



Strathprints Institutional Repository

Corr, Jamie and Thompson, Keith and Weiss, Stephan and Proudler, Ian K. and McWhirter, John G. (2015) Shortening of paraunitary matrices obtained by polynomial eigenvalue decomposition algorithms. In: 5th Conference of the Sensor Signal Processing for Defence, 2015-07-09 - 2015-07-10, Scotland. , <http://dx.doi.org/10.1109/SSPD.2015.7288523>

This version is available at <http://strathprints.strath.ac.uk/53817/>

Strathprints is designed to allow users to access the research output of the University of Strathclyde. Unless otherwise explicitly stated on the manuscript, Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. Please check the manuscript for details of any other licences that may have been applied. You may not engage in further distribution of the material for any profitmaking activities or any commercial gain. You may freely distribute both the url (<http://strathprints.strath.ac.uk/>) and the content of this paper for research or private study, educational, or not-for-profit purposes without prior permission or charge.

Any correspondence concerning this service should be sent to Strathprints administrator: strathprints@strath.ac.uk

Shortening of Paraunitary Matrices Obtained by Polynomial Eigenvalue Decomposition Algorithms

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

*Department of Electronic & Electrical Engineering, University of Strathclyde, Glasgow, Scotland

†School of Electrical, Electronics and Systems Engineering, Loughborough University, Loughborough, UK

‡School of Engineering, Cardiff University, Cardiff, Wales, UK

{jamie.corr,keith.thompson,stephan.weiss}@strath.ac.uk

Abstract—This paper extends the analysis of the recently introduced row-shift corrected truncation method for paraunitary matrices to those produced by the state-of-the-art sequential matrix diagonalisation (SMD) family of polynomial eigenvalue decomposition (PEVD) algorithms. The row-shift corrected truncation method utilises the ambiguity in the paraunitary matrices to reduce their order. The results presented in this paper compare the effect a simple change in PEVD method can have on the performance of the paraunitary truncation. In the case of the SMD algorithm the benefits of the new approach are reduced compared to what has been seen before however there is still a reduction in both reconstruction error and paraunitary matrix order.

I. INTRODUCTION

Broadband array processing problems are often formulated with a space-time covariance matrix $\mathbf{R}[\tau]$, which in addition to a spatial component contains an explicit lag τ . Its z -transform, $\mathbf{R}(z) \bullet\text{---}\circ \mathbf{R}[\tau]$, yields a polynomial cross-spectral density (CSD) matrix, which extends the symmetric or Hermitian property known from standard matrix algebra to the parahermitian case as $\mathbf{R}(z) = \tilde{\mathbf{R}}(z)$, whereby the parahermitian operator $\{\cdot\}$ consists of a Hermitian transposition $\{\cdot\}^H$ and time reversal i.e. $\tilde{\mathbf{R}}(z) = \mathbf{R}^H(z^{-1})$.

Since the eigenvalue decomposition (EVD) of a covariance matrix provides an optimal factorisation for numerous signal processing problems, its extension from the narrowband to the broadband case has led to polynomial EVD (PEVD) as proposed in [1]. For a parahermitian matrix $\mathbf{R}(z)$,

$$\mathbf{R}(z) \approx \tilde{\mathbf{Q}}(z)\mathbf{D}(z)\mathbf{Q}(z) \quad (1)$$

factorises $\mathbf{R}(z)$ into a paraunitary $\mathbf{Q}(z)$, such that $\mathbf{Q}(z)\tilde{\mathbf{Q}}(z) = \mathbf{I}$, and a diagonal parahermitian $\mathbf{D}(z)$,

$$\mathbf{D}(z) = \text{diag}\{D_0(z) D_1(z) \dots D_{M-1}(z)\} \quad (2)$$

Extending the idea of an ordered EVD [2], the polynomial eigenvalues in $\mathbf{D}(z)$ are spectrally majorised, i.e. their power spectral densities $D_m(e^{j\Omega}) = D_m(z)|_{z=e^{j\Omega}}$ satisfy

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

For FIR paraunitary matrices, the equality in (1) is not guaranteed [1] but likely valid in close approximation for high orders of $\mathbf{Q}(z)$ [3].

The PEVD enables a number of applications spanning from filter bank-based channel coding [4], to the design of

broadband precoding and equalisation of MIMO systems [5], subband coding [6], broadband angle of arrival estimation [7], and others. A number of these applications are directly influenced by the order of the paraunitary matrix $\mathbf{Q}(z)$, such as polynomial subspace decomposition techniques [4], [5], [7], and paraunitary matrices of low order therefore can be crucial.

A number of iterative algorithms have been developed to approximate (1). For example, an approximate PEVD (APEVD) algorithm with fixed order has been reported in [8], but has not been proved to converge. Other algorithms have been proven to converge towards a diagonalised $\mathbf{D}(z)$ and can achieve better diagonalisation than APEVD, including the family of second order sequential best rotation (SBR2) algorithms [1], [6] and the family of sequential matrix diagonalisation (SMD) algorithms [9], [10]. Although guaranteed to diagonalise $\mathbf{R}(z)$, the SBR2 and SMD algorithms [1], [6], [9], [10] are unconstrained in their order and therefore the polynomial degrees of both $\mathbf{D}(z)$ and $\mathbf{Q}(z)$ grow with the number of iterations.

The growing order of the diagonal parahermitian matrix causes difficulties, as its increase is responsible for the rising complexity of iterative PEVD algorithms such as [1], [6], [9], [10] as iterations go on. To this end, trimming small coefficients at the ends of this matrix has been suggested in [1], [11] in order to reduce the complexity of iterative PEVD algorithms. However, the paraunitary matrix also grows with the number of iterations; while its increase does not impact on the complexity during iterations, the application cost of the finally extracted paraunitary matrix can be high for polynomial subspace-based applications as mentioned above. Therefore, reducing the order of the paraunitary matrix has been suggested in [12], whereby similarly to [1], [11] small outer matrix coefficients are truncated.

In addition to the trimming approach in [12], an enhanced alternative method has been suggested in [13] and applied to SBR2. The aim of this paper therefore is to investigate the so-called shift-corrected version in [13] for both SBR2 and SMD families of iterative PEVD algorithms. To accomplish this, Sec. II reviews the PEVD algorithms that will be used to generate paraunitary matrices. Sec. III provides an overview of the two paraunitary truncation approaches to be compared. The results from applying the different truncation methods to the paraunitary matrices produced by the two PEVD methods are presented in Sec. IV and conclusions are given in Sec. V.

II. PEVD ALGORITHMS

A. General Approach

Starting from the parahermitian matrix $\mathbf{R}(z)$, all PEVD algorithms apply a series of elementary paraunitary matrices in order to iterate towards an approximately diagonal polynomial matrix $\mathbf{D}(z)$. At the i -th iteration, building on a partially diagonalised parahermitian matrix $\mathbf{S}^{(i-1)}(z)$, PEVD algorithms build an elementary paraunitary matrix $\mathbf{Q}^{(i)}(z)$ from two components: a shift matrix $\mathbf{\Lambda}^{(i)}(z)$ which transfers large off-diagonal elements of $\mathbf{S}^{(i-1)}(z)$ onto the zero lag, followed by a rotation $\mathbf{Q}^{(i)}$ which moves the transferred elements' energy onto the diagonal. Thus, for the i th iteration we have

$$\mathbf{Q}^{(i)}(z) = \mathbf{Q}^{(i)} \mathbf{\Lambda}^{(i)}(z) \quad (4)$$

While various PEVD algorithms share this same general approach, both $\mathbf{\Lambda}^{(i)}(z)$ and $\mathbf{Q}^{(i)}$ are algorithm dependent. The delay matrix, $\mathbf{\Lambda}^{(i)}(z)$, is determined by the search strategy, while the rotation $\mathbf{Q}^{(i)}$ is defined by the family of algorithms used. The i th iteration of any iterative PEVD algorithm is then implemented as

$$\mathbf{S}^{(i)}(z) = \mathbf{Q}^{(i)} \mathbf{S}^{(i-1)}(z) \tilde{\mathbf{Q}}^{(i)} \quad (5)$$

The algorithm is stopped after I iterations if either the off-diagonal energy of $\mathbf{S}^{(i)}(z)$ falls below a predefined threshold or I reaches a selected limit. Thereafter $\hat{\mathbf{D}}(z) = \mathbf{S}^{(I)}(z)$ and

$$\hat{\mathbf{Q}}(z) = \prod_{i=1}^I \mathbf{Q}^{(i)} \mathbf{\Lambda}^{(i)}(z) \quad (6)$$

Generally, (6) can be calculated after the algorithm has been executed, as storing the parameters of $\mathbf{Q}^{(i)}$ and $\mathbf{\Lambda}^{(i)}(z)$ is more efficient than storing and updating the whole paraunitary matrix. Based on the PEVD outline above the following subsections will go into the unique details for both PEVD algorithms utilised in the results section.

B. Second Order Sequential Best Rotation

The second order sequential best rotation (SBR2) algorithm is an extension of the classical Jacobi algorithm for scalar matrices to the polynomial case [1]. Like the Jacobi algorithm, SBR2 starts with a search for the maximum off diagonal element but this now extends to all lags of the polynomial matrix. Starting from $\mathbf{S}^{(0)}(z) = \mathbf{R}(z)$, at the i -th iteration the maximum off diagonal element is found using

$$\{k^{(i)}, \tau^{(i)}\} = \arg \max_{k, \tau} \|\hat{\mathbf{s}}_k^{(i-1)}[\tau]\|_{\infty}, \quad i = 1 \dots I \quad (7)$$

With the transform pair $\mathbf{S}^{(i-1)}[\tau] \circ \bullet \mathbf{S}^{(i-1)}(z)$, the modified $k^{(i)}$ th column vector, $\hat{\mathbf{s}}_k^{(i-1)}[\tau]$, contains only off diagonal elements. The delay matrix $\mathbf{\Lambda}^{(i)}(z)$ is then constructed using

$$\mathbf{\Lambda}^{(i)}(z) = \text{diag}\left\{ \underbrace{1 \dots 1}_{k^{(i)}-1} z^{-\tau^{(i)}} \underbrace{1 \dots 1}_{M-k^{(i)}} \right\}, \quad (8)$$

where the parameters $k^{(i)}$ and τ are used to advance or delay the $k^{(i)}$ th column and row, shifting the maximum element

by $\tau^{(i)}$ lags onto the zerolag. Finally, the energy from the maximum element is transferred onto the diagonal using the Jacobi transformation

$$\mathbf{Q}^{(i)} = \begin{bmatrix} \mathbf{I}_1 & & & & \\ & \cos \varphi^{(i)} & \dots & e^{j\vartheta^{(i)}} \sin \varphi^{(i)} & \\ & \vdots & & \vdots & \\ -e^{-j\vartheta^{(i)}} \sin \varphi^{(i)} & & \mathbf{I}_2 & & \\ & & & \cos \varphi^{(i)} & \\ & & & & \mathbf{I}_3 \end{bmatrix}, \quad (9)$$

where the rotation angles $\varphi^{(i)}$ and $\vartheta^{(i)}$ are determined by the value of the maximum element. The dimensions of the identities in (9), \mathbf{I}_n , $n = 1, 2, 3$, are $(\min\{m^{(i)}, k^{(i)}\} - 1)$, $(|m^{(i)} - k^{(i)}| - 1)$ and $(M - \max\{m^{(i)}, k^{(i)}\} + 1)$ respectively. Although (9) has to be applied to all lags of the parahermitian matrix its sparse nature means that only two rows and columns are modified when proceeding from $\mathbf{S}^{(i-1)}(z)$ to $\mathbf{S}^{(i)}(z)$.

C. Sequential Matrix Diagonalisation

In addition to the three main PEVD steps described above the sequential matrix diagonalisation (SMD) [9] algorithm includes an additional initialisation step that brings all off diagonal zerolag energy onto the diagonal prior to any shift operations. This results in

$$\mathbf{S}^{(0)}(z) = \mathbf{Q}^{(0)} \mathbf{R}(z) \mathbf{Q}^{(0)H}, \quad (10)$$

where $\mathbf{Q}^{(0)}$ is the modal matrix for the EVD of the zerolag of $\mathbf{R}(z)$.

The search step of the SMD is also different in that the l_{∞} norm in (7) is replaced by an l_2 norm, thereby changing the search from maximum element to maximum column norm

$$\{k^{(i)}, \tau^{(i)}\} = \arg \max_{k, \tau} \|\hat{\mathbf{s}}_k^{(i-1)}[\tau]\|_2 \quad (11)$$

Using the same modified column vector, $\hat{\mathbf{s}}_k^{(i-1)}[\tau]$, the parameters $k^{(i)}$ and $\tau^{(i)}$ are again used in (8). For the SMD algorithm, an energy transfer matrix, $\mathbf{Q}^{(i)}$, that clears all zero lag off-diagonal elements can be found by an EVD at zero lag.

In general, the SMD algorithm transfers more energy per iteration than the SBR2 algorithm. Due to the full EVD being non-sparse and requiring a full matrix multiplication for each lag in the parahermitian matrix, each SMD iteration is more computationally costly than a similar SBR2 iteration. Therefore, SMD overall has a higher complexity than SBR2 to compute an approximate PEVD, but is capable of producing paraunitary matrices of lower order [9], making operations involving $\hat{\mathbf{Q}}(z)$ less costly to apply once the PEVD is calculated.

III. PARAUNITARY MATRIX TRUNCATION METHODS

To reduce the cost of applying the paraunitary matrix $\hat{\mathbf{Q}}(z)$, two different truncation methods have been proposed in [12], [13], which are reviewed below.

A. Lag Based Truncation

The truncation approach specified in [12] reduces the order of the paraunitary matrix by removing the N_1 leading and N_2 trailing lags, unlike the method for parahermitian truncation in [1], [11] this is done asymmetrically. The trim function for paraunitary matrices can be defined as

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

The proportion of energy removed in the N_1 leading and N_2 trailing lags of $\hat{\mathbf{Q}}[n]$ by the $f_{\text{trim}}(\cdot)$ operation is given by

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

where $\|\cdot\|_{\text{F}}$ is the Frobenius norm. To control the impact of the truncation operation on the paraunitary matrix, the parameter μ is used as an upper bound for the proportion of energy removed, γ_{trim} . Here we want to maximise the number of lags removed, $N_1 + N_2$, whilst keeping the energy removed below μ , hence the constrained optimisation problem:

$$\text{maximise } (N_1 + N_2) \quad (12)$$

$$\text{s.t. } \gamma_{\text{trim}} \leq \mu . \quad (13)$$

The implementation of $f_{\text{trim}}(\cdot)$ is as simple as sequentially removing the outermost matrix coefficients of polynomial $\mathbf{Q}(z)$, at either leading or trailing lags, which possess the smallest Frobenius norm whilst ensuring (13) is satisfied.

B. Row-Shift Corrected Truncation

The row-shift corrected truncation method [13] exploits the ambiguity in paraunitary matrices [13], [14]. The ambiguity permits $\mathbf{Q}(z)$, from (1), to be replaced by $\tilde{\mathbf{Q}}(z)$ where $\tilde{\mathbf{Q}}(z) = \mathbf{\Gamma}(z)\mathbf{Q}(z)$. As argued in [13], the only viable option for the polynomial matrix $\mathbf{\Gamma}(z)$ takes the form

$$\mathbf{\Gamma}(z) = \text{diag}\{z^{-\tau_1} \ z^{-\tau_2} \ \dots \ z^{-\tau_M}\} \quad (14)$$

which consists of M row shifts by τ_m samples, where $m = 1 \dots M$, i.e. for each row of the paraunitary matrix. These row shifts can now be used to align the maximum values in each row so that the overall paraunitary matrix can be truncated further.

We can subdivide the paraunitary matrix, $\hat{\mathbf{Q}}(z)$, into its M row vectors $\hat{\mathbf{q}}_m(z)$, $m = 1 \dots M$,

$$\tilde{\hat{\mathbf{Q}}}(z) = [\hat{\mathbf{q}}_1(z) \ \dots \ \hat{\mathbf{q}}_M(z)] . \quad (15)$$

For the row-shift corrected method each row is truncated individually using

$$f_{\text{shift}}(\hat{\mathbf{q}}_m[n]) = \begin{cases} \hat{\mathbf{q}}_m[n + N_{1,m}] & 0 \leq n < T_m \\ \mathbf{0} & \text{otherwise} \end{cases} , \quad (16)$$

where the overall length of the truncated vector is $T_m = N - N_{2,m} - N_{1,m}$. The row shifts, τ_m , in (14) are then set equal

to $N_{1,m} \ \forall m = 1 \dots M$. As each vector has unit energy the proportion of energy to be removed from each row, using the vector-valued truncation, $f_{\text{shift}}(\hat{\mathbf{q}}_m[n])$, becomes

$$\gamma_{\text{shift},m} = 1 - \sum_n \|f_{\text{shift}}(\hat{\mathbf{q}}_m[n])\|_2^2 . \quad (17)$$

Similar to the lag-based method, this presents us with the following constrained optimisation problem for $f_{\text{shift}}(\cdot)$:

$$\text{maximise } \min_m (N_{1,m} + N_{2,m}) \quad (18)$$

$$\text{s.t. } \gamma_{\text{shift},m} \leq \frac{\mu'}{M} \ \forall m = 1 \dots M . \quad (19)$$

The length of resulting paraunitary matrix will be $\max_m T_m$ and the maximum proportion of energy removed will be μ' . The process outlined above is equivalent to truncating each row of $\hat{\mathbf{Q}}(z)$ with the lag based truncation method in Sec. III-A and [12].

IV. RESULTS

To compare the performance of the different truncation approaches on the two PEVD methods, performance metrics for this are first defined, followed by a number of simulation scenarios.

A. Performance Metrics

Reconstruction Error. When the paraunitary matrix, $\hat{\mathbf{Q}}(z)$, is truncated, the paraunitary property is lost. The paraunitary property states $\hat{\mathbf{Q}}(z)\tilde{\hat{\mathbf{Q}}}(z) = \mathbf{I}$ therefore the difference from paraunitary is

$$\mathbf{E}(z) = \mathbf{I}_{M \times M} - \hat{\mathbf{Q}}_{\text{T}}(z)\tilde{\hat{\mathbf{Q}}}_{\text{T}}(z) . \quad (20)$$

with $\hat{\mathbf{Q}}_{\text{T}}(z)$ being the truncated matrix, and the transform $\mathbf{E}[\tau] \circ \bullet \mathbf{E}(z)$. When $\hat{\mathbf{Q}}(z)$ is a filter bank, the loss in paraunitarity can be measured as the reconstruction error [15]

$$\xi = \frac{1}{M} \sum_{\tau} \|\mathbf{E}[\tau]\|_{\text{F}}^2 . \quad (21)$$

Diagonalisation. The goal of the PEVD algorithms is to iteratively approximate a diagonal parahermitian matrix. Therefore, a second performance criterion measures the reduction in off-diagonal energy calculated as

$$E_{\text{norm}}^{(i)} = \frac{\sum_{\tau} \sum_{k=1}^M \|\hat{\mathbf{s}}_k^{(i)}[\tau]\|_2^2}{\sum_{\tau} \|\mathbf{R}[\tau]\|_{\text{F}}^2} , \quad (22)$$

where $\hat{\mathbf{s}}_k^{(i)}[\tau]$ is the modified column vector from (7) containing only off-diagonal elements.

B. Simulation Scenario

For the following results, the PEVD algorithms are run for 100 iterations recording the performance metrics from Sec. IV-A along with the paraunitary matrix order at each iteration. All results apart from those shown in Sec. IV-E have been averaged over an ensemble of 10^3 instantiations. The initial parahermitian matrix, $\mathbf{R}(z)$, for all of the simulations below was generated using the source model described in [9], which

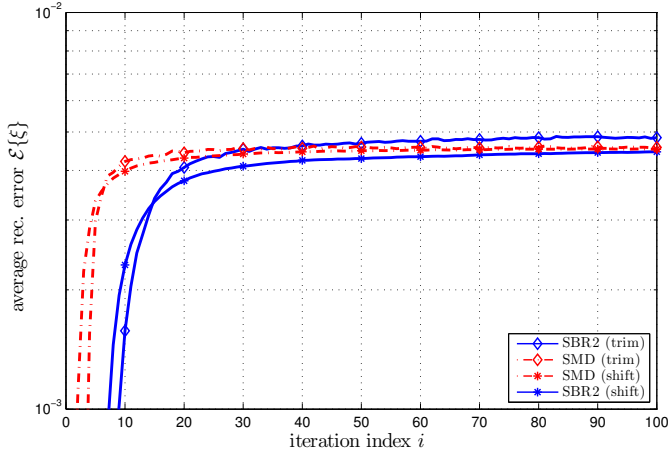


Fig. 1. Ensemble reconstruction error $\mathcal{E}\{\xi\}$ vs. PEVD iterations for the different truncation approaches and PEVD methods.

is randomised to produce a unique $\mathbf{R}(z)$ for each instantiation. The paraunitary matrix produced is $\mathbf{R}(z) \in \mathbb{C}^{6 \times 6}$ for all of the simulations excluding Sec. IV-E where the dimensions are reduced to $\mathbf{R}(z) \in \mathbb{C}^{4 \times 4}$ with the number of lags for all $\mathbf{R}(z)$ set at 47.

C. Reconstruction Error

In [13], one of the major benefits of the row-shift corrected truncation was the drastic reduction in reconstruction error. With the row-shift corrected method it was found that the amount of energy removed from the paraunitary matrix could be increased by a factor of 5 whilst maintaining a similar level of reconstruction error to the lag based truncation. Here for comparison, the truncation parameters across the PEVD methods remain the same with $\mu = 10^{-4}$ and $\mu' = 5\mu$. Fig. 1 shows the reconstruction error for the different PEVD methods, here the SBR2 algorithm using the row-shift corrected method performs the best with an error of 4.5×10^{-4} and SBR2 using the original truncation is worst with an error of 4.8×10^{-4} after 100 iterations. Initially the error curves start very low but they quickly increase as the outer elements become smaller and the truncation algorithms begin to remove their full quota of energy, be it μ or μ' . Even with the compensation of $\mu' = 5\mu$, the row-shift correction method still tends to have a slightly lower error for both PEVD methods despite being permitted to remove five times the energy.

D. Truncated Order and Diagonalisation

As previously shown in [13], the row-shift corrected method has a significant effect on reducing the paraunitary order for the SBR2 method, however with the SMD algorithm the same reduction paraunitary order is not apparent for the selected source model. There is still a slight benefit to using the row-shift corrected truncation but due to the nature of the SMD algorithm there tends to be fewer outliers which need to be corrected by the row-shift truncation.

Fig. 3 shows the diagonalisation measure vs. paraunitary order for the different PEVD algorithms and truncation methods. As with the previous figures these results are over 100

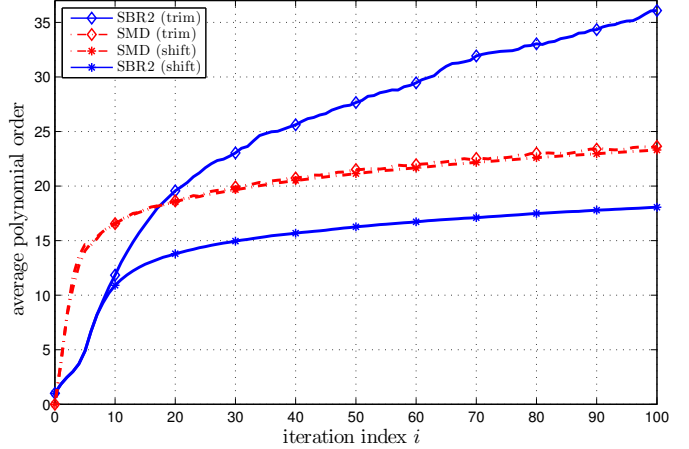


Fig. 2. Average order after truncation of $\hat{\mathbf{Q}}(z)$ vs. PEVD iterations for the different truncation approaches.

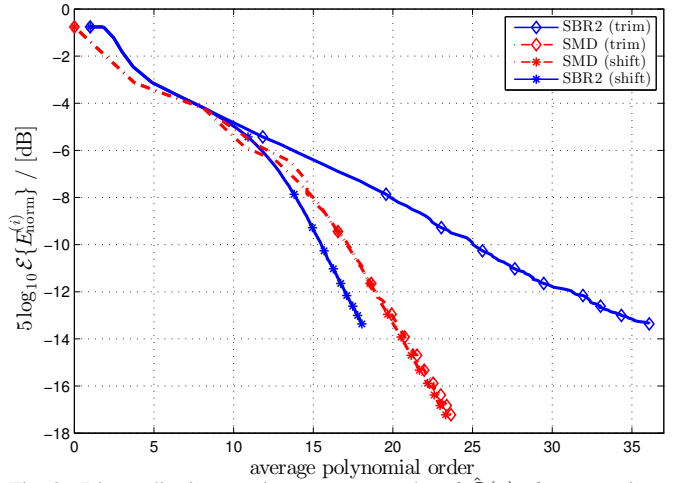


Fig. 3. Diagonalisation metric vs. average order of $\hat{\mathbf{Q}}(z)$ after truncation.

iterations with the diagonalisation recorded after each iteration. In Fig. 3 there is a similar trend to Fig. 2, with the row-shift corrected method SBR2 outperforming the SMD equivalent. Crucially for a set level of diagonalisation, SBR2 with the row-corrected truncation generates a paraunitary matrix of lower cost than SMD, thereby negating one of the benefits of the SMD approach identified in Sec. II and in [9] for this particular source model.

E. Examples of Truncated Paraunitary Matrices

Figs. 4 and 5 illustrate the differences in the paraunitary matrices for both the lag-based truncation method with the row-corrected results overlaid. For clarity of the figures, the initial parahermitian matrix has been reduced to $\mathbf{R}(z) \in \mathbb{C}^{4 \times 4}$ with all other simulation parameters remaining the same. Clearly in the SBR2 paraunitary matrix the row shift corrected approach has more of an effect than it does for SMD. Whereas the maxima in the rows of the SBR2 paraunitary matrix are delayed and spread out with respect to one another, the SMD paraunitary maxima are clustered and delayed by a similar amount. With SMD, the row-shift correction does not aid in reducing the paraunitary matrix length. For the examples

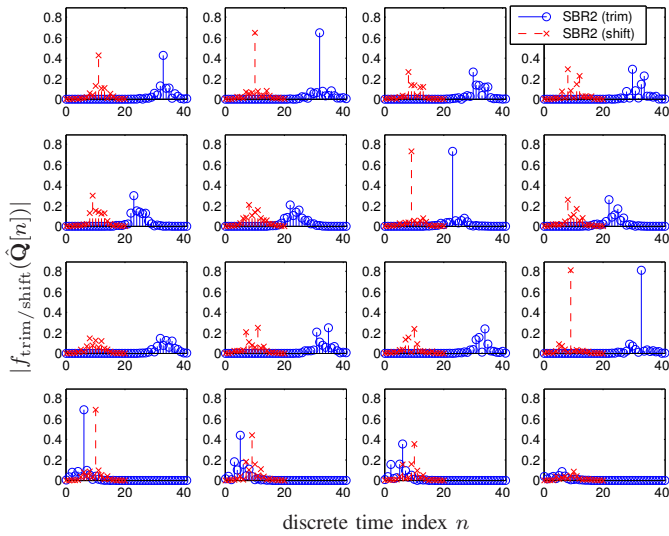


Fig. 4. SBR2 paraunitary matrix truncated with $\mu = 10^{-4}$ using the lag based [12] and row-shift corrected [13] truncation methods.

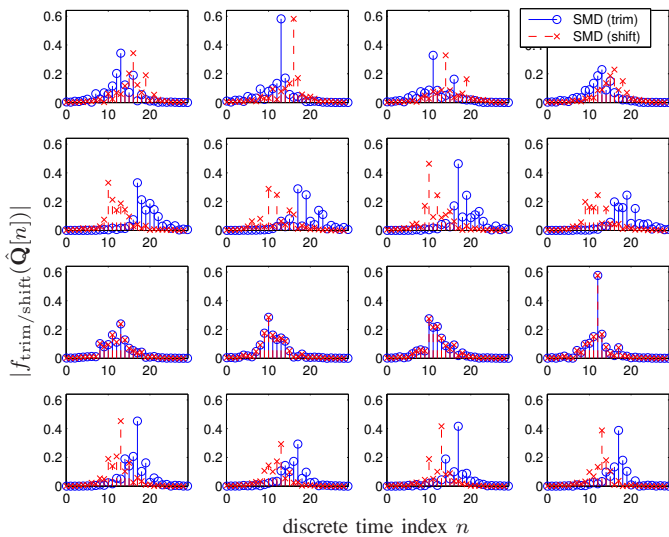


Fig. 5. SMD paraunitary matrix truncated with $\mu = 10^{-4}$ using the lag based [12] and row-shift corrected [13] truncation methods.

shown in Figs. 4 and 5 the reduction in order for the row-corrected method is 21 for the SBR2 paraunitary matrix but only 1 for SMD, and the total row corrected lengths are 21 and 29 respectively.

V. CONCLUSION

This paper has investigated the application of the newly developed row-shift corrected truncation to paraunitary matrices produced by both the SBR2 and SMD PEVD algorithms. Previously the SMD algorithm has been shown to generate paraunitary matrices of lower order than SBR2.

Using the row-shift corrected truncation method the simulation results show only a marginal improvement in both reconstruction error and paraunitary order with SMD. For the SBR2 algorithm we see a larger benefit to using the new truncation approach to the extent that the SBR2 algorithm has now been shown to produce lower order paraunitary matrices

than SMD for a set level of diagonalisation. Based on the results presented here, to minimise the order for an application that extracts the paraunitary matrix, the best approach is to use the SBR2 PEVD method combined with the recently developed row-shift corrected truncation.

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] 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.
- [2] G.H. Golub, C.F. Van Loan. *Matrix Computations*. John Hopkins, 3rd ed., 1996.
- [3] S. Icart, P. Comon. Some Properties of Laurent Polynomial Matrices. In *Conf. Math. Signal Proc.*, Birmingham, UK, Dec. 2012.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] A. Tkacenko. Approximate Eigenvalue Decomposition of Para-Hermitian Systems through Successive FIR Paraunitary Transformations. In *IEEE ICASSP*, pp. 4074–4077, Dallas, TX, Mar. 2010.
- [9] 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.
- [10] 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.
- [11] 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.
- [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] J. Corr, K. Thompson, S. Weiss, I. Proudler, and J. McWhirter. Row-Shift Corrected Truncation of Paraunitary Matrices for PEVD Algorithms. In *European Signal Processing Conference*, Nice, France, September 2015. (submitted).
- [14] A. Jafarian, J. McWhirter. A Novel Method for Multichannel Spectral Factorization. In *EUSIPCO*, pp. 1069–1073, Bucharest, Romania, Aug. 2012.
- [15] P.P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice Hall, 1993.