UNIVERSITY OF SOUTHAMPTON

FACULTY OF ENGINEERING, SCIENCE & MATHEMATICS

School of Electronics and Computer Science

Polynomial Matrix Decompositions and Paraunitary Filter Banks

by

Soydan Redif

A thesis submitted in partial fulfilment of the requirements

for the degree of

Doctor of Philosophy

5 July 2006

Supervisors:

Dr. Stephan Weiss

(University of Southampton)

Professor John G. McWhirter FRS, FREng

(QinetiQ)

To my dear wife,

Gülnaz

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE & MATHEMATICS
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

Polynomial Matrix Decompositions and Paraunitary Filter Banks

by Soydan Redif

There are an increasing number of problems that can be solved using paraunitary filter banks. The design of optimal orthonormal filter banks for the efficient coding of signals has received considerable interest over the years. In contrast, very little attention has been given to the problem of constructing paraunitary matrices for the purpose of broadband signal subspace estimation. This thesis begins by relating these two areas of research. A frequency-domain method of diagonalising parahermitian polynomial matrices is proposed and shown to have fundamental limitations. Then the thesis focuses on the development of a novel time-domain technique that extends the eigenvalue decomposition to polynomial matrices, referred to as the second order sequential best rotation (SBR2) algorithm. This technique imposes strong decorrelation on its input signals by applying a sequence of elementary paraunitary matrices which constitutes a generalisation of the classical Jacobi algorithm to the field of polynomial matrices. It is shown to be highly applicable to the problems of broadband signal subspace estimation and data compression. Variations on the algorithm are presented which give a significant improvement in subspace estimation accuracy and data compression performance.

Discussions are then mainly concerned with the application of the SBR2 algorithm to the problem of data compression, particularly the adaptation of the SBR2 algorithm to subband coding. The relevance of the algorithm to traditional orthonormal filter bank design methods is examined, highlighting that these techniques are based on an implicit assumption regarding the statistics of the input signal. This provides motivation for the development of a method of exploiting this knowledge for use with the SBR2 algorithm. The resulting algorithm can design orthonormal filter banks for subband coding. The suboptimality, in the sense of maximising the coding gain, of the filter bank constructed becomes negligible as the number of algorithm iterations increases. The technique is shown to compare favourably to the state-of-the-art on a set of benchmark problems.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to sincerely thank my supervisor Dr. Stephan Weiss for his exceptional guidance, excellent support and the encouragement he has provided me. Your attention to detail, insightful suggestions and excellent understanding of signal processing has been tremendously helpful in preparing this thesis. Thank you for the inspirational discussions and research collaborations. I am thankful to Dr. Paul White of ISVR for his thorough reviews and suggestions during the writing of this thesis. Thanks also to Prof. Bernard Mulgrew for his insightful comments.

I wish to express my profound gratitude to my external supervisor Prof. John G. McWhirter of QinetiQ for all the invaluable support and direction he has given me. J.G. is a fascinating mentor, immensely inspirational and a friend. He is also a thorough, brilliant researcher and I have benefited greatly from working under his guidance. I have enjoyed the many long and time-warped discussions we have had, which often paved the way to clarity of the topic and new understanding. Thank you.

I am greatly thankful to Dr. Tom Cooper of QinetiQ for his assistance in producing this work. I have benefited considerably from his exceptional knowledge of mathematics and his style of thinking. You have managed to bring more clarity and soundness to ideas we have discussed. Through our many long and diverse discussions you have provided me with valuable advice, support and encouragement, and have been a thoughtful supervisor and a good friend.

Thanks to both Prof. Ian Proudler and Dr. Edward Warner of QinetiQ for the valuable advice and support they have given me over the years. Dr. Warner has been a thoughtful mentor, always ready to share his expertise with me and provide invaluable nuggets of knowledge. I am grateful to Dr. Proudler for believing in me and providing me with a strong platform from which to begin my doctoral studies. He is also in the elite of exceptional researchers whom I am very fortunate to be working with. Thank you both for your support and friendship. Many thanks also to colleagues in CSIP for the meaningful discussions we had and to QinetiQ's Corporate Research Programme for providing the financial backing for this work.

My greatest blessing is my family for which I am very thankful. I am indebted to my parents for their unconditional love, unending support and for providing me with the opportunity to pursue my ambitions. I would like to thank my two sisters and my only brother for their love and encouragement. Lastly, but by no means least, I am grateful indeed to my delightful daughter, Meliz, and my loving wife, Gülnaz, for their patience and immeasurable support while I wrote this thesis and for their understanding through trying times.

Acronyms

ARMA	autoregressive moving-average
BPSK	binary phase-shift keying
BSS	blind signal separation
DFT	discrete Fourier transform
DSP	digital signal processing
EVD	eigenvalue decomposition
FIR	finite impulse response
IDFT	inverse discrete Fourier transform
IFB	independent frequency band
i.i.d.	independent and identically distributed
IIR	infinite impulse response
ISI	intersymbol interference
KLT	Karhunen-Loeve transform
LTI	linear time-invariant
MIMO	multi-input multi-output
QPSK	quaternary phase-shift keying
PCA	principal component analysis
PCFB	principal component filter bank
PCM	pulse code modulation
PEVD	polynomial matrix eigenvalue decomposition
PRFB	perfect reconstruction filter bank
PSD	power spectral density
PSVD	polynomial matrix singular value decomposition
SNR	signal-to-noise ratio
SVD	singular value decomposition
WSS	wide-sense stationary

Principal Symbols

General Notation

a	scalar quantity
$\mathbf{a}$	vector quantity
$\mathbf{A}$	scalar matrix
a_{km}	(k, m) element of a scalar matrix
$a(t)$	function of a discrete variable t
a_t	alternative form of $a(t)$
$A(e^{j\omega})$	discrete Fourier spectrum (PSD) of a discrete function $a(t)$
$A(z)$	z-transform of a discrete function $a(t)$: a polynomial or Laurent polynomial in the indeterminate variable z^{-1} or an algebraic power series
$\underline{A}(z)$	vector of polynomials or algebraic power series
$A_k(z)$	k^{th} row entry in a polynomial vector or vector of algebraic power series
$\mathbf{a}(t)$	space-time matrix
$\mathbf{A}(z)$	matrix of polynomials (z-transform of $\mathbf{a}(t)$)
$a_{km}(t)$	(k, m) entry of $\mathbf{a}(t)$
$A_{km}(z)$	(k, m) entry of $\mathbf{A}(z)$
$\mathbf{A}(e^{j\omega})$	PSD matrix (i.e. $\mathbf{A}(z)\big\|_{z=e^{j\omega}}$)
$A_{km}(e^{j\omega})$	(k, m) entry of a PSD matrix (PSD of $a_{km}(t)$)

Relations and Operators

$(\cdot)^*$	complex conjugate
$(\cdot)^T$	matrix transpose
$(\cdot)^H$	Hermitian transpose (conjugate transpose)
$\tilde{\mathbf{A}}(z)$	paraconjugate transpose $\tilde{\mathbf{A}}(z) = \mathbf{A}^H(1/z^*)$

$*$	convolution operator
$\lfloor \cdot \rfloor$	floor operator (round off)
$\mathrm{E}[\cdot]$	statistical expectation operator
$\det[\cdot]$	determinant of a matrix operator
$\mathrm{diag}[\cdot]$	operator that takes a vector to a diagonal matrix
$\mathrm{trace}[\cdot]$	operator that computes the sum of the diagonal elements of a matrix

Sets and Spaces

$\mathbb{C}$	set of complex numbers
$\mathbb{C}^{M\times N}$	set of $M \times N$ matrices with complex scalar elements
$\mathbb{C}^{M\times N}(z)$	set of $M \times N$ matrices with complex polynomial entries in the indeterminate variable z^{-1}
$\mathbb{N}$	set of natural numbers $\{\lvert a\rvert : a \in \mathbb{Z}\} = \mathbb{N}$
$\mathbb{R}$	set of real numbers
$\mathbb{R}^{M\times N}$	set of $M \times N$ matrices with real scalar elements
$\mathbb{R}^{M\times N}(z)$	set of $M \times N$ matrices with real polynomial entries in the indeterminate variable z^{-1}
$\mathbb{Z}$	set of integer numbers $\{a : \lvert a\rvert \in \mathbb{N}\} = \mathbb{Z}$

Principal Symbols

γ	measure of the accuracy of broadband subspace decomposition
$\delta(t)$	Kronecker delta function, $\delta(0) = 1$ and $\delta(t) = 0$, for $t \neq 0$
ε	filter bank reconstruction error
t	discrete time index at a high sample rate for subband processing
λ	measure of the extent of polynomial matrix diagonalisation
θ, φ	angles parameterising a Jacobi rotation matrix
τ	correlation lag
ϕ	frequency index

ω	normalised angular frequency
$\mathbf{\Lambda}(z)$	paraunitary delay matrix: 2×2 identity matrix with a pure unit time-delay on the lower diagonal element
$\mathbf{\Lambda}^{(k,\tau)}(z)$	arbitrary (paraunitary) delay matrix: $M\times M$ identity matrix with a pure time-delay τ on the k^{th} diagonal element
$\mathbf{a}(\tau)$	true space-time covariance matrix for the input vector signal $\mathbf{x}(t)$
$\mathbf{A}(z)$	true polynomial covariance matrix for the input polynomial vector $\underline{X}(z)$
$\mathbf{A}(e^{j\omega})$	true PSD matrix for the input polynomial vector $\underline{X}(z)$
a_{kk0}	true variance of the k^{th} input signal to a filter bank
$\mathbf{c}(\tau)$	true space-time covariance matrix for the vector of output sequences $\mathbf{y}(t)$
$\mathbf{C}(z)$	true polynomial covariance matrix for the output polynomial vector $\underline{Y}(z)$
$\mathbf{C}(e^{j\omega})$	true PSD matrix for the output polynomial vector $\underline{Y}(z)$
c	cosine of an angle θ
c_{kk0}	true variance of the k^{th} output signal from a filter bank
$\mathbf{E}(z)$	polyphase matrix of the analysis filter bank
$\mathbf{F}(z)$	polyphase matrix of the synthesis filter bank
$\mathbf{G}(z)$	polynomial (FIR) mixing matrix
G	coding gain measure
$\hat{G}$	ideal coding gain
G_{comp}	compaction gain measure
$\hat{G}_{comp}$	ideal compaction gain
G_c	measure based on the coding gain for arbitrary multiple channels
$G_k(z)$	k^{th} filter transfer function of the synthesis filter bank
G_m	compression gain measure
$\mathbf{H}(z)$	paraunitary polynomial matrix produced by a PEVD

$H_k(z)$	k^{th} filter transfer function of the analysis filter bank
$H_k(e^{j\omega})$	k^{th} frequency response of the analysis filter bank
$\mathbf{I}_K$	identity matrix of dimension $K \times K$
K	number of input channels to a $M \times K$ mixing polynomial matrix
L	number of iterations of the classical Jacobi algorithm or the SBR2 algorithm
M	number of output channels from a $M \times K$ mixing polynomial matrix
	or number of subband channels (i.e. decimation factor)
N_1	measure of 2^{nd} order statistical independence
N_2	squared norm of the zero-lag covariance matrix in $\mathbf{r}(\tau)$
N_3	measure of 2^{nd} order statistical dependence
N_4	squared norm of the entire space-time covariance matrix $\mathbf{r}(\tau)$
N	order of a FIR filter
N_a	order of an ARMA process
N_g	order of the FIR filters in a polynomial mixing matrix
O	computational complexity order
$\mathbf{P}(z)$	elementary paraunitary matrix
Q	quantisation function
$\mathbf{Q}(\theta)$	real Jacobi rotation matrix
$\mathbf{Q}(\theta,\varphi)$	complex Jacobi rotation matrix
$\mathbf{r}(\tau)$	sample space-time covariance matrix for the vector of input sequences $\mathbf{x}(t)$
$\mathbf{R}(z)$	sample polynomial covariance matrix for the input polynomial vector $\underline{X}(z)$
$\mathbf{R}(e^{j\omega})$	sample PSD matrix for the output polynomial vector $\underline{X}(z)$
$\hat{\mathbf{R}}(z)$	windowed sample polynomial covariance matrix
r_{kk0}	sample variance of the k^{th} input signal to a filter bank
$\mathbf{s}(\tau)$	sample space-time covariance matrix for the vector of output sequences $\mathbf{y}(t)$

$\mathbf{S}(z)$	sample polynomial covariance matrix for output polynomial vector $\underline{Y}(z)$
s	sine of an angle θ
s_{kk0}	sample variance of the k^{th} output signal from a filter bank
t	general discrete time index or discrete time index at a low sample rate for subband processing
T	total number of data samples (i.e. data length)
T_c	maximum true correlation time (i.e. order of the covariance polynomials)
T_w	length of the window function $w(t)$
$v_k(t)$	k^{th} output subband signal from a filter bank
$w(t)$	window function
$\mathbf{x}(t)$	vector of input signals to a filter bank
$x(t)$	input signal to a filter bank at high sample rate
$x_k(t)$	k^{th} input signal to a filter bank
$\underline{X}(z)$	polynomial vector of input signals to a filter bank
$x(Mt)$	signal decimated by a factor M
$X(z)\|_{\downarrow M}$	z-transform of $x(Mt)$
$y(t)$	output signal from a filter bank at high sample rate
$y_k(t)$	k^{th} output signal from a filter bank
$\underline{Y}(z)$	polynomial vector of output signals from a filter bank

1 Introduction

1.1 Motivation

1.1.1 Filter Banks and Subspace Decomposition

The use of multirate systems in digital signal processing (DSP) has, in recent years, been at the forefront of modern technology. A focal topic in this area is subband processing, which is being exploited in an increasing number of applications, including digital communications [1,75,78], image and audio coding [12,16,32,63], beamforming [84] and channel coding [41,85]. Fig. 1.1 shows the block diagram of a subband processor in its polyphase equivalent form [73]. A subband processor works by first converting the input data sequence $x(t)$ into demultiplexed signals $\mathbf{x}(\tau)$ [1]: time-delayed and decimated versions of the input signal. Then a bank of filters, often referred to as the analysis bank, is applied to these signals. These operations essentially split the spectrum of the input signal into multiple frequency bands or subband channels $\mathbf{v}(\tau)$. This allows for the application of different processing procedures to the different subband channels. The filter banks are allowed to operate at a lower sample rate than the rate of $x(t)$, which makes for greater computational efficiency.

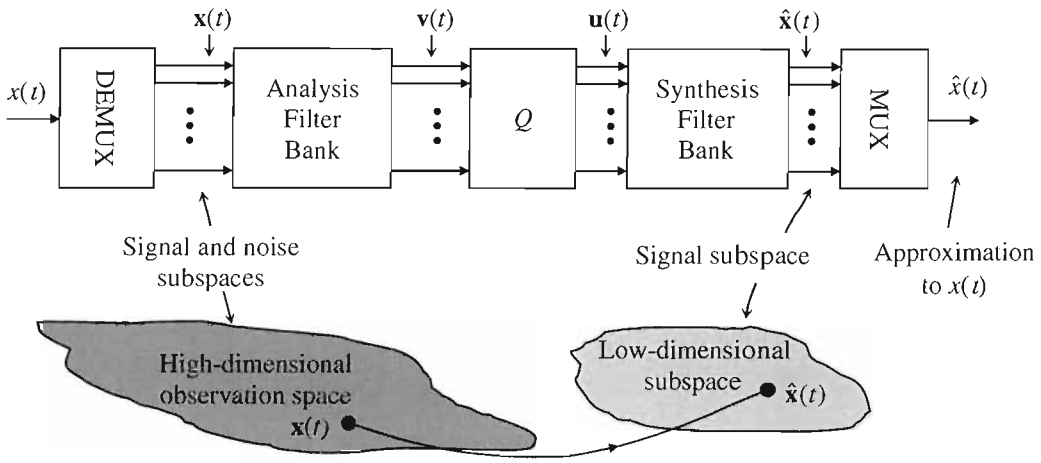
In coding applications, the subband processing stage, Q, typically involves quantisation which employs a bit allocation strategy. This is commonly referred to as subband coding. The quantised subband samples $\mathbf{u}(\tau)$ are then input into a reconstruction stage, called a synthesis bank, to form $\hat{x}(t)$: an approximation to $x(t)$. This procedure offers the possibility of reduced transmission rate, reduced storage requirement and/or increased signal representation accuracy.

Correlation between the demultiplexed signals is a type of redundancy. The analysis bank is essentially a linear transformation that is designed to remove these correlations and perform energy compaction: the compaction of as much of the total power into as few subband channels as possible. Since most of the total input signal power resides in a reduced number of subbands (dominant subbands) a correspondingly reduced number of bits is required to represent (encode) the information. As a consequence the input data is compressed. An appropriate bit allocation strategy to adopt in this case is one that assigns more bits to dominant subbands and fewer bits

---

[1] In the context of filter banks, we choose to denote the time index for the input signal by t, the Greek letter 'iota', and the time index for the decimated sequences by τ.

to the subbands with lower power (weak subbands), which usually also results in more accurate representation of the data. This is in contrast to scalar (uniform) quantisation where the same number of bits is allocated to every sample.



DEMUX: demultiplexer

MUX: multiplexer

Figure 1.1: Subband processor and its connection to subspace decomposition.

If the power of the weak subband signals is small enough then they may be assigned zero bits, i.e. they may be discarded. It is easy to see from this how subband processing can be applied to the problem of noise reduction (denoising) for additive white noise [3,8]: to reduce the noise power in the inputs $\mathbf{x}(t)$, only a subset of the transformed signals $\mathbf{v}(t)$ (those with high energies) are allowed to propagate to the reconstruction stage. As a result, the reconstructed signals $\hat{\mathbf{x}}(t)$ have a higher signal-to-noise ratio than the inputs. In this light, a subband coder may be viewed as a process that extracts a low-dimensional subspace from a higher-dimensional observation space as depicted in Fig. 1.1. The transformation is a matrix decomposition that permits the partitioning of the observation space into two orthogonal spaces; viz., the signal-plus-noise subspace and a noise-only subspace[2]. This process is sometimes known as subspace decomposition [22]. Those subband channels that contain information

[2] Note that we will refer to the signal-plus-noise subspace as simply the 'signal subspace' since the signal dominates this subspace. Also note that we use the term 'noise subspace' to mean the subspace that contains signals from the noise process only.

energy constitute the signal subspace. The rest of the channels represent the noise in $\mathbf{x}(t)$ and correspond to the noise subspace.

If the demultiplexed signals $\mathbf{x}(t)$ are related by instantaneous correlations only, then the eigenvalue decomposition (EVD) can be used to decorrelate the signals and perform subspace decomposition [20,22]. The EVD computes the eigenvalues and eigenvectors (basis vectors) of the Hermitian covariance matrix calculated from the signals $\mathbf{x}(t)$. The eigenvectors constitute a matrix of complex scalars which is said to be a unitary matrix: an energy preserving transformation. Satisfaction of the unitarity property guarantees that the eigenvalues represent the true power associated with the transformed signals. As a result, it is often possible to identify and separate the signal and noise subspaces in terms of the magnitudes of the eigenvalues. It follows that the EVD implicitly performs energy compaction. In the context of subband coding, the EVD is known as the Karhunen-Loeve transform (KLT) and has been shown to be optimal (in the coding gain sense) for subband coding, as well as noise reduction, under certain conditions [24,73]. An alternative to the EVD is the singular value decomposition (SVD) [20,21] which, unlike the EVD, may be applied directly to the signals themselves. Although the SVD is sometimes preferred for its slightly better arithmetic precision, the EVD and SVD are regarded as essentially equivalent in this thesis.

The SVD also plays an important role in narrowband sensor array signal processing. In this application, a multi-sensor antenna array is used to receive a mixture of signals arriving from different directions. The direction-of-arrival of a signal is one of the most important parameters to estimate, which can usually be achieved using high resolution direction finding techniques, such as the MUSIC algorithm [23,59]. The SVD forms the basis of this technique. If all the signals received by all the sensors can be described in terms of their relative phases and amplitudes then the beamformer is referred to as narrowband. The propagation of the source signals to the sensors may be represented as a matrix of complex scalars, that is, each received signal is an instantaneous mixture of the source signals. Under these conditions, decorrelation of the received signals, and subspace decomposition for that matter, may be achieved using the SVD. The unitary matrix found by the SVD is designed to modify the phase and amplitude of the signals and to combine them such that their estimated covariance matrix is diagonal. The SVD is also used as the first stage of instantaneous blind signal separation (BSS) [10] because it acts as a decorrelator.

Over recent years, there has been a growing requirement for subspace estimation of broadband signals [9,79]. This is also necessary, for example, if the signals to be processed are the weighted-sum of past and present samples of sources, e.g. if they are a convolutive mixture of the sources. The mixing cannot be modelled by a scalar mixing matrix; instead a matrix of

finite impulse response (FIR) filters is required. If each filter of the mixing matrix is represented as a polynomial, corresponding to its transfer function, the propagation model then takes the form of a polynomial (mixing) matrix [31,73]. Such a transformation produces sensor signals that relate to a covariance matrix whose entries are polynomials, i.e. a *polynomial covariance matrix*, which is *parahermitian*: a natural extension of the Hermitian property to the space of polynomial matrices.

In the case of convolutively mixed signals, the sensor outputs will generally be correlated with one another. However, they can no longer be decorrelated using the EVD (or SVD), which only measure and remove instantaneous correlation, i.e. correlation between pairs of signals sampled at the same instant in time. This is not sufficient for accurate broadband signal subspace estimation. Therefore, it is necessary to impose decorrelation, not just at the same time instant for all signals, but over all relative time delays. This property is known as *strong decorrelation* [74] and a matrix of suitably chosen FIR filters is required to achieve it.

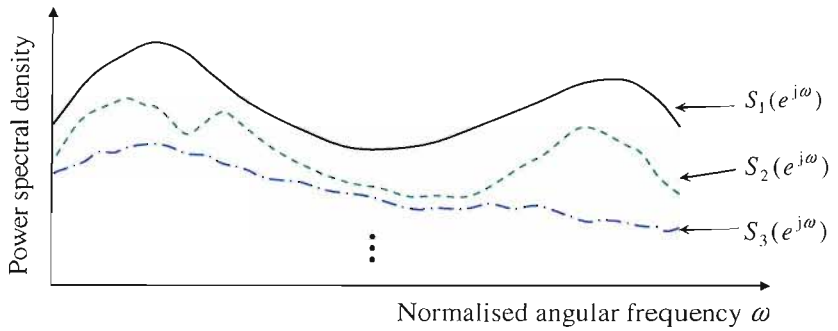
1.1.2 Paraunitary Filter Bank Design Methods

One way of diagonalising a polynomial parahermitian matrix is to apply a transformation that is a generalisation of the EVD to polynomial matrices. A critically important feature of the EVD is that the total spectral energy of the received signals is preserved after the decomposition process. The extension of this to polynomial matrices is the requirement that the transformation preserves the total signal power at every frequency. Such a transformation is called a *paraunitary* polynomial matrix and represents a multi-channel all-pass filter [73]. Polynomial matrices have been used for many years in the area of control [31]. They play an important role in the realisation of multi-variable transfer functions associated with multiple-input multiple-output (MIMO) systems. Numerical procedures have been developed for a range of polynomial matrix factorisation and reduction operations such as the Smith-McMillan decomposition [19,73]. To date, however, very little attention seems to have been devoted to polynomial matrix decompositions that are natural extensions of the EVD (or SVD) to broadband signals, i.e. a polynomial matrix EVD (PEVD).

Subband Coder Design Methods

In subband coding, the problem of designing a PEVD has been tackled indirectly, from the viewpoint of filter bank optimisation for subband coding [13,16,29,32-35,48,56,68-77,88]. One body of work on this problem concerns the design of energy compaction filters for the case where the input signal $x(t)$ is a known zero mean, wide-sense stationary (WSS) process [32,48,74]. In [74], Vaidyanathan has shown that optimality can be achieved using unrestricted (ideal) filters to construct paraunitary filter banks. He proves, under mild assumptions, that an

ideal optimal paraunitary subband coder satisfies strong decorrelation and a property known as *spectral majorisation*: the ordering imposed on the signal variances at each frequency is independent of frequency. An example of spectrally majorised subbands is shown in Fig. 1.2. Spectral majorisation is seen as a generalisation of the eigenvalue ordering done by the SVD.



$S_k(e^{j\omega})$: power spectral density of the k^{th} output signal from a subband coder

Figure 1.2: Example of spectrally majorised signals.

A filter bank that is optimal for subband coding is known as the principal component filter bank (PCFB) [68]. In fact, the PCFB has been shown to be optimal for other objectives, e.g., noise reduction [3,4]. In general, the existence of a PCFB cannot be assured for FIR filter banks, except for the special case of two-channel filter banks [33]. A number of authors have proposed design methods for suboptimal paraunitary FIR filter banks [13,16,29,32,48,53-56,69,77,88]. A technique that uses linear programming to design such a filter bank is presented in [13,48]. Regalia and Huang [56] propose a method based on the parameterisation of fixed degree two-channel lossless (i.e. stable, causal and all-pass) multivariable lattice filters [73]. The filter optimisation is computationally complex and nonlinear; the update equations resemble those of standard gradient descent methods. A simpler and more efficient approach is proposed in [32], called the window method. The basic idea is to window Vaidyanathan's ideal filter bank solution. This technique can be used to design two-channel FIR paraunitary filter banks; a design strategy for the multiple-subband case has been conjectured in [32] with little detail.

Signal Subspace Analysis Methods

Other authors have presented paraunitary filter bank design algorithms in the context of broadband subspace decomposition [17,37,49,57]. Another fixed degree parameterisation technique is introduced in [57] (a similar approach to that in [56] above). In contrast to the other

method, the filter bank is constructed using infinite impulse response (IIR) filters, which raises stability issues.

A fairly obvious way of diagonalising a polynomial covariance matrix is to split the input spectra into narrow frequency-bands (subbands) and use the SVD to decorrelate the signals at each subband. Lambert *et al* [37] propose a frequency-domain technique based on this premise for broadband signal subspace estimation that is proclaimed to be an extension of the SVD to polynomial matrices. The method involves applying an EVD at each frequency to a set of conditioned data and the approximate inversion of FIR filters. The drawbacks with these frequency-domain approaches mean that they are limited in the extent to which the signals are decorrelated and they introduce phase discontinuities around the band edges. These are problems typically encountered with the independent frequency band (IFB) approach used in space-time adaptive processing for phased array radar [36].

In [5], a novel time-domain technique is proposed for addressing the convolutive BSS problem. A variant of the algorithm which is relevant to this thesis has two stages: the first stage uses second-order statistics to design a paraunitary matrix for imposing strong decorrelation upon the signals, and is referred to as the second order sequential best rotation (SBR2) algorithm [44]; the second stage uses a fourth-order cost function to construct a paraunitary separation matrix, and is called the fourth order sequential best rotation (SBR4) algorithm. Both stages involve constructing a paraunitary matrix in an iterative fashion; at each step of the algorithm a time-delay matrix and a rotation matrix is found, which constitute an elementary paraunitary matrix that maximises the cost function. The two algorithms construct a paraunitary matrix as the product of elementary paraunitary matrices.

An initial evaluation of the SBR2 algorithm has revealed that it can perform strong decorrelation, to a good approximation, in the case of two convolutively mixed signals. This is a crucial property of a polynomial matrix EVD (PEVD). However, further development and assessment of this technique is required to improve its applicability, e.g. a method for extending it to the multichannel case; a thorough investigation of its behaviour and performance to show, for example, the level of accuracy achieved in the estimation of broadband signal subspaces.

Comparison of Methods

There is a fundamental difference between traditional subband coder design methods and the signal subspace analysis techniques. The latter are designed to process arbitrary multichannel data. By contrast, the former are typically intended for processing a single known input sequence which is assumed to originate from a WSS process [74].

1.2 Thesis Objectives and Contributions

The purpose of this thesis is to present a contribution to further understanding of polynomial matrix decompositions for the task of broadband signal subspace estimation and their significance in the design of orthonormal FIR filter banks for subband coding. To this end, the research presented focuses on the development of the second-order sequential best rotation (SBR2) algorithm. A frequency-domain counterpart, referred to as the bandwise algorithm, is also developed. A major part of the thesis is then concerned with the extension of the SBR2 algorithm for effective application to broadband subspace decomposition and subband coding. The following summarises the novel contributions made by this thesis.

A family of multichannel frequency-domain polynomial matrix EVD (PEVD) methods, known as the bandwise algorithm, is introduced. Here, we use classical methods for spectral estimation [28,50] in defining new PEVD estimation algorithms. In particular, a sliding-window bandwise algorithm is introduced which essentially reduces the variance of the spectral estimates in the manner of [87]; a windowed covariance-domain bandwise algorithm that adapts a well-known technique for spectral estimation in [7] to the problem of PEVD; and a method for improving the frequency resolution of the bandwise algorithm.

The SBR2 algorithm has been taken from a very basic form to one which handles multiple channels; offers a choice of cost function; and has an extended range of applicability. Variations on the algorithm have been designed and characterised, which offer significant improvement in performance. This work includes the following:

- Extension of SBR2 to the multichannel case by generalisation of the classical Jacobi algorithm [20] to polynomial matrices, and development of an understanding of the spectral majorisation performed by the algorithm.
- Development of the algorithm for broadband subspace decomposition, particularly broadband noise reduction for sensor arrays, which has been published in [44]; further joint work on the algorithm has been submitted for publication [45,46]. The algorithm has also been demonstrated as applicable to the problem of channel coding [41,85], which is outside the scope of this thesis.
- Adaptation of an efficient spectral estimation technique in [7] to the SBR2 algorithm, which meant applying a window function to the entries of the polynomial covariance matrix.
- Derivation of a new cost function for SBR2 which is based on the coding gain for subband coding [73].

The SBR2 algorithm is adapted to the design of M-channel uniform, paraunitary filter banks [53]. This involved designing a data pre-processing stage for exploitation of knowledge about the signal statistics, as performed by traditional subband coder design methods. A solution to the negative semidefiniteness problem associated with windowing the space-time covariance matrix is presented, which is used to great effect with the SBR2 algorithm.

1.3 Overview

In the following, a brief overview of the remaining chapters in this thesis is presented.

Chapter 2 Fundamental concepts in linear algebra and multirate DSP are described on which the rest of the thesis builds. To begin with, a review of subspace decomposition for narrowband signals is presented, which includes a discussion on matrix decompositions and algorithms for their computation. Multi-input multi-output (MIMO) systems are presented in the context of filter banks along with an introduction to the z-transform, polynomial matrices and paraunitarity. We conclude this chapter with a study of multirate systems.

Chapter 3 A review of the literature on the theory and optimisation of orthonormal filter banks for subband coding is provided which is relied on by the work in Chapter 6. We begin by describing the different classes of filter banks; orthonormality and perfect reconstruction are introduced. The problem of optimising subband coders is discussed. The link between optimal compaction filters, optimal orthonormal subband coders and the principal component representation of signals is explained. We present a review of FIR paraunitary filter bank design methods for subband coding. Difficulties in using the Vaidyanathan structure are demonstrated. Finally, a computationally efficient technique [32] for compaction filter design, known as the window method, is assessed and used as a benchmark later.

Chapter 4 We introduce the concept of polynomial matrix EVD as a means of solving the problem of subspace estimation for broadband signals and present various algorithms for this task. The extension of the EVD and SVD from scalar matrices to the realm of polynomial matrices is considered. An understanding of broadband subspace decomposition is given along with some fundamental limitations. Two performance measures are introduced for evaluating the performance of polynomial matrix EVD algorithms. The current state-of-the-art in subspace estimation for broadband signals proposed by Lambert [37] is reviewed, including an evaluation of the algorithm. Well known limitations connected to frequency-domain techniques are discussed and an alternative method is presented: the bandwise algorithm. Extensions of this algorithm are introduced, evaluated and compared to the prior art.

Chapter 5 The SBR2 algorithm is presented as a novel time-domain polynomial matrix EVD technique that avoids the problems associated with the Vaidyanathan structure. We extend the classical Jacobi algorithm to the domain of polynomial matrices, resulting in a simple algorithm that can impose on a set of signals strong decorrelation and spectral majorisation, to a good approximation. Variations on the algorithm are discussed, including a version that uses a new cost function based on the coding gain measure; a 'windowed' covariance-domain approach to the algorithm which gives a significant saving in computational cost and improves the subspace estimation accuracy. A new strategy for covariance-domain windowing is proposed which can be used to produce a windowed space-time covariance matrix that is positive semidefinite; this is necessary when using the coding gain based cost function. Efficiency issues regarding the SBR2 algorithm are highlighted and alleviating techniques are introduced. This includes replacing a slow iterative time-domain process with a more economical covariance-domain approach. Finally, we consider an alternative search technique for SBR2 that is based on the cyclic-by-rows Jacobi algorithm.

Chapter 6 The applicability of the SBR2 algorithm to the problem of data compression is investigated. While the chapter does describe a modification of the algorithm that is suited for the encoding of multichannel arbitrary broadband data, the focus is on extending and adapting SBR2 for subband coding. We start by defining a new measure that is an extension of the coding gain for multichannel data. The PEVD is related to the PCFB. The relevance of SBR2 to traditional filter bank design methods is examined, highlighting the WSS assumption made by these techniques. This study motivates the development of a method whereby the SBR2 algorithm can exploit this knowledge. The resulting algorithm, called the SBR2 coder, is compared with the prior art (window method) [32]. Finally, we present experimental results which suggest that the SBR2 coder outperforms the window method on a set of benchmark problems.

Chapter 7 A summary of overarching conclusions is presented along with recommended avenues of further work to be pursued.

Appendices Contain some proofs and definitions.

2 Preliminary Technical Background

In this chapter, a review of relevant mathematical tools and fundamental signal processing concepts is provided. This serves as reference material for the thesis and familiarises the reader with notations and symbolisms used throughout. We start by classifying signals in terms of their bandwidth and their interactions with systems in section 2.1. This serves to define and distinguish between the narrowband and broadband problems. This is followed by a discussion on subspace decomposition and its application to data compression and noise reduction in section 2.2. From a linear matrix algebra point of view, we will focus here on the SVD and the classical Jacobi algorithm. In section 2.3, the linear time-invariant multi-input multi-output (MIMO) system is treated along with an introduction to the polynomial matrix, which provides a compact representation. We discuss the paraunitary property and its importance in the design of such systems. Finally, in section 2.4, the basic components of multirate systems and digital filter banks are discussed. A filter bank may be represented by a MIMO system, which allows for more efficient implementation and easier analysis.

2.1 Signal Classification

In DSP, signals are represented, stored, and processed as discrete sequences of real or complex numbers. These are usually referred to as samples of the analogue signal. We denote discrete-time signals as $x(t)$, $y(t)$, etc, where t is an integer called the (*discrete*) *time index*. We use these notations to either indicate the entire sequence (e.g. $-T \le t \le T$) or to denote the t^{th} sample. The context in which the notation is used will clarify the exact meaning.

2.1.1 Linear Independence

Let the discrete-time, complex-valued sequences $w_k(t)$ and $w_m(t)$ be realisations from the stochastic, ergodic random processes $W_k(t)$ and $W_m(t)$, respectively. The *mean* of $W_k(t)$ is given by $\mu_k = \mathrm{E}[W_k(t)]$ [1], where $\mathrm{E}[\cdot]$ is the statistical expectation operator. This is generally dependent on t, the discrete-time index. Henceforth, we assume that statistical processes have

---

[1] For brevity, we will use lower case letters, e.g. $w_k(t)$, to denote both a random process and a realisation from a random process. So, we can write $\mathrm{E}[W_k(t)] = \mathrm{E}[w_k(t)]$.

zero mean; this assumption does not cause any difficulty in the application of a system to stationary signals, since the system can be made a zero-mean system by subtracting the mean from the input signals. The quantity

$$c_{km}(t,\tau) = \mathrm{E}\left[w_k(t)w_m^*(t-\tau)\right] \tag{2.1}$$

is the *crosscorrelation* function (or *crosscovariance* sequence since $\mu_k = \mu_m = 0$), where the superscript asterisk denotes complex conjugation. The crosscovariance is a measure of the linear dependence between the two sequences at different discrete times t and τ. The sequence $c_{kk}(t,\tau)$ is the *autocorrelation* (*autocovariance*) of $w_k(t)$.

Wide-sense Stationary Processes. A stochastic process is said to be wide-sense stationary (WSS) iff: (i) $\mathrm{E}[w_k(t)] = \mathrm{E}[w_k(t-\tau)]$, for all integers t and τ and (ii) $c_{kk}(t,\tau)$ is independent of t, i.e. $c_{kk}(\tau) = \mathrm{E}\left[w_k(t)w_k^*(t-\tau)\right]$, $\forall\tau$. The variable τ in $c_{kk}(\tau)$ is called the time-lag. This condition is usually assumed for stochastic processes encountered in many applications, such as communications channel modelling, image processing and speech processing. For example, speech signals may be assumed to be WSS when observed over short (and much longer) time periods – except in pathological conditions. The analysis of linear systems becomes easier when signals are assumed to be WSS. In particular, the wide-sense stationarity assumption simplifies subband coder analysis and design to a great extent (see chapters 3 and 6). In this thesis, we assume that random processes are stationary in the wide sense, unless stated otherwise.

Two other useful quantities can be defined using $c_{kk}(\tau)$. The mean-square value or *variance* of a WSS sequence $w_k(t)$ can be written as

$$c_{kk}(0) = \mathrm{E}\left[\left|w_k(t)\right|^2\right]. \tag{2.2}$$

The power spectrum or power spectral density (PSD) of this sequence can be defined, by the Wiener-Khinchin theorem [80], as the Fourier transform of its autocovariance function:

$$C_{kk}(e^{j\omega}) = \sum_{\tau=-\infty}^{\infty} c_{kk}(\tau)e^{-j\omega\tau}. \tag{2.3}$$

The crosscovariance, variance and PSD are usually preceded by 'true' to indicate that the quantity is an expected value, i.e., the true statistics are known. For example, $c_{kk}(\tau)$ may also be referred to as the *true autocovariance* function. In real applications, the true statistics are often unknown and samples from a given (noisy) data set have to be used instead to formulate and solve a problem. Therefore, there are only a finite number of samples T available, and only estimates of the correlations can be formed. In this thesis, we typically assume that T is fixed.

An estimate of the true crosscovariance can be obtained by using noisy data samples $x_k(t)$ and replacing the ensemble average with a time average:

$$s_{km}(\tau) = \frac{1}{T}\sum_{t=0}^{T-1} x_k(t)x_m^*(t-\tau)\,. \tag{2.4}$$

We refer to this as the *sample crosscovariance* (or sample autocovariance if $k = m$) function to distinguish it from (2.1). The sample variance, $s_{kk}(0)$, and the sample PSD, $S_{kk}(e^{j\omega})$, may be defined similarly.

Covariance Matrix. Let $\mathbf{x}(t) \in \mathbb{C}^M$ represent the samples of M discrete-time, stochastic, random, zero mean, WSS processes at time t. Then the covariance matrix is defined as [86]:

$$\mathbf{C} = \mathrm{E}\left[(\mathbf{x}-\mu_{\mathbf{x}})(\mathbf{x}-\mu_{\mathbf{x}})^{\mathrm{H}}\right] = \mathrm{E}\left[\mathbf{x}\mathbf{x}^{\mathrm{H}}\right] = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1M} \\ c_{21} & c_{22} & \cdots & c_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ c_{M1} & c_{M2} & \cdots & c_{MM} \end{pmatrix}, \tag{2.5}$$

where $(.)^{\mathrm{H}}$ denotes conjugate transposition. Key properties of the covariance matrix are that it is positive definite (or positive semidefinite) and, by definition, it is Hermitian (self-adjoint): a matrix that is equal to its own conjugate transpose, i.e. $\mathbf{C} = \mathbf{C}^{\mathrm{H}}$. This is equivalent to the condition that $c_{km}(\tau) = c_{mk}^*(-\tau)$ for crosscovariance sequences. Important properties of a Hermitian matrix are that it can be diagonalised by a unitary matrix; all its eigenvalues are real and positive semidefinite; and the eigenvectors form an orthonormal basis. This type of decomposition is referred to as the EVD, which is described in section 2.2.

2.1.2 Instantaneous and Convolutive Mixing

The signals $w_m(t)$ are said to be linear *instantaneous mixtures* of the sources $x_k(t)$ if they can be expressed as a weighted sum of present samples of the individual sources, i.e.

$$w_m(t) = \sum_{k=1}^{K} g_{mk}\, x_k(t)\,, \tag{2.6}$$

where the g_{mk} are some scalars. The mixing does not involve time-delaying any of the sources. It is clear from (2.6) that the signals $w_m(t)$ will generally be correlated with each other.

By contrast, *convolutive mixing* involves combining samples of $x_k(t)$ at different time-delays with different weighting coefficients. This is equivalent to passing the signals $x_k(t)$ through a FIR filter whose impulse response, $h_k(t)$, corresponds to the weighting coefficients in the discrete-time convolution, i.e.

$$y_k(t) = h_k(t) * x_k(t) = \sum_{\tau=0}^{T} h_k(\tau) x_k(t-\tau) , \tag{2.7}$$

where the symbol $*$ denotes the convolution operator. The resulting mixed signal is usually the linear combination of the filtered signals $y_k(t)$, i.e. $y(t) = \sum_{k=1}^{K} y_k(t)$. For a generalisation of this to multiple mixed signals refer to section 2.3.3.

In the problem of BSS, the type of unmixing required for successful signal separation depends on the interaction of signals with systems. As a consequence, the problem of BSS is commonly divided into two categories, viz., instantaneous BSS and convolutive BSS. The former is often used to describe the problem of BSS of instantaneously mixed narrowband signals. The latter term is typically used to describe BSS for convolutively mixed broadband signals. The convolutive problem is considered by some to also encompass the case of instantaneously mixed broadband signals and convolutively mixed narrowband signals.

2.1.3 Narrowband and Broadband Signals

There are many ways to classify signals. A categorisation that is relevant to this thesis is one based on a signal's bandwidth: the band of frequencies for which the signal takes on non-zero values. Let B and f_c be the bandwidth and the centre frequency of a signal, respectively. Then, the following two types of signal are defined in terms of their fractional bandwidth $\Delta f = B / f_c$:

Narrowband Signals. A signal is said to be *narrowband* if the inequality $\Delta f < 0.05$ is satisfied [51]. A narrowband signal with a bandwidth of zero gives rise to a true autocorrelation sequence with infinite time extent (unbounded in time). The covariance matrix for a set of narrowband signals is a scalar matrix.

Broadband Signals. It follows from the definition of narrowband signals that a signal is *broadband* if $\Delta f > 0.05$. The covariance matrix of such signals has polynomial entries; the definition for polynomial matrices is given in section 2.3.2.

In the context of sensor array processing, the case of instantaneously mixed narrowband signals is often said to be a 'narrowband problem'. Another situation that falls under this category is that of convolutively mixed narrowband signals, provided the signals have the same frequency. In both these cases, the time taken by the impinging waveform to travel from sensor-to-sensor can be represented accurately as phase shifts, so beamforming can often be achieved using just complex gains. On the other hand, convolutively mixed broadband signals typically present a situation where the delay between sensors cannot be represented accurately as phase shifts alone [52]. This is sometimes termed the 'broadband problem'. A broadband beamformer, which can implement the required time delays, is often necessary [9,79].

2.2 Subspace Decomposition

The extraction of a lower dimensional subspace from a data space is a requirement in many DSP applications [10,22,73]. The first step in achieving this is the application of a matrix decomposition that finds a, usually orthonormal, basis for the entire M-dimensional observation (data) space, $X \in \mathbb{R}^M$. An orthonormal basis is a set of M linearly independent, orthonormal vectors that span X. In many cases, the data, say $\mathbf{x}(t)$, represents a set of desired (information) signals as well as noise and possibly redundancy. It is often possible to represent the desired signals by a linear combination of K of the basis vectors, where $K \leq M$. This subset of vectors span a space $S \in \mathbb{R}^K$, called the signal-plus-noise subspace or simply the signal subspace. The orthogonal complement of S is referred to as the noise subspace $N \in \mathbb{R}^{M-K}$. The dissection of the space X into S and N is sometimes referred to as *subspace decomposition* [22,31]. An approximation to $\mathbf{x}(t)$ can be obtained by an orthogonal projection of $\mathbf{x}(t)$ onto the signal subspace, which produces a set of new signals $\hat{\mathbf{x}}(t)$ as illustrated in Fig. 2.1. The success of subspace decomposition techniques depends on the method for estimating the orthonormal basis vectors and the determination of a suitable signal-subspace dimension, i.e., the number of basis vectors used in the projection.

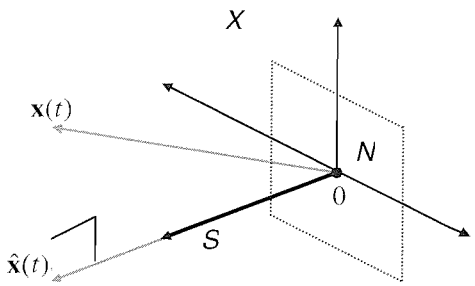


Figure 2.1: Orthonormal projection involved in subspace decomposition. The vector $\mathbf{x}(t)$ is projected from the entire vector space X on the signal subspace S. The noise subspace N is the plane outlined by the dotted lines.

Depending on the area of application, the matrix decomposition appears in various guises and is known by different names. In statistics, it is called principal components analysis (PCA) and has the property that the variance of the signal subspace is maximised. This is a criterion that has been applied in many applications, for example, pattern recognition and signal subspace estimation [22,23]. It is also used as the first step in BSS [10] since it may be viewed as minimising second-order statistical dependencies between the input signals. That is, PCA may

be used to perform *second-order* BSS on instantaneously mixed narrowband signals. It is blind in the sense that no assumptions are made about either the original source signals or the mixing process. A complementary property of the PCA is that the projection of the input data onto the signal subspace minimises the reconstruction error in the mean-square sense: the distortion between the input data and the approximation. In this context, the PCA is known as the KLT, particularly in the subband coding community [34,73].

The KLT can be computed easily using the EVD, which decomposes a Hermitian covariance matrix related to complex data into a set of eigenvalues and eigenvectors. The eigenvectors are the bases for the data; the largest eigenvalues correspond to the signal subspace. In many applications, a numerical technique known as the SVD is used instead [20,21]. The SVD is one of the most important algorithms in numerical algebra for providing accurate information about the structure of a system of linear equations. The SVD is sometimes preferred over the EVD because it finds a more arithmetically precise solution than that found by the EVD and can be applied to an arbitrary matrix, such as to the input data matrix.

2.2.1 Singular Value Decomposition

The SVD can generate a set of decorrelated signals by linear transformation of a set of correlated signals. Suppose the vector $\mathbf{x}(t) = [x_1(t), x_2(t), \ldots, x_M(t)]^{\mathrm{T}}$ represents M correlated signals having true statistics as stated in section 2.1.1. Given T samples of these signals $\{\mathbf{x}(t) \mid t = 0,1,\ldots T-1\}$ with $T \geq M$, the matrix $\mathbf{X} \in \mathbb{C}^{M \times T}$ may be defined to be $\mathbf{X} = [\mathbf{x}(0), \mathbf{x}(1), \ldots, \mathbf{x}(T-1)]$. The SVD finds matrices $\mathbf{U} \in \mathbb{C}^{M \times M}$ and $\mathbf{V} \in \mathbb{C}^{M \times T}$ such that [20]

$$\mathbf{X} = \mathbf{U}^{\mathrm{H}} \mathbf{D} \mathbf{V}, \tag{2.8}$$

where $\mathbf{D} = \mathrm{diag}[d_1, d_2, \ldots, d_M] \in \mathbb{R}^{M \times M}$ and the operator diag[·] takes a vector to a diagonal matrix. The elements d_k are known as singular values and conventionally they are ordered as $d_1 \geq d_2 \geq \ldots \geq d_M \geq 0$. The matrices $\mathbf{U}$ and $\mathbf{V}$ are unitary, i.e.

$$\mathbf{U}^{\mathrm{H}} \mathbf{U} = \mathbf{U} \mathbf{U}^{\mathrm{H}} = \mathbf{I} \in \mathbb{R}^{M \times M} \text{ and } \mathbf{V} \mathbf{V}^{\mathrm{H}} = \mathbf{I} \in \mathbb{R}^{M \times M}. \tag{2.9}$$

This property can be re-expressed by saying that the transformation is energy preserving:

$$\sum_{k=1}^{M} |\mathbf{x}_k|^2 = \mathrm{trace}[\mathbf{X}\mathbf{X}^{\mathrm{H}}] = \mathrm{trace}[\mathbf{U}^{\mathrm{H}} \mathbf{D}^2 \mathbf{U}] = \mathrm{trace}[\mathbf{D}^2] = \sum_{k=1}^{M} d_k^2, \tag{2.10}$$

where $\mathbf{x}_k$ denotes the k^{th} row of the matrix $\mathbf{X}$ and the operator trace[·] sums over the diagonal elements of a matrix. In words, the l_2-norm of the signals is invariant under a unitary transformation.

The rows of **DV** may be regarded as samples from a set of decorrelated signals defined by

$$\mathbf{y}(t) = \mathbf{U}\mathbf{x}(t) = \mathbf{D}\mathbf{v}(t) \in \mathbb{C}^{M\times T}, \tag{2.11}$$

where $\mathbf{v}(t) \in \mathbb{C}^{M}$ is a column taken from the matrix $\mathbf{V}$ at time t. This follows from the unitary property of $\mathbf{U}$ and $\mathbf{V}$ since

$$\mathbf{C} = \mathrm{E}\left[\mathbf{y}(t)\mathbf{y}^{\mathrm{H}}(t)\right] \cong \mathbf{D}\mathbf{V}\mathbf{V}^{\mathrm{H}}\mathbf{D}/T = \mathbf{D}^{2}/T \tag{2.12}$$

and the energy of the m^{th} signal $y_m(t)$ is d_m^2. The matrix $\mathbf{C} \in \mathbb{C}^{M\times M}$ is the outer-product true covariance matrix for the output signals.

Let the sample covariance matrix for the inputs be $\mathbf{R} = \mathbf{X}\mathbf{X}^{\mathrm{H}}/T \in \mathbb{C}^{M\times M}$. The matrix $\mathbf{R}$ has two special properties, which are that it is (i) Hermitian, i.e. $\mathbf{R}^{\mathrm{H}} = \mathbf{R}$, and (ii) positive semidefinite [21]. The SVD corresponds to computing the EVD of $\mathbf{R}$, i.e. performing the unitary diagonalisation given by

$$\mathbf{U}\mathbf{X}\mathbf{X}^{\mathrm{H}}\mathbf{U}^{\mathrm{H}} = \mathbf{D}^{2}. \tag{2.13}$$

The diagonals of $\mathbf{D}^2$ are the eigenvalues of $\mathbf{R}$, which are real valued and non-negative. The rows of $\mathbf{U}$ (left singular vectors) are the eigenvectors of $\mathbf{R}$. The matrix $\mathbf{U}$ is unique, up to a unitary transformation of $\mathbf{R}$.

2.2.2 Subspace Decomposition and its Applications

Consider a $M\times T$ data matrix $\mathbf{X}$ comprising of M correlated narrowband signals $\mathbf{x}(t)$, which are now in the presence of additive white Gaussian noise. Furthermore, suppose that $T >> M$ and that the signal-to-noise ratio (SNR) of the signals is sufficiently high. The unitary matrix $\mathbf{U}$ found by the SVD may be applied to the signals to produce M new decorrelated signals: $\mathbf{y}(t) = \mathbf{U}\mathbf{x}(t)$. The unitary matrix is designed to modify the signals in phase and amplitude and combine them such that their estimated correlation matrix is diagonal. Since the transformation is unitary, the square of the associated singular values represent the true energy associated with each of the decorrelated components. The decorrelation of the signals may be construed as a second-order blind signal separation process[2]. Since the SNR of the signals is high, the last $M - K$ singular values of the diagonal matrix $\mathbf{D}$ in (2.8) are very small, for $1 \le K \le M$. A new approximate reconstruction of the signals $\mathbf{x}(t)$ is:

---

[2] Note that, in general, higher order statistics are exploited to determine the "hidden" rotation matrix for completion of the separation process.

$$\hat{\mathbf{x}}(t) = \mathbf{U}^{\mathrm{H}}\mathbf{P}\mathbf{U}\mathbf{x}(t)\,. \tag{2.14}$$

Here, $\mathbf{P}$ is a projection matrix of the form $\mathrm{diag}[p_1, p_2, \ldots, p_k, \ldots, p_M]$, where $p_k = 1$, for $1 \le k \le K$ and $p_k = 0$, for $K < k \le M$. This is the Karhunen-Loève expansion truncated at $k = K$, as illustrated in Fig. 2.2. We may use the system in the figure to perform a number of tasks, particularly data compression and noise reduction, as will be explained next.

Noise Reduction

Consider the problem of increasing the SNR of desired narrowband signals in noisy data derived, for example, from an antenna sensor array. The diagonal matrix in (2.8) is approximately of the form $\mathbf{D} = \begin{pmatrix} \mathbf{D}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{I}\sigma \end{pmatrix}$ since the signals are decorrelated from the noise. Here, σ is the square root of the noise variance, $\mathbf{I}\sigma$ is a $(M-K)\times(M-K)$ matrix, and $\mathbf{D}_1$ is a $K\times K$ diagonal matrix with elements $d_k >> \sigma$, for $1 \le k \le K$. The matrix $\mathbf{D}$ is partitioned, which corresponds to separating large singular values in $\mathbf{D}_1$ from smaller ones in $\mathbf{I}\sigma$. Appropriately, the (unmixing) matrix $\mathbf{U}$ is partitioned as $\mathbf{U} = [\mathbf{U}_1, \mathbf{U}_2]^{\mathrm{T}}$, where $\mathbf{U}_1$ comprises the eigenvectors $\mathbf{v}_1^{\mathrm{T}}, \mathbf{v}_2^{\mathrm{T}}, \ldots, \mathbf{v}_K^{\mathrm{T}}$ corresponding to $\mathbf{D}_1$, and $\mathbf{U}_2$ contains the eigenvectors $\mathbf{v}_{K+1}^{\mathrm{T}}, \mathbf{v}_{K+2}^{\mathrm{T}}, \ldots, \mathbf{v}_M^{\mathrm{T}}$ corresponding to $\mathbf{I}\sigma$. The rows of $\mathbf{U}_1$ define the signal-plus-noise subspace. The rows of $\mathbf{U}_2$ define the space orthogonal to the signal subspace; this is the complement of the signal subspace, sometimes called the noise subspace, although the whole M-dimensional space contains values from the noise.

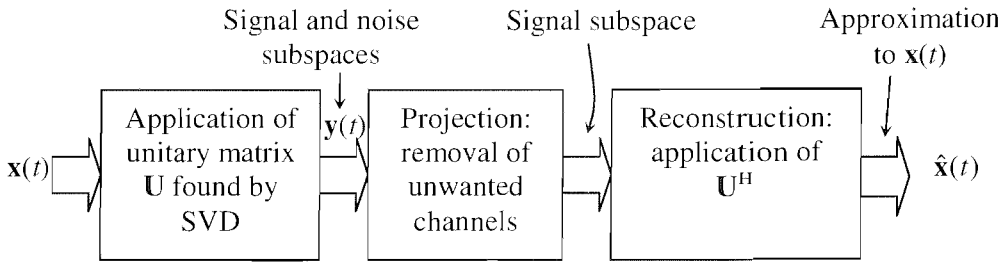


Figure 2.2: Block diagram representation of the Karhunen-Loève expansion.

Note that the distribution of energy between the signal and noise subspaces is physically meaningful because of the invariance of the l_2-norm under the unitary transformation. Also note that, if there were two signals with very different power levels, the process described here would accomplish signal separation, to a large extent.

Data Compression

Consider the task of minimally coding multichannel data. Correlation between the input signals $\mathbf{x}(t)$ is a type of redundancy. The SVD (or EVD) removes this redundancy by removing instantaneous cross-correlations between the signals. If we keep the first K of the decorrelated signals $\mathbf{y}(t)$ and discard the rest there will be a corresponding reconstruction error. The unitary matrix found by the SVD is the one that minimises this error and thus most of the information in the input data is contained in the first K signals. In other words, amongst all unitary choices of $\mathbf{U}$, the total energy in the first K signals, for $1 \leq k \leq K$, is maximised; this property is known as *energy compaction* in subband coding [73,74]. This choice of $\mathbf{U}$ minimises (amongst all unitary $\mathbf{U}$) the product of the data channel variances, i.e. $\prod_{k=1}^{M} | y_k(t) |^2$. After possibly a quantisation stage, which is omitted in Fig. 2.2, the amount of storage capacity required to hold the data produced by all the sources is minimised, to a fixed level of accuracy. In (2.14), the $\mathbf{U}\mathbf{x}(t)$ and the $\mathbf{U}^H\mathbf{y}(t)$ parts of the expression may be seen, respectively, as the analysis stage and the synthesis stage found in filter banks (discussed later in section 2.4).

2.2.3 Classical Jacobi Algorithm

The EVD can be computed in a step-by-step fashion using the classical *Jacobi algorithm*, which consists of a sequence of orthogonal similarity transformations [20]. Each transformation is an elementary plane rotation (*Jacobi rotation*) designed to selectively zero an off-diagonal of a matrix. Consider a positive semidefinite symmetric (real) matrix $\mathbf{R} \in \mathbb{R}^{M \times M}$, which may be, for example, the sample covariance matrix for M real signals. Note that the transformation required for the more general case of a Hermitian matrix is discussed later in section 5.2. The Jacobi rotation matrix is defined for the intersection of rows n and p with columns n and p as:

$$\mathbf{Q}^{(n,p)}(\theta) = \begin{pmatrix} \mathbf{I}_{n-1} & & & & \mathbf{0} \\ & c & & s & \\ & & \mathbf{I}_{p-n-1} & & \\ & -s & & c & \\ \mathbf{0} & & & & \mathbf{I}_{M-p} \end{pmatrix}, \tag{2.15}$$

where c and s denote the cosine and sine respectively of the angle $\theta \in \mathbb{R}$ and $\mathbf{I}_{M-p}$ is the $(M-p)\times(M-p)$ identity matrix. Notice that $\mathbf{Q}^{(n,p)}(\theta)$ is an orthonormal matrix. In a single step of the Jacobi algorithm, the plane rotation in (2.15) is used to transform $\mathbf{R}$ according to

$$\mathbf{R}' = \mathbf{Q}^{(n,p)}(\theta)\mathbf{R}\mathbf{Q}^{(n,p)^{\mathrm{T}}}(\theta), \tag{2.16}$$

where $(.)^T$ denotes matrix transposition. Using the symmetric property of $\mathbf{R}$, multiplying out (2.16), and setting $r'_{np} = 0$ we have that

$$r'_{np} = \left(\cos^2\theta - \sin^2\theta\right) r_{np} + \cos\theta\sin\theta\left(r_{pp} - r_{nn}\right) = 0 . \tag{2.17}$$

If $r_{np} = 0$ then the matrix is already diagonal and we can set $\theta = \pi/2$. Otherwise, we have the following expression for the rotation angle:

$$\tan 2\theta = \frac{2r_{np}}{r_{nn} - r_{pp}} . \tag{2.18}$$

The denominator $r_{nn} - r_{pp}$ and the numerator $2r_{np}$ of the quotient in (2.18) are the components of a vector $\mathbf{v}$. The task of zeroing the cross-correlation terms r_{np} and r_{pn} may be interpreted as one of rotating the vector $\mathbf{v}$ onto the $(r_{nn} - r_{pp})$-axis so that the $2r_{np}$ component of the vector vanishes. One possible solution to (2.17) is to apply a clockwise rotation of 2θ radians so that $\mathbf{v}$ is brought onto the positive $(r_{nn} - r_{pp})$-axis, as illustrated by the example shown in Fig. 2.3.

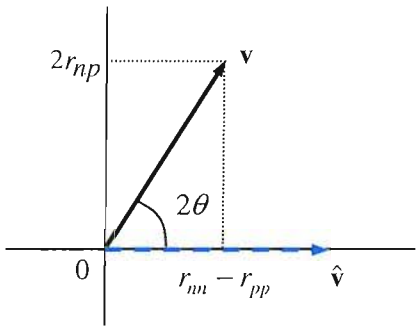


Figure 2.3: Plane rotation of a symmetric matrix.

The Jacobi algorithm applies a sequence of such transformations to produce a matrix $\mathbf{S}$ that is more diagonal than $\mathbf{R}$. It can be thought of as iteratively modifying the symmetric matrix such that the off-diagonal terms diminish with each step. The EVD orthogonal matrix $\mathbf{U}$ comprises the product of successive Jacobi rotation matrices $\mathbf{Q}_t$. This is true since every orthogonal (or unitary) matrix can, up to a channel negation, be decomposed into the product of a sequence of rotation matrices [73]. The operations carried out by the Jacobi algorithm are as follows:

Initialise: $\mathbf{R}' := \mathbf{R}$ and $\mathbf{U} := \mathbf{I}_M$.

Iterate: $\mathbf{R}' \leftarrow \mathbf{Q}^{(n,p)}(\theta)\mathbf{R}'\mathbf{Q}^{(n,p)^{\mathrm{T}}}(\theta)$ and $\mathbf{U} \leftarrow \mathbf{Q}^{(n,p)}(\theta)\mathbf{U}$.

It follows that after every step, the $\mathbf{R}'$ is still symmetric and the identity $\mathbf{U}\mathbf{R}\mathbf{U}^{\mathrm{H}} = \mathbf{R}'$ continues to hold.

The $\mathbf{Q}^{(n,p)}(\theta)$ chosen is the one that annihilates the off-diagonal in $\mathbf{R}'$ with the largest magnitude. It turns out that $\mathbf{Q}^{(n,p)}(\theta)$ minimises the sum of the squares of the off-diagonal terms. Equivalently, since the transformation is unitary, it maximises the sum of the squares of the diagonal terms. Successive transformations undo some of the off-diagonals that were previously annihilated, but nevertheless at each stage of the algorithm, the sum of the squares of the diagonal terms of $\mathbf{R}'$ increases. Using this fact, it can be shown that the off-diagonal elements get smaller and smaller until the symmetric matrix is diagonal to machine precision. It is complete after a permutation (orthogonal transformation) is applied to the channels to order the diagonal of $\mathbf{S}$.

An alternative strategy for applying plane rotations with the aim of obtaining the EVD (or SVD) is the cyclic Jacobi algorithm [20,21]. The algorithm applies the plane rotation defined in (2.15) to the symmetric matrix $\mathbf{R} \in \mathbb{R}^{M \times M}$ for a total of $(M^2 - M)/2$ different index pairs. Such a sequence of transformations is called a sweep. The pairs are selected on a fixed cyclic-by-rows or cyclic-by-columns basis. As an example, consider the following transformations:

$T_{12} : \mathbf{R}' \leftarrow \mathbf{Q}_{12}\mathbf{R}\mathbf{Q}_{12}^{\mathrm{T}}$ (operation on signal pair 1,2)

$T_{13}T_{12} : \mathbf{R}' \leftarrow \mathbf{Q}_{13}\mathbf{Q}_{12}\mathbf{R}\mathbf{Q}_{12}^{\mathrm{T}}\mathbf{Q}_{13}^{\mathrm{T}}$ (then operation on signal pair 1,3)

$T_{14}T_{13}T_{12} : \mathbf{R}' \leftarrow \mathbf{Q}_{14}\mathbf{Q}_{13}\mathbf{Q}_{12}\mathbf{R}\mathbf{Q}_{12}^{\mathrm{T}}\mathbf{Q}_{13}^{\mathrm{T}}\mathbf{Q}_{14}^{\mathrm{T}}\cdots$ (then operation on signal pair 1,4, and so on)

A different and more popular method for computing the EVD (or SVD) is the QR algorithm, which is generally computationally superior to the Jacobi algorithms described above [20]. However, in comparison to the QR algorithm, the Jacobi algorithms are very simple, and therefore they are attractive for extension to the realm of polynomial matrices. In chapter 4, we present an algorithm that is essentially a generalisation of the classical Jacobi algorithm to polynomial matrices. The extension of the cyclic-by-rows Jacobi algorithm to the same field is also considered to a lesser extent in the chapter.

2.3 Multi-input Multi-output systems

2.3.1 The z-transform

Signal processing analysis is often simplified by considering the frequency domain representation of signals and systems. A commonly used alternative representation of $x(t)$ is its z-transform:

$$X(z)=\sum_{-\infty}^{\infty}x(t)z^{-t}. \tag{2.19}$$

The z-transform encodes the data as polynomials, power series, Laurent series or Laurent polynomials in z^{-1} (a two-sided polynomial) [15,86]. We will use the term polynomial to include Laurent polynomials. Note that the Fourier transform, $X(e^{j\omega})$, of $x(t)$ is $X(z)$ evaluated on the unit circle, i.e. for $z=e^{j\omega}$; the evaluation of $X(z)$ for different points on the unit circle in the z-plane gives the discrete Fourier transform (DFT) of $x(t)$, provided $x(t)$ has finite sample support.

A feature of the z-transform representation is that operations such as convolution or the filtering of data corresponds to taking the product of two polynomials. For a filter $g(t)$ of order $2N_g$, the convolution sum (as in (2.7)) may be expressed as

$$G(z)X(z)=\sum_{t}z^{-t}\sum_{-N_g}^{N_g}g(\tau)x(t-\tau). \tag{2.20}$$

2.3.2 Polynomial Matrices

An *M×M polynomial matrix* $\mathbf{H}(z)$ in the indeterminate variable z^{-1} can be expressed as

$$\mathbf{H}(z)=\sum_{t=-t_1}^{t_2}\mathbf{h}(t)z^{-t}=\begin{pmatrix}H_{11}(z) & H_{12}(z) & \cdots & H_{1M}(z)\\ H_{21}(z) & H_{22}(z) & \cdots & H_{2M}(z)\\ \vdots & \vdots & \ddots & \vdots\\ H_{M1}(z) & H_{M2}(z) & \cdots & H_{MM}(z)\end{pmatrix}, \tag{2.21}$$

where $t\in\mathbb{Z}$ $t_1\geq 0$, $t_2>0$,

$$H_{km}(z)=\sum_{t=-t_1}^{t_2}h_{km}(t)z^{-t} \tag{2.22}$$

are (Laurent) polynomials in the indeterminate z^{-1} and $\mathbf{h}(t)\in\mathbb{C}^{M\times M}$ or equivalently we write $\mathbf{H}(z)\in\mathbb{C}^{M\times M}(z)$. The elements of the matrix $\mathbf{h}(t)$ are denoted by $[\mathbf{h}(t)]_{km}=h_{km}(t)$. In keeping

with the standard notation for linear systems and signal processing, we have chosen to denote the indeterminate variable by z^{-1} since this is normally used to represent a unit delay. Multiplying a polynomial by z^{-1} will sometimes be referred to as applying a delay. Since the leading term of $z^{-t_1}\mathbf{H}(z)$ is constant, the effective *order* of $\mathbf{H}(z)$ is $N = t_2 + t_1$. If $t_1 = 0$ and $h(t_2) \neq 0$, then $\mathbf{H}(z)$ represents a FIR system of order $N = t_2$. The matrix $\mathbf{H}(z)$ may also be viewed as a matrix polynomial, i.e., a polynomial whose coefficients are matrices. Polynomial vectors and vector polynomials are defined accordingly.

Polynomial matrices provide an easy way of describing and analysing multidimensional systems. They have been used for many years in the areas of control [31] and multirate systems [73], and more recently in MIMO communications [47,58]. Polynomial matrices play an important role in this thesis. In particular, they are used to represent the estimated covariance matrix of broadband (or convolutively mixed) signals, which we refer to as a polynomial covariance matrix, as discussed in chapter 4.

Degree of a Polynomial Matrix. In general, the order of a polynomial matrix is different from its *degree* (or McMillan degree). The degree of the polynomial matrix $\mathbf{H}(z)$ is the minimum number of delay elements (i.e. z^{-1} elements) required to implement it[3] [73].

Example. Let

$$\mathbf{H}(z) = \begin{pmatrix} z^{-1} & 1 \\ -1 & z^{-1} \end{pmatrix}. \tag{2.23}$$

It is clear that, in order to implement this polynomial matrix, at least two delays are required; see Fig. 2.4. So the degree of $\mathbf{H}(z)$ is two, whereas the order is $N = 1$. A tool for computing the degree is defined in [73], which uses the Smith-McMillan decomposition of a polynomial matrix.

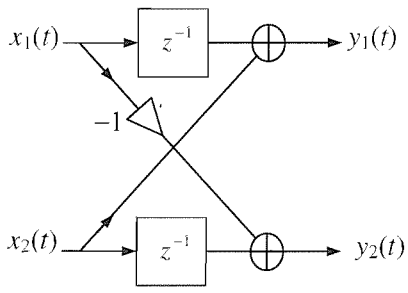


Figure 2.4: Implementation of the system in (2.23).

[3] Note that the degree is not defined for Laurent polynomials (or non-causal systems) because they cannot by implemented purely using delays.

Properties of Polynomial Matrices

Paraconjugation. The paraconjugate of a polynomial matrix $\mathbf{H}(z) \in \mathbb{C}^{M\times M}(z)$ is defined as:

$$\tilde{\mathbf{H}}(z) = \mathbf{H}^{\mathrm{H}}(1/z^*), \tag{2.24}$$

where the superscript asterisk represents complex conjugation. Example: Let $A(z) = b - c\,z^{-1}$, then $\tilde{A}(z) = b^* - c^*\,z$. If $a(k)$ is real, then $\tilde{A}(z) = A(z^{-1})$. Note that in the degenerate case of an order-zero polynomial matrix, this corresponds to the usual Hermitian transpose.

Parahermitian. The polynomial matrix $\mathbf{H}(z)$ is referred to as parahermitian if it is identical to its paraconjugate, i.e. if

$$\tilde{\mathbf{H}}(z) = \mathbf{H}(z). \tag{2.25}$$

In other words,

$$[\mathbf{h}(t)]_{km} = h_{km}(t) = h^*_{mk}(-t) = [\mathbf{h}(-t)]^*_{km}, \quad k,m \in \{1,2,...M\}. \tag{2.26}$$

This property requires the limits of summation, t_1 and t_2 as defined in (2.21), to be identical. It should be noted that in the degenerate case of an order-zero polynomial matrix, this corresponds to the usual Hermitian (symmetric) matrix.

Paraunitarity. Recall from section 2.2.1, that the SVD could be used to provide a unitary transformation of a matrix of scalars. The extension of the unitary property to polynomial matrices leads to the definition of the *paraunitary* property [73]. The polynomial matrix $\mathbf{H}(z)$ satisfies the paraunitary property iff

$$\mathbf{H}(z)\tilde{\mathbf{H}}(z) = \tilde{\mathbf{H}}(z)\mathbf{H}(z) = \mathbf{I}. \tag{2.27}$$

An important property of a paraunitary matrix $\mathbf{H}(z)$ is that it is an energy-preserving transformation, that is, in terms of a MIMO system, the total spectral energy of the input signals is conserved; this is shown later in chapter 4. This implies that the filters of the system have constant gain over all frequencies (i.e. it is a multidimensional extension of the all-pass filter). If the paraunitary system comprises causal FIR (stable) filters, then we say that it is *lossless*, a natural choice in the design of FIR filter banks – see chapter 3.

2.3.3 FIR MIMO Systems

Consider the multi-input multi-output (MIMO) linear time-invariant (LTI) system with K inputs and N outputs shown in Fig. 2.5. Say $\mathbf{x}_k(t) = [x_k(0), x_k(1), \ldots, x_k(t), \ldots]^{\mathrm{T}}$ is the k^{th} input. Let $\mathbf{X} = [\mathbf{x}_1(t), \mathbf{x}_2(t), \ldots, \mathbf{x}_k(t), \ldots, \mathbf{x}_K(t)]^{\mathrm{T}}$. Then the time series in $\mathbf{X}$ may be represented as a

vector of algebraic power series: $\underline{X}(z) = [X_1(z), X_2(z), \ldots, X_K(z)]^T$. An LTI system is characterised by its impulse response or equivalently its transfer function. The MIMO system can be characterised by $\mathbf{H}(z) \in \mathbb{C}^{M \times K}(z)$, whose entries $h_{km}(t)$ may be viewed as the impulse responses between the k^{th} input to the m^{th} output. Equivalently, the system may be thought of as a set of transfer functions $H_{mk}(z)$. Assuming the transfer functions are LTI digital filters (e.g. FIR filters), the m^{th} output of $\mathbf{H}(z)$ in response to all inputs is given by

$$Y_m(z) = \sum_{k=1}^{K} H_{mk}(z) X_k(z), \qquad 1 \le m \le M \ . \tag{2.28}$$

The operation in (2.28) can be described compactly as

$$\underline{Y}(z) = \mathbf{H}(z)\underline{X}(z) \in \mathbb{C}^M(z), \tag{2.29}$$

where $\underline{Y}(z) = [Y_m(z)]$ is a vector of power series corresponding to the outputs from this transformation.

The MIMO systems we consider in this thesis are limited to the case where $M \ge K$, i.e. for square polynomial matrices or the overdetermined problem. This is advantageous in, for example, communications and sensor array signal processing, where the number of antenna sensors is equal to or greater than the number of source signals. Such a scenario provides a sufficient number of degrees-of-freedom for exploitation of the spatial diversity. A typical application of MIMO systems is communications, where for example, they may be employed to overcome bandwidth limitations and increase signal quality [66]. In multirate DSP, the representation of a filter bank as a MIMO system results in more efficient implementations and easier analysis [73].

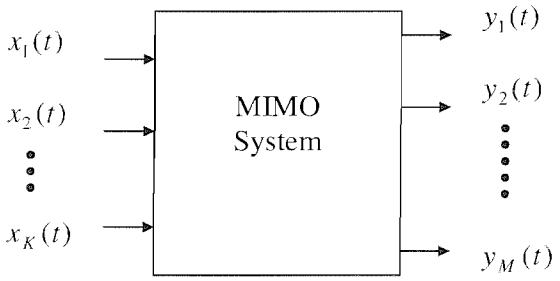


Figure 2.5: A K-input M-output LTI system.

2.4 Introduction to Multirate Systems

2.4.1 Basic Components

In DSP, digital signals are often obtained by sampling and quantising continuous-time signals. The discretisation of a continuous-time signal leads to the notion of resolution in time and amplitude. The higher the sample rate and the larger the number of quantisation levels the more accurate the digital approximations. The cost of having higher sample rates and more quantisation levels is of course larger amounts of data to process. The widespread popularity of multirate DSP systems is mainly due to their ability to process data in a very efficient manner and thus reduce these costs. This is achieved by allowing sampling rate variation throughout the system and, moreover, keeping the sample rate at internal points to a minimum. Multirate DSP techniques have been exploited in many applications, including spectrum analysis, digital communications [1,75,78], image and audio coding [32,63,71], equalisation [82] and noise reduction [3]. They offer the possibility of, for example, reduced communications bandwidth, reduced storage requirement, improved noise rejection, and/or reduced computational complexity.

A multirate DSP system is usually composed of three basic building blocks that operate on a signal $x(t)$ [4]. Two of these are decimation and interpolation, which enable the efficient alteration of the data rate. An M-fold decimator and the expander stage of the interpolator are shown in Fig. 2.6. The decimation retains every M^{th} sample, i.e. the sampling rate at the output of the decimator is M times lower than the rate of the input. The expander inserts $M-1$ zeros in between every original sample. The decimation process generates frequency-shifted stretched versions (spectral images or alias components) of the original information signal. If the original signal is not suitably bandlimited, then direct decimation of the signal will cause aliasing: corruption of the baseband signal spectrum by overlapping alias components [73]. Without some sort of compensation, aliasing results in the loss of information. For example, an analytic bandpass signal of bandwidth b_w can be decimated by a factor $M = \pi / b_w$ without creating overlap of the images.

The other building block is a transformation, such as a LTI system (filter), as shown in Fig. 2.7(a). The transformation usually serves to transform $x(t)$ into a domain where processing is more convenient. The transformation can be viewed as a decomposition of the signal into

---

[4] When the distinction is required, we use the variable t as an index for the samples of a high rate signal and the variable t for indexing the samples of a low rate (decimated) signal (i.e. the subband signals).

basis vectors, and the inverse transformation may be looked upon as the reconstruction with the transform coefficients. The systems shown in Figs. 2.6 and 2.7 are single-input signal-output systems since they operate on a scalar signal (e.g. $x(t)$).

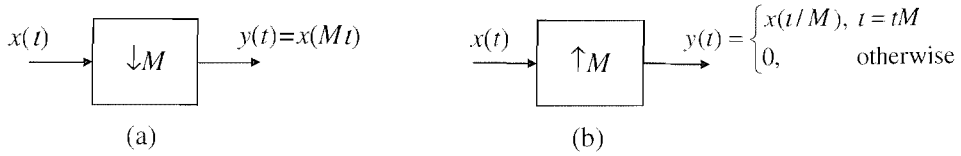


Figure 2.6: (a) decimation and (b) expansion.

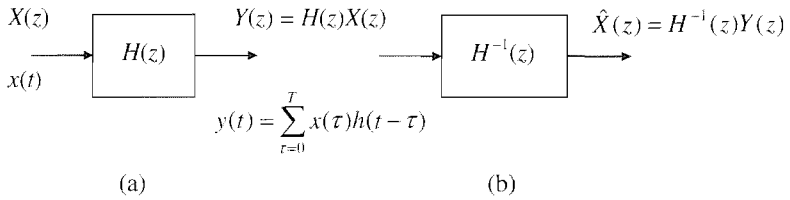


Figure 2.7: (a) filter operation (LTI system) and (b) its inverse.

2.4.2 Filter banks

Another type of transformation, which is used predominantly in multirate DSP, is the filter bank, as given in Fig. 2.8. The input signal $x(t)$ is split into M (high sample rate) bandlimited signals, $x_k(t)$, by a set of filters $\{H_k(z)\}$ called the analysis bank. Each filtered signal $x_k(t)$ is then decimated to produce new (low rate) time sequences or *subband signals*, $v_k(t)$. This process is known as subband decomposition. In Fig. 2.9, we show a typical set of frequency responses for the analysis filters. Each of the subbands corresponds to a different portion of the input power spectrum. The type of subband processing used is application dependent, although generally they are converted to digital form by a quantiser, which constrains the amplitudes of the signals to a set of discrete values. To reconstruct the original signal, the signals $v_k(t)$ are expanded by a factor of M and passed through a synthesis bank $\{G_k(z)\}$ before they are recombined. The recombined output, $y(t)$, is at the same sampling rate as $x(t)$.

We often desire the inverse of a transformation in order to generate (or approximate) the original input signal. In Fig 2.7, the signal $x(t)$ is transformed by the LTI filter $H(z)$ to

produce $x(t)$. A simple method of recovering $x(t)$ from $y(t)$ is to pass $y(t)$ through the inverse filter, $H^{-1}(z)$. Such a system is said to be a perfect reconstruction system. However, in the case of the filter banks in Fig. 2.8 it is more difficult to design a perfect reconstruction system on the basis that the synthesis filters form the inverse of the analysis bank. This is because the decimation and expansion stages introduce aliasing, which needs to be cancelled by the combination of the analysis and synthesis banks. Given the set of FIR filters $\{H_k(z)\}$, an FIR-inverse for the synthesis bank, $\{F_k(z)\}$, does not necessarily exist. The inverse may consist of infinite impulse response (IIR) filters. A problem with using IIR filters is that they are notoriously unstable. However, this is not the only way of achieving perfect reconstruction. In [73], Vaidyanathan shows that it is possible to construct a perfect reconstruction system using only FIR filters. Any aliasing caused by the decimation stage of a filter bank can be cancelled by careful selection of the analysis and synthesis banks.

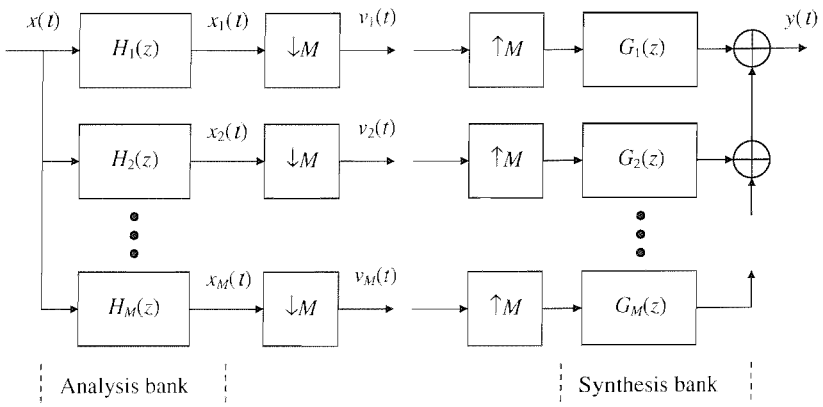


Figure 2.8: Analysis and synthesis filter bank for multirate DSP.

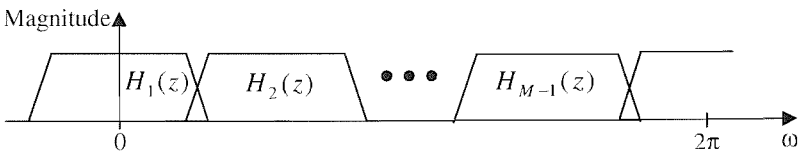


Figure 2.9: Typical frequency response of an analysis (or synthesis) bank.

Polyphase Representation

An important concept in multirate DSP is the *polyphase decomposition* [6,73] of signals and systems. It can considerably simplify theoretical results and produce computationally efficient

implementations of multirate systems. Since the polyphase representation of filter banks plays an important role in the rest of this thesis, we formally define it here. Let $H(z)=\sum_{-\infty}^{\infty} h(t)z^{-t}$ be the transfer function of a digital filter. It is possible to decompose $H(z)$ as:

$$H(z)=\sum_{k=0}^{M-1} z^{-k}E_k(z^M) \qquad \text{(Type 1 polyphase representation)}, \tag{2.30}$$

where $E_k(z)=\sum_{-\infty}^{\infty} e_k(t)z^{-t}$ are called the Type 1 polyphase components of $H(z)$. The coefficients of each polyphase component are given by $e_k(t)=h(Mt+k),\ 0\le k\le M-1$. In words, the impulse response sequence $h(t)$ is divided into M non-overlapping subsequences $e_k(t)$, and the k^{th} subsequence is merely the M-fold decimated version of $h(t+k)$. Note that the quantity $E_k(z)$ depends on M for a given $H(z)$, e.g. the lengths of the sequences $e_k(t)$ depend on M, and are less than the length of $h(t)$.

Essentially, polyphase decomposition allows for the representation of a filter bank as a MIMO system that operates on a vector signal, $\mathbf{x}(t)$. The vector $\mathbf{x}(t)$ is obtained by passing the scalar signal $x(t)$ through a delay-chain and decimation (or *demultiplexing* or blocking) network shown in Fig. 2.10(a). It has been shown that the analysis bank in Fig. 2.8 is equivalent to having a demultiplexer network followed by a MIMO system in [73], as shown in Fig. 2.11. The noble identities [73] have been used to obtain the representation shown. The analysis bank is now described by the $M\times M$ polynomial (polyphase) matrix $\mathbf{E}(z)\in\mathbb{C}^{M\times M}(z)$. Similarly, the synthesis bank may be represented in this way as shown in Figs. 2.10(b) and 2.11.

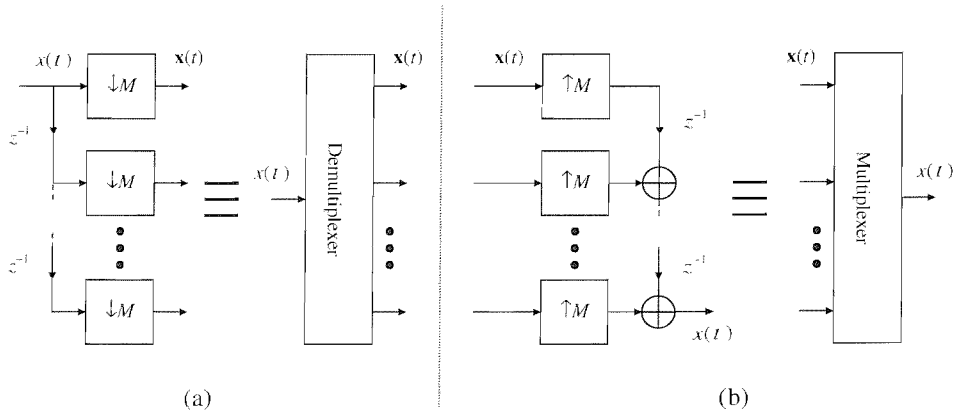


Figure 2.10: (a) Conversion of a scalar signal to a vector signal and (b) conversion of a vector signal to a scalar one.

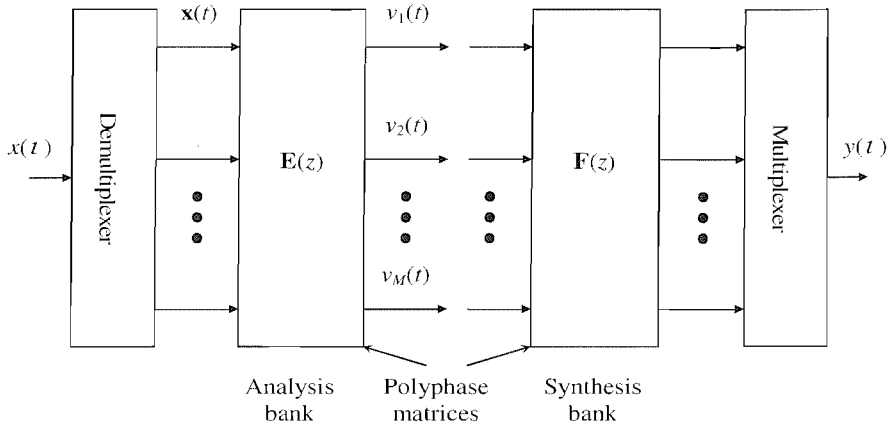


Figure 2.11: Polyphase representations of the analysis and synthesis banks.

We can now see that a filter bank is a natural extension of a matrix transform for scalar matrices by comparing Figs. 2.2 (ignoring the transform domain operation shown) and 2.11. The MIMO representation has converted sequential processing to parallel processing. This makes for more efficient implementation since all the processing (filtering) is performed at lower sampling frequencies. For example, the sample rate of the input $\mathbf{x}(t)$ to $\mathbf{E}(z)$ in Fig. 2.11 is M times lower than that of the input signal to the filters $\{H_k(z)\}$ in Figs. 2.8.

2.5 Concluding Remarks

The aim in this chapter has been to provide some basic concepts that are used in the remainder of this thesis. We have introduced the EVD (and SVD) for instantaneous decorrelation of narrowband signals from which the discussion has progressed to considering their use for subspace decomposition in two typical applications: data compression and noise reduction. We have presented the classical Jacobi algorithm for implementation of the EVD. An introduction to MIMO systems and polynomial matrices was presented. We have looked at the extension of unitarity to polynomial matrices, which lead to the notion of a paraunitary matrix. A short study of multirate building blocks and their interconnections with digital filters is provided. This includes an introduction to filter banks and their polyphase representation. With the introduction of polynomial matrices we have motivated the extension of classical EVD / SVD algorithms to the case of polynomial matrices, which will be addressed, in one form or another, in the next three chapters.

3 Theory and Design of Optimal Orthonormal Filter Banks

In this chapter, a review of relevant theory and results for the design of filter banks is presented. We begin by defining different classes of filter banks in section 3.1. An introduction to the optimisation of orthonormal filter banks for subband coding is provided in section 3.2. Optimal orthonormal filter banks and optimal compaction filters are found by the principal component filter bank (PCFB), if one exists. A PCFB exists for ideal filter banks, block transforms, and any two-channel filter bank, but, in general, will not exist for FIR filter banks of arbitrary number of channels. The procedure in designing a PCFB using ideal filters is described in section 3.3, which is adopted by a state-of-the-art FIR filter bank design tool, called the window method. This method, along with other prior-art techniques, is studied in section 3.4. A short evaluation of the window method is presented, which provides a benchmark for algorithms proposed later.

3.1 Filter Bank Classification

Consider the filter bank in Fig. 3.1. The filter bank shown is said to be a *uniform* filter bank because each of the subband signals is decimated by the same factor, M [35,76]. In this case, the filters typically have identical but shifted frequency responses, as shown in Fig. 2.9. The average sampling rate over the subbands is equal to the input sampling rate. It is possible to have different decimation factors for different channels provided the overall number of samples across all subbands, per time interval, remains the same as that for the input signal. This type of filter bank is said to be non-uniform and is suited for coding of certain types of signal, e.g. those targeted for human perception. The filter bank shown in the figures are also said to be *maximally* (critically) *decimated* because the decimation factor is equal to the number of channels. In recent years, filter banks with M less than the number of subband channels have been shown to offer advantages over maximally decimated systems for applications such as equalisation of acoustics [82] and adaptive beamforming [84]. Our discussions in this thesis are restricted to the uniform, maximally decimated filter bank in Fig. 3.1. We consider subclasses of all such filter banks, as defined in the following.

Unconstrained Filter Banks. A filter bank with filters $H_k(z)$ that have no restriction on the order, N, is called an unconstrained filter bank or *ideal* filter bank. The filters of such a filter bank are allowed to be IIR filters and/or non-causal.

Block Transforms (or *transform coder*). A filter bank with the constraint that N is less than the number of channels M, i.e. $N < M$. In terms of the polyphase representation in Fig.3.1(b), this is the special case where $\mathbf{E}(z)$ and $\mathbf{F}(z)$ are scalar matrices, i.e. $\mathbf{E}$ and $\mathbf{F} \in \mathbb{C}^M$. The operation of the filter bank is simply a matrix transformation of the vector signal $\mathbf{x}(t)$ (the 'blocked' version of the input signal $x(t)$).

Constrained (FIR) Filter Banks. A filter bank with the constraint $N < \infty$. The matrices $\mathbf{E}(z)$ and $\mathbf{F}(z)$ of such a filter bank are polynomial matrices, as given in (2.21). In relation to the block transform, an FIR filter bank can be viewed as a lapped transform; i.e. the present output vector is the sum of the matrix transformations of the present and all past blocks of data $\mathbf{x}(t)$.

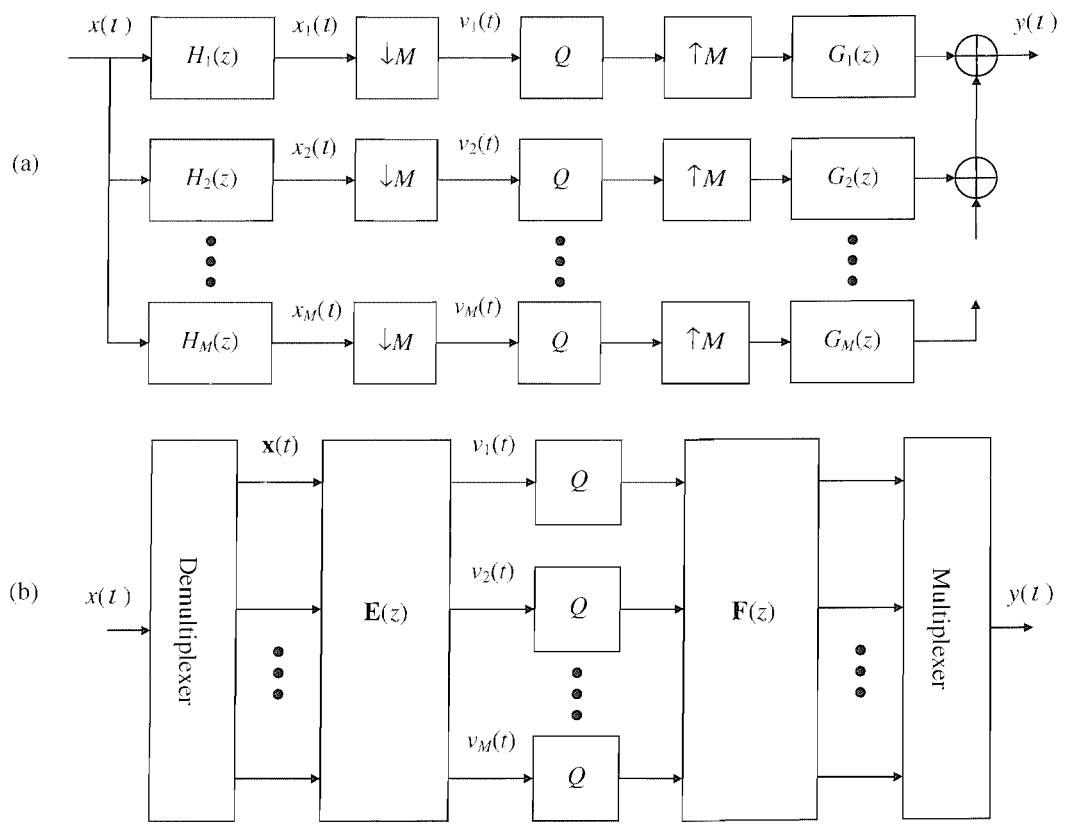


Figure 3.1: (a) M-band uniform, maximally decimated subband coder and (b) its polyphase equivalent.

Orthonormal Filter Banks. The set of filters $\{H_k(z)\}$ in Fig. 3.1(a) are orthonormal if [73]

$$H_k(e^{j\omega})H_m^*(e^{j\omega})\Big|_{\downarrow M} = \delta(k-m), \tag{3.1}$$

where $(\cdot)\big|_{\downarrow M}$ denotes the z-transform of a sequence that is decimated by a factor M. This, in particular, implies that each filter $H_k(e^{j\omega})$ satisfies the Nyquist(M) constraint:

$$\left|H_k(e^{j\omega})\right|^2\Big|_{\downarrow M} = 1, \text{ i.e. } \sum_{n=1}^{M} |H_k(e^{j(\omega-2\pi(n-1)/M)})|^2 = M. \tag{3.2}$$

In words, the filters $\{H_k(z)\}$ are orthogonal with impulse responses that have unit energy. Orthonormal filter banks have several advantages which make them very attractive to use:

1. The total signal energy in the subband channels is preserved, which guarantees that errors generated by the quantisation process (quantisation noise) are not amplified;
2. The design of either the analysis or synthesis bank is required only – see perfect reconstruction filter banks below;
3. If the analysis filters $\{H_k(z)\}$ are FIR, then the corresponding synthesis filters $\{G_k(z)\}$ and synthesis polyphase matrix $\mathbf{F}(z)$ are also necessarily FIR;
4. Allows for efficient implementations using unitary (Jacobi rotation) matrices.

The Nyquist(M) constraint translates into the following specification for a filter, $h(t)$: $h(0)=c$ and $h(Mt)=0$ for $t\neq 0$, where c is some constant. Such a filter is referred to as a Nyquist (or M^{th}-band) filter. An ideal brickwall filter, i.e. one that has a sinc impulse response, would satisfy this criterion; however, such a filter would be impractical for use in real systems. The Nyquist criterion does not define a unique spectrum and so other constraints can be included, e.g., a smooth transition band in the filter's frequency response. The Nyquist filters may also be understood from a heuristic stand point. For example, consider the design of the transmitter and receiver filters, $H(z)$ and $F(z)$ respectively, for data transmission through a bandlimited communications channel. The filters are designed such that there is zero intersymbol interference (ISI) at the sampling instants of the filtered signal. In the presence of additive white Gaussian noise, the SNR is maximised if $F(z)$ is matched to $H(z)$, i.e. $F(z)=\tilde{H}(z)$ [51], in which case both the zero ISI and the Nyquist(M) properties are satisfied. Nyquist filters are particularly useful in subband coding, as will be discussed in section 3.3.

The orthonormality of a filter bank can also be expressed in terms of the polyphase representation in Fig. 3.1(b). The filter bank is orthonormal if $\mathbf{E}(z)\tilde{\mathbf{E}}(z)=\mathbf{I}$ (or $\mathbf{E}(e^{j\omega})$ is

unitary for all ω), i.e. it is paraunitary. A transform coder ($\mathbf{E}(z) = \mathbf{E}$) is orthonormal when $\mathbf{E}$ is a unitary matrix.

Perfect Reconstruction Filter Banks. A *perfect reconstruction filter bank* (PRFB) is one that, in the absence of any subband-processing, yields the output [73]

$$y(t) = cx(t-\tau), \quad \forall t, \tau, \tag{3.3}$$

and for some constant $c \neq 0$, where $x(t)$ and $y(t)$ are, respectively, the input and output of the filter bank in Fig. 3.1. This implies that a PRFB minimises the mean-square error (MSE) or *reconstruction error*:

$$\varepsilon = \left| y(t) - x(t) \right|^2 . \tag{3.4}$$

To design a PRFB the polyphase synthesis bank $\mathbf{F}(z)$ is chosen to be the inverse of $\mathbf{E}(z)$ in Fig. 3.1(b). More generally, a filter bank has the perfect reconstruction property if:

$$\mathbf{F}(z)\mathbf{E}(z) = cz^{-\tau}\mathbf{I} \in \mathbb{C}^{M\times M}(z), \quad \text{for } \tau > 0 . \tag{3.5}$$

A PRFB is able to completely cancel *aliasing* and its outputs are free from amplitude distortion and phase distortion. The perfect reconstruction property is desirable in applications where lossless signal representation is required (e.g. lossless compression).

In most practical applications, it is important to find a PRFB with FIR filters $H_k(z)$ and $G_k(z)$ in order to avoid issues such as non-causality and instability, typically associated with IIR filters. It has been shown in [73] that a PRFB can be designed using only FIR filter banks by imposing the paraunitary property on $\mathbf{E}(z)$ and using its paraconjugate to design $\mathbf{F}(z)$, i.e.,

$$\mathbf{F}(z) = cz^{-\tau}\tilde{\mathbf{E}}(z) . \tag{3.6}$$

Example. A two-channel PRFB may be designed using the 20-tap lowpass FIR filter in [73] as the first filter, $H_1(z)$, of the analysis bank. The second analysis filter, $H_2(z)$, can be obtained simply by taking the time reversed version of $H_1(z)$ and alternately changing the sign of the resulting filter coefficients. This yields a filter with a highpass response. The frequency response of the FIR PRFB analysis filters are shown in Fig. 3.2. The synthesis bank is found simply by taking the paraconjugate transpose of $\mathbf{E}(z)$ corresponding to the filters $\{H_k(z)\}$.

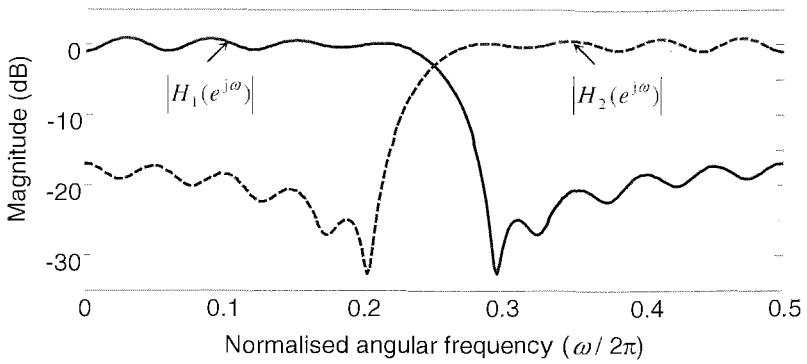


Figure 3.2: Magnitude-frequency responses of the analysis filters of a two-channel PRFB with coefficients as in [73].

3.2 Subband Coding and Optimisation

One of the most fruitful applications of filter banks has been subband coding [2,12,32-35,48,73,74,88]. The uniform, maximally decimated filter bank in Fig. 3.1 is shown in a subband coding scheme, where the blocks labelled Q are quantisers. This type of arrangement is commonly referred to as an *M-band subband coder*. The block transform coder, lapped transform coder [42] and wavelet coder [63,70] may be looked upon as subclasses of the subband coder. The subband coder can be used to compress (or minimally encode) the input data by quantising each subband signal, $v_k(t)$, with a different number of bits b_k. This is achieved through use of a *bit allocation strategy* [61,73], which has a direct impact on the coding performance of the subband coder.

An intuitive way of accomplishing data compression is to allocate the quantiser bits according to the signal spectrum. For example, more bits can be assigned to the subbands with greater energy (variance) and fewer bits to the subbands with lower variance, as depicted in Fig. 3.3. If a PRFB is used and there are subband channels with zero variance, then those channels may be discarded. Usually, this kind of strategy can also result in a more accurate representation of the signal since there are more bits being used for subbands that contain most of the energy related to the information. Hence, a subband coder may be thought of as a sophisticated quantiser that can encode data more efficiently and/or accurately than direct techniques, such as pulse code modulation (PCM), which allocates bits uniformly without discretion. By comparison, in a time-domain coding strategy, the $b_k s$ may be adapted according to the energy in the signal at different time periods. This is exploited by wavelet coders (zero-tree coding) for image compression [63], which are highly suited to the human auditory and visual responses.

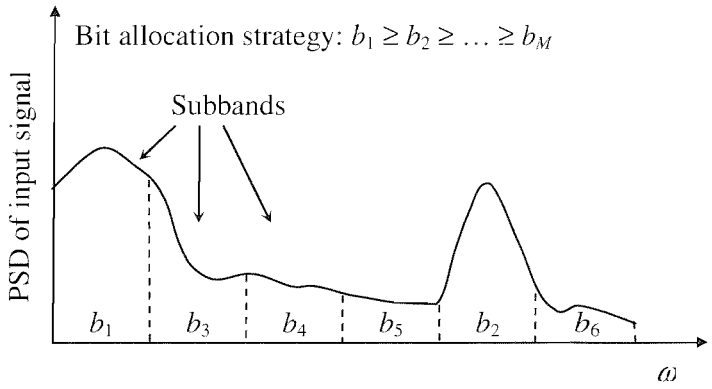


Figure 3.3: Example of a bit allocation strategy based on the subband variances for a six-band subband coder.

An *optimal subband coder* is the one that minimises the MSE given in (3.4) due to the subband quantisation [74]. In the case where the filter bank is orthonormal, it has been shown [73] that the reconstruction error has average variance $\sigma_\varepsilon^2 = (1/M)\sum_{k=1}^{M} \sigma_k^2$, called the quantisation error, where σ_k^2 is the quantisation noise variance due to the k^{th} quantiser. For a subband coding scheme that employs a PRFB, the reconstruction error becomes the quantisation error, i.e. $\varepsilon = \sigma_\varepsilon^2$. In this thesis, we are mainly concerned with the optimisation of orthonormal filter banks for subband coding. When the term 'optimal' is used to describe a system it is assumed that an *optimum* bit allocation strategy is employed.

3.2.1 Optimal Subband Coders with High Bit Rate Quantisers

The first theoretical results on the optimisation of subband coders were obtained for the transform coder [24,61]. In [24], Huang and Schultheiss show that the KLT is optimal for a wide class of signals under mild assumptions on the quantisation noise sources, particularly for arbitrary bit rates (see the next section). Here, the unitary matrix $\mathbf{E}$ is defined to be that given by the EVD of the covariance matrix for the demultiplexed signals.

The optimality problem is more complicated for subband coders than it is for transform coders because $\mathbf{E}(e^{j\omega})$ should be specified for all ω. Vaidyanathan [74] has shown that optimal subband coders can be constructed using unconstrained (ideal) orthonormal filter banks. This is under the assumption that the quantisers Q operate at high bit rates. Furthermore, the quantisation noise (error) process is assumed to be a uniformly distributed, white, WSS random process [73]. These assumptions allows one to model the quantiser noise with the simple model $\sigma_k^2 = dc_{kk0}2^{-2b_k}$, where d is a constant assumed to be the same for all subbands and c_{kk0} is the

variance of the signal $v_k(t)$ (output of $H_k(z)$), i.e. the zero-lag term of the true autocovariance function $c_{kk}(\tau) = \mathrm{E}[v_k(t)v_k(t-\tau)]$. Thus, the noise decays exponentially with number of bits b_k. It turns out that at high bit rates, optimality of orthonormal ideal filter banks is the same as the maximisation of a measure known as the coding gain [73,74]:

$$G = \frac{\frac{1}{M}\sum_{k=1}^{M} c_{kk0}}{\left(\prod_{k=1}^{M} c_{kk0}\right)^{1/M}}. \tag{3.7}$$

The coding gain is a signal dependent quantity whose value is determined by the input PSD and frequency response of the analysis filters. It can be interpreted as the ratio of the mean-square values of the direct quantisation error (e.g. round-off error in PCM) to that of ε in (3.4). The numerator of (3.7) represents the variance of the subband signals (arithmetic mean) when a direct quantisation scheme is used. This quantity is not affected by subband coding (i.e. it is a constant) since the total power in the subbands is preserved under the application of a lossless filter bank. The denominator is the geometric mean of the subband signal variances. If, for example, the variances of the subbands were the same, the geometric mean would be equal to the arithmetic mean, and therefore the coding performed would be identical to that of PCM.

The high bit rate assumption is generally unsatisfactory in practice. In [3,4], the optimality of orthonormal subband coders is proved under a more general quantiser model. The authors show that the optimum subband coder is the principal component filter bank; see section 3.2.3. Hence, in general, the coding gain does not represent the objective in the design of optimal subband coders.

A quantity that is closely related to the coding gain is the entropy: the average amount of information emitted from a source [51,62]. A measure of the efficiency of a source encoding method is obtained by comparing the entropy to the average number of binary digits per output symbol/character from the source.

3.2.2 Conditions for Coding Gain Optimality

From the expression of the coding gain in (3.7), it is clear that the minimisation of the denominator term (i.e. the product of the subband variances) leads to the maximisation of the coding gain. It is easy to see why this leads to optimal compression by expressing the denominator of (3.7) in terms of the number of bits required to encode the subband signals, thus

$$\log_2\left(\prod_{k=1}^{M} c_{kk0}\right) \equiv \sum_{k=1}^{M} \log_2(c_{kk0}). \tag{3.8}$$

So, the minimisation of the product of the subband variances results in the minimisation of the number of bits required in encoding each subband channel, which leads to optimally compressed data; the less uniform the energy distribution across the subband channels is, the fewer the bits required in quantising the signals. Vaidyanathan provides a set of necessary and sufficient conditions for the coding gain optimality of a paraunitary subband coder in [74]. The coding gain is maximised if and only if the output subband signals simultaneously satisfy *strong decorrelation* and *spectral majorisation*. It is also shown that, while each condition individually is only necessary, together they form a set of necessary and sufficient conditions.

Strong Decorrelation. If the subband signals $v_k(t)$ of Fig. 3.1 are decorrelated at all relative time lags then they are said to be strongly decorrelated, i.e.,

$$\mathrm{E}\left[v_k(t)v_m(t-\tau)\right] = 0, \text{ for } k \neq m \text{ and } \forall \tau . \tag{3.9}$$

Equivalently, the true PSD matrix of the vector signal $\mathbf{v}(t)$ is diagonal:

$$\mathbf{C}(e^{\mathrm{j}\omega}) = \sum_{\tau} \mathrm{E}\left[\mathbf{v}(t)\mathbf{v}^{\mathrm{H}}(t-\tau)\right]e^{-\mathrm{j}\omega\tau} = \mathrm{diag}\left[C_{11}(e^{\mathrm{j}\omega}), C_{22}(e^{\mathrm{j}\omega}), \ldots, C_{MM}(e^{\mathrm{j}\omega})\right] , \tag{3.10}$$

where $C_{kk}(e^{\mathrm{j}\omega})$ is the power spectrum of $v_k(t)$, i.e. the Fourier transform of the true autocorrelation sequence $c_{kk}(\tau)$ of the k^{th} subband signal.

Spectral Majorisation. Assume, without loss of generality, that the subbands are numbered such that $c_{kk0} \geq c_{(k+1)(k+1)0}$. The set of subband spectra $\left\{C_{kk}(e^{\mathrm{j}\omega})\right\}$ has the spectral majorisation property if

$$C_{11}(e^{\mathrm{j}\omega}) \geq C_{22}(e^{\mathrm{j}\omega}) \geq \ldots \geq C_{MM}(e^{\mathrm{j}\omega}), \quad \forall \omega . \tag{3.11}$$

An example of majorised subband power spectra is given for a three-channel filter bank in Fig. 3.4b. In Fig. 3.4a, we show subband channels that are not majorised; one signal dominates for some frequencies and the other signals for other frequencies.

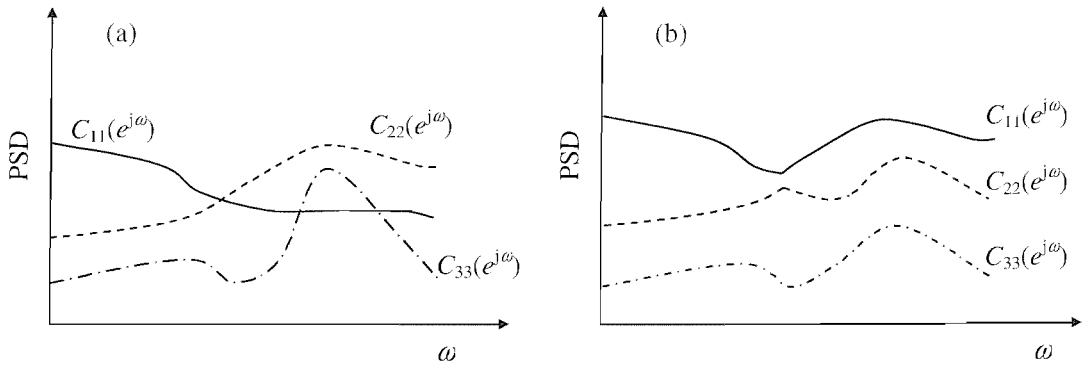


Figure 3.4: Example of subband signals that (a) are not spectrally majorised and (b) are spectrally majorised.

Optimal Compaction Filter

Consider the filter $H_k(e^{j\omega})$ of order N on the k^{th} branch of the M-channel filter bank in Fig. 3.1(a). The filter is said to be an *optimal energy compaction filter* (or optimal compaction filter) for the pair (M, N) if it is designed such that the variance of its output is maximised subject to the constraint that the *product filter*, $\left|H_k(e^{j\omega})\right|^2$, be Nyquist($M$) [74]. A measure of the energy compaction achieved by the filter is defined as

$$G_{comp}(M,N) = \frac{c_{kk0}}{a_0}, \tag{3.12}$$

where $a_0 = \mathrm{E}\left[\left|x(t)\right|^2\right]$ is the true variance of the input signal. The aim is to maximise the compaction gain under the Nyquist(M) assumption, which depends on the filter $H_k(e^{j\omega})$, the input PSD, and the integers N and M. The maximum (or ideal) compaction gain, $\hat{G}_{comp}$, is achieved if $H_k(e^{j\omega})$ is the optimal compaction filter with an *ideal* 'brickwall' frequency response for the input PSD. The ideal compaction gain is bounded thus $1 \le \hat{G}_{comp}(M,N) \le M$ [74]. By contrast, the coding gain in (3.7) has no such bounds. The design of optimal compaction filters has been of interest in recent years because of their known connection to optimal subband coding and the principal component representation of signals [2,16,68,70,74].

3.2.3 Principal Component Filter Bank

Consider the filter bank in Fig. 3.1. Suppose the first $m < M$ subband signals are retained without quantising and the others are discarded. Then there will be a corresponding distortion

(reconstruction error) between $x(t)$ and $y(t)$. The filter bank that minimises this distortion for each $m = 1,2,\ldots,M$ is called a *principal component filter bank* (PCFB). From this intuitive explanation one can see that the PCFB is an extension of PCA to polynomial matrices. We now review a mathematical result that will be useful in formally defining the PCFB.

Majorisation Theory. Let $\mathcal{A} = \{a_0, a_1, \ldots, a_{M-1}\}$ and $\mathcal{B} = \{b_0, b_1, \ldots, b_{M-1}\}$ be two sets of real numbers. The set $\mathcal{A}$ is said to *majorise* the set $\mathcal{B}$ if, after reordering such that $a_0 \geq a_1 \geq \ldots \geq a_n$ and $b_0 \geq b_1 \geq \ldots \geq b_n$,

1. $\sum_{k=0}^{M-1} a_k = \sum_{k=0}^{M-1} b_k$ and
2. $\sum_{k=1}^{n} a_k \geq \sum_{k=1}^{n} b_k$ holds for $0 \leq n \leq M-1$.

Thus every partial sum of set $\mathcal{A}$ is at least as large as the corresponding partial sum of set B.

Example. Let $\mathcal{A} = \{a_0, a_1, \ldots, a_{M-1}\}$ be any arbitrary set such that $\sum_{k=0}^{M-1} a_k = 1$. It is easy to see that $\left\{\frac{1}{M}, \ldots, \frac{1}{M}\right\} \leq \mathcal{A} \leq \{1,0,\ldots,0\}$. This suggests that majorisation may be used as a measure of the non-uniformity among the elements of a set.

A formal definition of the PCFB is as follows [3]: A PCFB is from a subclass C^p of orthonormal, uniform, maximally decimated filter banks. A filter bank $\mathbf{F}$ in the subclass C^p is said to be a PCFB for that class and for a given PSD if it has the set of subband variances that majorises every other set of subband variances (produced by other filter banks) in that class.

This definition broadly implies two conditions: Firstly, a PCFB maximises the partial sum

$$\sum_{k=1}^{n} c_{kk0} \tag{3.13}$$

for each $n \leq M$. In particular, when $n = 1$, then c_{110} is maximised by the choice of $H_1(z)$; that is, the first filter is an *optimal compaction filter* (maximises the compaction gain in (3.12))[1]. The second property is that a PCFB minimises the product

$$\prod_{k=1}^{M} c_{kk0}, \tag{3.14}$$

and therefore is an optimal orthonormal filter bank for subband coding, i.e. it maximises the coding gain in (3.7). In fact, the PCFB, if one exists, is optimal for subband coding for all bit

[1] Note that for a set of all possible paraunitary filter banks, if optimal energy compaction is achieved then the data is optimally compressed; this is in the sense of maximising the coding gain in (3.7). However, this would not necessarily be true for a set of arbitrary paraunitary matrices.

rates and for all bit allocation strategies, particularly for the optimal bit allocation [34]. Recently, the PCFB has been shown to be optimal for a broader class of objectives, such as noise reduction, [3,4].

Existence of a FIR PCFB

There are three subclasses of filter bank (special cases) for which the existence of a PCFB is assured. Consider the subclass of filter banks C^p. A PCFB exists for the subclass C^p in the following cases [3,4,33]:

i) if C^p is the **class of unconstrained (ideal) filter banks** C^u;

ii) if C^p is the **class of transform coders** C^t;

iii) if $M = 2$, i.e. the **classes of all two-channel filter banks**.

The problems of orthonormal filter bank optimisation and compaction filter optimisation are solved by a PCFB for these classes of filter bank only and for a given input signal PSD. In [33], it was shown that a PCFB cannot be guaranteed for the intermediate case where N is constrained (FIR filter banks) for arbitrary M, i.e. the **constrained class of filter banks** C^c. This is because the description of a PCFB is associated with an optimisation problem that has a multitude of objectives, which generally do not admit a common solution. In the recent past, suboptimal techniques have been proposed for this class, which basically try to find an approximate PCFB for a given PSD. A review of these methods is presented in section 3.4.

The PCFB has been an active area of research for some time [3,4,33,68,70,88]. The optimality of the PCFB was first independently observed in [68,70] for different objectives. In [68], Tsatsanis and Giannakis propounded the PCFB for multiresolution representation of signals using filters with a brick-wall (ideal) frequency-domain support. A strategy for designing a PCFB using ideal filters, introduced in [74], is detailed in the next section. In the case of the transform coder class, a PCFB is obtained with the SVD as explained in section 3.2.1. For the classes of two-channel filter banks the PCFB is found by designing an optimal compaction filter; then the other filter is determined trivially, as described in section 3.3.2. It turns out that the existence of a PCFB in these three cases is assured by their very construction.

3.3 Design of Optimal Orthonormal Filter Banks

A methodology for the design of optimal orthonormal filter banks for subband coding using filters among the class C^u is proposed by Vaidyanathan [74]. It turns out that the filter bank

obtained using this technique is a PCFB [34]. The methodology entails successively finding a set of M ideal optimal energy compaction filters for a given power spectrum. In effect, energy compaction is used as the objective function for the design of optimal filter banks [13,48,68]. In this section, we present a description of the design methodology. This is detailed because it is employed in a more 'practical' approach, known as the window method; an evaluation of this method is presented in section 3.4.2 and used to provide benchmark performances in chapter 6.

3.3.1 Compaction Filter Design with Unconstrained Filters

The procedure in designing an optimal compaction filter $H(e^{j\omega})$ is described here as in [74]. The following design is for the case where $H(e^{j\omega})$ is an ideal 'brickwall' filter with passband and stopband responses $\sqrt{M}$ and 0, respectively. Also, it is assumed that the filter is *antialias*(M): a filter whose output can be decimated without aliasing, that is, the shifted versions $H(e^{j[\omega-2\pi n/M]})$ do not overlap for distinct n in $0 \le n \le M-1$. The filter $H(e^{j\omega})$ can be constructed using the *ideal compaction filter design* technique:

1. Let $A(e^{j\omega})$ be the PSD of a WSS input signal. Divide the frequency axis of the spectrum into M equal (alias) regions (width of all passbands is $2\pi/M$). Thus, for each frequency ω_0 in $0 \le \omega < 2\pi/M$, define the M alias frequencies $\omega_n = \omega_0 + 2\pi n/M$.

2. Compare the values of $A(e^{j\omega_n})$ at these M alias frequencies $\{\omega_n\}$. Let ∂ be the smallest integer such that $A(e^{j\omega_\partial})$ is a maximum in this set. Then assign

$$H\{e^{j[\omega_0+(2\pi n/M)]}\} = \begin{cases} \sqrt{M}, & \text{when } n = \partial \\ 0, & \text{otherwise.} \end{cases} \tag{3.15}$$

3. The filter $H(e^{j\omega})$ can be completely defined for all ω in $0 \le \omega < 2\pi$ by repeating step 2 for each ω_0 in the region $0 \le \omega < 2\pi/M$. This filter satisfies the Nyquist(M) constrain, moreover, maximises its output variance under this constraint.

Example. Consider the construction of an optimal compaction filter $H(e^{j\omega})$ for an input PSD, $A(e^{j\omega})$, as in Fig. 3.5(a), and let $M = 3$. Firstly, the frequency axis is divided into 3 equal regions labelled 0, 1, and 2. Secondly, in the spectral band $4\pi/3 \le \omega_0 < 14\pi/9$, $A(e^{j\omega_0})$ dominates the spectrum at the alias frequencies $0 \le \omega_0 < 2\pi/9$ and $2\pi/3 \le \omega_0 < 8\pi/9$ in regions 0 and 1, respectively. Next, for the band $2\pi/9 \le \omega_0 < 4\pi/9$, $A(e^{j\omega_0})$ dominates the PSD at the alias frequencies $8\pi/9 \le \omega_0 < 10\pi/9$ and $14\pi/9 \le \omega_0 < 16\pi/9$ in regions 1 and 2,

respectively. Finally, for the frequencies $10\pi/9 \le \omega_o < 4\pi/3$, $A(e^{j\omega_o})$ dominates the PSD at the alias frequencies $4\pi/9 \le \omega_o < 2\pi/3$ and $16\pi/9 \le \omega_o < 18\pi/9$ in regions 0 and 2, respectively. The optimal compaction filter designed is shown in Fig. 3.5(b).

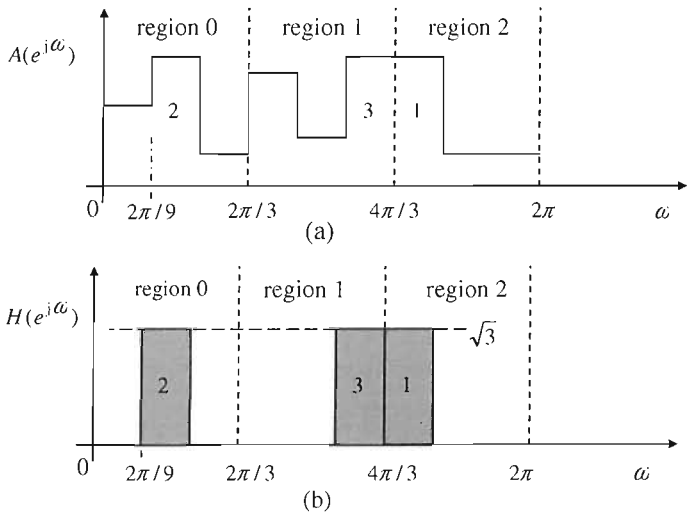


Figure 3.5: (a) input signal PSD and (b) optimal compaction filter for this signal for $M = 3$.

3.3.2 Optimal Filter Bank Design with Unconstrained Filters

We are now equipped to design an optimal orthonormal subband coder $\{H(e^{j\omega})\}$. The procedure for the construction of the analysis bank $\{H_k(e^{j\omega})\}$ is described through an example. The filter bank may be found by successive application of the ideal compaction filter design technique described in the previous section. Consider the design of a three-channel optimal orthonormal subband coder for the input power spectrum presented in Fig. 3.6(a). The first step is to design an optimal compaction filter $H_1(e^{j\omega})$ for the given input PSD, $A(e^{j\omega})$, using the procedure described above. The frequency response of this analysis filter is shown in Fig. 3.6(b). Let the passband support of $H_1(e^{j\omega})$ be denoted as ς_1. A new 'partial' power spectrum is defined:

$$C^{(1)}(e^{j\omega}) = \begin{cases} 0, & \text{in } \varsigma_1 \\ A(e^{j\omega}), & \text{otherwise.} \end{cases} \tag{3.16}$$

as given in Fig. 3.6(c). Thus, the partial power spectrum $C^{(1)}(e^{j\omega})$ is defined by 'peeling off' the portion of $A(e^{j\omega})$ falling in the passband of $H_1(e^{j\omega})$.

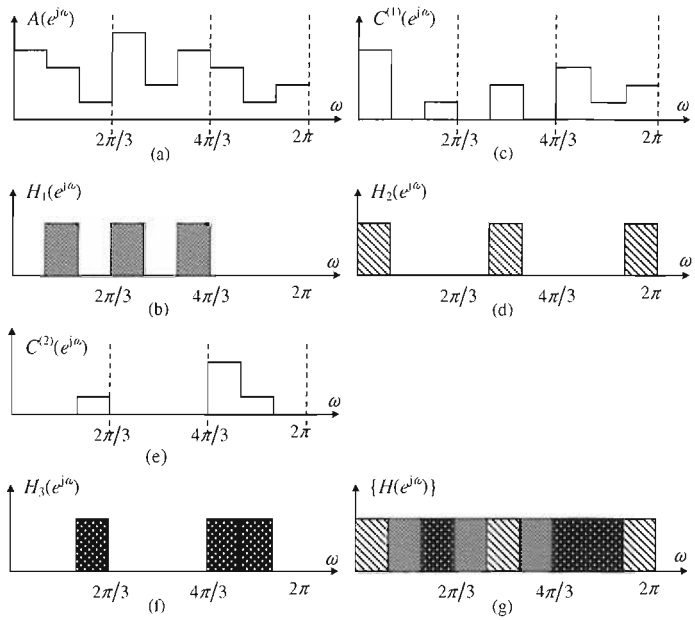


Figure 3.6: A graphical description of the ideal design procedure $M = 3$. (a) Input signal PSD and (b) corresponding optimal compaction filter magnitude-frequency response. (c)-(f) Partial power spectra and corresponding optimal compaction filters. (g) Frequency responses of the three optimal analysis filters.

The next step is to construct an optimal energy compaction filter $H_2(e^{j\omega})$ for $C^{(1)}(e^{j\omega})$. The magnitude-frequency response for $H_2(e^{j\omega})$ is shown in Fig. 3.6(d). The next partial PSD $C^{(2)}(e^{j\omega})$, shown in Fig. 3.6(e), is obtained by peeling off the portions of $C^{(1)}(e^{j\omega})$ falling on the passband of $H_2(e^{j\omega})$. Finally, $H_3(e^{j\omega})$ (Fig. 3.6(f)) is constructed as the optimal compaction filter for $C^{(2)}(e^{j\omega})$. The portions of $C^{(2)}(e^{j\omega})$ falling on the passband of $H_3(e^{j\omega})$ are then removed to leave a partial spectrum with, notionally, no energy. Hence, the analysis filters of an optimal orthonormal filter bank for the given PSD have been identified, as presented in Fig. 3.6(g).

Since the filters have non-overlapping frequency spectra strong decorrelation is satisfied, moreover, spectral majorisation is also satisfied. Therefore, it follows that the filter bank constructed using this algorithm maximises the coding gain in (3.7). It has been shown that this design procedure yields a PCFB [34,74].

Two-channel Optimal Filter Bank Design

It is easy to see how the PCFB for the class of two-channel filter banks is constructed: Consider the ideal compaction filter design procedure given in section 3.3.1, the coding gain G, and the

filter bank in Fig. 3.1 for $M = 2$. Let the input variance of $x(t)$ be unity, i.e. $a_0 = 1$. The output variances of $H_1(z)$ and $H_2(z)$ are denoted by c_{110} and c_{220}, respectively. Orthonormality implies the equality $c_{110} + c_{220} = 2$, which is the upper bound on the compaction gain (G_{comp}) given in (3.12). For the first channel, $G_{comp} = c_{110}/a_0 = c_{110}$. The compaction gain of the second channel is $c_{220}/a_0 = c_{220} = 2 - c_{110} = 2 - G_{comp}$. Thus, the coding gain becomes

$$G(2) = \frac{\sigma_x^2}{(c_{110}c_{220})^{1/2}} = \frac{1}{(c_{110}c_{220})^{1/2}} = \frac{1}{\left(G_{comp}\left[2 - G_{comp}\right]\right)^{1/2}}. \tag{3.17}$$

Since the sum $c_{110} + c_{220}$ is constant, the only variable is $c_{110} = G_{comp}$. The filter bank is found by designing the first filter $H_1(z)$ to be an optimal compaction filter such that it minimises the denominator (i.e. maximises c_{110}).

The choice of $H_1(z)$ is subject to the orthonormality of the filter bank and is equivalent to the Nyquist(2) constraint, i.e., $\left|H_1(e^{j\omega})\right|^2 + \left|H_1(-e^{j\omega})\right|^2 = 2$. Hence, the other filter is determined from the first such that the two filters form an orthonormal set. Assuming $H_1(z)$ has order N, the other filter may be obtained by the well-known relationship [73]

$$H_2(z) = z^{-N}\tilde{H}_1(-z). \tag{3.18}$$

In the real-coefficients case (3.18) means $\left|H_2(e^{j\omega})\right| = \left|H_1(e^{j(\omega-\pi)})\right|$. Therefore, optimising one of the analysis filters for maximum energy compaction is equivalent to optimising a two-channel orthonormal filter bank according to input statistics. It follows (by construction) that there is a PCFB for all classes of two-channel filter banks.

3.4 Review of Suboptimal Filter Bank Design Algorithms

As already discussed, the design and optimisation of filter banks has been solved for the unconstrained class, transform coder class, and the classes of all two-channel filter banks. In these cases, the PCFB is the solution. However, in general, the PCFB does not exist for the class of FIR filter banks. A number of authors have proposed suboptimal techniques that achieve good approximations of the optimum coding gain [13,16,29,32,48,56,69,77,88]. Other authors have presented paraunitary filter bank design in the context of subspace analysis [17,37,44,49,57]; this topic is treated in chapter 4. A review of conventional design methods is presented here, and their relative merits and short-falls are investigated. This is done with the

goal of selecting one as a benchmark algorithm for the SBR2 coder, introduced later in section 6.2 as a method for subband coder design.

A type of transform that is relevant to the design of paraunitary FIR filter banks is the lapped orthogonal transform (LOT) [42]. Although studied independently in the past, the LOT and the uniform maximally decimated paraunitary FIR filter bank are equivalent [43]. The difference between the two types of filter bank is in the way they are designed and implemented. Algorithms for LOT-based filter bank design are not considered in this thesis.

3.4.1 Lattice Parameterisation

A paraunitary filter bank can be constructed by cascading L blocks in an M-channel lattice structure [73]. Each block in the lattice has a parameter matrix defined by rotation angles, which are optimised subject to a constraint (or constraints). An approach that is found predominantly in the literature involves the optimisation of the parameter matrices subject to two constraints: (i) the decorrelation of the signals and (ii) the imposition of the paraunitary condition. These techniques adapt the lattice parameters according to the received signal samples iteratively. Two optimisation methods for this type of architecture are given in the following.

Gradient Descent

A method that is commonly found in the literature is one that uses an adaptive algorithm based on the stochastic gradient descent/ascent [16,29]. Each block (matrix of rotation angles) is optimised iteratively: parameter values are optimised as the data samples are input, i.e. online. In [29], the authors show that these algorithms can be made to be more computationally efficient. However, they are highly nonlinear and suffer from the usual problem of convergence to local minima associated with nonlinear optimisation.

Eigenstructure

The other type of optimisation scheme available for lattice optimisation is eigenstructure algorithms [25,56]. Eigenstructure algorithms can avoid converging to local minima, which is an advantage over the gradient-based algorithms. These techniques involve the explicit decomposition of the covariance matrix with the aim that dominant and weak eigenvalues and associated eigenvectors are identified. It is possible to use a suitably parameterised two-channel lattice structure as a relatively straightforward way of generating an FIR paraunitary filter bank. That is, one that is guaranteed to be paraunitary irrespective of the parameter values. Paraunitary matrices may be cascaded to form matrices that satisfy the paraunitary condition in (2.27). A very good example is given by Vaidyanathan in [73], as stated in the following: every FIR paraunitary matrix can, up to some permutation of the inputs, be represented as the product of a

sequence of Jacobi rotations interspersed by pure time-delay matrices. In specific terms, a 2×2 paraunitary matrix may be decomposed as a sequence of paraunitary stages:

$$\mathbf{H}(z) = \mathbf{Q}_N \mathbf{\Lambda}(z) \ldots \mathbf{\Lambda}(z) \mathbf{Q}_1 \mathbf{\Lambda}(z) \mathbf{Q}_0 , \tag{3.19}$$

where $\mathbf{\Lambda}(z)\mathbf{Q}_n$ constitutes a single stage, $\mathbf{Q}_n \in \mathbb{R}^{2\times2}$ is a Jacobi rotation (unitary) matrix parameterised by a rotation angle θ_n in (2.15), i.e.

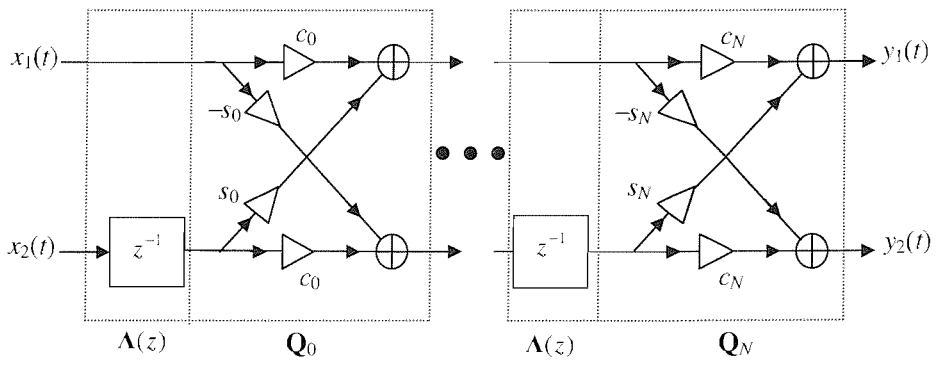
$$\mathbf{Q}_n = \begin{pmatrix} \cos\theta_n & \sin\theta_n \\ -\sin\theta_n & \cos\theta_n \end{pmatrix}, \tag{3.20}$$

and $\mathbf{\Lambda}(z) \in \mathbb{R}^{2\times2}(z)$ is a pure time-delay (paraunitary) matrix and of the form

$$\mathbf{\Lambda}(z) = \begin{pmatrix} 1 & 0 \\ 0 & z^{-1} \end{pmatrix}. \tag{3.21}$$

In Fig. 3.7, we show a flow diagram of the decomposition in (3.19).

Recalling from section 2.3.2 that the degree of a polynomial matrix is the minimum number of delay elements required to implement it, we note that the degree (and order) of $\mathbf{H}(z)$ is N. Also note that in the degenerate case of degree zero, $\mathbf{H}(z)$ takes the form of a single unitary matrix as required for the EVD of a conventional Hermitian matrix. Even if the degree N could be established in advance there are a few problems with using the lattice filter of (3.19) to construct a paraunitary matrix. These are demonstrated with the help of an example.



where $c_n = \cos\theta_n$, $s_n = \sin\theta_n$ and $\theta_n \in \mathbb{R}$

Figure 3.7: A 2×2 paraunitary lattice filter consisting of N paraunitary blocks.

Example. Consider that the optimal paraunitary filter bank to be found is a simple one given by:

$$\mathbf{H}(z)=\begin{pmatrix}\cos\theta & \sin\theta \\ -\sin\theta & z^{-16}\cos\theta\end{pmatrix}, \tag{3.22}$$

which has degree and order 16. Suppose that the parameters are determined in a sequential manner. Then the following problems may exist:-

- The construction of (3.22) would involve the parameterisation of a 16 stage lattice filter requiring the estimation of 16 parameters (rotations). This is even though only a single parameter is needed. The estimation of the other 15 rotation angles is likely to result in the design of a suboptimal filter bank.
- The correct estimation of the final parameter could be very difficult since it is likely that the effect of the first 15 rotations on the data is to increase the sample noise. This problem would be exacerbated by the fact that it is impossible for later rotations to undo the effects of preceding rotations.

In order to overcome these problems, the optimisation of the multivariable lattice filter would need to be nonlinear, and therefore complex and computationally costly. If the rotation angles θ_n could be computed independently, then these problems would be alleviated. This kind of approach is adopted by the SBR2 algorithm in constructing paraunitary matrices, as described in section 5.

Regalia and Huang [56] have proposed an eigenstructure-based signal-adapted algorithm for the fixed degree parameterisation of a two-channel lossless filter bank described by (3.19). The difficult nonlinear optimisation is re-formulated using a state space approach that lends itself to an iterative solution. The algorithm optimises the rotation matrices such that the energy in the second output channel is minimised, i.e., the cost function used is $E[y_2^2(t)]$. This is achieved by seeking to make $E[y_2^2(t)]$ the smallest eigenvalue of the covariance matrix. The filter bank produced by this method is assured to be paraunitary by its very construction. An inherent advantage of this technique over other eigenstructure techniques is that the cost function is quadratic in the rotation angles, which makes for a computationally simpler approach. The method avoids the problems of local minima associated with gradient descent techniques. A crucial disadvantage of this technique is that it is limited to the design of two-channel filter banks; an extension to the $M > 2$ case is not considered by the authors.

3.4.2 Linear Programming and the Window Method

Linear Programming

Another approach to filter bank construction consists of designing signal-adapted filters using energy compaction as the adaptation criterion [13,48]: for a given power spectrum, the components of an orthonormal filter bank are designed so that the energy in the first channel is maximised; the energy in each of the remaining channels is maximised successively. The energy compaction problem is different to the coding gain problem for constrained-order filters; this is explained in terms of the PCFB in section 3.2.3. In the case of FIR filters, there are spare degrees-of-freedom after the first stage for maximising the energy in the rest of the channels.

In [48], Moulin *et al* propose the use of linear programming for the design of optimal paraunitary filter banks. An overview of the algorithm is provided in the following. It is well known that the polyphase analysis bank (Fig. 3.1(b)) can be factorised as [73]:

$$\mathbf{E}(z) = \mathbf{U}\mathbf{Q}_1(z)\mathbf{Q}_2(z)\ldots\mathbf{Q}_N(z) \in \mathbb{C}^{M\times M}(z)\,, \tag{3.23}$$

where N is the order and degree of $\mathbf{E}(z)$; $\mathbf{U}$ is a unitary matrix (possibly KLT or discrete cosine transform (DCT)); $\mathbf{Q}_n(z) = \mathbf{I}_M - \mathbf{q}_n\mathbf{q}_n^{\mathrm{T}} + z^{-1}\mathbf{q}_n\mathbf{q}_n^{\mathrm{T}}$ are similar in form to Householder matrices; and $\mathbf{q}_n$ are unit-norm real valued vectors with $1 \le n \le N$. The polyphase matrix is parameterised by the rows, $\{\mathbf{q}_n\}$ and $\{\mathbf{u}_n\}$, of $\mathbf{Q}_n(z)$ and $\mathbf{U}$, respectively. The variance of the first subband signal c_{110} is a linear function of the product filter coefficients $g_1(t)$ of $G_1(e^{\mathrm{j}\omega}) = \left|H_1(e^{\mathrm{j}\omega})\right|^2$. It is maximised subject to the constraints that $G_1(e^{\mathrm{j}\omega})$ is Nyquist(M) (trivially achieved) and that $G_1(e^{\mathrm{j}\omega}) \ge 0$ for all ω. The latter is written as a linear inequality for each ω in term of $g_1(t)$. Hence, the maximisation problem is a linear programming one with finitely many variables and infinitely many inequality constraints, which yields a linear semi-infinite problem. The second part of the method consists of performing an eigenvector decomposition of the $M \times M$ correlation matrix for the signals transformed by $\mathbf{Q}(z) = \sum_n \mathbf{Q}_n(z)$. The optimal filters are spectral factors of the product filter. This method produces filter banks that are paraunitary. The compaction gain achieved by its filters asymptotes to the optimal one as N increases. An improvement to the method is proposed in [32], which involves windowing the linear programming solution.

Window Method

The window method, proposed by in [32], is based on a common practice in FIR filter design: approximate an ideal filter response by windowing its impulse response. The basic idea behind

the technique is to construct the product filter $G_1(e^{j\omega}) = |H_1(e^{j\omega})|^2$ by windowing a periodic function. The periodic function is determined by applying the ideal compaction filter design algorithm, as described in section 3.3.1, at P uniform DFT frequencies. Hence, there are a finite number of comparisons to be made on a finite frequency grid.

In formulating the above idea, the impulse response of the product filter, $g(\iota)$, is

$$g(\iota) = \varpi(\iota) f_P(\iota), \tag{3.24}$$

where $\varpi(\iota)$ is a window function of the same length as $g(\iota)$ with a non-negative DFT and $f_P(\iota)$ is a periodic sequence with period $P = \Phi M > 2N$. Here, Φ is the number of design (comparison) frequencies and N is the order of $g(\iota)$ (and hence the resulting compaction filter). Spectral factorisation [30] is performed on $g(\iota)$ to extract the filter $h(\iota)$. Given a positive semidefinite sample correlation sequence $r(\iota)$, $\{\iota \in \mathbb{Z} \,|\, 0 \le \iota \le N\}$, the expression in (3.24) is implemented:

1. Compute the PSD of the input signal, $R_P(\phi)$, i.e., the P-point DFT coefficients of conjugate symmetric sequence $\hat{r}(\iota) = \varpi_r^*(\iota) r(\iota)$.
2. Design an ideal compaction filter at each frequency $\{\phi \in \mathbb{Z} \,|\, 0 \le \phi \le \Phi\}$. That is, determine the index m_0 for which $R_P(\phi + m_0\Phi)$ is maximum, and make the assignments $F_P(\phi + m_0\Phi) = M$ and $F_P(k + m_n\Phi) = 0$, $n = 1, \ldots, M$.
3. Compute $f_P(\iota)$: the inverse DFT (IDFT) of $F_P(k)$.
4. Determine the optimal window $\varpi(\iota)$ and form the product filter $g(\iota) = \varpi(\iota) f_P(\iota)$.
5. Find $h(\iota)$ by spectrally factorising $g(\iota)$.

The window method produces compaction filters that are Nyquist(M) as long as Φ is a multiple of M and greater than the filter order N. The suboptimality of its filters diminishes as N increases. Note that an appropriate value of N must be chosen prior to using the above algorithm. The window method is an offline technique and does not require optimisation tools or iterative methods. The number of design frequencies to use depends on the autocorrelation function of the input process, which in turn influences the choice N.

The two-channel orthonormal filter bank can be easily obtained from the compaction filter designed by the window method; therefore, orthonormality is guaranteed by construction. The originators of the window method recommend a possible strategy for the design of M-channel orthonormal subband coders in [32]. There suggested scheme involves using the procedure proposed in [48] to construct a multichannel filter bank; the compaction filter generated by the

window method is used as the basis for the remaining filters of an M-channel orthonormal filter bank. An evaluation of the M-channel strategy will not be presented in this thesis.

Example. Let the input, $x(t)$, be an order $N_g = 5$ autoregressive moving-average (or ARMA(5)) process with a multiband spectrum, $A(e^{j\omega})$, as shown by the dashed curve in Fig 3.8(a). The ARMA(5) process was implemented using a Yule-Walker IIR filter with coefficients

$$\frac{B_x(z)}{A_x(z)} = \frac{0.6903 - 0.0160z^{-1} - 0.1453z^{-2} + 0.3302z^{-3} - 0.5426z^{-4} - 0.3141z^{-5}}{0.6867 - 0.4363z^{-1} + 0.1255z^{-2} - 0.3162z^{-3} + 0.4688z^{-4} - 0.0516z^{-5}}. \qquad (3.25)$$

We have chosen this PSD based on examples given in [32]. The design of the two-channel analysis bank using the window method involved the following: The window method was implemented directly from [32] and used to construct a compaction filter $H_1(z)$ for $x(t)$. A cepstral-FFT based spectral factorisation algorithm [30] was used to complete the compaction filter construction. The filter designed in this way is both stable and causal [18,26], i.e. it is non-minimum phase, as can be ascertained from the locations of the poles and zeros of $H_1(z)$ shown in Fig. 3.9(a). The second filter, $H_2(z)$, is simply determined from the first by time-reversing the impulse response of $h_1(t)$ and alternately changing the sign of the time-reversed filter coefficients – see (3.18).

For $x(t)$ and $N = 65$, the 2 analysis filters produced using the window method have the frequency responses shown in Fig. 3.8(a). The compaction gain (obtained from the true statistics of the signal) for this filter indicated in the paper was $G_{comp} \approx 1.86$ and that obtained from our experiment is 1.84. The disparity between the results is likely to be due to two minor differences between the experiments; viz., the type of spectral factorisation algorithm used and the input signal. These results helped in validating our implementation of the window method: results obtained are very similar to those in [32].

The window method has designed a multiband compaction filter with passbands that coincide with dominant signal frequency components, as is clear from Fig. 3.8(a). This is indicative of a high compaction gain since the filter is accepting most of the signal energy (into the first channel). In Fig. 3.8(b), we show the true PSD of the subband channels, $A_k(e^{j\omega})$. It is obvious from this graph that the window method has performed spectral majorisation.

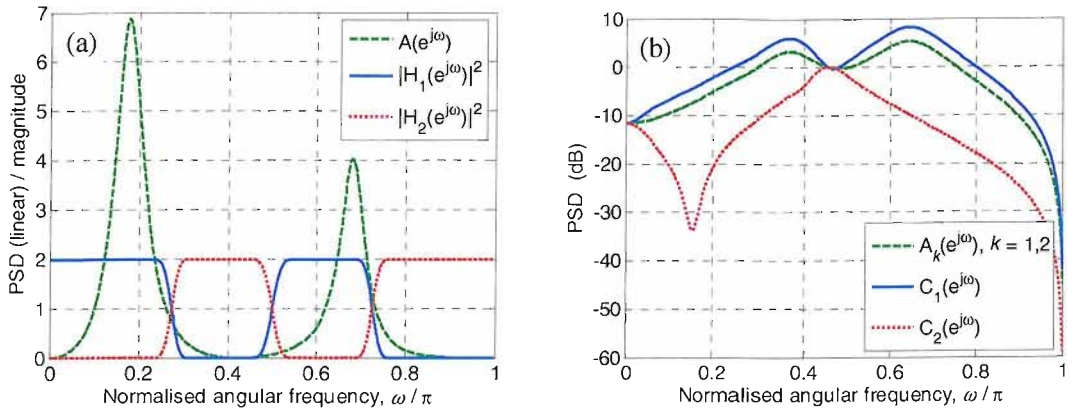


Figure 3.8: (a) PSD of an ARMA(5) process and the magnitude-frequency response of the filters designed using the window method for N = 65 [32]. (b) PSD of the input and subband channels.

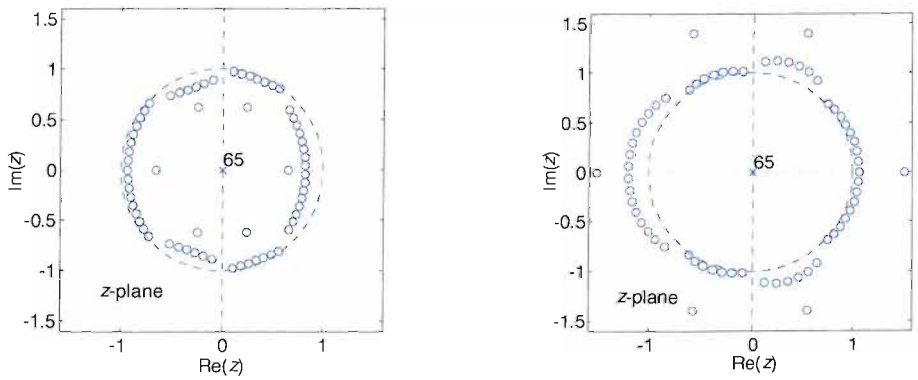


Figure 3.9: Analysis of the FIR filter designed by the window method for $N = 65$: poles and zeros of (a) $H_1(z)$ and (b) $H_2(z)$.

3.4.3 Comparison of Suboptimal Design Algorithms

We have seen that the linear programming, eigenstructure, and gradient descent based algorithms adapt to each input signal sample, i.e. they are on-line techniques. By contrast, the window method is a non-adaptive technique, so it uses all the samples in a data set to compute its filters instead of adapting to each data sample. Both the adaptive and non-adaptive methods can produce filters that achieve near optimal compaction gains; it follows that the filter banks obtained can perform strong decorrelation and spectral majorisation. The main advantage of

adaptive algorithms is that they can usually yield high coding gains with very short filters. Therefore, the resources required in storing or conveying the filter impulse responses would be comparatively low. However, it is often the case that the coding gains attained by the window method are better than those attained by adaptive algorithms for longer filters. Also, a drawback of the adaptive algorithms is that they are highly complex and computationally expensive as a consequence of their adaptive nature. The complexity of the window method is greatly simplified because it involves the mere comparison of some frequency bins and windowing, and essentially solves a deterministic problem. This makes it much more computationally efficient than the adaptive algorithms. For the reasons just cited we have chosen the performance and efficiency of the window method as benchmarks in assessing the subband coder design method proposed in chapter 6.

There is a level of a priori knowledge required for successful application of the subband coder design methods discussed in this section. The input is assumed to originate from a WSS process [74]. In the case of a WSS input process, the special form of the polynomial covariance matrix of the subband signals is exploited by these methods, as will be discussed in section 6.2.

3.5 Concluding Remarks

In this chapter, we have presented a study of filter bank theory and a survey of practical methods for the design of filter banks. The solution to the energy compaction problem is the same as that of the coding gain problem when ideal filters are used: the PCFB designs an optimal orthonormal subband coder which minimises the reconstruction error and whose first filter provides optimal energy compaction. However, the PCFB is only guaranteed to exist for certain classes of filter bank; it has been shown not to exist, in generally, for FIR filters.

We have described various suboptimal approaches for the design of (practical) FIR paraunitary filter banks and made performance comparisons based on the literature. The suboptimality of these techniques does not seem to be significant for practical signals. This review has revealed that the window method has certain salient advantages over the other state-of-the-art algorithms. Most importantly, it can construct near-optimal compaction filters at low computational cost. The window method is therefore a good technique to use as a benchmark in evaluating an algorithm for the design of subband coders, introduced later in this thesis.

4 Polynomial Matrix Eigenvalue Decompositions

In this chapter, we extend some of the most important matrix computations required for subspace estimation, which were introduced in chapter 2, from conventional matrices to the relatively unexplored domain of polynomial matrices. The properties of the resultant decomposition of polynomial matrices are observed to be those that are satisfied by PCFBs discussed in chapter 3. To begin with, the problem of diagonalising a polynomial matrix is formulated in section 4.1. This makes apparent the requirements of a polynomial matrix decomposition and serves to demonstrate the fundamental limitations of the EVD/SVD for the problem. Generalisations of the EVD and SVD for scalar matrices to parahermitian polynomial matrices are defined and their limitations are discussed in section 4.2. We show that a polynomial matrix EVD (PEVD) satisfies properties that are generalisations of those satisfied by the EVD, particularly energy preservation, energy compaction, strong decorrelation and spectral majorisation. These generalised properties are satisfied by optimal orthonormal filter banks for subbands in section 3.2. Two natural performance measures are introduced that quantify the diagonalisation performance and broadband subspace decomposition accuracy of a PEVD technique. Energy compaction and data encoding are recognised as possible tasks for an algorithm that computes a PEVD. In section 4.3, a brief review of existing algorithms for broadband signal subspace estimation is given along with some assessment of their performances. A family of frequency-domain approaches to PEVD, known as the bandwise algorithm, is introduced and evaluated in section 4.4. They rely on well-known methods for classical spectral estimation and contribute to the novel work in this thesis. The performances of these methods are assessed.

4.1 Motivation

Recall from section 2.2 that for a set of correlated narrowband signals it is possible to use the SVD or EVD to find a unitary matrix that diagonalises the (Hermitian) covariance matrix of the signals. These decompositions also order the eigenvalues of the Hermitian matrix in terms of their power. This gives a good estimate of the signal and noise subspaces and is a way of solving many problems, including signal detection, noise reduction, signal separation and data compression [22,72]. In sensor array signal processing, the signal subspace usually represents

the signals of interest. For data compression applications, the signal subspace contains the bulk of the information in the data whilst the noise subspace represents redundancy and noise.

In recent years the need for techniques that can perform subspace decomposition for broadband signals and convolutively mixed signals has grown [9,11,17,27,37,44,49, 52,57,79,81,84]. Instantaneous decorrelation is not sufficient for accurate signal subspace estimation of these types of signals. This is because the signals are correlated not just at a single relative time-lag, but usually at many time-lags. Therefore, decorrelation at all relative time-lags is required, i.e. we need to perform strong decorrelation. In addition, we require that the set of transformed signals have the spectral majorisation property. These properties were discussed in the context of filter bank design for subband coding in section 3.2, but, as will be shown in this chapter, they can be imposed on arbitrary signals as well.

In terms of matrices, the entries of the corresponding covariance matrix for the correlated signals are not scalars, but polynomials. We require a transformation (filter bank) for polynomial matrices that satisfies properties that are generalisations of those satisfied by the EVD: the diagonalisation of the parahermitian matrix; the fixed ordering of its diagonals (polynomial eigenvalues) in terms of decreasing norm; and conservation of the total energy of the signals, i.e. paraunitarity. In this chapter, we show that strongly decorrelating a set of input sequences is equivalent to diagonalising the corresponding polynomial covariance matrix. The satisfaction of this property and spectral majorisation by the outputs from a multichannel filter bank would allow for accurate estimation of the broadband signal and noise subspaces.

It is worth noting that the paraunitary filter bank design methods introduced in this chapter are intended for application to arbitrary multichannel data. By contrast, the filter bank design algorithms for subband coding presented in section 3.4 are based around the assumption that the input signal is WSS. This is a fundamental difference between the subband coder design methods and the techniques presented in this chapter.

4.1.1 Convolutive Mixing Problem

Consider the linear convolutive mixing exemplar in Fig. 4.1. This type of mixing is encountered in numerous signal processing applications. For illustrative purposes, we shall consider this problem in the context of sensor array signal processing. The diagram shown depicts a model for the propagation of K source signals, $w_k(t)$, emitted from separate transmit antennas and received by an antenna array of M sensors. The received (sensor) signals, $x_k(t)$, are then processed, which usually involves beamforming: for background on sensor arrays and beamforming, refer to [23]. If the sensor signals are narrowband, then the relative delay between different propagation paths can be represented in terms of different instantaneous phase and

amplitude factors imposed at the sensor elements. Under these circumstances, the propagation of the sources to sensors may be represented as a scalar (mixing) matrix (i.e. the instantaneous mixture model given by (2.6)). However, the instantaneous mixture model is not, in general, a sufficient model for the case of broadband source signals or for the situation where narrowband signals have been convolutively mixed (as shown in the figure). Instead, the mixing must be represented as a linear superposition of delayed samples of the signals emitted by each source.

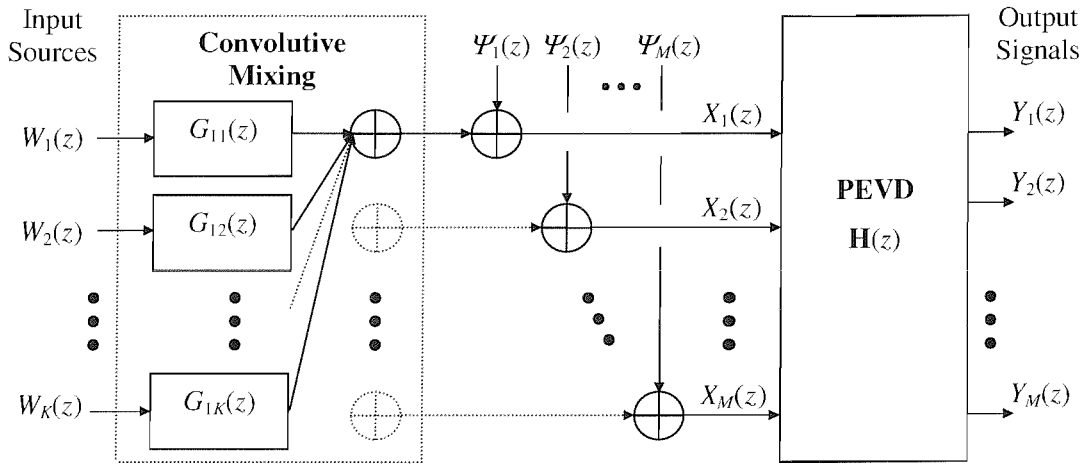


Figure 4.1: Block diagram illustration of the linear convolutive mixing problem followed by *second-order* blind signal separation: the source signals are mixed to produce data which is then transformed into decorrelated outputs.

The input to output channels of the mixing process are described by different unknown FIR filters $g_{mk}(t)$ $(k = 1,2,\ldots,K;\ m = 1,2,\ldots,M)$ that model the effects of multipath propagation and dispersion: commonly referred to as convolutive mixing. The mixing matrix is thus a $M \times K$ matrix of FIR filters, not a scalar matrix. Suppose the K source signals $w_k(t)$ are statistically independent and identically distributed (i.i.d.) with zero mean and unit variance and let $M \geq K$. The m^{th} received signal at time t is:

$$x_{mt} = \left(\sum_{k=1}^{K} g_{mk} * w_k\right)_t + \psi_{mt}, \tag{4.1}$$

The sequences $\psi_m(t)$ represent the system noise on the m^{th} receive channel (e.g. the thermal noise due to the antenna array and electronics), the samples of which are drawn from an independent white Gaussian noise process with variance σ_ψ^2. We can express the sequences $w_k(t)$, $\psi_m(t)$ and $x_m(t)$ as algebraic power series, respectively:

$$W_k(z) = \ldots w_k(-t)z^{t} + \ldots + w_k(0) + w_k(1)z^{-1} + \ldots + w_k(t)z^{-t} \ldots,$$

$$\Psi_m(z) = \ldots \psi_m(-t)z^{t} + \ldots + \psi_m(0) + \psi_m(1)z^{-1} + \ldots + \psi_m(t)z^{-t} \ldots \text{ and}$$

$$X_m(z) = \ldots x_m(-t)z^{t} + \ldots + x_m(0) + x_m(1)z^{-1} + \ldots + x_m(t)z^{-t} \ldots.$$

The function $g_{mk}(t)$ can be written as a Laurent polynomial in the indeterminate variable z^{-1}: $G_{mk}(z) = g_{mk}(-t)z^{t} + \ldots + g_{mk}(0) + g_{mk}(1)z^{-1} + \ldots + g_{mk}(t)z^{-t}$. The convolution in (4.1) is expressed as the product $X_m(z) = G_{mk}(z)W_k(z)$ – see section 2.3.1. The mixing problem can now be expressed in the form:

$$\begin{pmatrix} X_1(z) \\ X_2(z) \\ \vdots \\ X_M(z) \end{pmatrix} = \begin{pmatrix} G_{11}(z) & G_{12}(z) & \cdots & G_{1K}(z) \\ G_{21}(z) & G_{22}(z) & \cdots & G_{2K}(z) \\ \vdots & \vdots & \ddots & \vdots \\ G_{M1}(z) & G_{M2}(z) & \cdots & G_{MK}(z) \end{pmatrix} \begin{pmatrix} W_1(z) \\ W_2(z) \\ \vdots \\ W_K(z) \end{pmatrix} + \begin{pmatrix} \Psi_1(z) \\ \Psi_2(z) \\ \vdots \\ \Psi_M(z) \end{pmatrix}. \tag{4.2}$$

This expression may be more compactly written using polynomial matrix notation:

$$\underline{X}(z) = \mathbf{G}(z)\underline{W}(z) + \underline{\Psi}(z) \in \mathbb{C}^M(z), \tag{4.3}$$

where $\mathbf{G}(z)$ is a $M \times K$ polynomial matrix that represents the convolutive mixing process and $\underline{W}(z)$, $\underline{\Psi}(z)$ and $\underline{X}(z)$ are vectors of algebraic power series representing the sources, noise and received signals, respectively.

Note that if we had knowledge of the mixing $\mathbf{G}(z)$ and σ_ψ^2 was very small, it would be possible to get a (least-squares) solution to this problem by applying the pseudo-inverse of $\mathbf{G}(z)$ to $\underline{X}(z)$ [20], in other words

$$\underline{W}(z) \approx \left(\tilde{\mathbf{G}}(z)\mathbf{G}(z)\right)^{-1} \tilde{\mathbf{G}}(z)\underline{X}(z) = \Gamma(z)\underline{X}(z), \tag{4.4}$$

where $\Gamma(z)$ is sometimes called the *unmixing matrix*[1]. This is provided $M \geq K$ (i.e., $\mathbf{G}(z)$ is a fully-determined or overdetermined system) and the rank of $\mathbf{G}(z)$ is equal to K. In general, since $\mathbf{G}(z)$ is a matrix of FIR filters, its inverse involves IIR filters. In 'real-life' applications a priori knowledge of the mixing is usually not available. Instead, multichannel BSS (blind deconvolution) techniques are applied that attempt to extract the source signals by estimating the unmixing matrix [14,40,80,89]. However, finding such an unmixing matrix (i.e. extraction of the sources) is not our interest here.

---

[1] In the communications context, $\Gamma(z)$ defines the zero-forcing equaliser which can separate the input signals and mitigate (equalise) the ISI [80].

4.1.2 Polynomial Covariance Matrix

Since the source signals are statistically independent they are strongly decorrelated, i.e. $\mathrm{E}\left[w_k(t)w_m^*(t-\tau)\right]=0$, for $m\neq k$ and $\tau\in\mathbb{Z}$. Let the vector, $\mathbf{w}(t)\in\mathbb{C}^K$, represent terms from the K source signals at time t. The matrix $\mathbf{a}_{ww}(\tau)\in\mathbb{C}^{K\times K}(\tau)$ is a *space-time covariance matrix* for the vector of sources and takes the diagonal form

$$\mathbf{a}_{ww}(\tau)=\mathrm{E}\left[\mathbf{w}(t)\mathbf{w}^{\mathrm{H}}(t-\tau)\right]=\begin{pmatrix} a_{11}(\tau) & & \mathbf{0} \\ & a_{22}(\tau) & \\ & & \ddots \\ \mathbf{0} & & a_{KK}(\tau) \end{pmatrix},\quad \tau\in\mathbb{Z} \tag{4.5}$$

where $a_{kk}(\tau)$ denotes the true autocorrelation sequence of the k^{th} source. This matrix describes both spatial and temporal relationships between signals. It follows that the z-transform of $\mathbf{a}_{ww}(\tau)$, which is called a *polynomial covariance matrix*, is also diagonal, in other words

$$\mathbf{A}_{ww}(z)=\sum_{\tau=-T_a}^{T_a}\mathbf{a}_{ww}(\tau)z^{-\tau}=\mathrm{diag}\left[A_{11}(z),A_{22}(z),\ldots,A_{KK}(z)\right]\in\mathbb{C}^{K\times K}(z), \tag{4.6}$$

where $A_{kk}(z)=\sum_{\tau=-T_a}^{T_a}a_{kk}(\tau)z^{-\tau}$, $T_a\in\mathbb{Z}$, $T_a\geq 0$. It is assumed that $\mathbf{a}_{ww}(\tau)=0,\ \forall|\tau|>T_a$, i.e. $a_{kk}(\tau)$ has finite support. Evaluation of $\mathbf{A}_{ww}(z)$ on the unit circle (i.e. for $z=e^{\mathrm{j}\omega}$) produces the true PSD matrix, $\mathbf{A}_{ww}(e^{\mathrm{j}\omega})\in\mathbb{C}^{K\times K}$ of the form of (3.10), which is also diagonal. The diagonal entries, $A_{kk}(e^{\mathrm{j}\omega})$, of $\mathbf{A}_{ww}(e^{\mathrm{j}\omega})$ are the true power spectrum of $w_k(t)$, i.e. the Fourier transform of $a_{kk}(\tau)$.

As a result of the mixing process in (4.1), the received signals $x_m(t)$ will generally be correlated, i.e.

$$a_{km}(\tau)=\mathrm{E}\left[x_k(t)x_m^*(t-\tau)\right]\neq 0,\ \text{for } k\neq m, \tag{4.7}$$

where $a_{km}(\tau)$ is an individual entry of the true space-time covariance matrix:

$$\mathbf{a}(\tau)=\mathrm{E}\left[\mathbf{x}(t)\mathbf{x}^{\mathrm{H}}(t-\tau)\right],\quad \tau\in\mathbb{Z} \tag{4.8}$$

for the signals $\underline{X}(z)$. It follows that $\mathbf{a}(\tau)$ will generally not be diagonal for $\forall\tau$. The true polynomial covariance matrix is given by

$$\mathbf{A}(z)=\sum_{\tau=-T_c}^{T_c}\mathbf{a}(\tau)z^{-\tau}\in\mathbb{C}^{M\times M}(z), \tag{4.9}$$

where $T_c \in \mathbb{Z}$, $T_c \geq 0$ and $\mathbf{a}(\tau) = 0$, $\forall |\tau| > T_c$. The matrix $\mathbf{A}(z)$ has the non-diagonal form:

$$\mathbf{A}(z) = \mathbf{G}(z)\mathbf{A}_{ww}(z)\tilde{\mathbf{G}}(z) + \mathbf{I}\sigma_\psi^2 = \begin{pmatrix} A_{11}(z) & A_{12}(z) & \cdots & A_{1M}(z) \\ A_{21}(z) & A_{22}(z) & \cdots & A_{2M}(z) \\ \vdots & \vdots & \ddots & \vdots \\ A_{M1}(z) & A_{M2}(z) & \cdots & A_{MM}(z) \end{pmatrix}. \tag{4.10}$$

The individual entries of this polynomial matrix, $A_{km}(z) | k \neq m$, correspond to the cross-correlation between the k^{th} and m^{th} channels. It follows directly that the PSD matrix $\mathbf{A}(e^{j\omega})$ will also not be diagonal.

By virtue of its construction, $\mathbf{A}(z)$ is parahermitian, i.e. $\mathbf{A}(z) = \tilde{\mathbf{A}}(z)$ or $[\mathbf{a}(\tau)]_{km} = a_{km}(\tau) = a_{mk}^*(-\tau) = [\mathbf{a}(-\tau)]_{mk}^*$, $k, m \in \{1,2,...M\}$ (see section 2.3.2). Furthermore, it is assumed $\mathbf{A}(z)$ is positive semidefinite: we shall say a parahermitian matrix is positive semidefinite if for all $\underline{P}(z)$, the parahermitian form

$$A(z) = \tilde{\underline{P}}(z)\mathbf{A}(z)\underline{P}(z) \text{ has } a(0) \geq 0. \tag{4.11}$$

where $a(0)$ is the zero-lag coefficient in the sequence $a(\tau)$. This definition is a generalisation of that for scalar matrices in [21].

Since the EVD can only measure and remove instantaneous correlations, it cannot sufficiently diagonalise the matrix $\mathbf{A}(z)$. Our goal is to find a matrix decomposition of $\mathbf{A}(z)$ that satisfies properties that are generalisations of those satisfied by the EVD given in section 2.2. That is, we require an energy preserving transformation that diagonalises $\mathbf{A}(z)$ and imposes a frequency-independent ordering regime on the diagonals of the resultant PSD matrix. Amongst other advantages (e.g. general rank determination), this type of transformation would allow for the estimation of the broadband signal and noise subspaces.

4.2 Extension of the EVD and SVD to Polynomial Matrices

4.2.1 Polynomial EVD

Consider the parahermitian matrix $\mathbf{A}(z)$ in (4.10). We desire a transformation $\mathbf{H}(z)$ such that

$$\mathbf{C}(z) = \mathbf{H}(z)\mathbf{A}(z)\tilde{\mathbf{H}}(z) \tag{4.12}$$

is a diagonal polynomial matrix, i.e.

$$\mathbf{C}(z) = \begin{pmatrix} C_{11}(z) & & \mathbf{0} \\ & C_{22}(z) & \\ \mathbf{0} & & \ddots \\ & & C_{MM}(z) \end{pmatrix}, \tag{4.13}$$

and the diagonal entries of the corresponding PSD matrix, $\mathbf{C}(e^{j\omega})$, are ordered (without loss of generality) in the manner

$$C_{kk}(e^{j\omega}) \geq C_{(k+1)(k+1)}(e^{j\omega}), \quad k = 1,2,\ldots M-1 \text{ and } \forall \omega. \tag{4.14}$$

Equation (4.14) is the spectral majorisation property given in (3.11) but for arbitrary multichannel data. In addition, we demand that the matrix $\mathbf{H}(z)$ is constrained to be paraunitary, i.e. $\mathbf{H}(z)\tilde{\mathbf{H}}(z) = \tilde{\mathbf{H}}(z)\mathbf{H}(z) = \mathbf{I}$. This ensures that the total energy, as constituted by the matrix $\mathbf{A}(z)$, is invariant to the application of the transformation. Note that in the degenerate case of order zero, this reduces to the definition of a conventional unitary matrix. We also require that the polynomial entries of $\mathbf{H}(z)$ have finite order, so that $\mathbf{H}(z)$ represents a matrix of FIR filters. Hence, the FIR paraunitary transformation $\mathbf{H}(z)$ is linear, convolutive and lossless (i.e. it is stable and causal with an all-pass frequency response).

The decomposition in (4.12) is not known to exist in general. In this thesis, we shall assume that an approximation to the decomposition exists. The challenge then is to compute a paraunitary matrix $\mathbf{H}(z)$ such that the polynomial matrix $\mathbf{C}(z)$ is as close to diagonal as possible. In general, it will not be possible to achieve exact diagonalisation since the paraunitary matrix $\mathbf{H}(z)$ represents an array of FIR filters; one cannot expect the filters to reverse the (second-order) transformations imposed by a mixing matrix of FIR filters. However, if the order of the entries of $\mathbf{H}(z)$ is sufficiently large, the diagonalisation can be achieved to a very good approximation.

The polynomial matrix decomposition of (4.12) may be looked upon as an extension of the conventional EVD to polynomial (space-time) covariance matrices, i.e. it is a *polynomial matrix EVD* (PEVD). Recall that the conventional EVD takes a Hermitian matrix, $\mathbf{A}$, and finds a unitary matrix, $\mathbf{U}$, such that $\mathbf{D} = \mathbf{U}\mathbf{A}\mathbf{U}^{\mathrm{H}}$ is a diagonal matrix with the eigenvalues of $\mathbf{A}$ on its diagonal. The eigenvalues are usually ordered with the most dominant located in the first (top) row. In the same way, the diagonals $\mathbf{C}(z)$ obtained by performing a PEVD may be regarded as eigenvalues that are functions of the indeterminate z, that is, they are *polynomial eigenvalues*. Correspondingly, the rows of $\mathbf{H}(z)$ may be seen as the *polynomial eigenvectors* of $\mathbf{A}(z)$ which form an orthonormal set [64,65]. In other words, at each frequency ω, $\mathbf{H}(e^{j\omega}) \in \mathbb{C}^{M\times M}$ is

comprised of the eigenvectors of the Hermitian matrix $\mathbf{A}(e^{j\omega})$. It is easy to see the connection between $C_{kk}(z)$ and the scalar eigenvalues d_k found by the EVD if we made the components of (4.12) scalar matrices; the expression would simply be that for the EVD.

A general polynomial matrix is not necessarily paraunitary, and therefore it is vital to ensure that the diagonalisation is carried out over the restricted subspace of paraunitary matrices. Methods of generating FIR paraunitary matrices have been discussed in the context of subband coding in chapter 3. Later in this chapter (sections 4.3 to 5), we explore techniques for constructing FIR paraunitary matrices for arbitrary multichannel data with the aim of performing broadband subspace decomposition.

Sample Polynomial Covariances

In practice, the design of a paraunitary matrix is usually based on an estimate of the true polynomial covariance matrix $\mathbf{A}(z)$. This is because we often do not have explicit knowledge of $\mathbf{A}(z)$. The received data vector $\mathbf{x}(t)$ in equation (4.3) is used to generate the sample polynomial covariance matrix, which is typically of the form:

$$\mathbf{R}(z) = \sum_{\tau=-t_1}^{t_1} \mathbf{r}(\tau) z^{-\tau}, \tag{4.15}$$

where

$$\mathbf{r}(\tau) = \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{x}(t)\mathbf{x}^{\mathrm{H}}(t-\tau), \tag{4.16}$$

T is the number of data samples used to obtain the estimate and $t_1 \geq 0$. Individual entries of $\mathbf{r}(\tau)$ are denoted as

$$[\mathbf{r}(\tau)]_{km} = r_{km}(\tau). \tag{4.17}$$

It is assumed that $\mathbf{r}(\tau) \cong 0$ for $|\tau| > t_1$ and that that $T >> t_1$. In practice, the value of t_1 is often measured experimentally. It follows that $r_{km}(\tau) = r_{mk}^{*}(-\tau)$, and so the polynomial matrix $\mathbf{R}(z)$ is parahermitian by construction.

Given the data $\mathbf{x}(t)$ or $\mathbf{R}(z)$, a PEVD algorithm can then be used to construct a paraunitary matrix $\mathbf{H}(z)$ such that

$$\mathbf{S}(z) = \mathbf{H}(z)\mathbf{R}(z)\tilde{\mathbf{H}}(z), \tag{4.18}$$

where $\mathbf{S}(z)$ is approximately diagonal; more specifically,

$$\mathbf{S}(z) \cong \mathrm{diag}[S_{11}(z), S_{22}(z), \ldots S_{MM}(z)], \tag{4.19}$$

and the diagonals $S_{kk}(z)$ are ordered, without loss of generality, according to (4.14); i.e.

$$S_{kk}(e^{j\omega}) \geq S_{(k+1)(k+1)}(e^{j\omega}), \quad k = 1,2,\ldots M-1 \text{ and } \forall \omega. \tag{4.20}$$

The transformed matrix $\mathbf{S}(z)$ is a consistent estimate of (4.13).

4.2.2 Polynomial SVD

The matrix $\mathbf{H}(z)$ is also a space-time analogue of the unitary matrix found by the SVD, and so may be applied to a more general polynomial matrix. We know that, conventionally, the EVD can be used to find the SVD of a scalar matrix. This can be extended to polynomial matrices as follows. Let $\mathbf{X}(z) \in \mathbb{C}^{M \times K}(z)$. A PEVD will find a paraunitary matrix $\mathbf{U}(z)$ such that the parahermitian matrix $\mathbf{A}(z) = \mathbf{X}(z)\tilde{\mathbf{X}}(z)$ is diagonal, i.e.

$$\mathbf{D}(z) = \mathbf{U}(z)\mathbf{A}(z)\tilde{\mathbf{U}}(z), \tag{4.21}$$

where $\mathbf{D}(z) = \mathrm{diag}\{D_{11}(z), D_{22}(z), \ldots D_{MM}(z)\}$ and $D_{kk}(e^{j\omega}) \geq D_{(k+1)(k+1)}(e^{j\omega}), \forall \omega$. We may apply $\mathbf{U}(z)$ directly to $\mathbf{X}(z)$ and obtain the equality $\mathbf{U}(z)\mathbf{X}(z) = \mathbf{V}(z)$. It is clear from this that $\mathbf{V}(z)\tilde{\mathbf{V}}(z) = \mathbf{D}(z)$ so the rows of $\mathbf{V}(z)$ are orthogonal. It follows that the identity

$$\mathbf{D}(z) = \mathbf{U}(z)\mathbf{X}(z)\tilde{\mathbf{V}}(z) \tag{4.22}$$

is an effective generalisation of the SVD to polynomial matrices, namely a space-time or *polynomial matrix* SVD (PSVD). The diagonals $D_{kk}(z)$ obtained with a PSVD may be viewed as the *polynomial singular values*; $\mathbf{U}(z)$ and $\tilde{\mathbf{V}}(z)$ are the left- and right-hand *polynomial singular vectors*, respectively.

4.2.3 Properties of a PEVD/PSVD

It is possible to generalise the requirements from properties of matrices of polynomials to properties of signals. This helps to highlight the connection of the paraunitary matrix found by a PEVD or PSVD to signal processing applications; particularly, broadband sensor array processing and subband coding.

Strong Decorrelation. The paraunitary polynomial matrix $\mathbf{H}(z)$, as defined in (4.12), may be applied to the received multichannel data $\underline{X}(z)$ (Fig. 4.1) to produce the transformed sequences $\underline{Y}(z)$ according to

$$\begin{pmatrix} Y_1(z) \\ Y_2(z) \\ \vdots \\ Y_M(z) \end{pmatrix} = \begin{pmatrix} H_{11}(z) & H_{12}(z) & \cdots & H_{1M}(z) \\ H_{21}(z) & H_{22}(z) & \cdots & H_{2M}(z) \\ \vdots & \vdots & \ddots & \vdots \\ H_{M1}(z) & H_{M2}(z) & \cdots & H_{MM}(z) \end{pmatrix} \begin{pmatrix} X_1(z) \\ X_2(z) \\ \vdots \\ X_M(z) \end{pmatrix} \tag{4.23}$$

or more compactly

$$\underline{Y}(z) = \mathbf{H}(z)\underline{X}(z), \tag{4.24}$$

where, as before, $\underline{X}(z)$ and $\underline{Y}(z)$ denote algebraic power series. Now we can make the following assertion.

Theorem. 4.1. *If the condition of equation (4.13) is satisfied, then the paraunitary matrix* $\mathbf{H}(z)$ *can be applied to* $\underline{X}(z)$ *to produce transformed signals that satisfy the strong decorrelation property.*

Proof. By analogy with (4.9), the polynomial covariance matrix of the transformed signals is:

$$\mathbf{C}_{yy}(z) = \sum_{\tau} \mathrm{E}\left[\mathbf{y}(t)\mathbf{y}^{\mathrm{H}}(t-\tau)\right] z^{-\tau} = \mathbf{H}(z)\mathbf{A}(z)\tilde{\mathbf{H}}(z). \tag{4.25}$$

where $\mathbf{y}(t)$ relates to the series $\underline{Y}(z)$. From (4.13) and (4.25) it is easy to see that the output signals $\underline{Y}(z)$ have been spatially whitened or strongly decorrelated[2] since

$$\mathbf{H}(z)\mathbf{A}(z)\tilde{\mathbf{H}}(z) = \mathbf{C}(z). \tag{4.26}$$

Q.E.D.

This proof naturally carries over to the case of sample statistics and the lossless filter bank $\mathbf{H}(z)$ in (4.18) for the sample polynomial matrix $\mathbf{R}(z)$ defined in (4.15). Hence, an approximation to $\mathbf{C}_{yy}(z)$ is given by

$$\mathbf{S}(z) = \mathbf{H}(z)\mathbf{R}(z)\tilde{\mathbf{H}}(z). \tag{4.27}$$

It follows that

$$\mathbf{S}(z) = \mathrm{diag}\left[S_{11}(z), S_{22}(z), \ldots S_{MM}(z)\right] \cong \mathbf{C}_{yy}(z), \tag{4.28}$$

where

---

[2] In the context of decorrelating multichannel broadband data, a practical PEVD (or PSVD) may be regarded as a *second-order* BSS (i.e. the first step in BSS) technique. The paraunitary matrix $\mathbf{H}(z)$ generated is an estimate of the second-order unmixing matrix. It is 'second-order' in the sense that second-order statistical dependencies between the signals are being minimised.

$$S_{kk}(z) \equiv C_{kk}(z) = \sum_{\tau} \mathrm{E}\left[y_k(t) y_k^*(t-\tau)\right] z^{-\tau}, \quad k = 1,2,\cdots M \,. \tag{4.29}$$

Spectral Majorisation. The ordering imposed on the diagonals in (4.14) and (4.20) amounts to ordering the signals $\mathbf{y}(t)$ such that the variance of $y_1(t)$ is greater than the variance of $y_2(t)$ at every frequency ω, and the variance of $y_2(t)$ is greater than that of $y_3(t)$ at every ω, and so on. That is, the signals $\mathbf{y}(t)$ satisfy spectral majorisation. Recalling the discussions in section 4.2.1, we can say that spectral majorisation is a generalisation of the eigenvalue ordering performed by the EVD to polynomial eigenvalues.

Theorem 4.2. *If the conditions of equations (4.13) and (4.14) are satisfied, then the set of power spectra* $\{C_{kk}(e^{\mathrm{j}\omega})\}$ *is unique.*

Proof. This may be shown by simple generalisation of the proof in [74]. The fact that $\mathbf{H}(e^{\mathrm{j}\omega})$ is unitary for each fixed ω implies that that the power spectra $C_{kk}(e^{\mathrm{j}\omega})$ are eigenvalues of $\mathbf{A}(e^{\mathrm{j}\omega})$ for each ω. Suppose the majorisation property is satisfied. Then at each and every frequency ω the eigenvalues are ordered as a sequence of decreasing magnitudes. Since the set of eigenvalues is unique, each diagonal entry in (4.13) is uniquely determined for each ω. Thus, the set of power spectra, which has the majorisation property, is unique[3]. Q.E.D.

It is easy to see from (4.29) that this proof also shows that the set of sample power spectra $\{S_{kk}(e^{\mathrm{j}\omega})\}$ are unique.

Energy Compaction. It is obvious from (4.14) that the transformed signals are also ordered in terms of their total spectral energy, i.e.

$$\mathrm{E}\left[\left|y_k(t)\right|^2\right] \geq \mathrm{E}\left[\left|y_{(k+1)(k+1)}(t)\right|^2\right], \quad k = 1,2,\ldots,M-1, \tag{4.30}$$

which is a weaker condition than spectral majorisation.

Energy Conservation. Since the PSD matrix transforms according to (4.12) and $\mathbf{H}(z)$ is paraunitary, it can be shown that

$$\begin{aligned} \sum_{k=1}^{M} \sum_{\tau=-\infty}^{\infty} \mathrm{E}\left[x_k(t) x_k^*(t-\tau)\right] z^{-\tau} &= \mathrm{trace}\{\mathbf{A}(z)\} = \mathrm{trace}\{\mathbf{H}(z)\mathbf{A}(z)\widetilde{\mathbf{H}}(z)\} \\ &= \mathrm{trace}\{\mathbf{C}(z)\} = \sum_{k=1}^{M} \sum_{\tau=-\infty}^{\infty} \mathrm{E}\left[y_k(t) y_k^*(t-\tau)\right] z^{-\tau} \end{aligned} \,. \tag{4.31}$$

---

[3] It should be noted that $\mathbf{H}(z)$ may not be unique since the matrix of eigenvectors $\mathbf{H}(e^{\mathrm{j}\omega})$ may not be unique.

and so the combined signal power is preserved. Without this property, the power of the output signals would have no physical significance. Setting $z = e^{j\omega}$ in (4.31) leads to the much stronger result that

$$\sum_{k=1}^{M} \left| X_k(e^{j\omega}) \right|^2 = \sum_{k=1}^{M} \left| Y_k(e^{j\omega}) \right|^2, \qquad \forall \omega. \tag{4.32}$$

In other words, the total power of the input signals is preserved at every frequency under the transformation. The transformation can redistribute energy between the signals; however, it cannot amplify or attenuate energy. Without this (energy preservation) property the output signals would have no physical significance.

Permutation Indeterminacy

A paraunitary $\mathbf{H}(z)$ that is found to diagonalise a parahermitian matrix is not unique. The lack of uniqueness is easiest explained in the Fourier domain. The filter bank $\mathbf{H}(z)$ is a unitary matrix that diagonalises the Hermitian matrix at each frequency. This means that the order in which the results in each frequency bin occur may differ to those in other bins. It is crucial to remove this ambiguity if, say the desire is to perform signal separation (which would require a second stage for minimisation of higher-order statistics).

A paraunitary permutation matrix can be cascaded with $\mathbf{H}(z)$ independently at each frequency in order to correct for any permutation. The filter bank $\mathbf{H}(z)$ retains the same property, since the permutation matrix is paraunitary. As specified by (4.13) and (4.14), our interest is to investigate the more restrictive case where $\mathbf{H}(z)$ not only strongly decorrelates the input signals but also spectrally majorises them. This removes the freedom to arbitrarily permute the channels at each frequency. Such a matrix is unique, up to the application of an all-pass matrix, such as $\mathrm{diag}\left[1,\ldots,1,z^{-\tau}\right]$.

Broadband Subspace Decomposition

It may not be immediately clear to the reader as to the physical meaning of broadband subspace decomposition. A possible explanation is given here by generalising narrowband subspace decomposition to broadband signals. Consider again the convolutive mixing scenario in Fig. 4.1. Assume that there are a large number of samples in the received data, the SNR of the signals is sufficiently high and the rank of $\mathbf{G}(z)$ is equal to the number of source signals. Now suppose a paraunitary matrix $\mathbf{H}(z)$ is computed that when applied to a set of correlated signals produces signals $\underline{Y}(z)$ that satisfy strong decorrelation. Further suppose that spectral majorisation holds for the corresponding set of spectra. This means that for each ω, the

variances of the strongly decorrelated signals are ordered in a decreasing fashion. The first ρ polynomial eigenvalues will represent most of the output signal power, and the rows of $\underline{Y}(z)$ corresponding to the first ρ diagonals constitute the broadband signal subspace. The other $M-\rho$ rows correspond to the broadband noise subspace, which may therefore be identified and separated.

At each ω, $\mathbf{H}(e^{j\omega})$ is a unitary matrix that diagonalises the Hermitian matrix $\mathbf{A}(e^{j\omega})$. The matrix $\mathbf{H}(e^{j\omega})$ is partitioned into a (narrowband) signal subspace, $\mathbf{H}(e^{j\omega})^{(s)}$, and noise subspace, $\mathbf{H}(e^{j\omega})^{(n)}$, for all ω. Since $\mathbf{H}(z)$ is paraunitary the distribution of energy between the resulting broadband signal subspace and noise subspaces is guaranteed to have proper physical significance. Note that if the signals transformed by the paraunitary matrix satisfy strong decorrelation and spectral majorisation then an estimate of the broadband signal and noise subspaces may be obtained.

There are other possible requirements of a PEVD that also correspond to properties of the EVD/SVD for narrowband sources. In particular, a technique that computes a PEVD can also be used to achieve efficient coding of data. This is the topic of discussion in chapter 6.

4.2.4 Performance Measures

Diagonalisation Performance Measure

In order to assess the performance of algorithms that compute the PEVD, a measure based on the true statistics of the input signals is introduced. This measure is defined in terms of the parahermitian matrix $\mathbf{C}(z)$ in (4.12):

$$\lambda = \frac{\sum_{k\neq m=1}^{M}\sum_{\tau}\left|c_{km\tau}\right|^2}{\sum_{m=1}^{M}\sum_{k=1}^{M}\sum_{\tau}\left|c_{km\tau}\right|^2}, \quad \text{where } \mathbf{C}(z)=\mathbf{H}(z)\mathbf{A}(z)\widetilde{\mathbf{H}}(z). \tag{4.33}$$

The numerator represents the sum of the squares of the moduli of the off-diagonal terms of $\mathbf{C}(z)$ at all time delays. The denominator is the sum of the squares of all the matrix terms at all time delays, which is conserved. The λ measure indicates how well a PEVD/PSVD algorithm would diagonalise a matrix (or strongly decorrelate the input signals). That is, it gauges the amount of energy leakage there is from the diagonals to off-diagonals of the output polynomial

covariance matrix[4]. This measure is always non-negative and takes values between 0 and 1. The closer it is to zero, the more successfully the algorithm is in performing polynomial matrix diagonalisation. The condition $\lambda = 0$ means perfect diagonalisation of the matrix has been attained.

Suppose that $\underline{W}(z)$ represents K real, discrete-time, stochastic, i.i.d. sources which are convolutively mixed as in Fig. 4.1. In this case, the quantities $c_{km\tau}$ may be computed as follows. These sources are mixed using a polynomial-mixing matrix $\mathbf{G}(z)$ and then transformed by a paraunitary matrix, $\mathbf{H}(z)$. Let $\mathbf{A}_{ww}(z)$ be the polynomial covariance matrix for the sources $\underline{W}(z)$. The (parahermitian) covariance matrix, $\mathbf{A}(z)$, for the mixed signals is:

$$\mathbf{A}(z) = \mathrm{E}\left[\mathbf{G}(z)\mathbf{A}_{ww}(z)\tilde{\mathbf{G}}(z)\right] = \mathbf{G}(z)\tilde{\mathbf{G}}(z), \tag{4.34}$$

since $\mathbf{A}_{ww}(z) = \mathbf{I}$. Similarly, $\mathbf{C}(z)$ is the mixed-decorrelated signal covariance matrix:

$$\mathbf{C}(z) = \mathrm{E}\left[\mathbf{H}(z)\mathbf{A}(z)\tilde{\mathbf{H}}(z)\right] = \mathbf{H}(z)\mathbf{G}(z)\tilde{\mathbf{G}}(z)\tilde{\mathbf{H}}(z). \tag{4.35}$$

Hence, the expectation of λ for a particular $\mathbf{H}(z)$ may be determined directly if we have knowledge of $\mathbf{G}(z)$.

Subspace Decomposition Performance Measure

The subspace estimation accuracy of a PEVD/PSVD algorithm for broadband signals may be quantified. Consider again the parahermitian matrix $\mathbf{C}(z)$ for the transformed broadband signals $\underline{Y}(z)$ in (4.24) and the convolutive mixing model illustrated in Fig. 4.1. We define a measure based on the projection of the source signals onto the estimate of the signal-plus-noise and noise subspaces obtained by a PEVD/PSVD. It is given by

$$\gamma = \frac{\gamma_\eta}{\gamma_s} = \frac{\sum_{k=K+1}^{M} c_{kk0}}{\sum_{k=1}^{K} c_{kk0}}, \tag{4.36}$$

where, as before, K is the number of source signals, M is the number of received signals and c_{kk0} is the zero-lag term of the autocovariance (i.e. the variance) of the k^{th} transformed signal. Heuristically, the quantity γ_s is a measure of the accuracy of the signal-plus-noise subspace estimated by the PEVD/PSVD. More formally, it is equivalent to taking the sum of the resultant

[4] It can also be viewed as a measure of how well the algorithm, as an initial step of BSS, could be expected to estimate the second-order mixing matrix up to permutation and scaling effects.

powers from projecting the original signals onto the PEVD/PSVD estimate of the signal-plus-noise subspace. The quantity γ_η indicates the accuracy of the noise subspace estimated by the PEVD/PSVD; it is equivalent to the sum of the resultant powers from projecting the original signals onto the noise subspace estimated by the PEVD/PSVD. A small value of γ indicates good subspace estimation. Perfect subspace estimation is achieved when $\gamma = 0$, which is not possible in practice.

The γ measure is dependent on the SNR of the inputs $\underline{X}(z)$, which is given by

$$SNR_{in} = \frac{P_X}{P_\psi} = \frac{\text{trace}[\mathbf{R}(z)]_{z^0}}{\sigma_\psi^2 \text{trace}[\mathbf{I}]} = \frac{\text{trace}\left[\mathbf{G}(z)\tilde{\mathbf{G}}(z)\right]\Big|_{z^0}}{\sigma_\psi^2 \text{trace}[\mathbf{I}]}. \tag{4.37}$$

The notation $\mathbf{A}(z)\big|_{z^0}$ is used to mean that $\mathbf{A}(z)$ is only evaluated for the zero lag term. A small value of γ can be achieved with a large value of SNR_{in}. In the case where the thermal noise power of the receive antenna is relatively large, γ is large.

4.3 Review of Algorithms for Obtaining a PEVD/PSVD

4.3.1 Lattice Parameterisation

A paraunitary matrix can be obtained by the parameterisation of Vaidyanathan's lattice structure presented in section 3.4.1. As demonstrated there, the parameterisation of this type of lattice can be computationally costly and complex optimisation is often required. Regalia and Loubaton [57] propose a filter bank design algorithm based on this premise for rational signal subspace analysis. It uses an eigenstructure based adaptive algorithm for the optimisation of multiple parameters. The algorithm constructs the paraunitary matrix (adapts the lattice parameters) according to the received signal samples iteratively (i.e., it is a signal-adapted or online technique). This technique can avoid converging to local minima, which is an advantage over gradient-based algorithms. However, the optimisation of the multivariable lattice filter is nonlinear (the update equations resemble those of standard gradient descent methods) and computationally complex.

4.3.2 Frequency-Domain Techniques

The strong decorrelation of signals may be performed in the frequency-domain. A rather simple but "naive" approach would be to take the DFT over the entire data length and apply the SVD/EVD to decorrelate the signals at each discrete frequency independently of other

frequencies. This reduces the broadband problem to narrowband form. The crucial problem with this is that there are many quantities (unitary matrices) to estimate with insufficient data samples (one sample per bin) to use in obtaining a reliable estimate. An extension of this approach is to split the input spectra into frequency bands (subbands) and apply a different (decorrelating) unitary matrix to the signals at each subband. The principle is the same as that behind the independent frequency band (IFB) method used in space-time adaptive processing [11,36]. There are two major drawbacks with an IFB type approach and thus the subband approach:

1. The phase (temporal) coherence (discontinuity) of the source signals may become corrupted because in each subband the SVD will order the output signal powers in order of decreasing energy irrespective of the ordering in neighbouring subbands. In other words, a unitary matrix applied to a subband is independent of that applied to any other subband; therefore, the signals in each subband will undergo a different phase change to those in any other subband. For this reason, an IFB-based approach would generally not be suitable for some applications where minimal signal corruption is necessary; e.g., signal separation of audio, in speech recognition, for example, and medical signals, e.g. electrocardiographic (ECG) and electroencephalographic (EEG) data.

2. Since a unitary matrix is computed independently for each subband, the algorithm will disregard correlations that may exist between the subbands, which may be important. The extent to which decorrelation is performed is restricted because these (usually relatively small) correlations are not minimised. This would be disadvantageous in applications such as subband coding, where the algorithm generally could not be expected to produce optimal filter banks (in the sense of the coding gain).

An alternative frequency-domain approach, namely the multipath-enabled SVD, is proposed by Lambert *et al* [37] for signal subspace estimation of multipath (broadband) signals. He represents the convolutive mixing problem in terms of DFT filter matrices as well as polynomial matrices. An extension of the SVD to polynomial matrices is obtained by generalising some conventional linear algebra and control theory methods from the complex number field to the field of rational functions. The technique initially applies of a rectangular window function to all the entries of the space-time sample covariance matrix of the input signals. The length of this window is ideally made equal to the length of the true autocovariance function of the signals. The windowing removes any noise related cross-covariance terms, thus improving the algorithms estimate of the signal statistics. The technique then diagonalises the covariance matrix at each and every frequency using an extension of the QR-algorithm: a Householder transformation is applied to the (instantaneous) covariance matrix at each and every frequency; an ordering of the polynomial singular values (or equivalently, the diagonals of the covariance matrix) is performed according to total energy. The algorithm involves the

approximate inversion of FIR filters, and is therefore quite distinct from the algorithms developed in this thesis. An evaluation of this algorithm is given in section 4.5, where we also compare its performances to that of three subband approaches that are proposed in this thesis.

Hung and Kaveh [27] propose the use of *focusing matrices* that align (linearly transform) the signal subspaces of the narrowband components within the bandwidth of the signal, which is called focusing. The method of focusing takes the correlation matrices at different frequency bins and combines them to form a general correlation matrix that is a sufficient statistic for the spatial observation vectors. This matrix may then be used as the data matrix that is processed by the conventional MUSIC algorithm. With regards to the single channel case the DFT of the signal would need to be taken and split into sub-frequencies (bins). Each frequency bin would then have a different signal subspace that would be transformed by the corresponding focusing matrix to a subspace representing the baseband signal. This algorithm requires an initial estimate of the spatial characteristics of the impinging signals and the array in order to work. Coherent subspace estimation algorithms, such as this one, are known to generate erroneous subspaces due to the focusing process, which is exacerbated when using poor initial estimates. For this reason, these techniques cannot be considered as prior-art algorithms for computing a PEVD, and therefore will not be investigated further.

4.4 Bandwise Algorithm

In order to gather a better understanding of the time-domain SBR2 algorithm, which will be described later in section 5, we introduce and develop a family of frequency-domain subband approaches to computing (estimating) a PEVD in this thesis. These techniques are based on methods that are typically applied to problems related to classical spectral estimation [28,50].

4.4.1 Derivation of the Regular Bandwise Algorithm

Suppose there are M data sequences of length T samples. The first stage of the bandwise algorithm is to divide the sequences into non-overlapping equal length segments (blocks), as depicted in Fig. 4.2. Then, the DFT of the blocks of data is computed. Let T_b be the number of samples in a block so that there are $B = T/T_b$ (for $T_b \leq T$) blocks per channel k. The b^{th} block has the data $[x_{k,0b}, x_{k,1b}, \ldots, x_{k,\phi b}, \ldots x_{k,(T_b-1)b}]$, for $\{\phi \in \mathbb{N} \,|\, \{\phi \leq T_b - 1\}\}$ and $\{b \in \mathbb{N} \,|\, \{b \leq B-1\}$, corresponding to a set of Fourier-domain samples (or design frequencies). Let $\mathbf{x}_{\phi b} = [x_{1,\phi b}, x_{2,\phi b}, \ldots, x_{M,\phi b}]^{\mathrm{T}}$. A data matrix containing all the Fourier domain data points for the ϕ^{th} frequency is defined as $\mathbf{X}_{\phi} = [\mathbf{x}_{\phi 0}, \mathbf{x}_{\phi 1}, \ldots, \mathbf{x}_{\phi b}, \ldots \mathbf{x}_{\phi(B-1)}] \in \mathbb{C}^{M \times B}$ – see Fig. 4.2. The sample

covariance matrix of the data, which is Hermitian, is estimated at each design frequency independently, i.e.

$$\mathbf{R}_{\phi} = \mathbf{X}_{\phi}\mathbf{X}_{\phi}^{\mathrm{H}} = \mathbf{R}(e^{\mathrm{j}\omega_{\phi}}) \approx \mathbf{A}(e^{\mathrm{j}\omega_{\phi}}) \in \mathbb{C}^{M\times M}. \tag{4.38}$$

There are B data samples to use for each estimate. At the other frequencies (not included in the set), the covariance at the closest of the set of design frequencies is used. A unitary matrix, $\mathbf{U}_{\phi} \in \mathbb{C}^{M\times M}$, is then found that diagonalises $\mathbf{R}_{\phi}$. The matrices $\mathbf{U}_{\phi}$ define the paraunitary matrix, $\mathbf{H}(z) \in \mathbb{C}^{M\times M}(z)$, in the frequency domain for the entire data set. Hence, estimates of the covariance matrix and thus the unitary matrix are created at a set of frequencies.

A Modified Schur Decomposition Technique. The matrices $\mathbf{U}_{\phi}$ may be found by computing the Schur decomposition of $\mathbf{R}_{\phi}$. The problem of permutation indeterminacy – see section 4.2.3 – caused by ambiguities in the rotation values can be resolved by permuting the eigenvectors (and eigenvalues) of $\mathbf{R}_{\phi}$ at each frequency in accordance to the ordering regime in (4.14).

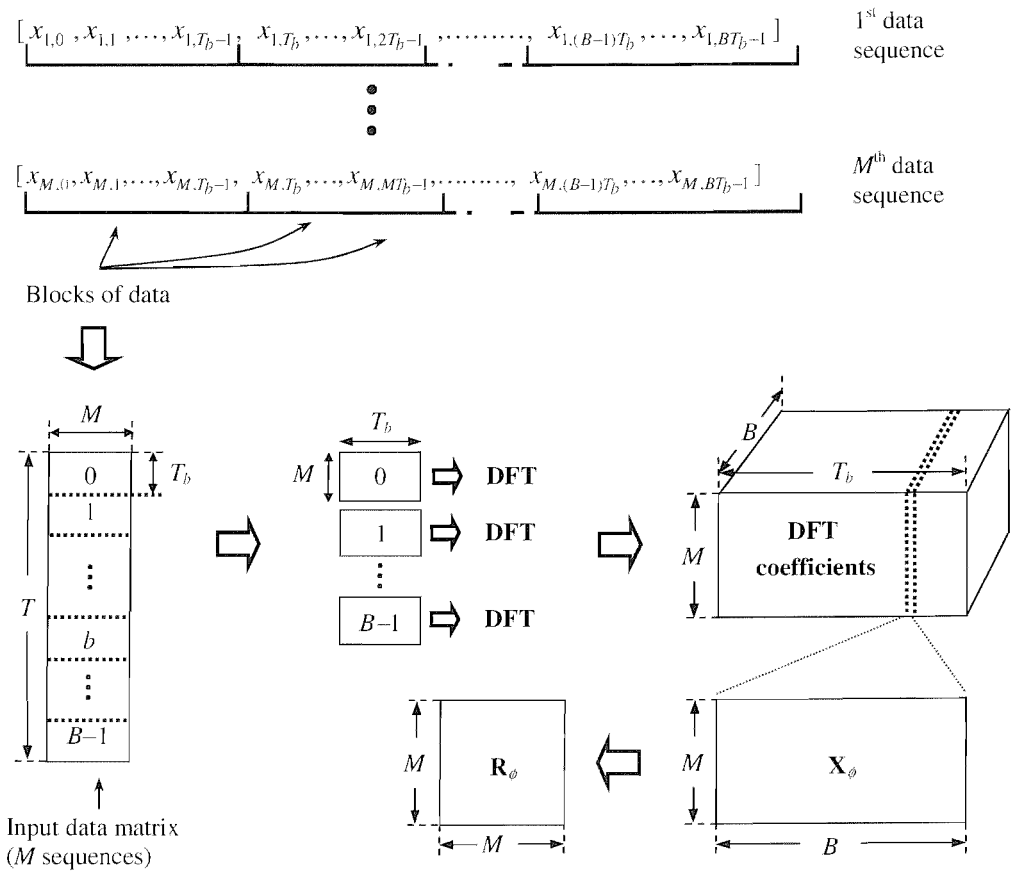


Figure 4.2: Illustration of the data sectioning performed by the bandwise algorithm.

In Fig. 4.3, we show a flowchart representation of the bandwise algorithm. The algorithm finds a frequency-domain paraunitary matrix $\mathbf{H}(z)$ that can strongly decorrelate signals. It may be applied directly to the data at each frequency by first obtaining the DFT of the data set. The matrix $\mathbf{H}(z)$ corresponds to a unitary matrix which is applied to the components (M complex numbers) at the ϕ^{th} frequency. For a set of frequencies ($T_b < T$), $\mathbf{H}(z)$ is also applied to those frequencies closest to the ϕ^{th} design frequency in the original sequences. The set of T_b frequencies may be viewed as unitary-matrix design frequencies for the original length-T sequences. The IDFT may be used to obtain the time-domain output signals.

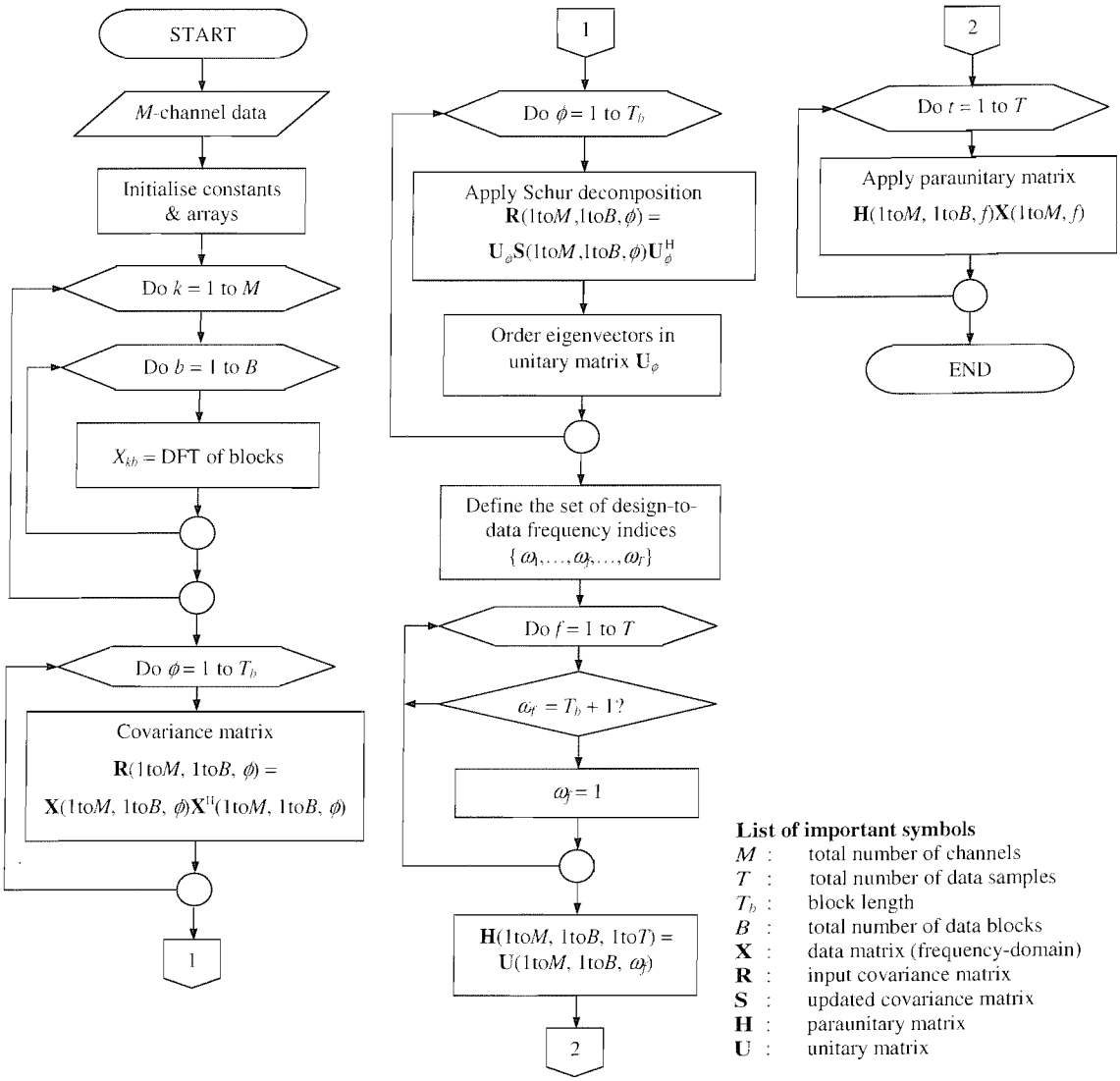


Figure 4.3: Flowchart of both the regular and SW-bandwise algorithms.

In order to simplify the bandwise algorithm the following assumptions are made:

1. The data is periodic (circulant): data sequences repeat periodically every T_b samples. Obviously, the simplification is gained since the assumption is naturally conducive to operating in the Fourier domain. However, this means that $\mathbf{H}(z)$ produced by the method does not have a time-domain equivalent; that is, $\mathbf{H}(z)$ is not the proper paraunitary matrix to apply to non-circulant data. The construction of a non-circulant $\mathbf{H}(z)$ with a frequency-domain algorithm seems to be a non-trivial task.

2. The data length T is an odd multiple of the block length T_b when assuming circulant data. Assuming this relationship between T and T_b guarantees that each frequency in the Fourier domain data is closest to a unique frequency from the block Fourier transforms.

Note that the description of the bandwise algorithm is for both complex-valued and real-valued data. However, our evaluation of the algorithm (presented later) is limited to the specific case of real-valued data. For this case, we must ensure that the complex numbers at opposite frequencies (positive and negative frequency components) of the Fourier domain data are exactly complex conjugates. Therefore, the unitary matrices selected for opposite frequencies will be complex conjugates as well. This guarantees that $\mathbf{H}(z)$ is real.

4.4.2 Sliding-Window Bandwise Algorithm

The bandwise algorithm can be improved upon by adopting the Welch method [87] of PSD estimation rather than the Bartlett estimator. The new algorithm, namely the sliding-window (SW) bandwise algorithm, carries out exactly the same processes as the regular bandwise algorithm with the exception of the following: the input signals are divided into *B overlapping* blocks of length T_b so that each block contains samples from neighbouring blocks, i.e. there is redundancy of samples in the blocks. The action of collecting data in this way may be viewed as sliding a window along the original data sequence and taking a 'snapshot' of data at periodic intervals, hence the algorithm's name. This is illustrated for a single time series in Fig. 4.4. The b^{th} block of the k^{th} signal is defined as

$$x'_{k,b}(\phi) = x_k(\phi + bT_o) \quad \text{for } \phi = 0,1,\ldots,T_b - 1, b = 0,1,\ldots,B-1, \tag{4.39}$$

where T_o is the number of samples from $x_k(t)$ that are not common to a given block and the preceding block (i.e. the overlap length is $T_b - T_o$). Recalling the assumptions made for simplification of the bandwise algorithm in section 4.4.1, the total number of blocks is given by

$B = \lfloor T - T_b/T_o \rfloor + 1$. The data matrix $\mathbf{X}_\phi \in \mathbb{C}^{M \times B}$ for the ϕ^{th} design frequency has the general form

$$\mathbf{X}_\phi = \begin{pmatrix} x_{1,\phi} & x_{1,T_o+\phi} & \cdots & x_{1,(B-1)T_o+\phi} \\ x_{2,\phi} & x_{2,T_o+\phi} & \cdots & x_{2,(B-1)T_o+\phi} \\ \vdots & \vdots & \ddots & \vdots \\ x_{M,\phi} & x_{M,T_o+\phi} & \cdots & x_{M,(B-1)T_o+\phi} \end{pmatrix}. \tag{4.40}$$

Using $\mathbf{X}_\phi$, an estimate of the covariance matrix is obtained for each design frequency ϕ.

We are now in a similar position to that in the bandwise algorithm, having estimates of the covariance matrix at a set of design frequencies. The SW-bandwise algorithm has two parameters: the block length T_b and the disparity length T_o. If $T_o = 1$, the disparity in data samples between consecutive overlapped blocks is a single data sample, in which case we say that there is maximal overlap. With $T_o = T_b$ there are contiguous blocks, i.e. no overlap, and the algorithm becomes the regular bandwise algorithm. A summary of the SW-bandwise algorithm is represented by the flow diagram for the regular bandwise algorithm in Fig. 4.3.

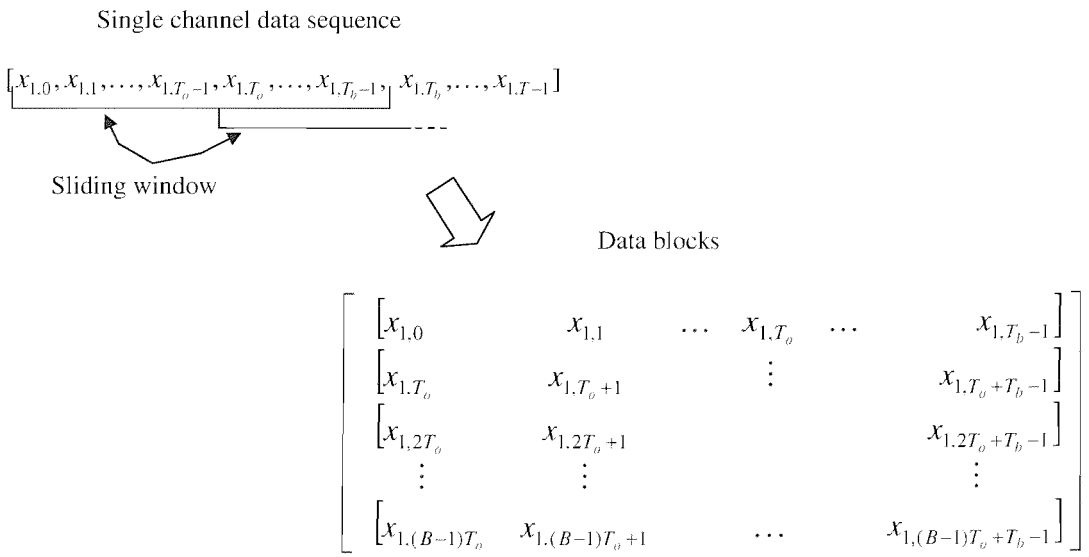


Figure 4.4: Illustration of the data segmentation employed by the SW-bandwise algorithm.

Data Windowing

The degree to which decorrelation can be achieved with the regular bandwise algorithm is constrained to an extent by the spreading of signal energy from one frequency component to nearby frequencies. This phenomenon, known as spectral leakage [50], occurs because the DFT

is based on the assumption that the signals are periodic. If, as in most cases, the signals operated on by the bandwise algorithm are not periodic, 'spiky' discontinuities occur at the endpoints of the blocks of data processed by the DFT, as shown in Fig. 4.5(a). The discontinuities give rise to spurious components in the signal's spectrum, causing a particular frequency component to appear as a spread of frequencies.

The effects of spectral leakage may be reduced by applying a tapered window function to the data sequence in each block [50]. A tapered window would attenuate the endpoints of each block smoothly – see Fig. 4.5(b) – and hence reduce the spread (smearing) of spectral energy. To elaborate, consider that the segmentation performed by the regular bandwise algorithm may be looked upon as implicit windowing with a rectangular window function. The Fourier transform of a rectangular window is a sinc function that has high sidelobe energy, as represented by the thin solid curve in Fig. 4.6(b). The sidelobes enable the distribution of signal energy in frequency. The problem is exacerbated for short data lengths. A tapered window may be applied that has low magnitude sidelobes, which alleviates spectral smearing.

The disadvantage of using a tapered window is the loss of spectral resolution since these windows have a broader mainlobe. A number of window functions have been devised that achieve a compromise between these conflicting requirements. Some popular choices are the Hann, Blackman and Hamming windows [50]. The time and frequency response of these windows may be compared in Fig. 4.6. For the SW-bandwise algorithm, a Blackman window function (represented by the thick solid curve in Fig. 4.6) is applied to each data block. A compromise is met between spectral leakage and frequency resolution with this window.

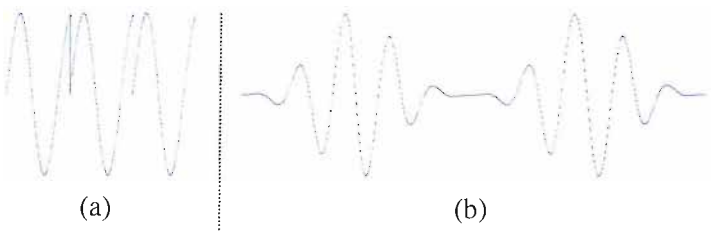


Figure 4.5: (a) Effect of windowing a signal with a rectangular window is discontinuities at the endpoints. (b) By comparison, when a tapered window is used the discontinuities are reduced.

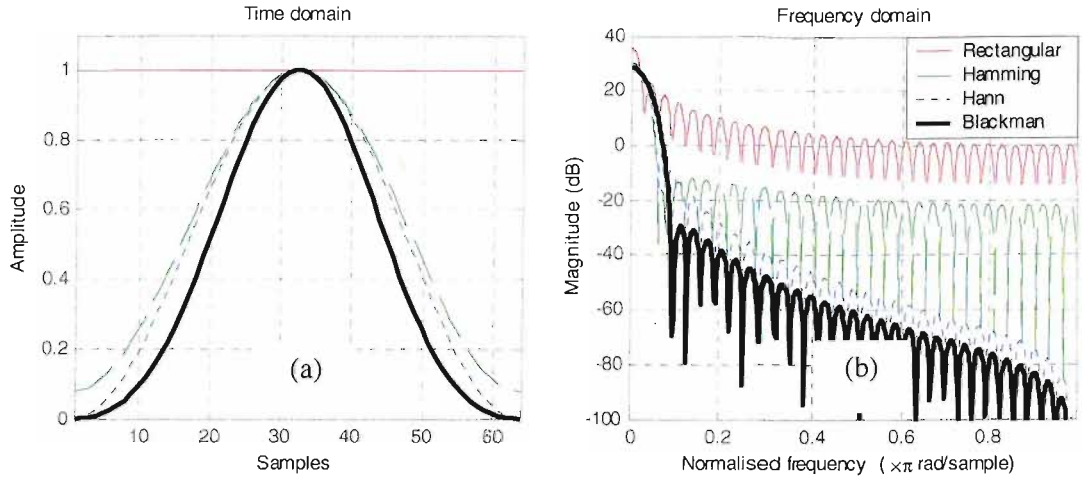


Figure 4.6: Comparison of well-known window functions.

4.4.3 Windowed Covariance-Domain Bandwise Algorithm

A third bandwise algorithm variant, termed the windowed covariance-domain (WCD) bandwise algorithm, may be obtained by adopting the Blackman-Tukey spectral estimation technique [7]. The WCD-bandwise algorithm carries out exactly the same processes as the other two (bandwise) variants with the exception of the following: the $M \times M$ sample polynomial covariance matrix of the data is computed; each of the entries (covariance functions) of the space-time parahermitian matrix is windowed (i.e., multiplied by a function of time-lag); and with a covariance matrix estimate $\mathbf{R}_f$ for each of the T frequencies, a paraunitary matrix is constructed as a unitary matrix given by the EVD at each and every frequency.

4.4.4 Other Possible Bandwise Algorithms

The DFT produces a sampled version of the true spectrum since there are only a finite number of samples that make up the data record. As a consequence, frequency components that are located between DFT samples (or bins) are attenuated. This effect is known as the 'picket-fence' phenomenon or scalloping loss [28,50]. A solution to this problem is to append zeros to the end of the block of data (i.e. zero-pad the data blocks). The result is in a smoother spectral estimate and allows for more precise localisation of a frequency component that does not correspond to an integer number of bins. Note that this effect should not be confused with improving spectral resolution. By the sampling theorem, we know that the only way of doing this would be to use more data samples in the DFT.

This technique may be incorporated into the first two bandwise techniques. After the possible application of a tapered window to each block of data, the blocks (initially of length T_b) are zero-padded such that the original data length of T samples is restored. The zero-padded blocks are then Fourier transformed. As in the bandwise algorithm, for each block and each frequency, these Fourier transforms are multiplied to give an estimate of the covariance matrix; then this estimate is improved by averaging over the blocks. Now we have estimates of the covariance matrix at a set of frequencies. However, unlike in the bandwise algorithm, the set of frequencies is not a reduced set but the set of all the frequencies present in the data. The paraunitary matrix is constructed as a unitary matrix at a set of frequencies, as in the bandwise algorithm. However, now the SVD produces a unitary matrix at each frequency, instead of at the closest frequency in the reduced set. The two variants of the bandwise algorithm discussed in this and the previous paragraph will not be evaluated due to time constraints.

4.5 PEVD Performance Analysis

4.5.1 Characterisation of the Bandwise Algorithm

The bandwise algorithm has a single parameter: the block length T_b, which is the number of design frequencies to use. The choice of T_b has a major influence on the algorithm's decorrelation performance. Residual correlations between output signals from the bandwise algorithm with an optimal block length T_b' are due to the same factors that typically limit the accuracy of non-parametric spectral estimators, in particular the Bartlett method [28]. Let the true polynomial covariance functions of the signals have finite support $2T_c+1$; covariance terms outside the range $[-T_c, T_c]$ are zero, as in (4.9). Suppose that $T_c << T$, where as before T is the finite data length. As discussed in section 4.2.4, the extent to which decorrelation has been performed can be gauged by observing the amount of energy movement from the off-diagonals, $c_{km\tau}, (k \neq m)$, to the zero-lag coefficient of the diagonals, c_{kk0}, of the true parahermitian matrix $\mathbf{C}(z)$. This quantity is given by the λ measure in (4.33). We can think of this measure for the bandwise algorithm (and other PEVD algorithms) as being governed by two factors:

Spectral (frequency) resolution. This is limited by two forms of error contributed in the mean value of the estimate $\mathbf{R}_f \approx \mathbf{A}(e^{j\omega_f})$, for $f = 1,2,\ldots,T$. Firstly, $\mathbf{R}_f$ is not separately estimated at each f, instead the estimate for the closest multiple ϕ of T_b is taken. Secondly, even at the correct value of f, the estimate is biased because it is only based on finite sample lengths (blocks

of data of length T_b). These two errors lead to a lack of spectral resolution; their contribution to λ is proportional to $1/T_b$, but independent of T.

Sample noise (error). This is due to using sample statistics rather than true statistics. Its contribution to λ is proportional to T_b/T. Hence, the shorter the data length compared to the block length, the greater the error is due to sample noise. Conversely, in the extreme case when $T_b = T$ (i.e. the naive approach), sample noise dominates the λ measure and so the designed filter bank would not be expected to perform decorrelation well in general; the algorithm would essentially fit to the data used to learn the filters[5].

In summary, there is a trade-off that has to be made between frequency resolution and sample noise because there are a fixed number of data samples to be divided between the statistical quantities (unitary matrices) for estimation. If a large number of samples are used for each unitary matrix estimate, then $T_b << T$, i.e. fewer independent quantities are estimated, and so averaging is performed across neighbouring frequencies. This results in a reduction in spectral resolution. Conversely, if there are many statistical quantities to be estimated independently (i.e. many degrees-of-freedom), the variance of the covariance matrix estimates increases. This is because the number of effective samples per quantity being estimated is small, i.e. a high sample noise scenario.

Characterisation of the SW-Bandwise Algorithm

We now consider how the analysis of λ is modified when applied to this algorithm. Recall that, for the bandwise algorithm, the second contribution to λ is due to sample errors. It follows that this source of error is considerably reduced in the SW-bandwise algorithm. This is because there are effectively more data samples used in estimating each of the T_b covariance matrices. The result of this is an improvement in the decorrelation performance of the filter bank. The algorithm is subject to errors that limit its spectral resolution in the same way that the regular bandwise algorithm is. The cost of this improvement in performance is an increase in computation time since there are more blocks of data to be obtained and used in the estimation process.

Another advantage this algorithm has over the regular bandwise algorithm is that the extent of spectral leakage is reduced because of the application of a tapered window to the

[5] Note that errors due to sample noise would not be apparent if sample statistics were used to obtain the diagonalisation performance for a single realisation of the given problem.

blocks. As with the IFB approaches and the regular bandwise algorithm, this algorithm has the problems of phase discontinuity and restricted decorrelation.

Characterisation of the WCD-Bandwise Algorithm

There are many possible choices of window that can be used with the WCD-bandwise algorithm. We chose to use a rectangular window, as in the case of the multipath-enabled SVD technique in section 4.3.2, centred at zero lag that is zero beyond the known temporal bounds of the true signal autocorrelation. For this window, the algorithm's only parameter is the window length, T_W. For optimality, this should be made equal to the length of the true correlation function, i.e. $T_W = 2T_c + 1$. For the mixing scenario of Fig. 4.1 with white source signals assumed, $T_W = 2N_g + 1$, where N_g is the order of the FIR filters in the mixing matrix $\mathbf{G}(z)$. If we set $T_W = 2N_g + 1$, the expectation of the windowed sample covariance matrix is the true covariance matrix. This is the optimal window length to use with a rectangular window. In effect, the windowing acts as a covariance-domain noise reduction filter. Since this algorithm does not process segmented data, it does not suffer from the associated problems of IFB processing.

In the case of the regular bandwise algorithm, only products of terms within the block are used in estimating the true autocovariance terms. If we chose to use a triangular (Bartlett) window function with the WCD-bandwise algorithm it would behave in a similar fashion to the regular bandwise algorithm. By contrast, the SW-bandwise algorithm, in effect, uses some products of terms that exist outside the block as well as all of those within the block for its estimates. The WCD-bandwise algorithm uses all of the products of terms at each lag to estimate the autocovariances. Its unitary matrix estimates should, therefore, have fewer errors due to the statistical estimates than those obtained by the other two algorithm versions[6]. For a rectangular window of optimal length the expectation of the windowed sample covariance matrix is the true covariance matrix, therefore sample noise is the only contributor to λ. Therefore, the algorithm constructs a more accurate estimate of $\mathbf{H}(z)$ and could be expected in general to perform better polynomial matrix diagonalisation than the other two bandwise algorithms. This performance improvement is not mysterious, it occurs because the WCD variant tackles an easier, 'less blind' problem, made easier by the fact that T_c is known.

---

[6] The WCD-bandwise algorithm would perform in a similar way to the SW variant if it was used with a rectangular window with curved corners.

4.5.2 Experimental Results

Through example simulations, an evaluation of the three bandwise algorithms described in sections 4.4.1 to 4.4.3 and the multipath-enabled SVD detailed in section 4.3.2 is presented. The performance of the conventional SVD, which is for scalar matrices, is also assessed. The algorithms were applied to the multichannel convolutive problem illustrated in Fig. 4.1. The following experiments also serve to reveal the influence of the various parameters on the decorrelation performance of the algorithms. Examples are given that indirectly highlight the validity of the performance limitations due to phase discontinuity. Furthermore, we investigate the spectral majorisation performance of the various algorithms.

In order to test the performance of the various algorithms, simulated data was created by modelling the propagation of M signals onto M sensors, as given by (4.3). Specifically, the mixing was modelled by an $M \times M$ polynomial mixing matrix $\mathbf{G}(z)$ whose entries were FIR filters of order $N_g = 5$ with coefficients drawn randomly from a uniform distribution in the range $[-1, 1]$. The source signals were independent binary phase-shift keying (BPSK) [51] sequences (with zero mean and unit variance) for which each sample takes the value ± 1 with a probability of 1/2. Additive noise (due to the receiver) was not included in these simulations, i.e. $\sigma_w^2 = 0$. For each experiment, the average of 100 realisations (simulation runs) were used to obtain each simulated point, which gives an estimate of the expected value of a quantity.

Strong Decorrelation

Dependence on T_b. The evolution of the λ measure with block length T_b is provided for the regular bandwise and SW-bandwise algorithms in Figs. 4.7(a) and (b), respectively. These results were produced by processing three signals, which were generated as in the previous paragraph, with $T = 729$ samples of data. The results confirm the analysis given in section 4.5.1: the left hand side of the optimal (minimum λ) point is dominated by errors that limit frequency resolution, whereas the right hand side is dominated by the sample noise. It is clear from the figures that the algorithms cannot achieve a zero λ value, i.e. they cannot perform strong decorrelation (in the strictest sense), for an optimal block length. A contributory factor to this shortfall is that these two bandwise methods are oblivious to relatively small but important correlations that may exist between subbands for the reasons cited in section 4.3.2. This means that, in general, these two algorithms and most other subband methods would not be expected to perform strong decorrelation in the strict sense of the term.

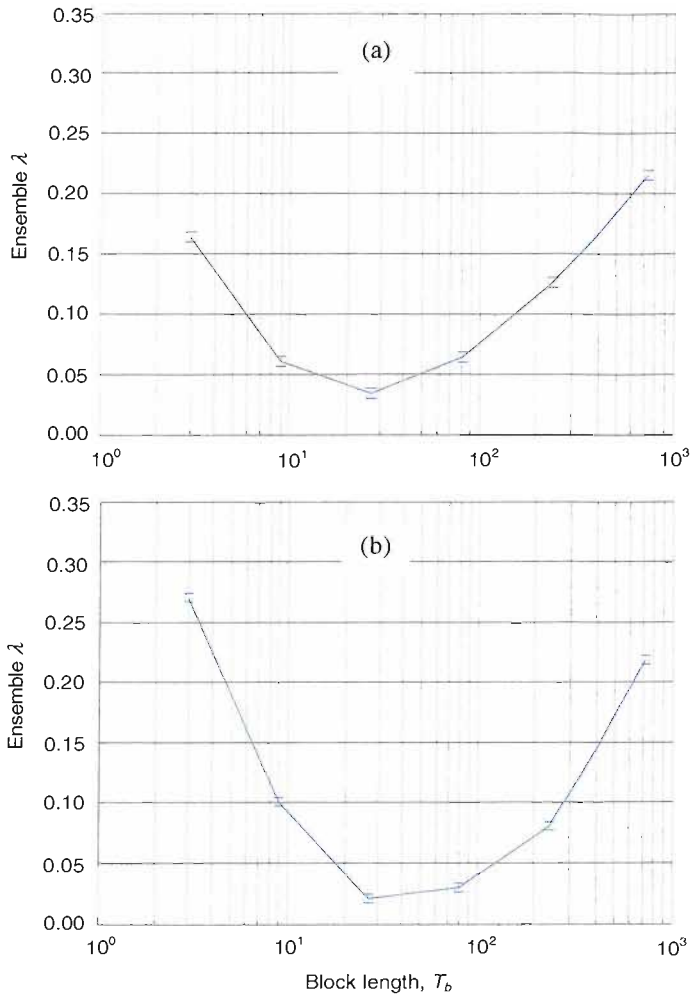


Figure 4.7: Diagonalisation performances as a function of T_b for the 3-channel (a) bandwise algorithm and (b) SW-bandwise algorithm.

Dependence on *T*. Our analysis in section 4.5.1 of the relationship between T and λ (for a fixed N_g) has been verified, and some results for the four-channel case are presented in Table 4.1. An optimal block length T_b' of $T_b = 27$ was set for the regular bandwise and SW-bandwise algorithms, this number being chosen from the previous experiment, which is optimal for $T = 729$; a maximum block overlap ($T_0 = 1$) was used with the SW-bandwise algorithm; and the correlation window length T_w was set to 11 for the WCD-bandwise algorithm.

If an optimal T_b (T_b') or T_w (depending on the algorithm) is used, the diagonalisation performance is primarily influenced by T and the true correlation time of the signals, T_c. Each independent quantity can be estimated using many samples if $T >> T_c$. However, if T_c is large compared to T, then there is insufficient data for a good estimate of $\mathbf{H}(z)$ to be found. As expected, the WCD-bandwise algorithm can generally perform better decorrelation than the

other methods. Results from using the conventional SVD and the multipath-enabled SVD are also provided in the table. It is apparent that the state-of-the-art method does poorly in comparison to the bandwise algorithms but appears to achieve greater diagonalisation than the conventional SVD. These results confirm that the SVD cannot perform strong decorrelation. The diagonality of the input parahermitian matrix is shown as reference (denoted by the symbol **I**, to signify the effective application of an identity transformation).

Data length / Algorithm	243 samples (λ)	729 samples (λ)	2187 samples (λ)
WCD-bandwise	5.5×10^{-2}	1.9×10^{-2}	6.5×10^{-3}
SW-bandwise	6.3×10^{-2}	2.8×10^{-2}	1.7×10^{-2}
Regular bandwise	1.1×10^{-1}	5.0×10^{-2}	2.5×10^{-2}
Multipath-enabled SVD	1.7×10^{-1}	1.0×10^{-1}	6.6×10^{-2}
SVD	3.5×10^{-1}	3.5×10^{-1}	3.6×10^{-1}
I (no transformation)	3.8×10^{-1}	3.8×10^{-1}	3.7×10^{-1}

Table 4.1: Comparison of the decorrelation (λ) performance of the bandwise algorithms, the multipath-enable SVD and the conventional SVD for different values of T.

Spectral Majorisation

We have observed that all versions of the bandwise algorithm produce output signals that are, in general, approximately spectrally majorised. This can be seen for the four-channel case for the SW bandwise with $T_b = 27$ and $T_0 = 1$ and the WCD-bandwise algorithm with $T_w = 11$ in Figs. 4.8(b) and (c), respectively. The four input signals used were of length $T = 2186$ samples and had spectra as in Fig. 4.8(a). Note that we have chosen to show two-sided spectra in order to show that the prior-art algorithm (in Fig. 4.9(b)) does not preserve the real nature of the signals (discussed below).

The output spectra from the regular bandwise algorithm (for an optimal block length of $T_b = T_b' = 27$) had similar characteristics to that of the SW bandwise algorithm; therefore, SW-

bandwise results are shown only. With close inspection of the figures, one can gather that the WCD-bandwise algorithm performs spectral majorisation slightly better than the other bandwise variants. A more obvious dissimilarity between the sets of output spectra is that the regular and SW-bandwise algorithms produce transformed signals that have spectra with sharp (jagged) transitions (Fig. 4.8(b)). In general the SW variant produces outputs with spectra that are smoother (less jagged features) than the output spectra from the regular bandwise algorithm. The jagged features are a sign that there is a lack of phase coherence in the output signals, that is, the temporal structure of the signals could have been corrupted somewhat. This could be due to the same problems found with the subband or IFB approaches in section 4.3.2. The jagged features are also indicative of large-order filters; this could be detrimental to many practical systems, where often the desire is to minimise the use of storage space and/or bandwidth.

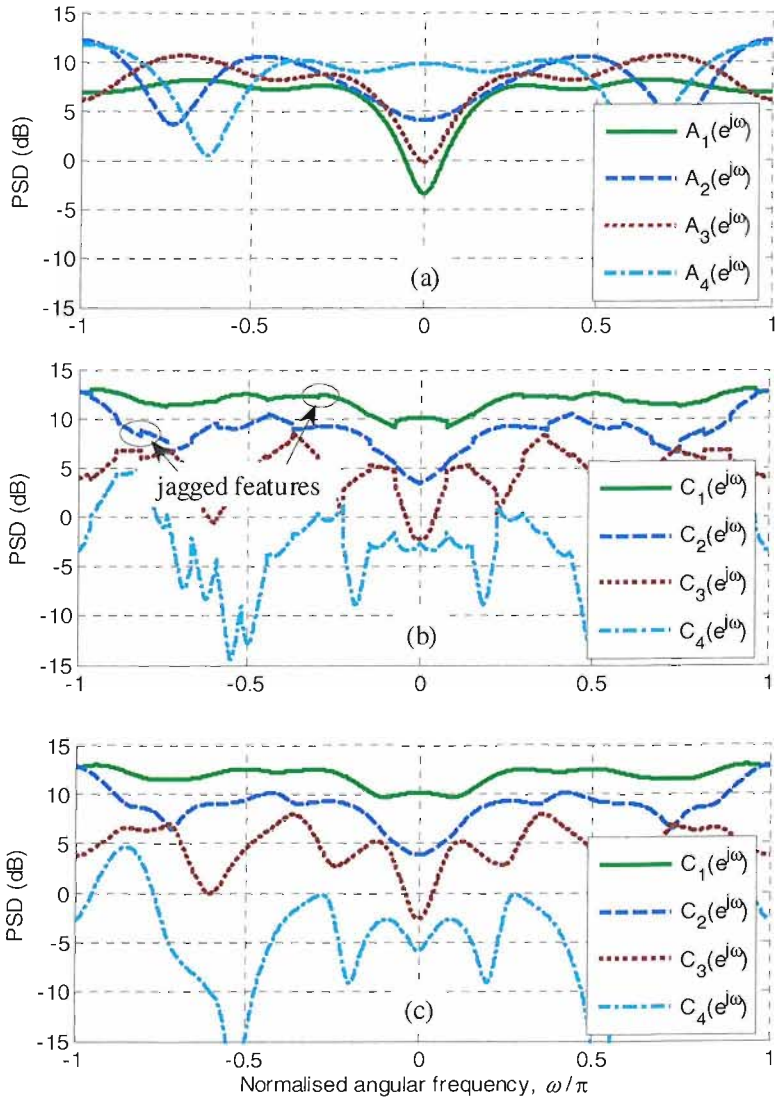


Figure 4.8: (a) Spectra of four input signals for processing. The output signal spectra of (b) the SW-bandwise algorithm and (c) the WCD-bandwise algorithm.

For comparison, the power spectra for the output signals from the (conventional) SVD and the multipath-enabled SVD are shown, respectively, in Figs. 4.9(a) and (b) for the input PSD shown in Fig. 4.8(a). We observe that the signals produced by the multipath-enabled SVD are somewhat majorised, which is in contrast to the transformed signals from the SVD. Notice that the output spectra related to the multipath-enabled SVD method are non-symmetric even though real-valued data was processed; this is an artefact introduced by the algorithm, which is due to the way it constructs its paraunitary matrix.

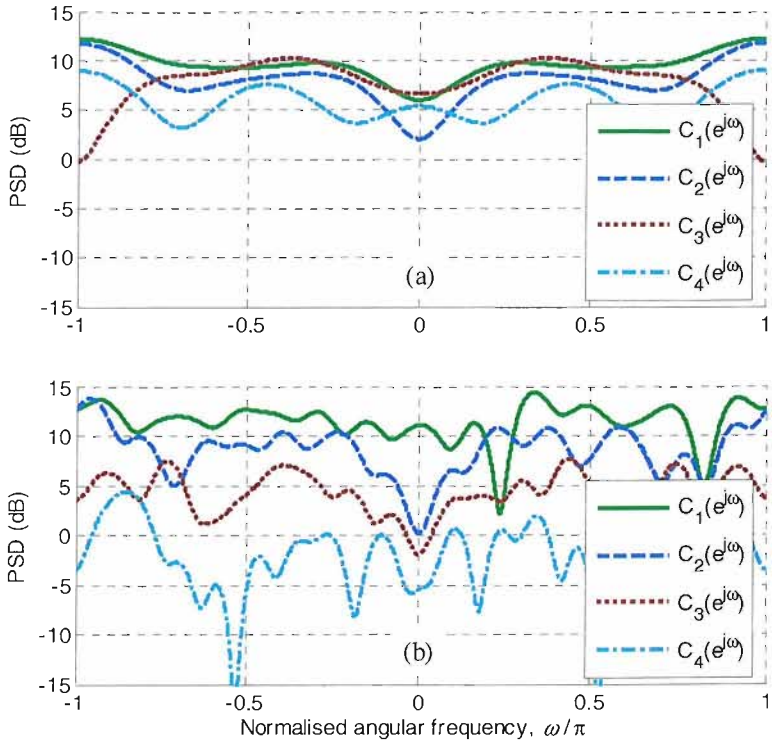


Figure 4.9: The output signal spectra of (a) the SVD and (b) the multipath-enabled SVD.

4.6 Concluding Remarks

In this chapter, the notion of a polynomial matrix EVD/SVD decomposition (PEVD/PSVD) in the context of broadband subspace estimation has been presented. A PEVD is equivalent to the conventional EVD in the sense that it can diagonalise a parahermitian matrix and order the resulting polynomial eigenvalues based on their spectra. This has been shown to be the same as producing output signals that are strongly decorrelated and spectrally majorised. A study of the

few known methods for broadband subspace decomposition has been presented and a state-of-the-art algorithm for polynomial matrix diagonalisation has been evaluated. We have proposed a family of frequency-domain PEVD algorithms, collectively known as the bandwise algorithm.

The bandwise technique essentially finds a paraunitary matrix which at each frequency is an estimate of the diagonalising unitary matrix for the set of data channels. This algorithm has been shown to give an effective extension of the SVD to convolutively mixed signals or broadband signals. Here, the role of a hidden (decorrelating) unitary matrix is extended to that of a hidden (strongly decorrelating) paraunitary matrix. We have demonstrated that the bandwise algorithm can impose strong decorrelation and spectral majorisation on a set of input signals. It also has the effect of compacting as much of the total signal power into as few channels as possible, which suggests that it could be useful in data compression applications. The extreme inadequacy of the EVD (or SVD) for the problem of broadband signals subspace estimation has been demonstrated through simulation. Also, preliminary assessment of a state-of-the-art subspace estimation method has shown that its subspace decomposition performance is inferior to that of the bandwise algorithm.

The relationship between the PEVD problem and spectral estimation has been highlighted; the factors that restrict an optimal bandwise algorithm in diagonalising a polynomial matrix (i.e. $\lambda = 0$) are those that typically limit the accuracy of spectral estimators. Applying a window to the sample space-time covariance matrix of the input data increases the diagonalisation (and thus the subspace estimation) performance of the algorithm. The windowing solves problems caused by the degenerate nature of the covariance matrix, and optimal polynomial diagonalisation may be achieved if knowledge of the true correlation time of the signals is available.

Crucially, a disadvantageous feature of the bandwise method is that the paraunitary matrix it produces is defined in the frequency-domain, i.e. it is for circulant data. Therefore, the filters generated by the algorithm are not the correct ones to use on non-circulant data. It is not clear how to obtain an equivalent paraunitary transform for non-circulant data from the filter bank generated by the bandwise algorithm. Another drawback of the bandwise algorithm is that it leads to a lack of temporal (phase) discontinuity across the bands. Long filters are required in order to realise these discontinuities, which further reduces the practicality of this technique.

5 Second-Order Sequential Best Rotation Algorithm

In this chapter, a novel technique is introduced for finding a paraunitary matrix that can impose, on a set of signals, strong decorrelation and spectral majorisation, to a good approximation. This algorithm is referred to as the second-order sequential best rotation (SBR2) [44]. It may be viewed as an extension of the EVD or SVD to polynomial matrices and can be used to obtain subspace decomposition of broadband signals. In contrast to the bandwise algorithm presented in section 4.4, the SBR2 algorithm is a time-domain PEVD technique that can design a paraunitary matrix for both circulant and non-circulant data. The development of SBR2 has dramatically improved its applicability and its performances, which are demonstrated through experimental results. The development, extensions and variations of the SBR2 algorithm presented here contributes to the original work in this thesis.

In section 5.2, we derive the SBR2 algorithm. In section 5.3, we observe important properties of the algorithm and define a new cost function through extension of the coding gain measure, given in section 3.2.1, as a means of improving the coding efficiency of the SBR2 algorithm. In section 5.4, a data pre-processing stage is introduced which involves the application of a window to the entries of the space-time polynomial covariance matrix. This improves the algorithm's diagonalisation performance and data compression ability. A strategy is given for covariance-domain windowing which guarantees a positive-semidefinite windowed space-time covariance matrix. All the relevant variations on the SBR2 algorithm are summarised in section 5.5. In section 5.6, we discuss some of the issues regarding the design and implementation of the algorithm, e.g., its computation speed. This includes replacing a slow recursive time-domain process with a more economical frequency-domain alternative. We also consider in brief the implementation and performance of the cyclic-by-rows Jacobi algorithm extended to polynomial matrices as an alternative to the classical greedy method employed by the SBR algorithm. Finally, the diagonalisation and broadband signal subspace estimation performances of the algorithm is assessed in section 5.7.

5.1 Introduction

Recall the Vaidyanathan decomposition of a paraunitary matrix given by (3.19). The construction is guaranteed to be paraunitary irrespective of the parameter values because the

design is carried out over the constrained space of paraunitary matrices. However, as demonstrated in section 3.4.1, the optimisation of the parameters of such a filter is quite difficult and requires a multi-parameter nonlinear optimisation routine, which could turn out to be highly complex and computationally costly. In this section, we develop and evaluate a novel time-domain technique as a method for simplifying these problems. The algorithm extends the EVD from conventional Hermitian matrices with complex scalar elements to parahermitian polynomial matrices, i.e. it performs PEVD. The algorithm takes the form of the sequential best rotation (SBR) algorithm [5] but only involves second order statistics, and is therefore referred to as the second-order SBR (SBR2) algorithm [44-46].

A brief reminder of the task of a 'practical' PEVD algorithm now follows. As discussed in section 4.2.1, the true polynomial covariance matrix $\mathbf{A}(z)$ (defined by (4.10)) is not usually known so the received data vector $\mathbf{x}(t)$ (equation (4.3)) must be used to generate an estimate of $\mathbf{A}(z)$. This is the sample polynomial covariance matrix $\mathbf{R}(z)$ in (4.15) given again here:

$$\mathbf{R}(z) = \frac{1}{T} \sum_{\tau=-t_1}^{t_1} \sum_{t=0}^{T-1} \mathbf{x}(t)\mathbf{x}^{\mathrm{H}}(t-\tau) z^{-\tau}, \tag{5.1}$$

where $t_1 \geq 0$ and it is assumed that $T >> t_1$; see the discussion in section 4.2.1.

As with $\mathbf{A}(z)$, $\mathbf{R}(z)$ is parahermitian (this property is defined in (2.25)). A PEVD algorithm can be used to construct a paraunitary matrix $\mathbf{H}(z)$ that transforms the parahermitian matrix $\mathbf{R}(z)$ according to (4.18); to reiterate,

$$\mathbf{S}(z) = \mathbf{H}(z)\mathbf{R}(z)\tilde{\mathbf{H}}(z), \tag{5.2}$$

where $\mathbf{S}(z)$ is approximately diagonal, i.e. $\mathbf{S}(z) \cong \mathrm{diag}[S_1(z), S_2(z), \ldots S_M(z)]$, and the set of diagonals have, to a good approximation, the spectral majorisation property defined in (4.14): $S_{kk}(e^{\mathrm{j}\omega}) \geq S_{(k+1)(k+1)}(e^{\mathrm{j}\omega})$, $k = 1,2,\ldots,M-1$ and $\forall \omega$. The matrix $\mathbf{S}(z)$ is a reliable estimate of the true polynomial covariance matrix $\mathbf{C}(z)$ in (4.12).

5.2 Derivation of the SBR2 Algorithm

The SBR2 algorithm constructs a paraunitary matrix that may be expressed as, up to a channel negation,

$$\mathbf{H}(z) = \mathbf{P}_L(z)\mathbf{P}_{L-1}(z)\ldots\mathbf{P}_i(z)\ldots\mathbf{P}_2(z)\mathbf{P}_1(z) \in \mathbb{C}^{M\times M}(z), \tag{5.3}$$

where the polynomial matrices $\mathbf{P}_i(z) \in \mathbb{C}^{M\times M}(z)$ are *elementary paraunitary matrices*:

$$\mathbf{P}_\ell(z) = \mathbf{Q}^{(n,p)}(\theta_\ell, \varphi_\ell)\boldsymbol{\Lambda}^{(p,\tau_\ell)}(z), \tag{5.4}$$

with specific values being chosen for the parameters $n, p, \tau_\ell, \theta_\ell, \varphi_\ell$. The scalar matrix $\mathbf{Q}^{(n,p)}(\theta_\ell, \varphi_\ell) \in \mathbb{C}^{M\times M}$ is a rank-two correction to the identity matrix at the intersection of the rows n and p with the columns n and p and takes the form

$$\mathbf{Q}^{(n,p)}(\theta_\ell, \varphi_\ell) = \begin{pmatrix} \mathbf{I}_{n-1} & & & & \mathbf{0} \\ & c_\ell & & s_\ell e^{\mathrm{j}\varphi_\ell} & \\ & & \mathbf{I}_{p-n-1} & & \\ & -s_\ell e^{-\mathrm{j}\varphi_\ell} & & c_\ell & \\ \mathbf{0} & & & & \mathbf{I}_{M-p} \end{pmatrix}, \tag{5.5}$$

where $c_\ell = \cos\theta_\ell$, $s_\ell = \sin\theta_\ell$, and $\{\theta_\ell, \varphi_\ell\} \in \mathbb{R}$ are rotation angles. The polynomial matrix $\boldsymbol{\Lambda}^{(p,\tau_\ell)}(z) \in \mathbb{R}^{M\times M}(z)$ is a delay matrix of the form

$$\boldsymbol{\Lambda}^{(p,\tau_\ell)}(z) = \begin{pmatrix} \mathbf{I}_{p-1} & & \mathbf{0} \\ & z^{-\tau_\ell} & \\ \mathbf{0} & & \mathbf{I}_{M-p} \end{pmatrix}. \tag{5.6}$$

where $\tau_\ell \in \mathbb{Z}$ is an arbitrary delay. The delay matrix imposes a τ_ℓ-fold delay to the p^{th} row of the polynomial matrix on which it operates[1].

Any polynomial matrix constructed according to (5.3) is paraunitary because each term is paraunitary. The paraunitary matrix $\mathbf{P}_\ell(z)$ is elementary in the sense that it only involves one rotation, but it does not necessarily have degree one. Since the parameters τ_ℓ may take on any integer, the degree of the paraunitary matrix in $\mathbf{H}(z)$ is no longer certain.

The SBR2 algorithm iteratively constructs a paraunitary matrix as in (5.3) by computing and applying a sequence of elementary paraunitary matrices designed to diagonalise a parahermitian matrix as much as possible. Each stage of the iterative process applies a single elementary paraunitary matrix, chosen to eliminate the off-diagonal polynomial coefficient with the highest magnitude square value (i.e. the dominant cross-term); this is clearly analogous to a single step of the classical Jacobi algorithm for diagonalising conventional Hermitian matrices.

---

[1] Note that in the case where $\tau_1 = 0$ and $\tau_\ell = -1$ for $1 < \ell \leq L$, equation (5.3) reduces to the familiar decomposition in (3.19). It is also worth noting that in the degenerate case where $L = 1$ and $\tau_1 = 0$, the matrix $\mathbf{H}(z)$ takes the form of a single rotation matrix that is found by the conventional EVD/SVD for the diagonalisation of a Hermitian matrix.

The algorithm begins by searching through the entries $r_{km}(\tau)=\sum_{t=0}^{T-1}x_k(t)x_m^*(t-\tau)/T$, $k,m\in\{1,2,...M\}$, of the polynomial covariance matrix in (5.1) for the largest modulus-square off-diagonal. This is a two-dimensional search in both space and time: a search considering off-diagonal terms corresponding to all possible channel pairs (i.e. a spatial search) and for a range of relative time delays between each signal-pair (i.e. a temporal search)[2].

Assume that the first search has produced $r_{np}(\tau_1)$ as the dominant coefficient, where $n<p$. The specific values n, p and τ_1, which define the dominant cross-term, are now used to specify the corresponding parameters of the first elementary paraunitary matrix, $\mathbf{P}_1(z)$, in (5.3). In accordance with (5.2), the elementary delay matrix is applied first to generate the transformed polynomial matrix:

$$\mathbf{R}'(z)=\mathbf{\Lambda}^{(p,\tau_1)}(z)\mathbf{R}(z)\tilde{\mathbf{\Lambda}}^{(p,\tau_1)}(z). \tag{5.7}$$

The effect of this transformation is to shift in time the largest cross-term $r_{np}(\tau_1)$ to the plane of order zero so that $r'_{np}(0)=r_{np}(\tau_1)=r_{pn}(-\tau_1)^*=r'_{pn}(0)^*$.

The parameters θ_1 and φ_1 of the Jacobi rotation matrix $\mathbf{Q}^{(n,p)}(\theta_1,\varphi_1)$ are now chosen to drive the dominant coefficient to zero. More specifically, they are chosen such that

$$\begin{pmatrix} c & se^{j\varphi_1} \\ -se^{-j\varphi_1} & c \end{pmatrix}\begin{pmatrix} r'_{nn}(0) & r'_{np}(0) \\ r'_{pn}(0) & r'_{pp}(0) \end{pmatrix}\begin{pmatrix} c & -se^{j\varphi_1} \\ se^{-j\varphi_1} & c \end{pmatrix}=\begin{pmatrix} r''_{nn}(0) & 0 \\ 0 & r''_{pp}(0) \end{pmatrix}. \tag{5.8}$$

This condition is satisfied when

$$\theta_1=\frac{1}{2}\tan^{-1}\left(\frac{2|r'_{np}(0)|}{r'_{nn}(0)-r'_{pp}(0)}\right) \tag{5.9}$$

and

$$\varphi_t=\arg(r'_{np}(0)). \tag{5.10}$$

Having computed the values of θ_1 and φ_1, the rotation matrix is applied thus:

---

[2] The search may be restricted to the upper (or lower) off-diagonal elements due to the parahermitian property defined in (2.25). The search in time for the optimum delay, τ_t, may be performed for a range of time delays (a delay set), where the maximum delay in the set is less than or equal to the total number of data samples, T. If the dominant coefficient is not unique, any one from the set of coefficients with largest (equal) magnitudes may be chosen.

$$\mathbf{R}''(z) = \mathbf{Q}^{(n,p)}(\theta_1, \varphi_1)\mathbf{R}'(z)\tilde{\mathbf{Q}}^{(n,p)}(\theta_1, \varphi_1) . \tag{5.11}$$

It should be clear that the polynomial matrices $\mathbf{R}(z)$ and $\mathbf{R}''(z)$ are related by the generalised similarity transformation:

$$\mathbf{R}''(z) = \mathbf{P}_1(z)\mathbf{R}(z)\tilde{\mathbf{P}}_1(z) . \tag{5.12}$$

The elementary paraunitary transformation in (5.12) constitutes one stage of the SBR2 algorithm designed to zero the dominant off-diagonal coefficient of $\mathbf{R}(z)$. The algorithm continues by making the substitution $\mathbf{R}(z) \leftarrow \mathbf{R}''(z)$ and repeating the process outlined above, i.e. applying a new elementary paraunitary transformation of the form given in (5.12) designed, this time, to zero the most significant coefficient of the updated polynomial matrix $\mathbf{R}(z)$. In practice, this iterative process is repeated until the magnitude of the dominant off-diagonal coefficient of $\mathbf{R}(z)$ is sufficiently small, say, smaller than or equal to some constant $\varepsilon_s > 0$: stopping criterion; at which point the polynomial matrix is declared to be diagonal. Assuming L iterations, the result will be a generalised similarity transformation of the form

$$\mathbf{H}_L(z)\mathbf{R}(z)\tilde{\mathbf{H}}_L(z) = \mathbf{S}_L(z) . \tag{5.13}$$

It is worth noting that the estimated polynomial covariance matrix in (5.1) is updated as part of the SBR2 algorithm and does not need to be recomputed from the transformed data. A flowchart of the SBR2 algorithm is shown in Fig. 5.1. The paraunitary matrix $\mathbf{H}(z)$ may be updated at each step of the algorithm or constructed afterwards: Initialisation: $\mathbf{H}'_0(z) := \mathbf{I}$; Iteration ℓ: $\mathbf{H}'_\ell(z) \leftarrow \mathbf{P}_\ell(z)\mathbf{H}'_{\ell-1}(z)$; Output: $\mathbf{H}_L(z) \leftarrow \mathbf{H}'_L(z)$. It should be noted that the paraunitary matrix found by this algorithm is unique, up to the application of all-pass filtering.

The paraunitary matrix $\mathbf{H}(z)$ can be applied to the input signals $\underline{X}(z)$, used to generate $\mathbf{R}(z)$ in (5.1), stage by stage during the SBR2 computation. This is achieved with the application of the elementary paraunitary matrices $\mathbf{P}_\ell(z)$ contributing to (5.3) to the input signals in the manner

$$\underline{Y}(z) = \mathbf{H}_L(z)\underline{X}(z) = \mathbf{P}_L(z)\mathbf{P}_{L-1}(z)\ldots\mathbf{P}_\ell(z)\ldots\mathbf{P}_2(z)\mathbf{P}_1(z)\underline{X}(z) . \tag{5.14}$$

Equation (5.14) is represented for the two-channel case by a flow diagram in Fig. 5.2. In words, at the ℓ^{th} stage, $\mathbf{P}_\ell(z)$ may be applied to the transformed signals produced as a result of the first $\ell - 1$ stages, in order to generate the output data from stage ℓ. This constitutes a sequence of pairwise delay-and-rotate operations since each elementary paraunitary matrix only affects two of the signal channels.

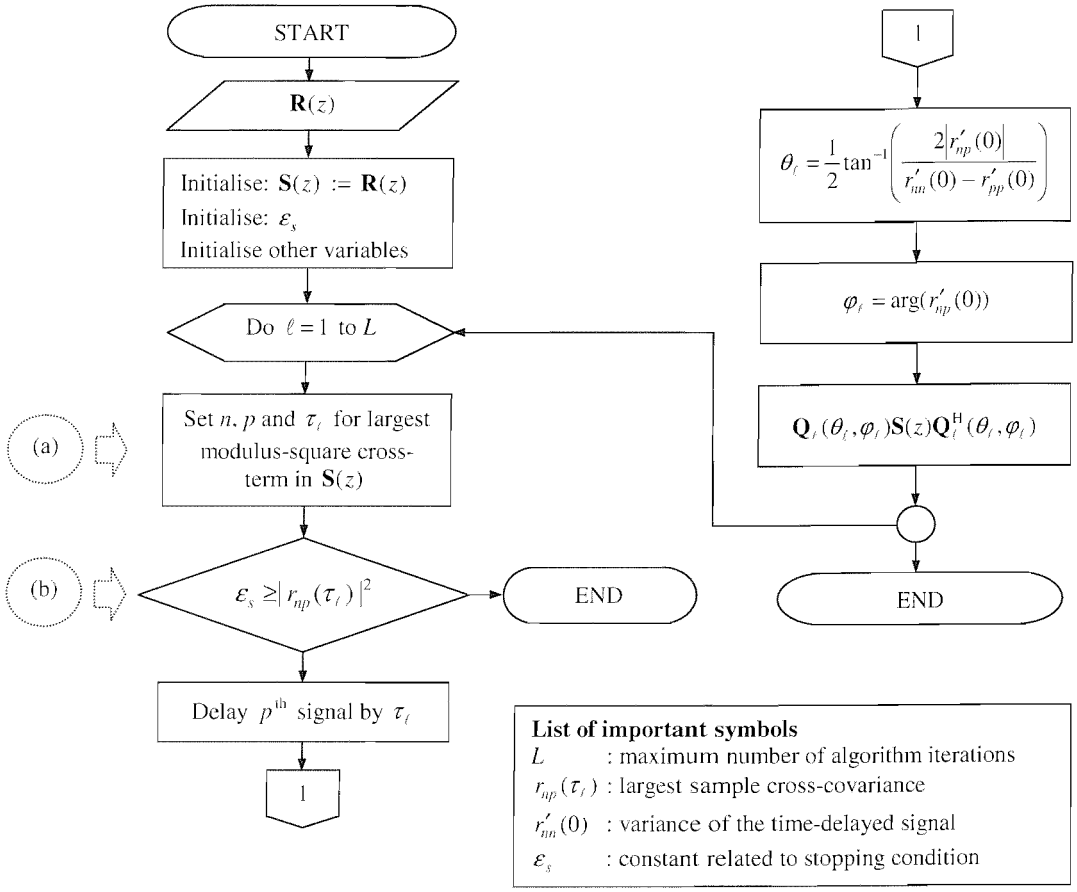


Figure 5.1: Flowchart of the SBR2 algorithm.

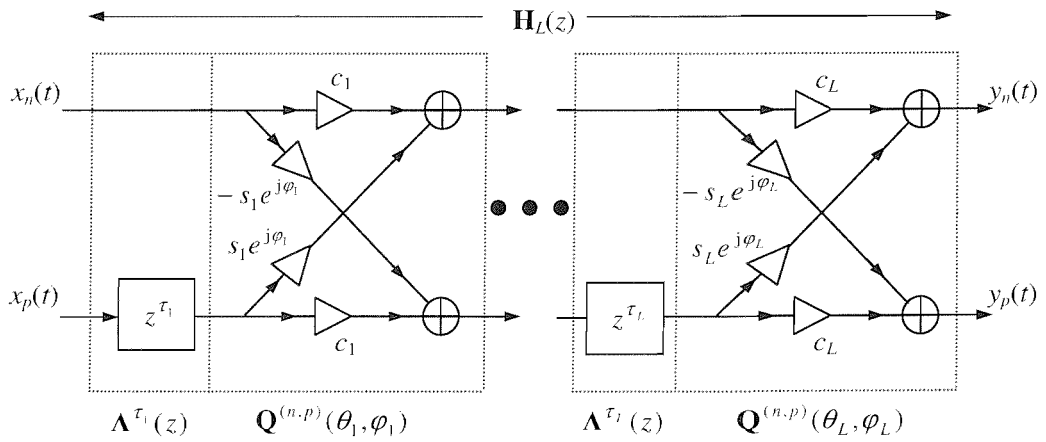


Figure 5.2: A diagrammatic representation of a cascade of a 2×2 paraunitary lattice filter consisting of L *elementary* paraunitary blocks.

At first sight, the strategy outlined above might not seem sensible since the successive elementary paraunitary matrices do not commute and applying a rotation does not just affect the current state but also the potential future gains of the algorithm. Unlike the narrowband case, applying a poorly chosen rotation is likely to make the problem more difficult by increasing the order of the mixing polynomial for no good reason. However, the freedom to choose an optimum delay for each stage makes this process much more meaningful.

A SBR2 algorithm that operates directly on the data (i.e. a data-domain algorithm) is not defined in this thesis. There are two reasons for this: (1) such an algorithm would be computationally less efficient than the covariance-domain algorithm given here, as explained later in section 5.6.2; (2) a better approximation to strongly decorrelated signals can be gained by working with a windowed version of the space-time covariance matrix, as discussed later in section 5.4.

5.3 Cost Functions

5.3.1 Cost Function Based on Correlation

In order to explain how the SBR2 algorithm achieves its objective, we introduce the following set of measures:

$$N_1 = \sum_{n=1}^{M} |r_{nn}(0)|^2 \tag{5.15}$$

$$N_2 = \sum_{n=1}^{M}\sum_{p=1}^{M} |r_{np}(0)|^2 , \tag{5.16}$$

$$N_3 = \sum_{n=1}^{M}\sum_{\substack{p=1 \\ p\neq n}}^{M} |r_{np}(0)|^2 \tag{5.17}$$

and

$$N_4 = \sum_{\tau}\sum_{n=1}^{M}\sum_{p=1}^{M} |r_{np}(\tau)|^2 . \tag{5.18}$$

Similarly, we define the sets $\{N_1', N_2', N_3', N_4'\}$ and $\{N_1'', N_2'', N_3'', N_4''\}$ for $\mathbf{R}'(z)$ and $\mathbf{R}''(z)$. It is easy to see that N_1 is a local measure of the instantaneous autocovariance associated with the signals while N_3 is the corresponding measure of instantaneous crosscovariance between the signals. The quantities N_2 and N_4 constitute the squares of the Frobenius norm of the

instantaneous covariance matrix $\mathbf{r}(0)$ and the space-time covariance matrix $\mathbf{r}(\tau)$, respectively. Note the following important properties:

- N_1 is preserved under the application of a delay matrix;
- N_2 is invariant to the application of a rotation matrix;
- $N_3 = N_2 - N_1$;
- N_4 is preserved under the application of an elementary paraunitary matrix;
- $N_1 \leq N_4$.

After applying the delay matrix $\mathbf{\Lambda}^{(p,\tau_i)}(z)$ to generate $\mathbf{R}'(z)$ in the procedure outlined above, we have $N_1' = N_1$ and $N_4' = N_4$ with new values for N_2' and N_3'. The effect of applying the rotation matrix $\mathbf{Q}^{(n,p)}(\theta_i, \varphi_i)$ in the procedure outlined above is to reduce N_3' by an amount $2g$, where $g = \left|r_{np}'(0)\right|^2$. By construction, this is the maximum value of $\left|r_{km}(\tau)\right|$ taken over all off-diagonal coefficients in $\mathbf{R}(z)$. Since N_2' is unchanged by the application of $\mathbf{Q}^{(n,p)}(\theta_i, \varphi_i)$, it follows that

$$\begin{aligned} N_1'' &= N_1' + 2g = N_1 + 2g \\ N_2'' &= N_2' \\ N_3'' &= N_3' - 2g \\ N_4'' &= N_4' = N_4 \end{aligned} \tag{5.19}$$

The result of each algorithm step (iteration) is to increase N_1 by $2g$ which constitutes the magnitude squared of the greatest off-diagonal polynomial coefficient. This leads naturally to the following assertion.

Theorem 5.1. *The maximum magnitude squared off-diagonal term of the parahermitian polynomial matrix* $\mathbf{R}(z)$ *tends to zero, that is the SBR2 algorithm is guaranteed to converge.*

Proof. Since N_1 increases monotonically and is bounded from above by N_4, which is constant, it must have a supremum S. It follows that for any $\varepsilon_s > 0$ there must be an iteration number L, say, for which $\left|S - N_1\right| < 2\varepsilon_s$ and so the increase in N_1 at all subsequent stages must satisfy $2g \leq \left|S - N_1\right| < 2\varepsilon_s$. In other words, for any $\varepsilon_s > 0$ there must be an iteration by which the maximum magnitude square off-diagonal polynomial coefficient is bounded by the stopping condition constant, ε_s. Q.E.D.

Note that the value of $\sqrt{g}$ does not necessarily decline monotonically. Each rotation is computed with reference to elements in the plane of order zero in $\mathbf{R}'(z)$, and is guaranteed to

increase N_1' by driving the (maximised) off-diagonal element to zero, thus reducing N_3'. However, in any other plane of the polynomial matrix $\mathbf{R}'(z)$, where the same rotation is being applied, it could have the effect of increasing the magnitude of the off-diagonal element whilst reducing the sum of the squares of the diagonal elements. As a result, the dominant off-diagonal element, taken over all values of τ, could be larger at the start of the next iteration[3]. The algorithm does not seek to reduce the on-diagonal coefficients for non-zero values of τ, let alone drive them to zero. In the context of strong decorrelation, this would correspond to temporal whitening of the decorrelated signals, which is often highly undesirable and cannot occur as the result of a paraunitary transformation (which preserves the total PSD). Note also that the quantity N_1 may be viewed as a *cost function* which the algorithm maximises by minimising the measure N_3. This algorithm is, therefore, referred to as the SBR2(N_1) algorithm. We will now illustrate the operation of this algorithm by means of some simple but insightful examples.

Example 1. The SBR2(N_1) algorithm was first applied to the parahermitian matrix given by:

$$\mathbf{R}(z) = \begin{pmatrix} 1 & -0.4\mathrm{j}z & 0 \\ 0.4\mathrm{j}z^{-1} & 1 & 0.5z^{-2} \\ 0 & 0.5z^{2} & 1 \end{pmatrix}. \tag{5.20}$$

In seven iterations, it converged to produce the following factorisation:

$$\mathbf{H}(z) = \begin{pmatrix} 0.4417 & -0.7071\mathrm{j}z & -0.5522\mathrm{j}z^{-1} \\ -0.7809\mathrm{j} & 0 & 0.6247z^{-1} \\ 0.4417 & -0.7071z & 0.5522z^{-1} \end{pmatrix} \text{ and } \mathbf{S}(z) = \begin{pmatrix} 1.6403 & 0 & 0 \\ 0 & 1.0000 & 0 \\ 0 & 0 & 0.3597 \end{pmatrix}. \tag{5.21}$$

The final value of g was zero (to computational precision). The value of N_1 increased from 3.0000 to 3.8200, which, in this case, is equal to the value of N_4. This reflects the fact the sum of the squares of the diagonal elements of $\mathbf{S}(z)$ was also zero except in the plane of order zero. Since $\mathbf{H}(z)$ is paraunitary, the inverse decomposition is given very simply by $\mathbf{R}(z) = \widetilde{\mathbf{H}}(z)\mathbf{S}(z)\mathbf{H}(z)$. The Frobenius norm between the result of this computation and the original matrix $\mathbf{R}(z)$ was zero (to computational precision).

This example is particularly simple in the sense that the matrix $\mathbf{R}(z)$ can be reduced to a scalar matrix by initially applying two successive delay transformations of the type specified in (5.7). The problem then reduces to one of standard Jacobi diagonalisation. Note, however, that this "trick" is not exploited by the SBR2 algorithm which performs a strict sequence of

[3] The difference between N_4 and the supremum of N_1 will generally be non-zero.

alternating delay and rotate operations as specified in section 5.2. Despite the simplicity of this example, it is worth bearing in mind that the same decomposition would be much more complicated if carried out in the frequency domain.

Example 2. The SBR2(N_1) algorithm was applied to the parahermitian matrix given by:

$$\mathbf{R}(z) = \begin{pmatrix} 1 & 0.8z^2 - 0.4z & 0.7z \\ 0.8z^{-2} - 0.4z^{-1} & 1 & 0.5z^{-2} \\ 0.7z^{-1} & 0.5z^2 & -1 \end{pmatrix}. \tag{5.22}$$

Note that the scalar reduction trick cannot be used successfully in this case. For ease of graphical representation, the coefficients of the parahermitian matrix were chosen to be real. A negative value was assigned to the (3,3) element so that the zero-lag coefficient matrix is not positive semidefinite. In this case, the SBR2 algorithm converged to a level of $\sqrt{g} < 10^{-3}$ in 37 iterations. The Frobenius norm of the off-diagonal elements of the diagonalised matrix $\mathbf{S}(z)$ was 3.4×10^{-3}, which should be compared to the total Frobenius norm given by $\sqrt{N_4} = 2.47$ and the value of $\sqrt{N_1}$ which increased from 1.73 to 2.45.

The algorithm generated the polynomial matrices $\mathbf{S}(z)$ and $\mathbf{H}(z)$ depicted in Fig. 5.3 and Fig. 5.4, respectively. These figures show a plot of the coefficients for each polynomial entry over a suitable range of values of τ. During the computation the effective order of $\mathbf{S}(z)$ grew to 101 $(-50 \le \tau \le 50)$ but we have only plotted terms for which $|\tau| \le 5$ since all terms outside that range are negligible. The order of the paraunitary matrix $\mathbf{H}(z)$ was restricted to terms for which $|\tau| \le 10$, as shown in Fig. 5.4. When the polynomial matrix $\mathbf{R}(z)$ was reconstructed by computing the inverse decomposition $\mathbf{R}(z) = \tilde{\mathbf{H}}(z)\mathbf{S}(z)\mathbf{H}(z)$ to the order of approximation represented in Fig. 5.3 and Fig. 5.4, the Frobenius norm of the difference matrix was $\sim 2\times10^{-3}$. This error is due to the truncation of $\mathbf{H}(z)$. Clearly, for this example, an excellent approximate decomposition can be implemented using polynomial matrices of very modest order.

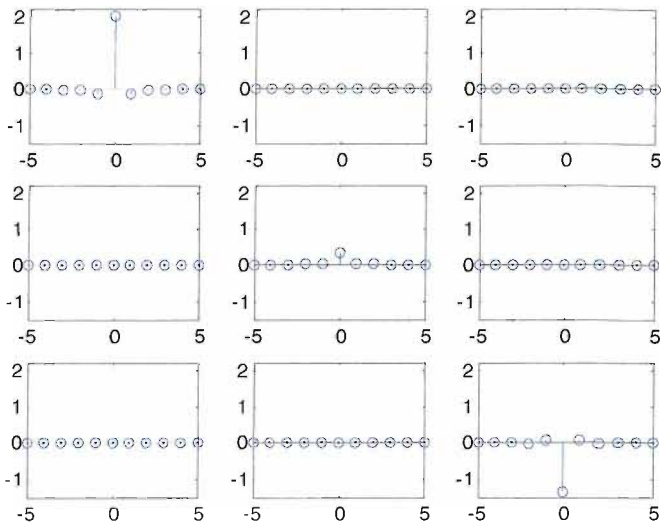


Figure 5.3: Diagonalised polynomial matrix obtained using the SBR2 algorithm for example 2.

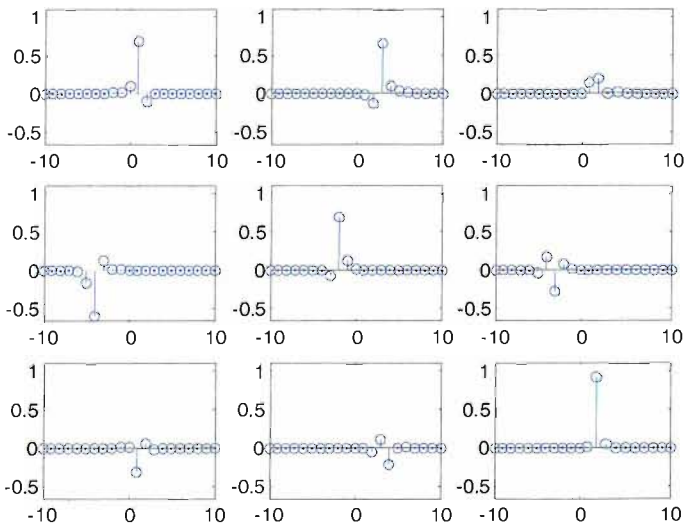


Figure 5.4: Paraunitary matrix obtained using the SBR2 algorithm for example 2.

As given by (5.14), the SBR2 algorithm can be applied directly to multichannel data. Then the transformed signals have the following important property.

Theorem 5.2. *The signals transformed by the elementary paraunitary matrices satisfy strong decorrelation, to a good approximation.*

Proof. As in (4.25), the polynomial covariance matrix is:

$$\mathbf{C}_{\underline{y}\underline{y}}(z) = \sum_{\tau} \mathrm{E}\left[\mathbf{y}(t)\mathbf{y}^{\mathrm{H}}(t-\tau)\right] z^{-\tau}. \tag{5.23}$$

It can be seen that the signals $\underline{Y}(z)$ are strongly decorrelated since

$$\mathbf{C}_{\underline{y}\underline{y}}(z) = \mathbf{H}(z)\mathbf{A}(z)\tilde{\mathbf{H}}(z) \tag{5.24}$$

and a consistent estimate of $\mathbf{C}_{\underline{y}\underline{y}}(z)$ is given by

$$\mathbf{R}_{\underline{y}\underline{y}}(z) = \mathbf{H}(z)\mathbf{R}(z)\tilde{\mathbf{H}}(z) = \mathbf{S}(z). \tag{5.25}$$

It follows from Theorem 5.1 that

$$\mathbf{R}_{\underline{y}\underline{y}}(z) \cong \mathrm{diag}[S_1(z), S_2(z), \ldots S_M(z)] \tag{5.26}$$

and so $\mathbf{C}_{\underline{y}\underline{y}}(z)$ is approximately diagonal. Q.E.D.

Another salient characteristic of the SBR2(N_1) algorithm is given as follows:

Conjecture 5.1. *The signals transformed by the elementary paraunitary matrices will tend to satisfy the spectral majorisation property, to a good approximation.*

Illustration. Consider the case of two input signals. From (5.15) we have that

$$N_1 = |r_{11}(0)|^2 + |r_{22}(0)|^2 \tag{5.27}$$

where

$$r_{kk}(0) = \sum_{t=0}^{T-1} |y_k(t)|^2 = \sum_{f=0}^{T-1} |Y_k(e^{\mathrm{j}2\pi ft})|^2, \tag{5.28}$$

as from Parseval's theorem. It follows from (4.32) that

$$r_{11}(0) + r_{22}(0) = \kappa \tag{5.29}$$

where κ is a constant determined by the input signals. Assuming the diagonals are to be ordered as defined in (4.14) and expressing (5.27) as

$$N_1 = \kappa^2 - 2r_{11}(0)r_{22}(0), \tag{5.30}$$

we see that N_1 increases to its maximum value as $r_{11}(0) \to \kappa$ and $r_{22}(0) \to 0$, i.e. the smaller the value of $r_{22}(0)$ the larger the value of N_1. Since the SBR2 algorithm maximises the quantity N_1, therefore minimising $r_{22}(0)$, and since (5.28) holds for $k = 2$, it intrinsically strives to perform spectral majorisation.

Needless to say, this does not constitute a proof of spectral majorisation for the SBR2 algorithm. In fact it is easy to see that two signals in non-overlapping subbands (or two different

sinusoids), whilst not spectrally majorised, will be strongly decorrelated from the outset, so the SBR2 algorithm can make no improvement. However, situations such as this rarely occur in practice and can be circumvented by means of a very small (paraunitary) perturbation of the input parahermitian matrix.

Note that in practice, the SBR2 algorithm only has a finite number of degrees-of-freedom. Therefore, the set of transformed signals, in general, does not strictly have the spectral majorisation property; it is a good approximation to a majorised set. If the algorithm were allowed to carry out an infinite number of stages then spectral majorisation would be satisfied by the transformed signals.

It may now be clear from the discussions in this sub-section and the definition of the PEVD in section 4.2.1 that the SBR2 algorithm provides an estimated PEVD of a parahermitian polynomial matrix.

5.3.2 Cost Function Based on the Coding Gain

There is an important limitation of the search strategy described in section 5.3.1. This is that the SBR2 algorithm tends to strongly decorrelate signals with large power and ignore signals that have relatively lower power. In other words, the SBR2(N_1) is proportionately more sensitive to changes in dominant signals than it is to changes in weak signals. This is because, in general, the largest cross-correlation coefficients occur between dominant signals, which are minimised by the algorithm at every step. The detrimental effect this has on the performance of the algorithm is twofold:

1. The extent to which polynomial diagonalisation (strong decorrelation) is carried out is restricted: After a number of iterations, the algorithm begins to zero noise-related cross-correlations between dominant signals rather than signal-related (true) cross-correlations between weaker signals.
2. The extent to which spectral majorisation is performed is limited: Energy in weaker, correlated signals is not compacted into as few channels as possible (lack of energy compaction). This is usually because energies due to cross-correlation terms, which are spread among pairs of weak signals, are not transferred to the auto-correlation (PSD) of the signals.

These problems can be alleviated by the use a cost function which is proportionately, equally sensitive to changes in any of the signals. A cost function that is based on the coding gain measure in (3.7) for subband coding would have this property. Hence, we define a new objective function simply by substituting quantities based on sample statistics for those in the expression for the coding gain. In other words,

$$G_c = \frac{1}{2}\log_2\left(\frac{\prod_{k=1}^{M} r_{kk0}}{\prod_{k=1}^{M} r''_{kk0}}\right). \tag{5.31}$$

where r_{kk0} and r''_{kk0} are, respectively, the diagonals of the zero-lag matrices $\mathbf{r}(0)$ and $\mathbf{r}''(0)$ related by (5.12). As with the coding gain, minimisation of the denominator (product of the transformed signal variances) leads to maximal energy compaction and spectral majorisation for paraunitary transformations. The SBR2 algorithm with this cost function is appropriately referred to as the SBR2(G_c) algorithm. We now make the following assertion.

Theorem 5.3. *The G_c cost function is maximised at each step of the SBR2 algorithm if the largest* ***normalised*** *magnitude squared off-diagonal term in $\mathbf{r}(\tau)$ is zeroed.*

Proof. Let r_{nn0} and r_{pp0} denote the sample variances of the n^{th} and p^{th} signals, respectively, prior to the application of an elementary paraunitary transformation of the form (5.4). Further consider that the numerator of (5.31) can be treated as a constant κ_1 during an iteration of the algorithm. Also, since the channels n and p are the only channels modified by the transformation, the product of all of the sample variances of the transformed output channels, except the n^{th} and p^{th}, is a constant, say κ_2. We may now write the G_c cost function as

$$G_c = \frac{1}{2}\log_2\left(\frac{\kappa_1}{\kappa_2 r_{nn0} r_{pp0}}\right) = \frac{1}{2}\log_2\left(\frac{1}{r_{nn0} r_{pp0}}\right) + \kappa, \tag{5.32}$$

where $\kappa = (1/2)\log_2(\kappa_1/\kappa_2)$. The change in G_c, denoted ΔG_c, due to an iteration of the algorithm can be expressed as

$$\Delta G_c = \frac{1}{2}\log_2\left(\frac{r_{nn0} r_{pp0}}{r''_{nn0} r''_{pp0}}\right), \tag{5.33}$$

where r''_{nn0} and r''_{pp0} are the lag-zero off-diagonal terms of the transformed space-time covariance matrix $\mathbf{r}''(\tau)$ in (5.12). Equation (5.33) is maximised with the application of an appropriate relative delay τ_i and rotation θ_i to the n^{th} and p^{th} signals:

$$J = \underset{n,p,\theta_i,\tau_i}{\arg\max}\left(\frac{r_{nn0} r_{pp0}}{r''_{nn0} r''_{pp0}}\right). \tag{5.34}$$

Let $\mathbf{r}'(\tau)$ be the resulting space-time covariance matrix after the application of the delay matrix in (5.7). The denominator of the quotient in (5.34) may be expressed in terms of the diagonal coefficients r'_{nn0} and r'_{pp0} and the off-diagonal r'_{np0} on the $\tau = 0$ plane of $\mathbf{r}'(\tau)$:

$$\arg\max_{n,p,\theta_\ell,\tau_\ell}\left(\frac{r_{nn0}r_{pp0}}{r''_{nn0}r''_{pp0}}\right) = \arg\max_{n,p,\tau_\ell}\left(\frac{r_{nn0}r_{pp0}}{r'_{nn0}r'_{pp0}-\left|r'_{np0}\right|^2}\right). \tag{5.35}$$

But the right-hand side of (5.35) is:

$$\arg\max_{n,p,\tau_\ell}\left(\frac{r_{nn0}r_{pp0}}{r_{nn0}r_{pp0}-\left|r_{np\tau_\ell}\right|^2}\right) = \arg\max_{n,p,\tau_\ell}\left(\frac{1}{1-\dfrac{\left|r_{np\tau_\ell}\right|^2}{r_{nn0}r_{pp0}}}\right). \tag{5.36}$$

Therefore, the objective in (5.34) is equivalently given by

$$J = \arg\max_{n,p,\tau_\ell}\left(\frac{\left|r_{np}(\tau_\ell)\right|^2}{r_{nn0}r_{pp0}}\right). \tag{5.37}$$

The denominator of (5.37) can be viewed as a normalisation factor that essentially stabilises the algorithm. In essence, this is because large cross-covariance coefficients due to, say, strong signals are attenuated relative to those associated with weaker signals. Hence, the maximisation of G_c entails a generalised classical Jacobi search for the largest normalised cross-covariance term. Q.E.D.

There are two modifications that need to be made to the SBR2 algorithm in order to obtain the SBR2(G_c) algorithm. The first is that the correlation based objective function, which is a function of $\left|r_{np}(\tau_\ell)\right|^2$, is replaced with the quotient in (5.37). Secondly, the stopping criterion ε_s is in terms of the quotient in (5.37) rather than $\left|r_{np}(\tau_\ell)\right|^2$. These two modifications are indicated, respectively, by the circles labelled as (a) and (b) in Fig. 5.1. These are the only modifications that are required to the algorithm described in section 5.3.1 for implementation of the SBR2(G_c) algorithm.

The cost function proposed here improves the strong decorrelation and spectral majorisation performances of the SBR2 algorithm. The satisfaction of these conditions is necessary for optimal data compression and broadband subspace decomposition. Hence, the SBR2(G_c) algorithm is more suited to these applications than its correlation-based counterpart. Note that the algorithm intrinsically aims to design a filter bank that is optimal for multichannel

data compression because its optimisation is exclusively geared towards the maximisation of energy compaction. It is not easy to see how one can obtain a proof of convergence for the SBR2(G_c) algorithm, since the condition $J = \infty$ will arise if r_{inn0} or r_{ipp0} are zero.

5.3.3 Cost Function Based on True Statistics

As a means of producing an upper bound on the performance for the SBR2 algorithm, we present a SBR2 variant that exploits knowledge about the true statistics of the input data. As a consequence, it gives near ideal performance. It is from a class of 'non-blind' SBR2 algorithms. Essentially, this technique is allowed to operate on the true covariances as opposed to the sample covariances. Correspondingly, the cost function used by this algorithm differs from those discussed thus far.

This algorithm is based on the SBR2(N_1) algorithm in the sense that it also finds the largest off-diagonal term in the given matrix at each step. Since the operations are performed on $\mathbf{C}(z)$, the cost function is not N_1, but is given by

$$\lambda' = \sum_{k=1}^{M} \left| c_{kk0}^2 \right|, \tag{5.38}$$

where c_{kk0} is the zero-lag term of the true autocorrelation function of the k^{th} signal. The quantity λ' is more suitably described as a measure rather than a cost function since true statistics are used. The algorithm described will be given the name SBR2(λ') for future reference. It finds a paraunitary matrix that is, to a good approximation (since we are dealing with FIR filters), the optimum decorrelation filter bank.

5.4 Windowed Covariance-Domain SBR2 Algorithm

A window function may be applied to the entries of the sample space-time covariance matrix $\mathbf{r}(\tau)$ in (5.1) prior to the application the SBR2 algorithm. The algorithm begins by constructing the sample parahermitian matrix $\mathbf{R}(z)$. A suitable window function, $w(\tau)$, is then applied to each of the polynomial entries of $\mathbf{r}(\tau)$, as performed by the WCD-bandwise algorithm. The windowed polynomial covariance matrix is typically given by

$$\hat{\mathbf{R}}(z) = \sum_{\tau=-t_1}^{t_1} w(\tau)\mathbf{r}(\tau)z^{-\tau}, \tag{5.39}$$

where

$$\mathbf{r}(\tau) = \sum_{t=0}^{T-1} \mathbf{x}(t)\mathbf{x}^{\mathrm{H}}(t-\tau)/T \tag{5.40}$$

and

$$w(\tau) = \begin{cases} \upsilon(\tau), & |\tau| \le t_1 \\ 0, & |\tau| > t_1 \end{cases} \tag{5.41}$$

is a window function of length $T_w = 2t_1 + 1$ with $\upsilon(\tau)$ the impulse response of a FIR filter. It is assumed that $\mathbf{r}(\tau) \cong 0$, for $|\tau| > t_1$ and that $T >> t_1$. In practice, the value of t_1 is often measured experimentally. If the correlation time interval $[-T_c, T_c]$ of the signals is known, then this is used to set the value T_w, i.e. $t_1 = T_c$ and $T_w = 2T_c + 1$. In which case, all sample correlation terms at lags greater than this maximum delay are zeroed. If $\upsilon(\tau) = 1$, then $w(\tau)$ is a rectangular window and the windowed space-time covariance matrix $\hat{\mathbf{r}}(\tau)$ is obtained simply by truncating the entries of the original space-time covariance matrix in (5.40).

The SBR2 algorithm is carried out on the windowed space-time covariance matrix $\hat{\mathbf{r}}(\tau)$. This algorithm is referred to as the windowed covariance domain (WCD) SBR2 algorithm. Typically, a rectangular window function is used prior to the application of the SBR2(N_1) algorithm. We refer to this algorithm as WCD-SBR2(N_1).

The matrix $\hat{\mathbf{r}}(\tau)$ is a more accurate estimate of the true space-time covariance matrix than the (unwindowed) sample space-time covariance matrix in (5.40), both for our purposes and in terms of sums of square errors. One consequence of this is that there is no set of data with sample variances equal to the windowed sample covariance of the original data. It follows that the WCD-SBR2 algorithm cannot be carried out in the data domain, and the covariance domain version of the algorithm must be used.

In the case of a rectangular window, the windowed space-time covariance matrix is parahermitian since $r_{km}(\tau) = r_{mk}^{*}(-\tau)$. However, this is not guaranteed for arbitrary window functions; e.g., asymmetric windows. Another important property is stated in the following.

Theorem 5.4. *The windowed parahermitian matrix is not necessarily positive semidefinite.*

Proof. This can be shown with an example. Suppose the entries of the sample space-time covariance matrix in (5.40), $r_{km}(\tau)$, are constant functions, so that the Fourier transforms $R_{km}(e^{\mathrm{j}\omega})$ are delta functions, i.e. $R_{km}(e^{\mathrm{j}\omega}) = \delta(\omega)$. Furthermore, suppose that the rectangular window function $w(\tau)$ of length $T_w = 2t_1 + 1$ is applied to $r_{km}(\tau)$. The Fourier transform of the rectangular window is a sinc function: $W(e^{\mathrm{j}\omega}) = \sin(t_1\omega)/t_1\omega = \mathrm{sinc}(t_1\omega)$. In the frequency

domain, the product of $w(\tau)$ with $r_{km}(\tau)$ is a convolution of the sinc function with the Kronecker delta function, as shown in Fig. 5.5, which gives a sinc function, i.e.

$$\hat{R}_{km}(e^{j\omega}) = W(e^{j\omega}) * R_{km}(e^{j\omega}) = \delta(\omega) * \mathrm{sinc}(t_1\omega) = \mathrm{sinc}(t_1\omega). \tag{5.42}$$

The sinc function has negative amplitudes at frequencies on and around multiples of $\omega = 3\pi/2t_1$, therefore the windowed spectra $\hat{R}_{kk}(e^{j\omega})$ will also have negative energy at these points. (Note that this, of course, could never happen for true spectra.) Since some of the terms in $\hat{r}_{kk}(\tau)$ take on negative values, it follows from the definition of positive semidefiniteness for polynomials in (4.11) that the polynomial matrix $\hat{\mathbf{r}}(\tau)$ is not positive semidefinite. Q.E.D

It is easy to see how this condition would arise in the more practical case where $r_{km}(\tau)$ is a slowly varying function of time, whose Fourier transform approximates the delta function. If the rectangular window to be applied is relatively short, then the Fourier transform of the windowed function $\hat{r}_{kk}(\tau)$ would be a good approximation to the sinc function.

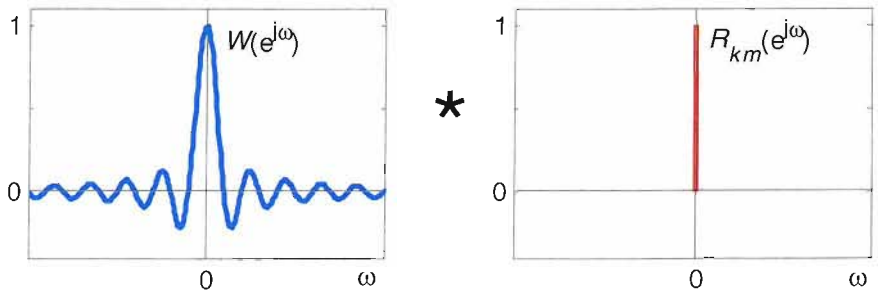


Figure 5.5: Showing the convolution of the frequency response a rectangular window function (left) with the Fourier transform of a constant function $r_{km}(\tau)$ (right).

Choice of Window Function and the G_c Cost Function

It may now be clear that the positive semidefiniteness of $\hat{\mathbf{r}}(\tau)$ depends, to a great extent, on the type of window function applied. A window function whose frequency response is characterised by non-negative amplitude values would guarantee that $\hat{\mathbf{r}}(\tau)$ is positive semidefinite for any true spectrum. Examples of common window functions that satisfy this criterion are those that have a sinc^2 frequency response, i.e. the triangular, Bartlett, Blackman-Harris and Chebyshev window functions.

It would not be sensible to operate on a parahermitian matrix that has been modified by a rectangular window with the version of the SBR2 algorithm that uses the G_c cost function, SBR2(G_c) introduced in (5.14). This is because the product in the denominator of the objective defined by (5.37) will be negative if one of the zero-lag terms, $\hat{r}_{kk}(0)$, is negative; cross-correlation terms that are negative will be ignored by the SBR2 algorithm. Instead of the rectangular window, one of the window functions mentioned in the previous paragraph would need to be used in combination with the G_c cost function, such as the Bartlett window.

5.5 SBR2 Family of Algorithms

We shall introduce a categorisation of the SBR2 algorithm based on whether or not a window is explicitly applied in the covariance-domain. Those versions that do not explicitly window the covariances are from the class of *regular* SBR2 algorithms and those that do are of the class of WCD-SBR2 algorithms. This will make comparing the two types of algorithm less confusing. All versions of the algorithm are collectively referred to as the SBR2 algorithm. Two important governors of the performance of SBR2 are:

1. The adaptation technique or algorithm core which estimates the parameters for each elementary paraunitary block, e.g. the classical Jacobi-type search strategy in section 5.2 and the cost functions introduced in section 5.3;
2. Any processing that is applied to the input data prior to the application of SBR2, e.g. covariance-domain windowing in section 5.4.

As shown in this chapter, the development of SBR2 has led to the creation of different versions of the algorithm with differing pre-processing stages and algorithm cores. Note that using a different cost function constitutes a change to the algorithm core. In order to show the relationship between the different SBR2 variants and what has been introduced in this thesis, a summary chart is presented in Fig. 5.6. Those algorithm variants that are marked with × have not been analysed in this thesis. This is mainly because they can be outperformed by other similar versions and their analysis would be superfluous in the presence of the characterisations of those other versions. In Fig. 5.7, we provide a flow diagram from which all the algorithm variants may be obtained.

Note that another way of modifying the algorithm core is to adopt a different search scheme, such as a generalisation of the cyclic Jacobi algorithm introduced in section 5.6.3. Due to time constraints, SBR2-algorithms based on this strategy of optimisation are not considered for analysis in this thesis. Therefore, they are left out altogether from Fig. 5.6 and Fig. 5.7.

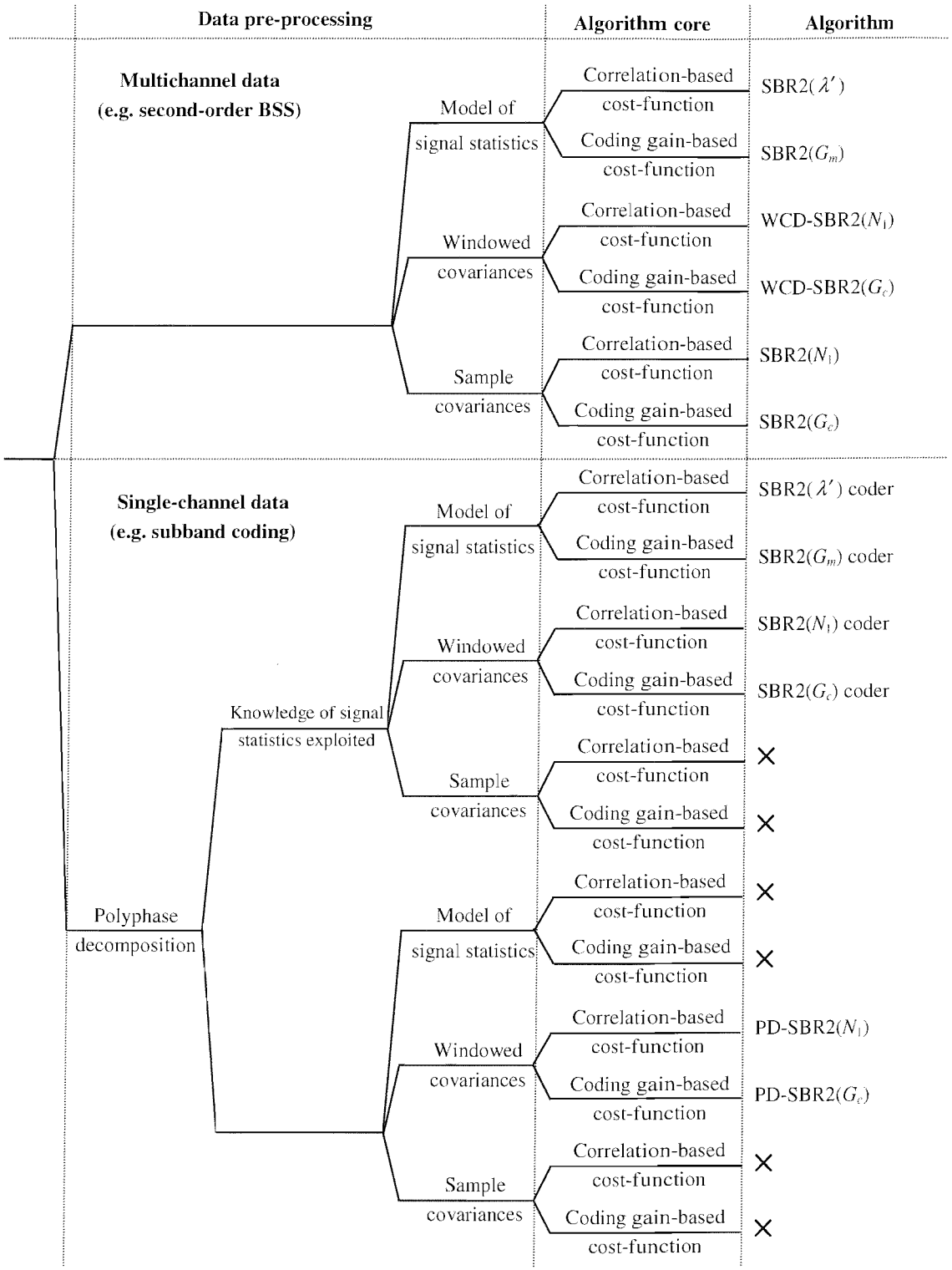


Figure 5.6: Chart of possible SBR2 algorithms and algorithm families.

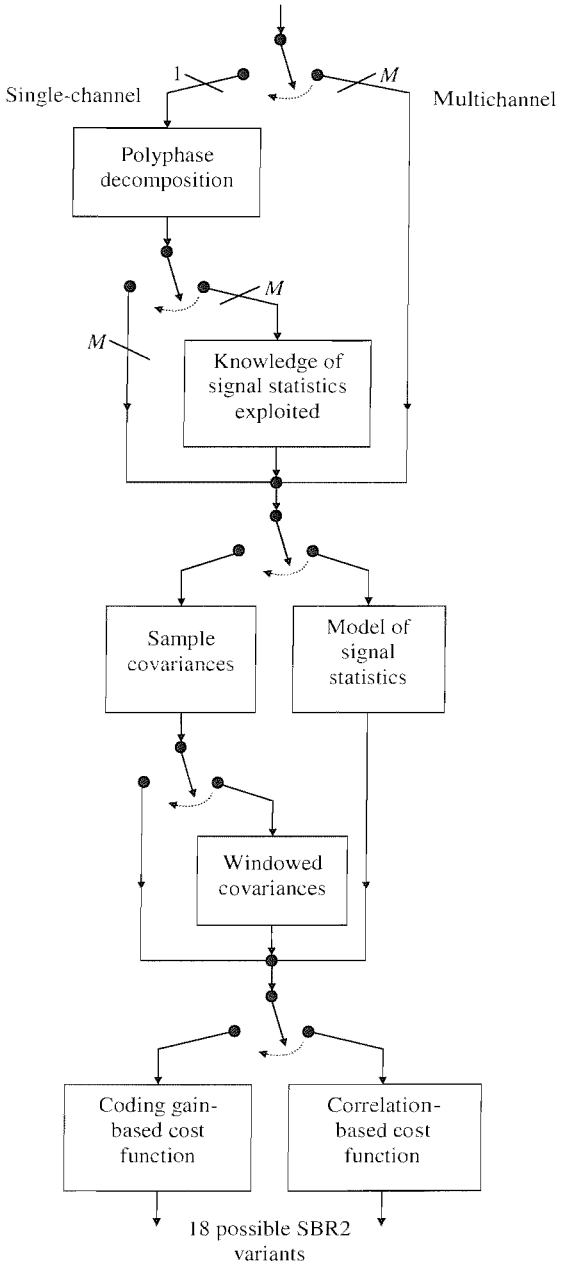


Figure 5.7: Variable flow diagram for obtaining anyone of the eighteen different SBR2 algorithm variants.

The algorithms labelled PD-SBR2 will be evaluated later in section 6.2 in order to motivate the development of a novel technique for adapting SBR2 to subband coding. This involves the invention of a pre-processing stage for effective application of the SBR2 algorithm to the subband-coder design problem; the resultant algorithm is termed the SBR2 coder, which is included in Fig. 5.6 and Fig. 5.7.

5.6 Implementation Aspects

5.6.1 Solution to the Rotation Angle

Recall from section 5.2 that, at each SBR2 iteration, the dominant off-diagonal r'_{np} is zeroed by rotating the elements of the covariance matrix $\mathbf{r}'(0)$ in (5.8) through an angle θ. This angle is found by solving a quadratic in $\tan(\theta)$, as expressed by (5.9). The numerator $y = 2\left|r'_{np}(0)\right|$ and denominator $x = r'_{nn}(0) - r'_{pp}(0)$ of the quotient in (5.9) are the components of a vector $\mathbf{v}$. The task of zeroing the cross-correlation terms $r'_{np}(0)$ and $r'_{pn}(0)$ may be interpreted as one of rotating the vector $\mathbf{v}$ onto the x-axis so that the y^{th} component of the vector vanishes. This is illustrated by the example shown in Fig. 5.8. One of the solutions to (5.8) is to rotate $\mathbf{v}$ onto the **negative** x-axis through an angle θ_i, as represented by the red dashed arrow in the diagram. This is the minimum rotation angle required for this particular example. The other possible solution is to rotate $\mathbf{v}$ clockwise by an angle θ_o so that it is brought onto the **positive** x-axis; see the blue dash-dotted arrow.

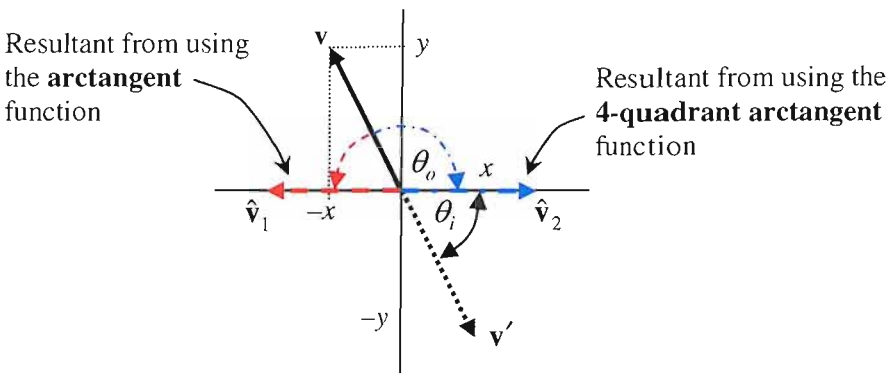


Figure 5.8: Example of the different rotation angle solutions.

The range of the arctangent function is commonly selected as the interval $(-\pi/2, \pi/2]$. This is adopted by the 'atan.m' function in Matlab [67]. The input argument to the atan.m function is the ratio in (5.9). In the example shown (Fig. 5.8), $\mathbf{v}$ lies in the upper-left quadrant so it is outside the range space of the arctangent function. The function interprets the negative

valued ratio as the vector $\mathbf{v}'$, which is the vector $\mathbf{v}$ rotated by π radians, and so, for the example shown, it will find the angle θ_i. Essentially, the sign of the individual coordinates x and y are ignored by this function. Equation (5.9) is in terms of double angles, so a rotation of the covariance functions by θ corresponds to a rotation of $x_n(t)$ with $x_p(t)$ by $\theta = (\pi + \theta_i)/2$. Therefore, a rotation of π radians corresponds to a 90° rotation of the signals. In other words, the two signals are swapped and the resultant lower indexed signal is negated.

Since the x-coordinate of $\mathbf{v}$ in Fig. 5.8 is negative, we have $r'_{pp}(0) > r'_{nn}(0)$, and an anti-clockwise rotation of $\mathbf{v}$ by θ_i preserves this condition. That is, energy from the cross-terms is moved mostly onto the lower diagonal (the autocovariance of the second signal $x_p(t)$). Because of this nature and since (by conjecture 5.1 in section 5.3) the SBR2 algorithm tends to impose spectral majorisation on the signals, we can infer that the ordering of the signal spectra after a number of iterations depends on the total power of the input (unprocessed) signals. Therefore, the output signals, even though approximately majorised, would not necessarily satisfy the ordering regime defined in (4.14); however, this is trivially obtained by permuting the majorised signals accordingly. An example of signals that are spectrally majorised but ordered according to the input signal powers is shown in Fig. 5.9(b) for the input spectra in Fig. 5.9(a). In Fig. 5.9(c) we show a set of spectra ordered in the manner of (4.14), which can be obtained from Fig. 5.9(b) by the application of a permutation matrix.

In contrast to the regular arctangent function, the four-quadrant arctangent function (or Matlab's atan2.m function) can take into account the quadrant in which the vector actually lies when determining a suitable angle. This is because its range space has the limits $(-\pi, \pi]$. The four-quadrant arctangent function finds an angle such that the vector $\mathbf{v}$ is always rotated onto the positive x-axis. Hence, for the example of Fig. 5.8, the solution to the optimal rotation angle is θ_o not $\pi + \theta_i$; i.e. there is no signal swapping. Using the four-quadrant arctangent function essentially constrains the ordering of the output signals to be independent of the variances of the input signals, and so the output spectra satisfy the condition in (4.14): the spectrum of the first output signal dominates that of the second output signal, subject to the spectrum of the second output signal dominating that of the third and so on, as in Fig. 5.9(c). Hence, the only difference in using the atan.m and the atan2.m functions in terms of observable results is the labelling of the output signals.

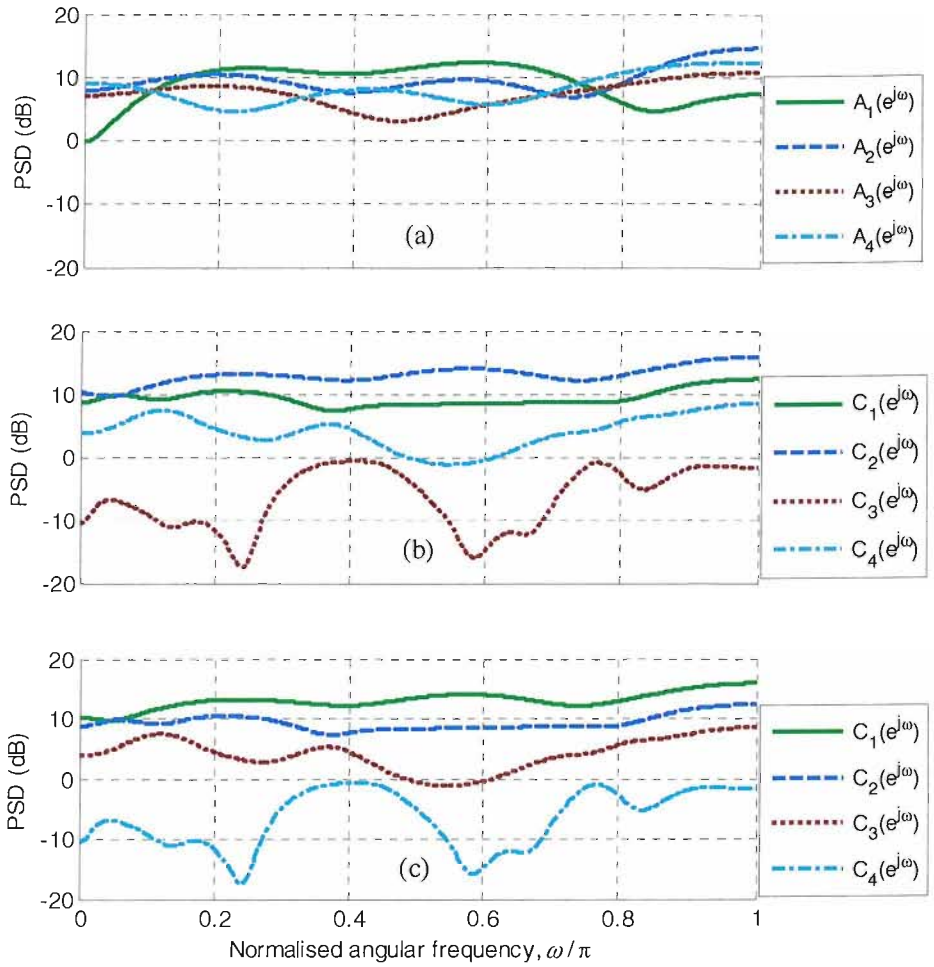


Figure 5.9: Example of channel swapping. (a) Input signal spectra. (b) Majorised spectra from using the regular arctangent function; the ordering is based on the input powers. (c) Majorised spectra from using the four-quadrant arctangent function; the spectra are ordered according to the ordering regime in (4.14).

5.6.2 Algorithm Efficiency

Circulant Assumption

It is convenient to assume that the data sequences repeat periodically every T samples, i.e. that the sequences are cyclic (or circulant). When data is shifted beyond T samples it is wrapped around to the zero time lag position. Therefore, we do not differentiate between delays whose lengths differ by a multiple of T. This permits the order of all polynomial matrices involved in the computation to be restricted to order T. It also provides a firm foundation for computing entries of the space-time covariance matrix using the DFT or FFT, as will be explained later in this subsection. This can help to reduce the computational cost, and has obvious advantages in

terms of computer programming by defining a fixed bound for the size of any associated arrays. This cyclic data version of SBR2 is also guaranteed to converge.

If circulant data is assumed, then the paraunitary matrix found by SBR is irrevocably tied to the circulant assumption; it is not easy to see how a non-circulant paraunitary matrix can be found from the circulant one. By contrast, we can treat non-circulant data as circulant and apply the circulant SBR2 algorithm to find a circulant paraunitary matrix. As long as the correlation time of the signals is much less than the circulant length T, the non-circulant paraunitary matrix, which is a product of the same delays and rotations, should give an effective EVD/SVD for the original non-circulant data. Note that, unlike the bandwise algorithm, the SBR2 algorithm is not restricted to operating on circulant data; it can construct a legitimate paraunitary matrix for non-circulant data without the circulant assumption in place.

Exploitation of Covariance Matrix Symmetry

One way of reducing the computational cost of the search procedure is to take advantage of the fact that the polynomial covariance matrix $\mathbf{R}(z)$ in (5.1) is parahermitian. This is done by considering the entries in the upper (or lower) triangle of $\mathbf{R}(z)$ only. Instead of searching through $M^2 - M$ off-diagonal entries, only $(M^2 - M)/2$ are necessary for the search, where M is the number of channels. This modification results in a factor of two saving in computation time. However, this is not significant in terms of overall algorithm execution time since both methods have computational complexity of order M^2, i.e. $O(M^2)$; this is for a single step and not taking into account the search in time.

Computing the Covariances in the Frequency-Domain

The largest cross-covariance term $r_{np}(\tau_\ell)$ can be found by an exhaustive time-domain approach; to find the optimum delay τ_ℓ would require a search through the $(M^2 - M)/2$ off-diagonal entries for all the elements of a set of time delays. This would, however, be rather computationally expensive. Let the range of delays considered be equal to the data length, T. The SBR2 algorithm using this search method requires $O(LM^2T^2)$ arithmetic operations, where, as before, L is the total number of algorithm iterations.

The speed of the algorithm can be considerably increased by using the DFT to obtain the correlations, provided that the circulant assumption is made. The cross-covariances between pairs of signals $x_k(t)$ and $x_m(t)$ can be computed thus $R_{km}(e^{j\omega}) = X_k(e^{j\omega})X_m^*(e^{j\omega})$, where $X_k(e^{j\omega})$ is the DFT of $x_k(t)$ and $X_m^*(e^{j\omega})$ is the DFT of $x_m(t)$ with all terms complex conjugated. The sequence $r_{km}(\tau)$ is then obtained for all τ by way of the IDFT. A flowchart of

this method is given in Fig. 5.10. The search for the optimum delay or largest magnitude square term in $r_{km}(\tau)$, for $k \neq m$ and $m > k$ (assuming a search in the upper-triangle), is then carried out. This algorithm computes the same result as the time-domain technique in $O(LM^2T\log_2 T)$ operations. It is clear that for large T, using this method for SBR2 is computationally more efficient than the time-domain approach.

Covariance-Domain Approach

As discussed in section 5.2, the SBR2 algorithm can operate on the sample space-time covariance matrix rather than operating on the data channels. A consequence of which is an increase in the algorithm's computational speed. The computation time is dominated by two factors. The first is the one-off cost of constructing the space-time covariance matrix (Fig. 5.10) at the start of the algorithm (before the SBR2 loop). This process requires $O(M^2T\log_2 T)$ operations. The second slowest point in the algorithm is within the main loop of the algorithm. This is the search for the largest off-diagonal term, which does $O(LM^2T)$ work. Thus, the complexity of this algorithm is $O(M^2T\log_2 T + LM^2T)$. In comparing the dominant term here, $O(M^2T\log_2 T)$, with that of the time-domain approach, $O(LM^2T^2)$, we see that the covariance-domain version of the SBR2 algorithm is faster. An idea of the speed of the algorithm can be gathered from the following example.

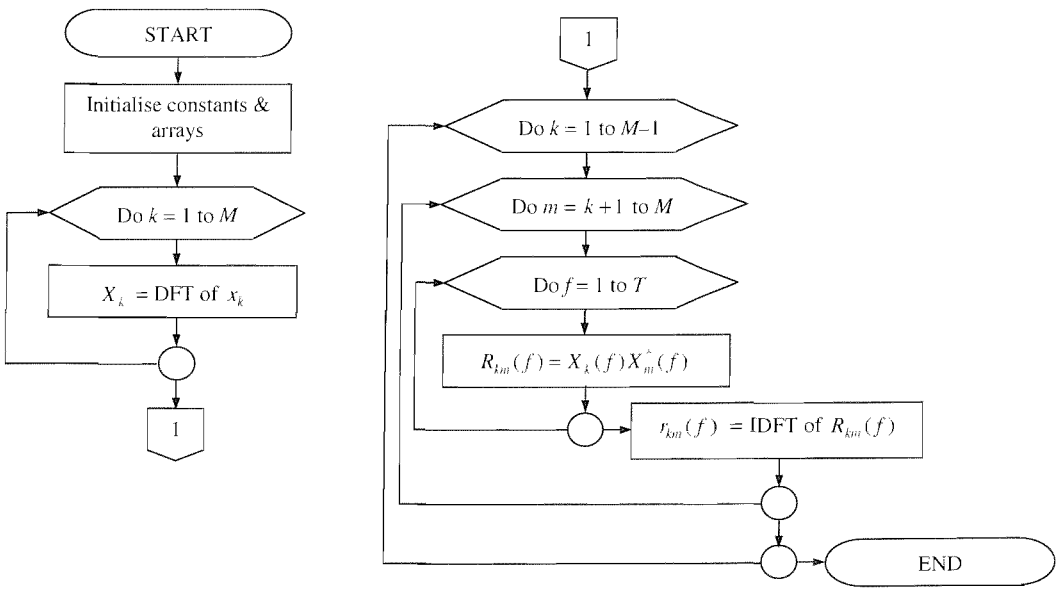


Figure 5.10: Algorithm for the transformation of the data to the covariance-domain.

Example. Let there be $M = 10$ signals with $T = 1000$ data samples. A good estimate of the optimal decorrelation filter bank can be obtained with $L = 600$ iterations of the SBR2 algorithm in approximately 16 seconds using a 3GHz Pentium 4 processor. Although this would typically be too slow for real-time applications, there is a great deal of scope for improving the computational performance of the algorithm.

5.6.3 Alternative Search Technique

In section 5.2, we extended the classical Jacobi algorithm to polynomial matrices to create the multichannel SBR2 algorithm. This strategy may be described as a 'greedy' approach that searches for the largest correlation between two signals at each iteration. An alternative scheme can be obtained by generalising the cyclic Jacobi algorithm, introduced in section 2.2.3, to polynomial matrices. The paraunitary transformation of (5.4) can be applied to the $(M^2 - M)/2$ off-diagonal entries (to zero the largest term in each entry) that are selected in some fixed order. The sequence of transformations constitutes a sweep, which may have a cyclic-by-rows or cyclic-by-columns construction.

In contrast to the SBR2(N_1) algorithm, this strategy treats all off-diagonal entries equally. Therefore, it is proportionately, equally sensitive to all the signals, which is also a property of the G_c cost function, discussed in section 5.3.2. A SBR2 algorithm that is based on the cyclic Jacobi algorithm would naturally perform better spectral majorisation than SBR2(N_1).

The computation time of the SBR2 algorithm with the cyclic-by-rows Jacobi method is greater than that for the SBR2 algorithm with the classical Jacobi algorithm in place. In each step of the SBR2 algorithm a Jacobi sweep is executed, which means that the algorithm does $O(M^2 T \log_2 T + LM^3 T)$ work. The increase in order arises because there are $(M^2 - M)/2$ delay-rotates, each of which requires $O(M)$ operations.

Note that there may be other ways of implementing the SBR2 algorithm that are not based on a pairwise search approach. One method is to search over a wide range of possible delays in a delay set for the entire polynomial covariance matrix to find the best delay and then apply a multichannel rotation. However, at first glance, this would seem to be a computationally costly technique. Investigation into a computationally efficient implementation of this scheme has not been performed due to constraints on time.

5.7 Performance Analysis

5.7.1 Characterisation of the SBR2 Algorithm

The performance limiting factors of the bandwise algorithms discussed in section 4.4 are not specific to those algorithms, but seem to occur in a hidden form in the SBR2 algorithm. As before, the limitations transcribe into a trade-off between frequency resolution and sample noise. The number of degrees-of-freedom the algorithm has increases as the number of algorithm steps L increases. A large value of L results in the algorithm having increased resolution in frequency space, but effectively reduces the number of data samples that it can use in estimating each quantity (unitary matrix). In this case, the diagonalisation performance (λ) is dominated by the sample errors. In contrast, for small L, the predominant contributing factor to λ is the lack of spectral resolution.

The SBR2 algorithm can be thought of as applying a window to the sample covariance functions. The window edges asymptote to a peak value obeying a power law behaviour of order η. The algorithm applies a window with a bounded value of η. The larger the value of η the better the diagonalisation performance of the algorithm for data with short covariance functions; the performance tends to be worse otherwise. Note that a large value of η will give performance approaching that of the WCD-bandwise algorithm with a rectangular window, as discussed in section 4.4.3.

Characterisation of the WCD-SBR2 Algorithm

The application of a covariance-domain window can be viewed as a modification to the SBR2 algorithm that increases the effective value of η. A rectangular window function effectively means that $\eta = \infty$. This algorithm has two parameters: the number of iterations L and the window length T_w. The application of a window in the covariance domain effectively reduces the number of independent estimates made and so the accuracy of these estimates can be significantly improved. The windowed space-time covariance matrix $\hat{\mathbf{r}}(\tau)$ is a better estimate of the true statistical measure $\mathbf{a}(\tau)$ than the original space-time covariance matrix $\mathbf{r}(\tau)$, both for our purposes and in terms of sums of square errors. The effect of the windowing is to smooth the signal spectra, as performed by the Blackman-Tukey spectral estimator [7].

It is important that the length of the window is not less than the time extent of the true correlations between the signals. This condition must be adhered to so that the algorithm has access to all true cross-covariance terms. In the case where a rectangular window is used, sample noise is the only contributor to λ. Because of this, the WCD-SBR2 algorithm offers an

improvement in performance over the regular SBR2 algorithm when the true covariance function has finite extent in time (windowing is ineffective otherwise).

The type of covariance-domain window function applied will have an effect on the performance of the algorithm, and its effectiveness depends on the true correlations of the signals. Therefore, knowledge about the underlying statistics of the signals will aid in choosing a suitable window function; a poorly chosen window may attenuate or even remove some essential cross-covariance terms. This may be demonstrated through some examples.

Example 1. Consider the case of two signals with a corresponding true cross-covariance function $a_{12}(\tau)$ as represented by the green curve in Fig 5.11(a). The greatest lag at which correlations exist is $T_c = 5$. The sample cross-covariance function, $r_{12}(\tau)$, consists of significant (signal-related) covariance coefficients in the range $[-T_c, T_c]$ and noise-related coefficients outside this range; this is the solid blue curve in Fig 5.11(b). The noise terms may be set to zero with the application of a rectangular window function of length $T_w = 2T_c + 1$, represented by the dotted red curve in Fig. 5.11(b). With such a window function the important cross-covariance terms are preserved, including those close to $|T_c|$, as seen in Fig. 5.11(c).

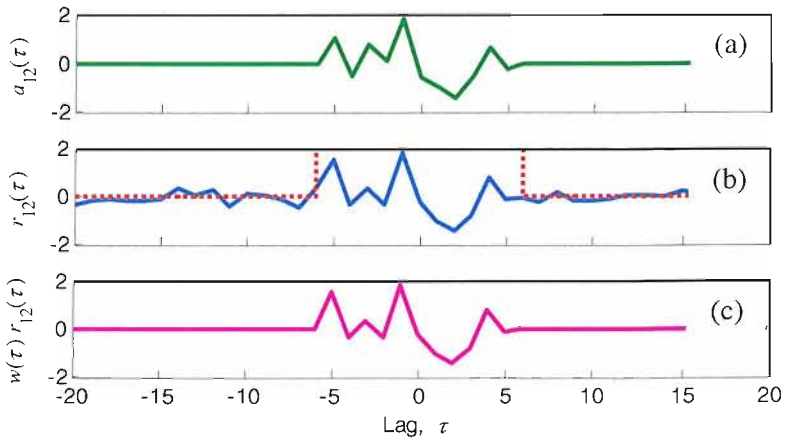


Figure 5.11: (a) True cross-covariance function. (b) sample cross-covariance function and a rectangular window. (c) Result of windowing the sample covariance in (b); all signal-related terms are preserved.

Example 2. Consider again the two-signal case with a true cross-covariance function as shown in Fig. 5.12(a). Suppose the SNR of the signals is low and so the sample covariance function is a noisy estimate of $a_{12}(\tau)$, as represented by the solid blue curve in Fig. 5.12(b). If a rectangular window (thin dotted black curve in Fig. 5.12(b)) is applied in this case, some dominant noise-related coefficients would be preserved; see the thin black dotted curve in Fig. 5.12(c). On the other hand, if a tapered window, such as the Bartlett window function represented by the thick dotted red curve in Fig. 5.12(b), is used, then the dominant noise-related terms are attenuated; see the solid pink curve in Fig. 5.12(c).

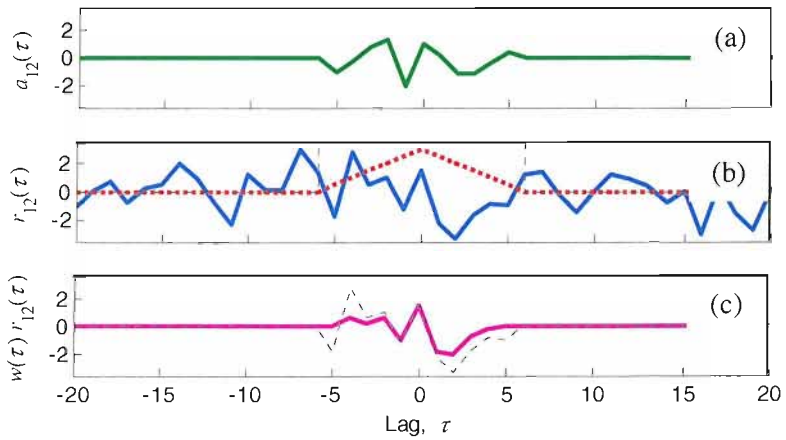


Figure 5.12: (a) True cross-covariance function. (b) sample cross-covariance sequence, a rectangular window and a Bartlett window function. (c) Results of windowing the sample covariance in (b) with a rectangular window (dashed curve with noise-related terms preserved) and a Bartlett window (solid curve with noise-related terms attenuated).

5.7.2 Experimental Results

Using computer simulations we evaluate the performance of the various SBR2 algorithms. The algorithms were applied to the multichannel convolutive problem represented by Fig. 4.1. The results provided here underline the validity of the performance characterisation given in sections 4.5.1 and 5.7.1. Finally, the subspace decomposition accuracy of the SBR2 algorithm is investigated.

For the following simulations the input signals to be processed were generated as follows: With reference to (4.3) (or Fig. 4.1), the propagation of M signals onto M sensors was modelled by means of a $M \times M$ polynomial mixing matrix $\mathbf{G}(z)$ whose entries were FIR filters of order

N_g with coefficients drawn randomly from a uniform distribution in the range $[-1, 1]$. The source signals took the form of independent BPSK sequences, which were convolutively mixed using $\mathbf{G}(z)$. This model is the same as that used in all the experiments in section 4.5. In some of the following experiments the source signals were modelled as quaternary phase-shift keying (QPSK) sequences with zero mean and unit variance. The samples took values $\pm 1/\sqrt{2} \pm j1/\sqrt{2}$, each with a probability of 1/4, which gives rotated $\pi/4$ constellation points [51]. For reasons given in section 5.6.2, we assume that the input data is circulant.

The WCD-SBR2(N_1) algorithm, like the WCD-bandwise algorithm, was run with a rectangular correlation window and a correlation window length of $T_w = 2N_g + 1 = 2T_c + 1$. For the reason given in section 5.4, the WCD-SBR2(G_c) algorithm was applied with a triangular window function, where $T_w = 4N_g + 1$. Expected performance quantities were obtained by averaging over 100 realisations of each experiment.

Strong Decorrelation

Dependence on *L*. In Fig. 5.13, the evolution of the diagonalisation performance measure (λ) for the SBR2(N_1), SBR2(G_c), WCD-SBR2(N_1) and WCD-SBR2(G_c) algorithms over a number of algorithm iterations L for the three-channel case is shown. Also included in these figures is the performance of the non-blind technique, SBR2(λ'). For this simulation, the order of the filters in $\mathbf{G}(z)$ was N_g = 5 and the number of data samples, T, used to estimate the space-time covariance matrix in (5.1) was chosen to be 729. The correlation window parameter for the WCD-SBR2(N_1) and WCD-SBR2(G_c) algorithms was set to 11 and 21, respectively.

It is obvious from these plots that there is an optimal number of steps, L', required at which the regular SBR2 algorithms achieve the lowest value of λ. The optimum performance point for the SBR2(G_c) algorithm (thick solid curve) is reached with a greater number of iterations than is required for SBR2(N_1) (thick dashed curve). This is because SBR2(G_c) is not as sensitive to large cross-correlations as SBR2(N_1) is, as explained later in section 6.1.2. For the same reason, the SBR2(G_c) obtains a lower value of λ than SBR2(N_1). The reduction in performance either side of the L' point for the regular SBR2 algorithm substantiates the analysis presented in section 5.7.1. For small values of L, λ is governed by the errors that restrict frequency resolution. For large values of L, errors are due to sample noise.

As expected, the WCD-SBR2 algorithms produce output signals that are a closer approximation to strongly decorrelated signals than those obtained from the regular SBR2 algorithms. Their success is directly attributed to the windowing performed on the sample covariances, which largely removes noise related crosscovariance terms (i.e. that are not related

to the true statistics of the signals). It is apparent that the performance of the WCD-SBR2(N_1) is slightly better than that of the WCD-SBR2(G_c). A likely reason for this is that a rectangular window is more suited to the true covariances of the signals, as discussed in section 5.7.1. This is evident in the fact that a larger window length was required when using a triangular window. By contrast to the regular SBR2 algorithms, a very low value of λ is attained by the WCD-SBR2 algorithms, which is due to the windowing. The SBR2(λ') algorithm finds a paraunitary matrix that is the optimum decorrelation filter bank, to a good approximation. Its success is certainly due to the fact that it uses true statistics to design near-optimal filters.

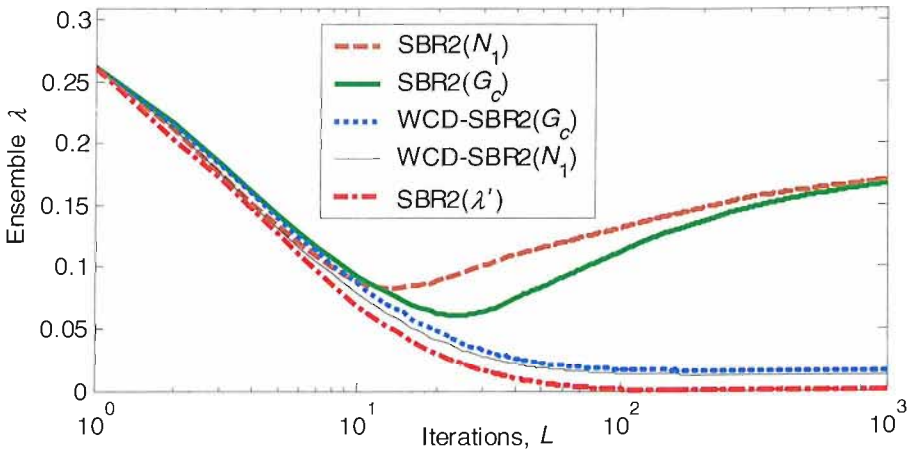


Figure 5.13: A comparison of the diagonalisation performance of the various SBR2 algorithms for the case of 3 signals with real-valued data samples.

The λ performance evolution of the various algorithms has been assessed for the case where there were three complex-valued signals; see Fig. 5.14. The source signals were QPSK sequences of length $T = 729$. The general performance characteristics of the algorithms are similar to those seen for real-valued data. However, a difference in absolute terms is observed: the regular SBR2 algorithms achieve a somewhat greater level of diagonalisation than for the real-signals case. A possible hypothesis for this is that there is effectively extra information about the source signals in the complex part of the data, which is exploited by SBR2. This may be easily verified by doubling the number of samples used in the real-signals case and noting the increase in performance, as will be demonstrated in the following experiment.

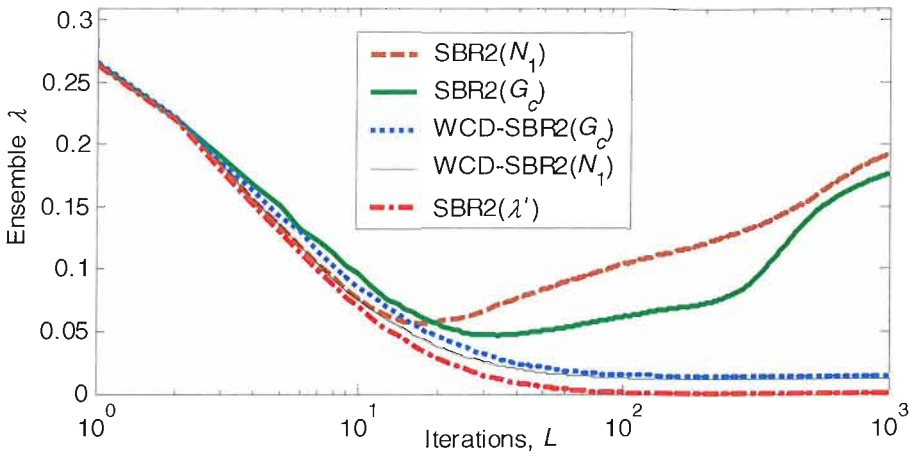


Figure 5.14: A comparison of the diagonalisation performances of the various SBR2 algorithms for the case of 3 complex-valued sequences.

Dependence on *T*. The dependence of the λ measure on the number of data samples T for varying L was obtained using SBR2(N_1) and WCD-SBR2(N_1), the results from which are provided, respectively, in Fig. 5.15 and Fig. 5.16. The simulations were for four signals and $N_g = 3$. It is clear that the more data samples there are available for processing the better the performances of the algorithms. For both algorithms, we see that λ is approximately independent of T for small L and for $T >> T_c$. This occurs because of a lack of spectral resolution, which is consistent with the analysis given in 5.7.1. Conversely, as L increases, sample errors become more dominant and therefore λ becomes more dependent on T. These relationships were also observed for SBR2(G_c) (not shown).

Identical simulations to those in Table 4.1 of section 4.5 have been carried out for the SBR2 algorithms with results tabulated in Table 5.1. The regular SBR2 algorithms were allowed to run for an optimum number of steps L', which ranged from 18 to 60 iterations depending on the algorithm and T: the values of L were determined by conducting experiments of the type for which the results in Fig. 5.13 relate. The WCD-SBR2 algorithms were allowed to run for 500 iterations. In comparing the λ performances of SBR2 with those of the bandwise algorithms, the multipath-enabled SVD and the SVD, we note the following: WCD-bandwise and WCD-SBR2 algorithms produce comparable performances and can generally achieve greater diagonalisation than the other methods; the SW-bandwise algorithm performs slightly better than the regular SBR2 algorithms with T_b' and L', respectively; and the SBR2(G_c) algorithm attains a lower value of λ generally than that obtained by the multipath-enabled SVD.

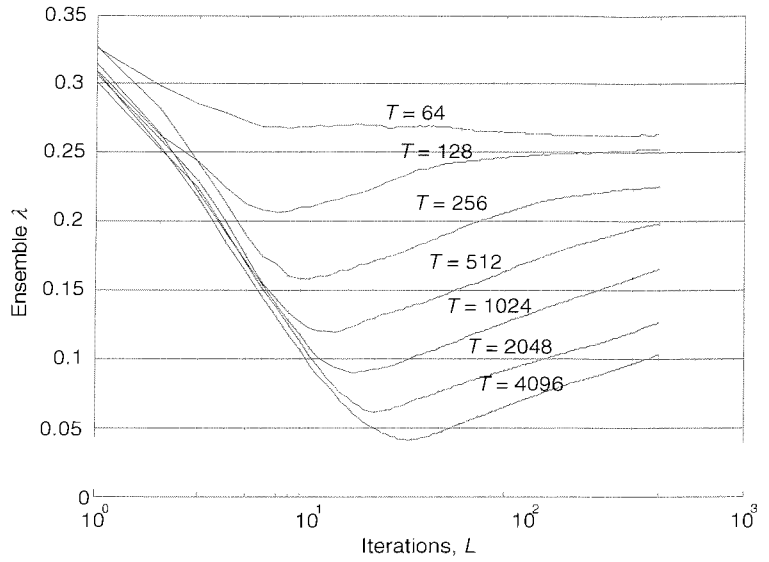


Figure 5.15: Dependence of the diagonalisation performance on T for SBR2(N_1).

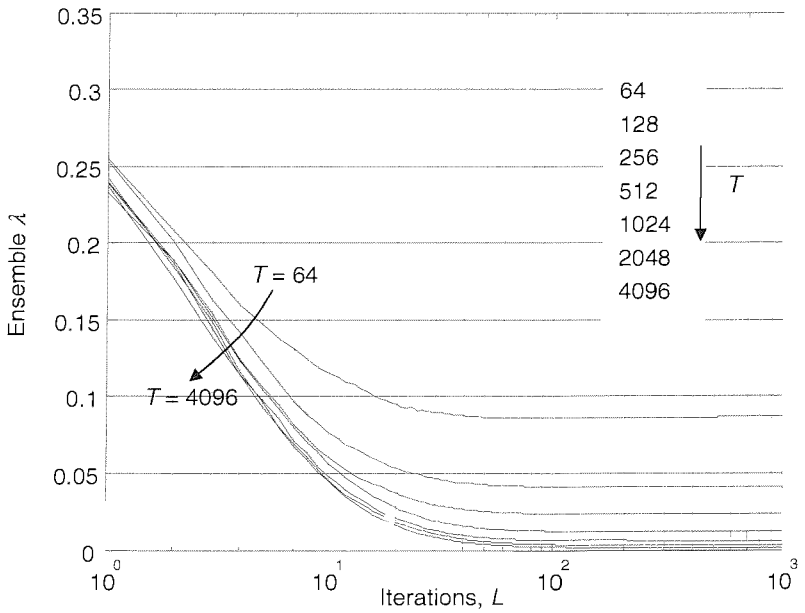


Figure 5.16: Dependence of the diagonalisation performance on T for WCD-SBR2(N_1).

Data length /Algorithm	243 samples (λ)	729 samples (λ)	2187 samples (λ)
SBR2(λ')	9.9×10^{-5}	1.0×10^{-4}	9.6×10^{-5}
WCD-SBR2(N_1)	5.6×10^{-2}	1.9×10^{-2}	6.6×10^{-3}
WCD-SBR2(G_c)	5.3×10^{-2}	2.3×10^{-2}	9.7×10^{-3}
SBR2(G_c)	1.7×10^{-1}	9.3×10^{-2}	4.8×10^{-2}
SBR2(N_1)	2.1×10^{-1}	1.3×10^{-1}	7.5×10^{-2}

Table 5.1: Comparison of the decorrelation performance of the SBR2 algorithms for different values of T.

Spectral Majorisation

In Fig. 5.17, the power spectral densities of the output signals from three SBR2 algorithm variants are shown for four input signals (identical to those used in section 4.5) with spectra as in Fig. 5.17(a). The SBR2(N_1) and SBR2(G_c) algorithms were allowed to run for an optimum number of steps, which was 20 and 35 iterations, respectively; see Figs. 5.17(b) and (c), respectively. The WCD-SBR2 algorithms were allowed to run for 300 iterations. We see that all the algorithms produced output signals that are approximately spectrally majorised. It is quite evident that the WCD-SBR2(N_1) algorithm – see Fig. 5.17(d) – produces significantly better results than the regular SBR2 algorithms. The WCD-SBR2(G_c) and 'near'-optimal algorithms produced majorised spectra (not shown) similar to those from WCD-SBR2(N_1).

In general, SBR2(G_c) and the bandwise methods have consistently performed better spectral majorisation than SBR2(N_1). As explained in section 5.3.2, a possible reason for the short coming of SBR2(N_1) is that after a few iterations, it concentrates on improving energy compaction in signals with large power and neglects compacting energy from weaker signals. It does this because powerful signals give rise to large crosscorrelations, which the algorithm targets in order to maximise the N_1 cost function. The condition is exacerbated when there are very weak signals present, and highlights a drawback of the cost function used and/or the method of search adopted (classical Jacobi method) for coding applications. In contrast, the G_c cost function is proportionately, equally sensitive to changes in any of the signals. Therefore, it enables SBR2 to perform greater data compression.

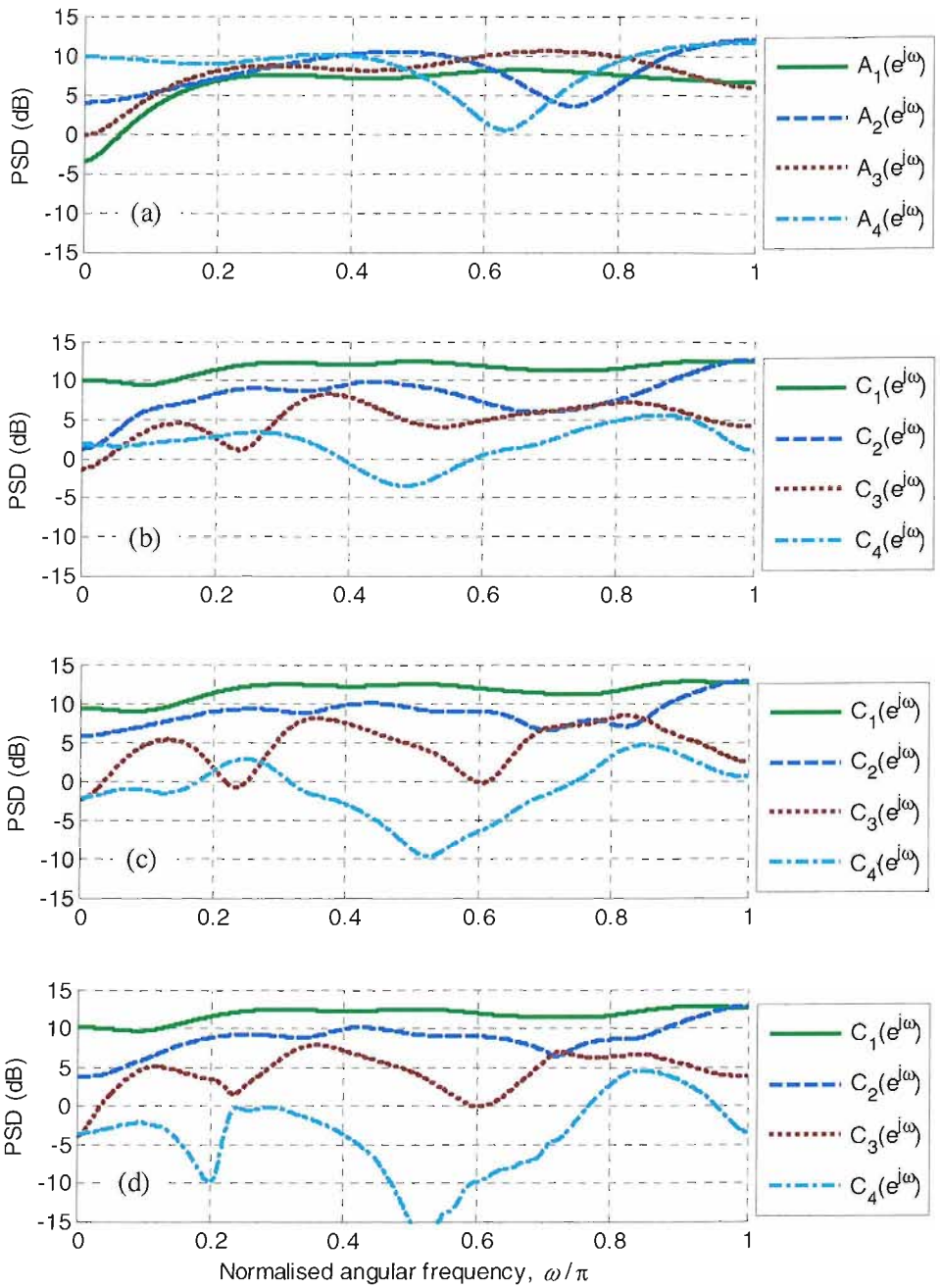


Figure 5.17: The majorised output signal spectra of (b) SBR2(N_1), (c) SBR2(G_c) and (d) WCD-SBR2(N_1) for the given input signal spectra in (a).

Broadband Subspace Decomposition

In order to demonstrate the capability of the SBR2 algorithm in performing broadband subspace decomposition, we present the results of a simple computer simulation. The propagation of three source signals onto six sensors was modelled by means of a 6×3 polynomial mixing matrix $\mathbf{G}(z)$ whose entries were order-5 FIR filters with coefficients drawn randomly from a uniform

distribution in the range $[-1,1]$. This overdetermined system is expressed in (4.3) and depicted in Fig. 4.1. The source signals were simulated as QPSK sequences of length $T = 2000$, as specified in section 5.7. Gaussian random noise was added to each simulated sensor output with variance, σ_ψ^2, chosen to achieve the desired SNR. From (4.37), the input SNR is given by $SNR_{in} = \text{trace}[\mathbf{G}(z)\tilde{\mathbf{G}}(z)]|_{z^0} / 6\sigma_\psi^2$ for this experiment.

For each chosen value of SNR, the SBR2(N_1) and WCD-SBR2(N_1) algorithms were used to strongly decorrelate the sensor signals by diagonalising the space-time covariance matrix according to (5.2). The SBR2(N_1) algorithm was allowed to run for $L = 50$ iterations which was the optimal number of steps for this problem, as determined from the corresponding ensemble λ performance. On the other hand, the WCD-SBR2(N_1) algorithm was run for 400 iterations, at which stage the modulus of the largest off-diagonal coefficient was extremely small. The signal and noise subspaces were then separated, assuming that the SBR2 algorithms had successfully achieved spectral majorisation. The signal subspace was defined by the first three output channels. The integrity of the signal and noise subspaces was quantified using the γ measure defined by (4.36) in section 4.2.4. The smaller the value of γ the more reliable the subspace estimation. The value of γ as a function of SNR for this experiment is plotted in Fig. 5.18. Each point on the graph represents the value of γ for 100 trials.

From Fig. 5.18, it can be seen that low values of γ were achieved by both SBR2 variants, and γ diminishes rapidly with increasing SNR. This indicates that the algorithms are capable of effective broadband subspace decomposition. The algorithms produce inaccurate subspace estimates below a certain input SNR level, which is different for the two algorithms. These curves show that a transition band exists where the algorithms go from producing many errors to providing good subspace estimation. We also make the following interesting observations:

1. As expected, more accurate subspace estimation can be achieved with WCD-SBR2 (dashed curve in Fig. 5.18) than is obtained by the regular algorithm (solid curve): at 0dB SNR, $\gamma \approx 0.2$ for SBR2(N_1), by contrast, $\gamma \approx 0.02$ for the WCD variant;

2. The variance of γ for moderate-to-high input SNR is somewhat smaller (i.e. a smoother curve) for WCD-SBR2 than that for the regular SBR2 algorithm;

3. The variance of the γ measure gets larger as the input SNR falls.

4. There are values of γ that are greater than unity (for low input SNRs).

A possible hypothesis for the first two points in this list is that the WCD-SBR2 algorithm operates on a more accurate estimate of the space-time covariance matrix so it makes less errors than the regular SBR2 algorithm. The last two observations can be easily explained as an effect

due to the fact that the construction of the paraunitary matrix is based on sample statistics, and so there are sample errors in the quantities being estimated.

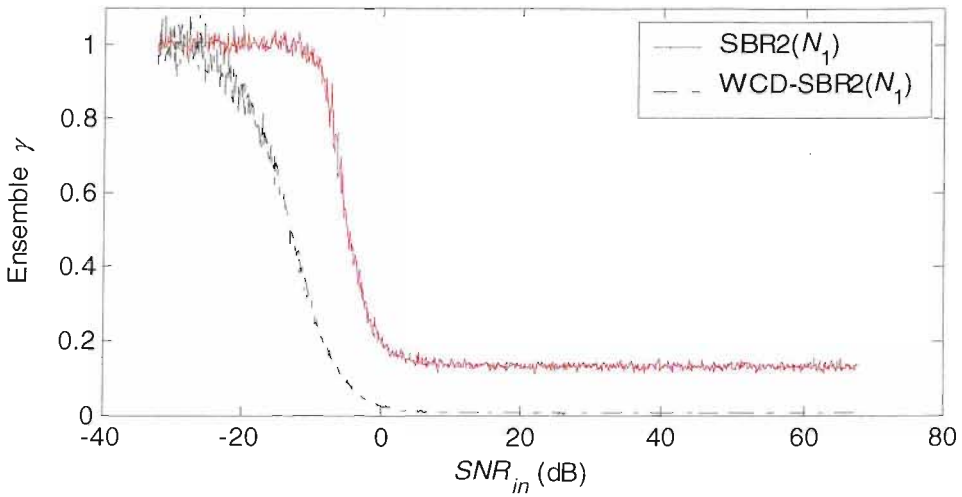


Figure 5.18: A comparison of subspace decomposition performances for regular SBR2 (solid curve) and WCD-SBR2 (dashed curve).

5.8 Concluding Remarks

In this chapter, we have introduced the second-order sequential best rotation (SBR2) algorithm as a method of constructing paraunitary matrices for polynomial matrix diagonalisation (or PEVD). In contrast to frequency-domain methods presented in chapter 4, the SBR2 algorithm is a time-domain approach that is not restricted to circulant (periodic) data. It uses the decomposition in [73] but with a fundamental difference that allows for a much simpler method of parameter optimisation: a generalisation of the classical Jacobi algorithm to the relatively unexplored domain of polynomial matrices. The SBR2 algorithm constructs a paraunitary matrix as a sequence of elementary paraunitary matrices computed one at a time. It can strongly decorrelate signals, to a good degree, in a perfectly conditioned manner and has a proof of convergence. The algorithm has been shown to give an effective extension of the EVD to polynomial matrices or the SVD to broadband signals.

We have introduced variations on the SBR2 algorithm which improve its strong decorrelation and spectral majorisation (and thus the broadband signal subspace estimation) performances. One of these variants is created by changing the algorithm's correlation-based

cost function to one that is based on the coding gain. Another version of the algorithm is proposed that applies a window to the sample space-time covariance matrix of the input data.

In this chapter we have only attempted to indicate the relevance of the SBR2 algorithm to signal processing by means of some typical examples. However, as with the conventional SVD/EVD, the SBR2 algorithm is a generic tool, applicable to a wide range of problems. It has already been applied successfully to data obtained from real sensor arrays in a number of application areas including sonar and seismology; however, discussion of the specific applications and results is beyond the scope of this thesis. It has also been adopted successfully by other researchers for the purpose of designing oversampled filter banks for channel coding [41,85] and for second order blind signal separation, applied to polarised signals from a 3-axis seismic sensor array using quaternion (hypercomplex) arithmetic [39]. It is envisaged that the SBR2 algorithm can also be applied to more esoteric applications, such as cancer diagnosis.

The SBR2 algorithm has the effect of compacting most of the total signal power into as few channels as possible. This implies that it is also suitable for application to multichannel data compression and subband coding. In chapter 6, we focus on extending the capability of the SBR2 algorithm to data compression, in particular the design of optimal orthonormal subband coders, in the following chapter.

6 PEVD for Data Compression

In chapter 5, the SBR2 algorithm was introduced as a time-domain approach to computing the PEVD of parahermitian matrices. The algorithm calculates a paraunitary filter bank that imposes, to a good approximation, strong decorrelation and spectral majorisation upon the input signals. In chapter 3, it was shown that these qualities are necessarily satisfied by the analysis bank of an optimal orthonormal subband coder, where optimality is in the sense of maximising the coding gain. In this chapter, we introduce variations on the SBR2 algorithm that are suitable for application to both multichannel data compression and subband coding. Correspondingly, the chapter is sectioned into two parts: in section 6.1, an investigation into the applicability of the SBR2 algorithm to the task of compressing multichannel arbitrary data is presented; in section 6.2, we introduce the SBR2 coder, which is an adaptation of the SBR2 algorithm for effective application to the problem of optimal subband coding. The SBR2 coder represents a major part of the innovation presented in this thesis.

In section 6.1, we begin by recognising that energy compaction and minimal data encoding for multichannel data are possible tasks for a PEVD algorithm. The connection between the PEVD and the PCFB, introduced in chapter 3, is highlighted. A new measure of data compression for multichannel arbitrary data is defined by extension of the coding gain given by (3.7). The coding ability of the various SBR2 algorithms is demonstrated through experimental results. In section 6.2, the SBR2 coder is introduced as a technique for producing orthonormal PRFBs for subband coding. Firstly, an examination of the effect of the delay-chain and decimation (demultiplexing) stage of a subband coder on the statistics of the input signal is given. We show that the space-time covariance matrix of the input vector signal has a special structure when the input signal is stationary in the wide sense. This is exploited by conventional subband coder design methods. We introduce a method of incorporating knowledge of this structure into the SBR2 algorithm. The resulting algorithm is coined the SBR2 coder. Certain relationships between the SBR2 coder and a prior-art algorithm proposed in [32] for designing compaction filters, called the window method, are then given. Finally, results are included showing that the SBR2 coder can produce filter banks that outperform those obtained using the window method for a set of benchmark problems.

6.1 Data Encoding of Multiple Signals

The correlations that exist between multiple signals constitute a type of redundancy. A linear transformation, such as the SVD, may be used to remove these correlations and perform optimal energy compaction (see chapter 2). As a result, most of the total input signal power resides in a reduced number of channels (dominant channels). Data compression may be achieved by encoding (quantising) the data with a reduced number of bits: typical strategies for quantisation have been discussed in section 3.2.

However, if the input multichannel data are strongly correlated, then instantaneous decorrelation would not yield optimal compression. For this case, optimality may be attained with a filter bank that imposes strong decorrelation and spectral majorisation upon the input signals – see section 4.2.3 for a definition of these two properties in the context of arbitrary multichannel data. It is also desirable for the filter bank to be paraunitary because it simplifies considerably the design of a stable PRFB and prevents the amplification of quantisation noise (see chapter 3). Such a filter bank can be found by the SBR2 algorithm, as shown in chapter 5. In this section, we evaluate the SBR2 algorithm for data compression of multichannel data.

6.1.1 Energy Compaction and Coding

It has been shown that the PCFB is the optimal filter bank for both energy compaction and subband coding, as discussed in section 3.2.3. Optimality in terms of the former objective is achieved when the partial sum in (3.13) is maximised. An optimal solution to the latter objective is found because a PCFB also maximises the non-uniformity of the variances of all the subband signals, and so the product of these variances is minimised (i.e. the coding gain in (3.7) is maximised). An algorithm for estimating a PEVD of arbitrary multichannel data may also be designed with these two objectives. This is now considered through trivial extension of these objective functions.

The energy compaction property of the PEVD (see section 4.2.3) is highly relevant and beneficial to the task of data compression. The PEVD produces a paraunitary matrix $\mathbf{H}(z)$ such that for all n less than M, the sum of the powers of the first n of the transformed channels is as high as possible, i.e. the partial sum

$$\sum_{k=1}^{n} c_{kk0} \quad \text{for each } n \leq M \tag{6.1}$$

is maximised. Here, M is the total number of arbitrary broadband data channels and $C_{kk}(z) = \sum_{l=-\infty}^{\infty} c_{kkl} z^{-l}$ are the diagonal entries of the true polynomial covariance matrix $\mathbf{C}(z)$ defined in (4.12) (related to the signals transformed by the PEVD). Now suppose that the

signals are reconstructed by applying $\tilde{\mathbf{H}}(z)$ (the paraconjugate transpose of $\mathbf{H}(z)$) to the reduced set of n transformed signals $\underline{V}(z)$, i.e. $\underline{Y}(z) = \tilde{\mathbf{H}}(z)\underline{V}(z)$. Since (6.1) is maximised and $\mathbf{H}(z)$ is paraunitary, the sum of the variances of the $M-n-1$ signals is minimised. It follows directly that the error in the reconstructed signal $\underline{Y}(z)$ is minimised, i.e. the multichannel reconstruction error $\varepsilon = (1/M)\sum_{k=n+1}^{M} c_{kk0}$ is minimised.

A paraunitary matrix that minimises the multichannel reconstruction error may be looked upon as an extension of the PCFB for subband coding to arbitrary multichannel data. It follows that we can make the following assertions:

- In the particular case when $n=1$ and c_{110} is maximised, the first row (polynomial vector) of $\mathbf{H}(z)$ is a generalisation of the optimal compaction filter (defined in section 3.2.2) to arbitrary multichannel signals.
- The sum

$$\sum_{k=1}^{M} \log_2 c_{kk0} \tag{6.2}$$

  is minimised. This is a simple generalisation of the un-normalised coding gain in (3.7) to arbitrary multichannel signals.

Since we are limited to the case where $\mathbf{H}(z)$ is comprised of FIR filters we cannot in general minimise ε and thus obtain optimal multichannel data compression, which is analogous to the intermediate condition for PCFB design given in section 3.2.3. If the aim is to produce minimally coded data, then a PEVD-estimating algorithm would be at an advantage if it aimed to maximise c_{110} (i.e. energy compaction) or minimise (6.2). It follows from the discussions in sections 3.2.2 that such an algorithm would implicitly impose strong decorrelation and spectral majorisation. The SBR2(G_c) algorithm, introduced in 5.3.2, would be suitable for the task of data compression since it uses the G_c cost function defined in (5.31), which is based on the coding gain measure. It stands to reason then that this algorithm aims to minimise (6.2).

6.1.2 Performance Measure Based on the Coding Gain

The methods for computing the PEVD introduced in sections 4.4 and chapter 5, produce filter banks that can be used to perform multichannel data compression. A measure is required to assess the data compression ability of the filter bank constructed by a PEVD algorithm. Recall from section 4.2.4 that the sum of the magnitude-square off-diagonal terms at all time delays gives a measure of the diagonality of a polynomial matrix, as represented by λ in (4.33). This quantity cannot be used for the purpose of measuring data compression performance since it is

proportionately more sensitive to changes in signals with high energy. It follows, therefore, that it is weakly sensitive to spectral majorisation, which is a necessary property when trying to correctly gauge the level of data compression achieved. Therefore, an alternative performance measure is defined, called the *compression gain*. The measure may be looked upon as a generalisation of the coding gain to multichannel data. It allows for the direct comparison of the data compression performance of the various PEVD-estimating techniques presented in this thesis.

The compression gain is defined as a measure of the number of bits per sample saved after encoding the data channels. This is provided that the original information content is faithfully reproducible. The compression gain, G_m, is simply the difference between the number of bits required to encode the input channels, N_i, to that required for the output channels, N_o. In other words,

$$G_m = N_i - N_o \text{ (number of bits).} \tag{6.3}$$

The measure may be expressed in terms of the variances of the input and output (transformed) channels, a_{kk0} and c_{kk0}, respectively: the zero-lag coefficient of the k^{th} diagonal entry of the true covariance matrices $\mathbf{A}(z)$ and $\mathbf{C}(z)$ defined in sections (4.10) and (4.13), respectively. Now, the number of bits/sample (data rate) required for M data channels before and after strong decorrelation may be expressed as, respectively,

$$\begin{aligned} N_i &= \log_2(q\sqrt{a_{110}}) + \log_2(q\sqrt{a_{220}}) + \ldots + \log_2(q\sqrt{a_{MM0}}) \\ &= \log_2(\sqrt{a_{110}a_{220}\ldots a_{MM0}}) + MQ\,. \end{aligned} \tag{6.4}$$

and

$$N_o = \log_2(\sqrt{c_{110}c_{220}\ldots c_{MM0}}) + MQ\,. \tag{6.5}$$

Here, q is a constant associated with the signal digitisation process, such as quantisation, and $Q = \log_2(q)$. We define the compression gain as

$$G_m = N_i - N_o = \frac{1}{2}\log_2\left(\frac{\prod_{k=1}^{M} a_{kk0}}{\prod_{k=1}^{M} c_{kk0}}\right). \tag{6.6}$$

This measure is maximised when the product of the variance of the transformed output channels is minimised. In Appendix 8.1, we prove that maximisation of the compression gain leads to paraunitary filter bank that is optimal in the sense of the compression gain.

Cost Function Based on True Statistics

Another variation on the SBR2 algorithm can be created by using the G_m measure in (6.6) as the cost function. This SBR2 variant belongs to the class of non-blind SBR2 algorithms defined earlier in section 5.3.3, and is referred to as SBR2(G_m). Viewed differently, the algorithm operates on the true space-time covariance matrix $\mathbf{C}(z)$ with the G_c cost function defined in (5.31). Therefore, the G_m measure is maximised when the largest normalised crosscovariance term in $\mathbf{C}(z)$ is zeroed. The SBR2(G_m) algorithm finds an 'almost'-optimal filter bank for multichannel data compression. The performance of this algorithm is an upper bound on the compression gain performance of the 'blind' SBR2 algorithms.

6.1.3 Performance Analysis

In the following we present the results of numerical simulations. The systems under investigation are the regular SBR2 and WCD-SBR2 algorithms. The inputs were three complex-valued data channels as generated in section 5.7. The WCD-SBR2 algorithms were run with the parameters as given in section 5.7. As before, expected performance quantities were obtained by taking the average across 100 trials of the experiments.

The evolution of the compression gain G_m for the SBR2(N_1), SBR2(G_c), WCD-SBR2(N_1) and WCD-SBR2(G_c) algorithms over a number of algorithm iterations L is shown in Fig. 6.1. The performance of the non-blind technique, SBR2(G_m), is also shown. As seen for the λ performance in section 5.7, there is an optimal number of steps, L', required at which point the regular SBR2 algorithms achieve the highest value of G_m obtainable with these algorithms. Notice that the optimum performance point for the SBR2(G_c) algorithm (thick solid curve) is obtained with a greater number of iterations than is required for SBR2(N_1) (thick dashed curve). As expected, the SBR2(G_c) performs data compression better than SBR2(N_1). The cause of the reduction in performance either side of L' (where the peak occurs) for the regular SBR2 algorithms are for the same reasons as cited in section 4.4.

Notice also that the WCD-SBR2 algorithms are superior to the regular SBR2 algorithms at data compression. This is attributed to the windowing performed on the sample covariances. Their compression gain performances asymptote to values just below optimality (as indicated by the thick dash-dot curve for SBR2(G_m)). The filter banks produced for this part of the curve attain high levels of energy compaction, as can be ascertained by comparing the variances of the input and output signals in Fig. 6.2. In general, the WCD-SBR2(N_1) algorithm achieves marginally better performance than that of the WCD-SBR2(G_c). This is because a rectangular window is more suited to the true autocovariance of the sources (providing a slightly better

estimate of the true space-time covariance matrix), as explained in section 5.7.1. The success of SBR2(G_m) is attributed solely to the fact that true statistics are used to design a near-optimal compaction filter bank. Results verifying the fulfilment of strong decorrelation and spectral majorisation for some of the algorithms evaluated here are given in section 5.7.

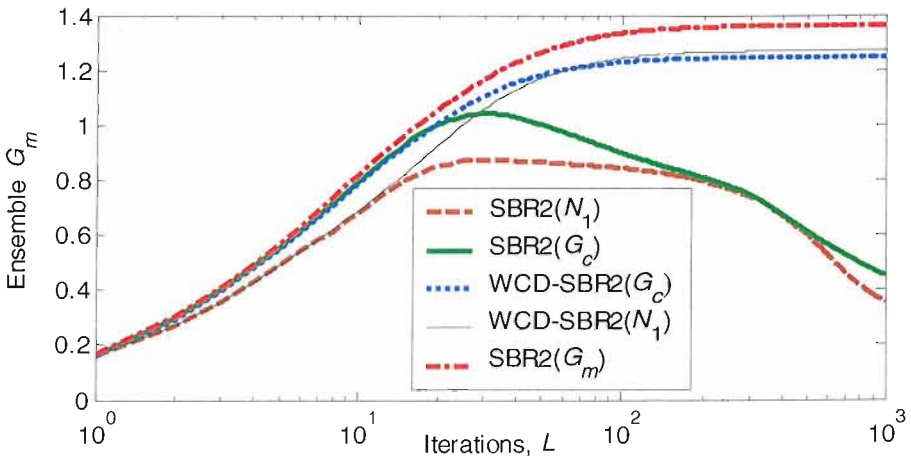


Figure 6.1: Comparison of the compression gain performances of the various SBR2 algorithms for complex valued data.

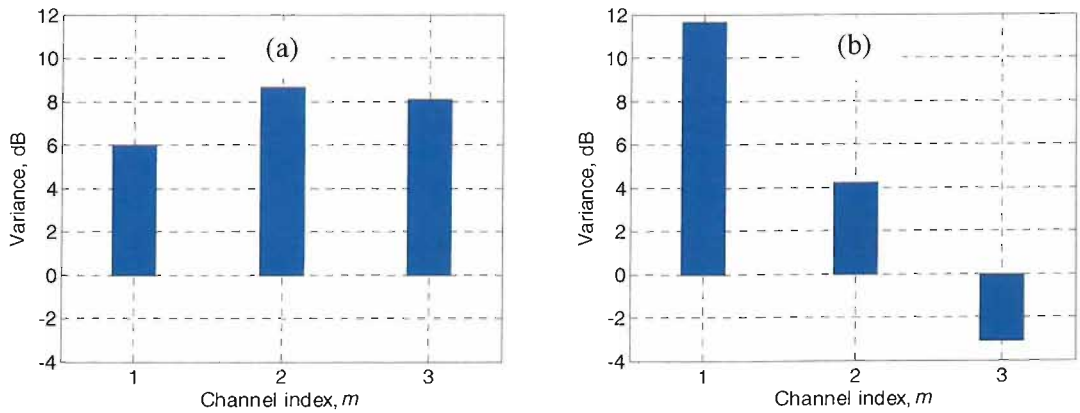


Figure 6.2: Total true power of each (a) input signal (a_{kk0}) and (b) output signal (c_{kk0}).

6.2 SBR2 Coder

In this section, we present an investigation into the applicability of SBR2 to optimal subband coding. A study of its relevance to the subband coder design algorithms found in the literature is presented, particularly its relevance to the window method, described in section 3.4.2, is assessed. Important differences between the window method and SBR2 (in relation to the

architecture) are identified and explained. Specifically, we show that the true polynomial covariance matrix of the demultiplexed signals $\mathbf{x}(t)$ in Fig. 3.1(b) has a special form for WSS signals, which is exploited by the window method. It is then shown that the SBR2 algorithm, as it is, is oblivious to this property, and so would not, in general, perform data compression as well as the window method. This motivates the development of a data pre-processing stage that exploits the extra knowledge about the statistics of the input signal. The relative performances of the modified SBR2 algorithm and the window method are then assessed.

6.2.1 Covariance of the Demultiplexed Signals

Pseudocirculant Matrices

Consider the $M\times M$ polynomial matrix $\mathbf{A}(z)$ with entries $A_{km}(z)$, where $\{k\in\mathbb{Z}\,|\,1\le k\le K\}$ and $\{m\in\mathbb{Z}\,|\,1\le m\le M\}$. From [73], the matrix $\mathbf{A}(z)$ is said to be pseudocirculant if there exists $\Phi_1(z),\Phi_2(z),\ldots,\Phi_M(z)$ such that

$$A_{km}(z)=\begin{cases}\Phi_{m-k+1}(z), & 1\le k\le m\\ z^{-1}\Phi_{m-k+M+1}(z), & m<k\le M.\end{cases} \tag{6.7}$$

In words, $\mathbf{A}(z)$ is a circulant matrix except that the entries below the main diagonal are multiplied by z^{-1}. Thus, a pseudocirculant matrix has the form

$$\mathbf{A}(z)=\begin{pmatrix}A_{11}(z) & A_{12}(z) & \cdots & A_{1M}(z)\\ A_{21}(z) & A_{22}(z) & \cdots & A_{2M}(z)\\ \vdots & \vdots & \ddots & \vdots\\ A_{M1}(z) & A_{M2}(z) & \cdots & A_{MM}(z)\end{pmatrix}=\begin{pmatrix}\Phi_1(z) & \Phi_2(z) & \cdots & \Phi_M(z)\\ z^{-1}\Phi_M(z) & \Phi_1(z) & \cdots & \Phi_{M-1}(z)\\ \vdots & \vdots & \ddots & \vdots\\ z^{-1}\Phi_2(z) & z^{-1}\Phi_3(z) & \cdots & \Phi_1(z)\end{pmatrix}. \tag{6.8}$$

The entries of $\mathbf{A}(z)$ that are equal, but for a permutation, are termed *related entries*; e.g., in the 4×4 case, the set $\{A_{13}(z),A_{24}(z),A_{31}(z),A_{42}(z)\}$ consists of related entries.

Properties of the Demultiplexed Covariances

In the following, we will show that the true polynomial covariance matrix of the vector signal $\mathbf{x}(t)$ is pseudocirculant if the input signal $x(t)$ is WSS, as was first discovered in [60]. This result will reveal the importance of pseudocirculants in the design of optimal subband coders; moreover, it will expose a performance limiting factor of the SBR2 algorithm if applied to problem of subband coding.

Consider the input signal $x(t)$ and the analysis bank of the subband coder in Fig. 3.1. Let $x_p(t)=x(Mt+p-1)$ $\{p\in\mathbb{Z}\,|\,1\le p\le M\}$ be the demultiplexed (low sample rate) signals,

which are naturally elements of the vector signal $\mathbf{x}(t) = [x(Mt)x(Mt+1),\ldots,x(Mt+M-1)]^{\mathrm{T}}$: the polyphase components of $x(t)$. Suppose that $x(t)$ is drawn from a zero-mean WSS process. The vector signal $\mathbf{x}(t)$ is, therefore, also zero-mean and WSS. The true autocovariance function of $x(t)$ is defined as

$$\mathrm{E}\left[x(t)x^{*}(t+\tau)\right] = a(\tau), \quad \forall t \text{ and } \forall \tau . \tag{6.9}$$

The space-time (or rather, time-time) covariance matrix for the polyphase components of $x(t)$ may be written as

$$\mathrm{E}\left[\mathbf{x}(t)\mathbf{x}^{\mathrm{H}}(t+\tau)\right] = \mathbf{a}(\tau) \in \mathbb{C}^{M\times M}(\tau), \forall t \text{ and } \forall \tau . \tag{6.10}$$

The z-transform of (6.10) is given by

$$\mathbf{A}(z) = \sum_{\tau} z^{-\tau}\mathbf{a}(\tau), \quad \forall \tau . \tag{6.11}$$

Theorem 6.1. *The matrix* $\mathbf{A}(z)$ *is pseudocirculant if* $x(t)$ *is a WSS signal.*

Proof. The $(k,m)^{\text{th}}$ entry of $\mathbf{a}(\tau)$ can be expressed as

$$a_{km}(\tau) = \mathrm{E}\left[x_k(t)x_m^{*}(t+\tau)\right], \quad \forall \tau . \tag{6.12}$$

The relationship between these entries and the samples of $x(t)$ may be expressed as

$$a_{km}(\tau) = \mathrm{E}\left[x\big(Mt+(k-1)\big)x^{*}\big(M(t+\tau)+(m-1)\big)\right] . \tag{6.13}$$

Since $x(t)$ is WSS, (6.13) can be rewritten for $1 \le k \le m$ as

$$\begin{aligned} a_{km}(\tau) &= \mathrm{E}\left[x(Mt)x^{*}\big(M(t+\tau)+(m-k)\big)\right] = a\big(M\tau+(m-k)\big) \\ &= \varphi_{m-k+1}(\tau) = \varphi_{p}(\tau) = a\big(M\tau+(p-1)\big). \end{aligned} \tag{6.14}$$

In the z-domain, (6.14) becomes

$$A_{km}(z) = \sum_{\tau} z^{-\tau}a_{km}(\tau) = \sum_{\tau} z^{-\tau}a\big(M\tau+(m-k)\big) = \sum_{\tau} z^{-\tau}\varphi_{m-k+1}(\tau) = \Phi_{m-k+1}(z). \tag{6.15}$$

For $m < k \le M$, the polyphase index $(m-k)$ in (6.15) is negative, but $m-k+M$ is positive, hence we have a new expression for the entries of $\mathbf{A}(z)$ given by

$$A_{km}(z) = \sum_{\tau} z^{-\tau}a_{km}(\tau) = \sum_{\tau} z^{-\tau}a\big(M\tau+(m-k)\big) = \sum_{\tau} z^{-\tau}a\big(M\tau-M+(m-k+M)\big)$$

$$= \sum_{\tau} z^{-\tau} a\big(M(\tau-1)+(m-k+M)\big) = \sum_{\tau'} z^{-\tau'-1}\varphi_{m-k+M+1}(\tau')$$

$$= z^{-1}\sum_{\tau'} z^{-\tau'}\varphi_{m-k+M+1}(\tau') = z^{-1}\Phi_{m-k+M+1}(z)\,. \tag{6.16}$$

From (6.7) we conclude that $\mathbf{A}(z)$ is pseudocirculant. Q.E.D.

There are some key points to note regarding the estimation of $A(z)$ using the entries in $\mathbf{A}(z)$:

1. The decimation process for the analysis bank causes the loss of information in each individual entry of $\mathbf{A}(z)$. Its diagonal and off-diagonal entries are computed using only, respectively, $(1/M)^{\text{th}}$ and $(2/M)^{\text{th}}$ of the information available.
2. The diagonals of $\mathbf{A}(z)$ are identical (i.e. an M fold redundancy of information) and represent an estimate of $A(z)$.
3. Due to the pseudocirculant structure of $\mathbf{A}(z)$, there is extra (useful) information about $A(z)$ in the combination of related entries, which can be used to determine $A(z)$.

The sample autocovariance function for the input signal $x(t)$ may be expressed as

$$R(z) = \sum_{\tau=-t_1}^{t_1}\left[\frac{1}{T}\sum_{t=0}^{T-1} x(t)x^{*}(t+\tau)\right]z^{-\tau}, \quad T\in\mathbb{Z}, \quad t_1 \geq 0, \quad t_1 \leq T, \quad \tau\in\mathbb{Z}. \tag{6.17}$$

The sample polynomial covariance matrix for the demultiplexed signals $\mathbf{x}(t)$ is given by

$$\mathbf{R}(z) = \sum_{\tau=-t_1}^{t_1}\mathbf{r}(\tau)z^{-\tau} \in \mathbb{C}^{M\times M}(\tau), \quad \tau\in\mathbb{Z}, \tag{6.18}$$

where

$$\mathbf{r}(\tau) = \frac{M}{T}\sum_{t=0}^{T-1}\mathbf{x}(t)\mathbf{x}^{\mathrm{H}}(t+\tau)\,. \tag{6.19}$$

There are points worthy of mention about the estimation of the autocovariance $A(z)$ using $\mathbf{R}(z)$:

1. As with the construction of the matrix $\mathbf{A}(z)$, the diagonal and off-diagonal entries of $\mathbf{R}(z)$ are computed using only, respectively, $(1/M)^{\text{th}}$ and $(2/M)^{\text{th}}$ of the information available. Since there are fewer samples to estimate $\mathbf{A}(z)$, and thus $A(z)$, the sample noise in the estimates is amplified.

2. The diagonal entries of $\mathbf{R}(z)$ are different noisy estimates of $A(z)$. Hence, they may be averaged to form a better estimate of $A(z)$.

The related entries of $\mathbf{R}(z)$ are different estimates of the cross-terms in $\mathbf{A}(z)$. Hence, by taking the mean across the related entries one can obtain a more accurate estimate of the true crosscovariances.

6.2.2 SBR2 Applied to Subband Coding

In chapter 5, we showed that, given a set of M arbitrary sequences, the SBR2 algorithm can generate an $M \times M$ paraunitary polynomial matrix, $\mathbf{E}(z)$. The paraunitary matrix can be applied to the signals to produce strongly decorrelated and spectrally majorised signals, to a good approximation. In other words, the algorithm aims to perform PEVD on the input sample covariance matrix $\mathbf{R}(z)$ expressed in (5.1). The accuracy of the decomposition depends on the accuracy of its estimate of the true input covariance matrix.

The SBR2 algorithm may be categorised as a lattice parameterisation method – see section 3.4.1 – since the algorithm constitutes the optimisation of parameters of a lattice structure. To further relate these algorithms, we may regard the optimisation performed by SBR2 as being eigenstructure based. This is because it essentially aims to decompose a covariance matrix into its (polynomial) eigenvalues and eigenvectors. The SBR2 algorithm can be classed as a blind technique since it does not use knowledge about the signals or the mixing matrix. Furthermore, its formulation is not based on knowledge of the input signal statistics save for the minor requirement that the mean value of the signals is zero, i.e. $E[x(t)] = 0$.

Consider now using SBR2 to design the polyphase analysis bank for the subband coder in Fig. 3.1. The aim is to perform optimal (or near-optimal) data compression (in the sense of maximising the coding gain in (3.7)) on the single input signal $x(t)$. One way of achieving this is to apply the paraunitary filter bank produced by SBR2 directly to the demultiplexed signals, that is

$$\underline{V}(z) = \mathbf{E}(z)\underline{X}(z), \tag{6.20}$$

where $\underline{X}(z)$ is a vector of algebraic power series related to the vector signal $\mathbf{x}(t)$. Here, $\mathbf{E}(z)$ is treated as the analysis (polyphase) matrix of the subband coder, which is illustrated in Fig. 6.3. A high coding gain may be achieved if SBR2 is allowed to operate on a good approximation to $A(z)$. Recall that the sample covariance matrix for the demultiplexed signals, $\mathbf{R}(z)$, represents a somewhat noisier and distorted estimate of $A(z)$ than that derived directly

from $x(t)$ (the sample autocovariance $R(z)$). If applied in the conventional way, the filter bank produced by SBR2 will be based on $\mathbf{R}(z)$ and not on $R(z)$.

In the above regime, SBR2 does not use knowledge that the demultiplexed signals $x_k(t)$ are jointly WSS to gain a better estimate of $A(z)$ since it is a blind algorithm. That is, the SBR2 algorithm does not exploit the special form (pseudocirculants) of $\mathbf{A}(z)$. Therefore, in general, the SBR2 algorithm will not perform data compression as proficiently as those techniques whose filters are based on $R(z)$, such as the window method in section 3.4.2. This is demonstrated by the following example.

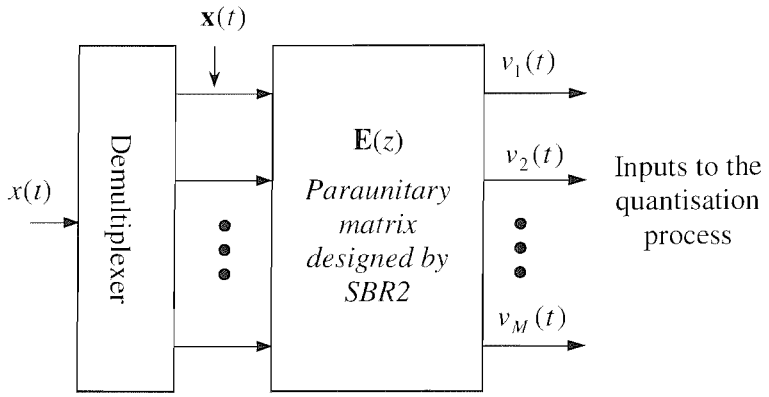


Figure 6.3: SBR2 can design the analysis bank of a subband coder by direct application to the M polyphase channels.

Example: ARMA(5) process. The window method and the WCD-SBR2(N_1) are used to design a two-channel ($M=2$) orthonormal filter bank for the ARMA(5) example given in section 3.4.2. The SBR2 algorithm was applied directly to the vector signal $\mathbf{x}(t)=[x_1(t),x_1(t),\ldots,x_M(t)]^{\mathrm{T}}$, as shown in Fig. 6.3. The algorithm produced a 2×2 paraunitary polynomial matrix $\mathbf{E}(z)$, with polyphase filters $E_{km}(z)$.

The coding gain performances of the two algorithms have been evaluated, the results from which are shown in Fig. 6.4. In the case of SBR2, the coding gain was computed for a number of iterations. For the window method, it was evaluated for a number of filter orders. The coding gain was computed directly from the statistical process (i.e. it is based on the true statistics of the data). The experiment was repeated 70 times and the mean value of the measure was graphed. The red dotted (horizontal) line represents the theoretical (ideal) coding gain $\hat{G}=1.94$, which is the maximum attainable coding gain – see the next section for a derivation of this.

As expected, the coding gain curves produced by the algorithms are below the theoretical value. It is evident that the window method is able to achieve greater levels of data compression than the SBR2 algorithm for filter orders (or the number of SBR2 steps) greater than ~50; however, for filter orders (number of iterations) less than ~50 the SBR2 algorithm can attain higher coding gains. We also observe that the filter bank produced by WCD-SBR2(N_1) asymptotes to the optimal one as the number of SBR2 steps increase. There seems to be a slight dip in the performance of the window method for large N, the reason for which is discussed later in section 6.2.5.

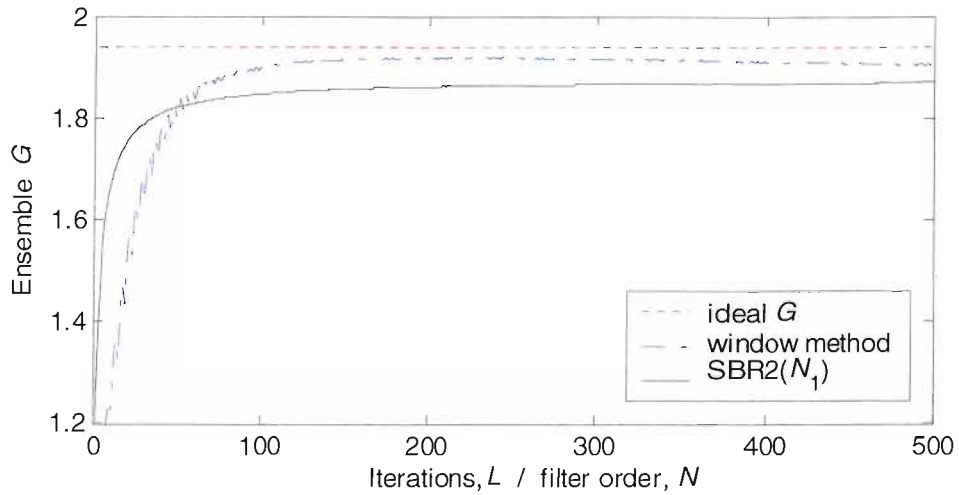


Figure 6.4: Comparison of coding gain performances between the window method and WCD-SBR2(N_1) algorithm for $M = 2$ and an ARMA(5) input process.

Exploitation of Signal Statistics

The SBR2 algorithm may be modified to exploit the stationarity of the input signal, i.e. take advantage of the pseudocirculant structure of $\mathbf{A}(z)$. A set of related entries in $\mathbf{R}(z)$ are different estimates of the same true cross-covariance. Therefore, to improve on the estimate of $\mathbf{A}(z)$, averaging may be performed across the related entries for all time lags. This would involve taking account of the delay between related entries either side of the main diagonal. It is instructive to examine the operations on $\mathbf{R}(z)$ for $M = 2$. Let the entries in $\mathbf{R}(z)$ be denoted $r_{km}(\tau)$. The diagonals $r'_{11}(\tau)$ and $r'_{22}(\tau)$ of the modified matrix $\mathbf{R}'(z) \in \mathbb{C}^{2\times 2}(z)$ are computed as

$$r'_{11}(\tau) = r'_{22}(\tau) = \left(r_{11}(\tau) + r_{22}(\tau)\right)/2 \qquad \forall \tau . \tag{6.21}$$

The off-diagonals are computed as, respectively,

$$r'_{12}(\tau) = \left(r_{12}(\tau) + r_{21}(\tau+1)\right)/2 \qquad \forall \tau \tag{6.22}$$

and

$$r'_{21}(\tau) = \left(r_{12}(\tau-1) + r_{21}(\tau)\right)/2 \qquad \forall \tau\,. \tag{6.23}$$

For the general $M \times M$ case, we define

$$\varphi_p(\tau) = \frac{1}{M}\left(\sum_{k=1}^{M-p} r_{k,k+p}(\tau) + \sum_{k=M-p+1}^{M} r_{k,k+p-M}(\tau+1) \right), \quad 0 \le p \le M-1 \tag{6.24}$$

and a typical entry of the new (averaged) sample polynomial covariance matrix $\mathbf{R}'(z)$ as

$$R'_{k,m}(z) = \begin{cases} \sum_\tau \varphi_{m-k}(\tau) z^{-\tau}, & 1 \le k \le m \le M \\ \sum_\tau \varphi_{m-k+M}(\tau) z^{-\tau-1}, & 1 \le m < k \le M\,. \end{cases} \tag{6.25}$$

The SBR2 algorithm can now be applied to our improved estimate of the covariance matrix $\mathbf{R}'(z)$. This modification yields the *SBR2 coder*. A schematic diagram of the process blocks involved in producing an analysis filter bank for subband coding using the SBR2 coder is shown in Fig. 6.5.

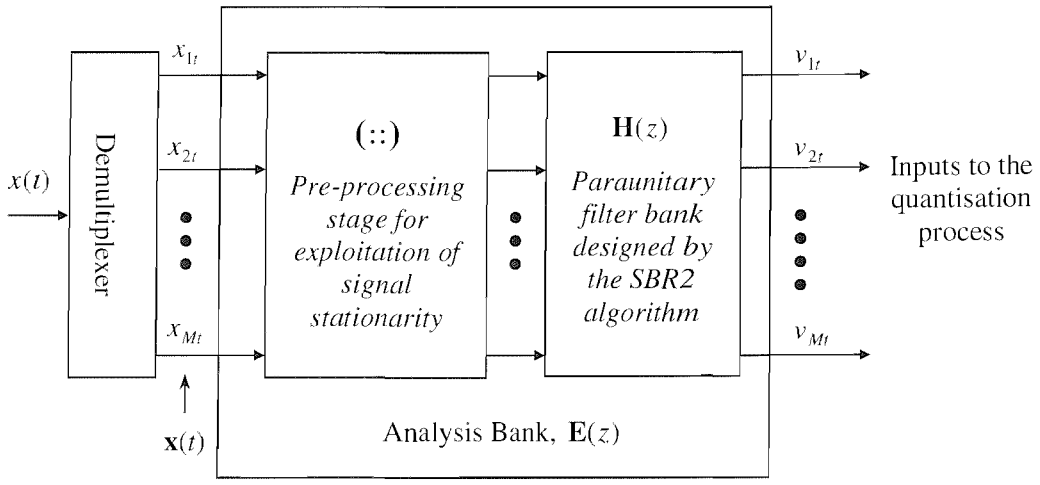


Figure 6.5: The SBR2 coder.

Conjecture 6.1. *The averaged sample polynomial covariance matrix is parahermitian.*

Illustration. It is easy to see that $\mathbf{R}'(z)$ is parahermitian by considering the two channel example above. We have that

$$2R'_{1,2}(z) = R_{1,2}(z) + zR_{2,1}(z) \tag{6.26}$$

and

$$2R'_{2,1}(z) = zR_{2,1}(z) + R_{1,2}(z) = z\big(R_{2,1}(z) + z^{-1}R_{1,2}(z)\big). \tag{6.27}$$

It is now easy to see that

$$2R'_{2,1}(z) = z\big(\tilde{R}_{1,2}(z) + z\tilde{R}_{2,1}(z)\big) = 2\tilde{R}'_{1,2}(z)\,. \tag{6.28}$$

6.2.3 Theoretical (Ideal) Performance

Our aim is to determine the theoretically optimal or *ideal* values for the compaction gain (related to the input and output of the first filter $H_1(z)$) in (3.12) and the coding gain in (3.7) and for the M-band subband coder of Fig. 3.1. These ideal values would be attained if the theoretically optimal transform were to be applied to the demultiplexed signals $x_k(t)$. The Fourier transform of these signals may be expressed in terms of the Fourier transform of the input $x(t)$:

$$X_k(e^{j\omega}) = \frac{1}{M}\sum_{\ell=0}^{M-1} X(e^{j\frac{(\omega-2\pi\ell)}{M}})\,. \tag{6.29}$$

The first term ($\ell = 0$) in (6.29) is of course an M-fold stretched version of $X(e^{j\omega})$, and the other (aliasing) terms ($\ell > 0$) are uniformly shifted versions of the first term. Suppose that the variances of the signals $X_k(e^{j\omega})$ are permuted at each frequency such that the spectra $\{\hat{C}_k(e^{j\omega})\}$ of the resulting (transformed) signals $\hat{V}_k(e^{j\omega})$ have the spectral majorisation property:

$$\hat{C}_1(e^{j\omega}) \geq \hat{C}_2(e^{j\omega}) \geq \cdots \geq \hat{C}_M(e^{j\omega})\,, \quad \forall\omega, \tag{6.30}$$

where $\hat{C}_k(e^{j\omega}) = \sum_{\tau=-\infty}^{\infty} \mathrm{E}\big[\hat{v}_k(t)\hat{v}_k^*(t+\tau)\big]z^{-\tau}\Big|_{z=e^{j\omega}}$. The mean spectral power (true variance) for the transformed signals and the input signal may be expressed as, respectively,

$$\hat{c}_{kk0} = \frac{1}{T}\sum_{\ell=0}^{T-1} \hat{C}_k(e^{2\pi\ell/T})\,, \quad \ell \in \mathbb{Z}, \tag{6.31}$$

and

$$a_0 = \frac{1}{M}\sum_{k=1}^{M}\frac{1}{T}\sum_{\ell=0}^{T-1} A_k(e^{2\pi\ell/T}) = \frac{1}{T}\sum_{\ell=0}^{T-1} A(e^{2\pi\ell/T})\,, \tag{6.32}$$

where $A_k(e^{j\omega}) = \sum_{\tau=-\infty}^{\infty} \mathrm{E}\left[x_k(t)x_k^*(t+\tau)\right]z^{-\tau}\Big|_{z=e^{j\omega}}$ and $A(e^{j\omega}) = \sum_{\tau=-\infty}^{\infty} \mathrm{E}\left[|x(t)|^2\right]z^{-\tau}\Big|_{z=e^{j\omega}}$. The ideal compaction gain is given by

$$\hat{G}_{comp} = \frac{\hat{c}_{110}}{a_0} \tag{6.33}$$

and the ideal coding gain may be expressed as

$$\hat{G} = \frac{a_0}{\left(\prod_{k=1}^{M} \hat{c}_{kk0}\right)^{1/M}} \tag{6.34}$$

An example for the two-channel case is provided in Fig. 6.6.

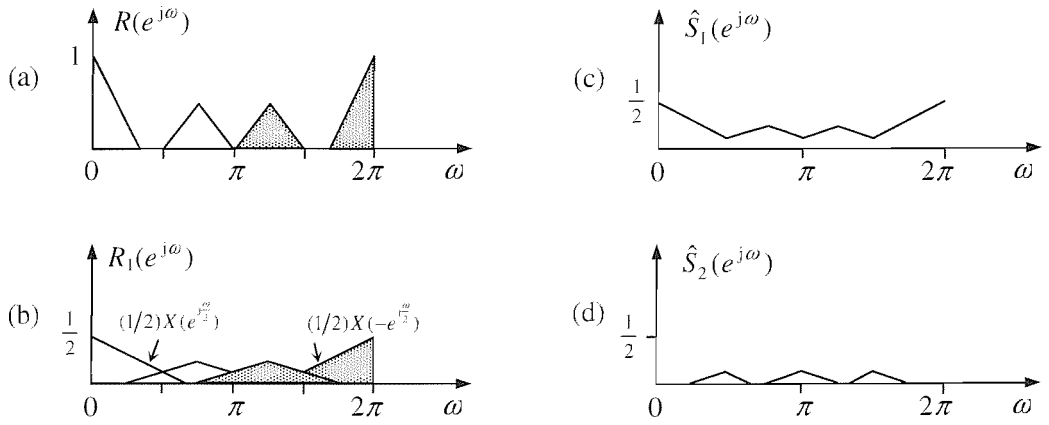


Figure 6.6: Illustration of the transform-domain effects for a theoretically optimal 2-channel subband coder and a multiband input process with PSD in (a). (b) PSD of the demultiplexed channels. (c) and (d) PSDs of the corresponding subband channels, which are spectrally majorised.

6.2.4 Comparison of the SBR2 Coder and the Window Method

Stationarity Assumption

The window method, as with other traditional subband coder design methods, has been formulated with the assumption that $x(t)$ is WSS. In other words, it intrinsically exploits the pseudocirculant structure of the covariance matrix $\mathbf{A}(z)$ for the subband signals in (6.8). An obvious drawback with the window method is that it is not suited to designing compaction filters for non-WSS signals. Using this algorithm in practical communications systems may

prove to be problematic, where in general the receiver (filters plus down-conversion system) may modify the statistics of the signal [51].

By contrast, the SBR2 algorithm can be used to design suboptimal paraunitary filter banks for arbitrary (zero-mean) signals; it does not exploit the prior information about the stationarity of the input-signal statistics. Although, as discussed in the previous section, esoteric knowledge about the pseudocirculants of the polynomial covariance matrix can be fed to the SBR2 algorithm by way of a simple averaging scheme, which gives the SBR2 coder. This is a fundamental difference between the SBR2 algorithm and the conventional design methods.

Filter Bank Efficiency

Evaluation of the window method has revealed that it is less complex than SBR2. It is expected, therefore, that the computation time required by the SBR2 algorithm in constructing a paraunitary filter bank for efficient coding would be greater than that taken by the window method. The fastest part of the window method is the application of the ideal compaction filter design. This is essentially a search for the frequency components with the greatest power; this entails a comparison of a set of baseband components with their images, where each component corresponds to a discrete design frequency. The window design and spectral factorisation routines in the algorithm are very simple and efficient algorithms as well. There is only one comparison to make at each of the design frequencies and the designed compaction filter can be used to determine the second filter. The computation time in determining the second filter is comparatively small and the complexity very simple. Consequently, the computational efficiency of the window method is significantly greater than that of SBR2.

In typical storage and communications systems, the level of compression achievable can sometimes have priority over the cost of computation. It is becoming more and more common to compress signals before their transmission since the available bandwidth of a communications link is at a premium. Therefore, in practice, an increase in computational cost may be justified by enhancement in the level of compression. With an adaptive filter bank design algorithm, like the window method and SBR2, information regarding how to construct the filter bank used would need to be stored or transmitted so that compression applied to the data may be undone (decompression). The amount of information required in describing the filter bank adds to the bandwidth required in transmitting the data.

Example. Consider the design of an M-channel filter bank with real filter coefficients for the task of data compression. The filters generated by the window method have been shown to be short for experiments presented in [32]: filter orders N of 31 and 65 were used to obtain near-optimal compaction filters. In the special case of two-channels, it is only necessary to store or transmit the first decimation filter, that is, $N+1$ real numbers. However, generally, for an M-

channel filter bank we would need to design M filters. Therefore, a maximum of $M(N+1)$ real numbers would need to be stored or communicated for faithful filter-bank reconstruction.

In the case of SBR2, it is the number of elementary paraunitary matrices required in producing the filter bank that is an important factor here and not the order of the filters. Recall that a single step of the SBR2 algorithm involves the application of a delay and a rotation. Consequently, after a predetermined number of iterations, L, SBR2 generates $2L$ parameters (assuming real coefficients) and the $2L$ indices (real numbers) for identifying the pair of signals to which each elementary paraunitary matrix is applied. Hence, $4L$ real numbers are required in order to reconstruct the filter bank generated by SBR2. Usually, these numbers would necessitate less storage space (or a narrower bandwidth) than that for filter coefficients. Note that information about the synthesis bank does not need to be conveyed because it can be determined from the analysis bank for paraunitary filter banks.

Paraunitarity and Stability

The window method is a technique for the design of optimal compaction filters. It cannot construct an M-channel filter bank. As a special case, the compaction filter found by the window method may be used to determine the second filter of a two-channel orthonormal filter bank – shown in section 3.3.2. Thus, paraunitarity is assured by construction. A possible scheme for the design of M-channel orthonormal subband coders has been given in [32]. The procedure introduced by Moulin and Mihcak in [48] obtains the remaining filters from the first (compaction filter). The design of the remaining filters is subject to the constraint that the filters are orthonormal. Therefore, we conjecture that such a technique (i.e. a multichannel window method) would produce paraunitary filter banks, as indeed does the SBR2 coder. This implies the following about the SBR2 coder and a 'potential' multichannel window method:

1. The algorithms do not modify the total signal power, and therefore cannot corrupt the signals.
2. The inverse of the filter bank can be used to obtain exactly the original signal, i.e. the algorithms can produce a PRFB for lossless data encoding.
3. The PRFB is guaranteed to be stable since the inverse of a paraunitary analysis bank is assured to be stable [73].

Causality and Phase

The application of an elementary paraunitary matrix can be restricted to only delaying one of the signals relative to the other (i.e. signals are not advanced). This ensures that the filter bank designed by SBR2 is causal and thus realisable. The window method guarantees causal

compaction filters [26]; because the spectral factorisation produces right-sided polynomials with real coefficients.

It is well known in the subband coding community that usually linear phase (i.e. constant group delay) and orthogonality are mutually exclusive design criteria for the construction of filter banks; each filter of a set of orthogonal filters generally has a non-linear phase response [73]. The SBR2 coder and the window method generally produce filters with asymmetric impulse responses, which is indicative of filters that have varying group delay. Therefore, it is expected that the filters designed by the SBR2 coder and a multichannel window method will in general have non-linear phase responses. Although the phase of the compaction filters is immaterial for the compaction gain and coding gain, it is vital in certain applications where minimal signal corruption is sought after, such as image compression [33].

Filter Order Constraint

Another important distinction to make between the two algorithms is that the SBR2 algorithm does not enforce a filter order constraint on the design directly. If the source signals are i.i.d. sequences, then the number of iterations of the algorithm required is essentially governed by the order of the mixing. This gives the algorithm the freedom to select the most important filter coefficients as it begins building the lattice filter. In contrast, the window method designs its compaction filter with a fixed constraint on the number of coefficients it can use.

Extendibility to MIMO Subband Coding

The capacity and reliability of a radio communications link can be improved by employing multiple antennas at both the transmitter and receiver, i.e. by using MIMO systems. MIMO communications systems are being used more and more to overcoming bandwidth limitations. Transmission of data from a sensor array over a communications network places significant demands on the available bandwidth. However, this could be greatly reduced by using *MIMO subband coding* to eliminate the high degree of spatial and temporal redundancy associated with sensor array signals. Notionally, an *M-channel* subband coder would be comprised of an array of M-band subband coders interconnected in some manner.

It is envisaged that the SBR2 coder can be extended naturally to the case of MIMO subband coding. There are several possible ideas for an SBR2-based M-channel subband coder, which are left for future exploration. On the other hand, we cannot immediately see how a multichannel window method would be used to design an M-channel subband coder.

6.2.5 Performance Analysis

In this section, we present simulation results that quantify the compaction gain and coding gain performances of the filter banks constructed using the SBR2 coder. Its performances are compared to those of the KLT and a prior-art algorithm called the window method (described in section 3.4.2). We also show frequency responses of the various filter banks.

The data simulated for the following experiments are based on some examples given in [32,48]. The algorithms were tested on three types of input signal with an ARMA(N_a) (or Markov-N_a) process. This type of process is regarded as a good model for many practical signals such as image and speech signals. In all cases, the input signal $x(t)$ was generated as follows: An i.i.d. sequence with unit variance and zero mean of length T was realised, which was, as will be indicated, either a BPSK or QPSK signal (specified in sections 4.5 and 5.7). The number of samples per subband channel was maintained at $T_S = T/M = 1024$. The signal $x(t)$ was then obtained from the output of an ARMA (Yule-Walker IIR) filter with order N_a when driven by the i.i.d. sequence. For all simulations $T >> N_a$. The three types of signal used were:

1. ARMA(5) process with poles $p = \left[0.1195, 0.8990e^{\pm j2.1472}, 0.8824e^{\pm j0.5594}\right]$ and zeros $z = \left[\pm 0.9992, -0.45416, 10020e^{\pm j1.3305}\right]$. This has a multiband PSD which allows for clear illustration of the capturing of the signal energy at different frequencies, as suggested in [32]. The ARMA(5) filter was used to filter a BPSK sequence.

2. ARMA(4) process with poles and zeros $p = 0.8456\left[e^{\pm j2.7328}, e^{\pm j0.4088}\right]$ and $z = \left[1.0053e^{\pm j1.5741}, 0.1575e^{\pm j1.9193}\right]$, respectively. This has a difficult multiband PSD and requires $M > 2$ for good coding. The ARMA(4) process was driven by a complex-valued i.i.d. process; viz., a QPSK sequence.

3. ARMA(2) process with poles $p = 0.6041e^{\pm j0.3864}$ and zeros $z = 1.0003e^{\pm j2.0434}$. This models certain types of image texture [48]. A BPSK sequence was used as input to the ARMA(2) filter.

As before, performance measures were obtained using the true statistics of the data, that is, true compaction gains and true coding gains are reported here. The sample measures were also computed for the above experiments but are not shown since they do not represent an accurate measure of the general performance of the algorithm. This is because the algorithms can adapt to (exploit) the noise and signal energies of a particular data set, allowing them to optimise their filters on the sample statistics (fit to the data) rather than the true statistics. Unless

stated otherwise, experiments quantifying compaction gain and coding gain performances were repeated over 50 realisations and the mean of the measures were taken.

The design of a two-channel orthonormal filter bank using the window method for a given PSD is described in section 3.4.2. The SBR2 coder, employing either the WCD-SBR2(N_1) or the WCD-SBR2(G_c) algorithms, was applied to the vector signal $\mathbf{x}(t) = [x_1(t), x_2(t), \ldots, x_M(t)]^T$: inputs to the analysis bank (of a subband coder) shown in Fig. 3.1. The algorithm then produces a paraunitary matrix $\mathbf{E}(z)$, with polyphase filters $E_{km}(z)$. For reasons stated in section 5.4, the SBR2(G_c) coder was applied with a triangular window function, whereas the SBR2(N_1) coder was applied with the (default) rectangular window. In the case of a triangular window, it was found that a window length of $T_w = 20N_a$ produced the best results. For the rectangular window, $T_w = 2N_a + 1$ gave the best performance.

SBR2 Coder and the Window Method for an ARMA(5) Process

In the first set of experiments, filter banks were designed mainly using the SBR2 coder and the window method with varying parameters for an ARMA(5) process with a PSD as in Fig. 6.7.

Example 1. The SBR2(G_c) coder was used to design a two-channel filter bank. It was allowed to run for 200 iterations and produced a filter bank with a maximum order of $N = 52$. The magnitude-square frequency response of the two filters designed by the algorithm is shown in Fig. 6.7(a): the frequency response of the compaction filter, $H_1(z)$, is shown as the solid (blue) curve and its orthogonal complement, $H_2(z)$, is represented by the dotted (red) curve. It can be seen that the algorithm has designed a multiband compaction filter with passbands that coincide with the dominant signal energies. Such a response is commensurate with high compaction gains. In Fig. 6.7(b), we show the PSD of the subband channels $\mathbf{C}_{kk}(e^{j\omega})$ (also included is the PSD of the demultiplexed signals $\mathbf{A}_{kk}(e^{j\omega})$). It is obvious from this graph that the SBR2 coder has performed spectral majorisation. The window method for $N = 200$ produces similar results to those in Fig. 6.7 (not shown).

Example 2: Dependence on L. Figs. 6.8(a) and (b) give a comparison of compaction gain and coding gain performances, respectively, between the two-channel filter bank designed using the window method and that produced by the SBR2(G_c) coder. The abscissa on this figure represents both the number of SBR2 iterations L and the order (N) of the filters produced using the window method. The red dotted (horizontal) line represents the ideal compaction gain and the ideal coding gain for the respective graphs. As expected, the maximum compaction gain and coding gain attained by the algorithms are below their respective ideal values. An important result is that a greater degree of data compression can be achieved using the SBR2 coder than

using the SBR2 algorithm alone (i.e. without a priori knowledge); this can be ascertained by comparing the coding gain performance in Fig. 6.8(b) to that in Fig. 6.4, which both relate to the same ARMA(5) input process as in Fig. 6.7(a). Moreover, for the given input process, the filter banks constructed by the SBR2 coder generally attain a higher coding gain than those of the window method. In particular, for L or $N < \sim 100$, SBR2 converges to a near optimal solution by ~50 iterations. This is because SBR2 has the freedom to choose the principal filter coefficients first. On the other hand, the window method has a fixed order filter, which it must parameterise.

An obvious characteristic of the compaction gain and coding gain curves of Fig. 6.8 is that the suboptimality of the filter banks designed by the SBR2 coder diminishes as L (N) increases. This is also true in the case of the window method for $N <\sim 200$. The slight fall in performance of the window method at high orders is because there are only a fixed and finite number of samples available, and so at large filter orders its estimate of the true PSD suffers from sample noise. Recall that this phenomenon was also observed for PEVD algorithms and related to spectral estimators in sections 4.4 and 5.

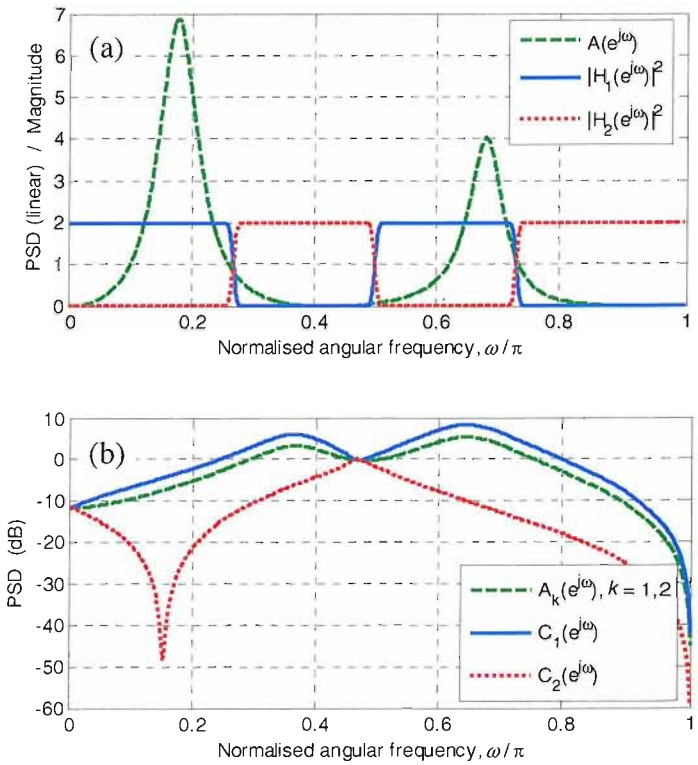


Figure 6.7: Analysis of the two-channel filter bank designed by the SBR2 coder run with 200 iterations (or the window method with order 200 filters). (a) PSD of the input signal (ARMA(5) process) and the magnitude-square of the two filters. (b) Majorised spectra of the subband channels and input signal PSD.

Example 3: Performance for small L. A comparison of the compaction gain and coding gain achieved by the SBR2 coder for $L=11$ (or $N=19$) and the window method for $N=11$ is given in Table 6.1. For this dataset, the SBR2 coder achieves a compaction gain and coding gain that are, respectively, 0.59dB and 1.46dB higher than those obtained using the window method. Figs. 6.9(a) and (b) show the frequency response of the filters produced by the window method and the SBR2 coder for $L=N=11$, respectively. It is clear that the filters constructed by the SBR2 coder have better passband and stopband characteristics than those of the window method. The table also includes the performance of the KLT for comparison. The KLT obtains relatively poor results because the required degrees-of-freedom are not available for the given problem.

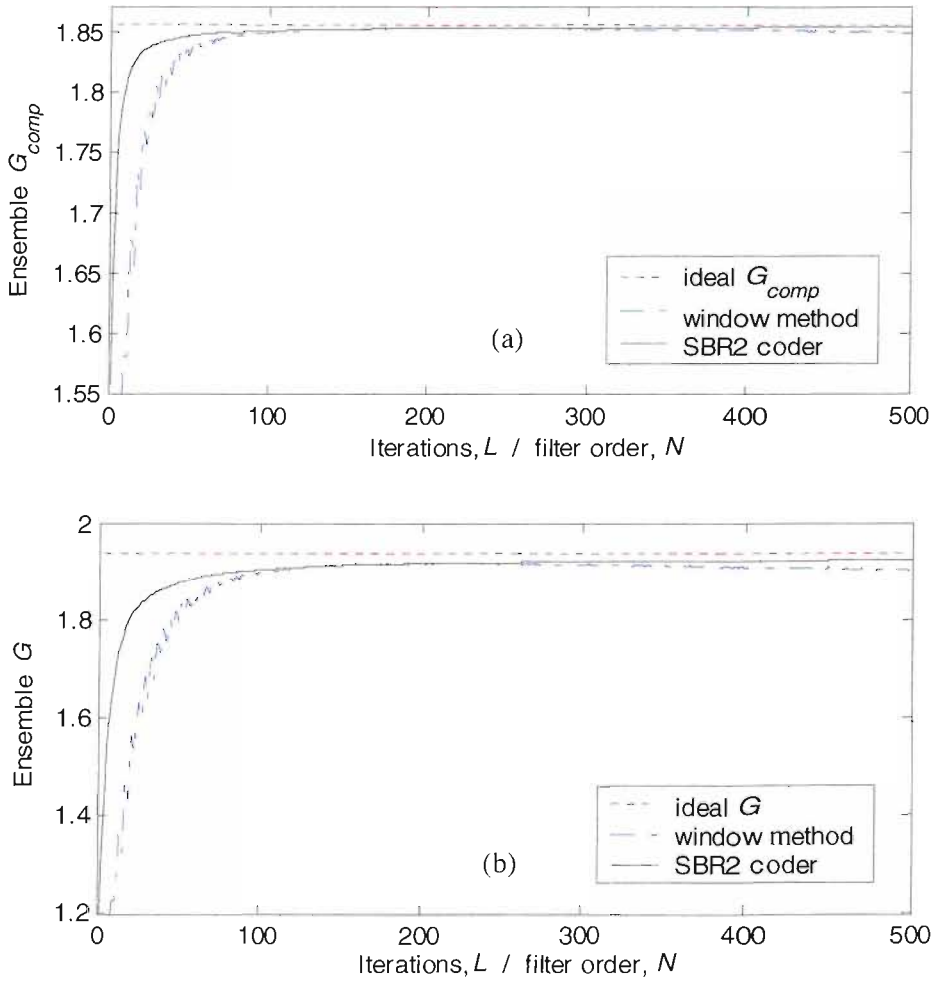


Figure 6.8: Comparison of the window method and SBR2(G_c) coder for the two-channel case. (a) Compaction gain as a function of L (or N) and (b) coding gain versus L (or N) for the ARMA(5) process with PSD as in Fig. 6.7.

Algorithm / measure	G_{comp}	G
Ideal values	1.86	1.94
KLT	0.79	1.07
Window method ($N = 11$)	1.58	1.23
SBR2 coder ($L = 11$)	1.81	1.72

Table 6.1: Comparison of the compaction and coding gain performances of two-channel filter banks designed using the KLT, window method and SBR2(G_c) coder for the PSD in Fig. 6.7(a).

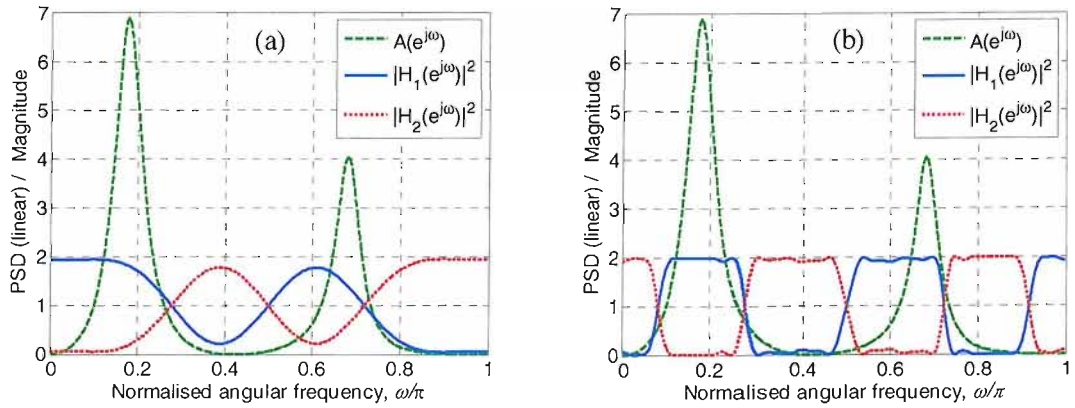


Figure 6.9: Comparison of the frequency responses of a two-channel filter bank designed by (a) the window method with $N = 11$ and (b) the SBR2 coder with $L = 11$ for an ARMA(5) process. PSD of the process is also shown.

Example 4: Dependence on T_S. The dependence of the coding gain performance on the input data length (or the number of low rate samples T_S) of the window method is revealed in Fig. 6.10(a). It is quite clear that the coding gain performance of the window method relies heavily on the number of data samples available, especially for large filter orders. This is in contrast to the coding gain curves produced when using the SBR2 coder for different T_S, as shown in Fig. 6.10(b).

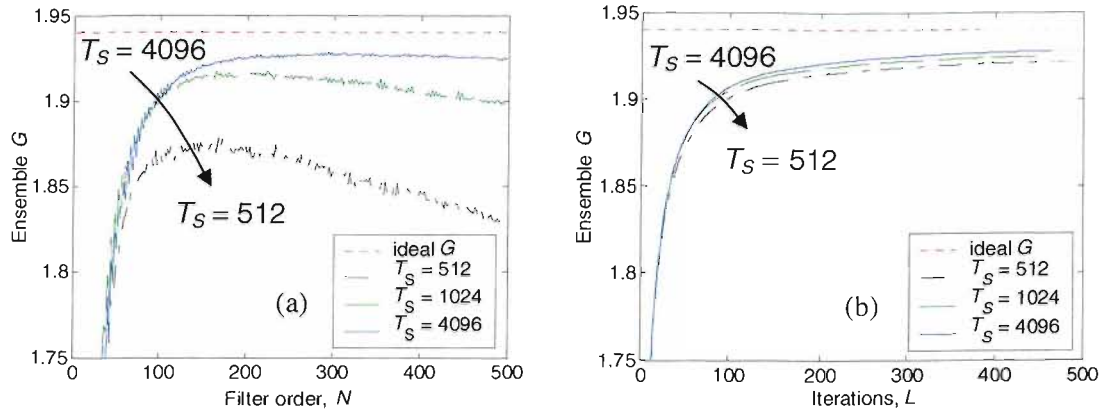


Figure 6.10: (a) Coding gain as a function of N for different data lengths T_S. (b) Coding gain as a function of L for three different data lengths T_S. These are for the two-channel case and for the coloured input signal with PSD as in Fig. 6.7(a).

Comparison of SBR2 Coder Variants for an ARMA(5) Process

Example 5: Effect of different cost functions and windows. The compression performances of the filter banks designed by the SBR2(N_1) coder and SBR2(G_c) coder were compared. In Figs. 6.11(a) and (b), the compaction gains and coding gains versus the number of iterations for four-channel filter banks are shown, respectively. The prominent result in these simulations is that the coding gains achieved by the filter banks constructed using the SBR2(G_c) coder are higher than those of the SBR2(N_1) coder for most L. In addition, we observe that the SBR2(G_c) coder designs a first filter with better energy compaction properties than that of the SBR2(N_1) coder for most L.

Note that the two algorithms would perform exactly the same in the two-channel case were it not for the difference in the window functions used. In fact, for $M > 2$, the success of SBR2(G_c) coder over the other variant is mainly due to the use of a triangular window function. A possible reason for this is that a triangular window is more suited to the input process. This hypothesis was tested by applying the SBR2(N_1) coder with a triangular window function to the same problem example (results not shown). Very similar findings to those obtained from using the SBR2(G_c) coder hold for this case. Specifically, for small L (up to ~300 steps), the coding gains achieved by the SBR2(G_c) coder were slightly higher than those attained by the SBR2(N_1) coder with a triangular window. Interestingly, the contrary is true for the compaction gain performance for the two algorithms; for large L, performances are almost identical.

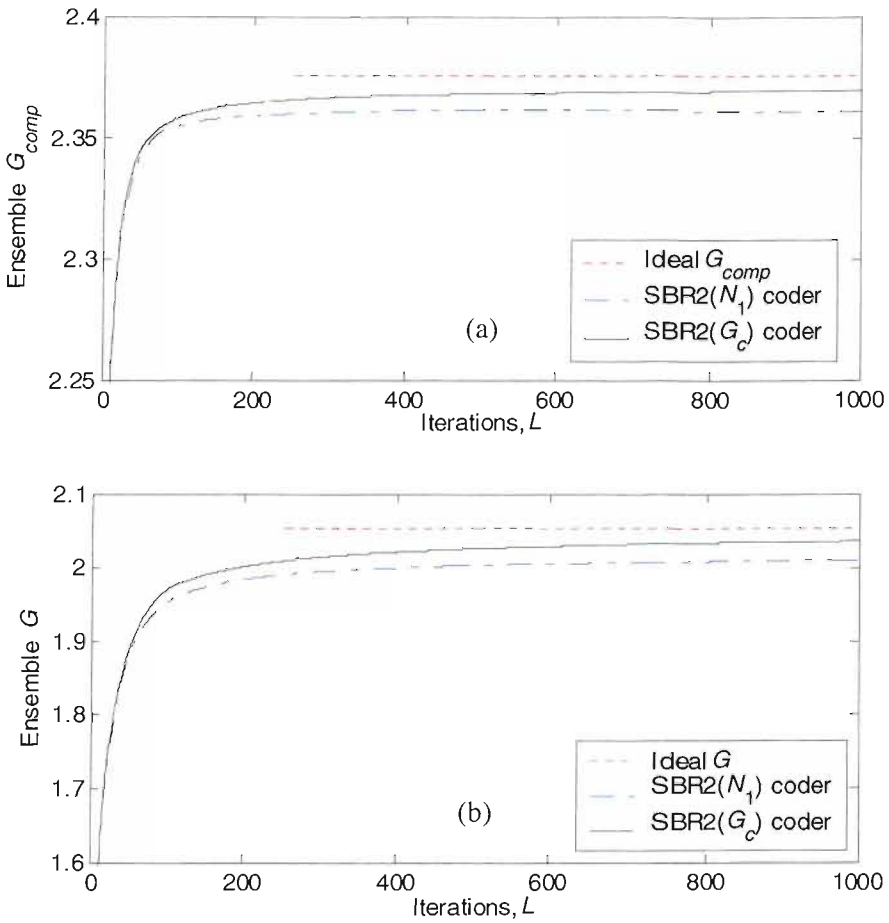


Figure 6.11: Comparison of four-channel filter banks designed by the SBR2(G_c) coder and SBR2(N_1) coder for the input process in Fig. 6.7(a). (a) Compaction gain versus L and (b) coding gain versus L.

Example 6: Dependence on M. Similar performance differences to those in Fig. 6.11 have been observed between the SBR2(N_1) and SBR2(G_c) coders for varying M, with $L=600$; see Fig. 6.12. Note that each point on the graph represents an average over 25 trials. A striking result is that, even though the coding gains (in Fig. 6.12(b)) attained by the SBR2(G_c) coder are greater than those of the SBR2(N_1) coder for nearly all M, the compaction gains (in Fig. 6.12(a)) achieved by the latter are actually higher in most cases for large M. This, in a sense, suggests that energy compaction based optimisation (see chapter 3) may generally give a different solution to coding gain optimisation, which is an observation also made by the authors of [48]. From this we make the conjecture that a multichannel window method based on the construction in [48] would not in general guarantee high coding gains.

A distinct feature in the coding gain results of Fig. 6.12(b) for the $L = 600$ case is that the performances diverge from the ideal value as M increases. A reason for this is as follows: The algorithm has a fixed number of degrees-of-freedom (iterations) to use in order to perform strong decorrelation. As the number of channels increases so does the complexity of the problem; there are more and more channel-pairs to go through and decorrelate in the Jacobi search (see section 5.2 for a description of this). This was tested by setting $L = 1200$, i.e. a two fold increase in the number of iterations, the results from which are shown in Fig. 6.12 and seem to corroborate the hypothesis. This could also be an explanation for the following: the superiority of the SBR2(G_c) coder over the N_1 based algorithm increases with M at first and then looks to plateau. A noticeable characteristic of the compaction gain curves is the ripple-like structure. This is a phenomenon that cannot be explained at present.

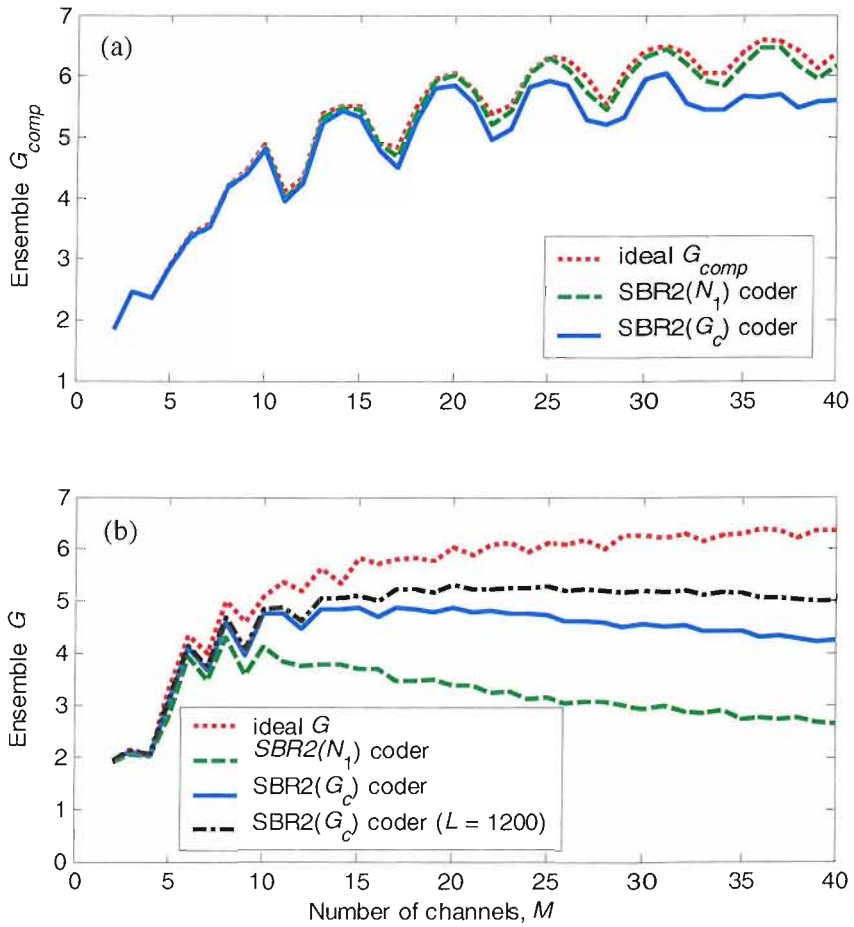


Figure 6.12: Comparison of the SBR2(G_c) coder and SBR2(N_1) coder for the coloured input signal with PSD in Fig. 6.7(a). (a) Compaction gain as a function of M, for $L = 600$. (b) Coding gain as a function of M, for $L = 600$ and $L = 1200$.

Comparison of SBR2 Coder Variants for an ARMA(4) Process

The second set of experiments is concerned with the application of the SBR2 coder with varying parameters to a complex-valued coloured signal with a PSD as in Fig. 6.13.

Example 7: $M = 3$. We have used the SBR2(G_c) coder to design a three-channel filter bank for the ARMA(4) process. The frequency response of the filters generated for $L = 100$ in Fig. 6.13(b) exhibit good energy compaction performance: the algorithm has designed a multiband compaction filter, $H_1(z)$, with passbands that coincide with dominant signal frequency components. Notice that the passbands of the other two decimation filters $H_2(z)$ and $H_3(z)$ fall mostly in low-energy regions. This is commensurate with a high coding gain since the distribution of the subband channel powers is highly non-uniform. By contrast, the two-channel filter bank designed by the SBR2 coder (or the window method) has poor compaction gain and coding gain performances for this type of signal, as shown in Fig. 6.13(a).

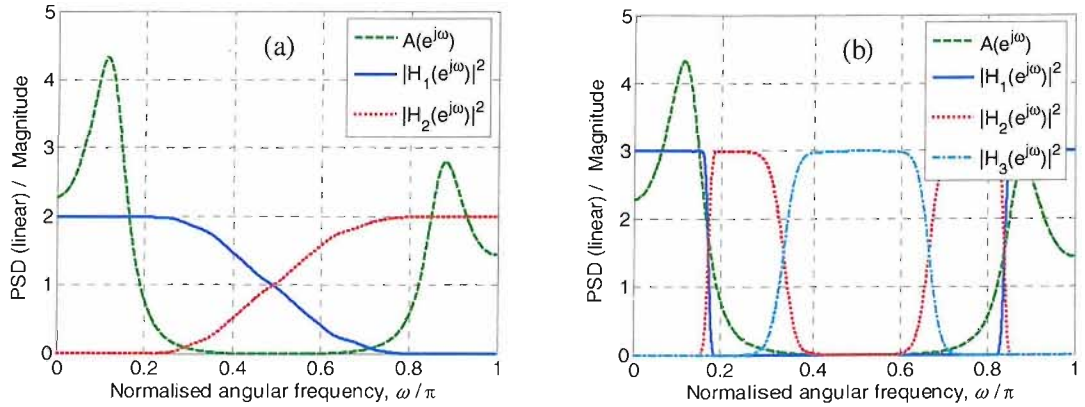


Figure 6.13: Frequency responses of a (a) two-channel filter bank and (b) three-channel filter bank designed by the SBR2(G_c) coder when run with L = 100. These are for an ARMA(4) process with PSD as represented by the dashed (green) curve.

Example 8: $M = 8$. The compression performances of eight-channel filter banks designed by the SBR2(N_1) coder and SBR2(G_c) coder have been compared for the ARMA(4) input process. In Fig. 6.14(a) and (b), the compaction gains and coding gains versus the number iterations are shown, respectively. Again we see that whilst the energy compaction properties of the algorithms are similar for $L > \sim 200$, the coding gains attained by SBR2(G_c) coder are actually much greater than those of the SBR2(N_1) coder for all L. The frequency responses of the filters produced by the SBR2(G_c) coder is shown in Figs. 6.15(a). We have also observed that the set of subband spectra have the spectral majorisation property (approximately); see Fig. 6.15(b).

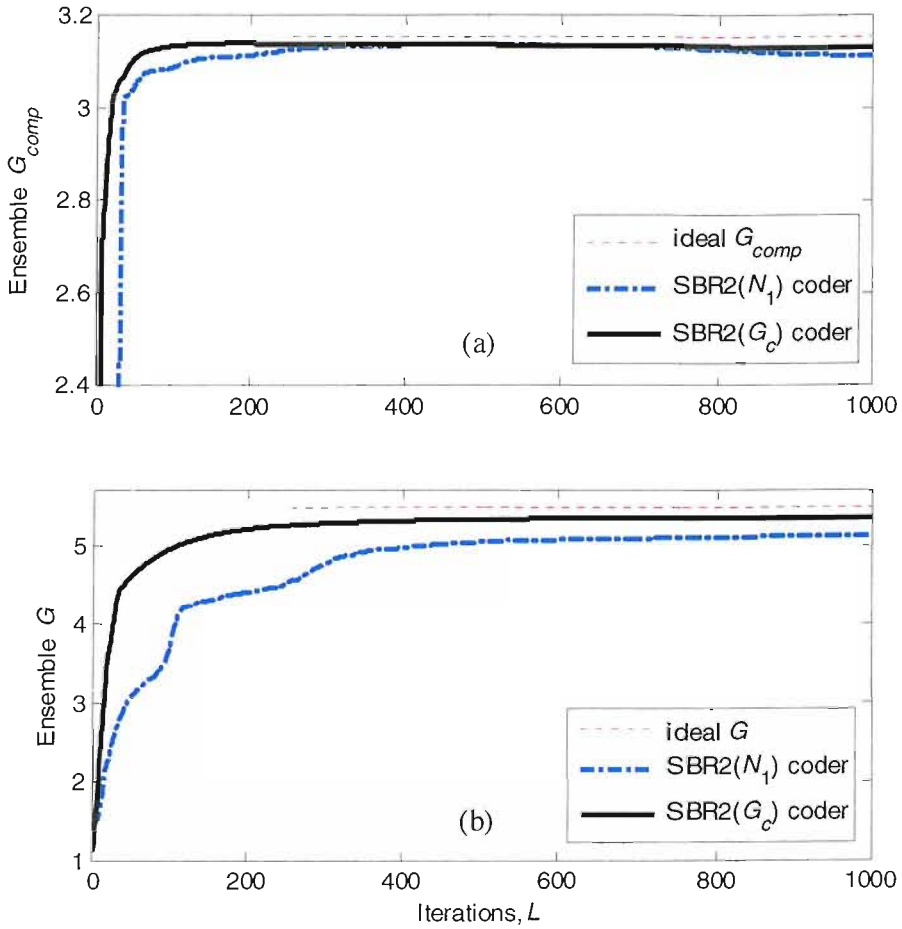


Figure 6.14: Comparison of the SBR2(G_c) and SBR2(N_1) coders for the PSD in Fig. 6.13. (a) Compaction gain versus L for $M = 8$. (b) Coding gain versus L for $M = 8$.

SBR2 Coder and the Window Method for an ARMA(2) Process

In the third and final set of experiments, we considered an ARMA(2) process and compared the compaction gain and coding gain performances of two-channel filter banks; see Table 6.2. The coding gain achieved by the filter bank produced by the SBR2 coder for $L=4$ ($N_s=2$) was 3.74dB higher than that obtained by using the window method for $N=4$; this is despite the fact that the first filter designed by the two algorithms attain virtually identical compaction gains.

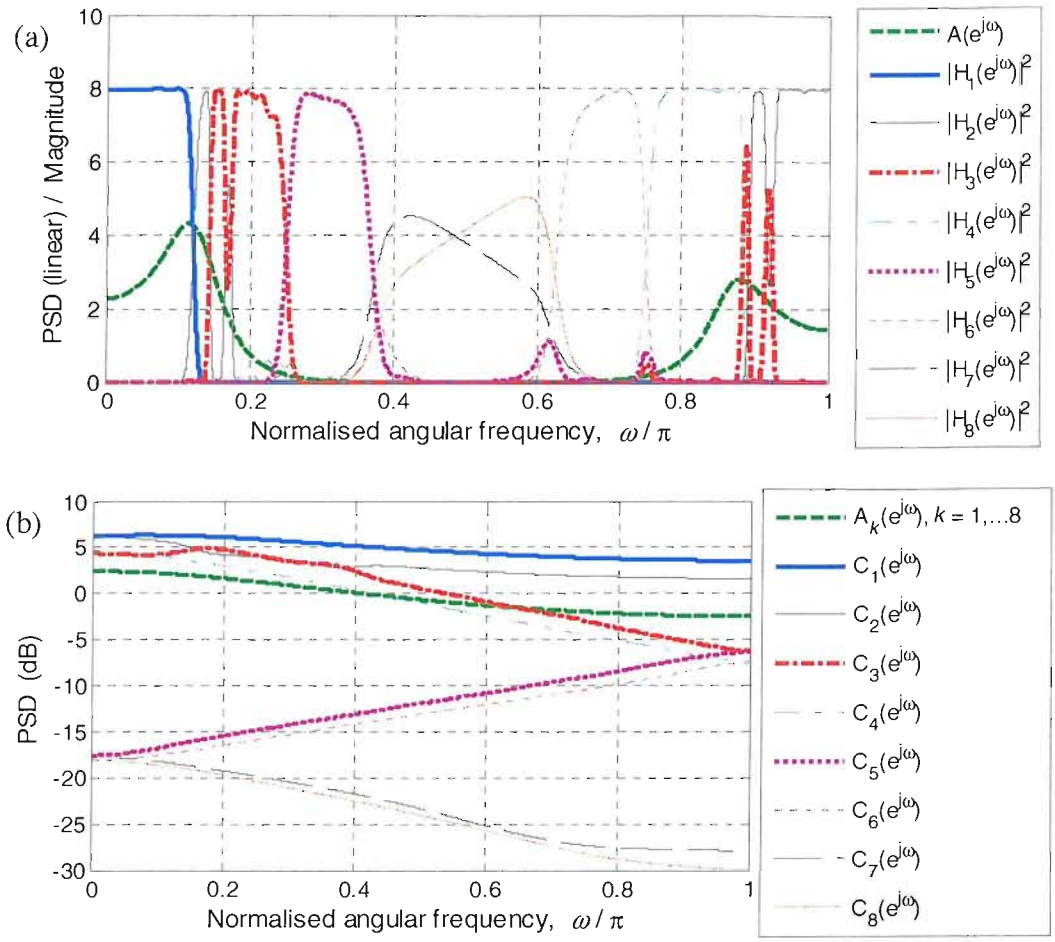


Figure 6.15: (a) PSD of an ARMA(4) process and the magnitude-square of the filters produced by the SBR2(G_c) coder run with L = 400. (b) Approximately majorised spectra of the subband signals along with the input signal PSD.

Algorithm / measure	G_{comp}	G
Ideal values	2.00	10.2
KLT	1.33	2.39
Window method ($N = 4$)	1.92	2.61
SBR2(G_c) coder ($L = 4$)	1.92	6.17

Table 6.2: Compaction and coding gain performances of two-channel filter banks designed using the KLT, window method and SBR2(G_c) coder for an ARMA(2) process.

6.3 Concluding Remarks

In this chapter, we have investigated the applicability of the SBR2 algorithm to multichannel data compression and, more extensively, optimal subband coding. The main result of this chapter is that the SBR2 algorithm can be applied as a very effective tool for optimal orthonormal subband coding. We have shown that the statistics of the demultiplexed signals have a special form for a wide-sense stationary input signal, which is implicitly exploited by prior-art design methods. This motivated an investigation into finding a means of adapting the algorithm for the purpose of subband coding. A data pre-processing stage to SBR2 has been designed which takes advantage of this special structure. The resultant algorithm, called the SBR2 coder, can produce 'near-optimal' M-channel paraunitary (perfect reconstruction) filter banks in a small number of iterations; the suboptimality of the algorithm diminishes as the number of steps increases. This algorithm is applicable to most types of input signal.

The SBR2 coder has been shown to outperform the window method for the two-channel case and for a set of typically encountered signals. Part of this success is due to the fact that SBR2 has the flexibility to select the most appropriate filter coefficients in a small number of steps. This difference in performance increases dramatically as the number of data samples is reduced. The window method cannot be used on its own to produce a subband coder for the case where there are more than two subband channels. A technique for optimally constructing an M-channel filter bank when given the first filter has been proposed in [48]. However, we have observed that having a better energy compaction filter does not in general guarantee better coding gains, as observed in [48].

In audio coding applications, filter banks are commonly used, usually requiring high orders. As discussed in chapters 1 and 3, there are many other applications where filter banks are being used. We believe the SBR2 coder will be valuable for such applications. The SBR2 coder may be extended naturally for MIMO subband coding. This algorithm would improve the utilisation of available bandwidth.

7 Conclusions

7.1 Review

To date, very little attention seems to have been devoted to polynomial matrix techniques equivalent to the eigenvalue decomposition (EVD) or singular value decomposition (SVD) for conventional matrices. In this thesis, a novel algorithm has been developed for the construction of paraunitary matrices for polynomial matrix diagonalisation. It is a natural generalisation of the EVD/SVD to polynomial matrices, and so could have as wide a range of applications for polynomial matrices as the EVD/SVD has for scalar matrices. The algorithm has been successfully applied to the problems of broadband signal subspace estimation and, in particular, optimal orthonormal subband coder design.

7.1.1 Polynomial Matrix EVD

The lack of numerical algorithms for estimating the signal subspace of broadband signals has been the motivation for the development of techniques for extending the EVD to polynomial matrices. We have developed a time-domain method that extends the classical Jacobi algorithm, which can compute the EVD, to the space-time domain, called the second order sequential best rotation (SBR2) algorithm. This approach is a good balance between simplicity and computational speed. The salient features of the algorithm are that it:

- Designs a paraunitary (energy preserving) filter bank;
- Is numerically stable with proven convergence;
- Imposes strong decorrelation and spectral majorisation, to a good approximation;
- Compacts most of the total signal energy into the first channel, i.e., it performs energy compaction.

The algorithm has been shown to provide a good estimate of the signal and noise subspaces of convolutively mixed signals, which is useful in applications such as broadband noise reduction and data compression. The solution provided by the SBR2 algorithm for multichannel arbitrary data is analogous to an estimate of that provided by the PCFB for subband signals. A key assertion made in this thesis is that the problems of broadband subspace decomposition and subband coder design are related through the need for a polynomial matrix EVD (PEVD).

A frequency-domain (counterpart) technique has also been developed, known as the bandwise algorithm. Variations on this algorithm have been proposed that vastly improve its performance. However, we have shown that frequency-domain algorithms have fundamental restrictions on performance, which are the same restrictions that typically limit the accuracy of spectral estimators. The SBR2 and bandwise algorithms have been shown to be superior to a stat-of-the-art algorithm for broadband signal subspace estimation proposed by Lambert in [38].

An objective function has been proposed for use with the SBR2 algorithm that significantly improves the diagonalisation and data compression performances of the algorithm. A covariance-domain version of both the bandwise and SBR2 algorithms have been developed which apply a window to the sample covariances of the input data. The windowing alleviates the degenerateness of the sample covariance and so increases the strong decorrelation performance of the algorithms.

It is expected that the applicability of the SBR2 algorithm to practical problems requiring off-line processing has been increased through improvements made in this thesis. Computationally, the cost of running SBR2 has been lowered by typically an order of magnitude. Although it is not quite fast enough for real time applications, there is much scope for further enhancement.

7.1.2 SBR2 Applied to Subband Coding

The principal component filter bank (PCFB) is optimal, in the mean-square error sense, for both subband coding and energy compaction but its existence cannot be guaranteed for the practical case of FIR filters. However, suboptimal techniques do exist in the literature. One such approach, called the window method in [32], designs a FIR compaction filter, which can be used to construct a two-channel orthonormal filter bank.

The relationship between suboptimal subband coder design techniques, such as the window method, and the SBR2 algorithm has been investigated. This revealed that the suboptimal methods implicitly exploit the special form of the covariance matrix for the demultiplexed signals that exists for a wide-sense stationary input signal. An adaptation of the SBR2 algorithm has been proposed that takes advantage of this fact to design multi-band orthonormal subband coders. The resultant algorithm, called the SBR2 coder, can converge to a solution that yields a perfect reconstruction filter bank which is approximately optimal for subband coding in a small number of iterations; the suboptimality of the algorithm diminishes as the number of steps increases.

The SBR2 coder has been shown to outperform the window method for the two-channel case and for a set of benchmark problems. This success is due to the flexibility of SBR2 to select the most important filter coefficients first. By contrast, the window method works by parameterising a filter of fixed order. We have conjectured that the computation speed of a potential multichannel window method may, in general, be greater than that of the SBR2 coder. However, this is justifiable on the basis that an enhancement in coding efficiency is gained by using the SBR2 coder.

7.2 Future Work

A number of ideas to further enhance and apply the SBR2 algorithm are identified in the following.

7.2.1 Extensions

The SBR2 algorithm is based on one of several possible approaches to diagonalising a polynomial matrix. In principle, the basic concept of an elementary paraunitary matrix, as introduced in this thesis, could be used to generalise other, more sophisticated EVD or SVD algorithms for application to polynomial matrices. They could also be used to generate polynomial matrix versions of entirely different numerical procedures, such as the QR algorithm. The generalisation of the QR algorithm to polynomial matrices could lead to a more efficient decomposition and faster processing.

The two implementations of the SBR2 algorithm described in this thesis, the classical and cyclic Jacobi algorithms, can be viewed as operating on the signals in a pairwise fashion. An alternative strategy is one that applies a sequence of delays such that the instantaneous cross-correlations between most channels are maximised, then an SVD may be applied to the delayed signals. One avenue of future work is to explore computationally efficient implementations of this strategy.

7.2.2 Applications

In this thesis, we have demonstrated that the SBR2 algorithm can be used to design filter banks for efficient subband coding. However, its range of applicability is not confined to just this problem. Some possible routes of exploitation are given in the following.

In section 4.2.2, we showed how the polynomial matrix SVD (PSVD) of a general polynomial matrix may be derived by carrying out the PEVD of the corresponding

parahermitian matrix to obtain the left hand polynomial singular vectors. In this way, the SBR2 algorithm can be used to compute the PSVD of the convolutive mixing process inherent in a multi-input multi-output (MIMO) communication channel. This has been used successfully to design orthogonal space-time channels for optimal data transmission over MIMO channels, i.e., broadband eigen-beamforming (precoding) for MIMO channels. This allows the transmitted beam pattern and the receiver directional sensitivity to be steered.

A related application of the SBR2 algorithm to that described above is its use for the design of oversampled filter banks for coding of channels corrupted by correlated noise [41,85]. This was achieved by identification of and transmission over the low-noise broadband subspace through exploiting the redundancy offered by oversampled filter banks, thus permitting the detection and correction of channel errors.

The SBR2 algorithm is a highly generic numerical technique for multichannel data. It is envisaged that it can be extended naturally for application to the problem of multichannel subband coding, that is, to jointly encode multiple input signal. In the context of digital communications using sensor arrays and NEC (network enabled capability), multichannel subband coding could be invaluable and should be investigated.

There are a number of possible applications which arise from the idea of using the SBR2 algorithm to estimate signal and interference subspaces based on a strong disparity between the signal and interference power spectra. In [55], the SBR2 algorithm was used as a power-based blind signal separation (BSS) algorithm for robust broadband adaptive beamforming. A priori knowledge, which was in the form of an estimated steering vector and the difference in the powers of the signals, was exploited to achieve signal separation. In [39], the SBR2 algorithm was extended using quaternion arithmetic for the purpose of performing second order BSS of polarised signals from a 3-axis seismic sensor array.

In the context of broadband adaptive beamforming, power-based BSS can be applied to the problems of acoustic interference suppression in multistatic active sonar. This would involve the estimation of the angle-of-arrival and range of a weak desired signal and the suppression of the effects of strong acoustic interference signals and multipath (reverberation). The same philosophy can be adopted for the purpose of mitigating radar clutter for phased array radar. The SBR2 algorithm may offer two advantages over conventional space-time adaptive processing (STAP) techniques: it is inherently robust to array calibration errors, since it does not rely on prior information about the array geometry; and it does not require training data to successfully detect difficult targets.

The SBR2 algorithm is particularly effective at producing a compact representation of some aspect of the environment, be it reverberation, e.g. underwater acoustics; radar clutter; or communication channels. It might therefore be a useful tool for detecting changes in an environment. This, in turn, could have potential application to intruder detection, for example harbour security.

Subband coding has been considered as the main deployment area for PEVD algorithms in this thesis. However, the above applications, which are partly hypothetical and partly already implemented with considerable impact, provide a glimpse of the capabilities of the introduced polynomial matrix decompositions and may likely be as wide-ranging for broadband scenarios as the SVD and EVD algorithms are for narrowband problems.

8 Appendices

8.1 Concise Statement of the SBR2 Algorithm

Begin

Step 1: Input $M \times M$ parahermitian matrix $\mathbf{R}(z)$ as defined in (5.1).

Step 2: Find (j,k,t) with $j \neq k$ and $t \geq 0$ such that $|r_{jk}(t)| \geq |r_{lm}(\tau)|$ for any coefficient $r_{lm}(\tau)$ in $\mathbf{R}(z)$ with $l \neq m$ and $\tau \geq 0$.

Step 3: Generate the transformed polynomial matrix $\mathbf{R}'(z) = \mathbf{\Lambda}^{(k,t)}(z)\mathbf{R}(z)\tilde{\mathbf{\Lambda}}^{(k,t)}(z)$ using the transformation defined in (5.6). This leads to a polynomial matrix with entries of the form

$$r'_{lm}(z) = r_{lm}(z) \qquad \forall\, l,m \in \{1,2\cdots M\}\setminus\{k\};\ r'_{lk}(z) = z^{t} r_{lk}(z) \qquad \forall\, l \in \{1,2\cdots M\}\setminus\{k\}$$

$$r'_{km}(z) = z^{-t} r_{km}(z) \qquad \forall\, m \in \{1,2\cdots M\}\setminus\{k\};\ \text{and}\ r'_{kk}(z) = r_{kk}(z) \qquad \forall\, k \in \{1,2\cdots M\}.$$

Step 4: Define $\phi = \arg(r'_{jk}(0))$, $c = \cos\theta$ and $s = \sin\theta$ where θ is the smaller of the two angles given by $\cot 2\theta = \dfrac{r'_{jj}(0) - r'_{kk}(0)}{2|r'_{jk}(0)|}$. Generate the transformed polynomial matrix $\mathbf{R}''(z) = \mathbf{Q}^{(j,k)}(\theta,\phi)\mathbf{R}'(z)\mathbf{Q}^{(j,k)\mathrm{H}}(\theta,\phi)$ using the rotation matrix defined in (5.8). This leads to a polynomial matrix with elements given by

$$r''_{lm}(z) = r'_{lm}(z) \quad \forall\, l,m \in \{1,2\cdots M\}\setminus\{j,k\};$$

$$r''_{jm}(z) = cr'_{jm}(z) + se^{i\phi} r'_{km}(z) \quad \forall\, m \in \{1,2\cdots M\}\setminus\{j,k\};$$

$$r''_{km}(z) = -se^{-i\phi} r'_{jm}(z) + cr'_{km}(z) \quad \forall\, m \in \{1,2\cdots M\}\setminus\{j,k\};$$

$$r''_{lj}(z) = cr'_{lj}(z) + se^{-i\phi} r'_{lk}(z) \quad \forall\, l \in \{1,2\cdots M\}\setminus\{j,k\};$$

$$r''_{lk}(z) = -se^{i\phi} r'_{lj}(z) + cr'_{lk}(z) \quad \forall\, l \in \{1,2\cdots M\}\setminus\{j,k\};$$

$$r''_{jj}(z) = c^2 r'_{jj}(z) + se^{i\phi} cr'_{kj}(z) + se^{-i\phi} cr'_{jk}(z) + s^2 r'_{kk}(z);$$

$$r''_{kk}(z) = s^2 r'_{jj}(z) - se^{i\phi} cr'_{kj}(z) - se^{-i\phi} cr'_{jk}(z) + c^2 r'_{kk}(z);$$

$$r''_{jk}(z) = -se^{i\phi} cr'_{jj}(z) - s^2 e^{i2\phi} r'_{kj}(z) + c^2 r'_{jk}(z) + se^{i\phi} cr'_{kk}(z);\ \text{and}$$

$$r''_{kj}(z) = -se^{-i\phi}cr'_{jj}(z) + c^2 r'_{kj}(z) - s^2 e^{-i2\phi} r'_{jk}(z) + se^{-i\phi}cr'_{kk}(z).$$

Step 5: Set $\mathbf{R}(z) = \mathbf{R}''(z)$ to define the input for the next iteration.

Repeat steps 2 to 5 until the maximum value for the modulus of any off-diagonal coefficient of the polynomial matrix is sufficiently small.

End.

8.2 Necessary and Sufficient Conditions for Optimality

In subband coding theory it is well known that strong decorrelation and spectral majorisation of the subband signals are necessary and sufficient for optimality (the maximisation of the coding gain) [74]. By simple generalisations of proofs given in [74], we aim to show that strong decorrelation and spectral majorisation of WSS broadband signals are necessary and sufficient conditions for optimality in the sense of maximising the G_m.

Theorem 8.1. *Strong decorrelation is necessary.*

Proof. It is sufficient to show that compression can be improved in the two-channel case because this transformation could then be applied to any two channels that are not strongly decorrelated. Suppose the pair of channels, say $X_1(z)$ and $X_2(z)$ are not orthogonalised. Then $E[\sum_t y_1(t) y_2^*(t+\tau)] \neq 0$ for some τ. By applying a time delay, $z^{-\tau}$, to one of the channels we can assume $\tau = 0$. It will be shown how the compression gain can be increased without violating the paraunitary condition. Suppose a unitary matrix $\mathbf{\Theta}$ is used to transform the pair $[X_1(z), X_2(z)]$ into a decorrelated pair $[Y_1(z), Y_2(z)]$. This can be done by choosing $\mathbf{\Theta}$ to be a matrix composed of the orthogonal eigenvectors of the $\tau = 0$ element of the space-time covariance matrix $\mathbf{A}(\tau)$. It can be concluded that the compression gain has increased if it is shown that the product of the channel variances has been reduced, i.e. $a_{110}a_{220} > c_{110}c_{220}$. Let $\mathbf{C}(z)$ be the covariance matrix of the vector $\underline{Y}(z) = [Y_1(z), Y_2(z)]^T$, then

$$\mathbf{C}(z) = \mathbf{\Theta}\mathbf{A}(z)\tilde{\mathbf{\Theta}}. \tag{8.1}$$

Notice that the diagonal elements of $\mathbf{A}(z)$ and $\mathbf{C}(z)$ are the quantities a_{kk0} and c_{kk0}. Now, note:

$$c_{110}c_{220} = \det[\mathbf{C}(z)] = \det[\mathbf{A}(z)] = a_{110}a_{220} - |a_{120}|^2 < a_{110}a_{220}. \tag{8.2}$$

where det[·] denotes the determinant of a matrix. The first equality follows because $\mathbf{C}(z)$ is diagonal since the signal pair $[Y_1(z), Y_2(z)]$ are decorrelated. The second equality follows from the unitarity of $\boldsymbol{\Theta}$. Therefore, for this transformation, the compression gain

$$G_m = \frac{1}{2}\log\left(\frac{a_{110}a_{220}}{c_{110}c_{220}}\right) > 0. \tag{8.3}$$

Q.E.D.

Strong decorrelation, while necessary, is not sufficient for the maximisation of G_m. A filter bank is optimal if the product of the output channel variances $\xi = \prod_{k=1}^{M} C_{kk}(e^{j\omega})$ is a minimum.

Theorem 8.2. *To minimise ξ, it is necessary that the eigenvalues of the parahermitian matrix be ordered in the same fashion for all frequencies ω, i.e., the set of output signals has the spectral majorisation property.*

Proof. Assuming that the majorisation property is not satisfied, it will be shown how the data can be compressed without violating the paraunitary condition. Again, since a transformation could be applied to any channel-pair that is not spectrally majorised, it is sufficient to show that compression can be improved in the two-channel case. Suppose $A_{11}(e^{j\omega}) \geq A_{22}(e^{j\omega})$ is not valid for all ω, even though $a_{110} \geq a_{220}$. That is, for some values of ω, $A_{22}(e^{j\omega}) \geq A_{11}(e^{j\omega})$. Suppose a *permutation* matrix $\mathbf{T}(e^{j\omega})$ is applied to the channels. The matrix $\mathbf{T}(e^{j\omega})$ is chosen as

$$\mathbf{T}(e^{j\omega}) = \begin{cases} \mathbf{I}_2, & \text{if } A_{11}(e^{j\omega}) \geq A_{22}(e^{j\omega}) \\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, & \text{if } A_{22}(e^{j\omega}) \geq A_{11}(e^{j\omega}). \end{cases} \tag{8.4}$$

The new pair of power spectra $C_{11}(e^{j\omega})$ and $C_{22}(e^{j\omega})$ will then satisfy the property $C_{11}(e^{j\omega}) \geq C_{22}(e^{j\omega})$ for all ω. Moreover, for each ω $C_{11}(e^{j\omega}) \geq A_{11}(e^{j\omega})$, whereas $A_{22}(e^{j\omega}) \geq C_{22}(e^{j\omega})$. Thus, the variances of the new signals $Y_1(z)$ and $Y_2(z)$ are such that $c_{110} \geq a_{110}$ and $c_{220} \leq a_{220}$. Since $\mathbf{T}(e^{j\omega})$ is paraunitary by construction, the filter bank remains paraunitary, and the sum of the variances is preserved, i.e. $c_{110} + c_{220} = a_{110} + a_{220}$. Thus

$$c_{110} = a_{110} + \delta,\ c_{220} = a_{220} - \delta, \text{and } \delta > 0 \tag{8.5}$$

so that $c_{110}c_{220} = a_{110}a_{220} - \delta^2 - \delta(a_{110} - a_{220})$. Since $\delta > 0$ and $a_{110} \geq a_{220}$, the preceding equation implies $c_{110}c_{220} < a_{110}a_{220}$; therefore, the compression gain is positive. Q.E.D.

Though strong decorrelation and spectral majorisation are necessary for optimality (maximisation of the compression gain), neither is individually sufficient.

Theorem 8.3. *The compression gain is maximised iff the output data channels simultaneously satisfy strong decorrelation and spectral majorisation.*

Proof. In view of the theorems set out in the first two paragraphs of this section, it only remains to prove that strong decorrelation and spectral majorisation together imply optimality. If a unitary matrix $\mathbf{H}(e^{j\omega})$ performs strong decorrelation then

$$\mathbf{C}(e^{j\omega}) = \mathbf{H}(e^{j\omega})\mathbf{A}(e^{j\omega})\tilde{\mathbf{H}}(e^{j\omega}) = \text{diag}\left[C_{11}(e^{j\omega}), C_{22}(e^{j\omega}), \ldots, C_{MM}(e^{j\omega})\right] , \tag{8.6}$$

where $\mathbf{A}(e^{j\omega})$ and $\mathbf{C}(e^{j\omega})$ are the input and output sample data covariance matrices, respectively, and $C_{kk}(e^{j\omega})$ is the k^{th} output channel power spectrum (or eigenvalue of $\mathbf{C}(e^{j\omega})$) for some frequency ω. Suppose the majorisation property holds. This means that for every ω the eigenvalues are ordered as in (4.14). Since the set of eigenvalues is unique, each diagonal element in (8.6) is uniquely determined for each ω. Since majorisation and strong decorrelation are necessary for optimality and since there is only one set of majorised decorrelated channel power spectra, it follows that majorisation together with decorrelation leads to optimality.

Q.E.D.

9 References

1. Akansu, A.N., Duhamel, P., Lin, X. and De Courville, M., "Orthogonal transmultiplexers in communication: A review," *IEEE Transactions on Signal Processing*, Vol.46(No.4):pp.979–995, April 1998.

2. Akansu, A.N. and Liu, Y., "On signal decomposition techniques," *Optical Engineering Journal*, volume 30, pages 912–920, July 1991.

3. Akkarakaran, S. and Vaidyanathan, P.P., "On optimization of filter banks with denoising applications," In *Proceedings of IEEE International Symposium on Circuits and Systems*, Orlando, FL, volume 3, pages 512–515, June 1999.

4. Akkarakaran, S. and Vaidyanathan, P.P., "Filter bank optimization with convex objectives, and the optimality of principal component forms," *IEEE Transactions on Signal Processing*, Vol.49(No.1):pp.100–114, January 2001.

5. Baxter, P.D. and McWhirter, J.G., "Blind signal separation of convolutive mixtures," In *Proceedings of Asilomar Conference on Circuits, Systems and Computers*, volume 1, pages 124–128, Pacific Grove, CA, November 2003.

6. Bellanger, M.G., Bonnerot, G. and Coudreuse, M., "Digital filtering by polyphase network: Application to sample-rate alteration and filter banks," *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol.24(No.2):pp.109–114, April 1976.

7. Blackman, R.B. and Tukey, J.W., *The measurement of power spectra from the point of view of communications engineering*, Dover, New York, 1959.

8. Bolcskei, H. and Hlawatsch, F., "Noise reduction in oversampled filter banks using predictive quantisation," *IEEE Transactions on Information Theory*, Vol.47(No.1):pp.155–172, January 2001.

9. Buckley, K.M. and Griffiths, L.J., "Broad-band signal subspace spatial-spectrum (BASS-ALE) estimation," *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol.36(No.7):pp.953–964, July 1988.

10. Comon, P., "Independent component analysis, a new concept," *IEEE Transactions on Signal Processing*, Vol.36(No.3):pp.287–314, 1994.

11. Compton, R.T., "The relationship between tapped-delay line and FFT processing in adaptive arrays," *IEEE Transactions on Antennas and Propagation*, Vol.36(No.1):15–26, January 1988.

12. Crochiere, R.E., Webber, S.M. and Flanagan, J.K.L., "Digital coding of speech in subbands," *Bell Systems Technical Journal*, Vol.55(No.8):1069–1085, October 1976.

13. Dane, G. and Nguyen, T.Q., "Signal-adapted binary tree-structured paraunitary filter bank design," *IEEE Transactions on Antennas and Propagation*, Vol.36(No.1):15–26, January 2002.

14. Dapena, A. and Serviere, C., "A simplified frequency-domain approach for blind separation of convolutive mixtures," *International Workshop on Independent Component Analysis and Signal Separation*, San Diego, CA, pages 569–574, December 2001.

15. Delgosha, F. and Faramarz, F., "Results on the factorization of multidimensional matrices for paraunitary filterbanks over the complex field," *IEEE Transactions on Signal Processing*, Vol.52(No.5):pp.1289–1303, May 2004.

16. Desarte, P., Macq, B. and Slock, D.T.M., "Signal-adapted multiresolution transform for image coding," *IEEE Transactions on Information Theory*, Vol.38(No.2):pp.897–904, March 1992.

17. Douglas, S.C., Amari, S. and Kung, S.-Y., "Adaptive paraunitary filter banks for spatio-temporal principal and minor subspace analysis," In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Phoenix, AZ, volume 2, pages 1089–1092, March 1999.

18. Dorf, R.C. and Bishop, R.H., *Modern Control Systems*, Addison-Wesley, New York, 7th Edition, 1995.

19. Gao, X., Nguyen, T.Q. and Strang G., "On factorization of M-channel paraunitary filterbanks," *IEEE Transactions on Signal Processing*, Vol.49(No.5):pp.1433–1446, July 2001.

20. Golub, G.H. and Van Loan, C.F, *Matrix Computations*, Johns Hopkins University Press, Baltimore, 3rd Edition, 1989.

21. Haykin, S., *Adaptive Filter Theory*, Prentice-Hall, Englewood Cliffs, NJ, 2nd Edition, 1991.

22. Haykin, S., *Adaptive Filter Theory*, Prentice-Hall, Upper Saddle River, NJ, 4th Edition, 2002.

23. Haykin, S., Litva, J. and Shepherd, T.J., *Radar Array Processing*, Springer-Verlag, 1992.

24. Huang, J.J.Y. and Schultheiss, P.M., "Block quantization of correlated Gaussian random variables," *IEEE Transactions on Communications Systems*, Vol.11(No.3):pp.289–296, September 1963.

25. Huang, D-Y., Regalia, P.A. and Bellanger, M., "Comparison of two eigenstructure algorithms for lossless multirate filter optimisation," In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Munich, Germany, volume 3, pages 2437–2440, April 1997.

26. Huelsman, L.P., *Active and Passive Analogue Filter Design: An introduction*, McGraw-Hill Inc., Singapore, 1993.

27. Hung, H., and Kaveh, "Focusing matrices for coherent signal-subspace processing," *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol.36(No.8):pp.1272–1282, August 1988.

28. Ifeachor, E.C. and Jervis, B.W., *Digital Signal Processing: A Practical Approach*, Addison-Wesley, New York, 1993.

29. Jahromi, O.S., Masnadi-Shirazi, M.A. and Fu, M., "A fast O(N) algorithm for signal-adaptive filter bank design," In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Seattle, WA, volume 3, page 1325–1328, May 1998.

30. Jezek, J., Hromcık, M., and Sebek, M., "New algorithm for spectral factorisation and its practical application," In *Proceedings of European Control Conference*, Porto, Portugal, pages 3104–3109, September 2001.

31. Kailath, T., *Linear Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1980.

32. Kirac, A. and Vaidyanathan, P.P., "Theory and design of optimum FIR compaction filters," *IEEE Transactions on Signal Processing*, Vol.46(No.4):pp.903–919, April 1998.

33. Kirac, A. and Vaidyanathan, P.P., "On existence of FIR principal component filter banks," In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Seattle, WA, volume 3, pages 1329–1332, May 1998.

34. Kirac, A. and Vaidyanathan, P.P., "Optimality of orthonormal transforms for subband coding," In *Proceedings of 8th IEEE DSP Workshop*, Utah, August 1998.

35. Kirac, A. and Vaidyanathan, P.P., "Optimal nonuniform orthonormal filter banks for subband coding and signal representation," In *Proceedings of IEEE International Conference on Image Processing*, Chicago, IL, October 1998.

36. Klemm, R., "Space-time adaptive processing principles and applications," *IEE Radar, Sonar, Navigation and Avionics Series 9*, 1998.

37. Lambert, R.H., Joho, M. and Mathis, H., "Polynomial singular values for number of wideband source estimation and principal component analysis," In *Proceedings of International Conference on Independent Component Analysis*, 2001.

38. Lambert, R.H., *Multichannel Blind Deconvolution: FIR Matrix Algebra and Separation of Multipath Mixtures*, PhD Thesis, University of Southern California, Electrical Engineering Department, 1996.

39. Le Bihan, N., "Diagonalisation de matrices polynomiales quaternioniques: application a la seperation de melanges convolutifs d'ondes polarisees," In *Proceedings of GRETSI Conference*, Louvaine-la-Neuve, Belgium, September 2005.

40. Lee, T-W, Bell, A., Lambert, R.H., "Blind separation of delayed and convolved sources," In *Advances in Neural Information Processing Systems 9*, Massachusetts Institute of Technology Press, Cambridge, MA, pages 758–764, 1997.

41. Liu, C., Weiss, S., Redif, S., Cooper, T., Lampe, L. and McWhirter, J.G., "Channel coding for power line communication based on oversampled filter banks," In *Proceedings of IEEE International Symposium on Power-Line Communications and its Applications*, Vancouver, Canada, pages 246–249, April 2005.

42. Malvar, H.S. and Staelin, D.H., "The LOT: Transform coding without blocking effects," *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol.37(No.4):pp.2703–2714, April 1989.

43. Malvar, H.S, "Extended lapped transforms: properties, applications, and fast algorithms," *IEEE Transactions on Signal Processing*, Vol.40(No.11):pp.553–559, November 1992.

44. McWhirter, J.G. and Baxter, P.D., "A novel technique for broadband SVD," In *Proceedings of 12th Annual Workshop on Adaptive Sensor Array Processing*, Massachusetts Institute of Technology Lincoln Laboratory, Cambridge, MA, March 2003.

45. McWhirter, J.G., Baxter, P.D., Cooper, T. and Redif, S., "A novel technique for broadband subspace decomposition," In *Proceedings of EURASIP European Signal Processing Conference (EUSIPCO)*, Florence, Italy, September 2006.

46. McWhirter, J.G., Baxter, P.D., Cooper, T. and Redif, S., "An algorithm for polynomial matrix EVD with application to signal processing," accepted for publication in *IEEE Transactions on Signal Processing*, to appear 2007.

47. Medles, A. and Slock, D.T.M., "Linear precoding for spatial multiplexing MIMO systems: blind channel estimation aspects," In *Proceedings of IEEE International Conference on Communications*, pages 401–404, April 2002.

48. Moulin, P. and Mihcak, M.K., "Theory and design of signal-adapted FIR paraunitary filter banks," *IEEE Transactions on Signal Processing*, Vol.46(No.4):pp.920–929, April 1998.

49. Moulines, E., Duhamel, P., Cardoso, J. and Mayrargue, S., "Subspace methods for the blind identification of multichannel FIR filters," *IEEE Transactions on Signal Processing*, Vol.43(No.2):pp.516–525, February 1995.

50. Mulgrew, B., Grant, P. and Thompson, J., *Digital Signal Processing: Concepts and Applications*, Palgrave, Basingstoke, 1999.

51. Proakis, J.G., *Digital Communications*, McGraw-Hill, New York, 2nd Edition, 1989.

52. Redif, S., *On Broadband Subspace Decomposition Techniques*, First Year PhD Report, University of Southampton, Department of Electronics and Computer Science, September 2003.

53. Redif, S. and Cooper, T., "Paraunitary filter bank design via a polynomial singular value decomposition," In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Philadelphia, PA, volume 4, pages 613–616, March 2005.

54. Redif, S. and Cooper, T., "Orthonormal filter bank design using a polynomial matrix decomposition," *6th IMA International Conference on Mathematics in Signal Processing*, The Royal Agricultural College, Cirencester, December 2004.

55. Redif, S., McWhirter, J.G., Baxter, P.D. and Cooper, T., "Robust broadband adaptive beamforming via polynomial eigenvalues," In *Proceedings of MTS/IEEE Oceans Conference*, Boston, MA, September 2006.

56. Regalia, P.A. and Huang, D.Y., "Attainable error bounds in multirate adaptive lossless FIR filters," In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Detroit, MI, volume 5, pages 1460–1463, May 1995.

57. Regalia, P.A. and Loubaton, P., "Rational subspace estimation using adaptive lossless filters," *IEEE Transactions on Signal Processing*, Vol.40(No.10):pp.2393–2405, October 1992.

58. Rota, L., Comon P. and Icart S., "Blind MIMO paraunitary equalizer," In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Seattle, WA, volume 4, pages 285–288, April 2003.

59. Schmidt, R.O., "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, Vol.34(No.3):pp.276–280, March 1986.

60. Sathe, V.P. and Vaidyanathan, P.P., "Effects of multirate systems on the statistical properties of random signals," *IEEE Transactions on Signal Processing*, Vol.41(No.1):pp.131–146, January 1993.

61. Segall, A., "Bit allocation and encoding for vectors," *IEEE Transactions on Information Theory*, Vol.22(No.2):pp.162–169, March 1976.

62. Shannon, C.E., "A mathematical theory of communications", *Bell Systems Technical Journal*, Vol.27(No.3):pp.379–423,623–656, July 1948.

63. Shapiro, J.M., "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, Vol.41(No.12):pp.3445–3462, December 1993.

64. Stetter, H.J., *Numerical Polynomial Algebra*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2004.

65. Szego, G., *Orthogonal Polynomials*, American Mathematical Society Colloquium Publications, volume 23, New York, 1939.

66. Tarokh, V., Seshadri, N. and Calderbank, A.R., "Space-time codes for high data rate wireless communication: Performance criterion and code construction," *IEEE Transactions on Information Theory*, Vol.44(No.2):pp.744–765, March 1998.

67. The MathWorks, Inc., *MATLAB 6.5.1 User's Guide*, The MathWorks, August 2003.

68. Tsatsanis, M.K. and Giannakis, G.B., "Principal component filter banks for optimal multiresolution analysis," *IEEE Transactions on Signal Processing*, Vol.43(No.8):pp.1766–1777, August 1995.

69. Tuqan, J. and Vaidyanathan, P.P., "A state space approach for the design of optimum FIR energy compaction filters," *IEEE Transactions on Signal Processing*, Vol.48(No.10):pp.2822–2838, October 1998.

70. Unser, M., "On the optimality of ideal filters for pyramid and wavelet signal approximation," *IEEE Transactions on Signal Processing*, Vol.41(No.12):pp.3591–3596, December 1993.

71. Unser, M. and Thierry, B., "Mathematical properties of the JPEG2000 wavelet filters," *IEEE Transactions on Image Processing*, Vol.12(No.9):pp.1080–1090, September 2003.

72. Vaccaro, R., ed., *SVD and Signals Processing, II: Algorithms, Analysis and Applications*, Elsevier, 1991.

73. Vaidyanathan, P.P., *Multirate Systems and Filter Banks*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

74. Vaidyanathan, P.P., "Theory of optimal orthonormal filter banks," *IEEE Transactions on Signal Processing*, Vol.46(No.6):pp.1528–1543, June 1998.

75. Vaidyanathan, P.P., "Filter banks in digital communications," In *IEEE Circuits and Systems Magazine*, Vol.1(No.2):pp.4–25, 2nd Quarter 2001.

76. Vaidyanathan, P.P. and Akkarakaran, S., "A review of the theory and application of optimal subband and transform coders," In *Proceedings of Applied Computational Harmonic Analysis*, Vol.10(No.3):pp.254-289, May 2001.

77. Vaidyanathan, P.P., Nguyen T.Q., Doganata Z. and Saramaki T., "Improved techniques for design of perfect reconstruction FIR QMF banks with lossless polyphase matrices," *Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 337–340, November 2002.

78. Vandendorpe, L., Cuvelier, L., Deryck, F., Louveaux, J. and Van De Wiel, O., "Fractionally spaced linear and decision feedback detectors for transmultiplexers," *IEEE Transactions Signal Processing*, Vol.46(No.4):pp.996–1011, April 1998.

79. Van Veen B.D. and Buckley K.M., "Beamforming: A versatile approach to spatial filtering," In *IEEE Acoustics, Speech and Signal Processing Magazine*, Vol.5(No.2):pp.4–24, April 1988.

80. Vaseghi, S.V., *Advanced Digital Signal Processing and Noise Cancellation*, John Wiley & Sons, Chichester, 2nd Edition, 2001.

81. Wang, H. and Kaveh, M. "Coherent signal subspace processing for detection and estimation of angle of arrival of multiple wideband sources," *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol.33(No.4):pp.823–831, August 1985.

82. Weiss, S., Dooley, S.R., Stewart, R.W. and Nandi, A.K., "Adaptive equalization in oversampled subbands," *IEE Electronics Letters*, Vol.34(No.15):pp.1452–1453, July 1998.

83. Weiss, S., *On Adaptive Filtering in Oversampled Subbands*, PhD Thesis, University of Strathclyde, Department of Electronic and Electrical Engineering, 1998.

84. Weiss, S., Stewart, R.W., Schabert, M., Proudler I.K. and Hoffman M.W., "An efficient scheme for broadband adaptive beamforming," *Asilomar Conference on Signals, Systems and Computers*, Monterey, CA, November 1999.

85. Weiss, S., Redif, S., Cooper, T., Baxter, P.D. and McWhirter, J.G., "Paraunitary oversampled filter bank design for channel coding," *EURASIP Journal on Applied Signal Processing*, vol.2006, Article ID 31346, 10 pages, 2006.

86. Weisstein, E.W., *CRC Concise Encyclopedia of Mathematics*, CRC Press LLC, New York, 1999.

87. Welch, P.D, "The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Transactions on Audio Electroacoustics*, Vol.15(No.2):pp.70–73, June 1967.

88. Xuan, B. and Bamberger, R.H., "FIR principal component filter banks," *IEEE Transactions on Signal Processing*, Vol.46(No.4):pp.930–940, April 1998.

89. Zheng, F.-C., McLaughlin, S., Mulgrew, B., "Blind deconvolution algorithms based on 3rd- and 4th-order cumulants," In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Toronto, Canada, volume 3, pages 1753–1756, April 1991.