SECOND ORDER SEQUENTIAL BEST ROTATION ALGORITHM WITH HOUSEHOLDER REDUCTION FOR POLYNOMIAL MATRIX EIGENVALUE DECOMPOSITION

Vincent W. Neo and Patrick A. Naylor

Department of Electrical and Electronic Engineering, Imperial College London, UK
Email: {vincent.neo09, p.naylor}@imperial.ac.uk

ABSTRACT

The Second-order Sequential Best Rotation (SBR2) algorithm, used for Eigenvalue Decomposition (EVD) on para-Hermitian polynomial matrices typically encountered in wideband signal processing applications like multichannel Wiener filtering and channel coding, involves a series of delay and rotation operations to achieve diagonalisation. In this paper, we proposed the use of Householder transformations to reduce polynomial matrices to tridiagonal form before zeroing the dominant element with rotation. Similar to performing Householder reduction on conventional matrices, our method enables SBR2 to converge in fewer iterations with smaller order of polynomial matrix factors because more off-diagonal Frobenius-norm (F-norm) could be transferred to the main diagonal at every iteration. A reduction in the number of iterations by 12.35% and 0.1% improvement in reconstruction error is achievable.

***Index Terms*—** Polynomial matrix eigenvalue decomposition, convolutive mixing, paraunitary lossless systems, strong decorrelation, wideband signal processing.

1. INTRODUCTION

The Eigenvalue Decomposition (EVD) of Hermitian matrices is widely used in many important signal processing applications such as subspace decomposition for data compression [1], noise reduction [2], spectral estimation [3], blind source separation [4] and adaptive beamforming [5, 6]. The matrix used is usually the multichannel spatial covariance matrix, computed using the complex outer product of the instantaneous data vector. This computation assumes that sources are narrowband, uncorrelated and zero-mean.

When wideband signals are involved, time delays between sensors cannot be simply modelled as phase shifts. Instead, correlation across different sensors and temporal lags need to be considered giving rise to space-time covariance matrices, usually modelled as polynomial matrices. This has motivated the development of a family of Polynomial Matrix Eigenvalue Decomposition (PEVD) algorithms [7–10], based on the Second-order Sequential Best Rotation (SBR2) [11], for wideband signal processing applications like polynomial MUltiple SIgnal Classification (MUSIC) [12, 13], adaptive beamforming [14], multichannel Wiener filtering [15], blind source separation [16], channel identification [17, 18] and channel coding [19]. At every iteration, SBR2 searches for the dominant off-diagonal element and seeks to zero it out using elementary delay and rotation operations. It exploits the para-Hermitian symmetry and is akin to Jacobi's eigenvalue algorithm [20]. Even though the SBR2 has been proven to converge, it tends to take many iterations because zeroing occurs pairwise during each step and the rotation applied to other lags may undo previous diagonalisation efforts.

While there were attempts to achieve one-step diagonalisation for faster convergence using a Householder-like factorisation, no closed form solution is available [7, 21]. Another approach, the Sequential Matrix Diagonalisation (SMD) [10], performs an EVD on the zero lag plane before applying the same unitary matrices across all lags to achieve quick convergence at the expense of complexity.

Motivated by the idea of reducing the matrix to condensed form in [22–25], we now show how to enhance SBR2 by reducing the zero lag plane to tridiagonal form using elementary reflection before applying rotation to speed up convergence.

2. REVIEW OF EVD ALGORITHMS

2.1. Conventional Hermitian Matrix

The eigenvalue decomposition of a Hermitian matrix [20], $\mathbf{A} = \mathbf{A}^H$ where $\mathbf{A} \in \mathbb{C}^{M\times M}$, can be expressed as

$$\mathbf{A} = \mathbf{U}\Lambda\mathbf{U}^H, \tag{1}$$

where the columns of the unitary matrix, $\mathbf{U}$, contain the orthonormal eigenvectors with associated real-valued distinct eigenvalues lying on the diagonal of $\mathbf{\Lambda}$ and $[.]^H$ is the Hermitian operator.

In numerical diagonalisation such as the Jacobi method, the unitary matrix, $\mathbf{U}$, can be constructed using a sequence of L unitary matrices $\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_L$ with each of them satisfying

$$\mathbf{U}_l\mathbf{U}_l^H = \mathbf{U}_l^H\mathbf{U}_l = \mathbf{I}, \tag{2}$$

where $\mathbf{I}$ is the identity matrix. The unitary matrix, $\mathbf{U}_l$, $l = 1, 2, \ldots, L$, can take the form of, for example, a Givens rotation,

$$\mathbf{G}_{(j,k)}(\theta,\phi) = \begin{bmatrix} \mathbf{I}_{j-1} & \mathbf{0} & \ldots & \ldots & \mathbf{0} \\ \mathbf{0} & \cos\theta & \mathbf{0} & \sin\theta e^{i\phi} & \vdots \\ \vdots & \mathbf{0} & \mathbf{I}_{k-j-1} & \mathbf{0} & \vdots \\ \vdots & -\sin\theta e^{-i\phi} & \mathbf{0} & \cos\theta & \vdots \\ \mathbf{0} & \ldots & \ldots & \ldots & \mathbf{I}_{M-k} \end{bmatrix}, \tag{3}$$

where $\mathbf{0}$ is the zero matrix and $i = \sqrt{-1}$, designed to zero the off-diagonal element pair on the j-th and k-th row or column such that

$$\mathbf{G}_{(j,k)} \begin{bmatrix} \ddots & \vdots & & \vdots & \iddots \\ \ldots & r_{jj} & \ldots & r_{jk} & \ldots \\ & \vdots & \ddots & \vdots & \\ \ldots & r_{kj} & \ldots & r_{kk} & \ldots \\ \iddots & \vdots & & \vdots & \ddots \end{bmatrix} \mathbf{G}_{(j,k)}^H = \begin{bmatrix} \ddots & \vdots & & \vdots & \iddots \\ \ldots & r'_{jj} & \ldots & 0 & \ldots \\ & \vdots & \ddots & \vdots & \\ \ldots & 0 & \ldots & r'_{kk} & \ldots \\ \iddots & \vdots & & \vdots & \ddots \end{bmatrix}, \tag{4}$$

where r_{mn} is the m-th row, n-th column matrix element and $[.]'$ denotes the resulting sub-matrix after multiplications. The values in (3) satisfying (4) are $\phi = \arg(r_{jk})$ and $\tan 2\theta = \frac{2|r_{jk}|}{r_{jj}-r_{kk}}$. Furthermore, the diagonal elements, r'_{jj}, r'_{kk} can be ordered in decreasing magnitude if θ is computed using the four quadrant inverse tangent to avoid permutation ambiguity arising from arbitrary ordering.

The Householder reflection matrix $\mathbf{H}_j$, which also satisfies (2), is designed to zero more than one off-diagonal pair such that

$$\mathbf{H}_j \begin{bmatrix} r_{j,j} & \mathbf{r}^H \\ \mathbf{r} & \mathbf{A}_{j+1} \end{bmatrix} \mathbf{H}_j^H = \begin{bmatrix} r_{j,j} & \mathbf{y}^H \\ \mathbf{y} & \mathbf{A}'_{j+1} \end{bmatrix}, \tag{5}$$

where $\mathbf{r}^H = [r_{j,j+1}, r_{j,j+2}, \ldots, r_{j,M}] \in \mathbb{C}^{1\times M}$ is the row vector, $\mathbf{y}^H = [\alpha, 0, \ldots, 0]$ is the target vector, $\alpha = \pm\|\mathbf{r}\|_2 \exp(i\phi_{r_{j,j+1}})$ is the first element in $\mathbf{y}^H$, $\phi_{r_{j,j+1}}$ is the phase angle of $r_{j,j+1}$. The Householder matrix is computed using $\mathbf{H}_j = \mathbf{I} - 2\frac{\mathbf{u}\mathbf{u}^H}{\mathbf{u}^H\mathbf{u}}$, where $\mathbf{u}^H = \mathbf{r}^H + \mathbf{y}^H$. Sub-matrices $\mathbf{A}_{j+1}$ and $\mathbf{A}'_{j+1}$ contain arbitrary values which are unimportant for the discussion here.

Moreover, if $r_{jj} = 0$ in the Givens rotation or if there is only one non-zero element in $\mathbf{r}^H$ in Householder reflection, the permutation matrix can be chosen to reorder the elements. Deeper treatment of unitary transformation can be found in [20, 25, 26].

2.2. Polynomial Matrix Eigenvalue Decomposition (PEVD)

The PEVD algorithm for a para-Hermitian polynomial matrix, $\mathbf{A}(z) = \mathbf{A}^P(z)$, where $\mathbf{A}^P(z) = \mathbf{A}^H(z^{-1})$, $\mathbf{A}(z) \in \mathbb{C}^{M\times M}$, proposed in [11], can be expressed as

$$\mathbf{A}(z) \approx \mathbf{U}(z)\mathbf{\Lambda}(z)\mathbf{U}^P(z), \tag{6}$$

where columns of $\mathbf{U}(z)$ are the eigenvectors with its corresponding eigenvalues on the diagonal matrix, $\mathbf{\Lambda}(z)$. Elements in $\mathbf{U}(z)$ and $\mathbf{\Lambda}(z)$ are in general, polynomials in z. The eigenvalues of $\mathbf{A}(z)$ are preserved if $\mathbf{U}(z)$ satisfies the para-unitary condition [27] given by

$$\mathbf{U}^P(z)\mathbf{U}(z) = \mathbf{U}(z)\mathbf{U}^P(z) = \mathbf{I}. \tag{7}$$

The Givens rotation and Householder reflection matrices described earlier also satisfy (7). They can be viewed as the degenerate case where the matrix elements are coefficients of z^0.

An elementary operator for a polynomial matrix, that changes the polynomial order, is the t-th delay matrix applied to the j-th row,

$$\mathbf{D}_j(z) = \begin{bmatrix} \mathbf{I}_{j-1} & 0 & 0 \\ 0 & z^{-t} & 0 \\ 0 & 0 & \mathbf{I}_{M-j} \end{bmatrix}. \tag{8}$$

For numerical diagonalisation in (6), $\mathbf{U}(z)$ can comprise a sequence of L para-unitary matrix factors, $\mathbf{U}_1(z), \mathbf{U}_2(z), \ldots, \mathbf{U}_L(z)$, built from elementary delay, rotation and/or reflection matrices.

2.3. Sign and Delay Ambiguity in Eigenvectors for PEVD

From [20], a conventional matrix with distinct eigenvalues has orthonormal eigenvectors, unique up to non-zero scaling factor, that lies in the columns of $\mathbf{U}$. Additionally, for polynomial matrices, the eigenvectors are also unique up to a sign and delay factor in z. This result will be exploited in Section 5. For example, if $M = 3$, then

$$\mathbf{\Lambda}(z) = \begin{bmatrix} \lambda_1(z) & 0 & 0 \\ 0 & \lambda_2(z) & 0 \\ 0 & 0 & \lambda_3(z) \end{bmatrix}, \tag{9}$$

$$\mathbf{U}(z) = \left[\pm z^{-p}\mathbf{u}_1(z),\ \pm z^{-q}\mathbf{u}_1(z),\ \pm z^{-r}\mathbf{u}_3(z)\right], \tag{10}$$

$$\mathbf{U}^P(z) = \left[\pm z^{p}\mathbf{u}_1(z^{-1}),\ \pm z^{q}\mathbf{u}_2(z^{-1}),\ \pm z^{r}\mathbf{u}_3(z^{-1})\right]^H, \tag{11}$$

where λ_m is the m-th eigenvalue with corresponding eigenvector, $\mathbf{u}_m^T = [u_{1m}, u_{2m}, u_{3m}]$ and its associated arbitrary delay, z^{-m}. The diagonal terms of $\mathbf{A}(z)$ in (6) are

$$a_{ii}(z) = \sum_{k=1}^{3} \lambda_k(z) u_{ik}(z) u^*_{ik}(z^{-1}) \tag{12}$$

and off-diagonal terms,

$$a_{ij}(z) = a^P_{ji}(z) = \sum_{k=1}^{3} \lambda_k(z) u_{ik}(z) u^*_{jk}(z^{-1}). \tag{13}$$

Observe that (12) and (13) hold for all arbitrary signs and delays because the factors cancel out. Each sign ambiguity can also be expressed as a complex scale factor which can be accounted for in either the eigenvector or z. Furthermore, if z is evaluated on the unit circle, the sign and delay ambiguity is equivalent to phase ambiguity, that is inherent in any power estimation problem using only second-order statistics. A deeper discussion is provided in [28].

3. SBR2 ALGORITHM

McWhirter et al. [11] proposed the SBR2 algorithm for factorising (6) using a series of para-unitary transformation. Consider the following 2-by-2 para-Hermitian polynomial matrix example,

$$\mathbf{A}(z) = \begin{bmatrix} 1 & -0.1z^{-1} + 0.5z^{1} \\ -0.1z^{1} + 0.5z^{-1} & 1 \end{bmatrix}. \tag{14}$$

For each iteration, SBR2 begins by searching for the largest off-diagonal element, $|\hat{g}|$, across all powers of z as depicted in Fig. 1(i) and comparing against a predefined threshold, δ. The element is indexed by j-th row, k-th column and z^t-plane. Shown in Fig. 1(ii), the dominant element is brought to the principal plane, z^0, using a delay matrix if it exceeds δ. A Givens rotation, computed based on the principal plane, is applied to the entire polynomial matrix as indicated in Fig. 1(iii). This zeros the elements containing $|\hat{g}|$ on the principal plane. Due to symmetry, the search is confined to half the off-diagonals and similarity transformation results in operations acting on dominant pairs rather than just elements.

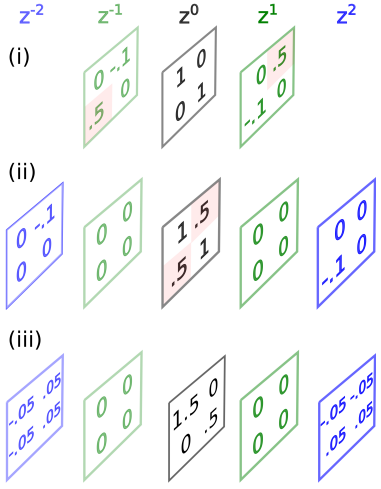

Fig. 1. Visualisation for the first iteration of SBR2 for (14) that demonstrates the (i) search, (ii) delay and (iii) rotate step (after [29]).

The overall para-unitary matrix which diagonalises the polynomial matrix approximately after L iterations is

$$\mathbf{U}(z) = \mathbf{U}_L(z)\mathbf{U}_{L-1}(z)\dots\mathbf{U}_1(z), \quad (15)$$

where $\mathbf{U}_l(z) = \mathbf{G}^l\mathbf{D}^l(z)$ is the product of rotation and delay matrices during the l-th iteration and satisfies (7).

Naturally, the number of delay operations may increase as the algorithm progresses with more iterations. Consequently, the order of the polynomial matrix may grow, as evident in Fig. 1, and becomes unnecessarily large. This will increase search and computation time as well as storage requirements.

For practical applications, $\tilde{\mathbf{U}}(z)$ and $\tilde{\mathbf{\Lambda}}(z)$, approximations of $\mathbf{U}(z)$ and $\mathbf{\Lambda}(z)$ in (6), are usually sufficient. This has led to the use of a reconstruction error measure, $\epsilon = \sum_{\forall z} \|\tilde{\mathbf{A}}(z) - \mathbf{A}(z)\|_F$ where $\|.\|_F$ is the Frobenius-norm (F-norm), to indicate the accuracy of the approximation. ϵ can also be expressed as a percentage of $\|\mathbf{A}(z)\|_F$. Several methods for trimming polynomial terms with small coefficients to keep the order compact have been proposed [9, 30, 31] but the energy-based technique [32] was chosen in SBR2. For a given ratio of squared F-norm or energy permitted to be lost, $\tilde{\mathbf{A}}(z)$ is approximated by the smallest $\tau_{\max}$ satisfying the inequality,

$$\text{trim}(\mathbf{A}(z), \mu) = \sum_{\tau=-\tau_{\max}}^{\tau_{\max}} \|\mathbf{A}(\tau)z^{-\tau}\|_F^2 \geq (1-\mu)\|\mathbf{A}(z)\|_F^2, \quad (16)$$

where μ is the trim parameter and $\mathbf{A}(\tau)$ are the matrix coefficients.

Because only one element pair is zeroed during each SBR2 iteration, all off-diagonal elements have to be iteratively processed to cycle through all elements. Consequently, more iterations and hence slower convergence is expected as the number of elements increases with the size of the matrix and polynomial order.

4. PROPOSED ENHANCEMENT TO SBR2

At each iteration, the Givens rotation increases the square of the trace norm on the principal plane by $2|\hat{g}|^2$ due to zeroing of the symmetric pair. However, if the matrix at the principal plane is reduced to tridiagonal form before rotation, the norm of the entire row and column, which includes the dominant pairs, becomes compactly stored in the super- and sub-diagonals. By applying the Givens rotations to the tridiagonal matrix, more of the F-norm of the off-diagonal entries on the principal plane can be transferred onto the diagonal at each iteration.

For the l-th iteration, the Householder matrix is the product of row-wise reflection matrices, $\mathbf{H}_j$, defined in (5), recursively computed from rows $1, 2, \dots, (M-2)$ giving

$$\mathbf{H}^l = \mathbf{H}_{M-2}\dots\mathbf{H}_2\mathbf{H}_1, \quad (17)$$

which reduces the principal plane to tridiagonal form. Each Givens rotation based on (3) can be computed to zero off-diagonal elements i.e. $k = j+1, j = 1, \dots, M-1$ on the principal plane so that

$$\mathbf{G}^l = \mathbf{G}_{(M-1,M)}\dots\mathbf{G}_{(2,3)}\mathbf{G}_{(1,2)}. \quad (18)$$

The unitary term for the l-th iteration fulfilling (7) is

$$\mathbf{U}_l(z) = \mathbf{G}^l\mathbf{H}^l\mathbf{D}^l(z), \quad (19)$$

a product of rotation, reflection and delay matrices and is applied to the polynomial matrix across all z.

The proof of convergence follows directly from [11] since the proposed algorithm transfers at least as much energy as SBR2 under the worst case where there is only 1 off-diagonal pair on the principal plane to be zeroed. With more than 1 off-diagonal pair, more energy will be transferred. Consequently, there exists a minimal upper bound for the number of iterations, L, where the algorithm diagonalises $\mathbf{A}(z)$ approximately such that the modulus square of the off-diagonals are arbitrarily small.

Although the trace norm of the principal plane increases monotonically, the same transformation applied to other z-planes may increase the magnitude of off-diagonal elements. With this, SBR2 with Householder reduction is summarised in Algorithm 1.

Algorithm 1 SBR2 with Householder Reduction.

Inputs: $\mathbf{A}(z) \in \mathbb{C}^{M\times M}$, δ, maxIter, μ.
Initialise: $l \leftarrow 0$, g$\leftarrow 1+\delta$, $\tilde{\mathbf{\Lambda}}(z) = \mathbf{A}(z)$,$\tilde{\mathbf{U}}(z) = \mathbf{I}$.
  while ($l <$maxIter **and** g$> \delta$) **do**
    g $\leftarrow \max|r_{jk}(z^t)|, k > j, \forall t.$
    if (g$> \delta$) **then**
      $l \leftarrow l+1$.
      $\tilde{\mathbf{\Lambda}}(z) \leftarrow \mathbf{D}_j(z)\tilde{\mathbf{\Lambda}}(z)\mathbf{D}_j^P(z)$,
      $\tilde{\mathbf{U}}(z) \leftarrow \mathbf{D}_j(z)\tilde{\mathbf{U}}(z)$ // delay according to (8).
      $\tilde{\mathbf{\Lambda}}(z) \leftarrow \mathbf{H}\tilde{\mathbf{\Lambda}}(z)\mathbf{H}^H$,
      $\tilde{\mathbf{U}}(z) \leftarrow \mathbf{H}\tilde{\mathbf{U}}(z)$ // reflect according to (17).
      $\tilde{\mathbf{\Lambda}}(z) \leftarrow \mathbf{G}(\theta,\phi)\tilde{\mathbf{\Lambda}}(z)\mathbf{G}^H(\theta,\phi)$,
      $\tilde{\mathbf{U}}(z) \leftarrow \mathbf{G}(\theta,\phi)\tilde{\mathbf{U}}(z)$ // rotate according to (18).
      $\tilde{\mathbf{\Lambda}}(z) \leftarrow$trim$(\tilde{\mathbf{\Lambda}}(z), \mu)$,
      $\tilde{\mathbf{U}}(z) \leftarrow$trim$(\tilde{\mathbf{U}}(z), \mu)$ // trim according to (16).
    end if
  end while
  return $\tilde{\mathbf{U}}(z)$, $\tilde{\mathbf{\Lambda}}(z)$.

5. SIMULATIONS AND RESULTS

The results presented here will follow the convention established in [11] in which (6) was defined as $\mathbf{A}(z) \approx \mathbf{U}^P(z)\mathbf{\Lambda}(z)\mathbf{U}(z)$. The eigenvectors lie in the rows of $\mathbf{U}(z)$ and delays in (8) were applied to index k instead of j. The first example is

$$\mathbf{A}_1(z) = \begin{bmatrix} 1 & -0.4iz & 0 \\ 0.4iz^{-1} & 1 & 0.5z^{-2} \\ 0 & 0.5z^2 & 1 \end{bmatrix}. \quad (20)$$

With $\delta = 0$, the convergence for this example using both methods is shown in Fig. 2. The convergence for the first two iterations were identical because the principal plane of the polynomial matrix had only one off-diagonal pair. From the third iteration, our method was able to transfer more energy to the diagonal as evident in the figure. Despite so, our method took one more iteration to produce the same polynomial factors as SBR2. This is because (20) is sparse with few non-zero polynomial elements and like conventional matrix, numerical diagonalisation using the Givens method may converge at least as fast as the Householder method. The second example is

$$\mathbf{A}_2(z) = \begin{bmatrix} 1 & 0.8z^2 - 0.4z & 0.7z \\ 0.8z^{-2} - 0.4z^{-1} & 1 & 0.5z^{-2} \\ 0.7z^{-1} & 0.5z^2 & -1 \end{bmatrix}, \quad (21)$$

for which our method took 17 iterations compared to 23 iterations with the original SBR2 using $\delta = 10^{-3}$ and $\mu = 10^{-4}$. As shown in Fig. 2, our method was able to transfer more energy at each iteration and drive down the dominant element faster. Furthermore, our method resulted in a lower reconstruction error, ϵ, from 0.0672 to

0.0478. This is because trimming at every iteration keeps the matrix factors compact at the expense of accuracy that accumulates with more iterations. $\mathbf{\Lambda}(z)$ computed using both algorithms were identical. For $\mathbf{U}(z)$ shown in Fig. 3, the eigenvectors produced by both methods are comparable up to a delay ambiguity as discussed in Section 2.3. The compensated delay was also computed and shown as black plus signs.

Without trimming, our method converges in 30 iterations compared to 37 iterations using the original SBR2. The order for the polynomial factors were comparable with slight savings at 216 versus 224 for $\mathbf{\Lambda}(z)$ and 106 versus 110 for $\mathbf{U}(z)$. Both algorithms gave $\epsilon = 0$ (up to machine precision).

The algorithms were compared with a decorrelation example involving 3 sensors and 2 sources using $\delta = \sqrt{\frac{N_1}{3\times 10^4}}$ where N_1 is the square of the trace norm at the principal plane and $\mu = 10^{-4}$. Each channel was a 5th order Finite Impulse Response (FIR) filter with coefficients drawn from a uniform distribution, $U\,[-1, 1]$. The source signals were independent, identically distributed sequences with each sample being assigned ± 1 with equal probability. Each sensor output was also corrupted by Gaussian noise with $\sigma = 1.8$. 1000 samples were used in the computation of $\mathbf{A}(\tau)$, assumed to be negligibly small outside the correlation window, W. With $W = 10$, an approximated z-transform, $\mathbf{A}(z) = \sum_{\tau=-W}^{W} \mathbf{A}(\tau) z^{-\tau}$, was used in the computation.

For the realisation shown in Fig. 4, our method took 64 iterations while SBR2 took 90 iterations and the resulting ϵ values were respectively, 1.4515 and 1.7722. Lower ϵ was likely due to fewer iterations and accumulation of inaccuracies arising from trimming. The order of $\mathbf{\Lambda}(z)$ was 27 for both methods while the order of $\mathbf{U}(z)$ was 33 and 35.

Without trimming, $\mu = 0$, our method took 80 iterations while SBR2 took 114 iterations and the resulting order of $\mathbf{\Lambda}(z)$ was 633 and 1148 and $\mathbf{U}(z)$ was 624 and 1138. We note, however, that the overall computational complexity depends on many factors including the number of iterations, search strategy, zeroing and trimming method, polynomial order at each iteration and the implementation.

To further compare their performance, a Monte-Carlo simulation of 1000 decorrelation trials with trimming was conducted. The relative iteration difference between both methods was computed based on $(L_{\text{Proposed}} - L_{\text{SBR2}}) \times 100/L_{\text{SBR2}}\%$ and relative error difference was based on $(\epsilon_{\text{Proposed}} - \epsilon_{\text{SBR2}}) \times 100/\|\mathbf{A}(z)\|_F\%$. From the histogram shown in Fig. 5, the proposed method gave an average of 12.35% reduction in the number of iterations over SBR2 with an average reduction of 0.1% in relative ϵ. Moreover, because the distribution was asymmetric, our proposed method enjoyed a reduction in the number of iterations for 82.4% of the trials. The results obtained with both methods were consistent to within 1% ϵ.

6. CONCLUSION

We have briefly reviewed SBR2, the classical algorithm used for PEVD, and proposed an enhancement using Householder reflection to reduce the principal plane to tridiagonal form prior to rotation. This enhanced method is able to transfer more energy to the principal diagonal at every iteration, enabling quicker convergence with an average of 12.35% reduction in iteration counts, and generally polynomial matrix factors with lower order and 0.1% improvement in reconstruction error. In the comparative simulation, our method enjoyed convergence speed improvements for 82.4% of the trials. Furthermore, our proposed method is related to SMD and is an alternate way to diagonalise the Hermitian principal plane at every iteration.

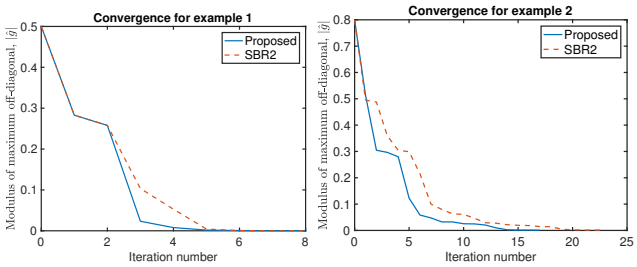

Fig. 2. Convergence of algorithms for (20) and (21).

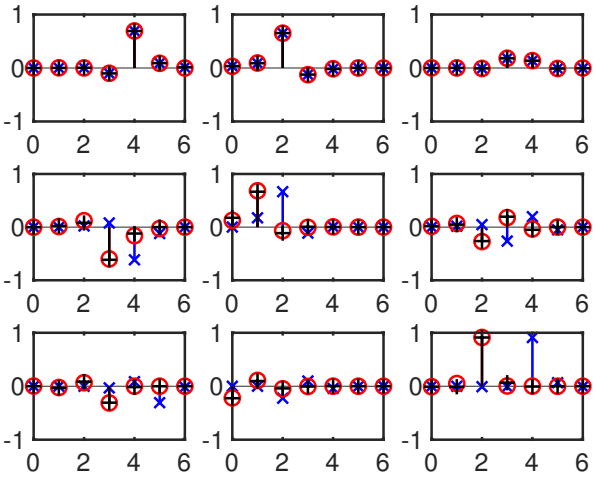

Fig. 3. $\mathbf{U}(z)$ for (21) where black pluses and blue crosses indicate results computed using our proposed method with and without delay compensation and red circles using SBR2.

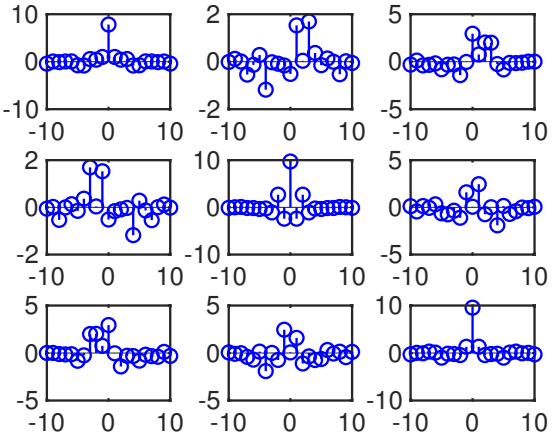

Fig. 4. Covariance matrix, $\mathbf{A}(z)$ for the decorrelation example.

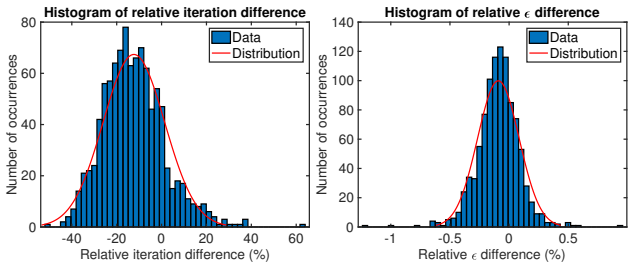

Fig. 5. Monte-Carlo simulation results for decorrelation trials.

7. REFERENCES

[1] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding*. Upper Saddle River, NJ, USA: Prentice Hall, 1995.

[2] P. C. Hansen and S. H. Jensen, "Subspace-based noise reduction for speech signals via diagonal and triangular matrix decompositions: Survey and analysis," *EURASIP J. on Advances in Signal Processing*, vol. 2007, no. 1, Jan. 2007.

[3] P. Stoica and R. Moses, *Spectral Analysis of Signals*. New Jersey, USA: Prentice Hall, 2005.

[4] G. Chabriel, M. Kleinsteuber, E. Moreau, H. Shen, P. Tichavsky, and A. Yeredor, "Joint matrices decompositions and blind source separation: A survey of methods, identification, and applications," *IEEE Signal Process. Mag.*, vol. 31, no. 3, pp. 34–43, May 2014.

[5] S. Haykin, *Adaptive Filter Theory*, 4th ed. Prentice Hall, 2002.

[6] T.-J. Shan and T. Kailath, "Adaptive beamforming for coherent signals and interference," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 3, pp. 527–536, Jun. 1985.

[7] S. Redif, S. Weiss, and J. G. McWhirter, "An approximate polynomial matrix eigenvalue decomposition algorithm for para-Hermitian matrices," in *Proc. Intl. Symp. on Signal Processing and Information Technology (ISSPIT)*, Bilbao, Spain, 2011, pp. 421–425.

[8] J. Corr, K. Thompson, S. Weiss, J. G. McWhirter, S. Redif, and I. K. Proudler, "Multiple shift maximum element sequential matrix diagonalisation for para-Hermitian matrices," in *Proc. IEEE/SP Workshop on Statistical Signal Processing*, Australia, 2014, pp. 844–848.

[9] Z. Wang, J. G. McWhirter, and S. Weiss, "Multichannel spectral factorization algorithm using polynomial matrix eigenvalue decomposition," in *Proc. Asilomar Conf. on Signals, Systems & Computers*, 2015, pp. 1714–1718.

[10] S. Redif, S. Weiss, and J. G. McWhirter, "Sequential matrix diagonalisation algorithms for polynomial EVD of para-Hermitian matrices," *IEEE Transactions Signal Processing*, vol. 63, no. 1, pp. 81–89, Jan. 2015.

[11] J. G. McWhirter, P. D. Baxter, T. Cooper, S. Redif, and J. Foster, "An EVD algorithm for para-Hermitian polynomial matrices," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 2158–2169, May 2007.

[12] M. A. Alrmah, S. Weiss, and S. Lambotharan, "An extension of the MUSIC algorithm to broadband scenarios using a polynomial eigenvalue decomposition," in *Proc. European Signal Processing Conf. (EUSIPCO)*, 2011, pp. 629–633.

[13] W. Coventry, C. Clemente, and J. Soraghan, "Enhancing polynomial MUSIC algorithm for coherent broadband sources through spatial smoothing," in *Proc. European Signal Processing Conf. (EUSIPCO)*, 2017, pp. 2448–2452.

[14] S. Weiss, S. Bendoukha, A. Alzin, F. K. Coutts, I. K. Proudler, and J. Chambers, "MVDR broadband beamforming using polynomial matrix techniques," in *Proc. European Signal Processing Conf. (EUSIPCO)*, 2015, pp. 839–843.

[15] J. Corr, J. Pestana, S. Weiss, I. K. Proudler, S. Redif, and M. Moonen, "Investigation of a polynomial matrix generalised EVD for multi-channel Wiener filtering," in *Proc. Asilomar Conf. on Signals, Systems & Computers*, 2016, pp. 1354–1358.

[16] S. Redif, S. Weiss, and J. G. McWhirter, "Relevance of polynomial matrix decompositions to broadband blind signal separation," *Signal Processing*, vol. 134, pp. 76–86, May 2017.

[17] A. W. H. Khong, P. A. Naylor, and J. Benesty, "A low delay and fast converging improved proportionate algorithm for sparse system identification," *EURASIP J. on Audio, Speech, and Music Processing*, vol. 2007, no. 1, p. 084376, Apr. 2007.

[18] S. Weiss, N. J. Goddard, S. Somasundaram, I. K. Proudler, and P. A. Naylor, "Identification of broadband source-array responses from sensor second order statistics," in *Sensor Signal Process. for Defence Conf. (SSPD)*, 2017.

[19] S. Weiss, S. Redif, T. Cooper, C. Liu, P. D. Baxter, and J. G. McWhirter, "Paraunitary oversampled filter bank design for channel coding," *EURASIP Journal on Advances in Signal Processing*, vol. 2006, no. 1, Mar. 2006.

[20] G. Strang, *Linear Algebra and Its Application*, 2nd ed. Academic Press, 1980.

[21] A. Tkacenko, "Approximate eigenvalue decomposition of para-Hermitian systems through successive FIR paraunitary transformations," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.

[22] R. S. Martin and J. H. Wilkinson, "Householder's tridiagonalisation of a symmetric matrix," *Numer. Math.*, vol. 11, pp. 181–195, 1968.

[23] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, 1st ed. Oxford, UK: Clarendon Press, 1965.

[24] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, 3rd ed. Cambridge University Press, 2007.

[25] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD, USA: John Hopkins University, 1996.

[26] A. H. Sayed, "Unitary transformations," in *Adaptive Filters*. Wiley-IEEE Press, 2008.

[27] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Upper Saddle River, NJ, USA: Prentice Hall, 1993.

[28] S. Weiss, J. Pestana, and I. K. Proudler, "On the existence and uniqueness of the eigenvalue decomposition of a para-Hermitian matrix," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2659–2672, May 2018.

[29] Z. Wang, J. G. McWhirter, J. Corr, and S. Weiss, "Multiple shift second order sequential best rotation algorithm for polynomial matrix EVD," in *Proc. European Signal Processing Conf. (EUSIPCO)*, 2015, pp. 844–848.

[30] J. Corr, K. Thompson, S. Weiss, I. Proudler, and J. G. McWhirter, "Shortening of paraunitary matrices obtained by polynomial eigenvalue decomposition algorithms," in *Proc. Sensor Signal Processing for Defence (SSPD)*, 2015.

[31] Z. Wang, J. G. McWhirter, J. Corr, and S. Weiss, "Order-controlled multiple shift SBR2 algorithm for para-Hermitian polynomial matrices," in *IEEE Sensor Array and Multichannel Signal Process. Workshop (SAM)*, 2016, pp. 1–5.

[32] J. Foster, J. G. McWhirter, and J. Chambers, "Limiting the order of polynomial matrices within the SBR2 algorithm," in *IMA Int. Conf. on Math. in Signal Process.*, 2006.