

**KATHOLIEKE UNIVERSITEIT LEUVEN**
FACULTEIT TOEGEPASTE WETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK
Kasteelpark Arenberg 10, 3001 Leuven (Heverlee)

# SUBBAND AND FREQUENCY–DOMAIN ADAPTIVE FILTERING TECHNIQUES FOR SPEECH ENHANCEMENT IN HANDS–FREE COMMUNICATION

Promotor:
Prof. dr. ir. M. Moonen

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de toegepaste wetenschappen

door

**Koen ENEMAN**

Maart 2002

© Katholieke Universiteit Leuven – Faculteit Toegepaste Wetenschappen
Arenbergkasteel, B-3001 Heverlee (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotocopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm or any other means without written permission from the publisher.

D/2002/7515/03

ISBN 90-5682-337-X

# Abstract

The telecommunications sector is characterized by an increasing demand for user–friendliness and interactivity. This explains the growing interest in hands–free communication systems. Signal quality in current hands–free systems is unsatisfactory. To overcome this, advanced signal processing techniques such as the subband and frequency–domain adaptive filter are employed to enhance the signal. These techniques are known to have computationally efficient solutions. Furthermore, thanks to the frequency–dependent processing and adaptivity, highly time–varying systems and signals with a continuously changing spectral content such as speech can be handled.

This thesis deals with subband and frequency–domain adaptive filtering techniques for speech enhancement in hands–free communication. The text consists of four parts. In the first part design methods for perfect and nearly perfect reconstruction DFT modulated filter banks are discussed. Part II deals with subband and frequency–domain adaptive filtering. The subband adaptive filter and the PBFDAF–algorithm are discussed. Next, the interrelation between both approaches is studied and a novel subband adaptation scheme is proposed. In part III of the thesis an extension to the PBFDAF algorithm is presented, called the PBFDRAP adaptive filter. The algorithm is analyzed and fast implementation schemes are derived. In the final part we describe applications of our algorithms to the acoustic echo cancellation problem. It is seen that the algorithms discussed in parts I–III can be successfully applied to real–world signal enhancement applications.

# Glossary

## Mathematical Notation

| | |
|---|---|
| $\mathbf{v}$ | vector $\mathbf{v}$ |
| $\mathbf{v}(z)$ | vector $\mathbf{v}$, function of the $z$–transform variable |
| $\mathbf{M}$ | matrix $\mathbf{M}$ |
| $\mathbf{M}(z)$ | matrix $\mathbf{M}$, function of the $z$–transform variable |
| $\underline{\mathbf{v}}$, $\underline{\mathbf{M}}$ | frequency–domain equivalents of $\mathbf{v}$ and $\mathbf{M}$ |
| $\mathbf{M}^T$ | transpose of matrix $\mathbf{M}$ |
| $\mathbf{M}^*$ | complex conjugate of matrix $\mathbf{M}$ |
| $\mathbf{M}^H = (\mathbf{M}^*)^T$ | Hermitian transpose of matrix $\mathbf{M}$ |
| $\mathbf{M}^{-1}$ | inverse of matrix $\mathbf{M}$ |
| $\mathbf{M}^\dagger$ | pseudo–inverse of matrix $\mathbf{M}$ |
| $\det \mathbf{M}$ | determinant of matrix $\mathbf{M}$ |
| $\operatorname{adj} \mathbf{M} = \mathbf{M}^{-1}.\ \det \mathbf{M}$ | adjugate of matrix $\mathbf{M}$ |
| $\operatorname{diag}\{\mathbf{v}\}$ | square diagonal matrix with vector $\mathbf{v}$ as diagonal |
| $\mathbf{M}_*(z)$ | complex conjugation of the coefficients of $\mathbf{M}(z)$ without changing $z$ |
| $\tilde{\mathbf{M}}(z) = \mathbf{M}_*^T(z^{-1})$ | paraconjugate of $\mathbf{M}(z)$ |
| $\mathbf{v}(m)$ | $m$–th element of vector $\mathbf{v}$ |
| $[\mathbf{v}(z)]_m$ | $m$–th element of vector function $\mathbf{v}(z)$ |
| $\mathbf{M}(m,n)$ | element on the $m$–th row and $n$–th column of matrix $\mathbf{M}$ |
| $[\mathbf{M}(z)]_{m,n}$ | element on the $m$–th row and $n$–th column of matrix function $\mathbf{M}(z)$ |
| $\mathbf{A} \otimes \mathbf{B}$ | Kronecker product of matrix $\mathbf{A}$ and $\mathbf{B}$ |
| $h[k]$ | discrete–time filter or time sequence $h$ |
| $H(z)$ | $z$–transform of $h[k]$ |
| $H(f)$ | Discrete Fourier Transform of $h[k]$ |
| $x \star y$ | convolution of $x[k]$ and $y[k]$ |
| $x \circledast y$ | circular convolution of $x[k]$ and $y[k]$ |
| $x \circledast y$ | circular correlation of $x[k]$ and $y[k]$ |
| $H_{l:L}(z)$ | the $l$–th out of $L$ polyphase components of FIR filter $H(z)$, $z$–transform of $h[l + Lk]$ |

| | |
|---|---|
| $h[k]_{N\downarrow}$ | $h[k]$ $N$–fold downsampled |
| $h[k]_{N\uparrow}$ | $h[k]$ $N$–fold upsampled |
| $\mathbb{N}$ | set of natural numbers |
| $\mathbb{N}_0 = \mathbb{N} \setminus \{0\}$ | set of natural numbers larger than 0 |
| $\mathbb{Z}$ | set of integer numbers |
| $\mathbb{Z}_0 = \mathbb{Z} \setminus \{0\}$ | set of integer numbers except 0 |
| $\mathbb{Q}$ | set of rational numbers |
| $\mathbb{R}$ | set of real numbers |
| $\mathbb{R}_0 = \mathbb{R} \setminus \{0\}$ | set of real numbers except 0 |
| $\mathbb{R}^+$ | set of positive real numbers |
| $\mathbb{C}$ | set of complex numbers |
| $\mathbb{R}^M$ | set of real $M$–dimensional vectors |
| $\mathbb{C}^M$ | set of complex $M$–dimensional vectors |
| $\mathbb{C}_{\mathbf{0}}^M = \mathbb{C}^M \setminus \{\mathbf{0}\}$ | set of complex $M$–dimensional vectors except $\mathbf{0}$ |
| $\Re\{x\}$ | real part of $x \in \mathbb{C}$ |
| $\Im\{x\}$ | imaginary part of $x \in \mathbb{C}$ |
| $x^*$ | complex conjugate of $x$ |
| $\mathrm{conj}(\cdot)$ | complex conjugation |
| $\hat{x}$ | estimate of $x$ |
| $\lfloor x \rfloor$ | largest integer smaller or equal to $x \in \mathbb{R}$ |
| $\lceil x \rceil$ | smallest integer larger or equal to $x \in \mathbb{R}$ |
| $\mathrm{rnd}(x)$ | round $x \in \mathbb{R}$ to the nearest integer |
| $\lvert \cdot \rvert$ | absolute value |
| $\lvert\lvert \cdot \rvert\rvert_2$ | 2–norm |
| $\mathcal{E}\{\cdot\}$ | expectation operator |
| $\sigma_x^2$ | variance of $x$ |
| $\mathrm{gcd}(M, N)$ | greatest common divisor of $M$ and $N$ |
| $\mathrm{lcm}(M, N)$ | least common multiple of $M$ and $N$ |
| $x \bmod y$ | remainder after division of $x \in \mathbb{N}$ by $y \in \mathbb{N}$ |
| $p = a : b$ | $p$ is an integer between $a \in \mathbb{Z}$ and $b \in \mathbb{Z}$, i.e. $a \leqslant p \leqslant b, \quad p \in \mathbb{Z}$ |
| $a \ll b$ | $a$ is much smaller than $b$ |
| $a \gg b$ | $a$ is much larger than $b$ |
| $a \approx b$ | $a$ is approximately equal to $b$ |

## Fixed Symbols

| | |
|---|---|
| $M$ | number of subbands, DFT size |
| $N$ | subsampling factor |
| $L$ | block size |
| $P$ | filter partitioning length |
| $K$ | least common multiple |
| $g$ | greatest common divisor |
| $k$ | discrete–time index |

| | |
|---|---|
| $f$ | frequency–domain variable |
| $\omega = 2\pi f$ | pulsation |
| $z$ | $z$–domain variable |
| $n$ | block time index |
| $f_s$ | sampling frequency |
| $w[k]$ | unknown FIR system, acoustic path |
| $\hat{w}[k]$, $\hat{w}^{(n)}[k]$ | (equivalent) fullband adaptive filter, estimate of $w[k]$ |
| $x$ | far–end (loudspeaker) signal |
| $s$ | local signal source–of–interest |
| $d = s + w \star x$ | near–end (microphone) signal |
| $e$ | error signal, output of the adaptive filter |
| $\epsilon_i$ | $i$–th subband error signal |
| $n_{rb}$ | number of real subbands to be processed |
| $n_{cb}$ | number of complex subbands to be processed |
| $\mu$ | adaptive filter stepsize |
| $\mathbf{R}_{\mathbf{xx}} = \mathcal{E}\{\mathbf{x}^*\mathbf{x}^T\}$ | autocorrelation matrix of vector $\mathbf{x}$ |
| $L_{FB}$ | length of the (equivalent) fullband adaptive filter |
| $L_{SB}$ | length of the subband adaptive filters |
| $L_f$ | length of the filter bank prototype |
| $L_f^a$ | length of the analysis filters |
| $L_f^s$ | length of the synthesis filters |
| $L_p$ | length of the synthesis polyphase filters |
| $L_{\text{ef}}$ | effective length of the analysis prototype filter |
| $L_{ac}$ | number of anti–causal filtering taps |
| $L_c$ | number of extra causal filtering taps |
| $\mathbf{0}$ | zero vector or zero matrix |
| $\mathbf{0}_N$ | $N \times N$ zero matrix |
| $\mathbf{0}_{M\times N}$ | $M \times N$ zero matrix |
| $\mathbf{I}_N$ | $N \times N$ identity matrix |
| $\mathbf{J}$ | exchange matrix with ones along the main anti–diagonal and zeros elsewhere |
| $\mathbf{F}$ | DFT matrix, $\mathbf{F}(m,n) = e^{-j\frac{2\pi m n}{M}}, \quad 0 \leqslant m,n < M$ |
| $\mathbf{H}(z)$ | analysis polyphase matrix |
| $\mathbf{G}(z)$ | synthesis polyphase matrix |
| $\mathbf{B}(z)$ | prototype polyphase matrix of a DFT modulated analysis filter bank |
| $\mathbf{C}(z)$ | prototype polyphase matrix of a DFT modulated synthesis filter bank |
| $h_0[k] \longleftrightarrow H_0(z)$ | analysis prototype filter |
| $g_0[k] \longleftrightarrow G_0(z)$ | synthesis prototype filter |
| $f_m[k] \longleftrightarrow F_m(z)$ | $m$–th subband adaptive filter |
| $j$ | $\sqrt{-1}$ |
| $\delta[k]$ | Dirac impulse at $k = 0$ |

## Acronyms and Abbreviations

| | |
|---|---|
| A/D | Analog–to–Digital converter |
| AEC | Acoustic Echo Cancellation |
| ALU | Arithmetic Logic Unit |
| ANC | Adaptive Noise Cancellation |
| APA | Affine Projection Algorithm |
| ASIC | Application–Specific Integrated Circuit |
| BLMS | Block–LMS adaptive filter |
| CD | Compact Disk |
| cf. | `confer` : compare with |
| CPU | Central Processing Unit |
| D/A | Digital–to–Analog converter |
| DCT | Discrete Cosine Transform |
| DFT | Discrete Fourier Transform |
| DRAM | Dynamic Random Access Memory |
| DSP | Digital Signal Processor |
| e.g. | `exempli gratia` : for example |
| Eq. | equation |
| ERLE | Echo Return Loss Enhancement |
| FDAF | Frequency–Domain Adaptive Filter |
| FFT | Fast Fourier Transform |
| FIR | Finite Impulse Response filter |
| HiFi | High Fidelity |
| IDFT | Inverse Discrete Fourier Transform |
| i.e. | `id est` : that is |
| iff | if and only if |
| IFFT | Inverse Fast Fourier Transform |
| IIR | Infinite Impulse Response filter |
| LMS | Least Mean Square adaptive filter |
| MAC | Multiply–Accumulate operation |
| MFlops | Millions of Floating point Operations Per Second |
| MIMO | Multi–Input Multi–Output system |
| MIPS | Millions of Instructions Per Second |
| NLMS | Normalized Least Mean Square adaptive filter |
| `op.` | number of equivalent real Operations |
| `ops.` | number of equivalent real Operations per Second |
| P/S | Parallel–to–Serial converter |
| PBFDAF | Partitioned Block Frequency–Domain Adaptive Filter |
| PBFDRAP | Partitioned Block Frequency–Domain RAP adaptive filter |
| PC | Personal Computer |
| PRA | Partial Rank Algorithm |

| | |
|---|---|
| QMF | Quadrature Mirror Filters |
| RAP | Row Action Projection |
| RLS | Recursive Least Squares adaptive filter |
| S/P | Serial–to–Parallel converter |
| SNR | Signal–to–Noise Ratio |
| SPL | Sound Pressure Level |
| SRAM | Static Random Access Memory |
| SVD | Singular Value Decomposition |
| VME | VERSA Module Eurocard (IEEE 1014) computer architecture |
| vs. | versus |
| w.r.t. | with respect to |
| @ | at |
| 0ws | zero wait state memory |

# Contents

**Voorwoord** i

**Abstract** iii

**Korte Inhoud** v

**Glossary** vii

**Contents** xiii

**Samenvatting** xxi

**1 Introduction** 1

1.1 Problem statement . . . . . 1

1.2 Hands–free communication . . . . . 3

1.2.1 Definition . . . . . 3

1.2.2 Examples of hands–free communication systems . . . . . 4

1.2.3 Signal deterioration . . . . . 6

1.3 Characteristics of speech and the acoustic environment . . . . . 7

1.3.1 Speech signals . . . . . 7

1.3.2 The acoustic environment . . . . . 8

1.4 Enhancement techniques . . . . . 10
1.4.1 Acoustic echo cancellation . . . . . 10
1.4.2 Noise suppression and interference cancellation . . . . . 13
1.4.3 Dereverberation . . . . . 14
1.5 Outline of the thesis and contributions . . . . . 15
1.5.1 Motivation . . . . . 15
1.5.2 Chapter by chapter overview and contributions . . . . . 15
1.6 Conclusions . . . . . 20

**2 Basic Concepts** **21**

2.1 Signal processing basics . . . . . 22
2.1.1 Representation of variables . . . . . 22
2.1.2 Multirate signal processing . . . . . 22
2.1.3 Some definitions related to matrix algebra . . . . . 23
2.2 Filter bank basics . . . . . 24
2.2.1 General subband scheme . . . . . 24
2.2.2 Modulated filter banks . . . . . 25
2.2.3 Polyphase implementation . . . . . 28
2.2.4 Perfect reconstruction . . . . . 30
2.2.5 Overview of filter bank design techniques . . . . . 30
2.3 Adaptive filtering techniques for speech enhancement . . . . . 33
2.3.1 Standard adaptive filtering techniques . . . . . 35
2.3.2 Block–based techniques . . . . . 39
2.4 Computational cost . . . . . 44
2.5 Conclusions . . . . . 46

# I DFT Modulated Filter Bank Design for Oversampled Subband Systems

**3 Perfect Reconstruction Oversampled DFT Modulated Filter Bank Design 47**

3.1 Oversampled DFT modulated subband systems . . . . . 48

3.1.1 DFT modulated analysis filter bank . . . . . 48

3.1.2 DFT modulated synthesis filter bank . . . . . 51

3.1.3 Implementation issues . . . . . 55

3.2 Perfect reconstruction . . . . . 55

3.2.1 Smith–McMillan decomposition based perfect reconstruction filter bank design . . . . . 57

3.2.2 Para–unitary filter banks . . . . . 60

3.3 Para–unitary filter bank design . . . . . 61

3.3.1 Imposing para–unitarity . . . . . 61

3.3.2 Para–unitary lattices . . . . . 63

3.3.3 Optimization of the para–unitary lattices . . . . . 64

3.3.4 Adjusting the prototype filter length . . . . . 65

3.3.5 Design examples . . . . . 68

3.4 Conclusions . . . . . 71

**4 Nearly Perfect Reconstruction DFT Modulated Filter Bank Design 73**

4.1 Nearly perfect reconstruction DFT modulated filter banks . . . . . 74

4.2 Frequency–domain optimization . . . . . 75

4.3 Mixed time/frequency–domain optimization . . . . . 77

4.4 Application . . . . . 83

4.5 Conclusions . . . . . 86

## II Subband and Frequency–Domain Adaptive Filtering

**5 Subband Adaptive Filtering 89**

5.1 Subband adaptive systems . . . . . 90

5.1.1 General subband adaptive filtering setup . . . . . 90

5.1.2 Subband versus fullband adaptive filtering . . . . . 91

5.1.3 Filter bank selection . . . . . 92

5.1.4 Polyphase implementation . . . . . 93

5.1.5 DFT modulated subband adaptive filters . . . . . 93

5.2 Design criteria for subband adaptive systems . . . . . 94

5.2.1 Frequency selectivity . . . . . 95

5.2.2 Perfect reconstruction . . . . . 95

5.2.3 Perfect path modelling . . . . . 97

5.3 Downsampling and aliasing : two extreme cases . . . . . 98

5.3.1 Critically downsampled subband schemes . . . . . 98

5.3.2 Two–fold oversampled subband systems . . . . . 98

5.4 Subband adaptive filter length . . . . . 99

5.4.1 Infinite–length subband filters . . . . . 99

5.4.2 Introducing anti–causal filter taps . . . . . 104

5.5 Implementation cost and complexity gain with respect to LMS . . . 110

5.5.1 Rough cost estimate . . . . . 110

5.5.2 Detailed cost analysis . . . . . 111

5.5.3 Cost evaluation . . . . . 112

5.6 Conclusions . . . . . 115

**6 Analysis of the Partitioned Block Frequency–Domain Adaptive Filter as a Subband Adaptive System 117**

6.1 Partitioned Block Frequency–Domain Adaptive Filter . . . . . 118

6.1.1 Derivation of the PBFDAF algorithm . . . . . 118
6.1.2 PBFDAF algorithm : equations and properties . . . . . 122
6.1.3 Normalization . . . . . 123
6.1.4 Constrained versus unconstrained updating . . . . . 124
6.1.5 Ambiguity compensation for $M > P + L - 1$ . . . . . 125
6.2 The PBFDAF as a special case of subband adaptive filtering . . . . 127
6.3 PBFDAF : design criteria . . . . . 133
6.4 Implementation cost . . . . . 135
6.4.1 Cost computation . . . . . 135
6.4.2 Cost evaluation and optimal parameter setting . . . . . 136
6.5 Conclusions . . . . . 140

**7 Fullband Error Adaptation Scheme 145**

7.1 Fullband error adaptation . . . . . 146
7.2 Computational complexity . . . . . 150
7.3 PBFDAF weight updating revisited . . . . . 153
7.4 Conclusions . . . . . 155

**III Iterated Partitioned Block Frequency–Domain Adaptive Filtering**

**8 Partitioned Block Frequency–Domain RAP 157**

8.1 Partitioned block frequency–domain RAP . . . . . 158
8.1.1 Definition . . . . . 158
8.1.2 Mechanism . . . . . 159
8.2 On iterating the PBFDRAP . . . . . 160
8.2.1 Computation of $\lim_{R\to\infty} \underline{\mathbf{w}}_p^{(n,R)}$ . . . . . 161
8.2.2 Unconstrained PBFDRAP : $\lim_{R\to\infty} \underline{\mathbf{w}}_p^{(n,R)}$ . . . . . 165

8.2.3 Constrained PBFDRAP : $\lim_{R\to\infty} \underline{\mathbf{w}}_p^{(n,R)}$ . . . 168
8.2.4 Summary . . . 173
8.3 Simulation examples . . . 175
8.4 Conclusions . . . 176

**9 Fast Partitioned Block Frequency–Domain RAP 179**
9.1 Fast PBFDRAP . . . 180
9.1.1 Fast PBFDRAP, version 1 . . . 180
9.1.2 Fast PBFDRAP, version 2 . . . 181
9.1.3 Fast PBFDRAP, version 3 . . . 181
9.1.4 Fast constrained PBFDRAP . . . 182
9.1.5 Summary . . . 183
9.2 Computational cost . . . 188
9.2.1 Unconstrained PBFDRAP . . . 188
9.2.2 Constrained PBFDRAP . . . 188
9.2.3 Unnormalized constrained PBFDRAP versus PRA . . . 191
9.3 Conclusions . . . 194

**IV Acoustic Echo Cancellation, Implementation and Experiments**

**10 Acoustic Echo Cancellation, Implementation & Experiments 195**
10.1 Robust operation and control . . . 196
10.1.1 Short–time energy . . . 197
10.1.2 Far–end activity detection . . . 198
10.1.3 Double–talk detection . . . 199
10.1.4 Post–processing . . . 200
10.2 Design examples and off–line comparison . . . 201

10.3 A real–time implementation of an acoustic echo canceller on DSP . . 204
10.3.1 DSP equipment . . . . . 205
10.3.2 Software . . . . . 206
10.3.3 Experiments . . . . . 207
10.4 Conclusions . . . . . 214

**11 Conclusions and Further Research** **217**
11.1 Conclusions . . . . . 217
11.2 Suggestions for further research . . . . . 220

**Bibliography** **223**

**Appendices** **241**
A Some definitions related to matrix algebra . . . . . 241
B Appendix to part I . . . . . 245
B.1 Proof of theorem 3.1 . . . . . 245
B.2 Properties of $\mathbf{B}(z)$ . . . . . 246
B.3 Proof of theorem 3.2 . . . . . 248
B.4 Proof of theorem 3.3 . . . . . 250
B.5 Proof of theorem 3.4 . . . . . 251
B.6 Inverse decomposition of para–unitary lattices . . . . . 253
B.7 Para–unitary parameterization for $M = 2N$ . . . . . 255
B.8 Para–unitary DFT modulated filter banks revisited . . . . . 259
C Appendix to part II . . . . . 261
C.1 Proof of theorem 5.2 . . . . . 261
C.2 Proof of theorem 5.3 . . . . . 265
C.3 Detailed complexity analysis for the subband adaptive filter . 267
C.4 Proof of theorem 6.1 . . . . . 271

C.5 Proof of theorem 6.2 . . . . . 272
C.6 "Time–reversed" PBFDAF . . . . . 273
C.7 Proof of theorem 6.3 . . . . . 276
C.8 Proof of theorem 6.4 . . . . . 279
C.9 Complexity analysis for the PBFDAF . . . . . 280
C.10 Proof of theorem 7.1 . . . . . 283
D Appendix to part III . . . . . 287
D.1 Proof of theorem 8.1 . . . . . 287
D.2 Proof of theorem 8.2 . . . . . 288
D.3 Proof of theorem 8.3 . . . . . 289
D.4 Proof of theorem 8.4 . . . . . 290
D.5 Proof of theorem 8.5 . . . . . 290
D.6 Constrained PBFDRAP : $L_{FB} < L$ . . . . . 291
D.7 Proof of theorem 8.7 . . . . . 293

# Chapter 1

# Introduction

In the first section of this introductory chapter a motivation is given for the techniques that will be developed in the forthcoming chapters of the thesis and we will present some future perspectives on hands–free communication, which is the application we have in mind.

In section 1.2 a few examples of hands–free communication systems are given and the different types of signal degradation that occur are identified.

It appears that the characteristics of speech and the properties of the acoustic environment impose specific constraints on the type of signal enhancement algorithm that can be used and on the way the algorithms are applied. Hence, in section 1.3 some basics of speech and acoustics are discussed.

For each type of signal degradation that can be identified in the hands–free communication setup, a number of enhancement techniques are known from the literature. In section 1.4 several signal enhancement algorithms are briefly addressed.

An outline and an overview of the different chapters and parts of the thesis will be presented in section 1.5. The main contributions are summarized and references will be given to the publications that were brought about in the frame of this work.

Some conclusions to this chapter are formulated in section 1.6.

## 1.1 Problem statement

The telecommunications market has rapidly expanded in recent years. This has brought about an important industrial activity with a huge economic impact. The

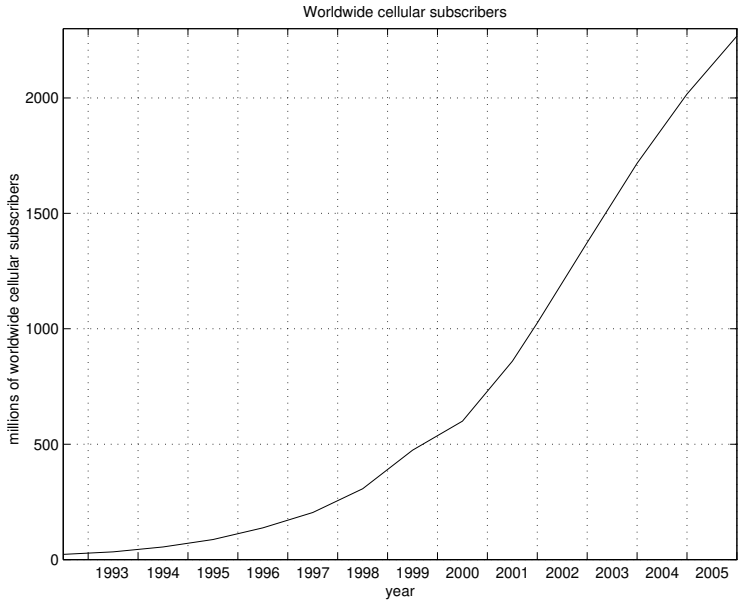


Figure 1.1: Number of worldwide cellular subscribers [39] [179]

annual revenue of the global telecommunications market in 1996 was estimated at US$ 645 billion and is expected to surpass US$ 1 trillion in 2002 [85]. This growth is partly due to the expansion of the mobile phone industry. As indicated in figure 1.1 the estimated number of worldwide cellular subscribers now exceeds one billion and it is expected that this number will continue to increase substantially in the near future.

The telecommunications industry is characterized by an ongoing tendency towards innovation and optimization. This implies, among other things, a focusing towards user–friendliness and interactivity and hence explains the increasing demand for hands–free communication systems today. As it is believed that more and more telecom applications will become hands–free in the near future, a large potential is expected for innovative and product–oriented research in the field of hands–free communication in the coming years. This is confirmed by the observation that the global hands–free market can grow from US$ 3 billion today to over US$ 9 billion in the next five years [151].

In present–day hands–free communication systems the signal quality is often unsatisfactory. Several types of signal deterioration can be distinguished, as will be indicated in section 1.2.3. To each of them a specific set of enhancement techniques can be applied.

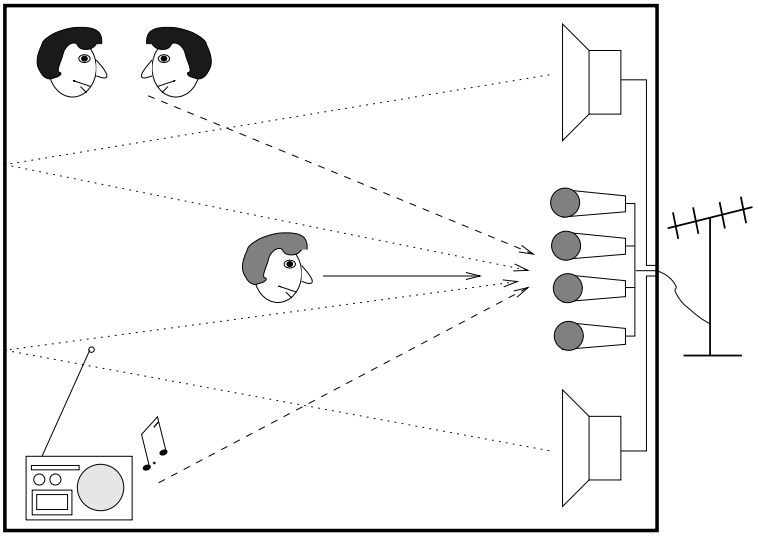

Figure 1.2: Hands–free communication setup

In this thesis subband and frequency–domain adaptive filtering techniques are studied. These signal processing algorithms can be used in a wide variety of applications where signal enhancement is required. In part I, II and III of the thesis several signal processing algorithms will be considered. In part IV it will be shown that these signal processing techniques can be applied to enhance the signal quality in hands–free communication systems. We will concentrate on one form of degradation in particular, which is caused by so–called acoustic echoes, and illustrate how the algorithms discussed in part I, II and III of the thesis can be employed.

## 1.2 Hands–free communication

### 1.2.1 Definition

Consider figure 1.2, which shows a typical hands–free communication setup. The conference room accommodates one or more correspondents, which interact with other people at a remote site via a wireless or wired communication channel. The room shown in figure 1.2 is called the near–end conference room as it accommodates the local or near–end speaker(s). At the remote site there is a similar room, called far–end conference room, with the far–end speaker(s).

As we are dealing with a hands–free communication system the correspondents don't wear a headset or hold a microphone. Contrary to classical communication

systems they are granted the freedom to walk around and to interact with each other in a natural way.

To establish hands–free communication, in each conference room a number of microphones are installed to record the local conversation. The recorded signals are then sent to the remote site where they are fed into a set of loudspeakers.

### 1.2.2 Examples of hands–free communication systems

**Hands–free telephony**

Different sorts of applications fit in the hands–free communication framework. Most important from an economic point of view is certainly hands–free telephony. Recently in many countries all over the world mobile telephony has been forbidden while driving. Mobile phone calls in cars are allowed only if hands–free kits are used. This is motivated by the observation that hand–held mobile phone calls distract the driver and increase the number of accidents. During a mobile phone call the driver misses 4 out of 10 road signs and fails to give way to other vehicles in 25% of the cases. It appears that the accident risk increases with 75%, which reduces to 24% if a hands–free kit is used [171].

It was found that people in North America spend a combined 500 million passenger–hours in their vehicles each week. Although 65 percent of all cell–phone conversations take place in a car or other form of transport, less than 15 percent of the cell–phone users in the US have hands–free accessories [25]. So, a huge market for hands–free kits is expected in the near future.

A little side–remark is however that cell–phone usage is responsible for only 1.5 percent of all accidents in the US. On the other hand outside distraction was responsible for almost 30 percent of all crashes. Adjusting the radio or changing a tape or CD was the second–biggest cause of accidents, amounting to 11 percent. Furthermore, it appears that the conversations themselves lead to a dangerous driving behavior, not the type of phone that is used [25]. It should be added however that in contrast to the US manual gear changes are still very popular in Europe. It is clear that it is almost impossible to change the gear, to use a mobile phone and to steer and drive safely at the same time.

The most common low–cost hands–free kits for mobile telephony in cars, such as the KX–TCA87 of Panasonic (± US$ 25), are headsets with a (directional) microphone and headphone. The quality is satisfactory, but according to our definition of hands–free systems in section 1.2.1 these systems are not true hands–free solutions. A second class of products, such as the hands–free car adapter NTN1583 of Motorola (± US$ 100), use a hands–free microphone and a built–in speaker, which are connected to the dashboard. These are hands–free systems, but the quality is not so good. Complete hands–free car kits on the other hand, such as the Nokia CARK–91US (± US$ 149 without installation), can be connected to the car radio

and guarantee a better sound quality. These systems however need to be built in and are integrated in the dashboard. The most advanced products rely on echo cancellation and noise suppression techniques. The Sonata III echo cancellation and voice enhancement system of NMS Communications[1] was developed for service providers of E1 long distance and digital wireless technology. It is expected that in the near future smaller and more advanced solutions for hands–free telephony will be developed, which can be integrated in the hand–held mobile phones themselves and provide high quality wideband speech enhancement.

### Teleconferencing

Apart from hands–free telephony also teleconferencing fits in the hands–free communication framework presented in section 1.2.1. Teleconferencing systems are commonly used in business meetings today. Teleclassing, which enables students to attend classes and lectures from a remote classroom, is a special case of this. As the participants in a teleconferencing meeting can stay in their local office unnecessary traveling is avoided. Hence, a large cost reduction is obtained and the loss of precious time is kept to a minimum. A market research report from Wainhouse Research states that the market for audio, video and web conferencing services will reach US$ 9.8 billion by 2006, up from US$ 2.8 billion in 2000 [135].

Powerful teleconferencing systems are already commercially available. Polycom, Inc., which acquired PictureTel Corporation in 2001, brings a range of full duplex audioconferencing equipment to the market. These solutions have a limited bandwidth and are suited for small business meetings. Larger systems are also available, such as the iPower™ 900 series of Polycom, Inc. They provide integrated audio and videoconferencing and offer better audio quality. Future systems will have to cope with higher bandwidths and multi–channel signal enhancement, for which efficient signal processing algorithms are needed.

### Domotic and voice–controlled systems

Nowadays there is an increasing interest in so–called domotic systems. More and more voice controlled systems are encountered in daily life at home and at work. These hands–free systems can be used for the automatic conditioning of a living room or the office at work (switching the light or the central heating on and off, opening the curtains, ...). Other examples are voice controlled electronic devices or HiFi systems, the on–board computer in your car, voice controlled PC software, ... . Telematics seems to be the next big challenge in the automotive industry, providing cellular voice and internet services in vehicles. In North America alone the market for telematics equipment is expected to grow to US$ 7 billion in 2007 [180].

[1] NMS Communications acquired the product line of echo cancellers and voice enhancement systems of Lucent Technologies in 2001.

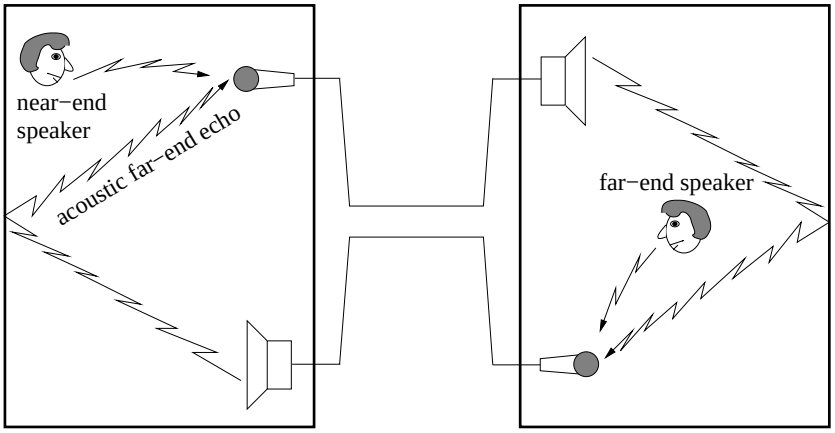


Figure 1.3: Full–duplex hands–free communication setup

In 2001 Ford and Vodafone announced a strategic partnership to provide in–car telematic services. Within five years nearly all new Ford vehicles will be fitted with some telematics system. These systems will include voice recognition and text–to–speech technology to recognize spoken phone numbers as well as the names of previously entered contacts. Advanced signal processing techniques will be needed for adequate signal conditioning and preprocessing.

### 1.2.3 Signal deterioration

Consider again figure 1.2. Ideally, the desired near–end speech signal, which stems from a local correspondent, is sent to the remote site without any quality losses. It is clear that in a hands–free system the signal quality is degraded in many ways. Due to the large speaker–to–microphone distance undesired background signals are recorded and are transmitted to the correspondent as well.

A first type of disturbance are so–called *acoustic echoes*, which arise whenever a far–end loudspeaker signal is picked up by the near–end microphone(s) and is sent to the remote site. At the far–end site the same coupling might exist between loudspeaker and microphone and hence the signal can circle around in the system. The local speaker hears an echo or a delayed version of his/her own speech (figure 1.3). Such delayed signals hinder smooth conversation and lower the speech intelligibility. Delays could be quite long (several hundreds of milliseconds), especially when satellite links are involved. In the worst case the closed–loop gain might become too large and the echo gets unstable, resulting in a harmful sinusoidal tone. A number of techniques can be called for to cancel the echo, hence the name acoustic echo cancellation.

A second source of signal deterioration is *"background noise"*. This type of disturbance can e.g. be generated by a ventilator or a computer fan. It can also come from people in the conference room not participating in the discussion but having a discussion among themselves in the background (cf. cocktail party). In car applications noise is being generated by the engine or by the car radio. It may also come from the wind passing around the car cabin or from the contact between road and tires [94] [160]. Signal processing techniques that are applied to reduce the background noise level are referred to as noise suppression or source separation algorithms. If a reference of the disturbing signal can be obtained, e.g. in the case of radio or engine noise, more specific enhancement techniques can be used. This is called interference cancellation and is very similar to acoustic echo cancellation.

Finally, remark that all signals propagate through the recording room. As a consequence *reverberation* is added to the signals, which leads to another type of signal distortion. Although signals (especially music) may sound more pleasant when reverberation is added, in general the intelligibility is lowered. In order to cope with this kind of deformation dereverberation or deconvolution techniques are called for.

## 1.3 Characteristics of speech and the acoustic environment

The characteristics of speech and the properties of the acoustic environment have an influence on the type of algorithm that is used and on the way the algorithms are applied. In this section some characteristics and peculiarities of speech and acoustics are discussed. Only those properties are mentioned that are important for the algorithms and techniques considered in this thesis. More detailed information on speech and signal processing for speech signals can be found in [29] [124]. A good reference on acoustics is [93].

### 1.3.1 Speech signals

Very often in hands–free applications the signal to be enhanced is speech. Speech is a signal with highly *time–varying characteristics*. Sometimes speech is quasi–periodic (e.g. vowels), at other instances it acts like colored noise (fricatives) or it is impulse–like (plosives). For example, in the word "peace" there is a clear difference between the plosive `/p/`, the vowel `/i:/` and the fricative `/s/`.

Speech is a *wideband signal* with frequency components between 100 and 8000 Hz, hence covering more than 6 octaves. For speech understanding frequencies between 300 and 3400 Hz, i.e. 3.5 octaves, are of most interest. Hence, a sampling rate of 8 kHz is sufficient for low–quality (e.g. classical telephony) speech recording and speech transmission. Nowadays, there is a demand for higher quality, resulting into

so–called wideband speech systems for which higher sampling rates, e.g. 16 kHz, are used.

It is observed that both the time envelope and the spectral content of speech are continuously changing : the energy of the speech signal is both time– and frequency–dependent. The mean frequency envelope of voiced speech is about -6 dB/octave. Signal enhancement algorithms have to cope with the changing frequency dependence and hence often rely on frequency–domain and subband techniques.

The time–domain evolution of the speech signal is characterized by its high *dynamic range* : speech pauses alternate with high energetic vowels or plosives, which significantly increase the short–time energy. This can e.g. be verified in figure 10.12 (chapter 10) were a speech signal is shown at the top. It is found that the amplitude of speech varies between 30 and 90 dB SPL [124]. In order to cope with these amplitude variations 12 to 16 bits linear quantization is commonly used for speech.

Furthermore, due to the high dynamic range of the speech signal, signal enhancement algorithms have to be normalized by the actual signal energy. In this way the algorithm can be prevented from diverging and at the same time slow convergence can be avoided.

### 1.3.2 The acoustic environment

It is observed from figure 1.2 that acoustic waves travel from source to listener and thereby propagate through the recording room. This propagation results in a signal attenuation and spectral distortion. It appears that the attenuation and the distortion can be modelled quite well by a linear filter. Nonlinear effects are typically of second order and mainly stem from the nonlinear characteristics of the loudspeakers. The linear filter that characterizes the acoustics and relates the emitted signal to the received signal, is called the *acoustic impulse response* and plays an important role in many signal enhancement techniques.

Acoustic impulse responses can be measured quite easily, an example of which is given in figure 1.4. Observe that the acoustic impulse response is characterized by a dead time. The dead time is the time needed for the acoustic wave to propagate from source to listener via the shortest, direct acoustic path. After the direct path impulse a set of early reflections are encountered, whose amplitude and delay is strongly determined by the shape of the recording room and the position of source and listener. Next come a set of late reflections, also called reverberation, which decay exponentially in time. These impulses stem from multi–path propagation as acoustic waves reflect on walls and objects in the recording room. Acoustic impulse responses are typically *highly time–varying*, as shown by the following experiment.

**Experiment 1.1** Consider the acoustic impulse response $w_1$ shown in figure 1.4. To compute $w_1$ a white noise signal $x$ was sampled at 44100 Hz and fed into a loud-

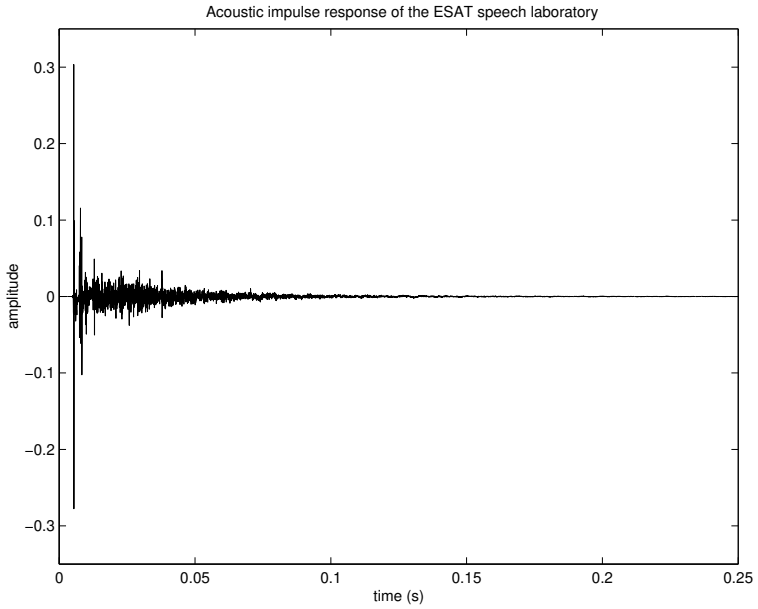


Figure 1.4: Acoustic impulse response of the ESAT speech laboratory

speaker. The response $y = w_1 \star x$ was recorded with a microphone. The distance between loudspeaker and microphone was approximately 180 cm. Based on the loudspeaker and microphone signal, $w_1$ could be determined. Then the experiment was repeated. The configuration was slightly changed, moving the microphone 1 cm to the left and leaving the position of the loudspeaker and the rest of the environment unchanged. Again the acoustic impulse response was computed, resulting in $w_2$. Despite the small change in microphone position the impulse response changed substantially : it was found that

$$\frac{||w_1 - w_2||_2}{||w_1||_2} = 72\%.$$

To simulate the effect of moving correspondents in the recording room a dummy was placed between loudspeaker and microphone and the impulse response ($w_3$) was computed. Then the dummy was moved approximately 1 cm. All other objects were left unchanged. Again the acoustic impulse response $w_4$ was determined. In this case

$$\frac{||w_3 - w_4||_2}{||w_3||_2} = 34\%.$$

$\bigtriangledown$

In order to cope with these time variations adaptive signal processing techniques

are called for. Thanks to the continuous updating these algorithms are more or less robust against possible system variations.

To characterize the amount of reverberation in a recording room the *reverberation time* ($RT_{60}$) is defined as the time that the sound pressure level or the intensity needs to decay to e.g. -60 dB of its original value. It is therefore a measure of the decay and of the duration of the acoustic impulse response. It appears to be independent of the actual position of source and listener. The reverberation time was computed for the impulse response shown in figure 1.4 following the method described in [60]. It appeared that $RT_{60} \approx 240$ ms.

Typical reverberation times are in the order of hundreds or even thousands of milliseconds. For a typical office room $RT_{60}$ is between 100 and 400 ms, for a church $RT_{60}$ can be several seconds long. If therefore in a digital signal enhancement application the acoustic impulse responses are characterized by *FIR filters* many hundreds or several thousands of filter taps are needed, depending on the sampling rate. Hence, computationally efficient algorithms are required.

In order to reduce the filter order, i.e. the number of delay elements, *IIR models* could be called for. It appears that although the order can be reduced in this way it still remains quite large, i.e. in the order of several hundreds [75] [108]. IIR–based enhancement techniques have to be relied on in that case, typically leading to either an increased computational load, or stability problems and convergence to local minima [108] [141].

In order to optimally control the experiments carried out in the frame of this thesis simulated room impulse responses were often used. These simulated acoustic impulse responses were designed following the method described in [4] [129] [154].

## 1.4 Enhancement techniques

Each of the three forms of signal degradation that arise in hands–free communication are now discussed in more detail, emphasizing existing algorithmic solutions that are known from the literature.

### 1.4.1 Acoustic echo cancellation

Experiments have shown that suppressing the acoustic echoes with 45 dB leads to satisfactory perceptual results, as long as the overall delay introduced by the echo canceller doesn't exceed a certain upper bound. The input–output delay is clearly constrained in hands–free telephony for instance, but is less stringent if no reaction from the remote site is required. Internationally standardized requirements

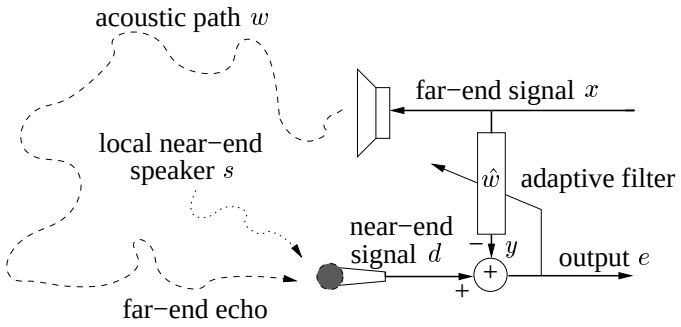


Figure 1.5: Adaptive acoustic echo cancellation

with respect to echo cancellation are contained in the ITU–T recommendations[2] (G.167) on acoustic echo controllers [86]. For instance, the end–to–end delay is recommended not to exceed 16 ms for wideband teleconferencing. The far–end signal suppression (when no near–end signal is present) should reach 40 dB for teleconferencing systems and 45 dB in hands–free telephony. In presence of near–end signals (double talk) the suppression should be at least 25 dB. Convergence to a 3 dB attenuation level should last less than 20 ms in the case of single talk.

To suppress the echo several *conventional acoustic echo cancellation techniques* can be applied [77]. For instance, highly directional loudspeakers and microphones and sound absorbing materials can be used to avoid reflections. Another popular technique is voice controlled switching or loss control, which mutes channels in which no or very low–energetic activity is measured. It is clear that these techniques rely on accurate voice activity detection and hence quickly degrade. Further, the stability margin of the closed–loop system can be improved using so–called howling control. Thereto almost inaudible nonlinear operations are inserted in the signal path to avoid instability of the closed–loop system, as this would result in a harmful sinusoidal tone circling around in the network. Frequency–shifting, comb filters and resonant peak removal are often used. Finally, nonlinear post–processing devices can be added to remove residual error signals and to make the signal more pleasant to listen to.

In practice nowadays acoustic echo cancellers are based on *adaptive filtering techniques* [76] [77] [106] [176]. Adaptive filters will be discussed in section 2.3. A general adaptive acoustic echo cancellation setup is shown in figure 1.5. If the adaptive filter $\hat{w}$ is a good estimate of the acoustic impulse response $w$ it is observed that

$$\begin{aligned} e[k] &= d[k] - y[k] && (1.1)\\ &= (s[k] + w \star x) - \hat{w} \star x && (1.2)\\ &\approx s[k], && (1.3) \end{aligned}$$

[2] ITU stands for International Telecommunication Union and replaces the CCITT since 1993.

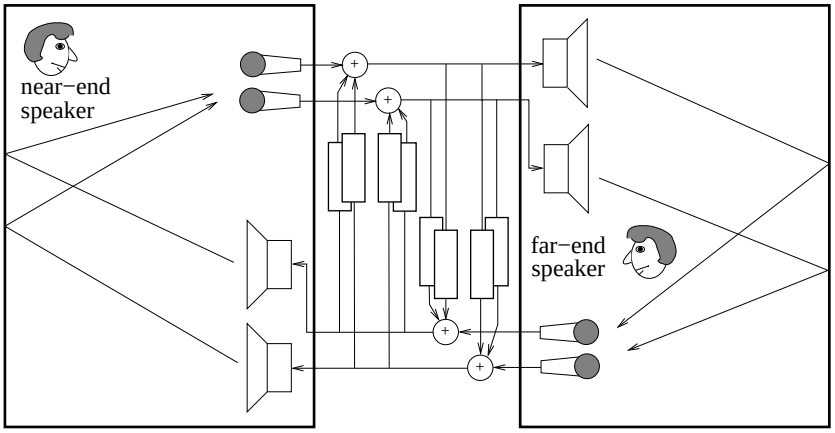


Figure 1.6: Stereo acoustic echo cancellation setup

hence the echo can be removed. The adaptive filter $\hat{w}$ is a self–designing system that uses a gradient algorithm that minimizes the error signal energy. In this way a good replica of the unknown system $w$ can be obtained. Apart from the ability to obtain a good echo path replica, time variations of the acoustic impulse response can be tracked as well, thanks to the adaptivity. However, accurate tracking of the acoustic impulse response $w$ is still a challenge even if fast and hence expensive adaptive filtering structures are applied [62] [162] as acoustic impulse responses are known to be highly time–varying (cf. experiment 1.1).

In more advanced systems two or more loudspeaker channels have to be cancelled as shown in figure 1.6. It can be proven that *stereo* or —in general *multi–channel*— *acoustic echo cancellation* inherently suffers from a non–uniqueness problem [113]. In practice however, a unique solution to the stereo echo cancellation problem does exist, but the underlying optimization that drives the adaptive filters appears to be severely ill–conditioned. Several techniques were developed that cope with this issue. They try to decorrelate the stereo channels by insertion of nonlinearities in the signal paths or by applying psycho–acoustic noise masking techniques [58] [68] [87] [121].

Although commercial adaptive echo controllers are available on the market nowadays, providing a merely satisfactory solution to the single–channel acoustic echo cancellation problem, further improvement and research will be necessary in the coming years. It is for instance clear that in the near future there will be a need for $N$–channel acoustic echo controllers (e.g. for stereo, surround systems, Dolby Digital 5.1). Remark that the number of adaptive filters in an $N$–channel echo cancellation system equals $N^2$ (cf. figure 1.6), hence computationally efficient and intelligent controllers are needed. Further, whereas present–day echo cancellers

mostly operate at rather low sampling rates (8 kHz) higher quality will be required in the near future (16 kHz, or even higher). As the complexity of an echo cancellation system using a linear adaptive filtering algorithm, changes quadratically with the sampling rate, again efficient adaptive structures will be needed. Finally, there will be a request for a better overall performance and more robustness in highly non–stationary and complex acoustic environments. This requires reliable control software, which is added to the adaptive filtering scheme.

### 1.4.2 Noise suppression and interference cancellation

*Single–channel noise reduction* methods have been known for a long time now. They exploit the characteristics of speech and the noise and enhance the SNR by appropriate (matched or Wiener) filtering operations [149]. More advanced techniques, commonly used today, rely on spectral subtraction [11] [182].

Noise suppression is a difficult problem. It is observed that the signal of interest and the background noise typically overlap both in the time and in the frequency domain. This is certainly true when both signals are speech. The signal of interest and the "noise" are therefore difficult to separate if classical spectro–temporal enhancement techniques are employed.

It is observed however that the correspondent and the background noise source are typically at a different position in the conference room. Hence, *multi–microphone techniques* can be called for, which exploit the spatial information present in the different microphone signals. This in general leads to spatio–temporal filtering operations and increases the performance.

A first class of enhancement techniques that rely on this spatial diversity is *beamforming*. The beamforming idea comes from telecommunications where it was introduced to design antenna arrays. Later it was successfully applied to acoustic applications as well. As the acoustic environment is inherently time–varying adaptive beamforming techniques are often called for. Broadband beamforming for speech enhancement is still a topic of ongoing research [17] [18] [34] [66] [74] [91] [92] [97] [122] [123] [125] [136] [150] [158] [161] [164] [165] [172] [174] [175].

More recently *optimal filtering techniques* have been proposed for the suppression of additive broadband noise. These techniques rely on powerful matrix decompositions such as the SVD and the Quotient SVD [33] [148]. They show a superior performance compared to classical beamforming approaches but are computationally more demanding.

If a reference of the noise signal can be obtained more specific signal enhancement techniques can be applied. For instance, in the case of engine noise in a car the spark signal can be measured and used to suppress the noise in the car cabin. Adaptive filtering techniques are then typically called for. Remark also that echo cancellation can be viewed as a special case of this.

Echo cancellation and noise suppression have been addressed independently for many years now. Recently, it has been recognized that both problems are better tackled in a combined approach, especially if multi–microphone settings are being used. Initial results indicate that the combined approach yields a better performance at a lower computational cost [1] [31] [63] [102] [103] [104] [105].

Multi–microphone noise reduction schemes are being commercialized nowadays. The systems that are available on the market however are typically rather basic solutions with a limited number of microphones and often relying on simple, not fully adaptive signal processing tools. There is certainly a need for more powerful and robust systems with a higher performance at an acceptable cost in the forthcoming years.

### 1.4.3 Dereverberation

Of the three types of signal deterioration that occur in hands–free communication dereverberation is least prominent. However, in rooms with a high reflectivity reverberation effects have a clearly negative impact on the intelligibility. Dereverberation techniques have been developed over the last years but the solutions available today are not yet satisfactory.

*Single–channel dereverberation* techniques were reported first. Inverse filtering can be called for, by trying to invert the acoustic impulse response. However, as the impulse responses are known to be non–minimum phase systems they have an unstable inverse [112] [120]. Cepstrum–based techniques are more promising [6] [126] [131] and rely on the separability of speech and the acoustics in the cepstral domain.

Through *multi–channel processing* the spatial diversity of the hands–free setup can be exploited, in general leading to a better performance. Acoustic beamforming techniques are being used, as apart from noise suppression they are known to partially dereverberate the signals as well. A second class of multi–channel dereverberation techniques is based on cepstral processing. It was shown that the single–channel cepstral based dereverberation algorithms can be extended to the multi–channel case [96].

Matched filtering algorithms were reported in [2] [167]. They rely on subspace tracking techniques. These algorithms show an improved dereverberation capability with respect to classical approaches but as some environmental parameters are assumed to be known in advance these approaches may be less suitable in practical applications.

During the last years MIMO blind system identification techniques have been developed for equalization in digital communications [80] [118] [163] [166]. These techniques might be extendible to speech enhancement applications but this is still a topic of ongoing research [67].

## 1.5 Outline of the thesis and contributions

In this section an outline and an overview of the thesis can be found. The main contributions are summarized and references will be given to the publications that were brought about in the frame of this work.

### 1.5.1 Motivation

In this thesis subband and frequency–domain adaptive filtering techniques are studied, putting forward acoustic echo cancellation as a possible and straightforward application.

Acoustic echo cancellation, as well as other signal enhancement problems in hands–free communication, deals with the retrieval of degraded speech embedded in "noise". To enhance the speech signal the acoustics of the recording room need to be estimated. In section 1.3 we discussed some properties of speech and the acoustic environment that impose specific constraints on the signal enhancement algorithm that is used. It was for instance observed that acoustic impulse responses are time–varying high–order systems. It was further indicated in section 1.4.1 that there will be a need for (multi–channel) acoustic echo controllers in the near future that offer a high performance at increasing sampling rates. Hence, *computationally efficient* and *adaptive* algorithmic solutions should be called for. Finally, as the time envelope and the spectral content of speech are continuously changing *time– and frequency–dependent* signal processing is required.

It will appear in the forthcoming chapters of the thesis that subband and frequency–domain adaptive filtering techniques meet all the requirements specified above, combining adaptivity and frequency–dependent processing, and offering a high performance at a low cost. Hence, subband and frequency–domain adaptive filters will be put forward as being appropriate approaches to solve the acoustic echo cancellation problem.

It is not only our objective to present existing and novel subband and frequency–domain adaptive solutions for acoustic echo cancellation, we will also dwell on the structures and principles that lie behind these techniques, in an attempt to get more insight in the underlying fundamentals. Whereas acoustic echo cancellation was presented as the starting point and a motive for this research, the main part of the text deals with signal processing as such. The presented techniques can be employed in many applications, going far beyond acoustic echo cancellation alone.

### 1.5.2 Chapter by chapter overview and contributions

The thesis consists of four parts. Each of them is divided in one or more chapters. A schematic overview of the different parts and chapters is presented in figure 1.7.

The introductory and concluding chapter are omitted however in the figure.

In **chapter 2** some basic concepts are discussed and the necessary signal processing tools will be presented to understand the main part of the text.

**Part I : DFT modulated filter bank design for oversampled subband systems**

It was motivated in this introductory chapter that frequency–dependent adaptive signal processing is required for adequate acoustic echo suppression. Frequency dependency can be achieved through the use of digital filter banks and the integration of these structures in existing adaptive filtering schemes, leading to so–called subband adaptive filters. In general however, digital filter banks introduce considerable signal and aliasing distortion. In part I of the thesis design methods for perfect and nearly perfect reconstruction DFT modulated filter banks are discussed. These filter banks introduce no or almost no signal distortion and are easily integrated in subband adaptive filtering structures.

In **chapter 3** design methods for perfect reconstruction oversampled DFT modulated filter banks are presented. A para–unitary filter bank design method is discussed, which was presented in [22]. With this method however the order of the filter banks cannot be adjusted accurately. We present an extension to this method, which basically allows to choose any desired filter length. Further, we show that based on the inverse parametrization of the filter bank parameters appropriate starting values can be obtained, which reduces the optimization time.

The stopband attenuation of perfect reconstruction filter banks is typically unsatisfactory if intermediate operations, such as adaptive filtering, are performed on the subband signals. In **chapter 4**, the perfect reconstruction condition is relaxed to nearly perfect reconstruction. Both a frequency–domain and a mixed time/frequency–domain based design method are presented for nearly perfect reconstruction DFT modulated filter banks. Subband adaptive filtering is taken as an example to illustrate that thanks to their lower stopband level nearly perfect reconstruction filter banks outperform perfect reconstruction systems.

Publications related to the first part of the thesis are [43] [45] [52].

**Part II : Subband and frequency–domain adaptive filtering**

In section 1.5.1 subband and frequency–domain adaptive filters were put forward as being appropriate approaches to solve the acoustic echo cancellation problem. In this part we will concentrate on subband and frequency–domain adaptive filtering

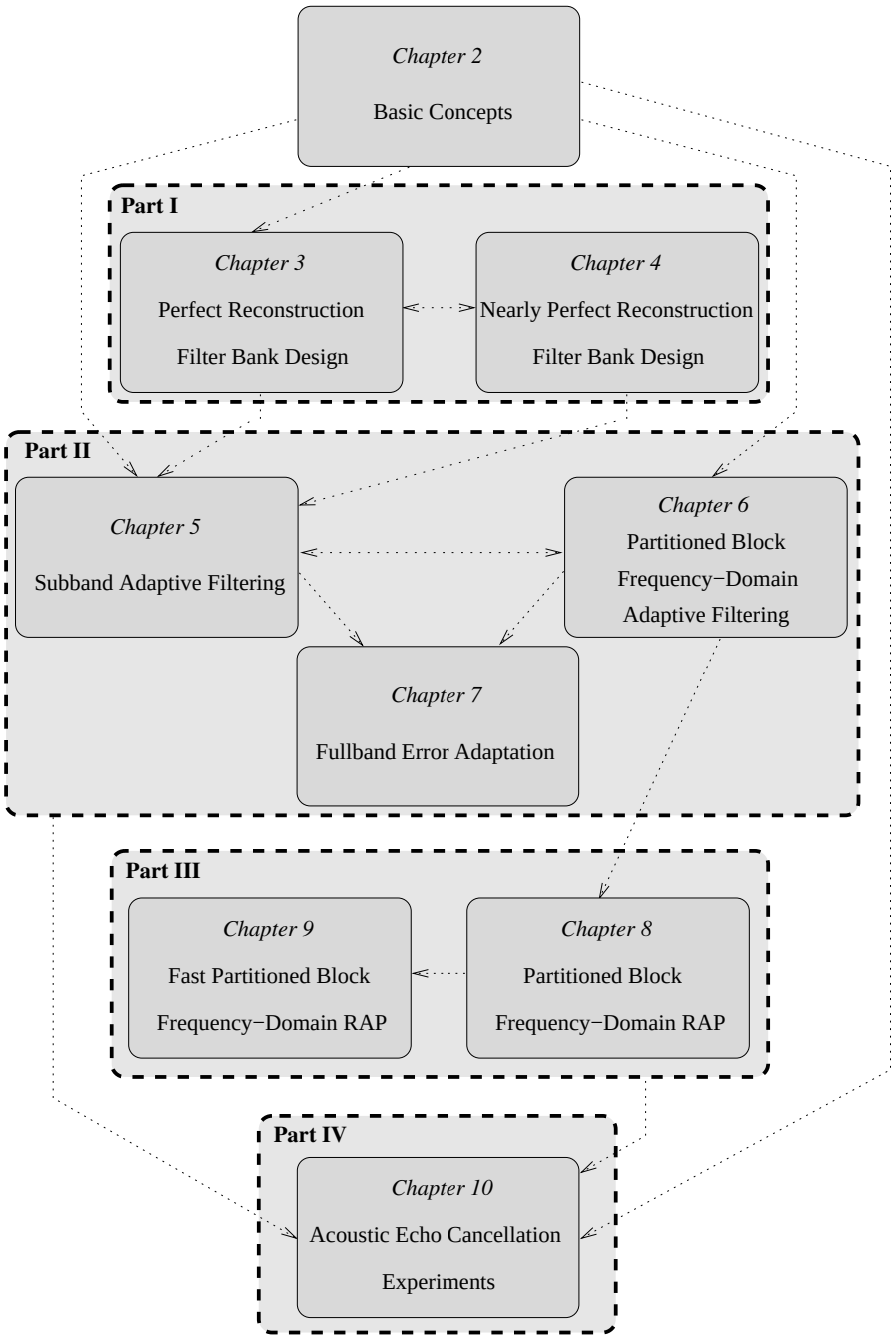


Figure 1.7: Schematic overview of the thesis

in more detail and discuss some of their properties. Although both approaches were developed independently in the literature they are strongly connected to each other. We will focus on the interrelation between both techniques and combine their mechanisms to obtain improved algorithmic structures.

The subband adaptive filter is discussed in **chapter 5**. A comparison is made between the subband approach and standard fullband adaptive filters in terms of complexity and performance. It will be shown that subband adaptive filtering structures suffer from a considerable residual undermodelling error unless extra (anti–)causal subband filter taps are inserted. Although the complexity gain w.r.t. the fullband approach is less than expected, still a considerable cost reduction can be obtained. Next, we formulate three design criteria for subband adaptive systems, which deal with frequency selectivity, perfect reconstruction and perfect path modelling. These conditions are necessary requirements to ensure satisfactory performance of the subband adaptive filter.

In **chapter 6** the partitioned block frequency–domain adaptive filter (PBFDAF) is studied. It appears that this algorithm, which is known from the literature for some years now, outperforms standard subband systems in terms of convergence behavior and modelling capabilities. It will be proven that the PBFDAF can be considered as a special subband adaptive filtering structure, which fulfills two out of the three design criteria for subband adaptive systems that are specified in chapter 5. It is further shown that the frequency–domain adaptive filter relies on a special error correction mechanism. Thanks to the error correction the filter coefficients can be updated with aliasing–free error signals, which leads to improved performance.

In an attempt to generalize the error correction mechanism of the frequency–domain approach to subband adaptive systems we propose a novel fullband error adaptation scheme for subband adaptive filters in **chapter 7**. The alternative adaptation scheme adjusts the subband filters based on the fullband error instead of using the subband errors, as is done in a classical subband adaptive system. In this way improved performance is obtained. It is shown that for some common parameter settings the weight update mechanism of the so–called unconstrained PBFDAF corresponds to that of the fullband error adaptation algorithm presented in this chapter. This proves that the fullband error adaptation algorithm can be considered as an extension of the frequency–domain error correction mechanism to a more general class of subband adaptive filters.

Publications related to this part are [41] [42] [47] [48] [49] [51] [53] [56] [57].

**Part III : Iterated partitioned block frequency–domain adaptive filtering**

In part III an extension to the PBFDAF is proposed, called the PBFDRAP, which is an adaptive filtering algorithm combining partitioned block frequency–domain adaptive filtering with so–called row action projection. The algorithm is presented and analyzed and fast implementation schemes are derived.

> In **chapter 8** the PBFDRAP is defined and it is explained how extra error suppression can be achieved w.r.t. the PBFDAF. Further, the asymptotic properties of the algorithm are analyzed : for some parameter settings the PBFDRAP algorithm approaches well–known adaptive filtering algorithms. Finally, it is shown that the PBFDRAP outperforms the PBFDAF in a realistic echo cancellation setup.
>
> Fast implementations are derived for the PBFDRAP algorithm in **chapter 9**. The different fast implementation schemes are compared with the standard implementation of the PBFDRAP for different parameter settings. It appears that a significant complexity reduction can be obtained. The PBFDRAP adaptive filter is then compared with the PRA algorithm from a computational complexity point of view. It is seen that for large block lengths the PBFDRAP is a cheaper alternative to the PRA.

Publications related to this part are [50] [54] [55].

**Part IV : Acoustic echo cancellation, implementation and experiments**

In the final part of the thesis the acoustic echo cancellation problem is revisited. It was pointed out in section 1.4.1 that in the near future there will be a request for more robust acoustic echo cancellation schemes offering a better overall performance in highly non–stationary and complex acoustic environments. This requires reliable control software, which is added to the adaptive filtering scheme to monitor the adaptation speed.

> **Chapter 10** illustrates how the different adaptive filters developed in the preceding chapters can be applied to an acoustic echo cancellation setup, providing them with control and so–called double–talk detection techniques, known from the literature. Several experiments are discussed, different adaptive filtering solutions are compared and some observations concerning a real–time implementation of an acoustic echo canceller on DSP are presented.

Publications related to this part are [44] [46].

The final conclusions and some suggestions for further research are formulated in **chapter 11**.

## 1.6 Conclusions

In the first section of this chapter the economic impact of telecommunication technology and hands–free communication in particular was highlighted and a motivation was given for the work that was performed in the frame of this thesis.

In section 1.2 hands–free communication was defined, examples were given and it was pointed out that different sorts of signal degradation do occur.

In section 1.3 some basics of speech and acoustics were studied.

It was shown in section 1.4 that a large variety of signal enhancement techniques are known from the literature. They can be employed in present–day hands–free communication systems to obtain a better signal quality.

In section 1.5 an outline and an overview was given of the different chapters and parts of the thesis and the main contributions were presented.

# Chapter 2

# Basic Concepts

In this chapter some basic concepts are discussed and the necessary signal processing tools are presented to fully understand the forthcoming chapters of the thesis.

Many of the algorithms described in this thesis are so–called block based adaptive filters. Often in this kind of algorithms signals with different sampling rates coexist, hence the name multirate systems. In section 2.1 some basics of signal processing and of multirate signal processing in particular are therefore discussed.

Part I of the thesis focuses on digital filter bank design. These filter banks can then be integrated in the subband adaptive filtering structures that are discussed in part II. Section 2.2 therefore discusses some filter bank fundamentals and presents the necessary background information that is needed to fully understand part I and II of this work.

The algorithms presented in part II and III are adaptive filters. A brief overview of existing adaptive filtering techniques will be given in section 2.3.

For many of the algorithms that are discussed further on, a cost analysis will be performed. The assumptions we will make for these cost analyses are summarized in section 2.4.

Section 2.5 formulates the conclusions to this chapter.

## 2.1 Signal processing basics

### 2.1.1 Representation of variables

Most of the signals, filters and systems that are referred to in this thesis are discrete–time variables. They are represented in the time, the frequency or in the $z$–domain.

The *time–domain* representation of a variable $h$

$$h[k] = \{ \ \ldots \quad h[-1] \quad h[0] \quad h[1] \quad h[2] \quad \ldots \ \} \tag{2.1}$$

depends on the discrete time $k$, which relates to the actual time $t = k/f_s$ via the sampling frequency $f_s$.

$H(z)$ is the *z–transform* of $h[k]$ and is defined as

$$H(z) = \sum_{k=-\infty}^{\infty} h[k] z^{-k}. \tag{2.2}$$

An in–depth discussion of the use and validity of the $z$–transform can be found in many books on signal processing [126] [134] or control theory [65] [110].

By evaluating $H(z)$ on the unit circle, i.e. replacing $z$ by $e^{j2\pi f}$ in Eq. 2.2, the *frequency–domain* representation of $h[k]$ is obtained :

$$H(f) = \sum_{k=-\infty}^{\infty} h[k] e^{-j2\pi kf}. \tag{2.3}$$

$H(f)$ is periodic in the frequency $f \in \mathbb{R}$. For the evaluation of the frequency–domain characteristics the fundamental interval (1 period) is usually considered, i.e. $-\frac{1}{2} < f \leqslant \frac{1}{2}$, in which $f = 0.5$ corresponds to the Nyquist frequency. The inverse frequency–domain transformation

$$h[k] = \int_{-\frac{1}{2}}^{\frac{1}{2}} H(f) e^{j2\pi kf} df \tag{2.4}$$

computes $h[k]$ from $H(f)$.

### 2.1.2 Multirate signal processing

In many of the adaptive filtering algorithms discussed in this thesis signals with different sampling rates are encountered. As different sampling rates coexist within the same algorithm these adaptive filtering structures are called multirate systems. The main reason to change the sampling rate is the complexity reduction that can be achieved.

To fully describe a multirate system in the time domain several discrete–time variables should be defined and used in parallel. It is however more convenient to represent these systems in the $z$–domain.

To describe the conversion from one sampling rate to another, two operations will be discussed here : the reduction of the sampling rate with an integer factor, called downsampling, and the increase of the sampling rate with an integer factor, which is referred to as upsampling.

**Downsampling**

$f[m]$ is an $N$–fold downsampled version of $h[k]$ if

$$f[m] = h[k]_{N\downarrow} = h[mN], \qquad \forall m \in \mathbb{Z}, N \in \mathbb{N}_0. \tag{2.5}$$

In this way the sampling rate is reduced by a factor $N$. In signal flow graphs $N$–fold decimators or downsamplers are represented as $\boxed{N\downarrow}$. It can be shown [156] that

$$F(z) = \frac{1}{N}\sum_{n=0}^{N-1} H(z^{1/N} e^{-j\frac{2\pi n}{N}}) \tag{2.6}$$

holds, in which $f[k] \longleftrightarrow F(z)$ and $h[k] \longleftrightarrow H(z)$ are $z$–transform pairs.

**Upsampling**

$f[m]$ is an $N$–fold upsampled version of $h[k]$ if

$$f[m] = h[k]_{N\uparrow} = \begin{cases} h[m/N] & \text{if } m = pN, \quad \forall p \in \mathbb{Z}, N \in \mathbb{N}_0 \\ 0 & \text{otherwise,} \end{cases} \tag{2.7}$$

which increases the sampling rate by a factor $N$. In signal flow graphs $N$–fold expanders or upsamplers are marked as $\boxed{N\uparrow}$. It can be shown [156] that

$$F(z) = H(z^N). \tag{2.8}$$

Both operations introduce artifacts. In the case of downsampling aliasing is added to the signal. Upsampling invokes so–called mirror frequencies. More information about this and how to get rid of the artifacts can be found in any good book on (multirate) signal processing, e.g. [156].

### 2.1.3 Some definitions related to matrix algebra

In appendix A a few matrix algebra definitions and properties are combined that will be used and referred to in the forthcoming chapters of the thesis. A good reference on matrix algebra that focuses on numerical aspects and implementation issues is [73].

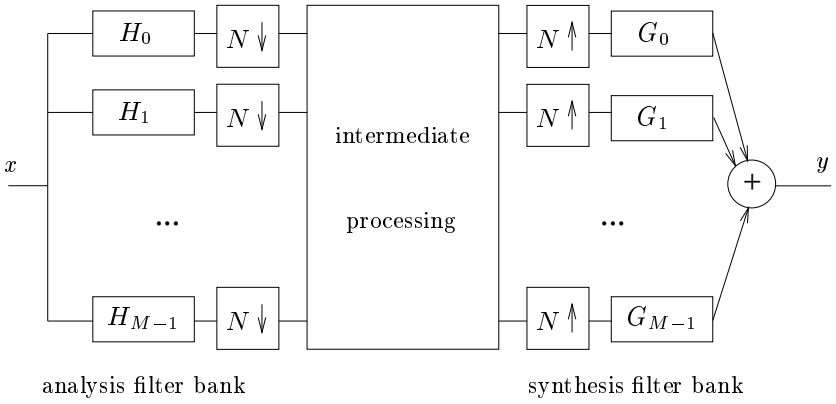


Figure 2.1: A general subband scheme : all intermediate operations can be done at the downsampled rate, which typically leads to a reduced implementation cost and improved performance.

## 2.2 Filter bank basics

Filter banks are widely used in digital signal processing [156]. Typical applications are subband coding [87] [169] and subband adaptive filtering [53] [142]. Subband techniques can improve the performance of standard fullband algorithms for speech, audio or image processing, as they allow an optimal tuning of the algorithm in each subband. In this way subband algorithms often outperform their globally tuned fullband counterparts. Furthermore, by using multirate techniques the implementation cost can typically be reduced.

### 2.2.1 General subband scheme

A general subband scheme is shown in figure 2.1. The so–called analysis filter bank splits the input $x$ in a set of smallband signals : the input is filtered with each of the $M$ analysis filters $H_0, \ldots, H_{M-1}$ and each subband signal is $N$–fold downsampled. Hence, intermediate operations can be performed on the subband signals at the downsampled rate and this typically leads to a cheaper implementation. Finally, a recombination operation takes place in the synthesis filter bank $G_0, \ldots, G_{M-1}$, which operates on the $N$–fold upsampled subband channels and results in the output $y$.

A filter bank is a set of parallel filters, which each filter out a part of the frequency spectrum. If all filters have the same bandwidth the filter bank is said to be uniformly spaced. In some cases non–uniformly spaced filter banks are preferred. Non–uniformly spaced filter banks more closely approach the mechanism of human

perception and therefore the bandwidth of the different filters is changed in a logarithmic way. Non–uniformly spaced filter banks are often tree– or wavelet–based [156] [169] and might be more suitable for applications such as audio or video. In this thesis only uniformly spaced filter banks will be considered.

Uniformly spaced filter banks are typically obtained by modulating, i.e. frequency shifting a well–designed lowpass prototype filter. Hence, they are called modulated filter banks. Each of the analysis filters $H_0, \ldots, H_{M-1}$ then filters out a part of the frequency spectrum and as there are $M$ subbands in total the bandwidth of each of the analysis filters is equal to (or larger than) $\frac{f_s}{M}$, in which $f_s$ represents the sampling frequency corresponding to the input signal signal $x$. Modulated filter banks can easily be implemented by decomposing the prototype filter in polyphase components and applying a DFT or DCT operation (see sections 2.2.3, 3.1 and [156]). The latter operations can be implemented efficiently using fast signal transforms.

#### Critically downsampled subband schemes

For a uniformly spaced modulated filter bank the bandwidth of each of the filters is larger or equal to $\frac{f_s}{M}$. Hence, if the downsampling factor $N$ is larger than $M$ a considerable amount of aliasing will be inserted in the subbands by the downsampling operation. Aliasing is often detrimental for the performance of the intermediate subband operations (e.g. subband adaptive filtering, see also figure 5.10). Hence in practice, $N$ is restricted to be smaller or equal to $M$, with $M = N$ being an upper bound for $N$. A subband system for which $M = N$ is called a critically downsampled subband scheme. In this case the implementation cost can be optimally reduced : the intermediate operations and in many cases also the filter bank operations (see section 2.2.3) can be done at the lowest sampling rate, as $N$ is as large as possible.

#### Oversampled subband schemes

In practice however, finite order filter banks have to be used in order to limit the processing delay and the computational complexity. Finite order filters have a non–negligible transition bandwidth and therefore aliasing will be inserted in the subbands even if the downsampling factor $N$ is smaller than $M$. Critically downsampled finite order subband schemes are strongly sensitive to aliasing and in many cases the loss in performance due to the critical downsampling is not acceptable. From this point of view oversampled subband schemes ($M > N$) are more attractive as they trade off between complexity reduction and aliasing distortion.

### 2.2.2 Modulated filter banks

Computationally efficient subband systems are often based on modulated filter banks, which typically involves DFT or DCT operations [156].

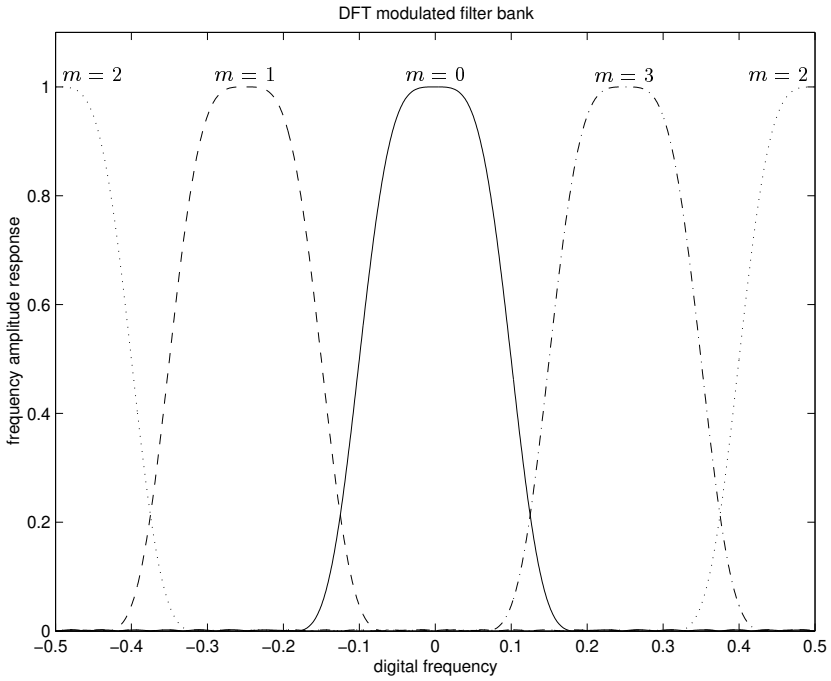


Figure 2.2: 4–band DFT modulated filter bank : frequency amplitude response

The $M$ subband filters of a DFT modulated filter bank are derived by frequency shifting a well–designed lowpass prototype filter $h_0[k]$ of length $L_f$ in the following way :

$$h_m[k] = h_0[k]e^{-j\frac{2\pi km}{M}}, \qquad \begin{cases} m = 0 : M-1 \\ k = 0 : L_f - 1 \end{cases} \tag{2.9}$$

$$\Longleftrightarrow$$

$$H_m(z) = H_0(e^{j\frac{2\pi m}{M}}z) \tag{2.10}$$

$$\Longleftrightarrow$$

$$H_m(f) = H_0\left(f + \frac{m}{M}\right). \tag{2.11}$$

Equation 2.10 follows from Eqs. 2.2 and 2.9 and Eq. 2.11 can be obtained from Eq. 2.10 by replacing $z$ by $e^{j2\pi f}$. The filters are frequency shifted versions of each other and the complete set of $M$ filters covers the whole frequency spectrum. An example of a DFT modulated filter bank, with $M = 4$ is shown in figure 2.2.

A variant of this is the Inverse DFT modulated filter bank, which is defined as

$$h_m[k] = h_0[k]e^{j\frac{2\pi km}{M}}, \qquad \begin{cases} m = 0 : M-1 \\ k = 0 : L_f - 1 \end{cases} \tag{2.12}$$

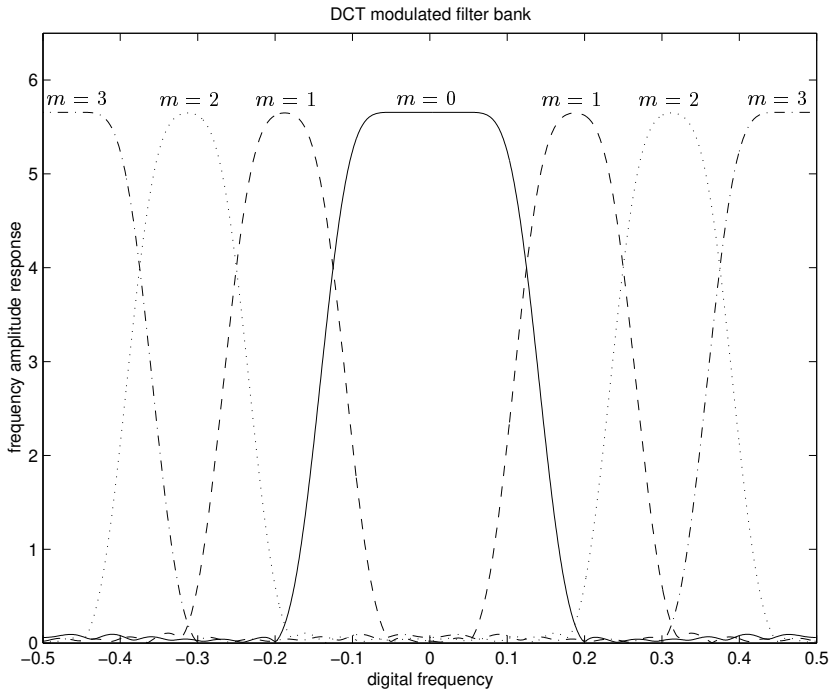


Figure 2.3: 4–band DCT modulated filter bank : frequency amplitude response

The $z$–transform and frequency–domain representation are given by

$$H_m(z) = H_0(e^{-j\frac{2\pi m}{M}}z). \tag{2.13}$$

$$\Longleftrightarrow$$

$$H_m(f) = H_0\left(f - \frac{m}{M}\right). \tag{2.14}$$

IDFT modulation differs from DFT modulation only in the order in which the filters are frequency shifted.

Apart from DFT modulated filter banks cosine or DCT modulated filter banks are often used. Based on a well–designed FIR prototype filter $p[k]$ of length $L_f$ the different DCT analysis filters $h_m[k]$ can be derived as follows [156] :

$$h_m[k] = 2p[k]\cos\left(\frac{\pi}{M}\left(m+\frac{1}{2}\right)\left(k-\frac{L_f-1}{2}\right)+(-1)^m\frac{\pi}{4}\right),\ m=0:M-1. \tag{2.15}$$

An example of a DCT modulated filter bank, with $M = 4$ is shown in figure 2.3. Whereas DCT modulated filter banks are typically used in critically downsampled subband systems, they are less suitable for oversampled schemes. Cosine modulated filters have a positive and negative frequency band. The positive and negative

frequent subbands tend to overlap when the subband signals are not–critically downsampled. In this way a large amount of aliasing is inserted in the subbands. As it is our goal to design oversampled filter banks ($M > N$) that introduce only a small quantity of subband aliasing, standard DCT modulated filter banks are not applicable. Some schemes have been proposed that combine real filter banks with unequal subsampling in different bands to overcome this problem [78].

In the case of DFT modulated filter banks on the other hand the subband filters filter out a single contingent frequency region. If there are $M$ subbands and the lowpass prototype filter has a good stopband rejection, the bandwidth of the bandpassed signals that are filtered out by each of the subband filters is approximately $\frac{f_s}{M}$. Hence, the subband signals are correctly projected into the new fundamental interval $[\frac{-f_s}{2N}, \frac{f_s}{2N}]$ by the downsampling operation if $M \geqslant N$, avoiding severe aliasing distortion. As a consequence, oversampled subband schemes are often based on DFT modulated filter banks because of their aliasing robustness and ease of implementation.

### 2.2.3 Polyphase implementation

The analysis and the synthesis filter bank are immediately followed, respectively preceded by downsampling or upsampling units (see figure 2.1). Hence, it is cheaper to do not only the intermediate processing, but also the filter bank operations at the downsampled rate, which can be achieved through polyphase decomposition.

#### Analysis bank

If the signals passing through the $M$–band analysis bank are subsequently $N$–fold downsampled, each subband filter $h_m[k]$ can be decomposed in its $N$–th order polyphase components :

$$H_m(z) = \sum_{n=0}^{N-1} z^{-n} H_{m_{n:N}}(z^N), \tag{2.16}$$

in which $H_{m_{n:N}}(z)$ is the $n$–th out of $N$ polyphase components of the $m$–th subband filter $h_m[k]$, in other words the $z$–transform of $h_m[n+Nk]$, $k = 0, 1, \ldots$ .

Swapping the polyphase components and the downsamplers leads to a more efficient implementation. The filtering operations can now be done at the lower sampling rate, as shown in figure 2.4 for the $m$–th subband.

The analysis bank can now schematically be represented as shown in figure 2.5. $\mathbf{H}(z)$ is called the analysis polyphase matrix [156]. Element $(m, n)$ of $\mathbf{H}(z)$ is

$$[\mathbf{H}(z)]_{m,n} = H_{m_{n:N}}(z) \qquad \begin{cases} m = 0 : M-1 \\ n = 0 : N-1 \end{cases} \tag{2.17}$$

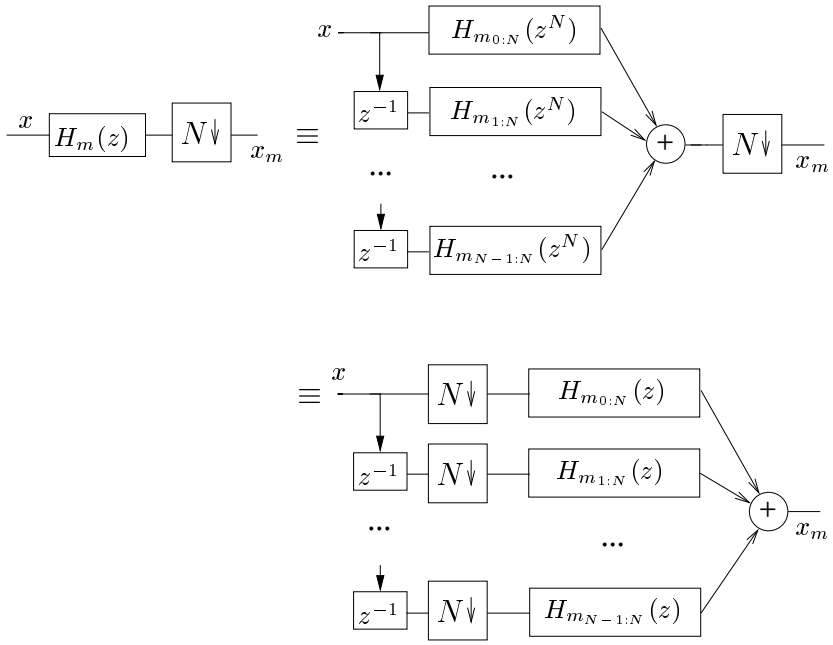


Figure 2.4: Analysis filter polyphase decomposition

**Synthesis part**

For the synthesis part a similar derivation can be made. By polyphase decomposition

$$G_m(z) = \sum_{n=0}^{N-1} z^{-n} G_{m_{n:N}}(z^N), \tag{2.18}$$

and swapping the polyphase components and the upsamplers figure 2.6 is obtained.

All $N$–th order polyphase components are contained in the synthesis polyphase matrix $\mathbf{G}(z)$ [156] :

$$[\mathbf{G}(z)]_{m,n} = G_{m_{n:N}}(z). \qquad \begin{cases} m = 0 : M-1 \\ n = 0 : N-1 \end{cases} \tag{2.19}$$

By combining the analysis and synthesis part figure 2.1 can be re–arranged, resulting in figure 2.7 (omitting the intermediate processing for a while). It is observed that for the analysis part the analysis polyphase matrix $\mathbf{H}(z)$ is used whereas at the synthesis side $\mathbf{J}\mathbf{G}^T(z)$ is found. $\mathbf{J}$ is the exchange matrix with ones along its main anti–diagonal and zeros elsewhere. Remark that matrix $\mathbf{J}$ needs to be inserted as the ordering of the delay–and–sum line in figure 2.7 was changed w.r.t. figure 2.6.

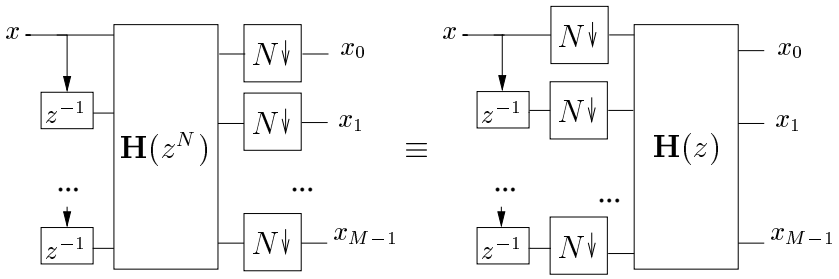


Figure 2.5: Analysis filter bank : polyphase decomposition

### 2.2.4 Perfect reconstruction

Ideally, in the absence of intermediate processing, the overall effect of a subband system is that of a pure delay, i.e. output $y$ is a delayed copy of the input $x$ (see figure 2.1). A subband system for which this property holds is called a perfect reconstruction system.

It can be observed that if

$$\mathbf{J}\mathbf{G}^T(z)\mathbf{H}(z) = \mathbf{I}_N \tag{2.20}$$

the inner filtering operations in figure 2.7 cancel each other out and the system reduces to a combined analysis and synthesis tapped delay line. It can be verified that this leads to perfect reconstruction [156].

The design of critically downsampled DFT or DCT modulated filter banks having the perfect reconstruction property is well–known and leads to the design of para–unitary or unimodular polyphase matrices [156] [170] [181]. Some references on perfect reconstruction design for oversampled DFT modulated filter banks can be found in more recent publications such as [22] [23] [24] [89] [168]. In chapter 3 design techniques for perfect reconstruction oversampled DFT modulated filter banks will be discussed.

In practice, small amplitude or phase distortion, aliasing and quantization related errors can be tolerated. In some applications the perfect reconstruction property can therefore be relaxed to "nearly" perfect reconstruction by admitting some amplitude, phase and aliasing distortion as long as the distortion is not noticeable or is acceptably low. In chapter 4 two design techniques for nearly–perfect reconstruction DFT modulated filter banks will be presented.

### 2.2.5 Overview of filter bank design techniques

In this section some common filter bank design techniques will be briefly addressed. A tutorial on filter bank design is [156]. More in–depth information can also be

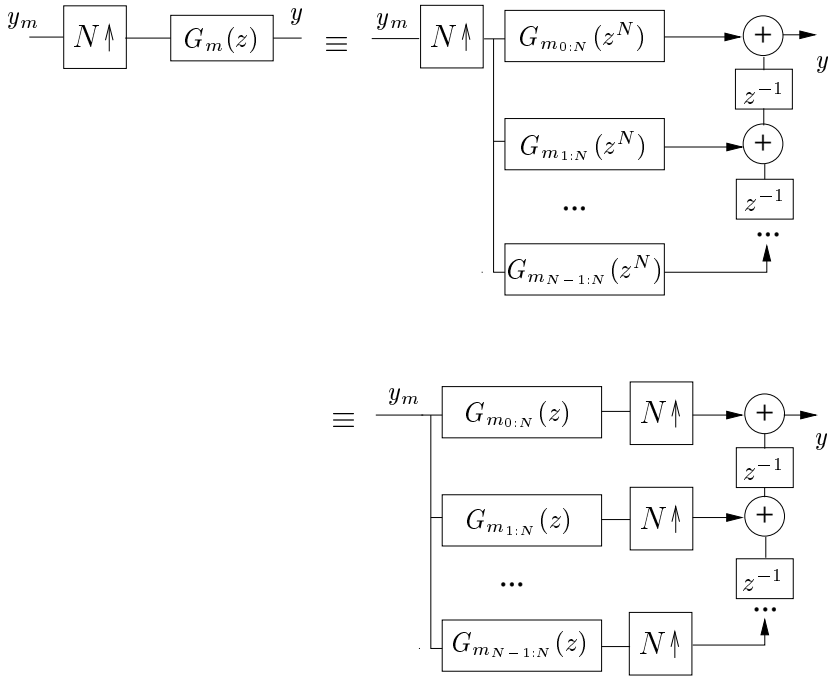


Figure 2.6: Synthesis filter polyphase decomposition

found in [156] as well as in many of the references given in this section.

Some design techniques lead to filter banks having the desired perfect reconstruction property discussed in section 2.2.4, implying amplitude, phase and aliasing distortion–free input–output behavior. However, in many applications imposing the perfect reconstruction property would be too restrictive. Hence, if perfect reconstruction behavior is not strictly required, some amplitude, phase and/or aliasing distortion can be tolerated. Several design procedures exist, which eliminate one or more of the artifacts (amplitude, phase and aliasing distortion) or reduce them to an acceptably low level. Apart from the kind and the amount of distortion tolerated other design criteria are considered, as there are frequency selectivity and implementation cost.

For almost 20 years a lot of attention has been paid to the design of aliasing–free and perfect reconstruction FIR digital filter banks. Initially, a maximally decimated, two–channel **Quadrature Mirror Filter bank (QMF)** was proposed [21] [156]. A simple relation between the analysis and synthesis filters ensures aliasing–free operation. Power symmetry[1] of the lowpass analysis filter and a special relation

[1] A filter $F(z)$ is called power symmetric if $\tilde{F}(z)F(z) + \tilde{F}(-z)F(-z) = 1$, $\tilde{F}(z) = F_*(z^{-1})$.

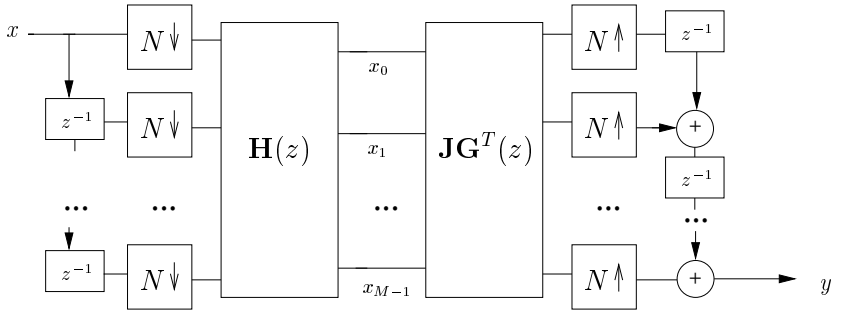


Figure 2.7: Subband system : polyphase implementation

between the lowpass and highpass analysis filter leads to perfect reconstruction. The extension to $M$–channel maximally decimated filter banks was done via polyphase decomposition of the analysis and synthesis filter bank [156].

Optimal computational savings are obtained for critically or maximally decimated subband schemes (see section 2.2.1). For maximally decimated FIR filter banks perfect reconstruction is easily obtained if the determinant of the analysis polyphase matrix $\mathbf{H}(z)$ takes on the form $\alpha z^{-\delta}$, with $\alpha \in \mathbb{R}$, $\alpha \neq 0$ and $\delta \in \mathbb{Z}$. In that case an FIR inverse for $\mathbf{H}(z)$ can be found by choosing

$$\mathbf{G}(z) = \mathbf{H}(z)^{-T}\mathbf{J} = \frac{z^{\delta}}{\alpha}\,(\text{adj}\ \mathbf{H}(z))^{T}\,\mathbf{J}. \tag{2.21}$$

Square *para–unitary* and *unimodular polyphase matrices*[2] have a determinant of the form $\alpha z^{-\delta}$ and are therefore used to design maximally decimated perfect reconstruction FIR filter banks [107] [111] [156] [157] [170] [181].

Contrary to DFT modulated filter banks, the filters of a **cosine (DCT) modulated filter bank** are restricted to be real, which is very appealing : roughly speaking, complex operations are 4 times more expensive than real operations. In this case $2M$ complex filters are combined to obtain a set of $M$ real cosine modulated filters (figure 2.3). The prototype filter $p[k]$ is supposed to be a good lowpass filter, blocking frequencies above $\frac{f_s}{2M}$, so only aliasing effects from neighboring bands have to be taken into account. An optimization procedure trying to reconcile good bandpass characteristics with a small amplitude distortion has been proposed and leads to so–called *pseudo–QMF cosine modulated filter banks* [156]. However, these filter banks do not have the perfect reconstruction property. By imposing para–unitarity on the analysis polyphase matrix *perfect reconstruction cosine modulated filter banks* can be designed [156]. The para–unitary constraint comes down to pairwisely imposing power complementarity on a set of prototype polyphase components. This technique is similar to the design procedure for para–unitary DFT

[2] See also section 2.1.3.

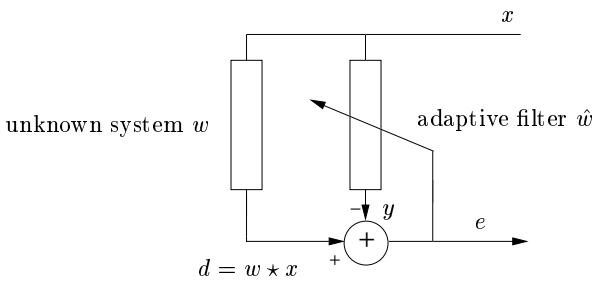


Figure 2.8: Adaptive filtering setup

modulated filter banks with 2–fold oversampling, which is discussed in appendix B.7. Finally, it is mentioned that computationally efficient and numerically stable implementations of DCT modulated filter banks are based on fast cosine transforms and lossless lattices.

Most oversampled subband schemes are based on **DFT modulated filter banks** for their aliasing robustness and ease of implementation (cf. section 2.2.2). The subsampling factor $N$ then typically varies between $\frac{M}{2}$ and $M$, with $M$ the number of subbands. Remark that overall aliasing cancellation does not exclude aliasing distortion in the subbands. In the case of subband adaptive filtering for instance, high aliasing levels will cause the convergence of the adaptive filter to degrade, which is not acceptable. However, it is possible to design DFT modulated filters with sufficiently low aliasing levels. Perfect reconstruction design for oversampled DFT modulated filter banks can be found in more recent publications such as [10] [22] [89] [168]. In chapter 3 design techniques for perfect reconstruction oversampled DFT modulated filter banks will be discussed.

## 2.3 Adaptive filtering techniques for speech enhancement

In figure 2.8 a typical adaptive filtering setup is presented. Input signal $x$ passes through an unknown system $w$ leading to the so–called desired signal $d$. Parallel to $w$ a filter $\hat{w}$ is updated such that $\hat{w}$ mimics the unknown system $w$. This can be done by minimizing the energy of the error output $e$. Hence, the adaptive filter consists of two major parts : a filtering operation and a weight update. If the adaptive filter models the unknown system $w$ perfectly, the error signal will be 0. The adaptation is done continuously through feedback of the error signal $e$. In this way $\hat{w}$ can track possible time–variations of $w$.

The unknown system $w$ is typically assumed to be linear and for many applications

this is a realistic assumption. In this thesis we will also assume that $w$ is an FIR system, or that $w$ can at least be described fairly well using an FIR model. As it is our intention to develop adaptive filtering techniques for audio and speech enhancement applications, the unknown system $w$ corresponds to an acoustic transfer function. Hence, $\hat{w}$ will be a high order FIR filter (see section 1.3.2) of many hundreds or even several thousands of filter taps (see also figure 1.4), that models $w$.

Different applications can be considered. In this thesis we will mainly concentrate on the echo cancellation problem (see figure 1.5), which fits in the adaptive filtering setup of figure 2.8. Typically in this kind of application a local signal source $s$ is added such that $d = s + w \star x$. It is assumed that $x$ and $d$ are known and that $w$ is estimated in order to reduce the error output $e$ and to retain only the signal of interest $s$. Both in the acoustic field as well as in digital communications adaptive filters are widely used to solve the echo cancellation problem. Another application is system modelling, in which not the error signal $e$ but the model $\hat{w}$ itself is of interest. Adaptive filtering techniques can be used to obtain reliable models of room acoustics for instance. More advanced adaptive filtering setups are used for interference and noise cancellation (see section 1.4.2) and equalization in digital communications e.g. where the output of the adaptive filter $y = \hat{w} \star x$ is of main interest.

Adaptive filtering algorithms are a powerful signal processing tool for signal enhancement and system modelling thanks to their self–learning capabilities. Adaptive filters can start from zero a–priori knowledge and by their inherent feedback structure and the continuous updating they autonomously form a model of the unknown system and can track possible system variations.

A large set of adaptive filtering techniques was developed in the literature during the last decades [81] [114] [178], all providing a solution for the different applications mentioned above. The algorithms differ in terms of performance (convergence, tracking, algorithmic delay), implementation cost (number of operations per second, memory requirements), stability (numerical stability, intrinsic stability) and implementability (algorithmic structure, modularity, parallelisability). In general it is not easy to decide which algorithm is optimal for a certain application, as it strongly depends on the available computational power and the designer's preference.

If the adaptive filter is used for audio processing or speech enhancement applications —as is the intention with the techniques discussed in this thesis— typically cheap algorithms are preferred. The complexity of the acoustic environment (see section 1.3.2) requires long adaptive filters to be updated and this in general leads to a high computational burden. Advanced adaptive filtering structures having a high complexity such as the RLS algorithm (see section 2.3.1) are therefore not often taken into consideration. For acoustic applications cheap algorithms such as the LMS and NLMS algorithm (see section 2.3.1), frequency–domain techniques (see section 2.3.2, chapter 6 and part III) and subband algorithms (see section 2.3.2 and part II) are commonly used. Thanks to the rapidly growing performance of silicon

based devices more advanced structures are being considered nowadays such as the APA algorithm and fast versions thereof as well as fast implementations of the RLS algorithm, both briefly discussed in section 2.3.1.

### 2.3.1 Standard adaptive filtering techniques

In this section and in section 2.3.2 some well–known adaptive filtering techniques will be discussed and references will be given to the literature, mainly serving as background information and as a starting point for the forthcoming chapters.

#### Least Mean Square adaptive filter

The most popular adaptive filtering algorithm for speech enhancement during the last decades is certainly the Least Mean Square (LMS) adaptive filter [81] [83] [114] [115] [178]. Although first proposed in 1960 by Widrow and Hoff it still serves as a standard and is used to benchmark more advanced algorithms nowadays.

The update equations defining the LMS algorithm are :

$$y[k] = \mathbf{x}_k^T \mathbf{w}_k \tag{2.22}$$

$$e[k] = d[k] - y[k] \tag{2.23}$$

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu \mathbf{x}_k^* e[k], \tag{2.24}$$

with $\mathbf{x}_k$ and $\mathbf{w}_k$ (complex–valued) column vectors of length $L_{FB}$ representing the input data

$$\mathbf{x}_k = \begin{bmatrix} x[k] \\ \vdots \\ x[k - L_{FB} + 1] \end{bmatrix} \updownarrow L_{FB} \tag{2.25}$$

and the adaptive filter weights at time $k$

$$\mathbf{w}_k = \begin{bmatrix} \hat{w}^{(k)}[0] \\ \vdots \\ \hat{w}^{(k)}[L_{FB} - 1] \end{bmatrix} \updownarrow L_{FB} \tag{2.26}$$

respectively[3] and $\mu$ a stepsize controlling the adaptation speed.

Like many adaptive filtering algorithms the LMS adaptive filter minimizes the cost function $\mathcal{E}\{|e[k]|^2\}$, using instantaneous estimates of the gradient vector corresponding to $\mathcal{E}\{|e[k]|^2\}$. In this way LMS approximates the well–known steepest–descent algorithm. It can indeed be proven that the expected convergence behavior of the LMS adaptive filter is that of the steepest–descent algorithm.

[3] Check figure 2.8 for signal conventions.

The time–evolution of the filter weights of the steepest–descent algorithm is governed by $(\mathbf{I} - \mu\mathbf{\Lambda_{xx}})^k$, where $k$ is the discrete–time index. Matrix $\mathbf{\Lambda_{xx}} = \text{diag}\{\lambda_i\}$ contains the eigenvalues $\lambda_i \in \mathbb{R}^+$ of the input autocorrelation matrix, which is defined as $\mathbf{R_{xx}} = \mathcal{E}\{\mathbf{x}_k^*\mathbf{x}_k^T\}$. It follows that for convergence $0 < \mu < \frac{2}{\lambda_{\max}}$. The larger $\mu$, the faster the convergence, but the larger the so–called excess mean–squared error : even when the algorithm has converged the filter coefficients will continue to bounce around. If $\mu$ is too large the algorithm becomes unstable. Furthermore, for input signals $x$ with a large eigenvalue spread (i.e. $\lambda_{\min} \ll \lambda_{\max}$), such as speech and audio signals, the convergence of the LMS algorithm is expected to be slow. This is a main disadvantage of the LMS adaptive filter. In practice, the eigenvalue decomposition of $\mathbf{R_{xx}}$ is hard to compute. Hence, a more conservative criterion

$$0 < \mu < \frac{2}{L_{FB}\ \mathcal{E}\{|x[k]|^2\}} \tag{2.27}$$

is used to determine the adaptation stepsize. For some applications a time–dependent stepsize $\mu_k$ is preferred.

Although very simple the LMS algorithm works remarkably well. Its biggest advantage is the simplicity and hence low implementation cost. The LMS updating (Eqs. 2.22–2.24) amounts to $16L_{FB} + 2$ operations if complex signals are involved and $4L_{FB} + 1$ operations for real signals[4].

### Normalized LMS adaptive filter

From Eq. 2.27 it can be observed that the upper bound for the LMS stepsize constant is inversely proportional to the input signal energy. For signals with a high dynamic range, such as speech for instance, the instantaneous signal energy continuously varies. If a fixed stepsize is applied, $\mu$ has to be taken smaller than $\frac{2}{L_{FB}\ E_{\max}}$, with $E_{\max}$ the maximum input signal energy. However, for this value of $\mu$ the filter will converge too slowly during low input data levels. The Normalized LMS (NLMS) algorithm was therefore defined as an extension to the LMS algorithm, having a stepsize that is inversely proportional to the actual input signal energy. Hence, the weight update equation (compare with Eq. 2.24) becomes

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu\ \frac{1}{\mathbf{x}_k^H\mathbf{x}_k + \delta}\ \mathbf{x}_k^* e[k], \tag{2.28}$$

in which $L_{FB}\,\mathcal{E}\{|x[k]|^2\}$ was approximated by $\mathbf{x}_k^H\mathbf{x}_k + \delta$, with $\delta \in \mathbb{R}^+$ a small constant preventing overflow if $\mathcal{E}\{|x[k]|^2\} \approx 0$. It can be proven that the NLMS algorithm is convergent in the mean square, i.e. $\mathcal{E}\{|x[k]|^2\}$ converges to a finite steady–state value, if (compare with Eq. 2.27)

$$0 < \mu < 2. \tag{2.29}$$

[4] The cost is expressed in number of equivalent real operations : a real multiplication and a real addition are assumed to be equally expensive. More information on cost computation can be found in section 2.4.

For speech and audio applications this normalized version of the LMS algorithm is typically used. In order to reduce the effect of the excess mean–squared error $\mu$ is typically chosen around 0.2.

Compared to the LMS algorithm, $\mu$ has to be divided by the energy of the input data vector. If the energy is computed using updating techniques (see Eqs. 10.1 and 10.3), the extra cost w.r.t. LMS is rather small.

### Recursive Least Squares adaptive filter

It can easily be proven that the weight vector that minimizes the least squares criterion

$$\min_{\mathbf{w}_k} \sum_{l=0}^{L-1} |e[k-l]|^2 = \min_{\mathbf{w}_k} ||\mathbf{d}_k - \mathbf{X}_k^T \mathbf{w}_k||^2 \tag{2.30}$$

with

$$\mathbf{d}_k = \begin{bmatrix} d[k-L+1] \\ \vdots \\ d[k] \end{bmatrix} \begin{matrix} \uparrow \\ \\ \downarrow \end{matrix} L \tag{2.31}$$

and

$$\mathbf{X}_k = \begin{bmatrix} x[k-L+1] & \dots & x[k] \\ \vdots & \ddots & \vdots \\ x[k-L-L_{FB}+2] & \dots & x[k-L_{FB}+1] \end{bmatrix} \begin{matrix} \uparrow \\ \\ \downarrow \end{matrix} L_{FB} \tag{2.32}$$

is given by[5]

$$\mathbf{w}_k = \mathbf{R}_k^{-1} \mathbf{X}_k^* \mathbf{d}_k \tag{2.33}$$

if $L \geqslant L_{FB}$. Matrix $\mathbf{R}_k$ is defined as $\mathbf{X}_k^* \mathbf{X}_k^T$. Due to the matrix inverse in Eq. 2.33 the computational complexity of this least–squares algorithm is a function of $L_{FB}^3$.

The complexity can be reduced by recursively updating $\mathbf{R}_k^{-1}$ starting from $\mathbf{R}_{k-1}^{-1}$. This leads to the so–called Recursive Least Squares (RLS) algorithm, having a weight update of the form

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mathbf{k}_k e[k], \tag{2.34}$$

in which $\mathbf{k}_k$ is the so–called Kalman gain. The complexity of the RLS adaptive filter depends on $L_{FB}^2$. Different versions exist, such as the sliding–window RLS or the exponentially weighted RLS algorithm [114].

Unfortunately, the RLS algorithm suffers from numerical problems. Numerically stable implementations of the RLS algorithm do exist and rely on the QR–decomposition [81] [114].

[5] For this the gradient of $||\mathbf{d}_k - \mathbf{X}_k^T \mathbf{w}_k||^2$ w.r.t. $\mathbf{w}_k$ has to be computed and set to $\mathbf{0}$. Information about how to compute the gradient of a real scalar w.r.t. a complex vector can be found in [81] and on page 265.

Owing to the fact that the filter weights are updated using $\mathbf{R}_k^{-1}$, RLS is definitely superior to LMS as far as convergence is concerned. Unlike the LMS algorithm RLS does not suffer from slow convergence for signals having an ill–conditioned autocorrelation matrix $\mathbf{R_{xx}} = \mathcal{E}\{\mathbf{x}_k^* \mathbf{x}_k^T\}$.

The complexity of the RLS algorithm depends quadratically on the filter length and therefore RLS is not taken into consideration for applications were long filters need to be adapted (e.g. speech enhancement).

Many fast versions can be derived for the RLS adaptive filter [81] [114]. They have a complexity that linearly depends on the filter length, therefore being much cheaper than the standard RLS algorithm if long filters are adapted. Still, the cost is considerably higher than that of the LMS algorithm : the Fast Transversal Filters algorithm e.g., one of the cheapest fast RLS variants, is still 4 times more expensive than LMS.

The use of fast RLS adaptive filters in the context of acoustic echo cancellation has already been reported in the literature [12] [58]. With the continuous increase in hardware performance it is expected that fast RLS techniques will gain importance in the field of speech enhancement in the coming years.

### Affine Projection Algorithm

An extension to the NLMS algorithm is the Affine Projection Algorithm (APA) [69] [116] :

$$\mathbf{e}_k = \mathbf{d}_k - \mathbf{X}_k^T \hat{\mathbf{w}}_k \tag{2.35}$$

$$\hat{\mathbf{w}}_{k+1} = \hat{\mathbf{w}}_k + \mu_k \mathbf{X}_k^* \left( \mathbf{X}_k^T \mathbf{X}_k^* \right)^{-1} \mathbf{e}_k . \tag{2.36}$$

$\mathbf{X}_k$ is an $L_{FB} \times L$ Toeplitz matrix containing the input data samples, defined by Eq. 2.32 and $\mathbf{d}_k$ is shown in Eq. 2.31. The $L_{FB} \times 1$ vector $\hat{\mathbf{w}}_k$ contains the adaptive filter weights and $\mu_k$ is the stepsize. Equation 2.36 is valid for $L \leqslant L_{FB}$.

Typically, $\mu_k$ is chosen equal to 1. In this case APA solves the underdetermined set of equations

$$\mathbf{X}_k^T \hat{\mathbf{w}}_k = \mathbf{d}_k \tag{2.37}$$

such that the a–posteriori error $\mathbf{e}_k^{\text{post}} = \mathbf{d}_k - \mathbf{X}_k^T \hat{\mathbf{w}}_{k+1}$ is $\mathbf{0}$, while minimizing the energy of the a–priori error output $\mathbf{e}_k$.

If $L = 1$, APA reduces to the NLMS algorithm. Hence, APA is scalable between the NLMS algorithm ($L = 1$) and the sliding–window RLS algorithm ($L = L_{FB}$). Very often a regularization term $\delta \mathbf{I}_L$, with $\delta$ a small positive constant, is added to $\mathbf{X}_k^T \mathbf{X}_k^*$ to protect the algorithm against overflow. Moreover in this case, Eq. 2.36 is also valid for $L > L_{FB}$.

Fast versions of the APA algorithm can be derived [69] [137], leading to a considerable complexity reduction.

### 2.3.2 Block–based techniques

In this section block–based adaptive filtering techniques will be discussed. The data is now processed in blocks, i.e. the algorithm gets its input data from input buffers as soon as they are full, and dumps a set of processed samples to the output buffer. Through the use of downsampling techniques and fast signal transforms such as the FFT, a considerable cost reduction can be obtained w.r.t. sample–based adaptive filters.

A disadvantage of block processing is the delay introduced by the algorithm, as the error signal is sent to the output only after a complete block of samples has been processed. On the other hand, in some applications such as echo cancellation for instance, some small delay is desirable as "looking ahead" leads to a more robust signal/noise and double–talk detection (see chapter 10), essential for good real–time operation. Furthermore, the usage of input and output buffers is the natural way to feed and extract data to and from the algorithmic core during a real–time operation on a microprocessor or DSP.

Contrary to the previous sample–based adaptive algorithms that are characterized by the discrete–time variable $k$, we will use the block index $n$ to indicate time in this section. By convention in this thesis block $n$ processes a set of $L$ data samples corresponding to time indices $nL+1$ to $(n+1)L$.

#### Block–LMS

Block–LMS (BLMS) is similar to standard LMS. The filter weights however are updated only at regular periods, i.e. each time a block, being a complete set of $L$ new input samples, has arrived. The equations defining the BLMS algorithm for block $n$ are :

$$\mathbf{X}_n = \begin{bmatrix} x[nL+1] & \dots & x[(n+1)L] \\ \vdots & \ddots & \vdots \\ x[nL-L_{FB}+2] & \dots & x[(n+1)L-L_{FB}+1] \end{bmatrix} \begin{matrix} \uparrow \\ \\ \downarrow \end{matrix} L_{FB} \tag{2.38}$$

$$\mathbf{d}_n = \begin{bmatrix} d[nL+1] \\ \vdots \\ d[(n+1)L] \end{bmatrix} \begin{matrix} \uparrow \\ \\ \downarrow \end{matrix} L \tag{2.39}$$

$$\mathbf{e}_n = \mathbf{d}_n - \mathbf{X}_n^T \mathbf{w}_n \tag{2.40}$$

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu_n \mathbf{X}_n^* \mathbf{e}_n. \tag{2.41}$$

Each time a BLMS iteration is performed, as described by Eqs. 2.38–2.41, $L$ new $x$–samples are taken in, and $L$ new filter output samples $e$ are produced. Vector $\mathbf{d}_n$

contains the $L$ most recent values of the desired signal $d$. For signal conventions, see figure 2.8. Further, $n$ is the block time index and $\mu_n$ is a block dependent stepsize controlling the adaptation speed. $L$ is called the block length, and hence the corresponding input/output delay of the algorithm is $2L-1$. Observe that the BLMS algorithm reduces to LMS if $L=1$.

Equation 2.41 can also be written as

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu_n \sum_{l=1}^{L} \mathbf{x}_{nL+l}\, e[nL+l], \tag{2.42}$$

in which vector $\mathbf{x}_{nL+l}$ is an $L_{FB} \times 1$ vector defined as

$$\mathbf{x}_{nL+l} = \begin{bmatrix} x[nL+l] \\ \vdots \\ x[nL+l-L_{FB}+1] \end{bmatrix} \updownarrow L_{FB}. \tag{2.43}$$

The gradient estimates of the BLMS algorithm are expected to be better than those of LMS thanks to the averaging in Eq. 2.42. They become more accurate as $L$ becomes larger. However, the larger $L$ the less frequently the weights are updated, which has a negative effect on the tracking capabilities of the algorithm.

For wide–sense stationary processes the time constants, misadjustment and final filter weights of the BLMS algorithm are identical to those of the LMS adaptive filter, if both are adapted with the same stepsize $\mu$. In practice, the convergence behavior of the BLMS algorithm is typically comparable to that of LMS or is slightly worse. However, the maximum stepsize $\mu_{\max}$ ensuring stability reduces proportionally with the block size $L$ [19], which makes the BLMS adaptive filter less favorable than LMS if fast convergence is preferred.

### Frequency–Domain Adaptive Filter

As a cheaper alternative to the LMS algorithm, the frequency–domain adaptive filter (FDAF) was introduced, which is a direct translation of BLMS to the frequency domain [64] [101] [142].

The core and hence most expensive operations of the BLMS algorithm are a linear convolution, i.e. the filtering operation (see Eq. 2.40) and a correlation, being the weight updating (see Eq. 2.41). However, as they are computed in the time domain their implementation is suboptimal.

It appears that these core operations can be implemented in a more efficient way by replacing the linear time–domain convolution and correlation by circular convolutions/correlations in the frequency domain. This requires some "restore" operations, which basically cope with the error that is introduced when circular instead of linear

operations are used. These "restore" operations are typically based on overlap–save or overlap–add techniques [20] [142].

Overlap–add and overlap–save correction techniques are equivalent as far as computational complexity and performance is concerned [142]. The algorithmic equations are slightly different however. As no further insight or additional performance may be expected from overlap–add techniques, overlap–save correction will be consistently applied throughout the following chapters.

The adaptive filter

$$\mathbf{w}_n = \begin{bmatrix} \hat{w}^{(n)}[0] \\ \vdots \\ \hat{w}^{(n)}[L_{FB}-1] \end{bmatrix} \tag{2.44}$$

is transformed to the frequency domain as follows :

$$\underline{\mathbf{w}}_n = \mathbf{F} \begin{bmatrix} \mathbf{w}_n \\ \mathbf{0} \end{bmatrix} \begin{matrix} \updownarrow L_{FB} \\ \updownarrow M-L_{FB} \end{matrix} . \tag{2.45}$$

$\mathbf{F}$ is the $M \times M$ DFT matrix : element $(p,q)$ of $\mathbf{F}$ is given by $\mathbf{F}(p,q) = e^{-j\frac{2\pi pq}{M}}$.

The overlap–save FDAF essentially comes down to the following procedure :

$$\underline{\mathbf{X}}^{(n)} = \text{diag}\left\{ \mathbf{F} \begin{bmatrix} x[(n+1)L-M+1] \\ \vdots \\ x[(n+1)L] \end{bmatrix} \right\} \Bigg\uparrow M, \tag{2.46}$$

$$\mathbf{y}^{(n)} = \begin{bmatrix} \mathbf{0}_{M-L} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \underline{\mathbf{X}}^{(n)} \underline{\mathbf{w}}_n \tag{2.47}$$

$$\mathbf{d}^{(n)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_n \end{bmatrix} \begin{matrix} \updownarrow M-L \\ \updownarrow L \end{matrix}, \qquad \mathbf{d}_n = \begin{bmatrix} d[nL+1] \\ \vdots \\ d[(n+1)L] \end{bmatrix} \Bigg\uparrow L \tag{2.48}$$

$$\mathbf{e}^{(n)} = \mathbf{d}^{(n)} - \mathbf{y}^{(n)} \tag{2.49}$$

$$\underline{\mathbf{w}}_{n+1} = \underline{\mathbf{w}}_n + \mathbf{F} \begin{bmatrix} \mathbf{I}_{L_{FB}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{M-L_{FB}} \end{bmatrix} \mathbf{F}^{-1} \boldsymbol{\Delta} \underline{\mathbf{X}}^{(n)*} \mathbf{F} \mathbf{e}^{(n)}. \tag{2.50}$$

In each iteration, $L$ new $x$–samples are taken in, and $L$ new filter output samples $e$ are produced. Vector $\mathbf{d}_n$ has length $L$ and contains the $L$ most recent values of the desired signal $d$. Equation 2.47 and 2.50 perform the filtered input signal correction and the gradient estimate correction respectively. These are the essential steps that transform the circular convolution and correlation into linear operations.

$L$ is called the block length, and hence the corresponding input/output delay of the FDAF algorithm is $2L-1$. A necessary condition for appropriate operation of the FDAF is that $M = L_{FB} + L - 1 + \sigma$, with $\sigma$ an integer such that $\sigma \geqslant 0$. In most applications $L_{FB} = L$, $\sigma = 1$ and $M$ is a power of 2.

The standard version of the algorithm, for which the stepsize matrix $\mathbf{\Delta} = \mu \mathbf{I}_M$, corresponds to an exact translation of Block–LMS to the frequency domain. In the most general case however, $\mathbf{\Delta} = \text{diag}\{\boldsymbol{\mu}_s^{(n)}\}$ contains frequency–dependent stepsizes that can vary from block to block. The stepsizes can be optimally tuned to the actual signal energy in each frequency band. Hence, improved convergence can be obtained.

There exist two variants of the algorithm, called the *constrained* and the *unconstrained FDAF*. The algorithm described by Eqs. 2.46–2.50 is the constrained FDAF. For the unconstrained version the "projection" type operation

$$\mathbf{F} \begin{bmatrix} \mathbf{I}_{L_{FB}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{M-L_{FB}} \end{bmatrix} \mathbf{F}^{-1} \tag{2.51}$$

is left out of Eq. 2.50. The unconstrained updating requires 3 (I)DFTs per iteration. The constrained FDAF is more expensive with 5 (I)DFTs per iteration. The constrained and the unconstrained algorithm both converge to the same Wiener solution under certain conditions. The maximum allowable stepsize for the unconstrained algorithm still guaranteeing stability, is reduced by a factor 2 with respect to the constrained case. Its convergence speed therefore is also approximately 2 times slower [95].

The implementation cost of this frequency–domain algorithm is typically considerably lower than that of its time–domain equivalent. A detailed cost analysis and discussion can be found in section 6.4. It appears e.g. that the FDAF algorithm is computationally attractive only if the block length $L$ has the same order of magnitude as the filter length $L_{FB}$. In practice however, this leads to unacceptable input/output delays : in a realistic acoustic echo cancellation setup for instance, with an adaptive filter $\hat{w}$ modelling 1000 taps in the time domain and sampled at 8 kHz, the delay is twice as long as the acoustic impulse response, i.e. 250 ms.

In general, one can state that frequency–domain algorithms such as the FDAF have mainly two advantages over time–domain techniques. First, a considerable cost reduction can be obtained thanks to block processing and the use of fast signal transforms. Secondly, better convergence properties are expected through the decorrelation capabilities of the signal transforms and hence the use of optimally tuned stepsizes per frequency band. The main disadvantage is the larger algorithmic delay.

### Subband adaptive filtering

The combination of the adaptive filtering techniques discussed in section 2.3.1 with the filter banks of section 2.2 gives rise to a promising block based adaptive system, known as the subband adaptive filter.

A general subband adaptive filtering setup is shown in figure 2.9. The input signals $x$ and $d$ are fed into identical $M$–band analysis filter banks. After subsampling by

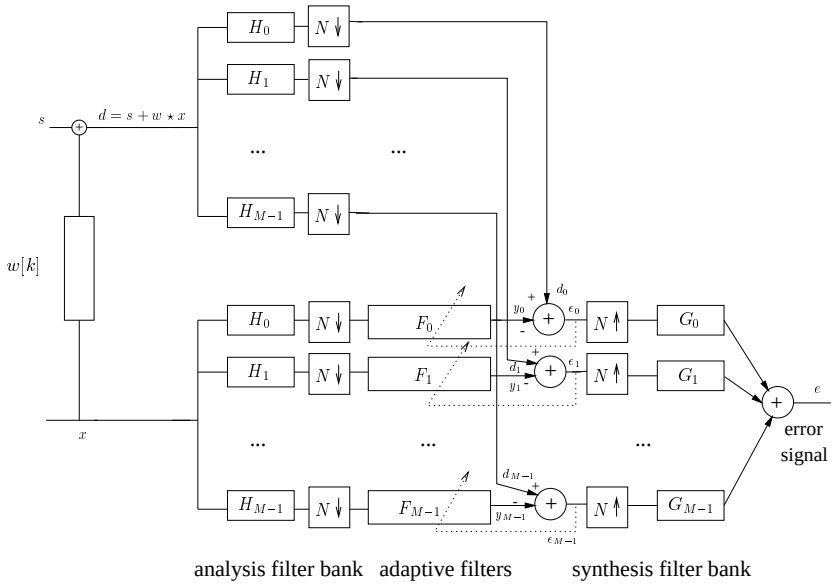


Figure 2.9: General subband adaptive filtering setup

a factor $N$, (typically LMS–based) adaptive filtering is done in each subband. The outputs of the subband adaptive filters are recombined in the synthesis filter bank, leading to the final output $e$. Due to aliasing effects, this setup will only work for $M \geqslant N$.

By splitting signals into subbands and subsampling better convergence and tracking properties are hoped for. As the adaptive computations as well as the filter bank operations can be done at the lower sampling rate, the subband approach is expected to give a better performance at a lower cost.

Subband adaptive filtering will be the key topic of part II of this thesis. More detailed information, performance as well as cost estimates and references to the literature can be found in chapter 5.

### Partial Rank Algorithm

The Partial Rank Algorithm (PRA) [69] is a block version of the Affine Projection Algorithm, which was defined in section 2.3.1. Hence, the PRA can be interpreted as a block–normalized version of the BLMS algorithm. Contrary to BLMS, which is a block version of the LMS algorithm with an improved gradient estimate, the performance of PRA is degraded w.r.t. the Affine Projection Algorithm, of which it is a mere block version.

For each block of $L$ samples the PRA performs the following steps :

$$\mathbf{X}_n = \begin{bmatrix} x[nL+1] & \dots & x[(n+1)L] \\ \vdots & \ddots & \vdots \\ x[nL-L_{FB}+2] & \dots & x[(n+1)L-L_{FB}+1] \end{bmatrix} \begin{matrix} \uparrow \\ \\ \downarrow \end{matrix} L_{FB} \tag{2.52}$$

$$\mathbf{d}_n = \begin{bmatrix} d[nL+1] \\ \vdots \\ d[(n+1)L] \end{bmatrix} \begin{matrix} \uparrow \\ \\ \downarrow \end{matrix} L \tag{2.53}$$

$$\mathbf{e}_n = \mathbf{d}_n - \mathbf{X}_n^T \hat{\mathbf{w}}_n \tag{2.54}$$

$$\hat{\mathbf{w}}_{n+1} = \hat{\mathbf{w}}_n + \mu_n \mathbf{X}_n^* \left( \mathbf{X}_n^T \mathbf{X}_n^* \right)^{-1} \mathbf{e}_n. \tag{2.55}$$

$\mathbf{X}_n$ is an $L_{FB} \times L$ Toeplitz matrix containing the $x$–samples. The $L_{FB} \times 1$ vector $\hat{\mathbf{w}}_n$ contains the adaptive filter weights and models the unknown system. Further, $\mu_n$ is a block dependent stepsize.

If $L = 1$ the PRA reduces to the NLMS algorithm. Hence, the PRA is scalable and lies in between the NLMS algorithm ($L = 1$) and a block adaptive least squares estimate of the unknown system ($L = L_{FB}$). Very often a regularization term $\delta \mathbf{I}_L$, with $\delta$ a small positive constant, is added to $\mathbf{X}_n^T \mathbf{X}_n^*$ to prevent the algorithm from overflowing.

More details on PRA and a cost analysis are given in section 9.2.3 and table 9.7.

## 2.4 Computational cost

A cost analysis will be performed for most of the algorithms that are discussed in this thesis. In this section it is pointed out what kind of assumptions are made for a series of elementary arithmetic operations. The cost or execution time associated with other operations such as memory accesses or data copying is not taken into account.

The cost will be expressed in *number of equivalent real operations* (`op.`) or *number of equivalent real operations per second* (`ops.`). An *equivalent real operation* is defined as the cheapest arithmetic operation on a general purpose floating point processor, which typically corresponds to an addition of 2 real–valued variables. Obviously, the implementation cost and corresponding execution time of a signal processing algorithm will strongly depend on the software (programmed in C or optimized assembler code, single or double precision floating point) and on the hardware (ALU architecture, dedicated signal processing functionality (e.g. DSP or ASIC), memory structure and access time, fixed point vs. floating point). In this thesis we try to predict the expected execution times under the assumption that the

algorithm is executed on a general purpose floating point CPU (e.g. a PC–based system).

The presented cost formulas give an idea about the dependence of the total cost on each of the algorithmic parameters and the weight of each of the terms in the cost formula. Furthermore, in this way computational bottlenecks can easily be detected and it can be determined how the cost should be further reduced if necessary.

It is assumed that a real multiplication and a real addition have the same complexity. A complex multiplication amounts to 6 equivalent real operations and a complex addition to 2 equivalent real operations. An elementary filter or multiply–accumulate operation (MAC) can be efficiently computed on dedicated processors such as DSPs and corresponds to 1 machine cycle on this type of CPU. Dedicated processors reach their optimal speed of 1 cycle per elementary instruction only if the data can be stored in specific fast registers or on–chip memory. Fast memory has a limited storage capacity however. Hence, a considerable amount of machine cycles are typically lost by data storage, initialization, subroutine calls and pipeline stalls. For a MAC, basically being a combined addition and multiplication, 2 equivalent real operations are counted, as the cost analysis is primarily intended for general purpose floating point CPUs. A total of 8 equivalent real operations are counted if complex signals are involved.

The equivalent cost for more advanced operations such as an inversion, division, sine, logarithm, ... seems to depend significantly on the type of processor, the software coding, the compiler and data buffer sizes. Furthermore, the arithmetic operations as such are hard to measure as they always involve data fetching and pointer in/decrements. Nevertheless, an attempt was made to obtain an indicative cost estimate.

**Experiment 2.1** Thereto some speed tests were performed on two general purpose processors, making realistic assumptions for the kind of application envisaged (speech enhancement). The results are presented in table 2.1. The software for the experiment was coded in C using double–precision floating point arithmetic. Positive reals were applied to the square–root and the logarithm. The test was run on random signal buffers of length 256 in 200000 consecutive loops for averaging. Observe from table 2.1 that the cost for the addition was taken as a reference. The Sun workstation appeared to be approximately 4 times slower than the PC in this experiment. ▽

For the computation of FFTs and IFFTs efficient numerical schemes such as the Radix Split algorithm can be used. In the case of real, complex conjugated or zero–padded inputs the algorithmic scheme can be simplified resulting in a lower implementation cost than that of the standard algorithm for complex–valued inputs. A short discussion on this can be found in [28]. The cost of an $M$–point real–input FFT is assumed to be $2M \log_2 M - 4M + 6$ operations [100]. For an $M$–point

| | Sun UltraSparc–2 @ 200 MHz | Pentium III–Xeon @ 933 MHz |
|---|---|---|
| $-a$ | 1.0 | 0.85 |
| $a+b$ | 1 | 1 |
| $a-b$ | 1.0 | 1.0 |
| $a*b$ | 1.0 | 1.0 |
| $1/a$ | 6.1 | 8.1 |
| $a/b$ | 6.3 | 8.1 |
| $\sqrt{a}$ | 6.3 | 14.5 |
| $\sin a$ | 41.9 | 21.2 |
| $\cos a$ | 39.8 | 20.2 |
| $\tan a$ | 48.0 | 27.7 |
| $\log a$ | 26.1 | 20.9 |
| $e^a$ | 31.3 | 31.6 |

Table 2.1: Comparative speed test on two general purpose processors : the relative execution times for a set of common arithmetic operations were computed

real–output IFFT $2M \log_2 M$ operations are counted.

## 2.5 Conclusions

In this chapter the necessary background information was provided to make the forthcoming chapters of the thesis more accessible to the reader.

In section 2.1 the very basics of signal processing and multirate systems were discussed.

Section 2.2 discusses digital filter banks. Different concepts were defined and an overview of existing filter bank design methods was given.

A brief survey of adaptive filtering techniques was presented in section 2.3. A set of algorithms that relate with the adaptive filters presented further on in the thesis were discussed in more detail.

In section 2.4 some remarks on cost and complexity estimation were presented.

# Part I

# DFT Modulated Filter Bank Design for Oversampled Subband Systems

In the introductory chapter it was motivated that frequency–dependent adaptive filtering techniques such as the subband adaptive filter are required for adequate acoustic echo suppression. Subband adaptive filters combine existing adaptive filtering techniques with digital filter bank schemes and multirate signal processing. Filter banks however, introduce considerable signal and aliasing distortion. In this part design techniques for perfect and nearly perfect reconstruction oversampled DFT modulated filter banks will be presented. These filter banks introduce no or almost no signal distortion and are easily integrated in subband adaptive filtering structures.

In **chapter 3** we discuss some properties of oversampled filter banks and derive a perfect reconstruction condition. In section 3.3 a design technique for perfect reconstruction oversampled DFT modulated filter banks is considered, which was presented in [22]. With this method however the order of the filter banks cannot be finely adjusted. We present an extension to this method, which basically allows to choose any desired filter length.

In **chapter 4** two design techniques for nearly perfect reconstruction oversampled DFT modulated filter banks are presented. A first technique is based on frequency–domain optimization, the other is a mixed time/frequency–domain based method. Finally, subband adaptive filtering is taken as an example to illustrate that thanks to their lower stopband level nearly perfect reconstruction filter banks outperform perfect reconstruction systems.

# Chapter 3

# Perfect Reconstruction Oversampled DFT Modulated Filter Bank Design

Filter banks are widely used in digital signal processing systems. They are often integrated in multirate schemes to reduce the implementation cost and to improve algorithmic performance. DFT modulated filter banks are commonly used to design oversampled subband schemes. In contrast to critically subsampled systems oversampled subband schemes trade off between complexity gain and aliasing distortion. Small aliasing levels are required for the adequate performance of the intermediate subband operations such as adaptive filtering.

Ideally, the overall effect of a subband system is that of a pure delay, i.e. the subband system has the perfect reconstruction property. In the absence of intermediate subband operations perfect reconstruction guarantees distortionless input/output behavior : the output of the subband system is an exact, delayed copy of the input.

In this chapter design techniques for perfect reconstruction oversampled DFT modulated filter banks are presented. In section 3.1 oversampled DFT modulated subband systems are discussed. It is shown, both for the analysis and the synthesis part, how the filter bank polyphase matrix can be decomposed in a structured polynomial matrix and a DFT operation.

In section 3.2 the perfect reconstruction property is analyzed in more detail. The perfect reconstruction requirement is translated into a mathematical definition for

the synthesis polynomial matrix $\mathbf{C}(z)$ and it is pointed out under which conditions a perfect reconstruction DFT modulated subband system can be obtained.

In section 3.3 a para–unitary filter bank design method is discussed, which was presented in [22]. Designing a para–unitary structured polynomial matrix $\mathbf{B}(z)$ is equivalent to imposing para–unitarity on a set of dense polynomial matrices, which can be parameterized using para–unitary lattices. It is shown how the parameterization can be optimized to obtain a perfect reconstruction subband system with good bandpass characteristics. With the method presented in [22] however the order of the filter banks cannot be adjusted accurately. We present an extension to this method, which basically allows to choose any desired filter length. Further, we show that based on the inverse parametrization appropriate starting values can be obtained for the lattice parameters, which reduces the optimization time.

Section 3.4 formulates the conclusions.

## 3.1 Oversampled DFT modulated subband systems

In section 2.2 the basics of digital filter bank theory were presented. It was pointed out that digital filter banks can be implemented efficiently using polyphase decomposition. In this chapter we will concentrate on uniformly spaced FIR filter banks based on DFT modulation. Oversampled DFT modulated filter banks introduce only a small amount of subband aliasing, which is a necessary condition for the adequate performance of the intermediate operations such as subband adaptive filtering. Not only the overall performance of the algorithm can be improved if DFT modulation filter banks are employed, also the filter bank cost can be reduced significantly if polyphase decomposition and fast signal transforms are used, as will be illustrated in section 3.1.3.

### 3.1.1 DFT modulated analysis filter bank

In [22] a general framework for oversampled DFT modulated filter banks was proposed. The analysis polyphase matrix $\mathbf{H}(z)$ contains the polyphase components of all analysis filters (Eq. 2.17, figure 2.5). In the case of a DFT modulated filter bank $\mathbf{H}(z)$ can be factorized as

$$\mathbf{H}(z) = \mathbf{F}\mathbf{B}(z). \tag{3.1}$$

$\mathbf{F}$ is the $M \times M$ DFT matrix, i.e.

$$\mathbf{F}(p,q) = e^{-j\frac{2\pi pq}{M}}, \qquad 0 \leqslant p,q \in \mathbb{N} < M \tag{3.2}$$

and polynomial matrix $\mathbf{B}(z)$ characterizes the filter bank.

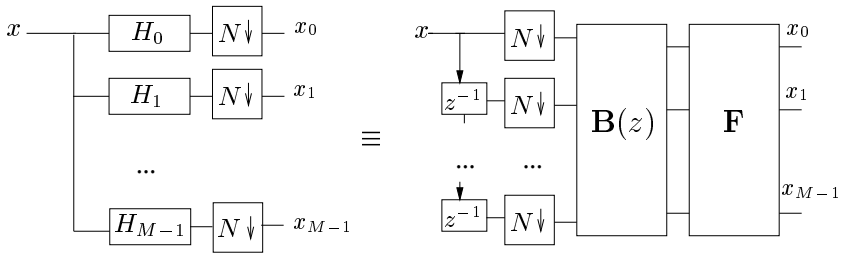


Figure 3.1: An $M$–band DFT modulated analysis filter bank followed by $N$–fold downsamplers. By employing FFT techniques ($\mathbf{F}$) and polyphase decomposition ($\mathbf{B}(z)$) an efficient implementation is obtained.

**Theorem 3.1** *$\boldsymbol{B}(z)$ is a structured $M \times N$ polynomial matrix, containing polyphase components of the analysis prototype filter $h_0$. Element $(m,n)$ of $\boldsymbol{B}(z)$ is given by :*

$$\begin{aligned} [\mathbf{B}(z)]_{m,n} &= z^{-q} H_{0_{(n+qN):K}}(z^Q) && \text{if } (m-n) \bmod g = 0 \\ &= 0 && \text{if } (m-n) \bmod g \neq 0 \end{aligned} \tag{3.3}$$

*in which $0 \leqslant m \in \mathbb{N} < M$ and $0 \leqslant n \in \mathbb{N} < N$. Further, $(n+qN)$ mod $M = m$, from which $q$ can be computed, $g = gcd(M,N)$, $Q = \frac{M}{g}$ and $K = \frac{MN}{g}$. $H_{0_{k:K}}(z)$ is the $k$–th out of $K$ polyphase components of the prototype filter $h_0[k]$.*

**Proof :** A proof can be found in [22] [24] and in appendix B.1.

Theorem 3.1 proves that an $M$–band DFT modulated analysis filter bank with $N$–fold downsampling can be implemented as a tapped delay line of size $N$ followed by a structured $M \times N$ polynomial matrix $\mathbf{B}(z)$, containing polyphase components of the prototype $h_0$, and an $M \times M$ DFT matrix $\mathbf{F}$ (see figure 3.1). The product $\mathbf{FB}(z)$ can be computed efficiently by means of a Fast Fourier Transform if $M = 2^r, r \in \mathbb{N}$. If $M$ is not a power of 2 less computationally attractive schemes have to be employed.

**Example 3.1** A 4–fold downsampled DFT modulated subband system with 6 subbands is considered. For this example

$$\begin{cases} M = 6 \\ N = 4 \\ g = \gcd(6,4) = 2 \\ K = \frac{6.4}{2} = 12 \\ Q = 3. \end{cases} \tag{3.4}$$

Let $H_0(z)$ be the lowpass prototype filter and $H_{0_{k:12}}(z)$ the polyphase components of $H_0(z)$. Following Eq. 3.3,

$$\mathbf{B}(z) = \begin{bmatrix} H_{0_{0:12}}(z^3) & 0 & z^{-1}H_{0_{6:12}}(z^3) & 0 \\ 0 & H_{0_{1:12}}(z^3) & 0 & z^{-1}H_{0_{7:12}}(z^3) \\ z^{-2}H_{0_{8:12}}(z^3) & 0 & H_{0_{2:12}}(z^3) & 0 \\ 0 & z^{-2}H_{0_{9:12}}(z^3) & 0 & H_{0_{3:12}}(z^3) \\ z^{-1}H_{0_{4:12}}(z^3) & 0 & z^{-2}H_{0_{10:12}}(z^3) & 0 \\ 0 & z^{-1}H_{0_{5:12}}(z^3) & 0 & z^{-2}H_{0_{11:12}}(z^3) \end{bmatrix}. \tag{3.5}$$

is obtained. Now, if $H_0(z)$ is e.g. defined as follows

$$h_0[k] = \begin{cases} k+1 & \text{if } 0 \leqslant k \leqslant 23 \\ 0 & \text{if } k < 0 \text{ or } k > 23, \end{cases} \tag{3.6}$$

then

$$\mathbf{B}(z) = \begin{bmatrix} 1+13z^{-3} & 0 & 7z^{-1}+19z^{-4} & 0 \\ 0 & 2+14z^{-3} & 0 & 8z^{-1}+20z^{-4} \\ 9z^{-2}+21z^{-5} & 0 & 3+15z^{-3} & 0 \\ 0 & 10z^{-2}+22z^{-5} & 0 & 4+16z^{-3} \\ 5z^{-1}+17z^{-4} & 0 & 11z^{-2}+23z^{-5} & 0 \\ 0 & 6z^{-1}+18z^{-4} & 0 & 12z^{-2}+24z^{-5} \end{bmatrix} \tag{3.7}$$

is found. △

Matrix $\mathbf{B}(z)$ has the following properties :

- entry $(m, n)$ of $\mathbf{B}(z)$ contains the $k$–th (out of $K$) polyphase component of the prototype $h_0$ if $m = k \bmod M$ and $n = k \bmod N$.
- $\mathbf{B}(z)$ has $MN$ entries of which only $\frac{MN}{g}$ can be non–zero. All entries for which $m - n$ is divisible by $g$ will have a contribution of just one polyphase component of $h_0$.
- it is observed that $\mathbf{B}(z)$ is an $M \times N$ structured matrix in which on each row and each column $g-1$ zero entries alternate with a single non–zero entry. The polyphase filters appear as diagonals in $\mathbf{B}(z)$ (cf. Eq. 3.5).
- $\mathbf{B}(z)$ can be thought of as a set of overlaid $M \times N$ submatrices of a diagonal $K \times K$ matrix. This is illustrated for the $(M = 6, N = 4)$–case in figure 3.2.
- if $M$ and $N$ are relatively prime, $\mathbf{B}(z)$ is a dense matrix, if $M$ and $N$ have common divisors, $\mathbf{B}(z)$ has several zero entries
- consider just the delay elements. The matrix containing the delay elements $z^{-q}$ is Toeplitz. Moreover, it can be seen as a part of an extended right–circulant $M \times M$ matrix. This is illustrated in figure 3.3 for $M = 6$ and $N = 4$.

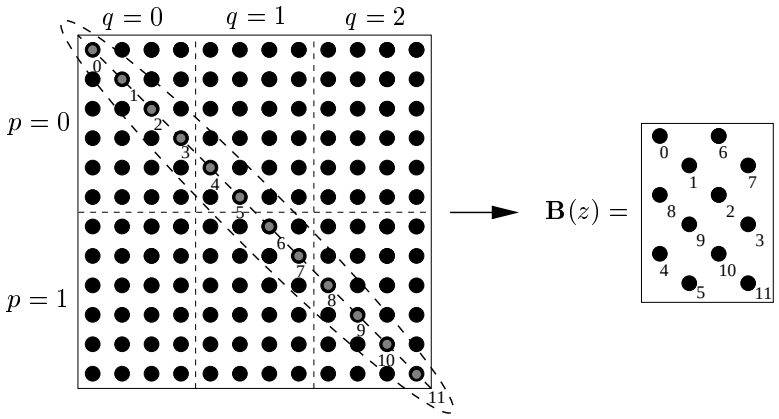


Figure 3.2: $\mathbf{B}(z)$ can be thought of as a set of overlaid $M \times N$ submatrices of a diagonal $K \times K$ matrix. This is illustrated for the $(M = 6, N = 4)$–case.

| | | | | | |
|---|---|---|---|---|---|
| 0 | × | 1 | × | 2 | × |
| × | 0 | × | 1 | × | 2 |
| 2 | × | 0 | × | 1 | × |
| × | 2 | × | 0 | × | 1 |
| 1 | × | 2 | × | 0 | × |
| × | 1 | × | 2 | × | 0 |

Figure 3.3: The Toeplitz matrix containing the delays for $M = 6$ and $N = 4$ can be embedded in a $6 \times 6$ right–circulant matrix

More detailed information and a formal proof can be found in appendix B.2.

### 3.1.2 DFT modulated synthesis filter bank

In the previous section it was shown how the analysis polyphase matrix of a DFT modulated filter bank can be factorized as $\mathbf{H}(z) = \mathbf{FB}(z)$. The synthesis polyphase matrix $\mathbf{G}(z)$ can be decomposed in an analogous way as

$$\mathbf{JG}^T(z) = \mathbf{C}(z)\mathbf{F}^{-1} \tag{3.8}$$

where $\mathbf{C}(z)$ is an $N \times M$ matrix. If both the analysis and the synthesis polyphase matrix are decomposed, figure 2.7 can be redrawn resulting in figure 3.4.

At first sight it is not clear whether $\mathbf{C}(z)$ has the same structure as $\mathbf{B}(z)$ in Eq.

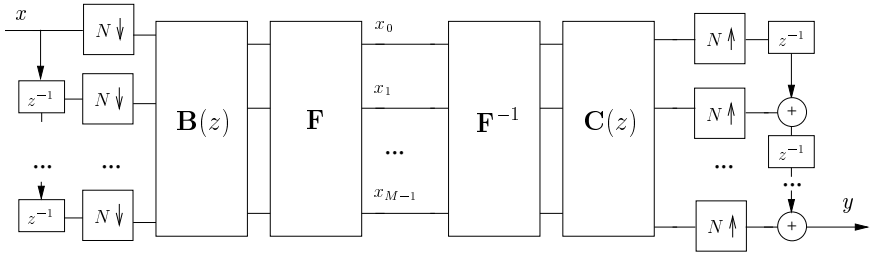


Figure 3.4: A DFT modulated subband system with polyphase decomposition for both the analysis and the synthesis filter bank.

3.3 if the synthesis filter bank is DFT or IDFT modulated. On the other hand, if $\mathbf{C}(z)$ is constructed having a sparse structure similar to that of $\mathbf{B}(z)$, it needs to be verified that indeed a modulated synthesis filter bank is obtained.

### IDFT modulated synthesis filter bank

First, $\mathbf{C}(z)$ is defined in a similar way as $\mathbf{B}(z)$ (cf. Eq. 3.3). In that case, $\mathbf{C}(z)$ is a structured $N \times M$ matrix . Element $(N-1-n, m)$ of $\mathbf{C}(z)$ can be chosen equal to

$$\begin{aligned} [\mathbf{C}(z)]_{N-1-n,m} &= M z^{-q} G_{0_{(n+qN):K}}(z^Q), && \text{if } (n-m) \bmod g = 0 \\ &= 0 && \text{if } (n-m) \bmod g \neq 0, \end{aligned} \tag{3.9}$$

in which $0 \leqslant m \in \mathbb{N} < M$ and $0 \leqslant n \in \mathbb{N} < N$. Further, $(n+qN) \bmod M = m$, $g = \gcd(M, N)$, $Q = \frac{M}{g}$ and $K = \frac{MN}{g}$. $G_{0_{k:K}}(z)$ is the $k$–th out of $K$ polyphase components of a prototype filter $g_0[k]$.

$\mathbf{G}(z)$ can be computed from Eq. 3.8 as

$$\mathbf{G}(z) = \frac{1}{M} \mathbf{F}^* \mathbf{C}^T(z) \mathbf{J}. \tag{3.10}$$

By comparing Eq. 3.10 with Eq. 3.1 and Eq. 3.9 with Eq. 3.3 and assuming that the coefficients of $G_0(z)$ are real–valued, it appears that $\mathbf{Q}(z) = \mathbf{G}_*(z)$ is the polyphase matrix of a DFT modulated filter bank, with

$$q_m[k] = g_0[k] e^{-j\frac{2\pi k m}{M}}. \tag{3.11}$$

From this, it follows that

$$g_m[k] = g_0[k] e^{j\frac{2\pi k m}{M}}. \tag{3.12}$$

Hence, *it can be concluded that the synthesis filters are IDFT modulated equivalents of the synthesis prototype (cf. Eq. 2.12).*

**Example 3.2** Consider the following example for which $M = 3$, $N = 2$ and

$$g_0[k] = \begin{cases} k+1 & \text{if } 0 \leqslant k \leqslant 11 \\ 0 & \text{if } k < 0 \text{ or } k > 11. \end{cases} \tag{3.13}$$

Following Eq. 3.9

$$\mathbf{C}(z) = 3 \begin{bmatrix} 4z^{-1} + 10z^{-4} & 2 + 8z^{-3} & 6z^{-2} + 12z^{-5} \\ 1 + 7z^{-3} & 5z^{-2} + 11z^{-5} & 3z^{-1} + 9z^{-4} \end{bmatrix} \tag{3.14}$$

is obtained. △

A disadvantage of this approach is that the analysis bank is DFT modulated, whereas the synthesis part is IDFT modulated. As prototype filters are typically lowpass, corresponding subband filters in the analysis and synthesis part do not match as they cover different spectral intervals.

### Modified DFT modulated synthesis filter bank

A more appropriate choice for $\mathbf{C}(z)$ might be

$$\begin{aligned} [\mathbf{C}(z)]_{n,m}(z) &= M z^{-\delta} z^q E_{(n+qN):K}(z^{-Q}), && \text{if } (n-m) \bmod g = 0 \\ &= 0 && \text{if } (n-m) \bmod g \neq 0, \end{aligned} \tag{3.15}$$

in which $0 \leqslant m \in \mathbb{N} < M$ and $0 \leqslant n \in \mathbb{N} < N$. Further, $(n + qN) \bmod M = m$, $g = \gcd(M, N)$, $Q = \frac{M}{g}$ and $K = \frac{MN}{g}$. $E_{k:K}(z)$ is the $k$–th out of $K$ polyphase components of a prototype filter $e[k]$ having real coefficients. If $L_f^s$ is the length of $e[k]$, a delay

$$\delta = \left\lceil \frac{L_f^s}{N} \right\rceil - 1 \tag{3.16}$$

is inserted to make the synthesis filters causal.

From Eq. 3.15

$$\mathbf{C}(z) = M z^{-\delta} \tilde{\mathbf{B}}_e(z) \tag{3.17}$$

is found, with $\tilde{\mathbf{B}}_e(z) = \mathbf{B}_{e_*}^T(z^{-1})$ the paraconjugate of $\mathbf{B}_e(z)$. Matrix $\tilde{\mathbf{B}}_e(z)$ is obtained by transposing $\mathbf{B}_e(z)$, replacing all filter coefficients by their complex conjugate and replacing $z$ by $z^{-1}$ (see also appendix A). By comparing Eq. 3.15, 3.17 and 3.3 it follows that $\mathbf{B}_e(z)$ is the prototype polyphase matrix of a DFT modulated filter bank based on $e[k]$.

By filling in definition 3.17 in Eq. 3.8,

$$\mathbf{G}^T(z) = M z^{-\delta} \mathbf{J} \tilde{\mathbf{B}}_e(z) \mathbf{F}^{-1} \tag{3.18}$$

is obtained. So,

$$\mathbf{G}(z) = M \mathbf{F}^{-T} \tilde{\mathbf{B}}_e^T(z) \mathbf{J} z^{-\delta} \tag{3.19}$$

$$= \tilde{\mathbf{H}}_e^T(z)\mathbf{J}z^{-\delta}, \tag{3.20}$$

with $\mathbf{H}_e(z) = \mathbf{F}\mathbf{B}_e(z)$ the polyphase matrix corresponding to a DFT modulated analysis filter bank based on the prototype $e[k]$. This implies that

$$\begin{aligned} G_{m_{n:N}}(z) &= [\mathbf{G}(z)]_{m,n} && (3.21)\\ &= z^{-\delta}\left[\mathbf{H}_{e_*}(z^{-1})\right]_{m,(N-1-n)} && (3.22)\\ &= z^{-\delta}E^*_{m_{N-1-n:N}}(z^{-1}) && (3.23)\end{aligned}$$

with $E_{m_{n:N}}(z)$ polyphase components of $E_m(z) = E(e^{j\frac{2\pi m}{M}}z)$. Hence from Eqs. 2.16, 3.16 and 3.23,

$$\begin{aligned} G_m(z) &= \sum_{n=0}^{N-1} z^{-n}z^{-N\delta}E^*_{m_{N-1-n:N}}(z^{-N}) && (3.24)\\ &= z^{-(N\delta+N-1)}\sum_{n=0}^{N-1} z^{N-1-n}E^*_{m_{N-1-n:N}}(z^{-N}) && (3.25)\\ &= z^{-(N\left\lceil\frac{L_f^s}{N}\right\rceil-1)}E^*_m(z^{-1}) && (3.26)\end{aligned}$$

is obtained. If $L_f^s$ is a multiple of $N$, the synthesis filters $g_m[k]$ can be expressed as

$$\begin{aligned} g_m[k] &= \mathrm{conj}\left(e_m[L_f^s-1-k]\right) && (3.27)\\ &= e[L_f^s-1-k]e^{j\frac{2\pi(L_f^s-1-k)m}{M}} && (3.28)\\ &= e^{j\frac{2\pi(L_f^s-1)m}{M}}g_0[k]e^{-j\frac{2\pi km}{M}}, && (3.29)\end{aligned}$$

with $g_0[k] = e[L_f^s-k-1]$. From Eq. 3.29 *it can be concluded that in this case the synthesis filters are modified DFT modulated equivalents of the synthesis prototype* $g_0[k]$.

**Example 3.3** Consider the following example for which $M = 4$, $N = 2$ and

$$g_0[k] = \begin{cases} k+1 & \text{if } 0 \leqslant k \leqslant 9\\ 0 & \text{if } k < 0 \text{ or } k > 9. \end{cases} \tag{3.30}$$

Based on Eq. 3.15

$$\mathbf{C}(z) = 4\begin{bmatrix} 9+5z^{-2}+1z^{-4} & 0 & 7z^{-1}+3z^{-3} & 0\\ 0 & 10+6z^{-2}+2z^{-4} & 0 & 8z^{-1}+4z^{-3}\end{bmatrix} \tag{3.31}$$

is found. △

### 3.1.3 Implementation issues

By employing FFT techniques and polyphase decomposition of the lowpass prototype an efficient implementation is feasible. For a real–valued FIR filter bank with filters of length $L_f$ implemented in the standard way as in figure 2.1, the number of elementary filtering operations[1] per second amounts to $\left(\frac{M}{2}+1\right) L_f f_s$, where $f_s$ is the sampling frequency of the input signal $x$. As the input signal is assumed to be real, only about half of the filters have to be computed, hence the factor $\frac{M}{2}+1$ for even values of $M$.

The number of elementary operations per second required to implement a DFT modulated filter bank in an efficient way using FFT and polyphase decomposition as in figure 3.4 is $(2M \log_2 M - 4M + 6)\frac{f_s}{N}$ for the FFT plus $L_f \frac{f_s}{N}$ for the computation of $\mathbf{B}(z)$, as only $K$ of the $MN$ entries of $\mathbf{B}(z)$ are non–zero filters of length $\frac{L_f}{K}$. Also here the filter bank and the input signal are assumed to be real. Instead of $\frac{M}{2}+1$ filters in the standard approach, in this case only the equivalent of 1 filter is implemented together with an FFT, both at the downsampled rate.

**Example 3.4** For a realistic setup for which $M = 16$, $N = 12$, $L_f = 48$ and $f_s = 8000$ Hz the standard implementation comes down to 3456000 operations per second. For the efficient implementation the total cost amounts to 78667 operations per second. This proves that the cost reduction that can be achieved by efficiently implemented DFT modulated banks is considerable. △

## 3.2 Perfect reconstruction

Ideally, signal $y$ at the output of the subband scheme (see figure 3.4) is an exact (delayed) copy of the input $x$. A DFT modulated analysis/synthesis filter bank set is therefore designed such that the following conditions are met :

1. for the analysis/synthesis filter bank set the *perfect reconstruction* property holds. A condition ensuring perfect reconstruction is

$$\mathbf{C}(z)\mathbf{B}(z) = z^{-\delta}\mathbf{I}_N, \qquad \delta \in \mathbb{N}. \tag{3.32}$$

   In this case the inner filtering operations (see figure 3.4) cancel each other out and the subband scheme reduces to a combined analysis and synthesis tapped delay line. It can be verified that this leads to perfect reconstruction [156]. The smallest $\delta$ is determined such that both the analysis and synthesis filters are causal and hence implementable.

[1] An elementary filtering or *multiply–accumulate* operation can be efficiently computed on dedicated processors (DSPs) and counts for 1 machine cycle. For more information about cost computation, see section 2.4.

Condition 3.32 can be relaxed leading to *nearly perfect reconstruction* up to a delay, which will be discussed in chapter 4.

In a practical design $\delta$ is initially set equal to zero, which in general leads to non–causal filters. The a–causality is then removed after the design has been completed.

2. the (analysis) filters are *frequency selective*. This reduces the amount of aliasing components inserted in the subbands. Inter–subband aliasing often has a negative impact on possible intermediate operations performed on the subband signals (e.g. subband adaptive filtering). Highly frequency selective filter banks on the other hand are expensive and introduce a substantial processing delay. This puts a constraint on the frequency selectivity, and hence on the downsampling factor $N$. However, computational savings are more or less proportional to $N$.

In practice the filter bank order is finite, hence there will always be some aliasing inserted in the subbands. Nevertheless, if a set $(\mathbf{B}(z), \mathbf{C}(z))$ is found satisfying Eq. 3.32 the overall input–output behavior is distortionless and aliasing–free.

Further, it is clear that once a perfect reconstruction subband system has been obtained changing the subsampling factor $N$ will in general not preserve the perfect reconstruction property.

**Theorem 3.2** *However, if an $M$–band, $N$–fold downsampled filter bank set satisfies the perfect reconstruction condition, perfect reconstruction will also hold for $\breve{N} = \frac{N}{p}$, with $N, \breve{N}, p \in \mathbb{N}_0$. Furthermore, it is found that the output/input level increases by a factor $p$ when $N$ is divided by $p$.*

**Proof:** Surprisingly no reference to this theorem has been found in the literature that was available to us. Hence, a proof of theorem 3.2 is given in appendix B.3.

**Example 3.5** If a 10–band filter bank set is designed such that $y = x$ for $N = 6$ (as will be done in example 3.12), perfect reconstruction will also hold for $N = 3$ (for which then $y = 2x$), for $N = 2$ (with $y = 3x$) and in the case where the signals are not subsampled ($y = 6x$). △

In order to fulfill Eq. 3.32, $\mathbf{C}(z)$ has to be a left inverse of $\mathbf{B}(z)$. This left inverse exists if $M \geqslant N$. However, the synthesis filter bank that is obtained by inversion is not necessarily FIR or DFT modulated. In the most general case $\mathbf{C}(z)$ is a dense rational matrix and hence, the corresponding filter bank is IIR. This is discussed in more detail in the next section.

### 3.2.1 Smith–McMillan decomposition based perfect reconstruction filter bank design

Assume that the Smith–McMillan decomposition of the $M \times N$ polynomial matrix $\mathbf{B}(z)$ is given by (see also appendix A)

$$\mathbf{B}(z) = \mathbf{R}(z)\mathbf{\Sigma}(z)\mathbf{T}(z) \tag{3.33}$$

where $\mathbf{R}(z)$ and $\mathbf{T}(z)$ are respectively $M \times M$ and $N \times N$ unimodular matrices. Further, $\mathbf{\Sigma}(z)$ is an $M \times N$ pseudo–diagonal polynomial matrix, which can be expressed as

$$\mathbf{\Sigma}(z) = \begin{bmatrix} \mathbf{D}(z) \\ \mathbf{0}_{(M-N)\times N} \end{bmatrix} \tag{3.34}$$

if $M \geqslant N$, with

$$\mathbf{D}(z) = \begin{bmatrix} d_0(z) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & d_{N-1}(z) \end{bmatrix}. \tag{3.35}$$

In order to obtain a filter bank set for which the perfect reconstruction condition (Eq. 3.32) is fulfilled, $\mathbf{C}(z)$ can be chosen equal to

$$\begin{aligned} \mathbf{C}(z) &= \mathbf{B}^{\dagger}(z) + \mathbf{N}(z) && (3.36) \\ &= \mathbf{T}^{-1}(z)\left(\begin{bmatrix}\mathbf{D}^{-1}(z) & \mathbf{0}_{N\times(M-N)}\end{bmatrix} + \begin{bmatrix}\mathbf{0}_{N\times N} & \mathbf{X}(z)\end{bmatrix}\right)\mathbf{R}^{-1}(z), && (3.37) \end{aligned}$$

provided $\mathbf{D}(z)$ is a full rank polynomial matrix. $\mathbf{N}(z)$ is an $N \times M$ rational matrix that lies in the left null space of $\mathbf{B}(z)$. $\mathbf{X}(z)$ is an $N \times (M-N)$ random rational matrix. $\mathbf{R}(z)$ and $\mathbf{T}(z)$ are unimodular polynomial matrices, so their inverse $\mathbf{R}^{-1}(z) = \operatorname{adj}\mathbf{R}(z)/\det\mathbf{R}(z)$ is also polynomial.

In the most general case $\mathbf{D}^{-1}(z)$ is rational and hence the synthesis filter bank is IIR. A true FIR inverse can be obtained if $\mathbf{D}(z)$ has monomial diagonal elements, i.e. $d_n(z) = \alpha_n z^{\delta_n}$, $\alpha_n \in \mathbb{R}_0$, $\delta_n \in \mathbb{Z}$, $\forall n$, and $\mathbf{N}(z)$ is polynomial (see also [24]). It is our experience that in most practical applications $\mathbf{D}(z)$ has monomial elements and that hence an FIR synthesis filter bank can be obtained. It is however not clear to us which conditions $\mathbf{B}(z)$ has to satisfy for this to be true.

In order to obtain a modulated synthesis bank, $\mathbf{C}(z)$ should be a polynomial matrix with a structure similar to that of $\mathbf{B}(z)$ (see section 3.1.2). In that case the synthesis filters have equal spectral characteristics, up to a frequency shift. Furthermore, a cheap implementation would be possible thanks to the sparse structure of $\mathbf{C}(z)$.

The Smith–McMillan decomposition diagonalizes $\mathbf{B}(z)$ by means of a finite number of elementary row and column operations, which are contained in the unimodular matrices $\mathbf{R}(z)$ and $\mathbf{T}(z)$ [156]. Hence, $\mathbf{R}^{-1}(z)$ and $\mathbf{T}^{-1}(z)$ perform the inverse elementary operations on $\begin{bmatrix}\mathbf{D}^{-1}(z) & \mathbf{0}\end{bmatrix}$, which apparently put $\mathbf{B}^{\dagger}(z)$ in the desired

format described by Eq. 3.15. If the analysis bank is therefore DFT modulated a modified DFT modulated synthesis filter bank can be obtained, as desired. If $\mathbf{N}(z)$, which is based on a random $\mathbf{X}(z)$, is added to $\mathbf{B}^{\dagger}(z)$ the structure will not be maintained in general.

**Example 3.6** A 12–taps FIR filter was randomly generated and is chosen to be the analysis prototype filter of a 4–band, 2–fold downsampled DFT modulated subband system. The corresponding analysis polynomial matrix $\mathbf{B}(z)$ is equal to

$$\mathbf{B}(z) = \begin{bmatrix} 0.456 - 0.363z^{-2} + 0.258z^{-4} & 0 \\ 0 & 0.539 - 0.463z^{-2} + 0.0716z^{-4} \\ -0.052z^{-1} - 0.064z^{-3} - 0.277z^{-5} & 0 \\ 0 & 0.163z^{-1} + 0.344z^{-3} - 0.367z^{-5} \end{bmatrix} \tag{3.38}$$

The Smith–McMillan decomposition of $\mathbf{B}(z)$ can be computed and is given by

$$\mathbf{B}(z) = \mathbf{R}(z)\mathbf{\Sigma}(z)\mathbf{T}(z) \tag{3.39}$$

with

$$\mathbf{R}(z) = \begin{bmatrix} r_{11} & 0 & r_{13} & 0 \\ 0 & r_{22} & 0 & r_{24} \\ r_{31} & 0 & r_{33} & 0 \\ 0 & r_{42} & 0 & r_{44} \end{bmatrix}, \quad \begin{cases} r_{11} = 1 - 0.79z^{-2} + 0.56z^{-4} \\ r_{13} = -0.49z^{-1} - 0.19z^{-3} \\ r_{22} = 1 - 0.86z^{-2} + 0.13z^{-4} \\ r_{24} = 3.1z^{-1} - 0.62z^{-3} \\ r_{31} = 0.11z^{-1} - 0.14z^{-3} - 0.61z^{-5} \\ r_{33} = 1 + 0.85z^{-2} + 0.2z^{-4} \\ r_{42} = 0.3z^{-1} + 0.64z^{-3} - 0.68z^{-5} \\ r_{44} = 1 + 1.8z^{-2} + 3.2z^{-4} \end{cases} \tag{3.40}$$

$$\mathbf{\Sigma}(z) = \begin{bmatrix} 0.456 & 0 \\ 0 & 0.539 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{D}(z) = \begin{bmatrix} 0.456 & 0 \\ 0 & 0.539 \end{bmatrix} \tag{3.41}$$

and

$$\mathbf{T}(z) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{3.42}$$

Based on Eqs. 3.40, 3.41 and 3.42 and taking $\mathbf{N}(z) = \mathbf{0}$, $\mathbf{C}(z)$ can be computed as

$$\mathbf{C}(z) = \mathbf{T}^{-1}(z) \begin{bmatrix} \mathbf{D}^{-1}(z) & \mathbf{0}_2 \end{bmatrix} \mathbf{R}^{-1}(z) \tag{3.43}$$

$$= \begin{bmatrix} c_{11} & 0 & c_{13} & 0 \\ 0 & c_{22} & 0 & c_{24} \end{bmatrix}, \quad \begin{cases} c_{11} = 2.2 + 1.9z^{-2} + 0.44z^{-4} \\ c_{13} = 1.1z^{-1} + 0.41z^{-3} \\ c_{22} = 1.9 + 3.3z^{-2} + 5.9z^{-4} \\ c_{24} = 5.7z^{-1} + 1.1z^{-3} \end{cases} \tag{3.44}$$

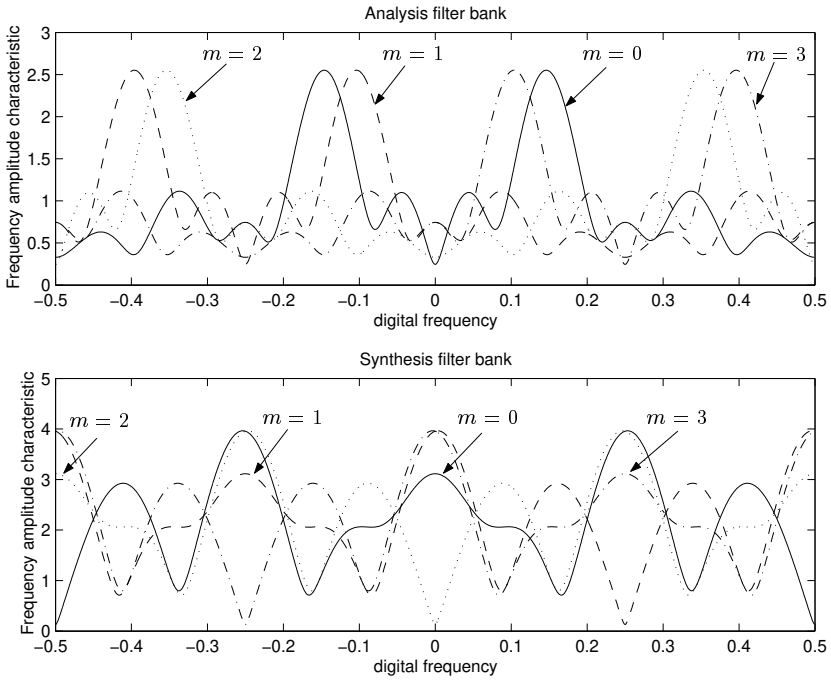


Figure 3.5: A 4–band, 2–fold downsampled perfect reconstruction DFT modulated filter bank set corresponding to Eqs. 3.38 and 3.44 and designed using the Smith–McMillan decomposition

such that

$$\mathbf{C}(z)\mathbf{B}(z) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{3.45}$$

It can be verified that the structure of matrix $\mathbf{C}(z)$ is in agreement with Eq. 3.15, i.e. the corresponding synthesis filter bank will be a modified DFT modulated filter bank. The analysis and synthesis filter bank are plotted in figure 3.5. △

For many realistic scenarios the synthesis filter bank coefficients turn out to be large in amplitude, typically much larger than the coefficients of the analysis filter bank. Due to the limited numerical precision during the computation of the Smith–McMillan decomposition on the one hand as well as during filtering when the filter banks eventually are in operation on the other hand, perfect reconstruction will not be achieved in practice. An appropriate choice for matrix $\mathbf{N}(z)$ may lead to a more robust solution, with smaller filter bank coefficients. This matrix should preferably be chosen such that the sparse structure of $\mathbf{C}(z)$ is maintained. It is however not clear how a suitable $\mathbf{N}(z)$ can be found.

Another disadvantage is that, although the analysis filter bank is designed beforehand and can be chosen to be highly frequency selective, this design method does not guarantee that also the synthesis prototype has a satisfactory lowpass characteristic.

### 3.2.2 Para–unitary filter banks

The Smith–McMillan decomposition is a very powerful design technique as for basically any analysis filter bank a synthesis filter bank can be computed leading to perfect reconstruction input/output behavior. In practice however the synthesis filter bank characteristics are not always as desired, suggesting that a more restrictive design technique might be more appropriate.

As an alternative to the previous design method now para–unitary filter banks are discussed. The analysis filter bank has to fulfill a para–unitary condition in this case, and cannot be chosen completely freely anymore. However, a numerically stable perfect reconstruction DFT modulated filter bank set can be obtained.

Starting again from Eq. 3.37 an elegant solution results when $\mathbf{X}(z) = \mathbf{0}$, $\mathbf{D}(z)$ and $\mathbf{T}(z)$ are chosen to be unit matrices and $\mathbf{R}(z)$ is para–unitary, i.e.

$$\tilde{\mathbf{R}}(z)\mathbf{R}(z) = \mathbf{I}_M, \qquad \mathbf{D}(z) = \mathbf{T}(z) = \mathbf{I}_N, \quad \mathbf{X}(z) = \mathbf{0}. \tag{3.46}$$

From Eq. 3.33 and 3.46 it follows that

$$\tilde{\mathbf{B}}(z)\mathbf{B}(z) = \mathbf{I}_N. \tag{3.47}$$

Now, $\mathbf{C}(z)$ can be chosen to be the paraconjugate of $\mathbf{B}(z)$, i.e.

$$\mathbf{C}(z) = \tilde{\mathbf{B}}(z). \tag{3.48}$$

Hence,

$$\mathbf{C}(z)\mathbf{B}(z) = \tilde{\mathbf{B}}(z)\mathbf{B}(z) = \mathbf{I}_N, \tag{3.49}$$

so that a perfect reconstruction subband system is obtained. If $\mathbf{C}(z)$ is designed following Eq. 3.48 a polynomial matrix is obtained having a sparse structure corresponding to Eq. 3.15.

As paraconjugation implies time inversion, $\mathbf{C}(z) = \tilde{\mathbf{B}}(z)$ leads to a non–causal synthesis filter bank, which can easily be made causal by inserting a delay :

$$\mathbf{C}(z) = z^{-\delta}\tilde{\mathbf{B}}(z), \qquad \delta \in \mathbb{N} \geqslant \left\lceil \frac{L_f^a}{N} \right\rceil - 1, \tag{3.50}$$

in which $L_f^a$ is the length of the analysis prototype filter $h_0[k]$.

The synthesis polyphase matrix can be computed based on Eqs. 3.8 and 3.50 :

$$\mathbf{J}\mathbf{G}^T(z) = \mathbf{C}(z)\mathbf{F}^{-1} \tag{3.51}$$

so that

$$\mathbf{G}^T(z) = z^{-\delta}\mathbf{J}\tilde{\mathbf{B}}(z)\mathbf{F}^{-1}. \tag{3.52}$$

Further, as for a DFT modulated filter bank the analysis polyphase matrix can be expressed as

$$\mathbf{H}(z) = \mathbf{F}\mathbf{B}(z) \tag{3.53}$$

finally

$$\mathbf{G}(z) = \frac{z^{-\delta}}{M}\mathbf{H}_*(z^{-1})\mathbf{J} \tag{3.54}$$

is found. The synthesis filters can be derived from Eqs. 2.9 and 3.54 (cf. Eq. 3.29). If $L_f^a$ is a multiple of $N$,

$$g_m[k] = \frac{1}{M}\mathrm{conj}(h_m[L_f^a - 1 - k]) \tag{3.55}$$

$$= \frac{1}{M}e^{j\frac{2\pi(L_f^a-1)m}{M}}h_0[L_f^a - 1 - k]e^{-j\frac{2\pi km}{M}}. \tag{3.56}$$

It appears that the synthesis filter bank is a modified oversampled DFT modulated filter bank. The synthesis filters $g_m[k]$ are shifted versions of each other and have the same frequency amplitude characteristic as the analysis filters.

## 3.3 Para–unitary filter bank design

If a para–unitary analysis polynomial matrix $\mathbf{B}(z)$ can be found (cf. Eq. 3.47) having the special format specified in Eq. 3.3, a perfect reconstruction oversampled DFT modulated filter bank set is straightforwardly obtained from Eqs. 3.50 or 3.55. In this section a para–unitary parameterization for the analysis polynomial matrix is discussed, which can be optimized to obtain a frequency selective subband system fulfilling the perfect reconstruction property.

### 3.3.1 Imposing para–unitarity

From appendix B.2 it is known that $\mathbf{B}(z)$ is a structured $M \times N$ matrix with non–zero entries appearing in a specific pattern (e.g. Eq. 3.5).

**Theorem 3.3** *After appropriate row and column permutations* $\boldsymbol{P}$ *and* $\boldsymbol{Q}$ *a block diagonal matrix*

$$\mathbf{P}\mathbf{B}(z)\mathbf{Q} = \begin{bmatrix} \mathbf{S}_0(z) & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{S}_{g-1}(z) \end{bmatrix} \tag{3.57}$$

*is obtained consisting of* $g$ *dense submatrices* $\boldsymbol{S}_\gamma(z)$ *of dimension* $\frac{M}{g} \times \frac{N}{g}$. *As already defined in Eq. 3.3,* $g$ *is the greatest common divisor of* $M$ *and* $N$. *Further,* $\gamma$ *is an integer between 0 and* $g-1$.

**Proof :** The theorem is formulated in [22] [24], but no explicit proof is given. A proof of theorem 3.3 can be found in appendix B.4.

**Example 3.7** Referring to example 3.1 and Eq. 3.5, $\mathbf{B}(z)$ can be transformed into the following block diagonal matrix :

$$\mathbf{PB}(z)\mathbf{Q} = \left[\begin{array}{cc|cc} H_{0_{0:12}}(z^3) & z^{-1}H_{0_{6:12}}(z^3) & 0 & 0 \\ z^{-2}H_{0_{8:12}}(z^3) & H_{0_{2:12}}(z^3) & 0 & 0 \\ z^{-1}H_{0_{4:12}}(z^3) & z^{-2}H_{0_{10:12}}(z^3) & 0 & 0 \\ \hline 0 & 0 & H_{0_{1:12}}(z^3) & z^{-1}H_{0_{7:12}}(z^3) \\ 0 & 0 & z^{-2}H_{0_{9:12}}(z^3) & H_{0_{3:12}}(z^3) \\ 0 & 0 & z^{-1}H_{0_{5:12}}(z^3) & z^{-2}H_{0_{11:12}}(z^3) \end{array}\right] \tag{3.58}$$

△

All $\frac{MN}{g}$ polyphase components of $h_0[k]$ are now contained in one of the submatrices $\mathbf{S}_\gamma(z)$. Note that the left and right multiplication with permutation matrices $\mathbf{P}$ and $\mathbf{Q}$ does not affect the para–unitarity of $\mathbf{B}(z)$, as for a permutation matrix $\mathbf{P}^T\mathbf{P} = \mathbf{I}$.

Block diagonal matrix $\mathbf{PB}(z)\mathbf{Q}$ is para–unitary if each submatrix $\mathbf{S}_\gamma(z)$ itself is para–unitary. Hence, if we can find a set of para–unitary matrices $\mathbf{S}_\gamma(z)$ also $\mathbf{PB}(z)\mathbf{Q}$ and $\mathbf{B}(z)$ will be para–unitary.

It appears that the non–zero elements of $\mathbf{PB}(z)\mathbf{Q}$, namely $z^{-q}H_{0_{(n+qN):K}}(z^Q)$ (cf. Eq. 3.3), are delayed $Q$–fold upsampled polyphase filters. In practice the downsampled polyphase filters $H_{0_{(n+lN):K}}(z)$ are optimized instead of their upsampled form in order to reduce the optimization time. At first sight, the delays $z^{-q}$ pose a problem : they cannot simply be downsampled as they are not powers of $z^Q$. Theorem 3.4 however shows that the delays can always be made multiples of $Q$ by pre– and post–multiplying $\mathbf{S}_\gamma(z)$ with appropriate para–unitary diagonal matrices.

**Theorem 3.4** *By pre– and post–multiplication of $\boldsymbol{S}_\gamma(z)$ with appropriate para–unitary diagonal matrices $\boldsymbol{L}_\gamma(z)$ and $\boldsymbol{R}_\gamma(z)$, all the elements of*

$$\hat{\mathbf{S}}_\gamma(z^Q) = \mathbf{L}_\gamma(z)\mathbf{S}_\gamma(z)\mathbf{R}_\gamma(z) \tag{3.59}$$

*depend on $z^Q$.*

**Proof :** Theorem 3.4 is illustrated in [22] [24] by means of two examples, but is not explicitly formulated or proven for the general case. A formal proof is given in appendix B.5.

**Example 3.8** We return to example 3.1 : the submatrices $\mathbf{S}_\gamma(z)$ have dimensions $3 \times 2$. After pre– and post–multiplication of $\mathbf{S}_0(z)$ (Eq. 3.58) for instance by $\mathbf{L}_0$

and $\mathbf{R}_0$ respectively

$$\hat{\mathbf{S}}_0(z^3) = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & z^{-1} & 0 \\ 0 & 0 & z \end{bmatrix}}_{\mathbf{L}_0(z)} \mathbf{S}_0(z) \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & z \end{bmatrix}}_{\mathbf{R}_0(z)} = \begin{bmatrix} H_{0_{0:12}}(z^3) & H_{0_{6:12}}(z^3) \\ z^{-3}H_{0_{8:12}}(z^3) & H_{0_{2:12}}(z^3) \\ H_{0_{4:12}}(z^3) & H_{0_{10:12}}(z^3) \end{bmatrix} \tag{3.60}$$

is obtained in which all elements depend on $z^3$. △

As $\hat{\mathbf{S}}_\gamma(z^Q)$ depends on $z^Q$, now $\hat{\mathbf{S}}_\gamma(z)$ can be optimized : if $\hat{\mathbf{S}}_\gamma(z)$ is para–unitary, so will $\hat{\mathbf{S}}_\gamma(z^Q)$. Designing a para–unitary $\mathbf{B}(z)$ then comes down to imposing para–unitarity on a set of dense matrices $\hat{\mathbf{S}}_\gamma(z)$. Once $\hat{\mathbf{S}}_\gamma(z)$ has been found, $\mathbf{S}_\gamma(z)$ and $\mathbf{B}(z)$ can be computed.

Each of these rectangular blocks $\hat{\mathbf{S}}_\gamma(z)$ of size $\frac{M}{g} \times \frac{N}{g}$ can now be embedded in a larger $\frac{M}{g} \times \frac{M}{g}$ para–unitary matrix $\mathbf{M}_\gamma(z)$ whose para–unitarity automatically implies the para–unitarity of a subset of its columns. More information about $\mathbf{M}_\gamma(z)$ and its parameterization using para–unitary lattices can be found in the next section.

### 3.3.2 Para–unitary lattices

Square polynomial matrices can be made para–unitary by parameterizing them as para–unitary lattices. A causal real–valued $\frac{M}{g} \times \frac{M}{g}$ FIR para–unitary matrix $\mathbf{M}_\gamma(z)$ of degree $L_\gamma$ can be decomposed as [24] [156]

$$\mathbf{M}_\gamma(z) = \mathbf{V}_{L_\gamma}(z)\mathbf{V}_{L_\gamma-1}(z)\ldots\mathbf{V}_1(z)\mathbf{U} \tag{3.61}$$

where $\mathbf{V}_l(z)$ are degree–1 para–unitary building blocks (Householder type)

$$\mathbf{V}_l(z) = \mathbf{I}_{\frac{M}{g}} - \mathbf{v}_l\mathbf{v}_l^T + z^{-1}\mathbf{v}_l\mathbf{v}_l^T. \tag{3.62}$$

$\mathbf{U}$ is an orthonormal matrix, which itself can be decomposed as

$$\mathbf{U} = \mathbf{U}_1\mathbf{U}_2\ldots\mathbf{U}_{\frac{M}{g}-1}. \tag{3.63}$$

in which $\mathbf{U}_l$ are Householder matrices

$$\mathbf{U}_l = \mathbf{I}_{\frac{M}{g}} - 2\mathbf{u}_l\mathbf{u}_l^T. \tag{3.64}$$

All $\mathbf{v}_l$ and $\mathbf{u}_l$ are $\frac{M}{g} \times 1$ unit norm vectors.

From Eqs. 3.63 and 3.64 it follows that the $\frac{M}{g} \times \frac{M}{g}$ orthonormal matrix $\mathbf{U}$ is characterized by $\frac{M}{g} - 1$ vectors $\mathbf{u}_l$. It can be proven (see [156]) that without any loss of generality the description matrix

$$\begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \ldots & \mathbf{u}_{\frac{M}{g}-1} \end{bmatrix} \tag{3.65}$$

can have the following lower triangular form :

$$\begin{bmatrix} \mathbf{u}_1(0) & 0 & \dots & 0 \\ \mathbf{u}_1(1) & \mathbf{u}_2(1) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_1(\frac{M}{g}-2) & \mathbf{u}_2(\frac{M}{g}-2) & \dots & \mathbf{u}_{\frac{M}{g}-1}(\frac{M}{g}-2) \\ \mathbf{u}_1(\frac{M}{g}-1) & \mathbf{u}_2(\frac{M}{g}-1) & \dots & \mathbf{u}_{\frac{M}{g}-1}(\frac{M}{g}-1) \end{bmatrix}. \tag{3.66}$$

### 3.3.3 Optimization of the para–unitary lattices

A total of $g$ square $\frac{M}{g} \times \frac{M}{g}$ matrices $\mathbf{M}_\gamma(z)$ are parameterized as in Eq. 3.61 to guarantee their para–unitarity, from which then automatically follows the para–unitarity of the $\frac{M}{g} \times \frac{N}{g}$ submatrices $\hat{\mathbf{S}}_\gamma(z)$. The lattices are optimized so to obtain a perfect reconstruction filter bank set with acceptable frequency selectivity. For this an optimization procedure will be proposed that in each iteration step computes the filter bank prototype filter and tries to maximize its frequency resolution.

The analysis prototype filter $h_0[k]$ can easily be computed starting from the parameter sets $(\mathbf{v}_l, \mathbf{u}_l)_\gamma$, which parameterize each of the $g$ para–unitary lattices. From $(\mathbf{v}_l, \mathbf{u}_l)_\gamma$ first $\mathbf{M}_\gamma(z)$ and then $\hat{\mathbf{S}}_\gamma(z)$ can be reconstructed following Eqs. 3.61–3.64. From Eqs. 3.59 and 3.57 the polyphase components $z^{-q} H_{0_{(n+qN):K}}(z^Q)$ of the analysis prototype $h_0[k]$ and hence $\mathbf{B}(z)$ can be obtained, from which immediately follows the filter bank characteristic.

The parameter sets $(\mathbf{v}_l, \mathbf{u}_l)_\gamma$ are optimized such that the para–unitary filter bank has a good frequency selectivity and hence aliasing insertion due to the subsampling process is minimal. Therefore the stopband energy of the analysis prototype is computed and minimized by a steepest descent optimization procedure :

$$\min_{(\mathbf{v}_l, \mathbf{u}_l)_\gamma} \int_{\frac{1}{2N}}^{\frac{1}{2}} |H_0(f)|^2 df. \tag{3.67}$$

Equation 3.67 can be computed based on the autocorrelation of $h_0[k]$ and a sinc–function [156]. In this way expensive frequency transforms are avoided.

The delays present in $\hat{\mathbf{S}}_\gamma(z)$ (cf. Eq. 3.60) are taken into account once the polyphase components of $h_0[k]$ are computed based on $\hat{\mathbf{S}}_\gamma(z)$. The delays force the polyphase components to be shifted into the anti–causal direction. Although the number of non–zero coefficients of $h_0[k]$ is not affected by this, the length of the prototype filter increases. As a consequence, some of the coefficients of $h_0[k]$ will typically be zero. The number of operations needed to implement the filter remains unchanged, but the required amount of storage elements typically increases.

Finding a good initialization for $\mathbf{v}_l$ and $\mathbf{u}_l$ is not trivial : the time needed to obtain a good lowpass prototype $h_0[k]$ depends on the quality of the initial values.

Furthermore, the optimization function (stopband energy) to be minimized is not unimodal and hence multiple local minima might exist. However, random initialization will typically lead to satisfactory results after some trial rounds. Another option is to start from a well–chosen lowpass filter $p[k]$, obtained from any FIR filter design toolbox, and derive the $\mathbf{v}_l$ and $\mathbf{u}_l$ parameters corresponding to it. In general this will not be possible as a "randomly" designed lowpass filter will not lead to a para–unitary filter bank and hence a $(\mathbf{v}_l, \mathbf{u}_l)_\gamma$–parameterization simply does not exist for $p[k]$. However, $p[k]$ will be approximated by its "nearest" para–unitary filter, typically leading to better starting values. The $(\mathbf{v}_l, \mathbf{u}_l)_\gamma$–parameterization of a para–unitary prototype can be found by inverse decomposition of Eqs. 3.61 and 3.63. This can be done using the singular value decomposition, pseudo–inverses and Householder factorization and is explained in appendix B.6.

For the special case of 2–fold oversampling ($M = 2N$) a more dedicated parameterization is possible, which is based on 2–channel lossless lattices. In this case the so–called lattice rotation angles are optimized in order to obtain good lowpass filter characteristics. More information can be found in appendix B.7. On page 257 and figure B.2 a design example is given. Inverse parameterization of 2–channel lossless lattices is also explained in appendix B.7 and can be used to find good starting values for the steepest descent optimization procedure.

### 3.3.4 Adjusting the prototype filter length

$L_\gamma$, the order of the para–unitary lattices (see Eq. 3.61) can be chosen freely and defines the length of the analysis prototype filter and hence the order of the filter bank. If the order of one of the para–unitary lattices is increased from $L_\gamma$ to $L_\gamma + 1$, each of the $\frac{MN}{g^2}$ polyphase filters defined by that specific lattice, becomes longer and the analysis prototype gets an extra $\frac{MN}{g^2}$ coefficients. As a consequence at first sight, the number of effective (i.e. non–zero) coefficients of the prototype $h_0[k]$ can only vary in steps of $\frac{MN}{g^2}$. This can be quite unattractive for some coprime parameter settings. For example if $M = 16$ and $N = 15$, one has $g = 1$ and $\frac{MN}{g^2} = 240$. The larger the greatest common divisor of $M$ and $N$, the more para–unitary lattices are involved and the better the length of the prototype can be adjusted.

The minimum number of effective filter coefficients is obtained when all lattices have order 0. In that case all lattices $\mathbf{M}_\gamma(z) = \mathbf{U}_\gamma$ would be orthonormal matrices and the number of non–zero coefficients of $h_0[k]$ amounts to $\frac{MN}{g}$, which still can be quite large. However, with a special choice for $\mathbf{U}_\gamma$, the identity matrix for instance, the number of effective coefficients can be further reduced.

More generally, each of the $g$ para–unitary lattices can be restricted to be a block diagonal matrix

$$\mathbf{M}_\gamma(z) = \begin{bmatrix} \mathbf{M}_{\gamma,0}(z) & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{M}_{\gamma,R_\gamma-1}(z) \end{bmatrix}. \tag{3.68}$$

Each of the $R_\gamma$ square blocks $\mathbf{M}_{\gamma,r}(z)$ is an $N_{\gamma,r} \times N_{\gamma,r}$ matrix such that

$$\sum_{r=0}^{R_\gamma - 1} N_{\gamma,r} = \frac{M}{g}, \tag{3.69}$$

i.e. the size of $\mathbf{M}_\gamma(z)$. If all $\mathbf{M}_{\gamma,r}(z)$ are parameterized as in Eqs. 3.61 and 3.63 with unit–norm vectors $\mathbf{v}_l$ and $\mathbf{u}_l$ to guarantee that $\tilde{\mathbf{M}}_{\gamma,r}(z)\mathbf{M}_{\gamma,r}(z) = \mathbf{I}_{N_{\gamma,r}}$, also $\mathbf{M}_\gamma(z)$ and $\hat{\mathbf{S}}_\gamma(z)$ will be para–unitary. Depending on the block partitioning $N_{\gamma,r}$ and the order $L_{\gamma,r}$ assigned to each of the blocks $\mathbf{M}_{\gamma,r}(z)$, some of the coefficients of $h_0[k]$ can be forced to be 0. In this way the effective length, i.e. the number of non–zero coefficients of $h_0[k]$, can be adjusted more precisely and is not forced to vary in steps of $\frac{MN}{g^2}$.

Two special cases may be considered : on the one hand, if $N_{\gamma,r} > 1$ and $L_{\gamma,r} = 0$, then $\mathbf{M}_{\gamma,r}(z) = \mathbf{U}$ is an orthonormal matrix. On the other hand, if $N_{\gamma,r} = 1$, only one coefficient can be different from 0 as $\tilde{\mathbf{M}}_{\gamma,r}(z)\mathbf{M}_{\gamma,r}(z) = 1$. Hence,

$$\mathbf{M}_{\gamma,r}(z) = z^{-L_{\gamma,r}}. \tag{3.70}$$

It is best to choose the lowest order in this case and take $L_{\gamma,r} = 0$. If $N_{\gamma,r} = 1$ no optimization is possible however as the corresponding coefficient of $h_0[k]$ is restricted to be 1.

**Example 3.9** In figure 3.6 a typical block partitioning is shown for a DFT modulated subband system with 12 subbands ($M = 12$) and downsampling factor 10 ($N = 10$) . As $g = 2$, $\mathbf{M}_\gamma(z)$ is a $6 \times 5$ block diagonal matrix with $R_\gamma = 3$ blocks of length $N_{\gamma,0} = 3$, $N_{\gamma,1} = 1$ and $N_{\gamma,2} = 2$. △

Given the size $N_{\gamma,r}$ and order $L_{\gamma,r}$ of all lattices and all blocks, and keeping in mind that only the first $\frac{N}{g}$ columns of $\mathbf{M}_\gamma(z)$ define $\hat{\mathbf{S}}_\gamma(z)$ and hence $h_0[k]$, the effective length of the prototype filter $h_0[k]$ can be computed and is given by

$$L_{\text{ef}} = \sum_{\gamma=0}^{g-1} \left( N_{\gamma,\bar{R}_\gamma} Q_\gamma (L_{\gamma,\bar{R}_\gamma} + 1) + \sum_{r=0}^{\bar{R}_\gamma - 1} N_{\gamma,r}^2 (L_{\gamma,r} + 1) \right) \tag{3.71}$$

in which

$$Q_\gamma = \frac{N}{g} - \sum_{r=0}^{\bar{R}_\gamma - 1} N_{\gamma,r} \tag{3.72}$$

and $\bar{R}_\gamma$ is such that

$$\sum_{r=0}^{\bar{R}_\gamma - 1} N_{\gamma,r} < \frac{N}{g} \leqslant \sum_{r=0}^{\bar{R}_\gamma} N_{\gamma,r}. \tag{3.73}$$

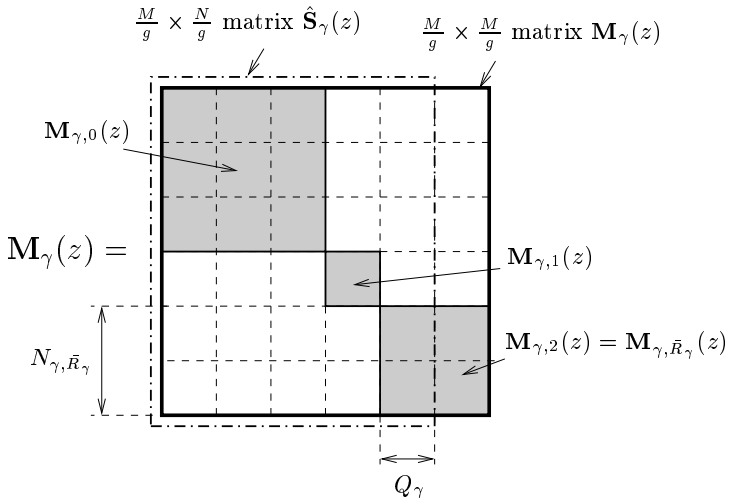


Figure 3.6: Typical block partitioning of $\mathbf{M}_\gamma(z)$ for $M = 12$ and $N = 10$.

**Example 3.10** Consider again the example of figure 3.6. It can be verified that $\bar{R}_\gamma = 2$, $N_{\gamma,\bar{R}_\gamma} = 2$ and $Q_\gamma = 1$. △

Basically all effective filter lengths $L_{\text{ef}}$ can be obtained with $N$ being the minimum. This minimum is reached when all $\mathbf{M}_\gamma(z)$ are equal to the identity matrix. In the case of subband oversampling $L_{\text{ef}}$ can be changed in steps of 2, starting from this minimum $N$, as for $M > N$

$$1 \leqslant \frac{N}{g} < \frac{M}{g}. \tag{3.74}$$

Hence, we can always find a block partitioning $N_{\gamma,r}$ for which $N_{\gamma,\bar{R}_\gamma} = 2$ and $Q_\gamma = 1$ such that $L_{\text{ef}}$ changes in steps of $N_{\gamma,\bar{R}_\gamma} Q_\gamma = 2$ (see Eq. 3.71). The effective filter length can be further refined using $1 \times 1$ para–unitary blocks, as a $1 \times 1$ block defines 1 coefficient, be it restricted to 1. For the special case of critical downsampling ($M = N$), there are $M$ para–unitary blocks $\mathbf{M}_\gamma(z)$ of size $1 \times 1$. As a consequence the prototype filter will consist of $M$ filter coefficients equal to 1 (cf. Eq. 3.70).

Finding the optimal block partitioning $N_{\gamma,r}$ and orders $L_{\gamma,r}$ to obtain a subband system with a desired effective filter length $L_{\text{ef}}$ requires a computationally rather intensive search over all possibilities. Typically, there exist multiple settings $(N_{\gamma,r}, L_{\gamma,r})$ leading to the same $L_{\text{ef}}$. The optimal partitioning should be found that minimizes the number of coefficients forced to 0. Although the number of operations required to implement the filter bank is only determined by the effective length $L_{\text{ef}}$, the number of coefficients forced to 0 should be kept as small as possible. The larger this number, the more memory is required to implement the filter bank.

Furthermore, a large number of inserted zeros or ones restricts the design freedom and makes the optimization of the para–unitary lattices harder. It will in general be more difficult to obtain a good lowpass frequency characteristic and an acceptable stopband level for the filter bank prototype when the number of inserted zeros and ones is large, as these filter coefficients are fixed and cannot be changed by the optimization procedure. Nearly perfect reconstruction filter banks (see chapter 4) may be employed to overcome this problem.

A flow chart of the para–unitary filter bank design method, which was discussed in section 3.3.1–3.3.4, is presented in figure 3.7.

### 3.3.5 Design examples

The above procedure for designing perfect reconstruction DFT modulated filter banks is illustrated by means of two examples.

**Example 3.11** First the $M = 6$, $N = 4$ case is considered (cf. Eq. 3.5, 3.58 and 3.60). As $g = \gcd(6, 4) = 2$, there are 2 para–unitary matrices $\mathbf{M}_0(z)$ and $\mathbf{M}_1(z)$ of size $3 \times 3$ to be optimized (cf. section 3.3.2). In this example $\mathbf{M}_0(z)$ and $\mathbf{M}_1(z)$ were not further split up into block diagonal matrices and hence $R_0 = R_1 = 1$ and $N_{0,0} = N_{1,0} = 3$ (see section 3.3.4 for notation). Both $\mathbf{M}_0(z)$ and $\mathbf{M}_1(z)$ were chosen to be of second order, i.e. $L_{0,0} = L_{1,0} = 2$. As a consequence, the length of the prototype $L_f^a$ is 40 and 36 taps appear to be non–zero as can be verified using Eq. 3.71. Through optimization a lowpass FIR prototype filter $h_0[k]$ is obtained. The frequency response of the analysis prototype $H_0(f)$ is shown in figure 3.8. The relative aliasing level $\eta$ introduced by the analysis (or the synthesis) filter bank, defined as

$$\eta = \sqrt{\frac{\int_{\frac{1}{2N}}^{\frac{1}{2}} |H_0(f)|^2 \, df}{\int_0^{\frac{1}{2N}} |H_0(f)|^2 \, df}} \tag{3.75}$$

is -37.8 dB. The characteristics of the synthesis prototype filter

$$g_0[k] = \frac{1}{M} h_0[L_f^a - 1 - k] \tag{3.76}$$

are similar. By optimizing the parameters $\mathbf{v}_l$ and $\mathbf{u}_l$ following the design procedure of section 3.3 the parameter set converges to a para–unitary filter bank set with a relatively good frequency characteristic, as can be observed from figure 3.8. △

As $R_0 = R_1 = 1$ in this first example the number of coefficients fixed to 0 is fairly small. Hence, a lowpass frequency response could easily be obtained. However, for some applications the stopband energy still remains too high. Nearly perfect reconstruction filter banks may be employed to overcome this problem (see chapter 4).

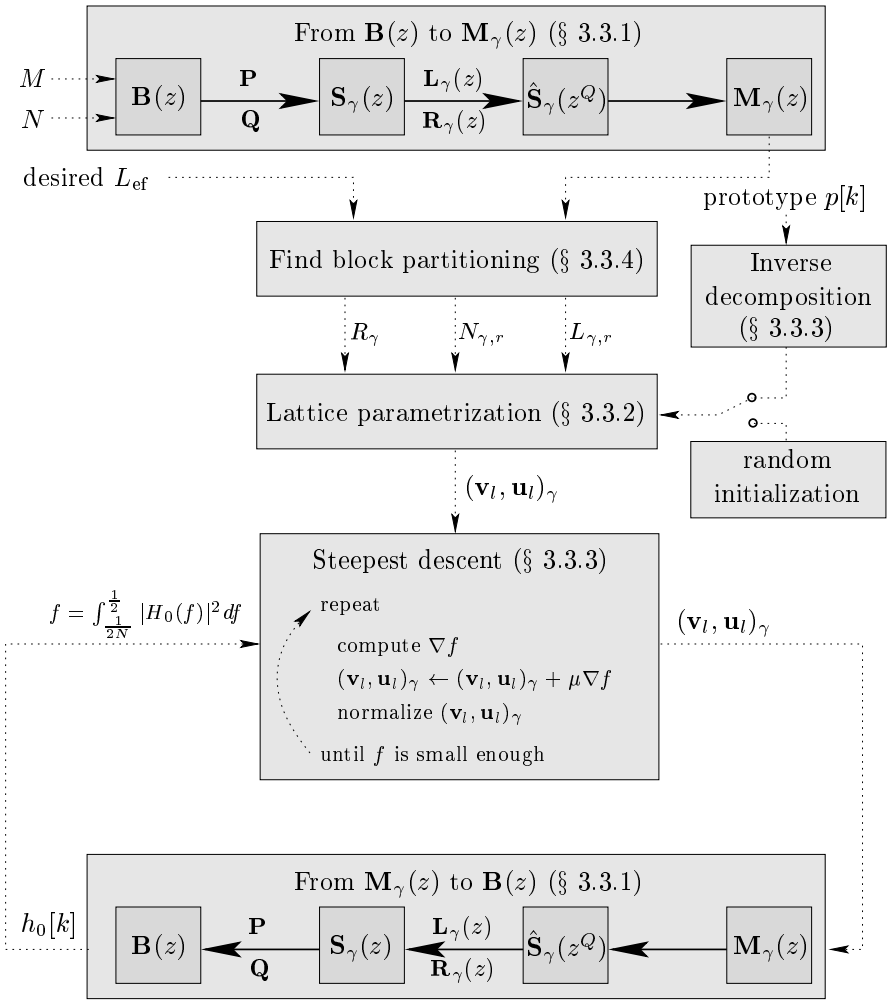


Figure 3.7: Para–unitary filter bank design method : flow chart

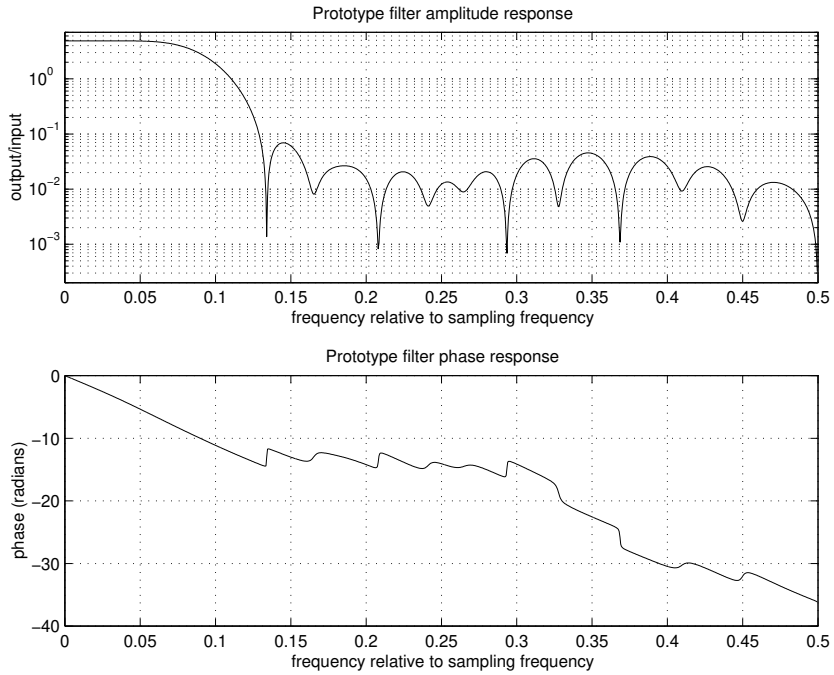


Figure 3.8: Perfect reconstruction oversampled filter bank ($M = 6$, $N = 4$) : prototype frequency characteristic

When a different block partitioning $N_{\gamma,r}$ is chosen that forces more coefficients to zero, the optimization problem will be harder and it will be more difficult in general to obtain a good lowpass characteristic.

**Example 3.12** To illustrate this a second example is considered. A perfect reconstruction subband system was designed with 10 subbands ($M = 10$) and 6–fold subsampling ($N = 6$). The greatest common divisor $g = \gcd(10, 6) = 2$, so there are 2 para–unitary matrices $\mathbf{M}_0(z)$ and $\mathbf{M}_1(z)$ of size $5 \times 5$. The optimal partitioning $N_{\gamma,r}$ of $\mathbf{M}_0(z)$ and $\mathbf{M}_1(z)$ was found such that the effective length of the filters was 77 and the number of filter coefficients fixed to 0 or 1 was minimized. For this $\mathbf{M}_0(z)$ was fully parameterized ($R_0 = 1$) and the number of parameters defining $\mathbf{M}_1(z)$ was reduced by splitting $\mathbf{M}_1(z)$ up as a block diagonal matrix ($R_1 = 2$) with $N_{0,0} = 5$, $N_{1,0} = 1$ and $N_{1,1} = 4$, and $L_{0,0} = 3, L_{1,0} = 0$ and $L_{1,1} = 1$. In this way the effective length of $h_0[k]$ could be fixed to 77. The total length of $h_0[k]$ is 135, of which 76 coefficients can be changed by the optimization procedure and 1 coefficient is fixed to 1. The frequency response of the analysis prototype $H_0(f)$ that was obtained is shown in figure 3.9. The relative aliasing level is -13 dB. The lowpass characteristics of the prototype filter are clearly inferior compared to the

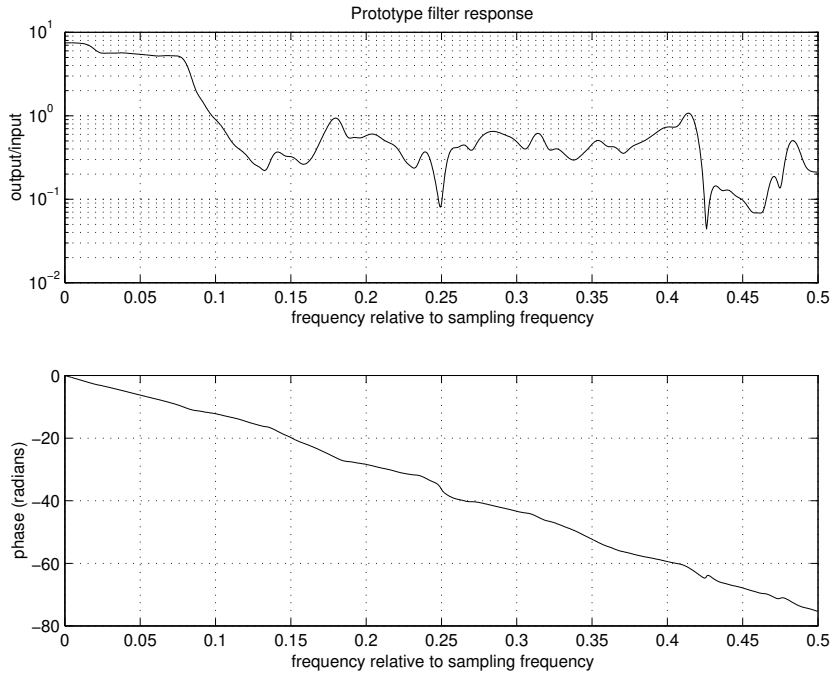


Figure 3.9: Perfect reconstruction oversampled filter bank ($M = 10, N = 6$) : prototype frequency characteristic

first example (figure 3.8). Due to the large number of filter coefficients that cannot be changed by the optimization routine, it is much harder to obtain a satisfactory lowpass behavior. Certainly in this case nearly perfect reconstruction filter banks (see chapter 4) will easily outperform the perfect reconstruction filter bank in terms of frequency selectivity and subband aliasing insertion. △

## 3.4 Conclusions

In this chapter design techniques for perfect reconstruction oversampled DFT modulated filter banks were presented. Following the instruction of [22] it was shown in section 3.1 that oversampled DFT modulated filter banks can be efficiently parameterized by decomposing the polyphase matrix into a structured polynomial matrix and a DFT operation.

In section 3.2 perfect reconstruction was discussed. A condition for perfect reconstruction was formulated in terms of the synthesis polynomial matrix $\mathbf{C}(z)$. We

considered the Smith–McMillan decomposition as a design method for perfect reconstruction filter banks. In practice however, several problems arise. Para–unitary filter bank design turns out to be a better alternative to impose perfect reconstruction.

In section 3.3 para–unitary DFT modulated filter bank design was considered. We discussed a design method that was presented in [22]. It was shown how DFT modulated filter banks can be parameterized and optimized to obtain a perfect reconstruction subband system with good bandpass characteristics. With this approach however the order of the filter banks cannot be adjusted accurately. We presented an extension to this method based on which the parameterization can be further refined. In this way the filter bank length can be better adjusted to the specifications. Further, we showed that based on the inverse parametrization appropriate starting values can be determined, which reduces the optimization time.

In order to fine–tune the effective length of the filter bank prototype some filter coefficients need to be fixed to 0 or 1. By means of some examples it was indicated that if the number of fixed filter coefficients is large, it is in general much harder to obtain a satisfactory lowpass behavior. It will be shown in the next chapter that nearly perfect reconstruction filter banks are a good alternative as they easily outperform perfect reconstruction filter banks in terms of frequency selectivity and subband aliasing insertion.

# Chapter 4

# Nearly Perfect Reconstruction DFT Modulated Filter Bank Design

Contrary to the previous chapter, which deals with perfect reconstruction DFT modulated filter bank design, we discuss nearly perfect reconstruction DFT modulated filter banks in this chapter. It was shown in section 3.3 how the analysis polyphase matrix of a DFT modulated filter bank can be made para–unitary, leading to perfect reconstruction. In section 3.3.5 some design examples were given. The frequency selectivity that can be obtained with these perfect reconstruction filter banks is limited however. It appears that this has a negative impact on the performance of intermediate subband operations such as adaptive filtering.

In section 4.1 and throughout the other sections of this chapter nearly perfect reconstruction filter banks are proposed as an alternative, as they easily outperform perfect reconstruction filter banks in terms of frequency selectivity and subband aliasing insertion.

Section 4.2 presents a frequency–domain design method for nearly perfect reconstruction DFT modulated filter banks. The lowpass properties of the analysis and/or synthesis prototype are traded off against overall amplitude distortion. In order to reduce the complexity of the optimization procedure the amplitude distortion is minimized here instead of checking a true perfect reconstruction condition.

A mixed time/frequency–domain design method for nearly perfect reconstruction

DFT modulated filter banks is proposed in section 4.3. Contrary to the frequency–domain based procedure, true perfect reconstruction can be traded off against frequency selectivity in this case.

In section 4.4 subband adaptive filtering is taken as an example to illustrate that nearly perfect reconstruction filter banks easily outperform perfect reconstruction subband systems without adding severe audible distortion to the near–end signal.

Section 4.5 finally formulates the conclusions.

## 4.1 Nearly perfect reconstruction DFT modulated filter banks

In chapter 3 perfect reconstruction subband systems were discussed. Perfect reconstruction guarantees distortionless input/output behavior : signals pass through the subband system without any quality losses. However, the perfect reconstruction property does not necessarily hold anymore when intermediate operations are performed on the subband signals such as coding or filtering. Furthermore, in practice perfect reconstruction is often cancelled by non–linearities in the signal path coming from A/D–converters and loudspeakers, by injected noise and by algorithmic imperfections (residual errors for instance in the case of adaptive filtering). Perfect reconstruction can therefore be relaxed to 'nearly' perfect reconstruction by admitting some amplitude, phase and aliasing distortion, as long as the distortion is not noticeable or is acceptably low. It depends on the application how far this nearly perfect behavior can divert from the ideal, perfect reconstruction case.

One of the disadvantages of perfect reconstruction filter banks is that the subband filters typically have a non–negligible stopband level. In this way a considerable amount of aliasing is inserted in the subbands, which could have a negative effect on the intermediate operations. A nearly perfect reconstruction filter bank is then more appropriate and leads to better perceptual results as a better filter bank characteristic and hence reduced aliasing levels result in a better performance of the intermediate algorithm. For example, it will be shown in section 4.4 that the performance of subband adaptive filtering schemes can be enhanced if nearly perfect reconstruction filter banks are employed.

Furthermore, in the approach taken in section 3.3 the analysis and the synthesis bank always have the same frequency amplitude characteristic. On the other hand, it is easy to design nearly perfect reconstruction systems for which the analysis and the synthesis filter bank characteristics or the order of the filter banks are different. Both a frequency–domain and a mixed time/frequency–domain based optimization algorithm were developed to design nearly perfect reconstruction subband systems. They will be presented in the next sections.

## 4.2 Frequency–domain optimization

Consider the general subband setup of figure 2.1 and ignore the intermediate processing. It was shown in [156] and in appendix B.3 that the system output $y$ relates to the input $x$ as follows :

$$Y(z) = \underbrace{\frac{1}{N}X(z)\left(\sum_{m=0}^{M-1} G_m(z)H_m(z)\right)}_{\text{amplitude + phase distortion}} + \underbrace{\frac{1}{N}\sum_{n=1}^{N-1} X(ze^{-j\frac{2\pi n}{N}})\left(\sum_{m=0}^{M-1} G_m(z)H_m(ze^{-j\frac{2\pi n}{N}})\right)}_{\text{aliasing distortion}}. \quad (4.1)$$

Perfect reconstruction is obtained if $Y(z) = z^{-\psi}X(z)$, $\psi \in \mathbb{N}$, i.e. if

$$\sum_{m=0}^{M-1} G_m(z)H_m(z) = Nz^{-\psi} \quad (4.2)$$

$$\sum_{m=0}^{M-1} G_m(z)H_m(ze^{-j\frac{2\pi n}{N}}) = 0, \qquad \text{for } n = 1 : N-1 \quad (4.3)$$

are both fulfilled.

In some applications, such as hands–free communication, small amplitude distortion will not lead to significant quality losses and phase distortion is not detrimental, as long as abrupt phase transitions are avoided. It is furthermore observed that the aliasing distortion will be small if the frequency selectivity of the filter banks is sufficiently high. Hence, instead of checking the true perfect reconstruction conditions, i.e. Eqs. 4.2 and 4.3, the lowpass properties of the analysis and/or synthesis prototype on the one hand and the overall amplitude distortion on the other hand can be optimized and traded off. In this way a nearly perfect reconstruction DFT modulated filter bank set can be designed. Instead of the overall amplitude distortion only the distortion from neighboring bands is taken into account in practice.

The frequency–domain cost function to be optimized may then for instance be defined as

$$\min_{H_0,G_0} \int_{\frac{1}{2N}}^{\frac{1}{2}} (\alpha|H_0(f)| + \beta|G_0(f)|)df + \sqrt{\int_0^{\frac{1}{M}} \left(|H_0(f)G_0(f) + H_{M-1}(f)G_{M-1}(f)|^2 - N^2\right)^2 df}, \quad (4.4)$$

where $M$ is the number subbands and $N$ is the subsampling factor (cf. figure 2.1). $H_0(f)$ and $G_0(f)$ are the frequency responses of respectively the analysis and the synthesis prototype, $H_{M-1}(f)$ and $G_{M-1}(f)$ are shifted versions of the prototypes and filter out frequency band $[\frac{1}{2M}, \frac{3}{2M}]$ as in figure 2.2. Finally, $\alpha$ and $\beta$ can be used to weigh the different terms in the cost function with respect to each other. The first term in Eq. 4.4 weighs the stopband level of the analysis and the synthesis prototype. The second term in Eq. 4.4 approximates the amplitude distortion in the interval $[0, \frac{1}{M}]$, taking into account the contributions of band 0 and $M-1$ only. As can be observed from Eq. 4.1 aliasing and phase distortion are not considered here. Further, the factor $N^2$ in Eq. 4.4 ensures that the input–output gain approaches 1 when the subband signals are $N$–fold downsampled.

Starting from a perfect reconstruction filter bank set (designed e.g. based on the procedure of section 3.3), one may want to relax the perfect reconstruction condition and enhance for instance the stopband level of the analysis prototype by minimizing Eq. 4.4 with a large value for $\alpha$.

Another starting point might be a filter bank set, based on an analysis and/or synthesis prototype with a sufficiently low stopband level obtained from a standard filter design tool [126] [134], not fulfilling the perfect reconstruction condition. In this case one may approach perfect reconstruction by changing the prototypes and minimizing the amplitude distortion (take $0 \leqslant \alpha, \beta \ll 1$ and optimize Eq. 4.4).

**Example 4.1** An example of a subband filter bank set, designed using this frequency–domain optimization approach can be found in figure 4.1. The subband system has 8 subbands, which are 6–fold downsampled. The analysis and synthesis bank characteristics are different. The 60–taps analysis prototype was designed with the Parks–McClellan exchange algorithm [134] and has a uniform stopband attenuation. The synthesis filter has 54 taps and was optimized following Eq. 4.4. The relative aliasing level (Eq. 3.75) introduced by the analysis bank is around -46.7 dB. The relative aliasing level of the synthesis bank prototype is -24.3 dB. The maximum amplitude distortion of the analysis–synthesis pair is 4.17% at 0.45 Hz and the maximum phase distortion is 16° at 0.09 Hz. The subband system was tested for audio and speech processing. It was verified that the perceptual distortion is sufficiently small. △

**Example 4.2** In a second example a 6–subband, 4–fold downsampled subband system was designed. The frequency characteristics of the analysis and synthesis prototype are shown in figure 4.2 (compare with figure 3.8). Both the analysis and synthesis prototype have 36 filter taps. The analysis prototype was designed with the Parks–McClellan exchange algorithm. The relative aliasing level (Eq. 3.75) introduced by the analysis bank is -57.6 dB. The relative aliasing level of the synthesis bank is -20.5 dB. The maximum amplitude distortion of the analysis–synthesis pair is 0.21% at 0.25 Hz and the maximum phase distortion is 0.0018° at 0.352 Hz. It was verified that the perceptual distortion is sufficiently small. △

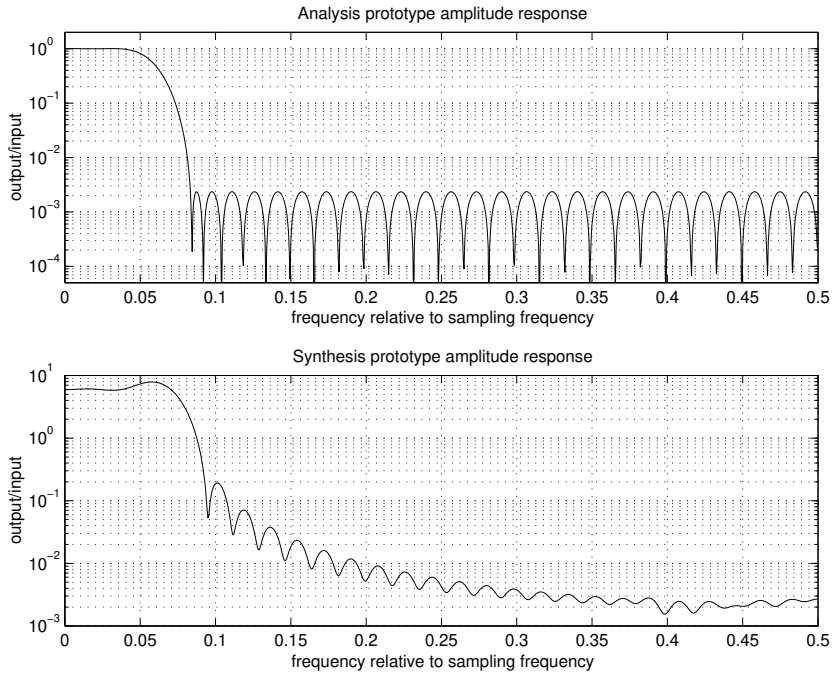


Figure 4.1: An 8–subband 6–fold downsampled nearly perfect reconstruction filter bank set obtained by frequency–domain optimization : prototype frequency characteristics

## 4.3 Mixed time/frequency–domain optimization

Another approach leading to nearly perfect reconstruction systems builds on the relation in the time domain between the subband signals on the one hand and the analysis bank input and the synthesis bank output on the other hand. A similar approach was found in [3].

Consider an $N$–fold downsampled subband system with $M$ subbands (cf. figure 2.1). The input of the subband system $x$ is related to the output $y$ via the analysis and synthesis filter bank. The samples $x[k]$, which are the input to the analysis bank, are put in an infinitely long vector $\mathbf{x}$ in the following way :

$$\mathbf{x} = \begin{bmatrix} \vdots \\ x[k-1] \\ x[k] \\ x[k+1] \\ \vdots \end{bmatrix} . \tag{4.5}$$

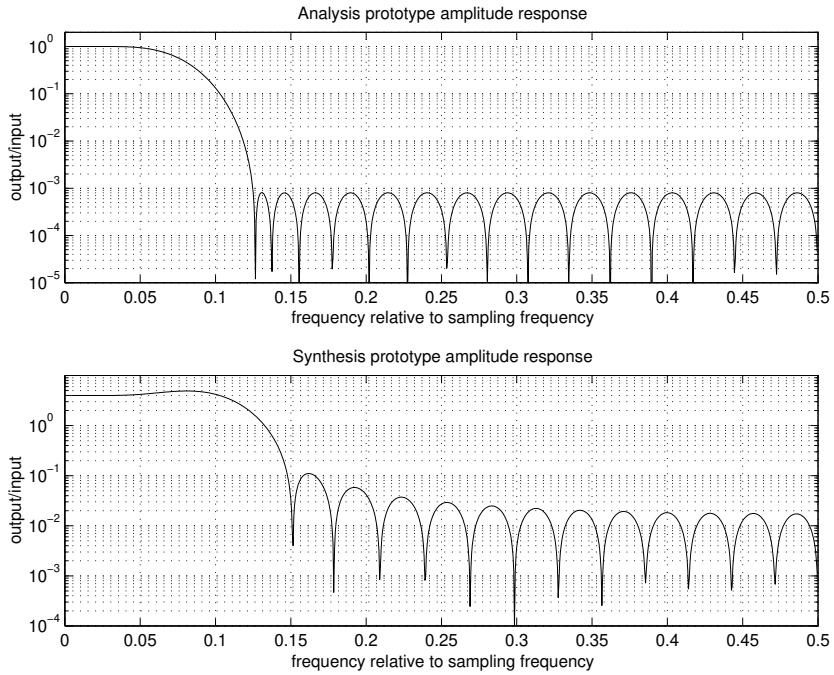


Figure 4.2: A 6–subband 4–fold downsampled nearly perfect reconstruction filter bank set obtained by frequency–domain optimization : prototype frequency characteristics

The downsampled subband samples $x_m[n]$, $m = 0 : M-1$ (see also figure 2.7) can be put in vector $\mathbf{s}$ :

$$\mathbf{s} = \begin{bmatrix} \vdots \\ x_{M-1}[n-1] \\ x_0[n] \\ \vdots \\ x_{M-1}[n] \\ x_0[n+1] \\ \vdots \end{bmatrix}. \tag{4.6}$$

Time–index $n = kN$ corresponds to the lower sampling rate. The relation between $\mathbf{x}$ and $\mathbf{s}$ is expressed as

$$\mathbf{s} = \bar{\mathbf{T}}_H \mathbf{x}, \tag{4.7}$$

in which matrix $\bar{\mathbf{T}}_H$ has a block Toeplitz structure and takes on the form shown in figure 4.3. The bar indicates that the elements appear in time–reversed order.

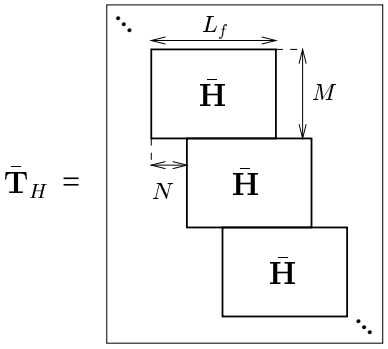


Figure 4.3: $\bar{\mathbf{T}}_H$ visualized

$\bar{\mathbf{H}}$ in figure 4.3 contains the analysis filter bank coefficients :

$$\bar{\mathbf{H}} = \begin{bmatrix} h_0[L_f-1] & \dots & h_0[0] \\ h_1[L_f-1] & \dots & h_1[0] \\ \vdots & \ddots & \vdots \\ h_{M-1}[L_f-1] & \dots & h_{M-1}[0] \end{bmatrix}. \tag{4.8}$$

If $L_f = PN$, with $P \in \mathbb{N}$, the analysis filters can be partitioned in $P$ blocks as shown in figure 4.4 :

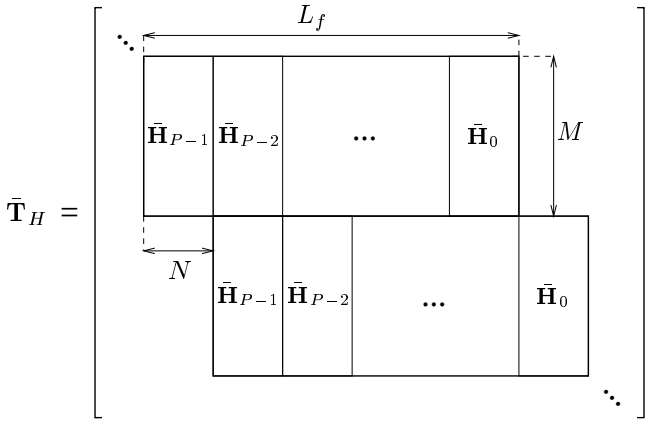


Figure 4.4: Block partitioning of $\bar{\mathbf{T}}_H$

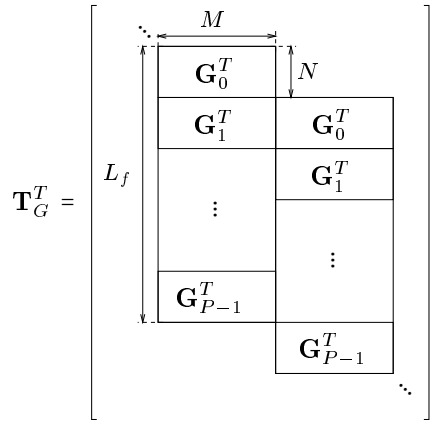


Figure 4.5: Block partitioning of $\mathbf{T}_G^T$

Similar expressions can be given for the synthesis part. An output vector $\mathbf{y}$

$$\mathbf{y} = \begin{bmatrix} \vdots \\ y[k-1] \\ y[k] \\ y[k+1] \\ \vdots \end{bmatrix} \tag{4.9}$$

and a synthesis matrix $\mathbf{T}_G$ are defined such that

$$\mathbf{y} = \mathbf{T}_G^T \mathbf{s}, \tag{4.10}$$

with $\mathbf{T}_G$ having a band diagonal structure containing matrix

$$\mathbf{G} = \begin{bmatrix} g_0[0] & \cdots & g_0[L_f-1] \\ g_1[0] & \cdots & g_1[L_f-1] \\ \vdots & \ddots & \vdots \\ g_{M-1}[0] & \cdots & g_{M-1}[L_f-1] \end{bmatrix} \tag{4.11}$$

with the synthesis filter bank coefficients. For the sake of conciseness, the analysis and synthesis filters are assumed to have equal length. If not, $\mathbf{H}$ or $\mathbf{G}$ should be padded with zeros. Partitioning $\mathbf{T}_G$ in the same way as $\mathbf{T}_H$, using blocks $\mathbf{G}_p$, leads to figure 4.5.

By combining Eq. 4.10 with 4.7 one obtains

$$\mathbf{y} = \mathbf{T}_G^T \bar{\mathbf{T}}_H \mathbf{x}. \tag{4.12}$$

Hence, the perfect reconstruction property follows from the inner matrix product $\mathbf{T}_G^T \bar{\mathbf{T}}_H$. Taking into account the filter partitioning, $\mathbf{T}_G^T \bar{\mathbf{T}}_H$ can be computed based on the block convolution of

$$\mathbf{V} = \begin{bmatrix} \mathbf{G}_0^T & \mathbf{G}_1^T & \dots & \mathbf{G}_{P-1}^T \end{bmatrix} \star \begin{bmatrix} \bar{\mathbf{H}}_0 & \bar{\mathbf{H}}_1 & \dots & \bar{\mathbf{H}}_{P-1} \end{bmatrix} \tag{4.13}$$

i.e. (assuming that $\bar{\mathbf{H}}_p = \mathbf{G}_p = \mathbf{0}$, if $p < 0$ or $p \geqslant P$)

$$\mathbf{V}_q = \sum_{p=0}^{P-1} \mathbf{G}_{P-1-p}^T \bar{\mathbf{H}}_{p+q}. \tag{4.14}$$

Perfect reconstruction is obtained if

$$\mathbf{V}_0 = \mathbf{I}_N \tag{4.15}$$

$$\mathbf{V}_q = \mathbf{0}, \qquad \forall q \in \mathbb{Z}_0. \tag{4.16}$$

In the case of a para–unitary DFT modulated filter bank for which $L_f = PN$, Eqs. 3.47, 3.48, 3.53 and 3.54 hold. It can be proven (see appendix B.8) that Eqs. 4.15 and 4.16 are fulfilled in this case.

A modulated filter bank can now be designed that optimally meets conditions 4.15 and 4.16. A trade–off between these conditions and the lowpass properties of the analysis and/or the synthesis prototype filter should lead to a nearly perfect reconstruction subband system with a good stopband rejection. The mixed time/frequency–domain cost function to be optimized may be defined as

$$\min_{H_0, G_0} \int_{\frac{1}{2N}}^{\frac{1}{2}} (\alpha |H_0(f)| + \beta |G_0(f)|) df + \left\| \begin{matrix} \mathbf{W}_{-P+1} \\ \vdots \\ \mathbf{W}_{P-1} \end{matrix} \right\|_2, \tag{4.17}$$

with $\mathbf{W}_0 = \mathbf{V}_0 - \mathbf{I}_N$ and $\mathbf{W}_q = \mathbf{V}_q$ if $q \neq 0$. Finally, $\alpha$ and $\beta$ are weighting factors taking into account the stopband level of the analysis and synthesis prototype filter.

The first term of Eq. 4.17 can be extended so that the filter banks do not have to be modulated. In this way however the optimization criterion becomes more complex and the optimization time is increased.

The mixed time/frequency–domain method is more advanced than the frequency–domain approach of section 4.2 as it trades off true perfect reconstruction against frequency selectivity and does not take into account amplitude distortion of neighboring bands only.

**Example 4.3** Figure 4.6 (compare with figure 3.8 and 4.2) shows the analysis and synthesis prototype filters for a 6–subband, 4–fold downsampled subband system. Both the analysis and synthesis prototype have 36 filter taps. The analysis proto-

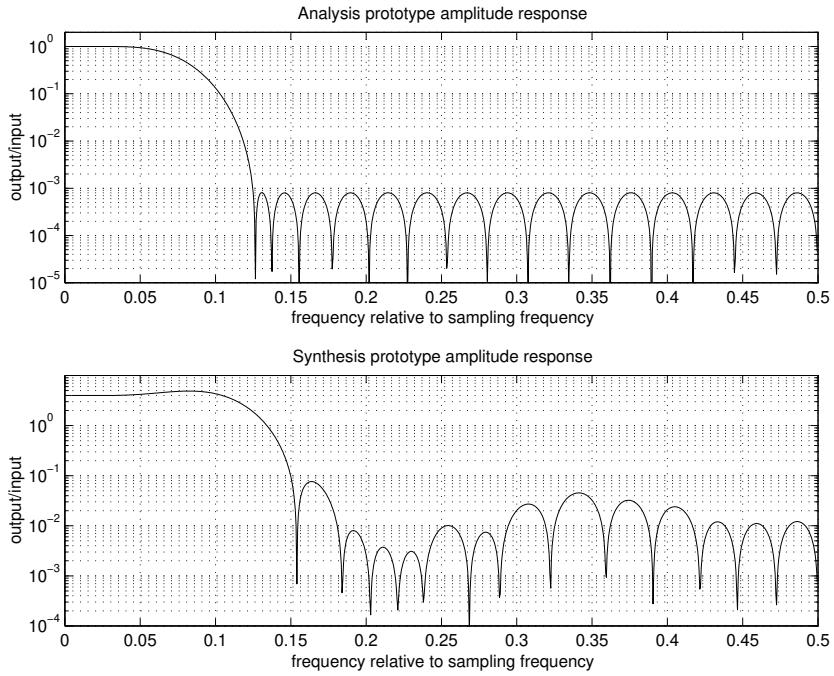


Figure 4.6: A 6–subband 4–fold downsampled nearly perfect reconstruction filter bank set obtained by mixed time/frequency–domain optimization : prototype frequency characteristics

type was designed with the Parks–McClellan exchange algorithm and is the same as in figure 4.2. The relative aliasing level (Eq. 3.75) introduced by the analysis bank is -57.6 dB. The relative aliasing level of the synthesis bank is -19.6 dB. The maximum amplitude distortion of the analysis–synthesis pair is 0.38% at 0.211 Hz and the maximum phase distortion is 0.0023° at 0.319 Hz. The perceptual distortion was verified to be sufficiently small. △

**Example 4.4** In example 3.12 a perfect reconstruction subband system was designed with 10 subbands and 6–fold subsampling. Due to the large number of fixed filter coefficients the optimization routine outputs a filter bank with a limited frequency selectivity (see figure 3.9). Nearly perfect reconstruction techniques can be employed to design a higher frequency selective filter bank. A 77–taps analysis prototype filter was designed using a simple FIR design toolbox. The relative aliasing level (Eq. 3.75) introduced by the analysis bank is -54.9 dB. Using the mixed time/frequency–domain approach explained in this section a 78–taps synthesis prototype filter was designed. The relative aliasing level of the synthesis bank is -48.5 dB. The frequency characteristics of the analysis and synthesis prototype filter are

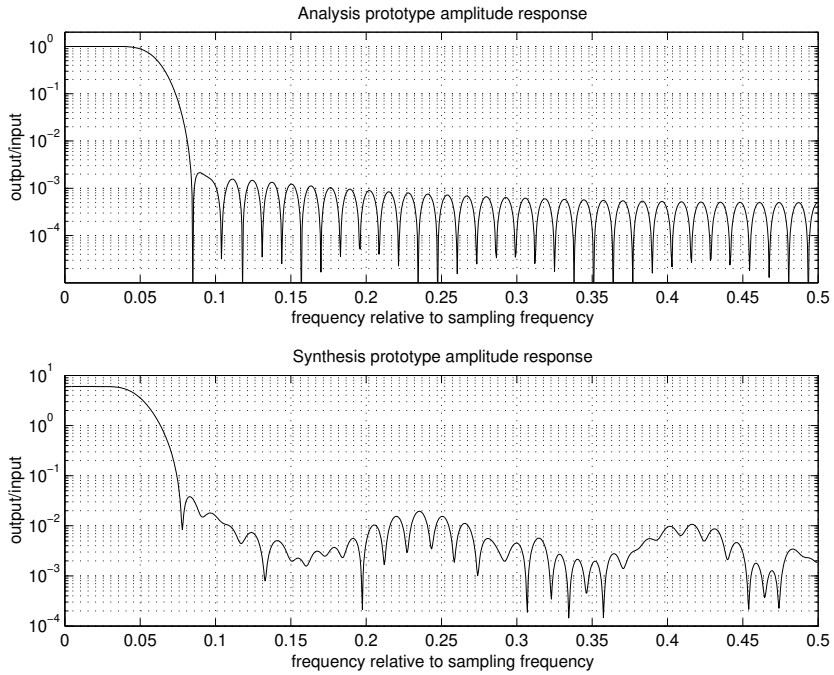


Figure 4.7: A 10–subband 6–fold downsampled nearly perfect reconstruction filter bank set obtained by mixed time/frequency–domain optimization : prototype frequency characteristics

shown in figure 4.7. The maximum amplitude distortion of the analysis–synthesis pair is 0.17% at 0.26 Hz and the maximum phase distortion is 0.09° at 0.46 Hz. The perceptual distortion was verified to be sufficiently small. The filter bank set presented in figure 4.7 clearly outperforms the perfect reconstruction system of figure 3.9. △

## 4.4 Application

One of the disadvantages of perfect reconstruction filter bank design is that the stopband level of the analysis prototype filter is difficult to control and to reduce to a sufficiently low level. The algorithms presented in this chapter offer more design freedom. As a result, it is easier to design and to specify the filter bank characteristics. In this way the amount of aliasing inserted in the subbands for instance can be kept small. This could have a positive impact on the operations performed on the subband signals, as will be illustrated in this section, taking

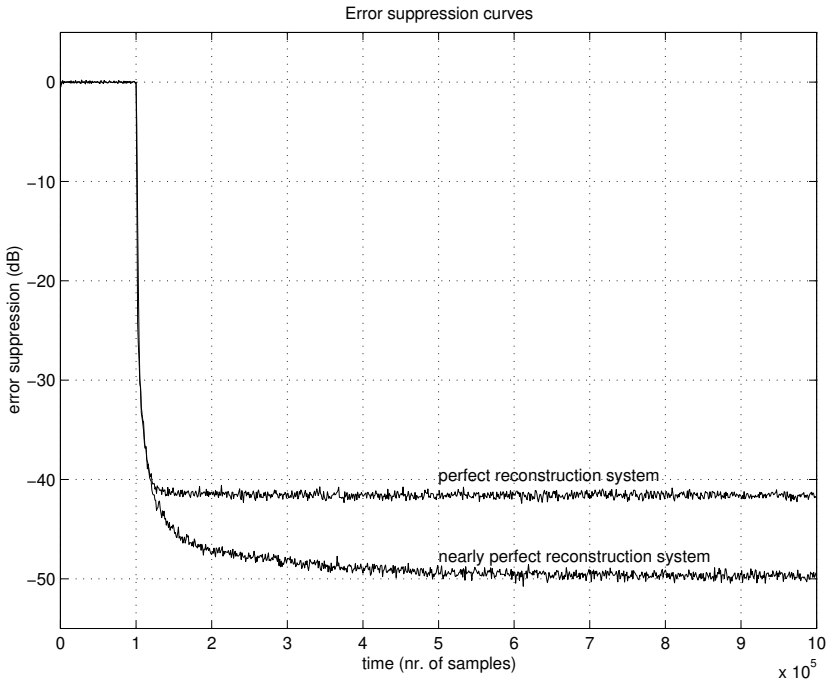


Figure 4.8: Simulation example : convergence curves for 2 subband adaptive systems ($M = 6, N = 4$), one based on a perfect reconstruction filter bank set, the other on nearly perfect reconstruction filter banks

subband adaptive filtering as an example. The general setup for a subband adaptive system is shown in figure 2.9. More information on subband adaptive filtering can be found in section 2.3.2 and in part II of the thesis.

**Example 4.5** An experiment was carried out in which a Gaussian white noise signal $x$ was fed through a random 60–taps linear FIR filter $w$. There was no near–end signal $s$ added to $d = w \star x$ in this experiment. A subband system as shown in figure 2.9 was used to estimate the unknown $w$. In this example $M = 6$ and $N = 4$.

In a first simulation the perfect reconstruction DFT modulated filter bank set of order 35, described in example 3.11 and shown in figure 3.8, was used. The adaptive filters have length 25 each. The error output of the algorithm is plotted in figure 4.8. Apparently some residual error remains. This is a well–known phenomenon and has to do with injected subband aliasing on the one hand and is related to an undermodelling problem that occurs with subband adaptive filtering on the other hand. More information on the residual error can be found in chapter 5.

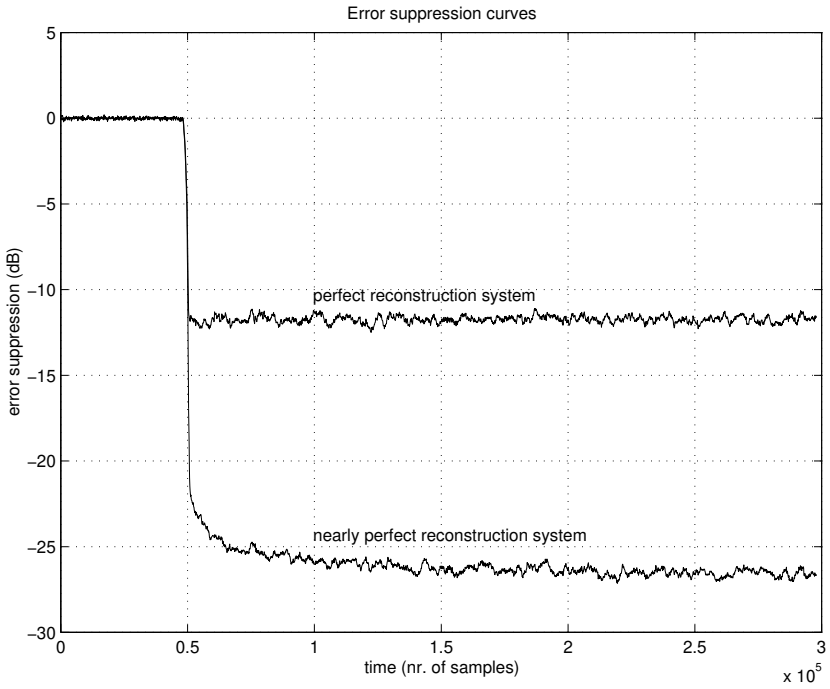


Figure 4.9: Simulation example : convergence curves for 2 subband adaptive systems ($M = 10, N = 6$), the curve at the top corresponding to a perfect reconstruction filter bank set, the bottom curve to a nearly perfect reconstruction system

In a second simulation the nearly perfect reconstruction filter banks, obtained by the frequency–domain and mixed time/frequency–domain optimization procedure of section 4.2 and 4.3 (see example 4.2, example 4.3, figure 4.2 and figure 4.6), were applied. The order of the filter banks and the length of the adaptive filters were the same as in the first simulation. As the characteristics of the mixed time/frequency–domain and frequency–domain based filter banks are similar, the error curves lie almost on top of each other. Hence, only the error curve corresponding to the mixed time/frequency–domain based filter bank set is shown in figure 4.8. It appears that with the nearly perfect reconstruction system a higher error suppression is obtained. △

**Example 4.6** In a second experiment again a Gaussian white noise signal $x$ was applied to the input of a 60–taps linear FIR filter $w$, with no near–end signal $s$ added to $d = w \star x$ (for signal conventions, see figure 2.9). In this second example $M = 10$ and $N = 6$. The perfect reconstruction DFT modulated filter bank set of example 3.12 and figure 3.9, was applied here, resulting in the upper curve of figure 4.9. Further, the filter banks designed in example 4.4 and figure 4.7 were

employed to illustrate the effect of nearly perfect reconstruction. The error output of the nearly perfect reconstruction algorithm is also plotted in figure 4.9 and corresponds to the bottom curve. Both for the perfect reconstruction as for the nearly perfect reconstruction system the length of the adaptive filters was taken equal to 15 and 5 anti–causal taps (see chapter 5) were added to obtain a satisfactory noise suppression. Also in this case the nearly perfect reconstruction system is superior to its perfect reconstruction counterpart. △

It appears that although the maximum convergence rate that can be reached may be somewhat slower, a lower error energy and hence a better modelling can be achieved with the nearly perfect reconstruction systems. Apparently the reduced stopband level of the nearly perfect reconstruction filter banks leads to a better modelling of the unknown system.

The number of perfect reconstruction filter bank coefficients fixed to 0 was small in the first example. Hence, a perfect reconstruction filter bank with a relatively good stopband attenuation could be designed (see also section 3.3.5). If more coefficients are fixed to 0, worse filter bank characteristics and higher aliasing levels are obtained. In that case the nearly perfect reconstruction system easily outperforms the perfect reconstruction scheme (figure 4.9).

Example 4.5 and 4.6 prove that thanks to their better filter bank characteristics nearly perfect reconstruction filter banks can outperform perfect reconstruction systems without introducing unacceptable audible distortion.

## 4.5 Conclusions

In this chapter nearly perfect reconstruction DFT modulated analysis/synthesis filter bank sets were designed. In this way it is prevented that too much inter–subband aliasing is inserted during the downsampling operation. Inter–subband aliasing has a negative impact on possible intermediate operations performed on the subband signals (adaptive filtering for instance).

In section 4.2 a frequency–domain optimization design algorithm was presented for nearly perfect reconstruction subband systems. The frequency selectivity of the filter banks is traded off against the amplitude distortion between two neighboring bands.

A mixed time/frequency–domain design procedure was proposed in section 4.3. Here, the frequency selectivity of the filter banks and a measure for the perfect reconstruction of the subband system are weighed and traded off against each other.

In section 4.4 subband adaptive filtering was taken as an example to illustrate and compare the different design techniques. The examples proved that thanks to their

better frequency characteristics nearly perfect reconstruction systems outperform their perfect reconstruction counterparts without introducing unacceptable audible distortion.

# Part II

# Subband and Frequency–Domain Adaptive Filtering

In the second part of the thesis subband and frequency–domain adaptive filters are studied. These multirate schemes are a good alternative to standard fullband algorithms if long FIR filters need to be adapted. Thanks to the use of subband or frequency–dependent stepsizes and multirate techniques improved convergence behavior and a reduced computational complexity can be obtained.

The combination of standard adaptive filtering schemes with filter bank techniques gives rise to a promising adaptive system, known as the subband adaptive filter, which is studied in **chapter 5**. A comparison is made between the subband approach and standard fullband adaptive filters. It will appear that the increase in performance and the complexity reduction w.r.t. the fullband approach are less than expected. Nevertheless, a considerable cost reduction can be obtained. It is further shown that a substantial residual undermodelling error appears if the length of the subband adaptive filters is not chosen appropriately. Next, three design criteria are formulated, which are necessary conditions for proper operation of the subband adaptive filter.

Frequency–domain adaptive filtering techniques are studied in **chapter 6**. It appears that this class of adaptive algorithms, which is known from the literature for several years now, outperforms standard subband systems in terms of convergence behavior and modelling capabilities. It will be shown that frequency–domain adaptive filters can be considered as a special subband adaptive system that relies on an overlap–save based error correction mechanism and that does not suffer from residual errors. It is further shown that two out of the three design criteria for subband adaptive systems specified in chapter 5 are fulfilled by the frequency–domain adaptive filter.

It is observed that frequency–domain adaptive algorithms have a desired convergence behavior thanks to the overlap–save error correction. In **chapter 7** an alternative fullband error adaptation scheme will be presented. In this way improved performance is obtained. It is shown that the fullband error adaptation algorithm can be considered as an extension of the frequency–domain error correction mechanism to a more general class of subband adaptive filters.

# Chapter 5

# Subband Adaptive Filtering

Multirate schemes such as the subband adaptive filter have been a topic of interest for many years now [142]. They are employed to identify high–order FIR systems and are a promising alternative to the classical LMS algorithm. For this kind of application the LMS adaptive filter is less attractive as it has a larger complexity and its convergence behavior is generally worse.

It is often claimed that subband processing leads to a considerable complexity reduction w.r.t. the fullband approach, offering a complexity gain equal to the number of subbands in the limiting case where the subbands are maximally, i.e. critically downsampled. Furthermore, better performance is expected owing to the fact that after appropriate subsampling each subband signal will have a flatter spectrum than the fullband signal.

In section 5.1 a general subband adaptive filtering setup is presented. A first comparison is made with the standard fullband approach, mentioning some of the (dis)advantages of subband processing. Next, it is briefly discussed how the filter banks should be selected.

Three design criteria for subband adaptive systems, which deal with frequency selectivity, perfect reconstruction and perfect path modelling, are formulated in section 5.2. These conditions are necessary requirements for the satisfactory performance of the subband adaptive filter.

Section 5.3 discusses the effect of the downsampling operation. Thanks to the downsampling a significant reduction of the implementation cost can be obtained. The side–effect of this intrinsically nonlinear operation is the insertion of aliasing in the subbands, which has a negative impact on the convergence of the adaptive filters. As a consequence, critically subsampled adaptive filtering schemes, although leading to the largest computational savings, are not be taken into consideration

for subband adaptive processing.

Next it is shown in section 5.4 that a residual undermodelling error will remain unless infinitely long adaptive filters are employed in each subband. It will be shown that the adaptive filter has to model (many) more coefficients than usually expected and that extra filter taps have to be added both in the causal as well as in the anti–causal direction. Two design methods will be presented based on which the appropriate subband filter order can be estimated.

It will be shown in section 5.5 that the expected cost reduction w.r.t. the fullband approach is often overestimated : in order to reduce the residual error more subband filter coefficients need to be adapted than usually expected. Furthermore, as critically subsampled schemes suffer from aliasing, oversampled subband schemes are called for, which are less attractive from a computational complexity point of view. Although the complexity gain w.r.t. the fullband approach is less than expected, still a considerable cost reduction can be obtained.

The conclusions are formulated in section 5.6.

## 5.1 Subband adaptive systems

Subband adaptive filtering schemes have been a topic of interest for many years now. They are employed to identify long FIR systems as they provide a reduced complexity solution for high–order problems and in this way outperform the standard fullband adaptive approaches.

### 5.1.1 General subband adaptive filtering setup

A general subband adaptive filtering system is shown in figure 5.1. Both input signals $x$ and $d$ are fed into identical $M$–band analysis filter banks, with $d$ being a filtered version of $x$ by the unknown system $w[k]$. In most applications a so–called near–end signal $s$ is added to $w \star x$ such that $d = s + w \star x$. The ultimate goal is to suppress $w \star x$ at the output $e$ and to retain a non–distorted version of $s$.

After $N$–fold subsampling, adaptive filtering is done in each subband separately. Basically any kind of adaptation algorithm (see section 2.3.1) can be employed for the update. It is however common to use (N)LMS to adapt the subband filters $f_m[k]$. Remark that in contrast to classical adaptive filtering structures, in this setup $w[k]$ is estimated using a set of parallel, independently adapted filters $f_m[k]$.

The outputs of the subband adaptive filters are recombined in the synthesis filter bank leading to the final output $e$. The ideal frequency amplitude characteristics of the analysis filters $H_m(z)$ and synthesis filters $G_m(z)$ are also shown (ideal bandpass filters).

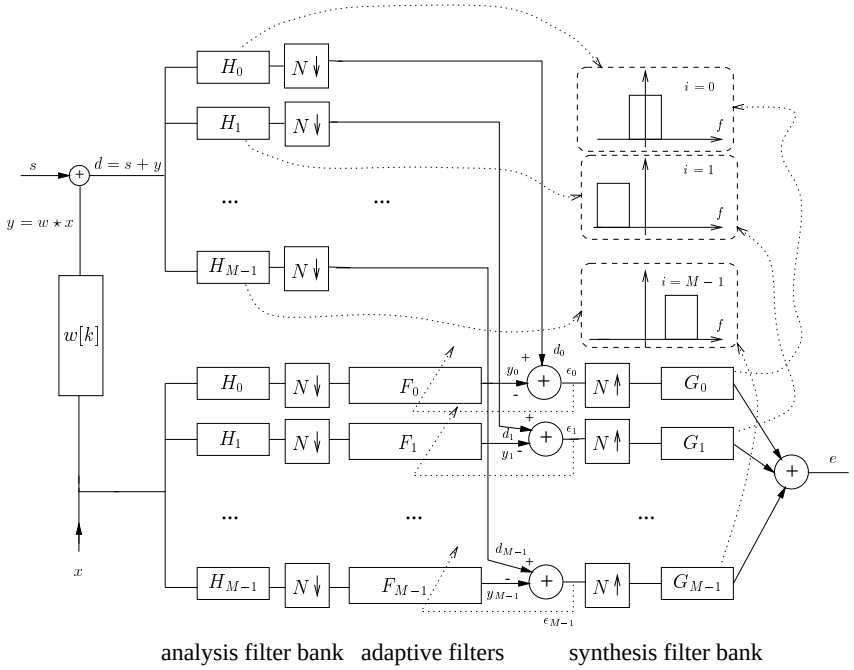


Figure 5.1: General subband adaptive filter with ideal filter banks

### 5.1.2 Subband versus fullband adaptive filtering

By splitting signals into subbands and using subsampling techniques faster (initial) convergence and better tracking properties are hoped for. Subband processing seems very promising since for colored input signals with a large eigenvalue spread such as speech, fullband adaptation algorithms like the LMS algorithm show slow convergence (see also section 2.3.1). In the subband case the subband signals will have a flatter frequency amplitude spectrum. The stepsizes of the subband adaptive filters can then be fine–tuned per subband, in general leading to improved convergence behavior.

A second advantage of subband systems over classical fullband adaptation is the lower implementation cost thanks to the downsampling (see also section 5.5.1). Optimal computational savings are expected whenever the signals are maximally, i.e. critically downsampled. In this way subband adaptive systems can outperform their frequency–domain counterparts for which two–fold oversampling is typically used (see chapter 6).

Unfortunately, this picture of the subband approach is certainly too optimistic. First, the achievable cost reduction w.r.t. fullband adaptive filtering is typically

less than expected. Further, —as will be illustrated further on in this chapter— various effects occur, which affect the performance. A reduced modelling capability is observed through the appearance of residual errors and convergence is typically slowed down by inter–subband aliasing.

### 5.1.3 Filter bank selection

The performance of the subband adaptive system depends (partly) on the quality of the filter banks. Hence, the filter banks should be chosen with care. A discussion on filter banks and filter bank design can be found in section 2.2. Filter banks however introduce aliasing, an inherent delay, extra cost and signal distortion. Aliasing will cause the convergence of the adaptive filters to degrade, which is not desirable.

Oversampled subband systems are most appropriate as aliasing insertion can be reduced in that way (see also section 5.3.1). It is further recommended to employ DFT modulated filter banks in the oversampling case. It was already suggested in section 2.2.2 that DCT modulated filter banks are less suitable if non–critically subsampled schemes are preferred. Cosine modulated filters have a positive and negative frequency band, which tend to overlap when they are not–critically downsampled, resulting in an unacceptable amount of aliasing insertion.

At first sight it is a pity that DFT instead of DCT modulated filter banks have to be employed : contrary to their DCT counterparts DFT modulated filters are complex–valued, which requires more complicated arithmetic.

However, looking in more detail into this, it is observed that a DFT modulated filter bank does —roughly speaking— not lead to a more expensive implementation than a cosine modulated bank. If the input signals $x$ and $d$ and the unknown system $w[k]$ are real, as is the case in the applications envisaged in this thesis, only half of the complex bands need to be processed as the other half are pairwisely complex conjugated.

Now, compare a DFT modulated with a DCT modulated filter bank having an equal number of processed bands. In the DFT modulated case, $M$ processed bands correspond to about $2M$ filters modelling the unknown system $w[k]$, as the bands are pairwisely complex conjugated. Hence, they can be downsampled at most $2M$ times. If the unknown system $w[k]$ has length $L_{FB}$, typically $L_{SB}^{\text{DFT}} = \frac{L_{FB}}{2M}$ taps are chosen per subband filter. If we now assume that a linear adaptation algorithm such as LMS is employed[1] and that $f_s$ is the sampling frequency corresponding to $x$ and $d$, the cost for the adaptation of $M$ filters is $16M L_{SB}^{\text{DFT}} \frac{f_s}{2M} = 4\frac{L_{FB} f_s}{M}$ `ops`.

For the corresponding cosine modulated bank on the other hand $w[k]$ is modelled by $M$ subbands, which are all processed. Hence, they can be maximally decimated only $M$ times. The number of filter taps equals $L_{SB}^{\text{DCT}} = \frac{L_{FB}}{M}$, the total cost being

[1] The cost for adapting an $L$–taps filter using the LMS algorithm is around $4L$ for real arithmetics and around $16L$ for complex arithmetics. See also section 2.3.1.

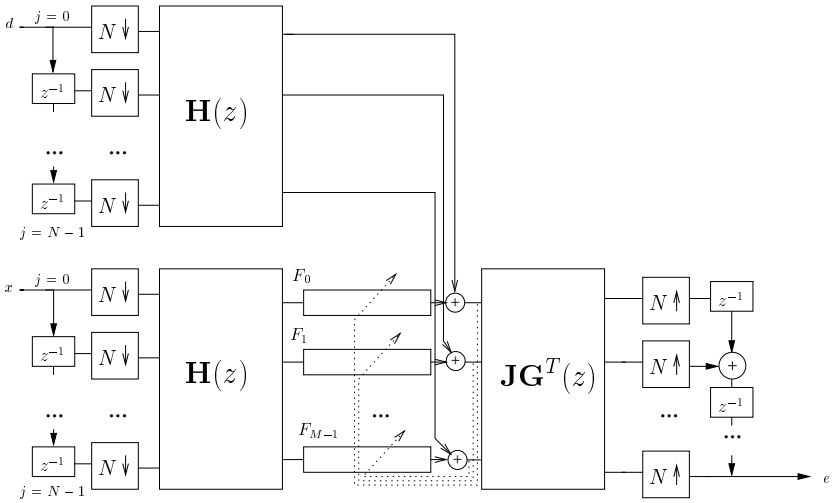


Figure 5.2: Subband adaptive filter : polyphase implementation

$4ML_{SB}^{\mathrm{DCT}}\frac{f_s}{M} = 4\frac{L_{FB}f_s}{M}$ ops. Apparently this compensates more or less for the extra factor 4, which is needed for complex arithmetic.

In the literature many references can be found on subband adaptive filtering. For more information, implementations and design suggestions we refer to [3] [13] [14] [15] [26] [27] [70] [71] [79] [90] [99] [109] [117] [119] [130] [132] [152] [177].

### 5.1.4 Polyphase implementation

Observe from figure 5.1 that the analysis filters are immediately followed by downsamplers. Hence, it is cheaper to do all filtering operations at the downsampled rate. Figure 5.1 can be re–arranged resulting in figure 5.2. $\mathbf{H}(z)$ and $\mathbf{G}(z)$ are respectively called the analysis and synthesis polyphase matrix. We refer to Eqs. 2.17, 2.19, section 2.2.3 and figure 2.7 for more information.

### 5.1.5 DFT modulated subband adaptive filters

It was motivated in section 5.1.3 that DFT modulated filter banks are most appropriate for subband adaptive filtering applications. DFT modulated filter banks may be implemented efficiently using polyphase decomposition and fast signal transforms. In section 3.1 a general framework for oversampled DFT modulated subband systems was discussed : a DFT modulated analysis filter bank with subsequent $N$–fold downsampling can be implemented as a tapped delay line of size $N$ with $N$–fold downsampling, followed by a structured $M \times N$ matrix $\mathbf{B}(z)$, containing polyphase

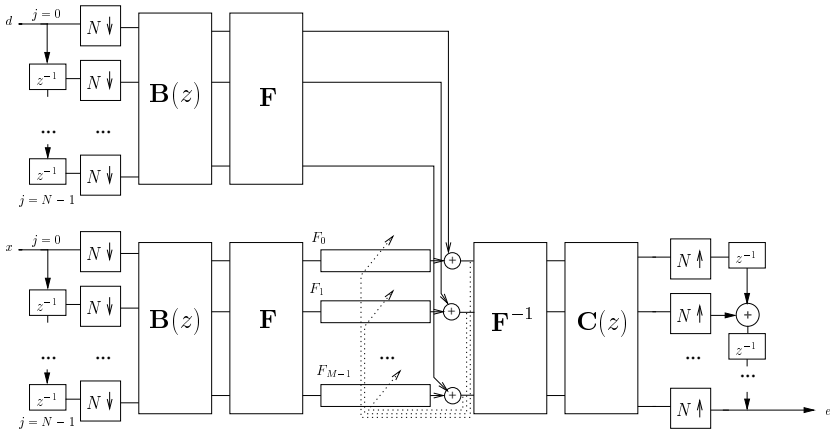


Figure 5.3: DFT modulated subband adaptive filter

components of the analysis prototype $h_0[k]$, and an $M \times M$ DFT matrix $\mathbf{F}$. The synthesis bank is constructed in a similar fashion with an $N \times M$ matrix $\mathbf{C}(z)$. Hence, figure 5.2 can be redrawn resulting in figure 5.3.

## 5.2 Design criteria for subband adaptive systems

In this section three design criteria for subband adaptive systems are formulated, which should lead to subband adaptive filters exhibiting better properties, similar to what can be obtained with frequency–domain techniques.

It will be shown in chapter 6 that frequency–domain adaptive algorithms can be considered as a special kind of subband adaptive filtering, typically relying on simple filter banks and often using two–fold oversampling. Most subband adaptive systems perform better on this part as their oversampling factors are normally less than 2 and more advanced filter banks are employed. A better performance at a lower cost is therefore expected from these true subband adaptive filtering structures. However, due to their error correction mechanism frequency–domain adaptive filters exhibit superior performance and outperform most subband structures, sometimes even at a lower cost (see chapter 6).

Bringing the ideas behind the subband and the frequency–domain algorithms together into a "hybrid" approach and trying to transplant and extend the mechanisms behind the frequency–domain algorithms to the broader class of subband adaptive filters should ultimately lead to a better overall performance of subband adaptive systems.

In this section therefore three design criteria are defined. These conditions are necessary requirements for the satisfactory performance of the subband adaptive filter. An attempt to extend the frequency–domain error correction mechanism to true subband adaptive filtering is treated in chapter 7.

An analysis/synthesis filter bank set is (preferably) designed such that three design criteria are met. First of all, the filter banks have to be frequency selective. The second condition is about perfect reconstruction and the third requirement deals with perfect path modelling. In the following sections each of these conditions are discussed in more detail.

### 5.2.1 Frequency selectivity

In order to reduce the amount of aliasing components that are inserted in the subbands the analysis filters should be frequency selective. Inter–subband aliasing has an inhibitive effect on the convergence of the adaptive filters, as will be shown in section 5.3.

The residual error, which occurs in any standard subband adaptive setup, then basically consists of four parts : an excess mean–squared error because of near–end source activity, an excess mean–squared error due to aliasing components, an undermodelling error and finally a so–called "lag error" as the adaptive system fails to track time–varying setups and will lag behind.

At first sight, the frequency characteristics of the synthesis bank do not seem to have an influence on the convergence of the subband adaptive filters (see figure 5.1 or 5.3). In chapter 7 however, an alternative adaptation scheme will be introduced for which the adaptation of the subband filter coefficients does depend on the synthesis filter bank. In that case the characteristics of the synthesis bank may be important.

Highly frequency selective filter banks introduce a substantial processing delay. This puts a constraint on the frequency selectivity, and hence on the downsampling factor $N$ as delay constraints make subband schemes unattractive. On the other hand, computational savings appear to be more or less proportional to $N$.

### 5.2.2 Perfect reconstruction

Both signal $x$ and $d = s + w \star x$ pass through the filter bank system (see figure 5.1). Ideally, signal $x$ is completely suppressed by the adaptive filtering operations and the desired part $s$ of signal $d$ passes to the output of the subband scheme without distortion, being an exact (delayed) copy of $s$. To minimize the distortion of $s$ (nearly) perfect reconstruction filter banks are called for. Remark that overall aliasing cancellation does not exclude aliasing insertion in the individual subbands, which leads to degraded convergence behavior of the adaptive filters.

Perfect reconstruction is obtained when (see figure 5.2)

$$\mathbf{J}\mathbf{G}^T(z)\mathbf{H}(z) = z^{-\delta}\mathbf{I}_N, \qquad \delta \in \mathbb{N}. \tag{5.1}$$

The smallest $\delta$ is determined such that Eq. 5.1 is fulfilled and such that both the analysis and synthesis filters are causal and hence implementable. It can be shown that if Eq. 5.1 holds the perfect reconstruction conditions Eqs. 4.2 and 4.3 are fulfilled for $\psi = N - 1 + \delta N$.

In the case of DFT modulated filter banks the perfect reconstruction property reduces to (see figure 5.3 and Eq. 3.32)

$$\mathbf{C}(z)\mathbf{B}(z) = z^{-\delta}\mathbf{I}_N, \qquad \delta \in \mathbb{N}. \tag{5.2}$$

The design of critically downsampled perfect reconstruction filter banks is well–known and is treated in the literature [156] [170] [181]. However, as discussed in section 5.3.1 these maximally decimated filter banks are not appropriate for subband adaptive filtering applications. So, oversampled subband systems should be called for.

Some references on *perfect reconstruction* design for oversampled DFT modulated filter banks can be found in more recent publications [22] [89] [168]. A discussion on perfect reconstruction oversampled filter bank design can also be found in chapter 3. Both the Smith–McMillan decomposition (section 3.2.1) and a para–unitary filter bank design method (section 3.2.2 and 3.3) were discussed.

Condition 5.1 can be relaxed to

$$\mathbf{J}\mathbf{G}^T(z)\mathbf{H}(z) \approx z^{-\delta}\mathbf{I}_N, \qquad \delta \in \mathbb{N} \tag{5.3}$$

leading to *nearly perfect reconstruction* up to a delay. The approximation in Eq. 5.3 implies that

$$\vartheta = \left\| \mathbf{J}\mathbf{G}^T(z)\mathbf{H}(z) - z^{-\delta}\mathbf{I}_N \right\|_2 \tag{5.4}$$

is small. Remark that Eq. 5.4 is similar to the second term of Eq. 4.17. If Eq. 5.4 is near zero, also Eqs. 4.2 and 4.3 will be almost zero. As a consequence, the amplitude, phase and aliasing distortion will be small.

In the same way condition 5.2 can be relaxed to

$$\mathbf{C}(z)\mathbf{B}(z) \approx z^{-\delta}\mathbf{I}_N, \qquad \delta \in \mathbb{N}. \tag{5.5}$$

More information on nearly perfect reconstruction filter bank design can be found in chapter 4.

Finally, note that adaptive filtering operations need to be done in each subband. Perfect reconstruction will only hold for the desired part $s$ of $d$ (upper branch of figure 5.1), as the adaptive filters in between the analysis and synthesis filter bank (lower branch of figure 5.1) do not preserve perfect reconstruction.

### 5.2.3 Perfect path modelling

Apart from the perfect reconstruction or nearly perfect reconstruction condition, another constraint has to be imposed on the filter banks to guarantee appropriate time–invariant system modelling.

Consider the lower, adaptive branch of figure 5.2. If perfect cancellation of the input signal $x$ is aimed at, the lower branch should model the unknown system $w[k]$ exactly.

It is known from the literature [156] that the lower branch can only model a time invariant system if

$$\mathbf{J}\mathbf{G}^T(z)\mathrm{diag}\{F_i(z)\}\mathbf{H}(z) = \underbrace{\begin{bmatrix} \hat{W}_0(z) & \hat{W}_1(z) & \dots & \hat{W}_{N-1}(z) \\ z^{-1}\hat{W}_{N-1}(z) & \hat{W}_0(z) & \dots & \hat{W}_{N-2}(z) \\ \vdots & \vdots & \ddots & \vdots \\ z^{-1}\hat{W}_1(z) & z^{-1}\hat{W}_2(z) & \dots & \hat{W}_0(z) \end{bmatrix}}_{=\text{pseudo–circulant}} \tag{5.6}$$

is a pseudo–circulant matrix (see appendix A).

In the case of DFT modulated filter banks equation 3.1 and 3.8 can be plugged in in Eq. 5.6, leading to

$$\mathbf{C}(z)\mathbf{F}^{-1}\mathrm{diag}\{F_i(z)\}\mathbf{F}\mathbf{B}(z) = \underbrace{\begin{bmatrix} \hat{W}_0(z) & \hat{W}_1(z) & \dots & \hat{W}_{N-1}(z) \\ z^{-1}\hat{W}_{N-1}(z) & \hat{W}_0(z) & \dots & \hat{W}_{N-2}(z) \\ \vdots & \vdots & \ddots & \vdots \\ z^{-1}\hat{W}_1(z) & z^{-1}\hat{W}_2(z) & \dots & \hat{W}_0(z) \end{bmatrix}}_{=\text{pseudo–circulant}}. \tag{5.7}$$

**Theorem 5.1** *It is observed that if Eq. 5.6 or 5.7 is fulfilled the following time–invariant path*

$$\hat{W}(z) = z^{-(N-1)} \sum_{n=0}^{N-1} z^{-n}\hat{W}_n(z^N) \tag{5.8}$$

*is being modelled by the lower branch of figure 5.2.*

**Proof:** The proof can be found in [156].

Remark that Eq. 5.8 is an $N$–th order polyphase decomposition of $\hat{W}(z)$ (cf. Eq. 2.16).

If complete modelling of an unknown system $W(z)$ using a set of subband adaptive filters $F_i(z)$ is aimed at, equation 5.6 should be fulfilled and the order of $\hat{W}(z)$ has to be sufficiently high to model $W(z)$. In general, condition 5.6 will only be satisfied if the adaptive filters $F_i(z)$ are infinitely long, as will be shown in section 5.4.

It is not clear however which constraints have to be imposed on the filter bank polyphase matrices $\{\mathbf{H}(z), \mathbf{G}(z)\}$ such that any finite–order system $W(z)$ can be modelled with finite–length filters $F_i(z)$ following equation 5.6. It will be shown however in chapter 6 that a solution does exist, as for the frequency–domain adaptive filter Eq. 5.7 is fulfilled for finite–length filters $F_i(z)$.

## 5.3 Downsampling and aliasing : two extreme cases

### 5.3.1 Critically downsampled subband schemes

For critically downsampled subband systems $M = N$. Critical downsampling seems attractive because optimal computational savings can be obtained as $N$ is as large as possible.

In the (apparently) ideal case with an infinite order filter bank the analysis and synthesis filters are ideal bandpass and their impulse response is a modulated sinc function (see figure 5.1). A causal implementation however would introduce an infinite delay, which obviously is intolerable. In practice however, filter banks have a restricted filter order. As a consequence there will always be a transition band in between the passband and the stopband of the filter characteristic. Hence, aliasing insertion cannot be avoided whenever critically downsampled subband schemes are employed.

In [70] [71] it is shown that critically downsampled subband adaptive systems suffer from a considerable residual error unless cross filters are included between neighboring subbands. Another solution to this problem could be the introduction of spectral gaps. Cross filters increase the complexity, which is unwanted. Furthermore, cross filters fail to converge quickly to the optimal solution. This suggests the use of oversampled subband schemes for which $M > N$. Whereas critically downsampled subband schemes inherently suffer from subband aliasing, in oversampled subband adaptive systems reduced aliasing distortion is traded off for extra computational costs and slower steady–state convergence.

### 5.3.2 Two–fold oversampled subband systems

In this case the subband signals are subsampled 2 times less compared to the critical scheme, i.e. $M = 2N$. From a computational complexity point of view two–fold oversampled subband systems are less interesting as —roughly speaking— their implementation cost is 4 times higher compared to the case with critical subsampling.

The required subband adaptive filters are roughly twice as long and operate at twice the sampling rate.

As will be shown in chapter 6, frequency–domain adaptive filters can be cast in the subband framework as a special oversampled subband system. For a typical parameter setting for the frequency–domain adaptive filter the oversampling factor $\frac{M}{N}$ is equal to 2. Frequency–domain adaptive filters have desirable convergence properties, and hence provide probably the best solution in those cases where two–fold oversampling is affordable.

In conclusion, two extremes were presented : the critically downsampled scheme, interesting at first sight from a computational complexity point of view but with poor performance, and the two–fold oversampled system with the frequency–domain scheme in mind, more expensive but with desirable convergence properties. The ultimate goal may then be to find subband schemes with an acceptable performance for which $\frac{M}{2} \leqslant N \leqslant M$.

## 5.4 Subband adaptive filter length

Subband adaptive filters are attractive as they offer a low complexity alternative to standard adaptive filtering schemes such as the LMS algorithm. Instead of a single fullband $L_{FB}$–taps FIR filter, $M$ shorter subband filters are adapted in parallel. At first sight one could suggest to choose the subband filter lengths equal to $L_{SB} = \frac{L_{FB}}{N}$, where $N$ is the downsampling factor. As the adaptive computations as well as the filter bank convolutions can be done at the reduced sampling rate, the subband approach is supposed to give a better performance at a lower cost. It appears however that with $\frac{L_{FB}}{N}$–length subband filters the performance is typically unsatisfactory. A discussion on this and some design rules to choose the appropriate subband filter length will be the topic of this section.

### 5.4.1 Infinite–length subband filters

We consider figure 5.1 and pick out one subband, as shown in figure 5.4. Assume now that the unknown system can be modelled as an $L_{FB}$–taps FIR filter $w[k]$. This implies that samples of input $x$ are correlated with samples of $d$ within a time horizon of $\frac{L_{FB}}{f_s}$ if $x$ is white, with $f_s$ being the sampling rate corresponding to $x$. As $w[k]$ is a finite–length FIR system no correlation is supposed to exist between $x$ and $d$ beyond this horizon.

One could expect that during the (ideal) bandpass filtering and $N$–fold downsampling the time horizon over which $x_i^s$ and $d_i^s$ are correlated would stay unchanged. As the sampling rate has been reduced with a factor $N$, the expected equivalent subband filter length $L_{SB}$ would then be $\frac{L_{FB}}{N}$.

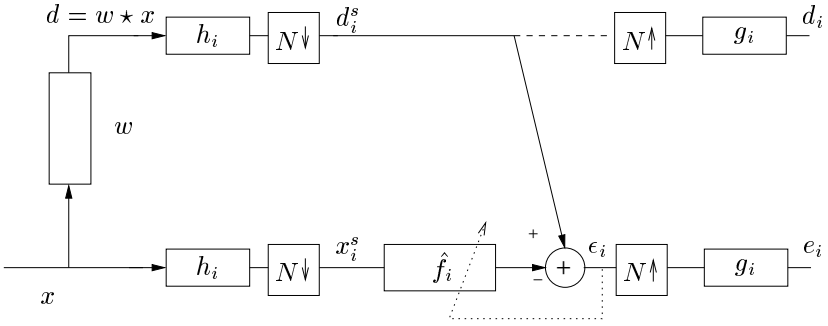


Figure 5.4: Adaptive filtering in subbands : lifting out one band.

As already suggested, this reasoning, based on physical relations between $x$ and $d$, is somewhat imprecise. The downsampling operation implies a kind of interpolation resulting in a much larger correlation time. It appears that $L_{SB}$ should be taken significantly larger than $\frac{L_{FB}}{N}$ in order to obtain a satisfactory error suppression.

**Theorem 5.2** *In the case of an $M$–band, $N$–fold downsampled ideally frequency selective DFT modulated filter bank the $i$–th subband adaptive filter $\hat{f}_i[k]$ in a setup as in figure 5.4 converges to*

$$f_i[k] = \left( w[m] \star \left( e^{-j\frac{2\pi im}{M}} \mathrm{sinc}(\frac{\pi m}{N}) \right) \right)_{N\downarrow}, \tag{5.9}$$

*i.e., $f_i[k]$ can be obtained by downsampling the convolution of the impulse response of the unknown system $w[k]$ and a modulated double–sided sinc.*

**Proof :** The proof is given in appendix C.1.

The sinc spreads out the impulse response in both directions of the time axis. Even when $w[k]$ is an FIR filter, $f_i[k]$ in general is not FIR. The adaptive identification process therefore has to track (many) more than $\frac{L_{FB}}{N}$ samples and due to the spreading–out in both directions of the time axis, an extra delay has to be inserted in the upper signal path. Neglecting the extra delay and the additional subband filter taps strongly limits the convergence of the adaptive filters and leads to a residual undermodelling error. This is illustrated by means of the following experiments.

**Experiment 5.1** In a first experiment an acoustic echo cancellation setup is considered (figure 1.5). The recording room ($4 \times 3 \times 2.5$ m$^3$) is strongly reverberant. The room impulse response is shown in figure 5.5, subplot 1. The distance between loudspeaker and microphone is 20 cm. The room response is an approximation and was generated following the method of Allen and Berkley [4] assuming a sampling

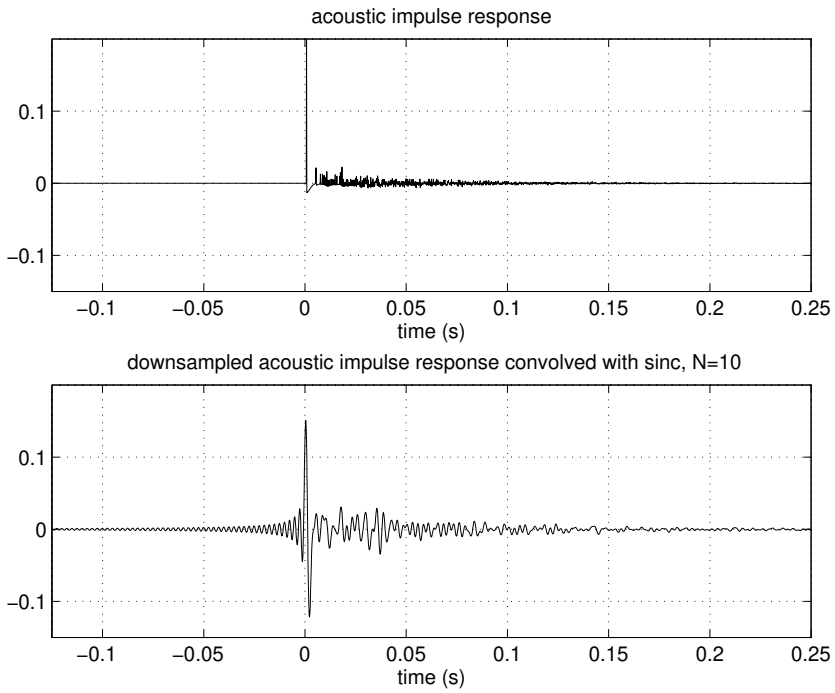


Figure 5.5: Impulse response (subplot 1) and the downsampled convolution with the sinc (subplot 2). The loudspeaker–to–microphone distance is 20 cm, $N = 10$, $i = 0$.

rate of 8 kHz. The downsampled convolution of the acoustic impulse response with the sinc following Eq. 5.9 with $N = 10$ and $i = 0$ is shown in subplot 2. $\triangledown$

Not providing additional subband filter taps can strongly hamper the convergence of the adaptive filters and leads to a residual undermodelling error (cf. experiment 5.5 and figure 5.10). Extending the adaptive filtering horizon in negative time direction seems most effective in reducing the residual error if acoustic impulse responses are involved. The spreading–out of the impulse response in the anti–causal direction is considerable because of the small loudspeaker–to–microphone distance in experiment 5.1. In this case, the acoustic impulse response contains a large dominant direct path peak and additional reverberant peaks with decaying amplitude. For larger microphone–to–loudspeaker distances, this effect is somewhat reduced, which is illustrated by the following experiment.

**Experiment 5.2** The configuration of this second experiment is the same as for the first experiment except that the distance between loudspeaker and microphone is

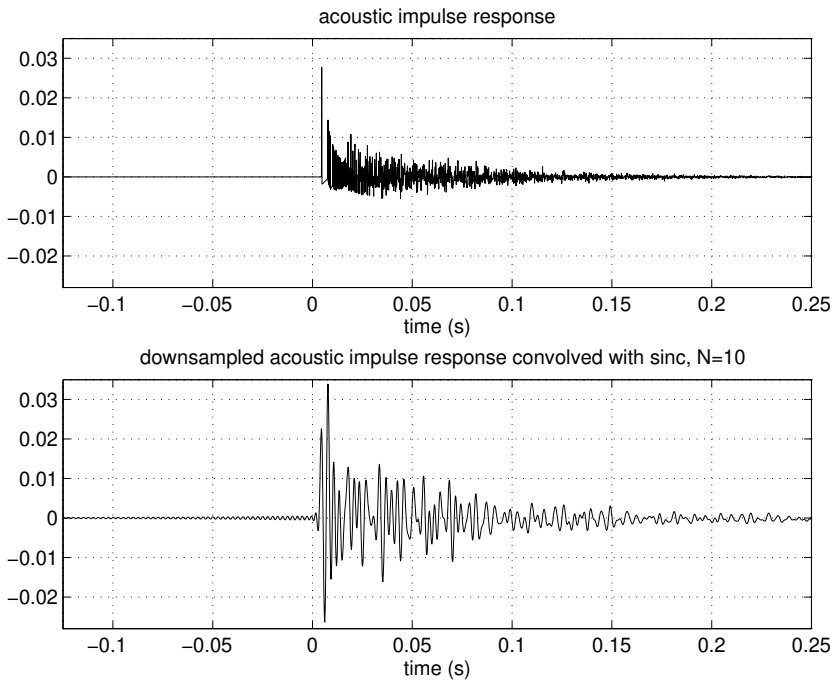


Figure 5.6: Impulse response (subplot 1) and the downsampled convolution with the sinc (subplot 2). The loudspeaker–to–microphone distance is 150 cm, $N = 10$, $i = 0$.

now 150 cm. In figure 5.6 the impulse response and the downsampled convolution with the sinc is shown. The number of extra taps needed is smaller than in the previous experiment. The spreading–out in the anti–causal direction starts at the dominant peak and hence falls partly within the causal part of the filter. $\bigtriangledown$

**Experiment 5.3** Experiment 5.1 is considered again, but now with 5–fold subsampling. The results are plotted in figure 5.7. Compare with figure 5.5. Lowering the downsampling factor reduces the spreading–out of the coefficients, i.e. the spreading seems to die out more quickly. $\bigtriangledown$

**Experiment 5.4** In the following experiment theorem 5.2 is verified using real–world data. The first (left, top) subplot of figure 5.8 shows an acoustic impulse response obtained from measured data. In subplot 2 (right, top), the acoustic impulse response is convolved with $\mathrm{sinc}(\frac{\pi k}{10})$. In subplot 3 (left, bottom), the downsampled version of subplot 2 is shown, corresponding to Eq. 5.9 with $N = 10$ and $i = 0$. On the other hand the input data ($x$ and $d$) were filtered with a sharp lowpass filter with cutoff 0.05 and were 10–fold downsampled (cf. figure 5.4). The

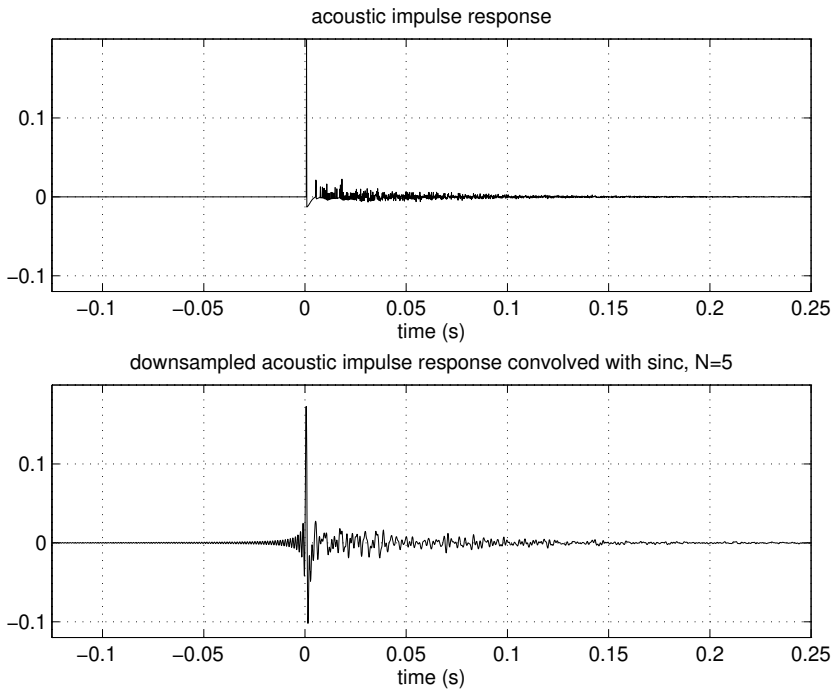


Figure 5.7: Impulse response (subplot 1) and the downsampled convolution with the sinc (subplot 2). The loudspeaker–to–microphone distance is 20 cm, $N = 5$, $i = 0$.

impulse response between the subband data was identified and is shown in subplot 4. Apart from some measurement noise it matches the downsampled convolution of subplot 3, which was theoretically obtained following Eq. 5.9. $\triangledown$

Note that it was assumed in the derivation of Eq. 5.9 that the filter bank is ideally frequency selective, i.e. that no aliasing is inserted by the subsampling. In practice this condition is not fulfilled. Strictly speaking, in that case Eq. 5.9 is not valid any longer. Furthermore, due to the aliasing an excess mean–squared error will appear on top of the undermodelling error as the aliasing will move the adaptive filters away from their Wiener solution.

Remark also that the spreading–out of $f_i[k]$ in the anti–causal direction is caused by the downsampling and not by the fact that ideal, hence infinite–order, anti–causal filter banks are applied.

Referring again to the three design criteria for subband adaptive systems specified in section 5.2, it follows from Eq. 5.9 that for the standard subband adaptive

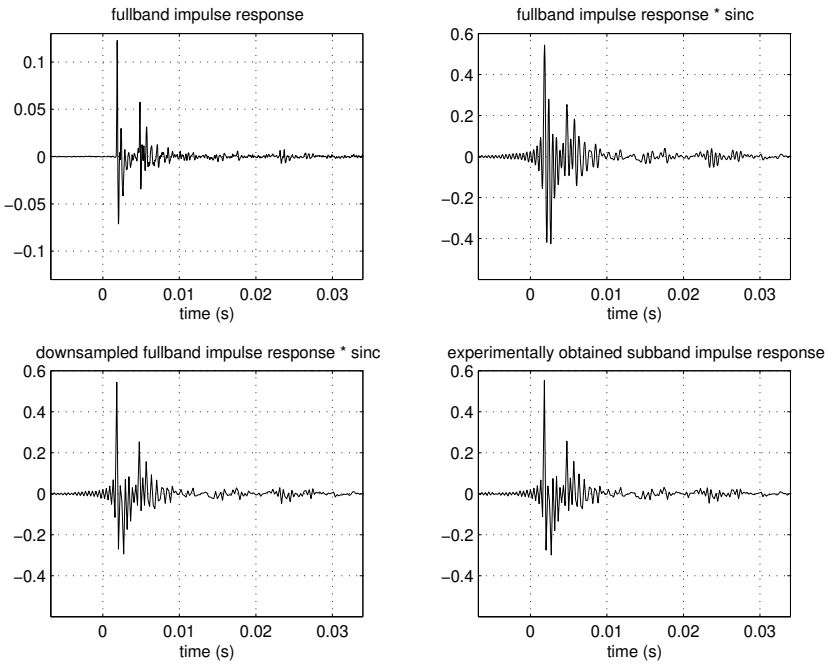


Figure 5.8: The first (left, top) subplot shows an acoustic impulse response obtained from measured data. In subplot 2 (right, top), the acoustic impulse response is convolved with $\text{sinc}(\frac{\pi k}{10})$. In subplot 3 (left, bottom), the downsampled version of subplot 2 is shown, corresponding to Eq. 5.9 with $N = 10$ and $i = 0$. The input data ($x$ and $d$) were also filtered with a sharp lowpass filter with cutoff 0.05 and were 10–fold downsampled (cf. figure 5.4). The impulse response between the subband data was identified and is shown in subplot 4. Apart from some measurement noise it matches the downsampled convolution of subplot 3, which was theoretically obtained following Eq. 5.9.

filter as presented in figure 5.1 all three conditions can be fulfilled if a high–order, frequency–selective perfect reconstruction filter bank set is applied with infinite length subband adaptive filters. The latter requirement can however not be fulfilled in practice. In the forthcoming chapters we will try to find solutions for which the third criterion on perfect path modelling is fulfilled for finite length subband adaptive filters.

### 5.4.2 Introducing anti–causal filter taps

In this section two design rules are presented to determine the number of extra subband filter taps that are needed to reduce the residual error. It was shown in the previous section that in acoustic signal enhancement applications the spreading–out effect is most clearly present in the anti–causal direction. Hence, we will concentrate

on the negative time axis in this section. The results are readily applicable to the causal direction as well.

It is observed from Eq. 5.9 that $f_i[k]$ is infinitely long. In practice the adaptive filter $\hat{f}_i[k]$ (see figure 5.4) has finite length and hence is an approximation of $f_i[k]$. Some of the coefficients of $f_i[k]$ will not be modelled, resulting in a residual error. Evidently, the more coefficients are modelled, the lower the residual error.

It is clear that in practice a certain residual error will always remain, whatever the number of extra added filter taps may be. It should be noted however that for many applications this is not necessarily a big issue. In the case of acoustic echo cancellation for instance the echo suppression is typically limited to 25 or 30 dB in practice due to non–linearities in the signal path, time variations of the acoustic transfer function and misdetection in the adaptation control algorithm (see also chapter 10). If the error due to the residual undermodelling then has the same order of magnitude as the errors mentioned before no significant performance loss is expected w.r.t. standard fullband adaptive filtering structures.

In a straightforward implementation $\hat{f}_i[k]$ would be an $\frac{L_{FB}}{N}$–taps filter for which $\hat{f}_i[k] = 0$ for $k < 0$ and $k \geqslant \frac{L_{FB}}{N}$. From section 5.4.1 it is known that some causal and anti–causal taps should be added. In the sequel $L_{ac}$ refers to the number of anti–causal taps and $L_c$ represents the extra taps added in the causal direction. The subband adaptive filter then has $\frac{L_{FB}}{N}$ main coefficients, complemented with $L_{ac}$ anti–causal taps and $L_c$ extra coefficients to compensate for the spreading–out in the causal direction. Therefore,

$$L_{SB} = \frac{L_{FB}}{N} + L_{ac} + L_c. \tag{5.10}$$

**Threshold–based method**

From Eq. 5.9 it follows that if $w[m] = \delta[m]$, i.e. if there is a Dirac impulse at the time origin,

$$|f_i[k]| = |\mathrm{sinc}(\pi k)| \leqslant \frac{1}{\pi|k|}. \tag{5.11}$$

It is observed that for all unmodelled coefficients on the negative time axis

$$|f_i[k]| < \frac{1}{\pi L_{ac}}, \qquad \text{if } -\infty < k < -L_{ac}, \quad k \in \mathbb{Z}. \tag{5.12}$$

Hence, if a threshold $\tau$ is defined the following design rule

$$L_{ac} \geqslant \frac{1}{\pi\tau} \tag{5.13}$$

determines the number of anti–causal taps that are needed such that all neglected coefficients fall below the threshold.

It is observed from Eq. 5.13 that $L_{ac}$ is a function of the desired threshold $\tau$ and that $L_{ac}$ is not depending on $M$ or $N$, i.e. independently of the downsampling factor a fixed number of coefficients are needed to reduce the residual error. Remark that $L_{ac}$ extra filter taps correspond to a delay of $L_{ac}\frac{f_s}{N}$, which does depend on $N$. This was also observed in experiment 5.3.

From a complexity point of view $L_{ac}$ should not be considered as such, but must be compared with the total filter length $L_{SB}$. As $\frac{L_{FB}}{N}$ is inversely proportional to $N$ equation 5.10 shows that $L_{ac} \approx L_{SB} \gg \frac{L_{FB}}{N}$ for large values of $N$. As a consequence, the complexity gain of the subband adaptive filter w.r.t. the LMS algorithm will be smaller than expected for "large" values of $N$. On the other hand, for large values of $N$ the filter banks tend to dominate the cost such that the effect of $L_{ac}$ is somewhat masked (see also section 5.5.3).

### Residual error–based method

Instead of thresholding the amplitude of the unmodelled coefficients as in equation 5.13 it is better to concentrate on the residual error to find an appropriate value for $L_{ac}$.

**Theorem 5.3** *Consider the general adaptive filtering setup of figure 2.8. Define $L_w \times 1$ vectors $\mathbf{x}$ and $\mathbf{w}$, containing the input samples $x$ and the coefficients of the unknown system $w$ respectively, and an $L_{\hat{w}} \times 1$ vector $\hat{\mathbf{w}}$ representing the adaptive filter. If $L_{\hat{w}} \leqslant L_w$, $\hat{\mathbf{w}}$ is an approximation of $\mathbf{w}$, modelling only a part of the unknown system. Under these assumptions the expected minimum residual error is given by*

$$\min_{\hat{\mathbf{w}}} \ \mathcal{E}\{|e[k]|^2\} = \mathcal{E}\{|d[k]|^2\} - \mathbf{w}^H \mathbf{R}_{xx} \mathbf{\Xi} \left( \mathbf{\Xi}^T \mathbf{R}_{xx} \mathbf{\Xi} \right)^{-1} \mathbf{\Xi}^T \mathbf{R}_{xx} \mathbf{w}, \tag{5.14}$$

*in which $\mathcal{E}\{|d[k]|^2\} = \mathbf{w}^H \mathbf{R}_{xx} \mathbf{w}$ and $\mathbf{R}_{xx} = \mathcal{E}\{\mathbf{x}\mathbf{x}^H\}$. Further, $\mathbf{\Xi}$ is an $L_w \times L_{\hat{w}}$ matrix containing ones and zeros indicating which coefficients of $\mathbf{w}$ are modelled, such that $e[k] = \mathbf{w}^H \mathbf{x} - (\mathbf{\Xi}\hat{\mathbf{w}})^H \mathbf{x}$.*

**Proof:** The proof is given in appendix C.2.

From theorem 5.3 it follows also that the maximum residual error suppression is given by

$$\max_{\hat{\mathbf{w}}} \ \frac{\mathcal{E}\{|d[k]|^2\}}{\mathcal{E}\{|e[k]|^2\}} = \frac{\mathbf{w}^H \mathbf{R}_{xx} \mathbf{w}}{\mathbf{w}^H \mathbf{R}_{xx} \mathbf{w} - \mathbf{w}^H \mathbf{R}_{xx} \mathbf{\Xi} \left( \mathbf{\Xi}^T \mathbf{R}_{xx} \mathbf{\Xi} \right)^{-1} \mathbf{\Xi}^T \mathbf{R}_{xx} \mathbf{w}}. \tag{5.15}$$

Formula 5.15 can now be applied to figure 5.4 to estimate the maximum subband error suppression

$$\mu_{\epsilon_i} = \max_{\hat{\mathbf{w}}} \ \frac{\mathcal{E}\{|d_i^s|^2\}}{\mathcal{E}\{|\epsilon_i|^2\}} = \frac{\mathcal{E}\{|d_i^s|^2\}}{\mathcal{E}\{|\epsilon_i|^2\}_{\min}}. \tag{5.16}$$

$\mathbf{w}$ in Eq. 5.15 corresponds to $f_i[k]$ as obtained from Eq. 5.9 and $\hat{\mathbf{w}}$ matches the $L_{SB} \times 1$ vector $\hat{f}_i[k]$. In theory $\mathbf{w}$ is infinitely long, in practice however we will consider an $L_w \times 1$ vector $\mathbf{w}$ containing only the central part of $f_i[k]$, with $L_w$ being significantly larger than $L_{SB}$. Matrix $\mathbf{\Xi}$ indicates which coefficients of $f_i[k]$ will be modelled. Given the numbers $L_{SB}$, $L_{ac}$ and $L_c$ matrix $\mathbf{\Xi}$ can be constructed as in Eq. C.29. Further, $\mathbf{R}_{xx}$ is the autocorrelation matrix corresponding to $x_i^s$. Hence, $\mathbf{R}_{xx}$ depends on the intrinsic coloring of the fullband input $x$ and on the frequency characteristic of the $i$–th analysis filter, and is affected by the downsampling operation. If $x$ can be written as $c \star n$, with $n$ a stationary white noise source and $c$ a coloring filter, the coloring of $x_i^s$ is approximately given by $\kappa = (c \star h_i)_{N\downarrow}$. It is explained in appendix C.2 how $\mathbf{R}_{xx}$ can be computed once the coloring filter $\kappa$ is known.

Hence, applying the preceding procedure to each subband $i$ of the general subband adaptive scheme (see figure 5.1), a set of $M$ values $\mu_{\epsilon_i}$ can be obtained. It is observed from figure 5.4 that the optimal suppression of the upsampled and filtered error $e_i$

$$\mu_{e_i} = \frac{\mathcal{E}\{|d_i|^2\}}{\mathcal{E}\{|e_i|^2\}_{\min}} \approx \mu_{\epsilon_i}. \tag{5.17}$$

From figure 5.1 and 5.4 it follows that

$$e = \sum_{m=0}^{M-1} e_m \tag{5.18}$$

and hence

$$\mathcal{E}\{|e|^2\} = \mathcal{E}\left\{\left|\sum_{m=0}^{M-1} e_m\right|^2\right\} \approx \sum_{m=0}^{M-1} \mathcal{E}\{|e_m|^2\}. \tag{5.19}$$

The approximation will be exact if the signals $e_m$ are uncorrelated. However, $\mathcal{E}\{e_m^* e_n\} \neq 0$ for $m \neq n$ because of the overlapping of the different subbands.

From Eq. 5.19 it follows that the global residual error suppression can be approximated by

$$\mu_e = \max_{\{\hat{f}_0,\ldots,\hat{f}_{M-1}\}} \frac{\mathcal{E}\{|d[k]|^2\}}{\mathcal{E}\{|e[k]|^2\}} \approx \frac{\sum_{m=0}^{M-1} \mathcal{E}\{|d_m^s|^2\}}{\sum_{m=0}^{M-1} \mathcal{E}\{|\epsilon_m|^2\}_{\min}}. \tag{5.20}$$

Although the maximum subband error suppression $\mu_{\epsilon_i}$ to which the subband adaptive filters converge can be estimated fairly well using this method the estimate of the global measure is less accurate.

From Eq. 5.15 the (subband dependent) parameters $L_{ac}$, $L_c$ and $L_{SB}$ can now be computed such that $\mu_{\epsilon_i}$ or $\mu_e$ is as desired.

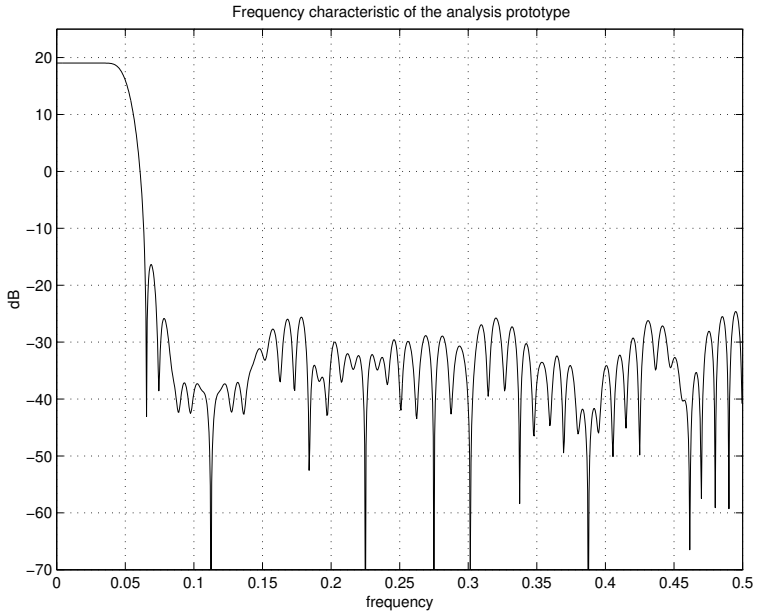


Figure 5.9: Frequency characteristic of the analysis prototype filter

**Experiment 5.5** For this experiment a randomly generated FIR system of order 31 was identified using a white noise input. Three different adaptive algorithms are compared :

a) a 10–band, 8–fold downsampled standard subband adaptive system (see figure 5.1) was designed with subband adaptive filters having length $\frac{L_{FB}}{N} = 4$. The NLMS algorithm was used to adapt the subband filters. Both the analysis and the synthesis filters are FIR, DFT modulated and are of order 223. The relative aliasing level introduced by the filter banks (Eq. 3.75) is around -38.4 dB. The filter bank set was designed such that the perfect reconstruction condition 3.32 is fulfilled. The frequency characteristic of the analysis prototype filter is shown in figure 5.9. From figure 5.10 it is clear that for the subband adaptive filter without extra taps a considerable undermodelling error remains.
Formula 5.15 was applied to each subband. It appeared that at the end of the simulation for each subband the error suppression was less than 1 dB below the optimum $\mu_{\epsilon_i}$. The global residual error suppression was estimated using Eq. 5.20 and appeared to be 13.4 dB, which is a fair approximation of the suppression observed from figure 5.10.

b) To the same subband system then extra filter and anti–causal taps were added

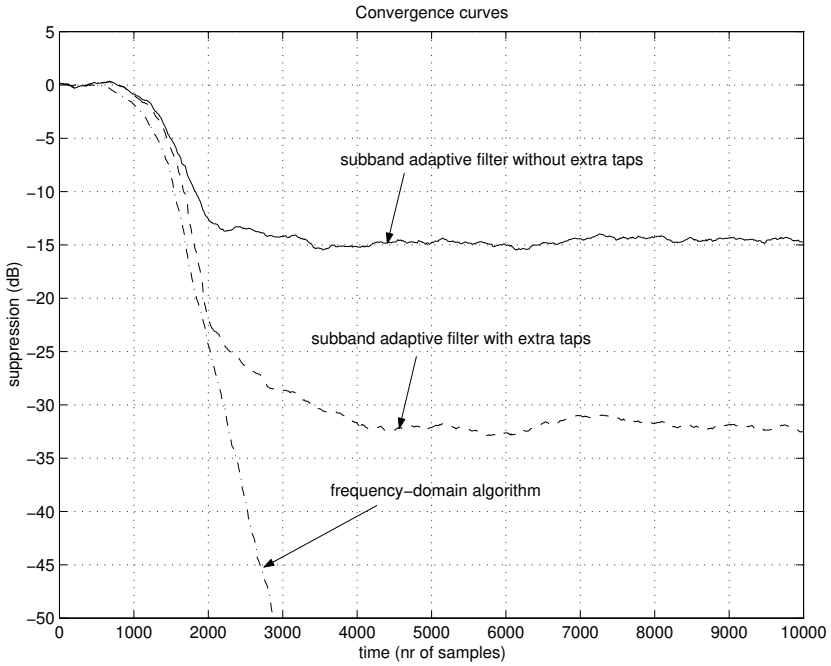


Figure 5.10: Convergence curves of 3 different adaptive systems. The FIR system to be identified is of order 31.

to allow a better modelling. From Eq. 5.13 it is observed that two anti–causal taps ($L_{ac} = 2$) corresponds to a threshold $\tau = -16$ dB. Hence 2 anti–causal taps were added per filter and there were a total of 10 filter taps per subband ($L_{ac} = 2, L_c = 4$), increasing the number of modelling parameters from 40 to 100. It can be observed from figure 5.10 that the steady–state undermodelling error is strongly reduced.
Also for this simulation the global residual error suppression was computed using Eq. 5.20. $\mu_e = 28.8$ dB was found, apparently being a less accurate estimate (being about -32 dB in figure 5.10).
Note that the unknown system was randomly generated. As a consequence, adding extra taps in the causal direction seems to be as effective as providing anti–causal filter taps. Furthermore, the error suppression obtained from the first simulation with $L_{ac} = L_c = 0$ is rather limited. If acoustic impulse responses are involved typically more error suppression is obtained even for $L_{ac} = L_c = 0$ and extending the adaptive filter to the negative time axis appears to be most effective.

c) Finally, the error convergence curve of a well–known frequency–domain adaptive filter, the so–called subband–normalized constrained PBFDAF (Eqs. 6.28-

6.32 and 6.35) is also shown in figure 5.10. Apparently, perfect modelling is possible with the PBFDAF as the residual error curve goes to 0. It will be proven in paragraph 6.2 that the PBFDAF can be considered as a kind of subband adaptive system. The filter applied here has 10 subbands, which are 2–fold downsampled. The subband adaptive filters of the PBFDAF used in this experiment have length 4, and hence, having the same number of modelling parameters as the first subband adaptive filtering scheme, the PBFDAF clearly outperforms both subband adaptive filters in terms of error suppression.

Note that the adaptation stepsizes were not optimized to obtain the fastest convergence, so no conclusions should be made about the relative convergence speed of the different adaptive filters based on figure 5.10. ▽

## 5.5 Implementation cost and complexity gain with respect to LMS

Subband adaptive filtering techniques provide a reduced complexity solution w.r.t. fullband algorithms such as the LMS adaptive filter. In this section the computational complexity of the subband adaptive filter will be estimated and the complexity gain w.r.t. the LMS algorithm will be computed.

Throughout this section it will be assumed that all input signals ($x$ and $d$) and the unknown system $w[k]$ (see figure 2.8) are real–valued. This is a realistic assumption for many applications such as acoustic echo cancellation. Further, it is assumed that $f_s$ is the sampling rate corresponding to the input signals $x$ and $d$.

### 5.5.1 Rough cost estimate

Making some assumptions a first rough cost estimate can be obtained. For small oversampling factors $\frac{M}{N}$, i.e. for setups for which $M \approx N$, the adaptive filtering cost —being the cost for the adaptation of a subband filter times the number of subbands to be processed times the sampling rate at which the adaptation is done— equals

$$c = 16 L_{SB} \frac{M}{2} \frac{f_s}{N} = 8 \frac{M L_{FB} f_s}{N^2} \approx 8 \frac{L_{FB} f_s}{N} \texttt{ops}. \tag{5.21}$$

It is assumed here that only about half of the subbands need to be processed as they are pairwisely complex conjugated, that a linear adaptation algorithm such as LMS[2] is applied and that $L_{SB} \approx \frac{L_{FB}}{N}$.

[2] A formula for the implementation cost of the LMS algorithm can be found in section 2.3.1.

A fullband approach using the LMS adaptive filter amounts to $4L_{FB}f_s$ ops. approximately. Hence, the subband algorithm appears to be a factor $\frac{N}{2}$ cheaper than a standard fullband solution.

### 5.5.2 Detailed cost analysis

It was already shown in section 5.4 that in practice $L_{SB} > \frac{L_{FB}}{N}$, whereas it is often claimed that $L_{SB} \approx \frac{L_{FB}}{N}$. Further, subband oversampling is mandatory, which implies that $M > N$, although one often assumes that critical subsampling will do. Finally, the inherent filter bank cost was neglected in the above computation. Hence, the computation presented in section 5.5.1 underestimates the implementation cost and overestimates the expected complexity gain w.r.t. LMS.

In this section a more detailed analysis is performed, leading to a more reliable estimate of the implementation cost and the complexity gain w.r.t. LMS. It will be shown in section 5.5.3 that the rough cost estimate of section 5.5.1 is a good approximation if $M = N$ and $L_{SB} = \frac{L_{FB}}{N}$, but that for realistic parameter settings the complexity gain w.r.t. LMS is smaller than $\frac{N}{2}$.

It is assumed that $L_{SB}$ is the length of the subband adaptive filters, that $M$ represents the number of subbands and that $N$ is the downsampling factor. The filter banks are (modified) DFT modulated. The length of the analysis prototype filter is $L_f^a$, the length of the synthesis prototype is $L_f^s$. The major cost contributions are :

- the adaptive filtering cost
- the filter bank cost, which implies
  - a DFT or IDFT operation
  - cost of the structured polynomial matrices $\mathbf{B}(z)$ and $\mathbf{C}(z)$

A detailed cost analysis can be found in appendix C.3. The results are summarized here.

If $M$ is a power of 2 fast signal transforms can be called for, leading to

$$c = (4L_{SB}n_{rb} + 16L_{SB}n_{cb} + 2n_{cb} + n_{rb} + 4L_f^a + 2L_f^s + 6M\log_2 M - N - 10M + 12)\frac{f_s}{N}\ \texttt{ops}. \quad (5.22)$$

For $M \neq 2^r, r \in \mathbb{N}$,

$$c = (4L_{SB}n_{rb} + 16L_{SB}n_{cb} - 2n_{cb} - n_{rb} + 4L_f^a + 2L_f^s - N - 3M + 3Mn_{rb} + 12Mn_{cb})\frac{f_s}{N}\ \texttt{ops}. \quad (5.23)$$

is obtained. $n_{rb}$ stands for the number of real subbands to be processed and $n_{cb}$ is the number of processed complex subbands. If $M$ is even, $n_{rb} = 2$ and $n_{cb} = \frac{M-2}{2}$ as only about half of the complex subbands need to be computed. For odd $M$ on the other hand, $n_{rb} = 1$ and $n_{cb} = \frac{M-1}{2}$.

### 5.5.3 Cost evaluation

For different combinations of the parameters the implementation cost of the subband adaptive filter is computed by evaluating formula 5.22 and 5.23.

Four simulations were done in which the complexity gain w.r.t. the LMS algorithm was computed as a function of the number of subbands $M$, changing the length of the (fullband) LMS adaptive filter $L_{FB}$, the number of causal and anti–causal taps $L_c$ and $L_{ac}$ and the oversampling factor $\frac{M}{N}$ from one simulation to another. The subband adaptive filter length was taken equal to $L_{SB} = L_{ac} + \frac{L_{FB}}{N} + L_c$. It is further assumed that both the analysis and the synthesis filter bank have the same number of coefficients, i.e. $L_f^a = L_f^s = 30 + 5M$ : the filter bank order is increased proportionally to the number of subbands $M$. This is a realistic assumption if e.g. FFT–based filter design techniques are used to generate the filter bank prototype filter.

**Example 5.1** Figure 5.11 shows the complexity gain for a subband adaptive filter with critically downsampled subbands. The fullband filter length was $L_{FB} = 2000$, leading to about 8000 `op.` per sample for the fullband LMS algorithm. Further, $L_{ac} = L_c = 0$. The highest complexity gain is expected in this simulation as $M = N$ and no (anti–)causal taps were added to reduce the residual error.
It is observed that for $M = 2^r, r \in \mathbb{N}$ the complexity gain strongly increases as fast signal transforms can be used. The dashed line indicates how the complexity gain would evolve if fast signal transforms were used also for intermediate values of $M$, not being a power of 2. Values of $M$ for which $M = 2^r$ are highlighted with an asterisk. △

**Example 5.2** In a second simulation the downsampling factor is chosen equal to $N = \text{rnd}(0.75M)$, leading to 33% oversampling. By applying formula 5.13 with $\tau = -20$ dB, it is found that $L_{ac} = 3$. Further, $L_c$ was chosen equal to 2 and $L_{FB} = 2000$. The results are plotted in figure 5.12. △

**Example 5.3** In figure 5.13 a similar simulation is shown now with $L_{FB} = 500$ being much shorter. The cost of the fullband LMS algorithm now decreases to 2000 `op.` per sample. Further, $L_{ac} = 3$, $L_c = 2$ and $N = \text{rnd}(0.75M)$. △

**Example 5.4** In the last simulation, shown in figure 5.14, 100% oversampling is considered, i.e. $N = \text{rnd}(0.5M)$. Further, $L_{FB}$ is again 2000, $L_{ac} = 3$ and $L_c = 2$. △

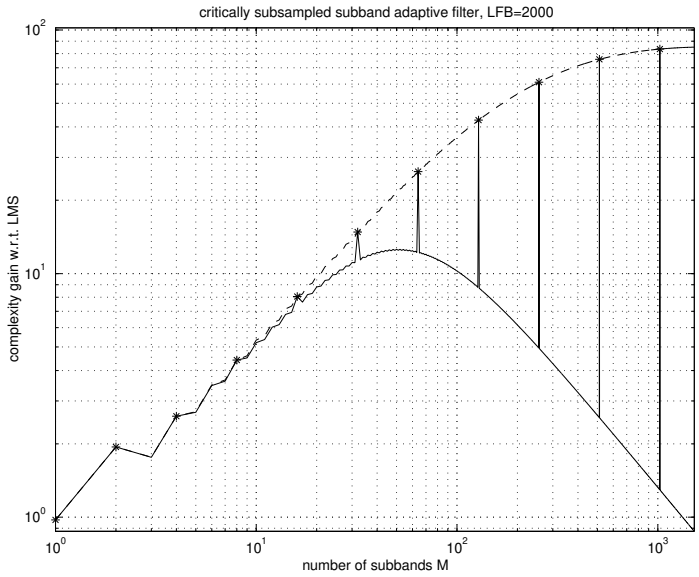


Figure 5.11: Cost analysis for a critically subsampled subband adaptive filter for which $L_{FB} = 2000$ : complexity gain w.r.t. LMS

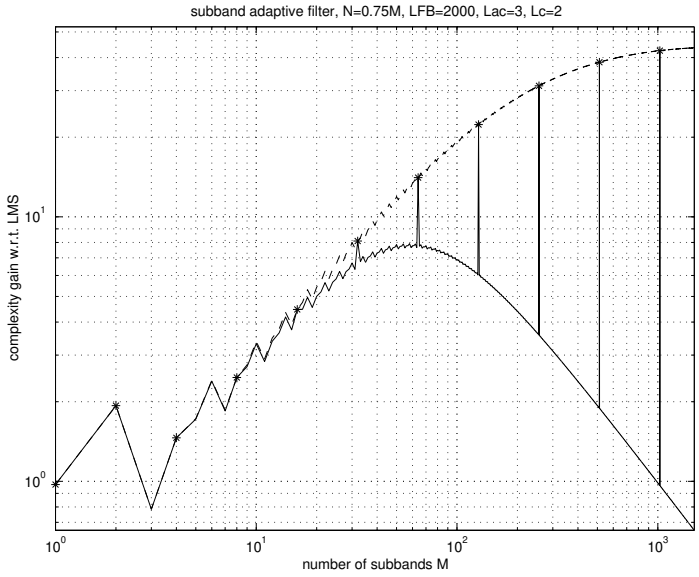


Figure 5.12: Cost analysis for an oversampled subband adaptive filter for which $N = \mathrm{rnd}(0.75M)$ and $L_{FB} = 2000$ : complexity gain w.r.t. LMS

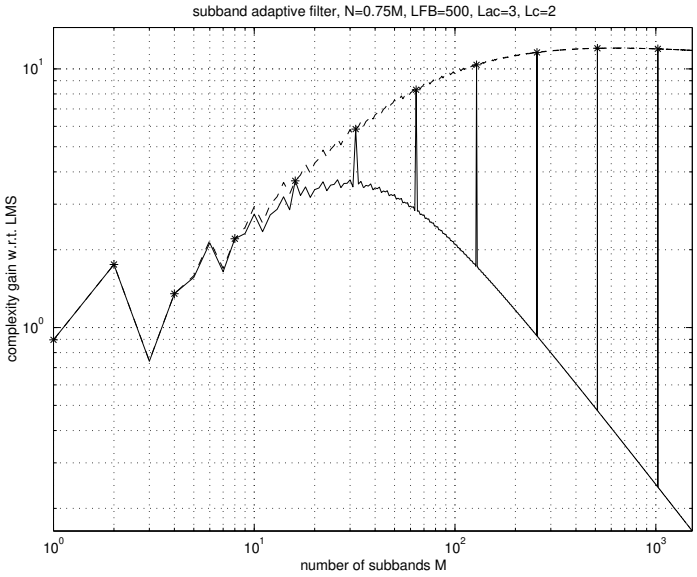


Figure 5.13: Cost analysis for an oversampled subband adaptive filter for which $N = \mathrm{rnd}(0.75M)$ and $L_{FB} = 500$ : complexity gain w.r.t. LMS

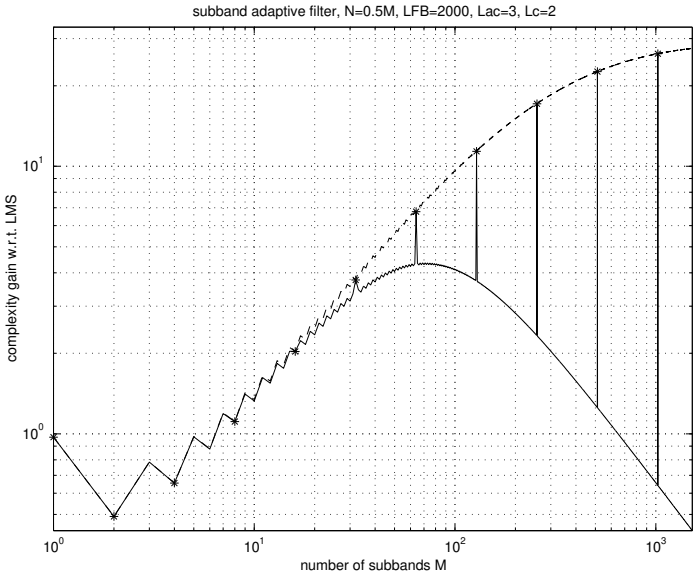


Figure 5.14: Cost analysis for an oversampled subband adaptive filter for which $N = \mathrm{rnd}(0.5M)$ and $L_{FB} = 2000$ : complexity gain w.r.t. LMS

By evaluating figures 5.11–5.14 it is observed that the complexity gain is substantial if fast signal transforms can be used, i.e. if $M = 2^r, r \in \mathbb{N}$. An explicit computation of the DFT–matrix times the signal vector for values of $M$ not being a power of 2 on the other hand leads to a complexity gain that is rather limited or sometimes even smaller than 1 and is therefore not a viable option.

Observe that the complexity gain for a 32–band critically subsampled system for which no extra (anti–)causal taps were added is almost 15 (see figure 5.11). This corresponds more or less to $\frac{N}{2}$, the rough estimate made for the complexity gain in section 5.5.1. Remark however that critically subsampled systems severely suffer from aliasing insertion and residual errors. In order to reduce the residual error extra filter taps are added and $N$ is taken smaller than $M$ to decrease the aliasing level. For a more realistic scenario for which e.g. $M = 32$, $N = 24$, $L_{ac} = 3$ and $L_c = 2$ (as in example 5.2) the complexity gain reduces from 15 to 8. It can be concluded that as the oversampling increases and more extra taps are added the complexity gain decreases, as could be expected.

Finally, it is observed that if $M$ is increased the complexity gain initially increases. However, for large values of $M$ (not shown on the plots) the gain decreases again. In that case also $N$ will be large and hence the subband filters have their smallest order, i.e. $\max(L_{ac} + L_c - 1, 0)$, which is now independent of $N$. Whereas for small $M$ the cost associated with the adaptive filters is most important, for large $M$ the filter banks start to dominate the implementation cost.

Finally, figures 5.11–5.14 show that for realistic parameter settings —as would be used in an acoustic echo cancellation setup for instance— the subband adaptive filter outperforms the LMS algorithm in terms of computational complexity. Although the complexity gain is smaller than suggested in section 5.5.1, still a substantial cost reduction can be obtained w.r.t. the LMS algorithm.

## 5.6 Conclusions

The identification of high–order FIR systems requires efficient adaptive algorithms such as the subband adaptive filter, which outperforms standard fullband approaches both from a complexity and a performance point of view.

In section 5.1 subband adaptive filtering was discussed in general, making a comparison with the standard fullband approach and discussing the choice of appropriate filter banks.

In section 5.2 three design criteria for appropriate subband modelling were presented. They are necessary requirements to ensure satisfactory performance of the subband adaptive filter.

In section 5.3 it was indicated that the downsampling inserts aliasing components

in the subbands, which hamper the convergence of the adaptive filters. It was argued that critically subsampled adaptive filters, although leading to the largest computational savings, are not a good candidate for subband adaptive processing.

It was shown in section 5.4 that subband adaptive filters suffer from a residual undermodelling error unless extra (anti–)causal subband filter taps are inserted. A formula was derived that describes the phenomenon and two design methods were presented from which the appropriate subband adaptive filter order can be computed.

In section 5.5 a cost analysis was performed. It appeared that although the complexity gain w.r.t. the fullband approach is less than expected, still a considerable complexity reduction can be obtained with subband adaptive filtering techniques.

# Chapter 6

# Analysis of the Partitioned Block Frequency–Domain Adaptive Filter as a Subband Adaptive System

As a cheaper alternative to the LMS algorithm, the frequency–domain adaptive filter (FDAF) was introduced, being a direct translation of Block–LMS to the frequency domain [142]. Instead of linear convolutions and correlations component–wise operations are performed in the frequency domain. This requires some "restore" operations, which are based on overlap–save or overlap–add techniques. A detailed discussion of the FDAF algorithm can be found on page 40.

The performance and the convergence properties of the FDAF algorithm are comparable to those of the LMS algorithm. It appears that the FDAF algorithm is computationally attractive only if the block length $L$ has the same order of magnitude as the filter length $L_{FB}$. In practice however, this leads to unacceptable input/output delays. Partitioning may then be used to reduce the overall delay. This leads to the so–called Partitioned Block Frequency–Domain Adaptive Filter (PBFDAF) [128] [145] [147], which is discussed in this chapter.

In section 6.1 the PBFDAF algorithm is derived from the Block–LMS adaptive filter. The overlap–save correction is explained and it is shown how the adaptive filter can be partitioned. Some properties of the PBFDAF are summarized and an ambiguity that can occur with the so–called unconstrained PBFDAF is discussed.

It will be shown in section 6.2 that the PBFDAF fits into the subband framework

presented in chapter 5. The PBFDAF can be viewed as a DFT modulated subband system using filter banks with a poor frequency selectivity, to which a special error correction module has been added. Thanks to the error correction the algorithm has good convergence characteristics despite the low frequency selectivity of its filter banks.

It will be proven in section 6.3 that the design criteria on perfect reconstruction and perfect path modelling as presented in section 5.2 for the subband adaptive filter are fulfilled for the PBFDAF algorithm.

In section 6.4 a cost analysis will be performed. A global cost estimate is derived for different algorithmic settings. The influence of the different algorithmic parameters on the total cost will be examined. It appears that for realistic parameter settings the PBFDAF is a cheaper alternative to the LMS algorithm.

The conclusions are formulated in section 6.5.

## 6.1 Partitioned Block Frequency–Domain Adaptive Filter

The frequency–domain adaptive filter (FDAF) as defined in section 2.3.2 is known as a cheaper alternative to the popular LMS algorithm. The main drawback of the FDAF is the large block length, leading to a processing delay that is unacceptable for real–time applications.

By splitting the adaptive filter $\hat{w}$ shown in figure 2.8 in equal parts and transforming these parts separately to the frequency domain, a mixed time/frequency–domain adaptive filter is obtained, called the Partitioned Block Frequency–Domain Adaptive Filter (PBFDAF) [5] [38] [59] [61] [128] [145] [147]. In this way the block length can be reduced, resulting in a cheap adaptive filter with acceptable processing delay. An interesting algorithm results, which is used in many commercial applications nowadays. It is known for instance that several advanced echo cancellation systems brought to the market these days are based on this kind of algorithm.

### 6.1.1 Derivation of the PBFDAF algorithm

Consider the Block–LMS algorithm presented in section 2.3.2. Recall equation 2.40, which defines the filtering operation $\hat{w} \star x$ :

$$\mathbf{y}_n = \mathbf{X}_n^T \mathbf{w}_n \tag{6.1}$$

$$= \begin{bmatrix} x[nL+1] & \dots & x[nL-L_{FB}+2] \\ \vdots & \ddots & \vdots \\ x[(n+1)L] & \dots & x[(n+1)L-L_{FB}+1] \end{bmatrix} \begin{bmatrix} \hat{w}^{(n)}[0] \\ \vdots \\ \hat{w}^{(n)}[L_{FB}-1] \end{bmatrix}. \tag{6.2}$$

The $L_{FB}$–taps adaptive filter $\hat{w}^{(n)}[k]$ can be partitioned in $\frac{L_{FB}}{P}$ parts $\hat{w}_p^{(n)}[k]$ of length $P$ each, whereby $P$ is a divisor of $L_{FB}$ :

$$\mathbf{y}_n = \begin{bmatrix} \mathbf{X}_{n,0}^T & \dots & \mathbf{X}_{n,\frac{L_{FB}}{P}-1}^T \end{bmatrix} \begin{bmatrix} \mathbf{w}_0^{(n)} \\ \vdots \\ \mathbf{w}_{\frac{L_{FB}}{P}-1}^{(n)} \end{bmatrix}, \tag{6.3}$$

with

$$\mathbf{X}_{n,p} = \begin{bmatrix} x[nL-pP+1] & \dots & x[(n+1)L-pP] \\ \vdots & \ddots & \vdots \\ x[nL-pP-P+2] & \dots & x[(n+1)L-pP-P+1] \end{bmatrix} \tag{6.4}$$

and

$$\mathbf{w}_p^{(n)} = \begin{bmatrix} \hat{w}^{(n)}[pP] \\ \vdots \\ \hat{w}^{(n)}[(p+1)P-1] \end{bmatrix}, \qquad p = 0 : \frac{L_{FB}}{P} - 1. \tag{6.5}$$

If now the $M \times 1$ vector

$$\mathbf{w}_p^{(n)} = \begin{bmatrix} \mathbf{w}_p^{(n)} \\ \mathbf{0}_{(M-P)\times 1} \end{bmatrix} \tag{6.6}$$

is defined and the $P \times L$ matrix $\mathbf{X}_{n,p}$ is embedded in the right top of the following $M \times M$ right–circulant matrix

$$\bar{\mathbf{X}}_{n,p} = \begin{bmatrix} \mathbf{x}_p^{(n)}(0) & \mathbf{x}_p^{(n)}(1) & \dots & \mathbf{x}_p^{(n)}(M-1) \\ \mathbf{x}_p^{(n)}(M-1) & \mathbf{x}_p^{(n)}(0) & \dots & \mathbf{x}_p^{(n)}(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_p^{(n)}(1) & \mathbf{x}_p^{(n)}(2) & \dots & \mathbf{x}_p^{(n)}(0) \end{bmatrix} \tag{6.7}$$

with

$$\mathbf{x}_p^{(n)} = \begin{bmatrix} \mathbf{x}_p^{(n)}(0) \\ \vdots \\ \mathbf{x}_p^{(n)}(M-1) \end{bmatrix} = \begin{bmatrix} x[(n+1)L-pP-M+1] \\ \vdots \\ x[(n+1)L-pP] \end{bmatrix} \begin{matrix} \uparrow \\ \\ \downarrow \end{matrix} M, \tag{6.8}$$

equation 6.3 can be extended to

$$\bar{\mathbf{y}}_n = \sum_{p=0}^{\frac{L_{FB}}{P}-1} \bar{\mathbf{X}}_{n,p}^T \mathbf{w}_p^{(n)}. \tag{6.9}$$

Equation 6.9 can be written as a sum of circular convolutions :

$$\bar{\mathbf{y}}_n = \sum_{p=0}^{\frac{L_{FB}}{P}-1} \mathbf{w}_p^{(n)} \circledast \mathbf{x}_p^{(n)}. \tag{6.10}$$

It is observed that only the last $M - P + 1$ elements of $\bar{\mathbf{y}}_n$ are meaningful and correspond to $\{y[(n+1)L - M + P], \ldots, y[(n+1)L]\}$. As we want to compute the $L \times 1$ vector $\mathbf{y}_n$ it follows that $M - P + 1 \geqslant L$, being equivalent to

$$M \geqslant L + P - 1. \tag{6.11}$$

As a consequence,

$$\mathbf{y}_n = \begin{bmatrix} \mathbf{0} & \mathbf{I}_L \end{bmatrix} \bar{\mathbf{y}}_n. \tag{6.12}$$

Remark also that the $M \times M$ matrix $\bar{\mathbf{X}}_{n,p}$ defined in Eq. 6.7 is right–circulant. As a consequence, the diagonal elements of $\mathbf{F}\bar{\mathbf{X}}_{n,p}^T\mathbf{F}^{-1}$ are the DFT coefficients of the first row of $\bar{\mathbf{X}}_{n,p}$, where $\mathbf{F}$ is the $M \times M$ DFT matrix (see also appendix A).

Applying this property to $\bar{\mathbf{X}}_{n,p}^T$ it follows from Eqs. 6.9 and 6.12 that

$$\mathbf{y}_n = \begin{bmatrix} \mathbf{0} & \mathbf{I}_L \end{bmatrix} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \mathbf{F}^{-1}(\mathbf{F}\bar{\mathbf{X}}_{n,p}^T\mathbf{F}^{-1})\mathbf{F}\mathfrak{w}_p^{(n)} \tag{6.13}$$

$$= \begin{bmatrix} \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)}\underline{\mathbf{w}}_p^{(n)}, \tag{6.14}$$

with

$$\underline{\mathbf{w}}_p^{(n)} = \mathbf{F}\mathfrak{w}_p^{(n)} \tag{6.15}$$

$$= \mathbf{F} \begin{bmatrix} \mathbf{w}_p^{(n)} \\ \mathbf{0} \end{bmatrix} \begin{matrix} \updownarrow P \\ \updownarrow M-P \end{matrix} \tag{6.16}$$

and

$$\underline{\mathbf{X}}_p^{(n)} = \operatorname{diag}\left\{\mathbf{F}\mathbf{x}_p^{(n)}\right\} = \operatorname{diag}\left\{\mathbf{F}\begin{bmatrix} x[(n+1)L - pP - M + 1] \\ \vdots \\ x[(n+1)L - pP] \end{bmatrix}\right\}. \tag{6.17}$$

The procedure described by Eqs. 6.9 and 6.12 or 6.14 is the so–called overlap–save correction method used to retransform the circular convolutions into linear convolutions.

As the filtering operation $\hat{w} \star x$ is now done partly in the time (multiply–add) and partly in the frequency domain (element–wise multiply) —which can be verified from Eq. 6.14— a substantial cost reduction is expected from the use of fast convolution techniques (FFT) on the one hand and signal subsampling on the other hand.

For the weight updating part a similar reasoning can be followed. Starting from the weight updating of the BLMS algorithm (Eq. 2.41) and partitioning in the same way as in Eq. 6.3 it follows that

$$\mathbf{w}_p^{(n+1)} = \mathbf{w}_p^{(n)} + \mu_n \mathbf{X}_{n,p}^* \mathbf{e}_n. \tag{6.18}$$

If

$$\mathbf{e}^{(n)} = \begin{bmatrix} \mathbf{0}_{(M-L)\times 1} \\ \mathbf{e}_n \end{bmatrix} \tag{6.19}$$

is defined, it follows from Eqs. 6.6, 6.7 and 6.18 that

$$\mathfrak{w}_p^{(n+1)} = \mathfrak{w}_p^{(n)} + \mu_n \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \bar{\mathbf{X}}_{n,p}^{*} \mathbf{e}^{(n)} \tag{6.20}$$

$$= \mathfrak{w}_p^{(n)} + \mu_n \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{x}_p^{(n)*} \otimes \mathbf{e}^{(n)}, \tag{6.21}$$

which corresponds to a circular correlation. Equation 6.21 can also be written as a circular convolution

$$\mathfrak{w}_p^{(n+1)} = \mathfrak{w}_p^{(n)} + \mu_n \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \breve{\mathbf{x}}_p^{(n)*} \circledast \mathbf{e}^{(n)} \tag{6.22}$$

with

$$\breve{\mathbf{x}}_p^{(n)} = \begin{bmatrix} x[(n+1)L - M - pP + 1] \\ x[(n+1)L - pP] \\ \vdots \\ x[(n+1)L - M - pP + 2] \end{bmatrix}. \tag{6.23}$$

Observe that only the first $M - L + 1$ elements of $\breve{\mathbf{x}}_p^{(n)*} \circledast \mathbf{e}^{(n)}$ are meaningful. As $P$ values are needed to compute $\mathbf{w}_p^{(n+1)}$, it is required that $M - L + 1 \geqslant P$, from which again condition 6.11 follows.

From Eq. 6.20,

$$\mathbf{F}\mathfrak{w}_p^{(n+1)} = \mathbf{F}\mathfrak{w}_p^{(n)} + \mu_n \mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{F}^{-1} (\mathbf{F}\bar{\mathbf{X}}_{n,p}^{*}\mathbf{F}^{-1}) \mathbf{F}\mathbf{e}^{(n)}, \tag{6.24}$$

is obtained, which is equivalent to (see Eq. 6.15)

$$\underline{\mathbf{w}}_p^{(n+1)} = \underline{\mathbf{w}}_p^{(n)} + \mu_n \mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{F}^{-1} \underline{\mathbf{X}}_p^{(n)H} \mathbf{F}\mathbf{e}^{(n)}, \tag{6.25}$$

where $\mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{F}^{-1}$ performs the gradient overlap–save correction.

### 6.1.2 PBFDAF algorithm : equations and properties

Summarizing, for block $n$ the $L_{FB}$–taps fullband adaptive filter $\hat{w}^{(n)}[k]$ is partitioned in $\frac{L_{FB}}{P}$ parts[1] $\mathbf{w}_p^{(n)}$ of length $P$ each, which are transformed to the frequency domain :

$$\mathbf{w}_p^{(n)} \stackrel{\forall p}{=} \begin{bmatrix} \hat{w}^{(n)}[pP] \\ \vdots \\ \hat{w}^{(n)}[(p+1)P-1] \end{bmatrix}, \quad p = 0 : \frac{L_{FB}}{P} - 1 \tag{6.26}$$

$$\underline{\mathbf{w}}_p^{(n)} \stackrel{\forall p}{=} \mathbf{F} \begin{bmatrix} \mathbf{w}_p^{(n)} \\ \mathbf{0} \end{bmatrix} \begin{matrix} \updownarrow P \\ \updownarrow M-P \end{matrix} \tag{6.27}$$

$\mathbf{F}$ is the $M \times M$ DFT matrix.

The equations defining the overlap–save PBFDAF are :

$$\underline{\mathbf{X}}_p^{(n)} \stackrel{\forall p}{=} \mathrm{diag}\left\{ \mathbf{F} \begin{bmatrix} x[(n+1)L - pP - M + 1] \\ \vdots \\ x[(n+1)L - pP] \end{bmatrix} \right\} \Bigg\uparrow M, \quad p = 0 : \frac{L_{FB}}{P} - 1 \tag{6.28}$$

$$\mathbf{y}^{(n)} = \begin{bmatrix} \mathbf{0}_{M-L} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \underline{\mathbf{w}}_p^{(n)} \tag{6.29}$$

$$\mathbf{d}^{(n)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_n \end{bmatrix} \begin{matrix} \updownarrow M-L \\ \updownarrow L \end{matrix}, \qquad \mathbf{d}_n = \begin{bmatrix} d[nL+1] \\ \vdots \\ d[(n+1)L] \end{bmatrix} \Bigg\uparrow L \tag{6.30}$$

$$\mathbf{e}^{(n)} = \mathbf{d}^{(n)} - \mathbf{y}^{(n)} \tag{6.31}$$

$$\underline{\mathbf{w}}_p^{(n+1)} \stackrel{\forall p}{=} \underline{\mathbf{w}}_p^{(n)} + \mathbf{G}\boldsymbol{\Delta}\underline{\mathbf{X}}_p^{(n)*}\mathbf{F}\mathbf{e}^{(n)}, \quad p = 0 : \frac{L_{FB}}{P} - 1 \tag{6.32}$$

Each time an iteration of the PBFDAF is performed, as described by Eqs. 6.28–6.32, $L$ new $x$–samples are taken in, and $L$ new filter output samples $e$ are produced. Vector $\mathbf{d}_n$ contains the $L$ most recent values of near–end signal $d$. Parameter $L$ is called the block length, and hence the corresponding input/output delay is $2L-1$. Further, $M = P + L - 1 + \sigma$, with $\sigma$ an integer larger than or equal to zero.

Matrix $\boldsymbol{\Delta}$ is called the stepsize matrix and $\mathbf{G}$ defines the type of updating. The updating can be constrained or unconstrained, depending on whether the gradient overlap–save correction is performed or not. More information on $\boldsymbol{\Delta}$ and $\mathbf{G}$ can be found in section 6.1.3 and 6.1.4 respectively.

If $P$ is divisible by $L$ —which is typically the case in practice— it follows that $\underline{\mathbf{X}}_p^{(n)} = \underline{\mathbf{X}}_0^{(n-pP/L)}$. Hence, equation 6.28 requires only 1 DFT operation, i.e. the

[1] It is assumed that $\frac{L_{FB}}{P} \in \mathbb{N}_0$. Otherwise, $\hat{w}^{(n)}[k]$ has to be padded with zeros.

one that corresponds to $p = 0$. The other $\underline{\mathbf{X}}_p^{(n)}$ can be recovered from previous iterations. In most practical applications $P = L$, $\sigma = 1$ and $M$ is a power of 2. Notice that the PBFDAF algorithm reduces to the FDAF presented in section 2.3.2 if $P = L_{FB}$.

The block length $L$ of the PBFDAF can be adjusted, resulting in a cheap adaptive filter with acceptable processing delay. Consider for example a realistic acoustic echo cancellation setup in which an unknown system $w$ is modelled using the PBFDAF with an adaptive filter $\hat{w}$ having 1000 equivalent filter taps in the time domain. If the signals are sampled at 8 kHz and a block length $L = 128$ is applied, the delay is 32 ms, which is acceptable, compared to the delay of 250 ms introduced by the FDAF algorithm as shown in section 2.3.2.

### 6.1.3 Normalization

The stepsize matrix $\mathbf{\Delta}$ controls the adaptation speed. If $\mathbf{\Delta} = \mu\mathbf{I}_M$, with $\mu$ a positive constant *no normalization* is applied. It was proven in section 6.1.1 that the unnormalized PBFDAF algorithm corresponds to an exact translation of Block–LMS to the frequency domain.

If signals such as e.g. speech are involved with a time–varying short–time energy, a *global normalization* is preferred. Therefore, $\mathbf{\Delta} = \mu_n\mathbf{I}_M$ is defined with[2]

$$\mu_n = \frac{\mu}{\frac{1}{M}\sum_{m=0}^{M-1}\sum_{p=0}^{\frac{L_{FB}}{P}-1}\underline{\mathbf{X}}_p^{(n)}(m)\underline{\mathbf{X}}_p^{(n)*}(m)} \tag{6.33}$$

$$= \frac{\mu}{\sum_{m=0}^{M-1}\sum_{p=0}^{\frac{L_{FB}}{P}-1}x[(n+1)L - pP - m]^2}. \tag{6.34}$$

If $\mathbf{\Delta} = \mathrm{diag}\{\boldsymbol{\mu}_s^{(n)}\}$ with $\boldsymbol{\mu}_s^{(n)}$ a vector of frequency and/or block dependent stepsizes the algorithm performs a *subband normalization*. If the stepsize matrix is chosen equal to

$$\mathbf{\Delta} = \mu\mathbf{P}_n^{-1} \tag{6.35}$$

with diagonal matrix $\mathbf{P}_n$ defined as

$$\mathbf{P}_n = \sum_{p=0}^{\frac{L_{FB}}{P}-1}\underline{\mathbf{X}}_p^{(n)}\underline{\mathbf{X}}_p^{(n)*} \tag{6.36}$$

the adaptation stepsize $\mu$ is divided by the subband energy in each frequency bin. As the PBFDAF —as will be shown in section 6.2— takes on the form of an oversampled subband adaptive filter the convergence can be improved in this way.

[2] $\underline{\mathbf{X}}_p^{(n)}(m)$ is the $m$–th diagonal element of diagonal matrix $\underline{\mathbf{X}}_p^{(n)}$.

If $P$ is divisible by $L$, it follows that $\underline{\mathbf{X}}_p^{(n)} = \underline{\mathbf{X}}_0^{(n-pP/L)}$ such that

$$\mathbf{P}_n = \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_0^{(n-pP/L)} \left(\underline{\mathbf{X}}_0^{(n-pP/L)}\right)^* \tag{6.37}$$

and

$$\mathbf{P}_{(n-\frac{P}{L})} = \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_0^{(n-(p+1)P/L)} \left(\underline{\mathbf{X}}_0^{(n-(p+1)P/L)}\right)^*. \tag{6.38}$$

Hence,

$$\mathbf{P}_n = \mathbf{P}_{(n-\frac{P}{L})} + \underline{\mathbf{X}}_0^{(n)} \underline{\mathbf{X}}_0^{(n)^*} - \underline{\mathbf{X}}_0^{(n-L_{FB}/L)} \left(\underline{\mathbf{X}}_0^{(n-L_{FB}/L)}\right)^* \tag{6.39}$$

can be used to update $\mathbf{P}_n$ instead of Eq. 6.36. Although numerically less robust the complexity is significantly reduced in this way for large values of $\frac{L_{FB}}{P}$.

Finally, remark that Eq. 6.33 can also be written as

$$\mu_n = \frac{\mu}{\frac{1}{M}\sum_{m=0}^{M-1} \mathbf{P}_n(m)}. \tag{6.40}$$

### 6.1.4 Constrained versus unconstrained updating

Just like with the FDAF there exist two variants of the algorithm, called the constrained and the unconstrained PBFDAF. Matrix $\mathbf{G}$ in equation 6.32 defines the type of updating : for the unconstrained PBFDAF, $\mathbf{G} = \mathbf{I}_M$, while for the constrained PBFDAF

$$\mathbf{G} = \mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{M-P} \end{bmatrix} \mathbf{F}^{-1}. \tag{6.41}$$

The unconstrained updating requires 3 (I)DFTs per iteration, provided $P$ is divisible by $L$, whereas the constrained PBFDAF is more expensive, having an extra $\frac{2L_{FB}}{P}$ (I)DFTs to compute. The latter on the other hand has better convergence properties. This can be verified in figure 10.2 where the PBFDAF was tested in an acoustic echo cancellation setup. Note that for the FDAF one has $L_{FB} = P$ and hence the number of (I)DFTs for unconstrained and constrained updating are respectively 3 and 5.

As for the unconstrained PBFDAF $\mathbf{G} = \mathbf{I}_M$, the filters $\underline{\mathbf{w}}_p^{(n)}$ will in general not maintain the format initially imposed by Eq. 6.27, as will be explained more in detail in the next section. To reconstruct $\hat{w}^{(n)}$ from $\underline{\mathbf{w}}_p^{(n)}$

$$\begin{bmatrix} \hat{w}^{(n)}[0] \\ \vdots \\ \hat{w}^{(n)}[L_{FB}-1] \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{L_{FB}} & \mathbf{0} \end{bmatrix} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \begin{bmatrix} \mathbf{0}_{pP\times 1} \\ \mathbf{F}^{-1}\underline{\mathbf{w}}_p^{(n)} \\ \mathbf{0}_{(L_{FB}-(p+1)P)\times 1} \end{bmatrix} \tag{6.42}$$

should then be applied.

Recall that from a computational complexity point of view, constrained updating is less favorable than unconstrained updating. Modified schemes trying to reach the performance of the constrained algorithm but almost operating at the cost of the unconstrained PBFDAF can be called for.

In [146] for instance the gradient corrector of the FDAF algorithm was approximated by a cosine window leading to a modified scheme having the complexity of the unconstrained FDAF with a performance almost as good as the constrained FDAF. This alternative will only work if the energy of the unknown system response decays sufficiently with time, which is a reasonable assumption if an acoustic setup is envisaged.

Other solutions rely on partial or round–robin based gradient correction. In the case of the partially update scheme only the gradients corresponding to the first filter partitions are being restored. The first part of the impulse response of the unknown system typically contains most of the energy and a significant performance increase is therefore expected for a small additional cost.

In the round–robin scheme a selected number of partitions are updated in turn every now and then. Also a mixed option is possible in which a first partition and a part of the remaining tail is updated. A round–robin scheme can then be used for the tail.

Each of these alternatives will improve the convergence speed, but the performance of the constrained algorithm will only be reached when none of the restore operations is left out. Further extensions try to reduce the implementation cost or to enhance the performance by decoupling the filtering and weight updating part, each with its own optimal partitioning. Unequal partitioning of $\hat{w}$ is another option [36] [37]. Finally, a multi–channel version of the PBFDAF was reported in [7] [139].

### 6.1.5 Ambiguity compensation for $M > P + L - 1$

The PBFDAF is most attractive from a computational complexity point of view if $M$ is a power of 2, as an efficient FFT–based implementation is then possible. Hence, $M$ is typically chosen equal to $M = P + L - 1 + \sigma = 2^r$ with $\sigma > 0$ and $\sigma, r \in \mathbb{N}$. In that case the last $L + \sigma$ elements of $\mathbf{y}^{(n)}$ are significant and not affected by the circular convolution. However, an ambiguity can then occur with the unconstrained PBFDAF, such that the subband filters are not uniquely defined anymore.

If unconstrained updating is applied it is not guaranteed that the vectors $\underline{\mathbf{w}}_p^{(n)}$ will obtain the format set in equation 6.6 or 6.27.

**Theorem 6.1** *A random value $\epsilon$ can appear as an extra $(P+q+1)$–st element of $\mathbf{w}_p^{(n)}$ in equation 6.27 if $q = 0 : \sigma - 1$ and $p = 0 : \frac{L_{FB}}{P} - 2$ and as long as it is compensated for, e.g. at the $(q+1)$–st element of $\mathbf{w}_{p+1}^{(n)}$.*

**Proof :** The proof can be found in appendix C.4.

As $\epsilon$ is random its amplitude cannot be controlled, so there is a possibility of running into numerical trouble. For the constrained PBFDAF the format set in equation 6.6 or 6.27 is re–imposed in each iteration step by the weight updating formula 6.32, hence no ambiguity is found. It can not occur either with the FDAF as $\frac{L_{FB}}{P} = 1$ in that case and hence there is no $p$ for which $0 \leqslant p \leqslant \frac{L_{FB}}{P} - 2$.

**Theorem 6.2** *The ambiguity can be removed by taking into account $L+\sigma$ instead of $L$ samples to determine the error $\mathbf{e}^{(n)}$. Therefore Eqs. 6.29 and 6.30 are slightly changed into*

$$\mathbf{d}^{(n)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_n \end{bmatrix} \begin{matrix} \updownarrow P-1 \\ \updownarrow L+\sigma \end{matrix}, \qquad \mathbf{d}_n = \begin{bmatrix} d[nL-\sigma+1] \\ \vdots \\ d[(n+1)L] \end{bmatrix} \Big\updownarrow L+\sigma \tag{6.43}$$

$$\mathbf{y}^{(n)} = \begin{bmatrix} \mathbf{0}_{P-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{L+\sigma} \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \underline{\mathbf{w}}_p^{(n)}. \tag{6.44}$$

The other equations remain unchanged and hence the additional algorithmic cost is almost negligible.

**Proof :** The proof can be found in appendix C.5.

The ambiguity compensation has an additional advantage : a more accurate gradient estimate is now computed as $L+\sigma$ instead of $L$ estimates are being averaged. Block length $L$ has remained unaltered, so the tracking performance has not changed.

By applying Eqs. 6.43 and 6.44 the unconstrained PBFDAF converges to a set of filters for which Eq. 6.27 is fulfilled. Remark that during the adaptation basically all elements of $\mathbf{F}^{-1}\underline{\mathbf{w}}_p^{(n)}$ are non–zero and that Eq. 6.27 will only be satisfied upon convergence. Indeed, as the gradient error correction is omitted if unconstrained updating is applied, the unconverged unconstrained PBFDAF will model a time–varying system, not being a replica of the Block–LMS algorithm in the frequency domain.

**Experiment 6.1** An experiment was carried out in which a Gaussian white noise signal was taken as the input to a random 5–th order linear FIR system $w$. The

| without ambiguity compensation | | | with ambiguity compensation | | |
|---|---|---|---|---|---|
| $\mathbf{F}^{-1}\underline{\mathbf{w}}_0^{(n)}$ | $\mathbf{F}^{-1}\underline{\mathbf{w}}_1^{(n)}$ | $\mathbf{F}^{-1}\underline{\mathbf{w}}_2^{(n)}$ | $\mathbf{F}^{-1}\underline{\mathbf{w}}_0^{(n)}$ | $\mathbf{F}^{-1}\underline{\mathbf{w}}_1^{(n)}$ | $\mathbf{F}^{-1}\underline{\mathbf{w}}_2^{(n)}$ |
| 1.1462 | -0.2255 | -0.4285 | 1.1462 | -1.2892 | -0.1238 |
| 1.0435 | -1.0675 | 0.5837 | 1.0435 | -1.0675 | 0.5837 |
| -1.0637 | 0.3047 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |

Table 6.1: Random values $\epsilon$ do appear and create an ambiguity.

filter coefficients were $\{1.1462, 1.0435, -1.2892, -1.0675, -0.1238, 0.5837\}$. The unconstrained PBFDAF with $M = 4, L = P = 2$ and $L_{FB} = 6$ was used to identify the system with and without the ambiguity compensation discussed in this paragraph. It is observed that —as predicted— random values $\epsilon$ do appear. Each of the 3 extended filter partitions of length 4 are shown in table 6.1 both for the unconstrained PBFDAF with and without ambiguity compensation. Observe for instance that if the third coefficient of the most left column $-1.0637$ is added to the first element of the second column $-0.2255$, the first element of the fifth column, $-1.2892$ , also being the third coefficient of $w$, is obtained. Further, if instead of using Eq. 6.42 $w$ is estimated following Eq. 6.27, i.e. by just retaining the first $P$ coefficients from each column and forgetting about the $(P+1)$–the element a wrong estimate is obtained if no ambiguity compensation is applied. $\bigtriangledown$

## 6.2 The PBFDAF as a special case of subband adaptive filtering

This section aims at presenting the PBFDAF algorithm as a subband adaptive filter. Through a graphical representation of the equations defining the PBFDAF and a comparison with the general DFT modulated subband adaptive scheme shown in figure 5.3 it will appear that the PBFDAF can be viewed as a DFT modulated subband adaptive filtering scheme to which an extra error correction module has been added.

Figure 6.1 represents the PBFDAF algorithm in a graphical way. It is indicated how Eqs. 6.28, 6.29 and 6.30, which define the filtering part of the PBFDAF, can be situated in the scheme. In figure 6.1 it is assumed that $P$ is divisible by $L$ such that $\underline{\mathbf{X}}_p^{(n)}$ for $p > 0$ can be recuperated from previous block iterations, as was shown in section 6.1.2.

The adaptive filters have $\frac{L_{FB}}{P}$ non–zero filter taps as can be verified from Eq. 6.29. Remark that the signal buffers leading to the diagonal matrices $\underline{\mathbf{X}}_p^{(n)}$ in Eq. 6.29 are

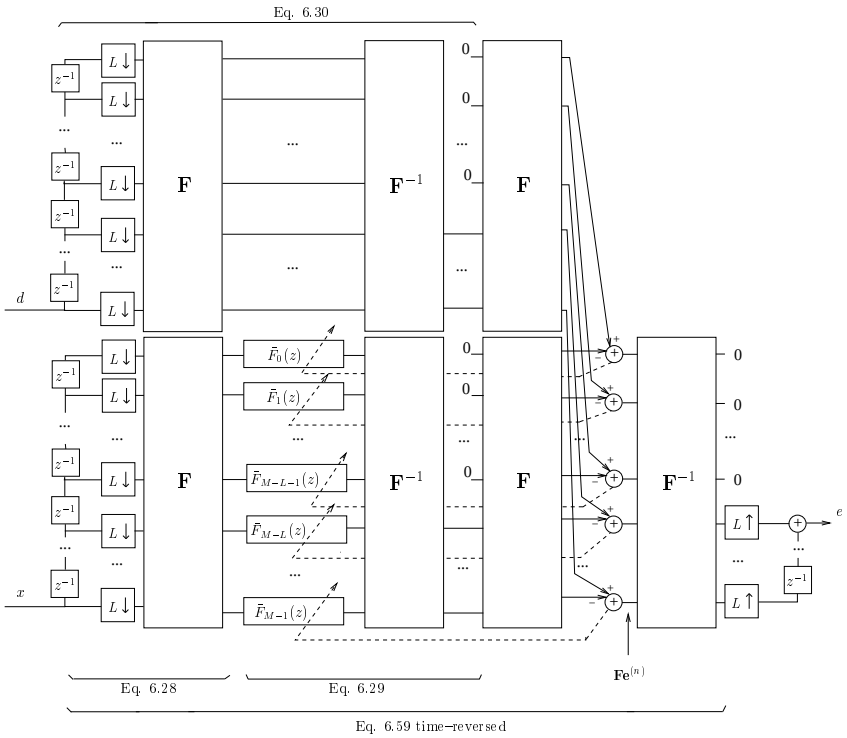


Figure 6.1: Partitioned block frequency–domain adaptive filtering : graphical representation

shifted $P$ time samples w.r.t. each other. In figure 6.1 however, as the signals are $L$–fold downsampled, the analysis filter bank passes an input to the subband filters every $L$ samples. The subband filters as depicted in figure 6.1 therefore depend on $z^{\frac{P}{L}}$. Hence, from Eq. 6.29 and figure 6.1

$$\bar{F}_i(z) = \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{w}}_p^{(n)}(i)\, z^{-p\frac{P}{L}} \tag{6.45}$$

is found, which illustrates how the vectors $\underline{\mathbf{w}}_p^{(n)}$ relate to the $M$ subband filters $\bar{F}_i(z)$.

By comparing figure 6.1 with figure 5.2 it appears that the PBFDAF can be viewed as a kind of subband adaptive system. Remark that the tapped delay lines are ordered differently. Further, instead of using the (aliased) subband errors —which could be found by subtracting the signals entering the left most IDFT block of the lower branch in figure 6.1 from those entering the IDFT block in the upper branch— to update the adaptive weights as in the standard subband scheme of figure 5.1, a

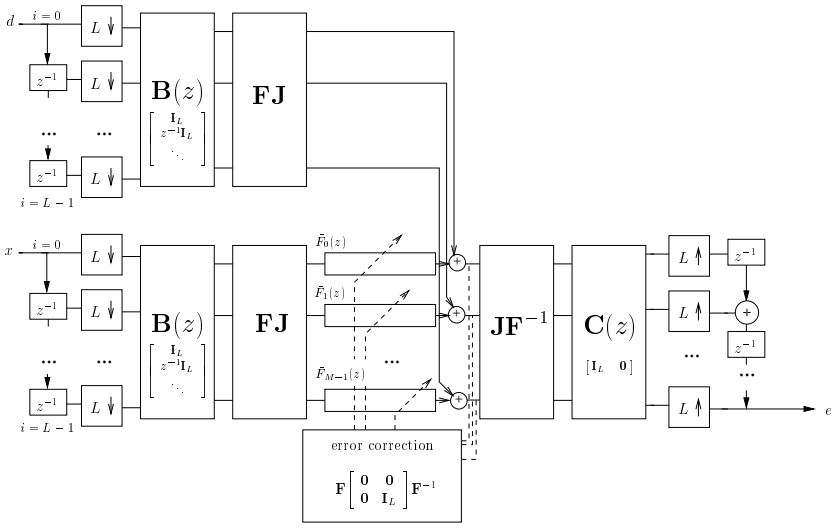


Figure 6.2: PBFDAF as an oversampled DFT modulated subband system : intermediate scheme

"restore" operation of the type

$$\mathbf{F} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \tag{6.46}$$

acts on the subband error vectors. Equation 6.46 represents the overlap–save compensation, which restores the aliased subband errors : the errors are computed in the time domain, are zero–padded and are then transformed back to the frequency–domain to adapt the subband filters.

Looking closer, figure 6.1 can be fit into the oversampled subband framework of figure 5.3. Figure 6.2 represents the PBFDAF with size $L$ (instead of size $M$) tapped delay lines and with the delay lines ordered upside–down. Observe that the exchange matrix $\mathbf{J}$ was inserted to compensate for the reversed order of the delay lines. An extra module called "error correction" was included, which represents the "restore operation", and which is an addition to the usual subband scheme.

The lower branch of figure 6.2 can be expressed as $\mathbf{C}(z)\mathbf{M}(z)\mathbf{B}(z)$ with

$$\begin{aligned} \mathbf{M}(z) &= \mathbf{J}\mathbf{F}^{-1}\mathrm{diag}\{\bar{F}_i(z)\}\mathbf{F}\mathbf{J} && (6.47)\\ &= \mathbf{F}^{-1}\mathbf{F}\mathbf{J}\mathbf{F}^{-1}\mathrm{diag}\{\bar{F}_i(z)\}\mathbf{F}\mathbf{J}\mathbf{F}^{-1}\mathbf{F} && (6.48)\\ &= \mathbf{F}^{-1}\mathrm{diag}\{F_i(z)\}\mathbf{F}. && (6.49) \end{aligned}$$

As $\mathrm{diag}\{\bar{F}_i(z)\}$ is diagonal $\mathbf{M}(z)$ is a circulant matrix, which again can be diagonalized by DFT operations. Hence, figure 6.2 can be redrawn, resulting in figure

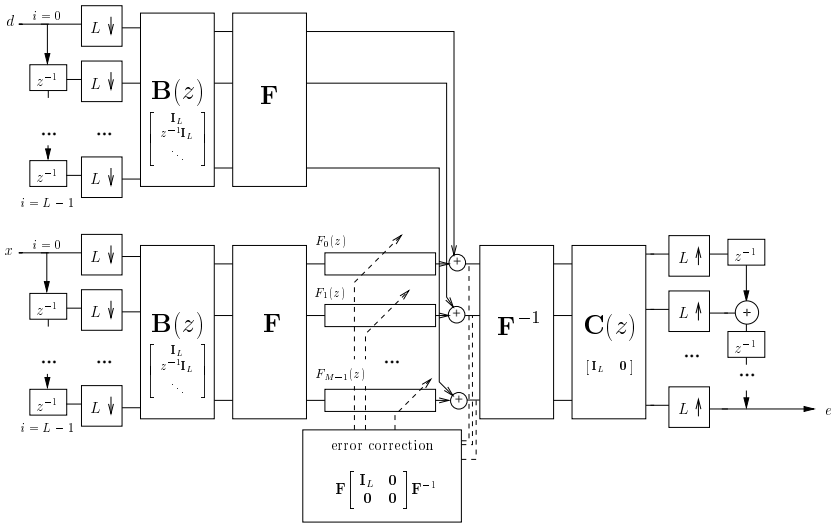


Figure 6.3: PBFDAF as an oversampled DFT modulated subband scheme

6.3 depicting the PBFDAF now completely in the subband "jargon". Remark that also the error correction was changed to cope with the time–reversed signals. In appendix C.6 the "time–reversed" PBFDAF is derived, which is an equivalent description of Eqs. 6.28–6.32 but with time–reversed signal vectors. It can be verified from the equations derived in appendix C.6 that the error correction now equals

$$\mathbf{F} \begin{bmatrix} \mathbf{I}_L & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{F}^{-1}. \tag{6.50}$$

The analysis prototype polyphase matrix $\mathbf{B}(z)$, which characterizes an $M$–band $L$–fold downsampled DFT modulated analysis filter bank, is a structured matrix satisfying equation 3.3. By comparing figure 6.2 with figure 6.1 it is observed that for the PBFDAF

$$\underbrace{\mathbf{B}(z)}_{M \times L} = \begin{bmatrix} \mathbf{I}_L \\ z^{-1}\mathbf{I}_L \\ \vdots \\ z^{-(\lfloor \frac{M-1}{L} \rfloor - 1)}\mathbf{I}_L \\ \mathbf{R}(z) \end{bmatrix} \tag{6.51}$$

in which $\mathbf{R}(z)$ is an $(M - L\lfloor \frac{M-1}{L} \rfloor) \times L$ matrix :

$$\mathbf{R}(z) = \begin{bmatrix} z^{-\lfloor \frac{M-1}{L} \rfloor} & \dots & 0 \\ \vdots & \ddots & \vdots \\ \dots & z^{-\lfloor \frac{M-1}{L} \rfloor} & \dots \end{bmatrix}. \tag{6.52}$$

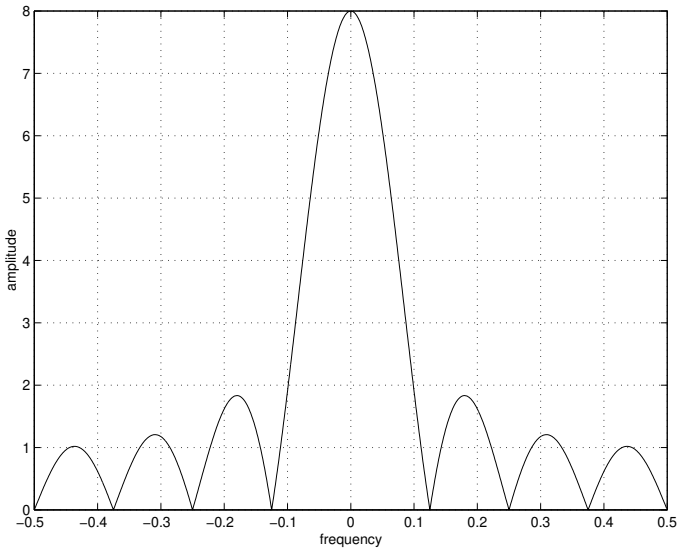


Figure 6.4: Frequency characteristics of the PBFDAF analysis prototype filter

Hence, from Eqs. 6.51 and 3.3 it follows that the analysis filter bank used by the PBFDAF is a simple DFT filter bank whose prototype $h_0[k]$ is defined as

$$H_{0\,k:K}(z) = 1 \qquad \text{if } k < M \tag{6.53}$$

$$= 0 \qquad \text{if } k \geqslant M. \tag{6.54}$$

**Example 6.1** The frequency response of the PBFDAF analysis prototype filter $h_0[k]$ has a sinc–like shape with a low frequency selectivity. The amplitude response of the analysis prototype filter is shown in figure 6.4 for $M = 8$ and $L = 4$. $\triangle$

For the synthesis part a similar derivation can be made. Compare again figure 6.2 with figure 6.1 and observe that the synthesis prototype polyphase matrix $\mathbf{C}(z)$ is given by :

$$\underbrace{\mathbf{C}(z)}_{L\times M} = \left[\, \mathbf{I}_L \quad \mathbf{0}_L \quad \ldots \quad \mathbf{0}_L \quad \mathbf{0} \,\right]. \tag{6.55}$$

From Eq. 3.8 it follows that

$$\mathbf{G}(z) = \frac{1}{M}\mathbf{F}^*\mathbf{C}(z)^T\mathbf{J} \tag{6.56}$$

$$= \frac{1}{M}\left(\mathbf{F}\begin{bmatrix}\mathbf{I}_L\\ \mathbf{0}\end{bmatrix}\right)^*\mathbf{J}. \tag{6.57}$$

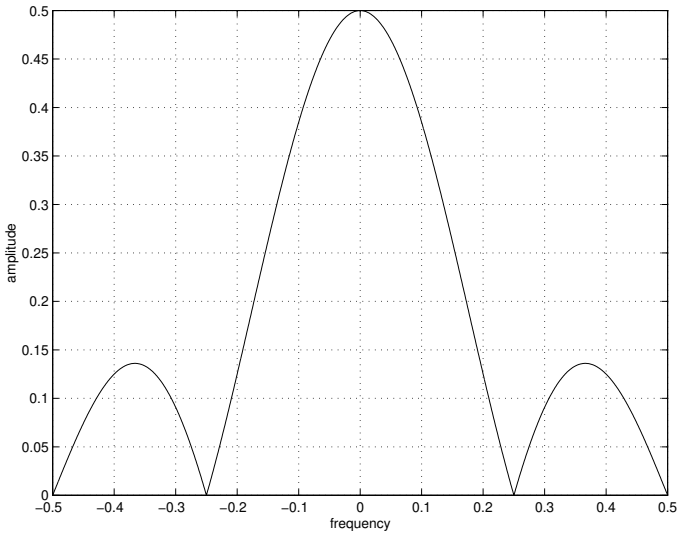


Figure 6.5: Frequency characteristics of the PBFDAF synthesis prototype filter

Hence, the synthesis filters can be obtained as time–reversed and complex conjugated versions of a DFT modulated filter bank with prototype polyphase components $E_{k:K}(z) = \frac{1}{M}$, $k < L$. The other polyphase components are zero. This follows also from Eq. 3.27.

**Example 6.2** The frequency response of the PBFDAF synthesis prototype $g_0[k]$ is shown in figure 6.5 also for $M = 8$ and $L = 4$. By comparing figure 6.5 with figure 6.4 it is observed that the main lobe of the synthesis prototype is twice as wide as that of the analysis prototype. △

A filter bank with sinc–shaped frequency amplitude response only shows a poor frequency selectivity. Inter–subband aliasing can not be avoided in this case. However, the overlap–save procedure apparently makes the adaptive filters converge properly. This is illustrated by means of the following simulation example.

**Experiment 6.2** A random FIR system of order 15 was generated and a white noise sequence was applied to the input. Three frequency–domain/subband algorithms are compared : the constrained PBFDAF, the unconstrained PBFDAF and a true subband adaptive filter. All adaptation schemes are unnormalized. The true subband algorithm uses the same filter banks as the (un)constrained PBFDAF but the overlap–save compensation on the error signal (cf. error correction in figure 6.2) is left out. In this way the subband filters are adapted on the subband errors directly.

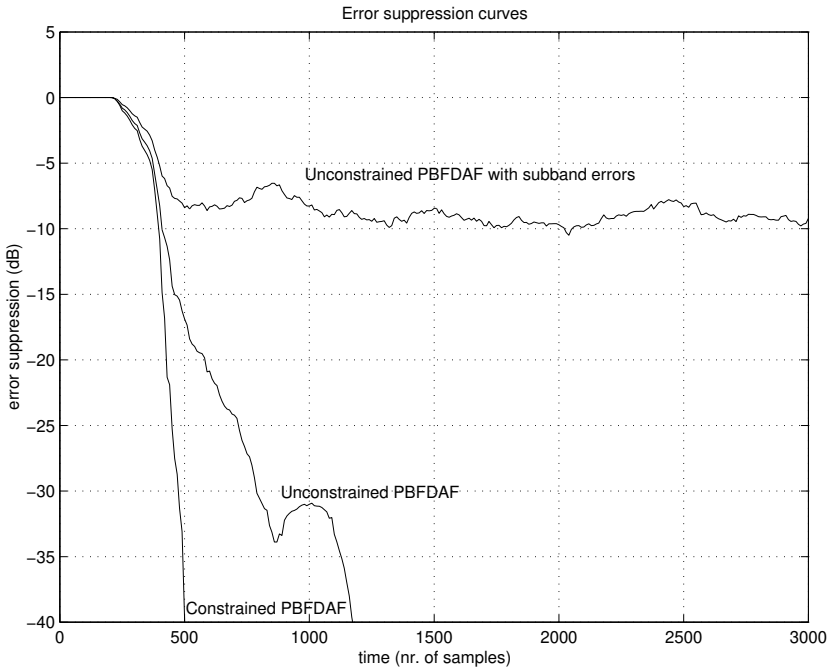


Figure 6.6: Convergence curves of 3 frequency–domain algorithms

Remark that all algorithms can be considered as subband adaptive systems. They all have 8 subbands, which are 4–fold subsampled, and the subband filters have 4 taps each. No anti–causal taps were added.
The convergence curves for each of the three algorithms are shown in figure 6.6. The constrained PBFDAF clearly outperforms the unconstrained PBFDAF, whereas for the subband algorithm without error compensation a considerable residual error remains. It is remarkable how unselective filter banks such as those used by the PBFDAF can lead to satisfactory results if appropriate error compensation is provided. ▽

## 6.3 PBFDAF : design criteria

In section 5.2 three design criteria on frequency selectivity, perfect reconstruction and perfect path modelling were defined for the subband adaptive filter. These conditions are necessary requirements for the satisfactory performance of the subband adaptive filter. In this section it will be verified whether these conditions are fulfilled for the PBFDAF.

The first criterion requires that the filter banks are *frequency selective*. A frequency selective analysis filter bank reduces the amount of aliasing components inserted in the subbands. Aliasing is known to have an negative impact on the convergence of the adaptive filters. It was shown in figure 6.4 and 6.5 that the PBFDAF uses poor filter banks with a sinc–like frequency amplitude response. So, it can be concluded that the first criterion is not fulfilled for the PBFDAF, although the algorithm doesn't seem to suffer from slow convergence due to aliasing, as can be observed in figure 6.6.

By filling in Eqs. 6.51 and 6.55 in Eq. 5.2 it is easily verified that the *perfect reconstruction condition* is fulfilled for the PBFDAF :

$$\left[\begin{array}{cccc} \mathbf{I}_L & \mathbf{0}_L & \dots & \mathbf{0} \end{array}\right] \left[\begin{array}{c} \mathbf{I}_L \\ z^{-1}\mathbf{I}_L \\ \vdots \\ \mathbf{R}(z) \end{array}\right] = \mathbf{I}_L. \tag{6.58}$$

As very simple filter banks are used no extra algorithmic delay is introduced by the filter banks.

For the *time–invariance condition* ensuring perfect path modelling equation 5.7 has to be satisfied. Eqs. 6.51 and 6.55 are therefore filled in in Eq. 5.7.

**Theorem 6.3** *For the constrained PBFDAF*

$$\left[\begin{array}{cccc} \mathbf{I}_L & \mathbf{0}_L & \dots & \mathbf{0} \end{array}\right] \mathbf{F}^{-1}\mathrm{diag}\{F_i(z)\}\mathbf{F} \left[\begin{array}{c} \mathbf{I}_L \\ z^{-1}\mathbf{I}_L \\ \vdots \\ \mathbf{R}(z) \end{array}\right] \tag{6.59}$$

*is a pseudo–circulant $L \times L$ matrix. In the case of unconstrained updating Eq. 6.59 is guaranteed to be pseudo–circulant only after convergence.*

As indicated in figure 6.1 equation 6.59 is represented by the lower branch of the adaptive scheme.

**Proof :** The proof is given in appendix C.7.

**Theorem 6.4** *A time–invariant finite–order system can be exactly modelled by the PBFDAF adaptive filter with a finite number of filter coefficients if $P$ is a multiple of $L$.*

**Proof :** The proof is given in appendix C.8.

It can be concluded that the PBFDAF fulfills the above mentioned conditions of perfect reconstruction and perfect path modelling. The first criterion requiring frequency selective filter banks is not fulfilled. Nevertheless, the PBFDAF shows fast convergence.

## 6.4 Implementation cost

In this section the implementation cost of the PBFDAF will be estimated following the guidelines of section 2.4. A cost analysis will be performed and the influence of the different algorithmic parameters on the total complexity will be examined.

### 6.4.1 Cost computation

The cost analysis of the PBFDAF will be based on formulas 6.28–6.32, which are also summarized in table 8.1 for $R = 1$. The results of this cost computation are included in the same table as well.

A list of parameters and assumptions :

- $f_s$ is the sampling rate
- $L_{FB}$ is the equivalent fullband adaptive filter length
- $L$ is the block length
- $P$ is the filter partition length. The effective length of the subband filters is therefore $\frac{L_{FB}}{P}$. It is assumed that $P$ is a divisor of $L_{FB}$.
- $M$ is the size of the (I)DFTs and is supposed to be a power of 2 such that fast signal transforms can be called for.
- all operations are done at the downsampled rate $\frac{f_s}{L}$
- only $\lfloor \frac{M}{2} \rfloor + 1$ out of $M$ subbands need to be processed as the input signals $x$ and $d$ and the unknown system $w$ are assumed to be real–valued

The detailed computations can be found in appendix C.9 and leads to the following cost estimates :

unnormalized unconstrained PBFDAF :

$$\boxed{\frac{f_s}{L}\left(6M\log_2 M + 8\frac{ML_{FB}}{P} + 16\frac{L_{FB}}{P} - 8M + L + 12\right) \text{ ops.}}$$

globally normalized unconstrained PBFDAF :

$$\boxed{\frac{f_s}{L}\left(6M\log_2 M + 8\frac{ML_{FB}}{P} + 16\frac{L_{FB}}{P} - 3M + L + 35\right)\ \texttt{ops}.}$$

subband–normalized unconstrained PBFDAF :

$$\boxed{\frac{f_s}{L}\left(6M\log_2 M + 8\frac{ML_{FB}}{P} + 16\frac{L_{FB}}{P} + L + 28\right)\ \texttt{ops}.}$$

unnormalized constrained PBFDAF :

$$\boxed{\frac{f_s}{L}\left(2M\log_2 M\left(3 + 2\frac{L_{FB}}{P}\right) + 4\frac{ML_{FB}}{P} + 22\frac{L_{FB}}{P} - 8M + L + 12\right)\ \texttt{ops}.}$$

globally normalized constrained PBFDAF :

$$\boxed{\frac{f_s}{L}\left(2M\log_2 M\left(3 + 2\frac{L_{FB}}{P}\right) + 4\frac{ML_{FB}}{P} + 22\frac{L_{FB}}{P} - 3M + L + 35\right)\ \texttt{ops}.}$$

subband–normalized constrained PBFDAF :

$$\boxed{\frac{f_s}{L}\left(2M\log_2 M\left(3 + 2\frac{L_{FB}}{P}\right) + 4\frac{ML_{FB}}{P} + 22\frac{L_{FB}}{P} + L + 28\right)\ \texttt{ops}.}$$

If $P$ is not a multiple of $L$ an extra

$$\frac{f_s}{L}\left(\left(\frac{L_{FB}}{P} - 1\right)(2M\log_2 M - 4M + 6)\right)\ \texttt{ops}. \tag{6.60}$$

are required to compute $\underline{\mathbf{X}}_p^{(n)}$.

For the normalization an extra

$$\frac{f_s}{L}\left(2\frac{ML_{FB}}{P} + 4\frac{L_{FB}}{P} - \frac{9M}{2} - 9\right)\ \texttt{ops}. \tag{6.61}$$

are needed if $\frac{L_{FB}}{P} \leqslant 2$ or if $P$ is not a multiple of $L$.

### 6.4.2 Cost evaluation and optimal parameter setting

Based on the results of the complexity analysis in appendix C.9 the influence of the algorithmic parameters on the total cost is now examined. As it is our objective to use the PBFDAF for acoustic signal enhancement realistic parameter settings were chosen for this application, in order to obtain relevant cost estimates.

**Experiment 6.3** In this first experiment the PBFDAF algorithm is evaluated w.r.t. its ability to model long impulse responses. It is assumed that a sampling frequency of 8000 Hz is applied and that $L_{FB} = 2048$ (cf. an acoustic impulse response of 256 ms @ 8 kHz). Furthermore, the FFT size is given by $M = 2P = 2L$. In table 6.2 a cost estimate is given in MFlops for a set of increasing block lengths $L$ and hence increasing input/output delays. A cost estimate was computed for the unnormalized (un)constrained PBFDAF and the subband–normalized (un)constrained PBFDAF.

| block length | in/out delay | unconstr. unnorm. | constr. unnorm. | unconstr. sub.–norm. | const. sub.–norm. |
|---|---|---|---|---|---|
| $L = 2$ | 0.375 ms | 196.7 | 286.8 | 196.9 | 287.0 |
| $L = 4$ | 0.875 ms | 82.1 | 153.8 | 82.3 | 154.0 |
| $L = 16$ | 3.88 ms | 17.8 | 50.9 | 17.9 | 51.0 |
| $L = 64$ | 15.9 ms | 4.71 | 17.0 | 4.84 | 17.2 |
| $L = 128$ | 31.9 ms | 2.71 | 9.89 | 2.84 | 10.0 |
| $L = 256$ | 63.9 ms | 1.77 | 5.87 | 1.90 | 6.00 |
| $L = 1024$ | 256 ms | 1.19 | 2.47 | 1.31 | 2.59 |
| $L = 2048$ | 512 ms | 1.16 | 1.86 | 1.25 | 1.95 |

Table 6.2: Cost estimate in MFlops for the PBFDAF as a function of $L$

The last line of the table corresponds to the Frequency–Domain Adaptive Filter (FDAF) presented in section 2.3.2. It is observed that a large complexity reduction can be obtained for large values of $L$ at the expense however of a considerable algorithmic delay. If large delays are undesired $L$ can be decreased and block partitioning is called for.
Remark that the constrained PBFDAF is clearly more expensive than its unconstrained counterpart. The extra cost for the normalization on the other hand is rather small in general.
Finally, it is observed that a real–time implementation of the PBFDAF algorithm is feasible, as realistic cost estimates are obtained for acceptable algorithmic delays. In the framework of this thesis an unconstrained PBFDAF–based long acoustic echo cancellation system was implemented in real time on a DSP. Further details and real cost measurements on the DSP are presented in section 10.3. $\bigtriangledown$

**Experiment 6.4** In figure 6.7 a cost estimate for the unnormalized PBFDAF is plotted against the fullband filter length $L_{FB}$ for both the unconstrained and the constrained updating mode. It was further assumed that $M = 2P = 2L = 128$.
It appears that the cost increases quasi–linearly for realistic values of $L_{FB}$. To obtain realistic experimental results the fullband filter length was set equal to $L_{FB} = P \left\lceil \frac{L_{FB}}{P} \right\rceil$. Hence, steps appear in the curves indicating that an extra filter partition was assigned as the order of the fullband adaptive filter $\hat{w}$ increases. $\bigtriangledown$

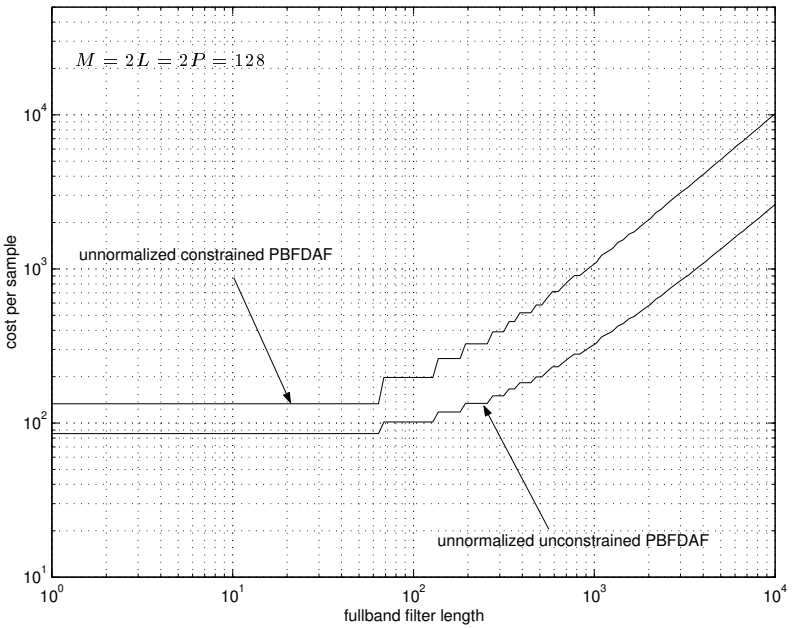


Figure 6.7: Cost estimate for the PBFDAF as a function of $L_{FB}$

In a practical design the block length $L$ is constrained by the maximum acceptable input/output delay. In an acoustic echo cancellation setup with $f_s$=8 kHz for instance the maximum delay could be 16 ms, i.e. $L$ is constrained to be smaller or equal to 64. A value for $P$ is then preferred that e.g. maximizes the complexity gain w.r.t. the LMS algorithm :

$$P = \underset{P}{\text{argmax}} \; \frac{\text{cost LMS}}{\text{cost PBFDAF}}.$$

**Experiment 6.5** In this experiment the optimal filter partition length $P$ is computed that maximizes the complexity gain w.r.t. the LMS algorithm. The block length $L$ was constant and set to 64. The DFT size was chosen equal to $M = P + L$ as it is required that $M \geqslant P + L - 1$. The implementation cost will show a local minimum whenever $M$ is a power of 2 and when $P$ is divisible by $L$.
In figure 6.8 the complexity gain w.r.t. LMS is plotted for the subband–normalized unconstrained PBFDAF as a function of $P$. The results for the subband–normalized constrained PBFDAF are shown in figure 6.9. The different curves correspond to $L_{FB} = 16$, 64, 256 and 2048.
The non–smoothness in these curves again comes from the fact that the fullband filter is zero–padded until $L_{FB}$ is a multiple of $P$. "Spikes" correspond to situations for which $P$ is divisible by $L$. Highlighted ($\star$) parts on the curves refer to setups

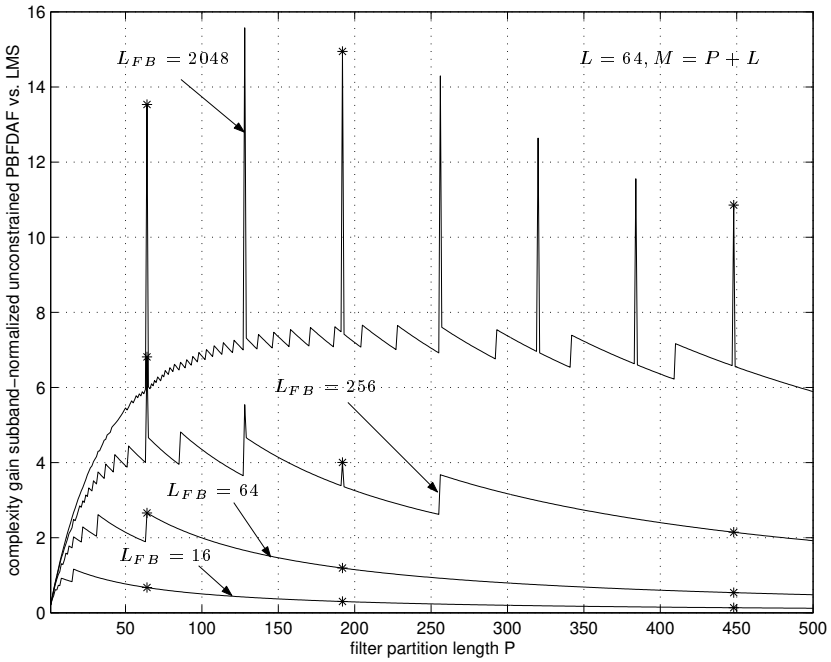


Figure 6.8: Complexity gain of the subband–normalized unconstrained PBFDAF w.r.t. LMS as a function of $P$

for which $M$ is a power of 2. For values of $M$ not being a power of 2 the complexity gain is over–estimated as fast signal transforms were assumed throughout the experiment. Apparently the "classical" PBFDAF with $L = P = \frac{M}{2}$ is preferred, except for large values of the fullband filter length, e.g. $L_{FB} = 2000$. In general, one can state that $L = P = \frac{M}{2}$ is a good choice. Only for large fullband filter lengths, $\frac{P}{L} > 1$ leads to a larger complexity gain. $\triangledown$

**Experiment 6.6** In a last experiment the parameter set $(L, P, M)$ that maximizes the complexity gain w.r.t. the LMS algorithm was computed as a function of the maximum acceptable algorithmic delay. The experiment was carried out for both the constrained and the unconstrained subband–normalized PBFDAF. It was assumed that the adaptive filter operates at a sampling rate of 8 kHz. The delay was changed between 1 and 519, corresponding to 0.125 ms and 64.9 ms respectively. For a given maximum delay the parameters $L$ and $P$ were changed over a broad range to find the optimum. The DFT size was set equal to $M = 2^{\lceil \log_2(L+P-1) \rceil}$. The experiment was carried out for a set of realistic fullband filter lengths.
The maximum complexity gain that could be obtained for a given delay is plotted in figure 6.10 and 6.11. Figure 6.10 corresponds to the subband–normalized

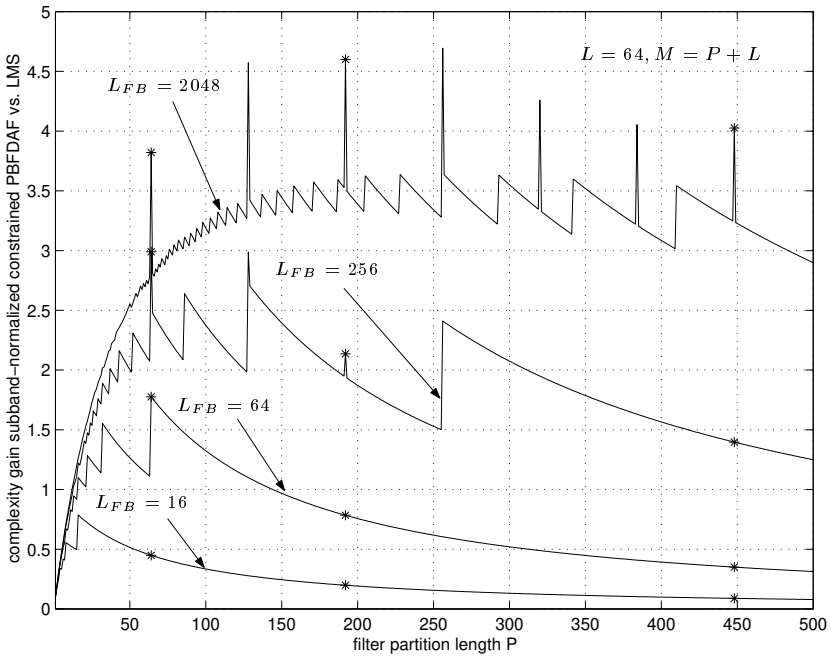


Figure 6.9: Complexity gain of the subband–normalized constrained PBFDAF w.r.t. LMS as a function of $P$

unconstrained PBFDAF and in figure 6.11 the results are shown for the subband–normalized constrained PBFDAF. The larger $L_{FB}$ the larger the complexity gain. Remark the non–smoothness in the curves.
In figure 6.12 the optimal $P/L$ is shown for the subband–normalized unconstrained PBFDAF, which maximizes the complexity gain as a function of the maximum delay. It is observed that $P/L$ increases as the block length decreases and as $L_{FB}$ becomes larger, which confirms the conclusions made based on experiment 6.5.
Finally, in figure 6.13 the optimal block length $L$ was plotted for the subband–normalized unconstrained PBFDAF for which $L_{FB} = 2048$. The curve rises linearly on the average, as could be expected. Nevertheless, deviations are found, which again illustrates the complex interrelation between the algorithmic parameters of the PBFDAF on the one hand and the global cost on the other hand. $\triangledown$

## 6.5 Conclusions

In this chapter the Partitioned Block Frequency–Domain Adaptive Filter (PBFDAF) was discussed. It is known from the literature that with the PBFDAF

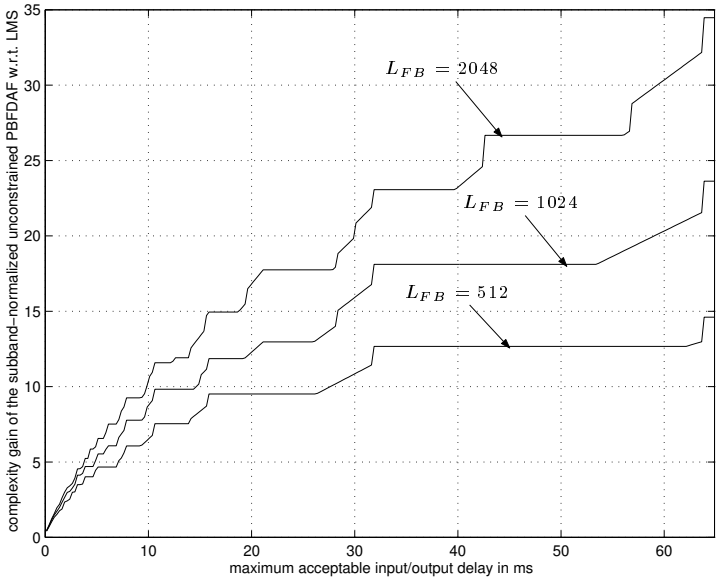


Figure 6.10: Complexity gain of the subband–normalized unconstrained PBFDAF w.r.t. LMS as a function of the maximum processing delay

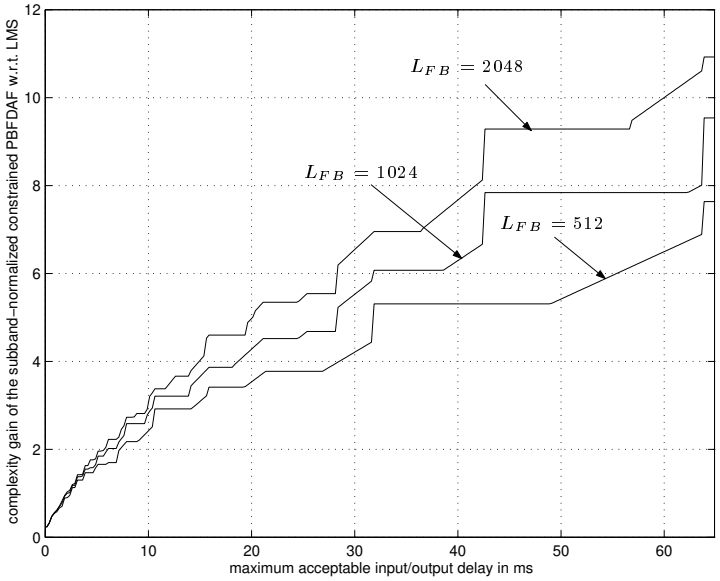


Figure 6.11: Complexity gain of the subband–normalized constrained PBFDAF w.r.t. LMS as a function of the maximum processing delay

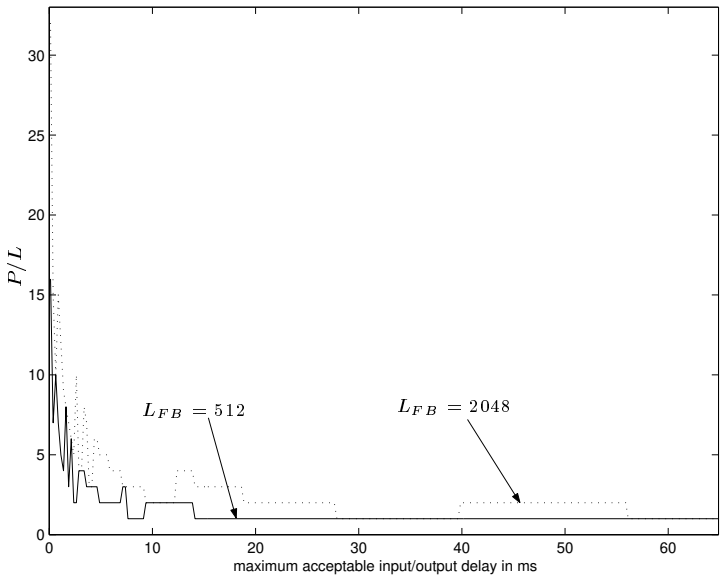


Figure 6.12: Optimal $P/L$ that maximizes the complexity gain for the subband–normalized unconstrained PBFDAF as a function of the maximum processing delay

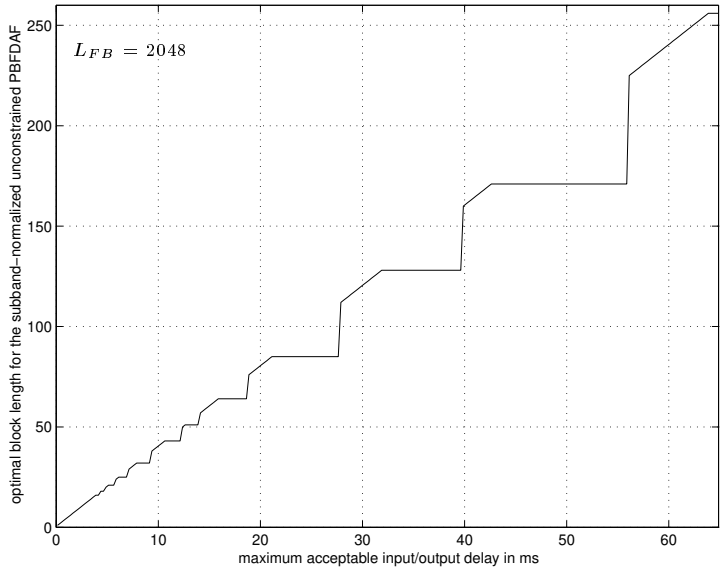


Figure 6.13: Optimal block length $L$ that maximizes the complexity gain for the subband–normalized unconstrained PBFDAF as a function of the maximum processing delay

a substantial cost reduction w.r.t. the LMS algorithm can be obtained without introducing unacceptable algorithmic delays as was the case with the FDAF discussed in section 2.3.2.

In section 6.1 it was shown how the PBFDAF algorithm can be derived from the Block–LMS adaptive filter and some of its properties were discussed. We showed that an ambiguity can occur with the unconstrained PBFDAF, which can easily be compensated for.

Based on a graphical representation of the algorithm it was shown in section 6.2 that the PBFDAF fits into the subband adaptive framework presented in chapter 5. Although the filter banks used by the PBFDAF have a rather poor frequency selectivity, thanks to an extra error correction module the algorithm has a superior convergence behavior compared to standard subband adaptive filters.

It was shown in section 6.3 that two out of the three design criteria for subband adaptive systems, which were formulated in chapter 5, are fulfilled for the PBFDAF. Although the first criterion requiring frequency selective filter banks is not fulfilled, the PBFDAF convergences remarkably fast.

In section 6.4 a cost analysis was performed for the PBFDAF. It was illustrated how the global cost changes as a function of different algorithmic parameters. It appears that for realistic parameter settings the PBFDAF is a cheaper alternative to the LMS algorithm.

# Chapter 7

# Fullband Error Adaptation Scheme

From experiment 5.5 and figure 5.10 it was observed that despite their low implementation cost frequency–domain adaptive filters such as the PBFDAF clearly outperform standard subband systems in terms of convergence behavior and modelling capabilities. It is therefore believed that a class of subband adaptive systems can be defined that adopt mechanisms and properties of the frequency–domain approach and in this way approximate the performance of the PBFDAF at a low implementation cost.

It was proven in section 6.3 that for the PBFDAF algorithm two out of three design criteria for subband adaptive systems are fulfilled. These criteria were defined in section 5.2 and are necessary conditions to let the standard subband adaptive filter, as presented in figure 5.1, operate properly. Apart from the design criteria it was shown in section 6.2 that the PBFDAF can be described as a special subband adaptive filter having an extra error correction module doing the overlap–save compensation. Thanks to the error correction the subband filters can be updated with aliasing–free error signals, which leads to improved performance.

In this chapter we extend the error correction of the frequency–domain approach to subband adaptive systems in general, which leads to a novel fullband error adaptation scheme. The alternative adaptation scheme adjusts the subband filters based on the fullband error instead of using the subband errors as in a classical subband adaptive system. The fullband error adaptation scheme is presented in section 7.1.

In section 7.2 the computational complexity of the fullband error adaptation scheme is discussed. Thanks to the fullband error adaptation the convergence behavior improves w.r.t. the standard subband adaptive filter. The computational cost is

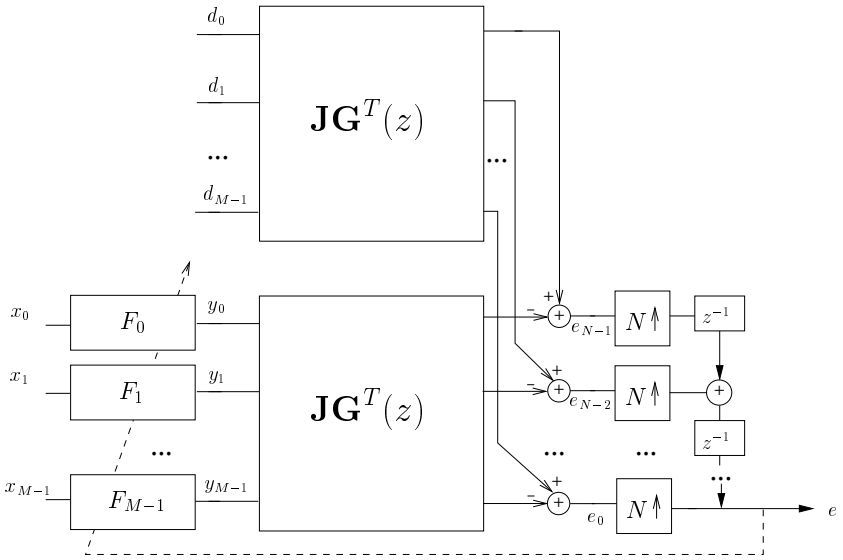


Figure 7.1: Oversampled subband adaptive filter : synthesis part

higher however as the subband errors pass through the synthesis filter bank twice.

It is further shown in section 7.3 that the weight update mechanism of the unconstrained PBFDAF algorithm, which was discussed in chapter 6, corresponds to that of the fullband error adaptation algorithm, as long as the filter partition length $P$ is a multiple of the block length $L$. This proves that the fullband error adaptation algorithm can be considered as an extension of the PBFDAF overlap–save error correction mechanism to a more general class of subband adaptive filters.

The conclusions are presented in section 7.4.

## 7.1 Fullband error adaptation

In this section an alternative adaptation scheme is presented that adjusts the subband adaptive filters $F_m$ as shown in figure 7.1 based on the fullband error $e$ instead of the subband error signals $\epsilon_m = d_m - y_m$, as is normally done in the standard subband scheme presented in figure 5.2. In order to derive the fullband error adaptation algorithm several signal vector and matrices need to be defined first.

The subband signals $x_m$ are filtered with the subband filters $F_m$, which can be

written as

$$\underbrace{\begin{bmatrix} \mathbf{y}_0^{(n)} \\ \vdots \\ \mathbf{y}_{M-1}^{(n)} \end{bmatrix}}_{\mathbf{y}_n} = \underbrace{\begin{bmatrix} \mathbf{X}_0^{(n)} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{X}_{M-1}^{(n)} \end{bmatrix}}_{\mathbf{X}_n} \underbrace{\begin{bmatrix} \mathbf{f}_0^{(n)} \\ \vdots \\ \mathbf{f}_{M-1}^{(n)} \end{bmatrix}}_{\mathbf{w}_n}, \tag{7.1}$$

in which (see figure 7.1)

$$\mathbf{X}_m^{(n)} = \underbrace{\begin{bmatrix} x_m[n-L_p+1] & \cdots & x_m[n-L_p-L_{SB}+2] \\ \vdots & \ddots & \vdots \\ x_m[n] & \cdots & x_m[n-L_{SB}+1] \end{bmatrix}}_{L_{SB}} \Bigg\updownarrow L_p, \tag{7.2}$$

and

$$\mathbf{f}_m^{(n)} = \begin{bmatrix} f_m^{(n)}[0] \\ \vdots \\ f_m^{(n)}[L_{SB}-1] \end{bmatrix} \Bigg\updownarrow L_{SB}, \quad \mathbf{y}_m^{(n)} = \begin{bmatrix} y_m[n-L_p+1] \\ \vdots \\ y_m[n] \end{bmatrix} \Bigg\updownarrow L_p. \tag{7.3}$$

$\mathbf{w}_n$ is the adaptive filter vector. $L_{SB}$ is the length[1] of the subband adaptive filters and $L_p$ is the length of the synthesis bank polyphase filters. Further, define

$$\underbrace{\begin{bmatrix} e_0[n] \\ \vdots \\ e_{N-1}[n] \end{bmatrix}}_{\mathbf{e}_n^{FB}} = \underbrace{\begin{bmatrix} \mathbf{s}_{0_{0:N}} & \cdots & \mathbf{s}_{M-1_{0:N}} \\ \vdots & \ddots & \vdots \\ \mathbf{s}_{0_{N-1:N}} & \cdots & \mathbf{s}_{M-1_{N-1:N}} \end{bmatrix}}_{\mathbf{S}^T} \underbrace{\begin{bmatrix} \boldsymbol{\epsilon}_0^{(n)} \\ \vdots \\ \boldsymbol{\epsilon}_{M-1}^{(n)} \end{bmatrix}}_{\mathbf{e}_n^{SB}} \tag{7.4}$$

in which

$$\underbrace{\begin{bmatrix} \boldsymbol{\epsilon}_0^{(n)} \\ \vdots \\ \boldsymbol{\epsilon}_{M-1}^{(n)} \end{bmatrix}}_{\mathbf{e}_n^{SB}} = \underbrace{\begin{bmatrix} \mathbf{d}_0^{(n)} \\ \vdots \\ \mathbf{d}_{M-1}^{(n)} \end{bmatrix}}_{\mathbf{d}_n} - \underbrace{\begin{bmatrix} \mathbf{y}_0^{(n)} \\ \vdots \\ \mathbf{y}_{M-1}^{(n)} \end{bmatrix}}_{\mathbf{y}_n}, \tag{7.5}$$

$$\mathbf{d}_m^{(n)} = \begin{bmatrix} d_m[n-L_p+1] \\ \vdots \\ d_m[n] \end{bmatrix} \Bigg\updownarrow L_p, \quad \boldsymbol{\epsilon}_m^{(n)} = \begin{bmatrix} \epsilon_m[n-L_p+1] \\ \vdots \\ \epsilon_m[n] \end{bmatrix} \Bigg\updownarrow L_p, \tag{7.6}$$

and

$$\mathbf{s}_{m_{l:N}} = \underbrace{\begin{bmatrix} g_{m_{l:N}}[L_p-1] & \cdots & g_{m_{l:N}}[0] \end{bmatrix}}_{L_p}. \tag{7.7}$$

[1] It is assumed that all filters are FIR.

Hence, $\mathbf{e}_n^{FB}$ is the fullband error vector, $\mathbf{e}_n^{SB}$ contains the subband errors, $\mathbf{S}$ represents the synthesis filter bank and $g_{m_{l:N}}[n] = g_m[nN + l]$ is the $l$–th out of $N$ polyphase components of the $m$–th synthesis filter $g_m[k]$.

Optimal error suppression is obtained when Eq. 7.4 is as small as possible :

$$\min_{\mathbf{w}_n} \mathcal{E}\left\{||\mathbf{e}_n^{FB}||_2^2\right\}. \tag{7.8}$$

Instead of the expected value of the fullband error energy its instantaneous value will be minimized. The optimal $\mathbf{w}_n$ can then be found from Eqs. 7.1, 7.4 and 7.5 :

$$\begin{aligned} \mathbf{w}_n^{\text{opt}} &= \underset{\mathbf{w}_n}{\text{argmin}}\ J_n^{FB} && (7.9)\\ &= \underset{\mathbf{w}_n}{\text{argmin}} ||\mathbf{S}^T\mathbf{d}_n - \mathbf{S}^T\mathbf{X}_n\mathbf{w}_n||_2^2. && (7.10) \end{aligned}$$

In practice, the optimal subband filters $\mathbf{w}_n^{\text{opt}}$ are estimated adaptively using a steepest descent algorithm. For this the gradient of $J_n^{FB}$ with respect to $\mathbf{w}_n$ has to be computed. Following the instruction of Eqs. C.36–C.38

$$\boldsymbol{\nabla} J_n^{FB} = 2\mathbf{X}_n^H\mathbf{S}^*\mathbf{S}^T(\mathbf{X}_n\mathbf{w}_n - \mathbf{d}_n) \tag{7.11}$$

is found. Therefore, $\mathbf{w}_n^{\text{opt}}$ can be computed iteratively using a fullband error block adaptation algorithm :

$$\begin{aligned} \mathbf{w}_{n+1} &= \mathbf{w}_n + 2\boldsymbol{\Delta}\mathbf{X}_n^H\mathbf{S}^*\mathbf{S}^T(\mathbf{d}_n - \mathbf{X}_n\mathbf{w}_n) && (7.12)\\ &= \mathbf{w}_n + 2\boldsymbol{\Delta}\mathbf{X}_n^H\mathbf{S}^*\mathbf{S}^T\mathbf{e}_n^{SB} && (7.13) \end{aligned}$$

with $\boldsymbol{\Delta} = \text{diag}(\mu_m)$ a diagonal matrix with subband dependent stepsizes. Remark that the subband errors pass through the synthesis bank twice ($\mathbf{S}^*\mathbf{S}^T$) before they are combined with $\mathbf{X}_n^H$.

In a true subband adaptation scheme the subband errors are minimized, leading to

$$\min_{\mathbf{w}_n} \mathcal{E}\left\{||\mathbf{e}_n^{SB}||_2^2\right\}. \tag{7.14}$$

Also in this case a steepest descent algorithm can be obtained. Then the gradient of $J_n^{SB} = ||\mathbf{d}_n - \mathbf{X}_n\mathbf{w}_n||_2^2$ with respect to $\mathbf{w}_n$ is computed, leading to

$$\boldsymbol{\nabla} J_n^{SB} = 2\mathbf{X}_n^H(\mathbf{X}_n\mathbf{w}_n - \mathbf{d}_n). \tag{7.15}$$

Therefore, the optimal $\mathbf{w}_n$ can be found iteratively now using subband errors :

$$\mathbf{w}_{n+1} = \mathbf{w}_n + 2\boldsymbol{\Delta}\mathbf{X}_n^H\mathbf{e}_n^{SB}. \tag{7.16}$$

In standard subband adaptive systems as presented in chapter 5 the following weight update equation

$$\mathbf{w}_{n+1} = \mathbf{w}_n + 2\boldsymbol{\Delta}\mathbf{X}_n^H \begin{bmatrix} \boldsymbol{\mathcal{Q}} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \boldsymbol{\mathcal{Q}} \end{bmatrix} \mathbf{e}_n^{SB} \tag{7.17}$$

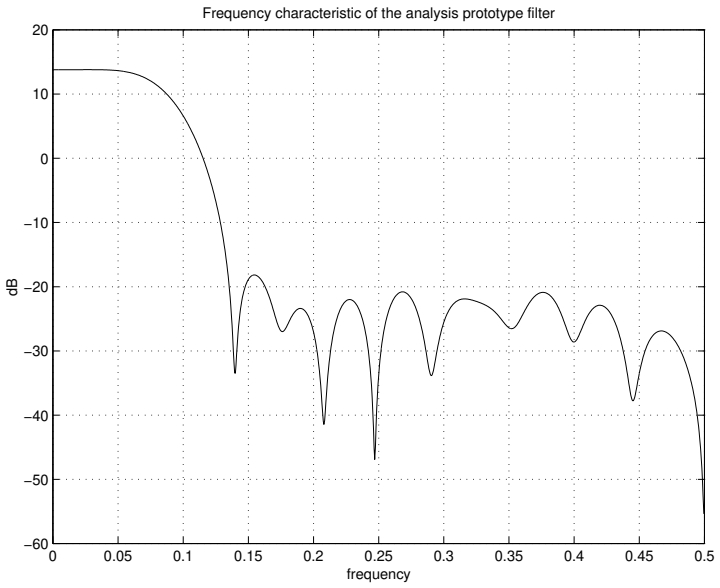


Figure 7.2: Simulation example : frequency characteristic of the analysis prototype filter

is applied to adapt $\mathbf{w}_n$ with $\mathcal{Q}$ an $L_p \times L_p$ diagonal matrix with only 1 element different from 0 :

$$\mathcal{Q} = \begin{bmatrix} \mathbf{0}_{(L_p-1)\times(L_p-1)} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{7.18}$$

Equation 7.17 then corresponds to a standard subband weight updating using the LMS algorithm in each subband. Equation 7.16 on the other hand uses the block–LMS algorithm to adapt the subband weights.

Remark that the compensation matrix $\mathbf{S}^*\mathbf{S}^T$, which is used by the fullband error adaptation algorithm, is absent in Eqs. 7.16 and 7.17. It is therefore hoped that thanks to $\mathbf{S}^*\mathbf{S}^T$ the fullband error adaptation scheme will outperform the standard subband weight updating of Eqs. 7.16 and 7.17.

**Example 7.1** A random FIR system of order 15 was generated and white noise was applied to the input. A 6–band, 4–fold downsampled analysis/synthesis filter bank set was designed ($M = 6$ and $N = 4$) fulfilling the perfect reconstruction property (Eq. 3.47). Both the analysis and the synthesis filters are FIR, DFT modulated and are of order 27. The frequency characteristic of the analysis prototype filter is shown in figure 7.2.
Based on this filter bank set different subband adaptive filters are compared. They

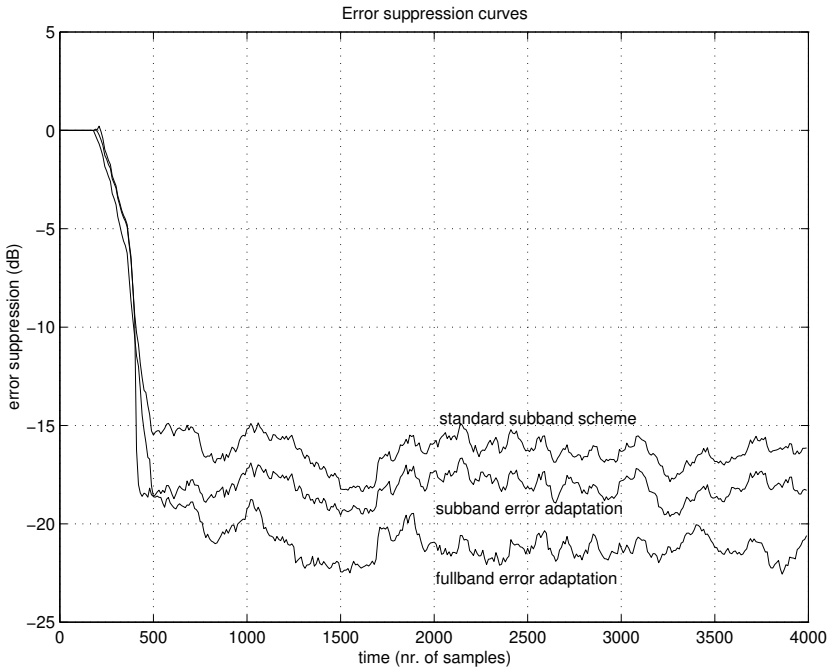


Figure 7.3: Simulation example : convergence curves of 3 different adaptive systems. The FIR system to be identified is of order 15.

all have subband filters of length $\frac{L_{FB}}{N} = 4$, so no extra taps were added. The time evolution of the error output is shown in figure 7.3 for three different subband adaptive systems : *i*) a standard subband adaptive filter with LMS–adapted subband filters as presented in figure 5.1, whose weight updating corresponds to Eq. 7.17 (upper curve), *ii*) a subband adaptive filter using block–LMS weight updating according to Eq. 7.16 (middle curve) and *iii*) a subband implementation using the fullband error adaptation scheme presented in Eq. 7.13 (bottom curve). The fullband error adaptation scheme clearly outperforms the subband error adaptation algorithm and the standard subband adaptive filter. △

## 7.2 Computational complexity

To update the subband filter weights with the fullband error adaptation algorithm the subband errors pass through the synthesis bank twice, as can be verified from Eq. 7.13. As a consequence at first sight, the corresponding implementation cost is rather high.

Now, from Eq. 7.4 it is observed that Eq. 7.13 can be written as

$$\mathbf{w}_{n+1} = \mathbf{w}_n + 2\mathbf{\Delta}\mathbf{X}_n^H \mathbf{S}^* \mathbf{e}_n^{FB}. \tag{7.19}$$

Notice that vector $\mathbf{e}_n^{FB}$ has to be computed anyway whether fullband error updating is applied or not, for $\mathbf{e}_n^{FB}$ is the output of the algorithm independent of the updating mode. Hence, as $\mathbf{S}$ is an $ML_p \times N$ complex matrix the additional cost compared to the subband error based adaptation of Eq. 7.16 is then given by

$$\text{to compute } \bar{\mathbf{e}}_n^{FB} = \mathbf{S}^* \mathbf{e}_n^{FB} \quad \longrightarrow \quad NML_p \text{ complex MACs} \approx 8NML_p \text{ op}.$$

The cost reduces however when the synthesis filter bank is (modified) DFT modulated and $e$ is a real signal. Only half of the subbands need to be computed in that case and furthermore $\mathbf{S}^*\mathbf{S}^T$ can be simplified.

Recall from Eq. 3.8 that the transpose of the polyphase matrix of a (modified) DFT modulated synthesis filter bank can be written as

$$\mathbf{G}^T(z) = \mathbf{J}\mathbf{C}(z)\mathbf{F}^{-1}. \tag{7.20}$$

$\mathbf{S}^T$ as defined by Eq. 7.4 and 7.7 is then closely related to $\mathbf{G}^T(z)$ and can be expressed as

$$\mathbf{S}^T = \mathbf{J}\mathbf{Q}\hat{\mathbf{F}}^{-1} \tag{7.21}$$

where $\hat{\mathbf{F}}$ is an extended DFT matrix, which is defined as the Kronecker product of the DFT matrix $\mathbf{F}$ and the identity matrix :

$$\hat{\mathbf{F}} = \mathbf{F} \otimes \mathbf{I}_{L_p}, \tag{7.22}$$

i.e.

$$\hat{\mathbf{F}} = \begin{bmatrix} e^0\mathbf{I}_{L_p} & \dots & \dots & e^0\mathbf{I}_{L_p} \\ \vdots & & & \\ & e^{-j\frac{2\pi pq}{M}}\mathbf{I}_{L_p} & & \\ \vdots & & & \\ e^0\mathbf{I}_{L_p} & \dots & \dots & e^{-j\frac{2\pi(M-1)(M-1)}{M}}\mathbf{I}_{L_p} \end{bmatrix}. \tag{7.23}$$

It follows that $\hat{\mathbf{F}}^{-1} = \mathbf{F}^{-1} \otimes \mathbf{I}_{L_p}$. Further, $\mathbf{Q}$ is a block $N \times M$ matrix and is constructed from $\mathbf{C}(z)$ in the same way as $\mathbf{S}^T$ was derived from $\mathbf{G}^T(z)$. Hence, each block element of $\mathbf{Q}$ is a real–valued $1 \times L_p$ vector such that

$$\mathbf{Q} = \begin{bmatrix} \boxed{c_{0,0}[L_p-1] \ \dots \ c_{0,0}[0]} & \dots & \boxed{c_{0,M-1}[L_p-1] \ \dots \ c_{0,M-1}[0]} \\ \vdots & \ddots & \vdots \\ \boxed{c_{N-1,0}[L_p-1] \ \dots \ c_{N-1,0}[0]} & \dots & \boxed{c_{N-1,M-1}[L_p-1] \ \dots \ c_{N-1,M-1}[0]} \end{bmatrix}. \tag{7.24}$$

$c_{n,m}$ refers to element $(n,m)$ of $\mathbf{C}(z)$, which is equal to

$$[\mathbf{C}(z)]_{n,m} = c_{n,m}[0] + z^{-1}c_{n,m}[1] + \cdots + z^{-(L_p-1)}c_{n,m}[L_p-1]. \tag{7.25}$$

From Eqs. 7.13, 7.19 and 7.21 it is found that

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \frac{2}{M}\mathbf{\Delta}\mathbf{X}_n^H\hat{\mathbf{F}}\mathbf{Q}^T\mathbf{J}\mathbf{e}_n^{FB} \tag{7.26}$$

with

$$\mathbf{e}_n^{FB} = \mathbf{J}\mathbf{Q}\hat{\mathbf{F}}^{-1}\mathbf{e}_n^{SB}. \tag{7.27}$$

Recall that for DFT modulated filter banks $\mathbf{C}(z)$ typically has the same structure as $\mathbf{B}(z)$ and hence it is a sparse matrix as can be verified in Eq. 3.3. As a result only about $NL_p$ of the $NML_p$ elements of $\mathbf{Q}$ are non–zero.

The matrix–vector product $\hat{\mathbf{F}}^{-1}\mathbf{e}_n^{SB}$ in Eq. 7.27 comes down to a set of IFFT–operations. For each of the $L_p$ time–lags in $\mathbf{e}_n^{SB}$ an $M$–point IFFT has to be computed. Now, $L_p-1$ IFFTs can be recuperated from previous blocks. Remark however that all elements of $\mathbf{y}_n$ and hence of $\mathbf{e}_n^{SB}$ depend on $\mathbf{w}_n$. Through recuperation of the IFFTs from previous blocks the algorithm is therefore slightly changed. Although this is a simplification w.r.t. the fullband error algorithm, it has been observed that the performance stays more or less the same. On the other hand it appears that the maximum stepsize is somewhat lower than for the true fullband error approach.

The recuperation of IFFTs from previous blocks highlights another difference between the fullband error approach of Eq. 7.13 and the subband error based approach of Eq. 7.16 on the one hand and the standard subband adaptive filter whose weight update was given in Eq. 7.17 on the other hand. Whereas in the standard subband adaptive filtering scheme only one new set of $M$ subband error signals is fed into the synthesis filter bank and the other $M(L_p-1)$ samples are being delayed and collected from previous block updates, in the block adaptation schemes of Eqs. 7.13 and 7.16 the synthesis bank is fed with a complete new set of $ML_p$ samples each time.

The cost associated with Eq. 7.27 is

$$\begin{aligned}
&\text{for } \bar{\mathbf{e}}_n^{SB} = \hat{\mathbf{F}}^{-1}\mathbf{e}_n^{SB} \quad \longrightarrow \quad 2ML_p\log_2 M \text{ op.} \\
&\qquad\qquad\qquad\qquad\qquad\quad 2M\log_2 M \text{ op. (if IFFTs are recuperated)} \\
&\text{for } \mathbf{e}_n^{FB} = \mathbf{J}\mathbf{Q}\bar{\mathbf{e}}_n^{SB} \quad \longrightarrow \quad NL_p \text{ real MACs} \approx 2NL_p \text{ op.}
\end{aligned}$$

The additional cost for the fullband error update compared to the subband error based scheme of Eq. 7.16 follows from Eq. 7.26 and is

$$\begin{aligned} \text{for } \tilde{\mathbf{e}}_n^{FB} = \mathbf{Q}^T\mathbf{J}\mathbf{e}_n^{FB} \quad &\longrightarrow \quad NL_p \text{ real MACs} \approx 2NL_p \text{ op.} \\ \text{for } \underline{\tilde{\mathbf{e}}}_n^{FB} = \hat{\mathbf{F}}\tilde{\mathbf{e}}_n^{FB} \quad &\longrightarrow \quad L_p(2M\log_2 M - 4M + 6) \text{ op.} \end{aligned}$$

Compared to the standard subband adaptive filter of chapter 5 it is furthermore observed that additional computations are required for the fullband and the subband error update scheme, which amount to[2]

$$\begin{aligned} \text{equation 7.1} \quad &\longrightarrow \quad (\tfrac{M}{2}+1)L_pL_{SB} \text{ cplx MACs} \approx (4M+8)L_pL_{SB} \text{ op.} \\ \text{equation 7.5} \quad &\longrightarrow \quad (\tfrac{M}{2}+1)L_p \text{ cplx adds} \approx (M+2)L_p \text{ op.} \\ \text{equation 7.16} \quad &\longrightarrow \quad (\tfrac{M}{2}+1)L_pL_{SB} \text{ cplx MACs} + (\tfrac{M}{2}+1)L_{SB} \text{ cplx adds} \\ &\qquad \approx (4M+8)L_pL_{SB} + (M+2)L_{SB} \text{ op.} \end{aligned}$$

The cost corresponding to equation 7.1 and 7.5 is reduced when buffers are recuperated from previous blocks (cf. Eq. 7.27). In that case Eqs. 7.1 and 7.5 cost as much as the subband filtering operation and the subband error update in a the standard subband adaptive filter, i.e. as in the case for which $L_p = 1$.

The cost corresponding to Eq. 7.16 can be quite high if $L_p$ and $L_{SB}$ are large. By comparing Eq. 7.16 with 7.17 it is found that the cost for the block weight update of Eq. 7.16 is $L_p$ times higher than for the sample–based update of Eq. 7.17.

## 7.3 PBFDAF weight updating revisited

It was shown in section 6.2 that the PBFDAF can be considered as a special subband adaptive filtering structure relying on filter banks with a poor frequency selectivity. Thanks to the overlap–save error correction mechanism the performance of the PBFDAF is remarkably good. The alternative adaptation scheme, which was studied in paragraph 7.1 and which leaded to the fullband error weight update equation 7.13, is now applied to a subband system whose filter banks are those of the PBFDAF. It will be shown that in this case the fullband error weight update equation corresponds to the weight updating of the unconstrained PBFDAF.

It is observed from Eq. 3.8 that the polyphase matrix of a (modified) DFT modulated synthesis filter bank can be written as

$$\mathbf{G}(z) = \mathbf{F}^{-T}\mathbf{C}^T(z)\mathbf{J}. \tag{7.28}$$

[2] For Eq. 7.16 we make abstraction of $\mathbf{\Delta}$ and replace $\mathbf{e}_n^{SB}$ by $\underline{\tilde{\mathbf{e}}}_n^{FB}$ if fullband error updating is required.

For the PBFDAF for which $P$ is a multiple of $L$, it was proven (see Eq. 6.55) that

$$\mathbf{C}(z) = \underbrace{\left[\begin{array}{cccc} \mathbf{I}_L & \mathbf{0}_L & \ldots & \mathbf{0}_L \quad \mathbf{0} \end{array}\right]}_{M}, \tag{7.29}$$

which is independent of $z$. Hence,

$$\mathbf{C}(z) = \mathbf{C} = \mathbf{Q} \tag{7.30}$$

and also $\mathbf{G}(z)$ is independent of $z$. This means the synthesis polyphase filters have length $L_p \leqslant 1$.

As the PBFDAF was presented as an $L$–fold downsampled subband system, whereas the fullband error adaptation algorithm as derived in section 7.1 assumes $N$–fold downsampling, it follows that $\mathbf{S}$ as defined in Eq. 7.4 is given by

$$\mathbf{S} \;=\; \mathbf{G}(z) = \mathbf{F}^{-T}\mathbf{C}^T\mathbf{J} \tag{7.31}$$

in which $N$ was replaced by $L$.

For a subband system using the filter banks of the PBFDAF the update equation 7.13 becomes

$$\mathbf{w}_{n+1} \;=\; \mathbf{w}_n + 2\frac{\boldsymbol{\Delta}}{M}\mathbf{X}_n^H\mathbf{F}\mathbf{C}^T\mathbf{C}\mathbf{F}^{-1}(\mathbf{d}_n - \mathbf{X}_n\mathbf{w}_n) \tag{7.32}$$

$$\;=\; \mathbf{w}_n + 2\frac{\boldsymbol{\Delta}}{M}\mathbf{X}_n^H\mathbf{F}\left[\begin{array}{cc} \mathbf{I}_L & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{M-L} \end{array}\right]\mathbf{F}^{-1}\mathbf{e}_n^{SB}, \tag{7.33}$$

where $\mathbf{e}_n^{SB}$ are the subband errors and $\mathbf{F}\mathbf{C}^T\mathbf{C}\mathbf{F}^{-1}$ does the error correction.

**Theorem 7.1** *The weight update equation of the unconstrained PBFDAF*

$$\underline{\mathbf{w}}_p^{(n+1)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n)} + \boldsymbol{\Delta}\,\underline{\mathbf{X}}_p^{(n)*}\,\mathbf{F}\mathbf{e}^{(n)} \tag{7.34}$$

*for which $P$ is a multiple of $L$, corresponds to equation 7.33, which was obtained by applying the fullband error adaptation update equation 7.13 to a subband systems with the filter banks of the PBFDAF.*

**Proof:** The proof can be found in appendix C.10.

It is observed from Eq. 7.19 that the subband filters $\mathbf{w}_n$ are adapted based on corrected subband errors. The correction implies that the subband errors pass through the synthesis filter bank in order to obtain a fullband, i.e. an aliasing–free error signal. Theorem 7.1 proves that this fullband error adaptation algorithm can be viewed as an extension to the overlap–save or overlap–add procedure upon which frequency–domain techniques are based. A class of subband adaptive systems can therefore be designed that fulfill the two design criteria on frequency selectivity and perfect reconstruction, formulated in section 5.2, and that rely on the adaptation mechanism presented in this chapter.

## 7.4 Conclusions

In this chapter a novel fullband error adaptation scheme for subband adaptive filters was proposed. This alternative adaptation scheme, which was presented in section 7.1, adjusts the subband filters by feeding back the fullband error instead of the subband errors, as is done in a classical subband adaptive system.

A cost analysis was performed in section 7.2. It was shown from a computational complexity point of view how the standard subband adaptive filter relates to the different adaptation schemes presented in this chapter. Although the fullband error adaptation scheme outperforms the standard subband adaptive filter in terms of convergence behavior, the computational cost is higher as the subband errors pass through the synthesis filter bank twice.

In section 7.3 we showed that the weight updating of the unconstrained PBFDAF algorithm for which $P$ is a multiple of $L$, corresponds to the fullband error adaptation algorithm using the filter banks of the PBFDAF. This demonstrates that the fullband error adaptation scheme presented in this chapter can be considered as an extension of the PBFDAF overlap–save error correction mechanism to a more general class of subband adaptive filters.

# Part III

# Iterated Partitioned Block Frequency–Domain Adaptive Filtering

In part III the Partitioned–Block Frequency–Domain RAP (PBFDRAP) adaptive filter will be discussed, which combines frequency–domain adaptive filtering with so–called "row action projection". The PBFDRAP is obtained by iterating the filter and weight updating part of the PBFDAF algorithm, leading to an improved error output and better estimated filter weights.

In **chapter 8** the PBFDRAP is presented. As the PBFDRAP is an iterated version of the PBFDAF it will be checked what happens if the number of iteration steps goes to infinity. It is found that for some parameter settings the PBFDRAP approaches well–known adaptive filtering algorithms such as the PRA, which was defined in section 2.3.2. It is furthermore shown that the PBFDRAP algorithm outperforms the PBFDAF in a realistic echo cancellation setup.

In **chapter 9** fast implementations are derived for the PBFDRAP algorithm. The different fast implementation schemes are compared with the standard implementation of the PBFDRAP for different parameter settings. It appears that a significant complexity reduction can be obtained. It is also shown that the fast PBFDRAP is a viable alternative to the PRA, if large block lengths are preferred.

# Chapter 8

# Partitioned Block Frequency–Domain RAP

For high quality acoustic echo cancellation long echoes have to be suppressed. Classical LMS–based adaptive filters are not attractive as they are suboptimal from a computational complexity point of view (see section 2.3.1). Multirate adaptive filtering schemes such as the partitioned block frequency–domain adaptive filter (PBFDAF) are a good alternative and are widely used in commercial echo cancellers nowadays. The PBFDAF was discussed in detail in chapter 6. In this chapter the PBFDRAP is analyzed, which combines partitioned block frequency–domain adaptive filtering with so–called "row action projection" (RAP). The RAP algorithm was initially proposed as an improvement to the LMS adaptive filter [69].

The PBFDRAP is defined in section 8.1 starting from the equations describing the PBFDAF. It will be shown that the PBFDRAP basically is an iterated variant of the PBFDAF algorithm. The mechanism of the algorithm is explained and it is pointed out how extra error suppression can be achieved w.r.t. the PBFDAF.

In section 8.2 it is checked how the algorithm behaves if the number of iteration steps goes to infinity. Different algorithmic settings are considered. For some parameter settings it is found that the PBFDRAP algorithm approaches well–known adaptive filtering algorithms such as the PRA, which was defined in section 2.3.2.

In section 8.3 it is shown that the PBFDRAP outperforms the PBFDAF in a realistic echo cancellation setup.

Conclusions are formulated in section 8.4.

## 8.1 Partitioned block frequency–domain RAP

### 8.1.1 Definition

Stepping several times through the filter and weight updating part of the PBFDAF algorithm (Eqs. 6.29, 6.31 and 6.32) while the block index $n$ is kept constant, leads to an improved estimate of the error output and filter weights. The $L_{FB}$–taps fullband adaptive filter $\hat{w}^{(n)}[k]$ is partitioned in $\frac{L_{FB}}{P}$ parts[1] $\mathbf{w}_p^{(n)}$ of length $P$ each, which are transformed to the frequency domain :

$$\mathbf{w}_p^{(n)} \overset{\forall p}{=} \begin{bmatrix} \hat{w}^{(n)}[pP] \\ \vdots \\ \hat{w}^{(n)}[(p+1)P-1] \end{bmatrix}, \quad p = 0 : \frac{L_{FB}}{P} - 1 \tag{8.1}$$

$$\underline{\mathbf{w}}_p^{(n,1)} \overset{\forall p}{=} \mathbf{F} \begin{bmatrix} \mathbf{w}_p^{(n)} \\ \mathbf{0} \end{bmatrix} \begin{matrix} \updownarrow P \\ \updownarrow M-P \end{matrix} \tag{8.2}$$

The equations defining the overlap–save PBFDRAP are

$$\underline{\mathbf{X}}_p^{(n)} \overset{\forall p}{=} \mathrm{diag}\left\{ \mathbf{F} \begin{bmatrix} x[(n+1)L-pP-M+1] \\ \vdots \\ x[(n+1)L-pP] \end{bmatrix} \right\} \Bigg\uparrow M, \quad p = 0 : \frac{L_{FB}}{P} - 1 \tag{8.3}$$

$$\mathbf{d}^{(n)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_n \end{bmatrix} \begin{matrix} \updownarrow P-1+\sigma \\ \updownarrow L \end{matrix}, \qquad \mathbf{d}_n = \begin{bmatrix} d[nL+1] \\ \vdots \\ d[(n+1)L] \end{bmatrix} \Bigg\uparrow L \tag{8.4}$$

`for` $r = 1$ `to` $R$ `do` $\{$

$$\mathbf{y}^{(n,r)} = \begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \underline{\mathbf{w}}_p^{(n,r)} \tag{8.5}$$

$$\mathbf{e}^{(n,r)} = \mathbf{d}^{(n)} - \mathbf{y}^{(n,r)} \tag{8.6}$$

$$\underline{\mathbf{w}}_p^{(n,r+1)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n,r)} + \mathbf{G}\boldsymbol{\Delta}\underline{\mathbf{X}}_p^{(n)*}\mathbf{F}\mathbf{e}^{(n,r)}, \quad p = 0 : \frac{L_{FB}}{P} - 1 \tag{8.7}$$

$\}$

$$\underline{\mathbf{w}}_p^{(n+1,1)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n,R+1)}, \quad p = 0 : \frac{L_{FB}}{P} - 1. \tag{8.8}$$

in which $\underline{\mathbf{w}}_p^{(n,r)}$, $\mathbf{y}^{(n,r)}$ and $\mathbf{e}^{(n,r)}$ are respectively the adaptive filter weights, the reconstructed system output and the error output at iteration step $r$.

This extension to the PBFDAF algorithm, which iterates several times on the same block of data $(\underline{\mathbf{X}}_p^{(n)}, \mathbf{d}_n)$, will be called the partitioned block frequency–domain

[1] It is assumed that $\frac{L_{FB}}{P}$ is integer.

RAP (PBFDRAP) adaptive filter. RAP stands for Row Action Projection and was initially proposed as an improvement to the LMS algorithm. A good reference on Row Action Projection is [69].

Also in this case there exist two variants of the algorithm, called the constrained and the unconstrained PBFDRAP. For the unconstrained PBFDAF, $\mathbf{G} = \mathbf{I}_M$, while for the constrained PBFDAF

$$\mathbf{G} = \mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{L-1+\sigma} \end{bmatrix} \mathbf{F}^{-1}. \tag{8.9}$$

The PBFDRAP is strongly related to the PBFDAF. More information on the algorithmic parameters can therefore be found in section 6.1.2.

In section 6.1.5 it was explained that an ambiguity can occur with the unconstrained PBFDAF algorithm if $\sigma > 0$. As the PBFDRAP algorithm is an extension of the PBFDAF the same ambiguity can occur in this case. The ambiguity can easily be compensated for by slightly changing Eqs. 8.4 and 8.5, i.e. computing $\mathbf{e}^{(n)}$ based on $L + \sigma$ instead of $L$ signal samples :

$$\mathbf{d}^{(n)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_n \end{bmatrix} \begin{matrix} \updownarrow P-1 \\ \updownarrow L+\sigma \end{matrix}, \qquad \mathbf{d}_n = \begin{bmatrix} d[nL-\sigma+1] \\ \vdots \\ d[(n+1)L] \end{bmatrix} \begin{matrix} \uparrow \\ \\ \downarrow \end{matrix} L+\sigma \tag{8.10}$$

$$\mathbf{y}^{(n,r)} = \begin{bmatrix} \mathbf{0}_{P-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{L+\sigma} \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \underline{\mathbf{w}}_p^{(n,r)}. \tag{8.11}$$

The other equations remain unchanged and the additional algorithmic cost is almost negligible.

The formulas defining the PBFDRAP adaptive filter and the complexity of the different algorithmic steps are summarized in table 8.1. Remark that the PBFDRAP reduces to the PBFDAF if $R = 1$.

### 8.1.2 Mechanism

The PBFDRAP algorithm iterates $R$ times on the same data $(\underline{\mathbf{X}}_p^{(n)}, \mathbf{d}_n)$. Hence, an improved weight update $\underline{\mathbf{w}}_p^{(n,R+1)}$ and a smaller a–posteriori error output $\mathbf{e}^{(n,R)}$ are obtained. If the number of iteration steps $R$ is increased the algorithm further reduces the a–posteriori error, partly by exploiting the signal characteristics, i.e. adapting the filter coefficients towards a solution that minimizes $\mathbf{e}^{(n,R)}$ for data block $(\underline{\mathbf{X}}_p^{(n)}, \mathbf{d}_n)$, rather than trying to improve the model of the unknown system $w$ as such. Therefore, a small a–posteriori error $\mathbf{e}^{(n,R)}$ does not necessarily imply a good system modelling. Hence, the quality of the model approximation should be evaluated by plotting the norm between the unknown system $w$ and the model

$\hat{w}$ (see figure 2.8) or, if $w$ is not known, by looking at the time evolution of the so–called a–priori error $\mathbf{e}^{(n,1)}$.

At first sight, iterating several times on the same block of data seems rather expensive (see table 8.1). However, in chapter 9 it is shown how the complexity can be reduced, leading to the fast PBFDRAP algorithm. Furthermore, nowadays echo cancellation is just one of the many signal processing building blocks that are combined to build a signal enhancement system, which includes apart from the echo cancellation also e.g. noise suppression and dereverberation. The signal enhancement unit may be used as a preprocessor to a more complex building block such as a speech recognizer for instance. Depending on the mode in which each of these units operate, the overall complexity of the system can vary from time to time such that at some time instances more processing power may be available for the preprocessing and the echo cancellation in particular. As the PBFDRAP is typically used to cancel long echo paths the number of iteration steps $R$ can therefore be adapted based on the available processing power, leading to better echo suppression if $R$ can be chosen large.

## 8.2 On iterating the PBFDRAP

The following equations[2]

$$y^{(k,r)} = \mathbf{x}_k^T \mathbf{w}^{(k,r)} \tag{8.12}$$

$$e^{(k,r)} = d[k] - y^{(k,r)} \tag{8.13}$$

$$\mathbf{w}^{(k,r+1)} = \mathbf{w}^{(k,r)} + \mu\, \mathbf{x}_k^* e^{(k,r)} \tag{8.14}$$

define the iterated LMS algorithm (compare with Eqs. 2.22–2.24), with $r = 1, 2, ...$ being the iteration index and $\mathbf{x}_k$ and $\mathbf{w}^{(k,1)}$ column vectors of length $L_{FB}$ representing the input data buffer and the initial adaptive filter at time $k$ respectively. It was shown [116] that reiterating the LMS algorithm leads to the normalized LMS adaptive filter. In other words, based on Eqs. 8.12–8.14, it can be proven [40] that

$$\lim_{R\to\infty} \mathbf{w}^{(k,R)} = \mathbf{w}^{(k,1)} + \mathbf{x}_k^* \frac{1}{\mathbf{x}_k^H \mathbf{x}_k} (d[k] - \mathbf{x}_k^T \mathbf{w}^{(k,1)}), \tag{8.15}$$

which is the weight update equation of the normalized LMS algorithm with stepsize $\mu = 1$ (cf. Eq. 2.28), guaranteeing maximum convergence speed.

The same sort of derivation can be done for the PBFDRAP algorithm to investigate what happens if the number of iteration steps $R$ goes to infinity. For the analysis different algorithmic settings will be considered, making a distinction between unconstrained and constrained updating on the one hand and subband normalization and unnormalized implementation of the PBFDRAP on the other hand. In this section the results of these computations are presented and discussed.

[2] For signal conventions : see figure 2.8

### 8.2.1 Computation of $\lim_{R\to\infty} \underline{\mathbf{w}}_p^{(n,R)}$

In order to compute $\lim_{R\to\infty} \underline{\mathbf{w}}_p^{(n,R)}$, we define

$$\mathbf{P}_n = \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{G} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)^*} \mathbf{F} = \begin{bmatrix} \boldsymbol{\mathcal{A}} & \boldsymbol{\mathcal{B}} \\ \boldsymbol{\mathcal{C}} & \boldsymbol{\mathcal{D}} \end{bmatrix} \tag{8.16}$$

and

$$\mathbf{N}_n = \begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{G} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)^*} \mathbf{F} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \boldsymbol{\mathcal{C}} & \boldsymbol{\mathcal{D}} \end{bmatrix} \tag{8.17}$$

where $\boldsymbol{\mathcal{D}}$ is an $L \times L$ matrix. It is further assumed that the eigenvalue decomposition of $\mathbf{N}_n$ is given by

$$\underbrace{\begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \boldsymbol{\mathcal{C}} & \boldsymbol{\mathcal{D}} \end{bmatrix}}_{\mathbf{N}_n} \underbrace{\begin{bmatrix} \mathbf{V}_\alpha & \mathbf{V}_\beta \\ \mathbf{V}_\gamma & \mathbf{V}_\delta \end{bmatrix}}_{\mathbf{V}} = \underbrace{\begin{bmatrix} \mathbf{V}_\alpha & \mathbf{V}_\beta \\ \mathbf{V}_\gamma & \mathbf{V}_\delta \end{bmatrix}}_{\mathbf{V}} \underbrace{\begin{bmatrix} \boldsymbol{\Gamma}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{P-1+\sigma} \end{bmatrix}}_{\boldsymbol{\Gamma}}, \tag{8.18}$$

in which $\boldsymbol{\Gamma}_1$ is a full rank diagonal $L \times L$ matrix, i.e. the rank of $\mathbf{N}_n$ is $L$. In section 8.2.2 and 8.2.3 it will be pointed out under which conditions this assumption is true.

The structure of $\mathbf{N}_n$ and its rank deficiency force $\mathbf{V}_\alpha$ to be $\mathbf{0}$. It follows that $\mathbf{V}_\beta$ and $\mathbf{V}_\gamma$ are square and invertible matrices. Hence, the eigenvector matrix has the following structure

$$\mathbf{V} = \begin{bmatrix} \mathbf{0} & \mathbf{V}_\beta \\ \mathbf{V}_\gamma & \mathbf{V}_\delta \end{bmatrix}. \tag{8.19}$$

Furthermore, it is easily verified using Eq. 8.18 that the eigenvalue decomposition of $\boldsymbol{\mathcal{D}}$ is given by

$$\boldsymbol{\mathcal{D}} = \mathbf{V}_\gamma \boldsymbol{\Gamma}_1 \mathbf{V}_\gamma^{-1} \tag{8.20}$$

and that

$$\mathbf{V}^{-1} = \begin{bmatrix} -\mathbf{V}_\gamma^{-1}\mathbf{V}_\delta\mathbf{V}_\beta^{-1} & \mathbf{V}_\gamma^{-1} \\ \mathbf{V}_\beta^{-1} & \mathbf{0} \end{bmatrix}. \tag{8.21}$$

**Theorem 8.1** *At the end of iteration step $R$ the error output $\boldsymbol{e}^{(n,R)}$ of the PBFDRAP algorithm can be written as*

$$\mathbf{e}^{(n,R)} = \left( \mathbf{I}_M - \begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{G} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)^*} \mathbf{F} \right)^{R-1} \mathbf{e}^{(n,1)} \tag{8.22}$$

*and the adaptive filter weights are given by*

$$\underline{\mathbf{w}}_p^{(n,R+1)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n,1)} + \mathbf{G} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)^*} \mathbf{F} \sum_{r=1}^{R} \mathbf{e}^{(n,r)}, \quad p = 0 : \frac{L_{FB}}{P} - 1. \tag{8.23}$$

**Proof:** The proof is given in appendix D.1.

As Eqs. 8.22 and 8.23 are fulfilled for all $R \in \mathbb{N}_0$, it follows from Eqs. 8.22 and 8.18 that

$$\sum_{r=1}^{R} \mathbf{e}^{(n,r)} = \sum_{r=1}^{R} \left(\mathbf{I}_M - \mathbf{N}_n\right)^{r-1} \mathbf{e}^{(n,1)} \tag{8.24}$$

$$= \mathbf{V} \sum_{r=1}^{R} \left(\mathbf{I}_M - \mathbf{\Gamma}\right)^{r-1} \mathbf{V}^{-1} \mathbf{e}^{(n,1)} \tag{8.25}$$

$$= \mathbf{V} \sum_{r=1}^{R} \begin{bmatrix} \mathbf{I}_L - \mathbf{\Gamma}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{P-1+\sigma} \end{bmatrix}^{r-1} \mathbf{V}^{-1} \mathbf{e}^{(n,1)}. \tag{8.26}$$

As $\mathbf{\Gamma}_1$ is assumed to be a full rank diagonal matrix

$$\sum_{r=1}^{R} \mathbf{e}^{(n,r)} = \mathbf{V} \begin{bmatrix} \mathbf{\Gamma}_1^{-1} \left(\mathbf{I}_L - \left(\mathbf{I}_L - \mathbf{\Gamma}_1\right)^R\right) & \mathbf{0} \\ \mathbf{0} & R\mathbf{I}_{P-1+\sigma} \end{bmatrix} \mathbf{V}^{-1} \mathbf{e}^{(n,1)}. \tag{8.27}$$

By inserting Eqs. 8.19 and 8.21 in Eq. 8.27 and defining

$$\mathbf{\Xi}^{(R)} = \mathbf{\Gamma}_1^{-1} \left(\mathbf{I}_L - \left(\mathbf{I}_L - \mathbf{\Gamma}_1\right)^R\right) \tag{8.28}$$

it is found that

$$\sum_{r=1}^{R} \mathbf{e}^{(n,r)} = \begin{bmatrix} R\mathbf{I}_{P-1+\sigma} & \mathbf{0} \\ \left(-\mathbf{V}_\gamma \mathbf{\Xi}^{(R)} \mathbf{V}_\gamma^{-1} + R\mathbf{I}_L\right) \mathbf{V}_\delta \mathbf{V}_\beta^{-1} & \mathbf{V}_\gamma \mathbf{\Xi}^{(R)} \mathbf{V}_\gamma^{-1} \end{bmatrix} \mathbf{e}^{(n,1)}. \tag{8.29}$$

From Eqs. 8.4, 8.5 and 8.6 it is verified that $\mathbf{e}^{(n,1)}$ can be written as

$$\mathbf{e}^{(n,1)} = \begin{bmatrix} \mathbf{0}_{P-1+\sigma \times 1} \\ \mathbf{e}_n \end{bmatrix}, \tag{8.30}$$

with $\mathbf{e}_n$ an $L \times 1$–vector. Hence,

$$\sum_{r=1}^{R} \mathbf{e}^{(n,r)} = \begin{bmatrix} \boldsymbol{\mathcal{K}} & \mathbf{0} \\ \boldsymbol{\mathcal{L}} & \mathbf{V}_\gamma \mathbf{\Xi}^{(R)} \mathbf{V}_\gamma^{-1} \end{bmatrix} \mathbf{e}^{(n,1)}, \tag{8.31}$$

in which $\boldsymbol{\mathcal{K}}$ and $\boldsymbol{\mathcal{L}}$ can be any matrix having the correct dimensions.

If $\gamma_i$, the diagonal elements of $\mathbf{\Gamma}_1$, are such that $0 \leqslant |1 - \gamma_i| < 1$,

$$\lim_{R \to \infty} \mathbf{\Xi}^{(R)} = \mathbf{\Gamma}_1^{-1} \tag{8.32}$$

and hence from Eq. 8.20 it is found that

$$\lim_{R\to\infty}\sum_{r=1}^{R}\mathbf{e}^{(n,r)} = \begin{bmatrix} \boldsymbol{\mathcal{K}} & \mathbf{0} \\ \boldsymbol{\mathcal{L}} & \mathbf{V}_\gamma\boldsymbol{\Gamma}_1^{-1}\mathbf{V}_\gamma^{-1} \end{bmatrix}\mathbf{e}^{(n,1)} \tag{8.33}$$

$$= \begin{bmatrix} \boldsymbol{\mathcal{K}} & \mathbf{0} \\ \boldsymbol{\mathcal{L}} & \boldsymbol{\mathcal{D}}^{-1} \end{bmatrix}\mathbf{e}^{(n,1)}. \tag{8.34}$$

Notice that the eigenvalues $\gamma_i$ depend on the stepsize matrix $\boldsymbol{\Delta}$.

From Eqs. 8.23 and 8.34,

$$\lim_{R\to\infty}\underline{\mathbf{w}}_p^{(n,R)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n,1)} + \mathbf{G}\boldsymbol{\Delta}\underline{\mathbf{X}}_p^{(n)^*}\mathbf{F}\begin{bmatrix} \boldsymbol{\mathcal{K}} & \mathbf{0} \\ \boldsymbol{\mathcal{L}} & \boldsymbol{\mathcal{D}}^{-1} \end{bmatrix}\mathbf{e}^{(n,1)} \tag{8.35}$$

$$\overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n,1)} + \mathbf{G}\boldsymbol{\Delta}\underline{\mathbf{X}}_p^{(n)^*}\left(\mathbf{F}\begin{bmatrix} \boldsymbol{\mathcal{K}}^{-1} & \mathbf{0} \\ -\boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{K}}^{-1} & \boldsymbol{\mathcal{D}} \end{bmatrix}\mathbf{F}^{-1}\right)^{-1}\mathbf{F}\mathbf{e}^{(n,1)} \tag{8.36}$$

is found. Choose for instance $\boldsymbol{\mathcal{L}} = -\boldsymbol{\mathcal{D}}^{-1}\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{K}}$. Then,

$$\lim_{R\to\infty}\underline{\mathbf{w}}_p^{(n,R)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n,1)} + \mathbf{G}\boldsymbol{\Delta}\underline{\mathbf{X}}_p^{(n)^*}\left(\mathbf{F}\begin{bmatrix} \boldsymbol{\mathcal{K}}^{-1} & \mathbf{0} \\ \boldsymbol{\mathcal{C}} & \boldsymbol{\mathcal{D}} \end{bmatrix}\mathbf{F}^{-1}\right)^{-1}\mathbf{F}\mathbf{e}^{(n,1)} \tag{8.37}$$

$$\overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n,1)} + \mathbf{G}\boldsymbol{\Delta}\underline{\mathbf{X}}_p^{(n)^*}\left(\mathbf{F}\begin{bmatrix} \boldsymbol{\mathcal{G}} & \boldsymbol{\mathcal{H}} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix}\begin{bmatrix} \boldsymbol{\mathcal{A}} & \boldsymbol{\mathcal{B}} \\ \boldsymbol{\mathcal{C}} & \boldsymbol{\mathcal{D}} \end{bmatrix}\mathbf{F}^{-1}\right)^{-1}\mathbf{F}\mathbf{e}^{(n,1)} \tag{8.38}$$

with

$$\boldsymbol{\mathcal{G}} = \left((\boldsymbol{\mathcal{A}} - \boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{D}}^{-1}\boldsymbol{\mathcal{C}})\boldsymbol{\mathcal{K}}\right)^{-1} \tag{8.39}$$

and

$$\boldsymbol{\mathcal{H}} = -\left((\boldsymbol{\mathcal{A}} - \boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{D}}^{-1}\boldsymbol{\mathcal{C}})\boldsymbol{\mathcal{K}}\right)^{-1}\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{D}}^{-1}, \tag{8.40}$$

provided $(\boldsymbol{\mathcal{A}} - \boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{D}}^{-1}\boldsymbol{\mathcal{C}})\boldsymbol{\mathcal{K}}$ is invertible. Hence from Eq. 8.16

$$\lim_{R\to\infty}\underline{\mathbf{w}}_p^{(n,R)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n,1)} + \mathbf{G}\boldsymbol{\Delta}\underline{\mathbf{X}}_p^{(n)^*}\left(\boldsymbol{\mathcal{P}}\sum_{p=0}^{\frac{L_{FB}}{P}-1}\underline{\mathbf{X}}_p^{(n)}\mathbf{G}\boldsymbol{\Delta}\underline{\mathbf{X}}_p^{(n)^*}\right)^{-1}\mathbf{F}\mathbf{e}^{(n,1)} \tag{8.41}$$

is found, where $\boldsymbol{\mathcal{P}}$ is a projection matrix

$$\boldsymbol{\mathcal{P}} = \mathbf{F}\begin{bmatrix} \boldsymbol{\mathcal{G}} & \boldsymbol{\mathcal{H}} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix}\mathbf{F}^{-1}. \tag{8.42}$$

Both $\boldsymbol{\Delta}$ and $\underline{\mathbf{X}}_p^{(n)}$ are diagonal matrices. Therefore Eq. 8.41 reduces to

$$\lim_{R\to\infty}\underline{\mathbf{w}}_p^{(n,R)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n,1)} + \mathbf{G}\underline{\mathbf{X}}_p^{(n)^*}\left(\boldsymbol{\mathcal{P}}\sum_{p=0}^{\frac{L_{FB}}{P}-1}\underline{\mathbf{X}}_p^{(n)}\mathbf{G}\underline{\mathbf{X}}_p^{(n)^*}\right)^{-1}\mathbf{F}\mathbf{e}^{(n,1)}. \tag{8.43}$$

Notice that $\boldsymbol{\Delta}$ has disappeared from the weight update equation. However in general, $\boldsymbol{\mathcal{P}}$ is still dependent on $\boldsymbol{\Delta}$.

Observe that equation 8.43 also can be written as

$$\lim_{R\to\infty} \underline{\mathbf{w}}_p^{(n,R)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n,1)} + \mathbf{G}\underline{\mathbf{X}}_p^{(n)*} \left( \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{G} \underline{\mathbf{X}}_p^{(n)*} \right)^{-1} \boldsymbol{\mathcal{P}}^{-1} \mathbf{F} \mathbf{e}^{(n,1)}, \tag{8.44}$$

with

$$\boldsymbol{\mathcal{P}}^{-1} = \mathbf{F} \begin{bmatrix} \boldsymbol{\mathcal{G}}^{-1} & -\boldsymbol{\mathcal{G}}^{-1}\boldsymbol{\mathcal{H}} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1}. \tag{8.45}$$

By choosing $\boldsymbol{\mathcal{K}} = \mathbf{0}$ and applying Eq. 8.30,

$$\lim_{R\to\infty} \underline{\mathbf{w}}_p^{(n,R)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n,1)} + \mathbf{G}\underline{\mathbf{X}}_p^{(n)*} \left( \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{G} \underline{\mathbf{X}}_p^{(n)*} \right)^{-1} \mathbf{F} \begin{bmatrix} \mathbf{0} & \boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{D}}^{-1} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{e}^{(n,1)} \tag{8.46}$$

$$\overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n,1)} + \mathbf{G}\underline{\mathbf{X}}_p^{(n)*} \left( \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{G} \underline{\mathbf{X}}_p^{(n)*} \right)^{-1} \mathbf{F} \begin{bmatrix} \boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{D}}^{-1}\mathbf{e}_n \\ \mathbf{e}_n \end{bmatrix}. \tag{8.47}$$

is found.

Both Eq. 8.43 and 8.47 give a closed expression for $\lim_{R\to\infty} \underline{\mathbf{w}}_p^{(n,R)}$ under the assumption that

1. $\boldsymbol{\Gamma}_1$ is a full rank diagonal $L \times L$ matrix, i.e. for all diagonal elements $\gamma_i$,

$$\gamma_i \neq 0, \qquad i = 0 : L-1, \tag{8.48}$$

2. the weighted subband energy matrix

$$\sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{G} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)*} \tag{8.49}$$

   is invertible and

3. all eigenvalues $\gamma_i$ of $\boldsymbol{\Gamma}_1$ are such that (see Eq. 8.32)

$$0 \leqslant |1-\gamma_i| < 1, \qquad i = 0 : L-1. \tag{8.50}$$

From Eqs. 8.43 and 8.47 it follows that under conditions 8.48, 8.49 and 8.50 the PBFDRAP approaches a normalized variant of the PBFDAF that is based on projected subband energies with stepsize $\mu = 1$, if the number of iteration steps $R$ goes to infinity. In section 8.2.2 and 8.2.3 it is verified whether and when the above–mentioned conditions are met.

Remark that Eqs. 8.17–8.50 are also applicable to the PBFDRAP with ambiguity compensation (Eqs. 8.10–8.11) by replacing $L$ by $L + \sigma$ and $P - 1 + \sigma$ by $P - 1$.

### 8.2.2 Unconstrained PBFDRAP : $\lim_{R\to\infty} \underline{\mathbf{w}}_p^{(n,R)}$

For the unconstrained PBFDRAP, $\mathbf{G} = \mathbf{I}_M$ and hence equation 8.16 becomes

$$\mathbf{P}_n = \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{\Delta} \underline{\mathbf{X}}_p^{(n)*} \mathbf{F} = \begin{bmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \end{bmatrix}. \tag{8.51}$$

where $\boldsymbol{\mathcal{D}}$ is again an $L \times L$ matrix. $\mathbf{\Delta} = \mathrm{diag}(\boldsymbol{\mu}_s^{(n)})$ contains subband dependent stepsizes $\boldsymbol{\mu}_s^{(n)}(m)$ and is therefore real–valued. If the PBFDRAP is adapted, all $\boldsymbol{\mu}_s^{(n)}(m) > 0$.

**Theorem 8.2** *For the unconstrained PBFDRAP, $\boldsymbol{P}_n$ as defined in Eq. 8.51 is a right–circulant hermitian matrix having the Toeplitz structure. If the PBFDRAP is adapted in a real–life identification setup, i.e. based on real–life signals, $\boldsymbol{P}_n$ is a full–rank positive definite matrix, i.e.*

$$\forall \mathbf{x} \in \mathbb{C}_{\mathbf{0}}^M, \qquad \mathbf{x}^H \mathbf{P}_n \mathbf{x} > 0. \tag{8.52}$$

*If furthermore all input signals and the unknown system $w$ are real–valued $\boldsymbol{P}_n$ will be real–valued as well.*

**Proof :** The proof can be found in appendix D.2.

As $\mathbf{P}_n$ has full rank, the rank of $\mathbf{N}_n$ (Eq. 8.17) will be $L$. As a consequence, $\mathbf{\Gamma}_1$ will be a full rank diagonal matrix (first condition, Eq. 8.48). Also the second condition (Eq. 8.49) is fulfilled as it is assumed that the data $\underline{\mathbf{X}}_p^{(n)}$ is obtained from a real–life adaptive identification scenario.

**Theorem 8.3** *If the unconstrained PBFDRAP is adapted in a real–life identification setup, matrix $\boldsymbol{\mathcal{D}}$, as defined by Eq. 8.51, is hermitian and positive definite.*

**Proof :** The proof can be found in appendix D.3.

Hence, the eigenvalues of $\boldsymbol{\mathcal{D}}$, i.e. the diagonal elements of $\boldsymbol{\Gamma}_1$ (see Eq. 8.20), are real and strictly positive. If the stepsizes $\boldsymbol{\mu}_s^{(n)}(m)$ and therefore $||\boldsymbol{\Delta}||$ are sufficiently small all eigenvalues of $\boldsymbol{\Gamma}_1$ are between 0 and 2. Then also the third condition (Eq. 8.50) is fulfilled and hence $\underline{\mathbf{w}}_p^{(n,R)}$ converges to equation 8.43 if $R$ goes to infinity.

**Unnormalized and globally normalized unconstrained PBFDRAP**

For the unnormalized unconstrained PBFDRAP, $\mathbf{G} = \mathbf{I}_M$ and $\boldsymbol{\Delta}$ is defined as $\mu\mathbf{I}_M$, with $\mu$ a constant. More generally, $\boldsymbol{\Delta} = \mu_n\mathbf{I}_M$, with $\mu_n$ a block dependent stepsize. For instance, if $\mu_n$ is as defined in Eq. 6.34 the algorithm performs a global normalization, leading to better convergence if signals with a high dynamic range (such as speech) are involved.

As $\boldsymbol{\Delta} = \mu_n\mathbf{I}_M$, the elements of $\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{C}}, \boldsymbol{\mathcal{D}}$ and $\mathbf{N}_n$ linearly depend on $\mu_n$ (see Eq. 8.16 and 8.17). From Eqs. 8.39, 8.40 and 8.42 it follows that $\boldsymbol{\mathcal{P}}$ will be independent of $\mu_n$ if $\boldsymbol{\mathcal{K}}$ is chosen inversely proportional to $\mu_n$. Hence, Eq. 8.43 is equivalent to

$$\lim_{R\to\infty} \underline{\mathbf{w}}_p^{(n,R)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n,1)} + \underline{\mathbf{X}}_p^{(n)*} \left( \boldsymbol{\mathcal{P}} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \underline{\mathbf{X}}_p^{(n)*} \right)^{-1} \mathbf{F}\mathbf{e}^{(n,1)}, \tag{8.53}$$

which can be made independent of $\mu_n$ with an appropriate choice for $\boldsymbol{\mathcal{K}}$.

*From Eq. 8.53 it can be concluded that the unnormalized and globally normalized unconstrained PBFDRAP approach a normalized variant of the unconstrained PBFDAF that is based on projected subband energies*

$$\boldsymbol{\mathcal{P}} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \underline{\mathbf{X}}_p^{(n)*} \tag{8.54}$$

*and that is adapted with stepsize $\mu_n = 1$, if the number of iteration steps $R$ goes to infinity.*

**Example 8.1** In this example it is illustrated how the convergence speed of the unnormalized unconstrained PBFDRAP algorithm improves as $R$ increases. A random FIR filter $w$ of order 31 was generated and fed with a white noise signal $x$. The unknown system was estimated using the standard adaptive identification setup of figure 2.8. The unnormalized unconstrained PBFDRAP ($P = L = 16$, $M = 32$ and $L_{FB} = 32$) was used to identify $w$. The ambiguity, which typically occurs with unconstrained updating, was compensated for by applying Eq. 8.10 and 8.11. In figure 8.1 the norm between $w$ and the equivalent time–domain representation $\hat{w}$ of the adaptive filter coefficients (Eqs. 8.1 and 8.2) is plotted as a function of time. It appears that if $R$ is increased the bottom curve ($R = \infty$) is approached, corresponding to Eq. 8.53. △

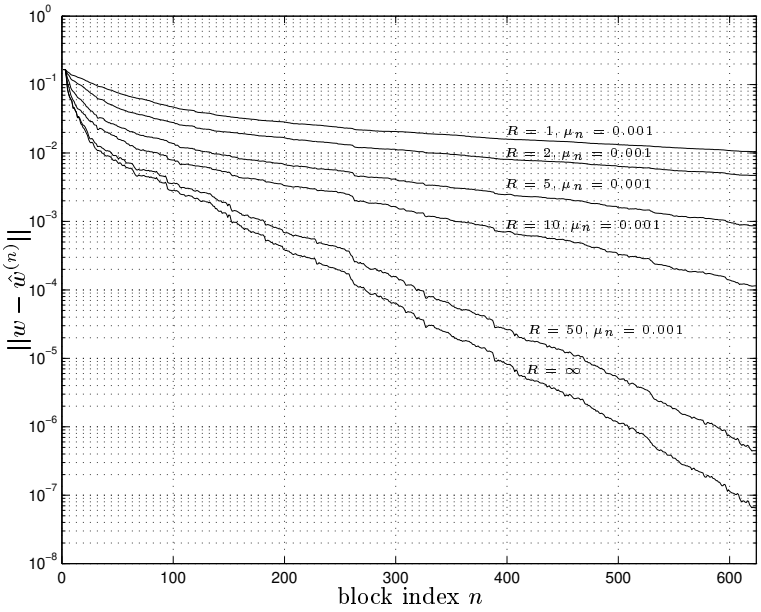


Figure 8.1: A random FIR filter $w$ of order 31 was generated and fed with a white noise signal. The unnormalized unconstrained PBFDRAP was used to identify $w$. The norm between $w$ and the equivalent time–domain representation $\hat{w}$ of the adaptive filter coefficients is plotted as a function of time. It appears that if $R$ is increased the bottom curve ($R = \infty$) is approached, corresponding to a normalized variant of the PBFDAF that is based on projected subband energies and adapted with stepsize $\mu_n = 1$.

### Subband–normalized unconstrained PBFDRAP

The PBFDRAP may be considered as a subband adaptive system (see also section 6.2). The convergence can be optimized by applying different stepsizes to each of the subband adaptive filters. Therefore, matrix $\mathbf{\Delta}$ is typically chosen as in Eq. 6.35. In this way in each subband the adaptation stepsize $\mu$ is divided by the subband energy.

Also in this case the elements of $\mathbf{N}_n$ linearly depend on $\mu$ (see Eq. 8.17). From Eqs. 8.39, 8.40 and 8.42 it follows that projection matrix $\boldsymbol{\mathcal{P}}$ will become independent of $\mu$ if $\boldsymbol{\mathcal{K}}$ is inversely proportional to $\mu$.

By inserting Eqs. 6.35 and 6.36 in Eq. 8.22,

$$\mathbf{e}^{(n,r)} = \begin{bmatrix} \mathbf{I}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & (1-\mu)^{r-1}\mathbf{I}_L \end{bmatrix} \mathbf{e}^{(n,1)} \tag{8.55}$$

is obtained. Therefore, if $\mu \neq 0$

$$\sum_{r=1}^{R} \mathbf{e}^{(n,r)} = \begin{bmatrix} R\mathbf{I}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \frac{1-(1-\mu)^R}{\mu}\mathbf{I}_L \end{bmatrix} \mathbf{e}^{(n,1)}. \tag{8.56}$$

If $0 < \mu < 2$ equation 8.56 converges to

$$\lim_{R\to\infty} \sum_{r=1}^{R} \mathbf{e}^{(n,r)} = \begin{bmatrix} \times & \mathbf{0} \\ \mathbf{0} & \frac{1}{\mu}\mathbf{I}_L \end{bmatrix} \mathbf{e}^{(n,1)}, \tag{8.57}$$

in which $\times$ can be any matrix having the correct size, thanks to the special structure of $\mathbf{e}^{(n,1)}$ (see also Eq. 8.30). If $\times$ is chosen equal to $\frac{1}{\mu}\mathbf{I}_{P-1+\sigma}$ for instance,

$$\lim_{R\to\infty} \sum_{r=1}^{R} \mathbf{e}^{(n,r)} = \frac{1}{\mu}\mathbf{e}^{(n,1)}, \tag{8.58}$$

which implies that $\boldsymbol{\mathcal{K}} = \frac{1}{\mu}\mathbf{I}_{P-1+\sigma}$ and $\boldsymbol{\mathcal{P}} = \mathbf{I}_M$, so that finally by inserting Eqs. 6.35 and 8.58 in Eq. 8.23

$$\lim_{R\to\infty} \underline{\mathbf{w}}_p^{(n,R)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n,1)} + \left( \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \underline{\mathbf{X}}_p^{(n)*} \right)^{-1} \underline{\mathbf{X}}_p^{(n)*} \mathbf{F}\mathbf{e}^{(n,1)} \tag{8.59}$$

is obtained.

*It follows from Eq. 8.59 that the subband–normalized unconstrained PBFDRAP converges to the subband–normalized unconstrained PBFDAF with $\mu = 1$ : increasing the number of iteration steps $R$ has the same effect as applying a larger stepsize $\mu$.*

**Example 8.2** A random FIR filter $w$ of order 15 was generated and fed with a noise signal $x$ (see figure 2.8) that was obtained by filtering a white noise sequence with a first order lowpass IIR filter having a pole at 0.4. The subband–normalized unconstrained PBFDRAP ($P = L = 8$, $M = 16$ and $L_{FB} = 16$) was applied to identify $w$. The ambiguity, which typically occurs with unconstrained updating, was compensated for by applying Eqs. 8.10 and 8.11. In figure 8.2 the norm between $w$ and the equivalent time–domain representation $\hat{w}$ of the adaptive filter coefficients (Eqs. 8.1 and 8.2) is plotted as a function of time. It appears that if $R$ is increased the bottom curve ($R = \infty$) is approached, which corresponds to the subband–normalized unconstrained PBFDAF with $\mu = 1$. △

### 8.2.3 Constrained PBFDRAP : $\lim_{R\to\infty} \underline{\mathbf{w}}_p^{(n,R)}$

In section 8.2.2 it was verified whether and when conditions 8.48, 8.49 and 8.50 are met if unconstrained updating is applied. In this section the constrained PBFDRAP

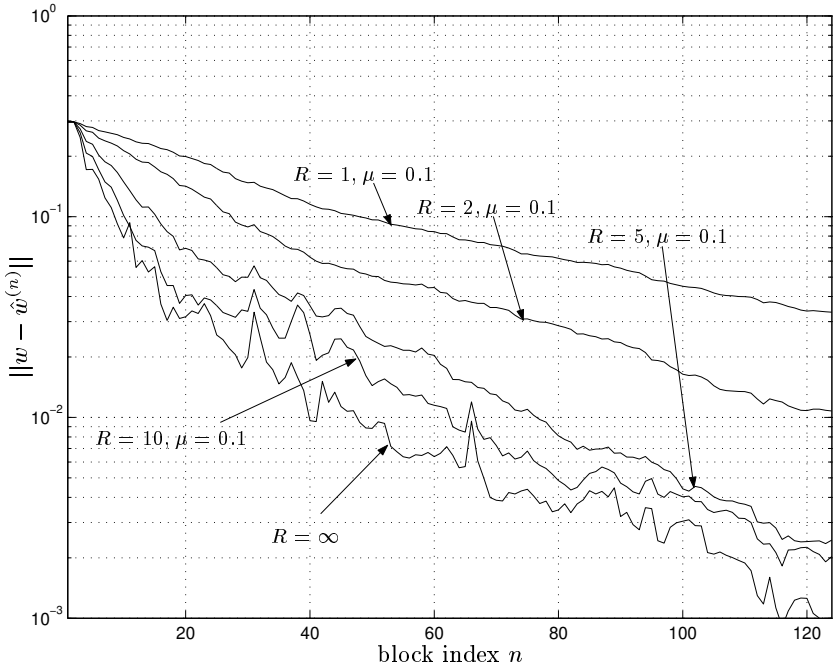


Figure 8.2: A random FIR filter $w$ of order 15 was generated and fed with a noise signal, which was obtained by filtering a white noise sequence with a first order lowpass IIR filter having a pole at 0.4. The subband–normalized unconstrained PBFDRAP was applied to identify $w$. The norm between $w$ and the equivalent time–domain representation $\hat{w}$ of the adaptive filter coefficients is plotted as a function of time. It appears that if $R$ is increased the bottom curve ($R = \infty$) is approached, which corresponds to the subband–normalized unconstrained PBFDAF with $\mu = 1$ : increasing the number of iteration steps $R$ has the same effect as applying a larger stepsize $\mu$.

is analyzed, also making a distinction between the unnormalized implementation and subband normalization.

If it is assumed that the data $(x, d)$ as defined in figure 2.8 come from a real–life scenario and that all stepsizes are different from zero, the rank of $\underline{\mathbf{X}}_p^{(n)} \mathbf{G} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)^*}$ is $P$, due to the structure of $\mathbf{G}$ (see Eq. 8.9). The rank of $\mathbf{P}_n$ (Eq. 8.16) on the other hand will be equal to $\min(M, P\frac{L_{FB}}{P}) = \min(M, L_{FB})$ under the same assumptions. The rank of

$$\mathbf{N}_n = \begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{G} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)^*} \mathbf{F} \tag{8.60}$$

is then given by $\min(\min(M, L_{FB}), L)$.

**Theorem 8.4** *For the constrained PBFDRAP*

$$\mathbf{P}_n = \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{L-1+\sigma} \end{bmatrix} \mathbf{F}^{-1} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)^*} \mathbf{F} \tag{8.61}$$

*is a real–valued matrix if the adaptive filter is applied to a real–life identification setup as in figure 1.5, for which all input signals and the unknown system $w$ are real–valued.*

**Proof:** The proof of theorem 8.4 can be found in appendix D.4.

### Unnormalized and globally normalized constrained PBFDRAP

**Theorem 8.5** *For the unnormalized and globally normalized constrained PBFDRAP for which* $\boldsymbol{\Delta} = \mu_n \boldsymbol{I}_M$,

$$\mathbf{P}_n = \mu_n \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{L-1+\sigma} \end{bmatrix} \mathbf{F}^{-1} \underline{\mathbf{X}}_p^{(n)^*} \mathbf{F} = \begin{bmatrix} \boldsymbol{\mathcal{A}} & \boldsymbol{\mathcal{B}} \\ \boldsymbol{\mathcal{C}} & \boldsymbol{\mathcal{D}} \end{bmatrix} \tag{8.62}$$

*is a hermitian positive semi–definite matrix. Further, also $L \times L$ matrix $\boldsymbol{\mathcal{D}}$ is hermitian and positive semi–definite.*

**Proof:** Theorem 8.5 is proven in appendix D.5.

From theorem 8.5 it can be concluded that the eigenvalues of $\boldsymbol{\mathcal{D}}$, i.e. the diagonal elements of $\boldsymbol{\Gamma}_1$ are real and positive.

In the following paragraphs it is checked which of the conditions 8.48, 8.49 and 8.50 are valid. Three cases can be distinguished depending on the relation between the equivalent time–domain length $L_{FB}$ of the adaptive filter $\hat{w}$ (Eqs. 8.1 & 8.2), the block length $L$ and the DFT size $M$.

#### $\boldsymbol{L_{FB} \geqslant L}$

For the type of application the PBFDRAP is typically used for, e.g. for the identification of high–order FIR systems, $L_{FB} \geqslant L$ is a realistic assumption. Hence, the rank of $\mathbf{N}_n$ is $\min(\min(M, L_{FB}), L) = L$, if the filter is adapted. $\boldsymbol{\Gamma}_1$ will be a full rank diagonal matrix and $\boldsymbol{\mathcal{D}}$ is strictly positive definite. If the stepsizes $\mu_n$ and hence $||\boldsymbol{\Delta}||$ are sufficiently small all eigenvalues of $\boldsymbol{\Gamma}_1$ are between 0 and 2 and $\boldsymbol{\Xi}^{(R)}$ converges to $\boldsymbol{\Gamma}_1^{-1}$ if $R$ goes to infinity (Eq. 8.32). As conditions 8.48 and 8.50 are fulfilled for the unnormalized and globally normalized constrained PBFDRAP if $L_{FB} \geqslant L$, equations 8.18–8.37 are applicable and the algorithm converges to equation 8.35 if $R$ goes to infinity.

***$L_{FB} \geqslant M$***

If furthermore $L_{FB} \geqslant M \geqslant L$, which is a more restrictive, but still realistic assumption if the PBFDRAP is used for the identification of high–order FIR systems, $\mathbf{P}_n$ has rank $M$ if $\mu_n \neq 0$. All conditions (Eqs. 8.48, 8.49 and 8.50) are now fulfilled, hence equations 8.38–8.47 are valid and the algorithm converges to equation 8.43 if $R$ goes to infinity.

***$L_{FB} < L$***

If $L_{FB} < L$, the algorithm still converges, but the formulas derived in Eqs. 8.19–8.21 and 8.27–8.47 are not applicable anymore. Nevertheless, also in this case $\lim_{R\to\infty} \underline{\mathbf{w}}_p^{(n,R)}$ can be computed. The derivations can be found in appendix D.6.

**Relation with the Affine Projection Algorithm**

**Theorem 8.6** *The weight updating of the unnormalized and globally normalized constrained PBFDRAP for which $L_{FB} \geqslant L$ (i.e. Eqs. 8.35, 8.43) approaches the weight updating of the Partial Rank Algorithm (PRA) with $\mu = 1$. Stepsize $\mu = 1$ guarantees maximum convergence speed.*

The PRA is a block version of the Affine Projection Algorithm (APA) and can be interpreted as a block–normalized variant of the block–LMS algorithm. More information on the Affine Projection Algorithm can be found in section 2.3.1. Block–LMS and the PRA are discussed in section 2.3.2.

**Proof:** In [40] it was proven that iterated block–LMS approaches PRA with stepsize $\mu_n = 1$, if the number of iteration steps goes to infinity. Stepsize $\mu_n = 1$ guarantees maximum convergence speed. As the unnormalized constrained PBFDAF is an exact representation of block–LMS in the frequency domain, the unnormalized and also the globally normalized constrained PBFDRAP will approach the PRA, if $R$ goes to infinity. □

**Theorem 8.7** *The unnormalized and globally normalized constrained PBFDRAP for which $L_{FB} \geqslant L$ approach the PRA by series expansion of the matrix inverse in Eq. 2.55.*

**Proof:** The proof can be found in appendix D.7.

**Experiment 8.1** An artificial acoustic environment was simulated using the image method presented in [4] : an FIR filter $w$ of 1024 taps was generated and fed with a colored noise signal $x$ (see figure 1.5) that was obtained by filtering a white noise sequence with a first order lowpass IIR filter having a pole at 0.4. The globally normalized constrained PBFDRAP ($P = L = 128$, $M = 256$ and $L_{FB} = 1024$) was used to identify $w$. The number of iteration steps $R$ was increased from 1 to 10.

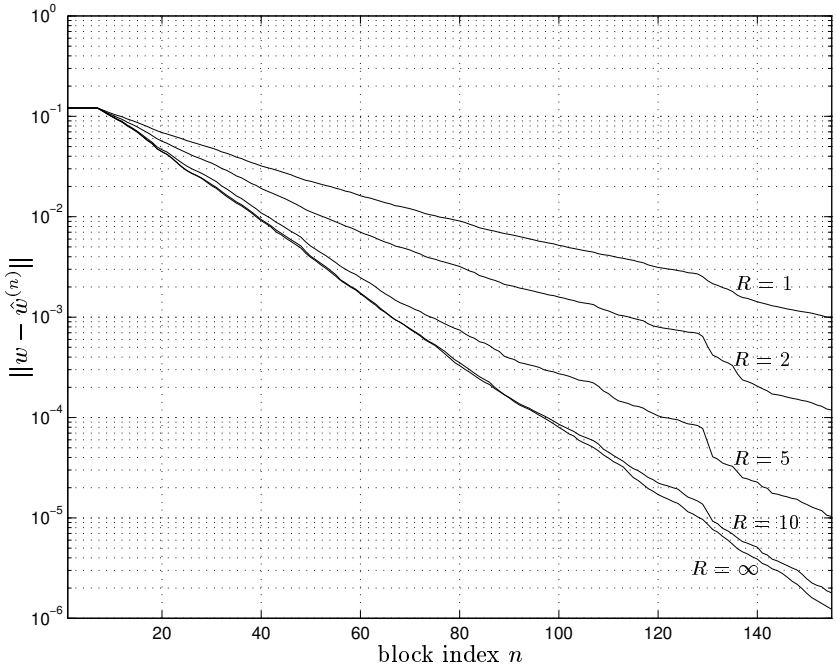


Figure 8.3: An artificial acoustic path was generated and fed with colored noise $x$. The globally normalized constrained PBFDRAP was used to identify $w$. The norm between $w$ and the equivalent time–domain representation $\hat{w}$ of the adaptive filter coefficients is plotted as a function of time. It appears that if $R$ is increased the bottom curve ($R = \infty$) is approached, which corresponds to the PRA with stepsize $\mu_n = 1$.

For each curve the adaptation constant $\mu$ (see Eq. 6.34) was optimized to obtain maximum convergence speed. In figure 8.3 the norm between $w$ and the equivalent time–domain representation $\hat{w}$ of the adaptive filter coefficients (Eqs. 8.1 and 8.2) is plotted as a function of time. It appears that if $R$ is increased the bottom curve ($R = \infty$) is approached, which corresponds to the PRA ($L = 128, L_{FB} = 1024$) with stepsize $\mu_n = 1$. $\bigtriangledown$

### Subband–normalized constrained PBFDRAP

Also for the subband–normalized constrained PBFDRAP a similar derivation can be made to check what happens if $R$ goes to infinity. Stability problems arise however in this case as it is found that sometimes the algorithm diverges for large values of $R$.

Unlike in section 8.2.3, matrix $\boldsymbol{\mathcal{D}}$, defined by Eq. 8.16, is typically not hermitian here. In the case of the subband–normalized constrained PBFDRAP for which $\boldsymbol{\Delta} = \mathrm{diag}(\boldsymbol{\mu}_s^{(n)})$, being a diagonal matrix with subband dependent stepsizes $\boldsymbol{\mu}_s^{(n)}(m)$, the eigenvalues of $\boldsymbol{\mathcal{D}}$, i.e. the diagonal elements of $\boldsymbol{\Gamma}_1$, are not always real and positive. Hence, for $L_{FB} \geqslant L$, (see Eq. 8.28)

$$\lim_{R\to\infty} \boldsymbol{\Xi}^{(R)} = \lim_{R\to\infty} \boldsymbol{\Gamma}_1^{-1} \left( \mathbf{I}_L - (\mathbf{I}_L - \boldsymbol{\Gamma}_1)^R \right) \tag{8.63}$$

may not be bounded and for some data sets the algorithm will convergence to an unstable filter. However, this does not automatically imply divergence of the algorithm for finite $R$, but indicates that for some data sets iterating this algorithm is expected to invoke stability problems. It appears that like with the subband–normalized unconstrained PBFDRAP (section 8.2.2) also in this case increasing the number of iteration steps $R$ has the same effect as applying a larger stepsize $\mu$.

**Experiment 8.2** A white noise signal $x$ was fed into a randomly generated FIR filter $w$ of order 15. Filter $w$ was adaptively estimated using the subband–normalized constrained PBFDRAP ($L_{FB} = 16$, $M = 16$, $P = L = 8$, $\mu = 0.2$). The results are shown in figure 8.4. If $R$ is increased from 1 to 10 the bottom curve ($R_{\mathrm{lim}}$) is approached. If $R$ is further increased to 50 and higher the filter becomes unstable. ▽

In order to overcome the instability problem and avoid divergence, regularization can be applied to the subband energy estimates. The subband adaptation stepsizes are then chosen inversely proportional to both the subband energy (Eq. 6.35) and a global energy estimate (Eq. 6.33) :

$$\boldsymbol{\Delta} = \mu \left( (1-\alpha) \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \underline{\mathbf{X}}_p^{(n)\,*} + \frac{\alpha}{M} \left( \sum_{m=0}^{M-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)}(m) \underline{\mathbf{X}}_p^{(n)\,*}(m) \right) \mathbf{I}_M \right)^{-1}, \tag{8.64}$$

in which $\alpha$ weighs the subband energy versus the global energy and is typically a small positive real number for which $0 \leqslant \alpha \ll 1$, $\alpha \in \mathbb{R}$. If $\alpha = 0$ the subband normalized PBFDRAP is obtained. On the other hand, $\alpha = 1$ corresponds to the globally normalized PBFDRAP. If $\alpha$ is increased from 0 to 1 stability will improve, but convergence typically will go down.

### 8.2.4 Summary

In section 8.2 it was shown that the PBFDRAP algorithm approaches well–known adaptive filtering algorithms if the number of iteration steps $R$ goes to infinity. Different algorithmic settings were considered : unconstrained versus constrained updating on the one hand and subband normalization versus an unnormalized or globally normalized implementation on the other hand.

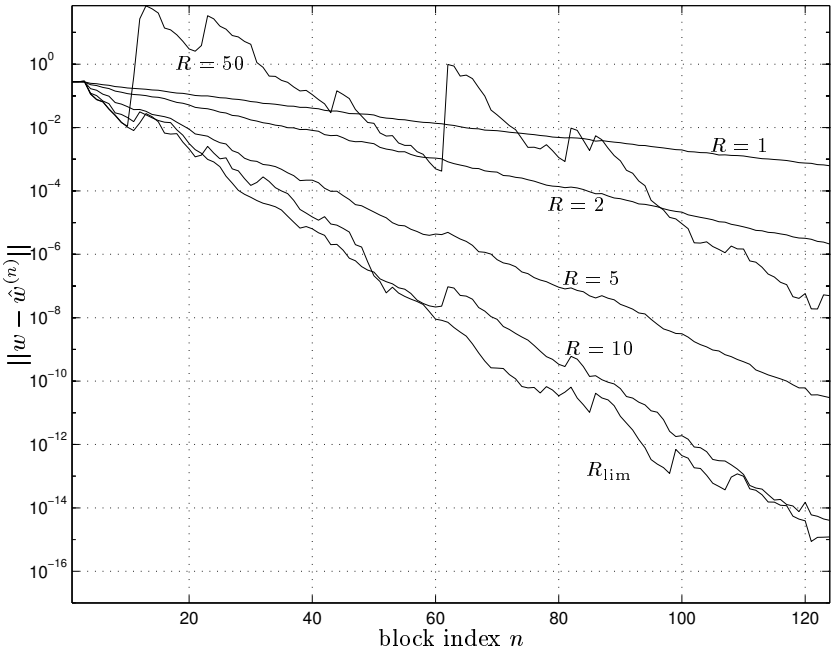


Figure 8.4: Subband–normalized constrained PBFDRAP : the norm between $w$ and $\hat{w}$, the equivalent time–domain representation of the adaptive filter coefficients, is plotted as a function of time. It appears that if $R$ is increased the subband–normalized constrained PBFDRAP becomes unstable.

It was found that

- the unnormalized unconstrained PBFDRAP approaches a normalized variant of the unconstrained PBFDAF that is based on projected subband energies and that is adapted with stepsize $\mu_n = 1$.
- for the subband–normalized unconstrained PBFDRAP increasing the number of iteration steps $R$ has the same effect as applying a larger stepsize $\mu$.
- the unnormalized constrained PBFDRAP on the other hand approaches the PRA with $\mu = 1$ by series expansion of the expensive matrix inverse operation, if $R$ goes to infinity.
- iterating the subband–normalized constrained PBFDRAP algorithm invokes stability problems.

These results are similar to the conclusions formulated in [40] and [116], which suggests that in general iterated unnormalized algorithms converge to their normalized

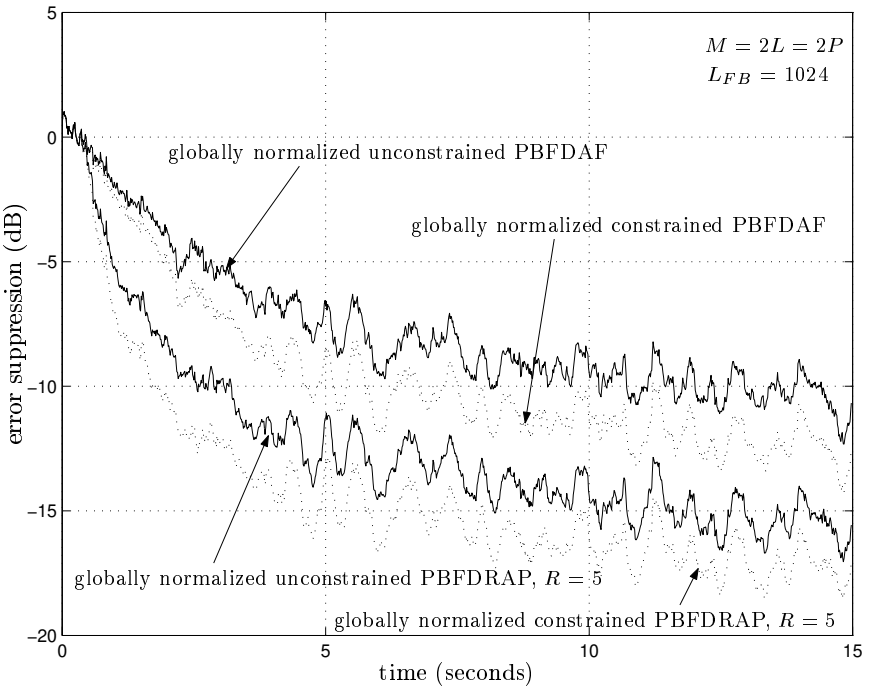


Figure 8.5: The globally normalized PBFDRAP was used to estimate an unknown stationary acoustic path, varying the type of updating (unconstrained, constrained) and the number of iteration steps $R$. The following parameters were applied : $L = P = 128$, $M = 256$, $L_{FB} = 1024$. The time evolution of the a–priori error output $e$ is shown for each of the adaptive filters.

counterparts with $\mu = 1$, leading to maximum convergence speed.

## 8.3 Simulation examples

The different algorithms presented in this chapter were applied to an acoustic echo cancellation setup (see figure 1.5) for validation and comparison. A lowpass noise signal having the spectral characteristics of speech was fed into a loudspeaker and recorded with a microphone in a controlled stationary laboratory environment. No near–end signal $s$ was added on top.

**Experiment 8.3** In a first experiment the unknown acoustic path was estimated using the globally normalized PBFDRAP (cf. Eq. 6.34), varying the type of updating (unconstrained, constrained) and the number of iteration steps $R$. The other

parameters were kept constant and were chosen as follows : $L = P = 128$, $M = 256$, $L_{FB} = 1024$ (see table 8.1 for more explanation). For the unconstrained update Eqs. 8.10 and 8.11 were employed to compensate for the ambiguity that can occur (see section 8.1.1).

For each of the algorithms presented here the optimal stepsize was computed that maximizes the convergence speed. This stepsize was divided by 10 and applied to the adaptive filter, in order to simulate the performance of the algorithms in the presence of double–talk. During double–talk (see also section 10.1.3) the local signal source $s$ is active (see figure 1.5). If double–talk is detected the adaptation constant $\mu$ must be set immediately to zero in order to freeze the coefficients of the adaptive filter. If the adaptation is not switched off the filter coefficients easily drift away from their Wiener solution, leading to a bad model and a larger error output $e$. If the adaptation constant $\mu$ is kept significantly smaller than the theoretical optimum large deviations of the filter coefficients can be avoided. The time evolution of the a–priori error output $e$ for each of the adaptive filters is shown in figure 8.5. $\bigtriangledown$

**Experiment 8.4** In a second experiment the unknown acoustic path was estimated using the subband–normalized PBFDRAP (cf. Eq. 6.35), varying the type of updating (unconstrained, constrained) and the number of iteration steps $R$. Also in this experiment $L = P = 128$, $M = 256$, $L_{FB} = 1024$. The ambiguity was compensated for in the case of unconstrained updating and the adaptation stepsizes were kept an order of magnitude below the optimum. The time evolution of the a–priori error output $e$ for each of the adaptive filters is shown in figure 8.6. $\bigtriangledown$

It is clear that by increasing the number of iteration steps $R$ the performance of the PBFDAF can be improved. As the input signal is colored, the subband–normalized PBFDRAP convergences faster than the globally normalized PBFDRAP. Although stability is not guaranteed for $R \to \infty$, the subband–normalized constrained PBFDRAP offers the best performance for real–life signals and realistic values of $R$. Finally, if the a–posteriori errors are plotted instead of the a–priori errors extra echo enhancement can be obtained. However, during double–talk a–priori errors should be passed to the output to avoid near–end signal cancellation.

## 8.4 Conclusions

In this chapter the PBFDRAP algorithm was considered, an adaptive filtering technique that combines partitioned block frequency–domain adaptive filtering and row action projection. In section 8.1 the algorithm was defined and its working mechanism was explained.

In section 8.2 it was investigated what happens if the number of iterations goes to infinity. Different algorithmic settings were considered. It appears that the

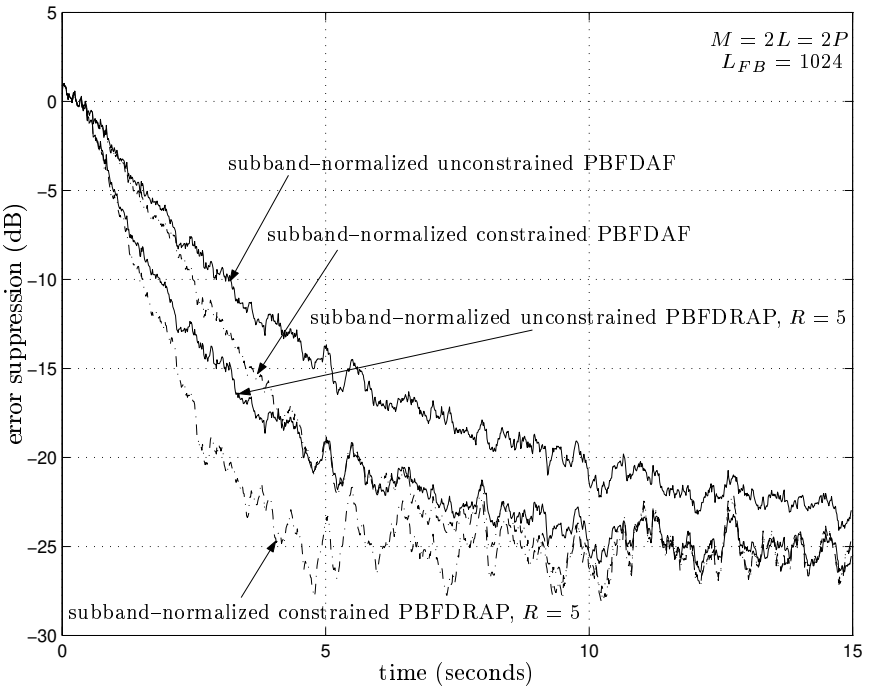


Figure 8.6: The subband–normalized PBFDRAP was used to estimate an unknown stationary acoustic path, varying the type of updating (unconstrained, constrained) and the number of iteration steps $R$. The following parameters were applied : $L = P = 128$, $M = 256$, $L_{FB} = 1024$. The time evolution of the a–priori error output $e$ is shown for each of the adaptive filters.

unnormalized unconstrained PBFDRAP approaches the unconstrained PBFDAF that is normalized based on projected subband energies and that is adapted with stepsize 1. The unnormalized constrained PBFDRAP approaches the PRA with stepsize $\mu = 1$ by series expansion of the expensive matrix inverse operation. For the subband–normalized PBFDRAP increasing the number of iterations has the same effect as applying a larger stepsize. Stability is not guaranteed however in the case of the subband–normalized constrained PBFDRAP.

It was shown in section 8.3 that the PBFDRAP outperforms the PBFDAF in a realistic echo cancellation setup.

for each block of $L$ samples do $\{$

$$\underline{\mathbf{X}}_p^{(n)} \overset{\forall p}{=} \operatorname{diag}\left\{ \mathbf{F} \begin{bmatrix} x((n+1)L - pP - M + 1) \\ \vdots \\ x((n+1)L - pP) \end{bmatrix} \right\}$$

$\hookrightarrow$ $2\mathtt{M}\log_2 \mathtt{M} - 4\mathtt{M} + 6$

(assuming $\underline{\mathtt{X}}_{\mathtt{p}}^{(\mathtt{n})} = \underline{\mathtt{X}}_{\mathtt{0}}^{(\mathtt{n}-\mathtt{pP}/\mathtt{L})}$)

$$\mathbf{d}^{(n)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_n \end{bmatrix} \begin{matrix} \updownarrow P-1+\sigma \\ \updownarrow L \end{matrix}, \quad \mathbf{d}_n = \begin{bmatrix} d[nL+1] \\ \vdots \\ d[(n+1)L] \end{bmatrix} \Big\updownarrow L$$

for $r = 1$ to $R$ do $\{$

$$\mathbf{y}^{(n,r)} = \begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \underline{\mathbf{w}}_p^{(n,r)}$$

$\hookrightarrow$ $\mathtt{R}(2\mathtt{M}\log_2 \mathtt{M} + 4\frac{\mathtt{ML_{FB}}}{\mathtt{P}} + 8\frac{\mathtt{L_{FB}}}{\mathtt{P}} - \mathtt{M} - 2)$

$$\mathbf{e}^{(n,r)} = \mathbf{d}^{(n)} - \mathbf{y}^{(n,r)}$$

$\hookrightarrow$ $\mathtt{RL}$

$$\underline{\mathbf{w}}_p^{(n,r+1)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n,r)} + \mathbf{G}\boldsymbol{\Delta}\underline{\mathbf{X}}_p^{(n)*}\mathbf{F}\mathbf{e}^{(n,r)}$$

$\hookrightarrow$ $\mathtt{R}(2\mathtt{M}\log_2 \mathtt{M} - 3\mathtt{M} + 8 + 4\frac{\mathtt{ML_{FB}}}{\mathtt{P}} + 8\frac{\mathtt{L_{FB}}}{\mathtt{P}})$

(if $\mathtt{G} = \mathtt{I_M}$)

$\hookrightarrow$ $\mathtt{R}((2\mathtt{M} + 4\frac{\mathtt{ML_{FB}}}{\mathtt{P}})\log_2 \mathtt{M} - 3\mathtt{M} + 8 + 14\frac{\mathtt{L_{FB}}}{\mathtt{P}})$

(if $\mathtt{G} = \mathtt{F}\begin{bmatrix} \mathtt{I} & \mathtt{0} \\ \mathtt{0} & \mathtt{0} \end{bmatrix}\mathtt{F}^{-1}$)

$\}$

$$\underline{\mathbf{w}}_p^{(n+1,1)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n,R+1)}$$

$\}$

$$\begin{array}{rcl@{\qquad}rcl} p & = & 0 : \frac{L_{FB}}{P} - 1 & M & = & P + L - 1 + \sigma \\ \mathbf{F}(p,q) & = & e^{-\frac{j2\pi pq}{M}} & \boldsymbol{\Delta} & = & \operatorname{diag}\{\boldsymbol{\mu}_s^{(n)}\} \\ \sigma & \geqslant & 0 & & & \end{array}$$

| unconstrained | constrained | PBFDAF | PBFDRAP |
|---|---|---|---|
| $\mathbf{G} = \mathbf{I}_M$ | $\mathbf{G} = \mathbf{F}\begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{L-1+\sigma} \end{bmatrix}\mathbf{F}^{-1}$ | $R = 1$ | $R > 1$ |

Table 8.1: Standard implementation of the PBFDRAP

# Chapter 9

# Fast Partitioned Block Frequency–Domain RAP

In chapter 8 the PBFDRAP was discussed, which is an iterated version of the PBFDAF algorithm presented in chapter 6. If the number of iteration steps is large the PBFDRAP outperforms the PBFDAF adaptive filter (see section 8.3). The additional cost for reiterating the algorithm following Eqs. 8.1–8.8 however is considerable.

Hence in section 9.1, fast implementations are derived for the PBFDRAP algorithm, making a large number of iteration steps more attractive. Some of the fast implementations presented in section 9.1 are specifically tuned towards the unconstrained PBFDRAP, others to the constrained PBFDRAP. Some versions are applicable to both update schemes.

In section 9.2 different fast implementation schemes are compared with the standard implementation of the PBFDRAP (see section 8.1.1) from a computational complexity point of view. Both for the unconstrained and for the constrained algorithm the complexity gain w.r.t. the standard implementation is computed. It appears that a significant complexity reduction can be obtained.

Theorem 8.6 shows that the infinitely iterated unnormalized and globally normalized constrained PBFDRAP approach the PRA adaptive filter presented in section 2.3.2. In section 9.2.3 therefore the unnormalized and globally normalized constrained PBFDRAP are considered as an alternative for the PRA algorithm. The PBFDRAP is compared with the PRA from a computational complexity perspective. It is seen that for large block lengths the PBFDRAP is a cheaper alternative to the PRA.

The conclusions are formulated in section 9.3.

## 9.1 Fast PBFDRAP

In the previous chapter partitioned block frequency–domain adaptive filtering was combined with the "row action projection" technique, leading to an iterated version of the PBFDAF algorithm, which we called the PBFDRAP. The equations defining the PBFDRAP were presented in section 8.1.1 and are summarized in table 8.1, together with the corresponding computational complexity. The cost estimates are based on the cost analysis performed in appendix C.9.[1]

The PBFDRAP can be considered as an improvement or extension to the PBFDAF algorithm, as by increasing the number of iteration steps $R$ better convergence rates are obtained (see section 8.3). However, the additional cost per iteration is considerable if the standard implementation of table 8.1 is used : iterating $R$ times on the same block of data increases the implementation cost almost by a factor $R$.

In this section fast versions will be derived for the PBFDRAP, leading to a cheaper implementation without changing the algorithmic performance. Some of the presented algorithms put constraints on certain algorithmic parameters (e.g. unnormalized/globally normalized/subband–normalized) or are only valid for one specific updating mode (e.g. unconstrained or constrained). Other fast versions do not impose any restrictions and are applicable to all settings.

### 9.1.1 Fast PBFDRAP, version 1

Referring to the update equations of the PBFDRAP (table 8.1) it is found that at the end of iteration step $R$ the error output $\mathbf{e}^{(n,R)}$ can be written as

$$\mathbf{e}^{(n,R)} = \left( \mathbf{I}_M - \begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{G} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)*} \mathbf{F} \right)^{R-1} \mathbf{e}^{(n,1)} \tag{9.1}$$

and that the adaptive filter weights are given by

$$\underline{\mathbf{w}}_p^{(n,R+1)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n,1)} + \mathbf{G} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)*} \mathbf{F} \sum_{r=1}^{R} \mathbf{e}^{(n,r)}, \quad p = 0 : \frac{L_{FB}}{P} - 1. \tag{9.2}$$

The proof was given in appendix D.1.

Using relations 9.1 and 9.2 the PBFDRAP can be reformulated. Matrix

$$\mathbf{B}_n = \mathbf{I}_M - \begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{G} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)*} \mathbf{F} \tag{9.3}$$

[1] Note that for the complexity analyses performed in this chapter it is always assumed that all signals and the unknown system $w$ are real–valued.

is defined and can be plugged in in Eq. 9.1. Hence, $\mathbf{e}^{(n,r)}$ can be written as

$$\mathbf{e}^{(n,r)} = \mathbf{B}_n^{r-1}\mathbf{e}^{(n,1)} = \mathbf{B}_n\mathbf{e}^{(n,r-1)} \tag{9.4}$$

for $r \in \mathbb{N} > 1$, such that only $\mathbf{e}^{(n,1)}$ needs to be computed explicitly following Eqs. 8.4–8.6. $\mathbf{B}_n$ is computed once for each data block and then $\mathbf{e}^{(n,r)}$, $r > 1$, follow from Eq. 9.4.

In order to compute $\underline{\mathbf{w}}_p^{(n,R+1)}$, Eq. 9.2 can be used. The intermediate error outputs $\mathbf{e}^{(n,r)}$ are accumulated and stored in vector $\mathbf{c}^{(n,r)}$ :

$$\begin{aligned} \mathbf{c}^{(n,1)} &= \mathbf{e}^{(n,1)}, \\ \mathbf{c}^{(n,r)} &= \mathbf{c}^{(n,r-1)} + \mathbf{e}^{(n,r)}, \quad r > 1 \end{aligned} \tag{9.5}$$

such that Eq. 9.2 becomes

$$\underline{\mathbf{w}}_p^{(n,R+1)} = \underline{\mathbf{w}}_p^{(n,1)} + \mathbf{G}\boldsymbol{\Delta}\underline{\mathbf{X}}_p^{(n)^*}\mathbf{F}\mathbf{c}^{(n,R)} \tag{9.6}$$

The different algorithmic steps of this first fast version of the PBFDRAP and a cost estimate are presented in table 9.1.

### 9.1.2 Fast PBFDRAP, version 2

The explicit matrix–vector multiplication $\mathbf{B}_n\mathbf{e}^{(n,r-1)}$ in Eq. 9.4, however, is quite expensive. The algorithm can be simplified by replacing the matrix–vector multiplications by FFT operations. Matrix $\mathbf{D}_n$ is introduced and is defined as

$$\mathbf{D}_n = \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)}\mathbf{G}\boldsymbol{\Delta}\underline{\mathbf{X}}_p^{(n)^*}, \tag{9.7}$$

such that Eq. 9.4 can be written as

$$\mathbf{e}^{(n,r)} = \mathbf{e}^{(n,r-1)} - \begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1}\mathbf{D}_n\mathbf{F}\mathbf{e}^{(n,r-1)}. \tag{9.8}$$

For the unconstrained PBFDRAP for which $\mathbf{G} = \mathbf{I}_M$, $\mathbf{D}_n$ is a diagonal matrix. The matrix–vector product $\mathbf{D}_n\mathbf{F}\mathbf{e}^{(n,r-1)}$ can therefore be replaced by an element–wise vector–vector multiplication, which leads to a reduced cost. The algorithm is summarized in table 9.2.

### 9.1.3 Fast PBFDRAP, version 3

If the order of the unknown FIR system $w$ (see figure 2.8) is high w.r.t. the filter partition length $P$, the algorithm can be further simplified. Note that $\boldsymbol{\Delta}$ and $\underline{\mathbf{X}}_p^{(n)^*}$

in Eq. 9.7 are diagonal matrices and hence commute. If $P = L$ equation 9.7 can be replaced by the following update equations :

$$\bar{\mathbf{D}}_n = \bar{\mathbf{D}}_{n-1} + \underline{\mathbf{X}}_0^{(n)} \mathbf{G} \underline{\mathbf{X}}_0^{(n)\,*} - \underline{\mathbf{X}}_{\frac{L_{FB}}{P}}^{(n)} \mathbf{G} \underline{\mathbf{X}}_{\frac{L_{FB}}{P}}^{(n)\,*} \tag{9.9}$$

$$\mathbf{D}_n = \bar{\mathbf{D}}_n \boldsymbol{\Delta}. \tag{9.10}$$

It this way extra savings can be made : for the unconstrained PBFDRAP the cost for updating $\mathbf{D}_n$ reduces from $2\frac{ML_{FB}}{P} + 4\frac{L_{FB}}{P}$ to $\frac{9}{2}M + 9$, if $\boldsymbol{\Delta} = \mu \mathbf{I}_M$ and all inputs and system $w$ are real. The algorithm is presented in table 9.3.

### 9.1.4 Fast constrained PBFDRAP

The algorithmic schemes presented in table 9.2 and 9.3 are less attractive if constrained updating is used. The matrix–vector product $\mathbf{D}_n \mathbf{F} \mathbf{e}^{(n,r-1)}$ cannot be replaced by an element–wise vector–vector multiplication as matrix $\mathbf{D}_n$ is not diagonal in this case (see Eqs. 9.7 and 8.9).

Starting from the algorithm presented in table 9.1 another fast algorithm can be derived. From Eqs. 8.4, 8.5 and 8.6 it can be verified that $\mathbf{e}^{(n,r)}$ can be written as

$$\mathbf{e}^{(n,r)} = \begin{bmatrix} \mathbf{0}_{P-1+\sigma \times 1} \\ \mathbf{e}_n^{(r)} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} \mathbf{e}_n^{(r)}, \tag{9.11}$$

if constrained updating is applied, with $\mathbf{e}_n^{(r)}$ an $L \times 1$ vector (cf Eq. 8.30). Hence, equation 9.4 can be rewritten as

$$\mathbf{e}_n^{(r)} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{B}_n \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} \mathbf{e}_n^{(r-1)}. \tag{9.12}$$

Now,

$$\hat{\mathbf{B}}_n = \begin{bmatrix} \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{B}_n \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} \tag{9.13}$$

can be updated instead of $\mathbf{B}_n$. If constrained updating is considered and $\boldsymbol{\Delta} = \mu_n \mathbf{I}_M$, Eq. 9.3 can be plugged in in Eq. 9.13, such that

$$\hat{\mathbf{B}}_n = \mathbf{I}_L - \mu_n \begin{bmatrix} \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{F}^{-1} \underline{\mathbf{X}}_p^{(n)\,*} \mathbf{F} \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} \tag{9.14}$$

$$= \mathbf{I}_L - \mu_n \sum_{p=0}^{\frac{L_{FB}}{P}-1} \begin{bmatrix} \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{F}^{-1} \underline{\mathbf{X}}_p^{(n)\,*} \mathbf{F} \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} \tag{9.15}$$

is obtained.

As $\underline{\mathbf{X}}_p^{(n)}$ is a diagonal matrix, defined by Eq. 8.3, $\bar{\mathbf{X}}_{n,p}^T = \mathbf{F}^{-1}\underline{\mathbf{X}}_p^{(n)}\mathbf{F}$ is a right–circulant $M \times M$ matrix (see also appendix A). Hence, the discrete Fourier transform of the first row of $\bar{\mathbf{X}}_{n,p}$ are the diagonal elements of $\underline{\mathbf{X}}_p^{(n)}$. Taking into account Eq. 8.3,

$$\bar{\mathbf{X}}_{n,p} = (\mathbf{F}^{-1}\underline{\mathbf{X}}_p^{(n)}\mathbf{F})^T \tag{9.16}$$

$$= \begin{bmatrix} x[(n+1)L-pP-M+1] & \dots & x[(n+1)L-pP] \\ x[(n+1)L-pP] & \dots & x[(n+1)L-pP-1] \\ \vdots & \ddots & \vdots \\ x[(n+1)L-pP-M+2] & \dots & x[(n+1)L-pP-M+1] \end{bmatrix} \tag{9.17}$$

is found.

Now, as $\bar{\mathbf{X}}_{n,p}^* = (\mathbf{F}^{-1}\underline{\mathbf{X}}_p^{(n)}\mathbf{F})^H = \mathbf{F}^{-1}\underline{\mathbf{X}}_p^{(n)^*}\mathbf{F}$

$$\hat{\mathbf{B}}_n = \mathbf{I}_L - \mu_n \sum_{p=0}^{\frac{L_{FB}}{P}-1} \left( \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \bar{\mathbf{X}}_{n,p}^* \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} \right)^H \left( \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \bar{\mathbf{X}}_{n,p}^* \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} \right) \tag{9.18}$$

$$= \mathbf{I}_L - \mu_n \sum_{p=0}^{\frac{L_{FB}}{P}-1} \mathbf{X}_{n,p}^T \mathbf{X}_{n,p}^* \tag{9.19}$$

with $\mathbf{X}_{n,p}$ the following $P \times L$ Toeplitz matrix :

$$\mathbf{X}_{n,p} = \begin{bmatrix} x[nL-pP+1] & \dots & x[(n+1)L-pP] \\ \vdots & \ddots & \vdots \\ x[nL-pP-P+2] & \dots & x[(n+1)L-pP-P+1] \end{bmatrix}. \tag{9.20}$$

If $P = L$ equation 9.19 can be replaced by the following update equation

$$\bar{\mathbf{B}}_n = \bar{\mathbf{B}}_{n-1} + \mathbf{X}_{n,0}^T \mathbf{X}_{n,0}^* - \mathbf{X}_{n,\frac{L_{FB}}{P}}^T \mathbf{X}_{n,\frac{L_{FB}}{P}}^* \tag{9.21}$$

$$\hat{\mathbf{B}}_n = \mathbf{I}_L - \mu_n \bar{\mathbf{B}}_n. \tag{9.22}$$

It this way extra savings can be made. The algorithm is presented in table 9.4.

### 9.1.5 Summary

In this section four fast implementations of the PBFDRAP adaptive filter were discussed. Which of the algorithmic schemes, presented in table 8.1, 9.1, 9.2, 9.3 or 9.4, will lead to the cheapest implementation, depends on the actual values of the parameters ($M, L_{FB}, L, P$ and $R$) and the type of updating ((un)normalized/ (un)constrained) that is used. For a typical application for which the PBFDRAP is used, i.e. for the identification of high–order FIR systems, the scheme presented in table 9.3 will in general be most appropriate if unconstrained updating is preferred. For the constrained PBFDRAP the first (non–fast) and the last scheme (table 8.1 and 9.4) seem most attractive.

for each block of $L$ samples do {

$$\underline{\mathbf{X}}_p^{(n)} \overset{\forall p}{=} \operatorname{diag}\left\{ \mathbf{F} \begin{bmatrix} x((n+1)L - pP - M + 1) \\ \vdots \\ x((n+1)L - pP) \end{bmatrix} \right\}$$

$\hookrightarrow$ $2\mathtt{M}\log_2\mathtt{M} - 4\mathtt{M} + 6$
(assuming $\underline{\mathtt{X}}_{\mathtt{p}}^{(\mathtt{n})} = \underline{\mathtt{X}}_{\mathtt{0}}^{(\mathtt{n}-\mathtt{pP}/\mathtt{L})}$)

$$\mathbf{B}_n = \mathbf{I}_M - \begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{G} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)^*} \mathbf{F}$$

$\hookrightarrow$ $2\mathtt{M}\log_2\mathtt{M} + 2\frac{\mathtt{ML_{FB}}}{\mathtt{P}} + 4\frac{\mathtt{L_{FB}}}{\mathtt{P}} + 1$
(if $\mathtt{G} = \mathtt{I_M}$, $\boldsymbol{\Delta} = \mu \mathtt{I_M}$)

$$\mathbf{y}^{(n,1)} = \begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \underline{\mathbf{w}}_p^{(n)}$$

$\hookrightarrow$ $2\mathtt{M}\log_2\mathtt{M} + 4\frac{\mathtt{ML_{FB}}}{\mathtt{P}} + 8\frac{\mathtt{L_{FB}}}{\mathtt{P}} - \mathtt{M} - 2$

$$\mathbf{d}^{(n)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_n \end{bmatrix} \begin{matrix} \updownarrow P-1+\sigma \\ \updownarrow L \end{matrix}, \quad \mathbf{d}_n = \begin{bmatrix} d[nL+1] \\ \vdots \\ d[(n+1)L] \end{bmatrix} \Big\updownarrow L$$

$$\mathbf{e}^{(n,1)} = \mathbf{d}^{(n)} - \mathbf{y}^{(n,1)}$$

$\hookrightarrow$ $\mathtt{L}$

$$\mathbf{c}^{(n,1)} = \mathbf{e}^{(n,1)}$$

for $r = 2$ to $R$ do {

$$\mathbf{e}^{(n,r)} = \mathbf{B}_n \mathbf{e}^{(n,r-1)}$$

$\hookrightarrow$ $(\mathtt{R}-1)(2\mathtt{L}^2 - \mathtt{L})$

$$\mathbf{c}^{(n,r)} = \mathbf{c}^{(n,r-1)} + \mathbf{e}^{(n,r)}$$

$\hookrightarrow$ $(\mathtt{R}-1)\mathtt{L}$

}

$$\underline{\mathbf{w}}_p^{(n+1)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n)} + \mathbf{G} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)^*} \mathbf{F} \mathbf{c}^{(n,R)}$$

$\hookrightarrow$ $2\mathtt{M}\log_2\mathtt{M} - 3\mathtt{M} + 8 + 4\frac{\mathtt{ML_{FB}}}{\mathtt{P}} + 8\frac{\mathtt{L_{FB}}}{\mathtt{P}}$
(if $\mathtt{G} = \mathtt{I_M}$)

}

$$\begin{array}{rcl@{\qquad}rcl} p & = & 0 : \frac{L_{FB}}{P} - 1 & M & = & P + L - 1 + \sigma \\ \mathbf{F}(p,q) & = & e^{-\frac{j2\pi pq}{M}} & \boldsymbol{\Delta} & = & \operatorname{diag}\{\boldsymbol{\mu}_s^{(n)}\} \\ \sigma & \geqslant & 0 & & & \end{array}$$

| unconstrained | constrained | PBFDAF | PBFDRAP |
|---|---|---|---|
| $\mathbf{G} = \mathbf{I}_M$ | $\mathbf{G} = \mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{L-1+\sigma} \end{bmatrix} \mathbf{F}^{-1}$ | $R = 1$ | $R > 1$ |

Table 9.1: Fast PBFDRAP, version 1

for each block of $L$ samples do {

$$\underline{\mathbf{X}}_p^{(n)} \overset{\forall p}{=} \operatorname{diag}\left\{\mathbf{F}\begin{bmatrix} x((n+1)L-pP-M+1) \\ \vdots \\ x((n+1)L-pP) \end{bmatrix}\right\}$$

$\hookrightarrow$ $2\mathtt{M}\log_2\mathtt{M} - 4\mathtt{M} + 6$
(assuming $\underline{\mathtt{X}}_{\mathtt{p}}^{(\mathtt{n})} = \underline{\mathtt{X}}_{\mathtt{0}}^{(\mathtt{n}-\mathtt{pP}/\mathtt{L})}$)

$$\mathbf{D}_n = \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{G}\boldsymbol{\Delta}\underline{\mathbf{X}}_p^{(n)\,*}$$

$\hookrightarrow$ $2\frac{\mathtt{ML_{FB}}}{\mathtt{P}} + 4\frac{\mathtt{L_{FB}}}{\mathtt{P}}$
(if $\mathtt{G} = \mathtt{I_M}$, $\boldsymbol{\Delta} = \mu\mathtt{I_M}$)

$$\mathbf{y}^{(n,1)} = \begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \underline{\mathbf{w}}_p^{(n)}$$

$\hookrightarrow$ $2\mathtt{M}\log_2\mathtt{M} + 4\frac{\mathtt{ML_{FB}}}{\mathtt{P}} + 8\frac{\mathtt{L_{FB}}}{\mathtt{P}} - \mathtt{M} - 2$

$$\mathbf{d}^{(n)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_n \end{bmatrix} \begin{matrix} \updownarrow P-1+\sigma \\ \updownarrow L \end{matrix}, \quad \mathbf{d}_n = \begin{bmatrix} d[nL+1] \\ \vdots \\ d[(n+1)L] \end{bmatrix} \Big\uparrow L \Big\downarrow$$

$$\mathbf{e}^{(n,1)} = \mathbf{d}^{(n)} - \mathbf{y}^{(n,1)}$$

$\hookrightarrow$ $\mathtt{L}$

$$\mathbf{c}^{(n,1)} = \mathbf{e}^{(n,1)}$$

for $r = 2$ to $R$ do {

$$\underline{\mathbf{e}}^{(n,r-1)} = \mathbf{F}\mathbf{e}^{(n,r-1)}$$

$\hookrightarrow$ $(\mathtt{R}-1)(2\mathtt{M}\log_2\mathtt{M} - 4\mathtt{M} + 6)$

$$\mathbf{a}^{(n,r)} = \begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1}\mathbf{D}_n\underline{\mathbf{e}}^{(n,r-1)}$$

$\hookrightarrow$ $(\mathtt{R}-1)(2\mathtt{M}\log_2\mathtt{M} + \mathtt{M} + 2)$
(if $\mathtt{G} = \mathtt{I_M}$)

$$\mathbf{e}^{(n,r)} = \mathbf{e}^{(n,r-1)} - \mathbf{a}^{(n,r)}$$

$\hookrightarrow$ $(\mathtt{R}-1)\mathtt{L}$

$$\mathbf{c}^{(n,r)} = \mathbf{c}^{(n,r-1)} + \mathbf{e}^{(n,r)}$$

$\hookrightarrow$ $(\mathtt{R}-1)\mathtt{L}$

}

$$\underline{\mathbf{w}}_p^{(n+1)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n)} + \mathbf{G}\boldsymbol{\Delta}\underline{\mathbf{X}}_p^{(n)\,*}\mathbf{F}\mathbf{c}^{(n,R)}$$

$\hookrightarrow$ $2\mathtt{M}\log_2\mathtt{M} - 3\mathtt{M} + 8 + 4\frac{\mathtt{ML_{FB}}}{\mathtt{P}} + 8\frac{\mathtt{L_{FB}}}{\mathtt{P}}$
(if $\mathtt{G} = \mathtt{I_M}$)

}

$$\begin{array}{rcl@{\qquad}rcl} p & = & 0 : \frac{L_{FB}}{P} - 1 & M & = & P + L - 1 + \sigma \\ \mathbf{F}(p,q) & = & e^{-\frac{j2\pi pq}{M}} & \boldsymbol{\Delta} & = & \operatorname{diag}\{\boldsymbol{\mu}_s^{(n)}\} \\ \sigma & \geqslant & 0 & & & \end{array}$$

| unconstrained | constrained | PBFDAF | PBFDRAP |
|---|---|---|---|
| $\mathbf{G} = \mathbf{I}_M$ | $\mathbf{G} = \mathbf{F}\begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{L-1+\sigma} \end{bmatrix}\mathbf{F}^{-1}$ | $R = 1$ | $R > 1$ |

Table 9.2: Fast PBFDRAP, version 2

for each block of $L$ samples do {

$$\underline{\mathbf{X}}_p^{(n)} \overset{\forall p}{=} \text{diag}\left\{ \mathbf{F} \begin{bmatrix} x((n+1)L - pP - M + 1) \\ \vdots \\ x((n+1)L - pP) \end{bmatrix} \right\}$$

$\hookrightarrow$ $2\mathtt{M}\log_2\mathtt{M} - 4\mathtt{M} + 6$
(assuming $\underline{\mathtt{X}}_{\mathtt{P}}^{(\mathtt{n})} = \underline{\mathtt{X}}_{\mathtt{0}}^{(\mathtt{n}-\mathtt{pP}/\mathtt{L})}$)

$$\bar{\mathbf{D}}_n = \bar{\mathbf{D}}_{n-1} + \underline{\mathbf{X}}_0^{(n)} \mathbf{G} \underline{\mathbf{X}}_0^{(n)*} - \underline{\mathbf{X}}_{L_{FB}/P}^{(n)} \mathbf{G} \underline{\mathbf{X}}_{L_{FB}/P}^{(n)*}$$
$$\mathbf{D}_n = \bar{\mathbf{D}}_n \boldsymbol{\Delta}$$

$\hookrightarrow$ $\frac{9}{2}\mathtt{M} + 9$
(if $\mathtt{G} = \mathtt{I_M}$ and $\boldsymbol{\Delta} = \mu \mathtt{I_M}$)

$$\mathbf{y}^{(n,1)} = \begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \underline{\mathbf{w}}_p^{(n)}$$

$\hookrightarrow$ $2\mathtt{M}\log_2\mathtt{M} + 4\frac{\mathtt{ML_{FB}}}{\mathtt{P}} + 8\frac{\mathtt{L_{FB}}}{\mathtt{P}} - \mathtt{M} - 2$

$$\mathbf{d}^{(n)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_n \end{bmatrix} \begin{matrix} \updownarrow P-1+\sigma \\ \updownarrow L \end{matrix}, \quad \mathbf{d}_n = \begin{bmatrix} d[nL+1] \\ \vdots \\ d[(n+1)L] \end{bmatrix} \begin{matrix} \uparrow \\ L \\ \downarrow \end{matrix}$$

$$\mathbf{e}^{(n,1)} = \mathbf{d}^{(n)} - \mathbf{y}^{(n,1)}$$

$\hookrightarrow$ $\mathtt{L}$

$$\mathbf{c}^{(n,1)} = \mathbf{e}^{(n,1)}$$

for $r = 2$ to $R$ do {

$$\underline{\mathbf{e}}^{(n,r-1)} = \mathbf{F}\mathbf{e}^{(n,r-1)}$$

$\hookrightarrow$ $(\mathtt{R}-1)(2\mathtt{M}\log_2\mathtt{M} - 4\mathtt{M} + 6)$

$$\mathbf{a}^{(n,r)} = \begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \mathbf{D}_n \underline{\mathbf{e}}^{(n,r-1)}$$

$\hookrightarrow$ $(\mathtt{R}-1)(2\mathtt{M}\log_2\mathtt{M} + \mathtt{M} + 2)$
(if $\mathtt{G} = \mathtt{I_M}$)

$$\mathbf{e}^{(n,r)} = \mathbf{e}^{(n,r-1)} - \mathbf{a}^{(n,r)}$$

$\hookrightarrow$ $(\mathtt{R}-1)\mathtt{L}$

$$\mathbf{c}^{(n,r)} = \mathbf{c}^{(n,r-1)} + \mathbf{e}^{(n,r)}$$

$\hookrightarrow$ $(\mathtt{R}-1)\mathtt{L}$

}

$$\underline{\mathbf{w}}_p^{(n+1)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n)} + \mathbf{G}\boldsymbol{\Delta}\underline{\mathbf{X}}_p^{(n)*}\mathbf{F}\mathbf{c}^{(n,R)}$$

$\hookrightarrow$ $2\mathtt{M}\log_2\mathtt{M} - 3\mathtt{M} + 8 + 4\frac{\mathtt{ML_{FB}}}{\mathtt{P}} + 8\frac{\mathtt{L_{FB}}}{\mathtt{P}}$
(if $\mathtt{G} = \mathtt{I_M}$)

}

$$\begin{array}{rcl@{\qquad}rcl} p & = & 0 : \frac{L_{FB}}{P} - 1 & M & = & P + L - 1 + \sigma \\ \mathbf{F}(p,q) & = & e^{-\frac{j2\pi pq}{M}} & \boldsymbol{\Delta} & = & \text{diag}\{\boldsymbol{\mu}_s^{(n)}\} \\ \sigma & \geqslant & 0 & P & = & L \end{array}$$

| unconstrained | constrained | PBFDAF | PBFDRAP |
|---|---|---|---|
| $\mathbf{G} = \mathbf{I}_M$ | $\mathbf{G} = \mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{L-1+\sigma} \end{bmatrix} \mathbf{F}^{-1}$ | $R = 1$ | $R > 1$ |

Table 9.3: Fast PBFDRAP, version 3 : $P = L$

for each block of $L$ samples do {

$$\underline{\mathbf{X}}_p^{(n)} \overset{\forall p}{=} \operatorname{diag}\left\{ \mathbf{F} \begin{bmatrix} x((n+1)L - pP - M + 1) \\ \vdots \\ x((n+1)L - pP) \end{bmatrix} \right\}$$

$\hookrightarrow 2\mathtt{M}\log_2 \mathtt{M} - 4\mathtt{M} + 6$

(assuming $\underline{\mathtt{X}}_p^{(\mathtt{n})} = \underline{\mathtt{X}}_0^{(\mathtt{n}-\mathtt{p}\mathtt{P}/\mathtt{L})}$)

$$\mathbf{X}_{n,p} \overset{\forall p}{=} \begin{bmatrix} x[nL - pP + 1] & \cdots & x[(n+1)L - pP] \\ \vdots & \ddots & \vdots \\ x[nL - pP - P + 2] & \cdots & x[(n+1)L - pP - P + 1] \end{bmatrix}$$

$$\bar{\mathbf{B}}_n = \bar{\mathbf{B}}_{n-1} + \mathbf{X}_{n,0}^T \mathbf{X}_{n,0}^* - \mathbf{X}_{n,L_{FB}/P}^T \mathbf{X}_{n,L_{FB}/P}^*$$

$\hookrightarrow \mathtt{L}^3 + \frac{3}{2}\mathtt{L}^2 + \frac{\mathtt{L}}{2}$

(assuming $\mathtt{P} = \mathtt{L}$)

$$\hat{\mathbf{B}}_n = \mathbf{I}_L - \mu_n \bar{\mathbf{B}}_n$$

$\hookrightarrow \frac{1}{2}\mathtt{L}^2 + \frac{3}{2}\mathtt{L}$

$$\mathbf{y}^{(n,1)} = \begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \underline{\mathbf{w}}_p^{(n)}$$

$\hookrightarrow 2\mathtt{M}\log_2 \mathtt{M} + 4\frac{\mathtt{ML_{FB}}}{\mathtt{P}} + 8\frac{\mathtt{L_{FB}}}{\mathtt{P}} - \mathtt{M} - 2$

$$\mathbf{d}^{(n)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_n \end{bmatrix} \begin{matrix} \updownarrow P-1+\sigma \\ \updownarrow L \end{matrix}, \quad \mathbf{d}_n = \begin{bmatrix} d[nL+1] \\ \vdots \\ d[(n+1)L] \end{bmatrix} \begin{matrix} \uparrow \\ \downarrow \end{matrix} L$$

$$\mathbf{e}^{(n,1)} = \mathbf{d}^{(n)} - \mathbf{y}^{(n,1)}$$

$\hookrightarrow \mathtt{L}$

$$\mathbf{c}^{(n,1)} = \mathbf{e}^{(n,1)}$$

$$\mathbf{e}_n^{(1)} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{e}^{(n,1)}$$

for $r = 2$ to $R$ do {

$$\mathbf{e}_n^{(r)} = \hat{\mathbf{B}}_n \mathbf{e}_n^{(r-1)}$$

$\hookrightarrow (\mathtt{R}-1)(2\mathtt{L}^2 - \mathtt{L})$

$$\mathbf{c}^{(n,r)} = \mathbf{c}^{(n,r-1)} + \begin{bmatrix} \mathbf{0} \\ \mathbf{e}_n^{(r)} \end{bmatrix}$$

$\hookrightarrow (\mathtt{R}-1)\mathtt{L}$

}

$$\underline{\mathbf{w}}_p^{(n+1)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n)} + \mu_n \mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{L-1+\sigma} \end{bmatrix} \mathbf{F}^{-1} \underline{\mathbf{X}}_p^{(n)\,*} \mathbf{F}\mathbf{c}^{(n,R)}$$

$\hookrightarrow (2\mathtt{M} + 4\frac{\mathtt{ML_{FB}}}{\mathtt{P}})\log_2 \mathtt{M} - 4\mathtt{M} + 6 + \mathtt{L} + 14\frac{\mathtt{L_{FB}}}{\mathtt{P}}$

}

$$\begin{array}{rcl@{\qquad}rcl} p & = & 0 : \frac{L_{FB}}{P} - 1 & M & = & P + L - 1 + \sigma \\ \mathbf{F}(p,q) & = & e^{-\frac{j2\pi pq}{M}} & P & = & L \\ \sigma & \geqslant & 0 & & & \end{array}$$

Table 9.4: Fast unnormalized constrained PBFDRAP : $P = L$

## 9.2 Computational cost

In this section the PBFDRAP algorithm is analyzed from a computational complexity point of view. Both for the unconstrained and the constrained updating the "standard" (non–fast) implementation (table 8.1) is compared with fast versions of the algorithm.

### 9.2.1 Unconstrained PBFDRAP

The equivalent number of real operations (for a definition, see section 2.4) to process a block of $L$ samples using a "standard" (non–fast) implementation of the unnormalized unconstrained PBFDRAP (table 8.1), is

$$2M\log_2 M(1+2R) - 4M(R+1) + 6(1+R) + RL + 16\frac{RL_{FB}}{P} + 8\frac{RML_{FB}}{P}. \quad (9.23)$$

It is assumed that $P$ is divisible by $L$, that $M$ is a power of 2 and that $x$ and $d$ and the unknown system $w$ are real–valued.

The cost for processing a block of $L$ samples with the fast unnormalized unconstrained PBFDRAP for which $P = L$ (table 9.3) is

$$2M\log_2 M(1+2R) - M(3R+\frac{1}{2}) + (13+8R) + 2RL - L + 16\frac{L_{FB}}{L} + 8\frac{ML_{FB}}{L}. \quad (9.24)$$

**Example 9.1** Cost estimates for the standard PBFDRAP and the fast PBFDRAP algorithm are presented in figure 9.1. For a realistic comparison the following parameters were chosen : $M = 256$, $P = L = 128$, $L_{FB} = 1024$. It appears that the fast PBFDRAP outperforms the standard PBFDRAP, except for $R = 1$, for which it is slightly more expensive. △

**Experiment 9.1** Some MATLAB® test were performed based on the same assumptions ($M = 256$, $P = L = 128$, $L_{FB} = 1024$). The executions times (in seconds) were measured for a data set of 100000 samples and are presented in table 9.5. Both the standard and the fast implementation give the same output, the fast algorithm being significantly cheaper. Iterating twice with the standard approach for instance is as expensive as iterating 4 times with the fast algorithm. ▽

### 9.2.2 Constrained PBFDRAP

For the constrained PBFDRAP the standard implementation presented in table 8.1 and the fast algorithm of table 9.4 seem most appropriate and typically lead to the smallest implementation cost. The equivalent number of real operations needed to

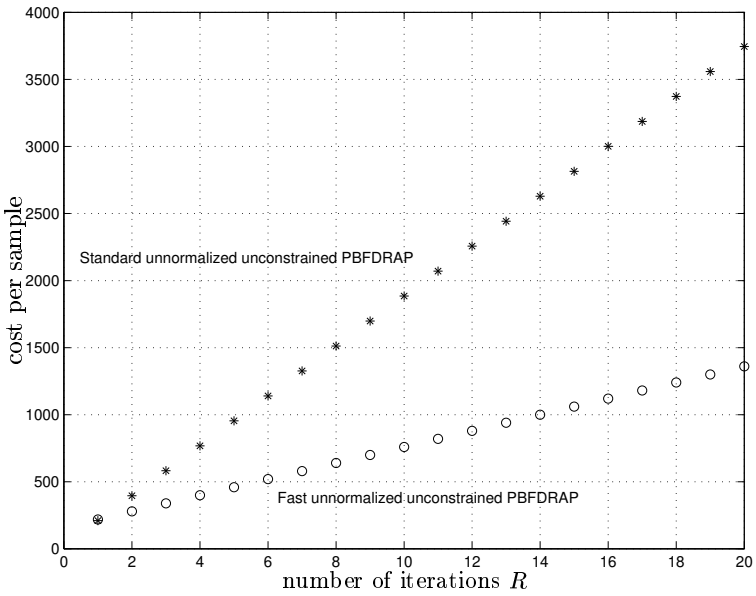


Figure 9.1: Cost standard unnormalized unconstrained PBFDRAP (table 8.1) vs. fast unnormalized unconstrained PBFDRAP (table 9.3) for a realistic parameter setting ($M = 256$, $P = L = 128$, $L_{FB} = 1024$)

| | standard PBFDRAP (table 8.1) | fast PBFDRAP (table 9.3) |
|---|---|---|
| $R = 1$ | 6.26 | 6.75 |
| $R = 2$ | 11.78 | 8.63 |
| $R = 3$ | 17.34 | 10.27 |
| $R = 4$ | 22.89 | 11.86 |
| $R = 5$ | 28.49 | 13.57 |
| $R = 10$ | 55.93 | 22.28 |

Table 9.5: The standard unconstrained unnormalized PBFDRAP is compared with a fast version of the algorithm. Some MATLAB® test were performed. The executions times (in seconds) were measured for a data set of 100000 samples, $M = 256$, $P = L = 128$, $L_{FB} = 1024$.

process a block of $L$ samples with the unnormalized constrained PBFDRAP using

the "standard" (non–fast) algorithm of table 8.1 is

$$2M(1+2R+2\frac{RL_{FB}}{P})\log_2 M - 4M(R+1)+6(1+R)+RL+22\frac{RL_{FB}}{P}+4\frac{RML_{FB}}{P}. \tag{9.25}$$

Also in this case it is assumed that $P$ is divisible by $L$, that $M$ is a power of 2 and that $x$ and $d$ and the unknown system $w$ are real–valued.

The equivalent formula for the fast algorithm of table 9.4 is ($P = L$)

$$2M(3+2\frac{L_{FB}}{L})\log_2 M - 9M + 10 + L^3 + 2RL^2 + 4L + 22\frac{L_{FB}}{L} + 4\frac{ML_{FB}}{L}. \tag{9.26}$$

**Example 9.2** In table 9.6 the equivalent number of real operations to process a block of data using the algorithms of table 8.1 and 9.4 (formula 9.25 and 9.26) are given for a set of realistic parameter settings. △

| $R$ | $L=P$ | $M$ | $L_{FB}$ | standard PBFDRAP (table 8.1) Eq. 9.25 | fast PBFDRAP (table 9.4) Eq. 9.26 |
|---|---|---|---|---|---|
| 1 | 16 | 32 | 128 | 7052 | 11674 |
| 2 | 16 | 32 | 128 | 13906 | 12186 |
| 1 | 16 | 32 | 256 | 13372 | 17994 |
| 2 | 16 | 32 | 256 | 26546 | 18506 |
| 1 | 32 | 64 | 256 | 16348 | 51194 |
| 3 | 32 | 64 | 256 | 48008 | 55290 |
| 1 | 64 | 128 | 512 | 37372 | 307770 |
| 10 | 64 | 128 | 512 | 362146 | 381498 |
| 1 | 128 | 256 | 512 | 47332 | 2177378 |
| 10 | 128 | 256 | 512 | 445618 | 2472290 |
| 1 | 128 | 256 | 1024 | 84284 | 2214330 |
| 10 | 128 | 256 | 1024 | 815138 | 2509242 |

Table 9.6: The equivalent number of real operations needed to process a block of data using the unnormalized constrained algorithms of table 8.1 and 9.4 (formula 9.25 and 9.26) are shown for some realistic parameter settings.

**Example 9.3** In figure 9.2 the number of iteration steps $R$ is plotted for which the standard implementation of the unnormalized constrained PBFDRAP (table 8.1) is as expensive as the fast implementation of table 9.4. Each curve corresponds to a different block length $L$. For a parameter set $(L_{FB}, R)$ above the curve, the fast

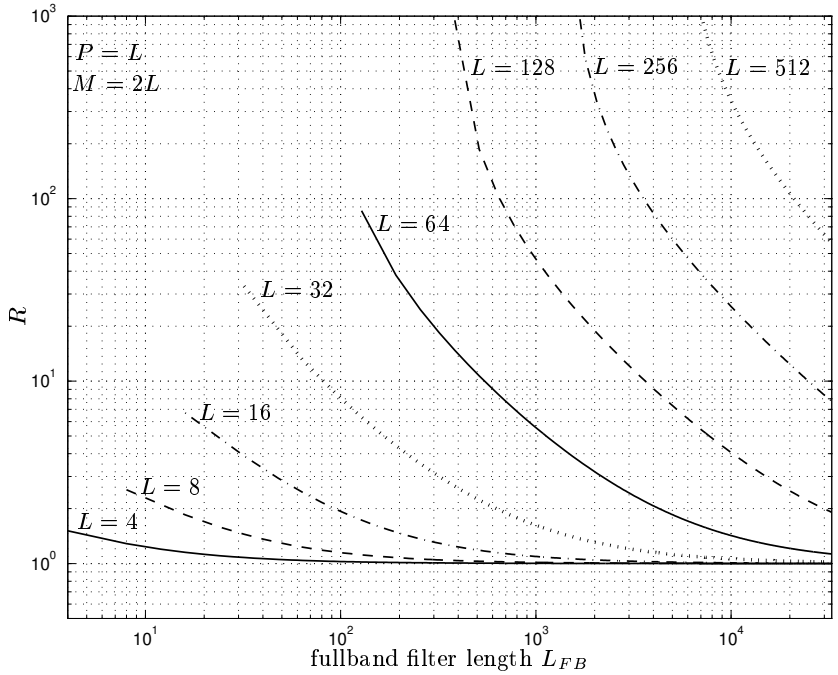


Figure 9.2: Number of iteration steps $R$ for which the standard implementation of the unnormalized constrained PBFDRAP (table 8.1) is as expensive as the fast implementation of table 9.4. For a parameter set $(L_{FB}, R)$ above the curve, the fast algorithm leads to the lowest cost.

algorithm leads to the lowest cost. Below the curve the standard algorithm of table 8.1 is most appropriate. △

It appears that Eq. 9.26 is more or less insensitive to changes in $R$, whereas Eq. 9.25 strongly depends on $R$. For small values of $L$, and $L_{FB}$ and $R$ sufficiently large, the algorithm of table 9.4 leads to the smallest cost.

### 9.2.3 Unnormalized constrained PBFDRAP versus PRA

Infinitely iterating the unnormalized constrained PBFDRAP leads to the PRA algorithm (see section 8.2.3). The equations defining the PRA algorithm are repeated in table 9.7. For the weight updating of the PRA the inverse of a hermitian matrix $\mathbf{M}_n = \mathbf{X}_n^T \mathbf{X}_n^*$ has to be computed (see Eq. 2.55). For stability reasons matrix $\mathbf{M}_n$ is regularized, i.e. the inverse of $\mathbf{X}_n^T \mathbf{X}_n^* + \delta \mathbf{I}_L$ is applied i.o. $\mathbf{X}_n^T \mathbf{X}_n^*$, with $\delta$ a small positive real number.

As the data is shifted block by block efficient QR–based updating of $\mathbf{X}_n^T\mathbf{X}_n^*$ is not possible for $L > 1$. However, if $L_{FB}$ is a multiple of $L$,

$$\mathbf{X}_n^l = \begin{bmatrix} x[nL - lL + 1] & \dots & x[nL + L - lL] \\ \vdots & \ddots & \vdots \\ x[nL - L + 2 - lL] & \dots & x[nL - lL + 1] \end{bmatrix}, \quad l = 0 : \frac{L_{FB}}{L} \tag{9.27}$$

can be defined and $\mathbf{X}_n^T\mathbf{X}_n^*$ is updated as

$$\mathbf{M}_n = \mathbf{X}_n^T\mathbf{X}_n^* \tag{9.28}$$

$$= \begin{bmatrix} \mathbf{X}_n^{0\,T} & \mathbf{X}_n^{1\,T} & \dots & \left(\mathbf{X}_n^{L_{FB}/L-1}\right)^T \end{bmatrix} \begin{bmatrix} \mathbf{X}_n^{0\,*} \\ \mathbf{X}_n^{1\,*} \\ \vdots \\ \left(\mathbf{X}_n^{L_{FB}/L-1}\right)^* \end{bmatrix} \tag{9.29}$$

$$= \mathbf{M}_{n-1} + \mathbf{X}_n^{0\,T}\mathbf{X}_n^{0\,*} - \left(\mathbf{X}_n^{L_{FB}/L}\right)^T \left(\mathbf{X}_n^{L_{FB}/L}\right)^*. \tag{9.30}$$

This reduces the cost substantially if $L_{FB}/L$ is large. As $\mathbf{M}_n$ is hermitian and

$$\left(\mathbf{X}_n^{L_{FB}/L}\right)^T \left(\mathbf{X}_n^{L_{FB}/L}\right)^*$$

can be recuperated from a previous iteration, the cost for updating $\mathbf{M}_n$ amounts to $L^3 + \frac{3}{2}L^2 + \frac{L}{2}$ if it is assumed that real signals are applied. Furthermore, it appears that $\mathbf{p}_n = (\mathbf{M}_n + \delta\mathbf{I}_L)^{-1}\mathbf{e}_n$ (see table 9.7) can be computed efficiently by solving the linear set of equations $(\mathbf{M}_n + \delta\mathbf{I}_L)\mathbf{p}_n = \mathbf{e}_n$ using Gauss elimination (cost $\frac{2L^3}{3}$) and back substitution (cost $L^2$) [73].

The equivalent number of real operations needed to process a block of $L$ real samples with the PRA (see table 9.7) is then equal to

$$4L_{FB}L + \frac{5L^3}{3} + \frac{5L^2}{2} + \frac{5L}{2}. \tag{9.31}$$

$R_{crit}$, the value for $R$ for which the unnormalized constrained PBFDRAP is as expensive as PRA, can now be computed. By combining Eq. 9.25 and 9.31 (PRA vs. non–fast PBFDRAP), it is found that

$$R_{crit1} = \frac{4L_{FB}L + \frac{5}{3}L^3 + \frac{5}{2}L^2 + \frac{5}{2}L - 2M\log_2 M + 4M - 6}{4M\log_2 M(1 + \frac{L_{FB}}{P}) - 4M + 6 + L + 22\frac{L_{FB}}{P} + 4\frac{ML_{FB}}{P}}. \tag{9.32}$$

In most practical applications, $M = 2L = 2P$. Hence,

$$R_{crit1} = \frac{4L_{FB}L + \frac{5}{3}L^3 + \frac{5}{2}L^2 + \frac{5}{2}L - 4L\log_2 L + 4L - 6}{8L\log_2 L(1 + \frac{L_{FB}}{L}) + 6 + L + 22\frac{L_{FB}}{L} + 16L_{FB}}. \tag{9.33}$$

for each block of $L$ samples do $\{$

$$\mathbf{X}_n^l \overset{\forall l}{=} \begin{bmatrix} x[nL-lL+1] & \dots & x[nL+L-lL] \\ \vdots & \ddots & \vdots \\ x[nL-L+2-lL] & \dots & x[nL-lL+1] \end{bmatrix}$$

$$\mathbf{X}_n = \begin{bmatrix} \mathbf{X}_n^0 \\ \mathbf{X}_n^1 \\ \vdots \\ \mathbf{X}_n^{(L_{FB}/L-1)} \end{bmatrix}$$

$$\mathbf{d}_n = \begin{bmatrix} d[nL+1] \\ \vdots \\ d[(n+1)L] \end{bmatrix}$$

$$\mathbf{y}_n = \mathbf{X}_n^T \hat{\mathbf{w}}_n \qquad \hookrightarrow 2\mathrm{L}\mathrm{L}_{\mathrm{FB}} - \mathrm{L}$$

$$\mathbf{e}_n = \mathbf{d}_n - \mathbf{y}_n \qquad \hookrightarrow \mathrm{L}$$

$$\mathbf{M}_n = \mathbf{M}_{n-1} + {\mathbf{X}_n^0}^T {\mathbf{X}_n^0}^* - \left(\mathbf{X}_n^{L_{FB}/L}\right)^T \left(\mathbf{X}_n^{L_{FB}/L}\right)^* \qquad \hookrightarrow \mathrm{L}^3 + \tfrac{3}{2}\mathrm{L}^2 + \tfrac{\mathrm{L}}{2}$$

$$\mathbf{p}_n = \left(\mathbf{M}_n + \delta \mathbf{I}_L\right)^{-1} \mathbf{e}_n \qquad \hookrightarrow \tfrac{2}{3}\mathrm{L}^3 + \mathrm{L}^2 + \mathrm{L}$$

$$\hat{\mathbf{w}}_{n+1} = \hat{\mathbf{w}}_n + \mu_n \mathbf{X}_n^* \mathbf{p}_n \qquad \hookrightarrow 2\mathrm{L}\mathrm{L}_{\mathrm{FB}} + \mathrm{L}$$

$\}$

$$l \quad = \quad 0 : \frac{L_{FB}}{L}$$

Table 9.7: Partial Rank Algorithm (PRA)

By combining Eq. 9.26 and 9.31 (PRA vs. fast PBFDRAP), one obtains

$$R_{crit2} = \frac{4L_{FB}L + \frac{2}{3}L^3 + \frac{5}{2}L^2 - \frac{3}{2}L - 2M(3 + 2\frac{L_{FB}}{L})\log_2 M}{2L^2} + \frac{9M - 10 - 22\frac{L_{FB}}{L} - 4\frac{ML_{FB}}{L}}{2L^2}. \quad (9.34)$$

**Example 9.4** In figure 9.3, $R_{crit} = \max(R_{crit1}, R_{crit2})$ is plotted for $\frac{L_{FB}}{L}$=1, 2, 5, 10 and 20. Block length $L$ varies between 2 and 1024 and $M = 2L = 2P$. $\triangle$

It can be concluded that for large block lengths, the iterated PBFDRAP is a cheaper alternative to the PRA.

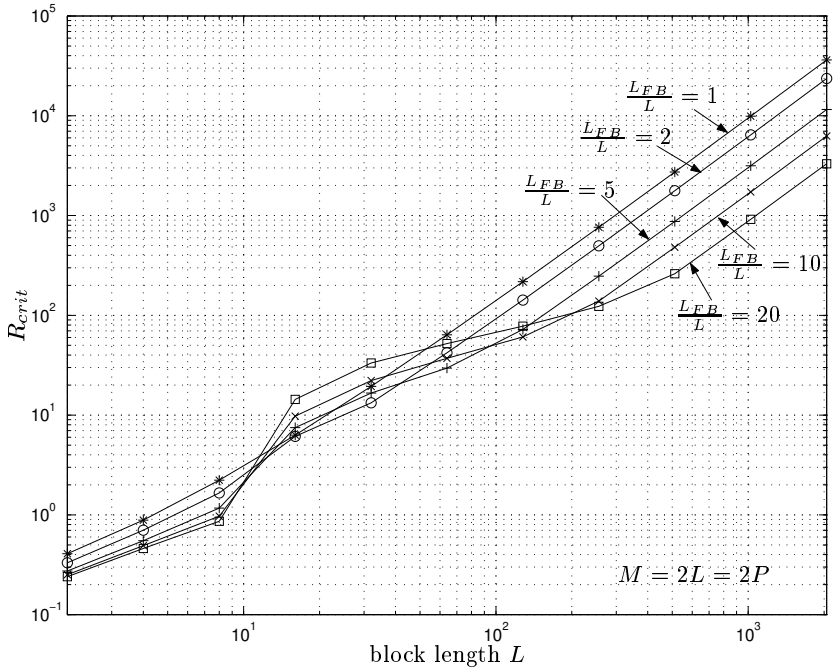


Figure 9.3: The number of iteration steps $R_{crit}$ for which the unnormalized constrained PBFDRAP is as expensive as the PRA, is plotted for different parameters settings.

## 9.3 Conclusions

In this chapter the fast PBFDRAP was discussed. If the number of iteration steps is large the standard (non–fast) PBFDRAP as presented in chapter 8 requires a considerable amount of processing power.

Several fast implementations were derived in section 9.1 leading to a significant reduction in the computational complexity. The different fast versions of the PBFDRAP are presented in tables 9.1–9.4.

Section 9.2 focussed on the complexity gain. The fast PBFDRAP was compared with the standard implementation of the algorithm presented in chapter 8 and table 8.1. Depending on the mode in which the algorithm operates and the choice of the parameters, a significant cost reduction can be obtained.

The unnormalized and globally normalized constrained PBFDRAP were compared with the PRA algorithm from a computational complexity point of view. It appears that for large block lengths, the PBFDRAP is a cheaper alternative to the PRA.

# Part IV

# Acoustic Echo Cancellation, Implementation and Experiments

In the final part it is shown that the different adaptive filters that are developed in the preceding chapters can be successfully applied to a real signal enhancement setup.

Practical problems are encountered whenever adaptive filtering algorithms are applied to a real signal enhancement setup. Apart from the adaptive filter, "control" software has to be included for proper and robust operation. This will be illustrated in **chapter 10** taking the acoustic echo cancellation problem as an example. Different adaptive algorithms will be compared under realistic echo cancellation conditions and a real–time DSP implementation of an acoustic echo cancellation system will be discussed.

# Chapter 10

# Acoustic Echo Cancellation, Implementation & Experiments

In the previous parts of the thesis different adaptive filtering and filter bank schemes were discussed, mainly putting emphasis on the design, as well as on algorithmic and complexity aspects. The presented techniques can be employed in many fields and applications, some of which were briefly addressed in chapter 1 and 2. In this chapter we focus on one specific application, namely acoustic echo cancellation, and illustrate how the different algorithms can be applied and embedded in a real–world engineering application.

It was motivated in chapter 2 that acoustic echo cancellation is an example of adaptive identification. The adaptive filters discussed in this thesis are therefore readily applicable to the echo cancellation problem. However, a real–life echo canceller has to cope with some practical and implementation related issues such as time–varying acoustics, non–linearities and double talk, making it still a long way from the algorithmic concept towards a consumer electronics product that can actually be sold and that is appreciated by its users.

In section 10.1 it is explained that apart from the adaptive filter a "control" algorithm has to be included, which basically monitors the adaptation stepsize $\mu$. In this way proper operation is ensured during double talk. Further, it is briefly explained how the short–time energy of the different signals can be computed. These energies are used for double–talk detection and stepsize normalization. Finally, a post–processing unit is typically added to the echo canceller to remove the remaining residual errors.

In section 10.2 two experiments are discussed in which some of the algorithms of the previous chapters are applied to a simulated acoustic echo cancellation setup for comparison.

A real–time acoustic echo canceller was programmed on a DSP system. Some observations and experiments related to the DSP implementation are considered in section 10.3.

The conclusions to this chapter are presented in section 10.4.

## 10.1 Robust operation and control

Consider the echo cancellation problem, which was introduced in section 1.4.1. In present–day echo cancellation systems adaptive filtering techniques are called for to cancel the echo : the adaptive filter is self–learning and can be initialized with zero knowledge. Furthermore, due to the time variations of the acoustic path the echo cancellation problem is far from trivial, as was explained in section 1.3.2. Thanks to the adaptivity these time variations can be tracked.

For proper operation the stepsize(s) controlling the adaptation speed of the adaptive filter(s) should be carefully monitored. The input to the echo cancellation system will typically be a speech signal having a continuously varying short–time energy, as was explained in section 1.3.1, hence applying stepsize normalization is mandatory. Secondly, during near–end signal activity in so–called double–talk situations, i.e. when both the far–end as the near–end speaker (see figure 1.3) are active, the adaptation must be frozen. The stepsizes should be immediately set to zero or kept very small in order to avoid system mismatch.

To make the echo canceller robust against double talk the adaptive filtering core has to be supplemented with intelligent control software. Different types of control parameters are therefore included. Intensive testing and tuning should eventually lead to a cheap control system having as much of the desired properties as possible. Some control issues will be discussed in this section and a few solutions will be given. More information can be found in [128]. In this paper the design of a complete PBFDAF echo canceller is presented and a lot of attention is paid to the robustness issue.

A more elaborated scheme replacing figure 1.5 is shown in figure 10.1. The A/D and D/A units represent analog–to–digital and digital–to–analog converters respectively and S/P and P/S stand for serial–to–parallel and parallel–to–serial converters. It is common to process the data in blocks, also called frames in this context. Frame–based processing facilitates the supervision and control as "looking ahead" leads to a more robust double–talk detection. Furthermore, block–based adaptive filters can be called for, which have desirable properties as was shown in section 2.3.2. The

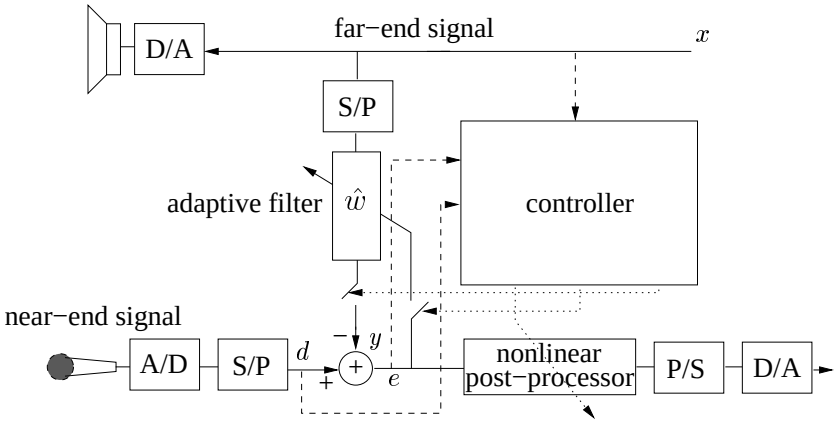


Figure 10.1: Adaptive acoustic echo canceller with control : layout

acceptable frame length is typically in the order of tens of milliseconds and hence corresponds to the length of elementary speech blocks.

### 10.1.1 Short–time energy

In acoustic echo cancellation short–time energy estimates are needed for stepsize normalization in the adaptive filtering procedure on the one hand (see e.g. Eq. 2.28, section 6.1.3, ...) and for far–end activity and double–talk detection in the "control" algorithm on the other hand. For far–end and double–talk detection basically four signals of interest are encountered : the far–end signal $x$, the near–end signal $d$, the reconstructed far–end echo $y$ and the error output $e$. The short–time energy of each of these four signals $E_x$, $E_d$, $E_y$ and $E_e$ can then be computed. For stepsize normalization only the energy $E_x$ of the far–end signal buffer has to be estimated.

For complexity reasons energy estimates $E_x$ are typically not recomputed from scratch (i.e. as $E_x = \mathbf{x}^T\mathbf{x}$) but are obtained by up- and downdating. If for instance a sample–based update is required, $E_x(k)$ can be computed as[1]

$$E_x(k) = E_x(k-1) + x[k]^2 - x[k-N]^2 \tag{10.1}$$

in which $N$ is the time horizon over which the energy is computed. The cost for this up– and downdate amounts to 4 real instructions per sample instead of $2N-1$ op. if computed from scratch.

Many echo cancellation algorithms however are frame–based (figure 10.1), i.e. the data is processed in block of, say, $L$ samples. As a consequence block–based rather

[1] As this chapter deals with acoustic echo cancellation all signals and the unknown system $w$ are assumed to be real–valued.

than sample–based energy estimates are required. If $N$ is a multiple of $L$ for instance, $E_x(n)$ can be computed as

$$E_x(n) = E_x(n-1) + e_n - e_{n-\frac{N}{L}} \tag{10.2}$$

where $e_n = \mathbf{x}_n^T \mathbf{x}_n$ is the energy of the far–end buffer during frame $n$. Equation 10.2 reduces to 10.1 for $L = 1$ and $e_n = x[n]^2$.

Formula 10.1 and 10.2 are not so robust however, as round–off errors could dominate $E_x$ after a while. An appealing alternative, having the same low complexity, uses a first–order IIR update, leading to

$$E_x(n) = (1-\beta)E_x(n-1) + \beta e_n. \tag{10.3}$$

Round–off errors fade away as long as $0 < \beta < 2$, for $\beta \in \mathbb{R}$. $E_x$ then is a smoothed estimate of the far–end energy.

For the adaptive weight update equation (e.g. Eq. 2.28) the inverse of $E_x$ is required. Inversion is an expensive operation on a DSP[2]. For small values of $\beta$ however the inverse of $E_x$ may be updated instead of $E_x$ itself, leading to [36]

$$E_x(n)^{-1} \approx (1+\beta)E_x(n-1)^{-1} - \beta e_n E_x(n-1)^{-2}. \tag{10.4}$$

### 10.1.2 Far–end activity detection

Once the energies have been computed, some control decisions can be made. For instance, if the far–end signal energy $E_x$ is lower than a certain threshold $\epsilon$, no far–end stimulus is assumed to be present. The adaptation process is frozen and the near–end signal $d$ is passed to the output without correction ($e = d$), which is symbolized by a controlled switch in the $y$–channel in figure 10.1. By continuing the adaptation the adaptive filter(s) could drift away from the acoustic path replica due to the activity of background noise at the near–end side.

$E_x$ might be recuperated from the normalization part of the adaptation algorithm as stepsize normalization is mandatory and hence $E_x$ is computed anyway. However, the horizon over which the energy is calculated does not always match the frame length of the control algorithm.

More robust far–end activity detectors than the simple threshold–based energy method presented here can be applied. More information on voice activity detection and related topics can be found in [30] [32] [35] [72] [84] [88] [98] [127] [138] [140] [143] [144] [159] [183].

[2] An inversion takes 7 machine cycles on a TMS320C4x floating–point DSP for instance.

### 10.1.3 Double–talk detection

As soon as a local near–end speaker is active or in the case of double talk, for which both speakers are active, the adaptation process must be frozen. This is indicated in figure 10.1 by a controlled switch in the adaptation arrow. Otherwise, the adaptive filter is driven away from its Wiener solution by the local non–stationary source. The adaptive coefficients would be moving around, agitated by the local source, resulting in an annoying echo–like disturbance. Near–end speech detection is thus crucial for a proper operation of the echo canceller.

Block–based algorithms can look into the near future and by doing so, (hopefully) detect near–end speaker activity as soon as possible. In the case of double talk this is a far from easy task as it comes down to the detection of speech in speech. The onsets of speech are difficult to detect and to separate from a non–stationary far–end signal. Furthermore, sudden changes in the acoustic path increase the error energy and hence can be confused with an upcoming near–end source.

A double–talk detector that is too sensitive will generate a lot of false alarms. The adaptation is regularly stopped, so the overall convergence speed will be low. On the other hand, when the detector is critically tuned, even a slightly too late detection of the near–end speech onset could lead to a significant misfit of the adaptive filter.

The echo path is supposed to attenuate the far–end signal level. Therefore, a comparison between the far–end and near–end instantaneous energy, $E_x$ and $E_d$, gives an idea about near–end source activity. A threshold $0 < \tau < 1$ can then be defined. If now

$$E_d > \tau E_x, \tag{10.5}$$

double talk is detected. Fine tuning threshold $\tau$ is crucial however. This method is not so robust. Another more advanced measure could be [128] :

$$\rho = \frac{E_x E_e}{E_x^2 + E_y^2}. \tag{10.6}$$

$\rho$ is smaller than 1 in the absence of double talk. When a local correspondent starts to speak, $\rho$ will rise.

The problem so far is that the adaptation is switched either off or on. A sliding stepsize $\mu$ may be more appropriate. Stepsize $\mu$ can then vary between 0 (near–end activity) and $\mu_{max}$ (only far–end activity) based on the probability that the near–end source is active.

In [82] a correlation based method was proposed. In the absence of near–end speech the loudspeaker and microphone signal are highly correlated. An estimate of the attenuation $\alpha = \mathcal{E}\{\frac{E_e}{E_x}\}$ can then be updated. If the short–time energy $E_e$ of the adaptive filter output is larger than expected ($E_e > \alpha E_x$), double talk occurs and the adaptation must be stopped. By comparing the short–time and long–time energies of both $x$ and $e$, the activity at the far–end side as well as the level of the

near–end background noise can be estimated. If no significant near–end activity is detected the adaptation stepsize is computed based on this near–end background noise level and the short–time error energy.

More advanced double–talk detectors are based on spectral content. Only that part of the frequency spectrum that contains important speech information is then analyzed. Twin–structures were also proposed and appeared to be more robust than classical schemes [133]. Other double–talk detection algorithms and a comparison study can be found in [8] [16] [173].

### 10.1.4 Post–processing

In practical applications a residual error $\frac{E_e}{E_d}$ is observed at the output of the adaptive filter. This residual error is typically about -30 dB and depends on both the algorithmic and the hardware quality of the setup. Reasons for the existence of the residual error are :

- the slow convergence of the adaptive algorithm
- the finite length of the adaptive filter (undermodelling error)
- time–variations in the environment (resulting in a lag error)
- local background noise (leading to an excess mean–squared error)
- wrong control decisions (double–talk misdetection)
- nonlinear distortion (e.g. due to loudspeaker nonlinearities [9])

A nonlinear enhancement module is typically inserted at the output of the echo canceller to reduce the residual error. A very simple device removing the residual error is the *center clipper*. It is a threshold device for which

$$x_{out} = \begin{cases} x_{in} + \delta & x_{in} < -\delta \\ 0 & -\delta \leqslant x_{in} \leqslant \delta \\ x_{in} - \delta & x_{in} > \delta \end{cases} \tag{10.7}$$

Threshold $\delta$ is a positive constant typically of the same order of magnitude as the mean background noise level. Remark that the center clipper also clips the near–end signal. Too high thresholds $\delta$ would result in an unacceptable near–end signal distortion. More advanced post–processing units are then applied and rely on *spectral subtraction* based noise attenuation or *Wiener filtering* for instance [11] [155] [182].

In commercial echo cancellers the post–processing can often be considered as the magic "black box" making the difference between a working algorithm that simply reduces the echo and a vendible product that is appreciated by its users and that

outperforms its competitors. Advanced post–processing modules not only reduce the residual error, but also try to make the output of the echo canceller more natural and pleasant to listen to.

Remark that a residual error will appear in practice, whatever adaptive filtering algorithm is being used. This fact pleads in favor of subband adaptive filtering, discussed in chapter 5, one of the disadvantages of which was the residual under-modelling error. As long as this residual error stays below, say, -30 dB, it will be masked by the residual error coming from the imperfections mentioned before and hence can be removed by the post–processing as well. More crucial are the initial convergence speed and the tracking capabilities of the adaptive filter.

## 10.2 Design examples and off–line comparison

In this section realistic tests will be performed with some of the adaptive algorithms discussed in the previous chapters. The different adaptive filters are compared and tested in an acoustic echo cancellation setup. For the tests a simulated echo environment was used.

**Experiment 10.1** In a first experiment the PBFDAF algorithm (chapter 6) is considered. The recording room is a strongly reverberating $4 \times 5 \times 3$ m$^3$ rectangular office. Loudspeaker and microphone are close to each other (20 cm). A simulated acoustic impulse response was generated using the image method presented in [4] and is 250 ms long. The sampling frequency is 10 kHz.

First, the *subband–normalized unconstrained PBFDAF* was used to identify the stationary acoustic path with a white noise reference input. The block length $L$ was taken equal to the filter partition length $P$ and is 64. Hence, the input–output delay of the algorithm is 12.7 ms. The FFT–size $M$ was 128. The fullband equivalent filter length $L_{FB}$ was taken equal to 2432. The computational load is estimated based on the results of section 6.4 and is 7.03 MFlops. The stepsize was optimized to obtain the fastest convergence. The time evolution of the short–time energy of the error output is shown in figure 10.2.

A *subband–normalized constrained PBFDAF algorithm* was also applied with the same parameters, i.e. $L = P = \frac{M}{2} = 64$ and $L_{FB} = 2432$, representing a cost of 25.3 MFlops. It appears from figure 10.2 that thanks to the constrained updating superior convergence behavior is obtained. $\bigtriangledown$

**Experiment 10.2** In a second experiment five adaptive algorithms are compared. Two different input signals were applied to the adaptive filters :

1. Gaussian distributed white noise

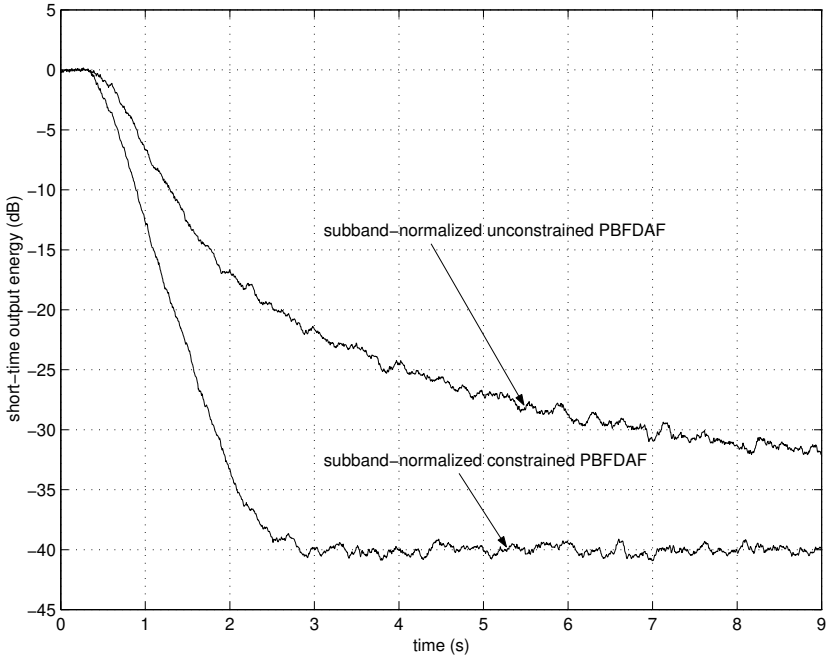


Figure 10.2: Short–time output energy for the PBFDAF

2. Gaussian distributed colored noise, which was obtained by filtering Gaussian white noise with a first order lowpass IIR filter, having a pole at 0.9, i.e.

$$H(z) = \frac{1}{1 - 0.9z^{-1}}.$$

Further, the algorithmic and environmental parameters are chosen so to simulate a realistic acoustic echo cancellation scenario. The sampling frequency is assumed to be 8 kHz. An artificial echo path of 250 ms was generated using the image method of [4]. The loudspeaker–to–microphone distance is 10 cm and the volume of the recording room is $3 \times 3.2 \times 2.6\ \mathrm{m}^3$. The reflection coefficients of the walls are chosen equal to 0.85. This corresponds to a $\mathrm{RT}_{60}$ reverberation time of 278 ms [60]. There are no active local near–end sources and the acoustics are stationary.

Five adaptive algorithms are compared :

1. fullband normalized LMS (section 2.3.1) [NLMS]

2. standard subband adaptive filter with NLMS adaptation in the subbands (section 5.1) [SUB1]

| parameter | symbol |
|---|---|
| equivalent fullband adaptive filter length | $L_{FB}$ |
| subband adaptive filter length | $L_{SB}$ |
| number of subbands, FFT size | $M$ |
| downsampling factor | $N$ |
| block length | $L$ |
| filter partition length | $P$ |
| stepsize relative to maximum stepsize | $\mu/\mu_{\max}$ |
| initial filter weights | $\hat{\mathbf{w}}_0$ |
| input–output delay in ms | $\delta_{\text{I-O}}$ |
| estimated cost in MFlops | $c_e$ |
| simulated cost in MFlops using MATLAB® | $c_s$ |
| filter bank prototype filter length | $L_f$ |
| number of anti–causal filter taps | $L_{ac}$ |
| number of extra causal filter taps | $L_c$ |
| number of iteration steps | $R$ |

Table 10.1: Algorithmic parameters : definition

3. subband adaptive filter with NLMS adaptation in the subbands and extra (anti–)causal filter taps (section 5.4.2) [SUB2]
4. subband–normalized unconstrained PBFDAF (section 6.1.2) [PBFDAF]
5. subband–normalized unconstrained PBFDRAP (section 8.1.1) [PBFDRAP]

For the subband adaptive filters a 16–band nearly–perfect reconstruction filter bank set was designed. An up- and downdating mechanism as in Eq. 10.1 was applied to normalize each of the five adaptation algorithms. Finally, for the PBFDRAP the standard implementation of table 8.1 was applied.

The algorithmic parameters used in this experiment are defined in table 10.1. The actual values of the parameters that were applied to each of the adaptive filters can be found in table 10.2.

The error convergence curves for the setup with the white noise input are shown in figure 10.3. For the PBFDRAP both the a–priori and the a–posteriori error are shown. Recall from section 8.1.2 that for proper assessment of the quality of the PBFDRAP model approximation the a–priori error should be considered. It is observed that the NLMS algorithm shows the fastest convergence. Further, a significant residual undermodelling error remains in the case of the first subband adaptive filter SUB1.

The convergence curves for the colored noise input are shown in figure 10.4. Observe that the performance of the NLMS now drops considerably, while the other algo-

| | NLMS | SUB1 | SUB2 | PBFDAF | PBFDRAP |
|---|---|---|---|---|---|
| $L_{FB}$ | 1152 | 1152 | 1296 | 1152 | 1152 |
| $L_{SB}$ | – | 96 | 108 | 18 | 18 |
| $M$ | – | 16 | 16 | 128 | 128 |
| $N, L$ | – | 12 | 12 | 64 | 64 |
| $P$ | – | – | – | 64 | 64 |
| $\mu/\mu_{\max}$ | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 |
| $\hat{\mathbf{w}}_0$ | **0** | **0** | **0** | **0** | **0** |
| $\delta_{\text{I-O}}$ | 0.125 | 16.6 | 31.6 | 15.9 | 15.9 |
| $c_e$ | 36.9 | 8.32 | 9.28 | 3.02 | 5.76 |
| $c_s$ | 46.2 | 12.9 | 14.2 | 5.90 | 11.1 |
| $L_f$ | – | 110 | 110 | – | – |
| $L_{ac}$ | 0 | 0 | 10 | 0 | 0 |
| $L_c$ | 0 | 0 | 2 | 0 | 0 |
| $R$ | – | – | – | – | 2 |

Table 10.2: Parameter values for each of the five adaptive filters

rithms do not seem to suffer from the coloring of the input, thanks to the subband processing.

The same conclusions can be drawn from table 10.3, in which the ERLE after 5 seconds, the initial convergence speed and the "time–to–20dB–ERLE" are given. The initial convergence speed was computed based on the time the algorithm needs to go from 3 dB to 10 dB ERLE. For the PBFDRAP two numbers are given. The first corresponds to the a–priori error output, the second to the a–posteriori error output.

Taking into account the observations from figures 10.3 and 10.4 and table 10.3 it can be concluded that the NLMS algorithm is not a good candidate for acoustic echo cancellation as the performance for colored inputs (cf. speech) is inferior and the implementation cost is rather high. Taking all observations into account the PBFDAF algorithm is preferred for this kind of application. ▽

## 10.3 A real–time implementation of an acoustic echo canceller on DSP

A real–time acoustic echo cancellation system was programmed on DSP. It now serves as a demo setup in the ESAT speech laboratory of the KULeuven to illustrate the effectiveness of adaptive acoustic echo cancellation and the necessity of signal enhancement techniques in hands–free communication systems. The echo canceller

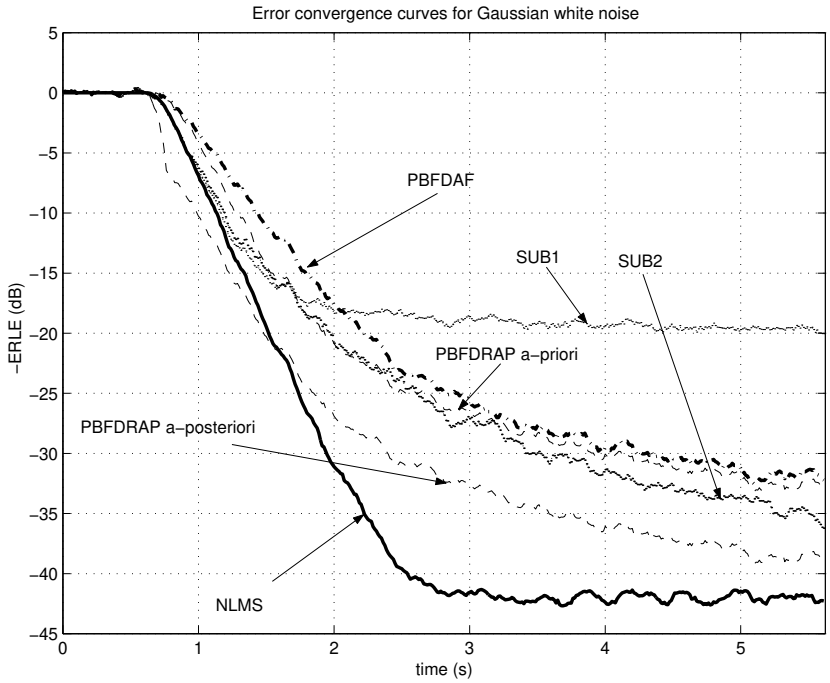


Figure 10.3: Error convergence curves for Gaussian white noise

basically consists of an adaptive filtering core and additional control software as in figure 10.1. Several adaptive filters can be plugged in for evaluation and comparison.

### 10.3.1 DSP equipment

The echo canceller was programmed on two DSP boards, which are placed in a VME–rack as shown in figure 10.5. The specifications of the DSP equipment are briefly described in table 10.4.

For this application only two of the seven available DSPs are used. A 25–MIPS TMS320C44[3]–processor clocked at 50 MHz is responsible for the data acquisition. The loudspeaker and microphone channel are first sampled at 16 kHz and are then digitally downsampled to 8 kHz to minimize the aliasing distortion. The input channels $x$ and $d$ are sent to a second processor, a 25–MIPS TMS320C40–DSP also clocked at 50 MHz, which does the echo cancellation. The output samples $e$ are transferred back to the first DSP and after digital upsampling, they are sent to a loudspeaker for evaluation.

[3] The TMS320C4x–family are standard floating–point DSPs from Texas Instruments.

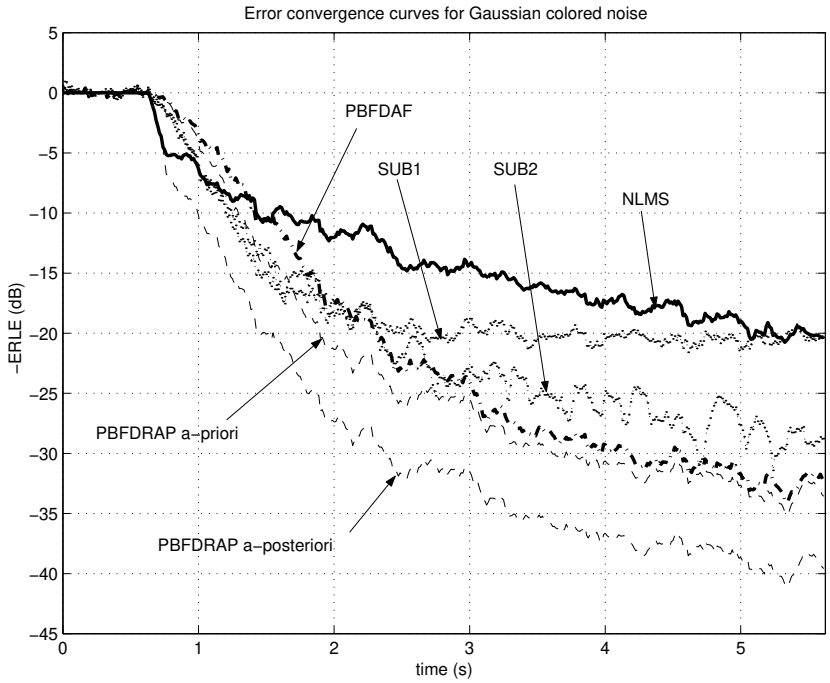


Figure 10.4: Error convergence curves for Gaussian colored noise

### 10.3.2 Software

Different adaptive algorithms were implemented on the DSP system, mainly programmed in assembler language. In this way some specific DSP operations such as circular addressing and parallel instructions are optimally used [153]. The different

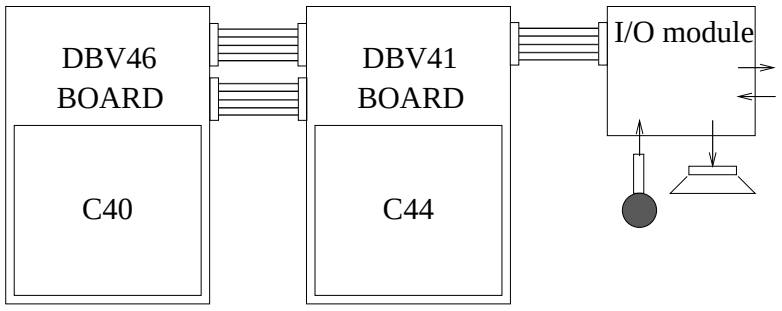


Figure 10.5: VME–based DSP system

| | NLMS | SUB1 | SUB2 | PBFDAF | PBFDRAP |
|---|---|---|---|---|---|
| white noise | 42.3 | 20.0 | 36.0 | 32.4 | 33.1/39.1 |
| colored noise | 20.6 | 20.7 | 29.4 | 37.5 | 38.1/44.2 |

Table 10.3.a: ERLE in dB after 5 seconds

| | NLMS | SUB1 | SUB2 | PBFDAF | PBFDRAP |
|---|---|---|---|---|---|
| white noise | 24.1 | 20.8 | 19.9 | 15.3 | 19.3/25.6 |
| colored noise | 9.75 | 19.2 | 17.7 | 16.6 | 19.9/24.4 |

Table 10.3.b: Initial convergence speed $\frac{\partial \text{ERLE}}{\partial t}$ in $\frac{\text{dB}}{\text{s}}$

| | NLMS | SUB1 | SUB2 | PBFDAF | PBFDRAP |
|---|---|---|---|---|---|
| white noise | 0.89 | 4.95 | 1.34 | 1.59 | 1.31/0.85 |
| colored noise | 4.43 | 2.04 | 1.68 | 1.69 | 1.27/0.83 |

Table 10.3.c: Time–to–20dB–ERLE in seconds

Table 10.3: Three performance indices for both the white and the colored noise input

echo cancellation algorithms were first developed in MATLAB® and C and were then programmed on the DSP system. For all algorithms there exist parallel versions in MATLAB®, C and on the DSPs, all giving consistent results up to within machine precision.

The control algorithm we used is based on [82] and was mainly programmed in C. Some of its features were already described in section 10.1.3. The adaptive filtering algorithms available on DSP are the NLMS algorithm (section 2.3.1), the subband–normalized unconstrained PBFDAF (section 6.1.2) and the unconstrained PBFDRAP (section 8.1.1).

### 10.3.3 Experiments

Some tests were carried out in the ESAT speech laboratory of the KULeuven in order to validate different sorts of algorithms under realistic conditions.

**Experiment 10.3** A loudspeaker (far–end signal $x$) and a microphone (near–end signal $d$) were placed 40 cm apart as shown in figure 10.6. In this first experiment there was no near–end speaker present in the recording room. The acoustic path was identified off–line and is shown in figure 10.7 for $f_s = 8$ kHz. The room was found to be moderately damped.

Then a white noise signal was sampled at 8 kHz and only the frequencies between 200 and 3700 Hz were retained using a bandpass filter. The noise signal was sent to

| board | # | modules | memory |
|---|---|---|---|
| SUN Sparc 4m @ 110 MHz | | | 16 Mb DRAM<br>2 Gb hard disk |
| LSI DBV41 | 1× | TMS320C44 @ 50 MHz | 4 Mb 0ws SRAM |
| | 4× | AM/D16SA–200 daughter modules 2 ch. in & 2 ch. out | $f_{s_{\max}}$=200 kHz 16 bit A/D D/A |
| LSI DBV46 | 2× | TMS320C40 @ 50 MHz | 4 Mb 0ws SRAM |
| | 1× | | 16 Mb shared DRAM |
| | 4× | TMS320C44 @ 50 MHz | 1 Mb 0ws SRAM |

Table 10.4: Specifications of the DSP setup

the far–end loudspeaker and its echo was recorded by the near–end side microphone.

Three adaptive algorithms are compared, which estimate the acoustic echo path with an FIR filter of $L_{FB} = 768$ (96 ms) equivalent taps in the time domain[4] :

1. subband–normalized unconstrained PBFDAF with $L = P = \frac{M}{2} = 64$ (Eqs. 6.28–6.32 and $\mathbf{G} = \mathbf{I}$)
2. subband–normalized constrained PBFDAF[5] with $L = P = \frac{M}{2} = 64$ (Eqs. 6.28–6.32 and 6.41)
3. NLMS (Eqs. 2.22, 2.23 and 2.28)

All data were processed in frames of 64 samples. The results are shown in figure 10.8. It is observed that after a fast initial convergence a residual error remains. This is mainly due to the finite length of the adaptive filters. ▽

In practice the echo suppression will not go below approximately 30 dB as was explained in section 10.1.4. Identifying very long acoustic paths is therefore not advised. It will slow down the convergence while improving the error suppression only marginally. By tracking only the dominant part of the acoustic path —100 ms e.g., as was done in this experiment— sufficient echo suppression is obtained.

The largest adaptive filter that could be implemented in real–time on the DSP using the unconstrained PBFDAF algorithm with $L = P = \frac{M}{2} = 64$ was 325 ms. For

[4] For more information on the algorithmic parameters we refer to previous chapters.

[5] No true DSP implementation was available for the constrained updating. The results were therefore obtained in MATLAB® based on the recorded signals.

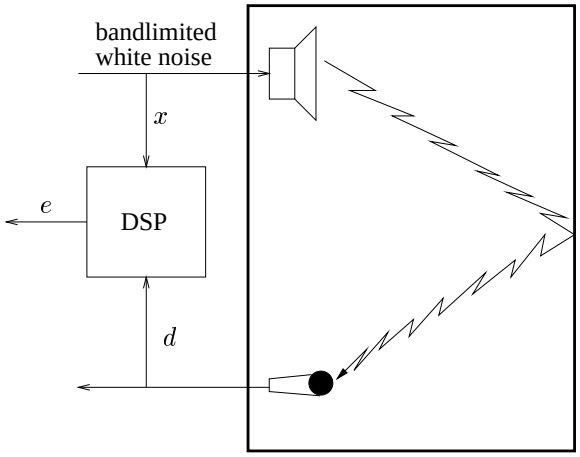


Figure 10.6: Demo setup experiment 10.3

the NLMS algorithm this number was limited to 100 ms : the on–chip memory of the C4x is very fast, but rather small and hence puts an upper bound on the filter length of the NLMS adaptive filter.

**Experiment 10.4** Returning to experiment 10.3, the execution times of the most important blocks of the DSP–code were measured for the unconstrained PBFDAF and the NLMS algorithm and are presented in table 10.5. The cost for the constrained updating or the PBFDRAP (standard implementation of table 8.1) can easily be extrapolated based on this table. It should be mentioned that the computationally intensive parts of the algorithm were optimized in assembler. Only less complex control operations were programmed in C.
It is observed that the PBFDAF is computationally more attractive than the NLMS algorithm. Furthermore, as $L_{FB} = 768$ the NLMS filter coefficients fitted in the small on–chip memory of the C40–processor. For larger filter lengths less efficient memory banks have to be addressed, which strongly increases the execution time for the NLMS. $\bigtriangledown$

**Experiment 10.5** Next, an experiment was set up to test the convergence and tracking properties of some of the many algorithms that were discussed in the thesis. A clean speech signal, sampled at 16 kHz, was played through a loudspeaker and recorded under realistic non–stationary acoustic conditions. The distance between loudspeaker and microphone was approximately 180 cm. The reverberation time was computed following [60]. It appeared that $\mathrm{RT}_{60} \approx 265$ ms. The acoustic path was left unaltered during the first 15 seconds of the recording. During the last 5 seconds the acoustic impulse response was substantially changed through hand movements in between the loudspeaker and microphone. The amount of in-

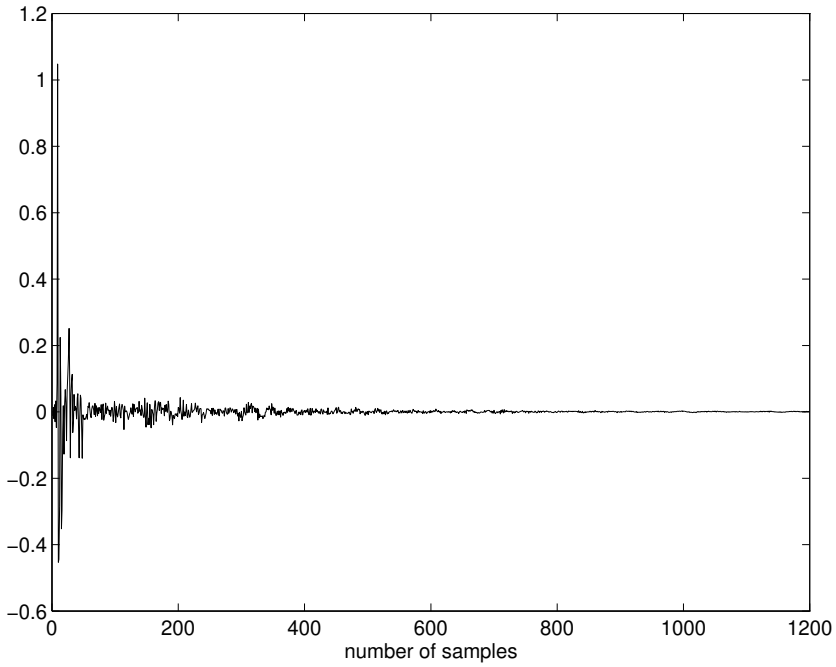


Figure 10.7: Acoustic impulse response of the ESAT speech laboratory, $f_s = 8$ kHz

serted near–end background noise was kept to a minimum throughout the complete recording. The following adaptive filtering algorithms are compared :

1. normalized LMS [Eqs. 2.22, 2.23 and 2.28]
2. subband–normalized constrained PBFDAF ($L = P = \frac{M}{2} = 256$) [Eqs. 6.28–6.32 and 6.41]
3. subband–normalized constrained PBFDRAP ($L = P = \frac{M}{2} = 256$ and $R = 5$) [Eqs. 8.3–8.8 and 6.41]
4. subband adaptive filter with NLMS–based subband adaptation using nearly perfect reconstruction filter banks [chapter 4 and 5], ($M = 16$, $N = 12$, $L_{SB} = 148$, $L_{ac} = 20$, $L_c = 0$, $L_f = 110$)[6]
5. subband adaptive filter with normalized fullband error adaptation [Eq. 7.13], using the same filter bank set and parameters as the previous algorithm

The number of equivalent time–domain filter taps that were modelled equals 1536 for all algorithms, which corresponds to an echo path of 96 ms. The stepsize was

[6] Consult table 10.1 for symbol conventions.

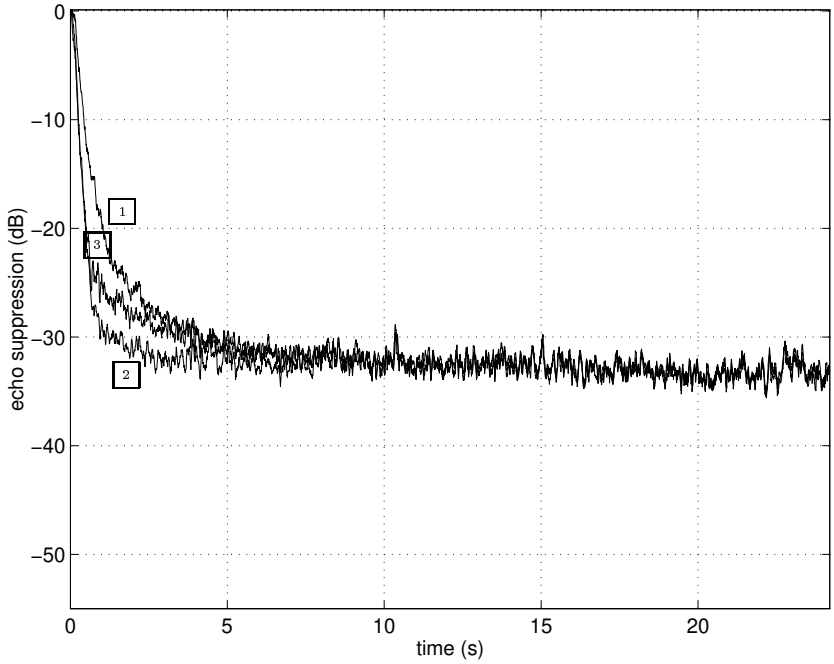


Figure 10.8: Convergence behavior of three adaptive algorithms : 1 corresponds to the subband–normalized unconstrained PBFDAF, 2 to the subband–normalized constrained PBFDAF and 3 is the NLMS algorithm.

optimized for each algorithm such that the fastest convergence was obtained during the first 3 seconds of the experiment. In figure 10.9 the different convergence curves are shown. It is observed that after about 17 seconds the error suppression is reduced somewhat due to the severe time variations of the acoustic path. The PBFDRAP and the NLMS algorithm apparently have the best performance. Further, it appears that in this experiment there is almost no difference in convergence behavior between the standard subband adaptive filter and the subband system using the fullband error adaptation algorithm. ▽

The good performance of the NLMS algorithm in experiment 10.5 should be relativized somewhat. This is proven by the following experiment.

**Experiment 10.6** Experiment 10.5 was repeated : the same algorithms were validated on the first 13.5 seconds, i.e. on the stationary part of the recording. After 10.5 seconds the adaptation was switched off. It is observed from figure 10.10 that the error suppression reduces again as soon as the adaptation is switched off. Although the error is suppressed significantly during the first 10.5 seconds, the

| cost per block of 64 samples | Unconstrained PBFDAF | NLMS |
|---|---|---|
| fill signal buffers | 579.9 $\mu$s | 310.9 $\mu$s |
| first control block | 840.9 $\mu$s | 846.3 $\mu$s |
| copying frames | 80.8 $\mu$s | – |
| FFT | 342.0 $\mu$s | – |
| power estimation | 52.5 $\mu$s | – |
| filtering | 794.9 $\mu$s | – |
| IFFT | 381.4 $\mu$s | – |
| loop | 98.6 $\mu$s | – |
| FFT | 342.2 $\mu$s | – |
| gradient estimation | 624.2 $\mu$s | – |
| adaptation | 498.0 $\mu$s | – |
| NLMS–updating | – | 6046.7 $\mu$s |
| second control block | 22.3 $\mu$s | 11.0 $\mu$s |
| nonlinear processing | 261.0 $\mu$s | 251.2 $\mu$s |
| total | 4922 $\mu$s | 7478 $\mu$s |

Table 10.5: Execution times measured on a 25–MIPS TMS320C40 DSP

adaptive filters apparently have not converged sufficiently : a small error signal does not automatically imply a good model. The effect is most prominent in the case of the NLMS algorithm. Although the error output is small the acoustic model is inferior to that of the other algorithms. $\bigtriangledown$

The echo canceller was further validated through listening tests in the speech laboratory, as shown in figure 10.11. Both the far–end and near–end source were activated. The near–end speaker was simulated by another loudspeaker to which a CD signal can be sent. In this way unwanted time–variations of the echo path by speaker's motion are avoided. Several tests were done under different conditions : multiple signals ((colored) noise, speech, music, ...) were applied, the sampling rate was changed and the source volume was modified.

**Experiment 10.7** In a final experiment the setup of figure 10.11 was used to test the acoustic echo cancellation algorithm under double–talk conditions. To the far–end loudspeaker a prerecorded radio news signal $x$ was sent. A set of isolated words $s$ were played through the near–end loudspeaker (about one word every 5 seconds). With a microphone the near–end signal $d$ was recorded. The test was carried out under stationary acoustics and with (almost) no background noise. Both the far–end and the near–end signal were sampled at 8 kHz.

The echo path was modelled with the subband–normalized unconstrained PBFDAF. Block length $L$ and filter partition length $P$ were 128 and the FFT–size $M = 256$. The equivalent time–domain filter length $L_{FB}$ was set to 896. In this way an echo

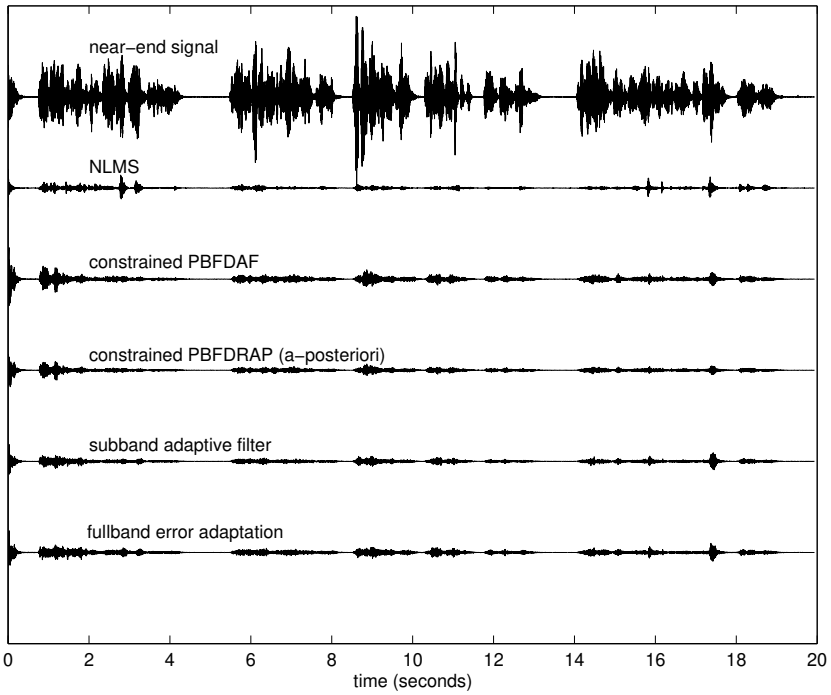


Figure 10.9: Tracking experiment : simulation results

path of 112 ms is being modelled and the input/output delay of the algorithm is 31.875 ms. The energy estimates for the subband normalization were computed following Eq. 10.3.

Apart from the adaptive filter a control algorithm was added that is based on [82]. At the output a center clipper was included as described by Eq. 10.7. The signals were processed off–line using MATLAB®. A time estimation of the same algorithm on DSP showed that using the settings described before 8027 $\mu$s were needed per 128 input samples, which is twice as fast as real–time.

In figure 10.12 the near–end signal $d$ and the error output $e$ of the algorithm are plotted at the top and at the bottom respectively. The middle curve shows the adaptation stepsize $\mu$ as defined in formula 6.35. Double–talk periods were identified off–line and are shown in the same figure as well to serve as a reference to validate the automatic double–talk detection algorithm. It is observed from the error curve that the algorithm converges quite well and that after a while the near–end words can be clearly identified. In the near–end signal $d$ they are almost completely masked by the far–end radio news signal. Also the control algorithm is doing quite well. The adaptation stepsize is kept to zero during double–talk periods. At the

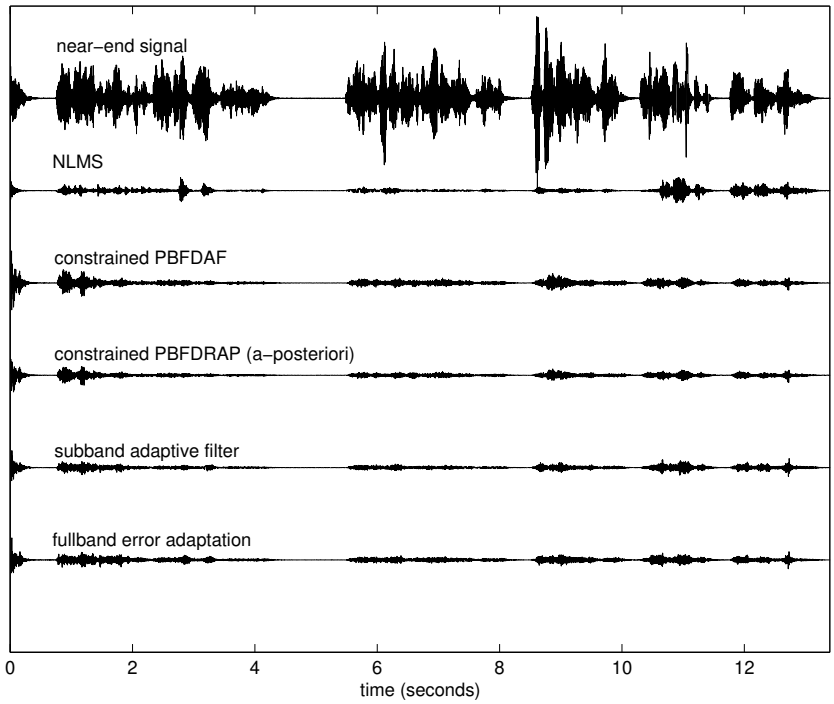


Figure 10.10: Tracking experiment : modelling validation

beginning of these periods large stepsizes are sporadically encountered due to too late double–talk detection. This leads to audible (but acceptable) signal distortion, which was confirmed by listening tests. Observe that as the double–talk detector is far from optimal also during single–talk periods the adaptation is switched off from time to time. ▽

The signals that are shown in figures 10.9, 10.10 and 10.12 can be downloaded from

`http://www.esat.kuleuven.ac.be/~eneman/demo_echo`

for evaluation.

## 10.4 Conclusions

In this chapter the acoustic echo cancellation problem was studied. Some realistic experiments were carried out and different adaptive algorithms that were described

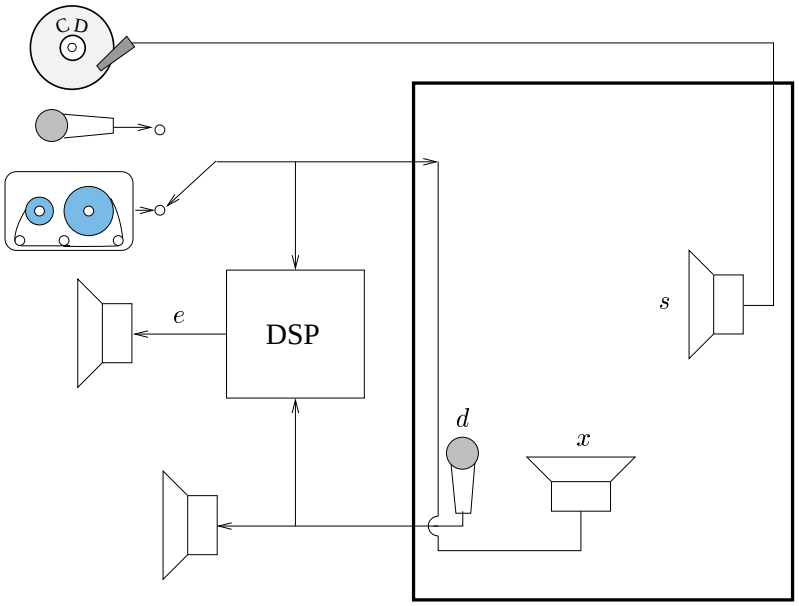


Figure 10.11: Setup listening tests in the speech laboratory

in previous chapters of this work were compared.

In section 10.1 it was motivated that, apart from the adaptive filtering core, echo cancellation systems require a control module, which basically steers the adaptation speed of the adaptive filter(s). Double–talk detection there plays an important role. Although accurate double–talk detection is crucial for proper operation of the echo canceller, reliable detectors are hard to find. Misdetection and other algorithmic artifacts can be neutralized by a post–processing unit, which is indispensable if high quality echo cancellation is aimed at.

Several adaptive filtering algorithms were applied to the acoustic echo cancellation problem and were compared in section 10.2. It appeared that the NLMS algorithm is inferior to subband adaptive filtering if colored signals are involved. Furthermore, it was shown that its implementation cost is higher. The PBFDRAP algorithm and subband adaptive filters using the fullband error adaptation algorithm have improved convergence characteristics, but at the expensive of a higher implementation cost. In general, the PBFDAF algorithm offers the best overall performance/cost–ratio and is hence preferred in most practical applications.

A real–time acoustic echo canceller was implemented on DSP. It was demonstrated in section 10.3 that with a present–day real–time system acoustic channels up to 325 ms can be modelled using the unconstrained PBFDAF algorithm. For the

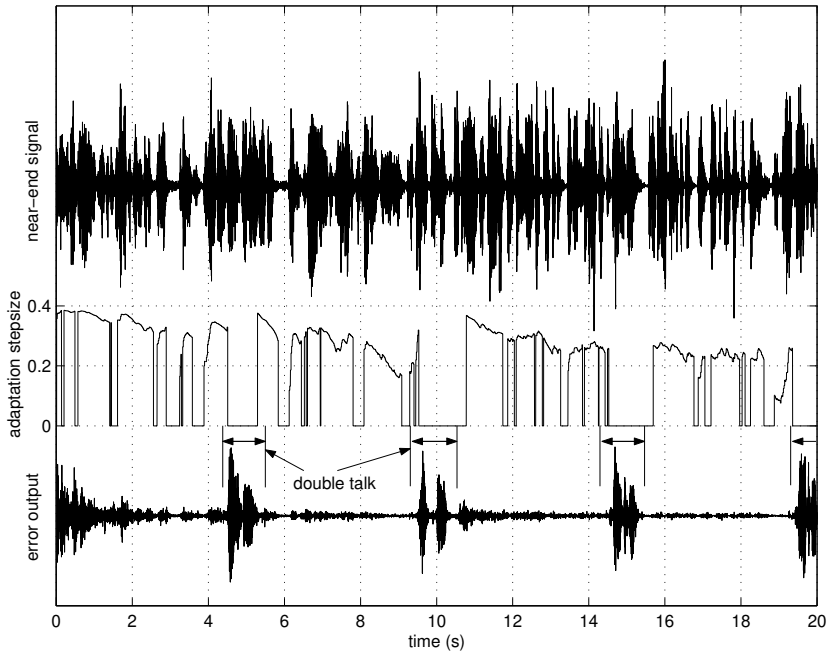


Figure 10.12: Real–time acoustic echo cancellation experiment : simulation results

NLMS algorithm this number was limited to 100 ms : the on–chip memory of the TMS320C4x processor is very fast, but rather small and hence puts a upper bound on the filter length of the NLMS adaptive filter.

# Chapter 11

# Conclusions and Further Research

In the first section of this chapter the global conclusions are formulated. In section 11.2 some suggestions are given for further research.

## 11.1 Conclusions

In **part I** of the thesis the design of perfect and nearly perfect reconstruction DFT modulated filter banks was considered.

Perfect reconstruction filter bank design was discussed in chapter 3. Following the method of [22] it was shown how the polyphase matrix of an oversampled DFT modulated filter bank can be parameterized as a structured polynomial matrix and a DFT operation. Conditions for perfect reconstruction input/output behavior were formulated in terms of the synthesis polynomial matrix $\mathbf{C}(z)$. We considered the Smith–McMillan decomposition as a design method for perfect reconstruction DFT modulated filter banks. Para–unitary filter bank design however turns out to be a better alternative. A design method was discussed that was presented in [24]. It was shown how DFT modulated filter banks can be parameterized and optimized leading to a perfect reconstruction subband system with good bandpass characteristics. We presented an extension to this method based on which the parameterization can be further refined. Hence, the filter bank length can be better adjusted to the design specifications. We further showed that appropriate starting values can be determined using the inverse parametrization. In this way the optimization time is reduced.

In chapter 4 two design procedures for nearly perfect reconstruction oversampled DFT modulated filter banks were presented. First we developed a frequency–domain optimization algorithm that trades off the frequency selectivity of the filter banks against the amplitude distortion between two neighboring bands. A mixed time/frequency–domain design procedure was further proposed. Here, the frequency selectivity and a measure for perfect reconstruction are weighed. It was shown by means of an example that thanks to their better frequency characteristics nearly perfect reconstruction systems outperform their perfect reconstruction counterparts without introducing unacceptable audible distortion.

In **part II** we studied subband and frequency–domain adaptive filters.

The subband adaptive filter was examined in chapter 5. We made a comparison with the standard fullband approach and gave an overview of the most important properties. Three design criteria for appropriate subband modelling were presented. They are necessary requirements to ensure satisfactory performance of the subband adaptive filter. It was indicated that the downsampling operation inserts aliasing components in the subbands, which hamper the convergence of the adaptive filters. It was further shown that subband adaptive filters suffer from a residual undermodelling error. We derived a formula that describes the phenomenon and presented two design methods from which the appropriate subband adaptive filter order can be computed. Finally, we performed a cost analysis for the subband adaptive filter. It appeared that although the complexity gain w.r.t. the equivalent fullband algorithm is less than expected, still a considerable complexity reduction can be obtained.

In chapter 6 the Partitioned Block Frequency–Domain Adaptive Filter (PBFDAF) was discussed. It is known from the literature that with the PBFDAF a substantial cost reduction can be obtained w.r.t. the LMS algorithm, without introducing unacceptable algorithmic delays. It was shown how the PBFDAF algorithm can be derived from the Block–LMS adaptive filter and some of its properties were discussed. We showed that an ambiguity can occur with the unconstrained PBFDAF, which can easily be compensated for. Based on a graphical representation of the algorithm it was shown that the PBFDAF fits into the subband adaptive framework presented in chapter 5. Although the filter banks used by the PBFDAF have a rather poor frequency selectivity, thanks to an extra error correction module the algorithm has a superior convergence behavior compared to standard subband adaptive filters. It was also shown that two out of the three design criteria for subband adaptive systems, which were formulated in chapter 5, are fulfilled by the PBFDAF. Finally, we performed a cost analysis for the PBFDAF. It was illustrated how the global cost changes as a function of different algorithmic parameters.

A fullband error adaptation scheme for subband adaptive filters was proposed in chapter 7. This novel update scheme adjusts the subband filters by feeding back the fullband error instead of using the subband errors as in a classical subband adaptive system. Different variants of the adaptation algorithm were discussed and

a cost analysis was performed. Although the fullband error adaptation scheme outperforms the standard subband adaptive filter in terms of convergence behavior, the computational cost is higher as the subband errors pass through the synthesis filter bank twice. We furthermore proved that the weight updating of the unconstrained PBFDAF algorithm corresponds to that of the fullband error adaptation algorithm if $P$ is a multiple of $L$. This demonstrates that the fullband error adaptation algorithm can be considered as an extension of the PBFDAF overlap–save error correction mechanism to a more general class of subband adaptive filters.

In **part III** the PBFDRAP algorithm was considered, which combines partitioned block frequency–domain adaptive filtering and row action projection.

In chapter 8 the PBFDRAP algorithm was defined and its working mechanism was explained. Next, the asymptotic properties of the algorithm were analyzed, for which different algorithmic settings were considered. It was found that the unnormalized unconstrained PBFDRAP approaches the unconstrained PBFDAF that is normalized based on projected subband energies and that is adapted with stepsize 1. The unnormalized constrained PBFDRAP approaches the PRA with stepsize 1 by series expansion of the expensive matrix inverse operation. For the subband–normalized PBFDRAP increasing the number of iteration steps has the same effect as applying a larger stepsize. Stability is not guaranteed however for the subband–normalized constrained PBFDRAP. Finally, it was shown that the PBFDRAP outperforms the PBFDAF in a realistic echo cancellation setup.

In chapter 9 fast versions for the PBFDRAP algorithm were derived, leading to a significant reduction of the computational complexity. The fast PBFDRAP variants were compared with a standard implementation of the algorithm. It was verified that a significant cost reduction can be obtained depending on the mode in which the algorithm operates and depending on the choice of the parameters. Finally, the unnormalized and globally normalized constrained PBFDRAP were compared with the PRA algorithm. We concluded that for large block lengths, the iterated PBFDRAP is a cheaper alternative to the PRA.

In **part IV** we described applications of our algorithms to the acoustic echo cancellation problem. It was seen that the algorithms that were designed in the preceding chapters can be successfully applied to a real–world signal enhancement setup.

We motivated in chapter 10 that, apart from the adaptive filtering core, echo cancellation systems require an adequate control module, which basically steers the adaptation speed of the adaptive filter(s). Double–talk detection there plays an important role. Although correct double–talk detection is crucial for proper operation of the echo canceller it is difficult to design reliable and robust detectors. Misdetection and other algorithmic artifacts can be neutralized by adding a post–processing unit, which is indispensable if high quality echo cancellation is aimed at. Several adaptive filtering algorithms were applied to the acoustic echo cancellation problem for comparison. It appeared that the NLMS algorithm is inferior to

subband echo cancellers if colored signals are involved and that its implementation cost is higher. The PBFDAF algorithm showed the best performance/cost–ratio and is hence preferred in many applications. A real–time acoustic echo canceller was implemented on DSP. It was demonstrated that with a real–time system based on the unconstrained PBFDAF algorithm acoustic channels up to 325 ms can be modelled.

## 11.2 Suggestions for further research

In chapter 3 several design methods were presented for perfect reconstruction filter banks. Through para–unitary parameterization perfect reconstruction DFT modulated filter banks could be obtained. The order of these filter banks can be adjusted to the designer's wishes. However, it appears that only for certain filter orders, for which a minimum number of coefficients is fixed to zero or one, the optimization procedure easily converges to a solution with frequency selective spectral characteristics. It is not clear whether more advanced techniques can be found by means of which frequency–selective perfect reconstruction filter banks with arbitrary filter order can be designed.

Further, throughout this thesis DFT modulated FIR filter banks were applied. Although the different filters have uniform frequency characteristics the performance of these filter banks is satisfactory in many practical applications. The main drawback of this approach is the inherent delay that is introduced by FIR filter banks. It is known on the other hand that with IIR filters an equal frequency selectivity can be obtained with a reduced filter order. On the other hand, polyphase decomposition, which forms the basis of the considerable complexity reduction that can be achieved with FIR filter banks, is not straightforwardly extendible to the IIR case. It would be interesting to look into IIR–based filter banks and check whether IIR perfect or nearly perfect reconstruction filter banks can easily be designed and implemented at an acceptable cost.

In part II subband and frequency–domain adaptive filtering techniques were discussed. It appeared that the partitioned block frequency–domain adaptive filter (PBFDAF) fits into the subband adaptive framework. Although the filter banks used by the PBFDAF have a rather poor frequency selectivity, the algorithm easily outperforms the standard subband adaptive filter in terms of cost and convergence behavior. The leitmotiv throughout part II therefore was first trying to figure out which properties of the PBFDAF make the algorithm perform so well, and then finding a way to extend these properties to a more general class of subband adaptive filters. As a result three design criteria were specified and a fullband error adaptation algorithm was proposed. Filter banks systems could be designed that use this fullband error adaptation algorithm and fulfill the first and the second criterion on frequency selectivity and perfect reconstruction. However, their per-

formance was still inferior to that of the PBFDAF. It is believed that this inferior performance is due to the fact that the third criterion on perfect path modelling could up till now not be fulfilled for a more general class of subband adaptive filters, unless infinite–order subband filters were employed. An interesting research topic would therefore be to check first whether there are subband systems apart from the PBFDAF that fulfill the third criterion with finite–length filters, and if so, how they can be designed.

The fullband error adaptation algorithm, which was proposed in chapter 7, can be considered as an extension of the PBFDAF overlap–save correction on the error output to a more general class of subband adaptive filters. Apart from an overlap–save compensation on the error also the weight vectors of the constrained PBFDAF undergo a similar transformation. In the same way we could try to derive a generalized weight update equation for the subband adaptive filter. It would be interesting to investigate the effect of such generalization on the overall performance of the subband adaptive system.

In part IV several subband and frequency–domain adaptive algorithms were applied to the acoustic echo cancellation problem to illustrate their validity and applicability to this real–life engineering problem. As argued several times in the text, it is believed that multirate techniques, of which the subband and the frequency–domain adaptive filter are typical examples, are a powerful means to reduce the complexity and to increase the performance w.r.t. standard fullband approaches in enhancement applications were time–varying broadband signals such as speech are involved. It would therefore be interesting to look also into other kind of enhancement applications, for instance noise reduction or dereverberation, to see whether subband or frequency–domain techniques and their underlying principles can be successfully applied to these applications as well.

# Bibliography


[1] C. Acker and P. Vary. Combined Implementation of Predictive Speech Coding and Acoustic Echo Cancellation. In J. Vandewalle, R. Boite, M. Moonen, and A. Oosterlinck, editors, *Signal Processing VI: Theories and Applications*, pages 1641–1644. Elsevier Science Publishers B.V., 1992.

[2] S. Affes and Y. Grenier. A Signal Subspace Tracking Algorithm for Microphone Array Processing of Speech. *IEEE Transactions on Speech and Audio Processing*, 5(5):425–437, September 1997.

[3] A. Akansu and M. Smith. *Subband and Wavelet Transforms*. Kluwer Academic Publishers, Boston, 1995.

[4] J. Allen and D. Berkley. Image method for efficiently simulating small–room acoustics. *Journal of the Acoustical Society of America*, 65(4):943–950, April 1979.

[5] M. Asharif and F. Amano. Acoustic Echo–Canceler Using the FBAF Algorithm. *IEEE Transactions on Communications*, 42(12):3090–3094, December 1994.

[6] D. Bees, M. Blostein, and P. Kabal. Reverberant Speech Enhancement Using Cepstral Processing. In *Proceedings of the 1991 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP91)*, pages 977–980, Toronto, Ontario, May 1991.

[7] J. Benesty, A. Gilloire, and Y. Grenier. A Frequency Domain Stereophonic Acoustic Echo Canceler exploiting the coherence between channels and using nonlinear transformations. In *Proceedings of the 1999 International Workshop on Acoustic Echo and Noise Control (IWAENC99)*, pages 28–31, Pocono Manor, Pennsylvania, September 1999.

[8] J. Benesty, D. Morgan, and J. Cho. A New Class of Doubletalk Detectors Based on Cross–Correlation. *IEEE Transactions on Speech and Audio Processing*, 8(2):168–172, March 2000.

[9] A. Birkett and R. Goubran. Limitations of Handsfree Acoustic Echo Cancellers due to Nonlinear Loudspreaker Distortion and Enclosure Vibration Effects. In *Proceedings of the 1995 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA95)*, New Paltz, New York, October 1995.

[10] H. Bölcskei, F. Hlawatsch, and H. Feichtinger. Oversampled FIR and IIR DFT Filter Banks and Weyl–Heisenberg Frames. In *Proceedings of the 1996 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP96)*, Atlanta, Georgia, May 1996.

[11] S. Boll. Suppression of Acoustic Noise in Speech Using Spectral Subtraction. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 27(2):113–120, April 1979.

[12] F. Capman, J. Boudy, and P. Lockwood. Acoustic Echo Cancellation using a Fast QR–RLS Algorithm and Multirate Schemes. In *Proceedings of the 1995 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP95)*, pages 969–972, Detroit, Michigan, May 1995.

[13] J. Chen, H. Bes, J. Vandewalle, I. Evers, and P. Janssens. A Zero–Delay FFT–Based Subband Acoustic Echo Canceller for Teleconferencing and Hands–Free Telephone Systems. *IEEE Transactions on Circuits and Systems*, 43(10):713–717, October 1996.

[14] J. Chen, H. Bes, J. Vandewalle, and P. Janssens. A New Structure for Sub–band Acoustic Echo Canceler. In *Proceedings of the 1988 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP88)*, pages 2574–2577, New York, New York, April 1988.

[15] J. Chen, J. Vandewalle, H. Bes, I. Evers, and P. Janssens. Design of a Zero Delay Subband Acoustic Echo Canceller. In *Proceedings of the 1988 IEEE International Symposium on Circuits and Systems (ISCAS88)*, pages 1325–1328, Espoo, Finland, June 1988.

[16] J. Cho, D. Morgan, and J. Benesty. An Objective Technique for Evaluating Doubletalk Detectors in Acoustic Echo Cancelers. *IEEE Transactions on Speech and Audio Processing*, 7:718–724, November 1999.

[17] I. Claesson and S. Nordholm. A Spatial Filtering Approach to Robust Adaptive Beaming. *IEEE Transactions on Antennas and Propagation*, 40(9):1093–1096, September 1992.

[18] I. Claesson, S. Nordholm, and B. Bengtsson. A Multi–DSP Implementation of a Broad–Band Adaptive Beamformer for Use in a Hands–Free Mobile Radio Telephone. *IEEE Transactions on Vehicular Technology*, 40(1):194–202, February 1991.

[19] G. Clark, S. Mitra, and S. Parker. Block Implementation of Adaptive Digital Filters. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29(3):744–752, June 1981.

[20] G. Clark, S. Parker, and S. Mitra. A Unified Approach to Time– and Frequency–Domain Realization of FIR Adaptive Digital Filters. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 31(5):1073–1083, October 1983.

[21] R. Cox. The Design of Uniformly and Nonuniformly Spaced Pseudoquadrature Mirror Filters. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 34(5):1090–1096, October 1986.

[22] Z. Cvetković. Oversampled Modulated Filter Banks and Tight Gabor Frames in $l^2(\mathbb{Z})$. In *Proceedings of the 1995 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP95)*, pages 1456–1459, Detroit, Michigan, May 1995.

[23] Z. Cvetković and M. Vetterli. FIR Oversampled Filter Banks and Frames in $l^2(\mathbb{Z})$. In *Proceedings of the European Signal Processing Conference (EUSIPCO96)*, Trieste, Italy, September 1996.

[24] Z. Cvetković and M. Vetterli. Oversampled Filter Banks. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 46(5):1245–1255, May 1998.

[25] J. Davies. Wireless carriers await fallout from N.Y. ban. The San Diego Union–Tribune, June 27 2001, `http://www.jabra.com/news/viewarticle.cfm?id=61`.

[26] P. De León II and D. Etter. Experimental Results with Increased Bandwidth Analysis Filters in Oversampled Subband Acoustic Echo Cancelers. *IEEE Signal Processing Letters*, 2(1):1–3, January 1995.

[27] P. De León II and D. Etter. Experimental Results of Subband Acoustic Echo Cancelers under Spherically Invariant Random Processes. In *Proceedings of the 1996 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP96)*, pages 961–964, Atlanta, Georgia, May 1996.

[28] M. Deisher and A. Spanias. Practical Considerations in the Implementation of a Frequency–Domain Adaptive Noise Canceller. *IEEE Transactions on Circuits and Systems*, 41(2):164–168, February 1994.

[29] J. Deller, J. Proakis, and J. Hansen. *Discrete–Time Processing of Speech Signals*. Prentice Hall, Englewood Cliffs, New Jersey, 1987.

[30] S. Doclo and E. De Clippel. Verbetering van spraakverstaan bij hoortoestellen via adaptieve ruisonderdrukking in reële tijd, master thesis, Katholieke Universiteit Leuven, 1997.

[31] S. Doclo, E. De Clippel, and M. Moonen. Combined acoustic echo and noise reduction using GSVD–based optimal filtering. In *Proceedings of the 2000 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP00)*, pages 1061–1064 (vol. 2), Istanbul, Turkey, June 2000.

[32] S. Doclo, K. Eneman, and M. Moonen. Voice activity detection. Technical Report 02–4, ESAT–SISTA, Katholieke Universiteit Leuven, Belgium, January 2002.

[33] S. Doclo and M. Moonen. GSVD–based Optimal Filtering for Multi–Microphone Speech Enhancement. In M. S. Brandstein and D. B. Ward, editors, *Microphone Arrays: Techniques and Applications*. Springer–Verlag (Heidelberg, Germany), 2001.

[34] E. Dowling, D. Linebarger, Y. Tong, and M. Munoz. An adaptive microphone array processing system. *Microprocessors and Microsystems*, 16(10):507–515, 1992.

[35] Y. Duk, K. Al–Naimi, and A. Kondoz. Improved Voice Activity Detection Based on a Smoothed Statistical Likelihood Ratio. In *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP01)*, Salt Lake City, Utah, May 2001.

[36] G. Egelmeers. *Real Time Realization Concepts of large Adaptive Filters.* PhD thesis, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, November 1995.

[37] G. Egelmeers and P. Sommen. Acoustic echo cancellation based on Non–Uniform Partitioned BFDAF. In *Proceedings of the ProRISC/IEEE Benelux workshop on Circuits, Systems and Signal Processing*, pages 67–73, Mierlo, The Netherlands, March 1995.

[38] G. Egelmeers, P. Sommen, and J. de Boer. Realization of an Acoustic Echo Canceller on a Single DSP. In *Proceedings of the European Signal Processing Conference (EUSIPCO96)*, pages 33–36, Trieste, Italy, September 1996.

[39] EMC. EMC forecasts subscribers to top 1 billion by end 2001. `http://www.emc-database.com/website.nsf/index/pr010718#this-page`, July 2001.

[40] K. Eneman, S. Doclo, and M. Moonen. A Comparison between Iterative block–LMS and Partial Rank Algorithm. Technical Report 99–52, ESAT–SISTA, Katholieke Universiteit Leuven, Belgium, March 1999.

[41] K. Eneman and M. Moonen. A Relation Between Subband and Frequency–Domain Adaptive Filtering. In *Proceedings of the 13th International Conference on Digital Signal Processing (DSP97)*, Thira, Santorini, Greece, July 1997.

[42] K. Eneman and M. Moonen. Filter Bank Constraints for Subband and Frequency–Domain Adaptive Filters. In *Proceedings of the 1997 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA97)*, New Paltz, New York, October 1997.

[43] K. Eneman and M. Moonen. Para–Unitary Filter Bank Design for Oversampled Subband Systems. Technical Report 97–119, ESAT–SISTA, Katholieke Universiteit Leuven, Belgium, December 1997.

[44] K. Eneman and M. Moonen. Real–Time Implementation of an Acoustic Echo Canceller on DSP. In *Proceedings of the ProRISC/IEEE Benelux workshop on Circuits, Systems and Signal Processing*, pages 163–170, Mierlo, The Netherlands, November 1997.

[45] K. Eneman and M. Moonen. DFT Modulated Filter Bank Design for Oversampled Subband Systems. In *Proceedings of the IEEE Benelux Signal Processing Symposium*, pages 87–90, Leuven, Belgium, March 1998.

[46] K. Eneman and M. Moonen. Real–Time Implementation of an Acoustic Echo Canceller. In *Proceedings of the COST#254 Workshop on Intelligent Communications*, pages 27–32, L'Aquila, Italy, June 1998.

[47] K. Eneman and M. Moonen. A Fullband Adaptation Scheme for Subband Adaptive Filters. Technical Report 99–103, ESAT–SISTA, Katholieke Universiteit Leuven, Belgium, December 1999.

[48] K. Eneman and M. Moonen. Hybrid Subband/Frequency–Domain Adaptive Systems for Acoustic Echo Cancellation. In *Proceedings of the 1999 International Workshop on Acoustic Echo and Noise Control (IWAENC99)*, pages 144–147, Pocono Manor, Pennsylvania, September 1999.

[49] K. Eneman and M. Moonen. Fullband Error Adaptation of Subband Adaptive Filters. In *Proceedings of the IEEE International Symposium 2000 on Adaptive Systems for Signal Processing, Communications and Control*, pages 293–298, Lake Louise, Alberta, October 2000.

[50] K. Eneman and M. Moonen. On Iterating the Partitioned Block Frequency–Domain Adaptive Filter. Technical Report 00–127, ESAT–SISTA, Katholieke Universiteit Leuven, Belgium, December 2000.

[51] K. Eneman and M. Moonen. On the Length of Subband Adaptive Filters. Technical Report 00–2, ESAT–SISTA, Katholieke Universiteit Leuven, Belgium, January 2000.

[52] K. Eneman and M. Moonen. DFT Modulated Filter Bank Design for Oversampled Subband Systems. *Signal Processing*, 81(9):1947–1973, September 2001.

[53] K. Eneman and M. Moonen. Hybrid Subband/Frequency–Domain Adaptive Systems. *Signal Processing*, 81(1):117–136, January 2001.

[54] K. Eneman and M. Moonen. The iterated partitioned block frequency–domain adaptive filter for acoustic echo cancellation. In *Proceedings of the 2001 IEEE/EURASIP International Workshop on Acoustic Echo and Noise Control (IWAENC01)*, pages 135–138, Darmstadt, Germany, September 2001.

[55] K. Eneman and M. Moonen. Iterated Partitioned Block Frequency–Domain Adaptive Filtering for Acoustic Echo Cancellation. Technical Report 00–83, ESAT–SISTA, Katholieke Universiteit Leuven, Belgium, May 2001, *Submitted for publication.*

[56] K. Eneman and M. Moonen. Adding Causal and Anti–Causal Filter Taps to Enhance the Performance of Subband Adaptive Filters. Technical Report 02–20, ESAT–SISTA, Katholieke Universiteit Leuven, Belgium, February 2002, *Accepted for publication in the Proceedings of the IEEE Benelux Signal Processing Symposium (SPS 2002).*

[57] K. Eneman, M. Moonen, and I. Proudler. Frequency–domain adaptive echo cancellation as a special case of subband echo suppression. In *Proceedings of the ProRISC/IEEE Benelux workshop on Circuits, Systems and Signal Processing*, pages 131–136, Mierlo, The Netherlands, November 1996.

[58] P. Eneroth, S. Gay, T. Gänsler, and J. Benesty. A Hybrid FRLS/NLMS Stereo Acoustic Echo Canceller. In *Proceedings of the 1999 International Workshop on Acoustic Echo and Noise Control (IWAENC99)*, pages 20–23, Pocono Manor, Pennsylvania, September 1999.

[59] P. Estermann and A. Kaelin. A Hands–free Phone System Based on a Partitioned Frequency–Domain Adaptive Echo Canceler. In *Proceedings of the European Signal Processing Conference (EUSIPCO96)*, pages 1131–1134, Trieste, Italy, September 1996.

[60] L. Faiget, R. Ruiz, and C. Legros. The True Duration of the Impulse Response Used to Estimate Reverberation Time. In *Proceedings of the 1996 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP96)*, pages 913–916, Atlanta, Georgia, May 1996.

[61] B. Farhang–Boroujeny. Analysis and Efficient Implementation of Partitioned Block LMS Adaptive Filters. *IEEE Transactions on Signal Processing*, 44(11):2865–2868, November 1996.

[62] B. Farhang–Boroujeny and S. Gazor. Performance of LMS–Based Adaptive Filters in Tracking a Time–Varying Plant. *IEEE Transactions on Signal Processing*, 44(11):2868–2871, November 1996.

[63] G. Faucon and R. Le Bouquin Jeannès. Joint System for Acoustic Echo Cancellation and Noise Reduction. In *Proceedings of 4th European Conference on Speech Communication and Technology (ECSCT95)*, pages 1525–1528, Madrid, Spain, September 1995.

[64] E. Ferrara. Fast Implementation of LMS Adaptive Filters. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(4):474–475, August 1980.

[65] G. Franklin, J. Powell, and A. Emami–Naeini. *Feedback Control of Dynamic Systems, Third Edition*. Addison–Wesley Publishing Company, 1994.

[66] O. Frost. An Algorithm for Linearly Constrained Adaptive Array Processing. *Proceeding of the IEEE*, 60(8):926–935, August 1972.

[67] S. Gannot and M. Moonen. Subspace Methods for Multi–Microphone Speech Dereverberation. In *Proceedings of the 2001 IEEE/EURASIP International Workshop on Acoustic Echo and Noise Control (IWAENC01)*, pages 47–50, Darmstadt, Germany, September 2001.

[68] T. Gänsler and J. Benesty. Multichannel acoustic echo cancellation: What's new ? In *Proceedings of the 2001 IEEE/EURASIP International Workshop on Acoustic Echo and Noise Control (IWAENC01)*, pages 14–18, Darmstadt, Germany, September 2001.

[69] S. Gay. *Fast Projection Algorithms with Application to Voice Echo Cancellation*. PhD thesis, Rutgers, The State University of New Jersey, New Brunswick, New Jersey, October 1994.

[70] A. Gilloire and M. Vetterli. Adaptive Filtering in Sub–Bands. In *Proceedings of the 1988 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP88)*, pages 1572–1575, New York, New York, April 1988.

[71] A. Gilloire and M. Vetterli. Adaptive Filtering in Subbands with Critical Sampling : Analysis, Experiments and Application to Acoustic Echo Cancellation. *IEEE Transactions on Signal Processing*, 40(8):1862–1875, August 1992.

[72] S. Gökhun Tanyer and H. Özer. Voice Activity Detection in Nonstationary Noise. *IEEE Transactions on Speech and Audio Processing*, 8(4):478–482, July 2000.

[73] G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins university press, Baltimore, 1993.

[74] Y. Grenier. A microphone array for car environments. *Speech Communication*, (12):25–39, March 1993.

[75] Y. Haneda. Common Acoustical Pole and Zero Modeling of Room Transfer Functions. *IEEE Transactions on Speech and Audio Processing*, 2(2):320–328, April 1994.

[76] E. Hänsler. The Hands–Free Telephone Problem. In *Proceedings of the 1992 IEEE International Symposium on Circuits and Systems (ISCAS92)*, pages 1914–1917, San Diego, California, May 1992.

[77] E. Hänsler. The hands–free telephone problem — An annotated bibliography. *Signal Processing*, 27:259–271, June 1992.

[78] M. Harteneck, R. Stewart, and J. Páez–Borrallo. A Filterbank Design for Oversampled Filter Banks without Aliasing in the Subbands. In *Proceedings of the 2nd UK Symposium on Applications of Time–Frequency and Time–Scale Models (TFTS97)*, pages 161–164, Warwick, UK, August 1997.

[79] M. Harteneck, R. Stewart, and J. Páez–Borrallo. An Oversampled Subband Adaptive Filter without Cross Adaptive Filters. In *Proceedings of the 1997 International Workshop on Acoustic Echo and Noise Control (IWAENC97)*, London, UK, December 1997.

[80] S. Haykin. *Blind Deconvolution.* Prentice Hall, Englewood Cliffs, New Jersey, 1994.

[81] S. Haykin. *Adaptive Filter Theory, 3th ed.* Prentice Hall, Englewood Cliffs, New Jersey, 1996.

[82] P. Heitkämper. An Adaptation Control for Acoustic Echo Cancellers. *IEEE Signal Processing Letters*, 4(6):170–172, June 1997.

[83] J. Homer. *Adaptive Echo Cancellation in Telecommunications.* PhD thesis, The Australian National University, Newcastle, Australia, April 1994.

[84] L.-S. Huang and C.-H. Yang. A Novel Approach to Robust Speech Endpoint Detection in Car Environments. In *Proceedings of the 2000 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP00)*, pages 1751–1754 (vol. 3), Istanbul, Turkey, June 2000.

[85] R. Hundt. New Economy and New Politics = The Medium is the Message. Stanford University (`http://www.stanford.edu/class/econ153/hundt%20stanforduniv..ppt`), October 2000.

[86] International Telecommunication Union. General Characteristics of International Telephone Connections and International Telephone Circuits: Acoustic Echo Controllers (ITU–T, Recommendation G.167). March 1993.

[87] N. Jayant, J. Johnston, and R. Safranek. Signal Compression Based on Models of Human Perception. *Proceeding of the IEEE*, 81(10):1385–1421, October 1993.

[88] J-C. Junqua, B. Mak, and B. Reaves. A Robust Algorithm for Word Boundary Detection in the Presence of Noise. *IEEE Transactions on Speech and Audio Processing*, 2(3):406–412, July 1994.

[89] K. Kajita, H. Kobayashi, S. Muramatsu, A. Yamada, and H. Kiya. A Design Method for Oversampled Paraunitary DFT Filter Banks using Householder Factorization. In *Proceedings of the European Signal Processing Conference (EUSIPCO96)*, pages 192–195, Trieste, Italy, September 1996.

[90] W. Kellermann. Analysis and Design of Multirate Systems for Cancellation of Acoustical Echoes. In *Proceedings of the 1988 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP88)*, pages 2570–2573, New York, New York, April 1988.

[91] W. Kellermann. A Self–Steering Digital Microphone Array. In *Proceedings of the 1991 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP91)*, pages 3581–3584, Toronto, Ontario, May 1991.

[92] R. Kennedy, T. Abhayapala, D. Ward, and R. Williamson. Nearfield Broadband Frequency Invariant Beamforming. In *Proceedings of the 1996 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP96)*, pages 905–908, Atlanta, Georgia, May 1996.

[93] H. Kuttruff. *Room Acoustics*. Applied Science Publishers LTD, Ripple Road, Barking, Essex, England, 2nd edition, 1979.

[94] I. Lecomte, M. Lever, J. Boudy, and A. Tassy. Car Noise Processing for Speech Input. In *Proceedings of the 1989 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP89)*, pages 512–515, Glasgow, Scotland, May 1989.

[95] X. Li and W. Jenkins. The Comparison of the Constrained and Unconstrained Frequency–Domain Block–LMS Adaptive Algorithm. *IEEE Transactions on Signal Processing*, 44(7):1813–1816, July 1996.

[96] Q-G. Liu, B. Champagne, and P. Kabal. A microphone array processing technique for speech enhancement in a reverberant space. *Speech Communication*, (18):317–334, June 1996.

[97] F. Lorenzelli, A. Wang, and K. Yao. Broadband Array Processing using Subband Techniques. In *Proceedings of the 1996 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP96)*, pages 2876–2879, Atlanta, Georgia, May 1996.

[98] J. Lynch, J. Josenhans, and R. Crochiere. Speech/Silence Segmentation for Real–Time Coding via Rule Based Adaptive Endpoint Detection. In *Proceedings of the 1987 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP87)*, pages 1348–1351, Dallas, Texas, April 1987.

[99] S. Makino, J. Noebauer, Y. Haneda, and A. Nakagawa. SSB Subband Echo Canceller using Low–order Projection Algorithm. In *Proceedings of the 1996 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP96)*, Atlanta, Georgia, May 1996.

[100] H. Malvar. *Signal Processing with Lapped Transforms.* Artech House, Boston, 1992.

[101] D. Mansour and A. Gray. Unconstrained Frequency–Domain Adaptive Filter. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 30(5):726–734, October 1982.

[102] R. Martin. Combined Acoustic Echo Cancellation, Spectral Echo Shaping, and Noise Reduction. In *Proceedings of the 1995 International Workshop on Acoustic Echo and Noise Control (IWAENC95)*, pages 48–51, Røros, Norway, June 1995.

[103] R. Martin and J. Altenhöner. Coupled Adaptive Filters for Acoustic Echo Control and Noise Reduction. In *Proceedings of the 1995 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP95)*, pages 3043–3046, Detroit, Michigan, May 1995.

[104] R. Martin and P. Vary. Combined acoustic echo cancellation, dereverberation and noise reduction : a two microphone approach. In *Annales des Télécommunications*, pages 429–438, 1994.

[105] R. Martin and P. Vary. Combined Acoustic Echo Control and Noise Reduction for Hands–Free Telephony — State of the Art and Perspectives. In *Proceedings of the European Signal Processing Conference (EUSIPCO96)*, pages 1107–1110, Trieste, Italy, September 1996.

[106] H. Matt and M. Walker. Handsfree Speaking for Communication Terminals. In *Proceedings of the European Signal Processing Conference (EUSIPCO96)*, pages 1139–1142, Trieste, Italy, September 1996.

[107] J. Mau, J. Valot, and D. Minaud. Time–Varying Orthogonal Filter Banks without Transient Filters. In *Proceedings of the 1995 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP95)*, pages 1328–1331, Detroit, Michigan, May 1995.

[108] M. Mboup and M. Bonnet. On the Adequateness of IIR Adaptive Filtering for Acoustic Echo Cancellation. In J. Vandewalle, R. Boite, M. Moonen, and A. Oosterlinck, editors, *Signal Processing VI: Theories and Applications*, pages 111–114. Elsevier Science Publishers B.V., 1992.

[109] R. Merched, P. Diniz, and M. Petraglia. A Delayless Alias–Free Subband Adaptive Filter Structure. In *Proceedings of the 1997 IEEE International Symposium on Circuits and Systems (ISCAS97)*, pages 2329–2332, Hong Kong, June 1997.

[110] R. Middleton and G. Goodwin. *Digital Control and Estimation.* Prentice Hall, Englewood Cliffs, New Jersey, 1990.

[111] S. Mitra, H. Babic, and V. Somayazulu. A Modified Perfect Reconstruction QMF Bank with an Auxiliary Channel. In *Proceedings of the 1989 IEEE International Symposium on Circuits and Systems (ISCAS89)*, pages 2132–2135, Portland, Oregon, May 1989.

[112] M. Miyoshi and Y. Kaneda. Inverse Filtering of Room Acoustics. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 36(2):145–152, February 1988.

[113] M. Mohan Sondhi, D. Morgan, and J. Hall. Stereophonic Acoustic Echo Cancellation — An Overview of the Fundamental Problem. *IEEE Signal Processing Letters*, 2(8):148–151, August 1995.

[114] M. Moonen and I. Proudler. *An Introduction to Adaptive Signal Processing.* Course Notes, Katholieke Universiteit Leuven, Belgium, 2001.

[115] D. Morgan. Slow Asymptotic Convergence of LMS Acoustic Echo Cancelers. *IEEE Transactions on Speech and Audio Processing*, 3(2):126–136, March 1995.

[116] D. Morgan and S. Kratzer. On a Class of Computationally Efficient, Rapidly Converging, Generalized NLMS Algorithms. *IEEE Signal Processing Letters*, 3(8):245–247, August 1996.

[117] D. Morgan and J. Thi. A Delayless Subband Adaptive Filter Architecture. *IEEE Transactions on Signal Processing*, 43(8):1819–1830, August 1995.

[118] E. Moulines, P. Duhamel, J.-F. Cardoso, and S. Mayrargue. Subspace Methods for the Blind Identification of Multichannel FIR Filters. *IEEE Transactions on Signal Processing*, 43(2):516–525, February 1995.

[119] P. Naylor and J. Hart. Subband Acoustic Echo Control using Non–Critical Frequency Sampling. In *Proceedings of the European Signal Processing Conference (EUSIPCO96)*, pages 37–40, Trieste, Italy, September 1996.

[120] P. Nelson and F. Orduña–Bustamante. Inverse Filter Design and Equalization Zones in Multichannel Sound Reproduction. *IEEE Transactions on Speech and Audio Processing*, 3(3):185–192, May 1995.

[121] P. Noll. Wideband Speech and Audio Coding. *IEEE Communications Magazine*, pages 34–44, November 1993.

[122] S. Nordebo, I. Claesson, and S. Nordholm. Weighted Chebyshev Approximation for the Design of Broadband Beamformers Using Quadratic Programming. *IEEE Signal Processing Letters*, 4(7):103–105, July 1994.

[123] S. Nordholm, I. Claesson, and B. Bengtsson. Adaptive Array Noise Suppression of Handsfree Speaker Input in Cars. *IEEE Transactions on Vehicular Technology*, 42(4):514–518, November 1993.

[124] D. O' Shaughnessy. *Speech Communication, Human and Machine*. Addison–Wesley Publishing Company, 1987.

[125] S. Oh, V. Viswanathan, and P. Papamichalis. Hands–Free Voice Communication in an Automobile With a Microphone Array. In *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP92)*, pages I–281–284, San Francisco, California, March 1992.

[126] A. Oppenheim and R. Schafer. *Digital Signal Processing*. Prentice Hall, Englewood Cliffs, New Jersey, 1975.

[127] H. Özer and S. Gökhun Tanyer. Residual Signal in Sub–band Acoustic Echo Cancellers. In *Proceedings of the European Signal Processing Conference (EUSIPCO98)*, pages 705–708, Rhodes, Greece, September 1998.

[128] J. Páez Borrallo and M. García Otero. On the implementation of a partitioned block frequency domain adaptive filter (PBFDAF) for long acoustic echo cancellation. *Signal Processing*, 27:301–315, June 1992.

[129] P. Peterson. Simulating the response of multiple microphones to a single acoustic source in a reverberant room. *Journal of the Acoustical Society of America*, 80(5):1527–1529, November 1986.

[130] M. Petraglia and R. Alves. New Results on Adaptive Filtering Using Filter Banks. In *Proceedings of the 1997 IEEE International Symposium on Circuits and Systems (ISCAS97)*, pages 2321–2324, Hong Kong, June 1997.

[131] A. Petropulu and S. Subramaniam. Cepstrum Based Deconvolution for Speech Dereverberation. In *Proceedings of the 1994 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP94)*, pages I–9–12, Adelaide, Australia, 1994.

[132] S. Pradhan and V. Reddy. A New Approach to Subband Adaptive Filtering. *IEEE Transactions on Signal Processing*, 47(3):655–664, March 1999.

[133] J. Prado and E. Moulines. Frequency–domain adaptive filtering with applications to acoustic echo cancellation. *Annales des Télécommunations*, 49(7–8):414–428, 1994.

[134] J. Proakis and D. Manolakis. *Digital Signal Processing : Principles, Algorithms and Applications*. Prentice Hall, Upper Saddle River, New Jersey, 1996.

[135] Wainhouse Research. New Research Reports from Wainhouse Research Says Market for Audio, Video, and Web Conferencing Services to Reach $9.8 Billion by 2006, up from $2.8 Billion in 2000. `http://www.wainhouse.com/prcms01v3.html`, January 2002.

[136] J. Rex and S. Elliott. An Optimal Microphone Array for Speech Reception in a Car. In *Proceedings of the European Signal Processing Conference (EUSIPCO94)*, pages 1752–1755, Edinburgh, Scotland, September 1994.

[137] G. Rombouts and M. Moonen. A fast exact frequency domain implementation of the exponentially windowed affine projection algorithm. In *Proceedings of the IEEE International Symposium 2000 on Adaptive Systems for Signal Processing, Communications and Control*, Lake Louise, Alberta, October 2000.

[138] M. Savoji. A Robust Algorithm for Accurate Endpointing of Speech Signals. *Speech Communication*, (8):45–60, March 1989.

[139] D. Schobben. *Efficient Adaptive Multi–Channel Concepts in Acoustics : Blind Signal Separation and Echo Cancelation*. PhD thesis, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, September 1999.

[140] J.-L. Shen, J.-W. Hung, and L.-S. Lee. Robust Entropy–based Endpoint Detection for Speech Recognition in Noisy Environments. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP98)*, pages III–1015–1018, Sydney, Australia, December 1998.

[141] J. Shynk. Adaptive IIR Filtering. *IEEE Acoustics, Speech an Signal Processing Magazine*, pages 4–21, April 1989.

[142] J. Shynk. Frequency–Domain and Multirate Adaptive Filtering. *IEEE Signal Processing Magazine*, 9(1):15–37, January 1992.

[143] J. Sohn, N. Kim, and W. Sung. A Statistical Model–Based Voice Activity Detection. *IEEE Signal Processing Letters*, 6(1):1–3, January 1999.

[144] J. Sohn and W. Sung. A Voice Activity Detector Employing Soft Decision Based Noise Spectrum Adaptation. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP98)*, pages I–365–368, Seattle, Washington, May 1998.

[145] P. Sommen. *Adaptive Filtering Methods*. PhD thesis, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, June 1992.

[146] P. Sommen, P. Van Gerwen, H. Kotmans, and A. Janssen. Convergence Analysis of a Frequency–Domain Adaptive Filter with Exponential Power Averaging and Generalized Window Function. *IEEE Transactions on Circuits and Systems*, 34(7):788–798, July 1987.

[147] J.-S. Soo and K. Pang. Multidelay Block Frequency Domain Adaptive Filter. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(2):373–376, February 1990.

[148] A. Spriet, M. Moonen, and J. Wouters. A multichannel subband GSVD based approach for speech enhancement in hearing aids. In *Proceedings of the 2001 IEEE/EURASIP International Workshop on Acoustic Echo and Noise Control (IWAENC01)*, pages 187–190, Darmstadt, Germany, September 2001.

[149] A. Spriet and K. Vanbleu. Ruisonderdrukking met behulp van iteratieve wiener–filters, master thesis, Katholieke Universiteit Leuven, 1999.

[150] C. Sydow. Broadband beamforming for a microphone array. *Journal of the Acoustical Society of America*, 96(2):845–849, August 1994.

[151] Cellport Systems. Cellport Chosen by Ford to Supply Universal, Hands–Free Cell Phone System. ITS America (`http://www.itsa.org/itsnews.nsf/$All/4B2CEADBF51529A185256A8B004DB772?OpenDocument`), July 2001.

[152] O. Tanrikulu, B. Baykal, A. Constantinides, and J. Chambers. Residual Signal in Sub–band Acoustic Echo Cancellers. In *Proceedings of the European Signal Processing Conference (EUSIPCO96)*, pages 21–24, Trieste, Italy, September 1996.

[153] J. Teerlinck and T. Loones. DSP–implementatie van een echo–cancellation–algoritme, master thesis, Katholieke Universiteit Leuven, 1997.

[154] T. Thompson, K. Chandra, and V. Mehta. Simulation of teleconferencing environments. In *Proceedings of the 1995 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA95)*, New Paltz, New York, October 1995.

[155] V. Turbin, A. Gilloire, and P. Scalart. Comparison of three Post–filtering Algorithms for Residual Acoustic Echo Reduction. In *Proceedings of the 1997 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP97)*, pages 307–310, München, Germany, April 1997.

[156] P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[157] P. Vaidyanathan and Z. Doğanata. The Role of Lossless Systems in Modern Digital Signal Processing : A Tutorial. *IEEE Transactions on Education*, 32(3):181–197, August 1989.

[158] D. Van Compernolle. Switching Adaptive Filters for Enhancing Noisy and Reverberant Speech from Microphone Array Recordings. In *Proceedings of the 1990 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP90)*, pages 833–836, Albuquerque, New Mexico, April 1990.

[159] D. Van Compernolle, T. Claes, F. Xie, and J. Smolders. Measuring the Signal–to–Noise Ration of Noisy Data. Technical Report 94–1, MI2–SPCH, Katholieke Universiteit Leuven, Leuven, Belgium, May 1994.

[160] D. Van Compernolle, J. Smolders, and F. Xie. Noise Suppression for Hands Free Mobile Telephone Communication. In *Proceedings of Inter Noise 93*, pages 1657–1660, Leuven, Belgium, August 1993.

[161] D. Van Compernolle and S. Van Gerven. Beamforming with Microphone Arrays. In V. Cappellini and A. Figueiras–Vidal, editors, *COST 229 : Applications of Digital Signal Processing to Telecommunications*, pages 107–131. 1995.

[162] L. van de Kerkhof and W. Kitzen. Tracking of a Time–Varying Acoustic Impulse Response by an Adaptive Filter. *IEEE Transactions on Signal Processing*, 40(6):1285–1294, June 1992.

[163] A.-J. van der Veen, S. Talwar, and A. Paulraj. Blind Identification of FIR Channels carrying Multiple Finite Alphabet Signals. In *Proceedings of the 1995 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP95)*, pages 1213–1216, Detroit, Michigan, May 1995.

[164] S. Van Gerven, D. Van Compernolle, P. Wauters, W. Verstraeten, K. Eneman, and K. Delaet. Multiple Beam Broadband Beamforming : Filter Design and Real Time Implementation. In *Proceedings of the 1995 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA95)*, New Paltz, New York, October 1995.

[165] B. Van Veen and K. Buckley. Beamforming : A Versatile Approach to Spatial Filtering. *IEEE Magazine on Acoustics, Speech and Signal Processing*, 36(7):953–964, July 1988.

[166] P. Vandaele. *Space–Time Processing Algorithms for Smart Antennas in Wireless Communication Networks.* PhD thesis, Katholieke Universiteit Leuven, Heverlee, Belgium, November 1999.

[167] P. Vandewalle. Dereverberatietechnieken voor de performantieverbetering van spraakgestuurde systemen, master thesis, Katholieke Universiteit Leuven, 2001.

[168] M. Vetterli and Z. Cvetković. Oversampled FIR Filter Banks and Frames in $l^2(\mathbb{Z})$. In *Proceedings of the 1996 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP96)*, pages 1530–1533, Atlanta, Georgia, May 1996.

[169] M. Vetterli and J. Kovačević. *Wavelets and Subband Coding.* Prentice Hall, Englewood Cliffs, New Jersey, 1995.

[170] M. Vetterli and D. Le Gall. Perfect Reconstruction FIR Filter Banks: Some Properties and Factorizations. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(7):1057–1071, July 1989.

[171] Belgisch Instituut voor de Verkeersveiligheid. Voorstelling van de sensibilisatiecampagne 'Rijden en bellen verdeelt je aandacht'. Press conference (`http://www.bivv.be/nl/pdf/0101cRijdenEnBellenIDMSB.pdf`), January 2001.

[172] D. Ward, R. Kennedy, and R. Williamson. Theory and design of broadband sensor arrays with frequency invariant far–field beam patterns. *Journal of the Acoustical Society of America*, 97(2):1023–1034, February 1995.

[173] D. Warnez. Detectie–algoritmen voor echo– en ruisonderdrukkingstoepassingen, master thesis, Katholieke Universiteit Leuven, 2000.

[174] P. Wauters, K. Eneman, K. Delaet, and R. Lauwereins. Adaptive Speech Beamforming Using the TMS320C40 Multi–DSP. Technical Report SPRA305, Texas Instruments Application Note, December 1995.

[175] P. Wauters, S. Van Gerven, M. Engels, and R. Lauwereins. Rapid Prototyping of an Adaptive Speech Beamformer using GRAPE–II. In *Proceedings of the 6–th International Conference on Signal Processing & Technology (ICSPAT95)*, Boston, Massachusetts, October 1995.

[176] R. Wehrmann. Concepts of Improving Hands–Free Speech Communication. In *Proceedings of the 1992 IEEE International Symposium on Circuits and Systems (ISCAS92)*, pages 1918–1921, San Diego, California, May 1992.

[177] S. Weiß, U. Sörgel, and R. Stewart. Computationally Efficient Adaptive System Identification in Subbands with Intersubband Tap Assignment for Undermodelled Problems. In *Proceedings of the 30th Asilomar Conference on Signals, Systems and Computers (ASILOMAR96)*, Pacific Grove, California, November 1996.

[178] B. Widrow and S. Stearns. *Adaptive Signal Processing*. Prentice Hall, Englewood Cliffs, New Jersey, 1985.

[179] GSM World. World Cellular Subscribers. `http://www.gsmworld.com/membership/graph1.html`, June 2001.

[180] J. Wrolstad. Intel Offers Free Tech Support to Telematics Designers. Wireless NewsFactor (`http://wireless.newsfactor.com/perl/story/13835.html`), September 2001.

[181] X.-G. Xia and B. Suter. FIR Paraunitary Filter Banks Given Several Analysis Filters: Factorizations and Constructions. *IEEE Transactions on Signal Processing*, 44(3):720–723, March 1996.

[182] F. Xie and D. Van Compernolle. Speech Enhancement by Spectral Magnitude Estimation — A Unifying Approach. *Speech Communication*, 19(2):89–104, August 1996.

[183] F. Xie and S. Van Gerven. Comparative Study of 3 Speech Detection Methods. Technical Report 95–8, MI2–SPCH, Katholieke Universiteit, Leuven, Leuven, Belgium, October 1995.

# Appendix

## A Some definitions related to matrix algebra

In this appendix a few matrix algebra definitions and properties are combined that will be used and referred to in the thesis. A good reference on matrix algebra that focuses on numerical aspects and implementation issues is [73].

### Hermitian matrix

Matrix $\mathbf{M}$ is hermitian iff $\mathbf{M}^H = (\mathbf{M}^*)^T = \mathbf{M}$. It can be shown that all the eigenvalues of a hermitian matrix are real.

### Unitary matrix

A square complex matrix $\mathbf{M}$ is unitary iff $\mathbf{M}\mathbf{M}^H = \mathbf{M}^H\mathbf{M} = \mathbf{I}$.

### Positive (semi–)definite matrix

If $\mathbf{M}$ is hermitian and

$$\mathbf{x}^H\mathbf{M}\mathbf{x} > 0, \qquad \forall \mathbf{x} \in \mathbb{C}_{\mathbf{0}}^M, \tag{A.1}$$

then $\mathbf{M}$ is a *positive definite* hermitian matrix. All the eigenvalues of a positive definite matrix are strictly positive.

If $\mathbf{M}$ is hermitian and

$$\mathbf{x}^H\mathbf{M}\mathbf{x} \geqslant 0, \qquad \forall \mathbf{x} \in \mathbb{C}_{\mathbf{0}}^M, \tag{A.2}$$

then $\mathbf{M}$ is *positive semi–definite*. All the eigenvalues of a positive semi–definite matrix are positive.

### Toeplitz matrix

An $M \times N$ matrix $\mathbf{T}$ has the Toeplitz structure if

$$\mathbf{T}(m+1, n+1) = \mathbf{T}(m, n) \tag{A.3}$$

for all values $m = 0 : M-2$ and $n = 0 : N-2$, i.e. $\mathbf{T}$ is constant along each of its diagonals.

### Circulant matrix

An $M \times N$ matrix $\mathbf{T}$ is (right–)circulant if $\mathbf{T}$ is Toeplitz and

$$\mathbf{T}(m+1,0) = \mathbf{T}(m,N-1) \tag{A.4}$$

for all values $m = 0 : M-2$.

If $\mathbf{T}$ is a square ($M = N$) right–circulant matrix, $\mathbf{FTF}^{-1}$ is diagonal, where $\mathbf{F}$ is the $M$–dimensional DFT matrix. Hence, $\mathbf{F}^{-1}$ are the eigenvectors of $\mathbf{T}$. It can furthermore be proven that the eigenvalues, i.e. the diagonal elements of $\mathbf{FTF}^{-1}$, are the DFT coefficients of the first column of $\mathbf{T}$.

### Pseudo–circulant matrix

An $M \times N$ rational matrix[1] $\mathbf{T}(z)$ is called pseudo–circulant if for all $m = 0 : M-2$ and $n = 0 : N-2$

$$[\mathbf{T}(z)]_{m+1,n+1} = [\mathbf{T}(z)]_{m,n} \tag{A.5}$$

and

$$[\mathbf{T}(z)]_{m+1,0} = z^{-1}[\mathbf{T}(z)]_{m,N-1}. \tag{A.6}$$

### Paraconjugate

$$\tilde{\mathbf{H}}(z) = \mathbf{H}_*^T(z^{-1}) \tag{A.7}$$

is the paraconjugate of $\mathbf{H}(z)$.

### Para–unitary matrix

An $M \times N$ rational matrix $\mathbf{H}(z)$ is para–unitary if

$$\tilde{\mathbf{H}}(z)\mathbf{H}(z) = c\mathbf{I}, \qquad c \in \mathbb{R}^+ \setminus \{0\}. \tag{A.8}$$

If $\mathbf{H}(z)$ is a square para–unitary polynomial matrix, $\frac{1}{c}\tilde{\mathbf{H}}(z)$ is the FIR inverse of $\mathbf{H}(z)$ and hence

$$\det \mathbf{H}(z) = \alpha z^{-\delta}, \quad \alpha \in \mathbb{R},\ \alpha \neq 0, \quad \delta \in \mathbb{Z}. \tag{A.9}$$

### Unimodular matrix

An $M \times M$ rational matrix $\mathbf{H}(z)$ is unimodular if the determinant of $\mathbf{H}(z)$ is a non–zero constant.

[1] A *rational matrix* is a matrix whose elements are rational transfer functions, i.e. a fraction of two polynomials. A *polynomial matrix* is a matrix whose elements are polynomials.

**Singular value decomposition**

The singular value decomposition of an $M \times N$ complex matrix $\mathbf{M}$ is given by

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H \tag{A.10}$$

where $\mathbf{U}$ is an $M \times M$ unitary matrix containing the left singular vectors, $\mathbf{V}$ is an $N \times N$ unitary matrix with the right singular vectors and $\mathbf{\Sigma}$ is an $M \times N$ pseudo–diagonal matrix containing $\sigma_n$, the $\min\{M, N\}$ singular values of $\mathbf{M}$ :

$$\begin{cases} \mathbf{\Sigma} = \begin{bmatrix} \mathbf{D} \\ \mathbf{0}_{(M-N)\times N} \end{bmatrix} & \text{if } M \geqslant N, \\ \mathbf{\Sigma} = \begin{bmatrix} \mathbf{D} & \mathbf{0}_{M\times(N-M)} \end{bmatrix} & \text{if } M < N, \end{cases} \qquad \mathbf{D} = \operatorname{diag}\{\sigma_n\}, \quad \sigma_n \in \mathbb{R}^+. \tag{A.11}$$

**Pseudo–inverse**

The pseudo–inverse of $\mathbf{M}$ is defined as

$$\mathbf{M}^\dagger = \mathbf{V}\mathbf{\Sigma}^\dagger\mathbf{U}^H \tag{A.12}$$

where

$$\begin{cases} \mathbf{\Sigma}^\dagger = \begin{bmatrix} \mathbf{D}^\dagger & \mathbf{0}_{N\times(M-N)} \end{bmatrix} & \text{if } M \geqslant N \\ \mathbf{\Sigma}^\dagger = \begin{bmatrix} \mathbf{D}^\dagger \\ \mathbf{0}_{(N-M)\times M} \end{bmatrix} & \text{if } M < N \end{cases} \tag{A.13}$$

$\mathbf{D}^\dagger$ is a diagonal matrix whose $n$–th diagonal element equals

$$\mathbf{D}^\dagger(n,n) = \begin{cases} \dfrac{1}{\sigma_n} & \text{if } \sigma_n > 0 \\ 0 & \text{if } \sigma_n = 0 \end{cases} \tag{A.14}$$

with $\sigma_n$ the $n$–th singular value of $\mathbf{M}$ and $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$ the singular value decomposition of $\mathbf{M}$.

**Smith–McMillan decomposition**

The Smith–McMillan decomposition of an $M \times N$ rational matrix $\mathbf{H}(z)$ is given by

$$\mathbf{H}(z) = \mathbf{R}(z)\mathbf{\Sigma}(z)\mathbf{T}(z) \tag{A.15}$$

where $\mathbf{R}(z)$ and $\mathbf{T}(z)$ are respectively $M \times M$ and $N \times N$ unimodular matrices. Further, $\mathbf{\Sigma}(z)$ is an $M \times N$ pseudo–diagonal polynomial matrix, which can be expressed as

$$\mathbf{\Sigma}(z) = \begin{bmatrix} \mathbf{D}(z) \\ \mathbf{0}_{(M-N)\times N} \end{bmatrix} \tag{A.16}$$

if $M \geqslant N$, and as

$$\boldsymbol{\Sigma}(z) = \begin{bmatrix} \mathbf{D}(z) & \mathbf{0}_{M\times(N-M)} \end{bmatrix} \tag{A.17}$$

if $M < N$, with

$$\mathbf{D}(z) = \begin{bmatrix} d_0(z) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & d_{\min\{M,N\}-1}(z) \end{bmatrix}. \tag{A.18}$$

It can be proven [156] that $d_i(z)$ is a factor of $d_{i+1}(z)$ for $0 \leqslant i < \min\{M, N\} - 1$.

# B Appendix to part I

## B.1 Proof of theorem 3.1

**Theorem 3.1** *$\boldsymbol{B}(z)$ is a structured $M \times N$ polynomial matrix, containing polyphase components of the analysis prototype filter $h_0$. Element $(m,n)$ of $\boldsymbol{B}(z)$ is given by :*

$$\begin{array}{ll} [\mathbf{B}(z)]_{m,n} = z^{-q} H_{0_{(n+qN):K}}(z^Q) & \text{if } (m-n) \bmod g = 0 \\ = 0 & \text{if } (m-n) \bmod g \neq 0 \end{array} \tag{B.1}$$

*in which $0 \leqslant m \in \mathbb{N} < M$ and $0 \leqslant n \in \mathbb{N} < N$. Further, $(n+qN)$ mod $M = m$, from which $q$ can be computed, $g = gcd(M,N)$, $Q = \frac{M}{g}$ and $K = \frac{MN}{g}$. $H_{0_{k:K}}(z)$ is the $k$–th out of $K$ polyphase components of the prototype filter $h_0[k]$.*

**Proof :** Starting from Eq. 3.1 and using Eq. 2.17, element $(m,n)$ of $\mathbf{B}(z)$ can be computed :

$$[\mathbf{B}(z)]_{m,n} = \frac{1}{M} \sum_{\mu=0}^{M-1} e^{j\frac{2\pi m\mu}{M}} H_{\mu_{n:N}}(z). \tag{B.2}$$

Define $K$ as the least common multiple of $M$ and $N$, then $K = PM = QN$ and $P$ and $Q$ are coprime. Now as

$$h_\mu[n+Nk] = \begin{cases} h_\mu[n+K\kappa] & \text{if } K\kappa = Nk \\ h_\mu[n+N+K\kappa] & \text{if } N+K\kappa = Nk \\ \cdots & \\ h_\mu[n+(Q-1)N+K\kappa] & \text{if } (Q-1)N+K\kappa = Nk, \end{cases} \tag{B.3}$$

with $n,k,\kappa \in \mathbb{N}$ and $h_\mu$ the $\mu$–th analysis filter, then

$$H_{\mu_{n:N}}(z) = \sum_{q=0}^{Q-1} z^{-q} H_{\mu_{(n+qN):K}}(z^Q). \tag{B.4}$$

Using Eqs. 2.2, 2.9 and 2.10 it can be shown that for the $\mu$–th analysis filter of a DFT modulated filter bank

$$h_\mu[\nu+K\kappa] = h_0[\nu+K\kappa] e^{-j\frac{2\pi(\nu+K\kappa)\mu}{M}}, \qquad 0 \leqslant \nu \in \mathbb{N} < K \tag{B.5}$$

and that its $z$–transform is given by

$$H_{\mu_{\nu:K}}(z) = e^{-j\frac{2\pi\nu\mu}{M}} H_{0_{\nu:K}}(e^{j\frac{2\pi\mu K}{M}} z) = e^{-j\frac{2\pi\nu\mu}{M}} H_{0_{\nu:K}}(z). \tag{B.6}$$

Taking into account Eq. B.6, equation B.4 can be rewritten as

$$H_{\mu_{n:N}}(z) = \sum_{q=0}^{Q-1} z^{-q} e^{-j\frac{2\pi(n+qN)\mu}{M}} H_{0_{(n+qN):K}}(z^Q). \tag{B.7}$$

Plugging in equation B.7 in Eq. B.2 :

$$[\mathbf{B}(z)]_{m,n} = \frac{1}{M}\sum_{\mu=0}^{M-1} e^{j\frac{2\pi m\mu}{M}} \sum_{q=0}^{Q-1} z^{-q} e^{-j\frac{2\pi(n+qN)\mu}{M}} H_{0_{(n+qN):K}}(z^Q) \tag{B.8}$$

$$= \frac{1}{M}\sum_{q=0}^{Q-1} z^{-q} H_{0_{(n+qN):K}}(z^Q) \sum_{\mu=0}^{M-1} e^{-j\frac{2\pi(n-m+qN)\mu}{M}}. \tag{B.9}$$

In section B.2 it will be shown that there exists at most one value $q \in \mathbb{N}$ smaller than $Q$ for which

$$\sum_{\mu=0}^{M-1} e^{-j\frac{2\pi(n-m+qN)\mu}{M}} \tag{B.10}$$

differs from 0 and that this unique value of $q$ exists if $m-n$ is a multiple of $g = \gcd(M,N)$. In that case $q$ can be found from

$$n+qN = m+pM, \qquad \begin{cases} 0 \leqslant q \in \mathbb{N} < Q \\ 0 \leqslant p \in \mathbb{N} < P \end{cases} \tag{B.11}$$

If condition B.11 is fulfilled, Eq. B.10 equals $M$ and due to the uniqueness of $q$ equation B.9 reduces to

$$[\mathbf{B}(z)]_{m,n} = z^{-q} H_{0_{(n+qN):K}}(z^Q), \qquad (n+qN) \bmod M = m \tag{B.12}$$

provided $m-n$ is a multiple of $g$. This completes the proof. □

## B.2 Properties of $\mathbf{B}(z)$

In section 3.1.1 and section B.1 it was shown that $\mathbf{H}(z)$ can be factorized as $\mathbf{H}(z) = \mathbf{F}\mathbf{B}(z)$, with $\mathbf{F}$ an $M \times M$ DFT matrix and $\mathbf{B}(z)$ defined as (Eq. B.9 is repeated here) :

$$[\mathbf{B}(z)]_{m,n} = \frac{1}{M}\sum_{q=0}^{Q-1} z^{-q} H_{0_{(n+qN):K}}(z^Q) \sum_{\mu=0}^{M-1} e^{-j\frac{2\pi(n-m+qN)\mu}{M}}. \tag{B.13}$$

$K$ is the least common multiple of $M$ and $N$, $K = PM = QN$ and $P$ and $Q$ are coprime. In order to simplify Eq. B.13 the last term

$$\sum_{\mu=0}^{M-1} e^{-j\frac{2\pi(n-m+qN)\mu}{M}} \tag{B.14}$$

is analyzed in more detail. In general we can state that

$$\sum_{m=0}^{M-1} e^{-j\frac{2\pi mn}{M}} = \sum_{m=0}^{M-1} e^{0} = M \qquad \text{if } n = pM,\ p \in \mathbb{Z} \tag{B.15}$$

and as it is a geometric series,

$$\sum_{m=0}^{M-1} e^{-j\frac{2\pi mn}{M}} = \frac{e^{-j\frac{2\pi Mn}{M}} - 1}{e^{-j\frac{2\pi n}{M}} - 1} = 0 \qquad \text{if } \nexists p \in \mathbb{Z} : n = pM. \tag{B.16}$$

Hence, Eq. B.14 will be different from 0 if a value $p \in \mathbb{Z}$ can be found such that

$$n + qN = m + pM, \qquad m, n, q \in \mathbb{N}, \tag{B.17}$$

with $0 \leqslant m < M$, $0 \leqslant n < N$ and $0 \leqslant q < Q$. Equation B.17 is equivalent to

$$(n + qN) \bmod M = m. \tag{B.18}$$

As $q < Q$, $m < M$ and $n < N$, it follows that $n+qN < QN = K = PM$. Therefore $p$ in Eq. B.17 has to be smaller than $P$.

If $g$ is the greatest common divisor of $M$ and $N$, then $N = \nu_g g$ and $M = \mu_g g$, with $\nu_g$ and $\mu_g$ coprime. As $K$ is the least common multiple of $M$ and $N$, $K = \frac{MN}{g}$. This also implies that $P = \frac{N}{g} = \nu_g$ and $Q = \frac{M}{g} = \mu_g$ and hence also $P$ and $Q$ are coprime.

**Theorem B.1** *There is at most one value $q \in \mathbb{N}$ smaller than $Q$ for which*

$$\sum_{\mu=0}^{M-1} e^{-j\frac{2\pi(n-m+qN)\mu}{M}} \tag{B.19}$$

*in Eq. B.13 differs from 0.*

**Proof:** Suppose there are two values for $q$ for which

$$\sum_{\mu=0}^{M-1} e^{-j\frac{2\pi m(n-m+qN)}{M}} \neq 0. \tag{B.20}$$

Then from Eq. B.17 it follows

$$\begin{aligned} n + q_1 N = m + p_1 M \quad & \& \quad n + q_2 N = m + p_2 M \\ & \Downarrow \\ (q_1 - q_2)N \quad &= \quad (p_1 - p_2)M. \end{aligned} \tag{B.21}$$

Dividing Eq. B.21 by $g$ leads to

$$(q_1 - q_2)P = (p_1 - p_2)Q. \tag{B.22}$$

As $P$ and $Q$ are coprime,

$$q_1 = q_2 + sQ \quad \& \quad p_1 = p_2 + tP, \qquad \text{with } s, t \in \mathbb{Z}. \tag{B.23}$$

As $0 \leqslant q_1, q_2 \in \mathbb{N} < Q$ and $0 \leqslant p_1, p_2 \in \mathbb{N} < P$, Eq. B.23 can only hold for $s = t = 0$. So, $p$ and $q$ in $n + qN = m + pM$ must be unique. □

**Theorem B.2** *Element $(m,n)$ of $\boldsymbol{B}(z)$ can only be different from 0 if $m-n$ is a multiple of $g$.*

**Proof:** Theorem B.1 proofs that there is at most one $q$ for which Eq. B.19 differs from 0, i.e. at most one $K$–th order polyphase component of $h_0$ can contribute to $[\mathbf{B}(z)]_{m,n}$. In order to find out for which $(m,n)$ Eq. B.19 differs from 0, we check if there exist $p,q \in \mathbb{Z}$ such that

$$\begin{aligned} n+qN &= m+pM && \text{(B.24)} \\ &\Longleftrightarrow \\ n+q\nu_g g &= m+p\mu_g g && \text{(B.25)} \\ &\Longleftrightarrow \\ m-n &= (q\nu_g - p\mu_g)g, && \text{(B.26)} \end{aligned}$$

in other words, $m-n$ must be divisible by $g$. In that case $[\mathbf{B}(z)]_{m,n}$ will be different from zero. □

**Theorem B.3** *Entry $(m,n)$ of $\boldsymbol{B}(z)$ contains the $k$–th (out of $K$) polyphase component of the prototype $h_0$ if $m = k \bmod M$ and $n = k \bmod N$.*

**Proof:** From Eqs. B.12 and B.17 it follows that if the $k$–th polyphase component of $h_0$ contributes to $[\mathbf{B}(z)]_{m,n}$,

$$k = n+qN = m+pM. \tag{B.27}$$

Now, every $0 \leqslant k \in \mathbb{N} < K$ can be decomposed in an integer part and remainder modulo $M$ or $N$. At most one polyphase component $k$ can contribute to entry $(m,n)$ (theorem B.1) and as decomposition B.27 is unambiguous for $m,n,p,q,M,N \in \mathbb{N}$ and $n < N$ and $m < M$, each polyphase filter has its unique entry $(m,n)$. All $K$ polyphase filters are located at different places. Hence, $\mathbf{B}(z)$ has $MN$ entries of which only $K = \frac{MN}{g}$ can be non–zero, i.e. all entries for which $m-n$ is divisible by $g$. □

## B.3 Proof of theorem 3.2

**Theorem 3.2** *If an $M$–band, $N$–fold downsampled filter bank set satisfies the perfect reconstruction condition, perfect reconstruction will also hold for $\breve{N} = \frac{N}{p}$, with $N,\breve{N},p \in \mathbb{N}_0$. Furthermore, it is found that the output/input level increases by a factor $p$ when $N$ is divided by $p$.*

**Proof:** Consider figure B.1, which shows a general subband scheme. The output $y$ can now be written as a function of the input $x$ in the $z$–domain.

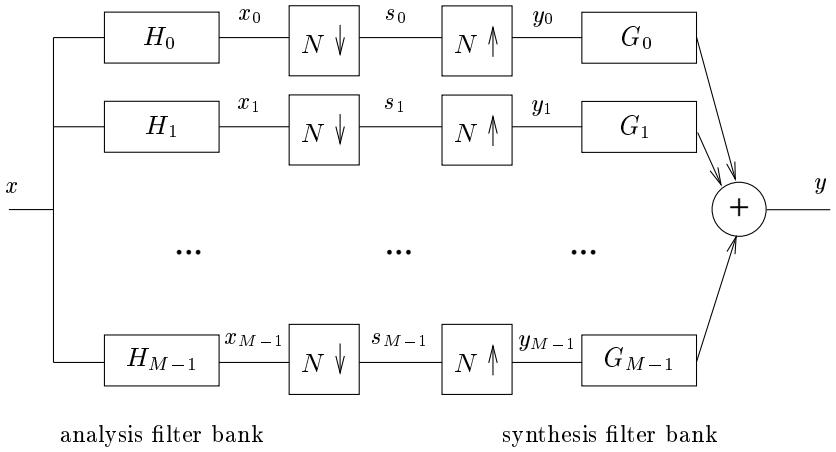


Figure B.1: A general subband scheme

As $X_m(z) = H_m(z)X(z)$ the $m$–th subband signal

$$S_m(z) = \frac{1}{N}\sum_{n=0}^{N-1} H_m(z^{1/N}e^{-j\frac{2\pi n}{N}})X(z^{1/N}e^{-j\frac{2\pi n}{N}}) \tag{B.28}$$

is found using Eq. 2.6. Hence, following Eq. 2.8

$$Y_m(z) = \frac{1}{N}\sum_{n=0}^{N-1} H_m(ze^{-j\frac{2\pi n}{N}})X(ze^{-j\frac{2\pi n}{N}}) \tag{B.29}$$

is obtained such that finally

$$\begin{aligned} Y(z) &= \frac{1}{N}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} G_m(z)H_m(ze^{-j\frac{2\pi n}{N}})X(ze^{-j\frac{2\pi n}{N}}) &\text{(B.30)}\\ &= \frac{1}{N}\sum_{n=0}^{N-1} X(ze^{-j\frac{2\pi n}{N}})\left(\sum_{m=0}^{M-1} G_m(z)H_m(ze^{-j\frac{2\pi n}{N}})\right). &\text{(B.31)} \end{aligned}$$

$X(ze^{-j\frac{2\pi n}{N}})$ with $n > 0$ are aliased versions of $X(z)$. A perfect reconstruction system is obtained if $Y(z) = z^{-\psi}X(z)$, $\psi \in \mathbb{N}$, i.e. if

$$\sum_{m=0}^{M-1} G_m(z)H_m(z) = Nz^{-\psi} \tag{B.32}$$

$$\sum_{m=0}^{M-1} G_m(z)H_m(ze^{-j\frac{2\pi n}{N}}) = 0, \qquad \text{for } n = 1 : N-1 \tag{B.33}$$

are both fulfilled.

If we assume that Eqs. B.32 and B.33 are satisfied for $N$, it should now be proven that they are also valid for $\breve{N} = \frac{N}{p}$, with $N, \breve{N}, p \in \mathbb{N}_0$, i.e. for all divisors of $N$.

It is found that for $\breve{N}$–fold downsampling

$$Y(z) = \frac{1}{\breve{N}} \sum_{\breve{n}=0}^{\breve{N}-1} X(ze^{-j\frac{2\pi\breve{n}}{\breve{N}}}) \left( \sum_{m=0}^{M-1} G_m(z) H_m(ze^{-j\frac{2\pi\breve{n}}{\breve{N}}}) \right) \tag{B.34}$$

$$= \frac{p}{N} \sum_{\breve{n}=0}^{N/p-1} X(ze^{-j\frac{2\pi\breve{n}p}{N}}) \left( \sum_{m=0}^{M-1} G_m(z) H_m(ze^{-j\frac{2\pi\breve{n}p}{N}}) \right) \tag{B.35}$$

$$= pX(z) \tag{B.36}$$

as Eq. B.32 is still valid and

$$\sum_{m=0}^{M-1} G_m(z) H_m(ze^{-j\frac{2\pi\breve{n}p}{N}}) = 0, \qquad \text{for } \breve{n} = 1 : \frac{N}{p} - 1, \tag{B.37}$$

because Eq. B.33 is valid for $n = \breve{n}p$, i.e. $n = p, 2p, \ldots, N - p$. □

## B.4 Proof of theorem 3.3

**Theorem 3.3** *After appropriate row and column permutations* $\boldsymbol{P}$ *and* $\boldsymbol{Q}$ *the analysis polynomial matrix* $\boldsymbol{B}(z)$ *can be transformed into a block diagonal matrix*

$$\mathbf{PB}(z)\mathbf{Q} = \begin{bmatrix} \mathbf{S}_0(z) & \ldots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \ldots & \mathbf{S}_{g-1}(z) \end{bmatrix} \tag{B.38}$$

*consisting of* $g$ *dense submatrices* $\boldsymbol{S}_\gamma(z)$ *of dimension* $\frac{M}{g} \times \frac{N}{g}$, *with* $g$ *being the greatest common divisor of* $M$ *and* $N$ *and* $\gamma$ *an integer between 0 and* $g - 1$.

**Proof:** As $g$ is the greatest common divisor of $M$ and $N$,

$$\exists\ \mu_g, \nu_g \in \mathbb{N} : \begin{cases} M = g\mu_g \\ N = g\nu_g \\ \gcd(\mu_g, \nu_g) = 1. \end{cases} \tag{B.39}$$

Now, Eq. B.12 can be written as

$$[\mathbf{B}(z)]_{k,l} = z^{-q} H_{0_{(l+qg\nu_g):K}}(z^Q), \qquad l + qg\nu_g = k + pg\mu_g \tag{B.40}$$

$$= z^{-q} H_{0_{(\gamma+g(\lambda_g+q\nu_g)):K}}(z^Q), \quad \gamma + g(\lambda_g + q\nu_g) = \gamma + g(\lambda_g + \alpha + p\mu_g) \tag{B.41}$$

using the fact that $l$ can be decomposed as $\gamma+g\lambda_g$ and that $k-l=\alpha g$ (see theorem B.2) with $\gamma,\lambda_g\in\mathbb{N}$, $\gamma<g$ and $\alpha\in\mathbb{Z}$. Equation B.41 can be rewritten as

$$[\mathbf{B}(z)]_{k,l}=z^{-q}H_{0_{(\gamma+g(\lambda_g+q\nu_g)):g\mu_g\nu_g}}(z^{\mu_g}),\qquad \lambda_g+q\nu_g=\kappa_g+p\mu_g \tag{B.42}$$

defining $k=(\gamma+g\lambda_g)+\alpha g=\gamma+g\kappa_g$ and using the fact that $K=\frac{MN}{g}$ and $Q=\frac{M}{g}$.

By comparing Eq. B.42 with B.12 it is observed that if $k-l$ is a multiple of $g$, element $(k,l)$ of $\mathbf{B}(z)$ corresponds to the $(\kappa_g,\lambda_g)$–th element of an $\frac{M}{g}\times\frac{N}{g}$ submatrix $\mathbf{S}_\gamma(z)$

$$[\mathbf{S}_\gamma(z)]_{\kappa_g,\lambda_g}=z^{-q}\hat{H}^{(\gamma)}_{0_{(\lambda_g+q\nu_g)}}(z^{\mu_g}),\qquad \lambda_g+q\nu_g=\kappa_g+p\mu_g \tag{B.43}$$

with

$$\hat{H}^{(\gamma)}_{0_{(\lambda_g+q\nu_g)}}(z)=H_{0_{(\gamma+g(\lambda_g+q\nu_g)):g\mu_g\nu_g}}(z). \tag{B.44}$$

As $\frac{M}{g}$ and $\frac{N}{g}$ are coprime, $\mathbf{S}_\gamma(z)$ will be a dense matrix, i.e. all elements contain a single polyphase component of $\mathbf{B}(z)$. Submatrix $\mathbf{S}_0(z)$ for instance can be regarded as the analysis polyphase matrix of a DFT modulated filter bank with $\frac{M}{g}$, $\frac{N}{g}$–fold downsampled subbands that contains polyphase component $H_{0_{gr:K}}(z)$ with $0\leqslant r<\mu_g\nu_g$. Submatrices $\mathbf{S}_\gamma(z)$ for which $\gamma>0$ will have the same overall form and equal delay structure as $\mathbf{S}_0(z)$. The polyphase component indices however are augmented with $\gamma$ (see Eq. 3.58).

All elements on position $(k,l)$ in $\mathbf{B}(z)$ whose column coordinate $l$ belongs to the same remainder class $\gamma$ (with respect to $g$), are gathered in a single dense $\frac{M}{g}\times\frac{N}{g}$ matrix $\mathbf{S}_\gamma(z)$. In this way the total of $\frac{MN}{g}$ non–zero elements of $\mathbf{B}(z)$ can be divided in $g$ subgroups, which can be shifted towards each other, resulting in the decomposition presented in Eq. B.38. □

## B.5 Proof of theorem 3.4

**Theorem 3.4** *By pre– and post–multiplication of $\boldsymbol{S}_\gamma(z)$ with appropriate para–unitary diagonal matrices $\boldsymbol{L}_\gamma(z)$ and $\boldsymbol{R}_\gamma(z)$*

$$\hat{\mathbf{S}}_\gamma(z^Q)=\mathbf{L}_\gamma(z)\mathbf{S}_\gamma(z)\mathbf{R}_\gamma(z), \tag{B.45}$$

*can be obtained, i.e. all elements of $\hat{\boldsymbol{S}}_\gamma(z^Q)$ depend on $z^Q$.*

**Proof:** Consider the following procedure, which is applicable to any of the submatrices $\mathbf{S}_\gamma(z)$, defined by Eq. B.43. As all $\mathbf{S}_\gamma(z)$ have an equal delay structure the same decomposition holds for all submatrices $\mathbf{S}_\gamma(z)$.

Applying the following steps to $\mathbf{S}_\gamma(z)$ will result in a matrix $\hat{\mathbf{S}}_\gamma(z^Q)$ depending on $z^Q$.

1. Multiply each column $l$ with $z^{q_l}$ so to get a first row containing anything but zero delays. Referring to Eq. B.12, this comes down to looking for appropriate integers $q_l$ and $p_l$ such that

$$l + q_l \frac{N}{g} = p_l \frac{M}{g}. \tag{B.46}$$

In this way element $(0, l)$ will be made delayless. Hence all elements on the first row of

$$\mathbf{S}_\gamma(z) \underbrace{\begin{bmatrix} z^{q_0} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & z^{q_{N/g-1}} \end{bmatrix}}_{\mathbf{R}_\gamma(z)} \tag{B.47}$$

have zero delay.

2. For each row $k$ now values $p_{kl}, q_{kl} \in \mathbb{N}$ can be found such that

$$l + q_{kl} \frac{N}{g} = k + p_{kl} \frac{M}{g}. \tag{B.48}$$

Combining Eq. B.48 with Eq. B.46 leads to

$$(q_{kl} - q_l) \frac{N}{g} = k + (p_{kl} - p_l) \frac{M}{g}. \tag{B.49}$$

As column $l$ has been multiplied with the appropriate $z^{q_l}$, the delay corresponding to element $(k, l)$ now equals $\Delta q_{kl} = q_{kl} - q_l$. From Eq. B.49 it follows that

$$\Delta q_{kl} = \frac{(k + (p_{kl} - p_l))M}{N}. \tag{B.50}$$

The same reasoning holds for any other column $\bar{l}$ on the same row. Hence,

$$\Delta q_{kl} - \Delta q_{k\bar{l}} = \frac{((p_{kl} - p_l) - (p_{k\bar{l}} - p_{\bar{l}}))M}{N}. \tag{B.51}$$

$(\Delta q_{kl} - \Delta q_{k\bar{l}}) \in \mathbb{Z}$, so $p_{kl} - p_l - p_{k\bar{l}} + p_{\bar{l}}$ must be a multiple of $\frac{N}{g}$, and hence $\Delta q_{kl} - \Delta q_{k\bar{l}}$ is a multiple of $\frac{M}{g} = Q$.

3. Now for each row $k$, find the appropriate

$$\delta_k = \min_{\text{all columns } l} \{\Delta q_{kl}\} \tag{B.52}$$

such that the minimum delay on each row of

$$\hat{\mathbf{S}}_\gamma(z^Q) = \underbrace{\begin{bmatrix} z^{\delta_0} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & z^{\delta_{M/g-1}} \end{bmatrix}}_{\mathbf{L}_\gamma(z)} \mathbf{S}_\gamma(z) \underbrace{\begin{bmatrix} z^{q_0} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & z^{q_{N/g-1}} \end{bmatrix}}_{\mathbf{R}_\gamma(z)} \tag{B.53}$$

is zero. For all $0 \leqslant l, \bar{l} \in \mathbb{N} < \frac{N}{g}$ it was proven that $\Delta q_{kl} - \Delta q_{k\bar{l}}$ is a multiple of $Q$. Further, there is at least one filter having delay 0 on each row and taking into account that the structure of the polynomial elements of $\mathbf{S}_\gamma(z)$ is given by Eq. B.43, it follows that $\hat{\mathbf{S}}_\gamma$ will depend on $z^Q$.

The procedure has been applied to example 3.1. The results are shown in Eq. 3.60.

It can easily be verified that $\mathbf{L}_\gamma(z)$ and $\mathbf{R}_\gamma(z)$ are para–unitary diagonal matrices. So, if $\hat{\mathbf{S}}_\gamma(z^Q)$ is para–unitary also $\mathbf{S}_\gamma(z)$ will be a para–unitary polynomial matrix. □

## B.6 Inverse decomposition of para–unitary lattices

Whereas in the decomposition problem (Eq. 3.61) $\mathbf{M}_\gamma(z)$ is parameterized such that the underlying filter bank is para–unitary, inverse decomposition deals with finding the unit–norm vectors $\mathbf{v}_l$ and $\mathbf{u}_l$ for a given $\mathbf{M}_\gamma(z)$.

The polyphase components of the prototype filter of a para–unitary filter bank can be put into submatrices $\hat{\mathbf{S}}_\gamma(z)$ (Eq. 3.59). Now, if $M > N$ each of the para–unitary submatrices $\hat{\mathbf{S}}_\gamma(z)$ is rectangular and is embedded in a larger square $\frac{M}{g} \times \frac{M}{g}$ matrix $\mathbf{M}_\gamma(z)$. We repeat formula 3.61 :

$$\mathbf{M}_\gamma^{(L_\gamma)}(z) = \mathbf{V}_{L_\gamma}(z)\mathbf{V}_{L_\gamma-1}(z)\ldots\mathbf{V}_1(z)\mathbf{U}. \tag{B.54}$$

$\mathbf{V}_l(z)$ are degree–1 para–unitary building blocks :

$$\mathbf{V}_l(z) = \mathbf{I} - \mathbf{v}_l\mathbf{v}_l^T + z^{-1}\mathbf{v}_l\mathbf{v}_l^T \tag{B.55}$$

and $\mathbf{U}^T\mathbf{U} = \mathbf{I}$. Further, $\mathbf{v}_l$ are $\frac{M}{g} \times 1$ column vectors having unit norm.

So,

$$\hat{\mathbf{S}}_\gamma(z) = \bar{\mathbf{M}}_\gamma^{(L_\gamma)}(z) = \mathbf{V}_{L_\gamma}(z)\mathbf{V}_{L_\gamma-1}(z)\ldots\mathbf{V}_1(z)\mathbf{U}|_{1:\frac{N}{g}} \tag{B.56}$$

in which $\mathbf{U}|_{1:\frac{N}{g}}$ contains the first $\frac{N}{g}$ columns of $\mathbf{U}$. By introducing

$$\bar{\mathbf{M}}_\gamma^{(L_\gamma-1)}(z) = \mathbf{V}_{L_\gamma-1}(z)\ldots\mathbf{V}_1(z)\mathbf{U}|_{1:\frac{N}{g}} \tag{B.57}$$

it appears that

$$\bar{\mathbf{M}}_\gamma^{(L_\gamma)}(z) = \mathbf{V}_{L_\gamma}(z)\bar{\mathbf{M}}_\gamma^{(L_\gamma-1)}(z). \tag{B.58}$$

From Eqs. B.55 and B.58 it follows that the $z^{-L_\gamma}$–part of $\bar{\mathbf{M}}_\gamma^{(L_\gamma)}(z)$ is

$$\mathbf{v}_{L_\gamma}\mathbf{v}_{L_\gamma}^T \cdot \left(z^{-(L_\gamma-1)}\text{–part of } \bar{\mathbf{M}}_\gamma^{(L_\gamma-1)}(z)\right), \tag{B.59}$$

which is a rank–1 matrix. Therefore $\mathbf{v}_{L_\gamma}$ can be found as the first left singular vector of the $z^{-L_\gamma}$–part of $\bar{\mathbf{M}}_\gamma^{(L_\gamma)}(z)$. The next step will be the recovery of $\bar{\mathbf{M}}_\gamma^{(L_\gamma-1)}(z)$.

Consider the general case where $\mathbf{v}_l$ and $\bar{\mathbf{M}}_\gamma^{(l)}(z)$ are known and $\bar{\mathbf{M}}_\gamma^{(l-1)}(z)$ needs to be found. Define $\mathbf{A}_l = \mathbf{I} - \mathbf{v}_l\mathbf{v}_l^T$ and $\mathbf{B}_l = \mathbf{v}_l\mathbf{v}_l^T$. Now from Eq. B.55,

$$\bar{\mathbf{M}}_\gamma^{(l)}(z) = (\mathbf{A}_l + z^{-1}\mathbf{B}_l)\bar{\mathbf{M}}_\gamma^{(l-1)}(z). \tag{B.60}$$

Define

$$\bar{\mathbf{M}}_\gamma^{(l)}(z) = \sum_{k=0}^{l} \bar{\mathbf{H}}_l^{(k)} z^{-k} \tag{B.61}$$

then

$$\sum_{k=0}^{l} \bar{\mathbf{H}}_l^{(k)} z^{-k} = (\mathbf{A}_l + z^{-1}\mathbf{B}_l) \sum_{k=0}^{l-1} \bar{\mathbf{H}}_{l-1}^{(k)} z^{-k}. \tag{B.62}$$

If it is assumed that $\bar{\mathbf{H}}_l^{(k)} = 0$ for $k < 0$ and $k > l$, then

$$\bar{\mathbf{H}}_l^{(k)} = \mathbf{A}_l\bar{\mathbf{H}}_{l-1}^{(k)} + \mathbf{B}_l\bar{\mathbf{H}}_{l-1}^{(k-1)}, \quad \forall k. \tag{B.63}$$

This leads to the following set of linear equations :

$$\begin{bmatrix} \bar{\mathbf{H}}_l^{(0)} \\ \bar{\mathbf{H}}_l^{(1)} \\ \vdots \\ \bar{\mathbf{H}}_l^{(l-1)} \\ \bar{\mathbf{H}}_l^{(l)} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_l & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{B}_l & \mathbf{A}_l & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{A}_l \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{B}_l \end{bmatrix} \begin{bmatrix} \bar{\mathbf{H}}_{l-1}^{(0)} \\ \bar{\mathbf{H}}_{l-1}^{(1)} \\ \vdots \\ \bar{\mathbf{H}}_{l-1}^{(l-1)} \end{bmatrix}. \tag{B.64}$$

$\bar{\mathbf{M}}_\gamma^{(l-1)}(z)$ can therefore be found based on $\bar{\mathbf{M}}_\gamma^{(l)}(z)$ and the pseudo–inverse of the double block diagonal matrix in the previous equation. Again $\mathbf{v}_{l-1}$ is found as the first left singular vector of the $z^{-(l-1)}$–part of matrix $\bar{\mathbf{M}}_\gamma^{(l-1)}(z)$. This procedure can be repeated until a zero–order rectangular orthonormal $\bar{\mathbf{M}}_\gamma^{(0)}(z)$ remains. However, a square orthonormal matrix $\mathbf{U}$ is needed to compute the $\mathbf{u}_l$'s. By QR–decomposition of $\bar{\mathbf{M}}_\gamma^{(0)}(z)$ a set of $\frac{M}{g} - \frac{N}{g}$ column vectors can be computed orthogonal to $\bar{\mathbf{M}}_\gamma^{(0)}(z)$. They can be combined with $\bar{\mathbf{M}}_\gamma^{(0)}(z)$ and serve as the $\frac{M}{g} - \frac{N}{g}$ missing columns to form an orthonormal $\mathbf{U}$.

The decomposition of $\mathbf{U}$ is based on its Householder factorization (Eq. 3.63) :

$$\mathbf{U} = \mathbf{U}_1\mathbf{U}_2 \dots \mathbf{U}_{\frac{M}{g}-1}. \tag{B.65}$$

$\mathbf{U}_l$ are Householder matrices : $\mathbf{U}_l = \mathbf{I} - 2\mathbf{u}_l\mathbf{u}_l^T$ in which $\mathbf{u}_l$ are $\frac{M}{g} \times 1$ column vectors having unit norm. They can be contained in a description matrix having the special format given in Eq. 3.66.

It can be proven (see [156]) that there exists a unit–norm vector $\mathbf{u}_1$ such that

$$\mathbf{U}_1\mathbf{U} = (\mathbf{I} - 2\mathbf{u}_1\mathbf{u}_1^T)\mathbf{U} = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_1 \end{bmatrix}, \qquad \text{with } \mathbf{T}_1^T\mathbf{T}_1 = \mathbf{I}. \tag{B.66}$$

So, with $\mathbf{U}|_1$ being the first column of $\mathbf{U}$,

$$-2\mathbf{u}_1\mathbf{u}_1^T\mathbf{U}|_1 = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} - \mathbf{U}|_1. \tag{B.67}$$

Normalizing both sides

$$\frac{-2\mathbf{u}_1\mathbf{u}_1^T\mathbf{U}|_1}{\sqrt{4\mathbf{U}|_1^T\mathbf{u}_1\mathbf{u}_1^T\mathbf{u}_1\mathbf{u}_1^T\mathbf{U}|_1}} = \frac{\begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} - \mathbf{U}|_1}{\left\| \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} - \mathbf{U}|_1 \right\|_2} \tag{B.68}$$

and taking into account that $\mathbf{u}_1^T\mathbf{u}_1 = 1$ and that $\mathbf{U}|_1^T\mathbf{u}_1$ is a scalar,

$$\mathbf{u}_1 = -\frac{\begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} - \mathbf{U}|_1}{\left\| \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} - \mathbf{U}|_1 \right\|_2} \tag{B.69}$$

is obtained.

From Eq. B.66 now $\mathbf{T}_1$ can be computed. Applying Eqs. B.66 and B.69 recursively to $\mathbf{T}_1$ leads to $\mathbf{u}_2$. $\mathbf{T}_2$ can be computed from $\mathbf{T}_1$ and $\mathbf{u}_2$ using Eq. B.66. This procedure can then further recursively be applied to obtain $\mathbf{u}_3, \ldots, \mathbf{u}_{\frac{M}{g}-1}$.

## B.7 Para–unitary parameterization for $M = 2N$

For the special case of 2–fold subband oversampling, i.e. for $M = 2N$, a more dedicated parameterization can be applied. In this case so–called lattice rotation angles are optimized instead of the parameter sets $(\mathbf{v}_l, \mathbf{u}_l)_\gamma$ as presented in section 3.3.2. A similar procedure can be applied to design perfect reconstruction cosine modulated filter banks [156].

### Para–unitary parameterization based on $2 \times 1$ lossless lattices

If $M = 2N$, $\mathbf{B}(z)$ is a $2N \times N$ matrix :

$$\mathbf{B}(z) = \begin{bmatrix} H_{0_{0:2N}}(z^2) & 0 & \cdots & 0 \\ 0 & H_{0_{1:2N}}(z^2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & H_{0_{N-1:2N}}(z^2) \\ z^{-1}H_{0_{N:2N}}(z^2) & 0 & \cdots & 0 \\ 0 & z^{-1}H_{0_{N+1:2N}}(z^2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & z^{-1}H_{0_{2N-1:2N}}(z^2) \end{bmatrix} \tag{B.70}$$

in which $H_{0_{n:2N}}(z)$ is the $n$–th out of $2N$ polyphase components of the prototype filter $h_0[k]$ (see also Eq. 3.3). After appropriate row permutations a block diagonal matrix

$$\mathbf{PB}(z) = \begin{bmatrix} H_{0_{0:2N}}(z^2) & 0 & \cdots & 0 \\ z^{-1}H_{0_{N:2N}}(z^2) & 0 & \cdots & 0 \\ 0 & H_{0_{1:2N}}(z^2) & \cdots & 0 \\ 0 & z^{-1}H_{0_{N+1:2N}}(z^2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & H_{0_{N-1:2N}}(z^2) \\ 0 & 0 & \cdots & z^{-1}H_{0_{2N-1:2N}}(z^2) \end{bmatrix} \tag{B.71}$$

can be obtained. Hence, perfect reconstruction is guaranteed through para–unitarity of the following set :

$$\left\{ \begin{bmatrix} H_{0_{n:2N}}(z) \\ H_{0_{N+n:2N}}(z) \end{bmatrix} \right\}_{n=0\rightarrow N-1}, \tag{B.72}$$

which comes down to finding a set of $N$ power complementary pairs :

$$\left\{ \tilde{H}_{0_{n:2N}}(z)H_{0_{n:2N}}(z) + \tilde{H}_{0_{N+n:2N}}(z)H_{0_{N+n:2N}}(z) = 1 \right\}_{n=0\rightarrow N-1} \tag{B.73}$$

Power complementarity can be imposed by means of a 2–channel lossless lattices [156].

A causal real–valued $2 \times 1$ lossless FIR system of order $L_n$ can be factorized as

$$\mathbf{M}^{(L_n)}(z) = \mathbf{R}_{L_n}\mathbf{\Lambda}(z)\mathbf{R}_{L_n-1}\ldots\mathbf{R}_1\mathbf{\Lambda}(z) \begin{bmatrix} \cos\theta_0 \\ -\sin\theta_0 \end{bmatrix} \tag{B.74}$$

where $\mathbf{R}_l$ are Givens rotations :

$$\mathbf{R}_l = \begin{bmatrix} \cos\theta_l & \sin\theta_l \\ -\sin\theta_l & \cos\theta_l \end{bmatrix} \tag{B.75}$$

and $\mathbf{\Lambda}(z)$ is a first–order $2 \times 2$ para–unitary building block :

$$\mathbf{\Lambda}(z) = \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix}. \tag{B.76}$$

A lossless lattice $\mathbf{M}^{(L_n)}(z)$ is now assigned to each pair

$$\begin{bmatrix} H_{0_{n:2N}}(z) \\ H_{0_{N+n:2N}}(z) \end{bmatrix}. \tag{B.77}$$

Starting from the lattice rotation angles $\theta_l$ the lattices $\mathbf{M}^{(L_n)}(z)$ can be computed using Eq. B.74. Then from Eq. B.71 $\mathbf{B}(z)$ and the prototype $h_0[k]$ can easily be

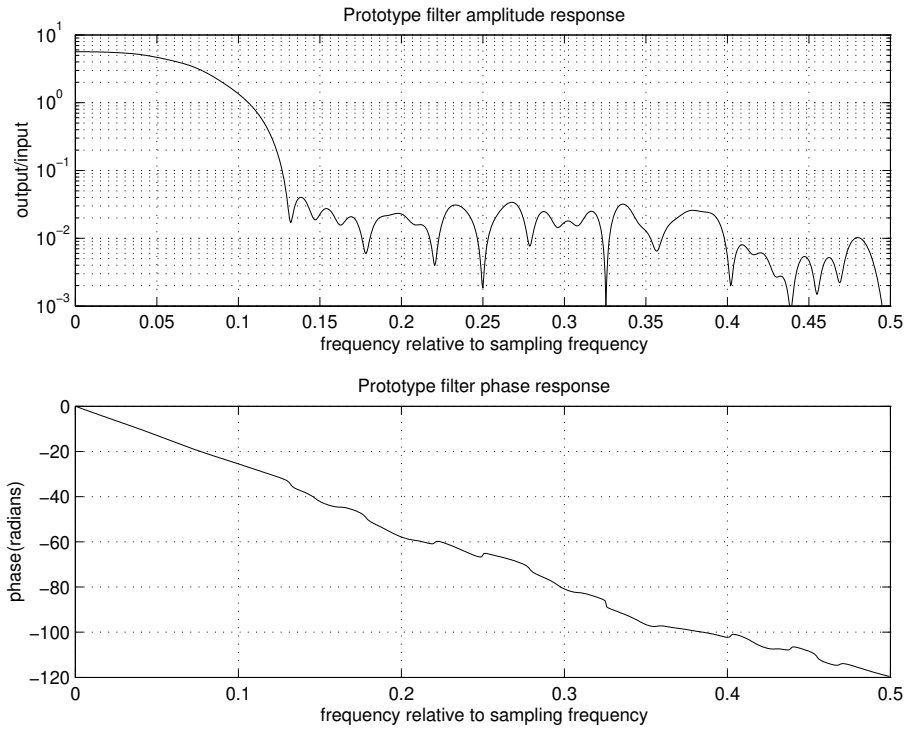


Figure B.2: Perfect reconstruction 2–fold oversampled filter bank : prototype frequency characteristic

obtained. A procedure optimizing the lattice rotation angles $\theta_l$ for all $N$ power complementary pairs finds a prototype filter with good lowpass characteristics. In this way the inserted aliasing level due to the subsampling is minimized. The stopband energy of $h_0[k]$ is computed and minimized using a steepest descent optimization procedure.

**Example B.1** An 8–channel perfect reconstruction filter bank set was designed through optimization of Eqs. B.74–B.76. In this example $M = 8$, $N = 4$ and $L_n = 9$ for each of the 4 power complementary pairs. The length of the prototype filter is 80. The frequency characteristic of the analysis prototype is shown in figure B.2. The aliasing level (Eq. 3.75) introduced by 1 filter bank is around -40.8 dB. $\triangle$

By inverse decomposition of the lossless lattices the lattice rotation angles corresponding to a prototype time response can be computed. This inverse parameterization can be used to find good starting values for the steepest descent optimization procedure.

**Inverse decomposition of 2 × 1 lossless FIR systems**

A causal real–valued $2 \times 1$ lossless FIR system of order $L$ can be factorized as (see also Eqs. B.74–B.76) :

$$\mathbf{M}^{(L)}(z) = \mathbf{R}_L \mathbf{\Lambda}(z) \ldots \mathbf{R}_{l+1} \mathbf{\Lambda}(z) \mathbf{M}^{(l)}(z) \tag{B.78}$$

where

$$\mathbf{M}^{(l)}(z) = \mathbf{R}_l \mathbf{\Lambda}(z) \ldots \mathbf{R}_1 \mathbf{\Lambda}(z) \begin{bmatrix} \cos\theta_0 \\ -\sin\theta_0 \end{bmatrix}, \tag{B.79}$$

$\mathbf{R}_l$ are Givens rotations

$$\mathbf{R}_l = \begin{bmatrix} \cos\theta_l & \sin\theta_l \\ -\sin\theta_l & \cos\theta_l \end{bmatrix} \tag{B.80}$$

and $\mathbf{\Lambda}(z)$ is a first–order $2 \times 2$ para–unitary building block :

$$\mathbf{\Lambda}(z) = \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix}. \tag{B.81}$$

Call

$$\mathbf{M}^{(l)}(z) = \begin{bmatrix} A^{(l)}(z) \\ B^{(l)}(z) \end{bmatrix}, \tag{B.82}$$

then

$$A^{(l+1)}(z) = A^{(l)}(z)\cos\theta_l + z^{-1}B^{(l)}(z)\sin\theta_l \tag{B.83}$$

$$B^{(l+1)}(z) = -A^{(l)}(z)\sin\theta_l + z^{-1}B^{(l)}(z)\cos\theta_l \tag{B.84}$$

follow from Eqs. B.78, B.80, B.81 and B.82. By inversion of Eqs. B.83 and B.84

$$A^{(l)}(z) = A^{(l+1)}(z)\cos\theta_l - B^{(l+1)}(z)\sin\theta_l \tag{B.85}$$

$$z^{-1}B^{(l)}(z) = A^{(l+1)}(z)\sin\theta_l + B^{(l+1)}(z)\cos\theta_l \tag{B.86}$$

is obtained. The order of $A^{(l)}(z)$ and $B^{(l)}(z)$ is $l$, so they can be expressed as

$$A^{(l)}(z) = \sum_{k=0}^{l} a_k^{(l)} z^{-k} \tag{B.87}$$

$$B^{(l)}(z) = \sum_{k=0}^{l} b_k^{(l)} z^{-k}. \tag{B.88}$$

From Eqs. B.85–B.88

$$a_k^{(l)} = a_k^{(l+1)}\cos\theta_l - b_k^{(l+1)}\sin\theta_l, \qquad \forall k \tag{B.89}$$

$$b_{k-1}^{(l)} = a_k^{(l+1)}\sin\theta_l + b_k^{(l+1)}\cos\theta_l, \qquad \forall k \tag{B.90}$$

are found. As $a_k^{(l)} = b_k^{(l)} = 0$ for $k < 0$ and $k > l$,

$$a_{l+1}^{(l+1)} \cos\theta_l - b_{l+1}^{(l+1)} \sin\theta_l = 0 \tag{B.91}$$

$$a_0^{(l+1)} \sin\theta_l + b_0^{(l+1)} \cos\theta_l = 0. \tag{B.92}$$

Hence,

$$\theta_l = \text{atan}\frac{a_{l+1}^{(l+1)}}{b_{l+1}^{(l+1)}} = -\text{atan}\frac{b_0^{(l+1)}}{a_0^{(l+1)}}. \tag{B.93}$$

Once $\mathbf{M}^{(l+1)}(z)$ and $\theta_l$ are known, $\mathbf{M}^{(l)}(z)$ can be obtained from Eqs. B.85 and B.86.

Once the prototype is given $\mathbf{M}^{(L)}(z)$ can be computed for all lattices. Starting from $\mathbf{M}^{(L)}(z)$ and applying the same procedure $\mathbf{M}^{(L-1)}(z)$ and $\theta_L$ can be computed. This procedure can be recursively applied to $\mathbf{M}^{(L-1)}(z)$ until all rotation angles $\theta_l$ are known.

## B.8 Para–unitary DFT modulated filter banks revisited

In this section it will be proven that a para–unitary DFT modulated filter bank as described by Eqs. 3.47, 3.48, 3.53 and 3.54 fulfills the perfect reconstruction conditions presented in Eqs. 4.15 and 4.16.

For an $M$–band, $N$–fold downsampled para–unitary DFT modulated filter bank

$$\tilde{\mathbf{B}}(z)\mathbf{B}(z) = \mathbf{I}_N, \tag{B.94}$$

cf. Eq. 3.47. As $\mathbf{H}(z) = \mathbf{F}\mathbf{B}(z)$ (Eq. 3.53),

$$\tilde{\mathbf{H}}(z)\mathbf{H}(z) = M\mathbf{I}_N. \tag{B.95}$$

Now the analysis polyphase matrix $\mathbf{H}(z)$ can be written as

$$\mathbf{H}(z) = \sum_{l=0}^{P-1} z^{-l}\mathbf{H}^{(l)}. \tag{B.96}$$

$\mathbf{H}^{(l)}$ is a matrix containing the $l$–th order coefficients of $\mathbf{H}(z)$, assuming that $L_f = PN$ and $P \in \mathbb{N}$. Element $(m, n)$ of $\mathbf{H}^{(l)}$ therefore equals

$$\mathbf{H}^{(l)}(m, n) = h_m[n + lN], \tag{B.97}$$

which follows from Eq. 2.17.

In figure 4.4 and 4.5 matrices $\mathbf{H}$ and $\mathbf{G}$ were partitioned, resulting in

$$\mathbf{H}_p = \begin{bmatrix} h_0[pN] & \dots & h_0[pN+N-1] \\ \vdots & \ddots & \vdots \\ h_{M-1}[pN] & \dots & h_{M-1}[pN+N-1] \end{bmatrix} \tag{B.98}$$

and

$$\mathbf{G}_p = \begin{bmatrix} g_0[pN] & \cdots & g_0[pN+N-1] \\ \vdots & \ddots & \vdots \\ g_{M-1}[pN] & \cdots & g_{M-1}[pN+N-1] \end{bmatrix}, \tag{B.99}$$

where $0 \leqslant p \in \mathbb{N} < P$. As for a para–unitary DFT modulated filter bank set Eq. 3.54 holds,

$$\mathbf{G}_p = \frac{1}{M}\bar{\mathbf{H}}^*_{P-1-p} = \frac{1}{M}\mathbf{H}^*_{P-1-p}\mathbf{J} \tag{B.100}$$

is obtained, assuming that $L_f = PN$, $P \in \mathbb{N}$ and $\delta = \frac{L_f}{N} - 1$. Now, Eq. 4.14 can be expressed as

$$\mathbf{V}_q = \sum_{p=0}^{P-1} \mathbf{G}^T_{P-1-p}\bar{\mathbf{H}}_{p+q} = \frac{1}{M}\sum_{p=0}^{P-1} \mathbf{J}\mathbf{H}^{*T}_p\mathbf{H}_{p+q}\mathbf{J}, \tag{B.101}$$

which should be unity for $q = 0$ and zero elsewhere for perfect reconstruction.

Comparing Eq. B.97 with B.98 it appears that $\mathbf{H}^{(l)}$ corresponds to $\mathbf{H}_l$ as defined in Eq. B.98. So, Eq. B.96 becomes

$$\mathbf{H}(z) = \sum_{l=0}^{P-1} z^{-l}\mathbf{H}_l. \tag{B.102}$$

Hence, equation B.95 can be rewritten as

$$(\mathbf{H}^{*T}_0 + z\mathbf{H}^{*T}_1 + \cdots + z^{P-1}\mathbf{H}^{*T}_{P-1})(\mathbf{H}_0 + z^{-1}\mathbf{H}_1 + \cdots + z^{-(P-1)}\mathbf{H}_{P-1}) = M\mathbf{I}_N, \tag{B.103}$$

being equivalent to

$$\frac{1}{M}\sum_{p=0}^{P-1} \mathbf{H}^{*T}_p\mathbf{H}_p = \mathbf{I}_N \qquad \text{term corresponding to } z^0 \tag{B.104}$$

$$\sum_{p=0}^{P-1} \mathbf{H}^{*T}_p\mathbf{H}_{p+q} = \mathbf{0}_N \qquad \text{terms corresponding to } z^{-q}, \tag{B.105}$$

$q \in \mathbb{Z}_0$, which is in accordance with Eqs. 4.15, 4.16 and B.101.

# C Appendix to part II

## C.1 Proof of theorem 5.2

**Theorem 5.2** *In the case of an $M$–band, $N$–fold downsampled ideally frequency selective DFT modulated filter bank the $i$–th subband adaptive filter $\hat{f}_i[k]$ in a setup as in figure 5.4 converges to*

$$f_i[k] = \left( w[m] \star \left( e^{-j\frac{2\pi i m}{M}} \mathrm{sinc}(\frac{\pi m}{N}) \right) \right)_{N\downarrow}, \tag{C.1}$$

*i.e., $f_i[k]$ can be obtained by downsampling the convolution of the impulse response of the unknown system $w[k]$ and a modulated double–sided sinc.*

**Proof:** Call $x$ the input to the unknown system and $d = w \star x$ the output (see figure 5.4), then $x_i^s$ and $d_i^s$ can be related to $x$ in the Fourier domain (apply Eq. 2.6 and change $z$ into $e^{j\omega} = e^{j2\pi f}$) :

$$X_i^s(e^{j\omega}) = \frac{1}{N} \sum_{n=0}^{N-1} H_i(e^{j\frac{\omega - 2\pi n}{N}}) X(e^{j\frac{\omega - 2\pi n}{N}}) \tag{C.2}$$

$$D_i^s(e^{j\omega}) = \frac{1}{N} \sum_{n=0}^{N-1} W(e^{j\frac{\omega - 2\pi n}{N}}) H_i(e^{j\frac{\omega - 2\pi n}{N}}) X(e^{j\frac{\omega - 2\pi n}{N}}). \tag{C.3}$$

$W(z)$ is the $z$–transform of $w[k]$ and $H_i(z)$ is the $i$–th analysis filter. In the case of a DFT modulated filter bank equation 2.10 can be applied, i.e. $H_i(z) = H_0(e^{j\frac{2\pi i}{M}} z)$.

Apparently, there is no simple relation between $X_i^s$ and $D_i^s$ due to the frequency shifting and summation.

***Lemma C.1*** *If the subband filters $H_i(z)$ in Eqs. C.2 and C.3 are frequency selective, i.e. the corresponding prototype filter $H_0(z)$ is capable of blocking every frequency component above $\frac{f_s}{2N}$, the $N$–fold downsampling will not introduce any aliasing.*

***Proof:*** There will not be any overlap of different frequency components in this case because a complex and frequency selective DFT modulated filter bank produces subband signals having a joint, non–zero bandpass frequency spectrum with a bandwidth equal to $\frac{f_s}{N}$. This is thanks to the lowpass characteristics of the prototype filter.

In general, when a bandlimited signal $s[k]$ with bandwidth $\frac{f_s}{N}$ is $N$–fold downsampled, the subsampled signal $v[k]$ can be expressed as (Eq. 5.9) :

$$V(e^{j\omega}) = \frac{1}{N} \sum_{n=0}^{N-1} S(e^{j\frac{\omega - 2\pi n}{N}}). \tag{C.4}$$

Aliasing would occur if for any $\omega$ in $V(e^{j\omega})$, there is more than one value of $n$ for which $S(e^{j\frac{\omega-2\pi n}{N}}) \neq 0$.

Suppose there are 2 different values of $n$ for which $V(e^{j\omega})$ is non–zero. Then $S(e^{j\Omega})$ would be evaluated at $\Omega_1 = \frac{\omega}{N} - \frac{2\pi n_1}{N}$ and $\Omega_2 = \frac{\omega}{N} - \frac{2\pi n_2}{N}$, with $n_1, n_2 \in \mathbb{N}$ and $n_1 \neq n_2$. As $\Omega_2 - \Omega_1 = \frac{2\pi(n_2-n_1)}{N}$, $S(e^{j\Omega})$ would have to be evaluated at 2 values of $\Omega$ being further apart than its bandwidth $\frac{2\pi}{N}$. This would be in contradiction with the assumption that $s[k]$ is a bandlimited signal with bandwidth $\frac{f_s}{N}$. $\diamond$

***Lemma C.2*** *The summation in Eqs. C.2 and C.3 runs over at most 2 values of n.*

***Proof:*** $N$–fold downsampling comes down to dividing the frequency spectrum in $N$ equal bins. The total input spectrum is scanned and evaluated going through the different bins simultaneously at a sampling rate that is $N$ times lower than the original sampling rate and summing the results together (see figure C.1 or Eq. C.4). When different bins lead to scans having overlapping contributions, aliasing occurs.

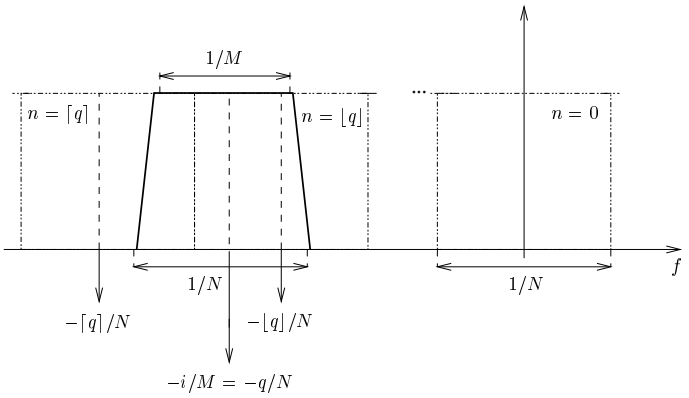


Figure C.1: Downsampling mechanism visualized

As there are $N$ frequency bins in total the position of the $n$–th bin is given by $\left[\frac{-1}{2N} - \frac{n}{N}, \frac{1}{2N} - \frac{n}{N}\right]$, whereas the $i$–th subband appears at $\left[\frac{-1}{2N} - \frac{i}{M}, \frac{1}{2N} - \frac{i}{M}\right]$, with $n, i \in \mathbb{N}$. For uniformity reasons $\left[-\frac{1}{2N} - 1, \frac{1}{2N}\right]$, is considered as the fundamental interval in this proof.

If index $q \in \mathbb{Q}$ in figure C.1 is now defined as

$$q = \frac{N}{M} i \tag{C.5}$$

it is observed that only the bins with index $\lfloor q \rfloor$ and $\lceil q \rceil$ lead to non–zero contributions in Eqs. C.2 and C.3

$$X_i^s(e^{j\omega}) = \frac{1}{N} \sum_{n=\lfloor q \rfloor, \lceil q \rceil} H_i(e^{j\frac{\omega-2\pi n}{N}}) X(e^{j\frac{\omega-2\pi n}{N}}) \tag{C.6}$$

$$D_i^s(e^{j\omega}) = \frac{1}{N} \sum_{n=\lfloor q \rfloor, \lceil q \rceil} W(e^{j\frac{\omega-2\pi n}{N}}) H_i(e^{j\frac{\omega-2\pi n}{N}}) X(e^{j\frac{\omega-2\pi n}{N}}), \tag{C.7}$$

due to the bandlimited character of the subband signals. This proofs the lemma. ◇

In general, $q$ is rational and there will be 2 values of $n$ contributing. Within frame $n = \lceil q \rceil$, there is a contribution for $\frac{-1}{2N} - \frac{i}{M} < \frac{\Omega}{2\pi} \leqslant \frac{1}{2N} - \frac{\lceil q \rceil}{N}$ as can be observed from figure C.1. The downsampling process folds these back to $\lambda < \omega \leqslant \pi$, with

$$\lambda = \pi(-1 + 2\lceil q \rceil - 2q). \tag{C.8}$$

Similarly, frame $n = \lfloor q \rfloor$ gives a contribution for $\frac{1}{2N} - \frac{\lceil q \rceil}{N} \leqslant \frac{\Omega}{2\pi} < \frac{1}{2N} - \frac{i}{M}$, folded back to $-\pi \leqslant \omega < \lambda$.

The subband filter to be identified is therefore :

$$F_i(e^{j\omega}) = \frac{D_i^s(e^{j\omega})}{X_i^s(e^{j\omega})} = \begin{cases} W(e^{j\frac{\omega-2\pi\lceil q \rceil}{N}}) & \text{if} \quad \lambda < \omega \leqslant \pi \\ W(e^{j\frac{\omega-2\pi\lfloor q \rfloor}{N}}) & \text{if} \quad -\pi \leqslant \omega < \lambda \end{cases} \tag{C.9}$$

As $\omega$ rises from $-\pi$ to $\pi$, $W(e^{j\Omega})$ is being evaluated for increasing values of $\Omega$, except at $\omega = \lambda$ where $\Omega$ suddenly drops by $\frac{2\pi}{N}$.

The time sequence $f_i[k]$ corresponding to (C.9) can be found by inverse Fourier transformation (Eq. 2.4) :

$$f_i[k] = \frac{1}{2\pi} \int_{-\pi}^{\pi} F_i(e^{j\omega}) e^{j\omega k} d\omega \tag{C.10}$$

$$= \frac{1}{2\pi} \int_{-\pi}^{\lambda} W(e^{j\frac{\omega-2\pi\lfloor q \rfloor}{N}}) e^{j\omega k} d\omega + \frac{1}{2\pi} \int_{\lambda}^{\pi} W(e^{j\frac{\omega-2\pi\lceil q \rceil}{N}}) e^{j\omega k} d\omega \tag{C.11}$$

$$= \frac{N}{2\pi} \int_{\frac{-\pi-2\pi\lfloor q \rfloor}{N}}^{\frac{\lambda-2\pi\lfloor q \rfloor}{N}} W(e^{j\nu}) e^{j\nu(Nk)} d\nu + \frac{N}{2\pi} \int_{\frac{\lambda-2\pi\lceil q \rceil}{N}}^{\frac{\pi-2\pi\lceil q \rceil}{N}} W(e^{j\nu}) e^{j\nu(Nk)} d\nu \tag{C.12}$$

$$= \frac{N}{2\pi} \int_{\mathcal{D}} W(e^{j\nu}) e^{j\nu(Nk)} d\nu, \tag{C.13}$$

$$\text{with } \mathcal{D} = \left[\frac{-\pi-2\pi\lfloor q \rfloor}{N}, \frac{\lambda-2\pi\lfloor q \rfloor}{N}\right] \cup \left[\frac{\lambda-2\pi\lceil q \rceil}{N}, \frac{\pi-2\pi\lceil q \rceil}{N}\right]$$

$$= \frac{N}{2\pi}\int_{-\pi}^{\pi} G_i(e^{j\nu})e^{j\nu(Nk)}\,d\nu \tag{C.14}$$

$$= g_i[kN]. \tag{C.15}$$

Hence, $f_i[k]$ is an $N$–fold downsampled version of

$$g_i[m] = Nw[m] \star p_i[m] \tag{C.16}$$

in which $p_i[m]$ is a frequency clipper removing all frequency components outside $\mathcal{D}$:

$$p_i[m] = \frac{1}{2\pi}\int_{-\pi}^{\pi} P_i(e^{j\nu})e^{j\nu m}\,d\nu \tag{C.17}$$

$$= \frac{1}{2\pi}\int_{\mathcal{D}} e^{j\nu m}\,d\nu \tag{C.18}$$

$$= \frac{1}{2\pi}\frac{e^{j\nu m}}{jm}\bigg|_{\mathcal{D}} \tag{C.19}$$

$$= \frac{1}{2\pi}\frac{e^{j(\frac{\lambda-2\pi\lfloor q\rfloor}{N})m} - e^{j(\frac{-\pi-2\pi\lfloor q\rfloor}{N})m} + e^{j(\frac{\pi-2\pi\lceil q\rceil}{N})m} - e^{j(\frac{\lambda-2\pi\lceil q\rceil}{N})m}}{jm}. \tag{C.20}$$

If $\lceil q\rceil = \lfloor q\rfloor = q$, $\mathcal{D}$ reduces to $\left[-\frac{\pi}{N} - \frac{2\pi q}{N}, \frac{\pi}{N} - \frac{2\pi q}{N}\right]$ and $\lambda = -\pi$. Hence,

$$p_i[m] = \frac{e^{-j\frac{2\pi qm}{N}}}{N}\mathrm{sinc}(\frac{\pi m}{N}). \tag{C.21}$$

By inserting Eq. C.5

$$p_i[m] = \frac{e^{-j\frac{2\pi im}{M}}}{N}\mathrm{sinc}(\frac{\pi m}{N}) \tag{C.22}$$

is obtained.

If $\lceil q\rceil \neq \lfloor q\rfloor$, $\mathcal{D}$ is $\left[\frac{\lambda}{N} - \frac{2\pi\lceil q\rceil}{N}, \frac{\lambda}{N} - \frac{2\pi\lfloor q\rfloor}{N}\right]$, hence

$$p_i[m] = \frac{e^{j\frac{m}{N}(\lambda-\pi(2\lfloor q\rfloor+1))}}{N}\mathrm{sinc}(\frac{\pi m}{N}) \tag{C.23}$$

and by inserting Eqs. C.5 and C.8

$$p_i[m] = \frac{e^{-j\frac{2\pi qm}{N}}}{N}\mathrm{sinc}(\frac{\pi m}{N}) \tag{C.24}$$

$$= \frac{e^{-j\frac{2\pi im}{M}}}{N}\mathrm{sinc}(\frac{\pi m}{N}) \tag{C.25}$$

is obtained.

In both cases, $P_i(z) = P_0(e^{j\frac{2\pi i}{M}}z)$, so —as could be expected— the frequency clipper $P_i(z)$ is just a frequency translated version of the lowpass clipper $P_0(z)$. This translation corresponds to the frequency shift between $H_i$ and $H_0$. □

## C.2 Proof of theorem 5.3

**Theorem 5.3** *Consider the general adaptive filtering setup of figure 2.8. Define $L_w \times 1$ vectors $\mathbf{x}$ and $\mathbf{w}$, containing the input samples $x$ and the coefficients of the unknown system $w$ respectively, and an $L_{\hat{w}} \times 1$ vector $\hat{\mathbf{w}}$ representing the adaptive filter. If $L_{\hat{w}} \leqslant L_w$, $\hat{\mathbf{w}}$ is an approximation of $\mathbf{w}$ modelling only a part of the unknown system. Under these assumptions the expected minimum residual error is given by*

$$\min_{\hat{\mathbf{w}}} \mathcal{E}\{|e[k]|^2\} = \mathcal{E}\{|d[k]|^2\} - \mathbf{w}^H \mathbf{R}_{xx} \mathbf{\Xi} \left( \mathbf{\Xi}^T \mathbf{R}_{xx} \mathbf{\Xi} \right)^{-1} \mathbf{\Xi}^T \mathbf{R}_{xx} \mathbf{w}, \tag{C.26}$$

*in which $\mathcal{E}\{|d[k]|^2\} = \mathbf{w}^H \mathbf{R}_{xx} \mathbf{w}$ and $\mathbf{R}_{xx} = \mathcal{E}\{\mathbf{x}\mathbf{x}^H\}$. Further, $\mathbf{\Xi}$ is an $L_w \times L_{\hat{w}}$ matrix containing ones and zeros indicating which coefficients of $\mathbf{w}$ are modelled, such that $e[k] = \mathbf{w}^H \mathbf{x} - (\mathbf{\Xi}\hat{\mathbf{w}})^H \mathbf{x}$.*

**Proof:** From figure 2.8 it is found that

$$\begin{aligned} e[k] &= d[k] - y[k] && \text{(C.27)} \\ &= \mathbf{w}^H \mathbf{x} - (\mathbf{\Xi}\hat{\mathbf{w}})^H \mathbf{x}, && \text{(C.28)} \end{aligned}$$

with $\mathbf{\Xi}$ an $L_w \times L_{\hat{w}}$ matrix containing ones and zeros indicating which coefficients of $\mathbf{w}$ are modelled. For instance, if the coefficients $\mathbf{w}(L_0)$ till $\mathbf{w}(L_0 + L_{\hat{w}} - 1)$ are modelled, take

$$\mathbf{\Xi} = \begin{bmatrix} \mathbf{0}_{L_0 \times L_{\hat{w}}} \\ \mathbf{I}_{L_{\hat{w}}} \\ \mathbf{0}_{(L_w - L_0 - L_{\hat{w}}) \times L_{\hat{w}}} \end{bmatrix}. \tag{C.29}$$

Now from Eq. C.28, it follows that

$$\begin{aligned} \hat{\mathbf{w}}_{\text{opt}} &= \underset{\hat{\mathbf{w}}}{\text{argmin}}\ \mathcal{E}\{|e[k]|^2\} && \text{(C.30)} \\ &= \underset{\hat{\mathbf{w}}}{\text{argmin}}\ (\mathbf{w} - \mathbf{\Xi}\hat{\mathbf{w}})^H \mathcal{E}\{\mathbf{x}\mathbf{x}^H\} (\mathbf{w} - \mathbf{\Xi}\hat{\mathbf{w}}). && \text{(C.31)} \end{aligned}$$

If $\mathbf{R}_{xx} = \mathcal{E}\{\mathbf{x}\mathbf{x}^H\}$ is defined, the cost function $J = \mathcal{E}\{|e[k]|^2\}$ becomes

$$\begin{aligned} J &= (\mathbf{w} - \mathbf{\Xi}\hat{\mathbf{w}})^H \mathcal{E}\{\mathbf{x}\mathbf{x}^H\} (\mathbf{w} - \mathbf{\Xi}\hat{\mathbf{w}}) && \text{(C.32)} \\ &= \mathbf{w}^H \mathbf{R}_{xx} \mathbf{w} - \mathbf{w}^H \mathbf{R}_{xx} \mathbf{\Xi}\hat{\mathbf{w}} - \hat{\mathbf{w}}^H \mathbf{\Xi}^T \mathbf{R}_{xx} \mathbf{w} + \hat{\mathbf{w}}^H \mathbf{\Xi}^T \mathbf{R}_{xx} \mathbf{\Xi}\hat{\mathbf{w}} && \text{(C.33)} \end{aligned}$$

In order to find $\hat{\mathbf{w}}_{\text{opt}}$ the gradient of $J$ w.r.t. $\hat{\mathbf{w}}$ has to be computed.

**Intermezzo**
It is explained in [81] how the gradient of a real cost function w.r.t. a complex–valued vector can be computed.

If $J \in \mathbb{R}$ is a real scalar depending on $\mathbf{w} \in \mathbb{C}^N$ such that

$$\mathbf{w} = \mathbf{w}_r + j\mathbf{w}_i = \begin{bmatrix} \mathbf{w}_r(0) + j\mathbf{w}_i(0) \\ \vdots \\ \mathbf{w}_r(N-1) + j\mathbf{w}_i(N-1) \end{bmatrix}, \tag{C.34}$$

with $\mathbf{w}_r, \mathbf{w}_i \in \mathbb{R}^N$, the gradient vector is defined as

$$\boldsymbol{\nabla} J = 2\frac{\partial J}{\partial \mathbf{w}^*} = \begin{bmatrix} \frac{\partial J}{\partial \mathbf{w}_r(0)} + j\frac{\partial J}{\partial \mathbf{w}_i(0)} \\ \vdots \\ \frac{\partial J}{\partial \mathbf{w}_r(N-1)} + j\frac{\partial J}{\partial \mathbf{w}_i(N-1)} \end{bmatrix}. \tag{C.35}$$

It is shown in [81] that

$$\text{if} \quad J = \mathbf{w}^H\mathbf{x} \quad \text{then} \quad \frac{\partial J}{\partial \mathbf{w}^*} = \mathbf{x}, \tag{C.36}$$

$$\text{if} \quad J = \mathbf{x}^H\mathbf{w} \quad \text{then} \quad \frac{\partial J}{\partial \mathbf{w}^*} = \mathbf{0}, \tag{C.37}$$

$$\text{if} \quad J = \mathbf{w}^H\mathbf{R}\mathbf{w} \quad \text{then} \quad \frac{\partial J}{\partial \mathbf{w}^*} = \mathbf{R}\mathbf{w}. \tag{C.38}$$

♣

By applying Eqs. C.35–C.38 to Eq. C.33 one finds

$$\boldsymbol{\nabla} J = -2\boldsymbol{\Xi}^T\mathbf{R}_{xx}\mathbf{w} + 2\boldsymbol{\Xi}^T\mathbf{R}_{xx}\boldsymbol{\Xi}\hat{\mathbf{w}}. \tag{C.39}$$

Setting the gradient to $\mathbf{0}$ gives

$$\boldsymbol{\Xi}^T\mathbf{R}_{xx}\mathbf{w} = \boldsymbol{\Xi}^T\mathbf{R}_{xx}\boldsymbol{\Xi}\hat{\mathbf{w}}_{\text{opt}} \tag{C.40}$$

such that

$$\hat{\mathbf{w}}_{\text{opt}} = \left(\boldsymbol{\Xi}^T\mathbf{R}_{xx}\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T\mathbf{R}_{xx}\mathbf{w}. \tag{C.41}$$

If now $x$ is a stationary white noise source, $\mathbf{R}_{xx} = \sigma_x^2\mathbf{I}_{L_w}$ and hence by plugging in Eq. C.29

$$\hat{\mathbf{w}}_{\text{opt}} = \boldsymbol{\Xi}^T\mathbf{w} \tag{C.42}$$

is found, i.e. $\hat{\mathbf{w}}_{\text{opt}}$ exactly corresponds to

$$\begin{bmatrix} \mathbf{w}(L_0) \\ \vdots \\ \mathbf{w}(L_0 + L_{\hat{w}} - 1) \end{bmatrix}. \tag{C.43}$$

If $x$ is a stationary colored signal, it is assumed that $x$ can be written as $\kappa \star n$, with $n$ a stationary white noise signal and $\kappa$ a coloring filter. Hence,

$$\mathbf{x} = \mathbf{K}^H\mathbf{n}, \tag{C.44}$$

with $\mathbf{K}$ an $(L_w + L_\kappa - 1) \times L_w$ matrix such that

$$\mathbf{K}^H = \begin{bmatrix} \kappa[0] & \dots & \kappa[L_\kappa - 1] & 0 & \dots & 0 & \dots & 0 \\ 0 & \kappa[0] & \dots & \kappa[L_\kappa - 1] & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \dots & \kappa[0] & \dots & \kappa[L_\kappa - 1] \end{bmatrix}. \tag{C.45}$$

From Eq. C.44 it follows that $\mathbf{R}_{xx} = \sigma_n^2 \mathbf{K}^H \mathbf{K}$ and that

$$\hat{\mathbf{w}}_{\text{opt}} = \left( \mathbf{\Xi}^T \mathbf{K}^H \mathbf{K} \mathbf{\Xi} \right)^{-1} \mathbf{\Xi}^T \mathbf{K}^H \mathbf{K} \mathbf{w}. \tag{C.46}$$

Further from Eqs. C.33, C.40 and C.41,

$$\begin{aligned} \min_{\hat{\mathbf{w}}} \ \mathcal{E}\{|e[k]|^2\} &= (\mathbf{w} - \mathbf{\Xi}\hat{\mathbf{w}}_{\text{opt}})^H \mathbf{R}_{xx} (\mathbf{w} - \mathbf{\Xi}\hat{\mathbf{w}}_{\text{opt}}) && \text{(C.47)} \\ &= \mathbf{w}^H \mathbf{R}_{xx} \mathbf{w} - \mathbf{w}^H \mathbf{R}_{xx} \mathbf{\Xi} \hat{\mathbf{w}}_{\text{opt}} && \text{(C.48)} \\ &= \mathcal{E}\{|d[k]|^2\} - \mathbf{w}^H \mathbf{R}_{xx} \mathbf{\Xi} \hat{\mathbf{w}}_{\text{opt}} && \text{(C.49)} \\ &= \mathcal{E}\{|d[k]|^2\} - \mathbf{w}^H \mathbf{R}_{xx} \mathbf{\Xi} \left( \mathbf{\Xi}^T \mathbf{R}_{xx} \mathbf{\Xi} \right)^{-1} \mathbf{\Xi}^T \mathbf{R}_{xx} \mathbf{w}, && \text{(C.50)} \end{aligned}$$

which completes the proof. □

## C.3 Detailed complexity analysis for the subband adaptive filter

In this section a detailed cost analysis will be performed for the subband adaptive filter. We will assume that both the input signals $x$ and $d$ and the unknown system $w[k]$ (see figure 2.3) are real–valued and that $f_s$ is the sampling rate corresponding to $x$ and $d$. Further, $L_{SB}$ is the length of the subband adaptive filters, $M$ represents the number of subbands and $N$ is the downsampling factor. All filter banks are assumed to be (modified) DFT modulated. The length of the analysis prototype filter is $L_f^a$, the length of the synthesis prototype filter is $L_f^s$. More information on the assumptions made for the cost computation can be found in section 2.4.

The major cost contributions are :

- the adaptive filtering cost
- the filter bank cost, which implies
  - a DFT or IDFT operation
  - cost of the structured polynomial matrices $\mathbf{B}(z)$ and $\mathbf{C}(z)$

### Adaptive filtering cost

Recall that only about half of the subbands need to be processed as most subbands are two by two complex conjugated. If the number of subbands $M$ is even, two subbands are real and only $\frac{M-2}{2}$ of the $M-2$ remaining complex bands have to be computed thanks to the symmetry. For odd $M$, there is only one real band and $\frac{M-1}{2}$ complex bands to be processed.

In the sequel two variables will be used : $n_{rb}$ stands for the number of real subbands to be processed and $n_{cb}$ is the number of complex subbands to be processed. If $M$ is even, $n_{rb} = 2$ and $n_{cb} = \frac{M-2}{2}$. For odd $M$ on the other hand, $n_{rb} = 1$ and $n_{cb} = \frac{M-1}{2}$.

It will be assumed that the LMS adaptive filter is used to update the subband filter weights. The cost for adapting an $L_{SB}$–taps filter using the LMS algorithm amounts to (see section 2.3.1) $16L_{SB}+2$ `op.` if complex signals are involved, and $4L_{SB}+1$ `op.` for real signals.

Hence, taking into account that $n_{rb}$ subband filters are updated using the real LMS algorithm and $n_{cb}$ filters using the complex LMS, and that all operations are done at the lower sampling rate $\frac{f_s}{N}$, the total cost for the adaptive filtering part is

$$c_{af} = ((4L_{SB}+1)n_{rb} + (16L_{SB}+2)n_{cb})\,\frac{f_s}{N}\ \texttt{ops}. \tag{C.51}$$

$$= (4L_{SB}n_{rb} + 16L_{SB}n_{cb} + 2n_{cb} + n_{rb})\,\frac{f_s}{N}\ \texttt{ops}. \tag{C.52}$$

### Filter bank implementation cost

In the general subband adaptive filtering scheme of figure 5.1 there are three filter banks : one synthesis bank and two analysis banks. For this cost analysis it is assumed that the filter banks are (modified) DFT modulated and FIR.

For the analysis banks, the incoming signals are fed into an $N$–th order tapped delay line and are subsequently $N$–fold subsampled (figure 5.2). The filter bank operations come down to a multiplication of the set of delayed and downsampled signals with a polyphase matrix $\mathbf{H}(z)$. As shown in section 5.1.5, $\mathbf{H}(z)$ can be factorized as $\mathbf{FB}(z)$ in which $\mathbf{F}$ is the $M \times M$ DFT matrix and $\mathbf{B}(z)$ is a real–valued structured $M \times N$ matrix containing the $K = \mathrm{lcm}(M,N)$ polyphase components of the filter bank prototype filter $h_0[k]$. To the synthesis part an analogous decomposition does apply.

### Computation of B(z)

It was shown in section 3.1.1 (see Eq. 3.3) that

$$[\mathbf{B}(z)]_{m,n} = \begin{cases} z^{-q} H_{0_{(n+qN):K}}(z^Q) & \text{if } (m-n) \bmod g = 0 \\ 0 & \text{if } (m-n) \bmod g \neq 0, \end{cases} \tag{C.53}$$

with $g = \gcd(M, N)$ and $Q = M/g$, i.e. the entries of $\mathbf{B}(z)$ contain products of upsampled polyphase components $H_{0_{(n+qN):K}}(z^Q)$ of the analysis prototype $H_0(z)$ and signal delays $z^{-q}$.

$L_f^a$ is the length of the analysis prototype $H_0(z)$. If it is assumed that $L_f^a$ is a multiple of $K$, the length of $H_{0_{(n+qN):K}}(z^Q)$ equals $\frac{L_f^a Q}{K}$. As $Q-1$ out of the $Q$ elements of $H_{0_{(n+qN):K}}(z^Q)$ are zero, the effective filter length equals $\frac{L_f^a}{K}$. As there are $K$ polyphase components in total the number of elementary filtering operations required for $\mathbf{B}(z)$ is $K\frac{L_f^a}{K} = L_f^a$. Hence, the computational load corresponding to $\mathbf{B}(z)$ is equivalent to that of a single $L_f^a$–taps FIR filter.

On DSP processors elementary FIR filtering operations (MACs) can be computed efficiently in one machine cycle. However, here it is assumed that the filtering operations are explicitly computed as a multiplication and an addition.

Recall that all filtering operations are real (also those in the synthesis filter bank) and are done at the lower sampling rate. The total cost therefore amounts to $K(2\frac{L_f^a}{K} - 1) = 2L_f^a - K$ `op`. The synthesis bank prototype polyphase matrix $\mathbf{C}(z)$ is completely analogous and hence also requires $2L_f^s - K$ `op`.

The filter contributions are added up resulting in $M$ entries for the DFT matrix. On each row of $\mathbf{B}(z)$ there are $\frac{K}{M}$ polyphase components. This requires another $M(\frac{K}{M} - 1) = K - M$ additions. For the synthesis bank the equivalent formula is $K - N$.

This leads to a total number of

$$c_{bz} = (2L_f^a - M)\frac{f_s}{N} \texttt{ops}. \tag{C.54}$$

for the computation of $\mathbf{B}(z)$ and

$$c_{cz} = (2L_f^s - N)\frac{f_s}{N} \texttt{ops}. \tag{C.55}$$

for $\mathbf{C}(z)$.

**Frequency–domain transformation**

Whenever $M$ is a power of 2, the (I)DFT matrix can be implemented efficiently by means of an (I)FFT. For an $M$–point real–input FFT $2M\log_2 M - 4M + 6$ `op`. are counted, whereas $2M\log_2 M$ `op`. for an $M$–point real–output IFFT (see also section 2.4).

When $M \neq 2^r, r \in \mathbb{N}$, less efficient computational schemes have to be called for. In the worst case the product of the (I)DFT matrix with the signal vector should be explicitly computed.

In the case of an explicit computation of the matrix–vector product for the analysis part only half of the DFT matrix is needed as the complex subbands are complex

conjugated. For the real subband(s) $M-1$ real additions/subtractions need to be done, leading to $n_{rb}(M-1)+n_{cb}(4M-2)$ `op`.

For the synthesis part the product of a signal vector $\underline{\mathbf{u}}$ with the $M \times M$ IDFT matrix has to be computed. Now, the $i$–th output of this matrix–vector product is

$$u_i = \frac{1}{M}\sum_{m=0}^{M-1} e^{j\frac{2\pi m i}{M}}\underline{\mathbf{u}}(m). \tag{C.56}$$

As, if $M$ is even, $\underline{\mathbf{u}}(m) = \underline{\mathbf{u}}^*(M-m)$ for $m = 1 : M-1$, Eq. C.56 becomes

$$\begin{aligned} M u_i &= \underline{\mathbf{u}}(0) + (-1)^i \underline{\mathbf{u}}(\frac{M}{2}) + \sum_{m=1}^{\frac{M}{2}-1} e^{j\frac{2\pi m i}{M}}\underline{\mathbf{u}}(m) + e^{j\frac{2\pi (M-m) i}{M}}\underline{\mathbf{u}}(M-m) && \text{(C.57)}\\ &= \underline{\mathbf{u}}(0) + (-1)^i \underline{\mathbf{u}}(\frac{M}{2}) + \sum_{m=1}^{\frac{M}{2}-1} e^{j\frac{2\pi m i}{M}}\underline{\mathbf{u}}(m) + e^{-j\frac{2\pi m i}{M}}\underline{\mathbf{u}}^*(m) && \text{(C.58)}\\ &= \underline{\mathbf{u}}(0) + (-1)^i \underline{\mathbf{u}}(\frac{M}{2}) + 2\sum_{m=1}^{\frac{M}{2}-1} \Re\left\{e^{j\frac{2\pi m i}{M}}\underline{\mathbf{u}}(m)\right\}. && \text{(C.59)} \end{aligned}$$

So, the matrix–vector product requires $2M^2 - 3M = 4Mn_{cb} + M(n_{rb}-1)$ `op`. For odd values of $M$

$$\begin{aligned} M u_i &= \underline{\mathbf{u}}(0) + \sum_{m=1}^{\frac{M-1}{2}} e^{j\frac{2\pi m i}{M}}\underline{\mathbf{u}}(m) + e^{j\frac{2\pi (M-m) i}{M}}\underline{\mathbf{u}}(M-m) && \text{(C.60)}\\ &= \underline{\mathbf{u}}(0) + \sum_{m=1}^{\frac{M-1}{2}} e^{j\frac{2\pi m i}{M}}\underline{\mathbf{u}}(m) + e^{-j\frac{2\pi m i}{M}}\underline{\mathbf{u}}^*(m) && \text{(C.61)}\\ &= \underline{\mathbf{u}}(0) + 2\sum_{m=1}^{\frac{M-1}{2}} \Re\left\{e^{j\frac{2\pi m i}{M}}\underline{\mathbf{u}}(m)\right\} && \text{(C.62)} \end{aligned}$$

is obtained, the cost being also $4Mn_{cb} + M(n_{rb}-1)$ `op`. in this case.

### Total cost

Summing all partial cost factors together and taking into account that there are two analysis filter banks and one synthesis filter bank leads to

$$\begin{aligned} c \;=\; &\big(4L_{SB}n_{rb} + 16L_{SB}n_{cb} + 2n_{cb} + n_{rb} + 2(2L_f^a - M) + (2L_f^s - N)\\ &+2(2M\log_2 M - 4M + 6) + (2M\log_2 M)\big)\frac{f_s}{N}\ \texttt{ops}. \end{aligned} \tag{C.63}$$

$$= \big(4L_{SB}n_{rb} + 16L_{SB}n_{cb} + 2n_{cb} + n_{rb} + 4L_f^a + 2L_f^s + 6M\log_2 M - N - 10M + 12\big)\frac{f_s}{N}\ \texttt{ops}. \quad \text{(C.64)}$$

if $M = 2^r, r \in \mathbb{N}$ and

$$c = \big(4L_{SB}n_{rb} + 16L_{SB}n_{cb} + 2n_{cb} + n_{rb} + 2(2L_f^a - M) + (2L_f^s - N) + 2(n_{rb}(M-1) + n_{cb}(4M-2)) + M(4n_{cb} + n_{rb} - 1)\big)\frac{f_s}{N}\ \texttt{ops}. \quad \text{(C.65)}$$

$$= \big(4L_{SB}n_{rb} + 16L_{SB}n_{cb} - 2n_{cb} - n_{rb} + 4L_f^a + 2L_f^s - N - 3M + 3Mn_{rb} + 12Mn_{cb}\big)\frac{f_s}{N}\ \texttt{ops}. \quad \text{(C.66)}$$

for $M \neq 2^r, r \in \mathbb{N}$.

## C.4 Proof of theorem 6.1

**Theorem 6.1** *A random value $\epsilon$ can appear as an extra $(P+q+1)$–st element of $\mathbf{w}_p^{(n)}$ in equation 6.27 if $q = 0 : \sigma - 1$ and $p = 0 : \frac{L_{FB}}{P} - 2$ and as long as it is compensated for, at the $(q+1)$–st element of $\mathbf{w}_{p+1}^{(n)}$ for instance.*

**Proof:** It is observed from Eqs. 6.5, 6.6, 6.7 and 6.13 that if $\begin{bmatrix}\mathbf{0} & \mathbf{I}_L\end{bmatrix}\mathbf{F}^{-1}\underline{\mathbf{X}}_p^{(n)}\underline{\check{\mathbf{w}}}_p^{(n)}$ is added to Eq. 6.14 such that

$$\begin{bmatrix}\mathbf{0} & \mathbf{I}_L\end{bmatrix}\mathbf{F}^{-1}\underline{\mathbf{X}}_p^{(n)}\underline{\check{\mathbf{w}}}_p^{(n)} = \begin{bmatrix}\mathbf{0} & \mathbf{I}_L\end{bmatrix}\bar{\mathbf{X}}_{n,p}^T\check{\mathfrak{w}}_p^{(n)} \quad \text{(C.67)}$$

$$= \begin{bmatrix}\mathbf{0} & \mathbf{I}_L\end{bmatrix}\bar{\mathbf{X}}_{n,p}^T\begin{bmatrix}\mathbf{0}_{(P+q)\times 1}\\ \epsilon \\ \mathbf{0}_{(M-P-q-1)\times 1}\end{bmatrix} \quad \text{(C.68)}$$

$$= \epsilon\begin{bmatrix}\vdots \\ x[(n+1)L - pP - P - q - 1] \\ x[(n+1)L - pP - P - q]\end{bmatrix}\Bigg\updownarrow L \quad \text{(C.69)}$$

is obtained. As $\bar{\mathbf{X}}_{n,p}$, defined by Eq. 6.7, is circulant Eq. C.69 contains a set of subsequent samples of $x$ only if

$$(n+1)L - pP - P - q - L + 1 \geqslant (n+1)L - pP - M + 1 \quad \text{(C.70)}$$

$$\Longleftrightarrow$$

$$q \leqslant M - P - L \quad \text{(C.71)}$$

such that Eq. C.69 becomes

$$\begin{bmatrix}\mathbf{0} & \mathbf{I}_L\end{bmatrix}\mathbf{F}^{-1}\underline{\mathbf{X}}_p^{(n)}\underline{\breve{\mathbf{w}}}_p^{(n)} = \epsilon\begin{bmatrix} x[(n+1)L-pP-P-q-L+1] \\ \vdots \\ x[(n+1)L-pP-P-q]\end{bmatrix} \tag{C.72}$$

$$= \begin{bmatrix}\mathbf{0} & \mathbf{I}_L\end{bmatrix}\bar{\mathbf{X}}_{n,p+1}^T\begin{bmatrix}\mathbf{0}_{q\times 1} \\ \epsilon \\ \mathbf{0}_{(M-1-q)\times 1}\end{bmatrix}. \tag{C.73}$$

Hence, the extra $(P+q+1)$–st element of $\mathfrak{w}_p^{(n)}$ can be compensated for at $\mathfrak{w}_{p+1}^{(n)}(q+1)$.

If now $q \geqslant uP$ with $u \in \mathbb{N}_0$, values can also be exchanged between vector $\mathfrak{w}_p^{(n)}$ and vector $\mathfrak{w}_{p+1+u}^{(n)}$ :

$$\begin{bmatrix}\mathbf{0} & \mathbf{I}_L\end{bmatrix}\mathbf{F}^{-1}\underline{\mathbf{X}}_p^{(n)}\underline{\breve{\mathbf{w}}}_p^{(n)} = \begin{bmatrix}\mathbf{0} & \mathbf{I}_L\end{bmatrix}\bar{\mathbf{X}}_{n,p+1+u}^T\begin{bmatrix}\mathbf{0}_{(q-uP)\times 1} \\ \epsilon \\ \mathbf{0}_{(M-1-q+uP)\times 1}\end{bmatrix} \tag{C.74}$$

or e.g. can be spread out over $\mathfrak{w}_{p+1}^{(n)}$ and $\mathfrak{w}_{p+1+u}^{(n)}$. Observe also that $p$ cannot be equal to $\frac{L_{FB}}{P}-1$ in Eqs. C.67–C.74 as $\bar{\mathbf{X}}_{n,\frac{L_{FB}}{P}}$ does not exist.

Observe from Eq. C.69 that for values of $q$ equal to or larger than $M-P-L+1=\sigma$ (cf. Eq. C.71), aliased components of $x$ are inserted in

$$\mathbf{y}_n = \begin{bmatrix}\mathbf{0} & \mathbf{I}_L\end{bmatrix}\mathbf{F}^{-1}\sum_{p=0}^{\frac{L_{FB}}{P}-1}\underline{\mathbf{X}}_p^{(n)}\underline{\mathbf{w}}_p^{(n)},$$

which has an negative effect on the convergence of the adaptive filters. Hence, the adaptive filter will not converge to a solution for which $q \geqslant \sigma$ unless $\epsilon = 0$. As a consequence, after convergence random values $\epsilon$ are only encountered as an extra $(P+q+1)$–st element of $\mathbf{w}_p^{(n)}$ provided $q = 0:\sigma-1$ and $p = 0:\frac{L_{FB}}{P}-2$, which concludes the proof. □

## C.5 Proof of theorem 6.2

**Theorem 6.2** *The ambiguity can be removed by taking into account $L+\sigma$ instead of $L$ samples to determine the error $\boldsymbol{e}^{(n)}$. Therefore Eqs. 6.29 and 6.30 are slightly changed into*

$$\mathbf{d}^{(n)} = \begin{bmatrix}\mathbf{0} \\ \mathbf{d}_n\end{bmatrix}\begin{matrix}\updownarrow P-1 \\ \updownarrow L+\sigma\end{matrix}, \qquad \mathbf{d}_n = \begin{bmatrix} d[nL-\sigma+1] \\ \vdots \\ d[(n+1)L]\end{bmatrix}\Big\updownarrow L+\sigma \tag{C.75}$$

$$\mathbf{y}^{(n)} = \begin{bmatrix}\mathbf{0}_{P-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{L+\sigma}\end{bmatrix}\mathbf{F}^{-1}\sum_{p=0}^{\frac{L_{FB}}{P}-1}\underline{\mathbf{X}}_p^{(n)}\underline{\mathbf{w}}_p^{(n)}. \tag{C.76}$$

**Proof:** It is observed that in this case Eq. C.69 becomes

$$\begin{bmatrix}\mathbf{0} & \mathbf{I}_{L+\sigma}\end{bmatrix}\mathbf{F}^{-1}\underline{\mathbf{X}}_p^{(n)}\breve{\underline{\mathbf{w}}}_p^{(n)} = \epsilon \begin{bmatrix}\vdots \\ x[(n+1)L-pP-P-q-1] \\ x[(n+1)L-pP-P-q]\end{bmatrix}\Bigg\updownarrow L+\sigma. \tag{C.77}$$

Hence, a set of subsequent samples of $x$ is obtained if

$$q \leqslant M-P-L-\sigma, \tag{C.78}$$

i.e. $q \leqslant -1$, which is not possible. Vector $\mathbf{y}_n$ would therefore always contain aliased components of $x$. Hence, all $\epsilon$ have to be 0 if the algorithm has converged. □

## C.6 "Time–reversed" PBFDAF

Eqs. 6.28–6.32 define the PBFDAF algorithm. Remark that in Eq. 6.28, 6.30 and 6.31 the most recent samples are encountered in the lower part of the signal vectors. Here, we will derive an equivalent formulation of the PBFDAF algorithm with time–reversed buffers, i.e. with the most recent samples encountered at the top. The "time–reversed" PBFDAF is the mathematical equivalent of the signal flow graph shown in figure 6.3.

By time–reversing the filter equation 6.29,

$$\mathbf{J}\mathbf{y}^{(n)} = \begin{bmatrix}\mathbf{I}_L & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{P-1+\sigma}\end{bmatrix}\mathbf{J}\mathbf{F}^{-1}\sum_{p=0}^{\frac{L_{FB}}{P}-1}\underline{\mathbf{X}}_p^{(n)}\underline{\mathbf{w}}_p^{(n)} \tag{C.79}$$

$$= \begin{bmatrix}\mathbf{I}_L & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{P-1+\sigma}\end{bmatrix}\mathbf{F}^{-1}\sum_{p=0}^{\frac{L_{FB}}{P}-1}\mathbf{F}\mathbf{J}\mathbf{F}^{-1}\underline{\mathbf{X}}_p^{(n)}\mathbf{F}\mathbf{F}^{-H}\mathbf{F}^H\mathbf{F}^{-1}\underline{\mathbf{w}}_p^{(n)} \tag{C.80}$$

$$= \begin{bmatrix}\mathbf{I}_L & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{P-1+\sigma}\end{bmatrix}\mathbf{F}^{-1}\sum_{p=0}^{\frac{L_{FB}}{P}-1}\breve{\underline{\mathbf{X}}}_p^{(n)}\breve{\underline{\mathbf{w}}}_p^{(n)}, \tag{C.81}$$

is obtained with

$$\breve{\underline{\mathbf{w}}}_p^{(n)} = \mathbf{F}^H\mathbf{F}^{-1}\underline{\mathbf{w}}_p^{(n)} \tag{C.82}$$

and

$$\breve{\underline{\mathbf{X}}}_p^{(n)} = \mathbf{F}\mathbf{J}\mathbf{F}^{-1}\underline{\mathbf{X}}_p^{(n)}\mathbf{F}\mathbf{F}^{-H}. \tag{C.83}$$

It can easily be verified that

$$\mathbf{F}\mathbf{F}^{-H} = \mathbf{F}^H\mathbf{F}^{-1} = \mathbf{F}^{-H}\mathbf{F} = \mathbf{F}^{-1}\mathbf{F}^H = \begin{bmatrix}1 & \mathbf{0}_{1\times M-1} \\ \mathbf{0}_{M-1\times 1} & \mathbf{J}_{M-1}\end{bmatrix}. \tag{C.84}$$

Hence, from Eq. C.82 it follows that $\breve{\underline{\mathbf{w}}}_p^{(n)}$ is obtained by swapping $\underline{\mathbf{w}}_p^{(n)}(1)$ till $\underline{\mathbf{w}}_p^{(n)}(M-1)$. If now $\mathbf{w}_p^{(n)} = \mathbf{F}^{-1}\underline{\mathbf{w}}_p^{(n)}$ is a real–valued vector, then $\breve{\underline{\mathbf{w}}}_p^{(n)} = \underline{\mathbf{w}}_p^{(n)^*}$.

Further, as $\underline{\mathbf{X}}_p^{(n)}$ is a diagonal matrix, $\mathbf{F}^{-1}\underline{\mathbf{X}}_p^{(n)}\mathbf{F}$ in Eq. C.83 is a right–circulant matrix and therefore the DFT coefficients of its first column are the diagonal elements of $\underline{\mathbf{X}}_p^{(n)}$ (see appendix A). Hence,

$$\mathbf{F}^{-1}\underline{\mathbf{X}}_p^{(n)}\mathbf{F} = \begin{bmatrix} x[(n+1)L-pP-M+1] & \dots & x[(n+1)L-pP-M+2] \\ \vdots & \ddots & \vdots \\ x[(n+1)L-pP] & \dots & x[(n+1)L-pP-M+1] \end{bmatrix}. \quad \text{(C.85)}$$

Now,

$$\breve{\underline{\mathbf{X}}}_p^{(n)} = \mathbf{F}\mathbf{J}(\mathbf{F}^{-1}\underline{\mathbf{X}}_p^{(n)}\mathbf{F})\mathbf{F}^{-H}\mathbf{F}\mathbf{F}^{-1} \quad \text{(C.86)}$$

$$= \mathbf{F}\breve{\mathbf{X}}_{n,p}\mathbf{F}^{-1} \quad \text{(C.87)}$$

is found and from Eqs. C.84 and C.85 the right–circulant matrix

$$\breve{\mathbf{X}}_{n,p} = \mathbf{J}(\mathbf{F}^{-1}\underline{\mathbf{X}}_p^{(n)}\mathbf{F}) \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{M-1} \end{bmatrix} \quad \text{(C.88)}$$

$$= \begin{bmatrix} x[(n+1)L-pP] & \dots & x[(n+1)L-pP-1] \\ \vdots & \ddots & \vdots \\ x[(n+1)L-pP-M+1] & \dots & x[(n+1)L-pP] \end{bmatrix}. \quad \text{(C.89)}$$

can be obtained. From Eqs. C.87 and C.89 it is now observed that

$$\breve{\underline{\mathbf{X}}}_p^{(n)} = \operatorname{diag}\left\{ \mathbf{F} \begin{bmatrix} x[(n+1)L-pP] \\ \vdots \\ x[(n+1)L-pP-M+1] \end{bmatrix} \right\}, \quad \text{(C.90)}$$

which corresponds to a "time–reversed" version of $\underline{\mathbf{X}}_p^{(n)}$.

By time–reversing also the other equations of the PBFDAF, Eqs. 6.30 and 6.31 for instance become

$$\mathbf{J}\mathbf{d}^{(n)} = \begin{bmatrix} \mathbf{I}_L & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{P-1+\sigma} \end{bmatrix} \begin{bmatrix} d[(n+1)L] \\ \vdots \\ d[(n+1)L-M+1] \end{bmatrix}, \quad \text{(C.91)}$$

and

$$\mathbf{J}\mathbf{e}^{(n)} = \mathbf{J}\mathbf{d}^{(n)} - \mathbf{J}\mathbf{y}^{(n)}. \quad \text{(C.92)}$$

Hence, Eqs. C.81, C.90, C.91 and C.92 replace Eqs. 6.28–6.31. All vectors appear to be "time–reversed".

The same transformation can be applied to the weight update equation (Eq. 6.32) of the PBFDAF. Using Eqs. C.82 and C.84, we obtain

$$\underline{\mathbf{w}}_p^{(n+1)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n)} + \mathbf{G}\boldsymbol{\Delta}\underline{\mathbf{X}}_p^{(n)*}\mathbf{F}\mathbf{e}^{(n)} \quad \text{(C.93)}$$

$$\mathbf{F}^H\mathbf{F}^{-1}\breve{\underline{\mathbf{w}}}_p^{(n+1)} \overset{\forall p}{=} \mathbf{F}^H\mathbf{F}^{-1}\breve{\underline{\mathbf{w}}}_p^{(n)} + \mathbf{G}\boldsymbol{\Delta}\underline{\mathbf{X}}_p^{(n)^*}\mathbf{F}\mathbf{J}\mathbf{J}\mathbf{e}^{(n)}, \tag{C.94}$$

which reduces to

$$\breve{\underline{\mathbf{w}}}_p^{(n+1)} \overset{\forall p}{=} \breve{\underline{\mathbf{w}}}_p^{(n)} + \mathbf{F}\mathbf{F}^{-H}\mathbf{G}\boldsymbol{\Delta}\underline{\mathbf{X}}_p^{(n)^*}\mathbf{F}\mathbf{J}\mathbf{F}^{-1}\mathbf{F}\mathbf{J}\mathbf{e}^{(n)}. \tag{C.95}$$

From Eq. C.83 it is observed that

$$\underline{\mathbf{X}}_p^{(n)} = \mathbf{F}\mathbf{J}\mathbf{F}^{-1}\breve{\underline{\mathbf{X}}}_p^{(n)}\mathbf{F}^H\mathbf{F}^{-1} \tag{C.96}$$

and hence that

$$\underline{\mathbf{X}}_p^{(n)^*} = \left(\underline{\mathbf{X}}_p^{(n)}\right)^H = \mathbf{F}^{-H}\mathbf{F}\breve{\underline{\mathbf{X}}}_p^{(n)^*}\mathbf{F}^{-H}\mathbf{J}\mathbf{F}^H \tag{C.97}$$

$$= \mathbf{F}^{-H}\mathbf{F}\breve{\underline{\mathbf{X}}}_p^{(n)^*}\mathbf{F}\mathbf{J}\mathbf{F}^{-1}. \tag{C.98}$$

Combining Eqs. C.95 and C.98 leads to

$$\breve{\underline{\mathbf{w}}}_p^{(n+1)} \overset{\forall p}{=} \breve{\underline{\mathbf{w}}}_p^{(n)} + \mathbf{F}\mathbf{F}^{-H}\mathbf{G}\boldsymbol{\Delta}\mathbf{F}^{-H}\mathbf{F}\breve{\underline{\mathbf{X}}}_p^{(n)^*}\mathbf{F}\mathbf{J}\mathbf{F}^{-1}\mathbf{F}\mathbf{J}\mathbf{F}^{-1}\mathbf{F}\mathbf{J}\mathbf{e}^{(n)} \tag{C.99}$$

$$\overset{\forall p}{=} \breve{\underline{\mathbf{w}}}_p^{(n)} + (\mathbf{F}\mathbf{F}^{-H}\mathbf{G}\mathbf{F}^H\mathbf{F}^{-1})(\mathbf{F}\mathbf{F}^{-H}\boldsymbol{\Delta}\mathbf{F}^{-H}\mathbf{F})\breve{\underline{\mathbf{X}}}_p^{(n)^*}\mathbf{F}\mathbf{J}\mathbf{e}^{(n)} \tag{C.100}$$

$$\overset{\forall p}{=} \breve{\underline{\mathbf{w}}}_p^{(n)} + \breve{\mathbf{G}}\breve{\boldsymbol{\Delta}}\breve{\underline{\mathbf{X}}}_p^{(n)^*}\mathbf{F}\mathbf{J}\mathbf{e}^{(n)}. \tag{C.101}$$

It is observed from Eq. C.84 that

$$\breve{\boldsymbol{\Delta}} = \mathbf{F}\mathbf{F}^{-H}\boldsymbol{\Delta}\mathbf{F}^{-H}\mathbf{F} \tag{C.102}$$

$$= \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{M-1} \end{bmatrix} \boldsymbol{\Delta} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{M-1} \end{bmatrix}. \tag{C.103}$$

Diagonal matrix $\breve{\boldsymbol{\Delta}}$ can therefore be obtained by swapping diagonal elements $\boldsymbol{\Delta}(1)$ till $\boldsymbol{\Delta}(M-1)$. This is in according with the transformation of $\underline{\mathbf{w}}_p^{(n)}$ into $\breve{\underline{\mathbf{w}}}_p^{(n)}$. Furthermore,

$$\breve{\mathbf{G}} = \mathbf{F}\mathbf{F}^{-H}\mathbf{G}\mathbf{F}^H\mathbf{F}^{-1} \tag{C.104}$$

$$= \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{M-1} \end{bmatrix} \mathbf{G} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{M-1} \end{bmatrix}. \tag{C.105}$$

In the case of unconstrained updating $\mathbf{G} = \mathbf{I}_M$ and hence also $\breve{\mathbf{G}} = \mathbf{I}_M$. For the constrained PBFDAF Eqs. 6.41 and C.84 are plugged in :

$$\breve{\mathbf{G}} = \mathbf{F}\mathbf{F}^{-H}\mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{F}^{-1}\mathbf{F}^H\mathbf{F}^{-1} \tag{C.106}$$

$$= \mathbf{F} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{M-1} \end{bmatrix} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{M-1} \end{bmatrix} \mathbf{F}^{-1} \tag{C.107}$$

$$= \mathbf{F} \begin{bmatrix} 1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{P-1} \end{bmatrix} \mathbf{F}^{-1}. \tag{C.108}$$

## C.7 Proof of theorem 6.3

**Theorem 6.3** *For the constrained PBFDAF*

$$\left[\begin{array}{cccc}\mathbf{I}_L & \mathbf{0}_L & \dots & \mathbf{0}\end{array}\right]\mathbf{F}^{-1}\mathrm{diag}\{F_i(z)\}\mathbf{F}\left[\begin{array}{c}\mathbf{I}_L\\ z^{-1}\mathbf{I}_L\\ \vdots\\ \mathbf{R}(z)\end{array}\right] \tag{C.109}$$

*is a pseudo–circulant $L\times L$ matrix. In the case of unconstrained updating Eq. C.109 is guaranteed to be pseudo–circulant only after convergence.*

**Proof :** On page 129 it was shown that

$$\mathbf{F}^{-1}\mathrm{diag}\{F_i(z)\}\mathbf{F} = \mathbf{M}(z) = \mathbf{J}\bar{\mathbf{M}}(z)\mathbf{J} \tag{C.110}$$

with $\bar{\mathbf{M}}(z) = \mathbf{F}^{-1}\mathrm{diag}\{\bar{F}_i(z)\}\mathbf{F}$ and $\bar{F}_i(z)$ the subband adaptive filters of the intermediate scheme 6.2.

It is known from appendix A that

$$\mathbf{F}^{-1}\mathrm{diag}\{\bar{F}_i(z)\}\mathbf{F} = \bar{\mathbf{M}}(z) \tag{C.111}$$

is a circulant matrix and that the filters $\bar{F}_i(z)$ correspond to the DFT coefficients of the first column of $\bar{\mathbf{M}}(z)$. Now, from Eq. C.110 it is observed that $\mathbf{M}(z) = \mathbf{J}\bar{\mathbf{M}}(z)\mathbf{J} = \bar{\mathbf{M}}^T(z)$ as $\bar{\mathbf{M}}(z)$ is circulant. Hence, the filters $F_i(z)$ correspond to the DFT coefficients of the first row of $\bar{\mathbf{M}}(z)$.

Eq. 6.45 shows how the frequency–domain transformed filter partitions $\underline{\mathbf{w}}_p^{(n)}$ relate to the filters $\bar{F}_i(z)$. As the filters $\bar{F}_i(z)$ correspond to the DFT coefficients of the first column of $\bar{\mathbf{M}}(z)$ it follows from Eqs. 6.45 and 6.27 that this first column of $\bar{\mathbf{M}}(z)$ is given by

$$\mathbf{F}^{-1}\sum_{p=0}^{\frac{L_{FB}}{P}-1}\underline{\mathbf{w}}_p^{(n)}z^{-p\frac{P}{L}} = \sum_{p=0}^{\frac{L_{FB}}{P}-1}\left[\begin{array}{c}\mathbf{w}_p^{(n)}\\ \mathbf{0}\end{array}\right]z^{-p\frac{P}{L}} \tag{C.112}$$

$$= \left[\begin{array}{c}\sum_{p=0}^{\frac{L_{FB}}{P}-1}\mathbf{w}_p^{(n)}z^{-p\frac{P}{L}}\\ \mathbf{0}_{(M-P)\times 1}\end{array}\right] \tag{C.113}$$

$$= \left[\begin{array}{c}W_0(z^{\frac{P}{L}})\\ \vdots\\ W_{P-1}(z^{\frac{P}{L}})\\ 0\\ \vdots\\ 0\end{array}\right], \tag{C.114}$$

in which $W_p(z)$ is the $p$–th out of $P$ polyphase components of $\hat{w}^{(n)}$, i.e.

$$W_p(z) = \sum_{k=0}^{\frac{L_{FB}}{P}-1} w^{(n)}[p+Pk]\; z^{-k}. \tag{C.115}$$

Observe that Eq. 6.27 and hence Eq. C.114 are valid only for the constrained PBFDAF and for the converged unconstrained PBFDAF with ambiguity compensation. For the unconstrained PBFDAF without ambiguity compensation

$$\mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{w}}_p^{(n)} z^{-p\frac{P}{L}} = \begin{bmatrix} \breve{W}_0(z^{\frac{P}{L}}) \\ \vdots \\ \breve{W}_{P-1+\sigma}(z^{\frac{P}{L}}) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{C.116}$$

will be valid after convergence, in which the filters $\breve{W}_p(z)$ don't have a direct physical meaning.

Hence, for the constrained PBFDAF

$$\bar{\mathbf{M}}(z) = \begin{bmatrix} W_0(z^{\frac{P}{L}}) & 0 & \dots & \dots & W_1(z^{\frac{P}{L}}) \\ \vdots & W_0(z^{\frac{P}{L}}) & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & W_{P-1}(z^{\frac{P}{L}}) \\ W_{P-1}(z^{\frac{P}{L}}) & \vdots & \ddots & \ddots & 0 \\ 0 & W_{P-1}(z^{\frac{P}{L}}) & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & \dots & W_0(z^{\frac{P}{L}}) \end{bmatrix} \tag{C.117}$$

is found. The subband filters $F_i(z)$ are the DFT coefficients of the first row of $\bar{\mathbf{M}}(z)$, i.e.

$$\mathbf{F}\begin{bmatrix} W_0(z^{\frac{P}{L}}) \\ 0 \\ \vdots \\ W_1(z^{\frac{P}{L}}) \end{bmatrix} = \mathbf{F}\begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{M-1} \end{bmatrix} \begin{bmatrix} W_0(z^{\frac{P}{L}}) \\ \vdots \\ W_{P-1}(z^{\frac{P}{L}}) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \overset{\text{Eq. C.84}}{=} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{M-1} \end{bmatrix} \mathbf{F} \begin{bmatrix} W_0(z^{\frac{P}{L}}) \\ \vdots \\ W_{P-1}(z^{\frac{P}{L}}) \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{C.118}$$

hence

$$F_0(z) = \bar{F}_0(z) \tag{C.119}$$

$$F_m(z) = \bar{F}_{M-m}(z), \qquad \text{for } m = 1 : M-1. \tag{C.120}$$

If furthermore the input signals $x$ and $d$ and the unknown system $w$ are real, $F_m(z) = \bar{F}_m^*(z)$, $\forall m$.

If Eqs. 6.55, C.110 and C.117 are plugged in in Eq. 5.7

$$\begin{bmatrix} W_0(z^{\frac{P}{L}}) & \dots & \dots & W_{P-1}(z^{\frac{P}{L}}) & 0 & \dots & \dots & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & W_0(z^{\frac{P}{L}}) & \dots & \dots & W_{P-1}(z^{\frac{P}{L}}) & 0 & \dots & 0 \end{bmatrix} \mathbf{B}(z) \tag{C.121}$$

is obtained. Using Eq. 6.51 equation C.121 can also be written as

$$\begin{bmatrix} W_0(z^{\frac{P}{L}}) & \dots & W_{L-1}(z^{\frac{P}{L}}) \\ 0 & \ddots & \dots \\ 0 & 0 & W_0(z^{\frac{P}{L}}) \end{bmatrix} + \sum_{n=1}^{\frac{P}{L}-1} z^{-n} \begin{bmatrix} W_{nL}(z^{\frac{P}{L}}) & \dots & W_{nL+L-1}(z^{\frac{P}{L}}) \\ \vdots & \ddots & \vdots \\ W_{nL-L+1}(z^{\frac{P}{L}}) & \dots & W_{nL}(z^{\frac{P}{L}}) \end{bmatrix}$$

$$+ z^{-\frac{P}{L}} \begin{bmatrix} 0 & \dots & 0 & 0 \\ W_{P-1}(z^{\frac{P}{L}}) & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ W_{P-L+1}(z^{\frac{P}{L}}) & \dots & W_{P-1}(z^{\frac{P}{L}}) & 0 \end{bmatrix}. \tag{C.122}$$

Hence, element $(i,j)$ of the $L \times L$ matrix $\mathbf{C}(z)\mathbf{F}^{-1}\text{diag}\{F_i(z)\}\mathbf{F}\mathbf{B}(z)$ can be written as

$$\sum_{n=-\infty}^{\infty} z^{-n} W_{j-i+nL}(z^{\frac{P}{L}}), \qquad \text{with } W_p(z) = 0 \text{ if } p < 0 \text{ or } p \geqslant P. \tag{C.123}$$

For the unconstrained PBFDAF Eq. C.121 becomes

$$\begin{bmatrix} \breve{W}_0(z^{\frac{P}{L}}) & \dots & \dots & \breve{W}_{P+\sigma-1}(z^{\frac{P}{L}}) & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \breve{W}_0(z^{\frac{P}{L}}) & \dots & \dots & \breve{W}_{P+\sigma-1}(z^{\frac{P}{L}}) \end{bmatrix} \mathbf{B}(z). \tag{C.124}$$

Hence, in this case element $(i,j)$ of $\mathbf{C}(z)\mathbf{F}^{-1}\text{diag}\{F_i(z)\}\mathbf{F}\mathbf{B}(z)$ is given by

$$\sum_{n=-\infty}^{\infty} z^{-n} \breve{W}_{j-i+nL}(z^{\frac{P}{L}}), \qquad \text{with } \breve{W}_p(z) = 0 \text{ if } p < 0 \text{ or } p \geqslant P+\sigma. \tag{C.125}$$

An $L \times L$ matrix $\mathbf{T}(z)$ is pseudo–circulant if for all $i = 0 : L-2$ and $j = 0 : L-2$ (see also appendix A)

$$[\mathbf{T}(z)]_{i+1,j+1} = [\mathbf{T}(z)]_{i,j} \tag{C.126}$$

and

$$[\mathbf{T}(z)]_{i+1,0} = z^{-1}[\mathbf{T}(z)]_{i,L-1}. \tag{C.127}$$

Expression C.123 and C.125 depend on $j-i$, hence all elements on the same diagonal are equal. Furthermore, element $(i, L-1)$ of Eq. C.123 is

$$\sum_{n=-\infty}^{\infty} z^{-n} W_{L-1-i+nL}(z^{\frac{P}{L}}) \overset{m=n+1}{=} z \sum_{m=-\infty}^{\infty} z^{-m} W_{-i-1+mL}(z^{\frac{P}{L}}), \tag{C.128}$$

which is element $(i+1, 0)$ multiplied by $z$. The same holds for Eq. C.125. Hence equation C.109 is indeed pseudo–circulant. □

## C.8 Proof of theorem 6.4

**Theorem 6.4** *A time–invariant finite–order system can be exactly modelled by the PBFDAF adaptive filter with a finite number of filter coefficients if $P$ is a multiple of $L$.*

**Proof :** From Eqs. 5.7, 5.8 and C.123 we know that for the constrained PBFDAF equation 6.59 models the following time–invariant system :

$$\hat{W}(z) = z^{-(L-1)} \sum_{l=0}^{L-1} \sum_{n=-\infty}^{\infty} z^{-nL-l} W_{l+nL}(z^P) \tag{C.129}$$

$$= z^{-(L-1)} \sum_{p=0}^{P-1} z^{-p} W_p(z^P) \tag{C.130}$$

in which $W_p$ are the $P$–th order polyphase components of the fullband adaptive filter $\hat{w}^{(n)}$. Eq. C.130 is also valid for the unconstrained PBFDAF with ambiguity compensation after convergence.

For the unconstrained PBFDAF without ambiguity compensation the equivalent expression is (see also Eq. C.125)

$$\hat{W}(z) = z^{-(L-1)} \sum_{p=0}^{P-1+\sigma} z^{-p} \breve{W}_p(z^P), \tag{C.131}$$

which is valid only after convergence.

If the filters $W_p$ or $\breve{W}_p$ have finite length and $P$ is a multiple of $L$, also the $F_i$ will be finite length filters because of Eq. C.110, and hence exact modelling of a finite–order system $w$ using finite–length filters $F_i(z)$ is possible. □

## C.9 Complexity analysis for the PBFDAF

In this section the global cost for the PBFDAF will be computed. We will refer to formulas 6.28–6.32 and the concepts presented in section 6.1.

It is assumed that :

- $f_s$ is the sampling rate
- $L_{FB}$ is the equivalent fullband adaptive filter length
- $L$ is the block length
- $P$ is the filter partition length. The effective length of the subband filters is therefore $\frac{L_{FB}}{P}$, hence $p = 0 : \frac{L_{FB}}{P} - 1$. It is further assumed that $P$ is a divisor of $L_{FB}$.
- $M$ is the size of the (I)DFTs and is supposed to be a power of 2 such that fast signal transforms can be called for.
- all operations are done at the downsampled rate $\frac{f_s}{L}$
- only $\lfloor \frac{M}{2} \rfloor + 1$ out of $M$ subbands need to be processed as the input signals $x$ and $d$ and the unknown system $w$ are assumed to be real–valued
- other assumptions concerning the cost of elementary arithmetic operations can be found in section 2.4

Recall that the update equations of the PBFDAF are summarized in table 8.1 for $R = 1$, including the results of this complexity analysis.

To process a block of $L$ samples with the PBFDAF a number of algorithmic steps are performed. Hereafter, the different algorithmic steps are enumerated chronologically and a cost is assigned to each step :

1. for the computation of Eq. 6.28 one real–input FFT is counted if $P$ is divisible by $L$ :

$$\underline{\mathbf{X}}_p^{(n)} \stackrel{\forall p}{=} \mathrm{diag}\{\mathbf{F}\mathbf{x}_n\} \quad \longrightarrow \quad 2M \log_2 M - 4M + 6 \ \mathtt{op}.$$

   On the other hand, $\frac{L_{FB}}{P}$ FFTs are needed if $P$ is not divisible by $L$.

2. normalization :

   - no normalization

$$\mu = \text{constant} \quad \longrightarrow \quad 0 \ \mathtt{op}.$$

$$\mathbf{\Delta} = \mu \mathbf{I}$$

- subband normalization (Eq. 6.35 and 6.39)

$$\mathbf{P}_n = \mathbf{P}_{(n-\frac{P}{L})} + \underline{\mathbf{X}}_0^{(n)} \underline{\mathbf{X}}_0^{(n)\,*} - \underline{\mathbf{X}}_0^{(n-L_{FB}/L)} \left(\underline{\mathbf{X}}_0^{(n-L_{FB}/L)}\right)^*$$

$$\longrightarrow \quad 2\left(3(\tfrac{M}{2}+1)\right) + 2(\tfrac{M}{2}+1) \text{ op.}$$

$$\mathbf{\Delta} = \mu \mathbf{P}_n^{-1} \quad \longrightarrow \quad 8(\tfrac{M}{2}+1) \text{ op.}$$

If $\frac{L_{FB}}{P} \leqslant 2$ it is better to compute $\mathbf{P}_n$ directly from Eq. 6.36 instead of using Eq. 6.39. In that case $2\frac{ML_{FB}}{P} + 4\frac{L_{FB}}{P} - \frac{M}{2} - 1$ op. are counted for $\mathbf{P}_n$. If $P$ is not divisible by $L$ Eq. 6.36 has to be applied anyway. Note that 8 op. were counted for a real division. This value was obtained from table 2.1 and also corresponds e.g. to the number of cycles for an inversion plus multiplication on a TMS320C4x DSP.

- global normalization (Eq. 6.39 and 6.40)

$$\mathbf{P}_n = \mathbf{P}_{(n-\frac{P}{L})} + \underline{\mathbf{X}}_0^{(n)} \underline{\mathbf{X}}_0^{(n)\,*} - \underline{\mathbf{X}}_0^{(n-L_{FB}/L)} \left(\underline{\mathbf{X}}_0^{(n-L_{FB}/L)}\right)^*$$

$$\longrightarrow \quad 2\left(3(\tfrac{M}{2}+1)\right) + 2(\tfrac{M}{2}+1) \text{ op.}$$

$$P_n = \frac{1}{M} \sum_{m=0}^{M-1} \mathbf{P}_n(m) \quad \longrightarrow \quad M + 7 \text{ op.}$$

$$\mu_n = \mu / P_n \quad \longrightarrow \quad 8 \text{ op.}$$

$$\mathbf{\Delta} = \mu_n \mathbf{I}$$

3. in each of the $\frac{M}{2}+1$ subbands to be processed

   (a) a complex filtering operation (Eq. 6.29) is done with $\frac{L_{FB}}{P}$–taps filters :

$$\underline{\mathbf{y}}^{(n)} = \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \underline{\mathbf{w}}_p^{(n)} \quad \longrightarrow \quad 4\frac{ML_{FB}}{P} + 8\frac{L_{FB}}{P} - M - 2 \text{ op.}$$

   (b) $\mathbf{y}^{(n)}$ is computed, which requires an $M$–point real–output IFFT :

$$\mathbf{y}^{(n)} = \begin{bmatrix} \mathbf{0}_{M-L} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \underline{\mathbf{y}}^{(n)} \quad \longrightarrow \quad 2M \log_2 M \text{ op.}$$

4. for Eq.6.31

$$\mathbf{e}^{(n)} = \mathbf{d}^{(n)} - \mathbf{y}^{(n)} \quad \longrightarrow \quad L \text{ op.}$$

are counted.

5. for the weight update (Eq. 6.32)

   (a) a real–input FFT is needed

   $$\underline{\mathbf{e}}^{(n)} = \mathbf{F}\mathbf{e}^{(n)} \quad \longrightarrow \quad 2M\log_2 M - 4M + 6 \text{ op.}$$

   (b) weighing each subband with the appropriate stepsize requires one multiplication of a real and a complex number per subband

   $$\underline{\mathbf{e}}_{\mathbf{\Delta}}^{(n)} = \mathbf{\Delta}\underline{\mathbf{e}}^{(n)} \quad \longrightarrow \quad 2(\tfrac{M}{2}+1) \text{ op.}$$

   (c) for each subband and for each filter tap a complex multiplication is needed to compute the gradient :

   $$\bar{\mathbf{\nabla}}_p \overset{\forall p}{=} \underline{\mathbf{X}}_p^{(n)*}\underline{\mathbf{e}}_{\mathbf{\Delta}}^{(n)} \quad \longrightarrow \quad 6\tfrac{L_{FB}}{P}(\tfrac{M}{2}+1) \text{ op.}$$

   (d) gradient error correction

   - unconstrained updating : no extra calculations involved

     $$\mathbf{\nabla}_p \overset{\forall p}{=} \bar{\mathbf{\nabla}}_p \quad \longrightarrow \quad 0 \text{ op.}$$

   - constrained updating : for each $p$ a real–input FFT and a real–output IFFT are required

     $$\mathbf{\nabla}_p \overset{\forall p}{=} \mathbf{F}\begin{bmatrix}\mathbf{I}_P & \mathbf{0}\\ \mathbf{0} & \mathbf{0}_{M-P}\end{bmatrix}\mathbf{F}^{-1}\bar{\mathbf{\nabla}}_p \quad \longrightarrow \quad \tfrac{L_{FB}}{P}(2M\log_2 M - 4M + 6 + 2M\log_2 M) \text{ op.}$$

   (e) $\underline{\mathbf{w}}_p^{(n)}$ has to be updated : one complex addition for each $p$ and for each subband

   $$\underline{\mathbf{w}}_p^{(n+1)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n)} + \mathbf{\nabla}_p \quad \longrightarrow \quad 2\tfrac{L_{FB}}{P}(\tfrac{M}{2}+1) \text{ op.}$$

This leads to the following global cost :

- unconstrained updating
  - no normalization

  $$\frac{f_s}{L}\left(6M\log_2 M + 8\frac{ML_{FB}}{P} + 16\frac{L_{FB}}{P} - 8M + L + 12\right) \text{ ops.} \qquad \text{(C.132)}$$

  - global normalization

  $$\frac{f_s}{L}\left(6M\log_2 M + 8\frac{ML_{FB}}{P} + 16\frac{L_{FB}}{P} - 3M + L + 35\right) \text{ ops.} \qquad \text{(C.133)}$$

  - subband normalization

  $$\frac{f_s}{L}\left(6M\log_2 M + 8\frac{ML_{FB}}{P} + 16\frac{L_{FB}}{P} + L + 28\right) \text{ ops.} \qquad \text{(C.134)}$$

- constrained updating
  - no normalization

$$\frac{f_s}{L}\left(2M\log_2 M\left(3+2\frac{L_{FB}}{P}\right)+4\frac{ML_{FB}}{P}+22\frac{L_{FB}}{P}-8M+L+12\right)\ \texttt{ops}. \tag{C.135}$$

  - global normalization

$$\frac{f_s}{L}\left(2M\log_2 M\left(3+2\frac{L_{FB}}{P}\right)+4\frac{ML_{FB}}{P}+22\frac{L_{FB}}{P}-3M+L+35\right)\ \texttt{ops}. \tag{C.136}$$

  - subband normalization

$$\frac{f_s}{L}\left(2M\log_2 M\left(3+2\frac{L_{FB}}{P}\right)+4\frac{ML_{FB}}{P}+22\frac{L_{FB}}{P}+L+28\right)\ \texttt{ops}. \tag{C.137}$$

If $P$ is not a multiple of $L$ an extra of

$$\frac{f_s}{L}\left(\left(\frac{L_{FB}}{P}-1\right)(2M\log_2 M-4M+6)\right)\ \texttt{ops}. \tag{C.138}$$

are required to compute $\underline{\mathbf{X}}_p^{(n)}$.

An extra of

$$\frac{f_s}{L}\left(2\frac{ML_{FB}}{P}+4\frac{L_{FB}}{P}-\frac{9M}{2}-9\right)\ \texttt{ops}. \tag{C.139}$$

are needed if $\frac{L_{FB}}{P}\leqslant 2$ or if $P$ is not a multiple of $L$.

## C.10 Proof of theorem 7.1

**Theorem 7.1** *The weight update equation of the unconstrained PBFDAF*

$$\underline{\mathbf{w}}_p^{(n+1)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n)} + \mathbf{\Delta}\,\underline{\mathbf{X}}_p^{(n)*}\,\mathbf{F}\mathbf{e}^{(n)} \tag{C.140}$$

*for which $P$ is a multiple of $L$, corresponds to equation 7.33, which was obtained by applying the fullband error adaptation update equation 7.13 to a subband systems with the filter banks of the PBFDAF.*

**Proof :** We first give a slightly different, but equivalent description of the fullband error adaptation algorithm presented in paragraph 7.1. The definition of $\mathbf{X}_n$ and $\mathbf{w}_n$ is changed and can be obtained by re–arranging the corresponding equations in paragraph 7.1 :

$$\underbrace{\begin{bmatrix}\mathbf{y}_0^{(n)}\\ \vdots \\ \mathbf{y}_{M-1}^{(n)}\end{bmatrix}}_{\mathbf{y}_n} = \underbrace{\begin{bmatrix}\mathbf{X}[n] & \dots & \mathbf{X}[n-L_{SB}+1]\end{bmatrix}}_{\bar{\mathbf{X}}_n}\underbrace{\begin{bmatrix}\bar{\mathbf{f}}_0^{(n)}\\ \vdots \\ \bar{\mathbf{f}}_{L_{SB}-1}^{(n)}\end{bmatrix}}_{\bar{\mathbf{w}}_n}, \tag{C.141}$$

in which

$$\mathbf{X}[n] = \underbrace{\begin{bmatrix} \boldsymbol{\chi}_0[n] & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \boldsymbol{\chi}_{M-1}[n] \end{bmatrix}}_{M} \Bigg\updownarrow ML_p, \tag{C.142}$$

and

$$\boldsymbol{\chi}_m[n] = \begin{bmatrix} x_m[n-L_p+1] \\ \vdots \\ x_m[n] \end{bmatrix}, \quad \bar{\mathbf{f}}_l^{(n)} = \begin{bmatrix} f_0^{(n)}[l] \\ \vdots \\ f_{M-1}^{(n)}[l] \end{bmatrix}. \tag{C.143}$$

The other equations 7.4, 7.5, 7.6 and 7.7 are left unchanged.

Apparently,

$$\mathbf{w}_n = \mathbf{P}\bar{\mathbf{w}}_n, \quad \mathbf{X}_n = \bar{\mathbf{X}}_n\mathbf{P}^T, \tag{C.144}$$

with $\mathbf{P}$ a permutation matrix, such that $\bar{\mathbf{w}}_n$ being a vector with $L_{SB}$ blocks of $M$ elements is projected on $\mathbf{w}_n$ having $M$ blocks of $L_{SB}$ elements. It appears that $\mathbf{y}_n = \bar{\mathbf{X}}_n\bar{\mathbf{w}}_n = \mathbf{X}_n\mathbf{w}_n$ has not changed. Hence, the set of equations C.141–C.143 is equivalent to the definitions given in paragraph 7.1.

The weight update equations 7.13 and 7.33 are still valid with $\bar{\boldsymbol{\Delta}} = \mathbf{P}^T\boldsymbol{\Delta}\mathbf{P}$ :

$$\bar{\mathbf{w}}_{n+1} = \bar{\mathbf{w}}_n + 2\frac{\bar{\boldsymbol{\Delta}}}{M}\bar{\mathbf{X}}_n^H\mathbf{F}\begin{bmatrix} \mathbf{I}_L & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{M-L} \end{bmatrix}\mathbf{F}^{-1}\mathbf{e}_n^{SB}. \tag{C.145}$$

It was proven in section 6.2 that, if $P$ is a multiple of $L$, the PBFDAF can be regarded as a subband adaptive system for which in each of the $M$ subbands there is an adaptive FIR filter with $\frac{L_{FB}}{P}$ non–zero coefficients.

Recall now Eqs. 6.28–6.32, which define the PBFDAF algorithm. In section C.6 an equivalent set of equations C.81, C.90, C.91, C.92 and C.101 was derived, which describe the PBFDAF with reversed time–domain buffers.

By stacking the weight update equation C.101 for all $p$, assuming $\mathbf{G} = \mathbf{I}_M$,

$$\begin{bmatrix} \breve{\underline{\mathbf{w}}}_0^{(n+1)} \\ \vdots \\ \breve{\underline{\mathbf{w}}}_{\frac{L_{FB}}{P}-1}^{(n+1)} \end{bmatrix} = \begin{bmatrix} \breve{\underline{\mathbf{w}}}_0^{(n)} \\ \vdots \\ \breve{\underline{\mathbf{w}}}_{\frac{L_{FB}}{P}-1}^{(n)} \end{bmatrix} + \begin{bmatrix} \breve{\boldsymbol{\Delta}} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \breve{\boldsymbol{\Delta}} \end{bmatrix} \begin{bmatrix} \breve{\underline{\mathbf{X}}}_0^{(n)^*} \\ \vdots \\ \breve{\underline{\mathbf{X}}}_{\frac{L_{FB}}{P}-1}^{(n)^*} \end{bmatrix} \mathbf{F}\mathbf{J}\mathbf{e}^{(n)} \tag{C.146}$$

is obtained in which

$$\mathbf{J}\mathbf{e}^{(n)} = \begin{bmatrix} \mathbf{I}_L & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{P-1+\sigma} \end{bmatrix}\mathbf{F}^{-1}\underline{\mathbf{e}}^{(n)} \tag{C.147}$$

and

$$\underline{\mathbf{e}}^{(n)} = \mathbf{F}\begin{bmatrix} d[(n+1)L] \\ \vdots \\ d[(n+1)L-M+1] \end{bmatrix} - \begin{bmatrix} \breve{\underline{\mathbf{X}}}_0^{(n)} & \dots & \breve{\underline{\mathbf{X}}}_{\frac{L_{FB}}{P}-1}^{(n)} \end{bmatrix} \begin{bmatrix} \breve{\underline{\mathbf{w}}}_0^{(n)} \\ \vdots \\ \breve{\underline{\mathbf{w}}}_{\frac{L_{FB}}{P}-1}^{(n)} \end{bmatrix} \tag{C.148}$$

follow from Eqs. C.81, C.91 and C.92.

By comparing Eqs. C.141 and 7.5 with C.148, and C.145 with C.146 and C.147, it is observed that the following conversion rules hold if $P = L$ :

$$\boxed{\textbf{fullband error}} \quad \longleftrightarrow \quad \boxed{\textbf{PBFDAF}} \tag{C.149}$$

$$N = P \tag{C.150}$$

$$L_{SB} = \frac{L_{FB}}{P} \tag{C.151}$$

$$\bar{\mathbf{w}}_n = \begin{bmatrix} \breve{\underline{\mathbf{w}}}_0^{(n)} \\ \vdots \\ \breve{\underline{\mathbf{w}}}_{\frac{L_{FB}}{P}-1}^{(n)} \end{bmatrix} \tag{C.152}$$

$$\bar{\mathbf{f}}_l^{(n)} = \breve{\underline{\mathbf{w}}}_l^{(n)} \tag{C.153}$$

$$L_p = 1 \tag{C.154}$$

$$\bar{\mathbf{X}}_n = \begin{bmatrix} \breve{\underline{\mathbf{X}}}_0^{(n)} & \dots & \breve{\underline{\mathbf{X}}}_{\frac{L_{FB}}{P}-1}^{(n)} \end{bmatrix} \tag{C.155}$$

$$\mathbf{X}[n-l] = \breve{\underline{\mathbf{X}}}_l^{(n)} \tag{C.156}$$

$$\mathbf{e}_n^{SB} = \underline{\mathbf{e}}^{(n)} \tag{C.157}$$

$$\mathbf{d}_n = \mathbf{F}\begin{bmatrix} d[(n+1)L] \\ \vdots \\ d[(n+1)L-M+1] \end{bmatrix} \tag{C.158}$$

$$\mathbf{y}_n = \begin{bmatrix} \breve{\underline{\mathbf{X}}}_0^{(n)} & \dots & \breve{\underline{\mathbf{X}}}_{\frac{L_{FB}}{P}-1}^{(n)} \end{bmatrix} \begin{bmatrix} \breve{\underline{\mathbf{w}}}_0^{(n)} \\ \vdots \\ \breve{\underline{\mathbf{w}}}_{\frac{L_{FB}}{P}-1}^{(n)} \end{bmatrix} \tag{C.159}$$

$$\frac{2}{M}\bar{\mathbf{\Delta}} = \begin{bmatrix} \breve{\mathbf{\Delta}} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \breve{\mathbf{\Delta}} \end{bmatrix}. \tag{C.160}$$

This proves that the weight update equation of the unconstrained PBFDAF, as it is mostly used in practice with $P = L$, corresponds to Eq. C.145 and hence to Eq. 7.33, which were derived based on fullband error feedback.

If $P$ is multiple of $L$ with $\frac{P}{L} > 1$, the equations do not match exactly anymore. In that case it suffices to replace $\bar{\mathbf{X}}_n$ in Eq. C.141 with

$$\bar{\mathbf{X}}_n = \begin{bmatrix} \mathbf{X}[n] & \mathbf{X}[n - \frac{P}{L}] & \dots & \mathbf{X}[n - (L_{SB} - 1)\frac{P}{L}], \end{bmatrix} \tag{C.161}$$

which corresponds to a slightly different fullband error adaptation algorithm. □

# D Appendix to part III

## D.1 Proof of theorem 8.1

**Theorem 8.1** *At the end of iteration step $R$ the error output $\boldsymbol{e}^{(n,R)}$ of the PBF-DRAP algorithm can be written as*

$$\mathbf{e}^{(n,R)} = \left( \mathbf{I}_M - \begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{G} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)^*} \mathbf{F} \right)^{R-1} \mathbf{e}^{(n,1)} \quad \text{(D.1)}$$

*and the adaptive filter weights are given by*

$$\underline{\mathbf{w}}_p^{(n,R+1)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n,1)} + \mathbf{G} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)^*} \mathbf{F} \sum_{r=1}^{R} \mathbf{e}^{(n,r)}, \quad p = 0 : \frac{L_{FB}}{P} - 1. \quad \text{(D.2)}$$

**Proof:** Taking into account Eq. 8.7, it is seen that Eqs. D.1 and D.2 are trivially fulfilled for $R = 1$. Now, assuming that Eqs. D.1 and D.2 hold for iteration step $R-1$, i.e.

$$\mathbf{e}^{(n,R-1)} = \left( \mathbf{I}_M - \begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{G} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)^*} \mathbf{F} \right)^{R-2} \mathbf{e}^{(n,1)} \quad \text{(D.3)}$$

and

$$\underline{\mathbf{w}}_p^{(n,R)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n,1)} + \mathbf{G} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)^*} \mathbf{F} \sum_{r=1}^{R-1} \mathbf{e}^{(n,r)}, \quad \text{(D.4)}$$

it should be proven that equation D.1 and D.2 also hold for iteration step $R$.

After the $(R-1)$–th iteration (see Eq. 8.7)

$$\underline{\mathbf{w}}_p^{(n,R)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n,R-1)} + \mathbf{G} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)^*} \mathbf{F} \mathbf{e}^{(n,R-1)}, \quad \text{(D.5)}$$

such that after iteration step $R$ (using Eqs. 8.5, 8.6, D.3 and D.5)

$$\mathbf{e}^{(n,R)} = \mathbf{d}^{(n)} - \begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \underline{\mathbf{w}}_p^{(n,R)} \quad \text{(D.6)}$$

$$= \mathbf{e}^{(n,R-1)} - \begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{G} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)^*} \mathbf{F} \mathbf{e}^{(n,R-1)} \quad \text{(D.7)}$$

$$= \left( \mathbf{I}_M - \begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{G} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)\,*} \mathbf{F} \right) \mathbf{e}^{(n,R-1)} \tag{D.8}$$

$$= \left( \mathbf{I}_M - \begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{G} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)\,*} \mathbf{F} \right)^{R-1} \mathbf{e}^{(n,1)} \tag{D.9}$$

is obtained and from Eqs. 8.7 and D.4

$$\underline{\mathbf{w}}_p^{(n,R+1)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n,R)} + \mathbf{G} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)\,*} \mathbf{F} \mathbf{e}^{(n,R)} \tag{D.10}$$

$$\overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n,1)} + \mathbf{G} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)\,*} \mathbf{F} \sum_{r=1}^{R} \mathbf{e}^{(n,r)} \tag{D.11}$$

is found, which completes the proof. Hence, Eqs. D.1 and D.2 are fulfilled for all $R \in \mathbb{N}_0$. □

## D.2 Proof of theorem 8.2

**Theorem 8.2** *For the unconstrained PBFDRAP*

$$\mathbf{P}_n = \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)\,*} \mathbf{F} \tag{D.12}$$

*is a right–circulant hermitian matrix having the Toeplitz structure. If the PBFDRAP is adapted in a real–life identification setup, $\boldsymbol{P}_n$ is a full–rank positive definite matrix. If furthermore all input signals and the unknown system $w$ are real–valued, $\boldsymbol{P}_n$ will be real–valued as well.*

**Proof:** Taking into account that $\mathbf{F}^H = M\mathbf{F}^{-1}$ and that $\boldsymbol{\Delta}$ and $\underline{\mathbf{X}}_p^{(n)}$ are diagonal matrices, it can be verified from Eq. D.12 that $\mathbf{P}_n = \mathbf{P}_n^H$, hence $\mathbf{P}_n$ is hermitian. As $\mathbf{F}\mathbf{P}_n\mathbf{F}^{-1}$ is diagonal, $\mathbf{P}_n$ is a right–circulant matrix (see appendix A) and therefore also Toeplitz. The diagonal elements of $\mathbf{F}\mathbf{P}_n\mathbf{F}^{-1}$ correspond to the eigenvalues of the hermitian matrix $\mathbf{P}_n$. Hence, they are all real. If the data $\underline{\mathbf{X}}_p^{(n)}$ come from a real–life scenario and if all stepsizes are different from zero, the diagonal elements of $\mathbf{F}\mathbf{P}_n\mathbf{F}^{-1}$ are strictly positive. Hence, $\mathbf{P}_n$ is a full rank positive definite matrix, i.e.

$$\forall \mathbf{x} \in \mathbb{C}_{\mathbf{0}}^M, \qquad \mathbf{x}^H \mathbf{P}_n \mathbf{x} > 0. \tag{D.13}$$

If $\mathbf{P}_n$ is multiplied with a random real vector $\mathbf{x}$,

$$\mathbf{P}_n\mathbf{x} = \mathbf{F}^{-1} \underbrace{\sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)*}}_{\boldsymbol{\Upsilon}} \underbrace{\mathbf{F}\mathbf{x}}_{\boldsymbol{\chi}} \tag{D.14}$$

is found. As $\mathbf{x}$ is real, its frequency–domain representation $\boldsymbol{\chi}$ has the following symmetry property[2] :

$$\begin{cases} \boldsymbol{\chi}(0) = \boldsymbol{\chi}^*(0) \\ \boldsymbol{\chi}(m) = \boldsymbol{\chi}^*(M-m), \quad \text{for } 0 < m < M. \end{cases} \tag{D.15}$$

Both $\boldsymbol{\Delta}$ and $\underline{\mathbf{X}}_p^{(n)}$ are diagonal matrices and as $x$, $d$ and $e$ (see figure 2.8) are assumed to be real–valued, the $m$–th and $(M-m)$–th diagonal elements of $\boldsymbol{\Delta}$ and $\underline{\mathbf{X}}_p^{(n)}$ are complex conjugated numbers. Therefore, $\boldsymbol{\Upsilon}$ is a real diagonal matrix and $\boldsymbol{\Upsilon}(m) = \boldsymbol{\Upsilon}(M-m)$ for $0 < m < M$.[3] If $\boldsymbol{\xi} = \boldsymbol{\Upsilon}\boldsymbol{\chi}$ is defined,

$$\begin{aligned} \boldsymbol{\xi}(m) &= \boldsymbol{\Upsilon}(m)\boldsymbol{\chi}(m) && \text{(D.16)} \\ &= \boldsymbol{\Upsilon}(M-m)\boldsymbol{\chi}^*(M-m) && \text{(D.17)} \\ &= \boldsymbol{\xi}^*(M-m) && \text{(D.18)} \end{aligned}$$

for $0 < m < M$ and $\boldsymbol{\xi}(0) = \boldsymbol{\xi}^*(0)$. Hence, $\mathbf{F}^{-1}\boldsymbol{\xi} = \mathbf{P}_n\mathbf{x}$ will be real–valued. As this holds for any random vector $\mathbf{x} \in \mathbb{R}^M$, the imaginary part of $\mathbf{P}_n$ must be zero. □

## D.3 Proof of theorem 8.3

**Theorem 8.3** *If the unconstrained PBFDRAP is adapted in a real–life identification setup, matrix $\boldsymbol{\mathcal{D}}$, as defined by Eq. 8.51, is hermitian and positive definite.*

**Proof :** By inserting Eq. 8.51 in Eq. 8.52 and making a special choice for $\mathbf{x}$ :

$$\forall \mathbf{x}_2 \in \mathbb{C}_{\mathbf{0}}^L, \qquad \begin{bmatrix} \mathbf{0} & \mathbf{x}_2^H \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{A}} & \boldsymbol{\mathcal{B}} \\ \boldsymbol{\mathcal{C}} & \boldsymbol{\mathcal{D}} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{x}_2 \end{bmatrix} > 0 \tag{D.19}$$

is obtained, which is equivalent to

$$\forall \mathbf{x}_2 \in \mathbb{C}_{\mathbf{0}}^L, \qquad \mathbf{x}_2^H \boldsymbol{\mathcal{D}} \mathbf{x}_2 > 0. \tag{D.20}$$

From Eq. D.20 it can be concluded that $\boldsymbol{\mathcal{D}}$ is positive definite. Furthermore, as $\mathbf{P}_n$ is a hermitian matrix, also $\boldsymbol{\mathcal{D}}$ is hermitian. □

[2] $\boldsymbol{\chi}(m)$ is the $m$–th element of vector $\boldsymbol{\chi}$.
[3] $\boldsymbol{\Upsilon}(m)$ is the $m$–th diagonal element of matrix $\boldsymbol{\Upsilon}$.

## D.4 Proof of theorem 8.4

**Theorem 8.4** *For the constrained PBFDRAP*

$$\mathbf{P}_n = \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{L-1+\sigma} \end{bmatrix} \mathbf{F}^{-1} \boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)^*} \mathbf{F} \tag{D.21}$$

*is a real–valued matrix if the adaptive filter is applied to a real–life identification setup as in figure 1.5, for which all input signals and the unknown system $w$ are real–valued.*

**Proof:** If $\mathbf{P}_n$ is multiplied with a random real vector $\mathbf{x}$,

$$\mathbf{P}_n \mathbf{x} = \sum_{p=0}^{\frac{L_{FB}}{P}-1} \mathbf{F}^{-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{L-1+\sigma} \end{bmatrix} \mathbf{F}^{-1} \underbrace{\boldsymbol{\Delta} \underline{\mathbf{X}}_p^{(n)^*}}_{\boldsymbol{\Upsilon}} \underbrace{\mathbf{F}\mathbf{x}}_{\boldsymbol{\chi}} \tag{D.22}$$

is obtained. Now $\boldsymbol{\chi}(m) = \boldsymbol{\chi}^*(M-m)$ and $\boldsymbol{\Upsilon}(m) = \boldsymbol{\Upsilon}^*(M-m)$ for $0 < m < M$ and $\boldsymbol{\chi}(0) = \boldsymbol{\chi}^*(0)$ and $\boldsymbol{\Upsilon}(0) = \boldsymbol{\Upsilon}^*(0)$ as all signals entering the adaptive filter are assumed to be real. If $\boldsymbol{\xi} = \boldsymbol{\Upsilon}\boldsymbol{\chi}$ is defined, $\boldsymbol{\xi}(m) = \boldsymbol{\Upsilon}(m)\boldsymbol{\chi}(m) = \boldsymbol{\xi}^*(M-m)$ for $0 < m < M$ and $\boldsymbol{\xi}(0) = \boldsymbol{\xi}^*(0)$. Hence, $\mathbf{F}^{-1}\boldsymbol{\xi}$ and also

$$\boldsymbol{\eta} = \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{L-1+\sigma} \end{bmatrix} \mathbf{F}^{-1} \boldsymbol{\xi} \tag{D.23}$$

will be real–valued. Using the same arguments, it is found that also $\mathbf{F}^{-1}\underline{\mathbf{X}}_p^{(n)}\mathbf{F}\boldsymbol{\eta}$ and hence $\mathbf{P}_n\mathbf{x}$ are real. As this holds for any random $\mathbf{x} \in \mathbb{R}^M$, the imaginary part of $\mathbf{P}_n$ must be zero. □

## D.5 Proof of theorem 8.5

**Theorem 8.5** *For the unnormalized and globally normalized constrained PBFDRAP for which $\boldsymbol{\Delta} = \mu_n \boldsymbol{I}_M$,*

$$\mathbf{P}_n = \mu_n \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{L-1+\sigma} \end{bmatrix} \mathbf{F}^{-1} \underline{\mathbf{X}}_p^{(n)^*} \mathbf{F} = \begin{bmatrix} \boldsymbol{\mathcal{A}} & \boldsymbol{\mathcal{B}} \\ \boldsymbol{\mathcal{C}} & \boldsymbol{\mathcal{D}} \end{bmatrix} \tag{D.24}$$

*is a hermitian and positive semi–definite matrix. Further, also $L \times L$ matrix $\boldsymbol{\mathcal{D}}$ is hermitian and positive semi–definite.*

**Proof:** It is easily verified that if $\boldsymbol{\Delta} = \mu_n \mathbf{I}_M$ matrix $\mathbf{P}_n$ is hermitian. Further, from Eq. D.24 it follows that $\forall \mathbf{x} \in \mathbb{C}^M$,

$$\mathbf{x}^H \mathbf{P}_n \mathbf{x} = \mu_n \sum_{p=0}^{\frac{L_{FB}}{P}-1} \mathbf{x}^H \mathbf{F}^{-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{L-1+\sigma} \end{bmatrix} \mathbf{F}^{-1} \underline{\mathbf{X}}_p^{(n)^*} \mathbf{F}\mathbf{x} \tag{D.25}$$

$$\mathbf{x}^H\mathbf{P}_n\mathbf{x} = \mu_n \sum_{p=0}^{\frac{L_{FB}}{P}-1} \left( \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{F}^{-1} \underline{\mathbf{X}}_p^{(n)^*} \mathbf{F}\mathbf{x} \right)^H \left( \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{F}^{-1} \underline{\mathbf{X}}_p^{(n)^*} \mathbf{F}\mathbf{x} \right) \tag{D.26}$$

$$\mathbf{x}^H\mathbf{P}_n\mathbf{x} \geqslant 0, \tag{D.27}$$

i.e. $\mathbf{P}_n$ is a positive semi–definite matrix. As $\mathbf{P}_n$ is hermitian and positive semi–definite all eigenvalues are real and positive. Now, for a special choice for vector $\mathbf{x}$ it holds that

$$\forall \mathbf{x}_2 \in \mathbb{C}^L, \qquad \begin{bmatrix} \mathbf{0} & \mathbf{x}_2^H \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{A}} & \boldsymbol{\mathcal{B}} \\ \boldsymbol{\mathcal{C}} & \boldsymbol{\mathcal{D}} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{x}_2 \end{bmatrix} \geqslant 0, \tag{D.28}$$

which is equivalent to

$$\forall \mathbf{x}_2 \in \mathbb{C}^L, \qquad \mathbf{x}_2^H \boldsymbol{\mathcal{D}} \mathbf{x}_2 \geqslant 0. \tag{D.29}$$

From Eq. D.29 it can be concluded that $\boldsymbol{\mathcal{D}}$ is positive semi–definite. Furthermore, $\boldsymbol{\mathcal{D}}$ is hermitian, just as $\mathbf{P}_n$. Hence, the eigenvalues of $\boldsymbol{\mathcal{D}}$, i.e. the diagonal elements of $\boldsymbol{\Gamma}_1$, are real and positive. □

## D.6 Constrained PBFDRAP : $L_{FB} < L$

Remark that if $L_{FB} < L$ the rank of $\mathbf{N}_n$ (Eq. 8.60) is $\min(\min(M, L_{FB}), L) = L_{FB}$ if $\mu_n \neq 0$. Hence, $\boldsymbol{\Gamma}_1$ (Eq. 8.18) is not of full rank and can be expressed as

$$\boldsymbol{\Gamma}_1 = \begin{bmatrix} \boldsymbol{\Gamma}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{L-L_{FB}} \end{bmatrix}, \tag{D.30}$$

with $\boldsymbol{\Gamma}_{11}$ a full rank $L_{FB} \times L_{FB}$ diagonal matrix.

Based on Eqs. 8.5–8.6, taking into account that $d = w \star x$ and partitioning both the unknown system $w$ as its model $\hat{w}^{(n)}$ according to Eq. 8.2, $\mathbf{e}^{(n,1)}$ can be written as

$$\mathbf{e}^{(n,1)} = \begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \underline{\tilde{\mathbf{w}}}_p^{(n)}, \tag{D.31}$$

with $\underline{\tilde{\mathbf{w}}}_p^{(n)}$ $M \times 1$ vectors of the form

$$\underline{\tilde{\mathbf{w}}}_p^{(n)} \overset{\forall p}{=} \mathbf{F} \begin{bmatrix} \tilde{\mathbf{w}}_p^{(n)} \\ \mathbf{0} \end{bmatrix} \begin{matrix} \updownarrow P \\ \updownarrow L-1+\sigma \end{matrix}, \quad p = 0 : \frac{L_{FB}}{P} - 1 \tag{D.32}$$

and the $j$–th element of $\tilde{\mathbf{w}}_p^{(n)}$ such that

$$\tilde{\mathbf{w}}_p^{(n)}(j) = w[j + pP] - \hat{w}^{(n)}[j + pP], \qquad p = 0 : \frac{L_{FB}}{P} - 1. \tag{D.33}$$

Equation D.31 can also be written as

$$\mathbf{e}^{(n,1)} = \begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{L-1+\sigma} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{w}}_p^{(n)} \\ \mathbf{0} \end{bmatrix}. \tag{D.34}$$

**Theorem D.1** *It can be shown that an $M \times 1$ vector $\mathbf{v}_n$ can be found, such that*

$$\tilde{\mathbf{w}}_p^{(n)} \stackrel{\forall p}{=} \left[ \begin{array}{cc} \mathbf{I}_P & \mathbf{0} \end{array} \right] \mathbf{F}^{-1} \underline{\mathbf{X}}_p^{(n)^*} \mathbf{F} \mathbf{v}_n. \tag{D.35}$$

**Proof:** Taking into account Eq. 8.1, equation D.35 can be rewritten as

$$\underbrace{\left[ \begin{array}{c} \tilde{\mathbf{w}}_0^{(n)} \\ \vdots \\ \tilde{\mathbf{w}}_{\frac{L_{FB}}{P}-1}^{(n)} \end{array} \right]}_{\tilde{\mathbf{w}}_n} = \underbrace{\left[ \begin{array}{ccccc} \mathbf{I}_P & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_P & \mathbf{0} \end{array} \right]}_{\mathbf{L}} \underbrace{\left[ \begin{array}{c} \mathbf{F}^{-1} \underline{\mathbf{X}}_0^{(n)^*} \mathbf{F} \\ \vdots \\ \mathbf{F}^{-1} \underline{\mathbf{X}}_{\frac{L_{FB}}{P}-1}^{(n)^*} \mathbf{F} \end{array} \right]}_{\mathbf{R}_n} \mathbf{v}_n. \tag{D.36}$$

For randomly chosen data the $\left(\frac{L_{FB}}{P} M\right) \times M$ matrix $\mathbf{R}_n$ has full rank $M$ as $L_{FB}$ is always equal to or larger than $P$, whereas the rank of the $L_{FB} \times \left(\frac{L_{FB}}{P} M\right)$ matrix $\mathbf{L}$ is $P \frac{L_{FB}}{P} = L_{FB}$. Hence, the $L_{FB} \times M$ matrix $\mathbf{L}\mathbf{R}_n$ has full rank $\min(L_{FB}, M) = L_{FB}$ as $L_{FB} < L \leqslant M$.
For every $\tilde{\mathbf{w}}_n$ an $M \times 1$ vector $\mathbf{v}_n$ can be found as the product of the pseudo–inverse of $\mathbf{L}\mathbf{R}_n$ and vector $\tilde{\mathbf{w}}_n$ :

$$\mathbf{v}_n = (\mathbf{L}\mathbf{R}_n)^\dagger \, \tilde{\mathbf{w}}_n, \tag{D.37}$$

which completes the proof. □

Hence Eq. D.34 becomes (see also Eq. 8.17)

$$\mathbf{e}^{(n,1)} = \left[ \begin{array}{cc} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{array} \right] \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{F} \left[ \begin{array}{cc} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{L-1+\sigma} \end{array} \right] \mathbf{F}^{-1} \underline{\mathbf{X}}_p^{(n)^*} \mathbf{F} \mathbf{v}_n \tag{D.38}$$

$$= \frac{1}{\mu_n} \mathbf{N}_n \mathbf{v}_n. \tag{D.39}$$

Now, by inserting Eqs. 8.18, D.30 and D.39 in equation 8.25,

$$\sum_{r=1}^{R} \mathbf{e}^{(n,r)} = \frac{1}{\mu_n} \mathbf{V} \sum_{r=1}^{R} \left(\mathbf{I}_M - \mathbf{\Gamma}\right)^{r-1} \mathbf{V}^{-1} \mathbf{N}_n \mathbf{v}_n \tag{D.40}$$

$$= \frac{1}{\mu_n} \mathbf{V} \sum_{r=1}^{R} \left[ \begin{array}{ccc} \mathbf{I}_{L_{FB}} - \mathbf{\Gamma}_{11} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{L-L_{FB}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{P-1+\sigma} \end{array} \right]^{r-1} \cdot \left[ \begin{array}{ccc} \mathbf{\Gamma}_{11} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{L-L_{FB}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{P-1+\sigma} \end{array} \right] \mathbf{V}^{-1} \mathbf{v}_n \tag{D.41}$$

$$= \frac{1}{\mu_n} \mathbf{V} \left[ \begin{array}{ccc} \mathbf{\Gamma}_{11}^{-1} \left( \mathbf{I}_{L_{FB}} - \left(\mathbf{I}_{L_{FB}} - \mathbf{\Gamma}_{11}\right)^R \right) \mathbf{\Gamma}_{11} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{L-L_{FB}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{P-1+\sigma} \end{array} \right] \mathbf{V}^{-1} \mathbf{v}_n \tag{D.42}$$

is obtained. As $\boldsymbol{\mathcal{D}}$ is positive semi–definite and hermitian (Eq. D.29) and $\boldsymbol{\Gamma}_{11}$ is of full rank, all elements of $\boldsymbol{\Gamma}_{11}$ are real and strictly positive. Hence, if $\mu_n$ is sufficiently small

$$\lim_{R\to\infty}\sum_{r=1}^{R}\mathbf{e}^{(n,r)} = \frac{1}{\mu_n}\mathbf{V}\begin{bmatrix}\mathbf{I}_{L_{FB}} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}_{L-L_{FB}} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{P-1+\sigma}\end{bmatrix}\mathbf{V}^{-1}\mathbf{v}_n \tag{D.43}$$

and the filter weights (Eq. 8.23) converge to

$$\lim_{R\to\infty}\underline{\mathbf{w}}_p^{(n,R)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n,1)} + \mathbf{G}\underline{\mathbf{X}}_p^{(n)\,*}\mathbf{F}\mathbf{V}\begin{bmatrix}\mathbf{I}_{L_{FB}} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}_{M-L_{FB}}\end{bmatrix}\mathbf{V}^{-1}\mathbf{v}_n. \tag{D.44}$$

## D.7 Proof of theorem 8.7

**Theorem 8.7** *The unnormalized and globally normalized constrained PBFDRAP for which $L_{FB} \geqslant L$ approach the Partial Rank Algorithm (PRA) by series expansion of the matrix inverse in Eq. 2.55.*

**Proof:** Consider the constrained PBFDRAP and assume that the stepsize matrix is given by $\boldsymbol{\Delta} = \mu_n\mathbf{I}_M$. Plugging in Eq. 8.22, which holds for all $R > 0$, in Eq. 8.23, leads to

$$\underline{\mathbf{w}}_p^{(n,R+1)} \overset{\forall p}{=} \underline{\mathbf{w}}_p^{(n,1)} + \mu_n\mathbf{F}\begin{bmatrix}\mathbf{I}_P & \mathbf{0}\\ \mathbf{0} & \mathbf{0}_{L-1+\sigma}\end{bmatrix}\mathbf{F}^{-1}\underline{\mathbf{X}}_p^{(n)\,*}\mathbf{F}\sum_{r=1}^{R}\mathbf{B}_n^{r-1}\mathbf{e}^{(n,1)} \tag{D.45}$$

in which

$$\mathbf{B}_n = \mathbf{I}_M - \mu_n\begin{bmatrix}\mathbf{0}_{P-1+\sigma} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}_L\end{bmatrix}\mathbf{F}^{-1}\sum_{p=0}^{\frac{L_{FB}}{P}-1}\underline{\mathbf{X}}_p^{(n)}\mathbf{F}\begin{bmatrix}\mathbf{I}_P & \mathbf{0}\\ \mathbf{0} & \mathbf{0}_{L-1+\sigma}\end{bmatrix}\mathbf{F}^{-1}\underline{\mathbf{X}}_p^{(n)\,*}\mathbf{F}. \tag{D.46}$$

From Eqs. 8.2 and 8.8 it is observed that

$$\underline{\mathbf{w}}_p^{(n,R+1)} \overset{\forall p}{=} \mathbf{F}\begin{bmatrix}\mathbf{w}_p^{(n+1)}\\ \mathbf{0}\end{bmatrix} \tag{D.47}$$

and that

$$\underline{\mathbf{w}}_p^{(n,1)} \overset{\forall p}{=} \mathbf{F}\begin{bmatrix}\mathbf{w}_p^{(n)}\\ \mathbf{0}\end{bmatrix}. \tag{D.48}$$

Hence, inserting Eqs. D.47, D.48 and 8.30 into Eq. D.45 gives

$$\mathbf{F}\begin{bmatrix}\mathbf{w}_p^{(n+1)}\\ \mathbf{0}\end{bmatrix} \overset{\forall p}{=} \mathbf{F}\left(\begin{bmatrix}\mathbf{w}_p^{(n)}\\ \mathbf{0}\end{bmatrix} + \mu_n\begin{bmatrix}\mathbf{I}_P & \mathbf{0}\\ \mathbf{0} & \mathbf{0}_{L-1+\sigma}\end{bmatrix}\mathbf{F}^{-1}\underline{\mathbf{X}}_p^{(n)\,*}\mathbf{F}\sum_{r=1}^{R}\mathbf{B}_n^{r-1}\begin{bmatrix}\mathbf{0}\\ \mathbf{I}_L\end{bmatrix}\mathbf{e}_n\right). \tag{D.49}$$

As $\underline{\mathbf{X}}_p^{(n)}$ is a diagonal matrix, defined by Eq. 8.3, $\bar{\mathbf{X}}_{n,p}^T = \mathbf{F}^{-1}\underline{\mathbf{X}}_p^{(n)}\mathbf{F}$ is a right–circulant $M \times M$ matrix (cf. Eqs. A.3 and A.4). Hence, the discrete Fourier transform of the first row of $\bar{\mathbf{X}}_{n,p}$ are the diagonal elements of $\underline{\mathbf{X}}_p^{(n)}$. Taking into account Eq. 8.3,

$$\bar{\mathbf{X}}_{n,p} = (\mathbf{F}^{-1}\underline{\mathbf{X}}_p^{(n)}\mathbf{F})^T \tag{D.50}$$

$$= \begin{bmatrix} x[(n+1)L-pP-M+1] & \dots & x[(n+1)L-pP] \\ x[(n+1)L-pP] & \dots & x[(n+1)L-pP-1] \\ \vdots & \ddots & \vdots \\ x[(n+1)L-pP-M+2] & \dots & x[(n+1)L-pP-M+1] \end{bmatrix} \tag{D.51}$$

is found. Hence,

$$\bar{\mathbf{X}}_{n,p}^* = (\mathbf{F}^{-1}\underline{\mathbf{X}}_p^{(n)}\mathbf{F})^H = \mathbf{F}^{-1}\underline{\mathbf{X}}_p^{(n)^*}\mathbf{F}. \tag{D.52}$$

From equation D.52 it follows that Eq. D.49 can be written as

$$\mathbf{w}_p^{(n+1)} \overset{\forall p}{=} \mathbf{w}_p^{(n)} + \mu_n \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \end{bmatrix} \bar{\mathbf{X}}_{n,p}^* \sum_{r=1}^{R} \mathbf{B}_n^{r-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} \mathbf{e}_n. \tag{D.53}$$

Now, from Eq. D.46 it is observed that

$$\mathbf{B}_n \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} = \left( \mathbf{I}_M - \mu_n \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{F}^{-1} \underline{\mathbf{X}}_p^{(n)^*} \mathbf{F} \right) \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} \tag{D.54}$$

$$= \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} - \mu_n \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \mathbf{F}^{-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{F}^{-1} \underline{\mathbf{X}}_p^{(n)^*} \mathbf{F} \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix}. \tag{D.55}$$

Recall that $\boldsymbol{\mathcal{D}}$ was defined by equation 8.16 and that $\boldsymbol{\mathcal{D}}$ therefore can be written as

$$\boldsymbol{\mathcal{D}} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{P}_n \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix}. \tag{D.56}$$

Hence, Eq. D.55 becomes

$$\mathbf{B}_n \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L - \boldsymbol{\mathcal{D}} \end{bmatrix}. \tag{D.57}$$

From Eq. D.57 it follows that

$$\mathbf{B}_n^r \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} = \mathbf{B}_n^{r-1} \mathbf{B}_n \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} \tag{D.58}$$

$$= \mathbf{B}_n^{r-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L - \boldsymbol{\mathcal{D}} \end{bmatrix} \tag{D.59}$$

$$= \mathbf{B}_n^{r-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} (\mathbf{I}_L - \boldsymbol{\mathcal{D}}) \tag{D.60}$$

$$= \mathbf{B}_n^{r-2} \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L - \boldsymbol{\mathcal{D}} \end{bmatrix} (\mathbf{I}_L - \boldsymbol{\mathcal{D}}) . \tag{D.61}$$

By recursion

$$\mathbf{B}_n^r \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} (\mathbf{I}_L - \boldsymbol{\mathcal{D}})^r \tag{D.62}$$

is obtained.

Now, from Eqs. 8.16, 8.22, D.51, D.52 and D.56

$$\boldsymbol{\mathcal{D}} = \mu_n \begin{bmatrix} \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{F}^{-1} \underline{\mathbf{X}}_p^{(n)*} \mathbf{F} \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} \tag{D.63}$$

$$= \mu_n \sum_{p=0}^{\frac{L_{FB}}{P}-1} \begin{bmatrix} \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{F}^{-1} \underline{\mathbf{X}}_p^{(n)*} \mathbf{F} \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} \tag{D.64}$$

$$= \mu_n \sum_{p=0}^{\frac{L_{FB}}{P}-1} \left( \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \bar{\mathbf{X}}_{n,p}^* \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} \right)^H \left( \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \bar{\mathbf{X}}_{n,p}^* \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} \right) \tag{D.65}$$

$$= \mu_n \sum_{p=0}^{\frac{L_{FB}}{P}-1} \mathbf{X}_{n,p}^T \mathbf{X}_{n,p}^* \tag{D.66}$$

is found in which $\mathbf{X}_{n,p}$ is the following $P \times L$ Toeplitz matrix (cf. Eq. 6.4) :

$$\mathbf{X}_{n,p} = \begin{bmatrix} x[nL - pP + 1] & \dots & x[(n+1)L - pP] \\ \vdots & \ddots & \vdots \\ x[nL - pP - P + 2] & \dots & x[(n+1)L - pP - P + 1] \end{bmatrix} . \tag{D.67}$$

Observe that the rows of $\mathbf{X}_{n,p}$ are delayed versions of each other, only if the time index of the left bottom element of $\mathbf{X}_{n,p}$ is larger than or equal to the time index of the left top element of $\bar{\mathbf{X}}_{n,p}$ in Eq. D.51. This implies that

$$(n+1)L - pP - M + 1 \leqslant nL - pP - P + 2, \tag{D.68}$$

which is equivalent to

$$M \geqslant P + L - 1, \tag{D.69}$$

being the well–known condition 6.11 that must be true for the PBFDAF.

Finally, Eq. D.66 becomes

$$\boldsymbol{\mathcal{D}} \;=\; \mu_n \mathbf{X}_n^T \mathbf{X}_n^* \tag{D.70}$$

where

$$\mathbf{X}_n = \begin{bmatrix} x[nL+1] & \dots & x[(n+1)L] \\ \vdots & \ddots & \vdots \\ x[nL-L_{FB}+2] & \dots & x[(n+1)L-L_{FB}+1] \end{bmatrix} \tag{D.71}$$

is as defined in Eq. 2.52.

By plugging in Eqs. D.70 and D.62, Eq. D.53 becomes

$$\mathbf{w}_p^{(n+1)} \overset{\forall p}{=} \mathbf{w}_p^{(n)} + \mu_n \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \end{bmatrix} \bar{\mathbf{X}}_{n,p}^* \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} \sum_{r=1}^{R} \left( \mathbf{I}_L - \mu_n \mathbf{X}_n^T \mathbf{X}_n^* \right)^{r-1} \mathbf{e}_n \tag{D.72}$$

$$\overset{\forall p}{=} \mathbf{w}_p^{(n)} + \mu_n \mathbf{X}_{n,p}^* \sum_{r=1}^{R} \left( \mathbf{I}_L - \mu_n \mathbf{X}_n^T \mathbf{X}_n^* \right)^{r-1} \mathbf{e}_n \tag{D.73}$$

in which $\mathbf{X}_{n,p}$ is defined in Eq. D.67 and $\mathbf{X}_n$ in Eq. D.71.

By stacking Eq. D.73 for all $p$

$$\begin{bmatrix} \mathbf{w}_0^{(n+1)} \\ \vdots \\ \mathbf{w}_{\frac{L_{FB}}{P}-1}^{(n+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{w}_0^{(n)} \\ \vdots \\ \mathbf{w}_{\frac{L_{FB}}{P}-1}^{(n)} \end{bmatrix} + \mu_n \begin{bmatrix} \mathbf{X}_{n,0}^* \\ \vdots \\ \mathbf{X}_{n,\frac{L_{FB}}{P}-1}^* \end{bmatrix} \sum_{r=1}^{R} \left( \mathbf{I}_L - \mu_n \mathbf{X}_n^T \mathbf{X}_n^* \right)^{r-1} \mathbf{e}_n \tag{D.74}$$

$$\mathbf{w}_{n+1} \;=\; \mathbf{w}_n + \mu_n \mathbf{X}_n^* \sum_{r=1}^{R} \left( \mathbf{I}_L - \mu_n \mathbf{X}_n^T \mathbf{X}_n^* \right)^{r-1} \mathbf{e}_n \tag{D.75}$$

is obtained. As it is assumed that $L_{FB} \geqslant L$ and that real–life data are used, $\mathbf{X}_n^T \mathbf{X}_n^*$ is invertible. Hence,

$$\mathbf{w}_{n+1} \;=\; \mathbf{w}_n + \mu_n \mathbf{X}_n^* \left( \mu_n \mathbf{X}_n^T \mathbf{X}_n^* \right)^{-1} \left( \mathbf{I}_L - \left( \mathbf{I}_L - \mu_n \mathbf{X}_n^T \mathbf{X}_n^* \right)^R \right) \mathbf{e}_n \tag{D.76}$$

$$\;=\; \mathbf{w}_n + \mathbf{X}_n^* \left( \mathbf{X}_n^T \mathbf{X}_n^* \right)^{-1} \left( \mathbf{I}_L - \left( \mathbf{I}_L - \mu_n \mathbf{X}_n^T \mathbf{X}_n^* \right)^R \right) \mathbf{e}_n. \tag{D.77}$$

$\mathbf{X}_n^T \mathbf{X}_n^*$ is a hermitian positive definite matrix if the data are obtained from a real–life identification scenario and if $L_{FB} \geqslant L$. Hence, if the stepsize $\mu_n$ is small enough, all the eigenvalues of $\mathbf{I}_L - \mu_n \mathbf{X}_n^T \mathbf{X}_n^*$ lie within the unit circle.

Further, from Eqs. 8.2, 8.4, 8.5, 8.6, 8.30 and D.50

$$\mathbf{e}_n = \mathbf{d}_n - \begin{bmatrix}\mathbf{0} & \mathbf{I}_L\end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \underline{\mathbf{X}}_p^{(n)} \underline{\mathbf{w}}_p^{(n,1)} \tag{D.78}$$

$$= \mathbf{d}_n - \begin{bmatrix}\mathbf{0} & \mathbf{I}_L\end{bmatrix} \sum_{p=0}^{\frac{L_{FB}}{P}-1} \mathbf{F}^{-1} \underline{\mathbf{X}}_p^{(n)} \mathbf{F} \begin{bmatrix} \mathbf{w}_p^{(n)} \\ \mathbf{0} \end{bmatrix} \tag{D.79}$$

$$= \mathbf{d}_n - \sum_{p=0}^{\frac{L_{FB}}{P}-1} \begin{bmatrix}\mathbf{0} & \mathbf{I}_L\end{bmatrix} \bar{\mathbf{X}}_{n,p}^T \begin{bmatrix} \mathbf{w}_p^{(n)} \\ \mathbf{0} \end{bmatrix} \tag{D.80}$$

$$= \mathbf{d}_n - \sum_{p=0}^{\frac{L_{FB}}{P}-1} \mathbf{X}_{n,p}^T \mathbf{w}_p^{(n)} \tag{D.81}$$

is obtained and finally using Eqs. D.74 and D.75 we find

$$\mathbf{e}_n = \mathbf{d}_n - \mathbf{X}_n^T \mathbf{w}_n. \tag{D.82}$$

If $R$ goes to infinity

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mathbf{X}_n^* \left( \mathbf{X}_n^T \mathbf{X}_n^* \right)^{-1} \mathbf{e}_n \tag{D.83}$$

is obtained from Eq. D.77, which confirms the conclusion made based on theorem 8.6 : by comparing Eq. 2.54 with D.82 and Eq. 2.55 with D.83, it is verified that if $\hat{\mathbf{w}}_0 = \mathbf{w}_0$, also $\hat{\mathbf{w}}_n = \mathbf{w}_n, \forall n$ and $\mu_n = 1$. Hence, the unnormalized and globally normalized constrained PBFDRAP converge to the PRA if $L_{FB} \geqslant L$ and the number of iteration steps goes to infinity.

From Eq. D.75 it follows that the unnormalized and globally normalized constrained PBFDRAP for which $L_{FB} \geqslant L$ approach the PRA by series expansion of the matrix inverse in Eq. 2.55. □