A New Multichannel Spectral Factorization Algorithm for Parahermitian Polynomial Matrices

Zeliang Wang, John G. McWhirter
School of Engineering, Cardiff University, Cardiff, Wales, UK
WangZ49@cardiff.ac.uk, McWhirterJG@cardiff.ac.uk

***Abstract*—A novel multichannel spectral factorization algorithm is illustrated in this paper. This new algorithm is based on an iterative method for polynomial eigenvalue decomposition (PEVD) called the second order sequential best rotation (SBR2) algorithm [1]. By using the SBR2 algorithm, multichannel spectral factorization problems are simply broken down to a set of single channel problems which can be solved by means of existing one dimensional spectral factorization algorithms. In effect, it transforms the multichannel spectral factorization problem into one which is much easier to solve. The proposed algorithm can be used to calculate the approximate spectral factor of any parahermitian polynomial matrix. Two worked examples are presented in order to demonstrate its ability to find a valid spectral factor, and indicate the level of accuracy which can be achieved.**

I. Introduction

Spectral factorization plays a crucial role in constructing a casual system which corresponds to a given spectral density function. The earliest applications are in the solution of control system and linear estimation problems. In recent years, it has attracted lots of interest in the areas of digital signal processing and communications, such as designing minimum phase filters, quadrature-mirror filter (QMF) bank [2], and calculating the optimum transmit filter matrices for the precoding, and the receive filter matrices for the equalization, of multiple-input and multiple-output (MIMO) communication systems [3]. A number of algorithms for single channel spectral facotorization were developed during the past decades, such as the Newton-Raphson based method proposed by Wilson [4], and the spectral factorization algorithms published by Janashia and coauthors [5]–[7]. A paper written by Sayed [8] presents a survey of spectral factorization methods including the Bauer method, the Schur algorithm, the Levinson-Durbin algorithm, and techniques based on the Riccati equation, the Kalman filter and so on. The paper written by Goodman [9] provides a useful performance comparison between the Bauer and Wilson methods. Most of these spectral factorization algorithms, with the exception of those due to Wilson and Janashia, do not extend to the multichannel situation. Wilson's algorithm seems to provide a viable approach to the multichannel spectral factorization problem in terms of stability and reliability but is reputed to run into problems when the number of channels grows too big.

In this paper we present an entirely different approach to the multichannel spectral factorization problem. It is based on an application of the SBR2 algorithm [1] to diagonalise the cross-spectral density matrix by means of a generalised similarity transformation which is entirely lossless.

Throughout this paper, polynomial matrices and vectors are denoted by underscored upper case bold characters and underscored lower case bold characters respectively. The notation of $\sim$ upon a polynomial matrix is used to denote the paraconjugate operation of a polynomial matrix, and the superscripts $*$, T and H stand for complex conjugate, matrix transpose and Hermitian transpose operation respectively. $\mathbf{I}_m$ represents the $m\times m$ identity matrix. Given a $m\times n$ polynomial matrix $\underline{\mathbf{X}}(z)$ with the indeterminate variable z^{-1}, it can be simply expressed as

$$\underline{\mathbf{X}}(z) = \sum_{\tau=T_1}^{T_2} \mathbf{X}(\tau) z^{-\tau} = \begin{bmatrix} \underline{x}_{11}(z) & \underline{x}_{12}(z) & \cdots & \underline{x}_{1n}(z) \\ \underline{x}_{21}(z) & \underline{x}_{22}(z) & \cdots & \underline{x}_{2n}(z) \\ \vdots & \vdots & \ddots & \vdots \\ \underline{x}_{m1}(z) & \underline{x}_{m2}(z) & \cdots & \underline{x}_{mn}(z) \end{bmatrix}, \tag{1}$$

where $\tau \in \mathbb{Z}$, $T_1 \leq T_2$, $\mathbf{X}(\tau) \in \mathbb{C}^{m\times n}$, and $\underline{x}_{kl}(z)$ is the polynomial matrix entity which can be expanded as

$$\underline{x}_{kl}(z) = \sum_{\tau=T_1}^{T_2} x_{kl}(\tau) z^{-\tau} \quad . \tag{2}$$

The effective polynomial order of $\underline{\mathbf{X}}(z)$ is given by $T_2 - T_1$.

In many signal processing applications involving multiple sensors, given a data vector $\mathbf{x}[n]$, the space-time covariance matrix is represented by $\mathbf{R}(\tau) = E\left\{\mathbf{x}[n]\mathbf{x}^{\mathrm{H}}[n-\tau]\right\}$, in which $E\{\cdot\}$ denotes the expectation. After applying z-transform to it, we get the cross-spectral density (CSD) matrix $\underline{\mathbf{R}}(z) = \sum_\tau \mathbf{R}(\tau) z^{-\tau}$. Note that the CSD matrix above is a parahermitian polynomial matrix, which satisfies $\tilde{\underline{\mathbf{R}}}(z) = \underline{\mathbf{R}}(z)$. Here $\tilde{\underline{\mathbf{R}}}(z)$ is the paraconjugate of $\underline{\mathbf{R}}(z)$, and it can be defined as $\tilde{\underline{\mathbf{R}}}(z) = \underline{\mathbf{R}}^{\mathrm{H}}(1/z)$. i.e. Applying Hermitian transpose to the polynomial coefficient matrices and time-reversing all the elements in $\mathbf{R}(\tau)$.

The paper is organised as follows. The next section introduces the one dimensional spectral factorization problem; Section III describes the SBR2 algorithm; Section IV outlines the new multichannel spectral factorization algorithm; Simulation results are presented in Section V and some conclusions are drawn in Section VI.

II. One Dimensional Spectral Factorization

The one dimensional spectral factorization problem can be stated briefly as follows. Given a data sequence $g(n)$, derive an associated causal sequence $h(n)$ such that

$$g(n) = h(n) \otimes h^*(-n) \quad , \tag{3}$$

or equivalently

$$G(z) = H(z)H^*(1/z) = G^+(z)G^-(z) \quad , \tag{4}$$

where $\otimes$ denotes the convolution operation. Both $g(n)$ and $h(n)$ represent discrete digital sequences. Note that the sequence $g(n)$ constitutes the autocorrelation of $h(n)$, and only when $g(n)$ is a symmetric sequence which satisfies $g(n) = g^*(-n)$, can it be factored as in (3). Equation (4) gives the z-transform of equation (3), and it can be expressed as the product of an 'outer' spectral factor $G^+(z)$ and an 'inner' spectral factor $G^-(z)$ [5]. Finding the spectral factor of $g(n)$, corresponds to evaluating all the roots of $G(z)$ expressed as proper polynomial in z. For example, considering the z-transform of a finite sequence $g(n)$ of length L we have

$$G(z) = \sum_{n=0}^{L-1} g(n)z^{-n} = g(0)+g(1)z^{-1}+\cdots+g(L-1)z^{-(L-1)}, \tag{5}$$

which can be written in the form

$$G(z) = \frac{g(0)z^{L-1} + g(1)z^{L-2} + \cdots + g(L-1)}{z^{L-1}} \tag{6}$$

which constitutes a proper polynomial as required. The zeros of this polynomial can be easily evaluated in MATLAB. Note that to form the minimum phase factor, corresponding to a stable filter, only the roots inside the unit circle, $|z| < 1$ and half of those roots on the unit circle, $|z| = 1$ can be chosen [10]. Then the problem remains to find $h(n)$ from the selected roots. In this paper, the one dimensional spectral factorization is calculated using the Newton-Raphson method, as adopted for the $spf(\cdot)$ function provided in the MATLAB polynomial matrix toolbox from PolyX [11].

III. The SBR2 Algorithm

The second order sequential best rotation (SBR2) algorithm provides a powerful iterative technique for diagonalising a parahermitian polynomial matrix, such as the CSD matrix mentioned above. It can generate optimal transforms for the problems of designing precoding and equalisation filters for MIMO communication systems [12], subband coding [13] etc. Given an $m \times m$ parahermitian polynomial matrix $\underline{\mathbf{R}}(z)$ the SBR2 algorithm implements a transformation of the form

$$\underline{\mathbf{H}}(z)\underline{\mathbf{R}}(z)\tilde{\underline{\mathbf{H}}}(z) \approx \underline{\mathbf{D}}(z) \quad , \tag{7}$$

where $\underline{\mathbf{D}}(z)$ is (ideally) a diagonal polynomial matrix, and $\underline{\mathbf{H}}(z)$ is a paraunitary matrix so that $\underline{\mathbf{H}}(z)\tilde{\underline{\mathbf{H}}}(z) = \tilde{\underline{\mathbf{H}}}(z)\underline{\mathbf{H}}(z) = \mathbf{I}_m$.

Equation (7) represents the diagonalization of a parahermitian polynomial matrix through pre-multiplication by a paraunitary polynomial matrix and post-multiplication by its paraconjugate. This can be considered as a paraunitary similarity transformation. Each iteration of SBR2 algorithm applies a single elementary paraunitary matrix, comprising a simple delay operator and an elementary Jacobi rotation, and is used to eliminate a pair of dominant off-diagonal polynomial coefficients. Let $\underline{\mathbf{P}}^{(i)}(z)$ and $\mathbf{Q}^{(i)}(\theta,\phi)$ represent the elementary delay and rotation matrices respectively, at the i-th iteration. The elementary paraunitary matrix at this iteration then takes the form

$$\underline{\mathbf{E}}^{(i)}(z) = \mathbf{Q}^{(i)}(\theta,\phi)\underline{\mathbf{P}}^{(i)}(z) \quad , \tag{8}$$

where θ and ϕ are parameters which define the elementary rotation. The algorithm continues until all the off-diagonal energy ($L2$ norm) has been transferred onto the diagonal of the coefficient matrix of order zero. Assuming that the algorithm has converged to sufficient accuracy after N times iterations, the generated paraunitary polynomial matrix is given by

$$\underline{\mathbf{H}}(z) = \underline{\mathbf{E}}^{(N)}(z)\cdots\underline{\mathbf{E}}^{(2)}(z)\underline{\mathbf{E}}^{(1)}(z) \quad . \tag{9}$$

Further details can be found in [1].

IV. Multichannel Spectral Factorization

The proposed multichannel spectral factorization starts by diagonalising the input parahermitian polynomial matrix using the SBR2 algorithm. This process breaks the multichannel problem down into a set of distinct single channel problems. Each polynomial element in the diagonal matrix defines a one dimensional spectral factorization problem which can be accurately solved using, for example, Wilson's algorithm. In essence, the SBR2 algorithm builds a bridge between multichannel and single channel spectral factorization. The resulting outer (inner) spectral factors of the diagonal matrix are then used to construct the spectral factor of the input parahermitian polynomial matirx. As the polynomial orders of $\underline{\mathbf{H}}(z)$ and $\underline{\mathbf{D}}(z)$ may potentially increase with each iterative paraunitary transformation in SBR2 algorithm, the computed spectral factors can accumulate time delays which are unnecessarily large. However, when the outer and inner spectral factors are multiplied together, such delays cancel and the resulting parahermitian polynomial matrix is none the less accurate. This reflects a fundamental indeterminacy in spectral factorization whereby if $\underline{\mathbf{R}}^+(z)$ is a valid outer spectral factor of $\underline{\mathbf{R}}(z)$ so also is $\underline{\mathbf{R}}^+(z)\underline{\mathbf{P}}(z)$ where $\underline{\mathbf{P}}(z)$ represents any paraunitary polynomial matrix which preserves the essential properties associated with an outer spectral factor. This includes simple examples such as $\underline{\mathbf{P}}(z) = z^N\mathbf{I}$, $\underline{\mathbf{P}}(z) = \mathbf{Q}$ where $\mathbf{Q}$ is a simple unitary matrix, or the case in which $\underline{\mathbf{P}}(z)$ takes the form of a diagonal matrix with each entry given by a power of z which need not be the same for all entries.

Outline of algorithm

After the input parahermitian polynomial matrix $\underline{\mathbf{R}}(z)$ is diagonalized as shown in equation (7), each entry within $\underline{\mathbf{D}}(z)$ can be expressed as the product of its outer and inner spectral factors, and so we may write

$$\begin{aligned}\underline{\mathbf{D}}(z) &= \text{diag}\{\underline{d}_1(z), \underline{d}_2(z), \cdots, \underline{d}_m(z)\} = \\ &\text{diag}\{\underline{d}_1^+(z), \underline{d}_2^+(z), \cdots, \underline{d}_m^+(z)\}\text{diag}\{\underline{d}_1^-(z), \underline{d}_2^-(z), \cdots, \underline{d}_m^-(z)\} \\ &= \underline{\mathbf{D}}^+(z)\underline{\mathbf{D}}^-(z) \quad ,\end{aligned} \tag{10}$$

where $\underline{d}_i^+(z)$ and $\underline{d}_i^-(z)$ are the outer and inner spectral factors of $\underline{d}_i(z)$ respectively, $i \in \{1, 2, \cdots, m\}$. By applying the inverse decomposition to equation (7), we get

$$\underline{\mathbf{R}}(z) \approx \tilde{\underline{\mathbf{H}}}(z)\underline{\mathbf{D}}(z)\underline{\mathbf{H}}(z) \quad , \tag{11}$$

and on substituting (10) into (11), this equation can be rewritten as

$$\underline{\mathbf{R}}(z) = \underline{\mathbf{R}}^+(z)\underline{\mathbf{R}}^-(z) \approx \tilde{\underline{\mathbf{H}}}(z)\underline{\mathbf{D}}^+(z)\underline{\mathbf{D}}^-(z)\underline{\mathbf{H}}(z) \quad , \tag{12}$$

where $\underline{\mathbf{R}}^{+}(z)$ and $\underline{\mathbf{R}}^{-}(z)$ denote the final outer and inner spectral factors of $\underline{\mathbf{R}}(z)$ respectively. Therefore, $\underline{\mathbf{R}}^{+}(z)$ can be estimated as $\underline{\tilde{\mathbf{H}}}(z)\underline{\mathbf{D}}^{+}(z)$, and $\underline{\mathbf{R}}^{-}(z)$ as $\underline{\mathbf{D}}^{-}(z)\underline{\mathbf{H}}(z)$ which is the paraconjugate of $\underline{\tilde{\mathbf{H}}}(z)\underline{\mathbf{D}}^{+}(z)$.

V. Simulation Results

In order to demonstrate this method, the 2×2 parahermitian polynomial matrix example used by Janashia [5] has been tackled using our algorithm. In this example we have

$$\underline{\mathbf{R}}_1(z) = \begin{bmatrix} 2z^{-1}+6+2z & 7z^{-1}+22+11z \\ 11z^{-1}+22+7z & 38z^{-1}+84+38z \end{bmatrix}. \quad (13)$$

The SBR2 algorithm was used to diagonalize $\underline{\mathbf{R}}_1(z)$ with a suitable trim function [1] to eliminate any redundant zero coefficients. The resulting diagonal polynomial matrix $\underline{\mathbf{D}}_1(z)$ is shown by means of the stem plot in Fig. 1, corresponding to the numerical result

$$\underline{\mathbf{D}}_1(z) = \begin{bmatrix} 40z^{-1}+90+40z & 0 \\ 0 & -0.01z^{-1}+0.03-0.01z \end{bmatrix}. \quad (14)$$

This confirms that the input $\underline{\mathbf{R}}_1(z)$ is almost generically rank deficient but not quite. Accordingly the inner spectral factor $\underline{\mathbf{D}}_1^{-}(z)$ obtained by two separate applications of the $spf(\cdot)$ function is

$$\underline{\mathbf{D}}_1^{-}(z) = \begin{bmatrix} 4.9+8.1z & 0 \\ 0 & -0.077+0.14z \end{bmatrix}. \quad (15)$$

These results are all quoted to the standard accuracy given by PolyX. The final inner spectral factor $\underline{\mathbf{R}}_1^{-}(z)$ obtained by forming the product $\underline{\mathbf{D}}_1^{-}(z)\underline{\mathbf{H}}_1(z)$ is given by

$$\underline{\mathbf{R}}_1^{-}(z) = \begin{bmatrix} 2.2729z^{8}+0.8964z^{9} & 7.7568z^{8}+4.8817z^{9} \\ -0.1317z^{8}+0.0748z^{9} & 0.0402z^{8}-0.0163z^{9} \end{bmatrix}. \quad (16)$$

The results in (16) only show the coefficient values for z^8 and z^9 which are dominant. The stem plot in Fig. 2 depicts all the coefficients from order 0 to order 21. As can be seen, the spectral factor generated has lots of very small values which are effectively zero. In theory, if the polynomial matrix goes from order $-t$ to t, the spectral factor should be either from order 0 to t or $-t$ to 0. In our case, the polynomial order of the spectral factor was broadened due to the paraunitary similarity transformation in SBR2 algorithm. Truncating the very small coefficients was found to have very little impact on the accuracy of the reconstituted parahermitian polynomial matrix (almost identical to $\underline{\mathbf{R}}_1(z)$ formed by the product of these spectral factors).

The accuracy of the proposed algorithm was assessed by calculating the energy difference between the input parahermitian polynomial matrix and the corresponding polynomial matrix $\underline{\mathbf{R}}_1'(z)$ generated by the product $\underline{\mathbf{R}}_1^{+}(z)\underline{\mathbf{R}}_1^{-}(z)$. The energy of a polynomial matrix is defined here as the sum of the squared Frobenius norm of its coefficient matrices. This takes the form

$$\|\underline{\mathbf{R}}(z)\|_{\mathrm{F}}^{2} = \sum_{\tau}\sum_{k=1}^{m}\sum_{l=1}^{m}|r_{kl}(\tau)|^2 \quad , \quad (17)$$

where $r_{kl}(\tau)$ denotes the element in the k-th row and l-th column of the coefficient matrix for $z^{-\tau}$, $k, l \in \{1, 2, \cdots, m\}$.

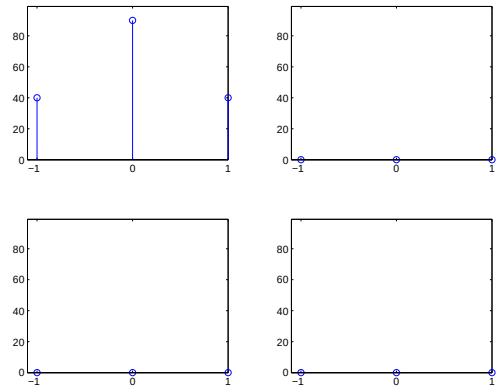

Fig. 1. The diagonalised polynomial matrix for example (13)

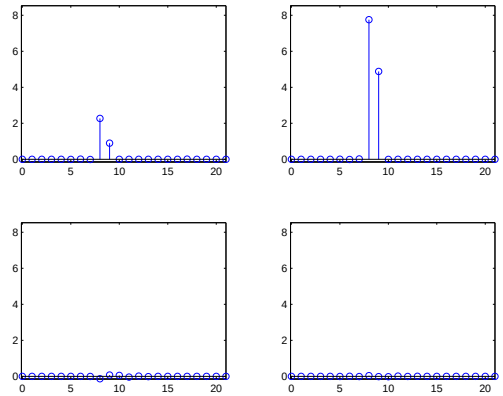

Fig. 2. The inner spectral factor for example (13)

For the example problem in (13), the input energy $\mathcal{E}_1$ is given by

$$\mathcal{E}_1 = \|\underline{\mathbf{R}}_1(z)\|_{\mathrm{F}}^{2} \approx 11296 \quad , \quad (18)$$

whereas the energy difference $\Delta\mathcal{E}_1$ is computed as

$$\Delta\mathcal{E}_1 = \|\underline{\mathbf{R}}_1(z) - \underline{\mathbf{R}}_1'(z)\|_{\mathrm{F}}^{2} \approx 3.95\times10^{-5} \quad . \quad (19)$$

The algorithm has been tested further by means of another, more realistic example. A 3×5 MIMO propagation channel with fast fading was modelled. The convolutive mixing was represented by a 5×3 polynomial matrix with coefficients selected randomly from a uniform distribution in the range $(-1, 1)$. The source signals were represented by independent, identically distributed sequences for which each sample was assigned the value ±1 with probability $1/2$. Gaussian random noise was added to the received signals with a signal-to-noise ratio (SNR) of 2.55 dB in the numerical experiment reported here. The CSD matrix $\underline{\mathbf{R}}_2(z)$ computed from the received signals is plotted in Fig. 3 with polynomial order of 14. The SBR2 algorithm was applied, leading to the diagonal matrix $\underline{\mathbf{D}}_2(z)$ plotted in Fig. 4. The inner spectral factor $\underline{\mathbf{R}}_2^{-}(z)$ generated by the product of $\underline{\mathbf{D}}_2^{-}(z)\underline{\mathbf{H}}_2(z)$, and suitably trimmed, is shown in Fig. 5.

The reconstituted parahermitian polynomial matrix given by $\underline{\mathbf{R}}_2'(z) = \underline{\mathbf{R}}_2^{+}(z)\underline{\mathbf{R}}_2^{-}(z)$ was computed as before. In this case, the input energy $\mathcal{E}_2$ and energy difference $\Delta\mathcal{E}_2$ took the

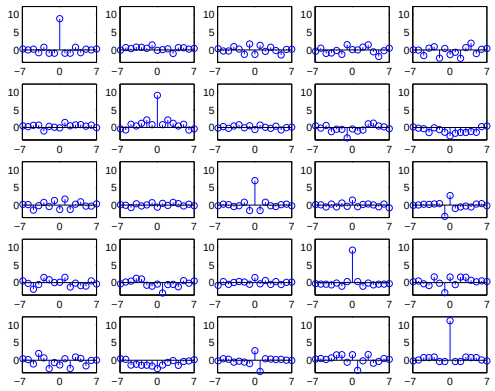

Fig. 3. Cross spectral density matrix for five mixed signals with SNR 2.55 dB.

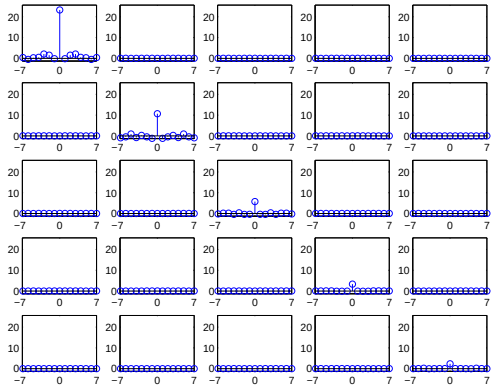

Fig. 4. Diagonalized cross spectral density matrix generated using SBR2 algorithm

values

$$\begin{aligned} \mathcal{E}_2 &= \|\underline{\mathbf{R}}_2(z)\|_{\mathrm{F}}^{\ 2} \approx 730.1825 \\ \Delta\mathcal{E}_2 &= \|\underline{\mathbf{R}}_2(z) - \underline{\mathbf{R}}'_2(z)\|_{\mathrm{F}}^{\ 2} \approx 0.1356 \end{aligned} \tag{20}$$

It is clear that the value of the energy difference $\Delta\mathcal{E}_2$ is very small compared to the total energy $\mathcal{E}_2$ in this convolutive mixing example. Thus the outer (inner) spectral factors have been generated to a high degree of accuracy using the novel method presented in this paper.

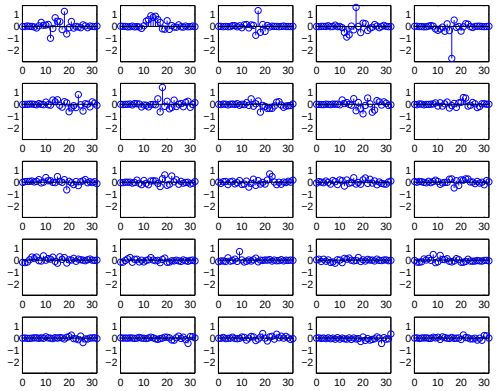

Fig. 5. The inner spectral factor of cross spectral density matrix for five mixed signals

VI. Conclusion

In conclusion, the proposed multichannel spectral factorization algorithm has been shown achieve a high degree of accuracy in terms of recovering the input parahermitian polynomial matrix from its outer and inner spectral factors. The algorithm is seen to offer a significant advantage in that the multichannel spectral factorization problem is reduced to a number of independent single channel problems for which suitable algorithms already exist. However, it must be noted that the spectral factors generted by this algorithm can accumulate numerous very small or zero coefficients which may be a nuisance but can be truncated by virtue of the fundamental indeterminacy associated with spectral factorization.

Acknowledgment

The authors would like to thank the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/K014307/1 and the MOD University Defence Research Collaboration in Signal Processing for partially supporting this work.

References

[1] J. McWhirter, P. Baxter, T. Cooper, S. Redif, and J. Foster, "An EVD algorithm for para-hermitian polynomial matrices," *Signal Processing, IEEE Transactions on*, vol. 55, no. 5, pp. 2158–2169, May 2007.

[2] C. Charoenlarpnopparut, "One-dimensional and multidimensional spectral factorization using Grobner basis approach," in *Communications, 2007. APCC 2007. Asia-Pacific Conference on*, Oct 2007, pp. 201–204.

[3] R. F. Fischer, "Sorted spectral factorization of matrix polynomials in MIMO communications," *IEEE Transactions on Communications*, vol. 12, pp. 945–951, June 2005.

[4] G. Wilson, "The factorization of matricial spectral densities," *SIAM Journal on Applied Mathematics*, vol. 23, no. 4, pp. 420–426, 1972. [Online]. Available: http://dx.doi.org/10.1137/0123044

[5] G. Janashia, E. Lagvilava, and L. Ephremidze, "A new method of matrix spectral factorization," *Information Theory, IEEE Transactions on*, vol. 57, no. 4, pp. 2318–2326, April 2011.

[6] L. Ephremidze, G. Janashia, and E. Lagvilava, "A new computational algorithm of spectral factorization for polynomial matrix-functions," in *Proceedings of A. Razmadze Mathematical Institute*, vol. 136, 2004, pp. 41–46.

[7] L. Ephremidze, G. Janashia, and E. Lagvilava, "A new efficient matrix spectral factorization algorithm," in *SICE, 2007 Annual Conference*, Sept 2007, pp. 20–23.

[8] A. H. Sayed and T. Kailath, "A survey of spectral factorization methods," *Numerical Linear Algebra with Applications*, vol. 8, no. 67, pp. 467–496, 2001.

[9] T. Goodman, C. Micchelli, G. Rodriguez, and S. Seatzu, "Spectral factorization of Laurent polynomials," *Advances in Computational Mathematics*, vol. 7, no. 4, pp. 429–454, 1997.

[10] I. Selesnick. Spectral factorization - polynomial root finding and the leja ordering. [Online]. Available: http://eeweb.poly.edu/iselesni/EL713/sfact/sfact.pdf

[11] PolyX, *The Polynomial Toolbox for MATLAB*, 2nd ed., PolyX, Ltd, Czech Republic.

[12] C. H. Ta and S. Weiss, "A design of precoding and equalisation for broadband MIMO systems," in *Digital Signal Processing, 2007 15th International Conference on*, July 2007, pp. 571–574.

[13] S. Redif, J. McWhirter, and S. Weiss, "Design of FIR paraunitary filter banks for subband coding using a polynomial eigenvalue decomposition," *Signal Processing, IEEE Transactions on*, vol. 59, no. 11, pp. 5253–5264, Nov 2011.