On the application of different numerical methods to obtain null-spaces of polynomial matrices. Part 1: block Toeplitz algorithms.

J.C. Zúñiga and D. Henrion

***Abstract*— Four different algorithms are designed to obtain the null-space of a polynomial matrix. In this first part we present two algorithms. These algorithms are based on classical methods of numerical linear algebra, namely the reduction into the column echelon form and the LQ factorization. Both algorithms take advantage of the block Toeplitz structure of the Sylvester matrix associated with the polynomial matrix. We present a full comparative analysis of the performance of both algorithms and also a brief discussion on their numerical stability.**

I. Introduction

By analogy with scalar rational transfer functions, matrix transfer functions of multivariable linear systems can be written in polynomial matrix fraction form as $N(s)D^{-1}(s)$ where $D(s)$ is a non-singular polynomial matrix [1], [8], [11]. Therefore, the eigenstructure of polynomial matrices $N(s)$ or $D(s)$ contains the structural information on the represented linear system and it can be used to solve several control problems.

In this paper we are interested in obtaining the null-spaces of polynomial matrices. The null-space of a polynomial matrix $A(s)$ is the part of its eigenstructure which contains all the non-zero vectors $v(s)$ which are non-trivial solutions of the polynomial equation $A(s)v(s) = 0$.

By analogy with the constant case, obtaining the null-space of a polynomial matrix $A(s)$ yields important information such as the rank of $A(s)$ and the relation between its linearly dependent columns or rows.

Applications of computational algorithms to obtain the null-space of a polynomial matrix can be found in the polynomial approach to control system design [11] but also in others related disciplines. For example, in fault diagnostics the residual generator problem can be transformed into the problem of finding a basis of the null-space of a polynomial matrix [12].

Pursuing along the lines of our previous work [15] where we showed how the Toeplitz approach can be an alternative to the classical pencil approach [14] to obtain infinite structural indices of polynomial matrices, and [16] where we showed how we can obtain all the eigenstructure of a polynomial matrix using these Toeplitz methods, in the present work we develop four Toeplitz algorithms to obtain null-spaces of polynomial matrices.

First, we develop two algorithms based on classical numerical linear algebra methods, and in the second part of this work [17] we present two other algorithms based on the displacement structure theory [9], [10]. The motivation for the two algorithms presented in [17] is to apply them to large dimension problems for which the classical methods could be inefficient.

In the next section 2 we define the null-space structure of a polynomial matrix. In section 4 we present the two algorithms to obtain the null-space of $A(s)$, based on the numerical methods described in section 3. In section 5 we analyze the complexity and the stability of the algorithms.

J.C. Zúñiga and D. Henrion are with LAAS-CNRS, 7 Avenue du Colonel Roche, 31077 Toulouse, France.

D. Henrion is also with the Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Pod vodárenskou věží 4, 18208 Prague, Czech Republic, and with the Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, 16627 Prague, Czech Republic.

II. Null-space structure

Consider a $k \times l$ polynomial matrix $A(s)$ of degree d. We define the null-space of $A(s)$ as the set of non-zero polynomial vectors $v(s)$ such that

$$A(s)v(s) = 0. \tag{1}$$

Let $\rho = \operatorname{rank} A(s)$. A basis of the null-space of $A(s)$ is formed by any set of $l - \rho$ linearly independent vectors satisfying (1). Let δ_i be the degree of each vector in the basis. If the sum of all the degrees δ_i is minimal then we have a minimal null-space basis in the sense of Forney [2].

The set of degrees δ_i and the corresponding polynomial vectors are called the *right null-space structure* of $A(s)$.

In order to find each vector of a minimal basis of the null-space of $A(s)$ we have to solve equation (1) for vectors $v(s)$ with minimal degree. Using the Sylvester matrix approach we can write the following equivalent system of equations

$$\begin{bmatrix} A_0 & & & 0 \\ \vdots & A_0 & & \\ A_d & \vdots & \ddots & \\ & A_d & & A_0 \\ & & \ddots & \vdots \\ 0 & & & A_d \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_\delta \end{bmatrix} = 0. \tag{2}$$

Notice that we can also consider the transposed linear system

$$\begin{bmatrix} v_0^T & v_1^T & \cdots v_\delta^T \end{bmatrix} \times \begin{bmatrix} A_0^T & \cdots & A_d^T & & & 0 \\ & A_0^T & \cdots & A_d^T & & \\ & & \ddots & & \ddots & \\ 0 & & & A_0^T & \cdots & A_d^T \end{bmatrix} = 0. \quad (3)$$

If we are interested in the *left null-space structure* of $A(s)$, i.e. the set of vectors $w(s)$ such that $w(s)A(s) = 0$, then we can consider the transposed relation $A^T(s)w^T(s) = 0$. Because of this duality, in the remainder of this work we only explain in detail how to obtain the right null-space structure of $A(s)$, that we call simply null-space of $A(s)$.

III. LQ FACTORIZATION AND COLUMN ECHELON FORM

From equations (2) or (3) we can see that the null-space of $A(s)$ can be recovered from the null-space of some suitable block Toeplitz matrices as soon as the degree δ of the null-space of $A(s)$ is known.

Let us first assume that degree δ is given. The methods that we propose to obtain the constant null-spaces of the corresponding Toeplitz matrices consist in the application of successive Householder transformations in order to reveal their ranks. We consider two methods: the LQ factorization and the Column Echelon Form.

A. LQ factorization

The LQ factorization is the dual of the QR factorization, see [3]. Consider a $k \times l$ constant matrix M. We define the LQ factorization of M as $M = LQ$ where Q is an orthogonal $l \times l$ matrix and $L = [L_r \quad 0]$ is a $k \times l$ matrix that we can always put in a lower trapezoidal form with a row permutation P:

$$PL = \begin{bmatrix} w_{11} & & \times & \\ \vdots & \ddots & & \\ w_{r1} & \cdots & w_{rr} & 0 \\ \vdots & & \vdots & \\ w_{k1} & \cdots & w_{kr} & \end{bmatrix} = [PL_r \quad 0]$$

where $r \leq \min(k, l)$ is the rank of M.

Notice that L has $l - r$ zero columns. Therefore, an orthogonal basis of the null-space of M is given by the last $l - r$ rows of Q, namely, $MQ(r + 1 : l, :)^T = 0$ and moreover, we can check that $M = L_r Q(1 : r, :)$.

We apply the LQ factorization to solve system (2). Here the number of columns of the Toeplitz matrix is not very large, so we obtain the orthogonal matrix Q and, as a by-product, the vectors of the null-space. This method is a cheap version of the one presented in [15] where we used the singular value decomposition (SVD) to obtain the rank and the null-spaces of the Toeplitz matrices. The SVD can be viewed as two successive LQ factorizations so, its execution time is about twice that of the LQ factorization. Nevertheless, in the literature the SVD is the frequently more used method to reveal the rank of a matrix because of its robustness [3], [7]. See section 5 for a brief discussion about the stability of the algorithms presented in this paper.

B. Column Echelon Form (CEF)

For the definition of the reduced echelon form of a matrix see [3]. Let L be the column echelon form of M. The particular form that could have L depends on the position of the r linearly independent rows of M. For example, let us take a 5×4 matrix M. Consider that after the Householder transformations H_i, applied onto the rows of M in order to obtain its LQ factorization, we obtain the column echelon form

$$L = MQ^T = MH_1H_2H_3 = \begin{bmatrix} \times_{11} & 0 & 0 & 0 \\ \times_{21} & 0 & 0 & 0 \\ \times_{31} & \times_{32} & 0 & 0 \\ \times_{41} & \times_{42} & \times_{43} & 0 \\ \times_{51} & \times_{52} & \times_{53} & 0 \end{bmatrix}. \quad (4)$$

This means that the second and the fifth rows of M are linearly dependent.

The column echelon form L allows us to obtain a basis of the left null-space of M. From equation (4) we can see that the coefficients of linear combination for the fifth row of M can be recovered by solving the linear triangular system

$$\begin{bmatrix} k_1 & k_3 & k_4 \end{bmatrix} \begin{bmatrix} \times_{11} & 0 & 0 \\ \times_{31} & \times_{32} & 0 \\ \times_{41} & \times_{42} & \times_{43} \end{bmatrix} = - \begin{bmatrix} \times_{51} & \times_{52} & \times_{53} \end{bmatrix}, \quad (5)$$

and then

$$\begin{bmatrix} k_1 & 0 & k_3 & k_4 & 1 \end{bmatrix} M = 0.$$

Also notice that when obtaining the basis of the left null-space of M from matrix L it is not necessary to store $Q^T = H_1H_2\ldots$. This fact can be used to reduce the algorithmic complexity.

We apply the CEF method to solve the system of equations (3). Since the number of columns of the Toeplitz matrix can be large, we do not store orthogonal matrix Q. We only obtain the CEF and we solve the corresponding triangular systems as in (5) to compute the vectors of the null-space. For more details about the CEF method described above see [6].

IV. BLOCK TOEPLITZ ALGORITHMS

For the algorithms presented here we consider that the rank ρ of $A(s)$ is given. There are several documented algorithms to obtain the rank of a polynomial matrix, see for example [5]. See also [16] where we show how we can obtain the rank of a polynomial matrix while obtaining its eigenstructure.

A. The LQ algorithm

The problem remaining when solving (2) is the estimation of the degree δ. To avoid this problem we process iteratively a block Toeplitz matrix of increasing size.

Consider the block Toeplitz matrix

$$R_i = \begin{bmatrix} A_0 & & & 0 \\ \vdots & A_0 & & \\ A_d & \vdots & \ddots & \\ & A_d & & A_0 \\ & & \ddots & \vdots \\ 0 & & & A_d \end{bmatrix} \qquad (6)$$

with dimensions $k(d+i) \times li$ at step index $i = 1, 2, \ldots$. Let $\gamma_1 = l - \operatorname{rank} R_1$, so that matrix $A(s)$ has γ_1 vectors of degree 0 in the basis of its null-space. The number of vectors of degree 1 is equal to $\gamma_2 = 2l - \operatorname{rank} R_2 - \gamma_1$ and so on. At step i we define

$$\gamma_i = li - \operatorname{rank} R_i - \sum_{j=1}^{i-1} \gamma_j \qquad (7)$$

and then the number of vectors of degree $i-1$ in the basis of the null-space of $A(s)$ is equal to $\gamma_i - \gamma_{i-1}$. Notice that $\gamma_i \leq \gamma_{i+1}$.

When $\gamma_f = l - \operatorname{rank} A(s)$ for some sufficently large index f, we have obtained all the vectors of a minimal basis of the null-space of $A(s)$.

With this algorithm we can take full advantage of the particular Toeplitz structure of matrix R_i. Consider for example a 3×3 polynomial matrix $A(s)$ of degree 2 and suppose that we have already computed the LQ factorization of R_1:

$$R_1 Q_1 = \begin{bmatrix} A_0 \\ \hline A_1 \\ \hline A_2 \end{bmatrix} Q_1 = \begin{bmatrix} \times & 0 & 0 \\ \times & 0 & 0 \\ \times & \times & 0 \\ \hline \times & \times & 0 \\ \times & \times & 0 \\ \times & \times & \times \\ \hline \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} = L_1.$$

In order to compute the LQ factorization of R_2, first we apply Q_1 in the following way:

$$\begin{bmatrix} A_0 & 0 \\ \hline A_1 & A_0 \\ \hline A_2 & A_1 \\ \hline 0 & A_2 \end{bmatrix} \begin{bmatrix} Q_1 & 0 \\ 0 & I_3 \end{bmatrix} = \begin{bmatrix} \times & 0 & 0 & \\ \times & 0 & 0 & 0 \\ \times & \times & 0 & \\ \hline \times & \times & 0 & \\ \times & \times & 0 & A_0 \\ \times & \times & \times & \\ \hline \times & \times & \times & \\ \times & \times & \times & A_1 \\ \times & \times & \times & \\ \hline 0 & 0 & 0 & \\ 0 & 0 & 0 & A_2 \\ 0 & 0 & 0 & \end{bmatrix},$$

and then we compute the smaller factorization

$$\begin{bmatrix} 0 & \\ 0 & A_0 \\ \times & \\ \hline \times & \\ \times & A_1 \\ \times & \\ \hline 0 & \\ 0 & A_2 \\ 0 & \end{bmatrix} \bar{Q}_2 = \bar{L}_2.$$

Finally notice that $R_2 Q_2 = L_2$ where

$$Q_2 = \begin{bmatrix} Q_1 & 0 \\ 0 & I_3 \end{bmatrix} \begin{bmatrix} I_2 & 0 \\ 0 & \bar{Q}_2 \end{bmatrix}$$

is the LQ factorization of R_2.

These relations indicate that the information obtained at step $i-1$ can be used at step i.

Algorithm `LQ`

For a $k \times l$ polynomial matrix $A(s)$ of degree d and rank ρ, this algorithm computes the vectors of a minimal basis of its null-space.

- Step 1:
  Compute the LQ factorization $R_1 Q_1 = L_1$ and obtain the number r_1 of linearly independent rows in the first k rows of R_1.
  From equation (7) obtain γ_1. If $\gamma_1 = \mu \neq 0$ then extract the μ vectors of degree 0 of the basis of the null-space of $A(s)$ from the last μ columns of Q_1.
- Step i:
  Consider that we obtained the LQ factorization $R_{i-1} Q_{i-1} = L_{i-1}$, and that r_{i-1} is the number of linearly independent rows in the first $(i-1)k$ rows of R_{i-1}.
  Compute the LQ factorization

  $$[\bar{L}_{i-1} \quad R_1] \bar{Q}_i = \bar{L}_i$$

  where $\bar{L}_{i-1}$ is L_{i-1} without its first r_{i-1} columns and without its first $(i-1)k$ rows.

The LQ factorization of R_i is given by $R_iQ_i = L_i$ where

$$Q_i = \begin{bmatrix} Q_{i-1} & 0 \\ 0 & I_l \end{bmatrix} \begin{bmatrix} I_{r_{i-1}} & 0 \\ 0 & \bar{Q}_i \end{bmatrix}.$$

Now, from equation (7) obtain γ_i. If $\gamma_i - \gamma_{i-1} = \mu \neq 0$ then extract the μ vectors of degree $i-1$ of the basis of the null-space of $A(s)$ from the last μ columns of Q_i.

- Stopping rule:
  When $\gamma_i = l - \rho$ we have all the vectors of a minimal basis of the null-space of $A(s)$.

B. The CEF algorithm

When solving system (3) we can obtain in a similar way structural indices γ_i from Toeplitz matrices

$$\bar{R}_i = \begin{bmatrix} A_0^T & \cdots & A_d^T & & & 0 \\ & A_0^T & \cdots & A_d^T & & \\ & & \ddots & & \ddots & \\ 0 & & & A_0^T & \cdots & A_d^T \end{bmatrix} \qquad (8)$$

with dimensions $li \times k(d+i)$.

Nevertheless, we cannot take advantage of the structure because the number of columns in matrix $\bar{R}_i$ could be very large and obtaining Q at each step is not efficient. On the other hand, we do not have to analyze $\bar{R}_i$ with $i = 1, 2, \ldots$ when searching the vectors of minimal degree in the basis. We can show that from the CEF of $\bar{R}_p$ we can recover the vectors of degree less than or equal to $p-1$.

For example, consider that a 3×3 polynomial matrix $A(s)$ of degree d has one vector of degree 0 and one vector of degree 5 in the minimal basis of its null-space. Consider that we compute the CEF of $\bar{R}_1$ and that we find a linearly dependent row, for example the third row. It means that the third column of $A(s)$ is a linear combination of degree 0 of the other columns. So, we can find the 0 degree vector by solving the corresponding triangular system as in (5). Now, consider that by computing the CEF of $\bar{R}_2$ we find that the third and 6th rows are linearly dependent. It means that the third column of $A(s)$ is also a linear combination of degree 1 of the other columns, nevertheless we know that the minimal basis has only a vector of degree 0. Finally, consider that when computing the CEF of $\bar{R}_6$ we find a different linearly dependent row, for example the 17th row. It means that the 2nd column of $A(s)$ is a linear combination of degree 5 of the other columns.

In summary, we can compute only the CEF of $\bar{R}_6$ and recover the vectors of the minimal basis of the null-space of $A(s)$. To recover the 0 degree vector we solve the triangular system corresponding to the 3th row of the CEF of $\bar{R}_6$, and to recover the 5 degree vector we solve the triangular system corresponding to the 17th row.

In that way we can easily show that the minimal basis of the null-space of any $k \times l$ polynomial matrix $A(s)$ of degree d can be recovered from the CEF of $\bar{R}_{d_{\max}}$ where

$$d_{\max} = \sum_{i=1}^{l} \deg(\mathrm{col}_i A(s)) - \min_i \deg(\mathrm{col}_i A(s)), \qquad (9)$$

with $\mathrm{col}_i A(s)$ denoting the ith column of $A(s)$. Degree $d_{\max}$ is the maximal possible degree of a vector in the null-space of $A(s)$. See [4, Lemma 4.1] for a proof of relation (9). Nevertheless, computing systematically the CEF of $\bar{R}_{d_{\max}}$ could be computationally expensive. A better approach consists in computing, in a first step, the CEF of $\bar{R}_j$ with $j = 1$ and in finding an optimal way to decide the value of j for the next step. The objective is to minimize the total amount of computations. In the following algorithm we propose as a heuristic to increase j, at each step, by a fixed ratio of $d_{\max}$.

Algorithm `CEF`
For a $k \times l$ polynomial matrix $A(s)$ of degree d and rank ρ, this algorithm computes the vectors of a minimal basis of its null-space.

- Initialization:
  Compute $d_{\max}$ as in (9). Start with $j = 1$ and $\Delta = 0$.
- Step i:
  Compute the CEF of $\bar{R}_j$. Derive the number μ of linearly dependent columns of $A(s)$. Update
  $$\Delta = \Delta + \frac{d_{\max}}{10} \quad \text{and} \quad j = j + \Delta.$$
- Stopping rule:
  When $\mu = l - \rho$ we can find from $\bar{R}_f$, the last CEF calculated, all the vectors in the basis of the null-space of $A(s)$.
- Obtaining the null-space:
  Compute the linear combination of the linearly dependent rows of $\bar{R}_f$ as in (5). If more than one row corresponds to the same linearly dependent column in $A(s)$, take the combination of the smallest degree.

This algorithm was implemented in version 2.0 of the Polynomial Toolbox for Matlab [13].

V. Performance analysis

Several numerical tests have been carried out to analyze the performance of the algorithms `LQ` and `CEF`. The algorithms have been programmed in Matlab using the Polynomial Toolbox [13] on a Sun Ultra 5 work-station. The algorithms have been implemented from scratch, without using Matlab built-in functions such as `qr` or `rref`.

A. Algorithmic complexity

The number of columns η_j of the Toeplitz matrices analyzed at each step depends on the dimension k of the polynomial matrix $A(s)$. For simplicity we consider square matrices. Classical methods to obtain the LQ factorization or the CEF of these Toeplitz matrices have complexity $O(\eta_j^3)$ [3]. So, it follows that the complexity of our algorithms is in $O(k^3)$. Nevertheless, some differences can be noticed. In algorithm LQ we use at each step the information of the previous step, so, the dimension of R_i does not grow by k columns at each step. On the other hand, in algorithm CEF, $\bar{R}_i$ has at least k columns more than $\bar{R}_{i-1}$. These differences result in a smallest execution time for algorithm LQ when k is large. Consider the polynomial matrix

$$A(s) = B(s) \oplus I_a = \begin{bmatrix} B(s) & 0 \\ 0 & I_a \end{bmatrix}$$

where the 4×4 polynomial matrix $B(s)$ of degree 4 and rank equal to 3 has a null space of degree 2. Notice that we can vary the dimension a of the identity matrix in order to modify the dimension k of $A(s)$ without modifying its null-space. In Figure 1 we apply our algorithms to obtain the null-space of $A(s)$ with $a = 1, 2, \ldots, 45$. Notice the difference between the execution times of both algorithms when k is large.

The degree of $A(s)$ is another parameter on which the dimension of the Toeplitz matrices depends. From (8) it is easy to see that a large degree implies a large number of columns in $\bar{R}_i$. We can show that the complexity of algorithm CEF is in $O(d^3)$. On the other hand, from (6) notice that the degree has no direct influence on the number of columns of R_i, but only on the number of rows. We can show that the complexity of algorithm LQ is in $O(d^2)$. Consider the polynomial matrix $A(s) = s^d \oplus B(s)$ where the 4×4 polynomial matrix $B(s)$ of degree 8 and rank equal to 2 has a null space of degree 4. Notice that we can vary the degree $d > 8$ of $A(s)$ without modifying its null-space. In Figure 2 we apply our algorithms to obtain the null-space of $A(s)$ with $d = 15, 22, 29, \ldots, 274$. Notice the difference between the execution times of both algorithms when d is large.

Other parameters can be considered for the complexity analysis, for example, the number of steps needed to find all the vectors in the basis of the null-space of $A(s)$. It is easy to see that if the number of steps is large, the Toeplitz matrices analyzed at the last steps can be very large. The number of steps depends on the degree d_z of the null-space. We can show that the complexity of algorithm LQ is in $O(d_z^3)$ and the complexity of algorithm CEF is in $O(d_z^2)$ because d_z has direct influence only on the number of rows of $\bar{R}_i$. Consider the polynomial matrix $A(s) = s^{50} \oplus B(s)$ whit the 3×3 polynomial matrix $B(s) = [b_1 \quad b_2 \quad s^{d_z} b_1]$. Notice that we can vary the degree d_z of the null-space of $A(s)$ without modifying its dimension k and its degree d. In Figure 3 we apply our algorithms to obtain the null-space of $A(s)$ with $d_z = 0, 1, 2, \ldots, 50$. Notice the smallest execution time for the algorithm CEF when d_z is large. Another important characteristic of the algorithm CEF is that a large d_z implies a large $d_{\max}$, allowing the algorithm to finish in a smallest number of steps.

B. Numerical stability

In brief, the numerical methods LQ and CEF are based on the application of successive Householder transformations.

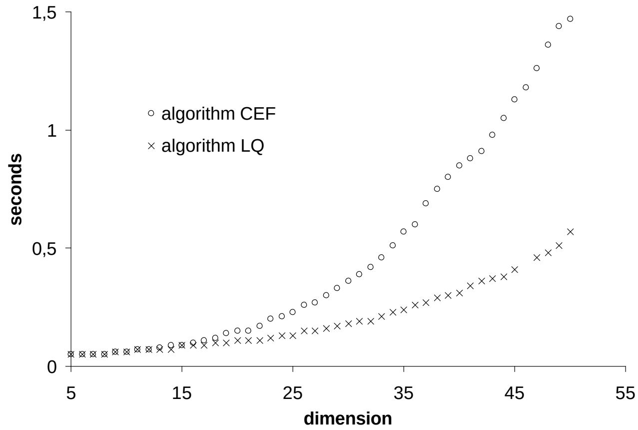

Fig. 1. Execution times for extracting null-spaces of polynomial matrices of different dimensions.

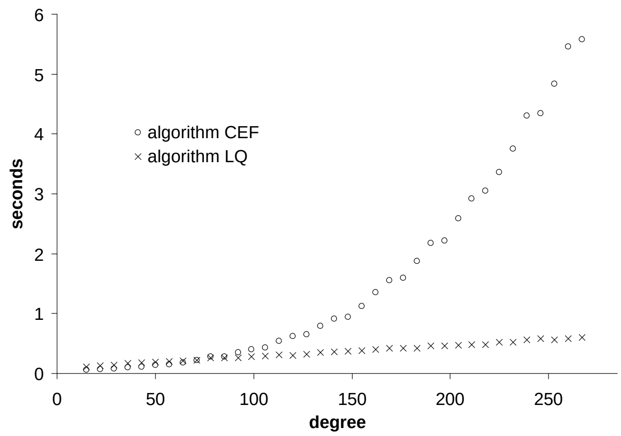

Fig. 2. Execution times for extracting null-spaces of polynomial matrices of different degrees.

It is possible to show that this sequence of transformations is backward-stable in the sense of [7]. So, if $\bar{L}$ is the computed CEF of a matrix A, we can find an orthogonal matrix Q such that $A + \Delta A = Q\bar{L}$ and moreover we can find an upper bound on the norm of ΔA in terms of the machine precision. See Theorems 18.3 and 18.4 in [7].

These numerical properties ensure the backward-stability of methods LQ and CEF. Nevertheless, in algorithms LQ and CEF we use these methods in order to reveal the rank of the corresponding Toeplitz matrices. In fact, the null-space structure of the polynomial matrix is revealed by the substraction of the different ranks computed by methods LQ and CEF at each step. Computationally speaking, the rank of an $n \times n$ matrix A is the number of singular values greater than $n\|A\|_1\epsilon$ where ϵ is the machine precision. In that way, the SVD is the best method to compute the rank of A. Others rank revealing methods like the LQ factorization or the obtaining of the CEF are more sensitive to small

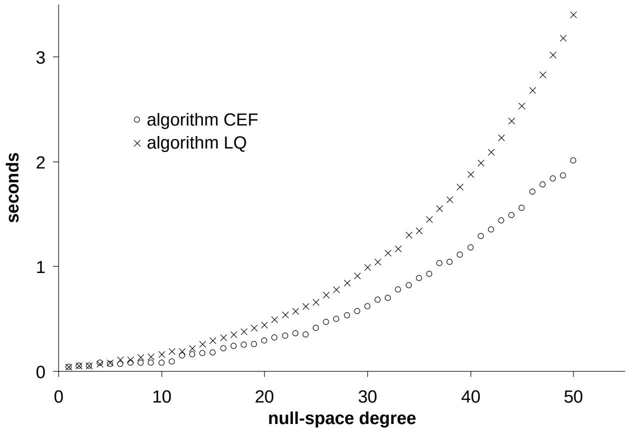

Fig. 3. Execution times for extracting null-spaces of different degrees.

variations in the entries of A. In summary, we cannot assure the backward-stability of algorithms `LQ` and `CEF`. However, we can see that, in practice, they perform quite well. It is difficult to find a polynomial matrix for which algorithms `LQ` and `CEF` fail.

VI. Conclusions

In the present work, we analyzed two different block Toeplitz algorithms to obtain the null-space of a polynomial matrix. One of these algorithms is based on the LQ factorization and the other on the Column Echelon Form (CEF) of succesive block Toeplitz matrices of increasing size.

From the complexity analysis of these algorithms our conclusions are as follows. The advantage of the LQ algorithm consists in the weak dependence on the degree of the analyzed polynomial matrix and in the possibility of using part of the work done at step $i-1$ in the computations at step i.

When the null-space degree is large, the LQ algorithm needs a large number of steps. Then, the dimension of the last Toeplitz matrices could be large, reducing the efficiency.

The advantage of the CEF algorithm consists in the possibility of recovering the vectors of the null-space from the last computed CEF. In other words, we do not have to analyze iteratively all the block Toeplitz matrices. This characteristic implies, in some cases, a smaller number of steps.

When the maximal expected null-space degree is large, the analyzed Toeplitz matrices can be large in a few steps. So, if the polynomial matrix has a null-space with actual low degree, the CEF algorithm can analyze block Toeplitz matrices of larger dimensions than necessary, reducing efficiency.

In summary, the LQ algorithm generally performs better. The CEF algorithm has also a good overall performance, but the actual heuristic to determine the dimension of the Toeplitz matrix at each step is not optimal. Finding this optimal solution, in order to minimize the number of steps, appears as a difficult problem deserving further study.

Regarding numerical stability, we can show that even if the backward-stability of the LQ and CEF algorithms is not guaranteed, they provide reliable results in almost all the practical cases.

See the conclusions of the second part [17] for more about numerical stability and complexity of our algorithms.

Acknowledgments

Juan-Carlos Zúñiga acknowledges support from the National Council of Science and Technology of Mexico (CONACYT) as well as from the Secretariat of Public Education of Mexico (SEP). Didier Henrion acknowledges support from project No. 102/02/0709 of the Grant Agency of the Czech Republic as well as from project No. ME 698/2003 of the Ministry of Education of the Czech Republic.

References

[1] Callier F.M. and Desoer C.A., *Multivariable Feedback Systems.* Springer Verlag, Berlin, 1982.
[2] Forney G., Minimal bases of rational vector spaces with applications to multivariable linear systems. *SIAM Journal on Control*, vol. 13, no. 3, pp. 493-520, 1975.
[3] Golub G.H. and Van Loan C.F., *Matrix Computations.* The Johns Hopkins University Press, New York, 1996.
[4] Henrion, D., *Reliable Algorithms for Polynomial Matrices.* Ph.D. Thesis, Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Prague, 1998.
[5] Henrion, D. and Šebek, M., Numerical methods for polynomial rank evaluation. *Proc. of the IFAC Symposium on System, Structure and Control*, Nantes, France, 1998.
[6] Henrion, D. and Šebek, M., An algorithm for polynomial matrix factor extraction. *Int. J. Control*, vol. 73, no. 8, pp. 686-695, 2000.
[7] Higham, N.J., *Accuracy and Stability of Numerical Algorithms*. SIAM, Philadelphia, 1996.
[8] Kailath T., *Linear Systems*. Prentice Hall, Englewood Cliffs, 1980.
[9] Kailath T. and Sayed A. H., Displacement Structure: theory and applications. *SIAM Review*, vol. 37, pp. 297-386, 1995.
[10] Kailath T. and Sayed A. H., *Fast Reliable Algorithms for Matrices with Structure.* SIAM, Philadelphia, 1999.
[11] Kučera V., *Discrete Linear Control: The Polynomial Equation Approach.* John Wiley and Sons, Chichester, 1979.
[12] Nyberg M. and Frisk E., A minimal polynomial basis solution to residual generation for fault diagnostics in linear systems. *Proc. of the IFAC World Congress*, Beijing, China, 1999.
[13] Polyx, Ltd. *The Polynomial Toolbox for Matlab.* Version 2.5 released in 2000. See `www.polyx.cz`
[14] Van Dooren P. and Dewilde P., The Eigenstructure of an Arbitrary Polynomial Matrix. Computational Aspects. *Linear Algebra and Its Applications*, Vol. 50, pp. 545–580, 1983.
[15] Zúñiga J.C. and Henrion D., Comparison of algorithms for computing infinite structural indices of polynomial matrices. *Proc. of the European Control Conference*, Cambridge, Uk, 2003.
[16] Zúñiga J.C. and Henrion D., Block Toeplitz Methods in Polynomial Matrix Computations. *To be registered as a LAAS-CNRS research report. Submitted to the Symposium on Mathematical Theory of Networks and Systems*, Leuven, Belgium, July 5-9, 2004.
[17] Zúñiga J.C. and Henrion D., On the application of different numerical methods to obtain null-spaces of polynomial matrices. Part 2: block displacement structure algorithms. *To be registered as a LAAS-CNRS research report. Submitted to the IEEE Conference on Decision and Control*, Paradise Island, Bahamas, December 14-17, 2004.