Regulator Synthesis Based on Polynomial Manipulation

Zhou, Zhao-Ying; Åström, Karl Johan

1981

CODEN: LUTFD2/(TFRT-7225)/1-044/(1981)

REGULATOR SYNTHESIS BASED ON POLYNOMIAL MANIPULATION

Zhou Zhao-ying
Karl Johan Åström

Department of Automatic Control
Lund Institute of Technology
June 1981

LUND INSTITUTE OF TECHNOLOGY
DEPARTMENT OF AUTOMATIC CONTROL
Box 725
S 220 07 Lund 7 Sweden

Document name
Internal report

Date of issue
June 1981

Document number
CODEN: LUTFD2/(TFRT-7225)/1-044/(1981)

Supervisor

Sponsoring organization

Author(s)
Zhou Zhao-ying
Karl Johan Åström

Title and subtitle

Regulator Synthesis Based on Polynomial Manipulation

Abstract

Key words

Classification system and/or index terms (if any)

For single input single output systems many design procedures can conveniently be expressed in terms of polynomial equations. A linear diophantine equation plays a central role for pole placement and LQG design. Algorithms for solving this equation are discussed in the report. A particular algorithm based on a linear transformation and the Euclidean algorithm is found particularly useful. This method has the advantage that the result of transformation can be used in both relative prime polynomials and existance of common factor. A test program is written in Pascal. An example shows the result which is simulated by SIMNON. The comparative study of state space approach and algebraic approach in SISO system is given by a theorem and some examples.

ırough

DOKUMENTDATABLAD RT 3/81

REGULATOR SYNTHESIS

BASED ON POLYNOMIAL MANIPULATION

Zhou Zhao-ying

Karl Johan Åström

Department of Automatic Control

Lund Institute of Technology

June 1981

CONTENT

1. INTRODUCTION

Consider a SISO discrete time system

$$A(z^{-1})\, y(t) = B(z^{-1})\, u(t) \tag{1.1}$$

The characteristic polynomial $P(z^{-1})$ of the desired closed loop system is given directly for pole placement design or indirectly via spectral factorization for linear quadratic control. A general linear feedback can be described by

$$R(z^{-1})\, u(t) = T(z^{-1})\, u_c(t) + S(z^{-1})\, y(t) \tag{1.2}$$

The basic synthesis equation when no cancellation occurs is a linear diophantine equation taken the form of

$$A(z^{-1})\, R(z^{-1}) + B(z^{-1})\, S(z^{-1}) = P_1(z^{-1}) \tag{1.3}$$

where

$P_1(z^{-1}) = P(z^{-1})\, T(z^{-1})$. For the sake of simplicity, the subscript of P is omitted from now on. It is assumed that all polynomials have real coefficients.

The general solution of (1.3) is well known. A particular solution, namely the minimum degree solution, is of interest in practice. From mathematic point of view, the minimum degree solution with respect to R and with respect to S may not necessarily be the same. A degree condition shows that if

$$\deg P <= \deg A + \deg B - 1 \tag{1.4}$$

then the minimum degrees of R and S are well defined. If (1.4) is not satisfied, we chose the solution which gives the minimal degree of S when there is a measurement noise in the system. Otherwise a minimum degree solution for R is chosen. A linear transformation based on the Euclidean algorithm, see Blankinship (1963), is used to find the general solution. An advantage accompanied with the transformation is that it can be used to defeat a common factor in A and B. When there is a common factor this is first cancelled. The synthesis procedure is then applied to the model obtained after cancellation.

Section 2 describes a conventional method for solving the basic synthesis equation which works well in some simple self-tuning regulators, Astrom (1979b), Astrom and Zhou (1981). Section 3 gives some basic relationships of the solution (1.3). An algorithm for a test program is discussed in section 4 in which Kucera's algorithm for solving the diophantine equation is used. An example with the regulator design and the result of simulation by SIMNON are also given. A comparative study of space state approach and algebraic approach is given in section 5.

2. CONVENTIONAL DESIGN METHOD

Degree Condition

Introduce

$$A(z^{-1}) = 1 + a_1 z^{-1} + \dots + a_n z^{-n}$$

$$B(z^{-1}) = b_1 z^{-1} + \dots + b_m z^{-m}$$

$$P(z^{-1}) = 1 + p_1 z^{-1} + \dots + p_\ell z^{-\ell}$$

When the condition (1.4) is satisfied, for existance of an unique minimal degree solution for (1.3) it requires in generic case

$$\left.\begin{aligned} \deg R &= \deg B - 1 \\ \deg S &= \deg A - 1 \end{aligned}\right\} \qquad (2.1)$$

If $\deg P > \deg A + \deg B - 1$ (2.2)

then there are two posible minimum solutions with

$$\left.\begin{aligned} \deg R + \deg A &= \deg P \\ \deg R + \deg S + 1 &= \deg P \end{aligned}\right\}$$

or

$$\left.\begin{aligned} \deg R &= \deg P - \deg A \\ &> \deg B - 1 \\ \deg S &= \deg A - 1 \end{aligned}\right\} \qquad (2.3)$$

which corresponds a minimum solution for $S(z^{-1})$, and

$$\left.\begin{aligned} \deg S + \deg B &= \deg P \\ \deg R + \deg S + 1 &= \deg P \end{aligned}\right\}$$

or

$$\left.\begin{aligned} \deg S &= \deg P - \deg B \\ &> \deg A - 1 \\ \deg R &= \deg B - 1 \end{aligned}\right\} \qquad (2.4)$$

which corresponds a minimum solution for $R(z^{-1})$. After the degrees of $R(z^{-1})$ and $S(z^{-1})$ are determined, equating coefficients of same powers of z from (1.3) yields

$$D\,X = P \tag{2.5}$$

where

$$D = \left[\begin{array}{cccccc} 1 & 0 & \vdots & b_1 & 0 & \vdots \\ a_1 & 1 & & b_2 & b_1 & \\ \vdots & a_1 & & \vdots & b_2 & \\ a_{n-1} & \vdots & & b_{m-1} & \vdots & \\ a_n & a_{n-1} & & b_m & b_{m-1} & \\ 0 & a_n & & 0 & b_m & \\ \vdots & 0 & & \vdots & 0 & \\ 0 & \vdots\;0 & \vdots & 0 & \vdots\;0 & \vdots \end{array}\right]$$

$$\qquad\qquad \deg R \qquad\qquad\qquad \deg S + 1$$

$X = \mathrm{col}\,[\, r_1 \;\; r_2 \; \dots \; r_{degR} \;\; s_0 \;\; s_1 \; \dots \; s_{degS} \,]$

$P = \mathrm{col}\,[\, p_1 - a_1 \;\; p_2 - a_2 \; \dots \; p_\ell - a_\ell \;\; \dots \; 0 \,]$

The design of regulator is then reduced to the solution of a set of linear equations (2.5)

<u>Commom Factor</u>

If a commom factor exists in (1.1), the determinant of D will vanish, i.e.

$$\det D = 0 \qquad \text{if } B_0(z^{-1}) \text{ divides } (\, A(z^{-1}),\, B(z^{-1}) \,) \tag{2.6}$$

This result can be shown by a nonsingular transformation as follows. Assume

$$A(z^{-1}) = (\, 1 + b_0 z^{-1})[\, 1 + a'_1 z^{-1} + \dots + a'_{n-1} z^{-(n-1)}]$$

$$B(z^{-1}) = (\, 1 + b_0 z \quad)[\, 1 + b'_1 z^{-1} + \dots + b'_{m-1} z^{-(m-1)}]$$

Then

$$D = \begin{bmatrix} 1 & 0 & \vdots & b'_1 & 0 & \vdots \\ b_0+a'_1 & 1 & & b_0b'_1+b'_2 & b'_1 & \\ \vdots & b_0+a'_1 & & \vdots & b_0b'_1+b'_2 & \\ b_0a'_{n-2}+a'_{n-1} & \vdots & & b_0b'_{m-2}+b'_{m-1} & \vdots & \\ b_0a'_{n-1} & b_0a'_{n-2}+a'_{n-1} & & b_0b'_{n-1} & b_0b'_{m-2}+b'_{m-1} & \\ 0 & b_0a'_{n-1} & & 0 & b_0b'_{m-1} & \\ \vdots & 0 & & \vdots & 0 & \\ 0 & \vdots \; 0 & \vdots & 0 & \vdots \; 0 & \vdots \end{bmatrix}$$

A nonsingular matrix T can be found

$$T = \begin{bmatrix} 1 & & & & \\ -1 & 1/b_0 & & 0 & \\ 1 & -1/b_0 & 1/b_0^2 & & \\ \cdots & & & & \\ (-1)^{k+1} & (-1)^k/b_0 & (-1)^{k+1}/b_0^2 & \cdots & 1/b_0^{k-1} \end{bmatrix} \tag{2.7}$$

where $k = \deg R + \deg S + 1$, such that

$$T\,D = D_1$$

where D_1 has a null row at its last row.

$$\det D_1 = \det T \det D$$
$$= 0$$

Hence (2.3) follows because $\det T \neq 0$.

EXAMPLE 1.

A simple self-tuning regulator controls a first order process with a lag and a time delay.

$$\frac{B(s)}{A(s)} = \frac{K\,e^{-\tau s}}{Ts+1}$$

The pulse transfer function of the process is given by

$$\frac{B(z^{-1})}{A(z^{-1})} = \frac{z^{-k}(b_1 + b_2 z^{-1})}{1 + a\, z^{-1}}$$

where $k = \tau \text{ div } h + 1$, see appendix A. Consider the case of $h > \tau$, i.e. $k=1$. A desired characteristic equation of the closed loop system is

$$P(z^{-1}) = 1 + p_1 z^{-1}$$

The basic synthesis equation (1.3) for a minimum degree solution is

$$(1 + az^{-1})(1 + r_1 z^{-1}) + (b_1 z^{-1} + b_2 z^{-2})\, s_0 = 1 + p_1 z^{-1}$$

Then

$$r_1 = \frac{b_2(p_1 - 1)}{b_2 - a\, b_1}$$

$$s_0 = \frac{a\,(p_1 - 1)}{b_2 - a\, b_1}$$

There is a common factor if

$$b_2 = a\, b_1$$

If an integral action is added to the process, then

$$\frac{B(z^{-1})}{A(z^{-1})} = \frac{b_1 z^{-1} + b_2 z^{-2}}{(1 + a\, z^{-1})(1 - z^{-1})}$$

If an observer polynomial $(1 + t_1 z^{-1})$ is also chosen, (1.3) becomes

$$(1 + a\, z^{-1})(1 - z^{-1})(1 + r_1 z^{-1}) + (b_1 z^{-1} + b_2 z^{-2})(s_0 + s_1 z^{-1})$$

$$= (1 + p_1 z^{-1})(1 + t_1 z^{-1})$$

The solution is obtained from (2.5) , i.e. from

$$\begin{bmatrix} 1 & b_1 & 0 \\ a-1 & b_2 & b_1 \\ -a & 0 & b_2 \end{bmatrix} \begin{bmatrix} r_1 \\ s_0 \\ s_1 \end{bmatrix} = \begin{bmatrix} p_1 + t_1 - a + 1 \\ p_1 t_1 + a \\ 0 \end{bmatrix}$$

The commom factor condition is given by

$$\det D = (b_2 - a\, b_1)(b_1 + b_2)$$
$$= 0$$

□

EXAMPLE 2.

Consider a control problem described by the polynomials

$$A(z^{-1}) = 1 + 1.5z^{-1} + 0.5z^{-2}$$
$$B(z^{-1}) = z^{-1} + 0.8z^{-2}$$

and

$$P(z^{-1}) = 1 + 0.6z^{-1} + 0.08z^{-2}$$

They satisfy the degree condition (1.4), therefore the problem has an unique minimum degree solution with

$$R(z^{-1}) = 1 + 4z^{-1}$$
$$S(z^{-1}) = -4.9 -2.5z^{-1}$$

To illustrate what may happen if the parameters of the system vary, assume that B becomes

$$B(z^{-1}) = z^{-1} + 0.5z^{-2}$$

It is clear that a conmom factor exists. After cancelling the stable factor, we have

$$A_1(z^{-1}) = 1 + z^{-1}$$
$$B_1(z^{-1}) = z^{-1}$$

The degree condition changes to (2.2) and the problem how has two minimum degree solutions

$$R_1(z^{-1}) = 1$$
$$S_1(z^{-1}) = -0.4 + 0.08z^{-1}$$

and

$$R_2(z^{-1}) = 1 + 0.08z^{-1}$$
$$S_2(z^{-1}) = -0.48$$

□

Example 2 illustrates what may happen in an adaptive control problem. Assume that an adaptive regulator is set up in such a way that deg R = deg S = 1. There will then normally be an unique minimum degree solution to the design problem. For parameters such that there is a pole-zero cancellation there are however no unique minimum degree solution. Special rules must therefore be introduced to choose one of the possible minimum degree solution.

3. BASIC RELATIONSHIPS IN THE DIOPHANTINE EQUATION

In this section we discuss the general and particular solution of the diophantine equation (1.3), and common factors all of which are based on a linear transformation.

The General Solution

Consider a matrix

$$S = \begin{bmatrix} A & 1 & 0 \\ B & 0 & 1 \end{bmatrix} \tag{3.1}$$

where A, B, 1 and 0 are row vectors.

$$A = [\ 1\ ,\ a_1 ,\ \ldots\ ,\ a_n\ ]$$
$$B = [\ 0\ ,\ b_1 ,\ \ldots\ ,\ b_m\ ]$$
$$1 = [\ 1\ ,\ 0\ ,\ \ldots\ ,\ 0\ ]$$
$$0 = [\ 0\ ,\ 0\ ,\ \ldots\ ,\ 0\ ]$$

A linear transformation T gives

$$T\ S = \begin{bmatrix} 1 & E & F \\ 0 & G & H \end{bmatrix} \tag{3.2}$$

which corresponds to the polynomial relations

$$\left.\begin{aligned} A(z^{-1})\ E(z^{-1}) + B(z^{-1})\ F(z^{-1}) &= 1 \\ A(z^{-1})\ G(z^{-1}) + B(z^{-1})\ H(z^{-1}) &= 0 \end{aligned}\right\} \tag{3.3}$$

The Euclidean algorithm is applied to solve the so-called monic greatest common divisor problem. From (3.3) it is follows that in the generic case without common factor

$$\left.\begin{aligned} \deg E &= \deg B - 1 \\ \deg F &= \deg A - 1 \\ \deg G &= \deg B \\ \deg H &= \deg A \end{aligned}\right\} \tag{3.4}$$

with $e_0 = 1$, $g_0 = 0$, then the general solution of (1.3) has the form of

$$\left.\begin{aligned} R(z^{-1}) &= E(z^{-1})\ P(z^{-1}) + G(z^{-1})\ V(z^{-1}) \\ S(z^{-1}) &= F(z^{-1})\ P(z^{-1}) + H(z^{-1})\ V(z^{-1}) \end{aligned}\right\} \tag{3.5}$$

where $V(z^{-1})$ is an arbitrary polynomial used for search the minimum degree solution. See Kucera (1979).

Particular Solutions

Using the general solution (3.5) and the degree condition (2.2), (2.3) or (2.4) the arbitrary polynomial $V(z^{-1})$ as well as the regulator polynomial $R(z^{-1})$ and $S(z^{-1})$ can be determined easily. For instance, with the condition (2.3) it follows that

$$\deg V = \deg P - 1 \tag{3.6}$$

Introduce

$$R(z^{-1}) = r_0 + r_1 z^{-1} + \dots + r_{m-1} z^{-(m-1)}$$

$$S(z^{-1}) = s_0 + s_1 z^{-1} + \dots + s_{n-1} z^{-(n-1)}$$

$$V(z^{-1}) = v_0 + v_1 z^{-1} + \dots + v_{\ell-1} z^{-(\ell-1)}$$

$$E(z^{-1}) = e_0 + e_1 z^{-1} + \dots + e_{m-1} z^{-(m-1)}$$

$$F(z^{-1}) = f_0 + f_1 z^{-1} + \dots + f_{n-1} z^{-(n-1)}$$

$$G(z^{-1}) = g_0 + g_1 z^{-1} + \dots + g_m z^{-m}$$

$$H(z^{-1}) = h_0 + h_1 z^{-1} + \dots + h_n z^{-n}$$

From (2.1) we have

$$v_{l-i} = -\left(\sum_{j=0}^{i-1} p_{\ell-j} e_{m-i+j} + \sum_{k=1}^{i-1} g_{m-k} v_{\ell-i+k} \right)/g_m$$

$$= -\left(\sum_{j=0}^{i-1} p_{\ell-j} f_{n-i+j} + \sum_{k=1}^{i-1} h_{n-k} v_{\ell-i+k} \right)/h_n \qquad \text{for } i=1,2,\dots,l$$

$$r_i = \sum_{j=0}^{i} (p_{i-j} e_j + g_{i-j} v_j) \qquad \text{for } i=0,1,\dots,m-1$$

$$s_i = \sum_{j=0}^{i} (p_{i-j} f_j + h_{i-j} v_j) \qquad \text{for } i=0,1,\dots,n-1$$

$$(3.7)$$

In general case, the polynomial division can be used to calculate $V(z^{-1})$, $R(z^{-1})$ and $S(z^{-1})$. Consider

$$V_1(z^{-1}) = - [\, E(z^{-1})\ P(z^{-1}) \bmod G(z^{-1})\,] \qquad (3.8)$$

$$V_2(z^{-1}) = - [\, F(z^{-1})\ P(z^{-1}) \bmod H(z^{-1})\,] \qquad (3.9)$$

$$R(z^{-1}) = E(z^{-1})\ P(z^{-1}) \operatorname{div} G(z^{-1}) \qquad (3.10)$$

$$S(z^{-1}) = F(z^{-1})\ P(z^{-1}) \operatorname{div} H(z^{-1}) \qquad (3.11)$$

If (1.4) is hold, then

$$V_1(z^{-1}) = V_2(z^{-1})$$

Hence $R(z^{-1})$ and $S(z^{-1})$ are determined uniquely. When (2.3) is chosen, (3.9) and (3.11) are used and $R(z^{-1})$ is given by substitution. When (2.4) is chosen, (3.8) and (3.10) are computed and $S(z^{-1})$ is from substitution.

<u>Common Factor</u>

Assume that there is a common factor in $A(z^{-1})$ and $B(z^{-1})$, i.e.

$$A(z^{-1}) = A_1(z^{-1})\ B_0(z^{-1})$$

$$B(z^{-1}) = B_1(z^{-1})\ B_0(z^{-1}) \qquad (3.12)$$

The linear transformation gives

$$\begin{bmatrix} E_1 & F_1 \\ G_1 & H_1 \end{bmatrix} \begin{bmatrix} A_1B_0 & 1 & 0 \\ B_1B_0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} B_0 & E_1 & F_1 \\ 0 & G_1 & H_1 \end{bmatrix}$$

with

$$\left.\begin{aligned} A(z^{-1})\ E_1(z^{-1}) + B(z^{-1})\ F_1(z^{-1}) &= B_0(z^{-1}) \\ A(z^{-1})\ G_1(z^{-1}) + B(z^{-1})\ H_1(z^{-1}) &= 0 \end{aligned}\right\} \qquad (3.9)$$

After cancelling common factor from $A(z^{-1})$ and $B(z^{-1})$, we have

$$\left.\begin{aligned} A_1(z^{-1})\, E_1(z^{-1}) + B_1(z^{-1})\, F_1(z^{-1}) &= 1 \\ A_1(z^{-1})\, G_1(z^{-1}) + B_1(z^{-1})\, H_1(z^{-1}) &= 0 \end{aligned}\right\} \qquad (3.10)$$

That means $E_1(z^{-1})$, $F_1(z^{-1})$, $G_1(z^{-1})$ and $H_1(z^{-1})$ are the solution corresponding to $A_1(z^{-1})$ and $B_1(z^{-1})$ which are now ralatively prime. The degrees of these elementary polynomial reduce to

$$\deg E_1 = \deg B - \deg B_0 - 1$$
$$\deg F_1 = \deg A - \deg B_0 - 1$$
$$\deg G_1 = \deg B - \deg B_0$$
$$\deg H_1 = \deg A - \deg B_0$$

Because the Euclidean algorithm gives the greatest common factor and the accociated transformation gives the solution corresponding to the polynomials $A_1(z^{-1})$ and $B_1(z^{-1})$, which are obtained after cancelling the common factor $B_0(z^{-1})$. the solution is given by (3.8) to (3.11).

EXAMPLE 3.

Given

$$A(z^{-1}) = (1 + 3z^{-1} + 2z^{-2})(1 + 0.5z^{-1})$$
$$= 1 + 3.5z^{-1} + 3.5z^{-2} + z^{-3}$$
$$B(z^{-1}) = (z^{-1} + 0.8z^{-2})(1 + 0.5z^{-1})$$
$$= z^{-1} + 1.3z^{-2} + 0.4z^{-3}$$
$$P(z^{-1}) = 1 + 0.6z^{-1} + 0.08z^{-2}$$

Find a causal solution for the system.

The linear transformation is performed by use of the Blankinship's calculation table

row	op (*)	od (**)	q	A (B)				E (G)				F (H)			
1				1	3.5	3.5	1	1	0	0	0	0	0	0	0
2				0	1	1.3	0.4	0	0	0	0	1	0	0	0
3	2	1	2.5	1	1	0.25	0	1	0	0	0	-1/4	0	0	0
4	3	2	1.6	0	-0.6	-0.3	0	0	-1.6	0	0	1	4	0	0
5	3	4	-1.2	1.2	0.6	0	0	1.2	-1.6	0	0	-2	4	0	0
6	5	3	5/12	1	0.5	0	0	1	-0.5	2/3	0	-2.5	5/6	5/3	0
7	5	6	5/6	0	0	0	0	0	5/6	2/3	0	-5/6	-2.5	-5/3	0
8		5	1.2	1	0.5	0	0	1	-4/3	0	0	-5/3	10/3	0	0

where

(*) op -- operator,

(**) od -- operand.

which gives

$$E(z^{-1}) = 1 - 4/3\ z^{-1}$$

$$F(z^{-1}) = -5/3 + 10/3\ z^{-1}$$

$$G(z^{-1}) = -5/6\ z^{-1} + 2/3\ z^{-2}$$

$$H(z^{-1}) = -5/6 - 2.5\ z^{-1} - 5/3\ z^{-2}$$

and common factor

$$B_0(z^{-1}) = A(z^{-1})\ E(z^{-1}) + B(z^{-1})\ F(z^{-1})$$

$$= 1 + 0.5z^{-1}$$

Cancelling the common factor from $A(z^{-1})$ and $B(z^{-1})$ yields

$$A_1(z^{-1}) = 1 + 3z^{-1} + 2z^{-2}$$

$$B_1(z^{-1}) = z^{-1} + 0.8z^{-2}$$

The unique solution of (3.7) is found by

$$V(z^{-1}) = 0.88 + 0.16z^{-1}$$

$$R(z^{-1}) = 1$$

$$S(z^{-1}) = -2.4$$

□

4. A TEST PROGRAM

A test program for regulator design is organized as follows.

<u>Step 1</u>. Given $A(z^{-1})$, $B(z^{-1})$ and $P(z^{-1})$.

For LQC, $P(z^{-1})$ is given by spectral factorization.

<u>Step 2</u>. Perform linear transformation (3.2) to find the greatest common factor, which is based on Kucera's program of monic greatest common divisor. If $B(z_0^{-1}) = 1$ then there there is no common factor in $A(z^{-1})$ and $B(z^{-1})$. Then go to step 4.

<u>Step 3</u>. If common factor exists, cancel it by use of a division algorithm.

<u>Step 4</u>. Obtain the particular solution (3.8) - (3.11) where the elementary polynomial $E(z^{-1})$, $F(z^{-1})$, $G(z^{-1})$ and $H(z^{-1})$ found in step 2 are also valid for step 3.

The computation process has been illustrated in example 3. Another example is shown not only by the test program but also by the simulation SIMNON.

EXAMPLE 4.

Consider a second order system with a time delay

$$\frac{B(s)}{A(s)} = \frac{2\, e^{-0.5s}}{(s+1)(s-2)}$$

Design a linear regulator such that the criterion

$$J = \sum_{k=0}^{\infty} [\, y^2(k) + \varrho\, u^2(k)\,] = \min$$

1) Use formulae in appendix A to obtain the discrete time description of the system when h=0.4

$$\frac{B(z^{-1})}{A(z^{-1})} = \frac{0.101z^{-1} + 0.288z^{-2} + 0.014z^{-3}}{1 - 2.896z^{-1} + 1.492z^{-2}}$$

In a self-tuning system the model of pulse fransfer function is obtained by on line parameter identification.

2) The linear quadratic criterion implies that $P(z)$ is given by the spectral factorization.

$$P(z)P(z^{-1}) = \varrho\, A(z)A(z^{-1}) + B(z)B(z^{-1})$$

Using the algorithm in Zhou and Astrom (1981) the result is

ϱ	p_0	p_1	p_2
0	0.2831	0.1150	0.0050
1	2.3434	-2.4101	0.6373
10	7.0827	-7.8498	2.1067
25	11.1568	-12.4404	3.3434

3) Linear transformation (3.2) shows that there is no common factor in $A(z^{-1})$ and $B(z^{-1})$. The particular solution is

ϱ	r_0	r_1	r_2	s_0	s_1
0	0.2831	0.4959	0.0238	4.3458	-2.5407
1	2.3434	2.4706	0.1176	18.8688	-12.5359
10	7.0827	7.1741	0.3413	54.3331	-36.3743
25	11.1568	11.2622	0.5358	85.2226	-57.0996

The test program RDP compute the values in 3).

4) The simulation of $\varrho=0$, 1, and 25 is made by SIMNON. See Appendix C. The result shows when ϱ is larger than a certain value, the transient behaviour will only change slightly with respect to the change of ϱ.

□

5. COMPARATIVE STUDY OF STATE SPACE APPROACH AND ALGEBRAIC APPROACH IN SISO SYSTEM

<u>Therom 5.1</u>

Consider a SISO system described by (1.1). If the criterion is given by

$$J = \sum_{k=0}^{T} [\, \varrho_1 y^2(k) + 2\varrho_{12} y(k)\, u(k) + \varrho_2 u^2(k) \,] \qquad (5.1)$$

then the solution of this optimazation problem gives a closed loop system whose characteristic polynomial $P(z^{-1})$ is given by

$$P(z)P(z^{-1}) = \varrho_1 B(z)B(z^{-1}) + \varrho_2 A(z)A(z^{-1}) + $$
$$+ \varrho_{12} [\, B(z)A(z^{-1}) + A(z)B(z^{-1}) \,] \qquad (5.2)$$

Proof:

After application of z-transformation to the canonical equation, Astrom(1963), see Appendix B, the characteristic equation of the desired closed loop system in steady state is obtained.

$$P(s)P(z^{-1}) = \det \begin{bmatrix} zI - \Phi + \Gamma Q_{22}^{-1} Q_{21} & \Gamma Q_{22}^{-1} \Gamma^T \\ - Q_{11} + Q_{12} Q_{22}^{-1} Q_{21} & z^{-1} I - \Phi + (\Gamma Q_{22}^{-1} Q_{21})^T \end{bmatrix}$$

$$= \det(zI-\Phi)\det(z^{-1}I-\Phi^T)\det[\, I + Q_{22}^{-1}\Gamma^T(z^{-1}I-\Phi^T)^{-1}Q_{11}(zI-\Phi)^{-1}\Gamma$$
$$+ Q_{22}^{-1}\Gamma^T(z^{-1}I-\Phi^T)^{-1}Q_{21}^T) + Q_{22}^{-1}Q_{21}(zI-\Phi)^{-1}\Gamma \,] \qquad (5.3)$$

Substituting

$$\det(zI-\Phi) = A(z)$$

$$\det[(zI-\Phi)^{-1}\Gamma] = \frac{B(z)}{A(z)}$$

$$Q_{11} = \varrho_1$$

$$Q_{22} = \varrho_2$$

$$Q_{12} = Q_{21} = \varrho_{12}$$

into (5.3), the result (5.2) follows.

<u>Corollary 1</u>

The simplest case of (5.1) is $\varrho_1=1$, $\varrho_{12}=0$ and $\varrho_2=\varrho$, we have

$$P(z)P(z^{-1}) = \varrho\, A(z)A(z^{-1}) + B(z)B(z^{-1}) \tag{5.4}$$

Example 5.

Consider a system

$$x(k+1) = x(k) + u(k) + e(k)$$

and the loss function

$$J = \sum_{k=1}^{\infty} [\, x^2(k) + u^2(k) \,]$$

The solution from the optimal control theory

$$u(k) = -L(k)\; X(k)$$

$$L(k) = \frac{S(k+1)}{1 + S(k+1)}$$

$$S(k) = S(k+1) + 1 - \frac{S^2(k+1)}{1 + S(k+1)}$$

Let $S(k) = S(k+1)$, we find the steady-state optimal control law with

$$S(\infty) = (1 + \sqrt{5}) / 2$$

$$L(\infty) = \frac{1 + \sqrt{5}}{3 + \sqrt{5}} = 0.618$$

The algebraic approach gives the same result. This is shown as follows. The polynomial spectral factorization gives

$$P(z)\; P(z^{-1}) = (1 - z)(1 - z^{-1}) + 1$$

$$= (1.618 - 0.618z)(1.618 - 0.618z^{-1})$$

The basic synthesis equation becomes

$$(1 - z^{-1})\, r_0 + z^{-1} s_0 = p_0 + p_1 z^{-1}$$

with

$$r_0 = 1.618$$

$$s_0 = 1$$

The linear regulator (1.3) is

$$u(k) = - s_0 x(k) / r_0$$
$$= - 0.618\ x(k)$$

which is the same result as obtained before. □

Corollary 2

Consider a process

$$A\ y = B\ u$$

$$A = \prod_{i=1}^{n} A_i$$

and

$$B = \prod_{i=1}^{n} B_i$$

The criterion is given by

$$J = \sum_{t=0}^{\infty} (\varrho_n y_n^2 + \varrho_{n-1} y_{n-1}^2 + \ldots + \varrho_2 y_2^2 + \varrho_1 y_1^2 + \varrho_0 u^2) \tag{5.6}$$

See Fig. 1. It corresponds a spectral factorization problem as follows.

$$\varrho_0 A(z)\ A(z^{-1}) + \varrho_n B(z)\ B(z^{-1}) +$$
$$+ \varrho_{n-1} A_n(z)\ B_{n-1}(z)\ ..\ B_1(z)\ B_1(z^{-1})\ ..\ B_{n-1}(z^{-1})\ A_n(z^{-1})$$
$$+ \ldots + \varrho_1 A_n(z)\ ..\ A_2(z)\ B_1(z)\ B_1(z^{-1})\ A_2(z^{-1})\ ..\ A_n(z^{-1})$$
$$= P(z)\ P(z^{-1}) \tag{5.7}$$

Example 6.

Consider a temperature control process which is studied by S.E.Mattsson using a state space approach.

$$y(k) - 1.664\ y(k-1) + 0.683\ y(k-2) = 0.0488\ u(k-4) +$$
$$+ 0.0042\ u(k-5) + v(k)$$

where disturbance $\{ v(k),\ k \in T \}$ is a white noise.

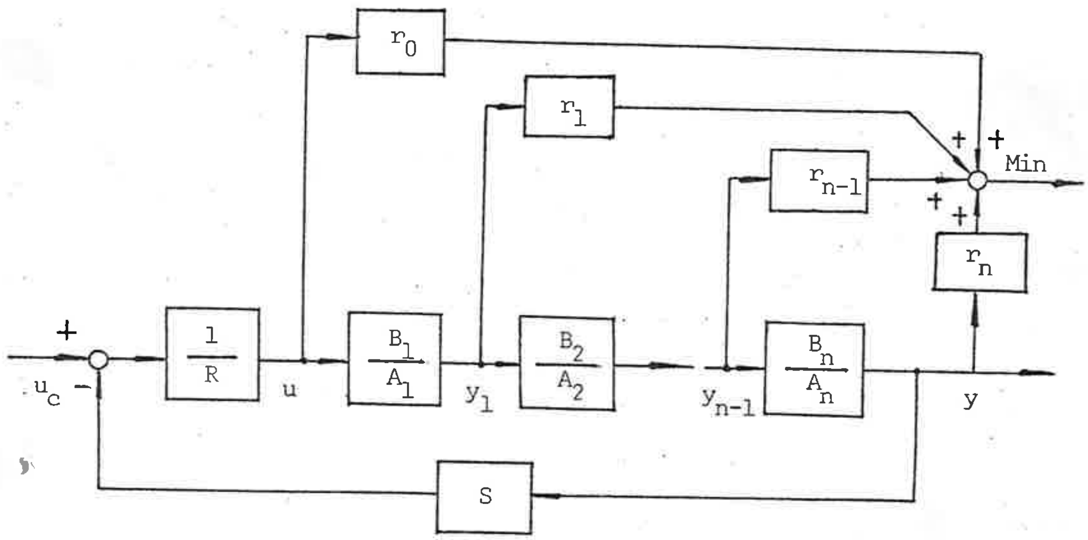


Fig. 1. A process in corollary 2.

Let

$$x_1(k) = y(k)$$

$$X_2(k) = -0.683\ y(k-1) + 0.0042\ u(k-4)$$

$$x_3(k) = u(k-3)$$

$$x_4(k) = u(k-2)$$

$$x_5(k) = u(k-1)$$

and introduce an integrator

$$x_0(k) = \frac{1}{1 - z^{-1}}\ x_1(k)$$

The state variable difference equation of the process is

$$X(k+1) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1.664 & 1 & 0.0488 & 0 & 0 \\ 0 & -0.683 & 0 & 0.0042 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} X(k) + \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u(k) \\ v(k) \end{bmatrix}$$

$$y(k) = [\,0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0\,]\ X(k)$$

The loss function is

$$J = \sum_{k=0}^{\infty} [\, X^T(k)\, Q_1 X(k) + u^T(k)\, Q_2 u(k) \,]$$

with

$$Q_1 = \mathrm{diag}\, [\, 5 \;\; 0.1 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \,] \quad \text{and}$$

$$Q_2 = 1$$

The optimal control strategy is given by

$$u(k) = - L(k)\; X(k)$$

After solving a matrix Riccati difference equation, the following solution is obtained

$$u(k) = - [\, 1.55 \;\; 25.04 \;\; 21.29 \;\; 1.11 \;\; 0.91 \;\; 0.71 \,]\; X(k)$$

$$= - \frac{(26.59 - 39.56z^{-1} + 14.54z^{-2})}{(1-z^{-1})(1+0.70z^{-1}+0.91z^{-2}+1.11z^{-3}+0.09z^{-4})}\; y(k)$$

The algebraic method is used as an alternative approach. Consider the process in a closed loop form, Fig. 2.

where

$$A(z^{-1}) = 1 - 1.664z^{-1} + 0.683z^{-2}$$

$$B(z^{-1}) = 0.0488z^{-4} + 0.0042z^{-5}$$

The loss function can be reformed as a spectral factorization problem.

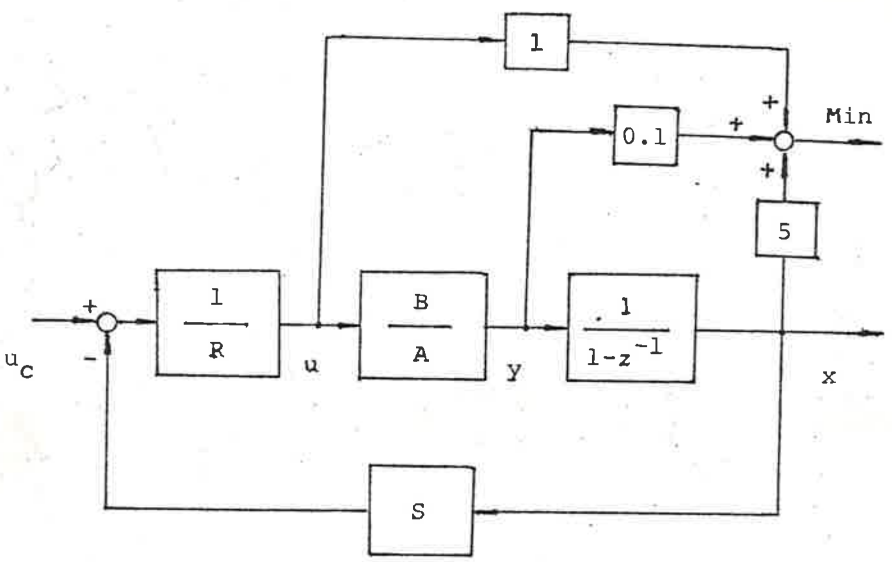


Fig. 2. A process in example 6.

$$(1-z)\ A(z)\ A(z^{-1})\ (1-z^{-1}) + 5\ B(z)\ B(z^{-1}) +$$
$$+ 0.1\ (1-z)\ B(z)\ B(z^{-1})\ (1-z^{-1}) = P(z)\ P(z^{-1})$$

with

$$P(z^{-1}) = 1.44 - 2.814z^{-1} + 1.967z^{-2} - 0.474z^{-3}$$

The diophantine equation (1.3) has the form of

$$(1-z^{-1})\ A(z^{-1})\ R(z^{-1}) + B(z^{-1})\ S(z^{-1}) = P(z^{-1})$$

Because the degree condition (1.4) is satisfied, it has an unique minimal degree solotion. This is obtained as follows

$$R(z^{-1}) = 1.44 + 1.022z^{-1} + 1.31z^{-2} + 1.6z^{-3} + 0.129z^{-4}$$

$$S(z^{-1}) = 36.06 - 54.75z^{-1} + 20.93z^{-2}$$

$$u(k) = - \frac{S(z^{-1})}{R(z^{-1})}\ x(k)$$

$$= - \frac{S(z^{-1})}{(1-z^{-1})\ R(z^{-1})}\ y(k)$$

$$= - \frac{(25.04-38.02z^{-1}+14.54z^{-2})}{(1-z^{-1})(1+0.71z^{-1}+0.91z^{-2}+1.11z^{-3}+0.09z^{-4})}\ y(k)$$

The control law is the same as the one obtained by the state space approach.

□

Corollary 3

Consider the same process as in Corollary 2 with

$$J = \sum_{t=0}^{T} [\ \varrho_n y_n^2 + \dots + \varrho_1 y_1^2 + \varrho_0 u^2 +$$
$$+ 2\varrho_{n,n-1} y_n y_{n-1} + \dots + 2\varrho_{1,0} y_1 u\ ] \qquad (5.8)$$

The corresponding spectral factorization problem is

$$\varrho_n BB^* + \dots + \varrho_1 (A_1 .. A_n B_2 B_1)(B^*_1 A^*_2 .. A^*_n) + \varrho_0 AA^* +$$
$$+ \varrho_{n,n-1} (B_n .. B_1 B^*_1 .. B^*_{n-1} A^*_n + A_n B_{n-1} .. B_1 B^*_1 .. B^*_n) + \dots +$$
$$+ \varrho_{1,0} (A_n .. A_2 B_1 A^*_1 .. A^*_n + A_n .. A_1 B^*_1 A^*_2 .. A^*_n) = PP^* \qquad (5.9)$$

where

$$A^* = A_n(z^{-1})\ A_{n-1}(z^{-1})\ ..\ A_1(z^{-1})$$

$$B^* = B_n(z^{-1})\ B_{n-1}(z^{-1})\ ..\ B_1(z^{-1})$$

Example 7.

Consider a first order system

$$\dot{x} = u$$

with the optimization criterion

$$J = q_0 x^2(t) + \int_0^T [\ q_1 x^2(t) + q_2 u^2(t)\ ]\ dt$$

Then the discrete version is given by Astrom(1963), see Appendix B.

$$x(k+1) = x(k) + hu(k)$$

$$J = q_0 x^2(t) + \sum_0^{T-1} [\ q_{11} x^2(k) + 2q_{12} x(k)u(k) + q_{22} u^2(k)\ ]$$

where

$$q_{11} = q_1 h$$

$$q_{12} = \frac{1}{2}\ q_1 h^2$$

$$q_{22} = q_2 h + \frac{1}{3}\ q_1 h^3$$

The optimal control law is

$$u(k) = -\ L(k)\ x(k)$$

$$s(k) = s(k+1) + q_{11} - \frac{[\ h\ s(k+1) + q_{12}\ ]^2}{h^2 s(k+1) + q_{22}}$$

$$s(T) = q_0$$

The steady solution of s(k) is obtained

$$s(\infty) = \sqrt{q_1 q_2}\ \sqrt{1 + \frac{q_1}{12 q_2}\ h^2}$$

and

$$L = \frac{h\, s(\infty)\, q_{12}}{h^2 s(\infty) + q_{22}}$$

$$= \frac{0.5 q_1 h^2 + h\sqrt{q_1 q_2}\sqrt{1 + \frac{q_1}{12 q_2} h^2}}{q_2 h + \frac{1}{3} q_1 h^3 + h^2 \sqrt{q_1 q_2}\sqrt{1 + \frac{q_1}{12 q_2} h^2}}$$

From an alternative method (5.3) we have

$$q_{11} h + q_{12} h\, [\, z(1-z^{-1}) + z^{-1}(1-z)\,] + q_{22}(1-z)(1-z^{-1})$$

$$= 2q_2 h + \frac{2}{3} q_1 h + (\frac{1}{6} q_1 h - q_2 h)(z + z^{-1})$$

$$= p_0^2 + p_1^2 + p_0 p_1 (z + z^{-1})$$

The solution is given by

$$p_0^2 = q_2 h + \frac{1}{3} q_1 h^3 + h^2 \sqrt{q_1 q_2}\sqrt{1 + \frac{q_1}{12 q_2} h^2}$$

$$p_1 = (\frac{1}{6} q_1 h^2 - q_2 h\,) / p_0$$

The basic synthesis equation is of the form

$$(1-z^{-1}) r_0 + h z^{-1} s_0 = p_0 + p_1 z^{-1}$$

with

$$r_0 = p_0$$

$$s_0 = (\, p_1 + r_0\,) / h$$

The linear regulator can be written as

$$u(k) = -\, s_0 / r_0\, x(k)$$

$$= -\, (\, p_0 p_1 + p_0^2\,) / (\, h p_0^2\,)\; x(k)$$

Substituting p_0 and p_1 into the above equality, we get the same result as previous state space approach.

□

Example 8.

Given a continuous system

$$\dot{X}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} X(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

with the problem of minimizing

$$J = \int_0^T [\, q_1 x_1^2(t) + q_2 x_2^2(t) + r\, u^2(t) \,]\, dt$$

The discrete time model is given by

$$X(k+1) = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} X(k) + \begin{bmatrix} 0.5h^2 \\ h \end{bmatrix} u(k)$$

and

$$J = \sum_0^T [\, q_{11} x_1^2 + q_{22} x_2^2 + q_{33} u^2 + 2q_{12} x_1 x_2 + 2q_{13} x_1 u + 2q_{23} x_2 u \,]$$

where

$$q_{11} = q_1 h$$

$$q_{22} = q_2 h + \frac{1}{3} q_1 h^3$$

$$q_{33} = \frac{1}{20} q_1 h^5 + \frac{1}{3} q_2 h^3 + rh$$

$$q_{12} = 0.5\, q_1 h^2$$

$$q_{13} = \frac{1}{6} q_1 h^2$$

$$q_{23} = \frac{1}{8} q_1 h^3 + \frac{1}{2} q_2 h$$

The block diagram of the system is shown in Fig. 3.
The matrix L(k) is given by

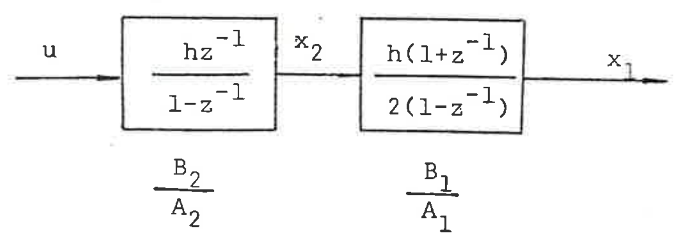


Fig. 3. A process in example 8.

$$L = [\,\Gamma S \Gamma + q_{33}\,]^{-1}[\,\Gamma^T S\Phi + Q_{12}\,]$$

$$= \frac{[\,0.5h^2 s_{11} + hs_{12} + q_{13} \quad 0.5h^3 s_{11} + 1.5h^2 s_{12} + hs_{22} + q_{23}\,]}{0.25h^4 s_{11} + h^3 s_{12} + h^2 s_{22} + q_{33}}$$

The steady solution of S(k) is obtained from

$$S(k) = \Phi^T S(k+1)\Phi + Q_{11} - L^T[\,\Gamma^T S(k+1)\Gamma + q_{33}\,]\,L$$

If h=1, the numerical values can be found as

$$S(\infty) = \begin{bmatrix} 1.8275 & 1.1282 \\ 1.1282 & 1.9094 \end{bmatrix}$$

$$Q = \begin{bmatrix} 1 & 0.5 & 0.1667 \\ 0.5 & 1.3333 & 0.6250 \\ 0.1667 & 0.6250 & 1.3833 \end{bmatrix}$$

and

$$L = [\,0.4528 \quad 1.0538\,]$$

Rewrite the control law of state feedback as the form of output feedback.

$$u(k) = -\ell_1 x_1(k) - \ell_2 x_2(k)$$

$$= -\ell_1 x_1(k) - \ell_2 \frac{2(1-z^{-1})}{h(1+z^{-1})} x_1(k)$$

$$= -\frac{(2\ell_2 + \ell_1 h) - (2\ell_2 - \ell_1 h)z^{-1}}{h(1+z^{-1})} x_1(k) \qquad (\,*\,)$$

$$= -\frac{2.5604 - 1.6548z^{-1}}{1+z^{-1}} x_1(k)$$

or

$$u(k) = -\ell_1 x_1(k) - \ell_2 \frac{hz^{-1}}{1-z^{-1}} u(k)$$

i.e.

$$u(k) = -\frac{\ell_1 (1-z^{-1})}{1+(\ell_2 h-1)z^{-1}} x_1(k) \qquad (*\,*)$$

$$= -\frac{0.4528(1-z^{-1})}{1+0.0538z^{-1}}\, x_1(k)$$

The basic synthesis equation for the algebraic design is

$$AR + BS = PT$$

with

$$P(z^{-1}) = 2\,(\,1 - 0.7198z^{-1} + 0.1726z^{-2})$$

$$T(z^{-1}) = 1 + z^{-1} \qquad \text{for } (\,*\,)$$

and

$$P(z^{-1}) = 2\,(\,1 - 0.7198z^{-1} + 0.1726z^{-2})$$

$$T(z^{-1}) = 1 - z^{-1} \qquad \text{for } (*\,*)$$

If the algebraic method is used, equation (5.9) gives

$$\begin{aligned} PP^* &= q_{11} B_1 B_2 B_2^* B_1^* + q_{22} A_1 B_2 B_2^* A_1^* + q_{33} A_1 A_2 A_2^* A_1^* + \\ &\quad + q_{12} [B_1 B_2 B_2^* A_1^* + A_1 B_2 B_2^* B_1^*] + q_{13} [B_1 B_2 A_2^* A_1^* + A_1 A_2 B_2^* B_1^*] \\ &\quad + q_{23} [A_1 B_2 A_2^* A_1^* + A_1 A_2 B_2^* A_1^*] \\ &= 30.1988 - 16.4660(z + z^{-1}) + 3.3666(z^2 + z^{-2}) \end{aligned}$$

$$P(z^{-1}) = 4.4171\,(\,1 - 0.7198z^{-1} + 0.1726z^{-2})$$

The two methods are equivalent in the sense that they have same steady state solution of optimal control.

□

6. ACKNOWLEDGEMENT

We are most thankful to Dr Kucera who provided us with algorithm for solving the diophantine equation.

7. REFERENCES

1. Astrom K.J. (1979a): Algebraic System Theory as a Tool for Regulator Design. Dept. of Automatic Control. Lund Institute of Technology. CODEN: LUTFD2 / (TFRT-7164) / 1-023 / (1979)

2. Astrom K.J. (1979b): Simple Self-tuners (1). CODEN: LUTFD2 / (TFRT-7184) / 1-063 / (1979)

3. Astrom K.J. and Zhou Z.Y. (1981): Self-tuner with Automatic Adjustment of the Sampling Period for Processes with Time Delays. To appear.

4. Astrom K.J. (1970): Introduction to Stochastic Control Theory. Academic Press. New York.

5. Astrom K.J. (1963): On the Choice of Sampling Rates in Optimal Linear Systems. Reseach report RJ-243. April 25, 1963.

6. Zhou Z.Y. and Astrom K.J. (1981): Polynomial Factorization Using Recursive Formulas For Complex Integral.

7. Kwakernaak H. and Sivan R. (1972): Linear Optimal Control Systems. Wiley-interscience. New York.

8. Blankinship W.A.(1963): A New Version of the Euclidean Algorithm. American Mathematics. Monthly 70 (1963) pp.742-745.

9. Kucera V.(1979): Discrete Linear Control. Academia Prague.

APPENDIX A

Sampling a Continous System with a Time Delay.

Consider a linear time invariant SISO system with time delay described by differential equations.

$$\dot{X}(t) = A\,X(t) + B\,u(t-\tau)$$

$$y(t) = C\,X(t)$$

where τ is the time delay of the system

$$\tau = k\,h + \tau_1$$

h is the sampling interval.

k is an integer.

Then the resulting discrete time system is given by

$$X(t+h) = F\,X(t) + G_1 u(t-kh) + G_2 u(t-kh-h)$$

$$y(t) = C\,X(t)$$

where

$$F = e^{Ah}$$

$$G_1 = \left(\int_0^{h-\tau_1} e^{As}\,ds \right) B$$

$$G_2 = \left(\int_{h-\tau_1}^{h} e^{As}\,ds \right) B$$

The pulse transfer functions of the system is given by

$$H(z^{-1}) = z^{-k} C\,(zI - F)^{-1} G_1 + z^{-k-1} C\,(zI - F)^{-1} G_2$$

when $X(0) = 0$.

EXAMPLE 1.

A first order system

$$\frac{B(s)}{A(s)} = \frac{K\,e^{-s\tau}}{Ts + 1}$$

The sampling version is described straightforword by

$$y(t+h) + a\,y(t) = b_1 u(t-kh) + b_2 u(t-kh-h)$$

where

$$a = e^{-h/t}$$

$$b_1 = K \, [\, 1 - e^{-(h - \tau \bmod h)/T} \,]$$

$$b_2 = K \, [\, e^{-(h - \tau \bmod h)/T} - e^{-h/T} \,]$$

$$k = \tau \operatorname{div} h$$

□

EXAMPLE 2.

A second order system containing two lags and a time delay is described by

$$\frac{B(s)}{A(s)} = \frac{K \, e^{-s\tau}}{(s+a)(s+b)}$$

Its state differential equation has the form

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

$$y(t) = \frac{K}{b-a} \, [\, 1 \;\; -1 \,] \, X(t)$$

Using the previous formulae we have

$$y(t+h) + a_1 y(t) + a_2 y(t-h) = b_1 u(t-kh) + b_2 u(t-kh-h) + \\ + b_3 u(t-kh-2h)$$

where

$$a_1 = -\,(\alpha + \beta)$$

$$a_2 = \alpha \, \beta$$

$$b_1 = \frac{K}{ab(b-a)} \, [\, b\,(1-\gamma) - a\,(1-\zeta) \,]$$

$$b_2 = \frac{K}{ab(b-a)} \, [\, (a-b)(\alpha+\beta) + b\gamma\,(1+\beta) - a\zeta\,(1+\alpha) \,]$$

$$b = \frac{K}{ab(b-a)} \, [\, (b-a)\,\alpha\,\beta + a\,\alpha\,\zeta - b\,\beta\,\gamma \,]$$

$$\alpha = \exp(-ah)$$

$$\beta = \exp(-bh)$$

$$\gamma = \exp[-a(h-\tau \bmod h)]$$

$$\zeta = \exp[-b(h-\tau \bmod h)]$$

$$k = \tau \operatorname{div} h$$

□

APPENDIX_B

Some Theorems of Optimal Control. Astrom (1963).

Theorem_B1

Consider a linear system described by a continuous time state space model

$$\dot{X} = A\,X + B\,U \tag{b.1}$$

The expected loss function is given by

$$J = X^T(T)Q_0 X(T) + \int_0^T \begin{bmatrix} X(t) \\ U(t) \end{bmatrix}^T Q(t) \begin{bmatrix} X(t) \\ U(t) \end{bmatrix} dt \tag{b.2}$$

where $Q(t)$ is a positive semi-definite symmetrical matrix with bounded elements

$$Q(t) = \begin{bmatrix} Q_{11}(t) & Q_{12}(t) \\ Q_{21}(t) & Q_{22}(t) \end{bmatrix}$$

$$Q_{12}(t) = Q_{21}(t)^T$$

Then the corresponding discrete version is given by

$$X(k+1) = \Phi\,X(k) + \Gamma\,U(k) \tag{b.3}$$

where

$$\Phi = e^{Ah}$$

$$\Gamma = \int_0^h e^{As} B\,ds$$

The expected loss function becomes

$$J = X^T(T)Q_0 X(T) + \sum_0^{T-1} \begin{bmatrix} X(k) \\ U(k) \end{bmatrix}^T \tilde{Q}(k) \begin{bmatrix} X(k) \\ U(k) \end{bmatrix} \tag{b.4}$$

where

$$\tilde{Q}(k) = \begin{bmatrix} \tilde{Q}_{11}(k) & \tilde{Q}_{12}(k) \\ \tilde{Q}_{21}(k) & \tilde{Q}_{22}(k) \end{bmatrix}$$

$$\tilde{Q}_{11}(k) = \int_0^h \Phi^T(t)\, Q_{11}\Phi(t)\, dt$$

$$\tilde{Q}_{12}(k) = \int_0^h [\, \Phi^T(t)\, Q_{11}\Gamma(t) + \Phi^T(t)\, Q_{12}\,]\, dt$$

$$\tilde{Q}_{21}(k) = \tilde{Q}^T_{12}(k)$$

$$\tilde{Q}_{22}(k) = \int_0^h [\, \Gamma^T(t)\, Q_{11}\Gamma(t) + \Gamma^T(t) Q_{12}(t) + Q_{21}(t)\Gamma(t) + \\ + Q_{22}\,]\, dt$$

Theorem_B2

The minimal value of loss function (b.4) of the discrete system is

$$J_{min} = X_0^T\, S(t_0,T)\, X_0 \tag{b.5}$$

where the symmetric matrix $S(t_0,T)$ is given by the recursive equation

$$S_k = \Phi^T S_{k+1}\Phi - L^T [\, \Gamma^T S_{k+1}\Gamma + \tilde{Q}_{22}\,]\, L + \tilde{Q}_{11}$$

and

$$L_k = [\, \Gamma^T S_{k+1}\Gamma + \tilde{Q}_{22}\,]^{-1} [\, \Gamma^T S_{k+1}\Phi + \tilde{Q}_{21}\,] \tag{b.6}$$

Then the sequence of control for which the minimal value (b.5) is attained is given by

$$U(k) = - L\, X(k)$$

The canonical equations, or Euler's equations, whose solutions are the characteristics of the Hamilton-Jacobi equation, are

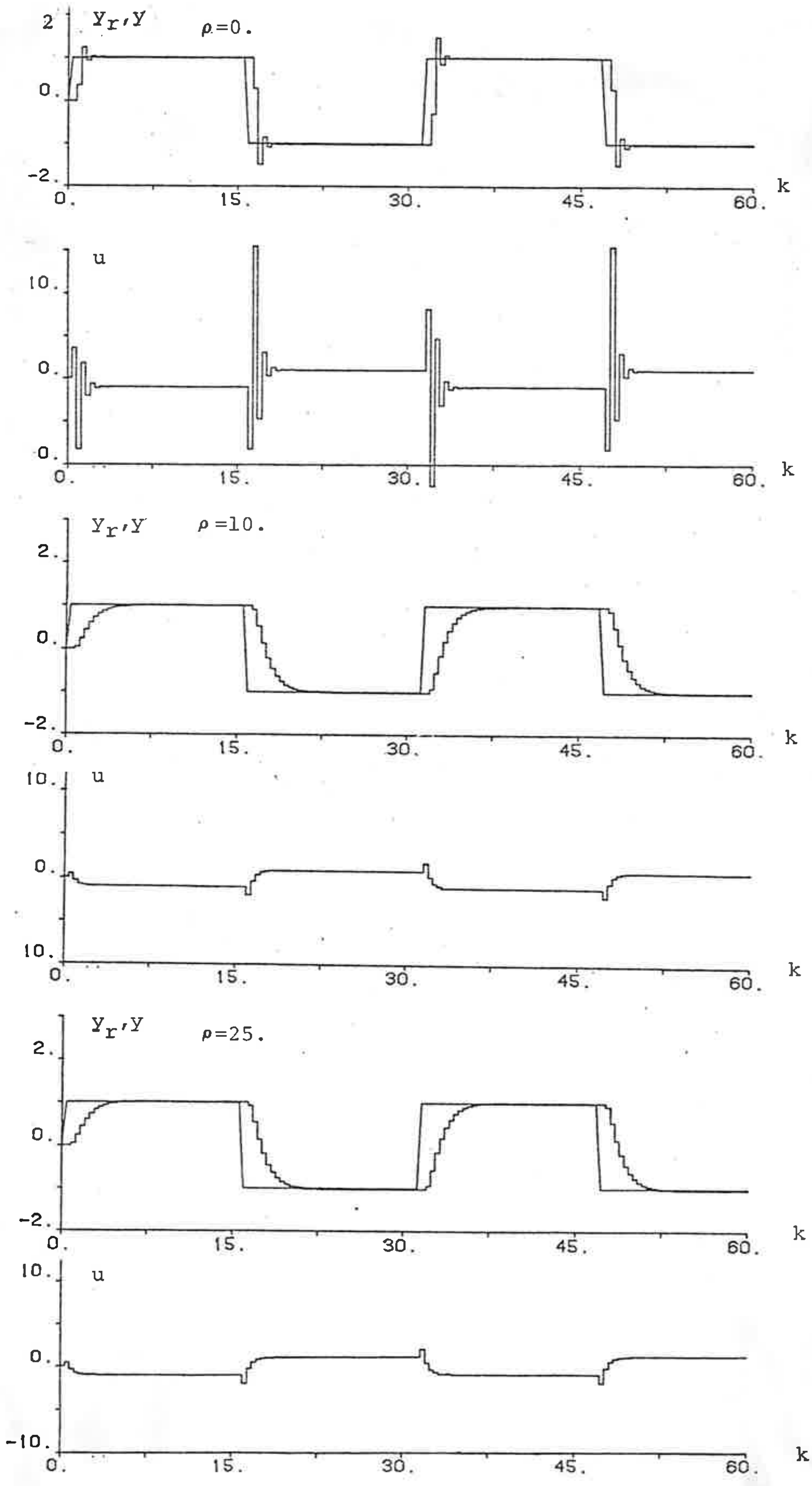
$$\begin{aligned} X(k+1) &= [\Phi-\Gamma\tilde{Q}_{22}^{-1}\tilde{Q}_{21}]X(k) - \Gamma\tilde{Q}_{22}^{-1}\Gamma^T p(k) \\ p(k-1) &= [\, \tilde{Q}_{11} - \tilde{Q}_{12}\tilde{Q}_{22}^{-1}\tilde{Q}_{21}]X(k) + [\Phi-\Gamma\tilde{Q}_{22}^{-1}\tilde{Q}_{21}]^T p(k) \end{aligned} \tag{b.7}$$

with the boundary conditions

$$X(0) = X(t_0)$$

$$p(T-1) = Q_0 X(T)$$

APPENDIX C: Simulation Results and Program of Example 4



```
Discrete System Proc4

Input   U
Output  Y
Time    T
Tsamp   ts
State   X1   X2   X3
New     N1   N2   N3

y=x1
n1=x2+h1*u
n2=x3+h2*u
n3=-a3*x1-a2*x2-a1*x3+h3*u

h1=b1
h2=b2-a1*h1
h3=b3-a1*h2-a2*h1

ts=t+dt

dt:0.4
a1:-2.896
a2:1.492
a3:0
b1:0.101
b2:0.288
b3:0.014

end



Discrete System Reg4

Input   Y   Yr
output  U
State   U1   U2   Y1
New     Nu1  Nu2  Ny1
Time    T
Tsamp   Ts

U=if con<1 then yr else t0*yr-(r1*u1+r2*u2+s0*y+s1*y1)/r0
nu1=u
nu2=u1
ny1=y

ts=t+dt

r0:0.2830
r1:0.4958
r2:0.0238
s0:4.3445
s1:-2.5399
t0:1

dt:0.4

con:5
end
```

```
connecting system conr4

Time T

yref=yr0*sign(sin(a0*t))
yr[reg5]=yref
y[reg5]=y[proc5]
u[proc5]=u[reg5]

a0:0.1
yr0:1

end
```

APPENDIX D: PROGRAM LISTING.

```
Program Regudesign;

{ Program called RDP.
  Author Zhou Z.Y.
  Date   Apr.1981.
  Reference
    Kucera V.(1979): Discrete Linear Control. Academia Prague.

  The program solves the diophantine equation
      AR + BS = C
The general solution is given by
      R = PE + GV
      S = PF + HV
where V is an arbitrary polynomial.

  If    deg p <= deg A + deg B -1
then R and S are found uniquely:
      V = - [ PE mod G ] = - [ PF mod H ]
      R = PE div G
      S = PF div H
else there are one minimum degree solution for R and
one for S.                                             }


const n=10;
      eps1=0.0001;
type  polytype=record
        z:array[0..n] of real;
        d:integer;
        end;
var   a,b,p,c,e,f,g,h,a0,b0,a1,b1:polytype;
      r1,r2,s1,s2,v1,v2,p1,p2,ep,fp:polytype;
      i,j,k,nm,rd:integer;
      eps:real;
      filename:array[1..14] of char;

{------------------------------------------------------------Input}
Procedure Input;
{ Input Na,Nb,A[i],B[i] and check if A or B are zero polynomials. }
begin
  writeln('Input Na, Nb and Np(the degree of A ,B and P.)');
  read(a.d,b.d,p.d);
  writeln('Please input A[i]');
  for i:=0 to a.d do read(a.z[i]);
  writeln('Please input B[i]');
  for i:=0 to b.d do read(b.z[i]);
  writeln('Please input P[i]');
  for i:=0 to p.d do read(p.z[i]);
  writeln('Please input eps');
  read(eps);
  writeln('Please input filename for stroe results');
  read(filename);
end;

{-------------------------------------------------------Initialize}
Procedure Initialize;
{ Set the initial values for polynomials. }
begin
```

```
  a0.d:=a.d;
  b0.d:=b.d;
  a1.d:=a.d;
  b1.d:=b.d;
  for i:=0 to a.d do begin
      a0.z[i]:=a.z[i];
      a1.z[i]:=a.z[i];
  end;
  for i:=0 to b.d do begin
      b0.z[i]:=b.z[i];
      b1.z[i]:=b.z[i];
  end;
  if a.d>b.d then nm:=a.d else nm:=b.d;
  c.d:=0;
  e.d:=b.d;
  g.d:=b.d;
  f.d:=a.d;
  h.d:=a.d;
  for i:=0 to n do begin
      e.z[i]:=0;
      f.z[i]:=0;
      g.z[i]:=0;
      h.z[i]:=0;
  end;
  e.z[0]:=1;
  h.z[0]:=1;
  c.z[0]:=1;
end;

{-----------------------------------------------------------Exchange}
Procedure Exchange(hi,hj:polytype; var hk,hl:polytype);
{ Exchange polynomials Hi and Hj. }
begin
  hl.d:=hi.d;
  hk.d:=hj.d;
  for i:=0 to hi.d do hl.z[i]:=hi.z[i];
  for i:=0 to hj.d do hk.z[i]:=hj.z[i];
end;

{---------------------------------------------------------------Norm}
Procedure Norm(hi:polytype);
{ Calculate the Euclidean norm of a polynomial Hi. }
var d1:real;
begin
  if hi.d>=0 then begin
     d1:=0.0;
     for i:=0 to hi.d do d1:=d1+hi.z[i]*hi.z[i];
     d1:=eps*sqrt(d1);
     rd:=round(d1);
  end;
end;

{-------------------------------------------------------------Reduce}
Procedure Reduce(var hi:polytype);
{ Reduce Ni of Hi(z) if its leading coefficient is smaller
  in modulus than an external variable. }
label 2,4;
begin
  if hi.d<0 then goto 4;
2:if abs(hi.z[hi.d])<=rd then begin
     hi.d:=hi.d-1;
```

```
      if hi.d<0 then goto 4;
      goto 2;
   end;
4:
end;

{------------------------------------------------------Transformation}
Procedure Transformation;
{ Performs the Euclidean transformation to find a general
  solution for the diophantine equation. }
label 10,20,30,40;
var   q:real;
begin
    if a.d<b.d then goto 20 else goto 30;
10: while a.d>=b.d do begin
      k:=a.d-b.d;
      q:=a.z[a.d]/b.z[b.d];
      a.d:=a.d-1;
      if k<=a.d then
         for i:=k to a.d do a.z[i]:=a.z[i]-q*b.z[i-k];
      for i:=k to nm do begin
          e.z[i]:=e.z[i]-q*g.z[i-k];
          f.z[i]:=f.z[i]-q*h.z[i-k];
      end;
      reduce(a);
      if a.d<b.d then goto 20;
    end;
20:
    exchange(a,b,a,b);
    exchange(e,g,e,g);
    exchange(f,h,f,h);
    if (b.d=0) and (abs(b.z[0])<eps1) then goto 40;
30: if b.d>=0 then begin
       norm(b);
       goto 10;
    end;
40: if abs(a.z[0])<eps1 then q:=1 else q:=a.z[0];
    for i:=0 to nm do begin
        e.z[i]:=e.z[i]/q;
        f.z[i]:=f.z[i]/q;
        a.z[i]:=a.z[i]/q;
    end;
    norm(e);
    reduce(e);
    norm(f);
    reduce(f);
    g.d:=g.d-a.d;
    h.d:=h.d-a.d;
end;

{------------------------------------------------------------Polymul}
Procedure Polymul(p,ef:polytype; var efp:polytype);
{ Polynomial multiplication efp=p*ef }
begin
  efp.d:=p.d+ef.d;
  for i:=0 to efp.d do efp.z[i]:=0;
  for i:=0 to p.d do
      for j:=0 to ef.d do
          efp.z[i+j]:=efp.z[i+j]+p.z[i]*ef.z[j];
end;
```

```
{------------------------------------------------------------Polydiv}
Procedure Polydiv(efp,gh:polytype; var rs,vv:polytype);
{ Polynomial division rs=efp mod gh.
                      vv=-(efp div gh). }
var efp1,gh1,rs1,vv1:array[0..n] of real;
begin
  vv.d:=efp.d-gh.d;
  rs.d:=gh.d-1;
  for i:=0 to efp.d do efp1[i]:=efp.z[efp.d-i];
  for i:=0 to gh.d do gh1[i]:=gh.z[gh.d-i];
  for i:=gh.d+1 to efp.d do gh1[i]:=0;
  vv1[0]:=efp1[0]/gh1[0];
  for i:=1 to efp.d do
      rs1[i]:=efp1[i]-vv1[0]*gh1[i];
  for i:=1 to vv.d do begin
      vv1[i]:=rs1[1]/gh1[0];
      rs1[efp.d+1]:=0;
      for j:=1 to efp.d do
          rs1[j]:=rs1[j+1]-vv1[i]*gh1[j];
  end;
  for i:=1 to rs.d+1 do rs.z[i-1]:=rs1[rs.d+2-i];
  for i:=0 to vv.d do vv.z[i]:=-vv1[vv.d-i];
end;

{------------------------------------------------------------Polyadd}
Procedure Polyadd(efp,ghv:polytype; var rs:polytype);
{ Polynomial addition rs=efp+ghv. }
var diff:integer;
begin
  diff:=efp.d-ghv.d;
  if diff>0 then begin
     rs.d:=efp.d;
     for i:=0 to efp.d do rs.z[i]:=efp.z[i];
     for i:=0 to ghv.d do
         rs.z[i]:=rs.z[i]+ghv.z[i];
     end
     else begin
       rs.d:=ghv.d;
       for i:=0 to ghv.d do rs.z[i]:=ghv.z[i];
       for i:=0 to efp.d do
           rs.z[i]:=rs.z[i]+efp.z[i];
     end;
end;

{----------------------------------------------------------Regulator}
Procedure Regulator;
{ computes coefficients of V1,V2,R1,R2,S1 and R2.}
var rs1:polytype;

begin
  polymul(p,e,ep);
  polymul(p,f,fp);
  polydiv(ep,g,r1,v1);
  polymul(v1,h,rs1);
  polyadd(fp,rs1,s1);
  norm(s1);
  reduce(s1);
  polydiv(fp,h,s2,v2);
  polymul(v2,g,rs1);
  polyadd(ep,rs1,r2);
  norm(r2);
```

```
  reduce(r2);
end;

{--------------------------------------------------------------Division}
Procedure Division;
{ Cancell the commom factor from A(z) and B(z). }
begin
  a.d:=a0.d-c.d;
  b.d:=b0.d-c.d;
  for i:=0 to a0.d do a.z[i]:=a0.z[i];
  for i:=0 to b0.d do b.z[i]:=b0.z[i];
  for j:=0 to a.d do begin
      a.z[j]:=a.z[j]/c.z[0];
      for i:=j+1 to c.d+j do
          a.z[i]:=a.z[i]-c.z[i-j]*a.z[j];
  end;
  for j:=0 to b.d do begin
      b.z[j]:=b.z[j]/c.z[0];
      for i:=j+1 to c.d+j do
          b.z[i]:=b.z[i]-c.z[i-j]*b.z[j];
  end;
  a1.d:=a.d;
  b1.d:=b.d;
  for i:=0 to a.d do a1.z[i]:=a.z[i];
  for i:=0 to b.d do b1.z[i]:=b.z[i];
end;

{------------------------------------------------------------------chek}
Procedure Check;
{ Chek result of AX+BY=C1. }
var ar1,bs1,ar2,bs2:polytype;
begin
  polymul(a1,r1,ar1);
  polymul(a1,r2,ar2);
  polymul(b1,s1,bs1);
  polymul(b1,s2,bs2);
  if ar1.d>bs1.d then begin
     p1.d:=ar1.d;
     for i:=bs1.d+1 to ar1.d do
         bs1.z[i]:=0.0;
     end
     else begin
            p1.d:=bs1.d;
            for i:=ar1.d+1 to bs1.d do
                ar1.z[i]:=0.0;
          end;
  if ar2.d>bs2.d then begin
     p2.d:=ar2.d;
     for i:=bs2.d+1 to ar2.d do
         bs2.z[i]:=0.0;
     end
     else begin
            p2.d:=bs2.d;
            for i:=ar2.d+1 to bs2.d do
                ar2.z[i]:=0.0;
          end;
  for i:=0 to p1.d do p1.z[i]:=ar1.z[i]+bs1.z[i];
  for i:=0 to p2.d do p2.z[i]:=ar2.z[i]+bs2.z[i];
end;

{-----------------------------------------------------------------Store}
```

```
Procedure Store;
var len:integer;
    outfile:file of char;
begin
  len:=-1;
  rewrite(outfile,filename,'DAT',len);
  if len=-1 then writeln('illfile');
  writeln(outfile,'The solution of AR+BS=P');
  write(outfile,'    D  ');
  for i:=0 to nm do write(outfile,'   P[',i:2,'] ');
  writeln(outfile);
  write(outfile,'Na =',a0.d:2);
    for i:=0 to a0.d do write(outfile,a0.z[i]:8:4);
    writeln(outfile);
  write(outfile,'Nb =',b0.d:2);
    for i:=0 to b0.d do write(outfile,b0.z[i]:8:4);
    writeln(outfile);
  write(outfile,'Np =',p.d:2);
    for i:=0 to p.d do write(outfile,p.z[i]:8:4);
    writeln(outfile);
  write(outfile,'Nc =',c.d:2);
    for i:=0 to c.d do write(outfile,c.z[i]:8:4);
    writeln(outfile);
  write(outfile,'Na1=',a1.d:2);
    for i:=0 to a1.d do write(outfile,a1.z[i]:8:4);
    writeln(outfile);
  write(outfile,'Nb1=',b1.d:2);
    for i:=0 to b1.d do write(outfile,b1.z[i]:8:4);
    writeln(outfile);
  write(outfile,'Ne =',e.d:2);
    for i:=0 to e.d do write(outfile,e.z[i]:8:4);
    writeln(outfile);
  write(outfile,'Nf =',f.d:2);
    for i:=0 to f.d do write(outfile,f.z[i]:8:4);
    writeln(outfile);
  write(outfile,'Ng =',g.d:2);
    for i:=0 to g.d do write(outfile,g.z[i]:8:4);
    writeln(outfile);
  write(outfile,'Nh =',h.d:2);
    for i:=0 to h.d do write(outfile,h.z[i]:8:4);
    writeln(outfile);
  write(outfile,'Nv1=',v1.d:2);
    for i:=0 to v1.d do write(outfile,v1.z[i]:8:4);
    writeln(outfile);
  write(outfile,'Nv2=',v2.d:2);
    for i:=0 to v2.d do write(outfile,v2.z[i]:8:4);
    writeln(outfile);
  write(outfile,'Nr1=',r1.d:2);
    for i:=0 to r1.d do write(outfile,r1.z[i]:8:4);
    writeln(outfile);
  write(outfile,'Ns1=',s1.d:2);
    for i:=0 to s1.d do write(outfile,s1.z[i]:8:4);
    writeln(outfile);
  write(outfile,'Nr2=',r2.d:2);
    for i:=0 to r2.d do write(outfile,r2.z[i]:8:4);
    writeln(outfile);
  write(outfile,'Ns2=',s2.d:2);
    for i:=0 to s2.d do write(outfile,s2.z[i]:8:4);
    writeln(outfile);
  writeln(outfile,' Chek AX+BY=C ');
  write(outfile,'Np1=',p1.d:2);
```

```
    for i:=0 to p1.d do write(outfile,p1.z[i]:8:4);
    writeln(outfile);
  write(outfile,'Np2=',p2.d:2);
    for i:=0 to p2.d do write(outfile,p2.z[i]:8:4);
    writeln(outfile);
  close(outfile);
end;

{===============================================Code of main program}
begin
    input;
    initialize;
    transformation;
    if a.d>0 then begin
       c.d:=a.d;
       for i:=0 to a.d do c.z[i]:=a.z[i];
       division;
    end;
    regulator;
    check;
    store;
end.
```