Reduced search space multiple shift maximum element sequential matrix diagonalisation algorithm

J Corr[*], K Thompson[*], S Weiss[*], I K Proudler[†], J G McWhirter[‡]

[*]*Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, Scotland*
[†]*School of Electronic, Electrical 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

The Multiple Shift Maximum Element Sequential Matrix Diagonalisation (MSME-SMD) algorithm is a powerful but costly approximate iterative polynomial eigenvalue decomposition (PEVD) for space-time covariance-type matrices encountered in e.g. broadband array processing. This paper discusses a newly developed search method that restricts the order growth of the MSME-SMD algorithm. In addition to enhanced control of the polynomial degree of paraunitary and parahermitian factors in this decomposition, the new search method is also computationally less demanding as fewer elements are searched compared to the original while the excellent diagonalisation of MSME-SMD is maintained.

1. Introduction

To accurately model the delay and multipath properties of broadband array processing systems explicit lag elements must be used rather than the phase shifts employed in narrowband systems. Using delays rather than phase shifts forms a space-time covariance matrix, $\mathbf{R}[\tau]$, captures both spatial and temporal dimensions. Taking the z-transform of the space-time covariance matrix, $\boldsymbol{R}(z) \bullet\!\!-\!\!\circ \mathbf{R}[\tau]$, generates the polynomial cross-spectral density (CSD) matrix. The structure of the CSD matrix produced can be seen as matrix with polynomial elements or as a polynomial that has matrices as its coefficients.

The traditional eigenvalue decomposition (EVD) which is used in many narrowband signal processing problems is not directly applicable to the polynomial CSD matrix. The polynomial EVD (PEVD) [1] is used to extend the EVD to the polynomial matrix case. The CSD matrix, $\boldsymbol{R}(z)$, exhibits a property known as parahermitian, similar to the Hermitian property of scalar matrices, where $\boldsymbol{R}(z) = \tilde{\boldsymbol{R}}(z)$. In addition to the conjugate symmetry a parahermitian operation, $\{\tilde{\cdot}\}$, also includes a time reversal such that $\tilde{\boldsymbol{R}}(z) = \boldsymbol{R}^{\mathrm{H}}(z^{-1})$. The PEVD can be used to factorise the parahermitian, $\boldsymbol{R}(z)$, into

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

where $\boldsymbol{Q}(z)$ is paraunitary such that $\boldsymbol{Q}(z)\tilde{\boldsymbol{Q}}(z) = \mathbf{I}$ and $\boldsymbol{D}(z)$ is a diagonal polynomial matrix. Although an exact decomposition in (1) cannot be guaranteed [1], [3] suggests the approximation can be accurate for sufficiently high orders of $\boldsymbol{Q}(z)$.

Recently a wide variety of applications for the PEVD have arisen in areas such as broadband angle of arrival estimation [7], filter bank-based channel coding [4], subband coding [6], and the design of broadband precoding and equalisation of MIMO systems [5]. The polynomial subspace decomposition techniques, including [4,5,7], require an accurate PEVD with low order paraunitary matrices to reduce the computational cost of the application.

The decomposition in (1) can be calculated through a variety of different iterative PEVD algorithms [1,8–10,17]. Each of the iterative PEVD methods may produce a different decomposition and each algorithm has its own merits. This paper considers the recently developed sequential matrix diagonalisation (SMD) family of algorithms [9,10]. Compared to other PEVD algorithms, such as the SBR2 algorithm, [1] the SMD methods converge in fewer iterations and produce decompositions exhibiting a greater degree of diagonalisation (thus greater accuracy). The original SMD algorithm [9] has been shown to give good results with the lowest order paraunitary matrices. The results of the multiple shift maximum element (MSME) SMD [10] are better in terms of convergence however the multiple shifts cause the order of the paraunitary matrices to grow faster. The aim of this paper is to restrict the order growth seen in MSME-SMD whilst maintaining a similar level of performance. A further benefit is that computational cost is also reduced due to less data being processed.

Sec. 2. reviews the current state of the art sequential matrix diagonalisation algorithms. Sec. 3. first analyses the worst case polynomial order growth for the SMD and MSME-SMD algorithms then highlights the benefits the restricted search in the MSME-SMD algorithm. Simulation results are presented in Sec. 5. to compare the different PEVD methods and conclusions are given in Sec. 6..

2. Iterative PEVD Algorithms based on Sequential Matrix Diagonalisation

This section first gives an overview of the main steps involved in the SMD family of iterative PEVD algorithms before explor-

ing the SMD and MSME-SMD algorithms in detail.

The SMD family of algorithms has an initialisation step where all off-diagonal energy on the zero lag is transferred onto the diagonal via an EVD of the zero lag,

$$\mathbf{S}^{(0)}[0] = \mathbf{Q}^{(0)\mathrm{H}}\mathbf{R}[0]\mathbf{Q}^{(0)} \quad . \tag{2}$$

The modal matrix $\mathbf{Q}^{(0)}(z)$ is then applied to all lags in the parahermitian matrix,

$$\mathbf{S}^{(0)}(z) = \mathbf{Q}^{(0)\mathrm{H}}\boldsymbol{R}(z)\mathbf{Q}^{(0)} \quad . \tag{3}$$

At each iteration a generic iterative PEVD algorithm consists of three main steps, first a search is carried out to determine which row(s) and column(s) are to be brought onto the zero lag. This search step is algorithm dependent and will be discussed in more detail below. Next the selected row(s) and column(s) are shifted onto the zero lag by means of a paraunitary shift operation,

$$\boldsymbol{S}^{(i)\prime}(z) = \tilde{\boldsymbol{\Lambda}}^{(i)}(z)\boldsymbol{S}^{(i-1)}(z)\boldsymbol{\Lambda}^{(i)}(z) , \quad i = 1\ldots I \quad . \tag{4}$$

Each PEVD iteration is then completed by bringing the off-diagonal energy at lag zero onto the diagonal, which is done by applying the modal matrix of the EVD from the zero lag to all lags of the parahermitian matrix,

$$\boldsymbol{S}^{(i)}(z) = \mathbf{Q}^{(i)\mathrm{H}}\boldsymbol{S}^{(i)\prime}(z)\mathbf{Q}^{(i)} \quad . \tag{5}$$

All SMD algorithms stop when either a set number of iterations I have been carried out or the search step returns a value which is below a predefined threshold. Upon completion the approximately diagonal parahermitian matrix is $\boldsymbol{S}^{(I)}(z)$, and the product of the steps is used to construct the paraunitary matrix for the decomposition i.e.

$$\boldsymbol{Q}(z) = \mathbf{Q}^{(0)}\prod_{i=1}^{I}\mathbf{Q}^{(i)}\boldsymbol{\Lambda}^{(i)}(z) \tag{6}$$

2.1 Sequential Matrix Diagonalisation

To determine which elements are brought onto the zero lag the original SMD algorithm uses a search based on column norms within the parahermitian matrix

$$\{k^{(i)}, \tau^{(i)}\} = \arg\max_{k,\tau}\|\hat{\mathbf{s}}_k^{(i-1)}[\tau]\|_2 , \quad i = 1\ldots I , \tag{7}$$

where $\hat{\mathbf{s}}_k^{(i-1)}[\tau]$ is a modified column vector which contains all elements excluding the on-diagonal entry. Once the column with the largest norm is found it is brought onto the zero lag using the delay matrix

$$\boldsymbol{\Lambda}^{(i)}(z) = \mathrm{diag}\{\underbrace{1 \ldots 1}_{k^{(i)}-1} \; z^{-\tau^{(i)}} \; \underbrace{1 \ldots 1}_{M-k^{(i)}}\} \tag{8}$$

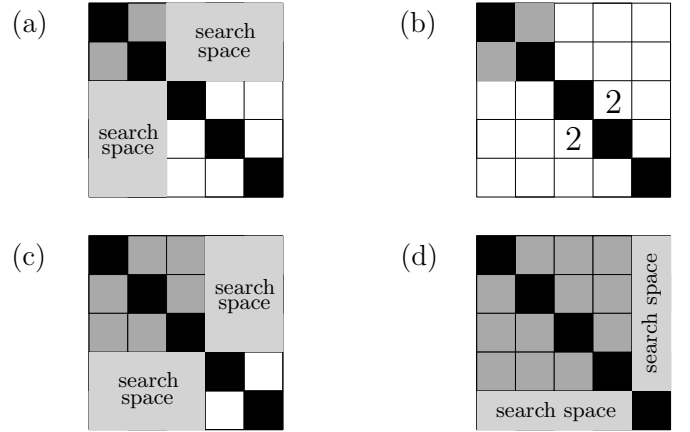where the parameters $k^{(i)}$ and $\tau^{(i)}$ are the column and lag indices obtained in (7).

Figure 1. View of a 5×5 parahermitian matrix during the ith iteration, not showing the lag dimension: starting from the top 2×2 matrix containing the maximum off-diagonal element in (a), (b) shows an example of an element resistant to permutations, the third and fourth stages of the set of reduced search space strategy are shown in (b) and (d).

2.2 Multiple Shift Maximum Element SMD

For the multiple shift maximum element search step (7) is modified to use the l_∞ rather than the l_2 norm,

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

where the symbols have the same meanings as (7). The maximum element search is carried out a total of $M-1$ times during each iteration in an attempt to maximise the amount of energy brought onto the zero lag

To ensure that the full quota of $M-1$ maximum elements are transferred onto the zero lag and that the shifts do not adversely affect one another, a set of reduced search spaces are required. The masks used to reduce the search spaces are shown in Fig. 1 for the case where $M = 5$.

Prior to Fig. 1 (a) the first, global, maximum element is found (without any restrictions) and permuted into the top left 2×2 sub-matrix. Any element found in the search space identified in Fig. 1 (a) can be brought onto the zero lag and permuted into the top left 3×3 sub-matrix with out affecting the initial global maximum. If the second element was found in the position of element 2 in Fig. 1 (b) it would not be possible to permute it into the upper left 3×3 sub-matrix. As a result we would have to proceed to Fig. 1 (d) meaning a total of only 3 elements could be found. Using the MSME-SMD search method, following the search space in Fig. 1 (a) and a permutation to bring the second element into the upper left 3×3 sub-matrix, Fig.1 (c) is obtained. The search, shift and permute process is repeated using the mask in Fig. 1 (c), however this time element 2 from Fig. 1 (b) could be chosen. Finally the fourth or $(M - 1)$-th element is found using the mask in Fig. 1 (d).

In practice the permutations mentioned above are not strictly required, they are only used here to help illustrate the search and mask process, without permutations the search spaces simply become split up and move. After the $(M - 1)$ maximum elements have been located they are then transferred onto the

zero lag using the delay matrix

$$\boldsymbol{\Lambda}^{(i)}(z) = \text{diag}\left\{ z^{-\tau_1^{(i)}} \ z^{-\tau_2^{(i)}} \ \dots \ z^{-\tau_M^{(i)}} \right\} \quad . \qquad (10)$$

3. Polynomial Order Growth

This section analyses the worst case polynomial order growth for both the SMD and MSME-SMD algorithms. It is likely that the actual polynomial matrix growth will be less than the worst case at each iteration. The analysis below assumes we have a parahermitian matrix, $\boldsymbol{S}^{(i)}(z)$, at the i-th iteration with a size of $M \times M \times 2L + 1$, i.e. the maximum lag in either direction is $|L|$.

The growth in order of the parahermitian, $\boldsymbol{S}^{(i)}(z)$, and paraunitary, $\boldsymbol{Q}^{(i)}(z)$, matrices is determined by the order of the shift matrix, $\boldsymbol{\Lambda}^{(i)}(z)$. To help analyse the problem we define the largest possible shift as Δ_{max} and the highest possible order for the shift matrix Λ_{max}.

In the case of the SMD algorithm the growth is simply determined by the magnitude of $\tau^{(i)}$ found in (7) which can have a maximum value of L, therefore $\Delta_{max} = L$. With $\Delta_{max} = L$ the maximum SMD shift matrix length, $\Lambda_{max} = L$. When $\boldsymbol{\Lambda}^{(i)}(z)$ of order Λ_{max} is applied to $\boldsymbol{S}^{(i)}(z)$ and $\boldsymbol{Q}^{(i)}(z)$ their order will increase by $2\Lambda_{max}$ or in this case $2L$. The parameter Λ_{max} is doubled because it is used to advance/delay a column and delay/advance a row onto the zero lag which grows the polynomial order in both directions.

For the MSME-SMD algorithm each of the $(M-1)$ shifts can potentially interact such that the maximum shift length, Δ_{max} is $\lceil ((M-1)L)/2 \rceil$. The multiple shift algorithm can both delay and advance elements onto the zerolag in a single iteration using $\boldsymbol{\Lambda}^{(i)}(z)$ therefore the maximum shift matrix order, Λ_{max}, is $2\lceil ((M-1)L)/2 \rceil$. As with SMD, when $\boldsymbol{\Lambda}^{(i)}(z)$ of order Λ_{max} is applied to the polynomial matrices their order will increase by $2\Lambda_{max}$. For the MSME-SMD algorithm the worst case polynomial order growth is $4\lceil ((M-1)L)/2 \rceil$. Even with reasonably small values of M the multiple shift algorithm can result in a significant increase in the worst case polynomial order growth.

The growth in polynomial order can be curtailed using appropriate parahermitian [11] and paraunitary [12,13] trim functions. Both methods are permitted to remove up to a predefined threshold of energy, μ, from the outer lags of the polynomial matrices. For a parahermitian matrix the trim is done symmetrically taking advantage of its parahermitian nature. The parahermitian property is also preserved by the trim function. In the case of paraunitary matrices the trim function is applied to both sides of $\boldsymbol{Q}(z)$ asymmetrically because the outer lags of a paraunitary matrix will have different energies. The paraunitary property is replaced by near-paraunitarity after the trim function is applied but the extent of this can be minimised by using the row-shift corrected trim from [13].

To reduce computational costs of the PEVD algorithms the parahermitian truncation can be carried out at the end of every iteration, with the resulting maximum total loss in energy after I iterations $I \times \mu_{PH}$. As the paraunitary matrix is only ever calculated when the PEVD is complete the trim function is only

Table 1. Summary of worst case polynomial order growth for the different SMD variants .

	SMD	MSME	RS-MSME
Δ_{max}	L	$\lceil ((M-1)L)/2 \rceil$	L
Λ_{max}	L	$2\lceil ((M-1)L)/2 \rceil$	$2L$
Ord. Growth	$2L$	$4\lceil ((M-1)L)/2 \rceil$	4L

applied once and so the resulting energy loss has a maximum of μ_{PU}.

4. Restricted Search MSME-SMD

In the restricted search MSME-SMD we impose an extra condition on the search spaces in Fig. 1 to control the polynomial order growth in $\boldsymbol{S}^{(i)}(z)$ and $\boldsymbol{Q}^{(i)}(z)$. Rather than allowing every search to select elements from any lag, we restrict it to elements closer to the zero lag than the global maximum, found during the first search of each iteration. The new approach still uses (9) but now once the first search of the i-th iteration finds a maximum element on $\tau^{(i)}$, the lag parameter, τ, in (9) is restricted such that $|\tau| \leq |\tau^{(i)}|$ for the remaining searches in the i-th iteration. Using this method the worst case maximum shift, Δ_{max}, is L, the maximum order for the shift matrix, Λ_{max}, is $2L$ and the polynomial order growth is $4L$. For comparison the maximum shift, shift matrix order and polynomial order growth are summarised in Tab. 1 for all three SMD variants. The worst case scenario sees the RS-MSME-SMD order grow twice as fast as SMD but this is significantly lower than the original MSME-SMD, especially when the matrix width M is increased.

Ultimately limiting the search space to lower lags will result in missing some elements and slow the algorithm slightly but these missed elements are likely to be found by searches during future iterations. The reduced search space will benefit the real time performance in two ways; first the searches during one iteration after the restriction will be on fewer elements and second the slower growth in parahermitian matrix means future searches will be over fewer elements, both result in better real time performance.

5. Results

To illustrate the performance of the different PEVD algorithms we first present the performance metrics, followed by the simulation set up. Finally the results are presented and the performance of the PEVD algorithms is analysed.

5.1 Performance Metrics

To confirm that the RS-MSME-SMD maintains a similarly high performance as the original MSME-SMD, the first test will measure diagonalisation the remaining off-diagonal energy after i iterations normalised by the energy in the initial parahermitian matrix, $\mathbf{R}[\tau]$,

$$E_{\text{norm}}^{(i)} = \frac{\sum_\tau \sum_{k=1}^M \|\hat{\mathbf{s}}_k^{(i)}[\tau]\|_2^2}{\sum_\tau \|\mathbf{R}[\tau]\|_{\text{F}}^2} \quad , \qquad (11)$$
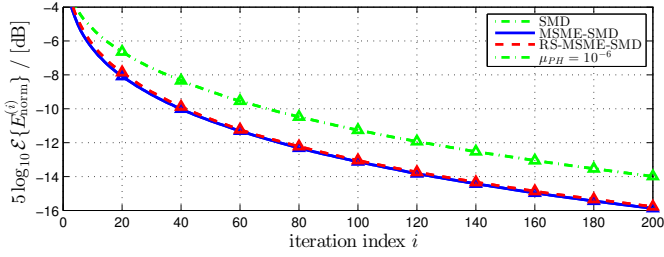
Figure 2. Diagonalisation vs. algorithm iterations for the SMD algorithm and the two MSME-SMD varients.

where $\|\cdot\|_F$ represents the Frobenius norm.

The main objective of the search space restriction discussed in this paper is to limit the order growth in the polynomial matrix. With this in mind the order of the parahermitian and paraunitary matrices are recorded after each iteration. The orders are recorded both when the trim function is only used to remove zeros and when it is permitted to remove 10^{-6} of the initial polynomial matrix energy.

A side effect of the reduction in parahermitian matrix order is a reduction in the computational cost of calculating the PEVD. Here we use execution time as a measure of the computational complexity of the PEVD algorithms implemented in Matlab 2014a with the following system specification: Ubuntu 14.04 on a workstation with Intel® Xeon® E5-1607V2 3.00 GHz x 4 cores and 8 GB RAM.

5.2 Simulation Set Up

The results were produced using the source model from [16] used to produce an ensemble of 10^2 parahermitian matrices which were not majorised with an average dynamic range of approximately 30 dB. The source model is randomised so that the parahermitian matrices produced are unique for each instantiation. The parahermitian matrix, $\boldsymbol{R}(z)$, is $\boldsymbol{R}(z) \in \mathbb{C}^{6 \times 6}$ with the initial number of lags set to 119. Each of the PEVD algorithms were run for 200 iterations with the performance metrics recorded after each iteration. The simulations are first run using $\mu_{PH} = \mu_{PU} = 0$, i.e. only removing zero filled lags, then repeated over the same ensemble for $\mu_{PH} = \mu_{PU} = 10^{-6}$.

5.3 Algorithm Convergence

Fig. 2 shows the reduction in off-diagonal energy vs. algorithm iterations for the SMD algorithm and the two versions of MSME-SMD. Despite the reduced search space we can see for the example in Fig. 2 the both MSME algorithms follow an almost identical convergence, both of which are significantly better than the SMD method.

As discussed in Sec. 4. the original MSME-SMD transfers marginally more energy per iteration than the new RS-MSME-SMD algorithm in Fig. 2 however these are still significantly better than convergence of the SMD approach.
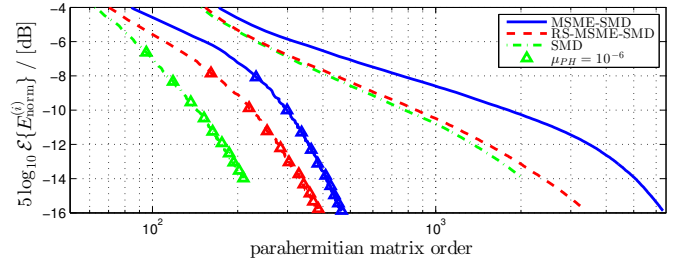

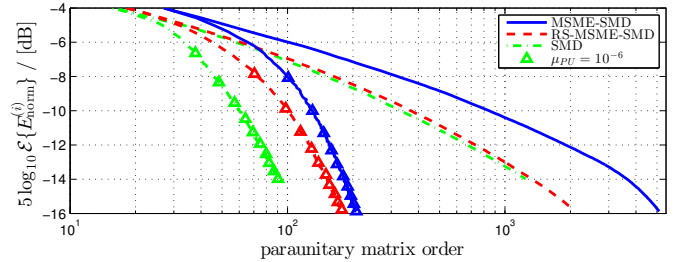**Figure 3.** Reduction in off-diagonal energy vs. growth in parahermitian matrix order.

Figure 4. Reduction in off-diagonal energy vs. growth in paraunitary matrix order.

5.4 Paraunitary/Parahermitian Matrix Order

This section investigates one of the main goals of the RS-MSME-SMD algorithm which is to reduce the growth in polynomial order of the parahermitian and paraunitary matrices.

Figs. 3 & 4 show the order growth of the parahermitian and paraunitary matrices for each of the selected PEVD algorithms. Generally the SMD method produces parahermitian and paraunitary matrices of lower order than any of the other PEVD methods [17]. Here we can see in both Fig. 3 & 4 that the matrices produced by RS-MSME-SMD are significantly shorter than their MSME-SMD equivalent and are a similar level to SMD. Even when a truncation algorithm such as those described in [11] and [13] are applied to the parahermitian and paraunitary matrices the reduced search method still out performs the original MSME-SMD as shown in Fig. 3 & 4 although it does loose out slightly to SMD.

5.5 Real Time Execution

Fig. 5 shows the time taken for each of these algorithms to carry out 200 iterations alongside the diagonalisation measure at each point. Despite their more complex search methods the MSME-SMD algorithms actually converge faster in real time than the SMD approach in all cases. When no truncation is used we can see that the new reduced search MSME method is more efficient than the original MSME search, in fact the new method takes on average around 10 seconds less than its predecessor to complete 200 iterations. When the parahermitain truncation methods are included both MSME-SMD variants obtain a significant performance improvement, where as the same change in SMD has a lesser affect. The performance benefits of the reduced search MSME-SMD are not as obvious
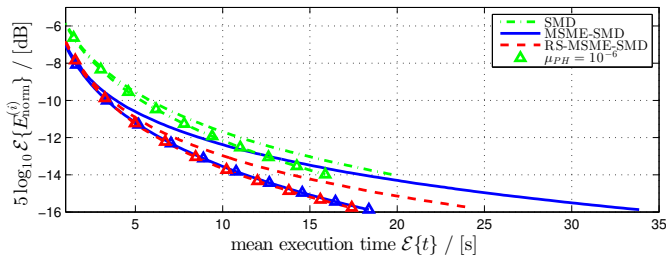
Figure 5. real time convergence of PEVD algorithms, diagonalisation measure vs. mean execution time.

when the parahermitian truncation is used but it still performs better than the original MSME-SMD.

6. Conclusion

Through analysis of the polynomial order growth of the SMD and MSME-SMD algorithms we have proposed a new search method which can significantly reduce the polynomial order growth of the MSME-SMD algorithm. Results indicate almost no degradation in energy transfer between the existing and proposed method. In addition the experiments have shown that the new method has a significant reduction in polynomial matrix order growth even when truncation methods are used. The reduced search spaces and resulting lower order parahermitian matrices also result in an improved real time convergence. When the parahermitian and paraunitary matrices are truncated the benefits of the new search method are reduced. In general the restriction of the search space slows the growth of both paraunitary and parahermitian matrices which leads to a faster execution time with minimal impact on the algorithm convergence.

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 *EUSIPCO*, Nice, France, September 2015.

[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.

[16] J. Corr, K. Thompson, S. Weiss, I. Proudler, and J. McWhirter. Impact of source model matrix conditioning on iterative PEVD algorithms In *IET Inteligent Signal Processing Conference*, London, England, December 2015. Submitted.

[17] J. Corr, K. Thompson, S. Weiss, I. Proudler, and J. McWhirter. Causality-Constrained Multiple Shift Sequential Matrix Diagonalisation for Parahermitian Matrices In *EUSIPCO*, Lisbon, Portugal, September 2014.