

Advanced Algorithms for Polynomial Matrix Eigenvalue
Decomposition

PhD Thesis

Jamie Corr

Centre for excellence in Signal and Image Processing
Department of Electronic and Electrical Engineering
University of Strathclyde, Glasgow

November 29, 2017

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

Abstract

Matrix factorisations such as the eigen- (EVD) or singular value decomposition (SVD) offer optimality in often various senses to many narrowband signal processing algorithms. For broadband problems, where quantities such as MIMO transfer functions or cross spectral density matrices are conveniently described by polynomial matrices, such narrowband factorisations are suboptimal at best. To extend the utility of EVD and SVD to the broadband case, polynomial matrix factorisations have gained momentum over the past decade, and a number of iterative algorithms for particularly the polynomial matrix EVD (PEVD) have emerged.

Existing iterative PEVD algorithms produce factorisations that are computationally costly (i) to calculate and (ii) to apply. For the former, iterative algorithms at every step eliminate off-diagonal energy, but this can be a slow process. For the latter, the polynomial order of the resulting factors, directly impacting on the implementation complexity, typically grows with every iteration of a PEVD algorithm. The work presented in this thesis helps to reduce both computational complexities.

To address algorithmic complexity and convergence speed, this thesis firstly proposes a multiple shift approach, which can eliminate more off-diagonal energy at every iteration compared to existing methods. Equally applicable to the second order sequential best rotation (SBR2) algorithm, the idea here is applied to the family of sequential matrix diagonalisation (SMD) algorithms, for which a convergence proof is presented. Maximum energy transfer requires a very laborious parameter search; it is demonstrated that a very significant reduction of this search space can be gained while retaining almost all of the transferred energy.

In addition a number of techniques have been developed which improve the efficiency of these PEVD algorithms. With each PEVD algorithm iteration the lengths of the polynomial eigenvectors and eigenvalues increase. To lower the order of the polynomial eigenvectors a novel truncation technique was developed which takes advantage of an ambiguity in the eigenvalues. A drawback of the multiple shift algorithms is the faster growth of the polynomial eigenvectors and eigenvalues; to mitigate this the search step has been analysed and modified to reduce the growth. The SMD algorithm suffers from a performance bottleneck in its EVD step. Here a cyclic-by-row Jacobi approximation is developed that significantly reduces the computational cost of the SMD algorithm with almost no impact on numerical performance.

The impact of these algorithmic advances is studied in a number of applications. Firstly, based on a source model, it is investigated how the conditioning of the input data affects algorithm performance. Secondly, to highlight the benefit of the better converging multiple shift SMD algorithm it is compared to the SBR2 algorithm for broadband angle of arrival estimation, where the proposed multiple shift versions can achieve a more accurate subspace decomposition and hence a more accurate AoA estimation. Lastly, it is demonstrated how PEVD algorithms can be utilised to extend other linear algebraic techniques to polynomial matrices. The example used here is the extension of the generalised eigenvalue decomposition (GEVD) to a polynomial matrix GEVD.

Contents

Abstract	ii
Abbreviations	viii
Mathematical Notations	x
Commonly used Symbols	xi
List of Figures	xi
List of Tables	xiv
List of Publications	xvi
Acknowledgements	xix
1 Introduction	1
1.1 Motivation	1
1.2 Objectives and Contributions	4
1.3 Overview	7
2 Background	10
2.1 Scalar Matrices	11
2.1.1 Covariance Matrix	11
2.1.2 Scalar EVD	11
2.2 Polynomial Matrices	14

Contents

2.2.1	Space-Time Covariance Matrix	14
2.2.2	Properties of Polynomial Matrices	14
2.3	Polynomial EVD	16
2.4	PEVD Algorithms	17
2.4.1	SBR2	18
2.4.2	SMD Algorithms	19
2.5	PEVD Ambiguity	20
2.5.1	Shift Matrix Ambiguity	21
2.5.2	Paraunitary Ambiguity	22
2.5.3	Truncation Methods	23
2.6	Chapter Summary	24
3	Multiple Shift PEVD Algorithms	26
3.1	Maximum Energy SMD	27
3.2	Multiple Shift Maximum Element SMD	31
3.2.1	Idea	31
3.2.2	Algorithm	31
3.3	Multiple Shift SBR2	36
3.3.1	Idea	36
3.3.2	Algorithm	37
3.4	Proof of Convergence	37
3.5	Search Complexities	40
3.5.1	Maximum Element Search	40
3.5.2	Column Norm Search	40
3.5.3	Multiple Shift Searches	40
3.5.4	Comparison	41
3.6	Results	42
3.6.1	Performance Metrics	42
3.6.2	Energy Transfer	43
3.6.3	Diagonalisation	46
3.6.4	Real Time Convergence	47

Contents

3.6.5	Order Increase	48
3.7	Chapter Summary & Conclusions	49
4	Efficient Implementations	52
4.1	Row-Shift Truncation	53
4.1.1	State-of-the-Art Truncation	53
4.1.2	Proposed Row-Shift Truncation	54
4.1.3	Truncation Example	55
4.1.4	Results	57
4.2	Restricted Search Algorithms	63
4.2.1	Polynomial Order Growth	63
4.2.2	Restricted Search Algorithms	70
4.2.3	Results	73
4.3	Cyclic-by-Row PEVD Approximation	77
4.3.1	EVD Approximation	78
4.3.2	PEVD Approximation	78
4.3.3	Cyclic-by-Row SMD Algorithms	79
4.3.4	Results	80
4.4	Chapter Summary & Conclusions	83
5	Applications of Iterative PEVD Algorithms	85
5.1	Implications of Source Model Conditioning	85
5.1.1	Source Model Conditioning	86
5.1.2	Polynomial Eigenvalue Decomposition	87
5.1.3	Eigenvalue Spread	89
5.1.4	Results	89
5.2	Angle of Arrival Estimation using Polynomial MUSIC	93
5.2.1	MUSIC Algorithm	94
5.2.2	Polynomial MUSIC Algorithm	96
5.2.3	Results	97
5.3	Polynomial Generalised Eigenvalue Decomposition	99

Contents

5.3.1	Generalised Eigenvalue Decomposition	100
5.3.2	GEVD Extended to Polynomial Matrices	102
5.3.3	Polynomial Matrix Inverse	103
5.3.4	Results	104
5.4	Chapter Summary & Conclusions	108
6	Conclusions & Future Work	111
6.1	Thesis Summary	111
6.1.1	Multiple Shift Algorithms	111
6.1.2	Efficient Implementations	112
6.1.3	PEVD Applications	112
6.2	Future Work	113
	References	116

Abbreviations

3-D	Three Dimensional
APEVD	Approximate PEVD Algorithm
approx.	approximate
C-MSME-SMD	Causality constrained MSME-SMD Algorithm
CbR-SMD	Cyclic-by-Rows approximation of SMD Algorithm
CSS	Coherent Signal Subspace
DFT	Discrete Fourier Transform
DSP	Digital Signal Processing
EVD	Eigenvalue Decomposition
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
GEVD	Generalised Eigenvalue Decomposition
IFB	Independent Frequency Bins
ME-SMD	Maximum Element SMD Algorithm
MIMO	Multiple Input Multiple Output
MS-SBR2	Multiple Shift SBR2 Algorithm

Chapter 0. Abbreviations

MSME-SMD	Multiple Shift Maximum Element SMD Algorithm
MUSIC	MUltiple SIgnal Classification
PEVD	Polynomial-matrix Eigenvalue Decomposition
PGEVD	Polynomial-matrix Generalised Eigenvalue Decomposition
P-MUSIC	Polynomial-matrix MUSIC
RS-MSME-SMD	Reduced Search space MSME-SMD Algorithm
SBR2	Second order Sequential Best Rotation Algorithm
SMD	Sequential Matrix Diagonalisation Algorithm
w.r.t	with respect to

Mathematical Notations

$\lceil \cdot \rceil$	Ceiling, round up
$\text{diag}\{\cdot\}$	Matrix containing entries in the brackets on the diagonal
$\mathcal{E}\{\cdot\}$	Expectation
$\lfloor \cdot \rfloor$	Floor, round down
$\{\cdot\}^H$	Hermitian transpose
$\mathbf{I}_{M \times M}$	$M \times M$ Identity matrix
$\mathcal{O}(\cdot)$	Order measure of computational complexity
$\tilde{\{\cdot\}}$	Parahermitian transpose
$\{\cdot\}^T$	Transpose

Commonly used Symbols

$\mathbf{R}(z)$	Array output cross spectral density matrix
$\mathbf{S}^{(0)}(z)$	Initial input into PEVD algorithm, usually $\mathbf{R}(z)$
$\mathbf{S}^{(i)}(z)$	Parahermitian matrix at the i th iteration of a PEVD algorithm
$\mathbf{S}^{(i)'}(z)$	$\mathbf{S}^{(i)}(z)$ after $\mathbf{\Lambda}^{(i)}(z)$ has been applied
$\mathbf{Q}(z)$	Paraunitary matrix containing polynomial eigenvectors
$\mathbf{D}(z)$	Diagonal polynomial matrix containing polynomial eigenvalues
$\mathbf{\Lambda}^{(i)}(z)$	Polynomial delay matrix from the i th step of a PEVD algorithm
$\mathbf{Q}^{(i)}$	Unitary matrix from the i th iteration of a PEVD algorithm
M	Number of array elements/Matrix width
L	Length of/number of lags in parahermitian matrix

List of Figures

2.1	Far field signal arriving at sensor array.	10
2.2	Cyclic-by-row Jacobi sweep of a 5×5 matrix	13
2.3	Examples of spectral majorisation	17
2.4	Initial parahermitian matrix (a) and result of delaying row 1 (b).	21
2.5	Results of advancing rows: 2 (a), 3 (b), 4 (c) and 5 (d).	22
3.1	Visualisation of possible delay matrices for $M = 4$ and $L = 1$	28
3.2	Trellis of paths representing possible shift matrices.	29
3.3	Number of shift combinations for an exhaustive search algorithm.	30
3.4	Sparsity structure of lag zero matrix.	32
3.5	Best case scenario for the Naive “Greedy” method.	32
3.6	Worst case scenario for the “Greedy” method.	33
3.7	Difference between best and worst case multiple shift scenarios.	33
3.8	Masking used in the method that maximises the number of shifts.	34
3.9	Masking used in the C-MSME-SMD algorithm	36
3.10	Masking used in the MS-SBR2 algorithm.	37
3.11	Percentage energy transfer comparison varying L	45
3.12	Percentage energy transfer comparison varying M	46
3.13	Relative energy transfer comparison.	47
3.14	Comparison of convergence for multiple shift algorithms.	48
3.15	Off-diagonal energy vs. execution time.	49
3.16	Growth of paraunitary matrices.	50

List of Figures

4.1	PSDs from source model and PEVD.	56
4.2	Row-shift corrected paraunitary truncation	57
4.3	Reconstruction error of different paraunitary truncation methods.	59
4.4	Paraunitary order of different paraunitary truncation methods.	61
4.5	Reconstruction error for different PEVD algorithms.	62
4.6	Truncated paraunitary order for different PEVD algorithms.	64
4.7	Initial $5 \times 5 \times 5$ parahermitian matrix.	65
4.8	The first (a) and second (b) steps of a single shift algorithm.	66
4.9	The first (a) and second (b) steps of the MSME search algorithm.	67
4.10	The third (a) and fourth (b) steps of the MSME search algorithm.	67
4.11	The fifth and final step of the MSME search algorithm.	68
4.12	The first (a) and second (b) steps of the MS-SBR2 search algorithm.	69
4.13	The final step of the MS-SBR2 search algorithm.	70
4.14	The first (a) and second (b) steps of the RS-MSME search algorithm.	71
4.15	The third (a) and fourth (b) steps of the RS-MSME search algorithm.	71
4.16	The first (a) and second (b) steps of the OC-MS-SBR2 search algorithm.	73
4.17	Comparison of MSME & RS-MSME -SMD convergence.	75
4.18	RS-MSME-SMD order growth.	76
4.19	RS-MSME-SMD real time convergence.	77
4.20	Cyclic-by-row algorithm convergence	81
4.21	Cyclic-by-row algorithm execution times	82
5.1	Source model block diagram.	86
5.2	Unmajorised source frequency reassignment.	87
5.3	Frequency reassignment using Haar and 32C filters.	88
5.4	Source model reduction in off-diagonal energy.	91
5.5	Source model paraunitary matrix order.	92
5.6	Source model 10 dB power spectral density.	93
5.7	Source model 20 dB power spectral density.	93
5.8	Comparison of SSP-MUSIC results.	98
5.9	Power spectral densities for polynomial GEVD inversion.	104

List of Figures

5.10	Space-time covariance matrices for polynomial GEVD.	106
5.11	Space-time covariances after joint diagonalisation.	107
5.12	Polynomial GEVD reduction in off-diagonal energy.	109
5.13	Closeness to the identity matrix.	109

List of Tables

3.1	Order comparison of SMD search methods.	41
4.1	Worst case order growth of PEVD algorithms.	73
5.1	Performance metrics for different source models.	94

List of Publications

1. P. Karagiannakis, K. Thompson, J. Corr, I. K. Proudler, S. Weiss
Distributed Processing of a Fractal Array Beamformer. IET Intelligent Signal Processing Conference, London, England, December 2013. [80]
2. J. Corr, K. Thompson, S. Weiss, J. G. McWhirter, S. Redif, I. K. Proudler
Multiple Shift Maximum Element Sequential Matrix Diagonalisation for Parahermitian Matrices. IEEE Workshop on Statistical Signal Processing, Gold Coast, Australia, July 2014. [39]
3. J. Corr, K. Thompson, S. Weiss, J. G. McWhirter, I. K. Proudler
Causality Constrained Multiple Shift Sequential Matrix Diagonalisation for Parahermitian Matrices. 22nd European Signal Processing Conference, Lisbon, Portugal September 2014. [40]
4. J. Corr, K. Thompson, S. Weiss, J. G. McWhirter, I. K. Proudler
Cyclic-by-Row Approximation of Iterative Polynomial EVD Algorithms. Sensor Signal Processing for Defence Conference 2014, Edinburgh, Scotland, September 2014. [45]
5. M. A. Alrmah, J. Corr, A. Alzin, K. Thompson, S. Weiss
Polynomial Subspace Decomposition for Broadband Angle of Arrival Estimation. Sensor Signal Processing for Defence Conference 2014, Edinburgh, Scotland, September 2014. [13]
6. J. Corr, K. Thompson, S. Weiss, J. G. McWhirter, I. K. Proudler
Maximum Energy Sequential Matrix Diagonalisation for Parahermitian Matrices.

Chapter 0. List of Publications

- 48th Asilomar Conference on Signals Systems and Computers, Pacific Grove, California, USA, November 2014. [41]
7. J. Corr, K. Thompson, S. Weiss, I. K. Proudler, J. G. McWhirter
Row-Shift Corrected Truncation of Paraunitary Matrices for PEVD Algorithms. 23rd European Signal Processing Conference, Nice, France September 2015. [43]
 8. Z. Wang, J. G. McWhirter, J. Corr, S. Weiss, I. K. Proudler
Multiple Shift Second Order Sequential Best Rotation Algorithm for Polynomial Matrix EVD. 23rd European Signal Processing Conference, Nice, France September 2015. [42]
 9. J. Corr, K. Thompson, S. Weiss, I. K. Proudler, J. G. McWhirter
Shortening of Paraunitary Matrices Obtained by Polynomial Eigenvalue Decomposition Algorithms. Sensor Signal Processing for Defence Conference 2015, Edinburgh, Scotland September 2015. [81]
 10. J. Corr, K. Thompson, S. Weiss, I. K. Proudler, J. G. McWhirter
Reduced Search Space Multiple Shift Maximum Element Sequential Matrix Diagonalisation Algorithm. IET Conference on Intelligent Signal Processing, London, England December 2015. [44]
 11. J. Corr, K. Thompson, S. Weiss, I. K. Proudler, J. G. McWhirter
Impact of Source Model Matrix Conditioning on PEVD Algorithms. IET Conference on Intelligent Signal Processing, London, England December 2015. [46]
 12. A. Alzin, F. K. Coutts, J. Corr, S. Weiss, I. K. Proudler, J. A. Chambers
Adaptive Broadband Beamforming with Arbitrary Array Geometry. IET Conference on Intelligent Signal Processing, London, England December 2015. [15]
 13. J. Corr, K. Thompson, S. Weiss, J. G. McWhirter, I. K. Proudler
Performance Trade-Offs in Sequential Matrix Diagonalisation Search Strategies. 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, Cancun, Mexico December 2015. [59]

Chapter 0. List of Publications

14. Z. Wang, J. G. McWhirter, J. Corr, S. Weiss
Order-Controlled Multiple Shift SBR2 Algorithm for Para-Hermitian Polynomial Matrices. 2016 IEEE Sensor Array and Multichannel Signal Processing Workshop, Rio de Janeiro, Brazil, July 2016. [61]
15. F. K. Coutts, J. Corr, K. Thompson, S. Weiss, I. K. Proudler, J. G. McWhirter
Memory and Complexity Reduction in Parahermitian Matrix Manipulations of PEVD Algorithms. 24th European Signal Processing Conference, Budapest, Hungary, September 2016. [82]
16. J. Corr, J. Pestana, S. Weiss, S. Redif, M. Moonen
Investigation of a Polynomial Matrix Generalised EVD for Multi-Channel Wiener Filtering. 50th Asilomar Conference on Signals Systems and Computers, Pacific Grove, California, USA, November 2016. [47]
17. F. K. Coutts, J. Corr, S. Weiss, I. K. Proudler, J. G. McWhirter
Complexity and Search Space Reduction in Cyclic-by-Row PEVD Algorithms. 50th Asilomar Conference on Signals Systems and Computers, Pacific Grove, California, USA, November 2016. [83]
18. A. Alzin, F. K. Coutts, J. Corr, S. Weiss, I. K. Proudler, J. Chambers
Polynomial Matrix Formulation Based Capon Beamformer. 2016 IMA International Conference on Signal Processing in Mathematics, Birmingham, England, December 2016. [16]
19. F. K. Coutts, J. Corr, K. Thompson, S. Weiss, I. K. Proudler, J. G. McWhirter
Multiple Shift QR Decomposition for Polynomial Matrices 2016 IMA International Conference on Signal Processing in Mathematics, Birmingham, England, December 2016. [34]
20. F. K. Coutts, J. Corr, K. Thompson, S. Weiss, I. K. Proudler
Divide-and-Conquer Sequential Matrix Diagonalisation Sensor Signal Processing for Defence Conference, London, England, December 2017. [56]

Acknowledgements

I would like to express my gratitude to Dr Stephan Weiss for his encouragement and support throughout the three and a half years of my PhD. In particular for giving me the opportunities to visit many interesting conferences and a placement at KU Leuven in Belgium. In addition to Dr Weiss I have been very lucky to be able to work closely with Prof. John McWhirter and Prof. Ian Proudler who have allowed me to tap into their extensive combined knowledge of signal processing as well as all the interesting tangents that we have ended up discussing along the way. Discussions with Dr Keith Thompson has also been of great help when I have been developing and refining my ideas. I must also mention Prof. Marc Moonen who along with his Stadius group facilitated my visit to KU Leuven and made my time in Belgium both productive and enjoyable.

During both my undergraduate and PhD at Strathclyde I have been very fortunate to meet and work with many friendly and inspiring individuals both here and abroad. In particular I have met many good friends and colleagues through the CeSIP group.

Finally I would like to thank my family and friends in Scotland and around the world who provided great support and encouragement to my studies.

Chapter 1

Introduction

1.1 Motivation

The scalar eigenvalue decomposition (EVD) is of great importance to many narrowband digital signal processing (DSP) problems. More generally scalar EVD is key to a large variety of DSP applications from angle of arrival estimation [1] to facial recognition [2] in image processing. For narrowband array processing situations specifically, phase shifts can be used to synthesise the time delays experienced between array elements. Data is then stored in a scalar covariance matrix consisting of complex gain factors; this is known as the instantaneous mixing model. Extending this idea to broadband signals, one could utilise the independent frequency bin (IFB) approach, whereby the broadband problem is split into a number of narrowband problems using the DFT or FFT. Using the IFB approach does however lose the correlations and phase-coherence between frequency bands [3, 4].

The scalar matrices used in the instantaneous mixing model are not particularly suitable in the case of broadband sensor arrays. Instead the mixing is better modelled by a matrix of finite impulse response (FIR) filters, this is termed the convolutive mixing model. The sensor outputs from the convolutive mixing model will generally be correlated with one another and this can no longer be decorrelated using the scalar EVD. The scalar EVD can only remove instantaneous correlation which is the correlation between signals sampled at the same time instant. With convolutive mixing the

signals will generally be correlated on more than one time instant and therefore the signals should be decorrelated over a suitable range of time delays [3]. This type of decorrelation is often referred to as strong decorrelation [5]. An added benefit of the convolutive mixing model is the ability to model the effects of multipath propagation, which is particularly important in many sensor array signal processing applications [3].

In broadband array processing the time delays between elements can be modelled as polynomials, similar to an FIR filter. The scalar covariance matrix from the narrowband case is then replaced with a polynomial space-time covariance matrix; which consists of auto- and cross-correlation sequences. Like the traditional signal processing notation the polynomials are in terms of z^{-1} ; multiplication by z^{-1} is therefore often referred to as applying a delay. The addition of the polynomial (or lag) dimension means that the scalar EVD used in narrowband problems is no longer suitable to diagonalise this three dimensional structure. An alternative polynomial-matrix EVD (PEVD) is therefore required to diagonalise the space-time covariance matrix and achieve strong decorrelation.

One of the original PEVD algorithms is the second order sequential best rotation algorithm (SBR2) [3]. The SBR2 method was introduced in 2007 by McWhirter et. al [3] is based on the classical Jacobi algorithm for scalar EVD; the origins of which date back to Jacobi's original paper in 1846 [6], with a full description of the classical Jacobi algorithm is given in [7]. The algorithm aims to systematically reduce the off-diagonal energy using a series of Jacobi transformations. The classical Jacobi algorithm consists of two main steps:

1. locate the maximum off-diagonal element;
2. apply a Jacobi transformation to bring the energy from maximum off-diagonal element onto the diagonal.

In the classical Jacobi method outlined above, at each step the scalar matrix is brought closer to a diagonal matrix. After a sufficient number of steps the diagonal matrix will approximate the eigenvalues of the original matrix. The eigenvectors are then simply the product of the Jacobi transformations used to diagonalise the matrix.

In [3] McWhirter et. al extend the Jacobi algorithm for use in computing a PEVD. As the PEVD is applied to polynomial matrices – which have a lag (polynomial) dimension – an additional step is needed to bring the maximum off-diagonal element onto the (central) zero lag. The general algorithm becomes:

1. find the maximum off-diagonal element;
2. if required bring the maximum element onto the zero lag using a time delay or advance operation;
3. apply a Jacobi transformation to eliminate the maximum off-diagonal element.

In SBR2, like the classical Jacobi method, at each step the off-diagonal energy of the polynomial matrix is reduced and with sufficient steps it will approximate the polynomial eigenvalues. The polynomial eigenvectors are then the product of the advance/delay operation and the Jacobi transform in each step. In both of these algorithms the stopping criterion is either when the off-diagonal energy or the maximum off-diagonal element falls below a predefined threshold, as a failsafe the number of iterations can also be limited. SBR2 with its added advance/delay step means that the polynomial matrices containing the approximate eigenvalues and (paraunitary – explained in Chapter 2) approximate eigenvectors grow in lag dimension at each iteration.

In addition to SBR2, a number of other iterative PEVD algorithms have been developed:

- Approximate PEVD (APEVD) [8, 9] is similar to the SBR2 approach but the polynomial/lag dimension is fixed by limiting the values that can be diagonalised at each iteration; however APEVD has not been proven to converge so will not be considered any further in this thesis.
- Sequential Matrix Diagonalisation algorithm (SMD) [10] differs from SBR2 as it replaces the Jacobi transform with a full scalar EVD of the zero lag at each iteration. SMD also comes in two variants:

1. the standard SMD approach uses column norm to select the rows/columns to be brought onto the zero lag;
2. the Maximum Element SMD (ME-SMD) uses the same maximum element search as SBR2 to reduce the computational expense of the SMD algorithm.

Polynomial matrices have been used for some time in control applications [11] and have more recently found their way into DSP applications. Some of the DSP applications include: direction of arrival estimation [12–14], beamforming [15–17], filterbank based channel coding [18–22], MIMO communications [23–28], broadband subspace decomposition [29] and optimal subband coders [30, 31]. This thesis focusses primarily on PEVD algorithms but polynomial matrix decompositions have also been developed for the singular value [32, 33] and QR [32, 34] decompositions. Most recently the idea of the polynomial singular value and QR decompositions has been extended to a range of different algebras in [35].

Distinct to the iterative methods mentioned above, two DFT domain approaches to calculate the PEVD are introduced in [36] and [37]. Both [36] and [37] implement a strict limitation on the polynomial order of the eigenvalues which results in accuracy issues. In [38] the polynomial EVD is solved on the unit circle so is similar to [37] but does not provide directly an implementable time domain solution. In addition the DFT domain approach requires some relaxations in the properties defined in [31] and [3] meaning that the PEVD obtained is not necessarily equivalent [10].

1.2 Objectives and Contributions

The objective of this research is to further develop the knowledge and understanding of polynomial-matrix eigenvalue decompositions. This work can be subdivided into three main parts:

- development of new powerful PEVD algorithms;
- reduce inefficiencies associated with new and existing PEVD algorithms;
- investigation of PEVD algorithms applied in a variety of different scenarios.

The main contributions from this research can be subdivided into the following four sections.

1. **Multiple Shift PEVD Algorithms** [39–42]

All pre-existing iterative PEVD algorithms share a common feature in that they only ever bring energy from one row/column pair at each iteration. An important part of this thesis is the development of multiple shift PEVD algorithms which aim to maximise the energy transfer at each algorithm iteration. The first multiple shift search technique introduced uses an exhaustive search on every possible shift combination to guarantee that the maximum amount of energy is transferred. Although the exhaustive search transfers the maximum amount of energy the implementation cost (due to the vast number of shift possibilities) makes it far too costly for any practical application. As a lower cost alternative the Multiple Shift Maximum Element (MSME) search technique which transfers slightly less energy than the exhaustive search is proposed. Although less effective than the exhaustive search the MSME search method outperforms all single shift algorithms. The MSME search method also lends itself well to a causal implementation which generates a paraunitary matrix that consists of only delays. A causal paraunitary matrix is desirable when a PEVD algorithm is utilised in multichannel spectral factorisation. The idea of multiple shifts has also been adapted to create the Multiple Shift (MS) SBR2 algorithm.

2. **Complexity Reduction in PEVD Algorithms** [43–45]

Through the development of the exhaustive multiple shift search method an interesting row-shift ambiguity in the paraunitary matrices was uncovered. Traditionally a delay ambiguity has been used in the truncation of paraunitary matrices however using the row-shift ambiguity results in a lower error. Based on the lower error, a more aggressive truncation can be applied resulting in truncated paraunitary matrices with a lower polynomial dimension. A drawback identified with the multiple shift methods is the growth in their polynomial matrices. After investigating the worst case growth for the MSME-SMD algorithm a minor modification has been identified which can significantly reduce the polynomial growth

and with it the overall implementation cost. Based on the position of the initial maximum element at each iteration, the modification sees subsequent searches restricted to lags no further from the zero lag than the initial maximum element. Despite transferring more energy at each iteration the SMD based algorithms have always been more computationally expensive than the SBR2 algorithm; meaning the SBR2 algorithm converges faster in real time. The performance difference comes down to the cost of applying a non-sparse EVD modal matrix to all lags of the parahermitian matrix. Based on the cyclic-by-rows Jacobi algorithm for scalar matrices a technique has been developed that replaces the full EVD step with a series of sparse Jacobi transformations. Using the cyclic-by-rows based EVD step the computational complexity of the SMD based PEVD algorithms is significantly reduced to the point where they outperform the SBR2 method in real time convergence.

3. Applications of PEVD Algorithms [13, 46, 47]

In addition to the developments around PEVD algorithms listed in the two items above various applications of PEVD algorithms have been considered. When testing PEVD algorithms one of the important criteria is the properties of the polynomial matrices fed into the algorithms. Varying the properties of the input polynomial matrices can have a significant impact on algorithm performance; here a number of parameters are varied and the results are analysed. An example use for PEVD algorithms is in a broadband angle of arrival estimation scenario. Previously the SBR2 algorithm has been incorporated into a polynomial version of the well known MUSIC algorithm. Now the MSME-SMD algorithm is brought into the polynomial MUSIC framework and compared to the implementation with SBR2. Besides direct EVD applications, the PEVD can also be utilised to implement a wider class of linear algebraic operations on polynomial matrices. This includes the polynomial SVD, and in this thesis has specifically targeted a polynomial version of the generalised eigenvalue decomposition. The generalised eigenvalue decomposition (GEVD) provides a powerful tool in the narrowband domain for low rank approximation of covariance matrices. Whereas the EVD

is applied to a single covariance matrix the GEVD is applied to two (or more) covariance matrices and creates generalised eigenvectors which diagonalise both. Here the Cholesky-based approach for calculating the GEVD of scalar matrices is extended for polynomial matrices.

4. **Matlab Based PEVD Toolbox** <http://pevd-toolbox.eee.strath.ac.uk/> [48]

An additional contribution that is not discussed in this thesis has been the development of a Matlab based PEVD toolbox. The toolbox includes functions to visualise and manipulate polynomial matrices, plus the SMD and SBR2 PEVD algorithms. The aim of the toolbox is to provide a starting point for anyone interested in applying the PEVD to new applications.

1.3 Overview

To elaborate on these contributions this thesis is laid out as follows: Chapter 2 provides background material upon which the following three contribution chapters will build. First scalar matrices and the scalar EVD are reviewed and some of their key characteristics are highlighted. Next polynomial matrices and some of their properties are introduced before discussing the PEVD and some of the details associated with it. Chapter 2 is concluded with a detailed look at the SBR2 and SMD PEVD algorithms followed by some of the ambiguities associated with the PEVD.

Chapter 3 investigates multiple shift PEVD algorithms. First an exhaustive search is developed which brings the maximum amount of energy onto the diagonal at each PEVD iteration. Due to its brute force approach, the exhaustive search is too computationally expensive for most PEVD applications. Using the idea of shifting multiple row/column pairs a significantly lower cost maximum element search based method is developed. The maximum element search means the search technique can be easily adapted to a causal method which generates a paraunitary matrix consisting of only delays. This chapter primarily focusses on multiple shift SMD algorithms but the idea has also been extended to the SBR2 algorithm to form MS-SBR2. The algorithms are

then tested in a variety of scenarios and the results are analysed. Two main issues arise, the rate of growth in the polynomial matrices (particularly for multiple shift algorithms) and the higher computational cost associated with the more powerful SMD type algorithms.

Given the issues that arise in Chapter 3, Chapter 4 aims to address these. First a new method for paraunitary matrix truncation is introduced based on one of the PEVD ambiguities highlighted in Chapter 2. The ambiguity means that row shifts can be applied to the paraunitary matrix without affecting the overall PEVD. The new truncation algorithm is then compared to its predecessor which it can outperform both in terms of reconstruction error and paraunitary matrix order. Next a simple modification to the multiple shift algorithm is proposed which can significantly reduce the growth in the polynomial matrices. Results show that the modification has a minimal impact on diagonalisation performance yet a significant reduction on the growth of the polynomial matrices produced. A side effect of lower growth is that the algorithm takes less time to complete as it is being applied to smaller matrices. The final part of Chapter 4 proposes an approximation for the costly full EVD step in the SMD algorithm. The approximation is based on the cyclic-by-rows Jacobi algorithm and is used to diagonalise the zero lag at each iteration. Results show that the approximation has almost no impact on the convergence of SMD based algorithms yet the real time performance is significantly improved.

Chapter 5 investigates the application of PEVD algorithms to various different scenarios. First the effects of variations in the input polynomial matrices is investigated. Here input polynomial matrices are generated using a source model that can be controlled to produce matrices with different dynamic ranges and spectral characteristics. Results are presented which show the impact of dynamic range and spectral characteristics on the SBR2, SMD and MSME-SMD algorithms. Next a broadband angle of arrival estimation scenario is considered, which has been addressed in the past using a polynomial version of the MUSIC algorithm. The polynomial MUSIC algorithm has previously used the SBR2 algorithm at its heart. In Chapter 5 the SBR2 algorithm is replaced with the MSME-SMD algorithm and their MUSIC results are compared for a

Chapter 1. Introduction

simple scenario. The final part of Chapter 5 introduces the scalar generalised eigenvalue problem for which a number of scalar approaches have been developed. Here one of the existing scalar methods is extended to a broadband, polynomial version using the SMD PEVD algorithm. Initial results show that the polynomial method can be used to diagonalise two parahermitian matrices in a similar fashion to the scalar approach.

Finally Chapter 6 presents conclusions for the thesis as a whole and discusses some potential future work.

Chapter 2

Background

This chapter covers the background material on the scalar EVD and its properties in Sec. 2.1, polynomial matrices and the polynomial EVD in Sec. 2.2 and Sec. 2.3. Ambiguities in the PEVD and truncation methods that exploit them are covered in Sec. 2.5 and a summary is given in Sec. 2.6. To introduce the concepts and properties associated with the scalar and polynomial EVDs an intuitive example is the multiple sensor array. Fig. 2.1 shows an array set up with a far field signal coming into the array at an angle. The signals $x_m[n], m = 1 \dots M$ are the signals received from each sensor in the array and $\mathbf{x}[n]$ is a vector containing the output from all sensors at sample index n .

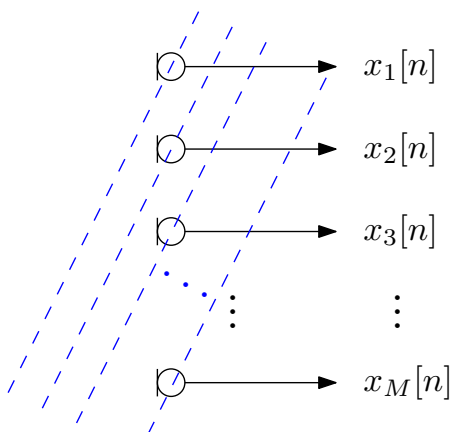


Figure 2.1: Far field signal arriving at sensor array from an angle, with sensor outputs $x_m[n], m = 1 \dots M$.

2.1 Scalar Matrices

2.1.1 Covariance Matrix

The $M \times M$ covariance matrix, \mathbf{R} , for the scenario in Fig. 2.1 is calculated as

$$\mathbf{R} = \mathcal{E}\{\mathbf{x}[n]\mathbf{x}^H[n]\} \quad (2.1)$$

where $\mathbf{x}[n]$ is a vector containing the output from the array sensors at sample index n and $\{\cdot\}^H$ is the conjugate or Hermitian transpose which combines the complex conjugate and transpose operations. The covariance matrix \mathbf{R} is a Hermitian matrix i.e. equal to its Hermitian transpose such that $\mathbf{R} = \mathbf{R}^H$, which for a real matrices results in a matrix that is symmetrical about the diagonal. In addition Hermitian matrices also have real eigenvalues, orthogonal eigenvectors, and can be diagonalised by a unitary matrix [49].

2.1.2 Scalar EVD

Taking the eigenvalue decomposition of the covariance matrix \mathbf{R} results in

$$\mathbf{R} = \mathbf{Q}\mathbf{D}\mathbf{Q}^H, \quad (2.2)$$

where \mathbf{Q} is a unitary matrix containing the eigenvectors, and \mathbf{D} is a diagonal matrix containing the eigenvalues. A unitary matrix is often referred to as an energy preserving matrix and when combined with its Hermitian transpose it gives the identity i.e. $\mathbf{Q}\mathbf{Q}^H = \mathbf{I}$, where \mathbf{I} is the $M \times M$ identity matrix.

Various algorithms for computing the eigenvalue decomposition exist [7] but two of the simplest (iterative) approaches generate the unitary \mathbf{Q} matrix through a series of orthogonal similarity transformations [7]. Each similarity transformation is used to selectively zero an off-diagonal element of the Hermitian matrix. The similarity

transformation used is the Jacobi transformation

$$\mathbf{Q}^{(i)} = \begin{bmatrix} \mathbf{I}_1 & & & & \\ & \cos \varphi^{(i)} & \dots & e^{j\vartheta^{(i)}} \sin \varphi^{(i)} & \\ & \vdots & \mathbf{I}_2 & \vdots & \\ -e^{-j\vartheta^{(i)}} \sin \varphi^{(i)} & \dots & \cos \varphi^{(i)} & & \\ & & & & \mathbf{I}_3 \end{bmatrix} \quad (2.3)$$

where the dimensions of the identity matrices (\mathbf{I}_1 , \mathbf{I}_2 and \mathbf{I}_3) are determined by the location of the elements to be zeroed. Starting from $\mathbf{S}^{(0)} = \mathbf{R}$, at each iteration $\mathbf{S}^{(i)}$ is computed as $\mathbf{S}^{(i)} = \mathbf{Q}^{\text{H}(i)} \mathbf{S}^{(i-1)} \mathbf{Q}^{(i)}$. The rotation parameters $\varphi^{(i)}$ and $\vartheta^{(i)}$ are based on the off-diagonal elements $s_{np}^{(i-1)}$ and $s_{pn}^{(i-1)}$ to be eliminated, and on the diagonal elements $s_{nn}^{(i-1)}$ and $s_{pp}^{(i-1)}$ [3, 7]. To zero the off-diagonal elements the parameters are set to

$$\vartheta^{(i)} = \arg(s_{np}^{(i-1)}) \quad , \quad (2.4)$$

and

$$\varphi^{(i)} = \frac{1}{2} \tan^{-1} \left(\frac{2|s_{np}^{(i-1)}|}{s_{nn}^{(i-1)} - s_{pp}^{(i-1)}} \right) \quad , \quad (2.5)$$

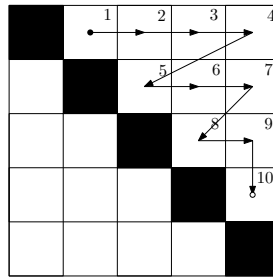
where $\varphi^{(i)}$ is chosen to be the four quadrant arctangent as it generally leads to the eigenvalues being ordered in decreasing power [3]. The unitary matrix containing the eigenvectors, \mathbf{Q} , is then simply the product of each of the I Jacobi transformations used

$$\mathbf{Q} = \mathbf{Q}^I \dots \mathbf{Q}^1 \quad . \quad (2.6)$$

The number of Jacobi transformations, I , depends on which of the following algorithms is used.

Classical Jacobi Algorithm

The classical Jacobi algorithm uses a maximum element search to find the next element to be zeroed using the Jacobi transformation from (2.3). The maximum element search identifies the row and column indices of the desired element and these are used to

Figure 2.2: Cyclic-by-row Jacobi sweep of a 5×5 matrix

determine the parameters in (2.3). The classical Jacobi algorithm continues until either the maximum off-diagonal element or the off-diagonal energy falls below a pre-defined threshold [7]. Although each Jacobi transformation is designed to zero the target element some energy will be moved into the adjacent row and column. It will therefore be necessary to perform multiple passes in order to sufficiently suppress the off-diagonal energy of the entire matrix. The SBR2 PEVD algorithm extends the classical Jacobi algorithm to polynomial matrices [3], and will be discussed further in Sec.2.4.

Cyclic-by-Rows Jacobi Algorithm

Rather than using a maximum element search the cyclic-by-rows algorithm performs a sweep of all off-diagonal elements. The sweep consists of $(M^2 - M)/2$ Jacobi transformations, which are systematically applied to the off-diagonal elements in a row-wise fashion as shown in Fig. 2.2. Like the classical approach it will be necessary to perform multiple sweeps of the matrix to sufficiently reduce the off-diagonal energy. Mitigating the maximum element search means that the cyclic-by-rows approach is lower cost than the classical Jacobi algorithm however it may require more Jacobi transformations [7]. Due to the added lag dimension in the polynomial matrices the number of sweeps would increase quite dramatically and therefore the cyclic-by-rows algorithm has not been directly extended to a PEVD algorithm. Rather than applying it to the entire polynomial matrix a method is developed in Chapter. 4 which utilises the cyclic-by-rows algorithm on part of the polynomial matrix.

2.2 Polynomial Matrices

Polynomial matrices arise in a number of different situations, a pertinent example is the case of broadband sensor arrays. Polynomial matrices can be envisaged as either a matrix with polynomial entries or a polynomial with matrix valued coefficients. In this case the polynomials are Laurent polynomials which include both positive and negative powers [50, 51].

2.2.1 Space-Time Covariance Matrix

The (broadband) space-time covariance matrix, $\mathbf{R}[\tau]$, for the scenario in Fig. 2.1 is obtained as

$$\mathbf{R}[\tau] = \mathcal{E}\{\mathbf{x}[n]\mathbf{x}^H[n - \tau]\} \quad , \quad (2.7)$$

rather than a single time instant as in 2.1, $\mathbf{R}[\tau]$ is evaluated at multiple time lags. Therefore $\mathbf{R}[\tau]$ is a matrix of auto- and cross-correlation sequences with a symmetry such that $\mathbf{R}[\tau] = \mathbf{R}^H[-\tau]$.

z -Transform

From the matrix $\mathbf{R}[\tau]$ the polynomial matrix, $\mathbf{R}(z)$, is obtained by taking the z -transform,

$$\mathbf{R}(z) = \sum_{\tau=t_1}^{t_2} \mathbf{R}[\tau]z^{-\tau} \quad , \quad (2.8)$$

which is a Laurent polynomial in z . In the z -domain filtering and convolution operations are simplified to the product of two polynomials. The Fourier transform can be obtained by evaluating the z -transform on the unit circle, i.e. when $z = e^{j\Omega}$.

2.2.2 Properties of Polynomial Matrices

Order and Degree

The dimension of $\mathbf{R}(z)$ is $M \times M$, with the polynomial order dependent on the support of the auto- and cross-correlation sequences contained within it. For the example in (2.8) the order is $(t_2 - t_1)$, in this particular scenario $|t_1| = |t_2|$ however in general this

is not the case. The degree of the polynomial matrix is defined as the number of delay elements required to implement it as an FIR filter therefore it is only defined for causal polynomial matrices [5]. The degree is quite different to the order of a polynomial matrix and to avoid confusion degree will not be discussed any further in this thesis.

Parahermitian Transpose

Also referred to as the paraconjugate, the parahermitian transpose is the extension of the Hermitian transpose from scalar matrices to polynomial matrices. In addition to the transpose and complex conjugation the parahermitian transpose also involves a time reversal of the polynomial elements, i.e. $\tilde{\mathbf{R}} = \mathbf{R}^H(1/z^*)$ where $\{\tilde{\cdot}\}$ is used to represent the parahermitian transpose. In the case where the polynomial matrix is of order zero the parahermitian transpose is equivalent to the Hermitian transpose.

Parahermitian Property

The parahermitian property is the polynomial matrix equivalent of the Hermitian property from scalar matrices. A polynomial matrix is parahermitian if it is equal to its parahermitian transpose, i.e. $\tilde{\mathbf{R}} = \mathbf{R}^H(1/z^*) = \mathbf{R}(z)$. An implication of this is that from (2.8) $|t_1| = |t_2|$ i.e. it must have the same number of positive and negative lags. Again in the case of an order zero polynomial matrix the parahermitian property is equivalent to the scalar Hermitian property. Taking the z-transform of a space-time covariance matrix, with its time reversed conjugate symmetry, results in a parahermitian matrix.

Paraunitary Property

When the idea of unitary, energy preserving, matrices is extended to polynomial matrices they are referred to as paraunitary matrices [5]. Extending the meaning of unitary matrices to the polynomial domain, a polynomial matrix is paraunitary if when applied to its parahermitian transpose it gives the identity, i.e. $\mathbf{Q}(z)\tilde{\mathbf{Q}}(z) = \mathbf{I}$ where \mathbf{I} is an $M \times M$ (scalar) identity matrix. As with the two preceding properties, an order zero paraunitary matrix is a unitary matrix.

2.3 Polynomial EVD

Like in the narrowband case the eigenvalue decomposition (EVD) provides a powerful tool for analysis of such a system. In the case of a broadband, parahermitian, system we must use a polynomial-matrix EVD (PEVD) [3]:

$$\mathbf{R}(z) \approx \tilde{\mathbf{Q}}(z)\mathbf{D}(z)\mathbf{Q}(z) \quad , \quad (2.9)$$

where $\mathbf{Q}(z)$ and $\mathbf{D}(z)$ contain the approximate polynomial eigenvectors and eigenvalues respectively. The polynomial matrix $\mathbf{Q}(z)$ is paraunitary and $\mathbf{D}(z)$ is diagonal

$$\mathbf{D}(z) = \text{diag}\{D_0(z) D_1(z) \dots D_{M-1}(z)\} \quad . \quad (2.10)$$

In addition to being diagonal the polynomial eigenvalues in $\mathbf{D}(z)$ are ordered such that

$$D_{m+1}(e^{j\Omega}) \geq D_m(e^{j\Omega}), \quad \forall \Omega, \quad m = 1 \dots M - 1 \quad . \quad (2.11)$$

This strict ordering of the eigenvalues at every frequency is referred to as spectral majorisation [5]. Two examples of spectral majorisation are shown in Fig. 2.3. The first scenario in Fig. 2.3 (a) shows the case of non-overlapping sources and the spectra of the eigenvalues are nice and smooth. In the second example, Fig. 2.3 (b), the sources are overlapping but the polynomial eigenvalues still enforce the spectral majorisation property resulting in the sharp corners in the spectra. In an array processing context, the spectra of the eigenvalues is directly influenced by the spectrum of each of the source signals impinging on the array. Even if the source signals are not spectrally ordered, as in (2.11) and Fig. 2.3 (b), the PEVD algorithms described in the following section will generally achieve spectral majorisation in their decompositions. A second important property of the source spectra is the dynamic range or the difference between the highest and lowest power spectral densities (PSDs). In both Fig. 2.3 (a) and (b) the dynamic range is 10 dB. The first part of Chapter 5 investigates the impact of applying PEVD algorithms to sources with different types of majorisation and dynamic ranges.

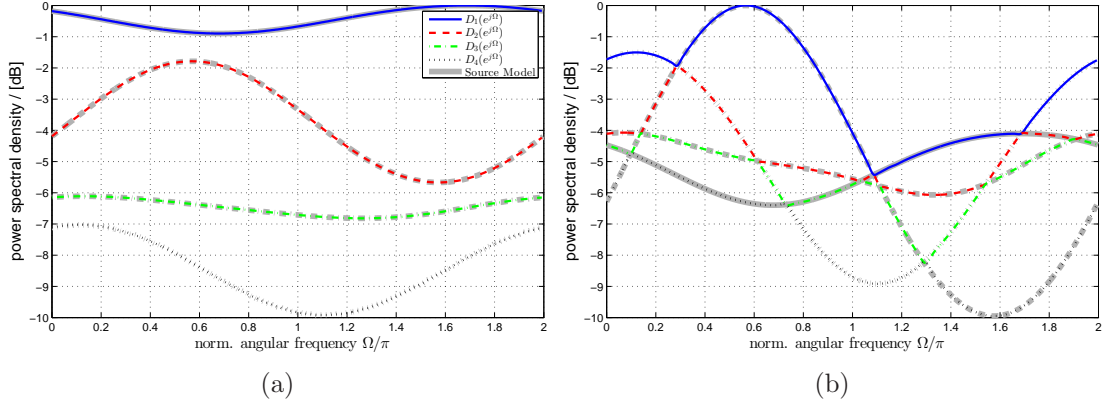


Figure 2.3: Examples of spectral majorisation for (a) non-overlapping sources and (b) overlapping sources.

2.4 PEVD Algorithms

The i th iteration of each of the established iterative PEVD algorithms [3, 10] starts with a search step that is algorithm-dependent and returns a column and lag index, $k^{(i)}$ and $\tau^{(i)}$ respectively. Based on the parameters $k^{(i)}$ and $\tau^{(i)}$ the shift matrix $\mathbf{\Lambda}^{(i)}(z)$ is constructed as

$$\mathbf{\Lambda}^{(i)}(z) = \text{diag}\{\underbrace{1 \dots 1}_{k^{(i)}-1} z^{-\tau^{(i)}} \underbrace{1 \dots 1}_{M-k^{(i)}}\}, \quad (2.12)$$

which will delay the $k^{(i)}$ th row of a polynomial matrix by $\tau^{(i)}$ samples. Applying $\mathbf{\Lambda}^{(i)}(z)$ to the partially diagonalised parahermitian matrix from the previous iteration, $\mathbf{S}^{(i-1)}(z)$, gives

$$\mathbf{S}^{(i)'}(z) = \mathbf{\Lambda}^{(i)}(z)\mathbf{S}^{(i-1)}(z)\tilde{\mathbf{\Lambda}}^{(i)}(z), \quad i = 1 \dots I, \quad (2.13)$$

where $\mathbf{S}^{(i)'}(z)$ is used to denote an intermediary parahermitian matrix. The final step of each PEVD algorithm iteration transfers off-diagonal energy from the zero lag onto the diagonal using

$$\mathbf{S}^{(i)}(z) = \mathbf{Q}^{(i)}\mathbf{S}^{(i)'}(z)\mathbf{Q}^{(i)H}. \quad (2.14)$$

where $\mathbf{Q}^{(i)}$ is a unitary matrix based on the off-diagonal energy of the zero lag and gets applied to all lags of $\mathbf{S}^{(i)'}(z)$.

The stopping criterion for these PEVD algorithms can either be a predetermined number of iterations, I , or based on the energy found in the search step. Once the algorithm has completed I iterations, the diagonal matrix $\mathbf{D}(z)$ from (2.9) is approximated as

$$\mathbf{D}(z) \approx \mathbf{S}^{(I)}(z) \quad . \quad (2.15)$$

The paraunitary matrix in (2.9) can be assembled via a series of simpler paraunitary matrices,

$$\mathbf{Q}(z) \approx \mathbf{G}^{(I)}(z) \dots \mathbf{G}^{(1)}(z) \quad , \quad (2.16)$$

where the simple paraunitary matrix $\mathbf{G}^{(i)}(z)$ is constructed from the delay and unitary operations from the i th step,

$$\mathbf{G}^{(i)}(z) = \mathbf{Q}^{(i)} \mathbf{\Lambda}^{(i)}(z) \quad . \quad (2.17)$$

Thus, the PEVD is approximated with (2.15) and (2.16).

2.4.1 SBR2

Based on a modified column vector $\hat{\mathbf{s}}_k^{(i)}[\tau] \in \mathbb{C}^{M-1}$ containing all elements in the $k^{(i)}$ th column of $\mathbf{S}^{(i)}[\tau]$ except for the diagonal element, the SBR2 parameter set for (2.12) is determined by the maximum off-diagonal element search

$$\{m^{(i)}, k^{(i)}, \tau^{(i)}\} = \arg \max_{m,k,\tau} \|\hat{\mathbf{s}}_k^{(i-1)}[\tau]\|_\infty \quad , \quad (2.18)$$

where in addition to $k^{(i)}$ and $\tau^{(i)}$ the row index, $m^{(i)}$, of the maximum element is also obtained. For the SBR2 algorithm the unitary matrix, $\mathbf{Q}^{(i)}$, from the diagonalisation step in (2.14) is constructed from a scalar Jacobi rotation matrix in (2.3), where the rotation angles $\varphi^{(i)}$ and $\vartheta^{(i)}$ are determined by the maximum element identified by the search in (2.18). The identity matrices \mathbf{I}_j , $j = 1, 2, 3$, in (2.3) have dimensions $(\min\{m^{(i)}, k^{(i)}\} - 1)$, $(|m^{(i)} - k^{(i)}| - 1)$ and $(M - \max\{m^{(i)}, k^{(i)}\} + 1)$, respectively. The resulting unitary matrix $\mathbf{Q}^{(i)}$ has to be left- and right-multiplied to every lag matrix $\mathbf{S}^{(i)'}[\tau]$ according to (2.14). No full matrix multiplication is required, due to the sparse

structure of (2.3) only two rows and columns in $\mathbf{S}^{(i)'}(z)$ will be affected.

The convergence of SBR2 has been proven in [3] by showing that the paraunitary operations do not alter the total energy in $\mathbf{S}^{(i)'}(z)$, while in every step the off-diagonal energy is further minimised. The algorithm stops after I iterations, once the maximum off-diagonal element

$$\max_{k,\tau} \|\hat{\mathbf{s}}_k^{(I)}[\tau]\|_\infty < \rho \quad (2.19)$$

falls below a predetermined threshold ρ . Most recently SBR2 has been proven to enforce spectral majorisation of the polynomial eigenvalues as in (2.11) [52].

2.4.2 SMD Algorithms

Different from SBR2, in the i th iteration the SMD approach will not just eliminate the largest off-diagonal element but fully diagonalises $\mathbf{S}^{(i)}[0]$. In addition the SMD approach uses the initialisation

$$\mathbf{S}^{(0)}[0] = \mathbf{Q}^{(0)}\mathbf{R}[0]\mathbf{Q}^{(0)\text{H}} \quad , \quad (2.20)$$

with $\mathbf{Q}^{(0)}$ the modal matrix obtained from the scalar EVD of $\mathbf{R}[0]$, every subsequent iteration brings one row and column to $\mathbf{S}^{(i)'}[0]$, whose energy is then transferred onto the diagonal by a scalar EVD.

As the SMD algorithm can diagonalise more than a single element the search step has been modified to make the most of the energy transfer. To maximise the reduction in off-diagonal energy, the SMD parameter selection in the i th iteration is

$$\{k^{(i)}, \tau^{(i)}\} = \arg \max_{k,\tau} \|\hat{\mathbf{s}}_k^{(i-1)}[\tau]\|_2 \quad , \quad (2.21)$$

which differs from (2.18) in the use of the l_2 instead of the l_∞ norm. To achieve complete diagonalisation of the zero lag, like the initialisation step $\mathbf{Q}^{(i)}$ is the modal matrix obtained by the scalar EVD of $\mathbf{S}^{(i)'}[0]$.

The convergence of SMD is proven in [10], with a stopping criterion similar to (2.19) but based on the l_2 norm according to (2.21). SMD has been shown to diagonalise

paraHermitian matrices with a lower number of iterations than SBR2, because more energy is transferred from off-diagonal to on-diagonal elements. However, an EVD has to be calculated at every iteration, and the modal matrix $\mathbf{Q}^{(i)}$ no longer has the sparse structure of the Jacobi transformation in (2.3); requiring a full matrix multiplication at every lag. Despite this the SMD algorithm will generally produce paraunitary matrices of lower order than SBR2. Therefore, for applications such as those requiring broadband signal subspace decompositions, SMD permits better performance with lower order paraunitary filter banks than SBR2.

The column norm search used by SMD transfers more energy than a maximum element search; in Chapter. 3 a new multiple shift search method is developed which aims to maximise the reduction in off-diagonal energy at each iteration. One of the fundamental drawbacks of the SMD algorithm is the additional time required to apply the non-sparse $\mathbf{Q}^{(i)}$ at each iteration. In Chapter. 4 the SMD approach is modified by substituting the EVD step for a number of sparse Jacobi transformations applied in a cyclic-by-rows fashion.

ME-SMD

An alternative version of the SMD algorithm, also introduced in [10], utilises the maximum element search in (2.18) and is termed the maximum element (ME) SMD algorithm. All other steps follow those of the SMD algorithm above with the intention of ME-SMD being a lower cost alternative. The maximum element search means that ME-SMD will transfer less energy at each iteration than SMD with its column norm search. Compared to SBR2 the ME-SMD algorithm will generally transfer more energy for a given iteration but like SMD each iteration will be more computationally costly than SBR2.

2.5 PEVD Ambiguity

The polynomial eigenvalue decomposition and the algorithms for calculating it introduced in Sec. 2.3 and Sec. 2.4 have within them a number of different ambiguities which

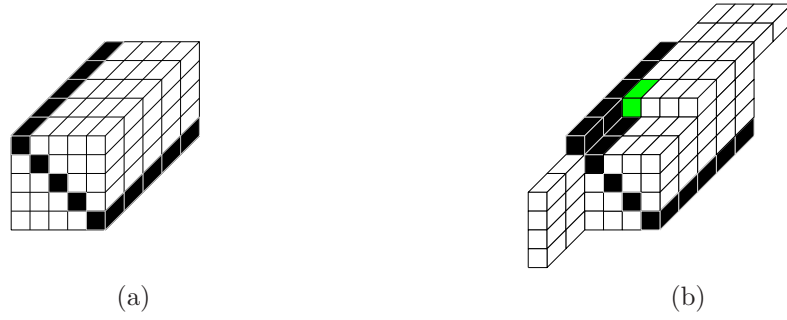


Figure 2.4: Initial parahermitian matrix (a) and result of delaying row 1 (b).

are discussed in the following subsections.

2.5.1 Shift Matrix Ambiguity

In the case of single shift algorithms discussed in Sec. 2.4 the ambiguity in the shift matrix $\mathbf{\Lambda}^{(i)}(z)$ is not particularly useful however it is a key part of the multiple shift algorithms in Chapter 3. The ambiguity in the shift matrix allows an arbitrary delay, such as $z^{-\tau}$, to be applied to the shift matrix without affecting the resultant parahermitian matrix; in short $\mathbf{\Lambda}^{(i)}(z) = z^{-\tau} \mathbf{\Lambda}^{(i)}(z)$.

Fig. 2.4 (a) shows a visual representation of a $5 \times 5 \times 5$ parahermitian matrix with the diagonal elements shown in black. A simple example of the shift matrix ambiguity is that delaying the first row of the parahermitian matrix in Fig. 2.4 (a) by 2 (using $\mathbf{\Lambda}^{(i)}(z) = \text{diag}\{z^{-2} \ 1 \ 1 \ 1 \ 1\}$), shown in Fig. 2.4 (b) gives the exact same result as advancing all other rows by 2 (using $z^2 \mathbf{\Lambda}^{(i)}(z) = \text{diag}\{1 \ z^2 \ z^2 \ z^2 \ z^2\}$), shown in Fig. 2.5 (a) – (d). Therefore $\mathbf{\Lambda}^{(i)}(z)$ has an infinite number of equivalent shift matrices, $z^{-\tau} \mathbf{\Lambda}^{(i)}(z) \forall \tau$. Although all shift matrices will be equivalent, many of them will add redundancy to the resultant paraunitary matrix as its order will be higher than is required e.g. $z^{\tau} \mathbf{\Lambda}^{(i)}(z) = \text{diag}\{z^{\tau-2} \ z^{\tau} \ z^{\tau} \ z^{\tau} \ z^{\tau}\}$ for $\tau > 2$ or $\tau < 0$.

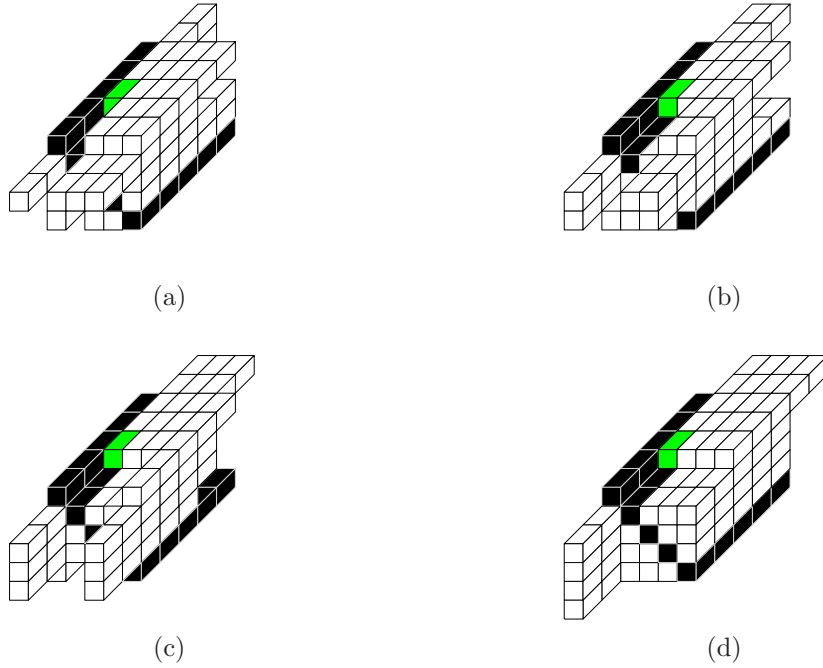


Figure 2.5: Results of advancing rows: 2 (a), 3 (b), 4 (c) and 5 (d).

2.5.2 Paraunitary Ambiguity

Assuming that for a parahermitian $\mathbf{R}(z)$, (2.9) holds with equality, we ask whether a second decomposition

$$\mathbf{R}(z) = \tilde{\mathbf{Q}}(z)\mathbf{D}(z)\mathbf{Q}(z) = \tilde{\tilde{\mathbf{Q}}}(z)\bar{\mathbf{D}}(z)\bar{\mathbf{Q}}(z) \quad (2.22)$$

can be found. With diagonalisation and spectral majorisation of $\mathbf{D}(z)$ providing uniqueness [31], it follows that $\bar{\mathbf{D}}(z) = \mathbf{D}(z)$. Hence, writing $\bar{\mathbf{Q}}(z) = \mathbf{\Gamma}(z)\mathbf{Q}(z)$, the modifying matrix $\mathbf{\Gamma}(z)$ must be paraunitary, diagonal and contain allpass filters in order to not affect $\mathbf{D}(z)$. While for general allpass filters either $\mathbf{\Gamma}(z)$ or $\tilde{\mathbf{\Gamma}}(z)$ can be unstable, a simple selection

$$\mathbf{\Gamma}(z) = \text{diag}\{z^{-\tau_1} \ z^{-\tau_2} \ \dots \ z^{-\tau_M}\} \quad (2.23)$$

is possible. This shifts the m th row of $\mathbf{Q}(z)$ by τ_m samples, where $m = 1 \dots M$ and M is the spatial dimension of $\mathbf{R}(z)$. A similar paraunitary ambiguity has been stated in [53] and [32] but has never been exploited.

Therefore, even if the diagonal $\mathbf{D}(z)$ is unique, a paraunitary matrix is ambiguous as a $\mathbf{Q}(z)$ of minimum order can be modified by row-shifts to $\tilde{\mathbf{Q}}(z)$ applied by $\mathbf{\Gamma}(z)$ to yield a factorisation with identical $\{\mathbf{R}(z), \mathbf{D}(z)\}$. The case where $\tau_1 = \tau_2 = \dots = \tau_M$ has previously been used for truncation in [3, 54] and is a natural extension of the shift matrix ambiguity in Sec. 2.5.1 – this will be discussed further in Sec. 2.5.3 and Sec. 4.1. The following example shows how the delays affect a diagonal matrix with $M = 3$ and $\mathbf{\Gamma}(z) = \text{diag}\{z^{\tau_1} \ z^{\tau_2} \ z^{\tau_3}\}$

$$\begin{aligned}
 \mathbf{R}(z) &= \tilde{\mathbf{Q}}(z)\bar{\mathbf{D}}(z)\bar{\mathbf{Q}}(z) \\
 &= \tilde{\mathbf{Q}}(z)\tilde{\mathbf{\Gamma}}(z)\bar{\mathbf{D}}(z)\mathbf{\Gamma}(z)\mathbf{Q}(z) \\
 &= \tilde{\mathbf{Q}}(z)\text{diag}\{z^{-\tau_1} \ z^{-\tau_2} \ z^{-\tau_3}\}\text{diag}\{\bar{D}_1(z) \ \bar{D}_2(z) \ \bar{D}_3(z)\}\text{diag}\{z^{\tau_1} \ z^{\tau_2} \ z^{\tau_3}\}\mathbf{Q}(z) \\
 &= \tilde{\mathbf{Q}}(z)\text{diag}\{z^{-\tau_1}\bar{D}_1(z)z^{\tau_1} \ z^{-\tau_2}\bar{D}_2(z)z^{\tau_2} \ z^{-\tau_3}\bar{D}_3(z)z^{\tau_3}\}\mathbf{Q}(z) \\
 &= \tilde{\mathbf{Q}}(z)\bar{\mathbf{D}}(z)\mathbf{Q}(z) \quad .
 \end{aligned} \tag{2.24}$$

Despite being in the background chapter, this is one of the contributions of this thesis and will be utilised in a novel paraunitary truncation approach [43] in Chapter 4.

2.5.3 Truncation Methods

The growth in polynomial order can be curtailed using appropriate parahermitian [55] and paraunitary [54] truncation 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 truncation is done symmetrically taking advantage of its parahermitian nature the outer most lags contain the same energy. The parahermitian property is preserved after the truncation function has been applied. In the case of paraunitary matrices the truncation function is applied to either end of $\mathbf{Q}(z)$ asymmetrically because the outer lags of a paraunitary matrix will have different energies. As a result of removing small values the paraunitary property is replaced by near-paraunitarity after the truncation function is applied; assuming $\mu > 0$. Removal of the lower order lags in the paraunitary matrix is possible due to the paraunitary

ambiguity where $\mathbf{\Gamma}(z)$ sees the same delays applied to all rows i.e. $\tau_1 = \tau_2 = \dots = \tau_M$ in (2.23).

To reduce computational costs of the PEVD algorithms the parahermitian truncation can be carried out at the end of every iteration, the resulting maximum total loss in energy after I iterations is $I \times \mu_{PH}$. As the paraunitary matrix is only ever calculated when the PEVD is complete the truncation function is only applied once and so the resulting energy loss has a maximum of μ_{PU} . In Chapter 4 a new method is proposed where the delays $(\tau_1, \tau_2, \dots, \tau_M)$ can take on different values; taking full advantage of the ambiguity from (2.24).

2.6 Chapter Summary

This chapter has introduced much of the background material required for the remainder of this thesis. Starting from the well known scalar EVD this chapter has shown how scalar concepts have been extended to polynomial matrices. Next the polynomial-matrix EVD (PEVD) was introduced followed by two of the established iterative PEVD algorithms. The SBR2 algorithm is a polynomial matrix extension of the classical Jacobi algorithm, and the SMD algorithm is a refinement upon SBR2 with modified search and diagonalisation steps. From the discussion on SMD two key areas for improvement were identified. The first area for improvement was the amount of energy being transferred at each iteration; for SMD this is better than SBR2 but it can be improved further using a novel method in Chapter 3. Another area for improvement was the cost of applying the non-sparse EVD modal matrix to the parahermitian matrix; in Chapter 4 this step is replaced with a cyclic-by-rows based approximation which results in a dramatic reduction in execution time. As the iterative PEVD algorithms proceed the polynomial dimension of the parahermitian and paraunitary matrices increase, within these decompositions some ambiguities have been identified which can impact the growth of the paraunitary matrix in particular. Based on the paraunitary matrix ambiguity identified in this chapter a new paraunitary matrix truncation algorithm is proposed in Chapter 4. In general the new truncation method has been shown

Chapter 2. Background

to truncate paraunitary matrices to a lower order and with a smaller error than its predecessor introduced at the end of this chapter.

Chapter 3

Multiple Shift PEVD Algorithms

This chapter aims to enhance the existing SMD algorithm by transferring more energy at each algorithm iteration. Rather than applying a single row/column shift, methods are developed here to shift multiple row/column pairs which will diagonalise more energy at each iteration. Here the main PEVD steps from Sec. 2.4 remain the same but the shift matrix, $\mathbf{\Lambda}^{(i)}(z)$, is modified to

$$\mathbf{\Lambda}^{(i)}(z) = \text{diag}\{1 \quad z^{-\tau^{(i,1)}} \quad \dots \quad z^{-\tau^{(i,M-1)}}\} \quad . \quad (3.1)$$

To obtain (3.1) the search stage of the PEVD algorithms is modified to identify multiple row/column pairs that can be brought onto the zero lag. In the SMD algorithm all zero lag energy is transferred onto the diagonal at each iteration, therefore a search method that maximises the energy transfer is advantageous. The aim of the following methods is to increase the energy transfer achieved by the PEVD methods and thereby speed up their real time convergence. First an exhaustive search for the SMD approach is proposed followed by approximations that have been shown to achieve a similar energy transfer with greatly reduced computational cost. In addition a multiple shift version of the SBR2 algorithm is also discussed which is based on the multiple shift SMD. Furthermore the idea of shifting multiple row/column pairs has been extended to a

polynomial QR decomposition in [34] and a similar concept is employed in the divide and conquer PEVD in [56].

3.1 Maximum Energy SMD

Idea

The maximum energy SMD approach finds the set of shifts for the delay matrix $\mathbf{\Lambda}^{(i)}(z)$ in (3.1) such that the maximum amount of off-diagonal energy is transferred onto the zero lag, $\mathbf{S}^{(i)'}[0]$, of the parahermitian matrix, $\mathbf{S}^{(i)'}(z)$, at the i th iteration. Below we focus on the i th iteration, and assume that $\mathbf{S}^{(i-1)}(z) \in \mathbb{C}^{M \times M}$ has a support of $2L + 1$, i.e. $\mathbf{S}^{(i-1)}[\tau] = \mathbf{0} \forall |\tau| > L$ where $\mathbf{S}^{(i-1)}(z)$ is the z -transform of $\mathbf{S}^{(i-1)}[\tau]$. The optimum $\mathbf{\Lambda}^{(i)}(z)$ can be determined via an exhaustive search which considers all possible shift combinations. An added complication of dealing with multiple shifts is the interaction between the row and column shifts which may move some energy away from the zero lag.

Exhaustive Search

An exhaustive search must consider the amount of energy brought onto the zero lag with every possible shift. In the single shift case this is fairly straightforward and is effectively what the SMD algorithm does with its column norm search strategy. When a multiple shift approach is used the only technique to measure the energy brought onto the zero lag is to apply every different shift combination; the number of combinations depends on both the matrix width M and polynomial dimension L .

In the case of a single shift algorithm, the maximum shift length, Δ_{\max} , is L . Under multiple shifts the movement of one row/column pair will affect other rows and columns and so the maximum shift, Δ_{\max} , becomes $\lceil (M - 1)L/2 \rceil$. The origin of $\Delta_{\max} = \lceil (M - 1)L/2 \rceil$ will be explained in detail in Sec. 4.2

Varying each of the M diagonal elements of $\mathbf{\Lambda}^{(i)}(z)$ in (3.1) over the interval $\tau \in [-\Delta_{\max}, \Delta_{\max}]$ will lead to $(2\Delta_{\max} + 1)^M$ possible shift combinations. This is visualised in Fig. 3.1 for the simple case where $M = 4$ and $L = 1$, with the m th row representing

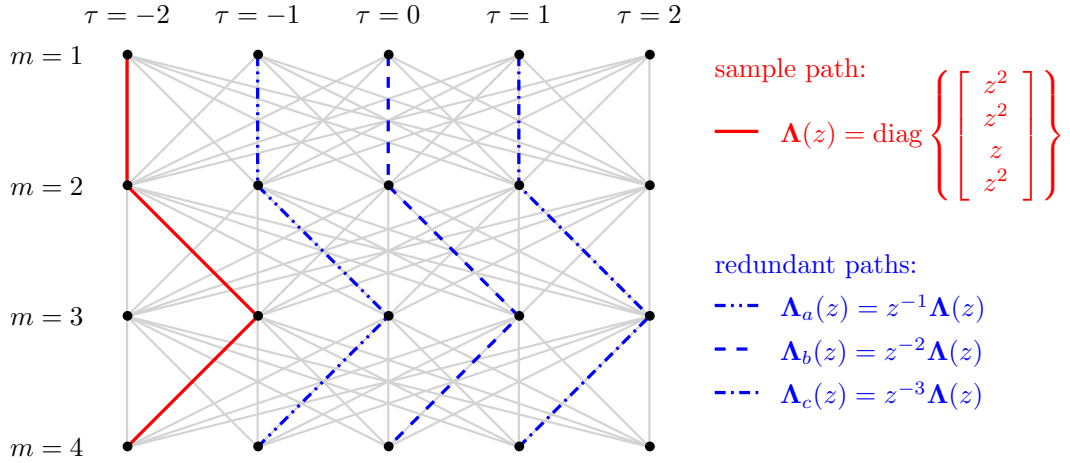


Figure 3.1: Visualisation of possible delay matrices for $M = 4$ and $L = 1$; each top-bottom path in the $(2\Delta_{\max} + 1) \times M$ trellis defines the parameters of a possible delay matrix; one specific matrix is highlighted in red, with redundant matrices for horizontally shifted paths.

all possible values for $\tau^{(i,m)}$ in (3.1). Each path from top to bottom represents one particular combination of shifts, with a total number of $(2\Delta_{\max} + 1)^M$ possibilities. For larger values of M or L , the diagram in Fig. 3.1 expands vertically or horizontally, respectively.

Of the established $(2\Delta_{\max} + 1)^M$ possible shift combinations a number of combinations are redundant; as explained in Sec. 2.5.1 $z^{-\tau} \mathbf{\Lambda}^{(i)}(z)$ with τ chosen arbitrarily will implement a shift that is identical to $\mathbf{\Lambda}^{(i)}(z)$. As an example for the case of $\mathbf{S}^{(i-1)}[\tau] \in \mathbb{C}^{4 \times 4}$ with $L = 1$, Fig. 3.1 highlights three redundant combinations that yield the same results as $\mathbf{\Lambda}^{(i)}(z) = \text{diag}\{z^2 \ z^2 \ z \ z^2\}$. Selecting one path, its redundant copies can be identified by strict horizontal shifts within the trellis, which correspond to an overall delay or advance encapsulated by $z^{-\tau}$.

Defining \mathcal{S} as the set of shift combinations, the following theorem states its cardinality:

Theorem 1 (Cardinality of \mathcal{S}). *For a parahermitian matrix $\mathbf{S}^{(i-1)}(z) \in \mathbb{C}^{M \times M}$ of order $2L$, such that $\mathbf{S}^{(i-1)}[\tau]$ with a support of $2L + 1$, the set \mathcal{S} of independent shift combinations that an exhaustive search algorithm has to evaluate has the cardinality*

$$|\mathcal{S}| = (2\Delta_{\max} + 1)^M - (2\Delta_{\max})^M \quad . \quad (3.2)$$

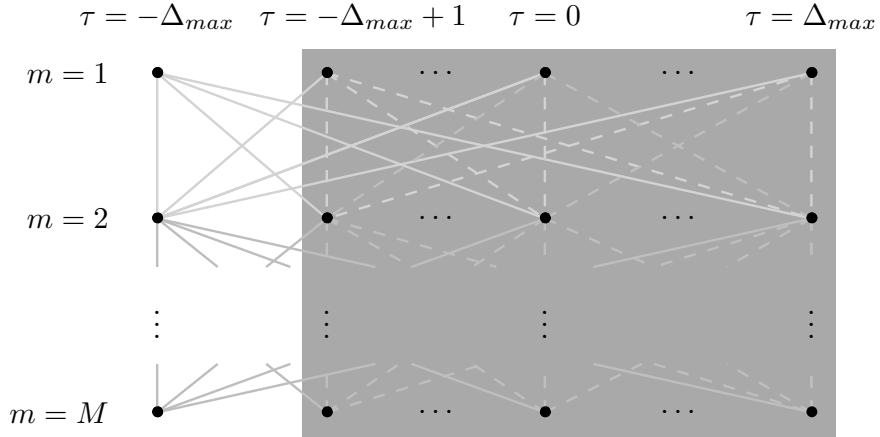


Figure 3.2: Trellis of paths representing possible shift matrices. Redundant shift matrices according to the definition of (3.4), which do not involve the left-most column with $\tau = -L$, have $(2\Delta_{\max})^M$ paths confined to the shaded area.

Proof. We define a redundant delay matrix $\mathbf{\Gamma}^{(i)}(z)$ as one that can be obtained by delaying a genuine delay matrix $\mathbf{\Lambda}^{(i)}(z) \in \mathcal{S}$, such that $\mathbf{\Gamma}^{(i)}(z) = z^{-\Delta}\mathbf{\Lambda}^{(i)}(z)$ with $\Delta > 0$. W.r.t. the sample trellis in Fig. 3.1, the path belonging to the parameter set of $\mathbf{\Lambda}^{(i)}(z)$ has to include at least one node in the left-most column of the trellis, i.e. if $\mathbf{\Lambda}^{(i)}(z)$ is constructed according to (3.1),

$$\mathbf{\Lambda}^{(i)}(z) \in \mathcal{S} \iff \min_m \tau^{(i,m)} = -\Delta_{\max} \quad . \quad (3.3)$$

Therefore, a redundant matrix $\mathbf{\Gamma}^{(i)}(z)$ — also constructed according to (3.1) — is characterised by not reaching any node in the left-most column, or

$$\mathbf{\Gamma}^{(i)}(z) \notin \mathcal{S} \iff \min_m \tau^{(i,m)} > -\Delta_{\max} \quad . \quad (3.4)$$

Since any path fulfilling (3.4) has to lie entirely within the shaded area in Fig. 3.2, occupying the $2\Delta_{\max}$ right-most columns of the trellis, there are $(2\Delta_{\max})^M$ paths belonging to redundant delay matrices. With a total possibility of $(2\Delta_{\max} + 1)^M$ combinations within the trellis, (3.2) is proven. \square

The cardinality $|\mathcal{S}|$ in (3.2) is important, as it restricts the exhaustive search and therefore limits its implementation cost. The definition of an independent delay matrix according to (3.3) is somewhat arbitrary, and its place in the set \mathcal{S} could also be taken

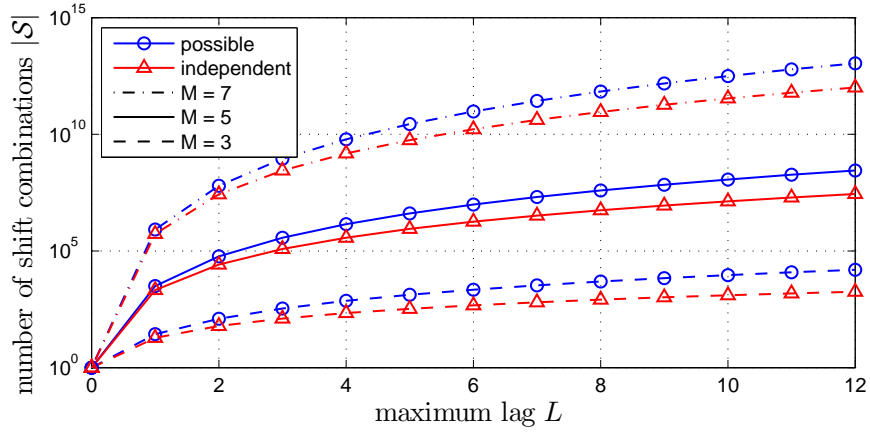


Figure 3.3: Number of shift combinations that an exhaustive search algorithm has to evaluate for $\mathbf{S}^{(i-1)}[\tau] \in \mathbb{C}^{M \times M}$ for $M = 3, 5, 7$ with support $2L + 1$.

by one of its redundant copies $z^{-\Delta} \mathbf{\Lambda}^{(i)}(z)$. This has no impact on $\mathbf{S}^{(i)'}(z)$ in (2.13) and therefore does not affect $\mathbf{D}(z)$ in (2.15). However, selecting a copy with minimum order in z^{-1} will influence the order of the paraunitary matrix $\mathbf{Q}(z)$ in (2.16), and a search algorithm would therefore have to operate with care to either keep the order increase low w.r.t. $\mathbf{Q}(z)$, or to ensure that any growth by trailing zero matrices is curtailed.

The size of the search space for delay matrices applicable to $\mathbf{S}^{(i-1)}[\tau] \in \mathbb{C}^{M \times M}$ is shown in Fig. 3.3 for the cases $M = 3, 5, 7$ and over a range of maximum lag values L . Both the number of possible as well as the independent number of shifts — those belonging to \mathcal{S} — are detailed. Even for a relatively small range of values of both M and L the number of shift combinations increases dramatically.

Implementation

The implemented algorithm first finds the list of all $(2\Delta_{\max} + 1)^M$ possible shifts, which is then pruned to remove the redundant $(2\Delta_{\max})^M$ delay operations to obtain \mathcal{S} . Within \mathcal{S} , the best possible shift combination for $\mathbf{S}^{(i-1)}[\tau] (\in \mathbb{C}^{M \times M}$ with support $2L + 1$) is then identified by applying each of the independent shifts to $\mathbf{S}^{(i-1)}[\tau]$.

According to Fig. 3.3, the complexity of the exhaustive search grows very fast for an increase in the spatial matrix dimension M , but also raising the lag dimension L has significant impact. In general the complexity is such that it is not possible to calculate a full maximum energy SMD algorithm, as it may reach large values of L during its

execution. Therefore, the maximum energy is only ever used as a benchmark for other PEVD search strategies.

3.2 Multiple Shift Maximum Element SMD

Although optimal in terms of energy transfer the maximum energy search is very computationally expensive and therefore impractical to implement as part of a full SMD algorithm. In this section we propose an approximation which relies on a significantly cheaper search method but maintains a similar high energy transfer through multiple shifts.

3.2.1 Idea

Rather than testing every possible shift combination this approach uses multiple (low cost) maximum element searches to generate the shift matrix in (3.1). The algorithm is therefore referred to as the multiple shift maximum element (MSME) SMD algorithm. Like the exhaustive search above the MSME-SMD algorithm aims to take advantage of the full EVD step in the SMD algorithm to transfer more energy at each iteration.

3.2.2 Algorithm

The initialisation of the proposed algorithm follows the SMD family with (2.20). At the i th iteration, (2.18) is first used to identify the maximum off-diagonal element, and time-shift it with its column onto the lag zero slice, resulting in the sparsity structure shown in Fig. 3.4 (a). By permuting this matrix to the structure in Fig. 3.4 (b), any subsequent operations within the i th iteration will not affect this maximum off-diagonal element as long as the upper 2×2 matrix remains untouched. Different strategies to identify and time-shift further columns within the i th iteration exist; the next three subsections outline three such approaches.

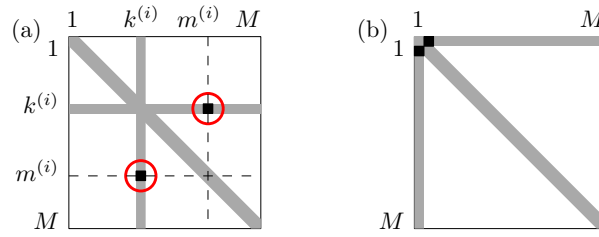


Figure 3.4: Sparsity structure of lag zero matrix $\mathbf{S}^{(i)'}[0]$ after i th iteration of (a) the SMD algorithm, indicating the maximum off-diagonal element in position $m^{(i)}$ of the $k^{(i)}$ th row, and (b) after permutation.

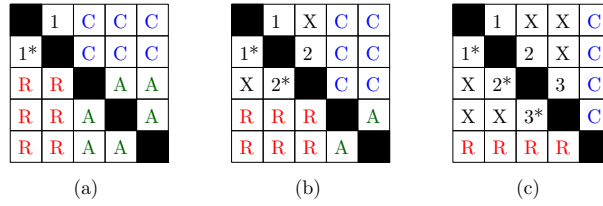


Figure 3.5: Best case scenario for the Naive “Greedy” method.

Naive “Greedy” Approach

Using a so-called “Greedy” approach the number of elements found is variable between $M - 1$ and $\lceil M/2 \rceil$, which is determined by the rows and columns that the elements are in and the order in which they are found. Fig. 3.5 shows the best case scenario for this method in a 5×5 example, neglecting the lag dimension i.e. elements in these positions can be taken from any lag. Assuming the first element pair is found in (or permuted to) the upper left off-diagonal position like Fig. 3.5 (a). The ‘C’s and ‘R’s indicate these elements must be shifted in their columns and rows respectively, the ‘A’s stand for ‘any’ and can be shifted either as rows or columns without affecting the first (or any previous) element. A best case scenario then finds the second element pair in the positions indicated by 2 and 2* in Fig. 3.5 (b) — some but not all elements can be permuted into this position. The ‘X’s now indicate elements that cannot be moved at all as they would in turn move either element 1 or 2. The third element pair is found in the positions (permutable to) 3 and 3* in Fig. 3.5 (c) and a final fourth element can be found in the remaining ‘C’ and ‘R’ positions.

Using the same annotations the worst case scenario is shown in Fig. 3.6. Again starting from the first element pair in the upper left most off-diagonal positions Fig. 3.6

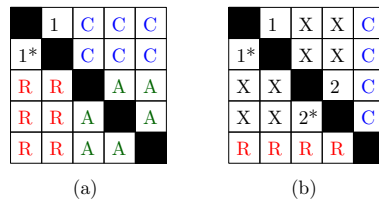


Figure 3.6: Worst case scenario for the “Greedy” method.

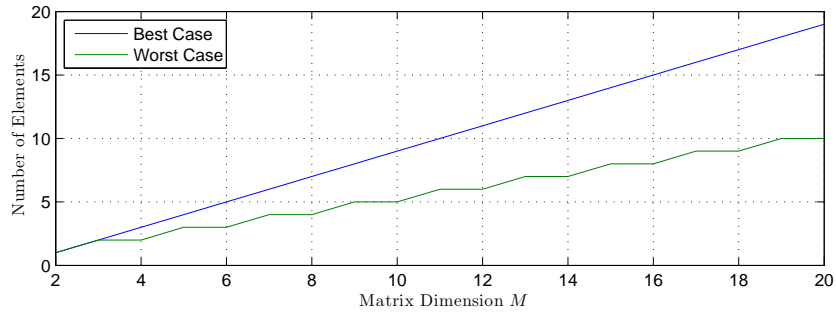


Figure 3.7: Difference between best and worst case multiple shift scenarios.

(a). This time a second element is chosen from one of the ‘A’ locations which cannot be permuted into the position of 2 in Fig. 3.5 (b). After the second element has been chosen only one further element can be picked from Fig. 3.6 (b), giving a total of 3 rather than 4 elements. For the simple 5×5 example shown there is only a difference of 1 element but as M grows this difference becomes more significant as shown in Fig. 3.7.

Maximum Number of Shifts

The only restriction of the “Greedy” approach is that previous elements are not moved by future shifts, this leads to a variable performance in terms of the number of shifts. From Fig. 3.5 and Fig. 3.6 it is possible to see that at certain stages some elements make better choices than others. To guarantee that the algorithm achieves the maximum number of shifts as in Fig. 3.5 the search spaces can be further restricted, using a mask, such that the m th element chosen can be permuted to the upper left $m + 1$ dimensional sub-matrix. The masking process is shown in Fig. 3.8 (a) where the ‘A’s from Fig. 3.5 (a) are replaced with ‘M’s to indicate that they are now masked from this search step. The algorithm is now forced to choose the better elements and the masking works such that no elements are missed in the course of one iteration i.e. any elements

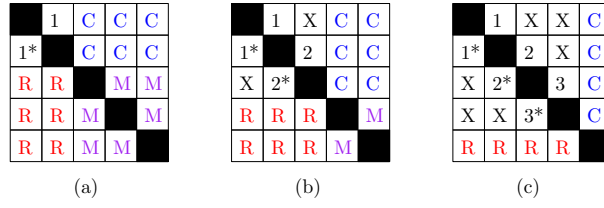


Figure 3.8: Masking used in the method that maximises the number of shifts.

masked during one search step will be available in future search steps (within the same iteration). Now any element found in either of the ‘C’ or ‘R’ sections in Fig. 3.8 (a) can be permuted into the upper left 3×3 sub-matrix (positions of the 2, 2*, or ‘X’s) in Fig. 3.8 (b); the same is true between Fig. 3.8 (b) and Fig. 3.8 (c). Using this masking procedure guarantees that $M - 1$ elements will be brought onto the zero lag at each iteration.

With the permutations used in Fig. 3.8, (3.1) can be redefined as a more complex delay & permutation matrix

$$\mathbf{\Lambda}^{(i)}(z) = \text{diag}\{1 \quad z^{-\tau^{(i,1)}} \quad \dots \quad z^{-\tau^{(i,M-1)}}\} \mathbf{P}^{(i)} \quad (3.5)$$

whereby the permutation matrix $\mathbf{P}^{(i)}$ accumulates all the column shift operations discussed above. The delays $\tau^{(i,m)}$, $m = 1 \dots (M - 1)$ are the lag values at which the maximum elements for the different columns in Fig. 3.8 have been found.

In reality the permutations described in Fig. 3.8 and (3.5) are not required, they are only used here to better illustrate the search and masking procedure so (3.1) can also be used without modification, only the search and masking procedure will be slightly different. This is the approach that is utilised in the multiple shift maximum element (MSME) SMD algorithm. As with the SMD algorithm in Sec. 2.4.2 the i th iteration of the MSME-SMD approach is completed with a full EVD of the zero lag matrix.

Causality Constrained MSME-SMD

In some cases, such as spectral factorisation [53,57] causality of the paraunitary matrix is important. Therefore this section discusses the causality constrained extension of the MSME-SMD algorithm. So far the causality of the (maximum element based) PEVD

algorithms has not been considered, as $\tau^{(i)}$ in (2.12) can be positive or negative. Note however, that if the maximum element is identified in column $k^{(i)}$ and row $m^{(i)}$ at lag $\tau^{(i)}$, the parahermitian nature of $\mathbf{S}^{(i-1)}(z) = \tilde{\mathbf{S}}^{(i-1)}(z)$ implies that a corresponding value sits in column $m^{(i)}$ and row $k^{(i)}$ at lag $-\tau^{(i)}$. Therefore, the same maximum element pair shifted by (2.12) can also be brought onto the zero lag matrix by

$$\mathbf{\Lambda}^{(i)}(z) = \text{diag}\{\underbrace{1 \dots 1}_{m^{(i)}-1} z^{\tau^{(i)}} \underbrace{1 \dots 1}_{M-m^{(i)}}\} \quad . \quad (3.6)$$

This alternative can be invoked to pick a causal operation from either (2.12) or (3.6) at the i th iteration, such that the overall paraunitary matrix in (2.16) consists of only causal elements. The two operations, although shifting the same two target elements, will however result in different parahermitian matrices $\mathbf{S}^{(i)'}(z)$. The SMD algorithm with its column norm based search cannot be made causal as the column norm search space doesn't have any symmetry.

Applying the idea of causal shifts to the MSME-SMD algorithm, the initial search is restricted to the first half of the parahermitian matrix (with no other restrictions), and the element found must be shifted in its row. Once the first element is brought onto the zero lag and permuted into the upper left off-diagonal position it will be akin to Fig. 3.9 (a). Although the elements 1 and 1* could be the same between Fig. 3.8 (a) and Fig. 3.9 (a) the non-causal method has the choice of whether 1 is brought on to the zero lag in its column or row where as C-MSME-SMD is restricted to only delay rows. Otherwise Fig. 3.9 (a) is identical to Fig. 3.8 (a) but the following steps differ.

Selecting the elements from the first half of the parahermitian matrix in the 'C' locations in Fig. 3.9 (a) would require an anti-causal row shift onto the zero lag therefore the C-MSME-SMD algorithm must choose elements from the 'R' positions only. With the causal restriction it is obvious to see that the search space is now halved compared to MSME-SMD. Fig. 3.9 (b) is very similar to Fig. 3.8 (b) apart from 2 and 2* have been switched meaning for C-MSME-SMD 2 was found in an 'R' element but for MSME-SMD it was found in one of the 'C' elements. If elements 2/2* and 2*/2 are the same between MSME-SMD and C-MSME-SMD, the other elements brought onto the zero

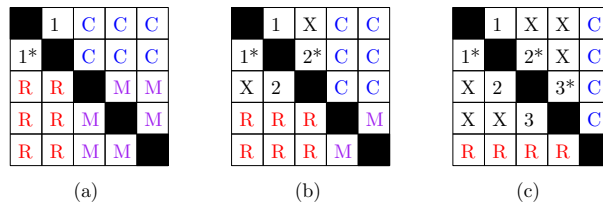


Figure 3.9: Masking used in the method that maximises the number of shift for the causal MSME-SMD algorithm.

lag will be the same because they are restricted to shift in terms of columns or rows to protect the previous elements. With a larger search space however, it is likely that MSME-SMD will bring different elements onto the zero lag. The remaining shifts to get Fig. 3.9 (c) and beyond follow the same routine as the second element i.e. only choosing from the locations marked ‘R’ and from the front half of the parahermitian matrix.

Like the MSME-SMD algorithm, this can either be done with the permutations shown in Fig. 3.9 and (3.5) or without permutations using a slightly different search method and (3.1) although $\tau^{(i,m)} \geq 0$ in both cases. As with all other SMD based algorithms the i th iteration is completed with a full EVD of the zero lag being applied to all lags of the parahermitian matrix.

3.3 Multiple Shift SBR2

In this section the multiple shift idea is applied to the SBR2 algorithm. The primary advantage of the SBR2 algorithm is its low cost; with the multiple shift technique the diagonalisation performance will improve leading to a powerful, low cost PEVD algorithm. The multiple shift second order sequential best rotation algorithm (MS-SBR2) [42] was based on [39–41] and developed in collaboration with J. G. McWhirter’s group from Cardiff University.

3.3.1 Idea

This approach was influenced by the MSME-SMD method described above however the search is modified such that the Jacobi transformations applied to each maximum

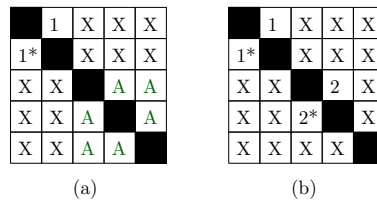


Figure 3.10: Masking used in the MS-SBR2 algorithm.

element do not affect any other chosen element i.e. no maximum element can share a row or column with another. The modification restricts the number of elements that can be brought onto the zero lag to be only $\lfloor M/2 \rfloor$.

3.3.2 Algorithm

The MS-SBR2 algorithm must take into account that the Jacobi transforms affect the row and column of the pair that they are applied to. The MS-SBR2 search must choose elements that do not share a common row or column so that no element is affected by the Jacobi transform of another element. The MS-SBR2 search is similar to the worst case scenario of the “Greedy” search shown in Fig. 3.6; although here we now avoid the elements that share a row or column with previous maximum elements. In Fig. 3.10 (a) the ‘C’s and ‘R’s from Fig. 3.6 are replaced with ‘X’s (to indicate they cannot be chosen) and we can only now choose from the elements marked with an ‘A’. As one may expect, limiting the search in this way also reduces the number of elements that can be found. In Fig. 3.10 (b) a total of only 2 elements are found and in general it is $\lfloor M/2 \rfloor$ elements i.e. about half the number found by MSME-SMD.

As with the multiple shift algorithms above the shifts can be applied using (3.5) or (3.1) depending on whether the permutations shown in the diagrams are used or not. The i th iteration of the MS-SBR2 is completed by eliminating each of the selected elements in turn with the Jacobi transform from (2.3).

3.4 Proof of Convergence

Theorem 2 (Convergence of the Multiple Shift Algorithms). *With a sufficiently large number of iterations I , the multiple-shift PEVD algorithms approximately diagonalise*

$\mathbf{R}(z)$ and decrease the power in off-diagonal elements to an arbitrarily low threshold $\epsilon > 0$.

Proof. A number of norms are required to prove Theorem 2. With $s_{m,m}^{(i)}[0]$ the m th diagonal element of $\mathbf{S}^{(i)}[0]$,

$$\mathcal{N}_1\{\mathbf{S}^{(i)}(z)\} \triangleq \sum_{m=1}^M |s_{m,m}^{(i)}[0]|^2 \quad (3.7)$$

is invariant to shifts and permutations, i.e.

$$\begin{aligned} \mathcal{N}_1\{\mathbf{S}^{(i)'}(z)\} &= \mathcal{N}_1\{\mathbf{\Lambda}^{(i)}(z)\mathbf{S}^{(i-1)}(z)\tilde{\mathbf{\Lambda}}^{(i)}(z)\} \\ &= \mathcal{N}_1\{\mathbf{S}^{(i-1)}(z)\} \quad . \end{aligned} \quad (3.8)$$

The energy of the lag zero matrix

$$\mathcal{N}_2\{\mathbf{S}^{(i)}(z)\} \triangleq \|\mathbf{S}^{(i)}[0]\|_{\mathbb{F}}^2 \quad (3.9)$$

is invariant under any unitary operation,

$$\begin{aligned} \mathcal{N}_2\{\mathbf{S}^{(i)}(z)\} &= \mathcal{N}_2\{\mathbf{Q}^{(i)}\mathbf{S}^{(i)'}(z)\mathbf{Q}^{(i)\text{H}}\} \\ &= \mathcal{N}_2\{\mathbf{S}^{(i)'}(z)\} \quad . \end{aligned} \quad (3.10)$$

Further,

$$\mathcal{N}_3\{\mathbf{S}^{(i)}(z)\} \triangleq \mathcal{N}_2\{\mathbf{S}^{(i)}(z)\} - \mathcal{N}_1\{\mathbf{S}^{(i)}(z)\} \quad (3.11)$$

$$\mathcal{N}_4\{\mathbf{S}^{(i)}(z)\} \triangleq \sum_{\tau} \|\mathbf{S}^{(i)}[\tau]\|_{\mathbb{F}}^2 \quad (3.12)$$

Chapter 3. Multiple Shift PEVD Algorithms

where $\|\cdot\|_F$ denotes Frobenius norm and the total energy $\mathcal{N}_4\{\cdot\}$ is invariant under the application of a paraunitary $\mathbf{G}^{(i)}(z)$ such that

$$\begin{aligned}\mathcal{N}_4\{\mathbf{S}^{(i)}(z)\} &= \mathcal{N}_4\{\mathbf{G}^{(i)}(z)\mathbf{S}^{(i-1)}(z)\tilde{\mathbf{G}}^{(i)}(z)\} \\ &= \mathcal{N}_4\{\mathbf{S}^{(i-1)}(z)\} \quad .\end{aligned}\tag{3.13}$$

For the off-diagonal norm at the i th iteration,

$$\mathcal{N}_3\{\mathbf{S}^{(i)'}(z)\} \geq 2\|\hat{\mathbf{s}}_{k^{(i)}}^{(i-1)}[\tau^{(i)}]\|_\infty^2 = 2\gamma^{(i)} \quad .\tag{3.14}$$

In the following diagonalisation step with $\mathbf{Q}^{(i)}$, this energy is transferred onto the main diagonal such that $\mathcal{N}_3\{\mathbf{S}^{(i)}(z)\} = 0$. With

$$\begin{aligned}\mathcal{N}_1\{\mathbf{S}^{(i)}(z)\} &> \mathcal{N}_1\{\mathbf{S}^{(i)'}(z)\} + 2\gamma^{(i)} \\ &= \mathcal{N}_1\{\mathbf{S}^{(i-1)}(z)\} + 2\gamma^{(i)}\end{aligned}\tag{3.15}$$

and $\gamma^{(i)} > 0$, $\mathcal{N}_1\{\mathbf{S}^{(i)}(z)\}$ increases monotonically with iteration index i . Since

$$\mathcal{N}_1\{\mathbf{S}^{(i)}(z)\} \leq \mathcal{N}_4\{\mathbf{S}^{(i)}(z)\} \quad \forall i \quad ,\tag{3.16}$$

with the overall energy, $\mathcal{N}_4\{\mathbf{S}^{(i)}(z)\}$, remaining constant, $\mathcal{N}_1\{\mathbf{S}^{(i)}(z)\}$ must have a supremum S ,

$$S = \sup_i \mathcal{N}_1\{\mathbf{S}^{(i)}(z)\} \quad .\tag{3.17}$$

It follows that for any $\epsilon > 0$ there must be an iteration number I for which $S - \mathcal{N}_1\{\mathbf{S}^{(I)}(z)\} < \epsilon$ and so the increase $2\gamma^{(I+i)}$, $i \geq 0$, at any subsequent stage must satisfy

$$2\gamma^{(I+i)} \leq S - \mathcal{N}_1\{\mathbf{S}^{(I)}(z)\} < \epsilon \quad .\tag{3.18}$$

Hence, for any $\epsilon > 0$, there must be an iteration I by which $\gamma^{(I+i)}$, $i \geq 0$, is bounded by ϵ . □

3.5 Search Complexities

The following subsections investigate the computational complexities associated with each of the search techniques. The complexities are derived for the general case where the parahermitian matrix, at the i th iteration has a matrix dimension M , and support of $2L + 1$.

3.5.1 Maximum Element Search

Both the SBR2 and ME-SMD PEVD algorithms use the l_∞ norm from (2.18). Due to the parahermitian symmetry the maximum element search can be carried out in a search space of only $M^2(L + 1)$. In each iteration, the maximum element can be identified without any explicit norm calculation but requires a search over a set of $\mathcal{O}(M^2(L + 1))$ elements; this can be simplified to $\mathcal{O}(M^2L)$ because as L grows the $+1$ becomes insignificant [58].

3.5.2 Column Norm Search

The original SMD algorithm in its i th iteration inspects the vectors $\hat{\mathbf{s}}_k^{(i-1)}[\tau]$, which are the columns of $\mathbf{S}^{(i-1)}[\tau]$ but modified by removing its on-diagonal elements. Using the column norm search from (2.21), a total of $M(2L + 1)$ column norms of the parahermitian matrix have to be calculated. Each norm requires a squaring of elements, but the square root operation can be omitted as only a comparison of norms but no explicit values are required. Thus, with each column vector having length M , the norm computation is $\mathcal{O}(M^2(2L + 1))$ followed by a search over $\mathcal{O}(M(2L + 1))$ elements. Like with the maximum element search these can be simplified to $\mathcal{O}(M^2(2L))$ and $\mathcal{O}(M(2L))$ respectively.

3.5.3 Multiple Shift Searches

MSME-SMD

The search method used in the MSME-SMD algorithm initiates every iteration by scanning the parahermitian matrix for its maximum off-diagonal element similar to

Table 3.1: Order comparison of SMD search methods.

method	norm calc.	comparisons	total
SMD	$\mathcal{O}(M^2(2L))$	$\mathcal{O}(M(2L))$	$\mathcal{O}(M^2(2L))$
ME-SMD	$\mathcal{O}(0)$	$\mathcal{O}(M^2L)$	$\mathcal{O}(M^2L)$
(C-)MSME-SMD	$\mathcal{O}(0)$	$\mathcal{O}(M^3L)$	$\mathcal{O}(M^3L)$
MS-SBR2	$\mathcal{O}(0)$	$\mathcal{O}(\lfloor M/2 \rfloor M^2L)$	$\mathcal{O}(\lfloor M/2 \rfloor M^2L)$

SBR2 and ME-SMD, employing the l_∞ instead of the l_2 norm. Like before the symmetry can be exploited such that only half of the parahermitian matrix needs to be searched.

This approach requires no norm evaluations but the complexity of the search is $\mathcal{O}(M^3L)$ because each iteration involves the $\mathcal{O}(M^2L)$ maximum element search a total of $M - 1$ times (where for asymptotic analysis $M - 1$ is simplified to M [58]). The search space for the C-MSME-SMD algorithm is the same number of elements as that of MSME-SMD; therefore the complexity is the same however more of the elements are masked in C-MSME-SMD for the causal restriction.

Multiple Shift SBR2

As with the MSME-SMD algorithm the MS-SBR2 algorithm utilises the l_∞ norm from (2.18). Like before the search is repeated however with MS-SBR2 it is only a total of $\lfloor M/2 \rfloor$ times. With the MS-SBR2 search because the number of elements found is only $\lfloor M/2 \rfloor$ the complexity goes down to $\mathcal{O}(\lfloor M/2 \rfloor M^2L)$. Like the other maximum element based search methods MS-SBR2 does not require any additional norm calculations.

3.5.4 Comparison

An overall order comparison of the four search methods is provided in Tab. 3.1, with a total search order provided on the basis that one comparison for the maximum search is about as expensive as one multiply-accumulate operation. Note the value of L grows at each iteration and the rate of growth also varies between algorithms.

3.6 Results

In this section a number of different scenarios are considered each of which requires specific performance metrics which are introduced below. The first experimental results compare each of the search methods to the exhaustive search for a small range of matrix sizes. Next the energy transfer of the various sub-optimal searches is tested for larger parahermitian matrix dimensions. Finally the search methods are put into PEVD algorithms and they are evaluated in terms of convergence, paraunitary order and computational complexity.

3.6.1 Performance Metrics

Each of the energy transfer experiments requires a slightly different performance metric. The initial set of experiments in this section concentrate on the energy that a particular search method can bring onto the zero lag at a given iteration i . To compare the different approaches we record the energy that is on the zero lag after the shift matrix, $\mathbf{\Lambda}^{(i)}(z)$, has been applied,

$$E = \sum_{k=1}^M \|\hat{\mathbf{s}}_k^{(i)'}[0]\|_2 \quad , \quad (3.19)$$

where $\hat{\mathbf{s}}_k^{(i)'}[\tau]$ is the modified (off-diagonal) column after $\mathbf{\Lambda}^{(i)}(z)$ has been applied. The energy transfer for each algorithm is then compared to that of the exhaustive, maximum energy search and shown as a percentage.

For the second set of tests the proportion of shifted energy, $E_{\text{shift}}^{(M,L)}$, is calculated as

$$E_{\text{shift}}^{(M,L)} = \frac{\sum_{k=1}^M \|\hat{\mathbf{s}}_k^{(M,L)'}[0]\|_2^2}{\sum_{\tau} \sum_{k=1}^M \|\hat{\mathbf{s}}_k^{(M,L)'}[\tau]\|_2^2} \quad , \quad (3.20)$$

where $\hat{\mathbf{s}}_k^{(M,L)'}[\tau]$ is the modified column vector for an $M \times M \times 2L + 1$ parahermitian matrix. The numerator in (3.20) is the the off-diagonal energy brought onto the zero lag and the denominator is the off-diagonal energy in the entire parahermitian matrix. The algorithm that shifts most energy onto the zero lag consequently produces the highest $E_{\text{shift}}^{(M,L)}$.

Finally the overall convergence is measured using a suitable normalised metric from [10] is

$$E_{\text{norm}}^{(i)} = \frac{\sum_{\tau} \sum_{k=1}^M \|\hat{\mathbf{s}}_k^{(i)}[\tau]\|_2^2}{\sum_{\tau} \|\mathbf{R}[\tau]\|_F^2} \quad (3.21)$$

based on the definition of $\hat{\mathbf{s}}_k^{(i)}[\tau]$ in (2.18). In the ideal case (3.21) should reduce to zero but in reality it tends to only approach but not reach zero.

Execution time is used to measure the computational complexity of the search methods in Matlab 2014a with the following system specification: Ubuntu 14.04 on a Dell Precision T3610 with Intel® Xeon® E5-1607V2 3.00 GHz x 4 cores and 8 GB RAM. The order of the paraunitary matrices are also recorded at each algorithm iteration; as all algorithms start with the same input parahermitian matrix the paraunitary order is directly related to the growth of the parahermitian matrix.

3.6.2 Energy Transfer

The first part of this section benchmarks the energy transfer of the various PEVD search algorithms as a percentage of the exhaustive search from Sec. 3.1. Due to the large computational cost of the exhaustive search the comparison is only carried out over a small range of matrix sizes. In the second part the exhaustive search is omitted and the PEVD search algorithms are compared using their relative performance over a larger range of matrix dimensions. These tests were originally developed for SMD based algorithms in [41, 59] so make the following two assumptions:

1. all energy shifted onto the zero lag gets diagonalised;
2. prior to the search taking place no energy is on the zero lag.

Although the SBR2 family of PEVD algorithms does not fit the two assumptions above the MS-SBR2 search algorithm is included here to give an idea of its performance relative to the other search methods. The inclusion of MS-SBR2 also highlights the importance of the masking steps in MSME-SMD search which guarantee the transfer of $M - 1$ elements.

Percentage Energy Transfer

Simulating the i th iteration of an SMD algorithm, $\mathbf{S}^{(i-1)}(z)$ is set as $\mathbf{S}(z)$ but with zeroed off-diagonal elements on the zero lag. Fig. 3.11 compares the values of E , from (3.19), as a percentage of that obtained by maximum energy SMD algorithm for $M = 3$ and $M = 4$ over a range of values for the maximum lag L . For $M = 3$, where the MSME algorithms are shifting 2 rows there is a significant difference between them and the single shift algorithms. When $M = 3$ the MS-SBR2 search can only shift 1 element and therefore performs the same as the single shift maximum element search. Increasing M to 4 the performance of the single shift algorithms begins to degrade compared to the multiple shift algorithms; this time the MSME algorithms shift 3 elements and MS-SBR2 shifts 2 elements. When the values of L are small the number of shift possibilities are reduced and it is more likely that the single shift algorithms will pick a near optimum shift. In general from Fig. 3.11, the performance of the algorithms as a percentage of the exhaustive search tend to deteriorate and level off as the lag dimension increases and the number of possible shifts increase.

The same situation is shown in Fig. 3.12, but for a fixed value of $L = 1$, the matrix dimension M is varied. Like Fig. 3.11 the MSME search algorithms perform very closely to the exhaustive search over the range of M values. As M increases the performance of the single shift algorithms deteriorates to the point where at $M = 7$ the maximum element search is transferring less than half of what can be achieved with the exhaustive search. The MS-SBR2 search initially follows the maximum element search until $M > 3$; at which point the algorithm allows more than one element to be brought onto the zero lag.

The major drawback in the exhaustive search algorithm is its extremely high computational cost, for example the simple case where $M = 7$ and $L = 1$ has 823,543 possible combinations of which 543,607 are independent and must be tested for each different parahermitian matrix. Due to this extremely high computational cost in the exhaustive search, the MSME search algorithms become a very attractive alternative. In the simulations shown in Fig. 3.11 and Fig. 3.12 the MSME search algorithms consistently perform at around 90% of the exhaustive search but at a fraction of the cost.

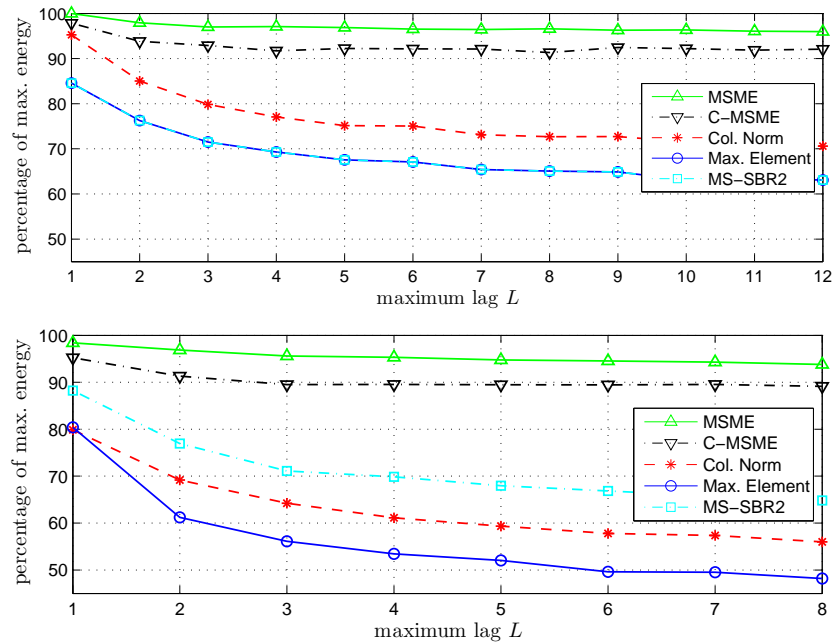


Figure 3.11: Percentage of maximum off-diagonal energy transferred by search algorithms in the i th iteration for (a) $\mathbf{S}^{(i)}(z) \in \mathbb{C}^{3 \times 3}$ for different maximum lag values L and (b) $\mathbf{S}^{(i)}(z) \in \mathbb{C}^{4 \times 4}$ for different maximum lag values L .

Relative Energy Transfer

Although the exhaustive search is far too costly for larger matrices the relative performance of each of the algorithms can be compared for a larger range of matrix dimensions. The relative energy transfer for the various algorithms, calculated using (3.20), is shown in Fig. 3.13 for a selection of parahermitian matrices where $M = 4, 6, 10$ and $L = 50, 100 \dots, 500$ over an ensemble of 1000 instantiations.

Like the scenarios in Fig. 3.11 and Fig. 3.12 the MSME search methods are superior to the rest, especially the single shift algorithms. The performance gap between the multiple and single shift algorithms gets significantly larger as M is increased due to the fact that the multiple shift algorithms start to bring even more energy onto the zero lag. The MS-SBR2 search algorithm also performs very well compared to the single shift algorithms but loses out slightly to the MSME search methods which allow more elements to be shifted. Comparing the single shift algorithms, the column norm search comes out better than the maximum element due to the fact that it considers

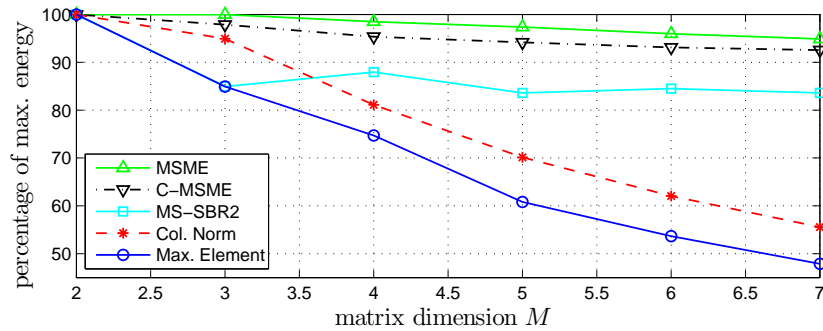


Figure 3.12: Percentage of maximum off-diagonal energy transferred by search algorithms in the i th iteration for $\mathbf{S}^{(i)}(z) \in \mathbb{C}^{M \times M}$ with $L = 1$ for variable M .

all elements brought onto the zero lag by a shift rather than just one. As mentioned in Sec. 3.1 the column norm search is effectively an exhaustive algorithm for the single shift case.

To give an idea of the cost of extending the exhaustive search simulations above to the size of matrices considered here, for the case of $M = 10$ and $L = 500$ there are approximately 3.41×10^{36} possible shift combinations from which around 7.6×10^{33} are independent and would need to be tested. The reason that the plots in Fig. 3.13 do not plateau like those in Fig. 3.11 is that the overall energy in the matrices (i.e. the denominator in (3.20)) increases with L whereas the energy that can be brought onto the zero lag remains similar. In short the exhaustive search energy (which the other algorithms are compared to) in Fig. 3.11 is close to constant whereas the denominator for the performance metric in Fig. 3.13 increases with L .

3.6.3 Diagonalisation

Due to the large range of different sized matrices investigated the previous two experiments have used randomly generated parahermitian matrices. For the remainder of this thesis a source model based on the one described in [10] is used to carefully control the majorisation and dynamic range of the source signals. The source model is explained in full detail in Chapter. 5. The experiment for this section is carried out over an ensemble of 1000 different matrices produced by the source model from [10]. The initial dimensions of the matrices are all $M = 6$ and $L = 59$ and the average dynamic

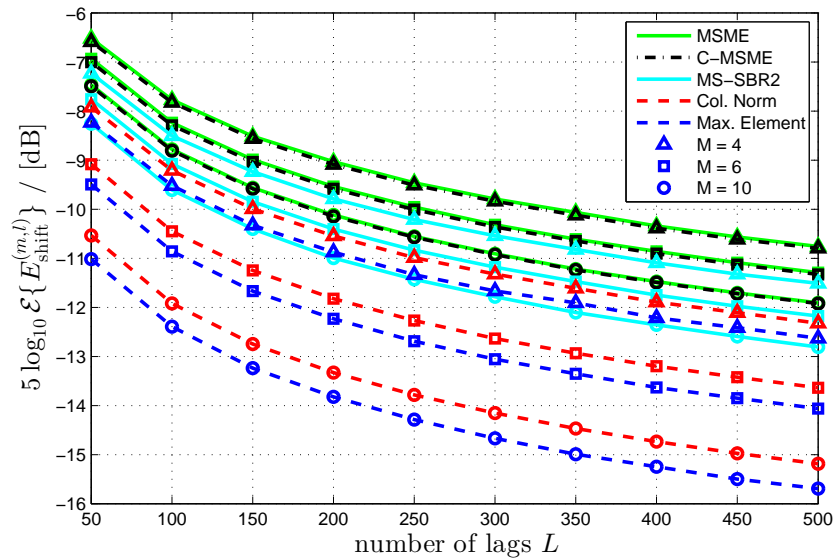


Figure 3.13: Average energy transferred for varying matrix size and search algorithm.

range of the ensemble is approximately 20 dB.

Fig. 3.14 shows the reduction in off-diagonal energy calculated using (3.21) against algorithm iterations. As expected, based on the results in the previous subsections, the two MSME-SMD PEVD algorithms have a superior energy transfer compared to the PEVD algorithms using the other search methods. Likewise MS-SBR2 outperforms the single shift SBR2 algorithm. Despite only shifting one row/column pair the SMD algorithm with its column norm search performs reasonably well compared to the other PEVD algorithms. Although it starts off well the diagonalisation performance of the ME-SMD algorithm gets relatively worse as the algorithm iterations increase.

3.6.4 Real Time Convergence

The results from Fig. 3.14 do not take into account the time required to compute each iteration of the various PEVD algorithms. Fig. 3.15 shows a real time convergence example for $M = 4, 6$ & 10 where the initial value for L is 59 and the average dynamic range is approximately 20 dB based on an ensemble of 1000 different parahermitian matrices for each value of M .

Despite the more complex search step the MSME-SMD algorithms converge faster

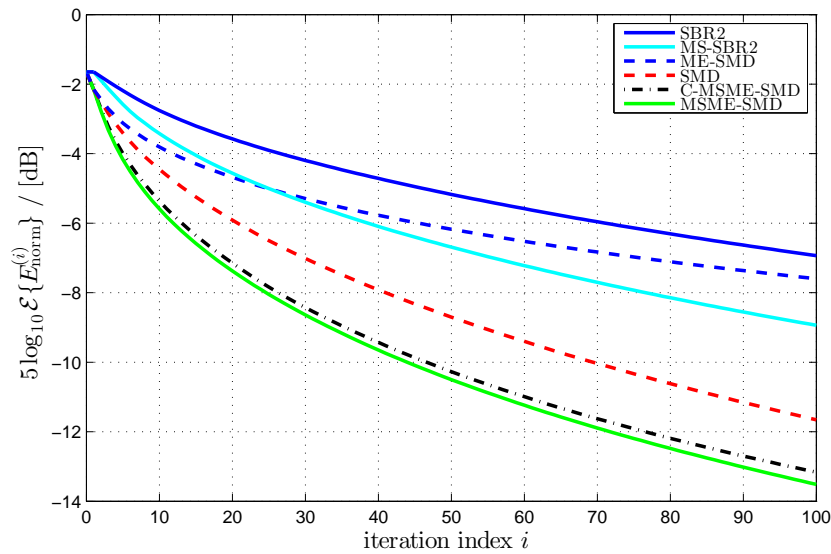


Figure 3.14: Diagonalisation vs. algorithm iterations for the SMD algorithm and the two MSME-SMD variants.

in real time than the relatively simpler ME-SMD and SMD algorithms. In reality the search step is only a small part of the overall execution time for the SMD algorithms. The execution time is dominated by the step which applies the EVD modal matrix to all lags of the parahermitian matrix. Therefore transferring more energy at each iteration outweighs the slight increase in complexity.

The results for the SBR2 based algorithms confirm the statements from Chapter. 2; i.e. in real time the SBR2 algorithm converges faster. Comparing the single vs. multiple shift algorithms for SBR2 we see that the simpler maximum element search is marginally faster i.e. in this case the search complexity has a larger impact on the execution time. The trends are very similar for all values of M used in Fig. 3.15.

3.6.5 Order Increase

Fig. 3.16 shows how the paraunitary matrices grow as the off-diagonal energy in the parahermitian matrix is reduced. One of the inherent downsides of shifting multiple row/column pairs onto the zero lag is the growth in the paraunitary (and as a consequence, parahermitian matrices). Most of the single shift algorithms, in particular the column norm based SMD, show a minimal growth in the paraunitary matrix order. As

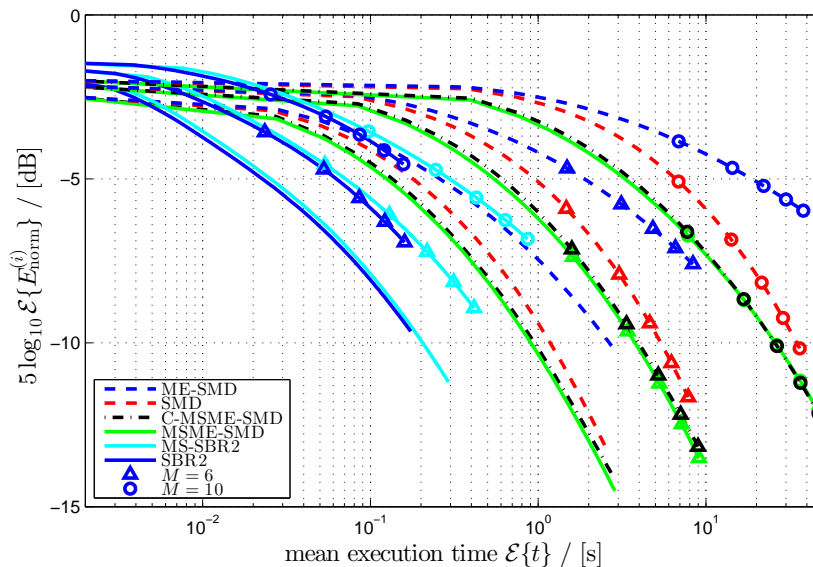


Figure 3.15: Reduction in off-diagonal energy vs. mean execution time over 100 algorithm iterations.

M is increased the paraunitary matrices for SBR2 and SMD stay relatively constant; in the case of SMD they slightly reduce in order as M is increased. For the multiple shift algorithms the growth gets larger with the matrix dimension M as they are allowed to shift more elements. Despite having the same search step as SBR2 and the same diagonalisation step as SMD, the ME-SMD algorithm performs worst of all; the matrices even grow significantly as M is increased.

3.7 Chapter Summary & Conclusions

This chapter first introduced the idea of shifting multiple rows and columns during one PEVD algorithm iteration. Based on the idea of multiple shifts, the possibility of an exhaustive search was outlined which would try every independent shift combination in order to determine the one which transfers the most energy onto the zero lag. As the parahermitian matrix grows in both width (M) and length (L), the number of independent shifts — and therefore the cost — grow significantly. In reality the exhaustive search approach is far too costly to implement as part of a PEVD algorithm; the multiple shift maximum element (MSME) search was therefore developed as an alternative. After investigating different search and masking approaches a method was

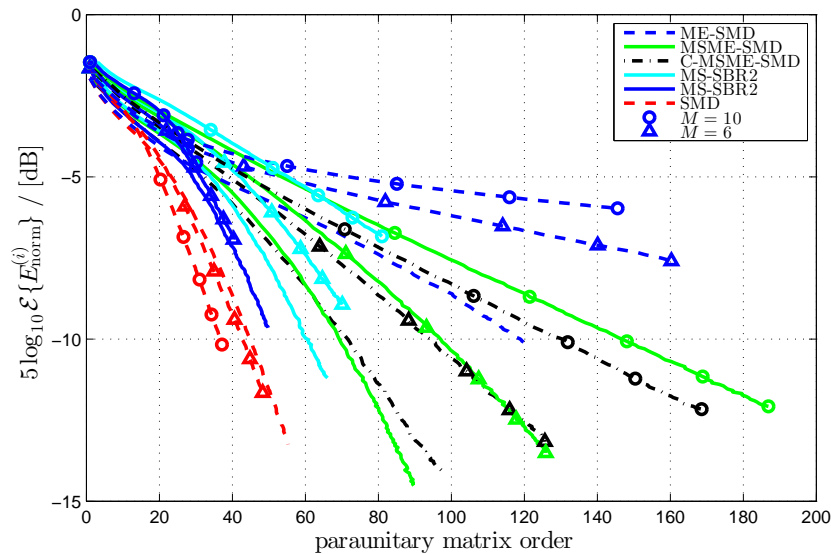


Figure 3.16: Length growth of paraunitary matrix during the i th iteration for $\mathbf{S}^{(i-1)}(z) \in \mathbb{C}^{M \times M}$ with maximum lag $L = 1$ and variable M .

developed that is guaranteed to bring $M - 1$ maximum elements onto the zero lag at each iteration. Based on the MSME method a causal approach has been developed and the multiple shift idea has influenced the development of the MS-SBR2 algorithm.

Although costly, the exhaustive search has been used as a benchmark to gauge the relative performance of each of the different search methods over a small range of matrix sizes. The results showed that the MSME search performs on average within 90% of the maximum energy approach. A similar test was then carried out for a larger range of matrix sizes and the absolute performance was recorded. These results confirmed the results from the previous test showing the higher energy transfer of the multiple shift algorithms. When incorporated into full PEVD algorithms the multiple shift methods were then shown to also transfer more energy at each iteration and therefore diagonalise parahermitian matrices in fewer iterations. Both the SBR2 and the MS-SBR2 algorithms converge significantly faster in real time than the SMD based algorithms; this issue which is inherent to all SMD based algorithms will be investigated in the next chapter. Despite the more complex search step the MSME-SMD algorithms have actually been shown to converge faster in real time than the single shift SMD algorithms. The final set of results highlighted an unwanted side effect of the MSME

Chapter 3. Multiple Shift PEVD Algorithms

search, which is the polynomial eigenvalues and eigenvectors grow faster than any of the other algorithms. In the next chapter methods to limit this polynomial growth will be explored along with a new method of truncating paraunitary matrices which is applicable to all PEVD algorithms.

Chapter 4

Efficient Implementations

In the results section of Chapter 3, three main performance metrics were investigated, namely:

- diagonalisation,
- polynomial matrix order growth, and
- real time execution.

While Chapter 3 mainly focused on improving the diagonalisation, this chapter introduces methods which target the other two performance metrics. First the row-shift method for paraunitary truncation is introduced; this allows a further reduction in the paraunitary matrix order compared to the current state-of-the-art. Next the restricted search (RS) MSME-SMD is discussed which limits the lags in which elements can be found. RS-MSME-SMD aims to reduce the polynomial matrix growth in the MSME-SMD algorithm. An added benefit of the RS-MSME-SMD algorithm is that the real time execution is reduced as the algorithm works on a lower number of coefficients. The final section of this chapter introduces the cyclic-by-rows based SMD approximation. The cyclic-by-rows approximation is intended to improve the real time execution of the SMD based algorithms by removing the costly EVD step.

4.1 Row-Shift Truncation

The truncation of paraunitary matrices in [54] follows the idea for trimming parahermitian matrices expressed in [3, 55]. Below, the approach in [54] is reviewed, before the proposed approach [43] is outlined, followed by a numerical example. Any truncation of paraunitary matrices results in a loss of the paraunitary property, which will be discussed further in the results section.

4.1.1 State-of-the-Art Truncation

The truncation method in [54] discards outer lags of a paraunitary matrix and can remove up to a predefined proportion of energy μ from $\mathbf{Q}(z) \bullet \circ \mathbf{Q}[n]$. If truncation is written as a non-linear operation $f_{\text{lag}}(\cdot)$, then the proportion of removed energy is given by

$$\begin{aligned} \gamma_{\text{lag}} &= 1 - \frac{\sum_n \|f_{\text{lag}}(\mathbf{Q}[n])\|_{\text{F}}^2}{\sum_n \|\mathbf{Q}[n]\|_{\text{F}}^2} \\ &= 1 - \frac{1}{M} \sum_n \|f_{\text{lag}}(\mathbf{Q}[n])\|_{\text{F}}^2 \quad , \end{aligned} \quad (4.1)$$

where $\|\cdot\|_{\text{F}}$ is the Frobenius norm. The energy is removed by omitting the leading N_1 and trailing N_2 matrices from $\mathbf{Q}[n]$ of length N , such that

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

This leads to the following constrained optimisation problem to perform the truncation:

$$\text{maximise} \quad (N_1 + N_2) \quad (4.2)$$

$$\text{s.t.} \quad \gamma_{\text{lag}} \leq \mu \quad . \quad (4.3)$$

In practice, this approach can be implemented by sequentially removing leading or trailing matrices of $\mathbf{Q}[n]$ — whichever has the smallest Frobenius norm — as long as the constraint (4.3) remains satisfied.

4.1.2 Proposed Row-Shift Truncation

The proposed method takes advantage of the paraunitary matrix ambiguity from Sec. 2.5, that allows us to modify $\mathbf{Q}(z)$ with $\mathbf{\Gamma}(z)$ to create an equivalent paraunitary matrix $\tilde{\mathbf{Q}}(z)$. The goal of the row-shift truncation is to introduce a modifying matrix $\mathbf{\Gamma}(z)$ which lowers the order of the paraunitary matrix. Defining $\mathbf{Q}(z)$ with its constituent row vectors $\mathbf{q}_m(z)$, $m = 1 \dots M$,

$$\tilde{\mathbf{Q}}(z) = [\mathbf{q}_1(z) \dots \mathbf{q}_M(z)] \quad , \quad (4.4)$$

note that $\tilde{\mathbf{q}}_i(z)\mathbf{q}_j(z) = \delta(i - j)$. Therefore each vector $\mathbf{q}_m(z)$ has unit energy, and it appears sensible to truncate the same proportion of energy from every vector. With a vector-valued truncation $f_{\text{row}}(\mathbf{q}_m[n])$, the proportion of removed energy is

$$\gamma_{\text{row},m} = 1 - \sum_n \|f_{\text{row}}(\mathbf{q}_m[n])\|_2^2 \quad . \quad (4.5)$$

Based on the truncation definition

$$f_{\text{row}}(\mathbf{q}_m[n]) = \begin{cases} \mathbf{q}_m[n + N_{1,m}] & 0 \leq n < T_m \\ \mathbf{0} & \text{otherwise} \end{cases} \quad , \quad (4.6)$$

with $T_m = N - N_{2,m} - N_{1,m}$, the optimum truncation based on the row-shift ambiguity is given by the constrained problem

$$\text{maximise} \quad \min_m (N_{1,m} + N_{2,m}) \quad (4.7)$$

$$\text{s.t.} \quad \gamma_{\text{row},m} \leq \frac{\mu'}{M} \quad \forall m = 1 \dots M \quad , \quad (4.8)$$

where μ' is the threshold of energy shed. With this, the row-shifts $\tau_m = N_{1,m}$, $m = 1 \dots M$, correcting the truncation are identified and can be applied via $\mathbf{\Gamma}(z)$ in (2.23). The truncated matrix after row correction will have length $\max_m(T_m)$.

In practise, every row vector of $\mathbf{Q}(z)$ is treated individually like the matrix in the previous approach of [54] and Sec. 4.1.1. Note that the main complexity of both

truncation approaches lies in the calculation of norms; therefore, the proposed approach has only little overhead compared to [54].

Naturally some of the truncated rows will come out shorter than others; one approach to dealing with this is to simply zero pad them to the length of the longest. Zero padding gives good results however the overall error can be reduced by identifying the shorter rows and replacing them with a version (from the paraunitary matrix prior to truncation) that has been truncated to the same length as the longest row. Replacing the shorter rows in this way increases the complexity of the trim function and can in the worst case (when there are $M - 1$ shorter rows) almost double the computational complexity of the row-shift truncation compared to zero padding. Although the zero padding is computationally simpler than the two stage approach the paraunitary matrix is only ever truncated once the PEVD algorithm is complete (i.e. not at each algorithm iteration); therefore the truncation step is a small portion of the overall complexity in generating the paraunitary matrix. The previously published results for row-shift truncation in [43] use the zero pad method but the results in this thesis will use the technique with the additional step.

4.1.3 Truncation Example

To demonstrate the potential benefit of the proposed truncation, a simple example is considered here. By generating a CSD matrix $\mathbf{R}(z) \in \mathbb{C}^{4 \times 4}$ through a source model detailed in [10], we know that an exact decomposition $\mathbf{R}(z) = \tilde{\mathbf{Q}}_{\text{source}}(z)\mathbf{D}_{\text{source}}(z)\mathbf{Q}_{\text{source}}(z)$ exists. The matrix $\mathbf{D}_{\text{source}}(z)$ is diagonal and of order 8; it is also spectrally majorised as shown by the shaded curves in Fig. 4.1. The paraunitary matrix $\mathbf{Q}_{\text{source}}(z)$ is of order 4.

Running SBR2 for 100 iterations yields a well-diagonalised matrix $\mathbf{D}(z)$, whose power spectral densities very closely match those of $\mathbf{D}_{\text{source}}(z)$, as demonstrated in Fig. 4.1. This accuracy is not met by the paraunitary matrix $\mathbf{Q}(z)$, which, when left untrimmed, has an order of 137. Even though this matrix has many very small trailing coefficients, its order is significantly larger than that of the ground truth matrix $\mathbf{Q}_{\text{source}}(z)$.

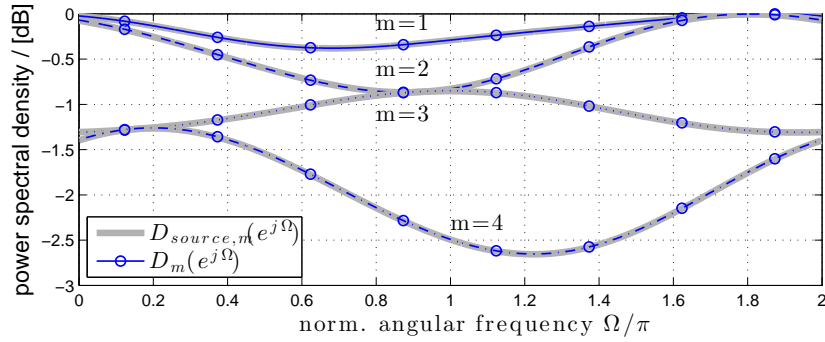


Figure 4.1: Power spectral densities of the source model $\mathbf{D}_{source}(z)$ and of the extracted matrix $\mathbf{D}(z)$ using SBR2.

Using a standard truncation as introduced in [54] with $\mu = 10^{-4}$ removes 0.1‰ of the total energy of $\mathbf{Q}(z)$. The resulting $f_{lag}(\mathbf{Q}[n])$ is shown in Fig. 4.2, and now only has order 29. Removing small trailing coefficients therefore has significantly reduced the order of $f_{lag}(\mathbf{Q}[n])$, and therefore the computational complexity that is required to implement such a system.

In Fig. 4.2, it is noticeable that the rows of $f_{lag}(\mathbf{Q}[n])$ are shifted with respect to each other: particularly the last row exhibits an advance compared to the remaining three, which is an indication of the manifold w.r.t. row shifts established during the analysis in Sec. 2.5. Therefore, with the proposed row-corrected truncation algorithm and the same shedding of 0.1‰ energy from $\mathbf{Q}[n]$, the resulting $f_{row}(\mathbf{Q}[n])$ of only order 12 is shown in Fig. 4.2 (in red). Here the modifying matrix $\mathbf{\Gamma}(z)$ is set to $\text{diag}\{z^{-7} \ 1 \ z^{-1} \ z^{-20}\}$.

Even though the diagonalised matrices $\mathbf{D}_{source}(z)$ and $\mathbf{D}(z)$ are similar, the paraunitary matrix $\mathbf{Q}(z)$ differs substantially from $\mathbf{Q}_{source}(z)$, and $f_{row}(\mathbf{Q}[n])$ only approaches $\mathbf{Q}_{source}[n] \circ \bullet \mathbf{Q}_{source}(z)$ in order but not appearance. Similar effects are known from the EVD, where small disturbances result in similar energies being extracted by eigenvalues, but much larger differences can emerge in the eigenvectors [60]. Irrespective of this, the proposed truncation approach appears very worthwhile in reducing the order of $\mathbf{Q}(z)$, which will be more exhaustively demonstrated in the following section.

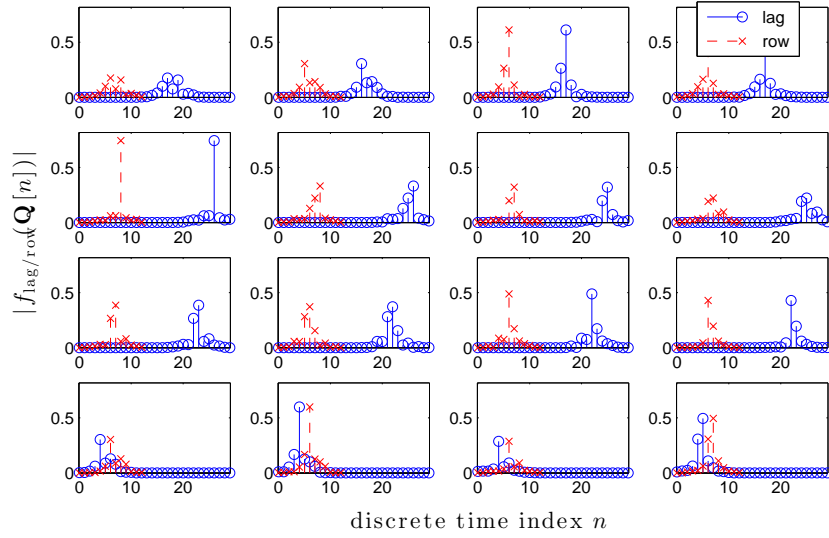


Figure 4.2: Paraunitary matrix truncated with $\mu = 10^{-4}$ using the lag based truncation and the proposed row based approach.

4.1.4 Results

To benchmark the proposed truncation approach, this section first defines performance metrics before setting out a simulation scenario, over which the performance metrics will be recorded. Initial experiments are carried out on the SBR2 algorithm before being extended to MS-SBR2, SMD and MSME-SMD.

Performance Metrics

In addition to the off-diagonal energy from (3.21) in Sec. 3.6.1 we record the length of the truncated paraunitary matrices at each iteration and the reconstruction error defined below.

Reconstruction Error. By truncating $\mathbf{Q}(z)$, its paraunitarity is lost. If interpreting $\mathbf{Q}(z)$ as a filter bank, the loss manifests itself as reconstruction error [5], and the difference to a paraunitary system can be assessed as

$$\mathbf{E}(z) = \mathbf{I}_{M \times M} - \mathbf{Q}_T(z)\tilde{\mathbf{Q}}_T(z) \quad . \quad (4.9)$$

where $\mathbf{Q}_T(z)$ is the truncated matrix, and with $\mathbf{E}[\tau] \circ \bullet \mathbf{E}(z)$ the reconstruction error

is given by

$$\xi = \frac{1}{M} \sum_{\tau} \|\mathbf{E}[\tau]\|_F^2 \quad . \quad (4.10)$$

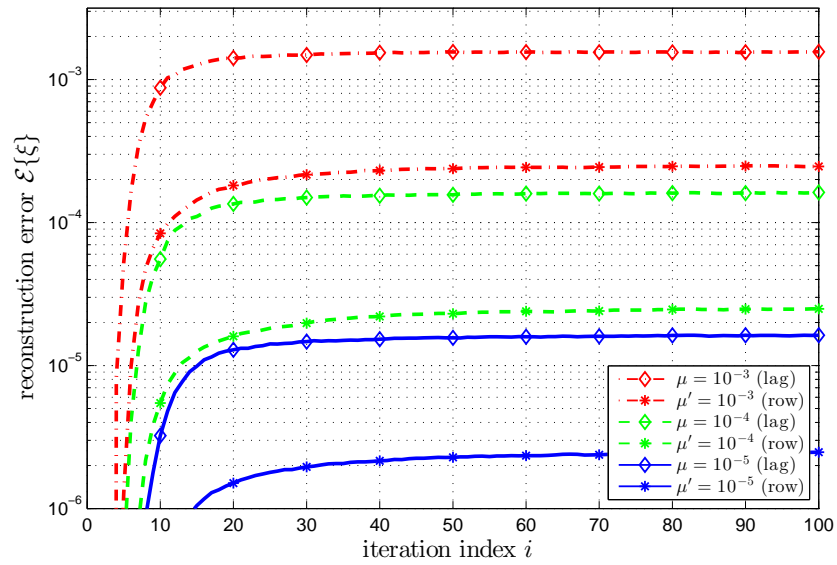
Simulation Scenario

The simulations below have been performed over an ensemble of 10^3 instantiations of $\mathbf{R}(z) \in \mathbb{C}^{4 \times 4}$ based on the randomised source model in [10]. In this source model, the order of $\mathbf{D}_{\text{source}}(z)$ is 30 and the order of $\mathbf{Q}_{\text{source}}(z)$ is 30, such that the total order of $\mathbf{R}(z)$ is 119. The dynamic range of the source model is constrained to ensure that in the ensemble the average is around 20 dB. First the two truncation methods are applied to paraunitary matrices produced by only the SBR2 algorithm to fine tune the truncation parameters. Using the parameters selected with the SBR2 experiments the truncation approaches are then applied to paraunitary matrices produced by the SMD, MS-SBR2 and MSME-SMD PEVD algorithms.

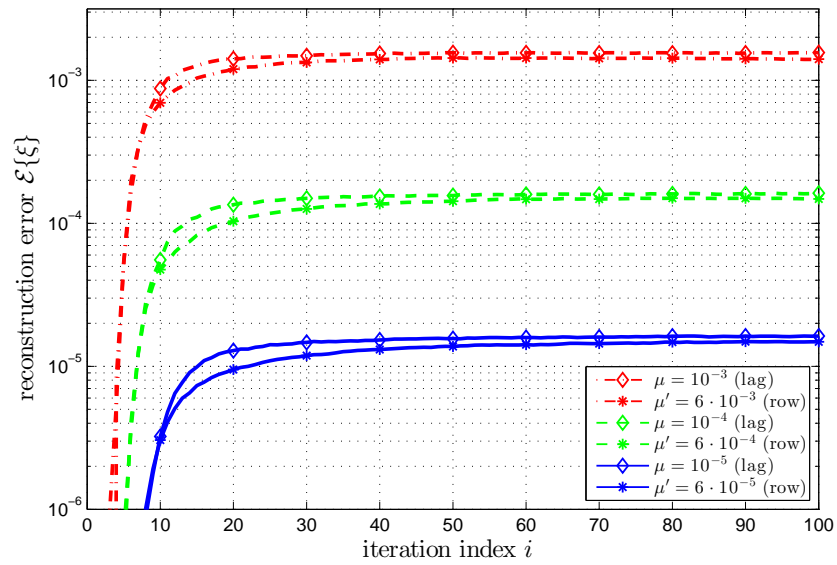
Reconstruction Error

The experiments have been carried out for three different truncation parameters $\mu = \{10^{-5}, 10^{-4}, 10^{-3}\}$ for $f_{\text{lag}}(\cdot)$ with the resulting reconstruction error ξ shown in Fig. 4.3 (a). With low iteration numbers, $\mathbf{Q}(z)$ is still of low order and there is limited choice for trimming, but with increased i , the truncation performs asymptotically to trim $\mathbf{Q}(z)$ by exactly μ . From Fig. 4.3 (a) the reconstruction error of the proposed row-shift truncation is significantly lower than the lag based method; meaning the truncation parameter μ can be increased to achieve a similar error.

With the proposed approach, it was found that μ can be scaled up by a factor of 6 to reach a similar error metric as the standard truncation, as shown in Fig. 4.3 (b). This more aggressive trimming for the same error metrics can be justified since in the standard truncation to remove whole matrix coefficients at the ends of $\mathbf{Q}(z)$ leads to larger errors ξ . In contrast, the proposed approach will truncate small coefficients evenly across rows and balance the overall error in ξ .



(a)



(b)

Figure 4.3: Ensemble reconstruction error $\mathcal{E}\{\xi\}$ vs. SBR2 iterations for the different truncation approaches and varying μ (a) $\mu' = \mu$ and (b) $\mu' = 6 \cdot \mu$.

Truncated Order and Diagonalisation

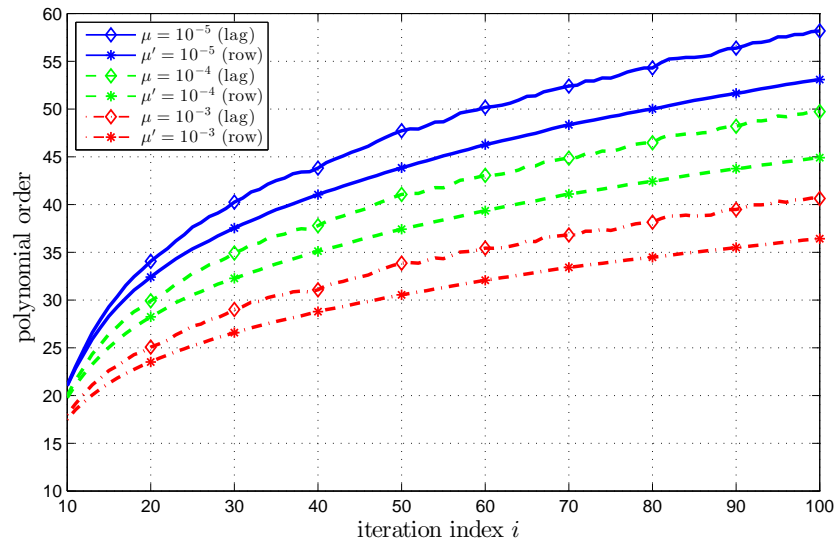
Using the different truncations $\mu = \{10^{-5}, 10^{-4}, 10^{-3}\}$ for the standard $f_{\text{lag}}(\cdot)$; the order of the truncated matrices $f_{\text{lag}}(\hat{\mathbf{Q}}[n])$ and $f_{\text{row}}(\hat{\mathbf{Q}}[n])$ are shown in Fig. 4.4 (a) for $\mu' = \mu$ and (b) for $\mu' = 6 \cdot \mu$ in the proposed $f_{\text{row}}(\cdot)$. As indicated in the example in Fig. 4.2, the proposed approach achieves a significant reduction in the order of the paraunitary matrices after truncation; particularly when $\mu' = 6 \cdot \mu$. In Fig. 4.4 (a) the paraunitary order for the row-shift truncation is less than the lag based truncation despite having a lower error in Fig. 4.3 (a). When the truncation parameter is compensated and $\mu' = 6 \cdot \mu$ the smaller truncation parameters ($\mu' = 6 \cdot 10^{-5}$ $\mu' = 6 \cdot 10^{-4}$) actually outperform the lag based method with larger truncation parameters ($\mu = 10^{-4}$ $\mu = 10^{-3}$) despite a significantly smaller error in Fig. 4.3 (b).

Row-Shift Truncation for Other PEVD Algorithms

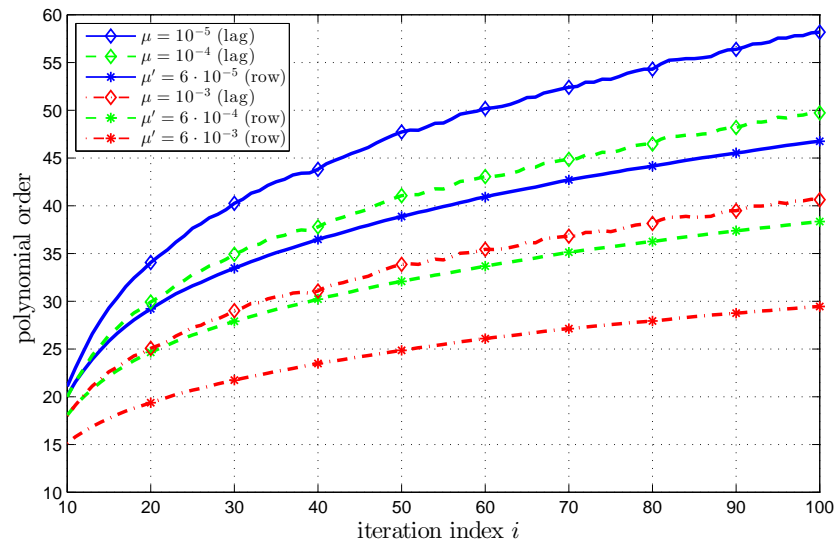
The previous sub-sections have discussed the benefits of row-shift truncation when used with the SBR2 algorithm, here it is applied to MS-SBR2, SMD, and MSME-SMD. To simplify the plots with the additional algorithms the simulations are shown for $\mu = 10^{-3}$ only.

The reconstruction error for each of the different algorithms is shown in Fig. 4.5. Despite the different PEVD algorithms used to produce the paraunitary matrices the reconstruction error is very similar for the different truncation schemes. Fig. 4.5 (a) shows the case where $\mu' = \mu$ and Fig. 4.5 (b) shows the same plot for $\mu' = 6 \cdot \mu$. The plots follow a similar trend to those in Fig. 4.3 with the compensation of $\mu' = 6 \cdot \mu$ giving a similar error. Comparing the different PEVD algorithms in both Fig. 4.5 (a) and (b), the more powerful PEVD algorithms (MSME-SMD, SMD & MS-SBR2) tend to reach their ‘error quota’ slightly faster than SBR2.

Given that each algorithm transfers a different amount of energy at each iteration it is most appropriate to show the cost, in terms of paraunitary matrix order, against diagonalisation measure. Fig. 4.6 shows the ensemble averaged paraunitary filter bank order vs. diagonalisation measure for each of the PEVD algorithms for (a) $\mu' = \mu$ and (b) $\mu' = 6 \cdot \mu$. Contrary to Fig. 4.4 (a) for the PEVD algorithms other than SBR2

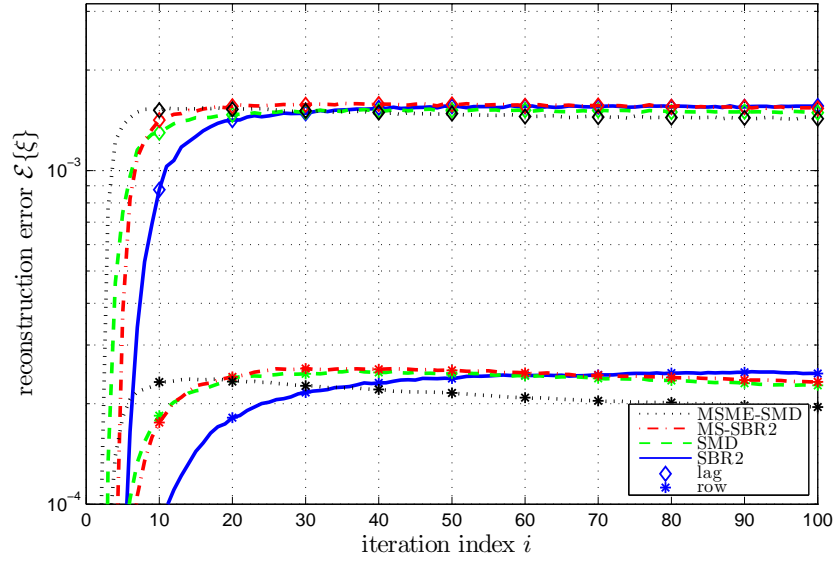


(a)

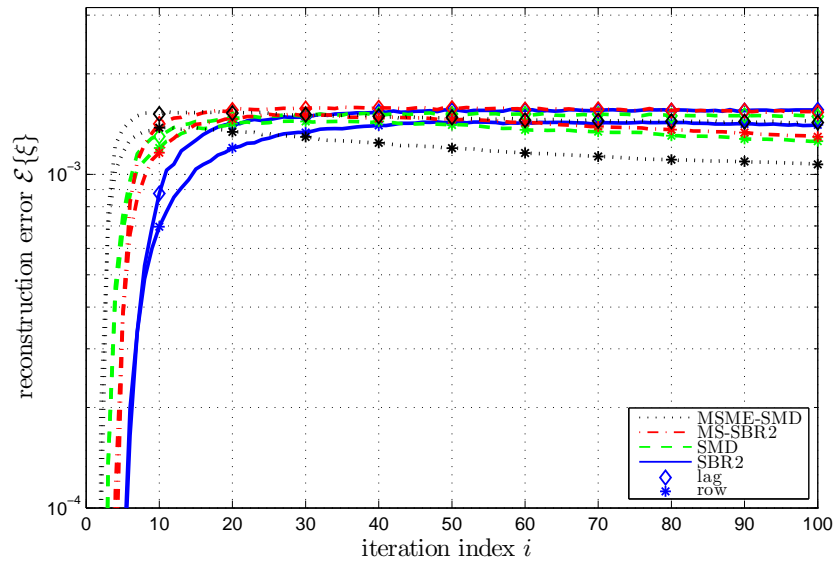


(b)

Figure 4.4: Average order after truncation of $Q(z)$ vs. SBR2 iterations for the different truncation approaches and varying μ (a) $\mu' = \mu$ and (b) $\mu' = 6 \cdot \mu$.



(a)



(b)

Figure 4.5: Ensemble reconstruction error $\mathcal{E}\{\xi\}$ vs. PEVD iterations for the different truncation approaches and PEVD algorithms (a) $\mu' = \mu$ and (b) $\mu' = 6 \cdot \mu$.

the row-shift truncation performs worse than the lag based truncation; particularly for the SMD algorithms. With $\mu' = 6 \cdot \mu$ in Fig. 4.6 (b) the performance of the row-shift truncation improves and is able to out perform the lag based approach for all PEVD algorithms, now replicating the results Fig. 4.4. It is interesting to note that for a set level of diagonalisation the SBR2 algorithm benefits most from the row-shift truncation. The extra benefit is likely due to the simple nature of the SBR2 row shifts and Jacobi rotations tending to leave outliers in the paraunitary matrix which are easily corrected with the row based truncation.

4.2 Restricted Search Algorithms

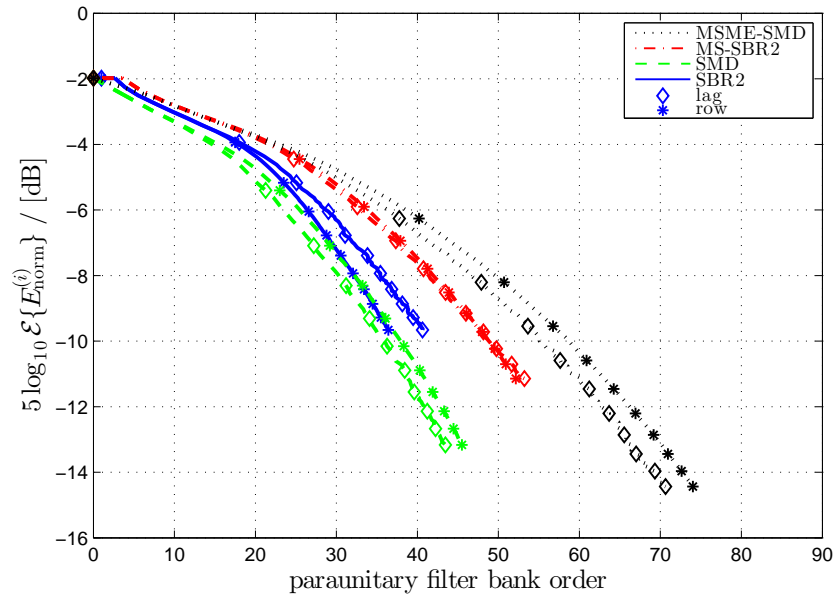
In both Chapter. 3 and Sec. 4.1 the high order growth of the multiple shift algorithms has been shown to be an issue. In this section the aim is to explain why the polynomial matrices in these algorithms grow faster and develop methods to slow their growth.

4.2.1 Polynomial Order Growth

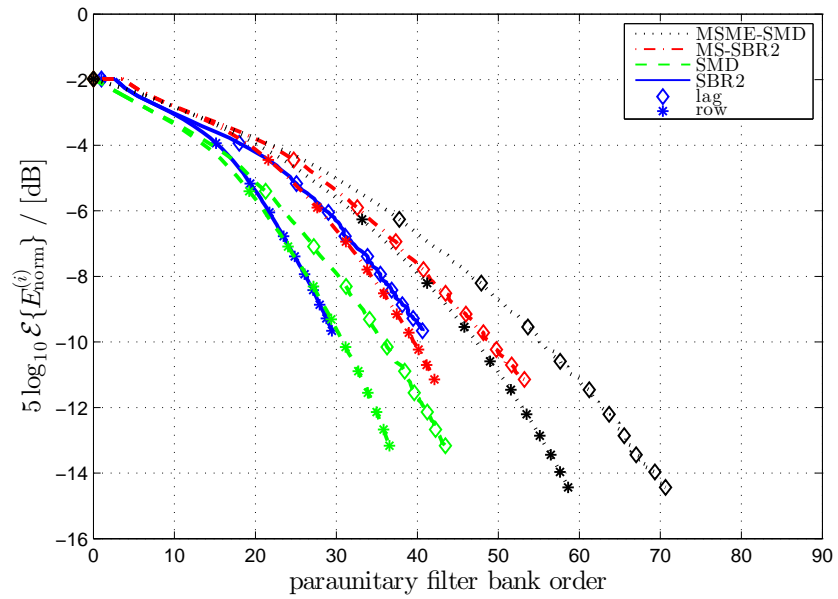
This section analyses the worst case polynomial order growth for PEVD algorithms. The analysis below assumes we have a parahermitian matrix, $\mathbf{S}^{(i-1)}(z)$, at the i -th iteration with a size of $M \times M \times 2L + 1$, i.e. the maximum lag in either direction is $|L|$. Visual examples are also included for the case where $\mathbf{S}^{(i-1)}(z)$ has a size of $5 \times 5 \times 5$ i.e. $M = 5$, $L = 2$.

The growth in order of the parahermitian, $\mathbf{S}^{(i-1)}(z)$, and paraunitary, $\mathbf{Q}^{(i-1)}(z)$, matrices is determined by the order of the shift matrix, $\mathbf{\Lambda}^{(i)}(z)$. To help analyse the problem the minimum shift that can bring any non-zero element onto the zero lag is defined as Δ_{\max} . Based on the shift length Δ_{\max} the highest possible order for the shift matrix is defined as Λ_{\max} . Larger shifts than Δ_{\max} can be applied to the parahermitian matrix but these will either be outside the initial range of L or can be synthesised by a shift matrix with shifts no larger than Δ_{\max} .

A visual example for a $5 \times 5 \times 5$ parahermitian matrix shown in Fig. 4.7; the diagonal elements are all shown in black to indicate that they are not included in any of the



(a)



(b)

Figure 4.6: Diagonalisation metric vs. average order of $\hat{Q}(z)$ after truncation for different PEVD algorithms (a) $\mu' = \mu$ and (b) $\mu' = 6 \cdot \mu$.

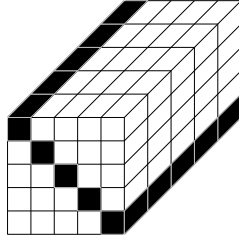


Figure 4.7: Initial $5 \times 5 \times 5$ parahermitian matrix.

search steps. In the following subsections the current search space is denoted by yellow elements, elements outside the current search space are indicated in white and the elements that cannot be searched due to a conflict are shown in red. It is unlikely that the algorithms will ever result in their worst case order growth but these do give an insight into how the matrices grow with the row/column shift interactions which is not always easily visualised.

Single Shift Algorithms

For the (single shift) SBR2, ME-SMD and SMD algorithms the growth is simply determined by the lag location, $\tau^{(i)}$, of the element/column found in (2.18)/(2.21). The lag parameter $\tau^{(i)}$ can have a maximum value of L , therefore $\Delta_{\max} = L$. With $\Delta_{\max} = L$ the maximum shift matrix length is $\Lambda_{\max} = L$. When $\mathbf{\Lambda}^{(i)}(z)$ of order Λ_{\max} is applied to $\mathbf{S}^{(i-1)}(z)$ it's 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. The parahermitian matrix, $\mathbf{Q}^{(i-1)}(z)$, will grow by Λ_{\max} or in this case L as the delay matrix is only applied once to $\mathbf{Q}^{(i-1)}(z)$.

As a visual example the worst case scenario occurs when the element or column of interest is located on the outermost lag e.g. element 'a' in Fig. 4.8 (a) for the maximum element case. The effect of moving element 'a' is shown in Fig. 4.8 (b) where the resulting parahermitian matrix is $5 \times 5 \times 9$ i.e. the matrix has grown by $2L$, as indicated in the analysis above.

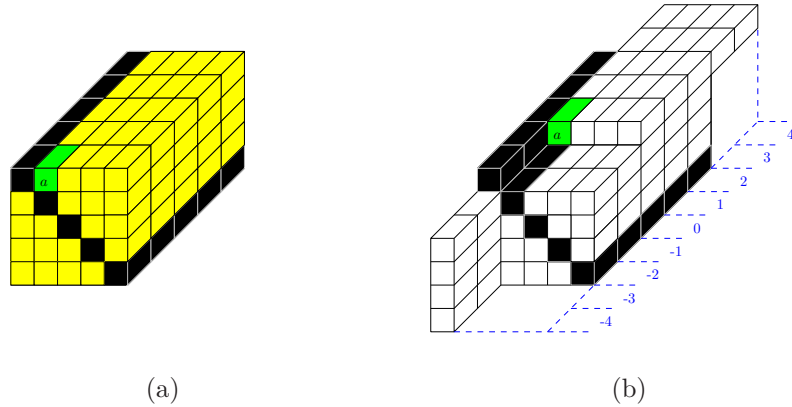


Figure 4.8: The first (a) and second (b) steps of a single shift algorithm.

$M - 1$ Multiple Shift SMD Algorithm

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 zero lag in a single iteration using $\mathbf{\Lambda}^{(i)}(z)$ therefore the maximum shift matrix order, Λ_{\max} , is $2\lceil ((M - 1)L)/2 \rceil$. As with single shift algorithms, when $\mathbf{\Lambda}^{(i)}(z)$ of order Λ_{\max} is applied to the parahermitian matrix, $\mathbf{S}^{(i-1)}(z)$, its order will increase by $2\Lambda_{\max}$. For the MSME-SMD algorithm the worst case parahermitian order growth is $4\lceil ((M - 1)L)/2 \rceil$. Applying $\mathbf{\Lambda}^{(i)}(z)$ of order Λ_{\max} to the paraunitary matrix results in an order growth of $2\lceil ((M - 1)L)/2 \rceil$. Even with reasonably small increases of M , the multiple shift algorithm can result in a significant increase in the worst case polynomial order growth.

For the MSME-SMD algorithm a worst case scenario occurs when the $M - 1$ elements are aligned just above the diagonal as shown by elements 'a'-'d' in Fig. 4.9 (a). As with the single shift example above the first element found is 'a' and like before this is shifted onto the zero lag as in Fig. 4.9 (b). In Fig. 4.9 (b) the elements that share a (length wise) fibre with 'a' are shown in red and excluded from the search as any further shifts on these would result in 'a' being pushed off the zero lag. To guarantee that we achieve the maximum quota of $M - 1$ shifts we then have to search the elements in yellow (this includes 'b'). Searching outside the yellow elements would mean that the maximum quota of shifts would not be achieved. Care must be taken when element 'b' is shifted

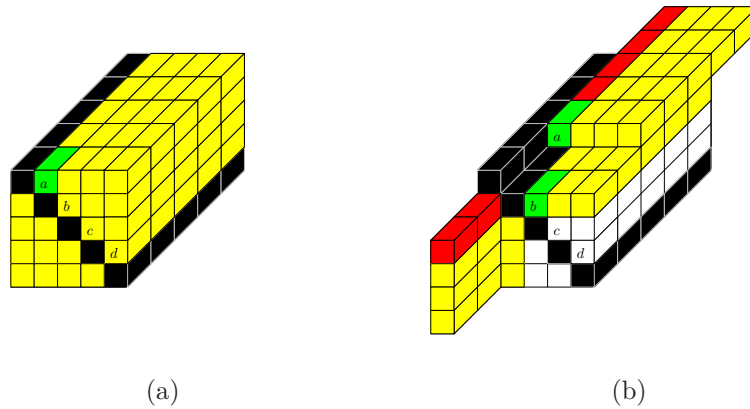


Figure 4.9: The first (a) and second (b) steps of the MSME search algorithm.

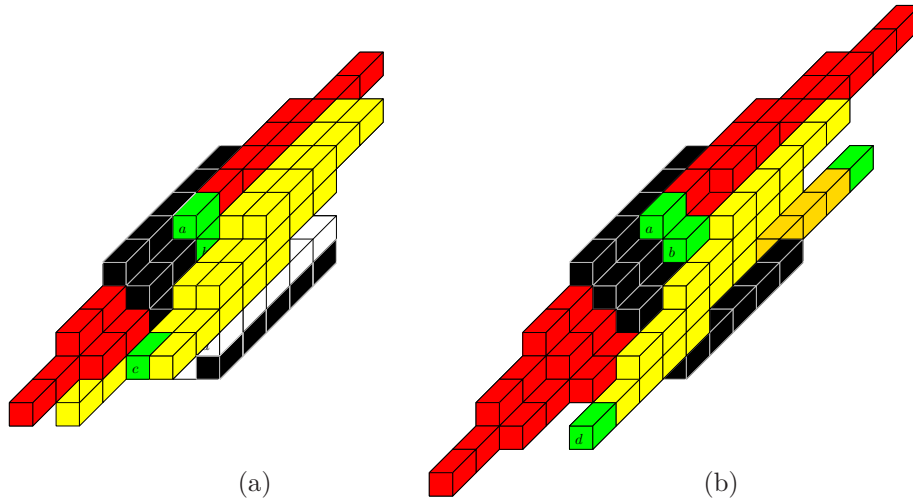


Figure 4.10: The third (a) and fourth (b) steps of the MSME search algorithm.

as a row shift would result in displacing ‘*a*’; therefore ‘*b*’ must be delayed column-wise - this is equivalent to advancing the 3rd row.

Fig. 4.10 (a) shows the result of moving element ‘*b*’ onto the zero lag. As we can see in Fig. 4.10 (a) the red exclusion has grown to include those elements that would affect ‘*b*’. The yellow search area has now moved over a new set of elements which again ensure $M - 1$ shifts can be done. Unfortunately the column delay of element ‘*b*’ has resulted in the next element ‘*c*’ actually being shifted further away from the zero lag. In order to get ‘*c*’ onto the zero lag its column must be delayed by 4, whereas both previous shifts were only by 2. The result of moving ‘*c*’ onto the zero lag is shown

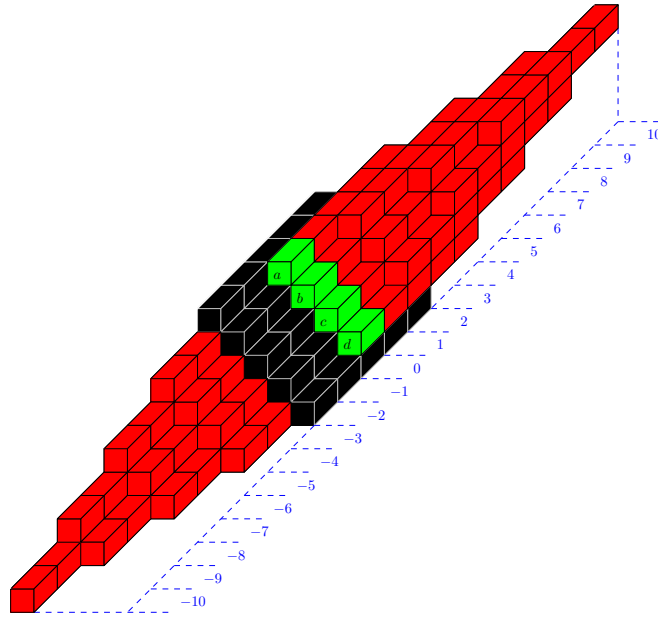


Figure 4.11: The fifth and final step of the MSME search algorithm.

in Fig. 4.10 (b). Again the exclusions shown in red have increased in number and the yellow search space now includes all other elements. As with element ‘c’, the previous shift has resulted in the next element ‘d’ being pushed away from the zero lag and so it now requires a delay of 6 elements. The result of this final shift is shown in Fig. 4.11. The overall length of the para-hermitian matrix has now grown (by $4\lceil((5-1)2)/2\rceil = 16$) from 5 to 21 i.e. L has increased (by $2\lceil((5-1)2)/2\rceil = 8$) from 2 up to 10.

For the example in Figs. 4.9 – 4.11 the largest shift required is $\tau = 6$, which seems to contradict $\Delta_{\max} = \lceil((5-1)2)/2\rceil = 4$. However, the shifts in this example are predominantly due to advancing rows 3, 4 and 5 i.e. $\Lambda^{(i)}(z) = \text{diag}\{z^{-2} \ 1 \ z^2 \ z^4 \ z^6\}$. Using the delay matrix ambiguity from Sec. 2.5.1 these shifts can be balanced such that there is the same number of advances as delays. Multiplying by z^{-2} then we get $z^{-2}\Lambda^{(i)}(z) = \text{diag}\{z^{-4} \ z^{-2} \ 1 \ z^2 \ z^4\}$ whose Δ_{\max} is now only 4 but provides exactly the same para-hermitian matrix as is in Fig. 4.11.

$\lfloor M/2 \rfloor$ **Multiple Shift SBR2 Algorithm.**

In the multiple shift SBR2 algorithm the masking of the shifts is done in such a way that the shifts will not interact with one another so the maximum shift length is the

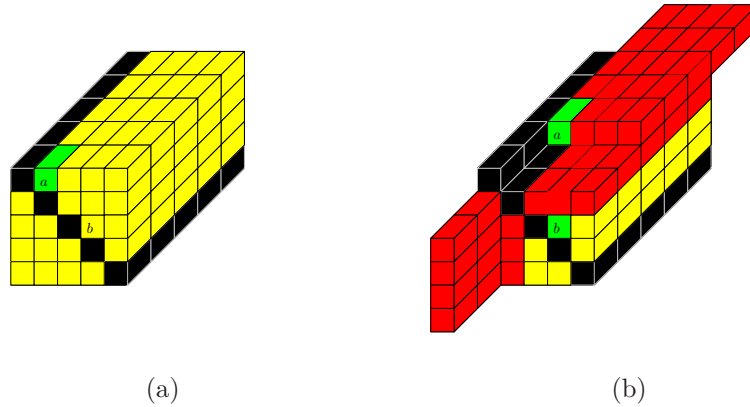


Figure 4.12: The first (a) and second (b) steps of the MS-SBR2 search algorithm.

same as the single shift algorithms i.e. $\Delta_{\max} = L$. As the multiple shifts can potentially consist of both advances or delays the maximum shift matrix length, Λ_{\max} is $2L$. As with all the algorithms the growth in the parahermitian matrix, $\mathbf{S}^{(i-1)}(z)$, is twice Λ_{\max} and therefore $4L$. The growth of the paraunitary matrix is simply the same as Λ_{\max} , which is $2L$.

Graphically a worst case scenario for the multiple shift SBR2 search method is similar to that of the MSME-SMD scenario but only two elements can be moved in the $5 \times 5 \times 5$ case. The diagram in Fig. 4.12 (a) shows the worst case scenario for multiple shift SBR2. Like the previous algorithms the first search, Fig. 4.12 (a), has no restrictions and upon finding element ‘a’ and shifting it onto the zero lag we end up with Fig. 4.12 (b). The search space is restricted by the fact that the Jacobi transforms interact with the rows and columns of element ‘a’ and so future elements must not occupy these locations. The restriction for the next multiple shift SBR2 element is then the lower right sub-matrix from which element ‘b’ is chosen. Using the original specification of the multiple shift SBR2 algorithm element ‘b’ can be shifted either in its column or row. The worst case growth happens when the row/column choice is the opposite of ‘a’ - in this case delaying the column of ‘b’. Fig. 4.13 shows the overall worst case scenario for the multiple shift SBR2. In this case the matrix has grown to 13 lags, although this can be improved with careful shift selection.

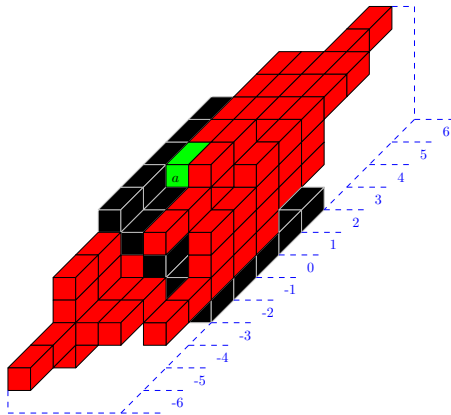


Figure 4.13: The final step of the MS-SBR2 search algorithm.

4.2.2 Restricted Search Algorithms

From Sec. 4.2.1 above, the examples of the worst case polynomial order growth have highlighted one of the problems with the multiple shift strategies. This section introduces two simple modifications that can be implemented to reduce the polynomial order growth in both multiple shift algorithms.

Restricted Search MSME-SMD

In the restricted search (RS) MSME-SMD we impose an extra condition on the standard MSME-SMD search spaces in Fig. 3.8 to control the polynomial order growth in both $\mathcal{S}^{(i-1)}(z)$ and $\mathcal{Q}^{(i-1)}(z)$. 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 (2.18) but now once the first search of the i th iteration finds a maximum element on $\tau_{k^{(i)}}^{(i)}$, the lag parameter, τ , in (2.18) is restricted such that $|\tau| \leq |\tau_{k^{(i)}}^{(i)}|$ for the remaining searches of 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 parahermitian order growth is $4L$; crucially these parameters no longer include the matrix dimension M .

The restricted search multiple shift maximum element search algorithm follows a similar approach to the original multiple shift maximum element method and so the

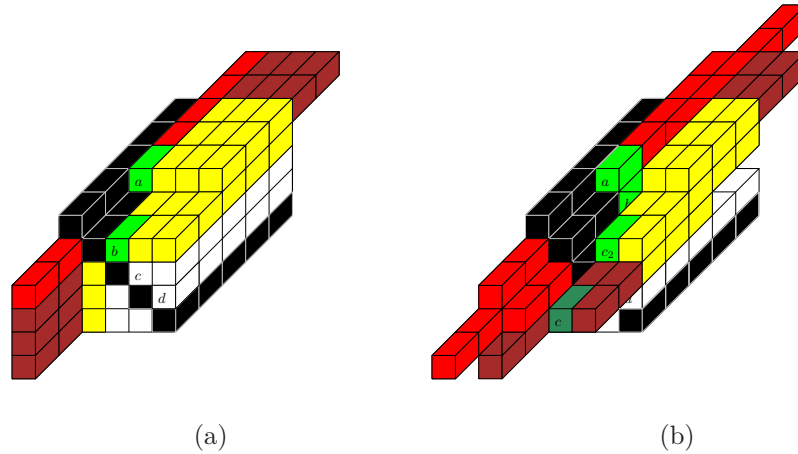


Figure 4.14: The first (a) and second (b) steps of the RS-MSME search algorithm.

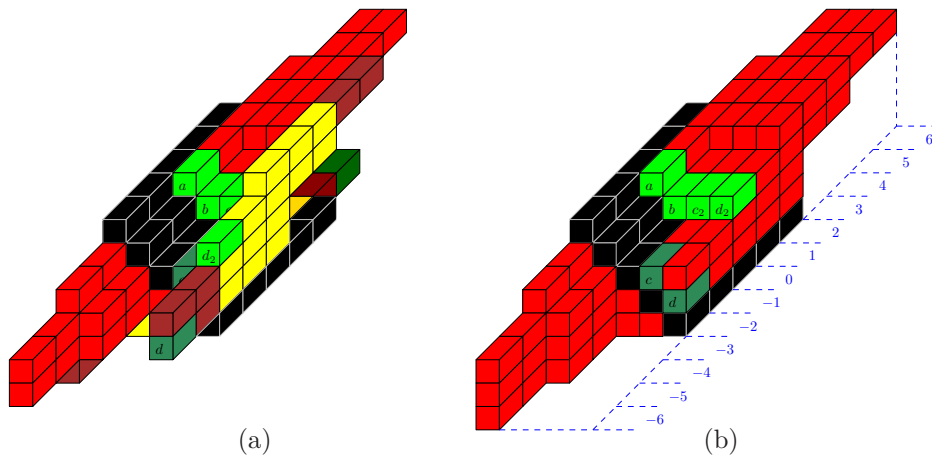


Figure 4.15: The third (a) and fourth (b) steps of the RS-MSME search algorithm.

first step, Fig. 4.14 (a), is identical to that in Fig. 4.9 (b). The restriction occurs once element ‘ a ’ has been shifted and the parahermitian matrix has increased to 9 lags as shown in Fig. 4.14 (a). Now the elements shown in brown, which are outside the original footprint of the parahermitian matrix, are no longer included in the search space. In the case of Fig. 4.14 (a) we can still select element ‘ b ’ so this is no different to the original MSME-SMD choice. The result of shifting ‘ b ’ for the RS-MSME-SMD method is shown in Fig. 4.14 (b). The shift of element ‘ b ’ means that element ‘ c ’ now resides in one of the brown sections therefore it can no longer be selected; instead an alternative element, ‘ c_2 ’, is chosen from the remaining yellow elements. The result of shifting element ‘ c_2 ’ is shown in Fig. 4.15 (a). Like the previous step we see that

element ‘ d ’ has been moved outside the footprint of the original parahermitian matrix so element ‘ d_2 ’ is chosen from the yellow search area. When ‘ d_2 ’ is brought onto the zero lag we get the scenario shown in Fig. 4.15 (b). Clearly the elements now on the zero lag will be lower values than what could have been achieved however the length of the parahermitian matrix is significantly less at 13 rather than 21 lags.

Ultimately limiting the search space to lower lags will result in missing some elements and slow the algorithm’s convergence 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 where the restriction, $|\tau| \leq |\tau_k^{(i)}|$, is applied will be on fewer elements and second the slower growth in parahermitian matrix means searches and matrix operations in future iterations will also be over fewer elements.

Order Controlled MS-SBR2

The ideas from the RS-MSME-SMD algorithm have also been employed in MS-SBR2 to create the order controlled (OC) MS-SBR2 algorithm [61]. For the standard MS-SBR2, with no consideration on polynomial growth, the maximum order growth is $4L$ at each iteration. By simply restricting subsequent shifts to follow the initial shift at each iteration the maximum order growth is reduced to just $2L$. As an added benefit the OC-MS-SBR2 approach does not restrict the elements which can be selected so the same energy can be transferred at each iteration. For reference the OC-MS-SBR2 algorithm has been used throughout this thesis so no results or analysis for the (non order controlled) MS-SBR2 algorithm are presented.

The OC-MS-SBR2 is almost identical to the standard MS-SBR2 method described above however when element ‘ b ’ is shifted from Fig. 4.16 (a) the row/column selection must follow that used in ‘ a ’. The result of the OC-MS-SBR2 is shown in Fig. 4.16 (b). The matrix now only has 9 lags (rather than 13) and so the general worst case scenario for OC-MS-SBR2 is $2L$ which is the same as the single shift SBR2 algorithm.

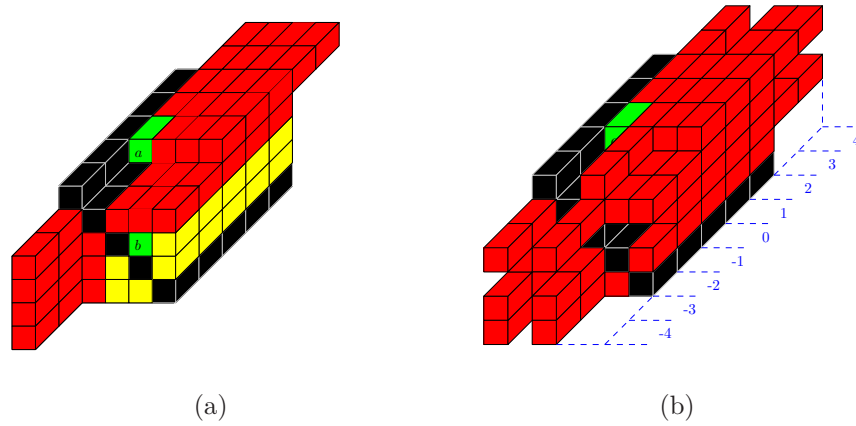


Figure 4.16: The first (a) and second (b) steps of the OC-MS-SBR2 search algorithm.

Table 4.1: Summary of worst case polynomial order growth for the different PEVD algorithms.

	Δ_{\max}	Λ_{\max}	PH Growth
Single Shift	L	L	$2L$
MSME-SMD	$\lceil ((M-1)L)/2 \rceil$	$2\lceil ((M-1)L)/2 \rceil$	$4\lceil ((M-1)L)/2 \rceil$
MS-SBR2	L	$2L$	$4L$
RS-MSME-SMD	L	$2L$	$4L$
OC-MS-SBR2	L	L	$2L$

Comparison

For comparison the maximum shift, shift matrix order and parahermitian order growth are summarised in Tab. 4.1 for all five 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. For the MS-SBR2 algorithm the worst case order growth has gone from $4L$ to only $2L$, i.e. the same as the single shift algorithms.

4.2.3 Results

To illustrate the performance of the different MSME-SMD algorithms first the performance metrics are introduced, followed by the simulation set up. Finally the results

are presented and the performance of the PEVD algorithms is analysed.

Performance Metrics

To confirm that the RS-MSME-SMD shifts a similar amount of energy at each iteration as the original MSME-SMD; the first test will measure diagonalisation performance using (3.21). The main objective of the search space restriction 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.

An added benefit of the reduction in parahermitian matrix order is a reduction in the computational cost of calculating the PEVD. Here execution time is used 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.

Simulation Set Up

The results were obtained using an ensemble of 10^3 parahermitian matrices produced using the source model from [10] where the source model has an average dynamic range of approximately 20 dB. The source model is randomised so that the parahermitian matrices produced are unique for each instantiation. The parahermitian matrix, $\mathbf{R}(z)$, is $\mathbf{R}(z) \in \mathbb{C}^{6 \times 6}$ with the initial number of lags set to 119. Each of the PEVD algorithms was run for 100 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 $\mu_{PH} = \mu_{PU} = 10^{-4}$.

Algorithm Convergence

Fig. 4.17 shows the reduction in off-diagonal energy vs. algorithm iterations for SBR2, MS-SBR2, SMD, and the two versions of MSME-SMD. The amount of energy transferred by both MSME-SMD algorithms is significantly higher than the SMD and SBR2 methods. Despite its search space restriction the RS-MSME-SMD algorithm actually marginally outperforms the original MSME-SMD. The performance improvement is

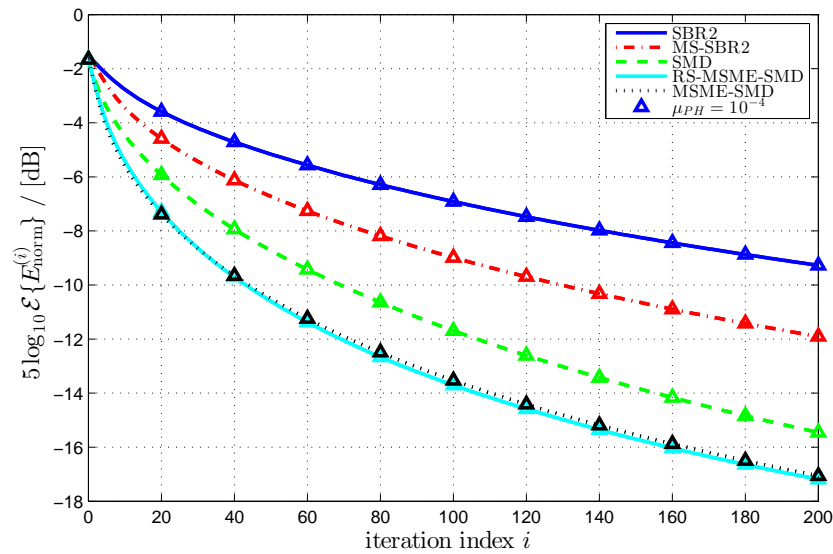
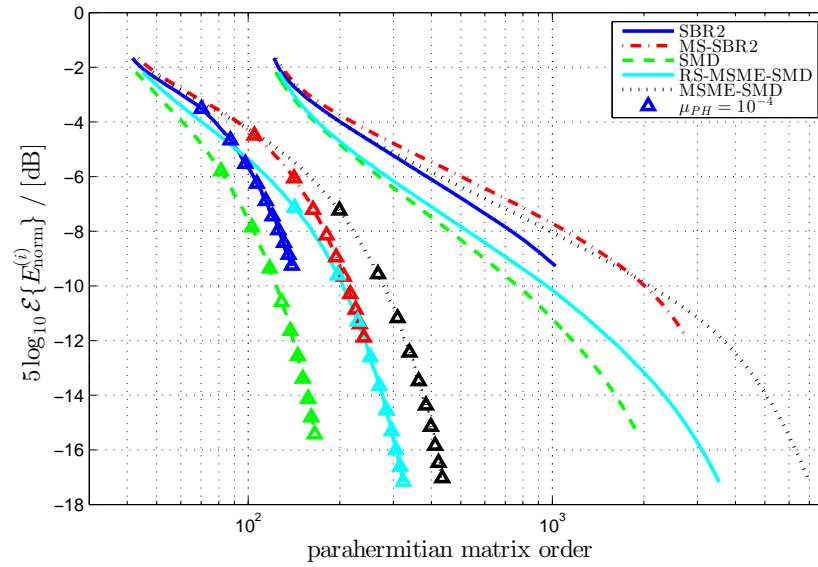


Figure 4.17: Diagonalisation vs. algorithm iterations for the SMD algorithm and the two MSME-SMD variants.

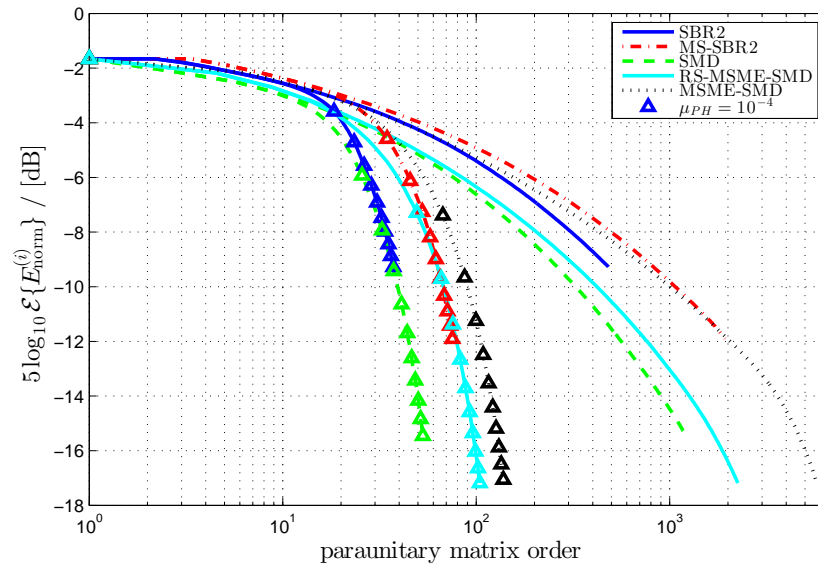
likely due to the RS-MSME-SMD algorithm selecting elements in a more systematic order than MSME-SMD.

Paraunitary/Parahermitian Matrix Order

This section investigates the main goal of the RS-MSME-SMD algorithm which is to reduce the growth in polynomial order of the parahermitian and paraunitary matrices. Figs. 4.18 (a) & (b) show the order growth of the parahermitian and paraunitary matrices for each of the selected PEVD algorithms for the case where only zero filled lags are removed and where the matrices have been truncated (shown with triangle markers). Here we can see in both Fig. 4.18 (a) & (b) that the matrices produced by RS-MSME-SMD are significantly shorter than their MSME-SMD equivalent and are a similar level to SMD and SBR2. Even when a truncation algorithm such as those described in [55] and Sec. 4.1 are applied to the parahermitian and paraunitary matrices the reduced search method still out-performs the original MSME-SMD as shown in Fig. 4.18 (a) & (b) although it does lose out slightly to SMD and SBR2.



(a)



(b)

Figure 4.18: Reduction in off-diagonal energy vs. growth in (a) parahermitian matrix and (b) paraunitary order.

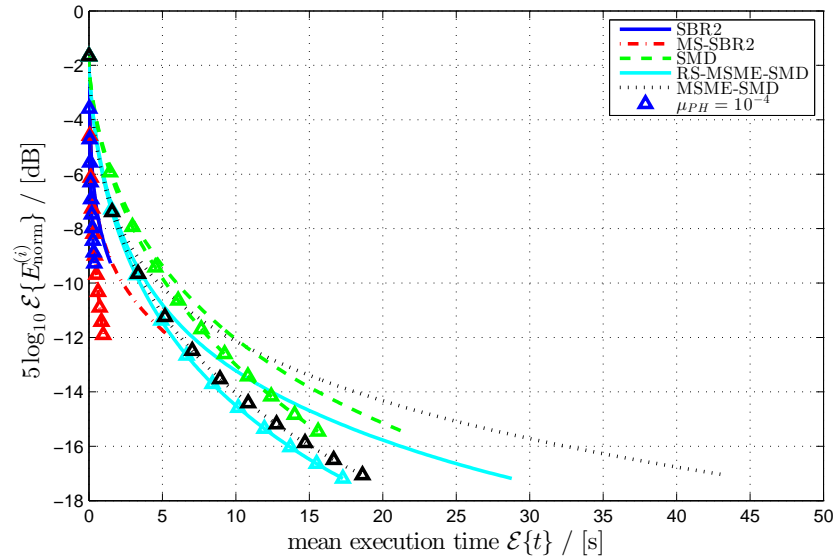


Figure 4.19: Real time convergence of PEVD algorithms, diagonalisation measure vs. mean execution time.

Real Time Execution

Fig. 4.19 shows the time taken for each of these algorithms to carry out 200 iterations alongside the diagonalisation measure at each point. When no truncation is used the new reduced search MSME method is more efficient than the original MSME search, in fact the new method takes on average around 15 seconds less than its predecessor to complete 200 iterations. When the parahermitian truncation methods are included both MSME-SMD variants obtain a significant performance improvement, whereas the same change in SMD has a lesser effect. The performance benefits of the reduced search MSME-SMD are not as obvious when the parahermitian truncation is used but it still performs better than the original MSME-SMD. Although RS-MSME-SMD performs better than the other SMD algorithms it still converges significantly slower than the SBR2 based algorithms.

4.3 Cyclic-by-Row PEVD Approximation

Sec. 2.4 identified the computational cost to compute an approximate PEVD via SMD algorithms as a potential obstacle. In experiments using the Matlab profiler, the cal-

ulation and application of an EVD per iteration step was singled out as the major contributor to this high cost. Therefore, this section proposes an inexpensive numerical approximation of the EVD by a Jacobi sweep consisting of a limited number of Jacobi transformations in a cyclic-by-row approach [7, 62]. Below, Sec. 4.3.1 reviews the cyclic-by-row Jacobi algorithm and Sec. 4.3.2 outlines the general procedure, which is then applied to a number of SMD algorithms in Sec. 4.3.3 with results presented in Sec. 4.3.4.

4.3.1 EVD Approximation

As discussed in Sec. 2.1 a number of iterative (scalar) EVD algorithms exist. One of the simplest is the cyclic-by-row Jacobi algorithm which systematically zeros the off diagonal elements in what is termed a Jacobi sweep. Each Jacobi sweep consists of $(M^2 - M)/2$ Jacobi transformations applied in the sequence shown in Fig. 2.2. Each Jacobi transformation as defined in (2.3) will transfer the energy of an off-diagonal element onto the diagonal while undoing some of the work of previous Jacobi transformations. However, over the course of one Jacobi sweep, the off-diagonal energy is reduced. By performing more Jacobi sweeps, the accuracy of the EVD approximation is improved.

4.3.2 PEVD Approximation

Iterative approximate PEVD algorithms such as SBR2 and SMD minimise off-diagonal energy until a predefined threshold ρ is reached, as described by (2.19). Therefore, within one iteration step of SMD, a full EVD with a suppression of off-diagonal energy to the numerical equivalent of zero appears to be an overkill, and a lower precision with a limited number of Jacobi sweeps will very likely suffice to achieve the task of reducing off-diagonal energy below the value ρ .

Experimentation has shown that for the approximation of the SMD algorithms detailed below, a single cyclic-by-row Jacobi sweep proved sufficient and provided the best cost-performance trade-off, as we will detail in Sec. 4.3.4. With this approach, the unitary Jacobi transformation matrix $\mathbf{Q}^{(i)}$ from SBR2 in (2.3) now becomes the

product of $N = (M^2 - M)/2$ Jacobi transformations,

$$\mathbf{Q}^{(i)} = \prod_{n=1}^N \mathbf{Q}^{(i,n)} \quad , \quad (4.11)$$

where $\mathbf{Q}^{(i,n)}$ is the n th Jacobi transformation used in the i th iteration of an iterative PEVD algorithm using the single sweep cyclic-by-row approach.

4.3.3 Cyclic-by-Row SMD Algorithms

The cyclic-by-row single Jacobi sweep approximation of the EVD can be embedded in all algorithms of the SMD family. It may be argued that the term sequential matrix *diagonalisation* is no longer appropriate, as the approximate EVD also results in only an approximate diagonalisation, and algorithms will therefore share some properties of SBR2, where only part of the off-diagonal energy of the zero lag matrix is transferred onto the diagonal. However, we assume that the approximation is within the bound ρ for off-diagonal energy, and that therefore the term *diagonalisation* is justified within the SMD family's limited, pre-defined accuracy of decomposition.

All SMD algorithms perform an initial diagonalisation by an EVD according to (2.20), which in the cyclic-by-row version is approximated by a single Jacobi sweep. The EVD in subsequent iterations is also replaced by a single Jacobi sweep, and the unitary matrix (4.11) as applied in (2.14) can be implemented as a sequence of Jacobi transformations rather than a full matrix multiplication. The specific SMD family versions therefore consistently apply the single Jacobi sweep approach, and only differ in the way columns and rows are identified for transfer to the lag zero matrix using (2.13) at the i th iteration:

- SMD [10]: in its original form, the sequential matrix diagonalisation algorithm transfers the column with the largest off-diagonal column norm onto the lag zero matrix;
- MSME-SMD (Sec. 3.2): the un-restricted multiple-shift version that transfers $(M - 1)$ columns identified by their maximum elements;

- RS-MSME-SMD (Sec. 4.2.2): the restricted multiple-shift version, which still transfers $(M - 1)$ elements.

The cyclic-by-row approximations of these algorithms will be compared to their standard versions as well as SBR2 and MS-SBR2 in the next section.

4.3.4 Results

Performance Metrics

The key performance criteria when evaluating the cyclic-by-row approximations of SMD algorithms are the execution time, measured on the same computer set up as Sec. 4.2.3, and the diagonalisation measure from (3.21) that has been used in most of the experiments so far.

Simulation Scenario

Like other results sections an ensemble of 10^3 parahermitian matrices have been produced using the source model from [10] with the dynamic range restricted to approximately 20 dB. Each parahermitian matrix is generated as $\mathbf{R}(z) \in \mathbb{C}^{6 \times 6}$ with the initial number of lags set to 119. Each of the PEVD algorithms was then run on the ensemble for 100 iterations and the performance metrics recorded after each iteration. The parahermitian matrix truncation parameter was set as $\mu_{PH} = 10^{-4}$. In addition to the SMD based algorithms and cyclic-by-row based equivalents the simulations include both SBR2 and MS-SBR2

Algorithm Convergence

The diagonalisation performance versus iterations is shown in Fig. 4.20 for the various algorithms. The SMD family generally transfers more energy at each iteration than SBR2, as also highlighted in [10], with the multiple-shift versions performing best. Interestingly, the cyclic-by row single Jacobi sweep approximations of the EVD lead to no noticeable performance degradation for the SMD family of algorithms, thereby confirming the single sweep selection in Sec. 4.3.2.

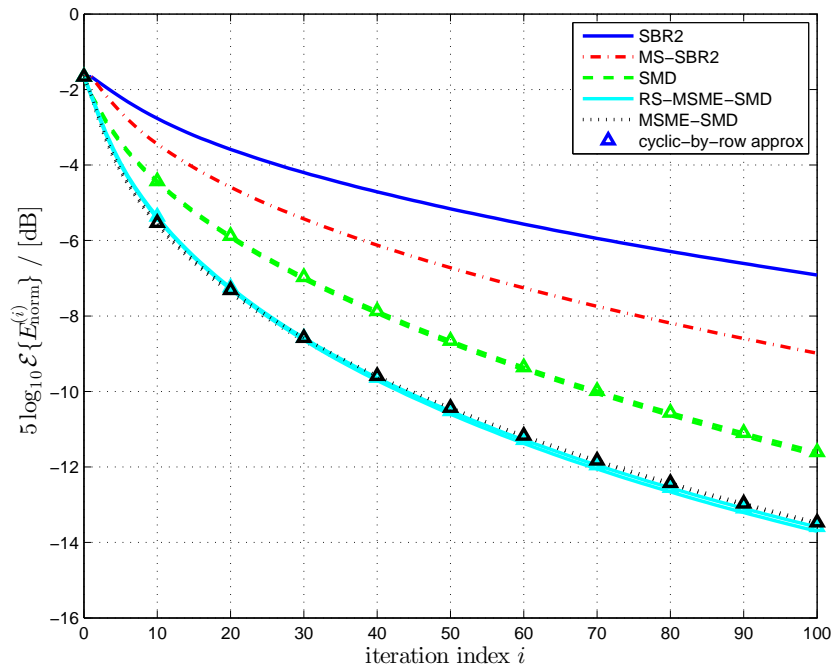
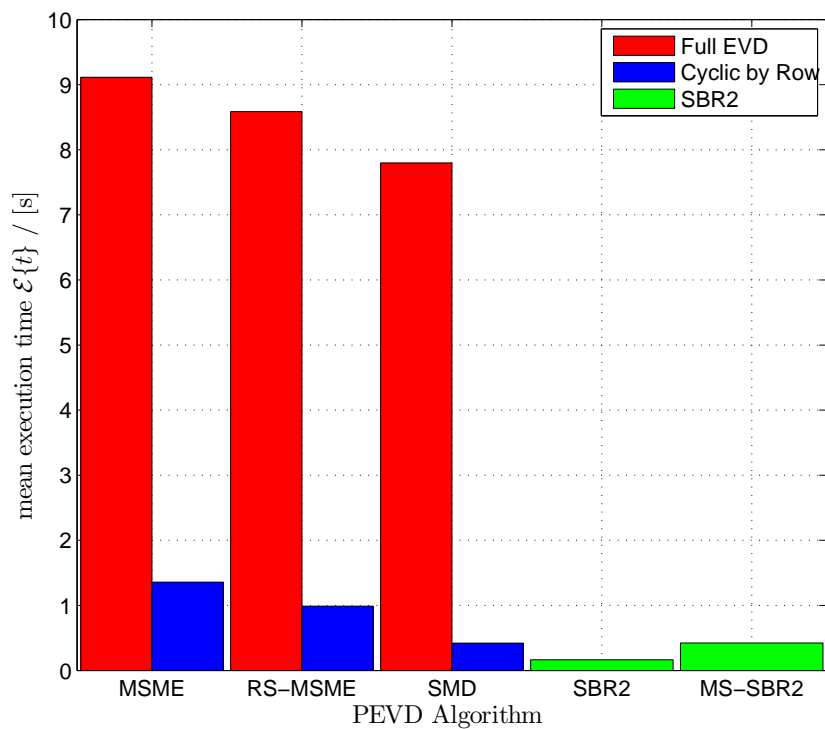


Figure 4.20: Normalised remaining off-diagonal energy $E_{\text{norm}}^{(i)}$ according to (3.21) for various iterative PEVD algorithms versus iterations.

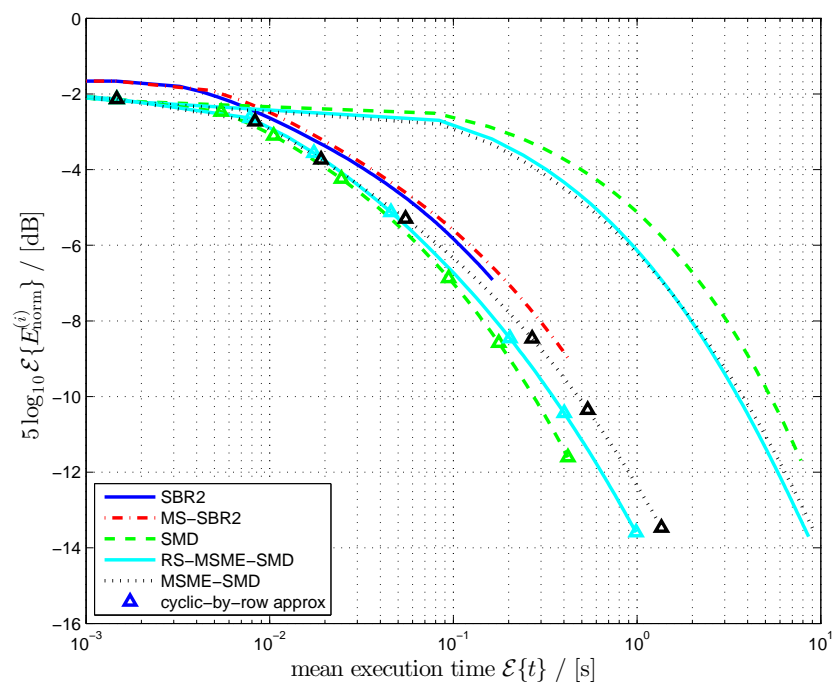
Real Time Performance

The computation time required for 100 iterations of the different iterative PVD algorithms is plotted in Fig. 4.21 (a). Compared to SBR2, the SMD algorithms require much more processing time. However, a significant reduction in cost can be noticed for the cyclic-by-row approximations of SMD algorithms. These are still more costly than the SBR2 algorithm, which only requires a single Jacobi transformation per iteration and therefore is guaranteed to have a lower complexity.

More interesting than the cost per iteration is the required execution time to reach a specific level of diagonalisation. Fig. 4.21 (b) shows the normalised remaining off-diagonal energy as a function of the time taken to calculate this specific decomposition. This graph is obtained by merging the information of Fig. 4.20 and 4.21 (a). Using a full EVD, the SMD family of algorithms are inferior to SBR2 when a limited diagonalisation suffices. For the best levels of diagonalisation, SBR2 cannot provide the required diagonalisation (within 100 iterations), and the computationally expensive SMD fam-



(a)



(b)

Figure 4.21: Execution times for PEVD algorithms (a) for 100 iterations and (b) against off-diagonal energy.

ily is the only option, with particular benefit for multiple-shift versions as established in [39, 40] and the previous results sections. Using the cyclic-by-row approximation, the actual cost to reach a specific level of diagonalisation is reduced below even what is required for SBR2. With the lower cost EVD step in the cyclic-by-row algorithms the simpler search of the single shift SMD converges slightly faster than the MSME-SMD methods.

4.4 Chapter Summary & Conclusions

The aim of this chapter was to improve the efficiency of PEVD algorithms and their resulting decompositions. First the truncation of paraunitary matrices was investigated. A new method was introduced that takes advantage of the paraunitary ambiguity outlined in Sec. 2.5 and is applicable to all PEVD algorithms (and more generally paraunitary matrices). Results show that the SBR2 algorithm gains the most from the new truncation approach, this is likely due to the fact that the single row/column shift and Jacobi transform on a single element encourage outliers that are easily corrected with this approach. With all algorithms the new truncation approach results in a significantly lower error compared to the existing approach when the same truncation parameters are used. The lower error characteristic allows the truncation parameters to be increased by six times whilst still maintaining a similar error. When the truncation parameters have been increased the benefit for the SBR2 algorithm is increased and the new approach becomes beneficial for all other PEVD algorithms.

Results from Chapter 3 show that the order growth of the polynomial matrices in the MSME-SMD algorithm is significantly higher than other algorithms. Through analysis of the polynomial order growth of the SMD and MSME-SMD algorithms a new search method which can significantly reduce the polynomial order growth of the MSME-SMD algorithm has been proposed. With the new search space restriction the maximum polynomial growth is decoupled from the spatial dimension of the parahermitian matrix, M ; this makes a significant difference as M is increased. Experiments have shown that the new method leads to a 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.

As indicated in Chapter 3 the major contributing factor to the cost of SMD based algorithms is the application of the full EVD to the entire parahermitian matrix. Here a new approach which approximates the EVD step using a series of Jacobi transformations is introduced. The EVD is approximated in a cyclic-by-row fashion which significantly reduces the computational cost. Despite the cyclic-by-row method being an approximation there is almost no difference in terms of algorithm convergence and the SMD based algorithms retain their superior energy transfer. The major difference comes in execution time where the cyclic-by-row methods take only a fraction of the time required for the SMD algorithms. Using the cyclic-by-row approach all SMD algorithms now converge faster in real time than the SBR2 based algorithms.

The efficient implementations presented in this chapter have brought the performance metrics of the current PEVD algorithms much closer together. Previously the SMD algorithm was able to produce the lowest order paraunitary matrices for a set level of diagonalisation however with the row-shift corrected truncation the SBR2 algorithm can match this. With a simple modification to the MSME-SMD algorithm there has also been a significant reduction in the order growth of its paraunitary and parahermitian matrices. For real time convergence the SBR2 algorithm, with its low complexity, has traditionally given the best results. However, employing the cyclic-by-row method introduced in this chapter all SMD based algorithms presented in this chapter outperform SBR2.

Chapter 5

Applications of Iterative PEVD Algorithms

The previous two chapters have investigated algorithmic aspects of the PEVD, in particular developing a better converging algorithm and methods that improve algorithm efficiency. In this chapter PEVD algorithms are used in three different scenarios. Sec. 5.1 investigates how the properties of the input data, given by a source model, impact the performance of the different PEVD algorithms [46]. Sec. 5.2 introduces the polynomial MUSIC algorithm and illustrates the benefits of using the new MSME-SMD algorithm over the SBR2 algorithm for the PEVD step [13]. In Sec. 5.3 the PEVD is used to extend the scalar generalised eigenvalue decomposition (GEVD) to polynomial matrices. The polynomial-matrix generalised eigenvalue decomposition (PGEVD) mimics the Cholesky method for the scalar generalised eigenvalue decomposition (GEVD) but in the polynomial matrix domain [47].

5.1 Implications of Source Model Conditioning

For the analysis and simulations in this section, we assume that the parahermitian matrices have a known ground truth decomposition. This enables us to control the condition of the problem that is addressed by the various PEVD algorithms, and also assess and compare the solution that is reached.

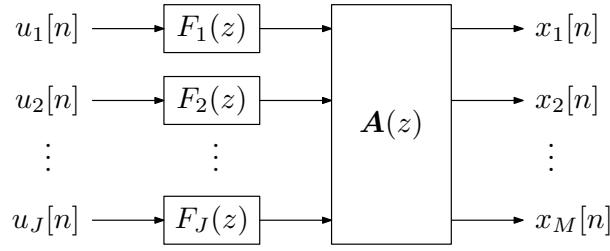


Figure 5.1: Source model with J unit variance zero mean uncorrelated complex Gaussian excitations $u_j[n]$, innovation filters with transfer functions $F_j(z)$, $j = 1 \dots J$, followed by a paraunitary convolutive mixing system $\mathbf{A}(z)$ generating M measurements $x_m[n]$, $m = 1 \dots M$.

5.1.1 Source Model Conditioning

The general model is depicted in Fig. 5.1. A total of J independent source signals with individual power spectral densities (PSDs) $F_j(z)\tilde{F}_j(z)$, $j = 1 \dots J$, are generated by exciting innovation filters $F_j(z)$ with unit variance zero-mean uncorrelated complex Gaussian sources $u_j[n]$ [63]. The order of the innovation filters $F_j(z)$ is P , and careful control of the filter gain and the maximum radius of zeros can determine the dynamic range of the source and whether they e.g. are spectrally majorised as in [10]. Convolutive mixing of the source signals is performed by a random paraunitary matrix $\mathbf{A}(z) \in \mathbb{C}^{M \times J}$ of order K , with $M \geq J$. This matrix is determined by extracting J columns from

$$\mathbf{A}'(z) = \prod_{k=1}^K (\mathbf{I} - \mathbf{v}_k \mathbf{v}_k^H + \mathbf{v}_k \mathbf{v}_k^H z^{-1}), \quad (5.1)$$

which is a product of K elementary paraunitary matrices [5], with $\mathbf{v}_k \in \mathbb{C}^M$, $k = 1 \dots K$, being random unit norm vectors.

The space-time covariance matrix constructed from the output vector $\mathbf{x}^T[n] = [x_1[n] \dots x_M[n]]$ is therefore given as

$$\mathbf{R}(z) = \sum_{\tau} \mathcal{E}\{\mathbf{x}[n]\mathbf{x}^H[n-\tau]\} z^{-\tau} \quad (5.2)$$

$$= \mathbf{A}(z)\mathbf{F}(z)\tilde{\mathbf{F}}(z)\tilde{\mathbf{A}}(z). \quad (5.3)$$

The diagonal matrix $\mathbf{F}(z) = \text{diag}\{F_1(z) \dots F_J(z)\}$ contains the J innovation filters.

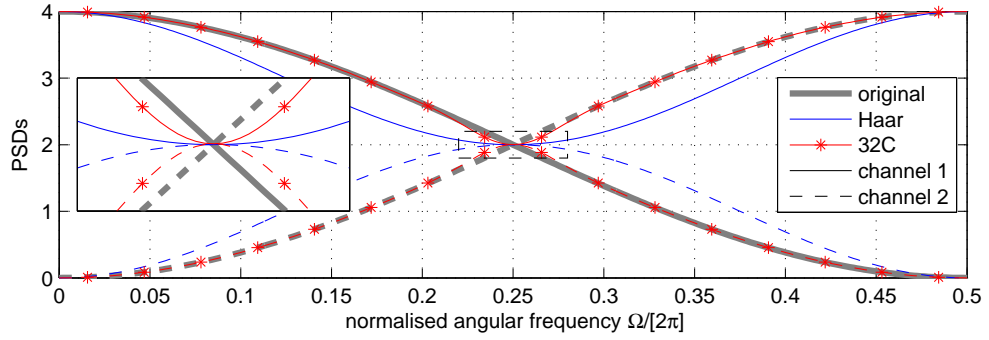


Figure 5.2: PSDs of unmajorised sources, and after frequency-reassignment using paraunitary matrices based on Haar [64] and 32C filters [65].

5.1.2 Polynomial Eigenvalue Decomposition

Given that the parahermitian matrix in (5.3) is factorised into paraunitary and diagonal parahermitian matrices, it bears close relation with the PEVD (2.9) of $\mathbf{R}(z)$. If $\mathbf{F}(z)$ is spectrally majorised, then indeed the PEVD $\mathbf{R}(z) = \tilde{\mathbf{Q}}(z)\mathbf{D}(z)\mathbf{Q}(z)$ exists with equality and is given by $\tilde{\mathbf{Q}}(z) = \mathbf{A}(z)$ and $\mathbf{D}(z) = \mathbf{F}(z)\tilde{\mathbf{F}}(z)$.

If $\mathbf{F}(z)$ is not spectrally majorised, then a PEVD satisfying both diagonalisation and spectral majorisation could be derived by re-assigning spectral components of $\mathbf{F}(z)$ via a paraunitary matrix $\mathbf{U}(z)$ such that $\mathbf{U}(z)\mathbf{F}(z)$ is spectrally majorised. For this, the filters in $\mathbf{U}(z)$ would ideally implement a binary mask. Then $\mathbf{D}(z) = \mathbf{U}(z)\mathbf{F}(z)\tilde{\mathbf{F}}(z)\tilde{\mathbf{U}}(z)$, and $\mathbf{U}(z)$ can be absorbed into $\mathbf{A}(z)$ to yield the polynomial modal matrix $\tilde{\mathbf{Q}}(z) = \mathbf{A}(z)\tilde{\mathbf{U}}(z)$. Since an ideal $\mathbf{U}(z)$ providing a binary spectral mask will require infinite support, the order of the factors $\mathbf{Q}(z)$ and $\mathbf{D}(z)$ is likely to be much higher than in the case where $\mathbf{F}(z)\tilde{\mathbf{F}}(z)$ is already spectrally majorised by virtue of the source model.

Example. Let $J = M = 2$ with a diagonal $\mathbf{F}(z) = \text{diag}\{1 + z^{-1}; 1 - z^{-1}\}$ generating the unmajorised PSDs in Fig. 5.2. If $U_{hi}(z)$, $h, i \in \{1, 2\}$, are the elements of a matrix $\mathbf{U}(z)$ to enforce spectral majorisation, then $U_{11}(z)$ and $U_{22}(z)$ must be halfband lowpass and $U_{12}(z)$ and $U_{21}(z)$ halfband highpass filters. If selected as quadrature mirror filters with $U_{21}(z) = -\tilde{U}_{12}(z)$ and $U_{22}(z) = \tilde{U}_{11}(z)$, the condition of paraunitarity

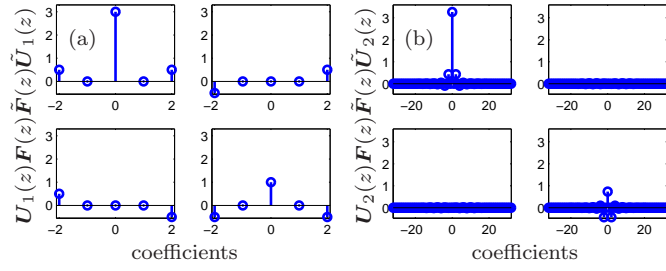


Figure 5.3: Approximately diagonalised matrices for paraunitary systems based on (a) Haar [64] and (b) 32C filters [65].

reduces to demand power complementarity [5],

$$\mathbf{U}_{11}(z)\tilde{\mathbf{U}}_{11}(z) + \mathbf{U}_{22}(z)\tilde{\mathbf{U}}_{22}(z) = 1. \quad (5.4)$$

For $\mathbf{U}(z)\mathbf{F}(z)\tilde{\mathbf{F}}(z)\tilde{\mathbf{U}}(z)$ to retain a diagonal structure, it can be shown that

$$\mathbf{U}_{11}(z)\mathbf{U}_{12}(z)\mathbf{F}_1(z)\tilde{\mathbf{F}}_1(z) = \mathbf{U}_{11}(z)\mathbf{U}_{12}(z)\mathbf{F}_2(z)\tilde{\mathbf{F}}_2(z) \quad (5.5)$$

is also required. This can be achieved only if $\mathbf{U}_{11}(z)\mathbf{U}_{12}(z) = 0$, i.e. they are ideal, complementary, infinite length halfband lowpass and highpass filters.

Using a Haar filter [64] of order 1 to construct $\mathbf{U}_1(z)$, the PSDs along the diagonal are now spectrally majorised as evident from Fig. 5.2. However, inspecting $\mathbf{U}_1(z)\mathbf{F}(z)\tilde{\mathbf{F}}(z)\tilde{\mathbf{U}}_1(z)$ in Fig. 5.3 (a), (5.5) is violated resulting in off-diagonal terms. Using the filter 32C from [65] to construct an approximately paraunitary $\mathbf{U}_2(z)$, the higher order of 31 now results in an approximately diagonalised $\mathbf{U}_2(z)\mathbf{F}(z)\tilde{\mathbf{F}}(z)\tilde{\mathbf{U}}_2(z)$ as demonstrated in Fig. 5.3 (b), which is also spectrally majorised according to Fig. 5.2. Therefore if sources contributing to $\mathbf{R}(z)$ are not spectrally majorised, a PEVD of $\mathbf{R}(z)$ in the sense of the definition in (2.9)–(2.11) requires higher order polynomial matrix factors than for a case where sources are spectrally majorised. Also as highlighted in Fig. 5.2 the sharp corners required by the spectrally majorised version can only ever be approximated with finite length filters.

5.1.3 Eigenvalue Spread

Since PEVD algorithms have a stopping criterion that is tied to a threshold for off-diagonal values, the resolution of sources depends on the dynamic range of the source signals. This dynamic range can be defined as the ratio between the maximum and minimum value across all source PSDs and frequencies,

$$\gamma = \frac{\max_{\Omega,j} |F_j(e^{j\Omega})|^2}{\min_{\Omega,j} |F_j(e^{j\Omega})|^2}. \quad (5.6)$$

For $M = J$, even in the case where sources are not spectrally majorised, (5.6) represents a polynomial matrix condition number,

$$\gamma = \frac{\max_{\Omega} D_1(e^{j\Omega})}{\min_{\Omega} D_M(e^{j\Omega})}, \quad (5.7)$$

as after re-assigning frequency bands between channels, the minimum and maximum PSD values remain unaltered as demonstrated in Sec. 5.1.2.

5.1.4 Results

The following subsections present the details of the simulation scenario followed by the performance metrics used to compare the different source models and PEVD algorithms. The final three subsections present and analyse the results of the simulations.

Performance Metrics

To assess the impact of source model conditioning on PEVD performance the following metrics are used. First the convergence of the PEVD algorithms is monitored via the normalised off-diagonal energy at the i -th iteration, $E_{\text{norm}}^{(i)}$ from (3.21). In addition to $E_{\text{norm}}^{(i)}$ for every iteration, the order of the truncated paraunitary matrices is recorded to show how the source model affects the growth of the paraunitary matrix, which directly represents the implementation cost of this lossless filter bank. To compare the diagonal matrices produced by the PEVD to the ground truth of the source model the power spectral densities (PSDs) are used. Ideally the PSDs extracted by PEVD algorithms

should match those of the source model, bar any frequency-reassignments in the case of spectrally unmajorised sources.

Simulation Scenarios

The first two sets of simulations present the results from 500 iterations of the SBR2, SMD, and MSME-SMD PEVD algorithms outlined in previous chapters for the spectrally majorised and unmajorised examples over an ensemble of 10^2 random instantiations. With $J = 4$ sources acquired by $M = 4$ sensors, for each instantiation the source model produces a distinct parahermitian matrix, $\mathbf{R}(z) \in \mathbb{C}^{4 \times 4}$. With $P = K = 30$, the order of $\mathbf{R}(z)$ is 119. For each ensemble, restrictions on the radii of zeros in the innovation filters create an average dynamic range of either 10 or 20 dB.

The final set of results demonstrate example PSDs, produced after 100 SMD iterations, compared to the original spectrally majorised and unmajorised source models. The final simulations use a single source model for each combination of majorisation and dynamic range rather than being averaged over an ensemble.

Algorithm Convergence

Figs. 5.4 (a) and (b) show how the different algorithms converge for the two source models identified in Sec. 5.1.1 for a dynamic range of 10 dB and 20 dB respectively. In both Figs. 5.4 (a) and (b) all algorithms initially converge faster for the unmajorised source but as the number of iterations increases, these curves slow down and are overtaken by the strictly majorised sources. The reason for the unmajorised source initially converging faster is that the parahermitian matrices generated from the unmajorised source tend to have larger valued off-diagonal elements leading to more energy transfer. After 500 iterations there is a noticeable difference between the two source models, with the strictly majorised converging better; this is apparent for both dynamic ranges and all three PEVD algorithms. The better convergence is likely due to the majorised source PSDs being simpler and easier to model with the PEVD. With the higher dynamic range in Fig. 5.4 (b) the curves all appear worse than their counterparts in Fig. 5.4 (a) and end up closer together. Typically the PEVD algorithms have difficulty resolving

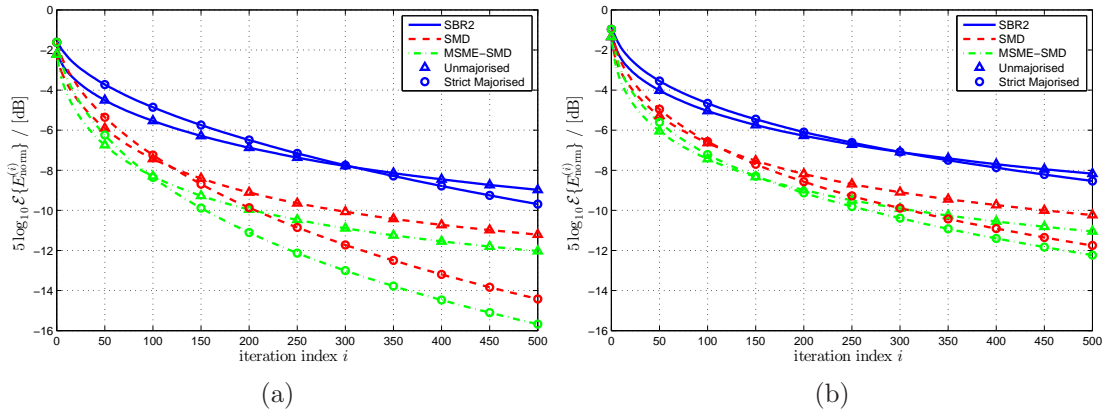


Figure 5.4: Reduction in off-diagonal energy for both majorisation types with a dynamic range of (a) 10 dB and (b) 20 dB for a selection of PEVD algorithms.

the lowest eigenvalues, with a higher dynamic range the lowest eigenvalues are smaller and this may indicate why the algorithms don't converge as well on larger dynamic ranges.

Paraunitary Order

The growth in paraunitary order for the PEVD methods using the unmajorised and strictly majorised sources at 10 dB is shown in Fig. 5.5 (a) with the larger dynamic range of 20 dB depicted in Fig. 5.5 (b). In both Figs. 5.5 (a) and 5.5 (b) the SMD and SBR2 algorithms perform similarly but the multiple shifts of the MSME-SMD algorithm cause the paraunitary order to grow faster. The paraunitary order for the MSME-SMD algorithm is also affected more when the dynamic range of the source increases. For all the algorithms over both dynamic ranges we see that the paraunitary orders for the unmajorised sources tends to be higher than for the strictly majorised sources. The main exception to this is the MSME-SMD with the strictly majorised (20 dB) source where for the most part it performs worse than its unmajorised equivalent. These results support the example given in Sec. 5.1.2 whereby the frequency reassignment in the unmajorised case results in a higher paraunitary order.

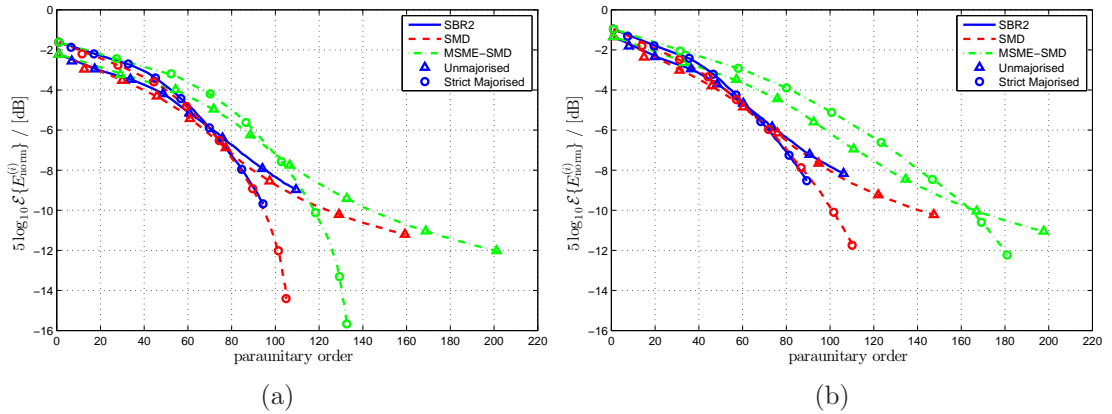


Figure 5.5: Paraunitary matrix order for both majorisation types with a dynamic range of (a) 10 dB and (b) 20 dB for a selection of PEVD algorithms.

Power Spectral Densities

This section investigates four example source models which have had the SMD algorithm applied for only 100 iterations each. PSDs of the source models are shown in Figs. 5.6 (a), & (b), 5.7 (a) & (b), first showing a 10 dB dynamic range for the strictly majorised source then the unmajorised equivalent followed by the same sources with a 20 dB dynamic range. Like the simple example in Fig. 5.3, the unmajorised sources in Figs. 5.6 (b) and 5.7 (b) are approximately majorised by channel permutations. Comparing Figs. 5.6 (a) and (b) both appear to quite accurately model their respective sources with only small deviations around the angular frequency 1.2π . For the 20 dB examples in Figs. 5.7 (a) and (b) there are some quite large deviations from the source model particularly in the strictly majorised example.

The performance metrics studied in the previous subsections are shown in Tab. 5.1 for the source decompositions in Figs. 5.6 & 5.7. It is interesting to note that for the 20 dB majorised source the SMD PEVD has a better diagonalisation measure yet the source representation appears worse. The performance metrics in Tab. 5.1 are recorded after 100 iterations which corresponds to the points in Figs. 5.4 & 5.5 where the curves for the majorised and unmajorised sources are close to one another. The fact that for 10 dB the unmajorised case has better diagonalisation and paraunitary order and for 20 dB has worse diagonalisation and paraunitary order is not surprising. Running the

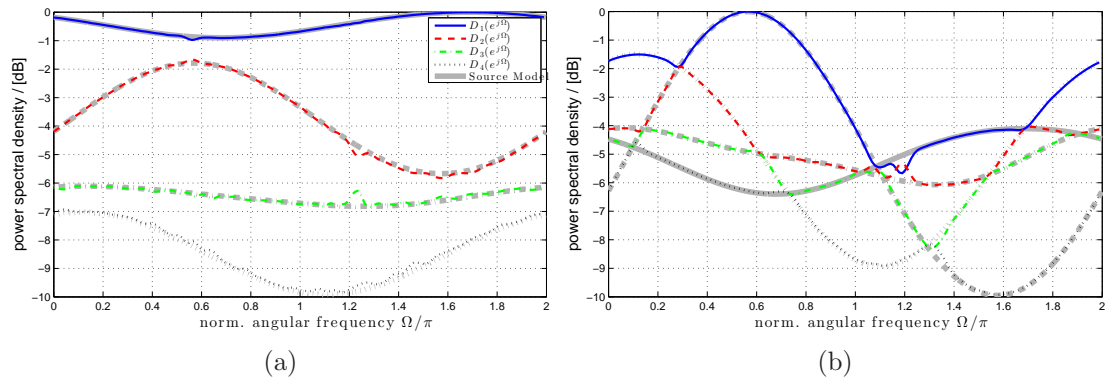


Figure 5.6: PSD shown for (a) majorised and (b) unmajorised source models with dynamic range of 10 dB overlaid with SMD decomposition after 100 iterations.

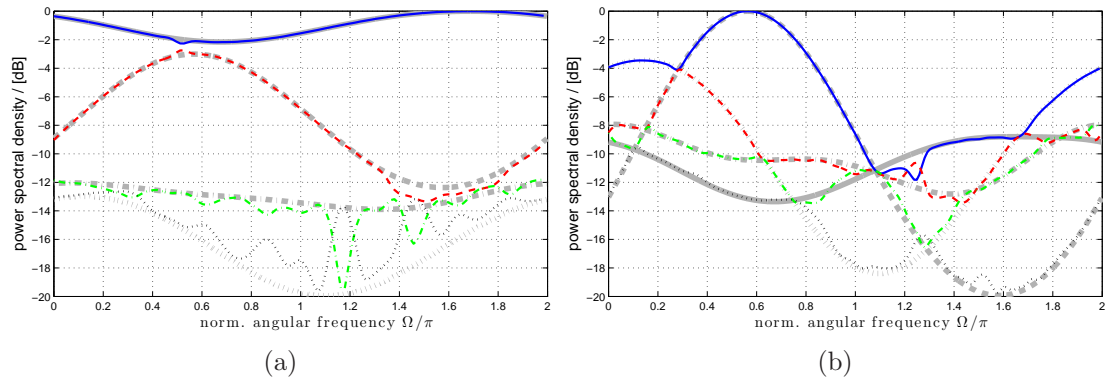


Figure 5.7: PSD shown for (a) majorised and (b) unmajorised source models with dynamic range of 20 dB overlaid with SMD decomposition after 100 iterations.

simulations over 500 iterations yields the results in brackets in Tab. 5.1 which match the final trends shown in Figs. 5.4 & 5.5. After 500 iterations of the SMD algorithm the PSDs shown in Figs. 5.6 and 5.7 appear to exactly match the source models apart from the frequency reassignment in the unmajorised case.

5.2 Angle of Arrival Estimation using Polynomial MUSIC

This section will investigate the performance improvement achieved when the MSME-SMD algorithm is used in an example compared with SBR2. The example here is broadband angle of arrival estimation using the polynomial version of the well known

Table 5.1: Performance metrics for different source models after 100 (and 500) SMD iterations

Source Model	Diag. Meas. (dB)	PU Order
Strict 10 dB	-13.11 (-29.90)	88 (123)
Unmajorised 10 dB	-14.69 (-22.35)	80 (151)
Strict 20 dB	-13.31 (-25.40)	66 (100)
Unmajorised 20 dB	-12.81 (-20.18)	84 (138)

Multiple Signal Classification (MUSIC) [1] algorithm, Polynomial-MUSIC or P-MUSIC [12].

5.2.1 MUSIC Algorithm

The multiple signal classification (MUSIC) algorithm has been around since the 1980s [1] and it relies on the (scalar) eigenvalue decomposition of a covariance matrix $\mathbf{R} = \mathcal{E}\{\mathbf{x}[n]\mathbf{x}^H[n]\}$ in order to determine the angle of arrival of sources. The M -element array collects data at time n in the vector $\mathbf{x}[n]$ which consists of contributions from J far-field sources, $s_j[n]$, plus noise, $\mathbf{v}[n]$

$$\mathbf{x}[n] = \sum_{j=1}^J \mathbf{s}_j + \mathbf{v}[n] \quad . \quad (5.8)$$

The projection of the j -th source, $s_j[n]$, onto the array is given by the vector $\mathbf{s}_j[n]$, taking the first sensor signal as reference the relative delays to the M sensors can be characterised as

$$\mathbf{s}_j[n] = \begin{bmatrix} s_j[n] \\ s_j[n - \Delta\tau_{j,1}] \\ \vdots \\ s_j[n - \Delta\tau_{j,M-1}] \end{bmatrix} \quad . \quad (5.9)$$

The time delay term $\Delta\tau_{j,m}$ is simply the time difference of arrival between the first sensor and m -th sensor i.e. $\Delta\tau_{j,m} = \tau_{j,m} - \tau_{j,0}$. In the narrowband case these time

delays can be synthesised with phase shifts. If the narrowband source has a normalised angular frequency Ω and the reference signal $s_j[n] = e^{j\Omega n}$ then (5.9) becomes

$$\mathbf{s}_j[n] = \begin{bmatrix} 1 \\ e^{-j\Omega\Delta\tau_{j,1}} \\ \vdots \\ e^{-j\Omega\Delta\tau_{j,M-1}} \end{bmatrix} e^{j\Omega n} = \mathbf{a}_{\Omega,\vartheta_j} e^{j\Omega n} \quad , \quad (5.10)$$

where $\mathbf{a}_{\Omega,\vartheta_j}$ is referred to as the narrowband steering vector for the angle of arrival ϑ_j . Using the narrowband steering vector notation (5.8) can be rewritten as

$$\mathbf{x}[n] = \sum_{j=1}^J \mathbf{a}_{\Omega,\vartheta_j} s_j[n] + \mathbf{v}[n] \quad . \quad (5.11)$$

Taking the EVD of $\mathbf{R} = \mathcal{E}\{\mathbf{x}[n]\mathbf{x}^H[n]\}$ and applying a threshold to the eigenvalues such that those close to the noise floor are contained in \mathbf{D}_n , and eigenvalues in \mathbf{D}_s are associated with the strongest sources, gives

$$\mathbf{R} = \mathbf{Q}\mathbf{D}\mathbf{Q}^H \quad (5.12)$$

$$= [\mathbf{Q}_s \ \mathbf{Q}_n] \begin{bmatrix} \mathbf{D}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_n \end{bmatrix} \begin{bmatrix} \mathbf{Q}_s^H \\ \mathbf{Q}_n^H \end{bmatrix} \quad . \quad (5.13)$$

Where the signal-plus-noise subspace is contained in \mathbf{Q}_s , and the noise-only subspace is \mathbf{Q}_n . The MUSIC algorithm then scans the noise-only subspace with a set of steering vectors, $\mathbf{a}_{\Omega,\vartheta}$. If the vector $\mathbf{Q}_n^H \mathbf{a}_{\Omega,\vartheta}$ is close to zero then it indicates that $\mathbf{a}_{\Omega,\vartheta}$ is a steering vector for the orthogonal signal-plus-noise subspace, \mathbf{Q}_s . The MUSIC spectrum, $P_{\text{MU}}(\vartheta)$, is generated by evaluating the reciprocal of the steering vectors applied to the noise subspace

$$P_{\text{MU}}(\vartheta) = \frac{1}{\mathbf{a}_{\Omega,\vartheta}^H \mathbf{Q}_n \mathbf{Q}_n^H \mathbf{a}_{\Omega,\vartheta}} \quad . \quad (5.14)$$

5.2.2 Polynomial MUSIC Algorithm

The broadband polynomial MUSIC algorithm follows a very similar procedure to the narrowband method in the previous section. The space-time covariance matrix is constructed as

$$\mathbf{R}[\tau] = \mathcal{E}\{\mathbf{x}[n]\mathbf{x}^H[n - \tau]\} \quad , \quad (5.15)$$

where $\mathbf{R}(z) \bullet \text{---} \circ \mathbf{R}[\tau]$. Similar to the narrowband MUSIC algorithm we can calculate the PEVD of $\mathbf{R}(z)$ and threshold the polynomial eigenvalues to those near the noise floor, $\mathbf{D}_n(z)$, and the signal-plus-noise eigenvalues, $\mathbf{D}_s(z)$

$$\mathbf{R}(z) = [\tilde{\mathbf{Q}}_s(z) \quad \tilde{\mathbf{Q}}_n(z)] \begin{bmatrix} \mathbf{D}_s(z) & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_n(z) \end{bmatrix} \begin{bmatrix} \mathbf{Q}_s(z) \\ \mathbf{Q}_n(z) \end{bmatrix} \quad . \quad (5.16)$$

Whereas steering vectors in the narrowband MUSIC algorithm can be implemented using phase shifts, in the broadband scenario fractional delays must be used. Here the fractional delay is implemented by an appropriately sampled sinc function

$$a_{j,m}[n] = \text{sinc}(nT_s - \Delta\tau_{j,m}) \quad . \quad (5.17)$$

Taking the z -transform of the fractional delay, $a_{j,m}[n] \circ \text{---} \bullet A_{j,m}(z)$, the broadband steering vector can be constructed as

$$\mathbf{a}_\vartheta(z) = \begin{bmatrix} A_{j,0}(z) \\ \vdots \\ A_{j,M-1}(z) \end{bmatrix} \quad , \quad (5.18)$$

where \mathbf{a}_ϑ is dependent on the angle of arrival, ϑ , due to the delay parameter $\Delta\tau_{j,m}$ from (5.17).

As with the narrowband MUSIC algorithm the noise subspace, $\mathbf{Q}_n(z)$, is tested with various broadband steering vectors

$$\gamma_\vartheta(z) = \tilde{\mathbf{a}}_\vartheta(z)\mathbf{Q}_n(z)\tilde{\mathbf{Q}}_n(z)\mathbf{a}_\vartheta(z) \quad . \quad (5.19)$$

Now $\gamma_\vartheta(z)$ is a power spectral density rather than the norm that is the result of narrowband MUSIC; the power spectral density leads to two versions of the P-MUSIC algorithm. The first version, spatial polynomial (SP) MUSIC takes the zero lag term $\gamma_\vartheta[0]$ of the autocorrelation like sequence, $\gamma_\vartheta[\tau] \circ \bullet \gamma_\vartheta(z)$, which is related to the energy in $\tilde{\mathbf{Q}}_n(z)\mathbf{a}_\vartheta(z)$. The energy in $\gamma_\vartheta[0]$ is only dependent on the angle of arrival and will be at a minimum in the direction of source signals so the reciprocal is used

$$P_{SP-MU}(\vartheta) = \frac{1}{\gamma_\vartheta[0]} \quad . \quad (5.20)$$

The second version of polynomial MUSIC is called spatio-spectral polynomial (SSP) MUSIC which can indicate both the angle of arrival and the frequency range over which the source is active

$$P_{SSP-MU}(\vartheta, \Omega) = \left(\sum_{\tau=-\infty}^{\infty} \gamma_\vartheta[\tau] e^{-j\Omega\tau} \right)^{-1} \quad . \quad (5.21)$$

5.2.3 Results

This section compares the impact of using two different PEVD methods – SBR2 and MSME-SMD – on the results of the spatio-spectral polynomial MUSIC algorithm. The PEVD methods achieve different levels of diagonalisation and so the subspaces identified will also differ in accuracy. The simulation scenario considered comprises of two broadband sources with partially overlapping spectra and different angles of arrival:

- source 1 — located at $\vartheta_1 = -20^\circ$, and active over a frequency range $\Omega_1 \in [15/32\pi, 30/32\pi]$,
- source 2 — located at $\vartheta_2 = 30^\circ$, and active over a frequency range $\Omega_2 \in [10/32\pi, 25/32\pi]$.

These sources are used to illuminate an $M = 8$ element sensor array, where the source signals are corrupted by uncorrelated independent and identically distributed complex Gaussian noise at 20 dB SNR. Each of the PEVD methods were run for 100 iterations on the same space time covariance matrix to obtain the respective noise-only subspaces.

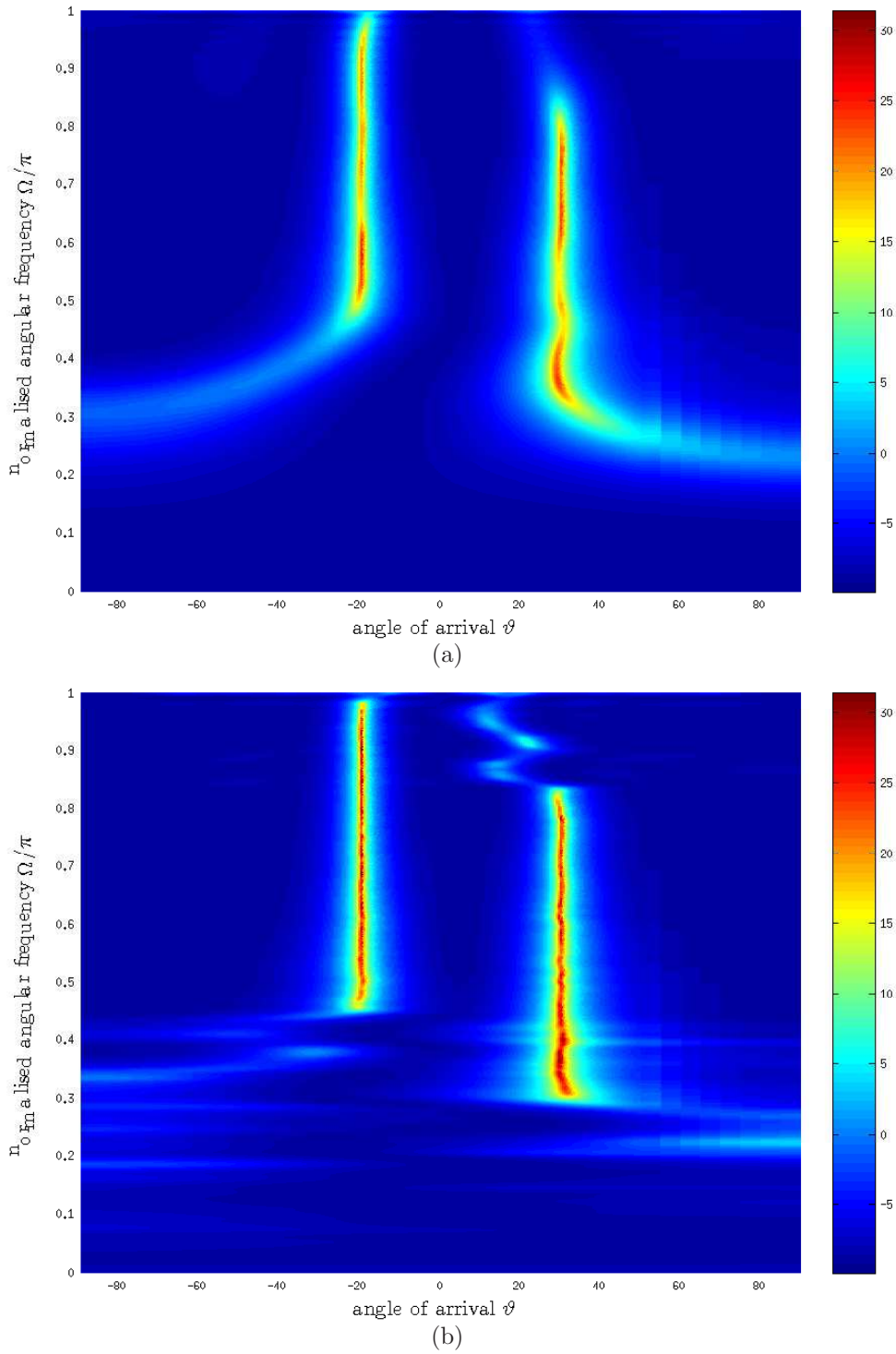


Figure 5.8: Performance of SSP-MUSIC based on (a) SBR2 and (b) MSME-SMD for PEVD for a scenario with two independent broadband sources located at $\vartheta_1 = -20^\circ$ and $\vartheta_2 = 30^\circ$ respectively.

The two noise-only subspaces are then used to create the spatio-spectral polynomial MUSIC spectrum from (5.21). The results for SSP-MUSIC are shown in Fig. 5.8 (a) and Fig. 5.8 (b) for SBR2 and MSME-SMD respectively. The accuracy of the decomposition is related to the width (in terms of AoA) and (colour) intensity of the sources indicated in Fig.5.8 (a) and (b). Numerically, for this example the MSME-SMD PEVD approach reduces the off-diagonal energy of $\mathbf{R}(z)$ by approximately an additional 2.5 dB compared with SBR2. The P-MUSIC algorithm and results presented here are also discussed in more detail in [12,13]. In addition P-MUSIC has been benchmarked against a number of other schemes: coherent signal subspace (CSS) methods, which apply a presteering in the direction of a suspected source, such that narrowband techniques can be applied [66–68], and another presteering approach in [69, 70], which presteers the data until the principal eigenvalue of the narrowband covariance matrix is maximised. The CSS method is powerful but requires some prior knowledge of the source to be detected, the latter is never able to detect more than a single source.

5.3 Polynomial Generalised Eigenvalue Decomposition

This section gives an example of the extension of a well established linear algebraic technique to polynomial matrices. The linear algebraic technique chosen is the generalised eigenvalue decomposition (GEVD) which is extended to the polynomial domain using the PEVD, forming the polynomial-matrix GEVD (PGEVD). In a multi-microphone scenario with M sensors, the generalised eigenvalue decomposition has been utilised to enable a low rank approximation of a multichannel Wiener filter (MWF) [71, 72]. This MWF is meant to separate a desired signal from $K - 1$ interfering signals, with $K < M$. The MWF $\hat{\mathbf{W}}$ used in [71] is calculated as

$$\hat{\mathbf{W}} = \mathbf{R}_{yy}^{-1} \mathbf{R}_{ss} \hat{\mathbf{E}}_d, \quad (5.22)$$

where \mathbf{R}_{yy} is the signal-plus-noise covariance matrix, \mathbf{R}_{ss} is the estimated signal-only covariance matrix, and $\hat{\mathbf{E}}_d$ is used to select the signal of interest.

In scenarios such as the audio processing examples in [71] and [72], the signal-plus-

noise covariance matrix is easily calculated from the received signals, but the signal-only covariance matrix cannot be obtained in this way. Instead, the signal-only covariance matrix must be estimated from the signal-plus-noise and noise-only covariance matrices; during this estimation process, the rank of the matrix is artificially increased. To avoid using matrices with an artificially high rank, the estimated covariance matrix is replaced by a low-rank approximation [71]. Methods using both the scalar EVD and GEVD have been explored in [72] with results indicating the GEVD based low-rank approximation performs better as it effectively selects the modes with the highest SNR.

In brief, the GEVD jointly diagonalises two separate covariance matrices using a common set of generalised eigenvectors. Since the audio application area is inherently broadband, while both the covariance matrix and the GEVD factorisation are defined for the narrowband case, a broadband extension of the GEVD is required. In [71], this is addressed by a frequency domain approach which solves independent narrowband problems in DFT bins. In order to exploit the spectral coherence of e.g. audio signals, the aim of this section is to find a broadband signal processing approach based on polynomial space-time covariance matrices. In this section, the PEVD is extended to a polynomial version of the GEVD (PGEVD) in order to directly address the broadband MWF problem. Wiener filters have previously been formulated using polynomial matrix techniques [73] but initially could not be solved due to an absence of tools. Here the polynomial Wiener filter solution in [19] is complimented by a PGEVD approach.

Below, Sec. 5.3.1 reviews the (narrowband) GEVD, in particular the Cholesky approach on which the extension will be based. Sec. 5.3.2 outlines the extension of the Cholesky approach to polynomial matrices. Inversion of polynomial matrices is covered in Sec. 5.3.3, with results presented in Sec. 5.3.4.

5.3.1 Generalised Eigenvalue Decomposition

The GEVD solves the problem of $\mathbf{R}_1\mathbf{v} = \psi\mathbf{R}_2\mathbf{v}$, $\mathbf{v} \neq \mathbf{0}$, where \mathbf{v} is an eigenvector of the pencil $\mathbf{R}_1 - \psi\mathbf{R}_2$ [7]. In matrix form we can write this as

$$\mathbf{R}_1\mathbf{V} = \mathbf{R}_2\mathbf{V}\Psi, \quad (5.23)$$

where the diagonal matrix Ψ contains the generalised eigenvalues and \mathbf{V} holds the corresponding generalised eigenvectors. Applying the generalised eigenvectors to \mathbf{R}_1 and \mathbf{R}_2 results in

$$\mathbf{V}^H \mathbf{R}_1 \mathbf{V} = \Psi , \quad (5.24)$$

$$\mathbf{V}^H \mathbf{R}_2 \mathbf{V} = \mathbf{I} , \quad (5.25)$$

where \mathbf{I} is the identity. Rearranging (5.23), or (5.24) and (5.25) to

$$\mathbf{R}_2^{-1} \mathbf{R}_1 = \mathbf{V} \Psi \mathbf{V}^{-1} , \quad (5.26)$$

it is evident that $\mathbf{R}_2^{-1} \mathbf{R}_1$ is no longer Hermitian and therefore \mathbf{V} is not guaranteed to be unitary, even if — as in the case of MWF [71, 72] — \mathbf{R}_1 and \mathbf{R}_2 are Hermitian covariance matrices.

Among a number of options to calculate the GEVD or joint diagonalisation of Hermitian matrices \mathbf{R}_1 and \mathbf{R}_2 is a Cholesky-based approach [7]. In a first step, a Cholesky decomposition is used to obtain

$$\mathbf{R}_2 = \mathbf{L} \mathbf{L}^H , \quad (5.27)$$

with \mathbf{L} lower left triangular. Next an intermediate Hermitian matrix \mathbf{C} is formed using \mathbf{R}_1 and \mathbf{L}^{-1} ,

$$\mathbf{C} = \mathbf{L}^{-1} \mathbf{R}_1 \mathbf{L}^{-H} . \quad (5.28)$$

Finally, an EVD is performed on the intermediate Hermitian matrix,

$$\mathbf{C} = \mathbf{Y} \Psi \mathbf{Y}^H , \quad (5.29)$$

such that Ψ contains the generalised eigenvalues and the generalised eigenvectors are the columns of $\mathbf{V} = \mathbf{L}^{-H} \mathbf{Y}$.

In both [71] and [72] the GEVD is used to obtain a low-rank approximation of the signal-only covariance matrix, \mathbf{R}_{ss} . Using the GEVD for low-rank approximation, \mathbf{R}_1

and \mathbf{R}_2 would be the signal-plus-noise and noise-only covariance matrices respectively. \mathbf{R}_1 is estimated from the array input data during a signal-plus-noise period, and \mathbf{R}_2 can either be estimated from a noise-only data segment or come from a-priori knowledge. Various methods for low-rank approximation of \mathbf{R}_{ss} are based around $\mathbf{R}_{ss} = \mathbf{R}_1 - \mathbf{R}_2$, and the GEVD based approach has been shown to give the most reliable estimation of \mathbf{R}_{ss} [72]. Rearranging (5.24) and (5.25) gives

$$\mathbf{R}_1 = \mathbf{V}^{-H} \boldsymbol{\Psi} \mathbf{V}^{-1} \quad , \quad (5.30)$$

$$\mathbf{R}_2 = \mathbf{V}^{-H} \mathbf{V}^{-1} \quad , \quad (5.31)$$

which can be used in the estimation of \mathbf{R}_{ss} to get

$$\mathbf{R}_{ss} = \mathbf{V}^{-H} \boldsymbol{\Psi} \mathbf{V}^{-1} - \mathbf{V}^{-H} \mathbf{V}^{-1} \quad (5.32)$$

$$= \mathbf{V}^{-H} (\boldsymbol{\Psi} - \mathbf{I}) \mathbf{V}^{-1} \quad . \quad (5.33)$$

If the generalised eigenvalues are ordered, a low-rank estimation can be achieved by setting the lower diagonal elements of $(\boldsymbol{\Psi} - \mathbf{I})$ in (5.33) to zero.

5.3.2 GEVD Extended to Polynomial Matrices

Extending the GEVD in (5.23) to polynomial, parahermitian, matrices $\mathbf{R}_i(z)$, $i = \{1, 2\}$, leads to a joint diagonalisation problem akin to (5.24) and (5.25)

$$\tilde{\mathbf{V}}(z) \mathbf{R}_1(z) \mathbf{V}(z) = \boldsymbol{\Psi}(z) \quad , \quad (5.34)$$

$$\tilde{\mathbf{V}}(z) \mathbf{R}_2(z) \mathbf{V}(z) = \mathbf{I} \quad . \quad (5.35)$$

Here $\boldsymbol{\Psi}(z)$ contains the polynomial generalised eigenvalues. The factorisations in (5.34) and (5.35) can be shown to exist if the PEVDs of $\mathbf{R}_i(z)$, $i = \{1, 2\}$, exist [50], and if $\mathbf{R}_2(z)$ is invertible. Due to the ambiguities of the PEVD from Chapter 2, the generalised polynomial eigenvectors in $\mathbf{V}(z)$ can at the very least be arbitrarily delayed w.r.t. each other, leading to a variability in their order.

For the computation of $\mathbf{V}(z)$ and $\boldsymbol{\Psi}(z)$, a two-step Cholesky-style approach can be

performed. The first step starts by calculating the PEVD

$$\mathbf{R}_2(z) = \tilde{\mathbf{Q}}_2(z)\mathbf{D}_2(z)\mathbf{Q}_2(z) \quad , \quad (5.36)$$

followed by the spectral factorisation outlined in Sec. 5.3.3. Now the factor $\mathbf{L}(z) = \tilde{\mathbf{Q}}_2(z)\mathbf{D}_2^{(+)}(z)$ is not lower left triangular but is easily inverted, such that $\mathbf{L}^{-1}(z) = (\mathbf{D}_2^{(+)}(z))^{-1}\mathbf{Q}_2(z)$, using the procedure described in Sec. 5.3.3.

The second step is initiated by constructing the intermediate (parahermitian) matrix

$$\mathbf{C}(z) = \mathbf{L}^{-1}(z)\mathbf{R}_1(z)\tilde{\mathbf{L}}^{-1}(z) \quad . \quad (5.37)$$

Next, using the PEVD, the intermediate matrix is decomposed into

$$\mathbf{C}(z) = \tilde{\mathbf{Q}}(z)\mathbf{\Psi}(z)\mathbf{Q}(z) \quad , \quad (5.38)$$

where $\mathbf{\Psi}(z)$ contains the polynomial generalised eigenvalues and the polynomial generalised eigenvectors are calculated as $\mathbf{V}(z) = \tilde{\mathbf{L}}^{-1}(z)\mathbf{Q}(z)$, noting a possible order reduction due to the ambiguity from Sec. 4.1.

5.3.3 Polynomial Matrix Inverse

In this approach spectral factorisation [57, 74] of the polynomial eigenvalues takes the place of the Cholesky decomposition. The spectral factorisation is carried out on each of the individual polynomial eigenvalues as in [57, 74] using a method such as *sfact()* from [75]. After spectral factorisation the PEVD equation from (2.9) becomes

$$\mathbf{R}_2(z) \approx \tilde{\mathbf{Q}}_2(z)\mathbf{D}_2^{(+)}(z)\mathbf{D}_2^{(-)}(z)\mathbf{Q}_2(z) \quad , \quad (5.39)$$

where $\mathbf{D}_2^{(+)}(z)$ and $\mathbf{D}_2^{(-)}(z) = \tilde{\mathbf{D}}_2^{(+)}(z)$ are the minimum and maximum phase components of $\mathbf{D}_2(z)$ respectively.

Like the scalar method outlined in Sec. 5.3.1, $\mathbf{L}(z)$ can be constructed as $\mathbf{L}(z) = \tilde{\mathbf{Q}}_2(z)\mathbf{D}_2^{(+)}(z)$ and so $\mathbf{R}_2(z) = \mathbf{L}(z)\tilde{\mathbf{L}}(z)$. It is possible to calculate the inverse of $\mathbf{L}(z)$ by inverting both $\tilde{\mathbf{Q}}_2(z)$ and $\mathbf{D}_2^{(+)}(z)$ independently. The inverse of a paraunitary

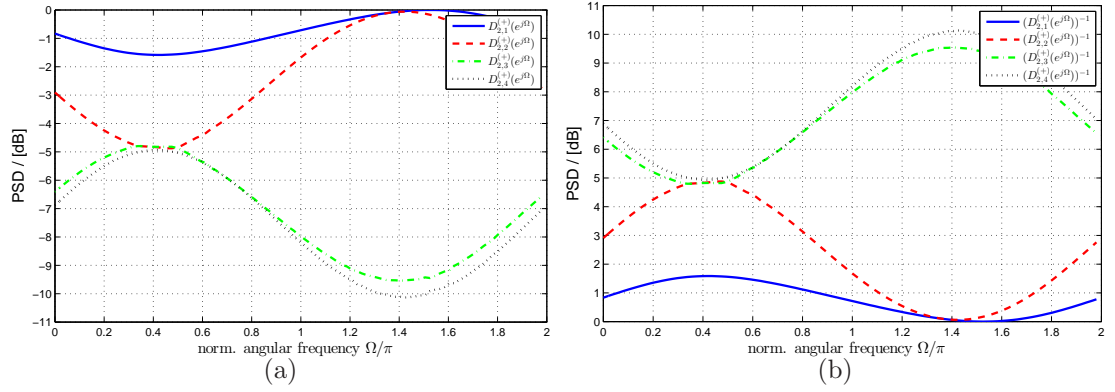


Figure 5.9: Power spectral density (PSD) of the (a) minimum phase components, $\mathbf{D}_2^{(+)}(z)$ and (b) inverted minimum phase components $(\mathbf{D}_2^{(+)}(z))^{-1}$.

matrix is simply its parahermitian transpose, i.e. $\mathbf{Q}_2(z)\tilde{\mathbf{Q}}_2(z) = \mathbf{I}$. The inverse of $\mathbf{D}_2^{(+)}(z) \bullet\circ \mathbf{D}_2^{(+)}[\tau]$ is calculated in the DFT domain by taking the FFT of each of the diagonal entries, $\mathbf{D}_{2,i}^{(+)}[\tau] \forall i = 1 \dots M$, where $\mathbf{D}_{2,i}^{(+)}[\tau]$ is the i th diagonal entry.

The inversion of $\mathbf{D}_2^{(+)}(z)$ can be illustrated by observing the power spectral densities of $\mathbf{D}_2^{(+)}(z)$ in Fig. 5.9(a), and its inverse $(\mathbf{D}_2^{(+)}(z))^{-1}$ in Fig. 5.9(b). Typically the PEVD methods from Sec. 2.3 will resolve the eigenvalue with the highest power $\mathbf{D}_{2,1}^{(+)}(z)$ better than the eigenvalue with the lowest power $\mathbf{D}_{2,4}^{(+)}(z)$; this is important for the inversion as small errors in $\mathbf{D}_{2,4}^{(+)}(z)$ can get significantly amplified in $(\mathbf{D}_{2,4}^{(+)}(z))^{-1}$.

5.3.4 Results

Performance Metrics

Diagonalisation. Since the goal of the GEVD is to minimise off-diagonal energy of both $\mathbf{R}_1(z)$ and $\mathbf{R}_2(z)$, a suitable normalised metric similar to (3.21) is

$$E_{n,\text{norm}}^{(i)} = \frac{\sum_{\tau} \sum_{k=1}^M \|\hat{\mathbf{s}}_{n,k}^{(i)}[\tau]\|_2^2}{\sum_{\tau} \|\mathbf{R}_n[\tau]\|_F^2} \quad (5.40)$$

where $\hat{\mathbf{s}}_{n,k}^{(i)}[\tau]$ is the modified k -th column vector containing all but the on-diagonal elements of $\mathbf{S}_n^{(i)}[\tau] \circ\bullet \mathbf{S}_n^{(i)}(z)$ for $n = 1, 2$. The partially diagonalised $\mathbf{S}_n^{(i)}(z)$ is calcu-

lated as

$$\mathbf{S}_1^{(i)}(z) = \tilde{\mathbf{V}}^{(i)}(z)\mathbf{R}_1(z)\mathbf{V}^{(i)}(z) \quad , \quad (5.41)$$

$$\mathbf{S}_2^{(i)}(z) = \tilde{\mathbf{V}}^{(i)}(z)\mathbf{R}_2(z)\mathbf{V}^{(i)}(z) \quad , \quad (5.42)$$

where $\mathbf{V}^{(i)}(z)$ is the generalised eigenvectors calculated after i iterations in each of the PEVD steps of Sec. 5.3.2.

Identity Error. Ideally the generalised eigenvectors in $\mathbf{V}(z)$ should turn $\mathbf{R}_2(z)$ into an $M \times M$ identity matrix; however, in reality $\mathbf{S}_2^{(i)}(z)$ only approximates the identity. Similar to (4.9) and (4.10), to quantify the closeness of $\mathbf{S}_2^{(i)}(z)$ to the identity the difference is defined as

$$\mathbf{E}^{(i)}(z) = \mathbf{I}_{M \times M} - \mathbf{S}_2^{(i)}(z) \quad . \quad (5.43)$$

With $\mathbf{E}^{(i)}[\tau] \circ \bullet \mathbf{E}^{(i)}(z)$ the identity error is given by the distance metric

$$\xi^{(i)} = \sum_{\tau} \|\mathbf{E}^{(i)}[\tau]\|_F^2 \quad . \quad (5.44)$$

Simulation Scenario

As an example, $\mathbf{R}_1(z) \in \mathbb{C}^{4 \times 4}$ and $\mathbf{R}_2(z) \in \mathbb{C}^{4 \times 4}$ are generated from the source model in [10]. The order of both $\mathbf{R}_1(z)$ and $\mathbf{R}_2(z)$ from the source model is 89 and the source model is set up such that $\mathbf{R}_1(z)$ is rank 1, with the contributing source having a dynamic range of approximately 10 dB. The matrix $\mathbf{R}_2(z)$ is full rank and its sources have a dynamic range of approximately 20 dB. The SMD algorithm in [10] is used to calculate the PEVD and is run for 500 iterations with the performance metrics recorded after every 10 iterations.

Joint Diagonalisation

The example PGEVD uses the two parahermitian matrices $\mathbf{R}_1(z)$ and $\mathbf{R}_2(z)$, shown in Fig. 5.10 (a) and (b) respectively. Following the algorithm outlined in Sec. 5.3.2, with each of the PEVD steps having 500 iterations, we obtain the jointly diagonalised systems in Fig. 5.11, closely approximating (5.34) and (5.35).

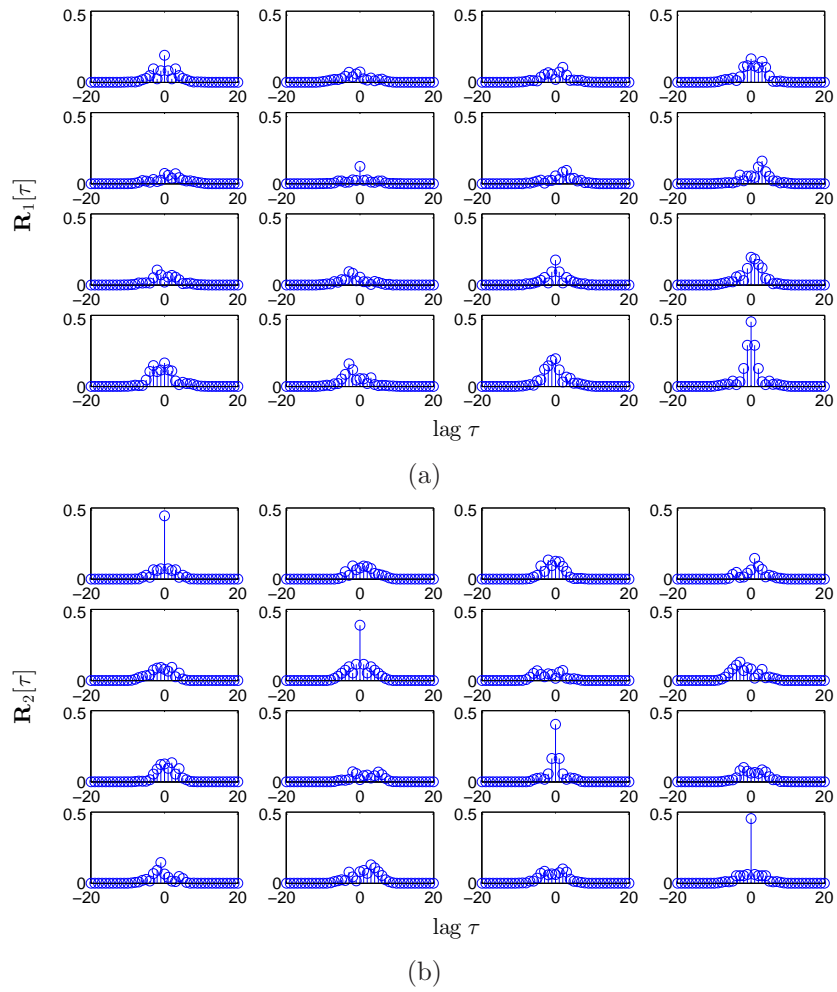


Figure 5.10: Space-time covariance matrix (a) $\mathbf{R}_1[\tau] \circ \bullet \mathbf{R}_1(z)$ and (b) $\mathbf{R}_2[\tau] \circ \bullet \mathbf{R}_2(z)$ showing only lags $|\tau| \leq 20$.

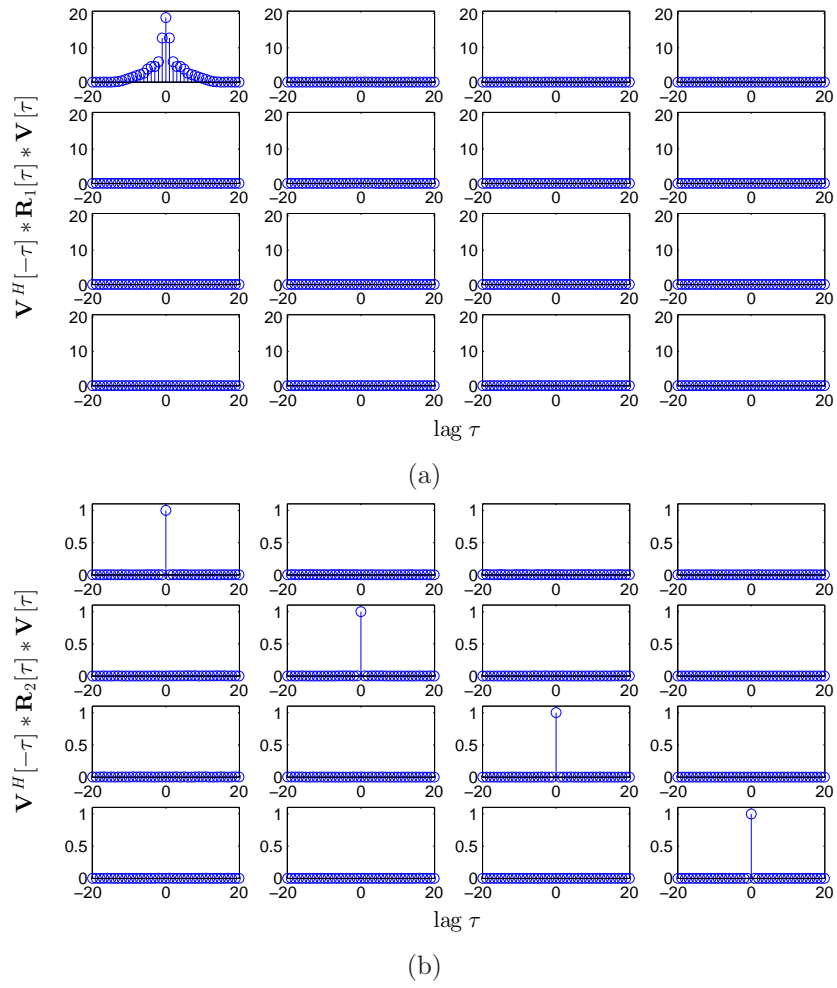


Figure 5.11: Results of joint diagonalisation for (a) $\mathbf{S}_1^{(500)}[\tau] \circ \bullet \mathbf{S}_1^{(500)}(z)$ and (b) $\mathbf{S}_2^{(500)}[\tau] \circ \bullet \mathbf{S}_2^{(500)}(z)$, for lags $|\tau| \leq 20$.

Algorithm Convergence

Although the PGEVD as such is not iterative, we are able to vary the number of PEVD iterations and show how performance metrics improve with more iterations. First we look at the reduction in off-diagonal energy for both $\tilde{\mathbf{V}}^{(i)}(z)\mathbf{R}_1(z)\mathbf{V}^{(i)}(z)$ and $\tilde{\mathbf{V}}^{(i)}(z)\mathbf{R}_2(z)\mathbf{V}^{(i)}(z)$ in Fig. 5.12. The convergence of $\mathbf{S}_1^{(i)}(z)$ is heavily influenced by the initial PEVD of $\mathbf{S}_2^{(i)}(z)$, and its effect can be beneficial or detrimental on the second PEVD stage as the number of iterations increases. If the number of iterations for the first PEVD were fixed, the curve for $\mathbf{R}_1(z)$ would be smooth; varying the number of iterations, as done here, causes the fluctuations in the curve for $\mathbf{R}_1(z)$.

In addition we can examine how close the decomposition $\tilde{\mathbf{V}}^{(i)}(z)\mathbf{R}_2(z)\mathbf{V}^{(i)}(z)$ is to the identity (delayed by an appropriate amount) at each iteration. Fig. 5.13 shows how $\tilde{\mathbf{V}}^{(i)}(z)\mathbf{R}_2(z)\mathbf{V}^{(i)}(z)$ converges towards to the identity as the number of PEVD iterations, i , increase.

5.4 Chapter Summary & Conclusions

This chapter has looked at three different types of PEVD application. First is the application of PEVD algorithms to parahermitian matrices with different properties. Next the MSME-SMD algorithm was applied to a broadband angle of arrival scenario in which the SBR2 algorithm had previously been used. Finally the generalised eigenvalue decomposition has been extended to a polynomial matrix GEVD by applying the PEVD in a Cholesky-like fashion.

The first part of this chapter investigated how the conditioning of the parahermitian matrix can affect the performance of a PEVD algorithm. Using the proposed source model, properties of the parahermitian matrix can be carefully controlled. A number of PEVD algorithms have been compared for different source model conditions. The results show that the speed of convergence is related to the source model used, in particular the dynamic range and the ordering of the eigenvalues. From the results presented in this chapter a higher dynamic range will typically cause the PEVD algorithms to converge more slowly in terms of reducing off-diagonal energy. The dynamic range has

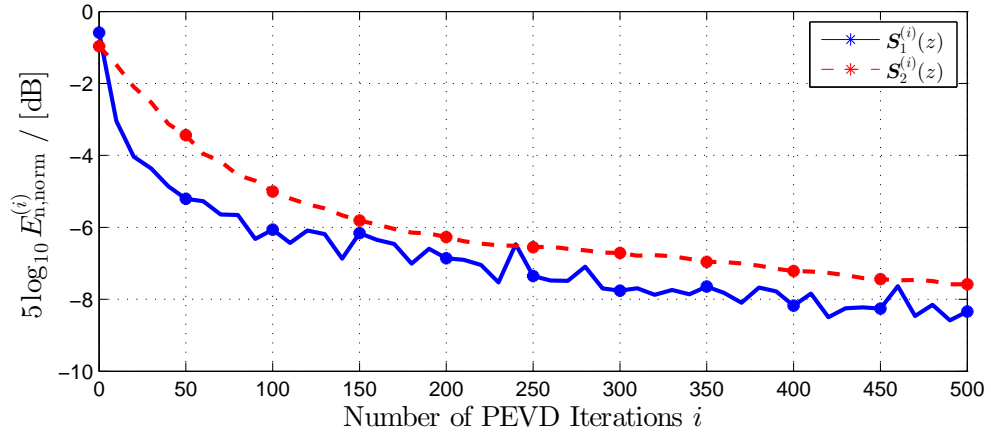


Figure 5.12: Reduction in off-diagonal energy for $\mathcal{S}_1^{(i)}(z)$ and $\mathcal{S}_2^{(i)}(z)$ as the number of PEVD iterations, i , increase.

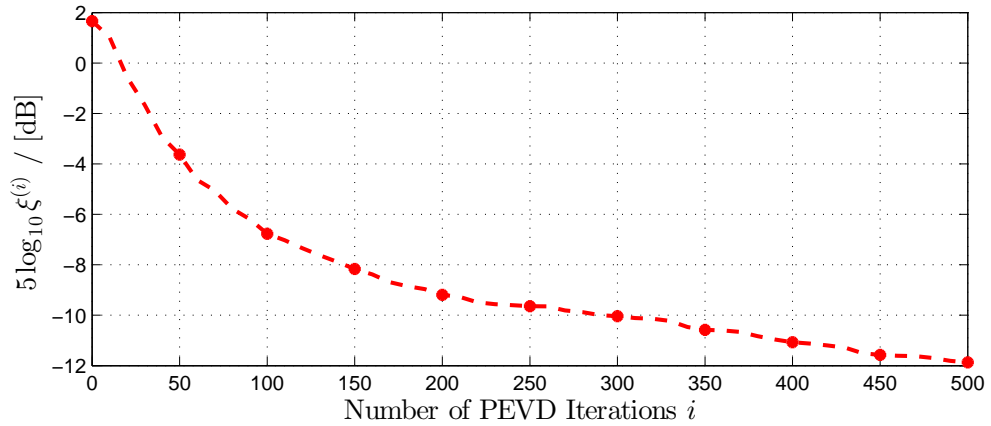


Figure 5.13: Difference between the identity matrix, \mathbf{I} , and $\mathcal{S}_2^{(i)}(z)$ as the number of PEVD iterations, i , increase.

minimal affect on the paraunitary orders for SBR2 and SMD algorithms, however in case of MSME-SMD the matrices tend to grow faster. When the ordering of the polynomial eigenvalues is changed, i.e. majorised vs. unmajorised, the majorised version will converge faster, to a better level of diagonalisation, with a lower order paraunitary matrix, independent of the PEVD algorithm used.

The second part of this chapter has explored the impact of iterative polynomial matrix eigenvalue decompositions – in particular the diagonalisation achieved by these algorithms – on subsequent processing relying on subspace information. Here the well established SBR2 algorithm is compared with the MSME-SMD implementation which excels in the suppression of off-diagonal energy. In simulations of a polynomial MUSIC algorithm for broadband angle of arrival estimation, the better diagonalisation leads to

a better identification of the relevant signal subspaces. Using MSME-SMD, P-MUSIC can extract a cleaner estimate with respect to both angle and frequency range of the estimated sources.

To extend the GEVD to broadband scenarios a polynomial version has been proposed. Since existing polynomial matrix factorisations are based on the paraunitary property, which the generalised polynomial eigenvectors do not fulfil, an indirect Cholesky-style approach has been suggested, which involves two PEVDs and the inversion of a parahermitian matrix. The effectiveness of this approach has been demonstrated using an example. Although not covered in this chapter, the numerical robustness of the polynomial matrix inversion can be enhanced by regularisation. Based on the proposed approach, it is therefore now possible to extend problems such as the multichannel Wiener filter, which elegantly rely on the GEVD, to the broadband case by utilising its polynomial extension.

Chapter 6

Conclusions & Future Work

6.1 Thesis Summary

Recently polynomial matrices have become more popular in DSP for their ability to model broadband systems more accurately than scalar matrices. Conventional scalar matrix methods have been extended to handle these polynomial matrices, in particular the EVD and PEVD have been highlighted as a useful tool in the realm of DSP. Based on two initial iterative PEVD algorithms this thesis has developed new powerful PEVD algorithms, reduced some of the inefficiencies of the algorithms, and applied these algorithms in different scenarios. The following three subsections summarise the main contributions of this thesis.

6.1.1 Multiple Shift Algorithms

In Chapter. 3 the idea of transferring more energy at each algorithm iteration was realised by shifting multiple rows and columns. An exhaustive approach was developed but the cost was such that it was not viable for an iterative PEVD algorithm. The multiple shift idea was tweaked to use a maximum element search and this was shown to perform very closely to the exhaustive search with only a fraction of the cost. A search method has also been developed to suit the SBR2 algorithm. Comparing the performance of the newly developed algorithms with their predecessors they transfer a significant amount of extra energy at each iteration, especially as the matrix dimension

is increased. Despite the more complex search step the SMD based algorithms actually converge faster in real time and the SBR2 method is only slightly slower. The main issue with the multiple shift algorithms was identified as the excessive polynomial matrix growth.

6.1.2 Efficient Implementations

To improve the efficiency of the resulting paraunitary filter banks a novel truncation method was developed that takes advantage of the ambiguity within paraunitary matrices. In general the new truncation method produces truncated paraunitary matrices which are both shorter and have a lower error metric than the existing approach. To reduce the order growth in the MSME-SMD algorithm a search space restriction was proposed that decouples the worst case polynomial matrix order growth from the matrix width. Experiments have shown that the restriction has a minimal impact on the algorithm convergence but improves both the order growth and with it the real time execution. The final refinement targets all SMD algorithms, the major cost of the SMD approach is the application of the non-sparse EVD modal matrix to all lags of the parahermitian matrix. The cyclic-by-row SMD approximation replaces the non-sparse matrix with several sparse Jacobi transformations. With only a single sweep of the off-diagonal elements the cyclic-by-row SMD approximation performs almost identically to the original SMD. The major difference comes when execution time is considered, with the cyclic-by-row approximation the SMD algorithms are significantly faster and even surpass SBR2 in real time convergence.

6.1.3 PEVD Applications

Previously many of the PEVD algorithms have been applied to a random assortment of parahermitian matrices with limited or no control on the properties of the parahermitian matrices. An important avenue of investigation was therefore to detail the impact of varying the properties of the input parahermitian matrix. In general the performance of the PEVD algorithms have been shown to vary quite dramatically with the properties of the input parahermitian matrix. To view the benefits of the new MSME-SMD

algorithm it was applied to a simple angle of arrival estimation example. The example uses the P-MUSIC framework which was originally developed using SBR2. Substituting in the MSME-SMD algorithm has been shown to give a cleaner estimate of the sources both in terms of the angle of arrival and frequency range. Chapter 5 concludes with a method that extends the generalised eigenvalue decomposition to polynomial matrices. The method is influenced by the scalar Cholesky based approach, however it does not include a polynomial Cholesky decomposition as such. The polynomial GEVD has been applied to a simple example and is able to produce generalised eigenvalues that diagonalise two parahermitian matrices in a way that mimics the scalar GEVD.

6.2 Future Work

This thesis has centred around polynomial matrix decompositions, in particular algorithms to compute the PEVD as well as a method to calculate a polynomial GEVD. Based on the contents of this thesis and the experience gained in the wider area possible items of interest for future work are listed below.

- Lower cost Maximum Energy SMD – Chapter 3 introduced the Maximum Energy SMD however the current implementation is too costly for any practical applications. The MSME-SMD algorithm provides a more practical method of achieving high energy transfer but it poses the question, is there an approach with similarly low cost that achieves better performance? The Maximum Energy SMD proves that it is possible to transfer more energy at each iteration but can this be done in a less costly fashion by a non-exhaustive method?
- Extension of multiple shifts to other iterative algorithms – the polynomial QR decomposition algorithm follows a very similar routine to the PEVD algorithms presented in this thesis. It would therefore make sense to extend the idea of multiple shift algorithms to these types of algorithm.
- Exploit the symmetry of the parahermitian matrix – although not specifically mentioned the search algorithms in Chapter 3 exploit the symmetry of the parahermitian matrix by only searching one half (plus the zero lag). It is almost

certainly possible that the entire PEVD algorithm could work on a similar basis with modifications to the shift steps. This would be particularly beneficial for the SMD algorithm as much of its cost comes from applying the EVD modal matrix to all lags.

- Further exploitation of the PEVD ambiguity – Chapter 4 utilised a basic allpass filter to reduce the paraunitary order in the PEVD. This allpass filter is just one basic approach, other methods such as infinite impulse response (IIR) may provide further benefits.
- Refining PEVD algorithms to reduce the polynomial matrix order growth – in Chapter 4 the reduced search space (RS-) MSME-SMD algorithm was introduced, as a method to control the polynomial order growth in the MSME-SMD algorithm. The RS-MSME-SMD method aims to set a balance between energy transfer and polynomial matrix growth, an interesting avenue may be to attempt something similar with the single shift algorithms where perhaps the shift selection is influenced by the distance from the zero lag. In addition a threshold could be set on the MS algorithms such that only worthwhile elements are shifted and the polynomial orders are not increased unnecessarily.
- Lower cost CbR-SMD – The cyclic-by-row SMD approximation could be made more efficient by using a threshold to determine whether it is worthwhile applying a Jacobi transformation to each off-diagonal element. If the element is smaller than the threshold it is simply skipped to reduce computational cost.
- Implementation of state-of-the-art PEVD algorithms and techniques in a high performance architecture – previously the SBR2 algorithm has been programmed onto an FPGA. It would be of interest to see how the more advanced algorithms compare and whether the approaches like the cyclic-by-row approximation and multiple shift algorithms are still beneficial on parallel architectures.
- Further investigate spectral majorisation – the first section in Chapter 5 investigated the impact of the source model conditioning. In particular the spectral

ordering of the source model was shown to affect the performance of the PEVD algorithms. When the PEVD algorithms have to deal with overlapping spectra the spectral majorisation they encourage leads to sharp corners in the spectra of the eigenvalues; typically sharp corners in the frequency domain lead to long filters in the time domain. It could be possible to avoid spectral majorisation and by implication reduce polynomial matrix lengths.

- Polynomial GEVD – in Chapter 5 the polynomial GEVD was introduced. The PGEVD opens several new avenues of research, for instance broadband Wiener filtering and distributed broadband beamforming to name a few.
- In addition to the work that directly follows on from the contributions presented in this thesis these contributions can be expected to benefit a number of applications that use polynomial matrix factorisations. Examples include MIMO communications such as [76–78] and beamforming [79] to name a few.

References

- [1] R. O. Schmidt. Multiple Emitter Location and Signal Parameter Estimation. *IEEE Transactions on Antennas and Propagation*, 34(3):276–280, March 1986.
- [2] P. Navarrete, J. Ruiz-Del-Solar. Analysis and Comparison of Eigenspace-Based Face Recognition Approaches. *International Journal of Pattern Recognition and Artificial Intelligence*, 16(7) 817–830, November 2002.
- [3] J. G. McWhirter, P. D. Baxter, T. Cooper, S. Redif, J. Foster. An EVD Algorithm for Para-Hermitian Polynomial Matrices. *IEEE Transactions on Signal Processing*, 55(5):2158–2169, May 2007.
- [4] R. Klemm. Space-Time Adaptive Processing Principles and Applications. *IEE Radar, Sonar, Navigation Avionics* IEE, London, U.K. 1998.
- [5] P. P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice Hall, 1993.
- [6] C. G. J. Jacobi. Über ein leichtes Verfahren die in der Theorie der Säcularstörungen vorkommenden Gleichungen numerisch aufzulösen. *Crelle's Journal* 30, pages 51-94. 1846.
- [7] G. H. Golub, C. F. Van Loan. *Matrix Computations*. John Hopkins, 3rd ed., 1996.
- [8] A. Tkacenko and P. Vaidyanathan. Iterative greedy algorithm for solving the fir paraunitary approximation problem. *IEEE Transactions on Signal Processing*, 54(1):146–160, Jan. 2006.
- [9] A. Tkacenko. Approximate Eigenvalue Decomposition of Para-Hermitian Systems through Successive FIR Paraunitary Transformations. In *IEEE International Con-*

References

- ference on Acoustics, Speech, and Signal Processing*, pp. 4074–4077, Dallas, TX, USA, March 2010.
- [10] S. Redif, S. Weiss, J. G. McWhirter. Sequential Matrix Diagonalization Algorithms for Polynomial EVD of Parahermitian Matrices. *IEEE Transactions on Signal Processing*, 63(1):81–89, January 2015.
- [11] T. Kailath. *Linear Systems*. Prentice-Hall, Englewood Cliffs, New Jersey, 1980.
- [12] M. Alrmah, S. Weiss, 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, August 2011.
- [13] M. A. Alrmah, J. Corr, A. Alzin, K. Thompson, S. Weiss. Polynomial Subspace Decomposition for Broadband Angle of Arrival Estimation. In *Sensor Signal Processing for Defence Conference*, Edinburgh, Scotland, September 2014.
- [14] S. Weiss, M. Alrmah, S. Lambotharan, J. G. McWhirter, M. Kaveh. Broadband Angle of Arrival Estimation Methods in a Polynomial Matrix Decomposition Framework. In *5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, St. Martin, December 2013.
- [15] A. Alzin, F. K. Coutts, J. Corr, S. Weiss, I. K. Proudler, J. A. Chambers. Adaptive Broadband Beamforming with Arbitrary Array Geometry. In *IET Conference on Intelligent Signal Processing*, London, England, December 2015.
- [16] A. Alzin, F. K. Coutts, J. Corr, S. Weiss, I. K. Proudler, J. Chambers. Polynomial Matrix Formulation Based Capon Beamformer. In *2016 IMA International Conference on Signal Processing in Mathematics*, Birmingham, England, December 2016.
- [17] S. Redif, J. G. McWhirter, P. D. Baxter, T. Cooper. Robust Broadband Adaptive Beamforming via Polynomial Eigenvalues. In *IEEE/MTS OCEANS*, Boston, Massachusetts September 2006.
- [18] S. Weiss, S. Redif, T. Cooper, C. Liu, P. D. Baxter, J. G. McWhirter. Paraunitary Oversampled Filter Bank Design for Channel Coding. *Journal of Applied Signal Processing*, 2006.

References

- [19] S. Weiss, C. H. Ta, C. Liu. A Wiener Filter Approach to the Design of Filter Bank Based single-carrier precoding and equalisation. In *IEEE International Symposium on Power Line Communications and Its Applications*, pages 493–498, Pisa, Italy, March 26-28 2007.
- [20] F. Labeau, L. Vandendorpe, and B. Macq. Oversampled Filter Banks as Error Correcting Codes. In *5th International Symposium on Wireless Personal Multimedia Communications*, volume 3, pages 1265–1269, Honolulu, HI, October 2002.
- [21] F. Labeau, J. C. Chaing, M. Keiffer, P. Duhamel, L. Vandendorpe, B. Macq. Oversampled Filter Banks as Error Correcting Codes: Theory and Impulse Noise Correction. *IEEE Transactions on Signal Processing*, 53(12):4619–4630, December 2005.
- [22] S. Weiss. On the Design of Oversampled Filter Banks for Channel Coding. In *12th European Signal Processing Conference*, Vienna, Austria, September 2004.
- [23] R. Brandt, M. Bengtsson. Wideband MIMO Channel Diagonalization in the Time Domain. In *IEEE 22nd International Symposium on Personal Indoor and Mobile Radio Communications*, Toronto, Ontario, Canada, September 2011.
- [24] C. H. Ta, S. Weiss. A Design of Precoding and Equalisation for Broadband MIMO Systems. In *41st Asilomar Conference on Signals Systems and Computers*, Pacific Grove, California, USA, November 2007.
- [25] N. Moret, A. Tonello, S. Weiss. MIMO Precoding for Filter Bank Modulation Systems Based on PSVD. In *IEEE 73rd Vehicular Technology Conference*, Budapest, Hungary, May 2011.
- [26] C. H. Ta, S. Weiss. A Design of Precoding and Equalisation for Broadband MIMO Systems. In *15th International Conference on Digital Signal Processing*, pp. 571–574, Cardiff, UK, July 2007.
- [27] C. H. Ta, S. Weiss. A Jointly Optimal Precoder and Block Decision Feedback Equaliser Design with Low Redundancy. In *15th European Signal Processing Conference*, Poznan, Poland, September 2007.

References

- [28] J. Foster, J. G. McWhirter, S. Lambbotharan, I. K. Proudler, M. Davies, J. Chambers. Polynomial Matrix QR Decomposition for the Decoding of Frequency Selective Multiple-Input Multiple-Output Communication Channels. *IET Signal Processing*, 6(7) 704–712, September 2012.
- [29] P. A. Regalia, P. Loubaton. Rational Subspace Estimation using Adaptive Lossless Filters *IEEE Transactions on Signal Processing*, 40(10) 2392–2405, October 1992.
- [30] S. Redif, J. G. McWhirter, S. Weiss. Design of FIR Paraunitary Filter Banks for Subband Coding using a Polynomial Eigenvalue Decomposition. *IEEE Transactions on Signal Processing*, 59(11):5253–5264, November 2011.
- [31] P. P. Vaidyanathan. Theory of Optimal Orthonormal Subband Coders. *IEEE Transactions on Signal Processing*, 46(6):1528–1543, June 1998.
- [32] J. Foster, J. G. McWhirter, M. Davies, J. Chambers. An Algorithm for Calculating the QR and Singular Value Decompositions of Polynomial Matrices. In *IEEE Transactions on Signal Processing*, 58(3), March 2010.
- [33] J. G. McWhirter. An Algorithm for Polynomial Matrix SVD Based on Generalised Kogbetliantz Transformations. In 18th European Signal Processing Conference, Aalborg, Denmark, August 2010.
- [34] F. K. Coutts, J. Corr, K. Thompson, S. Weiss, I. K. Proudler, J. G. McWhirter. Multiple Shift QR Decomposition for Polynomial Matrices. In *2016 IMA International Conference on Signal Processing in Mathematics*, Birmingham, England, December 2016.
- [35] P. Ginzberg, C. Mavroyiakoumou. The QRD and SVD of Matrices Over a Real Algebra. *Linear Algebra and its Applications*, 504 27–47, Elsevier, April 2016.
- [36] R. H. Lambert, M. Joho, H. Mathis. Polynomial Singular Values for a Number of Wideband Source Estimation and Principal Components Analysis. In *3rd International Conference on Independent Component Analysis and Blind Signal Separation*, San Diego, California, December 2001.

References

- [37] M. Tohidian, H. Amindavar, A. M. Reza. A DFT-based Approximate Eigenvalue and Singular Value Decomposition of Polynomial Matrices. *EURASIP Journal on Advances in Signal Processing*, December 2013, 2013:93.
- [38] I. Gohberg, P. Lancaster, and L. Rodman. *Matrix Polynomials*. Academic Press, New York, 1982.
- [39] J. Corr, K. Thompson, S. Weiss, J. G. McWhirter, S. Redif, I. K. Proudler. Multiple Shift Maximum Element Sequential Matrix Diagonalisation for Parahermitian Matrices. In *IEEE Statistical Signal Processing Workshop*, pp. 312–315, Gold Coast, Australia, June 2014.
- [40] J. Corr, K. Thompson, S. Weiss, J. G. McWhirter, I. K. Proudler. Causality-Constrained Multiple Shift Sequential Matrix Diagonalisation for Parahermitian Matrices. In *22nd European Signal Processing Conference*, Lisbon, Portugal, September 2014.
- [41] J. Corr, K. Thompson, S. Weiss, J. G. McWhirter, I. K. Proudler. Maximum Energy Sequential Matrix Diagonalisation for Parahermitian Matrices. In *48th Asilomar Conference on Signals Systems and Computers*, Pacific Grove, California, USA, November 2014.
- [42] Z. Wang, J. G. McWhirter, J. Corr, S. Weiss, I. K. Proudler. Multiple Shift Second Order Sequential Best Rotation Algorithm for Polynomial Matrix EVD. In *23rd European Signal Processing Conference*, Nice, France, September 2015.
- [43] J. Corr, K. Thompson, S. Weiss, I. K. Proudler, J. G. McWhirter. Row-Shift Corrected Truncation of Paraunitary Matrices for PEVD Algorithms. In *23rd European Signal Processing Conference*, Nice, France September 2015.
- [44] J. Corr, K. Thompson, S. Weiss, I. K. Proudler, J. G. McWhirter. Reduced Search Space Multiple Shift Maximum Element Sequential Matrix Diagonalisation Algorithm. In *IET Conference on Intelligent Signal Processing*, London, England December 2015.

References

- [45] J. Corr, K. Thompson, S. Weiss, J. G. McWhirter, I. K. Proudler. Cyclic-by-Row Approximation of Iterative Polynomial EVD Algorithms. In *Sensor Signal Processing for Defence Conference*, Edinburgh, Scotland, September 2014.
- [46] J. Corr, K. Thompson, S. Weiss, I. K. Proudler, J. G. McWhirter. Impact of Source Model Matrix Conditioning on PEVD Algorithms. In *IET Conference on Intelligent Signal Processing*, London, England, December 2015.
- [47] J. Corr, J. Pestana, S. Weiss, S. Redif, M. Moonen. Investigation of a Polynomial Matrix Generalised EVD for Multi-Channel Weiner Filtering. In *50th Asilomar Conference on Signals Systems and Computers*, Pacific Grove, California, USA, November 2016.
- [48] S. Weiss, J. Corr, K. Thompson, J. G. McWhirter, I. K. Proudler. Polynomial EVD Toolbox Available online: <http://pevd-toolbox.eee.strath.ac.uk/> Published 2014, Accessed March 2017.
- [49] E. W. Weisstein. *The CRC Concise Encyclopedia of Mathematics* CRC Press, New York, 1999.
- [50] S. Icart, P. Comon. Some Properties of Laurent Polynomial Matrices. In *IMA International Conference on Mathematics in Signal Processing*, Birmingham, UK, December 2012.
- [51] F. Delgoshia, F. Faramarz. Results on the Factorization of Multidimensional Matrices for Paraunitary Filterbanks over the Complex Field. *IEEE Transactions on Signal Processing*, 52(5) 1289–1303, May 2004.
- [52] J. G. McWhirter, Z. Wang. A Novel Insight to the SBR2 Algorithm for Diagonalising Para-Hermitian Matrices. In *11th International Conference on Mathematics in Signal Processing*, Birmingham, England, December 2016.
- [53] A. Jafarian, J. G. McWhirter. A Novel Method for Multichannel Spectral Factorization. In *20th European Signal Processing Conference*, pp. 1069–1073, Bucharest, Romania, August 2012.

References

- [54] C. H. Ta, S. Weiss. Shortening the order of paraunitary matrices in SBR2 algorithm. In *International Conference on Information Communications and Signal Processing*, pp. 1-5, Singapore, December 2007.
- [55] J. Foster, J. G. McWhirter, J. Chambers. Limiting the Order of Polynomial Matrices within the SBR2 Algorithm. In *IMA International Conference on Mathematics in Signal Processing*, Cirencester, UK, December 2006.
- [56] F. K. Coutts, J. Corr, K. Thompson, S. Weiss, I. K. Proudler. Divide-and-Conquer Sequential Matrix Diagonalisation. *Sensor Signal Processing for Defence Conference*, London, England, December 2017.
- [57] Z. Wang, J. G. McWhirter, and S. Weiss. Multichannel Spectral Factorization Algorithm Using Polynomial Matrix Eigenvalue Decomposition. In *49th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, November 2015.
- [58] A. Aho, J. Ullman. *Foundations of Computer Science* W. H. Freeman, New York, 1994.
- [59] J. Corr, K. Thompson, S. Weiss, J. G. McWhirter, I. K. Proudler. Performance Trade-Offs in Sequential Matrix Diagonalisation Search Strategies. In *6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, Cancún, Mexico, December 2015.
- [60] T.R. Crossley, B. Porter. Eigenvalue and Eigenvector Sensitivities in Linear Systems Theory. *International Journal of Control*, 10(2):163–170, 1969.
- [61] Z. Wang, J. G. McWhirter, J. Corr, S. Weiss. Order-Controlled Multiple Shift SBR2 Algorithm for Para-Hermitian Polynomial Matrices. In *2016 IEEE Sensor Array and Multichannel Signal Processing Workshop*, Rio de Janeiro, Brazil, July 2016.
- [62] J. Götze. Monitoring the Stage of Diagonalization in Jacobi-type Methods. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Adelaide, Australia, vol. III, pp. 441–444, April 1994.

References

- [63] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York, 3rd edition, 1991.
- [64] N. J. Fliege. *Multirate Digital Signal Processing: Multirate Systems, Filter Banks, Wavelets*. John Wiley & Sons, Chichester, 1994.
- [65] R. E. Crochiere, L. R. Rabiner. *Multirate Digital Signal Processing*. Prentice Hall, Englewood Cliffs, NJ, 1983.
- [66] H. Hung and M. Kaveh. Focussing matrices for coherent signal-subspace processing. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 36(8):1272–1281, Aug. 1988.
- [67] H. Wang and M. Kaveh. Coherent signal-subspace processing for the detection and estimation of angles of arrival of multiple wide-band sources. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 33(4):823–831, Aug 1985.
- [68] H. Wang and M. Kaveh. On the performance of signal-subspace processing—part ii: Coherent wide-band systems. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(11):1583–1591, Nov 1987.
- [69] J. Dmochowski, J. Benesty, and S. Affes. Direction of arrival estimation using eigenanalysis of the parameterized spatial correlation matrix. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume I, pages 1–4, April 2007.
- [70] J. Dmochowski, J. Benesty, and S. Affes. Direction of arrival estimation using the parameterized spatial correlation matrix. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(4):1327–1339, May 2007.
- [71] A. Hassani, A. Bertrand, and M. Moonen. GEVD-Based Low-Rank Approximation for Distributed Adaptive Node-Specific Signal Estimation in Wireless Sensor Networks. *IEEE Transactions on Signal Processing*, 64(10):2557–2572, May 2016.
- [72] R. Serizel, M. Moonen, B. V. Dijk, and J. Wouters. Low-Rank Approximation Based Multichannel Wiener Filter Algorithms for Noise Reduction With Application in Cochlear Implants. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4):785–799, April 2014.

References

- [73] A. Mertins. MMSE Design of Redundant FIR Precoders for Arbitrary Channel Lengths. *IEEE Transactions on Signal Processing*, 51(9):2402–2409, September 2003.
- [74] S. Weiss, A. Millar, R. W. Stewart. Inversion of Parahermitian Matrices. In *18th European Signal Processing Conference*, pages 447–451, Aalborg, Denmark, August 2010.
- [75] I. W. Selesnick. The Double Density Discrete Wavelet Transform. *Wavelets in Signal and Image Analysis: From Theory to Practice*. Kluwer, 2001.
- [76] M. Caus and A. Perez-Neira. Multi-stream transmission for highly frequency selective channels in MIMO-FBMC/OQAM systems. *IEEE Transactions on Signal Processing*, 62(4):786–796, February 2014.
- [77] X. Mestre and D. Gregoratti. A parallel processing approach to filterbank multicarrier MIMO transmission under strong frequency selectivity. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8078–8082, May 2014.
- [78] A. Sandmann, A. Ahrens, and S. Lochmann. Resource allocation in svd-assisted optical mimo systems using polynomial matrix factorization. In *Proceedings of 16. ITG Symposium Photonic Networks*, pages 1–7, May 2015.
- [79] M. Davies, S. Lambbotharan, and J. McWhirter. Broadband MIMO beamforming using spatial-temporal filters and polynomial matrix decomposition. In *15th International Conference on Digital Signal Processing*, pages 579–582, Cardiff, UK, July 2007.
- [80] P. Karagiannakis, K. Thompson, J. Corr, I. K. Proudler, S. Weiss. Distributed Processing of a Fractal Array Beamformer. In *IET Intelligent Signal Processing Conference*, London, England, December 2013.
- [81] J. Corr, K. Thompson, S. Weiss, I. K. Proudler, J. G. McWhirter. Shortening of Paraunitary Matrices Obtained by Polynomial Eigenvalue Decomposition Algorithms. In *Sensor Signal Processing for Defence Conference*, Edinburgh, Scotland, September 2015.

References

- [82] F. K. Coutts, J. Corr, K. Thompson, S. Weiss, I. K. Proudler, J. G. McWhirter. Memory and Complexity Reduction in Parahermitian Matrix Manipulations of PEVD Algorithms. In *24th European Signal Processing Conference*, Budapest, Hungary, September 2016.
- [83] F. K. Coutts, J. Corr, S. Weiss, I. K. Proudler, J. G. McWhirter. Complexity and Search Space Reduction in Cyclic-by-Row PEVD Algorithms. In *50th Asilomar Conference on Signals Systems and Computers*, Pacific Grove, California, USA, November 2016.