2

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 17636.13-MA	2. GOVT ACCESSION NO. AD-A115828	3. RECIPIENT'S CATALOG NUMBER N/A
4. TITLE (and Subtitle) Algorithms for Solvents and Spectral Factors of Matrix Polynomials		5. TYPE OF REPORT & PERIOD COVERED Reprint
		6. PERFORMING ORG. REPORT NUMBER N/A
7. AUTHOR(s) Leang S. Shieh Yih T. Tsay Norman P. Coleman		8. CONTRACT OR GRANT NUMBER(s) DAAG29 80 K 0077
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Houston Houston, TX 77004		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS N/A
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Research Office P. O. Box 12211 Research Triangle Park, NC 27709		12. REPORT DATE 1981
		13. NUMBER OF PAGES 14
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Submitted for announcement only.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)		

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ARO 17636.13-MA

INT. J. SYSTEMS SCI., 1981, VOL. 12, NO. 11, 1303–1316

Algorithms for solvents and spectral factors of matrix polynomials

LEANG S. SHIEH†, YIH T. TSAY† and NORMAN P. COLEMAN‡

A generalized Newton method, based on the contracted gradient of a matrix polynomial, is derived for solving the right (left) solvents and spectral factors of matrix polynomials. Two methods of selecting initial estimates for rapid convergence of the newly developed numerical method are proposed. Also, new algorithms for solving complete sets of the right (left) solvents and spectral factors without directly using the eigenvalues of matrix polynomials are derived. The proposed computer-aided method can be used to determine the spectral factorization of a matrix polynomial for optimal control, filtering and estimation problems.

1. Introduction

Linear time-invariant multi-input multi-output systems are often described by a set of coupled high-degree multivariable differential equations. The Laplace transform descriptions of such systems are the matrix fraction descriptions or polynomial matrix descriptions (Kailath 1980). The non-singular denominator matrix polynomial of the matrix fraction description, called a lambda (λ) matrix (Frazer *et al.* 1952, Lancaster 1966), characterizes the properties of a multivariable (multi-input–output) system. The stability property of a λ-matrix has been investigated by Lancaster (1966), Jury (1974), Anderson and Bitmead (1977), Papaconstantinou (1975), Shieh *et al.* (1978), Shieh and Sacheti (1978), and Shieh and Tajvari (1980). Recently, the algebraic theory of λ-matrices has been investigated by Dennis *et al.* (1976), Gohberg *et al.* (1978), Markus and Mereuca (1978), and Tsay *et al.* (1981). Also, numerical methods for solving solvents and spectral factors of a matrix polynomial have been proposed by Dennis *et al.* (1978) and Shieh and Chahin (1981).

In this paper, a new numerical method, based on the contracted gradient of a matrix polynomial, is proposed for solving a complete set of right (left) solvents and spectral factors of a λ-matrix. The proposed method can be applied to determine the spectral factorization of a matrix polynomial for optimal control, filtering and estimation problems.

2. Solvents and spectral factorization of λ-matrices

An nth-degree mth-order monic λ-matrix is defined as

$$A(\lambda)=\sum_{i=0}^{n} A_i\lambda^{n-i} \tag{1}$$

where A_i, $i=1, \ldots, n$ are $m\times m$ complex matrices and $A_0(=I_m)$ is an $m\times m$

Received 25 June 1981.

† Department of Electrical Engineering, University of Houston, Houston, Texas 77004, U.S.A.

‡ U.S. Army Armament Research and Development Command, Dover, N.J. 07801, U.S.A.

0020-7721/81/1211 1303 $02·00 © 1981 Taylor & Francis Ltd

identity matrix. λ is a complex variable. Let X be an $m \times m$ complex matrix. Two matrix polynomials, defined by

$$A_{\mathrm{R}}(X) = \sum_{i=0}^{n} A_i X^{n-i} \tag{2}$$

and

$$A_{\mathrm{L}}(X) = \sum_{i=0}^{n} X^{n-i} A_i \tag{3}$$

are referred to as the right and left matrix polynomials associated with λ-matrix $A(\lambda)$, respectively. A right solvent R of $A(\lambda)$ is defined by

$$A_{\mathrm{R}}(R) = \sum_{i=0}^{n} A_i R^{n-i} = 0_m \tag{4}$$

and the left solvent L of $A(\lambda)$ is defined by

$$A_{\mathrm{L}}(L) = \sum_{i=0}^{n} L^{n-i} A_i = 0_m \tag{5}$$

where 0_m is an $m \times m$ null matrix. The right (left) solvents play an important role in the analysis of the λ-matrix. An important set of solvents is the complete set of regular solvents (Gohberg *et al.* 1978, Markus and Mereuca 1978). The complete regular right solvents R_i, $i = 1, \dots, n$ have the following properties :

$$\rho(R_i) \cap \rho(R_j) = \emptyset, \quad i \neq j, \quad i, j = 1, \dots, n \tag{6 a}$$

and

$$\bigcup_{i=1}^{n} \rho(R_i) = \rho(A(\lambda)) \tag{6 b}$$

where $\rho(R_i)$ is the spectrum of R_i and $\rho(A(\lambda))$ is the spectrum of $A(\lambda)$.

Similarly, a complete set of regular left solvents L_i, $i = 1, \dots, n$ has the following properties :

$$\rho(L_i) \cap \rho(L_j) = \emptyset, \quad i \neq j, \quad i, j = 1, \dots, n \tag{7 a}$$

$$\bigcup_{i=1}^{n} \rho(L_i) = \rho(A(\lambda)) \tag{7 b}$$

The existence and uniqueness of the complete set of solvents have been investigated by Lancaster (1966), Dennis *et al.* (1976), Gohberg *et al.* (1978) and Markus and Mereuca (1978).

If a monic λ-matrix can be decomposed into the product of first-degree linear λ-matrices :

$$A(\lambda) = (\lambda I_m - S_n)(\lambda I_m - S_{n-1}) \dots (\lambda I_m - S_1) \tag{8 a}$$

then $(\lambda I_m - S_i)$, $i = 1, \dots, n$, are referred to as the complete set of linear spectral factors. The $m \times m$ complex matrices S_i, $i = 1, \dots, n$, are the spectral factors of the λ-matrix $A(\lambda)$. Notice that S_1 is a right solvent of $A(\lambda)$, whereas S_n is the left solvent of $A(\lambda)$. Other spectral factors are, in general, not the right or left solvents of $A(\lambda)$.

Since S_1 is a right solvent ($\triangleq R_1$) of $A(\lambda)$, a deflated λ-matrix can be obtained by factoring out the linear spectral factor $(\lambda I_m - R_1)$ from $A(\lambda)$, as follows :

$$A^{(1)}(\lambda) \triangleq A(\lambda)(\lambda I_m - R_1)^{-1} = (\lambda I_m - S_n) \dots (\lambda I_m - S_2) \tag{8 b}$$

Determining a right solvent ($\triangleq R_2^{(1)}$) of $A^{(1)}(\lambda)$ yields the spectral factor S_2 in (8 *b*). By continuing the same procedure, we can determine all spectral factors of $A(\lambda)$ in (8 *a*). Notice that $R_2^{(1)}$ is a right solvent of $A^{(1)}(\lambda)$ but not that of $A(\lambda)$. In the same fashion, the spectral factors of $A(\lambda)$ can be computed from the successive left solvents of $A(\lambda)$ and the deflated $A(\lambda)$.

In optimal control and estimation problems, $A(\lambda)$ is an even degree real λ-matrix with the property

$$A(\lambda) = A^{\mathrm{T}}(-\lambda) \tag{9 a}$$

where T designates transpose. The spectral factorization (Youla 1961, Anderson 1967) of $A(\lambda)$ is described by

$$A(\lambda) = A_1(\lambda) A_1^{\mathrm{T}}(-\lambda) \tag{9 b}$$

where $A_1(\lambda)$ contains all latent roots of $A(\lambda)$ in the LHP of the complex plane and $A_1(-\lambda)$ in the RHP of the complex plane. To determine $A_1(\lambda)$ the procedure for obtaining the spectral factors of $A(\lambda)$ in (8 *a*) can be applied. However, by using the property shown in (9 *a*), the computations can be simplified. In other words, if R is a right solvent of $A(\lambda)$ in (9) with all eigenvalues in the RHP, then

$$A(\lambda) = A^{(1)}(\lambda)(\lambda I_m - R) \tag{10 a}$$

where $A^{(1)}(\lambda)$ is an $(n-1)$ degree λ-matrix. From the property of (9 *b*) we have

$$A(\lambda) = A^{\mathrm{T}}(-\lambda) = (-\lambda I_m - R^{\mathrm{T}}) A^{(1)\mathrm{T}}(-\lambda) \tag{10 b}$$

From (10 *b*) we observe that $-R^{\mathrm{T}}$ is a left solvent of $A(\lambda)$. Thus, the factorization of $A(\lambda)$ in (9 *b*) can be simplified by factoring out $(\lambda I_m - R)$ and $(\lambda I_m + R^{\mathrm{T}})$ from the right and left of $A(\lambda)$, respectively, if a right solvent R is found. The deflated $A(\lambda)$ becomes

$$A^{(2)}(\lambda) = (\lambda I_m + R^{\mathrm{T}})^{-1} A(\lambda)(\lambda I_m - R)^{-1} \tag{10 c}$$

where $A^{(2)}(\lambda)$ is an $(n-2)$ degree λ-matrix.

Repeating the same procedure for (10 *c*) gives the desired spectral factors. Thus, the spectral factorization of $A(\lambda)$ can be determined from the obtained spectral factors.

In this paper, a new numerical method for solving the right (left) solvents $R_i(L_i)$ and spectral factors (S_i) of $A(\lambda)$ is derived by using the contracted gradient of a matrix polynomial.

3. Gradient of a matrix polynomial

The gradient of an nth-degree mth-order matrix polynomial is defined as a fourth-order tensor $\nabla A(X)$ with components (Vetter 1973, Fong 1971)

$$\{\nabla A(X)\}_{i,j,k,l} = \frac{\partial}{\partial X_{k,l}} \{A(X)\}_{i,j} \, ; \quad i, j, k, l = 1, \dots, m \tag{11}$$

where $X_{k,l}$ denotes the (k, l) element of X, $\{A(X)\}_{i,j}$ designates the (i, j) element of $A(X)$ and $\{\nabla A(X)\}_{i,j,k,l}$ denotes the (i, j, k, l) element of $\nabla A(X)$.

Direct use of the gradient for the purposes of this paper involves the inversion of a fourth-order tensor, thus causing computational difficulty. A contraction operation (Fong 1971) on $\nabla A(X)$ with respect to an arbitrary $m \times m$ square matrix Y is defined as

$$\{\nabla A(X)Y\}_{i,j} = \sum_{k=1}^{m} \sum_{l=1}^{m} \{\nabla A(X)\}_{i,j,k,l} Y_{k,l}; \quad i, j = 1, \ldots, m \tag{12 a}$$

from which it follows that

$$\nabla A(X)Y = \frac{d}{d\eta}(A(X+\eta Y))|_{\eta=0} \tag{12 b}$$

where η is a scalar variable. Having the contraction operation defined in (12), the contracted gradient $\nabla A(X)Y$ becomes an $m \times m$ matrix. Thus, computational algorithms can be easily derived and the computational difficulty can be greatly reduced. The contraction operation in (12) can also be extended to the right (left) matrix polynomial in (2) and (3), as follows.

Given a λ-matrix in (1) and the right matrix polynomial in (2), the contraction operation on $\nabla A_R(X)$ with respect to a matrix Y is

$$\begin{aligned}\nabla A_R(X)Y &= \frac{d}{d\eta}(A_R(X+\eta Y))|_{\eta=0} \\ &= \sum_{i=0}^{n} \frac{d}{d\eta}(A_i(X+\eta Y)^{n-i})|_{\eta=0}\end{aligned} \tag{13 a}$$

Each term inside the summation of (13 a) can be computed as

$$\frac{d}{d\eta}(A_i(X+\eta Y)^r)|_{\eta=0} = A_i \sum_{q=0}^{r-1} X^q Y X^{r-q-1} \tag{13 b}$$

Substituting (13 b) into (13 a) and rearranging the indexes gives

$$\nabla A_R(X)Y = \sum_{i=0}^{n-1} A_i \sum_{q=0}^{n-i-1} X^q Y X^{n-i-q-1} \tag{14}$$

Performing index transformations or letting $k = i+q+1$ and $j = i$, (14) becomes

$$\nabla A_R(X)Y = \sum_{k=1}^{n} B_{kR}(X) Y X^{n-k} \tag{15 a}$$

where $B_{kR}(X)$ is the right matrix polynomial of the following λ-matrix $B_k(\lambda)$:

$$B_k(\lambda) = \sum_{j=0}^{k-1} A_j \lambda^{k-j-1} \tag{15 b}$$

Similarly, for the left matrix polynomial $A_L(X)$, we have

$$\nabla A_L(X)Y = \sum_{k=1}^{n} X^{n-k} Y B_{kL}(X) \tag{16}$$

where $B_{kL}(X)$ is the left matrix polynomial of the λ-matrix $B_k(\lambda)$ defined in (15 b).

4. A generalized Newton method for solving solvents

The right matrix polynomial $A_R(X)$ can be expanded around an $m \times m$ matrix X_0 as

$$A_R(X) = A_R(X_0) + \nabla A_R(X_0)(X - X_0) + O((X - X_0)) \tag{17}$$

where $O((X - X_0))$ is a matrix polynomial with high-degree terms of $(X - X_0)$, and $\nabla A_R(X_0)(X - X_0)$ is a contracted gradient of dimension $m \times m$.

The first-degree approximation of (17) with $\|\Delta X\| < 1$ becomes

$$A_R(X) \cong A_R(X_0) + \nabla A_R(X_0)\Delta X \tag{18}$$

where $\Delta X \triangleq X - X_0$.

Define a recursive formula

$$X_{i+1} = X_i + \Delta X_{i+1} \tag{19 a}$$

Thus, (18) becomes

$$A_R(X_{i+1}) \cong A_R(X_i) + \nabla A_R(X_i)\Delta X_{i+1} \tag{19 b}$$

If X_{i+1} is the right solvent of $A(\lambda)$, or $A_R(X_{i+1}) = 0_m$, then

$$A_R(X_i) + \nabla A_R(X_i)\Delta X_{i+1} = 0_m \tag{20}$$

Solving ΔX_{i+1} from (20) and substituting it into (19 *a*) gives the recursive formula for solving the right solvent of $A(\lambda)$. To solve the ΔX_{i+1} we use the contracted gradient derived in (15), as follows.

From (15) we have

$$\nabla A_R(X_i)\Delta X_{i+1} = \sum_{k=1}^{n} B_{kR}(X_i)\Delta X_{i+1}X_i^{n-k} \tag{21}$$

Substituting (21) into (20) yields

$$\sum_{k=1}^{n} B_{kR}(X_i)\Delta X_{i+1}X_i^{n-k} = -A_R(X_i) \tag{22}$$

ΔX_{i+1} can be solved by using Lancaster's Kronecker-product expansion method (Tsay *et al.* 1981, Lancaster 1970) as follows:

$$\text{Vec}\,\{\Delta X_{i+1}\} = -G(X_i)^{-1}\,\text{Vec}\,\{A_R(X_i)\} \tag{23 a}$$

where

$$G(X_i) = \sum_{k=1}^{m} (X_i^{n-k})^T \otimes B_{kR}(X_i) \tag{23 b}$$

$\otimes$ designates the Kronecker product (Barnett 1971) and Vec $\{M\}$ is a column vector consisting of the column vectors M_j of an $m \times m$ matrix M or

$$\text{Vec}\,\{M\} \triangleq [M_1^T, M_2^T, \ldots, M_m^T]^T \tag{23 c}$$

Note that the dimension of $G(X_i)$ is $m^2 \times m^2$. $B_{kR}(X_i)$ is a constant matrix of the right matrix polynomial of the λ-matrix $B_k(\lambda)$ shown in (15 *b*) evaluated at X_0.

Similarly, the recursive formula for solving the left solvents of $A(\lambda)$ is

$$X_{i+1} = X_i + \Delta X_{i+1} \tag{24}$$

where ΔX_{i+1} is the solution of the following linear matrix equation :

$$\nabla A_{\mathrm{L}}(X_i)\Delta X_{i+1} = \sum_{k=1}^{n} X_i^{n-k}\,\Delta X_{i+1} B_{k\mathrm{L}}(X_i) = -A_{\mathrm{L}}(X_i) \tag{25}$$

or

$$\mathrm{Vec}\,\{\Delta X_{i+1}\} = -H(X_i)^{-1}\,\mathrm{Vec}\,\{A_{\mathrm{L}}(X_i)\} \tag{26 a}$$

where

$$H(X_i) = \sum_{k=1}^{m} B_{k\mathrm{L}}(X_i)^{\mathrm{T}} \otimes X_i^{n-k} \tag{26 b}$$

The convergent criterion in both (19) and (24) is

$$\|\Delta X_{i+1}\| < \epsilon \tag{27}$$

where ϵ is an assigned small value.

Note that when $m=1$, $A(\lambda)$ in (1) becomes a scalar λ-polynomial. The recursive algorithm in (19) and (24) is the well-known Newton method. Thus, the proposed algorithm in this section is a generalized Newton method.

5. Algorithms for solving complete sets of solvents

To construct a similarity block transformation (Dennis *et al.* 1976, Tsay *et al.* 1981) for the analysis of a large-scale multivariable system, it is necessary to determine a complete set of solvents.

Given a λ-matrix $A(\lambda)$ in (1), if a right solvent R is obtained, then $A(\lambda)$ can be described as

$$A(\lambda) = B(\lambda)(\lambda I_m - R_1) \tag{28 a}$$

where

$$B(\lambda) = \sum_{i=0}^{n-1} B_i \lambda^{n-1-i} \tag{28 b}$$

and the coefficients B_i can be determined by

$$B_i = A_i + B_{i-1}R_1, \quad i=1, \ldots, n-1 \tag{28 c}$$

The right solvent of $B(\lambda)$ is a spectral factor of $A(\lambda)$ but in general not the right solvent of $A(\lambda)$. To resolve this problem, we have to determine the left solvent of $A(\lambda)$ associated with R_1 by using the following relationship (Tsay *et at.* 1981) :

$$L_1 = Q_1^{-1} R_1 Q_1 \quad \text{if rank } Q_1 = m \tag{29}$$

where Q_1 is the solution of the following linear matrix equation (Tsay *et al.* 1981, Lancaster 1970) :

$$\sum_{i=0}^{n-1} R_1^{n-1-i} Q_1 B_i = I_m \tag{30}$$

or

$$\mathrm{Vec}\,\{Q_1\} = \left[\sum_{i=0}^{n-1} B_i^{\mathrm{T}} \otimes (R_1^{n-i-1})\right]^{-1} \mathrm{Vec}\,\{I_m\} \tag{31}$$

Equations (29) and (30) can be easily verified by substituting L_1 of (29) into the left matrix polynomial of $A(\lambda)$ in (28), or $A_L(L_1)$, and using the result of (30). Also, following the similar formulation in (25) and solution in (26), we can solve (30) for Q_1, which is shown in (31). Once L_1 is found, $A(\lambda)$ can be expressed by

$$A(\lambda)=(\lambda I_m-L_1)A^{(1)}(\lambda)=\lambda A^{(1)}(\lambda)-L_1A^{(1)}(\lambda) \qquad (32\,a)$$

where

$$A^{(1)}(\lambda)=\sum_{i=0}^{n-1} A_i^{(1)}\,\lambda^{n-1-i} \qquad (32\,b)$$

and

$$A_i^{(1)}=A_i+L_1A_{i-1}^{(1)}, \quad i=1,2,\ldots,n-1 \qquad (32\,c)$$

Assuming that R_2 is a right solvent of $A^{(1)}(\lambda)$, or $A^{(1)}(R_2)=0_m$, then from (32 *a*) we have the right matrix polynomial

$$A_R(R_2)=A_R^{(1)}(R_2)R_2-L_1A_R^{(1)}(R_2)=0_m \qquad (33)$$

Therefore, a right solvent of $A^{(1)}(\lambda)$ is a right solvent of $A(\lambda)$. Repeating the same procedures, we can determine the complete set of right solvents. The algorithm can be summarized as follows.

Algorithm 1—*Complete set of right solvents*

(1) Set an index $i=1$.

(2) Find a right solvent R_i of $A(\lambda)$ by using the recursive algorithm in (19 *a*) and (23).

(3) Find

$$B(\lambda)=\sum_{j=0}^{n-i} B_j\lambda^{n-j-i}$$

where

$$B_j=A_j+B_{j-1}R_i, \quad j=1,2,\ldots,n-i, \quad \text{and} \quad B_0=A_0$$

(4) Find $L_i=Q_i^{-1}R_iQ_i$ by using (31)

$$\text{Vec}\,\{Q_i\}=\left\{\sum_{j=0}^{n-i} B_j^{\mathrm{T}}\otimes(R_i^{n-i-j})\right\}^{-1}\text{Vec}\,\{I_m\}$$

(5) Construct

$$A^{(1)}(\lambda)=\sum_{j=0}^{n-i} A_j^{(1)}\,\lambda^{n-j-i}$$

where

$$A_j^{(1)}=A_j+L_iA_{j-1}^{(1)}, \quad j=1,2,\ldots,n-i, \quad \text{and} \quad A_0^{(1)}=A_0$$

(6) Replace $A(\lambda)$ by $A^{(1)}(\lambda)$ and raise the index i by 1.

(7) If $i\leqslant n$, then go to (2); otherwise stop.

Similarly, the algorithm for determining the complete set of left solvents is described in Algorithm 2 as follows.

Algorithm 2—Complete set of left solvents

(1) Set an index $i=1$.

(2) Find a left solvent L_i of $A(\lambda)$ by the recursive algorithm in (24) and (26).

(3) Find

$$B(\lambda)=\sum_{j=0}^{n-i} B_j \lambda^{n-j-i}$$

where

$$B_j=A_j+L_i B_{j-1}, \quad j=1, \ldots, n-i, \quad \text{and} \quad B_0=A_0$$

(4) Find

$$R_i=P_i L_i P_i^{-1}$$

where the $m \times m$ matrix P_i can be constructed by

$$\operatorname{Vec}\{P_i\}=\left\{\sum_{j=0}^{n-i}(L_i^{n-i-j})^{\mathrm{T}} \otimes B_j\right\}^{-1} \operatorname{Vec}\{I_m\}$$

(5) Construct $A^{(1)}(\lambda)$ of degree $n-i$ by

$$A^{(1)}(\lambda)=\sum_{j=0}^{n-i} A_j^{(1)} \lambda^{n-j-i}$$

where

$$A_j^{(1)}=A_j+A_{j-1}^{(1)} R_i, \quad j=1, \ldots, n-i, \quad \text{and} \quad A_0^{(1)}=A_0$$

(6) Replace $A(\lambda)$ by $A^{(1)}(\lambda)$ and raise the index i by 1.

(7) If $i \leqslant n$, then go to (2); otherwise stop.

If a set of linear spectral factors is desired, then the following algorithm which is derived from (8 *a*) and Algorithm 1 can be applied.

Algorithm 3—Linear spectral factors

(1) Set an index $i=1$.

(2) Find a right solvent $R_i(=S_i)$ of $A(\lambda)$ by the recursive algorithm in (19 *a*) and (23).

(3) Find

$$B(\lambda)=\sum_{j=0}^{n-i} B_j \lambda^{n-i-j}$$

where

$$B_j=A_j+B_{j-1} S_i, \quad j=1, \ldots, n-i, \quad \text{and} \quad B_0=A_0$$

(4) Replace $A(\lambda)$ by $B(\lambda)$ and raise the index i by 1.

(5) If $i \leqslant n$, then go to (2); otherwise stop.

When the λ-matrix of interest has the property that $A(\lambda)=A^{\mathrm{T}}(-\lambda)$ of $2n$ degree and the spectral factorization is desired, we have the following algorithm.

Algorithm 4—Spectral factorization

(1) Set an index $i = 1$.

(2) Find a right solvent $R_i(=S_i)$ of $A(\lambda)$ by the recursive algorithm in (19 *a*) and (23).

(3) Find

$$B(\lambda) = \sum_{j=0}^{n-2i+1} B_j \lambda^{n-2i+1-j}$$

where

$$B_j = A_j + B_{j-1} S_i, \quad j = 1, \ldots, n-2i+1, \quad \text{and} \quad B_0 = A_0$$

(4) Find

$$A^{(1)}(\lambda) = \sum_{j=0}^{n-2i} A_j^{(1)} \lambda^{n-2i-j}$$

where

$$A_j^{(1)} = B_j - S_i^{\mathrm{T}} B_{j-1}, \quad j = 1, \ldots, n-2i, \quad \text{and} \quad A_0^{(1)} = B_0$$

(5) Replace $A(\lambda)$ by $A^{(1)}(\lambda)$ and raise the index i by 1.

(6) If $i \leqslant n/2$, then go to (2) ; otherwise stop.

The resulting spectral factorization is

$$A(\lambda) = D^{\mathrm{T}}(-\lambda) D(\lambda)$$

where

$$D(\lambda) = (\lambda I_m - S_{n/2}) \ldots (\lambda I_m - S_2)(\lambda I_m - S_1)$$

6. Normalization and initial guesses

The rate of convergence of the proposed numerical method heavily depends upon the initial guesses. In order to derive a systematic method for determining the initial guesses, the λ-matrix is normalized such that the latent roots of the λ-matrix are uniformly distributed around a unit circle with centre at $\lambda = 0$.

The arithmetic mean of the latent roots of a given λ-matrix in (1) is

$$k_1 = -\frac{\sum_{i=1}^{nm} \lambda_i}{nm} = \frac{-t_r A_1}{n} \tag{34}$$

where λ_i are the latent roots of $A(\lambda) = 0$ and $t_r A_1$ is the trace of a matrix A_1.

Substituting

$$\lambda = y + k_1 \tag{35}$$

into (1) results in

$$A^{(1)}(y) = I_m y^n + A_1^{(1)} y^{n-1} + \ldots + A_n^{(1)} \tag{36}$$

where

$$t_r A_1^{(1)} = 0$$

The geometric mean of the latent roots of $A^{(1)}(y)$ is

$$k_2 = \left(\prod_{i=1}^{nm} \alpha_i \right)^{1/nm} = (\det A_n^{(1)})^{1/nm} \tag{37}$$

where α_i are the latent roots of $A^{(1)}(y) = 0$.

Assuming that $k_2 \neq 0$ and substituting

$$y = |k_2| z \tag{38}$$

into (36), we have

$$A^{(2)}(z) = I_m z^n + A_1^{(2)} z^{n-1} + \dots + A_{n-1}^{(2)} z + A_n^{(2)} \tag{39}$$

where $|\det A_n^{(2)}| = 1$.

$A^{(2)}(z)$ is referred to as the normalized λ-matrix of $A(\lambda)$. The initial guesses for Algorithms 1 to 4 can be set in the following two ways.

(1) *Lin's method* (Lin 1943)

The scalar version of Lin's method is extended to the matrix case for determining the initial guess. The initial guess for the recursive algorithm of the right solvent of $A^{(2)}(z)$ in (39) is

$$R_0 = \{A_{n-1}^{(2)}\}^{-1} A_n^{(2)} \quad \text{if rank } A_{n-1}^{(2)} = m \tag{40}$$

The initial guess for the recursive algorithm of the left solvent of $A^{(2)}(z)$ becomes

$$L_0 = A_n^{(2)} \{A_{n-1}^{(2)}\}^{-1} \quad \text{if rank } A_{n-1}^{(2)} = m \tag{41}$$

(2) *Unitary matrix method*

Since the latent roots of the normalized λ-matrix has its arithmetic mean at zero and the geometric mean at unity, the initial guess can be set to a matrix with the magnitudes of all eigenvalues equal to unity. The matrix is a unitary matrix.

The desired solvents or spectral factors of the λ-matrix can be determined by denormalizing the solvents or spectral factors obtained from $A^{(2)}(z)$ via the transformations in (38) and (35).

7. Examples

Two examples are illustrated for finding the solvents, spectral factors and spectral factorization of λ-matrices. The computations were carried out on a PDP-11/70 using single precision floating point only.

Example 1

Given the λ-matrix

$$A(\lambda) = I_2 \lambda^3 + \begin{bmatrix} 4 & 2 \\ -2 & 7 \end{bmatrix} \lambda^2 + \begin{bmatrix} 12 & 11 \\ -2 & 28 \end{bmatrix} \lambda + \begin{bmatrix} 19 & 14 \\ 16 & 36 \end{bmatrix} \tag{42}$$

find the complete set of regular right solvents and its linear spectral factors.

Using the normalization scheme in § 6, we have

$$k_1 = -1{\cdot}83333 \tag{43 a}$$

$$k_2 = 1{\cdot}071363 \tag{43 b}$$

and the normalized matrix polynomial $A^{(2)}(z)$ in (39) becomes

$$A^{(2)}(z) = I_2 z^3 + \begin{bmatrix} -1{\cdot}40009 & 1{\cdot}866678 \\ -1{\cdot}86678 & 1{\cdot}40009 \end{bmatrix} z^2 + \begin{bmatrix} 6{\cdot}46154 & 3{\cdot}19447 \\ 4{\cdot}64650 & 10{\cdot}8176 \end{bmatrix} z + \begin{bmatrix} 3{\cdot}48240 & 0{\cdot}451770 \\ 10{\cdot}5263 & 1{\cdot}65273 \end{bmatrix} \tag{44}$$

The initial guess by using Lin's method is

$$\hat{R}_0 = \begin{bmatrix} -0{\cdot}0734806 & 0{\cdot}00712910 \\ -0{\cdot}941503 & -0{\cdot}155843 \end{bmatrix} \tag{45 a}$$

Applying Algorithm 1 to (44) and using the initial guess $\hat{R}_0$ gives the right solvent of $A^{(2)}(z)$ at the fourth iteration with an error tolerance of $\epsilon = 10^{-5}$ in (27). Denormalizing the obtained solvent by using (35) and (38) yields the desired right solvent

$$R_1 = \begin{bmatrix} -2{\cdot}00000 & -0{\cdot}37887 \times 10^{-7} \\ -1{\cdot}00000 & -2{\cdot}00000 \end{bmatrix} \tag{45 b}$$

The corresponding left solvent, which can be determined from (29) and (31), is

$$L_1 = \begin{bmatrix} -2{\cdot}46154 & 0{\cdot}115385 \\ -1{\cdot}84615 & -0{\cdot}153846 \end{bmatrix} \tag{45 c}$$

Now factoring out $(\lambda I_2 - L_1)$ from the left of $A(\lambda)$, or using (32), we have

$$A^{(1)}(\lambda) = I_2\lambda^2 + \begin{bmatrix} 1{\cdot}53846 & 2{\cdot}11538 \\ -3{\cdot}84615 & 5{\cdot}46154 \end{bmatrix} \lambda + \begin{bmatrix} 7{\cdot}76923 & 6{\cdot}42308 \\ 10{\cdot}7692 & 15{\cdot}6923 \end{bmatrix} \tag{46 a}$$

The normalized λ-matrix with $k_1 = -1{\cdot}750$ and $k_2 = 2{\cdot}70594$ is

$$\hat{A}^{(1)}(z) = I_2 z^2 + \begin{bmatrix} -0{\cdot}724901 & 0{\cdot}781756 \\ -1{\cdot}42138 & 0{\cdot}72490 \end{bmatrix} z + \begin{bmatrix} 1{\cdot}11162 & 0{\cdot}371635 \\ 1{\cdot}06632 & 1{\cdot}25607 \end{bmatrix} \tag{46 b}$$

The initial guess by Lin's method is

$$\hat{R}_0 = \begin{bmatrix} 0{\cdot}0474374 & 1{\cdot}21660 \\ -1{\cdot}37797 & 0{\cdot}652733 \end{bmatrix} \tag{46 c}$$

The right solvent of $\hat{A}^{(1)}(z)$ can be obtained at the thirteenth iteration with the same error tolerance of $\epsilon = 10^{-5}$. After denormalization we have the desired right selvent of $A^{(1)}(\lambda)$ or another right solvent of $A(\lambda)$, as follows :

$$R_2 = \begin{bmatrix} -0{\cdot}999976 & 1{\cdot}49996 \\ -2{\cdot}00004 & -2{\cdot}00001 \end{bmatrix} \tag{46 d}$$

The corresponding left solvent is

$$L_2 = \begin{bmatrix} -6{\cdot}94525 & 3{\cdot}35920 \\ -9{\cdot}64537 & 3{\cdot}94526 \end{bmatrix} \tag{46 e}$$

After factoring out $(\lambda I_2 - L_2)$ from the left of $A^{(1)}(\lambda)$ we have

$$A^{(2)}(\lambda) = I_2\lambda + \begin{bmatrix} -5{\cdot}40679 & 5{\cdot}47459 \\ -13{\cdot}4915 & 9{\cdot}40680 \end{bmatrix} \tag{47 a}$$

From (47 *a*) we have the desired right solvent of $A^{(2)}(\lambda)$ or the last right solvent of $A(\lambda)$ as

$$R_3 = \begin{bmatrix} 5{\cdot}40679 & -5{\cdot}47459 \\ 13{\cdot}49150 & -9{\cdot}40680 \end{bmatrix} \tag{47 b}$$

The complete set of linear spectral factors of $A(\lambda)$ in (42) can be determined by using Algorithm 3. The result is

$$A(\lambda) = (\lambda I_2 - S_3)(\lambda I_2 - S_2)(\lambda I_2 - S_1) \tag{48}$$

where

$$S_1 = \begin{bmatrix} -2{\cdot}00000 & -0{\cdot}37887 \times 10^{-7} \\ -1{\cdot}00000 & -2{\cdot}00000 \end{bmatrix}_{\text{iteration number} = 4}$$

$$S_2 = \begin{bmatrix} -0{\cdot}99996 & 2{\cdot}99991 \\ -1{\cdot}00002 & -2{\cdot}00004 \end{bmatrix}_{\text{iteration number} = 13}$$

$$S_3 = \begin{bmatrix} -1{\cdot}00004 & -4{\cdot}99991 \\ 4{\cdot}00002 & -2{\cdot}99986 \end{bmatrix}_{\text{iteration number} = 0}$$

It might be interesting to investigate the relationship between the eigenvalues of $A(\lambda)$ and the corresponding solvents and spectral factors. The eigenvalues of $A(\lambda)$ or solvents and spectral factors are

$$\left.\begin{aligned} \sigma(S_1) &= \sigma(R_1) = -2,\ -2 \\ \sigma(S_2) &= \sigma(R_2) = -1{\cdot}5 \pm j\sqrt{11} \\ \sigma(S_3) &= \sigma(R_3) = -2 \pm j\sqrt{7} \end{aligned}\right\} \tag{49}$$

From (49) we observe that, although we have repeated latent roots with a Jordan chain of length 2 and two pairs of complex roots, the corresponding solvents and spectral factors are all real. *This peculiarity has important* applications in the analysis and synthesis of large-scale multivariable control systems.

Example 2

Given a λ-matrix

$$A(\lambda)=I_2\lambda^4+\begin{bmatrix}0 & -6\\ 6 & 0\end{bmatrix}\lambda^3+\begin{bmatrix}-37 & 3\\ 3 & -18\end{bmatrix}\lambda^2 +\begin{bmatrix}0 & 49\\ -49 & 0\end{bmatrix}\lambda+\begin{bmatrix}68 & 2\\ 2 & 26\end{bmatrix} \qquad (50\,a)$$

and

$$A(\lambda)=A^{\mathrm{T}}(-\lambda) \qquad (50\,b)$$

it is desirable to find the spectral factorization of $A(\lambda)$ such that

$$A(\lambda)=A_1(\lambda)A_1^{\mathrm{T}}(-\lambda) \qquad (51)$$

where $A_1(\lambda)$ contains all latent roots of $A(\lambda)$ in the LHP of the complex plane and $A_1^{\mathrm{T}}(-\lambda)$ in the RHP of the complex plane.

Using Algorithm 4 and the initial guess I_2 (a unitary matrix), we have

$$A_1(\lambda)=(\lambda I_2-S_1)(\lambda I_2-S_2) \qquad (52)$$

where

$$S_1=\begin{bmatrix}-0{\cdot}99999 & 2{\cdot}00001\\ -1{\cdot}00000 & -0{\cdot}99999\end{bmatrix}\begin{matrix}\text{iteration number}=11\\ \text{error tolerance } \epsilon=10^{-4}\end{matrix}$$

$$S_2=\begin{bmatrix}-3{\cdot}99998 & 1{\cdot}99996\\ -0{\cdot}99999 & -2{\cdot}99999\end{bmatrix}\begin{matrix}\text{iteration number}=8\\ \text{error tolerance } \epsilon=10^{-4}\end{matrix}$$

Thus, $A_1(\lambda)$ becomes

$$A_1(\lambda)=I_2\lambda^2+\begin{bmatrix}4{\cdot}99997 & -3{\cdot}99997\\ 1{\cdot}99999 & 3{\cdot}99998\end{bmatrix}\lambda+\begin{bmatrix}1{\cdot}99996 & -7{\cdot}99996\\ 4{\cdot}99996 & 0{\cdot}99990\end{bmatrix} \qquad (53)$$

Observe that, although the latent roots of $A(\lambda)$ are all complex, i.e.

$$\lambda=-1\pm j\sqrt{2}\,;\quad \tfrac{1}{2}(-7\pm j\sqrt{7})\,;\quad 1\pm j\sqrt{2}\,;\quad \tfrac{1}{2}(7\pm j\sqrt{7}) \qquad (54)$$

S_1 and S_2 are real matrices.

8. Conclusions

A generalized Newton method, which is based on the contracted gradient of a matrix polynomial, has been derived for solving the complete sets of solvents and spectral factors of a matrix polynomial without directly using the eigenvalues of the matrix polynomial. Lin's method and the unitary matrix method have been proposed for determining the initial estimates of the solvents and spectral factors. Thus, the convergence speed of the new numerical method is improved. The proposed method can be used to determine the spectral factorization of a matrix polynomial for optimal control, filtering and estimation problems. The obtained solvents can be used to construct a block similarity transformation matrix for the decoupling of a large-scale multivariable control system.

Acknowledgments

This work was supported in part by the U.S. Army Research Office, under Grant DAAG 29-80-K-0077 and by the U.S. Army Missile Research and Development Command, under Contract DAAH 01-80-C-0323.

References

Anderson, B. D. O., 1967, *I.E.E.E. Trans. autom. Control*, **12**, 410.

Anderson, B. D. O., and Bitmead, R. E., 1977, *Int. J. Control*, **26**, 235.

Barnett, S., 1971, *Matrices in Control Theory* (New York : Van Nostrand Reinhold).

Dennis, J. E., Traub, J. F., and Weber, R. P., 1976, *SIAM J. numer. Anal.*, **13**, 831 ; 1978, *Ibid.*, **15**, 523.

Fong, J. T., 1971, *J. Res. natn Bur. Stand.*, **75B**, 97.

Frazer, R. A., Duncan, W. J., and Collar, A. R., 1952, *Elementary Matrices and some Applications to Dynamics and Differential Equations* (Cambridge : The University Press).

Gohberg, I., Kaashoek, M. A., and Rodman, L., 1978, *Spectral Analysis of Families of Operator Polynomial and a Generalized Vandermonde Matrix.* 1—*The Finite-Dimensional Case*, Topics in Functional Analysis (New York : Academic Press), pp. 91–128.

Jury, E. I., 1974, *Inners and Stability of Dynamic Systems* (New York : Wiley Interscience).

Kailath, T., 1980, *Linear Systems* (New Jersey : Prentice-Hall).

Lancaster, P., 1966, *Lambda-Matrices and Vibrating Systems* (New York : Pergamon Press) ; 1970, *SIAM Rev.*, **12**, 544.

Lin, S., 1943, *J. Math. Phys.*, **22**, 60.

Markus, A. S., and Mereuca, I. V., 1978, *Math. USSR Izvestija*, **7**, 1105.

Papaconstantinou, C., 1975, *Proc. Instn elect. Engrs*, **122**, 312.

Shieh, L. S., and Chahin, N., 1981, *Int. J. Systems Sci.*, **12**, 305.

Shieh, L. S., and Sacheti, S., 1978, *Int. J. Control*, **27**, 245.

Shieh, L. S., Shih, C. D., and Yates, R. E., 1978, *J. dynam. Syst. Meas. Control*, **214**.

Shieh, L. S., and Tajvari, A., 1980, *Proc. Instn elect. Engrs*, **127**, Pt. D, 19.

Tsay, Y. T., Shieh, L. S., and Yates, R. E., 1981, *JACC*, Charlottesville, Virginia, June.

Vetter, W. J., 1973, *SIAM Rev.*, **15**, 352.

Youla, D. C., 1961, *IRE Trans. Inf. Theory*, **7**, 172.