

On The Block Decomposition and Spectral Factors of λ -Matrices

Belkacem Bekhiti¹, Abdelhakim Dahimene¹, Kamel Hariche¹ and George F. Fragulis²

¹Signal and System Laboratory, Electronics and Electrotechnics Institute University of Boumerdes, Algeria, IGEE Ex:(INELEC)

²Western Macedonia University of Applied Sciences, Kozani, Hellas.

March 29, 2018

Abstract

In this paper we factorize matrix polynomials into a complete set of spectral factors using a new design algorithm and we provide a complete set of block roots (solvents). The procedure is an extension of the (scalar) Horner method for the computation of the block roots of matrix polynomials. The Block-Horner method brings an iterative nature, faster convergence, nested programmable scheme, needless of any prior knowledge of the matrix polynomial. In order to avoid the initial guess method we proposed a combination of two computational procedures. First we start giving the right Block-*Q. D. (Quotient Difference)* algorithm for spectral decomposition and matrix polynomial factorization. Then the construction of new block Horner algorithm for extracting the complete set of spectral factors is given.

Keywords: Block roots; Solvents; Spectral factors; Block-Q.D. algorithm; Block-Horner's algorithm; Matrix polynomial

1 Introduction

In the early days of control and system theory, frequency domain techniques were the principal tools of analysis, modeling and design for linear systems. Dynamic systems that can be modeled by a scalar m^{th} order linear differential (difference) equation with constant coefficients are amenable to this type of analysis see [[1]] and [[2]] - other references [[3]]. In the case of a single input - single output (SISO) system the transfer function is a ratio of two scalar polynomials. The dynamic properties of the system (time response, stability, etc.) depend

on the roots of the denominator of the transfer function or in other words on the solution of the underlying homogeneous differential equation (difference equation in discrete-time systems) [[4],[5],[6]] [[7],[8],[9]]. The denominator of such systems is a scalar polynomial and its spectral characteristics depend on the location of its roots in the s-plane. Hence the factorization (root finding) of scalar polynomials is an important tool of analysis and design for linear systems[[10]] . In the case of multi input - multi output (MIMO) systems the dynamics can be modeled by high-degree coupled differential equations or l^{th} degree m^{th} order vector linear differential (difference) equations with matrix constant coefficients.

In this paper, we treat the dynamic properties of multivariable systems using the latent roots and /or the spectral factors of the corresponding matrix polynomial, following the research by a number of recent publications see [[11]], [[12]], [[13]], [[14]], [[15]], [[16]] and [[17]] .

The algebraic theory of matrix polynomials has been investigated by Gohberg et al. [[18]], Dennis et al. [[19]], Denman [[20],[21]], Shieh et al. [[22],[23]], [[25]], [[26]] and Tsai et al. [[24]]. Various computational algorithms [[19]], [[21]], [[18]], [[22]] and [[27]] are available for evaluating the solvents and spectral factors of a matrix polynomial. Recent approaches [[19],[28],[29]] are the use of the eigenvalues and eigenvectors of the block companion form matrix of the corresponding matrix polynomial (the λ -matrices) to construct the solvents of that matrix polynomial based in the use of solvents.

It is often inefficient to explicitly determine the eigenvalues and eigenvectors of a matrix, which can be ill conditioned and either non-defective or defective. On the other hand, without prior knowledge of the eigenvalues and eigenvectors of the matrix, the Newton-Raphson method [[22],[30]] has been successfully utilized for finding the solvents, as well as the block-power method (Tsai et al.) [[24]] for finding the solvents and spectral factors of a general nonsingular (monic or comonic)polynomial matrix.

The matrix polynomial must have distinct block solvents, and the convergence rate of the power method depends strongly on the ratio of the two block eigenvalues of largest magnitude [[31]]. There are numerous numerical methods for computing the block roots of matrix polynomials without any prior knowledge of the eigenvalues and eigenvectors of the matrix polynomial. The most efficient and more stable one that give the complete set of solvent at time is the *Q.D.* (quotient-difference) algorithm. The use of the *Q.D.* algorithm for such purpose has been suggested by K. Hariche [[32]].

The purpose of this paper is to illustrate the so called Block quotient-difference (*Q.D.*) algorithm and extend the (scalar) Horner method to its block form for the computation of the block roots of matrix polynomial and the determination of complete set of solvents and spectral factors of a monic polynomial, without any prior knowledge of the eigenvalues and eigenvectors of the matrix. See also Pathan and Collyer [[33]] where there is a presentation on Horner's method and its application in solving polynomial equations by determining the location of roots.

The objectives of this paper are described as follows:

- Illustration and finalization of the Block quotient-difference (*Q.D.*) algorithm for spectral decomposition and matrix polynomial factorization.
- Construction of a new block-Horner array and block-Horner algorithm for extracting the complete set of spectral factors of matrix polynomials.
- Combined the above algorithms for fast convergence, high stability and avoiding initial guess.

2 Preliminaries

In this section we give some background material.

2.1 Survey on matrix polynomials

Definition 2.1. *Given the set of $m \times m$ complex matrices A_0, A_1, \dots, A_l , the following matrix valued function of the complex variable λ is called a matrix polynomial of degree l and order m :*

$$A(\lambda) = A_0\lambda^l + A_1\lambda^{l-1} + \dots + A_{l-1}\lambda + A_l \quad (1)$$

Definition 2.2. *The matrix polynomial $A(\lambda)$ is called:*

- Monic if A_0 is the identity matrix.*
- Comonic if A_l is the identity matrix.*
- Regular or nonsingular if $\det(A(\lambda)) \neq 0$.*
- Unimodular if $\det(A(\lambda))$ is nonzero constant.*

Definition 2.3. *The complex number λ_i is called a latent root of the matrix polynomial $A(\lambda)$ if it is a solution of the scalar polynomial equation $\det(A(\lambda)) = 0$. The nontrivial vector p , solution of $A(\lambda_i)p = 0_m$ is called a primary right latent vector associated with λ_i . Similarly the nontrivial vector q solution of $q^T A(\lambda_i) = 0_m$ is called a primary left latent vector associated with λ_i .*

Remark 2.4. *If $A(\lambda)$ has a singular leading coefficient (A_l) then $A(\lambda)$ has latent roots at infinity. From the definition we can see that the latent problem of a matrix polynomial is a generalization of the concept of eigenproblem for square matrices. Indeed, we can consider the classical eigenvalues/vector problem as finding the latent root/vector of a linear matrix polynomial $(\lambda I - A)$. We can also define the spectrum of a matrix polynomial $A(\lambda)$ as being the set of all its latent roots (notation $\sigma(\lambda)$).*

Definition 2.5. *A right block root also called solvent of λ -matrix $A(\lambda)$ and is an $m \times m$ real matrix R such that:*

$$\begin{aligned}
R^l + A_1 R^{l-1} + \dots + A_{l-1} R + A_l &= O_m \\
\Leftrightarrow A_R(R) &= \sum_{i=0}^l A_i R^{l-i} = O_m
\end{aligned} \tag{2}$$

While a left solvent is an $m \times m$ real matrix L such that:

$$\begin{aligned}
L^l + L^{l-1} A_1 + \dots + L A_{l-1} + A_l &= O_m \\
\Leftrightarrow A_L(L) &= \sum_{i=0}^l L^{l-i} A_i = O_m
\end{aligned} \tag{3}$$

Remark 2.6. From [[34]] we have the following:

- Solvents of a matrix polynomial do not always exist.
- Generalized right (left) eigenvectors of a right (left) solvent are the generalized latent vectors of the corresponding matrix polynomial

Definition 2.7. : A matrix R (respectively: L) is called a right (respectively: left) solvent of the matrix polynomial if and only if the binomial $(\lambda I - R)$ (respectively: $(\lambda I - L)$) divides exactly $A(\lambda)$ on the right (respectively: left).

From [[35],[36]], [37]], [38]] we have

Theorem 2.8. Given a matrix polynomial

$$A(\lambda) = A_0 \lambda^l + A_1 \lambda^{l-1} + \dots + A_{l-1} \lambda + A_l \tag{4}$$

a) The remainder of the division of $A(\lambda)$ on the right by the binomial $(\lambda I - X)$ is $A_R(X)$

b) The remainder of the division of $A(\lambda)$ on the left by the binomial $(\lambda I - X)$ is $A_L(X)$

Hence there exist matrix polynomials $Q(\lambda)$ and $S(\lambda)$ such that:

$$\begin{aligned}
A(\lambda) &= Q(\lambda)(\lambda I - X) + A_R(X) \\
&= (\lambda I - X)S(\lambda) + A_L(X)
\end{aligned} \tag{5}$$

Corollary 2.9. The fundamental relation that exist between right solvent (respectively: left solvent) and right (respectively: left) linear factor:

$$\begin{aligned}
A_R(X) &= 0 \text{ iff } A(\lambda) = Q(\lambda)(\lambda I - X) \\
A_L(X) &= 0 \text{ iff } A(\lambda) = (\lambda I - X)S(\lambda)
\end{aligned} \tag{6}$$

In reference [2] it is stated the following:

Theorem 2.10. : Consider the set of solvents $\{R_1, R_2, \dots, R_l\}$ constructed from the eigenvalues $(\lambda_1, \lambda_2, \dots, \lambda_l)$ of a matrix A_c . $\{R_1, R_2, \dots, R_l\}$ is a complete set of solvents if and only if:

$$\begin{cases} \cup \sigma(R_i) = \sigma(A_c) \\ \sigma(R_i) \cap \sigma(R_j) = \emptyset \\ \det(V_R(R_1, R_2, \dots, R_l)) \neq 0 \end{cases} \quad (7)$$

Where: σ denotes the spectrum of the matrix. V_R Vandermonde matrix corresponding to $\{R_1, R_2, \dots, R_l\}$ given as

$$V_R(R_1, R_2, \dots, R_l) = \begin{pmatrix} I_m & I_m & \dots & I_m \\ R_1 & R_2 & \dots & R_l \\ \vdots & \vdots & \dots & \vdots \\ R_1^{l-1} & R_2^{l-1} & \dots & R_l^{l-1} \end{pmatrix} \quad (8)$$

Remark 2.11. : we can define a set of left solvents in the same way as in the previous theorem. The relationship between latent roots, latent vectors, and the solvents can be stated as follows:

From [[39]] we have the following:

Theorem 2.12. :If $A(\lambda)$ has $n = ml$ linearly independent right latent vectors p_{ij} ($i = 1, \dots, l$) and ($j = 1, \dots, m$) (left latent vectors q_{ij} $i = 1, \dots, l$ and $j = 1, \dots, m$) corresponding to latent roots λ_{ij} then $P_i \Lambda_i P_i^{-1}, (Q_i \Lambda_i Q_i^{-1})$ is a right (left) solvent.

Where: $P_i = [p_{i1}, p_{i2}, \dots, p_{im}]$, $Q_i = [q_{i1}, q_{i2}, \dots, q_{im}]^T$ and $\Lambda_i = \text{diag}(\lambda_{i1}, \lambda_{i2}, \dots, \lambda_{im})$.

Theorem 2.13. If $A(\lambda)$ has $n = ml$ latent roots $\lambda_{i1}, \lambda_{i2}, \dots, \lambda_{im}$ and the corresponding right latent vectors $p_{i1}, p_{i2}, \dots, p_{im}$ has as well as the left latent vectors $q_{i1}, q_{i2}, \dots, q_{im}$ are both linearly independent, then the associated right solvent R_i and left solvent L_i are related by: $R_i = W_i L_i W_i^{-1}$ Where: $W_i = P_i Q_i$ and $P_i = [p_{i1}, p_{i2}, \dots, p_{im}]$, $Q_i = [q_{i1}, q_{i2}, \dots, q_{im}]^T$. and "T" stands for transpose

For analysis and design of large-scale multivariable systems, it is necessary to determine a complete set of solvents of the matrix polynomial. Given the matrix polynomial $A(\lambda)$ if a right solvent R is obtained, the left solvent of L of $A(\lambda)$ associated with R can be determined by using the following [[39]]:

$$L = Q^{-1} R Q \quad \text{rank}(Q) = m \quad (9)$$

Where Q is the solution of the following linear matrix equation [[39]]:

$$\sum_{i=0}^{l-1} R^{l-1-i} Q B_i = I_m \quad (10)$$

Or in vector form using the Kronecker product we have

$$Vec(Q) = \left(\sum_{i=0}^{l-1} B_i^T \otimes (R^{l-1-i}) \right)^{-1} Vec(I_m) \quad (11)$$

Where: \otimes designates the Kronecker product, and B_i are the matrix coefficients of $B(\lambda)$ with $\lambda I_m - R$ factored out from $A(\lambda)$, i.e.,

$$B(\lambda) = A(\lambda) (\lambda I_m - R)^{-1} = \sum_{i=0}^{l-1} B_i \lambda^{l-1-i} \quad (12)$$

$$B(\lambda) = B_0 \lambda^{l-1} + B_1 \lambda^{l-2} + \dots + B_{l-1} \quad (13)$$

We can compute the coefficients B_i , using the algorithm of synthetic division:

$$\begin{aligned} B_0 &= I_m \\ B_1 &= B_0 A_1 + B_0 R \\ &\dots \\ B_k &= B_0 A_k + B_{k-1} R \quad k = 1, 2, \dots, l-1 \\ O_m &= B_0 A_l + B_{l-1} R \end{aligned}$$

Theorem 2.14. *If the elementary divisors of $A(\lambda)$ are linear, then $A(\lambda)$ can be factored into the product of l -linear monic λ -matrices called a complete set of spectral factors.*

$$A(\lambda) = (\lambda I_m - Q_l)(\lambda I_m - Q_{l-1}) \dots (\lambda I_m - Q_1) \quad (14)$$

where: $(\lambda I_m - Q_i), i = 1 \dots l$ are referred to as a complete set of linear spectral factors. The $m \times m$ complex matrices $Q_i, i = 1 \dots l$ are called the spectral factors of the λ -matrix $A(\lambda)$.

The most right spectral factor Q_1 is a right solvent of $A(\lambda)$ and the most left spectral factor Q_l is a left solvent of $A(\lambda)$, whereas the spectral factors may or may not be solvents of $A(\lambda)$. The relationship between solvents and spectral factors are studied by Shieh and Tsay in reference [24].

2.2 Transformation of solvents to spectral factors

The diagonal forms of a complete set of solvents and those of a complete set of spectral factors are identical and are related by similarity transformation.

Theorem 2.15. *:[23] Consider a complete set of right solvents $\{R_1, R_2, \dots, R_l\}$ of monic λ -matrix $A(\lambda)$, then $A(\lambda)$ can be factored as:*

$$A(\lambda) = N_l(\lambda) = (\lambda I_m - Q_l)(\lambda I_m - Q_{l-1}) \dots (\lambda I_m - Q_1)$$

By using the following recursive scheme: (for $k = 1, \dots, l$)

$$Q_k = (N_{(k-1)R}(R_k)) R_k (N_{(k-1)R}(R_k))^{-1} \quad (15)$$

where:

$$N_k(\lambda) = (\lambda I_m - Q_k)N_{k-1}(\lambda) \quad k = 1, \dots, l \quad (16)$$

and for any j we write

$$N_{kR}(R_j) = N_{(k-1)R}(R_j)R_j - Q_k N_{(k-1)R}(R_j), \quad k = 1, \dots, l$$

with:

$$N_0(\lambda) = I_m \quad N_{0R}(R_j) = I_m \quad \text{for any } j \text{ and}$$

$$\text{rank}(N_{(k-1)R}(R_k)) = m, \quad k = 1, \dots, l$$

Similarly the spectral factors can be obtained from the known L_i of $A(\lambda)$ as follow: (for $k = 1, \dots, l$)

$$Q_k = Q_{l+1-k} \quad (17)$$

$$Q_k = (M_{(k-1)L}(L_k))^{-1} L_k (M_{(k-1)L}(L_k)) \quad (18)$$

where:

$$M_k(\lambda) = M_{k-1}(\lambda)(\lambda I_m - Q_k) \quad k = 1, \dots, l \quad (19)$$

and for any j we write

$$M_{kL}(L_j) = L_j M_{(k-1)L}(L_j) - M_{(k-1)L}(L_j)Q_k, \quad k = 1, \dots, l$$

with:

$$M_0(\lambda) = I_m \quad M_{0L}(L_j) = I_m \quad \text{for any } j$$

$$\text{rank}(M_{(k-1)L}(L_k)) = m \quad k = 1, \dots, l$$

$M_{(k-1)L}(L_i)$ is a left matrix polynomial of $M_{(k-1)}(\lambda)$ having λ replaced by a left solvent L_j the spectral factorization of $A(\lambda)$ becomes:

$$A(\lambda) = M_l(\lambda) = (\lambda I_m - Q_1)(\lambda I_m - Q_2)\dots(\lambda I_m - Q_l)$$

2.3 Transformation of spectral factors to solvents

Given a complete set of spectral factors of a λ -matrix $A(\lambda)$, then a corresponding complete set of right (left) solvents can be obtained. The transformation of spectral factors to right (left) solvents of a λ -matrix can be derived as follow [[23]]:

Theorem 2.16. *Given a monic λ -matrix with all elementary divisors being linear*

$$A(\lambda) = (\lambda I_m - Q_l)(\lambda I_m - Q_{l-1})\dots(\lambda I_m - Q_1)$$

where $Q_i (\triangleq Q_{l+1-i})$ $i = 1, \dots, l$ are a complete set of spectral factors of a λ -matrix $A(\lambda)$, and $Q_i \cap Q_j = \emptyset$

Now let us define λ -matrices $N_i(\lambda)$ $i = 1, \dots, l$ as follow:

$$N_i(\lambda) = (\lambda I - Q_i)^{-1} N_{i-1}(\lambda) \quad (20)$$

$$N_i(\lambda) = I_m \lambda^{l-i} + A_{1i} \lambda^{l-i-1} + \dots + A_{(l-i-1)i} \lambda + A_{(l-i)i} \quad (21)$$

with $N_0 = A(\lambda)$ then the transformation matrix P_i which transforms the spectral factor Q_i ($\triangleq Q_{l+1-i}$) to the right solvent R_i ($\triangleq R_{l+1-i}$) of $A(\lambda)$ can be constructed from the new algorithm as follow: ($\text{rank}(P_i) = m$)

$$R_i \triangleq R_{l+1-i} = P_i Q_i P_i^{-1} \quad i = 1, \dots, l \quad (22)$$

where: the $m \times m$ matrix P_i can be solved from the following matrix equation $i = 1, \dots, m$

$$\text{Vec}(P_i) = (G_{N_i}(Q_i))^{-1} \text{Vec}(I_m) \quad (23)$$

where $G_{N_i}(Q_i)$ ($\text{rank}(G_{N_i}(Q_i)) = m^2$) is defined by:

$$G_{N_i}(Q_i) \triangleq (Q_i^{l-i})^T \otimes I_m + (Q_i^{l-i-1})^T \otimes A_{1i} + \dots + Q_i^T \otimes A_{(l-i)i} + I_m \otimes A_{(l-i)i}$$

in the same way the complete set of spectral factors Q_i , $i = 1, 2, \dots, l$ can be converted into the left solvent L_i , $i = 1, 2, \dots, l$ using the following algorithm:

$$M_i(\lambda) = M_{i-1}(\lambda)(\lambda I_m - Q_i)^{-1}, \quad i = 1, \dots, l \quad (24)$$

$$M_i(\lambda) = I_m \lambda^{l-i} + A_{1i} \lambda^{l-i-1} + \dots + A_{(l-i-1)i} \lambda + A_{(l-i)i} \quad (25)$$

$$H_{M_i}(Q_i) \triangleq I_m \otimes Q_i^{l-i} + A_{1i}^T \otimes Q_i^{l-i-1} + \dots + A_{(l-i-1)i}^T \otimes Q_i + A_{(l-i)i}^T \otimes I_m$$

$$\text{Vec}(S_i) = (H_{M_i}(Q_i))^{-1} \text{Vec}(I_m), \quad \text{rank}(H_{M_i}(Q_i)) = m^2$$

$$L_i = S_i^{-1} Q_i S_i \quad i = 1, \dots, l \quad (26)$$

2.4 Block companion forms

A useful tool for the analysis of matrix polynomials is the block companion form matrix. Given a λ -matrix as in eq(1) where $A_i \in C^{m \times m}$ and $\lambda \in C$, the associated block companion form matrices right (left) are:

$$A_R = \begin{pmatrix} O_m & I_m & \cdots & O_m \\ O_m & O_m & \cdots & O_m \\ \vdots & \vdots & \cdots & O_m \\ O_m & O_m & \vdots & I_m \\ -A_l & -A_{l-1} & \cdots & -A_1 \end{pmatrix}, \quad A_L = \begin{pmatrix} O_m & \cdots & O_m & -A_l \\ I_m & \cdots & O_m & -A_{l-1} \\ \vdots & \vdots & \vdots & \vdots \\ O_m & \cdots & O_m & -A_2 \\ O_m & \cdots & I_m & -A_1 \end{pmatrix} \quad (27)$$

Note that: A_L is the block transpose of A_R . If the matrix polynomial $A(\lambda)$ has a complete set of solvents, these companion matrices can be respectively block diagonalised via the right(left) block Vandermande matrix defined by:

$$V_R = \begin{pmatrix} I_m & I_m & \cdots & I_m \\ R_1 & R_2 & \cdots & R_l \\ \vdots & \vdots & \ddots & \vdots \\ R_1^{l-1} & R_2^{l-1} & \cdots & R_l^{l-1} \end{pmatrix} V_L = \begin{pmatrix} I_m & L_1 & \cdots & L_1^{l-1} \\ I_m & L_2 & \cdots & L_2^{l-1} \\ \vdots & \vdots & \ddots & \vdots \\ I_m & L_l & \cdots & L_l^{l-1} \end{pmatrix} \quad (28)$$

where R_1, R_2, \dots, R_l and/or L_1, L_2, \dots, L_l represent the complete set of right (left) solvents. Since the block Vandermande matrices are nonsingular [[?]], [[?]] and [[41]] we can write

$$V_R^{-1} A_R V_R = \text{Blockdiag}(R_1, R_2, \dots, R_l) \quad (29)$$

$$V_L^{-1} A_L V_L = \text{Blockdiag}(L_1, L_2, \dots, L_l) \quad (30)$$

These similarity transformations do a block decoupling of the spectrum of $A(\lambda)$ which is very useful in the analysis and design of large order control systems.

3 Special Fractorization Algorithms

In this section we are going to present algorithms that can factorize a linear term from a given matrix polynomial. Firstly we give the generalized quotient difference algorithm and next we give a new extended algorithm based on the Horner scheme. The matrix quotient-difference $Q.D.$ algorithm is a generalization of the scalar case [[42]] and it is developed in [[43]] The scalar $Q.D.$ algorithm is used for finding the roots of a scalar polynomial. The Quotient-Difference scheme for matrix polynomials can be defined just like the scalar one [[10]] and it consists on building a table that we call the $Q.D.$ tableau.

3.1 The right block matrix Q.D. algorithm

Given a matrix polynomial with nonsingular coefficients as in eq(1).The objective is to find the spectral factors of $A(\lambda)$ that will allow as write $A(\lambda)$ as a product of n -linear factors as in eq(1). The block left companion form, is:

$$C_3 = \begin{pmatrix} -A_1 & I_m & \cdots & O_m \\ -A_2 & O_m & \cdots & O_m \\ \vdots & \vdots & \ddots & \vdots \\ -A_{l-1} & \cdots & \cdots & I_m \\ -A_l & \cdots & \cdots & O_m \end{pmatrix} \quad (31)$$

The required transformation is a sequence of LR decomposition such that:

$$C_3 = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} = \begin{pmatrix} I_m & O_m \\ X_m & I_m \end{pmatrix} \begin{pmatrix} A & B \\ C & D \end{pmatrix} \quad (32)$$

where:

$$C_{11} = \begin{pmatrix} -A_1 & I_m & \cdots & O_m \\ -A_2 & O_m & \cdots & O_m \\ \vdots & \vdots & \ddots & \vdots \\ -A_{l-2} & \cdots & \cdots & I_m \\ -A_{l-1} & \cdots & \cdots & O_m \end{pmatrix}, C_{12} = \begin{pmatrix} O_m \\ O_m \\ \vdots \\ O_m \\ I_m \end{pmatrix}$$

$$C_{21} = [-A_l \ O_m \ O_m, \dots, O_m], \quad C_{22} = [O_m]$$

It is required to have $C = 0$, then let

$$X = [-X_1 \ X_2 \ X_3, \dots, X_{l-1}] \quad (33)$$

We obtain the following set of equations:

$$\begin{aligned} X_1 A_1 + X_2 A_2 + \dots + X_{l-1} A_{l-1} &= A_l \\ X_1 &= X_2 = \dots = X_{l-2} = 0 \\ X_{l-1} + D &= 0 \end{aligned}$$

Leading the following decomposition of C_3 :

$$C_3 = \begin{pmatrix} I_m & \cdots & O_m & O_m \\ O_m & \cdots & O_m & O_m \\ \vdots & \vdots & \vdots & \vdots \\ O_m & \cdots & I_m & O_m \\ O_m & \cdots & A_l A_{l-1}^{-1} & I_m \end{pmatrix} \begin{pmatrix} -A_1 & I_m & \cdots & O_m \\ -A_2 & O_m & \cdots & O_m \\ \vdots & \vdots & \ddots & \vdots \\ -A_{l-1} & O_m & \cdots & I_m \\ O_m & O_m & \cdots & -A_l A_{l-1}^{-1} \end{pmatrix}$$

Hence C_3 can be written as:

$$C_3 = L_{-(l-2)} R_{-(l-2)} \quad (34)$$

Continuing this process of the block C_{11} up when C_3 is equivalent to a matrix R_0

$$C_3 = L_{-(l-2)} L_{-(l-3)} \dots L_0 R_0 \quad (35)$$

$$R_0 = \begin{pmatrix} -A_1 & I_m & \cdots & O_m & O_m \\ O_m & -A_2 A_1^{-1} & \cdots & O_m & O_m \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ O_m & O_m & \cdots & -A_{l-1} A_{l-2}^{-1} & I_m \\ O_m & O_m & \cdots & O_m & -A_l A_{l-1}^{-1} \end{pmatrix} \quad (36)$$

It is clear that if the matrices $L_0, L_{-1}, \dots, L_{l-2}$ are identity matrices, then the block companion matrix C_3 will be :

$$M = \begin{pmatrix} Q_1 & I_m & \cdots & O_m & O_m \\ O_m & Q_2 & \cdots & O_m & O_m \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ O_m & O_m & \cdots & Q_{l-1} & I_m \\ O_m & O_m & \cdots & O_m & Q_l \end{pmatrix} \quad (37)$$

The following theorem shows that under certain conditions, the sequence of $L_0, L_{-1}, \dots, L_{l-2}$ converges to identities see [[44]] and [[10]]:

Theorem 3.1. *Let $M = X\Lambda X^{-1}$ where*

$$\Lambda = \begin{pmatrix} R_1 & O_m & \cdots & O_m \\ O_m & R_2 & \cdots & O_m \\ \vdots & \vdots & \ddots & \vdots \\ O_m & O_m & \cdots & R_l \end{pmatrix} \quad (38)$$

If the following conditions are satisfied:

- a) *dominance relation between R_k : $R_1 > R_2 > \dots R_l$*
- b) *$X^{-1} = Y$ has a Block LR factorization $L_y R_y$*
- c) *X has a Block LR factorization $L_x R_x$*

Then the block LR algorithm just defined converges (i.e. $L_k \rightarrow I$)

The above theorem states that we can start the *Q.D.* algorithm by considering:

$$\begin{aligned} E_1^{(0)} &= -A_2 A_1^{-1}, & E_2^{(-1)} &= -A_3 A_2^{-1}, \\ E_3^{(-2)} &= -A_4 A_3^{-1}, \dots, & E_{l-1}^{-(l-2)} &= -A_l A_{l-1}^{-1} \\ Q_1^{(0)} &= -A_1, & Q_2^{(-1)} &= O_m, \dots, & Q_{l-1}^{-(l-2)} &= O_m \end{aligned}$$

The last two equations provide us with the first two rows of the *Q.D.* tableau (one row of Q 's and one row of E 's). Hence, we can solve the rhombus rules for the bottom element (called the south element by Henrici [[42]]). We obtain the row generation of the *Q.D.* algorithm:

$$\begin{cases} Q_i^{(j+1)} = Q_i^{(j)} + E_i^{(j)} - E_i^{(j+1)} \\ E_i^{(j+1)} = Q_{i+1}^{(j)} E_i^{(j)} [Q_i^{(j+1)}]^{-1} \end{cases} \quad (39)$$

Writing this in tabular form yield

where the $Q_i^{(j)}$ are the spectral factors of $A(\lambda)$. In addition, the *Q.D.* algorithm gives all spectral factors simultaneously and in dominance order. We have chosen, in the above, the row generation algorithm because it is numerically stable see reference [[10]] for details.

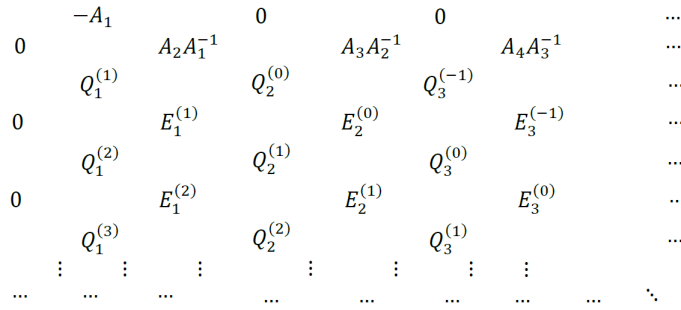


Figure 1: *
Q.D algorithm

Example 3.2. Consider a matrix polynomial of 2nd order and 3rd degree with the following matrix coefficients.

$$A_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad A_1 = \begin{pmatrix} -27.1525 & 0.8166 \\ -179.7826 & 38.1525 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 116.4 & 85.0 \\ 1043.4 & 836.7 \end{pmatrix}, \quad A_3 = \begin{pmatrix} 126.9 & 353.5 \\ 1038.7 & 2947.6 \end{pmatrix}$$

We apply now the generalized row generation Q.D. algorithm to find the complete set of spectral factors and then we use the similarity transformations given by Shieh [[39]] to obtain the complete set of solvents both left and right equations(19)-(26).

Step 1: initialization of the program to start

Enter the degree and the order $m = 2, l = 3$

Enter the number of iterations $N = 35$

Enter the matrix polynomial coefficients A_i

Step 2: Construct Q_1 and E_1 the first row of Q 's and first row of E 's

$$Q_1 = [-A_1A_0^{-1} \ O_2 \ O_2], \quad E_1 = [O_2 \ A_2A_1^{-1} \ A_3A_2^{-1} \ O_2]$$

Step 3: Building or generating the rest rows using the rhombus rules

```

For n = 1 : N
E2 = []; Q2 = [];
For k = 1 : 2 : m * l
q2 = (E1(:, k + 2 : k + 3) - E1(:, k : k + 1)) + Q1(:, k : k + 1);
Q2 = [Q2, q2];
End
Q2;
For k = 1 : 2 : m * l - 2
e2 = (Q2(:, k + 2 : k + 3))(E1(:, k : k + 1))(Q2(:, (k : k + 1)))-1;
E2 = [E2, e2];
End
E2 = [O2, E2, O2];
Q1 = Q2;
E1 = E2;
End
Q1;
S1 = Q1(:, 1 : 2) S2 = Q1(:, 3 : 4) S3 = Q1(:, 5 : 6)

```

Running the above steps (1) to (3) we obtain the following complete set of spectral factors S_i :

$$Q_1 = \begin{pmatrix} 3.0000 & 2.0000 & -8.2908 & 0.7118 & 32.4434 & -3.5284 \\ -90.000 & -15.000 & -16.8400 & 8.1248 & 286.6226 & -31.2773 \end{pmatrix}$$

$$S_1 = \begin{pmatrix} 3.0 & 2.0 \\ -90.0 & -15.0 \end{pmatrix}, S_2 = \begin{pmatrix} -8.2908 & 0.7118 \\ -16.8400 & 8.1248 \end{pmatrix},$$

$$S_3 = \begin{pmatrix} 32.4434 & -3.5284 \\ 286.6226 & -31.2773 \end{pmatrix}$$

Now, we should extract a complete set of right solvent from those block spectra using the algorithmic similarity transformations in equations from (21) to (24).

Step 4: Reverse the orientation of spectral factors

$$U_1 = S_3; U_2 = S_2; U_3 = S_1$$

Step 5: Evaluate the coefficients using the synthetic long division and then find the corresponding transformation matrix as in theorem 2.16.

$$\begin{aligned}
N_{11} &= A_1 + U_1; \\
N_{12} &= A_2 + U_1 \star N_{11}; \\
G_1 &= (U_1^2)^T \otimes I_2 + (U_1)^T \otimes N_{11} + I_2 \otimes N_{12}; \\
\text{vec} p_1 &= G_1^{-1} \star [1; 0; 0; 1] \\
p_1 &= [\text{vec} p_1(1 : 2), \text{vec} p_1(3 : 4)]; \\
R_1 &= p_1 \star U_1 \star (p_1)^{-1};
\end{aligned}$$

You can verify the first solvent using:

$$\text{rightzero1} = A_0 \star (R_1)^3 + A_1 \star (R_1)^2 + A_2 \star R_1 + A_3$$

Step 6: redo the same process for the next right solvents

$$\begin{aligned}
N_{21} &= N_{11} + U_2; \\
G_2 &= (U_2)^T \otimes I_2 + I_2 \otimes N_{21}; \\
\text{vec}p_2 &= G_2^{-1} \star [1; 0; 0; 1] \\
p_2 &= [\text{vec}p_2(1 : 2), \text{vec}p_2(3 : 4)]; \\
R_2 &= p_2 \star U_2 \star (p_2)^{-1};
\end{aligned}$$

For verification also you can use:
 $\text{rightzero} = A_0 \star (R_2)^3 + A_1 \star (R_2)^2 + A_2 \star R_2 + A_3$

Step 7: The last solvents are obtained directly from the most left spectral factor:
 $R_3 = S_1$ or by using the transformation:

$$\begin{aligned}
G_3 &= (I_2)^T \otimes I_2; \\
\text{vec}p_3 &= G_3^{-1} \star [1; 0; 0; 1] \\
p_3 &= [\text{vec}p_3(1 : 2), \text{vec}p_3(3 : 4)]; \\
R_3 &= p_3 \star U_3 \star p_3^{-1} = U_3;
\end{aligned}$$

The final results are :

$$\begin{aligned}
R_1 &= \begin{pmatrix} 0.3637 & -4.5495 \\ -0.8183 & 0.8024 \end{pmatrix}, R_2 = \begin{pmatrix} 7.2354 & 1.4024 \\ 1.2995 & -7.4015 \end{pmatrix}, \\
R_3 &= \begin{pmatrix} 3.0000 & 2.0000 \\ -90.000 & -15.000 \end{pmatrix}
\end{aligned}$$

Finally, we can also obtain the corresponding complete set of left solvents using the algorithmic similarity transformation described in equations from (10) to (12).

Step 8: coefficients determination using the synthetic long division

$$\begin{aligned}
B_{0i} &= I_2; \\
B_{1i} &= A_1 + R_i; \\
B_{2i} &= A_2 + B_{1i} \star R_i;
\end{aligned}$$

Step 9: find the corresponding similarity transformation matrix as in equations from (10) to (12).

For $i = 1 : l$

$$\begin{aligned}
\text{vec}Q_i &= ((B_{0i})^T \otimes (R_i)^2 + (B_{1i})^T \otimes R_i + (B_{2i})^T \otimes I_2)^{-1} \star [1; 0; 0; 1]; \\
Q_i &= [\text{vec}Q_i(1 : 2), \text{vec}Q_i(3 : 4)]; \\
L_i &= (Q_i)^{-1} \star R_i \star Q_i;
\end{aligned}$$

You can verify the left solvents using:
 $\text{Leftzero} = L_i^3 \star A_0 + L_i^2 \star A_1 + L_i \star A_2 + A_3$
End

The left solvents are now obtained:

$$L_1 = \begin{pmatrix} 32.443 & -3.5284 \\ 286.622 & -31.2773 \end{pmatrix}, L_2 = \begin{pmatrix} 25.1323 & -2.8370 \\ 204.5931 & -25.2983 \end{pmatrix},$$

$$L_3 = \begin{pmatrix} 21.0123 & -4.6531 \\ 178.0910 & -33.0123 \end{pmatrix}$$

3.2 Extended Horner algorithm

Horner's method is a technique to evaluate polynomials quickly. It needs l multiplications and l additions and it is also a nested algorithmic programming that can decompose a polynomial into a multiplication of l linear factors (Horner's method) based on the Euclidian synthetic long division.

Similarly Horner's method is a nesting technique requiring only l multiplications and l additions to evaluate an arbitrary l^{th} -degree polynomial [[45]].

Theorem 3.3. Let the function $P(x)$ be the polynomial of degree l defined on the real field $P : R \rightarrow R$ where: a_i are constant coefficients and x is real variable.

$$P(x) = a_0x^l + a_1x^{l-1} + \dots + a_{l-1}x + a_l \quad (40)$$

If $b_0 = a_0$ and $b_k = a_k + b_{k-1}\alpha$, $k = l, \dots, 2, 1$

Then $b_l = P(\alpha)$ and $P(x)$ can be written as:

$$P(x) = (x - \alpha)Q(x) + b_l \quad (41)$$

Where:

$$Q(x) = b_0x^{l-1} + b_1x^{l-2} + \dots + b_{l-2}x + b_{l-1} \quad (42)$$

Proof. The theorem can be proved using a direct calculation.

$$P(x) = a_0x^l + a_1x^{l-1} + \dots + a_{l-1}x + a_l$$

$$P(x) = (x - \alpha)(b_0x^{l-1} + b_1x^{l-2} + \dots + b_{l-2}x + b_{l-1}) + b_l$$

Identifying the coefficients of x with different powers we get:

$$\begin{aligned} b_0 &= a_0 \\ b_1 &= a_1 + b_0\alpha \\ &\vdots \\ b_k &= a_k + b_{k-1}\alpha \quad \text{where } k = l, l-1, \dots, 2 \end{aligned}$$

Now if α is a root of the polynomial $P(x)$, then b_l should be zero, and $a_l + b_{l-1}\alpha = 0$.

Hence, we may write

$$\alpha = -\left(\frac{a_l}{b_{l-1}}\right) \quad \text{or} \quad x_{k+1} = -\left(\frac{a_l}{b_{l-1,k}}\right) \quad k = 0, 1, \dots$$

The algorithm of Horner method in its recursive formula is then:

$$b_{i,k} = a_k + b_{i,k-1}x_k; \quad i = 1, \dots, l \quad \text{and} \quad b_{0,k} = a_0$$

□

Now we generalize this nested algorithm to matrix polynomials, consider the monic λ -matrix $A(\lambda)$ and according to theorem 2.8 the matrix $A(\lambda)$ can be factored as:

$$A(\lambda) = Q(\lambda)(\lambda I - X) + A_R(X) \quad (43)$$

where

$$A(\lambda) = A_0\lambda^l + A_1\lambda^{l-1} + \dots + A_l = \sum_{i=0}^l A_i\lambda^{l-i}$$

$$Q(\lambda) = B_0\lambda^{l-1} + B_1\lambda^{l-2} + \dots + B_{l-1} = \sum_{i=0}^{l-1} B_i\lambda^{l-i-1}$$

$$A_R(X) = \text{cst}$$

Using the algorithm of synthetic long division for matrices we obtain:

$$\begin{aligned} B_0 &= A_0 = I_m \\ B_1 &= B_0A_1 + B_0X \\ &\dots \\ B_k &= B_0A_k + B_{k-1}X \quad k = 1, 2, \dots, l-1 \\ O_m &= B_0A_l + B_{l-1}X \end{aligned}$$

From the last two equations we can iterate the process to get recursive algorithm as follow:

Algorithm:

Enter the number of iterations N

For $k = 0 : N$

Enter the degree and the order m, l

Enter the matrix polynomial coefficients A_i

X_0 =initial guess;

For $i = 1 : l$

$$B_{i,k} = B_0A_i + B_{i-1,k}X_k;$$

End

$$X_{k+1} = - \left(B_{(l-1),k} \right)^{-1} B_0A_l;$$

$$X_k = X_{k+1};$$

End

When you get the first spectral factor repeat the process until you get the complete set.

Example 3.4. Consider a matrix polynomial of 2nd order and 3rd degree with the following matrix coefficients.

$$A(\lambda) = A_0\lambda^3 + A_1\lambda^2 + A_2\lambda + A_3$$

With

$$A_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad A_1 = \begin{pmatrix} 11.0000 & -1.0000 \\ 6.7196 & 17.0000 \end{pmatrix},$$

$$A_2 = \begin{pmatrix} 30.0000 & -11.0000 \\ 70.9107 & 82.5304 \end{pmatrix}, \quad A_3 = \begin{pmatrix} -0.0000 & -30.0000 \\ 182.0000 & 89.8393 \end{pmatrix}$$

We apply now the extended Horner's method via its algorithmic version to find the complete set of spectral factors and then we use the similarity transformations given in [[23]] to obtain the complete set of left and right solvents

The Block Horner scheme is:

When $(X_0 = O_2)$ is an initial guess

Initial starting value of iterations $k = 0$ and $(X_0 = O_2)$

η : tolerance error

Given ξ where $\xi \geq \eta$

While $\xi \geq \eta$

$$A_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$A_1 = \begin{pmatrix} 11.0000 & -1.0000 \\ 6.7196 & 17.0000 \end{pmatrix} \quad B_0 = A_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 30.0000 & -11.0000 \\ 70.9107 & 82.5304 \end{pmatrix}, \quad \begin{cases} B_1(k) = B_0A_1 + B_0X_k \\ B_1(0) = \begin{pmatrix} 11.0000 & -1.0000 \\ 6.7196 & 17.0000 \end{pmatrix} \end{cases}$$

$$A_3 = \begin{pmatrix} -0.0000 & -30.0000 \\ 182.0000 & 89.8393 \end{pmatrix}, \quad \begin{cases} B_2(k) = B_0A_2 + B_1(k)X_k \\ B_2(0) = \begin{pmatrix} 30.0000 & -11.0000 \\ 70.9107 & 82.5304 \end{pmatrix} \end{cases}$$

$$X_{k+1} = -(B_2(k))^{-1}B_0A_3$$

$$\xi = 100. \frac{\|X_{k+1} - X_k\|}{\|X_k\|}$$

$$X_k = X_{k+1}$$

$$k = k + 1$$

End

Running the above algorithm we obtain the next complete set of spectral factors:

$$S_1 = \begin{pmatrix} 0.00 & 1.00 \\ -3.25 & 2.00 \end{pmatrix}, \quad S_2 = \begin{pmatrix} -5.000 & 0.000 \\ -1.6042 & -7.000 \end{pmatrix}$$

$$S_3 = \begin{pmatrix} -6.000 & -0.000 \\ -1.8655 & 8.000 \end{pmatrix}$$

Finally when we apply the similarity transformation algorithm as in equations from (21) to (24) to right (or left) solvent form we get:

$$R_1 = \begin{pmatrix} -5.9574 & 0.2553 \\ -0.3404 & -8.0426 \end{pmatrix}, \quad R_2 = \begin{pmatrix} -4.9412 & 0.2941 \\ -0.4118 & -7.0588 \end{pmatrix}$$

$$R_3 = \begin{pmatrix} 0.000 & 1 \\ -3.25 & -2 \end{pmatrix}$$

3.2.1 Reformulation of the Block Horner method

An alternative form of the previous algorithm (under matrix and algebraic manipulations is:

$$\begin{aligned} B_0(k) &= A_0 = I_m \\ B_1(k) &= B_0(k)A_1 + B_0(k)X(k) \\ &\dots \\ B_{l-1}(k) &= B_0(k)A_{l-1} + B_{l-2}(k)X(k) \\ O_m &= B_0(k)A_l + B_{l-1}(k)X(k) \end{aligned}$$

$$\begin{aligned} B_{l-1}(k) &= A_{l-1} + \dots + A_1X^{l-1}(k) + B_0X^l(k) \\ \Rightarrow B_{l-1}(k) &= [A_R(X_k) - A_l]X_k^{-1} \\ \Rightarrow (B_{l-1}(k))^{-1} &= X_k[A_R(X_k) - A_l]^{-1} \end{aligned}$$

Finally we obtain the following iterative formula: ($k = 0, 1, \dots$)

$$X_{k+1} = -(B_{l-1}(k))^{-1}A_l = X_k[A_l - A_R(X_k)]^{-1}A_l \quad (44)$$

Algorithm 3.5. Enter the degree and the order m, l
Enter the matrix polynomial coefficients $A_i \in R^{m \times m}$
 $X_0 \in R^{m \times m} =$ initial guess;
Give some small η and ($\delta =$ initial start) $> \eta$
 $k = 0$

While $\delta \geq \eta$

$$\begin{aligned} X_{k+1} &= X_k[A_l - A_R X_k]^{-1}A_l; \\ \delta &= 100. \frac{\|X_{k+1} - X_k\|}{\|X_k\|}; \\ X_k &\leftarrow X_{k+1}; \\ k &= k + 1; \end{aligned}$$

Convergence condition: Using equations (44) we obtain the conditions for the the algorithm to converge to the solution.

1. *Upper bound*

$$\begin{aligned}
\text{eq(45)} &\Leftrightarrow X_{k+1} - X_k = X_{k+1}A_l^{-1}A_R(X_k) \\
\text{eq(45)} &\Leftrightarrow \|X_{k+1} - X_k\| = \|X_{k+1}A_l^{-1}A_R(X_k)\| \\
\text{eq(45)} &\Leftrightarrow \|X_{k+1} - X_k\| \leq \|X_{k+1}\| \cdot \|A_l^{-1}\| \cdot \|A_R X_k\| \\
\text{eq(45)} &\Leftrightarrow \frac{\|X_{k+1} - X_k\|}{(\|X_{k+1}\| \cdot \|A_l^{-1}\|)} \leq \|A_R(X_k)\|
\end{aligned}$$

Now if X_k tends to constant matrix $\|X_k\| \rightarrow M$ as $k \rightarrow \infty$ and $\|X_k^{-1}\| \rightarrow N$ with $\|A_l^{-1}\| = \gamma$ and $\|A_l\| = \delta$ then:

$$\begin{aligned}
\lim_{k \rightarrow \infty} \|A_R(X_k)\| &\geq \frac{\|X(k+1) - X_k\|}{(\|X(k+1)\| \cdot \|A_l^{-1}\|)} \\
&\Rightarrow \lim_{k \rightarrow \infty} \|A_R(X_k)\| \geq \frac{\xi_k}{\gamma \cdot M}
\end{aligned} \tag{45}$$

2. *Lower bound*

$$\begin{aligned}
A_R(X_k) &= A_l[I - X_{k+1}^{-1}X_k] = A_l(X_{k+1})^{-1}[X_{k+1} - X_k] \\
&\Rightarrow \|A_R(X_k)\| \leq \|A_l\| \cdot \|(X_{k+1})^{-1}\| \cdot \|X_{k+1} - X_k\| \\
&\Rightarrow \lim_{k \rightarrow \infty} \|A_R(X_k)\| \leq \delta \cdot N \xi_k
\end{aligned} \tag{46}$$

From eq. (45) and (46) we obtain:

$$\frac{\xi_k}{\gamma \cdot M} \leq \lim_{k \rightarrow \infty} \|A_R(X_k)\| \leq \delta \cdot N \xi_k \tag{47}$$

Finally if the matrix X_k tends to constant matrix $X_k \rightarrow S$ and $(A_l - A_R(X_k))$ is nonsingular matrix then S is a solvent of the matrix polynomial $A_R(S) = O_m$.

Convergence Type: To get the convergence type we should evaluate a ratio relationship between any two successive differences.

$$X_{k+1} - S = X_k([A_l - A_R(X_k)]^{-1}A_l - I) + X_k - S \tag{48}$$

We define $F(X_k) = ([A_l - A_R(X_k)]^{-1}A_l - I)$ then we have:

$$\|X_k - S\| - \|X_k F(X_k)\| \leq \|X_{k+1} - S\| \leq \|X_k - S\| + \|X_k F(X_k)\| \tag{49}$$

We know that:

$$\lim_{k \rightarrow \infty} \|X_k F(X_k)\| = \lim_{k \rightarrow \infty} \Delta_k = \xi \tag{50}$$

from equations (49) and (50) we deduce that:

$$1 - \frac{\Delta_k}{\|X_k - S\|} \leq \frac{\|X_{k+1} - S\|}{\|X_k - S\|} \leq 1 + \frac{\Delta_k}{\|X_k - S\|}$$

Finally:

$$\lim_{k \rightarrow \infty} \frac{\|X_{k+1} - S\|}{\|X_k - S\|} = 1 \quad (51)$$

Example 3.6. consider the following matrix polynomial with repeated spectral factor:

$$A(\lambda) = \left[\lambda I - \begin{pmatrix} -7.1230 & -6.3246 \\ 5.9279 & 5.1230 \end{pmatrix} \right]^2 = A_0 \lambda^2 + A_1 \lambda + A_2$$

With

$$A_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad A_1 = \begin{pmatrix} 14.2461 & 12.6493 \\ -11.8557 & -10.2461 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 13.2461 & 12.6493 \\ -11.8557 & -11.2461 \end{pmatrix}$$

Find X such that $A_R(X) = O_2$

$$A_R(X) = A_0 X^2 + A_1 X + A_2$$

If we apply the Block Horner algorithm we find

$$X_1 = \begin{pmatrix} -2.7323 & -1.8068 \\ 1.6798 & 0.7521 \end{pmatrix}, \quad X_2 = \begin{pmatrix} -11.5138 & -10.8424 \\ 10.1759 & 9.4939 \end{pmatrix}$$

Remark 3.7. The Proposed Horner algorithm finds the whole set of spectral factors if it exists, even if there is no dominance factor among them.

3.2.2 Crossbred Newton_Horner method

In order to accelerate the Block Horner method we make a crossbred (Hybrid) Generalized Newton algorithm which is very fast due to its restricted local nature (i.e. Quadratic convergence).

Horner iteration	Newton iteration
$X_{k+1} - X_k = X_{k+1} A_l^{-1} A_R(X_k)$	$X_{k+1} - X_k = -J^{-1}(X_k) A_R(X_k)$

Combine them we get:

$$X_{(k+1)} = X_k + (X_k - J^{-1}(X_k) A_R(X_k)) A_l^{-1} A_R(X_k) \quad (52)$$

where $J(X_k)$ is the Frechet differential.

Definition 3.8. Let B_1 and B_2 be Banach spaces and A_R a nonlinear operator from B_1 to B_2 . If there exists a linear operator L from B_1 to B_2 such that:

$$\begin{aligned} B_1 &\rightarrow B_2 \\ H &\rightarrow L(X + H) \end{aligned}$$

Where:

$$\begin{aligned} \|A_R(X + H) - A_R(X) - L(X + H)\| &= O(\|H\|) \\ X, H \in B_1 \quad A_R(X), L(X + H) &\in B_2 \end{aligned}$$

Then $L(X + H)$ is called the Frechet derivative of A_R at X and sometimes is written $dA_R(X, H)$. Also is read the Frechet derivative of A_R at X in the direction H . And $J(X_k).H = L(X + H)$

Algorithm

Begin

Enter the degree and the order m, l

Enter the matrix polynomial coefficients $A_i \in R^{m \times m}$

$X_0 \in R^{m \times m}$ = initial guess;

Give initial $J(X_0) \in R^{m \times m}$

Give some small η and (δ =initial start) $> \eta$

$k = 0$

While $\delta \geq \eta$

$$\begin{aligned} X_{k+1} &= X_k(I_m + A_l^{-1}A_R(X_k)) - J^{-1}(X_k)A_R(X_k)A_l^{-1}A_R(X_k); \\ H_k &= X_{k+1} - X_k; \\ J(X_k) &= (A_R(X(k+1)) - A_R(X_k))H_k^{-1}; \\ \delta &= 100. \frac{\|X_{k+1} - X_k\|}{\|X_k\|}; \\ X_k &\leftarrow X_{k+1}; \\ k &= k + 1; \end{aligned}$$

End

3.2.3 Two stage Block Horner algorithm

To accelerate the block Horner algorithm we use now a two stage Newton like iteration. Now by using theorem 2.8 we obtain:

$$\begin{aligned} A_R(X) &= (X - \Theta)(X^{l-1} + B_1X^{l-2} + \dots + B_{l-1}) + B_l \\ \Rightarrow A_R(X) &= (X - \Theta)Q(X) + B_l \\ \Rightarrow X - \Theta &= (A_R(X) - B_l)Q^{-1}(X) \end{aligned}$$

where $Q(X) = X^{l-1} + B_1X^{l-2} + \dots + B_{l-1}$ and $A_R(\Theta) = B_l$ Now if Θ is a solvent then $B_l = O_m$ If now assume that $\Theta = X_{k+1}$ is a solvent to the matrix polynomial A_R then $A_R(X_{k+1}) = O_m$ and $X_{k+1} = X_k - A_R(X_k)Q^{-1}(X_{k+1})$. Set also $Q(X) = (X - \Theta)(X^{l-1} + C_1X^{l-2} + \dots + C_{l-2}) + C_{l-1}$ From the Horner scheme we can evaluate both B_i and C_i recursively:

$$\begin{array}{ll}
B_0 = I_m & C_0 = I_m \\
B_1 = A_1 + B_0X & C_1 = B_1 + C_0X \\
B_2 = A_2 + B_1X & C_2 = B_2 + C_1X \\
\vdots & \vdots \\
B_{l-1} = A_{l-1} + B_{l-2}X & \vdots \\
O_m = B_l = A_l + B_{l-1}X = A_R(X) & C_{l-1} = B_{l-1} + C_{l-2}X = Q(X)
\end{array}$$

After iterating the last equation we get: After iterating the last equation we get:

$$\begin{array}{ll}
B_l(k) = A_l + B_{l-1}(k)X_k & C_{l-1}(k) = B_{l-1}(k) + C_{l-2}(k)X_k \\
B_l(k) = A_R(X_k) & C_{l-1}(k) = Q(X_k)
\end{array}$$

Algorithm:

X_0 =initial guess, $(B_0 = C_0 = I) \in R^{m \times m}$
enter small enough number η (Tolerance error)
 $(\delta = \text{initial start}) > \eta$
enter the set of $m \times m$ (A_0, A_1, \dots, A_l) matrices
 $k = 0$

While $\delta > \eta$

For $i = 1 : 1 : l$

$$B_i(k) = A_i + B_{i-1}(k)X_k$$

End

For $i = 1 : 1 : l - 1$

$$C_i(k) = B_i(k) + C_{i-1}(k)X_k$$

End

$$X_{k+1} = X_k - B_l(k)(C_{l-1}(k))^{-1}$$

$$\delta = 100 \cdot \frac{\|X_{k+1} - X_k\|_2}{\|X_k\|_2}$$

$$X_k \leftarrow X_{k+1}$$

$$k \leftarrow k + 1$$

End

Remark 3.9. *this two stage algorithm gathers the two advantages of Horner sagem and Newton algorithm because it is nested programed nature, large sense independence on initial conditions and faster in execution due to the likeness or the conformity to Newton method.*

Example 3.10. *Given the following matrix polynomial*

$$A_R(X) = A_0X^3 + A_1X^2 + A_2X + A_3$$

Where:

$$A_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad A_1 = \begin{pmatrix} 12.8793 & -0.4881 \\ -2.0989 & 15.1207 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 56.5645 & -8.7887 \\ 10.2686 & 55.9659 \end{pmatrix}, \quad A_3 = \begin{pmatrix} 95.9331 & -37.5549 \\ 160.9539 & -6.0938 \end{pmatrix}$$

We apply the two stage Horner algorithm and after 15 iterations we get:

$$X_0 = \begin{pmatrix} 5.2114 & 4.8890 \\ 2.3159 & 6.2406 \end{pmatrix}, \quad X_1 = \begin{pmatrix} -3.0729 & 1.4058 \\ -4.6569 & 1.0730 \end{pmatrix}$$

With

$$A_R(X_{15}) = \begin{pmatrix} -0.0081 & 0.0106 \\ 0.0265 & 0.0145 \end{pmatrix}$$

3.2.4 Reformulation of the two stage Block Horner method

After back substitution of the nested programmed scheme and accumulation we obtain:

$$B_l(k) = A_R(X_k) \quad \text{and} \\ C_{l-1}(k) = lX_k^{l-1} + l-1A_1X_k^{l-2} + \dots + A_{l-1} = \Delta(X_k)$$

The two stage Block Horner Varian algorithm can be obtained when we use the compact forms of the matrices $B_l(k)$ and $C_{l-1}(k)$ in term of A_R lead us to Newton like iterated process.

Algorithm:

X_0 =initial guess

Enter small enough number η (Tolerance error) and (δ =initial start) $> \eta$

Enter the set of $m \times m$ (A_0, A_1, \dots, A_l) matrices

$k = 0$

For $i = 0 : 1 : l - 1$

$\Delta_i = (l - i)A_i$

End

While $\delta > \eta$

$$A_R(X_k) = A_0X_k^l + A_1X_k^{l-1} + \dots + A_l \\ \Delta(X_k) = \Delta_0X_k^{l-1} + \Delta_1X_k^{l-2} + \dots + \Delta_{l-1} \\ X_{k+1} = X_k - A_R(X_k)(\Delta(X_k))^{-1} \\ \delta = 100. \frac{\|X_{k+1} - X_k\|_2}{\|X_k\|_2} \\ X_k \leftarrow X_{k+1} \\ k \leftarrow k + 1$$

End

3.3 Comments

- A numerical method for solving a given problem is said to be local if it is based on local (simpler) model of the problem around the solution. From this definition, we can see that in order to use a local method, one has to provide an initial approximation of the solution. This initial approximation can be provided by a global method. As shown in Dahimene [[10]], local methods are fast converging while global ones are quite slow. This implies that a good strategy is to start solving the problem by using a global method and then refine the solution by a local method.
- The proposed hybrid or two stage Block-Horner's algorithm converges rapidly as it performs a recursive iteration and is easily implemented in a digital computer. Horner's algorithm could be used for evaluation of solvents of a matrix polynomial, but this method depends largely upon the initial guess even in some cases the initial value of X_k is randomly chosen. Hence in sometimes it is very hard to find suitable solutions. Our $Q.D.$ algorithm is numerically more stable and its initial starting values are well defined and evaluated.
- The complete program starts with the $Q.D.$ algorithm. It is then followed by a refinement of the right factor by Horner's algorithm. After deflation, Horner's algorithm is again applied using the next Q output from the $Q.D.$ algorithm and the process is repeated until we find a linear term. The above process can be applied only to polynomial matrices that satisfy the conditions of theorem (i.e. complete right and left factorization and complete dominance relation between solvents).
- Many research works have been done on the spectral decomposition for matrix polynomials to achieve complete factorization and reconstruction of the block roots using algebraic and geometric numerical approaches, but (to our knowledge) nothing has been done for Block-Horner's algorithm and/or Block- $Q.D.$ algorithm.

4 Application in control engineering

The system under examination is a power plant gas turbine (GE MS9001E) with single shaft, used as an electricity generator, installed in power station unit Sonelgaz at M'SILA, Algeria. The dynamic model of this gas turbine obtained via MIMO Recursive Least square estimator, using experimental inputs/outputs data acquired on-site and the obtained model is of order $n=6$ with two inputs: (Output Pressure Compressor (OPC), and Output Temperature Compressor (OTC)), and two outputs: (Exhaust Temperature and Rotor Speed) [[46]]. In figure 2, the fundamental components of the system under study are given .

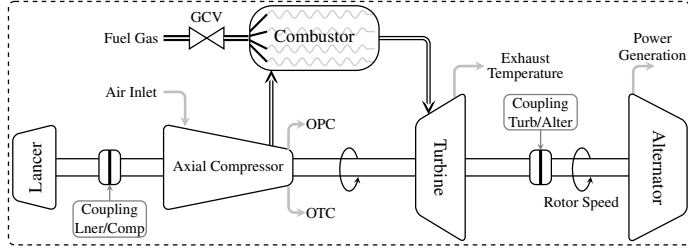


Figure 2: schematic

The dynamic model of this power plant gas turbine is a linear time invariant multi input multi output system, described by a set of high degree coupled vector differential equations with matrix constant coefficients (or a matrix transfer function). In our case the relationship between the input and output is a ratio of two matrix polynomials, expressed as a right (or left) matrix fraction description (RMFD or LMFD):

$$\begin{cases} H(\lambda) = N_R(\lambda)D_R^{-1}(\lambda) \\ = D_L^{-1}(\lambda)N_L(\lambda) \end{cases} \quad (53)$$

where: N_R , D_R , N_L and D_L are matrix polynomials and λ stands for the $\left(\frac{d}{dt}\right)$ operator. see [[4]], [[40]] and [[48]] and the reference therein. The obtained λ -matrix transfer function of the power plant gas turbine system is:

$$H(\lambda) = N(\lambda)D(\lambda)^{-1} = \begin{pmatrix} H_{11}(\lambda) & H_{12}(\lambda) \\ H_{21}(\lambda) & H_{22}(\lambda) \end{pmatrix}$$

Where

$$H_{11}(\lambda) = \left(\frac{34.01\lambda^5 - 116.9\lambda^4 - 221.2\lambda^3 - 102.5\lambda^2 - 1.955 \times 10^4\lambda - 2077}{\lambda^6 - 30.41\lambda^5 - 45.33\lambda^4 + 515.1\lambda^3 - 1343\lambda^2 + 1.805 \times 10^4\lambda + 9102} \right)$$

$$H_{21}(\lambda) = \left(\frac{36.58\lambda^5 - 42.7\lambda^4 + 696\lambda^3 + 3295\lambda^2 - 208.4\lambda + 1.979 \times 10^4}{\lambda^6 - 30.41\lambda^5 - 45.33\lambda^4 + 515.1\lambda^3 - 1343\lambda^2 + 1.805 \times 10^4\lambda + 9102} \right)$$

$$H_{12}(\lambda) = \left(\frac{27.37\lambda^5 - 53.9\lambda^4 + 12.19\lambda^3 - 767.4\lambda^2 - 7918\lambda - 8267}{\lambda^6 - 30.41\lambda^5 - 45.33\lambda^4 + 515.1\lambda^3 - 1343\lambda^2 + 1.805 \times 10^4\lambda + 9102} \right)$$

$$H_{22}(\lambda) = \left(\frac{32.93\lambda^5 - 66.88\lambda^4 + 1.5\lambda^3 - 1474\lambda^2 - 8675\lambda - 1.675 \times 10^4}{\lambda^6 - 30.41\lambda^5 - 45.33\lambda^4 + 515.1\lambda^3 - 1343\lambda^2 + 1.805 \times 10^4\lambda + 9102} \right)$$

We try to decouple the power plant gas turbine dynamic model. Let us first factorize (*decompose*) the numerator matrix polynomial $N(\lambda)$ into a complete set of spectral factors, then we use those block zeros into the denominator $D(\lambda)$

via state feedback. Hence the decoupling objectives are achieved.

Consider the square matrix transfer function:

$$\begin{aligned} H(\lambda) &= N(\lambda)D^{-1}(\lambda) = \left(\sum_{i=0}^k N_i \lambda^i \right) \left(\sum_{i=0}^l D_i \lambda^i \right)^{-1} \\ &= (N_k \lambda^k + \dots + N_1 \lambda + N_0)(D_l \lambda^l + \dots + D_1 \lambda + D_0)^{-1} \end{aligned}$$

with:

$D_l = I$ is an $m \times m$ identity matrix and

$N_i \in R^{m \times m}$, ($i = 0, 1, \dots, k$)

$D_i \in R^{m \times m}$, ($i = 0, 1, \dots, l$), $l > k$

Assume that $N(\lambda)$ can be factorized into k Block zeros and $D(\lambda)$ can be factorized into l Block roots (*using one of the proposed algorithms*):

$$N(\lambda) = N_k(\lambda I - Z_1) \dots (\lambda I - Z_k) \quad (54)$$

$$D(\lambda) = (\lambda I - Q_1) \dots (\lambda I - Q_l). \quad (55)$$

The matrix transfer function can be written : $H(\lambda) = C(\lambda I - A)^{-1}B$. Also via the use of state feedback the control law becomes a state dependent and be rewritten as $u(t) = -K.X(t) + F.r(t)$. Hence we obtain the following closed loop system:

$$(H(\lambda))_{closed} = C(\lambda I - A + BK)^{-1}BF = N(\lambda)D_d^{-1}(\lambda)F$$

where: $D_d(\lambda) = (\lambda I - Q_{d1}) \dots (\lambda I - Q_{dl})$ and Q_{di} : are the desired spectral factors to be placed

$$H(\lambda)_{closed} = N(\lambda)D_d^{-1}(\lambda)F$$

Hence, the closed loop matrix transfer function is of the form:

$$H(\lambda)_{closed} = N_k(\lambda I - Z_1) \dots (\lambda I - Z_k)(\lambda I - Q_{dl})^{-1} \dots (\lambda I - Q_{d1})^{-1}F \quad (56)$$

In order to achieve perfect block decoupling we choose:

$$\begin{aligned} Q_{d1} &= N_k J_1 N_k^{-1}, \dots, Q_{d(l-k)} = N_k J_{(l-k)} N_k^{-1} \\ Q_{d(l-k+1)} &= Z_1, \dots, Q_{dl} = Z_k \\ J_i &= \text{diag}(\lambda_{i1}, \dots, \lambda_{im}), \quad F = (N_k)^{-1} \end{aligned}$$

Now by assigning those block roots the system is decoupled and the closed loop matrix transfer function is:

$$H(\lambda)_{closed} = (\lambda I - J_1)^{-1} \dots (\lambda I - J_{l-k})^{-1} \quad (57)$$

Now we should construct the numerator and denominator matrix polynomials of the gas turbine system from the matrix transfer function (see [[13],[14] and [17]]):

$$\begin{cases} D(\lambda) &= D_3 \lambda^3 + D_2 \lambda^2 + D_1 \lambda + D_0, & D_3 = I_2 \\ N(\lambda) &= N_3 \lambda^3 + N_2 \lambda^2 + N_1 \lambda + N_0, & N_3 = O_2 \end{cases}$$

where:

$$\begin{pmatrix} D_0 \\ D_1 \\ D_2 \end{pmatrix} = \begin{pmatrix} -37.0170 & 28.2888 \\ -223.8750 & -74.7887 \\ 34.8029 & 20.9798 \\ -280.2609 & -216.8345 \\ -14.0378 & -7.5183 \\ -12.3898 & -16.3740 \end{pmatrix}$$

$$\begin{pmatrix} N_0 \\ N_1 \\ N_2 \end{pmatrix} = \begin{pmatrix} 211.7886 & 61.4727 \\ 331.6250 & 199.1758 \\ 100.8960 & 74.4572 \\ 148.1818 & 120.4265 \\ 34.0105 & 27.3669 \\ 36.5764 & 32.9324 \end{pmatrix}$$

Let we decompose the numerator and denominator matrix polynomials and reconstruct their block roots:

$$N(\lambda) = N_2(\lambda I - Z_1)(\lambda I - Z_2)$$

and

$$D(\lambda) = (\lambda I - Q_1)(\lambda I - Q_2)(\lambda I - Q_3)$$

The Block spectral factors are approximately computerized with a residual normed tolerance error given by:

$$\varepsilon_i = \frac{\|Z_i^* - Z_i\|}{\|Z_i^*\|} \quad i = 1, 2,$$

and

$$\xi_i = \frac{\|Q_i^* - Q_i\|}{\|Q_i^*\|} \quad i = 1, 2, 3$$

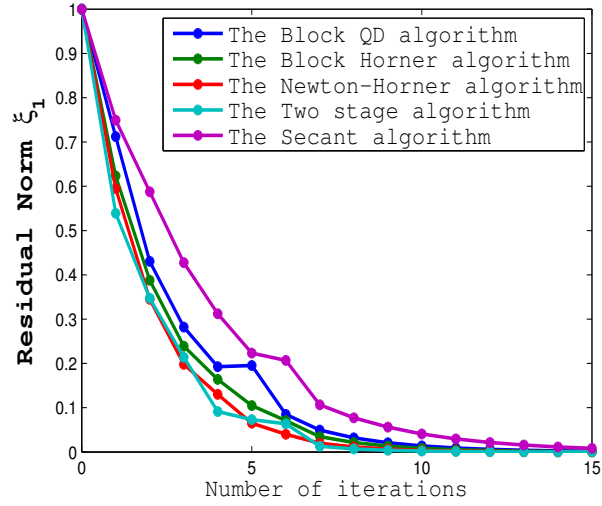
Remark 4.1. *The last Block pole Q_3 can be constructed using the synthetic long division. Figure (3) illustrates a comparison between the proposed algorithms in term of the convergence speed and residual normed tolerance error.*

The numerator block zeros are computed using the proposed algorithms compared to recent developed method called the generalized secant method [[49]] which can factorize matrix polynomial into a complete set of block roots. Numerical results of the developed procedures as illustrated in [[49]] give:

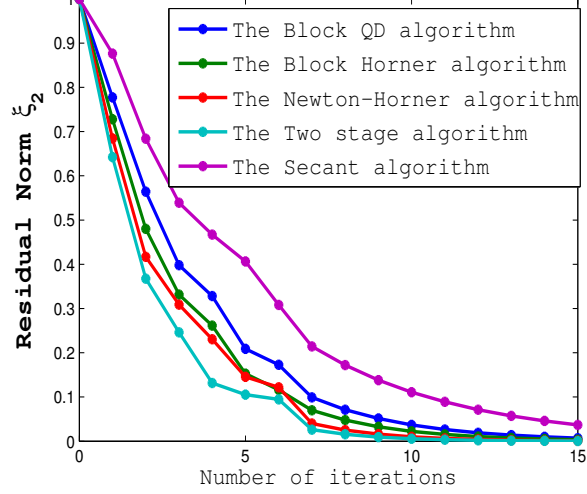
$$N(Z_i) = O_2 \Rightarrow$$

$$Z_1 = \begin{pmatrix} 24.7235 & 23.1394 \\ -27.4494 & -24.9281 \end{pmatrix}, \quad Z_2 = \begin{pmatrix} -18.5711 & -16.0841 \\ 16.1166 & 13.4353 \end{pmatrix}$$

1st Block Pole Q_1



2nd Block Pole Q_2



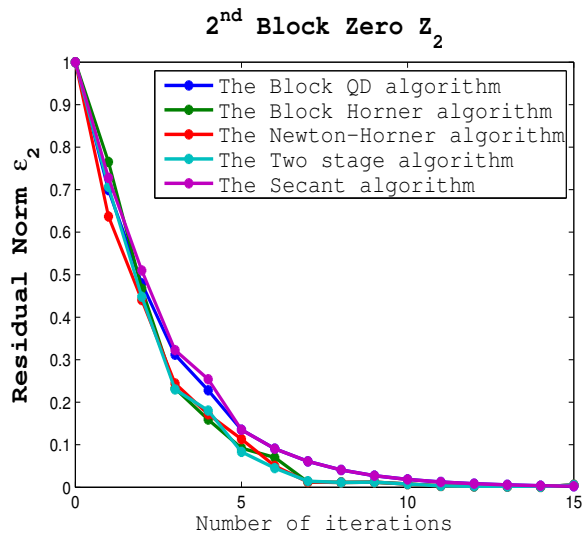
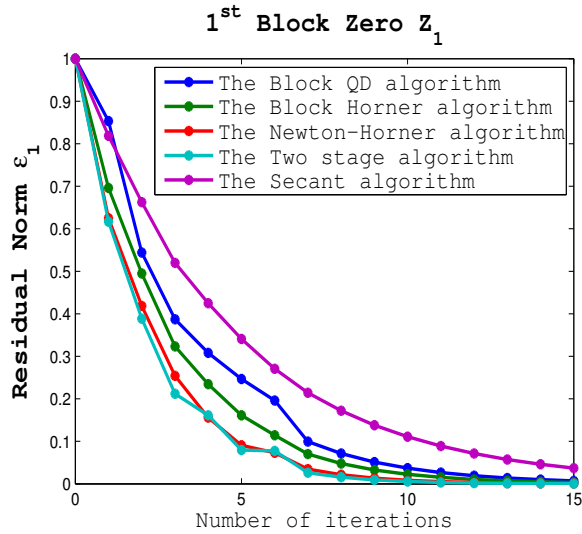


Figure 3: The Residual Error Norm comparison study

Although the Block Newton method aims to improve the convergence speed over the Block Horner method, it cannot always achieve this goal. The Newton-Horner's method converges quadratically due to its conformity to Newton method. As a consequence, the number of significant values is roughly doubled every iteration, provided that X_i is close to the root ($\varepsilon = 0.521 \times 10^{-4}$). The two stage algorithm is the best method of finding roots, it is simple and fast (at first seven iterations the average error becomes $\varepsilon = 0.213 \times 10^{-3}$ and $\xi = 0.341 \times 10^{-3}$).

The only drawback of the two stage method is that it uses the matrix inversion, and partially is dependent on the initial guess. Our refinement algorithm avoided this obstacle. As indicated in figure (2) the Q-D algorithm converges, but it is of global nature with no initial independence. The global convergence characteristics of the secant method are poor, as indicated in this figure.

The desired denominator is of third order written in the form:

$$D_d(\lambda) = D_{d3}\lambda^3 + D_{d2}\lambda^2 + D_{d1}\lambda + D_{d0}$$

Using the prescribed decoupling algorithm we obtain:

$$F = N_2^{-1}, J_1 = \begin{pmatrix} -1 & 0 \\ 0 & -2 \end{pmatrix}$$

$$Q_{d1} = N_2^{-1}J_1N_2^{-1}, Q_{d2} = Z_1, Q_{d3} = Z_2$$

$$\begin{aligned} D_d(\lambda) &= (\lambda I - Q_{d1})(\lambda I - Q_{d2})(\lambda I - Q_{d3}) \\ &= I\lambda^3 + D_{d2}\lambda^2 + D_{d1}\lambda + D_{d0} \end{aligned}$$

Where:

$$\begin{aligned} D_{d2} &= -(Q_{d1} + Q_{d2} + Q_{d3}) \\ &= \begin{pmatrix} -13.5596 & -14.6249 \\ 21.7809 & 21.8999 \end{pmatrix} \end{aligned}$$

$$\begin{aligned} D_{d1} &= (Q_{d1}Q_{d2} + Q_{d1}Q_{d3} + Q_{d2}Q_{d3}) \\ &= \begin{pmatrix} -126.4282 & -121.5061 \\ 161.6710 & 152.4741 \end{pmatrix} \end{aligned}$$

$$\begin{aligned} D_{d0} &= -(Q_{d1}Q_{d2}Q_{d3}) \\ &= \begin{pmatrix} -178.9732 & -164.0512 \\ 223.2851 & 202.6227 \end{pmatrix} \end{aligned}$$

The state feedback gain matrix of the Block controller form is obtained by see [[47]],[[48]]:

$$K_{ci} = D_{di} - D_i \text{ With } i = 0, 1, 2 \text{ and } K_c = [K_{c0}, K_{c1}, K_{c2}]$$

Now let we go back to original base by similarity transformation T_c as found in [[4]],[[40]] and [[48]]:

$$\begin{aligned} K &= K_c T_c \\ &= \begin{pmatrix} -0.7616 & -3.2690 & 3.5737 & -0.0716 & -2.3462 & 1.4801 \\ 3.0439 & 5.4047 & -2.3853 & 2.4117 & 4.2560 & 0.0494 \end{pmatrix} \end{aligned}$$

The new model of the decoupled system after state feedback is:

$$A_d = (A - BK), \quad B_d = BF, \quad \text{and} \quad C_d = C$$

$$\begin{aligned}
H(\lambda)_{closed} &= C(\lambda I - A + BK)^{-1}BF \\
&= N(\lambda)D_d^{-1}(\lambda)F = \begin{pmatrix} \frac{1}{\lambda+1} & 0 \\ 0 & \frac{1}{\lambda+2} \end{pmatrix}
\end{aligned}$$

Based on the results we deduce that the Block roots are well computed, both numerator and denominator matrix polynomials ($N(\lambda)$ and $D(\lambda)$) are perfectly decomposed using the proposed procedure.

4.1 Suggestions for further research

The results obtained during this research work arose many questions and problems which are subject for future research

- Finding other globalization techniques for the Block-Horner's algorithm to avoid the local restriction and the problem of initial guess, so to arrive at very fast global nested program. Also exploring and extending other scalar numerical methods to factorize matrix polynomials.
- Both of The Block-Horner's algorithm and the Block- $Q.D.$ algorithm as used in our work converges to factors of a matrix polynomial. By using the defined similarity transformations, we can derive the solvents. However, it would be convenient to have a global algorithm that converges rapidly and directly to all solvents.
- If a column in the $Q.D.$ tableau converges, it implies that there exists a factorization of the matrix polynomial that splits the spectrum into a dominant set and a dominated one. If the system under consideration is a digital system, we know that the largest modulus latent roots affect the dynamic properties of the system. In such case, the $Q.D.$ algorithm can become a tool for system reduction (using the dominant mode concept).
- The computational procedure for finding the solvents of a matrix polynomial with repeated block roots (solvents) and/or spectral factors need to be investigated further.

5 Conclusion

In this paper we have introduced new numerical approaches for determining the complete sets of spectral factors and solvents of a monic matrix polynomial. For avoiding the initial guess we have proposed a systematic method for the Block-Horner's algorithm via a refinement of the Block- $Q.D.$ algorithm. At least three advantages are offered by the proposed technique: (i) an algorithm with global nature is obtained; hence there is no initial-guess problem during the whole procedure, (ii) high speed convergence to each solution and only a few iterations are required (iii) via the help of refinement and direct cascading, the algorithms are easily coupled together and the whole scheme is suitable for

programming in a digital computer digital. The obtained solvents can be considered as a useful tool for carrying out the block partial fraction expansion for the inverse of a matrix polynomial. Those partial fractions are matrix transfer functions of reduced order linear systems such that the realization of them leads to block diagonal (block-decoupling) or parallel decomposed multivariable linear time invariant system. The dynamic properties of MIMO systems depend on block pole of its characteristic matrix polynomial. Therefore they can be used as tools for block-pole placement, block-system identification and block-model order reduction. In addition, the proposed method can be employed to carry out the block spectral factorization of a matrix polynomial for problems in optimal control, filtering and estimation.

Acknowledgement

Dr. George F. Fragulis work is supported by the program Rescom 80159 of the Univ. of Applied Science of Western Macedonia, Hellas.

References

- [1] J. J. DiStefano, A. R. Stubberud, I. J. Williams, *Theory and Problems of Feedback and Control Systems*, Mc. Graw Hill, 1967.
- [2] B. N. Parlett, *The LU and QR Algorithms*, in Ralston and Wilf, *Mathematical Methods for Digital Computers*, Vol. 2, John Wiley, 1967.
- [3] Bachir Nail, Abdellah Kouzou, Ahmed Hafaifa, *Parametric Identification of Multivariable Industrial System Using Left Matrix Fraction Description*, J. Automation and Systems Engineering 10-4 (2016): 221-230
- [4] Malika Yaici, Kamel Hariche, *On eigenstructure assignment using block poles placement*, European Journal of Control, May 2014.
- [5] Malika Yaici, Kamel Hariche, Clark Tim, *A Contribution to the Polynomial Eigen Problem*, International Journal of Mathematics, Computational, Natural and Physical Engineering, Vol:8 No 10, 2014.
- [6] Nicholas J. Higham, *Function of Matrices: theory and computation*, SIAM 2008.
- [7] George F. Fragulis *Transformation of a PMD into an implicit system using minimal realizations of its transfer function matrix in terms of finite and infinite spectral data*, Journal of the Franklin Institute, vol.333, pages 41-56, 1996.
- [8] George F. Fragulis *Generalized Cayley-Hamilton theorem for polynomial matrices with arbitrary degree*, International Journal of Control, vol.62, pages 1341-1349, 1995.

- [9] B.G. Mertzios and George F. Fragulis *The fundamental matrix in ARMA systems*, IEEE International Conference on Industrial Technology, pages 1541-1542, 2004.
- [10] A.Dahimene,*Algorithms for the factorization of matrix polynomials*, Master Thesis INELEC , 1992.
- [11] A.Dahimene, *Incomplete matrix partial fraction expansion*, Control and Cybernetics vol. 38 (2009) No.3
- [12] S. M. Ahn, *Stability of a Matrix Polynomial in Discrete Systems*, IEEE trans. on Auto. Contr., Vol. AC-27, pp. 1122-1124, Oct. 1982.
- [13] C. T. Chen, *Linear System Theory and Design*, Holt, Reinhart and Winston, 1984.
- [14] T. Kailath, W. Li, *Linear Systems*,Prentice Hall, 1980.
- [15] V. Kucera, *Discrete Linear Control: The Polynomial Equation Approach*,John Wiley, 1979.
- [16] P. Resende, E. Kaskurewicz, *A Sufficient Condition for the Stability of Matrix Polynomials*,IEEE trans, on. Auto. Contr., Vol. AC-34, pp. 539-541, May 1989.
- [17] Leang S. Shieh, F. R Chang, B. C. Mcinnis, *The Block partial fraction expansion of a matrix fraction description with repeated block poles*, IEEE Trans. Auto. Cont. AC-31, pp. 236-239, 1986.
- [18] E. D. Denman and A. N. Beavers, *The matrix sign function and computations in systems*,Appl. Math. Comput. 2:63-94 (1976).
- [19] M. K. Solak, *Divisors of Polynomial Matrices: Theory and Applications*,IEEE trans. on Auto. Contr., Vol. AC-32, pp. 916-919, Oct. 1987.
- [20] J. E. Dennis, J. F. Traub, and R. P. Weber, *Algorithms for solvents of matrix polynomials*,SZAhJ. Numer. Anal. 15:523-533 (1978).
- [21] E. D. Denman, *Matrix polynomials, roots, and spectral factors*, A & Math. Comput. 3:359-368, (1977).
- [22] L. S. Shieh, S. Sacheti, *A Matrix in the Block Schwarz Form and the Stability of Matrix Polynomials*,Int. J. Control, Vol. 27, pp. 245-259, 1978.
- [23] L. S. Shieh, Y. T. Tsay, and N. I. Coleman, *Algorithms for solvents and spectral factors of matrix polynomials*,Internat. 1. Control 12:1303-1316 (1981).
- [24] L. S. Shieh and Y. T. Tsay, *Transformation of solvent and spectral factors of matrix polynomial, and their applications*,Internat. J. Control 34:813-823 (1981).

- [25] J. S. H. Tsai, I. S. Shieh, and T. T. C. Shen, *Block power method for computing solvents and spectral factors of matrix polynomials*, Internut. Computers and Math. Appl. 16:683-699 (1988).
- [26] P. Henrici, *The Quotient-Difference Algorithm*, Nat. Bur. Standards Appl. Math. Series, Vol. 49, pp. 23-46, 1958.
- [27] L. S. Shieh, F. R. Chang, and B. C. Mcinnis, *The block partial fraction expansion of a matrix fraction description with repeated block poles*, IEEE Trans. Automut. Control. 31:23-36 (1986).
- [28] J. E. Dennis, J. F. Traub, and R. P. Weber, *The algebraic theory of matrix polynomials*, SZAIZI J. Numer. Anal. 13:831-845 (1976).
- [29] L. S. Shieh and Y. T. Tsay, *Block modal matrices and their applications to multivariable control systems*, IEE Proc. D, Control Theory Appl. 2:41-48(1982).
- [30] L. S. Shieh and N. Chahin, *A computer-aided method for the factorization of matrix polynomials*, Internat. Systems Sci. 12:1303-1316 (1981).
- [31] J. S. H. Tsai and C. M. Chen and L. S. Shieh, *A Computer-Aided Method for Solvents and Spectral Factors of Matrix Polynomials*, Applied mathematics and computation 47:211-235 (1992).
- [32] K. Hariche, *Interpolation Theory in the Structural Analysis of λ -matrices*: Chapter3, Ph. D. Dissertation, University of Houston, 1987.
- [33] R. L. Burden and J. D. Faires, *Numerical Analysis*, 8th Edition, Thomson Brooks/Cole, Belmont, CA, USA, 2005.
- [34] L. S. Shieh and Y. T. Tsay, *Transformation of a class of multivariable control systems to block companion forms*, IEEE Truns. Autonwt. Control 27: 199-203 (1982).
- [35] K. Hariche, E. D. Denman, *interpolation Theory and λ -Matrices*, journal of mathematical analysis and applications 143, 53 and 547 (1989).
- [36] K. Hariche and E. D. Denman, *On Solvents and Lagrange Interpolating λ -Matrices*, applied mathematics and computation 25321-332 (1988).
- [37] E. Periera, *On solvents of matrix polynomials*, Appl. Numer. Math., vol. 47, pp. 197-208, 2003.
- [38] E. Periera, *Block eigenvalues and solution of differential matrix equation*, mathematical notes, Miskolc, Vol. 4, No.1 (2003), pp. 45-51.
- [39] L. S. Shieh and Y. T. Tsay, *Algebraic-geometric approach for the modal reduction of large-scale multivariable systems*, IEE Proc. D Control Theory Appl. 131(1):23-26 (1984).

- [40] Malika Yaici, Kamel Hariche, *On Solvents of Matrix Polynomials*, International Journal of Modeling and Optimization, Vol. 4, No. 4, August 2014.
- [41] I. Gohberg, P. Lancaster, L. Rodman, *Matrix Polynomials*, Academic Press, 1982.
- [42] A. Pathan and T. Collyer, *The wonder of Horner's method*, Mathematical Gazette, 87 (2003), No. 509, 230-242.
- [43] H. Zabot, K. Hariche, *On solvents-based model reduction of MIMO systems*, International Journal of Systems Science, 1997, volume 28, number 5, pages 499-505.
- [44] S. Barnett, *Matrices in Control Theory*, Van Nostrand Reinhold, New York, 1971.
- [45] I. Gohberg, M. A. Kaashoek, and L. Rodman, *Spectral analysis of operator polynomial and a generalized Vandermonde matrix*, 1. the finite-dimensional case, in Topics in Functional Analysis, Academic, 1978, pp. 91-128.
- [46] B. Bekhiti, A. Dahimene, B. Nail, K. Hariche, and A. Hamadouche, *On Block Roots of Matrix Polynomials Based MIMO Control System Design*, 4th International Conference on Electrical Engineering ICEE Boumerdes, 2015.
- [47] Carl D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM 2000.
- [48] Yih T. Tsay and Leang S. Shieh *Block decompositions and block modal controls of multivariable control systems*, Automatica, Vol. 19, No.1, pp. 29-40, 1983.
- [49] Marlliny Monsalve and Marcos Raydan, *Newton's method and secant methods: A longstanding relationship from vectors to matrices* Portugaliae Mathematica European Mathematical Society, Vol. 68, Fasc. 4, 2011, 431475 6 DOI 10.4171/PM/1901