Numerical Algorithms for Polynomial Matrix Systems

Pernebo, Lars

1980

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

Link to publication

Citation for published version (APA):
Pernebo, L. (1980). *Numerical Algorithms for Polynomial Matrix Systems.* (Technical Reports TFRT-7205). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:
1

CODEN: LUTFD2/(TFRT-7205)/1-048/(1980)

NUMERICAL ALGORITHMS FOR POLYNOMIAL MATRIX SYSTEMS

LARS PERNEBO

DEPARTMENT OF AUTOMATIC CONTROL
LUND INSTITUTE OF TECHNOLOGY
OCTOBER 1980

CONTENTS

NUMERICAL ALGORITHMS FOR POLYNOMIAL MATRIX SYSTEMS

Lars Pernebo

Organization **LUND INSTITUTE OF TECHNOLOGY** Department of Automatic Control Box 725 S-220 07 LUND 7 Sweden	Document name Internal report Date of issue October 1980 CODEN: LUTFD2/(TFRT-7205)/1-048/(1980)
Author(s) Lars Pernebo	Sponsoring organization Swedish Board for Technical Development (STU), contract no 77-3548

Title and subtitle

Numerical Algorithms for Polynomial Matrix Systems

A4 A5

Abstract

This report contains a number of algorithms for polynomial matrix manipulations. First some mathematical problems are considered. Algorithms are given for e.g. finding greatest common divisiors, solving polynomial matrix equations and factorizing polynomial matrices. Then linear dynamical systems, described by polynomial matrices, are considered. Algorithms are given for e.g. transformation to state space form, computation of poles, zeros and decoupling zeros, and computation of inverses.

Key words A4 A5

Classification system and/or index terms (if any)

Supplementary bibliographical information	Language English
ISSN and key title	ISBN

Recipient's notes	Number of pages 48	Price
	Security classification	

Distribution by (name and address)

DOKUMENTDATABLAD enl SIS 61 41 21

1. Introduction

Polynomial matrices were introduced into the theory for linear multivariable systems by Rosenbrock (1970). A system, described by linear differential or difference equations as well as static relations, may be written

$$T(\mu)\xi = U(\mu)u \tag{1.1 a}$$
$$y = V(\mu)\xi + W(\mu)u. \tag{1.1 b}$$

Here $T(\mu)$, $U(\mu)$, $V(\mu)$, and $W(\mu)$ are polynomial matrices in the operator μ, which is the differential operator for continuous time systems and the shift operator for discrete time systems. In Rosenbrock (1970) systems of the type (1.1) are analysed in detail.

During the last few years it has been shown that many design problems can be solved with a polynomial matrix approach. In Wolovich (1974) state estimation and state feedback have been expressed in terms of polynomial matrices. The servo problem is treated in e.g. Wang, Davison (1973), Anderson, Scott (1977), and Pernebo (1978) and the output regulation problem is treated in e.g. Bengtsson (1977), Cheng, Pearson (1978), and Pernebo (1978). Polynomial matrices have also been used to solve the feedback realization problem, Pernebo (1978) Practical use of the design theory requires computer programs which allow the user to utilize the results without knowing the whole theory. Such programs must be based on good numerical algorithms. The purpose of this report is to show how algorithms for polynomial matrix design methods can be constructed. In Pernebo (1980) it is shown how these algorithms may be incorporated in an interactive computer program for analysis and design of linear systems.

In chapter 2 of this report some basic algorithms are given. There are some mathematical problems that occur in most design procedures, e.g. the problem of finding the greatest common divisor of two polynomial matrices or of solving certain polynomial matrix equations. Algorithms for solving this kind of problems are given in chapter 3. In chapter 4, finally, algorithms for analysis and transformation of dynamical systems are given. It is also shown how some design problems may be solved.

Some Notational Remarks

The algorithms in this report will be written

$$[A_1, A_2, \ldots, A_k] = \text{ALG}[B_1, B_2, \ldots, B_h],$$

where ALG is the name of the algorithm. Furthermore, $B_1,\ldots,B_h$ are the input data, e.g. polynomial matrices or real numbers, and $A_1,\ldots,A_k$ are the output data. If the algorithm is used by another algorithm and not all the output data are needed then the A_i, that are not needed, are replaced by dots. Analogously, the B_i, that are not needed to be specified in a particular case, are replaced by dots.

2. Basic Algorithms

In this chapter two different algorithms will be presented. They both solve essentially the same problem, but give the result in different forms. In many applications it does not matter which of the algorithms is used.

2.1 Background

Let $A(s)$ be an $n \times m$ polynomial matrix in the indeterminate s. If r is the rank of $A(s)$ then there is an $n \times r$ polynomial matrix $L(s)$ and an $m \times m$ unimodular polynomial matrix $N(s)$, such that

$$A(s)N(s) = [L(s) \quad 0] \tag{2.1}$$

This is shown in e.g. Pernebo (1978). It will be shown that many problems, involving polynomial matrices, can be solved with an algorithm which computes $L(s)$ and $N(s)$ such that (2.1) holds.

The matrix $[L(s) \quad 0]$ can be obtained by performing elementary column operations on $A(s)$, since $N(s)$ is unimodular. The following three types of elementary column operations on $A(s)$ are needed.

- Interchange columns i and j. (2.2)
- Multiply column i by a nonzero real number c. (2.3)
- Add column j, multiplied by a polynomial $a(s)$, to column i. (2.4)

Now, suppose that there exists a scheme of column operations which brings $A(s)$ to $[L(s) \quad 0]$ in (2.1). The transformation matrix $N(s)$ and its inverse $N^{-1}(s)$ can then be obtained in the following way. Let $D(s)$ be an arbitrary polynomial matrix with m columns. For every column operation, applied to $A(s)$, apply the same column operation to $D(s)$. The result is a polynomial matrix $E(s)$, given by

$$E(s) = D(s)N(s). \tag{2.5}$$

The matrix $N(s)$ is obtained with the choice $D(s) = I$. In many cases, however, the product $D(s)N(s)$ is of interest rather than $N(s)$ itself.

Let $F(s)$ be an arbitrary polynomial matrix with m rows. To each of the column operations (2.2) - (2.4) define a corresponding row operation in the following way.

- Interchange rows i and j. (Corresponds to (2.2)). (2.6)
- Divide row i by c in (2.3). (Corresponds to (2.3)). (2.7)
- Subtract row i, multiplied by $a(s)$ in (2.4), from row j. (Corresponds to (2.4)). (2.8)

For every column operation, applied to $A(s)$, apply the corresponding row operation to $F(s)$. The result is a polynomial matrix $H(s)$, given by

$$H(s) = N^{-1}(s)F(s). \tag{2.9}$$

The polynomial matrix $N^{-1}(s)$ is obtained with the choice $F(s) = I$. In many cases, however, the product $N^{-1}(s)F(s)$ is of interest rather than $N^{-1}(s)$ itself.

Two different schemes of column operations which bring $A(s)$ to $[L(s) \quad 0]$ will be given in the following two sections. They will thus give rise to two different algorithms for computation of $L(s)$, $E(s)$, and $H(s)$ from $A(s)$, $D(s)$, and $F(s)$.

2.2 Column Proper Matrices

The concept of column proper matrices was defined in Wolovich (1974). A slightly modified definition will be used here.

Definition 2.1 The column degree of column i in the polynomial matrix $A(s)$ is equal to the highest power of s occurring in column i.

Definition 2.2 Let $B(s)$ be the matrix formed by the nonzero columns of $A(s)$ and let d_i be the column degree of column i in $B(s)$. Then $A(s)$ is column proper if the matrix $\lim_{s\to+\infty} B(s)\mathrm{diag}(s^{-d_1}\ldots s^{-d_k})$ has linearly independent columns.

Any polynomial matrix $A(s)$ can be made column proper via a series of elementary column operations. Such a scheme of column operations is given in the proof of lemma 4.6.1 in Pernebo (1978). Consequently, there exists a unimodular matrix $N(s)$, such that

$$A(s)N(s) = [L(s) \quad 0], \tag{2.10}$$

where $[L(s) \quad 0]$ is column proper. The nonzero columns are here collected in the matrix $L(s)$, which by definition 2.2 has linearly independent columns. It follows that $L(s)$ has $r = \operatorname{rank} A(s)$ columns and (2.10) is a relation of the type (2.1).

An algorithm, which makes a given matrix column proper, is given below. The following two algorithms are assumed to be available.

(i) A matrix $W = (V \quad U)$ is given. The matrix V is either empty (has no columns) or has linearly independent columns. A basis for the column space of W is selected so that it contains all the columns of V and some of the columns of U. Let η be a vector with the same number of components as U has columns and such that

$\eta_i = 0$ if column i of U belongs to the basis.
$\eta_i = 1$ otherwise.

The algorithm will be written

$\eta = LD[V,U]$.

(ii) The matrices P and Q are given. The range space of Q is included in the range space of P and P has linearly independent columns. The unique solution to the equation $PX = Q$ should be found. The algorithm will be written

$X = SOLV[P,Q]$.

Let the matrix $A(s)$ with column degrees $\{d_i\}$ be given. If a column is zero then define its degree as $d_j = -1$. Furthermore, let a_j be the vector of coefficients of s^{d_j} in column j of $A(s)$. The following algorithm uses the method of the proof of lemma 4.6.1 in Pernebo (1978) to make a polynomial matrix $A(s)$ column proper. The idea behind the method is the following. If $A(s)$ is not column proper, then there is a column, whose column degree can be reduced by addition of a linear combination of columns with lower degree. This procedure can be repeated until the matrix is column proper.

```
1.  Permute the columns of $A(s)$ so that the nonzero columns are to the left.
2.  Let $W_{-1}$ be an empty matrix and put $k=0$.
3.  Let $V_k$ be the matrix formed by the columns $\{a_i\}_{d_i=k}$
4.  Compute $\eta = LD[W_{k-1}, V_k]$.
5.  If $\eta \neq 0$ go to 9.
6.  If $k = \max_i d_i$ then stop.
7.  $W_k = (W_{k-1} \;\; V_k)$
8.  $k:=k+1$ and go to 3.
9.  Let the matrices $U_0$ and $U_1$ be formed from the columns of $V_k$ corresponding
    to $\eta=0$ and $\eta=1$ respectively.
10. Compute $Z = SOLV[(W_{k-1} \; U_0), U_1]$.
11. Let $m_i$ be the index $j$ of the $a_j$ that forms the i:th column of $(W_{k-1} \; U_0)$
    and let $n_i$ be the index $j$ of the $a_j$ that forms the i:th column of $U_1$.
12. $i = 1$.
13. To column $n_i$ in $A(s)$ is added
        $-\sum_j Z_{ji} A_{m_j}(s) s^{d_{n_i} - d_{m_j}}$
    where $A_{m_j}(s)$ is column $m_j$ of $A(s)$.
14. Update the value of $d_{n_i}$.
15. If $i$ is equal to the number of columns in $U_1$ go to 17.
16. $i:=i+1$ and go to 13.
17. $k = \min_j d_{n_j}$.
18. If $k=-1$ go to 1 else go to 3.
```

The algorithm will be written

$$[L(s), E(s), H(s)] = COLPRO[A(s), D(s), F(s)],$$

where E(s) and H(s) are computed as is shown in section 2.1. The relations between the matrices are given by (2.10), (2.5), and (2.9) for some unimodular matrix N(s).

2.3 Matrices in Echelon Form

One method for obtaining [L(s) 0] in (2.1) from an arbitrary polynomial matrix A(s) was given in the previous section. Another method is given below.

Find the entry of lowest degree in row 1 of $A(s)$ and suppose that it is in the i:th column. Add multiples of the i:th column to the other columns so that the other entries in row 1 get lower degree than the i:th has. Repeat the procedure until all entries but one of row 1 are zero. Change columns so that the nonzero entry comes to the left. Repeat the procedure for row 2, but exclude column 1. Continue like this for all the rows. The result is a matrix $[L(s)\ 0]$, where $L(s)$ is in Echelon form of the type shown in figure 2.1. The entries marked x are nonzero and the entries marked y are arbitrary.

$$L(s) = \begin{pmatrix} x & 0 & 0 & 0 & 0 \\ y & x & 0 & 0 & 0 \\ y & y & 0 & 0 & 0 \\ y & y & x & 0 & 0 \\ y & y & y & x & 0 \\ y & y & y & y & 0 \\ y & y & y & y & x \end{pmatrix}$$

Figure 2.1. The form of $L(s)$.

It follows that there exists a unimodular matrix $N(s)$, such that

$$A(s)N(s) = [L(s)\ 0]. \tag{2.11}$$

This is a relation of the type (2.1) since $L(s)$ has $r = \text{rank}\ A(s)$ linearly independent columns.

Let $a_{ij}(s) = a_{ij}^k s^k + \ldots + a_{ij}^o$ be the entry in position i,j of $A(s)$ and let $a_i(s)$ be column i of $A(s)$. The following algorithm brings the nxm polynomial matrix $A(s)$ to Echelon form.

```
1.  $r=0$, $k=1$.
2.  $r:=r+1$. If $r \geq n+1$ then stop
3.  If $a_{rj}(s) = 0$, $k \leq j \leq m$ then go to 2.
4.  $d = \min\{\deg a_{rj}(s) \mid k \leq j \leq m,\ a_{rj}(s) \neq 0\}$
5.  Determine $\ell \in S$, such that $|a_{r\ell}^d| \geq |a_{rj}^d|$
    $\forall\ j \in S \setminus \{\ell\}$. Here $S = \{i \mid k \leq i \leq m,\ \deg a_{ri}(s) = d\}$.
    If there are more than one candidate for $\ell$ choose the smallest one.
```

```
6.  $p=k$.
7.  If $p=\ell$ then $p:=p+1$.
8.  If $p \geq m+1$ then go to 13.
9.  If $a_{rp}(s) = 0$ then go to 12.
10. $a_p(s) := a_p(s) - \frac{a_{rp}^e}{a_{r\ell}^d} s^{e-d} a_\ell(s)$, where $e = \deg a_{rp}(s)$.
11. If $\deg a_{rp}(s) \geq d$ then go to 10.
12. $p:=p+1$ and go to 7.
13. If $a_{rj}(s) \neq 0$ for some $j \in \{j \mid k < j \leq m,\ j \neq \ell\}$ then go to 4.
14. Change columns $\ell$ and $k$.
15. $k:=k+1$. If $k \geq m$ then stop
16. Go to 2.
```

The algorithm will be written

$$[L(s),\ E(s),\ H(s)] = \text{ECHELON}[A(s),\ D(s),\ F(s)],$$

where E(s) and H(s) are computed as is shown in section 2.1. The relations between the matrices are given by (2.11), (2.5), and (2.9) for some unimodular matrix N(s).

2.4 The Basic Algorithm

In many applications it does not matter which of the algorithms COLPRO and ECHELON that is used to obtain (2.1). In such a case the algorithm will be written

$$[L(s),\ E(s),\ H(s)] = \text{BASIC}[A(s),\ D(s),\ F(s)].$$

3. Polynomial Matrix Problems

In this chapter algorithms for solving some mathematical problems involving polynomial matrices will be given. The problems are such that occur in most design procedures based on polynomial matrix theory.

3.1 Greatest Common Divisors

The theoretical background for this section is given in section 2.4 of part 2 of Pernebo (1978).

Let $A(s)$ and $B(s)$ be $n\times m$ and $n\times k$ polynomial matrices. It was mentioned in chapter 2 that there exists a unimodular matrix $N(s)$, such that

$$[A(s) \quad B(s)]N(s) = [L(s) \quad 0], \tag{3.1}$$

where $L(s)$ has linearly independent columns. Furthermore, $L(s)$ is the greatest common left divisor (g.c.l.d.) of $A(s)$ and $B(s)$.

Let $N(s)$ be partitioned as

$$N(s) = \begin{pmatrix} X(s) & N_1(s) \\ Y(s) & N_2(s) \end{pmatrix}, \tag{3.2}$$

where $X(s)$ is $m\times r$ and $r = \operatorname{rank}\,[A(s) \quad B(s)]$. Observe that $L(s)$ has r columns. It then follows from (3.1) that $L(s)$ can be expressed as

$$L(s) = A(s)X(s) + B(s)Y(s). \tag{3.3}$$

Let the inverse of $N(s)$ be partitioned as

$$N^{-1}(s) = \begin{pmatrix} A_0(s) & B_0(s) \\ N_3(s) & N_4(s) \end{pmatrix}, \tag{3.4}$$

where $A_0(s)$ is $r\times m$. It follows from (3.1) that

$$A(s) = L(s)A_0(s) \tag{3.5 a}$$
$$B(s) = L(s)B_0(s), \tag{3.5 b}$$

where $A_0(s)$ and $B_0(s)$ are relatively left prime.

The following algorithm computes the g.c.l.d. $L(s)$ of $A(s)$ and $B(s)$. Furthermore it computes the matrices $X(s)$, $Y(s)$, $A_0(s)$, and $B_0(s)$ in (3.3) and (3.5).

1. $[L(s), N(s), \hat{N}(s)] = \text{BASIC}\,[(A(s)\ B(s)), I, I]$
2. Let r be the number of columns in $L(s)$ and partition $N(s)$ as

$$N(s) = \begin{pmatrix} X(s) & N_1(s) \\ Y(s) & N_2(s) \end{pmatrix},$$

   where $X(s)$ is mxr.
3. Partition $\hat{N}(s)$ as

$$\hat{N}(s) = \begin{pmatrix} A_0(s) & B_0(s) \\ N_3(s) & N_4(s) \end{pmatrix},$$

   where $A_0(s)$ is rxm.

The algorithm will be written

$$[L(s), A_0(s), B_0(s), X(s), Y(s)] = \text{GCLD}[A(s), B(s)],$$

where $L(s)$ is the g.c.l.d. of $A(s)$ and $B(s)$ and the relations between the matrices are given by (3.3) and (3.5).

3.2 Matrix Equations

The theoretical background for this section can be found in section 2.5 and 5.4 of part 2 of Pernebo (1978). Let $A(s)$ and $B(s)$ be nxm and nxk polynomial matrices and consider the equation

$$B(s) = A(s)X(s). \tag{3.6}$$

The algorithm, given below, finds all polynomial solutions, if there are any. If there are no polynomial solutions the algorithm gives all rational solutions, if there are any. Observe that the algorithm can be used to solve equations of the type

$$B(s) = A_1(s)X_1(s) + \ldots + A_\ell(s)X_\ell(s), \tag{3.7}$$

if A(s) in (3.6) is chosen as $A(s) = [A_1(s)\ldots A_\ell(s)]$.

All rational (including all polynomial) solutions to (3.6) can be computed as is shown below (see Pernebo (1978), theorem 5.4).

Definition 3.1 Let Z(s) be an nxm rational matrix with rank r. The mx(m-r) polynomial matrix Q(s) is a prime polynomial basis for the nullspace of Z(s) if Z(s)Q(s) = 0 and the Smithform of Q is $(I_{m-r}\ \ 0)^T$.

Let $(Y_1^T(s)\ \ Y_2^T(s))^T$, where $Y_1(s)$ has k rows, be a prime polynomial basis for the nullspace of (B(s) -A(s)) and let M(s) be a unimodular matrix, such that

$$Y_1(s)M(s) = (D(s)\ \ 0), \tag{3.8}$$

where D(s) has linearly independent columns. Define K(s) and H(s) through

$$\begin{pmatrix} Y_1(s) \\ Y_2(s) \end{pmatrix} M(s) = \begin{pmatrix} D(s) & 0 \\ K(s) & H(s) \end{pmatrix}. \tag{3.9}$$

All solutions to (3.6) are given by

$$X(s) = Q(s)P^{-1}(s) \tag{3.10 a}$$

$$P(s) = D(s)R(s) \tag{3.10 b}$$

$$Q(s) = K(s)R(s) + H(s)Z(s), \tag{3.10 c}$$

where R(s) and Z(s) are arbitrary polynomial matrices with R(s) square and non-singular. It follows that there exists a rational solution if and only if D(s) is square and a polynomial solution if and only if D(s) is unimodular.

It remains to be shown how a prime polynomial basis for the nullspace of $(B(s)\ -A(s))$ can be computed. Determine a unimodular matrix $N(s)$ such that

$$(B(s)\quad -A(s))N(s) = (L(s)\quad 0), \tag{3.11}$$

where $L(s)$ has $r = \text{rank}\ (B(s)\quad -A(s))$ linearly independent columns. Partition $N(s)$ as

$$N(s) = (N_1(s)\quad N_2(s)) \tag{3.12}$$

where $N_2(s)$ has $m+k-r$ columns. The matrix $N_2(s)$ has the Smithform $(I\quad 0)^T$, since it is a part of a unimodular matrix. It follows from definition 3.1 that $N_2(s)$ is a prime polynomial basis for the nullspace of $(B(s)\quad -A(s))$.

The following algorithm computes all polynomial solutions to (3.6), if there exist any. Otherwise it computes all rational solutions in the form (3.10), if there exist any.

```
1.  [$L(s)$, $N(s)$, $\cdot$] = BASIC[$(B(s)\quad -A(s))$, $I$, $\cdot$]
2.  Let $r$ be the number of columns in $L(s)$
    If $r=m+k$ then there exists no rational solution. Stop.
3.  Partition $N(s)$ as

    $N(s) = \begin{pmatrix} E_1(s) & Y_1(s) \\ E_2(s) & Y_2(s) \end{pmatrix}$,

    where $E_1(s)$ is $k \times r$.
4.  [$D(s)$, $C(s)$, $\cdot$] = BASIC[$Y_1(s)$, $Y_2(s)$, $\cdot$]
5.  Let $\ell$ be the number of columns in $D(s)$.
    If $\ell \neq k$ then there exists no rational solution. Stop.
6.  Partition $C(s)$ as

    $C(s) = (K(s)\quad H(s))$,

    where $K(s)$ has $k$ columns.
```

7. If D(s) is not unimodular then stop.
   (If COLPRO is used in step 4 then D(s) is unimodular if and only if D(s) is independent of s. If ECHELON is used in step 4 then D(s) is unimodular if and only if the diagonal entries are independent of s).
8. $K(s):=K(s)D^{-1}(s)$.
   (If COLPRO is used in step 4 then write
   $K(s) = K_o + K_1 s + \dots + K_j s^j$ and compute
   $(\tilde{K}_o^T \ \ \tilde{K}_1^T \dots \tilde{K}_j^T) = \text{SOLV}[D^T,(K_o^T \ \ K_1^T \dots K_j^T)]$
   $K(s):=\tilde{K}_o + \tilde{K}_1 s + \dots + \tilde{K}_j s.$

   If ECHELON is used in step 4 then perform the following column operations on $(D^T(s) \ \ K^T(s))^T$ for i=k, k-1,..., 1.

   a) Divide column i with the i:th diagonal entry of D(s).
   b) Add suitable multiples of column i to the other columns so that all other entries of row i become zero.

   This gives a new K(s) and D(s) matrix, where the new D(s) is equal to I).
9. D(s):=I

The algorithm will be written

$$[D(s), K(s), H(s)] = \text{POMEQ}[A(s), B(s)]$$

and the solution to (3.6) is given by (3.10).
Observe that it follows from (3.6) and (3.10 a) that this algorithm can be used to compute P(s) and Q(s), such that

$$A^{-1}(s)B(s) = Q(s)P^{-1}(s), \tag{3.13}$$

if A(s) is square and invertible. This is a computation often required in design theory.

3.3 Factorization of Polynomial Matrices

The theoretical background to this section is found in section 2.3 of part 2 of Pernebo (1978). Let an arbitrary nxm polynomial matrix A(s) with rank r be given. Then A(s) can be factorized as

$$A(s) = \hat{A}(s)\tilde{A}(s), \tag{3.14}$$

where $\hat{A}(s)$ and $\tilde{A}(s)$ are nxr and rxm polynomial matrices. Let Λ be an arbitrary symmetric (w.r.t. the real axis) subset of the complex plane. Then the factorization can be made such that the zeros of the invariant factors of $\hat{A}(s)$ are the zeros of the invariant factors of A(s) that belong to Λ. Furthermore, the zeros of the invariant factors of $\tilde{A}(s)$ are those of the invariant factors of A(s) that do not belong to Λ.

The factorization can be obtained in the following way. Apply the scheme of section 2.3 to A(s) to obtain

$$A(s)N(s) = [L(s) \quad 0], \tag{3.15}$$

where L(s) has the form shown in figure 2.1 and r columns. Apply an analogue scheme to the rows of L(s) to obtain

$$M(s)L(s) = \begin{bmatrix} 0 \\ T(s) \end{bmatrix}, \tag{3.16}$$

where M(s) is unimodular and T(s) is an rxr lower left triangular matrix. From (3.15) and (3.16) it follows that

$$A(s) = M^{-1}(s)\begin{bmatrix} 0 & 0 \\ T(s) & 0 \end{bmatrix}N^{-1}(s). \tag{3.17}$$

Partition $M^{-1}(s)$ and $N^{-1}(s)$ as

$$M^{-1}(s) = [M_1(s) \; M_2(s)], \quad N^{-1}(s) = \begin{bmatrix} N_1(s) \\ N_2(s) \end{bmatrix}, \tag{3.18}$$

where $M_2(s)$ has r columns and $N_1(s)$ has r rows.

Then it follows that

$$A(s) = M_2(s)T(s)N_1(s). \tag{3.19}$$

The zeros of the invariant factors of A(s) are the zeros of the diagonal entries of T(s). The factorization (3.19) is a rank factorization of A(s). It remains to make a spectral factorization of T(s) as

$$T(s) = \hat{T}(s)\tilde{T}(s), \tag{3.20}$$

where all the zeros of $\det\hat{T}(s)$ belong to Λ and none of the zeros of $\det\tilde{T}(s)$ belong to Λ. This can be done in the following way. Write

$$T(s) = \begin{pmatrix} \hat{t}_1\cdot\tilde{t}_1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & & & & & & & & \cdot \\ \cdot & & \cdot & & & & & & & \cdot \\ \cdot & & & \hat{t}_{r-2}\cdot\tilde{t}_{r-2} & & & 0 & & & 0 \\ \cdot & & & t_{r-1,r-2} & & & \hat{t}_{r-1}\cdot\tilde{t}_{r-1} & & & 0 \\ t_{r,1} & & & t_{r,r-2} & & & t_{r,r-1} & & & \hat{t}_r\cdot\tilde{t}_r \end{pmatrix}, \tag{3.21}$$

where $\hat{t}_i$ has all its zeros in Λ and $\tilde{t}_i$ has all its zeros outside Λ.

Obviously, $\tilde{t}_r$ can be factored out to the right directly, leaving only $\hat{t}_r$ in position (r,r). It is not possible to do the same with $\tilde{t}_{r-1}$, because $t_{r,r-1}$ does not in general contain the factor $\tilde{t}_{r-1}$. A multiple of column r can, however, always be added to column r-1 so that the new entry in position (r,r-1) contains the factor $\tilde{t}_{r-1}$.

The equation

$$t_{r,r-1} = \hat{t}_r x + \tilde{t}_{r-1} y \tag{3.22}$$

has a solution because $\hat{t}_r$ and $\tilde{t}_{r-1}$ are relatively prime. Now, add -x times column r to column r-1. This gives $\tilde{t}_{r-1}y$ in position (r, r-1). It is now possible to factor out $\tilde{t}_{r-1}$ too to the right.

A multiple of column r-1 can now be added to column r-2 to get an entry, including the factor $\tilde{t}_{r-2}$, at position (r-1, r-2). Then a multiple of column r can be added to column r-2 to get an entry, including $\tilde{t}_{r-2}$, at position (r, r-2). Now all the entries of column r-2 include the factor $\tilde{t}_{r-2}$ and it can be factored out to the right.

It is obviously possible to continue like this and factor out all $\{\tilde{t}_i\}$ to the right. The result is a factorization of the type (3.20).

A factorization algorithm will be given below. An algorithm for finding the unique solution, with deg x < deg b, to the polynomial equation

$$c = ax + by \tag{3.23}$$

is assumed to be available.

The following algorithm makes a spectral factorization of an rxr lower left triangular polynomial matrix T(s). The right factor $\tilde{T}(s)$ in (3.20) is multiplied by an arbitrary rxℓ polynomial matrix B(s) giving C(s), i.e.

$$T(s)B(s) = \hat{T}(s)C(s). \tag{3.24}$$

The choice B(s) = I thus gives C(s) = $\tilde{T}(s)$. Let b_k be row k of B(s) and t_{ij} the entry at position (i,j) in T(s).

```
1.  Compute the zeros of the diagonal entries $\{t_{ii}\}$.
2.  Factorize $\{t_{ii}\}$ as $t_{ii} = \hat{t}_i \cdot \tilde{t}_i$ for i=1,...,r, where all zeros of $\hat{t}_i$ belong
    to $\Lambda$ and no zeros of $\tilde{t}_i$ belong to $\Lambda$ (see remark below). If $t_{ii}$ has no zeros
    in $\Lambda$ then choose $\tilde{t}_i = 1$ and $\hat{t}_i = t_{ii}$. Otherwise $\hat{t}_i$ is chosen monic.
3.  k=r
4.  $t_{kk} := \hat{t}_k$
5.  $b_k := \tilde{t}_k b_k$
6.  If k=1 then go to 16.
7.  k:=k-1
8.  If deg $\tilde{t}_k$=0 then go to 6.
9.  j=k+1
10. Solve $t_{jk} = t_{jj}x + \tilde{t}_k y$ with deg x < deg $\tilde{t}_k$.
11. $t_{jk} := y$
12. $b_j := b_j + x b_k$
13. If j=r then go to 4.
14. $t_{ik} := t_{ik} - x t_{ij}$ for i = j+1, ..., r.
15. j:=j+1 and go to 10.
16. Let $\hat{T}(s)$ and C(s) be the results of the transformation on T(s) and B(s).
```

The algorithm will be written

$[\hat{T}(s), C(s)] = \text{FACTRI}[T(s), B(s)],$

where the matrices are related via (3.20) and (3.24).

Remark In step 2 all the zeros of $\{t_{ii}\}$ are displayed to the user, who marks the ones that belong to Λ.

An algorithm for permutation of the columns of an nxm polynomial matrix A(s) is assumed to be available. The resulting matrix B(s) is obtained as

$$b_i = a_{m-i+1}, \; i=1,\dots, m, \tag{3.25}$$

where a_j and b_j are columns j in A(s) and B(s). The algorithm is written

$B(s) = \text{PERM}[A(s)].$

The following algorithm makes the rank and spectral factorization, shown in (3.14), of an nxm polynomial matrix A(s).

```
1.  $[L(s), \cdot, \hat{N}(s)] = \text{ECHELON}[A(s), \cdot, I]$

2.  $L_1(s) = \text{PERM}[L(s)]$

3.  $[T_1(s), \cdot, M_1(s)] = \text{ECHELON}[L_1^T(s), \cdot, I]$

4.  $T_2(s) = \text{PERM}[T_1(s)]$

    $\hat{M}(s) = \text{PERM}[M_1^T(s)]$

    $T(s) = \text{PERM}[T_2^T(s)]$

5.  Let r be the number of columns in L(s).
    $N_1(s)$ is the r first rows of $\hat{N}(s)$
    $M_2(s)$ is the r last columns of $\hat{M}(s)$.
```

6. $[\hat{T}(s), \tilde{A}(s)] = \mathrm{FACTRI}[T(s), N_1(s)]$

7. $\hat{A}(s) = M_2(s)\hat{T}(s)$

The algorithm will be written

$$[\hat{A}(s), \tilde{A}(s)] = \mathrm{RASPFA}[A(s)],$$

where $A(s)$, $\hat{A}(s)$ and $\tilde{A}(s)$ are related via (3.14).

If only the zeros of the invariant factors of $A(s)$ are of interest the algorithm can be simplified considerably as follows.

1. $[L(s), \cdot\, , \cdot] = \mathrm{ECHELON}[A(s), \cdot, \cdot]$
2. $L_1(s) = \mathrm{PERM}[L(s)]$
3. $[T_1(s), \cdot\, , \cdot] = \mathrm{ECHELON}[L_1^T(s), \cdot, \cdot]$
4. Compute the vector z of zeros to the diagonal entries of $T_1(s)$.

The algorithm will be written

$$z = \mathrm{INVZER}[A(s)].$$

Observe that if $A(s)$ is square and nonsingular then the zeros of the invariant factors are the zeros of $\det A(s)$. The algorithm can thus be used to calculate the zeros of $\det A(s)$.

4. Linear Dynamical Systems

In this chapter dynamical systems of the type

$$T(\mu)\xi = U(\mu)u \tag{4.1 a}$$
$$y = V(\mu)\xi + W(\mu)u \tag{4.1 b}$$

will be considered. Here $T(\mu)$, $U(\mu)$, $V(\mu)$ and $W(\mu)$ are polynomial matrices of dimension $r\mathrm{x}r$, $r\mathrm{x}\ell$, $m\mathrm{x}r$ and $m\mathrm{x}\ell$ and $T(\mu)$ is nonsingular. The operator μ is the differential operator p for continuous time systems and the forward shift operator q for discrete time systems. Systems of this type are analysed in detail in Rosenbrock (1970).

Algorithms to analyse and transform the system (4.1) will be given. An algorithm to find the inverse system will also be presented.

4.1 Transformation to State Space Form

Any system of the type (4.1) can be transformed to a system in state space form. It was shown in Rosenbrock (1970) that this can be done with transformation of strict system equivalence (s.s.e.). In this section it is assumed that the equations (4.1) are obtained from basic physical relations. Therefore, the internal vaiable ξ, the input u and the output y are assumed to have physical interpretations. Furthermore, it is assumed that it is desirable to have a simple relation between ξ and the obtained state variable x. In such a case it is possible to give x a physical interpretation as well. A state space system, which is s.s.e. to a system of the type (4.1), can be computed with the method of this section. Furthermore, the state variable x can be chosen such that it is related in a very simple way to the internal variable ξ. In the description of the method the system is assumed to be a continuous time system so that $\mu = p$, the differential operator. The method can of course formally be applied to discrete time systems as well. Derivatives should then be interpreted as forward time shifts. For discrete time systems, however, the equations obtained from basic physical relations are likely to contain the backward shift operator q^{-1}. The system will then not be in the form (4.1). It can be brought to the form (4.1), but not with transformations of s.s.e. Therefore it is questionable whether a method based on s.s.e. should be used on a discrete time system. This topic is discussed in detail in part 1 of Pernebo (1978). The method for transformation to state space form is described below.

There exists a system of the type

$$(Ep - F)z = Q(p)u \tag{4.2 a}$$

$$y = P(p)z + W(p)u, \tag{4.2 b}$$

which is s.s.e. to (4.1). The matrices E, F, Q(p), P(p), and W(p) and the internal variable z are given below.

Let d_j be the column degree of column j in T(p). Suppose for the moment that all d_j are nonzero and let

$$t^{ij}(p) = t^{ij}_{d_j}p^{d_j} + t^{ij}_{d_j-1}p^{d_j-1} + \dots + t^{ij}_o \tag{4.3}$$

be the element at position (i,j) in T(p). Write the (i,j) element of V(p) as

$$v^{ij}(p) = v^{ij}_{d_j-1}(p)p^{d_j-1} + v^{ij}_{d_j-2}p^{d_j-2} + \dots + v^{ij}_o, \tag{4.4}$$

where $v^{ij}_{d_j-1}(p)$ is polynomial but $v^{ij}_{d_j-2}, \dots, v^{ij}_o$ are real numbers. If all $d_j>0$ define the real matrices E and F as

$$E = \left[\begin{array}{cccc|cccc|cccc}
1 & & & & 0 & & & 0 & & & & \\
 & \cdot & & & & & & & & & & \\
 & & \cdot & & & & & & & & & \\
 & & & 1 & & & & 0 & & & & \\
0 & & & t^{11}_{d_1} & 0 & \dots & 0 & t^{12}_{d_2} & & & & \\
\hline
0 & & & 0 & 1 & & & 0 & & & & \\
\cdot & & & & & \cdot & & & & & & \\
\cdot & & & & & & \cdot & & & & & \\
 & & & 0 & & & & & & & & \\
0 & & & t^{21}_{d_1} & 0 & & & t^{22}_{d_2} & & & & \\
\hline
 & & & & & & & & 1 & & & \\
 & & & & & & & & & \cdot & & \\
 & & & & & & & & & & \cdot & \\
 & & & & & & & & & & & 1 \\
 & & & & & & & & & & & t_{d_r}
\end{array}\right] \tag{4.5}$$

$$F = \left[\begin{array}{ccc:ccc:ccc}
0 & 1 & & 0 & \cdots & 0 & & & \\
 & \ddots & \ddots & \vdots & & \vdots & & & \\
 & 0 & 1 & 0 & \cdots & 0 & & & \\
-t_o^{11} & & -t_{d_1-1}^{11} & -t_o^{12} & & -t_{d_2-1}^{12} & & & \\
\hdashline
0 & \cdots & 0 & 0 & 1 & & & & \\
\vdots & & \vdots & & \ddots & \ddots & & & \\
0 & \cdots & 0 & & 0 & 1 & & & \\
-t_o^{21} & \cdots & -t_{d_1-1}^{21} & -t_o^{22} & & -t_{d_2-1}^{22} & & & \\
\hdashline
 & & & & & & 0 & 1 & \\
 & & & & & & & \ddots & \ddots \\
 & & & & & & & 0 & 1 \\
 & & & & & & -t_o^{rr} & & -t_{d_r-1}^{rr}
\end{array}\right] \tag{4.6}$$

where block row (and column) i in both E and F contains d_i rows (columns). Define the polynomial matrices P(p) and Q(p) as

$$P(p) = \left[\begin{array}{ccc:cc:c:cc}
v_o^{11} \cdots & v_{d_1-2}^{11} & v_{d_1-1}^{11}(p) & v_o^{12} \cdots & v_{d_2-1}^{12}(p) & \cdots & v_o^{1r} \cdots & v_{d_r-1}^{1r}(p) \\
\vdots & & & & & & & \\
v_o^{m1} \cdots & v_{d_1-2}^{m1} & v_{d_1-1}^{m1}(p) & v_o^{m2} \cdots & \cdots & \cdots & \cdots & v_{d_r-1}^{mr}(p)
\end{array}\right] \tag{4.7}$$

$$Q(P) = \left[\begin{array}{c}
0 \\ \vdots \\ 0 \\ \underline{U_1}(p) \\
\hdashline
0 \\ \vdots \\ \underline{U_2}(p) \\
\hdashline
\vdots \\ U_r(p)
\end{array}\right] \tag{4.8}$$

where block row i in Q(p) contains d_i rows. Define the internal variable $z^T = (\xi_1\ \xi_1^{(1)} \ldots \xi_1^{(d_1-1)} \xi_2 \ldots \xi_2^{(d_2-1)} \ldots \xi_r^{(d_r-1)})^T$, where $\xi_i^{(j)}$ is the j:th derivative of component i in ξ.

If some of the column degrees, say d_k, is zero then the definitions of E, F, P(p) and Q(p) are valid with the following modifications. Write the elements in column k of T(p) and V(p) as t^{ik} and $v^{ik}(p)$ respectively. Block row k of E, F and Q(p) will contain one row each and is defined as in (4.5), (4.6), and (4.8). Block column k of E, F, and P(p) will contain one column each. All elements in block column k of E will be zero. Block column k of F will, for all i, have $-t^{ik}$ in the last row of block row i and zero in the other rows of block row i. Block column k of P(p) will have $v^{ik}(p)$ in row i for all i. The internal variable z will have ξ_k as the only element in block row k.

The system (4.2) can now be brought to state space form using the following operations of s.s.e.

Multiply E_p - F from the left by a unimodular matrix $N_1(p)$, such that the result becomes row proper and multiply Q(p) by the same matrix. This gives

$$N_1(p)(E_p - F) \triangleq \begin{pmatrix} E_1 p - F_1 \\ F_2 \end{pmatrix}, \quad N_1(p)Q(p) \triangleq \begin{pmatrix} Q_1(p) \\ Q_2(p) \end{pmatrix}. \tag{4.9}$$

If E is nonsingular then F_2 and $Q_2(p)$ disappear. In the sequel it is assumed that E is singular. If E is nonsingular then it is easy to see how the method can be simplified.

Let H be a permutation matrix and make the change of variables

$$v = H^{-1} z \tag{4.10}$$

so that F_{22} in

$$\begin{pmatrix} E_1 p - F_1 \\ F_2 \end{pmatrix} H \triangleq \begin{pmatrix} E_{11} p - F_{11} & E_{12} p - F_{12} \\ F_{21} & F_{22} \end{pmatrix} \tag{4.11}$$

becomes square and nonsingular. This is possible because the rows of F_2 are linearly independent.

P(p) will then be transformed to

$$P_1(p) = P(p)H. \tag{4.12}$$

Multiply (4.11) and $N_1(p)Q(p)$ from the left by the unimodular matrix

$$N_2(p) = \begin{pmatrix} I & -(E_{12}p-F_{12})F_{22}^{-1} \\ 0 & F_{22}^{-1} \end{pmatrix}.$$

This gives

$$N_2(p)\begin{pmatrix} E_{11}p-F_{11} & E_{12}p-F_{12} \\ F_{21} & F_{22} \end{pmatrix} = \begin{pmatrix} E_{11}p-F_{11} - (E_{12}p-F_{12})F_{22}^{-1}F_{21} & 0 \\ F_{22}^{-1}F_{21} & I \end{pmatrix} \triangleq$$

$$\triangleq \begin{pmatrix} \bar{E}p - \bar{F} & 0 \\ G & I \end{pmatrix} \tag{4.13}$$

and

$$\begin{pmatrix} Q_3(p) \\ Q_4(p) \end{pmatrix} \triangleq N_2(p)\begin{pmatrix} Q_1(p) \\ Q_2(p) \end{pmatrix} \tag{4.14}$$

It follows that $\bar{E} = E_{11} - E_{12}F_{22}^{-1}F_{21}$ is nonsingular since

$$\det\bar{E}\det F_{22} = \det\begin{pmatrix} E_{11}-E_{12}F_{22}^{-1}F_{21} & E_{12} \\ 0 & F_{22} \end{pmatrix} =$$

$$= \det\begin{pmatrix} E_{11} & E_{12} \\ F_{21} & F_{22} \end{pmatrix} = \det\begin{pmatrix} E_1 \\ F_2 \end{pmatrix} \neq 0$$

because $\begin{pmatrix} E_1p-F_1 \\ F_2 \end{pmatrix}$ is row proper.

Multiply (4.13) and (4.14) from the left by

$$N_3 = \begin{pmatrix} \bar{E}^{-1} & 0 \\ 0 & I \end{pmatrix}$$

This gives

$$N_3\begin{pmatrix} \bar{E}p-\bar{F} & 0 \\ G & I \end{pmatrix} \triangleq \begin{pmatrix} pI-A & 0 \\ G & I \end{pmatrix} \tag{4.15}$$

$$\begin{pmatrix} Q_5(p) \\ Q_4(p) \end{pmatrix} \triangleq N_3\begin{pmatrix} Q_3(p) \\ Q_4(p) \end{pmatrix} \tag{4.16}$$

where the partitioning in (4.16)is compatible with that in (4.15). If $P_1(p)$ is partitioned as $P_1(p) = (P_2(p) \; P_3(p))$ it follows that

$$\begin{pmatrix} pI-A & 0 \\ G & I \end{pmatrix}\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} Q_5(p) \\ Q_4(p) \end{pmatrix}u \tag{4.17 a}$$

$$y = (P_2(p) \quad P_3(p))\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} + W(p)u \tag{4.17 b}$$

which is s.s.e. to the original system.

Use the division algorithm to obtain $Y(p)$ and B such that

$$Q_5(p) = (pI-A)Y(p) + B \tag{4.18}$$

and make the change of variables

$$\begin{pmatrix} x \\ x_0 \end{pmatrix} = \begin{pmatrix} I & 0 \\ G & I \end{pmatrix}\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} + \begin{pmatrix} -Y(p) \\ -Q_4(p) \end{pmatrix}u \tag{4.19}$$

Introducing (4.19) into (4.17) gives

$$\begin{pmatrix} pI-A & 0 \\ 0 & I \end{pmatrix}\begin{pmatrix} \dot{x} \\ \dot{x}_o \end{pmatrix} = \begin{pmatrix} B \\ 0 \end{pmatrix}u \qquad (4.20\ a)$$

$$y = (P_4(p) \quad P_3(p))\begin{pmatrix} x \\ x_o \end{pmatrix} + W_1(p)u \qquad (4.20\ b)$$

where

$$P_4(p) = P_2(p) - P_3(p)G \qquad (4.21)$$

$$W_1(p) = W(p) + P_2(p)Y(p) + P_3(p)Q_4(p) - P_3(p)GY(p). \qquad (4.22)$$

Define X(p) and C through

$$P_4(p) = X(p)(pI-A) + C. \qquad (4.23)$$

Add $(-X(p) \quad -P_3(p))$ times (4.20 a) to (4.20 b).

This gives

$$y = (C \quad 0)\begin{pmatrix} x \\ x_o \end{pmatrix} + D(p) \qquad (4.24)$$

where

$$D(p) = W_1(p) + X(p)B. \qquad (4.25)$$

It follows that the system

$$(pI-A)x = Bu \qquad (4.26\ a)$$
$$y = Cx + D(p)u \qquad (4.26\ b)$$

is s.s.e. to (4.1). The state variable x is given by

$$x = (I \quad 0)Hz^{-1} - Y(p)u \qquad (4.27)$$

The inverse of (4.27) is obtained from (4.19)

$$z = H\begin{pmatrix} I \\ -G \end{pmatrix}x + H\begin{pmatrix} Y(p) \\ Q_4(p)-GY(p) \end{pmatrix}u \triangleq Kx + L(p)u \tag{4.28}$$

because $x_0 = 0$ by (4.20 a).

It is shown in Rosenbrock (1970) that $\bar{x}$ is the state variable of a system in state space form which is s.s.e. to (4.1) if and only if there is a non-singular T such that

$$\bar{x} = Tx. \tag{4.29}$$

Introduce (4.29) into (4.28)

$$KT^{-1}\bar{x} = z - L(p)u \tag{4.30}$$

This means that any state variable for a system s.s.e. to (4.1) must fulfil (4.30) for some nonsingular T. From (4.30) it follows that a component of $-L(p)u$ is the linear combination of u and its derivatives that must be added to the corresponding component of z in order to make it a state variable.

Now, suppose that it is desirable that each component of the state variable $\bar{x}$ is equal to one component of z plus, if necessary, a linear combination of u and its derivatives. This is obtained by the choice procedure described below.

Let K_i and $L_i(p)$ be the i:th row of K and L(p), respectively, and let z_i be the i:th component of z. Choose one of the $z_i-L_i(p)u$ with $K_i \neq 0$ as the first state variable. Say that the i_1:th $z_i-L_i(p)u$ was chosen. Any of the $z_i-L_i(p)u$ with K_i outside the row space of K_{i_1} can now be chosen as the second state variable. Say that the i_2:th was chosen. Any of the $z_i-L_i(p)u$ with K_i outside the row space of K_{i_1} and K_{i_2} can then be chosen as the third state variable, and so on.

Let n be the dimension of $\bar{x}$. Then it is possible to choose n $z_i-L_i(p)u$ in this way since K by (4.28) has rank n. All the chosen K_{i_j} are linearly independent.

It follows from (4.30) that

$$\begin{bmatrix} K_{i_1} \\ \vdots \\ K_{i_n} \end{bmatrix} T^{-1}\bar{x} = \begin{bmatrix} z_{i_1} \\ \vdots \\ z_{i_n} \end{bmatrix} - \begin{bmatrix} L_{i_1}(p) \\ \vdots \\ L_{i_n}(p) \end{bmatrix} u \tag{4.31}$$

Choosing $T = \begin{bmatrix} K_{i_1} \\ \vdots \\ K_{i_n} \end{bmatrix}$ gives the desired state variables

$$\bar{x} = \begin{bmatrix} z_{i_1} \\ \vdots \\ z_{i_n} \end{bmatrix} - \begin{bmatrix} L_{i_1}(p) \\ \\ L_{i_n}(p) \end{bmatrix} u \tag{4.32}$$

The corresponding state space system is obtained by making the coordinate change (4.29) on the system (4.26).

Other choice procedures are possible. If the state variables are allowed to contain linear combinations of the z_i:s then (4.30) can be multiplied by a nonsingular matrix M

$$\bar{K}T^{-1}\bar{x} = Mz - ML(p)u \tag{4.33}$$

where $\bar{K} = MK$. The same procedure can now be used with (4.33) substituted for (4.30). The matrix M can for instance be used to decrease the row degrees of L(p).

An algorithm for transformation to state space form is given below. It is assumed that an algorithm, which makes the transformation

$$\bar{A} = TAT^{-1}, \bar{B} = TB, \bar{C} = CT^{-1}$$

of a state space system, is available. This algorithm will be written

$$[\bar{A}, \bar{B}, \bar{C}] = TSYS[A, B, C, T].$$

Furthermore, the division algorithm is needed. Let $P(s)$ be an nxm polynomial matrix and A an nxn real matrix. Then there is an nxm polynomial matrix $Q(s)$ and an nxm real matrix Z, such that

$$P(s) = (sI-A)Q(s) + Z.$$

Write

$$P(s) = P_k s^k + \ldots + P_o.$$

Then $Q(s)$ can be written

$$Q(s) = Q_{k-1}s^{k-1} + \ldots + Q_o.$$

The following algorithm computes $Q(s)$ and Z.

1. If $k=0$ then $Q(s) = 0$ and $Z = P_o$. Stop.
2. $Q_{k-1} = P_k$
3. $Q_i = P_{i+1} + AQ_{i+1}$ for $i = k-2,\ldots,0$.
4. $Z = P_o + AQ_o$.

The algorithm will be written

$$[Q(s), Z] = \mathrm{DIV}[P(s), A].$$

The following algorithm transforms a given system of the form (4.1) to state space form, using the method of this section.

1. Compute the column degrees $\{d_i\}$ of $T(p)$ and form the matrices E, F, $P(p)$ and $Q(p)$ given by (4.5) - (4.8).

2. $[E_o^Tp - F_o^T, Q_o^T(p), \cdot] = \mathrm{COLPRO}[E^Tp - F^T, Q^T(p), \cdot]$

   Let n be the number of nonzero rows of E_o.

3. If all rows of E_o are nonzero then put $\bar{E} = E_o$, $\bar{F} = F_o$, $Q_3(p) = Q_o(p)$, $P_4(p) = P(p)$, $\gamma = 1$ and go to 10, else put $\gamma = 0$.

4. Permute the rows of $[E_o p - F_o \quad Q_o(p)]$ so that it becomes

$$\begin{bmatrix} \begin{pmatrix} E_1 p - F_1 \\ F_2 \end{pmatrix} & \begin{pmatrix} Q_1(p) \\ Q_2(p) \end{pmatrix} \end{bmatrix}$$

with E_1 having linearly independent rows.

5. $\eta = LD[\emptyset, F_2]$, where $\emptyset$ denotes an empty matrix.

6. Let H be a permutation matrix such that $F_2H = [F_{21} \quad F_{22}]$ where F_{22} consists of the columns of F_2 having $\eta_i = 0$ and form

$$\begin{pmatrix} E_{11}p - F_{11} & E_{12}p - F_{12} \\ F_{21} & F_{22} \end{pmatrix} = \begin{pmatrix} E_1 p - F_1 \\ F_2 \end{pmatrix} H \text{ and}$$

$P_1(p) = P(p)H.$

7. Write $Q_2(p) = Q_2^k p^k + \dots + Q_2^o$ and compute

$(G \quad Q_4^k \dots Q_4^o) = SOLV[F_{22}, (F_{21} \quad Q_2^k \dots Q_2^o)]$

and form $Q_4(p) = Q_4^k p^k + \dots + Q_4^o$.

8. $\bar{E} = E_{11} - E_{12}G \quad \bar{F} = F_{11} - F_{12}G$

9. $Q_3(p) = Q_1(p) - (E_{12}p - F_{12})Q_4(p)$

10. Write $Q_3(p) = Q_3^k p^k + \dots + Q_3^o$ and compute

$(A \quad Q_5^k \dots Q_5^o) = SOLV[\bar{E}, (\bar{F} \quad Q_3^k \dots Q_3^o)]$

and form $Q_5(p) = Q_5^k p^k + \dots + Q_5^o$.

11. $[Y(p), B] = DIV[Q_5(p), A]$. If $\gamma = 1$ go to 13.

12. Partition $P_1(p) = [P_2(p) \quad P_3(p)]$, where $P_2(p)$ has the same number of columns as F_{21}, and compute $P_4(p) = P_2(p) - P_3(p)G$.

13. $[X^T(p),\ C^T] = \mathrm{DIV}\ [P_4^T(p),\ A^T]$

14. If $\gamma = 0$ then $D(p) = W(p) + P_2(p)Y(p) + P_3(p)Q_4(p) - P_3(p)GY(p) + X(p)B$
    If $\gamma = 1$ then $D(p) = W(p) + P(p)Y(p) + X(p)B$.

15. If $\gamma = 0$ then

    $$K = H\begin{pmatrix} I \\ -G \end{pmatrix} \quad \text{and } L(p) = H\begin{pmatrix} Y(p) \\ Q_4(p) - GY(p) \end{pmatrix}$$

    If $\gamma = 1$ then $K = I_n$ and $L(p) = Y(p)$.

The algorithm will be written

$$[A,\ B,\ C,\ D(p),\ K,\ L(p),\ n] = \mathrm{CSTATE}[T(p),\ U(p),\ V(p),\ W(p)],$$

where A, B, C, D(p) is a state space representation of order n, of the system (4.1). The relation between the internal variable of (4.1) and the state variable is given by (4.28).

The algorithm CSTATE computes a state space representation and the relation between the state variable and the original internal variable. The following algorithm computes linear combinations of the internal variables. The linear combinations are determined so that the derivatives of u, that must be added to make them state variables, are of lower order than for each of the included internal variables. The user of the algorithm can then, in an interactive manner, choose state variables from these linear combinations and the original internal variables. The user can also construct other linear combinations and it is tested if these are valid as state variables.

In the following algorithm the internal variable z and the input u occur. They should be interpreted as vectors of symbols, where each symbol stands for the corresponding component of z or u. Furthermore, each component of e.g. L(p)u, where L(p) is a polynomial matrix, is a linear combination of the symbols for the components of u and their derivatives.

1. Let ℓ be the highest degree in L(p).

2. $\bar{L}(p) := L(p)$, $\bar{K} := K$ and $\bar{z} := z$, where z is the internal variable of the system (4.8)

3. $i := \ell$

4. Define $\hat{L}(p)$ as those rows of $\bar{L}(p)$ that have row degree equal to i. Let $\hat{K}$ be the corresponding rows of $\bar{K}$ and $\hat{z}$ the corresponding rows of $\bar{z}$.

5. Let $\hat{L}$ be the coefficient matrix of p^i in $\hat{L}(p)$.

6. Compute a basis X for the orthogonal complement of the columns of $\hat{L}$.

7. $\bar{L}(p) := \begin{pmatrix} \bar{L}(p) \\ X^T\hat{L}(p) \end{pmatrix}$, $\bar{K} := \begin{pmatrix} \bar{K} \\ X^T\hat{K} \end{pmatrix}$ and $\bar{z} := \begin{pmatrix} \bar{z} \\ X^T\hat{z} \end{pmatrix}$

8. $i := i-1$

9. If $i \geq 0$ then go to 4.

10. Let K_i and $L_i(p)$ be the i:th row of $\bar{K}$ and $\bar{L}(p)$.

11. Define $M = \{i \mid K_i \neq 0\}$

12. $k := 1$

13. Display the vector $v = \bar{z} - \bar{L}(p)u$ to the user and mark the components that belong to M.

14. The user chooses a linear combination $x_k = \sum_j a_j v_j$ as his k:th state variable. (Observe that at least one of the v_j in x_k must have index j in M).

15. $\bar{K}_k = \sum_j a_j K_j$

16. $\eta = LD[(\bar{K}_1^T \ldots \bar{K}_{k-1}^T), \bar{K}_k^T]$

17. If $\eta = 1$ then notify the user that this is an invalid choice of state variable and go to 14.

18. If $k=n$ then go to 22.

19. Compute $\eta_i = \mathrm{LD}[(\bar{K}_1^T \ldots \bar{K}_k^T), K_i^T]$, $\forall i \in M$.

20. Delete $\{i | \eta_i = 1\}$ from M.

21. k:=k+1 and go to 13.

22. $$T = \begin{pmatrix} \bar{K}_1 \\ \vdots \\ \bar{K}_n \end{pmatrix}$$

23. $[\bar{A}, \bar{B}, \bar{C}] = \mathrm{TSYS}[A, B, C, T]$

The algorithm will be written

$$[\bar{A}, \bar{B}, \bar{C}] = \mathrm{CHOICE}[A, B, C, K, L(p), n].$$

Here the input data are the output from the algorithm CSTATE. The resulting state space system is obtained as $\bar{A}$, $\bar{B}$ and $\bar{C}$ from the algorithm CHOICE and D(p) from the algorithm CSTATE.

4.2 Controllable or Observable Systems

Consider the system (4.1) and assume that the matrices $T(\mu)$ and $U(\mu)$ are relatively prime. This means that the system is controllable. Introduce the system matrix (c.f. Rosenbrock (1970))

$$P(\mu) = \begin{pmatrix} T(\mu) & U(\mu) \\ -V(\mu) & W(\mu) \end{pmatrix}. \qquad (4.34)$$

It can be shown that $P(\mu)$ is s.s.e. to the system matrix

$$P_o(\mu) = \begin{pmatrix} Q(\mu) & I \\ -R(\mu) & 0 \end{pmatrix}. \qquad (4.35)$$

It will be shown how $P_o(\mu)$ can be computed from $P(\mu)$.

Since $T(\mu)$ and $U(\mu)$ are relatively left prime there is a unimodular $N(\mu)$, such that

$$[T(\mu) \quad -U(\mu)]N(\mu) = [I \quad 0] \qquad (4.36)$$

Partition $N(\mu)$ as

$$N(\mu) = \begin{pmatrix} N_1(\mu) & N_2(\mu) \\ N_3(\mu) & N_4(\mu) \end{pmatrix}, \tag{4.37}$$

where $N_1(\mu)$ is rxr.

Form the system matrix

$$P_1(\mu) = \left(\begin{array}{cc|c} I_\ell & 0 & 0 \\ 0 & T(\mu) & U(\mu) \\ \hline 0 & -V(\mu) & W(\mu) \end{array}\right)$$

Compute

$$P_2(\mu) \triangleq \left(\begin{array}{cc|c} I & 0 & I \\ -U(\mu) & T(\mu) & 0 \\ \hline 0 & -V(\mu) & W(\mu) \end{array}\right) = \left(\begin{array}{cc|c} I & 0 & 0 \\ -U(\mu) & I & 0 \\ \hline 0 & 0 & I \end{array}\right) P_1(\mu) \left(\begin{array}{cc|c} I & 0 & I \\ 0 & I & 0 \\ \hline 0 & 0 & I \end{array}\right)$$

$$P_3(\mu) \triangleq \left(\begin{array}{cc|c} T(\mu) & -U(\mu) & 0 \\ 0 & I & I \\ \hline -V(\mu) & 0 & W(\mu) \end{array}\right) = \left(\begin{array}{cc|c} 0 & I & 0 \\ I & 0 & 0 \\ \hline 0 & 0 & I \end{array}\right) P_2(\mu) \left(\begin{array}{cc|c} 0 & I & 0 \\ I & 0 & 0 \\ \hline 0 & 0 & I \end{array}\right)$$

$$P_4(\mu) \triangleq \left(\begin{array}{cc|c} I & 0 & 0 \\ N_3(\mu) & N_4(\mu) & I \\ \hline -V_1(\mu) & -V_2(\mu) & W(\mu) \end{array}\right) = P_3(\mu) \left(\begin{array}{cc|c} N_1(\mu) & N_2(\mu) & 0 \\ N_3(\mu) & N_4(\mu) & 0 \\ \hline 0 & 0 & I \end{array}\right)$$

Here $V_1(\mu) = V(\mu)N_1(\mu)$ and $V_2(\mu) = V(\mu)N_2(\mu)$.

$$P_5(\mu) \triangleq \left(\begin{array}{cc|c} I & 0 & 0 \\ 0 & N_4(\mu) & I \\ \hline 0 & -V_2(\mu) & W(\mu) \end{array}\right) = \left(\begin{array}{cc|c} I & 0 & 0 \\ -N_3(\mu) & I & 0 \\ \hline V_1(\mu) & 0 & I \end{array}\right) P_4(\mu)$$

$$P_6(\mu) \triangleq \left(\begin{array}{cc|c} I & 0 & 0 \\ 0 & Q(\mu) & I \\ \hline 0 & -R(\mu) & 0 \end{array}\right) = \left(\begin{array}{cc|c} I & 0 & 0 \\ 0 & I & 0 \\ \hline 0 & -W(\mu) & I \end{array}\right) P_5(\mu)$$

Here $Q(\mu) = N_4(\mu)$ and $R(\mu) = V_2(\mu) + W(\mu)N_4(\mu)$. The system matrix $P_o(\mu)$ is thus s.s.e. to $P(\mu)$.

The system corresponding to the system matrix $P_o(\mu)$ is

$$Q(\mu)\bar{\xi} = u \tag{4.38 a}$$
$$y = R(\mu)\bar{\xi} \tag{4.38 b}$$

It has been shown that there are polynomial matrices $H(\mu)$ and $K(\mu)$, such that

$$P_1(\mu) = H(\mu)P_6(\mu)K(\mu). \tag{4.39}$$

It was shown in Perhebo (1977) that the relation between $\bar{\xi}$, ξ and u is given by

$$\begin{pmatrix} 0 \\ \bar{\xi} \\ -u \end{pmatrix} = K(\mu) \begin{pmatrix} 0 \\ \xi \\ -u \end{pmatrix}. \tag{4.40}$$

If $N^{-1}(\mu)$ is partitioned as

$$N^{-1}(\mu) = \begin{pmatrix} M_1(\mu) & M_2(\mu) \\ M_3(\mu) & M_4(\mu) \end{pmatrix}, \tag{4.41}$$

where $M_1(\mu)$ is rxr, then $K(\mu)$ is given by

$$K(\mu) = \begin{pmatrix} M_2(\mu) & M_1(\mu) & -M_2(\mu) \\ M_4(\mu) & M_3(\mu) & -M_4(\mu) \\ 0 & 0 & I \end{pmatrix}. \tag{4.42}$$

It now follows from (4.40) and (4.42) that

$$\bar{\xi} = M_3(\mu)\xi + M_4(\mu)u. \tag{4.43}$$

The following algorithm computes a system (4.38), which is s.s.e. to a given system (4.1). It also computes the relation (4.43) between the internal variables. The relation (4.43) will be written

$$\bar{\xi} = Z(\mu)\xi + Y(\mu)u \qquad (4.44)$$

1. $[\cdot, N(\mu), M(\mu)] = \mathrm{BASIC}[(T(\mu) \;\; -U(\mu)), I, I]$

2. Partition $N(\mu)$ and $M(\mu)$ as

$$N(\mu) = \begin{pmatrix} \hat{N}_1(\mu) & N_2(\mu) \\ \hat{N}_3(\mu) & N_4(\mu) \end{pmatrix} \quad \text{and } M(\mu) = \begin{pmatrix} \hat{M}_1(\mu) & \hat{M}_2(\mu) \\ M_3(\mu) & M_4(\mu) \end{pmatrix},$$

   where $\hat{N}_1(\mu)$ and $\hat{M}_1(\mu)$ are rxr.

3. $Q(\mu) = N_4(\mu)$
   $R(\mu) = V(\mu)N_2(\mu) + W(\mu)N_4(\mu)$
   $Z(\mu) = M_3(\mu)$
   $Y(\mu) = M_4(\mu)$

The algorithm will be written

$$[Q(\mu), R(\mu), Z(\mu), Y(\mu)] = \mathrm{CON}[T(\mu), U(\mu), V(\mu), W(\mu)].$$

The algorithm can be applied to (4.1) even if it is not controllable. The resulting system of the type (4.38) will then represent only the controllable part of (4.1). The systems will not be s.s.e., but only system equivalent (Rosenbrock (1970)). The relation between the internal variables will still be given by (4.44).

If the system (4.1) is observable, then the algorithm can be used to compute a system

$$A(\mu)y = B(\mu)u, \qquad (4.45)$$

which is s.s.e. to (4.1). This is done by application of the algorithm to the dual system. In this case $Z(\mu)$ and $Y(\mu)$ no longer give the relation between the internal variables. This relation is instead given by

$$y = V(\mu)\xi + W(\mu)u, \qquad (4.1\text{ b})$$

since y is the internal variable of (4.45).

4.3 Poles, Zeros, and Decoupling Zeros

Let the system (4.1) have the system matrix

$$P(\mu) = \begin{pmatrix} T(\mu) & U(\mu) \\ -V(\mu) & W(\mu) \end{pmatrix}. \tag{4.46}$$

The poles of the system (4.1) are the zeros of $\det T(\mu)$, which in turn are the zeros of the invariant factors of $T(\mu)$. Therefore, the algorithm INVZER in section 3.3 can be used to compute the poles. Let the components of the complex vector p be the poles of the system. Then p is given by

$$p = \text{INVZER}[T(\mu)].$$

The zeros of the system (4.1) are defined as the zeros of the invariant factors of the system matrix $P(\mu)$. Let the components of the complex vector z be the zeros of the system. Then z is given by

$$z = \text{INVZER}[P(\mu)].$$

The input decoupling zeros (i.d.z.) of the system are the zeros of the determinant of the g.c.l.d. of $T(\mu)$ and $U(\mu)$. They can therefore be computed in the following way.

1. $[L(\mu), \cdot, \cdot] = \text{ECHELON}[(T(\mu) \quad U(\mu)), \cdot, \cdot]$

2. Compute the zeros of the diagonal entries of $L(\mu)$.
   These zeros are the i.d.z. of the system.

The output decoupling zeros (o.d.z.) are computed analogously. The input-output decoupling zeros (i.o.d.z.) are those o.d.z. that disappear when all the i.d.z. are removed. They can be computed in the following way.

1. $[\cdot, T_0(\mu), \cdot, \cdot, \cdot] = \text{GCLD}[T(\mu), U(\mu)]$

2. Compute the o.d.z., called $\{\gamma_i\}$, of $T(\mu)$ and $V(\mu)$.

3. Compute the o.d.z., called $\{\theta_i\}$, of $T_0(\mu)$ and $V(\mu)$.

4. The i.o.d.z. are given by $\{\gamma_i\} \setminus \{\theta_i\}$.

4.4 Change of Operators

Consider the following change of operators in the system (4.1).

$$\lambda = \frac{a\mu + b}{c\mu + d}, \qquad ad \neq bc \tag{4.47}$$

This includes the change from the forward shift operator q to the backward shift operator q^{-1} and vice versa. It also includes the change of operators that is required in the design method of Pernebo (1978). The transformation (4.47) may also be used in connection with stability check, like e.g. the Routh algorithm.

All transformations of the type (4.47) can be achieved with repeated use of the following two transformations

$$\lambda = a\mu + b, \; a \neq 0 \tag{4.48}$$

and

$$\lambda = \frac{1}{\mu}. \tag{4.49}$$

Let the polynomial

$$p(\mu) = p_k\mu^k + \ldots + p_o \tag{4.50}$$

be given. The following algorithm computes

$$q(\lambda) = q_k\lambda^k + \ldots + q_o \triangleq p(\frac{1}{a}\lambda - \frac{b}{a}), \tag{4.51}$$

```
1.  Compute $\alpha = \frac{1}{a}$ and $\beta = \frac{b}{a}$.

2.  $i:=k-1, \; q(\lambda):=p_k$

3.  If $i \leqslant -1$ then stop.

4.  $q(\lambda):=(\alpha\lambda-\beta)\cdot q(\lambda)+p_i$

5.  $i:=i-1$ and go to 3.
```

The algorithm will be written

$$q(\lambda) = \mathrm{POTRAN}[p(\mu), a, b].$$

The following algorithm computes the polynomial

$$r(\lambda) = \lambda^d p(\frac{1}{\lambda}), \ d \geqslant k. \tag{4.52}$$

1. $r_{d-i} = p_i$ for $i = 0, 1, \ldots, k$

2. $r_{d-i} = 0$ for $i = k+1, \ldots, d$.

The algorithm will be written

$$r(\lambda) = \mathrm{POINV}[p(\mu), d].$$

Only systems of the type

$$A(\mu)y = B(\mu)u \tag{4.53}$$

will be considered. Here $A(\mu)$ is mxm and $B(\mu)$ is $m\mathrm{x}\ell$.

An algorithm, which makes the change of operators (4.48), is obtained if the algorithm POTRAN is applied to all entries of $A(\mu)$ and $B(\mu)$. The algorithm will be written

$$[\bar{A}(\lambda), \bar{B}(\lambda)] = \mathrm{TRAN}[A(\mu), B(\mu), a, b],$$

where λ is given by (4.48).

The following algorithm makes the change of operators (4.49) and multiplies each row of (4.53) by a suitable power of λ to make the result polynomial.

1. $i:=1$

2. Compute the row degree d_i of row i of $[A(\mu) \ B(\mu)]$.

3. Compute $a_{ij}^*(\lambda) = \mathrm{POINV}[a_{ij}(\mu), d_i]$ for $j=1, \ldots, m$.

4. Compute $b_{ij}^*(\lambda) = \mathrm{POINV}[b_{ij}(\mu), d_i]$ for $j=1, \ldots, \ell$.

5. If $i = m$ then stop.

6. $i:=i+1$ and go to 2.

The algorithm will be written

$$[A^*(\lambda), B^*(\lambda)] = \text{INV}[A(\mu), B(\mu)],$$

where λ is given by (4.49).

Remark Observe that the resulting system

$$A^*(\lambda)y = B^*(\lambda)u \tag{4.54}$$

may have i.d.z. at the origin. These i.d.z. are introduced if and only if the matrix $[A(\mu)\ \ B(\mu)]$ is not row proper. Consequently, they are avoided if $[A(\mu)\ \ B(\mu)]$ is made row proper, e.g. with the algorithm COLPRO, before INV is applied.

4.5 Structure Matrices

The structure matrices play an important role in the design theory of part 2 of Pernebo (1978). The theoretical background is found in sections 2.3 and 3.5 of that reference.

Consider a system with two input vectors and two output vectors as is shown in figure 4.1.

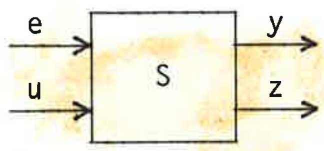

Figure 4.1 The system.

The input u is the control input and e is the disturbance. The output y is the output to be controlled and z is the measured output. Suppose that the system is described by

$$D(\mu)\xi = I \begin{pmatrix} u \\ e \end{pmatrix} \tag{4.55 a}$$

$$\begin{pmatrix} y \\ z \end{pmatrix} = E(\mu)\xi, \tag{4.55 b}$$

where $D(\mu)$ and $E(\mu)$ are relatively right prime polynomial matrices. Introduce the operator

$$\lambda = \frac{1}{\mu - a}, \tag{4.56}$$

where a belongs to the stable region of the complex plane and compute the corresponding polynomial matrix system in λ as is shown in section 4.4.
This gives

$$D^*(\lambda)\xi = I\begin{pmatrix} u \\ e \end{pmatrix} \tag{4.57 a}$$

$$\begin{pmatrix} y \\ z \end{pmatrix} = E^*(\lambda)\xi, \tag{4.57 b}$$

where $D^*(\lambda)$ and $E^*(\lambda)$ are polynomial matrices. The matrix $D^*(\lambda)$ can be made block triangular by multiplication from the right by a unimodular matrix, giving

$$\begin{pmatrix} D_1(\lambda) & D_3(\lambda) \\ 0 & D_2(\lambda) \end{pmatrix}\begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} = \begin{pmatrix} I \\ 0 \end{pmatrix}u + \begin{pmatrix} 0 \\ I \end{pmatrix}e \tag{4.58 a}$$

$$\begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} E_1(\lambda) & E_2(\lambda) \\ E_3(\lambda) & E_4(\lambda) \end{pmatrix}\begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix}. \tag{4.58 b}$$

Let Λ be the image of the unstable region of the complex plane under the mapping (4.56). Make a factorization, of the type shown in section 3.3, of $E_1(\lambda)$ as

$$E_1(\lambda) = \hat{E}(\lambda)\tilde{E}(\lambda). \tag{4.59}$$

Here all the zeros of the invariant factors of $\hat{E}(\lambda)$ belong to Λ and none of the zeros of the invariant factors of $\tilde{E}(\lambda)$ belong to Λ. The matrix $\hat{E}(\lambda)$ is defined as the left Λ-structure matrix of the system. It can be computed with the following algorithm.

1. $\left[\begin{pmatrix} \bar{D}(\mu) \\ \bar{E}(\mu) \end{pmatrix}, \cdot, \cdot\right] = \text{COLPRO}\left[\begin{pmatrix} D(\mu) \\ E(\mu) \end{pmatrix}, \cdot, \cdot\right]$

2. $[T(\eta), U(\eta)] = \text{TRAN}[\bar{D}^T(\mu), \bar{E}^T(\mu), 1, -a]$

3. $[D^{*T}(\lambda), E^{*T}(\lambda)] = \text{INV}[T(\eta), U(\eta)]$

4. Partition $D^*(\lambda)$ as

$$D^*(\lambda) = \begin{pmatrix} D_1^*(\lambda) \\ D_2^*(\lambda) \end{pmatrix},$$

where $D_1^*(\lambda)$ has ℓ rows. (ℓ is the number of components of u).

5. $[\cdot, E_1^*(\lambda), \cdot] = \text{BASIC}[D_2^*(\lambda), E^*(\lambda), \cdot]$

6. Partition $E_1^*(\lambda)$ as

$$E_1^*(\lambda) = \begin{pmatrix} E_2(\lambda) & E_1(\lambda) \\ E_4(\lambda) & E_3(\lambda) \end{pmatrix},$$

where $E_1(\lambda)$ is $m \times \ell$. (m is the number of components in y).

7. $[\hat{E}(\lambda), \cdot] = \text{RASPFA}[E_1(\lambda)]$

The algorithm will be written

$$\hat{E}(\lambda) = \text{STRMAT}[D(\mu), E(\mu), a, \ell, m],$$

where $\hat{E}(\lambda)$ is the left Λ-structure matrix of the system (4.55). The operator λ is given by (4.56), where a is a real number in the stable region of the complex plane. Furthermore, ℓ and m are the numbers of components in u and y, respectively.

4.6 Inverses

Let $G(\mu)$ be a stable and proper transfer function. Suppose that a stable and proper transfer function $K(\mu)$ should be determined so that the cascade system in figure 4.2 has a given transfer function $H(\mu)$.

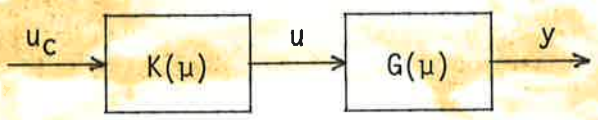

Figure 4.2 The system with compensator.

A stable and proper $K(\mu)$ should thus be found, such that

$$G(\mu)K(\mu) = H(\mu). \tag{4.60}$$

It is shown in chapter 5 of part 2 of Pernebo (1978) that this is possible if and only if the left Λ-structure matrix of $G(\mu)$ is a left divisor of the left Λ-structure matrix of $H(\mu)$.

An algorithm to compute $K(\mu)$ with the method of section 5.4 in Pernebo (1978) will be given. Let $G(\mu)$ and $H(\mu)$ be represented as

$$G(\mu) = T_1^{-1}(\mu)U_1(\mu) \tag{4.61}$$

$$H(\mu) = T_2^{-1}(\mu)U_2(\mu). \tag{4.62}$$

Introduce the operator

$$\lambda = \frac{1}{\mu - a}, \tag{4.63}$$

where a belongs to the stable region of the complex plane, into (4.61) - (4.62). This gives

$$G^*(\lambda) = T_3^{-1}(\lambda)U_3(\lambda) \tag{4.64}$$

$$H^*(\lambda) = T_4^{-1}(\lambda)U_4(\lambda). \tag{4.65}$$

Let Λ be the image of the unstable region under the mapping (4.63). The origin belongs to Λ by definition. Then a $K^*(\lambda)$ with no poles in Λ should be determined such that

$$T_3^{-1}(\lambda)U_3(\lambda)K^*(\lambda) = T_4^{-1}(\lambda)U_4(\lambda). \tag{4.66}$$

A stable and proper $K(\mu)$, satisfying (4.60) is then obtained as

$$K(\mu) = K^*\left(\frac{1}{\mu - a}\right). \tag{4.67}$$

There are unique polynomial matrices $T_5(\lambda)$ and $T_6(\lambda)$, such that

$$T_5^{-1}(\lambda)T_6(\lambda) = T_3(\lambda)T_4^{-1}(\lambda). \tag{4.68}$$

Multiply the equation (4.66) from the left by the nonsingular polynomial matrix $T_5(\lambda)T_3(\lambda)$. This gives

$$T_5(\lambda)U_3(\lambda)K^*(\lambda) = T_6(\lambda)U_4(\lambda), \tag{4.69}$$

which has the same solutions as (4.66). The algorithm of section 3.2 can now be used to find all solutions of (4.69).

The following algorithm computes a stable and proper $K(\mu)$ satisfying (4.60).

1. $[\bar{T}_1(\eta),\ \bar{U}_1(\eta)] = \text{TRAN}\ [T_1(\mu),\ U_1(\mu),\ 1,\ -a]$

   $[\bar{T}_2(\eta),\ \bar{U}_2(\eta)] = \text{TRAN}\ [T_2(\mu),\ U_2(\mu),\ 1,\ -a]$

2. $[T_3(\lambda),\ U_3(\lambda)] = \text{INV}[\bar{T}_1(\eta),\ \bar{U}_1(\eta)]$

   $[T_4(\lambda),\ U_4(\lambda)] = \text{INV}[\bar{T}_2(\eta),\ \bar{U}_2(\eta)]$

3. $\left[\cdot, \begin{pmatrix} N_1(\lambda) & N_2(\lambda) \\ A(\lambda) & B(\lambda) \end{pmatrix}^T, \cdot\right] = \text{BASIC}\left[\begin{pmatrix} T_3(\lambda) \\ -T_4(\lambda) \end{pmatrix}^T, \begin{pmatrix} U_3(\lambda) & 0 \\ 0 & U_4(\lambda) \end{pmatrix}^T, \cdot\right]$

4. $[\bar{D}(\lambda),\ \bar{N}(\lambda),\ \cdot] = \text{POMEQ}[A(\lambda),\ B(\lambda)]$

5. $z = \text{INVZER}[D(\lambda)]$

6. If some z_i belongs to Λ then no stable and proper solution exists.

7. $[\tilde{D}(\eta),\ \tilde{N}(\eta)] = \text{INV}[\bar{D}^T(\lambda),\ \bar{N}^T(\lambda)]$

8. $[D^T(\mu),\ N^T(\mu)] = \text{TRAN}\ [\tilde{D}(\eta),\ \tilde{N}(\eta),\ 1,\ a]$

The algorithm will be written

$$[D(\mu),\ N(\mu)] = \text{INVERS}[T_1(\mu),\ U_1(\mu),\ T_2(\mu),\ U_2(\mu),\ a]$$

and the solution $K(\mu)$ to (4.60), with $G(\mu)$ and $H(\mu)$ as in (4.61) and (4.62), is given by

$$K(\mu) = N(\mu)D^{-1}(\mu). \tag{4.70}$$

The point a should be chosen on the real axis in the stable region of the complex plane. The solution $K(\mu)$ will in general have some poles at a.

5. References

Anderson BDO, Scott RW (1977): Parametric Solution of the Stable Exact Model Matching Problem. IEEE Trans. Aut. Contr. AC-22, 137 - 138.

Bengtsson G (1977): Output Regulation and Internal Model - A Frequency Domain Approach. Automatica 13, 333 - 345.

Cheng L, Pearson JB (1978): Frequency Domain Synthesis of Multivariable Linear Regulators. IEEE Trans. Aut. Contr. AC-23, 3 - 15.

Pernebo L (1977): Notes on Strict System Equivalence. Int. J. Contr. 25, 21 - 38.

Pernebo L (1978): Algebraic Control Theory for Linear Multivariable Systems. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, CODEN: LUTFD2/(TFRT-1016)/1-307/(1978).

Pernebo L (1980): A Command Guide for a Computer Program for Polynomial Matrix Systems. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, CODEN: LUTFD2/(TFRT-7206)/1-022/(1980).

Rosenbrock HH (1970): State - Space and Multivariable Theory. Nelson, London.

Wang SH, Davison EJ (1973): A Minimization Algorithm for the Design of Linear Multivariable Systems. IEEE Trans. Aut. Contr. AC-18, 220 - 225.

Wolovich WA (1974): Linear Multivariable Systems. Springer-Verlag, New York.

APPENDIX

List of Algorithms

Name	Purpose	Def. on page
BASIC	This name is used for either COLPRO or ECHELON when it does not matter which of the two is used.	2.6
CHOICE	Interactive choice of state variables. To follow CSTATE.	4.14
CSTATE	Transformation of a continuous time system from polynomial matrix form to state space form.	4.12
COLPRO	Makes a polynomial matrix column proper.	2.4
CON	Transformation of a polynomial matrix system to controllable form.	4.17
DIV	Division algorithm for polynomial matrices.	4.10
ECHELON	Brings a polynomial matrix to Echelon form.	2.6
FACTRI	Spectral factorization of a triangular polynomial matrix.	3.9
GCLD	Computes the greates common left divisor of a set of polynomial matrices.	3.2
INV	Makes the change of operators $\lambda = \mu^{-1}$ in a polynomial matrix system in controllable or observable form.	4.21
INVERS	Computes a cascade compensator.	4.25
INVZER	Computes the invariant zeros of a polynomial matrix.	3.10

LD	Selects a basis from a set of column vectors.	2.3
PERM	Permutation of columns.	3.9
POMEQ	Solves the polynomial matrix equation AX = B for X.	3.5
POINV	Makes the change of variables $\lambda = \mu^{-1}$ in a scalar polynomial.	4.20
POTRAN	Makes the change of variables $\lambda = \mu + a$ in a scalar polynomial.	4.20
RASPFA	Makes a rank and spectral factorization of a polynomial matrix.	3.10
SOLV	Solves the matrix equation AX = B for X when A and B are real matrices.	2.3
STRMAT	Computes the left structure matrix for a dynamical system.	4.23
TRAN	Makes the change of operators $\lambda = \mu + a$ in a polynomial matrix system.	4.20
TSYT	Makes a change of coordinates in a state space system.	4.9