Generalised Sequential Matrix Diagonalisation for the SVD of Polynomial Matrices

Faizan A. Khattak, Ian K. Proudler, John G. McWhirter, and Stephan Weiss
Department of Electronic & Electrical Engineering, University of Strathclyde, Glasgow G1 1XW, Scotland
{faizan.khattak,ian.proudler,stephan.weiss}@strath.ac.uk

***Abstract*—To extend the singular value decomposition (SVD) to matrices of polynomials, an existing algorithm — a polynomial version of the Kogbetliantz SVD — iteratively targets the largest off-diagonal elements, and eliminates these through delay and Givens operations. In this paper, we perform a complete diagonalisation of the matrix component that contains this maximum element, thereby transfering more off-diagonal energy per iteration step. This approach is motivated by — and represents a generalisation of — the sequential matrix diagonalisation method for parahermitian matrices. In simulations, we demonstrate the benefit of this generalised SMD over the Kogbetliantz approach, both in terms of diagonalisation and the order of the extracted factors.**

I. Introduction

The singular value decomposition (SVD) is a standard linear algebraic tool for the diagonalisation of a rectangular matrix [1]. It has proven central in signal processing to provide solutions to many different challenges [2]. Often solutions can be optimal in various respects; for example for such as for precoding and equalisation for the diagonalisation of multiple-input multiple-output (MIMO) channels [3], where the SVD leads to optimality in least squares and channel capactiy senses. Such matrices typically describe narrowband systems.

In the broadband case, where impulse responses rather than complex-valued gain factors between sources and sensors have to be considered, a multiple-input multiple-output system created by N transmitters and M receivers becomes a matrix of transfer functions. For example, the system $\mathbf{A}(z) : \mathbb{C} \rightarrow \mathbb{C}^{M\times N}$,

$$\boldsymbol{A}(z) = \begin{bmatrix} a_{1,1}(z) & \dots & a_{1,N}(z) \\ \vdots & \ddots & \vdots \\ a_{M,1}(z) & \dots & a_{M,N}(z) \end{bmatrix} , \tag{1}$$

contains in its mth row and nth column the z-transform $a_{m,n}(z) = \sum_\tau a_{m,n}[\tau]z^{-\tau}$, or for short $a_{m,n}(z) \bullet\!\!-\!\!\circ\, a_{m,n}[\tau]$, where $a_{m,n}[\tau]$ is the impulse response between the nth transmitter and the mth receiver. If these impulse responses are finite and causal, then $\boldsymbol{A}(z)$ in (1) is a polynomial matrix. For such matrices, the standard SVD can diagonalise (1) for only one specific value of z, or equivalently $\mathbf{A}[\tau]$ for one one value of τ.

Therefore a different SVD factorisation is required for (1), that can simulaneously diagonalise $\boldsymbol{A}(z)$ for all z, or equivalently $\mathbf{A}[\tau]$ for all τ, such that

$$\boldsymbol{A}(z) \approx \boldsymbol{U}(z)\boldsymbol{\Sigma}(z)\boldsymbol{V}^{\mathrm{P}}(z) \; . \tag{2}$$

The approximation sign is due to the potential truncation of infinite series and other effects that we will briefly review in Sec. II. Decompositions such as (2) have in the past been realised via two polynomial eigenvalue decompositions (PEVDs) applied to two parahermitian matrices $\boldsymbol{A}(z)\boldsymbol{A}^{\mathrm{P}}(z)$ and $\boldsymbol{A}^{\mathrm{P}}(z)\boldsymbol{A}(z)$ [6]. The parahermitian transposition $\boldsymbol{A}^{\mathrm{P}}(z) = \{\boldsymbol{A}(1/z^*)\}^{\mathrm{H}}$ involves a Hermitian transposition and time reversal; if a matrix $\boldsymbol{R}(z)$ satisfies $\boldsymbol{R}^{\mathrm{P}}(z) = \boldsymbol{R}(z)$, it is also termed a parahermitian matrix, for which a number of eigenvalue factorisations have been reported [4], [5], [6], [7], [8], [9], [10], [11]. To avoid the route via two PEVDs, a polynomial QR decomposition has been exploited in [12]. Further, a direct polynomial SVD has been created by a Kogbetliantz-type approach to the SVD [13]. Such algorithms can enable a number of applications ranging from e.g. MIMO communications [14], [15], beamforming [16], to filter bank design and paraunitary matrix completion [17].

The Kogbetliantz method in [13] is a powerful approach that generally yields better diagonalisation and lower order factors than those achieved via two PEVDs. The approach is a generalisation of the second order sequential best rotation (SBR2) algorithm, which calculates the PEVD of parahermitian matrices [6], [8]. It is an iterative algorithm, that in every step eliminates the largest off-diagonal element by transfering its energy onto the diagonal. For the EVD of parahermitian matrices, SBR2 performs a similarity transform, where an elementary paraunitary matrix and its parahermitian transpose are left- and right-multiplied against the result from the previous iteration. In the Kogbetliantz approach in [13], this approach is modified to permit different paraunitary matrices for the left- and right-multiplications. In this paper, we want to explore whether performance improvements can be attained by borrowing ideas from a sequential matrix diagonalisation (SMD) algorithm for parahermitian matrices. In SMD, more energy is transfered per iteration, whereby for the PEVD of parahermitian matrix significant advantages have been reported [18], [19].

Therefore, below we review some aspects of a polynomial SVD in Sec. II before introducting the proposed generalised

The work of Faizan Khattak is supported by a Commonwealth Scholarship. This work was also in part supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/S000631/1 and the MOD University Defence Research Collaboration in Signal Processing.

This is a peer reviewed, accepted author manuscript of the following research contribution: Khattak, F. A., Proudler, I. K., McWhirter, J. G., & Weiss, S. (2023). Generalised sequential matrix diagonalisation for the SVD of polynomial matrices. Paper presented at 12*th International Conference on Sensor Signal Processing for Defence*, Edinburgh, United Kingdom.

SMD algorithm for a direct polynomial SVD in Sec. III. Its performance is explored in Sec. IV.

II. Polynomial SVD

In this section, we highlight the properties of a polynomial SVD as in (2), followed by a brief review of the Kogbetliantz approach in [13].

A. Analytic SVD

For an analytic $\boldsymbol{A}(z) : \mathbb{C} \rightarrow \mathbb{C}^{M\times N}$ that is not tied to a multiplexing operation, there exists an analytic, diagonal $\boldsymbol{\Sigma}(z)$ that is real-valued on the unit circle, as well as analytic left- and right-singular vectors as in the columns of $\boldsymbol{U}(z)$ and $\boldsymbol{V}(z)$ that are unique up to common allpass filters [20], [21]. However, the singular values in $\boldsymbol{\Sigma}(z)$, when evaluated on the unit circle, must be permitted to intersect and even take on negative values — an observation previously made also for the case of matrices in a continuous time parameter [22], [23]. In this case, (2) holds with equality.

Time domain polynomial matrix factorisation methods such as [6], [12], [13], [8], [19], [24] typically encourage spectrally majorised eigen- or singular values, such that for $\boldsymbol{\Sigma}(z) = \mathrm{diag}\{\sigma_1(z), \ldots, \sigma_K(z)\}$ with $K = \min M, N$,

$$\sigma_k(\mathrm{e}^{\mathrm{j}\Omega}) \geq \sigma_{k+1}(\mathrm{e}^{\mathrm{j}\Omega}) \qquad \forall\Omega, \quad k = 1, \ldots, (K-1) \tag{3}$$

is satisfied, i.e. they must not intersect. In case of the polynomial EVD, the SBR2 algorithm in [6], [8] has been explicitly proven to yield a spectrally majorised result [25]. Thus, algorithmic solutions may deviate from the analytic solution, such that, either due to falsely assumed spectral majorisation or due to the approximation of infinite order factors by Laurent polynomials, (2) does not hold with equality. In practice however, due to estimation errors, singular values will be spectrally majorised with probability one [26], such that only approximation errors impact on the precision of (2).

B. Polynomial Kogbetliantz Algorithm

The polynomial Kogletliantz approach in [13] is a generalisation of the SBR2 algorithm, that extends the application of the latter from parahermitian to general matrices. The algorithm starts with the initialisation $\boldsymbol{S}^{(0)}(z) = \boldsymbol{A}(z)$. This initialisation may involve a unitary phase correction to the entire matrix, such that for $\mathbf{S}^{(0)}[\tau] \circ\!\!-\!\!\!\bullet\, \boldsymbol{S}^{(0)}(z)$, the coefficient matrix of order zero, $\mathbf{S}^{(0)}[0]$, is real valued. We will refer to this coefficient matrix for $\tau = 0$ of a polynomial matrix as the 'zero plane' below.

At the ith iteration, the Kogbetliantz approach generates an enhanced diagonalised $\boldsymbol{S}^{(i)}(z)$ in two steps. Firstly, the maximum off-diagonal component is brought to the zero-plane by delay operations. Secondly, a Givens rotation transfers the energy of this component onto the diagonal. For the first step,

$$\{m_i, n_i, \tau_i\} = \arg \max_{\substack{m,n,\tau \\ j\neq k}} |s_{m,n}^{(i-1)}[\tau]| \tag{4}$$

determines the location of the maximum off-diagonal element, where $s_{m,n}^{(i-1)}[\tau]$ is the element in the mth row and nth column of $\mathbf{S}^{(i-1)}[\tau]$. With the identified parameters, two delay matrices

$$\boldsymbol{B}_{\mathrm{r}}^{(i)}(z) = \mathrm{blockdiag}\{\mathbf{I}_{n_i}, z^{-\tau_i}, \mathbf{I}_{M-n_i}\} \tag{5}$$

$$\boldsymbol{B}_{\mathrm{l}}^{(i)}(z) = \mathrm{blockdiag}\{\mathbf{I}_{n_i}, z^{\tau_i}, \mathbf{I}_{N-n_i}\} \tag{6}$$

are formed. Thus,

$$\boldsymbol{S}^{(i-\frac{1}{2})}(z) = \boldsymbol{B}_{\mathrm{l}}^{(i)}(z)\boldsymbol{S}^{(i-1)}(z)\boldsymbol{B}_{\mathrm{r}}^{(i)}(z) \tag{7}$$

contains in its zero plane the same diagonal elements as $\boldsymbol{S}^{(i-1)}(z)$, but the maximum off-diagonal component and also some other elements in the n_ith row and n_ith column have been transfered such that $s_{m_i,n_i}^{(i-\frac{1}{2})}[0] = s_{m_i,n_i}^{(i-1)}[\tau_i]$. This is accomplished by $\boldsymbol{B}_{\mathrm{r}}^{(i)}(z)$ delaying the n_ith column by τ_i samples, while $\boldsymbol{B}_{\mathrm{r}}^{(i)}(z)$ advances the n_ith row.

For the second step, a Givens rotation via matrices $\mathbf{G}_{\mathrm{r}}^{(i)}$ and $\mathbf{G}_{\mathrm{l}}^{(i)}$ transfers the the energy of $s_{m_i,n_i}^{(i-\frac{1}{2})}[0]$ onto the diagonal, such that

$$\boldsymbol{S}^{(i)}(z) = \mathbf{G}_{\mathrm{l}}^{(i)}\boldsymbol{S}^{(i-\frac{1}{2})}(z)\mathbf{G}_{\mathrm{r}}^{(i)} \ . \tag{8}$$

Since this operation is applied across the entire matrix and not just to the zero plane, it may undo some of the efforts in previous iterations. Overall, since the energy in the diagonal monotonuously increases while the overall energy in $\mathbf{S}^{(i)}(z)$ remains unaltered from $\mathbf{S}^{(i-1)}(z)$, the algorithm can be proven to converge [13]. The iterative process is terminated when either the maximum off-diagonal element falls below a given threshold or a specified maximum number of iterations has been reached. Then after K iterations, for the singular values, $\boldsymbol{\Sigma}(z)$ is extracted as the diagonal of $\boldsymbol{S}^{(K)}(z)$, and

$$\boldsymbol{U}(z) = \prod_{i=1}^{K} \left\{\boldsymbol{B}_{\mathrm{l}}^{(i)}(z)\right\}^{\mathrm{P}} \left\{\mathbf{G}_{\mathrm{l}}^{(i)}\right\}^{\mathrm{H}} \tag{9}$$

$$\boldsymbol{V}(z) = \prod_{k=0}^{K-1} \boldsymbol{B}_{\mathrm{r}}^{(K-k)}(z)\mathbf{G}_{\mathrm{r}}^{(K-k)} \tag{10}$$

will provide the left- and right-singular vectors. In order to limit the polynomial order of the extracted factors, it may be advantageous to apply trimming during the iterative process, or after the algorithm has terminated [27], [6], [28], [24], [29].

III. Generalized Sequential Matrix Diagonalization

The Kogbetliantz approach in Sec. II-B represents a generalisation of SBR2; it eliminates one largest off-diagonal element per iteration. The cost is moderate, but the polynomial order of the result grows with every iteration. Therefore, in this section, we generalise the SMD algorithm, which in general is capable of transfering more energy per iteration than SBR2. This generalised SMD algorithm is outlined in this section.

A. Initialisation

The proposed algorithm starts with $\boldsymbol{A}(z) : \mathbb{C} \rightarrow \mathbb{C}^{M\times N}$. Without loss of generality, we assume $M \geq N$, as otherwise we can operate with $\boldsymbol{A}^{\mathrm{P}}(z)$ instead. We first perform a diagonalisation of its zero plane matrix $\mathbf{A}[0]$ via an SVD,

$$\mathbf{A}[0] = \mathbf{U}^{(0)}\mathbf{S}^{(0)}[0]\mathbf{V}^{(0),\mathrm{H}} \ . \tag{11}$$

Based on this factorisation, the initial step for the algorithm is

$$\boldsymbol{S}^{(0)}(z) = \mathbf{U}^{(0),\mathrm{H}}\boldsymbol{A}(z)\mathbf{V}^{(0)} \ . \tag{12}$$

Note that $\mathbf{U}^{(0)}$ and $\mathbf{V}^{(0)}$ are applied to the entirely of $\mathbf{A}(z)$, but will ensure that the zero plane matrix $\mathbf{S}^{(0)}[0]$ is indeed diagonal and real-valued. We also record initial estimates for the left- and right-singular vectors as $\boldsymbol{U}^{(0)}(z) = \mathbf{U}^{(0)}$ and $\boldsymbol{V}^{(0)}(z) = \mathbf{V}^{(0)}$.

B. Iterative Procedure

Following the initialisation in (12), any subsequent iterations $i = 1, 2, \ldots$ repeat the three steps below. Firstly, in the ith iteration, we transfer the n_ith column and the n_ith row of $\boldsymbol{A}^{(i-1)}(z)$ to the zero plane. This step exploits the matrices $\boldsymbol{B}_{\mathrm{r}}^{(i)}(z)$ and $\boldsymbol{B}_{\mathrm{l}}^{(i)}(z)$ defined in (5) and (6), and generates a shifted version $\boldsymbol{S}^{(i-\frac{1}{2})}(z)$ according to (7). The particular row and delay selection for this step can differ from (4), and we define a general column norm that excludes any diagonal elements, such that

$$\|\hat{\mathbf{s}}_n^{(i-1)}[\tau]\|_p = \left\{ \sum_{m=1,m\neq n}^{M} |s_{m,n}^{(i-1)}[\tau]|^p \right\}^{\frac{1}{p}} \ . \tag{13}$$

For $p \to \infty$, the norm picks the maximum element, and the selection is identical to (4). The resulting algorithm version is termed the maximum element generalised SMD (ME-GSMD). Since we ultimately want to perform a complete diagonalisation of the zero plane matrix, it appears promising to shift more energy to the zero plane than with ME-GSMD. This can be accomplished for $p = 2$, and we term the resulting precedure the GSMD algorithm.

Secondly, we now diagonalise the zero plane of the shifted matrix $\boldsymbol{S}^{(i-\frac{1}{2})}(z) = \boldsymbol{B}_{\mathrm{l}}^{(i)}(z)\boldsymbol{S}^{(i-1)}(z)\boldsymbol{B}_{\mathrm{r}}^{(i)}(z)$. By computing an SVD of its zero plane matrix $\mathbf{S}^{(i-\frac{1}{2})}[0] = \mathbf{U}^{(i)}\mathbf{D}^{(i)}\mathbf{V}^{(i),\mathrm{H}}$, we determine

$$\boldsymbol{S}^{(i)}(z) = \mathbf{U}^{(i),\mathrm{H}}\mathbf{S}^{(i-\frac{1}{2})}(z)\mathbf{V}^{(i)} \ . \tag{14}$$

This operation diagonalises the zero plane matrix, but also modifies all other entries in $\mathbf{S}^{(i-\frac{1}{2})}(z)$.

Thirdly, we update the left- and right-singular values as

$$\boldsymbol{U}^{(i)}(z) = \boldsymbol{U}^{(i-1)}(z)\boldsymbol{B}_{\mathrm{l}}^{(i),\mathrm{P}}(z)\mathbf{U}^{(i)} \tag{15}$$

$$\boldsymbol{V}^{(i)}(z) = \boldsymbol{V}^{(i-1)}(z)\boldsymbol{B}_{\mathrm{r}}^{(i)}(z)\mathbf{V}^{(i)} \ , \tag{16}$$

based on the previous estimates, the delay matrices, and the unitary matrices obtained from the application of an SVD in (14).

At each iteration, $\boldsymbol{S}^{(i)}(z)$ grows in order by $2|\tau_i|$, and $\boldsymbol{U}^{(i)}(z)$, and $\boldsymbol{V}^{(i)}(z)$ each grow in order by $|\tau_i|$. It may therefore be opportune to apply trimming [27], [6], [28], [24], [29] at each iteration step, thus stemming the order growth and somewhat arresting the computational complexity of the algorithm.

C. Convergence and Termination

It can be shown that with each iteration step, the overall energy within $\boldsymbol{A}(z)$ remains unaltered while the energy on the diagonal monotonously increases. However, such a proof is beyond the scope of this paper; for the special case of $\boldsymbol{A}(z)$ being a parahermitian matrix, the GSMD algorithm reduces to the SMD algorithm, for which an explicit convergence proof is reported in [19].

The iterations continue until either a sufficiently low threshold for the off-diagonal elements defined via (13) is attained, or until a predefined maximum number of iterations is reached. Thus, after L iterations, we can extract the approximate polynomial SVD factors of (2) as

$$\hat{\boldsymbol{U}}(z) = \boldsymbol{U}^{(L)}(z) \ , \qquad \hat{\boldsymbol{V}}(z) = \boldsymbol{V}^{(L)}(z) \ , \tag{17}$$

while

$$\hat{\boldsymbol{\Sigma}}(z) = \boldsymbol{S}^{(L)}(z) \ . \tag{18}$$

Note that $\hat{\boldsymbol{\Sigma}}(z)$ may still contain some non-zero off-diagonal components albeit of small magnitude.

IV. Simulations and Results

To compare the two proposed approaches, ME-GSMD and GSMD, against the polynomial Kogbetliantz approach in [13] — a generalisation of the SBR2 algorithm [6], [8] — this section presents some simulation results.

A. Performance Metrics

In order to assess the performance of the various polynomial SVD algorithms, we utilise the diagonalisation η,

$$\eta = \frac{\sum_\tau \|\overline{\boldsymbol{\Sigma}}[\tau]\|_{\mathrm{F}}^2}{\sum_\tau \|\hat{\boldsymbol{\Sigma}}[\tau]\|_{\mathrm{F}}^2} \ , \tag{19}$$

where $\overline{\boldsymbol{\Sigma}}[\tau]$ is same as $\hat{\boldsymbol{\Sigma}}[\tau]$ but with its off-diagonal elements set to zero. For a completely diagonalised $\hat{\boldsymbol{\Sigma}}[\tau]$, this metric η would be unity. To assess the algorithmic performance, we also measure the execution time of the algorithms.

For gauging the computational complexity of a practical implementation in general communications or signal processing application, the order of the achieved polynomial SVD factors is important. Therefore we assess the orders of both right and left singular vectors, denoted by $\mathcal{O}\{\hat{\boldsymbol{U}}(z)\}$ and $\mathcal{O}\{\hat{\boldsymbol{U}}(z)\}$ respectively.

B. Numerical Example

Before conducting extensive simulations, we demonstrate the performance of the proposed method using a numerical example. We create a 4×3 polynomial matrix $\boldsymbol{A}(z)$ of order 2, with coefficients drawn from a normal distribution with zero mean and unit variance. The generated matrix is characterised in Fig. 1.

Prior to executing the algorithm, the diagonalisation ratio η is 0.2452. Running GSMD, we achieve a diagonalisation ratio

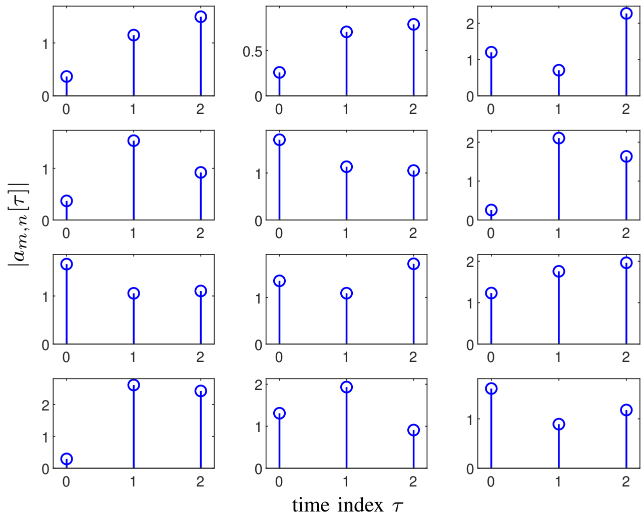

Fig. 1. Matrix $\mathbf{A}[\tau] \circ\!\!-\!\!\bullet \boldsymbol{A}(z)$ for numerical example, showing the moduli of its elements, $|a_{m,n}[\tau]|$, $m = 1, \dots, 4$ and $n = 1, 2, 3$.

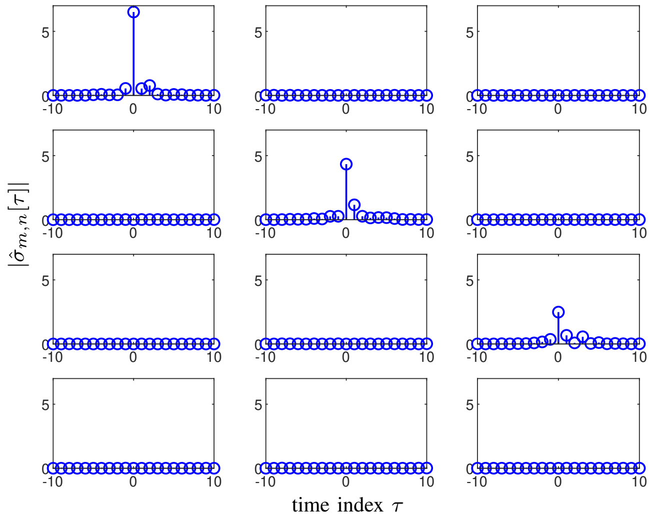

Fig. 2. Approximately diagonalised matrix $\hat{\mathbf{\Sigma}}[\tau]$ derived from $\mathbf{A}[\tau]$ in Fig. 1 using the GSMD algorithm with $L = 100$ iterations.

of $\eta = 0.9998$ after 100 iterations. The resulting diagonalised matrix is shown in Fig. 2 where only 10 central lags are displayed, and trailing values close to zero are suppressed. As a comparison, the generalised SBR2 approach that forms the polynomial Kogbetliantz method in [13] requires 205 iterations to reach a similar diagonalisation ratio as GSMD.

C. Ensemble Results

We construct an ensemble of 500 random instantiations of $\boldsymbol{A}(z) : \mathbb{C} \to \mathbb{C}^{5\times 3}$ of order $\mathcal{O}\{\boldsymbol{A}(z)\} = 2$, whose coefficients are drawn from a normal distribution with zero mean and unit variance. We compare the proposed two algorithms — GSMD and ME-GSMD — against the polynomial Kogbetliantz approach, which implements a generalised SBR2 (GSBR2) algorithm. The various results below represent averages across the ensemble.

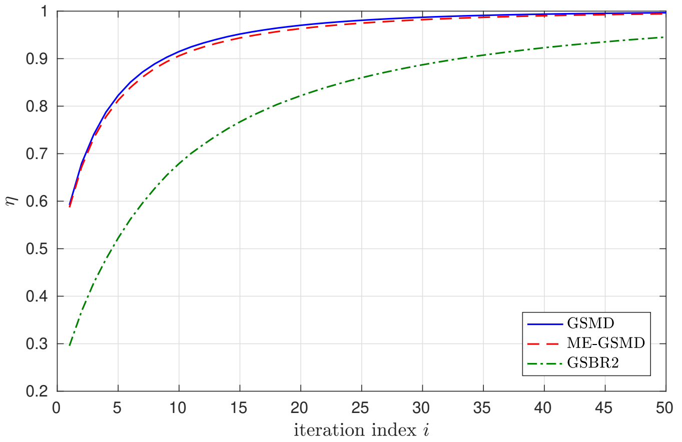

Fig. 3. Ensemble average of η versus iteration number.

First, we compare the diagonalisation performance η of the algorithms over a run of $L = 50$ iteration. For this test, the order of the polynomial SVD factors are of a lesser importance, and in order to ensure that algorithms perform the desired number of iterations, the trimming threshold is set to zero. The average diagonalisation is shown in Fig. 3. All algorithms converge towards $\eta = 1$ as the number of iterations L increases. The GSMD algorithm, utilising the L_2 norm in (13) and hence transfering the column with maximum power in each iteration, provides a slightly faster convergence than the ME-GSMD, which looks for the maximum off-diagonal element via the L_∞ norm in (13). Both of these proposed method converge significantly faster than the benchmark, polynomial Kogbetliantz approach in [13], here refered to as GSBR2.

The execution times over 50 iterations are 0.042 ± 0.08s for GSMD, 0.044 ± 0.009 for ME-GSMD, and 0.04 ± 0.01 for GSBR2. Hence, the algorithms exhibit very similar overall complexities, and Fig. 3 can also be taken as a rough indication of what a comparison of diagonalisation versus execution time would provide.

The order growth of the paraunitary matrices $\hat{\boldsymbol{U}}(z)$ and $\hat{\boldsymbol{V}}(z)$ is illustrated in Figs. 4 and 5, displaying the achievable diagonalisation η versus the orders of the left-singular vectors, $\mathcal{O}\{\hat{\boldsymbol{U}}(z)\}$, in Fig. 4 and the orders of the right-singular vectors, $\mathcal{O}\{\hat{\boldsymbol{V}}(z)\}$, in Fig. 5. The results demonstrate that GSMD and ME-GSMD yield paraunitary matrices that can be more economically applied in order to achieve good diagonalisation, with an advantage for GSMD over ME-GSMD. The former tends to transfer more energy per iteration, and since the order of the polynomial SVD factors are likely to grow with each iteration, the smaller number of iterations indicated in Fig. 3 also translate into lower orders in Figs. 4 and 5. Both methods yield significantly lower order paraunitary matrices compared to the benchmark, GSBR2. Note that this difference in performance is more pronounced for the left-singular vectors in Fig. 4 with $\hat{\boldsymbol{U}}(z) : \mathbb{C} \to \mathbb{C}^{5\times 5}$ compared to the shorter right-singular vectors in the smaller $\hat{\boldsymbol{V}}(z) : \mathbb{C} \to \mathbb{C}^{3\times 3}$ in Fig. 5.

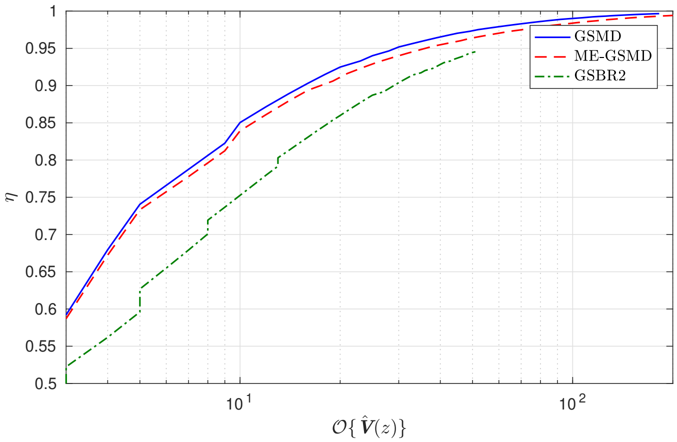

Fig. 4. Ensemble average of η versus ensemble median of order of $\hat{\boldsymbol{U}}(z)$.

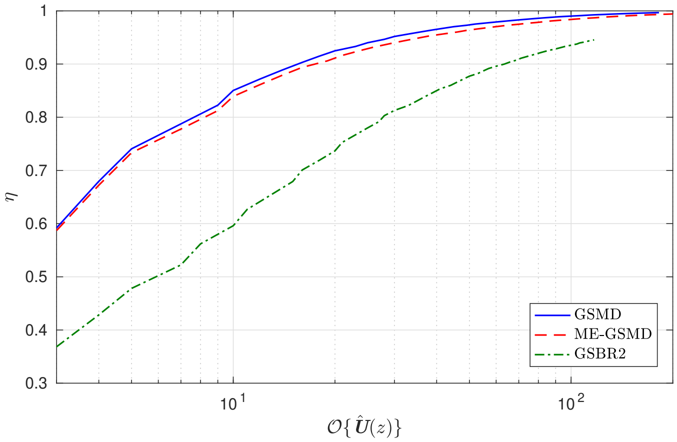

Fig. 5. Ensemble average of η versus ensemble median of order of $\hat{\boldsymbol{V}}(z)$.

V. Conclusion

In this paper, we have proposed a generalised sequential matrix diagonalization algorithm for the SVD of polynomial matrices. This is an extension of the SMD algorithm [18], [19], which is applicable to parahermitian matrices, to the more general case of rectangular matrices of transfer functions. This generalisation is akin to the way a benchmark algorithm, the polynomial Kogbetliantz approach in [13] extends the SRB2 algorithm [6], [8]. Ensemble simulations show that GSMD can achieve better diagonalisation with lower order polynomial matrices compared to this benchmark.

References

[1] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, 1996.

[2] M. Moonen and B. de Moor, Eds., *SVD and Signal Processing, III: Algorithms, Architectures and Applications*. Elsevier, 1995.

[3] M. Vu and A. Paulraj, "MIMO wireless linear precoding," *IEEE Signal Processing Magazine*, **24**(5):86–105, Sep. 2007.

[4] A. Tkacenko and P. Vaidyanathan, "On the spectral factor ambiguity of fir energy compaction filter banks," *IEEE Transactions on Signal Processing*, vol. 54, no. 1, pp. 380–385, Jan. 2006.

[5] ——, "Iterative greedy algorithm for solving the fir paraunitary approximation problem," *IEEE Transactions on Signal Processing*, vol. 54, no. 1, pp. 146–160, Jan. 2006.

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

[7] A. Tkacenko, "Approximate eigenvalue decomposition of para-hermitian systems through successive FIR paraunitary transformations," in *IEEE International Conference on Acoustics Speech and Signal Processing*, Dallas, TX, Mar. 2010, pp. 4074–4077.

[8] S. Redif, J. McWhirter, and S. Weiss, "Design of FIR paraunitary filter banks for subband coding using a polynomial eigenvalue decomposition," *IEEE Trans. Signal Processing*, bf 59(11):5253–5264, Nov. 2011.

[9] M. Tohidian, H. Amindavar, and A. M. Reza, "A DFT-based approximate eigenvalue and singular value decomposition of polynomial matrices," *EURASIP J. Advances Sig. Processing*, **2013**(1):1–16, 2013.

[10] V. W. Neo and P. A. Naylor, "Second order sequential best rotation algorithm with householder reduction for polynomial matrix eigenvalue decomposition," in *IEEE ICASSP*, Brighton, UK, pp. 8043–8047, May 2019.

[11] S. Weiss, I. K. Proudler, and F. K. Coutts, "Eigenvalue decomposition of a parahermitian matrix: Extraction of analytic eigenvalues," *IEEE Trans. Signal Processing*, **69**:722–737, 2021.

[12] J. Foster, J. McWhirter, M. Davies, and J. Chambers, "An algorithm for calculating the qr and singular value decompositions of polynomial matrices," *IEEE Trans. Signal Processing*, **58**(3):1263–1274, Mar. 2010.

[13] J. G. McWhirter, "An algorithm for polynomial matrix SVD based on generalised Kogbetliantz transformations," in *EUSIPCO*, Aalborg, Denmark, pp. 457–461, Aug. 2010.

[14] A. Mertins, "MMSE design of redundant FIR precoders for arbitrary channel lengths," *IEEE Trans. Signal Processing*, **51**(9):2402–2409, Sep. 2003.

[15] N. Moret, A. Tonello, and S. Weiss, "MIMO precoding for filter bank modulation systems based on PSVD," in *IEEE VTC-Spring*, May 2011.

[16] D. Hassan, S. Redif, J.G. McWhirter, and S. Lambotharan, "Polynomial GSVD beamforming for two-user frequency-selective mimo channels," *IEEE Trans. Signal Processing*, **69**:948–959, 2021.

[17] S. Weiss and S. J. Schlecht, "Polynomial procrustes problem: Paraunitaryapproximation of matrices of analytic functions," in *EUSIPCO*, Helsinki, Finland, 2023.

[18] 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 SSP*, Gold Coast, Australia, pp. 312–315, June 2014.

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

[20] G. Barbarino and V. Noferini, "On the Rellich eigendecomposition of para-Hermitian matrices and the sign characteristics of $*$-palindromic matrix polynomials," *arXiv:2211.15539*, 2022.

[21] S. Weiss, I. K. Proudler, G. Barbarino, J. Pestana, and J. G. McWhirter, "On properties and structure of the analytic singular value decomposition,", submitted.

[22] B. De Moor and S. Boyd, "Analytic properties of singular values and vectors," KU Leuven, Tech. Rep., 1989.

[23] A. Bunse-Gerstner, R. Byers, V. Mehrmann, and N. K. Nicols, "Numerical computation of an analytic singular value decomposition of a matrix valued function," *Numer. Math*, **60**:1–40, 1991.

[24] 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, pp. 849–853, Aug. 2015

[25] J. G. McWhirter and Z. Wang, "A novel insight to the SBR2 algorithm for diagonalising para-hermitian matrices," in *11th IMA Conf. Maths in Signal Processing*, Birmingham, UK, Dec. 2016.

[26] F.A. Khattak, S. Weiss, I.K. Proudler, and J.G. McWhirter, "Space-time covariance matrix estimation: Loss of algebraic multiplicities of eigenvalues," in *56th Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, Oct. 2022.

[27] J. Foster, J.G. McWhirter, and J. Chambers, "Limiting the order of polynomial matrices within the SBR2 algorithm," in *IMA Int. Conf. Maths in Signal Processing*, Cirencester, UK, Dec. 2006.

[28] C.H. Ta and S. Weiss, "Shortening the order of paraunitary matrices in SBR2 algorithm," in *ICICSP*, Singapore, Dec. 2007.

[29] J. Corr, K. Thompson, S. Weiss, I. Proudler, and J. McWhirter, "Shortening of paraunitary matrices obtained by polynomial eigenvalue decomposition algorithms," in *Sensor Signal Processing for Defence*, Edinburgh, Scotland, Sep. 2015.