NUMERICAL METHODS FOR POLYNOMIAL MATRIX RANK EVALUATION *

Didier Henrion ** **Michael Šebek** ***,****

** *LAAS-CNRS*
7 Avenue du Colonel Roche
31 077 Toulouse, Cedex 4, France
E-mail: henrion@laas.fr

*** *Trnka Laboratory of Automatic Control*
Faculty of Electrical Engineering
Czech University of Technology
Prague, Czech Republic

**** *Institute of Information Theory and Automation*
Academy of Sciences of the Czech Republic
182 08 Prague 8, Czech Republic
E-mail: msebek@utia.cas.cz

Abstract: Two algorithms are proposed for evaluating the rank of an arbitrary polynomial matrix. They rely upon constant matrix rank evaluations and therefore are more reliable than already existing elementary polynomial operations techniques. Some applications are mentioned, such as polynomial null-space extraction or conditions of existence of solutions to matrix polynomial equations arising in control problems.

Résumé: Deux algorithmes sont proposés pour évaluer le rang d'une matrice polynomiale quelconque. Ils sont basés sur des évaluations de rangs de matrices constantes et sont donc plus fiables que les techniques déjà existantes d'opérations polynomiales élémentaires. Quelques applications sont mentionnées, comme l'extraction du noyau polynomial ou des conditions d'existence de solutions aux équations polynomiales matricielles provenant de problèmes de commande.

Keywords: Numerical algorithms, Matrix polynomial equations.

1. INTRODUCTION

Rank evaluation over the ring of polynomials is a fundamental component of a large number of algorithms dealing with polynomial matrices. It is an especially necessary step when developing numerical routines in the scope of the polynomial approach to linear systems analysis and design – see for example (Kučera, 1979; Kučera, 1991; Kučera, 1993; Özgüler, 1994) and the references therein. Rank evaluation may also prove useful when testing relative primeness of polynomial matrices, see Lemma 6.3-6 in (Kailath, 1980). In spite of this, already existing numerical methods for polynomial matrix rank evaluation cannot be

* This work was supported by the Barrande Project No. 97/005-97/026, by the Grant Agency of the Czech Republic under contract No. 102/97/0861, by the Ministry of Education of the Czech Republic under contract No. VS97/034 and by the French Ministry of Education and Research under contract No. 10-INSA-96.

considered as reliable. Indeed, they are based on elementary polynomial operations (Callier and Desoer, 1982; Chen, 1984; Kailath, 1980) that are known to suffer from bad numerical properties.

Accordingly, the contribution of this note is to propose two simple and new algorithms for polynomial matrix rank evaluation. They only hinge on real matrix rank evaluation, hence relying upon powerful and well tried numerical tools such as the singular value decomposition (Golub and Van Loan, 1989). The first algorithm is based on interpolation whereas the second algorithm uses Sylvester matrices. This paper is therefore a continuation of our research efforts initiated in (Henrion and Šebek, 1998*a*; Henrion and Šebek, 1998*b*), where various algorithms for dealing with polynomial matrices were devised through a unified and systematic use of both techniques.

Throughout the presentation of these algorithms, closely related issues such as polynomial null-space extraction and existence of solutions to matrix polynomial equations arising in control problems are furthermore touched on.

The algorithms presented here have been developed in the scope of the Polynomial Toolbox for MATLAB [1] (Kwakernaak and Šebek, 1997). They are available upon request.

The following notations are used throughout the paper. $\mathbb{C}$ stands for the field of complex numbers and $\mathbb{C}[\mathbf{s}]$ for the ring of polynomials of $\mathbf{s}$ with coefficients in $\mathbb{C}$. When referring to a polynomial matrix $A(\mathbf{s})$, we shall drop the dependence on indeterminate $\mathbf{s}$ whenever convenient. The degree, i-th row degree and j-th column degrees of a polynomial matrix A are respectively denoted by $\delta(A)$, $\delta_{\mathrm{r}i}(A)$ and $\delta_{\mathrm{c}j}(A)$. The tranpose of matrix A is written A'. Finally, $\mathbf{0}_{m\times n}$ is the $m \times n$ matrix with zero entries and $\mathbf{I}_n$ is the $n \times n$ identity matrix.

2. DEFINITIONS OF THE RANK

There are several definitions of the rank of a polynomial matrix. Hopefully all of them are equivalent. Hereafter is a list which is far from being exhaustive but that covers major statements found in control theory textbooks (Callier and Desoer, 1982; Chen, 1984; Kailath, 1980; Kučera, 1979; Kučera, 1991).

Definition 1. The *rank* (or *normal rank* or *determinantal rank*) of a polynomial matrix $A(\mathbf{s})$ is

(1) the number of linearly independent columns (resp. rows) of A in the polynomial sense
(2) the number of non-zero columns (resp. rows) of the column (resp. row) Hermite form of A
(3) the number of invariant polynomials of the Smith form of A
(4) the largest order of minors of A that are not identically zero
(5) the dimension of the module generated by the columns (resp. rows) of A
(6) the difference between the number of columns (resp. rows) of A and the dimension of the polynomial right (resp. left) null-space of A
(7) the number

$$\max_{\lambda\in\mathbb{C}} \mathrm{rank} A(\lambda)$$

3. INTERPOLATION ALGORITHM

The first algorithm for polynomial matrix rank evaluation is based on substitution. The indeterminate $\mathbf{s}$ is assigned a set of complex values $\mathbf{s}_0, \mathbf{s}_1, \mathbf{s}_2, \ldots$ and ranks of constant matrices $A(\mathbf{s}_i)$ are successively evaluated. The principle lying behind substitution is interpolation, an elegant numerical technique often encountered in control theory – see (Antsaklis and Gao, 1993) for a comprehensive treatment. Generally speaking, interpolation is based on the following intuitive reasoning: a property or value of a polynomial matrix $A(\mathbf{s})$ is inferred from the same property or value assessed on a set of constant interpolated matrices $A(\mathbf{s}_0), A(\mathbf{s}_1), A(\mathbf{s}_2), \ldots$ The sequel aims at demonstrating that this approach can also straightforwardly be employed for rank evaluation.

Recall that one standard definition of the rank of a polynomial matrix is

$$r = \mathrm{rank} A(\mathbf{s}) = \max_{\lambda\in\mathbb{C}} \mathrm{rank} A(\lambda) \qquad (1)$$

The following concept of a zero plays a key role once combined with relation (1).

Definition 2. The invariant zeros of the determinant of the minors of polynomial matrix $A(\mathbf{s})$ are the complex values $\mathbf{z}_i$ at which A drops below its normal rank, i.e.

$$\mathrm{rank} A(\mathbf{z}_i) < r. \qquad (2)$$

They are called the *zeros* of A.

Fact 3. The number of zeros of a polynomial matrix $A \in \mathbb{C}^{m\times n}[\mathbf{s}]$ is always less than or equal to the degree of the determinant of the largest minor of A. In other words, there are only up to

$$k = \min\left(\sum_{i=1}^{m}\delta_{\mathrm{r}i}(A), \sum_{j=1}^{n}\delta_{\mathrm{c}j}(A)\right) \qquad (3)$$

[1] MATLAB is a trademark of The MathWorks, Inc.

complex values $\mathbf{z}_i$ such that (2) holds.

Finally, a topological property of the zeros of a polynomial – see §0.16 in (Wonham, 1985) – will be the last component required to design the interpolation algorithm.

Fact 4. If $\mathbf{z}_0 \in \mathbb{C}$ is a zero of a polynomial $p(\mathbf{s}) \in \mathbb{C}[\mathbf{s}]$ and if p vanishes identically in some neighborhood of $\mathbf{z}_0$, then p vanishes on $\mathbb{C}$.

Clearly, it is always possible to pick up a set of $k+1$ real values $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_k$ that are sufficiently far from each other. According to Definition 2 and Properties 3 and 4, at least one element $\mathbf{s}_i$ of this set *is not a zero* of A, hence

$$\operatorname{rank} A(\mathbf{s}_i) = \max_{j=0,\ldots,k} \operatorname{rank} A(\mathbf{s}_j).$$

The exposition of the first algorithm is now in order.

Algorithm 1. Given a polynomial matrix $A(\mathbf{s}) \in \mathbb{C}^{m\times n}[\mathbf{s}]$, the following algorithm evaluates the rank of A to working precision.

- *Step 1* Select a random number $\mathbf{s}_0 \in \mathbb{C}$. Let $i = 0$ and $r = 0$.
- *Step 2* Let $r = \max(r, \operatorname{rank} A(\mathbf{s}_i))$. If $r = \min(m, n)$ then return r.
- *Step 3* Let $i = i+1$. If $i > k$, where k is given in (3), then return r.
- *Step 4* Select a random number $\mathbf{s}_i \in \mathbb{C}$ different enough from previous numbers $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{i-1}$. More specifically, select $\mathbf{s}_i$ such that for a sufficiently large real ε, the inequality $||\mathbf{s}_i| - |\mathbf{s}_j|| \geq \varepsilon|\mathbf{s}_i||\mathbf{s}_j|$ holds for any $j = 0, 1, \ldots, i-1$. Then go to Step 2.

It may be pointed out that if A has full rank then usually Algorithm 1 stops *almost immediately* in Step 2 since it is very unlikely that a random number $\mathbf{s}_0$ be a zero of A. Conversely, if A is rank deficient then $k+1$ constant matrix rank evaluations will be needed.

4. SYLVESTER MATRIX ALGORITHM

A second rank evaluation technique is now spelled out. It originates from the concept of Sylvester matrix and is tightly connected to the notion of the null-space of a polynomial matrix. As a result, a necessary condition of existence of solutions to general linear matrix polynomial equations is exposed hereafter.

4.1 *Rank evaluation and polynomial null-space*

Given a polynomial matrix $A(\mathbf{s}) \in \mathbb{C}^{m\times n}[\mathbf{s}]$, suppose that all q linearly independent polynomial vectors $k_i(\mathbf{s}) \in \mathbb{C}^n[\mathbf{s}]$ such that $A(\mathbf{s})k_1(\mathbf{s}) = A(\mathbf{s})k_2(\mathbf{s}) = \ldots = A(\mathbf{s})k_q(\mathbf{s}) = \mathbf{0}_{m\times 1}$ are collected to form a polynomial matrix

$$K(s) = [k_1(\mathbf{s}) \quad k_2(\mathbf{s}) \ \ldots \ k_q(\mathbf{s})].$$

Matrix K is clearly a representation of the polynomial right null-space of A and verifies

$$A(\mathbf{s})K(\mathbf{s}) = \mathbf{0}_{m\times q}. \tag{4}$$

From Definition 1.6, it holds

$$\operatorname{rank} A = n - q. \tag{5}$$

Now write A and K in terms of increasing powers of the indeterminate $\mathbf{s}$

$$\begin{aligned} A(\mathbf{s}) &= A_0 + \mathbf{s}A_1 + \ldots + \mathbf{s}^{\delta(A)}A_{\delta(A)} \\ K(\mathbf{s}) &= K_0 + \mathbf{s}K_1 + \ldots + \mathbf{s}^{\delta(K)}K_{\delta(K)}. \end{aligned}$$

By equating the powers of $\mathbf{s}$ in matrix polynomial equation (4), one can build an equivalent linear system of equations

$$\underbrace{\begin{bmatrix} A_0 & & \mathbf{0} \\ A_1 & A_0 & \\ \vdots & A_1 & \ddots \\ A_{\delta(A)} & \vdots & A_0 \\ & A_{\delta(A)} & A_1 \\ & & \ddots \ \vdots \\ \mathbf{0} & & A_{\delta(A)} \end{bmatrix}}_{\langle A\rangle_{\delta(K)}} \underbrace{\begin{bmatrix} K_0 \\ K_1 \\ \vdots \\ K_{\delta(K)} \end{bmatrix}}_{\overline{K}} = \mathbf{0} \tag{6}$$

Matrix $\langle A\rangle_{\delta(K)}$ is called the *Sylvester matrix* of A of order $\delta(K)$. Since matrix polynomial equation (4) and linear system (6) are equivalent, extracting polynomial null-space K amounts to computing the null space of constant matrix $\langle A\rangle_{\delta(K)}$, the only issue remaining the estimation of degree $\delta(K)$. This has motivated the following lemma which provides an upper bound on $\delta(K)$ and therefore on the size of the Sylvester matrix.

Lemma 5. Given a polynomial matrix $A(\mathbf{s}) \in \mathbb{C}^{m\times n}[\mathbf{s}]$, let

$$d_R^* = \sum_{i=1}^{n} \delta_{ci}(A) - \min_{i=1,\ldots,n} \delta_{ci}(A). \tag{7}$$

Then the polynomial right null-space of A is generated by the columns of a polynomial matrix of degree less than or equal to d_R^*.

Proof. Let rank$A = n-q$ and U be a unimodular matrix that transforms A into column Hermite form (Henrion and Šebek, 1998*b*), i.e.

$$\begin{aligned} A(\mathbf{s})U(\mathbf{s}) &= A(\mathbf{s})[V(\mathbf{s}) \quad K(\mathbf{s})] \\ &= [A(\mathbf{s})V(\mathbf{s}) \quad \mathbf{0}_{m\times q}]. \end{aligned}$$

The $n \times q$ polynomial matrix K is a polynomial right null-space of A and obviously

$$\delta(K) \leq \delta(U). \tag{8}$$

Now suppose that P is a constant permutation matrix such that

$$P \begin{bmatrix} A(\mathbf{s}) \\ \mathbf{I}_n \end{bmatrix} U(\mathbf{s}) = \begin{bmatrix} M(\mathbf{s}) \\ \star \end{bmatrix} U(\mathbf{s}) = \begin{bmatrix} N(\mathbf{s}) \\ \star \end{bmatrix}$$

where a star denotes an irrelevant block, M and N are non-singular and N is lower triangular. Since U is unimodular,

$$\delta(\det N) = \delta(\det M) = \sum_{i=1}^{n} \delta_{\mathrm{c}i}(N) \tag{9}$$

Let M_{adj} stand for the adjoint polynomial matrix of M, whence

$$U(\mathbf{s})\det M(\mathbf{s}) = M_{\mathrm{adj}}(\mathbf{s})N(\mathbf{s}). \tag{10}$$

From (10) and (9), it follows that

$$\begin{aligned} \delta(U) &\leq \delta(M_{\mathrm{adj}}) + \delta(N) - \delta(\det M) \\ &\leq \delta(M_{\mathrm{adj}}) + \max_{i=1,\ldots,n} \delta_{\mathrm{c}i}(N) \\ &\quad - \textstyle\sum_{i=1}^{n} \delta_{\mathrm{c}i}(N). \end{aligned} \tag{11}$$

Since entries of M_{adj} are minors of A of order $n-1$, we have

$$\delta(M_{\mathrm{adj}}) \leq d_R^* \tag{12}$$

where d_R^* is given in (7). The claim then follows from (12), (11) and (8). □

In (Beelen *et al.*, 1988) it is stated that the bound d_R^* is sharp. As a direct consequence of the above lemma, it is sufficient to extract the right null-space of constant matrix $\langle A\rangle_{d_R^*}$ in order to derive a basis for the polynomial right null-space of A.

A similar result for the polynomial left null-space of A is captured by the following corollary to Lemma 5.

Corollary 6. Given a polynomial matrix $A(\mathbf{s}) \in \mathbb{C}^{m\times n}[\mathbf{s}]$, let

$$d_L^* = \sum_{i=1}^{m} \delta_{\mathrm{r}i}(A) - \min_{i=1,\ldots,m} \delta_{\mathrm{r}i}(A). \tag{13}$$

Then the polynomial left null-space of A is generated by the rows of a polynomial matrix of degree less than or equal to d_L^*.

For notational ease, assume henceforth that $\langle A\rangle_d = \mathbf{0}$ if $d < 0$. The next theorem summarizes the implications of Lemma 5 and Corollary 6 for rank evaluation and simultaneously provides a second algorithm.

Algorithm 2. (Theorem 2). Given a polynomial matrix $A(\mathbf{s}) \in \mathbb{C}^{m\times n}[\mathbf{s}]$,

$$\begin{aligned} \mathrm{rank}A = \min\{&\mathrm{rank}\langle A\rangle_{d^*} - \mathrm{rank}\langle A\rangle_{d^*-1}, \\ &\mathrm{rank}\langle A'\rangle_{d^*} - \mathrm{rank}\langle A'\rangle_{d^*-1}\} \end{aligned} \tag{14}$$

where

$$d^* = \min(d_R^*, d_L^*) \tag{15}$$

and d_R^*, d_L^* are respectively given in (7) and (13).

Proof. First consider the subspace generated by the columns of Sylvester matrix $\langle A\rangle_d$ for a given order d. Let q_i be the number of linearly dependent columns in the $(i+1)$th block column of $\langle A\rangle_d$. Because of the structure of the Sylvester matrix, q_i increases monotonically with i – see §G.5 in (Chen, 1984). Furthermore, Lemma 5 provides an upper bound $q_{d_R^*}$ such that

$$\begin{aligned} &0 \leq q_0 \leq q_1 \leq \ldots \leq q_{d_R^*} \\ &q_{d_R^*} = q_{d_R^*+1} = q \end{aligned} \tag{16}$$

where q is the dimension of the polynomial right null-space of A. Hence for any $d \geq 0$,

$$\begin{aligned} \mathrm{rank}\langle A\rangle_d &= (n-q_0) + \cdots + (n-q_d) \\ &= (d+1)n - \sum_{i=0}^{d} q_i. \end{aligned} \tag{17}$$

From relations (5), (16) and (17), one can check that

$$\begin{aligned} \mathrm{rank}A &= \mathrm{rank}\langle A\rangle_{d_R^*} - \mathrm{rank}\langle A\rangle_{d_R^*-1} \\ &= n-q. \end{aligned} \tag{18}$$

Similarly, consider now the subspace generated by the rows of Sylvester matrix $\langle A\rangle_d$. Using Corollary 6 and the same reasoning as above, it holds

$$\mathrm{rank}A = \mathrm{rank}\langle A\rangle_{d_L^*} - \mathrm{rank}\langle A\rangle_{d_L^*-1}. \tag{19}$$

Relations (14) and (15) then readily follow from (18) and (19). □

Similarly, it is possible to decide whether a polynomial matrix has full column rank or not, and by extension, whether a polynomial matrix is singular or not.

Corollary 7. A polynomial matrix $A(\mathsf{s}) \in \mathbb{C}^{m\times n}[\mathsf{s}]$ has full column rank if and only if $\langle A\rangle_{d^*}$ has full column rank, where d^* is given in (15).

Proof. If A has full column rank then according to Theorem 2, $\mathrm{rank}\langle A\rangle_{d^*} - \mathrm{rank}\langle A\rangle_{d^*-1} = \mathrm{rank}A = n - q_{d^*} = n$. Since $q_{d^*} = 0$, $\langle A\rangle_{d^*}$ has full column rank. Conversely, if $\langle A\rangle_{d^*}$ has full column rank then there is no solution to (6) when $\delta(K) = d^*$ and as shown by Lemma 5, the right null-space of A is zero. □

Corollary 7 can be considered as a computationally tractable extension of Lemma G-2 in (Chen, 1984) where the rank condition was only established on an infinite set of Sylvester matrices of increasing orders $\langle A\rangle_0$, $\langle A\rangle_1$, $\langle A\rangle_2, \ldots$.

4.2 *Linear matrix polynomial equations*

The equation

$$A(\mathsf{s})X(\mathsf{s}) = B(\mathsf{s}) \tag{20}$$

where A, B are given and X is unknown, can be considered as the most general type of linear matrix polynomial equation. It encompasses a wide class of fundamental equations arising in control problems, such as the Diophantine equation (Kučera, 1993), the coupled polynomial equations (Ježek and Hunt, 1993), the skew prime polynomial equation (Wolovich, 1978) and also more general bilateral linear matrix equations arising in disturbance decoupling or model matching problems (Özgüler, 1994). To see that, just note that every linear matrix polynomial equation can be written as (20) by proper use of Kronecker products and stacking operators.

The results developed in §4.1 will now be applied to derive a numerically tractable necessary condition of existence of a solution to (20).

Lemma 8. Given two polynomial matrices $A(\mathsf{s}) \in \mathbb{C}^{m\times n}[\mathsf{s}]$ and $B(s) \in \mathbb{C}^{m\times p}[\mathsf{s}]$, let

$$\begin{aligned} d_j^\dagger = & \sum_{i=1}^{n} \delta_{ci}(A) + \delta_{cj}(B) \\ & - \min\left(\min_{i=1,\ldots,n} \delta_{ci}(A), \quad \delta_{cj}(B)\right) \end{aligned} \tag{21}$$

and

$$d^\dagger = \max_{j=1,\ldots,p} d_j^\dagger. \tag{22}$$

Then if the Sylvester matrix

$$< [A(\mathsf{s}) \;\; B(\mathsf{s})] >_{d^\dagger} \tag{23}$$

has full rank there is no polynomial solution $X(\mathsf{s})$ to equation (20).

Proof. First write (20) columnwise :

$$A(\mathsf{s})[x_1(\mathsf{s}) \; \ldots \; x_p(\mathsf{s})] = [b_1(\mathsf{s}) \; \ldots \; b_p(\mathsf{s})].$$

If (20) holds then for any $j = 1, 2, \ldots, p$,

$$[A(\mathsf{s}) \quad b_j(\mathsf{s})] \begin{bmatrix} x_j(\mathsf{s}) \\ -1 \end{bmatrix} = \mathbf{0}_{m\times 1}.$$

From Lemma 5, the degree of the polynomial right null-space of matrix $[A(\mathsf{s}) \; b_j(\mathsf{s})]$ is bounded from above by $d_j^\dagger$. Since this holds for any $j = 1, 2, \ldots, p$, a necessary condition for a polynomial solution X to (20) to exist is that a null-space of degree less than or equal to $d^\dagger$ exists for matrix $[A(\mathsf{s}) \; B(\mathsf{s})]$. The rank condition on constant matrix (23) then directly proceeds from the Sylvester matrix formulation. □

The next example will finally aim to reveal that Lemma 8 can shed light on classical resolution problems.

Example 9. Suppose that one is seeking a polynomial solution X to the polynomial matrix equation

$$\begin{bmatrix} 1 & 0 \\ \mathsf{s} & 1 \end{bmatrix} X(\mathsf{s}) = \begin{bmatrix} \mathsf{s} \\ 1 \end{bmatrix}. \tag{24}$$

A first glance to respective degrees of polynomial matrices appearing at both sides of the above equation could suggest that it is sufficient to look for a solution $X(\mathsf{s})$ of degree 1. However Lemma 8 states that it is necessary to expect solutions X of degree up to $d^\dagger = 2$. Sylvester matrix (23) then reads

$$\left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right]$$

and does not have full rank. Thus, a solution of degree 2 may exist. Indeed, using standard resolution tools (Kwakernaak and Šebek, 1997) it can be checked that

$$X(\mathsf{s}) = \begin{bmatrix} \mathsf{s} \\ 1 - \mathsf{s}^2 \end{bmatrix}$$

is (the unique) solution to equation (24).

5. CONCLUSION

Two simple algorithms were exposed that evaluate the rank of a polynomial matrix. The former uses substitution whereas the latter relies upon Sylvester matrices. Both methods are based on constant matrix rank computation. In view of the availability of powerful tools that deal with constant matrices, it is believed that these new algorithms constitute a significant improvement over existing elementary operation techniques.

Links with polynomial null-space extraction were also enlightened. More specifically, an upper bound on the degree of the null-space of an arbitrary polynomial matrix was established.

An application to a highly general type of polynomial matrix equations arising in control problems was finally mentioned in terms of a numerically tractable condition of existence of a solution. The resolution of such equations was out of the scope of the present paper and is studied in detail in (Henrion and Šebek, 1998*c*).

6. REFERENCES

Antsaklis, P. J. and Z. Gao (1993). Polynomial and rational matrix interpolation: theory and control applications. *International Journal of Control* **58**(2), 349–404.

Beelen, T., G. J. Van Den Hurk and C. Praagman (1988). A new method for computing a column reduced polynomial matrix. *Systems & Control Letters* **10**, 217–224.

Callier, F. M. and C. A. Desoer (1982). *Multivariable feedback systems.* Springer Verlag.

Chen, C. T. (1984). *Linear system theory and design.* Holt, Rinehart & Winston.

Golub, G. H. and C. F. Van Loan (1989). *Matrix computations.* The Johns Hopkins University Press.

Henrion, D. and M. Šebek (1998*a*). Symmetric matrix polynomial equation : interpolation results. *To appear in Automatica.*

Henrion, D. and M. Šebek (1998*b*). Efficient algorithms for polynomial matrix triangularization. *Proceedings of the IEEE Mediterranean Conference on Control and Automation*, Alghero, Sardinia, Italy.

Henrion, D. and M. Šebek (1998*c*). Numerical algorithms for polynomial matrices in control. *Submitted for publication.*

Ježek, J. and K. J. Hunt (1993). Coupled polynomial equations of LQ control synthesis and an algorithm for solution. *International Journal of Control* **58**(5), 1155–1167.

Kailath, T. (1980). *Linear systems.* Prentice Hall.

Kučera, V. (1979). *Discrete linear control: the polynomial equation approach.* John Wiley & Sons.

Kučera, V. (1991). *Analysis and design of discrete linear control systems.* Prentice Hall.

Kučera, V. (1993). Diophantine equations in control - a survey. *Automatica* **29**(6), 1361–1375.

Kwakernaak, H. and M. Šebek (1997). The Polynomial Toolbox for MATLAB. The precommercial version 1.5 is available at `http://www.math.utwente.nl/polbox`.

Özgüler, A. B. (1994). *Linear multichannel control - a system matrix approach.* Prentice Hall.

Wolovich, W. A. (1978). Skew prime polynomial matrices. *IEEE Transactions on Automatic Control* **23**(5), 880–887.

Wonham, W. M. (1985). *Linear multivariable control - a geometric approach.* Springer Verlag.