Extraction of Infinite Zeros of Polynomial Matrices*

Didier Henrion[1,2,3]
Javier Ruiz León[4]
Michael Šebek[5,6]

Abstract

An algorithm is described for computing the structure at infinity and extracting the infinite zeros of a given polynomial matrix. The algorithm relies on numerically reliable operations only. Applications include computation of the subspace of impulsive solution of a set of linear differential equations, derivation of the Smith form at infinity of a polynomial matrix, or also enhanced computation of the characteristic polynomial of a linear system described by polynomial matrix fractions. The numerical routines described in this paper are implemented in the new release 3.0 of the Polynomial Toolbox for Matlab.

Keywords: Linear Systems, Polynomial Matrix, Infinite Zeros, Stucture at Infinity.

1 Introduction

Polynomials and polynomial matrices arise naturally and cannot be avoided in linear system theory. It is taught in every signals and systems course that first principle modeling of single-input single-output linear systems with input u and output y after elimination of all internal variables very often leads to differential equations of the form

$$D(\frac{d}{dt})y(t) = N(\frac{d}{dt})u(t) \tag{1}$$

[1]Laboratoire d'Architecture et d'Analyse des Systèmes, Centre National de la Recherche Scientifique, 7 avenue du Colonel Roche, 31077 Toulouse, cedex 4, France.

[2]Institut National des Sciences Appliquées, 135 avenue de Rangueil, 31077 Toulouse, cedex 4, France.

[3]Corresponding author. FAX: +33 5 61 33 69 69. E-mail: `henrion@laas.fr`

[4]Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, Unidad Guadalajara, P.O. Box 31-438, 44550 Guadalajara, Jalisco, Mexico.

[5]Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Pod vodárenskou věží 4, 18208 Prague 8, Czech Republic.

[6]Trnka Laboratory for Automatic Control, Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University, Technická 2, 16627 Prague 6, Czech Republic.

*This work was supported by the National Council of Science and Technology of Mexico (CONACYT) under grant No. 31844-A, by the Grant Agency of the Czech Republic under contract No. 102/99/1368 and by the Ministry of Education of the Czech Republic under contract No. VS97/034.

with D and N polynomials. Indeed, D and N are the denominator and numerator polynomials, respectively, of the transfer function

$$H(s) = \frac{N(s)}{D(s)} \tag{2}$$

of the system. For multi-input multi-output systems with vector-valued input u and vector-valued output y a description in the form of the IO differential equation (1) still holds except that D and N are polynomial matrices with dimensions determined by the dimensions of u and y. The system now has a transfer matrix that may be expressed as

$$H(s) = D^{-1}(s)N(s) \tag{3}$$

This transfer matrix is in polynomial matrix fraction form. More details on polynomial matrices and polynomial matrix fractions can be found in numerous textbooks [15, 14, 4].

Zeros of polynomial matrices naturally represent either poles or zeros of linear multivariable systems described by polynomial matrix fractions. They are frequently encountered when analyzing and/or designing linear systems or filters. Associated with the zeros are the finite and infinite structures of a polynomial matrix, defined from specific canonical forms under matrix equivalence:

- The finite structure of a polynomial matrix $A(s)$ is displayed in its Smith form, where the finite zeros of $A(s)$ are the roots of the invariant polynomials appearing on the diagonal of the Smith form of $A(s)$ [7].
- The infinite structure of $A(s)$ is defined via its Smith-MacMillan form at infinity, which can be regarded as a canonical form considering equivalence through biproper matrices [25]. As part of its infinite structure, any polynomial matrix may have a certain number of infinite poles and infinite zeros of different orders.

In this paper, we are concerned with developing a numerical algorithm for computing the structure at infinity and extracting the infinite zeros of a given polynomial matrix.

Our first motivation in doing so is that important system features are captured by the structure at infinity of polynomial matrices. For instance, the infinite zeros of the standard linear system

$$\begin{aligned} dx(t)/dt &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned}$$

correspond to the infinite zeros of the polynomial matrix

$$\begin{bmatrix} sI - A & B \\ C & 0 \end{bmatrix}$$

also known as the system matrix. Dynamically, infinite zero orders of the system are related to the number of times that the outputs of the system have to be differentiated for a component of the input vector to appear explicitly. Thus, this information is crucial in the solution of important control problems. For instance, a linear multivariable system is row by row decouplable by static state feedback if and only if the sum of its infinite zero orders equals the sum of its row infinite zero orders [6]. Other typical problems where the infinite structure of the system plays a fundamental role are model matching [17] and disturbance rejection [3]. Infinite zeros were recently used for decoupling and model matching [18] and eigenstructure assignment of descriptor systems [27].

Our second motivation originates from the study of the impulsive solutions of linear homogeneous matrix differential equations

$$A(\frac{d}{dt})x(t) = 0$$

where $A(s)$ is an $n \times n$ non-singular polynomial matrix in the differential operator $s = d/dt$, and $x(t)$ is an n-dimensional vector valued function assumed to be infinitely continuously differentiable and that is to be found. As shown in [25, §4.2.2], the subspace of impulsive solutions to the above differential equation is strongly related to the infinite structure of polynomial matrix $A(s)$. This infinite structure can be computed by the algorithm that will be proposed in this paper.

Our third motivation comes from difficulties encountered when developing numerical routines for computing the determinant hence the zeros of polynomial matrices. Recent numerical experiments [20] reveal that two of the best known techniques for polynomial matrix determinant and zeros computation are

- interpolation of the determinant via FFT (see [13] for a detailed treatment) and computation of the roots of the determinant via Schur decomposition of a companion matrix [8];
- direct computation of the zeros via the QZ decomposition of a pencil companion matrix [16, 8].

One of the main stumbling blocks when computing the determinant and the zeros is the presence of indesirable infinite zeros, namely

- when interpolating the determinant of a polynomial matrix, infinite zeros will introduce artificial non-zero leading coefficients in the determinant. An ad-hoc technique called zeroing is then traditionally used to cancel these terms and to avoid undue degree swelling when performing subsequent computations [20];
- when pursuing the other approach, the zeros are computed as the ratios of the diagonal entries of two matrices simultaneously reduced to upper triangular form via the QZ decomposition. Infinite zeros theoretically correspond to zero denominators in the ratios. Practically, these denominators are small, non-zero numbers and it is quite hard, when not impossible, to distinguish between genuine polynomial matrix zeros and zeros artifically introduced by numerical round-off errors [20].

The algorithm proposed in this paper aims at extracting the infinite zeros of a polynomial matrix without altering its finite zero structure. Therefore, it may be viewed as a preprocessing step for getting rid of the above mentioned issues and improving polynomial matrix zero computation. Naturally, the algorithm can also improve polynomial matrix determinant computation and may be considered as a alternative approach to the technique recently proposed in [11].

In the numerical linear algebra literature, several algorithms based on generalized state-space realizations and the correspondance between the Smith-MacMillan form of a rational matrix and the Kronecker canonical form of an associated pencil matrix were proposed by Van Dooren and co-workers [23, 2]. These pencil algorithms allow to compute at once the whole structure (finite and infinite) of a polynomial matrix [24]. In contrast, the algorithm proposed in our paper is not based on the pencil matrix associated with the input polynomial matrix. It is rather similar in spirit to the algorithm proposed in [22]. In this reference, the structure of a rational matrix at any point is computed via orthogonal transformations on Toeplitz matrices built from the Laurent expansion.

The algorithm described in this paper in an extension of a method recently proposed in [12] to compute and extract the finite structure of a polynomial matrix. The algorithm is designed while keeping with our main impetus, which is the development of reliable numerical methods for dealing with polynomial matrices [9, 10] and their implementation in a user-friendly MATLAB package called the POLYNOMIAL TOOLBOX [21, 19]. In this regard, all the routines used in the algorithm are numerically stable in the sense that they are based on backward stable (orthogonal) transformations. Moreover, the routines take advantage of the special structure of the matrices so as to reduce the overall computational cost. In this regard, our algorithm can thus be viewed as a "polynomial approach" alternative to the "state-space approach" algorithms proposed in [23, 2] and recently implemented in the FORTRAN library SLICOT [26].

The outline of the paper is as follows. In Section 2 we recall the notions of zeros and structure at infinity. Then we state the problem to be solved. In Section 3 we propose a subroutine for computing the structure a polynomial matrix at infinity. This subroutine will be instrumental to the design in Section 4 of an algorithm for extracting infinite zeros from a polynomial matrix. The algorithm will be illustrated on our motivating problems: computation of the subspace of impulsive solutions of a linear differential equation (Section 5.1) and neat distinction between finite and infinite zeros of a polynomial matrix (Section 5.2).

The numerical routines described in this paper are implemented in the new release 3.0 of the POLYNOMIAL TOOLBOX for MATLAB [19].

2 Infinite Zeros and Infinite Structure

In this section, we recall standard facts on the structure of polynomial matrices at infinity.

Consider the $n \times n$ non-singular polynomial matrix

$$A(s) = A_0 + sA_1 + .. + s^{d-1}A_{d-1} + s^dA_d$$

and define as in [25, §4.2.1] the dual polynomial matrix $\tilde{A}(s)$ of $A(s)$ as

$$\tilde{A}(s) = A_d + sA_{d-1} + .. + s^{d-1}A_1 + s^d A_0.$$

For any polynomial matrix $A(s)$ there exist biproper matrices $U_L(s)$, $U_R(s)$ and a unique matrix $S^{\infty}_{A(s)}(s)$, known as the Smith-MacMillan form at infinity of $A(s)$, such that

$$U_L(s)A(s)U_R(s) = S^{\infty}_{A(s)}(s) = \mathrm{diag}\{s^{q_1}, \dots, s^{q_k}, \frac{1}{s^{q_{k+1}}}, \dots, \frac{1}{s^{q_r}}\}$$

where $q_1 \geq \cdots \geq q_k$ and $q_{k+1} \leq \cdots \leq q_r$ are positive integers. The matrix $S^{\infty}_{A(s)}(s)$ contains the infinite structure of the polynomial matrix $A(s)$, where $A(s)$ has infinite poles of orders $q_1, \dots, q_k$ and infinite zeros of orders $q_{k+1}, \dots, q_r$.

The infinite structure of the matrix $A(s)$ can also be obtained as

$$S^{\infty}_{A(s)}(s) = s^{q_1} S^0_{\tilde{A}(s)}(\frac{1}{s})$$

where $S^0_{\tilde{A}(s)}(s)$ is the local Smith form of the dual matrix $\tilde{A}(s)$ at the origin, see [25].

Since the structure of polynomial matrix $A(s)$ at infinity is strongly related to the structure of dual polynomial matrix $\tilde{A}(s)$ at the origin, we define in the sequel the generalized characteristic vectors at the origin [1, 12].

Theorem 1 *If $s = 0$ is a zero of dual polynomial matrix $\tilde{A}(s) = A_d + A_{d-1}s + \cdots + A_1 s^{d-1} + A_0 s^d$ with algebraic multiplicity m and geometric multiplicity p then there exists $m = k_1 + k_2 + \cdots + k_p$ vectors v_{i1}, v_{i2}, $\cdots, v_{ik_i}$, $i = 1, 2, \dots, p$ such that*

$$\begin{bmatrix} A_d & & & 0 \\ A_{d-1} & A_d & & \\ \vdots & & \ddots & \\ A_{d-k_i+1} & \cdots & A_{d-1} & A_d \end{bmatrix} \begin{bmatrix} v_{i1} \\ v_{i2} \\ \vdots \\ v_{ik_i} \end{bmatrix} = 0$$

with v_{11}, v_{21}, $\dots v_{p1}$ non-zero and linearly independent. Vectors v_{ij} are generalized characteristic vectors (GCVs) of $\tilde{A}(s)$ at the origin. Integer k_i is the length of the ith chain of GCVs.

Note that the presence of infinite zeros in $A(s)$ is related to the leading coefficient matrix in $A(s)$. However, up to our knowledge, there is no simple test available for detecting infinite zeros in a given polynomial matrix. The simplest test of absence of zeros at infinity we are aware of is recalled in [11]. It is based on Sylvester matrices and was first indirectly proposed in [22]. Note that it is true that column- or row-reducedness of a polynomial matrix implies absence of infinite zeros. But the converse is not true. A matrix may have no infinite zeros and being neither column- nor row-reduced. Similarly, singularity of the highest coefficient matrix does not imply presence of infinite zeros, see e.g. Comment 2.1 in [5].

3 Computation of the Infinite Structure

As shown in Theorem 1, any generalized characteristic vector v_{ij} of dual matrix $\tilde{A}(s)$ at the origin belongs to the null-space of a Toeplitz matrix built from coefficient matrices A_i. In the sequel we will assume for notational simplicity and without loss of generality that integers k_i are such that $k_1 \leq k_2 \leq \cdots \leq k_p$. Recalling Theorem 1 it holds

$$\underbrace{\begin{bmatrix} v'_{1k_1} & \cdots & v'_{11} & & & 0 \\ v'_{2k_2} & & \cdots & v'_{21} & & \\ \vdots & & & & \ddots & \\ v'_{pk_p} & & & & \cdots & v'_{p1} \end{bmatrix}}_{\mathbf{V}} \underbrace{\begin{bmatrix} A'_d & & & 0 \\ A'_{d-1} & A'_d & & 0 \\ \vdots & & \ddots & \\ A'_{d-k_p+1} & \cdots & A'_{d-1} & A'_d \end{bmatrix}}_{\mathbf{T}_{k_p-1}} = 0. \tag{4}$$

In view of the above matrix equation, computing GCVs amounts to extracting some specific vectors from the left null-space of Toeplitz matrix $\mathbf{T}_k$. In [12], an algorithm called GCV is designed to perform this extraction. It is based on the two following subroutines

- Algorithm CEF: reduction into column echelon form via successive column Householder transformations;
- Algorithm NullREF: computation of a null-space basis in row echelon form via back substitutions.

The GCVs are then retrieved from specific rows of the null-space basis in row echelon form. Both subroutines are based on numerically stable operations only. Moreover, they take advantage of the special Toeplitz structure of matrix $\mathbf{T}_k$. The interested reader is referred to [12] for a comprehensive description of these subroutines. For conciseness, in the sequel we will use the following notation

- Algorithm Struct∞: computation of the structure at infinity of a given polynomial matrix with the help of Algorithms CEF and NullREF.

Here also, the interested reader is referred to the description of Algorithm GCV in [12] for a description of Algorithm Struct∞.

By definition, the local Smith form of $A(s)$ at infinity contains factors $s^{d-k_1}, s^{d-k_2}, \ldots, s^{d-k_p}$ in p different locations on the diagonal. Consequently, one can readily derive the Smith form of a polynomial matrix at infinity by calling Algorithm Struct∞.

4 Extraction of Infinite Zeros

As stated in the introduction, we want to extract the infinite zeros from polynomial matrix $A(s)$. Equivalently, we want to extract the zeros at the origin from the dual polynomial matrix $\tilde{A}(s)$, i.e. we seek a right polynomial matrix $\tilde{R}(s)$ with zeros at the origin such that

$$\tilde{A}(s) = \tilde{L}(s)\tilde{R}(s) \tag{5}$$

and left polynomial matrix factor $\tilde{L}(s)$ has no zeros at the origin. Now if $R(s)$ is the polynomial matrix dual to $\tilde{R}(s)$, it follows that

$$A(s) = L(s)R(s)$$

where $L(s)$ is the left polynomial matrix factor dual to $\tilde{L}(s)$.

It is important to keep in mind that dual matrix $\tilde{R}(s)$ must have zeros at the origin only. In particular, $\tilde{R}(s)$ must have no zero at infinity, so that $R(s)$ be a unimodular right factor, i.e. a polynomial matrix with zeros at infinity only.

If equation (5) holds then $\tilde{A}(s)$ and $\tilde{R}(s)$ share the same right structure at the origin. The idea is then to extract dual factor $\tilde{R}(s)$ with the help of the GCVs of dual polynomial matrix $\tilde{A}(s)$ at the origin, as computed with Algorithm STRUCT∞.

Let $\tilde{R}(s) = R_q + R_{q-1}s + .. + R_1 s^{q-1} + R_0 s^q$ where q is the degree of $\tilde{R}(s)$ and $R(s)$. From Theorem 1, it holds

$$\begin{bmatrix} R_q & & & 0 \\ R_{q-1} & R_q & & \\ \vdots & & \ddots & \\ R_{q-k_i+1} & \cdots & R_{q-1} & R_q \end{bmatrix} \begin{bmatrix} v_{i1} \\ v_{i2} \\ \vdots \\ v_{ik_i} \end{bmatrix} = 0, \quad i = 1, 2, \ldots, p \tag{6}$$

where p is the geometric multiplicity of the zero of $A(s)$ at the origin. Let

$$\tilde{R} = \begin{bmatrix} R_q & R_{q-1} & \cdots & R_1 & R_0 \end{bmatrix}.$$

Equation (6) can equivalently be written

$$\tilde{R}V_i = 0$$

where

$$V_i = \begin{bmatrix} v_{i1} & v_{i2} & \cdots & v_{ik_i} \\ & v_{i1} & & \\ & & \ddots & \vdots \\ 0 & & & v_{i1} \end{bmatrix} \tag{7}$$

is a $n(q+1) \times k_i$ constant matrix associated with the ith chain of GCVs corresponding to the zero at the origin of geometric multiplicity p. If now we consider all the chains of GCVs associated with this zero, we get

$$\tilde{R}\underbrace{\begin{bmatrix} V_1 & V_2 & \cdots & V_p \end{bmatrix}}_{W} = 0. \tag{8}$$

In reference [1], relation (8) is referred to as an interpolation equation for polynomial matrix $\tilde{R}(s)$.

Note however that no assumption has been made so far regarding the degree q of polynomial matrix $\tilde{R}(s)$. Its value is not known beforehand, yet its choice appears to be crucial since inner matrix dimensions in equation (8) depend on q. The same remark can be made about column degrees of $\tilde{R}(s)$. In [12] it is shown that Algorithms CEF and NULLREF can be applied to equation (8) to produce a factor $\tilde{R}(s)$ with minimum column degrees. Actually, it can be proved that the resulting factor is always column- and row-reduced with identity column- and row-leading coefficient matrices. Therefore, dual factor $\tilde{R}(s)$ has no zero at infinity, and factor $R(s)$ is unimodular, as required. The interested reader is referred to [12] for a comprehensive description of the factor extraction algorithm. For conciseness, we only provide below the synopsis of the overall procedure.

Algorithm EXTRACT∞ : Extraction of Infinite Zeros

Input A non-singular $n \times n$ polynomial matrix $A(s)$.

Output A unimodular polynomial matrix $R(s)$ such that there exists a non-singular matrix $L(s)$ with no zero at infinity that satisfies $A(s) = L(s)R(s)$.

Step 1 Compute with Algorithm STRUCT∞ the GCVs of dual polynomial matrix $\tilde{A}(s)$ at the origin. Build interpolation matrix W based on equations (6), (7) and (8).

Step 2 With Algorithms CEF and NULLREF, extract a basis in row echelon form for the left null-space of matrix W.

Step 3 For each index $i = 1, 2, \ldots, n$ select the row in the null-space basis that features a leading entry at column location $i + k_i n$ for the smallest possible integer $k_i \geq 0$. The matrix $\tilde{R} = [R_q \quad R_{q-1} \quad \cdots \quad R_1 \quad R_0]$ made up from these rows gives rise to the required polynomial matrix factor $R(s) = R_0 + R_1 s + \cdots + R_{q-1}s^{q-1} + R_q s^q$.

5 Illustrative Examples

In this section we provide three illustrative examples of the techniques described above. In the first example we use Algorithm STRUCT∞ to derive the structure of a polynomial matrix at infinity and build a basis for the impulsive solution space of a linear differential equation. In the second and third examples, we show how Algorithm EXTRACT∞ can be used to distinguish between finite and infinite polynomial matrix zeros.

All the numerical computations were performed in MATLAB language. In order to save space and to improve the presentation of the results, we rounded the entries of the numerical matrices to the nearest integers. To avoid any confusion, we underline the fact that we never used any kind of symbolic operation at any time.

5.1 Impulsive Solutions of Differential Equations

Consider the set of linear homogeneous differential equation

$$\begin{array}{rcl} x_1(t) + d^3x_2(t)/dt & = & 0 \\ x_2(t) + dx_3(t)/dt & = & 0 \\ x_3(t) & = & 0 \end{array}$$

defined for $t \geq 0$ and studied in [25, Example 4.48]. Let

$$A(s) = \begin{bmatrix} 1 & s^3 & 0 \\ 0 & 1 & s \\ 0 & 0 & 1 \end{bmatrix}$$

be the associated polynomial matrix. Note that $A(s)$ is unimodular, i.e. it has no finite zeros. We are interested in the impulsive solutions of the above set of equations.

Following Algorithm STRUCT∞, we successively build Toepliz matrices $\mathbf{T}_k$ as in equation (4) for increasing values of $k = 0, 1, \ldots$. Using Algorithm REF, we find that there is no linearly dependent rows in the last block row of $\mathbf{T}_k$ for $k = 8$. For illustration, Toeplitz matrix $\mathbf{T}_7$ is shown below, where entries equal to one are represented by $+$ and zero entries are left blank.

$$\mathbf{T}_7 = \left[\begin{array}{ccc|ccc|ccc|ccc|cc|c|c}
&&&&&&&&&&&&&&& \\
+&&&&&&&&&&&&&&& \\
&&&&&&&&&&&&&&& \\ \hline
&&&&&&&&&&&&&&& \\
&&&+&&&&&&&&&&&& \\
&&&&&&&&&&&&&&& \\ \hline
&&&&&&&&&&&&&&& \\
&&&&&&+&&&&&&&&& \\
&+&&&&&&&&&&&&&& \\ \hline
+&&&&&&&&&&&&&&& \\
&+&&&&&&&&+&&&&&& \\
&&+&&+&&&&&&&&&&& \\ \hline
&&&+&&&&&&&&&&&& \\
&&&&+&&&&&&&&+&&& \\
&&&&&+&&+&&&&&&&& \\ \hline
&&&&&&+&&&&&&&&& \\
&&&&&&&+&&&&&&&+& \\
&&&&&&&&+&&+&&&&& \\ \hline
&&&&&&&&&+&&&&&& \\
&&&&&&&&&&+&&&&&+ \\
&&&&&&&&&&&+&&+&&
\end{array}\right].$$

Using Algorithm NULLREF, we obtain the following basis in row echelon form for the left null-space of matrix $\mathbf{T}_7$, where entries equal to minus one are represented by $-$.

$$
\mathbf{V}_7 = \left[\begin{array}{ccc|ccc|ccc|cc|c|c|c}
+ & & & & & & & & & & & & & \\
 & & + & & & & & & & & & & & \\
 & & & + & & & & & & & & & & \\
\hline
 & & & & & + & & & & & & & & \\
\hline
 & & & & & & + & & & & & & & \\
 & - & & & & & & & & + & & & & \\
 & & & & - & & & & & & & + & & \\
 & & & & & & & - & & & & & + & \\
\hline
 & & & & & & & & + & & - & & & +
\end{array}\right].
$$

A first chain of $k_1 = 2$ GCVs can be extracted from row $t_1 = 4$ and the above matrix. A second chain of $k_2 = 7$ GCVs can be extracted from row $t_2 = 9$ of the above matrix. Consequently, the local Smith form of polynomial matrix $\tilde{A}(s)$ at infinity reads

$$
S_\infty(s) = s^3 \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{s^2} & 0 \\ 0 & 0 & \frac{1}{s^7} \end{bmatrix} = \begin{bmatrix} s^3 & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & \frac{1}{s^4} \end{bmatrix}.
$$

As explained in [25, Example 4.48], the GCVs of the second chain, namely vectors

$$
v_{21} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad v_{22} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad v_{23} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad v_{24} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}
$$
$$
v_{25} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad v_{26} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad v_{27} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}
$$

give rise to the impulsive solutions $x^\infty(t)$ of the differential equation provided the initial condition $x(0-)$ and its derivatives $x^{(1)}(0-)$, $x^{(2)}(0-)$ are chosen appropriately. More specifically, we have

- $x_1^\infty(t) = \begin{bmatrix} \delta(t) \\ 0 \\ 0 \end{bmatrix}$ if $x(0-) = -v_{22}$, $x^{(-1)}(0-) = -v_{23}$ and $x^{(-2)}(0-) = -v_{24}$,

- $x_2^\infty(t) = \begin{bmatrix} \delta^{(1)}(t) \\ 0 \\ 0 \end{bmatrix}$ if $x(0-) = -v_{23}$, $x^{(1)}(0-) = -v_{24}$ and $x^{(2)}(0-) = -v_{25}$,

- $x_3^\infty(t) = \begin{bmatrix} \delta^{(2)}(t) \\ 0 \\ 0 \end{bmatrix}$ if $x(0-) = -v_{24}$, $x^{(2)}(0-) = -v_{25}$ and $x^{(2)}(0-) = -v_{26}$,

- $x_4^\infty(t) = \begin{bmatrix} \delta^{(3)}(t) \\ -\delta(t) \\ 0 \end{bmatrix}$ if $x(0-) = -v_{25}$, $x^{(1)}(0-) = -v_{26}$ and $x^{(2)}(0-) = -v_{27}$

where $\delta(t)$ is the Dirac impulsion. The above four vectors span the subspace of all impulsive solutions to the matrix differential equation.

5.2 Extraction of Infinite Zeros

Let

$$A(s) = \left[\begin{array}{cc} 1-s^2-2s^3+2s^4 & 3s-4s^2+s^3-2s^4+2s^5 \\ 1-s-s^2+s^3 & 1-s-s^3+s^4 \end{array}\right]$$

be a non-singular polynomial matrix. We would like to compute the zeros of $A(s)$. Following the approach proposed in [16], we build the singular matrix pencil

$$P(s) = \left[\begin{array}{cc|cc|cc|cc|cc} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline -1 & 0 & 0 & -3 & 1 & 4 & 2 & -1 & -2 & 2 \\ -1 & -1 & 1 & 1 & 1 & 0 & -1 & 1 & 0 & -1 \end{array}\right] + s \left[\begin{array}{cc|cc|cc|cc|cc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

whose generalized eigenvalues are zeros of polynomial matrix $A(s)$. These eigenvalues are obtained as ratios of diagonal elements of the matrices obtained via the QZ decomposition [8] of the above pencil. With the MATLAB built-in macro `qz`, we obtained the following eigenvalues

$$\begin{array}{rcl} \infty &=& (-2.2361 + i0.0000)/(0.0000 + i0.0000) \\ 3745.0224 + i0.7938 &=& (1.0000 + i0.0000)/(0.0003 + i0.0000) \\ 1197.2563 - i3503.8749 &=& (1.1831 + i0.0048)/(0.0001 + i0.0003) \\ 1195.7419 + i3504.3632 &=& (1.1085 + i0.0060)/(0.0000 - i0.0003) \\ -2926.3942 - i2166.3825 &=& (0.8133 - i0.0050)/(-0.0002 + i0.0001) \\ -2927.3301 + i2165.0920 &=& (0.8947 - i0.0029)/(-0.0002 - i0.0001) \\ 1.0000 + i0.0000 &=& (-0.5067 + i0.0000)/(-0.5067 + i0.0000) \\ 1.0000 + i0.0000 &=& (0.9829 + i0.0035)/(0.9829 + i0.0035) \\ 1.0000 + i0.0000 &=& (-0.8881 + i0.0032)/(-0.8881 + i0.0032) \\ 1.0000 + i0.0000 &=& (-1.0596 + i0.0000)/(-1.0596 + i0.0000). \end{array}$$

Using macro `smith` of the POLYNOMIAL TOOLBOX, we can compute the Smith form

$$\left[\begin{array}{cc} -1 & 2s \\ 1+s & 1-2s-2s^2 \end{array}\right] A(s) \left[\begin{array}{cc} -1 & s \\ 0 & -1 \end{array}\right] = \left[\begin{array}{cc} (1-s)^2 & 0 \\ 0 & (1-s)^2 \end{array}\right]$$

thus showing that the first 6 eigenvalues computed with the QZ decomposition actually correspond to infinite zeros of polynomial matrix $A(s)$.

Now we will illustrate how Algorithm EXTRACT∞ can be used to extract these infinite zeros from $A(s)$. With Algorithm STRUCT∞ we compute the GCVs of dual polynomial matrix $\tilde{A}(s)$ at the origin and obtain the interpolation matrix

$$W = \left[\begin{array}{cccccc} 1 & 2 & 0 & 0 & 0 & 0 \\ 0 & -1 & -2 & 1 & 1 & -2.5 \\ \hline 0 & 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & -1 & -2 & 1 & 1 \\ \hline 0 & 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & -1 & -2 & 1 \\ \hline 0 & 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & -1 & -2 \\ \hline 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

corresponding to a zero at the origin of algebraic multiplicity 6 and geometric multiplicity 1. Algorithm NULLREF allows us to derive a basis in row echelon form for the left null-space of the above matrix. It reads

$$\tilde{R} = \left[\begin{array}{cc|cc|cc|cc|cc|cc|cc} -0 & 2 & 2 & 2 & 2 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & -2 & -1 & -1 & -2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0 & 2 & 2 & 1 & 1 & 2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & -2 & -2 & -2 & -2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -0 & 2 & 2 & 2 & 2 & 2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -0 & 0 & -0 & -0 & -0 & -0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right].$$

Using the row selection scheme described in Algorithm EXTRACT∞, rows 1 and 2 in the above matrix give rise to the following minimum column-degree right factor

$$R(s) = \begin{bmatrix} 1+2s+2s^2 & 3s+2s^2+2s^3 \\ -s-2s^2 & 1-2s-s^2-2s^3 \end{bmatrix}.$$

One can check that $R(s)$ is unimodular, as expected. Left factor $L(s)$ such that $A(s) = L(s)R(s)$ is readily retrieved with the numerically stable macro `xab` of the POLYNOMIAL TOOLBOX, see [21]. We obtain

$$L(s) = \begin{bmatrix} 1-2s+s^2 & 0 \\ 1-2s+s^2 & 1-2s+s^2 \end{bmatrix}$$

which appears to be a triangular form of $A(s)$, see [10]. All the infinite zeros of $A(s)$ are captured by $R(s)$. All the finite zeros of $A(s)$ are captured by $L(s)$.

Now if we apply the QZ decomposition to the matrix pencil

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & -0 & 2 & -0 \\ -1 & -1 & 2 & 2 \end{bmatrix} + s \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

associated with polynomial matrix $L(s)$, we obtain the required eigenvalues

$$\begin{array}{rcl} 1.0000 + i0.0000 & = & (-1.0041 + i0.0000)/(-1.0041 + i0.0000) \\ 1.0000 + i0.0000 & = & (1.1135 + i0.0000)/(1.1135 + i0.0000) \\ 1.0000 + i0.0000 & = & (-0.9437 + i0.0000)/(-0.9437 + i0.0000) \\ 1.0000 + i0.0000 & = & (0.9478 + i0.0000)/(0.9478 + i0.0000) \end{array}$$

that are not corrupted anymore by infinite zeros of $A(s)$.

6 Conclusion

In this paper we have proposed numerically reliable algorithms to compute and extract the structure at infinity of a given polynomial matrix. Our algorithms, implemented in the POLYNOMIAL TOOLBOX 3.0 for MATLAB [19], can be viewed as alternatives to the pencil algorithms implemented in the FORTRAN routines of the SLICOT library [26]. A natural continuation of our work is therefore a thorough numerical comparison of these routines, both in terms of numerical reliability and computational requirements.

References

[1] P. J. Antsaklis and Z. Gao, "Polynomial and Rational Matrix Interpolation : Theory and Control Applications", *International Journal of Control*, Vol. 58, No. 2, pp. 349–404, 1993.

[2] T. Beelen and P. Van Dooren, "An Improved Algorithm for the Computation of Kronecker's Canonical Form of a Singular Pencil", *Linear Algebra and Its Applications*, Vol. 105, pp. 9–65, 1988.

[3] S. P. Bhattacharyya, "Frequency Domain Conditions for Disturbance Rejection", *IEEE Transactions on Automatic Control*, Vol. 25, No. 6, pp. 1211–1213, 1980.

[4] F. M. Callier and C. A. Desoer, "Linear System Theory", *Springer Verlag*, Berlin, 1982.

[5] F. M. Callier, "On Polynomial Matrix Spectral Factorization by Symmetric Extraction", *IEEE Transactions on Automatic Control*, Vol. 30, No. 5, pp. 453–464, 1985.

[6] J. Descusse and J. M. Dion, "On the Structure at Infinity of Linear Square Decoupled Systems", *IEEE Transactions on Automatic Control*, Vol. 27, No. 4, pp. 971–974, 1982.

[7] F. R. Gantmacher, "Matrix Theory", *Chelsea Publishing*, New York, 1959.

[8] G. H. Golub and C. F. Van Loan, "Matrix Computations - Third Edition", *The Johns Hopkins University Press*, New York, 1996.

[9] D. Henrion, "Reliable Algorithms for Polynomial Matrices", *Ph. D. Thesis*, Institute of Information Theory and Automation, Czech Academy of Sciences, Prague, Czech Republic, 1998.

[10] D. Henrion and M. Šebek, "Reliable Numerical Methods for Polynomial Matrix Triangularization", *IEEE Transactions on Automatic Control*, Vol. 44, No. 3, pp. 497–508, 1999.

[11] D. Henrion and M. Šebek, "Improved Polynomial Matrix Determinant Computation", *IEEE Transactions on Circuits and Systems*, Part I, Vol. 46, No. 10, pp. 1307–1308, 1999.

[12] D. Henrion and M. Šebek, "An Algorithm for Polynomial Matrix Factor Extraction", to appear in *International Journal of Control*, 2000.

[13] M. Hromčík and M. Šebek, "New Algorithms for Polynomial Matrices Based on FFT", *Research Report*, Institute of Information Theory and Automation, Czech Academy of Sciences, Prague, Czech Republic, 1998. Submitted for publication.

[14] T. Kailath, "Linear Systems", *Prentice Hall*, Englewood Cliffs, 1980.

[15] V. Kučera, "Discrete Linear Control: The Polynomial Equation Approach", *John Wiley and Sons*, Chichester, 1979.

[16] H. Kwakernaak and M. Šebek, "Polynomial J-Spectral Factorization", *IEEE Transactions on Automatic Control*, Vol. 39, No. 2, pp. 315–328, 1994.

[17] M. Malabre and V. Kučera, "Infinite Structure and Exact Model Matching Problem: a Geometric Approach", *IEEE Transactions on Automatic Control*, Vol. 29, No. 3, pp. 266–268, 1984.

[18] S. Mondié, P. Zagalak and V. Kučera, "On Some Problems of Linear Control Theory Involving State Feedback", *Proceedings of the European Control Conference*, Vol. G, Paper No. F-206, Karlsruhe, Germany, 1999.

[19] Polyx, Ltd. "The POLYNOMIAL TOOLBOX for MATLAB", 2000. See the web page `www.polyx.com` or `www.polyx.cz`.

[20] M. Šebek, F. Kraus, S. Pejchová, H. Kwakernaak and D. Henrion, "Numerical Methods for Zeros and Determinant of Polynomial Matrix", *Proceedings of the IEEE Mediterranean Symposium on New Directions in Control and Automation*, pp. 488-491, Chania, Crete, Greece, 1996.

[21] M. Šebek, H. Kwakernaak, D. Henrion and S. Pejchová "New Algorithms for Polynomial Matrices used in the Polynomial Toolbox 2.0 for MATLAB", *Proceedings of the IEEE Conference on Decision and Control*, pp. 3661–3668, Tampa, Florida, 1998.

[22] P. Van Dooren, P. Dewilde and J. Vandewalle "On the Determination of the Smith-MacMillan Form of a Rational Matrix from its Laurent Expansion", *IEEE Transactions on Circuits and Systems*, Vol. 26, No. 3, pp. 180–189, 1979.

[23] P. Van Dooren "The Computation of Kronecker's Canonical Form of a Singular Pencil", *Linear Algebra and Its Applications*, Vol. 27, pp. 103–141, 1979.

[24] P. Van Dooren and P. Dewilde "The Eigenstructure of an Arbitrary Polynomial Matrix. Computational Aspects", *Linear Algebra and Its Applications*, Vol. 50, pp. 545–580, 1983.

[25] A. I. G. Vardulakis, "Linear Multivariable Control. Algebraic Analysis and Synthesis Methods", *Wiley*, Chichester, 1991.

[26] Working Group on Software, "Control and Systems Library SLICOT", 2000. See the web page `www.win.tue.nl/wgs`.

[27] P. Zagalak and V. Kučera, "Eigenstructure Assignment in Linear Descriptor Systems", *IEEE Transactions on Automatic Control*, Vol. 40, No. 1, pp. 144–147, 1995.