An Algorithm for Polynomial Matrix Factor Extraction

Didier Henrion *
Michael Šebek †‡

Abstract

An algorithm is described for extracting a polynomial matrix factor featuring any subset of the zeros of a given non-singular polynomial matrix. It is assumed that the zeros to be extracted are given as input data. Complex or repeated zeros are allowed. The algorithm is based on interpolation and relies upon numerically reliable subroutines only. It makes use of a procedure that computes the generalized characteristic vectors of a polynomial matrix at a given point. The extracted factor is provided in column- and row-reduced Popov form. Applications of the algorithm include polynomial matrix interpolation, plus/minus factorization, column- and row-reduction, or computation of the Smith form of a polynomial matrix. The numerical routines described in this paper are implemented in the new release 2.0 of the Polynomial Toolbox for MATLAB.

1 Introduction

Let $A(s)$ be a given non-singular polynomial matrix. Let $\mathcal{Z}$ be a given subset of the zeros of $A(s)$. The aim of this paper is to provide a numerical method for extracting from $A(s)$ a polynomial matrix right factor $R(s)$ such that $\mathcal{Z}$ is the set of zeros of $R(s)$. In other words, matrix $R(s)$ is such that there exists a polynomial matrix left factor $L(s)$ that satisfies

$$A(s) = L(s)R(s). \tag{1}$$

Our main assumption is that the zeros in $\mathcal{Z}$ are given as input data.

*Corresponding author. LAAS-CNRS, 7 Avenue du Colonel Roche, 31 077 Toulouse, Cedex 4, France. E-mail: henrion@laas.fr. FAX: (33 5) 61 33 69 69.

†Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Pod vodárenskou věží 4, 182 08 Prague 8, Czech Republic.

‡Trnka Laboratory of Automatic Control, Faculty of Electrical Engineering, Czech University of Technology, Technická 2, 166 27 Prague 6, Czech Republic

Our motivation for studying the factor extraction problem originally stems from standard polynomial matrix plus/minus factorization or more general good/bad factorizations, where the set of zeros of polynomial matrix $A(s)$ must be split into two subsets, namely the zeros of right factor $R(s)$ and the zeros of left factor $L(s)$. The zeros of $R(s)$ are expected to satisfy, say, some good properties - such as location in the left half-plane or in a given disc of the complex plane - whereas zeros of $L(s)$ do not satisfy these properties. Such factorizations arise when solving standard control problems with the polynomial approach, see e.g. [8, 9, 10]. However, as will be shown, the results presented in this paper fall far beyond the sole scope of polynomial matrix factorization.

Besides our main contribution, which is an algorithm for performing factor extraction, we also provide a numerical method for computing the generalized characteristic vectors of a polynomial matrix at a given point. Applications arise in polynomial matrix interpolation, a powerful technique frequently used when dealing with polynomial matrix equations found in linear systems theory [1]. As a byproduct, we also end up with a technique that provides the invariant polynomials, hence the Smith form of a polynomial matrix [9]. Finally, we show that our algorithm returns a factor in a special column- and row-reduced polynomial echelon form, or Popov form [7, 3]. As a result, the algorithm may also be used for performing column- and row-reduction by column operations, a transformation frequently required when working with matrix fraction descriptions of linear systems [7, 3].

The outline of the paper is as follows. In Section 2, we review some standard techniques of numerical linear algebra that will be used throughout the manuscript. We also define basic concepts such as algebraic and geometric multiplicities of repeated zeros and generalized characteristic vectors of polynomial matrices. In Section 3, a subroutine is described for computing the generalized characteristic vectors of a polynomial matrix at a given point. This subroutine will be instrumental to the design, in Section 4, of an algorithm for polynomial matrix factor extraction. Illustrating numerical examples are collected in Section 5 and we conclude the paper with some important remarks.

The numerical routines described in this paper are implemented in the new release 2.0 of the Polynomial Toolbox for MATLAB[1] presented in [12].

2 Preliminaries

We describe two standard techniques of numerical algebra that are the backbone of the routines developed in this paper: reduction of a constant matrix into Column Echelon Form (CEF) by stable Householder transformations and extraction of a basis in Row Echelon Form (REF) for the left null-space of a constant matrix by stable back substitutions. These routines are covered in considerable detail in reference textbooks [13, 5]. They are briefly described here for the sake of clarity.

Some standard facts about the characteristic structure of a polynomial matrix - repeated zeros and Generalized Characteristic Vectors (GCVs) - are also recalled. A comprehensive

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

treatment can be found in [1].

2.1 Column echelon form

Reduction into column echelon form will prove useful when searching for the linearly independent rows of a constant matrix. It is based on successive column Householder transformations. For notational simplicity, the algorithm is described below in the special case of a 5×5 matrix of rank 3.

Suppose we are given a constant (real or complex) matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix}.$$

A first column Householder transformation is used to cancel in the first row all the entries but the first one. In matrix notation, the transformation reads

$$\overline{A} = AH_1 = \begin{bmatrix} \bar{a}_{11} & 0 & 0 & 0 & 0 \\ \bar{a}_{21} & \bar{a}_{22} & \bar{a}_{23} & \bar{a}_{24} & \bar{a}_{25} \\ \bar{a}_{31} & \bar{a}_{32} & \bar{a}_{33} & \bar{a}_{34} & \bar{a}_{35} \\ \bar{a}_{41} & \bar{a}_{42} & \bar{a}_{43} & \bar{a}_{44} & \bar{a}_{45} \\ \bar{a}_{51} & \bar{a}_{52} & \bar{a}_{53} & \bar{a}_{54} & \bar{a}_{55} \end{bmatrix}.$$

Matrix H_1 is a unitary column Householder transformation matrix. By applying the same operation on the sub-matrix of $\overline{A}$ obtained by removing the 1st row and 1st column, we cancel in the second row of $\overline{A}$ all the entries but the first and second ones. In matrix notation, the transformation reads

$$\overline{\overline{A}} = AH_1H_2 = \begin{bmatrix} \bar{\bar{a}}_{11} & 0 & 0 & 0 & 0 \\ \bar{\bar{a}}_{21} & \bar{\bar{a}}_{22} & 0 & 0 & 0 \\ \bar{\bar{a}}_{31} & \bar{\bar{a}}_{32} & \bar{\bar{a}}_{33} & \bar{\bar{a}}_{34} & \bar{\bar{a}}_{35} \\ \bar{\bar{a}}_{41} & \bar{\bar{a}}_{42} & \bar{\bar{a}}_{43} & \bar{\bar{a}}_{44} & \bar{\bar{a}}_{45} \\ \bar{\bar{a}}_{51} & \bar{\bar{a}}_{52} & \bar{\bar{a}}_{53} & \bar{\bar{a}}_{54} & \bar{\bar{a}}_{55} \end{bmatrix}.$$

Matrix H_2 is also a unitary Householder matrix. Repeating this procedure, we finally come up with a matrix, say,

$$\tilde{A} = A\underbrace{H_1H_2\cdots}_{H} = \begin{bmatrix} \tilde{a}_{11} & 0 & 0 & 0 & 0 \\ \tilde{a}_{21} & \tilde{a}_{22} & 0 & 0 & 0 \\ \tilde{a}_{31} & \tilde{a}_{32} & 0 & 0 & 0 \\ \tilde{a}_{41} & \tilde{a}_{42} & \tilde{a}_{43} & 0 & 0 \\ \tilde{a}_{51} & \tilde{a}_{52} & \tilde{a}_{53} & 0 & 0 \end{bmatrix}$$

transformed into CEF with a unitary transformation matrix H. As can be seen, the 1st, 2nd and 4th rows of $\tilde{A}$ are linearly independent of the previous rows. Since only columns transformations were used to transform A into $\tilde{A}$, the same dependence relations hold for the rows of A. Therefore, this algorithm can be used to detect linearly independent rows in a matrix. In matrix $\tilde{A}$, the 3rd row is linearly dependent of the 1st and 2nd rows and the 5th row is linearly dependent of the 1st, 2nd and 4th rows. Such linearly dependent rows are referred to as linearly echelon dependent rows in the sequel.

It can be shown that the overall reduction procedure is numerically stable [5]. The synopsis of the procedure is given below for later reference. Details of the algorithm are not provided here for conciseness. The interested reader may consult [13, 5].

Algorithm CEF **: Reduction into column echelon form**

Input	A matrix A with real or complex entries.
Output	A matrix $\tilde{A} = AH$ in column echelon form. Matrix H is unitary.

2.2 Null-space basis in row echelon form

Once a matrix is in CEF, one can easily find coefficients of combination for its linearly echelon dependent rows. As will be shown, this is actually equivalent to finding a basis for its left null-space. The procedure used in this paper, referred to as back substitution, provides a null-space basis in Row Echelon Form (REF). It is now described in the case of the matrix $\tilde{A}$ given above.

Recall the linearly echelon dependent rows in matrix $\tilde{A}$ are the 3rd and 5th rows. To find coefficients of linear combination for the 3rd row in $\tilde{A}$, we solve

$$\begin{bmatrix} k_{31} & k_{32} & 1 & 0 & 0 \end{bmatrix} \tilde{A} = 0$$

or, equivalently, the triangular linear system of equations

$$\begin{bmatrix} k_{31} & k_{32} \end{bmatrix} \begin{bmatrix} \tilde{a}_{11} & 0 \\ \tilde{a}_{21} & \tilde{a}_{22} \end{bmatrix} = -\begin{bmatrix} \tilde{a}_{31} & \tilde{a}_{32} \end{bmatrix}.$$

To find coefficients of linear combination for the 5th row, we solve

$$\begin{bmatrix} k_{51} & k_{52} & 0 & k_{54} & 1 \end{bmatrix} \tilde{A} = 0$$

or, equivalently, the triangular linear system of equations

$$\begin{bmatrix} k_{51} & k_{52} & k_{54} \end{bmatrix} \begin{bmatrix} \tilde{a}_{11} & 0 & 0 \\ \tilde{a}_{21} & \tilde{a}_{22} & 0 \\ \tilde{a}_{41} & \tilde{a}_{42} & \tilde{a}_{43} \end{bmatrix} = -\begin{bmatrix} \tilde{a}_{51} & \tilde{a}_{52} & \tilde{a}_{53} \end{bmatrix}.$$

Then we get a basis in REF for the left null-space of $\tilde{A}$. It reads

$$K = \begin{bmatrix} k_{31} & k_{32} & 1 & 0 & 0 \\ k_{51} & k_{52} & 0 & k_{54} & 1 \end{bmatrix}.$$

It is such that

$$K\tilde{A} = 0.$$

Entries equal to one in matrix K are called leading entries.

It can be shown that the above procedure is numerically stable [5]. The synopsis of the procedure is given below for later reference. A complete description of the algorithm can be found in [13, 5].

Algorithm NULLREF **: Left null-space basis in row echelon form**

Input	A matrix $\tilde{A}$ in column echelon form.
Output	A matrix K in row echelon form such that $K\tilde{A} = 0$.

2.3 Characteristic values and vectors

Throughout the paper, $A(s)$ stands for a $n \times n$ non-singular polynomial matrix with real or complex coefficients.

Definition 1 *A characteristic value, or zero, of polynomial matrix $A(s)$ is a complex number z such that there exists a non-zero complex vector v satisfying*

$$A(z)v = 0.$$

Vector v is referred to as a characteristic vector associated to zero z.

Clearly, zeros of $A(s)$ are zeros of polynomial $\det A(s)$. Since a polynomial may have repeated zeros, we need the following concepts to be defined.

Definition 2 *If z is a zero of $\det A(s)$ repeated m times, then the quantity m is the algebraic multiplicity of z and the quantity $p = n - \operatorname{rank} A(z)$ is the geometric multiplicity of z.*

In the sequel, let

$$\mathbf{A}_k = \frac{1}{k!}\left[\frac{d^k A(s)}{ds^k}\right]_{s=z}$$

denote the kth derivative of polynomial matrix $A(s)$ evaluated at a given complex number $s = z$ and divided by factorial k. In order to capture the structure of a non-singular polynomial matrix featuring repeated zeros, we recall the following fundamental result [1, Theorem A.1].

Theorem 1 *If z is a zero of $A(s)$ with algebraic multiplicity m and geometric multiplicity p then there exist a series of integers $k_i > 0$ for $i = 1, 2, \ldots, p$ such that $m = k_1 + k_2 + \cdots + k_p$ and a series of non-zero vectors $v_{i1}, v_{i2}, \ldots v_{ik_i}$ for $i = 1, 2, \ldots, p$ such that*

$$\begin{bmatrix} \mathbf{A}_0 & & & 0 \\ \mathbf{A}_1 & \mathbf{A}_0 & & \\ \vdots & & \ddots & \\ \mathbf{A}_{k_i-1} & \cdots & \mathbf{A}_1 & \mathbf{A}_0 \end{bmatrix} \begin{bmatrix} v_{i1} \\ v_{i2} \\ \vdots \\ v_{ik_i} \end{bmatrix} = 0$$

with v_{11}, v_{21}, $\ldots v_{p1}$ linearly independent. Vectors v_{ij} are generalized characteristic vectors (GCVs) of $A(s)$ at z. Integer k_i is the length of the ith chain of GCVs.

It is instructive to point out that zeros and GCVs of pencil matrix $A(s) = sI - A_0$ are respectively known as eigenvalues and generalized eigenvectors of constant matrix A_0.

3 Computation of GCVs

In this section, we focus on a numerical method for computing GCVs of a polynomial matrix corresponding to a given zero.

As shown in Theorem 1, any GCVs v_{ij}, $i = 1, 2, \ldots p$, $j = 1, 2, \ldots, k_i$ associated to a zero z belong to the null-space of a Toeplitz matrix built from constant matrices $\mathbf{A}_k$. In the sequel, we will assume for notational simplicity and without loss of generality that integers k_i are such that

$$k_1 \leq k_2 \leq \cdots \leq k_p.$$

With these notations, it holds

$$\underbrace{\begin{bmatrix} v'_{1k_1} & \cdots & v'_{11} & & & 0 \\ v'_{2k_2} & & \cdots & v'_{21} & & \\ \vdots & & & & \ddots & \\ v'_{pk_p} & & & & \cdots & v'_{p1} \end{bmatrix}}_{\mathbf{V}} \underbrace{\begin{bmatrix} \mathbf{A}'_0 & & & 0 \\ \mathbf{A}'_1 & \mathbf{A}'_0 & & 0 \\ \vdots & & \ddots & \\ \mathbf{A}'_{k_p-1} & \cdots & \mathbf{A}'_1 & \mathbf{A}'_0 \end{bmatrix}}_{\mathbf{T}_{k_p-1}} = 0. \qquad (2)$$

In view of the above matrix equation, computing GCVs amounts to extracting some specific vectors from the left null-space of Toeplitz matrix $\mathbf{T}_{k_p-1}$. In the sequel, we show that Algorithms CEF and NullREF are particularly suited for this purpose.

Let d denote the degree of polynomial matrix $A(s)$. Let

$$\mathbf{T}_k^i = \begin{bmatrix} \mathbf{A}'_i & \cdots & \mathbf{A}'_1 & \mathbf{A}'_0 & 0 & \cdots & 0 \end{bmatrix}$$

denote the $(i+1)$th block row in matrix $\mathbf{T}_k$, $i = 0, 1, \ldots, k$ and let $\mathcal{S}_i$ denote the set of indices of linearly echelon dependent rows in $\mathbf{T}_k^i$. Similarly, let $r_i = \operatorname{card} \mathcal{S}_i$ denote the number of linearly echelon dependent rows in $\mathbf{T}_k^i$. Since the algebraic multiplicity of a zero cannot exceed nd, it holds

$$r_{nd} = 0. \qquad (3)$$

Moreover, the Toeplitz structure of $\mathbf{T}_k$ implies that

$$\mathcal{S}_0 \supseteq \mathcal{S}_1 \supseteq \cdots \supseteq \mathcal{S}_{nd-1} \supseteq \mathcal{S}_{nd} = \mathcal{S}_{nd+1} = \cdots = \emptyset$$

and

$$r_0 \geq r_1 \geq \cdots \geq r_{nd-1} \geq r_{nd} = r_{nd+1} = \cdots = 0. \qquad (4)$$

Our approach for computing GCVs is based on relation (4) and successive extractions of null-spaces of Toeplitz matrices $\mathbf{T}_0$, $\mathbf{T}_1$, $\ldots$ $\mathbf{T}_{nd}$. For the sake of clarity, we will expose our reasoning in the special case that $n = 4$.

Assume that, upon application of Algorithms CEF and NULLREF, we come up with a left null-space basis in REF for Toeplitz matrix $\mathbf{T}_0$ that reads, say,

$$\underbrace{\begin{bmatrix} \times & \times & 1 & . \\ \times & \times & . & 1 \end{bmatrix}}_{\mathbf{V}_0} \underbrace{\mathbf{A}_0'}_{\mathbf{T}_0} = 0$$

where a cross denotes a non-zero entry and a dot denotes a zero entry. Consequently, linearly echelon dependent rows in $\mathbf{T}_0$ are the 3rd and 4th rows, hence $\mathcal{S}_0 = \{3, 4\}$ and $r_0 = 2$.

Then consider the extraction of the left null-space in REF for Toeplitz matrix $\mathbf{T}_1$. Suppose we get, for instance,

$$\underbrace{\left[\begin{array}{cccc|cccc} \times & \times & 1 & . & . & . & . & . \\ \times & \times & . & 1 & . & . & . & . \\ \hline \times & \times & . & . & \times & \times & 1 & . \end{array}\right]}_{\mathbf{V}_1} \underbrace{\left[\begin{array}{cc} \mathbf{A}_0' & 0 \\ \hline \mathbf{A}_1' & \mathbf{A}_0' \end{array}\right]}_{\mathbf{T}_1 = \left[\begin{array}{c} \mathbf{T}_1^0 \\ \hline \mathbf{T}_1^1 \end{array}\right]} = 0.$$

Observe that $\mathcal{S}_1 = \{3\} \subset \mathcal{S}_0$ and $r_1 = 1 < r_0$. The 4th row, linearly dependent in $\mathbf{T}_1^0$, becomes linearly independent in $\mathbf{T}_1^1$. Coefficients of dependence of this row with respect to the previous ones are displayed in the 2nd row of matrix $\mathbf{V}_1$, or, equivalently, in the 2nd row of matrix $\mathbf{V}_0$. Therefore, they correspond to a first GCV chain that consists of only one vector v_{11} such that $k_1 = 1$ and

$$\begin{bmatrix} v_{11}' & 0 & \cdots & 0 \end{bmatrix} \mathbf{T}_k = 0$$

for any $k \geq k_1 - 1 = 0$.

Now assume that the left null-space of $\mathbf{T}_2$ reads, say,

$$\underbrace{\left[\begin{array}{cccc|cccc|cccc} \times & \times & 1 & . & . & . & . & . & . & . & . & . \\ \times & \times & . & 1 & . & . & . & . & . & . & . & . \\ \hline \times & \times & . & . & \times & \times & 1 & . & . & . & . & . \end{array}\right]}_{\mathbf{V}_2} \underbrace{\left[\begin{array}{ccc} \mathbf{A}_0' & 0 & 0 \\ \hline \mathbf{A}_1' & \mathbf{A}_0' & 0 \\ \hline \mathbf{A}_2' & \mathbf{A}_1' & \mathbf{A}_0' \end{array}\right]}_{\mathbf{T}_2 = \left[\begin{array}{c} \mathbf{T}_2^0 \\ \hline \mathbf{T}_2^1 \\ \hline \mathbf{T}_2^2 \end{array}\right]} = 0$$

which means that $\mathcal{S}_2 = \emptyset$ and $r_2 = 0$. The 3rd row, linearly dependent in $\mathbf{T}_2^1$, becomes linearly independent in $\mathbf{T}_2^2$. Coefficient of dependence of this row with respect to the previous ones are displayed in the 3rd row of matrix $\mathbf{V}_2$, or, equivalently, in the 3rd row of matrix $\mathbf{V}_1$. They correspond to a second GCV chain that consists of two vectors v_{21}, v_{22} such that $k_2 = 2$ and

$$\begin{bmatrix} v_{22}' & v_{21}' & 0 & \cdots & 0 \end{bmatrix} \mathbf{T}_k = 0$$

for any $k \geq k_2 - 1 = 1$.

Recalling relation (4), no further null-space extraction is required since $r_k = 0$ for all $k \geq 2$. As a result, we have entirely described the set of GCVs. In our example, there are $p = 2$ chains corresponding to a zero with algebraic multiplicity $m = k_1 + k_2 = 3$. A generic algorithm based on the above ideas can be described in the following terms.

Algorithm GCV **: Computation of generalized characteristic vectors at a given point**

Input	A non-singular $n \times n$ polynomial matrix $A(s)$ and a complex number z.
Output	The generalized characteristic vectors of $A(s)$ at z.
Step 1	Let $k = 0$ and $t_1 = t_2 = \cdots = t_n = 0$.
Step 2	Build Toeplitz matrix $\mathbf{T}_k$ as in equation (2).
Step 3	With Algorithm CEF, compute the set $\mathcal{S}_k = \{s_1, s_2, \ldots, s_{r_k}\}$ of indices of the r_k linearly echelon dependent rows in the $(k+1)$th block row in matrix $\mathbf{T}_k$. Let $t_{s_i} = r_{k-1} + i$ for $i = 1, 2, \ldots, r_k$.
Step 4	If $\mathcal{S}_k$ is non-empty, let $k = k + 1$ and go to Step 2. Otherwise, let $k = k - 1$.
Step 5	With Algorithm NullREF, extract a basis $\mathbf{V}_k$ in REF for the left null-space of Toeplitz matrix $\mathbf{T}_k$.
Step 6	Matrix $\mathbf{V}$ in equation (2) and the required GCVs are obtained by keeping the rows of $\mathbf{V}_k$ indexed by non-zero elements $t_1, t_2, \ldots, t_n$. If a vector v_{ij} in $\mathbf{V}$ is zero for some $j > 1$, then it can always be made non-zero by suitable combination of rows in matrix $\mathbf{V}$.

By definition [7, 9], the Smith normal form of a given polynomial matrix $A(s)$ contains factors $(s-z)^{k_1}, (s-z)^{k_2}, \ldots, (s-z)^{k_p}$ in p different locations on the diagonal, provided p is the geometric multiplicity and $m = k_1 + k_2 + \ldots + k_p$ is the algebraic multiplicity of a zero z of $A(s)$. If we are given the set of zeros of $A(s)$, the Smith form of $A(s)$ therefore readily follows from the knowledge of integers $p, k_1, k_2, \ldots, k_p$ associated with each zero of $A(s)$. Consequently, one can derive the Smith form of $A(s)$ by calling Algorithm GCV for each distinct zero of $A(s)$. This is captured as follows.

Algorithm SmithForm **: Computation of the Smith form**

Input	A non-singular $n \times n$ polynomial matrix $A(s)$ and the set $\mathcal{Z} = \{z_1, z_2, \ldots, z_q\}$ of all its zeros.
Output	The Smith form $S(s)$ of $A(s)$.
Step 1	Let $\varepsilon_1(s) = \varepsilon_2(s) = \cdots = \varepsilon_n(s) = 1$.
Step 2	For $i = 1, 2, \ldots, q$ compute with Algorithm GCV the algebraic multiplicity p_i and the geometric multiplicity $m_i = k_{i1} + k_{i2} + \ldots + k_{ip_i}$ of the zero z_i of $A(s)$.
Step 3	Let $\varepsilon_{n-j+1}(s) = (s - z_i)^{k_{ij}} \varepsilon_{n-j+1}(s)$ for $j = 1, 2, \ldots, p_i$.
Step 4	Let $S(s) = \text{diag}\,\{\varepsilon_1(s), \varepsilon_2(s), \ldots \varepsilon_n(s)\}$.

Upon completing the first version of this manuscript, we realized that a similar GCV computation method was independently developed by Dahleh and Diaz-Bobillo in Chapter

6 of their textbook [4]. In this reference, GCVs are shown to be instrumental to the formulation of interpolation conditions, in turn used for characterizing all achievable maps of a closed-loop dynamical system. This leads to the formulation of a general controller synthesis problem including $\mathcal{H}_2$, $\mathcal{H}_\infty$ and l_1 specifications. Algorithm GCV as described above is essentially the same as the one proposed in [4], with the notable exception that we focus more on computational aspects.

4 Factor Extraction

In order to solve the factorization problem stated in the introduction, we now characterize polynomial matrix right factor $R(s)$ through GCVs of polynomial matrix $A(s)$.

If $A(s) = L(s)R(s)$ then GCVs of $R(s)$ corresponding to zeros in $\mathcal{Z}$ are also GCVs of $A(s)$, see [1, Theorem A.7]. Let

$$\mathbf{R}_k = \frac{1}{k!}\left[\frac{d^k R(s)}{ds^k}\right]_{s=z}$$

stand for the kth derivative of polynomial matrix $R(s)$ evaluated at $s = z$ and divided by factorial k. From Theorem 1, it holds

$$\begin{bmatrix} \mathbf{R}_0 & & & 0 \\ \mathbf{R}_1 & \mathbf{R}_0 & & \\ \vdots & & \ddots & \\ \mathbf{R}_{k_i-1} & \cdots & \mathbf{R}_1 & \mathbf{R}_0 \end{bmatrix} \begin{bmatrix} v_{i1} \\ v_{i2} \\ \vdots \\ v_{ik_i} \end{bmatrix} = 0, \quad i = 1, 2, \ldots, p. \tag{5}$$

Let $S(s) = \begin{bmatrix} 1 & s & \cdots & s^d \end{bmatrix}'$ and suppose $R(s)$ has degree d, i.e.

$$R(s) = R_0 + R_1 s + \cdots + R_d s^d = \underbrace{\begin{bmatrix} R_0 & R_1 & \cdots & R_d \end{bmatrix}}_{R} \underbrace{\begin{bmatrix} I \\ sI \\ \vdots \\ s^d I \end{bmatrix}}_{S(s)\otimes I}.$$

where $\otimes$ stands for the Kronecker product. Finally, define

$$\mathbf{S}_k = \frac{1}{k!}\left[\frac{d^k S(s)}{ds^k}\right]_{s=z} \otimes I$$

so that $\mathbf{R}_k = R\mathbf{S}_k$. With these notations, equation (5) can equivalently be written

$$RV_i = 0$$

where

$$V_i = \begin{bmatrix} \mathbf{S}_0 & \mathbf{S}_1 & \cdots & \mathbf{S}_{k_i-1} \end{bmatrix} \begin{bmatrix} v_{i1} & v_{i2} & \cdots & v_{ik_i} \\ & v_{i1} & & \\ & & \ddots & \vdots \\ 0 & & & v_{i1} \end{bmatrix} \tag{6}$$

is a $n(d+1) \times k_i$ constant matrix associated to the ith chain of GCVs corresponding to the zero z with geometric multiplicity p. By considering all the chains of GCVs associated to z for $i = 1, 2, \ldots, p$, we get

$$R \underbrace{\left[\begin{array}{cccc} V_1 & V_2 & \cdots & V_p \end{array}\right]}_{V} = 0 \tag{7}$$

where V is a $n(d+1) \times m$ constant matrix and $m = k_1 + k_2 + \cdots + k_p$ is the algebraic multiplicity of z. In the sequel, matrix V is referred to as the characteristic matrix of z. Any constant matrix R associated to a right factor $R(s)$ sharing the same zero z as polynomial matrix $A(s)$ satisfies equality (7).

The same reasoning can be applied to any zero belonging to the set $\mathcal{Z} = \{z_1, z_2, \ldots, z_q\}$ of zeros common to $A(s)$ and $R(s)$. Let W_j denote the characteristic matrix associated to z_j, so that we have

$$R \underbrace{\left[\begin{array}{cccc} W_1 & W_2 & \cdots & W_q \end{array}\right]}_{W} = 0. \tag{8}$$

In reference [1], relation (8) is referred to as an interpolation equation for polynomial matrix $R(s)$. Any factor $R(s)$ with required zeros $z_1, z_2, \ldots, z_q$ will satisfy interpolation equation (8).

Note however that no assumption has been made so far regarding the degree d of polynomial matrix $R(s)$. Its value is not known beforehand, yet its choice appears to be crucial since inner matrix dimensions in equation (8) depend on d. The same remark can be made about column degrees of $R(s)$. A trivial upper bound on column degrees is the number of zeros to be extracted, including multiplicities. We will show that, given this upper bound, Algorithms CEF and NULLREF can be applied to infer correct column degrees of $R(s)$ and therefore extract a right factor featuring interesting column- and row-reducedness properties.

For simplicity, we illustrate our reasoning in the case that $n = 2$ and $q = 2$. Suppose that algebraic multiplicities of zeros z_1, z_2 are $m_1 = 1$, $m_2 = 2$, respectively. An upper bound on column degrees of $R(s)$ is $d = 3$, so that matrix W is made of $n(d+1) = 8$ rows and $m_1 + m_2 = 3$ columns. As shown by equation (8), in order to extract polynomial matrix factor $R(s)$, we need to extract $n = 2$ rows from the left null-space of constant matrix W.

Applying Algorithms CEF and NULLREF yields a basis in REF for the left null-space of the CEF of W, say,

$$\left[\begin{array}{cc|cc|cc|cc} \times & \times & 1 & . & . & . & . & . \\ \times & \times & . & \times & 1 & . & . & . \\ \times & \times & . & \times & . & 1 & . & . \\ \times & \times & . & \times & . & . & 1 & . \\ \times & \times & . & \times & . & . & . & 1 \end{array}\right] \left[\begin{array}{c|cc} \times & . & . \\ \times & \times & . \\ \hline \times & \times & . \\ \times & \times & \times \\ \hline \times & \times & \times \\ \times & \times & \times \\ \hline \times & \times & \times \\ \times & \times & \times \end{array}\right] = 0.$$

From the 5 rows of the above null-space basis, we must select 2 rows to build constant matrix

$$R = \left[\begin{array}{cccc} R_0 & R_1 & R_2 & R_3 \end{array}\right]$$

hence the required factor $R(s) = R_0 + R_1 s + R_2 s^2 + R_3 s^3$. In the first row, the leading entry is located in the first column of the column block matrix associated to R_1. If we keep this row, the first column degree of $R(s)$ will be equal to one. In the second row, the leading entry is located in the first column of the column block matrix associated to R_2. If we keep this row, the first column degree of $R(s)$ will be equal to two. A natural choice is therefore to keep the first row instead of the second, so that the first column degree in $R(s)$ is minimized. Proceeding this way for the second column degree of $R(s)$, we keep the third row of the null-space basis. It corresponds to a minimized second column degree equal to two. Our choice leads to

$$R = \left[\begin{array}{cc|cc|cc|cc} \times & \times & 1 & . & . & . & . & . \\ \times & \times & . & \times & . & 1 & . & . \end{array}\right]$$

therefore $R(s)$ is a polynomial matrix of degree two. Its column-wise and row-wise leading coefficient matrices are the identity matrix, hence $R(s)$ is both column- and row-reduced.

More generally, the above row selection scheme ensures by construction that column degrees of right factor $R(s)$ are minimized. Because of the row echelon form of the null-space basis, it can be checked that the column-wise leading coefficient matrix of $R(s)$ is always the identity matrix. Similarly, the row-wise leading coefficient matrix of $R(s)$ is always a lower-left triangular matrix with ones along the diagonal. As a result, $R(s)$ is always column- and row-reduced. Actually, right factor $R(s)$ is in the so-called polynomial row echelon form or Popov row form defined in [7, Section 6.7.2] or [3, Appendix G-6]. Our factor extraction algorithm is now described in a more synthetic form.

Algorithm Factor **: Factor extraction**

Input A non-singular $n \times n$ polynomial matrix $A(s)$ and a set $\mathcal{Z} = \{z_1, z_2, \ldots, z_q\}$ of zeros of $A(s)$.

Output A polynomial matrix $R(s)$ in Popov form such that $\mathcal{Z}$ is the set of zeros of $R(s)$ and there exists a non-singular matrix $L(s)$ that satisfies $A(s) = L(s)R(s)$.

Step 1 For each zero z_j, $j = 1, 2, \ldots, q$ compute with Algorithm GCV the GCVs of $A(s)$. Build associated characteristic matrix V_j using relations (6) and (7). Build interpolation matrix W based on equation (8).

Step 2 With Algorithms CEF and NullREF, extract a basis in REF 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 $R = [R_0 \quad R_1 \quad R_2 \quad \cdots \quad]$ made up from these rows gives rise to the required polynomial matrix factor $R(s) = R_0 + R_1 s + R_2 s^2 + \cdots$. Column and row degrees of $R(s)$ are the integers $k_1, k_2, \ldots, k_n$.

In Step 3 of the above algorithm, the existence of an integer k_i such that a leading entry appears at column location $i + k_i n$ in the REF of the left null-space of matrix W is guaranteed by the fact that there always exists a right factor $R(s)$ of $A(s)$ that can be chosen column-reduced by proper column operations. Moreover, its degree is always less than or equal to the number of zeros to be extracted, including multiplicities.

Note finally that if $\mathcal{Z}$ is the whole set of zeros of $A(s)$, then left factor $L(s)$ does not feature any finite zeros since all of them are captured by right factor $R(s)$. Polynomial matrix $L(s)$ is therefore unimodular. Hence, Algorithm FACTOR may be used to perform column- and row-reduction of a non-singular matrix by row operations. The column- and row-reduced form of $A(s)$ reads $R(s) = L^{-1}(s)A(s)$.

5 Illustration

Consider the polynomial matrix

$$A(s) = \begin{bmatrix} 1+s & 0 & 2+s-s^2 \\ -1+s-3s^2+s^3 & -2s+3s^2-s^3 & -2-s+s^2-s^3 \\ 2s-s^2 & -2s+s^2 & s^2 \end{bmatrix}.$$

With the help of the Polynomial Toolbox for MATLAB [12], we found that its zeros are located at 0, -1 and 2.

5.1 Algorithm GCV

First, we illustrate the results of Section 3 and Algorithm GCV. We compute the GCVs of $A(s)$ at $s = 0$. Using Algorithm CEF, Toeplitz matrix $\mathbf{T}_0$ is transformed into CEF as follows

$$\mathbf{T}_0 = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 0 \\ 2 & -2 & 0 \end{bmatrix} \Longrightarrow \begin{bmatrix} -1.4142 & . & . \\ 0 & . & . \\ -2.8284 & . & . \end{bmatrix}$$

where a dot denotes a zero entry. There are $r_0 = 2$ linearly echelon dependent rows whose indices are stored into $\mathcal{S}_0 = \{2, 3\}$, thus $t_1 = 0, t_2 = 1, t_3 = 2$. Next, Toeplitz matrix $\mathbf{T}_1$ is transformed into CEF as follows

$$\mathbf{T}_1 = \left[\begin{array}{ccc|ccc} 1 & -1 & 0 & . & . & . \\ 0 & 0 & 0 & . & . & . \\ 2 & -2 & 0 & . & . & . \\ \hline 1 & 1 & 2 & 1 & -1 & 0 \\ 0 & -2 & -2 & 0 & 0 & 0 \\ 1 & -1 & 0 & 2 & -2 & 0 \end{array}\right] \Longrightarrow \left[\begin{array}{ccc|ccc} -1.4142 & . & . & . & . & . \\ 0 & . & . & . & . & . \\ -2.8284 & . & . & . & . & . \\ \hline 0 & -2.8284 & . & . & . & . \\ -1.4142 & 2.1213 & 1.2247 & . & . & . \\ -1.4142 & -1.4142 & 2.4495 & . & . & . \end{array}\right].$$

There is only $r_1 = 1$ linearly echelon dependent row in the second block row of $\mathbf{T}_1$ thus $\mathcal{S}_1 = \{3\}$ and $t_1 = 0, t_2 = 1, t_3 = 3$. Then, Toeplitz matrix $\mathbf{T}_2$ is transformed into CEF as follows

$$\mathbf{T}_2 = \left[\begin{array}{ccc|ccc|ccc} 1 & -1 & 0 & . & . & . & . & . & . \\ 0 & 0 & 0 & . & . & . & . & . & . \\ 2 & -2 & 0 & . & . & . & . & . & . \\ \hline 1 & 1 & 2 & 1 & -1 & 0 & . & . & . \\ 0 & -2 & -2 & 0 & 0 & 0 & . & . & . \\ 1 & -1 & 0 & 2 & -2 & 0 & . & . & . \\ \hline 0 & -3 & -1 & 1 & 1 & 2 & 1 & -1 & 0 \\ 0 & 3 & 1 & 0 & -2 & -2 & 0 & 0 & 0 \\ -1 & 1 & 1 & 1 & -1 & 0 & 2 & -2 & 0 \end{array}\right]$$

$$\Longrightarrow \left[\begin{array}{ccc|ccc|ccc} -1.4142 & . & . & . & . & . & . & . & . \\ 0 & . & . & . & . & . & . & . & . \\ -2.8284 & . & . & . & . & . & . & . & . \\ \hline 0 & -2.8284 & . & . & . & . & . & . & . \\ -1.4142 & 2.1213 & 1.2247 & . & . & . & . & . & . \\ -1.4142 & -1.4142 & 2.4495 & . & . & . & . & . & . \\ \hline -2.1213 & 1.7678 & 1.0206 & -3.0551 & . & . & . & . & . \\ 2.1213 & -2.4749 & 0.2041 & -2.4004 & 1.2536 & . & . & . & . \\ 1.4142 & -1.4142 & 0.8165 & -1.5275 & 2.3932 & 0.5222 & . & . & . \end{array}\right].$$

There are $r_2 = 0$ linearly echelon dependent rows in the third block row of $\mathbf{T}_2$, so $\mathcal{S}_2 = \emptyset$ and there is no need to search for linearly echelon dependent rows in next Toeplitz matrix $\mathbf{T}_3$. Algorithm NULLREF returns the following basis in REF for the left null-space of $\mathbf{T}_1$:

$$\left[\begin{array}{ccc|ccc} 0 & 1 & . & . & . & . \\ -2 & . & 1 & . & . & . \\ 1 & . & . & -2 & -2 & 1 \end{array}\right] \left[\begin{array}{ccc|ccc} -1.4142 & . & . & . & . & . \\ 0 & . & . & . & . & . \\ -2.8284 & . & . & . & . & . \\ \hline 0 & -2.8284 & . & . & . & . \\ -1.4142 & 2.1213 & 1.2247 & . & . & . \\ -1.4142 & -1.4142 & 2.4495 & . & . & . \end{array}\right] = 0.$$

Matrix $\mathbf{V}$ is obtained by keeping the rows numbered $t_2 = 1$ and $t_3 = 3$ in the above null-space basis. Thus $k_1 = 1$, $k_2 = 2$ and the GCVs at $s = 0$ read

$$v_{11} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad v_{21} = \begin{bmatrix} -2 \\ -2 \\ 1 \end{bmatrix}, \quad v_{22} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

Zero $s = 0$ has geometric multiplicity $p = 2$ and algebraic multiplicity $m = k_1 + k_2 = 3$.

5.2 Algorithm FACTOR

Second, we illustrate the results of Section 4 and Algorithm FACTOR. We compute a right factor $R(s)$ in Popov form such that $R(s)$ features a triple zero at $s = 0$. Using the GCVs obtained above, a basis in REF for the left null-space of characteristic matrix W reads

$$\underbrace{\left[\begin{array}{ccc|ccc} 2 & 0 & 4 & 1 & . & . \\ 2 & 0 & 4 & . & 1 & . \\ -1 & 0 & -2 & . & . & 1 \end{array}\right]}_{R} \underbrace{\left[\begin{array}{c|cc} 0 & -2 & 1 \\ 1 & -2 & 0 \\ 0 & 1 & 0 \\ \hline . & . & -2 \\ . & . & -2 \\ . & . & 1 \end{array}\right]}_{W} = 0.$$

The row selection scheme in R is straightforward and yields

$$R(s) = \begin{bmatrix} 2+s & 0 & 4 \\ 2 & s & 4 \\ -1 & 0 & -2+s \end{bmatrix}.$$

Using the Polynomial Toolbox equation solver, we recover left factor

$$L(s) = \begin{bmatrix} 0 & 0 & -1-s \\ 2-4s+s^2 & -2+3s-s^2 & 1-s-s^2 \\ 3-s & -2+s & 2+s \end{bmatrix}$$

such that $A(s) = L(s)R(s)$. It can be checked that $R(s)$ has a triple zero at $s = 0$ whereas $L(s)$ has two zeros at -1 and 2. Moreover, $R(s)$ is both column- and row-reduced.

5.3 Reduction into Popov form

Third, we show how Algorithm Factor can be used to simultaneously perform column- and row-reduction on $A(s)$. As pointed out previously, we just have to extract a factor $R(s)$ featuring all the zeros of $A(s)$, namely 0, -1 and 2. Using Algorithm GCV, we compute the GCVs associated to -1 and 2. They are respectively

$$v_3 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad v_4 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

A basis in REF for the left null-space of characteristic matrix W reads

$$\underbrace{\left[\begin{array}{ccc|ccc|ccc|c} 1 & 0 & 2 & 1 & 0 & 1 & . & . & . & \\ -2 & 0 & -4 & -1 & 0 & . & 1 & . & . & \\ -2 & 0 & -4 & 1 & -2 & . & . & 1 & . & \cdots \\ 0 & 0 & 0 & 0 & 0 & . & . & . & 1 & \\ & \cdots & & & \cdots & & & \cdots & & \end{array}\right]}_{R} \underbrace{\left[\begin{array}{c|cc|c|c} 0 & -2 & 1 & 1 & 0 \\ 1 & -2 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ \hline . & . & -2 & -1 & 0 \\ . & . & -2 & -1 & 2 \\ . & . & 1 & 0 & 0 \\ \hline . & . & . & 1 & 0 \\ . & . & . & 1 & 4 \\ . & . & . & 0 & 0 \\ \hline & & \cdots & & \end{array}\right]}_{W} = 0.$$

Using our row selection scheme, minimum column degrees for $R(s)$ are evaluated to $k_1 = 2$, $k_2 = 2$ and $k_3 = 1$. The first three rows in the above null-space basis are kept to yield

$$R(s) = \begin{bmatrix} -2-s+s^2 & 0 & -4 \\ -2+s & -2s+s^2 & -4 \\ 1+s & 0 & 2+s \end{bmatrix}.$$

Corresponding left factor reads

$$L(s) = \begin{bmatrix} 1 & 0 & 3-s \\ -2+2s & 1-s & -3+3s-s^2 \\ -2 & 1 & -2+s \end{bmatrix}.$$

One can check that $\det L(s) = 1$ therefore $L(s)$ is a unimodular matrix. As a result

$$R(s) = L^{-1}(s)A(s)$$

is a column- and row-reduced Popov form of $A(s)$. The unimodular reduction matrix

$$L^{-1}(s) = \begin{bmatrix} 1 & 3-s & -3+4s-s^2 \\ 2 & 4-s & -3+5s-s^2 \\ 0 & -1 & 1-s \end{bmatrix}$$

is readily retrieved with the Polynomial Toolbox.

5.4 Algorithm Smith

Finally, let us point out that, in view of the above results, Algorithm Smith returns

$$S(s) = \operatorname{diag}\{1,\ s,\ s^2(s+1)(s-2)\}$$

as the Smith form of $A(s)$.

6 Conclusion

An algorithm was described for performing polynomial matrix factor extraction. We assumed that the zeros of the polynomial matrix to be extracted are given as input data. Immediate byproducts of our algorithm are numerical routines for computing generalized characteristic vectors at a given point, the column- and row-reduced Popov form, and the Smith form of a polynomial matrix. Some important remarks are in order.

If the main assumption of this paper is relaxed, i.e. if the zeros to be extracted have not been computed at a previous stage, then an alternative approach to factorization can still be pursued. It is described in [11] in the special case of symmetric factor extraction for spectral factorization. Based on the Schur or QZ decomposition of a matrix pencil, this method computes the zeros of the polynomial matrix to be factorized together with invariant subspaces spanned by its characteristic vectors. By suitable ordering, a basis for the invariant subspace corresponding to the zeros to be extracted can be used, as in Section 4, so as to interpolate the required polynomial matrix factor. However, when pursuing this later approach, the column degrees of the extracted factor must be known in advance. For instance, when used for polynomial J-spectral factorization, this necessarily involves diagonal reduction of the input matrix as a preprocessing step [11]. This is in sharp contrast with the method described in this paper, where column reducedness of the input matrix is not required. Indeed, our factorization algorithm is designed to overcome the problem of degree predictability by simultaneously computing column degrees and factor coefficients.

As an additional advantage, our algorithm can also be used to successively extract first-order factors corresponding to multiple roots. This only requires the computation of a single characteristic vector at each step. If the associated root is multiple, then its remaining copies are simply extracted at the next extraction step.

Finally, it must be underlined that all the numerical methods described in this paper are based on numerically stable subroutines. Nevertheless, we do not think that the factorization algorithm as a whole can be considered as numerically stable. The main reason behind this is that we assumed that the zeros to be extracted were computed beforehand. Numerical problems are thus to be expected at this early stage since it is well-known that computation of the zeros of a polynomial matrix is a particularly tough numerical task, especially when the zeros are repeated or close to each other. Recent progress in polynomial matrix determinant computation [6] and repeated polynomial zero computation [2] reflect that this is still an open and promising research topic.

Acknowledgments

The authors would like to thank Huibert Kwakernaak (University of Twente, The Netherlands) for helpful discussions.

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] L. Brugnano and D. Trigiante, "Polynomial Roots: The Ultimate Answer?", *Linear Algebra and Its Applications*, Vol. 225, pp. 207–220, 1995.

[3] C. T. Chen, "Linear System Theory and Design - Second Edition", *Holt, Rinehart and Winston*, New York, 1984.

[4] M. A. Dahleh and I. J. Diaz-Bobillo, "Control of Uncertain Systems – A Linear Programming Approach", *Prentice Hall*, Englewood Cliffs, New Jersey, 1995.

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

[6] 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.

[7] T. Kailath, "Linear Systems", *Prentice Hall*, Englewood Cliffs, New Jersey, 1980.

[8] V. Kučera and M. Šebek, "On Deadbeat Controllers", *IEEE Transactions on Automatic Control*, Vol. 29, No. 8, pp. 719–722, 1984.

[9] V. Kučera, "Analysis and Design of Discrete Linear Control Systems", *Prentice Hall*, London, 1991.

[10] H. Kwakernaak, "Robust Control and $\mathcal{H}_\infty$ Optimization – Tutorial Paper", *Automatica*, Vol. 29, No. 2, pp. 255–273, 1993.

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

[12] 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, December 1998. See also the Polynomial Toolbox web page at `www.polyx.cz`.

[13] J. H. Wilkinson "The Algebraic Eigenvalue Problem", *Oxford University Press*, Oxford, 1965.