LUND UNIVERSITY

A Command Guide for a Computer Program 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). *A Command Guide for a Computer Program for Polynomial Matrix Systems*. (Technical Reports TFRT-7206). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:
1

General rights
Unless other specific re-use rights are stated the following general rights apply:
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.
• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

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

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

LUND UNIVERSITY

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

CODEN: LUTFD2/(TFRT-7206)/1-022/(1980)

A COMMAND GUIDE FOR A COMPUTER PROGRAM FOR POLYNOMIAL MATRIX SYSTEMS

LARS PERNEBO

DEPARTMENT OF AUTOMATIC CONTROL
LUND INSTITUTE OF TECHNOLOGY
OCTOBER 1980

A COMMAND GUIDE FOR A COMPUTER PROGRAM 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-7206)/1-022/(1980)

Author(s)
Lars Pernebo

Sponsoring organization
Swedish Board for Technical Development (STU), contract no 77-3548

Title and subtitle
A Command Guide for a Computer Program for Polynomial Matrix Systems

A4 A5

Abstract
This report contains a number of suggested commands for an interactive computer program. The program is assumed to contain algorithms for treatment of polynomial matrices and dynamical systems described by polynomial matrices. A suggested interaction between user and program is described in detail.

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
22

Price

Security classification

Distribution by (name and address)

DOKUMENTDATABLAD enl SIS 61 41 21

Contents

Introduction

Polynomial matrices were introduced into system theory by Rosenbrock (1970). Linear multivariable systems, described by polynomial matrices, were analysed in detail. During recent years, design methods, based on polynomial matrix descriptions, have appeared, e.g. Wang, Davison (1973), Wolovich (1974), Bengtsson (1977), Cheng, Pearson (1978), and Pernebo (1978).

Computations involving polynomial matrices tend to be very tedious. Therefore, there is a great need for algorithms suitable for computer calculations. Such algorithms for some polynomial matrix problems are presented in Pernebo (1980). It is also important to make the algorithms easy to use for a person who is not familiar with all the details of the theory. In other words, it is important to have an efficient way of interaction between the algorithms and the user.

At the Department of Automatic Control in Lund there is an interactive computer program for analysis and transformation of linear dynamical systems. The program is called Modpac and the interaction between the program and the user is described in Schönthal (1978). The algorithms in Pernebo (1980) will be included in Modpac and in this report it is described how the interaction between these new algorithms and the user can be made.

The program Modpac is command operated. The matrices and system descriptions are stored in files on a mass storage. The commands then operate on these files. The command structure is essentially the following

Command [Result] ← Input [Par]
Command - Command identifier
Result - Name of file for resulting data
Input - Name of file for input data
Par - Parameters.

Arguments within square brackets are optional.

In this report new commands, that are needed for the algorithms in Pernebo (1980), are described. It is also shown how these commands are intended to be used. In chapter 2 some, already existing, basic commands in Modpac are

discussed. In chapter 3, commands for computations on polynomial matrices are given and in chapter 4, commands for transformation and analysis of dynamical systems are described.

2. Basic Commands

The program Modpac, as presented in Schönthal (1978), contains algorithms for treatment of systems in state space form or in transfer function form, but not in polynomial matrix form. Consequently, the commands, that operate on matrices, are made for real matrices. These commands have to be modified, so that they can operate on polynomial matrices as well. The modifications are discussed in this chapter.

The command ENTER creates a matrix file on mass storage and with the command ALTER matrix elements can be changed. These two commands do not necessarily have to be generalized to polynomial matrices because there is a command POLY. With POLY it is possible to create polynomial matrix files and to change coefficients in the elements of a polynomial matrix.

With the command EXPAN it is possible to create a large matrix formed by composition of many small matrices. The command REDUC creates a new matrix, which is a part of a larger matrix. These two commands have to be generalized to polynomial matrices.

The command MATOP creates a new matrix as an algebraic expression of old matrices. This command should also be generalized to polynomial matrices. The operations needed for polynomial matrices are addition, subtraction, multiplication and transposition. It should also be possible to have both real and polynomial matrices in the same algebraic expression.

With the command LIST it is, among other things, possible to display real matrices on an output device. It should be possible to display polynomial matrices as well. Furthermore, a command PLEV, to plot a set of complex numbers in the complex plane, is needed. Finally, a command SYST, which creates system description files also for polynomial matrix systems, is needed.

3. Polynomial Matrices

In this chapter the commands for treatment of polynomial matrices are presented. It is also explained how the commands should be used. The underlying algorithms are given in chapters 2 and 3 of Pernebo (1980).

3.1 Elementary Column and Row Operations; The Command CROP

With the command CROP it is possible to perform elementary column and row operations on a given polynomial matrix. The transformation matrices and their inverses can also be computed.

Command:

CROP [M1] ← M2

M1 - Output matrix file name. Default M1 = M2
M2 - Input matrix file name

Subcommands:

LT [N1] [←N2]

Performs all subsequent row operations on N2 as well. The result is N1. Either N1 or N2 is optional. If N2 is omitted an identity matrix is assumed, while N1 = N2 by default.

RT [N1] [←N2]

Performs all subsequent column operations on N2 as well. (For default values see LT).

LTI [N1] [←N2]

Performs the inverse operations of all subsequent row operations on the columns of N2. (For default values see LT).

RTI [N1] [←N2]

Performs the inverse operations of all subsequent column operations on the rows of N2. (For default values see LT).

TURN DEV STATE [M]
DEV = 'DIS'/'LP'/'TT', where DIS = display, LP = line printer and TT = teletype.
STATE = 'ON'/'OFF'.
M - File name of output matrix of one of the commands CROP, LT, RT, LTI, RTI. Default is output matrix of CROP.

When STATE = ON the matrix M is shown on the selected device. A new value of M is shown after every subcommand that performs column or row operations.

CADD I J POL

The polynomial POL times column J is added to column I. POL is either a polynomial name (the name of a polynomial file) or a polynomial image. A polynomial image is an expression of the type e.g. $k_2S\uparrow 2 \pm k_1S \pm k_0$. The indeterminate should be denoted by S. The coefficient k_i is either a real number or an algebraic expression in the coefficients of the entries of the matrix on which the column operation is performed (i.e. the present value of the matrix M1 in the main command). Such a coefficient is denoted by MIJK, where I is the row number, J the column number and K is the degree.

RADD I J POL

The polynomial POL times row J is added to row I. (See CADD).

CMULT I R

The column I is multiplied by the nonzero real number R.

RMULT I R

The row I is multiplied by the nonzero real number R.

CPERM I1 I2 ... IN

Permutes the columns. I1 is the number of the column that should be permuted to position 1, and so on. N is the number of columns of the matrix M2 of the main command. Every column number must appear precisely once.

RPERM I1 I2 ... IN

Permutes the rows (See CPERM).

COLPRO

Makes the matrix column proper via column operations.

ROWPRO

Makes the matrix row proper via row operations.

TRIPOM TYPE OP

Makes the matrix triangular.

TYPE = 'LL'/'UR', where LL = lower left and UR = upper right.

OP = 'C'/'R'. The matrix is made triangular via column operations (C) or row operations (R).

X

Executes main command and subcommands and leaves subcommand mode.

KILL

Leaves subcommand mode without execution of main command and subcommands.

Let M1 and M2 be the matrices in the main command. Then there are unimodular matrices U and V, such that

$$M1 = U \cdot M2 \cdot V. \qquad (3.1)$$

If the subcommand LT is given before any row operations are made then

$$N1 = U \cdot N2. \tag{3.2}$$

Therefore, the subcommand:

$$LT\ N1 \tag{3.3}$$

will give N1 equal to the left transformation matrix U. If some row operations have been performed before the subcommand LT is given, then these row operations will not be included in U of (3.2) (i.e. this U will not be identical to U in (3.1)).

The subcommand RT works analogously. If it is given before any column operations are made, then

$$N1 = N2 \cdot V, \tag{3.4}$$

where V is given by (3.1).

If the subcommand LTI is given before any row operations are made, then

$$N1 = N2 \cdot U^{-1}, \tag{3.5}$$

where U is given by (3.1). This can be used computing the inverse U^{-1} of the left transformation matrix. (The column operations that are the inverse operations to the elementary row operations are stated in section 2.1 of Pernebo (1980).)

If the subcommand RTI is given before any column operations are made, then

$$N1 = V^{-1} \cdot N2, \tag{3.6}$$

where V is given by (3.1).

The subcommand TURN can be used to check the effect of the elementary operations. If e.g. the command

TURN DIS ON (3.7)

is given then the present value of the matrix M1 is shown on the display (i.e. the value M1 would have if the command X were given directly after the TURN command). The value of M1 on the display is then updated after every new column or row operation.

All elementary row and column operations can be obtained via the subcommand CADD, RADD, CMULT, RMULT, CPERM and RPERM. Observe that it is possible to let the coefficients of the polynomial POL, of subcommands CADD and RADD, be algebraic expressions in the coefficients of the entries of the matrix. This can be used e.g. to reduce the degree of some entries via column or row operations.

The subcommand COLPRO or TRIPOM (OP = C) will give an M1 of the form

$$M1 = (L\ 0), \tag{3.8}$$

where L has linearly independent columns. Either COLPRO or TRIANG can therefore be used together with LT to compute a unimodular V, such that

$$(L\ 0) = M2 \cdot V. \tag{3.9}$$

It is shown in Pernebo (1978 b) that L and V in (3.9) can be used to solve many problems concerning polynomial matrices, e.g. finding greatest common divisors.

With the command TRIPOM it is possible to compute lower left or upper right triangular matrices. Observe, however, that if TRIPOM is combined with CPERM or RPERM, then also lower right and upper left matrices can be obtained.

3.2 Greatest Common Divisors; The Commands GCLD and GCRD

The commands for computation of the greatest common left divisor GCLD and the greatest common right divisor GCRD of a set of polynomial matrices will be described simultaneously.

Command:

COMMAND ALG D M1 ... MN

COMMAND = 'GCLD'/'GCRD', where GCLD = greatest common left divisor and GCRD = greatest common right divisor.

ALG = 'PRO'/'TRI'. It is possible to choose between two algorithms. One algorithm makes the resulting g.c.l.d. (g.c.r.d.) column (row) proper (PRO), while the other algorithm makes it triangular (TRI).

D - Output matrix file name. The g.c.l.d. (g.c.r.d.)
M1 ... MN - Input matrix file names. The set of N polynomial matrices for which the g.c.l.d. (g.c.r.d.) is computed.

Subcommands:

FACT F1 ... FN

F1 ... FN - Output matrix file names.
Computes the factors FI, I = 1, ..., N, such that

MI = D·FI for GCLD and
MI = FI·D for GCRD.

EQ X1 ... XN

X1 ... XN - Output matrix file names.
Computes a solution X1 ... XN to the equation

D = M1·X1 + ... + MN·XN for GCLD and
D = X1·M1 + ... + XN·MN for GCRD.

X

Executes main command and subcommands and leaves subcommand mode.

KILL

Leaves subcommand mode without execution of main command and subcommands.

Observe that it can easily be checked if the matrices are relatively left (right) prime. If ALG = PRO then the matrices are relatively left (right) prime, if and only if D is square and of degree zero. If ALG = TRI, then the matrices are relatively left (right) prime, if and only if D is square with all diagonal entries of degree zero.

3.3 Polynomial Matrix Equations; The Command POMEQ

With the command POMEQ it is possible to compute all solutions to two different polynomial matrix equations.

Command:

POMEQ TYPE ALG B A1 ... AN

TYPE = 'L'/'R'. With TYPE = L the equation
B = A1·Y1 + ... + AN·YN is considered. With TYPE = R the equation
B = Y1·A1 + ... + YN·AN is considered.

ALG = 'PRO'/'TRI'. Algorithm selector.
B - Input matrix file name.

A1 ... AN - Input matrix file names.

It is checked if the equation

B = A1·Y1 + ... + AN·YN (TYPE = L) or
B = Y1·A1 + ... + YN·AN (TYPE = R)

has a solution. The appropriate one of the following three messages is displayed.
1. There exists a polynomial solution.
2. There exists a rational but no polynomial solution.
3. There exists no solution.

Subcommands:

SOLU X1 ... XN [D]

Computes a solution if one exists.

X1 ... XN - Output matrix file names.
D - Output matrix file name. Can be omitted if and only if a polynomial solution exists.

The solution is given by

$$YI = XI \cdot D^{-1} \text{ for } I = 1, \ldots, N \quad (TYPE = L)$$
$$YI = D^{-1} \cdot XI \text{ for } I = 1, \ldots, N \quad (TYPE = R).$$

The matrix D will get the value of an identity matrix if and only if a polynomial solution exists. In this case it can be omitted from the argument list of the command and the solution is given by

$$YI = XI \text{ for } I = 1, \ldots, N.$$

HOM Z1 ... ZN

Z1 ... ZN - Output matrix file names.

Computes a polynomial solution to the homogeneous equation

$$0 = A1 \cdot Z1 + \ldots + AN \cdot ZN \qquad (TYPE = L)$$
$$0 = Z1 \cdot A1 + \ldots + ZN \cdot AN \qquad (TYPE = R).$$

X

Executes main command and subcommands and leaves subcommand mode.

KILL

Leaves subcommand mode without execution of main command and subcommands.

The algorithm selector ALG is used to choose one of two algorithms for the computation. With ALG = PRO an algorithm is chosen, which makes the resulting matrix D of subcommand SOLU column proper (TYPE = L) or row proper (TYPE = R). The choice ALG = TRI makes D triangular. Also observe that if numerical problems occur with one of the algorithms then it might be worth-while to try the other.

Assume that a polynomial solution exists. Then any polynomial solution can be obtained as

$$YI = XI + ZI \cdot K,\ I = 1, \ldots, N\ (TYPE = L)$$
$$YI = XI + K \cdot ZI,\ I = 1, \ldots, N\ (TYPE = R)$$

for some polynomial matrix K. Any rational solution can be obtained for some rational matrix K. If there exists a rational solution but not polynomial solution, then any rational solution can be obtained as

$$YI = XI \cdot D^{-1} + ZI \cdot K,\ I = 1, \ldots, N\ (TYPE = L)$$
$$YI = D^{-1} \cdot XI + K \cdot ZI,\ I = 1, \ldots, N\ (TYPE = R)$$

for some rational matrix K.

3.4 Rank and Spectral Factorization; The Command RASPFA

The rank and spectral factorization of a polynomial matrix is defined in section 3.5 of Pernebo (1978). It is also discussed in section 3.3 of Pernebo (1980).

Command:

RASPFA M1 M2 ← M3

M1 - Output matrix file name of the left factor.
M2 - Output matrix file name of the right factor.
M3 - Input matrix file name.

The polynomial matrix M3 is factorized as

$$M3 = M1 \cdot M2,$$

where M1 has linearly independent columns and M2 has linearly independent rows.

The zeros of the invariant factors of M3 are displayed in a table of the form

NR	VALUE
1	Z1
2	Z2
.	.
.	.
.	.
N	ZN

Some of these zeros will be the zeros of the invariant factors of M1 and the rest will be the zeros of the invariant factors of M2. The subcommands below are used to decide how the zeros shall be distributed between M1 and M2. They will all belong to M2 by default.

Subcommands:

PLOT

Plots the zeros of the invariant factors of M3 in the complex plane.

LEFT I1 ... IK

I1 ... IK - The numbers of the zeros that shall belong to the left factor M1.

RIGHT I1 ... IJ

I1 ... IJ - The numbers of the zeros that shall belong to the right factor M2.

X

Executes main command and subcommands and leaves subcommand mode.

KILL

Leaves subcommand mode without execution of main command and subcommands.

3.5 The Zeros of the Invariant Factors; The Command INVZER

If only the zeros of the invariant factors are desired, then the command INVZER can be used instead of RASPFA.

Command:

INVZER Z ← M

Computes the zeros of the invariant factors of the polynomial matrix M.

Z - Output file name for the complex vector of zeros of the invariant factors of M.

M - Input matrix file name.

If M is square and nonsingular then the zeros of the invariant factors are identical to the zeros of det M. The command can thus, in this case, be used to compute the zeros of det M.

4. Linear Dynamical Systems

Commands for transformation and analysis of linear, multivariable systems are presented in this section. The system is assumed to be described as

$$T(\mu)z = U(\mu)u \qquad (4.1\ a)$$
$$y = V(\mu)z + W(\mu)u, \qquad (4.1\ b)$$

where $T(\mu)$, $U(\mu)$, $V(\mu)$ and $W(\mu)$ are rxr, rxl, mxr and mxl polynomial matrices with $T(\mu)$ nonsingular. The operator μ is equal to the differential operator p for continuous time systems and the forward shift operator q for discrete time systems. The algorithms, corresponding to the commands of this section, are found in chapter 4 of Pernebo (1980).

4.1 Transformation to State Space Form; The Command STATE

The command STATE is based on the algorithm in section 4.1 of Pernebo (1980). It is assumed that the system is a continuous time system, i.e. μ=p in (4.1). The command can be applied also to discrete time systems. Derivatives must then be interpreted as forward time shifts.

Command:

STATE [S1]←S2
Transforms the system to state space form.
S1 - Output system description file name. Default S1 = S2.

S2 - Input system description file name.

A table of linear combinations L1, L2, ..., LK of the internal variables, their derivatives, the inputs, and their derivatives is displayed. An example is shown below.

```
*L1 = Z1 - 1.5*U2
 L2 = DZ1 + 0.2*U1 + 0.8*DU2
 L3 = D2Z1 + 0.4*DU1 + U1 - D2U2
*L4 = Z2 + 3*U2
*L5 = 2*Z1 + Z2
```

Here ZI denotes the Ith internal variable of the system S2 and UI denotes the Ith input. Furthermore, DI denotes the Ith derivative. State variables are selected from this table with the subcommand STVAR below. Any LI, marked '*', or any linear combination of the LI, including at least one LI marked '*', can be chosen as the first state variable. Then a new table is displayed and the procedure is repeated for the second state variable and so on.

If it does not matter how the state variables are chosen, then the subcommand STVAR can be omitted. An arbitrary state space representation is then chosen.

Subcommands:

STVAR

Choice of state variables.
The programme displays 'X1 =' and a linear combination of the LI in the table above should be written after the equality sign. A new table is then displayed followed by 'X2 ='. In this way all the state variables can be chosen.

TR V H M

Computation of the relation between the state variables of the system S1 and the internal variables of the system S2.

V - Output text file name.

H - Output real matrix file name.

M - Output polynomial matrix file name.

The relation between the internal variables of the system S2 and the state variables x of the system S1 is given by

$$v = Hx + Mu. \tag{4.2}$$

The components of the vector v are the internal variables of S2 and some of their derivatives. Furthermore, x is the state variable of S1 and u is the input. The text file V contains the information about which variables are contained in v.

X

Executes main command and subcommands and leaves subcommand mode.

KILL

Leaves subcommand mode without execution of main command and subcommands.

With the subcommand STVAR it is possible to obtain state equations with state variables that are related to the internal variables of the original system S2 in a simple way. The state equations can therefore be analysed to obtain important information about the original internal variables. If, however, the only reason for transformation to state space form is to simulate the system then there is no need for the subcommand STVAR. Any state space representation can be simulated together with (4.2) to obtain a simulation of the original internal variables.

4.2 Controllable or Observable Systems; The Commands CON and OBS

If the system (4.1) is controllable, i.e. has no input decoupling zeros, then it can be transformed to a system of the form

$$Q(\mu)\xi = u \tag{4.3 a}$$
$$y = P(\mu)\xi. \tag{4.3 b}$$

Analogously, if it is observable, i.e. has no output decoupling zeros, then it can be transformed to a system of the form

$$A(\mu)y = B(\mu)u. \tag{4.4}$$

Command:

COMMAND [S1] [M1 M2] ← S2 ALG

COMMAND - 'CON'/'OBS'. The command CON transforms a controllable system to the form (4.3) and the command OBS transforms an observable system to the form (4.4).

S1 - Output system description file name. Default S1 = S2

M1, M2 - Output matrix file names. Let z_1 be the internal variable of the system S1 and z_2 the internal variable of the system S2. Then the relation between z_1 and z_2 is given by

$$z_1 = M1 \cdot z_2 + M2 \cdot u,$$

where u is the input.

S2 - Input system description file name.

ALG = 'PRO'/'TRI'. Algorithm selector. With ALG = PRO the algorithm "Colpro" in Pernebo (1980) is chosen and with ALG = TRI the algorithm "Tripom" is chosen.

The command CON can also be applied to a non-controllable system S2. The resulting system S1 will then only represent the controllable part of S2. Similarly, the command OBS can be applied to a non-observable system S2 to obtain the observable part S1.

4.3 Change of Operators; The Command CHOP

Any change of operators of the type

$$\lambda = \frac{a\mu+b}{c\mu+d}, \quad ad \neq bc \tag{4.5}$$

can be achieved via repeated use of the following types of changes

$$\lambda = a\mu + b \qquad a \neq 0 \tag{4.6}$$

and

$$\lambda = \mu^{-1}. \tag{4.7}$$

Command:

CHOP [S1] ← S2

Change of operators.

S1 - Output system description file name. Default S1 = S2.

S2 - Input system description file name.

Subcommands:

TRANS A B

Makes a translatational change of operators.

A, B - Real numbers, such that the new operator λ is given by $\lambda = A\mu+B$, $A \neq 0$.

INV

This subcommand is only valid if the system is in the form (4.3) or (4.4). The operator μ is replaced by λ^{-1} and every row (if (4.3)) or column (if (4.4)) is multiplied by its least common denominator. This gives a system in polynomial matrix form of the type (4.3) or (4.4) by with the operator λ instead of μ.

X

Executes main command and subcommands and leaves subcommand mode.

KILL

Leaves subcommand mode without execution of main command and subcommands.

The change of operators (4.5) include the change of operators that is required in the design method of Pernebo (1978 a). It is obtained with a = 0 and b = c = 1 in (4.5). The transformation of a discrete time system from a forward shift operator to a backward shift operator representation is obtained with a = d = 0 and b = c = 1.

Observe that the subcommand INV may give a system with output decoupling zeros (if applied to (4.3)) or input decoupling zeros (if applied to (4.4)) at the origin. These decoupling zeros are introduced by the transformation. If the system is of the type (4.3) then they are avoided if and only if the matrix $(Q^T(\mu)\ P^T(\mu))^T$ is made column proper before INV is applied. Analogously, if the system is of the type (4.4) then the input decoupling zeros at the origin are

avoided if and only if the matrix (A(μ) B(μ)) is made row proper before INV is applied.

4.4 Poles, Zeros, and Decoupling Zeros; The Commands POLES, ZEROS, IDZER, ODZER and IODZER

The commands in this section are based on the algorithms in section 4.3 of Pernebo (1980). All the commands have the same structure and will be treated simultaneously.

Command:

COMMAND Z S

COMMAND = 'POLES'/'ZEROS'/'IDZER'/'ODZER'/'IODZER' for computation of the poles, zeros, input decoupling zeros, output decoupling zeros and input-output decoupling zeros, respectively.

Z - Output file name. Complex vector containing the poles, zeros or decoupling zeros of the system.

S - Input system description file name.

5. References

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

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

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): Numerical Algorithms for Polynomial Matrix Systems. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, CODEN: LUTFD2/(TFRT-7205)/1-048/(1980).

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

Schönthal T (1978): Modpac V3A - Command Guide. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, CODEN: LUTFD/(TFRT-7139)/1-053/(1978).

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

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