

# Advanced Polynomial Matrix Factorization Techniques

## PhD Thesis

Faizan Ahmad Khattak

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

February 5, 2024

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

The advancements in polynomial matrix algebra have significantly bolstered the formulation of broadband array problems, enabling more effective solutions. This is particularly evident through the utilization of polynomial matrix factorization techniques such as the polynomial eigenvalue decomposition (PEVD), the polynomial singular value decomposition (PSVD), and the polynomial QR decomposition (PQRD), which have notably revolutionized the approach to solving these complex problems. Therefore, these polynomial matrix factorization methods have received significant attention over the last decades. Still, the polynomial order of the decompositions, complexity scaling with spatial and temporal dimensions of the matrix, and issues regarding parallelizability remain pertinent. Therefore, this thesis address these issues in these three mentioned polynomial factorization methods.

This thesis demonstrates that in most practical situations, eigen- and singular values will be spectrally majorised. We exploit this property in the algorithms proposed in this thesis. The first and foremost algorithm is for applications such as data compaction or dominant component extraction, where the power method is extended to the para-Hermitian polynomial matrices for the extraction of a dominant eigenvector. This approach prevents computing an entire PEVD of a para-Hermitian matrix. Later, this extension is combined with a deflation approach to compute the PEVD of a low rank polynomial matrix. Perturbation bounds are computed which show that these bounds increase with repeated deflations. Ensemble tests reveal its superior performance over state-of-the-art PEVD algorithms. To accommodate non-para-Hermitian polynomial matrices, the polynomial power method is extended to general polynomial matrices for the extraction of dominant left and right singular vectors and the corresponding

dominant singular value.

To reduce iterations in the polynomial power method to just one, the provided polynomial matrix is decomposed into a sum of rank-one matrices using a computationally inexpensive approach in the discrete Fourier transform (DFT) domain. Subsequently, each rank-one term is subjected to a single iteration of the polynomial power method, resulting in the corresponding eigenpair or the left- and right singular vectors and the associated singular value. For the PQRD, rank one terms are obtained by computing QRDs in the DFT bins. To obtain any column of the paraunitary matrix and the corresponding row of the upper-right triangular matrix, any column of the respective rank one matrix is normalized to unit norm on the unit circle. This rank one decomposition based method is termed as unified-I algorithm which is highly parallelizable and outperforms all state-of-the-art algorithms in accuracy of the decomposition and execution time.

Lastly, this thesis demonstrates that a further reduction in the order of any of the above decompositions can be achieved via assessing the auto- and cross-correlation terms of eigen- or singular vectors. A spectral factorisation of these terms can then lead to the compact polynomial order factors. This spectral factorization is obtained via polynomial root finding method to avoid the positive definite restriction of a Laurent polynomial on the unit circle. Ensemble results show its superior performance over state-of-the-art algorithms including the rank decomposition method proposed in this thesis.

# Contents

<b>Abstract</b>	<b>2</b>
<b>Abbreviations</b>	<b>9</b>
<b>Mathematical Notations</b>	<b>10</b>
<b>List of Symbols</b>	<b>11</b>
<b>List of Figures</b>	<b>16</b>
<b>List of Publications</b>	<b>17</b>
<b>Acknowledgements</b>	<b>19</b>
<b>1 Introduction</b>	<b>20</b>
1.1 Motivation . . . . .	20
1.2 Objective of Research . . . . .	22
1.3 Original Contributions . . . . .	23
1.4 Organization of Thesis . . . . .	25
<b>2 Background</b>	<b>26</b>
2.1 Polynomial Matrices in Broadband Scenarios . . . . .	26
2.1.1 Instantaneous and Convolutional Mixing: Channel Matrix . . . . .	26
2.1.2 Space-Time Covariance and Cross-Spectral Density (CSD) Matrix	28
2.2 Polynomial Matrix Eigenvalue Decomposition . . . . .	29
2.2.1 Existence of Analytic EVD . . . . .	29

2.2.2	PEVD Algorithms . . . . .	30
2.3	Polynomial Matrix Singular Value Decomposition . . . . .	34
2.3.1	Existence of Analytic SVD . . . . .	34
2.3.2	PSVD Algorithms . . . . .	34
2.4	Polynomial Matrix QR Decomposition . . . . .	36
2.4.1	Existence and Uniqueness of the PQRD . . . . .	36
2.4.2	Existing Iterative PQRD Techniques . . . . .	37
2.5	Summary . . . . .	39
<b>3</b>	<b>Space-Time Covariance Estimation</b>	<b>41</b>
3.1	Introduction . . . . .	41
3.2	Estimation and Its Perturbation Effects . . . . .	42
3.2.1	Perturbation of Eigenvalues . . . . .	43
3.2.2	Unbiased Estimator . . . . .	43
3.3	Space-Time Covariance Estimation via System Identification . . . . .	45
3.3.1	Source Model and Space-Time Covariance Matrix . . . . .	45
3.3.2	Estimation via System Identification . . . . .	45
3.3.3	Simulations and Comparison . . . . .	48
3.3.4	Conclusion . . . . .	50
3.4	Eigenvalues of an Estimated Space-Time Covariance Matrix . . . . .	51
3.4.1	Eigenvalues at an Algebraic Multiplicity . . . . .	51
3.4.2	Impact on Analytic Eigenvalues . . . . .	52
3.4.3	Impact of Sample Size . . . . .	53
3.4.4	Impact on Applications . . . . .	56
3.4.5	Conclusions . . . . .	58
3.5	Summary . . . . .	59
<b>4</b>	<b>Polynomial Power Method</b>	<b>60</b>
4.1	Introduction . . . . .	60
4.2	Power Iterations Method . . . . .	62
4.3	Polynomial Power Method (PPM) . . . . .	63

4.3.1	Power Method Extension to Para-Hermitian Matrices . . . . .	63
4.3.2	Normalisation and Principal Eigenvalue . . . . .	65
4.3.3	Order Limitation by Shift-Corrected Truncation . . . . .	66
4.3.4	Stopping Criterion . . . . .	68
4.3.5	Simulations and Results . . . . .	69
4.4	Low-rank PEVD through Deflation . . . . .	73
4.4.1	EVD via Power Method and Deflation . . . . .	73
4.4.2	Para-Hermitian Matrix Rank One Representation and Deflation . . . . .	75
4.4.3	Perturbation Analysis and Error Propagation . . . . .	76
4.4.4	Application and Ensemble Simulation . . . . .	79
4.5	Ordinary Generalised Power Method . . . . .	81
4.6	Generalized Polynomial Power Method . . . . .	82
4.6.1	Polynomial Iterations Analysis . . . . .	82
4.6.2	Sufficient DFT Size . . . . .	84
4.6.3	Simulations and Results . . . . .	85
4.7	Summary . . . . .	88
<b>5</b>	<b>Factorization Via Rank One Decomposition</b>	<b>90</b>
5.1	Introduction . . . . .	90
5.2	PEVD by Rank One Decomposition . . . . .	92
5.2.1	Objective and Rationale . . . . .	92
5.2.2	Para-Hermitian Matrix Rank One Decomposition . . . . .	92
5.2.3	Single Polynomial Power Iteration . . . . .	94
5.2.4	Normalization . . . . .	95
5.2.5	Choices and Issues in Initialization . . . . .	96
5.2.6	Solution to Normalization Issue . . . . .	96
5.2.7	Truncation . . . . .	103
5.2.8	Reduced PEVD to Full PEVD . . . . .	104
5.3	Normalization Free PEVD Approach . . . . .	105
5.3.1	Analysis for Normalization Free Approach . . . . .	106
5.3.2	Eigenvector Support and Initialization Choices . . . . .	107

5.4	PSVD by Rank One Decomposition . . . . .	107
5.4.1	Rank One Decomposition of $\mathbf{A}(z)$ . . . . .	108
5.4.2	Single Polynomial Power Iteration Application . . . . .	109
5.4.3	Reduced to Full PSVD . . . . .	110
5.5	Normalization Free PSVD Approach . . . . .	110
5.6	PQRD by Rank One Decomposition . . . . .	111
5.6.1	QR-based Rank One Decomposition . . . . .	111
5.6.2	Normalizing Columns of $\hat{\mathbf{A}}_n(z)$ for PQRD . . . . .	112
5.7	Normalization Free PQRD Approach . . . . .	113
5.8	Simulation and Results . . . . .	114
5.8.1	Performance Metrics . . . . .	114
5.8.2	PEVD . . . . .	115
5.8.3	PSVD . . . . .	116
5.8.4	PQRD . . . . .	118
5.9	Summary . . . . .	119
<b>6</b>	<b>Factorization Via Spectral Factorization</b>	<b>122</b>
6.1	Introduction . . . . .	122
6.2	Spectral Factorization of Para-Hermitian Laurent Polynomials . . . . .	124
6.2.1	Roots Method . . . . .	124
6.2.2	Cepstral Method . . . . .	126
6.3	Modified Cepstral Method . . . . .	128
6.4	PEVD Via Spectral Factorization . . . . .	128
6.4.1	Auto- and Cross-Correlation Functions of the Component of Analytic Eigenvectors . . . . .	128
6.4.2	Support Estimation of Analytic Eigenvector . . . . .	130
6.4.3	Spectral Factorization Ambiguity in Context of PEVD . . . . .	131
6.4.4	PEVD Method via Cepstral Method . . . . .	132
6.4.5	PEVD via Roots Method . . . . .	134
6.5	PSVD via Spectral Factorization . . . . .	136
6.6	PQRD via Spectral Factorization . . . . .	138

6.7	Simulation and Results . . . . .	138
6.7.1	PEVD . . . . .	138
6.7.2	PSVD . . . . .	141
6.7.3	PQRD . . . . .	143
6.8	Summary . . . . .	143
<b>7</b>	<b>Conclusion &amp; Future Work</b>	<b>147</b>
7.1	Thesis Summary . . . . .	147
7.1.1	Space-Time Covariance Estimation and Loss of Multiplicities . .	147
7.1.2	Polynomial Power Method . . . . .	148
7.1.3	Rank One Decomposition . . . . .	148
7.1.4	Spectral Factorization for Polynomial Matrix Decomposition . .	149
7.2	Applications-Oriented Comparison . . . . .	149
7.3	Future Work . . . . .	149
	<b>Bibliography</b>	<b>164</b>

# Abbreviations

MIMO	Multiple Input Multiple Output
EVD	Eigenvalue Decomposition
SVD	Singular Value Decomposition
FIR	Finite Impulse Response
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
IFFT	Inverse Fast Fourier Transform
QRD	QR Decomposition
PEVD	Polynomial Eigenvalue Decomposition
PSVD	Polynomial
PQRD	Polynomial QR Decomposition
SBR2	Second Order Sequential Best Rotation
GSBR2	Generalized Second Order Sequential Best Rotation
SMD	Sequential Matrix Diagonalization
GSMD	Generalized Sequential Matrix Diagonalization
PQRD-BC	Polynomial QR Decomposition By Columns
PQRD-BS	Polynomial QR Decomposition By Steps
PPM	Polynomial Power Method
GPPM	Generalized Polynomial Power Method
RPEVD	Roots Based Polynomial Eigenvalue Decomposition
CSD	Cross-Spectral Density

# Mathematical Notations

$\mathcal{E}\{\cdot\}$	Expectation
$\mathcal{O}\{\cdot\}$	polynomial order of the argument
$\{\cdot\}^H$	Hermitian
$\{\cdot\}^P$	Para-Hermitian operation
$\text{diag}\{\cdot\}$	Diagonal matrix constructed of entries in bracket
$\bullet \rightarrow \circ$	Transform pair
$\hat{\{\cdot\}}$	Estimate of the argument
$\lceil \cdot \rceil$	Ceiling operator

# List of Symbols

$\mathbf{R}[\tau]$	Space-time covariance matrix
$\mathbf{R}(z)$	Cross spectral density matrix or para-Hermitian matrix
$\mathbf{R}_m(z)$	The $m$ th rank one term of $\mathbf{R}(z)$
$\mathbf{R}_k$	The $k$ th DFT component of the $K$ -point DFT of $\mathbf{R}[\tau]$
$\mathbf{\Lambda}(z)$	Para-Hermitian diagonal matrix of polynomial eigenvalues
$\lambda_m(z)$	The $m$ th singular value of $\mathbf{R}(z)$
$\mathbf{Q}(z)$	Paraunitary matrix containing eigenvectors
$\mathbf{q}_m(z)$	The $m$ th eigenvector of $\mathbf{R}(z)$ or the $m$ th column of $\mathbf{Q}(z)$
$\mathbf{q}_{m,k}$	The $m$ th eigenvector of $\mathbf{R}_k$ obtained through an ordinary EVD
$\mathbf{A}(z)$	Channel matrix or non-para-Hermitian polynomial matrix
$\mathbf{A}_m(z)$	The $m$ th rank one term of $\mathbf{A}(z)$ through PSVD
$\mathbf{U}(z)$	Paraunitary matrix containing left singular vectors
$\mathbf{u}_m(z)$	The $m$ th column of $\mathbf{U}(z)$
$\mathbf{V}(z)$	Paraunitary matrix containing right singular vectors
$\mathbf{v}_m(z)$	The $m$ th column of $\mathbf{V}(z)$
$\mathbf{\Sigma}(z)$	Diagonal matrix containing polynomial singular values
$\sigma_m(z)$	The $m$ th singular value of $\mathbf{A}(z)$
$\underline{\mathbf{A}}_m(z)$	The $m$ th rank one term of $\mathbf{A}(z)$ through PQRD
$\underline{\mathbf{R}}(z)$	Upper-right triangular matrix
$\underline{\mathbf{Q}}(z)$	Paraunitary matrix of polynomial QR decompsition
$\underline{\mathbf{q}}_m(z)$	The $m$ th column of $\underline{\mathbf{Q}}(z)$

$\mathbf{x}[n]$	Sensor array measurement at time index $n$
$\mathbf{x}^{(i)}(z)$	$i$ th iteration polynomial vector of polynomial power method
$a_{m,n}[\tau]$	The $m$ th row and $n$ th column element of $\mathbf{A}[\tau]$
$a_{m,n}(z)$	The $m$ th row and $n$ th column element of $\mathbf{A}(z)$
$r_{m,n}(z)$	The $m$ th row and $n$ th column element of $\mathbf{R}(z)$
$M$	Number of sensors in an array
$L$	Number of sources in MIMO system
$\xi_\sigma$	Normalized error of the estimated polynomial singular value
$\xi_\lambda$	Normalized error of the estimated polynomial eigenvalue
$\zeta_{\{\cdot\}}$	Time-domain aliasing of the argument in brackets

# List of Figures

2.1	System matrix based source model for the measurement vector $\mathbf{x}[n]$ . . .	27
3.1	Ensemble results for $\zeta_{\text{norm}}$ when obtaining $\hat{\mathbf{R}}[\tau]$ in dependence of adaptive filter length, $L_f$ . . . . .	48
3.2	Comparison of estimation methods via an ensemble of $\mathbf{R}[\tau] \in \mathbb{C}^{2 \times 2}$ , showing the measured error via the unbiased estimator and the system identification approach. . . . .	50
3.3	Example for (a) analytic eigenvalues and (b) Hermitian angles of their corresponding analytic eigenvectors. . . . .	52
3.4	Normalised approximate probability density functions $p(\hat{\lambda})$ for eigenvalues $\hat{\lambda}$ of $\hat{\mathbf{R}}(e^{j\pi/4})$ , estimated for a number of different sample sizes $N$ , from each $10^5$ instances. . . . .	53
3.5	(a) eigenvalues $\hat{\lambda}_m(e^{j\Omega})$ for $N = 100$ (coloured curves) and ground truth $\lambda_m(e^{j\Omega})$ (in grey, underlaid); Hermitian angles $\alpha_m(\Omega)$ for the corresponding eigenvectors $\hat{\mathbf{q}}(e^{j\Omega})$ and $\mathbf{q}(e^{j\Omega})$ . . . . .	55
3.6	(a) eigenvalues $\hat{\lambda}_m(e^{j\Omega})$ for $N = 10^4$ (coloured curves) and ground truth $\lambda_m(e^{j\Omega})$ (in grey, underlaid); Hermitian angles $\alpha_m(\Omega)$ for the corresponding eigenvectors $\hat{\mathbf{q}}(e^{j\Omega})$ and $\mathbf{q}(e^{j\Omega})$ . . . . .	55
3.7	(a) eigenvalues $\hat{\lambda}_m(e^{j\Omega})$ for $N = 10^6$ (coloured curves) and ground truth $\lambda_m(e^{j\Omega})$ (in grey, underlaid); Hermitian angles $\alpha_m(\Omega)$ for the corresponding eigenvectors $\hat{\mathbf{q}}(e^{j\Omega})$ and $\mathbf{q}(e^{j\Omega})$ . . . . .	57

3.8	(a) eigenvalues $\hat{\lambda}_m(e^{j\Omega})$ estimated via system identification at 20 dB SNR for $N = 10^6$ (coloured curves) and ground truth $\lambda_m(e^{j\Omega})$ (in grey, underlaid); Hermitian angles $\alpha_m(\Omega)$ for the corresponding eigenvectors $\hat{\mathbf{q}}(e^{j\Omega})$ and $\mathbf{q}(e^{j\Omega})$ . . . . .	57
4.1	Optimal shift for approximation of a 100 order vector $\mathbf{q}(z)$ , with unit norm on the unit circle, by an $N = 50$ th order polynomial using a suitable window to minimise the truncation error. . . . .	67
4.2	Eigenvalues for the example matrix $\mathbf{R}(z)$ . . . . .	70
4.3	Effects of types of order limitation on algorithm convergence, measuring (a) angle deviation metric $\gamma$ , and (b) accuracy of the extracted eigenvalue. . . . .	70
4.4	Convergence of $\beta(\Omega) = \angle\{\tilde{\mathbf{v}}_{\text{norm}}^{(k)}(e^{j\Omega}), \mathbf{q}_1(e^{j\Omega})\}$ in case of a singularity of $c_1(e^{j\Omega})$ for $\Omega = 0$ . . . . .	72
4.5	Performance metrics comparison (a) $\xi_\lambda$ , (b) $\mathcal{O}\{\hat{\mathbf{q}}_1(z)\}$ , and (c) execution time of SBR2, SMD and PPM in dependence of the ground truth order of the principal eigenvector. . . . .	73
4.6	Example for the ground truth (shaded grey) and estimated eigenvalues using the proposed deflation approach (in colour) $\hat{\lambda}_m(e^{j\Omega}), m = 1, 2, 3$ for the example matrix $\mathbf{R}(z)$ when injecting perturbation through insufficient convergence of the PPM algorithm. . . . .	78
4.7	Ensemble results illustrated as box-plots for (a) reconstruction error, (b) $\mathcal{O}\{\hat{\mathbf{Q}}(z)\}$ , and (c) execution time. (red marks show outliers) . . . . .	80
4.8	(a) GPPM, and (b) GSBR2 [11] based estimated dominant singular value coefficients for the numerical example. . . . .	87
5.1	Example 7 illustration showing the smooth evolution of both the real and imaginary parts of the components of the normalized vector with $c_m(e^{j\Omega})$ with zero crossing with no phase jump across it. . . . .	97
5.2	Normalization results for $\mathbf{x}^{(1)}(z)$ on the unit circle in presence of even number of zeros of $c_m(e^{j\Omega})$ : given case illustrates 2 zero crossings. . . . .	98

5.3	Inverted section of the curves, shown in Fig. 5.2, between two zero crossings i.e. $\Omega = \{\pi/2, 3\pi/2\}$ . . . . .	99
5.4	Normalized vector components real and imaginary components with even number of zero crossing in $c_m(e^{j\Omega})$ (a) without out sign change, (b) with sign change applied in alternate manner i.e. $\Omega \in \{[0, \pi/4], [\pi/2, 3\pi/4], [\pi, 5\pi/4], [3\pi/2, 2\pi]\}$ . . . . .	100
5.5	Normalized vector in presence of odd number of zero crossings in $c_m(e^{j\Omega})$ , here showing example of 3 with (a) without oversampling and sign inversion (b) with oversampling by 2 and applying sign inversion to section of curve between alternate zero crossings. (arrows indicate zero crossings associated with phase discontinuities) . . . . .	101
5.6	Detection of zero crossings of $c_m(e^{j\Omega})$ associated with phase discontinuities via a combined observation from (a) phase discontinuities observed in the components of $\mathbf{x}_{\text{norm}}^{(1)}(e^{j\Omega})$ , and (b) magnitude profile $\ \mathbf{x}^{(1)}(e^{j\Omega})\ _2$ in case some or all zero crossing do not coincide with the DFT bin. . . . .	103
5.7	PEVD algorithm comparison on an ensemble showing (a) $\xi_{\hat{R}}$ , (b) $\mathcal{O}\{\hat{\mathbf{Q}}(z)\}$ , and (c) execution time. . . . .	116
5.8	Comparison of the normalization-free PSVD algorithm against the GSBR2 [11] through: (a) $\mathcal{O}\{\hat{\mathbf{V}}(z)\}$ , (b) $\mathcal{O}\{\hat{\mathbf{U}}(z)\}$ , (c) $\xi_A$ , and (d) time (s) versus order of ground-truth $\mathbf{U}(z)$ . . . . .	117
5.9	Normalization-free PQRD algorithm comparison with SM-PQRD [19] and PQRD-BC [12] through (a) $\eta$ , (b) $\mathcal{O}\{\hat{\mathbf{Q}}(z)\}$ , and (c) time (s). . . . .	119
6.1	Performance metrics computed over an ensemble of randomized para-Hermitian matrices for SBR2 [13], SMD [22], normalization free variant of the rank one decomposition based PEVD (here denoted as N-Free) and roots based PEVD (RPEVD) with (a) $\xi_R$ , (b) $\mathcal{O}\{\hat{\mathbf{Q}}(z)\}$ and (c) execution time in seconds. . . . .	140

6.2	Performance metrics computed over an ensemble of randomized polynomial matrices for GSR2 [11], 2 PEVDs based PSVD via SMD and roots based PSVD (RPEVD) with (a) $\xi_A$ , (b) $\mathcal{O}\{\hat{\mathbf{Q}}(z)\}$ and (c) execution time in seconds. . . . .	142
6.3	Performance metrics computed over an ensemble of randomized polynomial matrices for PQRD-BC [12], SM-PQRD [19], normalization free variant of rank decomposition method (N-Free) and roots based PQRD (RPEVD) with (a) $\eta$ , (b) $\mathcal{O}\{\hat{\mathbf{Q}}(z)\}$ and (c) execution time in seconds. . . . .	144

# List of Publications

1. F. A. Khattak, S. Weiss, I. K. Proudler, and J. G. McWhirter, “Space-Time Covariance Matrix Estimation: Loss of Algebraic Multiplicities of Eigenvalues,” in 56th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, Oct. 2022, pp. 975–979.
2. F. A. Khattak, I. K. Proudler, and S. Weiss, “Enhanced Space-Time Covariance Estimation Based on a System Identification Approach,” in Sensor Signal Processing for Defence Conference (SSPD), London, United Kingdom, Sep. 2022, pp. 1–5.
3. F. A. Khattak, I. K. Proudler, and S. Weiss, “Extension of power method to para-Hermitian matrices: Polynomial power method,” in 31st European Signal Processing Conference (EUSIPCO), Helsinki, Finland, 2023, pp. 1564–1568.
4. F. A. Khattak, I. K. Proudler, and S. Weiss, “Generalized polynomial power method,” in Sensor Signal Processing for Defence Conference (SSPD), Edinburgh, United Kingdom, 2023, pp. 1–5.
5. F. A. Khattak, I. K. Proudler, and S. Weiss, “Low rank para-Hermitian matrix EVD via polynomial power method with deflation,” in IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), San Jose, Costa Rica, accepted, 2023.
6. F. Khattak, S. Weiss, and I. K. Proudler, “Fast Givens rotation approach to second order sequential best rotation algorithms,” in Sensor Signal Processing for Defence Conference (SSPD), Edinburgh, United Kingdom, 2021, pp. 1–5.

7. F. A. Khattak, I. K. Proudler, J. G. McWhirter, and S. Weiss, “Generalised sequential matrix diagonalisation for the SVD of polynomial matrices,” in Sensor Signal Processing for Defence Conference (SSPD), Edinburgh, United Kingdom, 2023, pp. 1–5.
8. F. A. Khattak, M. Bakhit, I. K. Proudler, and S. Weiss, “Smooth QR decomposition of polynomial matrices,” in IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), San Jose, Costa Rica, accepted, 2023.
9. F. A. Khattak, I. K. Proudler, and S. Weiss, “Support estimation of analytic eigenvectors of parahermitian matrices,” in International Conference on Recent Advances in Electrical Engineering & Computer Sciences (RAEE & CS), Islamabad, Pakistan, 2022, pp. 1–6.
10. M. Bakhit, F. A. Khattak, I. K. Proudler, S. Weiss, and G. Rice, “Compact order polynomial singular value decomposition of a matrix of analytic functions,” in IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), San Jose, Costa Rica, accepted, 2023.
11. S. Weiss, I. K. Proudler, F. K. Coutts, and F. A. Khattak, “Eigenvalue decomposition of a parahermitian matrix: Extraction of analytic eigenvectors,” *IEEE Transactions on Signal Processing*, vol. 71, pp. 1642–1656, 2023.

# Acknowledgements

All praise is for Allah, who blessed me with everything I needed to carry out this research. I would also like to acknowledge the help, support, and expert guidance of my supervisor, Prof. Stephan Weiss at the University of Strathclyde. I am also highly indebted to Ian K. Proudler, who immensely helped me throughout my entire PhD journey with constructive suggestions, critical analysis of my drafts, and refining my ideas regarding new and improved algorithms.

# Chapter 1

## Introduction

### 1.1 Motivation

Ordinary matrix decomposition techniques, also known as matrix factorization methods, are fundamental tools in linear algebra and play an important role in different fields, including mathematics, engineering, data analysis, and computer science [1–5]. These techniques aim to break down a complex matrix into simpler factors, and provide valuable insights into the structure and properties of matrices, enabling efficient and meaningful manipulation of data in many real-world scenarios. There are many matrix factorization methods, but three of the most commonly used ones are the eigenvalue decomposition (EVD), singular value decomposition (SVD), and QR decomposition (QRD) [1]. These matrix factorization methods have been used in angle of arrival estimation [6], image processing, dimensionality reduction and data compaction [5], precoding and equalization for multiple-input and multiple-output (MIMO) system design, multiple signal classification (MUSIC) [7], solving linear system, subspace decomposition and beamforming [8] to name but a few.

Unlike narrowband sensor array system where the matrices of interest, such as the channel matrix or the instantaneous covariance matrix, are ordinary standard matrices, broadband scenarios are represented via polynomial matrices. Due to broadband signals spanning potentially several octaves, the mixing is convolutive and so sensor signals have temporal correlation. To adequately capture this correlation such that

the directional information can be extracted, one cannot not only rely on instantaneous covariance matrix, but instead a space-time covariance matrix is required. Each element of a space-time covariance matrix is a cross-correlation sequence. Since the space-time covariance matrix inherits the symmetries of auto- and cross-correlation sequences, its  $z$ -transform is a polynomial matrix also known as a cross-spectral density (CSD) matrix, which has a para-Hermitian symmetry. For these polynomial matrices, ordinary matrix decomposition techniques are not appropriate. Although one could split a broadband problem into a number of narrowband problems where then ordinary matrix decomposition techniques can be utilized, this leads to loss of coherence between the adjacent frequency bands.

In order to appropriately address broadband sensor array problems, polynomial matrix decomposition techniques can be employed [9]. Therefore, the conventional SVD and QRD techniques have been extended to polynomial SVD (PSVD) [10, 11] and polynomial QRD (PQRD) [12] to decompose the polynomial channel matrix. In addition, the scalar EVD has been extended from Hermitian to para-Hermitian polynomial matrices and defined as polynomial EVD (PEVD) [13]. These polynomial extensions have found similar applications as mentioned above, such as broadband beamforming [8], polynomial MUSIC (PMUSIC) [14–16], speech enhancement [17, 18], frequency-selective channel equalization [19], optimal subband coding [20], and source spectral density estimation [21] etc.

With all recent developments in these polynomial matrix factorization techniques, the implementation cost — directly related to the polynomial order of a decomposition — the accuracy, and the execution time are still relevant issues which need to be addressed. The reason is that all the available time-domain PSVD, PQRD and PEVD algorithms offer approximate factorization with higher order polynomial factors [11–13, 22–26]. Many efforts have been focused on limiting the order of polynomial factors through trimming [27–30] which often somehow reduce the order but can still exceeds the ground-truth polynomial order by an order of magnitude. Moreover, the iterative trimming, particularly during algorithm iterations, causes errors and error propagation. To specifically address the complexity growth with spatial dimension,

a divide-and-conquer method has been employed for time-domain PEVD algorithms. This divide-and-conquer strategy reduces the complexity for higher spatial dimension polynomial matrices at the cost of an increase in both the approximation error and order of the paraunitary matrix [31, 32].

The second category of PEVD, PSVD and PQRD algorithms are DFT-domain techniques which are more focused on producing compact polynomial order decompositions. While they offer lower order polynomial factorizations, they can be computationally far more expensive than the traditional time-domain methods. Despite their enhanced accuracy, the practical deployment of these techniques for broadband applications remains limited due to several challenges. One major constraint is the optimization of a non-convex objective function required for estimating lower order decompositions [33–35]. As a result, the computational overhead associated with these DFT-based methods outweighs the advantages they offer, preventing widespread adoption in real-world applications. Furthermore, the complexity of DFT-based methods grows significantly w.r.t both spatial and temporal dimension [35], making them computationally more intensive compared to the traditional time-domain methods. The difference in complexity can be orders of magnitude, posing a substantial hurdle for their efficient implementation in practical systems.

## 1.2 Objective of Research

The primary aim of this thesis is to address the broadband sensor array challenge related to the estimation process, followed by data processing utilizing polynomial matrix decomposition or factorization techniques.

- The estimation part objective is the enhancement of the estimation of the space-time covariance or CSD matrix from sensor measurements with the analysis of effects of the estimation process on the majorisation [13, 36] property of both channel and CSD matrix.
- The data processing part includes the aim to enhance the existing polynomial matrix decomposition methods and to pioneer novel, computationally efficient

algorithms with superior time and performance metrics.

- Additionally, special emphasis is placed on the development of algorithms that can be seamlessly implemented across diverse platforms, including both fully parallel and partially parallel computing systems. Presently available PEVD, PSVD, and PQRD methods suffer from limited parallelizability, which restricts the full utilization of parallel computing platforms like FPGAs, leading to suboptimal execution timing performance. Consequently, the need arises for advanced algorithms capable of unlocking the potential of real-time broadband sensor array processing, achieved through fully parallelizable algorithms. This parallelizable algorithms need is also pursued in this thesis however, without any hardware implementation.

### 1.3 Original Contributions

To the best of our knowledge, the following aspects represent the original contributions of thesis:

1. **Enhancement of space-time covariance estimation and loss of algebraic multiplicities** [37, 38]

Broadband sensor array applications require estimation of the space-time covariance matrix from sensor measurements via an un-biased estimator. In order to achieve better estimates, a system identification based estimation is studied at various signal-to-noise ratios (SNR). Furthermore, impact of an estimation process on the loss of algebraic multiplicities of eigenvalues of CSD matrix is also investigated.

2. **Extension of power method to polynomial matrices** [39–41] This contribution is composed of several minor contributions which are outlined as following:

- The first contribution extends the well known power method, which is only applicable to ordinary matrices, to para-Hermitian polynomial matrices to extract its dominant eigenpair.

- Low-rank para-Hermitian matrix PEVD, which is often required in broadband sensor array applications, is computed by combining the Hotellings deflation approach with polynomial equivalent of the power method. In addition, perturbation analysis due to repeated deflation is also carried out.
- The polynomial power method is then generalized to ordinary polynomial matrices for the extraction of dominant singular vectors and the corresponding singular value.

### 3. Rank one decomposition based unified algorithm I

Due to the existence of an analytic EVD and SVD for analytic, non-multiplexed para-Hermitian and general polynomial matrix, respectively, any polynomial matrix can be represented as a sum of rank one matrices. Therefore, a DFT-based rank one decomposition method is proposed which simplifies the computation of the PEVD and PSVD through polynomial equivalent of the power method. The same rank one decomposition is also utilized for computing the PQRD of a polynomial matrix. The proposed technique is computationally cheaper but may result in higher order polynomial factorization. The proposed approach performed significantly better against state-of-the-art algorithms in accuracy of decomposition and execution time. The unified algorithm extracts eigenvectors or singular vectors independently, making it fully parallelizable and fulfilling our objective.

### 4. Spectral factorization based unified algorithm II

Due to the availability of cheap and economical algorithms for performing spectral factorization of an auto-correlation sequence of finite time-domain support signal, this contribution adopts similar approach to propose a cheap and an accurate method to compute the PEVD, PSVD and PQRD of a polynomial matrix. The resulting order of the paraunitary filter banks in all three types of decomposition is significantly closer to ground-truth support. Furthermore, it is computationally cheaper and fully parallelizable due to independent extractions of eigenvectors or singular vectors, and hence presents a combination of all desired merits in one

unified algorithm. Limitations of the proposed method along with its comparison with proven benchmarks is reported in detail.

## 1.4 Organization of Thesis

The thesis is organized as follows:

- Chapter 2 presents the background on polynomial matrix factorization and recent algorithmic advancements.
- Chapter 3 introduces the concept of system identification-based enhanced space-time covariance estimation and highlights the loss of eigenvalue multiplicities during the estimation process.
- Chapter 4 extends the power method to polynomial power methods, combines it with deflation, and generalizes it for general polynomial matrices.
- Chapter 5 details the decomposition of polynomial matrices into sums of rank terms, reducing the polynomial power method to a single iteration.
- Chapter 6 focuses on spectral factorization-based matrix factorization for PEVD, PSVD, and PQRD to further reduce the polynomial order of decomposition.
- Chapter 7 provides a summary of the entire thesis.

## Chapter 2

# Background

This chapter offers an overview of polynomial matrix decomposition techniques for broadband sensor array data processing. Section 2.1 provides the background for understanding why polynomial matrices are more suitable for modelling broadband sensor array problems than ordinary matrices. Section 2.2 presents a concise overview of the existence of the PEVD, its associated ambiguities, and the number of algorithms currently available for its computation. Similarly, Sections 2.3 and 2.4 delve into the existence and related ambiguities, along with the currently accessible algorithms, for calculating the PSVD and the PQRD of a polynomial matrix, respectively.

## 2.1 Polynomial Matrices in Broadband Scenarios

### 2.1.1 Instantaneous and Convolutional Mixing: Channel Matrix

Consider a MIMO system with  $L$  sources and  $M$  sensors. The uncorrelated source signals at time instance  $n$  are denoted as  $s_\ell[n]$ , where  $\ell = 1, \dots, L$ , while the sensor measurements are denoted as  $x_m[n]$ , with  $m = 1, \dots, M$  as shown in Fig. 2.1. The measurement noise  $v_m[n]$  is assumed to be zero-mean unit variance and have Gaussian distribution. In a narrowband scenario, the delay between the signal arriving at different sensors is effectively translated into phase shifts. Consequently, the signal propagation can be conceptualized as being governed by a complex gain factor. This factor's magnitude is responsible for signal attenuation or scaling, while its phase ac-

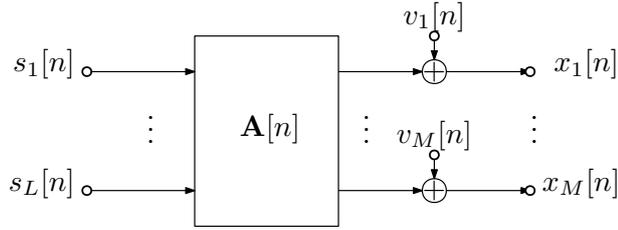


Figure 2.1: System matrix based source model for the measurement vector  $\mathbf{x}[n]$ .

counts for the signal's propagation delay. In light of this, a narrowband MIMO system can be mathematically represented by a complex-valued ordinary matrix such that in the model illustrated in Fig. 2.1 the support of  $\mathbf{A}[n]$  would be unity i.e.  $\mathbf{A} \in \mathbb{C}^{M \times L}$ . This form of signal mixing is commonly referred to as instantaneous mixing, and the communication channel is typically described as a frequency-flat channel. For this reason, in applications such as MIMO decoupling or channel equalisation, conventional SVD and QRD techniques have been used [3, 42, 43].

This instantaneous mixing model is not appropriate for the broadband case. The reason is that the sensor measurements are not simply scaled and phase-shifted versions of the source signals, as is in the case of instantaneous mixing. Instead, the signals become interwoven through convolution, resulting in a more complex relationship between the source signals and the sensor measurements. Therefore, the impulse response between any source and sensor is represented as an FIR filter i.e.  $a_{m,\ell}[n]$  is the impulse response between the  $\ell$ th source and the  $m$ th sensor. The matrix of impulse responses representing the convolutive mixing of  $M \times L$  MIMO system, illustrated in 2.1 is given as

$$\mathbf{A}[n] = \begin{bmatrix} a_{1,1}[n] & a_{1,2}[n] & \dots & a_{1,L}[n] \\ a_{2,1}[n] & a_{2,2}[n] & \dots & a_{2,L}[n] \\ \vdots & & \ddots & \vdots \\ a_{M,1}[n] & a_{M,2}[n] & \dots & a_{M,L}[n] \end{bmatrix}. \quad (2.1)$$

The contribution of source signals to the  $m$ th sensor signal in the presence of additive

noise  $v_m[n]$  can be expressed as

$$x_m[n] = \sum_{\ell=1}^L a_{m,\ell}[n] * s_\ell[n] + v_m[n], \quad (2.2)$$

where  $*$  denotes convolution. With  $\mathbf{s}[n] = [s_1[n], \dots, s_L[n]]^T$  and  $\mathbf{v}[n] = [v_1[n], \dots, v_M[n]]^T$ , the sensor measurement vector  $\mathbf{x}[n]$  can be given as

$$\mathbf{x}[n] = \mathbf{A}[n] * \mathbf{s}[n] + \mathbf{v}[n], \quad (2.3)$$

where  $\mathbf{A}(z) \bullet\!\!\!\rightarrow \mathbf{A}[n]$ ,  $z$ -transform of  $\mathbf{A}[n]$ , is a polynomial matrix. The notation  $\bullet\!\!\!\rightarrow$  denotes a transform pair. To emulate the narrowband MIMO communication design methodology, specifically the employment of pre-coders and equalizers, it is necessary to utilize the polynomial equivalents of SVD and QRD. This ensures the provision of SVD and QRD for each value of  $z$ . Consequently, the conventional SVD and QRD techniques, which enable diagonalization or triangularization for specific values of  $z$ , cannot be readily applied in such scenarios. It should be noted that the notation used in the model in Fig. 2.1 makes it seem causal i.e. only a polynomial matrix, but for generality, this thesis does not restrict the algorithms to polynomial matrix instead to general Laurent polynomial matrices i.e.  $\mathbf{A}(z) \bullet\!\!\!\rightarrow \mathbf{A}[\tau]$  where  $\tau \in \mathbb{Z}$ .

### 2.1.2 Space-Time Covariance and Cross-Spectral Density (CSD) Matrix

In convolutive mixing systems, sensor signals are correlated with each other over a range of time lags. Therefore, to adequately capture the second-order statistics, correlation has to be computed over a range of lags via the space-time covariance matrix

$$\mathbf{R}[\tau] = \mathcal{E}\{\mathbf{x}[n]\mathbf{x}^H[n - \tau]\} \in \mathbb{C}^{M \times M}, \quad (2.4)$$

where  $\mathcal{E}\{\cdot\}$  is the expectation operator,  $\mathbf{x}[n] = [x_1[n], \dots, x_M[n]]^T$  is the sensor measurement vector in (2.3) and  $\tau \in \mathbb{Z}$  is the lag parameter. For  $\tau = 0$ , the space-time covariance reduces to the instantaneous covariance matrix for narrowband sensor ar-

ray applications. The diagonal elements of  $\mathbf{R}[\tau]$  are the auto-correlation sequences and the off-diagonal terms are the cross-correlation sequences. Due to these elements being correlation sequences, the space-time covariance matrix possesses symmetry i.e.  $\mathbf{R}^H[-\tau] = \mathbf{R}[\tau]$ .

The  $z$ -transform of the space-time covariance matrix, denoted as  $\mathbf{R}(z)$ , is referred to as the cross-spectral density (CSD) matrix. The source power spectral densities determine whether  $\mathbf{R}(z)$  is a Laurent polynomial or a Laurent series. However, this thesis assumes that the source signals are uncorrelated and mutually independent, and that  $\mathbf{A}(z)$  is of finite order, which results in  $\mathbf{R}(z)$  being a Laurent polynomial rather than a Laurent series. Because  $\mathbf{R}^H[-\tau] = \mathbf{R}[\tau]$ , the  $z$ -transform  $\mathbf{R}(z)$  exhibits para-Hermitian symmetry. This symmetry implies that  $\mathbf{R}^P(z) = \mathbf{R}(z)$ , where  $\{\cdot\}^P$  represents a para-Hermitian operation that combines time-reversal and Hermitian conjugation i.e.  $\mathbf{R}^P(z) = \mathbf{R}^H(1/z^*)$ . If the channel matrix  $\mathbf{A}(z)$  is known and sources are zero-mean unit variance, the CSD matrix can be given as  $\mathbf{R}(z) = \mathbf{A}^P(z)\mathbf{A}(z) + \text{diag}\{\sigma_{v_1}^2, \dots, \sigma_{v_M}^2\}$  where  $\sigma_{v_m}^2$  is power of noise signal at  $m$ th sensor.

Since the conventional EVD cannot completely decorrelate the broadband sensor array signals, PEVD algorithms are required that can diagonalize  $\mathbf{R}(z)$  for every  $z$ .

## 2.2 Polynomial Matrix Eigenvalue Decomposition

### 2.2.1 Existence of Analytic EVD

A para-Hermitian polynomial matrix  $\mathbf{R}(z) \in \mathbb{C}^{M \times M}$  that is analytic<sup>1</sup> in  $z \in \mathbb{C}$  i.e. can be represented as a convergent Laurent or power series and is infinitely differentiable, and is not connected to any subband type of application, admits an analytic EVD [44, 45]

$$\mathbf{R}(z) = \mathbf{Q}(z)\mathbf{\Lambda}(z)\mathbf{Q}^P(z). \quad (2.5)$$

In (2.5), it is possible to select right hand side factors that are analytic in  $z$ . The columns of the paraunitary matrix  $\mathbf{Q}(z)$ , such that  $\mathbf{Q}(z)\mathbf{Q}^P(z) = \mathbf{I}$ , represent analytic

---

<sup>1</sup>which is identical to its Taylor series expansion

eigenvectors and the diagonal, para-Hermitian matrix  $\mathbf{A}(z) = \text{diag}\{\lambda_1(z), \dots, \lambda_M(z)\}$  contains the analytic eigenvalues. While the eigenvalues are unique up to a permutation, the eigenvectors are subject to an allpass ambiguity: if  $\mathbf{q}_m(z)$  is the  $m$ th column of  $\mathbf{Q}(z)$  and therefore the  $m$ th eigenvector, then  $\phi_m(z)\mathbf{q}_m(z)$  is also a valid  $m$ th eigenvector, where  $\phi_m(z)$  is an arbitrary allpass filter.

Even with  $\mathbf{R}(z)$  being a Laurent polynomial, often  $\mathbf{Q}(z)$  and  $\mathbf{A}(z)$  in (2.5) are Laurent series and may even represent transcendental functions [44, 45]. Due to analyticity, the coefficients of both factors are absolutely convergent. Therefore, both factors can be approximated arbitrarily closely in the least squares sense by Laurent polynomials. In case of the para-Hermitian  $\mathbf{A}(z)$ , such Laurent series is symmetrically truncated from both sides [35] whereas in case of  $\mathbf{Q}(z)$ , a suitable finite order approximation is obtained by masking [34].

### 2.2.2 PEVD Algorithms

PEVD algorithms have witnessed more advancement compared to other polynomial matrix decomposition techniques such as PSVD, PQRD, and others. However, it is important to note that not all PEVD algorithms converge to the decomposition as given in (2.5). As mentioned Chapter 1, PEVD algorithms can be categorized into time and DFT-domain algorithms. However, in practice, all to-date time-domain PEVD algorithms [13, 22, 24, 25, 39, 46] tend to converge to a spectrally majorized solution [13, 36], where the eigenvalues are ordered in a decreasing manner at each frequency point on the unit circle i.e.

$$\lambda_1(e^{j\Omega}) \geq \lambda_2(e^{j\Omega}) \geq \dots \geq \lambda_M(e^{j\Omega}) \quad \forall \Omega \in \mathbb{R}.$$

On the other hand, the DFT-domain methods converge to smooth and analytic decomposition that are not necessarily spectrally majorised [33–35, 47].

## Time Domain PEVD Algorithms

- **Second-order sequential best rotation (SBR2)**

SBR2 is considered as the first PEVD algorithm that diagonalizes a para-Hermitian polynomial matrix in an iterative manner by employing elementary paraunitary operations [48]. This algorithm locates a maximum off-diagonal element and transfer it to the zero-lag slice of the given para-Hermitian polynomial matrix via paraunitary time shift. The zero-lag slice refers to the slice located at  $\tau = 0$  i.e.  $\mathbf{R}[0]$ . Thereafter, this element is eliminated by transferring its energy to diagonal via Givens rotation. This iterative process is repeated until maximum iterations are elapsed or the off-diagonal threshold is satisfied. Due to the paraunitary time-shift matrices, the order of the partially diagonalized and intermediate paraunitary matrix increases with each iteration. To limit the order growth of these matrices for lower order polynomial decomposition, truncation is applied on the outer lags of these matrices [27, 28, 30, 49]. The most notable truncation method is the shift correction truncation that truncates each eigenvector independent of others.

SBR2 and its variant converge towards spectrally majorised solution irrespective of the ground-truth decomposition given in (2.5) [50]. Although, the order of decomposition produced by SBR2 is high, this issue exacerbate if the input  $\mathbf{R}(z)$  is spectrally un-majorised i.e. if eigenvalues intersect. As in this case, SBR2's produced eigenvalues would converge towards non-differential function, the eigenvectors to discontinuous functions. Consequently, the polynomial order of the eigenvalues and eigenvectors is very large [35]. Multiple-shift SBR2 (MS-SBR2) variants transfer multiple elements to the zero-lag slice and hence can provide a slightly faster convergence [25]. Another variant of SBR2 utilizes fast Givens rotations [51, 52] instead of Givens rotation and therefore provide a slight improvement in execution time [46].

- **Sequential Matrix Diagonalization (SMD)**

Instead of shifting a single element, SMD transfers an entire column/row with

a maximum 2- or  $\infty$ -norm to the zero-lag slice of  $\mathbf{R}[\tau]$ . Subsequently, instead of applying a Givens rotation, a full EVD of the zero-lag slice is performed. As a result, each iteration transfers more energy onto the diagonal, leading to faster convergence of the algorithm [22]. If the algorithm employs the  $\infty$ -norm, it is termed as maximum element SMD (ME-SMD); otherwise, it is referred to as SMD. Similar to MS-SBR2, a multiple-shift variant of ME-SMD has been reported in [24], which converges faster than normal SMD but exhibits higher order growth with each iteration. MSME-SMD is slightly improved by reducing the search area to limit this order growth and to reduce the execution time. This variant is known as reduced search MSME-SMD (RS-MSME-SMD) [53]. A Householder transformation based SMD variant reduces the zero-lag slice to a tridiagonal form in each iteration before applying Givens rotation. This variant claims significantly improved convergence speed compared to the original SBR2 algorithm [54]. However, it is worth noting that although this variant requires fewer iterations, each iteration is computationally more expensive compared to the standard SBR2 algorithm.

Indeed, while these two methods are commonly used in PEVD-based broadband sensor array applications, they cannot be considered as the best options because they provide only approximate decompositions. Achieving an accurate decomposition using these methods might require the polynomial order of both eigenvalues and eigenvectors to be excessively large. Especially when the ground-truth eigenvalues intersect/cross at some frequencies, the SBR2 and SMD algorithms will not converge to an analytic solution, and hence the resulting order of eigenvalues and eigenvector will be extremely high. However, if the ground-truth is spectrally majorised, these iterative methods' solution will converge to an analytic solution. However, for this the number of iterations may tend towards infinity, which is impractical. As a result, there is a need for more advanced and efficient algorithms that can offer accurate decomposition without the drawbacks of high polynomial orders and large number of iterations.

## DFT Domain PEVD Algorithm

The first DFT-based PEVD algorithm was proposed by Tohidian et.al. [33] where the EVD is performed in the DFT bins i.e.  $\mathbf{R}(z)|_{z=e^{j\Omega_k}}, k = 1, \dots, K$  and thereafter a smooth decomposition is attempted so that the polynomial order of the decomposition is compact. Initially, the eigenvalues in each bin are sorted in decreasing order. Later, the bin-wise eigenvalues are sorted in each bin by assessing the orthogonality of the eigenvectors of adjacent bins. This sorting mechanism becomes ambiguous if any bin has algebraic multiplicity greater than one. Moreover, the algorithm does not put forward any mechanism to determine a sufficient DFT size to extract the eigenvalues. Apart from this issue, it also requires an a priori estimate of the order of eigenvectors which in practical cases is not known in advance. Additionally, a quadratically constrained quadric programming (QCQP) objective function is optimized via the Powell's dogleg algorithm [55] for phase smoothing of eigenvectors. Given its substantial computational complexity due to the need of the phase smoothing procedure, this approach is less practical for real-world applications.

Another DFT-based eigenvalue extraction method is reported in [35, 56] which does not suffer from the issue of algebraic multiplicity. This algorithm permutes the eigenvalues in each bin such that if a periodic interpolation is performed across the eigenvalues in each, the smoothness is maximum. This method iteratively increases the DFT size to extract the eigenvalues with minimum time-aliasing to ensure that eigenvalues are extracted at sufficient DFT size. In addition, it has proven convergence. The downside of this algorithm is its complexity which in fact grows with the factorial of the spatial dimension of the input para-Hermitian matrix. Once the eigenvalues are extracted, eigenvectors are estimated via optimizing a QCQP problem through the Newton method [34]. In comparison to the method of [33], this method has proven convergence. The DFT domain approach is quite practical for small sizes, but doesn't scale well: the eigenvalue extraction's complexity grows with  $M!$ , the eigenvectors extraction become increasingly complex for para-Hermitian matrices with large temporal dimension which require higher DFT orders.

## 2.3 Polynomial Matrix Singular Value Decomposition

### 2.3.1 Existence of Analytic SVD

For an analytic, non-multiplexed polynomial matrix  $\mathbf{A}(z) \in \mathbb{C}^{M \times L}$ ,  $M \geq L$ , with singular values having even number of spectral zeros on the unit circle, the analytic SVD exists [10] as

$$\mathbf{A}(z) = \mathbf{U}(z)\mathbf{\Sigma}(z)\mathbf{V}^P(z), \quad (2.6)$$

such that  $\mathbf{\Sigma}(z) = \text{diag}\{\sigma_1(z), \dots, \sigma_L(z)\} \in \mathbb{C}^{M \times L}$  contains the analytic singular values and the matrices  $\mathbf{U}(z) \in \mathbb{C}^{M \times M}$ ,  $\mathbf{V}(z) \in \mathbb{C}^{L \times L}$  are paraunitary i.e.  $\mathbf{U}(z)\mathbf{U}^P(z) = \mathbf{I}$ ,  $\mathbf{V}(z)\mathbf{V}^P(z) = \mathbf{I}$ , and contains the left- and right analytic singular vectors, respectively. Unlike singular values of constant matrices, which must be real and positive semi-definite [1], the analytic singular values evaluated on unit circle for  $z = e^{j\Omega}$  must be permitted to take on negative values. This is similarly known for matrices that depend analytically on a continuous, real parameter on some interval [57, 58]. Similar to the analytic EVD factors' finite order polynomial approximation, the analytic SVD factors can similarly be approximated in case any of the three factors on the r.h.s of (2.6) are Laurent series.

If the number of zero crossings of any singular value on the unit circle are odd, analytic SVD of  $\mathbf{A}(z)$  will not exist. Instead, its up-sampled version by 2 i.e.  $\mathbf{A}(z^2)$  would admit analytic SVD [10].

### 2.3.2 PSVD Algorithms

The PSVD of a polynomial matrix was initially computed through two PEVD operations by using either the SBR2 or SMD for PEVD computation [13, 26]. This method of PSVD computation is expensive, involving the calculation of the PEVD for two para-Hermitian matrices, each having an order twice that of the original polynomial matrix. Additionally, the independent estimation of singular vectors results in complex singular values if evaluated on the unit circle, which deviates from the conventional expectation. Instead of depending on PEVD algorithms, as suggested in [12], a subsequent proposal

in the same work advocated computing PSVD through a series of PQRD operations. This approach is claimed to offer improved stability and superior performance when compared to the two PEVDs-based method, as discussed in [43].

The methods mentioned above can be classified as indirect approaches for computing the PSVD of a polynomial matrix. Direct methods, on the other hand, utilize specialized PSVD algorithms, which are again categorized as time-domain or DFT-domain methods. In DFT-domain methods, SVD is performed within DFT bins instead of EVD, and phase-coherence is subsequently established between the bin-wise singular vectors, as explained in [33] and [59]. However, it is worth noting that these methods share the same limitations as DFT-based PEVD approaches. The time-domain methods are, in fact, generalizations of SBR2 and SMD, which are reviewed as follows:

### **Kogbetliantz based PSVD Algorithm**

This PSVD algorithm [11] is a generalisation of the SBR2 algorithm, that extends the application of the latter from para-Hermitian to general matrices, and hence is termed in this thesis as generalized SBR2 (GSBR2). It employs either Givens rotations or the complex Kogbetliantz transformation [60], an extension of the Jacobi transformation to non-symmetric matrices. These techniques are used to transfer the off-diagonal energy onto the diagonal, depending on the location of the maximum off-diagonal element. When the maximum off-diagonal element is located outside the upper  $L \times L$  sub-matrix, a Givens rotation is applied from the left to eliminate it. If the maximum off-diagonal element is within the sub-matrix, the Kogbetliantz transformation is applied. This transformation is a combination of Givens rotation, symmetrization, and Jacobi transformation. While this is a direct method for diagonalization, it has the drawback of slow convergence and higher computational cost compared to the SBR2 algorithm, mainly due to the complexity of the Kogbetliantz transformation. Furthermore, the multiple shift strategy, already applied to SBR2, appears incompatible with the Kogbetliantz transformation due to its combination of Givens rotation, symmetrization, and Jacobi transformation.

## Generalized Sequential Matrix Diagonalization (GSMD)

Similar to how the SBR2 algorithm was extended to GSR2, the SMD algorithm also has a generalized version called the GSMD algorithm [61]. This extension was developed within the scope of this thesis; however, it was not included in the contributions due to space constraints. As a result, only a concise overview of the GSMD algorithm is provided here.

Unlike GSR2, the GSMD algorithm transfers more energy to the diagonal in each iteration. It achieves this by shifting the column with the maximum 2- or  $\infty$ -norm to the zero-lag position,  $\mathbf{A}[0]$ , and then performing a full SVD of the zero-lag matrix instead of employing Kogbetliantz transformation or Givens rotation. This key difference allows GSMD to outperform GSR2 and two PEVD based approaches in terms of convergence and provides lower-order eigenvectors [61]. Just like in the SMD algorithm, the GSMD algorithm can benefit from a multiple-shift strategy to further improve its convergence speed.

## 2.4 Polynomial Matrix QR Decomposition

### 2.4.1 Existence and Uniqueness of the PQRD

For a polynomial matrix  $\mathbf{A}(z) : \mathbb{C} \rightarrow \mathbb{C}^{M \times L}$ , the aim of the PQRD is to achieve a decomposition

$$\mathbf{A}(z) = \underline{\mathbf{Q}}(z)\underline{\mathbf{R}}(z), \quad (2.7)$$

where  $\underline{\mathbf{Q}}(z) \in \mathbb{C}^{M \times M}$  is paraunitary [36], and an upper triangular  $\underline{\mathbf{R}}(z) \in \mathbb{C}^{M \times L}$ . The underscore in notation is to differentiate the paraunitary and the upper-triangular matrix of the QR decomposition from the eigenvector matrix  $\mathbf{Q}(z)$  and para-Hermitian matrix  $\mathbf{R}(z)$ , respectively. To use the  $z$ -domain notation,  $\mathbf{A}(z)$ ,  $\underline{\mathbf{Q}}(z)$ , and  $\underline{\mathbf{R}}(z)$  in (2.7) must be analytic in  $z$  with some region of convergence including  $|z| = 1$ .

Even though proofs exist for the related polynomial eigenvalue [44, 45, 62] and polynomial singular value decompositions [10, 62], there exists no formal proof for an

analytic QR existence. However, [63, 64] have represented these factors as functions of  $z$  with region of convergence, even though these factors may be transcendental functions. Despite this lack, the factorisation (2.7) has been assumed in some already established PQRD algorithms [12, 23, 65] that are discussed below. Analyticity implies infinite differentiability, i.e. smoothness, of  $\mathbf{A}(z)|_{z=e^{j\Omega}}$  and hence motivates the considerations in [63, 64] that  $\underline{\mathbf{Q}}(z)$ , and  $\underline{\mathbf{R}}(z)$  can be Laurent series. If so, then the best polynomial approximation to (2.7) can be achieved by masking and truncation [34], similar to the case of the analytic PEVD factors hinted to in Section 2.2.

Assuming that (2.7) is valid, interesting parallels can be drawn to the ambiguity of the conventional QRD. That is, if an ordinary matrix  $\mathbf{A} \in \mathbb{C}^{M \times L}$  admits  $\mathbf{A} = \underline{\mathbf{Q}}\underline{\mathbf{R}}$ , both unitary  $\underline{\mathbf{Q}} \in \mathbb{C}^{M \times M}$  and upper-triangular  $\underline{\mathbf{R}} \in \mathbb{C}^{M \times L}$  are ambiguous w.r.t. an arbitrary common phase factor  $\underline{\Phi} = \text{diag}\{e^{j\phi_1}, \dots, e^{j\phi_M}\}$  i.e.  $\underline{\mathbf{A}} = \underline{\mathbf{Q}}\underline{\Phi}\underline{\Phi}^H\underline{\mathbf{R}} = \underline{\mathbf{Q}}'\underline{\mathbf{R}}'$ . In other words, the  $m$ th column of the unitary  $\underline{\mathbf{Q}}$  and the  $m$ th row of the upper triangular matrix  $\underline{\mathbf{R}}$  are ambiguous w.r.t a common arbitrary phase shift  $\phi_m$ ,  $m = 1, \dots, M$ . Note that we can equivalently have this ambiguity in (2.7) as

$$\mathbf{A}(z) = \underline{\mathbf{Q}}(z)\underline{\Phi}(z)\underline{\Phi}^P(z)\underline{\mathbf{R}}(z) = \underline{\mathbf{Q}}'(z)\underline{\mathbf{R}}'(z), \quad (2.8)$$

with a paraunitary matrix  $\underline{\Phi}(z) = \text{diag}\{\phi_1(z), \dots, \phi_M(z)\}$  that is diagonal and containing allpass filters  $\phi_m(z)$ ,  $m = 1, \dots, M$ . Hence  $\underline{\mathbf{Q}}'(z) = \underline{\mathbf{Q}}(z)\underline{\Phi}(z)$  and  $\underline{\mathbf{R}}'(z) = \underline{\Phi}^P(z)\underline{\mathbf{R}}(z)$  remain paraunitary and upper right triangular matrices, and hence valid PQRD factors. Unlike for ordinary matrices, the allpass factor  $\underline{\Phi}(z)$  determines the support of the PQRD. For example, if  $\underline{\mathbf{Q}}(z)$  and  $\underline{\mathbf{R}}(z)$  have finite order, then  $\underline{\mathbf{Q}}'(z)$  and  $\underline{\mathbf{R}}'(z)$  would have infinite order unless  $\underline{\Phi}(z)$  takes the form of simple delays. Therefore, the impact of finding a suitable  $\underline{\Phi}(z)$  is crucial for the order and hence for the smoothness and implementation complexity of the QR factors in (2.7).

#### 2.4.2 Existing Iterative PQRD Techniques

To date iterative PQRD algorithms are either based on the concept of SBR2 [13] or SMD [22] algorithms, which are reviewed below.

- **SBR2-based PQRD**

The SBR2 algorithm [13] is an iterative polynomial matrix EVD method that in each operation eliminates the largest off-diagonal component via an elementary paraunitary operation comprising a delay and a Givens rotation. This concept has been extended to operate as a PQRD approach by iteratively applying elementary paraunitary operations until all elements in the lower left-triangular part of a matrix are either sufficiently suppressed, or until the maximum element within that part for the matrix falls below a preset threshold.

There are two reported variants: (i) PQRD by steps (PQRD-BS) [23], and (ii) PQRD by column (PQRD-BC) [12]. The former suppresses one polynomial lower left triangular entry at a time such that all coefficients are driven below the threshold before proceeding to the next polynomial elements. The PQRD-BC variant instead aims to approximately zero a column at a time. This is performed via an iterative search and elimination of the successively largest elements in the selected column. While its convergence is faster than PQRD-BS, the search operation in PQRD-BC is comparatively more expensive [12].

- **SMD-based PQRD**

The idea of the SMD-based PQRD algorithm [19] is to temporally shift as much energy as possible to a particular lag component, where then a full QR decomposition triangularizes that. The SM-PQRD performs this iteratively, until a stopping criterion similar to the SBR2-based PQRD algorithms is satisfied. As a result, SM-PQRD provides faster convergence but each iteration is more expensive than an iteration of the SBR based PQRD algorithms.

- **DFT-based PQRD**

This approach has been developed within the framework of this thesis. It employs the eigenvector phase smoothing procedure introduced by [34]. This algorithm conducts a conventional QR decomposition (QRD) within each DFT bin. Subsequently, it establishes phase coherence among the neighbouring bin-wise QRD factors, leading to the derivation of compact order polynomial QR factors for

a given polynomial matrix [66]. Compared to the SBR2 and SMD type PQRD approach, this technique attains a notably reduced decomposition order. Furthermore, its decomposition accuracy surpasses that of the aforementioned methods by several orders of magnitude. However, it is important to note that its computational complexity grows with both spatial and temporal dimension making it only suitable for lower-order polynomial matrices.

## 2.5 Summary

This chapter has provided the necessary background material for this thesis while also identifying areas in need of improvement. In the context of PEVD algorithms, the time-domain methods reviewed here, such as SBR2 and SMD, have been the subject of recent research focus. Their iterative nature achieves only an approximate diagonalization and results in higher-order decompositions. In addition, these methods are not fully parallelizable whereas for the realization of real-time implementation of broadband sensor array applications like angle of arrival estimation [67] and speech-enhancement [17, 18], parallelization can be considered a possible solution. In contrast, DFT-domain alternatives are parallelizable since each eigenvector is extracted independently. However, their computational complexity remains a significant challenge, making them best suited for lower-order polynomial matrices. A summary of the comparison of the PEVD methods based on the performance metrics considered in this thesis is presented in Table 2.1. Similar improvements are needed in case of both PSVD, whose comparison is shown in Table 2.2 and PQRD for applications like MIMO equalization and decoupling.

The remainder of this thesis relies on certain pertinent background material, which is not specifically detailed within this chapter, related to the estimation theory and standard matrix algebra, both of which are succinctly discussed within their respective contribution chapters.

In light of the limitations of both time and DFT domain polynomial matrix factorization techniques mentioned above, the thesis primarily emphasizes the development of novel methods aimed at substantial computational improvements over state-of-the-art algorithms. Rather than pursuing incremental modifications to established algorithms,

Table 2.1: Performance Comparison of PEVD Methods

<b>Methods</b>	<b>Complexity</b>	<b>Accuracy</b>	<b>Parallelizability</b>	$\mathcal{O}\{Q(z)\}$
SBR2	Low	Very Low	Very Low	Very High
SMD	Medium	Average	Very Low	High
Smooth [33]	High	High	High	Low
Analytic [34, 35]	Very High	Very High	Average	Compact

Table 2.2: Performance Comparison of PSVD Methods

<b>Methods</b>	<b>Complexity</b>	<b>Accuracy</b>	<b>Parallelizability</b>	$\mathcal{O}\{U(z)\}$
GSR2	Very Low	Very Low	Very Low	Very High
2-PEVD [26]	Average	Average	Least	High
PQRD [12]	High	Very Low	Very Low	Very High
Smooth [33]	High	High	High	Low

the emphasis is on introducing innovative solutions. Furthermore, as new methods are developed, the thesis also underscores the importance of ensuring that the proposed method is highly parallelizable. This parallelizability will provide the flexibility to perform either full or partial decomposition, allowing for adaptation to the specific requirements of users or applications.

## Chapter 3

# Space-Time Covariance

# Estimation and Loss of Algebraic

# Multiplicities

### 3.1 Introduction

As covered in the previous chapter, broadband sensor array problems can be formulated using the second-order statistics in form of the space-time covariance matrix  $\mathbf{R}[\tau]$  [13]. In almost all of these broadband scenarios, the space-time covariance matrix cannot be obtained via expectations but must be estimated from finite data. The estimate  $\hat{\mathbf{R}}[\tau]$  will be prone to estimation errors, and the variance of the unbiased estimator based on  $N$  snapshot of data  $\mathbf{x}[n] \in \mathbb{C}^M$ ,  $n = 0, \dots, (N - 1)$  has been investigated in [68]. This deviation from the ground truth  $\mathbf{R}[\tau]$  will in turn result in a perturbation of the eigenvalues and eigenspaces [4, 69, 70]. This perturbation has profound consequences in subspace decomposition applications such as speech enhancement [71], fetal ECG extraction [72], angle of arrival of estimation [73] etc. The impact of estimation errors is twofold. Firstly, an estimation error causes imprecision e.g. through subspace leakage for the above applications. Secondly, e.g. overestimating the support of the space-time covariance matrix will result in polynomial matrices of higher order than necessary [74], counteracting many efforts to keep computational complexity low via e.g. numerical

efficiency [31, 46, 75] or trimming of polynomials [27, 28].

Therefore, the primary aim of this chapter is to enhance the estimate  $\hat{\mathbf{R}}[\tau]$  and thus reduce the perturbation of its eigenvalue decomposition, as well as aid in keeping the polynomial orders of all factors low. This is achieved if the source signals are accessible, such that the convolutive mixing system that contributes to  $\mathbf{x}[n]$  can be estimated via system identification. This type of estimation for  $\hat{\mathbf{R}}[\tau]$  is possible e.g. in loudspeaker-microphone setups such as in [17, 18, 71, 76].

The secondary aim of this chapter is to investigate the perturbation of the eigenvalues and eigenspaces of the estimated space-time covariance matrix from finite data. Previously, it has been linked to the ground-truth space-time covariance, the sample size  $N$  and the distance between the eigenvalues evaluated on isolated point  $\Omega_0$  [68, 70]. However, the impact of the estimation process on the overall factors of  $\hat{\mathbf{R}}(e^{j\Omega})$  for continuous  $\Omega$  has not been previously investigated. The prime focus of this analysis is the effects on the non-trivial algebraic multiplicities of the eigenvalues that are present if at least some eigenvalue intersect on the unit circle. The loss of such algebraic multiplicities, and therefore the loss of such intersections, is a fundamental challenge for space-time covariance matrix estimation.

Below, Section 3.2 provides a brief recap of the unbiased estimator and the perturbation of the eigenvalues from [68, 70, 74]. Section 3.3 explains the system identification approach along with numerical examples and ensemble tests whereas Section 3.4 investigates the loss of algebraic multiplicities in the estimation process.

## 3.2 Estimation and Its Perturbation Effects

Prior to presenting the novel system identification based estimation of space-time covariance estimation, we offer a concise summary of the background material on the unbiased estimator and the impact of estimation errors on the perturbation of eigenvalues.

### 3.2.1 Perturbation of Eigenvalues

The effects of estimation errors have been investigated in [68, 70] by deriving the bounds for the cumulative square difference between the eigenvalues  $\lambda_m(z)$  and  $\hat{\lambda}_m(z)$  of  $\mathbf{R}(z)$  and  $\hat{\mathbf{R}}(z)$ , respectively, on the unit circle. Thus, if evaluated for  $z = e^{j\Omega_0}$ , through the Hoffman-Wielandt theorem [69], the inequality

$$\sum_{m=1}^M \left( \hat{\lambda}_m(e^{j\Omega_0}) - \lambda_m(e^{j\Omega_0}) \right)^2 \leq \|\mathbf{E}(e^{j\Omega_0})\|_F^2, \quad (3.1)$$

can be established, where  $\mathbf{E}(e^{j\Omega_0}) = \mathbf{R}(e^{j\Omega_0}) - \hat{\mathbf{R}}(e^{j\Omega_0})$ . This shows that the bin-wise perturbation of the eigenvalues is directly related to the estimation error. A similar relation has been established for the eigenspaces in [70]. Hence, in order to limit the perturbation, the estimation error has to be kept as small as possible.

### 3.2.2 Unbiased Estimator

With assumptions made in Chapter 2 for the source model in Fig. 2.1, the ground-truth space-time covariance matrix  $\mathbf{R}[\tau]$  will have time-domain support of  $2\tau_{\max} + 1$  i.e.  $\mathbf{R}[\tau] = 0$  for  $|\tau| > \tau_{\max}$ . Each individual element of  $\mathbf{R}[\tau]$ , being a cross-correlation sequence, can be computed as

$$\begin{aligned} r_{\ell,m}[\tau] &= \mathcal{E}\{x_\ell[n]x_m^*[n-\tau]\} \\ &= \sum_n \sum_{k=1}^L a_{\ell,k}[n]a_{m,k}^*[n-\tau] + \sigma_v^2 \delta[\tau] \delta[l-m], \quad m, l = 1, \dots, M \end{aligned} \quad (3.2)$$

These elements of a space-time covariance matrix are often estimated directly from the finite data received at a sensor array. For  $N$  measurements i.e.  $\mathbf{x}[n], n = 0, \dots, (N-1)$ , the unbiased estimator for (3.2) is defined as [68]

$$\hat{r}_{\ell,m}[\tau] = \begin{cases} \frac{1}{N-|\tau|} \sum_{n=0}^{N-|\tau|-1} x_\ell[n+\tau]x_m^*[n], & \tau \geq 0, \\ \frac{1}{N-|\tau|} \sum_{n=0}^{N-|\tau|-1} x_\ell[n]x_m^*[n-\tau], & \tau < 0. \end{cases} \quad (3.3)$$

The variance of the estimator, derived in [68] under assumptions made in Chapter 2, is

$$\begin{aligned} \text{var}\{\hat{r}_{\ell,m}[\tau]\} &= \frac{1}{(N - |\tau|)^2} \sum_{t=-N+|\tau|+1}^{N-|\tau|-1} (N - |\tau| - |t|) \\ &\quad \cdot (r_{\ell,\ell}[t]r_{m,m}^*[t] + \bar{r}_{\ell,m}[\tau + t]\bar{r}_{\ell,m}^*[\tau - t]), \end{aligned} \quad (3.4)$$

where  $\bar{r}_{\ell,m}[\tau] = \mathcal{E}\{x_\ell[n]x_m[n - \tau]\}$  refers to complementary cross-correlation [74]. It shows that the estimator variance depends upon  $\mathbf{R}[\tau]$  and the sample size  $N$ . The optimum support for the estimated space-time covariance is obtained by minimizing the overall estimation error

$$\tau_{\text{opt}} = \underset{\tau_{\text{est}}}{\text{argmin}} \zeta_{\text{UE}}, \quad (3.5)$$

where

$$\zeta_{\text{UE}} = \mathcal{E} \left\{ \sum_{\tau} \|\mathbf{E}[\tau]\|_{\text{F}}^2 \right\} = \sum_{-\tau_{\text{est}}}^{\tau_{\text{est}}} \mathcal{E} \{ \|\mathbf{E}[\tau]\|_{\text{F}}^2 \} + 2 \sum_{\tau_{\text{est}}+1}^{\tau_{\text{max}}} \|\mathbf{R}[\tau]\|_{\text{F}}^2 \quad (3.6)$$

with  $\mathbf{E}[\tau] = \mathbf{R}[\tau] - \hat{\mathbf{R}}[\tau]$ ,  $2\tau_{\text{est}} + 1$  is the support of  $\hat{\mathbf{R}}[\tau]$ . The subscript UE in  $\zeta_{\text{UE}}$  stands for "unbiased estimator". The first summation term is an estimation error while the second term represents the truncation part. It must be noted that for  $\tau_{\text{est}} > \tau_{\text{max}}$ , the truncation part would be zero. The optimum support for estimation leads to the balance between these two error components which makes up the total estimation error. It can be observed that the truncation part depends on the ground-truth space-time covariance matrix  $\mathbf{R}[\tau]$  and the support over which the estimation is made, i.e.,  $\tau_{\text{est}}$ , while the estimation error depends on the sample size, the ground-truth space-time covariance, and the estimation support  $\tau_{\text{est}}$  [68, 74].

### 3.3 Space-Time Covariance Estimation via System Identification

#### 3.3.1 Source Model and Space-Time Covariance Matrix

With the source covariance  $\mathcal{E}\{\mathbf{s}[n]\mathbf{s}^H[n-\tau]\} = \mathbf{I}_L\delta[\tau]$  and the noise covariance matrix  $\mathcal{E}\{\mathbf{v}[n]\mathbf{v}^H[n-\tau]\} = \sigma_v^2\mathbf{I}_M\delta[\tau]$ , the space time covariance  $\mathbf{R}[\tau] = \mathcal{E}\{\mathbf{x}[n]\mathbf{x}^H[n-\tau]\} \in \mathbb{C}^{M \times M}$  can be tied to the source model of Fig. 2.1 through  $\mathbf{A}[n]$  as

$$\mathbf{R}[\tau] = \sum_n \mathbf{A}[n]\mathbf{A}^H[n-\tau] + \sigma_v^2\mathbf{I}_M\delta[\tau], \quad (3.7)$$

where the individual elements of  $\mathbf{R}[\tau]$ , the cross-correlation sequences, is defined in (3.2).

#### 3.3.2 Estimation via System Identification

In the case source signals are accessible and can be controlled, system identification can be performed to estimate  $\mathbf{A}[n]$  with finite time-domain support  $L_A$ . With a system matrix estimate  $\hat{\mathbf{A}}[n]$ , the estimate of  $\mathbf{R}[\tau]$  can be obtained directly from (3.7) as

$$\hat{\mathbf{R}}[\tau] = \sum_n \hat{\mathbf{A}}[n]\hat{\mathbf{A}}^H[n-\tau] + \hat{\sigma}_v^2\mathbf{I}_M\delta[\tau], \quad (3.8)$$

where  $\hat{\sigma}_v^2$  variance estimate of the additive noise in Fig. 2.1 which can be obtained by finding the minimum mean square error as explained further below. It can be seen that the system identification method requires more knowledge and control over the source model and therefore, it is expected to produce better space-time covariance estimate than the unbiased estimator which relies only on the measurement vector.

The entire procedure for system identification, noise variance estimate and the optimum support estimate is described below.

## Adaptive Filter Theory

To perform the system identification, a variety of approaches can be utilized such as the least mean square and recursive least squares algorithms [2]. For lower minimum mean squared error, the Wiener solution [2] can be utilized to identify the channel responses between  $L$  sources and  $M$  sensors. The  $M$  separate  $L$ -channel adaptive filter problem can be formulated using

$$\hat{x}_m[n] = \sum_{\ell=1}^L \hat{\mathbf{a}}_{m,\ell}^H \mathbf{s}_\ell[n] = \hat{\mathbf{w}}_m^H \mathbf{y}[n], \quad m = 1, \dots, M \quad (3.9)$$

where  $\hat{\mathbf{a}}_{m,\ell} = [\hat{a}_{m,\ell}^*[0], \dots, \hat{a}_{m,\ell}^*[L_f-1]]^T$ ,  $\mathbf{s}_\ell[n] = [s_\ell[n], \dots, s_\ell[n-L_f+1]]^T$  is a tap-delay line with  $L_f$  filter coefficients,  $\hat{\mathbf{w}}_m = [\hat{\mathbf{a}}_{m,1}, \dots, \hat{\mathbf{a}}_{m,L}]^T$  and  $\mathbf{y}[n] = [\mathbf{s}_1[n], \dots, \mathbf{s}_L[n]]^T$ , to obtain the optimal filter coefficients by solving

$$\hat{\mathbf{w}}_{m,\text{opt}} = \arg \min_{\hat{\mathbf{w}}_m} \mathcal{E} \{ |\mathbf{x}_m[n] - \hat{\mathbf{x}}_m[n]|^2 \} . \quad (3.10)$$

This is solved using the Wiener Hopf solution as

$$\hat{\mathbf{w}}_{m,\text{opt}} = \hat{\mathbf{R}}^{-1} \hat{\mathbf{p}}_m , \quad (3.11)$$

where  $\hat{\mathbf{R}}$ , the sample covariance matrix, and  $\hat{\mathbf{p}}_m$ , the correlation vector, estimate the quantities  $\mathcal{E} \{ \mathbf{y}[n] \mathbf{y}^H[n] \}$  and  $\mathcal{E} \{ \mathbf{y}[n] x_m[n] \}$ , respectively, from  $N$  instances. The l.h.s of (3.11) i.e.  $\hat{\mathbf{w}}_{m,\text{opt}}$  is the minimum mean squared error estimate of the conjugate of the coefficients in the  $m$ th row of  $\mathbf{A}[n]$ .

## Noise Variance Estimate

In the ideal case where  $\hat{\mathbf{w}}_{m,\text{opt}}$  accurately reflects the appropriate coefficients of  $\mathbf{A}[n]$ , the variance estimate  $\hat{\sigma}_v^2$  is equivalent to the minimum mean square error [2],

$$\hat{\sigma}_{v,m}^2 = \hat{\sigma}_{x_m}^2 - \hat{\mathbf{p}}_m^H \hat{\mathbf{R}}^{-1} \hat{\mathbf{p}}_m , \quad (3.12)$$

where  $\hat{\sigma}_{x_m}^2$  is the power estimated over the  $N$  samples of  $x_m[n]$ . Due to the  $M$  separate multichannel identification,  $\hat{\sigma}_v^2$  can be assumed to be the average of all  $M$  individual noise power estimates, and so  $\hat{\sigma}_v^2 = \frac{1}{M} \sum_m \hat{\sigma}_{v,m}^2$ .

### Optimum Filter Length and Lag Support

Once the channel matrix and noise variance estimates have been obtained,  $\mathbf{R}[\tau]$  can be estimated using (3.8). Similar to the unbiased estimator, the error  $\zeta_{\text{SI}}$ , where SI subscript stands for system identification, defined akin to  $\zeta_{\text{UE}}$  in (3.6) as

$$\zeta_{\text{SI}} = \mathcal{E} \left\{ \sum_{\tau} \|\mathbf{E}[\tau]\|_{\text{F}}^2 \right\} = 2 \sum_{L_f}^{L_A-1} \|\mathbf{R}[\tau]\|_{\text{F}}^2 + \sum_{-L_f+1}^{L_f-1} \mathcal{E} \{ \|\mathbf{E}[\tau]\|_{\text{F}}^2 \} , \quad (3.13)$$

where  $\mathbf{E}[\tau] = \mathbf{R}[\tau] - \hat{\mathbf{R}}[\tau]$  with  $\mathbf{R}[\tau]$  defined in (3.7) and  $\hat{\mathbf{R}}[\tau]$  in (3.8). Similar to  $\zeta_{\text{UE}}$ , two terms contribute to  $\zeta_{\text{SI}}$  (i) a truncation term in case the adaptive filter length  $L_f$  falls short of the ground truth system  $\mathbf{A}[n]$  length, denoted with  $L_A$ ; and (ii) an estimation or perturbation term that impacts on the coefficients of  $\hat{\mathbf{w}}_{m,\text{opt}}$  in (3.11), which grows with the number of coefficients. Hence, the optimal length  $L_{f,\text{opt}}$  is where these two error terms result in least overall error similar to (3.5). To illustrate the impact of each error term with respect to the adaptive filter, a numerical example is provided below.

**Example 1.** An experiment over an ensemble consisting of 300 instances of a channel matrix  $\mathbf{A}[n] \in \mathbb{C}^{2 \times 2}$  with  $L_A = 30$  is presented where the average SNR at the sensors, defined as

$$\text{SNR} = \frac{\sum_n \|\mathbf{A}[n]\|_{\text{F}}^2}{M\sigma_v^2} , \quad (3.14)$$

is set to 20 dB via  $\sigma_v^2$ . System identification is performed for every instance in the ensemble with adaptive filter length  $L_f$  being varied from 20 to 40 over different sample sizes i.e.  $N$ . In order to measure the error for an ensemble, a metric is defined as

$$\zeta_{\text{norm}} = \frac{\sum_{\tau} \|\mathbf{R}[\tau] - \hat{\mathbf{R}}[\tau]\|_{\text{F}}^2}{\sum_{\tau} \|\mathbf{R}[\tau]\|_{\text{F}}^2} . \quad (3.15)$$

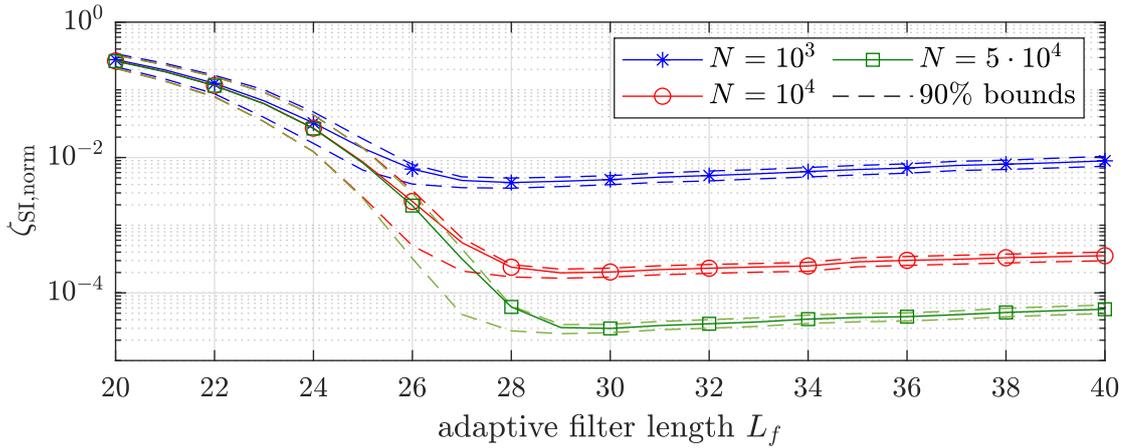


Figure 3.1: Ensemble results for  $\zeta_{\text{norm}}$  when obtaining  $\hat{\mathbf{R}}[\tau]$  in dependence of adaptive filter length,  $L_f$ .

The numerator of this metric relates to the bin-wise perturbation bound on the eigenvalues in (3.1) via Parseval’s theorem [77]. Normalizing it by the Frobenius norm of the ground-truth ensures that the metric can be applied to extract ensemble results for different instances of  $\mathbf{R}[\tau]$ . This metric is depicted in Fig. 3.1 for the ensemble test, illustrating the above trade-off mentioned earlier. For low values of  $L_f$ , the truncation error dominates, while at higher values of  $L_f$ , the error increases due to the noisy coefficients. In addition, the ensemble optimum filter length  $L_{f,\text{opt}}$  depends on the filter length. In Fig. 3.1, note that  $L_{f,\text{opt}}$  is 28, 29 and 30 for  $N = 1e3$ ,  $1e4$  and  $5e4$  respectively. The filter length, for which the minimum is reached, therefore converges towards the ground truth support  $L_A$ .  $\triangle$

### 3.3.3 Simulations and Comparison

This subsection provides a comparative analysis of system identification with the unbiased estimator across different SNRs and sample sizes ( $N$ ). To facilitate the comparison, the normalized error metric defined in (3.15) is employed

## Scenario and Parameters

To compare both methods, an ensemble of 500 random instances of  $\mathbf{R}[\tau] \in \mathbb{C}^{2 \times 2}$  with moderately large support  $L_A = 30$  is employed. The estimates are made over various sample sizes  $N$  ranging from  $10^3$  to  $10^6$  and noise levels of 10 dB and 20 dB SNR according to (3.14). The optimal lag support for the unbiased estimator is selected on the basis of the lowest value of  $\zeta$  by varying the lag support between 1 and 29 because  $\tau_{\text{opt}} < \tau_{\text{max}} = 30$  [68]. In contrast, the adaptive filter length for the system identification is set equal to 30 from the experiment of Example 1.

## Ensemble Results

Fig. 3.2 shows the ensemble results for the experiment. For each case, curves for 10 dB and 20 dB SNR are shown, together with the bounds within which 90% of the ensemble results fall.

It can be observed that the unbiased estimator, which treats measurement noise as part of the data, is independent of the SNR. In contrast, the noise terms acts as observation noise for the system identification approach, which therefore yields increased accuracy as the SNR grows. All curves converge with approximately  $1/N$ , but the system identification approach generally is capable of reaching better accuracy than the unbiased estimator. This is due to the additional information that in this case is known for the system — the source signals  $s_\ell[n]$ . In contrast, for lower SNR, the system identification performance will drop below that of the unbiased estimator, as the known signals  $s_\ell[n]$  will be dwarfed by the unknown observation noise  $v_m[n]$  which then start to dominate.

It is important to note that for small values of  $N$ , both the sources and noise signals generated in MATLAB<sup>©</sup> are usually not significantly random, that is, they may exhibit mutual and temporal correlations. Therefore, the considered assumptions do not hold true, and this effect can be seen in the insignificant difference between 10 and 20 dB cases for sample size  $N < 10^4$ .

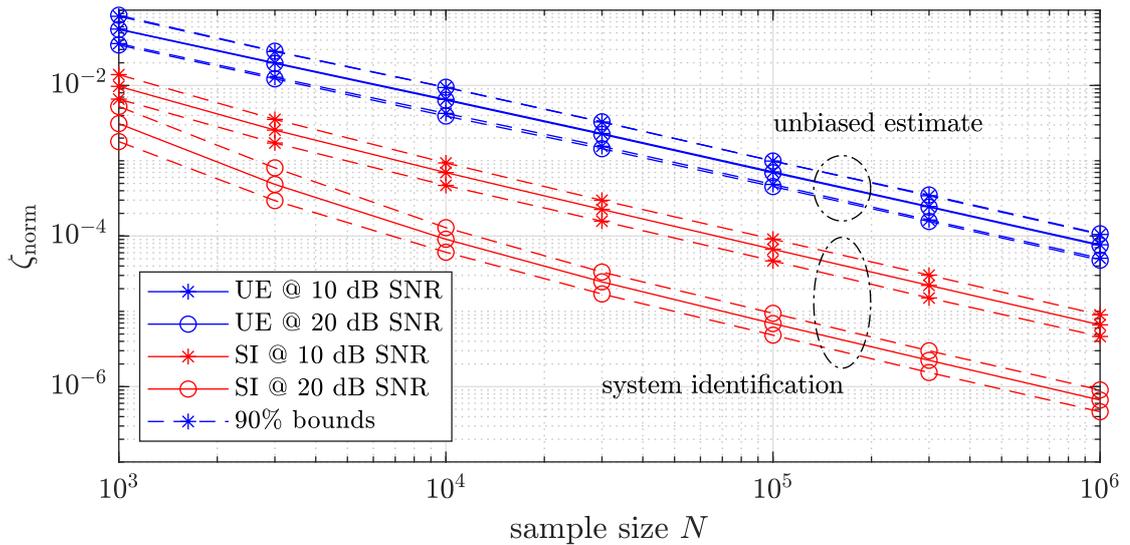


Figure 3.2: Comparison of estimation methods via an ensemble of  $\mathbf{R}[\tau] \in \mathbb{C}^{2 \times 2}$ , showing the measured error via the unbiased estimator and the system identification approach.

### 3.3.4 Conclusion

This section compared the unbiased estimator with the proposed system identification approach for the estimation of a space time covariance matrix. Unlike the unbiased estimator, the latter can only be exploited in case the source signals are known. This method consists of the identification of the convolutive mixing system by a Wiener filter approach, and the estimation of the additive noise power via the minimum mean square error of the Wiener filter. An ensemble experiment carried out at various noise levels demonstrates that the system identification approach performs significantly better than the unbiased estimator for reasonable to high SNRs. This is important, as the enhanced accuracy results in a lower bin-wise perturbation of the eigenvalue decomposition of this matrix, which is key to formulating and solving a number of relevant broadband array problems.

## 3.4 Eigenvalues of an Estimated Space-Time Covariance Matrix

### 3.4.1 Eigenvalues at an Algebraic Multiplicity

As outlined in Sec. 3.2, the variance of the unbiased estimator depends on both the ground truth  $\mathbf{R}[\tau]$  and the sample size  $N$ . Thus, the eigenvalues of  $\hat{\mathbf{R}}[\tau]$  are perturbed, and now are random variables [68]; this is well-known from random matrix theory, see e.g. [78–80]. This is particularly noticeable where the eigenvalues of  $\mathbf{R}(z)$  possess an algebraic multiplicity greater than one, i.e. where at least two eigenvalues are identical. When now inspecting the eigenvalues of  $\hat{\mathbf{R}}(z)$  instead, we find that these eigenvalues are drawn from probability distributions, and that we thus obtain distinct eigenvalues with probability one, unless for the sample size we have  $N \rightarrow \infty$ .

**Example 2.** Consider the parahermitian matrix

$$\mathbf{R}(z) = \begin{bmatrix} \frac{1-j}{2}z + 3 + \frac{1+j}{2}z^{-1} & \frac{1+j}{2}z^2 + \frac{1-j}{2} \\ \frac{1+j}{2} + \frac{1-j}{2}z^{-2} & \frac{1-j}{2}z + 3 + \frac{1+j}{2}z^{-1} \end{bmatrix} \quad (3.16)$$

from [44]. This matrix possesses the analytic eigenvalues  $\lambda_1(z) = z + 3 + z^{-1}$ , and  $\lambda_2(z) = jz + 3 - jz^{-1}$ , which are shown, evaluated on the unit circle, in Fig. 3.3(a). The analytic eigenvectors can be selected as  $\mathbf{q}_m(z) = [1, \pm z^{-1}]^T / \sqrt{2}$ ,  $m = 1, 2$ . The evolution of the eigenvectors along the unit circle is visualised in Fig. 3.3(b) via the Hermitian angle  $\alpha_m(\Omega)$ , with  $\cos \alpha_m(\Omega) = |\mathbf{q}_1^H(e^{j\Omega})\mathbf{q}_m(e^{j\Omega})|$ , whereby the DC value for the first eigenvalue,  $\mathbf{q}_1^H(e^{j0})$ , is chosen as an arbitrary reference point. Note that due to analyticity, both eigenvalues and the angles of the eigenvectors evolve smoothly.

Based on this para-Hermitian matrix, the source model in Fig. 2.1 with  $\mathbf{A}(z) = \mathbf{Q}(z)\text{diag}\{\sqrt{\lambda_1(z)}, \sqrt{\lambda_2(z)}\}$  to generate an ensemble of  $10^5$  data sequences with various sample sizes  $N$ , from which the distributions of the eigenvalues at  $\Omega = \frac{\pi}{4}$  are estimated. Note that at  $\Omega = \frac{\pi}{4}$ , there exists a non-trivial multiplicity i.e.  $\lambda_1(e^{j\pi/4}) = \lambda_2(e^{j\pi/4})$ , as evident in Fig. 3.3(a). The approximated distributions for the two eigenvalues are shown in Fig. 3.4; note that only for the transition  $N \rightarrow \infty$  will one have a distribution

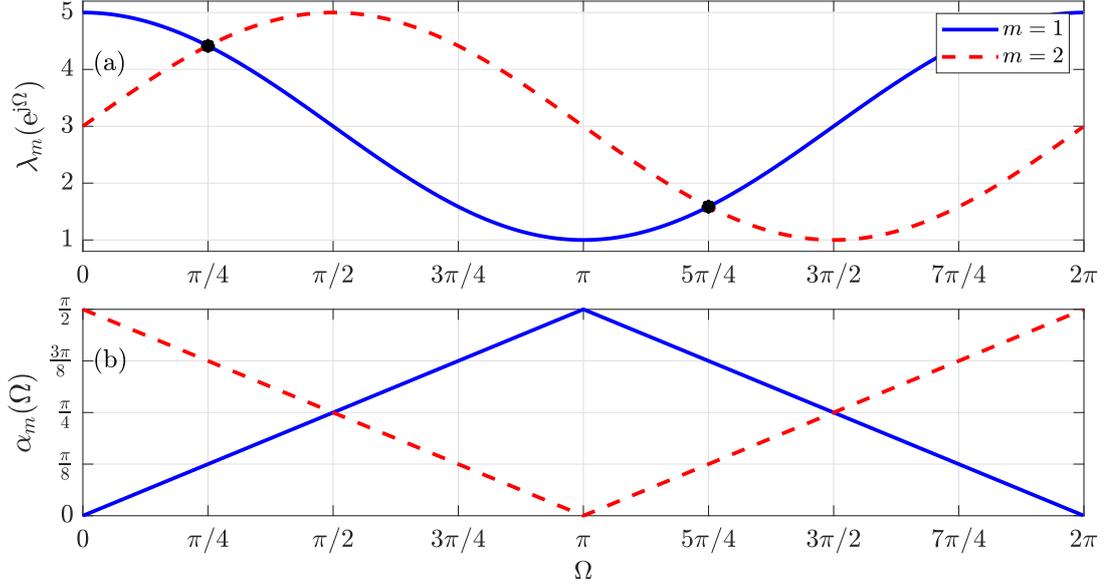


Figure 3.3: Example for (a) analytic eigenvalues and (b) Hermitian angles of their corresponding analytic eigenvectors.

that guarantees two identical eigenvalues, i.e. an algebraic multiplicity of two.  $\triangle$

### 3.4.2 Impact on Analytic Eigenvalues

As it has been argued, at a given frequency  $\Omega_0$  with an algebraic multiplicity of the eigenvalues of  $\mathbf{R}(e^{j\Omega_0})$  greater than one, the eigenvalues  $\hat{\lambda}(e^{j\Omega_0})$  of  $\hat{\mathbf{R}}(e^{j\Omega_0})$  must be distinct with probability one. Since the eigenvalues  $\hat{\lambda}(e^{j\Omega})$  are random variables for all  $\Omega$ ,  $\hat{\mathbf{R}}(e^{j\Omega})$  has distinct eigenvalues with probability one for all frequencies  $\Omega$ .

Since  $\hat{\mathbf{R}}(z)$  is analytic, e.g. because it is estimated with only finite support  $|\tau| \leq \tau_{\max}$ , its eigenvalues  $\hat{\lambda}_m(z)$ ,  $m = 1, \dots, M$  must also be analytic. However, since the eigenvalues are distinct at all frequencies, if ordered in descending values, they must now be strictly spectrally majorised, such that on the unit circle

$$\hat{\lambda}_m(e^{j\Omega}) > \hat{\lambda}_{m+1}(e^{j\Omega}) \quad \forall \Omega, m = 1, \dots, (M-1). \quad (3.17)$$

Spectral majorisation has been a feature of two families of polynomial EVD algorithms [13, 20, 22, 50], but here is not an algorithmic detail but expresses the nature of the estimated space-time covariance matrix.

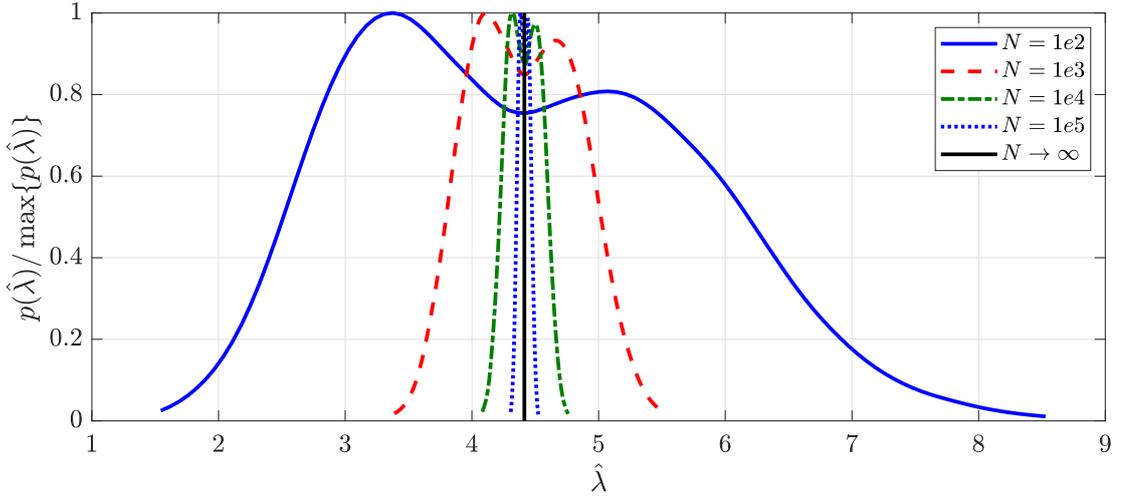


Figure 3.4: Normalised approximate probability density functions  $p(\hat{\lambda})$  for eigenvalues  $\hat{\lambda}$  of  $\hat{\mathbf{R}}(e^{j\pi/4})$ , estimated for a number of different sample sizes  $N$ , from each  $10^5$  instances.

### 3.4.3 Impact of Sample Size

It is interesting to note that the loss of algebraic multiplicities or the strict spectral majorisation of eigenvalues cannot be alleviated by enhancing estimates. This includes, for example, limiting the perturbation of eigenvalues through optimum support estimation [74]. Bypassing some estimation errors through performing a system identification of the source model, as already explained in Sec. 3.3 and [38], generally still retains some finite error, for example due to observation noise. Simply increasing the sample size  $N$  on which the estimate is based will not bypass this challenge unless the transition  $N \rightarrow \infty$  is made [68].

A detrimental effect occurs for the analytic EVD as  $N$  increases. Let  $\lambda'_m(e^{j\Omega})$  and  $\mathbf{q}'_m(e^{j\Omega})$  be permuted versions of the EVD factors  $\lambda_m(e^{j\Omega})$  and  $\mathbf{q}_m(e^{j\Omega})$  of  $\mathbf{R}(e^{j\Omega})$ , such that the modified eigenvalues  $\lambda'_m(e^{j\Omega})$  are spectrally majorised,

$$\lambda'_m(e^{j\Omega}) \geq \lambda'_{m+1}(e^{j\Omega}) \quad \forall \Omega, m = 1, \dots, (M-1). \quad (3.18)$$

If the analytic eigenvalues  $\lambda_m(e^{j\Omega})$  are not spectrally majorised, then  $\lambda'_m(e^{j\Omega})$  will only be piece-wise analytic: at frequencies where permutations occur, they will be continuous but not infinitely differentiable. Further the corresponding eigenvectors  $\mathbf{q}'_m(e^{j\Omega})$  will

be discontinuous at permutation frequencies [44]. Thus, as  $N$  increases, we find that

$$\hat{\lambda}_m(e^{j\Omega}) \longrightarrow \lambda'_m(e^{j\Omega}). \quad (3.19)$$

Therefore, with increasing sample size  $N$ ,  $\lambda'_m(e^{j\Omega})$  tends towards a function that is not infinitely differentiable. The eigenvectors of  $\hat{\mathbf{R}}(z)$ ,  $\mathbf{q}'_m(e^{j\Omega})$ , converge towards discontinuous functions. With increasing  $N$ , both eigenvalues and eigenvectors remain analytic but become more and more difficult to approximate by polynomials or Laurent polynomials [44], requiring them to be of higher orders than for a lower value of  $N$ . Lets consider an example that further explains this issue.

**Example 3.** In this example, the setup of Example 2 is further investigated such that the inspection of analytic eigenvalues  $\hat{\lambda}_m(e^{j\Omega})$  and eigenvectors  $\hat{\mathbf{q}}_m(e^{j\Omega})$  across the range  $\Omega = (0; 2\pi)$  can be made. These are extracted by taking EVDs within individual bins obtained with a DFT of  $\hat{\mathbf{R}}[\tau]$  of sufficient length. Due to (3.17), the eigenvalues are straightforward to associate across the DFT bins [35]. For the eigenvectors, EVDs in individual frequency bins will not be phase-aligned [34, 47]; this however does not affect the subspaces in which these analytic eigenvectors exist [34], and the Hermitian angle evaluated in Example 2 will measure the smoothness of these subspaces.

Fig. 3.5 shows the case of  $\hat{\mathbf{R}}[\tau]$  estimated with a sample size  $N = 10^2$  via (3.3). Due to this small size, the estimation error can be significant, particularly if the support of  $\mathbf{R}[\tau]$  is overestimated [68]. Here and in the following examples, the support is optimised to yield the smallest possible estimation error [74]. Nonetheless, the eigenvalues and eigenspaces are perturbed and significantly deviate from the eigenvalues and eigenvector angles of the ground truth space-time covariance  $\mathbf{R}[\tau]$ . For  $N = 10^4$  in Fig. 3.6, the eigenvalues  $\hat{\lambda}_m(e^{j\Omega})$  are strictly spectrally majorised according to (3.17) and now follow  $\lambda_m(e^{j\Omega})$  closely on a bin-wise basis. However, permutations w.r.t.  $\lambda_m(e^{j\Omega})$  occur at  $\Omega = \frac{\pi}{4}$  and  $\Omega = \frac{5\pi}{4}$ . The angles  $\alpha_m(\Omega)$  of the associated eigenvectors  $\hat{\mathbf{q}}(e^{j\Omega})$  closely follow those of  $\mathbf{q}_m(e^{j\Omega})$  on a bin-wise basis, but are also permuted at  $\Omega = \frac{\pi}{4}$  and  $\Omega = \frac{5\pi}{4}$ . Since  $\mathbf{q}'_m(e^{j\Omega})$  would be discontinuous at those points, but  $\hat{\mathbf{q}}(e^{j\Omega})$  has to be analytic, some sharp transitions occur around the permutation frequencies.

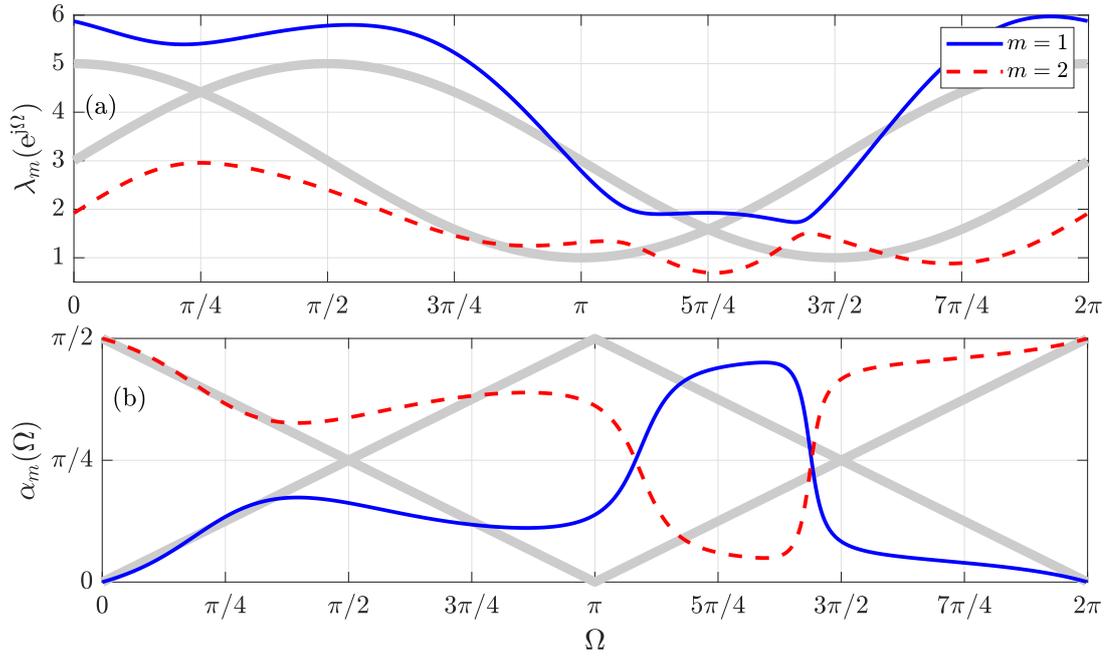


Figure 3.5: (a) eigenvalues  $\hat{\lambda}_m(e^{j\Omega})$  for  $N = 100$  (coloured curves) and ground truth  $\lambda_m(e^{j\Omega})$  (in grey, underlaid); Hermitian angles  $\alpha_m(\Omega)$  for the corresponding eigenvectors  $\hat{\mathbf{q}}(e^{j\Omega})$  and  $\mathbf{q}(e^{j\Omega})$ .

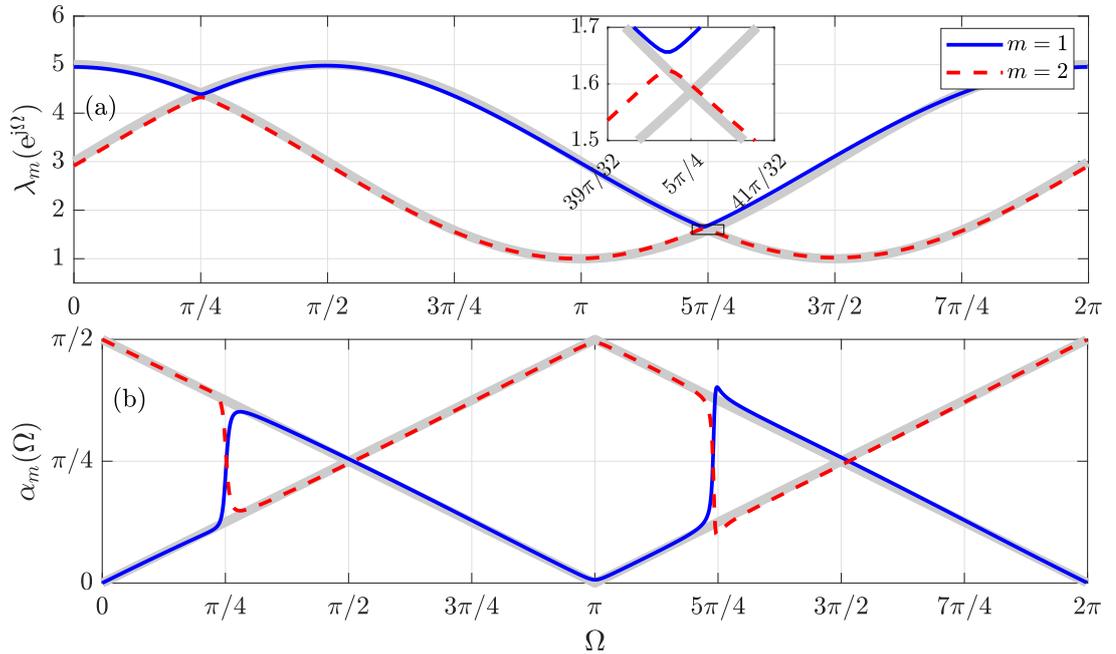


Figure 3.6: (a) eigenvalues  $\hat{\lambda}_m(e^{j\Omega})$  for  $N = 10^4$  (coloured curves) and ground truth  $\lambda_m(e^{j\Omega})$  (in grey, underlaid); Hermitian angles  $\alpha_m(\Omega)$  for the corresponding eigenvectors  $\hat{\mathbf{q}}(e^{j\Omega})$  and  $\mathbf{q}(e^{j\Omega})$ .

The results for a further increase to  $N = 10^6$  are shown in Fig. 3.7. The approximation of a discontinuity of the Hermitian angles  $\alpha_m(e^{j\Omega})$  in Fig. 3.7(b) indicates that the eigenvalues  $\hat{\lambda}_m(e^{j\Omega})$  in Fig. 3.7(a) remain strictly spectrally majorised. Compared to Fig. 3.6(b), the transition at the permutation frequencies  $\Omega = \frac{\pi}{4}$  and  $\Omega = \frac{5\pi}{4}$  is now sharpened, and show behaviour similar to Gibbs phenomena when approximating discontinuities. As a consequence, the eigenvalues  $\hat{\mathbf{q}}_m(z)$  need a higher approximation order or than those obtainable for a smaller sample size  $N$ . Although, the example is founded on the estimate of the unbiased estimator, the same is true for a system identification estimate which causes the loss of algebraic multiplicities as shown in Fig. 3.8. The system identification is performed at 20 dB SNR from  $N = 10^6$  snapshots. While it's clear that the perturbation in the system identification case is lower compared to the unbiased estimator case, it's important to note that both cases result in the loss of multiplicities.  $\triangle$

### 3.4.4 Impact on Applications

The strict spectral majorisation of eigenvalues of an estimated space-time covariance matrix can have both positive and negative consequences, which this subsection briefly highlights.

#### Subspace Methods

For subspace-based methods such as the polynomial multiple signal classification (P-MUSIC) approach [14, 67, 76] or transient signal detection in the noise-only subspace [81–83], an accurate estimation of the signal-plus-noise and noise-only subspaces is required. The effect caused by permutations of the ground truth analytic EVD factors causes an increase in the approximation orders for the eigenvectors, and hence for computational complexity that the paraunitary matrices incur when implemented.

Additionally, since the permutations at algebraic multiplicities greater than one of the eigenvalues of  $\mathbf{R}(z)$  cause switching between subspaces in  $\hat{\mathbf{R}}(z)$ , the switching itself and the associated Gibbs-type phenomena that could be observed in Example 3 — see Fig. 3.7(b) — may cause challenges when performing projections.

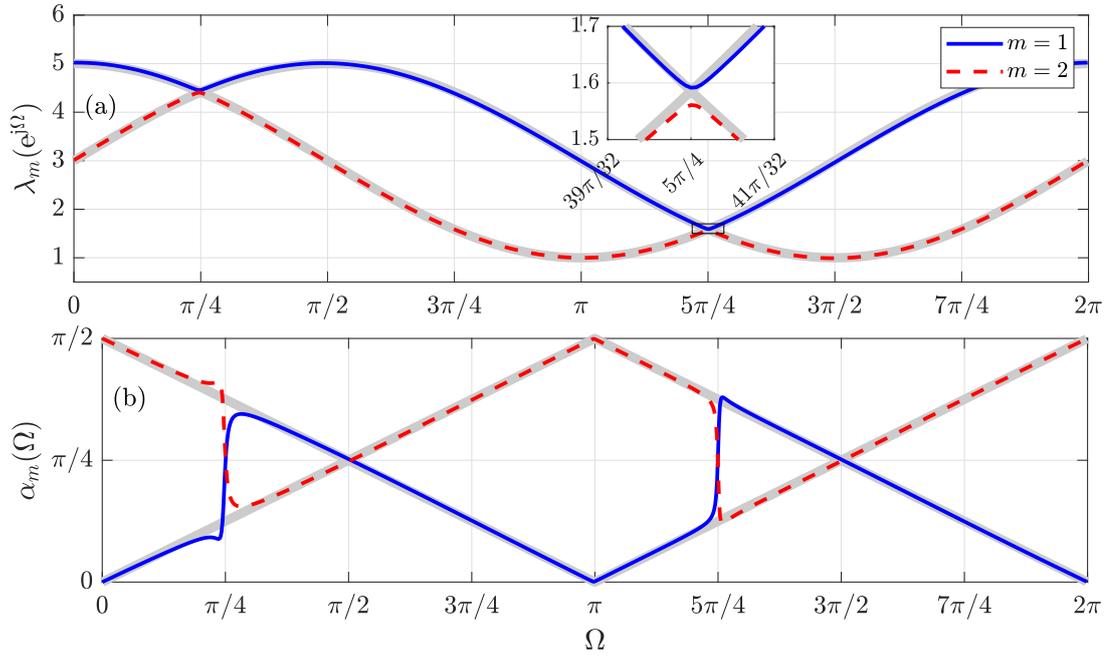


Figure 3.7: (a) eigenvalues  $\hat{\lambda}_m(e^{j\Omega})$  for  $N = 10^6$  (coloured curves) and ground truth  $\lambda_m(e^{j\Omega})$  (in grey, underlaid); Hermitian angles  $\alpha_m(\Omega)$  for the corresponding eigenvectors  $\hat{\mathbf{q}}(e^{j\Omega})$  and  $\mathbf{q}(e^{j\Omega})$ .

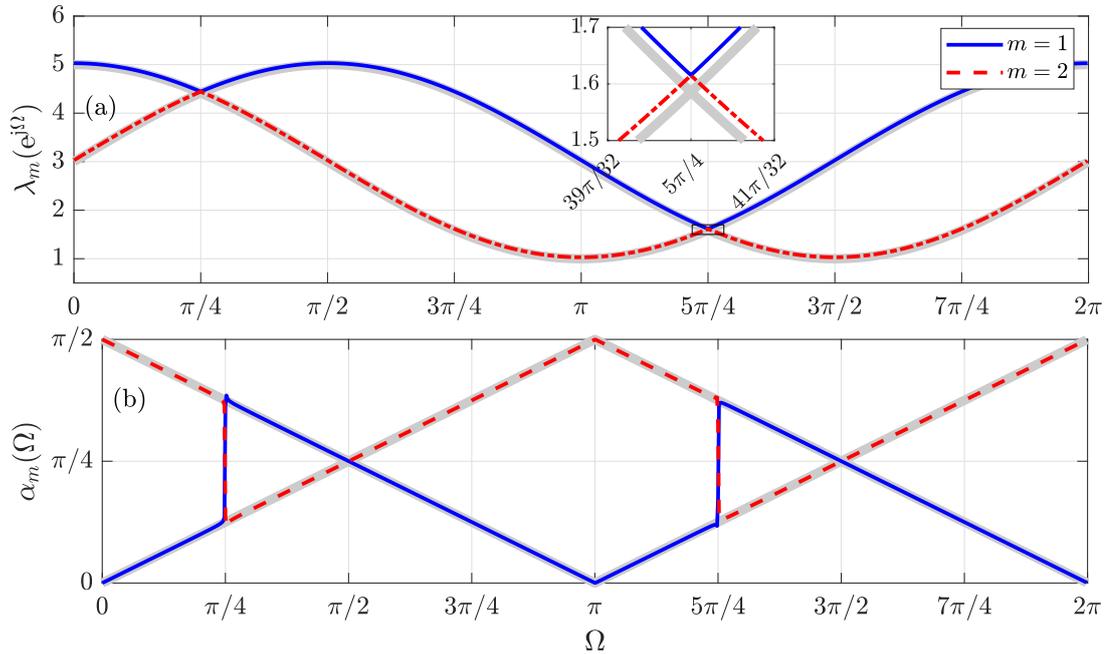


Figure 3.8: (a) eigenvalues  $\hat{\lambda}_m(e^{j\Omega})$  estimated via system identification at 20 dB SNR for  $N = 10^6$  (coloured curves) and ground truth  $\lambda_m(e^{j\Omega})$  (in grey, underlaid); Hermitian angles  $\alpha_m(\Omega)$  for the corresponding eigenvectors  $\hat{\mathbf{q}}(e^{j\Omega})$  and  $\mathbf{q}(e^{j\Omega})$ .

## Spectral Majorisation

Applications such as subband coding are optimal in terms of the coding gain if the space-time covariance matrix of the subband signals is strongly decorrelated, i.e. if  $\mathbf{R}[\tau]$  is diagonalised, and its eigenvalues are spectrally majorised [48]. Methods such as in [20] and signal compaction approaches [18] rely on this, and are supported by a number of numerical techniques to perform the analytic decomposition. This includes the class of second order sequential best rotation (SBR2) and sequential matrix diagonalisation (SMD) algorithms and their variants [13, 22, 25, 46], which tend — or in some cases are guaranteed [50] — to converge to the spectrally majorised solution.

## Analytic EVD of Multiplexed Systems

If the data vector  $\mathbf{x}[n]$  emerges from a multiplexing operation, such as for subband coding [20], then the analytic EVD of the ground truth space-time covariance  $\mathbf{R}(z)$  does not exist [44, 45]. This is due to the eigenvalues possessing a longer periodicity of  $2\pi F$ , with  $F$  representing the multiplexing factor. However, spectral majorisation will enforce a  $2\pi$  periodicity, such that an analytic EVD becomes feasible. This has been noted in [45] but without realising that the estimation error when estimating the space-time covariance from finite data, and the associated loss of algebraic multiplicities greater than one, is responsible for this beneficial effect.

### 3.4.5 Conclusions

This section investigated a fundamental effect that results in the loss of algebraic multiplicities greater than one in the eigenvalues of a space-time covariance matrix that is estimated from finite data. This effect cannot be alleviated by increasing the sample size; rather, such an increase will result in the analytic EVD factors requiring an increasing order if the ground-truth eigenvalues intersect, as non-differentiabilities and discontinuities have to be approximated when extracting the eigenvalues and eigenvectors of such an estimated space-time covariance matrix.

In terms of applications, the effect can be both beneficial or detrimental, and favours a revival of algorithms that target spectrally majorised eigenvalues for polynomial ma-

trix factorisations, which are supported by substantial algorithmic developments and implementations [31, 46]. Alternatively, analytic eigenvalue and eigenvector extraction algorithms [34, 35, 56, 84] can also yield such solutions with guaranteed spectral majorisation where current time domain methods may fail due to a large dynamic range in the eigenvalues.

### 3.5 Summary

Typically, the space-time covariance matrix is estimated from sensor measurements. However, in this chapter, a system identification approach was reviewed for a more accurate estimation. Unlike the unbiased estimator, the system identification method requires additional control over the sources to estimate the channel matrix. The system identification-based estimate results in lower bin-wise perturbation of the eigenvalues and eigenvectors, particularly for reasonable to low SNR. However, this perturbation in bin-wise eigenvalues has a significant consequence, leading to the loss of algebraic multiplicities even when the ground-truth eigenvalue has non-trivial multiplicities. It's important to note that this issue is independent of how accurate the estimate is made; even a system identification approach to the source model cannot bypass this fundamental challenge.

Consequently, building upon the discoveries presented in this chapter, the subsequent sections of this thesis adopt the assumption that any para-Hermitian or general polynomial matrix exhibits spectral majorization. Furthermore, it is assumed that eigenvalues or singular values do not exhibit any common multiplicities along the unit circle.

## Chapter 4

# Polynomial Power Method and Its Generalization

### 4.1 Introduction

In cases such as coding or compaction, which work by extracting the dominant signal component from multichannel data [17, 18, 72], a complete PEVD may be unnecessary, and it often suffices to extract the largest eigenvalue and its corresponding eigenvector, termed as principal eigenpair. For the standard EVD, this can be accomplished e.g. by applying the power method [1]. Moreover, due to the complexity of performing a full PEVD, a partial or reduced PEVD can also be considered useful for low rank applications, such as in speech enhancement where a large number of microphones may record only a very limited number of speakers [17]. In the narrowband case, the power method in conjunction with Hotelling's deflation approach [5] is well suited for factorising rank-deficient matrices, where the number of eigenvalues and eigenvectors to be determined is smaller than the dimension of the matrix.

To extend the utility of the power method [1] from narrowband to broadband sensor array problems, the power method can be extended to the polynomial power method using polynomial matrix notation with some modest modifications. Narrowband algorithms can be readily extended to their equivalent broadband cases; an example is the generalisation of the multiple signal classification (MUSIC) algorithm [7] to the

polynomial MUSIC approach [14, 15]. In addition, due to the existence of the analytic EVD [44, 45, 62], and the fact that analytic functions can be closely approximated by polynomials of finite order, the deflation concept appears viable for polynomial matrices. Hence, through the integration of the polynomial power method with polynomial matrix deflation, it becomes achievable to leverage the benefits of the power method in both narrowband and broadband contexts. While this amalgamation facilitates the PEVD of para-Hermitian polynomial matrices with low ranks, it is essential to note that the iterative application of the polynomial power method to a deflated polynomial matrix necessitates an analysis of eigenvalue perturbations.

In addition, for a general polynomial matrix, an SVD with analytic factors exists [10, 85], such that there are unique singular values that are real on the unit circle, and left- and right-singular vectors that share a common ambiguity w.r.t. arbitrary allpass functions. The PSVD algorithms mentioned in Chapter 2 ignore this coupled ambiguity, and hence typically yield complex-valued approximations of the singular values. Therefore, it further motivates the generalization of the polynomial power method from para-Hermitian matrices to general polynomial matrices for the extraction of the left- and right-singular polynomial vectors corresponding to the dominant singular value.

Therefore this chapter discusses this extension of the ordinary power method to para-Hermitian polynomial matrices, and then generalizes it to general polynomial matrices. The extension multiplies a polynomial vector with a given para-Hermitian matrix and then in each iteration performs a normalization to unity everywhere on the unit circle instead of the normalization of the approximated eigenvector to unit length in the ordinary power method. The order of the product vector, which grows with each iteration due to repeated multiplication, is limited by truncating it either to the support estimate discussed in Chapter 6, also in [86], or by trimming trailing values that fall below some threshold. The proposed approach requires that the dominant eigenvalue spectrally majorises the remaining eigenvalues; for estimated CSD matrix, this assumption is satisfied with probability one in Chapter 3 and in [37]. The polynomial equivalent of the power method is then combined with a deflation approach for the

PEVD of low rank matrices with perturbation analysis. Similarly, the generalization of the polynomial power method is only possible if the dominant singular value spectrally majorises others on the unit circle. And this assumption is similarly true for an estimated polynomial matrix because the estimation process leads to loss of algebraic multiplicities.

The chapter is organized as follows: Section 4.2 briefly explains the ordinary power method with relation to a narrowband instantaneous covariance matrix. Section 4.3 explains the extension of the ordinary power method from narrowband covariance matrices to broadband space-time covariance matrices — para-Hermitian polynomial matrices — and draws a comparison with previous PEVD algorithms. In order to compute the PEVD of low rank para-Hermitian matrices, this extension is combined with a polynomial deflation approach in Section 4.4. With ordinary generalized power method covered in Section 4.5, this polynomial power method is then generalized to ordinary polynomial matrices in Section 4.6. Every approach is compared against state-of-the-art algorithms in their respective sections and summary is provided in Section 4.7.

## 4.2 Power Iterations Method

The ordinary power method [1], while considered a fundamental concept, is briefly discussed in this context to establish a foundation for the polynomial power method.

The power method is explained here in reference to the instantaneous covariance matrix  $\mathbf{R} \in \mathbb{C}^{M \times M}$  of a narrowband MIMO system. This instantaneous covariance matrix is related to the space-time covariance matrix  $\mathbf{R}[\tau]$  as  $\mathbf{R} = \mathbf{R}[0]$ . Further, it is assumed that the eigenvalues  $\lambda_1, \dots, \lambda_m$  of  $\mathbf{R}$  satisfy  $|\lambda_1| > |\lambda_m|$  for  $m = 2, \dots, M$ . Now, if a non-zero vector  $\mathbf{v}^{(0)} \in \mathbb{C}^M$  is repeatedly multiplied against  $\mathbf{R}$ , a sequence of vectors  $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \dots, \mathbf{v}^{(k)}$  can be obtained, which can be written as

$$\mathbf{v}^{(k)} = \mathbf{R}\mathbf{v}^{(k-1)} = \mathbf{R}^k\mathbf{v}^{(0)}, \quad k = 1, 2, \dots \quad (4.1)$$

As  $\mathbf{R}$  has Hermitian symmetry and the eigenvalues are unique, its  $M$  eigenvectors  $\mathbf{q}_1, \dots, \mathbf{q}_M$  form an orthonormal basis for  $\mathbb{C}^M$ . This permits representing  $\mathbf{v}^{(0)}$  as a

linear combination of eigenvectors,

$$\mathbf{v}^{(0)} = c_1 \mathbf{q}_1 + c_2 \mathbf{q}_2 + \dots + c_M \mathbf{q}_M. \quad (4.2)$$

Substituting  $\mathbf{v}^{(0)}$  from (4.2) into (4.1), we get

$$\mathbf{v}^k = \mathbf{R}^k \sum_{m=1}^M c_m \mathbf{q}_m = \sum_{m=1}^M c_m \lambda_m^k \mathbf{q}_m = \lambda_1^k (c_1 \mathbf{q}_1 + \sum_{m=2}^M c_m (\lambda_m/\lambda_1)^k \mathbf{q}_m). \quad (4.3)$$

Over a sufficient number of iterations  $k$ , the direction  $\mathbf{v}^k$  converges to that of  $\mathbf{q}_1$  for  $c_1 \neq 0$ . It means that a random vector repeatedly multiplied with  $\mathbf{R}$  converges towards a scaled version of its principal eigenvector provided that the initial vector has a non-zero component in that direction [1]. This constitutes the power method. In order to avoid overflow or underflow,  $\mathbf{v}^k$  can be normalised in every iteration step, which makes its converge towards

$$\lim_{k \rightarrow \infty} \mathbf{v}_{\text{norm}}^{(k)} = e^{j\phi} \mathbf{q}_1, \quad \text{where } \phi = \angle c_1 \quad (4.4)$$

reflecting the phase ambiguity of eigenvectors. The rate of convergence depends upon the ratio  $|\lambda_2/\lambda_1|$ .

### 4.3 Polynomial Power Method (PPM)

This section presents the extension of the power method [1] to polynomial para-Hermitian matrices. This work has been published in [39].

#### 4.3.1 Power Method Extension to Para-Hermitian Matrices

Let us assume some vector of analytic functions  $\mathbf{v}^{(0)}[n] \bullet \rightarrow \mathbf{v}^{(0)}(z) \in \mathbb{C}^M$ . Thus the weighting functions can be written as  $\mathbf{c}(z) = \mathbf{Q}^P(z) \mathbf{v}^{(0)}(z)$ , where  $\mathbf{Q}(z)$  is the matrix of eigenvectors of  $\mathbf{R}(z)$ . Due to the analyticity of both  $\mathbf{Q}(z)$  and  $\mathbf{v}^{(0)}(z)$ ,  $\mathbf{c}(z)$  is also

analytic. Further, the paraunitarity of  $\mathbf{Q}(z)$  i.e.  $\mathbf{Q}(z)\mathbf{Q}^P(z) = \mathbf{I}$  permits

$$\mathbf{v}^{(0)}(z) = \mathbf{Q}(z)\mathbf{c}(z) = c_1(z)\mathbf{q}_1(z) + \cdots + c_M(z)\mathbf{q}_M(z), \quad (4.5)$$

i.e. the vector  $\mathbf{v}^{(0)}(z)$  can be expressed as a superposition of analytic eigenvectors, weighted by some analytic functions  $c_m(z)$ ,  $m = 1, \dots, M$ .

This polynomial vector  $\mathbf{v}^{(0)}(z)$  can be utilized as the initialisation for PPM. Due to analyticity, the analysis can be based on the unit circle with  $z = e^{j\Omega}$  for simplicity, and so  $\mathbf{v}(z)|_{z=e^{j\Omega}}$  will be

$$\mathbf{v}^{(0)}(e^{j\Omega}) = c_1(e^{j\Omega})\mathbf{q}_1(e^{j\Omega}) + \cdots + c_M(e^{j\Omega})\mathbf{q}_M(e^{j\Omega}). \quad (4.6)$$

Note that  $\mathbf{q}_m(e^{j\Omega})$  can be seen as either the  $m$ th analytic eigenvector evaluated at  $z = e^{j\Omega}$  or the  $m$ th eigenvector of a Hermitian matrix viz.  $\mathbf{R}(z)$  evaluated at a specific frequency  $\Omega$ . This permits to replicate the procedure described in Section 4.2, where by repeated multiplications, a similar sequence of frequency dependent vectors can be obtained. Hence the  $k$ th iteration vector can be written as

$$\mathbf{v}^{(k)}(e^{j\Omega}) = \mathbf{R}(e^{j\Omega})\mathbf{v}^{(k-1)}(e^{j\Omega}) = \mathbf{R}^k(e^{j\Omega})\mathbf{v}^{(0)}(e^{j\Omega}). \quad (4.7)$$

Combining (4.6) and (4.7), the frequency dependent version of (4.3) will be

$$\mathbf{v}^{(k)}(e^{j\Omega}) = (\lambda_1(e^{j\Omega}))^k \left[ c_1(e^{j\Omega})\mathbf{q}_1(e^{j\Omega}) + \sum_{m=2}^M c_m(e^{j\Omega}) \left( \frac{\lambda_m(e^{j\Omega})}{\lambda_1(e^{j\Omega})} \right)^k \mathbf{q}_m(e^{j\Omega}) \right]. \quad (4.8)$$

From the fact that the estimated  $\mathbf{R}(z)$  is strictly spectrally majorised with its eigenvalues satisfying

$$|\lambda_m(e^{j\Omega})| > |\lambda_{m+1}(e^{j\Omega})| \quad \forall \Omega \text{ for } m = 1, \dots, M-1,$$

the summation term in (4.8) will decay to zero as  $k \rightarrow \infty$ .

What is required that the first term in (4.8) converges to an unnormalised version of  $\mathbf{q}_1(z)$  evaluated at  $z = e^{j\Omega}$ ? Similar to the discussion in Sec. 4.2, it must be ensured that

the initialisation  $\mathbf{v}^{(0)}(z)$  contains a portion of the principal eigenvector  $\mathbf{q}_1(z)$ , i.e. that  $c_1(z)$  does not vanish. Since  $c_1(z)$  must be analytic, unless  $c_1(z) = 0 \forall z$ , on the unit circle  $c_1(e^{j\Omega})$  can only possess isolated zero crossings as a result of the uniqueness theorem of analytic functions [87]. Hence  $c_1(e^{j\Omega})$  must in general be non-zero except for a finite number of zero-crossings. Such zero-crossings will generate challenges for PPM; one solution is normalised regularisation which we will introduce further below.

Thus, for a sufficiently large value of  $k^1$ ,  $\mathbf{v}^{(k)}(e^{j\Omega})$  becomes

$$\mathbf{v}^{(k)}(e^{j\Omega}) = (\lambda_1(e^{j\Omega}))^k c_1(e^{j\Omega}) \mathbf{q}_1(e^{j\Omega}) . \quad (4.9)$$

If (4.9) is normalized to unit norm  $\forall \Omega \in \mathbb{R}$ , which is discussed in Section 4.3.2, it produces

$$\lim_{k \rightarrow \infty} \mathbf{v}_{\text{norm}}^{(k)}(e^{j\Omega}) = \varphi_1(e^{j\Omega}) \mathbf{q}_1(e^{j\Omega}) , \quad (4.10)$$

where  $\varphi_1(e^{j\Omega}) = c_1(e^{j\Omega})/|c_1(e^{j\Omega})|$  is an arbitrary allpass filter noting the ambiguity of analytic eigenvectors discussed in Chapter 2. This also generalizes the phase-ambiguity of the eigenvector of ordinary matrices shown in (4.4). With an analytic  $\varphi(z)$ ,  $\mathbf{v}^{(k)}(z)$  will generally be an absolutely convergent but infinite series. Since with no control over either  $c_1(e^{j\Omega})$  or  $\varphi_1(z)$ , it is necessary to perform truncation after normalization. Through repeated truncation, it is effectively similar to performing phase smoothing that helps to minimise the order [33, 47].

### 4.3.2 Normalisation and Principal Eigenvalue

For  $\mathbf{v}^{(k)}(z)$  to formally take on properties of an eigenvector, it requires normalisation, such that for the normalised vector we have  $\mathbf{v}_{\text{norm}}^{(k),P}(z) \mathbf{v}_{\text{norm}}^{(k)}(z) = 1$ . This can be accomplished as

$$\mathbf{v}_{\text{norm}}^{(k)}(e^{j\Omega}) = \frac{\mathbf{v}^{(k)}(e^{j\Omega})}{\sqrt{\mathbf{v}^{(k),H}(e^{j\Omega}) \mathbf{v}^{(k)}(e^{j\Omega}) + \varepsilon}} . \quad (4.11)$$

---

<sup>1</sup>The  $k$  will be sufficient when  $\sum_{m=2}^M c_m(e^{j\Omega}) (\lambda_m(e^{j\Omega})/\lambda_1(e^{j\Omega}))^k$  becomes negligible

where  $0 < \varepsilon \ll 1$  is a regularisation parameter. This regularisation serves two purposes: (1) it prevents a division by zero, in case the singularity of  $c_1(e^{j\Omega})$  causes problems; (2) at frequencies where the norm of  $\mathbf{v}^{(k)}(e^{j\Omega})$  is depressed because of  $c_1(e^{j\Omega})$ , it will create a bias, such that we may move away from a poor initialisation  $\mathbf{v}^{(0)}(z)$ . However, if  $\mathbf{c}(e^{j\Omega})$  is missing the components of  $\mathbf{q}_1(e^{j\Omega})$  i.e.  $c_1(e^{j\Omega}) = 0 \forall \Omega$ , PPM will not reinsert them.

To evaluate (4.11), a DFT domain approach can be applied. For a sufficiently large DFT length  $K$ , (4.11) can be computed for  $z = e^{j\Omega_i}$ ,  $i = 0, \dots, (K - 1)$ . Since the overall result is potentially infinite but absolutely convergent due to analyticity,  $\mathbf{v}_{\text{norm}}^{(k)}[n] \circ \bullet \mathbf{v}_{\text{norm}}^{(k)}(z)$  can be obtained via a  $K$ -point inverse DFT in good approximation. The solution may be non-causal and exceed the order necessary for a good approximation. Therefore, after performing normalization, a shift-corrected trimming of  $\mathbf{v}_{\text{norm}}^{(k)}[n]$  may be applied to restore causality and to limit the order of the normalised product in (4.11). This is discussed in Section 4.3.3.

The corresponding principal eigenvalue can be determined via the Rayleigh quotient

$$R_{Q,k}(z) = \frac{\mathbf{v}^{(k),P}(z)\mathbf{R}(z)\mathbf{v}^{(k)}(z)}{\mathbf{v}^{(k),P}(z)\mathbf{v}^{(k)}(z)} = \mathbf{v}_{\text{norm}}^{(k),P}(z)\mathbf{R}(z)\mathbf{v}_{\text{norm}}^{(k)}(z), \quad (4.12)$$

i.e. by a weighted inner product of the eigenvector estimate. Thus the principal eigenvalue can be related to  $R_{Q,k}(z)$  as  $\lambda_1(z) = \lim_{k \rightarrow \infty} R_{Q,k}(z)$ .

### 4.3.3 Order Limitation by Shift-Corrected Truncation

To limit the order-growth of  $\mathbf{v}_{\text{norm}}^{(k)}(z)$ , two different order-limitation approaches are defined.

#### Order Limitation to Eigenvector Support

It is possible to accurately estimate the support of  $\mathbf{Q}(z)$  before extracting any of its columns constituting the eigenvectors [86]. This estimated support  $N$  can be exploited

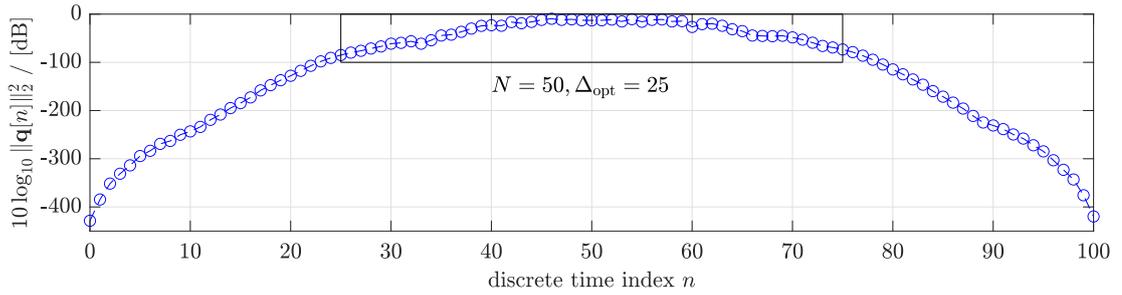


Figure 4.1: Optimal shift for approximation of a 100 order vector  $\mathbf{q}(z)$ , with unit norm on the unit circle, by an  $N = 50$ th order polynomial using a suitable window to minimise the truncation error.

by truncating  $\mathbf{v}_{\text{norm}}^{(k)}[n]$  subject to an optimal shift-correction akin to [27, 34], such that

$$\tilde{\mathbf{v}}_{\text{norm}}^{(k)}[n] = \mathbf{v}_{\text{norm}}^{(k)}[n - \Delta_{\text{opt}}]p_N[n], \quad (4.13)$$

where  $p_N[n]$  is a rectangular window of size  $N$ , and the shift  $\Delta_{\text{opt}}$  is calculated via

$$\Delta_{\text{opt}} = \arg \max_{\Delta} \sum_{n=0}^{N-1} \|\mathbf{v}[n - \Delta]\|_2^2. \quad (4.14)$$

This actually minimizes the truncation error in truncating  $\mathbf{v}_{\text{norm}}^{(k)}(z)$  to best approximate it in least square sense via an  $N$ th order vector. The optimal shift parameter is further explained via an example.

**Example 4.** A paraunitary matrix  $\mathbf{Q}(z) \in \mathbb{C}^{2 \times 2}$  of order 100 is generated through a concatenation of elementary paraunitary elements [36]. The norm of one of the column i.e.  $\mathbf{q}[n] \circ \bullet \mathbf{q}(z)$  is shown in Fig. 4.1. To approximate  $\mathbf{q}(z)$  by  $N = 50$ th order polynomials with minimum truncation error, the optimum shift found via (4.14) is  $\Delta_{\text{opt}} = 25$ . △

The truncation in (4.13) has to be performed in each iteration to ease the computational burden of the multiplication and normalization steps. This repeated truncation of the time domain vector is analogous to phase smoothing in the frequency domain. Its purpose is to find the eigenvector with minimum time-domain support given its ambiguity w.r.t. an arbitrary allpass filter  $\phi_1(z)$ , as discussed in Chapter 2.

## Direct Truncation of Coefficients

This method truncates the outer lags of  $\mathbf{v}_{\text{norm}}^{(k)}[n]$  on either end if its norm  $\|\mathbf{v}_{\text{norm}}^{(k)}[n]\|_2$  falls below some threshold. Since for a low threshold the resulting order will be high, this method may produce higher-order approximations compared to the previous method, particularly if the truncation length  $N$  is inappropriate for intermediate solution after insufficient iteration steps. While the direct truncation is simple, there currently does not exist any well-defined criterion to determine the appropriate truncation threshold.

### 4.3.4 Stopping Criterion

In order to define a suitable stopping criterion for the PPM, it is necessary to know how closely aligned two polynomial vectors  $\mathbf{v}^{(k)}(z)$  and  $\mathbf{v}^{(k-1)}(z)$  are. Evaluated on the unit circle, the Hermitian angle  $\angle\{\mathbf{a}(z), \mathbf{b}(z)\}$  between any two polynomial vectors  $\mathbf{a}(z), \mathbf{b}(z) : \mathbb{C} \rightarrow \mathbb{C}^M$  can be employed, which is defined as

$$\angle\{\mathbf{a}(e^{j\Omega}), \mathbf{b}(e^{j\Omega})\} = \text{acos} \left( \frac{|\mathbf{a}^H(e^{j\Omega})\mathbf{b}(e^{j\Omega})|}{\|\mathbf{a}(e^{j\Omega})\|_2 \cdot \|\mathbf{b}(e^{j\Omega})\|_2} \right). \quad (4.15)$$

This Hermitian angle concept can be exploited in defining a suitable a metric

$$\alpha(\Omega) = \angle\{\tilde{\mathbf{v}}_{\text{norm}}^{(k)}(e^{j\Omega}), \tilde{\mathbf{v}}_{\text{norm}}^{(k-1)}(e^{j\Omega})\}. \quad (4.16)$$

It should be noted that the normalisation in (4.15) is retained in case of errors introduced due to truncation. If two successive estimates are aligned, the respective  $\alpha(\Omega) = 0 \forall \Omega$ . To measure an overall deviation,

$$\gamma = \frac{1}{2\pi} \int_{-\pi}^{\pi} |\alpha(\Omega)|^2 d\Omega \quad (4.17)$$

is a suitable criterion for convergence, while  $\alpha(\Omega)$  permits to check on any frequency-dependent differences in convergence.

### 4.3.5 Simulations and Results

#### Order Limitation and Initialisation Effects

To evaluate the impact of the two types of order limitations, mentioned in Sec-4.3.3, on convergence and efficiency, a simple para-Hermitian matrix  $\mathbf{R}(z) : \mathbb{C} \rightarrow \mathbb{C}^{3 \times 3}$  with eigenvalues

$$\begin{aligned}\lambda_1(z) &= z(6 + j)/100 + 1.01 + z^{-1}(6 - j)/100 \\ \lambda_2(z) &= -z(1 - 2j)/100 + 0.86 - z^{-1}(1 + 2j)/100 \\ \lambda_3(z) &= z(5 - 2j)/100 + 0.71 + z^{-1}(5 + 2j)/100 ,\end{aligned}$$

which are illustrated in Fig. 4.2, is being considered. The matrix of eigenvectors  $\mathbf{Q}(z)$  is defined by a sequence of elementary paraunitary operations [48],

$$\mathbf{Q}(z) = [\mathbf{q}_1(z), \dots, \mathbf{q}_3(z)] = \prod_{i=1}^4 (\mathbf{I} + (z^{-1} - 1)\mathbf{e}_i\mathbf{e}_i^H) ,$$

with unit norm vectors  $\mathbf{e}_i$ ,  $i = 1, \dots, 4$ ,

$$\mathbf{e}_{i=\{1,3\}} = \frac{1}{\sqrt{2}} [1, 0, \mp 1]^T , \quad \mathbf{e}_{i=\{2,4\}} = \frac{1}{\sqrt{2}} [\pm 1, 1, 0]^T .$$

With 500 random initializations for  $\mathbf{v}^{(0)}[n]$ , Algorithm 1 is executed over  $k = 300$  iterations, recording both the vector alignment  $\gamma$  of (4.17) and the normalised error in the principal eigenvalue,

$$\xi_\lambda = \frac{\sum_\tau |\lambda[\tau] - \hat{\lambda}[\tau]|^2}{\sum_\tau |\lambda[\tau]|^2} . \quad (4.18)$$

Both types of limitation strategies are tested. For the second method, the threshold for the truncation of trailing coefficients is set to  $10^{-3}$ .

The results, illustrated in Fig. 4.3, show that Algorithm 1 with truncation method 1 (in black) is generally less costly per iteration and therefore terminates within shorter time. The execution time for the truncation method depends on the truncation threshold. If the coefficient truncation threshold is small, the execution time will be higher

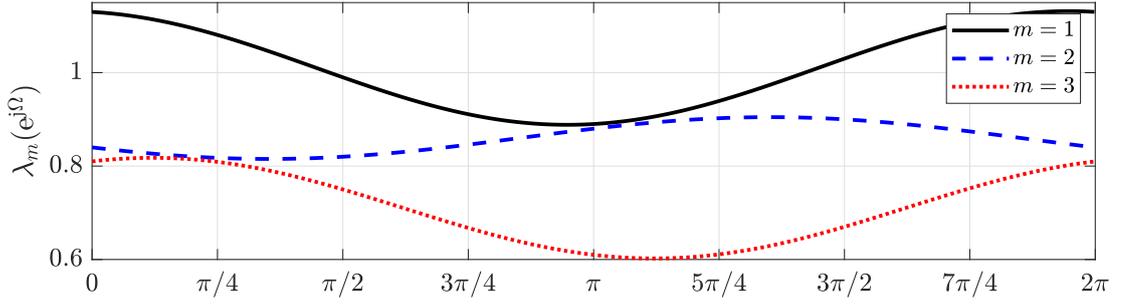


Figure 4.2: Eigenvalues for the example matrix  $\mathbf{R}(z)$ .

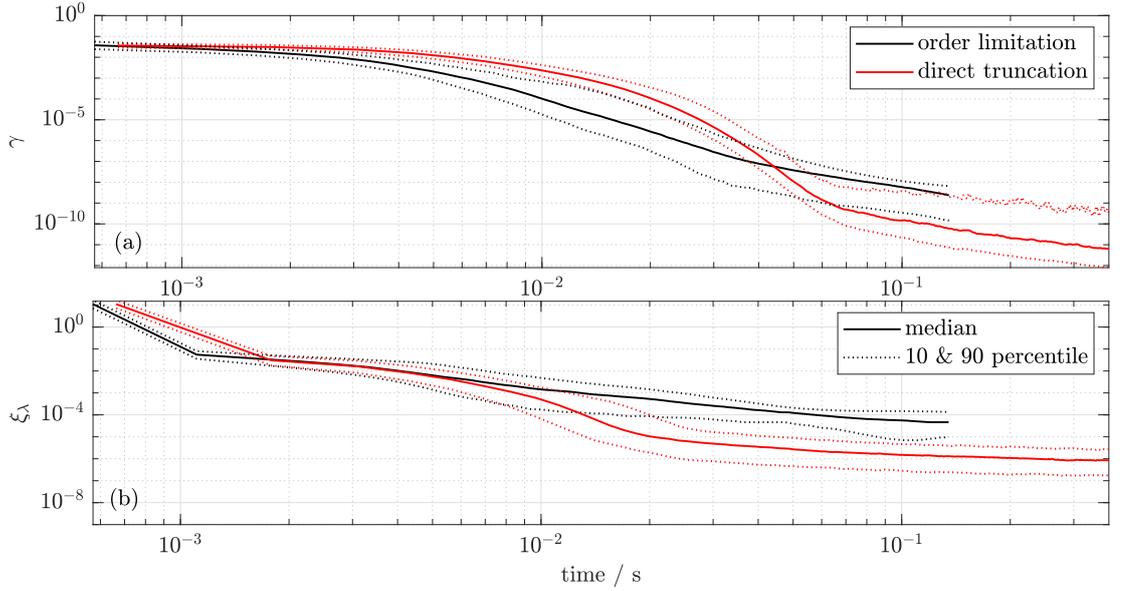


Figure 4.3: Effects of types of order limitation on algorithm convergence, measuring (a) angle deviation metric  $\gamma$ , and (b) accuracy of the extracted eigenvalue.

because a of more relaxed order limitation and hence higher cost of each iteration. For the given ensemble experiment, the order of the resulting  $\mathbf{v}_{\text{norm}}^{(k)}(z)$  is 4 and 49 under the order limitation method 1 and the truncation method 2, respectively. Despite all this, the direct truncation method employed by Algorithm 1 achieves a lower value of overall deviation metric i.e.  $\gamma$  in the extraction of principal eigenvector which makes it more accurate in comparison to the order limitation method (see Fig. 4.3(a)). For instance, at time  $10^{-1}$ s, the direct truncation method attains  $\gamma = 10^{-10}$  and the order limitation achieves  $\gamma = 10^{-8}$ . Similarly, the direct truncation method obtains a more accurate eigenvalue estimate than the order limitation method, reflected in the form of

---

**Algorithm 1: PPM Algorithm**

---

**Input:**  $\mathbf{R}(z)$ ,  $\epsilon$ ,  $k_{\max}$   
**Output:**  $\hat{\mathbf{q}}_1(z)$ ,  $\hat{\lambda}_1(z)$   
 $\mathbf{v}^{(0)}(z) \in \mathbb{C}^M$ ,  $k \leftarrow 0$ ,  $\gamma = \infty$ ;  
 $\tilde{\mathbf{v}}_{\text{norm}}^{(0)}(z) \leftarrow$  normalise & order limit  $\mathbf{v}^{(0)}(z)$  ;  
**while**  $\gamma > \epsilon$  &  $k < k_{\max}$  **do**  
     $k \leftarrow k + 1$ ;  
     $\mathbf{v}^{(k)}(z) \leftarrow \mathbf{R}(z)\tilde{\mathbf{v}}_{\text{norm}}^{(k-1)}(z)$  ;  
     $\mathbf{v}_{\text{norm}}^{(k)}(z) \leftarrow$  normalisation  $\mathbf{v}^{(k)}(z)$ ;  
     $\tilde{\mathbf{v}}_{\text{norm}}^{(k)}(z) \leftarrow$  order limitation of  $\mathbf{v}_{\text{norm}}^{(k)}(z)$ ;  
    update  $\gamma$   
**end**  
 $\hat{\mathbf{q}}_1(z) = \tilde{\mathbf{v}}_{\text{norm}}^{(k)}(z)$ ;  
 $\hat{\lambda}_1(z) = \tilde{\mathbf{v}}_{\text{norm}}^{(k),\text{P}}(z)\mathbf{R}(z)\tilde{\mathbf{v}}_{\text{norm}}^{(k)}(z)$ ;

---

a lower value of the normalized error metric  $\xi_\lambda$  as shown in Fig. 4.3.

To assess the impact of a singularity in  $c_1(e^{j\Omega})$ , we use a simple system with  $\lambda_1(e^{j\Omega}) = 1$  and  $\lambda_2(e^{j\Omega}) = \frac{1}{2}$ . With eigenvectors  $\mathbf{q}_{1,2}(z) = [1; \pm z^{-1}]/\sqrt{2}$  and initialisation  $\mathbf{v}^{(0)}(z) = [(1 - z^{-1})\mathbf{q}_1(z) + (1 + z^{-1})\mathbf{q}_2(z)]/\sqrt{2}$ , we have  $c_1(e^{j\Omega}) = 0$  for  $\Omega = 0$ . Nonetheless, a regularisation  $\epsilon = 0.1$  leads to the convergence of the Hermitian angle in Fig. 4.4, showing that the singularity is overcome, even though the bias introduced by  $\epsilon$  causes a noise floor to the alignment of the estimated principal eigenvector with the ground truth.

### Comparison with State-Of-The-Art

The proposed PPM approach is compared to the state-of-the-art algorithms SBR2 [13] and SMD [22], through an ensemble of  $10^3$  randomised spectrally majorised para-Hermitian matrices  $\mathbf{R}(z) \in \mathbb{C}^{4 \times 4}$  based on the source model in [22]. In this model, the order of the ground truth eigenvectors  $\mathcal{O}\{\mathbf{Q}(z)\}$ , where  $\mathcal{O}\{\cdot\}$  measures the polynomial order of its argument, is varied from 20 to 100 in steps of 20. The proposed method is executed with  $k_{\max} = 10^3$  and  $\epsilon = 10^{-10}$ . The support of  $\tilde{\mathbf{v}}_{\text{norm}}^{(k)}[n]$  is truncated with a threshold of  $10^{-3}$  via the second method and the regularisation term  $\epsilon$  is set to zero to see if any issue is encountered with random normalization. Additionally, truncating

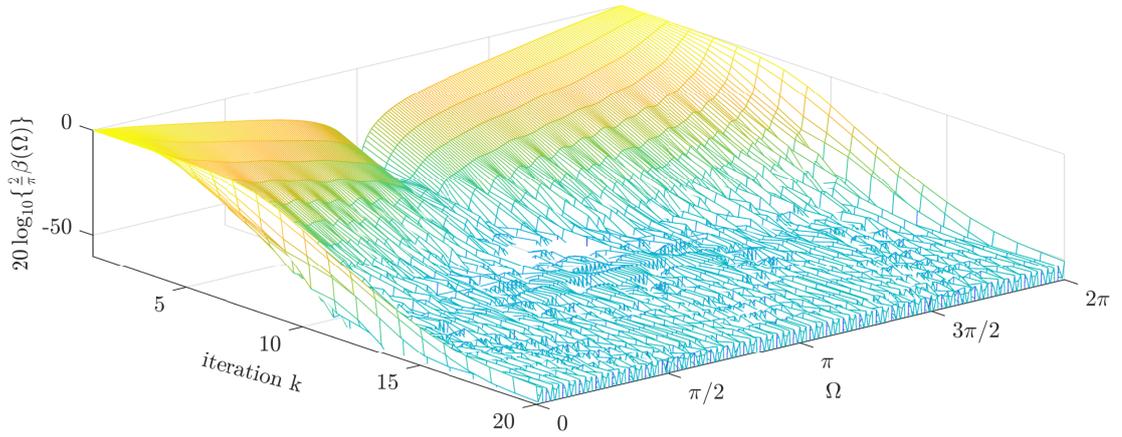


Figure 4.4: Convergence of  $\beta(\Omega) = \angle\{\tilde{\mathbf{v}}_{\text{norm}}^{(k)}(e^{j\Omega}), \mathbf{q}_1(e^{j\Omega})\}$  in case of a singularity of  $c_1(e^{j\Omega})$  for  $\Omega = 0$ .

the order of  $\tilde{\mathbf{v}}_{\text{norm}}^{(k)}[n]$  to have same order as the ground truth may impede the PPM's convergence for some initializations or it may not converge at all because of the non-convex nature of the phase smoothing process [34]. The SBR2 and SMD algorithms, on the other hand, are allowed a maximum of  $10^3$  iterations with truncation parameter  $\mu$ , meant to limit the order of any intermediate  $\hat{\mathbf{q}}_1(z)$  by truncating its outer lags with the same threshold of  $10^{-3}$  as for PPM, while the maximum off-diagonal threshold to is chosen as  $10^{-6}$ .

All the above methods are evaluated based on the order of extracted eigenvector  $\mathcal{O}\{\hat{\mathbf{q}}_1(z)\}$ ,  $\xi_\lambda$  and the execution time. Note that none of the instances of the ensemble are excluded due to possible poor initialisation. The ensemble results are illustrated in Fig. 4.5 which shows that the PPM performance is better than both SBR2 and SMD w.r.t all three metrics. Especially, the order of the extracted eigenvector is orders of magnitude lower than that of SBR2 and SMD. Although convergence is yet to be proven, the ensemble results suggest that the proposed method converges to an acceptable approximate principal eigenpair as evident from Fig. 4.5. Furthermore, the ensemble results, which look fairly acceptable at least within 10 to 90 percentile, show that there is only a low probability of encountering a zero crossings for  $c_1(z)$  when randomly initialising  $\mathbf{v}^{(0)}(z)$ .

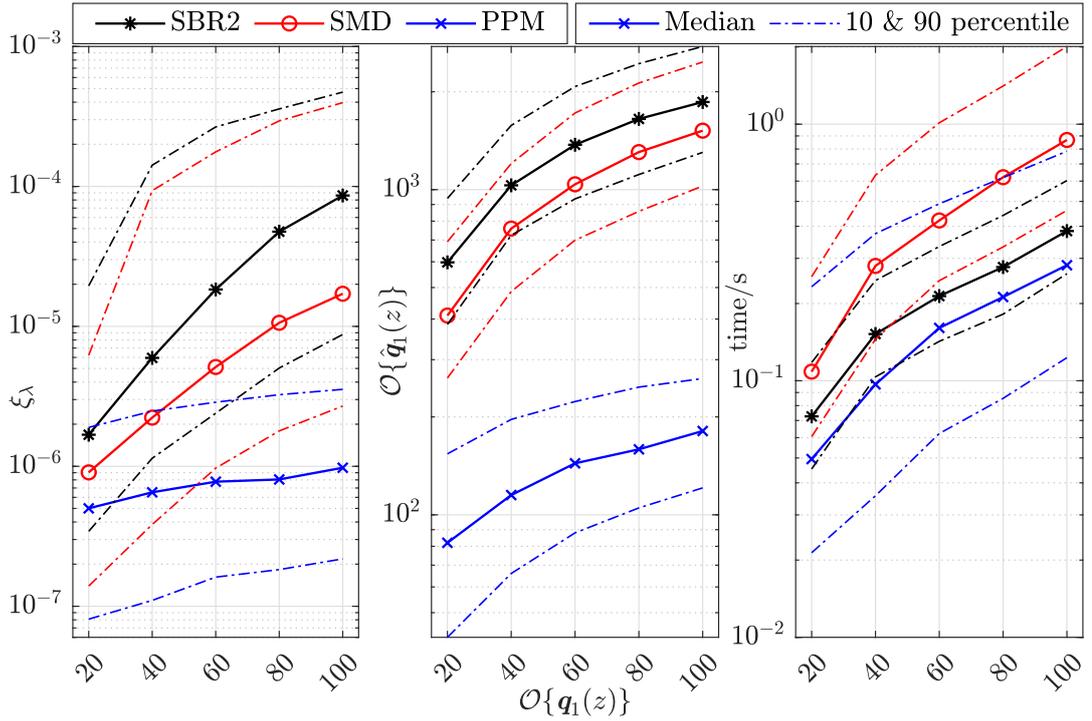


Figure 4.5: Performance metrics comparison (a)  $\xi_\lambda$ , (b)  $\mathcal{O}\{\hat{\mathbf{q}}_1(z)\}$ , and (c) execution time of SBR2, SMD and PPM in dependence of the ground truth order of the principal eigenvector.

## 4.4 Low-rank PEVD through Deflation

This section applies the polynomial power method to compute the PEVD of a rank-deficient para-Hermitian matrix with the help of deflation. Moreover, perturbation of the eigenvalues and eigenvectors due to error propagation in the deflation process is also analysed.

### 4.4.1 EVD via Power Method and Deflation

A Hermitian matrix  $\mathbf{R} \in \mathbb{C}^{M \times M}$  with  $p \leq M$  non-zero eigenvalues  $\lambda_m \in \mathbb{R}$ ,  $m = 1, \dots, p$  can be represented as a sum of rank one terms

$$\mathbf{R} = \sum_{m=1}^p \mathbf{q}_m \mathbf{q}_m^H \lambda_m = \sum_{m=1}^p \mathbf{R}_m, \quad (4.19)$$

where  $\mathbf{q}_m$  is the  $m$ th eigenvector and  $\mathbf{R}_m$  is a rank one Hermitian matrix, whose columns are spanned by  $\mathbf{q}_m$ . It is assumed that  $\mathbf{R}$  is positive semi-definite, and its  $p$  non-zero eigenvalues are distinct and majorised as  $\lambda_m > \lambda_{m+1}, m = 1, \dots, (p-1)$ . With the dominant eigenpair, i.e.  $\{\mathbf{q}_1, \lambda_1\}$ , determined via the power method, the matrix  $\mathbf{R}$  can be deflated by removing the contribution of the dominant eigenpair as

$$\mathbf{R}^{(2)} = \mathbf{R} - \mathbf{R}_1 = \mathbf{R} - \mathbf{q}_1 \lambda_1 \mathbf{q}_1^H. \quad (4.20)$$

If the estimated eigenpair is sufficiently accurate, then the deflated matrix  $\mathbf{R}^{(2)}$  has the decremented rank  $(p-1)$ , and its dominant eigenpair is now  $\{\mathbf{q}_2, \lambda_2\}$ . This second eigenpair can be extracted by a repeat of the power method on the matrix  $\mathbf{R}^{(2)}$ . In turn,  $\mathbf{R}^{(2)}$  can now be deflated, and through a total of  $p$  iteration, all eigenpairs of  $\mathbf{R}$  can be determined. As a recursive formulation with the initialisation  $\mathbf{R}^{(1)} = \mathbf{R}$ , the scheme operates via

$$\mathbf{R}^{(m+1)} = \mathbf{R}^{(m)} - \lambda_m \mathbf{q}_m \mathbf{q}_m^H, \quad (4.21)$$

with  $m = 1, \dots, p$ . Ideally, by exactly extracting the dominant eigenpair  $\{\mathbf{q}_m, \lambda_m\}$  at the  $m$ th iteration, we finally end up with  $\mathbf{R}^{(p+1)} = \mathbf{0}$ .

If an eigenpair, such as the first one in (4.21), is inaccurate, i.e. an estimate  $\{\hat{\mathbf{q}}_m, \hat{\lambda}_m\}$  is obtained via a limited number of iterations  $k$  instead of the exact  $\{\mathbf{q}_m, \lambda_m\}$ , then  $\mathbf{R}^{(m+1)}$  will be perturbed by the error  $\mathbf{R}_m - \hat{\mathbf{R}}_m$ , with  $\hat{\mathbf{R}}_m = \hat{\mathbf{R}}^{(m)} - \hat{\mathbf{q}}_m \hat{\mathbf{q}}_m^H \hat{\lambda}_m$ . This perturbation term will (i) lead to an insufficient rank reduction, and (ii) cause error propagation as subsequent eigenpairs are estimated with decreasing accuracy. To investigate such perturbation effects, and hence potential bounds on the estimation error of subsequently extracted eigenpairs, a perturbation analysis [4], in part for reduced-rank perturbations [88, 89], can be employed.

#### 4.4.2 Para-Hermitian Matrix Rank One Representation and Deflation

Similar to a Hermitian matrix  $\mathbf{R}$ , a para-Hermitian matrix can be represented as the sum of rank one para-Hermitian matrices  $\mathbf{R}_m(z)$ ,

$$\mathbf{R}(z) = \sum_{m=1}^p \mathbf{q}_m(z) \mathbf{q}_m^P(z) \lambda_m(z) = \sum_{m=1}^p \mathbf{R}_m(z), \quad (4.22)$$

where  $p \leq M$  is the number of non-zero analytic eigenvalues. This entire section assumes that  $\mathbf{R}(z)$  possesses rank  $p < M$ , in which case  $\lambda_{p+1}(e^{j\Omega}) = \dots = \lambda_M(e^{j\Omega}) = 0 \forall \Omega$ , i.e. that there are  $(M - p)$  eigenvalues that are identical to zero. In the case of  $p < M$  for  $\mathbf{R}(z)$ , this thesis refers to  $\mathbf{R}(z)$  as a low-rank polynomial matrix. This shows that if an eigenpair is available, deflation can be performed to reduce the rank of  $\mathbf{R}(z)$ . For instance, if  $\{\mathbf{q}_1(z), \lambda_1(z)\}$  is extracted via the polynomial power method, its contribution can be removed from the original para-Hermitian matrix as

$$\mathbf{R}^{(2)}(z) = \mathbf{R}(z) - \mathbf{q}_1(z) \mathbf{q}_1^P(z) \lambda_1(z) = \mathbf{R}(z) - \mathbf{R}_1(z). \quad (4.23)$$

The allpass ambiguity of the extracted eigenvector mentioned in Section 2.2 does not cause any issue since with  $\phi_1(z) \phi_1^P(z) = 1$  this ambiguity drops out.

The polynomial power method can be repeated on  $\mathbf{R}^{(2)}(z)$ , if the dominant eigenpair is accurate. Thus over  $p - 1$  deflations and  $p$  application of the polynomial power method, an analytic EVD can be computed using a recursive procedure akin to (4.21), such that with  $\mathbf{R}^{(1)}(z) = \mathbf{R}(z)$ ,

$$\mathbf{R}^{(m+1)}(z) = \mathbf{R}^{(m)} - \underbrace{\mathbf{q}_m(z) \mathbf{q}_m^P(z) \lambda_m(z)}_{\mathbf{R}_m(z)}. \quad (4.24)$$

The approach in (4.24) requires the accurate determination of eigenpairs  $\{\mathbf{q}_m(z), \lambda_m(z)\}$  via the polynomial power method at every stage. Estimation errors due to a limited number of iterations  $k$  in the polynomial power method will result not only in estimated eigenpairs  $\{\hat{\mathbf{q}}_m(z), \hat{\lambda}_m(z)\}$  that may differ from the desired quantities, but will also lead to potentially inaccurate estimates  $\hat{\mathbf{R}}_m(z)$  of the rank one

matrices and  $\hat{\mathbf{R}}^{(m+1)}(z)$  of the deflated matrices w.r.t. the quantities defined in (4.24). It is therefore required to investigate how incorrect eigenpairs and rank-one estimates perturb the subsequent extraction of any remaining eigenpairs.

#### 4.4.3 Perturbation Analysis and Error Propagation

The perturbation and error propagation analysis here is defined as per frequency bin, i.e. for any specific frequency  $\Omega$  on the unit circle. This is carried out by assessing the difference  $\mathbf{E}^{(m+1)}(e^{j\Omega})$  between the correctly deflated matrix  $\mathbf{R}^{(m+1)}(e^{j\Omega})$  after the  $m$ th rank deflation, and its estimate,  $\hat{\mathbf{R}}^{(m+1)}(e^{j\Omega})$ :

$$\mathbf{E}^{(m+1)}(e^{j\Omega}) = \mathbf{R}^{(m+1)}(e^{j\Omega}) - \hat{\mathbf{R}}^{(m+1)}(e^{j\Omega}), \quad (4.25)$$

for  $m = 1, \dots, (p-1)$ . Since due to the rank one deflations

$$\mathbf{R}^{(m+1)}(e^{j\Omega}) = \mathbf{R}(e^{j\Omega}) - \mathbf{R}_1(e^{j\Omega}) - \dots - \mathbf{R}_m(e^{j\Omega}), \quad (4.26)$$

$$\hat{\mathbf{R}}^{(m+1)}(e^{j\Omega}) = \mathbf{R}(e^{j\Omega}) - \hat{\mathbf{R}}_1(e^{j\Omega}) - \dots - \hat{\mathbf{R}}_m(e^{j\Omega}), \quad (4.27)$$

we have

$$\begin{aligned} \mathbf{E}^{(m+1)}(e^{j\Omega}) &= \sum_{\mu=1}^m \hat{\mathbf{R}}_{\mu}(e^{j\Omega}) - \mathbf{R}_{\mu}(e^{j\Omega}) \\ &= \sum_{\mu=1}^m \hat{\mathbf{q}}_{\mu}(e^{j\Omega}) \hat{\mathbf{q}}_{\mu}^H(e^{j\Omega}) \hat{\lambda}_{\mu}(e^{j\Omega}) - \mathbf{q}_{\mu}(e^{j\Omega}) \mathbf{q}_{\mu}^H(e^{j\Omega}) \lambda_{\mu}(e^{j\Omega}). \end{aligned} \quad (4.28)$$

This can also be written recursively as

$$\mathbf{E}^{(m+1)}(e^{j\Omega}) = \mathbf{E}^{(m)}(e^{j\Omega}) + \left( \hat{\mathbf{q}}_m(e^{j\Omega}) \hat{\mathbf{q}}_m^H(e^{j\Omega}) \hat{\lambda}_m(e^{j\Omega}) - \mathbf{q}_m(e^{j\Omega}) \mathbf{q}_m^H(e^{j\Omega}) \lambda_m(e^{j\Omega}) \right). \quad (4.29)$$

Since the eigenvectors of a para-Hermitian matrix can be selected to be orthonormal, the different terms in the sum of (4.28) are approximately (due to estimation errors)

orthogonal, and for sufficiently small perturbations it can be shown that

$$\begin{aligned} \|\mathbf{E}^{(m+1)}(e^{j\Omega})\|_{\mathbb{F}}^2 &\approx \|\mathbf{E}^{(m)}(e^{j\Omega})\|_{\mathbb{F}}^2 + \left\| \hat{\mathbf{q}}_m(e^{j\Omega}) \hat{\mathbf{q}}_m^H(e^{j\Omega}) \hat{\lambda}_m(e^{j\Omega}) - \mathbf{q}_m(e^{j\Omega}) \mathbf{q}_m^H(e^{j\Omega}) \lambda_m(e^{j\Omega}) \right\|_{\mathbb{F}}^2 \\ &\geq \|\mathbf{E}^{(m)}(e^{j\Omega})\|_{\mathbb{F}}^2. \end{aligned} \quad (4.30)$$

Hence, the error norm does not improve over subsequent deflation operations, and generally tends to grow.

The effect of the above error on the eigenpair  $\{\mathbf{q}_{m+1}(e^{j\Omega}), \lambda_{m+1}(e^{j\Omega})\}$  that is intended to be extracted from  $\mathbf{R}^{(m+1)}(e^{j\Omega})$  can now be assessed. Similarly, the estimate  $\{\hat{\mathbf{q}}_{m+1}(e^{j\Omega}), \hat{\lambda}_{m+1}(e^{j\Omega})\}$  is extracted from  $\hat{\mathbf{R}}^{(m+1)}(e^{j\Omega})$ . Utilizing (4.25) and referring to the Bauer-Fike theorem [90], it can be observed that the accuracy of the extracted  $(m+1)$ st eigenvalue is upper-bounded as follows:

$$|\lambda_{m+1}(e^{j\Omega}) - \hat{\lambda}_{m+1}(e^{j\Omega})| \leq \|\mathbf{E}^{(m+1)}(e^{j\Omega})\|_{\mathbb{F}}^2. \quad (4.31)$$

Therefore, the worst case accuracy of the  $(m+1)$ st eigenvalue is determined by the cumulative error  $\mathbf{E}^{(m+1)}(e^{j\Omega})$ . Regarding the eigenvectors, the subspace distance  $\mathcal{U}_{m+1}(e^{j\Omega})$  can be defined using the spectral norm  $\|\cdot\|_2$  [1] to measure the difference between projections as

$$\mathcal{U}_{m+1}(e^{j\Omega}) = \left\| \mathbf{q}_{m+1}(e^{j\Omega}) \mathbf{q}_{m+1}^H(e^{j\Omega}) - \hat{\mathbf{q}}_{m+1}(e^{j\Omega}) \hat{\mathbf{q}}_{m+1}^H(e^{j\Omega}) \right\|_2. \quad (4.32)$$

Then perturbation theory [1] provides an upper bound

$$\mathcal{U}_{m+1}(e^{j\Omega}) \leq \frac{4}{d} \|e_{m+1}(e^{j\Omega})\|_2, \quad (4.33)$$

where  $d = \lambda_{m+1}(e^{j\Omega}) - \lambda_{m+2}(e^{j\Omega})$  is the distance to the next-nearest eigenvalue, and  $e_{m+1}(e^{j\Omega}) \in \mathbb{C}^{M-1}$  comes from a partition of  $\mathbf{E}^{(m+1)}(e^{j\Omega})$ ,

$$\mathbf{E}^{(m+1)}(e^{j\Omega}) = \begin{bmatrix} e_{m+1}(e^{j\Omega}) & \mathbf{e}_{m+1}^H(e^{j\Omega}) \\ e_{m+1}(e^{j\Omega}) & \mathbf{E}_{2,m+1}(e^{j\Omega}) \end{bmatrix}. \quad (4.34)$$

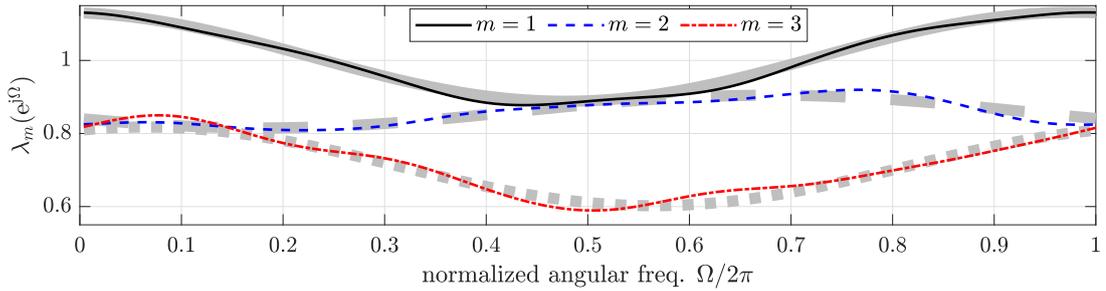


Figure 4.6: Example for the ground truth (shaded grey) and estimated eigenvalues using the proposed deflation approach (in colour)  $\hat{\lambda}_m(e^{j\Omega})$ ,  $m = 1, 2, 3$  for the example matrix  $\mathbf{R}(z)$  when injecting perturbation through insufficient convergence of the PPM algorithm.

Thus, the upper bound on the accuracy of the  $m+1$ st eigenvector extracted by deflation also depends on the accumulated errors in  $\mathbf{E}^{(m+1)}(e^{j\Omega})$ . Hence, any inaccuracies in a rank one estimate will impact on and further degrade the precision bounds with which any remaining eigenpairs can be determined.

**Example 5.** To demonstrate the deflation concept combined with the polynomial power method, a spectrally majorised para-Hermitian example from Section 4.3.5 is considered. The exact eigenvalues of  $\mathbf{R}(z)$  are shown in Fig. 4.6. These are compared to eigenvalues extracted by the deflation approach based on the polynomial power method executed with  $k_{\max} = 5e3$ ,  $\epsilon = 10^{-4}$  and  $\mathbf{x}^{(0)}(z) = \sum_{i=0}^4 z^{-i}$ . The difference between the estimated and the ground-truth eigenvalues can be seen in Fig. 4.6, and is measured for the  $m$ th eigenvalue  $\lambda_m[\tau] \circ \bullet \lambda_m(z)$  via (4.18). For  $\hat{\lambda}_1(z)$ , which is extracted by the polynomial power method from  $\mathbf{R}^{(1)}(z) = \mathbf{R}(z)$ , we obtain  $\xi_{\lambda_1} = 6.8 \times 10^{-5}$ . By subsequent deflation, from  $\hat{\mathbf{R}}^{(2)}$  we obtain the second eigenpair with  $\xi_{\lambda_2} = 1.55 \times 10^{-4}$ . This shows that the extracted second eigenvalue is not as accurate as the first one. The third eigenvalues is then obtained from  $\hat{\mathbf{R}}^{(3)}(z)$  with  $\xi_{\lambda_3} = 3.6 \times 10^{-4}$ . It can be seen that due to error propagation, indeed  $\xi_{\lambda_3} > \xi_{\lambda_2} > \xi_{\lambda_1}$ , and the error is increasing with each extraction. The impact of perturbations and error propagation can also be seen in the estimated eigenvalues shown in Fig. 4.6, where the third eigenvalue appears to have the largest estimation error.  $\triangle$

#### 4.4.4 Application and Ensemble Simulation

This section provides an ensemble test to demonstrate the enhanced performance of the proposed approach for the PEVD of low-rank para-Hermitian matrices. The performance metrics selected for comparison are the resulting order of the estimated paraunitary matrix  $\hat{\mathbf{Q}}(z)$ , denoted as  $\mathcal{O}(\hat{\mathbf{Q}}(z))$ , the execution time  $t$  of the approach, and the reconstruction metric  $\xi_R$ . The latter is defined as

$$\xi_R = \frac{\sum_{\tau} \|\mathbf{R}[\tau] - \hat{\mathbf{R}}[\tau]\|_{\mathbb{F}}^2}{\sum_{\tau} \|\mathbf{R}[\tau]\|_{\mathbb{F}}^2}, \quad (4.35)$$

and measures the accuracy of the decomposition, where with the convolution operator  $*$ ,  $\hat{\mathbf{R}}[\tau] = \hat{\mathbf{Q}}[\tau] * \hat{\mathbf{A}}[\tau] * \hat{\mathbf{Q}}^{\text{H}}[-\tau]$ .

For an exhaustive test, an ensemble comprising of 100 instantiations of  $6 \times 6$  para-Hermitian matrices of rank two is constructed, where each instance represents a system of two spectrally majorised broadband sources illuminating an array of  $M = 6$  sensors through a convolutive mixing system. The instantiations are generated using the source model in [22], with the source power spectral densities and the convolutive paraunitary mixing defining the ground truth analytic EVD. The concatenation of spectral shaping and mixing jointly form a system  $\mathbf{H}(z) : \mathbb{C} \rightarrow \mathbb{C}^{6 \times 2}$  of order 100. The resulting cross-spectral density matrix  $\mathbf{R}(z) = \mathbf{H}(z)\mathbf{H}^{\text{P}}(z)$  is therefore of order 200.

The PPM algorithm is executed with  $k_{\text{max}} = 10^3$  and  $\epsilon = 10^{-7}$ . The trailing coefficients of the normalized vector are truncated once they fall below a threshold of  $10^{-3}$ . The support of the initial vector  $\mathbf{x}^{(0)}(z)$  is set to the estimated support of the eigenvectors, which can be evaluated via [86]; its coefficients are drawn from a complex-valued normal distribution. The state-of-the-art algorithms SBR2 [13] and SMD [22] are run for comparison, and are permitted to reach a maximum of 500 iterations or run until the maximum off-diagonal element magnitude falls below  $10^{-6}$ . The intermediate para-Hermitian and paraunitary matrices are truncated by removing the outer lags via a threshold  $\mu_{\text{PH}} = \mu_{\text{PU}} = 10^{-6}$  [13, 30, 91].

The ensemble results in the form of the three metrics are illustrated as box-plots in Fig. 4.7. In Fig. 4.7(a), it can be seen that the reconstruction metric  $\xi_R$  is orders

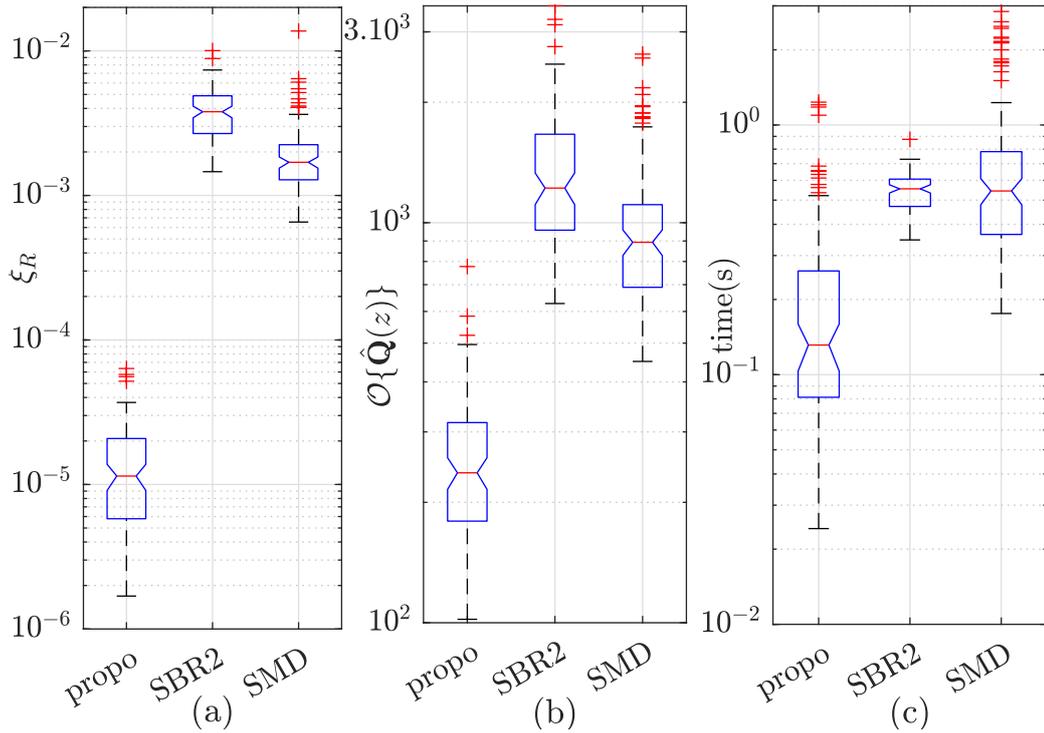


Figure 4.7: Ensemble results illustrated as box-plots for (a) reconstruction error, (b)  $\mathcal{O}\{\hat{Q}(z)\}$ , and (c) execution time. (red marks show outliers)

of magnitude lower for the proposed approach than for SBR2 and SMD. This suggests that the proposed combination of PPM and deflation can compute the PEVD of a para-Hermitian matrix significantly more accurately than both benchmark algorithms. Moreover, in Fig. 4.7(b) the order of the estimated paraunitary  $\hat{Q}(z)$  is lower for the proposed method. This indicates that the perturbation potentially introduced by the deflation process is negligible, as otherwise the order might grow as successive eigenpairs are extracted. The lower order is also significant, since this determines the complexity of implementing the paraunitary  $Q(z)$  for subspace projection-type applications. Lastly, the proposed approach executes faster compared to SBR2 and SMD, as evident from Fig. 4.7(c).

## 4.5 Ordinary Generalised Power Method

In a manner akin to Section 4.2, which delves into the ordinary power method, the generalization of this method is a fundamental concept in the standard matrix algebra. However, in this chapter, it serves as the foundation for the exploration of the polynomial power method's generalization.

The conventional reduced SVD of  $\mathbf{A} \in \mathbb{C}^{M \times L}$  with  $M \geq L$ , given as  $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^H$  with  $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_L] \in \mathbb{C}^{M \times L}$ ,  $\mathbf{\Sigma} \in \mathbb{R}^{L \times L}$  and  $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_L] \in \mathbb{C}^{L \times L}$ , can be obtained through the ordinary power method [1], which is briefly explained in Section 4.2. In order to determine the right-singular vectors, the power method can be applied to  $\mathbf{A}^H\mathbf{A} \in \mathbb{C}^{L \times L}$  as its eigenvectors are in fact the right-singular vectors of  $\mathbf{A}$ . After  $\hat{\mathbf{v}}_i = e^{j\phi}\mathbf{v}_i$ ,  $i = 1, \dots, L$  has been computed, the singular values and left-singular vectors can be obtained via

$$\sigma_i = \|\mathbf{A}\hat{\mathbf{v}}_i\|_2, \Rightarrow \hat{\mathbf{u}}_i = \frac{\mathbf{A}\hat{\mathbf{v}}_i}{\sigma_i} = e^{j\phi}\mathbf{u}_i, \quad i = 1, \dots, M, \quad (4.36)$$

where  $e^{j\phi}$  is an arbitrary phase shift. Note that the phase ambiguity of the left- and right-singular vectors is coupled because the previously extracted right-singular vector is used for computing the left-singular vector.

Alternatively, the left and right singular vectors can be determined by applying the power method to  $\mathbf{A}\mathbf{A}^H$  and  $\mathbf{A}^H\mathbf{A}$ , respectively, and then the corresponding singular value is computed. Since left- and right-singular vectors are determined independently, their phase ambiguities are no longer coupled. Hence, if  $\hat{\mathbf{u}}_1 = e^{j\phi'}\mathbf{u}_1$  and  $\hat{\mathbf{v}}_1 = e^{j\phi}\mathbf{v}_1$ , the resulting estimated singular value will be  $\hat{\sigma}_1 = \hat{\mathbf{u}}_1^H\mathbf{A}\hat{\mathbf{v}}_1 = e^{-j\phi'}\sigma_1e^{j\phi}$  i.e. it will not in general be real-valued. However, real-valuedness, and therefore phase coupling of the left- and right-singular vectors, can be achieved by adjusting the phase of  $\hat{\sigma}_1$ . Such a procedure may be possible in the case of polynomial matrices but may be expensive as will be discussed in Section 4.6. To avoid confusion in the following sections, this alternative approach is termed as second method.

The arguments in the preceding paragraph indicate shows that the power iteration is not restricted to Hermitian matrices, but can indeed be applied to any matrix.

This motivates the need to combine the above concept with the already established polynomial power iteration. The aim is to drop the restriction to para-Hermitian matrices, such that the dominant singular vectors may be computed in first instance, with the option of later performing a full SVD of a polynomial matrix through deflation.

## 4.6 Generalized Polynomial Power Method

This section extends the polynomial power method reviewed in Section 4.3 to generalise the SVD approach summarised in Section 4.5 to the case of a polynomial matrix  $\mathbf{A}(z) : \mathbb{C} \rightarrow \mathbb{C}^{M \times L}$  which has spectrally majorised singular values.

### 4.6.1 Polynomial Iterations Analysis

For an initial  $\mathbf{x}^{(0)}(z) = \mathbf{V}(z)\mathbf{c}(z)$ , the polynomial iteration can be applied to a para-Hermitian matrix  $\mathbf{A}^P(z)\mathbf{A}(z)$  in a similar manner as described in Section 4.3. Thus after  $k$  iterations, with  $z$  substituted by  $e^{j\Omega}$ , it produces

$$\begin{aligned} \mathbf{x}^{(k)}(e^{j\Omega}) &= \mathbf{A}^H(e^{j\Omega})\mathbf{A}(e^{j\Omega})\mathbf{x}^{(k-1)}(e^{j\Omega}) \\ &= \sum_{\ell=1}^L \mathbf{v}_\ell(e^{j\Omega})\sigma_\ell^{2k}(e^{j\Omega})\mathbf{v}_\ell^H(e^{j\Omega})\mathbf{V}(e^{j\Omega})\mathbf{c}(e^{j\Omega}), \end{aligned} \quad (4.37)$$

which can be re-written as

$$\mathbf{x}^{(k)}(e^{j\Omega}) = \sigma_1^{2k}(e^{j\Omega}) \left[ c_1(e^{j\Omega})\mathbf{v}_1(e^{j\Omega}) + \sum_{\ell=2}^L c_\ell(e^{j\Omega}) \left( \frac{\sigma_\ell(e^{j\Omega})}{\sigma_1(e^{j\Omega})} \right)^{2k} \mathbf{v}_\ell(e^{j\Omega}) \right]. \quad (4.38)$$

Since the singular values of  $\mathbf{A}(z)$  are spectrally majorised, so are the eigenvalues of the para-Hermitian matrix  $\mathbf{A}^P(z)\mathbf{A}(z)$  i.e.  $\sigma_\ell^2(e^{j\Omega}) \geq \sigma_{\ell+1}^2(e^{j\Omega})$   $\ell = 1, \dots, L-1$ . Hence  $\mathbf{x}^{(k)}(e^{j\Omega})$  converges to a scaled version of  $\mathbf{v}_1(e^{j\Omega})$  for sufficiently large  $k$  similar to (4.9). Similarly, the normalized vector will be

$$\mathbf{x}_{\text{norm}}^{(k)}(e^{j\Omega}) = \hat{\mathbf{v}}_1(e^{j\Omega}) = \phi_1(e^{j\Omega})\mathbf{v}_1(e^{j\Omega}) \quad \forall \Omega, \quad (4.39)$$

where  $\phi(e^{j\Omega}) = c_1(e^{j\Omega})/|c_1(e^{j\Omega})|$ . Similar to the PPM, this generalised approach also includes truncation and normalization in each iteration which has already been discussed in Sections 4.3.3 and 4.3.2, respectively. Similarly, the problem of singularities in  $c_1(e^{j\Omega})$  can be handled either through regularization or a modification to the initialization if a spectral zero is encountered.

The estimation of the  $s$  and the corresponding left-singular vector is not straightforward and needs careful consideration. Assuming that the first method described in Section 4.5 is adopted i.e. using (4.36), which is to determine the singular value and then the left-singular vector, the frequency dependent version for extracting the dominant singular value will be

$$\hat{\sigma}_m(e^{j\Omega}) = \|\mathbf{A}(e^{j\Omega})\hat{\mathbf{v}}_m(e^{j\Omega})\|_2, \quad m = 1, \dots, N. \quad (4.40)$$

This forces  $\sigma_m(e^{j\Omega})$  to be positive  $\forall \Omega \in \mathbb{R}$  due to the norm operator whereas the theory behind the analytic decomposition existence shows that the singular value can be negative on the unit-circle [58]. Forcing the singular values to be positive violates this condition, thus with this method, the obtained decomposition might differ from the decomposition given in [10] and it would also not be analytic in case the ground-truth is negative. Alternatively, if the matrix  $\mathbf{A}(z)$  is known to be positive semi-definite, the singular values are guaranteed to be real and positive and so this method gives the correct decomposition. Once the singular value is obtained with an acceptable accuracy, which we discuss further below, via (4.40), the dominant left-singular vector can be obtained as

$$\tilde{\mathbf{u}}_1(e^{j\Omega}) = \mathbf{A}(e^{j\Omega})\tilde{\mathbf{v}}_1(e^{j\Omega})/\hat{\sigma}_1(e^{j\Omega}) = \phi_1(e^{j\Omega})\mathbf{u}_1(e^{j\Omega}). \quad (4.41)$$

The allpass factor  $\phi_1(e^{j\Omega})$  is the same as that of the right-singular vector, such that their ambiguities are coupled. This coupling results in the singular value being real-valued on the unit-circle. Both (4.40) and (4.41) can be implemented in the DFT domain. Adjusting the size of this DFT is discussed further below.

The second method, described in Section 4.5, which motivates to determine the

left- and right-singular vectors by applying the polynomial power method independently to  $\mathbf{A}(z)\mathbf{A}^P(z)$  and  $\mathbf{A}^P(z)\mathbf{A}(z)$ , respectively, and then computes the singular value as  $\hat{\sigma}_1(z) = \tilde{\mathbf{u}}_1^P(z)\mathbf{A}(z)\tilde{\mathbf{v}}_1(z)$ . This method does not impose the condition of singular values being positive on the unit circle, and so it can allow the analytic decomposition proven in [10] to be achievable for any  $\mathbf{A}(z)$ . However, to retain real-valuedness for the singular values on the unit circle, the left- and right-singular vector have to have a common allpass factor. Thus if both the left- and right-singular vectors are independently extracted by applying the polynomial power method to  $\mathbf{A}^P(z)\mathbf{A}(z)$  and  $\mathbf{A}(z)\mathbf{A}^P(z)$ , respectively, the allpass factor in  $\tilde{\mathbf{u}}_1(z)$  and  $\tilde{\mathbf{v}}_1(z)$  will, in general, not be coupled. Hence, the second method may not be desirable to be used unless a common phase shift can be found. This coupling can be regained in the DFT domain by adjusting the phase until singular values are real in each bin. However, this method is still avoided as it invokes two PPM calculations which is computationally more expensive compared to the first approach.

To optimize the computational efficiency of the first approach, the right singular vector is computed via the PPM in case of  $M \geq L$ . For the case  $M \leq L$ , all of the above approaches can be used instead to factorise  $\mathbf{A}^P(z)$ .

## 4.6.2 Sufficient DFT Size

### Dominant Singular Value

Once  $\mathbf{v}_1(z)$  is determined with Algorithm 1, the singular value can be determined via (4.40) in the DFT domain. To determine a sufficient DFT size, time-domain aliasing can be utilized [35]. Thus (4.40) can be evaluated at increasing DFT sizes until

$$\zeta_\sigma = \frac{\sum_\tau |\hat{\sigma}_1^{(K)}[\tau] - \hat{\sigma}_1^{(K/2)}[\tau]|^2}{\sum_\tau |\hat{\sigma}_1^{(K)}[\tau]|^2}, \quad (4.42)$$

where  $\hat{\sigma}_1^{(K)}[\tau]$  represents the time-domain equivalent of (4.40) obtained with a  $K$ -point inverse DFT, falls below a certain low threshold  $\varepsilon_1$ . A small value of  $\zeta_\sigma$  indicates that  $K/2$  can be considered sufficient to approximate the dominant singular because the difference between the singular value estimate at size  $K/2$  and at size  $K$  will be small.

Otherwise, they will differ, and the singular value estimated with  $K/2$  DFT size will not just be a truncated version of the estimate at DFT size  $K$ , but will also suffer from time domain aliasing due to the inverse DFT. Hence, then  $\zeta_\sigma$  will be significant. Once  $\tilde{\mathbf{v}}_i(z)$  is extracted with satisfactory accuracy, a DFT size of  $K = \mathcal{O}\{\mathbf{A}(z)\tilde{\mathbf{v}}_i(z)\} + 1$  should generally suffice.

### Left Singular Vector

Similarly, to determine a sufficient DFT size for (4.41), time-domain aliasing may be captured via the error w.r.t. normality in the time-domain as

$$\zeta_u = \sum_{\tau} |\hat{\mathbf{u}}_1^H[-\tau] * \hat{\mathbf{u}}_1[\tau] - \delta[\tau]|_2^2, \quad \tau \in \mathbb{Z}. \quad (4.43)$$

A similar criterion has been utilised for the DFT size in [34]. There, it is shown as a necessary criterion; while sufficiency has not been proven, in practise it has generally been shown to suffice in all simulations.

It follows that for a sufficient DFT size,  $\zeta_u$  will be small since  $\hat{\mathbf{u}}_1[\tau]$  should be normal. Thus (4.41) is implemented at increasing DFT size until  $\zeta_u$  falls below a some given threshold  $\varepsilon_u$ .

### 4.6.3 Simulations and Results

**Example 6.** To demonstrate the potential of the generalized polynomial power method, a simple case of  $\mathbf{A}(z)$  is assumed whose ground-truth SVD is known such that  $\Sigma(z) \in \mathbb{C}^{3 \times 2}$  contains

$$\sigma_1(z) = \frac{1}{2}z + 4 + \frac{1}{2}z^{-1}, \quad (4.44)$$

$$\sigma_2(z) = \frac{1}{4}z + 1 + \frac{1}{4}z^{-1}, \quad (4.45)$$

which are spectrally majorised. The left-singular vectors matrix  $\mathbf{U}(z)$  constructed via elementary paraunitary operations given as [36]

$$\mathbf{U}(z) = \prod_{i=1}^2 \{\mathbf{I} - (1 - z^{-1})\mathbf{e}_i\mathbf{e}_i^H\}, \quad (4.46)$$

where  $\mathbf{e}_{i=1,2} = [1, 1, \mp 1]^T/\sqrt{3} \in \mathbb{C}^3$  are unit-norm vectors. The right-singular vectors in  $\mathbf{V}(z) \in \mathbb{C}^{2 \times 2}$  of order 2 are generated by the same approach with  $\mathbf{e}_1 = [1, -1]^T/\sqrt{2}$  and  $\mathbf{e}_2 = [-1, 0]^T$ . The polynomial matrix  $\mathbf{A}(z)$  is then defined as  $\mathbf{U}(z)\mathbf{\Sigma}(z)\mathbf{V}^P(z)$ .

Algorithm 1 is executed with  $\epsilon = 10^{-12}$ ,  $k_{\max} = 10^3$ ,  $\mathbf{R}(z) = \mathbf{A}^P(z)\mathbf{A}(z)$  and  $\mathbf{v}^{(0)}(z) = \mathbf{x}^{(0)}(z) = 1$ . The truncation method employed is the order limitation described in Section 4.3.3 where the order of  $\mathbf{x}_{\text{norm}}^{(k)}(z)$  is limited to the estimated support obtained from [86]. Algorithm 1 converges in 44 iterations resulting in  $\zeta_v = 1.4 \times 10^{-11}$ . Once the left-singular vector is estimated, the corresponding singular value is estimated via (4.40). With  $K = 16$ , the time-domain aliasing  $\zeta_\sigma = 8 \times 10^{-28}$ . The trailing coefficients of  $\hat{\sigma}_1[\tau]$  are truncated on either side of  $\tau = 0$  via a threshold of  $10^{-10}$ . This results in a order of 6 whereas the ground-truth singular value has an order of 2. The coefficients are illustrated in Fig. 4.8(a) where the coefficients at  $|\tau| \leq 1$  exactly match the ground-truth coefficients in (4.44), whereas the coefficients at  $|\tau| > 1$  are smaller than  $10^{-5}$ . The normalized squared difference between the estimated and the ground-truth singular value, which can be defined similar to (4.42) as

$$\xi_\sigma = \frac{\sum_\tau |\sigma_1[\tau] - \hat{\sigma}_1[\tau]|^2}{\sum_\tau |\sigma_1[\tau]|^2}, \quad (4.47)$$

is  $3.5 \times 10^{-12}$ . The corresponding left-singular vector is then obtained from (4.41) with a DFT of size  $K = 16$ . Thereafter, the order is limited by a shifted truncation to 3, which achieves a metric of  $\zeta_v = 9 \times 10^{-12}$ .

The GSBR2 algorithm [11] is executed with  $\mu_{\text{PU}} = 10^{-5}$ ,  $\epsilon = 10^{-5}$  and  $\mu_{\text{PH}} = 10^{-5}$  for 1000 iterations and results in  $\zeta_v = 9.6 \times 10^{-5}$ ,  $\zeta_u = 1.5 \times 10^{-5}$  and  $\xi_\sigma = 2.7 \times 10^{-2}$ . The dominant singular value estimated with the GSBR2 has order 8 whose coefficients are illustrated in Fig. 4.8(b). It is evident that the estimated singular value is neither conjugate symmetric and nor the coefficients match the ground truth coefficients except

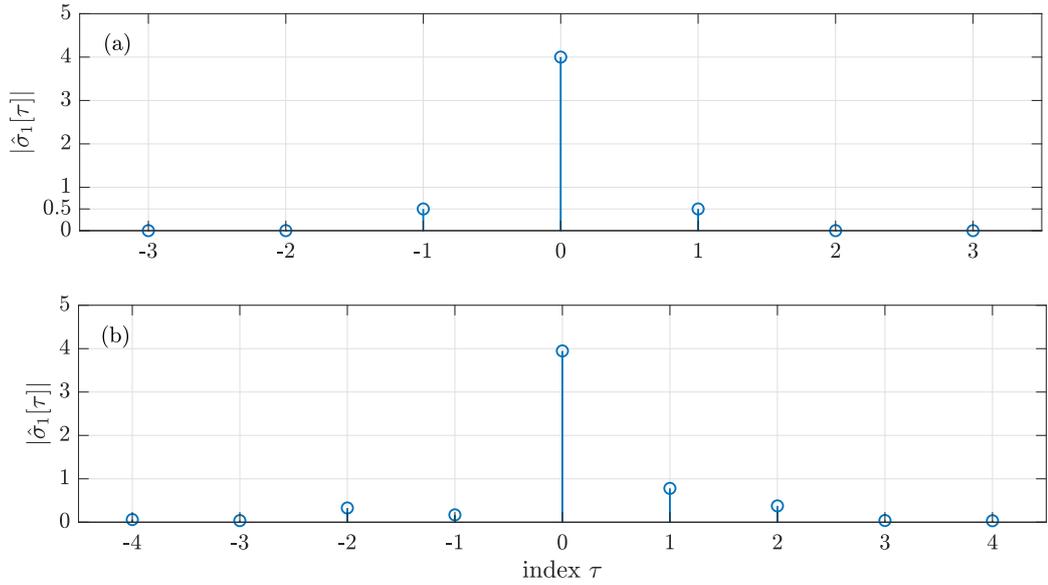


Figure 4.8: (a) GPPM, and (b) GSR2 [11] based estimated dominant singular value coefficients for the numerical example.

at  $\tau = 0$  where  $\hat{\sigma}[0] = 3.95 \approx 4$ . This loss of conjugate symmetry may be the cause of a large value of  $\xi_\sigma$ .  $\triangle$

### Ensemble Test

In a more extensive test, the proposed method is evaluated against the GSR2 algorithm via an ensemble consisting of 500 randomised instantiations of  $\mathbf{A}(z) \in \mathbb{C}^{3 \times 2}$  such that each instance has  $\mathcal{O}\{\mathbf{U}(z)\} = \mathcal{O}\{\mathbf{V}(z)\} = 10$  and  $\mathcal{O}\{\boldsymbol{\Sigma}(z)\} = 20$ . All the instantiations have spectrally majorised singular values.

For the proposed method, Algorithm 1 is simulated with  $\epsilon = 10^{-10}$ ,  $k_{\max} = 10^3$ ,  $\mathbf{R}(z) = \mathbf{A}^P(z)\mathbf{A}(z)$  and  $\mathbf{x}^{(0)}(z) = 1$ . The order of the product vector is limited to 10, with its order estimated through the method in [86], followed by shifted-truncation [27, 39]. The corresponding singular value and the left-singular vector are extracted at  $K = 2^{\lceil \log_2(\mathcal{O}\{\mathbf{A}(z)\tilde{\mathbf{u}}_1(z)\}) \rceil}$  where  $\lceil \cdot \rceil$  denotes ceiling operation. The trailing coefficients of the estimated left-singular vectors are truncated below a threshold of  $10^{-10}$  while the right-singular vector is similarly order-limited to its estimated support. The GSR2 is simulated with  $\mu_{\text{PU}} = 10^{-4}$ ,  $\mu_{\text{PH}} = 2 \times 10^{-10}$  employing the original

Table 4.1: Performance Comparison of GSR2 and GPPM

<b>Metrics</b>	<b>GSR2</b>	<b>GPPM</b>
$\mathcal{O}\{\tilde{\mathbf{u}}_1(z)\}$	$966 \pm 185$	10
$\mathcal{O}\{\tilde{\mathbf{v}}_1(z)\}$	$422 \pm 126$	10
$\mathcal{O}\{\hat{\sigma}_1(z)\}$	$96 \pm 38$	$57 \pm 4$
$\zeta_v$	$(1.2 \pm 0.8) \times 10^{-3}$	$(5.5 \pm 4.5) \times 10^{-5}$
$\zeta_u$	$(1.6 \pm 0.85) \times 10^{-3}$	$(5.5 \pm 4.5) \times 10^{-5}$
$\xi_\sigma$	$0.09 \pm 0.07$	$(1.5 \pm 1.3) \times 10^{-5}$
time(s)	$0.67 \pm 0.15$	$0.44 \pm 0.19$

truncation method of SBR2/SMD [13, 22]. The algorithm is allowed to perform a maximum of 200 iteration; however, the execution is terminated if the off-diagonal terms fall below  $10^{-6}$ .

The ensemble average for all the metrics is shown in Table 4.1. It is evident that the proposed method provides a more compact order approximation for both the left- and right-singular vectors and the singular value compared to the GSR2. Moreover, the error metrics  $\zeta_u$  and  $\zeta_v$  of the proposed method's extracted singular vectors reach orders of magnitude below those obtained with GSR2. Likewise, the normalized squared difference between the estimated and ground-truth singular value is orders of magnitude lower for the polynomial method than GSR2. The potential reason for the large deviation of the GSR2's estimated singular value might be the imperfect conjugate symmetry. Also, the SBR2-type algorithms are known to only achieve a relatively poor diagonalisation compared to their DFT-domain counterparts in e.g. [34, 35].

## 4.7 Summary

This chapter first introduced the polynomial equivalent of the well known power method for the extraction of the dominant eigenpair of a spectrally majorised para-Hermitian. Since the repeated multiplication of a polynomial matrix with a polynomial vector increases the order of the product vector, two methods of truncation are demonstrated to limit the order growth. Proposed polynomial extension has shown superior performance over benchmark algorithms.

Secondly the approach of combining the polynomial power method with deflation for

the PEVD of a low-rank para-Hermitian polynomial matrix has been presented. It has been shown that it is possible for almost all para-Hermitian matrices to apply deflation similar to the approach for ordinary matrices. The perturbation of the eigenpairs of the deflated matrix has been studied and has been shown to relate directly to the accuracy of the successively extracted eigenpairs. Over an ensemble of low-rank para-Hermitian matrices, the proposed method has outperformed state-of-the-art algorithms in terms of accuracy, speed, and implementation complexity.

Lastly, the polynomial power method, which was initially proposed for para-Hermitian matrices, is extended into the generalized polynomial power method for computing the dominant left- and right-singular vectors and their corresponding singular value of a polynomial matrix. This extension provides better estimation of the singular vectors with lower order approximation as compared to the only direct PSVD algorithm based on the Kogbetliantz method. The proposed method promises better results and can be further utilized to compute the PSVD of a low-rank polynomial matrix through the polynomial matrix deflation analogous to a low-rank PEVD.

With the proposed novel method in form of an extension of the power method, the problem of a large number of iterations still needs to be addressed, as otherwise this makes the polynomial power method computationally expensive. Due to this reason only a low-rank PEVD can be accomplished through the polynomial power method if combined with a deflation approach. Therefore, this method is further improved in the next chapter by reducing the number of iterations to one by decomposing the given polynomial matrix into sum of rank one matrices.

## Chapter 5

# Unified Algorithm I: Rank One Decomposition of a Polynomial Matrix

### 5.1 Introduction

While the previous chapter introduced the polynomial and generalized polynomial power method, this chapter builds upon those findings to achieve further improvements. The primary goal is to reduce the computational cost by minimizing the number of iterations. To accomplish this, the key idea is to decompose any given polynomial matrix into rank-one polynomial matrices. This is made possible by leveraging the fact that analytic EVD and SVD exist for an analytic para-Hermitian and general polynomial matrix, respectively, as discussed in Chapter 2.

A rank-one polynomial matrix inherently possesses a single non-zero eigenvalue or singular value, making it feasible to achieve convergence within a single polynomial power iteration. This facilitates the efficient computation of the PEVD and PSVD of para-Hermitian and general polynomial matrices, respectively. While the polynomial power method and its generalization iteratively reduce the order of the resulting vector to control the order growth, the resulting decomposition order, on average, remains relatively compact. However, in cases where only a single iteration is performed,

achieving order reduction may not be feasible. While this limitation is noteworthy, the advantages is the independent extraction of each eigenpair or singular value and its corresponding vectors. This approach offers the potential for improved accuracy and execution time.

In addressing the singularity issue that arises when normalizing a product vector  $\mathbf{x}^{(k)}(z)$ , which involves the use of a regularization parameter (see (4.11)), it becomes evident that this problem becomes particularly pronounced when only a single iteration is performed. Additionally, it is observed that the initial vector  $\mathbf{x}^{(0)}(z)$  may sometimes lack components in the direction of the desired eigenvector or singular vector. To overcome the former issue, a solution similar to the one proposed in [10], which involves employing upsampling and sign changes, can be adopted with some additional modifications. In terms of the latter issue, one potential approach is to restrict the choices for initialization, thereby ensuring that the problem does not arise on the unit circle or within the region of convergence.

In the previous chapter, the emphasis was solely on PEVD and PSVD. In the current chapter, it is evident from the algorithm's name that all three polynomial matrix decomposition methods are covered, with rank decomposition serving as the foundational principle. To apply this methodology for computing the PQRD of a polynomial matrix, the rank-one matrix terms can be estimated using QR decompositions in the sample points of a polynomial matrix evaluated on the unit circle. When the columns of these terms are normalized, they yield a paraunitary matrix capable of transforming a polynomial matrix into an upper triangular form.

The chapter is structured as follows: Section 5.2 provides an overview of rank one decomposition of a para-Hermitian matrix for a quick and accurate PEVD. Its simplified version, referred to in this thesis as normalization-free, is explained in Section 5.3. Similarly, Section 5.4 outlines the PSVD while Section 5.6 presents the PQRD of polynomial matrices achieved through rank-one decompositions. Their respective normalization-free variants are introduced in Section 5.5 and Section 5.7, respectively. Simulation results are discussed in Section 5.8, and a summary is provided in Section 5.9.

## 5.2 PEVD by Rank One Decomposition

### 5.2.1 Objective and Rationale

The ordinary power method described in Section 4.2 will converge in a single iteration provided that the Hermitian matrix is rank one. Therefore, if a rank one decomposition is available for a Hermitian matrix, an EVD can be easily be performed through a single iteration of the power method. However, if the given Hermitian matrix is already rank deficient i.e.  $p < M$ , then after the first  $p$  dominant eigenpairs have been extracted, the remaining  $M - p$  eigenvectors can be extracted through Gram-Schmidt orthogonalization [1].

Similarly, if any para-Hermitian polynomial matrix can effortlessly be decomposed into sum of rank one matrices, a single iteration of the polynomial power method will suffice to extract any eigenpair. However, the case where  $p < M$ , the Gram-Schmidt orthogonalization procedure will have to be extended to the polynomial domain.

### 5.2.2 Para-Hermitian Matrix Rank One Decomposition

For a rank  $p$  para-Hermitian matrix, its rank one terms given in (4.22) can be estimated via a DFT domain method without computing its PEVD. This method assumes that  $\mathbf{R}(z)$  is estimated from finite data and therefore is spectrally majorised. The EVD of sample points of  $\mathbf{R}(z)$  along the unit circle i.e.  $\mathbf{R}_k = \mathbf{R}(e^{j\Omega_k})$  where  $\Omega_k = \frac{2\pi k}{K}$ , is given as

$$\mathbf{R}_k = \mathbf{Q}_k \mathbf{\Lambda}_k \mathbf{Q}_k^H, \quad \text{for } k = 1, \dots, K, \quad (5.1)$$

where  $\mathbf{\Lambda}_k = \text{diag}\{\lambda_{1,k}, \dots, \lambda_{M,k}\}$  where  $\lambda_{m,k} \geq \lambda_{m+1,k}$  for  $m = 1, \dots, M - 1$  and  $\mathbf{Q}_k = [\mathbf{q}_{1,k}, \dots, \mathbf{q}_{M,k}]$  denotes the  $k$ th bin eigenvectors. The rank one decomposition of  $\mathbf{R}_k$  similar to (4.19) can be given as

$$\mathbf{R}_k = \sum_{m=1}^M \mathbf{R}_{m,k}, \quad \text{where } \mathbf{R}_{m,k} = \mathbf{q}_{m,k} \lambda_{m,k} \mathbf{q}_{m,k}^H. \quad (5.2)$$

For an estimated para-Hermitian matrix, the sample points of its analytic PEVD in (2.5) relate to its bin-wise EVD in (5.1) as

$$\mathbf{A}(e^{j\Omega_k}) = \mathbf{\Lambda}_k \quad (5.3)$$

$$\mathbf{Q}(e^{j\Omega_k}) = \mathbf{\Phi}_k \mathbf{Q}_k, \quad (5.4)$$

where  $\mathbf{\Phi}_k = \text{diag}\{e^{j\phi_{1,k}}, \dots, e^{j\phi_{M,k}}\}$  is a diagonal phase matrix.

Now the objective is to find the relation between  $\mathbf{R}_{m,k}$  and the sample points of  $\mathbf{R}_m(z)$  on the unit circle at the same frequency points i.e.  $\mathbf{R}(e^{j\Omega_k})$  so that  $\mathbf{R}_m(z)$  can be estimated through  $\mathbf{R}_{m,k}$  available from (5.2). To establish this relation, the relations in (5.3) and (5.4) can be used, so it leads to

$$\begin{aligned} \mathbf{R}_m(e^{j\Omega_k}) &= \mathbf{q}_m(e^{j\Omega_k}) \mathbf{q}_m^H(e^{j\Omega_k}) \lambda(e^{j\Omega_k}) = e^{j\phi_{m,k}} \mathbf{q}_{m,k} e^{-j\phi_{1,k}} \mathbf{q}_{m,k}^H \lambda_{m,k} \\ &= \mathbf{q}_{m,k} \mathbf{q}_{m,k}^H \lambda_{m,k} = \mathbf{R}_{m,k}. \end{aligned}$$

This shows that sample points of  $\mathbf{R}_m(z)$  on the unit circle are in-fact the bin-wise rank one terms  $\mathbf{R}_{m,k}$  obtained via the conventional EVD in the DFT bins. Furthermore, the above analysis also shows that phase smoothing — the costly process of determining  $\mathbf{\Phi}[k]$  [33, 47] — is not required for the estimation of rank one terms as it is eliminated automatically.

With access to the sample points of  $\mathbf{R}_m(z)$  along the unit circle, the estimate of  $\mathbf{R}_m[\tau] \bullet \rightarrow \mathbf{R}_m(z)$  can be derived from  $\mathbf{R}_m[k]$  using the inverse DFT (IDFT) over a suitably extensive range of  $K$ . It is important to highlight that  $\mathbf{R}_m[\tau]$  is not required to possess an identical support as  $\mathbf{R}[\tau] \bullet \rightarrow \mathbf{R}(z)$ . As a result, the determination of an appropriate value for  $K$  becomes essential in order to extract rank one matrices from their equivalent in the DFT domain. The time-domain equivalent of  $[\mathbf{R}_m[1], \dots, \mathbf{R}_m[K]]$ , obtained via the IDFT, is denoted as  $\tilde{\mathbf{R}}_m[\tau]$ , the estimate of  $\mathbf{R}_m[\tau]$ . This estimate will be periodic as

$$\tilde{\mathbf{R}}_m[\tau] = \sum_{\mu=-\infty}^{\infty} \mathbf{R}_m[\tau - K\mu] \quad (5.5)$$

If  $K$  is shorter than the support of  $\mathbf{R}_m[\tau]$ , then (5.5) would have time-domain aliasing. Moreover,  $\mathbf{R}_m(z)$  can be an infinite Laurent series however due to analyticity, the corresponding  $\mathbf{R}_m[\tau]$  would converge absolutely which permits it to be approximated with a finite order Laurent polynomial [35]. In order to determine a sufficient value for  $K$  that exceeds the optimum finite support, first it is important to isolate the fundamental period of (5.5) through a rectangular window  $p_K[\tau]$  of size  $K - 1$  centred at  $\tau = 0$

$$p_K[\tau] = \begin{cases} 1 & |\tau| \leq \frac{K-1}{2}, \text{ odd } K \\ 0 & \text{otherwise} \end{cases} \quad (5.6)$$

as

$$\mathbf{R}_m^{(K)}[\tau] = p_K[\tau] \tilde{\mathbf{R}}_m[\tau]. \quad (5.7)$$

For determining a value for  $K$  which results in minimum time-domain aliasing in (5.5), the DFT size  $K$  can be increased by a factor of two until  $\gamma_R$ , defined as

$$\gamma_R = \frac{\sum_{|\tau| \geq K/2} \|\mathbf{R}_m^{(K)}[\tau] - \mathbf{R}_m^{(K/2)}[\tau]\|_{\mathbb{F}}^2}{\sum_{\tau} \|\mathbf{R}_m^{(K)}[\tau]\|_{\mathbb{F}}^2}, \quad (5.8)$$

to measure the normalized time-domain aliasing, falls below a certain suitably selected threshold  $\epsilon_{\gamma_R}$ . Once the desired  $K$  is reached, the required estimate of  $\mathbf{R}_m[\tau]$  will be  $\hat{\mathbf{R}}_m[\tau] = \mathbf{R}_m^{(K/2)}[\tau]$  and so (4.22) can be re-written as

$$\mathbf{R}(z) \approx \sum_{m=1}^M \tilde{\mathbf{R}}_m(z) \text{ with } \tilde{\mathbf{R}}_m(z) = \tilde{\mathbf{q}}_m(z) \tilde{\lambda}_m(z) \tilde{\mathbf{q}}_m^{\text{P}}(z), \quad (5.9)$$

with  $\tilde{\mathbf{q}}_m(z)$  and  $\tilde{\lambda}_m(z)$  being the accurate estimates of  $\mathbf{q}_m(z)$ , and  $\lambda(z)$ , respectively. This thesis assumes that  $\tilde{\mathbf{R}}_m(z)$  and  $\tilde{\mathbf{q}}_m(z)$  has shortest time-domain support.

### 5.2.3 Single Polynomial Power Iteration

With access to the rank one estimate, a single iteration of the polynomial power method can now be applied. Analogous to the ordinary power method, the single polynomial

iteration can be applied to a rank one para-Hermitian matrix  $\tilde{\mathbf{R}}_m(z)$  to obtain  $\hat{\mathbf{q}}_m(z)$ . For an initial  $\mathbf{x}^{(0)}(z) = \mathbf{Q}(z)\mathbf{c}(z)$ , the single iteration results in

$$\begin{aligned}\mathbf{x}^{(1)}(e^{j\Omega}) &= \tilde{\mathbf{R}}_m(e^{j\Omega})\mathbf{x}^{(0)}(e^{j\Omega}) \\ &= \tilde{\mathbf{q}}_m(e^{j\Omega})\tilde{\lambda}_m(e^{j\Omega})\tilde{\mathbf{q}}_m^H(e^{j\Omega})\mathbf{Q}(e^{j\Omega})\mathbf{c}(e^{j\Omega}).\end{aligned}\quad (5.10)$$

Since  $\tilde{\mathbf{q}}_m^H(e^{j\Omega})\mathbf{q}_i(e^{j\Omega}) = k'\delta[m-i] \forall \Omega$  with  $k' \approx 1$ , (5.10) reduces to

$$\mathbf{x}^{(1)}(e^{j\Omega}) = \tilde{\mathbf{q}}_m(e^{j\Omega})\tilde{\lambda}_m(e^{j\Omega})c_m(e^{j\Omega}), \quad (5.11)$$

which shows that  $\mathbf{x}^{(1)}(e^{j\Omega})$  is a scaled version of the  $m$ th eigenvector of  $\mathbf{R}(z)$  or the dominant eigenvector of  $\tilde{\mathbf{R}}_m(z)$ . Once normalized, the eigenvalues can similarly be determined through the Rayleigh quotient in (4.12).

#### 5.2.4 Normalization

After normalizing (5.11) to unit norm  $\forall \Omega$ , discussed in Section 4.3, it produces

$$\begin{aligned}\mathbf{x}_{\text{norm}}^{(1)}(e^{j\Omega}) &= \frac{\mathbf{x}^{(1)}(e^{j\Omega})}{\sqrt{\mathbf{x}^{(1),H}(e^{j\Omega})\mathbf{x}^{(1)}(e^{j\Omega})}} = \frac{c_m(e^{j\Omega})\tilde{\mathbf{q}}_m(e^{j\Omega})}{|c_m(e^{j\Omega})|} \\ &= \varphi_m(e^{j\Omega})\tilde{\mathbf{q}}_m(e^{j\Omega}), \quad \text{for } k = 1, \dots, K,\end{aligned}\quad (5.12)$$

where the factor  $\varphi_m(e^{j\Omega}) = c_m(e^{j\Omega})/|c_m(e^{j\Omega})|$  is an allpass factor. This all-pass factor often complicates the compact order approximation of the eigenvector and generally, this is the reason why phase smoothing [33, 34, 47] is required. If this phase factor is a simple delay,  $\mathbf{x}_{\text{norm}}^{(1)}(z)$  will have the shortest possible order as  $\tilde{\mathbf{q}}_m(z)$  itself has shortest support. During the normalization procedure, two potential challenges may arise: Firstly, we need to find a sufficiently large DFT size to perform normalize or implement (5.12). This issue has already been discussed in Section 4.6.

Secondly,  $c_m(e^{j\Omega})$  might have a finite number of singularities at distinct frequencies which will lead to division by zero or  $c_m(z) = 0 \forall z$ . To resolve this, the regularisation approach proposed in Section 4.3 must be avoided, because here only a single iteration is performed instead of multiple iterations. Moreover, if regularization is still used, the

resulting normalized vector may not represent a valid eigenvector. Therefore, this issue can be avoided by a suitable initialization as described in following subsections.

### 5.2.5 Choices and Issues in Initialization

To avoid singularities and accurately estimate the eigenvector, it is important to choose an appropriate initialization for  $\mathbf{x}^{(0)}(z)$ . However, initializing the vector can sometimes lead to singularities for certain values of  $z$ , or even result in  $c_m(z) = 0 \forall z$ . To solve the problem of  $c_m(z) = 0 \forall z$ , one potential solution is to restrict the initialization to any of the  $M$  columns of the rank one matrix. For example, if  $\mathbf{x}^{(0)}(z)$  is initialized with the  $n$ th column of  $\tilde{\mathbf{R}}_m(z)$  i.e.  $\mathbf{x}^{(0)}(z) = \tilde{\mathbf{q}}_m(z)\tilde{\lambda}_m(z)\tilde{\mathbf{q}}_{m,n}^P(z)$  where  $c_m(z) = \tilde{\lambda}_m(z)\tilde{\mathbf{q}}_{m,n}^P(z)$ , this prevents the issue of  $c_m(z) = 0 \forall z$ . However,  $c_m(z)$  may have a finite number of zeros at distinct frequencies on the unit circle which can be avoided through the method discussed in next subsection.

Unfortunately, determining the optimum order of the initialization vector  $\mathbf{x}^{(0)}(z)$  and how it affects the order of the estimated eigenvector, particularly the order of  $\varphi_m(z)$ , can be difficult. Ideally, the initialization should result in  $\varphi_m(z)$  being a simple delay, but this is unlikely to happen with random initialization. Therefore, the only approach is to experiment with different initializations and select the one resulting in the lowest possible order for the estimated eigenvector.

### 5.2.6 Solution to Normalization Issue

#### Types of Zero Crossings

Due to the potential presence of zero crossings in  $c_m(e^{j\Omega})$ , the phase function  $\varphi(e^{j\Omega})$  in (5.12) may experience phase discontinuities. Consequently, achieving an approximation of the normalized vector would necessitate an infinitely long DFT size. The zero crossings of  $c_m(e^{j\Omega})$  across which the phase is continuous do not cause any problem in the process of normalization. This type of situation is experienced when the multiplicity of a zero is even. As in such case, the  $c_m(e^{j\Omega})$  can easily be factored into an analytic phase times a magnitude function. Therefore, one needs to carry out the normalization given in (5.12) at unevenly spaced frequency points on the unit circle or by slightly

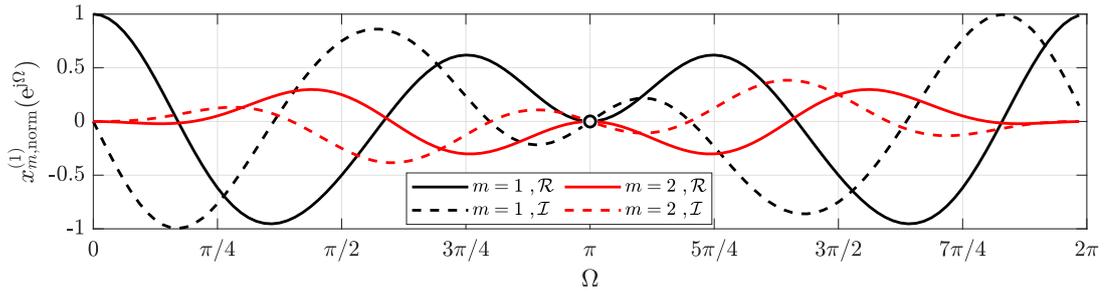


Figure 5.1: Example 7 illustration showing the smooth evolution of both the real and imaginary parts of the components of the normalized vector with  $c_m(e^{j\Omega})$  with zero crossing with no phase jump across it.

shifting the DFT bins by a small shift such that none of the bins coincide with zero crossings. To elaborate more on this, let us consider a simple example.

**Example 7.** Let us assume  $\mathbf{x}^{(1)}(z) = (1 + 2z^{-1} + z^{-2})\mathbf{q}_1(z)$  with  $\mathbf{q}_1(z) \in \mathbb{C}^3$  is considered from Section 4.3.5. Since  $\angle c_m(e^{j(\pi+\epsilon)}) - \angle c_m(e^{j(\pi-\epsilon)}) \approx 2n\pi$  where  $n \in \mathbb{Z}$ , which is not a phase discontinuity instead a phase wrap, this does not cause any issue in the smooth evolution of both the real and imaginary components of  $x_{m,norm}^{(1)}(e^{j\Omega})$ ,  $m = 1, 2$ , as illustrated in Fig. 5.1 except at  $z = e^{j\pi}$ . Therefore, after the normalization has been performed in the DFT domain with a DFT size of 64 such that none of the bins has a zero crossings, the resulting  $\hat{\mathbf{x}}_{norm}^{(1)}[\tau] = \hat{\mathbf{q}}_1[\tau]$  exhibits time domain aliasing of  $\zeta_q = 2.6 \times 10^{-30}$ . This estimated eigenvector extracts the dominant eigenvalue using the Rayleigh quotient, defined in (4.12), at a normalized error of  $\xi_\lambda = 2.3 \times 10^{-8}$ .  $\triangle$

With Example 7, it is evident that not all zero crossings pose issues. However, in scenarios where  $c_m(e^{j\Omega})$  exhibits zero crossings across which there are phase discontinuities, the solution can be complex depending upon the number of zero crossings. Let's first address the straightforward situation where the number of zero crossings is even. In such cases, an even count of phase discontinuities ( $\pi$  radians) at the zero crossing locations can be effectively compensated by applying an additional  $\pi$  phase. This facilitates the analytic factorization of  $c_m(e^{j\Omega})$  into an allpass and a real-valued function. This method shares similarities with the approach outlined in [10], designed for computing the analytic SVD existence of a polynomial matrix. For a clearer grasp, let us delve into an illustrative example.

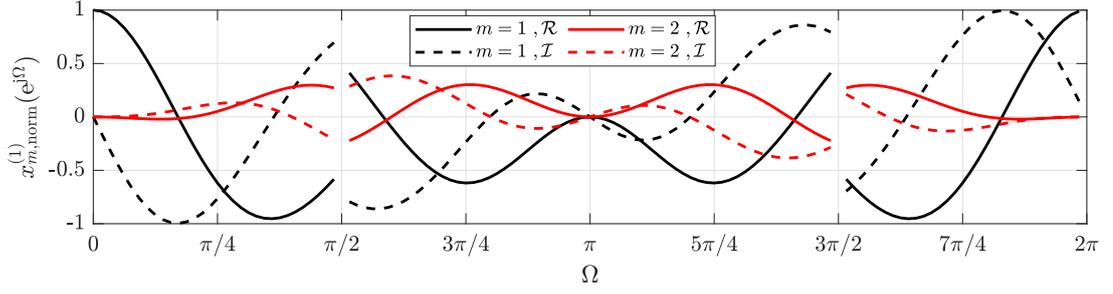


Figure 5.2: Normalization results for  $\mathbf{x}^{(1)}(z)$  on the unit circle in presence of even number of zeros of  $c_m(e^{j\Omega})$ : given case illustrates 2 zero crossings.

**Example 8.** We consider the setting of Example 7 except with  $c_m(e^{j\Omega})$  set to  $(1 + z^{-2})$ . It has two zero crossings at  $z = \{e^{j\pi/2}, e^{j3\pi/2}\}$  both associated with a  $\pi$  phase discontinuity. For illustration, the real and imaginary components of the normalized vector  $\mathbf{x}_{\text{norm}}^{(1)}(e^{j\Omega})$  is shown in Fig. 5.2. It can be seen that both real and imaginary components of the two drawn components have discontinuities across the zero crossings at  $\pi/2$  and  $3\pi/2$ . In addition, if the sections of all curves shown in Fig. 5.2 between these two zero crossings are inverted i.e. multiplied by a negative sign, the resulting curves becomes smooth as illustrated in Fig. 5.3. The reason is that these zero crossings had  $\pi$  phase discontinuities which can be compensated by applying an extra  $\pi$  phase to either the sections between the zero crossings or before and after the zero crossings. Now except at zero crossings frequency points, both the real and imaginary part of all components of  $\mathbf{x}_{\text{norm}}^{(1)}(e^{j\Omega})$  evolves smoothly. Therefore, the normalized  $\mathbf{x}_{\text{norm}}^{(1)}[\tau] = \hat{\mathbf{q}}_1[\tau]$  is obtained via an IFFT after performing inversion, and the resulting time-domain aliasing is  $1.54 \times 10^{-31}$  which estimates the dominant eigenvalue through the Rayleigh quotient with  $\xi_\lambda = 2.3 \times 10^{-8}$ .  $\triangle$

Example 8 demonstrated a simple case of two zero crossings. In case of  $N_z$  zero crossings where  $N_z$  is even, and all of these zero crossings are associated with phase discontinuities, the inversion or application of additional  $\pi$  phase to compensate for the  $\pi$  phase discontinuities have to be carried out in special manner. For instance, if  $k_1, \dots, k_{N_z}$  are zero crossing bins, if sign is change is applied to bins  $k = k_1 + 1, \dots, k_2 - 1$

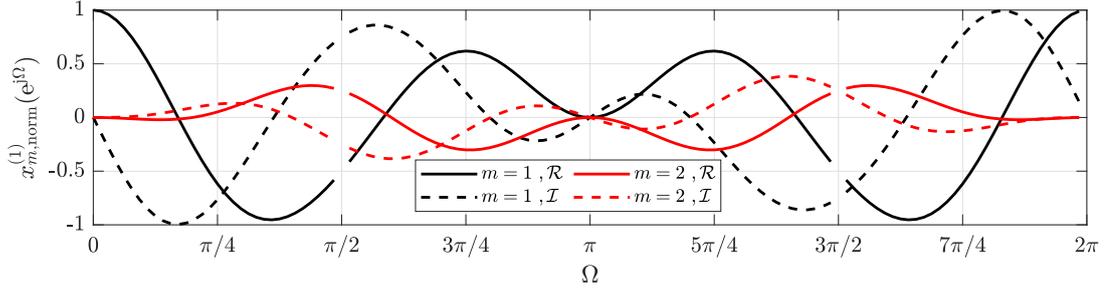


Figure 5.3: Inverted section of the curves, shown in Fig. 5.2, between two zero crossings i.e.  $\Omega = \{\pi/2, 3\pi/2\}$ .

as

$$\mathbf{x}_{\text{norm}}^{(1)}(e^{j\Omega_k}) \leftarrow -\mathbf{x}_{\text{norm}}^{(1)}(e^{j\Omega_k}), \quad k = k_1 + 1, \dots, k_2 - 1,$$

then the next adjacent range of bins i.e.  $k_2 + 1, \dots, k_3 - 1$  will not be applied negative sign. Similarly,  $k_3 + 1, \dots, k_4 - 1$  will require sign change while  $k_4 + 1, \dots, k_5 - 1$  will not implement sign inversion. Similarly, sign change is applied in alternate manner. To figuratively explain this procedure of sign inversion, consider an additional example where  $c_m(z)$  has 6 zero crossings on the unit circle.

**Example 9.** With  $\mathbf{x}^{(1)}(z) = c_m(z)\mathbf{q}_1(z)$ , where  $c_m(e^{j\Omega})$  has zero crossings at  $\Omega = n\pi/4$  where  $n = 1, \dots, 6$ , the components of the normalized vector are illustrated in Fig. 5.4(a), drawn using a DFT size of  $2^{10}$  bins. The zero crossings coincides with bins  $k \in \{129, 257, 385, 513, 641, 769\}$ . Additional phase shift of  $\pi$  is applied to the normalized vector in bins  $\{1, \dots, 128\}$ ,  $\{258, \dots, 384\}$ ,  $\{514, \dots, 640\}$ , and  $\{770, \dots, 2^{10}\}$ , which represents bins before  $\pi/4$  or the first zero crossing, between the 2nd and 3rd, 4th and 5th zero crossings, and after the 6th zero crossing, respectively. With application of  $\pi$  phase shift in alternating manner, the resulting time-domain equivalent of the normalized vector has time-domain aliasing of  $5 \times 10^{-29}$  which estimates the corresponding eigenvalue with accuracy of  $\xi_\lambda = 5.5 \times 10^{-12}$ .  $\triangle$

Contrary to an even number of zeros, an odd number of zeros of  $c_m(e^{j\Omega})$  linked with phase discontinuities will not permit  $\phi_m(e^{j\Omega})$  to be continuous in nature. However, the oversampled version by a factor 2,  $c_m(z^2)$ , will possess an even number of zeros on the

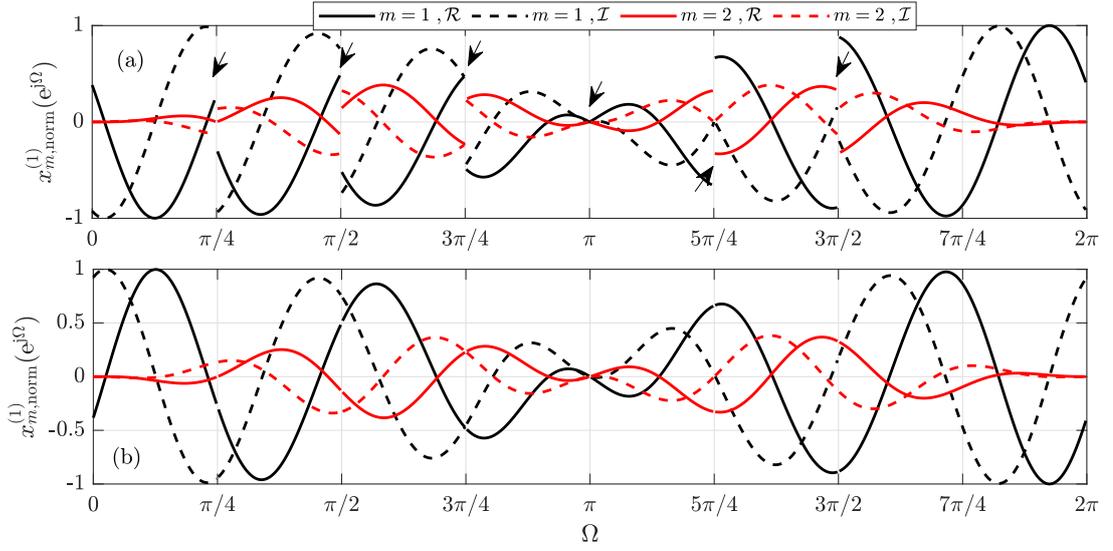


Figure 5.4: Normalized vector components real and imaginary components with even number of zero crossing in  $c_m(e^{j\Omega})$  (a) without out sign change, (b) with sign change applied in alternate manner i.e.  $\Omega \in [\{0, \pi/4\}, \{\pi/2, 3\pi/4\}, \{\pi, 5\pi/4\}, \{3\pi/2, 2\pi\}]$ .

unit circle, making it possible to derive a non discontinuous allpass factor  $\phi_m(e^{j\Omega})$ , as described through Example 8-9. The same procedure of sign inversion is applied, ensuring a smooth evolution of both the real and imaginary parts of all components of the normalized vector. While only the phase of the normalized vector in the DFT domain is affected, it retains the characteristics of an eigenvector within each DFT bin. The distinction lies in its oversampled state. Consequently, once the time-domain sequence is obtained via an IFFT, downsampling in the time-domain is performed to extract the required eigenvector. This assertion is further reinforced through the example described below.

**Example 10.** This example considers the  $\mathbf{x}^{(1)}(z) = (\sum_{n=0}^3 z^{-n})\mathbf{q}_1(z)$  where  $c_m(z) = 1 + z^{-1} + z^{-2} + z^{-3}$  has three zero crossing on unit circle at  $\Omega = \pi/2, \pi, 3\pi/2$ , depicted in the normalized vector illustrated in Fig. 5.5(a) via arrows. Upon applying a sign change is applied to alternate section of the normalized curve similar to as described previously, both the real and imaginary part of all the components of the normalized vector will not be periodic in the DFT domain. The reason is the odd number of zero crossing. To address this, the normalized vector in the DFT domain is oversampled by a

factor of 2 before applying the sign inversion. The oversampled version exhibits 6 zeros, shown in Fig. 5.5(b) after the alternate sign inversion. It can be observed that the real and imaginary components evolve smoothly, except at the zero crossings where these values are undefined. As all zero crossings in the selected example lie at odd numbered DFT bins,  $\hat{\mathbf{x}}_{\text{norm}}^{(1)}[\tau]$  is obtained via an IFFT performed by considering only the even numbered DFT bins. The resulting time-domain equivalent is down sampled by 2 which resulted in time-domain aliasing of  $3.5 \times 10^{-30}$ , providing an accurate estimation of the corresponding eigenvalue through the Rayleigh quotient with  $\xi_\lambda = 10^{-11}$ .  $\triangle$

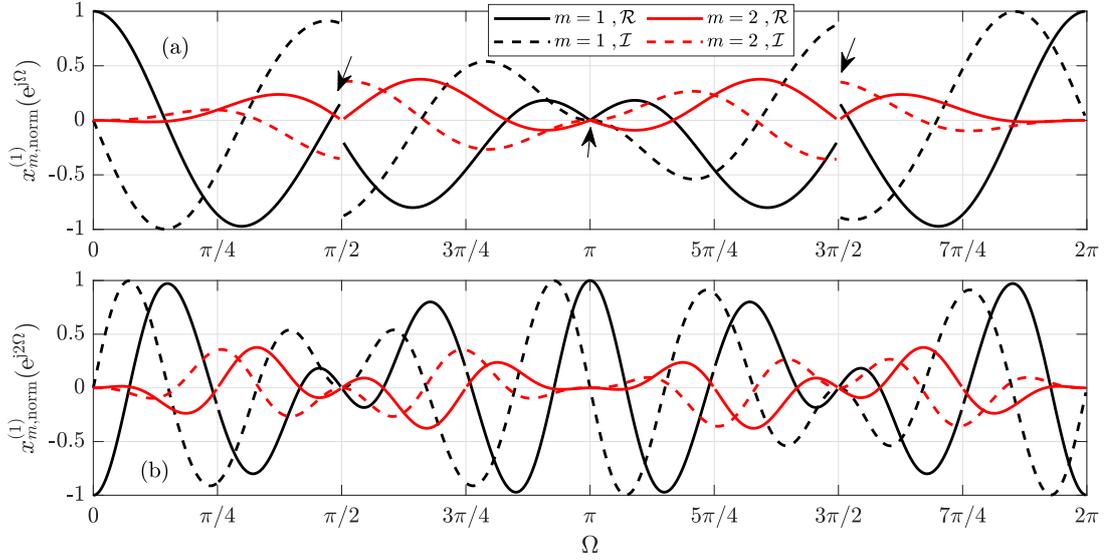


Figure 5.5: Normalized vector in presence of odd number of zero crossings in  $c_m(e^{j\Omega})$ , here showing example of 3 with (a) without oversampling and sign inversion (b) with oversampling by 2 and applying sign inversion to section of curve between alternate zero crossings. (arrows indicate zero crossings associated with phase discontinuities)

### Detecting zeros with phase discontinuities

Since  $c_m(z)$  is unknown, instead we have access to  $\mathbf{x}^{(1)}(z)$ , therefore, zero crossings can only be detected on the unit circle via  $\mathbf{x}^{(1)}(e^{j\Omega})$ . In addition, examining the phases of the normalized vector components,  $\mathbf{x}_{\text{norm}}^{(1)}(z)$ , can help identify whether a particular zero is associated with a phase discontinuity. When the zero crossings of the eigenvector components don't align with those of  $c_m(z)$  on the unit circle, any phase discontinuities in all components of the normalized vector will indicate the locations of zeros in  $c_m(e^{j\Omega})$

linked with phase discontinuities. In the worst-case scenario, only  $M - 1$  components of an eigenvector can share a common zero on the unit circle with  $c_m(z)$  due to the constraint  $\|\mathbf{q}_m(e^{j\Omega})\|_2 = 1$ . Therefore, the zero crossings of  $c_m(e^{j\Omega})$  can be identified by detecting the zeros of  $\|\mathbf{x}^{(1)}(e^{j\Omega})\|_2$ . Meanwhile, its phase discontinuities can be determined by analyzing the phase of the individual components of  $\mathbf{x}_{\text{norm}}^{(1)}(e^{j\Omega})$ , which will be common across all  $M$  components. If a zero crossing does not align with any the DFT bins, it can still be detected by observing phase discontinuities within the normalized vector's components. Any phase discrepancies would be uniform across all  $M$  components at any frequency point, unless a specific component of the eigenvector possesses a zero crossing at that frequency. An example is presented below to elaborate on situations where zero crossings do not align with bins.

**Example 11.** Consider  $\mathbf{x}^{(1)}(z) = (1 + (0.98 - 1.035j)z^{-1} - (0.052 + 0.999j)z^{-2})\mathbf{q}_1(z)$  where  $c_m(e^{j\Omega})$  exhibits two zero crossings at  $\Omega = \pi/2 - \pi/180$  and  $3\pi/2 - \pi/60$ , respectively, with a  $\pi$  phase discontinuity around each. The zero crossings of  $c_m(e^{j\Omega})$  do not coincide with any DFT bins or the zero crossing of any components of  $\mathbf{q}_1(e^{j\Omega})$  if normalization is performed in the DFT bin at a DFT size of 256. However, it can be detected by analysing the phase of the components of  $\mathbf{x}_{\text{norm}}^{(1)}(e^{j\Omega_k})$ ,  $k = 1, \dots, 256$  as illustrated in Fig. 5.6(a). Phase discontinuity of a  $\pi$  radians around  $\Omega = \pi/2$  and  $3\pi/2$  are common in all components of the normalized vector even though none of the bins coincide with zero crossings of  $c_m(e^{j\Omega})$ . However, the phase discontinuity around  $\Omega = \pi$  for  $x_{m,\text{norm}}^{(1)}(e^{j\Omega})$  for  $m = 1, 2$  is not encountered by the third component which negate the possibility of any zero crossings of  $c_m(e^{j\Omega})$  around  $\Omega = \pi$ . Further evidence from the magnitude profile of  $\mathbf{x}^{(1)}(e^{j\Omega})$  i.e.  $\|\mathbf{x}^{(1)}(e^{j\Omega})\|_2$  (depicted in Fig. 5.6(b)) indicates no dip around  $\Omega = \pi$ . Consequently,  $c_m(e^{j\Omega})$  exhibits two zero crossings proximal to  $\Omega = \pi/2$  and  $3\pi/2$  that correspond to phase discontinuity.

Hence, when an even number of zero crossings is observed, oversampling is not conducted. Instead, a sign inversion is performed to the segment of  $\mathbf{x}_{\text{norm}}^{(1)}(e^{j\Omega})$  starting from  $\Omega = 0$  until just before the first detected zero crossing frequency, and from the section after the second zero crossing until  $\Omega = 2\pi$ . In the context of the DFT domain, sign inversion occurs for bins  $k = 1, \dots, 64$  and  $191, \dots, 256$ . The

resulting estimated  $\hat{\mathbf{x}}_{\text{norm}}^{(1)}[\tau] = \hat{\mathbf{q}}_1[\tau]$ , obtained by performing a 256–point IFFT of  $\mathbf{x}_{\text{norm}}^{(1)}(e^{j\Omega_k}), k = 1, \dots, 256$ , exhibits a time-domain aliasing of  $\zeta_q = 1.65 \times 10^{-30}$ . This estimated vector is then utilized to compute the corresponding eigenvalue via the Rayleigh quotient, resulting in a substantially reduced value of  $\xi_\lambda = 1.4 \times 10^{-9}$ . This demonstration emphasizes that the zero crossings need not align with DFT bins to be detected. Furthermore, it indicates that distinguishing zero crossings associated with phase jumps from those without such jumps is relatively straightforward. Additionally, the accuracy provided by this example in normality and eigenvalue estimation errors underscores that detecting the precise position of the zero crossings of  $c_m(e^{j\Omega})$  may not be imperative.

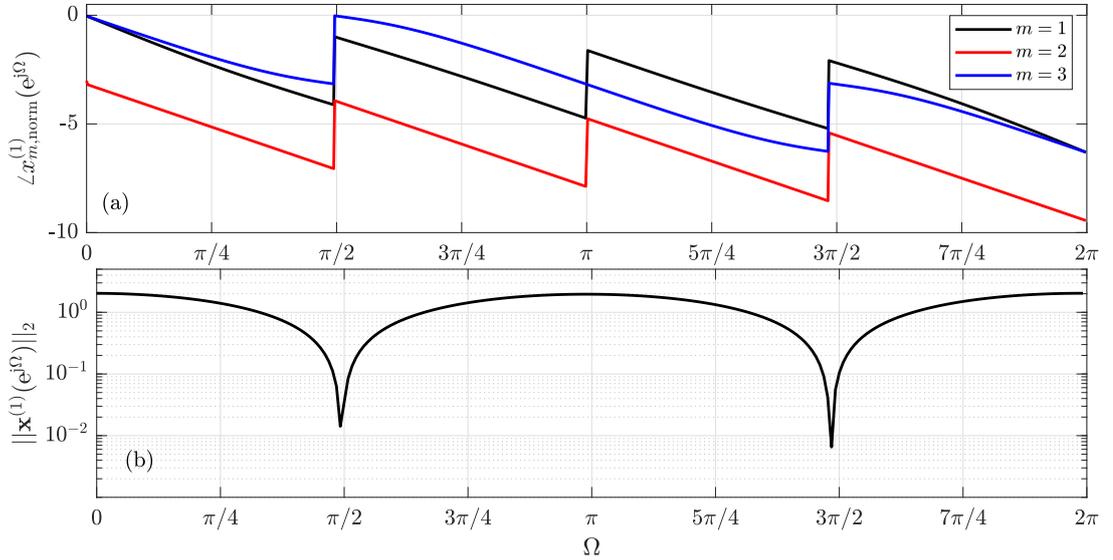


Figure 5.6: Detection of zero crossings of  $c_m(e^{j\Omega})$  associated with phase discontinuities via a combined observation from (a) phase discontinuities observed in the components of  $\mathbf{x}_{\text{norm}}^{(1)}(e^{j\Omega})$ , and (b) magnitude profile  $\|\mathbf{x}^{(1)}(e^{j\Omega})\|_2$  in case some or all zero crossing do not coincide with the DFT bin.

△

### 5.2.7 Truncation

The coefficients of the eigenpair decay at least exponentially, assuming the eigenpair is analytic, which permits it to be truncated via a small threshold [13, 27, 28, 30] to reduce

its support for reducing the implementation cost in signal processing applications. The threshold for truncating the coefficients of the estimated eigenvalues and eigenvectors are denoted as  $\mu_{\text{PH}}$  and  $\mu_{\text{PU}}$ , respectively.

### 5.2.8 Reduced PEVD to Full PEVD

Given that solely  $p$  out of the  $M$  rank one matrices can be effectively estimated using this approach, the remaining  $M - p$  matrices become zero matrices due to the condition  $\lambda_m(z) = 0 \forall z$  for  $m = p + 1, \dots, M$ . To recover the remaining  $M - p$  eigenvectors in a manner that ensures orthogonality among themselves and with the already estimated  $p$  eigenvectors i.e.  $\hat{\mathbf{q}}_m(z)\hat{\mathbf{q}}_j(z) \approx \delta[m - j]$ , thereby enabling the construction of a complete  $M \times M$  dimension paraunitary filter bank, the Gram-Schmidt orthogonalization procedure [1] can be explored.

#### Polynomial Gram-Schmidt Orthogonalization

The eigenvectors corresponding the zero eigenvalues  $\mathbf{q}_m(z), m = p + 1, \dots, M$  can be determined from the already estimated  $p$  eigenvectors  $\hat{\mathbf{q}}_m(z) m = 1, \dots, p$ , by a polynomial vector ortho-normalization on the unit circle i.e. for  $z = e^{j\Omega}$ . This method is the extension of the famous Gram-Schmidt orthogonalization [1] process to the polynomial domain. Thus for randomly initialized  $\mathbf{x}(z)$ , the frequency dependent version of the Gram-Schmidt process for determining  $\mathbf{q}_m(z)$  can be given as

$$\mathbf{w}(e^{j\Omega}) = \mathbf{x}(e^{j\Omega}) - \sum_{n=1}^{m-1} \hat{\mathbf{q}}_n^H(e^{j\Omega})\mathbf{x}(e^{j\Omega})\hat{\mathbf{q}}_n(e^{j\Omega}), \quad (5.13)$$

which, if normalized at sufficient DFT size, results in

$$\mathbf{w}_{\text{norm}}(e^{j\Omega}) = \frac{\mathbf{w}(e^{j\Omega})}{\|\mathbf{w}(e^{j\Omega})\|_2} = g_m(e^{j\Omega})\mathbf{q}_m(e^{j\Omega}) \quad (5.14)$$

with some allpass function  $g_m(e^{j\Omega})$ . The above relation is implemented in the DFT domain such that the DFT size is iteratively increased until the resulting time-domain aliasing, captured by a metric  $\zeta_q$  defined akin to (4.43), satisfies a preset threshold. For measuring the metric, the  $K$ -point IDFT of  $[\mathbf{w}_{\text{norm}}(e^{j\Omega_1}), \dots, \mathbf{w}_{\text{norm}}(e^{j\Omega_K})]$  will

---

**Algorithm 2:** Rank One Decomposition Based PEVD Algorithm

---

**Input:**  $\mathbf{R}(z)$ ,  $\epsilon_{\zeta_q}$ ,  $\epsilon_{\gamma_R}$ ,  $\mu_{\text{PH}}$ ,  $\mu_{\text{PU}}$

**Output:**  $\hat{\mathbf{Q}}(z)$ ,  $\hat{\mathbf{\Lambda}}(z)$

Determine  $\hat{\mathbf{R}}_m(z)$ ,  $m = 1, \dots, p$ ;  $\zeta_q = 1$ ;

**for**  $m = 1 : p$  **do**

    initialize  $\mathbf{x}^{(0)}(z)$ ;  $\mathbf{x}^{(1)}(z) \leftarrow \hat{\mathbf{R}}_m(z)\mathbf{x}^{(0)}(z)$ ;

**while**  $\zeta_q > \epsilon_{\zeta_q}$  **do**

$K \leftarrow 2K$ ;

$\hat{\mathbf{q}}_m(z) \leftarrow \text{Normalization } \mathbf{x}^{(1)}(z)$ ;

**end**

$\hat{\lambda}_m(z) = \hat{\mathbf{q}}_m^{\text{P}}(z)\mathbf{R}_m(z)\hat{\mathbf{q}}_m(z)$

**end**

Estimate  $\hat{\mathbf{q}}_m(z)$ ,  $m = p + 1, \dots, M$  via Gram-Schmidt orthogonalization;

Truncate  $\hat{\mathbf{Q}}(z)$  and  $\hat{\mathbf{\Lambda}}(z)$  with thresholds  $\mu_{\text{PU}}$  and  $\mu_{\text{PH}}$ ;

---

produce the time-domain vector  $\hat{\mathbf{q}}_m[\tau]$ . When the metric falls below a threshold of  $\epsilon_q$ , the value of  $K$  can be considered adequate. The support of the resultant estimated eigenvector or the necessary DFT size relies on the initialization of  $\mathbf{x}(z)$ . Ideally,  $\mathbf{x}(z)$  should be chosen in a manner that transforms the allpass factor  $g_m(z)$  into a straightforward delay, which in reality may not be possible. Likewise, in the event of encountering any spectral zero during the normalization process, the same approach as elucidated above can be readily applied. The entire rank one decomposition based PEVD algorithm is outlined in Algorithm 2.

### 5.3 Normalization Free PEVD Approach

With choices of initialization of  $\mathbf{x}^{(0)}(z)$  restricted to the  $M$  columns of  $\hat{\mathbf{R}}_m(z)$ , the rank one terms estimation and the normalization step in (5.12) may not be necessary in order to compute the PEVD of a para-Hermitian matrix. Therefore, this section delves into an in-depth analysis of how such initialization circumvents the normalization step and provides an overview of the general procedural framework of this method.

### 5.3.1 Analysis for Normalization Free Approach

In order to estimate  $\mathbf{q}_m(z)$ , let us assume  $\mathbf{x}^{(0)}(z)$  is initialized with the  $n$ th column of  $\hat{\mathbf{R}}_m(z)$  i.e.  $\hat{\mathbf{q}}_m(z)\hat{\lambda}_m(z)\hat{q}_{n,m}^P(z)$  where  $\hat{q}_{n,m}(z)$  is the  $n$ th element of  $\hat{\mathbf{q}}_m(z) \in \mathbb{C}^M$ . A single iteration will produce  $\mathbf{x}^{(1)}(z) = \hat{\mathbf{q}}(z)\hat{\lambda}^2(z)\hat{q}_{m,n}^P(z)$  for which the normalized vector, with the normalization step carried out in the DFT-domain with DFT size of  $K$ , will be

$$\mathbf{x}_{\text{norm}}^{(1)}(e^{j\Omega_k}) = \frac{\hat{\mathbf{q}}(e^{j\Omega_k})\hat{\lambda}^2(e^{j\Omega_k})\hat{q}_{n,m}^*(e^{j\Omega_k})}{\sqrt{\mathbf{x}^{(1),H}(e^{j\Omega_k})\mathbf{x}^{(1)}(e^{j\Omega_k})}} = \frac{\hat{\mathbf{q}}(e^{j\Omega_k})\hat{q}_{n,m}^*(e^{j\Omega_k})}{|\hat{q}_{n,m}(e^{j\Omega_k})|}, \quad k = 1, \dots, K. \quad (5.15)$$

Now the aforementioned results can be arrived at directly from the bin-wise EVD results without either performing any rank one decomposition or single iteration. Substituting  $\hat{\mathbf{q}}_m(e^{j\Omega_k}) = e^{j\phi_{m,k}}\mathbf{q}_{m,k}$  and  $\hat{q}_{n,m}(z) = e^{j\phi_{m,k}}q_{n,m,k}$  from (5.4) into (5.15) produces

$$\mathbf{x}_{\text{norm}}^{(1)}(e^{j\Omega_k}) = \frac{\mathbf{q}_{m,k}q_{n,m,k}^*}{|q_{n,m,k}|} = \mathbf{q}_{m,k}e^{j\angle q_{n,m,k}^*}, \quad k = 1, \dots, K. \quad (5.16)$$

Here  $q_{n,m,k}$  refers to the  $n$ th element of the  $m$ th eigenvector at the  $k$ th bin i.e.  $\mathbf{q}_{m,k}$ . It can be seen that for this particular initialization, the end results are such that they are easily obtainable from the bin-wise EVD results. Hence, if the bin-wise eigenvectors are accessible, it becomes possible to directly acquire the single iteration normalized vector, as indicated by (5.16). Consequently, both the estimation of rank one terms and the vector-product multiplication can be circumvented, in addition to the normalization step. In the estimation of the  $m$ th eigenvector, any of the  $M$  components of  $\mathbf{q}_{m,k}$  can be employed in (5.16). Unlike in proper single iteration approach, up-sampling has to be carried out in the DFT domain since the time-domain sequence of the unnormalized version of  $\mathbf{x}_{\text{norm}}^{(1)}(e^{j\Omega_k})$  i.e.  $\mathbf{x}^{(1)}[\tau]$  is not known beforehand. To up-sample by a factor of 2, we concatenate two sets of  $K$ -bins as  $[\mathbf{x}_{\text{norm}}^{(1)}(e^{j\Omega_1}), \dots, \mathbf{x}_{\text{norm}}^{(1)}(e^{j\Omega_K}), \mathbf{x}_{\text{norm}}^{(1)}(e^{j\Omega_1}), \dots, \mathbf{x}_{\text{norm}}^{(1)}(e^{j\Omega_K})]$ .

The single polynomial iteration based approach reduces to the normalization-free approach if  $c_m(z)$  is set equal to  $q_{n,m}^P(z)$ . Therefore, it becomes imperative to identify the zero crossings of  $q_{n,m}^P(z)$  on the unit circle, specifically those that are associated with phase discontinuities or have odd multiplicities. In this regard, similar procedure

to that described in Section 5.2.6 is applicable here. In case of an even number of zeros with odd multiplicities or phase discontinuities, only a sign inversion is applied to the alternate section between zero crossings bins. However, in case of an odd number of zero crossings associated with phase discontinuities, up-sampling is required along with sign inversion. Subsequently, the required down sampling is performed in the time-domain vector obtained via an IFFT of the upsampled DFT domain vector.

### 5.3.2 Eigenvector Support and Initialization Choices

This approach restricts the initialization options for an eigenvector extraction to  $M$ , corresponding to the number of components in each eigenvector. As a result, the  $m$ th eigenvector can be derived via (5.16) for any value of  $n$  within the range  $1, \dots, M$ , albeit with varying support depending on the chosen  $n$ . To achieve the smallest feasible support using the available  $n$  choices, it is possible to compute  $\mathbf{x}_{\text{norm}}^{(1)}(e^{j\Omega_k})$  at different values of  $n$ , commencing with the smallest DFT size, and subsequently selecting the  $n$  that yields the lowest  $\zeta_q$  value.

**Example 12.** Consider the example  $\mathbf{R}(z) \in \mathbb{C}^{3 \times 3}$  from Section 4.3.5 where the first eigenvector is extracted through this normalization-free approach. This example attempts to estimate the dominant eigenvector via (5.16) using various values of  $n = 1, 2, 3$ . For each value of  $n$ , the time-domain sequence of the eigenvector is estimated at different DFT sizes. Subsequently, truncation is applied to the outer lags using a threshold value  $\mu_{\text{PU}} = 10^{-8}$ . The resulting time-domain support for  $n = 1, 2$ , and  $3$  is found to be 28, 5, and 54, respectively. Moreover, the time-domain aliasing metric  $\zeta_q$  for all these truncated eigenvector estimates remains below  $10^{-18}$ , indicating a high level of accuracy in the estimates. This example effectively demonstrates that different values of  $n$  result in different support sizes for the estimated eigenvector.  $\triangle$

## 5.4 PSVD by Rank One Decomposition

Like for the PEVD of a para-Hermitian  $\mathbf{R}(z)$ , achieving the PSVD of an analytic and non-multiplexed general matrix  $\mathbf{A}(z) \in \mathbb{C}^{M \times L}$  necessitates a rank one decomposition

along with the execution of a single iteration. To obtain the left- and right-singular vectors, a single polynomial power iteration has to be applied to every rank one component of the para-Hermitian matrix  $\mathbf{A}(z)\mathbf{A}^P(z)$  or  $\mathbf{A}^P(z)\mathbf{A}(z)$  similar to the generalized polynomial power method discussed in Section 4.6. To circumvent the necessity of a rank one decomposition for both para-Hermitian matrices, an alternative approach is adopted where the rank one decomposition of  $\mathbf{A}(z)$  is directly ascertained. Due to the analytic PSVD existence for an analytic, non-multiplexed  $\mathbf{A}(z)$  as detailed in Section 2.3, the rank one decomposition is similarly possible to a para-Hermitian matrix. Therefore, for  $\mathbf{A}(z)$  with  $p \leq L$  non-zero singular values, the following holds true

$$\mathbf{A}(z) = \sum_{n=1}^p \mathbf{A}_n(z) \text{ with } \mathbf{A}_n(z) = \mathbf{u}_n(z)\sigma_n(z)\mathbf{v}_n^P(z), \quad (5.17)$$

where  $\mathbf{A}_n(z)$  is a rank one matrix.

#### 5.4.1 Rank One Decomposition of $\mathbf{A}(z)$

The sample points of the rank one  $\mathbf{A}_n(z)$  can be obtained via the bin-wise SVD as done in (5.1). With the assumption of spectral majorisation, the bin-wise SVD of the sample points of  $\mathbf{A}(z)$  on the unit-circle  $\mathbf{A}(e^{j\Omega_k}) = \mathbf{A}_k$ ,

$$\mathbf{A}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^H, \quad (5.18)$$

can be related to the sample points of analytic functions in (2.6)

$$\mathbf{\Sigma}(e^{j\Omega_k}) = \mathbf{\Sigma}_k, \quad (5.19)$$

$$\mathbf{U}(e^{j\Omega_k}) = \mathbf{U}_k \mathbf{\Phi}_k, \quad \mathbf{V}(e^{j\Omega_k}) = \mathbf{V}_k \mathbf{\Phi}_k. \quad (5.20)$$

Here it can be seen that left- and right singular vectors are phase coupled to common arbitrary allpass functions.

Following a similar methodology as described in Section 5.2.2, the sample points of

$\mathbf{A}_n(z)$  can be obtained from the bin-wise SVD i.e. from (5.18) to (5.20) as

$$\mathbf{A}_n(e^{j\Omega_k}) = \mathbf{u}_n(e^{j\Omega_k})\mathbf{v}_n^H(e^{j\Omega_k})\sigma_n(e^{j\Omega_k}) = e^{j\phi_{n,k}}\mathbf{u}_{n,k}e^{-j\phi_{1,k}}\mathbf{v}_{n,k}^H\sigma_{n,k} \quad (5.21)$$

$$= \mathbf{u}_{n,k}\mathbf{v}_{n,k}^H\sigma_{n,k} = \mathbf{A}_{n,k}, \quad n = 1, \dots, p. \quad (5.22)$$

Therefore, the sample points of each ground-truth rank one matrix on the unit circle can actually be obtained from the bin-wise SVD. Hence, the rank one estimates can be obtained from these samples via a  $K$ -point IDFT with  $K$  large enough so that the time-aliasing metric  $\gamma_A$ , defined akin to (5.8), falls below a small threshold. The choice of the windowing function  $p_K[\tau]$  depends on number of positive and negative lags in  $\mathbf{A}[\tau]$ . In general, channel matrix does not exhibit negative lags, so  $p_K[\tau]$  will be a positive time domain window. The resulting rank one decomposition is

$$\mathbf{A}(z) \approx \sum_{n=1}^p \hat{\mathbf{A}}_n(z) \quad \text{with} \quad \hat{\mathbf{A}}_n(z) = \hat{\mathbf{u}}_n(z)\tilde{\sigma}_n(z)\hat{\mathbf{v}}_n^P(z). \quad (5.23)$$

#### 5.4.2 Single Polynomial Power Iteration Application

With rank one estimates, a single iteration of the polynomial power method can be applied separately to  $\hat{\mathbf{A}}_n(z)\hat{\mathbf{A}}_n^P(z)$  and  $\hat{\mathbf{A}}_n^P(z)\hat{\mathbf{A}}_n(z)$ , which are para-Hermitian matrices, to obtain the left- and right-singular vectors, respectively. These independently extracted left- and right singular vectors will not be coupled to a common allpass function. Consequently, the singular value extracted via these uncoupled singular vectors will be complex valued on the unit circle.

An alternative approach for extracting singular vectors with a shared allpass factor involves obtaining the right singular vector via a single iteration applied to  $\hat{\mathbf{A}}_n^P(z)\hat{\mathbf{A}}_n(z)$ . Upon obtaining the right-singular vector, the singular value along with its corresponding left-singular vector can be extracted as outlined in Section 4.6, or more specifically, by utilizing equations (4.40) and (4.41), respectively. Notably, the determination of a suitable DFT size follows a similar methodology as proposed in Section 4.6.

### 5.4.3 Reduced to Full PSVD

The concept of extending the reduced PSVD to a full PSVD, wherein  $\mathbf{U}(z)$  and  $\mathbf{V}(z)$  possess dimensions of  $M \times M$  and  $L \times L$  respectively, rather than  $M \times p$  and  $p \times L$ , can be achieved by utilizing the Gram-Schmidt orthogonalization technique described in Section 5.2.8.

## 5.5 Normalization Free PSVD Approach

To derive an equivalent normalization variant of the aforementioned PSVD algorithm, one can apply the normalization-free PEVD to the matrix  $\mathbf{A}^P(z)\mathbf{A}(z)$ , which, by definition, is a para-Hermitian matrix. This approach yields its eigenvectors, which also represent the right-singular vectors. Subsequently, the remaining components, specifically the singular values and the right singular vector, can be determined following the procedures outlined in Chapter 4 within the section dedicated to the generalized polynomial power method.

However, it is important to note that there is no actual requirement to formulate  $\mathbf{A}^P(z)\mathbf{A}(z)$  explicitly to compute the PSVD. Instead, same approach as that used for PEVD can be applied directly to  $\mathbf{A}(z)$  instead of  $\mathbf{A}^P(z)\mathbf{A}(z)$ , with the only modification being the substitution of bin-wise EVD with SVD. Following this,  $\mathbf{x}_{\text{norm}}^{(1)}(e^{j\Omega_k})$ , particularly in the case of estimating the right singular vector, can be obtained as

$$\mathbf{x}_{\text{norm}}^{(1)}(e^{j\Omega_k}) = \mathbf{v}_{m,k} e^{j\angle v_{n,m,k}^*} \text{ for } k = 1, \dots, K. \quad (5.24)$$

The subsequent steps align with those detailed for the normalization-free PEVD approach in Section 5.3. Upon acquiring the right-singular vector, one can retrieve the associated singular value and left singular vector using the methodology established in the generalized polynomial power method expounded in Section 4.6. Specifically, this can be achieved by employing equations (4.40) and (4.41), respectively.

## 5.6 PQRD by Rank One Decomposition

Although the existence of an analytic polynomial QR decomposition (PQRD) has not been formally proven, this thesis operates under the assumption that the decomposition, as given in (2.7), exists, thereby enabling the decomposition process as follows:

$$\mathbf{A}(z) = \sum_{n=0}^p \underline{\mathbf{A}}_n(z) \quad \text{where } \underline{\mathbf{A}}_n(z) = \underline{\mathbf{q}}_n(z)\underline{\mathbf{r}}_n(z). \quad (5.25)$$

Here,  $\underline{\mathbf{q}}_n(z)$  represents the  $n$ th column of the paraunitary matrix  $\underline{\mathbf{Q}}(z)$ , and  $\underline{\mathbf{r}}_n(z)$  denotes the  $n$ th row of the upper-right triangular matrix  $\underline{\mathbf{R}}(z)$ . For subsequent sections, it is assumed that  $\underline{\mathbf{q}}_n(z)$  and  $\underline{\mathbf{r}}_n(z)$  has the shortest time-domain support. This means that  $\underline{\mathbf{q}}'_n(z) = \phi_n(z)\underline{\mathbf{q}}_n(z)$  and  $\underline{\mathbf{q}}'_n(z) = \phi_n^P(z)\underline{\mathbf{r}}_n(z)$ , which are also valid factors in the context of analytic QR decomposition, will have compact order if  $\phi_n(z)$  takes on a simple delay. Similar to previous rank one decomposition,  $\underline{\mathbf{A}}_n(z), n = 1, \dots, p$  denotes rank one sum of  $\mathbf{A}(z)$ . The number of non-zero rows in  $\underline{\mathbf{R}}(z)$  is equal to the number of non-zero singular values i.e.  $p$ . It is important to emphasize that the rank one SVD component  $\mathbf{A}_n(z)$  is distinct from  $\underline{\mathbf{A}}_n(z)$  due to potential variations in their non-zero singular values.

### 5.6.1 QR-based Rank One Decomposition

To estimate rank one terms in (5.25), first the bin-wise QRD is performed for bin-wise rank one decompositions as

$$\mathbf{A}_k = \underline{\mathbf{Q}}_k \underline{\mathbf{R}}_k; \quad \mathbf{A}(e^{j\Omega_k}) = \mathbf{A}_k = \sum_{n=1}^p \underline{\mathbf{A}}_{n,k} \quad \text{where } \underline{\mathbf{A}}_{n,k} = \underline{\mathbf{q}}_{n,k} \underline{\mathbf{r}}_{n,k}, n = 1, \dots, p, \quad (5.26)$$

where  $\underline{\mathbf{r}}_{n,k}$  is the  $n$ th row of  $\underline{\mathbf{R}}_k$  and  $\underline{\mathbf{q}}_{n,k}$  is the  $n$ th column of  $\underline{\mathbf{Q}}_k$ . The above bin-wise QRD can be related to the sample points of analytic functions in (2.7) as

$$\underline{\mathbf{Q}}(e^{j\Omega_k}) = \underline{\mathbf{Q}}_k \underline{\Phi}_k; \quad \underline{\mathbf{R}}(e^{j\Omega_k}) = \underline{\Phi}_k^H \underline{\mathbf{R}}_k \quad (5.27)$$

where the diagonal phase matrix actually manifests the phase ambiguity of QR decomposition similar to EVD and SVD. Via (5.26), the sample points of  $\underline{\mathbf{A}}_n(z)$  on the unit-circle can be obtained via  $\underline{\mathbf{A}}_{n,k}$  because

$$\underline{\mathbf{A}}_n(e^{j\Omega_k}) = \underline{\mathbf{q}}_n(e^{j\Omega_k})\underline{\mathbf{r}}_n(e^{j\Omega_k}) = e^{j\phi_{n,k}}\underline{\mathbf{q}}_{n,k}e^{-j\phi_{n,k}}\underline{\mathbf{r}}_{n,k} = \underline{\mathbf{q}}_{n,k}\underline{\mathbf{r}}_{n,k} = \underline{\mathbf{A}}_{n,k} \quad (5.28)$$

It can be seen that the sample points of the ground truth rank one terms  $\underline{\mathbf{A}}_n(z)$  on the unit circle are same as the terms  $\underline{\mathbf{A}}_{n,k}$  which are available via bin-wise QRDs. Thus, through a  $K$ -point IDFT of  $\underline{\mathbf{A}}_{n,k}$ , QR-based rank one terms can be estimated. The sufficient DFT size can be determined by assessing the time-domain aliasing metric  $\gamma_{\underline{\mathbf{A}}}$ , defined akin to (5.8), ensuring its value is acceptably low.

### 5.6.2 Normalizing Columns of $\hat{\underline{\mathbf{A}}}_n(z)$ for PQRD

The QR-based rank one terms estimate, denoted by  $\hat{\underline{\mathbf{A}}}_n(z)$ , has columns as

$$\hat{\underline{\mathbf{A}}}_n(z) = [\hat{\underline{\mathbf{q}}}_n(z)\hat{r}_{n,1}(z), \dots, \hat{\underline{\mathbf{q}}}_n(z)\hat{r}_{n,N}(z)],$$

where  $\hat{r}_{n,i}$  is the  $i$ th element of the  $n$ th row of  $\hat{\underline{\mathbf{R}}}(z)$ , which shows that each column of rank one estimate is a scaled version of  $\underline{\mathbf{q}}_n(z)$ , the  $n$ th column of  $\underline{\mathbf{Q}}(z)$ . Therefore, any of the  $\{n, n+1, \dots, L\}$  columns of  $\hat{\underline{\mathbf{A}}}_n(z)$  can be normalized to obtain the estimate of the unit norm vector  $\underline{\mathbf{q}}_n(z)$ . For instance, if an  $i$ th column, denoted with  $\underline{\mathbf{a}}(z) = \hat{\underline{\mathbf{q}}}_n(z)\hat{r}_{n,i}(z)$ , is normalized, where  $i < L$ , the resulting normalization on the unit-circle yields

$$\underline{\mathbf{a}}_{\text{norm}}(e^{j\Omega}) = \frac{\hat{\underline{\mathbf{q}}}_n(e^{j\Omega})\hat{r}_{n,i}(e^{j\Omega})}{\|\hat{\underline{\mathbf{q}}}_n(e^{j\Omega})\hat{r}_{n,i}(e^{j\Omega})\|_2} = \frac{\hat{\underline{\mathbf{q}}}_n(e^{j\Omega})\hat{r}_{n,i}(e^{j\Omega})}{|\hat{r}_{n,i}(e^{j\Omega})|} = \hat{\underline{\mathbf{q}}}_n(e^{j\Omega})\varphi(e^{j\Omega}), \quad (5.29)$$

where  $\varphi(e^{j\Omega}) = \hat{r}_{n,i}(e^{j\Omega})/|\hat{r}_{n,i}(e^{j\Omega})|$ , which is the desired result. In case, the allpass factor  $\varphi(z)$  is a simple delay,  $\hat{\underline{\mathbf{q}}}_n(z)$  would have lowest order. Following the procedure discussed in the previous section, the normalization is performed in the DFT domain where the DFT size is iteratively increased until the time-domain aliasing in normalized  $\underline{\mathbf{a}}_{\text{norm}}[\tau]$ , obtained via an IFFT, diminishes to a value below a designated threshold.

The time-domain aliasing metric is defined akin to (4.43), Due to the assumption that  $M \geq L$ , rank  $p$  of  $\mathbf{A}(z)$  can be  $p \leq L$ . Therefore, the proposed approach can only extract  $p$  or  $L$  of the  $M$  columns of the paraunitary  $\underline{\mathbf{Q}}(z)$ , depending upon the rank of  $\mathbf{A}(z)$ . The remaining  $M - p$  or  $M - L$  normal vectors can be determined through the Gram-Schmidt orthogonalization as detailed in Section 5.2.8. Once all  $M$  columns of  $\hat{\underline{\mathbf{Q}}}(z)$  have been obtained,  $\hat{\underline{\mathbf{R}}}(z)$  can be easily obtained from  $\hat{\underline{\mathbf{Q}}}^{\text{P}}(z)\mathbf{A}(z)$ .

Furthermore, it is worth noting that there exists no assurance that  $\underline{r}_{n,i}(z)$  or  $\hat{\underline{r}}_{n,i}(z)$  would be devoid of zeros on the unit circle, potentially rendering normalization unattainable. Therefore, in such cases, the approach described in Section 5.2.6 can be applied. A noteworthy consideration is that the support of the extracted column from  $\underline{\mathbf{Q}}(z)$  may vary contingent on the specific column of  $\hat{\underline{\mathbf{A}}}_n(z)$  being utilized. Given the unpredictability of which column will yield the lowest order post-normalization, one could experiment with multiple columns and select the one with the smallest support. This strategy accommodates the uncertainty surrounding the optimal choice of column for achieving the lowest order after normalization.

## 5.7 Normalization Free PQRD Approach

Upon a comprehensive examination of the normalization procedure outlined in (5.29), it becomes apparent that resorting to QR-based rank one decomposition may not be obligatory. Instead, a bin-wise QR decomposition might suffice to attain the result presented in (5.29).

Leveraging the relationship expressed in (5.27), the normalized vector at  $e^{j\Omega_k}$ , as defined in (5.29), can be obtained through bin-wise QR decomposition. This involves the respective components  $\underline{\mathbf{q}}_{n,k}$  and  $r_{n,i,k}$ , leading to the following expression:

$$\begin{aligned} \underline{\mathbf{a}}_{\text{norm}}(e^{j\Omega_k}) &= \frac{\hat{\underline{\mathbf{q}}}_n(e^{j\Omega_k})\hat{\underline{r}}_{n,i}(e^{j\Omega_k})}{|\hat{\underline{r}}_{n,i}(e^{j\Omega_k})|} = \frac{e^{j\phi_{n,k}}\underline{\mathbf{q}}_{n,k}e^{-j\phi_{n,k}}r_{n,i,k}}{|r_{n,i,k}|} \\ &= \frac{\underline{\mathbf{q}}_{n,k}r_{n,i,k}}{|r_{n,i,k}|} = \underline{\mathbf{q}}_{n,k}e^{j\angle r_{n,i,k}}, \end{aligned} \quad (5.30)$$

This observation highlights that even without performing a rank one decomposition,

the normalized vector  $\underline{\mathbf{a}}_{\text{norm}}(e^{j\Omega_k})$  can be directly obtained from a bin-wise QR decomposition. In this context, the phase of  $r_{n,i,k}$  is a crucial element, which can be readily accessed via a bin-wise QR decomposition. Then, following the standard practice, the time-domain sequence can be obtained by applying the IDFT to the normalized vector using a DFT size that is sufficiently large to minimize time-domain aliasing. The components of  $\underline{\mathbf{a}}_{\text{norm}}(e^{j\Omega_k})$  may have discontinuities between the adjacent bins due to phase jumps encountered due to zero crossings in  $r_{n,i}(e^{j\Omega})$  for which a similar procedure, described in Section 5.2.6, can be adopted. Likewise PEVD and PSVD, after extracting a set of  $p \leq L$  columns, the remaining  $M - p$  columns of  $\underline{\mathbf{Q}}(z)$  can be obtained via the Gram-Schmidt procedure. This resultant set of all  $M$  vectors i.e.  $\hat{\underline{\mathbf{q}}}_m(z)$ ,  $m = 1, \dots, M$  can then be employed for estimating the matrix  $\underline{\mathbf{R}}(z)$ .

## 5.8 Simulation and Results

This thesis only evaluates the normalization-free variant for the PEVD, PSVD and PQRD against the state-of-the-art algorithms. The underlying reason for not evaluating the rank one decomposition approach for the mentioned decompositions is that it is computationally more intensive than the normalization-free variant. This computational overhead arises from the extensive number of FFT and IFFTs:  $M^2$  in case of PEVD and  $LM$  times in case of PSVD and PQRD. These FFTs/IFFTs are repeated at different DFT sizes in the estimation of each rank one term. Additionally, after estimating the rank one terms, obtaining eigenvectors or singular vectors involves polynomial matrix vector multiplication. Subsequent normalization adds an extra pair of FFT/IFFT operations. In contrast, the normalization-free approach directly estimates the necessary components from the bin-wise EVDs, SVDs, and QRDs, eliminating the need for additional FFTs and IFFTs.

### 5.8.1 Performance Metrics

To evaluate and compare the performance of the proposed method against state-of-the-art algorithms, this thesis employs two reconstruction error metrics:

- $\xi_R$ , defined in (4.35), for the PEVD;
- $\xi_A$ , defined akin to (4.35) however in terms of  $\mathbf{A}(z)$ , for PSVD where the reconstructed polynomial matrix is defined as  $\hat{\mathbf{A}}(z) = \hat{\mathbf{U}}(z)\hat{\mathbf{\Sigma}}(z)\hat{\mathbf{V}}(z)$ ;
- triangularization ratio  $\eta$  for comparing PQRD algorithms is defined as

$$\eta = 1 - \frac{\sum_{\tau} \|\tilde{\mathbf{R}}[\tau]\|_{\mathbb{F}}^2}{\sum_{\tau} \|\mathbf{A}[\tau]\|_{\mathbb{F}}^2}, \quad (5.31)$$

where  $\tilde{\mathbf{R}}[\tau]$  is same as the estimated upper triangular matrix  $\hat{\mathbf{R}}[\tau]$  but with the sub-matrix below the main diagonal forced set to zero.

To gauge the computational complexity of the proposed algorithms, this thesis considers execution time as a suitable metric, measured using the "tic-toc" function in the MATLAB environment. The implementation cost of any algorithm is reflected through the order or support of the resulting decomposition matrices, such as eigenvectors in the case of the PEVD, left- and right singular vectors in the case of the PSVD, and the paraunitary  $\hat{\mathbf{Q}}(z)$  matrix for the PQRD.

### 5.8.2 PEVD

For the tests, an ensemble of  $10^3$  spectrally majorised randomized para-Hermitian matrices  $\mathbf{R}(z) \in \mathbb{C}^{3 \times 3}$  is constructed using the source model from [22] with  $\mathcal{O}\{\mathbf{Q}(z)\} \in \{20, 40, \dots, 100\}$  and  $\mathcal{O}\{\mathbf{A}(z)\} \in \{40, 80, \dots, 200\}$ . Both the SBR2 and SMD algorithms are allowed a maximum of 500 iterations with  $\mu_{\text{PH}} = \mu_{\text{PU}} = 10^{-6}$  and the maximum off-diagonal energy threshold set to  $10^{-6}$ .

The normalization-free approach is executed over the entire ensemble with  $\epsilon_{\zeta_q} = 10^{-5}$ ,  $\mu_{\text{PH}} = \mu_{\text{PU}} = 10^{-4}$  and  $K_{\text{max}} = 2^{\lceil \log_2[\mathcal{O}(\mathbf{R}(z))] \rceil + 3}$ . Fig. 5.7 illustrates the ensemble results, showing that the proposed normalization-free approach achieves an order of magnitude lower reconstruction error compared to SBR2 and SMD. It is also worth noting that the resulting order of  $\hat{\mathbf{Q}}(z)$  for the proposed method is on average slightly lower than that of SBR2 and SMD's output. Moreover, the proposed method is considerably faster than SBR2 and SMD as evident from the execution time curve in Fig. 5.7(c).

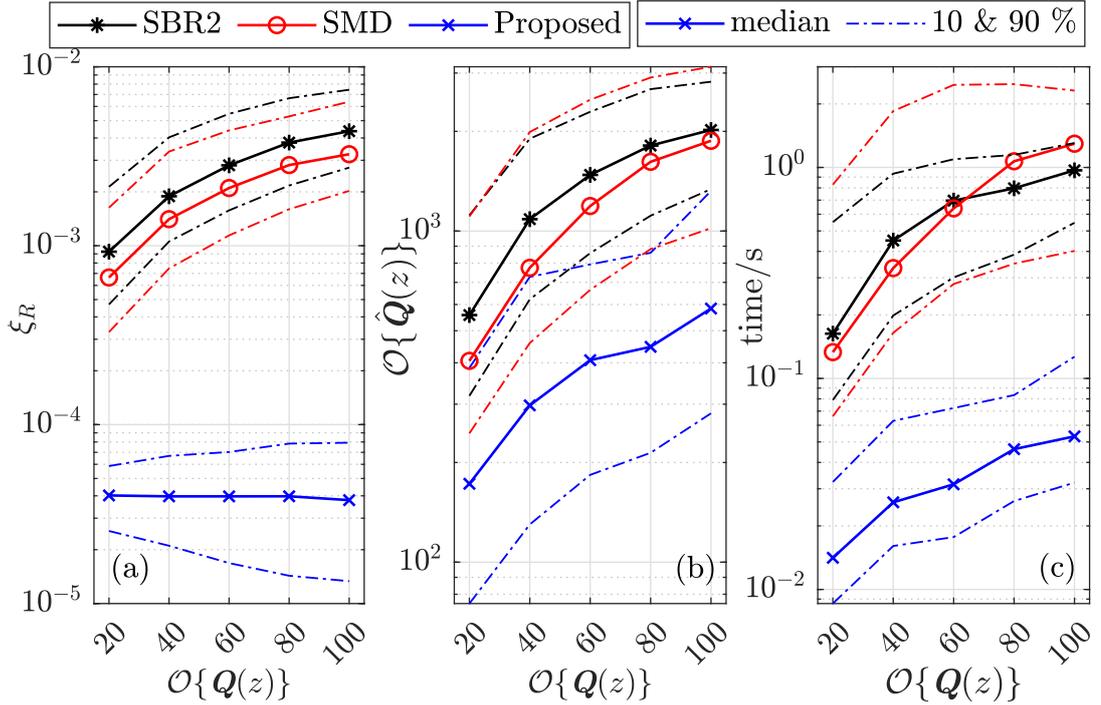


Figure 5.7: PEVD algorithm comparison on an ensemble showing (a)  $\xi_{\hat{R}}$ , (b)  $\mathcal{O}\{\hat{Q}(z)\}$ , and (c) execution time.

This suggests that with the proposed method can perform a quick and accurate decomposition of a para-Hermitian matrix with comparable or lower order approximation for  $\hat{Q}(z)$  compared to the SBR2 and SMD algorithms.

### 5.8.3 PSVD

Although the PSVD can be accomplished through two PEVDs as outlined in [26] or via a PQRD [12], in this comparison, this thesis evaluates the normalization-free variant of the proposed method against the dedicated PSVD algorithm GSR2 [11]. The ensemble employed for the comparison purpose consists of 500 instantiations of  $\mathbf{A}(z) \in \mathbb{C}^{4 \times 3}$  where the orders of  $\mathbf{U}(z)$  and  $\mathbf{V}(z)$  are jointly varied from 10 to 50 in steps of 10, and the order of  $\mathbf{\Sigma}(z)$  from 20 to 100 in steps of 20.

The GSR2 algorithm is executed with the following parameters:  $\mu_{\text{PU}} = 10^{-4}$ ,  $\mu_{\text{PH}} = 10^{-6}$ , utilizing the truncation method described in references [13] and [22], with a maximum off-diagonal threshold  $\epsilon$  set to  $10^{-5}$ . This algorithm is constrained

to execute a maximum of 100 iterations if the maximum off-diagonal threshold is not met. The normalization-free variant is executed with  $\mu_{\text{PU}} = \mu_{\text{PH}} = 10^{-6}$ ,  $\epsilon_{\zeta_u} = 10^{-6}$  and  $K_{\text{max}} = 2^{\lceil \log_2[\mathcal{O}(\mathbf{R}(z))] \rceil + 5}$ . The ensemble results are presented in Fig. 5.8, with

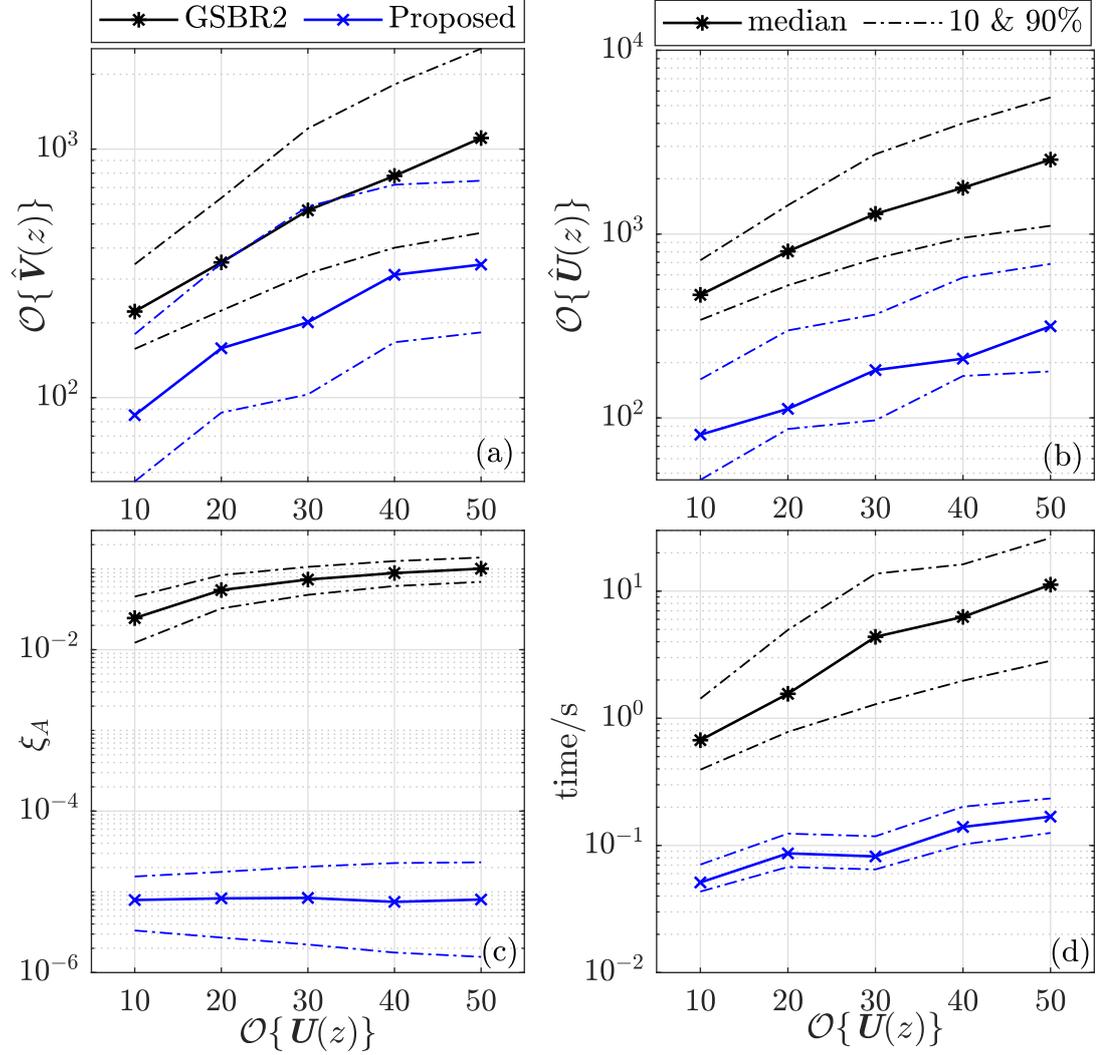


Figure 5.8: Comparison of the normalization-free PSVD algorithm against the GSR2 [11] through: (a)  $\mathcal{O}\{\hat{\mathbf{V}}(z)\}$ , (b)  $\mathcal{O}\{\hat{\mathbf{U}}(z)\}$ , (c)  $\xi_A$ , and (d) time (s) versus order of ground-truth  $\mathbf{U}(z)$ .

the first two figures, (a) and (b), showing the resulting order of both paraunitary matrices compared to the order of the ground-truth  $\mathbf{U}(z)$ . Based on the provided results, it is evident that the proposed method outperforms GSR2 slightly in terms of overall performance. However, when it comes to the accuracy of the decomposition,

the proposed method demonstrates an order of magnitude improvement over GSB2. This superiority is also reflected in the execution time, where the normalization-free approach delivers a lower-order and highly accurate decomposition in significantly less time.

#### 5.8.4 PQRD

In this experiment, both the PQRD-BC and SM-PQRD approaches are compared against the normalization-free PQRD approach outlined in this chapter. The comparison is carried out over an ensemble of  $10^3$  instantiations of  $\mathbf{A}(z) \in \mathbb{C}^{4 \times 4}$  constructed from a known ground-truth paraunitary  $\underline{\mathbf{Q}}(z)$  and upper triangular  $\underline{\mathbf{R}}(z)$ , whose coefficients are drawn from a normal distribution. Ensemble experiments are carried out at various orders of  $\underline{\mathbf{Q}}(z)$  and  $\underline{\mathbf{R}}(z)$ .

Both iterative algorithms are permitted 200 iterations unless the maximum element below the main diagonal falls below  $10^{-8}$ . The intermediate paraunitary and triangular matrix are truncated via  $\mu_{\text{PU}} = \mu_{\text{R}} = 10^{-6}$ . The proposed normalization-free approach is simulated with parameters  $K_{\text{max}} = 2^{\lceil \log_2[\mathcal{O}(\mathbf{A}(z))] \rceil + 3}$  and  $\epsilon_{\zeta_q} = 10^{-6}$ . At each DFT size,  $n$  is varied from  $\{1, \dots, M\}$  and the best  $n$  is decided which satisfies  $\epsilon_{\zeta_q}$ . The proposed algorithm initially estimates  $\underline{\mathbf{Q}}(z)$ , and then use it to estimate  $\underline{\mathbf{R}}(z)$ .

Ensemble results are depicted in Fig. 5.9. A smaller value of parameter  $\eta$  corresponds to a more effective triangularization. The proposed method achieves significantly improved triangularization accuracy compared to the time-domain iterative algorithms as visible in Fig. 5.9 (a). The resulting order of the paraunitary matrix  $\underline{\mathbf{Q}}(z)$  is notably lower for the proposed method in comparison to PQRD-BC and SM-PQRD. The complexity of the proposed algorithm, as reflected in its execution time, is orders of magnitude lower than that of PQRD-BC and SM-PQRD. These findings demonstrate that the proposed method outperforms previously suggested algorithms across all metrics, particularly in terms of accuracy and execution time.

This demonstrates that the proposed method allows for accurate frequency-selective channel equalization, resulting in a lower bit-error-rate (BER) compared to both SM-PQRD [19] and PQRD-BC methods [12, 43]. This conclusion is contingent upon achiev-

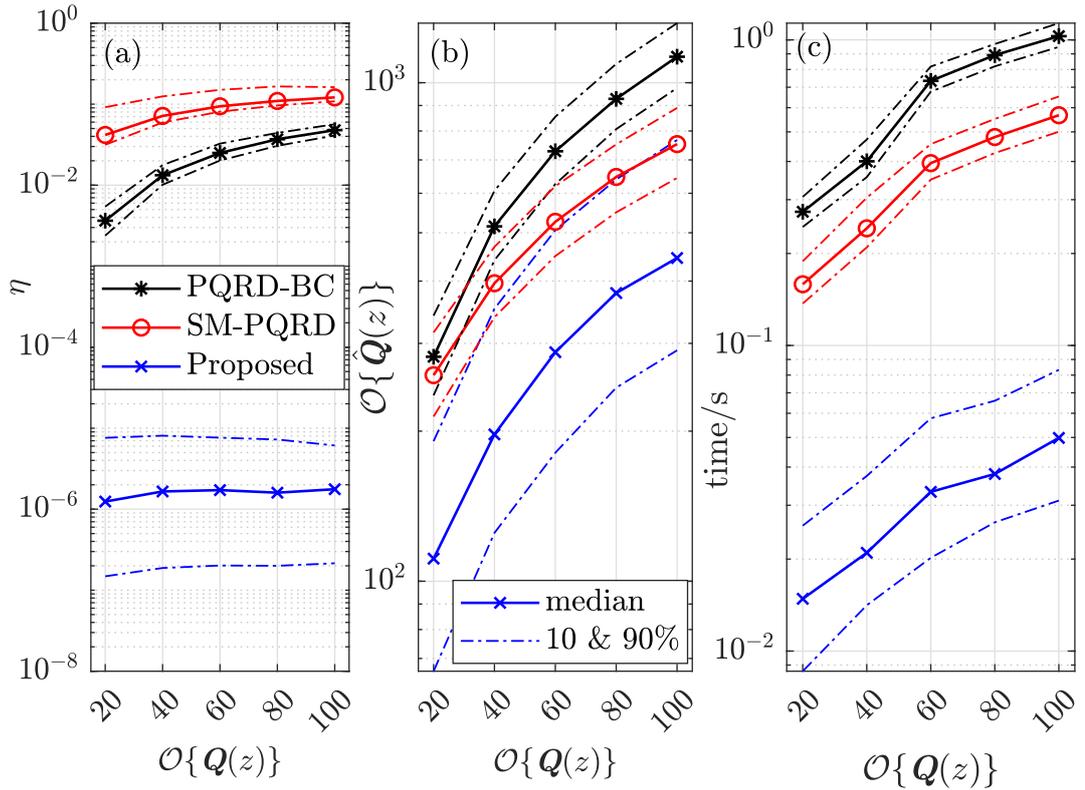


Figure 5.9: Normalization-free PQRD algorithm comparison with SM-PQRD [19] and PQRD-BC [12] through (a)  $\eta$ , (b)  $\mathcal{O}\{\hat{Q}(z)\}$ , and (c) time (s).

ing an accurate PQRD decomposition with the proposed method, as it directly influences the equalization calculations. However, it should be noted that if the channel estimate is inaccurate, the impact of an accurate PQRD decomposition may vary, and definitive conclusions cannot be drawn.

## 5.9 Summary

This chapter has introduced a unified algorithm for computing the PEVD of a para-Hermitian matrix, as well as the PSVD and PQRD of a general polynomial matrix. The approach involves decomposing these matrices into a sum of rank one polynomial matrices. This idea is an extension of the polynomial power method presented in Chapter 4, where a polynomial matrix is decomposed into rank-one matrices, and a single iteration of the polynomial power method is applied for estimation.

The rank-based decomposition PEVD algorithm first splits the given para-Hermitian matrix into rank-one para-Hermitian terms and then applies the polynomial power method to determine the respective eigenpair. However, normalizing the product vector after a single iteration can encounter singularities. This issue is addressed through modulation and sign changes. Furthermore, this method can be computationally improved by restricting the initialization choices of the initial vector  $\mathbf{x}^{(0)}(z)$  to the  $M$  columns of the rank-one terms. This improvement arises from the fact that these initialization choices lead to the same conclusions as those obtained directly from a bin-wise EVD without estimating rank-one matrices or applying a single iteration. Although no normalization is required in this restricted initialization approach, which we refer to in this thesis as normalization-free PEVD, the modulation and sign change trick is still necessary in the case of singularities. Simulation experiments demonstrate a significant improvement in both decomposition accuracy and execution time.

A similar approach is presented for the PSVD and the PQRD of a polynomial matrix, with slightly different rank-one terms for PSVD and PQRD. In the case of the PSVD, rank-one samples on the unit circle are obtained via a bin-wise SVD, whereas for the PQRD, they are obtained through a bin-wise QR decomposition. The PSVD method similarly applies a single iteration to a rank-one para-Hermitian matrix obtained from the rank-one term of the given polynomial matrix to obtain one of the singular vectors. The corresponding singular value and the other singular vector are obtained through a method outlined in Chapter 4 for the generalized polynomial power method. This approach also has a normalization-free variant, which follows the same principles as the PEVD approach.

While the PQRD cannot use a single iteration, any of the  $M$  columns can be normalized to obtain the columns of the paraunitary matrix  $\mathbf{Q}(z)$ . Then, an upper-right triangular matrix can be easily obtained with the help of  $\mathbf{A}(z)$  and  $\mathbf{Q}(z)$ . To avoid QR-based rank decomposition, the normalization-free approach directly estimates the columns of the paraunitary matrix  $\mathbf{Q}(z)$  from bin-wise QR results. The proposed normalization-free approach shows significant improvements compared to state-of-the-art algorithms.

To complete the paraunitary matrix, the Gram-Schmidt orthogonalization is extended to the polynomial domain, which can convert a reduced PSVD to full a PSVD. Thus, with the availability of any number of columns of a paraunitary matrix, the remaining columns can be easily estimated via this extension.

The use of any of the two approaches explained in this chapter can depend on the nature of the application and requirements. Because the original rank decomposition method is computationally expensive compared to its normalization-free variant, it is recommended to use a normalization-free variant. The only disadvantage of the normalization-free approach with respect to the original rank decomposition method is that it produces a solution with the same initialization choice, in which there is no possibility of a low-order polynomial vector, which can be expected with some initializations in the original rank decomposition method.

While these methods represent significant advancements over state-of-the-art algorithms, the decomposition order remains notably high, prompting the need for further investigation. This concern arises from the practical applications of paraunitary filter banks, which are often implemented on hardware platforms. The complexity of these filter banks is directly tied to the order of a decomposition. Additionally, the latency also generally increases with the filter length. Therefore, the focus of the next chapter will be to address the challenge of reducing the decomposition order while maintaining or improving execution time and accuracy.

## Chapter 6

# Unified Algorithm II: PEVD, PSVD and PQRD via Spectral Factorization of Laurent Polynomials

### 6.1 Introduction

The polynomial factorization methods presented in previous chapters result in high-order decompositions, which can be deemed computationally expensive for real-world applications. The reason for large polynomial order can be tied to the allpass ambiguity. Reducing this allpass factor to a simple delay can lead to a compact order decomposition. Notably, none of the methods proposed in previous chapters or the current time-domain iterative approaches that compute the PEVD, PSVD and PQRD address this issue. Unlike time-domain methods, the DFT based methods address this issue in order to obtain compact order decompositions [33–35, 47, 56]. However, it is important to note that these methods are more suitable for low order temporal and spatial dimension para-Hermitian or general polynomial matrices [33–35, 59]. The primary reason for such limitations lies in the complexity of the phase smoothing process, which is fundamental to achieving compact-order decomposition. As explained previously, the phase

smoothing procedure establishes phase coherence between the eigenvectors obtained independently in each DFT bin through EVD, leading to a compact-order time-domain eigenvector. This phase smoothing procedure involves numerical optimization and utilizes the Newton method in [35, 47] and Powell’s dogleg algorithm [55] in [33], both of which rely on the Hessian matrix inversion in each iteration. Furthermore, this procedure needs to be repeated for each eigen- or singular vector, rendering it rather costly for real-world applications.

Moreover, the SBR2 and SMD algorithms can no longer compete in the pursuit of providing compact-order decompositions. This holds also true for the SBR2 and SMD based PSVD and PQRD algorithms. As a result, this chapter introduces an entirely new approach to address this problem. A subtle hint in this direction was previously provided in the context of support estimation for analytic eigenvectors [86]. In that endeavour, the auto-correlation function — which is blind to any arbitrary phase terms — of a component of any analytic eigenvector was computed to estimate its support. However, the current idea is to make use of the auto- and possibly cross-correlation functions of the components of eigenvectors, which can be easily estimated with extreme accuracy, to spectrally factorize it in order to estimate the eigenvector’s components.

The spectral factorization of an eigenvector component’s auto-correlation function exists by definition, allowing for the application of spectral factorization methods [92]. However, many of these methods assume the para-Hermitian polynomial to be positive definite on the unit circle, which might not hold true for the auto-correlation function. While the true auto-correlation function remains non-negative, it may contain zeros on the unit circle. In the estimated function, truncation during the estimation process might even cause negativity on the unit circle. Consequently, not all spectral factorization methods are suitable for polynomial matrix decomposition.

The structure of this chapter is as follows: Section 6.2 provides an overview of the spectral factorization of para-Hermitian polynomials and reviews two spectral factorization methods that are applied in the context of polynomial matrix factorization. Section 6.3 introduces a time-domain aliasing metric in order to avoid unnecessary rep-

etition of all steps in the cepstral method at each DFT size until a sufficient DFT size is reached. Sections 6.4 through 6.6 apply this concept to PEVD, PSVD, and PQRD, respectively, and results are presented in Section 6.7. Section 6.8 provides a summary of the chapter.

## 6.2 Spectral Factorization of Para-Hermitian Laurent Polynomials

Let us assume that  $r_{ss}(z)$  is a Laurent polynomial having para-Hermitian symmetry i.e.  $r_{ss}(z) = r_{ss}^*(z^{-1}) = r_{ss}^P(z)$  and is positive semi-definite on the unit circle i.e.  $r_{ss}(e^{j\Omega}) \geq 0$ . This Laurent polynomial admits a spectral factorization

$$r_{ss}(z) = s(z)s^P(z), \quad (6.1)$$

where  $s(z)$  is an analytic function. In order to spectrally factorize  $r_{ss}(z)$ , most of the spectral factorization methods, namely the Cepstral method [77, 93], Bauer's method [92, 93], MinPh method [93], Wilson's method [94], impose the condition on the Laurent polynomial to be positive definite on the unit circle i.e.  $r_{ss}(e^{j\Omega}) > 0 \forall \Omega$ .

While  $s(z)$  is an unknown, its spectral density function or its autocorrelation function i.e.  $r_{ss}(z) = s(z)s^P(z)$  is known. From the available  $r_{ss}(z)$ , the objective is to spectrally factorize it such that the estimated spectral factor  $\hat{s}(z)$  has a polynomial order equal to half of the polynomial order of  $r_{ss}(z)$ . To facilitate this type of factorization, this thesis reviews two of the well-established methods which are further extended to compute the PEVD, PSVD and PQRD of a polynomial matrix. Readers seeking more detailed information are encouraged to consult [77, 92, 93, 95, 96].

### 6.2.1 Roots Method

This method estimates the roots of a given para-Hermitian Laurent polynomial, which subsequently facilitates the construction of spectral factors [93]. This estimation leverages the insight that complex roots off the unit circle occur in quadruples – two inside and two outside the unit circle. Conversely, real roots located off the unit circle ex-

hibit even multiplicity, comprising one root inside and one outside the circle. A similar pattern holds for roots situated on the unit circle. To compute these roots, one can estimate the eigenvalues of a companion matrix, which is constructed from the coefficients of  $r_{ss}(z)$  and corresponds to the Laurent polynomial [1]. The eigenvalues of the companion matrix can be determined using the QR decomposition [1, 93]. It should be noted that this method does not presuppose the strict positivity of  $r_{ss}(e^{j\Omega})$ .

As an illustration, consider the Laurent polynomial  $r_{ss}(z) = 4z^2 + 15z + 26 + 15z^{-1} + 4z^{-2}$ . To construct the companion matrix, the polynomial is first divided by 4 to make the last coefficient unity i.e. a monic polynomial, resulting in:

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & -15/4 & -26/4 & -15/4 \end{bmatrix}. \quad (6.2)$$

The eigenvalues of this companion matrix, which are  $\{-1.5 \pm 1.323j, -0.375 \pm 0.3307j\}$  correspond to the roots of the given polynomial. Two of the roots are situated inside the unit circle, forming a conjugate pair, while the other two are outside the unit circle. Consequently, the spectral factor  $s(z)$  can be constructed from various combinations of these roots, including using the roots inside the unit circle, those outside the unit circle, or a combination of one inside and one outside. For instance, the minimum-phase spectral factor can be determined as follows

$$s(z) = 4(z + 0.375 - 0.3307j)(z + 0.375 + 0.3307j) = 4z^2 + 3z + 1. \quad (6.3)$$

There is also a built-in function in MATLAB<sup>®</sup> called **roots** which utilizes the QR decomposition to compute polynomial roots. There are various other methods of finding roots of high order polynomials [97–99]. This method of spectral factorization can easily be used even in case of polynomials with roots on the unit circle unlike other methods.

## 6.2.2 Cepstral Method

This method is characterized by its simplicity, ease of implementation, and cost effectiveness. Initially, it was introduced to ascertain the presence and timing of echoes in the context of seismological data analysis [100]. Subsequently, this concept has found application in various fields, including speech processing and the broader domain of signal processing [77, 101]. It leverages the fact that for positive definite  $r_{ss}(z)|_{z=e^{j\Omega}}$ , the logarithm of  $r_{ss}(z)$  should exhibit properties of being an absolutely convergent, conjugate symmetric Laurent series

$$p(z) = \log(r_{ss}(z)) = \sum_{n \in \mathbb{Z}} p_n z^n . \quad (6.4)$$

The coefficients of  $p(z)$  decay at least as rapidly as an exponential function due to its analyticity, making it feasible to approximate it with a finite-order series in practical terms. One can split  $p(z)$  into left and right series as follows:

$$p(z) = p_-(z) + p_+(z) , \quad (6.5)$$

where  $p_+(z) = p_-^P(z) = p_0/2 + \sum_{n=1,2,\dots} p_n z^{-n}$ . The spectral factor  $s(z)$  is related to  $p_+(z)$  through an exponential function

$$s(z) = \exp(p_+(z)) . \quad (6.6)$$

This method is implemented in the DFT domain and therefore heavily relies on the FFT algorithm as evident from Algorithm 3, outlining the entire cepstral algorithm. It can be seen that the entire procedure is repeated at increasing DFT size unless the reconstruction error threshold  $\epsilon_{\zeta_r}$  is satisfied. There is no proper method to determine a sufficient DFT size. The review paper [93] also does not mention any direct method to determine a sufficient DFT size. However, in line with [102] and [103], which appear to have merely reproduced this work without introducing any innovations, the authors do not explicitly state the criteria for determining sufficiency. Instead, they propose setting the DFT size to be 10 to 50 times the order of  $r_{ss}(z)$ . Furthermore, [101] hints

---

**Algorithm 3:** Spectral Factorization via Cepstral Method [96, 100, 102–104]

---

**Input:**  $r_{ss}(z)$

**Output:**  $\hat{s}(z)$

1. evaluate  $r_{ss}(z)$  at  $z = e^{j\Omega_k}, k = 1, \dots, K$
  2. compute  $p(e^{j\Omega_k}) = \log(r_{ss}(e^{j\Omega_k}))$ ,  $k = 1, \dots, K$
  3. obtain  $\hat{p}[\tau]$  via  $K$ -point IFFT of  $p(e^{j\Omega_k}), k = 1, \dots, K$
  4. obtain  $p_+[\tau] = \begin{cases} \hat{p}[\tau]/2, & \tau = 0 \\ \hat{p}[\tau] & \tau > 0 \end{cases}$
  5. perform  $K$ -point FFT of  $p_+[\tau]$  to obtain  $p_+(e^{j\Omega_k}), k = 1, \dots, K$
  6. perform  $K$ -point IFFT of  $e^{p_+(e^{j\Omega_k})}, k = 1, \dots, K$  to obtain  $\hat{s}[\tau]$
  7. if  $\sum_{\tau} |r_{ss}[\tau] - \hat{s}[\tau] * \hat{s}^*[-\tau]| < \epsilon_{\zeta_r}$ , terminate, otherwise repeat at  $K = 2K$
- 

at the concept of time-domain aliasing but primarily in the context of comparing DFT with DTFT, without providing a specific metric or method to determine a sufficient DFT size for spectral factorization.

Furthermore, it is essential to highlight that the issue of a zero on the unit circle has not been adequately addressed in the existing literature. The proposed solutions often involve approximations or adding small biases, which can introduce errors and, hence, may not be suitable for polynomial matrix decomposition. Additionally, a zero in close proximity of the unit circle can introduce a dip in the power spectral density curve, necessitating a significantly larger DFT size for an accurate spectral factorization.

Therefore, the next section introduces a time-domain aliasing metric to ascertain if a DFT size for this method is sufficient or not to avoid unnecessary repetition of all steps within Algorithm 3 at each DFT size. However, it must be noted that the issue of zeros on the unit circle is not addressed. Instead, the roots method is used as an alternative.

## 6.3 Modified Cepstral Method

The cepstral method must undergo modifications to make it more cost-effective for polynomial matrix factorization. Hence, in Algorithm 3, before proceeding from step 3 to 4, it becomes necessary to measure the time-domain aliasing. This is because if the chosen value of  $K$  is insufficient to approximate  $\hat{p}[\tau]$  with minimal time-domain aliasing, further steps in the algorithm would be futile. Consequently, this additional check enhances the computational efficiency.

To assess the time-domain aliasing in the estimation of  $\hat{p}[\tau]$ , a time-domain aliasing metric  $\zeta_p$  can be defined in a manner similar to (4.42). To calculate this metric, step 3 should be adjusted to obtain  $\hat{p}^{(K)}[\tau]$  and  $\hat{p}^{(K/2)}[\tau]$  using  $K$  and  $K/2$ -point IFFTs, respectively. The algorithm should not proceed unless this metric is below a sufficiently considerably low threshold, denoted as  $\epsilon_{\zeta_p}$ . Meeting this threshold signifies that the chosen DFT size is sufficient for estimating  $\hat{p}[\tau]$ . After obtaining  $p_+[\tau]$ , the next step involves calculating  $\hat{s}[\tau]$  from  $e^{p_+(e^{j\Omega_k})}$  at minimum time-domain aliasing because it is implemented in the DFT domain. However, it is important to note that the DFT size at which  $\zeta_p$  meets the threshold condition is typically adequate for obtaining  $\hat{s}[\tau]$ , but it may not always be necessary. Therefore, it becomes advantageous to assess and compute time-domain aliasing independently.

## 6.4 PEVD Via Spectral Factorization

This section applies para-Hermitian Laurent polynomial spectral factorization methods to compute the PEVD of a para-Hermitian polynomial matrix.

### 6.4.1 Auto- and Cross-Correlation Functions of the Component of Analytic Eigenvectors

Given that phase smoothing is the computationally intensive step in deriving compact-order PEVD elements, and it has been established as a non-convex problem in [34], this thesis aims to circumvent the need for phase smoothing. Instead, the underlying approach involves calculating the autocorrelation function of the eigenvector compo-

---

**Algorithm 4: Modified Cepstral Method**


---

**Input:**  $r_{ss}(z)$ ,  $\epsilon_{\zeta_p}$

**Output:**  $\hat{s}(z)$

1. evaluate  $r_{ss}(z)$  at  $z = e^{j\Omega_k}$ ,  $k = 1, \dots, K$
  2. compute  $p(e^{j\Omega_k}) = \log(r_{ss}(e^{j\Omega_k}))$ ,  $k = 1, \dots, K$
  3. obtain  $\hat{p}[\tau]$  via  $K$ -point IFFT of  $p_k$ ,  $k = 1, \dots, K$
  4. if  $\zeta_p > \epsilon_{\zeta_q}$ , goto 1 with  $K = 2K$
  5. obtain  $p_+[\tau] = \begin{cases} \hat{p}[\tau]/2, & \tau = 0 \\ \hat{p}[\tau] & \tau > 0 \end{cases}$
  6. perform  $K$ -point FFT of  $p_+[\tau]$  to obtain  $p_+(e^{j\Omega_k})$ ,  $k = 1, \dots, K$
  7. perform  $K$ -point IFFT of  $e^{p_+(e^{j\Omega_k})}$  to obtain  $\hat{s}[\tau]$
  8. if  $\sum_{\tau} |r_{ss}[\tau] - \hat{s}[\tau] * \hat{s}^*[-\tau]| < \epsilon_{\zeta_r}$  terminate, otherwise goto 6 with  $K = 2K$
- 

nents, followed by the application of the spectral factorization techniques outlined in Section 6.2.

As discussed in Section 6.1, it is important to maintain phase coherence between the eigenvectors of adjacent bins to achieve a compact-order estimation for  $\mathbf{Q}(z)$ . However, it is worth noting that the autocorrelation function of the components of an analytic eigenvector can be directly obtained from these bin-wise eigenvector components without any phase smoothing. For instance, if  $r_{nn,m}(z) = q_{n,m}(z)q_{n,m}^P(z)$  represents the  $z$ -transform of the autocorrelation function of the  $n$ th component of the  $m$ th analytic eigenvector, its samples on the unit circle can be extracted directly from the components of bin-wise eigenvectors such that

$$r_{nn,m}(e^{j\Omega_k}) = q_{n,m}(e^{j\Omega_k})q_{n,m}^*(e^{j\Omega_k}) = |q_{n,m}(e^{j\Omega_k})|^2 = |e^{j\phi_{m,k}}q_{n,m,k}|^2 = |q_{n,m,k}|^2, \quad (6.7)$$

where  $q_{n,m,k}$  is the  $n$ th component of bin-wise eigenvector  $\mathbf{a}_{m,k}$ . The samples of  $r_{nn,m}(z)$  on the unit circle are effectively the absolute squared values of the corresponding bin-wise eigenvector components. These values can be readily obtained from a bin-wise EVD. Consequently, it becomes feasible to accurately estimate  $r_{nn,m}(z)$  using a suffi-

ciently large  $K$ -point IFFT where  $K$  is chosen to minimize aliasing in  $\hat{r}_{nm,m}[\tau]$ . The time-domain aliasing of the resulting  $\hat{r}_{nm,m}[\tau]$  can be measured via a metric  $\zeta_{r_q}$ , defined akin to (4.42). This metric, as discussed earlier, relies on obtaining  $\hat{r}_{nm,m}^{(K)}[\tau]$  and  $\hat{r}_{nm,m}^{(K/2)}[\tau]$  at two different DFT sizes, namely  $K$  and  $K/2$ . As this metric quantifies the normalized difference between these two estimates, a small threshold  $\epsilon_{\zeta_{r_q}}$  can be deemed suitable for all scenarios. Consequently, the process is reiterated with increasing DFT sizes until  $\zeta_{r_q}$  drops below  $\epsilon_{\zeta_{r_q}}$ . It is important to note that a lower value of  $\epsilon_{\zeta_{r_q}}$  indicates a more accurate estimate, albeit at the expense of having a larger time-domain support. It is crucial to highlight that (6.7) remains valid when none of the bins has eigenvalues with multiplicity greater than one, and this condition holds true for any estimated para-Hermitian matrix, as established in Chapter 3.

The above analysis is shown for an auto-correlation function of the  $n$ th component of the  $m$ th eigenvector, a similar analysis is possible for cross-correlation between any two different components of the  $m$ th eigenvector. For example  $r_{n\ell,m}(z) = q_{n,m}(z)q_{\ell,m}^P(z)$  is a cross-correlation between the  $n$ th and  $\ell$ th component of the  $m$ th eigenvector. Its sample points on the unit circle can be obtained from the bin-wise component as

$$r_{n\ell,m}(e^{j\Omega_k}) = q_{n,m}(e^{j\Omega_k})q_{\ell,m}^*(e^{j\Omega_k}) = e^{j\phi_{m,k}}q_{n,m,k}e^{-j\phi_{m,k}}q_{\ell,m,k}^* = q_{n,m,k}q_{\ell,m,k}^*, \quad (6.8)$$

where again the phase ambiguity cancels out. Thus both the auto- and cross-correlation sequences of the components of an eigenvector can be estimated via the proposed method. The entire procedure for estimating  $\hat{r}_{n\ell,m}(z)$ ,  $n, \ell = 1, \dots, M$  is outlined in Algorithm 5. The resulting autocorrelation function i.e. for  $\ell = n$  is a para-Hermitian Laurent polynomial in the  $z$ -domain and so spectral factorization can be performed. Before the entire procedure of computing the PEVD is discussed, the autocorrelation function estimation can be further used to estimate the support of the analytic eigenvector so that the spectral factor  $\hat{s}[\tau]$  can be truncated appropriately.

### 6.4.2 Support Estimation of Analytic Eigenvector

As previously mentioned, an analytic eigenvector exhibits an allpass ambiguity, which can extend its support infinitely even if it was originally finite. However, if this allpass

---

**Algorithm 5:** Auto- and cross-correlation function of the components of an analytic eigenvector

---

**Input:**  $\mathbf{R}(z)$ ,  $m$ ,  $n$ ,  $\ell$ ,  $\epsilon_{\zeta_{r_q}}$

**Output:**  $\hat{r}_{n\ell,m}$

set initial  $K$ ;  $\zeta_{r_q} = 1$ ;

**while**  $\zeta_{r_q} > \epsilon_{\zeta_{r_q}}$  **do**

$\mathbf{Q}_k \leftarrow \text{EVD}(\mathbf{R}_k), k = 1, \dots, K$ ;

$r_{n\ell,m}(e^{j\Omega_k}) = q_{n,m,k} q_{\ell,m,k}^*$ ;

$\hat{r}_{n\ell,m}^{(K)}[\tau] \leftarrow K$ -point IFFT of  $r_{n\ell,m}(e^{j\Omega_k}), k = 1, \dots, K$ ;

$\hat{r}_{n\ell,m}^{(K/2)}[\tau] \leftarrow K/2$ -point IFFT of  $r_{n\ell,m}(e^{j\Omega_k}), k = 1, 3, 5, \dots, (2K - 1)$ ;

$\zeta_{r_q} = \sum_{\tau} |\hat{r}_{n\ell,m}^{(K)}[\tau] - \hat{r}_{n\ell,m}^{(K/2)}[\tau]|^2 / \sum_{\tau} |\hat{r}_{n\ell,m}^{(K)}[\tau]|^2$ ;

$K \leftarrow 2K$

**end**

$\hat{r}_{n\ell,m} \leftarrow \hat{r}_{n\ell,m}^{(K/2)}(z)$

---

factor takes on a simple delay, an analytic eigenvector can be approximated sufficiently accurately with a compact order polynomial. The support of the analytic eigenvector can be determined from the estimated autocorrelation function of its components.

In the case of a signal with support length  $N$ , the time-domain support of its autocorrelation function is typically  $2N - 1$ . Applying a similar concept to the autocorrelation function of the components of analytic eigenvectors, the support of the analytic eigenvector would fall within the bounds of  $K/8 < \hat{N} \leq K/4$  when  $\hat{r}_{n\ell,m}[\tau]$  is estimated using Algorithm 5. However, since the DFT size is increased by a factor of 2, the support of the analytic eigenvector needs further refinement i.e. the bounds need to be decreased. This can be achieved by either truncating or by conducting additional iterations within Algorithm 5, where the DFT size varies between  $K/2$  and  $K$ , until the difference in the DFT sizes over which the  $\zeta_{r_q}$  metric is computed becomes small. For more detailed information, interested readers are referred to [86].

### 6.4.3 Spectral Factorization Ambiguity in Context of PEVD

For any given para-Hermitian polynomial  $r_{ss}(z)$ , the spectral factors can be categorized as minimum phase, maximum phase, or mixed phase. The number of potential choices for  $p_+(z)$  depends on the order of  $r_{ss}(z)$ . For example, let us consider  $r_{ss}(z) = 6(z^{-2} + z^2) + 35(z^{-1} + z) + 62$ , a para-Hermitian polynomial of order 4 with

roots  $\{-1/3, -1/2, -2, -3\}$ . Although, in the field of signal and system theory, there is often a preference for minimum phase systems, it is essential to recognize that the possible choices for the roots of  $p_+(z)$ , the spectral factor of  $r_{ss}(z)$ , can be numerous:

$$\{-1/3, -2\}, \{-3, -2\}, \{-3, -1/2\},$$

where each pairing represents the roots of a valid spectral factor of  $r_{ss}(z)$ . The number of possible combinations increases with the order of  $r_{ss}(z)$ . However, in the context of the PEVD, only one specific combination results in the lowest order for all components of any eigenvector. For instance, when we consider the spectral factor of  $r_{11,m}(z) = q_{1,m}(z)q_{1,m}^P(z)$ , which we refer to as  $p_{q_{1,m}}(z)$ , it must be such that it shares all of its zeros with  $r_{1\ell,m}(z) = q_{1,m}(z)q_{\ell,m}^P(z)$ , where  $\ell = 2, \dots, M$ . In summary,  $r_{1\ell}(z)/p_{q_{1,m}}(z)$  must have the lowest possible order, which can only be achieved when all the zeros of the spectral factor of  $r_{11,m}(z)$  coincide with the zeros of  $r_{1\ell,m}(z)$ . Otherwise, if zeros are not common,  $r_{1\ell}(z)/p_{q_{1,m}}(z)$  will have an infinite order. Even if an attempt is made to approximate it with a finite-order polynomial, the resulting approximation order will be significantly higher than the potential compact order in case of common zeros. This ambiguity in spectral factorization has a direct impact on the order of the resulting eigenvectors.

#### 6.4.4 PEVD Method via Cepstral Method

The cepstral method performs spectral factorization without addressing the ambiguity mentioned above. Consequently, it is expected that the decomposition order obtained through this method will be higher than necessary. To compute the PEVD of a para-Hermitian polynomial matrix  $\mathbf{R}(z)$ , the auto-correlation function of any component of the eigenvector is obtained using Algorithm 5 (i.e., for  $n = \ell$ ). Let us assume that  $\hat{r}_{n'n',m}(z)$  is estimated using Algorithm 5. Its spectral factorization via the modified cepstral method will produce  $\hat{p}_{q_{n',m}}(z)$  which will have same magnitude as that of  $q_{n,m}(z)$  but differ in phase if evaluated on the unit circle. Therefore, this spectral factor can be considered as an estimate of  $q_{n,m}(z)$  and can be used to phase-smooth

the remaining component as follows:

$$q'_{n,m,k} = q_{n,m,k} \frac{\hat{p}_{q_{n',m}}(e^{j\Omega_k})}{q_{n',m,k}}, \quad \text{for } n = 1, \dots, M, k = 1, \dots, K. \quad (6.9)$$

Applying this phase i.e.  $\frac{\hat{p}_{q_{n',m}}(e^{j\Omega_k})}{q_{n',m,k}}, k = 1, \dots, K$  from the spectral factor to the bin-wise components of the  $m$ th eigenvector will make it smooth. The resulting time-domain sequence of all  $M$  components of the  $m$ th analytic eigenvector can be obtained via an IFFT of  $q'_{n,m,k}$  where the DFT size should be iteratively increased until the time-domain aliasing

$$\sum_{\tau} |\hat{\mathbf{q}}_m^H[-\tau] \hat{\mathbf{q}}_m[\tau] - \delta[\tau]|_2^2$$

becomes negligible. Here  $\hat{\mathbf{q}}_m[\tau]$  denotes the time-domain sequence of  $q'_{n,m,k}$  for  $n = 1, \dots, M, k = 1, \dots, K$ . It is worth noting that a similar issue of division by zero can arise when applying the phase to bin-wise eigenvector components in (6.9). Consequently, the up-sampling and sign change solution, as previously described in Section 5.2.6, should be applied.

It should be evident that the time-domain support of  $\hat{q}_{n',m}[\tau]$  will be compact as it was phase smooth via its spectral factor  $\hat{p}_{q_{n',m}}(z)$  but the remaining components of the same eigenvector  $\hat{q}_{n,m}[\tau], n \neq n'$  will be of large order. Therefore, one might think of performing spectral factorization of the auto-correlation function of each component of the  $m$ th eigenvector independently in order to estimate the eigenvector with compact support. However, this is not possible because each component of the eigenvector will be subjected to a different phase adjustment and hence the vector will not retain the properties of an eigenvector. Therefore, the only possible option is to obtain one of the component of an eigenvector through the spectral factorization of its auto-correlation function estimated through Algorithm 5 while the remaining  $M - 1$  eigenvector components can be estimated via (6.9).

### 6.4.5 PEVD via Roots Method

This method is straightforward but can be considerably more expensive than the cepstral method as it involves finding polynomial roots. The procedure requires the autocorrelation function of any component of an eigenvector and the cross-correlation function of that component with another i.e.  $r_{n'n',m}(z) = q_{n',m}(z)q_{n',m}^P(z)$  and  $r_{n'\ell,m}(z) = q_{n',m}(z)q_{\ell,m}^P(z)$  where  $\ell \neq n'$ . These two functions can be estimated via Algorithm 5. Once estimated, the roots of both these Laurent polynomials are determined. Since  $q_{n',m}(z)$  is a common factor between  $r_{n'n',m}(z)$  and  $r_{n'\ell,m}(z)$ , half the number of roots of  $r_{n'n',m}(z)$  will be common with  $r_{n'\ell,m}(z)$  and these roots belong to  $q_{n',m}(z)$ . Therefore, the roots of both these polynomials should be compared and matched, and the common roots can be used to construct a polynomial  $\hat{p}_{q_{n',m}}(z)$  such that  $r_{n'n',m}(z) = g^2 \hat{p}_{q_{n',m}}(z) \hat{p}_{q_{n',m}}^P(z)$  where  $g \in \mathbb{R}$ . Hence the estimate of  $q_{n',m}(z)$  will be  $g \hat{p}_{q_{n',m}}(z)$  where

$$g = \sqrt{\frac{r_{n'n',m}[0]}{\sum_{\tau} |\hat{p}_{q_{n',m}}[\tau]|^2}},$$

with  $r_{n'n',m}[\tau] \bullet \circ r_{n'n',m}(z)$  and  $\hat{p}_{q_{n',m}}[\tau] \bullet \circ \hat{p}_{q_{n',m}}(z)$ . This estimate of  $q_{n',m}(z)$  can now be used to phase smooth the remaining bin-wise components akin to (6.9). This root matching method produces eigenvectors with order equal to the ground truth order if the ground truth autocorrelation function is known. For an estimated auto- and cross-correlation function where half of the roots are matching, the resulting order of the eigenvector will also be compact but may not be equal to the ground truth order. The downside of this root matching approach is that for each eigenvector extraction, two  $(2N)$ th order polynomial roots are estimated where  $N$  is the order of the ground-truth analytic eigenvector.

**Example 13.** Lets consider a para-Hermitian matrix from Section 4.3.5. This example demonstrates the estimation of  $\mathbf{q}_1(z)$  via roots method. Similar example with the cepstral method is not possible due to zeros on the unit circle for the considered para-Hermitian matrix. For sake of explanation,  $r_{11,1}(z)$  and  $r_{12,1}(z)$  are assumed to be

known which in practice can also be estimated via Algorithm 5. Roots of these two Laurent polynomials are

$$\begin{aligned}\text{roots}\{r_{11,1}(z)\} &= [0.2778, 0.0334 \pm 0.3025j, -1, -1, 0.3611 \pm 3.266, 3.6] \\ \text{roots}\{r_{12,1}(z)\} &= [0.0334 \pm 0.3025j, -1, 1, 1, -1, 3.6]\end{aligned}$$

which shows that there are roots of  $r_{11,1}(z)$  which are also roots of  $r_{12,1}(z)$ . Therefore,  $\hat{p}_{q_1,1}(z)$  is constructed from  $[3.6, 0.0334 \pm 0.3025j, -1]$  and its phase is then applied to all bin-wise components of  $\mathbf{q}_{1,k}$  for  $k = 1, \dots, K$  similar to (6.9) which produces  $\mathbf{q}'_{1,k}$ . The time-domain eigenvector, obtained via an IFFT of  $\mathbf{q}'_{1,k}$ , which if truncated via an threshold of  $10^{-10}$  produces same order as that of the ground truth i.e. 5. The resulting eigenvalue estimated through this estimated eigenvector results in  $\xi_{\lambda_1} = 1.4 \times 10^{-9}$ , which is sufficiently accurate. This demonstrates that the roots matching method extracts compact order eigenvector if the ground truth eigenvector components' auto- and cross-correlation function is known.  $\triangle$

While this method performs effectively when roots are accurately estimated and both the auto- and cross-correlation functions share common roots, it does face a significant challenge related to root mismatch. Because both the auto- and cross-correlation functions are estimated, errors in the estimation process can perturb the roots, resulting in mismatch. In practical situations, relying on a matching approach is often not feasible, and determining a suitable threshold to compare the roots of both sets is a complex task. Furthermore, when dealing with a considerably large temporal dimension, employing a root finding method can be computationally expensive. Hence, an approach based on function evaluations can be employed to determine the roots necessary for constructing the spectral factor of the auto-correlation function.

Let us assume that  $r_{n'n',m}^{(i)}, i = 1, \dots, 2N$  represents the roots of  $r_{n'n',m}(z)$  arranged in ascending order of the absolute value. Due to the para-Hermitian symmetry of  $r_{n'n',m}(z)$ ,  $r_{n'n',m}^{(i)}, i = 1, \dots, N$  will be located inside or on the unit circle, while the remaining roots will be outside or on the unit circle. As a result, the spectral factor can include either  $r_{n'n',m}^{(i)}$  or its conjugate inverse  $1/r_{n'n',m}^{(i)*} = r_{n'n',m}^{(N-i+1)}$  as a root for any

given value of  $i$ , but not both simultaneously. To select the roots for constructing the spectral factor, one can determine this based on the minimum value of the function  $r_{n'\ell,m}(z)$  by evaluating it over these roots as

$$\min\{r_{n'\ell,m}(r_{n'n',m}^{(i)}), r_{n'\ell,m}(r_{n'n',m}^{(2N-i+1)})\}, i = 1, \dots, N.$$

The root that yields the minimum value is chosen, while its conjugate reciprocal is discarded. After constructing the spectral factor  $\hat{p}_{qn',m}(z)$ , the remaining components can be obtained through phase smoothing using the constructed spectral factor according to (6.9). This approach, although involving  $2N$  function evaluations, only requires roots of a single polynomial of order  $2N$ . As a result, it is less computationally expensive compared to the root-matching approach.

Following this strategy, it becomes possible to estimate any number of eigenvectors independently. Consequently, this method is entirely parallelizable and thus e.g. compatible with FPGAs.

## 6.5 PSVD via Spectral Factorization

Similar to the PEVD, the PSVD can be computed by estimating the auto- and cross-correlation functions of either the right or left singular vectors' components and then applying the roots method. Once one of these paraunitary matrices is estimated, for instance, let us say  $\mathbf{V}(z)$  is estimated, the left singular vectors can be estimated through an IFFT of

$$\mathbf{u}'_{m,k} = \mathbf{u}_{m,k} \frac{-\hat{p}_{v_{n',m}}(e^{j\Omega_k})}{v_{n',m,k}}, \quad m = 1, \dots, L, k = 1, \dots, K, \quad (6.10)$$

where  $\hat{p}_{v_{n',m}}(z)$  is the spectral factor of  $v_{n',m}(z)v_{n',m}^P(z)$  and  $v_{n',m,k}$  is the bin-wise  $n'$ th component of  $\mathbf{v}_{m,k}$ . With the phase smoothing process carried out in each bin, applying an IFFT will yield  $\hat{\mathbf{u}}_m[\tau]$ . It is essential to increase the DFT size until the time-domain aliasing in  $\hat{\mathbf{u}}_m[\tau]$  becomes negligible. Subsequently, the corresponding singular values can be estimated as  $\hat{\sigma}_m(z) = \hat{\mathbf{u}}_m^P(z)\mathbf{A}(z)\hat{\mathbf{v}}_m(z)$ ,  $m = 1, \dots, L$  to obtain  $\hat{\Sigma}(z)$  having on

its diagonal the estimated singular values.

To complete the paraunitary matrix  $\mathbf{U}(z)$  from an initial size of  $M \times L$  to a final size of  $M \times M$ , there are two possible approaches. In case where  $M = L + 1$ ,  $\mathbf{u}_M(z)$  can be obtained in a manner similar to the previous singular vectors. This involves determining the auto- and cross-correlation functions using the bin-wise singular vectors and then applying the roots method for spectral factorization. However, if  $M > L + 1$ , the auto- and cross-correlation functions of the components of these singular vectors cannot be derived from the bin-wise singular vectors, specifically  $\mathbf{u}_{m,k}$  for  $m = L + 1, \dots, M$ , using Algorithm 5. The reason for this is that the singular values from  $L + 1$  to  $M$  are zero, leading to an algebraic multiplicity in each bin. Consequently, only an  $(M - L)$  dimensional subspace is defined, within which smooth 1-d subspaces would have to be first found before phase smoothing can be applied within each individual 1-d subspace. This process is beyond the methods in this thesis or, in fact, beyond any methods yet developed in the area of polynomial matrix factorisations. As a result,  $u_{n',m,k}u_{\ell,m,k}$  is likely to exhibit discontinuities for any DFT size, making it impossible to determine the auto- and cross-correlation functions of the components of  $\mathbf{u}_m(z)$  for  $m = L + 1, \dots, M$ . In simpler terms, the vectors  $\mathbf{u}_{m,k}$  can be ambiguous up to an arbitrary unitary matrix. This means that if we have  $\mathbf{U}_{o,k} = [\mathbf{u}_{L+1,k}, \dots, \mathbf{u}_{M,k}] \in \mathbb{C}^{M \times (M-L)}$ , then  $\mathbf{U}_{o,k}\mathbf{B}_k$ , where  $\mathbf{B}_k \in \mathbb{C}^{(M-L) \times (M-L)}$ , represents a unitary matrix [34]. There is an unknown matrix  $\mathbf{B}_k$  in every bin, and it therefore cannot be resolved as in [34] by interpolating across isolated algebraic multiplicities. Therefore, the only possible option is to adopt the polynomial extension of the Gram-Schmidt orthogonalization procedure described in Chapter 5. Since  $\mathbf{u}_{m,k}$  cannot be relied upon to estimate its auto- and cross correlation function for spectral factorization due discontinuities,  $\mathbf{w}_m(e^{j\Omega_k})$  can be obtained via (5.13) to estimate these correlation functions, one auto and one cross-correlation function. One may estimate  $\mathbf{u}_m(z)$  directly from the IFFT of  $\mathbf{w}_m(e^{j\Omega_k})$  at sufficient DFT size, but it will result in higher order polynomial due to the phase ambiguity introduced due to bin-wise normalization as given in (5.14). Therefore, instead  $w_{n',m}(z)w_{n',m}^P(z)$  and  $w_{n',m}(z)w_{\ell,m}^P(z)$  are estimated from  $\mathbf{w}_m(e^{j\Omega_k})$  via Algorithm 5. In this case, the phase ambiguity would cancel out. Subsequently, its spectral factorization via the roots

method will provide the compact order  $(L + 1), \dots, M$ th singular vectors.

## 6.6 PQRD via Spectral Factorization

In the PQRD factorization, the auto- and cross-correlation function of the components of the columns of  $\underline{\mathbf{Q}}(z)$  can similar be obtain via Algorithm 5 however by replacing the EVD by the QR decomposition. The auto- and cross-correlation function can be used to estimate the respective columns via the roots method employing function evaluations. With all columns estimated,  $\underline{\mathbf{R}}(z)$ , the upper right-triangular matrix, is estimated through the estimated paraunitary matrix  $\underline{\mathbf{Q}}(z)$ . For  $M = L + 1$ , the  $M$ th column of  $\underline{\mathbf{Q}}(z)$  can be obtained similarly as all previous columns by estimating the auto- and cross-correlation function followed by a spectral factorization. However, for  $M > L + 1$ , the polynomial Gram-Schmidt orthogonalization has to be applied to determine  $L + 1, \dots, M$  indexed columns of  $\underline{\mathbf{Q}}(z)$ . The polynomial Gram-Schmidt orthogonalization might produce orthonormal vectors with unnecessary large polynomial order due to the allpass factor i.e.  $\hat{\mathbf{w}}_m(z) = \phi_m(z)\underline{\mathbf{q}}_m(z)$ . Therefore, similar to the discussion in Section 6.5,  $w_{n',m}(z)w_{n',m}^P(z)$  and  $w_{n',m}(z)w_{\ell,m}^P(z)$  can be estimated from  $\mathbf{w}_m(e^{j\Omega_k})$  via Algorithm 5 which is blind to the allpass factor. Thereafter, its spectral factorization through the roots method can be performed to obtain compact order vectors.

## 6.7 Simulation and Results

In this section, the proposed spectral factorization-based polynomial matrix decomposition technique is simulated for all three categories i.e. PEVD, PSVD, and PQRD. The roots finding based approach is compared against state-of-the-art algorithms the performance metrics detailed in Section 5.8.

### 6.7.1 PEVD

Here the comparison of the proposed approach against state-of-the-art algorithms is drawn over an ensemble of spectrally majorised instantiations of  $\mathbf{R}(z) \in \mathbb{C}^{4 \times 4}$ . The instantiation are created using the source model in [22] where the order of  $\mathbf{Q}(z)$  is

varied from 100 to 500 in steps of 100 and  $\mathbf{A}(z)$  from 200 to 1000 in steps of 200. For every order of  $\mathbf{Q}(z)$ , 100 instances are generated. Due to how the ground-truth paraunitary  $\mathbf{Q}(z)$  is generated via elementary paraunitary operations, its outer lags decays to significantly small values; and so are the coefficients of  $\mathbf{R}[\tau]$ . Therefore,  $\mathbf{R}[\tau]$  is truncated symmetrically around  $\tau = 0$  via a significantly low threshold  $\mu_{\text{PH}} = 10^{-20}$ . This truncation removes the exterior lags having  $\mu_{\text{PH}}$  times of the total energy.

The proposed method estimates an auto- and cross-correlation function via Algorithm 5 with  $\epsilon_{\zeta_{r_q}} = 10^{-16}$ . Thereafter, the roots method is applied to estimate the eigenvector and then the corresponding eigenvalue. The estimated eigenvector is truncated using a threshold of  $\mu_{\text{PU}} = 10^{-10}$ , employing a method similar to [27]. For the SBR2 and SMD algorithms, a maximum of 500 iterations is allowed, but termination occurs if the off-diagonal energy threshold falls below  $10^{-8}$ . As both of these methods are iterative algorithms, intermediate parunitary matrices are truncated using  $\mu_{\text{PU}} = 10^{-10}$ , employing the row-shifted corrected truncation method proposed in [27], which is currently the most promising truncation method for these iterative techniques. The normalization-free algorithm from Chapter 5 is executed with  $\epsilon_{\zeta_q} = \mu_{\text{PU}} = 10^{-6}$  and  $K_{\text{max}} = 2^{\lceil \log_2[\mathcal{O}(\mathbf{R}(z))] \rceil + 5}$ .

The ensemble results are presented in Fig. 6.1, where it is evident that the reconstruction error for SBR2, SMD and normalization-free approach is significantly high whereas the proposed roots based PEVD method achieves an error of orders of  $10^{-10}$ . This highlights that, with the settings mentioned above, the reconstruction error for SBR2 and SMD is approximately  $10^6$  times larger than the roots method, denoted as RPEVD in Fig. 6.1. Moreover, the order of the resulting eigenvectors with the proposed method is close to the ground-truth for lower order ground truth order but eventually falls below the ground-truth. This discrepancy in order is due to the fact that the trailing coefficients of the ground-truth eigenvectors are small, and thus an approximation with lower-order polynomials is possible. In contrast, SBR2 and SMD, despite utilizing row-shifted corrected truncation [27], still yield eigenvectors with large orders. It is important to emphasize that the high-order eigenvectors produced by SBR2 and SMD can only yield poor reconstruction errors, as demonstrated in Fig. 6.1(a). This

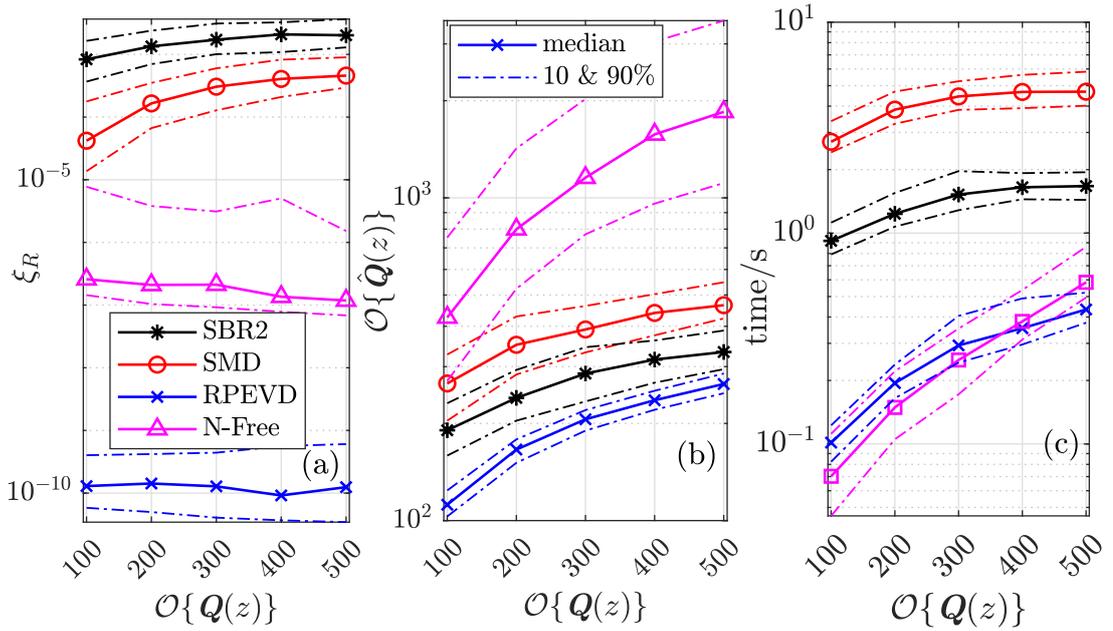


Figure 6.1: Performance metrics computed over an ensemble of randomized para-Hermitian matrices for SBR2 [13], SMD [22], normalization free variant of the rank one decomposition based PEVD (here denoted as N-Free) and roots based PEVD (RPEVD) with (a)  $\xi_R$ , (b)  $\mathcal{O}\{\hat{Q}(z)\}$  and (c) execution time in seconds.

implies that the order would need to be increased even more if a lower reconstruction error is desired. Here it is also important to emphasize that the normalization free variant have resulted in large polynomial order than the SBR2 and SMD which seems contradictory to Chapter 5 simulation results. In fact, this difference is due to the use of row-shifted corrected truncation approach for both SBR2 and SMD. Furthermore, the execution time of the proposed roots method and the normalization-free method is orders of magnitude lower than that of both SBR2 and SMD. Once again, if SBR2 and SMD are required to produce better results, the execution time gap between the proposed method and these two iterative methods will widen. If analysed minutely, the normalization-free approach, from Chapter 5, execution time is least of all four algorithms upto  $\mathcal{O}\{Q(z)\} = 300$  and then exceeds that of the roots method for higher orders of  $Q(z)$ .

This thesis have not included a comparison of the proposed method with the analytic PEVD method [33–35], which is more suitable for para-Hermitian matrices of small

spatial and temporal dimensions. Specifically, the eigenvector phase smoothing process in the analytic method involves optimizing an objective function via the Newton or Powell’s dogleg [55] optimization techniques which computes matrix inversion of an order  $K \times K$ , where  $K > N$  with  $N$  being the order of eigenvectors, in each iteration.

### 6.7.2 PSVD

The proposed algorithm is compared against the GSBR2 [11] and the approach which employs two PEVDs in order to compute a PSVD. For computing the PEVDs, the SMD algorithm is utilized. All methods are tested over an ensemble containing 100 randomized polynomial matrices  $\mathbf{A}(z) \in \mathbb{C}^{4 \times 4}$  for each value of  $\mathcal{O}\{\mathbf{U}(z)\} = \mathcal{O}\{\mathbf{V}(z)\}$  which is jointly varied from 100 to 500 in steps of 100. The outer lags of  $\mathbf{A}[n]$   $\bullet \circ$   $\mathbf{A}(z)$  may become very small, therefore each instance in the ensemble is truncated to eliminate its trailing outer lags to ensure such that the reconstruction error remains below  $10^{-20}$ . The proposed method and the SMD algorithm (utilized for PEVDs to compute the PSVD) are executed with the parameter settings of Section 6.7.1. The GSBR2 algorithm is allowed to run for a maximum of 200 iterations. However, it terminates before reaching the iteration limit if the maximum off-diagonal energy falls below  $10^{-6}$ . The intermediate paraunitary truncation threshold set to  $\mu_{\text{PU}} = 10^{-5}$ . Both the GSBR2 and SMD are executed with row-shifted corrected truncation strategy for lower order paraunitary matrices [27].

Ensemble results are depicted in Figure 6.2, where the reconstruction error achieved by the proposed algorithm is orders of magnitude lower than that of both iterative methods. In fact, the proposed method attains a reconstruction error that is  $10^8$  times lower than that achieved by the GSBR2 and 2-SMDs. While the orders of singular vectors obtained by the iterative methods may seem comparable to the order generated by the proposed method, it is important to note that these singular vectors are not accurate enough to diagonalize the underlying polynomial matrix effectively as reflected by high reconstruction error. On the other hand, the proposed method achieves a negligible normalized reconstruction error with extremely compact-order paraunitary matrices, and it does so in significantly less time as evidenced by Fig. 6.2(d). Therefore, the

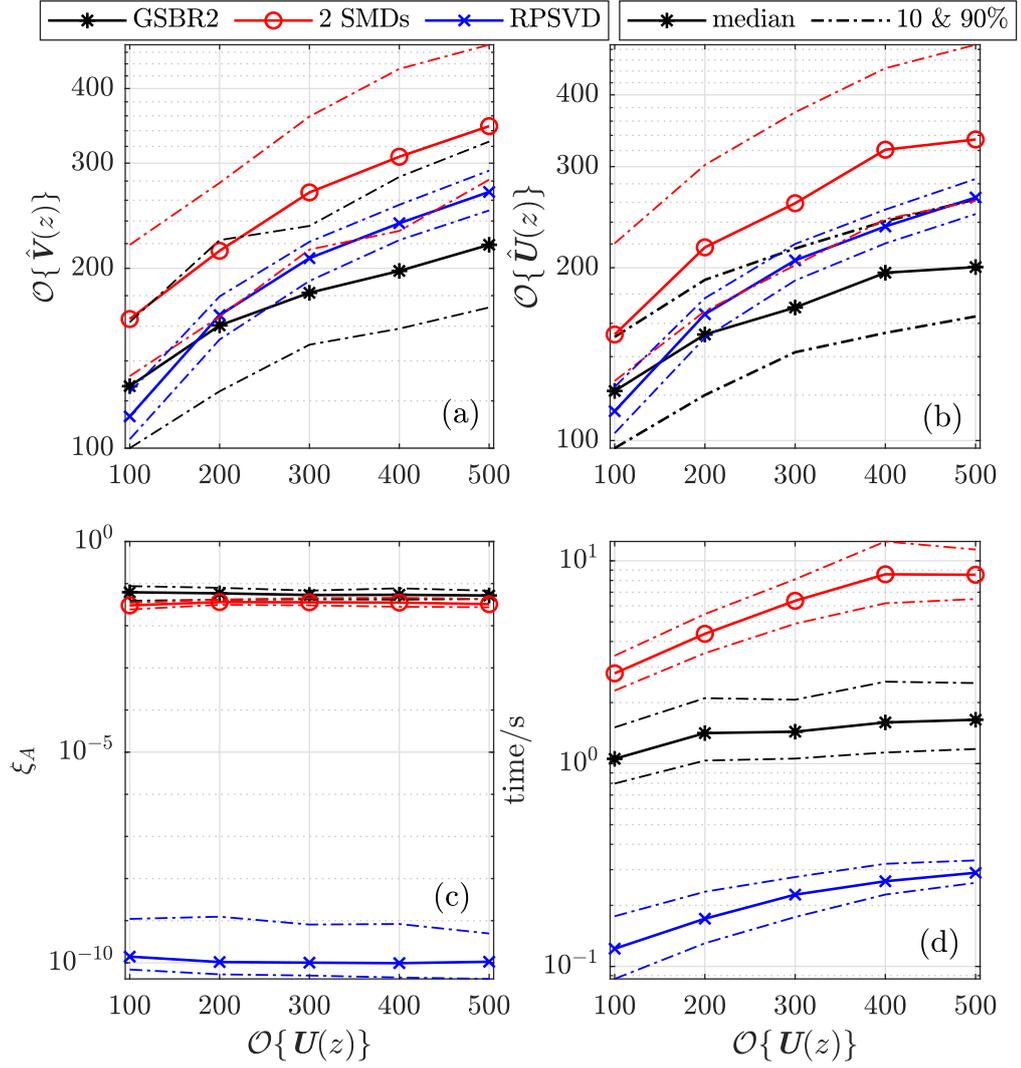


Figure 6.2: Performance metrics computed over an ensemble of randomized polynomial matrices for GSRB2 [11], 2 PEVDs based PSVD via SMD and roots based PSVD (RPEVD) with (a)  $\xi_A$ , (b)  $\mathcal{O}\{\hat{Q}(z)\}$  and (c) execution time in seconds.

proposed method outperforms state-of-the-art algorithms equally in all performance metrics. In addition, the proposed algorithm estimates each singular vectors separately which makes it highly parallelizable compared to all state-of-the-art algorithms. Normalization free approach is not included in this comparison because it performs similar to as shown in case of the PEVD illustrated in Fig. 6.1.

### 6.7.3 PQRD

Similar to the rank decomposition-based PQRD algorithm in Chapter 5, the proposed PQRD approach, referred to here as RPQRD, is compared against PQRD-BC [12] and SM-PQRD [19]. This comparison is carried out using an ensemble of randomized matrices  $\mathbf{A}(z) \in \mathbb{C}^{4 \times 4}$ . The ensemble is constructed following the methodology described in Section 5.8.4. However, in this case, the ground-truth order for  $\underline{\mathbf{Q}}(z)$  and the upper triangular matrix  $\underline{\mathbf{R}}(z)$  have higher orders. The ensemble is generated separately for each value of  $\mathcal{O}\{\mathbf{A}(z)\}$  in the range  $\{200, 400, \dots, 1000\}$ , where the orders of  $\mathcal{O}\underline{\mathbf{Q}}(z)$  and  $\mathcal{O}\underline{\mathbf{R}}(z)$  vary in the range  $\{100, 200, \dots, 500\}$ . Both PQRD-BC and SM-PQRD are simulated using the same settings as detailed in Section 5.8.4. The proposed method, RPQRD, is executed with the same settings as those used for the PEVD and PSVD methods mentioned above. The normalization free variant of the rank decomposition method, explained in Chapter 5, is executed with  $\epsilon_{\zeta_q} = \mu_{\text{PU}} = 10^{-6}$  and  $K_{\text{max}} = 2^{\lceil \log_2[\mathcal{O}(\underline{\mathbf{R}}(z))] \rceil + 5}$ .

Ensemble results for all three metrics are shown in Fig. 6.3(a)-(c). The proposed approach triangularizes the given polynomial matrix with compact order paraunitary matrix in significantly reduced amount of time. The computational complexity of the proposed algorithm is evident from the results shown which can further be improved as the proposed algorithm is highly parallelizable as each column of  $\underline{\mathbf{Q}}(z)$  can be estimated independently.

## 6.8 Summary

This chapter has introduced a novel approach for computing the factorization of polynomial matrices. Recognizing the computational expense of the phase smoothing procedure required to estimate a compact-order factorization, this approach is based on the principle of estimating the auto- and cross-correlation sequences of the decomposition elements, thereby eliminating the need for phase smoothing. Subsequently, spectral factorization is applied to recover the compact-order component, which is essential for determining the necessary phase adjustments for the other remaining components. In

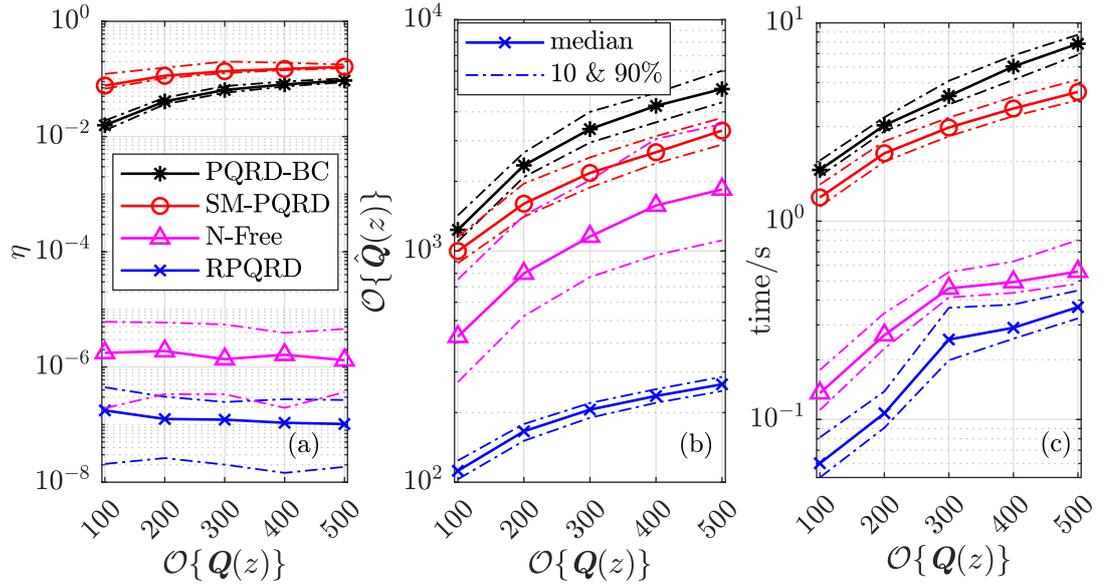


Figure 6.3: Performance metrics computed over an ensemble of randomized polynomial matrices for PQRD-BC [12], SM-PQRD [19], normalization free variant of rank decomposition method (N-Free) and roots based PQRD (RPEVD) with (a)  $\eta$ , (b)  $O\{\hat{Q}(z)\}$  and (c) execution time in seconds.

this context, two spectral factorization methods have been analyzed.

The cepstrum-based cepstral method [96, 100] is an iterative technique designed for positive definite para-Hermitian Laurent polynomials. However, the auto-correlation function of the components of an eigenvector or singular vector may not be positive definite, which can pose challenges. This method involves taking the logarithm in the DFT domain to transform the product of spectral factors, which can then be retrieved through exponentiation. One of the limitations of the cepstral method is that, due to the logarithm operation, it may not converge for any DFT size if the auto-correlation function has zeros on the unit circle. Due to this issue, cepstral method is not preferred as the condition of positive definiteness may not be satisfied. Moreover, this method also blindly spectrally factorizes an auto-correlation, usually resulting in minimum phase spectral factor, which may not result in compact polynomial order for the remaining component of an eigenvectors.

While spectral factorization is being performed, it should be considered that the spectral factor  $r_{nn,m}(z)$  must share all of its zero with  $r_{n\ell,m}(z)$  where  $n \neq \ell$ . As other-

wise, the polynomial order of the estimated eigenvector/singular vector or the columns of  $\underline{Q}(z)$  in the PQRD will not be compact, To address this issue of specific combinations of zeros of the auto-correlation function for constructing the spectral factor, the roots-based spectral factorization method is employed. This approach involves estimating both an auto-correlation function and a cross-correlation function of any component of an eigenvector or singular vector, and then determining their roots. The roots that are common to both functions are used to construct the spectral factors, resulting in the lowest possible order. However, due to estimation errors, the roots can be perturbed. In this case, instead of directly comparing the roots, the specific combination of roots is determined based on function evaluations. This allows for a more robust and accurate determination of the spectral factors, even in the presence of perturbations in the root estimates.

The proposed roots-based approach has been compared with state-of-the-art algorithms such as SBR2 and SMD for PEVD, GSR2 and 2-SMDs for PSVD, and PQRD-BC and SM-PQRD for PQRD. It has consistently outperformed these algorithms in terms of the order of decomposition, reconstruction error, and execution time, achieving orders of magnitude improvement. One of the key advantages of the proposed method is its high parallelizability. Each eigenvector, singular vector, or column of  $\underline{Q}(z)$  in PQRD can be estimated independently, making it a suitable candidate for platforms like FPGA where real-time broadband applications such as angle-of-arrival or speech enhancement are relevant. Additionally, this approach can be applied to various types of polynomial matrix decomposition, extending linear transformations like Givens rotations, Jacobians, Householder transformations, and Gram-Schmidt orthogonalization to the polynomial domain. This versatility makes it a valuable tool in various applications.

The spectral factorization method is generally preferred over the rank decomposition approach and its normalization-free variant in applications that necessitate lower-order polynomial factor decomposition or increased accuracy. The normalization-free variant is simple to implement in hardware, as it only requires basic implementation blocks such as FFT/IFFT and standard matrix factorization. As a result, it is recommended

for rapid and straightforward hardware analysis, the normalization-free variant is a suitable choice.

## Chapter 7

# Conclusion & Future Work

### 7.1 Thesis Summary

With the increasing interest in using polynomial matrices to model and represent broadband sensor array problems, polynomial matrix algebra has found numerous applications. As a result, issues related to computational complexity, accuracy, and polynomial order of decomposition have gained significant attention. Based on previous research compiled in [105–107], in this thesis, the three most commonly used polynomial matrix decomposition techniques PEVD, PSVD, and PQRD have been investigated. Here is a summary of the major findings:

#### 7.1.1 Space-Time Covariance Estimation and Loss of Multiplicities

Chapter 3 introduced the concept of system identification for improved space-time covariance estimation from sensor measurements, assuming known and controllable sources. This approach enhanced the accuracy of the estimation process with reduced sensor data requirements. Despite the ability to improve estimates at various SNRs through system identification, it was shown that the loss of algebraic multiplicities in the estimated space-time covariance still occurs with probability one. This insight allowed for more flexibility in defining new analytic PEVD and PSVD algorithms that rely on spectrally majorised eigen- and singular values.

### 7.1.2 Polynomial Power Method

Chapter 4 extended the power iteration method to the polynomial domain for estimating the dominant eigenvector of a para-Hermitian matrix. This method assumes that the estimated para-Hermitian matrix already has the spectral majorization property due to the loss of multiplicities during the estimation process. The approach was further combined with deflation to compute the PEVD of a low or reduced-rank para-Hermitian matrix. Lastly, the polynomial power method was generalized to estimate the dominant right and left singular vectors of a PSVD, along with the corresponding dominant singular value.

### 7.1.3 Rank One Decomposition

In Chapter 5, rank one decomposition-based PEVD, PSVD, and PQRD algorithms were introduced. This method represents any polynomial matrix as a sum of rank one terms and then applies a single iteration of the polynomial power method to determine the respective eigenvector and the corresponding eigenvalue or singular vector and the corresponding singular value. To address the issue of initializing the vector required for power iteration and polynomial power iteration, a normalization-free approach was proposed, which considers the columns of rank one terms as the initial vector. This proposed approach is fully parallelizable and provides accurate decomposition in the least amount of time compared to all available polynomial matrix factorization methods in literature except the spectral factorization method proposed in this thesis.

While a single iteration of the polynomial power method is not applicable for a PQRD computation, the columns of rank one terms are directly normalized to unit norm on the unit circle to estimate the columns of the paraunitary matrix for accomplishing the PQRD of a given polynomial matrix. However, the normalization process can encounter division by zero, which leads to a phase discontinuity and potentially requires a representation of infinite order to estimate the normalized vector. This issue was addressed by up-sampling the corresponding function to make the phase continuous across zero crossings.

### 7.1.4 Spectral Factorization for Polynomial Matrix Decomposition

Chapter 6 introduced a novel approach to compute spectral factors of the auto-correlation function of components of eigenvectors, singular vectors, or columns of  $\underline{Q}(z)$  for PEVD, PSVD, and PQRD, respectively. This approach eliminated the need for complex phase smoothing procedures. The chapter reviewed two spectral factorization methods, with the polynomial roots-based method found to perform better in terms of execution time and the order of the resulting paraunitary matrix. The proposed unified algorithm was compared to state-of-the-art algorithms in all three categories, demonstrating its significant accuracy, speed, and ability to produce compact order paraunitary matrices. Additionally, the method estimated each column of the paraunitary matrix independently, making it potentially suitable for fully parallel implementation, unlike current state-of-the-art algorithm, in real-time applications.

## 7.2 Applications-Oriented Comparison

A comparison chart has been provided in Table 7.1 between the spectral factorization-based method, commonly referred to as the root method, and the rank decomposition method, including its normalization-free variant, known as N-Free. This chart can aid in making a quick decision regarding the most suitable method to use based on the specific requirements of the application.

Table 7.1: Application-Oriented Comparison of the Proposed Approaches

<b>Applications</b>	<b>Rank Decomposition</b>	<b>N-Free</b>	<b>Root Method</b>
Accuracy	Low	Low	High
Order	High	High	Low
Execution Time	High	Low	Low
Ease of Implementation	High	High	Low

## 7.3 Future Work

Based on the research presented in this thesis, possible future work that will further improve or broaden its scope can include:

- The proof of convergence for polynomial power method is yet to be formally formulated which could be beneficial, especially for improving the initialization vector to expedite its convergence.
- extending the generalized polynomial power method in combination with deflation for PSVD may necessitate a distinct approach compared to the PEVD case. Conducting an analysis of perturbations and error bounds within this context could provide valuable insights for theoretical advancements.
- it is highly likely that with the proposed unified algorithm-II, all feasible linear unitary transformations can be seamlessly extended to the polynomial domain. Especially the Jacobian transformation or Givens rotation extension to polynomial domain would offer a significantly different perspective of the SBR2 algorithm considering its reliance on the standard Jacobian transformation in each iteration. Furthermore, the Householder transformation based SBR2 variant would also benefit from this polynomial extension, potentially reducing the number of required transformations. These possibilities for polynomial extensions could also enhance the iterative PQRD algorithms, enabling them to operate on an entire element of a polynomial matrix instead of being restricted to a single slice.
- With the proof of existence and the underlying condition for an analytic SVD of a polynomial matrix, there is a possibility of an un-majorised para-Hermitian polynomial estimated through system identification. For instance if  $\mathbf{A}(z)$ , with negative singular values on the unit circle, is estimated via system identification and then a para-Hermitian matrix  $\hat{\mathbf{A}}^P(z)\hat{\mathbf{A}}(z)$  or  $\hat{\mathbf{A}}(z)\hat{\mathbf{A}}^P(z)$  is constructed, there is a possibility that this type of estimated CSD matrix for given MIMO system might be spectrally un-majorised.
- Polynomial Gram-Schmidt orthogonalization can be utilized in different applications, such as for the block matrix construction of a polynomial generalized sidelobe canceller in the area of adaptive broadband beamforming.

- Except for the normalization-free approach, which has demonstrated superior performance in speech enhancement, surpassing SBR2 and SMD, all other proposed methods in the thesis require evaluation using real-world data to compare their performance more effectively against state-of-the-art algorithms.
- Applications for the various techniques proposed in this thesis could be explored. This includes for example the utilisation of the PSVD for the best paraunitary approximation of a general polynomial matrix, i.e. a polynomial extension of the Procrustes problem [1, 108], in order to find lossless transfer function matrices in the context of acoustics and artificial reverberation of multichannel audio [109, 110].

# Bibliography

- [1] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, 1996.
- [2] S. Haykin, *Adaptive filter theory*, 4th ed. Upper Saddle River, NJ: Prentice Hall, 2002.
- [3] A. Paulraj, R. Nabar, and D. Gore, *Introduction to space-time wireless communications*. Cambridge university press, 2003.
- [4] T. Kato, *Perturbation Theory for Linear Operators*, 2nd ed. Berlin: Springer Verlag, 1995.
- [5] I. Jolliffe, *Principal Component Analysis*. John Wiley & Sons, Ltd, 2014.
- [6] D. Kundu, “Modified MUSIC algorithm for estimating doa of signals,” *Signal processing*, vol. 48, no. 1, pp. 85–90, 1996.
- [7] R. Schmidt, “Multiple emitter location and signal parameter estimation,” *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, Mar. 1986.
- [8] S. Weiss, S. Bendoukha, A. Alzin, F. K. Coutts, I. K. Proudler, and J. Chambers, “MVDR broadband beamforming using polynomial matrix techniques,” in *23rd European Signal Processing Conference (EUSIPCO)*, 2015, pp. 839–843.
- [9] V. W. Neo, S. Redif, J. G. McWhirter, J. Pestana, I. K. Proudler, S. Weiss, and P. A. Naylor, “Polynomial eigenvalue decomposition for multichannel broadband

- signal processing,” *IEEE Signal Processing Magazine*, vol. 40, no. 7, pp. 18–37, 2023.
- [10] S. Weiss, I. K. Proudler, G. Barbarino, J. Pestana, and J. G. McWhirter, “On properties and structure of the analytic singular value decomposition,” submitted.
- [11] J. G. McWhirter, “An algorithm for polynomial matrix SVD based on generalised kogbetliantz transformations,” in *2010 18th European Signal Processing Conference*, 2010, pp. 457–461.
- [12] J. A. Foster, J. G. McWhirter, M. R. Davies, and J. A. Chambers, “An algorithm for calculating the QR and singular value decompositions of polynomial matrices,” *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1263–1274, 2010.
- [13] J. G. McWhirter, P. D. Baxter, T. Cooper, S. Redif, and J. Foster, “An EVD algorithm for para-hermitian polynomial matrices,” *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 2158–2169, 2007.
- [14] M. A. Alrmah, S. Weiss, and S. Lambotharan, “An extension of the MUSIC algorithm to broadband scenarios using a polynomial eigenvalue decomposition,” in *2011 19th European Signal Processing Conference*, 2011, pp. 629–633.
- [15] W. Coventry, C. Clemente, and J. Soraghan, “Enhancing polynomial music algorithm for coherent broadband sources through spatial smoothing,” in *25th European Signal Processing Conference (EUSIPCO)*, 2017, pp. 2448–2452.
- [16] —, “Broadband direction of arrival estimation via spatial co-prime sampling and polynomial matrix methods,” *The Journal of Engineering*, vol. 2019, no. 19, pp. 6259–6263, 2019.
- [17] V. W. Neo, C. Evers, and P. A. Naylor, “PEVD-based speech enhancement in reverberant environments,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 186–190.
- [18] —, “Polynomial matrix eigenvalue decomposition of spherical harmonics for

- speech enhancement,” in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 786–790.
- [19] D. Hassan, S. Redif, S. Lambbotharan, and I. K. Proudler, “Sequential Polynomial QR Decomposition and Decoding of Frequency Selective MIMO Channels,” in *2018 26th European Signal Processing Conference (EUSIPCO)*, Sep. 2018, pp. 460–464, iSSN: 2076-1465.
- [20] S. Redif, J. G. McWhirter, and S. Weiss, “Design of FIR paraunitary filter banks for subband coding using a polynomial eigenvalue decomposition,” *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5253–5264, 2011.
- [21] S. Weiss, N. Goddard, S. Somasundaram, I. Proudler, and P. Naylor, “Identification of broadband source-array responses from sensor second order statistics,” in *2017 Sensor Signal Processing for Defence Conference (SSPD)*, 2017, pp. 1–5.
- [22] S. Redif, S. Weiss, and J. G. McWhirter, “Sequential matrix diagonalization algorithms for polynomial EVD of parahermitian matrices,” *IEEE Transactions on Signal Processing*, vol. 63, no. 1, pp. 81–89, 2015.
- [23] J. Foster, J. McWhirter, and J. Chambers, “A Polynomial Matrix QR Decomposition with Application to MIMO Channel Equalisation,” in *2007 Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers*, Nov. 2007, pp. 1379–1383, iSSN: 1058-6393.
- [24] J. Corr, K. Thompson, S. Weiss, J. G. McWhirter, S. Redif, and I. K. Proudler, “Multiple shift maximum element sequential matrix diagonalisation for parahermitian matrices,” in *2014 IEEE Workshop on Statistical Signal Processing (SSP)*, 2014, pp. 312–315.
- [25] Z. Wang, J. G. McWhirter, J. Corr, and S. Weiss, “Multiple shift second order sequential best rotation algorithm for polynomial matrix EVD,” in *2015 23rd European Signal Processing Conference (EUSIPCO)*, 2015, pp. 844–848.
- [26] Z. Wang, A. Sandmann, J. G. McWhirter, and A. Ahrens, “Multiple shift SBR2 algorithm for calculating the SVD of broadband optical MIMO systems,” in

*2016 39th International Conference on Telecommunications and Signal Processing (TSP)*, 2016, pp. 433–436.

- [27] J. Corr, K. Thompson, S. Weiss, I. K. Proudler, and J. G. McWhirter, “Row-shift corrected truncation of paraunitary matrices for PEVD algorithms,” in *2015 23rd European Signal Processing Conference (EUSIPCO)*, 2015, pp. 849–853.
- [28] J. Corr, K. Thompson, S. Weiss, I. Proudler, and J. McWhirter, “Shortening of paraunitary matrices obtained by polynomial eigenvalue decomposition algorithms,” in *2015 Sensor Signal Processing for Defence (SSPD)*, 2015, pp. 1–5.
- [29] J. Foster, J. G. McWhirter, and 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.
- [30] C. H. Ta and S. Weiss, “Shortening the order of paraunitary matrices in SBR2 algorithm,” in *6th International Conference on Information, Communications & Signal Processing*, 2007, pp. 1–5.
- [31] F. K. Coutts, I. K. Proudler, and S. Weiss, “Efficient implementation of iterative polynomial matrix EVD algorithms exploiting structural redundancy and parallelisation,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 12, pp. 4753–4766, 2019.
- [32] F. Coutts, J. Corr, K. Thompson, I. Proudler, and S. Weiss, “Divide-and-conquer sequential matrix diagonalisation for parahermitian matrices,” in *2017 Sensor Signal Processing for Defence Conference (SSPD)*, 2017, pp. 1–5.
- [33] M. Tohidian, H. Amindavar, and A. M. Reza, “A DFT-based approximate eigenvalue and singular value decomposition of polynomial matrices,” *EURASIP Journal on Advances in Signal Processing*, vol. 2013, no. 1, p. 93, 2013.
- [34] S. Weiss, I. K. Proudler, F. K. Coutts, and F. A. Khattak, “Eigenvalue decomposition of a parahermitian matrix: Extraction of analytic eigenvectors,” *IEEE Transactions on Signal Processing*, vol. 71, pp. 1642–1656, 2023.

- [35] S. Weiss, I. K. Proudler, and F. K. Coutts, “Eigenvalue decomposition of a Parahermitian matrix: Extraction of analytic eigenvalues,” *IEEE Transactions on Signal Processing*, vol. 69, pp. 722–737, 2021.
- [36] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. USA: Prentice-Hall, Inc., 1993.
- [37] F. A. Khattak, S. Weiss, I. K. Proudler, and J. G. McWhirter, “Space-Time Covariance Matrix Estimation: Loss of Algebraic Multiplicities of Eigenvalues,” in *2022 56th Asilomar Conference on Signals, Systems, and Computers*, Oct. 2022, pp. 975–979, iSSN: 2576-2303.
- [38] F. A. Khattak, I. K. Proudler, and S. Weiss, “Enhanced Space-Time Covariance Estimation Based on a System Identification Approach,” in *2022 Sensor Signal Processing for Defence Conference (SSPD)*, Sep. 2022, pp. 1–5.
- [39] —, “Extension of power method to para-Hermitian matrices: Polynomial power method,” in *2023 31st European Signal Processing Conference (EUSIPCO)*, 2023, pp. 1564–1568.
- [40] —, “Generalized polynomial power method,” in *Sensor Signal Processing for Defence Conference (SSPD)*, 2023, pp. 1–5.
- [41] —, “Low rank para-Hermitian matrix EVD via polynomial power method with deflation,” in *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, accepted.
- [42] G. Raleigh and J. Cioffi, “Spatio-temporal coding for wireless communication,” *IEEE Transactions on Communications*, vol. 46, no. 3, pp. 357–366, 1998.
- [43] J. Foster, J. McWhirter, S. Lambotharan, I. Proudler, M. Davies, and J. Chambers, “Polynomial matrix QR decomposition for the decoding of frequency selective multiple-input multiple-output communication channels,” *IET signal processing*, vol. 6, no. 7, pp. 704–712, 2012.

- [44] S. Weiss, J. Pestana, and I. K. Proudler, “On the existence and uniqueness of the eigenvalue decomposition of a parahermitian matrix,” *IEEE Transactions on Signal Processing*, vol. 66, no. 10, pp. 2659–2672, 2018.
- [45] S. Weiss, J. Pestana, I. K. Proudler, and F. K. Coutts, “Corrections to on the existence and uniqueness of the eigenvalue decomposition of a parahermitian matrix,” *IEEE Transactions on Signal Processing*, vol. 66, no. 23, pp. 6325–6327, 2018.
- [46] F. Khattak, S. Weiss, and I. K. Proudler, “Fast Givens rotation approach to second order sequential best rotation algorithms,” in *Sensor Signal Processing for Defence Conference (SSPD)*, 2021, pp. 1–5.
- [47] S. Weiss, I. K. Proudler, F. K. Coutts, and J. Deeks, “Extraction of Analytic Eigenvectors From a Parahermitian Matrix,” in *2020 Sensor Signal Processing for Defence Conference (SSPD)*, Sep. 2020, pp. 1–5.
- [48] P. Vaidyanathan, “Theory of optimal orthonormal subband coders,” *IEEE Transactions on Signal Processing*, vol. 46, no. 6, pp. 1528–1543, 1998.
- [49] Z. Wang, J. G. McWhirter, J. Corr, and S. Weiss, “Order-controlled multiple shift sbr2 algorithm for para-hermitian polynomial matrices,” in *2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, 2016, pp. 1–5.
- [50] J. McWhirter and Z. Wang, “A novel insight to the sbr2 algorithm for diagonalising para-hermitian matrices,” in *IMA Conference on Mathematics in Signal Processing, 2016*. Institute of Mathematics, 2016.
- [51] W. Rath, “Fast Givens rotations for orthogonal similarity transformations,” *Numerische Mathematik*, vol. 40, pp. 47–56, 1982.
- [52] J. Kowalik and S. Kumar, “Fast Givens transformations applied to the homogeneous optimization method,” *Applied Mathematics and Computation*, vol. 4, no. 3, pp. 239–252, 1978.

- [53] J. Corr, K. Thompson, S. Weiss, I. K. Proudler, and J. G. McWhirter, “Reduced search space multiple shift maximum element sequential matrix diagonalisation algorithm,” in *2nd IET International Conference on Intelligent Signal Processing 2015 (ISP)*, 2015, pp. 1–5.
- [54] V. W. Neo and P. A. Naylor, “Second order sequential best rotation algorithm with householder reduction for polynomial matrix eigenvalue decomposition,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 8043–8047.
- [55] M. POWELL, “A new algorithm for unconstrained optimization,” in *Nonlinear Programming*, J. Rosen, O. Mangasarian, and K. Ritter, Eds. Academic Press, 1970, pp. 31–65.
- [56] S. Weiss, I. K. Proudler, F. K. Coutts, and J. Pestana, “Iterative Approximation of Analytic Eigenvalues of a Parahermitian Matrix EVD,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 8038–8042, iSSN: 2379-190X.
- [57] B. De Moor and S. Boyd, “Analytic properties of singular values and vectors,” KU Leuven, Tech. Rep., 1989.
- [58] A. Bunse-Gerstner, R. Byers, V. Mehrmann, and N. K. Nichols, “Numerical computation of an analytic singular value decomposition of a matrix valued function,” *Numerische Mathematik*, vol. 60, no. 1, pp. 1–39, Dec. 1991.
- [59] M. Bakhit, F. A. Khattak, I. K. Proudler, S. Weiss, and G. Rice, “Compact order polynomial singular value decomposition of a matrix of analytic functions,” in *2023 IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, accepted, 2023.
- [60] G. E. Forsythe and P. Henrici, “The cyclic jacobi method for computing the principal values of a complex matrix,” *Transactions of the American Mathematical Society*, vol. 94, no. 1, pp. 1–23, 1960.

- [61] F. A. Khattak, I. K. Proudler, J. G. McWhirter, and S. Weiss, “Generalised sequential matrix diagonalisation for the SVD of polynomial matrices,” in *Sensor Signal Processing for Defence Conference (SSPD)*, 2023, pp. 1–5.
- [62] G. Barbarino and V. Noferini, “On the real eigendecomposition of para-Hermitian matrices and the sign characteristics of  $*$ -palindromic matrix polynomials,” *Linear Algebra and its Applications*, vol. 672, pp. 1–27, 2023.
- [63] D. Cescato and H. Bölcskei, “QR decomposition of laurent polynomial matrices sampled on the unit circle,” *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4754–4761, Sept 2010.
- [64] —, “Algorithms for interpolation-based QR decomposition in MIMO-OFDM systems,” *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1719–1733, April 2011.
- [65] M. Davies, S. Lambotharan, J. Foster, J. Chambers, and J. McWhirter, “Polynomial Matrix QR Decomposition and Iterative Decoding of Frequency Selective MIMO Channels,” in *2009 IEEE Wireless Communications and Networking Conference*, Apr. 2009, pp. 1–6, iSSN: 1558-2612.
- [66] F. A. Khattak, M. Bakhit, I. K. Proudler, and S. Weiss, “Smooth QR decomposition of polynomial matrices,” in *2023 IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, accepted, 2023.
- [67] S. Weiss, M. Alrmah, S. Lambotharan, J. G. McWhirter, and M. Kaveh, “Broadband angle of arrival estimation methods in a polynomial matrix decomposition framework,” in *2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Dec. 2013, pp. 109–112.
- [68] C. Delaosa, J. Pestana, N. J. Goddard, S. Somasundaram, and S. Weiss, “Sample space-time covariance matrix estimation,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 8033–8037.

- [69] G. W. Stewart and J. guang Sun, *Matrix Perturbation Theory*, 1990.
- [70] C. Delaosa, F. K. Coutts, J. Pestana, and S. Weiss, “Impact of space-time covariance estimation errors on a parahermitian matrix EVD,” in *IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, 2018, pp. 164–168.
- [71] V. W. Neo, C. Evers, and P. A. Naylor, “Enhancement of noisy reverberant speech using polynomial matrix eigenvalue decomposition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3255–3266, 2021.
- [72] S. Redif, “Fetal electrocardiogram estimation using polynomial eigenvalue decomposition,” *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 24, no. 4, pp. 2483–2497, 2016.
- [73] C. Delaosa, J. Pestana, S. Weiss, and I. K. Proudler, “Subspace perturbation bounds with an application to angle of arrival estimation using the MUSIC algorithm,” in *2020 Sensor Signal Processing for Defence Conference (SSPD)*, 2020, pp. 1–5.
- [74] C. Delaosa, J. Pestana, N. J. Goddard, S. D. Somasundaram, and S. Weiss, “Support estimation of a sample space-time covariance matrix,” in *Sensor Signal Processing for Defence Conference (SSPD)*, 2019, pp. 1–5.
- [75] F. K. Coutts, J. Corr, K. Thompson, S. Weiss, I. K. Proudler, and J. G. McWhirter, “Complexity and search space reduction in cyclic-by-row PEVD algorithms,” in *2016 50th Asilomar Conference on Signals, Systems and Computers*, 2016, pp. 1349–1353.
- [76] A. O. T. Hogg, V. W. Neo, S. Weiss, C. Evers, and P. A. Naylor, “A polynomial eigenvalue decomposition MUSIC approach for broadband sound source localization,” in *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2021, pp. 326–330.
- [77] A. V. Oppenheim, A. S. Willsky, S. H. Nawab, and J.-J. Ding, *Signals and systems*. Prentice hall Upper Saddle River, NJ, 1997, vol. 2.

- [78] X. Mestre, “On the asymptotic behavior of the sample estimates of eigenvalues and eigenvectors of covariance matrices,” *IEEE Transactions on Signal Processing*, vol. 56, no. 11, pp. 5353–5368, 2008.
- [79] E. Ollila, D. E. Tyler, V. Koivunen, and H. V. Poor, “Complex Elliptically Symmetric Distributions: Survey, New Results and Applications,” *IEEE Transactions on Signal Processing*, vol. 60, no. 11, pp. 5597–5625, Nov. 2012.
- [80] M. Greco, S. Fortunati, and F. Gini, “Maximum likelihood covariance matrix estimation for complex elliptically symmetric distributions under mismatched conditions,” *Signal Processing*, vol. 104, pp. 381–386, Nov. 2014.
- [81] S. Weiss, C. Delaosa, J. Matthews, I. K. Proudler, and B. A. Jackson, “Detection of Weak Transient Signals Using a Broadband Subspace Approach,” in *2021 Sensor Signal Processing for Defence Conference (SSPD)*, Sep. 2021, pp. 1–5.
- [82] V. W. Neo, S. Weiss, and P. A. Naylor, “A Polynomial Subspace Projection Approach for the Detection of Weak Voice Activity,” in *2022 Sensor Signal Processing for Defence Conference (SSPD)*, Sep. 2022, pp. 1–5.
- [83] V. W. Neo, S. Weiss, S. W. McKnight, A. O. T. Hogg, and P. A. Naylor, “Polynomial Eigenvalue Decomposition-Based Target Speaker Voice Activity Detection in the Presence of Competing Talkers,” in *2022 International Workshop on Acoustic Signal Enhancement (IWAENC)*, Sep. 2022, pp. 1–5.
- [84] S. Weiss, J. Selva, and M. D. Macleod, “Measuring smoothness of trigonometric interpolation through incomplete sample points,” in *2020 28th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 2319–2323.
- [85] G. Barbarino and V. Noferini, “On the rellich eigendecomposition of parahermitian matrices and the sign characteristics of \*-palindromic matrix polynomials,” 2022.
- [86] F. Khattak, I. K. Proudler, and S. Weiss, “Support estimation of analytic eigenvectors of parahermitian matrices,” in *2022 International Conference on Recent*

- Advances in Electrical Engineering & Computer Sciences (RAEE & CS)*, 2022, pp. 1–6.
- [87] L. V. Ahlfors, *Complex analysis: An introduction to the theory of analytic functions of one complex variable*. New York: McGraw-Hill, 1953.
- [88] L. Batzke, C. Mehl, A. C. M. Ran, and L. Rodman, “Generic rank- $k$  perturbations of structured matrices,” in *Operator Theory, Function Spaces, and Applications*, T. Eisner, B. Jacob, A. Ran, and H. Zwart, Eds. Cham: Springer International Publishing, 2016, pp. 27–48.
- [89] A. Ran and M. Wojtylak, “Global properties of eigenvalues of parametric rank one perturbations for unstructured and structured matrices,” *Complex Analysis and Operator Theory*, vol. 15, no. 44, pp. 1–27, 2021.
- [90] J. Wilkinson, *The Algebraic Eigenvalue Problem*. Clarendon Press, Oxford, 1965.
- [91] J. Foster, J. G. McWhirter, and 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.
- [92] A. H. Sayed and T. Kailath, “A survey of spectral factorization methods,” *Numerical linear algebra with applications*, vol. 8, no. 6-7, pp. 467–496, 2001.
- [93] T. N. Goodman, C. A. Micchelli, G. Rodriguez, and S. Seatzu, “Spectral factorization of laurent polynomials,” *Advances in Computational Mathematics*, vol. 7, pp. 429–454, 1997.
- [94] G. Wilson, “Factorization of the covariance generating function of a pure moving average process,” *SIAM Journal on Numerical Analysis*, vol. 6, no. 1, pp. 1–7, 1969.
- [95] D. Youla and N. Kazanjian, “Bauer-type factorization of positive matrices and the theory of matrix polynomials orthogonal on the unit circle,” *IEEE Transactions on Circuits and Systems*, vol. 25, no. 2, pp. 57–69, 1978.

- [96] B. P. Bogert, "The quefreny alanalysis of time series for echoes: Cepstrum, pseudoautocovariance, cross-cepstrum and saphe cracking," in *Proc. Symposium Time Series Analysis, 1963*, 1963, pp. 209–243.
- [97] M. Lang and B.-C. Frenzel, "Polynomial root finding," *IEEE Signal Processing Letters*, vol. 1, no. 10, pp. 141–143, 1994.
- [98] C. Schmidt and L. Rabiner, "A study of techniques for finding the zeros of linear phase fir digital filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 1, pp. 96–98, 1977.
- [99] "Efficient polynomial root-refiners: A survey and new record efficiency estimates," *Computers & Mathematics with Applications*, vol. 63, no. 1, pp. 239–254, 2012.
- [100] B. P. Bogert, "The quefreny alanalysis of time series for echoes: Cepstrum, pseudoautocovariance, cross-cepstrum and saphe cracking," in *Proc. Symposium Time Series Analysis, 1963*, 1963, pp. 209–243.
- [101] L. R. Rabiner, R. W. Schafer *et al.*, "Introduction to digital speech processing," *Foundations and Trends® in Signal Processing*, vol. 1, no. 1–2, pp. 1–194, 2007.
- [102] J. Ježek, M. Hromčík, and M. Šebek, "New algorithm for spectral factorization and its practical application," in *2001 European Control Conference (ECC)*, 2001, pp. 3104–3109.
- [103] M. Hromčík and M. Šebek, "Numerical algorithms for polynomial plus/minus factorization," *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 17, no. 8, pp. 786–802, 2007.
- [104] V. Belevitch, "Review of 'ein direktes iterations verfahren zur Hurwitz-Zerlegung eines polynoms (a direct iterative process for the hurwitz-decomposition of a polynomial)'," *IRE Transactions on Circuit Theory*, vol. 2, no. 4, pp. 370–371, 1955.

- [105] F. K. Coutts, “Algorithmic enhancements to polynomial matrix factorisations,” PhD thesis, University of Strathclyde, Glasgow, Scotland, May 2019.
- [106] J. Corr, “Advanced algorithms for polynomial matrix eigenvalue decomposition,” PhD thesis, University of Strathclyde, Glasgow, Scotland, November 2017.
- [107] J. A. Foster, “Algorithms and techniques for polynomial matrix decompositions,” PhD thesis, Cardiff University, Cardiff, England, May 2008.
- [108] S. Weiss, S. J. Schlecht, O. Das, and E. De Sena, “Polynomial Procrustes problem: Paraunitary approximation of matrices of analytic functions,” in *EUSIPCO*, Helsinki, Finland, 2023.
- [109] S. J. Schlecht and E. A. P. Habets, “On lossless feedback delay networks,” *IEEE Transaction on Signal Processing*, vol. 65, no. 6, pp. 1554–1564, March 2017.
- [110] S. J. Schlecht, “Allpass feedback delay networks,” *IEEE Trans. Signal Process.*, vol. 69, pp. 1028–1038, 2021.