(X) CORRESPONDING
C            TO X ABOVE WILL BE RETURNED
C     G      A REAL ARRAY OF N ELEMENTS USED TO STORE AN ESTIMATE OF
C            THE GRADIENT VECTOR. NOT TO BE SET ON ENTRY
C     H      A REAL ARRAY OF N*(N+1)/2 ELEMENTS IN WHICH AN ESTIMATE OF
C            HESSIAN MATRIX IS STORED. THE MATRIX IS REPRESENTED IN
C            THE PRODUCT FORM LD(L)T WHERE L IS A LOWER TRIANGULAR
C            MATRIX WITH UNIT DIAGONALS AND D IS A DIAGONAL MATRIX.
C            THE LOWER TRIANGLE OF L IS STORED BY COLUMNS IN H EXCEPTING
C            THAT THE UNIT DIAGONAL ELEMENTS ARE REPLACED BY THE
C            CORRESPONDING ELEMENTS OF D. THE SETTING OF H ON ENTRY IS
C            CONTROLLED BY THE PARAMETER MODE.
C     W      A REAL ARRAY OF 4*N ELEMENTS USED AS WORKING SPACE
C     DFN    A REAL NUMBER WHICH MUST BE SET SO AS TO GIVE NUFLET AN
C            ESTIMATE OF THE LIKELY REDUCTION TO BE OBTAINED IN F(X).
C            DFN IS USED ONLY ON THE FIRST ITERATION SO AN ORDER OF
C            MAGNITUDE ESTIMATE WILL SUFFICE.
C             DFN>0 THE SETTING OF DFN ITSELF WILL BE TAKEN AS THE
C             LIKELY REDUCTION TO BE OBTAINED IN F(X)
C             DFN=0 IT WILL BE ASSUMED THAT AN ESTIMATE OF THE MINIMUM
C             VALUE OF F(X) HAS BEEN SET IN ARGUMENT F, AND THE LIKELY
C             REDUCTION IN F(X) WILL BE COMPUTED ACCORDING TO THE INITIA
C             FUNCTION VALUE
C             DFN<0 A MULTIPLE ABS(DFN) OF THE MODULUS OF THE INITIAL
C             FUNCTION VALUE WILL BE TAKEN AS AN ESTIMATE OF THE LIKELY
C             REDUCTION.
C     XM     A REAL ARRAY OF N ELEMENTS USED IN SCALING,SEE BELOW
C     HH     A REAL NUMBER. THE STEP LENGTH USED WHEN CALCULATING G(I)
C            BY DIFFERENCES IS HH*XM(I). HINT GIVEN BY FLETCHER: SET HH
C            EQUAL TO 2**(-T/2) WHERE T IS THE NUMBER OF SIGNIFICANT
C            BINARY DIGITS IN THE CALCULATION OF F.
C     EPS    A REAL NUMBER. THE ACCURACY REQUIRED IN X(I) IS EPS*XM(I).
C     MODE   AN INTEGER WHICH CONTROLS THE SETTING OF THE INITIAL
C            ESTIMATE OF THE HESSIAN MATRIX IN THE PARAMETER H.
C             MODE=1 AN ESTIMATE CORRESPONDING TO THE UNIT MATRIX IS SET
C             IN H BY NUFLET
C             MODE=3 NUFLET ASSUMES THAT THE HESSIAN MATRIX HAS BEEN SET
C             IN H IN PROCUCT FORM.
```

```
C     IDIFF  INDICATES IF FORWARD DIFFERENCES (IDIFF=1) OR CENTRAL
C            DIFFERENCES (IDIFF=2) SHALL BE USED IN NUMERIACAL
C            DIFFERENTIATION
C     IEXIT  AN INTEGER GIVING THE REASON FOR EXIT FROM NUFLET.THIS WILL
C            BE SET BY NUFLET AS FOLLOWS
C             IEXIT=0 THE NORMAL EXIT IN WHICH ABS(DX(I))<EPS*XM(I) FOR
C             I=1,2,..N WHERE DX(I) IS THE CHANGE IN X ON AN ITERATION
C             IEXIT=1 (MODE=2 ONLY) HESSIAN MATRIX IS NOT POSITIVE DEFIN
C             IEXIT=2 GT*DX .GE. 0 EITHER DUE TO ROUNDING ERRORS
C             BECAUSE EPS IS SET TOO SMALL FOR THE COMPUTER WORD LENGTH
C             OR THE TRUNCATION ERROR IN THE FINITE DIFFERENCE FORMULA
C             FOR G BEING DOMINANT.
C             IEXIT=3 FUNCT CALLED MAXFN TIMES.
C             IEXIT=4 MINIMIZING TERMINATED FROM FUNCT.
C
C     SUBROUTINE REQUIRED
C            (FUNCT)
C
      DIMENSION X(8),G(8),H(36),W(32),XM(8),P(8),XACT(8),
     1          AMAX(8),AMIN(8),DELMAX(8)
      COMMON/RTCOM/IBUFF(512)
C
      EQUIVALENCE (IBUFF(1),IND),(IBUFF(2),N),(IBUFF(3),MODE),
     1(IBUFF(4),IDIFF),(IBUFF(5),IEXIT),(IBUFF(6),NP),
     2(IBUFF(7),N1),(IBUFF(8),NN),(IBUFF(9),IS),(IBUFF(10),IU),
     3(IBUFF(11),IV),(IBUFF(12),IB),(IBUFF(13),ITN),
     4(IBUFF(14),INT),(IBUFF(15),I),(IBUFF(16),LINK),
     5(IBUFF(30),P(1)),(IBUFF(46),X(1)),(IBUFF(62),G(1)),
     6(IBUFF(78),XM(1)),
     7(IBUFF(114),ALOSS),(IBUFF(116),F),(IBUFF(118),DFN),
     8(IBUFF(120),HH),(IBUFF(122),EPS),(IBUFF(124),DMIN),
     9(IBUFF(126),Z),(IBUFF(128),DF),(IBUFF(130),GS0),
     1(IBUFF(132),AEPS),(IBUFF(134),ALPHA),(IBUFF(136),FF),
     2(IBUFF(138),TOT),(IBUFF(140),F1),(IBUFF(142),F2),
     3(IBUFF(144),GYS),(IBUFF(146),DGS),(IBUFF(148),SIG),
     4(IBUFF(150),ZZ),(IBUFF(152),XACT(1)),(IBUFF(168),ZACT),
     5(IBUFF(170),ZZACT),(IBUFF(172),AMAX(1)),(IBUFF(188),AMIN(1)),
     6(IBUFF(204),DELMAX(1)),(IBUFF(220),H(1)),(IBUFF(292),W(1))
C
      GO TO (1,205,212,222,232,242,252,262),IND
    1 CONTINUE
      NP=N+1
      N1=N-1
      NN=N*NP/2
      IS=N
      IU=N
      IV=N+N
      IB=IV+N
      IEXIT=1
      IF(MODE.EQ.3) GO TO 15
      IJ=NN+1
      DO 5 I=1,N
      DO 6 J=1,I
      IJ=IJ-1
    6 H(IJ)=0.
    5 H(IJ)=1.
   15 CONTINUE
      IJ=NP
      DMIN=H(1)
      DO 16 I=2,N
      IF(H(IJ).GE.DMIN) GO TO 16
      DMIN=H(IJ)
   16 IJ=IJ+NP-I
      IF(DMIN.LE.0.) RETURN
```

```
      Z=F
      ITN=0
      IND=2
      DO 202 I=1,N
  202 P(I)=X(I)
      RETURN
  205 F=ALOSS
      DO 207 I=1,N
  207 XACT(I)=P(I)
      DF=DFN
      IF(DFN.EQ.0.) DF=F-Z
      IF(DFN.LT.0.) DF=ABS(DF*F)
      IF(DF.LE.0.) DF=1.
   17 CONTINUE
      DO 19 I=1,N
   19 W(I)=X(I)
      LINK=1
      IF(IDIFF-1) 100,100,110
   18 CONTINUE
   20 CONTINUE
      ITN=ITN+1
      W(1)=-G(1)
      DO 22 I=2,N
      IJ=I
      I1=I-1
      Z=-G(I)
      DO 23 J=1,I1
      Z=Z-H(IJ)*W(J)
   23 IJ=IJ+N-J
   22 W(I)=Z
      W(IS+N)=W(N)/H(NN)
      IJ=NN
      DO 25 I=1,N1
      IJ=IJ-1
      Z=0.
      DO 26 J=1,I
      Z=Z+H(IJ)*W(IS+NP-J)
   26 IJ=IJ-1
   25 W(IS+N-I)=W(N-I)/H(IJ)-Z
      Z=0.
      GS0=0.
      DO 29 I=1,N
      IF(Z*XM(I).GE.ABS(W(IS+I))) GO TO 29
      Z=ABS(W(IS+I))/XM(I)
   29 GS0=GS0+G(I)*W(IS+I)
      AEPS=EPS/Z
      IEXIT=2
      IF(GS0.GE.0.) GO TO 92
      ALPHA=-2.*DF/GS0
      IF(ALPHA.GT.1.) ALPHA=1.
      FF=F
      TOT=0.
      INT=0
      IEXIT=0
   30 CONTINUE
      DO 329 I=1,N
      IF(ALPHA*ABS(W(IS+I)).GT.DELMAX(I)) ALPHA=
     1  DELMAX(I)/ABS(W(IS+I))
  329 CONTINUE
      DO 31 I=1,N
   31 W(I)=X(I)+ALPHA*W(IS+I)
      IND=3
      DO 210 I=1,N
  210 P(I)=W(I)
```

```
      RETURN
  212 F1=ALOSS
      IF(F1.GE.F) GO TO 40
      F2=F
      TOT=TOT+ALPHA
   32 CONTINUE
      DO 33 I=1,N
      XACT(I)=P(I)
   33 X(I)=W(I)
      F=F1
      IF(INT-1) 35,49,50
   35 CONTINUE
      DO 334 I=1,N
      IF(ALPHA*ABS(W(IS+I)).GT.DELMAX(I)) ALPHA=
     1 DELMAX(I)/ABS(W(IS+I))
  334 CONTINUE
      DO 34 I=1,N
   34 W(I)=X(I)+ALPHA*W(IS+I)
      IND=4
      DO 220 I=1,N
  220 P(I)=W(I)
      RETURN
  222 F1=ALOSS
      IF(F1.GE.F) GO TO 50
      IF(F1+F2.GE.F+F.AND.7.*F1+5.*F2.GT.12.*F) INT=2
      TOT=TOT+ALPHA
      ALPHA=2.*ALPHA
      GO TO 32
   40 CONTINUE
      IF(ALPHA.LT.AEPS) GO TO 92
      ALPHA=.5*ALPHA
      DO 41 I=1,N
   41 W(I)=X(I)+ALPHA*W(IS+I)
      IND=5
      DO 230 I=1,N
  230 P(I)=W(I)
      RETURN
  232 F2=ALOSS
      IF(F2.GE.F) GO TO 45
      TOT=TOT+ALPHA
      F=F2
      DO 42 I=1,N
      XACT(I)=P(I)
   42 X(I)=W(I)
      GO TO 49
   45 CONTINUE
      Z=.1
      IF(F1+F.GT.F2+F2) Z=1.+.5*(F-F1)/(F+F1-F2-F2)
      IF(Z.LT..1) Z=.1
      ALPHA=Z*ALPHA
      INT=1
      GO TO 30
   49 CONTINUE
      IF(TOT.LT.AEPS) GO TO 92
   50 CONTINUE
      ALPHA=TOT
      DO 56 I=1,N
      W(I)=X(I)
   56 W(IB+I)=G(I)
      LINK=2
      IF(IDIFF-1) 100,100,110
   54 CONTINUE
      GYS=0.
      DO 55 I=1,N
```

```
      W(I)=W(IB+I)
   55 GYS=GYS+G(I)*W(IS+I)
      DF=FF-F
      DGS=GYS-GS0
      IF(DGS.LE.0.) GO TO 20
      LINK=1
      IF(DGS+ALPHA*GS0.GT.0.) GO TO 52
      DO 51 I=1,N
   51 W(IU+I)=G(I)-W(I)
      SIG=1./(ALPHA*DGS)
      GO TO 70
   52 CONTINUE
      ZZ=ALPHA/(DGS-ALPHA*GS0)
      Z=DGS*ZZ-1.
      DO 53 I=1,N
   53 W(IU+I)=Z*W(I)+G(I)
      SIG=1./(ZZ*DGS**2)
      GO TO 70
   60 CONTINUE
      LINK=2
      DO 61 I=1,N
   61 W(IU+I)=W(I)
      IF(DGS+ALPHA*GS0.GT.0.) GO TO 62
      SIG=1./GS0
      GO TO 70
   62 CONTINUE
      SIG=-ZZ
   70 CONTINUE
      W(IV+1)=W(IU+1)
      DO 71 I=2,N
      IJ=I
      I1=I-1
      Z=W(IU+I)
      DO 72 J=1,I1
      Z=Z-H(IJ)*W(IV+J)
   72 IJ=IJ+N-J
   71 W(IV+I)=Z
      IJ=1
      DO 75 I=1,N
      Z=H(IJ)+SIG*W(IV+I)**2
      IF(Z.LE.0.) Z=DMIN
      IF(Z.LT.DMIN) DMIN=Z
      H(IJ)=Z
      W(IB+I)=W(IV+I)*SIG/Z
      SIG=SIG-W(IB+I)**2*Z
   75 IJ=IJ+NP-I
      IJ=1
      DO 80 I=1,N1
      IJ=IJ+1
      I1=I+1
      DO 80 J=I1,N
      W(IU+J)=W(IU+J)-H(IJ)*W(IV+I)
      H(IJ)=H(IJ)+W(IB+I)*W(IU+J)
   80 IJ=IJ+1
      GO TO (60,20),LINK
   90 CONTINUE
      IEXIT=3
      GO TO 94
   92 CONTINUE
   94 CONTINUE
      IND=1
      RETURN
  100 CONTINUE
      I=0
```

```
102 I=I+1
    Z=HH*XM(I)
    W(I)=W(I)+Z
    IND=6
    DO 240 K=1,N
240 P(K)=W(K)
    RETURN
242 F1=ALOSS
    ZACT=P(I)-XACT(I)
    G(I)=(F1-F)/ZACT
101 W(I)=W(I)-Z
    IF(I.LT.N) GO TO 102
    GO TO (18,54),LINK
110 CONTINUE
    I=0
112 I=I+1
    Z=HH*XM(I)
    W(I)=W(I)+Z
    IND=7
    DO 250 K=1,N
250 P(K)=W(K)
    RETURN
252 F1=ALOSS
    ZACT=P(I)-XACT(I)
    W(I)=W(I)-Z-Z
    IND=8
    DO 260 K=1,N
260 P(K)=W(K)
    RETURN
262 F2=ALOSS
    ZZACT=XACT(I)-P(I)
    G(I)=(F1-F2)/(ZACT+ZZACT)
111 W(I)=W(I)+Z
    IF(I.LT.N) GO TO 112
    GO TO (18,54),LINK
    END
```

```
      SUBROUTINE PRINT
C
C     PRINTS LOSS-FUNCTION,REGULATOR-PARAMETERS,GRADIENTS AND
C     HESSIAN MATRIX.
C
C     SUBROUTINE REQUIRED
C        HESS
C
      DIMENSION FT(14),VFT(8),A(8,8),H(36),G(8),RPO(8),TEXT(7),TAB(13),
     1          IREG(2)
      COMMON /RTCOM/IBUFF(512)
      EQUIVALENCE
     1(IBUFF(1),IND),(IBUFF(2),N),(IBUFF(62),G(1)),
     2(IBUFF(220),H(1)),(IBUFF(381),IREG(1)),
     3(IBUFF(400),A1),(IBUFF(402),A2),(IBUFF(410),IST),
     4(IBUFF(411),ALOSO),(IBUFF(413),RPO(1))
C
      DATA TEXT/5HDC-SE,3HRVO,3HBOM,3HPI ,3HPD ,3HPID,3HB-B/
      DATA TAB/5HALOSS,3HG1 ,3HTI1,3HTD1,3HGD1,3HG2 ,3HTI2,3HTD2,3HGD2,
     1        3HK1 ,3HDZ1,3HK2 ,3HDZ2/
      DATA FT(1),FT(3),FT(14)/4H(1X,, 5HF13.4, 1H)/
      DATA VFT/5HF9.3,, 3H9X,, 1H,, 5HE13.1, 1H3, 3H13X, 1H2, 4H26X,/
C
C
      IF(IST.NE.0) GO TO 10
C
      WRITE(16,100)
100   FORMAT(1H1,33HOPTIMERING AV REGULATORPARAMETRAR/1X,33(1H*)////
     1      1X,8HPROCESS:,15X,10HREGULATOR:)
      IF(IREG(2).NE.0) GO TO 1
      WRITE(16,101) TEXT(1),TEXT(2),TEXT(3+IREG(1))
101   FORMAT(1H+,9X,A5,A3,17X,A3,7H (LAGE))
      IF(IREG(1).EQ.4) WRITE(16,102) A1
102   FORMAT(1H+,48X,2HA=,F5.2)
      GO TO 2
1     WRITE(16,103) TEXT(3),TEXT(3+IREG(1))
103   FORMAT(1H+,9X,A3,22X,A3,7H (LAGE))
      IF(IREG(1).EQ.4) WRITE(16,102) A1
      WRITE(16,104) TEXT(3+IREG(2))
104   FORMAT(35X,A3,9H (VINKEL))
      IF(IREG(2).EQ.4) WRITE(16,102) A2
C
2     IF(IREG(1).EQ.4) GO TO 3
      WRITE(16,105) TAB(1),(TAB(1+I),I=1,4)
105   FORMAT(//4X,A5,8X,8(A3,10X))
      GO TO 4
3     WRITE(16,105) TAB(1),TAB(10),TAB(11)
4     IF(IREG(2).EQ.0) GO TO 6
      IF(IREG(2).EQ.4) GO TO 5
      WRITE(16,106) (TAB(5+I),I=1,4)
106   FORMAT(1H+,68X,4(A3,10X))
      GO TO 6
5     WRITE(16,106) TAB(12),TAB(13)
6     WRITE(16,107)
107   FORMAT(1H )
      IST=1
C
10    FT(2)=VFT(1)
      FT(4)=VFT(3)
      FT(5)=FT(3)
      IF(IREG(1).EQ.2) FT(5)=VFT(4)
      FT(6)=VFT(3)
      FT(7)=VFT(5)
```

```
      IF(IREG(1).EQ.4) FT(7)=VFT(8)
      FT(8)=FT(3)
      FT(9)=VFT(3)
      FT(10)=FT(3)
      IF(IREG(2).EQ.2) FT(10)=VFT(4)
      FT(11)=VFT(3)
      FT(12)=VFT(7)
      FT(13)=FT(3)
      NR=N
      IF(IREG(1).GE.3) GO TO 11
      NR=NR+1
      IF(IREG(1).EQ.1) NR=NR+1
11    IF(IREG(2).EQ.0.OR.IREG(2).GE.3) GO TO 12
      NR=NR+1
      IF(IREG(2).EQ.1) NR=NR+1
12    WRITE(16,FT) ALOSO,(RPO(I),I=1,NR)
C
      IF(IND.EQ.1) WRITE(16,110)
110   FORMAT(1X,7HRESTART)
      IF(IND.EQ.3) GO TO 13
C
      RETURN
C
13    FT(2)=VFT(2)
      IF(IREG(1).EQ.1) FT(7)=VFT(8)
      IF(IREG(1).EQ.2) FT(5)=VFT(6)
      IF(IREG(2).EQ.2) FT(10)=VFT(6)
      WRITE(16,FT) (G(I),I=1,N)
      WRITE(16,111)
111   FORMAT(1H+,10H GRADIENT:/16X,14HHESSIAN MATRIX)
      CALL HESS(N,H,A)
      DO 14 I=1,N
14    WRITE(16,FT) (A(I,K),K=1,N)
C
      RETURN
      END
```

```
      SUBROUTINE HESS(N,H,A)
C
C     ROUTINE FOR TRANSFORMING THE HESSIAN MATRIX FROM THE PRODUCT
C     FORM LD(L)T TO A SINGEL MATRIX.
C
C     N         THE NUMBER OF VARIABLES (MIN 2,NO MAX)
C     H         A REAL ARRAY OF N*(N+1)/2 ELEMENTS IN WHICH AN ESTIMATE OF
C               THE HESSIAN MATRIX IS STORED.THE MATRIX IS REPRESENTED IN
C               THE PRODUCT FORM LD(L)T WHERE L IS A LOWER TRIANGULAR
C               MATRIX WITH UNIT DIAGONALS AND D IS A DIAGONAL MATRIX.
C               THE LOWER TRIANGEL OF L IS STORED BY COLUMNS IN H EXCEPTING
C               THAT THE UNIT DIAGONAL ELEMENTS ARE REPLACED BY THE
C               CORRESPONDING ELEMENTS OF D.
C     A         A REAL FIELD IN WHICH THE TRANSFORMED HESSIAN MATRIX IS
C               STORED.
C
      DIMENSION H(1),A(8,8)
C
C     COMPUTE THE LOWER TRIANGULAR MATRIX L
      M=0
      DO 1 I=1,N
      DO 1 J=1,N
1     A(I,J)=0.
      DO 2 J=1,N
      DO 2 I=J,N
      M=M+1
2     A(I,J)=H(M)
      DO 3 I=1,N
3     A(I,I)=1.
C
C     COMPUTE LD
      K=1
      DO 4 J=1,N
      M=K
      DO 4 I=J,N
      K=K+1
4     A(I,J)=A(I,J)*H(M)
C
C     COMPUTE LD(L)T
      NJ=N-1
      DO 5 J=1,N
      JN=N+1-J
      NK=JN-1
      DO 5 I=1,N
      M=0
      DO 5 K=1,NK
      M=M+N+1-K
      MM=M+1-J
5     A(I,JN)=A(I,JN)+A(I,K)*H(MM)
C
      RETURN
      END
```