Cyclic-by-Row Approximation of Iterative Polynomial EVD Algorithms

Jamie Corr*, Keith Thompson*, Stephan Weiss*, John G. McWhirter†, and Ian K. Proudler‡
*Department of Electronic & Electrical Engineering, University of Strathclyde, Glasgow, Scotland
†School of Engineering, Cardiff University, Cardiff, Wales, UK
‡School of Electrical, Electronics and Systems Engineering, Loughborough University, Loughborough, UK
{jamie.corr,keith.thompson,stephan.weiss}@strath.ac.uk

***Abstract*—A recent class of sequential matrix diagonalisation (SMD) algorithms has been demonstrated to provide a fast converging solution to iteratively approximating the polynomial eigenvalue decomposition of a parahermitian matrix. However, the calculation of an EVD, and the application of a full unitary matrix to every time lag of the parahermitian matrix in the SMD algorithm results in a high numerical cost. In this paper, we replace the EVD with a limited number of Givens rotations forming a cyclic-by-row Jacobi sweep. Simulations indicate that a considerable reduction in computational complexity compared to SMD can be achieved with a negligible sacrifice in diagonalisation performance, such that the benefits in applying the SMD are maintained.**

I. Introduction

For many signal processing and in particular array problems, the eigenvalue decomposition of that the data's covariance matrix is an essential tool, as it permits e.g. subspace decompositions or the construction of optimum linear processing of MIMO system. For the broadband case with a multichannel data vector $\mathbf{x}[n]$, the consideration of narrowband phase shifts is insufficient, and time delays have to be properly addressed. The space-time covariance matrix $\mathbf{R}[\tau] = \mathcal{E}\{\mathbf{x}[n]\mathbf{x}^{\mathrm{H}}[n-\tau]\}$, where $\mathcal{E}\{\cdot\}$ is the expectation operator and $\{\cdot\}^{\mathrm{H}}$ denotes the Hermitian transpose, arises. Its z-transform is the cross-spectral density (CSD) matrix $\boldsymbol{R}(z) = \sum_\tau \mathbf{R}[\tau] z^{-\tau}$, which is a polynomial matrix that cannot be decomposed by the narrowband EVD.

A polynomial EVD (PEVD) was defined in the context of subband coding [2], which is applied to the parahermitian $\boldsymbol{R}(z) = \tilde{\boldsymbol{R}}(z)$ such that

$$\boldsymbol{R}(z) \approx \boldsymbol{H}(z)\boldsymbol{D}(z)\tilde{\boldsymbol{H}}(z) \quad , \tag{1}$$

with the parahermitian operator $\tilde{\boldsymbol{R}}(z) = \boldsymbol{R}^{\mathrm{H}}(z^{-1})$. The factors on the l.h.s. of (1) include the paraunitary matrix $\boldsymbol{H}(z)$ such that $\boldsymbol{H}(z)\tilde{\boldsymbol{H}}(z) = \tilde{\boldsymbol{H}}(z)\boldsymbol{H}(z) = \mathbf{I}$, and the parahermitian $\boldsymbol{D}(z)$, which is diagonal

$$\boldsymbol{D}(z) = \mathrm{diag}\{D_0(z)\ D_1(z)\ \ldots\ D_{M-1}(z)\} \tag{2}$$

and spectrally majorised, such that the power spectral densities $D_m(e^{j\Omega}) = D_m(z)|_{z=e^{j\Omega}}$ satisfy $D_{m+1}(e^{j\Omega}) \geq D_m(e^{j\Omega})\ \forall\ \Omega\ , m = 0 \ldots (M-1)$. In [2], (1) is stated with equality for the optimum subband coder, even though the existence of an exact PEVD with FIR paraunitary matrices is not guaranteed [3]. However, the close approximation in (1) is supported in [4].

To calculate the PEVD, a number of iterative algorithms have been developed. The second order sequential best rotation (SBR2) algorithm [3] is a generalisation of Jacobi whereby in every step the maximum off-diagonal element of $\boldsymbol{R}(z)$ is eliminated by an elementary paraunitary operation. In this operation, a shift operation brings this largest off-diagonal element into the zero lag matrix, where its energy is transfered onto the diagonal by a Givens rotation. An alternative approach in [5] uses a fixed order with a lower computational complexity than SBR2, but unlike SBR2 is not guaranteed to converge. A modified SBR2 version, optimised for the coding gain problem, was proposed in [6], which identifies and eliminates the maximum normalised off-diagonal element at every iteration step.

A new family of algorithms, termed sequential matrix diagonalisation (SMD), has been introduced in [7]–[9]. Different from SBR2, these algorithm do not only transfer the maximum element, but eliminate all off-diagonal elements in the zero lag plane by means of an EVD. SMD algorithms have been demonstrated to diagonalise a parahermitian matrix in fewer iterations, with higher precision, and with a lower order paraunitary matrix than SBR2, which makes SMD very attractive for applications that e.g. require accurate broadband subspace decomposition [10]. A number of SMD algorithms will be review later in this paper; while their application is attractive, the computational complexity of computing the actual decomposition is significantly higher than SBR2.

Therefore, this paper proposes a low cost approximation of SMD algorithms. Based on a review of iterative PEVD algorithms in Sec. II, Sec. III outlines the proposed approach, which replaced the EVD at every iteration by a single Jacobi sweep implemented by a fixed number of Given rotations in a cyclic-by-row arrangement [11], [12]. Simulation results are presented in Sec. IV to hightlight the performance of cyclic-by-row approximations of the SMD benchmarked against state-of-the-art algorithms. Finally , conclusions are drawn in Sec. V.

II. Iterative PEVD Algorithms

A number of iterative PEVD algorithms have been discussed in the literature, of which SBR2 [3], [6] is the most prominent

due to its simplicity and proven convergence. Here we will not repeat the derivation of SBR2, but instead focus on the recently derived class of sequential matrix diagonalisation (SMD) algorithms [7], [9] and their characteristics.

A. General Iteration Approach

The SMD and SBR2 algorithm classes both perform a sequence of paraunitary operations $\boldsymbol{H}^{(i)}(z)$, with iteration index i, by which a parahermitian matrix $\boldsymbol{R}(z)$ is eventually approximately diagonalised, such that the off-diagonal energy of

$$\boldsymbol{S}^{(I)}(z) = \tilde{\boldsymbol{H}}(z)\boldsymbol{R}(z)\boldsymbol{H}(z) \tag{3}$$

$$\boldsymbol{H}(z) = \prod_{i=0}^{I} \boldsymbol{H}^{(i)}(z) = \prod_{i=0}^{I} \boldsymbol{\Lambda}^{(i)}(z)\mathbf{Q}^{(i)} \tag{4}$$

is minimised. Each iteration i consists of a delay operation by a diagonal matrix $\boldsymbol{\Lambda}^{(i)}(z)$, which creates an intermediate parahermitian matrix

$$\boldsymbol{S}^{(i)\prime}(z) = \tilde{\boldsymbol{\Lambda}}^{(i)}(z)\boldsymbol{S}^{(i-1)}(z)\boldsymbol{\Lambda}^{(i)}(z)\,, \quad i = 1 \ldots I\,, \tag{5}$$

and a unitary operation $\mathbf{Q}^{(i)}$ such that

$$\boldsymbol{S}^{(i)}(z) = \mathbf{Q}^{(i)\mathrm{H}}\boldsymbol{S}^{(i)\prime}(z)\mathbf{Q}^{(i)} \tag{6}$$

completes the ith iteration.

The delay operation $\boldsymbol{\Lambda}^{(i)}(z)$ is based on an elementary delay

$$\boldsymbol{\Lambda}^{(i,j)}(z) = \mathrm{diag}\{\underbrace{1\ \ldots\ 1}_{k^{(i,j)}-1}\ z^{-\tau^{(i,j)}}\ \underbrace{1\ \ldots\ 1}_{M-k^{(i,j)}}\} \quad , \tag{7}$$

which delays the $k^{(i,j)}$th column by $\tau^{(i,j)}$ lags. In general, at the ith iteration, a maximum of $M-1$ columns could be shifted, resulting in

$$\boldsymbol{\Lambda}^{(i)}(z) = \prod_{j=1}^{J} \boldsymbol{\Lambda}^{(i,j)}(z) \tag{8}$$

with $J \leq M-1$. For the standard versions of SBR2 [3], [6] and SMD as introduced in [9], only a single elementary delay $J = 1$ is used.

Per iteration, the various iterative PEVD algorithms differ in the selection of the parameters $k^{(i,j)}$, $\tau^{(i,j)}$, and the construction of the unitary matrix $\mathbf{Q}^{(i)}$. This parameter selection will be detailed below. The iteration stops when a threshold for a norm on the off-diagonal elements is passed, whereby the norm selection of specific to the particular iterative PEVD method.

B. Second Order Sequential Best Rotation Algorithm

In the SBR2 algorithm, at every iteration the maximum off-diagonal element is identified, transferred to the lag-zero slice by a single elementary delay matrix, $J = 1$, and then eliminated by a Givens rotation.

Based on a modified column vector $\hat{\mathbf{s}}_k^{(i)}[\tau] \in \mathbb{C}^{M-1}$ containing all elements in the $k^{(i)}$th column of $\mathbf{S}^{(i)}[\tau]$ except for the diagonal element, the optimum parameter set for (7) is determined by

$$\{k^{(i,1)}, \tau^{(i,1)}\} = \arg\max_{k,\tau} \|\hat{\mathbf{s}}_k^{(i-1)}[\tau]\|_\infty \quad , \tag{9}$$

such that in the ith iteration the optimum off-diagonal element will now lie in the lag zero matrix $\mathbf{S}^{(i)\prime}[0]$.

The maximum element is then transfered onto the diagonal of $\mathbf{S}^{(i)\prime}[0]$ by means of a Givens rotation, such that

$$\mathbf{Q}^{(i)} = \begin{bmatrix} \mathbf{I}_1 & & & \\ & \cos\varphi_i & \ldots & e^{j\vartheta^{(i)}}\sin\varphi^{(i)} \\ & \vdots & \mathbf{I}_2 & \vdots \\ & -e^{-j\vartheta^{(i)}}\sin\varphi^{(i)} & \ldots & \cos\varphi^{(i)} \\ & & & \mathbf{I}_3 \end{bmatrix} \tag{10}$$

where the rotation angles $\varphi^{(i)}$ and $\vartheta^{(i)}$ are determined by the maximum element identified by the search in (9). The identity matrices $\mathbf{I}_j$, $j = 1, 2, 3$, in (10) have dimensions $(\min\{m^{(i)}, k^{(i)}\} - 1)$, $(|m^{(i)} - k^{(i)}| - 1)$ and $(M - \max\{m^{(i)}, k^{(i)}\} + 1)$, respectively. The resulting unitary matrix $\mathbf{Q}^{(i)}$ has to be left- and right-multiplied to every lag matrix $\mathbf{S}^{(i)\prime}[\tau]$ according to (6); however, no full matrix multiplication is required, due to the sparse structure of $\mathbf{Q}^{(i)}$ only two rows and columns in $\boldsymbol{S}^{(i)\prime}(z)$ will be affected.

The convergence of SBR2 has been proven in [3] by showing that the paraunitary operations do not alter the total energy in $\boldsymbol{S}^{(i)\prime}(z)$, while in every step the off-diagonal energy is further minimised. The algorithm stops after I iterations, once the maximum off-diagonal element

$$\max_{k,\tau} \|\hat{\mathbf{s}}_k^{(I)}[\tau]\|_\infty < \rho \tag{11}$$

falls below a predetermined threshold ρ.

C. Sequential Matrix Diagonalisation Algorithm

Different from SBR2, in the ith iteration the SMD approach will not just eliminate the largest off-diagonal element but diagonalises $\mathbf{S}^{(i)}[0]$. Based on the initialisation

$$\mathbf{S}^{(0)}[0] = \mathbf{Q}^{(0)\mathrm{H}}\mathbf{R}[0]\mathbf{Q}^{(0)} \quad , \tag{12}$$

with $\mathbf{Q}^{(0)}$ the modal matrix obtained from the EVD of $\mathbf{R}[0]$, every subsequent iteration brings on row and column to $\mathbf{S}^{(i)\prime}[0]$, whose energy is then transferred onto the diagonal by an EVD.

To maximise the diagonalisation, the SMD parameter selection in the ith iteration is

$$\{k^{(i,1)}, \tau^{(i,1)}\} = \arg\max_{k,\tau} \|\hat{\mathbf{s}}_k^{(i-1)}[\tau]\|_2 \quad , \tag{13}$$

which differs from (9) in the use of the l_2 instead of the l_∞ norm. To achieve complete diagonalisation of $\mathbf{S}^{(i)}[0]$, $\mathbf{Q}^{(i)}$ is the modal matrix obtained by the EVD of $\mathbf{S}^{(i)\prime}[0]$.

The convergence of SMD is proven in [9], with a stopping criterion similar to (11) but based on the l_2 norm according to (13). SMD has been shown to provide a much faster diagonalisation than SBR2 with a lower number of iterations,

because more energy is transferred from off-diagonal to on-diagonal elements. However, an EVD has to be calculated at every iteration, and the modal matrix $\mathbf{Q}^{(i)}$ has no longer the sparse structure of the Givens rotation in (10), requiring a full matrix multiplication at every lag τ.

While the EVD and full matrix multiplications makes the SMD algorithm much more costly to calculate compared to SBR2, two benefits can be noted:

- SMD can reach levels of diagonalisation that SBR2 cannot achieve;
- SMD generally leads to lower-order paraunitary matrices $\boldsymbol{H}(z)$, which have a lower cost compared to those obtainable by SBR2.

Therefore, for applications such as those requiring broadband signal subspace decompositions, SMD permits better performance with lower order paraunitary filter banks compared to SBR2.

D. Multiple Shift Maximum Element SMD Algorithm

To simplify the search of the SMD algorithm and avoid the calculation of modified column norms at every iteration, a maximum element SMD (ME-SMD) algorithm has been proposed in [9]. Replacing the l_2 norm in (13) by the l_∞ norm, the parameter search and delay operation within a single iteration is identical to SBR2. However, the zero-lag slice $\mathbf{S}^{(i)}[0]$ is still diagonalised at every step, such that from iteration to iteration step, ME-SMD differs from SBR2. The ME-SMD algorithm has a very similar behaviour to the standard SMD algorithm, in terms of convergence, but also in terms of complexity, which is dominated by the EVD calculation and modal matrix application, rather than the parameter search.

Realising that the main advantage of SMD over SBR2 comes from the enhanced reduction in off-diaginal energy at every iteration, a multiple shift ME-SMD algorithm has been proposed [7], which shifts not one but several columns with elements onto the lag zero slice. As a column operation will partially undo previous row operations and vice versa, the sequence of these shifts is important. This sequence is guided by the definition of search spaces $\mathcal{S}^{(i,j)}$ to which the parameter search is limited during the jth step of the ith iteration [7], [8], with $\mathcal{S}^{(i,j)}$ being a function of both $\mathcal{S}^{(i,j-1)}$ and $k^{(i,j-1)}q$. Based on $\mathcal{S}^{(i,j)}$, the identified parameter set will be

$$\{k^{(i,j)}, \tau^{(i,j)}\} = \arg\max_{k \in \mathcal{S}^{(i,j)}, \tau} \|\hat{\mathbf{s}}_k^{(i-1)}[\tau]\|_2 \quad , \qquad (14)$$

Different search spaces have been defined to ensure that a total of $J = M - 1$ maximal can be brought onto the zero lag matrix [7], [8]. All have in common a considerably enhanced diagonalisation over SMD as discussed in Sec. II-C.

III. Cyclic-by-Row Iterative PEVD

Sec. II-C identified the computational cost to compute an approximate PEVD via SMD algorithms as a potential obstacle. In experiments using Matlab's profiler, the calculation of an EVD per iteration step was singled out as the major contributor to this high cost. Therefore, this section proposes an inexpensive numerical approximation of the EVD by a Jacobi sweep consisting of a limited number of Givens rotations in a cyclic-by-row approach [12]. Below, Sec. III-A motivates the approach and Sec. III-B outlines the general procedure, which is then applied to a number of SMD algorithms in Sec. III-C.

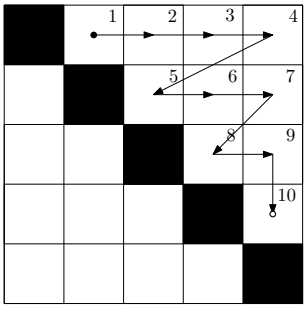

Fig. 1. Cyclic-by-row execution of Givens rotations implementing one Jabobi sweep, exemplified for a 5×5 matrix with start $\bullet$ and end point $\circ$.

A. EVD Approximation

A number of variations exist to implement the EVD of a Hermitian matrix [11], [12]. An iterative approach to an approximate EVD is the classical Jacobi algorithm [12], which consists of a sequence of Givens rotation targetting the maximum off-diagonal element at each iteration step — note that the SBR2 algorithm is a generalisation of this technique to the parahermitian case [3]. A simpler alternative which does not require a maximum search at every step is the cyclic-by-row algorithm [11], [12], which is outlined below

The cyclic-by-row approach uses a sequence of so-called Jacobi sweeps until off-diagonal elements are surpressed below a given threshold or a predefined number of sweeps has been executed. A Jacobi sweep for an $M \times M$ matrix consists of a fixed number of $(M^2 - M)/2$ Givens rotations, which are applied in a cyclic row approach as highlighted in Fig. 1. Each Givens rotation as defined in (10) will transfer the energy of an off-diagonal element onto the diagonal while undoing some of the work of previous Givens rotations. However, over the course of one Jacobi sweep, the off-diagonal energy is reduced. The EVD approximation becomes more accurate the more Jacobi sweeps are performed.

B. Cyclic-by-Row Algorithm

Iterative approximate PEVD algorithms such as SBR2 and SMD minimise off-diagonal energy until a predefined threshold ρ is reached, as described by (11). Therefore, within one iteration step of SMD, a full EVD with a suppression of off-diagonal energy to numerically zero appears to be an overkill, and a lower precision with a limited number of Jacobi sweeps will very likely suffice to achieve the task of reducing off-diagonal energy below the value ρ.

Experimentation has shown that for the combination with SMD algorithms detailed below, a single cyclic-by-row Jacobi sweep proved sufficient and provided the best cost-performance trade-off, as we will detail in Sec. IV. With this approach, the unitary Jacobi rotation matrix $\mathbf{Q}^{(i)}$ from SBR2

in (10) now becomes the product of $N = (M^2 - M)/2$ Givens rotations,

$$\mathbf{Q}^{(i)} = \prod_{n=1}^{N} \mathbf{Q}^{(i,n)} \quad , \tag{15}$$

where $\mathbf{Q}^{(i,n)}$ is the nth Givens rotation used in the ith iteration of an iterative PEVD algorithm using the single sweep cyclic-by-row approach.

C. Cyclic-by-Row SMD Algorithms

The cyclic-by-row single Jacobi sweep approximation of the EVD can be embedded in all algorithms of the SMD family. It may be argued that the term sequential matrix *diagonalisation* is no longer appropriate, as the approximate EVD also results in only an approximate diagonalisation, and algorithms will therefore share some properties of SBR2, where only part of the off-diagonal energy of the lag zero matrix is transfered onto the diagonal. However, we assume that the approximation is within the bound ρ for off-diagonal energy, and that therefore the term *diagonalisation* is justified within the SMD family's limited, pre-defined accuracy of decomposition.

All SMD algorithms perform an initial diagonalisation by an EVD according to (12), which in the cyclic-by-row version is approximated by a single Jacobi sweep. The EVD in subsequent iterations is also replaced by a single Jacobi sweep, and the unitary matrix (15) as applied in (6) can be implemented as a sequence of Givens rotations rather than a full matrix multiplication. The specific SMD family versions therefore consistently apply the single Jacobi sweep approach, and only differ in the way columns and rows are identified for transfer to the lag zero matrix using (5) at the ith iteration:

- SMD [9]: in its original form, the sequential matrix diagonalisation algorithm transfers the column with the largest off-diagonal column norm onto the lag zero matrix;
- ME-SMD [9]: with a simplified search compared to SMD, the column containing the maximum off-diagonal element is transfered to the lag zero matrix;
- MSME-SMD [7]: this multiple-shift version transfers $(M-1)$ colmns identified by their maximum elements;
- C-MSME-SMD [8]: a causally-constrained multiple-shift version, with a more restricted search space for maxima as compared to MSME-SMD.

The cyclic-by-row approximations of these algorithms will be compared to their standard versions as well as to SBR2 in the next section.

IV. Results

A. Simulation Set-Up and Performance Metrics

To assess the proposed iterative PEVD algorithms, we consider the diagonalisation performance over an ensemble of 10^3 random 5×5 parahermitian matrices $\boldsymbol{R}(z)$ of order 11. Each ensemble probe is generated from a random polynomial matrix $\boldsymbol{A}(z) \in \mathbb{C}^{5\times 5}$ of order 6 with independent and identically distributed zero mean unit variance complex Gaussian entries, where $\boldsymbol{R}(z) = \boldsymbol{A}(z)\tilde{\boldsymbol{A}}(z)$.

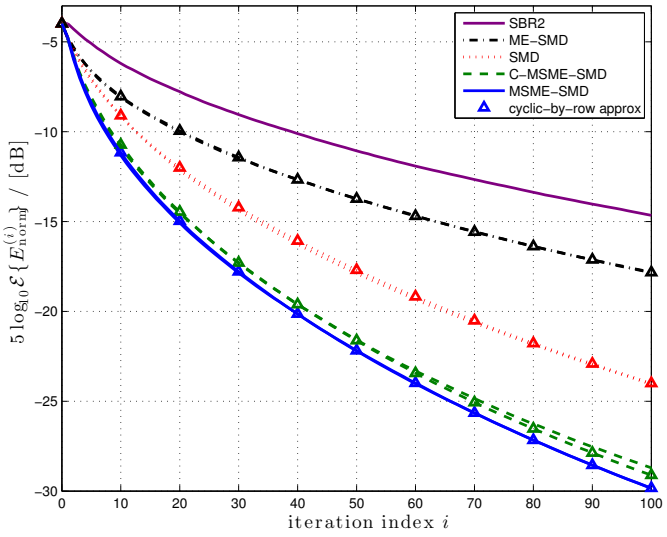

Fig. 2. Normalised remaining off-diagonal energy $E^{(i)}_{\mathrm{norm}}$ according to (16) for various iterative PEVD algorithms versus iterations.

To measure diagonalisation, the remaining normalised off-diagonal energy after i iterations,

$$E^{(i)}_{\mathrm{norm}} = \frac{\sum_\tau \sum_{k=1}^{M} \|\hat{\mathbf{s}}^{(i)}_k[\tau]\|_2^2}{\sum_\tau \|\mathbf{R}[\tau]\|_{\mathrm{F}}^2} \quad , \tag{16}$$

is considered, where $\hat{\mathbf{s}}^{(i)}_k[\tau]$ is the modified column vector of (9) and $\|\cdot\|_{\mathrm{F}}$ the Frobenius norm. The computational complexity of various algorithms is measured by the execution time in Matlab R2013a under Ubuntu 12.04 on a Dell Precision T3610 with Intel® Xeon® E5-1607V2 3.00 GHz x 4 cores and 7.7 GB RAM.

B. Convergence Speed

The diagonalisation performance versus iterations is shown in Fig. 2 for the various algorithms. The SMD family converges generally significantly faster than SBR2, as also highlighted in [7]–[9], with the multiple-shift versions performing best. Interestingly, the cyclic-by row single Jacobi sweep approximations of the EVD lead to no noticeable performance degradation for the SMD family of algorithms, thereby confirming the single sweep selection in Sec. III-B.

C. Calculation Cost

The computation time required for i iterations of the different iterative PVD algorithms is plotted in Fig. 3. Compared to SBR2, the SMD algorithms require much more processing time. However, a significant reduction in cost can be noticed for the cyclic-by-row approximations of SMD algorithms. These are still more costly than the SBR2 algorithm, which only requires a single Givens rotation per iteration and therefore is guaranteed to have a lower complexity.

More interesting than the cost per iteration is the required execution time to reach a specific diagonalisation. Fig. 4 shows the normalised remaining off-diagonal energy in dependency of the time taken to calculate this specific decomposition. This

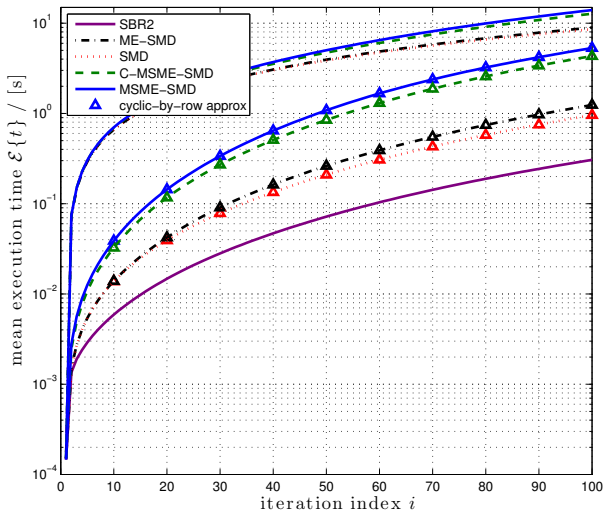

Fig. 3. Execution time vs iterations for various iterative PEVD algorithms.

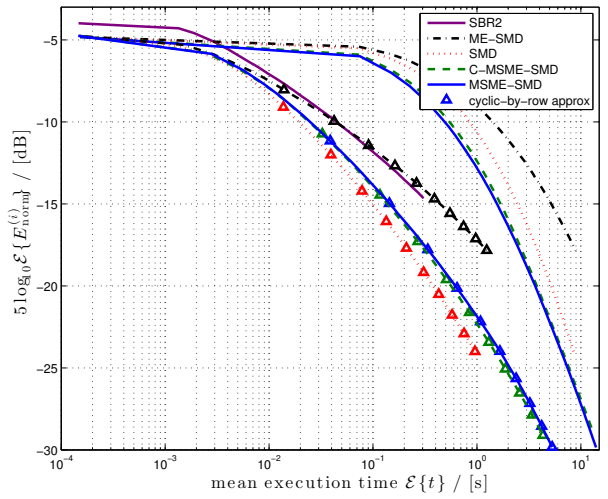

Fig. 4. Normalised remaining off-diagonal energy versus mean execution time for various iterative PEVD algorithms.

graph is obtained by merging the information of Fig. 2 and 3 through eliminating the number of iterations. Using a full EVD, the SMD family of algorithms are inferior to SBR2 is only low diagonalisation is required. For high values of diagonalisation, SBR2 cannot provide the required diagonalisation, and the computationally expensive SMD family is on the only option, with particular benefit for multiple-shift versions as established in [7], [8].

Using the cyclic-by-row approximation, the actual cost to reach a specific level of diagonalisation is reduced below even what is required for SBR2. Therefore, the cyclic-by-row approximation of SMD algorithms does not only yield lower application cost — as established in [7], [9], SMD decompositions yield paraunitary $\boldsymbol{H}(z)$ of lower order — but in terms of total cost per decomposition is also less expensive to calculate than SBR2.

V. Conclusion

This paper has presented a cyclic-by-row approximation of sequential matrix diagonalisation algorithms. Advantageous in application because of their excellent diagonalisation performance and lower order paraunitary filter banks compared to sequential best rotation algorithms, the SMD family has been previously computationally costly to calculate due an EVD required at every iteration. This costly step has been replaced by a single Jacobi sweep using a fixed number of Givens rotations in a cyclic-by-row approach.

Simulation results indicate that the algorithm performance is uncompromised by this approximation, such that SMD algorithms retain their superior features for application. However, the computational cost is reduced such that a significantly lower complexity to calculate a decomposition compared to standard SMD implementation is achieved. This holds also for lower levels of diagonalisation, where SBR2 previously retained an advantage in terms of calculation cost, such that the proposed cyclic-by-row approximations of SMD algrithms now globally offer the best cost-performance trade-off both in terms of calculating and applying an iterative PEVD.

Acknowledgement

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/K014307/1 and the MOD University Defence Research Collaboration in Signal Processing.

References

[1] P. Vaidyanathan. Theory of optimal orthonormal subband coders. *IEEE Trans SP*, 46(6):1528–1543, June 1998.

[2] P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice Hall, New Jersey, 1993.

[3] 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.

[4] S. Icart and P. Comon. Some properties of Laurent polynomial matrices. In *9th IMA Conf. Mathematics in Signal Processing*, Birmingham, UK, Dec. 2012.

[5] A. Tkacenko. Approximate eigenvalue decomposition of para-hermitian systems through successive fir paraunitary transformations. In *IEEE Int. Conf. Acoustics Speech Signal Proc.*, pp. 4074–4077, Mar. 2010.

[6] 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.

[7] J. Corr, K. Thompson, S. Weiss, J.G. McWhirter, S. Redif, I.K. Proudler Multiple Shift Maximum Element Sequential Matrix Diagonalisation for Parahermitian Matrices. Submitted to *IEEE Workshop on Statistical Signal Proc.*, Gold Coast, Australia, June. 2014.

[8] J. Corr, K. Thompson, S. Weiss, J. G. McWhirter, I. K. Proudler Causal Multiple Shift Maximum Element Sequential Matrix Diagonalisation for Parahermitian Matrices. Submitted to *22nd European Signal Proc. Conf.*, Lisbon, Portugal, Sept. 2014.

[9] S. Redif, S. Weiss, and J.G. McWhirter. Sequential Matrix Diagonalisation Algorithms for Polynomial EVD of Parahermitian Matrices. *IEEE Trans SP*, submitted, 2014.

[10] M. Alrmah, S. Weiss, and S. Lambotharan. An extension of the MUSIC algorithm to broadband scenarios using polynomial eigenvalue decomposition. In *19th European Signal Proc. Conf.*, pp. 629–633, Barcelona, Spain, Aug. 2011.

[11] J. Götze Monitoring the stage of diagonalization in Jacobi-type methods *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Adelaide, Australia, vol. III, pp. 441–444, Apr. 2014.

[12] G.H. Golub and C.F. Van Loan. *Matrix Computations*. John Hopkins University Press, 3rd ed., 1996.