Divide-and-Conquer Sequential Matrix Diagonalisation for Parahermitian Matrices

Fraser K. Coutts*, Jamie Corr*, Keith Thompson*, Ian K. Proudler*,†, Stephan Weiss*

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

University of Strathclyde Engineering

Background

Motivation: Several algorithms for the calculation of a polynomial matrix eigenvalue decomposition (PEVD) have been developed. The PEVD can be used in a number of broadband multichannel problems, including MIMO, beamforming, and angle of arrival estimation.

Aim: Develop a low complexity algorithm for the PEVD by employing divide-and-conquer strategies alongside existing sequential matrix diagonalisation (SMD) [1] algorithm.

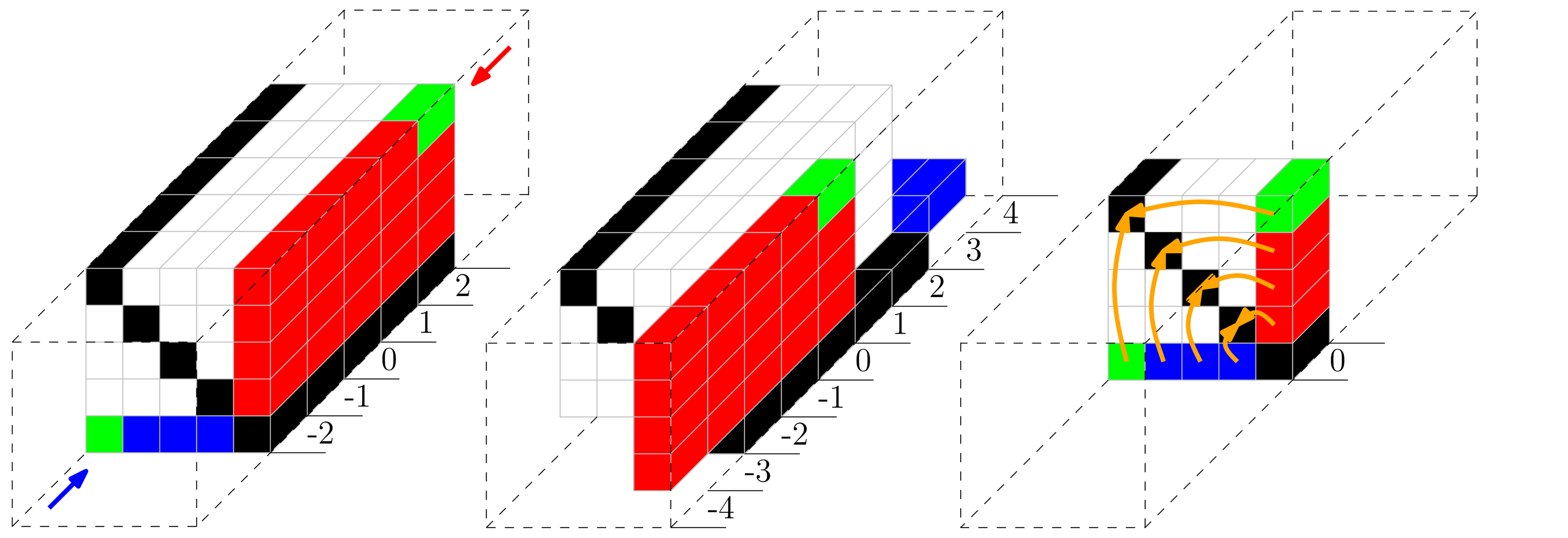
- A space-time covariance matrix $\mathbf{R}[\tau] = \mathcal{E}\{\mathbf{x}[n]\mathbf{x}^{\mathrm{H}}[n-\tau]\}$, $\mathbf{R}[\tau] \in \mathbb{C}^{M\times M}$, can be constructed from the auto- & cross- correlation sequences of vector $\mathbf{x}[n] \in \mathbb{C}^{M}$, where $\mathbf{x}[n]$ is obtained from, e.g., an M-element sensor array.
- $\mathbf{R}[\tau]$ exhibits symmetry about its centre: $\mathbf{R}[\tau] = \mathbf{R}^{\mathrm{H}}[-\tau]$.
- Cross spectral density matrix $\boldsymbol{R}(z) = \sum_\tau \mathbf{R}[\tau]z^{-\tau}$ is a polynomial matrix and exhibits parahermitian symmetry: $\tilde{\boldsymbol{R}}(z) = \boldsymbol{R}^{\mathrm{H}}(1/z^*) = \boldsymbol{R}(z)$.
- The PEVD has been defined as an extension of the eigenvalue decomposition (EVD) to parahermitian polynomial matrices in [2]. The PEVD uses finite impulse response (FIR) paraunitary matrices [3] to approximately diagonalise and spectrally majorise [4] a space-time covariance matrix:

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

Existing Iterative PEVD Algorithms

- Existing iterative PEVD algorithms consist of three major steps:

1. Determine the elements to be shifted onto the zero-lag;
2. Shift the appropriate row and column onto the zero-lag;
3. Transfer energy from the zero-lag onto the diagonal.

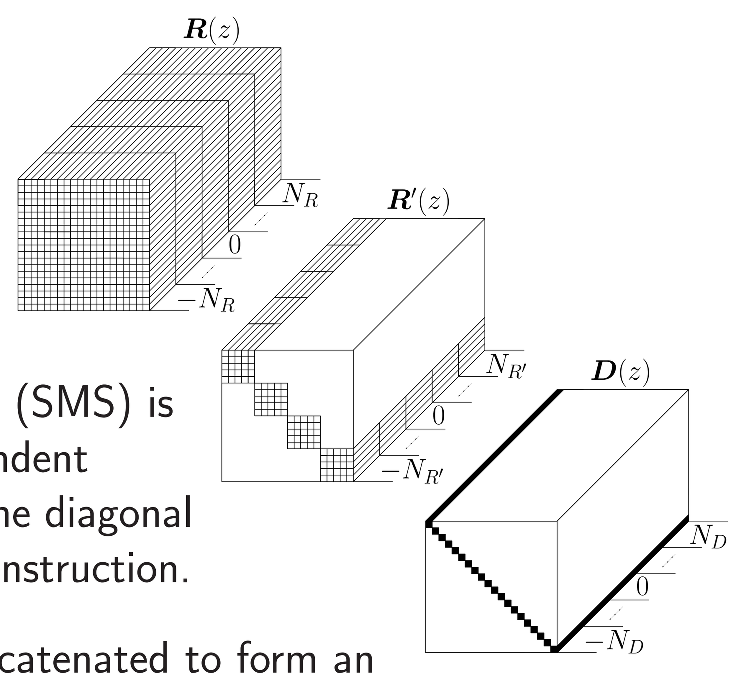


1. $\{k^{(i)},\tau^{(i)}\} = \arg\max_{k,\tau} \|\hat{\mathbf{s}}_k^{(i-1)}[\tau]\|_\infty$ 2. $\mathbf{S}^{(i)\prime}(z) = \mathbf{\Lambda}^{(i)}(z)\mathbf{S}^{(i-1)}(z)\tilde{\mathbf{\Lambda}}^{(i)}(z)$ 3. $\mathbf{S}^{(i)}(z) = \mathbf{Q}^{(i)}\mathbf{S}^{(i)\prime}(z)\mathbf{Q}^{(i)\mathrm{H}}$

- Second order Sequential Best Rotation (SBR2) [2] algorithm uses a Jacobi transformation applied to all lags for step 3.
- The Sequential Matrix Diagonalisation (SMD) [1] algorithm uses a full EVD of the zero-lag (applied to all lags) for step 3.
- Product over I iterations is the paraunitary matrix $\boldsymbol{F}(z) = \prod_{i=1}^{I} \mathbf{Q}^{(i)}\mathbf{\Lambda}^{(i)}(z)$.

Divide-and-Conquer Sequential Matrix Diagonalisation

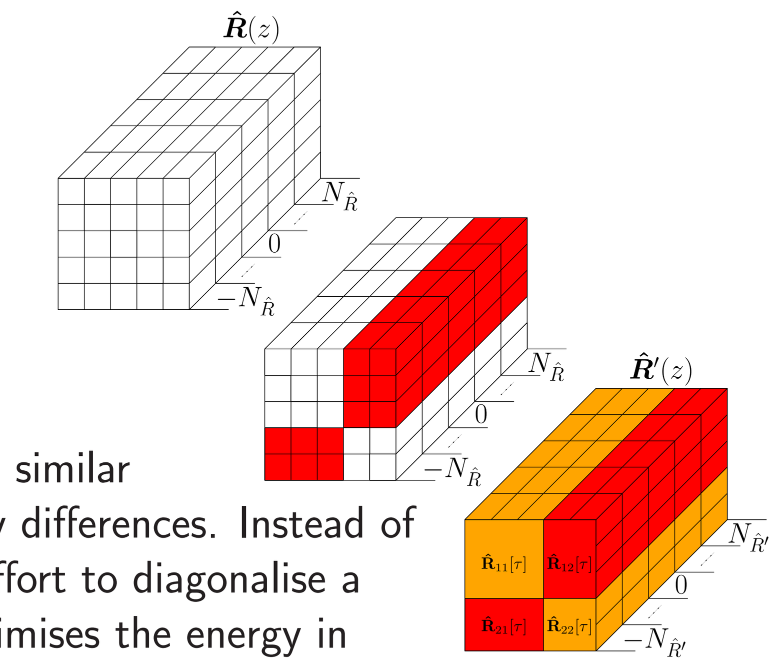
- Research in [5]–[7] has demonstrated that complexity reduction can be obtained by using a divide-and-conquer approach to eigenproblems.
- Inspired by this work, here we describe a divide-and-conquer approach for the PEVD, which can be utilised to reduce algorithm complexity.
- The framework of the developed algorithm — titled divide-and-conquer sequential matrix diagonalisation (DC-SMD) — is based on the SMD algorithm.
- While the SMD algorithm attempts 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.



- An algorithm named sequential matrix segmentation (SMS) is used to recursively divide $\boldsymbol{R}(z)$ into multiple independent parahermitian matrices. Each of these is stored on the diagonal of matrix $\boldsymbol{R}'(z)$; thus, $\boldsymbol{R}'(z)$ is block diagonal by construction.
- The paraunitary matrices generated by SMS are concatenated to form an overall dividing matrix $\boldsymbol{G}(z)$ such that $\boldsymbol{R}(z) \approx \tilde{\boldsymbol{G}}(z)\boldsymbol{R}'(z)\boldsymbol{G}(z)$.
- Each block on the diagonal of matrix $\boldsymbol{R}'(z)$ is then diagonalised in sequence through the use of the SMD algorithm.
- The diagonalised outputs, $\hat{\boldsymbol{D}}(z)$, are placed on the diagonal of matrix $\boldsymbol{D}(z)$, and the corresponding paraunitary matrices, $\hat{\boldsymbol{H}}(z)$, are stored on the diagonal of matrix $\boldsymbol{H}(z)$.
- $\boldsymbol{R}'(z) \approx \tilde{\boldsymbol{H}}(z)\boldsymbol{D}(z)\boldsymbol{H}(z)$.
- By extension, $\boldsymbol{R}(z) \approx \tilde{\boldsymbol{G}}(z)\tilde{\boldsymbol{H}}(z)\boldsymbol{D}(z)\boldsymbol{H}(z)\boldsymbol{G}(z) = \tilde{\boldsymbol{F}}(z)\boldsymbol{D}(z)\boldsymbol{F}(z)$.

Recursive Polynomial Matrix Segmentation

- Sequential matrix segmentation (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.

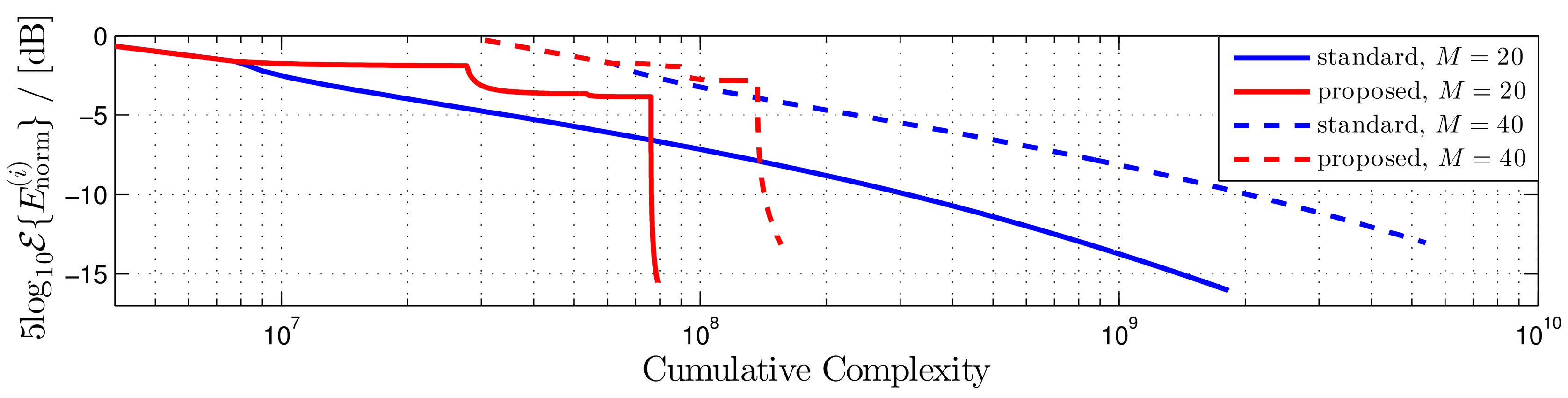


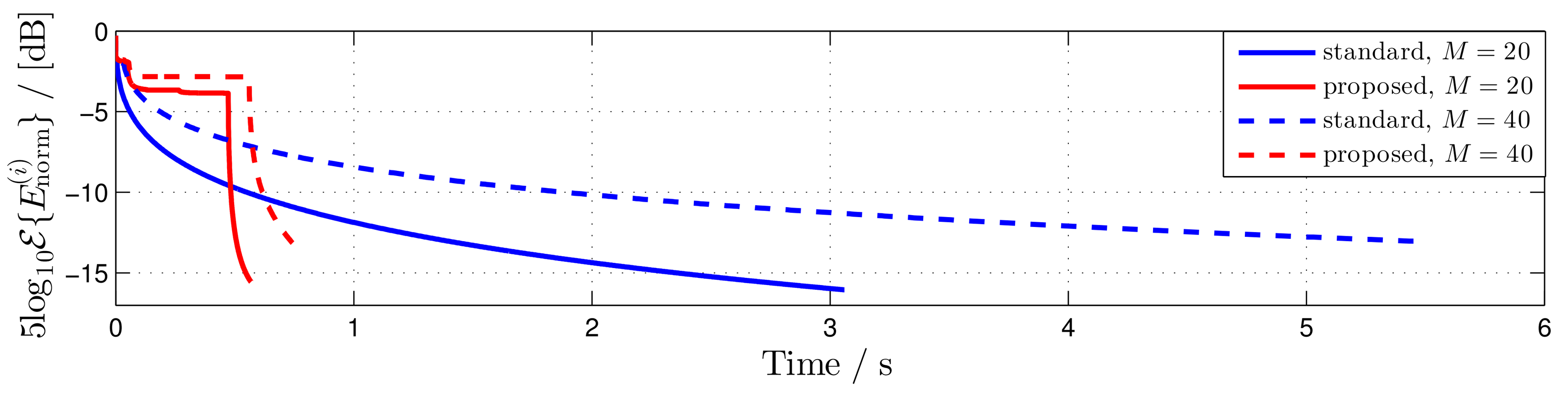
- The SMS algorithm is initialised and operates in a similar manner to the SMD algorithm, but with a few key differences. Instead of iteratively shifting single row-column pairs in an effort to diagonalise a parahermitian matrix $\boldsymbol{S}^{(i)}(z)$, SMS iteratively minimises the energy in select regions of $\boldsymbol{S}^{(i)}(z)$ in an attempt to segment the matrix.

Independent Conquering of Divided Polynomial Matrices

- At this stage of DC-SMD, $\boldsymbol{R}(z) \in \mathbb{C}^{M\times M}$ has been segmented into multiple independent parahermitian matrices, which are stored as blocks on the diagonal of $\boldsymbol{R}'(z)$.
- Each matrix can now be diagonalised individually through the use of the SMD algorithm.
- Upon completion, the SMD algorithm returns matrices $\hat{\boldsymbol{H}}(z)$ and $\hat{\boldsymbol{D}}(z)$, which contain the polynomial eigenvectors and eigenvalues for input matrix $\boldsymbol{A}(z)$, respectively.
- At iteration γ of this stage, $\boldsymbol{A}(z)$ contains the γth block of $\boldsymbol{R}'(z)$ from the bottom-right.

Results





Method	MSE		Paraunitary filter length	
	$M = 20$	$M = 40$	$M = 20$	$M = 40$
standard (SMD)	1.991×10^{-6}	5.643×10^{-7}	116.8	79.13
proposed (DC-SMD)	7.991×10^{-6}	3.477×10^{-6}	154.3	121.8

Conclusions

- We have proposed an alternative technique to compute the polynomial EVD of a parahermitian matrix; this algorithm — named DC-SMD — makes use of a divide-and-conquer approach to the PEVD.
- DC-SMD operates with lower computational complexity and execution time than the traditional SMD algorithm.
- These benefits come with the disadvantage of increasing the mean squared reconstruction error and the paraunitary filter length.
- A further advantage of the DC-SMD algorithm is its ability to produce multiple independent parahermitian matrices, which may be processed in parallel.
- Simulation results demonstrate that DC-SMD outperforms SMD more significantly for larger values of M; therefore, DC-SMD is suitable for broadband multichannel applications with a large number of sensors.

References

[1] S. Redif et al. Sequential Matrix Diagonalisation Algorithms for Polynomial EVD of Parahermitian Matrices. *IEEE Trans. on Signal Processing*, 63(1):81–89, Jan. 2015.
[2] J. G. McWhirter et al. An EVD Algorithm for Para-Hermitian Polynomial Matrices. *IEEE Trans. on Signal Processing*, 55(5):2158–2169, May 2007.
[3] S. Icart and P. Comon. Some properties of Laurent polynomial matrices. In *IMA Int. Conf. Math. Signal Proc.*, Birmingham, UK, Dec. 2012.
[4] P. Vaidyanathan. Theory of optimal orthonormal subband coders. *IEEE TSP*, 46(6):1528–1543, June 1998.
[5] J. J. M. Cuppen. A divide and conquer method for the symmetric tridiagonal eigenproblem. *Num. Mathematik*, 36(2):177–195, June 1980.
[6] J. J. Dongarra and D. C. Sorensen. A fully parallel algorithm for the symmetric eigenvalue problem. *SIAM JSSC*, 8(2):139–154, March 1987.
[7] 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.

fraser.coutts@strath.ac.uk