MATLAB FUNCTION FOR POLYNOMIAL PLUS/MINUS FACTORIZATION BASED ON BAND STRUCTURED MATRIX DECOMPOSITION

Martin Hromčík, Michael Šebek

Institute of Information Theory and Automation, Czech Academy of Sciences,
Prague, Czech Republic
hromcik@utia.cas.cz

Abstract

A new algorithm for the plus/minus factorization of a scalar discrete-time polynomial and its implementation in the MATLAB programming language are presented in this report. The objects and standard routines of the "Polynomial Toolbox for MATLAB" are employed. The method is based on the relationship of polynomial algebra to the algebra of band structured infinite dimensional matrices. Employing standard numerical routines for factorizations of constant matrices, as they are implemented via MATLAB dedicated fast built-in functions, brings computational efficiency and reliability. Performance of the proposed algorithm is demonstrated by a practical application. Namely the problem of computing an l1-optimal output feedback dynamic compensator to a discrete time SISO plant is considered. Involved plus-minus factorization is resolved by our new method.

1 Introduction

This paper describes a new method for the plus-minus factorization of a discrete-time polynomial. Given a polynomial in the z variable,

$$p(z) = p_0 + p_1 z + p_2 z^2 + \cdots + p_n z^n,$$

without any roots on the unit circle, its plus/minus factorization is defined as

$$p(z) = p^+(z)p^-(z) \tag{1}$$

where $p^+(z)$ has all roots inside and $p^-(z)$ outside the unit disc. Clearly, the scalar plus/minus factorization is unique up to a scaling factor.

Polynomial plus/minus factorization has many applications in control and signal processing problems. For instance, efficient algebraic design methods for time-optimal controllers [1], quadratically optimal filters for mobile phones [15, 16], and l_1 optimal regulators [6], to name just a few, all recall the +/- factorization as a crucial computational step.

2 Existing Methods

From the computational point of view, nevertheless, the task is not well treated. There are two quite natural methods.

One of them is based on direct computation of roots. Using standard methods for polynomial roots evaluation, see [8, 17] for instance, one can separate the stable and unstable roots of $p(s)$ directly and construct the plus and minus parts from related first order factors or, alternatively, employ a more efficient recursive procedure based on the matrix eigenvalue theory [17].

Alternative algorithm relies on polynomial spectral factorization and gratest polynomial divisor computation. If $q(z)$ is the spectral factor of the symmetric product $p(z)p(z^{-1})$ then the greatest common divisor of $p(z)$ and $q(z)$ is obviously the plus factor of $p(z)$. The minus factor can be derived similarly from $p(z^{-1})$ and $q(z^{-1})$. As opposed to the previous approach based on direct roots computation which typically makes problems for higher degrees and/or roots multiplicities, this procedure relies on numerically reliable algorithms for polynomial spectral factorization [13, 5]. Unfortunately, the polynomial greatest common divisor computation is much more sensitive. As a result, both these techniques do not work properly for high degrees (say over 50).

Quite recently, a new approach to the problem was suggested by the authors of this report in [14]. The method is inspired by an efficient algorithm for polynomial spectral factorization, see [5]. It provides both a fruitful view on the relation between DFT and the Z-transform theory, and a powerful computational tool in the form of the fast Fourier transform algorithm.

Success of adapting a powerful spectral factorization algorithm for the plus-minus factorization was inspiring for us. We decided to undertake a similar way with another spectral factorization procedure, namely the Bauer's method, which is described in the following sections.

3 Bauer's method for polynomial spectral factorization

F. I. Bauer published his method for spectral factorization of a discrete-time scalar polynomial in 1955, see [2, 3]. The procedure is based on the relationship between polynomials and related infinite Toeplitz-type Sylvester matrices.

3.1 Algebra of Sylvester matrices

Given a two-sided polynomial $p(z) = p_{-m}z^{-m} + \cdots + p_0 + \cdots + p_n z^n$, we define its Sylvester companion matrix T_p^N of order N,

$$N \geq \max(n, m)$$

as an N by N square matrix constructed according to the following scheme:

$$T_p^N = \begin{pmatrix} p_0 & p_1 & \dots & p_n & 0 & \dots & 0 \\ p_{-1} & p_0 & p_1 & \dots & p_n & \ddots & \vdots \\ \vdots & p_{-1} & \ddots & \ddots & & \ddots & 0 \\ p_{-m} & \vdots & \ddots & & & & p_n \\ 0 & p_{-m} & & & & \ddots & \vdots \\ \vdots & \ddots & \ddots & & \ddots & \ddots & p_1 \\ 0 & \cdots & 0 & p_{-m} & \dots & p_{-1} & p_0 \end{pmatrix}$$

To show the relation between the polynomial algebra and the algebra of Sylvester matrices, let us consider two simple polynomials $p_1(z) = 3z^{-1} + 2 + z$ and $p_2(z) = z^{-1} + 3$. Their companion matrices of order four read respectively

$$T_{p1}^4 = \begin{pmatrix} 2 & 1 & 0 & 0 \\ 3 & 2 & 1 & 0 \\ 0 & 3 & 2 & 1 \\ 0 & 0 & 3 & 2 \end{pmatrix}$$

$$T^4_{p2} = \begin{pmatrix} 3 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 \\ 0 & 1 & 3 & 0 \\ 0 & 0 & 1 & 3 \end{pmatrix}$$

Their sum $p_3(z) = p_1(z) + p_2(z)$ equals

$$p_3(z) = 4z^{-1} + 5 + z$$

and its companion matrix can be computed as direct sum of related companion matrices T^4_{p1}, T^4_{p2}:

$$T^4_{p3} = \begin{pmatrix} 5 & 1 & 0 & 0 \\ 4 & 5 & 1 & 0 \\ 0 & 4 & 5 & 1 \\ 0 & 0 & 4 & 5 \end{pmatrix}$$

Similarly, their product $p_4 = p_1 p_2 = 3z^{-2} + 11z^{-1} + 7 + 3z$ has a companion matrix

$$T^4_{p4} = T^4_{p1} T^4_{p2} = \begin{pmatrix} 7 & 3 & 0 & 0 \\ 11 & 7 & 3 & 0 \\ 3 & 11 & 7 & 3 \\ 0 & 3 & 11 & 6 \end{pmatrix}$$

3.2 Bauer's method for spectral factorization

As we have illustrated above, finite dimensional matrices are sufficient to accommodate "finite" algebraic problems. On the other hand, if we do not restrict to finite dimensionality of related matrices, transcendent problems, including spectral factorization, involving polynomials can be resolved by this approach as well.

We will illustrate the Bauer's spectral factorization method by means of a simple example. An interested reader can find detailed description in the original work [2] or, alternatively, in the survey paper [4].

Given $p(z) = 2z^{-1} + 5 + 2z$ its companion matrix of order five reads

$$T_p = \begin{pmatrix} 5 & 2 & 0 & 0 & 0 \\ 2 & 5 & 2 & 0 & 0 \\ 0 & 2 & 5 & 2 & 0 \\ 0 & 0 & 2 & 5 & 2 \\ 0 & 0 & 0 & 2 & 5 \end{pmatrix}$$

As p is symmetric and positive definite on the unit circle its spectral factor x exists such that

$$x^\star x = p$$

holds and x is stable. The star stands for polynomial discrete-time conjugation, $z \rightarrow z^{-1}$. The spectral factor coefficients can be approximated using the Cholesky factorization of T_p:

$$T_x = \begin{pmatrix} 2.236 & 0.8944 & 0 & 0 & 0 \\ 0 & 2.049 & 0.9759 & 0 & 0 \\ 0 & 0 & 2.012 & 0.9941 & 0 \\ 0 & 0 & 0 & 2.003 & 0.9985 \\ 0 & 0 & 0 & 0 & 2.001 \end{pmatrix}$$

The diagonals of T_x obviously converge to the genuine spectral factor coefficients: $x(z) = 1 + 2z$.

An interesting feature of this routine is that particular columns of T_x can be computed iteratively, using only latest preceding column and the coefficients of $p(z)$, see [4] for details. As a result, the final algorithm is favorably memory efficient. Mainly for this reason the method is still quite popular in spite of the fact that some later approaches, see eg. [13, 5], provide a faster rate of convergence.

4 Plus-minus factorization and Bauer's method

A modification of the Bauer's method for the non-symmetric polynomial plus-minus factorization is worked out in this section.

4.1 LU factorization

As we have shown in section II., algebra of companion matrices is not limited to the symmetric case. Also the matrix theory provides useful factorization techniques for non-symmetric matrices along with stable and efficient procedures for their computation.

Bauer's method calls for the Cholesky factorization to get the desired spectral factor. This routine assumes the input matrix to be symmetric and positive definite which is the case in the spectral factorization problem. However, if we aim at modifying the method in order to capture the non-symmetric plus/minus factorization case, we need to leave this concept and employ another technique since the companion matrix is no longer symmetric.

The Cholesky factorization decomposes the input matrix into a product of two matrices basically that are upper and lower triangular respectively. Considering this observation, the most natural alternative for the non-symmetric plus/minus case seems to be the LU-factorization concept.

Definition (general LU-factorization): LU factorization expresses any square matrix A as the product of a permutation of a lower triangular matrix and an upper triangular matrix,

$$A = LU$$

where L is a permutation of a lower triangular matrix with ones on its diagonal and U is an upper triangular matrix.

The permutations are necessary for theoretical reasons in the general case. For instance, the matrix

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

cannot be expressed as the product of triangular matrices without interchanging its two rows. However, the special band structure of the companion matrices can be exploited to show that the permutations are not necessary and the factorization can be expressed simply as a product of a lower and an upper triangular matrix.

Lemma 1: Given a scalar discrete-time two-sided polynomial $p(z)$ with roots not lying on the unit circle, its companion matrix can be factored in the form $T_p = LU$ where L and U are lower and upper triangular matrices respectively.

Proof: If a (possibly two-sided) polynomial p is nonzero at the unit circle then the principal minors of its companion matrix are known to be nonzero, see the reasoning in [2]. Further, according to [7], Theorem 3.2.1, a matrix A has the desired lower-upper triangular factorization if its all principal minors are nonzero. Combining these two observations, we arrive at the statement of the lemma.

Following Lemma 1, a new algorithm for polynomial plus-minus factorization is suggested in the next subsection.

4.2 Plus/minus factorization algorithm

Given a (scalar, one-sided) polynomial

$$p(z) = p_0 + p_1 z + \cdots + p_d z^d \ ,$$

nonzero for $|z| = 1$, we first apply a direct degree shift to arrive at a two-sided polynomial

$$\tilde{p}(z) = p_0 z^{-\delta} + \cdots + p_d z^{d-\delta},$$

where δ is the number of roots of $p(z)$ lying inside the unit circle. Now, instead of solving equation (1), we look for $\tilde{p}^+(z) = \tilde{p}_0^+ + \tilde{p}_1^+ z^{-1} + \cdots + \tilde{p}_\delta^+ z^{-\delta}$ and $\tilde{p}^-(z) = \tilde{p}_0^- + \tilde{p}_1^- z + \cdots + \tilde{p}_{d-\delta}^- z^{d-\delta}$ such that

$$\tilde{p}(z) = \tilde{p}^+(z)\tilde{p}^-(z) \tag{2}$$

Relation between the pairs $\tilde{p}^+, \tilde{p}^-$ and p^+, p^- are obvious.

Having composed the companion matrix $T_{\tilde{p}}^N$ of sufficiently high order N, its LU factorization is performed. An approximation to the plus and minus factors of $\tilde{p}$ can then be read from the last column of the L and U factors respectively, similarly to the spectral factorization case.

The degree shift yielding the two-sided polynomial $\tilde{p}$ is necessary to assure correct decomposition of $\tilde{p}$ into stable and antistable parts. If the shift were not performed or were different from δ, the decomposition would still work in principle , however, the strict stability and antistability of particular factors would be lost.

Detailed description of the resulting algorithm follows.

Algorithm 1: Scalar discrete-time plus-minus factorization.

Input: Scalar polynomial
$p(z) = p_0 + p_1 z + \cdots + p_d z^d$, nonzero for $|z| = 1$.

Output: Polynomials $p^+(z)$ and $p^-(z)$, the plus and minus factors of $p(z)$.

Step 1 - *Choice of the companion matrix size.*
Decide about the number N. N approximately 10 to 50 times larger than d is recommended up to our practical experience.

Step 2 - *Degree shift.*
Find out the number δ of zeros of $p(z)$ inside the unit disc. A modification of well known Schur stability criterion can be employed, see [10] for instance.

Having δ at hand, construct a two-sided polynomial $\tilde{p}(z)$ as

$$\tilde{p}(z) = p(z) z^{-\delta} = p_0 z^{-\delta} + \cdots + p_d z^{d-\delta} =$$

$$= \tilde{p}_{-\delta} z^{-\delta} + \cdots + \tilde{p}_0 + \cdots + \tilde{p}_{d-\delta} z^{d-\delta}$$

Step 3 - *Construction of $T_{\tilde{p}}^N$:*
Following the section III.A, construct the Sylvester companion matrix related to $\tilde{p}$ of order N.

Step 4 - *LU decomposition of $T_{\tilde{p}}^N$:*
Perform the LU decomposition of $T_{\tilde{p}}^N$:

$$T_{\tilde{p}}^N = LU$$

L and U are lower and upper triangular matrices respectively.

Step 5 - *Construction of polynomial factors:*

Columns of the L and U matrices contains a nonzero vector l, u of length $\delta+1$ and $d-\delta+1$ lying under and above the main diagonal respectively. Take the last full column $l=[l_0,l_1,\ldots,l_\delta]$ to create the plus factor of $p(z)$ as

$$p^+(z)=l_0+l_1z+\cdots+l_\delta z^\delta$$

The minus factor is constructed in a similar way using the last vector u. ◇

5 Example

To illutrate the usefulness of polynomial plus-minus factorization and to demonstrate the power of the proposed algorithm at the same time, we will discuss the l_1 optimal control problem.

l_1 optimization is a modern design technique, see [11] for a survey. The design goal lies in minimizing the l_1 norm of a closed loop transfer function. Such a way, the magnitude of measured output signal is minimized with respect to bounded, yet persistent input disturbances. l_1 optimal controllers have already found an application in some irrigation channel regulation problem, see [12] for instance.

Quite recently a new method has been suggested by Z. Hurak et al. for the computation of an l_1 optimal discrete-time SISO compensator, see [6]. Unlike their predcessors, the authors rely on the transfer function description purely and carefully exploit the algebraic structure of the problem. The resulting algorithm is given in [6] along with the following example.

Let us compute a feedback controller the minimizes ℓ_1 norm of the sensitivity function for a plant described by

$$G(z^{-1})=\frac{b(z)}{a(z)}=\frac{-45-132z^{-1}+9z^{-2}}{-20-48z^{-1}+5z^{-2}}$$

The solution consists of the following computational steps

1. plus-minus factorization of $a(z^{-1})=a^+(z^{-1})a^-(z^{-1})$ and $b(z^{-1})=b^+(z^{-1})b^-(z^{-1})$
2. find the minimum degree solution to $a(z^{-1})x_0(z^{-1})+b(z^{-1})y_0(z^{-1})=1$
3. find a solution to $a^-(z^{-1})b^-(z^{-1})x(z^{-1})+y(z^{-1})=a(z^{-1})x_0(z^{-1})$ of given degree of $y(z^{-1})$ and with minimum $\|.\|_1$ norm.
4. the optimal controller is given by

$$C(z^{-1})=\frac{a^+(z^{-1})b^+(z^{-1})y_0(z^{-1})+a(z^{-1})x(z^{-1})}{a^+(z^{-1})b^+(z^{-1})x_0(z^{-1})-b(z^{-1})x(z^{-1})}$$

The first step can be efficiently and reliably performed using the algorithm proposed in section IV.B of this report. We take small-size Sylvester matrices first for illustrative purposes, say N equal to 4. T_a and T_b read respectively

$$Ta=\begin{bmatrix}-48 & 5 & 0 & 0\\ -20 & -48 & 5 & 0\\ 0 & -20 & -48 & 5\\ 0 & 0 & -20 & -48\end{bmatrix}$$

$$Tb=\begin{bmatrix}-132 & -45 & 0 & 0\\ 9 & -132 & -45 & 0\\ 0 & 9 & -132 & -45\\ 0 & 0 & 9 & -132\end{bmatrix}$$

and their LU factorization gives rise to

$$T_a^+ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.4167 & 1 & 0 & 0 \\ 0 & 0.3993 & 1 & 0 \\ 0 & 0 & 0.4003 & 1 \end{bmatrix}$$

$$T_a^- = \begin{bmatrix} -48 & 5 & 0 & 0 \\ 0 & -50.083 & 5 & 0 \\ 0 & 0 & -49.997 & 5 \\ 0 & 0 & 0 & -50 \end{bmatrix}$$

and

$$T_b^+ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -0.06818 & 1 & 0 & 0 \\ 0 & -0.06663 & 1 & 0 \\ 0 & 0 & -0.06667 & 1 \end{bmatrix}$$

$$T_b^- = \begin{bmatrix} -132 & -45 & 0 & 0 \\ 0 & -135.1 & -45 & 0 \\ 0 & 0 & -135 & -45 \\ 0 & 0 & 0 & -135 \end{bmatrix}$$

These matrix factors give a fair approximation to a^+, a^-, b^+, b^- polynomials:

$$a^+ = 0.40003z^{-1} + 1, a^- = -49.997z^{-1} + 5$$

$$b^+ = -0.067z^{-1} + 1, b^- = -135z^{-1} - 45$$

To get more accurate results, N is increased. Taking $N = 20$ yields perfectly accurate results,

$$a^+ = 2/5z^{-1} + 1, a^- = -50z^{-1} + 5$$

$$b^+ = -1/15z^{-1} + 1, b^- = -135z^{-1} - 45$$

6 Further Research

At this stage the LU decomposition is performed via standard routines, see [7] for instance, implemented in standard packages such as LAPACK or commercial MATLAB. Nevertheless, thanks to the strong structurallity of involved Toeplitz matrices, dedicated efficient routines for their LU factorization are likely to exist. Now, the authors have been seeking such algorithms. Hopefully the results of this research will be published in the final version of this report.

7 Conclusion

A new method for the discrete-time plus-minus factorization problem in the scalar case has been proposed. The new method relies on numerically stable and efficient LU factorization of associated Toeplitz matrices. Besides its good numerical properties, the derivation of the routine also provides an interesting look into the related mathematics, combining the results of the matrix theory and algebraic design approach. The suggested method is employed in a practical application of l_1 optimal control problem.

Acknowledgements

This work was supported by the Ministry of Education of the Czech Republic under contract No. ME496.

References

[1] Kučera V., *Analysis and Design of Discrete Linear Control Systems*, Academia Prague (1991).

[2] F. L. Bauer. Ein direktes iterations verfahren zur Hurwitz-zerlegung eines polynoms (in German), Arch. Elektr. Uebertragung, vol. 9, pp. 285290, 1955.

[3] F. L. Bauer. Beitr age zur entwicklung numerischer verfahren f ur programmgesteuerte rechenanlagen, II. Direkte faktorisierung eines polynoms (in German), Sitz. Ber. Bayer. Akad. Wiss., pp. 163203, 1956.

[4] Sayed A.H., Kailath T., *A Survey of Spectral Factorization Methods*, Numerical Linear Algebra with Applications, 2001, **8**, pp. 467-496.

[5] M. Hromcik, J. Jezek, M. Sebek, *New Algorithm for Spectral Factorization and its Practical Application*, Proceedings of the European Control Conference ECC'2001, Porto, Portugal, September 1-5, 2001.

[6] Z. Hurak, M. Sebek, *Algebraic Approach to l-1 Optimal Control*, to appear.

[7] Golub G. H., Van Loan C. F., "Matrix Computations", The Johns Hopkins University Press, Baltimore and London, 1990.

[8] Higham N. J., *Accuracy and Stability of Numerical Algorithms*, S.I.A.M., Philadelphia (1996).

[9] Kwakernaak H., Šebek M., *PolyX Home Page*,
`http://www.polyx.cz/, http://www.polyx.com/`.

[10] Barnett S., *Polynomials and Linear Control*. Marcel Dekker, New York and Basel (1983).

[11] Dahleh M.A., Diaz-Bobillo I.J., *Control of Uncertain Systems: A Linear Programming Approach*, Prentice Hall, New Jersey, 1995.

[12] Malaterre P.-O., Khammash M. *l-1 Controller Design for a High-Order 5-pool Irrigation Canal System*, IEEE-CDC conference, December, 2000, in Sydney, Australia.

[13] Ježek J. and Kučera V., *Efficient Algorithm for Matrix Spectral Factorization*,Automatica, vol. 29, pp. 663-669, 1985.

[14] Hromčík M., Šebek M, *New Algorithm for Polynomial Plus/minus Factorization Based on FFT*, submitted to the European Control Conference ECC'03, Cambridge, UK.

[15] M. Sternad and A. Ahlén, *Robust Filtering and Feedforward Control Based on Probabilistic Descriptions of Model Errors*, Automatica, 29, pp. 661-679.

[16] K. Ohrn, A. Ahlén and M. Sternad, *A Probabilistic Approach to Multivariable Robust Filtering and Open-loop Control*, IEEE Transactions on Automatic Control, 40, pp. 405-417.

[17] The Mathworks, *Using MATLAB 5.3*, The Matrhworks, 1999.