

MULTIPLE SHIFT SECOND ORDER SEQUENTIAL BEST ROTATION ALGORITHM FOR POLYNOMIAL MATRIX EVD

Zeliang Wang*, John G. McWhirter*, Jamie Corr†, Stephan Weiss†

* School of Engineering
Cardiff University
Cardiff, Wales, UK

† Department of Electronic & Electrical Engineering
University of Strathclyde
Glasgow, Scotland, UK

ABSTRACT

In this paper, we present an improved version of the second order sequential best rotation algorithm (SBR2) for polynomial matrix eigenvalue decomposition of para-Hermitian matrices. The improved algorithm is entitled multiple shift SBR2 (MS-SBR2) which is developed based on the original SBR2 algorithm. It can achieve faster convergence than the original SBR2 algorithm by means of transferring more off-diagonal energy onto the diagonal at each iteration. Its convergence is proved and also demonstrated by means of a numerical example. Furthermore, simulation results are included to compare its convergence characteristics and computational complexity with the original SBR2, sequential matrix diagonalization (SMD) and multiple shift maximum element SMD algorithms.

Index Terms— Polynomial matrix eigenvalue decomposition, multiple shift SBR2.

1. INTRODUCTION

The conventional EVD algorithm is suitable for diagonalizing the covariance matrix of narrowband signals. When the problem is extended to broadband scenarios, polynomial matrix eigenvalue decomposition (PEVD) techniques need to be taken into account. The idea of the PEVD is generalized as [1]

$$\underline{\mathbf{H}}(z)\underline{\mathbf{R}}(z)\tilde{\underline{\mathbf{H}}}(z) \approx \underline{\mathbf{D}}(z), \quad (1)$$

where $\underline{\mathbf{H}}(z)$ is a paraunitary matrix such that $\underline{\mathbf{H}}(z)\tilde{\underline{\mathbf{H}}}(z) = \tilde{\underline{\mathbf{H}}}(z)\underline{\mathbf{H}}(z) = \mathbf{I}$, and $\underline{\mathbf{R}}(z)$ is the input para-Hermitian matrix. $\underline{\mathbf{D}}(z)$ is (ideally) a diagonal polynomial matrix. Throughout this paper, polynomial matrices are denoted as underscored boldface capital letters, and the notation of $\tilde{\cdot}$ upon a polynomial matrix denotes the paraconjugate operation. In the case of broadband sensor arrays, the paraunitary matrix

$\underline{\mathbf{H}}(z)$ acts as a multichannel all-pass filter, and it aims to diagonalize the para-Hermitian matrix $\underline{\mathbf{R}}(z)$ by means of paraunitary similarity transformation while still preserving the total signal energy [2]. Assuming the received signals $\mathbf{x}(t)$ have zero mean, the space-time covariance matrix is given by $\mathbf{R}(\tau) = E\{\mathbf{x}(t)\mathbf{x}^H(t-\tau)\}$, where $E\{\cdot\}$ denotes the expectation, the superscript H denotes Hermitian transpose, and $\tau \in \mathbb{Z}$. Then the cross-spectral density (CSD) matrix can be defined in terms of the z -transform given by $\underline{\mathbf{R}}(z) = \sum_{\tau} \mathbf{R}(\tau)z^{-\tau}$. Note that the CSD matrix is para-symmetric, since it satisfies $\tilde{\underline{\mathbf{R}}}(z) = \underline{\mathbf{R}}(z)$. Here $\tilde{\underline{\mathbf{R}}}(z)$ is the paraconjugate of $\underline{\mathbf{R}}(z)$, defined as $\tilde{\underline{\mathbf{R}}}(z) = \underline{\mathbf{R}}^H(1/z)$, i.e. performing Hermitian transpose of the polynomial coefficient matrix $\mathbf{R}(\tau)$ and time-reversing all entries inside.

PEVD techniques have attracted lots of interest in digital signal processing and communications over the past few years. Applications of PEVD have been found in areas, such as strong decorrelation of the signals received by broadband sensor arrays [1], broadband angle of arrival estimation [3,4], subband coding [5], precoding and equalization for multiple-input multiple-output (MIMO) communications [6], spectral factorization [7], and blind source separation [11–13] etc.

Besides the original SBR2 algorithm [1], a number of SMD versions have emerged, including the original SMD algorithm [8] and its improved versions, multiple shift maximum element SMD (MSME-SMD) [9] and the causality-constrained multiple shift SMD [10] algorithm. The SMD family can achieve better diagonalization than SBR2 at the expense of a higher computational cost. Therefore, the aim of this paper is to see whether some of the ideas of MSME-SMD can be harnessed to create a faster converging version of SBR2 that however still enjoys the SBR2 family's low complexity.

As to the following parts of the paper, we start by briefly reviewing some existing PEVD algorithms in Sec. 2. Sec. 3 presents the proposed algorithm. Sec. 4 shows the results of the comparison among different PEVD algorithms. Conclusions are drawn in Sec. 5.

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.

2. EXISTING PEVD ALGORITHMS

2.1. Second Order Sequential Best Rotation Algorithm

For each iteration, the SBR2 algorithm [1] starts by locating the maximum off-diagonal element. An elementary delay matrix and Jacobi rotation are applied to bring the element onto the zero-lag coefficient matrix $\mathbf{R}^{(i-1)}(0) \in \mathbb{C}^{M \times M}$, and then rotate its energy onto the diagonal. Here the superscript i denotes the iteration index. To find the maximum off-diagonal element, we define a matrix $\mathbf{S}^{(i)}(\tau)$, which contains only the upper triangular elements in $\mathbf{R}^{(i-1)}(\tau)$ with the remaining elements set to zero. Thus the location of the maximum off-diagonal element $s_{jk}^{(i)}(\tau)$, ($j < k$) found at i -th iteration satisfies

$$\{j^{(i)}, k^{(i)}, \tau^{(i)}\} = \arg \max_{j,k,\tau} \|\mathbf{S}^{(i)}(\tau)\|_{\infty}, \quad (2)$$

where $j^{(i)}$, $k^{(i)}$ and $\tau^{(i)}$ are the corresponding row, column and time lag index. Assuming $\underline{\mathbf{P}}^{(i)}(z)$ and $\underline{\mathbf{Q}}^{(i)}$ denote the elementary delay and rotation matrix respectively, the maximum off-diagonal element $r_{jk}^{(i)}(\tau)$ and its complex conjugate $r_{kj}^{(i)}(-\tau)$ can be transferred onto the diagonal of the zero-lag ($\tau = 0$) matrix $\mathbf{R}^{(i)}(0)$ by performing the following transformations.

$$\underline{\mathbf{R}}'^{(i)}(z) = \underline{\mathbf{P}}^{(i)}(z)\underline{\mathbf{R}}^{(i-1)}(z)\tilde{\underline{\mathbf{P}}}^{(i)}(z), \quad (3)$$

$$\underline{\mathbf{R}}^{(i)}(z) = \underline{\mathbf{Q}}^{(i)}\underline{\mathbf{R}}'^{(i)}(z)\underline{\mathbf{Q}}^{\text{H}(i)}, \quad (4)$$

where $\underline{\mathbf{R}}'^{(i)}(z)$ stands for the intermediate variable for the elementary delay operation. Let $\underline{\mathbf{E}}^{(i)}(z)$ be the elementary paraunitary matrix at the i -th iteration, then it can be expressed as

$$\underline{\mathbf{E}}^{(i)}(z) = \underline{\mathbf{Q}}^{(i)}\underline{\mathbf{P}}^{(i)}(z). \quad (5)$$

The algorithm continues its iterative process until all the off-diagonal elements are smaller than a given threshold ϵ which can be set to a very small value to achieve sufficient accuracy. Assuming that the algorithm has converged at the N -th iteration, the diagonalized para-Hermitian matrix in (1) takes the form of

$$\underline{\mathbf{D}}(z) = \text{diag}\{\underline{d}_1(z), \underline{d}_2(z), \dots, \underline{d}_M(z)\}, \quad (6)$$

and the generated paraunitary polynomial matrix is given by

$$\underline{\mathbf{H}}(z) = \prod_{i=1}^N \underline{\mathbf{E}}^{(i)}(z) = \underline{\mathbf{E}}^{(N)}(z) \cdots \underline{\mathbf{E}}^{(2)}(z)\underline{\mathbf{E}}^{(1)}(z). \quad (7)$$

2.2. Sequential Matrix Diagonalization Algorithm

Unlike the SBR2 algorithm, the SMD algorithm [8] requires a initialization step to diagonalize the zero-lag coefficient matrix $\mathbf{R}^{(0)}(0)$ before all iterations. This is implemented by computing a full EVD to $\mathbf{R}^{(0)}(0)$ and then applying the corresponding modal matrix to the rest of coefficient matrices

$\mathbf{R}^{(0)}(\tau)$, $\tau \neq 0$. For the i -th iteration, it starts by locating the column that contains the maximum off-diagonal energy, and then according to the location information $k^{(i)}$ and $\tau^{(i)}$, it shifts the corresponding row and column pair onto the zero-lag plane. As to the rotation step, rather than just using a single Jacobi rotation as with SBR2, the SMD algorithm computes a full EVD operation for the zero-lag matrix $\mathbf{R}'^{(i)}(0)$.

2.3. Multiple Shift Maximum Element SMD Algorithm

The MSME-SMD algorithm [9] introduced a distinguishing search and shift strategy, which can shift more energy than both the SBR2 and SMD onto the diagonal at each iteration. For each iteration, more than one maximum off-diagonal element is found by using a reduced search space strategy. Every row and column pair containing a maximum off-diagonal element will then be shifted to the zero-lag matrix. This is different to the way the SMD algorithm operates. The SMD algorithm always shifts the row and column pair containing the maximum off-diagonal energy rather than the maximum off-diagonal element as in MSME-SMD. At the rotation step, the MSME-SMD algorithm follows the same procedure as the SMD algorithm transferring all the off-diagonal elements in $\mathbf{R}'^{(i)}(0)$ onto the diagonal.

3. MULTIPLE SHIFT SBR2 ALGORITHM

3.1. Outline of Algorithm

The multiple shift SBR2 algorithm (MS-SBR2) was developed based on the SBR2 algorithm, with an additional energy transferred onto the diagonal in every iteration step akin to the MSME-SMD algorithm, so that it can achieve the diagonalization with less iterations. With MS-SBR2, there are two main steps involved at each iteration. The first step involves multiple shifts operations, and the second step is to perform a sequence of Jacobi rotations corresponding to the multiple shifts. Therefore for the i -th iteration, the elementary paraunitary matrix in MS-SBR2 can be defined as

$$\hat{\underline{\mathbf{E}}}^{(i)}(z) = \hat{\underline{\mathbf{Q}}}^{(i)}\hat{\underline{\mathbf{P}}}^{(i)}(z) = \prod_{l=1}^{L^{(i)}} \underline{\mathbf{Q}}^{(l,i)} \prod_{l=1}^{L^{(i)}} \underline{\mathbf{P}}^{(l,i)}(z), \quad (8)$$

where $\hat{\underline{\mathbf{Q}}}^{(i)} = \prod_{l=1}^{L^{(i)}} \underline{\mathbf{Q}}^{(l,i)}$, $\hat{\underline{\mathbf{P}}}^{(i)}(z) = \prod_{l=1}^{L^{(i)}} \underline{\mathbf{P}}^{(l,i)}(z)$ and $L^{(i)}$ denotes the total number of off-diagonal elements shifted to the zero-lag coefficient matrix at the i -th iteration. The resulting para-Hermitian matrix at this iteration can be computed by performing the similarity transform as

$$\underline{\mathbf{R}}'^{(i)}(z) = \hat{\underline{\mathbf{P}}}^{(i)}(z)\underline{\mathbf{R}}^{(i-1)}(z)\tilde{\hat{\underline{\mathbf{P}}}}^{(i)}(z), \quad (9)$$

$$\underline{\mathbf{R}}^{(i)}(z) = \hat{\underline{\mathbf{Q}}}^{(i)}\underline{\mathbf{R}}'^{(i)}(z)\hat{\underline{\mathbf{Q}}}^{\text{H}(i)}. \quad (10)$$

The MS-SBR2 algorithm is summarized in Table 1.

1:	Input $M \times M$ para-Hermitian polynomial matrix $\underline{\mathbf{R}}(z)$.
2:	Compute the square of Frobenius norm (F-norm): $\mathcal{N}_4 = \ \underline{\mathbf{R}}(z)\ _F^2$, i.e. the sum of squares of the entries of all the polynomial coefficient matrices $\mathbf{R}(\tau)$.
3:	Specify maximum number of iterations, I , convergence parameter, ϵ , and trim factor μ .
4:	Initialize variables: $iter \leftarrow 0$, $\underline{\mathbf{H}}^{(0)}(z) \leftarrow \mathbf{I}$ and $g \leftarrow 1 + \epsilon$.
5:	while $iter < I$ and $g > \epsilon$
6:	Step 1: locate and shift multiple off-diagonal elements
7:	Initialize variables: the total number of off-diagonal elements shifted at the i -th iteration, $L^{(i)} \leftarrow 0$.
8:	while $g > \epsilon$ and the search space exists
9:	Define the search space in $\underline{\mathbf{R}}^{(i)}(z)$, and locate the maximum off-diagonal element, $r_{jk}^{(i)}(t)$.
10:	Set $g = r_{jk}^{(i)}(t) $
11:	if $g > \epsilon$
12:	Update $\underline{\mathbf{R}}'^{(i)}(z) = \underline{\mathbf{P}}^{(i)}(z)\underline{\mathbf{R}}^{(i-1)}(z)\tilde{\underline{\mathbf{P}}}^{(i)}(z)$, $\underline{\mathbf{H}}'^{(i)}(z) = \underline{\mathbf{P}}^{(i)}(z)\underline{\mathbf{H}}^{(i-1)}(z)$ according to (3).
13:	$L^{(i)} \leftarrow L^{(i)} + 1$
14:	end
15:	end
16:	Step 2: perform a sequence of Jacobi rotations
17:	for $l = 1 : L^{(i)}$
18:	Update $\underline{\mathbf{R}}^{(i)}(z) = \mathbf{Q}^{(i)}\underline{\mathbf{R}}'^{(i)}(z)\mathbf{Q}^{H(i)}$, and $\underline{\mathbf{H}}^{(i)}(z) = \mathbf{Q}^{(i)}\underline{\mathbf{H}}'^{(i)}(z)$ according to (4).
19:	end
20:	$iter \leftarrow iter + 1$
21:	Trim the polynomial order: $\underline{\mathbf{R}}^{(i)}(z) \leftarrow trim(\underline{\mathbf{R}}^{(i)}(z), \mu)$, according to [1].
22:	end

Table 1. Summary of the MS-SBR2 Algorithm

The search strategy for MS-SBR2 differs from the one employed in SBR2, and it is based on a set of reduced search spaces. Since each Jacobi rotation operation will act on both the columns and rows j and k of the maximum off-diagonal element $r_{jk}^{(i)}(0)$ at the zero-lag plane, the search space for the next off-diagonal element will have to exclude those two columns and rows to avoid the previous off-diagonal element being affected.

Assuming the input para-Hermitian matrix has dimension 6×6 , the first off-diagonal element at the i -th iteration can be located according to (2). Once the first element a and its complex conjugate a^* have been shifted to the zero-lag matrix as shown in Fig. 1(a), the gray areas shown in Fig. 1(b) will be eliminated from the search space of the next off-diagonal element, which leads the white parts of the upper triangular area to be the search space of the second element. By continuing from Fig. 1(b), if the second element b was found at row 3 and column 6, its complex conjugate b^* will be at row 6 and column 3 according to the para-Hermitian property. After bringing them to the zero-lag matrix as shown in Fig. 1(c), the search space for the third element c^* will be the remaining position row 1 and column 5 shown in Fig. 1(d). This search strategy applies to a general $M \times M$ para-Hermitian matrix. Generally speaking, there are at most $M/2$ off-diagonal elements which can be located at each iteration. Therefore, in the case of $M \leq 3$, the MS-SBR2 algorithm is identical to

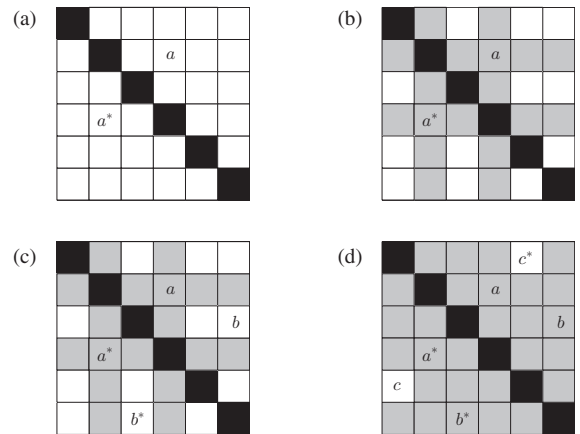


Fig. 1. MS-SBR2 search strategy for a 6×6 para-Hermitian matrix showing (a) the first maximum element found, (b) the reduced search spaces, (c) the second maximum chosen, and (d) the last element found.

the SBR2 algorithm. In other words, if there is only one off-diagonal elements shifted to the zero-lag coefficient matrix at each iteration, i.e. $L^{(i)} = 1$, then MS-SBR2 reduces to SBR2.

The convergence proof of the MS-SBR2 algorithm is very similar to that of the original SBR2 algorithm [1] and the es-

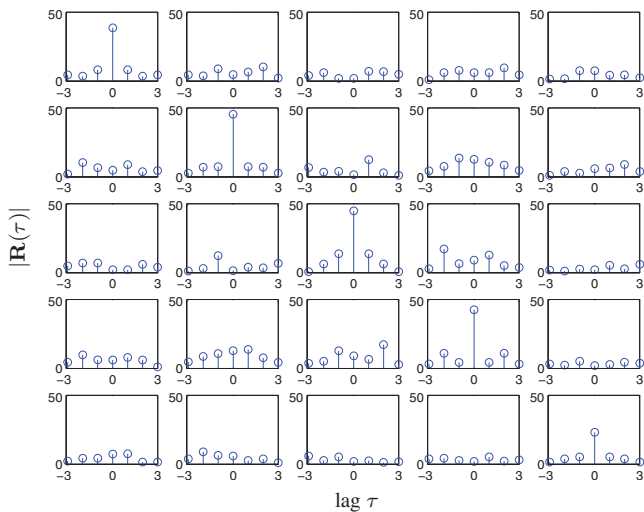


Fig. 2. Example of 5×5 para-Hermitian polynomial matrix in Sec. 3.2

sential difference is that for MS-SBR2 we have

$$\gamma^{(i)} = \sum_{l=1}^{L^{(i)}} |r_{j^{(l)}k^{(l)}}(\tau^{(l)})|^2 \quad (11)$$

representing the norm of all the off-diagonal elements found at the i -th iteration.

3.2. Worked Example

In order to demonstrate its capability of diagonalizing a para-Hermitian matrix, a 5×5 para-Hermitian matrix $\underline{\mathbf{R}}(z)$ with polynomial order of 7 was taken as the input for testing this algorithm. This random matrix was generated from a matrix $\underline{\mathbf{A}}(z) \in \mathbb{C}^{5 \times 5}$ of order 4 with independent, identically distributed zero mean unit variance complex Gaussian entries, with $\underline{\mathbf{R}}(z) = \underline{\mathbf{A}}(z)\underline{\mathbf{A}}^H(z)$. The input parameters have been set as $I = 2000$, $\epsilon = 10^{-3}$, and $\mu = 10^{-4}$.

The stem plot in Fig. 2 shows the magnitudes of all the elements in $\underline{\mathbf{R}}(\tau)$, $\tau \in \{-3, -2, \dots, 2, 3\}$, and Fig. 3 depicts the magnitudes of the elements in the final diagonalized para-Hermitian matrix $\underline{\mathbf{D}}(z)$. In this case, the MS-SBR2 algorithm terminated after 167 th iterations with the converged value of 0.001 as shown in Fig. 4, bearing in mind that only the first maximum off-diagonal element found at each iteration was used to generate the convergence plot, since it is the maximum element found at that iteration.

4. RESULTS

The proposed algorithm was assessed in terms of the normalized remaining off-diagonal energy at the i -th iteration. This is defined as

$$\eta_{\text{norm}}^{(i)} = \frac{\sum_{\tau} \sum_{m=1}^M \sum_{n=1, n \neq m}^M |r_{mn}^{(i)}(\tau)|^2}{\mathcal{N}_4\{\underline{\mathbf{R}}(z)\}}, \quad (12)$$

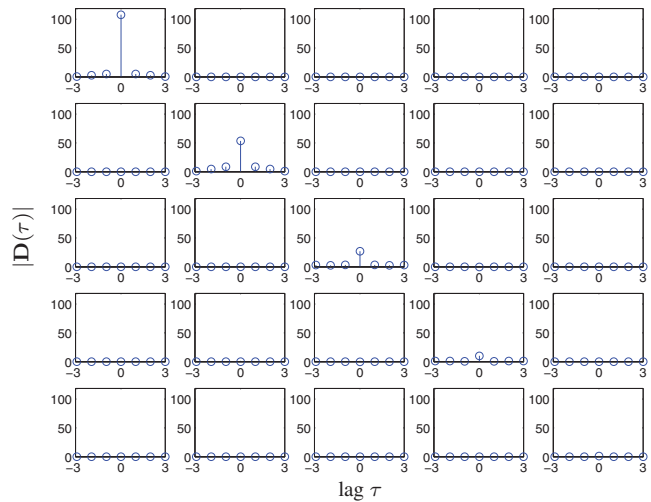


Fig. 3. Diagonalized polynomial matrix (trimmed) obtained using the MS-SBR2 algorithm

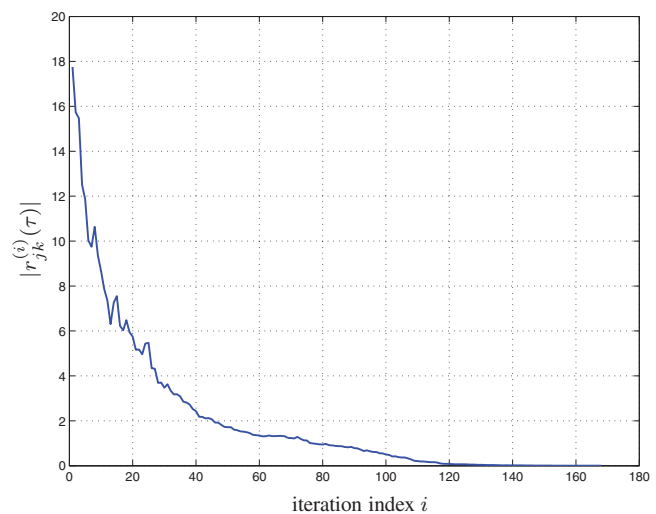


Fig. 4. Convergence plot of MS-SBR2 for the example in Sec. 3.2

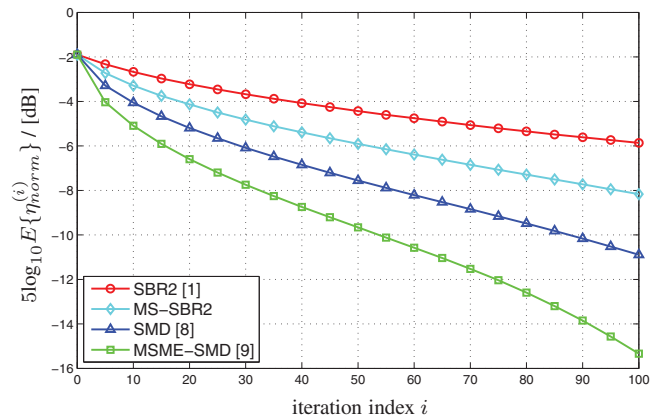


Fig. 5. Comparison of normalized off-diagonal energy among SBR2, MS-SBR2, SMD and MSME-SMD, showing ensemble averages versus iterations

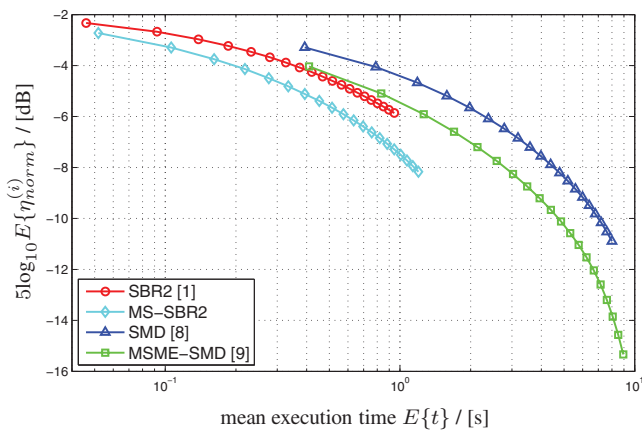


Fig. 6. Normalized off-diagonal energy versus mean execution time for different PEVD algorithms

where the numerator represents the off-diagonal energy at the i -th iteration. The comparison among the SBR2, MS-SBR2, SMD and MSME-SMD algorithms is calculated over an ensemble of 100 realizations of a random 6×6 para-Hermitian matrix $\underline{\mathbf{R}}(z)$ of order 13, which is generated using the same method mentioned in Sec. 3.2. Fig. 5 shows the normalized remaining off-diagonal energy for various PEVD algorithms within 100 iterations. Obviously, both the SMD and MSME-SMD algorithms outperform SBR2 and MS-SBR2 in terms of eliminating the off-diagonal energy, and the MS-SBR2 algorithm performs better than the SBR2 algorithm since more than one off-diagonal element has been transferred onto the diagonal at each iteration by using the distinguishing search strategy.

The mean execution time for each PEVD algorithm has also been measured in order to evaluate the computational cost, and this is implemented in Matlab R2014a on a desktop PC with characteristics Intel(R) Core(TM) i7-3770T CPU@2.50 GHz and 16 GB RAM. The graph shown in Fig. 6 depicts the remaining off-diagonal energy versus mean execution time. With the same level of diagonalization, the MS-SBR2 algorithm requires the lowest calculation cost compared to the rest of the PEVD algorithms. In contrast, the SMD algorithm requires the longest execution time due to the calculation of the column norms for each search step and the full EVD operation at each iteration.

5. CONCLUSION

We have presented an improved SBR2 algorithm for computing the EVD of a para-Hermitian polynomial matrix. Simulation results indicate that the proposed MS-SBR2 algorithm provides faster convergence than the conventional SBR2 algorithm, especially in the case of high dimension para-Hermitian matrices. In addition, the MS-SBR2 algorithm appears to provide a compromise with much lower computational complexity than the SMD family with respect to computing the decomposition.

REFERENCES

- [1] J.G. McWhirter, P.D. Baxter, T. Cooper, S. Redif, and J. Foster, "An EVD Algorithm for Para-Hermitian Polynomial Matrices," *IEEE Trans. SP*, **55**(5):2158–2169, May 2007.
- [2] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice-Hall, 1993.
- [3] M.A. Alrmah, S. Weiss, and S. Lambotharan, "An extension of the MUSIC algorithm to broadband scenarios using polynomial eigenvalue decomposition," in *EU-SIPCO*, Barcelona, Spain, pp. 629–633, Sep. 2011.
- [4] M.A. Alrmah, J. Corr, A. Alzin, K. Thompson, and S. Weiss, "Polynomial subspace decomposition for broadband angle of arrival estimation," in *Sensor Signal Processing for Defence*, Edinburgh, Scotland, pp. 1–5, Sep. 2014.
- [5] S. Redif, J.G. McWhirter, and S. Weiss, "Design of FIR Paraunitary Filter Banks for Subband Coding Using a Polynomial Eigenvalue Decomposition," *IEEE Trans. SP*, **59**(11):5253–5264, Nov. 2011.
- [6] C.H. Ta and S. Weiss, "A Design of Precoding and Equalisation for Broadband MIMO Systems," in *Asilomar Conf. Signals, Systems & Computers*, Pacific Grove, CA, pp. 1616–1620, Nov. 2007.
- [7] Z. Wang and J.G. McWhirter, "A New Multichannel Spectral Factorization Algorithm for Parahermitian Polynomial Matrices," in *10th IMA Int. Conf. Mathematics in Signal Processing*, Birmingham, England, Dec. 2014.
- [8] S. Redif, S. Weiss, and J.G. McWhirter, "Sequential Matrix Diagonalization Algorithms for Polynomial EVD of Parahermitian Matrices," *IEEE Trans. SP*, **63**(1):81–89, Jan. 2015.
- [9] J. Corr, K. Thompson, S. Weiss, J.G. McWhirter, S. Redif, and I.K. Proudler, "Multiple shift maximum element sequential matrix diagonalisation for parahermitian matrices," in *IEEE SSP*, Gold Coast, Australia, pp. 312–315, Jun. 2014.
- [10] J. Corr, K. Thompson, S. Weiss, J.G. McWhirter, and I.K. Proudler, "Causality-constrained multiple shift sequential matrix diagonalisation for parahermitian matrices," in *EUSIPCO*, Lisbon, Portugal, pp. 1277–1281, Sep. 2014.
- [11] P. Comon, and L. Rota, "Blind separation of independent sources from convolutive mixtures," *IEICE Trans. FECCS*, **E86-A**(3):542–549, Mar. 2003.
- [12] S. Icart, P. Comon, and L. Rota, "Blind paraunitary equalization," *Signal Processing*, **89**:283–290, 2009.
- [13] M. Sorensen, L. De Lathauwer, S. Icart, and L. Deneire, "On Jacobi-type methods for blind equalization of paraunitary channels," *Signal Processing*, **92**:617–624, 2012.