



LUND UNIVERSITY

Implementation of a Self-Tuning Regulator for Non-Minimum Phase Systems on PDP 11/03

Gustavsson, Ivar

1980

Document Version:

Publisher's PDF, also known as Version of record

[Link to publication](#)

Citation for published version (APA):

Gustavsson, I. (1980). *Implementation of a Self-Tuning Regulator for Non-Minimum Phase Systems on PDP 11/03*. (Technical Reports TFRT-7209). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:

1

General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

CODEN: LUTFD2/(TFRT-7209)/1-093/(1980)

IMPLEMENTATION OF A SELF-TUNING REGULATOR FOR
NON-MINIMUM PHASE SYSTEMS ON PDP 11/03

IVAR GUSTAVSSON

DEPARTMENT OF AUTOMATIC CONTROL
LUND INSTITUTE OF TECHNOLOGY
OCTOBER 1980

<p>Organization LUND INSTITUTE OF TECHNOLOGY Department of Automatic Control Box 725 S-220 07 LUND 7 SWEDEN</p>	<p>Document name Internal report</p> <p>Date of issue October 1980</p> <p>CODEN: LUTFD2/(TFRT-7209)/1-093/(1980)</p>
<p>Author(s) Ivar Gustavsson</p>	<p>Sponsoring organization</p>
<p>Title and subtitle Implementation of a self-tuning regulator for non-minimum phase systems on PDP 11/03</p>	
<p>Abstract A program package implementing a selftuning regulator for non-minimum phase systems on PDP 11/03 is described. The intention is that the regulator should be easy to apply to different laboratory and industrial processes.</p>	
<p>Key words</p>	
<p>Classification system and/or index terms (if any)</p>	
<p>Supplementary bibliographical information</p>	
<p>ISSN and key title</p>	
<p>Recipient's notes</p>	
<p>Distribution by (name and address)</p>	

TABLE OF CONTENTS

	page
1. INTRODUCTION	1
2. THE ALGORITHM	2
3. HOW TO RUN THE PROGRAM	9
4. PROGRAM STRUCTURE	18
5. HOW TO CHANGE IN THE PROGRAMS	24
6. REFERENCES	27
APPENDIX A	
APPENDIX B	

1. INTRODUCTION

Self-tuning regulators have received a rapidly growing interest during the last decade, see e.g. Åström et al (1977). Not only theoretical results have been obtained but also many successful applications to different industrial processes have been reported. To facilitate simple testing of self-tuning regulators it was decided to develop software for PDP 11/03 computers implementing different types of self-tuning regulators. One type, the basic STURE1 algorithm, see Åström and Wittenmark (1973), has been implemented in PASCAL, Åström and Andersson (1979). In this program another type of self-tuning algorithm, STURE2, see Åström (1974), is implemented in FORTRAN with a very simple run-to-completion structure. The idea behind these implementations is that it should be possible to test self-tuning algorithms on different industrial and laboratory processes with a moderate amount of work and equipment. The only equipment needed is a PDP 11/03 with AI/AO/DI and with a floppy disc unit.

The algorithm is given in the next section. In Section 3 it is described how to run the program. Section 4 gives the program structure and in Section 5 it is described how changes in the program can be done.

The program has been run for a great number of test cases. The results have been compared with the results obtained in the simulation package described in Gustavsson (1978). These tests have been successful. However, the complexity of the program and the great number of possibilities to choose between have made it impossible to test all the variants.

2. THE ALGORITHM

The algorithm implemented in this program package is based on the hypothesis of separation of identification and control. The algorithm is principally similar to the so called STURE2 algorithm, see Åström (1974) and Åström and Wittenmark (1974). STURE2 uses a least squares estimation algorithm to estimate the process model. The extension of this algorithm to the case where the process model may include coloured noise is made. This algorithm as well as STURE2 can be used also for nonminimum phase systems. This algorithm was originally reported in Bengtsson and Egardt (1974). It was implemented also in the program package described in Gustavsson (1978). The estimation algorithm used is a recursive maximum likelihood scheme, see Söderström, Ljung and Gustavsson (1978).

The algorithm consists of the following steps, see Åström (1974):

- 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

The algorithm is based on the process model

$$y(t) + a_1 y(t-1) + \dots + a_n y(t-n) = b_1 u(t-k-1) + \dots + b_n u(t-k-n) + e(t) + c_1 e(t-1) + \dots + c_n e(t-n) \quad (2.1)$$

which can be written as

$$A(q^{-1}) y(t) = q^{-k} B(q^{-1}) u(t) + C(q^{-1}) e(t)$$

with suitably defined polynomials $A(q^{-1})$, $B(q^{-1})$, and $C(q^{-1})$ in the backward shift operator q^{-1} . It is assumed that all of the parameters are unknown but constant. The process is allowed to be nonminimum phase, i.e. the polynomial $q^n B(q^{-1})$ may have zeros outside the unit circle. In the implemented program it is actually

possible to allow for a different number of parameters in the polynomials, n_A , n_B , resp. n_C . In order to simplify the notations let n be $\max(n_A, n_B, n_C)$ and replace missing parameters by zeros.

The purpose of the control algorithm is to obtain a control strategy for the system (2.1). The criterion is chosen to be

$$\min E [y^2(t) + q_2 u^2(t)] \quad (2.2)$$

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 $t-1$ (or possibly up to time t).

Identification

The parameters of the model (2.1) are estimated using the recursive maximum likelihood method, see Söderström, Ljung and Gustavsson (1978). The algorithm is of the following form,

$$\begin{aligned} \hat{\theta}(t+1) &= \hat{\theta}(t) + P(t+1) \varphi(t+1) \varepsilon(t+1) \\ P(t+1) &= \left[P(t) - \frac{P(t) [\varphi(t+1)\varphi(t+1)^T + I] P(t)}{\lambda(t+1) + \varphi(t+1)^T P(t) \varphi(t+1)} + R_1 I \right] / \lambda(t+1) \\ \lambda(t+1) &= \lambda_0 \lambda(t) + 1 - \lambda_0 \end{aligned} \quad (2.3)$$

$\hat{\theta}(t)$ is a vector of parameter estimates based on data up to time t and $\varepsilon(t+1)$ is (an estimate of) the one step prediction error. For a definition of $\varphi(t)$, a vector containing filtered old input and output values, and a discussion of $\lambda(t)$, see Söderström, Ljung and Gustavsson (1978).

State vector

The state vector is defined as

$$\begin{aligned}
 x_1(t) &= y(t) - e(t) \\
 x_2(t) &= -a_2 y(t-1) - \dots - a_n y(t-n+1) + b_1 u(t-k) + \dots + \\
 &\quad + b_n u(t-n-k+1) + c_2 e(t-1) + \dots + c_n e(t-n+1) \\
 &\quad \vdots \\
 x_{k+1}(t) &= -a_{k+1} y(t-1) - \dots - a_n y(t-n+k) + b_1 u(t-1) + \dots + \\
 &\quad + b_n u(t-n) + c_{k+1} e(t-1) + \dots + c_n e(t-n+k) \\
 &\quad \vdots \\
 x_n(t) &= -a_n y(t-1) + b_{n-k} u(t-1) + \dots + b_n u(t-k-1) + c_n e(t-1) \\
 x_{n+1}(t) &= b_{n-k+1} u(t-1) + \dots + b_n u(t-k) \\
 &\quad \vdots \\
 x_{n+k}(t) &= b_n u(t-1).
 \end{aligned}$$

Prediction

The state variable $x(t)$ given by (2.4) 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 should be a function of data observed up to time $t-1$ only, it is necessary to predict the state, i.e. to compute $\hat{x}(t|t-1)$ and use the feedback law

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

The predicted state vector $\hat{x}(t|t-1)$ can be obtained using Kalman filtering theory

$$\hat{x}(t|t-1) = \hat{\phi}\hat{x}(t-1|t-2) + \hat{\Gamma}u(t-1) + \hat{K}[y(t-1) - \hat{C}\hat{x}(t-1|t-2)]. \quad (2.5)$$

The identification algorithm computes the residuals $\epsilon(t)$ and thus (2.5) can be replaced by

$$\hat{x}(t|t-1) = \hat{\phi}\hat{x}(t-1|t-2) + \hat{\Gamma}u(t-1) + \hat{K} \varepsilon(t)$$

The prediction depends on old parameter estimates and is thus rewritten as, see Bengtsson and Egardt (1974):

$$\hat{x}(t|t-1) = \begin{bmatrix} -\hat{a}_1 & \dots & \dots & -\hat{a}_n & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\hat{a}_n & & & 0 & & & \\ 0 & & & & & & \\ \vdots & & & & & & \\ \vdots & & & & & & \\ 0 & & & & & & \\ y(t-1) & & & & & & \\ \vdots & & & & & & \\ \vdots & & & & & & \\ y(t-n) & & & & & & \\ 0 & & & & & & \\ \vdots & & & & & & \\ \vdots & & & & & & \\ 0 & & & & & & \end{bmatrix} + \begin{bmatrix} u(t-1) & & & & & & \\ \vdots & & & & & & \\ \vdots & & & & & & \\ u(t-k-1) & & & & & & \\ \vdots & & & & & & \\ \vdots & & & & & & \\ u(t-k-n) & & & & & & \end{bmatrix} + \begin{bmatrix} \hat{c}_1 & \dots & \dots & \hat{c}_n & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hat{c}_n & & & 0 & & & \\ 0 & & & & & & \\ \vdots & & & & & & \\ \vdots & & & & & & \\ 0 & & & & & & \\ \varepsilon(t-1) & & & & & & \\ \vdots & & & & & & \\ \vdots & & & & & & \\ \varepsilon(t-n) & & & & & & \\ 0 & & & & & & \\ \vdots & & & & & & \\ \vdots & & & & & & \\ 0 & & & & & & \end{bmatrix}$$

Control Law

The control law is determined based on the assumption that the estimated parameters are correct. With the process model given by (2.4) and the criterion by (2.2) the control which minimizes the criterion is given by

$$u(t) = -L(t) x(t)$$

where

$$L(t) = \Gamma^T S(t) \phi / [q_2 + \Gamma^T S(t) \Gamma]$$

and S is the solution of the Riccati equation

$$S(t-1) = \phi^T S(t) [\phi - \Gamma L(t)] + Q_1$$

with the initial condition

$$S(N) = 0.$$

Q_1 is defined as $\text{diag}[1, 0, \dots, 0]$.

If the control law

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

is desired, the estimate $\hat{x}(t|t)$ must be computed from $\hat{x}(t|t-1)$.

This can be done observing that

$$\hat{x}(t|t) = \hat{x}(t|t-1) + K_1 \epsilon(t)$$

and thus

$$\hat{x}(t+1|t) = \phi \hat{x}(t|t-1) + \phi K_1 \epsilon(t)$$

Because of the equation (2.4)

$$\phi K_1 = (c_1^{-a_1}, c_2^{-a_2}, \dots, c_n^{-a_n}, 0, \dots, 0)^T$$

The elements of the vector K_1 can be solved from this system of equations provided $a_n \neq 0$.

Scaling

There is a possibility to scale the problem so that it is insured that the zeros of $q^n B(q^{-1})$ outside $|q| = r_0$ are reflected inside the circle with radius r_0 . The idea behind this scaling of system parameters, input, output and residual values is explained in Åström (1974).

Convergence

The convergence properties for these types of self-tuning regulators are still not fully understood. For STURE2 the following result is known, Åström and Wittenmark (1974). The parameters will converge to the minimum variance regulator, if

- o The parameters converge at all
- o The model has enough parameters
- o The penalty on the control signal is zero
- o $C(q^{-1}) = 1$ or the process is minimum phase
- o $k \leq 2$

3. HOW TO RUN THE PROGRAM

The program is implemented on a LSI/11 computer. It is started in the following way. Mount the system floppy disc on DX0 and the floppy disc No 160 on DX1. Then start up the computer system normally and start this self-tuning regulator program by

```
.RUN DX1:STURE2
```

Underlined characters are written by the operator. The program gives after a few seconds the following print-out on the system teletype unit (which normally is a alpha-numeric display but is called teletype in the sequel).

```
THIS PROGRAM REALIZES A SELF-TUNING REGULATOR OF STURE2-TYPE USING A RECURSIVE
MAXIMUM LIKELIHOOD ESTIMATOR
THE PROGRAM OPERATES IN TWO MODES: OPERATOR COMMUNICATION MODE
ESTIMATION/CONTROL MODE
THE OPERATOR COMMUNICATION PROGRAM IS OF COMMAND TYPE. AVAILABLE COMMANDS:
NA IVAL / KDEL IVAL / IMEAS IVAL / DU VAL / Q2 VAL / TH0 IVAL VAL
NB IVAL / BIAS IVAL / MAN VAL / UL VAL / RO VAL / PO IVAL VAL
NC IVAL / ITER IVAL / RLIMH VAL / UH VAL / RUN / R1 IVAL VAL
K IVAL / IRES IVAL / DELTA VAL / WD VAL / STOP / LIST [(DEV)] PAGNR
ILS IVAL / IOUT IVAL / TS VAL / W1 VAL / INIT /
WBIAS VAL/ RLIML VAL

THE COMMAND RUN TRANSFERS THE PROGRAM TO THE ESTIMATIONCONTROL MODE
THE PROGRAM USES TWO DIGITAL INPUTS, 0 AND 1:
0:FALSE TRANSFER BACK TO THE OPERATOR COMMUNICATION MODE
TRUE REMAINS IN ESTIMATION/CONTROL MODE
1:FALSE ESTIMATION ONLY
TRUE ESTIMATION AND CONTROL
THE PROCESS OUTPUT SIGNAL SHOULD BE CONNECTED TO ANALOG INPUT 0 [IMEAS]
THE CONTROL SIGNAL IS AVAILABLE AT ANALOG OUTPUT 0 [IOUT]
```

Fig. 3.1 - The listing obtained by LIST (LP) PAG1.

This information about the program can later be rewritten on the teletype using the command

```
>LIST PAG1
```

The program is a run-to-completion program with an operator communication mode of command type based on the program HPAC, Essebo (1978). HPAC in turn is a short version of INTRAC, Elmquist and Wieslander (1978). In some cases, however, the command structure is completed with a question. This is the case when the operator

tries to go from the operator communication mode to the estimation/control mode, or when he tries to stop the program. The transfer back to the operator communication mode is made by changing the digital input \emptyset . The digital input 1 controls whether the program only makes the estimation (i.e. with a fixed controller) or actually both estimates the process model and uses these estimates to update the control law parameters.

DIGITAL INPUT \emptyset :

FALSE If in estimation/control mode, transfer back to the operator communication mode
 TRUE Remain in estimation/control mode.

DIGITAL INPUT 1:

FALSE Estimation only (with fixed regulator)
 TRUE Estimation and tuning of the regulator.

The process output signal should be connected to an analog input channel, the number of which is determined by the command IMEAS IVAL; default \emptyset . The control signal from the program is available at an analog output channel, the number of which is determined by the command IOOUT; default \emptyset .

Command list

LIST Lists information about the regulator on teletype or lineprinter
 RUN Starts estimation and control and if digital input 1 is true also tuning of the regulator
 STOP Stops the program
 INIT Initialization of the "state" of the algorithm
 MAN Manually introduced control signal

NA	} <p>Defines the process model structure:</p> $y(t) + a_1 y(t-1) + \dots + a_{NA} y(t-NA) = b_1 u(t-k-1) + b_2 u(t-k-2) + \dots + b_{NB} u(t-k-NB) + \epsilon(t) + c_1 \epsilon(t-1) + \dots + c_{NC} \epsilon(t-NC)$
KDEL	
IBIAS	
ILS	
ITER	
IRES	0: The a priori residual estimate is used 1: The a posteriori residual estimate is used
IOUT	Analog output channel number for the control signal
IMEAS	Analog input channel number for the process output
TS	Sampling interval. If the computations turn out to be longer, the program is interrupted and control passed over to the operator communication node. The user can then choose a longer sampling interval and proceed
DU	0: Ordinary algorithm 1: The algorithm works on differences of $u(t)$ rather than on $u(t)$ itself
UL	Lower limit of the control signal
UH	Higher limit of the control signal
WØ	} The weighting factor is computed as $\lambda(t+1) = \lambda_0 \lambda(t) + 1 - \lambda_0$ with $WØ = \lambda(0)$ and $W1 = \lambda_0$.
W1	
RLIMH	The residuals in the estimation algorithm can be limited to be absolutely smaller than RLIMH
RLIML	If the residuals are absolutely smaller than RLIML then the parameter estimates and the covariance matrix P are not updated
DELTA	Used in the updating formula of P, see (2.3)

- Q2 Penalty on the control signal
- R \emptyset Possibility to restrict the poles of the closed loop system to be inside a circle with radius R \emptyset
- WBIAS The weighting factor for the bias estimate
- TH \emptyset Vector of initial parameter estimates
- P \emptyset Vector of initial diagonal elements of P, P(0) = P \emptyset *I
- R1 Vector used in the updating of P, see (2.3)

The syntax of these commands are described below. The list of commands and their syntax can be displayed by the command

>LIST PAG1

Detailed command description

>LIST [(DEV)] NAME

DEV: TT Output on teletype
 LP Output on lineprinter

NAME: PAG1 Displays command list, digital inputs used and requested analog input/output connections. Example: see fig. 3.1.

PAG2 Displays the basic parameters of the regulator. Example: see fig. 3.2.

PAG3 Displays some additional parameters of the regulator which in general are changed more infrequently. Example: see fig. 3.3.

PAG4 Displays the last time history of the input, output and residuals and the actual parameter estimates and control law parameters. Example: see fig. 3.4.

PAG5 Displays information that tries to prevent a transfer to control mode by mistake. Example: see fig. 3.5.

>RUN Starts the estimation and control algorithm. The actual operation is determined by the status of the digital inputs 0 and 1, see above. Before the algorithm starts

```

REGULATOR STRUCTURE WITH RML
-1 -K -1 -1 -1 -1 -NB
MODEL: A(Q)Y(T)=Q B(Q)U(T)+C(Q)E(T) B(Q)=B1*Q +...+BNB*Q
MODEL PARAMETERS: DIFFERENCES OF INPUT USED [DU] NO
ORDER OF A-,B-,C-POLYNOMIALS [NA],[NB],[NC] 1 1 0
PURE TIME DELAY [K] 0 ESTIMATE OF BIAS [BIAS] NO
SAMPLING INTERVAL [TS] 1.000 S W(0)=0.9900

IDENTIFICATION PARAMETERS:
WEIGHTING PROFILE [W0],[W1] W(T+1)=1.0000*W(T)+(1.-1.0000)
FORGETTING FACTOR FOR BIAS ESTIMATE [WBIAS] 1.000
INITIAL PARAMETER 0.0000 1.000
ESTIMATES [TH0]
INITIAL P-MATRIX 100.0 100.0
DIAGONAL [PD]

REGULATOR PARAMETERS (IN AUTO MODE):
CONTROL SIGNAL U(T) DEPENDS ON Y(T) [KDEL] NO (KDEL:1)
PENALTY ON CONTROL SIGNAL [Q2] 0.0000
CLOSED LOOP POLES INSIDE A CIRCLE WITH RADIUS [RO] 1.000
LOW AND HIGH LIMIT OF CONTROL SIGNAL [UL],[UH] -100.0 100.0

```

Fig. 3.2 - Listing obtained by LIST (LP) PAG2.

```

ADDITIONAL IDENTIFICATION PARAMETERS:
NO UPDATING IF RESIDUALS LESS THAN [RLIML]= 0.0000
PARAMETER NOISE 0.0000 0.0000
COVARIANCE [R1]
A PRIORI ESTIMATE OF RESIDUALS [IRES] YES (IRES:0)
UPPER LIMIT OF RESIDUALS [RLIMH] 1.0000E+04
DELTA IN P-UPDATING EQ. [DELTA] 0.0000
LS-ESTIMATE UNTIL STEP [ILS] 0

ADDITIONAL REGULATOR PARAMETERS (IN AUTO MODE):
MAXIMUM NUMBER OF ITERATIONS OF RICATTI EQ. [ITER] 10

```

Fig. 3.3 - Listing obtained by LIST (LP) PAG3.

R E G U L A T O R S T A T E							
TIME	OUTPUT	CONTROL SIGNAL	RESIDUALS	NR	PARAMETER ESTIMATES	REGULATOR PARAMETERS	REGULATOR STATES
T-0	-0.1006	0.5942	0.8405	1	-0.8951	0.6370	0.7448
T-1	-0.4351	-0.3828	-1.044	2	1.405		
T-2	0.4012	0.1683	0.6677				
T-3	-0.6727	0.2404	-0.2944				
T-4	0.7562	-0.7441	-0.4294				
T-5	-0.2720	0.9822	1.317				
T-6	-1.038	-0.4035	-1.686				
T-7	0.9147	-0.1456	0.6790				
T-8	0.9092	-0.4082	0.2458				
T-9	1.467	-0.4721	0.6937				

CURRENT WEIGHTING FACTOR: 1.000
FORGETTING FACTOR FOR BIAS ESTIMATE: 1.000

LOSS FUNCTION: 155.2

Fig. 3.4 - Listing obtained by LIST (LP) PAG4.

YOU ARE NOW LEAVING THE OPERATOR COMMUNICATION MODE.

CHECK THE DIGITAL INPUTS 0 AND 1 AND CORRECT IF NECESSARY.
THE DIGITAL INPUTS NOW HAVE THE MEANING:

0: GO BACK TO OPERATOR COMMUNICATION MODE
1: ESTIMATION ONLY

IF YOU HAVE TYPED THE COMMAND INIT BEFORE RUN THE REGULATOR IS INITIALIZED
AT YOUR DESIRE

IF INIT IS NOT TYPED BETWEEN TWO RUN COMMANDS THE REGULATOR WILL CONTINUE
FROM THE PREVIOUS STATE

IF YOU WANT TO CONTINUE TO THE ESTIMATION/CONTROL MODE, TYPE YES
IF YOU WANT TO GO BACK TO THE OPERATOR COMMUNICATION MODE, TYPE NO

Fig. 3.5 - Listing obtained by LIST (LP) PAG5.

PAG5, see above, is displayed and the operator should by his answer verify that he wants to proceed by typing YES. If he types NO he returns to the operator communication mode. This facility provides the extra checking of regulator parameters, digital input status etc. Notice that if digital input \emptyset is FALSE then the program transfers back to the operator communication mode immediately. The control signal remains unchanged but can be set to zero by the command INIT, see below.

>STOP

The program stops. The operator receives a question if this is actually desired. The answer NO returns the program to the operator communication mode.

>INIT

The state of the regulator is initialized. The following is done:

The matrix containing old measurements and residuals is filled with zeroes. The state vector for the regulator algorithm, the vector of control law parameters and the vector containing old filtered signal values for the recursive maximum likelihood algorithm are filled with zeroes. The vector of parameter estimates is initialized to the values given by the command TH \emptyset (default 0.). The matrix P is initialized as P \emptyset *I, where the values of P \emptyset are given by the command P \emptyset (default 100.). The weighting factor is initialized as the value given by the command W \emptyset (default 1.). The sum of squared process output values is initialized to zero. The control signal is put equal to zero and this value is set at the analog output.

>MAN NUMBR

The operator can regulate the process manually. NUMBR is any positive or negative real number or integer. Notice that the given value is added (or subtracted if negative) from the actual control signal and that the resulting control signal is checked versus the limits set

with the commands UL and UH. Also notice the scaling with 100., so that e.g. NUMBR = 100 means 1 V at the analog output.

All the parameter setting commands have the syntax

```
>PARNAM NUMBR
```

except those which refer to vectors (THØ, PØ, R1) which are set by

```
>PARNAM INTEG NUMBR
```

e.g.

```
>NA 2
```

```
>WØ 0.99
```

and

```
>THØ 2 1.
```

respectively.

NUMBR means as before any positive or negative real number or integer; INTEG means an integer. For the different parameters the Table 3.1 gives the limits checked and the default values. The value of INTEG in the command PARNAM INTEG NUMBR should be between 1 and 10.

Parameter	Limits checked		Default value	Comment
	Lower limit	Upper limit		
NA	0	10	0	NA+NB+NC+IBIAS<10 MAX(NA,NB,NC)+K<10
NB	0	10	0	
NC	0	10	0	
K	0	10	0	
KDEL	0	1	1	0 or 1
IBIAS	0	1	0	0 or 1
ILS	0	10000	0	
ITER	0	10000	10	
IRES	0	1	0	0 or 1
IOUT	0	15	0	
IMEAS	0	15	0	
TS	0	0	1.	
DU	0	0	0	0 or 1
UL			-100	Means -1 V at A/O
UH			+100	Means 1 V at A/O
WØ	0	1	1	
W1	0	1	1	
RLIMH	0		10000	
DELTA	0		0	
Q2	0		0	
RØ	0		1	
THØ			0	
PØ			100	
R1			0	
WBIAS			1.	
RLIML			0.	

Table 3.1 - Lower and upper limits of the different parameters, that are checked by the program and the default values of the parameters

4. PROGRAM STRUCTURE

The program is written in FORTRAN (except one routine in HPAC for string manipulations). It is of a very simple run-to-completion structure. When the computations in the regulator belonging to a certain sampling instant are finished, the program must lie in a wait loop until it is time for the next sampling. This wait function is also programmed in FORTRAN. If the computations are longer than the sampling interval, the regulator is stopped and the program will afterwards be in the operator communication mode.

The program is built up of a main program and a number of sub-routines. The main program has a very simple structure, revealing that the whole program is essentially composed of two parts,

- (i) Operator communication part, OPCOM
- (ii) Estimation/regulation part, REGUL

Besides these two parts there is the wait function. The block diagram of the main program, MAIN, is shown in fig. 4.1.

The structure of the complete program is shown in figs. 4.2-4.4. In fig. 4.2 the structure of the pure estimation/regulation part is shown. In fig. 4.3 the rest of the program which includes the subroutines special for this application is shown, mainly the operator communication program. In fig. 4.4 finally the structure of the calls of the subroutines in HPAC is displayed.

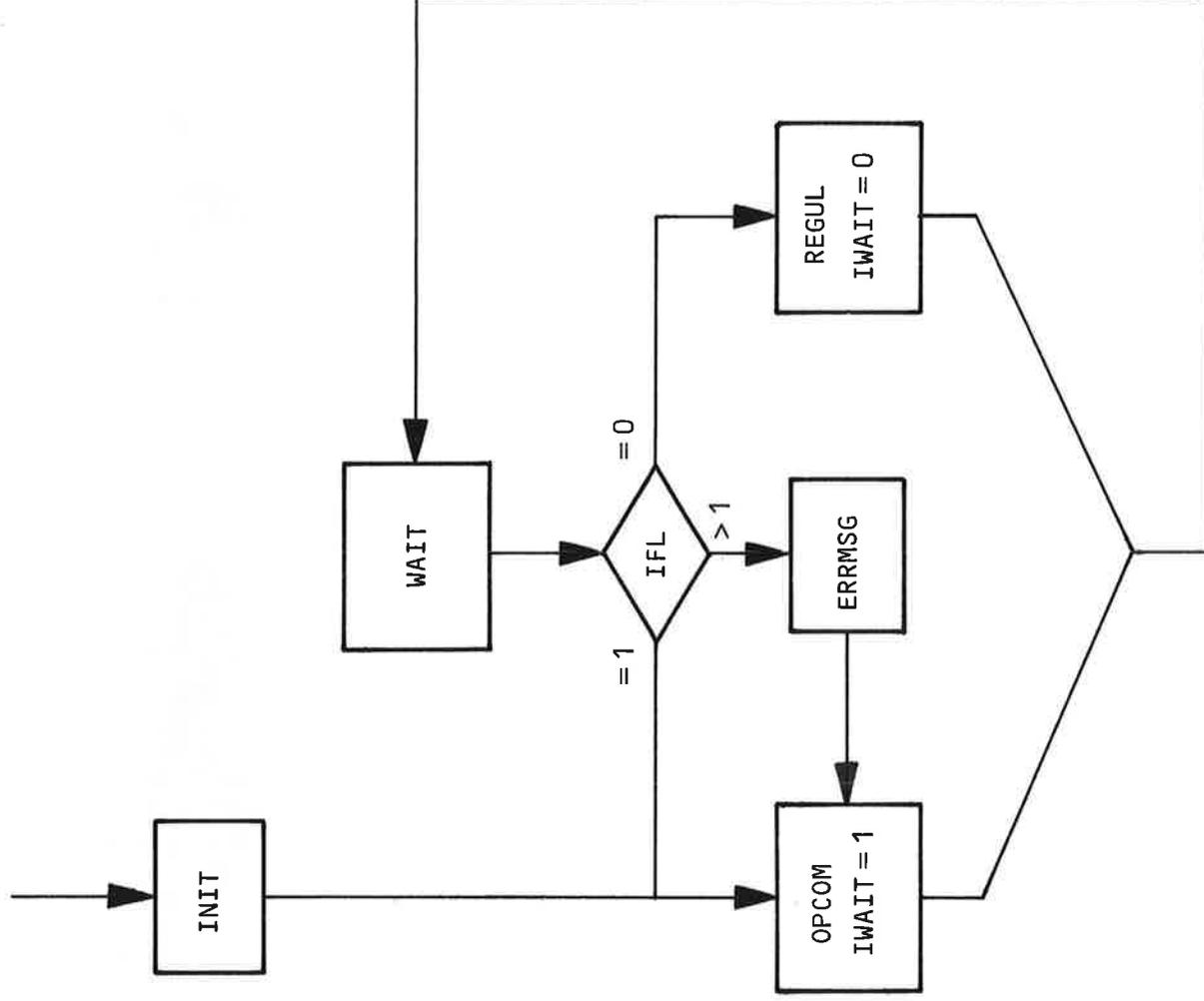


Fig. 4.1 - The value of the flag IFL is set by the subroutine WAIT. IFL = 0 or 1 is controlled by the digital input \emptyset . IFL > 0 indicates erroneous behaviour of WAIT, e.g. that the computations in REGUL could not be finished within one sampling interval. IWAIT = 1 initializes WAIT. INIT denotes the initialization part of MAIN. ERRMSG is a subroutine for error messages to the operator.

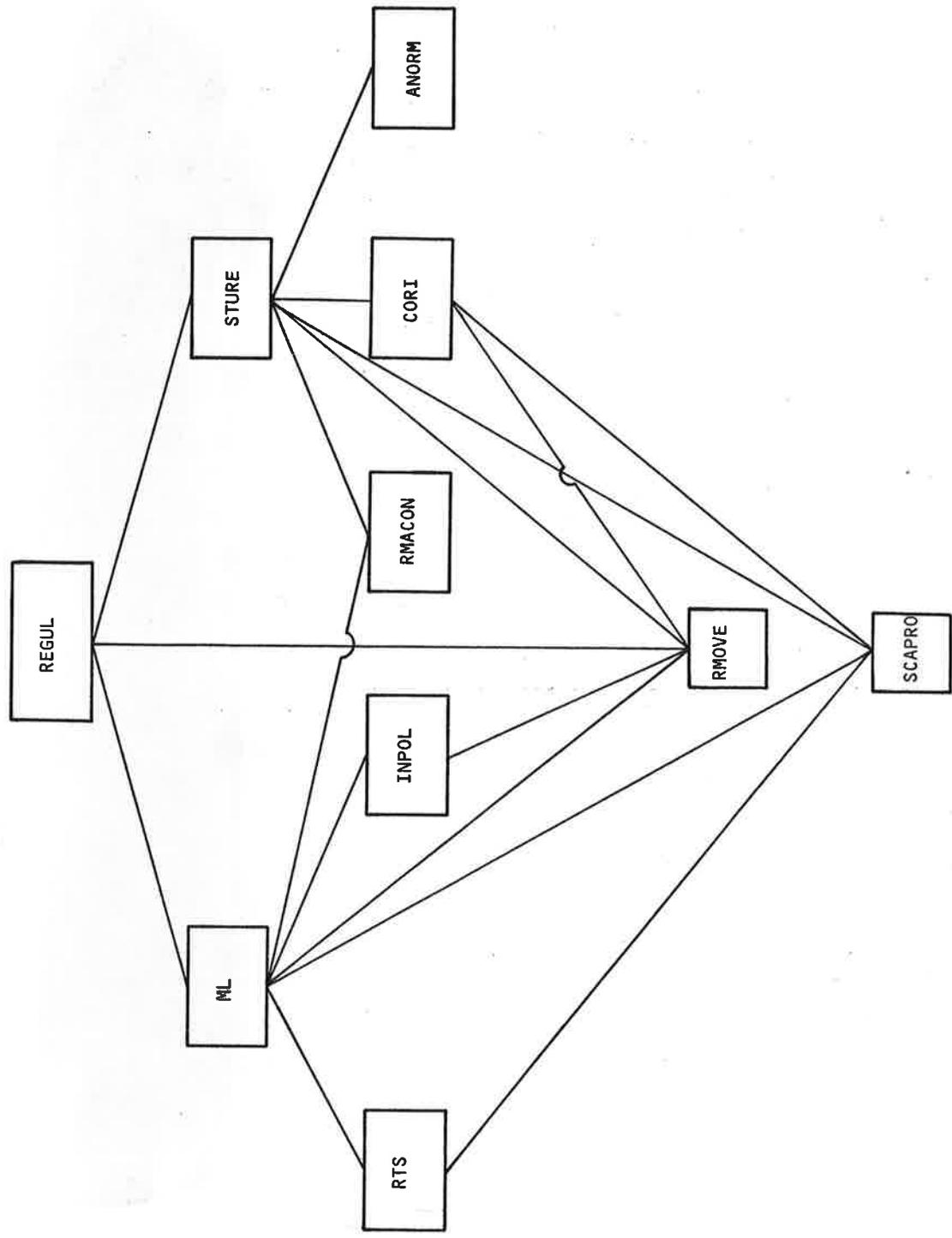


Fig. 4.2 - The structure of the estimation/regulation part of the program.

Fig. 4.3 - The main structure of the program. Notice that the pure estimation/regulation part is shown in Fig.4.2 and that the calls of the subroutines of HPAC are shown in Fig. 4.4. The specially marked box of REGUL is just indicates that this subroutine is also in the block diagram of Fig. 4.2.

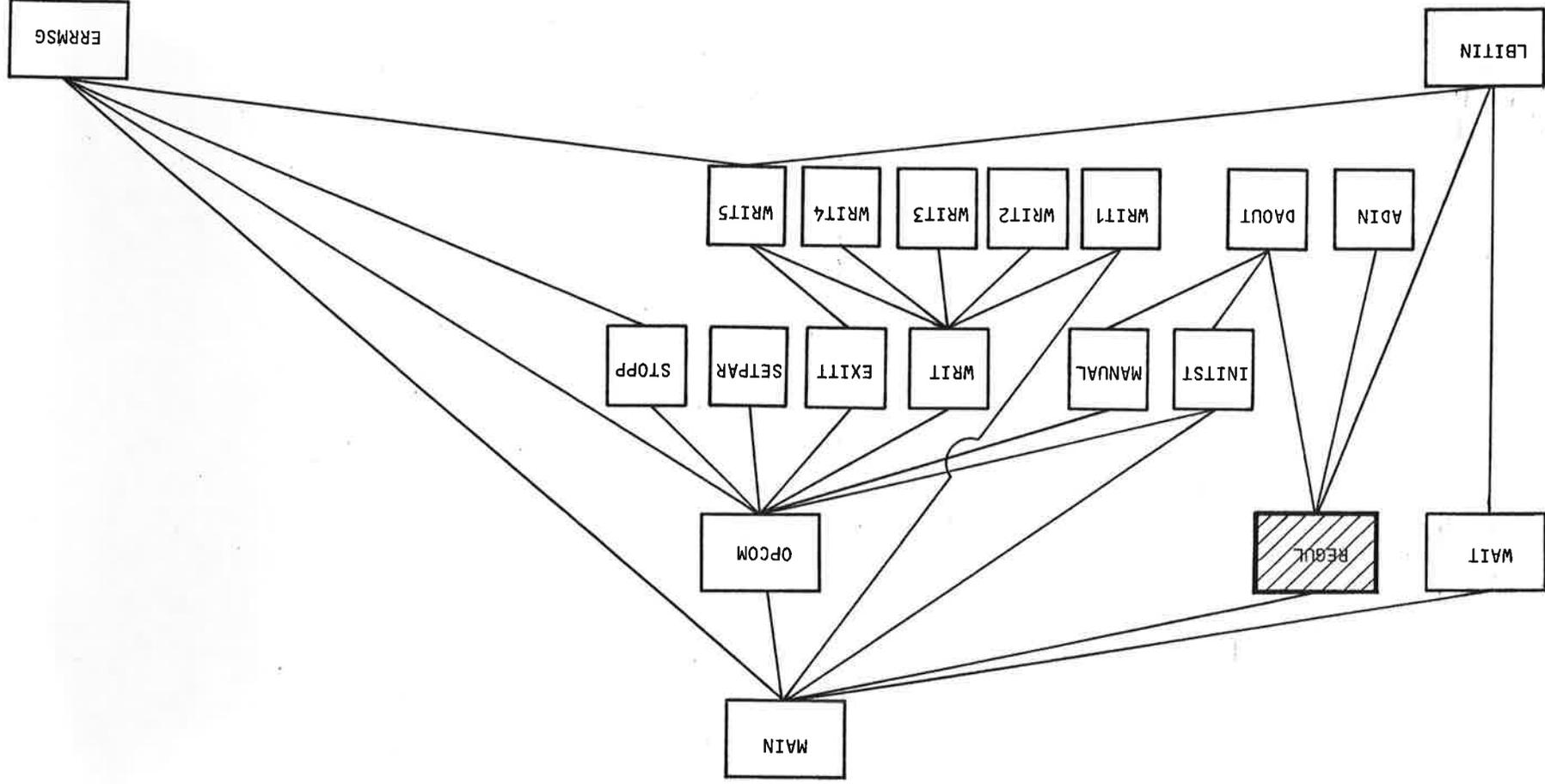
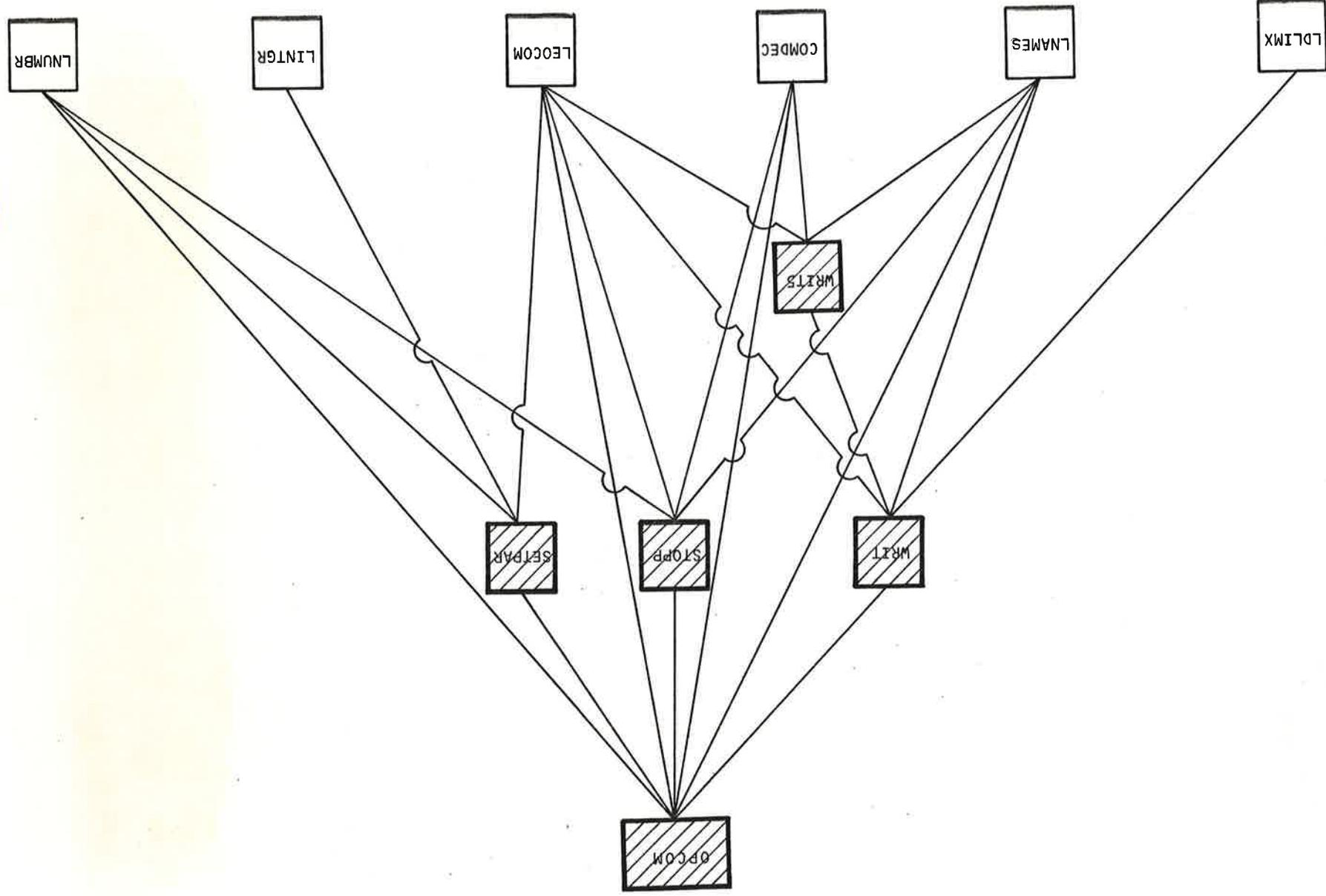


Fig. 4.4 - Structure for the calls of the subroutines in HPAC. The specially marked boxes indicates that the corresponding subroutines are also in the block diagram in Fig. 4.3.



All programs except those of HPAC are listed in Appendix A in alphabetical order. Below is a list of the programs with a short explanation and with references to Appendix A.

MAIN	Main program	A:13
OPCOM	Operator communication program	A:21
REGUL	Estimation/regulation program	A:23
WAIT	Wait function	A:39
INITST	Initialization of the state of the program	A:9
MANUAL	Manual control	A:15
WRIT	Organizes the output to teletype/lineprinter	A:42
WRIT1	} Output to teletype/lineprinter	A:44
WRIT2		A:46
WRIT3		A:48
WRIT4		A:49
WRIT5		A:50
EXITT	Run of regulator program (exit OPCOM)	A:8
SETPAR	Setting of parameters	A:31
STOPP	Stop of the program	A:34
ERRMSG	Error message routine	A:5
LBITIN	Checking of digital input	
DAIN	Analog input	
ADOUT	Analog output	
ML	Maximum likelihood estimation routine	A:16
STURE	The STURE algorithm	A:35
RTS	Updating of θ - and P-equations	A:27
INPOL	Checking stability for estimation routine	A:11
CORI	Computing the Ricatti equation solution	A:3
ANORM	Computes the norm of a matrix	A:1
RMACON	Computer dependent quantities are set	
RMOVE	Moves elements from one array to another	A:26
SCAPRO	Computes scalar products	A:30

The linking also uses the libraries REGLIB (to get ADIN and DAOUT) and COMLIB (to get HPAC), see fig. 5.2. The linking is done in the following way.

```
.R LINK
```

```
>DX1:STURE[-1],LP:=DX1:MAIN,ST2LIB,DX:SYSLIB,DX1:REGLIB,COMLIB/F 2
```

Notice also the use of SYSLIB to get e.g. the double integer computation routines used in WAIT. A lineprinter output from a linking is given in Appendix B. Also in Appendix B are given two subroutines, ADIN and DAOUT, to replace the ordinary subroutines for A/D and D/A in a program used for test purposes. The ADIN routine simulates a system

$$\begin{cases} x(t) + a_1x(t-1) + a_2x(t-2) = \\ = b_1u(t-1) + b_2u(t-2) + b_3u(t-3) + e(t) + c_1e(t-1) + c_2e(t-2) \\ y(t) = x(t) + y_{ref}(t) \end{cases}$$

where $y_{ref}(t)$ is a square wave command input with a period length to be chosen. This test program can be run by the command

```
.RUN DX1:STURE2.TST
```

when floppy disc nr 160 is mounted.

6. REFERENCES

- Åström, K.J. (1974):
A self-tuning regulator for nonminimum phase systems. TFRT-7411,
Dept of Automatic Control, Lund Institute of Technology, Sweden.
- Åström, K.J. and L. Andersson (1979):
Private communication.
- Åström, K.J., U. Borisson, L. Ljung, and B. Wittenmark (1977):
Theory and applications of self-tuning regulators. *Automatica* 13,
pp. 457-476.
- Åström, K.J. and B. Wittenmark (1973):
On self-tuning regulators. *Automatica* 9, pp. 185-199.
- Åström, K.J. and B. Wittenmark (1974):
Analysis of a self-tuning regulator for non-minimum phase systems.
Preprints IFAC Symposium on Stochastic Control, Budapest, Hungary.
- Bengtsson, B. and B. Egardt (1974):
Ett interaktivt programpaket för simulering av självinställande
regulatorer. TFRT-7141, Dept of Automatic Control, Lund Institute
of Technology, Sweden.
- Elmqvist, H. and J. Wieslander (1978):
INTRAC - a communication module for interactive programs. Language
manual. Dept of Automatic Control, Lund Institute of Technology,
Sweden. CODEN LUTFD2/(TFRT-3149)/-060/(1978).
- Essebo, T. (1978):
Private communication.

Gustavsson, I. (1978):

User's guide for a program package for simulation of self-tuning regulators. Dept of Automatic Control, Lund Institute of Technology, Sweden. CODEN LUTFD2/(TFRT-7149)/-076/(1978).

Söderström, T., L. Ljung and I. Gustavsson (1978):

A theoretical analysis of recursive identification methods. *Automatica* 14, pp. 231-244.

APPENDIX A

FORTRAN IV

V01C-03F+ THU 08-NOV-79 08:39:43

PAGE 001

```

C      NAME: ANORM          NUMBER:
C      -----
C
C      SUBTITLE: COMPUTES THE MINIMAX NORM OF A SQUARE MATRIX
C      -----
C
C      LANGUAGE: FORTRAN IV
C      -----
C
C      KEYWORDS: NORM,MATRIX
C      -----
C
C      IMPLEMENTOR: KRISTER MORTENSSON
C      -----
C
C      INSTITUTE:
C      -----
C      DEPARTMENT OF AUTOMATIC CONTROL
C      LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C      ACCEPTED:
C      -----
C
C      =====
C      PURPOSE
C      =====
C
C      COMPUTES THE MINIMAX NORM OF A SQUARE MATRIX
C
C      USAGE
C      =====
C
C      PROGRAM TYPE: FUNCTION
C      -----
C
C      ARGUMENTS:
C      -----
C      ANORM(A,N,IA)
C
C      ANORM - MINIMAXNORM OF A
C      A     - SQUARE MATRIX, SIZE(N,N), DIMENSIONED(IA,..), (I)
C      N     - ACTUAL DIMENSION OF THE MATRIX A, (I)
C      IA    - DECLARED FIRST DIMENSION OF A, (I)
C
C      METHOD
C      =====
C
C      ANORM = MIN( MAX( SUM( ABS(A(I,J)), I=1,N ), J=1,N )
C      MAX( SUM( ABS(A(I,J)), J=1,N ), I=1,N ) )
C
C      CHARACTERISTICS
C      =====
C

```



```

0001      SUBROUTINE CORI(A,B,Q2,S,AL,N,R3,IS,W1,W2)
C
C      THIS SUBROUTINE ITERATES THE RICATTI 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      REVISED I. GUSTAVSSON 77-03-14
C
C      A - VECTOR CONTAINING THE FIRST COLUMN OF THE MATRIX A IN
C      (*), I.E. A(I,1)=-A(I)
C      B - VECTOR B IN (*) AND (**)
C      Q2 - SCALAR Q2 IN (**)
C      S - SOLUTION OF THE RICATTI EQUATION
C      AL - VECTOR L IN (**)
C      N - ACTUAL ORDER OF THE SYSTEM
C      R3 - DENOMINATOR (Q2+BT*S*B)
C      IS - DIMENSION PARAMETER OF THE MATRIX S
C      W1,W2 - MATRIX WORK AREAS OF ORDER IS X IS
C
0002      DIMENSION A(1),B(1),AL(1),S(IS,1)
0003      DIMENSION W1(IS,1),W2(IS,1)
C
0004      DO 10 I=1,N
0005      W1(I,1)=SCAPRO(B(1),S(1,I),1,1,N)
0006      R3=SCAPRO(B(1),W1(1,1),1,1,N)+Q2
C
C      R3 NOW CONTAINS BT*S*B+Q2
C
0007      AL(1)=-SCAPRO(W1(1,1),A(1),1,1,N)/R3
0008      IF(N.EQ.1) GO TO 25
0010      DO 20 I=2,N
0011      I1=I-1
0012      AL(I)=W1(I1,1)/R3
C
C      COMPUTATION OF L=BT*S*A/(Q2+BT*S*B). COMPLETE RESULT
C      STORED IN VECTOR AL
C
0013      R=AL(1)
0014      DO 30 I=1,N
0015      W1(I,1)=-A(I)-B(I)*R
0016      IF(N.EQ.1) GO TO 55
0018      DO 50 J=2,N
0019      R=AL(J)
0020      DO 50 I=1,N
0021      R1=0.
0022      IF(I+1-J) 50,40,50
0023      R1=1.
0024      W1(I,J)=R1-B(I)*R
C

```

FORTRAN IV VD1C-03F+ THU 08-NOV-79 08:45:12

PAGE 002

```

C
C
0025      S1 NOW CONTAINS A-B*L
0026      DO 60 I=1,N
0027      W2(I,1)=-SCAPRO(A(1),S(I,1),1,IS,N)
0029      IF(N.EQ.1) GO TO 75
0030      NM1=N-1
0031      DO 70 J=1,NM1
0032      CALL RMOVE(S(1,J),W2(1,J+1),1,1,N)
C
C
0033      S2 NOW CONTAINS S*A
0034      DO 80 I=1,N
0035      DO 80 J=1,N
0036      S(I,J)=SCAPRO(W2(1,I),W1(1,J),1,1,N)
0037      S(1,1)=S(1,1)+1.
C
0038      RETURN
0039      END
```

```

0001      SUBROUTINE ERRMSG
C
C      PRINTS ERROR MESSAGES FOR COMMAND DECODER
C
C      AUTHOR TOMMY ESSEBO 1978-05-10
C      REVISED IVAR GUSTAVSSON 1979-08-20
C      DEPARTMENT OF AUTOMATIC CONTROL
C      LUND INSTITUTE OF TECHNOLOGY
C      LUND , SWEDEN
C
C      COMMON/ERRIND/IERR,ERRNAM(2)
C
C      COMMON/IODEVS/LKB,LTO
C
C      COMMON/CMBUFF/NCHC,CBUFF(20),IPCOM,DUM1(2),IDUM1(2)
C
C      DATA POINT/4H: /,SPACE/4H /
C
C      IF(IERR.GT.20) GO TO 200
C
C      -----
C
C      IERR=1 ... 20
C      SYNTAX ERRORS
C      BAD ARGUMENT WILL BE POINTED OUT
C
C      IERR1=IERR
C      GO TO (1,2,3,4,5,6,7,8,9),IERR1
C
C      1      WRITE(LTO,1001)
C      1001      FORMAT(' NAME EXPECTED' )
C      GO TO 100
C
C      2      WRITE(LTO,1002)
C      1002      FORMAT(' INTEGER EXPECTED' )
C      GO TO 100
C
C      3      WRITE(LTO,1003)
C      1003      FORMAT(' STRING QUOTE EXPECTED' )
C      GO TO 100
C
C      4      WRITE(LTO,1004)
C      1004      FORMAT(' NUMBER EXPECTED' )
C      GO TO 100
C
C      5      WRITE(LTO,1005)
C      1005      FORMAT(' . EXPECTED' )
C      GO TO 100
C
C      6      WRITE(LTO,1006)
C      1006      FORMAT(' END OF COMMAND LINE EXPECTED' )
C      GO TO 100
C
C      7      WRITE(LTO,1007)
C      1007      FORMAT(' ) EXPECTED' )

```

```

0030          GO TO 100
C
0031      8  WRITE(LTO,1008)
0032      1008 FORMAT(' ILLEGAL COMMAND')
0033          GO TO 100
C
0034      9  WRITE(LTO,1009)
0035      1009 FORMAT(' ILLEGAL NAME')
0036          GO TO 100
C
0037      100 NWD=(NCHC+3)/4
0038          WRITE(LTO,1100)(CBUFF(I),I=1,NWD)
0039      1100 FORMAT(1X,19A4)
0040          WRITE(LTO,1105)(SPACE,I=1,IPCOM),POINT
0041      1105 FORMAT(80A1)
0042          RETURN
C
0043      200 IF(IERR.GT.30) GO TO 300
C
C -----
C      IERR=21 ... 30
C      ERROR MESSAGE + ERRNAM WRITTEN
C
0045      IERR1=IERR-20
0046          GO TO (11,12),IERR1
C
0047      11  WRITE(LTO,1011)
0048      1011 FORMAT(' FILE NOT FOUND:')
0049          GO TO 250
C
0050      12  WRITE(LTO,1012) ERRNAM(1)
0051      1012 FORMAT(' SORRY, YOU ARE',1PG10.2,' S TOO LATE')
0052          RETURN
C
0053      250 WRITE(LTO,1100)ERRNAM
0054          RETURN
C
C -----
C      IERR=31 ...
C      ERROR MESSAGE PRINTED
C
0055      300 IERR1=IERR-30
0056          GO TO(21,22,23,24,25,26,27,28,29,30,31,32,33),IERR1
C
0057      21  WRITE(LTO,1021)
0058      1021 FORMAT(' NO FILE DEFINED')
0059          RETURN
C
0060      22  WRITE(LTO,1022)
0061      1022 FORMAT(' NO VARIABLE DEFINED')
0062          RETURN
C
0063      23  WRITE(LTO,1023)
0064      1023 FORMAT(' TOO MANY CHARACTERS IN TEXT STRING')

```

```
0065          RETURN
C
0066      24 WRITE(LTO,1024)
0067      1024 FORMAT(' ILLEGAL DEVICE')
0068          RETURN
C
0069      25 WRITE(LTO,1025)
0070      1025 FORMAT(' SYNTAX ERROR. SYNTAX: LIST [(DEV)] PAGNR, DEV=L P OR
          * TT, NR=1,2,3,4 OR 5')
0071          RETURN
C
0072      26 WRITE(LTO,1026)
0073      1026 FORMAT(' NONNEGATIVE INTEGER EXPECTED')
0074          RETURN
C
0075      27 WRITE(LTO,1027)
0076      1027 FORMAT(' NONNEGATIVE NUMBER EXPECTED')
0077          RETURN
C
0078      28 WRITE(LTO,1028)
0079      1028 FORMAT(' NUMBER BETWEEN 0. AND 1. EXPECTED')
0080          RETURN
C
0081      29 WRITE(LTO,1029)
0082      1029 FORMAT(' UPPER LIMIT FOR THIS PARAMETER EXCEEDED')
0083          RETURN
C
0084      30 WRITE(LTO,1030)
0085      1030 FORMAT(' TOO MANY PARAMETERS. NA+NB+NC+BIAS.LE.10 AND
          * MAX(NA,NB,NC)+K.LE.10')
0086          RETURN
C
0087      31 WRITE(LTO,1031)
0088      1031 FORMAT(' ERROR IN DOUBLE INTEGER MANIPULATION IN SUBROUTINE
          *WAIT')
0089          RETURN
C
0090      32 WRITE(LTO,1032)
0091      1032 FORMAT(' WRONG SIGN IN DOUBLE INTEGER MANIPULATION IN
          *SUBROUTINE WAIT')
0092          RETURN
C
0093      33 WRITE(LTO,1033)
0094      1033 FORMAT(' BT*S*B+Q2=0! SET A NON-ZERO INITIAL VALUE
          * TO A B-PARAMETER (OR Q2.NE.0.))')
0095          END
```

```

FORTRAN IV      VO1C-03F+  FRI 09-NOV-79 08:52:24      PAGE 001

0001      SUBROUTINE EXITT( IERR, IFL)
           C
           C EXECUTES THE COMMAND RUN, I.E. THE TRANSFER FROM THE OPERATOR
           C COMMUNICATION MODE TO THE ESTIMATION/REGULATION MODE
           C
0002      COMMON/REGPAR/IPAR(20),PAR(50)
0003      COMMON/IODEVS/LKB,LTO,LPA
0004      COMMON/STATE/S(3,11),T(10),ZZ(241)
           C
0005      DATA EPS/1.E-5/
           C
           C TEST IF ALL B-PARAMETERS ARE ZERO
           C
0006      IF(PAR(9).GT.EPS) GO TO 30
0008      NA=IPAR(8)
0009      NB=IPAR(9)
0010      IF(NB.EQ.0) GO TO 20
0012      N1=NA+1
0013      N2=NA+NB
0014      DO 10 I=N1,N2
0015      IF(ABS(T(I)).GT.EPS) GO TO 30
0017      CONTINUE
           C
           C INITIAL VALUES FOR STURE NOT ACCEPTABLE
           C
0018      IERR=43
0019      RETURN
           C
0020      IFL=0
0021      CALL WRITS(LTO,IFL)
           C
0022      RETURN
0023      END

```

FORTRAN IV VO1C-03F+ THU 08-NOV-79 08:55:17

PAGE 001

```

0001      SUBROUTINE INITST
C
C      INITIALIZES DATA AREAS, STATES AND VARIABLES OF THE REGULATOR,
C      AND SETS THE CONTROL SIGNAL TO ZERO
C
0002      COMMON/REGPAR/IPAR(20),PAR(50)
0003      COMMON/STATE/S(3,11),T(10),P(10,10),Z(20),WT,X(10),S1(10,10),
      *T1(10)
0004      COMMON/RES/RES,UR,AL(10),IND,UROLD,W
C
0005      DATA IP/10/
C
0006      IP1=IP+1
C
C      INITIALIZE THE DATA AREA CONTAINING PREVIOUS SIGNAL VALUES, S
C
0007      DO 10 I=1,IP1
0008      DO 10 J=1,3
0009      S(J,I)=0.
C
C      ZERO THE STATE VECTOR OF THE STATE MODEL USED BY STUREZ, X;
C      THE VECTOR OF CONTROL LAW PARAMETERS, AL; INITIALIZE THE SOLUTION
C      OF THE RICATTI EQUATION, S1, AS THE UNIT MATRIX; AND ZERO THE
C      COVARIANCE MATRIX IN THE ESTIMATION ALGORITHM, P
C
0010      DO 30 I=1,IP
0011      X(I)=0.
0012      AL(I)=0.
0013      DO 20 J=1,IP
0014      P(I,J)=0.
0015      DO 30 S1(I,J)=0.
0016      DO 30 S1(I,I)=1.
C
C      INITIALIZE THE PARAMETER ESTIMATE VECTOR, T, AND THE DIAGONAL OF
C      THE COVARIANCE MATRIX, P, TO THE VALUES GIVEN BY THE
C      DEFAULT VALUES OF THD AND PD RESP. OR TO THE VALUES CHANGED
C      BY THE COMMANDS THD INTEG NUMBR AND PD INTEG NUMBER RESP.
C
0017      DO 40 I=1,IP
0018      T(I)=PAR(10+I)
0019      T1(I)=T(I)
0020      DO 40 P(I,I)=PAR(20+I)
C
C      INITIALIZE STATE VECTOR USED FOR THE FILTERED DATA IN THE
C      ESTIMATION ALGORITHM
C
0021      DO 50 I=1,20
0022      Z(I)=0.
C
C      INITIALIZE THE WEIGHTING FACTOR TO THE DEFAULT VALUE OF WTI OR
C      TO THE VALUE CHANGED BY THE COMMAND WTI NUMBR
C
0023      WT=PAR(5)

```


FORTRAN IV

V01C-03F+ THU 08-NOV-79 08:44:45

PAGE 001

```
0001      FUNCTION SCAPRO(AVEC,BVEC,ISTEPA,ISTEPB,NTERMS)
          C
          C      SUBROUTINE REQUIRED
          C      NONE
          C
          C      COMPUTES THE SCALAR PRODUCT
          C
          C      AVEC  - FIRST VECTOR OPERAND
          C      BVEC  - SECOND VECTOR OPERAND
          C      ISTEPA - EVERY ISTEPA:TH ELEMENTS OF AVEC WILL BE USED
          C      ISTEPB - EVERY ISTEPB:TH ELEMENT OF BVEC WILL BE USED
          C      NTERMS - NUMBER OF TERMS IN THE SCALAR PRODUCT
          C
          C      DOUBLE PRECISION DP
          C
          C      DIMENSION AVEC(1),BVEC(1)
          C
          C      DP=0.0D+00
          C      JA=1
          C      JB=1
          C
          C      DO 200 J=1,NTERMS
          C      DP=DP+DBLE( AVEC(JA)*BVEC(JB) )
          C      JA=JA+ISTEPA
          C      JB=JB+ISTEPB
          C      CONTINUE
          C      200
          C
          C      SCAPRO=DP
          C      RETURN
          C      END
0012
0013
0014
```

```

0001      SUBROUTINE SETPAR(ICOM,IERR)
      C
      C      SETS THE PARAMETERS OF THE VECTORS IPAR AND PAR
      C
      C      LOGICAL LINTGR,LNUMBR,LEOCOM
      C
      C      DIMENSION IPMAX(11)
      C
      C      COMMON/REGPAR/IPAR(20),PAR(50)
      C
      C      DATA IPMAX/4*10,2*1,2*10000,1,2*15/
      C
      C      ICOM1=ICOM-10
      C      IF(ICOM1.GE.12) GO TO 20
      C
      C      IPAR(8)-(16) ARE SET
      C
      C      IF(.NOT.LINTGR(INUM)) GO TO 902
      C      IF(.NOT.LEOCOM(0)) GO TO 906
      C
      C      IF(INUM.LT.0) GO TO 936
      C      IF(INUM.GT.IPMAX(ICOM1)) GO TO 939
      C
      C      IF(ICOM1.GE.10) GO TO 10
      C
      C      IPAR(ICOM1+7)=INUM
      C      IP1=IPAR(8)+IPAR(9)+IPAR(10)+IPAR(13)
      C      IP2=MAX0(IPAR(8),IPAR(9),IPAR(10))+IPAR(11)
      C      IF(IP1.GT.10.OR.IP2.GT.10) GO TO 940
      C
      C      RETURN
      C
      C      IPAR(1)-(2) ARE SET
      C
      C      IPAR(ICOM1-9)=INUM
      C      RETURN
      C
      C      10      ICOM1=ICOM1-11
      C      IF(ICOM1.GE.11) GO TO 30
      C
      C      PAR(1)-(10) ARE SET
      C
      C      IF(.NOT.LNUMBR(RNUM)) GO TO 904
      C      IF(.NOT.LEOCOM(0)) GO TO 906
      C
      C      IF(RNUM.LT.0..AND.(ICOM1.LE.2.OR.ICOM1.GE.5)) GO TO 937
      C      IF(RNUM.GT.1..AND.(ICOM1.EQ.5.OR.ICOM1.EQ.6)) GO TO 938
      C
      C      PAR(ICOM1)=RNUM
      C      RETURN
      C
      C

```

```
C
C
0040      ICOM1=ICOM1-10
0041      IF(ICOM1.GT.3) GO TO 40
C
C      PAR(11)-(20);(21)-(30);(31)-(40) ARE SET
C
0043      IF(.NOT.LINTGR(INUM)) GO TO 902
0045      IF(.NOT.LNUMBER(RNUM)) GO TO 904
0047      IF(.NOT.LEOCOM(O)) GO TO 906
C
0049      IF(INUM.LT.1) GO TO 936
0051      IF(INUM.GT.10) GO TO 939
C
0053      I1=ICOM1*10+INUM
0054      PAR(I1)=RNUM
0055      RETURN
C
C
0056      ICOM1=ICOM1-3
C
C      PAR(41)- ARE SET
C
0057      IF(.NOT.LNUMBER(RNUM)) GO TO 904
0059      IF(.NOT.LEOCOM(O)) GO TO 906
0061      PAR(40+ICOM1)=RNUM
0062      RETURN
C
C
C      SET ERROR FLAGS
C
0063      IERR=2
0064      RETURN
C
0065      IERR=4
0066      RETURN
C
0067      IERR=6
0068      RETURN
C
0069      IERR=36
0070      RETURN
C
0071      IERR=37
0072      RETURN
C
0073      IERR=38
0074      RETURN
C
0075      IERR=39
0076      RETURN
C
0077      IERR=40
```

FORTRAN IV

VD1C-03F+ THU 08-NOV-79 08:53:03

PAGE 003

0078 RETURN

 C

0079 END

FORTRAN IV VD1C-03F+ THU 08-NOV-79 09:02:39

PAGE 001

```

0001            SUBROUTINE STOPP
          C
          C        EXECUTES THE COMMAND STOP
          C
0002            LOGICAL LEOCOM,LNAMES
          C
0003            DIMENSION ANSWER(2,2)
          C
0004            COMMON/ERRIND/IERR,ERRNAM(2)
          C
0005            DATA ANSWER/4HYES ,4H    ,4HNO ,4H    /
          C
          C        TEST IF COMMAND LINE IS CORRECT
          C
0006            IF(.NOT.LEOCOM(0)) GO TO 906
          C
          C        ASK IF THE OPERATOR WANTS TO PROCEED
          C
0008            10 CALL COMDEC(51HDO YOU REALLY MEAN STOPPING THE PROGRAM? (YES/NO)
          *    ,51)
          C
          C        INTERPRET AND CHECK IF THE ANSWER IS CORRECT
          C
0009            IF(.NOT.LNAMES(ANSWER,2,IANS)) GO TO 1908
0011            IF(.NOT.LEOCOM(0)) GO TO 1906
          C
          C        IF THE ANSWER WAS YES STOP THE PROGRAM
          C
0013            IF(IANS.EQ.1) STOP
          C
0015            RETURN
          C
          C        IF THE ANSWER TO THE QUESTION ABOVE IS INCORRECT CALL THE
          C        ERROR MESSAGE ROUTINE
          C
0016            800 CALL ERRMSG
0017            IERR=0
0018            GO TO 10
          C
          C        COMMAND LINE INCORRECT. RETURN TO OPCOM
          C
0019            906 IERR=6
0020            RETURN
          C
          C        ANSWER INCORRECT. RETURN TO QUESTION ABOVE
          C
0021            1906 IERR=6
0022            GO TO 800
          C
0023            1908 IERR=8
0024            GO TO 800
          C
0025            END

```



```

C      AL      - VECTOR OF CONTROL LAW PARAMETERS, (O)
C      X      - STATE USED , (O)
C      IND     - INDICATOR FOR HOW MANY ITERATIONS ARE PERFORMED IN THE
C              RICATTI EQUATION SOLUTION
C      IS      - DECLARED FIRST DIMENSION OF S, (I)
C      IS1     - DECLARED FIRST DIMENSION OF S1, (I)
C      WA      - WORK VECTOR, SIZE (
C      WB      - WORK VECTOR, SIZE (
C      WC      - WORK VECTOR, SIZE (
C      WY      - WORK VECTOR, SIZE (
C      WU      - WORK VECTOR, SIZE (
C      WE      - WORK VECTOR, SIZE (
C      WK      - WORK VECTOR, SIZE (
C      W1      - WORK MATRIX, SIZE (
C      W2      - WORK MATRIX, SIZE (
C      W3      - WORK MATRIX, SIZE (
C              , DIMENSIONED (IS1,1)
C              , DIMENSIONED (IS1,1)
C              , DIMENSIONED (IS1,1)
C
C      CHARACTERISTICS:
C      =====
C
C      LIBRARY SUBROUTINES REQUIRED:
C      -----
C      SCAPRO
C      REMOVE
C      NORM
C      CORI
C
C      SUBROUTINE STURE(S,ACT,T,N,K,KDEL,IBIAS,S1,ITER,Q2,RD,UR,AL,X,
C      *IND,IS,IS1,WA,WB,WC,WY,WU,WE,WK,W1,W2,W3)
C
C      DIMENSION S(IS,1),T(1),N(1),K(1),S1(IS1,1),AL(1),X(1)
C      DIMENSION WA(1),WB(1),WC(1),WY(1),WU(1),WE(1),WK(1)
C      DIMENSION W1(IS1,1),W2(IS1,1),W3(IS1,1)
C
C      INITIALIZATION
C
C      NA=N(1)
C      NB=N(2)
C      NC=N(3)
C
C      KB=K(2)
C
C      NM=MAXD(NA,NB,NC)
C      NAB=NA+NB
C      NMK=NM+KB
C
C      EPS=100.*RMACON(1)
C
C      INITIALIZATION OF S1 MATRIX
C
C      DO 4 I=1,NMK
C      DO 2 J=1,NMK

```

```

0015      2  S1(I,J)=0.
0016      4  S1(I,I)=1.
      C
      C  ORGANIZE SYSTEM PARAMETERS
      C
0017      X1=0.
0018      CALL RMOVE(X1,WA(1),0,1,NMK)
0019      CALL RMOVE(X1,WB(1),0,1,NMK)
0020      CALL RMOVE(X1,WC(1),0,1,NMK)
      C
0021      IF(NA.NE.0) CALL RMOVE(T(1),WA(1),1,1,NA)
0022      IF(NB.NE.0) CALL RMOVE(T(NA+1),WB(KB+1),1,1,NB)
0023      IF(NC.NE.0) CALL RMOVE(T(NAB+1),WC(1),1,1,NC)
      C
      C  SCALE SYSTEM PARAMETERS
      C
0027      SF=1.
0028      DO 10 I=1,NMK
0029      SF=SF*RO
0030      WA(I)=WA(I)/SF
0031      WB(I)=WB(I)/SF
0032      WC(I)=WC(I)/SF
      C
      C  SCALE INPUT,OUTPUT AND RESIDUAL VECTOR
      C
0033      SF=1.
0034      DO 20 I=1,NMK
0035      SF=SF*RO
0036      WY(I)=-S(1,I+1-KDEL)*SF
0037      WU(I)=S(2,I+1-KDEL)*SF
0038      WE(I)=S(3,I+1-KDEL)*SF
      C
      C  COMPUTE THE PREDICTED STATE VARIABLES
      C
0039      CALL RMOVE(X1,X(1),0,1,NMK)
0040      DO 30 I=1,NM
0041      X(I)=X(I)-SCAPRO(WY(1),WA(I),1,1,NM+1-I)
      *          +SCAPRO(WE(1),WC(I),1,1,NM+1-I)
0042      NPL=NMK-I+1
0043      X(NPL)=X(NPL)+SCAPRO(WU(1),WB(NPL),1,1,I)
0044      IF(KB.LE.0) GO TO 50
0045      DO 40 I=1,KB
0046      NPL=KB-I+2
0047      X(I)=X(I)+SCAPRO(WU(NPL),WB(KB+1),1,1,NM)
0048      GO TO 50
      C
      C  COMPUTE THE FILTER ESTIMATE
      C
0049      50  IF(KDEL.NE.0) GO TO 90
      C
0051      MAC=MAXD(NA,NC)
0052      IF(ABS(WA(MAC)).GT.EPS) GO TO 55
0053      X1=0.
0054      GO TO 58
0055      55  X1=WC(MAC)/WA(MAC)
0056

```

```

0057 WK(1)=1.-X1
0058 IF(MAC.EQ.1) GO TO 70
0060 DO 70 I=2,MAC
0061 WK(I)=WC(I-1)-WA(I-1)*X1
C
0062 RES=ACT-X(1)
0063 DO 80 I=1,MAC
0064 X(I)=X(I)+WK(I)*RES
C
C COMPUTE THE FEEDBACK
C
0065 DO 120 NLOOP=1,ITER
0066 IND=-NLOOP
0067 DO 100 I=1,NMK
0068 DO 100 J=1,NMK
0069 W3(I,J)=S1(I,J)
C
0070 CALL CORI(WA,WB,Q2,S1,AL,NMK,R3,IS1,W1,W2)
C
0071 DO 110 I=1,NMK
0072 DO 110 J=1,NMK
0073 W3(I,J)=W3(I,J)-S1(I,J)
C
C HAS THE ITERATION CONVERGED?
C
0074 RNORM=ANORM(W3,NMK,IS1)
0075 SNORM=ANORM(S1,NMK,IS1)
0076 IF(RNORM.LT.EPS*SNORM) GO TO 130
0078 CONTINUE
0079 IND=1
C
C COMPUTE THE CONTROL SIGNAL
C
0080 UR=-SCAPRO(AL(1),X(1),1,1,NMK)
0081 IF(IBIAS.EQ.0.OR.NB.EQ.0) RETURN
C
C CORRECT CONTROL SIGNAL IF BIAS IS ESTIMATED
C
0083 SF=0.
0084 DO 140 I=1,NB
0085 SF=SF+T(NA+I)
0086 NP=NA+NB+NC
C
0087 UR=UR-T(NP+1)/SF
C
0088 RETURN
0089 END

```

FORTRAN IV

VD1C-03F+ THU 08-NOV-79 08:43:06

PAGE 001

```

0001      SUBROUTINE WAIT(MODE,IFL,ICHAN,IBIT1)
C
C      IMPLEMENTS A BUSY WAIT FUNCTION, I.E. THE COMPUTER STAYS IN THIS
C      SUBROUTINE UNTIL IT IS TIME FOR THE NEXT SAMPLING
C
C      ERROR FLAG IFL 0   NORMAL RETURN WHEN IT IS TIME FOR SAMPLING
C      1   DIGITAL INPUT ICHAN/IBIT1 (SEE LBITIN) IS
C           FALSE
C      2-3  ERROR IN DOUBLE INTEGER MANIPULATIONS
C      4   THE ROUTINE WAS CALLED ERRNAM(1) S TOO
C           LATE FOR THE REQUESTED SAMPLING TIME
C
C      INITIALIZATION  MODE  THE FIRST CALL OF THE SUBROUTINE SHOULD
C                           USE MODE=1, THE REST MODE=0
C
C      VARIABLES      IMAX  NUMBER OF TICKS PER DAY
C                     ITSTIC THE SAMPLING INTERVAL IN TICKS
C                     JTIME1 COMPUTER TIME FETCHED WITH SUBROUTINE GTIM
C                     JTIME  COMPUTER TIME WITH THE DOUBLE INTEGER WORDS
C                           IN CORRECT ORDER
C                     JTIME2  PREVIOUS TIME
C                     ITMSAM  TIME FOR THE NEXT SAMPLING
C                     OVER   LOGICAL VARIABLE INDICATING WHEN A NEW DATE
C                           HAS OCCURRED
C
C      COMMON/TIME/  TS      THE SAMPLING INTERVAL
C                     ITMSAM TIME FOR THE NEXT SAMPLING
C
C      COMMON/ERRIND/ IERR   ERROR FLAG TO THE ERROR MESSAGE ROUTINE
C                     ERRNAM INFORMATION TO THE ERROR MESSAGE ROUTINE
C
0002      LOGICAL OVER
C
0003      INTEGER*4  JTIME,ITSTIC,ITMSAM,JDIF,JSUM,JTIME1,IMAX,JTIME2
C
0004      DIMENSION IY(2),IZ(2)
C
0005      COMMON/TIME/TS,ITMSAM
0006      COMMON/ERRIND/IERR,ERRNAM(2)
C
0007      EQUIVALENCE (JTIME1,IZ),(IY,JTIME)
C
0008      DATA V/4320000./
C
C      INITIALIZATION OF ERROR FLAG
C
0009      IFL=0
C
0010      IF(MODE.NE.1) GO TO 10
C
C      INITIALIZATION (MODE=1)
C
0012      OVER=.FALSE.
C

```

```

0013 TSTIC=50.*TS
0014 I1=JAFIX(TSTIC,ITSTIC)
0015 IF(I1.EQ.-2) GO TO 900
0017 IF(I1.LE.0) GO TO 910
0019 I1=JAFIX(V,IMAX)
      C
      C TEST OF DIGITAL INPUT USED. IF FALSE RETURN.
      C
0020 10 IF(.NOT.LBITIN(ICHAN,IBIT1)) GO TO 60
      C
      C FETCH THE COMPUTER TIME
      C
0022 CALL GTIM(JTIME1)
0023 IY(2)=IZ(1)
0024 IY(1)=IZ(2)
      C
0025 IF(MODE.NE.1) GO TO 15
      C
      C INITIALIZATION (MODE=1)
      C
0027 I1=JMOV(JTIME,JTIME2)
0028 I1=JADD(JTIME,ITSTIC,JSUM)
0029 GO TO 40
      C
      C CHECK IF NEXT REQUESTED SAMPLING IS LATER, NOW OR HAS ALREADY
      C OCCURRED
      C
0030 15 IF(JCMP(JTIME,JTIME2).EQ.-1) OVER=.FALSE.
0032 I1=JMOV(JTIME,JTIME2)
0033 I1=JMOV(ITMSAM,JSUM)
0034 IF(OVER) I1=JADD(ITMSAM,IMAX,JSUM)
0036 I1=JSUB(JTIME,JSUM,JDIF)
0037 IF(I1.EQ.-2) GO TO 900
      C
      C IF NOT YET TIME FOR SAMPLING RETURN TO LABEL 10
      C
0039 IF(I1) 10,30,20
      C
      C NEXT SAMPLING TIME REQUESTED HAS ALREADY OCCURRED
      C
0040 20 TIMLAG=AJFLT(JDIF)
0041 GO TO 920
      C
      C NEXT SAMPLING TIME REQUESTED IS NOW
      C
0042 30 I1=JADD(ITMSAM,ITSTIC,JSUM)
0043 IF(I1.EQ.-2) GO TO 900
0045 IF(I1.LE.0) GO TO 910
      C
0047 40 I1=JSUB(JSUM,IMAX,JDIF)
0048 IF(I1.EQ.-2) GO TO 900
0050 IF(I1.GE.0) GO TO 50
0052 I1=JMOV(JSUM,ITMSAM)
      C

```

FORTRAN IV

VO1C-03F+ THU 08-NOV-79 08:43:06

PAGE 003

```

C
C
C
0053      RETURN
C
C
C
0054      50      I1=JMOV(JDIF,ITMSAM)
0055             OVER=.TRUE.
0056      RETURN
C
C
C
0057      60      IFL=1
0058      RETURN
C
C
C
0059      900     IERR=41
0060             IFL=2
0061      RETURN
C
C
C
0062      910     IERR=42
0063             IFL=3
0064      RETURN
C
C
C
0065      920     IERR=22
0066             IFL=4
0067             ERRNAM(1)=TIMLAG/50.
0068      RETURN
C
0069      END
```

NORMAL RETURN WHEN IT IS TIME FOR SAMPLING. ITMSAM NOW CONTAINS
NEXT SAMPLING TIME.

RETURN

NEXT SAMPLING WILL OCCUR ON THE NEXT DAY (AFTER 24.00)

I1=JMOV(JDIF,ITMSAM)
OVER=.TRUE.
RETURN

DIGITAL INPUT CHECKED WAS FALSE

IFL=1
RETURN

ERRORS IN DOUBLE INTEGER COMPUTATIONS

IERR=41
IFL=2
RETURN

IERR=42
IFL=3
RETURN

THE ROUTINE WAS CALLED TOO LATE FOR THE NEXT REQUESTED SAMPLING

IERR=22
IFL=4
ERRNAM(1)=TIMLAG/50.
RETURN

END

```

FORTRAN IV      V01C-03F+  THU 08-NOV-79 08:41:34      PAGE 001

0001      SUBROUTINE WRIT(IERR)
0002      HANDLES THE COMMAND LIST [(DEV)] PAGNR
0003      LOGICAL LDLIMX,LNAMES,LEOCOM
0004      DIMENSION DEV(2,2),PAGE(2,5)
0005      COMMON/IODEVS/LKB,LTO,LPA
0006      DATA PARL,PARR/4H( ,4H) /
0007      DATA DEV/4HTT ,4H ,4HLP ,4H /
0008      DATA PAGE/4HPAG1,4H ,4HPAG2,4H ,4HPAG3,4H ,4HPAG4,
0009      *4H ,4HPAG5,4H /
0010      CODING THE COMMAND
0011      IF(.NOT.LDLIMX(PARL)) GO TO 10
0012      IF(.NOT.LNAMES(DEV,2,IDEV)) GO TO 900
0013      IF(.NOT.LDLIMX(PARR)) GO TO 910
0014      GO TO 20
0015      IDEV=1
0016      IF(.NOT.LNAMES(PAGE,5,IPAG)) GO TO 920
0017      IF(.NOT.LEOCOM(0)) GO TO 930
0018      LU=LTO
0019      IF(IDEV.EQ.2) LU=LPA
0020      CHOICE OF WRITING ROUTINE
0021      GO TO(100,200,300,400,500),IPAG
0022      CALL WRIT1(LU)
0023      RETURN
0024      CALL WRIT2(LU)
0025      RETURN
0026      CALL WRIT3(LU)
0027      RETURN
0028      CALL WRIT4(LU)
0029      RETURN
0030      IFL=2
0031      CALL WRIT5(LU,IFL)
0032      IFL=0
0033      RETURN
0034      SETTING OF ERROR FLAGS
0035      IERR=34
0036      RETURN
0037

```

```
0038 C 910 IERR=7  
0039 RETURN  
0040 C 920 IERR=35  
0041 RETURN  
0042 C 930 IERR=6  
0043 RETURN  
0044 C END
```

FORTRAN IV

V01C-03F+ THU 08-NOV-79 08:47:17

PAGE 001

0001 SUBROUTINE WRIT1(LU)

C

C WRITES INITIAL INFORMATION TO THE OPERATOR. CAN ALSO BE DIS-
 C PLAYED BY THE COMMAND
 C LIST [(DEV)] PAG1

0002 COMMON/REGPAR/IPAR(20),PAR(50)

C

0003 WRITE(LU,100)

0004 WRITE(LU,110)

0005 WRITE(LU,120)

0006 WRITE(LU,130)

0007 WRITE(LU,140)

0008 WRITE(LU,150)

0009 WRITE(LU,160)

0010 WRITE(LU,170)

0011 WRITE(LU,175)

0012 WRITE(LU,180)

0013 WRITE(LU,190)

0014 WRITE(LU,200)

0015 WRITE(LU,210)

0016 WRITE(LU,220)

0017 WRITE(LU,230)

0018 WRITE(LU,240) IPAR(2)

0019 WRITE(LU,250) IPAR(1)

C

0020 FORMAT(1H1,'THIS PROGRAM REALIZES A SELF-TUNING REGULATOR OF
 * STURE2-TYPE USING A RECURSIVE',/,', MAXIMUM LIKELIHOOD ESTIMATOR')
 0021 FORMAT(' THE PROGRAM OPERATES IN TWO MODES: OPERATOR
 * COMMUNICATION MODE',/,36X,'ESTIMATION/CONTROL MODE')
 0022 FORMAT(' THE OPERATOR COMMUNICATION PROGRAM IS OF COMMAND TYPE.
 * AVAILABLE COMMANDS:')
 0023 FORMAT(' NA IVAL / KDEL IVAL / IMEAS IVAL / DU VAL / Q2 VAL /
 * TH0 IVAL VAL')
 0024 FORMAT(' NB IVAL / BIAS IVAL / MAN VAL / UL VAL / RO VAL /
 * PO IVAL VAL')
 0025 FORMAT(' NC IVAL / ITER IVAL / RLIMH VAL / UH VAL / RUN /
 * R1 IVAL VAL')
 0026 FORMAT(' K IVAL / IRES IVAL / DELTA VAL / WO VAL / STOP /
 * LIST [(DEV)] PAGNR')
 0027 FORMAT(' ILS IVAL / IOUT IVAL / TS VAL / W1 VAL / INIT /
 *
 *
 0028 FORMAT(' WBIAS VAL/ RLIML VAL',/)
 0029 FORMAT(' THE COMMAND RUN TRANSFERS THE PROGRAM TO THE ESTIMATION/
 *CONTROL MODE')
 0030 FORMAT(' THE PROGRAM USES TWO DIGITAL INPUTS, 0 AND 1:')
 0031 FORMAT(' 0:FALSE TRANSFER BACK TO THE OPERATOR
 * COMMUNICATION MODE')
 0032 FORMAT(' TRUE REMAINS IN ESTIMATION/CONTROL MODE')
 0033 FORMAT(' 1:FALSE ESTIMATION ONLY')
 0034 FORMAT(' TRUE ESTIMATION AND CONTROL')
 0035 FORMAT(' THE PROCESS OUTPUT SIGNAL SHOULD BE CONNECTED TO
 * ANALOG INPUT ',I5', [IMEAS]')
 0036 FORMAT(' THE CONTROL SIGNAL IS AVAILABLE AT ANALOG OUTPUT',

FORTRAN IV V01C-03F+ THU 08-NOV-79 08:47:17

PAGE 002

 *14X,I5,' [IOUT]')

 C

0037 RETURN

0038 END

FORTRAN IV V01C-03F+ THU 08-NOV-79 08:48:14

PAGE 001

```

0001 SUBROUTINE WRIT2(LU)
C
C WRITES CURRENT VALUES OF THE IDENTIFICATION AND REGULATOR
C PARAMETERS: NA, NB, NC, K, BIAS, WBIAS, TS, WO, W1, THO, PO,
C KDEL, Q2, RO, UL, UH. CAN BE DISPLAYED BY THE COMMAND
C LIST [(DEV)] PAG2
C
C COMMON/REGPAR/IPAR(20),PAR(50)
C
C DATA PY/4H YES/,PN/4H NO/
C
C WRITE(LU,100)
C WRITE(LU,110)
C WRITE(LU,120)
C P=PN
C IF(PAR(2).GT.0.5) P=PY
C WRITE(LU,130) P
C WRITE(LU,140) IPAR(8),IPAR(9),IPAR(10)
C P=PY
C IF(IPAR(13).EQ.0) P=PN
C WRITE(LU,150) IPAR(11),P
C WRITE(LU,160) PAR(1)
C WRITE(LU,170)
C WRITE(LU,180) PAR(6),PAR(6),PAR(5)
C WRITE(LU,155) PAR(41)
C NP=IPAR(8)+IPAR(9)+IPAR(10)+IPAR(13)
C NS=MINO(NP,5)
C IF(NS.EQ.0) GO TO 20
C WRITE(LU,190) (PAR(10+I),I=1,NS)
C WRITE(LU,200)
C IF(NP.LE.5) GO TO 10
C WRITE(LU,210) (PAR(10+I),I=6,NP)
C GO TO 15
C
C 10 WRITE(LU,215)
C 15 WRITE(LU,220) (PAR(20+I),I=1,NS)
C WRITE(LU,230)
C IF(NP.LE.5) GO TO 20
C WRITE(LU,240) (PAR(20+I),I=6,NP)
C GO TO 25
C
C 20 WRITE(LU,215)
C 25 WRITE(LU,240)
C P=PY
C IF(IPAR(12).EQ.1) P=PN
C WRITE(LU,250) P,IPAR(12)
C WRITE(LU,260) PAR(9)
C WRITE(LU,270) PAR(10)
C WRITE(LU,280) PAR(3),PAR(4)
C
C 100 FORMAT(1H1,15X,'R E G U L A T O R S T U R E 2 W I T H R M
C * L',/)
C 110 FORMAT(11X,'-1',7X,'-K',3X,'-1',9X,'-1',14X,'-1',6X,'-1',10X,
C *'-NB')
C 120 FORMAT(' MODEL: A(Q)Y(T)=Q B(Q)U(T)+C(Q)E(T) B(Q)
C *=B1*Q +...+BNB*Q ',/)

```

```

FORTRAN IV      VD1C-03F+  THU 08-NOV-79 08:48:14      PAGE 002

0049      130      FORMAT(' MODEL PARAMETERS:',16X,' DIFFERENCES OF INPUT USED',
          *7X,A4)
0050      140      FORMAT(' ORDER OF A-,B-,C-POLYNOMIALS [NA],[NB],[NC]',9X,3I5)
0051      150      FORMAT(' PURE TIME DELAY [K]',I5,7X,' ESTIMATE OF BIAS [BIAS]',
          *9X,A4)
0052      155      FORMAT(' FORGETTING FACTOR FOR BIAS ESTIMATE [WBIAS]',8X,
          *1PG11.4)
0053      160      FORMAT('/', ' SAMPLING INTERVAL [TS] ',1PG11.4,' S', '/')
0054      170      FORMAT(' IDENTIFICATION PARAMETERS:')
0055      180      FORMAT(' WEIGHTING PROFILE [W0],[W1] W(T+1)=' ,F6.4,' *W(T)
          *+(1.-',F6.4,') W(0)=' ,F6.4)
0056      190      FORMAT(' INITIAL PARAMETER',5(1X,1PG11.4))
0057      200      FORMAT('$ ESTIMATES [TH0]',1X)
0058      210      FORMAT(2H+ ,5(1X,1PG11.4))
0059      215      FORMAT(2H+ )
0060      220      FORMAT(' INITIAL P-MATRIX ',5(1X,1PG11.4))
0061      230      FORMAT('$ DIAGONAL [PO] ',')
0062      240      FORMAT('/', ' REGULATOR PARAMETERS (IN AUTO MODE):')
0063      250      FORMAT(' CONTROL SIGNAL U(T) DEPENDS ON Y(T) [KDEL]',16X,
          *A4,' (KDEL:',I1,')')
0064      260      FORMAT(' PENALTY ON CONTROL SIGNAL [Q2]',21X,1PG11.4)
0065      270      FORMAT(' CLOSED LOOP POLES INSIDE A CIRCLE WITH RADIUS [RO]',
          *1X,1PG11.4)
0066      280      FORMAT(' LOW AND HIGH LIMIT OF CONTROL SIGNAL [UL],[UH]',
          *5X,2(1PG11.4))

C
0067      RETURN
0068      END

```

FORTRAN IV

VD1C-03F+ THU 08-NOV-79 08:49:07

PAGE 001

```

0001      SUBROUTINE WRIT3(LU)
      C
      C      WRITES CURRENT VALUES OF ADDITIONAL IDENTIFICATION AND
      C      REGULATOR PARAMETERS: RLIML, R1, IRES, RLIMH, DELTA, ILS,
      C      ITER. WILL BE DISPLAYED BY THE COMMAND
      C      LIST [(DEV)] PAG3
      C
0002      COMMON/REGPAR/IPAR(20),PAR(50)
      C
0003      DATA PY/4H YES/,PN/4H NO/
      C
0004      WRITE(LU,100)
0005      WRITE(LU,105) PAR(42)
0006      NP=IPAR(8)+IPAR(9)+IPAR(10)+IPAR(13)
0007      NS=MIND(NP,5)
0008      IF(NS.EQ.0) GO TO 10
0010      WRITE(LU,110) (PAR(30+I),I=1,NS)
0011      WRITE(LU,120)
0012      IF(NP.LE.5) GO TO 5
0014      WRITE(LU,130) (PAR(30+I),I=6,NP)
0015      GO TO 10
0016      WRITE(LU,135)
0017      P=PY
0018      IF(IPAR(16).EQ.1) P=PN
0020      WRITE(LU,140) P,IPAR(16)
0021      WRITE(LU,150) PAR(7)
0022      WRITE(LU,160) PAR(8)
0023      WRITE(LU,170) IPAR(14)
0024      WRITE(LU,180)
0025      WRITE(LU,190) IPAR(15)
      C
0026      FORMAT(1H1,' ADDITIONAL IDENTIFICATION PARAMETERS:')
0027      FORMAT(' NO UPDATING IF RESIDUALS LESS THAN [RLIML]=',6X,
      *1PG11.4)
0028      FORMAT(' PARAMETER NOISE',2X,5(1X,1PG11.4))
0029      FORMAT('$ COVARIANCE [R1]',6X)
0030      FORMAT(2H+ ,5(1X,1PG11.4))
0031      FORMAT(2H+ )
0032      FORMAT(' A PRIORI ESTIMATE OF RESIDUALS [IRES]',19X,A4,' (
      *IRES:',11,')')
0033      FORMAT(' UPPER LIMIT OF RESIDUALS [RLIMH]',17X,1PG11.4)
0034      FORMAT(' DELTA IN P-UPDATING EQ. [DELTA]',18X,1PG11.4)
0035      FORMAT(' LS-ESTIMATE UNTIL STEP [ILS]',22X,110,/)
0036      FORMAT(' ADDITIONAL REGULATOR PARAMETERS (IN AUTO MODE):')
0037      FORMAT(' MAXIMUM NUMBER OF ITERATIONS OF RICATTI EQ. [
      *ITER]',110)
      C
0038      RETURN
0039      END

```

```

FORTRAN IV      V01C-03F+   THU 08-NOV-79 08:49:44      PAGE 001

0001      SUBROUTINE WRIT4(LU)
0002      COMMON/REGPAR/IPAR(20),PAR(50)
0003      COMMON/STATE/S(3,11),T(10),P(10,10),Z(20),WT,X(10),S1(10,10),
          *T1(10)
0004      COMMON/RES/RES,UR,AL(10),IND,UROLD,W
0005      WRITE(LU,100)
0006      WRITE(LU,110)
0007      WRITE(LU,120)
0008      N1=IPAR(8)+IPAR(9)+IPAR(10)+IPAR(13)
0009      N2=MAX0(IPAR(8),IPAR(9),IPAR(10))+IPAR(11)
0010      DO 10 I=1,10
0011      I1=I-1
0012      I2=I+1
0013      SS=-S(1,I2)
0014      IF(I.LE.N1.AND.I.LE.N2) WRITE(LU,130) I1,SS,S(2,I2),S(3,I2),
          *I,T(I),AL(I),X(I)
0016      IF(I.GT.N1.AND.I.GT.N2) WRITE(LU,130) I1,SS,S(2,I2),S(3,I2)
0018      IF(I.GT.N1.AND.I.LE.N2) WRITE(LU,140) I1,SS,S(2,I2),S(3,I2),
          *I,AL(I),X(I)
0020      IF(I.LE.N1.AND.I.GT.N2) WRITE(LU,130) I1,SS,S(2,I2),S(3,I2),
          *I,T(I)
0022      10 CONTINUE
0023      WRITE(LU,150) WT
0024      WRITE(LU,155) PAR(41)
0025      WRITE(LU,160) W
0026      100 FORMAT(1H1,25X,'R E G U L A T O R   S T A T E',/)
0027      110 FORMAT(1X,'TIME',3X,'OUTPUT',5X,'CONTROL',4X,'RESIDUALS',2X,
          *'NR',3X,'PARAMETER',3X,'REGULATOR',3X,'REGULATOR')
0028      120 FORMAT(20X,'SIGNAL',20X,'ESTIMATES',2X,'PARAMETERS',4X,'STATES',
          */)
0029      130 FORMAT(1X,'T-',I1,3(1X,1PG11.4),1X,I2,1X,3(1X,1PG11.4))
0030      140 FORMAT(1X,'T-',I1,1X,3(1X,1PG11.4),1X,I2,12X,2(1X,1PG11.4))
0031      150 FORMAT(/,' CURRENT WEIGHTING FACTOR:',11X,1PG11.4)
0032      155 FORMAT(' FORGETTING FACTOR FOR BIAS ESTIMATE:',1PG11.4)
0033      160 FORMAT(/,' LOSS FUNCTION:',11X,1PG11.4)
0034      RETURN
0035      END

```

FORTRAN IV VD1C-03F+ THU 08-NOV-79 08:42:23 PAGE 001

```

0001            SUBROUTINE WRIT5(LU,IFL)
C
C            WRITES MESSAGE TO OPERATOR BEFORE LEAVING OPERATOR
C            COMMUNICATION MODE. CAN ALSO BE DISPLAYED BY THE COMMAND
C            LIST [(DEV)] PAGES
C
0002            LOGICAL LBITIN,LEOCOM,LNAMES
C
0003            DIMENSION ANSWER(2,2)
C
0004            COMMON/ERRIND/IERR,ERRNAM(2)
C
0005            DATA ICHAN,IBIT1,IBIT2/D,0,0,1/
0006            DATA ANSWER/4HYES ,4H        ,4HNO ,4H        /
C
0007            WRITE(LU,100)
0008            WRITE(LU,110)
0009            WRITE(LU,120)
0010            IF(.NOT.LBITIN(ICCHAN,IBIT1)) WRITE(LU,140)
0012            IF(LBITIN(ICCHAN,IBIT1)) WRITE(LU,130)
0014            IF(.NOT.LBITIN(ICCHAN,IBIT2)) WRITE(LU,150)
0016            IF(LBITIN(ICCHAN,IBIT2)) WRITE(LU,160)
0018            WRITE(LU,170)
0019            WRITE(LU,180)
C
0020            IF(IFL.EQ.2) RETURN
C
0022            WRITE(LU,190)
0023            WRITE(LU,200)
C
0024            10    CALL COMDEC(4H> ,1)
C
0025            IF(.NOT.LNAMES(ANSWER,2,IANS)) GO TO 1908
0027            IF(.NOT.LEOCOM(0)) GO TO 1906
C
0029            IFL=0
0030            IF(IANS.EQ.2) IFL=1
C
0032            RETURN
C
0033            800    CALL ERRMSG
0034            IERR=0
0035            GO TO 10
C
0036            1906    IERR=6
0037            GO TO 800
C
0038            1908    IERR=8
0039            GO TO 800
C
0040            100    FORMAT(1H1,'YOU ARE NOW LEAVING THE OPERATOR COMMUNICATION
                  * MODE.',/)
0041            110    FORMAT(' CHECK THE DIGITAL INPUTS 0 AND 1 AND CORRECT IF
                  * NECESSARY.')
```

FORTRAN IV

V01C-03F+ THU 08-NOV-79 08:42:23

PAGE 002

```
0042 120 FORMAT(' THE DIGITAL INPUTS NOW HAVE THE MEANING:')
0043 130 FORMAT(' 0: START ESTIMATION (/CONTROL)')
0044 140 FORMAT(' 0: GO BACK TO OPERATOR COMMUNICATION MODE')
0045 150 FORMAT(' 1: ESTIMATION ONLY')
0046 160 FORMAT(' 1: ESTIMATION AND CONTROL')
0047 170 FORMAT('/', ' IF YOU HAVE TYPED THE COMMAND INIT BEFORE RUN THE
      * REGULATOR IS INITIALIZED', '/', ' AT YOUR DESIRE')
0048 180 FORMAT(' IF INIT IS NOT TYPED BETWEEN TWO RUN COMMANDS
      * THE REGULATOR WILL CONTINUE', '/', ' FROM THE PREVIOUS STATE')
0049 190 FORMAT('/', ' IF YOU WANT TO CONTINUE TO THE ESTIMATION/
      *CONTROL MODE, TYPE YES')
0050 200 FORMAT(' IF YOU WANT TO GO BACK TO THE OPERATOR COMMUNICA
      *TION MODE, TYPE NO')
```

C

0051 END

APPENDIX B