ROW-SHIFT CORRECTED TRUNCATION OF PARAUNITARY MATRICES FOR PEVD ALGORITHMS

Jamie Corr, Keith Thompson*, Stephan Weiss*, Ian K. Proudler†, John G. McWhirter‡*

* Department of Electronic & Electrical Engineering, University of Strathclyde, Glasgow, Scotland
† School of Electrical, Electronics & Systems Engineering, Loughborough Univ., Loughborough, UK
‡ School of Engineering, Cardiff University, Cardiff, Wales, UK

ABSTRACT

In this paper, we show that the paraunitary (PU) matrices that arise from the polynomial eigenvalue decomposition (PEVD) of a parahermitian matrix are not unique. In particular, arbitrary shifts (delays) of polynomials in one row of a PU matrix yield another PU matrix that admits the same PEVD. To keep the order of such a PU matrix as low as possible, we propose a row-shift correction. Using the example of an iterative PEVD algorithm with previously proposed truncation of the PU matrix, we demonstrate that a considerable shortening of the PU order can be accomplished when using row-corrected truncation.

1. INTRODUCTION

For broadband array processing problems, delay and multipath propagation cannot be sufficiently captured by phase shifts of the data as done in the narrowband case. Instead, explicit lag elements must be included when e.g. formulating a space-time covariance matrix in the time domain. This covariance matrix $\mathbf{R}[\tau]$ therefore contains both spatial and temporal dimensions, and its z-transform, $\boldsymbol{R}(z) \bullet\!\!-\!\!\circ\, \mathbf{R}[\tau]$, yields a polynomial cross-spectral density (CSD) matrix, which can either be viewed as a matrix containing polynomial entries, or as a polynomial with matrix-valued coefficients.

In the narrowband case, the eigenvalue decomposition (EVD) of a covariance matrix provides a factorisation that forms the basis of numerous optimal signal processing techniques. To extend the utility of the EVD to the polynomial matrix case, a polynomial EVD (PEVD) has been proposed in [8]. A parahermitian matrix, $\boldsymbol{R}(z)$, has the property $\boldsymbol{R}(z) = \tilde{\boldsymbol{R}}(z)$, whereby the parahermitian operator $\tilde{\{\cdot\}}$ consists of a Hermitian transposition $\{\cdot\}^{\mathrm{H}}$ and time reversal i.e. $\tilde{\boldsymbol{R}}(z) = \boldsymbol{R}^{\mathrm{H}}(z^{-1})$. This permits an approaximate factorisation

$$\boldsymbol{R}(z) \approx \tilde{\boldsymbol{Q}}(z)\boldsymbol{D}(z)\boldsymbol{Q}(z) \quad . \tag{1}$$

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

The factor $\boldsymbol{Q}(z)$ is paraunitary, i.e. $\boldsymbol{Q}(z)\tilde{\boldsymbol{Q}}(z) = \mathbf{I}$, and $\boldsymbol{D}(z)$ is diagonal,

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

As an extension of an ordered EVD [5], the polynomial eigenvalues in $\boldsymbol{D}(z)$ are spectrally majorised, such that the power spectral densities $D_m(e^{j\Omega}) = D_m(z)|_{z=e^{j\Omega}}$ satisfy

$$D_{m+1}(e^{j\Omega}) \geq D_m(e^{j\Omega}),\ \forall\ \Omega,\ m = 0 \ldots M-1 \quad . \tag{3}$$

Equality in (1) for FIR paraunitary matrices is not guaranteed [8], but is likely to be valid in close approximation for high orders of $\boldsymbol{Q}(z)$ [6].

The PEVD can be used in various applications including filter bank-based channel coding [16], design of broadband precoding and equalisation of MIMO systems [11], subband coding [9], broadband angle of arrival estimation [1], and others. Some of these methods rely on polynomial subspace decomposition techniques [1, 11, 16], where the order of the extracted paraunitary matrices directly impacts on the implementation complexity for these applications.

To approximate (1), a number of iterative algorithms have been developed. An approximate PEVD (APEVD) algorithm with fixed order has been reported in [13], but has not been proved to converge. Other algorithms have been proven to converge towards a diagonalised $\boldsymbol{D}(z)$ and can achieve better diagonalisation than APEVD, including the family of second order sequential best rotation (SBR2) algorithms [8, 9] and the family of sequential matrix diagonalisation (SMD) algorithms [2, 10]. However, although guaranteed to diagonalise $\boldsymbol{R}(z)$, the latter algorithms [2, 8–10] are unconstrained in order and therefore the polynomial degrees of both $\boldsymbol{D}(z)$ and $\boldsymbol{Q}(z)$ grow with the number of iterations.

The order growth in the parahermitian matrix is problematic, as such an increase will lead to a significant increase in computational complexity of the schemes [2, 8–10] as iterations go on. Therefore, trimming small coefficients at the ends of this matrix has been suggested in [4,8] in order to curb the complexity of iterative PEVD algorithms such as SBR2. The paraunitary matrix also grows with the number of iterations; while this does not impact on the complexity during

978-0-9928626-3-3/15/$31.00 ©2015 IEEE

iterations, the application of the final paraunitary matrix can be costly for polynomial subspace-based applications as mentioned above. Therefore, trimming of the paraunitary matrix has been performed in [12], whereby similarly to [4, 8] small outer matrix coefficients are truncated.

Sec. 2, reviews the SBR2 algorithm which will be used here to produce the paraunitary matrices. The manifold of paraunitary matrices that can admit an otherwise identical PEVD is demonstrated in Sec. 2. Based on the ambiguity identified in Sec. 2, we propose a new truncation method in Sec. 4, that finds a paraunitary matrix with a lower order. The approach is simulated and benchmarked in Sec. 5, with conclusions drawn in Sec. 6.

2. SECOND ORDER SEQUENTIAL BEST ROTATION

The second order sequential best rotation (SBR2) algorithm approximates the PEVD using a series of elementary paraunitary operations to iteratively diagonalise the parahermitian matrix, $\boldsymbol{R}(z)$. Each elementary paraunitary operation is made up of two steps: first a delay step is used to bring the maximum off diagonal element onto the zero lag, then a Jacobi step transfers its energy onto the diagonal.

The SBR2 algorithm is initialised with $\boldsymbol{S}^{(0)}(z) = \boldsymbol{R}(z)$, and the ith iteration begins by locating the maximum off-diagonal element using

$$\{k^{(i)}, \tau^{(i)}\} = \arg\max_{k,\tau} \|\hat{\mathbf{s}}_k^{(i-1)}[\tau]\|_\infty \;, \quad i = 1 \dots I \;, \quad (4)$$

where $\hat{\mathbf{s}}_k^{(i-1)}[\tau]$ is the modified $k^{(i)}$th column vector of $\mathbf{S}^{(i-1)}[\tau]$, with $\mathbf{S}^{(i-1)}[\tau] \circ\!\!-\!\!\bullet\, \boldsymbol{S}^{(i-1)}(z)$ a transform pair, containing all the elements apart from the one on the diagonal. The parameters $k^{(i)}$ and $\tau^{(i)}$ are then used to bring the maximum element onto the zero lag with the delay step,

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

where the delay matrix, $\boldsymbol{\Lambda}^{(i)}(z)$ is set to

$$\boldsymbol{\Lambda}^{(i)}(z) = \mathrm{diag}\{\underbrace{1 \dots 1}_{k^{(i)}-1}\; z^{-\tau^{(i)}}\; \underbrace{1 \dots 1}_{M-k^{(i)}}\} \quad . \quad (6)$$

To conclude the ith iteration of SBR2, the maximum off-diagonal element is eliminated by a Jacobi rotation $\mathbf{Q}^{(i)}$ [8],

$$\boldsymbol{S}^{(i)}(z) = \mathbf{Q}^{(i)}\boldsymbol{S}^{(i)\prime}(z)\mathbf{Q}^{(i)\mathrm{H}} \quad . \quad (7)$$

The SBR2 algorithm stops after I iterations when either a predefined number of steps have elapsed or the maximum off-diagonal element in (4) falls below a predefined threshold ρ. The approximate decomposition $\boldsymbol{R}(z) = \tilde{\hat{\boldsymbol{Q}}}(z)\hat{\boldsymbol{D}}(z)\hat{\boldsymbol{Q}}$ is obtained with $\hat{\boldsymbol{D}}(z) = \boldsymbol{S}^{(I)}(z)$ and

$$\hat{\boldsymbol{Q}}(z) = \prod_{i=1}^{I} \mathbf{Q}^{(i)}\boldsymbol{\Lambda}^{(i)}(z) \quad . \quad (8)$$

Curtailing the growing order of $\boldsymbol{S}^{(i)}(z)$ during the execution of SBR2 according to [4, 8] will benefit the complexities of the parameter search in (4) and the matrix multiplications in (5) and (7). Trimming the paraunitary matrix could also be performed on a step-by-step basis. However, since this has little impact on the execution cost of SBR2, and if memory is not considered, a single truncation of $\hat{\boldsymbol{Q}}(z)$ after convergence suffices and will be discussed below.

3. PEVD AMBIGUITY

We shall investigate the uniqueness of a PEVD. For this purpose, we assume that for a parahermitian $\boldsymbol{R}(z)$, (1) holds with equality, and ask whether a second decomposition

$$\boldsymbol{R}(z) = \tilde{\boldsymbol{Q}}(z)\boldsymbol{D}(z)\boldsymbol{Q}(z) = \tilde{\bar{\boldsymbol{Q}}}(z)\bar{\boldsymbol{D}}(z)\bar{\boldsymbol{Q}}(z) \quad (9)$$

can be found.

With diagonalisation and spectral majorisation of $\boldsymbol{D}(z)$ providing uniqueness [14], it follows that $\bar{\boldsymbol{D}}(z) = \boldsymbol{D}(z)$. Hence, writing $\bar{\boldsymbol{Q}}(z) = \boldsymbol{\Gamma}(z)\boldsymbol{Q}(z)$, the modifying matrix $\boldsymbol{\Gamma}(z)$ must be paraunitary, diagonal and contain allpass filters in order to not affect $\boldsymbol{D}(z)$. While for general allpass filters either $\boldsymbol{\Gamma}(z)$ or $\tilde{\boldsymbol{\Gamma}}(z)$ can be unstable, a simple selection

$$\boldsymbol{\Gamma}(z) = \mathrm{diag}\{z^{-\tau_1}\; z^{-\tau_2}\; \dots\; z^{-\tau_M}\} \quad (10)$$

is possible. This shifts the mth row of $\boldsymbol{Q}(z)$ by τ_m samples, where $m = 1 \dots M$ and M is the spatial dimension of $\boldsymbol{R}(z)$. A similar paraunitary ambiguity has been stated in [7].

Therefore, even if the diagonal $\boldsymbol{D}(z)$ is unique, a paraunitary matrix is ambiguous as a $\boldsymbol{Q}(z)$ of minimum order can be modified by row-shifts to $\boldsymbol{Q}(z)$ applied by $\boldsymbol{\Gamma}(z)$ to yield a factorisation with identical $\{\boldsymbol{R}(z), \boldsymbol{D}(z)\}$. Below, we will exploit this ambiguity in the paraunitary matrix to find a $\boldsymbol{\Gamma}(z)$ which reduces order of $\boldsymbol{Q}(z)$ from the factorisation returned by an iterative PEVD algorithm, which here exemplary uses SBR2.

4. SHIFT-CORRECTED TRUNCATION OF PARAUNITARY MATRICES

The truncation of paraunitary matrices in [12] follows the idea for trimming parahermitian matrices expressed in [4, 8]. Below, we briefly review the approach in [12], before the proposed approach is outlined, followed by a numerical example. Any truncation of paraunitary matrices results in a loss of the paraunitary property, which will be discussed further in the results section.

4.1. State-of-the-Art Truncation

The truncation method in [12] can remove up to a predefined proportion of energy μ from $\hat{\boldsymbol{Q}}(z) \bullet\!\!-\!\!\circ\, \hat{\mathbf{Q}}[n]$. If truncation is

written as a non-linear operation $f_{\text{trim}}(\cdot)$, then the proportion of removed energy is given by

$$\begin{aligned}\gamma_{\text{trim}} &= 1 - \frac{\sum_n \|f_{\text{trim}}(\hat{\mathbf{Q}}[n])\|_{\text{F}}^2}{\sum_n \|\hat{\mathbf{Q}}[n]\|_{\text{F}}^2} \\ &= 1 - \frac{1}{M}\sum_n \|f_{\text{trim}}(\hat{\mathbf{Q}}[n])\|_{\text{F}}^2 \quad ,\end{aligned}$$

where $\|\cdot\|_{\text{F}}$ is the Frobenius norm. The energy is removed by omitting the leading N_1 and trailing N_2 matrices from $\hat{\mathbf{Q}}[n]$ of length N, such that

$$f_{\text{trim}}(\hat{\mathbf{Q}}[n]) = \begin{cases} \hat{\mathbf{Q}}[n+N_1] & 0 \le n < N - N_2 - N_1 \\ \mathbf{0} & \text{otherwise} \end{cases} .$$

This leads to the following constrained optimisation problem to perform the truncation:

$$\text{maximise} \quad (N_1 + N_2) \tag{11}$$

$$\text{s.t.} \quad \gamma_{\text{trim}} \le \mu \quad . \tag{12}$$

In practise, this approach can be implemented by sequentially removing leading or trailing matrices of $\hat{\mathbf{Q}}[n]$ — which ever has the smallest Frobenius norm — as long as the constraint (12) remains satisfied.

4.2. Proposed Row-Corrected Truncation

Defining $\hat{\boldsymbol{Q}}(z)$ with its constituent row vectors $\hat{\mathbf{q}}_m(z)$, $m = 1 \dots M$,

$$\tilde{\hat{\boldsymbol{Q}}}(z) = [\hat{\mathbf{q}}_1(z) \ \dots \ \hat{\mathbf{q}}_M(z)] \quad , \tag{13}$$

note that $\tilde{\hat{\mathbf{q}}}_i(z)\hat{\mathbf{q}}_j(z) = \delta(i-j)$. Therefore each vector $\hat{\mathbf{q}}_m(z)$ has unit energy, and it appears sensible to truncate the same proportion of energy from every vector. With a vector-valued truncation $f_{\text{shift}}(\hat{\mathbf{q}}_m[n])$, the proportion of removed energy is

$$\gamma_{\text{shift},m} = 1 - \sum_n \|f_{\text{shift}}(\hat{\mathbf{q}}_m[n])\|_2^2 \quad . \tag{14}$$

Based on the truncation definition

$$f_{\text{shift}}(\hat{\mathbf{q}}_m[n]) = \begin{cases} \hat{\mathbf{q}}_m[n+N_{1,m}] & 0 \le n < T_m \\ \mathbf{0} & \text{otherwise} \end{cases} , \tag{15}$$

with $T_m = N - N_{2,m} - N_{1,m}$, the optimum truncation based on row-correction is given by the constrained problem

$$\text{maximise} \quad \min_m (N_{1,m} + N_{2,m}) \tag{16}$$

$$\text{s.t.} \quad \gamma_{\text{shift},m} \le \frac{\mu'}{M} \ \forall\, m = 1 \dots M \quad , \tag{17}$$

where μ' is the threshold of energy shed. With this, the row-shifts $\tau_m = N_{1,m}, m = 1 \dots M$, correcting the truncation are identified and can be applied via $\boldsymbol{\Gamma}(z)$ in (10) . The truncated matrix after row correction will have length $\max_m T_m$.

In practise, every row vector of $\hat{\boldsymbol{Q}}(z)$ is treated individually like the matrix in the previous approach of [12] and Sec. 4.1. Note that the main complexity of both truncation approaches lies in the calculation of norms; therefore, the proposed approach has only little overhead compared to [12].

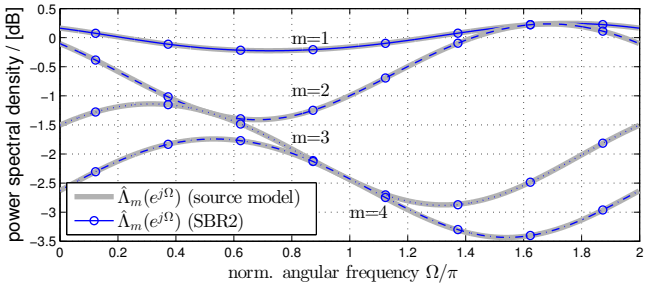

Fig. 1. Power spectral densities of the source model $\boldsymbol{D}(z)$ and of the extracted matrix $\hat{\boldsymbol{D}}(z)$ using SBR2.

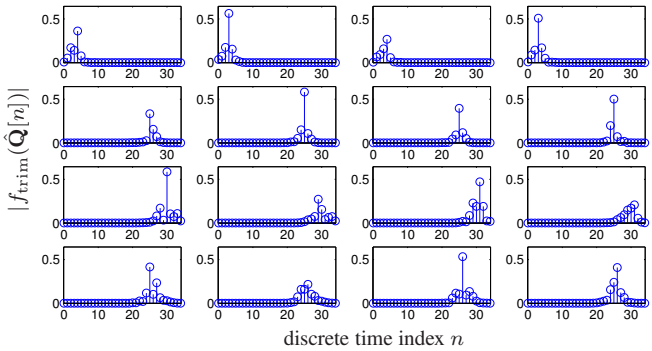

Fig. 2. Paraunitary matrix truncated with $\mu = 10^{-4}$ using the state-of-the-art approach in [12].

4.3. Truncation Example

To demonstrate the potential benefit of the proposed truncation, a simple example is considered here. By generating a CSD matrix $\boldsymbol{R}(z) \in \mathbb{C}^{4\times4}$ through a source model detailed in [10], we know that an exact decomposition $\boldsymbol{R}(z) = \tilde{\boldsymbol{Q}}(z)\boldsymbol{D}(z)\boldsymbol{Q}(z)$ exists. The matrix $\boldsymbol{D}(z)$ is diagonal and of order 8; it is also spectrally majorised as shown by the shaded curves in Fig. 1. The paraunitary matrix $\boldsymbol{Q}(z)$ is of order 4.

Running SBR2 for 100 iterations yields a well-diagonalised matrix $\hat{\boldsymbol{D}}(z)$, whose power spectral densities very closely match those of $\boldsymbol{D}(z)$, as demonstrated in Fig. 1. This accuracy is not met by the paraunitary matrix $\hat{\boldsymbol{Q}}(z)$, which, when left untrimmed, has an order of 181. Even though this matrix has many very small trailing coefficients, its polynomial degree is almost two orders of magnitude larger than that of the ground truth matrix $\boldsymbol{Q}(z)$.

Using a standard truncation as introduced in [12] with $\mu = 10^{-4}$ removes 0.1‰ of the total energy of $\hat{\boldsymbol{Q}}(z)$. The resulting $f_{\text{trim}}(\hat{\boldsymbol{Q}}[n])$ is shown in Fig. 2, and now only has order 33. Removing small trailing coefficients therefore has significantly reduced the order of $f_{\text{trim}}(\hat{\boldsymbol{Q}}[n])$, and therefore the computational complexity that is required to implement such a system.

In Fig. 2, it is noticeable that the rows of $f_{\text{trim}}(\hat{\mathbf{Q}}[n])$ are shifted with respect to each other: particularly the first row exhibits an advance compared to the remaining three, which is an indication of the manifold w.r.t. row shifts established

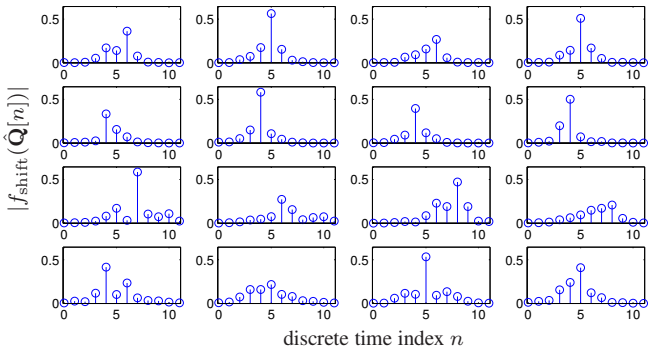

Fig. 3. Paraunitary matrix truncated with $\mu = 10^{-4}$ using the proposed approach.

during the analysis in Sec. 3. Therefore, with the proposed row-corrected truncation algorithm and the same shedding of 0.1‰ energy from $\hat{\mathbf{Q}}[n]$, the resulting $f_{\text{shift}}(\hat{\mathbf{Q}}[n])$ of only order 11 is shown in Fig. 3. Here the modifying matrix, $\boldsymbol{\Gamma}(z)$, is $\text{diag}\{z^{-25}\ z^{-2}\ 1\ z^{-2}\}$.

Even though the diagonalised matrices $\boldsymbol{D}(z)$ and $\hat{\boldsymbol{D}}(z)$ are similar, the paraunitary matrix $\hat{\boldsymbol{Q}}(z)$ differs substantially from $\boldsymbol{Q}(z)$, and $f_{\text{shift}}(\hat{\mathbf{Q}}[n])$ only approaches $\mathbf{Q}[n] \circ\!\!-\!\!\bullet\, \boldsymbol{Q}(z)$ in order but not appearance. Similar effects are known from the EVD, where small disturbances result in similar energies being extracted by eigenvalues, but much larger differences can emerge in the eigenvectors [3]. Irrespective of this, the proposed truncation approach appears very worthwhile in reducing the order of $\boldsymbol{Q}(z)$, which will be more exhaustively demonstrated in the following section.

5. RESULTS

To benchmark the proposed truncation approach, this section first defines performance metrics before setting out a simulation scenario, over which then simulations will be performed.

5.1. Performance Metrics

Reconstruction Error. By truncating $\hat{\boldsymbol{Q}}(z)$, its paraunitarity is lost. If interpreting $\hat{\boldsymbol{Q}}(z)$ as a filter bank, the loss manifests itself as reconstruction error [15], and the difference to a paraunitary system can be assessed as

$$\boldsymbol{E}(z) = \mathbf{I}_{M\times M} - \hat{\boldsymbol{Q}}_{\text{T}}(z)\tilde{\hat{\boldsymbol{Q}}}_{\text{T}}(z) \quad . \tag{18}$$

where $\hat{\boldsymbol{Q}}_{\text{T}}(z)$ is the truncated matrix, and with $\mathbf{E}[\tau] \circ\!\!-\!\!\bullet\, \boldsymbol{E}(z)$ the reconstruction error is given by

$$\xi = \frac{1}{M}\sum_{\tau} \|\mathbf{E}[\tau]\|_F^2 \quad . \tag{19}$$

Diagonalisation. Since SBR2 iteratively minimises off-diagonal energy, a suitable normalised metric from [10] is

$$E_{\text{norm}}^{(i)} = \frac{\sum_\tau \sum_{k=1}^{M} \|\hat{\mathbf{s}}_k^{(i)}[\tau]\|_2^2}{\sum_\tau \|\mathbf{R}[\tau]\|_{\text{F}}^2} \tag{20}$$

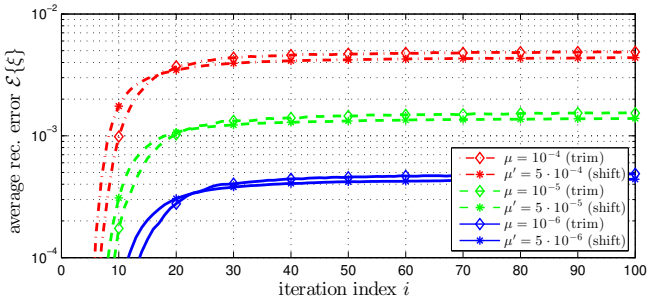

Fig. 4. Ensemble reconstruction error $\mathcal{E}\{\xi\}$ vs. SBR2 iterations for the different truncation approaches and varying μ.

based on the definition of $\hat{\mathbf{s}}_k^{(i)}[\tau]$ in (4).

5.2. Simulation Scenario

The simulations below have been performed over an ensemble of 10^3 instantiations of $\boldsymbol{R}(z) \in \mathbb{C}^{6\times6}$ based on the randomised source model in [10]. In this source model, the order of $\boldsymbol{D}(z)$ is 24 and the order of $\boldsymbol{Q}(z)$ is 12, such that the total order of $\boldsymbol{R}(z)$ is 47. The dynamic range of the source model is constrained to ensure that in the ensemble the average is around 25 dB. SBR2 is run with 100 iterations, and at every iteration step the metrics defined in Sec. 5.1 are recorded, together with the order of the paraunitary matrices.

5.3. Reconstruction Error

The experiments were repeated for 3 different truncation parameters $\mu = \{10^{-6}, 10^{-5}, 10^{-4}\}$ for $f_{\text{trim}}(\cdot)$ with the resulting reconstruction error ξ shown in Fig. 4. With low iteration numbers, $\hat{\boldsymbol{Q}}(z)$ is still of low order and there is limited choice for trimming, but with increased i, the truncation performs asymptotically to trim $\hat{\boldsymbol{Q}}(z)$ by exactly μ.

With the proposed approach, it was found that μ can be scaled up by a factor of 5 to reach the same error metric as the standard truncation, as also shown in Fig. 4. This more aggressive trimming for the same error metrics can be justified since in the standard truncation to remove whole matrix coefficients at the ends of $\hat{\boldsymbol{Q}}(z)$ leads to larger errors ξ. In contrast, the proposed approach will truncate small coefficients evenly across rows and balance the overall in ξ.

5.4. Truncated Order and Diagonalisation

Using the different truncations $\mu = \{10^{-6}, 10^{-5}, 10^{-4}\}$ for the standard $f_{\text{trim}}(\cdot)$ and $\mu' = 5\mu$ for the proposed $f_{\text{shift}}(\cdot)$, the order of the truncated matrices $f_{\text{trim}}(\hat{\mathbf{Q}}[n])$ and $f_{\text{shift}}(\hat{\mathbf{Q}}[n])$ are shown in Fig. 5. As indicated in the example in Figs. 2 and 3, the proposed approach achieves a significant reduction in the order of the paraunitary matrices after truncation.

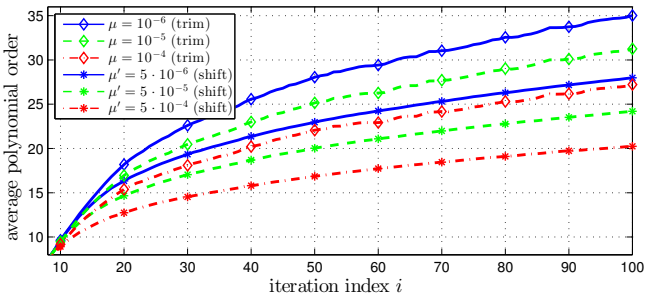

Fig. 5. Average order after truncation of $\hat{\boldsymbol{Q}}(z)$ vs. SBR2 iterations for the different truncation approaches and varying μ.

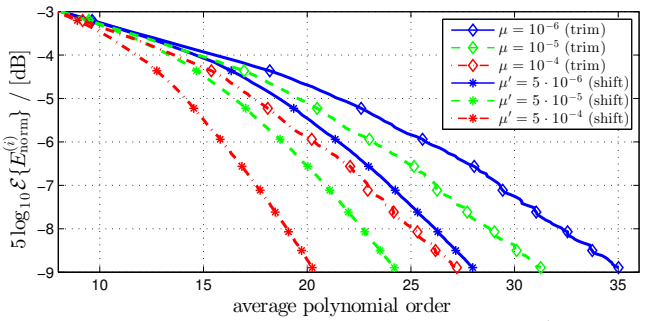

Fig. 6. Diagonalisation metric vs. average order of $\hat{\boldsymbol{Q}}(z)$ after truncation.

For an application rather than the calculation of the PEVD, the number of iterations required for SBR2 do not matter, the performance criteria are the achieved diagonalisation and the computation cost to implement $f_{\text{trim}}(\hat{\mathbf{Q}}[n])$ and $f_{\text{shift}}(\hat{\mathbf{Q}}[n])$. Fig. 6 shows the achievable average diagonalisation from (20) vs. the order of the truncated paraunitary matrix showing a significant performance advantage for the proposed truncation over the current approach [12].

6. CONCLUSION

The ambiguity in the paraunitary matrix of a polynomial EVD w.r.t arbitrary row shifts has been exploited to reduce the order of the factorisation returned by iterative PEVD algorithms such as SBR2. We have proposed a shift-corrected truncation that can find a lower order decomposition than state-of-the-art. The results show that we can achieve the same performance metrics as an existing method with a more aggressive truncation since the overall error is better balanced across all rows. In the examples the the source model dynamic range is carefully constrained but when it is greater the performance gap is larger. When designing PEVD implementations for real applications on finite wordlength processors, a loss in paraunitarity is inevitable and can be carefully controlled with the proposed method.

Code for our method, simulations and figures is available at `pevd-toolbox.eee.strath.ac.uk`.

REFERENCES

[1] M. Alrmah, S. Weiss, S. Lambotharan. An extension of the music algorithm to broadband scenarios using polynomial eigenvalue decomposition. In *EUSIPCO*, pp. 629–633, Barcelona, Spain, Aug. 2011.

[2] J. Corr, K. Thompson, S. Weiss, J. McWhirter, S. Redif, I. Proudler. Multiple shift maximum element sequential matrix diagonalisation for parahermitian matrices. In *IEEE SSP*, pp. 312–315, Gold Coast, Australia, June 2014.

[3] T.R. Crossley, B. Porter. Eigenvalue and eigenvector sensitivities in linear systems theory. *Int. J. Control*, 10(2):163–170, 1969.

[4] J. Foster, J.G. McWhirter, J. Chambers. Limiting the order of polynomial matrices within the SBR2 algorithm. In *IMA Math. Signal Proc.*, Cirencester, UK, Dec. 2006.

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

[6] S. Icart, P. Comon. Some properties of Laurent polynomial matrices. In *Conf. Math. Signal Proc.*, Birmingham, UK, Dec. 2012.

[7] A. Jafarian, J. McWhirter. A novel method for multichannel spectral factorization. In *EUSIPCO*, pp. 1069–1073, Bucharest, Romania, Aug. 2012.

[8] J.G. McWhirter, P.D. Baxter, T. Cooper, S. Redif, J. Foster. An EVD algorithm for para-hermitian polynomial matrices. *IEEE Trans. SP*, 55(5):2158–2169, May 2007.

[9] S. Redif, J. McWhirter, 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.

[10] S. Redif, S. Weiss, J. McWhirter. Sequential matrix diagonalization algorithms for polynomial EVD of parahermitian matrices. *IEEE Trans. SP*, 63(1):81–89, Jan. 2015.

[11] C.H. Ta, S. Weiss. A Design of Precoding and Equalisation for Broadband MIMO Systems. In *15th Int. Conf. DSP*, pp. 571–574, Cardiff, UK, July 2007.

[12] C.H. Ta, S. Weiss. Shortening the order of paraunitary matrices in SBR2 algorithm. In *Int. Conf. Inf. Comm. & SP*, pp. 1-5, Singapore, Dec. 2007.

[13] A. Tkacenko. Approximate eigenvalue decomposition of para-hermitian systems through successive fir paraunitary transformations. In *IEEE ICASSP*, pp. 4074–4077, Dallas, TX, Mar. 2010.

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

[15] P.P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice Hall, 1993.

[16] S. Weiss, S. Redif, T. Cooper, C. Liu, P.D. Baxter, J.G. McWhirter. Paraunitary oversampled filter bank design for channel coding. *J. Applied SP*, 2006.