Analysing the Performance of Divide-and-Conquer Sequential Matrix Diagonalisation for Large Broadband Sensor Arrays

Fraser K. Coutts*, Keith Thompson*, Stephan Weiss*, Ian K. Proudler*,†
* Department of Electronic & Electrical Engineering, University of Strathclyde, Glasgow, Scotland
† School of Electrical, Electronics & Systems Engineering, Loughborough Univ., Loughborough, UK
{fraser.coutts,keith.thompson,stephan.weiss,ian.proudler}@strath.ac.uk

***Abstract*—A number of algorithms capable of iteratively calculating a polynomial matrix eigenvalue decomposition (PEVD) have been introduced. The PEVD is an extension of the ordinary EVD to polynomial matrices and will diagonalise a parahermitian matrix using paraunitary operations. Inspired by recent work towards a low complexity divide-and-conquer PEVD algorithm, this paper analyses the performance of this algorithm — named divide-and-conquer sequential matrix diagonalisation (DC-SMD) — for applications involving broadband sensor arrays of various dimensionalities. We demonstrate that by using the DC-SMD algorithm instead of a traditional alternative, PEVD complexity and execution time can be significantly reduced. This reduction is shown to be especially impactful for broadband multichannel problems involving large arrays.**

I. Introduction

Polynomial matrix representations can be used to express broadband multichannel problems. Such formulations can be used in a number of areas, including broadband MIMO precoding and equalisation [1], polyphase analysis and synthesis matrices for filter banks [2], broadband beamforming [3], [4], and broadband angle of arrival estimation [5], [6]. Typically, these problems involve parahermitian polynomial matrices, which are identical to their parahermitian conjugate, i.e., $\boldsymbol{R}(z) = \tilde{\boldsymbol{R}}(z) = \boldsymbol{R}^{\mathrm{H}}(1/z^*)$ [2]. This matrix $\boldsymbol{R}(z)$ can arise as the z-transform of a space-time covariance matrix $\mathbf{R}[\tau]$,

$$\boldsymbol{R}(z) = \textstyle\sum_\tau \mathbf{R}[\tau] z^{-\tau} \ , \tag{1}$$

where $\boldsymbol{R}(z)$ is a cross power spectral density (CSD) matrix,

$$\mathbf{R}[\tau] = \mathcal{E}\{\mathbf{x}[n]\mathbf{x}^{\mathrm{H}}[n-\tau]\} \ , \tag{2}$$

and $\mathbf{x}[n] \in \mathbb{C}^M$ is a data vector collected by an M-element broadband array. Here, $\mathcal{E}\{\cdot\}$ denotes the expectation operator.

As an extension of the eigenvalue decomposition to parahermitian matrices, a polynomial matrix eigenvalue decomposition (PEVD) has been defined in [7], [8]. The PEVD uses a finite impulse response (FIR) paraunitary matrix [9] $\boldsymbol{F}(z)$ to approximately diagonalise and spectrally majorise [10] a cross power spectral density matrix $\boldsymbol{R}(z)$ such that

$$\boldsymbol{D}(z) \approx \boldsymbol{F}(z)\boldsymbol{R}(z)\tilde{\boldsymbol{F}}(z) \ , \tag{3}$$

where $\boldsymbol{D}(z) = \mathrm{diag}\{\boldsymbol{D}_1(z) \ \ \boldsymbol{D}_2(z) \ \ \ldots \ \ \boldsymbol{D}_M(z)\}$ is diagonalised and spectrally majorised with PSDs $\boldsymbol{D}_{i+1}(e^{j\Omega}) \geq \boldsymbol{D}_i(e^{j\Omega}) \ \forall \ \Omega, \ i = 1 \ldots (M-1)$, with $\boldsymbol{D}_i(e^{j\Omega}) = \boldsymbol{D}_i(z)|_{z=e^{j\Omega}}$. The diagonal of $\boldsymbol{D}(z)$ contains polynomial eigenvalues, and the rows of $\boldsymbol{F}(z)$ are polynomial eigenvectors. Equation (3) has only approximate equality, as the PEVD of a finite order polynomial matrix is generally not of finite order. The paraunitary matrix $\boldsymbol{F}(z)$ is important for broadband signal processing applications such as MIMO [1] or beamforming [3], [4], which rely on accurate but numerically inexpensive subspace decompositions.

Existing PEVD algorithms include sequential matrix diagonalisation (SMD) [11], second-order sequential best rotation (SBR2) [8], and various evolutions of the algorithm families [12]–[14]. Each of these algorithms uses an iterative approach to approximately diagonalise a parahermitian matrix. For matrices of high dimensionality, these algorithms can be computationally costly to compute; therefore, any cost savings will be advantageous for applications.

In an effort to reduce the cost of PEVD algorithms, previous work in [8], [15]–[18] has focussed on the trimming of polynomial matrices to curb growth in order, as such growth translates directly into an increase in computational complexity and memory storage requirements. Recently, techniques in [19], [20] have successfully reduced the complexity of existing PEVD algorithms through the removal of algorithmic redundancy.

Inspired by research in [21]–[23], which demonstrates that complexity reduction can be obtained by using a divide-and-conquer approach to eigenproblems, work in [24] describes a divide-and-conquer approach for the PEVD. This algorithm — titled divide-and-conquer sequential matrix diagonalisation (DC-SMD) — can be utilised to reduce algorithm complexity with minimal loss in accuracy, and has a framework based on the SMD algorithm.

Here, we investigate the performance increase DC-SMD offers over the existing sequential matrix diagonalisation algorithm [11] for the decomposition of parahermitian matrices with varying spatial dimension. Such matrices are generated when computing the space-time covariance matrix according to (2) for data from a broadband sensor array with M elements; in this scenario, $\mathbf{R}[\tau] \in \mathbb{C}^{M\times M}$. By testing the performance of DC-SMD for various M, we can therefore establish its ability to process data from various sizes of broadband sensor array. Performance is measured as the

cumulative complexity — in terms of multiply-accumulate (MAC) operations — and algorithm execution time required to decompose matrix $\boldsymbol{R}(z)$.

In [24], it was demonstrated that DC-SMD generally produces paraunitary filters of greater order than SMD for a similar level of performance. Work in [15], [16] has shown that by employing a row-shift truncation (RST) scheme for paraunitary matrices, filter order can be reduced. We investigate the utilisation of this approach alongside DC-SMD to test if similar paraunitary matrix order reductions are possible.

Below, Sec. II will provide a brief overview over the DC-SMD algorithm. A row-shift truncation method to reduce the order of paraunitary filters generated by DC-SMD is outlined in Sec. III. Simulation results comparing the performance of DC-SMD to SMD for various scenarios are presented in Sec. IV, with conclusions drawn in Sec. V.

II. Divide-and-Conquer Sequential Matrix Diagonalisation

This section outlines the components of the divide-and-conquer sequential matrix diagonalisation (DC-SMD) PEVD algorithm [24]. Following an overview of DC-SMD in Sec. II-A, Sec. II-B and Sec. II-C explain the key stages of this algorithm by detailing the divide and conquer steps, respectively. The complexity requirements of this algorithm are derived in Sec. II-D.

A. Divide-and-Conquer Sequential Matrix Diagonalisation

The DC-SMD algorithm approximates the PEVD using a series of elementary paraunitary operations to iteratively diagonalise a parahermitian matrix $\boldsymbol{R}(z) \in \mathbb{C}^{M\times M}$ and its associated coefficient matrix, $\mathbf{R}[\tau]$. Similarly to other PEVD algorithms, DC-SMD generates an output diagonal matrix $\boldsymbol{D}(z)$ containing eigenvalues, and a paraunitary matrix $\boldsymbol{F}(z)$ containing eigenvectors, such that (3) is satisfied.

While traditional PEVD algorithms — such as SMD [11] — attempt to diagonalise an entire $M \times M$ parahermitian matrix at once, the DC-SMD algorithm first divides the matrix into a number of smaller, independent parahermitian matrices, before diagonalising — or conquering — each matrix separately. For example, a matrix $\boldsymbol{R}(z) \in \mathbb{C}^{20\times 20}$ might be brought into block-diagonal form comprising of four 5×5 parahermitian matrices, each of which can be diagonalised independently. Fig. 1 shows the state of the parahermitian matrix at each stage of the process for this example.

If matrix $\boldsymbol{R}(z)$ is of spatial dimension greater than $\hat{M} \times \hat{M}$ — where $\hat{M}$ is an arbitrary user-defined value — an algorithm named sequential matrix segmentation (SMS) [24] is used to recursively divide the matrix into multiple independent parahermitian matrices. Each parahermitian matrix is then diagonalised in sequence through the use of the SMD algorithm. If $M \leq \hat{M}$, the divide step is skipped, and the input matrix is processed via SMD. To reduce the order of the paraunitary matrix prior to implementation, $\boldsymbol{F}(z)$ is truncated using a parameter μ; this process is described in Sec. III-A.

The individual steps of DC-SMD are summarised in more detail in [24].

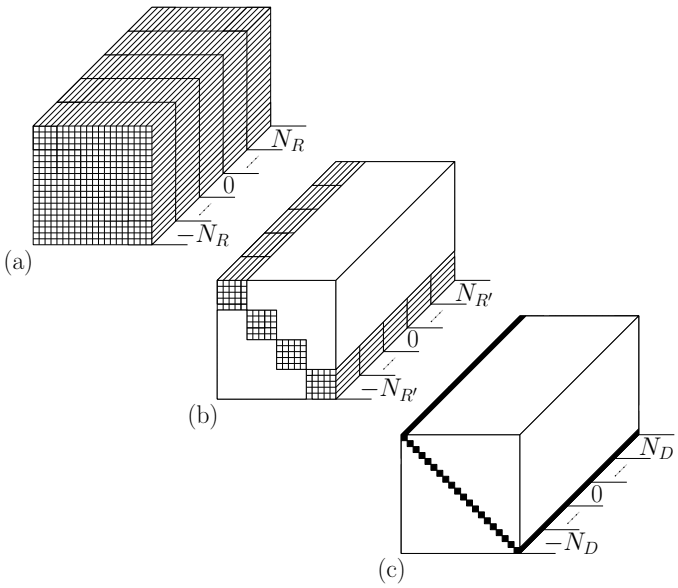

Fig. 1. (a) Original matrix $\mathbf{R}[\tau] \in \mathbb{C}^{20\times 20}$, (b) segmented result $\mathbf{R}'[\tau]$, and (c) diagonalised output $\mathbf{D}[\tau]$. N_R, $N_{R'}$, and N_D are the maximum lags for matrices $\mathbf{R}[\tau]$, $\mathbf{R}'[\tau]$, and $\mathbf{D}[\tau]$, respectively.

B. Recursive Polynomial Matrix Segmentation

If $\boldsymbol{R}(z)$ has spatial dimension $M > \hat{M}$, the divide stage of DC-SMD comes into effect. This stage recursively applies sequential matrix segmentation (SMS) [24] to divide $\boldsymbol{R}(z)$ into multiple independent parahermitian matrices. SMS is a novel variant of SMD designed to segment an input matrix $\hat{\boldsymbol{R}}(z) \in \mathbb{C}^{M'\times M'}$ into two independent parahermitian matrices $\hat{\boldsymbol{R}}_{11}(z) \in \mathbb{C}^{(M'-P)\times(M'-P)}$ and $\hat{\boldsymbol{R}}_{22}(z) \in \mathbb{C}^{P\times P}$, and two matrices $\hat{\boldsymbol{R}}_{12}(z) \in \mathbb{C}^{(M'-P)\times P}$ and $\hat{\boldsymbol{R}}_{21}(z) \in \mathbb{C}^{P\times(M'-P)}$, where $\hat{\boldsymbol{R}}_{12}(z) = \tilde{\hat{\boldsymbol{R}}}_{21}(z)$ are approximately zero.

The divide step of DC-SMD operates recursively. In the first recursion, the matrix $\hat{\boldsymbol{R}}(z)$ input to SMS is equal to $\boldsymbol{R}(z)$ and $M' = M$. Output matrix $\hat{\boldsymbol{R}}_{22}(z)$ is stored and subsequently diagonalised during the conquer step. If the second output matrix $\hat{\boldsymbol{R}}_{11}(z)$ is of spatial dimension greater than $\hat{M} \times \hat{M}$, the second recursion of the divide step uses $\hat{\boldsymbol{R}}_{11}(z)$ as the input to SMS, and M' is set equal to $M - P$. Recursions continue in this fashion until $(M' - P) \leq \hat{M}$. The dimensions of the smaller matrix produced during division, P, is forced to satisfy $P \leq \hat{M}$.

SMS iteratively minimises the energy in select regions of a parahermitian matrix in an attempt to segment the matrix. Fig. 2 illustrates the segmentation process for $M' = 5$ and $P = 2$.

The SMS algorithm continues operating until I_D iterations have been executed, or when the energy in the targeted regions, $E(\hat{\boldsymbol{R}}_{12}(z)) + E(\hat{\boldsymbol{R}}_{21}(z))$, falls below a threshold $2\delta E(\hat{\boldsymbol{R}}(z))$. Here, δ is some arbitrary value, and $E(\cdot)$ computes the energy in a polynomial matrix according to

$$E(\hat{\boldsymbol{R}}(z)) = \textstyle\sum_\tau \|\hat{\mathbf{R}}[\tau]\|_{\mathrm{F}}^2 \,, \tag{4}$$

where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm. A parameter μ is used to

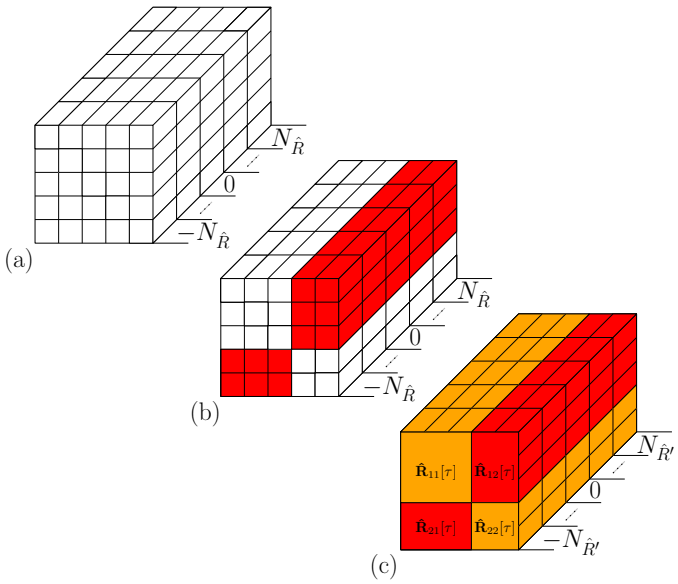

Fig. 2. (a) Original matrix $\hat{\mathbf{R}}[\tau] \in \mathbb{C}^{5\times 5}$, (b) regions (red) to be iteratively driven to zero in SMS for $P = 2$, and (c) segmented result. $N_{\hat{R}}$ and $N_{\hat{R}'}$ are the maximum lags for the original and segmented matrices, respectively.

truncate the parahermitian and paraunitary matrices generated at each iteration of SMS. More detail on the implementation of this truncation can be found in [17], [24].

C. Independent Conquering of Divided Polynomial Matrices

At this stage of DC-SMD, $\boldsymbol{R}(z)$ has been segmented into multiple independent parahermitian matrices. Each matrix can now be diagonalised individually through the use of the SMD PEVD algorithm [11]. Each instance of SMD is provided with a parameter I_C — which defines the maximum possible number of algorithm iterations — and a truncation parameter μ.

D. Algorithm Complexity

A matrix multiplication step dominates the complexity of the SMS and SMD functions internal to DC-SMD [19]. In this step, which occurs at every iteration i of both algorithms, every matrix-valued coefficient in a parahermitian matrix of length $L^{(i)}$ must be left- and right-multiplied with a unitary matrix. Note that $L^{(i)}$ is not known in advance, and only emerges during an iteration. Accounting for a multiplication of 2 $M \times M$ matrices by M^3 MACs, a total of $2L^{(i)}M^3$ MACs arise to generate the updated parahermitian matrix in each algorithm.

In DC-SMD, one instance of the SMS algorithm has a maximum cumulative complexity of $\sum_{i=1}^{I_D} 2L^{(i)}M_\alpha^3$, and SMD has a similar maximum of $\sum_{i=1}^{I_C} 2L^{(i)}M_\gamma^3$, where M_α and M_γ are the dimensions of the matrices input to each algorithm, respectively. The total cumulative complexity of DC-SMD can be approximated by summing the cumulative complexities of each instance of the SMS and SMD algorithms.

III. Row-Shift Truncation Approach

In [24], it was found that DC-SMD generally produces paraunitary filters of greater order than SMD for a similar level of performance. Work in [15], [16] has shown that by employing a row-shift truncation (RST) scheme for paraunitary matrices, filter order can be reduced with little loss to paraunitarity, such that $\boldsymbol{F}(z)\tilde{\boldsymbol{F}}(z) \approx \mathbf{I}_M$ — where $\mathbf{I}_M$ is an $M \times M$ identity matrix. By employing this approach alongside DC-SMD, similar paraunitary matrix order reductions should be possible.

A. Traditional Truncation Method

The paraunitary matrix truncation method from [18] is employed within DC-SMD. This approach reduces the order of the paraunitary matrix $\boldsymbol{F}(z)$ by removing the N_1 leading and N_2 trailing lags using a trim function

$$f_{\text{trim}}(\mathbf{F}[n]) = \begin{cases} \mathbf{F}[n+N_1], & 0 \leq n < N - N_2 - N_1 \\ \mathbf{0}, & \text{otherwise} \end{cases} . \tag{5}$$

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

$$\begin{aligned} \gamma_{\text{trim}} &= 1 - \frac{\sum_n \|f_{\text{trim}}(\mathbf{F}[n])\|_{\text{F}}^2}{\sum_n \|\mathbf{F}[n]\|_{\text{F}}^2} \\ &= 1 - \frac{1}{M}\sum_n \|f_{\text{trim}}(\mathbf{F}[n])\|_{\text{F}}^2 . \end{aligned} \tag{6}$$

A parameter μ is used to provide an upper bound for γ_{trim}. Given the above, the truncation procedure can be expressed as the constrained optimisation problem:

$$\text{maximise } (N_1 + N_2) , \quad \text{s.t.} \quad \gamma_{\text{trim}} \leq \mu . \tag{7}$$

This is implemented by removing the outermost matrix coefficients of matrix $\boldsymbol{F}(z)$ until γ_{trim} approaches μ from above.

B. Row-Shift Truncation Method

The row-shift truncation method [15], [16] exploits the ambiguity in paraunitary matrices [15], [25]. This arises as a generalisation of a phase ambiguity inherent to eigenvectors from a standard EVD [26], which in the polynomial case extends to arbitrary phase responses or all-pass filters. The simplest manifestation of such filters can form an integer number of unit delays. Therefore, following completion of the DC-SMD algorithm, this ambiguity permits $\boldsymbol{F}(z)$ to be replaced by $\hat{\boldsymbol{F}}(z)$, where $\hat{\boldsymbol{F}}(z) = \boldsymbol{\Gamma}(z)\boldsymbol{F}(z)$. From [15], $\boldsymbol{\Gamma}(z)$ must take the form

$$\boldsymbol{\Gamma}(z) = \text{diag}\{z^{-\tau_1}\ z^{-\tau_2}\ \dots\ z^{-\tau_M}\} . \tag{8}$$

The delay matrix $\boldsymbol{\Gamma}(z)$ therefore has the effect of shifting the mth row of the paraunitary matrix $\boldsymbol{F}(z)$ by τ_m. These row shifts can be used to align the maximum values in each row of $\boldsymbol{F}(z)$ such that $\hat{\boldsymbol{F}}(z)$ can be truncated more effectively.

Paraunitary matrix $\hat{\boldsymbol{F}}(z)$ can be subdivided into its M row vectors $\hat{\mathbf{f}}_m(z)$, $m = 1 \dots M$,

$$\hat{\boldsymbol{F}}(z) = \begin{bmatrix} \hat{\mathbf{f}}_1(z) \\ \vdots \\ \hat{\mathbf{f}}_M(z) \end{bmatrix} . \tag{9}$$

Each row is then truncated individually according to

$$f_{\text{shift}}(\hat{\mathbf{f}}_m[n]) = \begin{cases} \hat{\mathbf{f}}_m[n+N_{1,m}], & 0 \leq n < T_m \\ \mathbf{0}, & \text{otherwise} \end{cases} , \quad (10)$$

where the length of row m becomes $T_m = N - N_{2,m} - N_{1,m}$. The row shifts, τ_m, in (8) are then set equal to $N_{1,m}\ \forall\ m = 1 \dots M$.

As each row has unit energy, the proportion of energy to be removed is given by

$$\gamma_{\text{shift},m} = 1 - \sum_n \| f_{\text{shift}}(\hat{\mathbf{f}}_m[n]) \|_2^2 \, . \quad (11)$$

As with the traditional truncation method, a constrained optimisation problem is obtained:

$$\begin{aligned} &\text{maximise } (N_{1,m} + N_{2,m}) \, , \\ &\quad \text{s.t.} \quad \gamma_{\text{shift},m} \leq \mu_{\text{RST}} \ \forall\ m = 1 \dots M \, . \end{aligned} \quad (12)$$

The maximum possible proportion of energy removed from each row is limited by μ_{RST}. Following row-shift truncation, each row has length T_m, and the length of the paraunitary matrix is $\max\limits_{m=1\dots M} \{T_m\}$.

IV. Results

To benchmark the proposed approach, this section first defines the performance metrics for evaluating the SMD and DC-SMD algorithms before setting out a simulation scenario, over which an ensemble of simulations will be performed.

A. Performance Metrics

Since SMD and DC-SMD both iteratively minimise off-diagonal energy, a suitable metric E_{norm}, defined in [11], is used; this metric divides the off-diagonal energy at each iteration of each algorithm by the total energy. During computation of E_{norm}, squared covariance terms are used; therefore a logarithmic notation of $5 \log_{10} E_{\text{norm}}$ is employed.

Metrics $\mathcal{E}\{C_{(.),-10\,\text{dB},M}\}$ and $\mathcal{E}\{t_{(.),-10\,\text{dB},M}\}$ represent the ensemble-averaged cumulative complexity and execution time required for a PEVD algorithm to achieve a diagonalisation of $5 \log_{10} E_{\text{norm}} = -10\,\text{dB}$ for spatial dimension M.

When truncation is employed, the eigenvectors and eigenvalues output from PEVD algorithms are only able to approximately reconstruct the input matrix. DC-SMD also introduces a segmentation error in its divide step, due to imperfect segmentation in SMS, which is higher for a larger threshold δ. A metric capable of measuring the difference between the original and reconstructed matrices is the mean squared error

$$\text{MSE} = \tfrac{1}{M^2 L'} \textstyle\sum_\tau \| \mathbf{E}_R[\tau] \|_{\text{F}}^2 \, , \quad (13)$$

where $\mathbf{E}_R[\tau] = \bar{\mathbf{R}}[\tau] - \mathbf{R}[\tau]\ \forall\ \tau$, $\bar{\boldsymbol{R}}(z) = \tilde{\boldsymbol{F}}(z)\boldsymbol{D}(z)\boldsymbol{F}(z)$, L' is the length of $\boldsymbol{E}_R(z)$, and $\boldsymbol{F}(z)$ and $\boldsymbol{D}(z)$ are obtained from SMD or DC-SMD.

The contents of Sec. II-D allow approximate measurements of cumulative complexity to be made at each iteration of both algorithms. The output paraunitary matrix $\boldsymbol{F}(z)$ can be used in broadband signal processing applications such as MIMO [1] or beamforming [3], [4]. A useful metric for gauging the implementation cost of $\boldsymbol{F}(z)$ is its length.

B. Simulation Scenario

The simulations below have been performed over an ensemble of 10^2 instantiations of $\boldsymbol{R}(z) \in \mathbb{C}^{M \times M}$, $M \in \{10;\ 20;\ 30;\ 40;\ 50;\ 60;\ 70\}$, based on the randomised source model in [11]. This source model generates $\boldsymbol{R}(z) = \tilde{\boldsymbol{U}}(z)\boldsymbol{W}(z)\boldsymbol{U}(z)$, whereby the diagonal $\boldsymbol{W}(z) \in \mathbb{C}^{M \times M}$ contains the power spectral densities (PSDs) of $M/2$ independent sources. These sources are spectrally shaped by innovation filters such that $\boldsymbol{W}(z)$ has an order of 120, and limits the dynamic range of the PSDs to about 30 dB. Random paraunitary matrices $\boldsymbol{U}(z) \in \mathbb{C}^{M \times M}$ of order 60 perform a convolutive mixing of these sources, such that $\boldsymbol{R}(z)$ has an order of 240.

During iterations, truncation parameters of $\mu = 10^{-6}$ and $\mu_{\text{RST}} \in \{10^{-6};\ 10^{-9};\ 10^{-12}\}$ were used. The standard SMD implementation was run until an input matrix was sufficiently diagonalised, such that the off-diagonal energy in the output matrix equalled one-tenth of the total energy in the matrix. DC-SMD was executed with input parameters $I_D = 100$, $I_C = 200$, $P = 10$, and $\hat{M} = 10$. At every iteration step of both algorithms, the diagonalisation and cumulative complexity metrics defined in Sec. IV-A were recorded together with the elapsed execution time. The MSE metric defined in (13) and the length of $\boldsymbol{F}(z)$ were recorded upon each algorithm's completion.

Simulations were performed within Matlab R2014a under Ubuntu 16.04 on an MSI GE60-2OE with Intel® Core™ i7-4700MQ 2.40 GHz × 8 cores and 8 GB RAM.

C. Diagonalisation

The ensemble-averaged diagonalisation was calculated for the SMD and DC-SMD implementations. By evaluating the cumulative complexities and execution times required for both algorithms to achieve a diagonalisation level of $5 \log_{10} E_{\text{norm}} = -10\,\text{dB}$, it is possible to directly compare the performance of both algorithms. Fig. 3 uses the ratio of these metrics to demonstrate algorithm performance for various spatial dimensions, where

$$C_{\text{ratio}} = \frac{\mathcal{E}\{C_{\text{SMD},-10\,\text{dB},M}\}}{\mathcal{E}\{C_{\text{DC-SMD},-10\,\text{dB},M}\}} \, , \quad (14)$$

and

$$t_{\text{ratio}} = \frac{\mathcal{E}\{t_{\text{SMD},-10\,\text{dB},M}\}}{\mathcal{E}\{t_{\text{DC-SMD},-10\,\text{dB},M}\}} \, . \quad (15)$$

From Fig. 3, it is clear that both C_{ratio} and t_{ratio} increase with increasing spatial dimension M; i.e., the use of DC-SMD over SMD becomes more important the larger the matrix to be factorised. Indeed, t_{ratio} reaches a value of 36 for $M = 70$, signifying that DC-SMD is 36 times faster than SMD on average for this dimensionality. Similarly, C_{ratio} reaches 69 for $M = 70$; this demonstrates that DC-SMD requires approximately 69 times fewer multiply-accumulate operations (MACs) than SMD in this scenario.

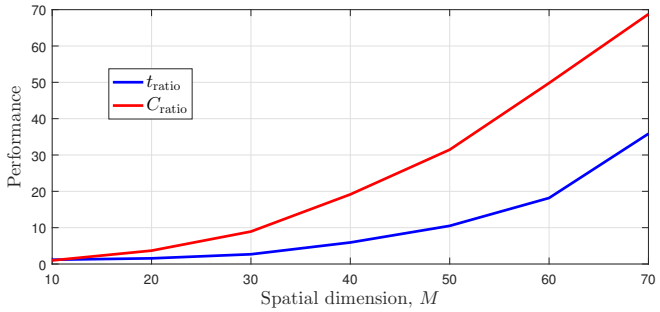

Fig. 3. Ratio of SMD to DC-SMD algorithm cumulative complexity (C_{ratio}) and execution time (t_{ratio}) required to achieve $5\log_{10} E_{\text{norm}} = -10\,\text{dB}$ for $M \in \{10;\ 20;\ 30;\ 40;\ 50;\ 60;\ 70\}$.

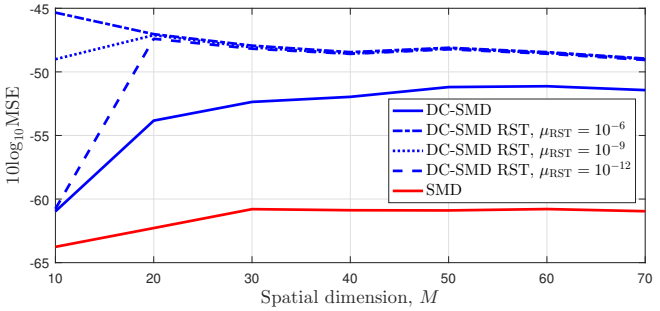

Fig. 4. Mean squared error versus spatial dimension M for SMD and DC-SMD with and without row-shift truncation for $M \in \{10;\ 20;\ 30;\ 40;\ 50;\ 60;\ 70\}$ and $\mu_{\text{RST}} \in \{10^{-6};\ 10^{-9};\ 10^{-12}\}$.

D. Reconstruction Error

The ensemble-averaged mean squared reconstruction error was calculated for both algorithms, according to (13). For DC-SMD, this metric was recorded before and after the utilisation of row-shift truncation, to estimate the method's impact. Fig. 4 shows the results for $M \in \{10;\ 20;\ 30;\ 40;\ 50;\ 60;\ 70\}$; from this, it is clear that the increased diagonalisation speed and lower cumulative complexity of DC-SMD come at the cost of a higher reconstruction error. To reduce this error, parameter δ can be decreased; however, this will reduce the speed and increase the complexity of the algorithm, as more effort will be contributed to the divide step. Note that the relative difference in average MSE remains reasonably constant for increasing M, and that the use of row-shift truncation results in a slightly higher reconstruction error for all dimensionalities.

The row-shift truncation step introduces further error by truncating small values from each row of $\boldsymbol{F}(z)$. This error can be decreased by using a smaller truncation parameter within the row-shift truncation step; however, this comes at the expense of a decreased reduction in paraunitary filter length. It can be observed that, for larger M, increasing μ_{RST} has little impact on the reconstruction error.

E. Paraunitary Filter Length

The ensemble-averaged paraunitary (PU) filter lengths were calculated for both algorithms. For DC-SMD, this metric was recorded before and after the utilisation of row-shift truncation,

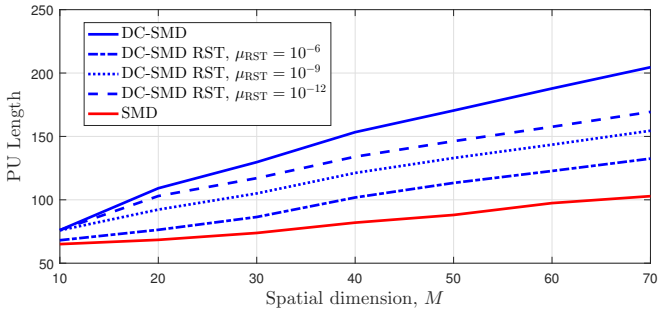

Fig. 5. Paraunitary filter length versus spatial dimension M for SMD and DC-SMD with and without row-shift truncation for $M \in \{10;\ 20;\ 30;\ 40;\ 50;\ 60;\ 70\}$ and $\mu_{\text{RST}} \in \{10^{-6};\ 10^{-9};\ 10^{-12}\}$.

to estimate the method's impact. Fig. 5 shows the results for $M \in \{10;\ 20;\ 30;\ 40;\ 50;\ 60;\ 70\}$. It can be seen from this graph that the average paraunitary filter length is larger for DC-SMD than SMD for all M. The relative difference in average paraunitary filter length becomes larger for increasing M; however, the use of row-shift truncation has successfully narrowed the gap.

Increasing μ_{RST} for this method of truncation only slightly increased reconstruction error for larger M in Fig. 4; however, in Fig. 5, a significant decrease in paraunitary filter length is observed as μ_{RST} is increased for $M > 10$.

Note that — as in [16] — row-shift truncation was found to have minimal impact when applied to the paraunitary filters generated by SMD.

While larger paraunitary filters are disadvantageous for application purposes, the increased performance of DC-SMD in other areas may be of greater importance. In addition, for applications where a small change in reconstruction error is acceptable, increasing parameter μ_{RST} can offer significant filter length reduction.

V. Conclusion

In this paper, we have analysed the performance of a recently developed PEVD algorithm, DC-SMD, for parahermitian matrices of various spatial dimensionality. The parameter M used to describe the dimensionality of such matrices can be directly related to the number of elements within a sensor array. Simulation results have demonstrated that DC-SMD offers significant complexity reduction over a traditional PEVD algorithm, SMD, when processing data analogous to that obtained from large sensor arrays. Furthermore, DC-SMD is able to provide substantially lower execution times than SMD; however, such benefits come with the disadvantage of increasing the mean squared reconstruction error and the paraunitary filter order.

By coupling a row-shift truncation step with DC-SMD, it has been shown that paraunitary filter order can be reduced. Unfortunately, this step also increases the error associated with the decomposition. Depending on the application scenario in which DC-SMD is deployed, a trade-off between mean squared error and paraunitary filter length can be reached.

When designing PEVD implementations for real applications, the potential for the DC-SMD algorithm to increase diagonalisation performance while reducing complexity requirements offers benefits. A further advantage of the DC-SMD algorithm is its ability to produce multiple independent parahermitian matrices, which may be processed in parallel.

Given the results of this paper, it can be concluded that DC-SMD is suitable for broadband multichannel applications with a large number of sensors.

ACKNOWLEDGEMENT

Fraser Coutts is the recipient of a Caledonian Scholarship; we would like to thank the Carnegie Trust for their support.

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.

REFERENCES

[1] C. H. Ta and S. Weiss. A design of precoding and equalisation for broadband MIMO systems. In *Asilomar SSC*, pp. 1616–1620, Pacific Grove, CA, USA, Nov. 2007.

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

[3] S. Weiss, S. Bendoukha, A. Alzin, F. Coutts, I. Proudler, and J. Chambers. MVDR broadband beamforming using polynomial matrix techniques. In *EUSIPCO*, pp. 839–843, Nice, France, Sep. 2015.

[4] A. Alzin, F. Coutts, J. Corr, S. Weiss, I. K. Proudler, and J. A. Chambers. Adaptive broadband beamforming with arbitrary array geometry. In *IET/EURASIP ISP*, London, UK, Dec. 2015.

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

[6] S. Weiss, M. Alrmah, S. Lambotharan, J. McWhirter, and M. Kaveh. Broadband angle of arrival estimation methods in a polynomial matrix decomposition framework. In *IEEE 5th Int. Workshop Comp. Advances in Multi-Sensor Adaptive Proc.*, St. Martin, pp. 109–112, Dec. 2013.

[7] I. Gohberg, P. Lancaster, and L. Rodman. *Matrix Polynomials*. Academic Press, New York, 1982.

[8] J. G. McWhirter, P. D. Baxter, T. Cooper, S. Redif, and J. Foster. An EVD Algorithm for Para-Hermitian Polynomial Matrices. *IEEE TSP*, 55(5):2158–2169, May 2007.

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

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

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

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

[13] Z. Wang, J. G. McWhirter, J. Corr, and S. Weiss. Multiple shift second order sequential best rotation algorithm for polynomial matrix EVD. In *EUSIPCO*, pp. 844–848, Nice, France, Sep. 2015.

[14] J. Corr, K. Thompson, S. Weiss, J. G. McWhirter, and I. K. Proudler. Causality-Constrained multiple shift sequential matrix diagonalisation for parahermitian matrices. In *EUSIPCO*, pp. 1277–1281, Lisbon, Portugal, Sep. 2014.

[15] J. Corr, K. Thompson, S. Weiss, I. Proudler, and J. McWhirter. Row-shift corrected truncation of paraunitary matrices for PEVD algorithms. In *EUSIPCO*, pp. 849–853, Nice, France, Sep. 2015.

[16] J. Corr, K. Thompson, S. Weiss, I. Proudler, and J. McWhirter. Shortening of paraunitary matrices obtained by polynomial eigenvalue decomposition algorithms. In *SSPD*, pp. 1–5, Edinburgh, UK, Sep. 2015.

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

[18] C. H. Ta and S. Weiss. Shortening the order of paraunitary matrices in SBR2 algorithm. In *ICICSP*, pp. 1–5, Singapore, Dec. 2007.

[19] F. Coutts, J. Corr, K. Thompson, S. Weiss, J. Proudler, and J. McWhirter. Memory and Complexity Reduction in Parahermitian Matrix Manipulations of PEVD Algorithms. In *EUSIPCO*, pp. 1633-1637, Budapest, Hungary, August 2016.

[20] F. Coutts, J. Corr, K. Thompson, S. Weiss, I. Proudler, and J. McWhirter. Complexity and Search Space Reduction in Cyclic-by-Row PEVD Algorithms. In *Asilomar SSC*, Pacific Grove, CA, Nov. 2016.

[21] J. J. M. Cuppen. A divide and conquer method for the symmetric tridiagonal eigenproblem. *Num. Mathematik*, 36(2):177–195, June 1980.

[22] J. J. Dongarra and D. C. Sorensen. A fully parallel algorithm for the symmetric eigenvalue problem. *SIAM JSSC*, 8(2):139–154, March 1987.

[23] D. Gill and E. Tadmor. An $O(N^2)$ method for computing the eigensystem of $N \times N$ symmetric tridiagonal matrices by the divide and conquer approach. *SIAM JSSC*, 11(1):161–173, Jan. 1990.

[24] F. Coutts, J. Corr, K. Thompson, I. Proudler, and S. Weiss. Divide-and-Conquer Sequential Matrix Diagonalisation for Parahermitian Matrices. Submitted to *SSPD*, London, UK, Dec. 2017.

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

[26] G.H. Golub and C.F. Van Loan, *Matrix computations*, John Hopkins University Press, Baltimore, Maryland, 4th edition, 2013.