Real Time Computing II Minimal Variance Control on Process Computer

Borisson, Ulf; Holst, Jan

1971

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

Link to publication

Citation for published version (APA):
Borisson, U., & Holst, J. (1971). *Real Time Computing II Minimal Variance Control on Process Computer*. (Research Reports TFRT-3032). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:
2

REAL TIME COMPUTING II
MINIMAL VARIANCE CONTROL ON PROCESS COMPUTER.

U. BORISSON
J. HOLST

REPORT 7108(B) SEPTEMBER 1971
LUND INSTITUTE OF TECHNOLOGY
DIVISION OF AUTOMATIC CONTROL

REAL TIME COMPUTING II

MINIMAL VARIANCE CONTROL ON PROCESS COMPUTER.

U. Borisson - J. Holst

ABSTRACT.

In this report a regulator using a minimal variance strategy is discussed and a FORTRAN algorithm for on-line control on process computer has been worked out. The optimal regulator adapts itself to a suboptimal strategy if its sensitivity to variations in the process parameters is great. A program for off-line computation of minimal variance strategies is also given. If the process parameters are constant, the regulator is designed by means of the off-line program. Deadbeat strategies can also be computed by the given algorithms.

TABLE OF CONTENTS. Page

Table of Contents (contd.)

1. INTRODUCTION.

Consider a discrete system given by

$$A^{*}(q^{-1})y(t) = q^{-K}B^{*}(q^{-1})u(t) + \lambda C^{*}(q^{-1})e(t)$$

where

u(t) is the input to the system at the time t,
y(t) is the output from the system at the time t,
e(t) (t = 0, ±1, ±2, ...) are equally distributed, independent, normal N(0,1) random variables,
q^{-1} is the backward shift operator,
A^{*}, B^{*} and C^{*} are polynomials of degree N,
K is the time delay in the system,
λ is a constant.

The purpose of the regulator is to minimize the variance of the system output.
If the polynomial $B(q) = q^{N}B^{*}(q^{-1})$ has all zeroes inside the unit circle the minimal variance strategy is given by

$$u(t) = - \frac{G^{*}(q^{-1})}{F^{*}(q^{-1})B^{*}(q^{-1})}\, y(t) \qquad (1)$$

where $F^{*}(q^{-1})$ and $G^{*}(q^{-1})$ are defined by the identity

$$C^{*}(q^{-1}) = F^{*}(q^{-1})A^{*}(q^{-1}) + q^{-K}G^{*}(q^{-1}) \qquad (2)$$

If the B polynomial has any zero outside the unit circle the above strategy is extremely sensitive to variations in the process parameters and it cannot be used in practice. In this case it is necessary to use a modified, suboptimal regulator. Consequently a minimal variance control algorithm must include a test which examines if all the zeroes of the B polynomial are inside the unit circle.

Introduce the polynomial partitioning

$$B^{*}(q^{-1}) = B1^{*}(q^{-1})\, B2^{*}(q^{-1})$$

where the degrees of B1 and B2 are N1 and N2. The polynomial $q^{N1}B1^{*}(q^{-1})$ is to have all zeroes inside the unit circle and $q^{N2}B2^{*}(q^{-1})$ all zeroes outside or on the unit circle. The minimal variance control law now becomes

$$u(t) = - \frac{G^{*}(q^{-1})}{F^{*}(q^{-1})B1^{*}(q^{-1})}\, y(t) \tag{3}$$

where the polynomials $F^{*}(q^{-1})$ and $G^{*}(q^{-1})$ are given by the identity

$$C^{*}(q^{-1}) \equiv F^{*}(q^{-1})A^{*}(q^{-1}) + q^{-K}B2^{*}(q^{-1})G^{*}(q^{-1}) \tag{4}$$

This strategy is suboptimal, but it is realizable in practice. It should be observed that the system of equations (4) cannot be solved recursively. However, this is possible to do with the system of equations (2). By putting $C^{*}(q^{-1}) \equiv 1$ in the above equations a dead-beat strategy is achieved. A detailed discussion of the above strategies is given in [1].

It is evident that it is more complicated to implement a suboptimal regulator than an optimal one. In the first case the zeroes of the B polynomial must be determined and that may demand long computing time. However, in Chapter 3 some methods to make this procedure less time demanding will be discussed.

If the zeroes of the B polynomial lie inside the unit circle but close to it, the optimal regulator may still be rather sensitive to variations in the process parameters. Therefore the on-line algorithm MIVRE has been carried out in such a way that a suboptimal strategy is computed if there is any zero of the B polynomial outside a circle with radius 0.9. In the off-line program MIVCO the radius of the circle is optional.

2. THE MINIMAL VARIANCE ALGORITHM.

Transferring of data between subprograms can be done either via parameters in the subprogram call or via a COMMON area shared by the calling and the called program. In the subprograms presented here the former method is used. The purpose for this is that it is easy to use the regulator algorithm in governing more than one process by storing the relevant parameters for each process in a superior program.

In most of the programs it has been assumed that polynomials may have an arbitrary coefficient in the highest degree term. Especially, for a system of order n, this means that n+1 coefficients must be specified in each of the A, B and C polynomials.

The programs do not allow system order and system delay to be greater than 10. Further, the sum of the system order and the number of zeroes of the B polynomial outside the critical circle must not be greater than 14. However, it is easy to modify those limits in accordance with the amount of storage available.

The critical circle has the radius 0.9 in MIVRE. The reason for not using the unit circle in this case is that the closed system is sensitive to variations in the system parameters when the B polynomial has zeroes inside but close to the unit circle. In MIVCO the radius of the critical circle must be specified by user.

2.1. ADMINISTRATIVE ROUTINES.

MIVRE (MInimal Variance REgulator) and MIVCO (MInimal Variance COefficients) are the two administrative routines. MIVRE is used in on-line and MIVCO in off-line calculations of minimal variance and dead-beat strategies. A block diagram for these programs is found in Appendix I.

2.1.1. MIVRE.

The regulator call is

CALL MIVRE (A,B,C,H,GØ,ZR,ZI,U,Y,N,K,NH,NØ,IL,IT,EPS)

A, B and C	are polynomials describing the system to be regulated. They are represented as one-dimensional vectors.
H and GØ	are the denominator and the numerator polynomials of the old regulator. They are one-dimensional vectors.
ZR and ZI	are vectors containing the real and the imaginary parts of the zeroes of the old B polynomial.
U and Y	are vectors containing the latest process inputs and outputs.
N	is the order of the system.
K	is the time delay in the system.
NH	is the degree of the H polynomial.
NØ	is the degree of the GØ polynomial.
IL,IT,EPS	are indicators for the calculations. IL is used in MIVRE. IT and EPS are transferred to the subroutine for root calculations.

Information about the system to be regulated is given to MIVRE by the parameters A, B, C, N and K. The latest process inputs and outputs are gathered in the U and Y vectors. The subroutine then produces the new input as the first element in the input vector. The vector C must be dimensioned 20 because of the calculations in EQ (see 2.3.2).

In the subroutine the value of IL is tested. If IL=2 the old regulator, saved in H, GØ, NH and NØ, is used in computing the new input to the process. When IL=1 the B polynomial is partitioned into two parts, B1 and B2, with zeroes inside resp. outside a circle with radius 0.9 (see 2.2). The B2 polynomial is used in the calculation of the F and G polynomials (see 1 and 2.3).

The computation of the F and G polynomials fails if the system of equations for the coefficients is not possible to solve. In this case the parameter IS is returned greater than zero from the subroutine DECOM. This causes the old regulator to be used in the computations of the new input. If IS is returned zero a new regulator is calculated.

2.1.2. MIVCO.

MIVCO is written as a conversational program. It asks for information about system order, system delay and the A, B and C polynomials describing the process. In order to facilitate the input of polynomials the subroutine POLIN is used (see 2.1.2.1). User must also specify the radius of the critical circle which is required in the partitioning of the B polynomial as was described above.

On the lineprinter the program prints out the input data and the partitioned B polynomial as well as the F and G polynomials and the resulting regulator. The program gives an error message if it was not possible to solve the system of equations for the F and G polynomials. There are two possible error messages,

ROW ZERO IN DECOM

and

PIVOT TOO SMALL IN DECOM

corresponding to IS=1 and IS=2 in DECOM (see 2.3.1).

At the end of the program user is asked whether another regulator is to be calculated or not. The answer (YES or NO) is interpreted by the subroutine OUI (see 2.1.2.2).

In both MIVRE and MIVCO the coefficient of the highest degree term in the B polynomial is transferred to the G polynomial. Therefore the coefficient of the highest degree term in the product $B^{*}F^{*}$ in the regulator denominator always is 1.0.

User may return to the start of the program by typing CTRL P if, for example, any parameter has been incorrectly specified.

In 2.2 the polynomial partitioning is discussed. The solution of the minimal variance identity for F and G is treated in 2.3 and in 2.4 the calculation of the new input is discussed.

2.1.2.1. POLIN.

POLIN is a subroutine which reads a specified number of polynomial coefficients written on the teletype in free format. The call is

CALL POLIN(A,N,T)

A is the polynomial which is to be read in,
N is the number of coefficients,
T is a Hollerith constant containing the name of the polynomial.

If T has the value 'A', the subroutine prints the text ENTER A-POLYNOMIAL before user is to type the coefficients.

2.1.2.2. OUI.

This is a subroutine which prints out the two possible answers of a question proposed just before the subroutine call. Then it interprets the given answer. The call is

CALL OUI(T,L)

T is a vector of dimension 2,
L is a logical variable.

T contains the possible answers as two Hollerith constants of not more than five letters. L is returned .TRUE. if the answer is T(1) and .FALSE. if it is T(2). The subroutine is not left until the given answer equals T(1) or T(2).

2.2. ROUTINES FOR POLYNOMIAL PARTITIONING.

For reasons mentioned in Chapter 1, it may be necessary to partition the B polynomial into two polynomials B1 and B2, where B1 has its zeroes inside and B2 outside or on the critical circle. This is done in POLPA (POLynomial PArtitioning).

The following subroutines are required in POLPA:

- o INPOL - a modified Schur-Cohn test (see 2.2.2)
- o FAMU - to multiply complex factors (see 2.2.3)
- o POLNO - to normalize a polynomial (see 2.2.4)
- o ROT - to find the zeroes of the B polynomial, [2].

2.2.1. POLPA.

POLPA is the main routine for polynomial partitioning. The call is

```
CALL POLPA(B,B1,B2,ZR,ZI,N,N1,N2,IN,R,IT,EPS)
```

B	is a vector containing the coefficients of the polynomial to be partitioned,
B1 and B2	are vectors containing the two parts of the B polynomial,
ZR and ZI	are vectors containing the real and the imaginary parts of the zeroes of the old B polynomial,
N	is the degree of B,
N1 and N2	are the degrees of B1 and B2,
IN	is a switch for the calculations in MIVAR

(see 2.3.1). IN is returned =1 if all zeroes lie inside the critical circle and =2 if not,

R — is the radius of the critical circle,

IT and EPS — are indicators for the calculation of the zeroes.

First the subroutine tests if all zeroes lie inside the circle. This is done by the integer function INPOL (see 2.2.2). If all zeroes lie inside the circle, the coefficient of the highest degree term is transferred to B2 and the coefficients of the normalized B polynomial to the vector B1. Further, N1 is set to N and N2 to zero.

If any zero lies outside or on the circle the explicit values of the zeroes must be calculated. The subroutine ROT demands that the coefficient of the highest degree term is 1.0 and that the vector containing the coefficients has as many elements as the degree of the polynomial. Therefore the B polynomial first must be normalized. This is done in POLNO (see 2.2.4). When computing the zeroes it is possible to use the old values of the zeroes as initial values. This is done by putting IT=1. EPS is the error in the computed zeroes.

The complex valued zeroes are sorted with regard to their position relatively the critical circle. They are stored in two pairs of vectors. Then they are multiplied in FAMU to form the B1 and B2 polynomials.

The coefficient of the term in B with the highest degree is transferred to B2. In B1 the corresponding coefficient is 1.0.

2.2.2. INPOL.

This is an integer function which performs a modified Schur-Cohn test to decide whether or not all zeroes of a given polynomial lie inside a specified circle. The call is

IN = INPOL(A,N,R)

A is a vector containing the coefficients of the polynomial,
N is the degree of the polynomial,
R is the radius of the critical circle.

In order to take account of the arbitrary radius R, the polynomial is written

$$A(1) \cdot R^N \cdot \left(\frac{z}{R}\right)^N + A(2) \cdot R^{N-1} \cdot \left(\frac{z}{R}\right)^{N-1} + \ldots + A(N+1)$$

Then an ordinary Schur-Cohn test is done on this polynomial. INPOL is returned =1 if all zeroes lie inside the circle and =2 if any zero lies outside or on the circle.

2.2.3. FAMU.

FAMU (FActor MUltiplication) multiplies complex factors of the type (s-a-ib) to form a complex polynomial. The call is

CALL FAMU(BR,BI,ZR,ZI,N)

BR and BI are vectors of dimension N containing the real and the imaginary parts of the coefficients in the resulting polynomial,

ZR and ZI are vectors of dimension N containing the real and the imaginary parts of the factors,

N is the number of factors to be multiplied.

As the routine works on factors of the type (s-a-ib), the coefficient of the highest degree term is 1.0 and BR(1)+BI(1)·i is the coefficient of the term with degree N-1.

2.2.4. POLNO.

POLNO (POLynomial NOrmalization) is used for normalization of a given polynomial. The call is

```
CALL POLNO(B,NB,BØ)
```

B is a vector containing the coefficients of the polynomial,

NB is the degree of the polynomial,

BØ is returned as the old value of the coefficient of the highest degree term.

It is assumed that the coefficient of the highest degree term in the input polynomial is specified in B(1). The coefficients of the resulting polynomial are calculated according to B(I)=B(I+1)/BØ where BØ is the old value of B(1). Observe that the input value of the B polynomial is destroyed.

2.3. ROUTINES FOR SOLVING THE MINIMAL VARIANCE IDENTITY.

In Chapter 1 the regulator was discussed. It was stated that the computed regulator would be optimal or suboptimal depending on the position of the zeroes of the B polynomial. The calculation of the F and G polynomials, upon which the regulator is based, is done in MIVAR (see 2.3.1). The system of equations for the minimal variance coefficients is shown in Fig.1. In order to shorten the code a special subroutine EQ has been written (see 2.3.2).

If the B polynomial has any zero outside or on the critical circle, it is necessary to solve a non-recursive system of equations. For this purpose DECOM and SOLVB are used.

2.3.1. MIVAR.

MIVAR (MInimal VARiance) is a routine for calculation of the F and G polynomials according to an optimal or suboptimal strategy as was outlined above. The call is

```
CALL MIVAR(A,B2,C,F,G,N,N2,K,IN,IS)
```

A and C — are vectors containing the coefficients of the A and C polynomials describing the system,

B2 — is a vector containing the coefficients of the part of the B polynomial which has all its zeroes outside or on the critical circle,

F and G — are vectors containing the coefficients of the resulting polynomials for the regulator building,

$$
\begin{bmatrix}
1 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots\ 0 \\
a_1 & 1 & 0 & & \vdots & \vdots & \vdots & \vdots \\
a_2 & a_1 & 1 & & \vdots & \vdots & \vdots & \vdots \\
a_3 & & & & \vdots & \vdots & \vdots & \vdots \\
\vdots & & & & & & & \\
a_{k-2} & & & & 0\ \cdots & 0 & 0\ \cdots & 0 \\
a_{k-1} & & & 1 & 0\ \cdots & 0 & b2_0 & \vdots \\
a_k & & & a_1 & 1 & \vdots & b2_1 & \vdots \\
\vdots & & & & \vdots & \vdots & \vdots & \vdots \\
a_n & & & & \vdots & \vdots & \vdots & \vdots \\
 & & & & \vdots & 1 & \vdots & \vdots \\
0 & & & & \vdots & a_1 & b2_{n2} & \vdots \\
\vdots & & & & \vdots & \vdots & 0 & \vdots \\
\vdots & & & & \vdots & \vdots & \vdots & 0 \\
\vdots & & & & \vdots & \vdots & \vdots & b2_0 \\
\vdots & & & & a_n\ \cdots & \cdots & \cdots & b2_1 \\
\vdots & & & & & \vdots & \vdots & \vdots \\
0 & \cdots & \cdots & \cdots & 0 & a_n & 0\ \cdots & b2_{n2}
\end{bmatrix}
\begin{bmatrix}
f_1 \\ f_2 \\ \vdots \\ \vdots \\ \vdots \\ f_{k-1} \\ f_k \\ f_{k+1} \\ \vdots \\ \vdots \\ \vdots \\ f_{k+n2-1} \\ g_0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ g_{n-1}
\end{bmatrix}
=
\begin{bmatrix}
c_1-a_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ c_n-a_n \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0
\end{bmatrix}
$$

<u>Fig. 1</u> - The system of equations for the minimal variance coefficients.

N   is the order of the system,

N2   is the degree of the B2 polynomial,

K   is the time delay in the system,

IN   is a switch from POLPA;
IN=1 if all zeroes of the B polynomial lie inside the circle, =2 if not,

IS   is an indicator from DECOM;
IS is returned =0 if it was possible to solve the identity for F and G. Otherwise IS is returned greater than zero.

The first K-1 unknown elements of the F polynomial may be calculated recursively. If the B polynomial has all zeroes inside the critical circle there are no more unknown elements in the F polynomial. In this case it is also possible to calculate the coefficients of the G polynomial recursively. The calculations are carried out in the subroutine EQ (see 2.3.2). The problems caused by the need for polynomial multiplication are solved with the same technique as will be described in POLMU (see 2.4.1).

However, it is not possible to calculate all elements of the F and G polynomials recursively if the strategy is suboptimal. In this case the result of the recursive equations, which when n2=0 gives the G polynomial, may be used as a part of the right-hand side vector in the system of equations for the remaining n2 elements in the F polynomial and the whole G polynomial. The resulting system of equations is shown in Fig. 2.

$$
\begin{pmatrix}
1 & 0 & \cdots & 0 & b2_0 & \cdots & \cdots & 0 \\
a_1 & & & 0 & b2_1 & b2_0 & & \vdots \\
\vdots & & & \vdots & \vdots & & & \vdots \\
\vdots & & & 1 & \vdots & & & \vdots \\
\vdots & & & a_1 & b2_{n2} & & & \vdots \\
\vdots & & & a_2 & 0 & & & \vdots \\
\vdots & & & \vdots & \vdots & & & b2_0 \\
a_n & & & \vdots & \vdots & & & b2_1 \\
0 & & & \vdots & \vdots & & & \vdots \\
\vdots & & & \vdots & \vdots & & & \vdots \\
0 & \cdots & 0 & a_n & 0 & \cdots & \cdots & b2_{n2}
\end{pmatrix}
\begin{pmatrix}
f_k \\ f_{k+1} \\ \vdots \\ f_{k+n2-1} \\ g_0 \\ \vdots \\ \vdots \\ g_n
\end{pmatrix}
=
$$

$$
= \begin{pmatrix}
c_k \\ c_{k+1} \\ \vdots \\ c_n \\ 0 \\ \vdots \\ \vdots \\ 0
\end{pmatrix}
-
\begin{pmatrix}
(a_k + a_{k-1}f_1 + \ldots + a_1 f_{k-1}) \\
(a_{k+1} + \ldots + a_2 f_{k-1}) \\
\vdots \\
(a_n + \ldots + a_{n-k+1} f_{k-1}) \\
\vdots \\
a_n f_{k-1} \\
0 \\
\vdots \\
0
\end{pmatrix}
$$

Fig. 2 - The system of equations for the coefficients in the suboptimal strategy.

The system of equations is solved by DECOM and SOLVB. An error termination is used in DECOM if the coefficient matrix is singular. The singularity bound is EP=10E-6.

2.3.2. EQ.

The subroutine EQ (EQuation) carries out the recursive calculations in MIVAR. The call is

CALL EQ(C,F,AL,K,NG,NP,IND)

C	is a vector containing the coefficients of the C polynomial,
F	is a vector containing the coefficients of the F polynomial,
AL	is a vector containing the resulting polynomial,
K	is the time delay in the system,
NG	is the number of coefficients to be calculated,
NP	is the starting index value in the C vector,
IND	is a switch for the number of multiplications needed in the calculation of the resulting coefficients.

In this subroutine the C vector has been extended with extra zeroes in order to decrease the code needed. This makes the computation of the G polynomial simpler.

2.4. REGULATOR REPRESENTATION.

The regulator is based upon the F and G polynomials from MIVAR and the B1 polynomial from POLPA. It may be represented both as a weighted sum of old inputs and old outputs (see 2.4.2) and, after a slight modification, as a dynamical system on canonical form (see 2.4.3).

The calculation involves polynomial multiplication which is carried out in POLMU.

2.4.1. POLMU.

POLMU (POLynomial MUltiplication) is a subroutine for multiplication of two real polynomials. It is not assumed that the coefficient of the term with the highest degree is 1.0, neither in the input nor in the output polynomial. The call is

```
CALL POLMU(P1,P2,NP1,NP2,P)
```

P1 and P2	are vectors containing the coefficients of the input polynomials,
NP1 and NP2	are the degrees of P1 and P2,
P	is a vector containing the coefficients of the output polynomial.

This subroutine uses SCAPRO, [2], to multiply real numbers. As SCAPRO does not allow negative steps, one of the polynomials must be rearranged. This is done by forming a polynomial of NP1 leading zeroes followed by the polynomial P2 in reverse order. The multiplications necessary when calculating the i:th coefficient can now be carried out by starting on the first element in P1 and on the (NP1 + NP2 + 2 - i):th element in the extended polynomial. The number of multiplications needed is min(i,NP1+1). Thus the polynomial

$$s^{NP2}P2(1) + \ldots + P2(NP2+1)$$

will be stored in the vector

$$\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ P2(NP2+1) \\ P2(NP2) \\ \vdots \\ P2(2) \\ P2(1) \end{pmatrix} \begin{array}{l} \left.\begin{array}{l} \\ \\ \\ \\ \end{array}\right\} \text{NP1 zeroes} \\ \left.\begin{array}{l} \\ \\ \\ \\ \\ \end{array}\right\} \text{P2 in reverse order} \end{array}$$

The technique is also used when doing the multiplications in EQ (see 2.3.2).

2.4.2. Regulator Based upon Pulse Transfer Function.

The regulator is formed as

$$u = - \frac{G^*}{B1^* F^*} y = - \frac{G^*}{H^*} y$$

(see 1). The coefficient of the highest degree term in H is always 1.0, since the coefficient of the highest degree term in the B polynomial is transferred to the B2 polynomial.

In MIVRE this representation is used when calculating the new process input. A state space representation is not used because of the difficulties arising in finding initial values of the state vector, when the system order has changed.

In MIVCO the F, G and H polynomials are printed out.

2.4.3. Regulator Based upon State Space Equations.

When the regulator is to be represented as a dynamical system on canonical form, the coefficients of the H polynomial must be moved one step downwards in index to give the coefficients of the characteristic polynomial of the regulator. The direct term in the regulator must also be separated. The following formulas are obtained:

direct term:

$$D = - \frac{(G^{*})_1}{(H^{*})_1}$$

coefficients of the characteristic polynomial:

$$(H^{*})_i = (H^{*})_{i+1}$$

elements of the C matrix:

$$(CC^{*})_i = - (G^{*})_{i+1} + (H^{*})_i D$$

In MIVCO this regulator is printed out together with the BB vector, col(1,0,...,0), which thus completes the system S(H,BB,CC,D) on controllable canonical form.

3. APPLICATIONS.

The subroutines described above can be used in on-line control of processes or in off-line computation of minimal variance and dead-beat strategies.

3.1. ON-LINE CONTROL.

If the process parameters are changing, MIVRE can be used for adaptive control. Then MIVRE should be used together with a subroutine for process identification giving current values of the parameters

A, B, C, N, K, IL and IT.

In [2] subroutines for process identification are given (RTLSID, KALID).

If the B polynomial has all zeroes inside the unit circle, it is possible to solve the minimal variance coefficients recursively. This leads to short computing times. When the B polynomial has zeroes outside the unit circle the computing times get considerably longer, because the zeroes of the B polynomial must be computed explicitly and the coefficients can no longer be solved recursively.

If the process parameters sometimes vary so slowly that they may be regarded as constant, the computing time can be decreased considerably. This depends on the fact that the regulator system does not change between the calls on MIVRE. When the new process output is available, the control signal can immediately be calculated by the already existing regulator. This is done by putting IL=2.

The subroutine ROT is able to use the zeroes calculated the time before as initial values for the following iterations. If the process parameters vary slowly the computing time can be decreased by letting ROT start the new iterations using the zeroes of the old B polynomial as initial values. This is done by putting IT=1. It must not be used if the system order N has changed since last call.

The parameter EPS is the error in the result of ROT. There is a possibility to speed up the calculations by allowing lower accuracy. Thus a greater value of EPS will give shorter computing times.

It is possible to let MIVRE control several processes by storing

H, GØ, ZR, ZI, U, Y, NH and NØ

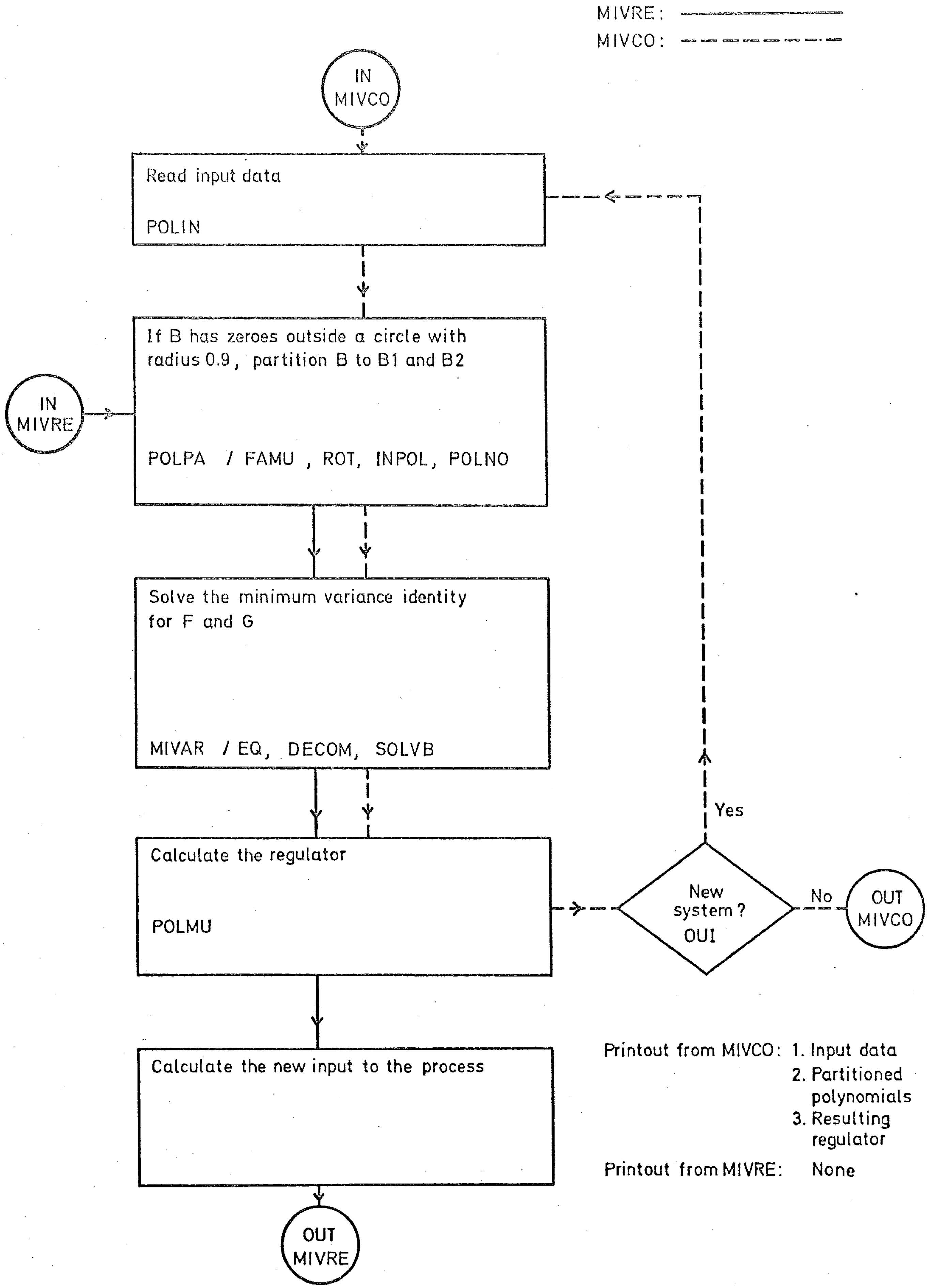
for each process. The stored values of H, GØ, U, Y, NH and NØ make it possible to use the old regulator, if the process parameters have not changed since last call. By storing ZR and ZI the new iterations in ROT can start from the zeroes of the old B polynomial, if a new regulator must be computed.

3.2. OFF-LINE COMPUTATION OF MINIMAL VARIANCE AND DEAD-BEAT STRATEGIES.

If the process parameters are constant, it is enough to compute the regulator system once. This can be done off-line. The regulator can then be implemented as a pulse transfer function or as state space equations on canonical form. In the latter case it is not necessary to store the old values of the process input and output, because the state vector of the regulator will contain all relevant information.

The off-line program MIVCO is of conversational type. It is built up as an executable file. User starts the execution and is then guided by outprints from the program. User is asked about required information, and then the desired strategy is computed by MIVCO. In Appendix IV there is given an example showing how the program is used.

BLOCK DIAGRAM FOR MIVRE AND MIVCO



APPENDIX II

Programs.

```
      SUBROUTINE MIVRE(A,B,C,H,G0,ZR,ZI,U,Y,N,K,NH,NO,IL,IT,EPS)
C
C     GIVEN THE POLYNOMIALS A,B,C DESCRIBING A DYNAMICAL
C     SYSTEM AND AN OUTPUT Y FROM THIS SYSTEM THIS
C     SUBROUTINE CALCULATES A MINIMUM VARIANCE INPUT U
C     TO THE SYSTEM
C
C     AUTHORS:   ULF BORISSON AND JAN HOLST 1971-03-09
C     REFERENCE: K.J.ÅSTRöM INTRODUCTION TO STOCHASTIC
C                           CONTROL THEORY; CHAP. 6
C
C     A-VECTOR OF DIMENSION N+1
C     B-VECTOR OF DIMENSION N+1
C     C-VECTOR OF DIMENSION N+K (FOR COMPUTATIONAL REASONS)
C     ZR,ZI-VECTORS OF DIMENSION N CONTAINING THE REAL AND
C           THE IMAGINARY PARTS OF THE ZEROES OF THE OLD
C           B-POLYNOMIAL.
C     N-ORDER OF THE SYSTEM (MAX 10)
C     K-SYSTEM DELAY (MAX 10)
C
C     IT IS ASSUMED THAT N+N2<15 WHERE N2 IS THE
C     NUMBER OF ZEROES OF THE POLYNOMIAL B WHICH LIE
C     OUTSIDE A CIRCLE WITH RADIUS 0.9.
C
C     IT IS ALSO ASSUMED THAT THE SYSTEM HAS ONLY ONE
C     INPUT AND ONE OUTPUT.
C
C     THE REGULATOR IS REPRESENTED BY ITS PULSE TRANSFER
C     FUNCTION OF ORDER NH.
C
C     Y -VECTOR OF DIMENSION N CONTAINING THE LATEST OUTPUTS
C        FROM THE PROCESS, WHERE Y(1)=Y(T),...,Y(N)=Y(T-N+1).
C     U -VECTOR OF DIMENSION NH+1 CONTAINING THE LATEST INPUTS
C        TO THE PROCESS, WHERE U(1) IS THE NEW INPUT TO THE PROCESS.
C        U(2),...,U(NH+1) ARE THE OLD VALUES OF U(1),...,U(NH).
C     H -CHARACTERISTIC POLYNOMIAL FOR THE REGULATOR SYSTEM
C        VECTOR OF DIMENSION NH.
C     G0-NUMERATOR OF THE REGULATOR SYSTEM
C        VECTOR OF DIMENSION NO.
C
C     IL,IT-INDICATORS FOR THE CALCULATIONS
C           PUT IL=1 IF A,B,C ARE CHANGED SINCE THE LAST CALL
C                 =2 OTHERWISE
C           PUT IT=0 IF ROT IS NOT TO USE THE OLD ZEROES OF THE
C                       B-POLYNOMIAL AS INITIAL VALUES
C                 =1 OTHERWISE
C     EPS  -ACCURACY IN THE COMPUTATION OF THE ZEROES OF B
C           (EPS = 1.0E-8 GIVES HIGHEST POSSIBLE ACCURACY)
C
C     THE WHOLE COMMON BLOCK /SLASK/ IS USED.
C
C     SUBROUTINES REQUIRED
C             POLPA(FAMU,INPOL,POLNO,ROT)
C             MIVAR(DECOM,SOLVB,EQ)
C             POLMU
C
C
      DIMENSION A(1),B(1),C(1),Y(1),U(1),H(1),G0(1),ZR(1),ZI(1)
      DIMENSION B1(11),B2(11),F(20),G(20)
C
```

```
C       PROGRAM MIVCO
C
C       GIVEN THE POLYNOMIALS A,B AND C DESCRIBING A DYNAMICAL
C       SYSTEM THIS PROGRAM CALCULATES THE MINIMUM
C       VARIANCE REGULATOR FOR THE SYSTEM.
C
C       AUTHORS:   JAN HOLST AND ULF BORISSON 1971-03-09
C       REFERENCE: K.J.ÅSTRÖM INTRODUCTION TO STOCHASTIC
C                             CONTROL THEORY; CHAP.6
C
C       A-VECTOR OF DIMENSION N+1
C       B-VECTOR OF DIMENSION N+1
C       C-VECTOR OF DIMENSION N+K (FOR COMPUTATIONAL REASONS)
C       N-ORDER OF THE SYSTEM (MAX 10)
C       K-SYSTEM DELAY (MAX 10)
C
C       IT IS ASSUMED THAT N+N2<15 WHERE N2 IS THE NUMBER
C       OF ZEROES OF THE POLYNOMIAL B WHICH LIE OUTSIDE
C       A CIRCLE WITH RADIUS R.
C
C       THE SYSTEM HAS ONE INPUT AND ONE OUTPUT.
C
C       THE REGULATOR IS GIVEN BOTH AS A WEIGHTED SUM OF OLD
C       INPUTS TO AND OUTPUTS FROM THE SYSTEM AND AS A SYSTEM
C       ON CANONICAL FORM WITH ONE INPUT AND ONE OUTPUT.
C
C       THE WHOLE COMMON BLOCK /SLASK/ IS USED.
C
C       ASSIGN TTA0 6 BEFORE STARTING.
C
C       SUBROUTINES REQUIRED
C               POLPA(FAMU,INPOL,POLNO,ROT)
C               MIVAR(DECOM,SOLVB,EQ)
C               POLMU
C               POLIN
C               OUI
C               RTTFF
C               ATTLP6
C
        LOGICAL L
        DIMENSION A(11),B(11),C(20),T(2)
        DIMENSION B1(11),B2(11),F(20),G(20),H(20),ZR(10),ZI(10)
C
```

```
      SUBROUTINE POLIN(A,N,T)
C
C     THIS SUBROUTINE REEDS A POLYNOMIAL FROM THE TELETYPE
C     ASSUMING THAT THE POLYNOMIAL IS WRITTEN IN FREE FORMAT.
C
C     AUTHORS: JAN HOLST AND ULF BORISSON 1971-03-09
C
C     A -POLYNOMIAL TO BE READ.
C     N -NUMBER OF COEFFICIENTS.
C     T -HOLLERITH CONSTANT CONTAINING THE NAME OF THE POLYNOMIAL
C        (MAX TWO LETTERS).
C
C
C     SUBROUTINES REQUIRED
C             RTTFF
C
C
      DIMENSION A(1)
C
```

```
      SUBROUTINE OUI(TEXTIN,LOGIC)
C
C     TO PRINT OUT THE TWO POSSIBLE ANSWERS TO A QUESTION
C     PROPOSED JUST BEFORE THE CALL.
C
C     AUTHORS: JAN HOLST AND ULF BORISSON 1971-03-09
C
C     TEXTIN-A VECTOR WITH TWO ELEMENTS CONTAINING THE
C            TWO POSSIBLE ANSWERS.
C            EACH ELEMENT MUST BE A HOLLERITH CONSTANT
C            OF MAX 5 LETTERS.
C
C     LOGIC -A LOGICAL VARIABLE WHICH IS RETURNED .TRUE. IF THE
C            ANSWER IS TEXTIN(1),
C            AND .FALSE. IF THE ANSWER IS TEXTIN(2).
C
C     OBSERVE THAT THE ROUTINE IS NOT LEFT UNTIL THE GIVEN
C     ANSWER EQUALS TEXTIN(1) OR TEXTIN(2).
C
C     SUBROUTINES REQUIRED
C             NONE
C
```

```
      SUBROUTINE POLPA(B,B1,B2,ZR,ZI,N,N1,N2,IN,R,IT,EPS)
C
C     PARTITIONS THE REAL POLYNOMIAL
C     B(1)*S**N + B(2)*S**(N-1)+...+B(N+1)
C
C       TO ONE REAL POLYNOMIAL WITH ALL ZEROES INSIDE A CIRCLE WITH
C       RADIUS R:
C                  B1(1)*S**N1 + B1(2)*S**(N1-1)+...+B1(N1+1)
C                  WHERE B1(1)=1.0
C
C       AND ANOTHER REAL POLYNOMIAL WITH ALL ZEROES OUTSIDE OR ON A
C       CIRCLE WITH RADIUS R:
C                              B2(1)*S**N2 + B2(2)*S**(N2-1)+...+B2(N2+1)
C                              WHERE B2(1)=B(1)
C
C     AUTHORS:   ULF BORISSON AND JAN HOLST 1971-03-09
C
C     B -VECTOR OF DIMENSION N+1
C     B1-VECTOR OF DIMENSION N1+1
C     B2-VECTOR OF DIMENSION N2+1
C     ZR,ZI-VECTORS OF DIMENSION N CONTAINING THE REAL AND IMAGINARY
C           PARTS OF THE ZEROES OF THE OLD B-POLYNOMIAL.
C     N -DEGREE OF THE POLYNOMIAL B   (MAX 10)
C     N1-DEGREE OF THE POLYNOMIAL B1
C     N2-DEGREE OF THE POLYNOMIAL B2
C     IN-IS RETURNED IN=1 IF ALL ZEROES OF THE POLYNOMIAL B ARE
C                         INSIDE THE CIRCLE
C                    IN=2 IF THERE IS AT LEAST ONE ZERO OUTSIDE OR ON
C                         THE CIRCLE
C
C     IT -INDICATOR FOR THE CALCULATIONS
C         PUT IT=0 IF ROT IS NOT TO USE THE OLD VALUES OF THE
C                 B-POLYNOMIAL AS INITIAL VALUES
C               =1 OTHERWISE
C     EPS-ACCURACY IN THE COMPUTATION OF THE ZEROES OF B
C         (EPS=1.0E-8 GIVES HIGHEST POSSIBLE ACCURACY)
C
C
C     THE FIELDS DUM7-DUM8 OF THE COMMON BLOCK /SLASK/ ARE USED.
C
C     SUBROUTINES AND FUNCTIONS REQUIRED
C            FAMU
C            INPOL
C            POLNO
C            ROT
C
C
C
      DIMENSION B(1),B1(1),B2(1),ZR(1),ZI(1)
C
      COMMON/SLASK/DUM16(384),DUMY(2),BS(11),DUM(11),Z1R(10),Z1I(10),
     *Z2R(10),Z2I(10),DUMM(64)
C
```

```
      FUNCTION INPOL(A,N,R)
C
C     TESTS IF ALL ZEROES OF THE REAL POLYNOMIAL
C     A(1)*S**N + A(2)*S**(N-1)+...+A(N+1)
C     LIE INSIDE A CIRCLE WITH RADIUS R
C
C     AUTHORS:   ULF BORISSON AND JAN HOLST 1971-03-09
C     REFERENCE: K.J.ÅSTRöM INTRODUCTION TO STOCHASTIC
C                            CONTROL THEORY; CHAP.5
C
C     INPOL-IS RETURNED INPOL=1 IF ALL ZEROES LIE INSIDE THE CIRCLE
C                       INPOL=2 IF THERE IS AT LEAST ONE ZERO OUTSIDE
C                               OR ON THE CIRCLE
C       A  -VECTOR OF DIMENSION N+1 CONTAINING THE COEFFICIENTS OF
C           THE POLYNOMIAL
C       N  -DEGREE OF THE POLYNOMIAL (MAX 10)
C       R  -RADIUS OF THE CIRCLE
C
C     ATTENTION. A ZERO ON THE CIRCLE MAY RETURN INPOL=1 BECAUSE
C                OF THE BINARY REPRESENTATION.
C
C     THE FIELD DUM8 OF THE COMMON BLOCK /SLASK/ IS USED.
C
C     SUBROUTINES REQUIRED
C              NONE
C
C
      DIMENSION A(1)
C
      COMMON/SLASK/DUM17(448),DUM(42),A11(11),A12(11)
C
```

```
      SUBROUTINE FAMU(BR,BI,ZR,ZI,N)
C
C     MULTIPLIES THE COMPLEX FACTORS
C
C     (S-ZR(1)-I*ZI(1))*(S-ZR(2)-I*ZI(2))*...*(S-ZR(N)-I*ZI(N))
C
C     AND GIVES THE RESULTING PRODUCT AS THE COMPLEX POLYNOMIAL
C
C     S**N + (BR(1)+I*BI(1))*S**(N-1)+,..+BR(N)+I*BI(N)
C
C     AUTHORS:   ULF BORISSON AND JAN HOLST 1971-03-09
C
C     BR,BI-VECTORS OF DIMENSION N CONTAINING THE REAL PARTS RESP.
C           IMAGINARY PARTS OF THE POLYNOMIAL
C     ZR,ZI-VECTORS OF DIMENSION N CONTAINING THE REAL PARTS RESP.
C           IMAGINARY PARTS OF THE FACTORS
C     N    -NUMBER OF FACTORS    (MAX 10)
C
C     THE FIELD DUM8 OF THE COMMON BLOCK /SLASK/ IS USED.
C
C     SUBROUTINES REQUIRED
C           NONE
C
      DIMENSION ZR(1),ZI(1),BR(1),BI(1)
C
      COMMON /SLASK/ DUM17(448),DUM(44),SR(10),SI(10)
C
```

```
      SUBROUTINE POLNO(B,NB,B0)
C
C     GIVEN THE POLYNOMIAL
C     B(1)*S**NB+B(2)*S**(NB-1)+...+B(NB+1)
C     THIS SUBROUTINE FORMS THE POLYNOMIAL
C     S**NB+B(1)*S**(NB-1)+...+B(NB)
C
C     HERE B(I)=B(I+1)/B0 AND B0 IS THE OLD VALUE OF B(1).
C
C     AUTHORS: ULF BORISSON AND JAN HOLST 1971-03-09
C
C     B -VECTOR OF DIMENSION NB+1
C     NB-DEGREE OF POLYNOMIAL B.
C
C     SUBROUTINES REQUIRED
C             NONE
C
C
      DIMENSION B(1)
C
```

```
      SUBROUTINE MIVAR(A,B2,C,F,G,N,N2,K,IN,IS)
C
C                                    *    *    *
C     GIVEN THE POLYNOMIALS A  ,B2   ,C
C                            N   N2   N
C     THIS SUBROUTINE CALCULATES THE MINIMUM VARIANCE
C     POLYNOMIALS F AND G FROM THE IDENTITY
C
C      *   * *         -K   *   *
C     C =A F        +Q  B2  G
C      N  N K+N2-1        N2 N-1
C
C     AUTHORS:   JAN HOLST AND ULF BORISSON 1971-03-09
C     REFERENCE: K.J.ÅSTRÖM INTRODUCTION TO STOCHASTIC
C                           CONTROL THEORY; CHAP.6
C
C     A -VECTOR OF DIMENSION N+1
C     B2-VECTOR OF DIMENSION N2+1
C     C -VECTOR OF DIMENSION N+K (FOR COMPUTATIONAL REASONS)
C     F -VECTOR OF DIMENSION K+N2
C     G -VECTOR OF DIMENSION N+1
C     N -DEGREE OF THE POLYNOMIALS A AND C (MAX 10)
C     N2-DEGREE OF THE POLYNOMIAL B2
C     K -SYSTEM DELAY (MAX 10)
C     IN-INDICATOR FROM POLPA
C        PUT IN=1 IF N2=0
C               =2 IF N2>0
C     IS-INDICATOR FROM DECOM
C        IS RETURNED =0 IF IT WAS POSSIBLE TO SOLVE THE
C                       EQ.-SYSTEM FOR F AND G
C        IS RETURNED >0 IF NOT
C
C     IT IS ASSUMED THAT N+N2<15.
C
C     THE WHOLE COMMON BLOCK /SLASK/ IS USED.
C
C     SUBROUTINES REQUIRED
C             DECOM
C             SOLVB
C             EQ
C
C
C
      DIMENSION A(1),B2(1),C(1),F(1),G(1)
C
      COMMON/SLASK/ AM(14,14),DREST(4),X(14,1),BA(14,1),DUM(28),
     1DUM57(192),IDUM(85),I1,I2,NPK,AA(20)
C
```

```
      SUBROUTINE EQ(C,F,AL,K,NG,NP,IND)
C
C     GIVEN THE POLYNOMIALS A AND C THIS SUBROUTINE SOLVS THE
C     MINIMUM VARIANCE COEFFICIENTS F -F     AND THE KNOWN PART IN
C                                    0  K-1
C     EQ.-SYSTEM FOR F -F       AND G -G
C                    K  K+N2-1       0  N-1
C     (IF N2=0 THEN THIS IS G).
C
C     AUTHORS:   ULF BORISSON AND JAN HOLST 1971-03-09
C
C     C  -VECTOR OF DIMENSION N+K (FOR COMPUTATIONAL REASONS)
C     F  -VECTOR OF DIMENSION K+N2
C     AL -RESULTING VECTOR OF DIMENSION NG+1
C     K  -SYSTEM DELAY
C     NG -NUMBER OF COEFFICIENTS TO BE CALCULATED.
C     NP -STARTING VALUE IN C-VECTOR
C     IND-INDICATOR FOR THE NUMBER OF MULTIPLICATIONS IN SCAPRO.
C
C     THE FIELD DUM8 OF THE COMMON BLOCK /SLASK/ IS USED.
C
C     SUBROUTINES REQUIRED
C             NONE
C
C
C
      DIMENSION C(1),F(1),AL(1)
C
      COMMON /SLASK/ DUM17(448),DUM(42),I1,I2,I3,NPK,AA(20)
C
```

```
      SUBROUTINE POLMU(P1,P2,NP1,NP2,P)
C
C     MULTIPLIES THE TWO REAL POLYNOMIALS
C
C     P1(1)*S**NP1+P1(2)*S**(NP1-1)+...+P1(NP1+1)
C     P2(1)*S**NP2+P2(2)*S**(NP2-1)+...+P2(NP2+1)
C
C     TO FORM THE POLYNOMIAL
C
C     P(1)*S**NP+P(2)*S**(NP-1)+...+P(NP+1)
C
C     AUTHORS:   JAN HOLST AND ULF BORISSON 1971-03-09
C
C
C     NP1-DEGREE OF THE POLYNOMIAL P1
C     NP2-DEGREE OF THE POLYNOMIAL P2
C     NP -DEGREE OF THE RESULTING POLYNOMIAL P
C     P1 -VECTOR OF DIMENSION NP1+1
C     P2 -VECTOR OF DIMENSION NP2+1
C     P  -VECTOR OF DIMENSION NP+1
C
C     NP1+NP2 MUST NOT BE GREATER THAN 20
C
C     THE FIELD DUM8 OF THE COMMON BLOCK /SLASK/ IS USED.
C
C     SUBROUTINES REQUIRED
C           NONE
C
C
      DIMENSION P1(1),P2(1),P(1)
      COMMON /SLASK/ DUM17(448),DUM(43),PL(21)
C
```

APPENDIX III

Storage Requirements and Execution Times.

Storage requirements for the routines:

Administration	MIVRE	418
	MIVCO	1565
Polynomial handling	POLPA	435
	INPOL	236
	FAMU	373
	POLNO	97
	POLMU	150
	ROT	1464
Solution of the minimal variance identity	MIVAR	408
	EQ	93
	DECOM	420
	SOLVB	265
Input-Output	POLIN	57
	OUI	91
COMMON	/SLASK/	1024

This means that when using MIVRE for adaptive regulation 5383 cells are needed for these routines. Then code, almost all data and COMMON areas are included. The vectors that are brought into MIVRE as parameters in the subroutine call are not taken into account. MIVCO needs 6678 cells.

However, both MIVRE and MIVCO are well suited for overlay structures, since POLPA (FAMU, INPOL, POLNO, ROT), MIVAR (EQ, DECOM, SOLVB) and POLMU form subprogram blocks. Of the three links mentioned, POLPA will be the largest requiring 2605 cells for the mentioned subprograms.

Average execution times for the subroutines are listed in the table below.

FAMU

Number of factors:	2	4	8
Time in ms:	3.7	14.8	54.1

INPOL

Degree of the polynomial:	2	4	8
Time in ms:	2.6	10.4	30.4

POLMU

Order of the polynomials:	2	4	8
Time in ms:	4.3	8.9	22.9

MIVAR

System order:	3	3	10	3	3	10
Time delay:	2	10	2	2	10	2
Degree of B2:	0	0	0	2	2	4
Time in ms:	4.6	12.7	11.3	72.1	81.2	784.

ROT		Real roots			Compl.conj.roots		
Log (EPS)		-7	-5	-3	-7	-5	-3
Degree of the	2	0.0	0.0	0.0	0.1	0.1	0.0
polynomial	4	0.9	0.7	0.6	0.8	0.6	0.5
(time in sec.)	8	4.0	3.5	3.0	3.4	2.9	2.5
	10	7.9	6.6	5.7	5.5	4.5	3.8

The last table is obtained from the program library [2].

Thus, when not all zeroes of the B polynomial lie inside the critical circle, execution time to a very great extent depends on ROT. In this case the calculation of the new process input takes about one second when the system order is four and about five seconds when the system order is eight.

However, when the B polynomial has all zeroes inside the critical circle the time needed for computing a new process input is considerably diminished. In the table below the average execution time for some different cases is listed.

MIVRE (no zeroes outside the critical circle)

n - system order
k - time delay
time in ms

n \ k	2	4	8
2	13.9		
4	22.9	25.5	
8	48.5	51.9	59.1

With IL=2, i.e. the old regulator is used for computing a new process input, the computing time is 2.8 ms for a system of the order 8 with the time delay 4.

APPENDIX IV

An Example Showing an Off-Line Computation of a Minimal Variance Regulator.

Consider the system

$$y(t) - 1.5y(t-1) + 0.75y(t-2) - 0.125y(t-3) =$$

$$= 2u(t-2) + 13u(t-3) + 22u(t-4) + 8u(t-5) +$$

$$+ e(t) - 1.2e(t-1) + 0.48e(t-2) - 0.064e(t-3)$$

A minimal variance regulator is to be computed. On the teletype is written:

```
KM15 V5A

$A TTAØ 6

$E MIVCO


EXECUTE V4A

THIS IS THE CONVERSATIONAL PROGRAM MIVCO
A CTRL P WILL ALWAYS TAKE YOU BACK TO THIS POSITION
TO START PLEASE STRIKE THE RETURN KEY.


ENTER SYSTEM ORDER AND SYSTEM DELAY.
#3 2

ENTER A -POLYNOMIAL
#1 -1.5 Ø.75 -Ø.125

ENTER B -POLYNOMIAL
#2 13 22 8

ENTER C -POLYNOMIAL
#1 -1.2 Ø.48 -Ø.Ø64
ENTER RADIUS OF CRITICAL CIRCLE.
#1
*OUTPUT ON LP*
*OUTPUT ON TT*
ANY MORE REGULATOR TO BE CALCULATED ?
ANSWER YES   OR NO  :NO

STOP  ØØ22ØØ


KM15 V5A
```

On the lineprinter is written:

```
INPUT TO MIVCO:
***************


SYSTEM ORDER  3        SYSTEM DELAY  2


A -POLYNOMIAL

A ( 1)=  0.1000000E+01
A ( 2)= -0.1500000E+01
A ( 3)=  0.7500000E+00
A ( 4)= -0.1250000E+00

B -POLYNOMIAL

B ( 1)=  0.2000000E+01
B ( 2)=  0.1300000E+02
B ( 3)=  0.2200000E+02
B ( 4)=  0.8000000E+01

C -POLYNOMIAL

C ( 1)=  0.1000000E+01
C ( 2)= -0.1200000E+01
C ( 3)=  0.4800000E+00
C ( 4)= -0.6400000E-01

THE CRITICAL CIRCLE HAS RADIUS   0.1000000E+01

ZEROES INSIDE THE CRITICAL CIRCLE

B1( 1)=  0.1000000E+01
B1( 2)=  0.5000000E+00

ZEROES OUTSIDE THE CRITICAL CIRCLE

B2( 1)=  0.2000000E+01
B2( 2)=  0.1200000E+02
B2( 3)=  0.1600000E+02



OUTPUT FROM MIVCO:
******************



REGULATOR AS A WEIGHTED SUM OF OLD INPUTS TO AND OLD OUTPUTS FROM THE SYST


F -POLYNOMIAL

F ( 1)=  0.1000000E+01
F ( 2)=  0.3000000E+00
F ( 3)=  0.1747789E+00
F ( 4)=  0.7166982E-01

G -POLYNOMIAL

G ( 1)=  0.2610546E-02
G ( 2)= -0.2414003E-02
G ( 3)=  0.5599205E-03
```

```
DENOMINATOR POLYNOMIAL

H ( 1)=  0.1000000E+01
H ( 2)=  0.8000000E+00
H ( 3)=  0.3247789E+00
H ( 4)=  0.1590593E+00
H ( 5)=  0.3583491E-01


REGULATOR AS A DYNAMICAL SYSTEM ON CANONICAL FORM S(H,BB,CC,D)

H -VECTOR

H ( 1)=  0.8000000E+00
H ( 2)=  0.3247789E+00
H ( 3)=  0.1590593E+00
H ( 4)=  0.3583491E-01

BB-VECTOR

BB( 1)=  0.1000000E+01
BB( 2)=  0.0000000E+00
BB( 3)=  0.0000000E+00
BB( 4)=  0.0000000E+00

CC-VECTOR (TRANSPOSED)

CC( 1)=  0.4502440E-02
CC( 2)=  0.2879299E-03
CC( 3)=  0.4152316E-03
CC( 4)=  0.9354869E-04

DIRECT TERM

D ( 1)= -0.2610546E-02

* OUTPUT TO .DAT+6 REROUTED TO ALTERNATE DEVICE *
```

The polynomial

$$B = q^3 B^*(q^{-1}) = 2q^3 + 13q^2 + 22q + 8$$

has the zeroes

$$q = -1/2$$
$$q = -2$$
$$q = -4$$

Thus, there are two zeroes outside the unit circle. A partitioning gives

$$B^* = B1^* \cdot B2^*$$

$$B^*(q^{-1}) = 2 + 13q^{-1} + 22q^{-2} + 8q^{-3}$$

$$B1^*(q^{-1}) = 1 + 0.5q^{-1}$$

$$B2^*(q^{-1}) = 2 + 12q^{-1} + 16q^{-2}$$

In this case MIVCO computes a suboptimal regulator, because all zeroes of the B polynomial are not inside the unit circle. The regulator is given both as a weighted sum of inputs and outputs and as a dynamic system on controllable form. The following results are received from the lineprinter:

$$A^*(q^{-1}) = 1 - 1.5q^{-1} + 0.75q^{-2} - 0.125q^{-3}$$

$$B^*(q^{-1}) = 2 + 13q^{-1} + 22q^{-2} + 8q^{-3}$$

$$C^*(q^{-1}) = 1 - 1.2q^{-1} + 0.48q^{-2} - 0.064q^{-3}$$

$$B1^{*}(q^{-1}) = 1 + 0.5q^{-1}$$

$$B2^{*}(q^{-1}) = 2 + 12q^{-1} + 16q^{-2}$$

$$F^{*}(q^{-1}) = 1 + 0.3q^{-1} + 0.1747789q^{-2} + \\ + 0.7166982 \cdot 10^{-1}q^{-3}$$

$$G^{*}(q^{-1}) = 0.2610546 \cdot 10^{-2} - 0.2414003 \cdot 10^{-2}q^{-1} + \\ + 0.5599205 \cdot 10^{-3}q^{-2}$$

The regulator written as a weighted sum of old inputs and outputs:

$$u(t) = - 0.8u(t-1) - 0.3247789u(t-2) - 0.1590593u(t-3) - \\ - 0.3583491 \cdot 10^{-1}u(t-4) - 0.2610546 \cdot 10^{-2}y(t) + \\ + 0.2414003 \cdot 10^{-2}y(t-1) - 0.5599205 \cdot 10^{-3}y(t-2)$$

The regulator written as a dynamic system on controllable form:

$$x(t+1) =$$

$$= \begin{bmatrix} -0.8 & -0.3247789 & -0.1590593 & -0.3583491 \cdot 10^{-1} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x(t) +$$

$$+ \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} y(t)$$

$$u(t) = [0.4502440 \cdot 10^{-2} \quad 0.2879299 \cdot 10^{-3} \\ 0.4152316 \cdot 10^{-3} \quad 0.9354869 \cdot 10^{-4}] \cdot \\ \cdot x(t) - 0.2610546 \cdot 10^{-2}y(t)$$

REFERENCES.

[1] Åström, K.J.: Introduction to Stochastic Control Theory, Academic Press, New York, 1970.

[2] PDP Program Library, Division of Automatic Control, Lund Institute of Technology, Lund, Sweden.