A Self-Tuning Regulator for Nonminimum Phase Systems

Åström, Karl Johan

1974

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

Link to publication

Citation for published version (APA):
Åström, K. J. (1974). *A Self-Tuning Regulator for Nonminimum Phase Systems*. (Research Reports TFRT-3113). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:
1

A SELF-TUNING REGULATOR FOR NONMINIMUM PHASE SYSTEMS

K.J. ÅSTRÖM

Report 7411(C) June 1974
Lund Institute of Technology
Division of Automatic Control

A SELF-TUNING REGULATOR FOR NONMINIMUM PHASE SYSTEMS

by

K. J. Åström

Abstract

This report describes an algorithm for a self-tuning regulator for a linear system with one input and one output. It is well known that the minimum variance regulator is very sensitive to parameter variations if the system is nonminimum phase. For such systems a suboptimal strategy with the property that all the poles of the closed loop system are inside a circle with radius r_o is therefore computed. The algorithm can thus be used for systems with unknown time delays and for nonminimum phase systems. To use the algorithm it is not necessary to know the order of the system or the timedelay exactly. Fewer parameters are, however, obtained if these quantities are known. If the algorithm converges, it will converge to a stable minimum variance regulator at least in the case of uncorrelated residuals. The properties of the algorithm when the residuals are correlated are not yet known.

1. INTRODUCTION

This algorithm is an extension of the one discussed in [1]. It permits the system to be controlled to be nonminimum phase. The algorithm is based on the model

$$y(t) + a_1y(t-1) + \dots + a_ny(t-n) = b_1u(t-k-1) + \dots +$$

$$+ b_mu(t-k-m) + v(t-k-m) \qquad (1.1)$$

The purpose of the algorithm is to obtain a control strategy for a linear system described by (1.1) with correlated disturbances. The criterion is taken as

$$\min E[y^2(t) + q_2u^2(t)] \qquad (1.2)$$

and the admissible control strategies are assumed to be such that the control signal at time t can be a function of the past control signals and the process outputs observed up to time t - 1. The algorithm is called self-tuning because it combines the steps of identification and control. To use the algorithm it is necessary to give the parameters

n = NA number of parameters a_i in the model (1.1)

m = NB number of parameters b_i in the model (1.1)

k - number of delays in the model.

It is also necessary to provide initial estimates of the parameter values, as well as an estimate of the uncertainty of the parameters i.e.

φ - parameter estimates

P - covariance of parameter estimates.

The algorithm contains some parameters which influence the convergence

r_ℓ - weighting factor in least squares parameter estimation

r_o - bound on poles of closed loop system

NLOP - maximum number of iterations when solving the riccatiequation

EPS - test quantity to terminate the iteration of the riccatiequation.

The computing time required in each step can be reduced significantly if an initial estimate to the solution of the riccatiequation is provided in each step. This is a typical example of the trade off between computing time and storage.

The algorithm consists of the following steps:

- o Identification
- o Determination of a state model
- o Determination of the state vector
- o Prediction
- o Computation of the control law
- o Computation of the control signal

These different steps are discussed in more detail in the following sections.

2. IDENTIFICATION

The parameters of the model (1) are determined using straight-forward least squares i.e. the equations

$$\begin{bmatrix} -y(t-1) & -y(t-2) & \ldots & -y(t-n) & u(t-k-1),\ldots,u(t-k-m) \\ -y(t-2) & -y(t-3) & \ldots & -y(t-n-1) & u(t-k-2),\ldots,u(t-k-m-1) \\ \vdots & & & & \\ & & & & \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \\ a_n \\ b_1 \\ b_2 \\ \\ b_m \end{bmatrix} = \begin{bmatrix} y(t) \\ y(t-1) \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \qquad (2.1)$$

are solved in each step using straightforward least squares. To forget past data an exponential weighting is also introduced. The criterion is chosen as

$$\sum_{s=0}^{t} y^2(s)\,\varepsilon^{t-s} \qquad (2.2)$$

3. A STATE MODEL

The control law which minimizes the criterion (1.2) for the system (1.1) can be determined by direct polynomial manipulations. The calculation of the control strategy can, however, be simplified if a state space representation is used. A state model can be introduced in many different ways. One possibility is given on the next page.

$$x(t+1) = \begin{bmatrix} -a_1 & 1 & 0 & \dots & 0 \\ -a_2 & 0 & 1 & \dots & 0 \\ \vdots & & & & \\ -a_{n-1} & 0 & 0 & \dots & 1 \\ -a_n & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} v(t)$$

$$y(t) = [1 \quad 0 \quad 0 \quad \dots \quad 0]x(t) \qquad (3.1)$$

For simplicity we have given the state model for case $n < m+k$ only. The appearance of the state model in general is shown in the following code and test examples.

The state model obtained is represented as:

$$x(t+1) = \begin{bmatrix} -\tilde{a}_1 & 1 & 0 & \dots & 0 \\ -a_2 & 0 & 1 & \dots & 0 \\ & & & \vdots & \\ -\tilde{a}_{\tilde{n}-1} & 0 & 0 & \dots & 1 \\ -\tilde{a}_{\tilde{n}} & 0 & 0 & \dots & 0 \end{bmatrix} x(t) + \begin{bmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \\ \tilde{b}_{n-1} \\ \tilde{b}_n \end{bmatrix} u(t)$$

$$y(t) = [1 \quad 0 \quad 0 \quad \dots \quad 0]x(t)$$

where

$$n = \max(n, m+k)$$

and $\tilde{a}_i$ and $\tilde{b}_i$ denote the transformed elements. Notice that some elements of $\tilde{a}$ and $\tilde{b}$ are zero. For simplicity we will leave out $\sim$ in the following. To see which elements are zero we will give some examples.

Example 3.1

NA = 1, NB = 2, K = 0

TH = (1,2,3)

AS = (1,0), BS = (2,3)

Example 3.2

NA = 1, NB = 2, K = 1

TH = (1,2,3)

AS = (1,0,0), BS = (0,2,3)

Example 3.3

NA = 1, NB = 2, K = 2

TH = (1,2,3)

AS = (1,0,0,0), BS = (0,0,2,3)

Example 3.4

NA = 4, NB = 2, K = 0

TH = (1,2,3,4,5,6)

AS = (1,2,3,4), BS = (5,6,0,0)

Example 3.5

NA = 4, NB = 2, K = 1

TH = (1,2,3,4,5,6)

AS = (1,2,3,4), BS = (0,5,6,0)

Example 3.6

NA = 4, NB = 2, K = 2

TH = (1,2,3,4,5,6)

AS = (1,2,3,4), BS = (0,0,5,6)

Example 3.7

NA = 4, NB = 2, K = 3

TH = (1,2,3,4,5,6)

AS = (1,2,3,4,0), BS = (0,0,0,5,6)

Example 3.8

NA = 4, NB = 2, K = 4

TH = (1,2,3,4,5,6)

AS = (1,2,3,4,0,0), BS = (0,0,0,0,5,6)

The code which generates the state model (3.1) from the parameters $a_1, \ldots, a_n, b_1, \ldots, b_m$ is given in List 1. The printout shown in List 2 shows that the program gives the correct results for the test examples given above.

List 1. PROGRAM FOR TESTING GENERATION AND SCALING OF STATE MODEL IN STURE.

```
C          TSTU21
C          THIS PROGRAM TESTS PART OF STURE2
           DIMENSION TH(12),AS(12),BS(12)
           DO 12 I=1,12
12         TH(I)=FLOAT(I)
C
           I1=1
1          GO TO (2,3,4,5,6,7,8,9,10),I1
2          I1=2
           NA=1
           NB=2
           K=0
           GO TO 50
3          I1=3
           K=1
           GO TO 50
4          I1=4
           K=2
           GO TO 50
5          I1=5
           NA=4
           K=0
           GO TO 50
6          I1=6
           K=1
           GO TO 50
7          I1=7
           K=2
           GO TO 50
8          I1=8
           K=3
           GO TO 50
9          I1=9
           K=4
           GO TO 50
10         STOP
50         CONTINUE
           N=MAX0(NA,NB+K)
           NM1=N-1
           R0=.1
C
           WRITE (6,100) NA,NB,K,N,R0
100        FORMAT (4H NA=,I5,4H NB=,I5,3H K=,I5,3H N=,I5,
           14H R0=,E12.4)
C
C          COMPUTE AND SET SCALED SYSTEM PARAMETERS
C
           CALL MOVE (TH(1),AS(1),NA+NA)
           IF (N-NA-1) 20,21,22
22         AS(NA+1)=0.0
           NS=N-NA-1
           CALL MOVE (AS(NA+1),AS(NA+2),NS+NS)
           GO TO 20
21         AS(NA+1)=0.0
20         IF (K-1) 24,25,26
26         BS(1)=0.0
           NS=K-1
           CALL MOVE (BS(1),BS(2),NS+NS)
           GO TO 24
25         BS(1)=0.0
24         CALL MOVE (TH(NA+1),BS(K+1),NB+NB)
           N1=NB+K+1
           IF (N-N1) 28,29,30
```

```
30        BS(N1)=0.0
          NS=N-N1
          CALL MOVE(BS(N1),BS(N1+1),NS+NS)
          GO TO 28
29        BS(N1)=0.0
28        CONTINUE
          WRITE (6,110)
110       FORMAT (18H VECTORS AS AND BS)
C
          WRITE (6,101) (AS(I),I=1,N)
          WRITE (6,101) (BS(I),I=1,N)
101       FORMAT (10E12.4)
C
          SF=1.
          DO 32 I=1,NM1
          SF=SF*P0
          N1=N-I
          AS(N1)=AS(N1)*SF
32        BS(N1)=BS(N1)*SF
          WRITE (6,111)
111       FORMAT (25H SCALED VECTORS AS AND BS)
C
          WRITE (6,101) (AS(I),I=1,N)
          WRITE (6,101) (BS(I),I=1,N)
C
          GO TO 1
C
          END
```

List 2. PRINTOUT OF TEST OF GENERATION AND SCALING OF PARAMETERS IN STATE MODEL.

```
NA=     1 NB=     2 K=     0 N=     2 R0=  0.1000E+00  Example 3.1
VECTORS AS AND BS
 0.1000E+01  0.0000E+00
 0.2000E+01  0.3000E+01
SCALED VECTORS AS AND BS
 0.1000E+00  0.0000E+00  ok
 0.2000E+00  0.3000E+01  ok
NA=     1 NB=     2 K=     1 N=     3 R0=  0.1000E+00  Example 3.2
VECTORS AS AND BS
 0.1000E+01  0.0000E+00  0.0000E+00  ok
 0.0000E+00  0.2000E+01  0.3000E+01  ok
SCALED VECTORS AS AND BS
 0.1000E-01  0.0000E+00  0.0000E+00
 0.0000E+00  0.2000E+00  0.3000E+01
NA=     1 NB=     2 K=     2 N=     4 R0=  0.1000E+00  Example 3.3
VECTORS AS AND BS
 0.1000E+01  0.0000E+00  0.0000E+00  0.0000E+00  ok
 0.0000E+00  0.0000E+00  0.2000E+01  0.3000E+01  ok
SCALED VECTORS AS AND BS
 0.1000E-02  0.0000E+00  0.0000E+00  0.0000E+00
 0.0000E+00  0.0000E+00  0.2000E+00  0.3000E+01
NA=     4 NB=     2 K=     0 N=     4 R0=  0.1000E+00  Example 3.4
VECTORS AS AND BS
 0.1000E+01  0.2000E+01  0.3000E+01  0.4000E+01  ok
 0.5000E+01  0.6000E+01  0.0000E+00  0.0000E+00  ok
SCALED VECTORS AS AND BS
 0.1000E-02  0.2000E-01  0.3000E+00  0.4000E+01
 0.5000E-02  0.6000E-01  0.0000E+00  0.0000E+00
NA=     4 NB=     2 K=     1 N=     4 R0=  0.1000E+00  Example 3.5
VECTORS AS AND BS
 0.1000E+01  0.2000E+01  0.3000E+01  0.4000E+01  ok
 0.0000E+00  0.5000E+01  0.6000E+01  0.0000E+00  ok
SCALED VECTORS AS AND BS
 0.1000E-02  0.2000E-01  0.3000E+00  0.4000E+01
 0.0000E+00  0.5000E-01  0.6000E+00  0.0000E+00
NA=     4 NB=     2 K=     2 N=     4 R0=  0.1000E+00  Example 3.6
VECTORS AS AND BS
 0.1000E+01  0.2000E+01  0.3000E+01  0.4000E+01  ok
 0.0000E+00  0.0000E+00  0.5000E+01  0.6000E+01  ok
SCALED VECTORS AS AND BS
 0.1000E-02  0.2000E-01  0.3000E+00  0.4000E+01
 0.0000E+00  0.0000E+00  0.5000E+00  0.6000E+01
NA=     4 NB=     2 K=     3 N=     5 R0=  0.1000E+00  Example 3.7
VECTORS AS AND BS
 0.1000E+01  0.2000E+01  0.3000E+01  0.4000E+01  0.0000E+00  ok
 0.0000E+00  0.0000E+00  0.0000E+00  0.5000E+01  0.6000E+01  ok
SCALED VECTORS AS AND BS
 0.1000E-03  0.2000E-02  0.3000E-01  0.4000E+00  0.0000E+00
 0.0000E+00  0.0000E+00  0.0000E+00  0.5000E+00  0.6000E+01
NA=     4 NB=     2 K=     4 N=     6 R0=  0.1000E+00
VECTORS AS AND BS                                    Example 3.8
 0.1000E+01  0.2000E+01  0.3000E+01  0.4000E+01  0.0000E+00  0.0000E+0
 0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  0.5000E+01  0.6000E+0
SCALED VECTORS AS AND BS
 0.1000E-04  0.2000E-03  0.3000E-02  0.4000E-01  0.0000E+00  0.0000E+0
 0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  0.5000E+00  0.6000E+0
```

4. THE STATE VECTOR

In order to use the state model given in section 3 it is necessary to compute the state variables from the inputs and outputs. This can be done as follows:

$$x_1(t) = y(t)$$

$$x_n(t) = -a_n x_1(t-1) + b_n u(t-1) = -a_n y(t-1) + b_n u(t-1)$$

$$x_{n-1}(t) = -a_{n-1} y(t-1) + x_n(t-1) + b_{n-1} u(t-1)$$

$$\vdots$$

$$x_2(t) = -a_2 y(t-1) + x_3(t-1) + b_2 u(t-1) \qquad (4.1)$$

Notice that some care must be exercised in the coding since some coefficients of a and b are zero and since the u:s and the y:s of the corresponding terms are not stored. It is thus necessary to distinguish three cases

1) NA < NB + K
2) NA = NB + K
3) NA > NB + K

The details are most conveniently expressed using a programing language. The FORTRAN code is given in the test program in List 3. The program is written in such a way that erroneous values in the vectors AS and BS outside the range defined by NA, NB and K will not give wrong results. For example, for NA = 2 the value AS(3) = 1000 will not cause any difficulties.

List 3. TESTPROGRAM FOR THE GENERATION OF STATE VARIABLES AND PREDICTION.


```
C       TSTU22
C       THIS PROGRAM TESTS COMPUTATION OF STATE VARIABLES IN STURE2
        DIMENSION AS(12),BS(12),Y(12),U(12),X(12)
        N=3
        I1=1
1       GO TO (2,3,4,5),I1
2       I1=2
        NA=2
        NB=3
        K=0
        Y(1)=0.
        Y(2)=-4.
        Y(3)=3.
        U(1)=6000.
        U(2)=1.
        U(3)=-1.
        AS(1)=1.
        AS(2)=2.
        AS(3)=1000.
        BS(1)=3.
        BS(2)=2.
        BS(3)=1.
        GO TO 50
C
3       I1=3
        NA=3
        NB=2
        K=0
        Y(1)=-1.
        Y(2)=-4.
        Y(3)=3.
        AS(3)=3.
        BS(3)=1000.
        GO TO 50
4       I1=4
        BS(3)=1
        NB=3
        Y(1)=0.
        GO TO 50
5       STOP
50      CONTINUE
        N=MAX0(NA,NB+K)
        NAP1=NA+1
        WRITE (6,110)
110     FORMAT (18H VECTORS AS AND BS)
        WRITE (6,100) (AS(I),I=1,N)
        WRITE (6,100) (BS(I),I=1,N)
100     FORMAT (10E12.4)
        WRITE (6,111)
111     FORMAT (16H VECTORS U AND Y)
        WRITE (6,100) (U(I),I=1,N)
        WRITE (6,100) (Y(I),I=1,N)
        X(1)=Y(1)
        IF(NA-NB-K) 40,41,42
40      DO 43 I=2,NA
        NS=NA-I+1
        NSB=N-I+1
43      X(I)=-SCAPRO(AS(I),1,Y(2),1,NS)+SCAPRO(BS(I),1,U(2),1,NSB)
        DO 44 I=NAP1,N
        NS=N-I+1
44      X(I)=SCAPRO(BS(I),1,U(2),1,NS)
        GO TO 49
41      DO 45 I=2,N
```

```
      NS=N-I+1
45    X(I)=-SCAPRO(AS(I),1,Y(2),1,NS)+SCAPRO(BS(I),1,U(2),1,NS)
      GO TO 49
42    N1=NR+K
      DO 46 I=2,N1
      NSA=N-I+1
      NS=N1-I+1
46    X(I)=-SCAPRO(AS(I),1,Y(2),1,NSA)+SCAPRD(BS(I),1,U(2),1,NS)
      N1=N1+1
      DO 47 I=N1,N
      NS=N-N1+1
47    X(I)=-SCAPRO(AS(I),1,Y(2),1,NS)
49    CONTINUE
      WRITE (6,112)
      WRITE (6,100) (X(I),I=1,N)
112   FORMAT (13H THE STATE IS)
C
C     PREDICT STATE VARIABLES
C
      RS=X(1)
      NM1=N-1
      DO 60 I=1,NM1
60    X(I)=-AS(I)*RS+X(I+1)+BS(I)*U(1)
      X(N)=-AS(N)*RS+BS(N)*U(1)
C
      WRITE (6,113)
113   FORMAT (16H PREDICTED STATE)
      WRITE (6,100) (X(I),I=1,N)
      GO TO 1
      END
```

To test the generation of state variables some examples will be computed by hand and compared with the results of the program given in List 3.

Example 4.1

$$x(t+1) = \begin{bmatrix} -1 & 1 & 0 \\ -2 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} u(t)$$

$$y(t) = [1 \quad 0 \quad 0]x(t)$$

Assume

$u(0) = 1$
$u(1) = -1$
$u(2) = 1$
$u(3) = 6000$
and $x(0) = \mathrm{col}[0 \quad 0 \quad 0]$

Then

$x(1) = \mathrm{col}[3 \quad 2 \quad 1]$
$x(2) = \mathrm{col}[-4 \quad -7 \quad -1]$
$x(3) = \mathrm{col}[0 \quad 9 \quad 1]$

The program receives the input-output variables and the model parameters and computes the state x(3). As seen by the results in List 4, the result is correct. Notice that the erroneous value of A$(3) 1000 instead of zero does not influence the results.

Example 4.2

$$x(t+1) = \begin{bmatrix} -1 & 1 & 0 \\ -2 & 0 & 1 \\ -3 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 3 \\ 2 \\ 0 \end{bmatrix} u(t)$$

$$y(t) = [1 \quad 0 \quad 0]x(t)$$

Assume

$u(0) = 1$
$u(1) = -1$
$u(2) = 1$
$u(3) = 6000$

and

$x(0) = \mathrm{col}[0 \quad 0 \quad 0]$

Then

$x(1) = \mathrm{col}[3 \quad 2 \quad 0]$
$x(2) = \mathrm{col}[-4 \quad -8 \quad -9]$
$x(3) = \mathrm{col}[-1 \quad 1 \quad 12]$

The program printout in List 4 shows that the calculation of x(3) is correct. The erroneous value of BS(3) 1000 instead of zero does not influence the result because NB = 2.

Example 4.3

$$x(t+1) = \begin{bmatrix} -1 & 1 & 0 \\ -2 & 0 & 1 \\ -3 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} u(t)$$

$$y(t) = [1 \quad 0 \quad 0]x(t)$$

Assume

$u(0) = 1$
$u(1) = -1$
$u(2) = 1$

and

$x(0) = \text{col}[0 \quad 0 \quad 0]$

Then

$x(1) = \text{col}[3 \quad 2 \quad 1]$
$x(2) = \text{col}[-4 \quad -7 \quad -10]$
$x(3) = \text{col}[0 \quad 0 \quad 13]$

The printout of the results shown in List 4 shows that the calculation of x(3) is correct.

List 4. PRINTOUT OF RESULTS OF PROGRAM TSTU22.

Example 4.1

```
VECTORS AS AND BS
 0.1000E+01  0.2000E+01  0.1000E+04
 0.3000E+01  0.2000E+01  0.1000E+01
VECTORS U AND Y
 0.6000E+04  0.1000E+01 -0.1000E+01
 0.0000E+00 -0.4000E+01  0.3000E+01
THE STATE IS
 0.0000E+00  0.9000E+01  0.1000E+01   ok
PREDICTED STATE
 0.1801E+05  0.1200E+05  0.6000E+04
```

Example 4.2

```
VECTORS AS AND BS
 0.1000E+01  0.2000E+01  0.3000E+01
 0.3000E+01  0.2000E+01  0.1000E+04
VECTORS U AND Y
 0.6000E+04  0.1000E+01 -0.1000E+01
-0.1000E+01 -0.4000E+01  0.3000E+01
THE STATE IS
-0.1000E+01  0.1000E+01  0.1200E+02   ok
PREDICTED STATE
 0.1800E+05  0.1201E+05  0.6000E+07
```

Example 4.3

```
VECTORS AS AND BS
 0.1000E+01  0.2000E+01  0.3000E+01
 0.3000E+01  0.2000E+01  0.1000E+01
VECTORS U AND Y
 0.6000E+04  0.1000E+01 -0.1000E+01
 0.0000E+00 -0.4000E+01  0.3000E+01
THE STATE IS
 0.0000E+00  0.0000E+00  0.1300E+02   ok.
PREDICTED STATE
 0.1800E+05  0.1201E+05  0.6000E+04
```

5. PREDICTION

The state variable $x(t)$ given by (4.1) is a function of the data obtained up to time t. When it is postulated that the control signal to be applied at time $t + 1$ should be a function of data observed up to time t only, it is necessary to predict the state i.e. compute $\hat{x}(t+1|t)$ and use the feedback law

$$u(t+1) = -L\,\hat{x}(t+1|t)$$

The prediction is easily obtained using the state space representation; we have

$$\hat{x}(t+1|t) = \Phi\, x(t) + \Gamma\, u(t) \tag{5.1}$$

The corresponding code is already given in List 5. The program TSTU23 only differs from TSTU22 in the sense that the data supplied is somewhat different. If it is desired to have a control law in such a way that $u(t)$ is a function of data obtained up to time t i.e. $y(t)$, $y(t-1), \ldots$, the only modification of the program that is necessary is to delete the prediction step. The prediction step is tested by the test examples 4.1, 4.2 and 4.3. The predicted states should be

List 5. PROGRAM FOR TESTING GENERATION OF STATE VARIABLES AND PREDICTION


```
C       TSTU23
C       THIS PROGRAM TESTS COMPUTATION OF STATE VARIABLES IN STURE2
        DIMENSION AS(12),BS(12),Y(12),U(12),X(12)
        N=3
        I1=1
1       GO TO (2,3,4,5),I1
2       I1=2
        NA=2
        NB=3
        K=0
        Y(1)=-4.
        Y(2)=3.
        Y(3)=0.
        U(1)=1.
        U(2)=-1.
        U(3)=1.
        AS(1)=1.
        AS(2)=2.
        AS(3)=0.
        BS(1)=3.
        BS(2)=2.
        BS(3)=1.
        GO TO 50
C
3       I1=3
        NA=3
        NB=2
        K=0
        AS(3)=3.
        BS(3)=0.
        GO TO 50
4       I1=4
        BS(3)=1.
        NB=3
        GO TO 50
5       STOP
50      CONTINUE
        N=MAX0(NA,NB+K)
        NAP1=NA+1
        WRITE (6,110)
110     FORMAT (18H VECTORS AS AND BS)
        WRITE (6,100) (AS(I),I=1,N)
        WRITE (6,100) (BS(I),I=1,N)
100     FORMAT (10E12.4)
        WRITE (6,111)
111     FORMAT (16H VECTORS U AND Y)
        WRITE (6,100) (U(I),I=1,N)
        WRITE (6,100) (Y(I),I=1,N)
        X(1)=Y(1)
        IF(NA-NB-K) 40,41,42
40      DO 43 I=2,NA
        NS=NA-I+1
        NSB=N-I+1
43      X(I)=-SCAPRO(AS(I),1,Y(2),1,NS)+SCAPRO(BS(I),1,U(2),1,NSB)
        DO 44 I=NAP1,N
        NS=N-I+1
44      X(I)=SCAPRO(BS(I),1,U(2),1,NS)
        GO TO 49
41      DO 45 I=2,N
        NS=N-I+1
45      X(I)=-SCAPRO(AS(I),1,Y(2),1,NS)+SCAPRO(BS(I),1,U(2),1,NS)
        GO TO 49
42      N1=NB+K
```

```
      DO 46 I=2,N1
      NSA=N-I+1
      NS=N1-I+1
46    X(I)=-SCAPRO(AS(I),1,Y(2),1,NSA)+SCAPRO(BS(I),1,U(2),1,NS)
      N1=N1+1
      DO 47 I=N1,N
      NS=N-N1+1
47    X(I)=-SCAPRO(AS(I),1,Y(2),1,NS)
49    CONTINUE
      WRITE (6,112)
      WRITE (6,100) (X(I),I=1,N)
112   FORMAT (13H THE STATE IS)
C
C     PREDICT STATE VARIABLES
C
      RS=X(1)
      NM1=N-1
      DO 60 I=1,NM1
60    X(I)=-AS(I)*RS+X(I+1)+BS(I)*U(1)
      X(N)=-AS(N)*RS+BS(N)*U(1)
C
      WRITE (6,113)
113   FORMAT (16H PREDICTED STATE)
      WRITE (6,100) (X(I),I=1,N)
      GO TO 1
      END
```

To test the program the systems used in the previous section are exploited. The calculated state agrees with x(2) and the predicted state is equal to x(3) since there are no measurement errors. The printout of the test program is given in List 6.

List 6. PRINTOUT OF TESTPROGRAM TSTU23 FOR TESTING PREDICTION OF STATES.

```
VECTORS AS AND BS                                      Example 4.1
 0.1000E+01  0.2000E+01  0.0000E+00
 0.3000E+01  0.2000E+01  0.1000E+01
VECTORS U AND Y
 0.1000E+01 -0.1000E+01  0.1000E+01
-0.4000E+01  0.3000E+01  0.0000E+00
THE STATE IS
-0.4000E+01 -0.7000E+01 -0.1000E+01   OK
PREDICTED STATE
 0.0000E+00  0.9000E+01  0.1000E+01   OK
VECTORS AS AND BS
 0.1000E+01  0.2000E+01  0.3000E+01                    Example 4.2
 0.3000E+01  0.2000E+01  0.0000E+00
VECTORS U AND Y
 0.1000E+01 -0.1000E+01  0.1000E+01
-0.4000E+01  0.3000E+01  0.0000E+00
THE STATE IS
-0.4000E+01 -0.8000E+01 -0.9000E+01   OK
PREDICTED STATE
-0.1000E+01  0.1000E+01  0.1200E+02   OK
VECTORS AS AND BS
 0.1000E+01  0.2000E+01  0.3000E+01                    Example 4.3
 0.3000E+01  0.2000E+01  0.1000E+01
VECTORS U AND Y
 0.1000E+01 -0.1000E+01  0.1000E+01
-0.4000E+01  0.3000E+01  0.0000E+00
THE STATE IS
-0.4000E+01 -0.7000E+01 -0.1000E+02   OK
PREDICTED STATE
 0.0000E+00  0.0000E+00  0.1300E+02   OK
```

6. THE CONTROL LAW

The control law will now be determined. We will then assume complete separation of identification and control i.e. we will determine a control law based on the assumption that the estimated parameters are correct. The parameter uncertainties will thus be neglected. The system model is given by

$$x(t+1) = \begin{bmatrix} -a_1 & 1 & 0 & \dots & 0 \\ -a_2 & 0 & 1 & \dots & 0 \\ \vdots & & & & \\ -a_{n-1} & 0 & 0 & \dots & 1 \\ -a_n & 0 & 0 & \dots & 0 \end{bmatrix} x(t) + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix} u(t)$$

$$y(t) = [1 \quad 0 \quad 0 \dots 0 \; x(t)] \tag{6.1}$$

The criterion is chosen as

$$\sum_{t=1}^{N} [y^2(t) + q\, u^2(t)] \tag{6.2}$$

The control strategy, which minimizes (6.2), is given by

$$u(t) = -L(t)x(t) \tag{6.3}$$

where

$$L(t) = B^T S(t)A/(q+B^T S(t)B) \tag{6.4}$$

and S is the solution of the riccatiequation

$$S(t-1) = A^T S(t)(A-BL(t)) + Q_1 \tag{6.5}$$

with the initial condition

$$S(N) = 0 \tag{6.6}$$

The matrices A, B, and Q are given by

$$A = \begin{bmatrix} -a_1 & 1 & 0 & \dots & 0 \\ -a_2 & 0 & 1 & \dots & 0 \\ & & & & \\ -a_{n-1} & 0 & 0 & \dots & 1 \\ -a_n & 0 & 0 & \dots & 0 \end{bmatrix}, \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \\ b_{n-1} \\ b_n \end{bmatrix}$$

$$Q_1 = \mathrm{diag}[1,0,\ \dots,\ 0] \tag{6.7}$$

The computations are organized as follows

$B^T S \rightarrow S1$

$B^T SB + Q2 = S1B + Q_2 \rightarrow S3$

$B^T SA/R3 = S1A/R3 \rightarrow L$

$A - BL \rightarrow S1$

$SA \rightarrow S2$

$S = S2^T S1 + Q1$

The algorithm is given in a special subroutine CORI. See List 7. Notice that the riccatiequation may have several solutions. To obtain the solution so that all poles of the closed loop system are inside the unit circle, it is necessary to start the iteration with a positive definite matrix.

List 7. SUBROUTINE FOR STEADY STATE SOLUTION OF RICCATI EQUATION

```
      SUBROUTINE CORI(A,B,Q2,S,AL,N,R3,IS)
C     THIS SUBROUTINE ITERATES THE RICCATI EQUATION
C             S=AT*S*(A-B*L)+Q1       (*)
C             L=BT*S*A/(Q2+BT*S*B)   (**)
C     IN THE SPECIAL CASE WHEN
C             A IS A COMPANION MATRIX
C             Q1=DIAG(1,0,...,0)
C             B IS A VECTOR
C
C     AUTHOR K.J. ASTROM 72-01-03
C
C             A   - VECTOR CONTAINING THE FIRST COLUMN
C                   OF THE MATRIX A IN (*) I.E. A(I,1)=-A(I)
C             B   - VECTOR B IN (*) AND (**)
C             Q2  - SCALAR Q2 IN (**)
C             S   - SOLUTION OF RICCATI EQUATION
C             AL  - VECTOR AL IN (**)
C             N   - ACTUAL ORDER OF SYSTEM
C             IS  - DIMENSION PARAMETER OF MATRIX S
C             R3  - DENOMINATOR (Q2+BT*S*B)
C
C     USES DUM7 AND DUM8 OF COMMON/SLASK/
C
      DIMENSION A(1),B(1),AL(1),S(1,1)
      COMMON /SLASK/DUM(384),S1(8,8),S2(8,8)
C
      DO 10 I=1,N
10    S1(I,1)=SCAPRO(B(1),1,S(1,I),1,N)
      R3=SCAPRO(B(1),1,S1(1,1),1,N)+Q2
C
C     R3 NOW CONTAINS BT*S*B+Q2
C
      AL(1)=-SCAPRO(S1(1,1),1,A(1),1,N)/R3
      DO 14 I=2,N
      I1=I-1
14    AL(I)=S1(I1,1)/R3
C
C     COMPUTATION OF L=BT*S*A/(Q2+BT*S*B) COMPLETE
C     RESULT STORED INVECTOR AL
C
      R=AL(1)
      DO 20 I=1,N
20    S1(I,1)=-A(I)-B(I)*R
      DO 22 J=2,N
      R=AL(J)
      DO 22 I=1,N
      R1=0.
      IF (I+1-J) 22,23,22
23    R1=1.
22    S1(I,J)=R1-B(I)*R
```

```
C
C       S1 NOW CONTAINS A-B*L
C
        DO 24 I=1,N
24      S2(I,1)=-SCAPRO(A(1),1,S(I,1),IS,N)
        NM1=N-1
        DO 26 J=1,NM1
26      CALL MOVE(S(1,J),S2(1,J+1),N*N)
C
C       S2 NOW CONTAINS S*A
C
        DO 30 I=1,N
        DO 30 J=1,N


30      S(I,J)=SCAPRO(S2(1,I),1,S1(1,J),1,N)
        S(1,1)=S(1,1)+1.
        RETURN
        END
```

To test the subroutine CORI a special conversational program has been written which enables the user to enter system and loss function from teletype. This program is given in List 8. The program will be tested against analytical solutions.

List 8. CONVERSATIONAL PROGRAM FOR USING CORI

```
C        THIS IS A PROGRAM FOR USING CORI. THE PROGRAM REQUESTS
C        DATA FROM TTY, CALLS CORI AND PRINTS THE RESULTS ON DEV 6
         DIMENSION A(16),B(16),AL(16),S(16,16)
         DATA AY/3HYES/
1        N1=1
         CALL ATTLP6(1)
         WRITE (6,100)
100      FORMAT (11H TYPE ORDER)
         ICNTRL=1
         N=RTTFF(ICNTRL)
         WRITE (6,101)
101      FORMAT (19H TYPE ELEMENTS OF A)
         ICNTRL=1
         DO 10 I=1,N
10       A(I)=RTTFF(ICNTRL)
C
         WRITE (6,102)
102      FORMAT(19H TYPE ELEMENTS OF B)
         ICNTRL=1
         DO 12 I=1,N
12       B(I)=RTTFF(ICNTRL)
C
         WRITE (6,104)
104      FORMAT (8H TYPE Q2)
         ICNTRL=1
         Q2=RTTFF(ICNTRL)
         WRITE (6,106)
106      FORMAT (26H TYPE NUMBER OF ITERATIONS)
         ICNTRL=1
         N2=RTTFF(ICNTRL)
C        INITIALIZE S
         DO 20 I=1,N
         DO 20 J=1,N
20       S(I,J)=0.0
         DO 21 I=1,N
21       S(I,I)=1.0
C        MAIN LOOP
25       DO 30 IT=N1,N2
30       CALL CORI(A,B,Q2,S,AL,N,R3)
C
         WRITE (6,200)
200      FORMAT (9H S-MATRIX)
         DO 40 I=1,N
40       WRITE (6,300) (S(I,J),J=1,N)
300      FORMAT (6E12.4)
         WRITE (6,202)
202      FORMAT (9H L-VECTOR)
         WRITE (6,300) (AL(I),I=1,N)
C
         WRITE (6,204)
204      FORMAT (24H DO YOU WISH TO CONTINUE)
         READ (8,206) ANS
206      FORMAT (A5)
         IF (ANS.NE.AY) GO TO 1
         N1=N2
         WRITE (6,208)
208      FORMAT (37H TYPE NUMBER OF ADDITIONAL ITERATIONS)
```

Example 6.1

Consider the system

$$x(t+1) = \begin{bmatrix} -a & 1 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ b \end{bmatrix} u(t)$$

with the criterion defined by

$Q1 = 1, \quad Q2 = 0$

To find the steady state solution to the riccatiequation and the control law it is necessary to separate two cases.

Case 1. $|b| \leq 1$

In this case the Riccati equation has one steady state solution .

$$S = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

The corresponding optimal control law is given by

$L = [-a \quad 1]$

Case 2. $|b| > 1$

In this case the riccatiequation has two non-negative solutions. The one previously given and

$$S = \begin{bmatrix} s_1 & s_2 \\ s_2 & s_3 \end{bmatrix}$$

where

$$s_1 = \frac{(b-a)^2 + a^2(b^2-1)}{(a-b)^2}$$

$$s_2 = -\frac{a(b^2-1)}{(a-b)^2}$$

$$s_3 = \frac{b^2 - 1}{(a-b)^2}$$

The corresponding control law is given by

$$L = \frac{1 - ab}{b(a-b)} [-a \quad 1]$$

The program CORI will now be tested using some specific numerical values

$a = 0.5$

$b = 2$

The analytical solution gives

$$S = \begin{bmatrix} 4/3 & -2/3 \\ -2/3 & 4/3 \end{bmatrix}$$

$L = [0 \quad 0]$

The results obtained when using UCORI conversational are given next

```
LOADER V9A
>←UCORI.\,CORI

TYPE ORDER
#2
TYPE ELEMENTS OF A
#.5 Ø.
TYPE ELEMENTS OF B
#1. 2.
TYPE Q2
#2.
TYPE NUMBER OF ITERATIONS
#2
S-MATRIX
 Ø.1286E+Ø1 -Ø.5714E+ØØ
-Ø.5714E+ØØ  Ø.1143E+Ø1
L-VECTOR
-Ø.7143E-Ø1  Ø.1429E+ØØ
DO YOU WISH TO CONTINUE

YES
TYPE NUMBER OF ADDITIONAL ITERATIONS
#1Ø
S-MATRIX
 Ø.1333E+Ø1 -Ø.6667E+ØØ
-Ø.6667E+ØØ  Ø.1333E+Ø1
L-VECTOR
-Ø.2235E-Ø7  Ø.4473E-Ø7
DO YOU WISH TO CONTINUE
NO
```

The program will thus give the correct results after 10 iterations.

Example 6.2

Consider the system

$$x(t+1) = \begin{bmatrix} -1.5 & 1 \\ 0.7 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0.9 \end{bmatrix} u(t)$$

with the criterion Q2 = 0. It is straightforward to show that

$$S = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

L = [-1.5 1]

The results obtained by CORI are given below.

```
TYPE ORDER
#2
TYPE ELEMENTS OF A
#1.5 -.7
TYPE ELEMENTS OF E
#1. .9
TYPE Q2
#0.
TYPE NUMBER OF ITERATIONS
#10
S-MATRIX
 0.1073E+01 -0.3187E-01
-0.3187E-01  0.1399E-01
L-VECTOR
-0.1549E+01  0.1021E+01
DO YOU WISH TO CONTINUE
YES
TYPE NUMBER OF ADDITIONAL ITERATIONS
#100
S-MATRIX
 0.1000E+01  0.5315E-07
 0.6661E-07 -0.2561E-07
L-VECTOR
-0.1500E+01  0.1000E+01
DO YOU WISH TO CONTINUE
↑C
```

7. SCALING

$$y(t) + a_1y(t-1) + \dots + a_ny(t-n) = b_1u(t-k-1) + \dots +$$

$$+ b_nu(t-k-m) + v(t-k-m) \tag{7.1}$$

Now this equation can be rewritten as

$$y(t) + (a_1/r_o)[r_oy(t-1)] + \dots + (a_n/r_o^n)[r_o^ny(t-n)] =$$

$$= (b_1/r_o)[r_ou(t-k-1)] + \dots + (b_m/r_o^m)[r_o^nu(t-k-m)] + v(t-k-m) \tag{7.2}$$

The models (7.1) and (7.2) are obviously identical. The polynomial

$$\tilde{B}(z) = z^{m-1}(b_1/r_o) + z^{m-2}b_2/r_o^2 + \dots + b_m/r_o^m$$

has zeros outside the unit circle if the polynomial B has zeros outside the circle $|z| = r_o$. Solving the riccati-equation associated with $\tilde{A}$ and $\tilde{B}$ it is then insured that the zeros of $\tilde{B}$ outside $|z| = r_o$ are reflected inside the circle. An example illustrates the point.

<u>Example 7.1</u>

Consider the system

$$y(t) + a\, y(t-1) = b\, u(t-1)$$

The strategy which minimized $\Sigma\, y^2(t)$ is

$$u(t) = \frac{a^2}{b}\, y(t-1) + a\cdot u(t-1)$$

If the variables are transformed we get

$$\tilde{y}(t) + \tilde{a}\,\tilde{y}(t-1) = \tilde{b}\,\tilde{u}(t-1)$$

where

$$\tilde{a} = a/r_o, \quad \tilde{b} = b/r_o, \quad \tilde{y}(t-1) = r_o\,\tilde{y}(t-1), \quad \tilde{u}(t) = r_o\tilde{u}(t-1)$$

Hence

$$\tilde{u}(t) = \frac{\tilde{a}^2}{\tilde{b}}\,\tilde{y}(t-1) + \tilde{a}\,\tilde{u}(t-1) = \frac{a^2}{b}\,y(t) + a\,u(t-1)$$

$\square$

The code for the scaling is given previously.

8. THE COMPLETE ALGORITHM

A FORTRAN program for the complete algorithm is given in the Appendix. A test program TSTU2 was written in order to test the algorithm. A listing of this program is found in List 9. In the test program the parameters a_1, b_1 and b_2 are read from the teletype. The program then computes the steady state solution to the riccatiequation and the feedback gains. A sequence of inputs and outputs are then generated and fed into the algorithm. The control actions computed by STURE, as well as intermediate results, are then printed.

```
C         TSTU2
C         THIS IS A TEST PROGRAM FOR STURE2
C
          DIMENSION U(10),Y(10),UM(10),YM(10),UC(10),TH(12),P(7,7),S(6,(
          1,S0(6,6)
          DIMENSION A(10),B(10)
          DIMENSION UCC(10)
          COMMON /SLASK/ DUM(272),AS(8),BS(8),US(8),YS(8),X(8),AL(8)
          1,R(8,8),DUM78(128)
C
          NA=1
          NB=2
          K=0
          N=2
          NP=3
          IP=7
          IS=6
          RL=1
          R0=1
          NPT=9
          LLOP=9
          A(1)=.7
          B(1)=1.
          B(2)=.8
99        WRITE (6,250)
250       FORMAT (1H1)
C
          WRITE (9,300)
300       FORMAT (23H TYPE R0, B(1) AND B(2))
          ICNTRL=1
          R0=RTTFF(ICNTRL)
          B(1)=RTTFF(ICNTRL)
          B(2)=RTTFF(ICNTRL)
          A1=A(1)/R0
          B1=B(1)/R0
          B2=B(2)/(R0**2)
C
          IF (B1-B2) 1,1,2
C
1         S(2,2)=(B2**2-B1**2)/((A1*B1-B2)**2)
          S(1,2)=-A1*S(2,2)
          S(2,1)=S(1,2)
          S(1,1)=1.+A1**2*S(2,2)
          GO TO 3
2         S(1,1)=1.
          S(1,2)=0.
          S(2,1)=0.
          S(2,2)=0.
C
3         RS=S(1,1)*B1*B1+2.*S(1,2)*B1*B2+S(2,2)*B2*B2
          RS=(S(1,1)*B1+S(1,2)*B2)/RS
          AL(1)=-A1*RS
          AL(2)=RS
C
          WRITE (6,200) R0
200       FORMAT (15H0TEST OF STURE2,4H R0=,F8.4)
          WRITE (6,201)
201       FORMAT (25H TRUE VALUE OF PARAMETERS)
          WRITE (6,202)
202       FORMAT (16H A AND B VECTORS)
          WRITE (6,100) (A(I),I=1,NA)
          WRITE (6,100) (B(I),I=1,NB)
          WRITE (6,203)
```

```
203   FORMAT (9H S-MATRIX)
      DO 10 I=1,N
10    WRITE (6,100) (S(I,J),J=1,N)
      WRITE (6,204)
204   FORMAT (9H L-VECTOR)
      WRITE (6,100) (AL(I),I=1,N)
C     GENERATE INPUTS AND OUTPUTS
      UM(1)=-1.
      UM(2)= 1.
      UM(3)= 1.
      UM(4)= 1.
      UM(5)=-1.
      UM(6)= 1.
      UM(7)=-1.
      UM(8)=-1.
      UM(9)= 1.
      UM(10)=1.
      YM(1)=0.
      YM(2)=0.
      UC(1)=0.
      UC(2)=0.
      DO 20 I=2,NPT
20    YM(I+1)=-A(1)*YM(I)+B(1)*UM(I)+B(2)*UM(I-1)
C     INITIALIZE
      Y(1)=YM(3)
      Y(2)=YM(2)
      U(1)=UM(3)
      U(2)=UM(2)
      U(3)=UM(1)
C
      DO 30 I=1,NP
      DO 30 J=1,NP
30    P(I,J)=0.
      DO 31 I=1,NP
      TH(I)=1.
31    P(I,I)=1.E5
C     MAIN LOOP
      DO 50 L=4,LLOP
      DO 51 I=1,N
      DO 51 J=1,N
51    S0(I,J)=S(I,J)
      IOF=0
      CALL TIME10(IM,ISEC,IS10,IOF)
      CALL STURE2(Y,U,TH,P,S0,RL,R0,NA,NB,K,IP,IS)
      IOF=1
      UC(L)=U(1)
      U(1)=UM(L)
      Y(1)=YM(L)
C
      WRITE (6,210)
210   FORMAT(23H0PARAMETES ESTIMATES TH)
      WRITE (6,100) (TH(I),I=1,NP)
      WRITE (6,211)
211   FORMAT (18H COVARIANCE MATRIX)
      DO 40 I=1,NP
40    WRITE (6,100) (P(I,J),J=1,NP)
      WRITE (6,212)
212   FORMAT (18H VECTORS AS AND BS)
      WRITE (6,100) (AS(I),I=1,N)
      WRITE (6,100) (BS(I),I=1,N)
      WRITE (6,213)
213   FORMAT (13H STATE VECTOR)
      WRITE (6,100) (X(I),I=1,N)
      WRITE (6,240)
```

```
 240    FORMAT (25H SCALED VECTORS YS AND US)
        WRITE (6,100) (YS(I),I=1,NA)
        WRITE (6,100) (US(I),I=1,NB)
        WRITE (6,203)
        DO 42 I=1,N
 42     WRITE (6,100) (S0(I,J),J=1,N)
        WRITE (6,204)
        WRITE (6,100) (AL(I),I=1,N)
        WRITE (6,205)
 205    FORMAT (19H CONTROL VARIABLE U)
        WRITE (6,100) (U(I),I=1,3)
        WRITE (6,206)
 206    FORMAT (15H PROCESS OUTPUT)
        IS1=60*IM+ISEC
        TIM=FLOAT(IS1)+.1*FLOAT(IS10)
        WRITE (6,100) (Y(I),I=1,2)
        WRITE (6,214) TIM
 50     CONTINUE
        WRITE (6,207)
 207    FORMAT (18H PROCESS INPUTS UM)
        WRITE (6,100) (UM(I),I=1,NPT)
        WRITE (6,208)
 208    FORMAT (19H PROCESS OUTPUTS YM)
        WRITE (6,100) (YM(I),I=1,NPT)
        WRITE (6,209)
 209    FORMAT (30H COMPUTED CONTROL VARIABLES UC)
        WRITE (6,100) (UC(I),I=1,NPT)
 100    FORMAT (5E12.4)
 214    FORMAT (15H COMPUTING TIME,F5.1,4H SEC)
 C
 C      COMPUTE CORRECT VALUES OF US
 C
        IF(B1-B2) 70,71,71
 71     DO 72 I=4,NPT
        SLAS=(A(1)*B(2)/B(1))*UM(I-2)
 72     UCC(I)=-(A(1)**2/B(1))*YM(I-1)-(B(2)-A(1)*B(1))/B(1)*UM(I-1)+SLA
        WRITE (6,230)
 230    FORMAT (36H CORRECT VALUES OF CONTROL VARIABLES)
        WRITE (6,100) (UCC(I),I=1,NPT)
        GO TO 99
 70     BET=B(1)/B(2)
        F1=BET-A(1)
        F2=A(1)*(A(1)-BET)/(1.-A(1)*BET)
        G0=-A(1)*F2
        DO 74 I=4,NPT
 74     UCC(I)=-G0/B(2)*YM(I-1)-F1*UM(I-1)-F2*UM(I-2)
        WRITE (6,230)
        WRITE (6,100) (UCC(I),I=1,NPT)
        GO TO 99
        END
```

Example 8.1

The following example was used as a test example.

$$y(t) + a\,y(t-1) = b_1u(t-1) + b_2u(t-2)$$

The control strategy is

$$u(t) = -\frac{a^2}{b_1}\,y(t-1) - \frac{b_2 - ab_1}{b_1}\,u(t-1) + \frac{ab_2}{b_1}\,u(t-2)$$

The numerical values used are

$a = 0.7$, $b_1 = 1$ and $b_2 = 0.8$

They give the control law

$$u_c(t) = -0.49y_m(t-1) - 0.1u_m(t-1) + 0.56u_m(t-2)$$

Introducing the following values for y_m and u_m

t	$y_m(t)$	$u_m(t)$
4		1
5	0.6380	-1
6	0.6466	1
7	0.6526	-1
8	0.6568	-1

we get

$$u_c(6) = 0.3474$$
$$u_c(7) = -0.3432$$
$$u_c(8) = 0.3502$$
$$u_c(9) = -0.1382$$

As is shown in the enclosed printout, List 10, the parameter estimates are correct after 6 steps and the computed actions are also correct. It is also shown that the results are invariant with respect to r_o unless r_o is chosen so small that the polynomial $\tilde{B}$ has a zero outside $|z| = r_o$. See List 11.

List 10. PRINTOUT FROM TSTU2 FOR RO = 1, B(1) = 1, B(2) = 0.8

```
TEST OF STURE2 RO=  1.0000
TRUE VALUE OF PARAMETERS
A AND B VECTORS
 0.7000E+00
 0.1000E+01  0.8000E+00
S-MATRIX
 0.1000E+01  0.0000E+00
 0.0000E+00  0.0000E+00
L-VECTOR
-0.7000E+00  0.1000E+01

PARAMETES ESTIMATES TH
 0.1000E+01  0.1100E+01  0.9000E+00
COVARIANCE MATRIX
 0.1000E+06  0.0000E+00  0.0000E+00
 0.0000E+00  0.5000E+05  0.5000E+05
 0.0000E+00  0.5000E+05  0.5000E+05
VECTORS AS AND BS
 0.1000E+01  0.0000E+00
 0.1100E+01  0.9000E+00
STATE VECTOR
 0.1800E+01  0.9000E+00
SCALED VECTORS YS AND US
 0.2000E+00
 0.1000E+01  0.1000E+01
S-MATRIX
 0.1000E+01  0.3325E-08
 0.3325E-08 -0.3325E-08
L-VECTOR
-0.9091E+00  0.9091E+00
CONTROL VARIABLE U
 0.1000E+01  0.1000E+01  0.1000E+01
PROCESS OUTPUT
 0.1660E+01  0.2000E+00
COMPUTING TIME  0.2 SEC

PARAMETES ESTIMATES TH
 0.1014E+01  0.1031E+01  0.8314E+00
COVARIANCE MATRIX
 0.9804E+05  0.9804E+04  0.9804E+04
 0.9804E+04  0.9809E+03  0.9804E+03
 0.9804E+04  0.9804E+03  0.9809E+03
VECTORS AS AND BS
 0.1014E+01  0.0000E+00
 0.1031E+01  0.8314E+00
STATE VECTOR
 0.1800E+00  0.8314E+00
SCALED VECTORS YS AND US
 0.1660E+01
 0.1000E+01  0.1000E+01
S-MATRIX
 0.1000E+01  0.1432E-07
 0.1551E-07 -0.1412E-07
L-VECTOR
-0.9829E+00  0.9696E+00
CONTROL VARIABLE U
-0.1000E+01  0.1000E+01  0.1000E+01
PROCESS OUTPUT
 0.6380E+00  0.1660E+01
COMPUTING TIME  0.2 SEC

PARAMETES ESTIMATES TH
```

12 and 13.

```
 0.7000E+00  0.1000E+01  0.8000E+00
COVARIANCE MATRIX
 0.9384E+00  0.4364E+00  0.4364E+00
 0.4364E+00  0.5780E+00  0.7799E-01
 0.4364E+00  0.7799E-01  0.5780E+00
VECTORS AS AND BS
 0.7000E+00  0.0000E+00
 0.1000E+01  0.8000E+00
STATE VECTOR
-0.6466E+00 -0.8000E+00
SCALED VECTORS YS AND US
 0.6380E+00
-0.1000E+01  0.1000E+01
S-MATRIX
 0.1000E+01 -0.7552E-08
-0.1352E-07  0.1079E-07
L-VECTOR
-0.7000E+00  0.1000E+01
CONTROL VARIABLE U
 0.1000E+01 -0.1000E+01  0.1000E+01
PROCESS OUTPUT
-0.6466E+00  0.6380E+00
COMPUTING TIME  0.2 SEC

PARAMETES ESTIMATES TH
 0.7000E+00  0.1000E+01  0.8000E+00
COVARIANCE MATRIX
 0.7879E+00  0.2408E+00  0.4921E+00
 0.2408E+00  0.3236E+00  0.1504E+00
 0.4921E+00  0.1504E+00  0.5574E+00
VECTORS AS AND BS
 0.7000E+00  0.0000E+00
 0.1000E+01  0.8000E+00
STATE VECTOR
 0.6526E+00  0.8000E+00
SCALED VECTORS YS AND US
-0.6466E+00
 0.1000E+01 -0.1000E+01
S-MATRIX
 0.1000E+01 -0.7552E-08
-0.1352E-07  0.1079E-07
L-VECTOR
-0.7000E+00  0.1000E+01
CONTROL VARIABLE U
-0.1000E+01  0.1000E+01 -0.1000E+01
PROCESS OUTPUT
 0.6526E+00 -0.6466E+00
COMPUTING TIME  0.2 SEC

PARAMETES ESTIMATES TH
 0.7000E+00  0.1000E+01  0.8000E+00
COVARIANCE MATRIX
 0.7459E+00  0.1872E+00  0.5066E+00
 0.1872E+00  0.2553E+00  0.1688E+00
 0.5066E+00  0.1688E+00  0.5524E+00
VECTORS AS AND BS
 0.7000E+00  0.0000E+00
 0.1000E+01  0.8000E+00
STATE VECTOR
-0.6568E+00 -0.8000E+00
SCALED VECTORS YS AND US
 0.6526E+00
-0.1000E+01  0.1000E+01
S-MATRIX
```

```
 0.1000E+01  0.0000E+00
 0.0000E+00  0.0000E+00
L-VECTOR
-0.7000E+00  0.1000E+01
CONTROL VARIABLE U
-0.1000E+01 -0.1000E+01  0.1000E+01
PROCESS OUTPUT
-0.6568E+00  0.6526E+00
COMPUTING TIME  0.1 SEC

PARAMETES ESTIMATES TH
 0.7000E+00  0.1000E+01  0.8000E+00
COVARIANCE MATRIX
 0.7255E+00  0.1617E+00  0.5131E+00
 0.1617E+00  0.2236E+00  0.1769E+00
 0.5131E+00  0.1769E+00  0.5504E+00
VECTORS AS AND BS
 0.7000E+00  0.0000E+00
 0.1000E+01  0.8000E+00
STATE VECTOR
-0.1340E+01 -0.8000E+00
SCALED VECTORS YS AND US
-0.6568E+00
-0.1000E+01 -0.1000E+01
S-MATRIX
 0.1000E+01  0.3758E-08
 0.6726E-08 -0.5368E-08
L-VECTOR
-0.7000E+00  0.1000E+01
CONTROL VARIABLE U
 0.1000E+01 -0.1000E+01 -0.1000E+01
PROCESS OUTPUT
-0.1340E+01 -0.6568E+00
COMPUTING TIME  0.2 SEC
PROCESS INPUTS UM
-0.1000E+01  0.1000E+01  0.1000E+01  0.1000E+01 -0.1000E+01
 0.1000E+01 -0.1000E+01 -0.1000E+01  0.1000E+01
PROCESS OUTPUTS YM
 0.0000E+00  0.0000E+00  0.2000E+00  0.1660E+01  0.6380E+00
-0.6466E+00  0.6526E+00 -0.6568E+00 -0.1340E+01
COMPUTED CONTROL VARIABLES UC
 0.0000E+00  0.0000E+00  0.0000E+00  0.8182E+00 -0.6292E+00
 0.3474E+00 -0.3432E+00  0.3402E+00 -0.1382E+00
CORRECT VALUES OF CONTROL VARIABLES
 0.0000E+00  0.0000E+00  0.0000E+00  0.3620E+00 -0.3534E+00
 0.3474E+00 -0.3432E+00  0.3402E+00 -0.1382E+00
```

OK

List 11. PRINTOUT FROM TSTU2 FOR RO = 0.9, B(1) = 1, B(2) = 0.8.

```
TEST OF STURE2 RO=  0.9000
TRUE VALUE OF PARAMETERS
A AND B VECTORS
 0.7000E+00
 0.1000E+01  0.8000E+00
S-MATRIX
 0.1000E+01  0.0000E+00
 0.0000E+00  0.0000E+00
L-VECTOR
-0.7000E+00  0.9000E+00

PARAMETES ESTIMATES TH
 0.1000E+01  0.1100E+01  0.9000E+00
COVARIANCE MATRIX
 0.1000E+06  0.0000E+00  0.0000E+00
 0.0000E+00  0.5000E+05  0.5000E+05
 0.0000E+00  0.5000E+05  0.5000E+05
VECTORS AS AND BS
 0.1111E+01  0.0000E+00
 0.1222E+01  0.1111E+01
STATE VECTOR
 0.1800E+01  0.1000E+01
SCALED VECTORS YS AND US
 0.1800E+00
 0.9000E+00  0.8100E+00
S-MATRIX
 0.1000E+01  0.6631E-08
 0.1584E-07 -0.5968E-08
L-VECTOR
-0.9091E+00  0.8182E+00
CONTROL VARIABLE U
 0.1000E+01  0.1000E+01  0.1000E+01
PROCESS OUTPUT
 0.1660E+01  0.2000E+00
COMPUTING TIME  0.2 SEC

PARAMETES ESTIMATES TH
 0.1014E+01  0.1031E+01  0.8314E+00
COVARIANCE MATRIX
 0.9804E+05  0.9804E+04  0.9804E+04
 0.9804E+04  0.9809E+03  0.9804E+03
 0.9804E+04  0.9804E+03  0.9809E+03
VECTORS AS AND BS
 0.1126E+01  0.0000E+00
 0.1146E+01  0.1026E+01
STATE VECTOR
 0.1800E+00  0.9237E+00
SCALED VECTORS YS AND US
 0.1494E+01
 0.9000E+00  0.8100E+00
S-MATRIX
 0.1000E+01  0.4202E-07
 0.2944E-07 -0.3730E-07
L-VECTOR
-0.9829E+00  0.8726E+00
CONTROL VARIABLE U
-0.1000E+01  0.1000E+01  0.1000E+01
PROCESS OUTPUT
 0.6380E+00  0.1660E+01
COMPUTING TIME  0.2 SEC

PARAMETES ESTIMATES TH
```

```
 0.7000E+00  0.1000E+01  0.8000E+00
COVARIANCE MATRIX
 0.9384E+00  0.4364E+00  0.4364E+00
 0.4364E+00  0.5780E+00  0.7799E-01
 0.4364E+00  0.7799E-01  0.5780E+00
VECTORS AS AND BS
 0.7778E+00  0.0000E+00
 0.1111E+01  0.9877E+00
STATE VECTOR
-0.6466E+00 -0.8889E+00
SCALED VECTORS YS AND US
 0.5742E+00
-0.9000E+00  0.8100E+00
S-MATRIX
 0.1000E+01 -0.3236E-07
-0.4285E-07  0.4161E-07
L-VECTOR
-0.7000E+00  0.9000E+00
CONTROL VARIABLE U
 0.1000E+01 -0.1000E+01  0.1000E+01
PROCESS OUTPUT
-0.6466E+00  0.6380E+00
COMPUTING TIME  0.2 SEC

PARAMETES ESTIMATES TH
 0.7000E+00  0.1000E+01  0.8000E+00
COVARIANCE MATRIX
 0.7879E+00  0.2408E+00  0.4921E+00
 0.2408E+00  0.3236E+00  0.1504E+00
 0.4921E+00  0.1504E+00  0.5574E+00
VECTORS AS AND BS
 0.7778E+00  0.0000E+00
 0.1111E+01  0.9877E+00
STATE VECTOR
 0.6526E+00  0.8889E+00
SCALED VECTORS YS AND US
-0.5819E+00
 0.9000E+00 -0.8100E+00
S-MATRIX
 0.1000E+01  0.1215E-07
 0.1361E-07 -0.1562E-07
L-VECTOR
-0.7000E+00  0.9000E+00
CONTROL VARIABLE U
-0.1000E+01  0.1000E+01 -0.1000E+01
PROCESS OUTPUT
 0.6526E+00 -0.6466E+00
COMPUTING TIME  0.2 SEC

PARAMETES ESTIMATES TH
 0.7000E+00  0.1000E+01  0.8000E+00
COVARIANCE MATRIX
 0.7459E+00  0.1872E+00  0.5066E+00
 0.1872E+00  0.2553E+00  0.1688E+00
 0.5066E+00  0.1688E+00  0.5524E+00
VECTORS AS AND BS
 0.7778E+00  0.0000E+00
 0.1111E+01  0.9877E+00
STATE VECTOR
-0.6568E+00 -0.8889E+00
SCALED VECTORS YS AND US
 0.5874E+00
-0.9000E+00  0.8100E+00
S-MATRIX
```

```
 0.1000E+01 -0.2051E-07
-0.2510E-07  0.2636E-07
L-VECTOR
-0.7000E+00  0.9000E+00
CONTROL VARIABLE U
-0.1000E+01 -0.1000E+01  0.1000E+01
PROCESS OUTPUT
-0.6568E+00  0.6526E+00
COMPUTING TIME  0.2 SEC

PARAMETES ESTIMATES TH
 0.7000E+00  0.1000E+01  0.8000E+00
COVARIANCE MATRIX
 0.7255E+00  0.1617E+00  0.5131E+00
 0.1617E+00  0.2236E+00  0.1769E+00
 0.5131E+00  0.1769E+00  0.5504E+00
VECTORS AS AND BS
 0.7778E+00  0.0000E+00
 0.1111E+01  0.9877E+00
STATE VECTOR
-0.1340E+01 -0.8889E+00
SCALED VECTORS YS AND US
-0.5912E+00
-0.9000E+00 -0.8100E+00
S-MATRIX
 0.1000E+01  0.2703E-07
 0.2299E-07 -0.3475E-07
L-VECTOR
-0.7000E+00  0.9000E+00
CONTROL VARIABLE U
 0.1000E+01 -0.1000E+01 -0.1000E+01
PROCESS OUTPUT
-0.1340E+01 -0.6568E+00
COMPUTING TIME  0.2 SEC
PROCESS INPUTS UM
-0.1000E+01  0.1000E+01  0.1000E+01  0.1000E+01 -0.1000E+01
 0.1000E+01 -0.1000E+01 -0.1000E+01  0.1000E+01
PROCESS OUTPUTS YM
 0.0000E+00  0.0000E+00  0.2000E+00  0.1660E+01  0.6380E+00
-0.6466E+00  0.6526E+00 -0.6568E+00 -0.1340E+01
COMPUTED CONTROL VARIABLES UC
 0.0000E+00  0.0000E+00  0.0000E+00  0.6182E+00 -0.6292E+00
 0.3474E+00 -0.3432E+00  0.3402E+00 -0.1382E+00
CORRECT VALUES OF CONTROL VARIABLES
 0.0000E+00  0.0000E+00  0.0000E+00  0.3620E+00 -0.3534E+00
 0.3474E+00 -0.3432E+00  0.3402E+00 -0.1382E+00
```

ok

List 12. PRINTOUT FROM TSTU2 FOR RO = 1, B(1) = 1, B(2) = 0.8.


```
TEST OF STURE2 RO=  1.0000
TRUE VALUE OF PARAMETERS
A AND B VECTORS
 0.7000E+00
 0.1000E+01  0.2000E+01
S-MATRIX
 0.1870E+01 -0.1243E+01
-0.1243E+01  0.1775E+01
L-VECTOR
 0.1077E+00 -0.1538E+00

PARAMETES ESTIMATES TH
 0.1000E+01  0.5000E+00  0.1500E+01
COVARIANCE MATRIX
 0.1000E+06  0.0000E+00  0.0000E+00
 0.0000E+00  0.5000E+05  0.5000E+05
 0.0000E+00  0.5000E+05  0.5000E+05
VECTORS AS AND BS
 0.1000E+01  0.0000E+00
 0.5000E+00  0.1500E+01
STATE VECTOR
 0.3000E+01  0.1500E+01
SCALED VECTORS YS AND US
-0.1000E+01
 0.1000E+01  0.1000E+01
S-MATRIX
 0.3000E+01 -0.2000E+01
-0.2000E+01  0.2000E+01
L-VECTOR
 0.6667E+00 -0.6667E+00
CONTROL VARIABLE U
 0.1000E+01  0.1000E+01  0.1000E+01
PROCESS OUTPUT
 0.3700E+01 -0.1000E+01
COMPUTING TIME  0.2 SEC

PARAMETES ESTIMATES TH
 0.1233E+01  0.7333E+00  0.1733E+01
COVARIANCE MATRIX
 0.6667E+05 -0.3333E+05 -0.3333E+05
-0.3333E+05  0.1667E+05  0.1667E+05
-0.3333E+05  0.1667E+05  0.1667E+05
VECTORS AS AND BS
 0.1233E+01  0.0000E+00
 0.7333E+00  0.1733E+01
STATE VECTOR
-0.2097E+01  0.1733E+01
SCALED VECTORS YS AND US
 0.3700E+01
 0.1000E+01  0.1000E+01
S-MATRIX
 0.6461E+01 -0.4428E+01
-0.4428E+01  0.3590E+01
L-VECTOR
 0.1206E+01 -0.9775E+00
CONTROL VARIABLE U
-0.1000E+01  0.1000E+01  0.1000E+01
PROCESS OUTPUT
 0.4100E+00  0.3700E+01
COMPUTING TIME  0.2 SEC

PARAMETES ESTIMATES TH
```

```
 0.7000E+00  0.1000E+01  0.2000E+01
COVARIANCE MATRIX
 0.8985E-01  0.6098E-01  0.6098E-01
 0.6098E-01  0.4154E+00 -0.8455E-01
 0.6098E-01 -0.8455E-01  0.4154E+00
VECTORS AS AND BS
 0.7000E+00  0.0000E+00
 0.1000E+01  0.2000E+01
STATE VECTOR
 0.7130E+00 -0.2000E+01
SCALED VECTORS YS AND US
 0.4100E+00
-0.1000E+01  0.1000E+01
S-MATRIX
 0.1870E+01 -0.1243E+01
-0.1243E+01  0.1775E+01
L-VECTOR
 0.1077E+00 -0.1538E+00
CONTROL VARIABLE U
 0.1000E+01 -0.1000E+01  0.1000E+01
PROCESS OUTPUT
 0.7130E+00  0.4100E+00
COMPUTING TIME  0.2 SEC

PARAMETES ESTIMATES TH
 0.7000E+00  0.1000E+01  0.2000E+01
COVARIANCE MATRIX
 0.8917E-01  0.5138E-01  0.6966E-01
 0.5138E-01  0.2787E+00  0.3920E-01
 0.6966E-01  0.3920E-01  0.3035E+00
VECTORS AS AND BS
 0.7000E+00  0.0000E+00
 0.1000E+01  0.2000E+01
STATE VECTOR
-0.1499E+01  0.2000E+01
SCALED VECTORS YS AND US
 0.7130E+00
 0.1000E+01 -0.1000E+01
S-MATRIX
 0.1870E+01 -0.1243E+01
-0.1243E+01  0.1775E+01
L-VECTOR
 0.1077E+00 -0.1538E+00
CONTROL VARIABLE U
-0.1000E+01  0.1000E+01 -0.1000E+01
PROCESS OUTPUT
-0.1499E+01  0.7130E+00
COMPUTING TIME  0.2 SEC

PARAMETES ESTIMATES TH
 0.7000E+00  0.1000E+01  0.2000E+01
COVARIANCE MATRIX
 0.8492E-01  0.6192E-01  0.5335E-01
 0.6192E-01  0.2525E+00  0.7963E-01
 0.5335E-01  0.7963E-01  0.2409E+00
VECTORS AS AND BS
 0.7000E+00  0.0000E+00
 0.1000E+01  0.2000E+01
STATE VECTOR
 0.2049E+01 -0.2000E+01
SCALED VECTORS YS AND US
-0.1499E+01
-0.1000E+01  0.1000E+01
S-MATRIX
```

```
 0.1870E+01 -0.1243E+01
-0.1243E+01  0.1775E+01
L-VECTOR
 0.1077E+00 -0.1538E+00
CONTROL VARIABLE U
-0.1000E+01 -0.1000E+01  0.1000E+01
PROCESS OUTPUT
 0.2049E+01 -0.1499E+01
COMPUTING TIME  0.1 SEC

PARAMETES ESTIMATES TH
 0.7000E+00  0.1000E+01  0.2000E+01
COVARIANCE MATRIX
 0.7552E-01  0.6826E-01  0.3424E-01
 0.6826E-01  0.2483E+00  0.9251E-01
 0.3424E-01  0.9251E-01  0.2021E+00
VECTORS AS AND BS
 0.7000E+00  0.0000E+00
 0.1000E+01  0.2000E+01
STATE VECTOR
-0.4435E+01 -0.2000E+01
SCALED VECTORS YS AND US
 0.2049E+01
-0.1000E+01 -0.1000E+01
S-MATRIX
 0.1870E+01 -0.1243E+01
-0.1243E+01  0.1775E+01
L-VECTOR
 0.1077E+00 -0.1538E+00
CONTROL VARIABLE U
 0.1000E+01 -0.1000E+01 -0.1000E+01
PROCESS OUTPUT
-0.4435E+01  0.2049E+01
COMPUTING TIME  0.2 SEC
PROCESS INPUTS UM
-0.1000E+01  0.1000E+01  0.1000E+01  0.1000E+01 -0.1000E+01
 0.1000E+01 -0.1000E+01 -0.1000E+01  0.1000E+01
PROCESS OUTPUTS YM
 0.0000E+00  0.0000E+00 -0.1000E+01  0.3700E+01  0.4100E+00
 0.7130E+00 -0.1499E+01  0.2049E+01 -0.4435E+01
COMPUTED CONTROL VARIABLES UC
 0.0000E+00  0.0000E+00  0.0000E+00 -0.1000E+01  0.4222E+01
-0.3845E+00  0.4691E+00 -0.5284E+00  0.1699E+00
CORRECT VALUES OF CONTROL VARIABLES
 0.0000E+00  0.0000E+00  0.0000E+00 -0.9077E-01  0.2635E+00
-0.3845E+00  0.4691E+00 -0.5284E+00  0.1699E+00
```

```
      SUBROUTINE STURE2(Y,U,TH,P,S,RL,R0,NA,NB,K,IP,IS)
C
C     SELFTUNING REGULATOR BASED ON LEAST SQUARES IDENTIFICATION
C     AND MINIMUM VARIANCE CONTROL
C     THE ALGORITHM IS BASED ON THE MODEL
C
C     Y(T)+A(1)*Y(T-1)+...+A(NA)*Y(T-NA)=
C     B(1)*U(T-K-1)+...+B(NB)*U(T-K-NB)          (*)
C
C     AT EACH STEP THE LEAST SQUARES ESTIMATES OF THE MODEL
C     PARAMETERS ARE COMPUTED, THE PROCESS INPUT U(T+1) TO BE
C     APPLIED TO THE PROCESS AT TIME T+1 IS THEN COMPUTED
C     FROM THE SOLUTION OF THE RICCATI EQUATION WHICH MINIMIZES
C     SUM Y(T)**2 UNDER THE CONSTRAINT THAT ALL POLES OF
C     THE CLOSED LOOP SYSTEM ARE WITHIN THE UNIT CIRCLE
C
C     WHEN APPLYING THE ALGORITM THE PROCESS OUTPUT IS THUS READ
C     AT TIME T, THE PROCESS INPUT U(T+1) TO BE APPLIED AT TIME T+1
C     IS THEN COMPUTED AT THE TIME INTERVAL (T,T+1)
C
C     AUTHOR KJ ASTROM 72-01-11
C
C     Y-VECTOR OF SCALED PROCESS INPUTS OF DIMENSION NA+1
C        AND ORGANIZED AS FOLLOWS
C
C        Y(1)=Y(T)
C        Y(2)=Y(T-1)                   RETURNED AS Y(T)
C        ...
C        Y(NA+1)=Y(T-NA)               RETURNED AS Y(T-NA+1)
C     U-VECTOR OF PROCESS OUTPUTS OF DIMENSION NB+K+1
C       ORGANIZED AS FOLLOWS
C       U(1)=U(T)                      RETURNED AS U(T+1)
C       U(2)=U(T-1)                    RETURNED AS U(T)
C       .
C       U(NB+K+1)=U(T-NB-K)            RETURNED AS U(T-NB-K+1)
C     TH-VECTOR OF ESTIMATED PARAMERS OF DIMENSION NA+NB
C        ORGANIZED AS FOLLOWS
C        TH(1)=AE(1)
C        TH(2)=AE(2)
C        .
C        TH(NA)=AE(NA)
C        TH(NA+1)=BE(1)
C        .
C        TH(NA+NB)=BE(NB)
C     P-COVARIANCE MATRIX OF THE PARAMETER ESTIMATES OF
C       ORDER (NA+NB)*(NA+NB)
C     S-SOLUTION TO THE RICCATI EQUATION WHICH GIVES THE
C       CONTROL LAW. THE INITIAL VALUE OF THE MATRIX SET
C       IN THE MAIN PROGRAM MUST BE POSITIVE DEFINITE
C       THE MATRIX S IS OF ORDER N*N (N=MAX(NA,NB+K))
C     RL-THE BASE OF THE EXPONENTIAL WEIGHTING FUNCTION
C     NA-NUMBER OF A-PARAMETERS  SEE (*)    MAX(NA)=8
C     NB-NUMBER OF B-PARAMETERS  SEE (*)    MAX(NB+K)=8
C     K-NUMBER OF TIMEDELAYS     SEE (*)    MAX(NB+K)=8
C     IP-DIMENSIONPARAMETER OF THE MATRIX P, MAX=32
C     IS- DIMENSION PARAMETER OF MATRIX S, MAX=8
C
C     USES DUM5-DUM8 OF COMMON /SLASK/
C
C
C     SUBROUTINES REQUIRED
C         RTLSID
C         SCAPRO
```

```
C          MOVE
C          CORI
C          NORM
C
      DIMENSION U(1),Y(1),TH(1),P(1,1),S(1,1)
      COMMON/SLASK/DUM(272),AS(8),BS(8),US(8),YS(8),X(8),AL(8)
     1,R(8,8),FI(64),DUM8(64)
C
      N=MAX0(NA,NB+K)
      NM1=N-1
      NAM1=NA-1
      NAP1=NA+1
      NBM1=NB-1
      NP1=N+1
      NP=NA+NB
C
C     SET FIXED PARAMETERS
C
      Q2=0.
      EPS=1.E-5
      NLOP=10
      IR=8
C
C     ORGANIZE DATA FOR IDENTIFICATION ROUTINE
C
      YIN=Y(1)
      DO 10 I=1,NA
   10 FI(I)=-Y(I+1)
      IF(NB) 12,12,11
   11 NS=2*NB
      CALL MOVE(U(K+2),FI(NA+1),NS)
C
   12 CALL RTLSID(TH,P,FI,YIN,NP,IP,RL,RES,DENOM)
C
C     SET PARAMETERS OF STATE MODEL
C
      CALL MOVE (TH(1),AS(1),NA+NA)
      IF (N-NA-1) 20,21,22
   22 AS(NA+1)=0.0
      NS=N-NA-1
      CALL MOVE (AS(NA+1),AS(NA+2),NS+NS)
      GO TO 20
   21 AS(NA+1)=0.0
   20 IF (K-1) 24,25,26
   26 BS(1)=0.0
      NS=K-1
      CALL MOVE (BS(1),BS(2),NS+NS)
      GO TO 24
   25 BS(1)=0.0
   24 CALL MOVE (TH(NA+1),BS(K+1),NB+NB)
      N1=NB+K+1
      IF (N-N1) 28,29,30
   30 BS(N1)=0.0
      NS=N-N1
      CALL MOVE(BS(N1),BS(N1+1),NS+NS)
      GO TO 28
   29 BS(N1)=0.0
C
C     SCALE SYSTEM PARAMETERS
C
   28 SF=1.
      DO 32 I=1,N
      SF=SF*R0
      AS(I)=AS(I)/SF
```

```
 32     BS(I)=BS(I)/SF
 C
 C      SCALE U AND Y
 C
        SF=1.
        DO 34 I=1,NA
        SF=SF*R0
 34     YS(I)=Y(I)*SF
        SF=1.
        NBK=NB+K
        DO 36 I=1,NBK
        SF=SF*R0
 36     US(I)=U(I)*SF
 C
 C      COMPUTE STATE VARIABLES
 C
        X(1)=YS(1)
        IF (NA-NB-K) 40,41,42
 40     IF (NA-2) 70,71,71
 71     DO 43 I=2,NA
        NS=NA-I+1
        NSB=N-I+1
 43     X(I)=-SCAPRO(AS(I),1,YS(2),1,NS)+SCAPRO(BS(I),1,US(2),1,NSB)
 70     DO 44 I=NAP1,N
        NS=N-I+1
 44     X(I)=SCAPRO(BS(I),1,US(2),1,NS)
        GO TO 49
 41     DO 45 I=2,N
        NS=N-I+1
 45     X(I)=-SCAPRO(AS(I),1,YS(2),1,NS)+SCAPRO(BS(I),1,US(2),1,NS)
        GO TO 49
 42     N1=NB+K
        DO 46 I=2,N1
        NSA=N-I+1
        NS=N1-I+1
 46     X(I)=-SCAPRO(AS(I),1,YS(2),1,NSA)+SCAPRO(BS(I),1,US(2),1,NS)
        N1=N1+1
        DO 47 I=N1,N
        NS=N-N1+1
 47     X(I)=-SCAPRO(AS(I),1,YS(2),1,NS)
 C
 C      PREDICT STATE VECTOR
 C
 49     RS=X(1)
        NM1=N-1
        DO 50 I=1,NM1
 50     X(I)=-AS(I)*RS+X(I+1)+BS(I)*US(1)
        X(N)=-AS(N)*RS+BS(N)*US(1)
 C
 C      COMPUTE CONTROL LAW
 C
        NLOOP=0
 60     NLOOP=NLOOP+1
        DO 62 I=1,N
        DO 62 J=1,N
 62     R(I,J)=S(I,J)
 C
        CALL CORI(AS,BS,Q2,S,AL,N,R3,IS)
 C
        DO 64 I=1,N
        DO 64 J=1,N
 64     R(I,J)=R(I,J)-S(I,J)
 C
 C      TEST IF ITERATION HAS CONVERGED
```

```
C
      CALL NORM(R,N,IR,RNORM)
      CALL NORM(S,N,IS,ANORM)
      IF (RNORM-EPS*SNORM) 66,66,65
   66 IND=-NLOOP
      GO TO 68
   65 IF (NLOOP-NLOP) 60,67,67
   67 IND=1
   68 CONTINUE
C
C
C     REORGANIZE DATA FOR NEXT STEP
C
      NS=2*NA
      CALL MOVE(Y(1),Y(2),-NS)
      NS=2*(NB+K)
      CALL MOVE(U(1),U(2),-NS)
C
C     COMPUTE CONTROL SIGNAL
C
      U(1)=-SCAPRO(AL(1),1,X(1),1,N)
C
C     THE CELL YS(1) IS NOW READY TO RECEIVE THE NEXT SCALED
C     PROCESS OUTPUT Y(T+1) AND THE CELL U(1) NOW CONTAINS THE
C     CONTROL SIGNAL U(T+1) TO BE USED AT TIME T+1
C
      RETURN
      END
```