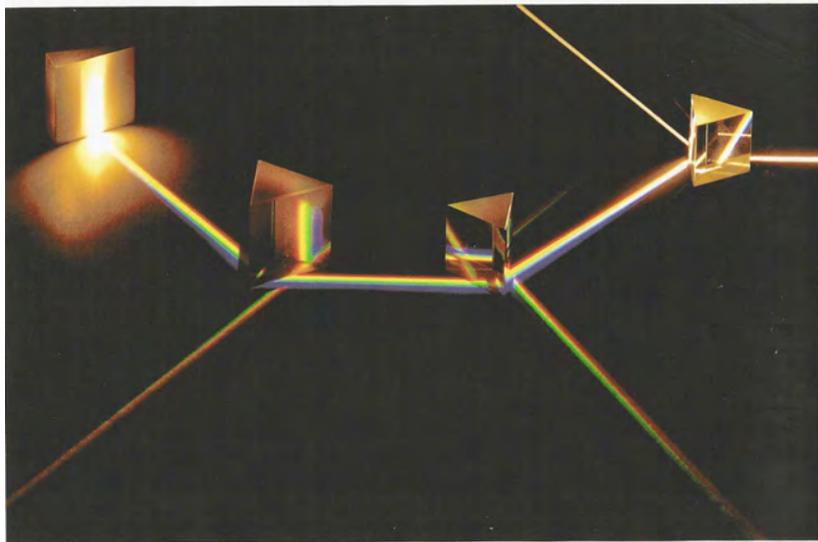


## Fourier and Wavelet Signal Processing



Jelena Kovačević

*Carnegie Mellon University*

Vivek K Goyal

*Massachusetts Institute of Technology*

Martin Vetterli

*École Polytechnique Fédérale de Lausanne*

January 17, 2013

**Copyright (c) 2013 Jelena Kovačević, Vivek K Goyal, and Martin Vetterli.**

**These materials are protected by copyright under the  
Attribution-NonCommercial-NoDerivs 3.0 Unported License  
from Creative Commons.**

Cover photograph by Christiane Grimm, Geneva, Switzerland.

Experimental set up by Prof. Libero Zuppiroli and Philippe Bugnon, Laboratory of Optoelectronics Molecular Materials, EPFL, Lausanne, Switzerland.

The cover photograph captures the experiment first described by Isaac Newton in *Opticks* in 1730, showing that white light can be split into its color components and then synthesized back into white light. It is a physical implementation of a decomposition of white light into its Fourier components—the colors of the rainbow, followed by a synthesis to recover the original.

# Contents

<b>Image Attribution</b>	<b>ix</b>
<b>Quick Reference</b>	<b>xi</b>
<b>Preface</b>	<b>xvii</b>
<b>Acknowledgments</b>	<b>xxi</b>
<b>1 Filter Banks: Building Blocks of Time-Frequency Expansions</b>	<b>1</b>
1.1 Introduction	2
1.2 Orthogonal Two-Channel Filter Banks	7
1.2.1 A Single Channel and Its Properties	7
1.2.2 Complementary Channels and Their Properties	10
1.2.3 Orthogonal Two-Channel Filter Bank	11
1.2.4 Polyphase View of Orthogonal Filter Banks	14
1.2.5 Polynomial Approximation by Filter Banks	18
1.3 Design of Orthogonal Two-Channel Filter Banks	20
1.3.1 Lowpass Approximation Design	20
1.3.2 Polynomial Approximation Design	21
1.3.3 Lattice Factorization Design	25
1.4 Biorthogonal Two-Channel Filter Banks	26
1.4.1 A Single Channel and Its Properties	29
1.4.2 Complementary Channels and Their Properties	32
1.4.3 Biorthogonal Two-Channel Filter Bank	32
1.4.4 Polyphase View of Biorthogonal Filter Banks	34
1.4.5 Linear-Phase Two-Channel Filter Banks	35
1.5 Design of Biorthogonal Two-Channel Filter Banks	37
1.5.1 Factorization Design	37
1.5.2 Complementary Filter Design	39
1.5.3 Lifting Design	40
1.6 Two-Channel Filter Banks with Stochastic Inputs	41
1.7 Computational Aspects	42
1.7.1 Two-Channel Filter Banks	42
1.7.2 Boundary Extensions	46

iv		Contents
	Chapter at a Glance	49
	Historical Remarks	52
	Further Reading	52
<b>2</b>	<b>Local Fourier Bases on Sequences</b>	<b>55</b>
2.1	Introduction	56
2.2	$N$ -Channel Filter Banks	59
2.2.1	Orthogonal $N$ -Channel Filter Banks	59
2.2.2	Polyphase View of $N$ -Channel Filter Banks	61
2.3	Complex Exponential-Modulated Local Fourier Bases	66
2.3.1	Balian-Low Theorem	67
2.3.2	Application to Power Spectral Density Estimation	68
2.3.3	Application to Communications	73
2.4	Cosine-Modulated Local Fourier Bases	75
2.4.1	Lapped Orthogonal Transforms	76
2.4.2	Application to Audio Compression	83
2.5	Computational Aspects	85
	Chapter at a Glance	87
	Historical Remarks	88
	Further Reading	90
<b>3</b>	<b>Wavelet Bases on Sequences</b>	<b>91</b>
3.1	Introduction	93
3.2	Tree-Structured Filter Banks	99
3.2.1	The Lowpass Channel and Its Properties	99
3.2.2	Bandpass Channels and Their Properties	103
3.2.3	Relationship between Lowpass and Bandpass Channels	105
3.3	Orthogonal Discrete Wavelet Transform	106
3.3.1	Definition of the Orthogonal DWT	106
3.3.2	Properties of the Orthogonal DWT	107
3.4	Biorthogonal Discrete Wavelet Transform	112
3.4.1	Definition of the Biorthogonal DWT	112
3.4.2	Properties of the Biorthogonal DWT	114
3.5	Wavelet Packets	114
3.5.1	Definition of the Wavelet Packets	115
3.5.2	Properties of the Wavelet Packets	116
3.6	Computational Aspects	116
	Chapter at a Glance	118
	Historical Remarks	119
	Further Reading	119
<b>4</b>	<b>Local Fourier and Wavelet Frames on Sequences</b>	<b>121</b>
4.1	Introduction	122
4.2	Finite-Dimensional Frames	133
4.2.1	Tight Frames for $\mathbb{C}^N$	133
4.2.2	General Frames for $\mathbb{C}^N$	140

Contents	v	
4.2.3	Choosing the Expansion Coefficients	145
4.3	Oversampled Filter Banks	151
4.3.1	Tight Oversampled Filter Banks	152
4.3.2	Polyphase View of Oversampled Filter Banks	155
4.4	Local Fourier Frames	158
4.4.1	Complex Exponential-Modulated Local Fourier Frames	159
4.4.2	Cosine-Modulated Local Fourier Frames	163
4.5	Wavelet Frames	165
4.5.1	Oversampled DWT	165
4.5.2	Pyramid Frames	167
4.5.3	Shift-Invariant DWT	170
4.6	Computational Aspects	171
4.6.1	The Algorithm à Trous	171
4.6.2	Efficient Gabor and Spectrum Computation	171
4.6.3	Efficient Sparse Frame Expansions	171
	Chapter at a Glance	173
	Historical Remarks	174
	Further Reading	174
<b>5</b>	<b>Local Fourier Transforms, Frames and Bases on Functions</b>	<b>177</b>
5.1	Introduction	178
5.2	Local Fourier Transform	178
5.2.1	Definition of the Local Fourier Transform	178
5.2.2	Properties of the Local Fourier Transform	182
5.3	Local Fourier Frame Series	188
5.3.1	Sampling Grids	188
5.3.2	Frames from Sampled Local Fourier Transform	188
5.4	Local Fourier Series	188
5.4.1	Complex Exponential-Modulated Local Fourier Bases	188
5.4.2	Cosine-Modulated Local Fourier Bases	188
5.5	Computational Aspects	188
5.5.1	Complex Exponential-Modulated Local Fourier Bases	188
5.5.2	Cosine-Modulated Local Fourier Bases	188
	Chapter at a Glance	188
	Historical Remarks	188
	Further Reading	188
<b>6</b>	<b>Wavelet Bases, Frames and Transforms on Functions</b>	<b>189</b>
6.1	Introduction	189
6.1.1	Scaling Function and Wavelets from Haar Filter Bank	190
6.1.2	Haar Wavelet Series	195
6.1.3	Haar Frame Series	202
6.1.4	Haar Continuous Wavelet Transform	204
6.2	Scaling Function and Wavelets from Orthogonal Filter Banks	208
6.2.1	Iterated Filters	208
6.2.2	Scaling Function and its Properties	209

6.2.3	Wavelet Function and its Properties	218
6.2.4	Scaling Function and Wavelets from Biorthogonal Filter Banks	220
6.3	Wavelet Series	222
6.3.1	Definition of the Wavelet Series	223
6.3.2	Properties of the Wavelet Series	227
6.3.3	Multiresolution Analysis	230
6.3.4	Biorthogonal Wavelet Series	239
6.4	Wavelet Frame Series	242
6.4.1	Definition of the Wavelet Frame Series	242
6.4.2	Frames from Sampled Wavelet Series	242
6.5	Continuous Wavelet Transform	242
6.5.1	Definition of the Continuous Wavelet Transform	242
6.5.2	Existence and Convergence of the Continuous Wavelet Transform	243
6.5.3	Properties of the Continuous Wavelet Transform	244
6.6	Computational Aspects	254
6.6.1	Wavelet Series: Mallat's Algorithm	254
6.6.2	Wavelet Frames	259
	Chapter at a Glance	259
	Historical Remarks	259
	Further Reading	259
<b>7</b>	<b>Approximation, Estimation, and Compression</b>	<b>261</b>
7.1	Introduction	262
7.2	Abstract Models and Approximation	262
7.2.1	Local Fourier and Wavelet Approximations of Piecewise Smooth Functions	262
7.2.2	Wide-Sense Stationary Gaussian Processes	262
7.2.3	Poisson Processes	262
7.3	Empirical Models	262
7.3.1	$\ell^p$ Models	262
7.3.2	Statistical Models	262
7.4	Estimation and Denoising	262
7.4.1	Connections to Approximation	262
7.4.2	Wavelet Thresholding and Variants	262
7.4.3	Frames	262
7.5	Compression	262
7.5.1	Audio Compression	262
7.5.2	Image Compression	262
7.6	Inverse Problems	262
7.6.1	Deconvolution	262
7.6.2	Compressed Sensing	262
	Chapter at a Glance	262
	Historical Remarks	262
	Further Reading	262

<b>Contents</b>	<b>vii</b>
<hr/>	
7.A Elements of Source Coding	263
7.A.1 Entropy Coding	263
7.A.2 Quantization	263
7.A.3 Transform Coding	263
<b>Bibliography</b>	<b>265</b>



# Image Attribution

Source	Permission
<sup>s1</sup> Christiane Grimm, Geneva, Switzerland	<sup>p1</sup> Permission of the copyright holder
<sup>s2</sup> Wikimedia Commons	<sup>p2</sup> Copyright expired
<sup>s3</sup> Nobelprize.org	<sup>p3</sup> GNU Free Documentation license
<sup>s4</sup> KerryR.net	<sup>p4</sup> Public domain
<sup>s5</sup> Edward Lee's home page	<sup>p5</sup> Common property, no original authorship
	<sup>p6</sup> NASA material, not protected by copyright
	<sup>p7</sup> Public domain in the US
	<sup>p8</sup> Unknown rights

## Front Material

- *Cover photograph*.<sup>s1,p1</sup> Experimental set up by Prof. Libero Zuppiroli, Laboratory of Optoelectronics Molecular Materials, EPFL, Lausanne, Switzerland.

x

Image Attribution

# Quick Reference

## Abbreviations

AR	Autoregressive
ARMA	Autoregressive moving average
AWGN	Additive white Gaussian noise
BIBO	Bounded input, bounded output
CDF	Cumulative distribution function
DCT	Discrete cosine transform
DFT	Discrete Fourier transform
DTFT	Discrete-time Fourier transform
DWT	Discrete wavelet transform
FFT	Fast Fourier transform
FIR	Finite impulse response
i.i.d.	Independent and identically distributed
IIR	Infinite impulse response
KLT	Karhunen–Loève transform
LOT	Lapped orthogonal transform
LPSV	Linear periodically shift varying
LSI	Linear shift invariant
MA	Moving average
MSE	Mean square error
PDF	Probability density function
POCS	Projection onto convex sets
ROC	Region of convergence
SVD	Singular value decomposition
WSCS	Wide-sense cyclostationary
WSS	Wide-sense stationary

## Abbreviations used in tables and captions but not in the text

FT	Fourier transform
FS	Fourier series
LFT	Local Fourier transform
WT	Wavelet transform

**Elements of Sets**

natural numbers	$\mathbb{N}$	$0, 1, \dots$
integers	$\mathbb{Z}$	$\dots, -1, 0, 1, \dots$
positive integers	$\mathbb{Z}^+$	$1, 2, \dots$
real numbers	$\mathbb{R}$	$(-\infty, \infty)$
positive real numbers	$\mathbb{R}^+$	$(0, \infty)$
complex numbers	$\mathbb{C}$	$a + jb$ or $re^{j\theta}$ with $a, b, r, \theta \in \mathbb{R}$
a generic index set	$\mathcal{I}$	
a generic vector space	$V$	
a generic Hilbert space	$H$	
real part of	$\Re(\cdot)$	
imaginary part of	$\Im(\cdot)$	
closure of set $S$	$\bar{S}$	
functions	$x(t)$	argument $t$ is continuous valued, $t \in \mathbb{R}$
sequences	$x_n$	argument $n$ is an integer, $n \in \mathbb{Z}$
ordered sequence	$(x_n)_n$	
set containing $x_n$	$\{x_n\}_n$	
vector $x$ with $x_n$ as elements	$[x_n]$	
Dirac delta function	$\delta(t)$	$\int_{-\infty}^{\infty} x(t)\delta(t) dt = x(0)$
Kronecker delta sequence	$\delta_n$	$\delta_n = 1$ for $n = 0$ ; $\delta_n = 0$ otherwise
indicator function of interval $I$	$1_I(t)$	$1_I(t) = 1$ for $t \in I$ ; $1_I(t) = 0$ otherwise

**Elements of Real Analysis**

integration by parts		$\int u dv = uv - \int v du$
----------------------	--	------------------------------

**Elements of Complex Analysis**

complex number	$z$	$a + jb, re^{j\theta}, a, b \in \mathbb{R}, r \in [0, \infty), \theta \in [0, 2\pi)$
conjugation	$z^*$	$a - jb, re^{-j\theta}$
conjugation of coefficients but not of $z$ itself	$X_*(z)$	$X^*(z^*)$
principal root of unity	$W_N$	$e^{-j2\pi/N}$

**Asymptotic Notation**

big O	$x \in O(y)$	$0 \leq x_n \leq \gamma y_n$ for all $n \geq n_0$ ; some $n_0$ and $\gamma > 0$
little o	$x \in o(y)$	$0 \leq x_n \leq \gamma y_n$ for all $n \geq n_0$ ; some $n_0$ , any $\gamma > 0$
Omega	$x \in \Omega(y)$	$x_n \geq \gamma y_n$ for all $n \geq n_0$ ; some $n_0$ and $\gamma > 0$
Theta	$x \in \Theta(y)$	$x \in O(y)$ and $x \in \Omega(y)$
asymptotic equivalence	$x \asymp y$	$\lim_{n \rightarrow \infty} x_n/y_n = 1$

**Standard Vector Spaces**

Hilbert space of square-summable sequences	$\ell^2(\mathbb{Z})$	$\left\{ x : \mathbb{Z} \rightarrow \mathbb{C} \mid \sum_n  x_n ^2 < \infty \right\}$ with inner product $\langle x, y \rangle = \sum_n x_n y_n^*$
Hilbert space of square-integrable functions	$\mathcal{L}^2(\mathbb{R})$	$\left\{ x : \mathbb{R} \rightarrow \mathbb{C} \mid \int  x(t) ^2 dt < \infty \right\}$ with inner product $\langle x, y \rangle = \int x(t)y(t)^* dt$
normed vector space of sequences with finite $p$ norm, $1 \leq p < \infty$	$\ell^p(\mathbb{Z})$	$\left\{ x : \mathbb{Z} \rightarrow \mathbb{C} \mid \sum_n  x_n ^p < \infty \right\}$ with norm $\ x\ _p = \left( \sum_n  x_n ^p \right)^{1/p}$
normed vector space of functions with finite $p$ norm, $1 \leq p < \infty$	$\mathcal{L}^p(\mathbb{R})$	$\left\{ x : \mathbb{R} \rightarrow \mathbb{C} \mid \int  x(t) ^p dt < \infty \right\}$ with norm $\ x\ _p = \left( \int  x(t) ^p dt \right)^{1/p}$
normed vector space of bounded sequences with supremum norm	$\ell^\infty(\mathbb{Z})$	$\left\{ x : \mathbb{Z} \rightarrow \mathbb{C} \mid \sup_n  x_n  < \infty \right\}$ with norm $\ x\ _\infty = \sup_n  x_n $
normed vector space of bounded functions with supremum norm	$\mathcal{L}^\infty(\mathbb{R})$	$\left\{ x : \mathbb{R} \rightarrow \mathbb{C} \mid \sup_t  x(t)  < \infty \right\}$ with norm $\ x\ _\infty = \sup_t  x(t) $

**Bases and Frames for Sequences**

standard Euclidean basis	$\{e_n\}$	$e_{n,k} = 1$ , for $k = n$ , and 0 otherwise
vector, element of basis or frame	$\varphi$	when applicable, a <i>column</i> vector
basis or frame	$\Phi$	set of vectors $\{\varphi_n\}$
operator	$\tilde{\Phi}$	concatenation of $\varphi_n$ s in a linear operator: $[\varphi_0 \ \varphi_1 \ \dots \ \varphi_{N-1}]$
vector, element of dual basis or frame	$\tilde{\varphi}$	when applicable, a <i>column</i> vector
operator	$\tilde{\tilde{\Phi}}$	set of vectors $\{\tilde{\varphi}_n\}$
operator	$\tilde{\Phi}$	concatenation of $\tilde{\varphi}_n$ s in a linear operator: $[\tilde{\varphi}_0 \ \tilde{\varphi}_1 \ \dots \ \tilde{\varphi}_{N-1}]$
expansion in a basis or frame	$x = \Phi \tilde{\Phi}^* x$	

**Transforms**

Fourier transform	$x(t) \xleftrightarrow{\text{FT}} X(\omega)$	$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt$ $x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{j\omega t} d\omega$
Fourier series	$x(t) \xleftrightarrow{\text{FS}} X_k$	$X_k = \frac{1}{T} \int_{-T/2}^{T/2} x(t)e^{-j(2\pi/T)kt} dt$ $x(t) = \sum_{k \in \mathbb{Z}} X_k e^{j(2\pi/T)kt}$
discrete-time Fourier transform	$x_n \xleftrightarrow{\text{DTFT}} X(e^{j\omega})$	$X(e^{j\omega}) = \sum_{n \in \mathbb{Z}} x_n e^{-j\omega n}$ $x_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega}) e^{j\omega n} d\omega$
discrete Fourier transform	$x_n \xleftrightarrow{\text{DFT}} X_k$	$X_k = \sum_{n=0}^{N-1} x_n W_N^{kn}$ $x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k W_N^{-kn}$
local Fourier transform	$x(t) \xleftrightarrow{\text{LFT}} X(\Omega, \tau)$	$X(\Omega, \tau) = \int_{-\infty}^{\infty} x(t)p(t-\tau)e^{-j\Omega t} dt$ $x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} X(\Omega, \tau) g_{\Omega, \tau}(t) d\Omega d\tau$
continuous wavelet transform	$x(t) \xleftrightarrow{\text{CWT}} X(a, b)$	$X(a, b) = \int_{-\infty}^{\infty} x(t) \psi_{a,b}(t) dt$ $x(t) = \frac{1}{C_\psi} \int_0^{\infty} \int_{-\infty}^{\infty} X(a, b) \psi_{a,b}(t) \frac{db da}{a^2}$
wavelet series	$x(t) \xleftrightarrow{\text{WS}} \beta_k^{(\ell)}$	$\beta_k^{(\ell)} = \int_{-\infty}^{\infty} x(t) \psi_{\ell,k}(t) dt$ $x(t) = \sum_{\ell \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} \beta_k^{(\ell)} \psi_{\ell,k}(t)$
discrete wavelet transform	$x_n \xleftrightarrow{\text{DWT}} \alpha_k^{(J)}, \beta_k^{(J)}, \dots, \beta_k^{(1)}$	$\alpha_k^{(J)} = \sum_{n \in \mathbb{Z}} x_n g_{n-2^J k}^{(J)}, \beta_k^{(\ell)} = \sum_{n \in \mathbb{Z}} x_n h_{n-2^\ell k}^{(\ell)}$ $x_n = \sum_{k \in \mathbb{Z}} \alpha_k^{(J)} g_{n-2^J k}^{(J)} + \sum_{\ell=1}^J \sum_{k \in \mathbb{Z}} \beta_k^{(\ell)} h_{n-2^\ell k}^{(\ell)}$
discrete cosine transform	$x_n \xleftrightarrow{\text{DCT}} X_k$	$X_0 = \sqrt{\frac{1}{N}} \sum_{n=0}^{N-1} x_n$ $X_k = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x_n \cos((2\pi/2N)k(n+1/2))$ $x_0 = \sqrt{\frac{1}{N}} \sum_{k=0}^{N-1} X_k$ $x_n = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} X_k \cos((2\pi/2N)k(n+1/2))$
z-transform	$x_n \xleftrightarrow{\text{ZT}} X(z)$	$X(z) = \sum_{n \in \mathbb{Z}} x_n z^{-n}$

**Discrete-Time Nomenclature**

<i>Sequence</i>	$x_n$	signal, vector
<i>Convolution</i>		
linear	$h * x$	$\sum_{k \in \mathbb{Z}} x_k h_{n-k} = \sum_{k \in \mathbb{Z}} h_k x_{n-k}$
circular	$h \circledast x$	$\sum_{k=0}^{N-1} x_k h_{(n-k) \bmod N} = \sum_{k=0}^{N-1} h_k x_{(n-k) \bmod N}$
	$(h * x)_n$	$n$ th element of the convolution result
	$h_{\ell-n} * x_{n-m}$	$\sum_{k \in \mathbb{Z}} x_{k-m} h_{\ell-n+k}$
<i>Eigensequence</i>		
	$v_n$	eigenfunction, eigenvector
infinite time	$v_n = e^{j\omega n}$	$h * v = H(e^{j\omega}) v$
finite time	$v_n = e^{j2\pi kn/N}$	$h \circledast v = H_k v$
<i>Frequency response</i>		
infinite time	$H(e^{j\omega})$	eigenvalue corresponding to $v_n$ $\sum_{n \in \mathbb{Z}} h_n e^{-j\omega n}$
finite time	$H_k$	$\sum_{n=0}^{N-1} h_n e^{-j2\pi kn/N} = \sum_{n=0}^{N-1} h_n W_N^{kn}$

**Continuous-Time Nomenclature**

<i>Function</i>	$x(t)$	signal
<i>Convolution</i>		
linear	$h * x$	$\int_{-\infty}^{\infty} x(\tau) h(t-\tau) d\tau = \int_{-\infty}^{\infty} h(\tau) x(t-\tau) d\tau$
circular	$h \circledast x$	$\int_0^T x(\tau) h(t-\tau) d\tau = \int_0^T h(\tau) x(t-\tau) d\tau$
	$(h * x)(t)$	convolution result at $t$
<i>Eigenfunction</i>		
	$v(t)$	eigenvector
infinite time	$v(t) = e^{j\omega t}$	$h * v = H(\omega) v$
finite time	$v(t) = e^{j2\pi kt/T}$	$h \circledast v = H_k v$
<i>Frequency response</i>		
infinite time	$H(\omega)$	eigenvalue corresponding to $v(t)$ $\int_{-\infty}^{\infty} h(t) e^{-j\omega t} dt$
finite time	$H_k$	$\int_{-T/2}^{T/2} h(\tau) e^{-j2\pi k\tau/T} d\tau$

**Two-Channel Filter Banks***Basic characteristics*

number of channels	$M = 2$
sampling factor	$N = 2$
channel sequences	$\alpha_n \quad \beta_n$

*Filters*

	<i>Synthesis</i>		<i>Analysis</i>	
	lowpass	highpass	lowpass	highpass
orthogonal	$g_n$	$h_n$	$g_{-n}$	$h_{-n}$
biorthogonal	$g_n$	$h_n$	$\tilde{g}_n$	$\tilde{h}_n$
polyphase components	$g_{0,n}, g_{1,n}$	$h_{0,n}, h_{1,n}$	$\tilde{g}_{0,n}, \tilde{g}_{1,n}$	$\tilde{h}_{0,n}, \tilde{h}_{1,n}$

**Tree-Structured Filter Banks (DWT)***Basic characteristics*

number of channels	$M = J + 1$
sampling at level $\ell$	$N^{(\ell)} = 2^\ell$
channel sequences	$\alpha_n^{(J)} \quad \beta_n^{(\ell)} \quad \ell = 1, 2, \dots, J$

*Filters*

	<i>Synthesis</i>		<i>Analysis</i>		
	lowpass	bandpass <sup>(<math>\ell</math>)</sup>	lowpass	bandpass <sup>(<math>\ell</math>)</sup>	
orthogonal	$g_n^{(J)}$	$h_n^{(\ell)}$	$g_{-n}^{(J)}$	$h_{-n}^{(\ell)}$	
biorthogonal	$g_n^{(J)}$	$h_n^{(\ell)}$	$\tilde{g}_n^{(J)}$	$\tilde{h}_n^{(\ell)}$	
polyphase component $j$	$g_{j,n}^{(J)}$	$h_{j,n}^{(\ell)}$	$\tilde{g}_{j,n}^{(J)}$	$\tilde{h}_{j,n}^{(\ell)}$	$j = 0, 1, \dots, 2^\ell - 1$

 **$N$ -Channel Filter Banks***Basic characteristics*

number of channels	$M = N$
sampling factor	$N$
channel sequences	$\alpha_{i,n} \quad i = 0, 1, \dots, N - 1$

*Filters*

	<i>Synthesis</i>	<i>Analysis</i>	
orthogonal filter $i$	$g_{i,n}$	$g_{i,-n}$	
biorthogonal filter $i$	$g_{i,n}$	$\tilde{g}_{i,n}$	
polyphase component $j$	$g_{i,j,n}$	$\tilde{g}_{i,j,n}$	$j = 0, 1, \dots, N - 1$

**Oversampled Filter Banks***Basic characteristics*

number of channels	$M > N$
sampling factor	$N$
channel sequences	$\alpha_{i,n} \quad i = 0, 1, \dots, M - 1$

*Filters*

	<i>Synthesis</i>	<i>Analysis</i>	
filter $i$	$g_{i,n}$	$\tilde{g}_{i,n}$	
polyphase component $j$	$g_{i,j,n}$	$\tilde{g}_{i,j,n}$	$j = 0, 1, \dots, N - 1$

# Preface

The aim of this book, together with its predecessor *Signal Processing: Foundations (SP:F)* [107], is to provide a set of tools for users of state-of-the-art signal processing technology and a solid foundation for those hoping to advance the theory and practice of signal processing. Many of the results and techniques presented here, while rooted in classic Fourier techniques for signal representation, first appeared during a flurry of activity in the 1980s and 1990s. New constructions for local Fourier transforms and orthonormal wavelet bases during that period were motivated both by theoretical interest and by applications, in particular in multimedia communications. New bases with specified time–frequency behavior were found, with impact well beyond the original fields of application. Areas as diverse as computer graphics and numerical analysis embraced some of the new constructions—no surprise given the pervasive role of Fourier analysis in science and engineering.

Now that the dust has settled, some of what was new and esoteric is now fundamental. Our motivation is to bring these new fundamentals to a broader audience to further expand their impact. We thus provide an integrated view of classical Fourier analysis of signals and systems alongside structured representations with time–frequency locality and their myriad of applications.

This book relies heavily on the base built in *SP:F*. Thus, these two books are to be seen as integrally related to each other. References to *SP:F* are given in italics.

**Signal Processing: Foundations** The first book covers the foundations for an extensive understanding of signal processing. It contains material that many readers may have seen before, but without the Hilbert space interpretations that are essential in contemporary signal processing research and technology. In Chapter 2, *From Euclid to Hilbert*, the basic geometric intuition central to Hilbert spaces is developed, together with all the necessary tools underlying the construction of bases. Chapter 3, *Sequences and Discrete-Time Systems*, is a crash course on processing signals in discrete time or discrete space. In Chapter 4, *Functions and Continuous-Time Systems*, the mathematics of Fourier transforms and Fourier series is reviewed. Chapter 5, *Sampling and Interpolation*, talks about the critical link between discrete and continuous domains as given by the sampling theorem and interpolation, while Chapter 6, *Approximation and Compression*, veers from the exact world to the approximate one. The final chapter is Chapter 7, *Localization and Uncertainty*, and

it considers time–frequency behavior of the abstract representation objects studied thus far. It also discusses issues arising in the real world as well as ways of adapting these tools for use in the real world. The main concepts seen—such as geometry of Hilbert spaces, existence of bases, Fourier representations, sampling and interpolation, and approximation and compression—build a powerful foundation for modern signal processing. These tools hit roadblocks they must overcome: finiteness and localization, limitations of uncertainty, and computational costs.

**Signal Processing: Fourier and Wavelet Representations** This book presents signal representations, including Fourier, local Fourier and wavelet bases, related constructions, as well as frames and continuous transforms.

It starts with Chapter 1, *Filter Banks: Building Blocks of Time-Frequency Expansions*, which presents a thorough treatment of the basic block—the two-channel filter bank, a signal processing device that splits a signal into a coarse, lowpass approximation, and a highpass detail.

We generalize this block in the three chapters that follow, all dealing with Fourier- and wavelet-like representations on sequences: In Chapter 2, *Local Fourier Bases on Sequences*, we discuss Fourier-like bases on sequences, implemented by  $N$ -channel modulated filter banks (first generalization of the two-channel filter banks). In Chapter 3, *Wavelet Bases on Sequences*, we discuss wavelet-like bases on sequences, implemented by tree-structured filter banks (second generalization). In Chapter 4, *Local Fourier and Wavelet Frames on Sequences*, we discuss both Fourier- and wavelet-like frames on sequences, implemented by oversampled filter banks (third generalization).

We then move to the two chapters dealing with Fourier- and wavelet-like representations on functions. In Chapter 5, *Local Fourier Transforms, Frames and Bases on Functions*, we start with the most natural representation of smooth functions with some locality, the local Fourier transform, followed by its sampled version/frame, and leading to results on whether bases are possible. In Chapter 6, *Wavelet Bases, Frames and Transforms on Functions*, we do the same for wavelet representations on functions, but in opposite order: starting from bases, through frames and finally continuous wavelet transform.

The last chapter, Chapter 7, *Approximation, Estimation, and Compression*, uses all the tools we introduced to address state-of-the-art signal processing and communication problems and their solutions. The guiding principle is that there is a domain where the problem at hand will have a sparse solution, at least approximately so. This is known as sparse signal processing, and many examples, from the classical Karhunen-Loève expansion to nonlinear approximation in discrete cosine transform and wavelet domains, all the way to contemporary research in compressed sensing, use this principle. The chapter introduces and overviews sparse signal processing, covering approximation methods, estimation procedures such as denoising, as well as compression methods and inverse problems.

**Teaching Points** Our aim is to present a synthetic view from basic mathematical principles to actual constructions of bases and frames, always with an eye on

---

concrete applications. While the benefit is a self-contained presentation, the cost is a rather sizable manuscript. Referencing in the main text is sparse; pointers to bibliography are given in *Further Reading* at the end of each chapter.

The material grew out of teaching signal processing, wavelets and applications in various settings. Two of the authors, Martin Vetterli and Jelena Kovačević, authored a graduate textbook, *Wavelets and Subband Coding* (originally with Prentice Hall in 1995), which they and others used to teach graduate courses at various US and European institutions. This book is now online with open access.<sup>1</sup> With more than a decade of experience, the maturing of the field, and the broader interest arising from and for these topics, the time was right for an entirely new text geared towards a broader audience, one that could be used to span levels from undergraduate to graduate, as well as various areas of engineering and science. As a case in point, parts of the text have been used at Carnegie Mellon University in classes on bioimage informatics, where some of the students are life-sciences majors. This plasticity of the text is one of the features which we aimed for, and that most probably differentiates the present book from many others. Another aim is to present side-by-side all methods that arose around signal representations, without favoring any in particular. The truth is that each representation is a tool in the toolbox of the practitioner, and the problem or application at hand ultimately determines the appropriate one to use.

**Free Version** This free, electronic version of the book contains all of the main material, except for solved exercises and exercises. Consequently, all references to exercises will show as ???. Moreover, this version does not contain PDF hyperlinks.

Jelena Kovačević, Vivek K Goyal and Martin Vetterli  
March 2012

---

<sup>1</sup><http://waveletsandsubbandcoding.org/>



# Acknowledgments

This project exists thanks to the help of many people, whom we attempt to list below. We apologize for any omissions. The current edition is a work in progress; we welcome corrections, complaints, and suggestions.

We are grateful to Prof. Libero Zuppiroli of EPFL and Christiane Grimm for the photograph that graces the cover; Prof. Zuppiroli proposed an experiment from Newton's treatise on Opticks [65] as emblematic of the book, and Ms. Grimm beautifully photographed the apparatus that he designed. Françoise Behn, Jocelyne Plantefol and Jacqueline Aeberhard typed and organized parts of the manuscript, Eric Strattman assisted with many of the figures, Krista Van Guilder designed and implemented the book web site, and Jorge Albaladejo Pomares designed and implemented the book blog. We thank them for their diligence and patience. Patrick Vandewalle designed, wrote and implemented most of the Matlab companion to the book. Similarly, S. Grace Chang and Yann Barbotin helped organize and edit the problem companion. We thank them for their expertise and insight. We are indebted to George Beck and Ed Pegg, Jr., for their Mathematica editorial comments on Wolfram Demonstrations inspired by this book.

Many instructors have gamely tested pre-alpha versions of this manuscript. Of these, Amina Chebira, Yue M. Lu, and Thao Nguyen have done far more than their share in providing invaluable comments and suggestions. We also thank Zoran Cvetković and Minh Do for teaching with the manuscript and providing many constructive comments, Matthew Fickus for consulting on some finer mathematical points, and Thierry Blu for providing a simple proof for a particular case of the Strang–Fix theorem. Useful comments have also been provided by Pedro Aguilar, A. Avudainayagam, Aniruddha Bhargava, André Tomaz de Carvalho, S. Esakkirajan, Germán González, Alexandre Haehlen, Mina Karzand, Hossein Rouhani, Noah Stein, and Christophe Tournery.

Martin Vetterli thanks current and former EPFL graduate students and post-docs who helped develop the material, solve problems, catch typos, and suggest improvements, among other things. They include Florence Bénézit, Amina Chebira, Minh Do, Ivan Dokmanic, Pier Luigi Dragotti, Ali Hormati, Ivana Jovanović, Mihailo Kolundzija, Jérôme Lebrun, Yue Lu, Pina Marziliano, Fritz Menzer, Reza Parhizkar, Paolo Prandoni, Juri Ranieri, Olivier Roy, Rahul Shukla, Jayakrishnan Unnikrishnan, Patrick Vandewalle and Vladan Velisavljević. He gratefully acknowledges support from the Swiss National Science Foundation through awards 2000-063664, 200020-103729, 200021-121935, and the European Research Council

through award SPARSAM 247006.

Jelena Kovačević thanks her present and past graduate students Ramu Bhagavatula, Amina Chebira, Jackie Chen, Siheng Chen, Charles Jackson, Xindian Long, Mike McCann, Anupama Kuruvilla, Tad Merryman, Vivek Oak, Aliaksei Sandryhaila and Gowri Srinivasa, many of whom served as TAs for her classes at CMU, together with Pablo Hennings Yeomans. Thanks also to all the students in the following classes at CMU: 42-431/18-496, 42-540, 42-731/18-795, 18-799, 42-540, taught from 2003 to 2012. She is grateful to her husband, Giovanni Pacifici, for the many useful scripts automating various book-writing processes. She gratefully acknowledges support from the NSF through awards 1017278, 1130616, 515152, 633775, 0331657, 627250 and 331657; the NIH through awards 1DC010283, EB008870, EB009875 and 1DC010283; the CMU CIT through Infrastructure for Large-Scale Biomedical Computing award; the PA State Tobacco Settlement, Kamlet-Smith Bioinformatics Grant; and the Philip and Marsha Dowd Teaching Fellowship.

Vivek Goyal thanks TAs and students at MIT for suggestions, in particular Baris Erkmen, Ying-zong Huang, Zahi Karam, Ahmed Kirmani, Ranko Sredojevic, Ramesh Sridharan, Watcharapan Suwansantisuk, Vincent Tan, Archana Venkataraman, Adam Zelinski, and Serhii Zhak. He gratefully acknowledges support from the NSF through awards 0643836, 0729069, 1101147, 1115159, and 1161413; Texas Instruments through its Leadership University Program; Hewlett-Packard, Inc.; MIT through an Esther and Harold E. Edgerton Career Development Chair; and the Centre Bernoulli of EPFL.

## Chapter 1

# Filter Banks: Building Blocks of Time-Frequency Expansions

### Contents

1.1	Introduction	2
1.2	Orthogonal Two-Channel Filter Banks	7
1.3	Design of Orthogonal Two-Channel Filter Banks	20
1.4	Biorthogonal Two-Channel Filter Banks	26
1.5	Design of Biorthogonal Two-Channel Filter Banks	37
1.6	Two-Channel Filter Banks with Stochastic Inputs	41
1.7	Computational Aspects	42
	Chapter at a Glance	49
	Historical Remarks	52
	Further Reading	52

The aim of this chapter is to build discrete-time bases with desirable time-frequency features and structure that enable tractable analysis and efficient algorithmic implementation. We achieve these goals by constructing bases via filter banks.

Using filter banks provides an easy way to understand the relationship between analysis and synthesis operators, while, at the same time, making their efficient implementation obvious. Moreover, filter banks are at the root of the constructions of wavelet bases in Chapters 3 and 6. In short, together with discrete-time filters and the FFT, filter banks are among the most basic tools of signal processing.

This chapter deals exclusively with two-channel filter banks since they are (1) the simplest; (2) reveal the essence of the  $N$ -channel ones; and (3) are used as building blocks for more general bases. We focus first on the orthogonal case, which is the most structured and has the easiest geometric interpretation. Due to its importance in practice, we follow with the discussion of the biorthogonal case. We consider real-coefficient filter banks exclusively; pointers to complex-coefficient ones, as well as to various generalizations, such as  $N$ -channel filter banks, multidimensional filter banks and transmultiplexers, are given in *Further Reading*.

## 1.1 Introduction

### Implementing a Haar Orthonormal Basis Expansion

At the end of the previous book, we constructed an orthonormal basis for  $\ell^2(\mathbb{Z})$  which possesses structure in terms of time and frequency localization properties (it serves as an almost perfect localization tool in time, and a rather rough one in frequency); and, is efficient (it is built from two template sequences, one lowpass and the other highpass, and their shifts). This was the so-called Haar basis.

What we want to do now is implement that basis using signal processing machinery. We first rename our template basis sequences from (??) and (??) as:

$$g_n = \varphi_{0,n} = \frac{1}{\sqrt{2}}(\delta_n + \delta_{n-1}), \quad (1.1a)$$

$$h_n = \varphi_{1,n} = \frac{1}{\sqrt{2}}(\delta_n - \delta_{n-1}). \quad (1.1b)$$

This is done both for simplicity, as well as because it is the standard way these sequences are denoted. We start by rewriting the reconstruction formula (??) as

$$\begin{aligned} x_n &= \sum_{k \in \mathbb{Z}} \underbrace{\langle x, \varphi_{2k} \rangle}_{\alpha_k} \varphi_{2k,n} + \sum_{k \in \mathbb{Z}} \underbrace{\langle x, \varphi_{2k+1} \rangle}_{\beta_k} \varphi_{2k+1,n} \\ &= \sum_{k \in \mathbb{Z}} \alpha_k \underbrace{\varphi_{2k,n}}_{g_{n-2k}} + \sum_{k \in \mathbb{Z}} \beta_k \underbrace{\varphi_{2k+1,n}}_{h_{n-2k}} \\ &= \sum_{k \in \mathbb{Z}} \alpha_k g_{n-2k} + \sum_{k \in \mathbb{Z}} \beta_k h_{n-2k}, \end{aligned} \quad (1.2)$$

where we have renamed the basis functions as in (1.1), as well as denoted the expansion coefficients as

$$\langle x, \varphi_{2k} \rangle = \langle x_n, g_{n-2k} \rangle_n = \alpha_k, \quad (1.3a)$$

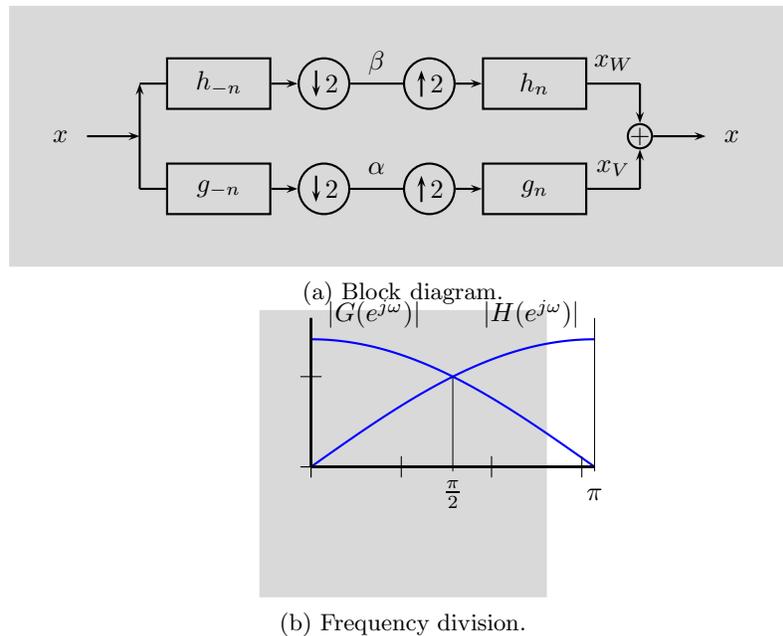
$$\langle x, \varphi_{2k+1} \rangle = \langle x_n, h_{n-2k} \rangle_n = \beta_k. \quad (1.3b)$$

Then, recognize each sum in (1.2) as the output of upsampling followed by filtering (3.203) with the input sequences being  $\alpha_k$  and  $\beta_k$ , respectively. Thus, the first sum in (1.2) can be implemented as the input sequence  $\alpha$  going through an upsampler by 2 followed by filtering by  $g$ , and the second as the input sequence  $\beta$  going through an upsampler by 2 followed by filtering by  $h$ .

By the same token, we can identify the computation of the expansion coefficients in (1.3) as (3.200), that is, both  $\alpha$  and  $\beta$  sequences can be obtained using filtering by  $g_{-n}$  followed by downsampling by 2 (for  $\alpha_k$ ), or filtering by  $h_{-n}$  followed by downsampling by 2 (for  $\beta_k$ ).

We can put together the above operations to yield a *two-channel filter bank* implementing a Haar orthonormal basis expansion as in Figure 1.1(a). The left part that computes the expansion coefficients is termed an *analysis filter bank*, while the right part that computes the projections is termed a *synthesis filter bank*.

As before, once we have identified all the appropriate multirate components, we can examine the Haar filter bank via matrix operations (linear operators). For



**Figure 1.1:** A two-channel analysis/synthesis filter bank. (a) Block diagram, where an analysis filter bank is followed by a synthesis filter bank. In the orthogonal case, the impulse responses of the analysis filters are time-reversed versions of the impulse responses of the synthesis filters. The filter  $g$  is typically lowpass, while the filter  $h$  is typically highpass. (b) Frequency responses of the two Haar filters computing averages and differences, showing the decomposition into low- and high-frequency content.

example, in matrix notation, the analysis process (1.3) can be expressed as

$$\begin{bmatrix} \vdots \\ \alpha_0 \\ \beta_0 \\ \alpha_1 \\ \beta_1 \\ \alpha_2 \\ \beta_2 \\ \vdots \end{bmatrix} = \frac{1}{\sqrt{2}} \underbrace{\begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ \dots & \boxed{1} & 1 & 0 & 0 & 0 & 0 & \dots \\ \dots & 1 & -1 & 0 & 0 & 0 & 0 & \dots \\ \dots & 0 & 0 & 1 & 1 & 0 & 0 & \dots \\ \dots & 0 & 0 & 1 & -1 & 0 & 0 & \dots \\ \dots & 0 & 0 & 0 & 0 & 1 & 1 & \dots \\ \dots & 0 & 0 & 0 & 0 & 1 & -1 & \dots \\ \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}}_{\Phi^T} \begin{bmatrix} \vdots \\ \boxed{x_0} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ \vdots \end{bmatrix}, \tag{1.4}$$

and the synthesis process (1.2) as

$$\underbrace{\begin{bmatrix} \vdots \\ \boxed{x_0} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ \vdots \end{bmatrix}}_x = \frac{1}{\sqrt{2}} \underbrace{\begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ \dots & \boxed{1} & 1 & 0 & 0 & 0 & 0 & \dots \\ \dots & 1 & -1 & 0 & 0 & 0 & 0 & \dots \\ \dots & 0 & 0 & 1 & 1 & 0 & 0 & \dots \\ \dots & 0 & 0 & 1 & -1 & 0 & 0 & \dots \\ \dots & 0 & 0 & 0 & 0 & 1 & 1 & \dots \\ \dots & 0 & 0 & 0 & 0 & 1 & -1 & \dots \\ \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} \vdots \\ \boxed{\alpha_0} \\ \beta_0 \\ \alpha_1 \\ \beta_1 \\ \alpha_2 \\ \beta_2 \\ \vdots \end{bmatrix}}_X, \quad (1.5)$$

or

$$x = \Phi\Phi^T x \quad \Rightarrow \quad \Phi\Phi^T = I. \quad (1.6)$$

Of course, the matrix  $\Phi$  is the same matrix we have seen in (??). Moreover, from (1.6), it is a unitary matrix, which we know from *Chapter 2, Chapter at a Glance*, implies that the Haar basis is an orthonormal basis (and have already shown in *Chapter 7*). Table 1.8 gives a summary of the Haar filter bank in various domains.

### Implementing a General Orthonormal Basis Expansion

What we have seen for the Haar orthonormal basis is true in general; we can construct an orthonormal basis for  $\ell^2(\mathbb{Z})$  using two template basis sequences and their even shifts. As we have seen, we can implement such an orthonormal basis using a two-channel filter bank, consisting of downsamplers, upsamplers and filters  $g$  and  $h$ . Let  $g$  and  $h$  be two real-coefficient, causal filters,<sup>2</sup> where we implicitly assume that these filters have certain time and frequency localization properties, as discussed in *Chapter 7* ( $g$  is lowpass and  $h$  is highpass). The synthesis (1.5) generalizes to

$$\begin{bmatrix} \vdots \\ \boxed{x_0} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \end{bmatrix} = \begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ \dots & \boxed{g_0} & h_0 & 0 & 0 & 0 & 0 & \dots \\ \dots & g_1 & h_1 & 0 & 0 & 0 & 0 & \dots \\ \dots & g_2 & h_2 & g_0 & h_0 & 0 & 0 & \dots \\ \dots & g_3 & h_3 & g_1 & h_1 & 0 & 0 & \dots \\ \dots & g_4 & h_4 & g_2 & h_2 & g_0 & h_0 & \dots \\ \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ \boxed{\alpha_0} \\ \beta_0 \\ \alpha_1 \\ \beta_1 \\ \alpha_2 \\ \vdots \end{bmatrix} = \Phi X, \quad (1.7)$$

with the basis matrix  $\Phi$  as before. To have an orthonormal basis, the basis sequences  $\{\varphi_k\}_{k \in \mathbb{Z}}$ —even shifts of template sequences  $g$  and  $h$ , must form an orthonormal set as in (2.89), or,  $\Phi$  must be unitary, implying its columns are orthonormal:

$$\langle g_n, g_{n-2k} \rangle_n = \delta_k, \quad \langle h_n, h_{n-2k} \rangle_n = \delta_k, \quad \langle g_n, h_{n-2k} \rangle_n = 0. \quad (1.8)$$

<sup>2</sup>While causality is not necessary to construct a filter bank, we impose it later and it improves readability here. We stress again that we deal exclusively with real-coefficient filters.

We have seen in (3.215) that such filters are called orthogonal; how to design them is a central topic of this chapter.

As we are building an orthonormal basis, computing the expansion coefficients of an input sequence means taking the inner product between that sequence and each basis sequence. In terms of the orthonormal set given by the columns of  $\Phi$ , this amounts to a multiplication by  $\Phi^T$ :

$$\underbrace{\begin{bmatrix} \vdots \\ \alpha_0 \\ \beta_0 \\ \alpha_1 \\ \beta_1 \\ \alpha_2 \\ \vdots \end{bmatrix}}_X = \underbrace{\begin{bmatrix} \vdots \\ \langle x_n, g_n \rangle_n \\ \langle x_n, h_n \rangle_n \\ \langle x_n, g_{n-2} \rangle_n \\ \langle x_n, h_{n-2} \rangle_n \\ \langle x_n, g_{n-4} \rangle_n \\ \vdots \end{bmatrix}}_X = \underbrace{\begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ \cdots & g_0 & g_1 & g_2 & g_3 & g_4 & \cdots \\ \cdots & h_0 & h_1 & h_2 & h_3 & h_4 & \cdots \\ \cdots & 0 & 0 & g_0 & g_1 & g_2 & \cdots \\ \cdots & 0 & 0 & h_0 & h_1 & h_2 & \cdots \\ \cdots & 0 & 0 & 0 & 0 & g_0 & \cdots \\ \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}}_{\Phi^T} \underbrace{\begin{bmatrix} \vdots \\ x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \end{bmatrix}}_x \quad (1.9)$$

As in the Haar case, this can be implemented with convolutions by  $g_{-n}$  and  $h_{-n}$ , followed by downsampling by 2—an analysis filter bank as in Figure 1.1(a). In filter bank terms, the representation of  $x$  in terms of a basis (or frame) is called *perfect reconstruction*.

Thus, what we have built is as in Chapter 7—an orthonormal basis with structure (time and frequency localization properties) as well as efficient implementation guaranteed by the filter bank. As in the Haar case, this structure is seen in the subspaces  $V$  and  $W$  on which the orthonormal basis projects; we implicitly assume that  $V$  is the space of coarse (lowpass) sequences and  $W$  is the space of detail (highpass) sequences. Figure 1.3 illustrates that, where a synthetic sequence with features at different scales is split into lowpass and highpass components. These subspaces are spanned by the lowpass template  $g$  and its even shifts ( $V$ ) and the highpass template  $h$  and its even shifts ( $W$ ) as in (??):

$$V = \overline{\text{span}}(\{\varphi_{0,n-2k}\}_{k \in \mathbb{Z}}) = \overline{\text{span}}(\{g_{n-2k}\}_{k \in \mathbb{Z}}), \quad (1.10a)$$

$$W = \overline{\text{span}}(\{\varphi_{1,n-2k}\}_{k \in \mathbb{Z}}) = \overline{\text{span}}(\{h_{n-2k}\}_{k \in \mathbb{Z}}), \quad (1.10b)$$

and produce the lowpass and highpass approximations, respectively:

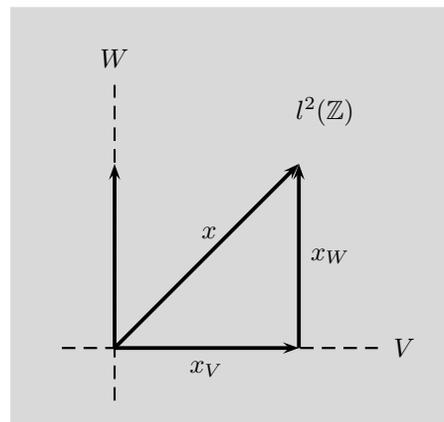
$$x_V = \sum_{k \in \mathbb{Z}} \alpha_k g_{n-2k}, \quad (1.11a)$$

$$x_W = \sum_{k \in \mathbb{Z}} \beta_k h_{n-2k}. \quad (1.11b)$$

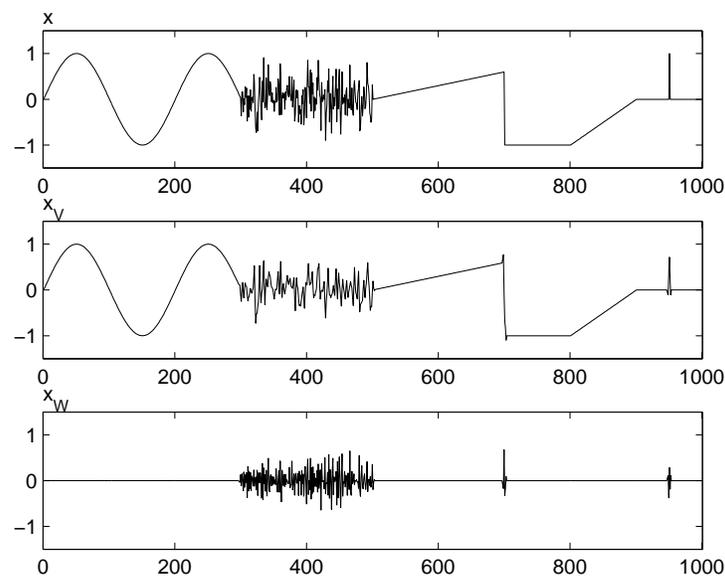
As the basis sequences spanning these spaces are orthogonal to each other and all together form an orthonormal basis, the two projection subspaces together give back the original space as in (??):  $\ell^2(\mathbb{Z}) = V \oplus W$ .

In this brief chapter preview, we introduced the two-channel filter bank as in Figure 1.1(a). It uses orthogonal filters satisfying (1.8) and computes an expansion

## 6 Chapter 1. Filter Banks: Building Blocks of Time-Frequency Expansions



**Figure 1.2:** A sequence  $x$  is split into two approximation sequences  $x_V$  and  $x_W$ . An orthonormal filter bank ensures that  $x_V$  and  $x_W$  are orthogonal and sum up to the original sequence. We also show the split of  $l^2(\mathbb{Z})$  into two orthogonal complements  $V$  (lowpass subspace) and  $W$  (highpass subspace).



**Figure 1.3:** A sequence and its projections. (a) The sequence  $x$  with different-scale features (low-frequency sinusoid, high-frequency noise, piecewise polynomial and a Kronecker delta sequence). (b) The lowpass projection  $x_V$ . (c) The highpass projection  $x_W$ .

with respect to the set of basis vectors  $\{g_{n-2k}, h_{n-2k}\}_{k \in \mathbb{Z}}$ , yielding a decomposition into approximation spaces  $V$  and  $W$  having complementary signal processing properties. Our task now is to find appropriate filters (template basis sequences) and develop properties of the filter bank in detail. We start by considering the lowpass filter  $g$ , since everything else will follow from there. We concentrate only on real-coefficient FIR filters since they are dominant in practice.

### Chapter Outline

We start by showing how orthonormal bases are implemented by orthogonal filter banks in Section 1.2 and follow by discussing three approaches to the design of orthogonal filter banks in Section 1.3. We then discuss the theory and design of biorthogonal filter banks in Sections 1.4 and 1.5. In Section 1.6, we discuss stochastic filter banks, followed by algorithms in Section 1.7.

*Notation used in this chapter:* In this chapter, we consider real-coefficient filter banks exclusively; pointers to complex-coefficient ones are given in *Further Reading*. Thus, Hermitian transposition will occur rarely; when filter coefficients are complex, the transposition in some places should be Hermitian transposition, however, only coefficients should be conjugated and not  $z$ . We will point these out throughout the chapter.  $\square$

## 1.2 Orthogonal Two-Channel Filter Banks

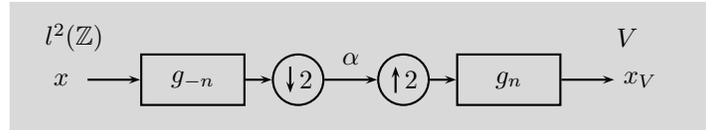
This section develops necessary conditions for the design of orthogonal two-channel filter banks implementing orthonormal bases and the key properties of such filter banks. We assume that the system shown in Figure 1.1(a) implements an orthonormal basis for sequences in  $\ell^2(\mathbb{Z})$  using the basis sequences  $\{g_{n-2k}, h_{n-2k}\}_{k \in \mathbb{Z}}$ . We first determine what this means for the lowpass and highpass channels separately, and follow by combining the channels. We then develop a polyphase representation for orthogonal filter banks and discuss their polynomial approximation properties.

### 1.2.1 A Single Channel and Its Properties

We now look at each channel of Figure 1.1 separately and determine their properties. As the lowpass and highpass channels are essentially symmetric, our approach is to establish (1) the properties inherent to each channel on its own; and (2) given one channel, establish the properties the other has to satisfy so as to build an orthonormal basis when combined. While we have seen most of the properties already, we summarize them here for completeness.

Consider the lower branch of Figure 1.1(a), projecting the input  $x$  onto its lowpass approximation  $x_V$ , depicted separately in Figure 1.4. In (1.11a), that lowpass approximation  $x_V$  was given as

$$x_V = \sum_{k \in \mathbb{Z}} \alpha_k g_{n-2k}. \quad (1.12a)$$



**Figure 1.4:** The lowpass branch of a two-channel filter bank, mapping  $x$  to  $x_V$ .

Similarly, in (1.11b), the highpass approximation  $x_W$  was given as

$$x_W = \sum_{k \in \mathbb{Z}} \beta_k h_{n-2k}. \quad (1.12b)$$

**Orthogonality of the Lowpass Filter** Since we started with an orthonormal basis, the set  $\{g_{n-2k}\}_{k \in \mathbb{Z}}$  is an orthonormal set. We have seen in *Section 3.7.4* that such a filter is termed orthogonal and satisfies (3.215):

$$\langle g_n, g_{n-2k} \rangle = \delta_k \quad \begin{array}{l} \xleftrightarrow{\text{Matrix View}} \\ \xleftrightarrow{\text{ZT}} \\ \xleftrightarrow{\text{DTFT}} \end{array} \quad \begin{array}{l} D_2 G^T G U_2 = I \\ G(z)G(z^{-1}) + G(-z)G(-z^{-1}) = 2 \\ |G(e^{j\omega})|^2 + |G(e^{j(\omega+\pi)})|^2 = 2 \end{array} \quad (1.13)$$

In the matrix view, we have used linear operators (infinite matrices) introduced in *Section 3.7*. These are: (1) downsampling by 2,  $D_2$ , from (3.183a); (2) upsampling by 2,  $U_2$ , from (3.189a) and (3) filtering by  $G$ , from (3.62). The matrix view expresses the fact that the columns of  $GU_2$  form an orthonormal set.<sup>3</sup> The DTFT version is the quadrature mirror formula from (3.214).

**Orthogonality of the Highpass Filter** Similarly to  $\{g_{n-2k}\}_{k \in \mathbb{Z}}$ , the set  $\{h_{n-2k}\}_{k \in \mathbb{Z}}$  is an orthonormal set, and the sequence  $h$  can be seen as the impulse response of an orthogonal filter satisfying:

$$\langle h_n, h_{n-2k} \rangle = \delta_k \quad \begin{array}{l} \xleftrightarrow{\text{Matrix View}} \\ \xleftrightarrow{\text{ZT}} \\ \xleftrightarrow{\text{DTFT}} \end{array} \quad \begin{array}{l} D_2 H^T H U_2 = I \\ H(z)H(z^{-1}) + H(-z)H(-z^{-1}) = 2 \\ |H(e^{j\omega})|^2 + |H(e^{j(\omega+\pi)})|^2 = 2 \end{array} \quad (1.14)$$

The matrix view expresses the fact that the columns of  $HU_2$  form an orthonormal set. Again, the DTFT version is the quadrature mirror formula from (3.214).

**Deterministic Autocorrelation of the Lowpass Filter** As it is widely used in filter design, we rephrase (1.13) in terms of the deterministic autocorrelation of  $g$ , given

<sup>3</sup>We remind the reader once more that we are considering exclusively real-coefficient filter banks, and thus transposition instead of Hermitian transposition in (1.13).

## 1.2. Orthogonal Two-Channel Filter Banks

9

**Lowpass Channel in a Two-Channel Orthogonal Filter Bank****Lowpass filter**

Original domain	$g_n$	$\langle g_n, g_{n-2k} \rangle_n = \delta_k$
Matrix domain	$G$	$D_2 G^T G U_2 = I$
$z$ -domain	$G(z)$	$G(z)G(z^{-1}) + G(-z)G(-z^{-1}) = 2$
DTFT domain	$G(e^{j\omega})$	$ G(e^{j\omega}) ^2 +  G(e^{j(\omega+\pi)}) ^2 = 2$ (quadrature mirror formula)
Polyphase domain	$G(z) = G_0(z^2) + z^{-1}G_1(z^2)$	$G_0(z)G_0(z^{-1}) + G_1(z)G_1(z^{-1}) = 1$

**Deterministic autocorrelation**

Original domain	$a_n = \langle g_k, g_{k+n} \rangle_k$	$a_{2k} = \delta_k$
Matrix domain	$A = G^T G$	$D_2 A U_2 = I$
$z$ -domain	$A(z) = G(z)G(z^{-1})$	$A(z) + A(-z) = 2$ $A(z) = 1 + 2 \sum_{k=0}^{\infty} a_{2k+1} (z^{2k+1} + z^{-(2k+1)})$
DTFT domain	$A(e^{j\omega}) =  G(e^{j\omega}) ^2$	$A(e^{j\omega}) + A(e^{j(\omega+\pi)}) = 2$

**Orthogonal projection onto smooth space**  $V = \overline{\text{span}}(\{g_{n-2k}\}_{k \in \mathbb{Z}})$   
 $x_V = P_V x$   $P_V = G U_2 D_2 G^T$

**Table 1.1:** Properties of the lowpass channel in an orthogonal two-channel filter bank. Properties for the highpass channel are analogous.

by (3.96):

$$\begin{array}{ccc}
 \langle g_n, g_{n-2k} \rangle = a_{2k} = \delta_k & \begin{array}{c} \text{Matrix View} \\ \leftarrow \text{ZT} \\ \leftarrow \text{DTFT} \end{array} & \begin{array}{l} D_2 A U_2 = I \\ A(z) + A(-z) = 2 \\ A(e^{j\omega}) + A(e^{j(\omega+\pi)}) = 2 \end{array} \quad (1.15)
 \end{array}$$

In the above,  $A = G^T G$  is a symmetric matrix with element  $a_k$  on the  $k$ th diagonal left/right from the main diagonal. Thus, except for  $a_0$ , all the other even terms of  $a_k$  are 0, leading to

$$A(z) \stackrel{(a)}{=} G(z)G(z^{-1}) = 1 + 2 \sum_{k=0}^{\infty} a_{2k+1} (z^{2k+1} + z^{-(2k+1)}), \quad (1.16)$$

where (a) follows from (3.143).

**Deterministic Autocorrelation of the Highpass Filter** Similarly to the lowpass filter,

$$\begin{array}{ccc}
 \langle h_n, h_{n-2k} \rangle = a_{2k} = \delta_k & \begin{array}{c} \text{Matrix View} \\ \leftarrow \text{ZT} \\ \leftarrow \text{DTFT} \end{array} & \begin{array}{l} D_2 A U_2 = I \\ A(z) + A(-z) = 2 \\ A(e^{j\omega}) + A(e^{j(\omega+\pi)}) = 2 \end{array} \quad (1.17)
 \end{array}$$

Equation (1.16) holds for this deterministic autocorrelation as well.

**Orthogonal Projection Property of the Lowpass Channel** We now look at the lowpass channel as a composition of four linear operators we just saw:

$$x_V = P_V x = GU_2 D_2 G^T x. \quad (1.18)$$

The notation is evocative of projection onto  $V$ , and we will now show that the lowpass channel accomplishes precisely this. Using (1.13), we check idempotency and self-adjointness of  $P$  *Definition 2.27*,

$$\begin{aligned} P_V^2 &= (GU_2 \underbrace{D_2 G^T}_{I} D_2 G^T) \stackrel{(a)}{=} GU_2 D_2 G^T = P_V, \\ P_V^T &= (GU_2 D_2 G^T)^T = G(U_2 D_2)^T G^T \stackrel{(b)}{=} GU_2 D_2 G^T = P_V, \end{aligned}$$

where (a) follows from (1.13) and (b) from (3.194). Indeed,  $P_V$  is an orthogonal projection operator, with the range given in (1.10a):

$$V = \overline{\text{span}}(\{g_{n-2k}\}_{k \in \mathbb{Z}}). \quad (1.19)$$

The summary of properties of the lowpass channel is given in Table 1.1.

**Orthogonal Projection Property of the Highpass Channel** The highpass channel as a composition of four linear operators (infinite matrices) is:

$$x_W = P_W x = HU_2 D_2 H^T x. \quad (1.20)$$

It is no surprise that  $P_W$  is an orthogonal projection operator with the range given in (1.10b):

$$W = \overline{\text{span}}(\{h_{n-2k}\}_{k \in \mathbb{Z}}). \quad (1.21)$$

The summary of properties of the highpass channel is given in Table 1.1 (table provided for lowpass channel, just make appropriate substitutions).

## 1.2.2 Complementary Channels and Their Properties

While we have discussed which properties each channel has to satisfy on its own, we now discuss what they have to satisfy with respect to each other to build an orthonormal basis. Intuitively, one channel has to keep what the other throws away; in other words, that channel should project to a subspace orthogonal to the range of the projection operator of the other. For example, given  $P_V$ ,  $P_W$  should project onto the leftover space between  $\ell^2(\mathbb{Z})$  and  $P_V \ell^2(\mathbb{Z})$ .

Thus, we start by assuming our filter bank in Figure 1.1(a) implements an orthonormal basis, which means that the set of basis sequences  $\{g_{n-2k}, h_{n-2k}\}_{k \in \mathbb{Z}}$  is an orthonormal set, compactly represented by (1.8). We have already used the orthonormality of the set  $\{g_{n-2k}\}_{k \in \mathbb{Z}}$  in (1.13) as well as the orthonormality of the set  $\{h_{n-2k}\}_{k \in \mathbb{Z}}$  in (1.14). What is left is that these two sets are orthogonal to each other, the third equation in (1.8).

**Orthogonality of the Lowpass and Highpass Filters** Using similar methods as before, we summarize the lowpass and highpass sequences must satisfy:

$$\begin{array}{rcc} \langle g_n, h_{n-2k} \rangle = 0 & \begin{array}{c} \xleftrightarrow{\text{Matrix View}} \\ \xleftrightarrow{\text{ZT}} \\ \xleftrightarrow{\text{DTFT}} \end{array} & \begin{array}{l} D_2 H^T G U_2 = 0 \\ G(z)H(z^{-1}) + G(-z)H(-z^{-1}) = 0 \\ G(e^{j\omega})H(e^{-j\omega}) + G(e^{j(\omega+\pi)})H(e^{-j(\omega+\pi)}) = 0 \end{array} \end{array} \quad (1.22)$$

**Deterministic Crosscorrelation of the Lowpass and Highpass Filters** Instead of the deterministic autocorrelation properties of an orthogonal filter, we look at the deterministic crosscorrelation properties of two filters orthogonal to each other:

$$\begin{array}{rcc} \langle g_n, h_{n-2k} \rangle = c_{2k} = 0 & \begin{array}{c} \xleftrightarrow{\text{Matrix View}} \\ \xleftrightarrow{\text{ZT}} \\ \xleftrightarrow{\text{DTFT}} \end{array} & \begin{array}{l} D_2 C U_2 = 0 \\ C(z) + C(-z) = 0 \\ C(e^{j\omega}) + C(e^{j(\omega+\pi)}) = 0 \end{array} \end{array} \quad (1.23)$$

In the above,  $C = H^T G$  is the deterministic crosscorrelation operator, and the deterministic crosscorrelation is given by (3.99). In particular, all the even terms of  $c$  are equal to zero.

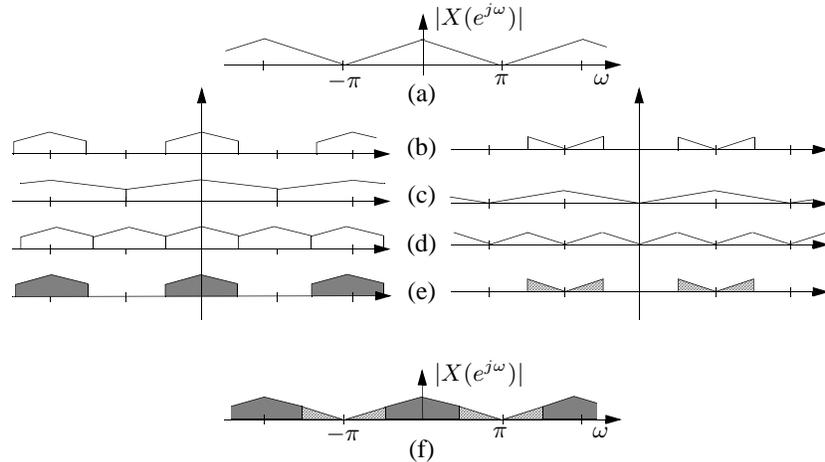
### 1.2.3 Orthogonal Two-Channel Filter Bank

We are now ready to put together everything we have developed so far. We have shown that the sequences  $\{g_{n-2k}, h_{n-2k}\}_{k \in \mathbb{Z}}$  form an orthonormal set. What is left to show is completeness: any sequence from  $\ell^2(\mathbb{Z})$  can be represented using the orthonormal basis built by our orthogonal two-channel filter bank. To do this, we must be more specific, that is, we must have an explicit form of the filters involved.

In essence, we start with an educated guess (and it will turn out to be unique, Theorem 1.2), inspired by what we have seen in the Haar case. We can also strengthen our intuition by considering a two-channel filter bank with ideal filters as in Figure 1.5. If we are given an orthogonal lowpass filter  $g$ , can we say anything about an appropriate orthogonal highpass filter  $h$  such that the two together build an orthonormal basis? A good approach would be to shift the spectrum of the lowpass filter by  $\pi$ , leading to the highpass filter. In time domain, this is equivalent to multiplying  $g_n$  by  $(-1)^n$ . Because of the orthogonality of the lowpass and highpass filters, we also reverse the impulse response of  $g$ . We will then need to shift the filter to make it causal again. Based on this discussion, we now show how, given an orthogonal filter  $g$ , it completely specifies an orthogonal two-channel filter bank implementing an orthonormal basis for  $\ell^2(\mathbb{Z})$ :

**THEOREM 1.1 (ORTHOGONAL TWO-CHANNEL FILTER BANK)** Given is an FIR filter  $g$  of even length  $L = 2\ell$ ,  $\ell \in \mathbb{Z}^+$ , orthonormal to its even shifts as in (1.13). Choose

$$h_n = \pm(-1)^n g_{-n+L-1} \quad \xleftrightarrow{\text{ZT}} \quad H(z) = \mp z^{-L+1} G(-z^{-1}). \quad (1.24)$$



**Figure 1.5:** Two-channel decomposition of a sequence using ideal filters. Left side depicts the process in the lowpass channel, while the right side depicts the process in the highpass channel. (a) Original spectrum. (b) Spectra after filtering. (c) Spectra after downsampling. (d) Spectra after upsampling. (e) Spectra after interpolation filtering. (f) Reconstructed spectrum.

Then,  $\{g_{n-2k}, h_{n-2k}\}_{k \in \mathbb{Z}}$  is an orthonormal basis for  $\ell^2(\mathbb{Z})$ , implemented by an orthogonal filter bank specified by analysis filters  $\{g_{-n}, h_{-n}\}$  and synthesis filters  $\{g_n, h_n\}$ . The expansion splits  $\ell^2(\mathbb{Z})$  as

$$\ell^2(\mathbb{Z}) = V \oplus W, \quad \text{with} \quad \begin{aligned} V &= \overline{\text{span}}(\{g_{n-2k}\}_{k \in \mathbb{Z}}), \\ W &= \overline{\text{span}}(\{h_{n-2k}\}_{k \in \mathbb{Z}}). \end{aligned} \quad (1.25)$$

*Proof.* To prove the theorem, we must prove that (i)  $\{g_{n-2k}, h_{n-2k}\}_{k \in \mathbb{Z}}$  is an orthonormal set and (ii) it is complete. The sign  $\pm$  in (1.24) just changes phase; assuming  $G(1) = \sqrt{2}$ , if the sign is positive,  $H(1) = \sqrt{2}$ , and if the sign is negative,  $H(1) = -\sqrt{2}$ . Most of the time we will implicitly assume the sign to be positive; the proof of the theorem does not change in either case.

- (i) To prove that  $\{g_{n-2k}, h_{n-2k}\}_{k \in \mathbb{Z}}$  is an orthonormal set, we must prove (1.8). The first condition is satisfied by assumption. To prove the second, that is,  $h$  is orthogonal to its even shifts, we must prove one of the conditions in (1.14). The definition of  $h$  in (1.24) implies

$$H(z)H(z^{-1}) = G(-z)G(-z^{-1}), \quad (1.26)$$

and thus,

$$H(z)H(z^{-1}) + H(-z^{-1})H(-z^{-1}) = G(-z)G(-z^{-1}) + G(z)G(z^{-1}) \stackrel{(a)}{=} 2,$$

## 1.2. Orthogonal Two-Channel Filter Banks

13

where (a) follows from (1.13).

To prove the third condition in (1.8), that is,  $h$  is orthogonal to  $g$  and all its even shifts, we must prove one of the conditions in (1.22):

$$\begin{aligned} G(z)H(z^{-1}) + G(-z)H(-z^{-1}) &\stackrel{(a)}{=} -z^{L-1}G(z)G(-z) + (-1)^L z^{L-1}G(-z)G(z) \\ &= -z^{L-1}G(z)G(-z) + z^{L-1}G(z)G(-z) \stackrel{(b)}{=} 0, \end{aligned}$$

where (a) follows from (1.24); and (b) from  $L = 2\ell$  even.

- (ii) To prove completeness, we prove that perfect reconstruction holds for any  $x \in \ell^2(\mathbb{Z})$  (an alternative would be to prove Parseval's equality  $\|x\|^2 = \|x_V\|^2 + \|x_W\|^2$ ). What we do is find  $z$ -domain expressions for  $X_V(z)$  and  $X_W(z)$  and prove they sum up to  $X(z)$ . We start with the lowpass branch. In the lowpass channel, the input  $X(z)$  is filtered by  $G(z^{-1})$ , and is then down- and upsampled, followed by filtering with  $G(z)$  (and similarly for the highpass channel). Thus, the  $z$ -transforms of  $x_V$  and  $x_W$  are:

$$X_V(z) = \frac{1}{2}G(z) [G(z^{-1})X(z) + G(-z^{-1})X(-z)], \quad (1.27a)$$

$$X_W(z) = \frac{1}{2}H(z) [H(z^{-1})X(z) + H(-z^{-1})X(-z)]. \quad (1.27b)$$

The output of the filter bank is the sum of  $x_V$  and  $x_W$ :

$$\begin{aligned} X_V(z) + X_W(z) &= \frac{1}{2} \underbrace{[G(z)G(-z^{-1}) + H(z)H(-z^{-1})]}_{S(z)} X(-z) \\ &\quad + \frac{1}{2} \underbrace{[G(z)G(z^{-1}) + H(z)H(z^{-1})]}_{T(z)} X(z). \end{aligned} \quad (1.28)$$

Substituting (1.24) into the above equation, we get:

$$\begin{aligned} S(z) &= G(z)G(-z^{-1}) + H(z)H(-z^{-1}) \\ &\stackrel{(a)}{=} G(z)G(-z^{-1}) + [-z^{-L+1}G(-z^{-1})] [-(z^{-1})^{-L+1}G(z)] \\ &= [1 + (-1)^{-L+1}] G(z)G(-z^{-1}) \stackrel{(b)}{=} 0, \end{aligned} \quad (1.29a)$$

$$\begin{aligned} T(z) &= G(z)G(z^{-1}) + H(z)H(z^{-1}) \\ &\stackrel{(c)}{=} G(z)G(z^{-1}) + G(-z^{-1})G(-z) \stackrel{(d)}{=} 2, \end{aligned} \quad (1.29b)$$

where (a) follows from (1.24); (b) from  $L = 2\ell$  is even; (c) from (1.26); and (d) from (1.13). Substituting this back into (1.28), we get

$$X_V(z) + X_W(z) = X(z), \quad (1.30)$$

proving perfect reconstruction, or, in other words, the assertion in the theorem statement that the expansion can be implemented by an orthogonal filter bank.

To show (1.25), we write (1.30) in the original domain as in (1.11):

$$x_n = \underbrace{\sum_{k \in \mathbb{Z}} \alpha_k g_{n-2k}}_{x_{V,n}} + \underbrace{\sum_{k \in \mathbb{Z}} \beta_k h_{n-2k}}_{x_{W,n}}, \quad (1.31)$$

showing that any sequence  $x \in \ell^2(\mathbb{Z})$  can be written as a sum of its projections onto two subspaces  $V$  and  $W$ , and these subspaces add up to  $\ell^2(\mathbb{Z})$ .  $V$  and  $W$  are orthogonal from (1.22) proving (1.25).

In the theorem,  $L$  is an even integer, which is a requirement for FIR filters of lengths greater than 1 (see Exercise ??). Moreover, the choice (1.24) is unique; this will be shown in Theorem 1.2. Table 1.9 summarizes various properties of orthogonal, two-channel filter banks we covered until now.

Along with the time reversal and shift, the other qualitative feature of (1.24) is modulation by  $e^{jn\pi} = (-1)^n$  (mapping  $z \rightarrow -z$  in the  $z$  domain, see (3.137)). As we said, this makes  $h$  a highpass filter when  $g$  is a lowpass filter. As an example, if we apply Theorem 1.1 to the Haar lowpass filter from (1.1a), we obtain the Haar highpass filter from (1.1b).

In applications, filters are causal. To implement a filter bank with causal filters, we make analysis filters causal (we already assumed the synthesis ones are) by shifting them both by  $(-L + 1)$ . Beware that such an implementation implies perfect reconstruction within a shift (delay), and the orthonormal basis expansion is not technically valid anymore. However, in applications this is often done, as the output sequence is a perfect replica of the input one, within a shift:  $\hat{x}_n = x_{n-L+1}$ .

### 1.2.4 Polyphase View of Orthogonal Filter Banks

As we saw in Section 3.7, downsampling introduces periodic shift variance into the system. To deal with this, we often analyze multirate systems in polyphase domain, as discussed in Section 3.7.5. The net result is that the analysis of a single-input, single-output, periodically shift-varying system is equivalent to the analysis of a multiple-input, multiple-output, shift-invariant system.

**Polyphase Representation of an Input Sequence** For two-channel filter banks, a polyphase decomposition of the input sequence is achieved by simply splitting it into its even- and odd-indexed subsequences as in (3.216), the main idea being that the sequence can be recovered from the two subsequences by upsampling, shifting and summing up, as we have seen in Figure 3.29. This simple process is called a polyphase transform (forward and inverse).

**Polyphase Representation of a Synthesis Filter Bank** To define the polyphase decomposition of the synthesis filters, we use the expressions for upsampling followed by filtering from (3.220):

$$g_{0,n} = g_{2n} \xleftrightarrow{\text{ZT}} G_0(z) = \sum_{n \in \mathbb{Z}} g_{2n} z^{-n}, \quad (1.32a)$$

$$g_{1,n} = g_{2n+1} \xleftrightarrow{\text{ZT}} G_1(z) = \sum_{n \in \mathbb{Z}} g_{2n+1} z^{-n}, \quad (1.32b)$$

$$G(z) = G_0(z^2) + z^{-1}G_1(z^2), \quad (1.32c)$$

where we split each synthesis filter into its even and odd subsequence as we have done for the input sequence  $x$ . Analogous relations hold for the highpass filter  $h$ . We can now define a *polyphase matrix*  $\Phi_p(z)$ :

$$\Phi_p(z) = \begin{bmatrix} G_0(z) & H_0(z) \\ G_1(z) & H_1(z) \end{bmatrix}. \quad (1.33)$$

As we will see in (1.37), such a matrix allows for a compact representation, analysis and computing projections in the polyphase domain.

**Polyphase Representation of an Analysis Filter Bank** The matrix in (1.33) is on the synthesis side; to get it on the analysis side, we can use the fact that this is an orthogonal filter bank. Thus, we can write

$$\tilde{G}(z) = G(z^{-1}) = G_0(z^{-2}) + zG_1(z^{-2}).$$

In other words, the polyphase components of the analysis filter are, not surprisingly, time-reversed versions of the polyphase components of the synthesis filter. We can summarize this as (we could have obtained the same result using the expression for polyphase representation of downsampling preceded by filtering (3.226)):

$$\tilde{g}_{0,n} = \tilde{g}_{2n} = g_{-2n} \xleftrightarrow{\text{ZT}} \tilde{G}_0(z) = \sum_{n \in \mathbb{Z}} g_{-2n} z^{-n}, \quad (1.34a)$$

$$\tilde{g}_{1,n} = \tilde{g}_{2n-1} = g_{-2n+1} \xleftrightarrow{\text{ZT}} \tilde{G}_1(z) = \sum_{n \in \mathbb{Z}} g_{-2n+1} z^{-n}, \quad (1.34b)$$

$$\tilde{G}(z) = G_0(z^{-2}) + zG_1(z^{-2}), \quad (1.34c)$$

with analogous relations for the highpass filter  $\tilde{h}$ , yielding the expression for the analysis polyphase matrix

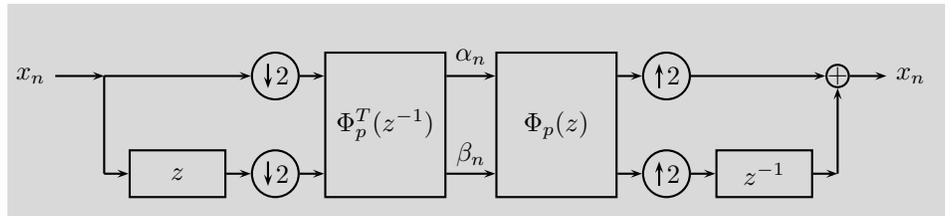
$$\tilde{\Phi}_p(z) = \begin{bmatrix} \tilde{G}_0(z) & \tilde{H}_0(z) \\ \tilde{G}_1(z) & \tilde{H}_1(z) \end{bmatrix} = \begin{bmatrix} G_0(z^{-1}) & H_0(z^{-1}) \\ G_1(z^{-1}) & H_1(z^{-1}) \end{bmatrix} = \Phi_p(z^{-1}). \quad (1.35)$$

A block diagram of the polyphase implementation of the system is given in Figure 1.6. The left part shows the reconstruction of the original sequence using the synthesis polyphase matrix.<sup>4</sup> The right part shows the computation of expansion coefficient sequences  $\alpha$  and  $\beta$ ; note that as usual, the analysis matrix (polyphase in this case) is taken as a transpose, as it operates on the input sequence. To check that, compute these expansion coefficient sequences:

$$\begin{aligned} \begin{bmatrix} \alpha(z) \\ \beta(z) \end{bmatrix} &= \Phi_p^T(z^{-1}) \begin{bmatrix} X_0(z) \\ X_1(z) \end{bmatrix} = \begin{bmatrix} G_0(z^{-1}) & G_1(z^{-1}) \\ H_0(z^{-1}) & H_1(z^{-1}) \end{bmatrix} \begin{bmatrix} X_0(z) \\ X_1(z) \end{bmatrix} \\ &= \begin{bmatrix} G_0(z^{-1})X_0(z) + G_1(z^{-1})X_1(z) \\ H_0(z^{-1})X_0(z) + H_1(z^{-1})X_1(z) \end{bmatrix}. \end{aligned} \quad (1.36)$$

We can obtain exactly the same expressions if we substitute (1.34) into the expression for downsampling by 2 preceded by filtering in (3.201a).

<sup>4</sup>A comment is in order: we typically put the lowpass filter in the lower branch, but in matrices it appears in the first row/column, leading to a slight inconsistency when the filter bank is depicted in the polyphase domain.



**Figure 1.6:** Polyphase representation of a two-channel orthogonal filter bank.

**Polyphase Representation of an Orthogonal Filter Bank** The above polyphase expressions allow us now to compactly represent an orthogonal two-channel filter bank in the polyphase domain:

$$X(z) = \begin{bmatrix} 1 & z^{-1} \end{bmatrix} \Phi_p(z^2) \Phi_p^T(z^{-2}) \begin{bmatrix} X_0(z^2) \\ X_1(z^2) \end{bmatrix}. \quad (1.37)$$

From (1.24), we get that the polyphase components of  $H$  are

$$H_0(z) = \pm z^{-L/2+1} G_1(z^{-1}), \quad (1.38a)$$

$$H_1(z) = \mp z^{-L/2+1} G_0(z^{-1}), \quad (1.38b)$$

leading to the polyphase matrix

$$\Phi_p(z) = \begin{bmatrix} G_0(z) & \pm z^{-L/2+1} G_1(z^{-1}) \\ G_1(z) & \mp z^{-L/2+1} G_0(z^{-1}) \end{bmatrix}. \quad (1.39)$$

Since  $g$  is orthogonal to its even translates, substitute (1.32) into the  $z$ -domain version of (1.13) to get the condition for orthogonality of a filter in polyphase form:

$$G_0(z)G_0(z^{-1}) + G_1(z)G_1(z^{-1}) = 1. \quad (1.40)$$

Using this, the determinant of  $\Phi_p(z)$  becomes  $-z^{-L/2+1}$ . From (1.37), the polyphase matrix  $\Phi_p(z)$  satisfies the following:

$$\Phi_p(z) \Phi_p^T(z^{-1}) = I, \quad (1.41)$$

a paraunitary matrix as in (3.303a). In fact, (1.39), together with (1.40), define the most general  $2 \times 2$ , real-coefficient, causal FIR lossless matrix, a fact we summarize in form of a theorem, the proof of which can be found in [101]:

**THEOREM 1.2 (GENERAL FORM OF A PARAUNITARY MATRIX)** The most general  $2 \times 2$ , real-coefficient, causal FIR lossless matrix is given by (1.39), where  $G_0$  and  $G_1$  satisfy (1.40) and  $L/2 - 1$  is the degree of  $G_0(z)$ ,  $G_1(z)$ .

EXAMPLE 1.1 (HAAR FILTER BANK IN POLYPHASE FORM) The Haar filters (1.1) are extremely simple in polyphase form: Since they are both of length 2, their polyphase components are of length 1. The polyphase matrix is simply

$$\Phi_p(z) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (1.42)$$

The form of the polyphase matrix for the Haar orthonormal basis is exactly the same as the Haar orthonormal basis for  $\mathbb{R}^2$ , or one block of the Haar orthonormal basis infinite matrix  $\Phi$  from (??). This is true only when a filter bank implements the so-called *block transform*, that is, when the nonzero support of the basis sequences is equal to the sampling factor, 2 in this case.

The polyphase notation and the associated matrices are powerful tools to derive filter bank results. We now rephrase what it means for a filter bank to be orthogonal—implement an orthonormal basis, in polyphase terms.

THEOREM 1.3 (PARAUNITARY POLYPHASE MATRIX AND ORTHONORMAL BASIS) A  $2 \times 2$  polyphase matrix  $\Phi_p(z)$  is paraunitary if and only if the associated two-channel filter bank implements an orthonormal basis for  $\ell^2(\mathbb{Z})$ .

*Proof.* If the polyphase matrix is paraunitary, then the expansion it implements is complete, due to (1.41). To prove that the expansion is an orthonormal basis, we must show that the basis sequences form an orthonormal set. From (1.39) and (1.41), we get (1.40). Substituting this into the  $z$ -domain version of (1.13), we see that it holds, and thus  $g$  and its even shifts form an orthonormal set. Because  $h$  is given in terms of  $g$  as (1.24),  $h$  and its even shifts form an orthonormal set as well. Finally, because of the way  $h$  is defined,  $g$  and  $h$  are orthogonal by definition and so are their even shifts.

The argument in the other direction is similar; we start with an orthonormal basis implemented by a two-channel filter bank. That means we have template sequences  $g$  and  $h$  related via (1.24), and their even shifts, all together forming an orthonormal basis. We can now translate those conditions into  $z$ -transform domain using (1.13) and derive the corresponding polyphase-domain versions, such as the one in (1.40). These lead to the polyphase matrix being paraunitary.

We have seen in *Chapter 3* that we can characterize vector sequences using deterministic autocorrelation matrices (see *Table 3.13*). We use this now to describe the deterministic autocorrelation of a vector sequence of expansion coefficients  $[\alpha_n \ \beta_n]^T$ , as

$$\begin{aligned} A_{p,\alpha}(z) &= \begin{bmatrix} A_\alpha(z) & C_{\alpha,\beta}(z) \\ C_{\beta,\alpha}(z) & A_\beta(z) \end{bmatrix} = \begin{bmatrix} \alpha(z) \alpha(z^{-1}) & \alpha(z) \beta(z^{-1}) \\ \beta(z) \alpha(z^{-1}) & \beta(z) \beta(z^{-1}) \end{bmatrix} \\ &= \begin{bmatrix} \alpha(z) \\ \beta(z) \end{bmatrix} [\alpha(z^{-1}) \ \beta(z^{-1})] \\ &\stackrel{(a)}{=} \Phi_p^T(z^{-1}) \begin{bmatrix} X_0(z) \\ X_1(z) \end{bmatrix} [X_1(z^{-1}) \ X_0(z^{-1})] \Phi_p(z) \\ &= \Phi_p^T(z^{-1}) A_{p,x}(z) \Phi_p(z), \end{aligned} \quad (1.43)$$

where (a) follows from (1.36), and  $A_{p,x}$  is the deterministic autocorrelation matrix of the vector of polyphase components of  $x$ . This deterministic autocorrelation matrix can be seen as a *filtered* deterministic autocorrelation of the input. We now have the following result:

**THEOREM 1.4 (FILTERED DETERMINISTIC AUTOCORRELATION MATRIX)** Given is a  $2 \times 2$  paraunitary polyphase matrix  $\Phi_p(e^{j\omega})$ . Then the filtered deterministic autocorrelation matrix,  $A_{p,\alpha}(e^{j\omega})$ , is positive semidefinite.

*Proof.* Since  $\Phi_p(z)$  is paraunitary,  $\Phi_p(e^{j\omega})$  is unitary on the unit circle. This further means that:

$$\begin{bmatrix} \cos \theta & \sin \theta \end{bmatrix} \Phi_p^T(e^{-j\omega}) = \begin{bmatrix} \cos \phi & \sin \phi \end{bmatrix}, \quad (1.44)$$

for some  $\phi$ . We can now write:

$$\begin{aligned} \begin{bmatrix} \cos \theta & \sin \theta \end{bmatrix} A_{p,\alpha}(e^{j\omega}) \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} &\stackrel{(a)}{=} \begin{bmatrix} \cos \theta & \sin \theta \end{bmatrix} \Phi_p^T(e^{-j\omega}) A_{p,x}(e^{j\omega}) \Phi_p(e^{j\omega}) \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} \\ &\stackrel{(b)}{=} \begin{bmatrix} \cos \phi & \sin \phi \end{bmatrix} A_{p,x}(e^{j\omega}) \begin{bmatrix} \cos \phi \\ \sin \phi \end{bmatrix} \stackrel{(c)}{\geq} 0, \end{aligned}$$

where (a) follows from (1.43); (b) from (1.44); and (c) from TBD, proving the theorem.

### 1.2.5 Polynomial Approximation by Filter Banks

An important class of orthogonal filter banks are those that have polynomial approximation properties; these filter banks will approximate polynomials of a certain degree<sup>5</sup> in the lowpass (coarse) branch, while, at the same time, blocking those same polynomials in the highpass (detail) branch. To derive these filter banks, we recall what we have learned in *Section 3.B.1: Convolution of a polynomial sequence  $x$  with a differencing filter  $(\delta_n - \delta_{n-1})$ , or, multiplication of  $X(z)$  by  $(1 - z^{-1})$ , reduces the degree of the polynomial by 1. In general, to block a polynomial of degree  $(N - 1)$ ,  $x_n = \sum_{k=0}^{N-1} a_k n^k$ , we need a filter of the form:*

$$(1 - z^{-1})^N R'(z). \quad (1.45)$$

Let us now apply what we just learned to two-channel orthogonal filter banks with polynomial sequences as inputs. We will construct the analysis filter in the highpass branch to have  $N$  zeros at  $z = 1$ , thus blocking polynomials of degree up to  $(N - 1)$ . Of course, since the filter bank is perfect reconstruction, whatever disappeared in the highpass branch must be preserved in the lowpass one; thus, the lowpass branch will reconstruct polynomials of degree  $(N - 1)$ . In other words,  $x_V$  will be a polynomial approximation of the input sequence a certain degree.

<sup>5</sup>We restrict our attention to finitely-supported polynomial sequences, ignoring the boundary issues. If this were not the case, these sequences would not belong to any  $\ell^p$  space.

## 1.2. Orthogonal Two-Channel Filter Banks

19

To construct such a filter bank, we start with the analysis highpass filter  $\tilde{h}$  which must be of the form (1.45); we write it as:

$$\tilde{H}(z) \stackrel{(a)}{=} (1 - z^{-1})^N \underbrace{\mp z^{L-1} R(-z)}_{R'(z)} = \mp z^{L-1} (1 - z^{-1})^N R(-z) \stackrel{(b)}{=} \mp z^{L-1} G(-z),$$

where in (a) we have chosen  $R'(z)$  to lead to a simple form of  $G(z)$  in what follows; and (b) follows from Table 1.9, allowing us to directly read the synthesis lowpass as

$$G(z) = (1 + z^{-1})^N R(z). \quad (1.46)$$

If we maintain the convention that  $g$  is causal and of length  $L$ , then  $R(z)$  is a polynomial in  $z^{-1}$  of degree  $(L-1-N)$ . Of course,  $R(z)$  has to be chosen appropriately, so as to obtain an orthogonal filter bank.

Putting at least one zero at  $z = -1$  in  $G(z)$  makes a lot of signal processing sense. After all,  $z = -1$  corresponds to  $\omega = \pi$ , the maximum discrete frequency; it is thus natural for a lowpass filter to have a zero at  $z = -1$  and block that highest frequency. Putting more than one zero at  $z = -1$  has further approximation advantages, as the Theorem 1.5 specifies, and as we will see in wavelet constructions in later chapters.

**THEOREM 1.5 (POLYNOMIAL REPRODUCTION)** Given is an orthogonal filter bank in which the synthesis lowpass filter  $G(z)$  has  $N$  zeros at  $z = -1$ . Then polynomial sequences up to degree  $(N-1)$  and of finite support are reproduced in the lowpass approximation subspace spanned by  $\{g_{n-2k}\}_{k \in \mathbb{Z}}$ .

*Proof.* By assumption, the synthesis filter  $G(z)$  is given by (1.46). From Table 1.9, the analysis highpass filter is of the form  $\mp z^{L-1} G(-z)$ , which means it has a factor  $(1 - z^{-1})^N$ , that is, it has  $N$  zeros at  $z = 1$ . From our discussion, this factor annihilates a polynomial input of degree  $(N-1)$ , resulting in  $\beta = 0$  and  $x_W = 0$ . Because of the perfect reconstruction property,  $x = x_V$ , showing that the polynomial sequences are reproduced by a linear combination of  $\{g_{n-2k}\}_{k \in \mathbb{Z}}$ , as in (1.11a).

Polynomial reproduction by the lowpass channel and polynomial cancellation in the highpass channel are basic features in wavelet approximations. In particular, the cancellation of polynomials of degree  $(N-1)$  is also called the *zero-moment property* of the filter (see (3.140a)):

$$m_k = \sum_{n \in \mathbb{Z}} n^k h_n = 0, \quad k = 0, 1, \dots, N-1, \quad (1.47)$$

that is,  $k$ th-order moments of  $h$  up to  $(N-1)$  are zero (see Exercise ??).

### 1.3 Design of Orthogonal Two-Channel Filter Banks

To design a two-channel orthogonal filter bank, it suffices to design one orthogonal filter—the lowpass synthesis  $g$  with the  $z$ -transform  $G(z)$  satisfying (1.13); we have seen how the other three filters follow (Table 1.9). The design is based on (1) finding a deterministic autocorrelation function satisfying (1.15) (it is symmetric, positive semi-definite and has a single nonzero even-indexed coefficient; and (2) factoring that deterministic autocorrelation  $A(z) = G(z)G(z^{-1})$  into its spectral factors (many factorizations are possible, see *Section 3.5*).<sup>6</sup>

We consider three different designs. The first tries to approach an ideal half-band lowpass filter, the second aims at polynomial approximation, while the third uses lattice factorization in polyphase domain.

#### 1.3.1 Lowpass Approximation Design

Assume we wish to get our lowpass synthesis filter  $G(e^{j\omega})$  to be as close as possible to an ideal lowpass halfband filter as in TBD. Since according to (3.96) the deterministic autocorrelation of  $g$  can be expressed in the DTFT domain as  $A(e^{j\omega}) = |G(e^{j\omega})|^2$ , this deterministic autocorrelation is an ideal lowpass halfband function as well:

$$A(e^{j\omega}) = \begin{cases} 2, & \text{if } |\omega| < \pi/2; \\ 0, & \text{otherwise.} \end{cases} \quad (1.48)$$

From *Table 3.5*, the deterministic autocorrelation sequence is

$$a_n = \frac{1}{2\pi} \int_{-\pi/2}^{\pi/2} 2 e^{jn\omega} d\omega = \text{sinc}(n\pi/2), \quad (1.49)$$

a valid deterministic autocorrelation; it has a single nonzero even-indexed coefficient ( $a_0 = 1$ ) and is positive semi-definite. To get a realizable function, we apply a symmetric window function  $w$  that decays to zero. The new deterministic autocorrelation  $a'$  is the pointwise product

$$a'_n = a_n w_n. \quad (1.50)$$

Clearly,  $a'$  is symmetric and still has a single nonzero even-indexed coefficient. However, this is not enough for  $a'$  to be a deterministic autocorrelation. We can see this in frequency domain,

$$A'(e^{j\omega}) \stackrel{(a)}{=} \frac{1}{2\pi} A(e^{j\omega}) * W(e^{j\omega}), \quad (1.51)$$

where we used the convolution in frequency property (3.94). In general, (1.51) is not nonnegative for all frequencies anymore, and thus not a valid deterministic autocorrelation. One easy way to enforce nonnegativity is to choose  $W(e^{j\omega})$  itself positive, for example as the deterministic autocorrelation of another window  $w'$ , or

$$W(e^{j\omega}) = |W'(e^{j\omega})|^2.$$

---

<sup>6</sup>Recall that we only consider real-coefficient filters, thus  $a$  is symmetric and not Hermitian symmetric.

If  $w'$  is of norm 1, then  $w_0 = 1$ , and from (1.50),  $a'_0 = 1$  as well. Therefore, since  $A(e^{j\omega})$  is real and positive,  $A'(e^{j\omega})$  will be as well. The resulting sequence  $a'$  and its  $z$ -transform  $A'(z)$  can then be used in spectral factorization (see *Section 3.5.3*) to obtain an orthogonal filter  $g$ .

EXAMPLE 1.2 (LOWPASS APPROXIMATION DESIGN OF ORTHOGONAL FILTERS)

We design a length-4 filter by the lowpass approximation method. Its deterministic autocorrelation is of length 7 with the target impulse response obtained by evaluating (1.49):

$$a = \left[ \dots \quad 0 \quad -\frac{2}{3\pi} \quad 0 \quad \frac{2}{\pi} \quad \boxed{1} \quad \frac{2}{\pi} \quad 0 \quad -\frac{2}{3\pi} \quad 0 \quad \dots \right]^T.$$

For the window  $w$ , we take it to be the deterministic autocorrelation of the sequence  $w'_n$ , which is specified by  $w'_n = 1/2$  for  $0 \leq n \leq 3$ , and  $w'_n = 0$  otherwise:

$$w = \left[ \dots \quad 0 \quad 0 \quad \frac{1}{4} \quad \frac{1}{2} \quad \frac{3}{4} \quad \boxed{1} \quad \frac{3}{4} \quad \frac{1}{2} \quad \frac{1}{4} \quad 0 \quad 0 \quad \dots \right]^T.$$

Using (1.50), we obtain the new deterministic autocorrelation of the lowpass filter as

$$a' = \left[ \dots \quad 0 \quad -\frac{1}{6\pi} \quad 0 \quad \frac{3}{2\pi} \quad \boxed{1} \quad \frac{3}{2\pi} \quad 0 \quad -\frac{1}{6\pi} \quad 0 \quad \dots \right]^T.$$

Factoring this deterministic autocorrelation (requires numerical polynomial root finding) gives

$$g \approx \left[ \dots \quad 0 \quad \boxed{0.832} \quad 0.549 \quad 0.0421 \quad -0.0637 \quad 0 \quad \dots \right]^T.$$

The impulse response and frequency response of  $g$  are shown in Figure 1.7.

The method presented is very simple, and does not lead to the best designs. For better designs, one uses standard filter design procedures followed by adjustments to ensure positivity. For example, consider (1.51) again, and define

$$\min_{\omega \in [-\pi, \pi]} A'(e^{j\omega}) = \varepsilon.$$

If  $\varepsilon \geq 0$ , we are done, otherwise, we simply choose a new function

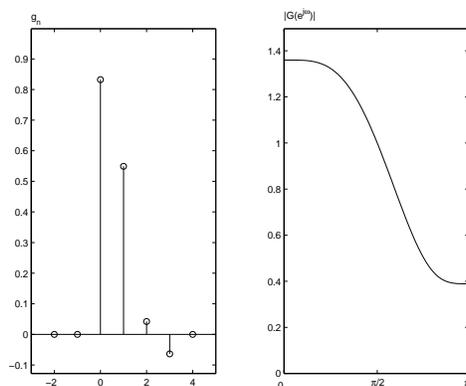
$$A''(e^{j\omega}) = A'(e^{j\omega}) - \varepsilon,$$

which is now nonnegative, allowing us to perform spectral factorization. Filters designed using this method are tabulated in [100].

### 1.3.2 Polynomial Approximation Design

Recall that a lowpass filter  $G(z)$  with  $N$  zeros at  $z = -1$  as in (1.46) reproduces polynomials up to degree  $(N - 1)$ . Thus, the goal of this design procedure is to find a deterministic autocorrelation  $A(z)$  of the form

$$A(z) = G(z)G(z^{-1}) = (1 + z^{-1})^N(1 + z)^N Q(z), \quad (1.52)$$



**Figure 1.7:** Orthogonal filter design based on lowpass approximation in Example 1.2. (a) Impulse response. (b) Frequency response.

with  $Q(z)$  chosen such that (1.15) is satisfied, that is,

$$A(z) + A(-z) = 2, \quad (1.53)$$

$Q(z) = Q(z^{-1})$  ( $q_n$  symmetric in time domain), and  $Q(z)$  is nonnegative on the unit circle. Satisfying these conditions allows one to find a spectral factor of  $A(z)$  with  $N$  zeros at  $z = -1$ , and this spectral factor is the desired orthogonal filter. We illustrate this procedure through an example.

**EXAMPLE 1.3 (POLYNOMIAL APPROXIMATION DESIGN OF ORTHOGONAL FILTERS)**

We will design a filter  $g$  such that it reproduces linear polynomials, that is,  $N = 2$ :

$$A(z) = (1 + z^{-1})^2(1 + z)^2 Q(z) = (z^{-2} + 4z^{-1} + 6 + 4z + z^2) Q(z).$$

Can we now find  $Q(z)$  so as to satisfy (1.53), in particular, a minimum-degree solution? We try with (remember  $q_n$  is symmetric)

$$Q(z) = az + b + az^{-1}$$

and compute  $A(z)$  as

$$A(z) = a(z^3 + z^{-3}) + (4a + b)(z^2 + z^{-2}) + (7a + 4b)(z + z^{-1}) + (8a + 6b).$$

To satisfy (1.53),  $A(z)$  must have a single nonzero even-indexed coefficient. We thus need to solve the following pair of equations:

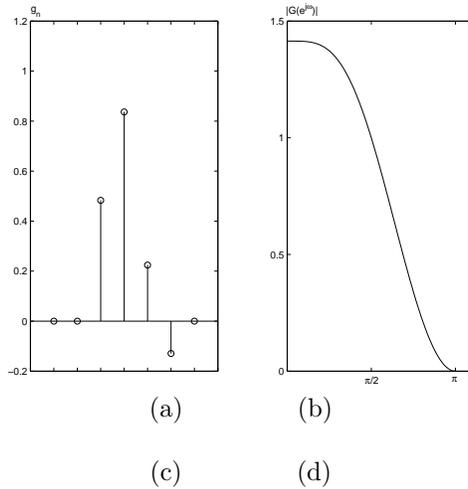
$$\begin{aligned} 4a + b &= 0, \\ 8a + 6b &= 1, \end{aligned}$$

yielding  $a = -1/16$  and  $b = 1/4$ . Thus, our candidate factor is

$$Q(z) = \frac{1}{4} \left( -\frac{1}{4}z^{-1} + 1 - \frac{1}{4}z \right).$$

## 1.3. Design of Orthogonal Two-Channel Filter Banks

23



**Figure 1.8:** Orthogonal filter design based on polynomial approximation in Example 1.3. (a) Impulse response. (b) Frequency response. (c) Linear  $x$  is preserved in  $V$ . (d) Only the linear portion of the quadratic  $x$  is preserved in  $V$ ; the rest shows in  $W$ .

It remains to check whether  $Q(e^{j\omega})$  is nonnegative:

$$Q(e^{j\omega}) = \frac{1}{4} \left( 1 - \frac{1}{4}(e^{j\omega} + e^{-j\omega}) \right) = \frac{1}{4} \left( 1 - \frac{1}{2} \cos \omega \right) > 0$$

since  $|\cos(\omega)| \leq 1$ . So  $Q(z)$  is a valid deterministic autocorrelation and can be written as  $Q(z) = R(z)R(z^{-1})$ . Extracting its causal spectral factor

$$R(z) = \frac{1}{4\sqrt{2}}(1 + \sqrt{3} + (1 - \sqrt{3})z^{-1}),$$

the causal orthogonal lowpass filter with 2 zeros at  $z = -1$  becomes

$$\begin{aligned} G(z) &= (1 + z^{-1})^2 R(z) \\ &= \frac{1}{4\sqrt{2}} \left[ (1 + \sqrt{3}) + (3 + \sqrt{3})z^{-1} + (3 - \sqrt{3})z^{-2} + (1 - \sqrt{3})z^{-3} \right]. \end{aligned}$$

This filter is one of the filters from the Daubechies family of orthogonal filters. Its impulse and frequency responses are shown in Figure 1.8. The rest of the filters in the filter bank can be found from Table 1.9.

In the example, we saw that solving a linear system followed by spectral factorization were the key steps. In general, for  $G(z)$  with  $N$  zeros at  $z = -1$ , the minimum-degree  $R(z)$  to obtain an orthogonal filter is of degree  $(N - 1)$ , corresponding to  $N$  unknown coefficients.  $Q(z) = R(z)R(z^{-1})$  is obtained by solving an  $N \times N$  linear system (to satisfy  $A(z) + A(z) = 2$ ), followed by spectral factorization

Step	Operation
1.	Choose $N$ , the number of zeros at $z = -1$
2.	$G(z) = (1 + z^{-1})^N R(z)$ , where $R(z)$ is causal with powers $(0, -1, \dots, -N + 1)$
3.	$A(z) = (1 + z^{-1})^N (1 + z)^N Q(z)$ , where $Q(z)$ is symmetric and has powers $(-(N - 1), \dots, 0, \dots, (N + 1))$
4.	$A(z) + A(-z) = 2$ . This leads to $N$ linear constraints on the coefficients of $Q(z)$
5.	Solve the $N \times N$ linear system for the coefficients of $Q(z)$
6.	Take the spectral factor of $Q(z) = R(z)R(z^{-1})$ (for example, the minimum-phase factor, see <i>Section 3.5</i> )
7.	The minimum phase orthogonal filter is $G(z) = (1 + z^{-1})^N R(z)$

**Table 1.2:** Design of orthogonal lowpass filters with maximum number of zeros at  $z = -1$ .

	$L = 4$	$L = 6$	$L = 8$	$L = 10$	$L = 12$
$g_0$	0.482962913	0.332670553	0.230377813309	0.160102398	0.111540743350
$g_1$	0.836516304	0.806891509	0.714846570553	0.603829270	0.494623890398
$g_2$	0.224143868	0.459877502	0.630880767930	0.724308528	0.751133908021
$g_3$	-0.129409522	-0.135011020	-0.027983769417	0.138428146	0.315250351709
$g_4$		-0.085441274	-0.187034811719	-0.242294887	-0.226264693965
$g_5$		0.035226292	0.030841381836	-0.032244870	-0.129766867567
$g_6$			0.032883011667	0.077571494	0.097501605587
$g_7$			-0.010597401785	-0.006241490	0.027522865530
$g_8$				-0.012580752	-0.031582039318
$g_9$				0.003335725	0.000553842201
$g_{10}$					0.004777257511
$g_{11}$					-0.001077301085

**Table 1.3:** Orthogonal filters with maximum number of zeros at  $z = -1$  (from [29]). For a lowpass filter of even length  $L = 2\ell$ , there are  $L/2$  zeros at  $z = -1$ .

to produce the desired result. (It can be verified that  $Q(e^{j\omega}) \geq 0$ .) These steps are summarized in Table 1.2, while Table 1.3 gives filter-design examples.

Note that  $A(z)$  has the following form when evaluated on the unit circle:

$$A(e^{j\omega}) = 2^N (1 + \cos \omega)^N Q(e^{j\omega}),$$

with  $Q(e^{j\omega})$  real and positive. Since  $A(e^{j\omega})$  and its  $(2N - 1)$  derivatives are zero at  $\omega = \pi$ ,  $|G(e^{j\omega})|$  and its  $(N - 1)$  derivatives are zero at  $\omega = \pi$ . Moreover, because of the quadrature mirror formula (3.214),  $|G(e^{j\omega})|$  and its  $(N - 1)$  derivatives are zero at  $\omega = 0$  as well. These facts are the topic of Exercise ??.

### 1.3.3 Lattice Factorization Design

When discussing the polyphase view of filter banks in Section 1.2.4, we saw that orthogonality of a two-channel filter bank is connected to its polyphase matrix being paraunitary. The following elegant factorization result is used in the design of that paraunitary matrix:

**THEOREM 1.6** The polyphase matrix of any real-coefficient, causal, FIR orthogonal two-channel filter bank can be written as

$$\Phi_p(z) = U \prod_{k=1}^{K-1} \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} R_k, \quad (1.54)$$

where  $U$  is a general unitary matrix as in (2.227) (either a rotation as in (2.229a) or a rotoinversion (2.229b)), and  $R_k$ ,  $k = 1, 2, \dots, K-1$ , are rotation matrices as in (2.229a).

The resulting filters are of even length  $2K$  (see Exercise ??). That the above structure produces an orthogonal filter bank is clear as the corresponding polyphase matrix  $\Phi_p(z)$  is paraunitary. Proving that any orthogonal filter bank can be written in the form of (1.54) is a bit more involved. It is based on the result that for two, real-coefficient polynomials  $P_{K-1}$  and  $Q_{K-1}$  of degree  $(K-1)$ , with  $p_{K-1}(0) p_{K-1}(K-1) \neq 0$  (and  $P_{K-1}, Q_{K-1}$  are power complementary as in (3.214)), there exists another pair  $P_{K-2}, Q_{K-2}$  such that

$$\begin{bmatrix} P_{K-1}(z) \\ Q_{K-1}(z) \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} P_{K-2}(z) \\ z^{-1} Q_{K-2}(z) \end{bmatrix}. \quad (1.55)$$

Repeatedly applying the above result to (1.39) one obtains the lattice factorization given in (1.6). The details of the proof are given in [100].

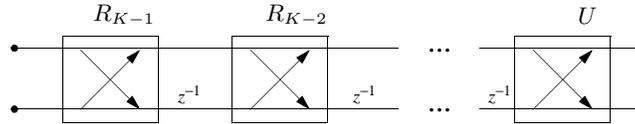
Using the factored form, designing an orthogonal filter bank amounts to choosing  $U$  and a set of angles  $(\theta_0, \theta_1, \dots, \theta_{K-1})$ . For example, the Haar filter bank in lattice form amounts to keeping only the constant-matrix term,  $U$ , as in (1.42), a rotoinversion. The factored form also suggests a structure, called a lattice, convenient for hardware implementations (see Figure 1.9).

How do we impose particular properties, such as zeros at  $\omega = \pi$ , or,  $z = -1$ , for the lowpass filter  $G(z)$ ? Write the following set of equations:

$$G(z)|_{z=1} \stackrel{(a)}{=} (G_0(z^2) + z^{-1}G_1(z^2))|_{z=1} = G_0(1) + G_1(1) \stackrel{(b)}{=} \sqrt{2}, \quad (1.56a)$$

$$G(z)|_{z=-1} \stackrel{(c)}{=} (G_0(z^2) + z^{-1}G_1(z^2))|_{z=-1} = G_0(1) - G_1(1) \stackrel{(d)}{=} 0, \quad (1.56b)$$

where (a) and (c) follow from (1.32); (b) from (1.13) and the requirement that  $G(z)$



**Figure 1.9:** Two-channel lattice factorization of paraunitary filter banks. The  $2 \times 2$  blocks  $R_k$  are rotation matrices, and  $U$  is a general unitary matrix (rotation or rotoinversion). The inputs are the polyphase components of the sequence  $x$ , and the output are the lowpass and highpass channels.

be 0 at  $z = -1$ , and similarly for (d). We can rewrite these compactly as:

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \underbrace{\begin{bmatrix} G_0(1) \\ G_0(1) \end{bmatrix}}_G = \begin{bmatrix} \sqrt{2} \\ 0 \end{bmatrix}. \quad (1.57)$$

The vector  $G$  above is just the first column of  $\Phi_p(z^2)|_{z=-1}$ , which, in turn, is either a product of (1)  $K$  rotations by  $(\theta_0, \theta_1, \dots, \theta_{K-1})$ , or, (2) one rotoinversion by  $\theta_0$  and  $K - 1$  rotations  $(\theta_1, \dots, \theta_{K-1})$ . The solution to the above is:

$$\sum_{k=0}^{K-1} \theta_k = 2n\pi + \frac{\pi}{4}, \quad U \text{ is a rotation,} \quad (1.58a)$$

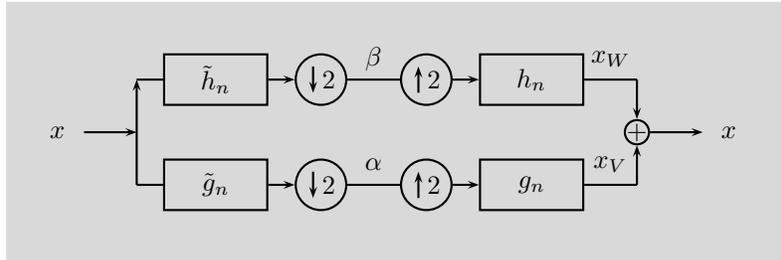
$$\theta_0 - \sum_{k=1}^{K-1} \theta_k = 2n\pi + \frac{\pi}{4}, \quad U \text{ is a rotoinversion,} \quad (1.58b)$$

for some  $n \in \mathbb{Z}$ . Imposing higher-order zeros at  $z = -1$ , as required for higher-order polynomial reproduction, leads to more complicated algebraic constraints. As an example, choosing  $\theta_0 = \pi/3$  and  $\theta_1 = -\pi/12$  leads to a double zero at  $z = -1$ , and is thus the lattice version of the filter designed in Example 1.3 (see Exercise ??). In general, design problems in lattice factored form are nonlinear and thus nontrivial.

## 1.4 Biorthogonal Two-Channel Filter Banks

While orthogonal filter banks have many attractive features, one eludes them: when restricted to real-coefficient, FIR filters, solutions that are both orthonormal and linear phase do not exist except for Haar filters. This is one of the key motivations for looking beyond the orthogonal case, as well as for the popularity of biorthogonal filter banks, especially in image processing. Similarly to the orthogonal case, we want to find out how to implement biorthogonal bases using filter banks, in particular, those having certain time and frequency localization properties. From *Definition 2.43*, we know that a system  $\{\varphi_k, \tilde{\varphi}_k\}$  constitutes a pair of biorthogonal bases of the Hilbert space  $\ell^2(\mathbb{Z})$ , if (1) they satisfy biorthogonality constraints (2.108):

$$\langle \varphi_k, \tilde{\varphi}_i \rangle = \delta_{k-i} \quad \leftrightarrow \quad \Phi \tilde{\Phi}^T = \tilde{\Phi} \Phi^T = I, \quad (1.59)$$



**Figure 1.10:** A biorthogonal two-channel analysis/synthesis filter bank. The output is the sum of the lowpass approximation  $x_V$  and its highpass counterpart  $x_W$ .

where  $\Phi$  is an infinite matrix having  $\varphi_k$  as its columns, while  $\tilde{\Phi}$  is an infinite matrix having  $\tilde{\varphi}_k$  as its columns; and (2) it is complete:

$$x = \sum_{k \in \mathbb{Z}} X_k \varphi_k = \Phi X = \sum_{k \in \mathbb{Z}} \tilde{X}_k \tilde{\varphi}_k = \tilde{\Phi} \tilde{X}, \tag{1.60}$$

for all  $x \in \ell^2(\mathbb{Z})$ , where

$$X_k = \langle \tilde{\varphi}_k, x \rangle \leftrightarrow X = \tilde{\Phi}^T x, \quad \text{and} \quad \tilde{X}_k = \langle \varphi_k, x \rangle \leftrightarrow \tilde{X} = \Phi^T x.$$

It is not a stretch now to imagine that, similarly to the orthogonal case, we are looking for two template basis sequences—a lowpass/highpass pair  $g$  and  $h$ , and a dual pair  $\tilde{g}$  and  $\tilde{h}$  so that the biorthogonality constraints (1.59) are satisfied. Under the right circumstances described in this section, such a filter bank will compute a biorthogonal expansion. Assume that indeed, we are computing such an expansion. Start from the reconstructed output as in Figure 1.10:

$$x = x_V + x_W = \sum_{k \in \mathbb{Z}} \alpha_k g_{n-2k} + \sum_{k \in \mathbb{Z}} \beta_k h_{n-2k},$$

or

$$\underbrace{\begin{bmatrix} \vdots \\ x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \end{bmatrix}}_x = \underbrace{\begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ \dots & g_0 & h_0 & 0 & 0 & 0 & 0 & \dots \\ \dots & g_1 & h_1 & 0 & 0 & 0 & 0 & \dots \\ \dots & g_2 & h_2 & g_0 & h_0 & 0 & 0 & \dots \\ \dots & g_3 & h_3 & g_1 & h_1 & 0 & 0 & \dots \\ \dots & g_4 & h_4 & g_2 & h_2 & g_0 & h_0 & \dots \\ \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} \vdots \\ \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \end{bmatrix}}_X = \Phi X, \tag{1.61}$$

exactly the same as (1.7). As in (1.7),  $g_{n-2k}$  and  $h_{n-2k}$  are the impulse responses of the synthesis filters  $g$  and  $h$  shifted by  $2k$ , and  $\alpha_k$  and  $\beta_k$  are the outputs of the

analysis filter bank downsampled by 2. The basis sequences are the columns of

$$\Phi = \{\varphi_k\}_{k \in \mathbb{Z}} = \{\varphi_{2k}, \varphi_{2k+1}\}_{k \in \mathbb{Z}} = \{g_{n-2k}, h_{n-2k}\}_{k \in \mathbb{Z}}, \quad (1.62)$$

that is, the even-indexed basis sequences are the impulse responses of the synthesis lowpass filter and its even shifts, while the odd-indexed basis sequences are the impulse responses of the synthesis highpass filter and its even shifts.

So far, the analysis has been identical to that of orthogonal filter banks; we repeated it here for emphasis. Since we are implementing a biorthogonal expansion, the transform coefficients  $\alpha_k$  and  $\beta_k$  are inner products between the dual basis sequences and the input sequence:  $\alpha_k = \langle x, \tilde{\varphi}_{2k} \rangle$ ,  $\beta_k = \langle x, \tilde{\varphi}_{2k+1} \rangle$ . From (3.60a),

$$\begin{aligned} \alpha_k &= \langle x, \tilde{\varphi}_{2k} \rangle = \langle x_n, \tilde{g}_{2k-n} \rangle_n = \tilde{g}_{n-2k} * x & \leftrightarrow & \alpha = \tilde{\Phi}_g^T x, \\ \beta_k &= \langle x, \tilde{\varphi}_{2k+1} \rangle = \langle x_n, \tilde{h}_{2k-n} \rangle_n = \tilde{h}_{n-2k} * x & \leftrightarrow & \beta = \tilde{\Phi}_h^T x, \end{aligned}$$

that is, we can implement the computation of the expansion coefficients  $\alpha_k$  and  $\beta_k$  using convolutions, exactly as in the orthogonal case. We finally get

$$X = \tilde{\Phi}^T x.$$

From above, we see that the dual basis sequences are

$$\tilde{\Phi} = \{\tilde{\varphi}_k\}_{k \in \mathbb{Z}} = \{\tilde{\varphi}_{2k}, \tilde{\varphi}_{2k+1}\}_{k \in \mathbb{Z}} = \{\tilde{g}_{2k-n}, \tilde{h}_{2k-n}\}_{k \in \mathbb{Z}}, \quad (1.63)$$

that is, the even-indexed dual basis sequences are the shift-reversed impulse responses of the analysis lowpass filter and its even shifts, while the odd-indexed basis sequences are the shift-reversed impulse responses of the analysis highpass filter and its even shifts.

We stress again that the basis sequences of  $\Phi$  are synthesis filters' impulse responses and their even shifts, while the basis sequences of  $\tilde{\Phi}$  are the *shift-reversed* analysis filters' impulse responses and their even shifts. This shift reversal comes from the fact that we are implementing our inner product using a convolution. Note also that  $\Phi$  and  $\tilde{\Phi}$  are completely interchangeable.

As opposed to the three orthonormality relations (1.8), here we have four biorthogonality relations, visualized in Figure 1.11:

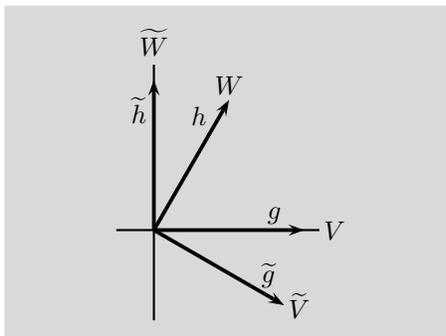
$$\langle g_n, \tilde{g}_{2k-n} \rangle_n = \delta_k, \quad (1.64a)$$

$$\langle h_n, \tilde{h}_{2k-n} \rangle_n = \delta_k, \quad (1.64b)$$

$$\langle h_n, \tilde{g}_{2k-n} \rangle_n = 0, \quad (1.64c)$$

$$\langle g_n, \tilde{h}_{2k-n} \rangle_n = 0. \quad (1.64d)$$

The purpose of this section is to explore the family of impulse responses  $\{g, h\}$  and their duals  $\{\tilde{g}, \tilde{h}\}$  so as to satisfy the biorthogonality constraints. This family is much larger than the orthonormal family, and will contain symmetric/antisymmetric solutions, on which we will focus.



**Figure 1.11:** In a biorthogonal basis,  $\tilde{g}$  is orthogonal to  $h$ , and  $\tilde{h}$  is orthogonal to  $g$ . Then,  $\tilde{g}$  and  $\tilde{h}$  are normalized so that the inner products with their duals are 1.

### 1.4.1 A Single Channel and Its Properties

As we have done for the orthogonal case, we first discuss channels in isolation and determine what they need to satisfy. Figure 1.12 shows the biorthogonal lowpass channel, projecting the input  $x$  onto its lowpass approximation  $x_V$ . That lowpass approximation  $x_V$  can be expressed identically to (1.12a):

$$x_V = \sum_{k \in \mathbb{Z}} \alpha_k g_{n-2k}. \tag{1.65a}$$

The highpass channel follows the lowpass exactly, substituting  $h$  for  $g$ ,  $\tilde{h}$  for  $\tilde{g}$ , and  $x_W$  for  $x_V$  (see Figure 1.12). The highpass approximation  $x_W$  is

$$x_W = \sum_{k \in \mathbb{Z}} \beta_k h_{n-2k}. \tag{1.65b}$$

**Biorthogonality of the Lowpass Filters** Since we started with a pair of biorthogonal bases,  $\{g_{n-2k}, \tilde{g}_{2k-n}\}_{k \in \mathbb{Z}}$  satisfy biorthogonality relations (1.64a). Similarly to the orthogonal case, these can be expressed in various domains as:

$$\langle g_n, \tilde{g}_{2k-n} \rangle_n = \delta_k \quad \begin{array}{l} \xrightarrow{\text{Matrix View}} \\ \xleftarrow{\text{ZT}} \\ \xleftrightarrow{\text{DTFT}} \end{array} \quad \begin{array}{l} D_2 \tilde{G} G U_2 = I \\ G(z) \tilde{G}(z) + G(-z) \tilde{G}(-z) = 2 \\ G(e^{j\omega}) \tilde{G}(e^{j\omega}) + G(e^{j(\omega+\pi)}) \tilde{G}(e^{j(\omega+\pi)}) = 2 \end{array} \tag{1.66}$$

In the matrix view, we have used linear operators (infinite matrices) as we did for the orthogonal case; it expresses the fact that the columns of  $GU_2$  are orthogonal to the rows of  $D_2\tilde{G}$ . The  $z$ -transform expression is often the defining equation of a biorthogonal filter bank, where  $G(z)$  and  $\tilde{G}(z)$  are not causal in general.

**Lowpass Channel in a Two-Channel Biorthogonal Filter Bank****Lowpass filters**

Original domain	$g_n, \tilde{g}_n$	$\langle g_n, \tilde{g}_{2k-n} \rangle_n = \delta_k$
Matrix domain	$G, \tilde{G}$	$D_2 \tilde{G} G U_2 = I$
$z$ domain	$G(z), \tilde{G}(z)$	$G(z)\tilde{G}(z) + G(-z)\tilde{G}(-z) = 2$
DTFT domain	$G(e^{j\omega}), \tilde{G}(e^{j\omega})$	$G(e^{j\omega})\tilde{G}(e^{j\omega}) + G(e^{j(\omega+\pi)})\tilde{G}(e^{j(\omega+\pi)}) = 2$
Polyphase domain	$G(z) = G_0(z^2) + z^{-1}G_1(z^2)$ $\tilde{G}(z) = \tilde{G}_0(z^2) + z\tilde{G}_1(z^2)$	$G_0(z)\tilde{G}_0(z) + G_1(z)\tilde{G}_1(z) = 1$

**Deterministic crosscorrelation**

Original domain	$c_n = \langle g_k, \tilde{g}_{k+n} \rangle_k$	
Matrix domain	$C = \tilde{G}G$	$D_2 C U_2 = I$
$z$ domain	$C(z) = G(z)\tilde{G}(z^{-1})$	$C(z) + C(-z) = 2$
DTFT domain	$C(e^{j\omega}) = G(e^{j\omega})\tilde{G}(e^{j\omega})$	$C(e^{j\omega}) + C(e^{j(\omega+\pi)}) = 2$

**Oblique projection onto smooth space**  $V = \overline{\text{span}}(\{g_{n-2k}\}_{k \in \mathbb{Z}})$   
 $x_V = P_V x$   $P_V = G U_2 D_2 \tilde{G}$

**Table 1.4:** Properties of the lowpass channel in a biorthogonal two-channel filter bank. Properties for the highpass channel are analogous. With  $\tilde{g}_n = g_{-n}$ , or,  $\tilde{G}(z) = G(z^{-1})$  in the  $z$ -transform domain, the relations in this table reduce to those in Table 1.1 for the orthogonal two-channel filter bank.

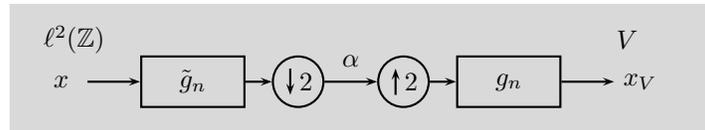
**Biorthogonality of the Highpass Filters**

$$\begin{array}{l}
 \langle h_n, \tilde{h}_{2k-n} \rangle_n = \delta_k \\
 \begin{array}{l}
 \xleftrightarrow{\text{Matrix View}} \\
 \xleftrightarrow{\text{ZT}} \\
 \xleftrightarrow{\text{DTFT}}
 \end{array}
 \end{array}
 \quad
 \begin{array}{l}
 D_2 \tilde{H} H U_2 = I \\
 H(z)\tilde{H}(z) + H(-z)\tilde{H}(-z) = 2 \\
 H(e^{j\omega})\tilde{H}(e^{j\omega}) + H(e^{j(\omega+\pi)})\tilde{H}(e^{j(\omega+\pi)}) = 2
 \end{array}
 \quad (1.67)$$

**Deterministic Crosscorrelation of the Lowpass Filters** In the orthogonal case, we rephrased relations as in (1.66) in terms of the deterministic autocorrelation of  $g$ ; here, as we have two sequences  $g$  and  $\tilde{g}$ , we express it in terms of the deterministic crosscorrelation of  $g$  and  $\tilde{g}$ , (3.99):

$$\begin{array}{l}
 \langle g_n, \tilde{g}_{2k-n} \rangle_n = c_{2k} = \delta_k \\
 \begin{array}{l}
 \xleftrightarrow{\text{Matrix View}} \\
 \xleftrightarrow{\text{ZT}} \\
 \xleftrightarrow{\text{DTFT}}
 \end{array}
 \end{array}
 \quad
 \begin{array}{l}
 D_2 C U_2 = I \\
 C(z) + C(-z) = 2 \\
 C(e^{j\omega}) + C(e^{j(\omega+\pi)}) = 2
 \end{array}
 \quad (1.68)$$

In the above,  $C = \tilde{G}G$  is a Toeplitz matrix with element  $c_{\pm k}$  on the  $k$ th diagonal left/right from the main diagonal (see (2.237)). While this deterministic crosscorrelation will be used for design as in the orthogonal case, unlike in the orthogonal case: (1)  $C(z)$  does not have to be symmetric; (2)  $C(e^{j\omega})$  does not have to be positive; and (3) any factorization of  $C(z)$  leads to a valid solution, that is, the roots of  $C(z)$  can be arbitrarily assigned to  $\tilde{G}(z)$  and  $G(z)$ .



**Figure 1.12:** The biorthogonal lowpass channel.

### Deterministic Crosscorrelation of the Highpass Filters

$$\begin{array}{ccc}
 \langle h_n, \tilde{h}_{2k-n} \rangle_n = c_{2k} = \delta_k & \begin{array}{c} \text{Matrix View} \\ \xleftarrow{\text{ZT}} \\ \xrightarrow{\text{DTFT}} \end{array} & \begin{array}{l} D_2 C U_2 = I \\ C(z) + C(-z) = 2 \\ C(e^{j\omega}) + C(e^{j(\omega+\pi)}) = 2 \end{array}
 \end{array} \quad (1.69)$$

**Projection Property of the Lowpass Channel** We now look at the lowpass channel as a composition of linear operators:

$$x_V = P_V x = G U_2 D_2 \tilde{G} x. \quad (1.70)$$

While  $P_V$  is a projection, it is not an orthogonal projection:

$$\begin{aligned}
 P_V^2 &= (G U_2 \underbrace{D_2 \tilde{G}}_I) (G U_2 D_2 \tilde{G}) = G U_2 D_2 \tilde{G} = P_V, \\
 P_V^T &= (G U_2 D_2 \tilde{G})^T = \tilde{G}^T (U_2 D_2)^T G^T = \tilde{G}^T U_2 D_2 G^T \neq P_V.
 \end{aligned}$$

Indeed,  $P_V$  is a projection operator (it is idempotent), but it is not orthogonal (it is not self-adjoint). Its range is as in the orthogonal case:

$$V = \overline{\text{span}}(\{g_{n-2k}\}_{k \in \mathbb{Z}}). \quad (1.71)$$

Note the interchangeable roles of  $\tilde{g}$  and  $g$ . When  $g$  is used in the synthesis, then  $x_V$  lives in the above span, while if  $\tilde{g}$  is used, it lives in the span of  $\{\tilde{g}_{n-2k}\}_{k \in \mathbb{Z}}$ . The summary of properties of the lowpass channel is given in Table 1.4.

**Projection Property of the Highpass Channel** The highpass projection operator  $P_W$  is:

$$x_W = P_W x = H U_2 D_2 \tilde{H} x; \quad (1.72)$$

again a projection operator (it is idempotent), but not orthogonal (it is not self-adjoint) the same way as for  $P_V$ . Its range is:

$$W = \overline{\text{span}}(\{h_{n-2k}\}_{k \in \mathbb{Z}}). \quad (1.73)$$

### 1.4.2 Complementary Channels and Their Properties

Following the path set during the analysis of orthogonal filter banks, we now discuss what the two channels have to satisfy with respect to each other to build a biorthogonal filter bank. Given a pair of filters  $g$  and  $\tilde{g}$  satisfying (1.66), how can we choose  $h$  and  $\tilde{h}$  to complete the biorthogonal filter bank and thus implement a biorthogonal basis expansion? The sets of basis and dual basis sequences  $\{g_{n-2k}, h_{n-2k}\}_{k \in \mathbb{Z}}$  and  $\{\tilde{g}_{2k-n}, \tilde{h}_{2k-n}\}_{k \in \mathbb{Z}}$  must satisfy (1.64). We have already used (1.64a) in (1.66) and similarly for the highpass sequences in (1.67). What is left to use is that these lowpass and highpass sequences are orthogonal to each other as in (1.64c)–(1.64d):

#### Orthogonality of the Lowpass and Highpass Filters

$$\begin{aligned} \langle h_n, \tilde{g}_{2k-n} \rangle_n = 0 & \quad \begin{array}{c} \text{Matrix View} \\ \longleftrightarrow \\ \xleftarrow{\text{ZT}} \\ \xrightarrow{\text{DTFT}} \\ \longleftrightarrow \end{array} & \quad \begin{array}{l} D_2 \tilde{G} H U_2 = 0 \\ H(z) \tilde{G}(z) + H(-z) \tilde{G}(-z) = 0 \\ H(e^{j\omega}) \tilde{G}(e^{j\omega}) + H(e^{j(\omega+\pi)}) \tilde{G}(e^{j(\omega+\pi)}) = 0 \end{array} \end{aligned} \quad (1.74a)$$

and similarly for  $g$  and  $\tilde{h}$ :

$$\begin{aligned} \langle g_n, \tilde{h}_{2k-n} \rangle_n = 0 & \quad \begin{array}{c} \text{Matrix View} \\ \longleftrightarrow \\ \xleftarrow{\text{ZT}} \\ \xrightarrow{\text{DTFT}} \\ \longleftrightarrow \end{array} & \quad \begin{array}{l} D_2 \tilde{H} G U_2 = 0 \\ G(z) \tilde{H}(z) + G(-z) \tilde{H}(-z) = 0 \\ G(e^{j\omega}) \tilde{H}(e^{j\omega}) + G(e^{j(\omega+\pi)}) \tilde{H}(e^{j(\omega+\pi)}) = 0 \end{array} \end{aligned} \quad (1.74b)$$

### 1.4.3 Biorthogonal Two-Channel Filter Bank

We now pull together what we have developed for biorthogonal filter banks. The following result gives one possible example of a biorthogonal filter bank, inspired by the orthogonal case. We choose the highpass synthesis filter as a modulated version of the lowpass, together with an odd shift. However, because of biorthogonality, it is the analysis lowpass that comes into play.

**THEOREM 1.7 (BIORTHOGONAL TWO-CHANNEL FILTER BANK)** Given are two FIR filters  $g$  and  $\tilde{g}$  of even length  $L = 2\ell$ ,  $\ell \in \mathbb{Z}^+$ , orthogonal to each other and their even shifts as in (1.66). Choose

$$h_n = (-1)^n \tilde{g}_{n-2\ell+1} \quad \xleftarrow{\text{ZT}} \quad H(z) = -z^{-L+1} \tilde{G}(-z) \quad (1.75a)$$

$$\tilde{h}_n = (-1)^n g_{n+2\ell-1} \quad \xleftarrow{\text{ZT}} \quad \tilde{H}(z) = -z^{L-1} G(-z) \quad (1.75b)$$

Then, sets  $\{g_{n-2k}, h_{n-2k}\}_{k \in \mathbb{Z}}$  and  $\{\tilde{g}_{2k-n}, \tilde{h}_{2k-n}\}_{k \in \mathbb{Z}}$  are a pair of biorthogonal bases for  $\ell^2(\mathbb{Z})$ , implemented by a biorthogonal filter bank specified by analysis filters  $\{\tilde{g}, \tilde{h}\}$  and synthesis filters  $\{g, h\}$ .

## 1.4. Biorthogonal Two-Channel Filter Banks

33

*Proof.* To prove the theorem, we must prove that (i)  $\{g_{n-2k}, h_{n-2k}\}_{k \in \mathbb{Z}}$  and  $\{\tilde{g}_{2k-n}, \tilde{h}_{2k-n}\}_{k \in \mathbb{Z}}$  are biorthogonal sets and (ii) they are complete.

- (i) To prove that  $\{g_{n-2k}, h_{n-2k}\}_{k \in \mathbb{Z}}$  and  $\{\tilde{g}_{2k-n}, \tilde{h}_{2k-n}\}_{k \in \mathbb{Z}}$  are biorthogonal sets, we must prove (1.64). The first condition, (1.64a), is satisfied by assumption. To prove the second, (1.64b), that is,  $h$  is orthogonal to  $\tilde{h}$  and its even shifts, we must prove one of the conditions in (1.67). The definitions of  $h$  and  $\tilde{h}$  in (1.75) imply

$$H(z)\tilde{H}(z) = G(-z)\tilde{G}(-z) \quad (1.76)$$

and thus,

$$H(z)\tilde{H}(z) + H(-z)\tilde{H}(-z) = G(-z)\tilde{G}(-z) + G(z)\tilde{G}(z) \stackrel{(a)}{=} 2,$$

where (a) follows from (1.66).

To prove (1.64c)–(1.64d), we must prove one of the conditions in (1.74a)–(1.74b), respectively. We prove (1.64c), (1.64d) follows similarly.

$$\begin{aligned} H(z)\tilde{G}(z) + H(-z)\tilde{G}(-z) &\stackrel{(a)}{=} -z^{L-1}\tilde{G}(-z)\tilde{G}(z) - (-1)^{-L+1}z^{L-1}\tilde{G}(z)\tilde{G}(-z) \\ &\stackrel{(b)}{=} -z^{L-1}G(-z)\tilde{G}(z) + z^{L-1}\tilde{G}(z)G(-z) = 0, \end{aligned}$$

where (a) follows from (1.75a); and (b)  $L = 2\ell$  even.

- (ii) To prove completeness, we prove that perfect reconstruction holds for any  $x \in \ell^2(\mathbb{Z})$ . What we do is find  $z$ -domain expressions for  $X_V(z)$  and  $X_W(z)$  and prove they sum up to  $X(z)$ . We start with the lowpass branch. The proof proceeds as in the orthogonal case.

$$X_V(z) = \frac{1}{2}G(z) \left[ \tilde{G}(z)X(z) + \tilde{G}(-z)X(-z) \right], \quad (1.77a)$$

$$X_W(z) = \frac{1}{2}H(z) \left[ \tilde{H}(z)X(z) + \tilde{H}(-z)X(-z) \right]. \quad (1.77b)$$

The output of the filter bank is the sum of  $x_V$  and  $x_W$ :

$$\begin{aligned} X_V(z) + X_W(z) &= \frac{1}{2} \underbrace{\left[ G(z)\tilde{G}(-z) + H(z)\tilde{H}(-z) \right]}_{S(z)} X(-z) \\ &\quad + \frac{1}{2} \underbrace{\left[ G(z)\tilde{G}(z) + H(z)\tilde{H}(z) \right]}_{T(z)} X(z). \end{aligned} \quad (1.78)$$

Substituting (1.75) into the above equation, we get:

$$\begin{aligned} S(z) &= G(z)\tilde{G}(-z) + H(z)\tilde{H}(-z) \\ &\stackrel{(a)}{=} G(z)\tilde{G}(z) + \left[ -z^{-L+1}\tilde{G}(-z) \right] \left[ -(-z)^{L-1}G(z) \right] \\ &= \left[ 1 + (-1)^{L+1} \right] G(z)\tilde{G}(-z) \stackrel{(b)}{=} 0, \\ T(z) &= G(z)\tilde{G}(z) + H(z)\tilde{H}(z) \\ &\stackrel{(c)}{=} G(z)\tilde{G}(z) + \tilde{G}(-z)G(-z) \stackrel{(d)}{=} 2, \end{aligned}$$

where (a) follows from (1.75); (b) from  $L = 2\ell$  even; (c) from (1.75); and (d) from (1.66). Substituting this back into (1.78), we get

$$X_V(z) + X_W(z) = X(z), \quad (1.79)$$

proving perfect reconstruction, or, in other words, the assertion in the theorem statement that the expansion can be implemented by a biorthogonal filter bank.

Note that we could have also expressed our design problem based on the synthesis (analysis) filters only.

Unlike the orthogonal case, the approximation spaces  $V$  and  $W$  are not orthogonal anymore, and therefore, there exist dual spaces  $\tilde{V}$  and  $\tilde{W}$  spanned by  $\tilde{g}_{-n}$  and  $\tilde{h}_{-n}$  and their even shifts. However,  $V$  is orthogonal to  $\tilde{W}$  and  $W$  is orthogonal to  $\tilde{V}$ . This was schematically shown in Figure 1.11. Table 1.10 summarizes various properties of biorthogonal, two-channel filter banks we covered until now.

#### 1.4.4 Polyphase View of Biorthogonal Filter Banks

We have already seen how polyphase analysis of orthogonal filter banks adds to the analysis toolbox. We now give a brief account of important polyphase notions when dealing with biorthogonal filter banks. First, recall from (1.33) that the polyphase matrix of the synthesis bank is given by<sup>7</sup>

$$\Phi_p(z) = \begin{bmatrix} G_0(z) & H_0(z) \\ G_1(z) & H_1(z) \end{bmatrix}, \quad \begin{aligned} G(z) &= G_0(z) + z^{-1}G_1(z), \\ H(z) &= H_0(z) + z^{-1}H_1(z). \end{aligned} \quad (1.80a)$$

By the same token, the polyphase matrix of the analysis bank is given by

$$\tilde{\Phi}_p(z) = \begin{bmatrix} \tilde{G}_0(z) & \tilde{H}_0(z) \\ \tilde{G}_1(z) & \tilde{H}_1(z) \end{bmatrix}, \quad \begin{aligned} \tilde{G}(z) &= \tilde{G}_0(z) + z\tilde{G}_1(z), \\ \tilde{H}(z) &= \tilde{H}_0(z) + z\tilde{H}_1(z). \end{aligned} \quad (1.80b)$$

Remember that the different polyphase decompositions of the analysis and synthesis filters are a matter of a carefully chosen convention.

For a biorthogonal filter bank to implement a biorthogonal expansion, the following must be satisfied:

$$\Phi_p(z) \tilde{\Phi}_p^T(z) = I. \quad (1.81)$$

From this, **[NOTE: Requires scrutiny. Possibly needs transposition.]**

$$\tilde{\Phi}_p(z) = (\Phi_p^T(z))^{-1} = \frac{1}{\det \Phi_p(z)} \begin{bmatrix} H_1(z) & -H_0(z) \\ -G_1(z) & G_0(z) \end{bmatrix}. \quad (1.82)$$

Since all the matrix entries are FIR, for the analysis to be FIR as well,  $\det \Phi_p(z)$  must be a monomial, that is:

$$\det \Phi_p(z) = G_0(z)H_1(z) - G_1(z)H_0(z) = z^{-k}. \quad (1.83)$$

<sup>7</sup>When we say *polyphase matrix*, we will mean the polyphase matrix of the synthesis bank; for the analysis bank, we will explicitly state *analysis polyphase matrix*.

In the above, we have implicitly assumed that  $\Phi_p(z)$  was invertible, that is, its columns are linearly independent. This can be rephrased in filter bank terms by stating when, given  $G(z)$ , it is possible to find  $H(z)$  such that it leads to a perfect reconstruction biorthogonal filter bank. Such a filter  $H(z)$  will be called a *complementary filter*.

**THEOREM 1.8 (COMPLEMENTARY FILTERS)** Given a causal FIR filter  $G(z)$ , there exists a complementary FIR filter  $H(z)$ , if and only if the polyphase components of  $G(z)$  are coprime (except for possible zeros at  $z = \infty$ ).

*Proof.* We just saw that a necessary and sufficient condition for perfect FIR reconstruction is that  $\det(\Phi_p(z))$  be a monomial. Thus, coprimeness is obviously necessary, since if there were a common factor between  $G_0(z)$  and  $G_1(z)$ , it would show up in the determinant.

Sufficiency follows from the Bézout's identity (3.290) that says that given two coprime polynomials  $a(z)$  and  $b(z)$ , the equation  $a(z)p(z) + b(z)q(z) = c(z)$  has a solution  $p(z)$ ,  $q(z)$ . Fixing  $a(z) = G_0(z)$ ,  $b(z) = G_1(z)$  and  $c(z) = z^{-k}$ , we see that Bézout's identity is equal to (1.83), and thus guarantees a solution  $p(z) = H_0(z)$  and  $q(z) = H_1(z)$ , that is, a complementary filter  $H(z)$ .

Note that the coprimeness of  $G_0(z)$  and  $G_1(z)$  is equivalent to  $G(z)$  not having any zero pairs  $\{z_0, -z_0\}$ . This can be used to prove that the binomial filter  $G(z) = (1 + z^{-1})^N$  always has a complementary filter (see Exercise ??).

The counterpart to Theorem 1.3 and Corollary 1.4 for orthogonal filter banks are the following theorem and corollary for the biorthogonal ones (we state these without proof):

**THEOREM 1.9 (POSITIVE DEFINITE MATRIX AND BIORTHOGONAL BASIS)** Given a filter bank implementing a biorthogonal basis for  $\ell^2(\mathbb{Z})$  and its associated polyphase matrix  $\Phi_p(e^{j\omega})$ , then  $\Phi_p(e^{j\omega})\Phi_p^T(e^{-j\omega})$  is positive definite.

**COROLLARY 1.10 (FILTERED DETERMINISTIC AUTOCORRELATION MATRIX IS POSITIVE SEMIDEFINITE)** Given is a  $2 \times 2$  polyphase matrix  $\Phi_p(e^{j\omega})$  such that  $\Phi_p(e^{j\omega})\Phi_p^T(e^{-j\omega})$ . Then the filtered deterministic autocorrelation matrix,  $A_{p,\alpha}(e^{j\omega})$ , is positive semidefinite.

### 1.4.5 Linear-Phase Two-Channel Filter Banks

We started this section by saying that one of the reasons we go through the trouble of analyzing and constructing two-channel biorthogonal filter banks is because they

allow us to obtain real-coefficient FIR filters with linear phase.<sup>8</sup> Thus, we now do just that: we build perfect reconstruction filter banks where all the filters involved are linear phase. Linear-phase filters were defined in (3.107).

As was true for orthogonal filters, not all lengths of filters are possible if we want to have a linear-phase filter bank. This is summarized in the following theorem, the proof of which is left as Exercise ??:

**THEOREM 1.11** In a two-channel, perfect reconstruction filter bank where all filters are linear phase, the synthesis filters have one of the following forms:

- (i) Both filters are odd-length symmetric, the lengths differing by an odd multiple of 2.
- (ii) One filter is symmetric and the other is antisymmetric; both lengths are even, and are equal or differ by an even multiple of 2.
- (iii) One filter is of odd length, the other one of even length; both have all zeros on the unit circle. Either both filters are symmetric, or one is symmetric and the other one is antisymmetric.

Our next task is to show that indeed, it is not possible to have an orthogonal filter bank with linear-phase filters if we restrict ourselves to the two-channel, FIR, real-coefficient case:

**THEOREM 1.12** The only two-channel perfect reconstruction orthogonal filter bank with real-coefficient FIR linear-phase filters is the Haar filter bank.

*Proof.* In orthogonal filter banks, (1.40)–(1.41) hold, and the filters are of even length. Therefore, following Theorem 1.11, one filter is symmetric and the other antisymmetric. Take the symmetric one,  $G(z)$  for example,

$$\begin{aligned} G(z) &\stackrel{(a)}{=} G_0(z^2) + z^{-1}G_1(z^2) \\ &\stackrel{(b)}{=} z^{-L+1}G(z^{-1}) \stackrel{(c)}{=} z^{-L+1}(G_0(z^{-2}) + zG_1(z^{-2})) \\ &= z^{-L+2}G_1(z^{-2}) + z^{-1}(z^{-L+2}G_0(z^{-2})), \end{aligned}$$

where (a) and (c) follow from (1.32), and (b) from (3.153). This further means that for the polyphase components, the following hold:

$$G_0(z) = z^{-L/2+1}G_1(z^{-1}), \quad G_1(z) = z^{-L/2+1}G_0(z^{-1}). \quad (1.84)$$

Substituting (1.84) into (1.40) we obtain

$$G_0(z) G_0(z^{-1}) = \frac{1}{2}.$$

<sup>8</sup>If we allow filters to have complex-valued coefficients or if we lift the restriction of two channels, linear phase and orthogonality can be satisfied simultaneously.

The only FIR, real-coefficient polynomial satisfying the above is

$$G_0(z) = \frac{1}{\sqrt{2}}z^{-m}.$$

Performing a similar analysis for  $G_1(z)$ , we get that  $G_1(z) = (1/\sqrt{2})z^{-k}$ , and

$$G(z) = \frac{1}{\sqrt{2}}(z^{-2\ell} + z^{-2k-1}), \quad H(z) = G(-z),$$

yielding Haar filters ( $m = k = 0$ ) or trivial variations thereof.

While the outstanding features of the Haar filters make it a very special solution, Theorem 1.12 is a fundamentally negative result as the Haar filters have poor frequency localization and no polynomial reproduction capability.

## 1.5 Design of Biorthogonal Two-Channel Filter Banks

Given that biorthogonal filters are less constrained than their orthogonal counterparts, the design space is much more open. In both cases, one factors a Laurent polynomial<sup>9</sup>  $C(z)$  satisfying  $C(z) + C(-z) = 2$  as in (1.68). In the orthogonal case,  $C(z)$  was a deterministic autocorrelation, while in the biorthogonal case, it is a deterministic crosscorrelation and thus more general. In addition, the orthogonal case requires spectral factorization (square root), while in the biorthogonal case, any factorization will do. While the factorization method is not the only approach, it is the most common. Other approaches include the complementary filter design method and the lifting design method. In the former, a desired filter is complemented so as to obtain a perfect reconstruction filter bank. In the latter, a structure akin to a lattice is used to guarantee perfect reconstruction as well as other desirable properties.

### 1.5.1 Factorization Design

From (1.66)–(1.68),  $C(z)$  satisfying  $C(z) + C(-z) = 2$  can be factored into

$$C(z) = G(z)\tilde{G}(z),$$

where  $G(z)$  is the synthesis and  $\tilde{G}(z)$  the analysis lowpass filter (or vice-versa, since the roles are dual). The most common designs use the same  $C(z)$  as those used in orthogonal filter banks, for example, those with a maximum number of zeros at  $z = -1$ , performing the factorization so that the resulting filters have linear phase.

**EXAMPLE 1.4 (BIORTHOGONAL FILTER BANK WITH LINEAR-PHASE FILTERS)** We reconsider Example 1.3, in particular  $C(z)$  given by

$$C(z) = (1 + z^{-1})^2 (1 + z)^2 \frac{1}{4} \left( -\frac{1}{4}z^{-1} + 1 - \frac{1}{4}z \right),$$

<sup>9</sup>A Laurent polynomial is a polynomial with both positive and negative powers, see *Appendix 3.B.1*.

which satisfies  $C(z) + C(-z) = 2$  by construction. This also means it satisfies (1.66) for any factorization of  $C(z)$  into  $\tilde{G}(z)G(z)$ . Note that we can add factors  $z$  or  $z^{-1}$  in one filter, as long as we cancel it in the other; this is useful for obtaining purely causal/anticausal solutions.

One possible factorization is

$$G(z) = z^{-1} (1 + z^{-1})^2 (1 + z) = (1 + z^{-1})^3 = 1 + 3z^{-1} + 3z^{-2} + z^{-3},$$

$$\tilde{G}(z) = z(1 + z) \frac{1}{4} \left( -\frac{1}{4}z^{-1} + 1 - \frac{1}{4}z \right) = \frac{1}{16} (-1 + 3z + 3z^2 - z^3).$$

The other filters follow from (1.75), with  $L = 2\ell = 4$ :

$$H(z) = -z^{-3} \frac{1}{16} (-1 - 3z + 3z^2 + z^3) = \frac{1}{16} (-1 - 3z^{-1} + 3z^{-2} + z^{-3}),$$

$$\tilde{H}(z) = -z^3 (1 - 3z^{-1} + 3z^{-2} - z^{-3}) = 1 - 3z + 3z^2 - z^3.$$

The lowpass filters are both symmetric, while the highpass ones are antisymmetric. As  $\tilde{H}(z)$  has three zero moments,  $G(z)$  can reproduce polynomials up to degree 2, since such sequences go through the lowpass channel only.

Another possible factorization is

$$G(z) = (1 + z^{-1})^2 (1 + z)^2 = z^{-2} + 4z^{-1} + 6 + 4z + z^2,$$

$$\tilde{G}(z) = \frac{1}{4} \left( -\frac{1}{4}z^{-1} + 1 - \frac{1}{4}z \right) = \frac{1}{16} (-z^{-1} + 4 - z),$$

where both lowpass filters are symmetric and zero phase. The highpass filters are (with  $L = 0$ ):

$$H(z) = -\frac{1}{16} z (z^{-1} + 4 + z),$$

$$\tilde{H}(z) = -z^{-1} (z^{-2} - 4z^{-1} + 6 - 4z + z^2),$$

which are also symmetric, but with a phase delay of  $\pm 1$  sample.

The zeros at  $z = -1$  in the synthesis lowpass filter become, following (1.75b), zeros at  $z = 1$  in the analysis highpass filter. Therefore, many popular biorthogonal filters come from symmetric factorizations of  $C(z)$  with a maximum number of zeros at  $z = -1$ .

**EXAMPLE 1.5 (DESIGN OF THE 9/7 FILTER PAIR)** The next higher-degree  $C(z)$  with a maximum number of zeros at  $z = -1$  is of the form

$$C(z) = 2^{-8} (1 + z)^3 (1 + z^{-1})^3 (3z^2 - 18z + 38 - 18z^{-1} + 3z^{-2}).$$

One possible factorization yields the so-called *Daubechies 9/7* filter pair (see Table 1.5). These filters have odd length and even symmetry, and are part of the JPEG 2000 image compression standard.

$n$	Daubechies 9/7		LeGall 5/3	
	$\tilde{g}_n$	$g_n$	$\tilde{g}_n$	$g_n$
0	0.60294901823635790	1.11508705245699400	3/4	1
$\pm 1$	0.26686411844287230	0.59127176311424700	1/4	1/2
$\pm 2$	-0.07822326652898785	-0.05754352622849957	-1/8	
$\pm 3$	-0.01686411844287495	-0.09127176311424948		
$\pm 4$	0.02674875741080976			

**Table 1.5:** Biorthogonal filters used in the still-image compression standard JPEG 2000. The lowpass filters are given; the highpass filters can be derived using (1.75a)–(1.75b). The first pair is from [5] and the second from [57].

### 1.5.2 Complementary Filter Design

Assume we have a desired synthesis lowpass filter  $G(z)$ . How can we find  $\tilde{G}(z)$  such that we obtain a perfect reconstruction biorthogonal filter bank? It suffices to find  $\tilde{G}(z)$  so that (1.66) is satisfied, which, according to Theorem 1.8, can always be done if  $G(z)$  has coprime polyphase components. Then  $\tilde{G}(z)$  can be found by solving a linear system of equations.

EXAMPLE 1.6 (COMPLEMENTARY FILTER DESIGN) Suppose

$$G(z) = \frac{1}{2}z + 1 + \frac{1}{2}z^{-1} = \frac{1}{2}(1+z)(1+z^{-1}).$$

We would like to find  $\tilde{G}(z)$  such that  $C(z) = G(z)\tilde{G}(z)$  satisfies  $C(z) + C(-z) = 2$ . (It is easy to verify that the polyphase components of  $G(z)$  are coprime, so such a  $\tilde{G}(z)$  should exist. We exclude the trivial solution  $\tilde{G}(z) = 1$ ; it is of no interest as it has no frequency selectivity.) For a length-5 symmetric filter  $\tilde{G}(z) = cz^2 + bz + a + bz^{-1} + cz^{-2}$ , we get the following system of equations:

$$a + b = 1 \quad \text{and} \quad \frac{1}{2}b + c = 0.$$

To get a unique solution, we could, for example, impose that the filter have a zero at  $z = -1$ ,

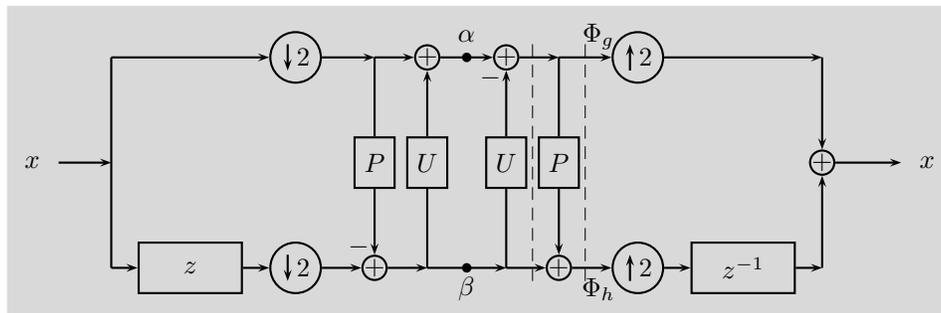
$$a - 2b + 2c = 0,$$

leading to  $a = 6/8$ ,  $b = 2/8$ , and  $c = -1/8$ :

$$\tilde{G}(z) = \frac{1}{8}(-z^2 + 2z + 6 + 2z^{-1} - z^{-2}).$$

All coefficients of  $(g, \tilde{g})$  are integer multiples of  $1/8$ , making the analysis and synthesis exactly invertible even with finite-precision (binary) arithmetic. These filters are used in the JPEG 2000 image compression standard; see Table 1.5.

As can be seen from this example, the solution for the complementary filter is highly nonunique. Not only are there solutions of different lengths (in the case above, any



**Figure 1.13:** The lifting filter bank, with  $P$  and  $U$  predict and update operators, respectively.

length  $3 + 4m$ ,  $m \in \mathbb{N}$ , is possible), but even a given length has multiple solutions. It can be shown that this variety is given by the solutions of a Diophantine equation related to the polyphase components of the filter  $G(z)$ .

### 1.5.3 Lifting Design

We conclude this section with the design procedure based on lifting. While the original idea behind lifting was to build shift-varying perfect reconstruction filter banks, it has also become popular as it allows for building discrete-time bases with non-linear operations. The trivial filter bank to start lifting is the polyphase transform which splits the sequence into even- and odd-indexed components as in Figure 1.13. In the first lifting step, we use a prediction filter  $P$  to predict the odd coefficients from the even ones. The even coefficients remain unchanged, while the result of the prediction filter applied to the even coefficients is subtracted from the odd coefficients yielding the highpass coefficients. In the second step, we use an update filter  $U$  to update the even coefficients based on the previously computed highpass coefficients. We start with a simple example.

**EXAMPLE 1.7 (HAAR FILTER BANK OBTAINED BY LIFTING)** The two polyphase components of  $x$  are  $x_0$  (even subsequence) and  $x_1$  (odd subsequence) as in (3.216). The purpose of the prediction operator  $P$  is to predict odd coefficients based on the even ones. The simplest prediction says that the odd coefficients are exactly the same as the even ones, that is  $p_n = \delta_n$ . The output of the highpass branch is thus the difference  $(\delta_n - \delta_{n-1})$ , a reasonable outcome. The purpose of the update operator  $U$  is to then update the even coefficients based on the newly computed odd ones. As we are looking for a lowpass-like version in the other branch, the easiest is to subtract half of this difference from the even sequence, leading to  $x_{0,n} - (x_{1,n} - x_{0,n})/2$ , that is, the average  $(x_{0,n} + x_{1,n})/2$ , again a reasonable output, but this time lowpass in nature. Within scaling, it is thus clear that the choice  $p_n = \delta_n$ ,  $u_n = (1/2)\delta_n$  leads to the Haar filter bank.

Let us now identify the polyphase matrix  $\Phi_p(z)$ :

$$\begin{aligned}\Phi_g(z) &= \alpha(z) - U(z)\beta(z), \\ \Phi_h(z) &= \beta(z) + P(z)\Phi_g(z) \\ &= \beta(z) + P(z)(\alpha(z) - U(z)\beta(z)) \\ &= P(z)\alpha(z) + (1 - P(z)U(z))\beta(z),\end{aligned}$$

which we can write as

$$\begin{bmatrix} \Phi_g(z) \\ \Phi_h(z) \end{bmatrix} = \begin{bmatrix} 1 & -U(z) \\ P(z) & 1 - P(z)U(z) \end{bmatrix} \begin{bmatrix} \alpha(z) \\ \beta(z) \end{bmatrix} = \Phi_p(z) \begin{bmatrix} \alpha(z) \\ \beta(z) \end{bmatrix}. \quad (1.85)$$

On the analysis side,  $\tilde{\Phi}_p(z)$  is:

$$\tilde{\Phi}_p(z) = (\Phi_p^T(z))^{-1} = \begin{bmatrix} 1 - P(z)U(z) & -P(z) \\ U(z) & 1 \end{bmatrix}. \quad (1.86)$$

As the  $\det(\Phi_p(z)) = 1$ , the inverse of  $\Phi_p(z)$  does not involve actual inversion, one of the reasons why this technique is popular. Moreover, we can write  $\Phi_p$  as

$$\Phi_p(z) = \begin{bmatrix} 1 & -U(z) \\ P(z) & 1 - P(z)U(z) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ P(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & -U(z) \\ 0 & 1 \end{bmatrix}, \quad (1.87)$$

decomposing  $\Phi_p(z)$  into a sequence of lower/upper triangular matrices—*lifting steps*. What we also see is that the inverse of each matrix of the form:

$$\begin{bmatrix} 1 & 0 \\ M & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ -M & 1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 & M \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & -M \\ 0 & 1 \end{bmatrix},$$

meaning to invert these one needs only reverse the sequence of operations as shown in Figure 1.13. This is why this scheme allows for nonlinear operations; if  $M$  is nonlinear, its inversion amounts to simply reversing the sign in the matrix.

## 1.6 Two-Channel Filter Banks with Stochastic Inputs

Our discussion so far assumed we are dealing with deterministic sequences as inputs into our filter bank, most often those with finite energy. If the input into our filter bank is stochastic, then we must use the tools developed in *Section 3.8*. The periodic shift variance for deterministic systems has its counterpart in wide-sense cyclostationarity. The notions of energy spectral density (3.96) (DTFT of the deterministic autocorrelation) and energy (3.98) have their counterparts in the notions of power spectral density (3.239) (DTFT of the stochastic autocorrelation) and power (3.240). We now briefly discuss the effects of a filter bank on an input WSS sequence.

Until now, we have seen various ways of characterizing systems with deterministic and stochastic inputs, among others via the deterministic and stochastic autocorrelations.

In a single-input single-output system:

- (i) For a deterministic sequence, its autocorrelation is Hermitian symmetric (see (3.16)) and can be factored as in (3.96), (3.143), that is, it is nonnegative on the unit circle. It is sometimes called energy spectral density.
- (ii) For a WSS sequence, the counterpart to the deterministic autocorrelation is the power spectral density given in (3.239).

In a multiple-input multiple-output system, such as a filter banks, where the multiple inputs are naturally polyphase components of the input sequence:

- (i) For a deterministic sequence, we have a matrix autocorrelation of the vector of polyphase components  $[x_0 \ x_1]^T$ , given by (3.219). In particular, we have seen it for a the vector of expansion coefficient sequences  $[\alpha \ \beta]^T$  in two-channel filter bank earlier in this chapter, in (1.43).
- (ii) For a WSS sequence  $x$ , we can also look at the matrix of power spectral densities of the polyphase components  $[x_0 \ x_1]^T$  as in (3.247). In what follows, we analyze that matrix for the vector of expansion coefficient sequences  $[\alpha \ \beta]^T$ .

**Filter-Bank Optimization Based on Input Statistics** The area of optimizing filter banks based on input statistics is an active one. In particular, principal-component filter banks have been shown to be optimal for a wide variety of problems (we give pointers to the literature on the subject in *Further Reading*). For example, in parallel to our discussion in *Chapter 6* on the use of KLT, it is known that the coding gain is maximized if the channel sequences are decorrelated, ( $\tilde{\Phi}_a$  is diagonal), and  $A_\alpha(e^{j\omega}) \geq A_\beta(e^{j\omega})$  if  $\text{var}(\alpha_n) \geq \text{var}(\beta_n)$ . We can diagonalize  $\tilde{\Phi}_a$  by factoring it as  $\tilde{\Phi}_a = QAQ^*$ , where  $Q$  is the matrix of eigenvectors of  $\tilde{\Phi}_a$ .

## 1.7 Computational Aspects

The power of filter banks is that they are a computational tool; they implement a wide variety of bases (and frames, see Chapter 4). As the two-channel filter bank is the basic building block for many of these, we now spend some time discussing various computational concerns that arise in applications.

### 1.7.1 Two-Channel Filter Banks

We start with a two-channel filter bank with synthesis filters  $\{g, h\}$  and analysis filters  $\{\tilde{g}, \tilde{h}\}$ . For simplicity and comparison purposes, we assume that the input is of even length  $M$ , filters are of even length  $L$ , and all costs are computed per input sample. From (1.80b), the channel signals  $\alpha$  and  $\beta$  are

$$\alpha = \tilde{g}_0 * x_0 + \tilde{g}_1 * x_1, \quad (1.88a)$$

$$\beta = \tilde{h}_0 * x_0 + \tilde{h}_1 * x_1, \quad (1.88b)$$

where  $\tilde{g}_{0,1}, \tilde{h}_{0,1}$  are the polyphase components of the analysis filters  $\tilde{g}$  and  $\tilde{h}$ . We have immediately written the expression in polyphase domain, as it is implicitly

clear that it does not make sense to do the filtering first and then discard every other product (see *Section 3.9.3*).

In general, (1.88) amounts to four convolutions with polyphase components  $x_0$  and  $x_1$ , each of half the original length, plus the necessary additions. Instead of using (3.275a), we compute directly the cost per input sample. The four convolutions operate at half the input rate and thus, for every two input samples, we compute  $4L/2$  multiplications and  $4((L/2) - 1) + 2$  additions. This leads to  $L$  multiplications and  $L - 1$  additions/input sample, that is, exactly the same complexity as a convolution by a single filter of size  $L$ . The cost is thus

$$C_{\text{biorth,time}} = 2L - 1 \quad \sim \quad O(L), \quad (1.89)$$

per input sample.

If an FFT-based convolution algorithm is used, for example, overlap-add, we need four convolutions using DFTs of length  $N$  as in (3.274), plus  $2N$  additions. Assume for simplicity and comparison purposes that  $M = L = N2$ .

$$C_{\text{biorth,freq}} = 16\alpha \log_2 L + 14 \quad \sim \quad O(\log_2 L), \quad (1.90)$$

per input sample.

In [73], a precise analysis is made involving FFTs with optimized lengths so as to minimize the operation count. Using the split-radix FFT algorithm, the number of operations becomes (for large  $L$ )

$$C_{\text{biorth,freq,optim}} = 4 \log_2 L + O(\log_2 \log_2 L),$$

again per input sample. Comparing this to  $C_{\text{biorth,freq}}$  (and disregarding the constant  $\alpha$ ), the algorithm starts to be effective for  $L = 8$  and a length-16 FFT, where it achieves around 5 multiplications per sample rather than 8, and leads to improvements of an order of magnitude for large filters (such as  $L = 64$  or 128). For medium-size filters ( $L = 6, \dots, 12$ ), a method based on fast running convolution is best (see [73]).

Let us now consider some special cases where additional savings are possible.

**Linear-Phase Filter Banks** It is well-known that if a filter is symmetric or anti-symmetric, the number of operations can be halved in (1.89) by simply adding (or subtracting) the two input samples that are multiplied by the same coefficient. This trick can be used in the downsampled case as well, that is, filter banks with linear-phase filters require half the number of multiplications, or  $L/2$  multiplications per input sample (the number of additions remains unchanged), for a total cost of

$$C_{\text{lp,direct}} = \frac{3}{2}L - 1 \quad \sim \quad O(L), \quad (1.91)$$

still  $O(L)$  but with a savings of roughly 25% over (1.89). If the filter length is odd, the polyphase components are themselves symmetric or antisymmetric, and the saving is obvious in (1.88).

Another option is to use a linear-phase lattice factorization:

$$\Phi_p(z) = \alpha \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \prod_{i=1}^{N/2-1} \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \begin{bmatrix} 1 & \alpha_i \\ \alpha_i & 1 \end{bmatrix}.$$

The individual  $2 \times 2$  symmetric matrices can be written as (we assume  $\alpha_i \neq 1$ )

$$\begin{bmatrix} 1 & \alpha_i \\ \alpha_i & 1 \end{bmatrix} = \frac{1 - \alpha_i}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \frac{1+\alpha_i}{1-\alpha_i} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

By gathering the scale factors together, we see that each new block in the cascade structure (which increases the length of the filters by two) adds only one multiplication. Thus, we need  $L/4$  multiplications, and  $(L-1)$  additions per input sample, for a total cost of

$$C_{\text{lp,lattice}} = \frac{5}{4}L - 1 \quad \sim \quad O(L), \quad (1.92)$$

per input sample. The savings is roughly 16% over (1.91), and 37.5% over (1.89).

Two-Channel Filter Bank	$\mu$	$\nu$	Cost	Order
Biorthogonal				
Frequency	$16\alpha \log_2 L$	14	$16\alpha \log_2 L + 14$	$O(\log_2 L)$
Time	$L$	$L - 1$	$2L - 1$	$O(L)$
Linear phase				
Direct form	$(1/2)L$	$L - 1$	$(3/2)L - 1$	$O(L)$
Lattice form	$(1/4)L$	$L - 1$	$(5/4)L - 1$	$O(L)$
Orthogonal				
Lattice form	$(3/4)L$	$(3/4)L$	$(3/2)L$	$O(L)$
Denormalized lattice	$(1/2)L + 1$	$(3/4)L$	$(5/4)L + 1$	$O(L)$
QMF	$(1/2)L$	$(1/2)L$	$L$	$O(L)$

**Table 1.6:** Cost per input sample of computing various two-channel filter banks with length- $L$  filters.

**Orthogonal Filter Banks** As we have seen, there exists a general form for a two-channel paraunitary matrix, given in (1.39). If  $G_0(z)$  and  $G_1(z)$  were of degree zero, it is clear that the matrix in (1.39) would be a rotation matrix, which can be implemented with three multiplications, as we will show shortly. It turns out that for arbitrary-degree polyphase components, terms can still be gathered into rotations, saving 25% of multiplications (at the cost of 25% more additions). This rotation property is more obvious in the lattice structure form of orthogonal filter banks (1.54), where matrices  $R_k$  can be written as:

$$\begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta_i - \sin \theta_i & 0 & 0 \\ 0 & \cos \theta_i + \sin \theta_i & 0 \\ 0 & 0 & \sin \theta_i \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & -1 \end{bmatrix}.$$

Thus, only three multiplications are needed, or  $3L/2$  for the whole lattice. Since the lattice works in the downsampled domain, the cost is  $3L/4$  multiplications per input sample and a similar number of additions, for a total cost of

$$C_{\text{orth,lattice}} = \frac{3}{2}L \quad \sim \quad O(L), \quad (1.93)$$

per input sample. We could also denormalize the diagonal matrix in the above equation (taking out  $\sin \theta_i$  for example) and gather all scale factors at the end of the lattice, leading to  $(L/2 + 1)$  multiplications per input sample, and the same number of additions as before, for a total cost of

$$C_{\text{orth,lattice,denorm}} = \frac{5}{4}L + 1 \quad \sim \quad O(L), \quad (1.94)$$

per input sample.

**QMF Filter Banks** The classic QMF solution discussed in Exercise ??, besides using even-length linear phase filters, forces the highpass filter to be equal to the lowpass, modulated by  $(-1)^n$ . The polyphase matrix is therefore:

$$\Phi_p(z) = \begin{bmatrix} G_0(z) & G_1(z) \\ G_0(z) & -G_1(z) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} G_0(z) & 0 \\ 0 & G_1(z) \end{bmatrix},$$

where  $G_0$  and  $G_1$  are the polyphase components of  $G(z)$ . The factorized form on the right indicates that the cost is halved. However, this scheme only approximates a basis expansion (perfect reconstruction) when using FIR filters. Table 1.6 summarizes the costs of various filter banks we have seen so far.

**Multidimensional Filter Banks** While we have not discussed multidimensional filter banks so far (some pointers are given in *Further Reading*), we do touch upon the cost of computing them. For example, filtering an  $M \times M$  image with a filter of length  $L \times L$  requires of the order of  $O(M^2 L^2)$  operations. If the filter is separable, that is,  $G(z_1, z_2) = G_1(z_1)G_2(z_2)$ , then filtering on rows and columns can be done separately and the cost is reduced to an order  $O(2M^2 L)$  operations ( $M$  row filterings and  $M$  column filterings, each using  $ML$  operations).

A multidimensional filter bank can be implemented in its polyphase form, bringing the cost down to the order of a single nondownsampling convolution, just as in the one-dimensional case. A few cases of particular interest allow further reductions in cost. For example, when both filters and downsampling are separable, the system is the direct product of one-dimensional systems, and the implementation is done separately over each dimension. Consider a two-dimensional system filtering an  $M \times M$  image into four subbands using the filters  $\{G(z_1)G(z_2), G(z_1)H(z_2), H(z_1)G(z_2), H(z_1)H(z_2)\}$  each of length  $M \times M$  followed by separable downsampling by two in each dimension. This requires  $M$  decompositions in one dimension (one for each row), followed by  $M$  decompositions in the other, for a total of  $O(2M^2 L)$  multiplications and a similar number of additions. This is a saving of the order of  $L/2$  with respect to the nonseparable case.

## 1.7.2 Boundary Extensions

While most of the literature as well as our exposition implicitly assume infinite-length sequences, in practice this is not the case. Given an  $N \times N$  image, for example, the result of processing it should be another image of the same size. In *Chapter 3*, we discussed the finite-length case by introducing periodic (circular) extension, when the appropriate convolution is the circular convolution and the appropriate Fourier transform is the DFT. In practice, however, periodic extension is rather artificial as it wraps the sequence around (for example, what is on the left boundary of the image would appear on the right boundary). Other extensions are possible, and while for some of them (for example, symmetric), appropriate notions of convolution and Fourier transform are available, in practice this is not done. Instead, different types of extensions are applied (zero-padding, symmetric, continuous, smooth) while still using the tools developed for the periodic extension. Throughout this subsection, we assume a sequence of length  $N$ ; also, we will be using the extension nomenclature adopted in Matlab, and will point out other names under which these extensions are known.

**Periodic Extension** From  $x$ , create a periodic  $y$  as

$$y_n = x_{n \bmod N}.$$

Of those we consider here, this is the only mathematically correct extension in conjunction with the DFT. Moreover, it is simple and works for any sequence length. The drawback is that the underlying sequence is most likely not periodic, and thus, periodization creates artificial discontinuities at multiples of  $N$ ; see Figure 1.14(b).<sup>10</sup>

**Zero-Padding Extension** From  $x$ , create  $y$  as

$$y_n = \begin{cases} x_n, & n = 0, 1, \dots, N-1; \\ 0, & \text{otherwise.} \end{cases}$$

Again, this extension is simple and works for any sequence length. However, it too creates artificial discontinuities as in Figure 1.14(c). Also, during the filtering process, the sequence is extended by the length of the filter (minus 1), which is often undesirable.

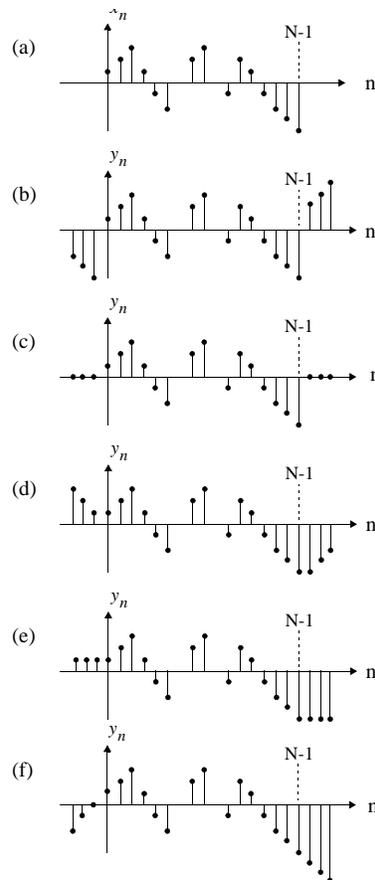
**Symmetric Extension** From  $x$ , create a double-length  $y$  as

$$y_n = \begin{cases} x_n, & n = 0, 1, \dots, N-1; \\ x_{2N-n-1}, & n = N, N+1, \dots, 2N-1, \end{cases}$$

and then periodize it. As shown in Figure 1.14(d), this periodic sequence of period  $2N$  does not show the artificial discontinuities of the previous two cases.<sup>11</sup> However,

<sup>10</sup>Technically speaking, a discrete sequence cannot be continuous or discontinuous. However, if the sequence is a densely sampled version of a smooth sequence, periodization will destroy this smoothness.

<sup>11</sup>It does remain discontinuous in its derivatives however; for example, if it is linear, it will be smooth but not differentiable at 0 and  $N$ .



**Figure 1.14:** Boundary extensions. (a) Original sequence  $x$  of length  $N$ . (b) Periodic extension:  $x$  is repeated with a period  $N$ . (c) Zero-padding extension: Beyond the support,  $y$  is set to zero. (d) Symmetric extension: The sequence is flipped at the boundaries to preserve continuity. (Half-point symmetry is shown.) (e) Continuous extension: The boundary value is replicated. (f) Smooth extension: At the boundary, a polynomial extension is applied to preserve higher-order continuity.

the sequence is now twice as long, and unless carefully treated, this redundancy is hard to undo. Cases where it can be handled easily are when the filters are symmetric or antisymmetric, because the output of the filtering will be symmetric or antisymmetric as well.

There exist two versions of the symmetric extension, depending on whether whole- or half-point symmetry is used. The formulation above is called *half-point symmetric* because  $y$  is symmetric about the half-integer index value  $N - \frac{1}{2}$ . An

alternative is *whole-point symmetric*  $y$

$$y_n = \begin{cases} x_n, & n = 0, 1, \dots, N-1; \\ x_{2N-n-2}, & n = N, N+1, \dots, 2N-2, \end{cases}$$

with even symmetry around  $N$ .

**Continuous Extension** From  $x$ , create a double-length  $y$  as

$$y_n = \begin{cases} x_n, & n = 0, 1, \dots, N-1; \\ x_{N-1}, & n = N, N+1, \dots, 2N-1; \\ x_0, & n = 0, -1, \dots, -N+1, \end{cases}$$

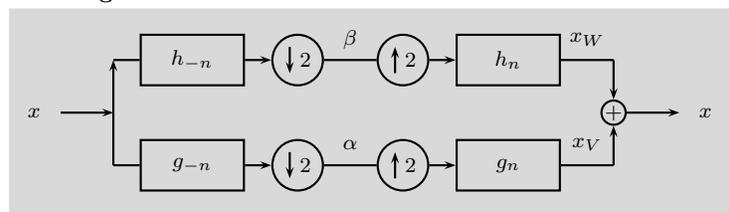
shown in Figure 1.14(e). This extension is also called *boundary replication extension*. It is a relatively smooth extension and is often used in practice.

**Smooth Extension** Another idea is to extend the sequence by polynomial extrapolation, as in Figure 1.14(f). This is only lightly motivated at this point, but after we establish polynomial approximation properties of the discrete wavelet transforms in Chapter 3, it will be clear that a sequence extension by polynomial extrapolation will be a way to get zeros as detail coefficients. The degree of the polynomial is such that on the one hand, it gets annihilated by the zero moments of the wavelet, and on the other hand, it can be extrapolated by the lowpass filter.

## Chapter at a Glance

Our goal in this chapter was to use signal processing machinery to build discrete-time bases with structure in terms of time-frequency localization properties. Moreover, we restricted ourselves to those bases generated by two prototype sequences, one that together with its shifts covers the space of lowpass sequences, and the other that together with its shifts covers the space of highpass sequences. The signal processing tool implementing such bases is a *two-channel filter bank*.

### Block diagram



### Basic characteristics

number of channels	$M = 2$
sampling factor	$N = 2$
channel sequences	$\alpha_n \quad \beta_n$

### Filters

### Synthesis

### Analysis

	lowpass	highpass	lowpass	highpass
orthogonal	$g_n$	$h_n$	$g_{-n}$	$h_{-n}$
biorthogonal	$g_n$	$h_n$	$\tilde{g}_n$	$\tilde{h}_n$
polyphase components	$g_{0,n}, g_{1,n}$	$h_{0,n}, h_{1,n}$	$\tilde{g}_{0,n}, \tilde{g}_{1,n}$	$\tilde{h}_{0,n}, \tilde{h}_{1,n}$

Table 1.7: Two-channel filter bank.

	Synthesis		Analysis	
	lowpass	highpass	lowpass	highpass
Time domain	$g_n$ $(\delta_n + \delta_{n-1})/\sqrt{2}$	$h_n$ $(\delta_n - \delta_{n-1})/\sqrt{2}$	$g_{-n}$ $(\delta_n + \delta_{n+1})/\sqrt{2}$	$h_{-n}$ $(\delta_n - \delta_{n+1})/\sqrt{2}$
$z$ -domain	$G(z)$ $(1 + z^{-1})/\sqrt{2}$	$H(z)$ $(1 - z^{-1})/\sqrt{2}$	$G(z^{-1})$ $(1 + z)/\sqrt{2}$	$H(z^{-1})$ $(1 - z)/\sqrt{2}$
DTFT domain	$G(e^{j\omega})$ $(1 + e^{-j\omega})/\sqrt{2}$	$H(e^{j\omega})$ $(1 - e^{-j\omega})/\sqrt{2}$	$G(e^{-j\omega})$ $(1 + e^{j\omega})/\sqrt{2}$	$H(e^{-j\omega})$ $(1 - e^{j\omega})/\sqrt{2}$

Table 1.8: Haar filter bank.

**Relationship between lowpass and highpass filters**

Time domain	$\langle h_n, g_{n-2k} \rangle_n = 0$
Matrix domain	$D_2 H^T G U_2 = 0$
$z$ domain	$G(z)H(z^{-1}) + G(-z)H(-z^{-1}) = 0$
DTFT domain	$G(e^{j\omega})H(e^{j\omega}) + G(e^{j(\omega+\pi)})H(e^{j(\omega+\pi)}) = 0$
Polyphase domain	$G_0(z)G_1(z^{-1}) + H_0(z)H_1(z^{-1}) = 0$

**Sequences**

Frequency domain	lowpass	highpass
Time domain	$\{g_{n-2k}\}_{k \in \mathbb{Z}}$	$\{h_{n-2k}\}_{k \in \mathbb{Z}}$

**Basis****Filters**

	lowpass	highpass		
Time domain	$g_n$	$\pm(-1)^n g_{-n+2\ell-1}$	lowpass	highpass
$z$ domain	$G(z)$	$\mp z^{-2\ell+1} G(-z^{-1})$	$G(z^{-1})$	$\mp z^{2\ell-1} G(-z)$
DTFT domain	$G(e^{j\omega})$	$\mp e^{j(-2\ell+1)\omega} G(e^{-j(\omega+\pi)})$	$G(e^{-j\omega})$	$\mp e^{j(2\ell-1)\omega} G(e^{j(\omega+\pi)})$

**Synthesis****Analysis****Matrix view**

Time domain	$\Phi$	$\begin{bmatrix} \dots & g_{n-2k} & h_{n-2k} & \dots \end{bmatrix}$
$z$ domain	$\Phi(z)$	$\begin{bmatrix} G(z) & H(z) \\ G(-z) & H(-z) \end{bmatrix}$
DTFT domain	$\Phi(e^{j\omega})$	$\begin{bmatrix} G(e^{j\omega}) & H(e^{j\omega}) \\ G(e^{j(\omega+\pi)}) & H(e^{j(\omega+\pi)}) \end{bmatrix}$
Polyphase domain	$\Phi_p(z)$	$\begin{bmatrix} G_0(z) & H_0(z) \\ G_1(z) & H_1(z) \end{bmatrix}$

**Basis****Constraints**

Time domain	$\Phi^T \Phi = I$
$z$ domain	$\Phi(z^{-1})^T \Phi(z) = I$
DTFT domain	$\Phi^T(e^{-j\omega}) \Phi(e^{j\omega}) = I$
Polyphase domain	$\Phi_p^T(z^{-1}) \Phi_p(z) = I$

**Orthogonality relations****Perfect reconstruction**

$\Phi \Phi^T = I$
$\Phi(z) \Phi^T(z^{-1}) = I$
$\Phi(e^{j\omega}) \Phi^T(e^{-j\omega}) = I$
$\Phi_p(z) \Phi_p^T(z^{-1}) = I$

**Table 1.9:** Two-channel orthogonal filter bank.

**Relationship between lowpass and highpass filters**

Time domain	$\langle h_n, g_{n-2k} \rangle_n = 0$
Matrix domain	$D_2 H^T G U_2 = 0$
$z$ domain	$G(z)H(z^{-1}) + G(-z)H(-z^{-1}) = 0$
DTFT domain	$G(e^{j\omega})H(e^{-j\omega}) + G(e^{j(\omega+\pi)})H(e^{j(-\omega+\pi)}) = 0$
Polyphase domain	$G_0(z)G_1(z^{-1}) + H_0(z)H_1(z^{-1}) = 0$

	Sequences		Dual basis	
	lowpass	highpass	lowpass	highpass
	$\{g_{n-2k}\}_{k \in \mathbb{Z}}$	$\{h_{n-2k}\}_{k \in \mathbb{Z}}$	$\{\tilde{g}_{2k-n}\}_{k \in \mathbb{Z}}$	$\{\tilde{h}_{2k-n}\}_{k \in \mathbb{Z}}$
	Filters		Analysis	
	lowpass	highpass	lowpass	highpass
Time domain	$g_n$	$\pm(-1)^n \tilde{g}_{n-2\ell+1}$	$\tilde{g}_n$	$\pm(-1)^n g_{n+2\ell-1}$
$z$ domain	$G(z)$	$\mp z^{-2\ell+1} \tilde{G}(-z)$	$\tilde{G}(z)$	$\mp z^{2\ell-1} G(-z)$
DTFT domain	$G(e^{j\omega})$	$\mp e^{j(-2\ell+1)\omega} \tilde{G}(e^{j(\omega+\pi)})$	$\tilde{G}(e^{j\omega})$	$\mp e^{j(2\ell-1)\omega} G(e^{j(\omega+\pi)})$
	Matrix view		Dual basis	
	lowpass	highpass	lowpass	highpass
Time domain	$\Phi$	$\begin{bmatrix} \dots & g_{n-2k} & h_{n-2k} & \dots \end{bmatrix}$	$\tilde{\Phi}$	$\begin{bmatrix} \dots & \tilde{g}_{2k-n} & \tilde{h}_{2k-n} & \dots \end{bmatrix}$
$z$ domain	$\Phi(z)$	$\begin{bmatrix} G(z) & H(z) \\ G(-z) & H(-z) \end{bmatrix}$	$\tilde{\Phi}(z)$	$\begin{bmatrix} \tilde{G}(z) & \tilde{H}(z) \\ \tilde{G}(-z) & \tilde{H}(-z) \end{bmatrix}$
DTFT domain	$\Phi(e^{j\omega})$	$\begin{bmatrix} G(e^{j\omega}) & H(e^{j\omega}) \\ G(e^{j(\omega+\pi)}) & H(e^{j(\omega+\pi)}) \end{bmatrix}$	$\tilde{\Phi}(e^{j\omega})$	$\begin{bmatrix} \tilde{G}(e^{j\omega}) & \tilde{H}(e^{j\omega}) \\ \tilde{G}(e^{j(\omega+\pi)}) & \tilde{H}(e^{j(\omega+\pi)}) \end{bmatrix}$
Polyphase domain	$\Phi_p(z)$	$\begin{bmatrix} G_0(z) & H_0(z) \\ G_1(z) & H_1(z) \end{bmatrix}$	$\tilde{\Phi}_p(z)$	$\begin{bmatrix} \tilde{G}_0(z) & \tilde{H}_0(z) \\ \tilde{G}_1(z) & \tilde{H}_1(z) \end{bmatrix}$
	Constraints		Perfect reconstruction	
	lowpass	highpass	lowpass	highpass
Time domain	$\Phi^T \tilde{\Phi} = I$		$\Phi \tilde{\Phi}^T = I$	
$z$ domain	$\Phi(z)^T \tilde{\Phi}(z) = I$		$\Phi(z) \tilde{\Phi}^T(z) = I$	
DTFT domain	$\Phi^T(e^{j\omega}) \tilde{\Phi}(e^{j\omega}) = I$		$\Phi(e^{j\omega}) \tilde{\Phi}^T(e^{j\omega}) = I$	
Polyphase domain	$\Phi_p^T(z) \tilde{\Phi}_p(z) = I$		$\Phi_p(z) \tilde{\Phi}_p^T(z) = I$	

**Table 1.10:** Two-channel biorthogonal filter bank.

## Historical Remarks



Filter banks have been popular in signal processing since the 1970s when the question of critically-sampled filter banks, those with the number of channel samples per unit of time conserved, arose in the context of subband coding of speech. In that method, a speech sequence is split into downsampled frequency bands, allowing for more powerful compression. However, downsampling can create a perceptually disturbing effect known as aliasing, prompting Esteban and Galand [38] in 1977 to propose a simple and elegant *quadrature mirror filters (QMF)* aliasing-removal technique. As QMF solution does not allow for perfect reconstruction, a flurry of work followed to solve the problem. Mintzer [64] as well Smith and Barnwell [83] proposed an orthogonal solution independently in the mid 1980s. Vaidyanathan [96] established a connection to lossless systems, unveiling the factorization and design of paraunitary matrices [100]. For wavelet purposes, Daubechies then designed filters with a maximum number of zeros at  $z = -1$  [29], a solution that goes back to Herrmann's design of maximally flat FIR filters [47]. The equivalent IIR filter design problem leads to Butterworth filters, as derived by Herley and Vetterli [45]. Vetterli solved the biorthogonal filter bank problem [103, 104], while Cohen, Daubechies and Feauveau [23] as well as Vetterli and Herley [105] tackled those with maximum number of zeros at  $z = -1$ . The polyphase framework was used by many authors working on filter banks, but really goes back to earlier work on transmultiplexers by Bellanger and Daguët [7]. The realization that perfect reconstruction subband coding can be used for perfect transmultiplexing appears in [104]. The idea of multichannel structures that can be inverted perfectly, including with quantization, goes back to ladder structures in filter design and implementation, in the works of Bruckens and van den Enden, Marshall, Shah and Kalker [14, 62, 79]. Sweldens generalized this idea under the name of lifting [90], deriving a number of new schemes based on this concept, including filter banks with nonlinear operators and nonuniform sampling.

## Further Reading

**Books and Textbooks** A few standard textbooks on filter banks exist, written by Vaidyanathan [98], Vetterli and Kovačević [106], Strang and Nguyen [87], among others.

**$N$ -Channel Filter Banks** One of the important and immediate generalizations of two-channel filter banks is when we allow the number of channels to be  $N$ . Numerous options are available, from directly designing  $N$ -channel filter banks, studied in detail by Vaidyanathan [96, 97], through those built by cascading filter banks with different number of branches, leading to almost arbitrary frequency divisions. The analysis methods follow closely those of the two-channel filter banks, albeit with more freedom; for example, orthogonality and linear phase are much easier to achieve at the same time. We discuss  $N$ -channel filter banks in detail in Chapter 2, with special emphasis on local Fourier bases.

**Multidimensional Filter Banks** The first difference we encounter when dealing with multidimensional filter banks is that of sampling. Regular sampling with a given density can be accomplished using any number of sampling lattices, each having any number of associated sampling matrices. These have been described in detail by Dubois in [35], and

have been used by Viscito and Allebach [109], Karlsson and Vetterli [50], Kovačević and Vetterli [55], Do and Vetterli [33], among others, to design multidimensional filter banks. Apart from the freedom coming with different sampling schemes, the associated filters can now be truly multidimensional, allowing for a much larger space of solutions.

**IIR Filter Banks** While IIR filters should be of importance because of their good frequency selectivity and computational efficiency, they have not been used extensively as their implementation in a filter-bank framework comes at a cost: one side of the filter bank is necessarily anticausal. They have found some use in image processing as the finite length of the input allows for storing the state in the middle of the filter bank and synthesizing from that stored state. Coverage of IIR filter banks can be found in [45, 70, 82].

**Oversampled Filter Banks** Yet another generalization occurs when we allow for redundancy, leading to overcomplete filter banks implementing frame expansions, covered in Chapter 4. These filter banks are becoming popular in applications due to inherent freedom in design.

**Complex-Coefficient Filter Banks** This entire chapter dealt exclusively with real-coefficient filter banks, due to their prevalence in practice. Complex-coefficient filter banks exist, from the very early QMFs [66] to more recent ones, mostly in the form of complex exponential-modulated local Fourier bases, discussed in Section 2.3, as well as the redundant ones, such as Gabor frames [10–12, 26, 39], discussed in Chapter 4.

**QMF Filter Banks** QMF filter banks showed the true potential of filter banks, as it was clear that one could have nonideal filters and still split and reconstruct the input spectrum. The excitement was further spurred by the famous linear-phase designs by Johnston [49] in 1980. Exercise ?? discusses derivation of these filters and their properties.

**Time-Varying Filter Banks and Boundary Filters** The periodic shift variance of filter banks can be exploited to change a filter bank essentially every period. This was done for years in audio coding through the so-called MDCT filter banks, discussed in Section 2.4.1. Herley and Vetterli proposed a more formal approach in [46], by designing different filters to be used at the boundary of a finite-length input, or a filter-bank change.

**Transmultiplexing** The dual scheme to a filter bank is known as a transmultiplexer, where two sequences are synthesized into a combined sequence from which the two parts can be extracted perfectly. An orthogonal decomposition with many channels leads to *orthogonal frequency-division multiplexing (OFDM)*, the basis for many modulation schemes used in communications, such as 802.11. The analysis of transmultiplexers uses similar tools as for filter banks [104], covered in Solved Exercise ?? for the orthogonal case, and in Exercise ?? for the biorthogonal case. Exercise ?? considers frequency-division multiplexing with Haar filters.

**Filter Banks with Stochastic Inputs** Among the wealth of filter banks available, it is often necessary to determine which one is the most suitable for a given application. A number of measures have been proposed, for example, quantifying shift variance of subband energies for deterministic inputs. Similarly, in [1, 2], the author proposes, among others, a counterpart measure based on the cyclostationarity of subband powers. We do not dwell on these here, rather we leave the discussion for Chapter 7. Akkarakaran and

---

**54 Chapter 1. Filter Banks: Building Blocks of Time-Frequency Expansions**

---

Vaidyanathan in [4] discuss bifrequency and bispectrum maps (deterministic and stochastic time-varying autocorrelations) in filter banks and answer many relevant questions; some similar issues are tackled by Therrien in [91]. In [99], Vaidyanathan and Akkarakaran give a review of optimal filter banks based on input statistics. In particular, principal-component filter banks offer optimal solutions to various problems, some of these discussed in [93] and [94,95].

## Chapter 2

# Local Fourier Bases on Sequences

### Contents

<b>2.1</b>	<b>Introduction</b>	<b>56</b>
<b>2.2</b>	<b><i>N</i>-Channel Filter Banks</b>	<b>59</b>
<b>2.3</b>	<b>Complex Exponential-Modulated Local Fourier Bases</b>	<b>66</b>
<b>2.4</b>	<b>Cosine-Modulated Local Fourier Bases</b>	<b>75</b>
<b>2.5</b>	<b>Computational Aspects</b>	<b>85</b>
	<b>Chapter at a Glance</b>	<b>87</b>
	<b>Historical Remarks</b>	<b>88</b>
	<b>Further Reading</b>	<b>90</b>

Think of a piece of music: notes appear at different instants of time, and then fade away. These are short-time frequency events the human ear identifies easily, but are a challenge for a computer to understand. These notes are well identified frequencies, but they are short lived. Thus, we would like to have access to a *local Fourier transform*, that is, a time-frequency analysis tool that understands the spectrum locally in time. While such a transform is known under many names, such as *windowed Fourier transform*, *Gabor transform* and *short-time Fourier transform*, we will use local Fourier transform exclusively throughout the manuscript. The local energy distribution over frequency, which can be obtained by squaring the magnitude of the local Fourier coefficients, is called the *spectrogram*, and is widely used in speech processing and time-series analysis.

Our purpose in this chapter is to explore what is possible in terms of obtaining such a local version of the Fourier transform of a sequence. While, unfortunately, we will see that, apart from short ones, there exist no good longer local Fourier bases, there exist good *local Fourier frames*, the topic we explore in Chapter 4. Moreover, there exist good *local cosine* bases, where the complex-exponential modulation is replaced by cosine modulation. These constructions will all be implemented using general, *N*-channel filter banks, the first generalization of the basic two-channel filter bank block we just saw in the last chapter.

## 2.1 Introduction

We now look at the simplest example of a local Fourier transform decomposing the spectrum into  $N$  equal parts. As we have learned in the previous chapter, for  $N = 2$ , two-channel filter banks do the trick; for a general  $N$ , it is no surprise that  $N$ -channel filter banks perform that role, and we now show just that.

If we have an infinite-length sequence, we could use the DTFT we discussed in *Chapter 3*; however, as we mentioned earlier, this representation will erase any time-local information present in the sequence. We could, however, use another tool also discussed in *Chapter 3*, the DFT. While we have said that the DFT is a natural tool for the analysis of either periodic sequences or infinite-length sequences with a finite number of nonzero samples, circularly extended, we can also use the DFT as a tool to observe the local behavior of an infinite-length sequence by dividing it into pieces of length  $N$ , followed by a length- $N$  DFT.

### Implementing a Length- $N$ DFT Basis Expansion

We now mimic what we have done for the Haar basis in the previous chapter, that is, implement the DFT basis using signal processing machinery. We start with the basis view of the DFT from *Section 3.6.1*; we assume this finite-dimensional basis is applied to length- $N$  pieces of our input sequence. The final basis then consists of  $\{\varphi_i\}_{i=0}^{N-1}$  from (3.162) and all their shifts by integer multiples of  $N$ , that is,

$$\Phi_{\text{DFT}} = \{\varphi_{i,n-Nk}\}_{i \in \{0,1,\dots,N-1\}, k \in \mathbb{Z}}. \quad (2.1)$$

In other words, as opposed to two template basis sequences generating the entire basis by shifting as in the Haar case, not surprisingly, we now have  $N$  template basis sequences generating the entire basis by shifting. We rename those template basis sequences to (we use the normalized version of the DFT):

$$g_{i,n} = \varphi_{i,n} = \frac{1}{\sqrt{N}} W_N^{-in}. \quad (2.2)$$

This is again done both for simplicity, as well as because it is the standard way these sequences are denoted.

Then, we rewrite the reconstruction formula (3.160b) as

$$\begin{aligned} x_n &= \sum_{i=0}^{N-1} \sum_{k \in \mathbb{Z}} \underbrace{\langle x_n, \varphi_{i,n-Nk} \rangle_n}_{\alpha_{i,k}} \varphi_{i,n-Nk} \\ &= \sum_{i=0}^{N-1} \sum_{k \in \mathbb{Z}} \alpha_{i,k} \underbrace{\varphi_{i,n-Nk}}_{g_{i,n-Nk}} = \sum_{i=0}^{N-1} \sum_{k \in \mathbb{Z}} \alpha_{i,k} g_{i,n-Nk}, \end{aligned} \quad (2.3a)$$

where we have renamed the basis sequences as explained above, as well as denoted the expansion coefficients as

$$\langle x_n, \varphi_{i,n-Nk} \rangle_n = \langle x_n, g_{i,n-Nk} \rangle_n = \alpha_{i,k}. \quad (2.3b)$$



with  $F$  as in (3.161a), and the synthesis process (2.3a) as

$$\begin{bmatrix} \vdots \\ \boxed{x_0} \\ \vdots \\ x_{N-1} \\ x_N \\ \vdots \\ x_{2N-1} \\ \vdots \end{bmatrix} = \frac{1}{\sqrt{N}} \underbrace{\begin{bmatrix} \ddots & & & & \\ & F^* & & & \\ & & F^* & & \\ & & & \ddots & \\ & & & & \ddots \end{bmatrix}}_{\Phi} \begin{bmatrix} \vdots \\ \boxed{\alpha_{0,0}} \\ \vdots \\ \alpha_{N-1,0} \\ \alpha_{0,1} \\ \vdots \\ \alpha_{N-1,1} \\ \vdots \end{bmatrix}. \quad (2.4b)$$

Of course,  $\Phi$  is a unitary matrix, since  $F/\sqrt{N}$  is.

**Localization Properties of the Length- $N$  DFT** It is quite clear that the time localization properties of the DFT are superior to those of the DTFT, as now, we have access to the time-local events at the resolution of length  $N$ . However, as a result, the frequency resolution must necessarily worsen; to see this, consider the frequency response of  $g_0$  (the other  $g_k$ s are modulated versions and therefore have the same frequency resolution):

$$G_0(e^{j\omega}) = \sqrt{N} \frac{\text{sinc}(\omega N/2)}{\text{sinc}(\omega/2)}, \quad (2.5)$$

that is, it is the DTFT of a box sequence (see Table 4.6. It has zeros at  $\omega = 2\pi k/N$ ,  $k = 1, 2, \dots, N-1$ , but decays slowly in between.

The orthonormal basis given by the DFT is just one of many basis options implementable by  $N$ -channel filter banks; many others, with template basis sequences with more than  $N$  nonzero samples are possible (similarly to the two-channel case). The DFT is a local Fourier version as the time events can be captured with the resolution of  $N$  samples.

## Chapter Outline

This short introduction leads naturally to the following structure of the chapter: In Section 2.2, we give an overview of  $N$ -channel filter banks. In Section 2.3, we present the local Fourier bases implementable by complex exponential-modulated filter banks. We then come to the crucial, albeit negative result: the Balian-Low theorem, which states the impossibility of good complex exponential-modulated local Fourier bases. We look into their applications: local power spectral density via periodograms, as well as in transmultiplexing. To mitigate the Balian-Low negative result, Section 2.4 considers what happens if we use cosine modulation instead of the complex one to obtain a local frequency analysis. In the block-transform case, we encounter the *discrete cosine transform*, which plays a prominent role in image processing. In the sliding window case, a cosine-modulated filter bank allows

the best of both worlds, namely an orthonormal basis with good time-frequency localization. We also discuss variations on this construction as well as an application to audio compression.

*Notation used in this chapter:* Unlike in the previous chapter, in this one, complex-coefficient filter banks are the norm. Thus, Hermitian transposition is used often, with the caveat that only coefficients should be conjugated and not  $z$ . We will point these out throughout the chapter.  $\square$

## 2.2 $N$ -Channel Filter Banks

We could imagine achieving our goal of splicing the spectrum into  $N$  pieces many ways; we have just seen one, achievable by using the DFT, a representation with reasonable time but poor frequency localization. Another option is using an ideal  $N$ th band filter and its shifts (we have seen it in *Table 3.5* and *Table 4.6*, as well as (3.108) with  $\omega_0 = 2\pi/N$ , but repeat it here for completeness):

$$g_{0,n} = \frac{1}{\sqrt{N}} \operatorname{sinc}(\pi n/N) \quad \xleftrightarrow{\text{DFT}} \quad G_0(e^{j\omega}) = \begin{cases} \sqrt{N}, & |\omega| \leq \pi/N; \\ 0, & \text{otherwise,} \end{cases} \quad (2.6)$$

which clearly has perfect frequency localization but poor time localization as its impulse response is a discrete sinc sequence. We have discussed this trade-off already in *Chapter 7*, and depict it in *Figure 2.2*.

The question now is whether there exist constructions in between these two extreme cases? Specifically, are there basis sequences with better frequency localization than the block transform, but with impulse responses that decay faster than the sinc impulse response (for example, a finite impulse response)?

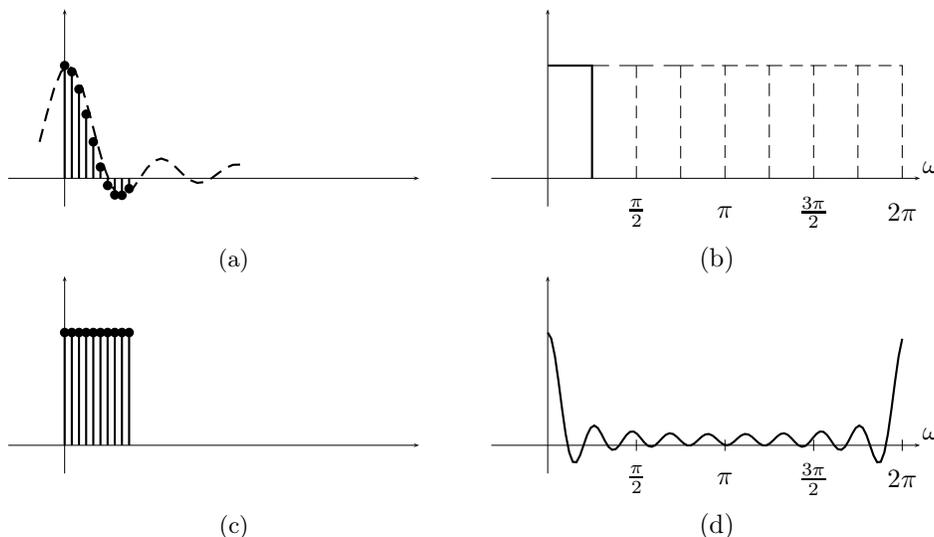
To explore this issue, we introduce general  $N$ -channel filter banks. These are as shown in *Figure 2.1*, where the input is analyzed by  $N$  filters  $\tilde{g}_i$ ,  $i = 0, 1, \dots, N-1$ , and downsampled by  $N$ . The synthesis is done by upsampling by  $N$ , followed by interpolation with  $g_i$ ,  $i = 0, 1, \dots, N-1$ .

The analysis of  $N$ -channel filter banks can be done in complete analogy to the two-channel case, by using the relevant equations for sampling rate changes by  $N$ . We now state these without proofs, and illustrate them on a particular case of a 3-channel filter bank, especially in polyphase domain.

### 2.2.1 Orthogonal $N$ -Channel Filter Banks

As for two-channel filter banks,  $N$ -channel orthogonal filter banks are of particular interest; the DFT is one example. We now briefly follow the path from the previous chapter and put in one place the relations governing such filter banks. The biorthogonal ones follow similarly, and we just touch upon them during the discussion of the polyphase view.

**Orthogonality of a Single Filter** Since we started with an orthonormal basis, the set  $\{g_{i,n-Nk}\}_{k \in \mathbb{Z}, i \in \{0,1,\dots,N-1\}}$  is an orthonormal set. We have seen in *Section 3.7.4*



**Figure 2.2:** Time- and frequency-domain behaviors of two orthonormal bases with  $N = 8$  channels. (a)–(b) Sinc basis. (a) Impulse response is a sinc sequence, with poor time localization. (b) Frequency response is a box function, with perfect frequency localization. (c)–(d) DFT basis. (c) Impulse response is a box sequence, with good time localization. (d) Frequency response is a sinc function, with poor frequency localization.

that each such filter is orthogonal and satisfies, analogously to (3.215):

$$\begin{aligned}
 \langle g_{i,n}, g_{i,n-Nk} \rangle &= \delta_k && \begin{array}{l} \text{Matrix View} \\ \xleftrightarrow{\text{ZT}} \\ \xleftrightarrow{\text{DTFT}} \end{array} && \begin{array}{l} D_N G_i^T G_i U_N = I \\ \sum_{k=0}^{N-1} G_i(W_N^k z) G_i(W_N^{-k} z^{-1}) = N \\ \sum_{k=0}^{N-1} |G_i(e^{j(\omega-(2\pi/N)k)})|^2 = N \end{array} \quad (2.7)
 \end{aligned}$$

As before, the matrix view expresses the fact that the columns of  $G_i U_N$  form an orthonormal set and the DTFT version is a generalization of the quadrature mirror formula (3.214). For example, take  $g_0$  and  $N = 3$ . The DTFT version is then

$$|G_0(e^{j\omega})|^2 + |G_0(e^{j(\omega-2\pi/3)})|^2 + |G_0(e^{j(\omega-4\pi/3)})|^2 = 3;$$

essentially, the magnitude response squared of the filter, added to its modulated versions by  $2\pi/N$  and  $4\pi/N$ , sum up to a constant. This is easily seen in the case of an ideal third-band filter, whose frequency response would be constant  $\sqrt{3}$  (see (2.6)), and thus squared and shifted across the spectrum would satisfy the above.

**Deterministic Autocorrelation of a Single Filter** With  $a_i$  the deterministic autocorrelation of  $g_i$ , the deterministic autocorrelation version of (2.7) is straightfor-

ward:

$$\begin{array}{r}
 \langle g_{i,n}, g_{i,n-Nk} \rangle = a_{i,Nk} = \delta_k \\
 \begin{array}{c}
 \xleftrightarrow{\text{Matrix View}} \\
 \xleftrightarrow{\text{ZT}} \\
 \xleftrightarrow{\text{DTFT}}
 \end{array}
 \end{array}
 \begin{array}{l}
 D_N A_i U_N = I \\
 \sum_{k=0}^{N-1} A_i (W_N^k z) = N \\
 \sum_{k=0}^{N-1} A_i (e^{j(\omega - (2\pi/N)k)}) = N
 \end{array}
 \quad (2.8)$$

**Orthogonal Projection Property a Single Channel** Analogously to two channels, a single channel with orthogonal filters projects onto a coarse subspace  $V_0$  or detail subspaces  $W_i$ ,  $i = 1, 2, \dots, N-1$ , depending on the frequency properties of the filter. Each of the orthogonal projection operators is given as

$$\begin{aligned}
 P_{V_0} &= G_0 U_N D_N G_0^T, \\
 P_{W_i} &= G_i U_N D_N G_i^T, \quad i = 1, 2, \dots, N-1,
 \end{aligned}$$

with the range

$$\begin{aligned}
 V_0 &= \text{span}(\{g_{0,n-Nk}\}_{k \in \mathbb{Z}}), \\
 W_i &= \text{span}(\{g_{i,n-Nk}\}_{k \in \mathbb{Z}}), \quad i = 1, 2, \dots, N-1.
 \end{aligned}$$

**Orthogonality of Filters** As in the previous chapter, once the orthonormality of each single channel is established, what is left is the orthogonality of the channels among themselves. Again, all the expressions are analogous to the two-channel case; we state them here without proof. We assume below that  $i \neq j$ .

$$\begin{array}{r}
 \langle g_{i,n}, g_{j,n-Nk} \rangle = 0 \\
 \begin{array}{c}
 \xleftrightarrow{\text{Matrix View}} \\
 \xleftrightarrow{\text{ZT}} \\
 \xleftrightarrow{\text{DTFT}}
 \end{array}
 \end{array}
 \begin{array}{l}
 D_N G_j^T G_i U_N = 0 \\
 \sum_{k=0}^{N-1} G_i (W_N^k z) G_j (W_N^{-k} z^{-1}) = 0 \\
 \sum_{k=0}^{N-1} G_i (e^{j(\omega - (2\pi/N)k)}) G_j (e^{-j(\omega + (2\pi/N)k)}) = 0
 \end{array}
 \quad (2.9)$$

**Deterministic Crosscorrelation of Filters** Calling  $c_{i,j}$  the deterministic crosscorrelation of  $g_i$  and  $g_j$ :

$$\begin{array}{r}
 \langle g_{i,n}, g_{j,n-Nk} \rangle = c_{i,j,Nk} = 0 \\
 \begin{array}{c}
 \xleftrightarrow{\text{Matrix View}} \\
 \xleftrightarrow{\text{ZT}} \\
 \xleftrightarrow{\text{DTFT}}
 \end{array}
 \end{array}
 \begin{array}{l}
 D_N C_{i,j} U_N = 0 \\
 \sum_{k=0}^{N-1} C_{i,j} (W_N^k z) = 0 \\
 \sum_{k=0}^{N-1} C_{i,j} (e^{j(\omega - (2\pi/N)k)}) = 0
 \end{array}
 \quad (2.10)$$

### 2.2.2 Polyphase View of $N$ -Channel Filter Banks

To cover the polyphase view for general  $N$ , we cover it through an example with  $N = 3$ , and then briefly summarize the discussion for a general  $N$ .

EXAMPLE 2.1 (ORTHOGONAL 3-CHANNEL FILTER BANKS) For two-channel filter banks, a polyphase decomposition is achieved by simply splitting both sequences and filters into their even- and odd-indexed subsequences; for 3-channel filter banks, we split sequences and filters into subsequences modulo 3. While we have seen the expression for a polyphase representation of a sequence and filters for a general  $N$  in (3.227), we write them out for  $N = 3$  to develop some intuition, starting with the input sequence  $x$ :

$$\begin{aligned} x_{0,n} &= x_{3n} \quad \xleftrightarrow{\text{ZT}} \quad X_0(z) = \sum_{n \in \mathbb{Z}} x_{3n} z^{-n}, \\ x_{1,n} &= x_{3n+1} \quad \xleftrightarrow{\text{ZT}} \quad X_1(z) = \sum_{n \in \mathbb{Z}} x_{3n+1} z^{-n}, \\ x_{2,n} &= x_{3n+2} \quad \xleftrightarrow{\text{ZT}} \quad X_2(z) = \sum_{n \in \mathbb{Z}} x_{3n+2} z^{-n}, \\ X(z) &= X_0(z^3) + z^{-1}X_1(z^3) + z^{-2}X_2(z^3). \end{aligned}$$

In the above,  $x_0$  is the subsequence of  $x$  at multiples of 3 downsampled by 3, and similarly for  $x_1$  and  $x_2$ :

$$\begin{aligned} x_0 &= [\dots \quad x_{-3} \quad \boxed{x_0} \quad x_3 \quad x_6 \quad \dots]^T, \\ x_1 &= [\dots \quad x_{-2} \quad \boxed{x_1} \quad x_4 \quad x_7 \quad \dots]^T, \\ x_2 &= [\dots \quad x_{-1} \quad \boxed{x_2} \quad x_5 \quad x_8 \quad \dots]^T. \end{aligned}$$

This is illustrated in Figure 2.3(a): to get  $x_0$  we simply keep every third sample from  $x$ ; to get  $x_1$ , we shift  $x$  by one to the left (advance by one represented by  $z$ ) and then keep every third sample; finally, to get  $x_2$ , we shift  $x$  by two to the left and then keep every third sample. To get the original sequence back, we upsample each subsequence by 3, shift appropriately to the right (delays represented by  $z^{-1}$  and  $z^{-2}$ ), and sum up.

Using (3.227), we define the polyphase decomposition of the synthesis filters:

$$g_{i,0,n} = g_{i,3n} \quad \xleftrightarrow{\text{ZT}} \quad G_{i,0}(z) = \sum_{n \in \mathbb{Z}} g_{i,3n} z^{-n}, \quad (2.11a)$$

$$g_{i,1,n} = g_{i,3n+1} \quad \xleftrightarrow{\text{ZT}} \quad G_{i,1}(z) = \sum_{n \in \mathbb{Z}} g_{i,3n+1} z^{-n}, \quad (2.11b)$$

$$g_{i,2,n} = g_{i,3n+2} \quad \xleftrightarrow{\text{ZT}} \quad G_{i,2}(z) = \sum_{n \in \mathbb{Z}} g_{i,3n+2} z^{-n}, \quad (2.11c)$$

$$G_i(z) = G_{i,0}(z^3) + z^{-1}G_{i,1}(z^3) + z^{-2}G_{i,2}(z^3),$$

where the first subscript denotes the filter, the second the polyphase component, and the last, the discrete time index. In the above, we split each synthesis filter

into its subsequences modulo 3 as we have done for the input sequence  $x$ :

$$\begin{aligned} g_0 &= \left[ \cdots \quad g_{-3} \quad \boxed{g_0} \quad g_3 \quad g_6 \quad \cdots \right]^T, \\ g_1 &= \left[ \cdots \quad g_{-2} \quad \boxed{g_1} \quad g_4 \quad g_7 \quad \cdots \right]^T, \\ g_2 &= \left[ \cdots \quad g_{-1} \quad \boxed{g_2} \quad g_5 \quad g_8 \quad \cdots \right]^T. \end{aligned}$$

We can now define the *polyphase matrix*  $\Phi_p(z)$ :

$$\Phi_p(z) = \begin{bmatrix} G_{0,0}(z) & G_{1,0}(z) & G_{2,0}(z) \\ G_{0,1}(z) & G_{1,1}(z) & G_{2,1}(z) \\ G_{0,2}(z) & G_{1,2}(z) & G_{2,2}(z) \end{bmatrix}.$$

The matrix above is on the synthesis side; to get it on the analysis side, we define the polyphase decomposition of analysis filters using (3.227) and similarly to what we have done in the two-channel case:

$$\begin{aligned} \tilde{g}_{i,0,n} = \tilde{g}_{i,3n} = g_{i,-3n} &\xleftrightarrow{ZT} \tilde{G}_{i,0}(z) = \sum_{n \in \mathbb{Z}} g_{i,-3n} z^{-n}, \\ \tilde{g}_{i,1,n} = \tilde{g}_{i,3n-1} = g_{i,-3n+1} &\xleftrightarrow{ZT} \tilde{G}_{i,1}(z) = \sum_{n \in \mathbb{Z}} g_{i,-3n+1} z^{-n}, \\ \tilde{g}_{i,2,n} = \tilde{g}_{i,3n-2} = g_{i,-3n+2} &\xleftrightarrow{ZT} \tilde{G}_{i,2}(z) = \sum_{n \in \mathbb{Z}} g_{i,-3n+2} z^{-n}, \\ \tilde{G}(z) &= G_{i,0}(z^{-3}) + zG_{i,1}(z^{-3}) + z^2G_{i,2}(z^{-3}). \end{aligned}$$

The three polyphase components are:

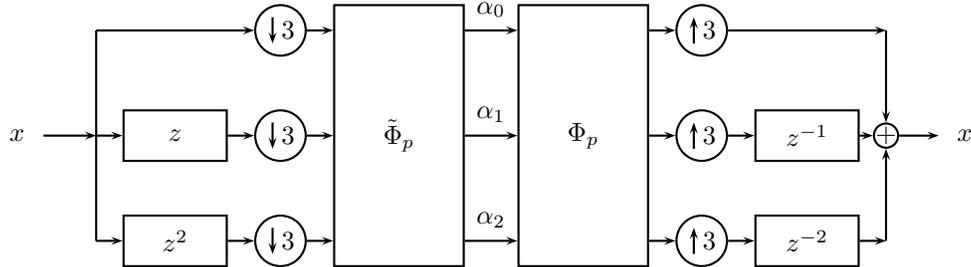
$$\begin{aligned} \tilde{g}_{0,n} &= \left[ \cdots \quad \tilde{g}_{-3} \quad \boxed{\tilde{g}_0} \quad \tilde{g}_3 \quad \tilde{g}_6 \quad \cdots \right]^T = \left[ \cdots \quad g_3 \quad \boxed{g_0} \quad g_{-3} \quad g_{-6} \quad \cdots \right]^T, \\ \tilde{g}_1 &= \left[ \cdots \quad \tilde{g}_{-4} \quad \boxed{\tilde{g}_{-1}} \quad \tilde{g}_2 \quad \tilde{g}_5 \quad \cdots \right]^T = \left[ \cdots \quad g_4 \quad \boxed{g_1} \quad g_{-2} \quad g_{-5} \quad \cdots \right]^T, \\ \tilde{g}_2 &= \left[ \cdots \quad \tilde{g}_{-5} \quad \boxed{\tilde{g}_{-2}} \quad \tilde{g}_1 \quad \tilde{g}_4 \quad \cdots \right]^T = \left[ \cdots \quad g_5 \quad \boxed{g_2} \quad g_{-1} \quad g_{-4} \quad \cdots \right]^T. \end{aligned}$$

Note that  $\tilde{G}_i(z) = G_i(z^{-1})$ , as in the two-channel case. With these definitions, the analysis polyphase matrix is:

$$\tilde{\Phi}_p(z) = \begin{bmatrix} G_{0,0}(z^{-1}) & G_{1,0}(z^{-1}) & G_{2,0}(z^{-1}) \\ G_{0,1}(z^{-1}) & G_{1,1}(z^{-1}) & G_{2,1}(z^{-1}) \\ G_{0,2}(z^{-1}) & G_{1,2}(z^{-1}) & G_{2,2}(z^{-1}) \end{bmatrix} = \Phi_p(z^{-1}).$$

Figure 2.3 shows the polyphase implementation of the system, with the reconstruction of the original sequence using the synthesis polyphase matrix on the right<sup>12</sup> and the computation of projection sequences  $\alpha_i$  on the left; note that as

<sup>12</sup>Remember that we typically put the lowpass filter in the lower branch, but in matrices it appears in the first row/column, leading to a slight inconsistency when the filter bank is depicted in the polyphase domain.



**Figure 2.3:** A 3-channel analysis/synthesis filter bank in polyphase domain.

usual, the analysis matrix (polyphase here) is taken as a transpose (to check it, we could mimic what we did in Section 1.2.4; we skip it here).

The upshot of all this algebra is that we now have a very compact input-output relationship between the input (decomposed into polyphase components) and the result coming out of the synthesis filter bank:

$$X(z) = \begin{bmatrix} 1 & z^{-1} & z^{-2} \end{bmatrix} \Phi_p(z^3) \Phi_p^*(z^{-3}) \begin{bmatrix} X_0(z^3) \\ X_1(z^3) \\ X_2(z^3) \end{bmatrix}.$$

Note that we use the Hermitian transpose here because we will often deal with complex-coefficient filter banks in this chapter. The conjugation is applied only to coefficients and not to  $z$ . The above example went through various polyphase concepts for an orthogonal 3-channel filter bank. We now summarize the same concepts for a general, biorthogonal  $N$ -channel filter bank, and characterize classes of solutions using polyphase machinery.

Using (3.227), in an  $N$ -channel filter bank, the polyphase decomposition of the input sequence, synthesis and analysis filters, respectively, is given by:

$$x_{j,n} = x_{Nn+j} \xleftrightarrow{\text{ZT}} X_j(z) = \sum_{n \in \mathbb{Z}} x_{Nn+j} z^{-n}, \quad (2.12a)$$

$$X(z) = \sum_{j=0}^{N-1} z^{-j} X_j(z^N), \quad (2.12b)$$

$$g_{i,j,n} = g_{i,Nn+j} \xleftrightarrow{\text{ZT}} G_{i,j}(z) = \sum_{n \in \mathbb{Z}} g_{i,Nn+j} z^{-n}, \quad (2.12c)$$

$$G_i(z) = \sum_{j=0}^{N-1} z^{-j} G_{i,j}(z^N), \quad (2.12d)$$

$$\tilde{g}_{i,j,n} = \tilde{g}_{i,Nn-j} \xleftrightarrow{\text{ZT}} \tilde{G}_{i,j}(z) = \sum_{n \in \mathbb{Z}} \tilde{g}_{i,Nn-j} z^{-n}, \quad (2.12e)$$

$$\tilde{G}_i(z) = \sum_{j=0}^{N-1} z^j \tilde{G}_{i,j}(z^N), \quad (2.12f)$$

leading to the corresponding polyphase matrices:

$$\Phi_p(z) = \begin{bmatrix} G_{0,0}(z) & G_{1,0}(z) & \dots & G_{N-1,0}(z) \\ G_{0,1}(z) & G_{1,1}(z) & \dots & G_{N-1,1}(z) \\ \vdots & \vdots & \ddots & \vdots \\ G_{0,N-1}(z) & G_{1,N-1}(z) & \dots & G_{N-1,N-1}(z) \end{bmatrix}, \quad (2.13a)$$

$$\tilde{\Phi}_p(z) = \begin{bmatrix} \tilde{G}_{0,0}(z) & \tilde{G}_{1,0}(z) & \dots & \tilde{G}_{N-1,0}(z) \\ \tilde{G}_{0,1}(z) & \tilde{G}_{1,1}(z) & \dots & \tilde{G}_{N-1,1}(z) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{G}_{0,N-1}(z) & \tilde{G}_{1,N-1}(z) & \dots & \tilde{G}_{N-1,N-1}(z) \end{bmatrix}. \quad (2.13b)$$

This formulation allows us to characterize classes of solutions. We state these without proof as they follow easily from the equivalent two-channel filter bank results, and can be found in the literature.

**THEOREM 2.1 ( $N$ -CHANNEL FILTER BANKS IN POLYPHASE DOMAIN)** Given is an  $N$ -channel filter bank and the polyphase matrices  $\Phi_p(z)$ ,  $\tilde{\Phi}_p(z)$ . Then:

- (i) The filter bank implements a biorthogonal expansion if and only if

$$\Phi_p(z) \tilde{\Phi}_p^*(z) = I. \quad (2.14a)$$

- (ii) The filter bank implements an orthonormal expansion if and only if

$$\Phi_p(z) \Phi_p^*(z^{-1}) = I, \quad (2.14b)$$

that is,  $\Phi_p(z)$  is paraunitary.

- (iii) The filter bank implements an FIR biorthogonal expansion if and only if  $\Phi_p(z)$  is unimodular (within scaling), that is, if

$$\det(\Phi_p(z)) = \alpha z^{-k}. \quad (2.14c)$$

Note that we use the Hermitian transpose in (2.14b) because we will often deal with complex-coefficient filter banks in this chapter. The conjugation is applied only to coefficients and not to  $z$ .

**Design of  $N$ -Channel Filter Banks** In the next two sections, we will discuss two particular  $N$ -channel filter bank design options, in particular, those that add localization features to the DFT. To design general  $N$ -channel orthogonal filter banks,

we must design  $N \times N$  paraunitary matrices. As in the two-channel case, where such matrices can be obtained by a lattice factorization (see Section 1.3.3),  $N \times N$  paraunitary matrices can be parameterized in terms of elementary matrices ( $2 \times 2$  rotations and delays). Here, we just give an example of a design of a  $3 \times 3$  paraunitary matrix leading to a 3-channel orthogonal filter bank; pointers to literature are given in *Further Reading*.

**EXAMPLE 2.2 (ORTHOGONAL  $N$ -CHANNEL FILTER BANKS)** One way of parameterizing paraunitary matrices is via the following factorization:

$$\Phi_p(z) = U_0 \left[ \prod_{k=1}^{K-1} \text{diag}([z^{-1}, 1, 1]) U_k \right], \quad (2.15a)$$

where

$$U_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_{00} & -\sin \theta_{00} \\ 0 & \sin \theta_{00} & \cos \theta_{00} \end{bmatrix} \begin{bmatrix} \cos \theta_{01} & 0 & -\sin \theta_{01} \\ 0 & 1 & 0 \\ \sin \theta_{01} & 0 & \cos \theta_{01} \end{bmatrix} \begin{bmatrix} \cos \theta_{02} & -\sin \theta_{02} & 0 \\ \sin \theta_{02} & \cos \theta_{02} & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$U_k = \begin{bmatrix} \cos \theta_{k0} & -\sin \theta_{k0} & 0 \\ \sin \theta_{k0} & \cos \theta_{k0} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_{k1} & -\sin \theta_{k1} \\ 0 & \sin \theta_{k1} & \cos \theta_{k1} \end{bmatrix}. \quad (2.15b)$$

The degrees of freedom in design are given by the angles  $\theta_{kj}$ . This freedom in design allows for constructions of orthogonal and linear-phase FIR solutions, not possible in the two-channel case.

## 2.3 Complex Exponential-Modulated Local Fourier Bases

At the start of the previous section, we considered two extreme cases of local Fourier representations implementable by  $N$ -channel filter banks: those based on the DFT (box in time/sinc in frequency, good time/poor frequency localization) and those based on ideal bandpass filters (sinc in time/box in frequency, good frequency/poor time localization). These two particular representations have something in common; as implemented via  $N$ -channel orthogonal filter banks, they are both obtained through a complex-exponential modulation of a single prototype filter.

**Complex-Exponential Modulation** Given a prototype filter  $p = g_0$ , the rest of the filters are obtained via complex-exponential modulation:

$$g_{i,n} = p_n e^{j(2\pi/N)in} = p_n W_N^{-in}, \quad (2.16)$$

$$G_i(z) = P(W_N^i z),$$

$$G_i(e^{j\omega}) = P(e^{j(\omega - (2\pi/N)i)}) = P(W_N^i e^{j\omega}),$$

for  $i = 1, 2, \dots, N-1$ . This is clearly true for the DFT basis from (2.2), (2.5), as well as that constructed from the ideal filters (2.6) (see also Exercise ??). A filter bank

implementing such an expansion is often called *complex exponential-modulated filter bank*. While the prototype filter  $p = g_0$  is typically real, the rest of the bandpass filters are complex.

### 2.3.1 Balian-Low Theorem

We are now back to the question whether we can find complex exponential-modulated local Fourier bases with a trade-off of time and frequency localization we have seen for the DFT and sinc bases in Figure 2.2. To that end, we might want to worsen the time localization of the DFT a bit in the hope of improving the frequency one; unfortunately, the following result excludes the possibility of having complex exponential-modulated local Fourier bases with support longer than  $N$ :<sup>13</sup>

**THEOREM 2.2 (DISCRETE BALIAN-LOW THEOREM)** There does not exist a complex exponential-modulated local Fourier basis implementable by an  $N$ -channel FIR filter bank, except for a filter bank with filters of length  $N$ .

*Proof.* To prove the theorem, we analyze the structure of the polyphase matrix of a complex exponential-modulated filter bank with filters as in (2.16). Given the polyphase representation (2.12d) of the prototype filter  $p = g_0$ ,

$$P(z) = P_0(z^N) + z^{-1}P_1(z^N) + \dots + z^{-(N-1)}P_{N-1}(z^N),$$

the modulated versions become

$$G_i(z) = P(W_N^i z) = P_0(z^N) + W_N^{-i} z^{-1} P_1(z^N) + \dots + W_N^{-(N-1)i} z^{-(N-1)} P_{N-1}(z^N)$$

for  $i = 1, 2, \dots, N - 1$ . As an example, for  $N = 3$ , the polyphase matrix is

$$\begin{aligned} \Phi_p(z) &= \begin{bmatrix} P_0(z) & P_0(z) & P_0(z) \\ P_1(z) & W_3^{-1}P_1(z) & W_3^{-2}P_1(z) \\ P_2(z) & W_3^{-2}P_2(z) & W_3^{-1}P_2(z) \end{bmatrix} \\ &= \begin{bmatrix} P_0(z) & & \\ & P_1(z) & \\ & & P_2(z) \end{bmatrix} \underbrace{\begin{bmatrix} 1 & 1 & 1 \\ 1 & W_3^{-1} & W_3^{-2} \\ 1 & W_3^{-2} & W_3^{-1} \end{bmatrix}}_{F^*}, \end{aligned} \quad (2.17)$$

that is, a product of a diagonal matrix of prototype filter polyphase components and the conjugated DFT matrix (3.161a). According to Theorem 2.1, this filter bank implements an FIR biorthogonal expansion if and only if  $\Phi_p(z)$  is a monomial. So,

$$\det(\Phi_p(z)) = \prod_{j=0}^{N-1} P_j(z) \underbrace{\det(F^*)}_{1/\sqrt{N}} = \sqrt{N} \prod_{j=0}^{N-1} P_j(z), \quad (2.18)$$

<sup>13</sup>This result is known as the Balian-Low theorem in the continuous-domain setting, see Section 5.4.1.

is a monomial if and only if each polyphase component is; in other words, each polyphase component of  $P(z)$  has exactly one nonzero term, or,  $P(z)$  has  $N$  nonzero coefficients (one from each polyphase component).

While the above theorem is a negative result in general, the proof shows the factorization (2.17) that can be used to derive a fast algorithm, shown in Section 2.5 (the same factorization is used in Solved Exercise ?? to derive the relationship between the modulation and polyphase matrices). Rewriting (2.17) for general  $N$ , as well as the analysis polyphase version of it,

$$\Phi_p(z) = \text{diag}([P_0(z), P_1(z), \dots, P_{N-1}(z)]) F^*, \quad (2.19a)$$

$$\tilde{\Phi}_p(z) = \text{diag}([\tilde{G}_{0,0}(z), \tilde{G}_{0,1}(z), \dots, \tilde{G}_{0,N-1}(z)]) F^*, \quad (2.19b)$$

this filter bank implements a basis expansion if and only if

$$\text{diag}([P_0(z), \dots, P_{N-1}(z)]) \text{diag}([\tilde{G}_{0,0}(z), \dots, \tilde{G}_{0,N-1}(z)])^* = z^{-k} N I,$$

a more constrained condition than the general one of  $\Phi_p(z)\tilde{\Phi}_p^*(z) = z^{-k}I$  ( $N$  appears here since we are using the unnormalized version of  $F$ ). We also see exactly what the problem is in trying to have an orthogonal complex exponential-modulated filter bank with filters of length longer than  $N$ : If the filter bank were orthogonal, then  $\tilde{G}_0(z) = G_0(z^{-1}) = P(z^{-1})$ , and the above would reduce to

$$\begin{aligned} \Phi_p(z)\Phi_p^*(z^{-1}) &= \text{diag}([P_0(z), P_1(z), \dots, P_{N-1}(z)]) F^* \\ &\quad F \text{diag}([P_0(z^{-1}), P_1(z^{-1}), \dots, P_{N-1}(z^{-1})])^* \\ &= N \text{diag}([P_0(z)P_0(z^{-1}), P_1(z)P_1(z^{-1}), \dots, P_{N-1}(z)P_{N-1}(z^{-1})]) \\ &= N I, \end{aligned}$$

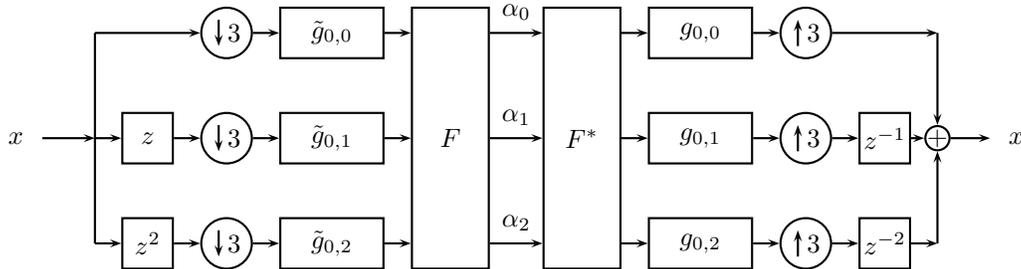
possible with FIR filters if and only if each polyphase component  $P_j(z)$  of the prototype filter  $P(z)$  were exactly of length 1 (we assumed the prototype to be real). Figure 2.4 depicts a complex exponential-modulated filter bank with  $N = 3$  channels. Solved Exercise ?? explores relationships between various matrix representations of a 3-channel complex exponential-modulated filter bank.

### 2.3.2 Application to Power Spectral Density Estimation

**[NOTE: This is a total mess. Not clear what is deterministic, what stochastic. We say we talk about periodograms and put a footnote explaining the difference between periodograms and spectrograms, but then continue on to explain something that sounds more like a spectrogram. Needs serious scrutiny.]** We now discuss the computation of periodograms, a widely used application of complex exponential-modulated filter banks.<sup>14</sup>

Given a discrete stochastic process, there exist various ways to estimate its autocorrelation. In *Chapter 3*, we have seen that, for a discrete WSS process, there

<sup>14</sup>The terms periodogram and spectrogram should not be confused with each other: the former computes the estimate of the power spectral density of a sequence, while the latter shows the dependence of the power spectral density on time.



**Figure 2.4:** A 3-channel analysis/synthesis complex exponential-modulated filter bank, with analysis filters  $\tilde{G}_i(z) = \tilde{G}_0(W_3^i z)$  and synthesis filters  $G_i(z) = G_0(W_3^i z) = P(W_3^i z)$ .  $F$  and  $F^*$  are unnormalized DFT matrices (thus  $3x$  at the output) and are implemented using FFTs.

is a direct link between the autocorrelation of a discrete stochastic process and the power spectral density, given by (3.239). However, when the process is changing over time, and we are interested in local behavior, we need a local estimate of the autocorrelation, and therefore, a local power spectral density. We thus need a local Fourier transform, by windowing the sequence appropriately, and squaring the Fourier coefficients to obtain a local power spectral density.

**Block-Based Power Spectral Density Estimation** A straightforward way to estimate local power spectral density is simply to cut the sequence into adjacent, but nonoverlapping, blocks of size  $M$ ,

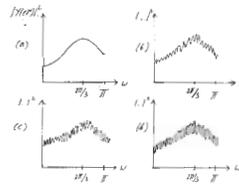
$$[\dots x_{-1} \boxed{x_0} x_1 \dots] = [\dots \underbrace{x_{nM} \dots x_{nM+M-1}}_{b_n} \underbrace{x_{(n+1)M} \dots x_{(n+1)M+M-1}}_{b_{n+1}} \dots], \tag{2.20}$$

with  $b_n$  the  $n$ th block of length  $M$ , and then take a length- $M$  DFT of each block,

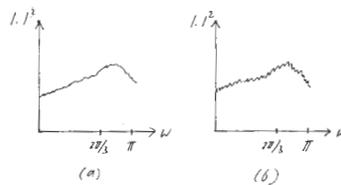
$$B_n = F b_n, \tag{2.21}$$

with  $F$  from (3.161a). Squaring the magnitudes of the elements of  $B_n$  leads to an approximation of a local power spectral density, known as a *periodogram*.

While this method is simple and computationally attractive (using an order  $O(\log M)$  operations per input sample), it has a major drawback we show through a simple example, when the sequence is white noise, or  $x_n$  is i.i.d. with variance  $\sigma_x^2$ . Since  $F$  is a unitary transform (within scaling), or, a rotation in  $M$  dimensions, the entries of  $B_n$  are i.i.d. with variance  $\sigma_x^2$ , independently of  $M$  (see Exercise ??). The power spectral density is a constant, but while the resolution increases with  $M$ , the variance does not diminish.



**Figure 2.5:** Power spectral density from (2.23). (a) Theoretical, as well as local estimates computed using (2.21) on the blocked version of the source  $y_n$ , with blocks of length (b)  $M = 64$ , (c)  $M = 256$  and (d)  $M = 1024$ , respectively.



**Figure 2.6:** Averaged power spectral density from (2.25). The theoretical power spectral density is the same as in Figure 2.5(a). (a) Average of 16 blocks of length 64. (b) Average of 4 blocks of length 256.

**EXAMPLE 2.3 (BLOCK-BASED POWER SPECTRAL DENSITY ESTIMATION)** Consider a source generated by filtering white Gaussian noise with a causal filter

$$H(z) = \frac{1 - \alpha z^{-1}}{(1 - 2\beta \cos \omega_0 z^{-1} + \beta^2 z^{-1})}, \quad \text{ROC} = \{z \mid |z| > \frac{1}{\beta}\}, \quad (2.22)$$

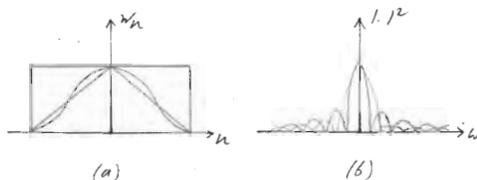
where  $\alpha, \beta$  are real and  $1 < \beta < \infty$ . This filter has poles at  $(1/\beta)e^{\pm j\omega_0}$  and zeroes at  $(1/\alpha)$  and  $\infty$ . The power spectral density of  $y = h * x$  is

$$A_y(e^{j\omega}) = \frac{|1 - \alpha e^{-j\omega}|^2}{|1 - 2\beta \cos \omega_0 e^{-j\omega} + \beta^2 e^{-j2\omega}|^2}, \quad (2.23)$$

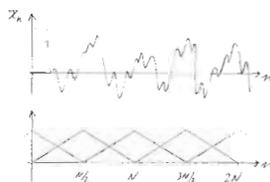
plotted in Figure 2.5(a) for  $\alpha = 1.1$ ,  $\beta = 1.1$  and  $\omega_0 = 2\pi/3$ . Figures 2.5 (b), (c) and (d) show the power spectral density calculated using (2.21) on the blocked version of  $y_n$ , with blocks of length  $M = 64$ , 256 and 1024. While the shape of the power spectral density can be guessed, the variance does indeed not diminish.

**Averaged Block-Based Power Spectral Density Estimation** When the sequence is stationary, the obvious fix is to average several power spectra. Calling  $A_{n,k}$  the block-based power spectral density, or, from (2.21)

$$A_{n,k} = |B_{n,k}|^2 = |(Fb_n)_k|^2, \quad (2.24)$$



**Figure 2.7:** Rectangular, triangle and Hamming windows of length  $M = 31$ , centered at the origin. (a) Time-domain sequences. (b) DTFT magnitude responses (in  $dB$ ).



**Figure 2.8:** Windowing with 50% overlap using a triangle window.

we can define an averaged local power spectrum by summing  $K$  successive ones,

$$A_{n,k}^{(K)} = \frac{1}{K} \sum_{n=0}^{K-1} A_{n,k}, \quad (2.25)$$

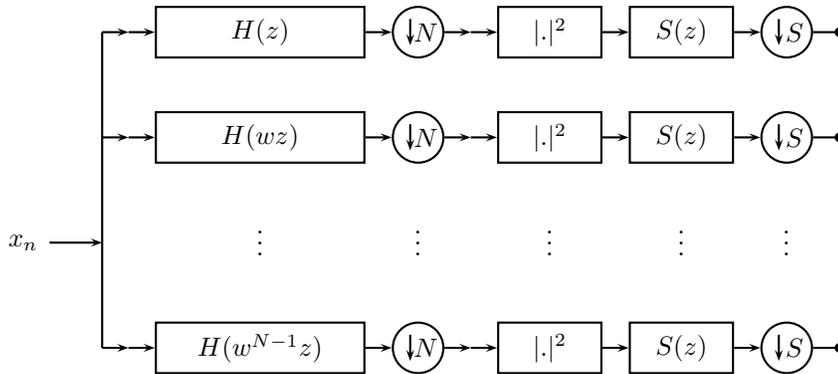
known as an *averaged periodogram*. Exercise ?? shows that the variance of  $A_{n,k}^{(K)}$  is about  $1/K$  the variance of  $A_{n,k}$ . Given a length- $L$  ( $L = KM$ ) realization of a stationary process, we can now vary  $M$  or  $K$  to achieve a trade-off between spectral resolution (large  $M$ ) and small variance (large  $K$ ).

#### EXAMPLE 2.4 (AVERAGED BLOCK-BASED POWER SPECTRAL DENSITY ESTIMATION)

Continuing Example 2.3, we now have a realization of length  $L = 1024$ , but would like to reduce the variance of the estimate by averaging. While any factorization of 1024 into  $K$  blocks of length  $1024/K$ , for  $K = 2^i$ ,  $i = 0, 1, \dots, 10$ , is possible, too many blocks lead to a poor frequency resolution ( $K = 1$  was shown in Figure 2.5 (d)). We consider two intermediate cases, 16 blocks of length 64 and 4 blocks of length 256, shown in Figure 2.6(a) and (b). These should be compared to Figure 2.5 (b) and (c), where the same block size was used, but without averaging.

**Estimation Using Windowing and Overlapping Blocks** In practice, both the periodogram and its averaged version are computed using a window, and possibly overlapping blocks.

We first discuss windowing. In the simplest case, the block  $b_n$  in (2.20) corresponds to applying a rectangular window from (3.13a) otherwise, shifted to location



**Figure 2.9:** Filter-bank implementation of the periodogram and averaged periodogram using a complex exponential-modulated filter bank with filters (2.16). The sampling factor  $N$  indicates the overlapping between blocks ( $N = M$ , basis;  $N \leq M$ , frame);  $|\cdot|^2$  computes the squared magnitude;  $S(z)$  computes the  $K$ -point averaging filter; and finally, the output is possibly downsampled by  $S$ . (TfBD:  $s$  should be  $S$ ,  $K$  should be  $N$ ,  $N$  should be  $M$ .)

$nM$  (most often the nonunit-norm version, so that the height of the window is 1). To smooth the boundary effects, smoother windows are used, of which many designs are possible. All windows provide a trade-off between the width of the main lobe (the breadth of the DTFT around zero, typically of the order of  $1/M$ ), and the height of the side lobes (the other maxima of the DTFT). Exercise ?? considers a few such windows and their respective characteristics. The upshot is that the rectangular window has the narrowest main but the highest side lobe, while others, such as the triangle window, have lower side but broader main lobes. Figure 2.7 shows three commonly used windows and their DTFT magnitude responses in  $dB$ .

Instead of computing the DFT on adjacent, but nonoverlapping blocks of size  $M$  as in (2.20), we can allow for overlap between adjacent blocks, for example (assume for simplicity  $M$  is even),

$$b_n = [x_{nM/2} \ x_{nM/2+1} \ x_{nM/2+2} \ \cdots \ x_{nM/2+M-1}], \quad (2.26)$$

a 50% overlap, shown in Figure 2.8, using a triangle window for illustration. In general, the windows move by  $N$  that is smaller or equal to  $M$ .

**Filter-Bank Implementation** The estimation process we just discussed has a natural filter bank implementation. Consider a prototype filter  $\tilde{g}_0 = p$  (we assume a symmetric window so time reversal is not an issue), and construct an  $M$ -channel complex exponential-modulated filter bank as in Figure 2.1 and (2.16). The prototype filter computes the windowing, and the modulation computes the DFT. With the sampling factor  $N = M$ , we get a critically sampled, complex exponential-modulated filter bank. The sampling factor  $N$  can be smaller than  $M$ , in which case the resulting filter bank implements a frame, discussed in Chapter 4 (see Figure 4.14).

## 2.3. Complex Exponential-Modulated Local Fourier Bases

73

Squaring the output computes a local approximation to the power spectral density. Averaging the output over  $K$  outputs computes the averaged periodogram, accomplished with a  $K$ -point averaging filter on each of the  $M$  filter bank outputs, or  $S(z) = (1/K) \sum_{m=0}^{K-1} z^{-m}$ . Finally, the output of the averaging filters maybe downsampled by a factor  $S \leq K$ . We have thus constructed a versatile device to compute local power spectral density, summarized in Figure 2.9, and Table 2.1.

Parameter	Filter-Bank Operation	Computes
$M$	Number of channels	Number of frequency bins (frequency resolution of the analysis)
$\tilde{G}_0(z) = P(z)$	Prototype filter	Windowing
$N$	Downsampling factor	Overlap between adjacent blocks ( $N = M$ , basis; $N < M$ , frame)
$S(z)$	Channel filter	Averaging and variance reduction

**Table 2.1:** Complex exponential-modulated filter-bank implementation of block-based power spectral density estimation.

The discussion so far focused on nonparametric spectral estimation, that is, we assumed no special structure for the underlying power spectral density. When a deterministic sequence, like a sinusoid, is buried in noise, we have a parametric spectral estimation problem, since we have prior knowledge on the underlying deterministic sequence. While the periodogram can be used here as well (with the effect of windowing now spreading the sinusoid), there exist powerful parametric estimation methods specifically tailored to this problem (see Exercise ??).

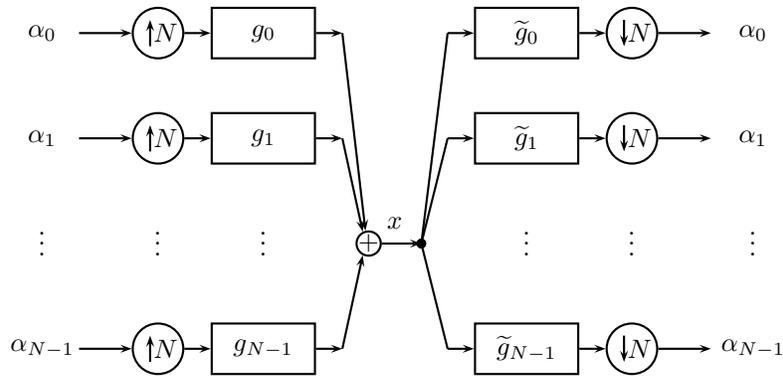
## 2.3.3 Application to Communications

Transmultiplexers<sup>15</sup> are used extensively in communication systems. They are at the heart of *orthogonal frequency division multiplexing* (OFDM), a modulation scheme popular both in mobile communications as well as in local wireless broadband systems such as IEEE 802.11 (Wi-Fi).

As we have seen in Chapter 1 and Solved Exercise ??, a transmultiplexer exchanges the order of analysis/synthesis banks. If the filters used are the complex exponential-modulated ones as we have seen above, transmultiplexers become computationally efficient. For example, consider  $N$  sequences  $\alpha_i$ ,  $i = 0, 1, \dots, N - 1$ , entering an  $N$ -channel complex exponential-modulated synthesis filter bank, to produce a synthesized sequence  $x$ . Analyzing  $x$  with an  $N$ -channel complex exponential-modulated analysis filter bank should yield again  $N$  sequences  $\alpha_i$ ,  $i = 0, 1, \dots, N - 1$  (Figure 2.10).

Similarly to what we have seen earlier, either length- $N$  filters (from the DFT (2.2)) or sinc filters, (2.6), lead to a basis. Using a good lowpass leads to approximate reconstruction (see Exercise ?? for an exploration of the end-to-end behavior).

<sup>15</sup>Such devices were used to modulate a large number of phone conversations onto large bandwidth transatlantic cables.



**Figure 2.10:** A transmultiplexer modulates  $N$  sequences into a single sequence  $x$  of  $N$ -times higher bandwidth, and analyzes it into  $N$  channels.

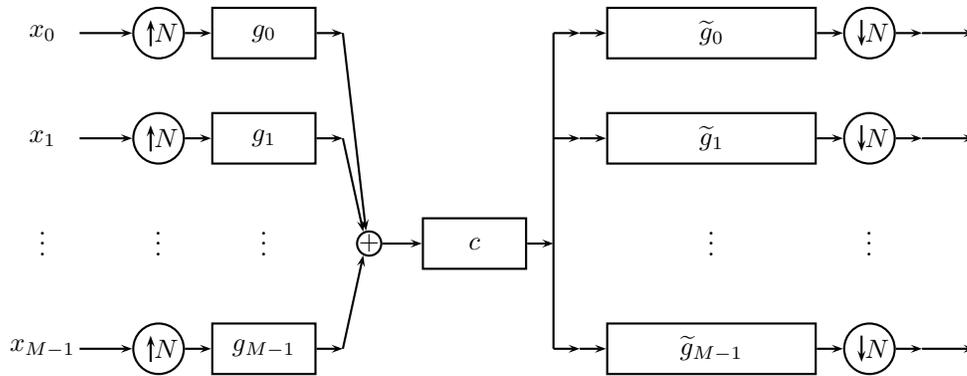
In typical communication scenarios, a desired signal is sent over a channel with impulse response  $c(t)$  (or equivalently  $c_n$  in a bandlimited and sampled case), and thus, the received signal is the input convolved with the channel. Often, the effect of the channel needs to be canceled, a procedure called *channel equalization*. If the channel is LSI, this equalization can be performed in Fourier domain, assuming  $c_n$  is known (either a priori or measured). This is a first motivation for using a Fourier-like decomposition. Moreover, as the complex sinusoids are eigensignals of LSI systems, such complex sinusoids (or approximations thereof) are good candidates for signaling over a known LSI channel. Namely, an input sinusoid of frequency  $\omega_0$  and a known amplitude  $A$  (or a set of possible amplitudes  $\{A_i\}$ ), will come out of the channel as a sinusoid, scaled by the channel frequency response at  $\omega_0$ , and perturbed by additive channel noise present at that frequency. Digital communication amounts to being able to distinguish a certain number of signaling waveforms per unit of time, given a constraint on the input (such as maximum power).

It turns out that an optimal way to communicate over an LSI channel with additive Gaussian noise is precisely to use Fourier-like waveforms. While an ideal system would require a very large number of perfect bandpass channels, practical systems use a few hundred channels (for example, 256 or 512). Moreover, instead of perfect bandpass filters (which require sinc filters), approximate bandpass filters based on finite windows are used. This time localization also allows to adapt to a changing channel, for example, in mobile communications.

The system is summarized in Figure 2.11, for  $M$  channels upsampled by  $N$ .<sup>16</sup> Such a device, allowing to put  $M$  sequences  $\{x_i\}_{i=0,\dots,M-1}$ , onto a single channel of  $N$ -times larger bandwidth, has been historically known as a transmultiplexer.

When the prototype filter is a rectangular window of length  $M$ , in the absence of channel effects, the synthesis/analysis complex exponential-modulated filter bank is perfect reconstruction. When a channel is present, and the prototype filter is a

<sup>16</sup>Again, if  $M > N$ , such a filter bank implements a frame, discussed in Chapter 4.



**Figure 2.11:** Communication over a channel using a complex exponential-modulated transmultiplexer. When  $M = N$ , it is critically sampled, while for  $M > N$ , the sequence entering the channel is redundant.

perfect lowpass filter of bandwidth  $[-\pi/M, \pi/M)$ , each bandpass channel is affected by the channel independently of the others, and can be individually equalized.

For filters with finite impulse response, one can either use a narrower-band prototype (so neighboring channels do not interact), or, use fewer than the critical number of channels, which in both cases means some redundancy is left in the synthesized sequence that enters the channel.

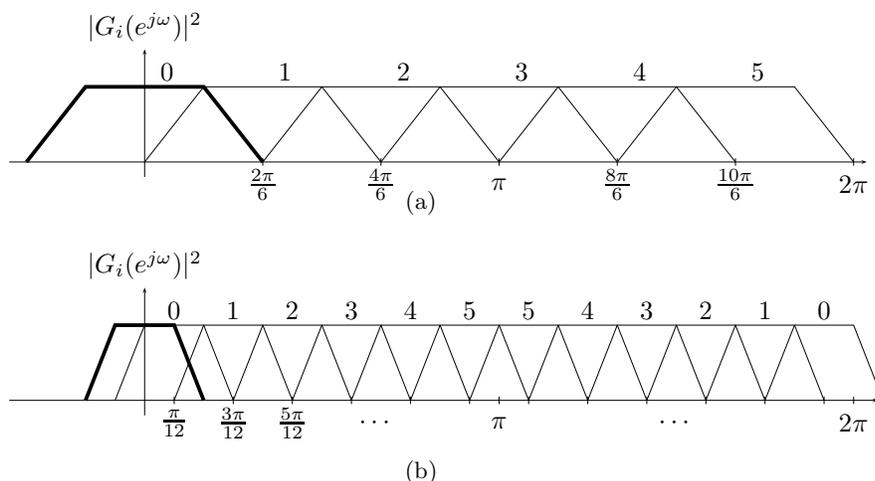
## 2.4 Cosine-Modulated Local Fourier Bases

A possible escape from the restriction imposed by the Balian-Low theorem is to replace complex-exponential modulation (multiplication by  $W_N^i = e^{-j2\pi i/N}$ ) with an appropriate cosine modulation. This has an added advantage that all filters are real if the prototype is real.

**Cosine Modulation** Given a prototype filter  $p$ , all of the filters are obtained via cosine modulation:

$$\begin{aligned}
 g_{i,n} &= p_n \cos\left(\frac{2\pi}{2N}\left(i + \frac{1}{2}\right)n + \theta_i\right) \\
 &= p_n \frac{1}{2} \left[ e^{j\theta_i} W_{2N}^{-(i+1/2)n} + e^{-j\theta_i} W_{2N}^{(i+1/2)n} \right], \\
 G_i(z) &= \frac{1}{2} \left[ e^{j\theta_i} P(W_{2N}^{(i+1/2)} z) + e^{-j\theta_i} P(W_{2N}^{-(i+1/2)} z) \right], \\
 G_i(e^{j\omega}) &= \frac{1}{2} \left[ e^{j\theta_i} P(e^{j(\omega - (2\pi/2N)(i+1/2))}) + e^{-j\theta_i} P(e^{j(\omega + (2\pi/2N)(i+1/2))}) \right],
 \end{aligned} \tag{2.27}$$

for  $i = 0, 1, \dots, N-1$ , and  $\theta_i$  is a phase factor that gives us flexibility in designing the representation; you may assume it to be 0 for now. Compare the above with (2.16) for the complex-exponential modulation; the difference is that given a real prototype



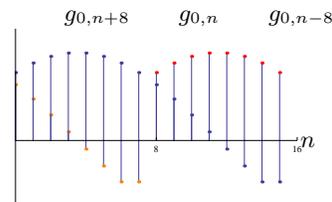
**Figure 2.12:** Complex exponential-modulated versus cosine-modulated filter bank with  $N = 6$ . (a) In the complex case, the bandwidth of the prototype is  $2\pi/6$ , and the center frequencies are  $2\pi i/6$ ,  $i = 0, 1, \dots, 5$ . (b) In the cosine case, the bandwidth of the prototype is  $2\pi/12$ , and the center frequencies are  $(2i + 1)\pi/12$ ,  $i = 0, 1, \dots, 5$ . Unlike in (a), the first filter does not correspond to the prototype, but is modulated to  $\pi/12$ .

filter, all the other filters are real. Moreover, the effective bandwidth, while  $2\pi/N$  in the case of complex-exponential modulation, is  $\pi/N$  here. The difference occurs because, the cosine-modulated filters being real, have two side lobes, which reduces the bandwidth per side lobe by two. The modulation frequencies follow from an even coverage of the interval  $[0, \pi]$  with side lobes of width  $\pi/N$ . This is illustrated in Figure 2.12 for  $N = 6$  for both complex as well as cosine modulation.

Will such a modulation lead to an orthonormal basis? One possibility is to choose an ideal lowpass filter as prototype, with support  $[-\pi/2N, \pi/2N]$ . However, as we know, this leads to a sinc-like basis with infinite and slowly-decaying impulse responses. Another solution is a block transform, such as the *discrete cosine transform (DCT)* discussed in Solved Exercise ??, too short for an interesting analysis. Fortunately, other solutions exist, with FIR filters of length longer than  $N$ , which we introduce next.

### 2.4.1 Lapped Orthogonal Transforms

The earliest example of such cosine-modulated bases was developed for filters of length  $2N$ , implying that the nonzero support of the basis sequences overlaps by  $N/2$  on each side of a block of length  $N$  (see Figure 2.13), earning them the name *lapped orthogonal transforms (LOT)*.



**Figure 2.13:** LOT for  $N = 8$ . The filters are of length  $2N = 16$ , and thus, they overlap with their nearest neighbors by  $N/2 = 4$  (only  $g_0$  is shown). Tails are orthogonal since the left half (red) is symmetric and the right half (orange) is antisymmetric.

### LOTs with a Rectangular Prototype Window

Consider  $N$  filters  $g_0, g_1, \dots, g_{N-1}$  of length  $2N$  given by (2.27). We start with a rectangular prototype window filter:

$$p_n = \frac{1}{\sqrt{N}}, \quad n = 0, 1, \dots, 2N - 1, \quad (2.28)$$

where, by choosing  $p_n = 1/\sqrt{N}$ , we ensured that  $\|g_i\| = 1$ , for all  $i$ .

As we always do, we first find conditions so that the set  $\{g_{i,n-Nk}\}$ ,  $k \in \mathbb{Z}$ ,  $i \in \{0, 1, \dots, N-1\}$ , is an orthonormal set. To do that, we prove (2.7) and (2.9).

**Orthogonality of a Single Filter** To prove that a single filter  $g_i$  is orthogonal to its shifts by  $N$ , it is enough to prove this for just two neighboring shifts (as the length of the filters is  $2N$ , see Figure 2.13). An easy way to force this orthogonality would be if the left half (tail) of the filter support (from  $0, 1, \dots, N-1$ ) were symmetric around its midpoint  $(N-1)/2$ , while the right half (tail) of the filter support (from  $N, N+1, \dots, 2N-1$ ) were antisymmetric around its midpoint  $(3N-1)/2$ . Then, the inner product  $\langle g_{i,n}, g_{i,n-N} \rangle$  would amount to the inner product of the right tail of  $g_{i,n}$  with the left tail of  $g_{i,n-N}$ , and would automatically be zero as a product of a symmetric sequence with an antisymmetric sequence. The question is whether we can force such conditions on all the filters. Fortunately, we have a degree of freedom per filter given by  $\theta_i$ , which we choose to be

$$\theta_i = -\frac{2\pi}{2N} \left(i + \frac{1}{2}\right) \frac{N-1}{2}. \quad (2.29)$$

After substituting it into (2.27), we get

$$g_{i,n} = \frac{1}{\sqrt{N}} \cos \left( \frac{2\pi}{2N} \left(i + \frac{1}{2}\right) \left(n - \frac{N-1}{2}\right) \right). \quad (2.30)$$

We now check the symmetries of the tails; for the left tail,

$$g_{i,N-n-1} = \frac{1}{\sqrt{N}} \cos \left( \frac{2\pi}{2N} \left(i + \frac{1}{2}\right) \left(-n + \frac{N-1}{2}\right) \right) \stackrel{(a)}{=} g_{i,n}, \quad (2.31a)$$

for  $n = 0, 1, \dots, N/2 - 1$ , that is, it is indeed symmetric. In the above, (a) follows from the symmetry of the cosine function. Similarly, for the right tail,

$$\begin{aligned}
 g_{i,2N-n-1} &= \frac{1}{\sqrt{N}} \cos\left(\frac{2\pi}{2N} \left(i + \frac{1}{2}\right) \left(-n + \frac{3N-1}{2}\right)\right) \\
 &\stackrel{(a)}{=} \frac{1}{\sqrt{N}} \cos\left(\frac{2\pi}{2N} \left(i + \frac{1}{2}\right) \left(n - \frac{3N-1}{2}\right)\right) \\
 &= \frac{1}{\sqrt{N}} \cos\left(\frac{2\pi}{2N} \left(i + \frac{1}{2}\right) \left(n + \frac{N+1}{2} - 2N\right)\right) \\
 &= \frac{1}{\sqrt{N}} \cos\left(\frac{2\pi}{2N} \left(i + \frac{1}{2}\right) \left(n + \frac{N+1}{2}\right) + \pi\right) \\
 &\stackrel{(b)}{=} -g_{i,N+n}, \tag{2.31b}
 \end{aligned}$$

for  $n = 0, 1, \dots, N/2 - 1$ , that is, it is indeed antisymmetric. In the above, (a) follows from the symmetry of the cosine function and (b) from  $\cos(\theta + \pi) = -\cos(\theta)$ . An LOT example with  $N = 8$  is given in Figure 2.14.

**Orthogonality of Filters** We now turn our attention to showing that all the filters are orthogonal to each other (and their shifts). As we have done in (2.27), we use (3.283), to express  $g_i$  from (2.30)

$$g_{i,n} = \frac{1}{2\sqrt{N}} \left( W_{2N}^{(i+1/2)(n-(N-1)/2)} + W_{2N}^{-(i+1/2)(n-(N-1)/2)} \right). \tag{2.32}$$

The inner product between two different filters is then:

$$\begin{aligned}
 \langle g_i, g_k \rangle &= \frac{1}{4N} \sum_{n=0}^{2N-1} \left( W_{2N}^{(i+1/2)(n-(N-1)/2)} + W_{2N}^{-(i+1/2)(n-(N-1)/2)} \right) \\
 &\quad \left( W_{2N}^{(k+1/2)(n-(N-1)/2)} + W_{2N}^{-(k+1/2)(n-(N-1)/2)} \right) \\
 &= \frac{1}{4N} \sum_{n=0}^{2N-1} \left( W_{2N}^{(i+k+1)(n-(N-1)/2)} + W_{2N}^{(i-k)(n-(N-1)/2)} + \right. \\
 &\quad \left. W_{2N}^{-(i-k)(n-(N-1)/2)} + W_{2N}^{-(i+k+1)(n-(N-1)/2)} \right).
 \end{aligned}$$

To show that the above inner product is zero, we show that each of the four sums are zero. We show it for the first sum; the other three follow the same way.

$$\frac{1}{4N} \sum_{n=0}^{2N-1} W_{2N}^{(i+k+1)(n-(N-1)/2)} = \frac{1}{4N} W_{2N}^{-(N-1)/2} \sum_{n=0}^{2N-1} (W_{2N}^{(i+k+1)})^n = 0, \tag{2.33}$$

because of the orthogonality of the roots of unity (3.285c).



where the columns of  $G_0$  and  $G_1$  are the left and right tails respectively,

$$\begin{bmatrix} G_0 \\ G_1 \end{bmatrix} = \begin{bmatrix} g_{0,0} & g_{1,0} & \cdots & g_{N-1,0} \\ g_{0,1} & g_{1,1} & \cdots & g_{N-1,1} \\ \vdots & \vdots & \ddots & \vdots \\ g_{0,N-1} & g_{1,N-1} & \cdots & g_{N-1,N-1} \\ g_{0,N} & g_{1,N} & \cdots & g_{N-1,N} \\ g_{0,N+1} & g_{1,N+1} & \cdots & g_{N-1,N+1} \\ \vdots & \vdots & \ddots & \vdots \\ g_{0,2N-1} & g_{1,2N-1} & \cdots & g_{N-1,2N-1} \end{bmatrix}. \quad (2.35)$$

Since the expansion is orthonormal,  $\Phi\Phi^T = I$ , but also  $\Phi^T\Phi = I$ , or,

$$G_0G_0^T + G_1G_1^T = I, \quad (2.36a)$$

$$G_1G_0^T = G_0G_1^T = 0, \quad (2.36b)$$

$$G_0^TG_0 + G_1^TG_1 = I, \quad (2.36c)$$

$$G_0^TG_1 = G_1^TG_0 = 0. \quad (2.36d)$$

Following the symmetry/antisymmetry of the tails, the matrices  $G_0$  and  $G_1$  have repeated rows. For example, for  $N = 4$ ,

$$G_0 = \begin{bmatrix} g_{0,0} & g_{1,0} & g_{2,0} & g_{3,0} \\ g_{0,1} & g_{1,1} & g_{2,1} & g_{3,1} \\ g_{0,1} & g_{1,1} & g_{2,1} & g_{3,1} \\ g_{0,0} & g_{1,0} & g_{2,0} & g_{3,0} \end{bmatrix} \quad \text{and} \quad G_1 = \begin{bmatrix} g_{0,4} & g_{1,4} & g_{2,4} & g_{3,4} \\ g_{0,5} & g_{1,5} & g_{2,5} & g_{3,5} \\ -g_{0,5} & -g_{1,5} & -g_{2,5} & -g_{3,5} \\ -g_{0,4} & -g_{1,4} & -g_{2,4} & -g_{3,4} \end{bmatrix}.$$

Denoting by  $\widehat{G}_0$  and  $\widehat{G}_1$  the upper halves of  $G_0$  and  $G_1$ , respectively, we can express  $G_0$  and  $G_1$  as

$$G_0 = \begin{bmatrix} I_{N/2} \\ J_{N/2} \end{bmatrix} \widehat{G}_0 \quad \text{and} \quad G_1 = \begin{bmatrix} I_{N/2} \\ -J_{N/2} \end{bmatrix} \widehat{G}_1, \quad (2.37)$$

where  $I_{N/2}$  is an  $N/2 \times N/2$  identity matrix and  $J_{N/2}$  is an  $N/2 \times N/2$  antidiagonal matrix (defined in *Section 2.B.2*). Note that  $J_N^2 = I_N$ , and that premultiplying by  $J_N$  reverses the row order (postmultiplying reverses the column order).

From the above, both  $G_0$  and  $G_1$  have rank  $N/2$ . We can easily check that the rows of  $\widehat{G}_0$  and  $\widehat{G}_1$  form an orthogonal set, with norm  $1/\sqrt{2}$ . Using all of the above, we finally unearth the special structure of the LOTs:

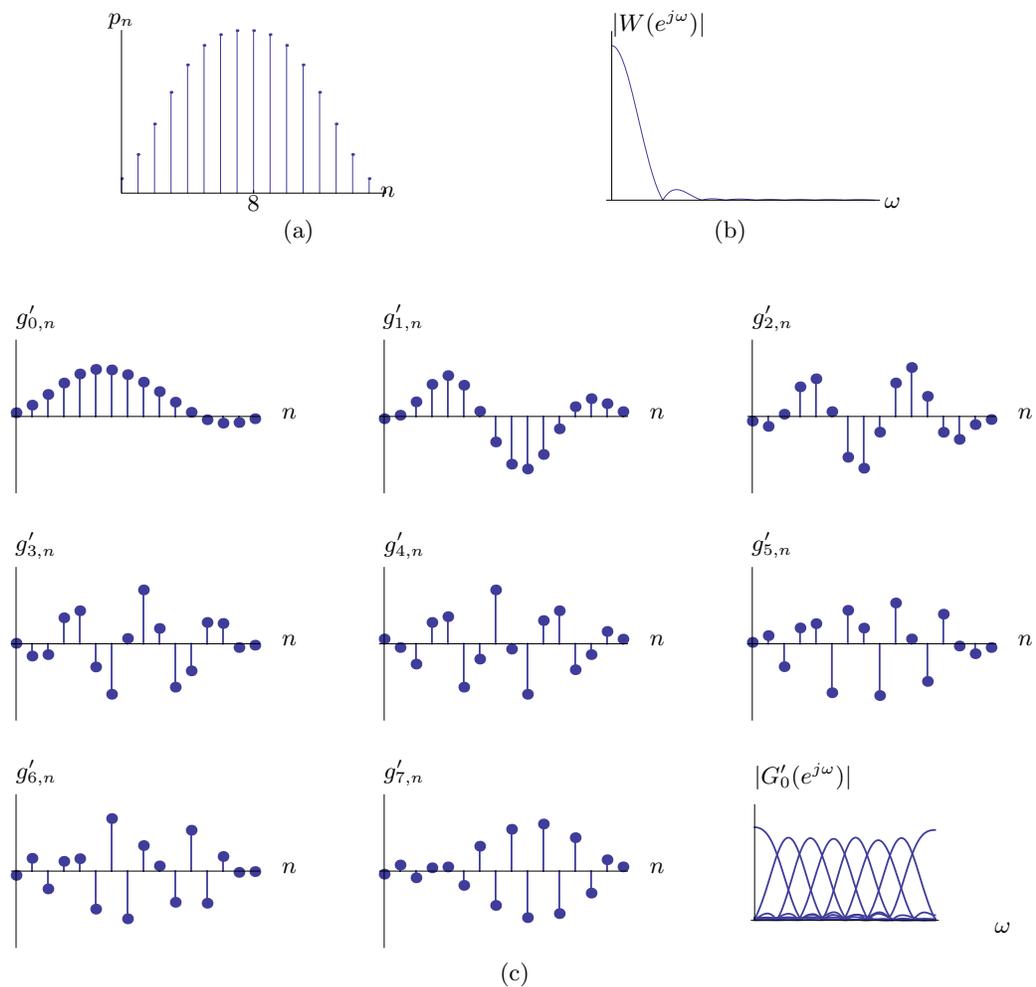
$$\begin{aligned} G_0G_0^T &= \begin{bmatrix} I_{N/2} \\ J_{N/2} \end{bmatrix} \widehat{G}_0\widehat{G}_0^T \begin{bmatrix} I_{N/2} & J_{N/2} \end{bmatrix} = \begin{bmatrix} I_{N/2} \\ J_{N/2} \end{bmatrix} \frac{1}{2}I_{N/2} \begin{bmatrix} I_{N/2} & J_{N/2} \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} I_{N/2} & J_{N/2} \\ J_{N/2} & I_{N/2} \end{bmatrix} = \frac{1}{2}(I_N + J_N), \end{aligned} \quad (2.38a)$$

$$G_1G_1^T = \frac{1}{2} \begin{bmatrix} I_{N/2} & -J_{N/2} \\ -J_{N/2} & I_{N/2} \end{bmatrix} = \frac{1}{2}(I_N - J_N), \quad (2.38b)$$

$$G_0^TG_1 = G_1^TG_0 = 0. \quad (2.38c)$$

## 2.4. Cosine-Modulated Local Fourier Bases

81



**Figure 2.15:** LOT for  $N = 8$  with a smooth, power-complementary prototype window from Table 2.2. Its (a) impulse response and (b) magnitude response. (c) The eight windowed basis sequences and their magnitude responses. Note the improved frequency resolution compared to Figure 2.14(b).

### LOTs with a Nonrectangular Prototype Window

At this point, we have  $N$  filters of length  $2N$ , but their impulse responses are simply rectangularly-windowed cosine sequences. Such a rectangular prototype window is discontinuous at the boundary, and thus not desirable; instead we aim for smooth tapering at the boundary. We now investigate whether we can window our previous solution with a smooth prototype window and still retain orthogonality.

$p_0$	$p_1$	$p_2$	$p_3$	$p_4$	$p_5$	$p_6$	$p_7$
0.0887655	2366415	0.4238081	0.6181291	0.7860766	0.9057520	0.9715970	0.9960525

**Table 2.2:** Power-complementary prototype window used in Figure 2.15. The prototype window is symmetric, so only half of the coefficients are shown.

For this, we choose a power-complementary,<sup>17</sup> real and symmetric prototype window sequence  $p$ , such that:

$$p_{2N-n-1} = p_n, \quad (2.39a)$$

$$|p_n|^2 + |p_{N-n-1}|^2 = 2, \quad (2.39b)$$

for  $n = 0, 1, \dots, N-1$ . Let

$$\begin{aligned} P_0 &= \text{diag}([p_0, p_1, \dots, p_{N-1}]), \\ P_1 &= \text{diag}([p_N, p_{N+1}, \dots, p_{2N-1}]). \end{aligned}$$

Then, (2.39) can be rewritten as

$$P_1 = J_N P_0 J_N, \quad (2.40a)$$

$$P_0^2 + P_1^2 = 2I. \quad (2.40b)$$

The counterpart to (2.35) are now the windowed impulse responses

$$\begin{bmatrix} G'_0 \\ G'_1 \end{bmatrix} = \begin{bmatrix} P_0 G_0 \\ P_1 G_1 \end{bmatrix} = \begin{bmatrix} P_0 & \\ & J_N P_0 J_N \end{bmatrix} \begin{bmatrix} G_0 \\ G_1 \end{bmatrix}. \quad (2.41)$$

Note that

$$P_0 J_N P_0 = P_1 J_N P_1. \quad (2.42)$$

These windowed impulse responses have to satisfy (2.36), or, (2.38) (substituting  $G'_i$  for  $G_i$ ). For example, we check the orthogonality of the tails (2.38c):

$$G_1'^T G_0' \stackrel{(a)}{=} G_1^T (J_N P_0 J_N) P_0 G_0 \stackrel{(b)}{=} \hat{G}_1^T \begin{bmatrix} I_{N/2} & -J_{N/2} \end{bmatrix} J_N P_0 J_N P_0 \begin{bmatrix} I_{N/2} \\ J_{N/2} \end{bmatrix} \hat{G}_0,$$

where (a) follows from (2.41), and (b) from (2.37). As the product  $J_N P_0 J_N P_0$  is diagonal and symmetric (the  $k$ th entry is  $p_k p_{N-k}$ ), we get

$$\begin{bmatrix} I_{N/2} & -J_{N/2} \end{bmatrix} J_N P_0 J_N P_0 \begin{bmatrix} I_{N/2} \\ J_{N/2} \end{bmatrix} = 0.$$

<sup>17</sup>The term *power complementary* is typically used to denote a filter whose magnitude response squared added to the frequency-reversed version of the magnitude response squared, sums to a constant, as in (3.214). We use the term more broadly here to denote a sequence whose magnitude squared added to the time-reversed version of the magnitude squared, sums to a constant.

To complete the orthogonality proof, we need to verify (2.36a) (with appropriate substitutions as above),

$$\begin{aligned} G'_0(G'_0)^T + G'_1(G'_1)^T &\stackrel{(a)}{=} P_0G_0G_0^T P_0 + P_1G_1G_1^T P_1 \\ &\stackrel{(b)}{=} \frac{1}{2}P_0(I_N + J_N)P_0 + \frac{1}{2}P_1(I_N - J_N)P_1 \\ &= \underbrace{\frac{1}{2}(P_0^2 + P_1^2)}_I + \underbrace{\frac{1}{2}(P_0J_NP_0 - P_1J_NP_1)}_0 \stackrel{(c)}{=} I, \end{aligned}$$

where (a) follows from (2.41); (b) from (2.38) and (c) from (2.40b) and (2.42). An example of a windowed LOT is shown in Figure 2.15 for  $N = 8$ . The prototype window is symmetric of length 16, with coefficients as in Table 2.2.

**Shift-Varying LOT Filter Banks** We end this section with a discussion of a variation on the theme of prototype windows, both for its importance in practice<sup>18</sup> and because it shows the same basic principles at work. Assume one wants to process a sequence with an  $N$ -channel filter bank and then switch to a  $2N$ -channel filter bank. In addition, one would like a smooth rather than an abrupt transition. Interestingly, to achieve this, it is enough for the two adjacent prototype windows to have overlapping tails that are power complementary (see Figure 2.16). Calling  $p^{(L)}$  and  $p^{(R)}$  the two prototype windows involved, then

$$|p_n^{(L)}|^2 + |p_n^{(R)}|^2 = 2$$

leads again to orthogonality of the overlapping tails of the two filter banks.

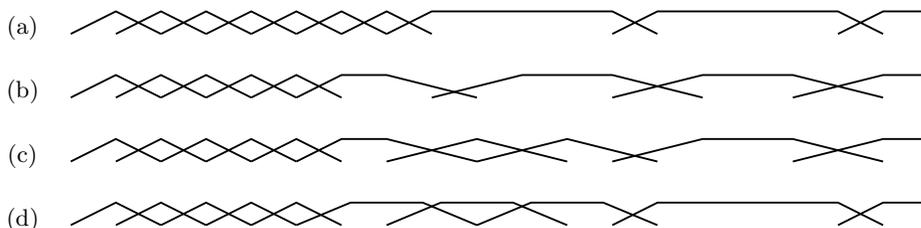
## 2.4.2 Application to Audio Compression

In Section 2.3.2, we have made numerous references to redundancy, which we will discuss in Chapter 4. In compression, the opposite is required: we want to remove the redundancy from the sequence as much as possible, and thus, typically, bases are used (in particular, orthonormal bases). While we will discuss compression in detail in Chapter 7, here, we just discuss its main theme: a small number of transform coefficients should capture a large part of the energy of the original sequence. In audio compression, the following characteristics are important:

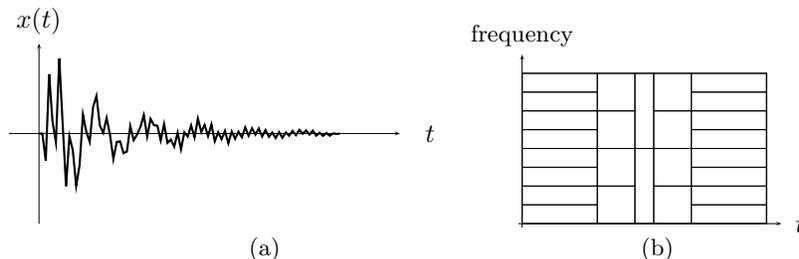
- (i) The spectrum is often harmonic, with a few dominant spectral components.
- (ii) The human auditory system exhibits a masking effect such that a large sinusoid masks neighboring smaller sinusoids.
- (iii) Sharp transitions, or attacks, are a key feature of many instruments.

It is clear that (i) and (iii) are in contradiction. The former requires long prototype windows, with local frequency analysis, while the latter requires short prototype windows, with global frequency analysis. The solution is to adapt the prototype window size, depending on the sequence content.

<sup>18</sup>Audio coding schemes use this feature extensively, as it allows for switching the number of channels in a filter bank, and consequently, the time and frequency resolutions of the analysis.



**Figure 2.16:** An example of the flexibility allowed by LOTs illustrated through different transitions from a 2-channel LOT to an 8-channel LOT. (a) Direct transition (both prototype windows have the same tails, a restriction on the 8-channel LOT as its prototype window must then be flat in the middle). (b) Transition using an asymmetric 4-channel LOT prototype window (allows for a greater flexibility in the 8-channel LOT prototype window). (c) Transition using an asymmetric 4-channel LOT prototype window and a symmetric 4-channel LOT prototype window. (d) Transition using several symmetric 4-channel LOT prototype windows (all the prototype windows are now symmetric and have the same tails).

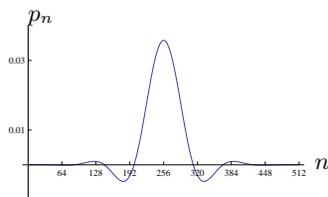


**Figure 2.17:** Analysis of an audio segment using a cosine-modulated filter bank. (a) Time-domain sequence. (b) Tiling of the time-frequency plane where shading indicates the square of the coefficient corresponding to basis sequence situated at that specific time-frequency location.

Both for harmonic analysis and for windowing, including changing the size of the filter bank (we have just seen this), we use cosine-modulated filter banks similar to those from Section 2.4, creating an adaptive tiling of the time-frequency plane, with local frequency resolution in stationary, harmonic segments, and local time resolution in transition, or, attack phases. The best tiling is chosen based on optimization procedures that try to minimize the approximation error when keeping only a small number of transform coefficients (we discuss such methods in Chapter 7). Figure 2.17 gives an example of adaptive time-frequency analysis.

For actual compression, in addition to an adaptive representation, a number of other tricks come into play, related to perceptual coding (for example, masking), quantization and entropy coding, all specifically tuned to audio compression.<sup>19</sup>

<sup>19</sup>All of this is typically done off line, that is, on the recorded audio, rather than in real time. This allows for complex optimizations, potentially using trial and error, until a satisfactory solution



**Figure 2.18:** Impulse response of the prototype window sequence modulating the cosine-modulated filter bank used in MP3.

**EXAMPLE 2.5 (FILTER BANKS USED IN AUDIO COMPRESSION)** The MPEG audio standard<sup>20</sup>, often called MP3 in consumer products, uses a 32-channel filter bank. It is not a perfect reconstruction filter bank; rather, it uses a symmetric prototype window  $p_n$  of length  $L = 2N - 1$  (in this case  $L = 511$ ) with a symmetry around  $n = 255$ . The  $i$ th filter is obtained by modulation of the prototype window as

$$g_{i,n} = p_n \cos \left( \frac{2\pi}{2N} \left( i + \frac{1}{2} \right) \left( n + \frac{N}{2} \right) \right), \quad (2.43)$$

for  $i = 0, 1, \dots, N-1$ . Comparing this to (2.30), we see that, except for the phase factor, the cosine modulation is the same. Of course, the prototype window is also different (it is of odd length hinting at the phase difference). The impulse response of the prototype window used in MP3 is displayed in Figure 2.18. Such a filter bank is called *pseudo-QMF*, because nearest neighbor aliasing is canceled as in a classical two-channel filter bank.<sup>21</sup> While aliasing from other, further bands, is not automatically canceled, the prototype is a very good lowpass suppressing it almost perfectly. The input-output relationship is not perfect (unlike for LOTs), but again, with a good prototype window, it is almost perfect.

## 2.5 Computational Aspects

The expressions for the synthesis and analysis complex exponential-modulated filter banks in (2.19a) and (2.19b) (see an illustration with three channels in Figure 2.4) lead to the corresponding fast algorithms given in Tables 2.3 and 2.4.

**Complex Exponential-Modulated Filter Banks** We now look into the cost of implementing the analysis filter bank; the cost of implementing the synthesis one is the same, as the two are dual to each other. Consider a prototype filter  $p$  of length  $L = NM$ ; each polyphase component is then of length  $N$ .

is obtained.

<sup>20</sup>While MPEG is a video standardization body, MP3 is its subgroup dealing with audio. Several different versions of audio compression, of different complexity and quality, have been developed, and the best of these, called layer III, gave the acronym MP3.

<sup>21</sup>The QMF filters are discussed in *Further Reading* of Chapter 1, as well as Exercise ??.

---

**ComplexModSynthesis**( $p, \{\alpha_0, \dots, \alpha_{N-1}\}$ )

**Input:** The prototype filter  $p$  and  $N$  channel sequences  $\{\alpha_0, \dots, \alpha_{N-1}\}$ .

**Output:** Original sequence  $x$ .

---

 Decompose prototype  $p$  into its  $N$  polyphase components  $p_{j,n} = p_{Nn+j}$ 
**for all**  $n$  **do**
*Fourier transform:* Transform channel sequences with the scaled inverse DFT

$$\begin{bmatrix} \alpha'_{0,n} \\ \alpha'_{1,n} \\ \vdots \\ \alpha'_{N-1,n} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & W_N^{-1} & W_N^{-2} & \dots & W_N^{-(N-1)} \\ 1 & W_N^{-2} & W_N^{-4} & \dots & W_N^{-2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{-(N-1)} & W_N^{-2(N-1)} & \dots & W_N^{-(N-1)^2} \end{bmatrix} \begin{bmatrix} \alpha_{0,n} \\ \alpha_{1,n} \\ \vdots \\ \alpha_{N-1,n} \end{bmatrix}$$

*Convolution:* Convolve each sequence  $\alpha'_{j,n}$  with the  $j$ th polyphase component of  $p$ 

$$\begin{bmatrix} \alpha''_{0,n} \\ \alpha''_{1,n} \\ \vdots \\ \alpha''_{N-1,n} \end{bmatrix} = \begin{bmatrix} p_{0,n} & & & \\ & p_{1,n} & & \\ & & \ddots & \\ & & & p_{N-1,n} \end{bmatrix} * \begin{bmatrix} \alpha'_{0,n} \\ \alpha'_{1,n} \\ \vdots \\ \alpha'_{N-1,n} \end{bmatrix}$$

*Inverse polyphase transform:* Upsample/interleave channel sequences to get  $x_{Nn+j} = \alpha''_{j,n}$ 
**end for**
**return**  $x$ 
**Table 2.3:** Fast implementation of a complex exponential-modulated synthesis filter bank.

First, we need to compute  $M$  convolutions, but on polyphase components of the input sequence, that is, at a rate  $M$  times slower. This is equivalent to a single convolution at full rate, or, of order  $O(N)$  operations per input sample. We then use an FFT, again at the slower rate. From (3.268), an FFT requires of the order  $O(\log_2 M)$  operations per input sample. In total, we have

$$C \sim \alpha \log_2 M + N \quad \sim \quad O(\log_2 M), \quad (2.44)$$

operations per input sample. This is very efficient, since simply taking a length- $M$  FFT for each consecutive block of  $M$  samples would already require  $\log_2 M$  operations per input sample. Thus, the price of windowing given by the prototype filter is of the order  $O(N)$  operations per input sample, or, the length of the prototype window normalized per input sample. A value for  $N$  depends on the desired frequency selectivity; a typical value can be of order  $O(\log_2 M)$ . Exercise ?? looks into the cost of a filter bank similar to those used in audio compression standards, such as MPEG from Example 2.5.

What is the numerical conditioning of this algorithm? Clearly, both the polyphase transform and the FFT are unitary maps, so the key resides in the diagonal matrix of polyphase components. While there are cases when it is unitary (such as in the block-transform case where it is the identity), it is highly dependent on the prototype window. See Exercise ?? for an exploration of this issue.

**ComplexModAnalysis**( $p, x$ )**Input:** The prototype filter  $p$  and input  $x$ .**Output:**  $N$  channel sequences  $\{\alpha_0, \dots, \alpha_{N-1}\}$ .Decompose prototype  $p$  into its  $N$  polyphase components  $p_{j,n} = p_{Nn-j}$ **for all**  $n$  **do***Polyphase transform:* Compute input sequence polyphase components  $x_{j,n} = x_{Nn+j}$ *Convolution:* Convolve polyphase components of prototype and input

$$\begin{bmatrix} \alpha'_{0,n} \\ \alpha'_{1,n} \\ \vdots \\ \alpha'_{N-1,n} \end{bmatrix} = \begin{bmatrix} p_{0,n} & & & & \\ & p_{1,n} & & & \\ & & \ddots & & \\ & & & & p_{N-1,n} \end{bmatrix} * \begin{bmatrix} x_{0,n} \\ x_{1,n} \\ \vdots \\ x_{N-1,n} \end{bmatrix}$$

*Fourier transform:* Compute channel sequences by applying the forward DFT

$$\begin{bmatrix} \alpha_{0,n} \\ \alpha_{1,n} \\ \vdots \\ \alpha_{N-1,n} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & W_N & W_N^2 & \dots & W_N^{N-1} \\ 1 & W_N^2 & W_N^4 & \dots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{(N-1)} & W_N^{2(N-1)} & \dots & W_N^{(N-1)^2} \end{bmatrix} \begin{bmatrix} \alpha'_{0,n} \\ \alpha'_{1,n} \\ \vdots \\ \alpha'_{N-1,n} \end{bmatrix}$$

**end for****return**  $\{\alpha_0, \dots, \alpha_{N-1}\}$ **Table 2.4:** Fast implementation of a complex exponential-modulated analysis filter bank.**Chapter at a Glance**

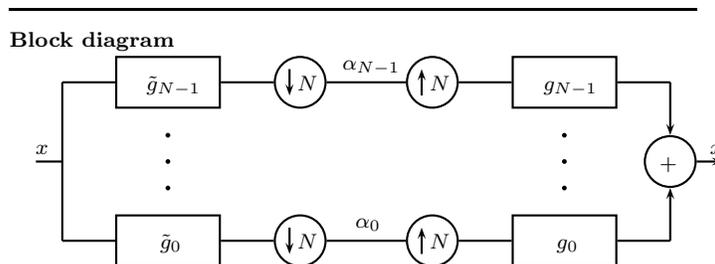
Our goal in this chapter was twofold: (1) to extend the discussion from Chapter 1 to more than two channels and associated bases; and (2) to consider those filter banks implementing local Fourier bases.

The extension to  $N$  channels, while not difficult, is a bit more involved as we now deal with more general matrices, and, in particular,  $N \times N$  matrices of polynomials. Many of the expressions are analogous to those seen Chapter 1; we went through them in some detail for orthogonal  $N$ -channel filter banks, as the biorthogonal ones are similar.

General, unstructured  $N$ -channel filter banks are rarely seen in practice; instead,  $N$ -channel modulated filter banks are widespread because (1) of their close connection to local Fourier representations, (2) computational efficiency, (modulated filter banks are implemented using FFTs), and (3) only a single prototype filter needs to be designed.

We studied uniformly-modulated filters bank using both complex exponentials and as well as cosines. The former, while directly linked to a local Fourier series (indeed, when the filter length is  $N$ , we have a blockwise DFT), is hampered by a negative result, Balian-Low theorem, which prohibits good orthonormal bases. The latter, with proper design of the prototype filter, leads to good, orthonormal, local cosine bases (LOTs). These are popular in audio and image processing using a prototype filter of length  $L = 2N$ .

To showcase their utility, we looked at the use of complex exponential-modulated filter banks in power spectral density estimation, communications (OFDM) and transmultiplexing (Wi-Fi), as well as that of cosine-modulated ones in audio compression (MP3).



**Basic characteristics**

number of channels	$M = N$
sampling factor	$N$
channel sequences	$\alpha_{i,n} \quad i = 0, 1, \dots, N - 1$

**Filters**

**Synthesis Analysis**

orthogonal filter $i$	$g_{i,n}$	$g_{i,-n}$	$i = 0, 1, \dots, N - 1$
biorthogonal filter $i$		$\tilde{g}_{i,n}$	
polyphase component $j$	$g_{i,j,n}$	$\tilde{g}_{i,j,n}$	$j = 0, 1, \dots, N - 1$

**Table 2.5:**  $N$ -channel filter bank.

Filters	Modulation	
	complex-exponential	cosine
$g_{i,n}$	$p_n W_N^{-in}$	$p_n \cos\left(\frac{2\pi}{2N}\left(i + \frac{1}{2}\right)n + \theta_i\right)$
$G_i(z)$	$P(W_N^i z)$	$\frac{1}{2} \left[ e^{j\theta_i} P(W_{2N}^{-(i+1/2)} z) + e^{-j\theta_i} P(W_{2N}^{(i+1/2)} z) \right]$
$G_i(e^{j\omega})$	$P(e^{j(\omega - (2\pi/N)i)})$	$\frac{1}{2} \left[ e^{j\theta_i} P(e^{j(\omega - (2\pi/2N)(i+1/2))}) + e^{-j\theta_i} P(e^{j(\omega + (2\pi/2N)(i+1/2))}) \right]$

**Table 2.6:** Local Fourier modulated filter bank.

**Historical Remarks**

The earliest application of a local Fourier analysis was by Dennis Gabor to the analysis of speech [40]. The idea of a local spectrum, or periodogram, was studied and refined by statisticians interested in time series of sun spots, floods, temperatures, and many others. It led to the question of windowing the data. Blackman, Tukey, Hamming, among others, worked on window designs, while the question of smoothing was studied by Bartlett and Welch, producing windowed and smoothed periodograms.

**MP3** For compression, especially speech and audio, real modulated local Fourier filter banks with perfect or almost perfect reconstruction appeared in the 1980's and 1990's. Nussbaumer, Rothweiler and others proposed pseudo-QMF filter banks, with nearly perfect

<b>Relationship between filters</b>		
Time domain	$\langle g_{i,n}, g_{j,n-Nk} \rangle_n = \delta_{i-j} \delta_k$	
Matrix domain	$D_N G_j^T G_i U_N = \delta_{i-j}$	
$z$ domain	$\sum_{k=0}^{N-1} G_i(W_N^k z) G_j(W_N^{-k} z^{-1}) = N \delta_{i-j}$	
DTFT domain	$\sum_{k=0}^{N-1} G_i(e^{j(\omega-(2\pi/N)k)}) G_j(W_N^k e^{-j\omega}) = N \delta_{i-j}$	
Polyphase domain	$\sum_{k=0}^{N-1} G_{i,k}(z) G_{j,k}(z) = \delta_{i-j}$	
<b>Basis sequences</b>	<b>Time domain</b>	<b>Frequency domain</b>
	$\{g_{i,n-2k}\}_{i=0,\dots,N-1,k \in \mathbb{Z}}$	$\{G_i(z)\}_{i=0,\dots,N-1}$
<b>Filters</b>	<b>Synthesis</b>	<b>Analysis</b>
	$g_{i,n}, G_i(z), G_i(e^{j\omega})$	$g_{i,-n}, G_i(z^{-1}), G_i(e^{-j\omega})$
<b>Matrix view</b>	<b>Basis</b>	
Time domain	$\Phi \begin{bmatrix} \dots & g_{0,n-2k} & g_{1,n-2k} & \dots & g_{N-1,n-2k} \end{bmatrix}$	
$z$ domain	$\Phi(z) \begin{bmatrix} G_0(z) & G_1(z) & \dots & G_{N-1}(z) \\ G_0(W_N z) & G_1(W_N z) & \dots & G_{N-1}(W_N z) \\ \vdots & \vdots & \ddots & \vdots \\ G_0(W_N^{N-1} z) & G_1(W_N^{N-1} z) & \dots & G_{N-1}^{N-1}(W_N z) \end{bmatrix}$	
DTFT domain	$\Phi(e^{j\omega}) \begin{bmatrix} G_0(e^{j\omega}) & G_1(e^{j\omega}) & \dots & G_{N-1}(e^{j\omega}) \\ G_0(W_N e^{j\omega}) & G_1(W_N e^{j\omega}) & \dots & G_{N-1}(W_N e^{j\omega}) \\ \vdots & \vdots & \ddots & \vdots \\ G_0(W_N^{N-1} e^{j\omega}) & G_1(W_N^{N-1} e^{j\omega}) & \dots & G_{N-1}^{N-1}(W_N e^{j\omega}) \end{bmatrix}$	
Polyphase domain	$\Phi_p(z) \begin{bmatrix} G_{0,0}(z) & G_{1,0}(z) & \dots & G_{N-1,0}(z) \\ G_{0,1}(z) & G_{1,1}(z) & \dots & G_{N-1,1}(z) \\ \vdots & \vdots & \ddots & \vdots \\ G_{0,N-1}(z) & G_{1,N-1}(z) & \dots & G_{N-1,N-1}(z) \end{bmatrix}$	
<b>Constraints</b>	<b>Orthogonality relations</b>	<b>Perfect reconstruction</b>
Time domain	$\Phi^* \Phi = I$	$\Phi \Phi^* = I$
$z$ domain	$\Phi(z^{-1})^* \Phi(z) = I$	$\Phi(z) \Phi^*(z^{-1}) = I$
DTFT domain	$\Phi^*(e^{-j\omega}) \Phi(e^{j\omega}) = I$	$\Phi(e^{j\omega}) \Phi^*(e^{-j\omega}) = I$
Polyphase domain	$\Phi_p^*(z^{-1}) \Phi_p(z) = I$	$\Phi_p(z) \Phi_p^*(z^{-1}) = I$

**Table 2.7:**  $N$ -channel orthogonal filter bank

reconstruction, frequency selective filters and high computational efficiency. This type of filter bank is used today in most audio coding standards, such as MP3. A different approach, leading to shorter filters and LOTs, was championed by Malvar, Princen and Bradley, among others. These are popular in image processing, where frequency selectivity is not as much of a concern.

**Wi-Fi** Frequency division multiplexing has been a popular communications method since the 1960's, and its digital version led to complex exponential-modulated transmultiplexers with FFTs, as proposed by Bellanger and co-workers. That perfect transmultiplexing is possible was pointed out by Vetterli. Multicarrier frequency signaling, which relies on efficient complex exponential-modulated transmultiplexers is one of the main communications methods, with orthogonal frequency division multiplexing (OFDM) being at the heart of many standards (for example, Wi-Fi, 802.11).



### Further Reading

**Books and Textbooks** For a general treatment of  $N$ -channel filter banks, see the books by Vaidyanathan [98], Vetterli and Kovačević [106], Strang and Nguyen [87], among others. For modulated filter banks, see [98] as well as Malvar's book [61], the latter with a particular emphasis on LOTs. For a good basic discussion of periodograms, see Porat's book on signal processing [68], while a more advanced treatment of spectral estimation can be found in Porat's book on statistical signal processing [67], and Stoica and Moses' book on spectral estimation [86].

**Design of  $N$ -Channel Filter Banks** General  $N$ -channel filter bank designs were investigated in [96,97]. The freedom in design offered by more channels shows in examples such as linear-phase, orthogonal FIR solutions, not possible in the two-channel case [84,108].

## Chapter 3

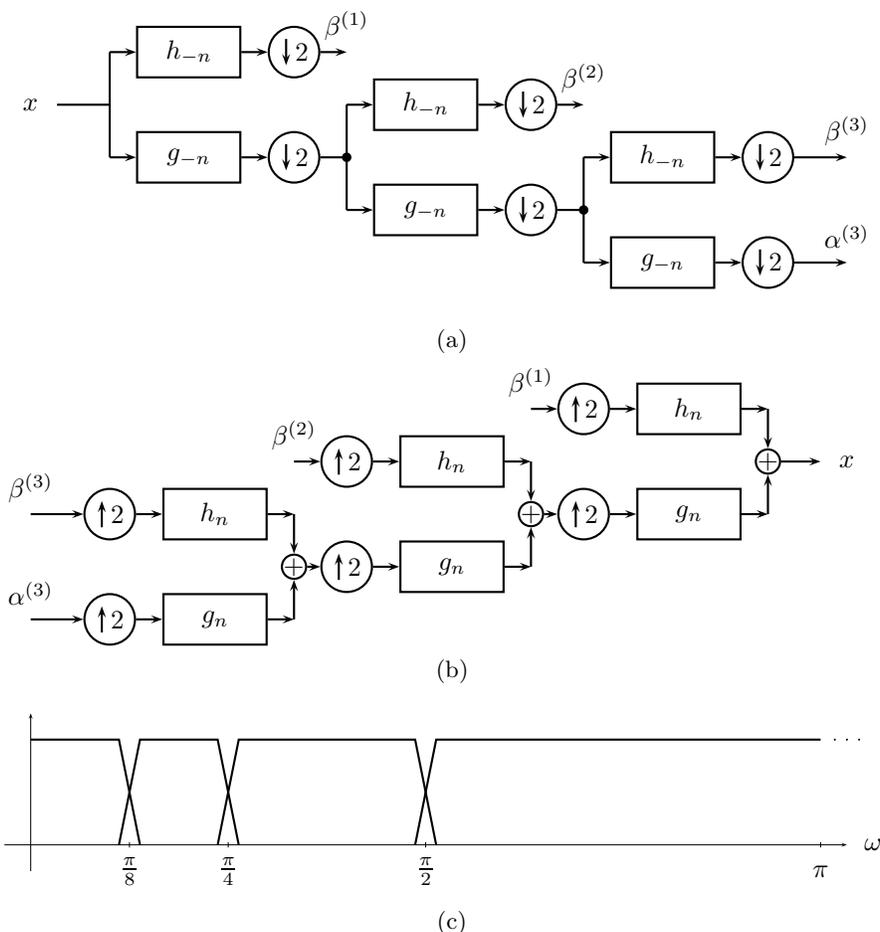
# Wavelet Bases on Sequences

## Contents

<b>3.1</b>	<b>Introduction</b>	<b>93</b>
<b>3.2</b>	<b>Tree-Structured Filter Banks</b>	<b>99</b>
<b>3.3</b>	<b>Orthogonal Discrete Wavelet Transform</b>	<b>106</b>
<b>3.4</b>	<b>Biorthogonal Discrete Wavelet Transform</b>	<b>112</b>
<b>3.5</b>	<b>Wavelet Packets</b>	<b>114</b>
<b>3.6</b>	<b>Computational Aspects</b>	<b>116</b>
	<b>Chapter at a Glance</b>	<b>118</b>
	<b>Historical Remarks</b>	<b>119</b>
	<b>Further Reading</b>	<b>119</b>

If the projection of the signal onto two subspaces is advantageous, projecting onto more subspaces might be even better. These projections onto multiple subspaces are implemented via multichannel filter banks, which come in various flavors: For example, there are direct multichannel filter banks, with  $N$  filters covering the entire spectrum, their outputs downsampled by  $N$ , covered in Chapter 2. There are also tree-structured multichannel filter banks, where a two-channel filter bank from Chapter 1 is used as a building block for more complex structures. While we will discuss arbitrary tree structures later in this chapter, most of the chapter deals with a particularly simple one that has some distinguishing features, both from mathematical as well as practical points of view. This elementary tree structure recursively splits the coarse space into ever coarser ones, yielding, in signal processing parlance, an *octave-band* filter bank. The input spectrum (subspace) from 0 to  $\pi$  is cut into a highpass part from  $\pi/2$  to  $\pi$ , with the remainder cut again into  $\pi/4$  to  $\pi/2$  and a new remainder from 0 to  $\pi/4$ , and so on. As an example, performing the split three times leads to the following 4-channel spectral division:

$$\left[0, \frac{\pi}{8}\right), \left[\frac{\pi}{8}, \frac{\pi}{4}\right), \left[\frac{\pi}{4}, \frac{\pi}{2}\right), \left[\frac{\pi}{2}, \pi\right),$$



**Figure 3.1:** A two-channel orthogonal filter bank iterated three times to obtain one coarse subspace with support  $[0, \pi/8)$ , and three bandpass subspaces. (a) Analysis filter bank. (b) Synthesis filter bank. (c) The corresponding frequency division.

yielding a lowpass (coarse) version and three bandpass (detail) versions, where each corresponds to an octave of the initial spectrum, shown in Figure 3.1(c).<sup>22</sup>

Such an unbalanced tree-structured filter bank shown in Figure 3.1 is a central concept both in filter banks as well as wavelets. Most of this chapter is devoted to its study, properties, and geometrical interpretation. In wavelet parlance, when the

<sup>22</sup>Another interpretation of octave-band filter banks is that the bandpass channels have constant relative bandwidth. For a bandpass channel, its relative bandwidth  $Q$  is defined as its center frequency divided by its bandwidth. In the example above, the channels go from  $\pi/2^{i+1}$  to  $\pi/2^i$ , with the center frequency  $3\pi/2^{i+2}$  and bandwidth  $\pi/2^{i+1}$ . The relative bandwidth  $Q$  is then  $3/2$ . In classic circuit theory, the relative bandwidth is called the  $Q$ -factor, and the filter bank above has constant- $Q$  bandpass channels.

lowpass filter is designed appropriately, the filter bank computes a *discrete wavelet transform (DWT)*. Even more is true: the same construction can be used to derive continuous-time wavelet bases, and the filter bank leads to an algorithm to compute wavelet series coefficients, the topic of Chapter 6.

### 3.1 Introduction

The iterated structure from Figure 3.1(a) clearly performs only the analysis operation. Given what we have learned so far, the channel sequences  $\beta^{(1)}$ ,  $\beta^{(2)}$ ,  $\beta^{(3)}$ ,  $\alpha^{(3)}$  compute projection coefficients onto some, yet unidentified, subspaces; it is left to establish which expansion has these as its projection/transform coefficients. Moreover, we should be able to then express the entire expansion using filter banks as we have done in Chapter 1. It is not difficult to see, that the synthesis filter bank corresponding to the analysis one from Figure 3.1(a), is the one in Figure 3.1(b). Every analysis two-channel block in Figure 3.1(a) has a corresponding synthesis two-channel block in Figure 3.1(b). We can thus use the whole machinery from Chapter 1 to study such an iterated structure. Moreover, the example with  $J = 3$  levels can be easily generalized to an arbitrary number of levels.

As we have done throughout the book, we introduce the main concepts of this chapter through our favorite example—Haar. Building upon the intuition we develop here, generalizations will come without surprise in the rest of the chapter.

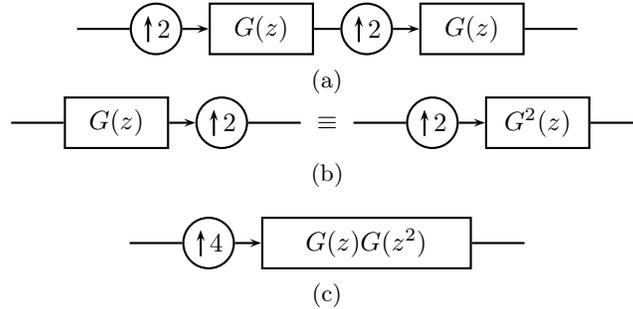
#### Implementing a Haar DWT Expansion

We start with a 3-level iterated filter bank structure as in Figure 3.1, where the two-channel filter bank block is the Haar orthogonal filter bank from Table 1.8, with synthesis filters

$$G(z) = \frac{1}{\sqrt{2}}(1 + z^{-1}), \quad H(z) = \frac{1}{\sqrt{2}}(1 - z^{-1}).$$

**Equivalent Filters** As mentioned earlier, we now have four channel sequences,  $\beta^{(1)}$ ,  $\beta^{(2)}$ ,  $\beta^{(3)}$ ,  $\alpha^{(3)}$ , and thus, we should be able to represent the tree structure from Figure 3.1 as a 4-channel filter bank, with four channel filters and four samplers. This is our aim now.

We first consider the channel sequence  $\alpha^{(3)}$  and its path through the lower branches of the first two filter banks, depicted in Figure 3.2(a). In part (b) of the same figure, we use one of the identities on the interchange of multirate operations and filtering we saw in *Chapter 3, Figure 3.28*, to move the first filter  $G(z)$  across the second upsampler, resulting in part (c) of the figure. In essence, we have compacted the sequence of steps “upsampler by 2—filter  $G(z)$ —upsampler by 2—filter  $G(z)$ ” into a sequence of steps “upsampler by 4—equivalent filter  $G^{(2)}(z) = G(z)G(z^2)$ ”. We can now iteratively continue the process by taking the equivalent filter and passing it across the third upsampler along the path of the lower branch in the last (rightmost) filter bank, resulting in a single branch with a single upsampler by 8 followed by a single equivalent filter  $G^{(3)}(z) = G(z)G(z^2)G(z^4)$ , resulting in the lowest branch of Figure 3.3.



**Figure 3.2:** Path through the lower branches of the first two filter banks in Figure 3.1. (a) Original system. (b) Use of one of the identities on the interchange of multirate operations and filtering from *Figure 3.28* results in moving the filter across the upsampler by upsampling its impulse response. (c) Equivalent system consisting of a single upsampler by 4 followed by an equivalent filter  $G^{(2)}(z) = G(z)G(z^2)$ .

Repeating the process on the other three branches transforms the 3-level tree-structured synthesis filter bank from Figure 3.1(b) into the 4-channel synthesis filter bank from Figure 3.3, with the equivalent filters:

$$H^{(1)}(z) = H(z) = \frac{1}{\sqrt{2}}(1 - z^{-1}), \quad (3.1a)$$

$$\begin{aligned} H^{(2)}(z) &= G(z)H(z^2) = \frac{1}{2}(1 + z^{-1})(1 - z^{-2}) \\ &= \frac{1}{2}(1 + z^{-1} - z^{-2} - z^{-3}), \end{aligned} \quad (3.1b)$$

$$\begin{aligned} H^{(3)}(z) &= G(z)G(z^2)H(z^4) = \frac{1}{2\sqrt{2}}(1 + z^{-1})(1 + z^{-2})(1 - z^{-4}) \\ &= \frac{1}{2\sqrt{2}}(1 + z^{-1} + z^{-2} + z^{-3} - z^{-4} - z^{-5} - z^{-6} - z^{-7}), \end{aligned} \quad (3.1c)$$

$$\begin{aligned} G^{(3)}(z) &= G(z)G(z^2)G(z^4) = \frac{1}{2\sqrt{2}}(1 + z^{-1})(1 + z^{-2})(1 + z^{-4}) \\ &= \frac{1}{2\sqrt{2}}(1 + z^{-1} + z^{-2} + z^{-3} + z^{-4} + z^{-5} + z^{-6} + z^{-7}). \end{aligned} \quad (3.1d)$$

If we repeated the above iterative process  $J$  times instead, the lowpass equivalent filter would have the  $z$ -transform

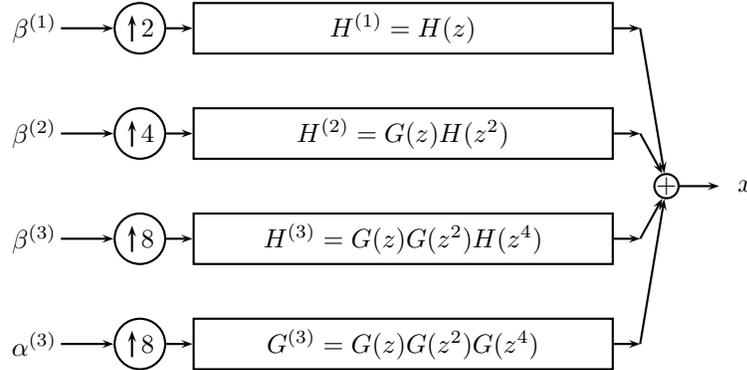
$$G^{(J)}(z) = \prod_{\ell=0}^{J-1} G(z^{2^\ell}) = \frac{1}{2^{J/2}} \sum_{n=0}^{2^J-1} z^{-n}, \quad (3.2a)$$

that is, it is a length- $2^J$  averaging filter

$$g_n^{(J)} = \frac{1}{2^{J/2}} \sum_{k=0}^{2^J-1} \delta_{n-k}, \quad (3.2b)$$

while the same-level bandpass equivalent filter follows from

$$H^{(J)}(z) = H(z^{2^{J-1}})G^{(J-1)}(z) = \frac{1}{\sqrt{2}} \left( G^{(J-1)}(z) - z^{-2^{J-1}} G^{(J-1)}(z) \right), \quad (3.2c)$$



**Figure 3.3:** Equivalent filter bank to the 3-level synthesis bank shown in Figure 3.1(b).

with the impulse response

$$h_n^{(J)} = \frac{1}{2^{J/2}} \left( \sum_{k=0}^{2^{J-1}-1} \delta_{n-k} - \sum_{k=2^{J-1}}^{2^J-1} \delta_{n-k} \right). \quad (3.2d)$$

**Basis Sequences** As we have done in Chapter 1, we now identify the resulting expansion and corresponding basis sequences. To each branch in Figure 3.3 corresponds a subspace spanned by the appropriate basis sequences. Let us start from the top. The first channel, with input  $\beta^{(1)}$ , has the equivalent filter  $h^{(1)} = h$ , just as in the basic two-channel filter bank, (1.10b), with upsampling by 2 in front. The corresponding sequences spanning the subspace  $W^{(1)}$  are (Figure 3.4(a)):

$$W^{(1)} = \overline{\text{span}}(\{h_{n-2k}^{(1)}\}_{k \in \mathbb{Z}}). \quad (3.3a)$$

The second channel, with input  $\beta^{(2)}$ , has the equivalent filter (3.1b) with upsampling by 4 in front. The corresponding sequences spanning the subspace  $W^{(2)}$  are (Figure 3.4(b)):

$$W^{(2)} = \overline{\text{span}}(\{h_{n-4k}^{(2)}\}_{k \in \mathbb{Z}}). \quad (3.3b)$$

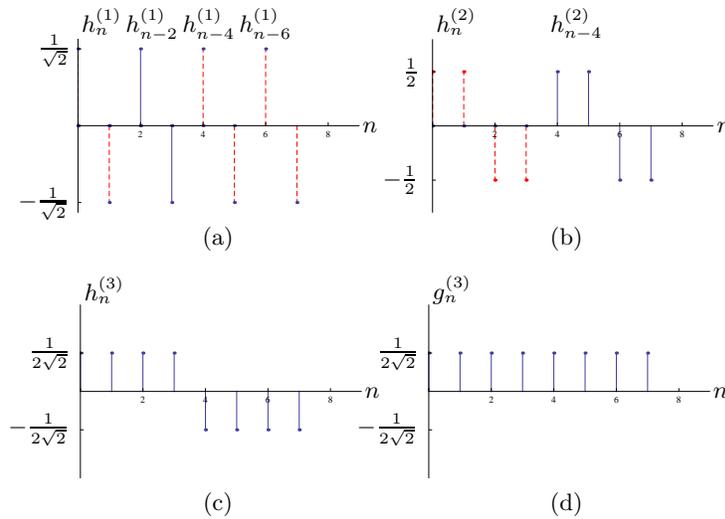
The third and fourth channels, with inputs  $\beta^{(3)}$  and  $\alpha^{(3)}$ , have the equivalent filters (3.1c), (3.1d), respectively, with upsampling by 8 in front. The corresponding sequences spanning the subspaces  $W^{(3)}$  and  $V^{(3)}$  are (Figure 3.4(c), (d)):

$$W^{(3)} = \overline{\text{span}}(\{h_{n-8k}^{(3)}\}_{k \in \mathbb{Z}}), \quad (3.3c)$$

$$V^{(3)} = \overline{\text{span}}(\{g_{n-8k}^{(3)}\}_{k \in \mathbb{Z}}). \quad (3.3d)$$

The complete set of basis sequences is thus:

$$\Phi = \{h_{n-2k}^{(1)}, h_{n-4k}^{(2)}, h_{n-8k}^{(3)}, g_{n-8k}^{(3)}\}_{k \in \mathbb{Z}}. \quad (3.3e)$$



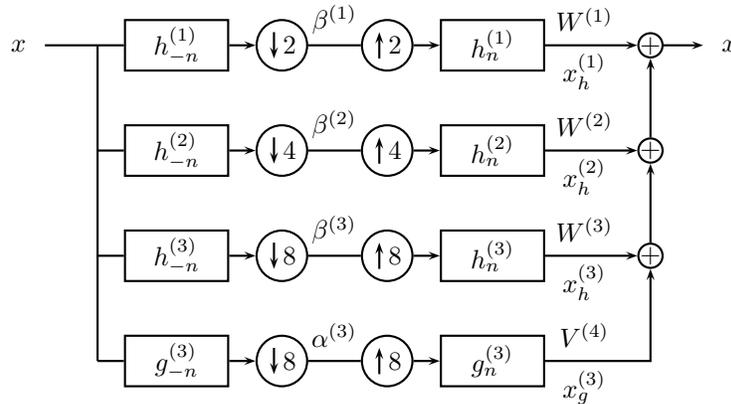
**Figure 3.4:** Discrete-time Haar basis. Eight of the basis sequences forming  $\Phi_0$ : (a) level  $\ell = 1$ ,  $h_n^{(1)}$  and three of its shifts by 2, (b) level  $\ell = 2$ ,  $h_n^{(2)}$  and one of its shifts by 4, (c) level  $\ell = 3$ ,  $h_n^{(3)}$ , and (d) level  $\ell = 3$ ,  $g_n^{(3)}$ . A basis sequence at level  $i$  is orthogonal to a basis sequence at level  $j$ ,  $i < j$ , because it changes sign over an interval where the latter is constant (see, for example, the blue basis sequences).

**Orthogonality of Basis Sequences** While we have called the above sequence basis sequences, we have not established yet that they indeed form a basis (although this is almost obvious from the two-channel filter bank discussion).

The sets spanning  $W^{(1)}$ ,  $W^{(2)}$ ,  $W^{(3)}$  and  $V^{(3)}$  are all orthonormal sets, as the sequences within those sets do not overlap. To show that  $\Phi$  is an orthonormal set, we must show that sequences in each of the above subsets are orthogonal to each other. To prove that, we have to show that  $h^{(1)}$  and its shifts by 2 are orthogonal to  $h^{(2)}$  and its shifts by 4,  $h^{(3)}$  and its shifts by 8, and  $g^{(3)}$  and its shifts by 8. Similarly, we must show that  $h^{(2)}$  and its shifts by 4 are orthogonal to  $h^{(3)}$  and its shifts by 8 and  $g^{(3)}$  and its shifts by 8, etc. For Haar filters, this can be done by observing, for example, that  $h^{(1)}$  and its shifts by 2 always overlap a constant portion of  $h^{(2)}$ ,  $h^{(3)}$  and  $g^{(3)}$ , leading to a zero inner product (see Figure 3.4). With more general filters, this proof is more involved and will be considered later in the chapter.

To prove completeness, we first introduce the matrix view of this expansion.

**Matrix View** We have already seen that while  $g^{(3)}$  and  $h^{(3)}$  move in steps of 8,  $h^{(2)}$  moves in steps of 4 and  $h^{(1)}$  moves in steps of 2. That is, during the nonzero portion of  $g^{(3)}$  and  $h^{(3)}$ ,  $h^{(2)}$  and its shift by 4 occur, as well as  $h^{(1)}$  and its shifts by 2, 4 and 6 (see Figure 3.4). Thus, as in Chapter 1, we can describe the action of



**Figure 3.5:** Projection of the input sequence onto  $V^{(3)}$ ,  $W^{(3)}$ ,  $W^{(2)}$  and  $W^{(1)}$ , respectively, and perfect reconstruction as the sum of the projections.

the filter bank via an infinite matrix:

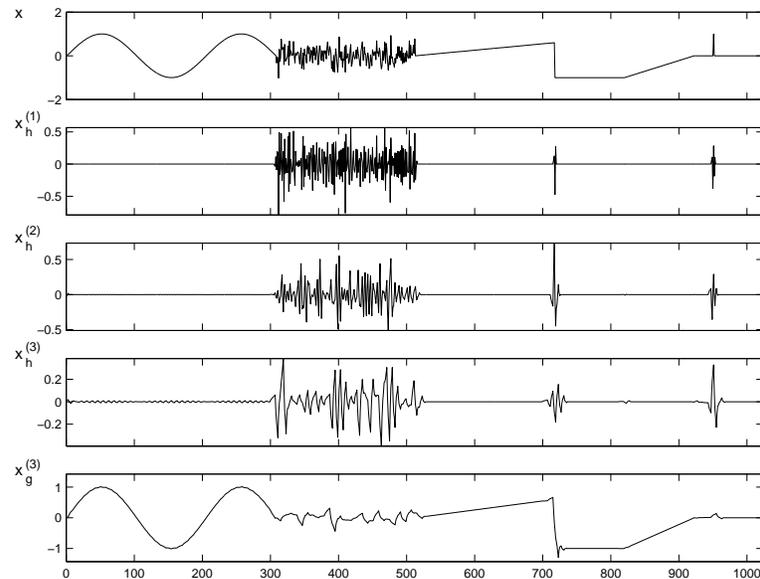
$$\Phi = \text{diag}(\Phi_0), \tag{3.4}$$

with  $\Phi_0$  as

$$\Phi_0 = \begin{bmatrix} h_n^{(1)} & h_{n-2}^{(1)} & h_{n-4}^{(1)} & h_{n-6}^{(1)} & h_n^{(2)} & h_{n-4}^{(2)} & h_n^{(3)} & g_n^{(3)} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ -1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & -1 & 0 & 1 & 1 \\ 0 & -1 & 0 & 0 & -1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & -1 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 1 & 0 & -1 & -1 & 1 \\ 0 & 0 & 0 & -1 & 0 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & & & & & & & \\ & \frac{1}{\sqrt{2}} & & & & & & \\ & & \frac{1}{\sqrt{2}} & & & & & \\ & & & \frac{1}{\sqrt{2}} & & & & \\ & & & & \frac{1}{2} & & & \\ & & & & & \frac{1}{2} & & \\ & & & & & & \frac{1}{2\sqrt{2}} & \\ & & & & & & & \frac{1}{2\sqrt{2}} \end{bmatrix}.$$

As before,  $\Phi$  is block diagonal only when the length of the filters in the original filter bank is equal to the downsampling factor, as is the case for Haar. The block is of length  $8 \times 8$  in this case, since the same structure repeats itself every 8 samples. That is,  $h^{(3)}$  and  $g^{(3)}$  repeat every 8 samples,  $h^{(2)}$  repeats every 4 samples, while  $h^{(1)}$  repeats every 2 samples. Thus, there will be 2 instances of  $h^{(2)}$  in block  $\Phi_0$  and 4 instances of  $h^{(1)}$  (see Figure 3.4). The basis sequences are the columns of the matrix  $\Phi$  at the center block  $\Phi_0$  and all their shifts by 8 (which corresponds to other blocks  $\Phi_0$  in  $\Phi$ ).  $\Phi$  is unitary, as each block  $\Phi_0$  is unitary, proving completeness for the Haar case. As we shall see, if each two-channel filter bank is orthonormal, even for longer filters, the orthonormality property will hold in general.



**Figure 3.6:** Approximation properties of the discrete wavelet transform. (a) Original sequence  $x$  with various components. The highpass approximation after the (b) first iteration  $x_h^{(1)}$ , (c) second iteration  $x_h^{(2)}$ , and (d) the third iteration  $x_h^{(3)}$ . (e) The lowpass approximation  $x_g^{(3)}$ .

**Projection Properties** In summary, the 3-level iterated two-channel filter bank from Figure 3.1, splits the original space  $\ell^2(\mathbb{Z})$  into four subspaces:

$$\ell^2(\mathbb{Z}) = V^{(3)} \oplus W^{(3)} \oplus W^{(2)} \oplus W^{(1)},$$

given in (3.3a)–(3.3d), again, a property that will hold in general. Figure 3.5 illustrates this split, where  $x_g^{(3)}$  denotes the projection onto  $V^{(3)}$ , and  $x_h^{(\ell)}$  denotes the projection onto  $W^{(\ell)}$ ,  $\ell = 1, 2, 3$ , while Figure 3.6 shows an example input sequence and the resulting channel sequences. The low-frequency sinusoid and the polynomial pieces are captured by the lowpass projection, white noise is apparent in all channels, and effects of discontinuities are localized in the bandpass channels.

## Chapter Outline

After this brief introduction, the structure of the chapter follows naturally. First, we generalize the Haar discussion and consider tree-structured filter banks, and, in particular, those that will lead to the DWT in Section 3.2. In Section 3.3, we study the orthogonal DWT and its properties such as approximation and projection capabilities. Section 3.4 discusses a variation on the DWT, the biorthogonal DWT, while Section 3.5 discusses another one, wavelet packets, which allow for rather arbitrary tilings of the time-frequency plane. We follow by computational aspects

in Section 3.6.

## 3.2 Tree-Structured Filter Banks

Out of the basic building blocks of a two-channel filter bank (Chapter 1) and an  $N$ -channel filter bank (Chapter 2), we can build many different representations (Figure 3.7 shows some of the options together with the associated time-frequency tilings). We now set the stage by showing how to compute equivalent filters in such filter banks, as a necessary step towards building an orthogonal DWT in the next section, biorthogonal DWT in Section 3.4, wavelet packets in Section 3.5. We will assume we iterate orthogonal two-channel filter banks only (the analysis is parallel for biorthogonal and/or  $N$ -channel filter banks), and that  $J$  times. We consider the equivalent filter along the lowpass branches separately, followed by the bandpass branches, and finally, the relationship between the lowpass and bandpass ones. While we could make the discussion more general, we consider bandpass channels to be only those iterated through lowpass branches until the last step, when a final iteration is through a highpass one, as is the case for the DWT (iterations through arbitrary combinations of lowpass and highpass branches would follow similarly).

### 3.2.1 The Lowpass Channel and Its Properties

We start with the lowpass channel iterated  $J$  times, leading to  $g^{(J)}$ . Using the same identity to move the filter past the upsampler as in Figure 3.2, a cascade of  $J$  times upsampling and filtering by  $G(z)$  leads to upsampling by  $2^J$  followed by filtering with the equivalent filter

$$G^{(J)}(z) = G(z)G(z^2)G(z^4)\dots G(z^{2^{J-1}}) = \prod_{\ell=0}^{J-1} G(z^{2^\ell}), \quad (3.5a)$$

as shown in Figure 3.8. If  $g$  is of length  $L$ , then  $g^{(J)}$  is of length

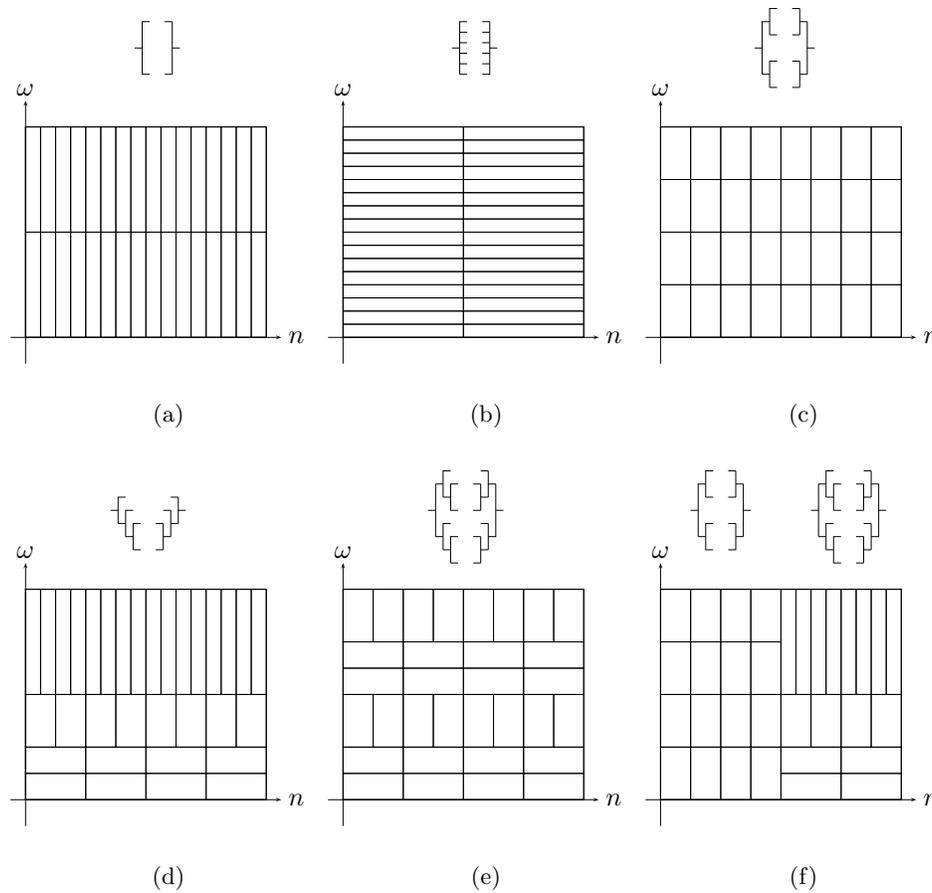
$$L^{(J)} = (L-1)(2^J-1) + 1 \leq (L-1)2^J. \quad (3.5b)$$

Moreover, we see that

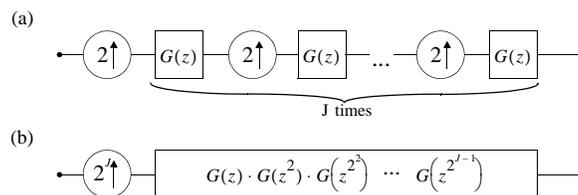
$$G^{(J)}(z) = G(z)G^{(J-1)}(z^2) = G^{(J-1)}(z)G(z^{2^{J-1}}). \quad (3.5c)$$

Some other recursive relations are given in Exercise ??.

**Orthogonality of the Lowpass Filter** We can use the orthogonality of the basic two-channel synthesis building block to show the orthogonality of the synthesis operator obtained by iterating. From this it must be true that the iterated lowpass filter is orthogonal to its translates by  $2^J$ . As in Section 1.2.1, we summarize the orthogonality properties here (as the proof is just a straightforward, albeit tedious,



**Figure 3.7:** Filter banks and variations together with the corresponding time-frequency tilings. (a) Two-channel filter bank (Chapter 1). (b)  $N$ -channel filter bank (Chapter 2). (c) The local Fourier transform filter bank (Chapter 2). (d) The DWT tree (present chapter). (e) The wavelet packet filter bank (present chapter). (f) The time-varying filter bank. [NOTE: The top-right in filter-bank representation in (f) should be as in (d), DWT.]



**Figure 3.8:** Cascade of  $J$  times upsampling and filtering. (a) Original system. (b) Equivalent system.

use of multirate operations, we leave it as Exercise ??):

$$\langle g_n^{(J)}, g_{n-2^J k}^{(J)} \rangle = \delta_k \quad \begin{array}{c} \text{Matrix View} \\ \xleftrightarrow{\text{ZT}} \\ \xleftrightarrow{\text{DTFT}} \end{array} \quad \begin{array}{l} D_{2^J} (G^{(J)})^T G^{(J)} U_{2^J} = I \\ \sum_{k=0}^{2^J-1} G^{(J)}(W_{2^J}^k z) G^{(J)}(W_{2^J}^{-k} z^{-1}) = 2^J \\ \sum_{k=0}^{2^J-1} |G^{(J)}(W_{2^J}^k e^{j\omega})|^2 = 2^J \end{array} \quad (3.6a)$$

In the matrix view, we have used linear operators (infinite matrices) introduced in *Section 3.7*: (1) downsampling by  $2^J$ ,  $D_{2^J}$ , from (3.187); (2) upsampling by  $2^J$ ,  $U_{2^J}$ , from (3.192); and (3) filtering by  $G^{(J)}$ , from (3.62). The matrix view expresses the fact that the columns of  $G^{(J)} U_{2^J}$  form an orthonormal set. The DTFT version is a version of the quadrature mirror formula we have seen in (3.214). This filter is an  $2^J$ -th-band filter (an ideal  $2^J$ -th-band filter would be bandlimited to  $|\omega| \leq \pi/2^J$ , see Figure 3.1(c) with  $J = 3$ ).

Let us check the  $z$ -transform version of the above for  $J = 2$ :

$$\begin{aligned} & \sum_{k=0}^3 G^{(2)}(W_4^k z) G^{(2)}(W_4^{-k} z^{-1}) \\ &= G^{(2)}(z) G^{(2)}(z^{-1}) + G^{(2)}(jz) G^{(2)}(-jz^{-1}) \\ & \quad + G^{(2)}(-z) G^{(2)}(-z^{-1}) + G^{(2)}(-jz) G^{(2)}(jz^{-1}) \\ & \stackrel{(a)}{=} G(z) G(z^2) G(z^{-1}) G(z^{-2}) + G(jz) G(-z^2) G(-jz^{-1}) G(-z^{-2}) \\ & \quad + G(-z) G(z^2) G(-z^{-1}) G(z^{-2}) + G(-jz) G(-z^2) G(jz^{-1}) G(-z^{-2}) \\ & \stackrel{(b)}{=} G(z^2) G(z^{-2}) \underbrace{(G(z) G(z^{-1}) + G(-z) G(-z^{-1}))}_2 \\ & \quad + G(-z^2) G(-z^{-2}) \underbrace{(G(jz) G(-jz^{-1}) + G(-jz) G(jz^{-1}))}_2 \\ & \stackrel{(c)}{=} 2 \underbrace{(G(z^2) G(z^{-2}) + G(-z^2) G(-z^{-2}))}_2 \stackrel{(d)}{=} 4, \end{aligned}$$

where (a) follows from the expression for the equivalent lowpass filter at level 2, (3.5a); in (b) we pulled out common terms  $G(z^2) G(z^{-2})$  and  $G(-z^2) G(-z^{-2})$ ; and (c) and (d) follow from the orthogonality of the lowpass filter  $g$ , (1.13). This, of course, is to be expected, because we have done nothing else but concatenate orthogonal filter banks, which we know already implement orthonormal bases, and thus, must satisfy orthogonality properties.

**Deterministic Autocorrelation of the Lowpass Filter** As we have done in Chapter 1, we rephrase the above results in terms of the deterministic autocorrelation of the filter. This is also what we use to prove (3.6a) in Exercise ?.?. The deterministic

---

**Lowpass Channel in a  $J$ -Level Octave-Band Orthogonal Filter Bank**

---

<b>Lowpass filter</b>		
Original domain	$g_n^{(J)}$	$\langle g_n^{(J)}, g_{n-2^J k}^{(J)} \rangle_n = \delta_k$
Matrix domain	$G^{(J)}$	$D_{2^J} (G^{(J)})^T G^{(J)} U_{2^J} = I$
$z$ -domain	$G^{(J)}(z) = \prod_{\ell=0}^{J-1} G(z^{2^\ell})$	$\sum_{k=0}^{2^J-1} G^{(J)}(W_{2^J}^k z) G^{(J)}(W_{2^J}^{-k} z^{-1}) = 2^J$
DTFT domain	$G^{(J)}(e^{j\omega})$	$\sum_{k=0}^{2^J-1}  G^{(J)}(W_{2^J}^k e^{j\omega}) ^2 = 2^J$
<b>Deterministic autocorrelation</b>		
Original domain	$a_n^{(J)} = \langle g_k^{(J)}, g_{k+n}^{(J)} \rangle_k$	$a_{2^J k}^{(J)} = \delta_k$
Matrix domain	$A^{(J)} = (G^{(J)})^T G^{(J)}$	$D_{2^J} A^{(J)} U_{2^J} = I$
$z$ -domain	$A^{(J)}(z) = G^{(J)}(z) G^{(J)}(z^{-1})$	$\sum_{k=0}^{2^J-1} A^{(J)}(W_{2^J}^k z) = 2^J$
DTFT domain	$A^{(J)}(e^{j\omega}) =  G^{(J)}(e^{j\omega}) ^2$	$\sum_{k=0}^{2^J-1} A^{(J)}(W_{2^J}^k e^{j\omega}) = 2^J$
<b>Orthogonal projection onto smooth space</b> $V^{(J)} = \overline{\text{span}}(\{g_{n-2^J k}^{(J)}\}_{k \in \mathbb{Z}})$		
	$x_{V^{(J)}} = P_{V^{(J)}} x$	$P_V = G^{(J)} U_{2^J} D_{2^J} (G^{(J)})^T$

---

**Table 3.1:** Properties of the lowpass channel in an orthogonal  $J$ -level octave-band filter bank.

autocorrelation of  $g^{(J)}$  is denoted by  $a^{(J)}$ .

$$\begin{array}{ccc}
 \langle g_n^{(J)}, g_{n-2^J k}^{(J)} \rangle = a_{2^J k}^{(J)} = \delta_k & \begin{array}{c} \xleftrightarrow{\text{Matrix View}} \\ \xleftrightarrow{\text{ZT}} \\ \xleftrightarrow{\text{DTFT}} \end{array} & \begin{array}{l} D_{2^J} A^{(J)} U_{2^J} = I \\ \sum_{k=0}^{2^J-1} A^{(J)}(W_{2^J}^k z) = 2^J \\ \sum_{k=0}^{2^J-1} A^{(J)}(W_{2^J}^k e^{j\omega}) = 2^J \end{array} \quad (3.6b)
 \end{array}$$

**Orthogonal Projection Property of the Lowpass Channel** We now look at the lowpass channel as a composition of four linear operators we just saw:

$$x_{V^{(J)}} = P_{V^{(J)}} x = G^{(J)} U_{2^J} D_{2^J} (G^{(J)})^T x. \quad (3.7)$$

As before, the notation is evocative of projection onto  $V^{(J)}$ , and we will now show that the lowpass channel accomplishes precisely this. Using (3.6a), we check idem-

potency and self-adjointness of  $P_{V^{(J)}}$  (*Definition 2.27*),

$$\begin{aligned} P_{V^{(J)}}^2 &= (G^{(J)}U_{2^J} \overbrace{D_{2^J}(G^{(J)})^T}^I (G^{(J)}U_{2^J} D_{2^J}(G^{(J)})^T) \\ &\stackrel{(a)}{=} G^{(J)}U_{2^J} D_{2^J}(G^{(J)})^T = P_{V^{(J)}}, \\ P_{V^{(J)}}^T &= (G^{(J)}U_{2^J} D_{2^J}(G^{(J)})^T)^T = G^{(J)}(U_{2^J} D_{2^J})^T (G^{(J)})^T \\ &\stackrel{(b)}{=} G^{(J)}U_{2^J} D_{2^J}(G^{(J)})^T = P_{V^{(J)}}, \end{aligned}$$

where (a) follows from (3.6a) and (b) from (3.194). Indeed,  $P_{V^{(J)}}$  is an orthogonal projection operator, with the range given in:

$$V^{(J)} = \overline{\text{span}}(\{g_{n-2^J k}^{(J)}\}_{k \in \mathbb{Z}}). \quad (3.8)$$

The summary of properties of the lowpass channel is given in Table 3.1.

### 3.2.2 Bandpass Channels and Their Properties

While we have only one lowpass filter, in a  $J$ -level octave-band filter bank leading to the DWT, we also have  $J$  bandpass filters, ideally, each bandlimited to  $\pi/2^{\ell+1} \leq |\omega| \leq \pi/2^\ell$ , for  $\ell = 0, 1, \dots, J-1$ , as in Figure 3.1(c) with  $J = 3$ . The analysis of an iterated filter bank constructed through arbitrary combinations of lowpass and highpass branches would follow similarly.

The filter  $H^{(\ell)}(z)$  corresponds to a branch with a highpass filter followed by  $(\ell - 1)$  lowpass filters (always with upsampling by 2 in between). The  $(\ell - 1)$ th lowpass filter branch has an equivalent filter  $G^{(\ell-1)}(z)$  as in (3.5a), preceded by upsampling by  $2^{\ell-1}$ . Passing this upsampling across the initial highpass filter changes  $H(z)$  into  $H(z^{2^{\ell-1}})$  and

$$H^{(\ell)}(z) = H(z^{2^{\ell-1}})G^{(\ell-1)}(z), \quad i = 1, \dots, J, \quad (3.9)$$

follows. The basis vectors correspond to the impulse responses and the shifts given by the upsampling factors. These upsampling factors are  $2^J$  for the lowpass branch and  $2^\ell$  for the bandpass branches.

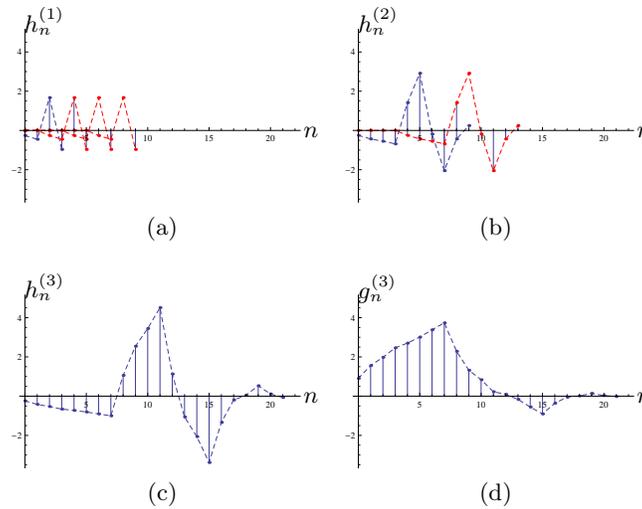
**EXAMPLE 3.1 (THE 3-LEVEL OCTAVE-BAND DAUBECHIES FILTER BANK)** We continue Example 1.3, the Daubechies filter with two zeros at  $z = -1$ :

$$G(z) = \frac{1}{4\sqrt{2}} \left[ (1 + \sqrt{3}) + (3 + \sqrt{3})z^{-1} + (3 - \sqrt{3})z^{-2} + (1 - \sqrt{3})z^{-3} \right]. \quad (3.10)$$

A 3-level octave-band filter bank has as basis sequences the impulse responses of:

$$\begin{aligned} H^{(1)}(z) &= H(z) = -z^{-3}G(-z^{-1}), \\ H^{(2)}(z) &= G(z)H(z^2), \\ H^{(3)}(z) &= G(z)G(z^2)H(z^4), \\ G^{(3)}(z) &= G(z)G(z^2)G(z^4), \end{aligned}$$

together with shifts by multiples of 2, 4, 8 and 8, respectively (see Figure 3.9).



**Figure 3.9:** Basis sequences for a 3-level octave-band filter bank based on a Daubechies orthonormal length-4 filter with two zeros at  $z = -1$ , as in (3.10). The basis sequences are  $h^{(1)}, h^{(2)}, h^{(3)}$  and  $g^{(3)}$ , together with shifts by multiples of 2, 4, 8 and 8, respectively. We show (a)  $h_n^{(1)}, h_{n-2}^{(1)}, h_{n-4}^{(1)}, h_{n-6}^{(1)}$ , (b)  $h_n^{(2)}, h_{n-4}^{(2)}$ , (c)  $h_n^{(3)}$ , and (d)  $g_n^{(3)}$ .

**Orthogonality of an Individual Bandpass Filter** Unlike in the simple two-channel case, now, each bandpass filter  $h^{(\ell)}$  is orthogonal to its shifts by  $2^\ell$ , and to all the other bandpass filters  $h^{(j)}$ ,  $\ell \neq j$ , as well as to their shifts by  $2^j$ . We expect these to hold as they hold in the basic building block, but state them nevertheless. While we could state together the orthogonality properties for a single level and across levels, we separate them for clarity. All the proofs are left for Exercise ??.

$$\begin{aligned}
 \langle h_n^{(\ell)}, h_{n-2^\ell k}^{(\ell)} \rangle = \delta_k & \quad \begin{array}{l} \text{Matrix View} \\ \xleftrightarrow{\text{ZT}} \\ \xleftrightarrow{\text{DTFT}} \end{array} \quad \begin{array}{l} D_{2^\ell} (H^{(\ell)})^T H^{(\ell)} U_{2^\ell} = I \\ \sum_{k=0}^{2^\ell-1} H^{(\ell)}(W_{2^\ell}^k z) H^{(\ell)}(W_{2^\ell}^{-k} z^{-1}) = 2^\ell \\ \sum_{k=0}^{2^\ell-1} |H^{(\ell)}(W_{2^\ell}^k e^{j\omega})|^2 = 2^\ell \end{array} \quad (3.11a)
 \end{aligned}$$

**Orthogonality of Different Bandpass Filters** Without loss of generality, let us assume that  $\ell < j$ . We summarize the orthogonality properties of the bandpass filter  $h^{(\ell)}$  and its shift by  $2^\ell$  to the bandpass filter  $h^{(j)}$  and its shift by  $2^j$ .

$$\begin{aligned}
 \langle h_{n-2^\ell k}^{(\ell)}, h_{n-2^j k}^{(j)} \rangle = 0 & \quad \begin{array}{l} \text{Matrix View} \\ \xleftrightarrow{\text{ZT}} \\ \xleftrightarrow{\text{DTFT}} \end{array} \quad \begin{array}{l} D_{2^j} (H^{(j)})^T H^{(\ell)} U_{2^\ell} = 0 \\ \sum_{k=0}^{2^\ell-1} H^{(\ell)}(W_{2^\ell}^k z) H^{(j)}(W_{2^\ell}^{-k} z^{-1}) = 0 \\ \sum_{k=0}^{2^\ell-1} H^{(\ell)}(W_{2^\ell}^k e^{j\omega}) H^{(j)}(W_{2^\ell}^k e^{-j\omega}) = 0 \end{array} \quad (3.11b)
 \end{aligned}$$

**Deterministic Autocorrelation of Individual Bandpass Filters**

$$\langle h_n^{(\ell)}, h_{n-2^\ell k}^{(\ell)} \rangle = a_{2^\ell k}^{(\ell)} = \delta_k \quad \begin{array}{l} \text{Matrix View} \\ \longleftrightarrow \\ \xrightarrow{\text{ZT}} \\ \xleftarrow{\text{DTFT}} \end{array} \quad \begin{array}{l} D_{2^\ell} A^{(\ell)} U_{2^\ell} = I \\ \sum_{k=0}^{2^\ell-1} A^{(\ell)}(W_{2^\ell}^k z) = 2^\ell \\ \sum_{k=0}^{2^\ell-1} A^{(\ell)}(W_{2^\ell}^k e^{j\omega}) = 2^\ell \end{array} \quad (3.11c)$$

**Deterministic Crosscorrelation of Different Bandpass Filters** Again without loss of generality, we assume that  $\ell < j$ .

$$\langle h_{n-2^\ell k}^{(\ell)}, h_{n-2^j k}^{(j)} \rangle = c_{2^\ell k}^{(\ell,j)} = 0 \quad \begin{array}{l} \text{Matrix View} \\ \longleftrightarrow \\ \xrightarrow{\text{ZT}} \\ \xleftarrow{\text{DTFT}} \end{array} \quad \begin{array}{l} D_{2^j} C^{(\ell,j)} U_{2^\ell} = 0 \\ \sum_{k=0}^{2^\ell-1} C^{(\ell,j)}(W_{2^\ell}^k z) = 0 \\ \sum_{k=0}^{2^\ell-1} C^{(\ell,j)}(W_{2^\ell}^k e^{j\omega}) = 0 \end{array} \quad (3.11d)$$

**Orthogonal Projection Property of Bandpass Channels** The lowpass channel computes a projection onto a space of coarse sequences spanned by  $g^{(J)}$  and its shifts by  $2^J$ . Similarly, each bandpass channel computes a projection onto a space of detail sequences spanned by each of  $h^{(\ell)}$  and its shifts by  $2^\ell$ , for  $\ell = 1, 2, \dots, J$ . That is, we have  $J$  bandpass projection operators, computing bandpass projections:

$$x_{W^\ell} = P_{W^{(\ell)}} x = H^{(\ell)} U_{2^\ell} D_{2^\ell} (H^{(\ell)})^T x, \quad (3.12)$$

for  $\ell = 1, 2, \dots, J$ . That  $P_{W^{(\ell)}}$  is an orthogonal projection operator is easy to show; follow the same path as for the lowpass filter. Each bandpass space is given by:

$$W^{(\ell)} = \overline{\text{span}}(\{h_{n-2^\ell k}^{(\ell)}\}_{k \in \mathbb{Z}}). \quad (3.13)$$

**3.2.3 Relationship between Lowpass and Bandpass Channels**

The only conditions left to show for the lowpass impulse response and its shifts by  $2^J$  and all the bandpass impulse responses and their appropriate shifts to form an orthonormal set, is the orthogonality of the lowpass and bandpass sequences. Since the proofs follow the same path as before, we again leave them for Exercise ??.

**Orthogonality of the Lowpass and Bandpass Filters**

$$\langle g_{n-2^J k}^{(J)}, h_{n-2^\ell k}^{(\ell)} \rangle = 0 \quad \begin{array}{l} \text{Matrix View} \\ \longleftrightarrow \\ \xrightarrow{\text{ZT}} \\ \xleftarrow{\text{DTFT}} \end{array} \quad \begin{array}{l} D_{2^\ell} (H^{(\ell)})^T G^{(J)} U_{2^J} = 0 \\ \sum_{k=0}^{2^\ell-1} G^{(J)}(W_{2^\ell}^k z) H^{(\ell)}(W_{2^\ell}^{-k} z^{-1}) = 0 \\ \sum_{k=0}^{2^\ell-1} G^{(J)}(W_{2^\ell}^k e^{j\omega}) H^{(\ell)}(W_{2^\ell}^k e^{-j\omega}) = 0 \end{array} \quad (3.14a)$$

**Deterministic Crosscorrelation of the Lowpass and Bandpass Filters**

$$\langle g_{n-2^J k}^{(J)}, h_{n-2^\ell k}^{(\ell)} \rangle = c_{2^\ell k}^{(J,\ell)} = 0 \quad \begin{array}{l} \text{Matrix View} \\ \longleftrightarrow \\ \xrightarrow{\text{ZT}} \\ \xleftarrow{\text{DTFT}} \end{array} \quad \begin{array}{l} D_{2^\ell} C^{(J,\ell)} U_{2^J} = 0 \\ \sum_{k=0}^{2^\ell-1} C^{(J,\ell)}(W_{2^\ell}^k z) = 0 \\ \sum_{k=0}^{2^\ell-1} C^{(J,\ell)}(W_{2^\ell}^k e^{j\omega}) = 0 \end{array} \quad (3.14b)$$

### 3.3 Orthogonal Discrete Wavelet Transform

Following our introductory Haar example, it is now quite clear what the DWT does: it produces a coarse projection coefficient  $\alpha^{(J)}$ , together with a sequence of ever finer detail projection (wavelet) coefficients  $\beta^{(1)}, \beta^{(2)}, \dots, \beta^{(J)}$ , using a  $J$ -level octave-band filter bank as a vehicle. As we have seen in that simple example, the original space is split into a sequence of subspaces, each having a spectrum half the size of the previous (octave-band decomposition). Such a decomposition is appropriate for smooth sequences with isolated discontinuities (natural images are one example of such signals; there is evidence that the human visual system processes visual information in such a manner exactly).

#### 3.3.1 Definition of the Orthogonal DWT

We are now ready to formally define the orthogonal DWT:

**DEFINITION 3.1 (ORTHOGONAL DWT)** The  $J$ -level orthogonal DWT of a sequence  $x$  is a function of  $\ell \in \{1, 2, \dots, J\}$  given by

$$\alpha_k^{(J)} = \langle x_n, g_{n-2^J k}^{(J)} \rangle_n = \sum_{n \in \mathbb{Z}} x_n g_{n-2^J k}^{(J)}, \quad k \in \mathbb{Z}, \quad (3.15a)$$

$$\beta_k^{(\ell)} = \langle x_n, h_{n-2^\ell k}^{(\ell)} \rangle_n = \sum_{n \in \mathbb{Z}} x_n h_{n-2^\ell k}^{(\ell)}, \quad \ell \in \{1, 2, \dots, J\}. \quad (3.15b)$$

The inverse DWT is given by

$$x_n = \sum_{k \in \mathbb{Z}} \alpha_k^{(J)} g_{n-2^J k}^{(J)} + \sum_{\ell=1}^J \sum_{k \in \mathbb{Z}} \beta_k^{(\ell)} h_{n-2^\ell k}^{(\ell)}. \quad (3.15c)$$

In the above, the  $\alpha^{(J)}$  are the *scaling coefficients* and the  $\beta^{(\ell)}$  are the *wavelet coefficients*.

The equivalent filter  $g^{(J)}$  is often called the *scaling sequence* and  $h^{(\ell)}$  *wavelets* (wavelet sequences),  $\ell = 1, 2, \dots, J$ ; they are given in (3.5a) and (3.9), respectively, and satisfy (3.6a)–(3.6b), (3.11a)–(3.11c), as well as (3.14a)–(3.14b).

To denote such a DWT pair, we write:

$$x_n \xleftrightarrow{\text{DWT}} \alpha_k^{(J)}, \beta_k^{(J)}, \beta_k^{(J-1)}, \dots, \beta_k^{(1)}.$$

The orthogonal DWT is implemented using a  $J$ -level octave-band orthogonal filter bank as in Figure 3.1. This particular version of the DWT is called the *dyadic* DWT as each subsequent channel has half of the coefficients of the previous one. Various generalizations are possible; for example, Solved Exercise ?? considers the DWT obtained from a 3-channel filter bank.

### 3.3.2 Properties of the Orthogonal DWT

Some properties of the DWT are rather obvious (such as linearity), while others are more involved (such as shift in time). We now list and study a few of these.

DWT properties	Time domain	DWT domain
Linearity	$ax_n + by_n$	$a \{\alpha_{x,k}^{(J)}, \beta_{x,k}^{(J)}, \dots, \beta_{x,k}^{(1)}\} + b \{\alpha_{y,k}^{(J)}, \beta_{y,k}^{(J)}, \dots, \beta_{y,k}^{(1)}\}$
Shift in time	$x_{n-2^J n_0}$	$\alpha_{k-n_0}^{(J)}, \beta_{k-n_0}^{(J)}, \beta_{k-2n_0}^{(J-1)}, \dots, \beta_{k-2^{J-1}n_0}^{(1)}$
Parseval's equality	$\ x\ ^2 = \sum_{n \in \mathbb{Z}}  x_n ^2$	$= \ \alpha^{(J)}\ ^2 + \sum_{\ell=1}^J \ \beta^{(\ell)}\ ^2$

**Table 3.2:** Properties of the DWT.

**Linearity** The DWT operator is a linear operator, or,

$$ax_n + by_n \xleftrightarrow{\text{DWT}} a \{\alpha_{x,k}^{(J)}, \beta_{x,k}^{(J)}, \dots, \beta_{x,k}^{(1)}\} + b \{\alpha_{y,k}^{(J)}, \beta_{y,k}^{(J)}, \dots, \beta_{y,k}^{(1)}\}. \quad (3.16)$$

**Shift in Time** A shift in time by  $2^J n_0$  results in

$$x_{n-2^J n_0} \xleftrightarrow{\text{DWT}} \alpha_{k-n_0}^{(J)}, \beta_{k-n_0}^{(J)}, \beta_{k-2n_0}^{(J-1)}, \dots, \beta_{k-2^{J-1}n_0}^{(1)}. \quad (3.17)$$

This property shows is that the DWT is not shift invariant; it is periodically shift varying with period  $2^J$ .

**Parseval's Equality** The DWT operator is a unitary operator and thus preserves the Euclidean norm (see (2.53)):

$$\|x\|^2 = \sum_{n \in \mathbb{Z}} |x_n|^2 = \|\alpha^{(J)}\|^2 + \sum_{\ell=1}^J \|\beta^{(\ell)}\|^2. \quad (3.18)$$

**Projection** After our Haar example, it should come as no surprise that a  $J$ -level orthogonal DWT projects the input sequence  $x$  onto one lowpass space

$$V^{(J)} = \overline{\text{span}}(\{g_{n-2^J k}^{(J)}\}_{k \in \mathbb{Z}}),$$

and  $J$  bandpass spaces

$$W^{(\ell)} = \overline{\text{span}}(\{h_{n-2^\ell k}^{(\ell)}\}_{k \in \mathbb{Z}}), \quad \ell = 1, \dots, J,$$

where  $g^{(J)}$  and  $h^{(\ell)}$  are the equivalent filters given in (3.5a) and (3.9), respectively. The input space  $\ell^2(\mathbb{Z})$  is split into the following  $(J+1)$  spaces:

$$\ell^2(\mathbb{Z}) = V^{(J)} \oplus W^{(J)} \oplus W^{(J-1)} \oplus \dots \oplus W^{(1)}.$$

**Polynomial Approximation** As we have done in Section 1.2.5, we now look at polynomial approximation properties of an orthogonal DWT,<sup>23</sup> under the assumption that the lowpass filter  $g$  has  $N \geq 1$  zeros at  $z = -1$ , as in (1.46):

$$G(z) = (1 + z^{-1})^N R(z).$$

Note that  $R(z)|_{z=-1}$  cannot be zero because of the orthogonality constraint (1.13). Remember that the highpass filter, being a modulated version of the lowpass, has  $N$  zeros at  $z = 1$ . In other words, it annihilates polynomials up to degree  $(N - 1)$  since it takes an  $N$ th-order difference of the sequence.

In the DWT, each bandpass channel annihilates finitely-supported polynomials of a certain degree, which are therefore carried by the lowpass branch. That is, if  $x$  is a polynomial sequence of degree smaller than  $N$ , the channel sequences  $\beta^{(i)}$  are all zero, and that polynomial sequence  $x$  is projected onto  $V^{(J)}$ , the lowpass approximation space:

$$x_n = n^i = \sum_{k \in \mathbb{Z}} \alpha_k^{(J)} g_{n-2^J k}^J, \quad 0 < i < N,$$

that is, the equivalent lowpass filter reproduces polynomials up to degree  $(N - 1)$ . As this is an orthogonal DWT, the scaling coefficients follow from (3.15a),

$$\alpha_k^{(J)} = \langle x_n, g_{n-2^J k}^{(J)} \rangle_n = \langle n^i, g_{n-2^J k}^{(J)} \rangle_n = \sum_{n \in \mathbb{Z}} n^i g_{n-2^J k}^{(J)}.$$

An example with the 4-tap Daubechies orthogonal filter from (3.10) is given in Figure 3.10. Part (a) shows the equivalent filter after 6 levels of iteration:  $J = 6$ ,  $G^{(6)}(z) = G(z)G(z^2)G(z^4)G(z^8)G(z^{16})G(z^{32})$  and length  $L^{(6)} = 190$  from (3.5b). Part (b) shows the reproduction of a linear polynomial (over a finite range, ignoring boundary effects).

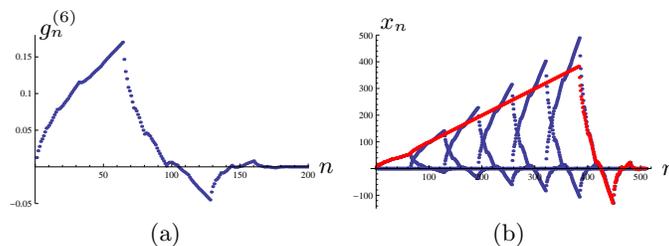
In summary, the DWT, when wavelet basis sequences have zero moments, will have very small inner products with smooth parts of an input sequence (and exactly zero when the sequence is locally polynomial). This will be one key ingredient in building successful approximation schemes using the DWT in Chapter 7.

**Characterization of Singularities** Due to its localization properties, the DWT has a unique ability to characterize singularities.

Consider a single nonzero sample in the input,  $x_n = \delta_{n-k}$ . This delayed Kronecker delta sequence now excites each equivalent filter's impulse response of length  $L^{(i)} = (L - 1)(2^i - 1) + 1$  at level  $\ell$  (see (3.5b)), which is then downsampled by  $2^i$  (see Figure 3.5 for an illustration with  $J = 3$ ). Thus, this single nonzero input creates at most  $(L - 1)$  nonzero coefficients in each channel. Furthermore, since each equivalent filter is of norm 1 and downsampled by  $2^\ell$ , the energy resulting at level  $\ell$  is of the order

$$\|\beta^{(\ell)}\|^2 \sim 2^{-\ell}.$$

<sup>23</sup>Recall that we are dealing with finitely-supported polynomial sequences, ignoring the boundary issues. If this were not the case, these sequences would not belong to any  $\ell^p$  space.



**Figure 3.10:** Polynomial reproduction, exact over a finite range. (a) Equivalent filter's impulse response after six iterations. (b) Reproduction of a linear polynomial (in red) over a finite range. We also show the underlying weighted basis sequences ( $\alpha_k^{(6)} g_{n-2^j k}^6$ ,  $k = 0, 1, \dots, 5$ ) contributing to the reproduction of the polynomial. While the plots are all discrete, they give the impression of being connected due to point density.

In other words, the energy of the Kronecker delta sequence is roughly spread across the channels according to a geometric distribution. Another way to phrase the above result is to note that as  $\ell$  increases, coefficients  $\beta^{(\ell)}$  decay roughly as

$$\beta^{(\ell)} \sim 2^{-\ell/2},$$

when the input is an isolated Kronecker delta.

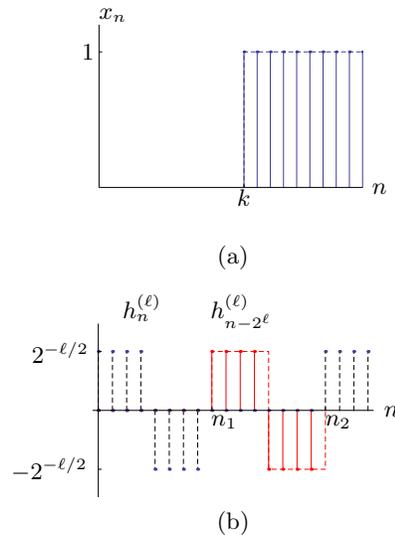
For a piecewise constant sequence, the coefficients behave instead as

$$\beta^{(\ell)} \sim 2^{\ell/2}.$$

Thus, two different types of singularities lead to different behaviors of wavelet coefficients across scales. In other words, if we can observe the behavior of wavelet coefficients across scales, we can make an educated guess of the type of singularity present in the input sequence, as we illustrate in Example 3.2. We will study this in more in detail in the continuous-time case, in Chapter 6.

**EXAMPLE 3.2 (CHARACTERIZATION OF SINGULARITIES BY THE HAAR DWT)**  
Convolution of the Kronecker delta sequence at position  $k$  with the Haar analysis filter  $h_{-n}^{(\ell)}$  in (3.2c) generates  $2^\ell$  coefficients; downsampling by  $2^\ell$  then leaves a single coefficient of size  $2^{-\ell/2}$ .

As an example of a piecewise constant sequence, we use the Heaviside sequence (3.10) delayed by  $k$ . A single wavelet coefficient will be different from zero at each scale, the one corresponding to the wavelet that straddles the discontinuity (Figure 3.11). At scale  $2^\ell$ , this corresponds to the wavelet with support from  $2^\ell \lfloor k/2^\ell \rfloor$  to  $2^\ell (\lfloor k/2^\ell \rfloor + 1)$ . All other wavelet coefficients are zero; on the left of the discontinuity because the sequence is zero, and on the right because the inner product is zero. The magnitude of the nonzero coefficient depends on the location  $k$  and varies between 0 and  $2^{\ell/2-1}$ . When  $k$  is a multiple of  $2^\ell$ , this magnitude is zero, and when  $k$  is equal to  $\ell 2^\ell + 2^{\ell/2}$ , it achieves its maximum value. The latter occurs when the discontinuity is aligned with the discontinuity



**Figure 3.11:** Characterization of singularities by the Haar DWT. (a) A Heaviside sequence at location  $k$ , and (b) the equivalent wavelet sequences, highpass filter  $h^{(\ell)}$  and its shifts, at scale  $2^\ell$ . A single wavelet, with support from  $n_1 = 2^\ell \lfloor k/2^\ell \rfloor$  to  $n_2 = 2^\ell (\lfloor k/2^\ell \rfloor + 1)$ , has a nonzero inner product with a magnitude of the order of  $2^{\ell/2}$ .

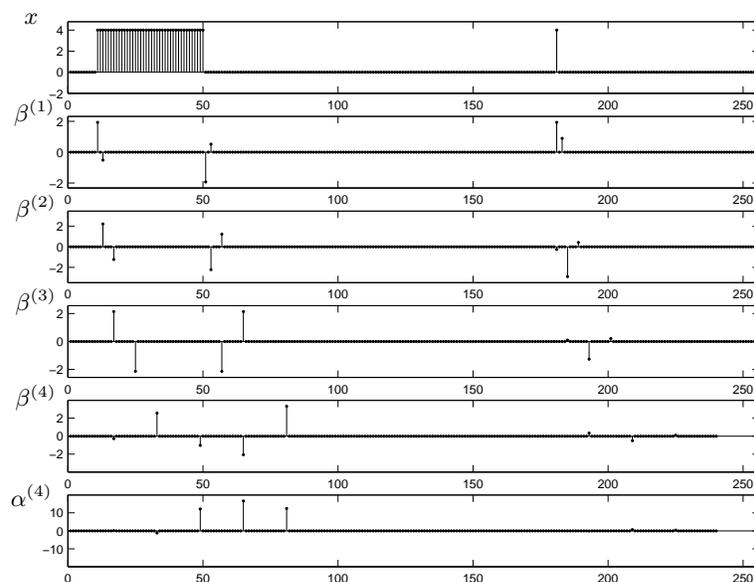
of the wavelet itself; then, the inner product is  $2^{\ell-1}2^{-\ell/2} = 2^{(\ell/2)-1}$ , and we obtain

$$\beta^{(\ell)} \sim 2^{(\ell/2)-1}.$$

We thus see that the magnitudes of the wavelet coefficients will vary, but as  $\ell$  increases, they will increase at most as  $2^{\ell/2}$ . In Figure 3.12(a), we show an example input sequence consisting of a piecewise constant sequence and a Kronecker delta sequences, and its DWT in (b). We see that the wavelet coefficients are gathered around the singular points (Kronecker delta, Heaviside step), and they decay or increase, depending on the type of singularity.

In the example above, we obtained precisely  $\beta^{(\ell)} = 2^{(\ell/2)-1}$  for the one nonzero wavelet coefficient at scale  $2^\ell$ . Figure 3.12 gives another example, with a sequence with more types of singularities and a DWT with longer filters. We again have  $\sim 2^{-\ell/2}$  scaling of wavelet coefficient magnitudes and a roughly constant number of nonzero wavelet coefficients per scale. We will study this effect in more detail in Chapter 7, where the bounds on the coefficient magnitudes will play a large role in quantifying approximation performance.

In summary, the DWT acts as a *singularity detector*, that is, it leads to nonzero wavelet coefficients around singular points of a sequence. The number of nonzero coefficients per scale is bounded by  $(L-1)$ . Moreover, the magnitude of the wavelet coefficients across scales is an indicator of the type of singularity. Together with its



**Figure 3.12:** A piecewise constant sequence plus a Kronecker delta sequence and its DWT. (a) The original sequence  $x$ . (b)–(e) Wavelet coefficients  $\beta^{(\ell)}$  at scales  $2^\ell$ ,  $\ell = 1, 2, 3, 4$ . (f) Scaling coefficients  $\alpha^{(4)}$ . To compare different channels, all the sequences have been upsampled by a factor  $2^\ell$ ,  $\ell = 1, 2, 3, 4$ .

polynomial approximation properties as described previously, this ability to characterize singularities will be the other key ingredient in building successful approximation schemes using the DWT in Chapter 7.

**Basis for  $\ell^2(\mathbb{Z})$**  An interesting twist on a  $J$ -level DWT is what happens when we let the number of levels  $J$  go to infinity. This will be one way we will be building continuous-time wavelet bases in Chapter 6. For sequences in  $\ell^2(\mathbb{Z})$ , such an infinitely-iterated DWT can actually build an orthonormal basis based on wavelet sequences (equivalent highpass filters) alone. The energy of the scaling coefficients vanishes in  $\ell^2$  norm; in other words, the original sequence is entirely captured by the wavelet coefficients, thus proving Parseval's equality for such a basis. While this is true in general, below we prove the result for Haar filters only; the general proof needs additional technical conditions that are beyond the scope of our text.

**THEOREM 3.2 (DISCRETE HAAR WAVELETS AS A BASIS FOR  $\ell^2(\mathbb{Z})$ )** The discrete-time wavelets  $h^{(\ell)}$  with impulse responses as in (3.2d) and their shifts by  $2^\ell$ ,

$$\Phi = \{h_{n-2^\ell k}^{(\ell)}\}_{k \in \mathbb{Z}, \ell=1, 2, \dots},$$

form an orthonormal basis for the space of finite-energy sequences,  $\ell^2(\mathbb{Z})$ .

*Proof.* To prove  $\Phi$  is an orthonormal basis, we must prove it is an orthonormal set and that it is complete. The orthonormality of basis functions was shown earlier in Section 3.1 for Haar filters and in Section 3.2 for the more general ones.

To prove completeness, we will show that Parseval's equality holds, that is, for an arbitrary input  $x \in \ell^2(\mathbb{Z})$ , we have

$$\|x\|^2 = \sum_{\ell=1}^{\infty} \|\beta^{(\ell)}\|^2, \quad (3.19)$$

where  $\beta^{(\ell)}$  are the wavelet coefficients at scales  $2^\ell$ ,  $\ell = 1, 2, \dots$ :

$$\beta_k^{(\ell)} = \langle x_n, h_{n-2^\ell k}^{(\ell)} \rangle_n.$$

For any finite number of decomposition levels  $J$ , the Parseval's equality (3.18) holds. Thus, our task is to show  $\lim_{J \rightarrow \infty} \|\alpha^{(J)}\|^2 = 0$ . We show this by bounding two quantities: the energy lost in truncating  $x$  and the energy in the scaling coefficients that represent the truncated sequence.

Without loss of generality, assume  $x$  has unit norm. For any  $\varepsilon > 0$ , we will show that  $\|\alpha^{(J)}\|^2 < \varepsilon$  for sufficiently large  $J$ . First note that there exists a  $K$  such that the restriction of  $x$  to  $\{-2^K, -2^K + 1, \dots, 0, 1, \dots, 2^K - 1\}$  has energy at least  $1 - \varepsilon/2$ ; this follows from the convergence of the series defining the  $\ell^2$  norm of  $x$ . Denote the restriction by  $\tilde{x}$ .

A  $K$ -level decomposition of  $\tilde{x}$  has at most two nonzero scaling coefficients  $\tilde{\alpha}_{-1}^{(K)}$  and  $\tilde{\alpha}_0^{(K)}$ . Each of these scaling coefficients satisfies  $|\tilde{\alpha}_k^{(K)}| \leq 1$  because  $\|\tilde{\alpha}^{(K)}\|^2 \leq \|\tilde{x}\|^2 = 1$  by Bessel's inequality. We will now consider further levels of decomposition beyond  $K$ . After one more level, the lowpass output  $\tilde{\alpha}^{(K+1)}$  has coefficients

$$\tilde{\alpha}_0^{(K+1)} = \frac{1}{\sqrt{2}} \tilde{\alpha}_0^{(K)} \quad \text{and} \quad \tilde{\alpha}_{-1}^{(K+1)} = \frac{1}{\sqrt{2}} \tilde{\alpha}_{-1}^{(K)}.$$

Similarly, after  $K + j$  total levels of decomposition we have

$$\tilde{\alpha}_k^{(K+j)} = \frac{1}{2^{j/2}} \tilde{\alpha}_k^{(K)} \leq \frac{1}{2^{j/2}}, \quad \text{for } k = -1, 0.$$

Thus,  $\|\tilde{\alpha}^{(K+j)}\|^2 = \left(\tilde{\alpha}_{-1}^{(K+j)}\right)^2 + \left(\tilde{\alpha}_0^{(K+j)}\right)^2 \leq 2^{-(j-1)}$ .

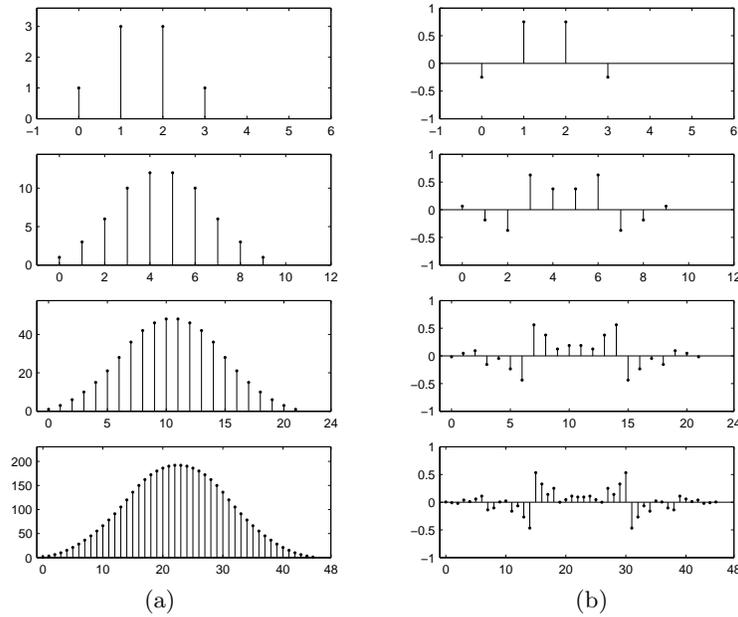
Let  $J = K + j$  where  $2^{-(j-1)} < \varepsilon/2$ . Then  $\|\alpha^{(J)}\|^2 < \varepsilon$  because  $\|\tilde{\alpha}^{(J)}\|^2 < \varepsilon/2$  and  $\|\alpha^{(J)}\|^2$  cannot exceed  $\|\tilde{\alpha}^{(J)}\|^2$  by more than the energy  $\varepsilon/2$  excluded in the truncation of  $x$ .

## 3.4 Biorthogonal Discrete Wavelet Transform

We have seen that the properties of the dyadic orthogonal DWT follow from the properties of the orthogonal two-channel filter bank. Similarly, the properties of the biorthogonal DWT follow from the properties of the biorthogonal two-channel filter bank. Instead of fully developing the biorthogonal DWT (as it is parallel to the orthogonal one), we quickly summarize its salient elements.

### 3.4.1 Definition of the Biorthogonal DWT

If, instead of an orthogonal pair of highpass/lowpass filters, we use a biorthogonal set  $\{h, g, \tilde{h}, \tilde{g}\}$  as in (1.64a)–(1.64d), we obtain two sets of equivalent filters, one for



**Figure 3.13:** Iteration of a biorthogonal pair of lowpass filters from (3.21): (a) the iteration of  $g_n$  leads to a *smooth*-looking sequence, while (b) the iteration of  $\tilde{g}$  does not.

the synthesis side,  $G^{(J)}, H^{(\ell)}, \ell = 1, 2, \dots, J$ , and the other for the analysis side,  $\tilde{G}^{(J)}, \tilde{H}^{(\ell)}, \ell = 1, 2, \dots, J$ :

$$G^{(J)}(z) = \prod_{k=0}^{J-1} G(z^{2^k}), \quad H^{(\ell)}(z) = H(z^{2^{\ell-1}})G^{(\ell-1)}(z), \quad (3.20a)$$

$$\tilde{G}^{(J)}(z) = \prod_{k=0}^{J-1} \tilde{G}(z^{2^k}), \quad \tilde{H}^{(\ell)}(z) = \tilde{H}(z^{2^{\ell-1}})\tilde{G}^{(\ell-1)}(z), \quad (3.20b)$$

for  $\ell = 1, \dots, J$ . This iterated product will play a crucial role in the construction of continuous-time wavelet bases in Chapter 6, as  $g^{(J)}$  and  $\tilde{g}^{(J)}$  can exhibit quite different behaviors; we illustrate this with an example.

**EXAMPLE 3.3 (BIORTHOGONAL DWT)** In Example 1.4, we derived a biorthogonal pair with lowpass filters

$$g_n = [\dots 0 \ 1 \ 3 \ 3 \ 1 \ 0 \ \dots], \quad (3.21a)$$

$$\tilde{g}_n = \frac{1}{4} [\dots 0 \ -1 \ 3 \ 3 \ -1 \ 0 \ \dots]. \quad (3.21b)$$

Figure 3.13 shows the first few iterations of  $g^{(J)}$  and  $\tilde{g}^{(J)}$  indicating a very different behavior. Recall that both filters are lowpass filters as are their iterated

versions. However, the iteration of  $\tilde{g}$  does not look *smooth*, indicating possible problems as we iterate to infinity (as we will see in Chapter 6).

Similarly to properties that equivalent filters satisfy in an orthogonal DWT (Section 3.2), we have such properties here mimicking the biorthogonal relations from Section 1.4; their formulation and proofs are left as an exercise to the reader.

**DEFINITION 3.3 (BIORTHOGONAL DWT)** The  $J$ -level biorthogonal DWT of a sequence  $x$  is a function of  $\ell \in \{1, 2, \dots, J\}$  given by

$$\alpha_k^{(J)} = \langle x_n, \tilde{g}_{n-2^J k}^{(J)} \rangle_n, \quad \beta_k^{(\ell)} = \langle x_n, \tilde{h}_{n-2^\ell k}^{(\ell)} \rangle_n, \quad k, n \in \mathbb{Z}, \ell \in \{1, 2, \dots, J\}. \quad (3.22a)$$

The inverse DWT is given by

$$x_n = \sum_{k \in \mathbb{Z}} \alpha_k^{(J)} g_{n-2^J k}^{(J)} + \sum_{\ell=1}^J \sum_{k \in \mathbb{Z}} \beta_k^{(\ell)} h_{n-2^\ell k}^{(\ell)}. \quad (3.22b)$$

In the above, the  $\alpha^{(J)}$  are the *scaling coefficients* and the  $\beta^{(\ell)}$  are the *wavelet coefficients*.

The equivalent filters  $g^{(J)}$ ,  $\tilde{g}^{(J)}$  are often called the *scaling sequences*, and  $h^{(\ell)}$ ,  $\tilde{h}^{(\ell)}$ ,  $\ell = 1, 2, \dots, J$ , *wavelets* (wavelet sequences).

### 3.4.2 Properties of the Biorthogonal DWT

Similarly to the orthogonal DWT, the biorthogonal DWT is linear and shift varying. As a biorthogonal expansion, it does not satisfy Parseval's equality. However, as we have now access to dual bases, we can choose which one to use for projection (analysis) and which one for reconstruction (synthesis). This allows us to choose a better-suited one between  $g$  and  $\tilde{g}$  to induce an expansion with desired polynomial approximation properties and characterization of singularities.

## 3.5 Wavelet Packets

So far, the iterated decomposition was always applied to the lowpass filter, and often, there are good reasons to do so. However, to match a wide range of sequences, we can consider an arbitrary tree decomposition. In other words, start with a sequence  $x$  and decompose it into a lowpass and a highpass version.<sup>24</sup> Then, decide if the lowpass, the highpass, or both, are decomposed further, and keep going until a given depth  $J$ . The DWT is thus one particular case when only the lowpass version is repeatedly decomposed. Figure 3.7 depicts some of these decomposition possibilities.

<sup>24</sup>Of course, there is no reason why one could not split into  $N$  channels initially.

For example, the full tree yields a linear division of the spectrum similar to the local Fourier transform from Chapter 2, while the octave-band tree performs a  $J$ -level DWT expansion. Such arbitrary tree structures were introduced as a family of orthonormal bases for discrete-time sequences, and are known under the name of *wavelet packets*. The potential of wavelet packets lies in the capacity to offer a rich menu of orthonormal bases, from which the *best* one can be chosen (best according to a given criterion). We discuss this in more detail in Chapter 7.

What we do here is define the basis functions and write down the appropriate orthogonality relations; since the proofs are in principle similar to those for the DWT, we chose to omit them.

### 3.5.1 Definition of the Wavelet Packets

**Equivalent Channels and Their Properties** Denote the equivalent filters by  $g_{i,n}^{(\ell)}$ ,  $i = 0, \dots, 2^\ell - 1$ . In other words,  $g_i^{(\ell)}$  is the  $i$ th equivalent filter going through one of the possible paths of length  $\ell$ . The ordering is somewhat arbitrary, and we will choose the one corresponding to a full tree with a lowpass in the lower branch of each fork, and start numbering from the bottom.

**EXAMPLE 3.4 (2-LEVEL WAVELET PACKET EQUIVALENT FILTERS)** Let us find all equivalent filters at level 2, or, the filters corresponding to depth-1 and depth-2 trees.

$$\begin{aligned} G_0^{(1)}(z) &= G_0(z), & G_1^{(1)}(z) &= G_1(z), \\ G_0^{(2)}(z) &= G_0(z) G_0(z^2), & G_1^{(2)}(z) &= G_0(z) G_1(z^2), \\ G_2^{(2)}(z) &= G_1(z) G_0(z^2), & G_3^{(2)}(z) &= G_1(z) G_1(z^2). \end{aligned} \quad (3.23)$$

With the ordering chosen in the above equations for level 2, increasing index does not always correspond to increasing frequency. For ideal filters,  $G_2^{(2)}(e^{j\omega})$  chooses the range  $[3\pi/4, \pi)$ , while  $G_3^{(2)}(e^{j\omega})$  covers the range  $[\pi/2, 3\pi/4)$ . Beside the identity basis, which corresponds to the no-split situation, we have four possible orthonormal bases (full 2-level split, full 1-level split, full 1-level split plus either lowpass or highpass split).

**Wavelet Packet Bases** Among the myriad of possible bases wavelet packets generate, one can choose that one best fitting the sequence at hand.

**EXAMPLE 3.5 (2-LEVEL WAVELET PACKET BASES)** Continuing Example 3.4, we have a family  $W = \{\Phi_0, \Phi_1, \Phi_2, \Phi_3, \Phi_4\}$ , where  $\Phi_4$  is simply  $\{\delta_{n-k}\}_{k \in \mathbb{Z}}$ :

$$\Phi_0 = \{g_{0,n-2^2k}^{(2)}, g_{1,n-2^2k}^{(2)}, g_{2,n-2^2k}^{(2)}, g_{3,n-2^2k}^{(2)}\}_{k \in \mathbb{Z}}$$

corresponds to the full tree;

$$\Phi_1 = \{g_{1,n-2k}^{(1)}, g_{1,n-2^2k}^{(2)}, g_{0,n-2^2k}^{(2)}\}_{k \in \mathbb{Z}}$$

corresponds to the DWT tree;

$$\Phi_2 = \{g_{0,n-2k}^{(1)}, g_{2,n-2^2k}^{(2)}, g_{3,n-2^2k}^{(2)}\}_{k \in \mathbb{Z}}$$

corresponds to the tree with the highpass split twice; and,

$$\Phi_3 = \{g_{0,n-2k}^{(0)}, g_{1,n-2k}^{(1)}\}_{k \in \mathbb{Z}}$$

corresponds to the usual two-channel filter bank basis.

In general, we will have Fourier-like bases, given by

$$\Phi_0 = \{g_{0,n-2^Jk}^{(J)}, \dots, g_{2^J-1,n-2^Jk}^{(J)}\}_{k \in \mathbb{Z}}, \quad (3.24)$$

and wavelet-like bases, given by

$$\Phi_1 = \{g_{1,n-2k}^{(1)}, g_{1,n-2^2k}^{(2)}, \dots, g_{1,n-2^Jk}^{(J)}, g_{0,n-2^Jk}^{(J)}\}_{k \in \mathbb{Z}}. \quad (3.25)$$

That these are all bases follows trivially from each building block being a basis (either orthonormal or biorthogonal).

### 3.5.2 Properties of the Wavelet Packets

Exercises at the end of this chapter discuss various forms and properties of wavelet packets: biorthogonal wavelet packets in Exercise ?? and arbitrary wavelet packets in Exercise ??.

**Number of Wavelet Packets** How many wavelet packets (different trees) are there? Call  $N^{(J)}$  the number of trees of depth  $J$ , then we have the recursion

$$N^{(J)} = (N^{(J-1)})^2 + 1, \quad (3.26)$$

since each branch of the initial two-channel filter bank can have  $N^{(J-1)}$  possible trees attached to it and the +1 comes from not splitting at all. As an initial condition, we have  $N^{(1)} = 2$  (either no split or a single split). It can be shown that the recursion leads to an order of

$$N^{(J)} \sim 2^{2^J} \quad (3.27)$$

possible trees. Of course, many of these trees will be poor matches to real-life sequences, but an efficient search algorithm allowing to find the best match between a given sequence and a tree-structured expansion is possible. The proof of (3.26) is left as Exercise ??.

## 3.6 Computational Aspects

We now consider the computational complexity of the DWT, and show an elementary but astonishing result, in large part responsible for the popularity of the DWT: the complexity is linear in the input size.

**Complexity of the DWT** Computing a DWT amounts to computing a set of convolutions but with a twist crucial to the computational efficiency of the transform; as the decomposition progresses down the tree (see, for example, Figure 3.1), the sampling rate decreases. The implementation of  $J$ -level DWT for a length- $N$  signal with a filter bank is equivalent to a factorization

$$\begin{bmatrix} I_{(1-2^{-J+1})N} & 0 \\ & \begin{bmatrix} H^{(J)} \\ G^{(J)} \end{bmatrix} \end{bmatrix} \cdots \begin{bmatrix} I_{3N/4} & 0 \\ & \begin{bmatrix} H^{(3)} \\ G^{(3)} \end{bmatrix} \end{bmatrix} \begin{bmatrix} I_{N/2} & 0 \\ & \begin{bmatrix} H^{(2)} \\ G^{(2)} \end{bmatrix} \end{bmatrix} \begin{bmatrix} H^{(1)} \\ G^{(1)} \end{bmatrix},$$

where  $H^{(\ell)}$ 's and  $G^{(\ell)}$  are the highpass and lowpass operators, respectively, each with downsampling by two (see (3.199) and (3.202) for  $\ell = 1$ ), both sparse.

In the DWT tree, the second level has similar cost to the first, but at half the sampling rate (see (3.276)). Continuing this argument, the cost of the  $J$ -level DWT is

$$L + \frac{L}{2} + \frac{L}{4} + \cdots + \frac{L}{2^{J-1}} < 2L$$

in both multiplications and additions, with the cost of the order of at most

$$C_{\text{DWT}} \sim 2NL \sim O(N), \quad (3.28)$$

that is, it is linear in the input size with a constant depending on the filter length.

While the cost remains bounded, the delay does not. If the first block contributes a delay  $D$ , the second will produce a delay  $2D$  and the  $\ell$ th block a delay  $2^{\ell-1}D$ , for a total delay of

$$D_{\text{DWT}} = D + 2D + 2^2D + \cdots + 2^{J-1}D = (2^J - 1)D.$$

This large delay is a serious drawback, especially for real-time applications such as speech coding.

**Complexity of the General Wavelet Packets** What happens for more general trees? Clearly, the worst case is for a full tree (see Figure 3.7(e)).

We start with a naive implementation of a  $2^J$ -channel filter bank, downsampled by  $2^J$ . Recall that, according to (3.5b), the length of the equivalent filters (in the DWT or the full tree) are of the order  $O(L2^J)$ . Computing each filter and downsampling by  $2^J$  leads to  $L$  operations per channel, or, for  $2^J$  channels we obtain

$$C_{\text{direct}} \sim NL2^J \sim O(NL2^J), \quad (3.29)$$

which grows exponentially with  $J$ . Exercise ?? compares these two implementations of the full tree with two Fourier-based ones, showing gains in computational cost.

As the sampling rate goes down, the number of channels goes up, and the two effects cancel each other. Therefore, for  $J$ , the cost amounts to

$$C_{\text{full}} \sim NLJ \sim O(NLJ), \quad (3.30)$$

multiplications or additions, again for a length- $N$  sequence and length- $L$  filter.

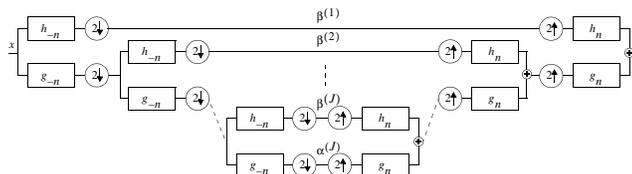
### Chapter at a Glance

The goal of this chapter was twofold: (1) to extend the discussion from Chapters 1 and 2 to more multichannel filter banks constructed as trees and associated bases; and (2) to consider those filter banks implementing the DWT and wavelet packets.

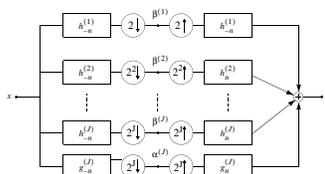
While in general, tree-structured filter banks can have as their building blocks general  $N$ -channel filter banks, here we concentrated mostly on those built using basic two-channel filter banks. Moreover, the bulk of the chapter was devoted to those tree-structured filter banks, octave-band, where only the lowpass channel is further decomposed, as they implement the DWT. Such a decomposition is a natural one, with both theoretical and experimental evidence to support its use. Experimentally, research shows that the human visual system decomposes the field of view into octave bands; in parallel, theoretically, the DWT is an appropriate tool for the analysis of smooth sequences with isolated discontinuities. Moreover, the DWT has interesting polynomial approximation powers as well as the ability to characterize singularities.

Wavelet packets extend these ideas to more general tilings of the time-frequency plane, adapting the decomposition to the sequence at hand. Here we discussed the decompositions only; the criteria for which decomposition to use are left for later.

**Block diagram: Tree structure**



**Block diagram: Multichannel structure**



**Basic characteristics**

number of channels	$M = J + 1$	
sampling at level $\ell$	$N^{(\ell)} = 2^\ell$	
channel sequences	$\alpha_n^{(j)}$ $\beta_n^{(\ell)}$	$\ell = 1, 2, \dots, J$

**Filters**

**Synthesis    Analysis**

	lowpass	bandpass <sup>(<math>\ell</math>)</sup>	lowpass	bandpass <sup>(<math>\ell</math>)</sup>	
orthogonal	$g_n^{(j)}$	$h_n^{(\ell)}$	$g_{-n}^{(j)}$	$h_{-n}^{(\ell)}$	$\ell = 1, 2, \dots, J$
biorthogonal	$g_n^{(j)}$	$h_n^{(\ell)}$	$\tilde{g}_n^{(j)}$	$\tilde{h}_n^{(\ell)}$	
polyphase component $j$	$g_{j,n}^{(j)}$	$h_{j,n}^{(\ell)}$	$\tilde{g}_{j,n}^{(j)}$	$\tilde{h}_{j,n}^{(\ell)}$	$j = 0, 1, \dots, 2^\ell - 1$

**Table 3.3:** DWT filter bank.

## Historical Remarks



The tree-structured filter banks, and, in particular, octave-band, or, constant-Q, ones, have been used in speech and audio. A similar scheme, but redundant, was proposed as a pyramid coding technique by Burt and Adelson [15], and served as the initial link between the discrete-time setting and the works in the continuous-time setting of Daubechies [29] and Mallat [59]. This prompted a flurry of connections between the wavelet transform, filter banks, and subband coding schemes, for example, the biorthogonal bases by Herley and Vetterli [44]. It, moreover, further opened the door to a formal treatment and definition of the DWT as a purely discrete transform, and not only as a vehicle for implementing continuous-time ones. Rioul [72] rigorously defined the discrete multiresolution analysis, and Coifman, Meyer, Quake and Wickerhauser [24] proposed wavelet packets as an adaptive tool for signal analysis. As a result of these developments and its low computational complexity, the DWT and its variations found their way into numerous applications and standards, JPEG 2000 among others.

## Further Reading

**Books and Textbooks** Numerous books cover the topic of the DWT, such as those by Vetterli and Kovačević [106], Strang and Nguyen [87] and Mallat [60], among others.

**Wavelet Packets** Wavelet packets were introduced in 1991 by Coifman, Meyer, Quake and Wickerhauser [24], followed by widely cited [112] and [113]. In 1993, Ramchandran and Vetterli showed for the first time a different cost measure for pruning wavelet packets, rate-distortion, as that suitable for compression [69], and Saito and Coifman extended the idea further with local discriminant bases [74, 75].



## Chapter 4

# Local Fourier and Wavelet Frames on Sequences

## Contents

---

4.1	Introduction	122
4.2	Finite-Dimensional Frames	133
4.3	Oversampled Filter Banks	151
4.4	Local Fourier Frames	158
4.5	Wavelet Frames	165
4.6	Computational Aspects	171
	Chapter at a Glance	173
	Historical Remarks	174
	Further Reading	174

---

Redundancy is a common tool in our daily lives; it helps remove doubt or uncertainty. Redundant signal representations follow the same idea to create robustness. Given a sequence, we often represent it in another domain where its characteristics are more readily apparent in the expansion coefficients. If the representation in that other domain is achieved via a basis, corruption or loss of expansion coefficients can be serious. If, on the other hand, that representation is achieved via a redundant representation, such problems can be avoided.

As introduced in *Chapter 2, Section 2.5.4*, the redundant counterpart of bases are called *frames*. Frames are the topic of the present chapter. The building blocks of a representation can be seen as words in a dictionary; while a basis uses a minimum number of such words, a frame uses an overcomplete set. This is similar to multiple words with slight variations for similar concepts, allowing for very short sentences describing complex ideas.<sup>25</sup> While in most of the previous chapters, our emphasis was on finding the *best* expansion/representation vectors (Fourier, wavelet, etc.), frames allow us even more freedom; not only do we look for the *best expansion*, we can also look for the *best expansion coefficients* given a fixed expansion and under desired constraints (sparsity as one example). This freedom is due to

---

<sup>25</sup>As the urban legend goes, Eskimos have a hundred words for snow.

the fact that in a redundant dictionary, the expansion coefficients are not unique, while in a basis they are.

We briefly introduced frames in  $\mathbb{R}^2$  in *Chapter 2, Section 2.1*. We followed this by a more formal discussion in *Section 2.5.4*. Our goal in this chapter is to explore the potential of such overcomplete (redundant) representations in specific settings; in particular, we study fundamental properties of frames, both in finite dimensions as well as in  $\ell^2(\mathbb{Z})$ . We look into designing frames with some structure, especially those implementable by oversampled filter banks, and more specifically those with Fourier-like or wavelet-like time-frequency behavior. We end the chapter with the discussion of computational aspects related to frame expansions.

*Notation used in this chapter:* We consider both real-coefficient and complex-coefficient frames here, unlike, for example, in Chapter 1. When Hermitian conjugation is applied to polyphase matrices, it is applied only to coefficients and not to  $z$ .  $\square$

## 4.1 Introduction

Redundant sets of vectors look like an *overcomplete* basis,<sup>26</sup> and we call such sets frames. Thus, a frame is an extension of a basis, where, for a given space, more vectors than necessary are used to obtain an expansion with desirable properties. In this section, we use the two frame examples from *Section 2.1* to introduce and discuss frame concepts in a simple setting but in more detail. For ease of presentation, we will repeat pertinent equations as well as figures.

### A Tight Frame for $\mathbb{R}^2$

Our first example is that from (2.15), a set of three vectors in  $\mathbb{R}^2$ ,

$$\varphi_0 = \begin{bmatrix} \sqrt{\frac{2}{3}} \\ 0 \end{bmatrix}, \quad \varphi_1 = \begin{bmatrix} -\frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}, \quad \varphi_2 = \begin{bmatrix} -\frac{1}{\sqrt{6}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}. \quad (4.1)$$

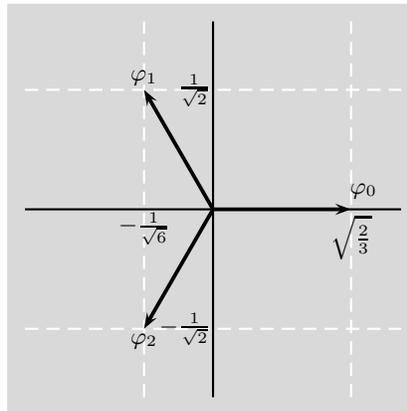
**Expansion** These vectors clearly span  $\mathbb{R}^2$  since any two of them do (see Figure 4.1). How do we represent a vector  $x \in \mathbb{R}^2$  as a linear combination of  $\{\varphi_i\}_{i=0,1,2}$ ,

$$x = \sum_{i=0}^2 \alpha_i \varphi_i? \quad (4.2)$$

Let us gather these vectors into a frame matrix  $\Phi$ ,

$$\Phi = [\varphi_0 \quad \varphi_1 \quad \varphi_2] = \begin{bmatrix} \sqrt{\frac{2}{3}} & -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}. \quad (4.3)$$

<sup>26</sup>Even though this term is a contradiction.



**Figure 4.1:** The three vectors  $\{\varphi_0, \varphi_1, \varphi_2\}$  from (4.1) form a frame for  $\mathbb{R}^2$  (the same as Figure 2.4(b)).

To compute the expansion coefficients  $\alpha_i$  in (4.2), we need a right inverse of  $\Phi$ . Since  $\Phi$  is rectangular, such an inverse is not unique, so we look for the simplest one. As the rows of  $\Phi$  are orthonormal,  $\Phi\Phi^T = I_2$ , and thus, a possible right inverse of  $\Phi$  is just its transpose:

$$\Phi^T = \begin{bmatrix} \sqrt{\frac{2}{3}} & 0 \\ -\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \varphi_0^T \\ \varphi_1^T \\ \varphi_2^T \end{bmatrix}. \quad (4.4)$$

Gathering the expansion coefficients into a vector  $\alpha$ ,

$$\alpha = \Phi^T x, \quad (4.5)$$

and, using the fact that

$$\Phi\alpha = \Phi\Phi^T x = x,$$

we obtain the following expansion formula:

$$x = \sum_{i=0}^2 \langle x, \varphi_i \rangle \varphi_i, \quad (4.6)$$

for all  $x \in \mathbb{R}^2$ , which looks exactly like the usual orthonormal expansion (2.91a), except for the number of vectors involved.

**Geometry of the Expansion** Let us understand why this is so. The rows of  $\Phi$  are orthonormal,

$$\Phi\Phi^T = I_2. \quad (4.7)$$

Actually,  $\Phi$  can be seen as two rows of a unitary matrix whose third row is orthogonal to the rows of  $\Phi$ . We call that third row  $\Phi^\perp$ , that is,  $\Phi^\perp = 1/\sqrt{3} [1 \ 1 \ 1]$ . That unitary matrix is then

$$\begin{bmatrix} \Phi \\ \Phi^\perp \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{2}{3}} & -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix}. \quad (4.8)$$

We can then write

$$\Phi\Phi^T = I_{2 \times 2}, \quad (4.9a)$$

$$\Phi(\Phi^\perp)^T = 0_{2 \times 1}, \quad (4.9b)$$

$$\Phi^\perp(\Phi^\perp)^T = I_{1 \times 1} = 1. \quad (4.9c)$$

Calling  $S$  the subspace of  $\mathbb{R}^3$  spanned by the columns of  $\Phi^T$ , and  $S^\perp$  its orthogonal complement in  $\mathbb{R}^3$  (spanned by the one column of  $(\Phi^\perp)^T$ ), we can write

$$\begin{aligned} S &= \text{span}(\Phi^T) = \text{span}\left(\begin{bmatrix} \sqrt{\frac{2}{3}} \\ -\frac{1}{\sqrt{6}} \\ -\frac{1}{\sqrt{6}} \end{bmatrix}, \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}\right), \\ S^\perp &= \text{span}((\Phi^\perp)^T) = \text{span}\left(\begin{bmatrix} \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \end{bmatrix}\right), \\ \mathbb{R}^3 &= S \oplus S^\perp. \end{aligned}$$

We just saw that the rows of  $\Phi$  are orthonormal; moreover, while not of unit norm, the columns of  $\Phi$  are of the same norm,  $\|\varphi_i\| = \sqrt{2/3}$ . Therefore,  $\Phi$  is a very special matrix, as can be guessed by looking at Figure 4.1.

Let us now understand the nature of the expansion coefficients  $\alpha$  a bit more in depth. Obviously,  $\alpha$  cannot be arbitrary; since  $\alpha = \Phi^T x$ , it belongs to the range of the columns of  $\Phi^T$ , or,  $\alpha \in S$ . What about some arbitrary  $\alpha' \in \mathbb{R}^3$ ? As the expansion coefficients must belong to  $S$ , we can calculate the orthogonal projection of  $\alpha'$  onto  $S$  by first calculating some  $x' = \Phi\alpha'$ , and then computing the unique orthogonal projection we call  $\alpha$  as

$$\alpha = \Phi^T x' = \Phi^T \Phi \alpha',$$

where  $G = \Phi^T \Phi$  is the Gram matrix from (2.118),

$$G = \Phi^T \Phi \stackrel{(a)}{=} \frac{1}{3} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}, \quad (4.10)$$

and (a) follows from (4.3), (4.4). We can therefore express any  $\alpha' \in \mathbb{R}^3$  as

$$\alpha' = \alpha + \alpha^\perp \quad (4.11a)$$

with  $\alpha \in S$  and  $\alpha^\perp \in S^\perp$ , and thus

$$\langle \alpha, \alpha^\perp \rangle = 0. \quad (4.11b)$$

Therefore, we see that, for our  $\Phi$ , many different  $\alpha'$  are possible as expansion coefficients. In other words,  $\Phi^T$  is not the only possible right inverse of  $\Phi$ . While this is not surprising, it allows frames to be extremely flexible, a fact we will explore in detail in the next section. Throughout this chapter, when we write  $\alpha'$ , we will mean any vector of expansion coefficients; in contrast,  $\alpha$  will be the unique one obtained using the canonical (unique) dual frame.

**Energy of Expansion Coefficients** For orthonormal bases, Parseval's equality (energy conservation) (2.93a) is fundamental. To find out what happens here, we compute the norm of  $\alpha$ ,

$$\|\alpha\|^2 = \alpha^T \alpha \stackrel{(a)}{=} x^T \Phi \Phi^T x \stackrel{(b)}{=} x^T x = \|x\|^2, \quad (4.12)$$

where (a) follows from (4.5); and (b) from (4.7), again formally the same as for an orthonormal basis. Beware though that the comparison is not entirely fair, as the frame vectors are not of unit norm; we will see in a moment what happens when this is the case.

**Robustness to Corruption and Loss** What does the redundancy of this expansion buy us? For example, what if the expansion coefficients get corrupted by noise? Assume, for instance, that  $\alpha$  is perturbed by noise  $\eta'$ , where the noise components  $\eta'_i$  are uncorrelated with  $\|\eta'\| = 1$ . Then, reconstruction will project the noise  $\eta' = \eta + \eta^\perp$ , and thus cancel that part of  $\eta$  not in  $S$ :

$$y = \Phi(\alpha' + \eta') = x + \underbrace{\Phi\eta}_{x_\eta} + \underbrace{\Phi\eta^\perp}_0.$$

To compute  $\|x_\eta\|^2$ , we write

$$\|x_\eta\|^2 = x_\eta^T x_\eta = \eta^T \Phi^T \Phi \eta = \eta^T U \Sigma U^T \eta,$$

where we have performed a singular value decomposition (2.222) on  $G = \Phi^T \Phi$  as

$$G = \Phi^T \Phi = U \Sigma U^T = \begin{bmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{3}} \\ 0 & \sqrt{\frac{2}{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & \\ & & 0 \end{bmatrix} \begin{bmatrix} -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{6}} & \sqrt{\frac{2}{3}} & -\frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix},$$

and  $U$  is a unitary matrix. So  $\|\eta^T U\| = 1$ , and thus  $\|x_\eta\|^2 = (2/3)\|\eta\|^2$ . We have thus established that the energy of the noise gets reduced during reconstruction by the *contraction* factor  $2/3$ .

We have just looked at the effect of noise on the reconstructed vector in a frame expansion. We now ask a different question: What happens if, for some reason, we

have access to only two out of three expansion coefficients during reconstruction (for example, one was lost)? As in this case, any two remaining vectors form a basis, we will still be able to reconstruct; however, the reconstruction is now performed differently. For example, assume we lost the first expansion coefficient  $\alpha_0$ . To reconstruct, we must behave as if we had started without the first vector  $\varphi_0$  and had computed the expansion coefficients using only  $\varphi_1$  and  $\varphi_2$ . This further means that to reconstruct, we must find the inverse of the  $2 \times 2$  submatrix of  $\Phi^T$  formed by taking its last two rows. This new reconstruction matrix  $\Phi^e$  (where  $e$  stands for *erasures*) is

$$\Phi^e = \begin{bmatrix} -\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} \end{bmatrix}^{-1} = \begin{bmatrix} -\frac{\sqrt{3}}{\sqrt{2}} & -\frac{\sqrt{3}}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix},$$

and thus, multiplying  $[\alpha_1 \ \alpha_2]^T$  by  $\Phi^e$  reconstructs the input vector:

$$\Phi^e \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} -\frac{\sqrt{3}}{\sqrt{2}} & -\frac{\sqrt{3}}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} -\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}.$$

**Unit-Norm Version** The frame we have just seen is a very particular frame and intuitively *close* to an orthonormal basis. However, there is one difference: while all the frame vectors are of the same norm  $\sqrt{2/3}$ , they are not of unit norm. We can normalize  $\|\varphi_i\|$  to be of norm 1, leading to

$$\Phi = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}, \quad \Phi^T = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}, \quad (4.13)$$

yielding the expansion

$$x = \frac{2}{3} \sum_{i=0}^2 \langle x, \varphi_i \rangle \varphi_i, \quad (4.14)$$

and the energy in the expansion coefficients

$$\|\alpha\|^2 = \frac{3}{2} \|x\|^2. \quad (4.15)$$

The difference between (4.13) and (4.3) is in normalization; thus, the factor  $(3/2)$  appears in (4.15), showing that the energy in the expansion coefficients is  $(3/2)$  times larger than that of the input vector. When the frame vectors are of unit norm as in this case, this factor represents the *redundancy*<sup>27</sup> of the system—we have  $(3/2)$  times more vectors than needed to represent a vector in  $\mathbb{R}^2$ .

The frame (4.3) and its normalized version (4.13) are instances of the so-called *tight frames*. A tight frame has a right inverse that is its own transpose and conserves energy (both within a scale factor). While the tight frame vectors we have seen are of the same norm, in general, this is not a requirement for tightness, as we will see later in the chapter.

<sup>27</sup>There exists a precise quantitative definition of redundancy, see *Further Reading* for details.

**Filter-Bank Implementation** As we have seen in previous chapters, infinite-dimensional expansions can be implemented using filter banks. Let us for a moment go back to the Haar expansion for  $\ell^2(\mathbb{Z})$  and try to draw some parallels. First, we have, until now, seen many times a  $2 \times 2$  Haar matrix  $\Phi$ , as a basis for  $\mathbb{R}^2$ . Then, in *Chapter 7*, we used these Haar vectors to form a basis for  $\ell^2(\mathbb{Z})$ , by slicing the infinite-length sequence into pieces of length 2 and applying a Haar basis to each of these. The resulting basis sequences for  $\ell^2(\mathbb{Z})$  are then obtained as infinite sequences with two nonzero elements only, shifted by integer multiples of 2,  $(??), (??)$ . Finally, in Chapter 1, (1.2)–(1.2), we showed how to implement such an orthonormal expansion for  $\ell^2(\mathbb{Z})$  by using a two-channel filter bank.

We can do exactly the same here. We slice an infinite-length input sequence into pieces of length 2 and apply the frame we just saw to each of these. To form a frame for  $\ell^2(\mathbb{Z})$ , we form three template frame vectors from the vectors (4.1) as:

$$\varphi_0 = \begin{bmatrix} \vdots \\ 0 \\ \sqrt{\frac{2}{3}} \\ 0 \\ 0 \\ \vdots \end{bmatrix}, \quad \varphi_1 = \begin{bmatrix} \vdots \\ 0 \\ -\frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{2}} \\ 0 \\ \vdots \end{bmatrix}, \quad \varphi_2 = \begin{bmatrix} \vdots \\ 0 \\ -\frac{1}{\sqrt{6}} \\ -\frac{1}{\sqrt{2}} \\ 0 \\ \vdots \end{bmatrix}. \quad (4.16)$$

We then form all the other frame sequences as versions of (4.16) shifted by integer multiples of 2:

$$\Phi = \{\varphi_{0,n-2k}, \varphi_{1,n-2k}, \varphi_{2,n-2k}\}_{k \in \mathbb{Z}}.$$

To implement this frame expansion using signal processing machinery, we do exactly the same as we did for Haar basis in Section 1.1: we rename the template frame sequences  $\varphi_0 = g_0$ ,  $\varphi_1 = g_1$  and  $\varphi_2 = g_2$ . Then we can write the reconstruction formula as

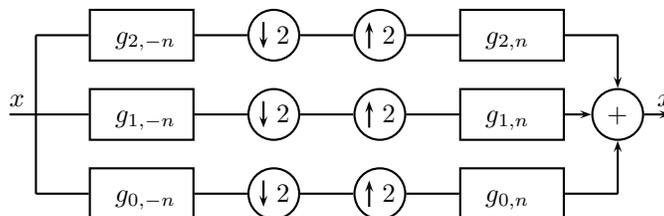
$$x_n = \sum_{k \in \mathbb{Z}} \alpha_{0,k} g_{0,n-2k} + \sum_{k \in \mathbb{Z}} \alpha_{1,k} g_{1,n-2k} + \sum_{k \in \mathbb{Z}} \alpha_{2,k} g_{2,n-2k}, \quad (4.17a)$$

with

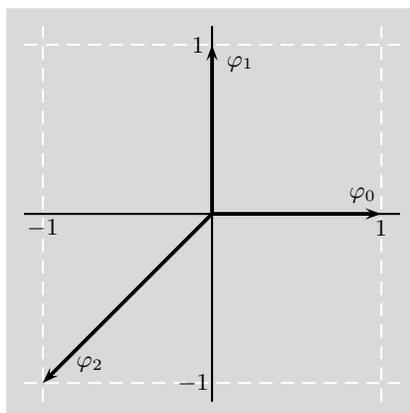
$$\alpha_{i,k} = \langle x_n, g_{i,n-2k} \rangle_n. \quad (4.17b)$$

There is really no difference between (4.17a)–(4.17b) and (1.2)–(1.3), except that we have 3 template frame sequences here instead of 2 template basis sequences for Haar.<sup>28</sup> We thus know exactly how to implement (4.17): it is going to be a 3-channel filter bank with down/upsampling by 2, as shown in Figure 4.2, with synthesis filters' impulse responses given by the frame vectors, and analysis filters' impulse responses given by the time-reversed frame vectors.

<sup>28</sup>Unlike for the Haar case, the  $\alpha_i$  are not unique.



**Figure 4.2:** A 3-channel filter bank with sampling by 2 implementing a tight frame expansion.



**Figure 4.3:** The three vectors  $\{\varphi_0, \varphi_1, \varphi_2\}$  from (4.18) form a frame for  $\mathbb{R}^2$  (the same as Figure 2.4(a)).

### A General Frame for $\mathbb{R}^2$

Our second example is that from (2.14), again a set of three vectors in  $\mathbb{R}^2$ ,

$$\varphi_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \varphi_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \varphi_2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \quad (4.18)$$

the standard orthonormal basis  $\{\varphi_0, \varphi_1\}$  plus a third vector. We follow the same path as we just did to spot commonalities and differences.

**Expansion** Again, these vectors clearly span  $\mathbb{R}^2$  since any two of them do (see Figure 4.3). We have already paved the path for representing a vector  $x \in \mathbb{R}^2$  as a linear combination of  $\{\varphi_i\}_{i=0,1,2}$  by introducing a matrix  $\Phi$  as in (2.16a),

$$\Phi = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix}. \quad (4.19)$$

Unlike (4.3), this  $\Phi$  does not have orthogonal rows, and thus,  $\Phi^T$  is not one of its right inverses. We call these right inverses  $\tilde{\Phi}^T$  and  $\tilde{\Phi}$  a *dual frame*. Then

$$\Phi \tilde{\Phi}^T = I_2. \quad (4.20)$$

We have seen one possible dual frame in (2.16c),<sup>29</sup>

$$\tilde{\Phi} = \begin{bmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \end{bmatrix},$$

with the associated expansion

$$x \stackrel{(a)}{=} \sum_{i=0}^2 \langle x, \tilde{\varphi}_i \rangle \varphi_i \stackrel{(b)}{=} \sum_{i=0}^2 \langle x, \varphi_i \rangle \tilde{\varphi}_i, \quad (4.21)$$

where we expressed  $x$  both in the frame (a) and the dual frame (b). This looks exactly like the usual biorthogonal expansion (2.111a), except for the number of vectors involved. The dual frame vectors are:

$$\tilde{\varphi}_0 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad \tilde{\varphi}_1 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \quad \tilde{\varphi}_2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}. \quad (4.22)$$

**Geometry of the Expansion** In the previous example, the geometry of the expansion was captured by  $\Phi$  and  $\Phi^\perp$  as in (4.8); here, we must add the dual frame  $\tilde{\Phi}$  and its complement  $\tilde{\Phi}^\perp$ . Two possible complements corresponding to  $\Phi$  and  $\tilde{\Phi}$  are

$$\Phi^\perp = [1 \ 1 \ 1], \quad \tilde{\Phi}^\perp = [1 \ 1 \ -1] \quad (4.23)$$

both of size  $1 \times 3$ . Then, the following capture the geometry of the matrices involved:

$$\Phi \tilde{\Phi}^T = I_{2 \times 2}, \quad (4.24a)$$

$$\Phi (\tilde{\Phi}^\perp)^T = 0_{2 \times 1}, \quad (4.24b)$$

$$\Phi^\perp \tilde{\Phi}^T = 0_{1 \times 2}, \quad (4.24c)$$

$$\Phi^\perp (\tilde{\Phi}^\perp)^T = I_{1 \times 1} = 1. \quad (4.24d)$$

Thus,  $\mathbb{R}^3$  is spanned by both  $\Phi^T \oplus (\Phi^\perp)^T$  and  $\tilde{\Phi}^T \oplus (\tilde{\Phi}^\perp)^T$ .

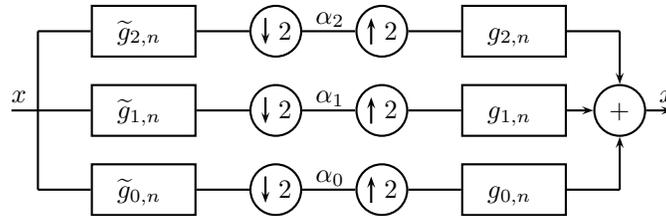
**Energy of Expansion Coefficients** We saw how energy is conserved in a Parseval-like manner before, what can we say about  $\|\alpha\|$  here?

$$\|\alpha\|^2 = \alpha^T \alpha = x^T \tilde{\Phi} \tilde{\Phi}^T x = x^T U \Sigma U^T x,$$

where we have performed a singular value decomposition (2.222) on the Hermitian matrix  $\tilde{\Phi} \tilde{\Phi}^T$  via (2.232a) as

$$\tilde{\Phi} \tilde{\Phi}^T = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} = \underbrace{\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}}_U \underbrace{\begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}}_\Sigma \underbrace{\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}}_{U^T}. \quad (4.25a)$$

<sup>29</sup>Note that while there exist infinitely many dual frames since  $\Phi$  has a nontrivial null space, here we concentrate on the canonical one as will be clear later in the chapter.



**Figure 4.4:** A 3-channel filter bank with sampling by 2 implementing a general frame expansion.

Because  $\tilde{\Phi}\tilde{\Phi}^T$  is a Hermitian matrix, (2.234) holds, that is,

$$\lambda_{\min}I \leq \tilde{\Phi}\tilde{\Phi}^T \leq \lambda_{\max}I, \quad (4.25b)$$

where  $\lambda_{\min}$  and  $\lambda_{\max}$  are the smallest and largest eigenvalues of  $\tilde{\Phi}\tilde{\Phi}^T$ . Thus, with  $\lambda_{\min} = 1$ ,  $\lambda_{\max} = 3$ , we get

$$\|x\|^2 \leq \|\alpha\|^2 = \|\tilde{\Phi}^T x\|^2 \leq 3\|x\|^2. \quad (4.26)$$

Therefore, although energy is not preserved, it is bounded from below and above by the eigenvalues of  $\tilde{\Phi}\tilde{\Phi}^T$ . Depending on the range between the minimum and maximum eigenvalues, the energy can fluctuate; in general, the closer (tighter) the eigenvalues are, the better-behaved the frame is.<sup>30</sup> The set of inequalities (4.26) is similar to how Riesz bases were defined in (2.86), and a similar relation holds for  $\Phi$ ,

$$\frac{1}{\lambda_{\max}}I \leq \Phi\Phi^T \leq \frac{1}{\lambda_{\min}}I,$$

and thus

$$\frac{1}{3}\|x\|^2 \leq \|\Phi^T x\|^2 \leq \|x\|^2.$$

This frame is related to the previous one in the same way a biorthogonal basis is related to an orthonormal basis, and is called a *general frame*.

**Filter-Bank Implementation** In parallel to what we have done for (4.1), we can use this finite-dimensional frame as an expansion for sequences in  $\ell^2(\mathbb{Z})$  by slicing the input sequence into pieces of length 2 and applying the frame we just saw to each of these. To form a frame for  $\ell^2(\mathbb{Z})$ , we form three template frame vectors

<sup>30</sup>This explains the word *tight* in tight frames (where the eigenvalues are equal).

from the three  $\mathbb{R}^2$  vectors (4.18)

$$\varphi_0 = \begin{bmatrix} \vdots \\ 0 \\ \boxed{1} \\ 0 \\ 0 \\ \vdots \end{bmatrix}, \quad \varphi_1 = \begin{bmatrix} \vdots \\ 0 \\ \boxed{0} \\ 1 \\ 0 \\ \vdots \end{bmatrix}, \quad \varphi_2 = \begin{bmatrix} \vdots \\ 0 \\ \boxed{-1} \\ -1 \\ 0 \\ \vdots \end{bmatrix}. \quad (4.27)$$

To form the dual frame, we form three template vectors from the three  $\mathbb{R}^2$  dual frame vectors, (4.22), leading to the frame  $\Phi$  and dual frame  $\tilde{\Phi}$ :

$$\Phi = \{\varphi_{0,n-2k}, \varphi_{1,n-2k}, \varphi_{2,n-2k}\}_{k \in \mathbb{Z}}, \quad \tilde{\Phi} = \{\tilde{\varphi}_{0,n-2k}, \tilde{\varphi}_{1,n-2k}, \tilde{\varphi}_{2,n-2k}\}_{k \in \mathbb{Z}}.$$

Renaming the template frame sequences  $\varphi_{i,n} = g_{i,n}$  and the dual ones  $\tilde{\varphi}_{i,n} = \tilde{g}_{i,-n}$ , we again have a 3-channel filter bank with down/upsampling by 2, as in Figure 4.4.

### Choosing the Frame Expansion and Expansion Coefficients

So far, we have seen two redundant representations, a tight frame (4.3), akin to an orthonormal basis, and a general frame (4.19), akin to a biorthogonal basis. We showed properties, including robustness to noise and loss. Given a sequence  $x$ , how do we then choose an appropriate frame expansion? Moreover, as we have already mentioned, we can have infinitely many dual frames, and thus, infinitely many expansion coefficients  $\alpha'$ , which one to choose? We tackle these questions in the next section; here, we just show a simple example that indicates the trade-offs.

**Choosing the Frame Expansion** Assume we are working in  $\mathbb{R}^N$  and we are given an input sequence  $x$  consisting of a single complex sinusoidal sequence of unknown frequency  $(2\pi/N)\ell$  and a Kronecker delta sequence of unknown location  $k$ :

$$x_n = \beta_1 e^{j(2\pi/N)\ell n} + \beta_2 \delta_{n-k}.$$

As discussed in *Chapter 7*, we can use a length- $N$  DFT to expand  $x$ ; we know this will effectively localize the sinusoid in frequency, but will do a poor job in time, and the location of the Kronecker delta impulse will be essentially lost. We can use the dual, standard basis, with the dual effect: it will do an excellent job of localizing the Kronecker delta impulse but will fail in localizing the frequency of the sinusoid.

While we could use a wavelet representation from *Chapter 3*, an even more obvious option is to use both bases at the same time, effectively creating a frame with the following  $2N$  frame vectors:

$$\Phi = [\text{DFT} \quad I]_{N \times 2N}, \quad (4.28)$$

where the first  $N$  are the DFT basis vectors (3.162),

$$\varphi_i = \frac{1}{\sqrt{N}} \begin{bmatrix} W_N^0 & W_N^i & \dots & W_N^{i(N-1)} \end{bmatrix}^T,$$

while the last  $N$  are the standard basis vectors  $\delta_{n-i}$ . Using (2.144), we see that this is a tight frame since<sup>31</sup>

$$\Phi \Phi^* = \begin{bmatrix} \text{DFT} & I \end{bmatrix} \begin{bmatrix} \text{DFT}^* \\ I \end{bmatrix} = \underbrace{\text{DFT} \text{DFT}^*}_I + I = 2I. \quad (4.29)$$

**Choosing the Expansion Coefficients** As  $x$  has only two components, there exists a way to write it as

$$x = \Phi \alpha',$$

where  $\alpha' = \tilde{\Phi}^* x$  has exactly 2 nonzero coefficients<sup>32</sup>

$$\alpha' = [0 \ \dots \ 0 \ \beta_1 \ 0 \ \dots \ 0 \ \beta_2 \ 0 \ \dots \ 0]^T,$$

where  $\beta_1$  is at the  $\ell$ th location and  $\beta_2$  at the  $(N+k)$ th location. Such an expansion is called *sparse*, in the sense that it uses a small number of frame vectors. This is different from  $\alpha$  obtained from the canonical dual  $\tilde{\Phi} = (1/2)\Phi$ ,

$$\alpha = \frac{1}{2} \Phi^* x,$$

which has two dominant components at the same locations as  $\alpha'$ , but also many more nonzero components. We will see later that, while  $\alpha'$  has fewer nonzero coefficients,  $\alpha$  has a smaller  $\ell^2$  norm (see Solved Exercise ??).<sup>33</sup> This is an important message; while in bases, the expansion coefficients are always unique, in frames they are not, and minimizing different norms will lead to different expansions.

## Chapter Outline

This chapter is somewhat unusual in its scope. While most of the chapters in Part II deal either with Fourier- or wavelet-like expansions, this chapter deals with both. However, there is one important distinction: these expansions are all overcomplete, or, redundant. Thus, our decision to keep them all in one chapter.

Unlike for bases, where we have discussed standard finite-dimensional expansions such as the DFT, we have not done so with frames until now, and thus, Section 4.2 investigates finite-dimensional frames. We then resume the structure we have been following starting with Chapter 1, that is, we discuss the signal-processing vehicle for implementing frame expansions—oversampled filter banks. We follow with local Fourier frames in Section 4.4 and wavelet frames in Section 4.5. Section 4.6 concludes with computational aspects.

The sections that follow can also be seen as redundant counterparts of previous chapters. For example, Section 4.2 on finite-dimensional frames, has its basis counterpart in *Chapter 2* and *Chapter 3*, where we discussed finite-dimensional bases in general (*Chapter 2*) as well as some specific ones, such as the DFT (*Chapter 3*).

<sup>31</sup>Note that now we use the Hermitian transpose of  $\Phi$  as it contains complex entries.

<sup>32</sup>Although it might not be obvious how to calculate that expansion.

<sup>33</sup>In fact,  $\alpha'$  can be chosen to minimize the  $\ell^1$  norm, while  $\alpha$  minimizes the  $\ell^2$  norm.

Section 4.3 on oversampled filter banks has its basis (critically-sampled filter bank) counterpart in Chapter 1 (two-channel critically-sampled filter banks), Chapter 2 ( $N$ -channel critically-sampled filter banks) and Chapter 3 (tree-structured critically-sampled filter banks). Section 4.4 on local Fourier frames has its basis counterpart in Chapter 2 (local Fourier bases on sequences), while Section 4.5 on wavelet frames has its basis counterpart in Chapter 3 (wavelet bases on sequences). Thus, this chapter also plays a unifying role in summarizing concepts on expansions of sequences. The two chapters that follow this one will deal with functions instead of sequences.

## 4.2 Finite-Dimensional Frames

We have just seen examples showing that finite-dimensional overcomplete sets of vectors have properties similar to orthonormal and/or biorthogonal bases. We will now look into general properties of such finite-dimensional frames in  $\mathbb{C}^N$ , with the understanding that  $\mathbb{R}^N$  is just a special case. We start with tight frames and follow with general frames. Since the representation in a frame is in general nonunique, we discuss how to compute expansion coefficients, and point out that, depending on which norm is minimized, different solutions are obtained.

Finite-dimensional frames are represented via rectangular matrices; thus, all the material in this section is basic linear algebra. We use this simplicity to develop the geometric intuition to be carried to infinite-dimensional frames that follow.

### 4.2.1 Tight Frames for $\mathbb{C}^N$

We work in a finite-dimensional space  $\mathbb{C}^N$ , where, a set of vectors  $\Phi = \{\varphi_i\}_{i=0}^{M-1}$ ,  $M > N$ , is a frame represented (similarly to (4.3) and (4.19)) by the frame matrix  $\Phi$  as

$$\Phi = [\varphi_0 \ \varphi_1 \ \dots \ \varphi_{M-1}]_{N \times M}. \quad (4.30)$$

Assume that the  $\text{rank}(\Phi) = N$ , that is, the column range of  $\Phi$  is  $\mathbb{C}^N$ . Thus, any  $x \in \mathbb{C}^N$  can be written as a nonunique linear combination of  $\varphi_i$ 's.

We impose a further constraint and start with frames that satisfy a Parseval-like equality:

**DEFINITION 4.1 (TIGHT FRAME)** A family  $\Phi = \{\varphi_i\}_{i=0}^{M-1}$  in  $\mathbb{C}^N$  is called a *tight frame*, or,  $\lambda$ -*tight frame* when there exists a constant  $0 < \lambda < \infty$  called the *frame bound*, such that for all  $x \in \mathbb{C}^N$ ,

$$\lambda \|x\|^2 = \sum_{i=0}^{M-1} |\langle x, \varphi_i \rangle|^2 = \|\Phi^* x\|^2. \quad (4.31)$$

**Expansion**

Equation (4.31) has a number of consequences. First, it means that

$$\Phi \Phi^* = \lambda I. \quad (4.32)$$

Thus,  $\Phi^*$  is a right inverse of  $\Phi$ . Calling  $\tilde{\Phi}$  the dual frame as before, we see that

$$\tilde{\Phi} = \frac{1}{\lambda} \Phi. \quad (4.33)$$

Then:

$$x = \Phi \alpha, \quad \alpha = \frac{1}{\lambda} \Phi^* x, \quad (4.34a)$$

$$x = \frac{1}{\lambda} \sum_{i=0}^{M-1} \langle x, \varphi_i \rangle \varphi_i. \quad (4.34b)$$

This looks very similar to an orthonormal basis expansion, except for the scaling factor  $(1/\lambda)$  and the fact that  $\Phi = \{\varphi_i\}_{i=0}^{M-1}$  cannot be a basis since  $\varphi_i$  are not linearly independent. We can pull the factor  $(1/\lambda)$  into the sum and renormalize the frame vectors as  $\varphi'_i = (1/\sqrt{\lambda})\varphi_i$  leading to the expression that formally looks identical to that of an orthonormal basis expansion:

$$x = \sum_{i=0}^{M-1} \langle x, \varphi'_i \rangle \varphi'_i. \quad (4.35)$$

We have already seen an example of such a renormalization in (4.1) and (4.13). A frame normalized so that  $\lambda = 1$  is called a *Parseval* tight frame or a 1-tight frame.

The expression for  $x$  is what is typically called a *reconstruction* or a *representation* of a sequence, or, in filter banks, *synthesis*, while the expression for the expansion coefficients  $\alpha$  is a *decomposition*, or, *analysis* in filter banks.

In the discussion above, we said nothing about the norms of the individual frame vectors. Since the analysis computes inner products  $\alpha_i = \langle x, \varphi_i \rangle$ , it often makes sense for all  $\varphi_i$  to have the same norm, leading to an *equal-norm* frame (which may not be tight). When we combine equal norm with tightness, we get a frame that acts every bit like an orthonormal basis, except for the redundancy. That is, all inner products  $\alpha_i = \langle x, \varphi_i \rangle$  are projections of  $x$  onto vectors of the same norm, allowing us to compare coefficients  $\alpha_i$  to each other. Moreover, because the frame is tight, the right inverse is simply its adjoint (within scaling). In finite dimensions, tightness corresponds to the rows of  $\Phi$  being orthogonal. Because of this, it is hard in general to obtain an equal-norm frame starting from the tight one.

**Geometry of the Expansion** Let us explore the geometry of tight frames. With the frame matrix  $\Phi$  as in (4.30), and as we did in (4.8), we introduce  $\Phi^\perp$ ,

$$\Phi^\perp = [\varphi_0^\perp \quad \varphi_1^\perp \quad \dots \quad \varphi_{M-1}^\perp]_{(M-N) \times M} \quad (4.36)$$

as one possible orthogonal complement of  $\Phi$  in  $\mathbb{C}^M$ :

$$\begin{bmatrix} \Phi \\ \Phi^\perp \end{bmatrix} = \begin{bmatrix} \varphi_0 & \varphi_1 & \cdots & \varphi_{M-1} \\ \varphi_0^\perp & \varphi_1^\perp & \cdots & \varphi_{M-1}^\perp \end{bmatrix}_{M \times M} \quad (4.37)$$

that is, the rows of  $\Phi^\perp$  are chosen to be orthogonal to the rows of  $\Phi$ , orthogonal to each other and of norm 1, or,

$$\begin{aligned} \begin{bmatrix} \Phi \\ \Phi^\perp \end{bmatrix} \begin{bmatrix} \Phi^* & (\Phi^\perp)^* \end{bmatrix} &= \begin{bmatrix} \Phi\Phi^* & \Phi(\Phi^\perp)^* \\ \Phi^\perp\Phi^* & \Phi^\perp(\Phi^\perp)^* \end{bmatrix} \\ &= \begin{bmatrix} I_{N \times N} & 0 \\ 0 & I_{(M-N) \times (M-N)} \end{bmatrix} = I_{M \times M}. \end{aligned} \quad (4.38)$$

Note that each vector  $\varphi_i$  is in  $\mathbb{C}^N$ , while each vector  $\varphi_i^\perp$  is in  $\mathbb{C}^{M-N}$ . We can rewrite (4.37) as

$$S = \text{span}(\Phi^T) \subset \mathbb{C}^M, \quad (4.39a)$$

$$S^\perp = \text{span}((\Phi^\perp)^T) \subset \mathbb{C}^M, \quad (4.39b)$$

$$\mathbb{C}^M = S \oplus S^\perp, \quad (4.39c)$$

and, because of (4.38),

$$\Phi\Phi^* = I_{N \times N}, \quad (4.40a)$$

$$\Phi(\Phi^\perp)^* = 0_{N \times (M-N)}, \quad (4.40b)$$

$$\Phi^\perp(\Phi^\perp)^* = I_{(M-N) \times (M-N)}. \quad (4.40c)$$

A vector of expansion coefficients  $\alpha' \in \mathbb{C}^M$  can be written as

$$\alpha' = \alpha + \alpha^\perp \quad (4.41a)$$

with  $\alpha \in S$  and  $\alpha^\perp \in S^\perp$ , and thus

$$\langle \alpha, \alpha^\perp \rangle = 0, \quad (4.41b)$$

as we have already seen in the simple example in the previous section, (4.11b).

### Relation to Orthonormal Bases

We now look into connections between tight frames and orthonormal bases.

**THEOREM 4.2** A 1-tight frame with unit-norm vectors is an orthonormal basis.

*Proof.* For a tight frame expansion with  $\lambda = 1$ ,

$$T = \Phi\Phi^* = I,$$

and thus, all the eigenvalues of  $T$  are equal to 1. Using one of the useful frame relations we introduce later in (4.64a),

$$N \stackrel{(a)}{=} \sum_{j=0}^{N-1} \lambda_j \stackrel{(b)}{=} \sum_{i=0}^{M-1} \|\varphi_i\|^2 \stackrel{(c)}{=} M,$$

where (a) follows from all eigenvalues being equal to 1; (b) from (4.64a); and (c) from all frame vectors being of unit norm. We get that  $M = N$ , and thus, our frame is an orthonormal basis.

In this result, we see once more the tantalizing connection between tight frames and orthonormal bases. In fact, even more is true: tight frames and orthonormal bases arise from the minimization of the same quantity called the *frame potential*:

$$\text{FP}(\Phi) = \sum_{i,j=0}^{M-1} |\langle \varphi_i, \varphi_j \rangle|^2. \quad (4.42)$$

In fact, minimizing the frame potential has two possible outcomes:

- (i) When  $M \leq N$ , the minimum value of the frame potential is

$$\text{FP}(\Phi) = N,$$

achieved when  $\Phi$  is an orthonormal set.

- (ii) When  $M > N$ , the minimum value of the frame potential is

$$\text{FP}(\Phi) = \frac{M^2}{N},$$

achieved when  $\Phi$  is a unit-norm tight frame.<sup>34</sup>

This tells us that unit-norm tight frames are a natural extension of orthonormal bases, that is, the theorem formalizes the intuitive notion that unit-norm tight frames are a generalization of orthonormal bases. Moreover, both orthonormal bases and unit-norm tight frames are results of the minimization of the frame potential, with different parameters (number of elements equal/larger than the dimension of the space). We give pointers to more details on this topic in *Further Reading*.

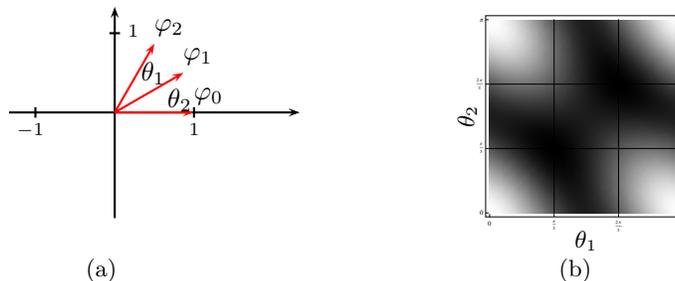
**EXAMPLE 4.1 (TIGHT FRAMES AND ORTHONORMAL BASES)** We illustrate this result with an example. Fix  $N = 2$ .

- (i) We first consider the case when  $M = N = 2$ . Then, we have two vectors only,  $\varphi_0$  and  $\varphi_1$ , both on the unit circle. According to (4.42), the frame potential is

$$\text{FP}(\{\varphi_0, \varphi_1\}) = \|\varphi_0\|^2 + \|\varphi_1\|^2 + 2|\langle \varphi_0, \varphi_1 \rangle|^2 \stackrel{(a)}{=} 2(1 + |\langle \varphi_0, \varphi_1 \rangle|^2), \quad (4.43)$$

where (a) follows from  $\varphi_0, \varphi_1$ , being of unit norm. The above expression is minimized when  $\langle \varphi_0, \varphi_1 \rangle = 0$ , that is, when  $\varphi_0$  and  $\varphi_1$  form an orthonormal basis. In that case, the minimum of the frame potential is  $FP = 2 = N$ .

<sup>34</sup>This lower bound for frames is known as *Welch bound* arising when minimizing interuser interference in a CDMA system (see *Further Reading* for pointers).



**Figure 4.5:** Minimization of the frame potential for frames with unit-norm vectors. (a) Three unit-norm vectors in  $\mathbb{R}^2$ . (b) Density plot of the frame potential as a function of angles between frame vectors. The two minima are identical and appear for  $\theta_1 = \pi/3$ ,  $\theta_2 = \pi/3$  and  $\theta_1 = 2\pi/3$ ,  $\theta_2 = 2\pi/3$ .

- (ii) We now look at  $M$  larger than  $N$ ; we choose  $M = 3$ . Let us fix  $\varphi_0 = [1 \ 0]$ ;  $\varphi_1$  is  $\theta_1$  away from  $\varphi_0$  in counterclockwise direction;  $\varphi_2$  is  $\theta_2$  away from  $\varphi_1$  in counterclockwise direction (see Figure 4.5(a)). The frame potential is now

$$\text{FP}(\{\theta_1, \theta_2\}) = \|\varphi_0\|^2 + \|\varphi_1\|^2 + \|\varphi_2\|^2 + \quad (4.44)$$

$$2(|\langle \varphi_0, \varphi_1 \rangle|^2 + |\langle \varphi_0, \varphi_2 \rangle|^2 + |\langle \varphi_1, \varphi_2 \rangle|^2) \\ = 3 + 2(\cos \theta_1 + \cos \theta_2 + \cos(\theta_1 + \theta_2)). \quad (4.45)$$

Figure 4.5(b) shows the density plot of  $\text{FP}(\{\theta_1, \theta_2\})$  for  $\theta_i \in [0, \pi]$ . From the figure, we see that there are two minima, for  $\theta_1 = \theta_2 = \pi/3$  and  $\theta_1 = \theta_2 = 2\pi/3$ , both of which lead to tight frames; the second choice is the frame we have seen in (4.13), the first choice is, in fact, identical to the second (within reflection). We thus see that the results of minimizing the frame potential in this case are tight frames, with the minimum of

$$\text{FP}\left(\left\{\frac{\pi}{3}, \frac{\pi}{3}\right\}\right) = \text{FP}\left(\left\{2\frac{\pi}{3}, 2\frac{\pi}{3}\right\}\right) = 3 + 2\left(\frac{1}{4} + \frac{1}{4} + \frac{1}{4}\right) = \frac{9}{2} = \frac{M^2}{N},$$

as per the theorem.

This simple example shows that minimizing the frame potential with different parameters leads to either orthonormal sets (orthonormal bases when  $M = N$ ) or unit-norm tight frames.

**Naimark's Theorem** Another powerful connection between orthonormal bases and tight frames is also a constructive way to obtain all tight frames. It is given by the following theorem, due to Naimark, which says that all tight frames can be obtained by projecting orthonormal bases from a larger space (of dimension  $M$ ) onto a smaller one (of dimension  $N$ ). We have seen one such example in (4.8), where the frame  $\Phi \in \mathbb{R}^2$  from (4.3) was obtained by projecting an orthonormal basis from  $\mathbb{R}^3$ .

**THEOREM 4.3** (NAIMARK [3], HAN & LARSON [42]) A frame  $\Phi \in \mathbb{C}^N$  is tight if and only if there exists an orthonormal basis  $\Psi \in \mathbb{C}^M$ ,  $M \geq N$ , such that

$$\Phi^* = \Psi[J], \quad (4.46)$$

where  $J \subset \{0, 1, \dots, M-1\}$  is the index set of the retained columns of  $\Psi$ , a process known as *seeding*.

Here, we considered only the tight-frame finite-dimensional instantiation of the theorem. For general finite-dimensional frames, a similar result holds, that is, any frame can be obtained by projecting a biorthogonal basis from a larger space. In Theorem 4.8 we formulate the statement for infinite-dimensional frames implementable by oversampled filter banks.

*Proof.* Given is a tight frame  $\Phi$ , with columns  $\varphi_i$ ,  $i = 0, 1, \dots, M-1$ , and rows  $\psi_j$ ,  $j = 0, 1, \dots, N-1$ . Because  $\Phi$  is a tight frame, it satisfies (4.32); without loss of generality, renormalize it by  $(1/\sqrt{\lambda})$  so that the frame we work with is 1-tight. This further means that

$$\langle \psi_i, \psi_j \rangle = \delta_{i-j},$$

that is,  $\{\psi_0, \psi_1, \dots, \psi_{N-1}\}$  is an orthonormal set, and, according to (4.39a), it spans the subspace  $S \subset \mathbb{C}^M$ . The whole proof in this direction follows from the geometry of tight frames we discussed earlier, by showing, as we did in (4.37), how to complete the tight frame matrix  $\Phi$  to obtain an orthonormal basis  $\Psi^*$ .

The other direction is even easier. Assume we are given a unitary  $\Psi$ . Choose any  $N$  columns of  $\Psi$  and call the resulting  $M \times N$  matrix  $\Phi^*$ . Because these columns form an orthonormal set, the rows of  $\Phi$  form an orthonormal set, that is,

$$\Phi\Phi^* = I.$$

Therefore,  $\Phi$  is a tight frame.

**Harmonic Tight Frames** We now look into an example of a well-known family of tight frames called *harmonic tight frames*, a representative of which we have already seen in (4.3). Harmonic tight frames are frame counterparts of the DFT, and are, in fact, obtained from the DFT by seeding, a process defined in Theorem 4.3.

Specifically, to obtain harmonic tight frames, we start with the DFT matrix  $\Psi = \text{DFT}_M$  given in (3.161a) and delete its last  $(M-N)$  columns, yielding:

$$\Phi = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & W_M & W_M^2 & \dots & W_M^{M-1} \\ 1 & W_M^2 & W_M^4 & \dots & W_M^{2(M-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_M^{(N-1)} & W_M^{(N-1) \cdot 2} & \dots & W_M^{(N-1)(M-1)} \end{bmatrix}, \quad (4.47a)$$

with the corresponding frame vectors

$$\varphi_i = \left[ W_M^0 \quad W_M^i \quad \dots \quad W_M^{i(N-1)} \right]^T, \quad (4.47b)$$

for  $i = 0, 1, \dots, M - 1$ , where  $W_M = e^{-j2\pi/M}$  is the principal  $M$ th root of unity (3.284). The norm of each frame vector is

$$\|\varphi_i\|^2 = \varphi_i^* \varphi_i = N,$$

the frame is  $M$ -tight

$$\Phi\Phi^* = MI,$$

and the Parseval-like equality is

$$\|\Phi^*x\|^2 = M\|x\|^2.$$

In its unit-norm version, we can compute its redundancy as  $(M/N)$ . Harmonic tight frames have a number of other interesting properties, some of which are explored in Exercise ??.

For example, we explore an interesting property of frames that holds for harmonic tight frames.

**DEFINITION 4.4 (FRAME MAXIMALLY ROBUST TO ERASURES)** A frame  $\Phi$  is called *maximally robust to erasures* when its every  $N \times N$  submatrix is invertible.

We have seen one example of a frame maximally robust to erasures: every  $2 \times 2$  submatrix of (4.3) is invertible. The motivation in that example was that such frames can sustain a loss of a maximum number of expansion coefficients and still afford perfect reconstruction of the original vector. In fact, harmonic tight frames in general possess such a property since every  $N \times N$  submatrix of (4.47a) is invertible (it is a Vandermonde matrix whose determinant is always nonzero, see (2.240) and Exercise ??).

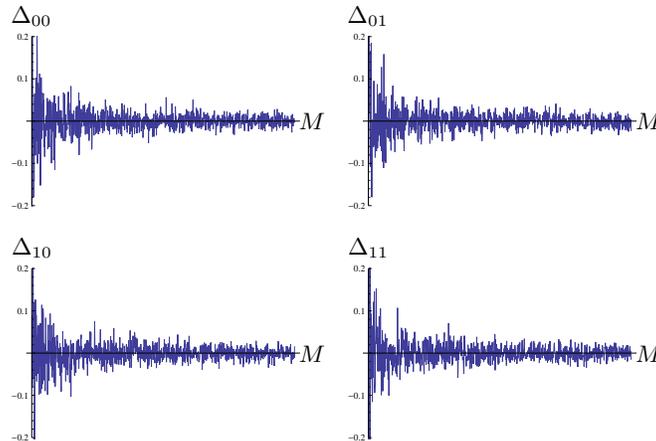
### Random Frames

While it seems that the tight frames as those we have just seen are very special, it turns out that any unit-norm frame with high redundancy will be almost tight. This is made precise by the following result:

**THEOREM 4.5 (TIGHTNESS OF RANDOM FRAMES [41])** Let  $\{\Phi_M\}_{M=N}^{\infty}$  be a sequence of frames in  $\mathbb{R}^N$  such that  $\Phi_M$  is generated by choosing  $M$  vectors independently with a uniform distribution on the unit sphere in  $\mathbb{R}^N$ . Then, in the mean-squared sense,

$$\frac{1}{M}\Phi\Phi^T \rightarrow \frac{1}{N}I_N \quad \text{elementwise as } M \rightarrow \infty.$$

An illustration of the theorem for  $N = 2$  is given in Figure 4.6.



**Figure 4.6:** Illustration of tightness of random frames for  $N = 2$ , and  $M = 2, 3 \dots, 1000$ . Since the convergence is elementwise, each graph plots the behavior  $\Delta_{ij} = [(1/M)\Phi\Phi^T - (1/2)I_2]_{ij}$  for  $i, j = 0, 1$ .

### 4.2.2 General Frames for $\mathbb{C}^N$

Tight frames are attractive the same way orthonormal bases are. They obey a Parseval-like energy conservation equality, and the dual frame is equal to the frame itself (the right inverse of  $\Phi$  is just its own Hermitian transpose, possibly within scaling). Tightness, however, has sometimes to be relaxed, just as orthonormality does (for example, when we wanted to design two-channel linear-phase filter banks in Chapter 1). Either a frame is given by a specific construction, and is not tight, or the constraints posed by tightness are too restrictive for a desired design.

Given a frame  $\Phi$  as in (4.30) of rank  $(\Phi) = N$ , we can find the *canonical* dual frame  $\tilde{\Phi}$  (formalized in Definition 4.7), also of size  $N \times M$ , made of dual frame vectors as

$$\tilde{\Phi} = (\Phi\Phi^*)^{-1}\Phi, \quad (4.48a)$$

$$= [\tilde{\varphi}_0 \quad \tilde{\varphi}_1 \quad \dots \quad \tilde{\varphi}_{M-1}]_{N \times M}, \quad (4.48b)$$

$$\tilde{\varphi}_i = (\Phi\Phi^*)^{-1}\varphi_i. \quad (4.48c)$$

The above are all well-defined, since  $T = \Phi\Phi^*$  is of rank  $N$  and can thus be inverted. Therefore,  $\Phi$  and  $\tilde{\Phi}$  play the same roles for frames as their namesakes do for biorthogonal bases, and, using (4.48a):

$$\Phi\tilde{\Phi}^* = \Phi\Phi^*(\Phi\Phi^*)^{-1} = I_N, \quad (4.49a)$$

$$\tilde{\Phi}\Phi^* = (\Phi\Phi^*)^{-1}\Phi\Phi^* = I_N. \quad (4.49b)$$

Note that the canonical dual frame  $\tilde{\Phi}$  chosen here is a particular right inverse; we will formalize this in Definition 4.7. Note also that when  $\Phi$  is tight, with this definition of the dual, we indeed obtain  $\tilde{\Phi} = \Phi$ .

**Expansion**

We know that  $\Phi = \{\varphi_i\}_{i=0}^{M-1}$  span  $\mathbb{C}^N$ , that is, any  $x \in \mathbb{C}^N$  can be written as

$$x = \sum_{i=0}^{M-1} \alpha_i \varphi_i \quad (4.50a)$$

$$= \sum_{i=0}^{M-1} \tilde{\alpha}_i \tilde{\varphi}_i, \quad (4.50b)$$

both of which follow from (4.49) by writing

$$x \stackrel{(a)}{=} \Phi \tilde{\Phi}^* x = \Phi \alpha \quad (4.51a)$$

$$\stackrel{(b)}{=} \tilde{\Phi} \tilde{\Phi}^* x = \tilde{\Phi} \tilde{\alpha}, \quad (4.51b)$$

where (a) leads to (4.50a) and (b) to (4.50b), respectively. Again, these are reconstruction (representation) of a sequence, or, in filter banks, synthesis.

We have used  $\alpha_i$  and  $\tilde{\alpha}_i$  liberally, defining them implicitly. It comes as no surprise that

$$\alpha_i = \langle x, \tilde{\varphi}_i \rangle \quad \alpha = \tilde{\Phi}^* x, \quad (4.52a)$$

$$\tilde{\alpha}_i = \langle x, \varphi_i \rangle \quad \tilde{\alpha} = \Phi^* x; \quad (4.52b)$$

they are interchangeable like the expansion expressions. As before, the expression for  $\alpha$  is sometimes called decomposition (or, analysis, in filter banks).

**Geometry of the Expansion** Similarly to tight frames, let us explore the geometry of general frames. For tight frames, we dealt with  $\Phi$  and  $\Phi^\perp$  as in (4.30),(4.37); here, we must add the dual frame  $\tilde{\Phi}$  and its complement  $\tilde{\Phi}^\perp$ :

$$\begin{array}{cc} \Phi & \tilde{\Phi} & N \times M \\ \Phi^\perp & \tilde{\Phi}^\perp & (M-N) \times M \end{array}$$

with, similarly to (4.38),

$$\Phi \tilde{\Phi}^* = I_{N \times N}, \quad (4.53a)$$

$$\Phi (\tilde{\Phi}^\perp)^* = 0_{N \times (M-N)}, \quad (4.53b)$$

$$\Phi^\perp \tilde{\Phi}^* = 0_{(M-N) \times N}, \quad (4.53c)$$

$$\Phi^\perp (\tilde{\Phi}^\perp)^* = I_{(M-N) \times (M-N)}. \quad (4.53d)$$

As in (4.39a),  $S$  is the subspace of  $\mathbb{C}^M$  spanned by the columns of  $\Phi^*$ , while  $S^\perp$  is the subspace of  $\mathbb{C}^M$  spanned by the columns of  $(\Phi^\perp)^*$ .<sup>35</sup> We will see when discussing projection operators shortly, that

$$\text{span}(\Phi^*) = \text{span}(\tilde{\Phi}^*), \quad (4.54a)$$

$$\text{span}((\Phi^\perp)^*) = \text{span}((\tilde{\Phi}^\perp)^*), \quad (4.54b)$$

<sup>35</sup>Note that the  $\text{span}(\Phi^*) = \text{span}(\Phi^T)$ .

and thus, from now on, we will use that

$$S = \text{span}(\Phi^*). \quad (4.55)$$

As before, an arbitrary vector of expansion coefficients  $\alpha' \in \mathbb{C}^M$  can be written as

$$\alpha' = \alpha + \alpha^\perp \quad (4.56a)$$

with  $\alpha \in S$  and  $\alpha^\perp \in S^\perp$ , and thus

$$\langle \alpha, \alpha^\perp \rangle = 0. \quad (4.56b)$$

**Frame Operator  $T$**  When calculating the dual frame, the so-called canonical dual of  $\Phi$ , the product  $\Phi\Phi^*$  is central; we call it

$$T_{N \times N} = \Phi\Phi^*. \quad (4.57)$$

It is a Hermitian and positive definite matrix (see (2.233)), and thus, all of its eigenvalues are real and positive. According to (2.232a),  $T$  can be diagonalized as

$$T = \Phi\Phi^* = U\Lambda U^*, \quad (4.58)$$

where  $\Lambda$  is a diagonal matrix of eigenvalues, and  $U$  a unitary matrix of eigenvectors. The largest  $\lambda_{\max}$  and smallest  $\lambda_{\min}$  eigenvalues play a special role. For tight frames,  $\lambda_{\max} = \lambda_{\min} = \lambda$ , and  $T$  is a scaled identity,  $T = \lambda I$ , as it possesses a single eigenvalue  $\lambda$  of multiplicity  $N$ .

**Energy of Expansion Coefficients** In the examples in the introduction as well as for tight frames earlier, we have seen how the energy of the expansion coefficients is conserved or bounded, (4.12), (4.26), and (4.31), respectively. We now look into it for general frames by computing the energy of the expansion coefficients  $\tilde{\alpha}$  as

$$\|\tilde{\alpha}\|^2 = \tilde{\alpha}^* \tilde{\alpha} \stackrel{(a)}{=} x^* \Phi\Phi^* x \stackrel{(b)}{=} x^* U\Lambda U^* x, \quad (4.59)$$

where (a) follows from (4.51b) and (b) from (4.57). Thus, using (4.25b),

$$\lambda_{\min} \|x\|^2 \leq \|\tilde{\alpha}\|^2 \leq \lambda_{\max} \|x\|^2. \quad (4.60a)$$

Therefore, the energy, while not preserved, is bounded from below and above by the eigenvalues of  $T$ . How close (tight) these eigenvalues are will influence the quality of the frame in question, as we will see later. The same argument above can be repeated for  $\|\alpha\|$ , leading to

$$\frac{1}{\lambda_{\max}} \|x\|^2 \leq \|\alpha\|^2 \leq \frac{1}{\lambda_{\min}} \|x\|^2. \quad (4.60b)$$

**Relation to Tight Frames** Given a general frame  $\Phi$ , we can easily transform it into a tight frame  $\Phi'$ . We do this by diagonalizing  $T$  as in (4.58). Then the tight frame is obtained as

$$\Phi' = U\Lambda^{-1/2}U^* \Phi.$$

### Frame Operators

The pair of inequalities (4.60a) leads to an alternate definition of a frame, similar in spirit to *Definition 2.43* for biorthogonal bases:

**DEFINITION 4.6 (FRAME)** A family  $\Phi = \{\varphi_i\}_{i=0}^{M-1}$  in  $\mathbb{C}^N$  is called a *frame* when there exist two constants  $0 < \lambda_{\min} \leq \lambda_{\max} < \infty$ , such that for all  $x \in \mathbb{C}^N$ ,

$$\lambda_{\min}\|x\|^2 \leq \sum_{i=0}^{M-1} |\langle x, \varphi_i \rangle|^2 \leq \lambda_{\max}\|x\|^2, \quad (4.61)$$

where  $\lambda_{\min}$ ,  $\lambda_{\max}$  are called *lower* and *upper frame bounds*.

Because of (4.60a), the frame bounds are clearly the eigenvalues of  $T$  as we have seen previously. From the definition, we can also understand the meaning of  $\lambda$  we have seen in (4.31); tight frames are obtained when the two frame bounds are equal, that is, when  $\lambda_{\max} = \lambda_{\min} = \lambda$ .

The operators we have seen so far are sometimes called: *analysis frame operator*  $\tilde{\Phi}^*$ , *synthesis frame operator*  $\Phi$ , and *frame operator*  $T = \Phi\Phi^*$ . The analysis frame operator is one of many, as there exist infinitely many dual frames for a given frame  $\Phi$ . In our finite-dimensional setting, the analysis frame operator maps an input  $x \in \mathbb{C}^N$  onto a subspace of  $\mathbb{C}^M$ , namely  $\alpha = \tilde{\Phi}^*x$  belongs to the subspace  $S$  spanned by the columns of  $\tilde{\Phi}^*$  as we have seen in (4.55).<sup>36</sup> These, together with other frame operators introduced shortly, are summarized in Table 4.1.

Given  $x \in \mathbb{C}^N$ , the frame operator  $T = \Phi\Phi^*$  is a linear operator from  $\mathbb{C}^N$  to  $\mathbb{C}^N$ , guaranteed to be of full rank  $N$  since (4.61) ensures that  $\lambda_{\min} > 0$ . Also,

$$\lambda_{\min}I \leq T = \Phi\Phi^* \leq \lambda_{\max}I. \quad (4.62)$$

On the other hand, given  $x \in \mathbb{C}^M$ , the operator  $\Phi^*\Phi$ , which we have seen before as a projection operator (4.10) in our simple example, maps the input onto a

<sup>36</sup>Remember that  $S$  is spanned by either  $\Phi^*$  or  $\tilde{\Phi}^*$ .

subspace  $S$  of  $\mathbb{C}^M$ . We called that operator a *Gram* operator in (2.118),  $G = \Phi^* \Phi$ :

$$\begin{aligned}
 G &= \begin{bmatrix} \varphi_0^* \\ \varphi_1^* \\ \vdots \\ \varphi_{M-1}^* \end{bmatrix} [\varphi_0 \ \varphi_1 \ \dots \ \varphi_{M-1}] \\
 &= \begin{bmatrix} \langle \varphi_0, \varphi_0 \rangle & \langle \varphi_0, \varphi_1 \rangle & \dots & \langle \varphi_0, \varphi_{M-1} \rangle \\ \langle \varphi_1, \varphi_0 \rangle & \langle \varphi_1, \varphi_1 \rangle & \dots & \langle \varphi_1, \varphi_{M-1} \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \varphi_{M-1}, \varphi_0 \rangle & \langle \varphi_{M-1}, \varphi_1 \rangle & \dots & \langle \varphi_{M-1}, \varphi_{M-1} \rangle \end{bmatrix}_{M \times M} \\
 &= \begin{bmatrix} \|\varphi_0\|^2 & \langle \varphi_0, \varphi_1 \rangle & \dots & \langle \varphi_0, \varphi_{M-1} \rangle \\ \langle \varphi_0, \varphi_1 \rangle^* & \|\varphi_1\|^2 & \dots & \langle \varphi_1, \varphi_{M-1} \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \varphi_0, \varphi_{M-1} \rangle^* & \langle \varphi_1, \varphi_{M-1} \rangle^* & \dots & \|\varphi_{M-1}\|^2 \end{bmatrix} = G^*. \quad (4.63)
 \end{aligned}$$

This matrix  $G$  contains correlations between different frame vectors, and, while of size  $M \times M$ , it is of rank  $N$  only.

The frame operator  $T = \Phi\Phi^*$  and the Gram operator  $G = \Phi^*\Phi$  have the same nonzero eigenvalues (see Section 2.B.2) and thus the same trace. This fact can be used to show that the sum of eigenvalues of  $T$  is equal to the sum of the norms of the frame vectors. We state this and three further useful frame facts here; their proofs are left for Exercise ??:

$$\sum_{j=0}^{N-1} \lambda_j = \sum_{i=0}^{M-1} \|\varphi_i\|^2, \quad (4.64a)$$

$$Tx = \sum_{i=0}^{M-1} \langle x, \varphi_i \rangle \varphi_i, \quad (4.64b)$$

$$\langle x, Tx \rangle = \sum_{i=0}^{M-1} |\langle x, \varphi_i \rangle|^2, \quad (4.64c)$$

$$\sum_{i=0}^{M-1} \langle \varphi_i, T\varphi_i \rangle = \sum_{i,j=0}^{M-1} |\langle \varphi_i, \varphi_j \rangle|^2. \quad (4.64d)$$

**Dual Frame Operators** We have already discussed the dual frame operator in (4.48a); we now formalize it a bit more.

**DEFINITION 4.7 (CANONICAL DUAL FRAME)** Given a frame satisfying (4.61), its canonical dual frame  $\tilde{\Phi}$  and dual frame vectors are:

$$\tilde{\Phi} = (\Phi\Phi^*)^{-1}\Phi = T^{-1}\Phi, \quad (4.65a)$$

$$\tilde{\varphi}_i = (\Phi\Phi^*)^{-1}\varphi_i = T^{-1}\varphi_i. \quad (4.65b)$$

From (4.60b), we can say that the dual frame  $\tilde{\Phi}$  is a frame with frame bounds  $(1/\lambda_{\max})$  and  $(1/\lambda_{\min})$ . We also see that

$$\tilde{T} = \tilde{\Phi}\tilde{\Phi}^* = T^{-1} \underbrace{\Phi\Phi^*}_T \underbrace{(T^{-1})^*}_{T^{-1}} = T^{-1}. \quad (4.66)$$

What is particular about this canonical dual frame is that among all right inverses of  $\Phi$ ,  $\tilde{\Phi}$  leads to the smallest expansion coefficients  $\alpha$  in Euclidean norm, as shown in Solved Exercise ???. We will also see later in this section, that expansion coefficients  $\alpha'$  other than the coefficients  $\alpha$  obtained from the canonical dual might be more appropriate when minimizing other norms (such as  $\ell^1$ - or  $\ell^\infty$  norms).

From (4.65b), we see that to compute those dual frame vectors, we need to invert  $T$ . While in finite dimensions, and for reasonable  $M$  and  $N$ , this is not a problem, it becomes an issue as  $M$  and  $N$  grow. In that case, the inverse can be computed via a series

$$T^{-1} = \frac{2}{\lambda_{\min} + \lambda_{\max}} \sum_{k=0}^{\infty} \left( I - \frac{2}{\lambda_{\min} + \lambda_{\max}} T \right)^k, \quad (4.67)$$

which converges faster when the frame bounds  $\lambda_{\min}$  and  $\lambda_{\max}$  are close, that is, when the frame is close to being tight. Solved Exercise ??? sketches a proof of (4.67), and Solved Exercise ??? illustrates it with examples.

**Projection Operators** We have seen various versions of frame operators, mapping  $\mathbb{C}^N$  to  $\mathbb{C}^N$ , as well as the Gram operator that maps  $\mathbb{C}^M$  to  $\mathbb{C}^M$ . We now look at two other operators,  $P = \tilde{\Phi}^*\Phi$  and  $\tilde{P} = \Phi^*\tilde{\Phi}$ . In fact, these are the same, as

$$P = \tilde{\Phi}^*\Phi = ((\Phi\Phi^*)^{-1}\Phi)^*\Phi = \Phi^*(\Phi\Phi^*)^{-1}\Phi = \Phi^*\tilde{\Phi} = \tilde{P}. \quad (4.68)$$

Therefore,  $P$  maps  $\mathbb{C}^M$  to a subspace of  $\mathbb{C}^M$ ,  $S$ , and is an orthogonal projection operator, as it is idempotent and self-adjoint (Definition 2.27):

$$P^2 = (\tilde{\Phi}^*\Phi) \underbrace{(\tilde{\Phi}^*\Phi)}_I \stackrel{(a)}{=} \tilde{\Phi}^*\Phi = P,$$

$$P^* = (\tilde{\Phi}^*\Phi)^* = \Phi^*\tilde{\Phi} \stackrel{(b)}{=} \Phi^*(T^{-1}\Phi) \stackrel{(c)}{=} (T^{-1}\Phi)^*\Phi = \tilde{\Phi}^*\Phi = P,$$

where (a) follows from (4.49a); (b) from (4.65a); and (c) from  $T$  being Hermitian and thus self-adjoint. This projection operator projects the input onto the column space of  $\tilde{\Phi}^*$ , or, since  $P$  and  $\tilde{P}$  are the same, onto the column space of  $\Phi^*$ . Table 4.1 summarizes various operators we have seen until now, Table 4.2 does so for frame expansions, Table 4.3 summarizes various classes of frames and their properties, while Figure 4.7 does so pictorially.

### 4.2.3 Choosing the Expansion Coefficients

Given a frame  $\Phi$  and a vector  $x$ , we have seen in (4.52a) that the expansion coefficients are given by  $\alpha = \tilde{\Phi}^*x$ ; for a tight frame, this reduces to  $\alpha = \Phi^*x$ .

Operator	Symbol	Expression	Size
Synthesis frame operator	$\Phi$		$N \times M$
Dual (analysis) frame operator	$\tilde{\Phi}$	$(\Phi\Phi^*)^{-1}\Phi$	$N \times M$
Frame operator	$T$	$\Phi\Phi^*$	$N \times N$
Gram operator	$G$	$\Phi^*\Phi$	$M \times M$
Projection operator	$P$	$\tilde{\Phi}^*\Phi$	$M \times M$

Table 4.1: Frame operators.

Expansion	
In $\Phi$	$x = \Phi\alpha \quad \alpha = \tilde{\Phi}^*x \quad \Phi\tilde{\Phi}^* = I \quad \lambda_{\min}I \leq T \leq \lambda_{\max}I$
In $\tilde{\Phi}$	$x = \tilde{\Phi}\tilde{\alpha} \quad \tilde{\alpha} = \Phi^*x \quad \tilde{\Phi}\Phi^* = I \quad (1/\lambda_{\max})I \leq T^{-1} \leq (1/\lambda_{\min})I$

Table 4.2: Frame expansions.

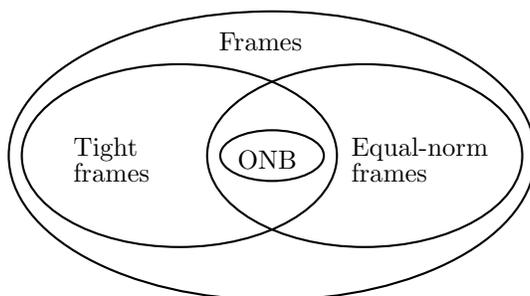


Figure 4.7: Frames at a glance. Tight frames with  $\lambda = 1$  and unit-norm vectors lead to orthonormal bases.

For frames, because  $\Phi$  has a nontrivial null space, there exists an infinite set of possible expansion coefficients (see also Solved Exercise ??). That is, given a frame  $\Phi$  and its canonical dual  $\tilde{\Phi}$  from (4.65a), from (4.56a), we can write  $x$  as

$$x = \Phi\alpha' = \Phi(\alpha + \alpha^\perp), \tag{4.69}$$

where  $\alpha'$  is a possible vector of expansion coefficients from  $\mathbb{C}^M$ ,  $\alpha$  is its unique projection onto  $S$ , and  $\alpha^\perp$  is an arbitrary vector in  $S^\perp$ . Within this infinite set of possible expansion coefficients, we can choose particular solutions by imposing further constraints on  $\alpha'$ . Typically, this is done by minimizing a particular norm, some of which we discuss now.

**Minimum  $\ell^2$ -Norm Solution** Among all expansion vectors  $\alpha'$  such that  $\Phi\alpha' = x$ , the solution with the smallest  $\ell^2$  norm is

$$\min \|\alpha'\|^2 = \|\alpha\|^2, \tag{4.70}$$

Frame	Constraints	Properties
General	$\{\varphi_i\}_{i=0}^{M-1}$ is a frame for $\mathbb{C}^N$	$\lambda_{\min}\ x\ ^2 \leq \sum_{i=0}^{M-1}  \langle x, \varphi_i \rangle ^2 \leq \lambda_{\max}\ x\ ^2$ $\lambda_{\min}I \leq T \leq \lambda_{\max}I$ $\text{tr}(T) = \sum_{j=0}^{N-1} \lambda_j = \text{tr}(G) = \sum_{i=0}^{M-1} \ \varphi_i\ ^2$
Equal-norm	$\ \varphi_i\  = \ \varphi_j\  = \varphi$ for all $i$ and $j$	$\lambda_{\min}\ x\ ^2 \leq \sum_{i=0}^{M-1}  \langle x, \varphi_i \rangle ^2 \leq \lambda_{\max}\ x\ ^2$ $\lambda_{\min}I \leq T \leq \lambda_{\max}I$ $\text{tr}(T) = \sum_{j=0}^{N-1} \lambda_j = \text{tr}(G) = \sum_{i=0}^{M-1} \ \varphi_i\ ^2 = M\varphi^2$
Tight	$\lambda_{\min} = \lambda_{\max} = \lambda$	$\sum_{i=0}^{M-1}  \langle x, \varphi_i \rangle ^2 = \lambda\ x\ ^2$ $T = \lambda I$ $\text{tr}(T) = \sum_{j=0}^{N-1} \lambda_j = N\lambda = \text{tr}(G) = \sum_{i=0}^{M-1} \ \varphi_i\ ^2$
Orthonormal basis	$\lambda_{\min} = \lambda_{\max} = 1$ $\ \varphi_i\  = 1$ for all $i$	$\sum_{i=0}^{M-1}  \langle x, \varphi_i \rangle ^2 = \ x\ ^2$ $T = I$ $\text{tr}(T) = \sum_{j=0}^{N-1} \lambda_j = N = \text{tr}(G) = \sum_{i=0}^{M-1} \ \varphi_i\ ^2 = M$ $N = M$

**Table 4.3:** Summary of properties for various classes of frames.

where  $\alpha = \tilde{\Phi}^*x$  is the expansion computed with respect to the canonical dual frame. The proof of this fact is along the lines of what we have seen in the introduction for the frame (4.1), since, from (4.69),

$$\|\alpha'\|^2 = \|\alpha\|^2 + \|\alpha^\perp\|^2 + 2\Re\langle \alpha, \alpha^\perp \rangle \stackrel{(a)}{=} \|\alpha\|^2 + \|\alpha^\perp\|^2,$$

where (a) follows from (4.56b). The minimum is achieved for  $\alpha^\perp = 0$ .

Since  $\Phi$  contains sets of  $N$  linearly independent vectors (often a very large number of such sets), we can write  $x$  as a linear combination of  $N$  vectors from one such set, that is,  $\alpha'$  will contain exactly  $N$  nonzero coefficients and will be sparse<sup>37</sup>. On the other hand, the minimum  $\ell^2$ -norm expansion coefficients  $\alpha$ , using the canonical dual, will typically contain  $M$  nonzero coefficients. We illustrate this in the following example:

**EXAMPLE 4.2 (NONUNIQUENESS OF THE DUAL FRAME)** Take  $\mathbb{R}^2$  and the unit-norm tight frame covering the unit circle at angles  $(2\pi i)/M$ , for  $i = 0, 1, \dots, M-1$ , an example of which we have already seen in (4.3) for  $M = 3$ . For  $M = 5$ , we get

$$\Phi = \begin{bmatrix} 1 & (-1 + \sqrt{5}) & -\frac{1}{4}(1 + \sqrt{5}) & -\frac{1}{4}(1 + \sqrt{5}) & \frac{1}{4}(-1 + \sqrt{5}) \\ 0 & \frac{1}{2\sqrt{2}}(\sqrt{5} + \sqrt{5}) & \frac{1}{2\sqrt{2}}(\sqrt{5} - \sqrt{5}) & -\frac{1}{2\sqrt{2}}(\sqrt{5} - \sqrt{5}) & -\frac{1}{2\sqrt{2}}(\sqrt{5} + \sqrt{5}) \end{bmatrix}.$$

<sup>37</sup>Remember that by *sparse* we mean an expansion that uses only  $N$  out of the  $M$  frame vectors.

The dual frame is just a scaled version of the frame itself,

$$\tilde{\Phi} = \frac{2}{5}\Phi.$$

For an arbitrary  $x$ ,  $\alpha = \tilde{\Phi}^*x$  will typically have 5 nonzero coefficients, but no fewer than 4 (when  $x$  is orthogonal to one of the  $\varphi$ 's). On the other hand, every set  $\{\varphi_i, \varphi_j\}$ ,  $i \neq j$ , is a biorthogonal basis for  $\mathbb{R}^2$ , meaning we can achieve an expansion with only 2 nonzero coefficients. Specifically, choose a biorthogonal basis  $\Psi = \{\varphi_i, \varphi_j\}$ , calculate its dual basis  $\tilde{\Psi} = \{\tilde{\varphi}_i, \tilde{\varphi}_j\}$ , and choose  $\alpha'$  as

$$\alpha'_k = \begin{cases} \langle x, \tilde{\varphi}_k \rangle, & k = i \text{ or } k = j; \\ 0, & \text{otherwise.} \end{cases}$$

We have  $\binom{5}{2} = 10$  possible bases; which ones are the best? As usual, those closer to an orthonormal basis will be better, because they are better conditioned. Thus, we should look for pairs  $\{\varphi_i, \varphi_j\}$  that have an inner product  $|\langle \varphi_i, \varphi_j \rangle|$  that is as small as possible. To do this, calculate the Gram operator (4.63) and take the absolute values of its entries  $|\langle \varphi_i, \varphi_j \rangle|$ :

$$\frac{1}{4} \begin{bmatrix} 4 & \sqrt{5}-1 & \sqrt{5}+1 & \sqrt{5}+1 & \sqrt{5}-1 \\ \sqrt{5}-1 & 4 & \sqrt{5}-1 & \sqrt{5}+1 & \sqrt{5}+1 \\ \sqrt{5}+1 & \sqrt{5}-1 & 4 & \sqrt{5}-1 & \sqrt{5}+1 \\ \sqrt{5}+1 & \sqrt{5}+1 & \sqrt{5}-1 & 4 & \sqrt{5}-1 \\ \sqrt{5}-1 & \sqrt{5}+1 & \sqrt{5}+1 & \sqrt{5}-1 & 4 \end{bmatrix},$$

and we see, as it is obvious from the geometry of the problem, that 5 bases,  $\{\{\varphi_0, \varphi_1\}, \{\varphi_0, \varphi_4\}, \{\varphi_1, \varphi_3\}, \{\varphi_2, \varphi_4\}, \{\varphi_3, \varphi_4\}\}$ , have a minimum inner product, those with  $|\langle \varphi_i, \varphi_j \rangle| = (\sqrt{5}-1)/4 \sim 0.31$ . Now which of these to choose? If we do not take into account  $x$ , it really does not matter. However, if we do take it into account, then it makes sense to first choose a vector  $\varphi_i$  that is most aligned with  $x$ :

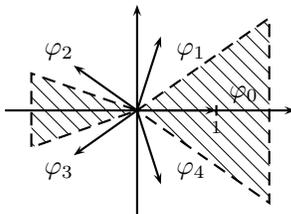
$$\max_i |\langle x, \varphi_i \rangle|.$$

Assume  $\varphi_0$  is chosen, that is,  $x$  is in the shaded region in Figure 4.8. Then, either  $\varphi_1$  or  $\varphi_4$  can be used. Let us choose an  $x$  in the shaded region, say  $x = [\sqrt{3}/2 \ 1/2]^T$ , and compute both  $\alpha = \tilde{\Phi}^*x$ , as well as  $\alpha' = \tilde{\Psi}^*x$  with the biorthogonal basis  $\Psi = \{\varphi_0, \varphi_1\}$ . Then,

$$\begin{aligned} \alpha &= [0.34641 \ 0.297258 \ -0.162695 \ -0.397809 \ -0.0831647]^T, \\ \alpha' &= [0.703566 \ 0.525731 \ 0 \ 0 \ 0]^T. \end{aligned}$$

As expected,  $\alpha$  has 5 nonzero coefficients, while  $\alpha'$  has only 2. Then,

$$\|\alpha\|_2 = 0.63246 < 0.87829 = \|\alpha'\|_2, \quad (4.71)$$



**Figure 4.8:** Unit-norm tight frame in  $\mathbb{R}^2$ . Those  $x$  belonging to the shaded region have the maximum inner product (in magnitude) with  $\varphi_0$ . One can then choose  $\varphi_1$  or  $\varphi_4$  as the other vector in a biorthogonal expansion.

as expected, as  $\alpha$  has the minimum  $\ell^2$  norm. However,

$$\|\alpha\|_1 = 1.28734 > 1.22930 = \|\alpha'\|_1, \quad (4.72)$$

and thus, the sparser expansion is worse with respect to the  $\ell^2$  norm, but is better with respect to the  $\ell^1$  norm, illustrating the wide range of possibilities for expansions in frames, as well as algorithmic issues that will be explored later.

**Minimum  $\ell^1$ -Norm Solution** Instead of the  $\ell^2$  norm, we can minimize the  $\ell^1$  norm. That is, solve

$$\min \|\alpha'\|_1 \quad \text{under the constraint} \quad \Phi\alpha' = x.$$

This can be turned into a linear program (see Section 4.6.3). Interestingly, minimizing the  $\ell^1$  norm will promote sparsity.

**EXAMPLE 4.3 (NONUNIQUENESS OF THE DUAL FRAME (CONT'D))** We now continue our previous example and calculate the expansion coefficients for the 5 biorthogonal bases  $\Psi_{01} = \{\varphi_0, \varphi_1\}$ ,  $\Psi_{04} = \{\varphi_0, \varphi_4\}$ ,  $\Psi_{13} = \{\varphi_1, \varphi_3\}$ ,  $\Psi_{24} = \{\varphi_2, \varphi_4\}$ ,  $\Psi_{34} = \{\varphi_3, \varphi_4\}$ . These, and their  $\ell^1$  norms are (we have already computed  $\alpha'_{01} = \alpha'$  above but repeat it here for completeness):

$\alpha'$	$\ \alpha'\ _1$		
$\alpha'_{01}$	0.703566	0.525731	1.22930
$\alpha'_{04}$	1.028490	-0.525731	1.55422
$\alpha'_{13}$	-0.177834	-1.138390	1.31623
$\alpha'_{24}$	-1.664120	-1.554221	3.21834
$\alpha'_{34}$	-1.028490	0.109908	1.13839

So we see that even among sparse expansions with exactly 2 nonzero coefficients there are differences. In this particular case,  $\Psi_{34}$  has the lowest  $\ell^1$  norm.

**Minimum  $\ell^0$ -Norm Solution** The  $\ell^0$  norm simply counts the number of nonzero entries in a vector:

$$\|x\|_0 = \lim_{p \rightarrow 0} \sum_{k \in \mathbb{Z}} |x_k|^p, \quad (4.73)$$

with  $0^0 = 0$ . Since a frame with  $M$  vectors in an  $N$ -dimensional space has necessarily a set of  $N$  linearly independent vectors, we can take these as a basis, compute the biorthogonal dual basis, and find an expansion  $\alpha'$  with exactly  $N$  nonzero components (as we have just done in Example 4.2). Usually, there are many such sets (see Exercise ??), all leading to an expansion with  $N$  nonzero coefficients. Among these multiple solutions, we may want to choose that one with the least  $\ell^2$  norm. This shows that there exists a sparse expansion, very different from the expansion that minimizes the  $\ell^2$  norm (which will typically uses all  $M$  frame vectors and is thus not sparse).

**Minimum  $\ell^\infty$ -Norm Solution** Among possible expansion coefficients  $\alpha'$ , we can also choose that one that minimizes the maximum value  $|\alpha'_i|$ . That is, solve

$$\min \|\alpha'\|_\infty \quad \text{under the constraint} \quad \Phi\alpha' = x.$$

This optimization problem can be solved using TBD. While such a solution is useful when one wants to avoid large coefficients, minimizing the  $\ell^2$  norm achieves a similar goal.

**Choosing the Expansion Coefficients** In summary, we have seen that the nonuniqueness of possible frame expansion coefficients leaves us with freedom to optimize some other criteria. For example, for a sparse expansion using only a few vectors from the frame, minimizing the  $\ell^0$  norm is a possible route, although computationally difficult. Instead, minimizing the  $\ell^1$  norm achieves a similar goal (as we will see in Chapter 7), and can be done with an efficient algorithm—namely, linear programming (see Section 4.6.3). Minimizing the  $\ell^2$  norm does not lead to sparsity; instead, it promotes small coefficients, similarly to minimizing the maximum absolute value of coefficients, or the  $\ell^\infty$  norm. We illustrate this discussion with a simple example:

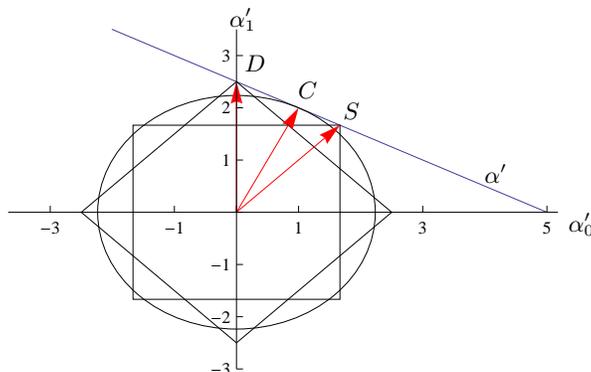
**EXAMPLE 4.4 (DIFFERENT NORMS LEAD TO DIFFERENT EXPANSIONS)** Consider the simplest example,  $N = 1$ ,  $M = 2$ . As a frame and its dual, choose

$$\Phi = \frac{1}{5} [1 \quad 2] \quad \tilde{\Phi} = [1 \quad 2] \quad \Phi\tilde{\Phi}^* = I.$$

Given an input  $x$ , the subspace of all expansion coefficients  $\alpha'$  that leads to  $x = \Phi\alpha'$  is described by

$$\alpha' = \alpha + \alpha^\perp = \begin{bmatrix} 1 \\ 2 \end{bmatrix} x + \begin{bmatrix} 2 \\ -1 \end{bmatrix} \gamma,$$

since the first term is colinear with  $\Phi$ , while the second is orthogonal to  $\Phi$ . In Figure 4.9 we show  $\alpha'$  for  $x = 1$ . It is a line of slope  $-1/2$  passing through the point  $[1 \quad 2]$ ,  $\alpha'_1 = -(1/2)\alpha'_0 + 5/2$ . We can choose any point on this line as a possible set  $[\alpha'_0 \quad \alpha'_1]^T$  for reconstructing  $x$  with the frame  $\Phi$ . Recalling Figure 2.7 depicting points with constant  $\ell^1$ -,  $\ell^2$ -, and  $\ell^\infty$  norms, we now see what the solutions are to the minimization problem in different norms:



**Figure 4.9:** The space of possible expansion coefficients in the frame  $\Phi = (1/5)[1 \ 2]$ , and the subspace  $\alpha' = [1 \ 2]^T x + [2 \ -1]^T \gamma$  for  $x = 1$ . To find the points of minimum  $\ell^1$ -,  $\ell^2$ -, and  $\ell^\infty$  norms, we grow a diamond, a circle and a square, respectively, and find the intercept points with the subspace  $\alpha'$  (see also *Figure 2.7* showing points with constant  $\ell^1$ -,  $\ell^2$ -, and  $\ell^\infty$  norms). These are  $D = [0 \ 5/2]$ ,  $C = [1 \ 2]$  and  $S = [5/3 \ 5/3]$ , respectively.

- (i) *Minimum  $\ell^2$ -norm solution:* The points with the same  $\ell^2$  norm form a circle. Thus, growing a circle from the origin to the intercept with  $\alpha'$  yields the point  $C = [1 \ 2]$  with the minimum  $\ell^2$  norm (see *Figure 4.9*). From what we know about the  $\ell^2$  norm, we could have also obtained it as the point on  $\alpha'$  closest to the origin (orthogonal projection of the origin onto the line of possible  $\alpha'$ ).
- (ii) *Minimum  $\ell^1$ -norm solution:* The points with the same  $\ell^1$  norm form a diamond. Thus, growing a diamond from the origin to the intercept with  $\alpha'$  yields the point  $D = [0 \ 5/2]$  with the minimum  $\ell^1$  norm (see *Figure 4.9*).
- (iii) *Minimum  $\ell^\infty$ -norm solution:* The points with the same  $\ell^\infty$  norm form a square. Thus, growing a square from the origin to the intercept with  $\alpha'$  yields the point  $S = [5/3 \ 5/3]$  with the minimum  $\ell^\infty$  norm (see *Figure 4.9*).

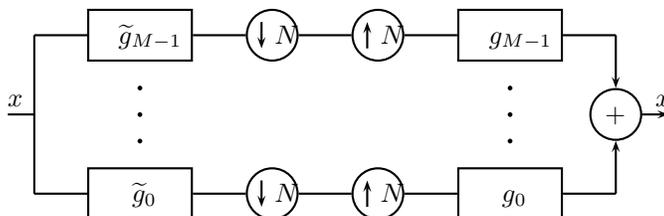
The table below numerically compares these three cases:

	$\ell^1$	$\ell^2$	$\ell^\infty$
$D$	<i>2.50</i>	2.50	2.50
$C$	3.00	<i>2.24</i>	2.00
$S$	3.33	2.36	<i>1.67</i>

Emphasized entries are the minimum values for each respective norm.

### 4.3 Oversampled Filter Banks

This section develops necessary conditions for the design of oversampled filter banks implementing tight frame expansions. We consider mostly those filter banks implementing tight frames, as the general ones follow easily and can be found in the literature.



**Figure 4.10:** A filter-bank implementation of a frame expansion: It is an  $M$ -channel filter bank with sampling by  $N$ ,  $M > N$ .

As we have done for filter banks implementing basis expansions (Chapters 1-3) we also look into their polyphase representation.

From everything we have learned so far, we may expect to have an  $M$ -channel filter bank, where each channel corresponds to one of the template frame vectors (a couple of simple examples were given in Section 4.1 and illustrated in Figures 4.2 and 4.4). The infinite set of frame vectors is obtained by shifting the  $M$  template ones by integer multiples of  $N$ ,  $N < M$ ; thus the redundancy of the system. This shifting can be modeled by the samplers in the system, as we have seen previously. Not surprisingly thus, a general oversampled filter bank implementing a frame expansion is given in Figure 4.10. We now go through the salient features in some detail; however, since this material is a simple extension of what we have seen previously for bases, we will be brief.

### 4.3.1 Tight Oversampled Filter Banks

We now follow the structure of the previous section and show the filter-bank equivalent of the expansion, expansion coefficients, geometry of the expansion, as well as look into the polyphase decomposition as a standard analysis tool, as we have done in the previous chapters.

As opposed to the previous section, we now work in an infinite-dimensional space,  $\ell^2(\mathbb{Z})$ , where formally, many things will look the same. However, we need to exercise care, and will point out specific instances when this is the case. Instead of a finite-dimensional matrix  $\Phi$  as in (4.30), we now deal with an infinite-dimensional one, and with structure: the  $M$  template frame vectors,  $\varphi_0, \varphi_1, \dots, \varphi_{M-1}$ , repeat themselves shifted in time, much the same way they do for bases. Renaming them  $g_0 = \varphi_0, g_1 = \varphi_1, \dots, g_{M-1} = \varphi_{M-1}$ , we get

$$\Phi = \left[ \cdots \boxed{g_{0,n}} \quad g_{1,n} \quad \cdots \quad g_{M-1,n} \quad g_{0,n-N} \quad g_{1,n-N} \quad \cdots \quad g_{M-1,n-N} \quad \cdots \right],$$

just like for critically-sampled filter banks (those with the number of channel samples per unit of time conserved, that is,  $M = N$ , or, those implementing basis expansions), except for the larger number of template frame vectors. We could easily implement finite-dimensional frame expansions we have seen in the last section

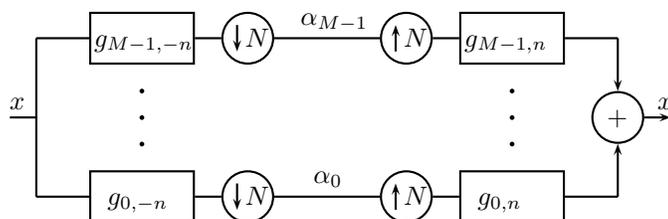


Figure 4.11: A filter-bank implementation of a tight frame expansion.

by just limiting the number of nonzero coefficients in  $g_i$  to  $N$ , resulting in

$$\Phi = \begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ \dots & \boxed{g_{0,0}} & \dots & g_{M-1,0} & 0 & \dots & 0 & \dots \\ \dots & g_{0,1} & \dots & g_{M-1,1} & 0 & \dots & 0 & \dots \\ \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots \\ \dots & g_{0,N-1} & \dots & g_{M-1,N-1} & 0 & \dots & 0 & \dots \\ \dots & 0 & \dots & 0 & g_{0,0} & \dots & g_{M-1,0} & \dots \\ \dots & 0 & \dots & 0 & g_{0,1} & \dots & g_{M-1,1} & \dots \\ \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots \\ \dots & 0 & \dots & 0 & g_{0,N-1} & \dots & g_{M-1,N-1} & \dots \\ \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

$$= \begin{bmatrix} \ddots & \vdots & \vdots & \ddots \\ \dots & \Phi_0 & \dots & \dots \\ \dots & & \Phi_0 & \dots \\ \ddots & \vdots & \vdots & \ddots \end{bmatrix},$$

that is, a block-diagonal matrix, with the finite-dimensional frame matrix  $\Phi_0$  of size  $N \times M$  on the diagonal. Recall that we concentrate on the tight-frame case, and therefore,  $\Phi_0 \Phi_0^* = I$ .

**Expansion** We can express the frame expansion formally in the same way as we did for finite-dimensional frames in (4.32) (again, because it is the tight-frame case)

$$\Phi \Phi^* = I, \tag{4.74}$$

except that we will always work with 1-tight frames by normalizing  $\Phi$  if necessary by  $1/\sqrt{\lambda}$ , for the filter bank to be perfect reconstruction. Writing out the expansion, however, we see its infinite-dimensional aspect:

$$x = \sum_{i=0}^{M-1} \sum_{k \in \mathbb{Z}} \langle x, g_{i,n-Nk} \rangle g_{i,n-Nk}. \tag{4.75}$$

The process of computing the expansion coefficients is implemented via an analysis filter bank, filtering by individual filters  $g_{i,-n}$ ,  $i = 0, 1, \dots, M - 1$ , and downsampling by  $N$ , as on the left side of Figure 4.11:

$$\alpha = \Phi^* x \quad \alpha_{i,k} = \langle x, g_{i,n-Nk} \rangle, \quad (4.76)$$

while the process of reconstructing  $x$  is implemented via a synthesis filter bank, upsampling by  $N$  and filtering by individual filters  $g_{i,n}$ ,  $i = 0, 1, \dots, M - 1$ , as on the right side of Figure 4.11:

$$x = \Phi \alpha \quad x = \sum_{i=0}^{M-1} \sum_{k \in \mathbb{Z}} \alpha_{i,k} g_{i,n-Nk}. \quad (4.77)$$

In all of the above,  $\Phi$  is an infinite matrix,  $\alpha$  and  $x$  are infinite vectors.

One can, of course, use the Fourier-domain or  $z$ -transform-domain expressions, as before. Since they are identical (except for the number of filters), we just give one as an example. For example, in  $z$ -transform-domain, we can find the expression of the effect of one single branch as

$$G_i(z) \frac{1}{N} \sum_{k=0}^{N-1} G_i(W_N^{-k} z^{-1}) X(W_N^k z).$$

Summing these over all branches,  $i = 0, 1, \dots, M - 1$ , we get

$$\begin{aligned} X(z) &= \sum_{i=0}^{M-1} G_i(z) \frac{1}{N} \sum_{k=0}^{N-1} G_i(W_N^{-k} z^{-1}) X(W_N^k z) \\ &= \frac{1}{N} \sum_{k=0}^{N-1} \left( \sum_{i=0}^{M-1} G_i(z) G_i(W_N^{-k} z^{-1}) \right) X(W_N^k z). \end{aligned}$$

Therefore, for perfect reconstruction, the term with  $X(z)$  must equal  $N$ , while all the others (aliasing terms) must cancel, that is:

$$\begin{aligned} \sum_{i=0}^{M-1} G_i(z) G_i(z^{-1}) &= N, \\ \sum_{i=0}^{M-1} G_i(z) G_i(W_N^{-k} z^{-1}) &= 0, \quad k = 1, 2, \dots, M - 1. \end{aligned}$$

For example, for  $N = 2$  and  $M = 3$ , we get that:

$$\begin{aligned} G_0(z) G_0(z^{-1}) + G_1(z) G_1(z^{-1}) + G_2(z) G_2(z^{-1}) &= 2, \\ G_0(z) G_0(-z^{-1}) + G_1(z) G_1(-z^{-1}) + G_2(z) G_2(-z^{-1}) &= 0. \end{aligned}$$

Compare this to its counterpart expression in two-channel filter banks in (1.28).

**Geometry of the Expansion** Analogously to bases, each branch (channel) projects onto a subspace of  $\ell^2(\mathbb{Z})$  we call  $V_0$  or  $W_i$ ,  $i = 1, 2, \dots, M - 1$ .<sup>38</sup> While each of these is on its own an orthogonal projection (because  $P_V$  in (1.18) is an orthogonal projection operator), they are not orthogonal to each other because of oversampling. Each of the orthogonal projection operators is given as

$$\begin{aligned} P_{V_0} &= G_0 U_N D_N G_0^T, \\ P_{W_i} &= G_i U_N D_N G_i^T, \quad i = 1, 2, \dots, M - 1, \end{aligned}$$

with the range

$$\begin{aligned} V_0 &= \text{span}(\{g_{0,n-Nk}\}_{k \in \mathbb{Z}}), \\ W_i &= \text{span}(\{g_{i,n-Nk}\}_{k \in \mathbb{Z}}), \quad i = 1, 2, \dots, M - 1. \end{aligned}$$

### 4.3.2 Polyphase View of Oversampled Filter Banks

To cover the polyphase view for general  $N$  and  $M$ , we cover it through an example with  $N = 2$ ,  $M = 3$ ; expressions for general  $N$  and  $M$  follow easily.

**EXAMPLE 4.5 (TIGHT OVERSAMPLED 3-CHANNEL FILTER BANKS)** For two-channel filter banks, a polyphase decomposition is achieved by simply splitting both sequences and filters into their even- and odd-indexed subsequences; this is governed by the sampling factor. In an oversampled tight filter bank with  $N = 2$  and  $M = 3$ , we still do the same; the difference is going to be in the number of filters, as before. We have already seen how to decompose an input sequence in (3.216), synthesis filters in (1.32), and analysis filters in (1.34). In our context, these polyphase decompositions are the same, except that for filters, we have more of them involved:

$$\begin{aligned} g_{i,0,n} &= g_{i,2n} & \xleftrightarrow{\text{ZT}} & G_{i,0}(z) = \sum_{n \in \mathbb{Z}} g_{i,2n} z^{-n}, \\ g_{i,1,n} &= g_{i,2n+1} & \xleftrightarrow{\text{ZT}} & G_{i,1}(z) = \sum_{n \in \mathbb{Z}} g_{i,2n+1} z^{-n}, \\ & & & G_i(z) = G_{i,0}(z^2) + z^{-1} G_{i,1}(z^2), \end{aligned}$$

for  $i = 0, 1, 2$  and synthesis filters. That is, we have 3 filters with 2 polyphase components each, leading to the following synthesis *polyphase matrix*  $\Phi_p(z)$ :

$$\Phi_p(z) = \begin{bmatrix} G_{0,0}(z) & G_{1,0}(z) & G_{2,0}(z) \\ G_{0,1}(z) & G_{1,1}(z) & G_{2,1}(z) \end{bmatrix}.$$

As expected, the polyphase matrix is no longer square; rather, it is a  $(2 \times 3)$  matrix of polynomials. Similarly, on the analysis side, since this is a filter bank implementing a tight frame with  $\tilde{\Phi} = \Phi$ , we assume the same filters as on the

<sup>38</sup>We assume here that the space  $V_0$  is lowpass in nature, while the  $W_i$  are bandpass.

synthesis side, only time reversed,

$$\begin{aligned} \tilde{g}_{i,0,n} = \tilde{g}_{i,2n} = g_{i,-2n} &\xleftrightarrow{ZT} \tilde{G}_{i,0}(z) = \sum_{n \in \mathbb{Z}} g_{i,-2n} z^{-n}, \\ \tilde{g}_{i,1,n} = \tilde{g}_{i,2n-1} = g_{i,-2n+1} &\xleftrightarrow{ZT} \tilde{G}_{i,1}(z) = \sum_{n \in \mathbb{Z}} g_{i,-2n+1} z^{-n}, \\ &\tilde{G}_i(z) = G_{i,0}(z^{-2}) + zG_{i,1}(z^{-2}), \end{aligned}$$

for  $i = 0, 1, 2$ . With this definition, the analysis polyphase matrix is, similarly to the one for the two-channel case:

$$\tilde{\Phi}_p(z) = \begin{bmatrix} G_{0,0}(z^{-1}) & G_{1,0}(z^{-1}) & G_{2,0}(z^{-1}) \\ G_{0,1}(z^{-1}) & G_{1,1}(z^{-1}) & G_{2,1}(z^{-1}) \end{bmatrix} = \Phi_p(z^{-1}),$$

where  $\tilde{\Phi}_p(z)$  is again a  $(2 \times 3)$  matrix of polynomials.

As before, this type of a representation allows for a very compact input-output relationship between the input (decomposed into polyphase components) and the result coming out of the synthesis filter bank:

$$X(z) = \begin{bmatrix} 1 & z^{-1} \end{bmatrix} \Phi_p(z^2) \Phi_p^*(z^{-2}) \begin{bmatrix} X_0(z^2) \\ X_1(z^2) \end{bmatrix},$$

where we have again used Hermitian transpose because we will often deal with complex-coefficient filter banks in this chapter. The above is formally the same as the expression for a critically-sampled filter bank with 2 channels; the oversampling is hidden in the dimensions of the rectangular matrices  $\Phi_p$  and  $\tilde{\Phi}_p$ . Clearly for the above to hold,  $\Phi_p(z^2) \Phi_p^*(z^{-2})$  must be an identity, analogously to orthogonal filter banks. This result for tight frames is formalized in Theorem 4.8.

The above example went through various polyphase concepts for a tight oversampled 3-channel filter bank. For general oversampled filter banks with  $N, M$ , expressions are the same as those given in (2.12c), (2.12e), except with  $M$  filters instead of  $N$ . The corresponding polyphase matrices are of sizes  $N \times M$  each.

**Frame Operators** All the frame operators we have seen so far can be expressed via filter bank ones as well.

The frame operator  $T$  for a general infinite-dimensional frame is formally defined as for the finite-dimensional one in (4.57), except that it is now infinite-dimensional itself. Its polyphase counterpart is:

$$T_p(z) = \Phi_p(z) \Phi_p^*(z^{-1}). \quad (4.80)$$

For a tight frame implemented by a tight oversampled filter bank, this has to be an identity as we have already said in the above example. In other words,  $\Phi_p$  is a rectangular paraunitary matrix. The frame operator  $T_p(z)$  is positive definite on the unit circle:

$$T_p(e^{j\omega}) = |\Phi_p(e^{j\omega})|^2 > 0. \quad (4.81)$$

The canonical dual frame operator has its polyphase counterpart in:

$$\tilde{\Phi}_p(z) = T_p(z)^{-1} \Phi_p(z). \quad (4.82)$$

Again, we can see that when the frame is tight,  $T_p(z) = I$ , then the dual polyphase matrix is the same as  $\Phi_p(z)$ .

**Polyphase Decomposition of an Oversampled Filter Bank** As before, the polyphase formulation allows us to characterize classes of solutions. The following theorem, the counterpart of Theorem 2.1 for critically-sampled filter banks, summarizes these without proof, the pointers to which are given in *Further Reading*.

**THEOREM 4.8 (OVERSAMPLED  $M$ -CHANNEL FILTER BANKS IN POLYPHASE DOMAIN)** Given is an  $M$ -channel filter bank with sampling by  $N$  and the polyphase matrices  $\Phi_p(z)$ ,  $\tilde{\Phi}_p(z)$ . Then:

- (i) *Frame expansion in polyphase domain*

A filter bank implements a general frame expansion if and only if

$$\Phi_p(z) \tilde{\Phi}_p^*(z) = I. \quad (4.83a)$$

A filter bank implements a tight frame expansion if and only if

$$T_p(z) = \Phi_p(z) \Phi_p^*(z^{-1}) = I, \quad (4.83b)$$

that is,  $\Phi_p(z)$  is paraunitary.

- (ii) *Naimark's theorem in polyphase domain*

An infinite-dimensional frame implementable via an  $M$ -channel filter bank with sampling by  $N$  is a general frame if and only if there exists a biorthogonal basis implementable via an  $M$ -channel filter bank with sampling by  $M$  so that

$$\Phi_p^*(z) = \Psi_p(z)[J], \quad (4.84)$$

where  $J \subset \{0, \dots, M-1\}$  is the index set of the retained columns of  $\Psi_p(z)$ , and  $\Phi_p(z)$ ,  $\Psi_p(z)$  are the frame/basis polyphase matrices, respectively.

An infinite-dimensional frame implementable via an  $M$ -channel filter bank with sampling by  $N$  is a tight frame if and only if there exists an orthonormal basis implementable via an  $M$ -channel filter bank with sampling by  $M$  so that (4.84) holds.<sup>39</sup>

- (iii) *Frame bounds*

The frame bounds of a frame implementable by a filter bank are given by:

$$\lambda_{\min} = \min_{\omega \in [-\pi, \pi)} T_p(e^{j\omega}), \quad (4.85a)$$

$$\lambda_{\max} = \max_{\omega \in [-\pi, \pi)} T_p(e^{j\omega}). \quad (4.85b)$$

The last statement on eigenvalues stems from the fact that the frame operator  $T$  and its polyphase counterpart  $T_p(e^{j\omega})$  are related via a unitary transformation. If the eigenvalues of  $T_p(e^{j\omega})$  are defined via  $T_p(e^{j\omega})v(\omega) = \lambda(\omega)v(\omega)$ , then the eigenvalues of  $T$  and  $T_p(e^{j\omega})$  are the same, leading to (4.85).

EXAMPLE 4.6 (TIGHT OVERSAMPLED 3-CHANNEL FILTER BANKS CONT'D) We now set  $M = 3$ ,  $N = 2$  and show how one can obtain a linear-phase tight frame with filters of length greater than 2, a solution not possible for critically-sampled filter banks with sampling by 2, as was shown in Theorem 1.12. We know that such a filter bank implementing a tight frame transform must be seeded from an orthogonal filter bank with a  $3 \times 3$  paraunitary matrix.

We use such a matrix in the example showing how to parameterize  $N$ -channel orthogonal filter banks, Example 2.2 with  $K = 2$ , that is, all polyphase components will be first-degree polynomials in  $z^{-1}$ . We form a tight frame by deleting its last column and call the resulting frame polyphase matrix  $\Phi_p^T(z)$ . Since there are 5 angles involved, the matrix is too big to explicitly state here; instead, we start imposing the linear-phase conditions to reduce the number of degrees of freedom. A simple solution with  $\theta_{00} = \pi/2$ ,  $\theta_{11} = \pi/2$ ,  $\theta_{02} = \pi/4$  and  $\theta_{10} = 3\pi/4$ , leads to the first two filters being symmetric of length 3 and the last antisymmetric of length 3. The resulting polyphase matrix is (where we have rescaled the first and third columns by  $-1$ ):

$$\Phi_p(z) = \begin{bmatrix} \frac{1}{2} \cos \theta_{01}(1 + z^{-1}) & \frac{1}{2} \sin \theta_{01}(1 + z^{-1}) & \frac{1}{2}(1 - z^{-1}) \\ \sin \theta_{01} & -\cos \theta_{01} & 0 \end{bmatrix}^T,$$

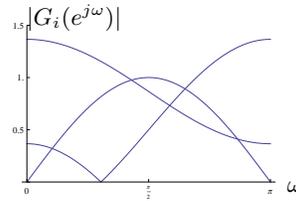
leading to the following three filters:

$$\begin{aligned} G_0(z) &= \frac{1}{2} \cos \theta_{01} + \sin \theta_{01} z^{-1} + \frac{1}{2} \cos \theta_{01} z^{-2}, \\ G_1(z) &= \frac{1}{2} \sin \theta_{01} - \cos \theta_{01} z^{-1} + \frac{1}{2} \sin \theta_{01} z^{-2}, \\ G_2(z) &= \frac{1}{2} - \frac{1}{2} z^{-2}. \end{aligned}$$

For example, with  $\theta_{01} = \pi/3$ , the three resulting filters have reasonable coverage of the frequency axis (see Figure 4.12).

## 4.4 Local Fourier Frames

Until now, the material in this chapter covered finite-dimensional frames (Section 4.2) and oversampled filter banks as a vehicle for implementing both finite-dimensional as well as certain infinite-dimensional frames (previous section). We now investigate a more specific class of frames; those obtained by modulating (shifting in frequency) a single prototype filter/frame vector, introduced in their basis form in Chapter 2. These are some of the oldest bases and frames, and some of the most widely used. The local Fourier expansions arose in response to the need to



**Figure 4.12:** Tight oversampled 3-channel filter bank with sampling by  $N = 2$  and linear-phase filters. The figure depicts the three magnitude responses.

create a *local* Fourier tool, able to achieve some localization in time, at the price of worsening the known excellent localization in frequency.

As in Chapter 2, we will consider two large classes of local Fourier frames, those obtained by complex-exponential modulation, as well as those obtained by cosine modulation of a single prototype filter/frame vector. In Chapter 2, we learned that, while there exist no good local Fourier bases (apart from those equivalent to a finite-dimensional basis), there do exist good local cosine bases. In this section, we go even farther; we show that there exist good local Fourier frames, due to the extra freedom redundancy buys us.

#### 4.4.1 Complex Exponential-Modulated Local Fourier Frames

Complex-exponential modulation is used in many instances, such as the DFT basis, (2.2), (2.5), as well as the basis constructed from the ideal filters (2.6), and is at the heart of the local Fourier expansion known as Gabor transform. The term *Gabor frame* is often used to describe any frame with complex-exponential modulation and overlapping frame vectors (oversampled filter banks with filters of lengths longer than the sampling factor  $N$ ). For complex exponential-modulated bases, we defined this modulation in (2.16); for complex exponential-modulated frames, we do it now.

**Complex-Exponential Modulation** Given a prototype filter  $p = g_0$ , the rest of the filters are obtained via complex-exponential modulation:

$$\begin{aligned} g_{i,n} &= p_n e^{j(2\pi/N)in} = p_n W_M^{-in}, \\ G_i(z) &= P(W_M^i z), \\ G_i(e^{j\omega}) &= P(e^{j(\omega - (2\pi/M)i)}) = P(W_M^i e^{j\omega}), \end{aligned} \quad (4.86)$$

for  $i = 1, 2, \dots, M-1$ . A filter bank implementing such a frame expansion is often called *complex exponential-modulated oversampled filter bank*. While the prototype filter  $p = g_0$  is typically real, the rest of the bandpass filters are complex. The above is identical to the expression for bases, (2.16); the difference is in the sampling factor  $N$ , smaller here than the number of filters  $M$ .

**Overcoming the Limitations of the Balian-Low Theorem** In Chapter 2, Theorem 2.2, we saw that there does not exist a complex exponential-modulated local

Fourier basis implementable by an  $N$ -channel FIR filter bank, except for a filter bank with filters of length  $N$ . We illustrated the proof with an example for  $N = 3$  in (2.17) and demonstrated that the only solution consisted of each polyphase component being a monomial, leading to a block-based expansion.

We now investigate what happens with frames. Start with the polyphase representation (2.12d) of the prototype filter  $p = g_0$ ,

$$P(z) = P_0(z^N) + z^{-1}P_1(z^N) + \dots + z^{-(N-1)}P_{N-1}(z^N),$$

where  $P_i(z)$ ,  $i = 0, 1, \dots, N - 1$  are its polyphase components. The modulated versions become

$$\begin{aligned} G_i(z) &= P(W_M^i z) \\ &= P_0(W_M^{iN} z^N) + \dots + W_M^{-(N-1)i} z^{-(N-1)} P_{N-1}(W_M^{iN} z^N), \end{aligned}$$

for  $i = 1, 2, \dots, M - 1$ . On a simple example, we now show that relaxing the basis requirement allows us to implement a tight frame expansion via an oversampled filter bank with FIR filters longer than the sampling factor  $N$ .

**EXAMPLE 4.7 (OVERCOMING LIMITATIONS OF BALIAN-LOW THEOREM)** Let  $N = 2$  and  $M = 3$ . The polyphase matrix corresponding to the complex exponential-modulated filter bank is given by

$$\begin{aligned} \Phi_p(z) &= \begin{bmatrix} P_0(z) & P_0(W_3^2 z) & P_0(W_3 z) \\ P_1(z) & W_3^2 P_1(W_3^2 z) & W_3 P_1(W_3 z) \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} P_u(z) & & & & & \\ & P_\ell(z) & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \end{bmatrix}, \quad (4.87) \end{aligned}$$

with  $P_u(z)$  and  $P_\ell(z)$  the diagonal matrices of polyphase components:

$$\begin{aligned} P_u(z) &= \text{diag}([P_0(z), P_0(W_3 z), P_0(W_3^2 z)]), \\ P_\ell(z) &= \text{diag}([P_1(z), P_1(W_3 z), P_1(W_3^2 z)]), \end{aligned}$$

and  $W_3^{-1} = W_3^2$ ,  $W_3^{-2} = W_3$ . Compare (4.87) to its basis counterpart in (2.17).

We now want to see whether it is possible for such a frame polyphase matrix

to implement a tight frame, in which case, it would have to satisfy (4.83b).

$$\begin{aligned}
& \Phi_p(z)\Phi_p^*(z^{-1}) = \\
& = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} P_u(z) & \\ & P_\ell(z) \end{bmatrix} \begin{bmatrix} I & W^{-1} \\ W & I \end{bmatrix} \begin{bmatrix} P_u(z^{-1}) & \\ & P_\ell(z^{-1}) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \\
& = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} P_u(z)P_u(z^{-1}) & W^{-1}P_u(z)P_\ell(z^{-1}) \\ W P_\ell(z)P_u(z^{-1}) & P_\ell(z)P_\ell(z^{-1}) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \\
& \stackrel{(a)}{=} \begin{bmatrix} \sum_{i=0}^2 P_0(W_3^{-i}z)P_0(W_3^i z^{-1}) & \sum_{i=0}^2 W_3^i P_0(W_3^{-i}z)P_1(W_3^i z^{-1}) \\ \sum_{i=0}^2 W_3^i P_0(W_3^{-i}z^{-1})P_1(W_3^i z) & \sum_{i=0}^2 P_1(W_3^{-i}z)P_1(W_3^i z^{-1}) \end{bmatrix} \\
& \stackrel{(a)}{=} \begin{bmatrix} \sum_{i=0}^2 P_0(W_3^i z)P_0(W_3^{-i} z^{-1}) & \sum_{i=0}^2 W_3^i P_0(W_3^i z)P_1(W_3^{-i} z^{-1}) \\ \sum_{i=0}^2 W_3^{-i} P_0(W_3^{-i} z^{-1})P_1(W_3^i z) & \sum_{i=0}^2 P_1(W_3^i z)P_1(W_3^{-i} z^{-1}) \end{bmatrix} = I,
\end{aligned}$$

where (a) follows again from  $W_3^{-1} = W_3^2$ ,  $W_3^{-2} = W_3$ ,  $W = \text{diag}([1, W_3, W_3^2])$ , and we assumed that  $p$  is real. It is clear that the set of conditions above is much less restrictive than that of every polyphase component of the prototype filter having to be a monomial (the condition that lead to the negative result in the discrete Balian-Low theorem, Theorem 2.2).

For example, we see that the conditions on each polyphase component:

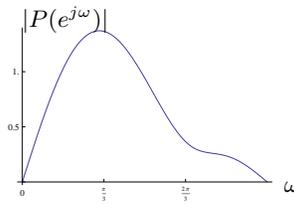
$$\begin{aligned}
& \sum_{i=0}^2 P_0(W_3^i z)P_0(W_3^{-i} z^{-1}) = 1, \\
& \sum_{i=0}^2 P_1(W_3^i z)P_1(W_3^{-i} z^{-1}) = 1,
\end{aligned}$$

are equivalent to those polyphase components being orthogonal filters as in (2.7). On the other hand, the conditions involving both polyphase components:

$$\begin{aligned}
& \sum_{i=0}^2 W_3^i P_0(W_3^i z)P_1(W_3^{-i} z^{-1}) = 0, \\
& \sum_{i=0}^2 W_3^{-i} P_0(W_3^{-i} z^{-1})P_1(W_3^i z) = 0,
\end{aligned}$$

are equivalent to  $P_0(z)$  and  $z^{-1}P_1(z)$  being orthogonal to each other as in (2.9).

For example, we know that the rows of (4.3) are orthogonal filters (since it is a tight frame and the rows are orthonormal vectors from a  $3 \times 3$  unitary



**Figure 4.13:** Magnitude response of the prototype filter  $P(z)$  of length 5.



**Figure 4.14:** Spectrogram of a speech segment. 64 frequency bins are evaluated between 0 and 4 KHz, and a triangle window with 50% overlap is used.

matrix via Naimark's theorem), so we can take (with normalization)

$$P_0(z) = \frac{1}{3}(\sqrt{2} - \frac{1}{\sqrt{2}}z^{-1} - \frac{1}{\sqrt{2}}z^{-2}), \quad P_1(z) = \frac{1}{\sqrt{6}}(1 - z^{-1}).$$

We can now get the prototype filter  $P(z)$  as

$$P(z) = P_0(z^2) + z^{-1}P_1(z^2) = \frac{1}{3\sqrt{2}}(2 + \sqrt{3}z^{-1} - z^{-2} - \sqrt{3}z^{-3} - z^{-4}),$$

a longer solutions than  $N = 2$ , with the magnitude response as in Figure 4.13. Another example, with  $N = 2$  and  $M = 4$  is left as Exercise ??.

**Application to Power Spectral Density Estimation** In Chapter 2, Section 2.3.2, we discussed the computation of periodograms as a widely used application of complex exponential-modulated filter banks. It is a process of estimating and computing the local power spectral density. That process has a natural filter-bank implementation described in the same section. The prototype filter  $p$  computes the windowing, and the modulation computes the DFT (see Figure 2.9 and Table 2.1). The down-sampling factor  $N$  can be smaller than  $M$ , which is when we have a frame. For example, with  $N = M/2$ , we have 50% overlap, and if  $N = 1$  (that is, no down-sampling) we are computing a sliding window DFT (with  $(M - 1)/M\%$  overlap). When both the time redundancy and the number of frequencies increases, this time-frequency frame approaches a continuous transform called the *local Fourier transform*, treated in detail in Chapter 5. A typical example for calculating the

periodogram of a speech signal uses  $M = 64$ ,  $N = 32$  (or 50% overlap) and a Hamming window. No averaging of the power spectral density coefficient is used. The result is shown in Figure 4.14. This display is often called a *spectrogram* in the speech processing literature. From this figure, one clearly sees the time-frequency behavior typical of signals that have time-varying spectra.

#### 4.4.2 Cosine-Modulated Local Fourier Frames

In Chapter 2, we saw that a possible escape from the restriction imposed by the discrete Balian-Low theorem was to replace complex-exponential modulation with an appropriate cosine modulation, with an added advantage that all filters are real if the prototype is real. While frames in general offer another such escape, cosine-modulated frames provides even more options.

**Cosine Modulation** Given a prototype filter  $p$ , one of the possible ways to use the cosine modulation is (other ways leading to different classes of cosine-modulated filter banks exist; see *Further Reading* for pointers):

$$\begin{aligned} g_{i,n} &= p_n \cos\left(\frac{2\pi}{2M}\left(i + \frac{1}{2}\right)n + \theta_i\right) \\ &= p_n \frac{1}{2} \left[ e^{j\theta_i} W_{2M}^{-(i+1/2)n} + e^{-j\theta_i} W_{2M}^{(i+1/2)n} \right], \\ G_i(z) &= \frac{1}{2} \left[ e^{j\theta_i} P(W_{2M}^{(i+1/2)} z) + e^{-j\theta_i} P(W_{2M}^{-(i+1/2)} z) \right], \\ G_i(e^{j\omega}) &= \frac{1}{2} \left[ e^{j\theta_i} P(e^{j(\omega - (2\pi/2M)(i+1/2))}) + e^{-j\theta_i} P(e^{j(\omega + (2\pi/2M)(i+1/2))}) \right], \end{aligned} \quad (4.88)$$

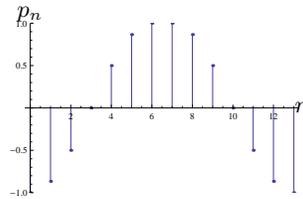
for  $i = 0, 1, \dots, M-1$ , and  $\theta_i$  is a phase factor that gives us flexibility in designing the representation. Compare the above with (4.86) for the complex-exponential modulation; the difference is that given a real prototype filter, all the other filters are real. Compare it also with (2.27) for the cosine modulation in bases. The two expressions are identical; the difference is in the sampling factor  $N$ , smaller here than the number of filters  $M$ .

**Matrix View** We look at a particular class of cosine-modulated frames, those with filters of length  $L = 2N$ , a natural extension of the LOTs from Section 2.4.1 (see also *Further Reading*). We choose the same phase factor as in (2.29), leading to

$$g_{i,n} = p_n \cos\left(\frac{2\pi}{2M}\left(i + \frac{1}{2}\right)\left(n - \frac{M-1}{2}\right)\right), \quad (4.89)$$

for  $i = 0, 1, \dots, M-1$ ,  $n = 0, 1, \dots, 2N-1$ . We know that for a rectangular prototype window,  $p_n = 1/\sqrt{M}$ , the above filters form a tight frame since they were obtained directly by seeding the LOT with the rectangular prototype window (compare (4.89) to (2.30)). We follow the same analysis as we did in Section 2.4.1.





**Figure 4.15:** An example prototype window design for  $N = 7$ .

$p_0$	$p_1$	$p_2$	$p_3$	$p_4$	$p_5$	$p_6$
-1	$-\sqrt{3}/2$	$-1/2$	0	$1/2$	$\sqrt{3}/2$	1

**Table 4.4:** Prototype window used in Figure 4.15. The prototype window is symmetric, so only half of the coefficients are shown.

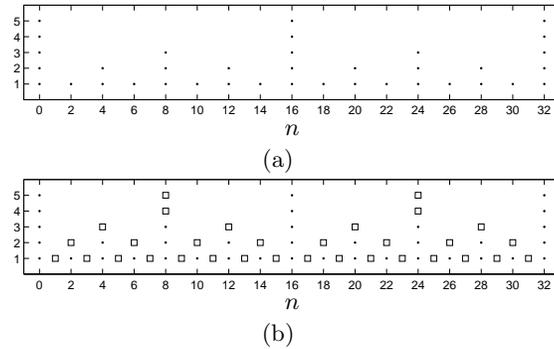
## 4.5 Wavelet Frames

We now move from the Fourier-like frames to those that are wavelet-like. We have seen examples of moving from bases to frames (DFT to harmonic tight frame, for example, see Table 4.7), and we would like to do that in the wavelet case as well. We start with the most obvious way to generate a frame from the DWT: by removing some downsamplers. Then we move on to the predecessor of wavelet frames originating in the work of Burt and Adelson on pyramid coding, and close the section with the fully-redundant frames called *shift-invariant DWT*.

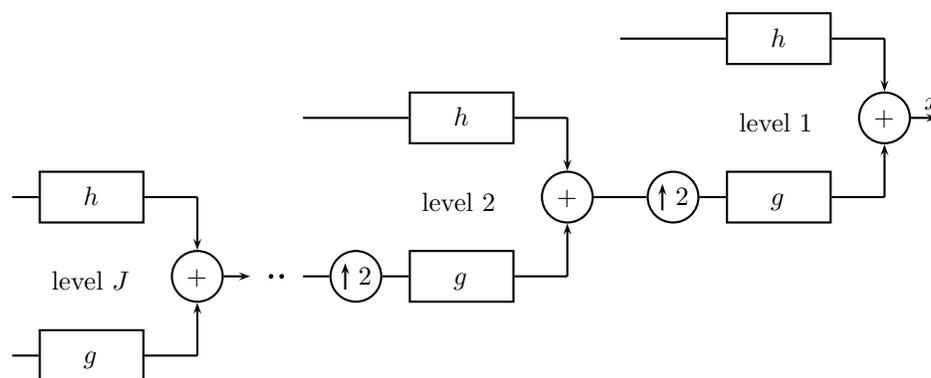
### 4.5.1 Oversampled DWT

How do we add redundancy starting from the DWT? We already mentioned that an obvious way to do that was to remove some downsamplers, thereby getting a finer time localization. Consider Figure 4.16(a), showing the sampling grid for the DWT (corresponding to the wavelet tiling from Figure 3.7(d)): at each subsequent level, only half of the points are present (half of the basis functions exist at that scale). Ideally, we would like to, for each scale, insert additional points (one point between every two). This can be achieved by having a DWT tree with the samplers removed at all free branches (see Figure 4.17). We call this scheme *oversampled DWT*, also known as the partial DWT (see *Further Reading*). The redundancy of this scheme at level  $\ell$  is  $A_j = 2$ , for a total redundancy of  $A = 2$ . The sampling grid with  $J = 4$  is depicted in Figure 4.16(b).

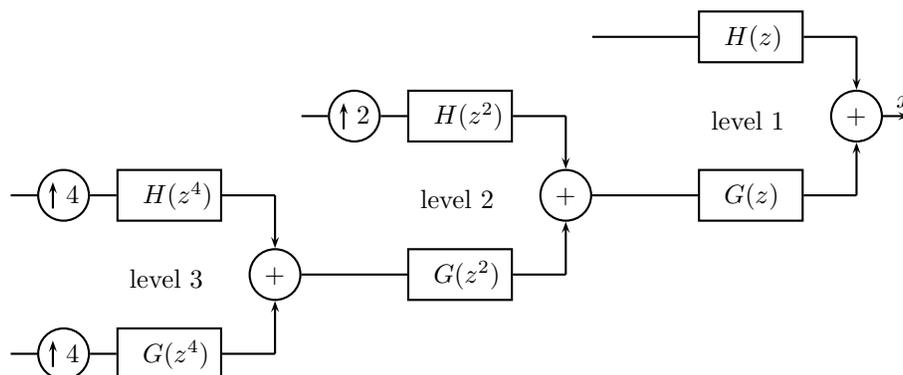
**EXAMPLE 4.8 (OVERSAMPLED DWT)** Let us now look at a simple example with  $J = 3$ . By moving upsamplers across filters, the filter bank in Figure 4.17 reduces to the one in Figure 4.18. The equivalent filters are then (we leave the



**Figure 4.16:** Sampling grids corresponding to the time-frequency tilings of (a) the DWT (points—nonredundant) and (b) the oversampled DWT (squares—redundant).



**Figure 4.17:** The synthesis part of the filter bank implementing the oversampled DWT. The samplers are omitted at all the inputs into the bank. The analysis part is analogous.



**Figure 4.18:** The synthesis part of the equivalent filter bank implementing the oversampled DWT with  $J = 3$  levels. The analysis part is analogous.

filter bank in its tree form as this is how it is actually implemented):<sup>40</sup>

$$H^{(1)}(z) = H(z), \quad (4.92a)$$

$$H^{(2)}(z) = G(z)H(z^2), \quad (4.92b)$$

$$H^{(3)}(z) = G(z)G(z^2)H(z^4), \quad (4.92c)$$

$$G^{(3)}(z) = G(z)G(z^2)G(z^4), \quad (4.92d)$$

and the frame can be expressed as

$$\Phi = \{h_{n-k}^{(1)}, h_{n-2k}^{(2)}, h_{n-4k}^{(3)}, g_{n-4k}^{(3)}\}_{k \in \mathbb{Z}}. \quad (4.93)$$

The template vector  $h$  moves by 1,  $h^{(2)}$  moves by multiples of 2, and  $h^{(3)}$  and  $g^{(3)}$  move by multiples of 4. Thus, the basic block of the infinite matrix is of size  $8 \times 16$  (the smallest period after which it starts repeating itself, redundancy of 2) and it moves by multiples of 8. However, even for filters such as Haar for which the DWT would become a block transform (the infinite matrix  $\Phi$  is block diagonal, see (3.4)), here this is not the case. Substituting Haar filters (see Table 1.8) into the expressions for  $H^{(1)}$ ,  $H^{(2)}$ ,  $H^{(3)}$  and  $G^{(3)}$  above, we get

$$H^{(1)}(z) = \frac{1}{\sqrt{2}}(1 - z^{-1}),$$

$$H^{(2)}(z) = \frac{1}{2}(1 + z^{-1} - z^{-2} - z^{-3}),$$

$$H^{(3)}(z) = \frac{1}{2\sqrt{2}}(1 + z^{-1} + z^{-2} + z^{-3} - z^{-4} - z^{-5} - z^{-6} - z^{-7}),$$

$$G^{(3)}(z) = \frac{1}{2\sqrt{2}}(1 + z^{-1} + z^{-2} + z^{-3} + z^{-4} + z^{-5} + z^{-6} + z^{-7}).$$

Renaming the template frame vectors, we can rewrite the frame  $\Phi$  as

$$\begin{aligned} \varphi_{k,n} &= h_{n-k}^{(1)}, & k = 0, 1, \dots, 7; \\ \varphi_{8+k,n} &= h_{n-2k}^{(2)}, & k = 0, 1, 2, 3; \\ \varphi_{12+k,n} &= h_{n-4k}^{(3)}, & k = 0, 1; \\ \varphi_{14+k,n} &= g_{n-4k}^{(3)}, & k = 0, 1; \end{aligned}$$

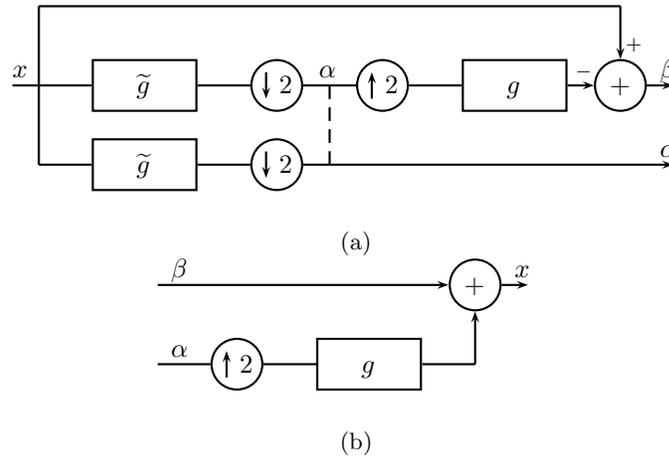
$$\Phi = \{\varphi_{i,n-8k}\}_{k \in \mathbb{Z}, i=0, 1, \dots, 15}. \quad (4.94)$$

Compare this to the DWT example from Section 3.1.

## 4.5.2 Pyramid Frames

Pyramid frames were introduced for coding in 1983 by Burt and Adelson. Although redundant, the pyramid coding scheme was developed for compression of images and was recognized in the late 1980s as one of the precursors of wavelet octave-band decompositions. The scheme works as follows: First, a coarse approximation  $\alpha$  is

<sup>40</sup>Remember that superscript ( $\ell$ ) denotes the level in the tree.



**Figure 4.19:** The (a) analysis and (b) synthesis part of the pyramid filter bank. This scheme implements a frame expansion. The dashed line indicates the actual implementation, as in reality, the lowest branch would not be implemented; it is indicated here for clarity and parallelism with two-channel filter banks.

derived (an example of how this could be done is in Figure 4.19).<sup>41</sup> Then, from this coarse version, the original is predicted (in the figure, this is done by upsampling and filtering) followed by calculating the prediction error  $\beta$ . If the prediction is good (as is the case for most natural images that have a lowpass characteristic), the error will have a small variance and can thus be well compressed. The process can be iterated on the coarse version. The outputs of the analysis filter bank are:

$$\alpha(z) \stackrel{(a)}{=} \frac{1}{2} \left[ \tilde{G}(z^{1/2})X(z^{1/2}) + \tilde{G}(-z^{1/2})X(-z^{1/2}) \right], \quad (4.95a)$$

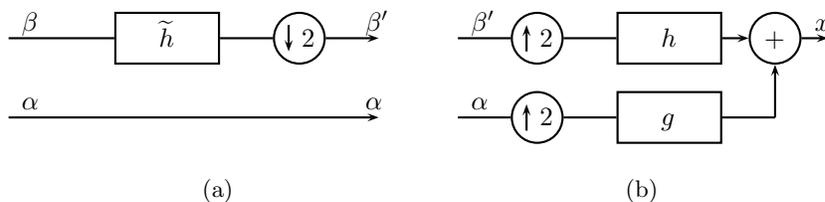
$$\begin{aligned} \beta(z) &\stackrel{(b)}{=} X(z) - \frac{1}{2}G(z) \left[ \tilde{G}(z)X(z) + \tilde{G}(-z)X(-z) \right] \\ &= X(z) - G(z)\alpha(z^2), \end{aligned} \quad (4.95b)$$

where (a) follows from (3.201a) and (b) from (1.77a). To reconstruct, we simply upsample and interpolate the prediction  $\alpha(z)$  and add it back to the prediction error  $\beta(z)$ :

$$G(z)\alpha(z^2) + \beta(z) = X(z). \quad (4.96)$$

Upsampling and interpolating is, however, only one way to obtain the prediction back at full resolution; any appropriate operator (even a nonlinear one) could have been simply inverted by subtraction. We can also see that in the figure, the redundancy of the system is 50%;  $\alpha$  is at half resolution while  $\beta$  is at full resolution, that is, after analysis, we have 50% more samples than we started with. With the analysis given in Figure 4.19(a), we now have several options:

<sup>41</sup>While in the figure the intensity of the coarse approximation  $\alpha$  is obtained by linear filtering and downsampling, this need not be so; in fact, one of the powerful features of the original scheme is that any operator can be used, not necessarily linear.



**Figure 4.20:** The pyramid filter bank implementing a basis expansion. With  $\{g, \tilde{g}, h, \tilde{h}\}$  a biorthogonal set, the scheme implements a biorthogonal basis expansion, while with  $g$  and  $\tilde{g}$  orthogonal, that is,  $\tilde{g}_n = g_{-n}$  and  $g$  satisfies (1.13), the scheme implements an orthonormal basis expansion. The output  $\beta$  from Figure 4.19(a) goes through (a) filtering and downsampling creating a new output  $\beta'$ . (b) Synthesis part.

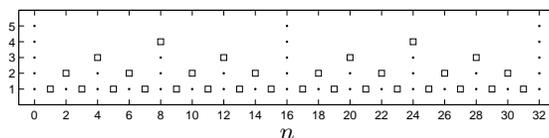
- Synthesis is performed by upsampling and interpolating  $\alpha$  by  $g$  as in Figure 4.19(b). In this case, the resulting scheme is clearly redundant, as we have just discussed, and implements a *frame expansion*, which can be either:
  - (i) *general*, when filters  $g$  and  $\tilde{g}$  are biorthogonal (they satisfy (1.66)), or,
  - (ii) *tight*, when filters  $g$  and  $\tilde{g}$  are orthogonal, that is,  $\tilde{g}_n = g_{-n}$  and  $g$  satisfies (1.13). We illustrate this case in Example 4.9.
- The analysis goes through one more stage, as in Figure 4.20(a), and synthesis is performed as in Figure 4.20(b). In this case, the scheme implements a *basis expansion*, which can be either (both are illustrated in Exercise ??):
  - (i) *biorthogonal*, when filters  $g$  and  $\tilde{g}$  are biorthogonal, or,
  - (ii) *orthonormal*, when filters  $g$  and  $\tilde{g}$  are orthogonal.

**EXAMPLE 4.9** We use the pyramid filter bank as in Figure 4.19. Let us assume that  $g$  is the Haar lowpass filter from (1.1a) and that  $\tilde{g}_n = g_{-n}$ . Then we know from Chapter 1, that  $\beta$  is nothing else but the output of the highpass branch, given in (?). For every two input samples, while  $\alpha$  produces one output sample,  $\beta$  produces two output samples; thus, the redundancy. We can write this as:

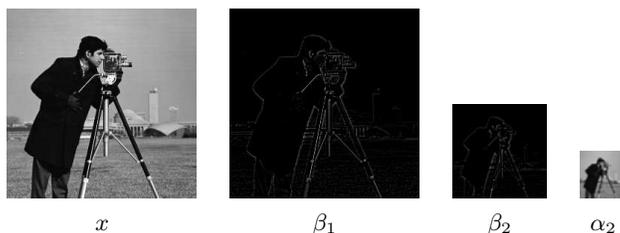
$$\begin{bmatrix} \alpha_n \\ \beta_{2n} \\ \beta_{2n+1} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix}}_{\tilde{\Phi}^T} \begin{bmatrix} x_{2n} \\ x_{2n+1} \end{bmatrix}.$$

We know, however, from our previous discussion that the above matrix is the dual frame matrix  $\tilde{\Phi}^T$ . Finding its canonical dual, we get that  $\Phi = \tilde{\Phi}$ , and thus, this pyramid scheme implements a tight frame expansion.

The redundancy for pyramid frames is  $A_1 = 3/2$  at level 1,  $A_2 = 7/4$  at level 2, leading to  $A_\infty = 2$  (see Figure 4.21), far less than the shift-invariant DWT construction we will see in a moment. Thanks to this constant redundancy, pyramid coding



**Figure 4.21:** Sampling grid corresponding to the time-frequency tiling of the pyramid coding scheme (points—nonredundant, squares—redundant).



**Figure 4.22:** Two-level pyramid decomposition of an image  $x$ . A first-level coarse approximation  $\alpha_1$  is computed. A first-level prediction error  $\beta_1$  is obtained as the difference of  $x$  and the prediction calculated on  $\alpha_1$ . A second-level coarse approximation  $\alpha_2$  is computed. A second-level prediction error  $\beta_2$  is obtained as the difference of  $\alpha_1$  and the prediction calculated on  $\alpha_2$ . The scheme is redundant, as the total number of samples in expansion coefficients  $\beta_1, \beta_2, \alpha_2$  is  $(1 + 1/4 + 1/16)$  times the number original image samples, yielding redundancy of about 31%.

has been used together with directional coding to form the basis for nonseparable multidimensional frames called contourlets (see *Further Reading*). An example of a pyramid decomposition of an image is given in Figure 4.22.

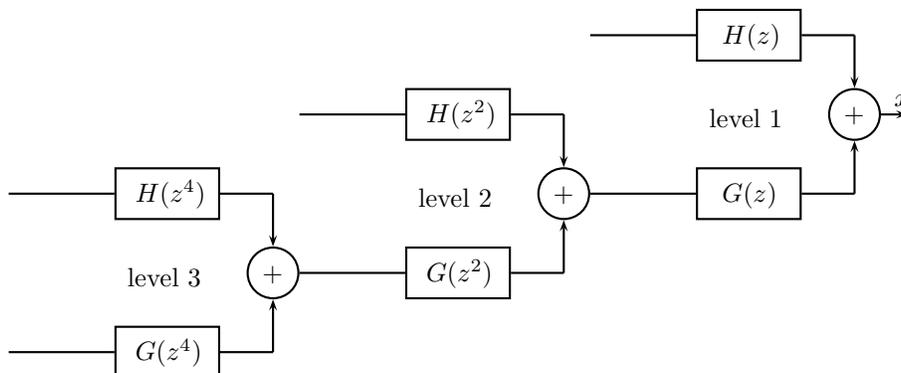
### 4.5.3 Shift-Invariant DWT

The shift-invariant DWT is basically the nondownsampling DWT (an example for  $J = 3$  levels is shown in Figure 4.23). It is sometimes called *stationary wavelet transform*, or, *algorithme à trous*,<sup>42</sup> due to its implementation algorithm by the same name (see Section 4.6.1).

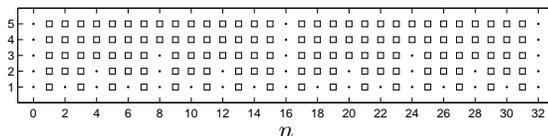
Let  $g$  and  $h$  be the filters used in this filter bank. At level  $\ell$  we will have equivalent upsampling by  $2^\ell$ , which means that the filter moved across the upsampler will be upsampled by  $2^\ell$ , inserting  $(2^\ell - 1)$  zeros between every two samples and thus creating holes (thus *algorithm with holes*).

Figure 4.24 shows the sampling grid for the shift-invariant DWT, from where it is clear that this scheme is completely redundant, as all points are computed. This is in contrast to a completely nonredundant scheme such as the DWT shown in Figure 4.16(a). In fact, while the redundancy per level of this algorithm grows exponentially since  $A_1 = 2, A_2 = 4, \dots, A_J = 2^J, \dots$ , the total redundancy for  $J$  levels is linear, as  $A = A_J 2^{-J} + \sum_{\ell=1}^J A_\ell 2^{-\ell} = (J+1)$ . This growing redundancy is

<sup>42</sup>From French for *algorithm with holes*, coming from the computational method that can take advantage of upsampled filter impulse responses, discussed in Section 4.6.1.



**Figure 4.23:** The synthesis part of the equivalent 3-channel filter bank implementing the shift-invariant DWT with  $J = 3$  levels. The analysis part is analogous and filters are given in (4.92). This is the same scheme as in Figure 4.18 with all the upsamplers removed.



**Figure 4.24:** Sampling grid corresponding to the time-frequency tiling of the shift-invariant DWT (points—nonredundant, squares—redundant).

the price we pay for shift invariance as well as the simplicity of the algorithm. The 2D version of the algorithm is obtained by extending the 1D version in a separable manner, leading to the total redundancy of  $A = A_J 2^{-J} + 3 \sum_{\ell=1}^J A_\ell 2^{-\ell} = (3J + 1)$ . Exercise ?? illustrates the redundancy of such a frame.

## 4.6 Computational Aspects

### 4.6.1 The Algorithm à Trous

This algorithm was introduced as a fast implementation of the dyadic (continuous) wavelet transform by Holschneider, Kronland-Martinet, Morlet, and Tchamitchian in 1989, and corresponds to the DWT with samplers removed. We introduced it in Section 4.5 as shift-invariant DWT and showed an example for  $J = 3$  in Figure 4.23. The equivalent filters in each branch are computed first, and then, the samplers are removed. Because the equivalent filters are convolutions with upsampled filters, the algorithm can be efficiently computed due to *holes* produced by upsampling.

### 4.6.2 Efficient Gabor and Spectrum Computation

### 4.6.3 Efficient Sparse Frame Expansions

#### Matching Pursuit

---

**aTrous**( $\alpha^{(0)}$ )  
**Input:**  $x = \alpha^{(0)}$ , the input signal.  
**Output:**  $\alpha^{(J)}, \beta^{(\ell)}, \ell = 1, 2, \dots, J$ , transform coefficients.

---

```

initialize
for  $\ell = 1$  to  $J$  do
   $\alpha^{(\ell)} = \alpha^{(\ell-1)} * (\uparrow 2^{\ell-1})g$ 
   $\beta^{(\ell)} = \alpha^{(\ell-1)} * (\uparrow 2^{\ell-1})h$ 
end for
return  $\alpha^{(J)}, \beta^{(\ell)}, \ell = 1, 2, \dots, J$ 

```

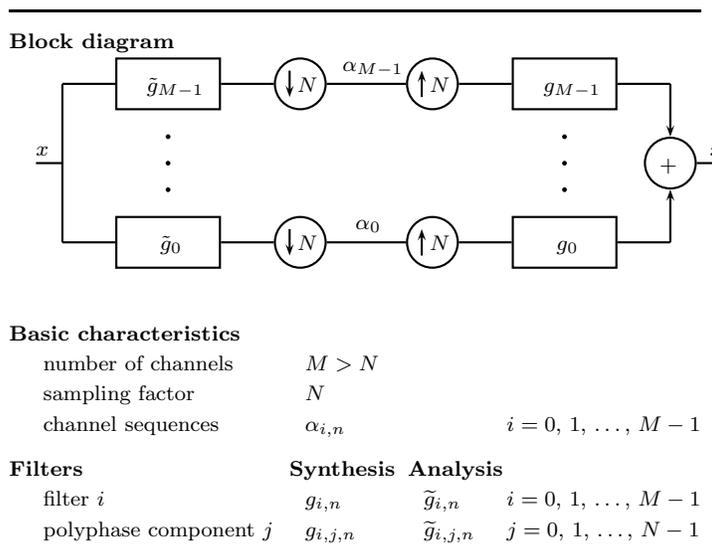
**Table 4.5:** Algorithm à trous implementing the shift-invariant DWT. Upsampling an impulse response  $g$  by a factor of  $n$  is denoted by  $(\uparrow n)g$ .

### Orthonormal Matching Pursuit

#### Linear Programming

### Chapter at a Glance

This chapter relaxed the constraint of nonredundancy bases carry, using frames to achieve robustness and freedom in choosing not only the *best expansion*, but also, given a fixed expansion, the *best expansion coefficients* under desired constraints. We introduced these mostly on finite-dimensional frames, as they can be easily visualized via rectangular matrices. The infinite-dimensional frames we discussed were only those implementable by oversampled filter banks, summarized in Table 4.6.



**Table 4.6:** Oversampled filter bank.

	Block transforms	Overlapped transforms (Fourier-like)	Time-frequency constraints (wavelet-like)
Bases	DFT	LOT	DWT
Frames	HTF	Local Fourier	Oversampled DWT

**Table 4.7:** Bases versus frames.

We discussed two big classes of frames following their counterparts in bases: local Fourier frames and wavelet frames. Table 4.7 depicts relationships existing between various classes of bases and frames. For example, the block-transform counterpart of the DFT are the harmonic tight frames, while the same for the LOT will be local Fourier frames, obtained by both complex-exponential modulation as well as cosine modulation. By increasing the support of basis functions we can go from the DFT to the LOT, and similarly, from harmonic tight frames to local Fourier frames. Imposing time-frequency constraints leads to new

classes of representations, such as the DWT, whose frame counterpart is the oversampled DWT.

## Historical Remarks

In the signal processing and harmonic analysis communities, frames are generally considered to have been born in 1952 in the paper by Duffin and Schaeffer [36]. Despite being over half a century old, frames gained popularity only in the 1990s, due mostly to the work of three wavelet pioneers—Daubechies, Grossman and Meyer [32]. An important piece to understanding frames came with Naimark's theorem, known for a long time in operator algebra and used in quantum information theory, and rediscovered by several people in the 1990s, among others, Han and Larson [42]; they came up with the idea that a frame could be obtained by compressing a basis in a larger space.

The idea behind the class of complex exponential-modulated frames, consisting of many families, dates back to Gabor [40] with insight of constructing bases by modulation of a single prototype function. Gabor originally used complex-exponential modulation, and thus, all those families with the same type of modulation are termed *complex exponential-modulated frames*, or sometimes, Gabor frames. Other types of modulation are possible, such as cosine modulation, and again, all those families with cosine modulation are termed *cosine-modulated frames*.

Frame-like ideas, that is, building redundancy into a signal expansion, can be found in numerous fields, from source and channel coding, to communications, classification, operator and quantum theory.

## Further Reading

**Books and Textbooks** The sources on frames are the book by Daubechies [31], a text by Christensen [22] a number of classic papers [18, 30, 42, 43] as well as an introductory tutorial on frames by Kovačević and Chebira [54].

**Results on Frames** A thorough analysis of oversampled filter banks implementing frame expansions is given in [12, 27, 28]. Following up on the result of Benedetto and Fickus [8] on minimizing frame potential from Section 4.2.1, Cassaza, Fickus, Kovačević, Leon and Tremain extended the result to nonequal-norm tight frames, giving rise to the *fundamental inequality*, which has ties to the capacity region in synchronous CDMA systems [110]. Casazza and Kutyniok in [20] investigated Gram-Schmidt-like procedure for producing tight frames. In [9], the authors introduce a quantitative notion of redundancy through local redundancy and a redundancy function, applicable to all finite-dimensional frames.

**Local Fourier Frames** For finite-dimensional frames, similar ideas to those of harmonic tight frames have appeared in the work by Eldar and Bölcskei [37] under the name *geometrically uniform frames*, frames defined over a finite Abelian group of unitary matrices both with a single as well as multiple generators. Harmonic tight frames have been generalized in the works by Vale and Waldron [102], as well as Casazza and Kovačević [19].

Harmonic tight frames, as well as equiangular frames (where  $|\langle \varphi_i, \varphi_j \rangle|$  is a constant) [71], have strong connections to *Grassmannian frames*. In a comprehensive paper [89], Strohmer and Heath discuss those frames and their connection to Grassmannian packings, spherical codes, graph theory and Welch Bound sequences (see also [48]).

The lack of infinite-dimensional bases with good time and frequency localization, the result of the discrete Balian-Low theorem, prompted the development of oversampled filter banks that use complex-exponential modulation. They are known under various names: *oversampled DFT filter banks*, *complex exponential-modulated filter banks*, *short-time Fourier filter banks* and *Gabor filter banks* and have been studied in [10–12,26,39,88]. Bölcskei and Hlawatsch in [11] have studied the other type of modulation, cosine. The connection between these two classes is deep as there exists a general decomposition of the frame operator corresponding to a cosine-modulated filter bank as the sum of the frame operator of the underlying complex exponential-modulated frame and an additional operator, which vanishes under certain conditions [6]. The *lapped tight frame transforms* were proposed as a way to obtain a large number of frames by seeding from LOTs [21,76].

**Wavelet Frames** Apart from those already discussed, like pyramid frames [15], many other wavelet-like frame families have been proposed, among them, the *dual-tree complex wavelet transform*, a nearly shift-invariant transform with redundancy of only 2, introduced by Kingsbury [51–53]. Selesnick in [77,78] followed with the *double-density DWT* and variations, which can approximately be implemented using a 3-channel filter bank with sampling by 2, again nearly shift invariant with redundancy that tends towards 2 when iterated. Some other variations include *power-shiftable DWT* [81] or *partial DWT* [85], which removes samplers at the first level but leaves them at all other levels, with redundancy  $A_j = 2$  at each level and again near shift invariance. Bradley in [13] introduces the *overcomplete DWT*, the DWT with critical sampling for the first  $k$  levels followed by the shift-invariant DWT for the last  $j - k$  levels.

**Multidimensional Frames** Apart from obvious, tensor-like, constructions of multidimensional frames, true multidimensional solutions exist. The oldest multidimensional frame seems to be the *steerable pyramid* introduced by Simoncelli, Freeman, Adelson and Heeger in 1992 [81], following on the previous work by Burt and Adelson on pyramid coding [15]. The steerable pyramid possesses many nice properties, such as joint space-frequency localization, approximate shift invariance, approximate tightness and approximate rotation invariance. An excellent overview of the steerable pyramid and its applications is given on Simoncelli's web page [80].

Another multidimensional example is the work of Do and Vetterli on *contourlets* [25,34], motivated by the need to construct efficient and sparse representations of intrinsic geometric structure of information within an image. The authors combine the ideas of pyramid filter banks [33] with directional processing, to obtain contourlets, expansions capturing contour segments. These are almost critically sampled, with redundancy of 1.33.

Some other examples include [58] where the authors build both critically-sampled and shift-invariant 2D DWT. Many "-lets" are also multidimensional frames, such as *curvelets* [16,17] and *shearlets* [56]. As the name implies, curvelets are used to approximate curved singularities in an efficient manner [16,17]. As opposed to wavelets, which use dilation and translation, shearlets use dilation, shear transformation and translation, and possess useful properties such as directionality, elongated shapes and many others [56].

**Applications of Frames** Frames have become extremely popular and have been used in many application fields. The text by Kovačević and Chebira [54] contains an overview of many of these and a number of relevant references. In some fields, frames have been used for years, for example in CDMA systems, in the work of Massey and Mittelholzer [63] on

Welch bound and sequence sets for CDMA systems. It turns out that the Welch bound is equivalent to the frame potential minimization inequality. The equivalence between unit-norm tight frames and Welch bound sequences was shown in [89]. Waldron formalized that equivalence for general tight frames in [111], and consequently, tight frames are referred in some works as Welch bound sequences [92].

## Chapter 5

# Local Fourier Transforms, Frames and Bases on Functions

## Contents

5.1	Introduction	178
5.2	Local Fourier Transform	178
5.3	Local Fourier Frame Series	188
5.4	Local Fourier Series	188
5.5	Computational Aspects	188
	Chapter at a Glance	188
	Historical Remarks	188
	Further Reading	188

*Dear Reader,*

*This chapter needs to be finished. The only existing section, Section 5.2 has been proofread and integrated with the previous text. The rest of the sections are yet to be written.*

*Please read on.*

— *MV, JK, and VKG*

The aim of this chapter follows that of Chapter 2, but for functions. We look for ways to localize the analysis Fourier transform provides by windowing the complex exponentials. As before, this will improve the time localization of the corresponding transform at the expense of the frequency localization. The original idea dates back to Gabor, and thus *Gabor transform* is frequently used; *windowed Fourier transform* and *short-time Fourier transform* are as well. We choose the intuitive *local Fourier transform*, as a counterpart to *local Fourier bases* from Chapter 2 and *local Fourier frames* from Chapter 4.

We start with the most redundant one, local Fourier transform, and then sample to obtain local Fourier frames. With critical sampling we then try for local Fourier bases, where, not surprisingly after what we have seen in Chapter 2, bases with simultaneously good time and frequency localization do not exist, the result

known as Balian-Low theorem. Again as in Chapter 2, cosine local Fourier bases do exist, as do wavelet ones we discuss in the next chapter.

## 5.1 Introduction

### Fourier Series Basis Expansion

#### Localization Properties of the Fourier Series

#### Chapter Outline

We start with the most redundant one version of the local Fourier transform, the local Fourier transform in Section 5.2, and then sample to obtain local Fourier frames in Section 5.3. With critical sampling we then try for local Fourier bases in Section 5.4, where, not surprisingly after what we have seen in Chapter 2, complex exponential-modulated local Fourier bases with simultaneously good time and frequency localization do not exist, the result known as Balian-Low theorem. Again as in Chapter 2, cosine-modulated local Fourier bases do exist, as do wavelet ones we discuss in the next chapter.

*Notation used in this chapter:* The prototype window function in this chapter is named  $p(t)$ ; this is for consistency with the prototype window sequences used in Chapters 2 and 4;  $g(t)$  is more commonly seen in the literature.  $\square$

## 5.2 Local Fourier Transform

Given a function  $x(t)$ , we start with its Fourier transform  $X(\omega)$  as in *Definition 4.10*. We analyze  $x(t)$  locally by using a prototype window function  $p(t)$ . We will assume  $p(t)$  is symmetric,  $p(t) = p(-t)$ , and real. The prototype function should be smooth as well, in particular, it should be smoother than the function to be analyzed.<sup>43</sup>

### 5.2.1 Definition of the Local Fourier Transform

We can look at our windowing with the prototype function  $p(t)$  in two ways:

- (i) We window the function  $x(t)$  as

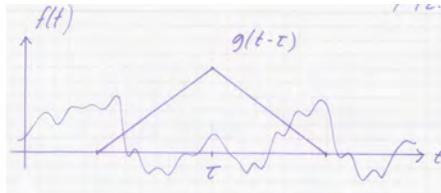
$$x_\tau(t) = p(t - \tau)x(t), \quad (5.1)$$

and then take its Fourier transform (4.42a),

$$X_\tau(\omega) = \langle x_\tau, v_\omega \rangle = \int_{t \in \mathbb{R}} x_\tau(t) e^{-j\omega t} dt = \int_{t \in \mathbb{R}} x(t) p(t - \tau) e^{-j\omega t} dt, \quad (5.2)$$

for  $\omega \in \mathbb{R}$ .

<sup>43</sup>Otherwise, it will interfere with the smoothness of the function to be analyzed; see Section 5.2.2.



**Figure 5.1:** Local Fourier transform. The prototype function  $p(t)$  is centered at  $\tau$ , and thus, the Fourier transform only *sees* the neighborhood around  $\tau$ . For simplicity, a triangle prototype function is shown; in practice, smoother ones are used.

(ii) We window the complex exponentials  $v_\omega(t) = e^{j\omega t}$  yielding

$$\begin{aligned} g_{\Omega, \tau}(t) &= p(t - \tau) e^{j\Omega t}, \\ G_\tau(\omega) &= e^{-j(\omega - \Omega)\tau} P(\omega - \Omega), \end{aligned} \quad (5.3)$$

for  $\tau, \Omega \in \mathbb{R}$ , and then define a new transform by taking the inner product between  $x$  and  $g_{\Omega, \tau}(t)$  as

$$X(\Omega, \tau) = \langle x, g_{\Omega, \tau} \rangle = \int_{t \in \mathbb{R}} x(t) p(t - \tau) e^{-j\Omega t} dt, \quad (5.4)$$

that is, this new transform  $X(\Omega, \tau)$  is the Fourier transform of the windowed function  $x_\tau$  as in (5.2).

From the construction, it is clear why this is called local Fourier transform, as shown in Figure 5.1. We are now ready to formally define it:

**DEFINITION 5.1 (LOCAL FOURIER TRANSFORM)** The local Fourier transform of a function  $x(t)$  is a function of  $\Omega, \tau \in \mathbb{R}$  given by

$$X(\Omega, \tau) = \langle x, g_{\Omega, \tau} \rangle = \int_{t \in \mathbb{R}} x(t) p(t - \tau) e^{-j\Omega t} dt, \quad \Omega, \tau \in \mathbb{R}. \quad (5.5a)$$

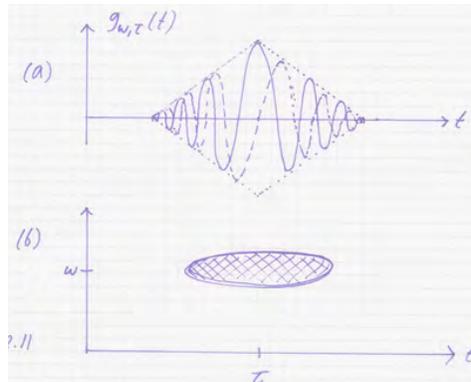
The inverse local Fourier transform of  $X(\Omega, \tau)$  is

$$x(t) = \frac{1}{2\pi} \int_{\Omega \in \mathbb{R}} \int_{\tau \in \mathbb{R}} X(\Omega, \tau) g_{\Omega, \tau}(t) d\Omega d\tau. \quad (5.5b)$$

To denote such a local Fourier-transform pair, we write:

$$x(t) \xleftrightarrow{\text{LFT}} X(\Omega, \tau).$$

We will prove the inversion formula (5.5b) in a moment. For the analysis of a function  $x(t)$ , the  $X(\Omega, \tau)$  uses time-frequency atoms  $g_{\Omega, \tau}(t)$  that are centered



**Figure 5.2:** Time-frequency atom used in the local Fourier transform. (a) Time-domain waveform  $g_{\omega, \tau}(t)$ . The prototype function is the triangle function, and the real and imaginary parts of the complex exponential-modulated prototype function are shown. (b) Schematic time-frequency footprint of  $g_{\omega, \tau}(t)$ .

around  $\Omega$  and  $\tau$ , as shown schematically in Figure 5.2. The local Fourier transform is highly redundant, mapping a one-dimensional function  $x(t)$  into a two-dimensional transform  $X(\Omega, \tau)$ .

**Prototype Window Function** The prototype function  $p(t)$  is critical in the local Fourier transform. The classical choice for  $p(t)$  is the unit-norm version of the Gaussian function given in (4.11a) with  $\gamma = (2\alpha/\pi)^{1/4}$ :

$$p(t) = \left(\frac{2\alpha}{\pi}\right)^{1/4} e^{-\alpha(t-\mu)^2}, \quad (5.6)$$

where  $\alpha$  is a scale parameter allowing us to tune the time resolution of the local Fourier transform.

Another classic choice is the unit-norm sinc function from (4.75),

$$p(t) = \sqrt{\frac{\omega_0}{2\pi}} \frac{\sin \omega_0 t/2}{\omega_0 t/2}, \quad (5.7)$$

that is, a perfect lowpass of bandwidth  $|\omega| \leq \omega_0/2$ . Here, the scale parameter allows us to tune the frequency resolution of the local Fourier transform.

Other prototype functions of choice include rectangular, triangular (hat) or higher-degree spline functions, as well as other classic prototype functions from spectral analysis. An example is the *Hanning*, or, *raised cosine window* (we have seen its discrete counterpart in (3.15)), its unit-norm version defined as

$$p(t) = \begin{cases} \sqrt{\frac{2}{3\alpha}}(1 + \cos(2\pi t/\alpha)), & |t| \leq \alpha/2; \\ 0, & \text{otherwise,} \end{cases}$$

where  $\alpha$  is a scale parameter.<sup>44</sup>

**Inversion of the Local Fourier Transform** While we have taken for granted that the inversion formula (5.5b) holds, this is not a given. However, given the redundancy present in the local Fourier transform, we expect such an inversion to be possible, which we now prove.

We are going to apply the generalized Parseval's equality to (5.5b), and we thus need the Fourier transform of  $X(\Omega, \tau)$  with respect to  $\tau$ . We have that

$$\begin{aligned} X(\Omega, \tau) &= \int_{t \in \mathbb{R}} x(t) p(t - \tau) e^{-j\Omega t} dt, \\ &\stackrel{(a)}{=} \int_{t \in \mathbb{R}} p(\tau - t) x(t) e^{-j\Omega t} dt \stackrel{(b)}{=} (p * x_\Omega)(\tau), \end{aligned}$$

where (a) follows from  $p(t) = p(-t)$ , and in (b) we introduced  $x_\Omega(t) = x(t)e^{-j\Omega t}$ . Using the shift-in-frequency property (4.54), the Fourier transform of  $x_\Omega(t)$  is  $X(\omega + \Omega)$ . Then, using the convolution property (4.61), the Fourier transform of  $X(\Omega, \tau)$  with respect to  $\tau$  becomes,

$$X(\Omega, \omega) = P(\omega) X(\omega + \Omega). \quad (5.8)$$

In (5.5b), the other term involving  $\tau$  is  $g_{\Omega, \tau}(t) = p(t - \tau)e^{j\Omega t}$ . Using the shift-in-time property (4.53) and because  $p(t)$  is symmetric, the Fourier transform of  $p(t - \tau)$  with respect to  $\tau$  is

$$p(t - \tau) \xrightarrow{\text{FT}} e^{-j\omega t} P(\omega). \quad (5.9)$$

We now apply the generalized Parseval's equality (4.70b) to the right side of (5.5b):

$$\begin{aligned} &\frac{1}{2\pi} \int_{\Omega \in \mathbb{R}} \left( \int_{\tau \in \mathbb{R}} X(\Omega, \tau) p(t - \tau) e^{j\Omega t} d\tau \right) d\Omega \\ &\stackrel{(a)}{=} \frac{1}{2\pi} \int_{\Omega \in \mathbb{R}} \left( \frac{1}{2\pi} \int_{\omega \in \mathbb{R}} X(\omega + \Omega) P(\omega) P^*(\omega) e^{j\omega t} e^{j\Omega t} d\omega \right) d\Omega \\ &\stackrel{(b)}{=} \frac{1}{2\pi} \int_{\omega \in \mathbb{R}} |P(\omega)|^2 \left( \frac{1}{2\pi} \int_{\Omega \in \mathbb{R}} X(\omega + \Omega) e^{j(\omega + \Omega)t} d\Omega \right) d\omega, \\ &\stackrel{(c)}{=} x(t) \frac{1}{2\pi} \int_{\omega \in \mathbb{R}} |P(\omega)|^2 d\omega \stackrel{(d)}{=} x(t). \end{aligned}$$

where (a) follows from (5.8), (5.9) and generalized Parseval's equality (4.70b); (b) from Fubini's theorem (see *Appendix 2.A.3*) allowing for the exchange of the order of integration; (c) from the inverse Fourier transform (4.42b); and (d) from  $p$  being of unit norm and Parseval's equality (4.70a).

<sup>44</sup>In the signal processing literature, the normalization factor is usually 1/2, such that  $p(0) = 1$ .

### 5.2.2 Properties of the Local Fourier Transform

We now look into the main properties of the local Fourier transform, including energy conservation, followed by basic characteristics such as localization properties and examples, including spectrograms, which are density plots of the magnitude of the local Fourier transform.

TBD: Table with properties.

**Linearity** The local Fourier transform operator is a linear operator, or,

$$\alpha x(t) + \beta y(t) \xleftrightarrow{\text{LFT}} \alpha X(\Omega, \tau) + \beta Y(\Omega, \tau). \quad (5.10)$$

**Shift in Time** A shift in time by  $t_0$  results in

$$x(t - t_0) \xleftrightarrow{\text{LFT}} e^{-j\Omega t_0} X(\Omega, \tau - t_0). \quad (5.11)$$

This is to be expected as it follows from the shift-in-time property of the Fourier transform, (4.53). To see that,

$$\begin{aligned} \int_{t \in \mathbb{R}} p(t - \tau) x(t - t_0) e^{-j\Omega t} dt &\stackrel{(a)}{=} e^{-j\Omega t_0} \int_{t' \in \mathbb{R}} p(t' - (\tau - t_0)) x(t') e^{-j\Omega t'} dt' \\ &\stackrel{(b)}{=} e^{-j\Omega t_0} X(\Omega, \tau - t_0), \end{aligned}$$

where (a) follows from change of variable  $t' = t - t_0$ ; and (b) from the definition of the local Fourier transform (5.5a). Thus, a shift by  $t_0$  simply shifts the local Fourier transform and adds a phase factor. The former illustrates the locality of the local Fourier transform, while the latter follows from the equivalent Fourier-transform property.

**Shift in Frequency** A shift in frequency by  $\omega_0$  results in

$$e^{j\omega_0 t} x(t) \xleftrightarrow{\text{LFT}} X(\Omega - \omega_0, \tau). \quad (5.12)$$

To see this,

$$\begin{aligned} &\int_{t \in \mathbb{R}} p(t - \tau) e^{j\omega_0 t} x(t) e^{-j\Omega t} dt \\ &= \int_{t \in \mathbb{R}} p(t - \tau) x(t) e^{-j(\Omega - \omega_0)t} dt = X(\Omega - \omega_0, \tau), \end{aligned}$$

the same as for the Fourier transform. As before, a shift in frequency is often referred to as *modulation*, and is dual to the shift in time.

**Parseval's Equality** The local Fourier-transform operator is a unitary operator and thus preserves the Euclidean norm (see (2.53)):

$$\|x\|^2 = \int_{t \in \mathbb{R}} |x(t)|^2 dt = \frac{1}{2\pi} \int_{\Omega \in \mathbb{R}} \int_{\tau \in \mathbb{R}} |X(\Omega, \tau)|^2 d\Omega d\tau = \frac{1}{2\pi} \|X\|^2. \quad (5.13)$$

We now prove Parseval's equality for functions that are both in  $\mathcal{L}^1$  and  $\mathcal{L}^2$ : It should come as no surprise that to derive Parseval's equality for the local Fourier transform, we use Parseval's equality for the Fourier transform. Start with the right side of (5.13) to get

$$\begin{aligned} & \frac{1}{2\pi} \int_{\tau \in \mathbb{R}} \int_{\Omega \in \mathbb{R}} |X(\Omega, \tau)|^2 d\Omega d\tau \\ & \stackrel{(a)}{=} \frac{1}{2\pi} \int_{\Omega \in \mathbb{R}} \left( \frac{1}{2\pi} \int_{\omega \in \mathbb{R}} |X(\omega + \Omega) P(\omega)|^2 d\omega \right) d\Omega \\ & \stackrel{(b)}{=} \frac{1}{2\pi} \int_{\omega \in \mathbb{R}} |P(\omega)|^2 \left( \frac{1}{2\pi} \int_{\Omega \in \mathbb{R}} |X(\omega + \Omega)|^2 d\Omega \right) d\omega, \\ & \stackrel{(c)}{=} \|x\|^2 \frac{1}{2\pi} \int_{\omega \in \mathbb{R}} |P(\omega)|^2 d\omega \stackrel{(d)}{=} \|x\|^2, \end{aligned}$$

where (a) follows from Parseval's equality (4.70a) and (5.8); (b) from Fubini's theorem (see *Appendix 2.A.3*) allowing for the exchange of the order of integration; (c) from the inverse Fourier transform (4.42b); and (d) from  $p$  being of unit norm and Parseval's equality (4.70a).

**Redundancy** The local Fourier transform maps a function of one variable into a function of two variables. It is thus highly redundant, and this redundancy is expressed by the *reproducing kernel*:

$$\begin{aligned} K(\Omega, \tau, \omega_0, t_0) &= \langle g_{\Omega, \tau}, g_{\omega_0, t_0} \rangle \\ &= \int_{t \in \mathbb{R}} p(t - \tau) p(t - t_0) e^{j(\omega_0 - \Omega)t} dt. \end{aligned} \quad (5.14)$$

While this is a four-dimensional object, its magnitude depends only on the two differences  $(\omega_0 - \Omega)$  and  $(t_0 - \tau)$ .<sup>45</sup>

**THEOREM 5.2 (REPRODUCING KERNEL FORMULA FOR THE LOCAL FOURIER TRANSFORM)**

A function  $X(\omega_0, t_0)$  is the local Fourier transform of some function  $x(t)$  if and only if it satisfies

$$X(\Omega, \tau) = \frac{1}{2\pi} \int_{t_1 \in \mathbb{R}} \int_{\omega_0 \in \mathbb{R}} X(\omega_0, t_0) K(\Omega, \tau, \omega_0, t_0) d\omega_0 dt_0. \quad (5.15)$$

*Proof.* If  $X(\omega_0, t_0)$  is a local Fourier transform, then there is a function  $x(t)$  such that

<sup>45</sup>This is expressed in a closely related function called the *ambiguity function*; see Exercise ??.

$X(\omega_0, t_0) = X(\omega_0, t_0)$ , or

$$\begin{aligned} X(\Omega, \tau) &= \int_{t \in \mathbb{R}} x(t) g_{\Omega, \tau}^*(t) dt, \\ &\stackrel{(a)}{=} \int_{t \in \mathbb{R}} \left( \frac{1}{2\pi} \int_{t_0 \in \mathbb{R}} \int_{\omega_0 \in \mathbb{R}} X(\omega_0, t_0) g_{\omega_0, t_0}(t) d\omega_0 dt_0 \right) g_{\Omega, \tau}^*(t) dt \\ &= \frac{1}{2\pi} \int_{t_0 \in \mathbb{R}} \int_{\omega_0 \in \mathbb{R}} X(\omega_0, t_0) \left( \int_{t \in \mathbb{R}} g_{\Omega, \tau}(t) g_{\omega_0, t_0}(t) dt \right) d\omega_0 dt_0, \\ &\stackrel{(b)}{=} \frac{1}{2\pi} \int_{t_0 \in \mathbb{R}} \int_{\omega_0 \in \mathbb{R}} X(\omega_0, t_0) K(\Omega, \tau, \omega_0, t_0) d\omega_0 dt_0, \end{aligned}$$

(5.5b) where (a) follows from the inversion formula (5.5b) and (b) from (5.15).

For the converse, write (5.15) by making  $K(\Omega, \tau, \omega_0, t_0)$  explicit as an integral over  $t$  (see (5.14)):

$$\begin{aligned} X(\Omega, \tau) &= \frac{1}{2\pi} \int_{t_1 \in \mathbb{R}} \int_{\omega_1 \in \mathbb{R}} \int_{t \in \mathbb{R}} X(\omega_0, t_0) g_{\Omega, \tau}(t) g_{\omega, t}(t) dt d\omega_0 dt_0 \\ &\stackrel{(a)}{=} \int_{t \in \mathbb{R}} g_{\Omega, \tau}(t) \left( \frac{1}{2\pi} \int_{t_1 \in \mathbb{R}} \int_{\omega_1 \in \mathbb{R}} X(\omega_0, t_0) g_{\omega, t}(t) d\omega_0 dt_0 \right) dt \\ &\stackrel{(b)}{=} \int_{t \in \mathbb{R}} g_{\Omega, \tau}(t) x(t) dt, \end{aligned}$$

where (a) follows from Fubini's theorem (see *Appendix 2.A.3*) allowing for the exchange of the order of integration, and (b) from the inversion formula (5.5b). Therefore,  $X(\Omega, \tau)$  is indeed a local Fourier transform, namely the local Fourier transform of  $x(t)$ .

The redundancy present in the local Fourier transform allows sampling and interpolation, and the interpolation kernel depends on the reproducing kernel.

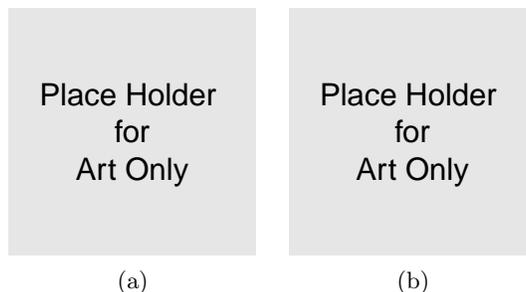
**Characterization of Singularities and Smoothness** To characterize singularities, we will take the view that the local Fourier transform is a Fourier transform of a windowed function  $x_\tau(t)$  as in (5.1). Since this is a product between the function and the prototype function, using the convolution-in-frequency property (4.64), in the Fourier domain this is a convolution. That is, singularities are smoothed by the prototype function.

We now characterize singularities in time and frequency, depicted in Figure 5.3.

- (i) *Characterization of singularities in time:* Take a function perfectly localized in time, the Dirac delta function  $x(t) = \delta(t - t_0)$ . Then

$$X(\Omega, \tau) = \int_{t \in \mathbb{R}} p(t - \tau) \delta(t - t_0) e^{-j\Omega t} dt \stackrel{(a)}{=} p(t_0 - \tau) e^{-j\Omega t_0},$$

where (a) follows from *Table 4.1*. This illustrates the characterization of singularities in time by the local Fourier transform: An event at time location  $t_0$  will spread around  $t_0$  according to the prototype function, and this across all frequencies. If  $p(t)$  has compact support  $[-T/2, T/2]$ , then  $X(\Omega, \tau)$  has support  $[-\infty, \infty] \times [t_0 - T/2, t_0 + T/2]$ .



**Figure 5.3:** Localization properties of the local Fourier transform. (a) A function perfectly localized in time, a Dirac delta function at  $\tau$ , with a compactly supported prototype function  $[-T/2, T/2]$ . (b) A function perfectly localized in frequency, a complex exponential function of frequency  $\omega$ , with a prototype function having a compactly supported Fourier transform  $[-B/2, B/2]$ .

- (ii) *Characterization of singularities in frequency:* Take now a function perfectly localized in frequency, a complex exponential function  $x(t) = e^{j\omega t}$ . Then,

$$\begin{aligned}
 X(\Omega, \tau) &= \int_{t \in \mathbb{R}} p(t - \tau) e^{-j(\Omega - \omega)t} dt \\
 &\stackrel{(a)}{=} e^{-j(\Omega - \omega)\tau} \int_{t' \in \mathbb{R}} p(t') e^{-j(\Omega - \omega)t'} dt' \\
 &\stackrel{(b)}{=} e^{-j(\Omega - \omega)\tau} P(\Omega - \omega),
 \end{aligned} \tag{5.16}$$

where (a) follows from change of variables  $t' = t - \tau$ ; and (b) from the Fourier transform of  $p(t)$ . This illustrates the characterization of singularities in frequency by the local Fourier transform: An event at frequency location  $\omega$  will spread around  $\omega$  according to the prototype function, and this across all time. If  $P(\omega)$  has compact support  $[-B/2, B/2]$ , then  $X(\Omega, \tau)$  has support  $[\omega - B/2, \omega + B/2] \times [-\infty, \infty]$ .

What is important to understand is that if singularities appear together within a prototype function, they appear mixed in the local Fourier transform domain. This is unlike the continuous wavelet transform we will see in the next chapter, where arbitrary time resolution is possible for the scale factor going to 0.

If the prototype function is smoother than the function to be analyzed, then the type of singularity (assuming there is a single one inside the prototype function) is determined by the decay of the Fourier transform.

**EXAMPLE 5.1 (SINGULARITY CHARACTERIZATION OF THE LOCAL FOURIER TRANSFORM)**

Let us consider, as an illustrative example, a triangle prototype function from (4.45):

$$p(t) = \begin{cases} 1 - |t|, & |t| < 1; \\ 0, & \text{otherwise,} \end{cases}$$

which has a Fourier transform (4.47) decaying as  $|\omega|^{-2}$  for large  $\omega$ .

Consider a function  $x(t) \in C^1$  (continuous and with at least one continuous derivative, see Section 2.2.4) except for a discontinuity at  $t = t_0$ . If it were not for the discontinuity, the Fourier transform of  $x(t)$  would decay faster than  $|\omega|^{-2}$  (that is, faster than  $|P(\omega)|$  does). However, because of the singularity at  $t = t_0$ ,  $|X(\omega)|$  decays only as  $|\omega|^{-1}$ .

Now the locality of the local Fourier transform comes into play. There are two modes, given by the regularity of the windowed function  $x_\tau(t)$ : (1) When  $\tau$  is far from  $t_0$ ,  $|\tau - t_0| > 1$ ,  $x_\tau(t)$  is continuous (but its derivative is not, because of the triangle prototype function), and  $|X(\Omega, \tau)|$  decays as  $|\omega|^{-2}$ . (2) When  $\tau$  is close to  $t_0$ ,  $|\tau - t_0| \leq 1$ , that is, it is close to the discontinuity,  $x_\tau(t)$  is discontinuous, and  $|X(\Omega, \tau)|$  decays only as  $|\omega|^{-1}$ .

This above example indicates that there is a subtle interplay between the smoothness and support of the prototype function, and the singularities or smoothness of the analyzed function. This is formalized in the following two results:

**THEOREM 5.3 (SINGULARITY CHARACTERIZATION OF THE LOCAL FOURIER TRANSFORM)**

Assume a prototype function  $p(t)$  with compact support  $[-T/2, T/2]$  and sufficient smoothness. Consider a function  $x(t)$  which is smooth except for a singularity of order  $n$  at  $t = t_0$ , that is, its  $n$ th derivative at  $t_0$  is a Dirac delta function. Then its local Fourier transform decays as

$$|X(\Omega, \tau)| \sim O\left(\frac{1}{1 + |\omega|^n}\right)$$

in the region  $\tau \in [t_0 - T/2, t_0 + T/2]$ .

The proof follows by using the decay property of the Fourier transform applied to the windowed function and is left as Exercise ??.

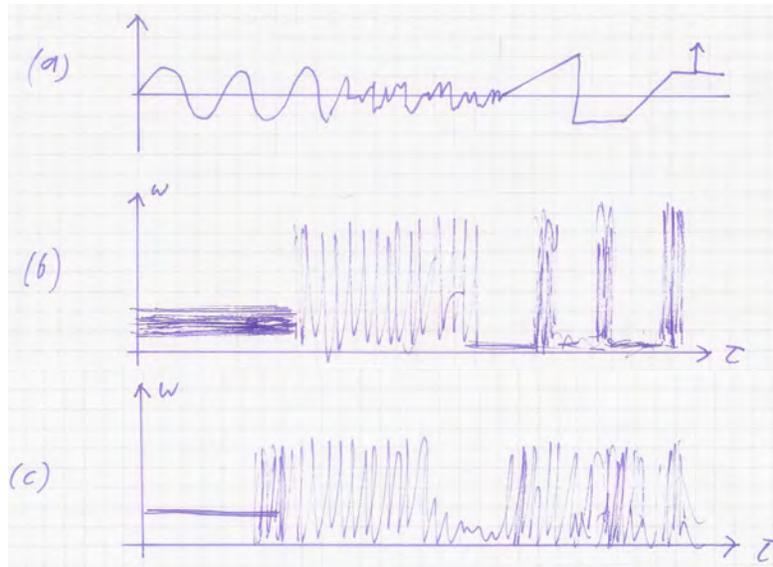
Conversely, a sufficiently decaying local Fourier transform indicates a smooth function in the region of interest.

**THEOREM 5.4 (SMOOTHNESS FROM DECAY OF THE LOCAL FOURIER TRANSFORM)**

Consider a sufficiently smooth prototype function  $p(t)$  of compact support  $[-T/2, T/2]$ . If the local Fourier transform at  $t_0$  decays sufficiently fast, or for some  $\alpha$  and  $\varepsilon > 0$ ,

$$|X(\Omega, \tau)| \leq \frac{\alpha}{1 + |\Omega|^{p+1+\varepsilon}}$$

then  $x(t)$  is  $C^p$  on the interval  $[t_0 - T/2, t_0 + T/2]$ .



**Figure 5.4:** The spectrogram. (a) A signal with various modes. (b) The spectrogram, or  $|X(\Omega, \tau)|$ , with a short prototype function. (c) The spectrogram with a long prototype function.

**Spectrograms** The standard way to display the local Fourier transform is as a density plot of  $|X(\Omega, \tau)|$ . This is called the *spectrogram* and is very popular, for example, for speech and music signals. Figure 5.4 shows a standard signal with various modes and two spectrograms.

As can be seen, the sinusoid is chosen, and the singularities are identified but not exactly localized due to the size of the prototype function. For the short prototype function in (Figure 5.4(b)), various singularities are still isolated, but the sinusoid is not well localized. The reverse is true for the long prototype function (Figure 5.4(c)), where the sinusoid is well identified, but some of the singularities are now mixed together. This is of course the fundamental tension between time and frequency localization, as governed by the uncertainty principle we have seen in *Chapter 7*.

## **5.3 Local Fourier Frame Series**

### **5.3.1 Sampling Grids**

### **5.3.2 Frames from Sampled Local Fourier Transform**

## **5.4 Local Fourier Series**

### **5.4.1 Complex Exponential-Modulated Local Fourier Bases**

#### **Complex-Exponential Modulation**

#### **Balian-Low Theorem**

### **5.4.2 Cosine-Modulated Local Fourier Bases**

#### **Cosine Modulation**

#### **Local Cosine Bases**

## **5.5 Computational Aspects**

### **5.5.1 Complex Exponential-Modulated Local Fourier Bases**

### **5.5.2 Cosine-Modulated Local Fourier Bases**

## **Chapter at a Glance**

TBD

## **Historical Remarks**

TBD

## **Further Reading**

TBD

## Chapter 6

# Wavelet Bases, Frames and Transforms on Functions

### Contents

---

<b>6.1</b>	<b>Introduction</b>	<b>189</b>
<b>6.2</b>	<b>Scaling Function and Wavelets from Orthogonal Filter Banks</b>	<b>208</b>
<b>6.3</b>	<b>Wavelet Series</b>	<b>222</b>
<b>6.4</b>	<b>Wavelet Frame Series</b>	<b>242</b>
<b>6.5</b>	<b>Continuous Wavelet Transform</b>	<b>242</b>
<b>6.6</b>	<b>Computational Aspects</b>	<b>254</b>
	<b>Chapter at a Glance</b>	<b>259</b>
	<b>Historical Remarks</b>	<b>259</b>
	<b>Further Reading</b>	<b>259</b>

---

The previous chapter started with the most redundant version of the local Fourier expansions on functions: the local Fourier transform. We lowered its redundancy through sampling, leading to Fourier frames. Ultimately, we wanted to make it nonredundant by trying to build local Fourier bases; however, we hit a roadblock, the Balian-Low theorem, prohibiting such bases with reasonable joint time and frequency localization. While bases are possible with cosine, instead of complex-exponential, modulation, we can do even better. In this chapter, we will start by constructing wavelet bases, and then go in the direction of increasing redundancy, by building frames and finally the continuous wavelet transform.

## 6.1 Introduction

Iterated filter banks from Chapter 3 pose interesting theoretical and practical questions, the key one quite simple: what happens if we iterate the DWT to infinity? While we need to make the question precise by indicating how this iterative process takes place, when done properly, and under certain conditions on the filters used in the filter bank, the limit leads to a wavelet basis for the space of square-integrable

functions,  $\mathcal{L}^2(\mathbb{R})$ . The key notion is that we take a discrete-time basis (orthonormal or biorthogonal) for  $\ell^2(\mathbb{Z})$ , and derive from it a continuous-time basis for  $\mathcal{L}^2(\mathbb{R})$ . This connection between discrete and continuous time is reminiscent of the concepts and aim of *Chapter 5*, including the sampling theorem. The iterative process itself is fascinating, but the resulting bases are even more so: they are scale invariant (as opposed to shift invariant) so that all basis vectors are obtained from a single function  $\psi(t)$  through shifting and scaling. What we do in this opening section is go through some salient points on a simple example we have seen numerous times in Part II: the Haar basis. We will start from its discrete-time version seen in *Chapter 1* (level 1, scale 0) and the iterated one seen in *Chapter 3* (level  $J$ , scale  $2^J$ ) and build a continuous-time basis for  $\mathcal{L}^2(\mathbb{R})$ . We then mimic this process and show how it can lead to a wealth of different wavelet bases. We will also look into the Haar frame and Haar continuous wavelet transform. We follow the same roadmap from this section, iterated filters—wavelet series—wavelet frame series—continuous wavelet transform, throughout the rest of the chapter, but in a more general setting. As the chapter contains a fair amount of material, some of it quite technical, this section attempts to cover all the main concepts, and is thus rather long. The details in more general settings are covered throughout the rest of the chapter.

### 6.1.1 Scaling Function and Wavelets from Haar Filter Bank

To set the stage, we start with the Haar filters  $g$  and  $h$  given in Table 1.8, *Chapter 1*, where we used their impulse responses and shifts by multiples by two as a basis for  $\ell^2(\mathbb{Z})$ . This orthonormal basis was implemented using a critically-sampled two-channel filter bank with down- and upsampling by 2, synthesis lowpass/highpass filter pair  $g_n, h_n$  from (1.1) (repeated here for easy reference)

$$g_n = \frac{1}{\sqrt{2}}(\delta_n + \delta_{n-1}) \xleftrightarrow{\text{ZT}} G(z) = \frac{1}{\sqrt{2}}(1 + z^{-1}) \quad (6.1a)$$

$$h_n = \frac{1}{\sqrt{2}}(\delta_n - \delta_{n-1}) \xleftrightarrow{\text{ZT}} H(z) = \frac{1}{\sqrt{2}}(1 - z^{-1}), \quad (6.1b)$$

and a corresponding analysis lowpass/highpass filter pair  $g_{-n}, h_{-n}$ .

We then used these filters and the associated two-channel filter bank as a building block for the Haar DWT in *Chapter 3*. For example, we saw that in a 3-level iterated Haar filter bank, the lowpass and highpass at level 3 were given by (3.1c)–(3.1d) and plotted in *Figure 3.4*:

$$\begin{aligned} G^{(3)}(z) &= G(z)G(z^2)G(z^4) \\ &= \frac{1}{2\sqrt{2}}(1 + z^{-1})(1 + z^{-2})(1 + z^{-4}) \\ &= \frac{1}{2\sqrt{2}}(1 + z^{-1} + z^{-2} + z^{-3} + z^{-4} + z^{-5} + z^{-6} + z^{-7}), \end{aligned} \quad (6.2a)$$

$$\begin{aligned} H^{(3)}(z) &= G(z)G(z^2)H(z^4) \\ &= \frac{1}{2\sqrt{2}}(1 + z^{-1})(1 + z^{-2})(1 - z^{-4}) \\ &= \frac{1}{2\sqrt{2}}(1 + z^{-1} + z^{-2} + z^{-3} - z^{-4} - z^{-5} - z^{-6} - z^{-7}). \end{aligned} \quad (6.2b)$$

**Iterated Filters** We now revisit these filters and their iterations, but with a new angle, as we let the iteration go to infinity by associating a continuous-time function to the discrete-time sequence (impulse response of the iterated filter).

We first write the expressions for the equivalent filters at the last level of a  $J$ -level iterated Haar filter bank:

$$\begin{aligned}
 G^{(J)}(z) &= \prod_{\ell=0}^{J-1} G(z^{2^\ell}) = 2^{-J/2} \prod_{\ell=0}^{J-1} (1 + z^{-2^\ell}) \\
 &= 2^{-J/2} \sum_{n=0}^{2^J-1} z^{-n} = G^{(J-1)}(z) \underbrace{\frac{1}{\sqrt{2}}(1 + z^{2^{J-1}})}_{G(z^{2^{J-1}})}, \quad (6.3) \\
 H^{(J)}(z) &= \prod_{\ell=0}^{J-2} G(z^{2^\ell}) H(z^{2^{J-1}}) = 2^{-J/2} \prod_{\ell=0}^{J-2} (1 + z^{-2^\ell})(1 - z^{-2^{J-1}}) \\
 &= 2^{-J/2} \left( \sum_{n=0}^{2^{J-1}-1} z^{-n} - \sum_{n=2^{J-1}}^{2^J-1} z^{-n} \right) = G^{(J-1)}(z) \underbrace{\frac{1}{\sqrt{2}}(1 - z^{2^{J-1}})}_{H(z^{2^{J-1}})}.
 \end{aligned}$$

We have seen the above expressions in (3.5c), (3.9) already; they construct the equivalent filter at the subsequent level, from the equivalent filters at the previous one.

We know that, by construction, these filters are orthonormal to their shifts by  $2^J$ , (3.6a), (3.11a), as well as orthogonal to each other, (3.14a), and their lengths are

$$L^{(J)} = 2^J. \quad (6.4)$$

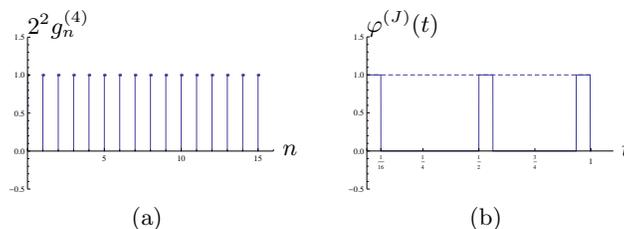
**Scaling Function and its Properties** We now associate a piecewise-constant function  $\varphi^{(J)}(t)$  to  $g_n^{(J)}$  so that  $\varphi^{(J)}(t)$  is of finite length and norm 1; we thus have to determine the width and height of the piecewise segments. Since the number of piecewise segments (equal to the number of nonzero coefficients of  $g_n^{(J)}$ ) grows exponentially with  $J$  because of (6.4), we choose their width as  $2^{-J}$ , upper bounding the length of  $\varphi^{(J)}(t)$  by 1. For  $\varphi^{(J)}(t)$  to inherit the unit-norm property from  $g_n^{(J)}$ , we choose the height of the piecewise segments as  $2^{J/2} g_n^{(J)}$ . Then, the  $n$ th piece of  $\varphi^{(J)}(t)$  contributes

$$\int_{n/2^J}^{(n+1)/2^J} |\varphi^{(J)}(t)|^2 dt = \int_{n/2^J}^{(n+1)/2^J} 2^J (g_n^{(J)})^2 dt = (g_n^{(J)})^2 \stackrel{(a)}{=} 2^{-J}$$

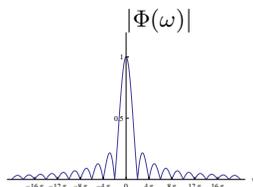
to  $\varphi^{(J)}(t)$  (where (a) follows from (6.3)). Summing up the individual contributions,

$$\|\varphi^{(J)}(t)\|^2 = \sum_{n=0}^{2^J-1} \int_{n/2^J}^{(n+1)/2^J} |\varphi^{(J)}(t)|^2 dt = \sum_{n=0}^{2^J-1} 2^{-J} = 1$$

as in Figure 6.1. We have thus defined our piecewise-constant function as



**Figure 6.1:** Example construction of a piecewise-constant function  $\varphi^{(J)}(t)$  from  $g_n^{(J)}$ . (a) Discrete-time sequence  $2^2 g_n^{(4)}$ . (b) Continuous-time piecewise-constant function  $\varphi^{(4)}(t)$  (we plot a few isolated piecewise segments for emphasis).



**Figure 6.2:** Magnitude response  $\Phi(\omega)$  of the scaling function  $\varphi(t)$ .

$$\varphi^{(J)}(t) = 2^{J/2} g_n^{(J)} = 1 \quad \frac{n}{2^J} \leq t < \frac{n+1}{2^J}. \quad (6.5)$$

As  $\varphi^{(J)}(t)$  is 1 on every interval of length  $2^{-J}$  and  $g^{(J)}$  has exactly  $2^J$  nonzero entries (see (6.4)), this function is actually 1 on the interval  $[0, 1)$  for every  $J$ , that is, the limit of  $\varphi^{(J)}(t)$  is the indicator function of the unit interval  $[0, 1]$  (or, a box function shifted to  $1/2$ ),  $\varphi(t)$ , independently of  $J$ ,

$$\varphi^{(J)}(t) = \begin{cases} 1, & 0 \leq t < 1; \\ 0, & \text{otherwise,} \end{cases} = \varphi(t). \quad (6.6)$$

Convergence is achieved without any problem, actually in one step!<sup>46</sup>

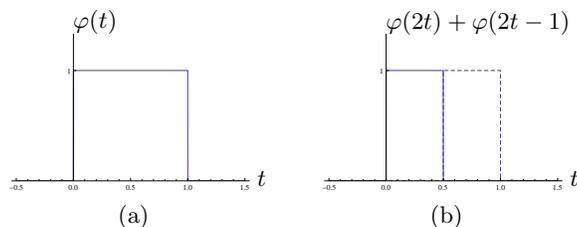
The function  $\varphi(t)$  is called the *Haar scaling function*. Had we started with a different lowpass filter  $g$ , the resulting limit, had it existed, would have lead to a different scaling function, a topic we will address later in the chapter.

In the Fourier domain,  $\Phi^{(J)}(\omega)$ , the Fourier transform of  $\varphi^{(J)}(t)$ , will be the same for every  $J$  because of (6.6), and thus, the Fourier transform of the scaling function will be the sinc function in frequency (see *Table 4.6* and Figure 6.2):

$$\Phi(\omega) = e^{-j\omega/2} \frac{\sin(\omega/2)}{\omega/2} = e^{-j\omega/2} \text{sinc}(\omega/2). \quad (6.7)$$

We now turn our attention to some interesting properties of the scaling function:

<sup>46</sup>Although we could have defined a piecewise-linear function instead of a piecewise-constant one, we chose not to do so as the behavior of the limit we will study does not change.



**Figure 6.3:** Two-scale equation for the Haar scaling function. (a) The scaling function  $\varphi(t)$  and (b) expressed as a linear combination of  $\varphi(2t)$  and  $\varphi(2t - 1)$ .

- (i) *Two-scale equation:* The Haar scaling function  $\varphi(t)$  satisfies

$$\varphi(t) = \sqrt{2}(g_0 \varphi(2t) + g_1 \varphi(2t - 1)) = \varphi(2t) + \varphi(2t - 1), \quad (6.8)$$

the so-called two-scale equation. We see that the scaling function is built out of two scaled versions of itself, illustrated in Figure 6.3. While in this Haar case, this does not come as a big surprise, it will when the scaling functions become more complex. To find the expression for the two-scale equation in the Fourier domain, we rewrite (6.8) as a convolution

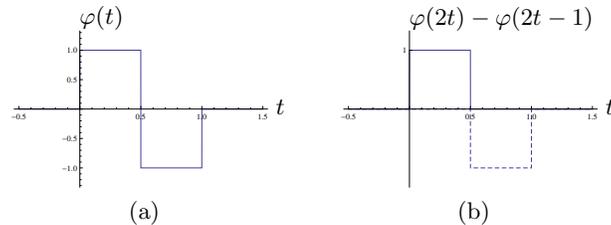
$$\varphi(t) = \varphi(2t) + \varphi(2t - 1) = \sqrt{2} \sum_{k=0}^1 g_k \varphi(2t - k). \quad (6.9)$$

We can then use the convolution-in-time property (4.61) and the scaling property (4.55a) of the Fourier transform to get

$$\Phi(\omega) = \frac{1}{\sqrt{2}} G(e^{j\omega/2}) \Phi(\omega/2) = \frac{1}{2}(1 + e^{-j\omega/2}) e^{-j\omega/4} \text{sinc}(\omega/4). \quad (6.10)$$

- (ii) *Smoothness:* The Haar scaling function  $\varphi(t)$  is not continuous.<sup>47</sup> This can also be seen from the decay of its Fourier transform  $\Phi(\omega)$ , which, as we know from (6.7), is a sinc function (see also Figure 6.2), and thus decays slowly (it has, however, only two points of discontinuity and is, therefore, not all-together ill-behaved).
- (iii) *Reproduction of polynomials:* The Haar scaling function  $\varphi(t)$  with its integer shifts can reproduce constant functions. This stems from the polynomial approximation properties of  $g$ , as in Theorem 1.5. In the next section, we will see how other scaling functions will be able to reproduce polynomials of degree  $N$ . The key will be the number of zeros at  $\omega = \pi$  of the lowpass filter  $G(e^{j\omega})$ ; from (6.1a), for the Haar scaling function, there is just 1.
- (iv) *Orthogonality to integer shifts:* The Haar scaling function  $\varphi(t)$  is orthogonal to its integer shifts, another property inherited from the underlying filter. Of

<sup>47</sup>In Theorem 6.1, we will see a sufficient condition for the limit function, if it exists, to be continuous (and possibly  $k$ -times differentiable).



**Figure 6.4:** Two-scale equation for the Haar wavelet. (a) The wavelet  $\psi(t)$  and (b) expressed as a linear combination of  $\varphi(2t)$  and  $\varphi(2t - 1)$ .

course, in this Haar case the property is obvious, as the support of  $\varphi(t)$  is limited to the unit interval. The property will still hold for more general scaling functions, albeit it will not be that obvious to see.

**Wavelet and its Properties** The scaling function we have just seen is lowpass in nature (if the underlying filter  $g$  is lowpass in nature). Similarly, we can construct a *wavelet* (or, simply *wavelet*) that will be bandpass in nature (if the underlying filter  $h$  is highpass in nature).

We thus associate a piecewise-constant function  $\psi^{(J)}(t)$  to  $h_n^{(J)}$  in such a way that  $\psi^{(J)}(t)$  is of finite length and of norm 1; we use the same arguments as before to determine the width and height of the piecewise segments, leading to

$$\psi^{(J)}(t) = 2^{J/2} h_n^{(J)} \quad \frac{n}{2^J} \leq t < \frac{n+1}{2^J}. \quad (6.11)$$

Like  $\varphi^{(J)}(t)$ , the function  $\psi^{(J)}(t)$  is again the same for every  $J$  since the length of  $h^{(J)}$  is exactly  $2^J$ . It is 1 for  $n = 0, 1, \dots, 2^{J-1} - 1$ , and is  $-1$  for  $n = 2^{J-1}, 2^{J-1} + 1, \dots, 2^J - 1$ . Thus, it comes as no surprise that the limit is

$$\psi(t) = \begin{cases} 1, & 0 \leq t < 1/2; \\ -1, & 1/2 \leq t < 1; \\ 0, & \text{otherwise,} \end{cases} \quad (6.12)$$

called the *Haar wavelet*, or, *Haar wavelet*,<sup>48</sup> (see Figure 6.4(a)).

Similarly to  $\Phi(\omega)$ , in the Fourier-domain,

$$\Psi(\omega) = \frac{1}{2}(1 - e^{-j\omega/2}) e^{-j\omega/4} \operatorname{sinc}(\omega/4) = \frac{1}{\sqrt{2}} H(e^{j\omega/2}) \Phi(\omega/2). \quad (6.13)$$

We now turn our attention to some interesting properties of the Haar wavelet:

- (i) *Two-scale equation:* We can see from  $\Psi(\omega)$  its highpass nature as it is 0 at  $\omega = 0$  (because  $H(1) = 0$ ). Using the convolution-in-time property (4.61) and

<sup>48</sup>Depending on the initial discrete-time filters, the resulting limits, when they exist, lead to different wavelets.

the scaling property (4.55a) of the Fourier transform, we see that the above can be written in the time domain as (see Figure 6.4(b))

$$\psi(t) = \sqrt{2} \langle h_k, \varphi(2t - k) \rangle_k = \varphi(2t) - \varphi(2t - 1). \quad (6.14)$$

In other words, the wavelet is built out of the scaling function at a different scale and its shift, its own two-scale equation, but involving scaled versions of the scaling function instead of itself. The last expression in (6.13) is the Fourier-domain version of the two-scale equation.

- (ii) *Smoothness*: Since the wavelet is a linear combination of the scaled scaling function and its shift, its smoothness is inherited from the scaling function; in other words, like the scaling function, it is not continuous, having 3 points of discontinuity.
- (iii) *Zero-moment property*: We have seen that the Haar scaling function can reproduce constant functions. The Haar wavelet has a complementary property, called zero-moment property. To see that,

$$\Phi(\omega)|_{\omega=0} = \int_{-\infty}^{\infty} \psi(t) dt = 0,$$

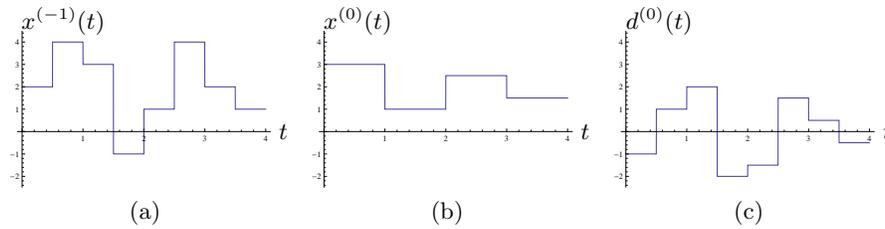
that is, the inner product between the wavelet and a constant function will be zero. In other words, the wavelet annihilates constant functions while the scaling function reproduces them.

- (iv) *Orthogonality to integer shifts*: Finally, like the scaling function, the wavelet is orthogonal with respect to integer shifts. Again, this is trivial to see for the Haar wavelet as it is supported on the unit interval only.
- (v) *Orthogonality of the scaling and wavelets*: It is also trivial to see that the scaling function and the wavelet are orthogonal to each other. All these properties are setting the stage for us to build a basis based on these functions.

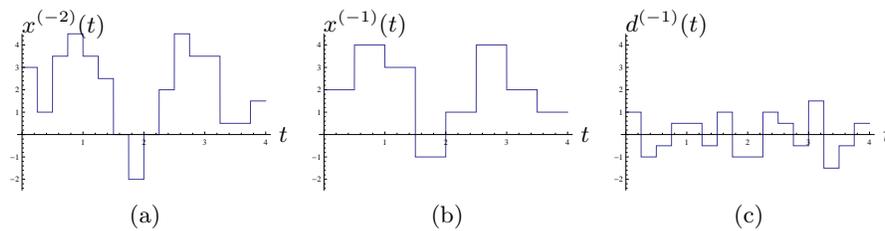
### 6.1.2 Haar Wavelet Series

Thus far, we have constructed two functions, the scaling function  $\varphi(t)$  and the wavelet  $\psi(t)$ , by iterating the Haar filter bank. That filter bank implements a discrete-time Haar basis, what about in continuous time? What we can say is that this scaling function and the wavelet, together with their integer shifts,  $\{\varphi(t - k), \psi(t - k)\}_{k \in \mathbb{Z}}$  do constitute a basis, for the space of piecewise-constant functions on intervals of half-integer length or more (see Figure 6.5(a)–(c)). We can see that as follows. Assume we are given a function  $x^{(-1)}(t)$  that equals  $a$  for  $0 \leq t < 1/2$ ;  $b$  for  $1/2 \leq t < 1$ ; and 0 otherwise. Then,

$$\begin{aligned} x^{(-1)}(t) &= \frac{a+b}{2} \varphi(t) + \frac{a-b}{2} \psi(t) \\ &= \langle x^{(-1)}(t), \varphi(t) \rangle \varphi(t) + \langle x^{(-1)}(t), \psi(t) \rangle \psi(t) \\ &= x^{(0)}(t) + d^{(0)}(t). \end{aligned} \quad (6.15)$$



**Figure 6.5:** Haar series decomposition of (a)  $x^{(-1)}(t)$ , a function constant on half-integer intervals, using  $\{\varphi(t-k), \psi(t-k)\}_{k \in \mathbb{Z}}$ , into (b)  $x^{(0)}(t)$  and (c)  $d^{(0)}(t)$ .



**Figure 6.6:** Haar series decomposition of (a)  $x^{(-2)}(t)$ , a function constant on quarter-integer intervals, using  $\{\sqrt{2}\varphi(2t-k), \sqrt{2}\psi(2t-k)\}_{k \in \mathbb{Z}}$ , into (b)  $x^{(-1)}(t)$  and (c)  $d^{(-1)}(t)$ .

Had the function  $x^{(-1)}(t)$  been nonzero on any other interval, we could have used the integer shifts of  $\varphi(t)$  and  $\psi(t)$ .

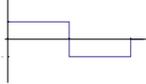
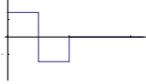
Clearly, this process scales by 2. In other words, the scaled scaling function and the wavelet, together with their shifts by multiples of  $1/2$ ,  $\{\sqrt{2}\varphi(2t-k), \sqrt{2}\psi(2t-k)\}_{k \in \mathbb{Z}}$  do constitute a basis, for the space of piecewise-constant functions on intervals of quarter-integer length or more (see Figure 6.6(a)–(c)). Assume, for example, we are now given a function  $x^{(-2)}(t)$  that equals  $c$  for  $0 \leq t < 1/4$ ;  $d$  for  $1/4 \leq t < 1/2$ ; and 0 otherwise. Then,

$$\begin{aligned} x^{(-1)}(t) &= \frac{c+d}{2} \varphi(2t) + \frac{c-d}{2} \psi(2t) \\ &= \langle x^{(-2)}, \sqrt{2}\varphi(2t) \rangle \sqrt{2}\varphi(2t) + \langle x^{(-2)}, \sqrt{2}\psi(2t) \rangle \sqrt{2}\psi(2t) \\ &= x^{(-1)}(t) + d^{(-1)}(t) \stackrel{(a)}{=} x^{(0)}(t) + d^{(0)}(t) + d^{(-1)}(t), \end{aligned}$$

where (a) follows from (6.15). In other words, we could have also decomposed  $x^{(-2)}(t)$  using  $\{\varphi(t-k), \psi(t-k)\}_{k \in \mathbb{Z}}$ .

Continuing this argument, to represent piecewise-constant functions on inter-

vals of length  $2^{-\ell}$ , we need the following basis:

Scale	Functions		
0	scaling function and its shifts	$\varphi(t - k)$	
	wavelet and its shifts	$\psi(t - k)$	
-1	wavelet and its shifts	$2^{1/2}\psi(2t - k)$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$-(\ell - 1)$	wavelet and its shifts	$2^{-(\ell-1)/2}\psi(2^{-(\ell-1)}t - k)$	

**Definition of the Haar Wavelet Series** From what we have seen, if we want to represent shorter and shorter constant pieces, we need to keep on adding wavelets with decreasing scale together with the scaling function at the coarsest scale. We may imagine, and we will formalize this in a moment, that if we let this process go to infinity, the scaling function will eventually become superfluous, and it does. This previous discussion leads to the Haar orthonormal set and a truly surprising result dating back to Haar in 1910, that this orthonormal system is in fact a basis for  $\mathcal{L}^2(\mathbb{R})$ . This result is the Haar continuous-time counterpart of Theorem 3.2, which states that the discrete-time wavelet  $h_n$  and its shifts and scales (equivalent iterated filters) form an orthonormal basis for the space of finite-energy sequences,  $\ell^2(\mathbb{Z})$ . The general result will be given by Theorem 6.6.

For compactness, we start by renaming our basis functions as:

$$\psi_{\ell,k}(t) = 2^{-\ell/2} \psi(2^{-\ell}t - k) = \frac{1}{2^{\ell/2}} \psi\left(\frac{t - 2^\ell k}{2^\ell}\right), \quad (6.16a)$$

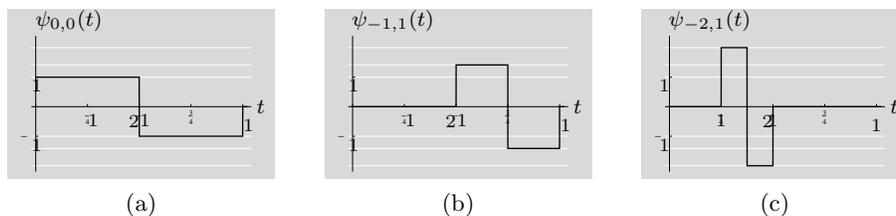
$$\varphi_{\ell,k}(t) = 2^{-\ell/2} \varphi(2^{-\ell}t - k) = \frac{1}{2^{\ell/2}} \varphi\left(\frac{t - 2^\ell k}{2^\ell}\right). \quad (6.16b)$$

A few of the wavelets are given in Figure 6.7. Since we will show that the Haar wavelets form an orthonormal basis, we can define the Haar wavelet series to be

$$\beta_k^{(\ell)} = \langle x, \psi_{\ell,k} \rangle = \int_{-\infty}^{\infty} x(t) \psi_{\ell,k}(t) dt, \quad \ell, k \in \mathbb{Z}, \quad (6.17a)$$

and the inverse Haar wavelet series

$$x(t) = \sum_{\ell \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} \beta_k^{(\ell)} \psi_{\ell,k}(t). \quad (6.17b)$$



**Figure 6.7:** Example Haar basis functions. (a) The prototype function  $\psi(t) = \psi_{0,0}(t)$ ; (b)  $\psi_{-1,1}(t)$ ; (c)  $\psi_{-2,1}(t)$ . (Repeated Figure 1.2.)

We call  $\beta^{(\ell)}$  the *wavelet coefficients*, and denote such a wavelet series pair by

$$x(t) \xleftrightarrow{\text{WS}} \beta_k^{(\ell)}.$$

### Properties of the Haar Wavelet Series

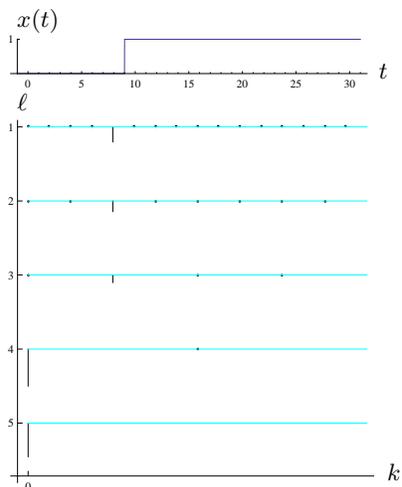
- (i) *Linearity:* The Haar wavelet series operator is a linear operator.
- (ii) *Parseval's Equality:* The Haar wavelet series operator is a unitary operator and thus preserves the Euclidean norm (see (2.53)):

$$\|x\|^2 = \int_{-\infty}^{\infty} |x(t)|^2 dt = \sum_{\ell \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} |\beta_k^{(\ell)}|^2. \quad (6.18)$$

- (iii) *Zero-Moment Property:* We have seen earlier that, while the Haar scaling function with its integer shifts can reproduce constant functions, the Haar wavelet with its integer shifts annihilates them. As the wavelet series uses wavelets as its basis functions, it inherits that property; it annihilates constant functions. In the remainder of the chapter, we will see this to be true for higher-degree polynomials with different wavelets.
- (iv) *Characterization of Singularities:* One of the powerful properties of wavelet-like representations is that they can characterize the type and position of singularities via the behavior of wavelet coefficients. Assume, for example, that we want to characterize the step singularity present in the Heaviside function (4.7) with the step at location  $t_0$ . We compute the wavelet coefficient  $\beta_k^{(\ell)}$  to get

$$\begin{aligned} \beta_k^{(\ell)} &= \int_{-\infty}^{\infty} x(t) \psi_{\ell,k}(t) dt \\ &= \begin{cases} 2^{\ell/2}k - 2^{-\ell/2}t_0, & 2^\ell k \leq t_0 < 2^\ell(k + \frac{1}{2}); \\ -2^{\ell/2}(k + 1) + 2^{-\ell/2}t_0, & 2^\ell(k + \frac{1}{2}) \leq t_0 < 2^\ell(k + 1). \end{cases} \end{aligned}$$

Because the Haar wavelets at a fixed scale do not overlap, there exists exactly one nonzero wavelet coefficient per scale, the one that straddles the discontinuity. Therefore, as  $\ell \rightarrow -\infty$ , the wavelet series zooms towards the singularity, shown in Figure 6.8. We see that as  $\ell$  decreases from 5 to 1, the single nonzero wavelet coefficients gets closer and closer to the discontinuity.



**Figure 6.8:** Behavior of Haar wavelet coefficients across scales. We plot  $\beta_k^{(\ell)}$  for  $\ell = 1, 2, \dots, 5$ , where  $k$  is dependent on the scale. Because the wavelet is Haar, there is exactly one nonzero coefficient per scale, the one corresponding to the wavelet that straddles the discontinuity.

**Multiresolution Analysis** In the discrete-time Chapters 1-4, we have often encountered coarse/detail approximating spaces  $V$  and  $W$ . We now use the same intuition and start from similar spaces to build the Haar wavelet series in reverse. What we will see is how the iterative construction and the two-scale equations are the manifestations of a fundamental embedding property explicit in the multiresolution analysis of Mallat and Meyer.

We call  $V^{(0)}$  the space of piecewise-constant functions over unit intervals, that is, we say that  $x(t) \in V^{(0)}$ , if and only if  $x(t)$  is constant for  $t \in [k, k+1)$ , and  $x(t)$  is of finite  $\mathcal{L}^2$  norm. Another way to phrase the above is to note that

$$V^{(0)} = \text{span}(\{\varphi(t-k)\}_{k \in \mathbb{Z}}) = \text{span}(\{\varphi_{0,k}\}_{k \in \mathbb{Z}}), \quad (6.19)$$

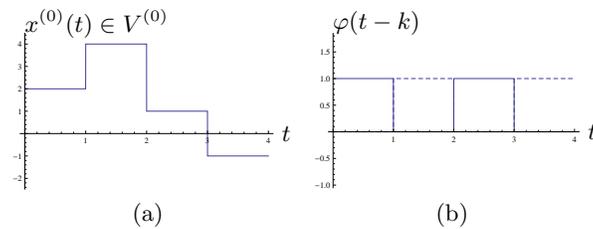
where  $\varphi(t)$  is the Haar scaling function (6.6), and, since  $\langle \varphi(t-k), \varphi(t-m) \rangle = \delta_{k-m}$ , this scaling function and its integer translates form an orthonormal basis for  $V^{(0)}$  (see Figure 6.9). Thus,  $x(t)$  from  $V^{(0)}$  can be written as a linear combination

$$x(t) = \sum_{k \in \mathbb{Z}} \alpha_k^{(0)} \varphi(t-k),$$

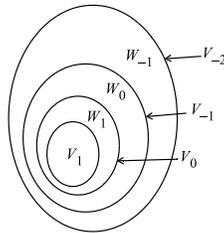
where  $\alpha_k^{(0)}$  is simply the value of  $x(t)$  on the interval  $[k, k+1)$ . Since  $\|\varphi(t)\| = 1$ ,

$$\|x(t)\|^2 = \|\alpha_k^{(0)}\|^2,$$

or, Parseval's equality for this orthonormal basis. We now introduce a scaled



**Figure 6.9:** Haar multiresolution spaces and basis functions. (a) A function  $x^{(0)}(t) \in V^{(0)}$ . (b) Basis functions for  $V^{(0)}$ .



**Figure 6.10:** Multiresolution spaces.

version of  $V^{(0)}$  called  $V^{(\ell)}$ , the space of piecewise-constant functions over intervals of size  $2^\ell$ , that is  $[2^\ell k, 2^\ell(k+1))$ ,  $\ell \in \mathbb{Z}$ . Then,

$$V^{(\ell)} = \text{span}(\{\varphi_{\ell,k}\}_{k \in \mathbb{Z}}),$$

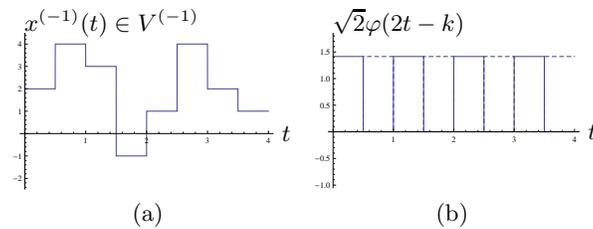
for  $\ell \in \mathbb{Z}$ . For  $\ell > 0$ ,  $V^{(\ell)}$  is a stretched version of  $V^{(0)}$ , and for  $\ell < 0$ ,  $V^{(\ell)}$  is a compressed version of  $V^{(0)}$  (both by  $2^\ell$ ). Moreover, the fact that functions constant over  $[2^\ell k, 2^\ell(k+1))$  are also constant over  $[2^m k, 2^m(k+1))$ ,  $\ell > m$ , leads to the *inclusion property* (see Figure 6.10),

$$V^{(\ell)} \subset V^{(m)} \quad \ell > m. \quad (6.20)$$

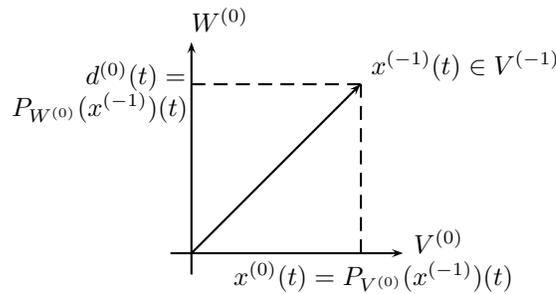
We can use this to derive the two-scale equation (6.8), by noting that because of  $V^{(0)} \subset V^{(-1)}$ ,  $\varphi(t)$  can be expanded in the basis for  $V^{(-1)}$ . Graphically, we show the spaces  $V^{(0)}$ ,  $V^{(-1)}$ , and their basis functions in Figures 6.9 and Figure 6.11; the two-scale equation was shown in Figure 6.3.

What about the detail spaces? Take a function  $x^{(-1)}(t)$  in  $V^{(-1)}$  but not in  $V^{(0)}$ ; such a function is constant over half-integer intervals but not so over integer intervals (see Figure 6.11(a)). Decompose it as a sum of its projections onto  $V^{(0)}$  and  $W^{(0)}$ , the latter the orthogonal complement of  $V^{(0)}$  in  $V^{(-1)}$  (see Figure 6.12),

$$x^{(-1)}(t) = P_{V^{(0)}}(x^{(-1)})(t) + P_{W^{(0)}}(x^{(-1)})(t). \quad (6.21)$$



**Figure 6.11:** Haar multiresolution spaces and basis functions. (a) A function  $x^{(-1)}(t) \in V^{(-1)}$ . (b) Basis functions for  $V^{(-1)}$ .



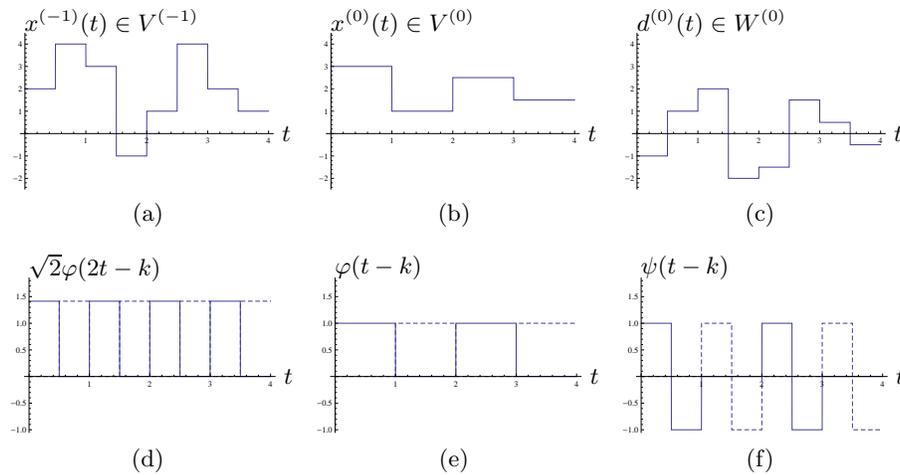
**Figure 6.12:** A function from  $V^{(1)}$  as the sum of its projections onto  $V^{(0)}$  and  $W^{(0)}$ .

We first find the projection of  $x^{(-1)}(t)$  onto  $V^{(0)}$  as

$$\begin{aligned}
 P_{V^{(0)}}(x^{(-1)})(t) &= x^{(0)}(t), \\
 &= \sum_{k \in \mathbb{Z}} \alpha_k^{(0)} \varphi(t - k) = \sum_{k \in \mathbb{Z}} \langle x^{(-1)}(t), \varphi(t - k) \rangle_t \varphi(t - k), \\
 &\stackrel{(a)}{=} \sum_{k \in \mathbb{Z}} \langle x^{(-1)}(t), \varphi(2t - 2k) + \varphi(2t - 2k - 1) \rangle_t \varphi(t - k), \\
 &\stackrel{(b)}{=} \sum_{k \in \mathbb{Z}} \frac{1}{2} \left[ x^{(-1)}(k) + x^{(-1)}(k + \frac{1}{2}) \right] \varphi(t - k), \tag{6.22}
 \end{aligned}$$

where (a) follows from the two-scale equation (6.8); and (b) from evaluating the inner product between  $x^{(-1)}(t)$  and the basis functions for  $V^{(-1)}$ . In other words,  $x^{(0)}(t)$  is simply the average of  $x^{(-1)}(t)$  over two successive intervals. This is the best least-squares approximation of  $x^{(-1)}(t)$  by a function in  $V^{(0)}$  (see Exercise ??).

We now find the projection of  $x^{(-1)}$  onto  $W^{(0)}$ . Subtract the projection  $x^{(0)}$  from  $x^{(-1)}$  and call the difference  $d^{(0)}$ . Since  $x^{(0)}$  is an orthogonal projection,  $d^{(0)}$



**Figure 6.13:** Haar decomposition of a function (a)  $x^{(-1)}(t) \in V^{(-1)}$  into a projection (b)  $x^{(0)}(t) \in V^{(0)}$  (average over two successive intervals) and (c)  $d^{(0)}(t) \in W^{(0)}$  (difference over two successive intervals). (d)–(f) Appropriate basis functions.

is orthogonal to  $V^{(0)}$  (see Figure 6.12). Using (6.22) leads to

$$\begin{aligned}
 P_{W^{(0)}}(x^{(-1)})(t) &= d^{(0)}(t) = x^{(-1)}(t) - x^{(0)}(t), \\
 &= \begin{cases} \frac{1}{2} [x^{(-1)}(k) - x^{(-1)}(k + \frac{1}{2})], & k \leq t < k + \frac{1}{2}; \\ -\frac{1}{2} [x^{(-1)}(k) - x^{(-1)}(k + \frac{1}{2})], & k + \frac{1}{2} \leq t < k + 1, \end{cases} \\
 &= \sum_k \frac{1}{2} [x^{(-1)}(k) - x^{(-1)}(k + \frac{1}{2})] \psi(t - k) \\
 &= \sum_k \frac{1}{\sqrt{2}} [\beta_{2k}^{(-1)} - \beta_{2k+1}^{(-1)}] \psi(t - k) = \sum_k \beta_k^{(0)} \psi(t - k). \quad (6.23)
 \end{aligned}$$

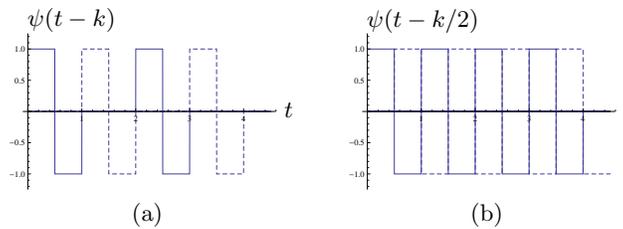
We have thus informally shown that the space  $V^{(-1)}$  can be decomposed as

$$V^{(-1)} = V^{(0)} \oplus W^{(0)} \quad (6.24)$$

(see an example in Figure 6.13). We also derived bases for these spaces, scaling functions and their shifts for  $V^{(-1)}$  and  $V^{(0)}$ , and wavelets and their shifts for  $W^{(0)}$ . This process can be further iterated on  $V_0$  (see Figure 6.10).

### 6.1.3 Haar Frame Series

The Haar wavelet series we just saw is an elegant representation, and completely nonredundant. As we have seen in Chapters 4 and 5, at times we can benefit from relaxing this constraint, and allowing some redundancy in the system. Our aim would then be to build frames. There are many ways in which we could do that. For example, by adding to the Haar wavelet basis wavelets at points halfway in



**Figure 6.14:** A few Haar wavelets at scale  $\ell = 0$  for the (a) Haar wavelet series (nonredundant) and (b) Haar frame series (redundant). Clearly, there are twice as many wavelets in (b), making it a redundant expansion with the redundancy factor 2.

between the existing ones, we would have twice as many wavelets, leading to a redundant series representation with a redundancy factor of 2, a simple example we will use to illustrate Haar frame series.

**Definition of the Haar Frame Series** We now relax the constraint of critical sampling, but still retain the series expansion. That is, we assume that expansion coefficients are

$$\beta_k^{(\ell)} = \langle x, \psi_{\ell,k} \rangle = \int_{-\infty}^{\infty} x(t) \psi_{\ell,k}(t) dt, \quad \ell, k \in \mathbb{Z}, \quad (6.25)$$

where  $\psi_{\ell,k}(t)$  is now given by

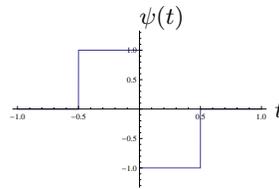
$$\psi_{\ell,k}(t) = a_0^{-\ell/2} \psi(a_0^{-\ell} t - b_0 k) = a_0^{-\ell/2} \psi\left(\frac{t - a_0^{\ell} b_0 k}{a_0^{\ell}}\right), \quad (6.26)$$

with  $a_0 > 1$  and  $b_0 > 0$ . With  $a_0 = 2$  and  $b_0 = 1$ , we get back our nonredundant Haar wavelet series. What we have allowed ourselves to do here is to choose different scale factors from 2, as well as different coverage of the time axis by shifted wavelets at a fixed scale. For example, keep the scale factor the same,  $a_0 = 2$ , but allow overlap of half width between Haar wavelets, that is, choose  $b_0 = 1/2$ . Figure 6.14(b) shows how many wavelets then populate the time axis at a fixed scale (example for  $\ell = 0$ ), compared to the wavelet series (part (a) of the same figure).

**Properties of the Haar Frame Series** Such a Haar frame series satisfies similar properties as the Haar wavelet series: it is linear, it is able to characterize singularities, it inherits the zero-moment property. One property though, Parseval's equality, bears further scrutiny. Let us express the energy of the expansion coefficients as:

$$\sum_{\ell \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} |\beta_{k/2}^{(\ell)}|^2 = \sum_{\ell \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} |\beta_k^{(\ell)}|^2 + \sum_{\ell \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} |\beta_{k+1/2}^{(\ell)}|^2 \stackrel{(a)}{=} 2 \|x\|^2,$$

where (a) follows from (6.18). In other words, this frame series behaves like two orthonormal bases glued together; it is then not surprising that the energy in the expansion coefficients is twice that of the input function, making this transform a tight frame.



**Figure 6.15:** The prototype wavelet in the Haar continuous wavelet transform.

### 6.1.4 Haar Continuous Wavelet Transform

We finally relax all the constraints and discuss the most redundant version of a wavelet expansion, where the Haar wavelet (6.12) is now shifted to be centered at  $t = 0$  (see Figure 6.15):

$$\psi(t) = \begin{cases} 1, & -1/2 \leq t < 0; \\ -1, & 0 \leq t < 1/2; \\ 0, & \text{otherwise.} \end{cases} \quad (6.27)$$

Then, instead of  $a = a_0^\ell$ , we allow all positive real numbers,  $a \in \mathbb{R}^+$ . Similarly, instead of shifts  $b = b_0 k$ , we allow all real numbers,  $b \in \mathbb{R}$ :

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right), \quad a \in \mathbb{R}^+, \quad b \in \mathbb{R}, \quad (6.28)$$

with  $\psi(t)$  the Haar wavelet. The scaled and shifted Haar wavelet is then centered at  $t = b$  and scaled by a factor  $a$ . All the wavelets are again of unit norm,  $\|\psi_{a,b}(t)\| = 1$ . For  $a = 2^\ell$  and  $b = 2^\ell k$ , we get the nonredundant wavelet basis as in TBD.

**Definition of the Haar Continuous Wavelet Transform** We then define the Haar continuous wavelet transform to be (an example is given in Figure 6.16):

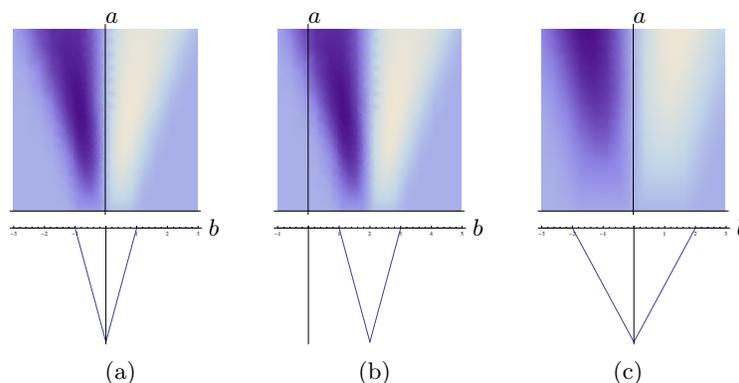
$$X(a, b) = \langle x, \psi_{a,b} \rangle = \int_{-\infty}^{\infty} x(t) \psi_{a,b}(t) dt, \quad a \in \mathbb{R}^+, \quad b \in \mathbb{R}, \quad (6.29a)$$

with  $\psi_{a,b}(t)$  from (6.28), with the inverse Haar continuous wavelet transform

$$x(t) = \frac{1}{C_\psi} \int_{a \in \mathbb{R}^+} \int_{b \in \mathbb{R}} X(a, b) \psi_{a,b}(t) \frac{db da}{a^2}, \quad (6.29b)$$

where the equality holds in  $\mathcal{L}^2$  sense. To denote such a pair, we write:

$$x(t) \xleftrightarrow{\text{CWT}} X(a, b).$$



**Figure 6.16:** (a) The Haar wavelet transform of an example function  $x(t)$  (hat function). (b) Illustration of the shift-in-time property for  $x(t - 2)$ . (c) Illustration of the scaling-in-time property for  $x(t/2)$ .

### Properties of the Haar Continuous Wavelet Transform

- (i) *Linearity:* The Haar continuous wavelet transform operator is a linear operator.
- (ii) *Shift in time:* A shift in time by  $t_0$  results in (see Figure 6.16(b))

$$x(t - t_0) \xleftrightarrow{\text{CWT}} X(a, b - t_0). \quad (6.30)$$

- (iii) *Scaling in time:* Scaling in time by  $\alpha$  results in (see Figure 6.16(c))

$$x(\alpha t) \xleftrightarrow{\text{CWT}} \frac{1}{\sqrt{\alpha}} X\left(\frac{a}{\alpha}, \frac{b}{\alpha}\right). \quad (6.31)$$

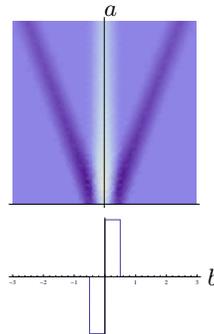
- (iv) *Parseval's equality:* Parseval's equality holds for the Haar continuous wavelet transform; we omit it here and revisit it in the general context in (6.112).
- (v) *Redundancy:* Just like for the local Fourier transform, the continuous wavelet transform maps a function of one variable into a function of two variables. It is thus highly redundant, and this redundancy is expressed by the *reproducing kernel*:

$$K(a_0, b_0, a, b) = \langle \psi_{a_0, b_0}, \psi_{a, b} \rangle, \quad (6.32)$$

a four-dimensional function. Figure 6.17 shows the reproducing kernel of the Haar wavelet, namely  $K(1, 0, a, b)$ ; note that the reproducing kernel is zero at all dyadic scale points as the wavelets are then orthogonal to each other.

- (vi) *Characterization of Singularities:* This is one of the most important properties of the continuous wavelet transform, since, by looking at its behavior, we can infer the type and position of singularities occurring in the function.

For example, assume we are given a Dirac delta function at location  $t_0$ ,  $x(t) = \delta(t - t_0)$ . At scale  $a$ , only those Haar wavelets whose support straddles



**Figure 6.17:** The Haar wavelet transform of the Haar wavelet (6.12). This is also the reproducing kernel,  $K(1, 0, a, b)$ , of the Haar wavelet.

$t_0$  will produce nonzero coefficients. Because of the form of the Haar wavelet, the nonzero coefficients will extend over a region of size  $2^a$  around  $t_0$ . As  $a \rightarrow -\infty$ , these coefficients focus arbitrarily closely on the singularity. Moreover, these coefficients grow at a specific rate, another way to identify the type of a singularity. We will go into more details on this later in the chapter.

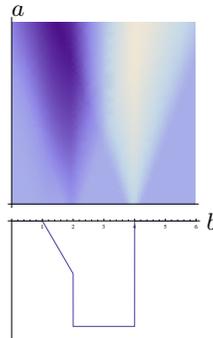
As a simple example, take  $x(t)$  to be the Heaviside function (4.7) with the step at location  $t_0$ . We want to see how the Haar wavelet (6.27) isolates and characterizes the step singularity. To do that, we will need two things: (1) The primitive of the wavelet, defined as

$$\theta(t) = \int_{-\infty}^t \psi(\tau) d\tau = \begin{cases} 1/2 - |t|, & |t| < 1/2; \\ 0, & \text{otherwise,} \end{cases} \quad (6.33)$$

that is, a triangle function (4.45) on the interval  $|t| < 1/2$ . Note that the primitive of the scaled and normalized wavelet  $a^{-1/2}\psi(t/a)$  is  $\sqrt{a}\theta(t/a)$ , or a factor  $a$  larger due to integration. (2) We also need the derivative of  $x(t)$ , which exists only in a generalized sense (using distributions) and can be shown to be a Dirac delta function at  $t_0$ ,  $x'(t) = \delta(t - t_0)$ .

Now, the continuous wavelet transform of the Heaviside function follows as

$$\begin{aligned} X(a, b) &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) x(t) dt \\ &\stackrel{(a)}{=} \left[ \sqrt{a} \theta\left(\frac{t-b}{a}\right) x(t) \right]_{-\infty}^{\infty} - \int_{-\infty}^{\infty} \sqrt{a} \theta\left(\frac{t-b}{a}\right) x'(t) dt \\ &\stackrel{(b)}{=} - \int_{-\infty}^{\infty} \sqrt{a} \theta\left(\frac{t-b}{a}\right) \delta(t - t_0) dt \\ &\stackrel{(c)}{=} \sqrt{a} \theta\left(\frac{t_0 - b}{a}\right), \end{aligned} \quad (6.34)$$



**Figure 6.18:** The Haar wavelet transform of a piecewise-polynomial function  $x(t)$  as in (6.35).

where (a) follows from integration by parts; (b) from  $\theta$  being of compact support; and (c) from the shifting property of the Dirac delta function in *Table 4.1*. Thus, as  $a \rightarrow 0$ , the continuous wavelet transform zooms towards the singularity and scales as  $a^{1/2}$ , with a shape given by the primitive of the wavelet  $\theta(t)$ ; thus, we may expect a triangle-like region of influence (see Figure 6.18 at the step discontinuity,  $t = 4$ , for illustration).

This discussion focused on the behavior of the Haar wavelet transform around a point of singularity; what about smooth regions? Take  $x(t)$  to be

$$x(t) = \begin{cases} t - 1, & 1 \leq t < 2; \\ 2, & 2 \leq t < 4; \\ 0, & \text{otherwise,} \end{cases} \quad (6.35)$$

and the Haar wavelet (6.27). The function  $x(t)$  has three singularities, discontinuities at  $t = 1, 2, 3$ . The wavelet has 1 zero moment, so it will have a zero inner product inside the interval  $[2, 3]$ , where  $x(t)$  is constant (see Figure 6.18). What happens in the interval  $[1, 2]$ , where  $x(t)$  is linear?

Calculate the continuous wavelet transform for some shift  $b \in [1, 2]$  for  $a$  sufficiently small so that the support of the shifted wavelet  $[b - a/2, b + a/2] \in [1, 2]$ :

$$X(a, b) = \frac{1}{\sqrt{a}} \left( \int_{b-a/2}^b t dt - \int_b^{b+a/2} t dt \right) = -\frac{1}{4} a^{3/2}. \quad (6.36)$$

Thus, the lack of a second zero moment (which would have produced a zero inner product) produces a residual of order  $a^{3/2}$  as  $a \rightarrow 0$ .

To study qualitatively the overall behavior, there will be two cones of influence at singular points 2 and 3, with an order  $a^{1/2}$  behavior as in (6.34), a constant of order  $a^{3/2}$  in the  $(1, 2)$  interval as in (6.36) (which spills over into the  $(2, 3)$  interval), and zero elsewhere, shown in Figure 6.18.

### Chapter Outline

We now follow the path set through this simple Haar example, and follow with more general developments. We start in Section 6.2 with iterated filter banks building scaling functions and wavelets as limits of iterated filters. We study issues of convergence and smoothness of resulting functions. In Section 6.3, we then define and look into the properties of wavelet series: localization, zero moments and decay of wavelet coefficients, before considering the characterization of singularities by the decay of the associated wavelet series coefficients. We study multiresolution analysis, revisiting the wavelet construction from an axiomatic point of view. In Section 6.4, we relax the constraints of nonredundancy to construct wavelet frames, midway between the nonredundant wavelet series and a completely redundant wavelet transform. We follow in Section 6.5 with the continuous wavelet transform. Section 6.6 is devoted to computational issues, in particular to Mallat's algorithm, which allows us to compute wavelet coefficients with an initial continuous-time projection followed by a discrete-time, filter-bank algorithm.

*Notation used in this chapter:* In most of this chapter, we consider real-valued wavelets only and thus the domain for the scale factor  $a$  is  $\mathbb{R}^+$ ; the extension to complex wavelets requires simply  $a \in \mathbb{R}$ ,  $a \neq 0$ .  $\square$

## 6.2 Scaling Function and Wavelets from Orthogonal Filter Banks

In the previous section, we set the stage for this section by examining basic properties of iterated filter banks with Haar filters. The results in this section should thus not come as a surprise, as they generalize those for Haar filter banks.

We start with an orthogonal filter bank with filters  $g$  and  $h$  whose properties were summarized in Table 1.9, Chapter 1, where we used these filters and their shifts by multiples by two as the basis for  $\ell^2(\mathbb{Z})$ . This orthonormal basis was implemented using a critically-sampled two-channel filter bank with down- and upsampling by 2, an orthogonal synthesis lowpass/highpass filter pair  $g_n, h_n$  and a corresponding analysis lowpass/highpass filter pair  $g_{-n}, h_{-n}$ . We then used these filters and the associated two-channel filter bank as building blocks for a DWT in Chapter 3. For example, we saw that in a 3-level iterated Haar filter bank, the lowpass and highpass at level 3 were given by (3.1c)–(3.1d) and plotted in Figure 3.4; we repeated the last-level filters in (6.2). Another example, a 3-level iterated filter bank with Daubechies filters, was given in Example 3.1.

### 6.2.1 Iterated Filters

As for the Haar case, we come back to filters and their iterations, and associate a continuous-time function to the discrete-time sequence representing the impulse response of the iterated filter.

We assume a length- $L$  orthogonal lowpass/highpass filter pair  $(g_n, h_n)$ , and

write the equivalent filters at the last level of a  $J$ -level iterated filter bank:

$$G^{(J)}(z) = \prod_{\ell=0}^{J-1} G(z^{2^\ell}) = G^{(J-1)}(z) G(z^{2^{J-1}}), \quad (6.37a)$$

$$H^{(J)}(z) = \prod_{\ell=0}^{J-2} G(z^{2^\ell}) H(z^{2^{J-1}}) = G^{(J-1)}(z) H(z^{2^{J-1}}). \quad (6.37b)$$

We know that, by construction, these filters are orthonormal to their shifts by  $2^J$ , (3.6a), (3.11a), as well as orthogonal to each other, (3.14a).

The equivalent filters  $g^{(J)}$ ,  $h^{(J)}$  have norm 1 and length  $L^{(J)}$ , which can be upper bounded by (see (3.5b))

$$L^{(J)} \leq (L-1)2^J. \quad (6.38)$$

### 6.2.2 Scaling Function and its Properties

We now associate a piecewise-constant function  $\varphi^{(J)}(t)$  to  $g_n^{(J)}$  so that  $\varphi^{(J)}(t)$  is of finite length and norm 1. Since the number of piecewise segments (equal to the number of nonzero coefficients of  $g_n^{(J)}$ ) grows exponentially with  $J$  (see (6.38)), we choose their width as  $2^{-J}$ , upper bounding the length of  $\varphi^{(J)}(t)$  by  $(L-1)$ :

$$\text{support} \left( \varphi^{(J)}(t) \right) \subset [0, L-1], \quad (6.39)$$

where  $\text{support}(\cdot)$  stands for the interval of the real line where the function is different from zero. For  $\varphi^{(J)}(t)$  to inherit the unit-norm property from  $g_n^{(J)}$ , we choose the height of the piecewise segments as  $2^{J/2}g_n^{(J)}$ . Then, the  $n$ th piece of the  $\varphi^{(J)}(t)$  contributes

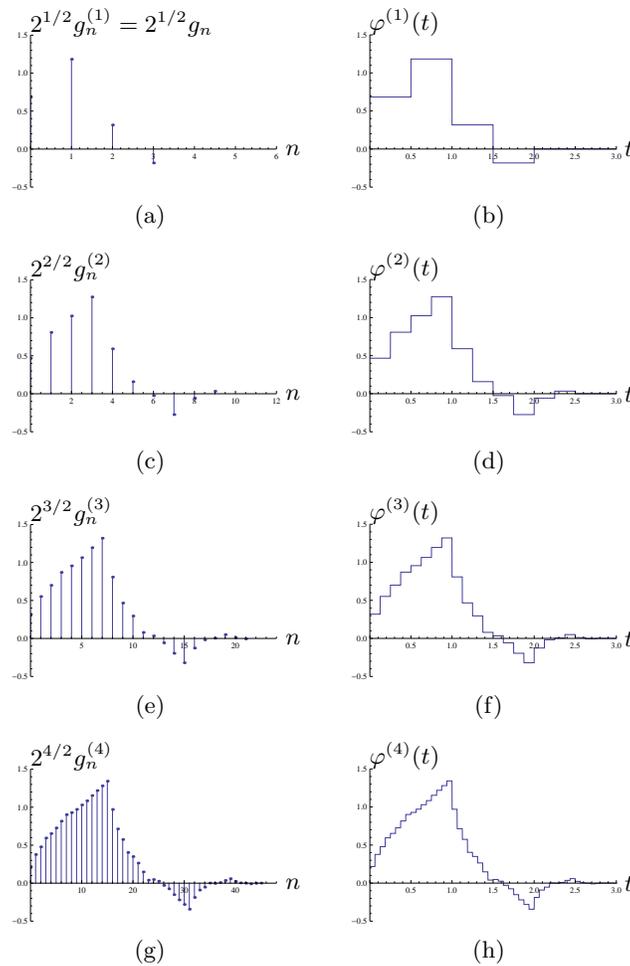
$$\int_{n/2^J}^{(n+1)/2^J} |\varphi^{(J)}(t)|^2 dt = \int_{n/2^J}^{(n+1)/2^J} 2^J (g_n^{(J)})^2 dt = (g_n^{(J)})^2$$

to  $\varphi^{(J)}(t)$ . Summing up the individual contributions,

$$|\varphi^{(J)}(t)|^2 = \sum_{n=0}^{L^{(J)}-1} \int_{n/2^J}^{(n+1)/2^J} |\varphi^{(J)}(t)|^2 dt = \sum_{n=0}^{L^{(J)}-1} (g_n^{(J)})^2 = 1.$$

We have thus defined the piecewise-constant function as

$$\begin{aligned} \varphi^{(J)}(t) &= 2^{J/2} g_n^{(J)}, & \frac{n}{2^J} &\leq t < \frac{n+1}{2^J}, \\ &= \sum_{n=0}^{L^{(J)}-1} g_n^{(J)} 2^{J/2} \varphi_h(2^J t - n), \end{aligned} \quad (6.40)$$



**Figure 6.19:** Iterated filter  $2^{J/2}g_n^{(J)}$  and associated piecewise-constant function  $\varphi^{(J)}(t)$  based on a 4-tap Daubechies lowpass filter (3.10) at level (a)–(b)  $J = 1$ ; (c)–(d)  $J = 2$ ; (e)–(f)  $J = 3$ ; and (g)–(h)  $J = 4$ . Note that we have rescaled the equivalent filters' impulse responses as well as plotted them at different discrete intervals to highlight the correspondences with their piecewise-constant functions.

where  $\varphi_h(t)$  is the Haar scaling function (box function) from (6.6). We verified that the above iterated function is supported on a finite interval and has unit norm.<sup>49</sup>

In Figure 6.19, we show a few iterations of a 4-tap filter from Example 3.1 and its associated piecewise-constant function. The piecewise-constant function  $\varphi^{(J)}$  has geometrically decreasing piecewise segments and a support contained in the

<sup>49</sup>We could have defined a piecewise-linear function instead of a piecewise-constant one, but it does not change the behavior of the limit we will study.

interval  $[0, 3]$ . From the figure it is clear that the *smoothness* of  $\varphi^{(J)}(t)$  depends on the *smoothness* of  $g_n^{(J)}$ . If the latter tends, as  $J$  increases, to a sequence with little local variation, then the piecewise-constant approximation will tend to a smooth function as well, as the piecewise segments become finer and finer. On the contrary, if  $g_n^{(J)}$  has too much variation as  $J \rightarrow \infty$ , the sequence of functions  $\varphi^{(J)}(t)$  might not have a limit as  $J \rightarrow \infty$ . This leads to the following necessary condition for the filter  $g_n$ , the proof of which is given in Solved Exercise ??:

**THEOREM 6.1 (NECESSITY OF A ZERO AT  $\pi$ )** For the  $\lim_{J \rightarrow \infty} \varphi^{(J)}(t)$  to exist, it is necessary for  $G(e^{j\omega})$  to have a zero at  $\omega = \pi$ .

As a direct corollary of this result, the necessity of a zero at  $\omega = \pi$  translates also to the necessity of

$$G(e^{j\omega})|_{\omega=0} = \sqrt{2}, \quad (6.41)$$

because of (1.13). We are now ready to define the limit function:

**DEFINITION 6.2 (SCALING FUNCTION)** We call the scaling function  $\varphi(t)$  the limit, when it exists, of:

$$\varphi(t) = \lim_{J \rightarrow \infty} \varphi^{(J)}(t), \quad (6.42)$$

**Scaling Function in the Fourier Domain** We now find the Fourier transform of  $\varphi^{(J)}(t)$ , denoted by  $\Phi^{(J)}(\omega)$ . The functions  $\varphi^{(J)}(t)$  is a linear combination of box functions, each of width  $1/2^J$  and height  $2^{J/2}$ , where the unit box function is equal to the Haar scaling function (6.6), with the Fourier transform  $\Phi_h(\omega)$  as in (6.7). Using the scaling-in-time property of the Fourier transform (4.55a), the transform of a box function on the interval  $[0, 1/2^J)$  of height  $2^{J/2}$  is

$$\Phi_h^{(J)}(\omega) = 2^{-J/2} e^{-j\omega/2^{J+1}} \frac{\sin(\omega/2^{J+1})}{\omega/2^{J+1}}. \quad (6.43)$$

Shifting the  $n$ th box to start at  $t = n/2^J$  multiplies its Fourier transform by  $e^{-j\omega n/2^J}$ . Putting it all together, we find

$$\begin{aligned} \Phi^{(J)}(\omega) &= \Phi_h^{(J)}(\omega) \sum_{n=0}^{L^{(J)}-1} e^{-j\omega n/2^J} g_n^{(J)} \stackrel{(a)}{=} \Phi_h^{(J)}(\omega) G^{(J)}(e^{j\omega/2^J}) \\ &\stackrel{(b)}{=} \Phi_h^{(J)}(\omega) \prod_{\ell=0}^{J-1} G(e^{j\omega 2^\ell/2^J}) \stackrel{(c)}{=} \Phi_h^{(J)}(\omega) \prod_{\ell=1}^J G(e^{j\omega/2^\ell}), \end{aligned} \quad (6.44)$$

where (a) follows from the definition of the DTFT (3.78a); (b) from (6.3); and (c) from reversing the order of the factors in the product.

In the sequel, we will be interested in what happens in the limit, when  $J \rightarrow \infty$ . For any finite  $\omega$ , the effect of the interpolation function  $\Phi_h^{(J)}(\omega)$  becomes negligible as  $J \rightarrow \infty$ . Indeed in (6.43), both terms dependent on  $\omega$  tend to 1 as  $J \rightarrow \infty$  and only the factor  $2^{-J/2}$  remains. So, in (6.44), the key term is the product, which becomes an infinite product, which we now define:

**DEFINITION 6.3 (FOURIER TRANSFORM OF THE INFINITE PRODUCT)** We call  $\Phi(\omega)$  the limit, if it exists, of the infinite product:

$$\Phi(\omega) = \lim_{J \rightarrow \infty} \Phi^{(J)}(\omega) = \prod_{\ell=1}^{\infty} 2^{-1/2} G(e^{j\omega/2^\ell}). \quad (6.45)$$

The corollary to Theorem 6.1, (6.41) is now clear; if  $G(1) > \sqrt{2}$ ,  $\Phi(0)$  would grow unbounded, and if  $G(1) < \sqrt{2}$ ,  $\Phi(0)$  would be zero, contradicting the fact that  $\Phi(\omega)$  is the limit of lowpass filters and hence a lowpass function.

A more difficult question is to understand when the limits of the time-domain iteration  $\varphi^{(J)}(t)$  (6.40) and the Fourier-domain iteration  $\Phi^{(J)}(\omega)$  (6.44) form the Fourier-transform pair. We will show in Example 6.1 that this is a nontrivial question. As the exact conditions are technical and beyond the scope of our text, we concentrate on those cases when the limits in Definitions 6.2 and 6.3 are well defined and form a Fourier-transform pair, that is, when

$$\varphi(t) \xleftrightarrow{\text{FT}} \Phi(\omega),$$

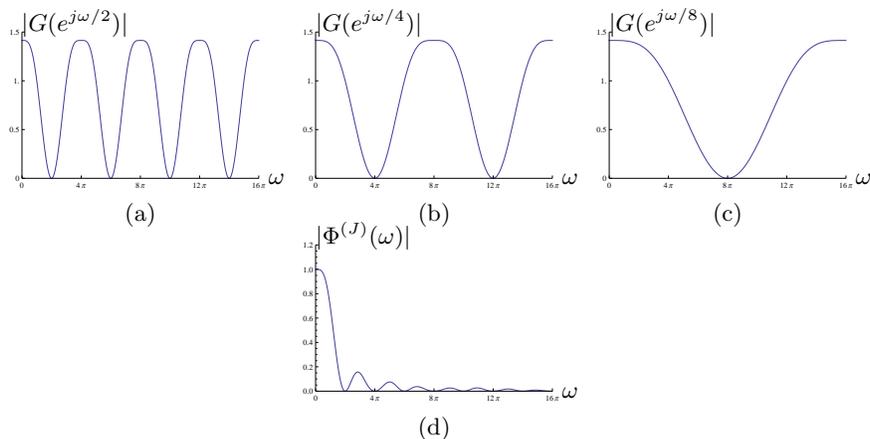
We now look into the behavior of the infinite product. If  $\Phi(\omega)$  decays sufficiently fast in  $\omega$ , the scaling function  $\varphi(t)$  will be smooth. How this can be done while maintaining other desirable properties (such as compact support and orthogonality) is the key result for designing wavelet bases from iterated filter banks.

**EXAMPLE 6.1 (FOURIER TRANSFORM OF THE INFINITE PRODUCT)** To gain intuition, we now look into examples of filters and their associated infinite products.

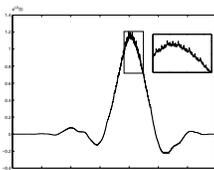
- (i) *Daubechies filter with two zeros at  $\omega = \pi$ :* We continue with our example of the Daubechies lowpass filter from (3.10) with its associated piecewise-constant function in Figure 6.19. In the Fourier-domain product (6.44), the terms are periodic with periods  $4\pi, 8\pi, \dots, 2^J 2\pi$ , since  $G(e^{j\omega})$  is  $2\pi$ -periodic (see Figure 6.20(a)–(c)). We show the product in part (d) of the figure. The terms are oscillating depending on their periodicity, but the product decays rather nicely. We will study this decay in detail shortly.
- (ii) *Length-4 filter designed using lowpass approximation method:* Consider the orthogonal filter designed using the window method in Example 1.2. This filter does not have a zero at  $\omega = \pi$ , since

$$G(e^{j\omega})|_{\omega=\pi} \approx 0.389.$$

Its iteration is shown in Figure 6.21, with noticeable high-frequency oscillations, prohibiting convergence of the iterated function  $\varphi^{(J)}(t)$ .



**Figure 6.20:** Factors (a)  $|G(e^{j\omega/2})|$ , (b)  $|G(e^{j\omega/4})|$ , and (c)  $|G(e^{j\omega/8})|$  that appear in (d) the Fourier-domain product  $\Phi^{(J)}(\omega)$ .



**Figure 6.21:** Iteration of a filter without a zero at  $\omega = \pi$ . The high-frequency oscillations prohibit the convergence of the iterated function  $\varphi^{(J)}(t)$ .

(iii) *Stretched Haar filter:* Instead of the standard Haar filter, consider:

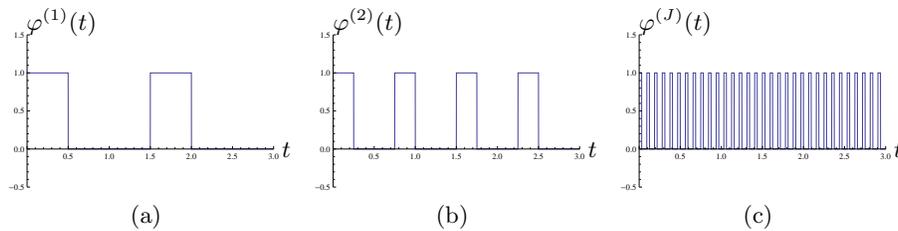
$$g = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \xleftrightarrow{ZT} G(z) = \frac{1}{\sqrt{2}}(1 + z^{-3}).$$

It is clearly an orthogonal lowpass filter and has one zero at  $\omega = \pi$ . However, unlike the Haar filter, its iteration is highly unsmooth. Consider the equivalent filter after  $J$  stages of iteration:

$$g^{(J)} = \frac{1}{2^{J/2}} \begin{bmatrix} \mathbf{1} & 0 & 0 & 1 & \dots & 1 & 0 & 0 & 1 \end{bmatrix}.$$

The piecewise-constant function  $\varphi^{(J)}(t)$  inherits this lack of smoothness, and does not converge pointwise to a proper limit, as shown graphically in Figure 6.22. Considering the frequency domain and the infinite product, it turns out that  $\mathcal{L}^2$  convergence fails as well (see Exercise ??).

The examples above show that iterated filters and their associated graphical functions behave quite differently. The Haar case we saw in the previous section



**Figure 6.22:** Iteration of the stretched Haar filter with impulse response  $g = [1/\sqrt{2} \ 0 \ 0 \ 1/\sqrt{2}]$ . (a)  $\varphi^{(1)}(t)$ . (b)  $\varphi^{(2)}(t)$ . (c)  $\varphi^{(J)}(t)$ .

was trivial, the 4-tap filters showed a smooth behavior, and the stretched Haar filter pointed out potential convergence problems.

In the sequel, we concentrate on orthonormal filters with  $N \geq 1$  zeros at  $\omega = \pi$ , or

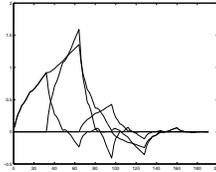
$$G(e^{j\omega}) = \left( \frac{1 + e^{-j\omega}}{2} \right)^N R(e^{j\omega}), \quad (6.46)$$

with  $R(e^{j\omega})|_{\omega=\pi} = \sqrt{2}$  for the limit to exist. We assume (1) pointwise convergence of the iterated function  $\varphi^{(J)}(t)$  to  $\varphi(t)$ , (2) pointwise convergence of the iterated Fourier-domain function  $\Phi^{(J)}(\omega)$  to  $\Phi(\omega)$ , and finally (3) that  $\varphi(t)$  and  $\Phi(\omega)$  are a Fourier-transform pair. In other words, we avoid all convergence issues and concentrate on the well-behaved cases exclusively.

**Two-Scale Equation** We have seen in Section 6.1.1 that the Haar scaling function satisfies a two-scale equation 6.8. This is true in general, except that more terms will be involved in the summation. To show this, we start with the Fourier-domain limit of the infinite product (6.45):

$$\begin{aligned} \Phi(\omega) &= \prod_{\ell=1}^{\infty} 2^{-1/2} G(e^{j\omega/2^\ell}) \stackrel{(a)}{=} 2^{-1/2} G(e^{j\omega/2}) \prod_{\ell=2}^{\infty} 2^{-1/2} G(e^{j\omega/2^\ell}), \\ &\stackrel{(b)}{=} 2^{-1/2} G(e^{j\omega/2}) \Phi(\omega/2) \stackrel{(c)}{=} 2^{-1/2} \sum_{n=0}^{L-1} g_n e^{-j\omega n/2} \Phi(\omega/2), \\ &\stackrel{(d)}{=} \sum_{n=0}^{L-1} g_n \left[ 2^{-1/2} e^{-j\omega n/2} \Phi(\omega/2) \right], \end{aligned} \quad (6.47)$$

where in (a) we took one factor out,  $2^{-1/2} G(e^{j\omega/2})$ ; in (b) we recognize the infinite product as  $\Phi(\omega/2)$ ; (c) follows from the definition of the DTFT (3.78a); and in (d) we just rearranged the terms. Then, using the scaling-in-time property of the Fourier transform (4.55a),  $\Phi(\omega/2) \xrightarrow{\text{FT}} 2\varphi(2t)$ , and the shift-in-time property



**Figure 6.23:** Two-scale equation for the Daubechies scaling function. (a) The scaling function  $\varphi(t)$  and (b) expressed as a linear combination of  $\varphi(2t - n)$ .

(4.53),  $e^{-j\omega n/2}X(\omega) \xleftrightarrow{\text{FT}} x(t - n/2)$ , we get the two-scale equation:

$$\varphi(t) = \sqrt{2} \sum_{n=0}^{L-1} g_n \varphi(2t - n), \quad (6.48)$$

shown in Figure 6.23 for the Daubechies 4-tap filter from Examples 6.1 and 6.2.

**Smoothness** As seen earlier, the key is to understand the infinite product (6.45) which becomes, using (6.46),

$$\begin{aligned} \Phi(\omega) &= \prod_{\ell=1}^{\infty} 2^{-1/2} \left( \frac{1 + e^{-j\omega/2^\ell}}{2} \right)^N R(e^{j\omega/2^\ell}) \\ &= \underbrace{\left( \prod_{\ell=1}^{\infty} \left( \frac{1 + e^{-j\omega/2^\ell}}{2} \right) \right)^N}_{A(\omega)} \underbrace{\prod_{\ell=1}^{\infty} 2^{-1/2} R(e^{j\omega/2^\ell})}_{B(\omega)}. \end{aligned} \quad (6.49)$$

Our goal is to see if  $\Phi(\omega)$  has a sufficiently fast decay for large  $\omega$ . We know from *Chapter 4*, (4.82a), that if  $|\Phi(\omega)|$  decays faster than  $1/|\omega|$  for large  $|\omega|$ , then  $\varphi(t)$  is bounded and continuous. Consider first the product

$$\prod_{\ell=1}^{\infty} \left( \frac{1 + e^{-j\omega/2^\ell}}{2} \right) \stackrel{(a)}{=} \prod_{\ell=1}^{\infty} 2^{-1/2} \frac{1}{\sqrt{2}} (1 + e^{-j\omega/2^\ell}) \stackrel{(b)}{=} e^{-j\omega/2} \frac{\sin(\omega/2)}{\omega/2},$$

where in (a) we extracted the Haar filter (6.1a), and (b) follows from (6.45) as well as the Haar case (6.7). The decay of this Fourier transform is of order  $O(1/|\omega|)$ , and thus,  $A(\omega)$  in (6.49) decays as  $O(1/|\omega|^N)$ .<sup>50</sup> So, as long as  $|B(\omega)|$  does not grow faster than  $|\omega|^{N-1-\varepsilon}$ ,  $\varepsilon > 0$ , the product (6.49) will decay fast enough to satisfy (4.82a), leading to a continuous scaling function  $\varphi(t)$ . We formalize this discussion in the following theorem, the proof of which is given in Solved Exercise ??:

<sup>50</sup>In time domain, it is the convolution of  $N$  box functions, or a  $B$  spline of order  $N - 1$  (see *Chapter 6*).

**THEOREM 6.4 (SMOOTHNESS OF THE SCALING FUNCTION)** With  $R(e^{j\omega})$  as in (6.46), if

$$B = \sup_{\omega \in [0, 2\pi]} |R(e^{j\omega})| < 2^{N-1/2}, \quad (6.50)$$

then, as  $J \rightarrow \infty$ , the iterated function  $\varphi^{(J)}(t)$  converges pointwise to a continuous function  $\varphi(t)$  with the Fourier transform

$$\Phi(\omega) = \prod_{\ell=1}^{\infty} 2^{-1/2} G(e^{j\omega/2^\ell}).$$

Condition (6.50) is sufficient, but not necessary: many filters fail the test but still lead to continuous limits (and more sophisticated tests can be used).

If we strengthened the bound to

$$B < 2^{N-k-1/2} \quad k \in \mathbb{N},$$

then  $\varphi(t)$  would be continuous and  $k$ -times differentiable (see Exercise TBD).

**EXAMPLE 6.2 (SMOOTHNESS OF THE SCALING FUNCTION)** We now test the continuity condition (6.50) on the two filters we have used most often.

The Haar filter

$$G(e^{j\omega}) = \frac{1}{\sqrt{2}}(1 + e^{-j\omega}) = \left( \frac{1 + e^{-j\omega}}{2} \right) \underbrace{\sqrt{2}}_{R(e^{j\omega})},$$

has  $N = 1$  zero at  $\omega = \pi$  and  $R(z) = \sqrt{2}$ . Thus,  $B = \sqrt{2}$ , which does not meet the inequality in (6.50). According to Theorem 6.4,  $\varphi(t)$  may or may not be continuous (and we know it is not).

The Daubechies filter (3.10)

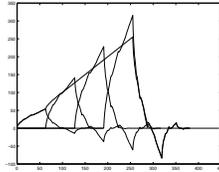
$$G(e^{j\omega}) = \left( \frac{1 + e^{-j\omega}}{2} \right)^2 \underbrace{\frac{1}{\sqrt{2}}(1 + \sqrt{3} + (1 - \sqrt{3})e^{-j\omega})}_{R(e^{j\omega})},$$

has  $N = 2$  zero at  $\omega = \pi$ . The supremum of  $|R(e^{j\omega})|$  is attained at  $\omega = \pi$ ,

$$B = \sup_{\omega \in [0, 2\pi]} |R(e^{j\omega})| = \sqrt{6} < 2^{3/2},$$

and thus, the scaling function  $\varphi(t)$  must be continuous.

**Reproduction of Polynomials** We have seen in *Chapter 6* that splines of degree  $N$  and their shifts can reproduce polynomials of degree up to  $N$ . Given that the scaling functions based on a filter having  $N$  zeros at  $\omega = \pi$  contain a *spline part* of



**Figure 6.24:** An example of the reproduction of polynomials by the scaling function and its shifts. The scaling function  $\varphi(t)$  is based on the Daubechies filter with two zeros at  $\pi$ , (3.10), and reproduces the linear function  $x(t) = t$  (on an interval because of the finite number of scaling functions used).

degree  $(N - 1)$ , linear combinations of  $\{\varphi(t - n)\}_{n \in \mathbb{Z}}$  can reproduce polynomials of degree  $(N - 1)$ . We illustrate this property in Figure 6.24, where the Daubechies filter with two zeros at  $\pi$ , (3.10), reproduces the linear function  $x(t) = t$ . (We give the proof of this property later in the chapter.)

**Orthogonality to Integer Shifts** As we have seen in the Haar case already, the scaling function is orthogonal to its integer shifts, a property inherited from the underlying filter:

$$\langle \varphi(t), \varphi(t - n) \rangle_t = \delta_n. \quad (6.51)$$

Since  $\varphi(t)$  is defined through a limit and the inner product is continuous in both arguments, orthogonality (6.51) follows from the orthogonality of  $\varphi^{(J)}(t)$  and its integer shifts for any  $J$ :

$$\langle \varphi^{(J)}(t), \varphi^{(J)}(t - n) \rangle_t = \delta_n, \quad J \in \mathbb{Z}^+, \quad (6.52)$$

which follows, in turn, from the same property for the iterated filter  $g_n^{(J)}$  in (3.6a):

$$\begin{aligned} & \langle \varphi^{(J)}(t), \varphi^{(J)}(t - k) \rangle_t \\ & \stackrel{(a)}{=} \left\langle \sum_{n=0}^{L^{(J)}-1} g_n^{(J)} 2^{J/2} \varphi_h(2^J t - n), \sum_{m=0}^{L^{(J)}-1} g_m^{(J)} 2^{J/2} \varphi_h(2^J(t - k) - m) \right\rangle_t \\ & \stackrel{(b)}{=} \sum_{n=0}^{L^{(J)}-1} \sum_{m=0}^{L^{(J)}-1} g_n^{(J)} g_m^{(J)} \int_{-\infty}^{\infty} 2^J \varphi_h(2^J t - n) \varphi_h(2^J t - 2^J k - m) dt \\ & \stackrel{(c)}{=} \sum_{n=0}^{L^{(J)}-1} g_n^{(J)} g_{n-2^J k}^{(J)} = \langle g_n^{(J)}, g_{n-2^J k}^{(J)} \rangle_n \stackrel{(d)}{=} \delta_k, \end{aligned}$$

where (a) follows from (6.40); in (b) we took the sums and filter coefficients out of the inner product; (c) from the orthogonality of the Haar scaling functions; and (d) from the orthogonality of the filters themselves, (3.6a).

The orthogonality (6.51) at scale 0 has counterparts at other scales:

$$\langle \varphi(2^\ell t), \varphi(2^\ell t - n) \rangle_t = 2^{-\ell} \delta_n, \quad (6.53)$$

easily verified by changing the integration variable.

### 6.2.3 Wavelet Function and its Properties

The scaling function we have just seen is lowpass in nature (if the underlying filter  $g$  is lowpass in nature). Similarly to what we have done for the scaling function, we can construct a *wavelet function* (or, simply *wavelet*) that will be bandpass in nature (if the underlying filter  $h$  is highpass in nature).

We thus associate a piecewise-constant function  $\psi^{(J)}(t)$  to  $h_n^{(J)}$ , the impulse response of (6.37b), in such a way that  $\psi^{(J)}(t)$  is of finite length and of norm 1; we use the same arguments as before to determine the width and height of the piecewise segments, leading to

$$\psi^{(J)}(t) = 2^{J/2} h_n^{(J)} \quad \frac{n}{2^J} \leq t < \frac{n+1}{2^J}. \quad (6.54)$$

Unlike  $\varphi^{(J)}(t)$ , our new object of interest  $\psi^{(J)}(t)$  is a bandpass function. In particular, because  $H(e^{j\omega})|_{\omega=\pi} = 0$ , its Fourier transform  $\Psi(\omega)$  satisfies

$$\Psi(\omega)|_{\omega=0} = 0.$$

Again, we are interested in what happens when  $J \rightarrow \infty$ . Clearly, this involves an infinite product, but it is the same infinite product we studied for the convergence of  $\varphi^{(J)}(t)$  towards  $\varphi(t)$ . In short, we assume this question to be settled. The development parallels the one for the scaling function, with the important twist of consistently replacing the lowpass filter  $G(z^{2^{J-1}})$  by the highpass filter  $H(z^{2^{J-1}})$ . We do not repeat the details, but rather indicate the main points. Equation (6.44) becomes

$$\Psi^{(J)}(\omega) = \Phi_h^{(J)}(\omega) H(e^{j\omega/2}) \prod_{\ell=2}^J G(e^{j\omega/2^\ell}). \quad (6.55)$$

Similarly to the scaling function, we define the wavelet as the limit of  $\psi^{(J)}(t)$  or  $\Psi^{(J)}(\omega)$ , where we now assume that both are well defined and form a Fourier-transform pair.

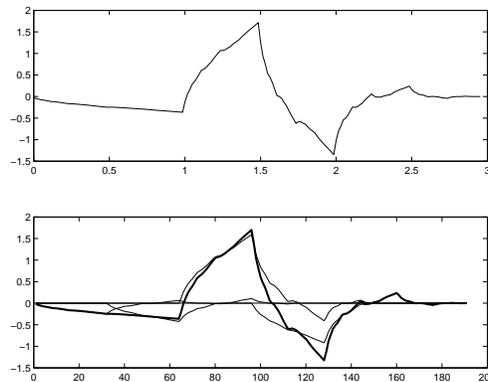
**DEFINITION 6.5 (WAVELET)** Assuming the limit to exist, we define the wavelet in time and frequency domains to be

$$\psi(t) = \lim_{J \rightarrow \infty} \psi^{(J)}(t), \quad (6.56a)$$

$$\Psi(\omega) = \lim_{J \rightarrow \infty} \Psi^{(J)}(\omega). \quad (6.56b)$$

From (6.55) and using the steps leading to (6.45), we can write

$$\Psi(\omega) = 2^{-1/2} H(e^{j\omega/2}) \prod_{\ell=2}^{\infty} 2^{-1/2} G(e^{j\omega/2^\ell}). \quad (6.57)$$



**Figure 6.25:** Wavelet based on the Daubechies highpass filter (6.60). (a) Wavelet  $\psi(t)$  and (b) the two-scale equation for the wavelet.

**Two-Scale Equation** Similarly to (6.47), we can rewrite (6.57) as

$$\Psi(\omega) = 2^{-1/2} H(e^{j\omega/2}) \Phi(\omega/2). \quad (6.58)$$

Taking the inverse Fourier transform, we get a relation similar to (6.48), namely

$$\psi(t) = \sqrt{2} \sum_{n=0}^{L-1} h_n \varphi(2t - n), \quad (6.59)$$

the two-scale equation for the wavelet. From the support of  $\varphi(t)$  in (6.39), it also follows that  $\psi(t)$  has the same support on  $[0, L - 1]$ . To illustrate the two-scale relation and also show a wavelet, consider the following example.

**EXAMPLE 6.3 (WAVELET AND THE TWO-SCALE EQUATION)** Take the Daubechies lowpass filter (3.10) and construct its highpass via (1.24). It has a double zero at  $\omega = 0$ , and is given by:

$$H(z) = \frac{1}{4\sqrt{2}} \left[ (\sqrt{3} - 1) + (3 - \sqrt{3})z^{-1} - (3 + \sqrt{3})z^{-2} + (1 + \sqrt{3})z^{-3} \right]. \quad (6.60)$$

Figure 6.25 shows the wavelet  $\psi(t)$  and the two-scale equation.

**Smoothness** Since the wavelet is a finite linear combination of scaling functions and their shifts as in (6.59), the smoothness is inherited from the scaling function, as illustrated in Figure 6.25(a).

**Zero-Moment Property** We assumed that the lowpass filter  $G(e^{j\omega})$  had  $N$  zeros ( $N \geq 1$ ) at  $\omega = \pi$ . Using (1.24) and applying it to (6.46), we get

$$H(z) = e^{j(L-1)\omega} \left( \frac{1 - e^{j\omega}}{2} \right)^N R(e^{j(\omega+\pi)}). \quad (6.61)$$

It has therefore  $N$  zeros at  $\omega = 0$ . These  $N$  zeros carry over directly to  $\Psi(\omega)$  because of (6.58) and  $\Phi(\omega)|_{\omega=0} = 1$ . Because of this

$$\left. \frac{d^n X(\omega)}{d\omega^n} \right|_{\omega=0} = 0. \quad (6.62)$$

We can now use the moment property of the Fourier transform, (4.59a), to find the Fourier-transform pair of the above equation, leading to

$$\int_{-\infty}^{\infty} t^n \psi(t) dt = 0 \quad n = 0, 1, \dots, N-1. \quad (6.63)$$

In other words, if  $p(t)$  is a polynomial function of degree  $(N-1)$ , its inner product with the wavelet at any shift and/or scale will be 0:

$$\langle p(t), \psi(at-b) \rangle_t = 0 \quad \text{for all } a, b \in \mathbb{R}. \quad (6.64)$$

Remembering that  $\varphi(t)$  is able to reproduce polynomials up to degree  $(N-1)$ , it is a good role split for the two functions: wavelets annihilate polynomial functions while scaling functions reproduce them.

**Orthogonality to Integer Shifts** In our quest towards building orthonormal bases of wavelets, we will need that the wavelet is orthogonal to its integer shifts. The derivation is analogous to that for the scaling function; we thus skip it here, and instead just summarize this and other orthogonality conditions:

$$\langle \psi(2^\ell t), \psi(2^\ell t - n) \rangle_t = 2^{-\ell} \delta_n, \quad (6.65a)$$

$$\langle \varphi(2^\ell t), \psi(2^\ell t - n) \rangle_t = 0. \quad (6.65b)$$

### 6.2.4 Scaling Function and Wavelets from Biorthogonal Filter Banks

As we have already seen with filter banks, not all cases of interest are necessarily orthogonal. In Chapter 1, we designed biorthogonal filter banks to obtain symmetric/antisymmetric FIR filters. Similarly, with wavelets, except for the Haar case, there exist no orthonormal and compactly-supported wavelet bases that are symmetric/antisymmetric. Since symmetry is often a desirable feature, we need to relax orthonormality. We thus set the stage here for the biorthogonal wavelet series by briefly going through the necessary concepts.

To start, we assume a quadruple  $(h_n, g_n, \tilde{h}_n, \tilde{g}_n)$  of biorthogonal impulse responses satisfying the four biorthogonality relations (1.64a)–(1.64d). We further require that both lowpass filters have at least one zero at  $\omega = \pi$ , and more if possible:

$$G(e^{j\omega}) = \left( \frac{1 + e^{-j\omega}}{2} \right)^N R(e^{j\omega}), \quad \tilde{G}(e^{j\omega}) = \left( \frac{1 + e^{-j\omega}}{2} \right)^{\tilde{N}} \tilde{R}(e^{j\omega}). \quad (6.66)$$

Since the highpass filters are related to the lowpass ones by (1.75), the highpass filters  $H(e^{j\omega})$  and  $\tilde{H}(e^{j\omega})$  will have  $\tilde{N}$  and  $N$  zeros at  $\omega = 0$ , respectively. In the biorthogonal case, unlike in the orthonormal one, there is no implicit normalization, so we will assume that

$$G(e^{j\omega})\big|_{\omega=0} = \tilde{G}(e^{j\omega})\big|_{\omega=0} = \sqrt{2},$$

which can be enforced by normalizing  $H(e^{j\omega})$  and  $\tilde{H}(e^{j\omega})$  accordingly.

Analogously to the orthogonal case, the iterated filters are given by

$$G^{(J)}(z) = \prod_{\ell=0}^{J-1} G(z^{2^\ell}), \quad \tilde{G}^{(J)}(z) = \prod_{\ell=0}^{J-1} \tilde{G}(z^{2^\ell}),$$

and we define scaling functions in the Fourier domain as

$$\Phi(\omega) = \prod_{\ell=0}^{\infty} 2^{-1/2} G(e^{j\omega/2^\ell}), \quad \tilde{\Phi}(\omega) = \prod_{\ell=0}^{\infty} 2^{-1/2} \tilde{G}(e^{j\omega/2^\ell}). \quad (6.67)$$

In the sequel, we will concentrate on well-behaved cases only, that is, when the infinite products are well defined. Also, the iterated time-domain functions corresponding to  $G^{(J)}(z)$  and  $\tilde{G}^{(J)}(z)$  have well-defined limits  $\varphi(t)$  and  $\tilde{\varphi}(t)$ , respectively, related to (6.67) by Fourier transform.

The two-scale relations follow similarly to the orthogonal case:

$$\Phi(\omega) = 2^{-1/2} G(e^{j\omega/2}) \Phi(\omega/2), \quad (6.68a)$$

$$\varphi(t) = \sqrt{2} \sum_{n=0}^{L-1} g_n \varphi(2t - n), \quad (6.68b)$$

as well as

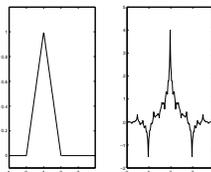
$$\tilde{\Phi}(\omega) = 2^{-1/2} \tilde{G}(e^{j\omega/2}) \tilde{\Phi}(\omega/2), \quad (6.69a)$$

$$\tilde{\varphi}(t) = \sqrt{2} \sum_{n=0}^{L-1} g_n \tilde{\varphi}(2t - n). \quad (6.69b)$$

EXAMPLE 6.4 (SCALING FUNCTION AND WAVELETS FROM LINEAR  $B$ -SPLINES)  
Choose as the lowpass filter

$$G(e^{j\omega}) = \sqrt{2} e^{j\omega} \left( \frac{1 + e^{-j\omega}}{2} \right)^2 = \frac{1}{2\sqrt{2}} (e^{j\omega} + 2 + e^{-j\omega}),$$

which has a double zero at  $\omega = 0$  and satisfies the normalization  $G(e^{j\omega})\big|_{\omega=0} = \sqrt{2}$ . Then, using (6.67), we compute  $\Phi(\omega)$  to be (4.47), that is, the Fourier transform of the triangle function (4.45), or linear  $B$ -spline. This is because  $G(z)$  is (up to a normalization and shift) the convolution of the Haar filter with



**Figure 6.26:** The triangle function and its dual. (a)  $\varphi(t)$  from the iteration of  $\frac{1}{2\sqrt{2}}[1, 2, 1]$ . (b)  $\tilde{\varphi}(t)$  from the iteration of  $\frac{1}{4\sqrt{2}}[-1, 2, 6, 2, -1]$ .

itself. Thus, the limit of the iterated filter is the convolution of the box function with itself, the result being shifted to be centered at the origin.

We now search for a biorthogonal scaling function  $\tilde{\varphi}(t)$  by finding first a (nonunique) biorthogonal lowpass filter  $\tilde{G}(z)$  satisfying (1.66). Besides the trivial solution  $\tilde{G}(z) = 1$ , the following is a solution as well:

$$\tilde{G}(z) = \frac{1}{4\sqrt{2}}(1+z)(1+z^{-1})(-z+4-z^{-1}) = \frac{1}{4\sqrt{2}}(-z^2+2z+6+2z^{-1}-z^{-2}),$$

obtained as one possible factorization of  $C(z)$  from Example 1.4. The resulting dual scaling function  $\tilde{\varphi}(t)$  looks quite irregular (see Figure 6.26). We could, instead, look for a  $\tilde{G}(z)$  with more zeros at  $\omega = \pi$  to obtain a smoother dual scaling function. For example, choose

$$\tilde{G}(z) = \frac{1}{64\sqrt{2}}(1+z)^2(1+z^{-1})^2(3z^2-18z+38-18z^{-1}+3z^{-2}),$$

leading to quite a different  $\tilde{\varphi}(t)$  (see Figure 6.27).

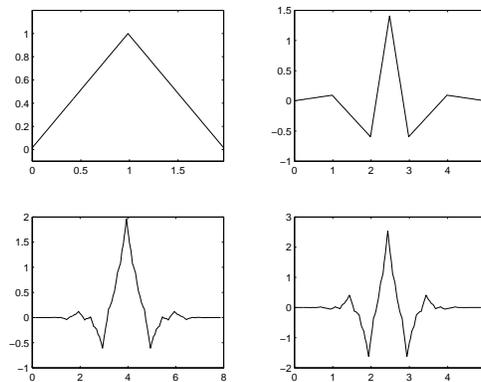
Choosing the highpass filters as in (1.75),

$$H(z) = z\tilde{G}(-z^{-1}), \quad \tilde{H}(z) = z^{-1}G(-z),$$

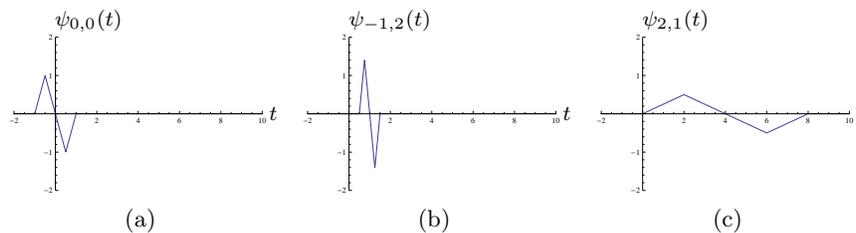
with only a minimal shift, since the lowpass filters are centered around the origin and symmetric, we get all four functions as in Figure 6.27.

### 6.3 Wavelet Series

So far, we have considered only a single scale with the two functions  $\varphi(t)$  and  $\psi(t)$ . Yet, as for the Haar case in Section 6.1, multiple scales are already lurking in the background through the two-scale equations (6.48),(6.59). And just like in the DWT in Chapter 3, the real action appears when all scales are considered as we have already seen with the Haar wavelet series.



**Figure 6.27:** Biorthogonal linear spline basis. (a) The linear  $B$ -spline is the triangle function  $\varphi(t)$ . (b) The linear  $B$ -spline wavelet  $\psi(t)$ . (c) The dual scaling function  $\tilde{\varphi}(t)$ . (d) The dual wavelet  $\tilde{\psi}(t)$ .



**Figure 6.28:** Example wavelets. (a) The prototype wavelet  $\psi(t) = \psi_{0,0}(t)$ ; (b)  $\psi_{-1,2}(t)$ ; (c)  $\psi_{2,1}(t)$ .

### 6.3.1 Definition of the Wavelet Series

We thus recall

$$\psi_{\ell,k}(t) = 2^{-\ell/2} \psi(2^{-\ell}t - k) = \frac{1}{2^{\ell/2}} \psi\left(\frac{t - 2^\ell k}{2^\ell}\right), \quad (6.70a)$$

$$\varphi_{\ell,k}(t) = 2^{-\ell/2} \varphi(2^{-\ell}t - k) = \frac{1}{2^{\ell/2}} \varphi\left(\frac{t - 2^\ell k}{2^\ell}\right), \quad (6.70b)$$

for  $\ell, k \in \mathbb{Z}$ , with the understanding that the basic scaling function  $\varphi(t)$  and the wavelet  $\psi(t)$  are no longer Haar, but can be more general. As before, for  $\ell = 0$ , we have the usual scaling function and wavelet and their integer shifts; for  $\ell > 0$ , the functions are stretched by a power of 2, and the shifts are proportionally increased; and for  $\ell < 0$ , the functions are compressed by a power of 2, with appropriately reduced shifts. Both the scaling function and the wavelet are of unit norm, and that at all scales (a few examples are given in Figure 6.28).

**Two-Scale Equations at Nonconsecutive Scales** Since we want to deal with multiple scales (not just two), we extend the two-scale equations for  $\varphi(t)$  and  $\psi(t)$  across arbitrary scales that are powers of 2:

$$\begin{aligned}\Phi(\omega) &\stackrel{(a)}{=} 2^{-1/2} G(e^{j\omega/2}) \Phi(\omega/2), \\ &\stackrel{(b)}{=} 2^{-1} G(e^{j\omega/2}) G(e^{j\omega/4}) \Phi(\omega/4), \\ &\stackrel{(c)}{=} 2^{-1} G^{(2)}(e^{j\omega/4}) \Phi(\omega/4), \\ &\vdots \\ &= 2^{-k/2} G^{(k)}(e^{j\omega/2^k}) \Phi(\omega/2^k),\end{aligned}\tag{6.71a}$$

$$\varphi(t) = 2^{k/2} \sum_{n=0}^{L-1} g_n^{(k)} \varphi(2^k t - n),\tag{6.71b}$$

for  $k \in \mathbb{Z}^+$ , where both (a) and (b) follow from the two-scale equation in the Fourier domain, (6.47); (c) from the expression for the equivalent filter, (6.37a); and (d) by repeatedly applying the same (see Exercise ??). The last expression is obtained by applying the inverse DTFT to (6.71a).

Using an analogous derivation for the wavelet, we get

$$\Psi(\omega) = 2^{-k/2} H^{(k)}(e^{j\omega/2^k}) \Phi(\omega/2^k),\tag{6.72a}$$

$$\psi(t) = 2^{k/2} \sum_{n=0}^{L-1} h_n^{(k)} \varphi(2^k t - n),\tag{6.72b}$$

for  $k = 2, 3, \dots$ . The attractiveness of the above expressions lies in their ability to express any  $\varphi_{\ell,k}(t)$ ,  $\psi_{\ell,k}(t)$ , in terms of a linear combination of an appropriately scaled  $\varphi(t)$ , where the linear combination is given by the coefficients of an equivalent filter  $g_n^{(k)}$  or  $h_n^{(k)}$ . We are now ready for the main result of this chapter:

**THEOREM 6.6 (ORTHONORMAL BASIS FOR  $\mathcal{L}^2(\mathbb{R})$ )** The continuous-time wavelet  $\psi(t)$  satisfying (6.59) and its shifts and scales,

$$\{\psi_{\ell,k}(t)\} = \left\{ \frac{1}{\sqrt{2^\ell}} \psi\left(\frac{t - 2^\ell k}{2^\ell}\right) \right\}, \quad \ell, k \in \mathbb{Z},\tag{6.73}$$

form an orthonormal basis for the space of square-integrable functions,  $\mathcal{L}^2(\mathbb{R})$ .

*Proof.* To prove the theorem, we must prove that (i)  $\{\psi_{\ell,k}(t)\}_{\ell,k \in \mathbb{Z}}$  is an orthonormal set and (ii) it is complete. The good news is that most of the hard work has already been done while studying the DWT in Theorem 3.2, Chapter 3.

- (i) We have already shown that the wavelets and their shifts are orthonormal at a single scale, (6.65a), and need to show the same across scales:

$$\begin{aligned} \langle \psi_{\ell,k}(t), \psi_{m,n}(t) \rangle_t &\stackrel{(a)}{=} 2^{-\ell} \langle \psi_{0,k}(\tau), \psi_{-i,n}(\tau) \rangle_{\tau}, \\ &\stackrel{(b)}{=} 2^{-\ell} \langle 2^{i/2} \sum_n h_n^{(i)} \varphi_{-i,2^i k+n}(\tau), \psi_{-i,n}(\tau) \rangle_{\tau}, \\ &\stackrel{(c)}{=} 2^{-\ell+i/2} \sum_n h_n^{(i)} \langle \varphi_{-i,2^i k+n}(\tau), \psi_{-i,n}(\tau) \rangle_{\tau} = 0, \end{aligned}$$

where (a) follows from assuming (without loss of generality)  $\ell = m + i$ ,  $i > 0$ , and change of variable  $t = 2^\ell \tau$ ; (b) from two-scale equation for the wavelet (6.72b); and (c) from the linearity of the inner product as well as orthogonality of the wavelet and scaling function (6.65a).

- (ii) The proof of completeness is more involved, and thus, we show it only for the Haar case. *Further Reading* gives pointers to texts with the full proof. Consider a unit-norm function  $x(t)$  such that  $x(t) = 0$  for  $t < 0$  with finite length at most  $2^J$  for some  $J \in \mathbb{Z}$ .<sup>51</sup> We approximate  $x(t)$  by a piecewise-constant approximation at scale  $\ell$ , (where  $\ell \ll J$ ), or

$$\begin{aligned} x^{(\ell)}(t) &= 2^{-\ell} \int_{2^\ell k}^{2^\ell(k+1)} x(\tau) d\tau, \quad 2^\ell k \leq t < 2^\ell(k+1), \\ &\stackrel{(a)}{=} \sum_{k \in \mathbb{Z}} \left( \int_{\tau \in \mathbb{R}} x(\tau) \varphi_{\ell,k}(\tau) d\tau \right) \varphi_{\ell,k}(t), \\ &\stackrel{(b)}{=} \sum_{k \in \mathbb{Z}} \langle x, \varphi_{\ell,k} \rangle \varphi_{\ell,k}(t) \stackrel{(c)}{=} \sum_{k \in \mathbb{Z}} \alpha_k^{(\ell)} \varphi_{\ell,k}(t), \end{aligned} \quad (6.74a)$$

where (a) follows from (6.16b); (b) from the definition of the inner product; and in (c) we introduced  $\alpha_k^{(\ell)} = \langle x, \varphi_{\ell,k} \rangle$ .

Because of the finite-length assumption of  $x(t)$ , the sequence  $\alpha_k^{(\ell)}$  is also of finite length (of degree  $2^{J-\ell}$ ). Since  $x(t)$  is of norm 1 and the approximation in (6.74a) is a projection,  $\|\alpha_k^{(\ell)}\| \leq 1$ . Thus, we can apply Theorem 3.2 and represent the sequence  $\alpha_k^{(\ell)}$  by discrete Haar wavelets only

$$\alpha_k^{(\ell)} = \sum_{n \in \mathbb{Z}} \sum_{i \in \mathbb{Z}^+} \beta_n^{(i)} h_{k-2^i n}^{(i)}.$$

Since the expression (6.74a) is the piecewise-constant interpolation of the sequence  $\alpha_k^{(\ell)}$ , together with proper scaling and normalization, by linearity, we can apply this interpolation to the discrete Haar wavelets used to represent  $\alpha_k^{(\ell)}$ , which leads

<sup>51</sup>Both of these restrictions are inconsequential; the former because a general function can be decomposed into a function nonzero on  $t > 0$  and  $t \leq 0$ , the latter because the fraction of the energy of  $x(t)$  outside of the interval under consideration can be made arbitrarily small by making  $J$  arbitrarily large.

to a continuous-time Haar wavelet representation of  $x^{(\ell)}(t)$ :

$$\begin{aligned} x^{(\ell)}(t) &= \sum_{k \in \mathbb{Z}} \alpha_k^{(\ell)} \varphi_{\ell,k}(t), \\ &= \sum_{k \in \mathbb{Z}} \sum_{n \in \mathbb{Z}} \sum_{i \in \mathbb{Z}^+} \beta_n^{(i)} h_{k-2^i n}^{(i)} \varphi_{\ell,k}(t), \\ &= \sum_{n \in \mathbb{Z}} \sum_{i \in \mathbb{Z}^+} \beta_n^{(i)} \psi_{\ell+i, n 2^i}(t). \end{aligned}$$

This last statement follows from  $h^{(i)}$  being of length  $2^i$ . Thus, for a fixed  $n$  and  $i$ ,  $\sum_{k \in \mathbb{Z}} h_{k-2^i n}^{(i)} \varphi_{\ell,k}(t)$  will equal the Haar wavelet of length  $2^i 2^\ell = 2^{i+\ell}$  at shift  $n 2^i$ . Again by Theorem 3.2, this representation is exact.

What remains to be shown is that  $x^{(\ell)}(t)$  can be made arbitrarily close, in  $\mathcal{L}^2$  norm, to  $x(t)$ . This is achieved by letting  $\ell \rightarrow -\infty$  and using the fact that piecewise-constant functions are dense in  $\mathcal{L}^2(\mathbb{R})$ ;<sup>52</sup> we get

$$\lim_{i \rightarrow -\infty} \|x(t) - x^{(\ell)}(t)\| = 0.$$

TBD: Might be expanded.

The proof once more shows the intimate relation between the DWT from Chapter 3 and the wavelet series from this chapter.

**Definition** We can now formally define the wavelet series:

**DEFINITION 6.7 (WAVELET SERIES)** The wavelet series of a function  $x(t)$  is a function of  $\ell, k \in \mathbb{Z}$  given by

$$\beta_k^{(\ell)} = \langle x, \psi_{\ell,k} \rangle = \int_{-\infty}^{\infty} x(t) \psi_{\ell,k}(t) dt, \quad \ell, k \in \mathbb{Z}, \quad (6.75a)$$

with  $\psi_{\ell,k}(t)$  the prototype wavelet. The inverse wavelet series is given by

$$x(t) = \sum_{\ell \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} \beta_k^{(\ell)} \psi_{\ell,k}(t). \quad (6.75b)$$

In the above,  $\beta^{(\ell)}$  are the *wavelet coefficients*.

To denote such a wavelet series pair, we write:

$$x(t) \xleftrightarrow{\text{WS}} \beta_k^{(\ell)}.$$

We derived such bases already and we will see other constructions when we talk about multiresolution analysis.

<sup>52</sup> That is, any  $\mathcal{L}^2$  function can be approximated arbitrarily closely by a piecewise-constant function over intervals that tend to 0. This is a standard result but technical, and thus we just use it without proof.

### 6.3.2 Properties of the Wavelet Series

We now consider some of the properties of the wavelet series. Many follow from the properties of the wavelet (Section 6.2.3) or of the DWT (Chapter 3), and thus our treatment will be brief.

**Linearity** The wavelet series operator is a linear operator, or,

$$ax(t) + by(t) \xleftrightarrow{\text{WS}} a\beta_k^{(\ell)} + b\beta_k^{(\ell)}. \quad (6.76)$$

**Shift in Time** A shift in time by  $2^m n$ ,  $m, n \in \mathbb{Z}$ , results in

$$x(t - 2^m n) \xleftrightarrow{\text{WS}} \beta_{k-2^m n}^{(\ell)}, \quad \ell \leq m. \quad (6.77)$$

This is a restrictive condition as it holds only for scales smaller than  $m$ . In other words, only a function  $x(t)$  that has a scale-limited expansion, that is, it can be written as

$$x(t) = \sum_{\ell=-\infty}^m \sum_{k \in \mathbb{Z}} \beta_k^{(\ell)} \psi_{\ell,k}(t),$$

will possess the shift-in-time property for all (of its existing) scales. This is a counterpart to the shift-in-time property of the DWT, (3.17), and the fact that the DWT is periodically shift varying.

**Scaling in Time** Scaling in time by  $2^{-m}$ ,  $m \in \mathbb{Z}$ , results in

$$x(2^{-m}t) \xleftrightarrow{\text{WS}} 2^{m/2} \beta_k^{(\ell-m)}. \quad (6.78)$$

**Parseval's Equality** The wavelet series operator is a unitary operator and thus preserves the Euclidean norm (see (2.53)):

$$\|x\|^2 = \int_{-\infty}^{\infty} |x(t)|^2 dt = \sum_{\ell \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} |\beta_k^{(\ell)}|^2. \quad (6.79)$$

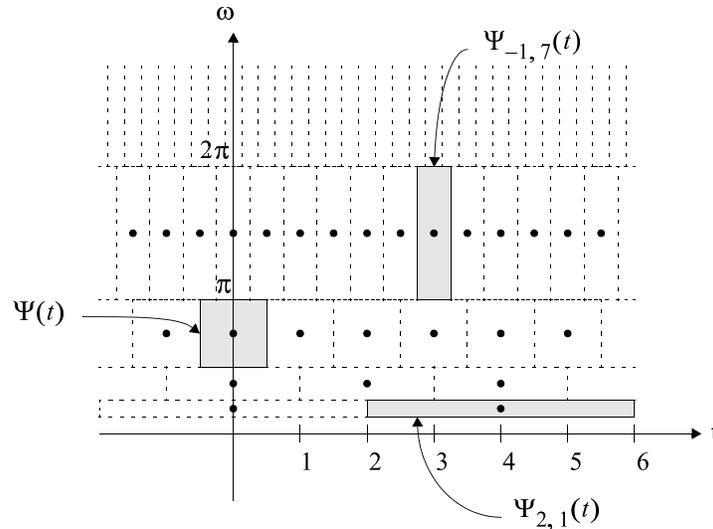
**Time-Frequency Localization** Assume that the wavelet  $\psi(t)$  is centered around  $t = 0$  in time and  $\omega = 3\pi/4$  in frequency (that is, it is a bandpass filter with the support of approximately  $[\pi/2, \pi]$ ). Then, from (6.70a),  $\psi_{\ell,0}(t)$  is centered around  $\omega = 2^{-\ell}(3\pi/4)$  in frequency (see Figure 6.29).

With our assumption of  $g$  being a causal FIR filter of length  $L$ , the support in time of the wavelets is easy to characterize. Since the support of  $\psi(t)$  is  $[0, L-1]$ ,

$$\text{support}(\psi_{\ell,k}(t)) \subseteq [2^\ell k, 2^\ell(k+L-1)]. \quad (6.80)$$

Because of the FIR assumption, the frequency localization is less precise (no compact support in frequency), but the center frequency is around  $2^{-\ell}(3\pi/4)$  and the passband is mostly in an octave band,

$$\text{support}(\Psi_{\ell,k}(\omega)) \sim [2^{-\ell} \frac{\pi}{2}, 2^{-\ell} \pi]. \quad (6.81)$$



**Figure 6.29:** Time-frequency localization of wavelet basis functions. Three wavelets are highlighted: at scale  $\ell = 0$ ,  $\psi(t)$ ; at scale  $\ell = -1$ , a higher-frequency wavelet  $\psi_{-1,7}(t)$ ; and at scale  $\ell = 2$ , a lower-frequency wavelet  $\psi_{2,1}(t)$ . These are centered along the dyadic sampling grid  $[2^\ell k - 2^{-\ell}(3\pi/4)]$ , for  $\ell, k \in \mathbb{Z}$ .

**Characterization of Singularities** As we have seen with the example of Haar wavelet series (see Figure 6.8), one of the powerful features of the wavelet series is its ability to characterize both the position and type of singularities present in a function.

Consider a function with the simplest singularity, a Dirac delta function at a location  $t_0$ , that is,  $x(t) = \delta(t - t_0)$ . At scale  $\ell$ , only wavelets having their support (6.80) straddling  $t_0$  will produce nonzero coefficients,

$$\beta_k^{(\ell)} \neq 0 \quad \text{for} \quad [t_0/2^\ell] - L < k \leq [t_0/2^\ell]. \quad (6.82)$$

Thus, there are  $L$  nonzero coefficients at each scale. These coefficients correspond to a region of size  $2^\ell(L-1)$  around  $t_0$ , or, as  $\ell \rightarrow -\infty$ , they focus arbitrarily closely on the singularity. What about the size of the coefficients at scale  $\ell$ ? The inner product of the wavelet with a Dirac delta function simply picks out a value of the wavelet. Because of the scaling factor  $2^{-\ell/2}$  in (6.70a), the nonzero coefficients will be of order

$$|\beta_k^{(\ell)}| \sim O(2^{-\ell/2}) \quad (6.83)$$

for the range of  $k$  in (6.82). That is, as  $\ell \rightarrow -\infty$ , the nonzero wavelet series coefficients zoom in onto the discontinuity, and they grow at a specific rate given by (6.83). An example for the Haar wavelet was shown in Figure 6.8.

Generalizing the Dirac delta function singularity, a function is said to have an  $n$ th-order singularity at  $t_0$  when its  $n$ th-order derivative has a Dirac delta function

component at  $t_0$ . The scaling (6.83) for a zeroth-order singularity is an example of the following result:

**THEOREM 6.8 (SCALING BEHAVIOR AROUND SINGULARITIES)** Given a wavelet  $\psi(t)$  with  $N$  zero moments, around a singularity of order  $n$ ,  $0 \leq n \leq N$ , the wavelet series coefficients  $\beta_k^{(\ell)}$  behave as

$$\left| \beta_k^{(\ell)} \right| \sim O(2^{\ell(n-1/2)}), \quad \ell \rightarrow -\infty. \quad (6.84)$$

*Proof.* We have analyzed  $n = 0$  earlier. We now give a proof for  $n = 1$ ; generalizing to  $n > 1$  is the topic of Exercise 6.3.

Assume the wavelet has at least one zero moment,  $N \geq 1$ . A function with a first-order singularity at  $t_0$  looks like a Heaviside function (4.7) locally (at  $t_0$ ). We can reduce the analysis to  $n = 0$  by considering the derivative  $x'(t)$ , which is a Dirac delta function at  $t_0$ . We use the fact that  $\psi$  has at least one zero moment and is of finite support. Then, as in (6.34), using integration by parts,

$$\begin{aligned} \langle x(t), \psi(t) \rangle_t &= \int_{-\infty}^{\infty} \psi(t)x(t) dt = - \int_{-\infty}^{\infty} \theta(t)x'(t) dt \\ &= -\langle x'(t), \theta(t) \rangle_t \\ \langle x(t), \psi_{\ell,k}(t) \rangle_t &= -\langle x'(t), \theta_{\ell,k}(t) \rangle_t, \end{aligned}$$

where  $\theta(t) = \int_{-\infty}^t \psi(\tau) d\tau$  is the primitive of  $\psi(t)$ ,  $\theta_{\ell,k}(t)$  is the primitive of  $\psi_{\ell,k}(t)$ , and  $x'(t)$  is the derivative of  $x(t)$ . Because  $\psi(t)$  has at least one zero at  $\omega = 0$  and is of finite support, its primitive is well defined and also of finite support. The key is now the scaling behavior of  $\theta_{\ell,k}(t)$  with respect to  $\theta(t)$ . Evaluating

$$\theta_{\ell,k}(t) = \int_{-\infty}^t 2^{-\ell/2} \psi(2^{-\ell}\tau - k) d\tau = 2^{\ell/2} \int_{-\infty}^{2^{-\ell}t - k} \psi(t') dt' = 2^{\ell/2} \theta(2^{-\ell}t - k),$$

we see that this scaling is given by  $2^{\ell/2}$ . Therefore, the wavelet coefficients scale as

$$\begin{aligned} \left| \beta_k^{(\ell)} \right| &= |\langle x(t), \psi_{\ell,k}(t) \rangle_t| = |-\langle x'(t), \theta_{\ell,k}(t) \rangle_t| \\ &\sim 2^{\ell/2} |\langle \delta(t - t_0), \theta(2^{-\ell}t - k) \rangle_t| \sim O(2^{\ell/2}), \end{aligned} \quad (6.85)$$

at fine scales and close to  $t_0$ .

**Zero-Moment Property** When the lowpass filter  $g$  has  $N$  zeros at  $\omega = \pi$ , we verified that  $\psi(t)$  has  $N$  zero moments (6.63). This property carries over to all scaled versions of  $\psi(t)$ , and thus, for any polynomial function  $p(t)$  of degree smaller than  $N$ ,

$$\beta_k^{(\ell)} = \langle p(t), \psi_{\ell,k}(t) \rangle_t = 0.$$

This allows us to prove the following result:

**THEOREM 6.9 (DECAY OF WAVELET SERIES COEFFICIENTS FOR  $x \in \mathbb{C}^N$ )** For a function  $x(t)$  with  $N$  continuous and bounded derivations, that is,  $x \in C^N$ , the wavelet series coefficients decay as

$$\left| \beta_k^{(\ell)} \right| \leq \alpha 2^{mN}$$

for some constant  $\alpha > 0$  and  $m \rightarrow -\infty$ .

*Proof.* Consider the Taylor series expansion of  $x(t)$  around some point  $t_0$ . Since  $x(t)$  has  $N$  continuous derivatives,

$$\begin{aligned} x(t_0 + \varepsilon) &= x(t_0) + \frac{x'(t_0)}{1!} \varepsilon + \frac{x''(t_0)}{2!} \varepsilon^2 + \cdots + \frac{x^{(N-1)}(t_0)}{(N-1)!} \varepsilon^{N-1} + R_N(\varepsilon), \\ &= p(t) + R_N(\varepsilon), \end{aligned}$$

where

$$|R_N(\varepsilon)| \leq \frac{\varepsilon^N}{N!} \sup_{t_0 \leq t \leq t_0 + \varepsilon} |x^{(N)}(t)|,$$

and we view it as a polynomial  $p(t)$  of degree  $(N-1)$  and a remainder  $R_N(\varepsilon)$ . Because of the zero-moment property of the wavelet,

$$\left| \beta_k^{(\ell)} \right| = |\langle x(t), \psi_{m,n}(t) \rangle| = |\langle p(t) + R_N(\varepsilon), \psi_{m,n}(t) \rangle| = |\langle R_N(\varepsilon), \psi_{m,n}(t) \rangle|,$$

that is, the inner product with the polynomial term is zero, and only the remainder matters. To minimize the upper bound on  $|\langle R_N(\varepsilon), \psi_{m,n} \rangle|$ , we want  $t_0$  close to the center of the wavelet. Since the spacing of the sampling grid at scale  $\ell$  is  $2^\ell$ , we see that  $\varepsilon$  is at most  $2^\ell$  and thus  $|\langle R_N(\varepsilon), \psi_{\ell,k} \rangle|$  has an upper bound of order  $2^{\ell N}$ .

A stronger result, in which  $N$  is replaced by  $N + 1/2$ , follows from Theorem 6.17 in the context of the continuous wavelet transform.

### 6.3.3 Multiresolution Analysis

We have already introduced the concept of multiresolution analysis with the Haar scaling function and wavelet in Section 6.1. As opposed to having a discrete-time filter and constructing a continuous-time basis from it, multiresolution analysis does the opposite: it starts from the multiresolution spaces to build the wavelet series. For example, we saw that the continuous-time wavelet basis generated a partition of  $\mathcal{L}^2(\mathbb{R})$  into a sequence of nested spaces

$$\dots \subset V^{(2)} \subset V^{(1)} \subset V^{(0)} \subset V^{(-1)} \subset V^{(-2)} \subset \dots,$$

and that these spaces were all scaled copies of each other, that is,  $V^{(\ell)}$  is  $V^{(0)}$  scaled by  $2^\ell$ . We will turn the question around and ask: assuming we have a sequence of nested and scaled spaces as above, does it generate a discrete-time filter bank? The

answer is yes; the framework is multiresolution analysis we have seen in the Haar case. We present it shortly in its more general form, starting with the axiomatic definition and followed by examples.

The embedded spaces above are very natural for piecewise-polynomial functions over uniform intervals of length  $2^\ell$ . For example, the Haar case leads to piecewise-constant functions. The next higher order is for piecewise-linear functions, and so on. The natural bases for such spaces are  $B$ -splines we discussed in *Chapter 6*; these are not orthonormal bases, requiring the use of orthogonalization methods.

**Axioms of Multiresolution Analysis** We now summarize the fundamental characteristics of the spaces and basis functions seen in the Haar case. These are also the axioms of multiresolution analysis.

- (i) *Embedding*: We work with a sequence of embedded spaces

$$\dots \subset V^{(2)} \subset V^{(1)} \subset V^{(0)} \subset V^{(-1)} \subset V^{(-2)} \subset \dots, \quad (6.86a)$$

where  $V^{(\ell)}$  is the space of piecewise-constant functions over  $[2^\ell k, 2^\ell(k+1))_{k \in \mathbb{Z}}$  with finite  $\mathcal{L}^2$  norm. We call the  $V^{(\ell)}$ s *successive approximation spaces*, since as  $\ell \rightarrow -\infty$ , we get finer and finer approximations.

- (ii) *Upward Completeness*: Since piecewise-constant functions over arbitrarily-short intervals are dense in  $\mathcal{L}^2$  (see Footnote 52),

$$\lim_{\ell \rightarrow -\infty} V^{(\ell)} = \overline{\bigcup_{\ell \in \mathbb{Z}} V^{(\ell)}} = \mathcal{L}^2(\mathbb{R}). \quad (6.86b)$$

- (iii) *Downward Completeness*: As  $\ell \rightarrow \infty$ , we get coarser and coarser approximations. Given a function  $x(t) \in \mathcal{L}^2(\mathbb{R})$ , its projection onto  $V^{(\ell)}$  tends to zero as  $\ell \rightarrow \infty$ , since we lose all the details. More formally,

$$\bigcap_{\ell \in \mathbb{Z}} V^{(\ell)} = \{0\}. \quad (6.86c)$$

- (iv) *Scale Invariance*: The spaces  $V^{(\ell)}$  are just scaled versions of each other,

$$x(t) \in V^{(\ell)} \Leftrightarrow x(2^m t) \in V^{(\ell-m)}. \quad (6.86d)$$

- (v) *Shift Invariance*: Because  $x(t)$  is a piecewise-constant function over intervals  $[2^\ell k, 2^\ell(k+1))$ , it is invariant to shifts by multiples of  $2^\ell$ ,

$$x(t) \in V^{(\ell)} \Leftrightarrow x(t - 2^\ell k) \in V^{(\ell)}. \quad (6.86e)$$

- (vi) *Existence of a Basis*: There exists  $\varphi(t) \in V^{(0)}$  such that

$$\{\varphi(t - k)\}_{k \in \mathbb{Z}} \quad (6.86f)$$

is a basis for  $V^{(0)}$ .

The above six characteristics, which naturally generalize the Haar multiresolution analysis, are the defining characteristics of a broad class of wavelet systems.

**DEFINITION 6.10 (MULTIRESOLUTION ANALYSIS)** A sequence  $\{V^{(\ell)}\}_{m \in \mathbb{Z}}$  of subspaces of  $\mathcal{L}^2(\mathbb{R})$  satisfying (6.86a)–(6.86f) is called a multiresolution analysis. The spaces  $V^{(\ell)}$  are called the successive approximation spaces, while the spaces  $W^{(\ell)}$ , defined as the orthogonal complements of  $V^{(\ell)}$  in  $V^{(\ell-1)}$ , that is,

$$V^{(\ell-1)} = V^{(\ell)} \oplus W^{(\ell)}, \quad (6.87)$$

are called the successive detail spaces.

**Definition** For simplicity, we will assume the basis in (6.86f) to be orthonormal; we cover the general case in Solved Exercise ??.

The two-scale equation (6.48) follows naturally from the scale-invariance axiom ((iv)). What can we say about the coefficients  $g_n$ ? Evaluate

$$\begin{aligned} \delta_k &\stackrel{(a)}{=} \langle \varphi(t), \varphi(t-k) \rangle_t \stackrel{(b)}{=} 2 \sum_{n \in \mathbb{Z}} \sum_{m \in \mathbb{Z}} g_n g_m \langle \varphi(2t-n), \varphi(2t-2k-m) \rangle_t \\ &\stackrel{(c)}{=} \sum_{n \in \mathbb{Z}} g_n g_{n-2k}, \end{aligned}$$

where (a) is true by assumption; in (b) we substituted the two-scale equation 6.48 for both  $\varphi(t)$  and  $\varphi(t-k)$ ; and (c) follows from  $\langle \varphi(2t-n), \varphi(2t-2k-m) \rangle_t = 0$  except for  $n = 2k + m$  when it is  $1/2$ . We thus conclude that the sequence  $g_n$  corresponds to an orthogonal filter (1.13). Assuming that the Fourier transform  $\Phi(\omega)$  of  $\varphi(t)$  is continuous and satisfies<sup>53</sup>

$$|\Phi(0)| = 1,$$

it follows from the two-scale equation in the Fourier domain that

$$|G(1)| = \sqrt{2},$$

making  $g_n$  a lowpass sequence. Assume it to be of finite length  $L$  and derive the equivalent highpass filter using (1.24). Defining the wavelet as in (6.59), we have:

**THEOREM 6.11** The wavelet given by (6.59) satisfies

$$\begin{aligned} \langle \psi(t), \psi(t-n) \rangle_t &= \delta_n, \\ \langle \psi(t), \varphi(t-n) \rangle_t &= 0, \end{aligned}$$

and  $W^{(0)} = \text{span}(\{\psi(t-n)\}_{n \in \mathbb{Z}})$  is the orthogonal complement of  $V^{(0)}$  in  $V^{(-1)}$ ,

$$V^{(-1)} = V^{(0)} \oplus W^{(0)}. \quad (6.88)$$

<sup>53</sup>If  $\varphi(t)$  is integrable, this follows from upward completeness (6.86b) for example.

We do not prove the theorem but rather just discuss the outline of a proof. The orthogonality relations follow from the orthogonality of the sequences  $g_n$  and  $h_n$  by using the two-scale equations (6.48) and (6.59). That  $\{\psi(t - n)\}_{n \in \mathbb{Z}}$  is an orthonormal basis for  $W^{(0)}$  requires checking completeness and is more technical. By construction, and in parallel to (6.86d),  $W^{(\ell)}$  are just scaled versions of each other,

$$x(t) \in W^{(\ell)} \Leftrightarrow x(2^m t) \in W^{(\ell-m)}. \quad (6.89)$$

Putting all the pieces above together, we have:

**THEOREM 6.12 (WAVELET BASIS FOR  $\mathcal{L}^2(\mathbb{R})$ )** Given a multiresolution analysis of  $\mathcal{L}^2(\mathbb{R})$  from Definition 6.10, the family

$$\psi_{\ell,k}(t) = \frac{1}{2^{\ell/2}} \psi\left(\frac{t - 2^\ell k}{2^\ell}\right) \quad \ell, k \in \mathbb{Z},$$

with  $\psi(t)$  as in (6.59), is an orthonormal basis for  $\mathcal{L}^2(\mathbb{R})$ .

*Proof.* Scaling (6.88) using (6.86d), we get that  $V^{(\ell)} = V^{(\ell+1)} \oplus W^{(\ell+1)}$ . Iterating it  $n$  times leads to

$$V^{(\ell)} = W^{(\ell+1)} \oplus W^{(\ell+2)} \oplus \dots \oplus W^{(\ell+n)} \oplus V^{(\ell+n)}.$$

As  $n \rightarrow \infty$  and because of (6.86c), we get<sup>54</sup>

$$V^{(\ell)} = \bigoplus_{i=\ell+1}^{\infty} W^{(i)},$$

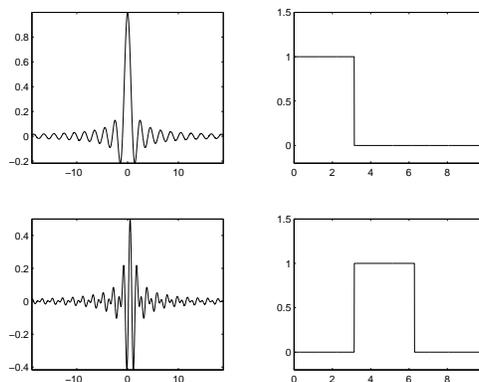
and finally, letting  $\ell \rightarrow -\infty$  and because of (6.86b), we obtain

$$\mathcal{L}^2(\mathbb{R}) = \bigoplus_{\ell \in \mathbb{Z}} W^{(\ell)}. \quad (6.90)$$

Since  $\{\psi(t - k)\}_{k \in \mathbb{Z}}$  is an orthonormal basis for  $W^{(0)}$ , by scaling,  $\{\psi_{\ell,k}(t)\}_{k \in \mathbb{Z}}$  is an orthonormal basis for  $W^{(\ell)}$ . Then, following (6.90), the family  $\{\psi_{\ell,k}(t)\}_{\ell, n \in \mathbb{Z}}$  is an orthonormal basis for  $\mathcal{L}^2(\mathbb{R})$ .

Thus, in a fashion complementary to Section 6.1, we obtain a split of  $\mathcal{L}^2(\mathbb{R})$  into a collection  $\{W^{(\ell)}\}_{\ell \in \mathbb{Z}}$  as a consequence of the axioms of multiresolution analysis (6.86a)–(6.86f) (see Figure 6.10 for a graphical representation of the spaces  $V^{(\ell)}$  and  $W^{(\ell)}$ ). We illustrate our discussion with examples.

<sup>54</sup>In the infinite sum, we imply closure.



**Figure 6.30:** Sinc scaling function and wavelet. (a) Scaling function  $\varphi(t)$ . (b) Magnitude Fourier transform  $|\Phi(\omega)|$ . (c) Wavelet  $\psi(t)$ . (d) Magnitude Fourier transform  $|\Psi(\omega)|$ .

### Examples

**EXAMPLE 6.5 (SINC MULTIREOLUTION ANALYSIS)** Let  $V^{(0)}$  be the space of  $\mathcal{L}^2$  functions bandlimited to  $[-\pi, \pi)$ , for which we know that

$$\varphi(t) = \frac{\sin(\pi t)}{\pi t} \quad (6.91)$$

and its integer shifts form an orthonormal basis. Define  $V^{(\ell)}$  to be the space of  $\mathcal{L}^2$  functions bandlimited to  $[-2^{-\ell}\pi, 2^{-\ell}\pi)$ . These are nested spaces of bandlimited functions, which obviously satisfy (6.86a), as they do the axioms of multiresolution analysis (6.86b)–(6.86f), that is, the union of the  $V^{(\ell)}$ s is  $\mathcal{L}^2(\mathbb{R})$ , their intersection is empty, the spaces are scaled versions of each other and are shift invariant with respect to shifts by integer multiples of  $2^\ell$ . The existence of the basis we stated in (6.91). The details are left as Exercise ??, including the derivation of the wavelet and the detail spaces  $W^{(\ell)}$ , the spaces of  $\mathcal{L}^2$  bandpass functions,

$$W^{(\ell)} = [-2^{-\ell+1}\pi, -2^{-\ell}\pi) \cup [2^{-\ell}\pi, 2^{-\ell+1}\pi). \quad (6.92)$$

Figure 6.30 shows the sinc scaling function and wavelet both in time as well as Fourier domains.

While the perfect bandpass spaces lead to a bona fide multiresolution analysis of  $\mathcal{L}^2(\mathbb{R})$ , the basis functions have slow decay in time. Since the Fourier transform is discontinuous, the tails of the scaling function and the wavelet decay only as  $O(1/t)$  (as can be seen in the sinc function (6.91)). We will see in latter examples possible remedies to this problem.

**EXAMPLE 6.6 (PIECEWISE-LINEAR MULTIREOLUTION ANALYSIS)** Let  $V^{(0)}$  be the space of continuous  $\mathcal{L}^2$  functions piecewise linear over intervals  $[k, k+1)$ , or  $x(t) \in V^{(0)}$  if  $\|x\| < \infty$  and  $x'(t)$  is piecewise constant over intervals  $[k, k+1)$ .

For simplicity, consider functions  $x(t)$  such that  $x(t) = 0$  for  $t < 0$ . Then  $x'(t)$  is specified by the sequence  $\{a_k\}$ , the slopes of  $x(t)$  over intervals  $[k, k+1)$ , for  $k \in \mathbb{N}$ . The nodes of  $x(t)$ , that is, the values at the integers, are given by

$$x(k) = \begin{cases} 0, & k \leq 0; \\ \sum_{i=0}^{k-1} a_i, & k > 0, \end{cases}$$

and the piecewise-linear function is

$$x(t) = [x(k+1) - x(k)](t - k) + x(k) = a_k(t - k) + \sum_{i=0}^{k-1} a_i \quad (6.93)$$

for  $t \in [k, k+1)$  (see Figure 6.31).

The spaces  $V^{(\ell)}$  are simply scaled versions of  $V^{(0)}$ ; they contain functions that are continuous and piecewise linear over intervals  $[2^\ell k, 2^\ell(k+1))$ . Let us verify the axioms of multiresolution.

- (i) *Embedding*: Embedding as in (6.86a) is clear.
- (ii) *Upward Completeness*: Similarly to the piecewise-constant case, piecewise-linear functions are dense in  $\mathcal{L}^2(\mathbb{R})$  (see Footnote 52), and thus upward completeness (6.86b) holds.
- (iii) *Downward Completeness*: Conversely, as  $\ell \rightarrow \infty$ , the approximation gets coarser and coarser, ultimately verifying downward completeness (6.86c).
- (iv) *Scale Invariance*: Scaling (6.86d) is clear from the definition of the piecewise-linear functions over intervals scaled by powers of 2.
- (v) *Shift Invariance*: Similarly, shift invariance (6.86e) is clear from the definition of the piecewise linear functions over intervals scaled by powers of 2.
- (vi) *Existence of a Basis*: What remains is to find a basis for  $V^{(0)}$ . As an educated guess, take the triangle function from (4.45) shifted by 1 to the right and call it  $\theta(t)$  (see Figure 6.32(a)). Then  $x(t)$  in (6.93) can be written as

$$x(t) = \sum_{k=0}^{\infty} b_k \theta(t - k),$$

with  $b_0 = a_0$  and  $b_k = a_k + b_{k-1}$ . We prove this as follows: First,

$$\theta'(t) = \varphi_h(t) - \varphi_h(t - 1),$$

where  $\varphi_h(t)$  is the Haar scaling function, the indicator function of the unit interval. Thus,  $x'(t)$  is piecewise constant. Then, the value of the constant between  $k$  and  $k+1$  is  $(b_k - b_{k-1})$  and thus equals  $a_k$  as desired. The only detail is that  $\theta(t)$  is clearly not orthogonal to its integer translates, since

$$\langle \theta(t), \theta(t - k) \rangle_t = \begin{cases} 2/3, & k = 0; \\ 1/6, & k = -1, 1; \\ 0, & \text{otherwise.} \end{cases}$$

We can apply the orthogonalization procedure in (??). The  $z$ -transform of the sequence  $[1/6 \ 2/3 \ 1/6]$  is

$$\frac{1}{6}(z + 4 + z^{-1}) \stackrel{(a)}{=} \frac{2 + \sqrt{3}}{6} \underbrace{[1 + (2 - \sqrt{3})z]}_{\text{right sided}} \underbrace{[1 + (2 - \sqrt{3})z^{-1}]}_{\text{left sided}},$$

where (a) follows from it being a deterministic autocorrelation, positive on the unit circle, and could thus be factored into its spectral roots. Choosing just the right-sided part, with the impulse response

$$\alpha_k = \sqrt{\frac{2 + \sqrt{3}}{6}} (-1)^k (2 - \sqrt{3})^k,$$

leads to

$$\varphi_c(t) = \sum_{k=0}^{\infty} \alpha_k \theta(t - k),$$

a function such that  $\varphi_c(t) = 0$  for  $t < 0$  and orthonormal to its integer translates. It is piecewise linear over integer pieces, but of infinite extent (see Figure 6.32).

Instead of the spectral factorization, we can just take the square root as in (??). In Fourier domain,

$$\frac{2}{3} + \frac{1}{6} e^{j\omega} + \frac{1}{6} e^{-j\omega} = \frac{1}{3} (2 + \cos(\omega)).$$

Then,

$$\Phi_s(\omega) = \frac{\sqrt{3} \theta(\omega)}{(2 + \cos(\omega))^{1/2}}$$

is the Fourier transform of a symmetric and orthogonal scaling function  $\varphi_s(t)$  (see Figure 6.32(c)).

Because of the embedding of the spaces  $V^{(\ell)}$ , the scaling functions all satisfy two-scale equations (Exercise ??). Once the two-scale equation coefficients are derived, the wavelet can be calculated in the standard manner. Naturally, since the wavelet is a basis for the orthogonal complement of  $V^{(0)}$  in  $V^{(-1)}$ , it will be piecewise linear over half-integer intervals (Exercise ??).

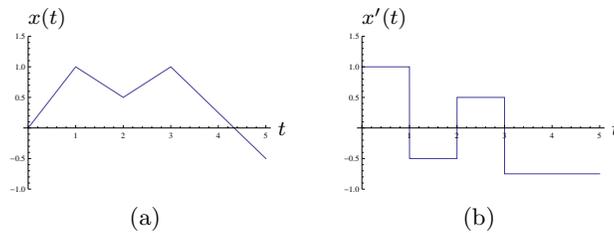
**EXAMPLE 6.7 (MEYER MULTIREOLUTION ANALYSIS)** The idea behind Meyer's wavelet construction is to smooth the sinc solution in Fourier domain, so as to obtain faster decay of the basis functions in the time domain. The simplest way to do this is to allow the Fourier transform magnitude of the scaling function,  $|\Phi(\omega)|^2$ , to linearly decay to zero, that is,

$$|\Phi(\omega)|^2 = \begin{cases} 1, & |\omega| < \frac{2\pi}{3}; \\ 2 - \frac{3|\omega|}{2\pi}, & \frac{2\pi}{3} < |\omega| < \frac{4\pi}{3}. \end{cases} \quad (6.94)$$

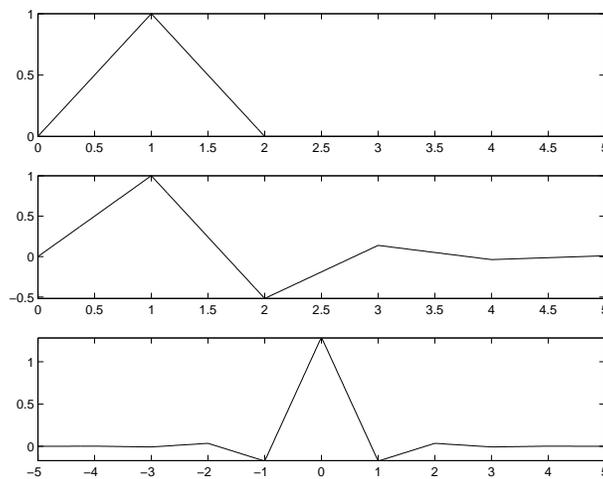
We start by defining a function orthonormal to its integer translates and the space  $V^{(0)}$  spanned by those, axiom (vi).

## 6.3. Wavelet Series

237



**Figure 6.31:** (a) A continuous and piecewise-linear function  $x(t)$  and (b) its derivative  $x'(t)$ .

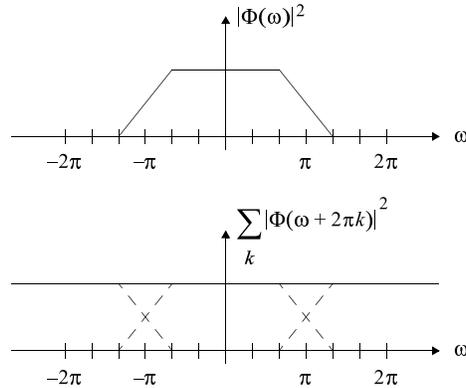


**Figure 6.32:** Basis function for piecewise-linear spaces. (a) The nonorthogonal basis function  $\theta(t)$ . (b) An orthogonalized basis function  $\varphi_c(t)$  such that  $\varphi_c(t) = 0$  for  $t < 0$ . (c) An orthogonalized symmetric basis function  $\varphi_s(t)$ .

- (i) *Existence of a Basis:* The basis function is shown in Figure 6.33, where we also show graphically that (??) holds, proving that  $\{\varphi(t - k)\}_{k \in \mathbb{Z}}$  is an orthonormal set. We now define  $V^{(0)}$  to be

$$V^{(0)} = \text{span}(\{\varphi(t - k)\}_{k \in \mathbb{Z}}).$$

- (ii) *Upward Completeness:* Define  $V^{(\ell)}$  as the scaled version of  $V^{(0)}$ . Then (6.86b) holds, similarly to the sinc case.
- (iii) *Downward Completeness:* Again, (6.86c) holds.
- (iv) *Scale Invariance:* Holds by construction.
- (v) *Shift Invariance:* Holds by construction.
- (vi) *Embedding:* To check  $V^{(0)} \subset V^{(-1)}$  we use Figure 6.34 to see that  $V^{(0)}$  is perfectly represented in  $V^{(-1)}$ . This means we can find a  $2\pi$ -periodic



**Figure 6.33:** Meyer scaling function, with a piecewise linear squared Fourier transform magnitude. (a) The function  $|\Phi(\omega)|^2$ . (b) Proof of orthogonality by verifying (??).

function  $G(e^{j\omega})$  to satisfy the two-scale equation in Fourier domain (6.47), illustrated in Figure 6.35.

Now that we have verified the axioms of multiresolution analysis, we can construct the wavelet. From (6.94), (6.47) and the figure, the DTFT of the discrete-time filter  $g_n$  is

$$|G(e^{j\omega})| = \begin{cases} \sqrt{2}, & |\omega| \leq \frac{\pi}{3}; \\ \sqrt{4 - \frac{6|\omega|}{\pi}}, & \frac{\pi}{3} \leq |\omega| < \frac{2\pi}{3}; \\ 0, & \frac{2\pi}{3} < |\omega| \leq \pi. \end{cases} \quad (6.95)$$

As the phase is not specified, we chose it to be zero making  $G(e^{j\omega})$  real and symmetric. Such a filter has an infinite impulse response, and its  $z$ -transform is not rational (since it is exactly zero over an interval of nonzero measure). It does satisfy, however, the quadrature formula for an orthogonal lowpass filter from (1.13). Choosing the highpass filter in the standard way, (1.24),

$$H(e^{j\omega}) = e^{-j\omega} G(e^{j(\omega+\pi)}),$$

with  $G(e^{j\omega})$  real, and using the two-scale equation for the wavelet in Fourier domain, (6.58), we get

$$\Psi(\omega) = \begin{cases} 0, & |\omega| < \frac{2\pi}{3}; \\ e^{-j\omega} \sqrt{\frac{3\omega}{2\pi} - 1}, & \frac{2\pi}{3} \leq |\omega| < \frac{4\pi}{3}; \\ e^{-j\omega} \sqrt{2 - \frac{3\omega}{4\pi}}, & \frac{4\pi}{3} \leq |\omega| < \frac{8\pi}{3}; \\ 0, & |\omega| \geq \frac{8\pi}{3}. \end{cases} \quad (6.96)$$

The construction and resulting wavelet (a bandpass function) are shown in Figure 6.36. Finally, the scaling function  $\varphi(t)$  and wavelet  $\psi(t)$  are shown, together with their Fourier transforms, in Figure 6.37.

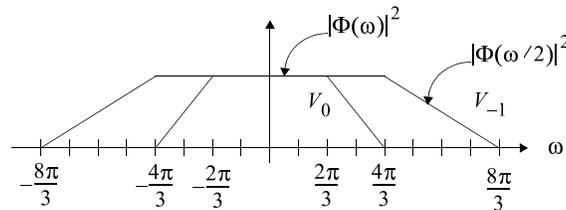


Figure 6.34: Embedding  $V^{(0)} \subset V^{(-1)}$  for the Meyer wavelet.

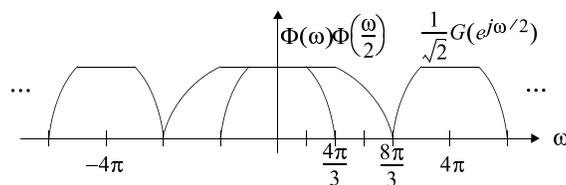


Figure 6.35: The two-scale equation for the Meyer wavelet in frequency domain. Note how the  $4\pi$ -periodic function  $G(e^{j(\omega/2+\pi)})$  carves out  $\Phi(\omega)$  from  $\Phi(\omega/2)$ .

The example above showed all the ingredients of the general construction of Meyer wavelets. The key was the orthogonality relation for  $\Phi(\omega)$ , the fact that  $\Phi(\omega)$  is continuous, and that the spaces  $V^{(\ell)}$  are embedded. Since  $\Phi(\omega)$  is continuous,  $\varphi(t)$  decays as  $O(1/t^2)$ . Smoother  $\Phi(\omega)$ 's can be constructed, leading to faster decay of  $\varphi(t)$  (Exercise ??).

### 6.3.4 Biorthogonal Wavelet Series

Instead of one scaling function and one wavelet, we now seek two scaling functions,  $\varphi(t)$  and  $\tilde{\varphi}(t)$ , as well as two corresponding wavelets,  $\psi(t)$  and  $\tilde{\psi}(t)$  as in Section 6.2.4, such that the families

$$\psi_{\ell,k}(t) = \frac{1}{\sqrt{2^\ell}} \psi\left(\frac{t - 2^\ell k}{2^\ell}\right), \tag{6.97a}$$

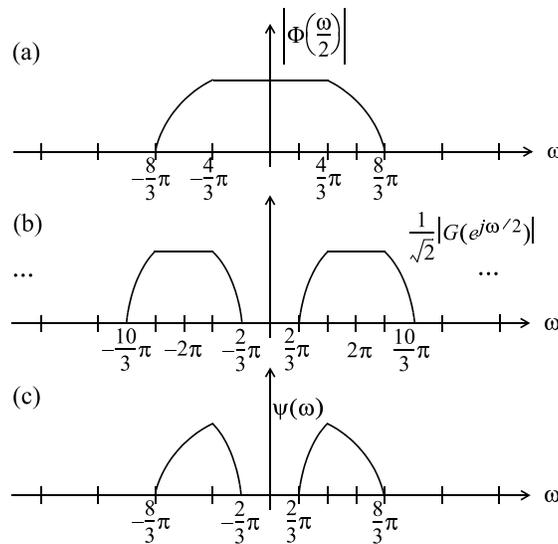
$$\tilde{\psi}_{\ell,k}(t) = \frac{1}{\sqrt{2^\ell}} \tilde{\psi}\left(\frac{t - 2^\ell k}{2^\ell}\right), \tag{6.97b}$$

for  $\ell, k, \in \mathbb{Z}$ , form a biorthogonal set

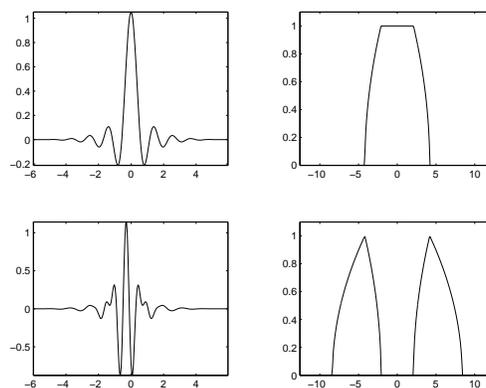
$$\langle \psi_{k,\ell}(t), \tilde{\psi}_{m,n}(t) \rangle = \delta_{n-\ell} \delta_{m-k},$$

and are complete in  $\mathcal{L}^2(\mathbb{R})$ . That is, any  $x(t) \in \mathcal{L}^2(\mathbb{R})$  can be written as either

$$x(t) = \sum_{\ell \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} \beta_k^{(\ell)} \psi_{\ell,k}(t), \quad \beta_k^{(\ell)} = \langle x, \tilde{\psi}_{\ell,k} \rangle,$$



**Figure 6.36:** Construction of the wavelet from the two-scale equation. (a) The stretched scaling function  $\Phi(\omega/2)$ . (b) The stretched and shifted lowpass filter  $G(e^{j(\omega/2+\pi)})$ . (c) The resulting bandpass wavelet  $\Psi(\omega)$ .



**Figure 6.37:** Meyer scaling function and wavelet. (a)  $\varphi(t)$ . (b)  $\Phi(\omega)$ . (c)  $\psi(t)$ . (d)  $\Psi(\omega)$ .

or

$$x(t) = \sum_{\ell \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} \tilde{\beta}_k^{(\ell)} \tilde{\psi}_{\ell,k}(t), \quad \tilde{\beta}_k^{(\ell)} = \langle x, \psi_{\ell,k} \rangle.$$

These scaling functions and wavelets will satisfy two-scale equations as before

$$\begin{aligned}\varphi(t) &= \sqrt{2} \sum_{n \in \mathbb{Z}} g_n \varphi(2t - n), & \tilde{\varphi}(t) &= \sqrt{2} \sum_{n \in \mathbb{Z}} \tilde{g}_n \tilde{\varphi}(2t - n), \\ \psi(t) &= \sqrt{2} \sum_{n \in \mathbb{Z}} h_n \varphi(2t - n), & \tilde{\psi}(t) &= \sqrt{2} \sum_{n \in \mathbb{Z}} \tilde{h}_n \tilde{\varphi}(2t - n).\end{aligned}$$

We can then define a biorthogonal multiresolution analysis by

$$V^{(0)} = \text{span}(\{\varphi(t - k)\}_{k \in \mathbb{Z}}), \quad \tilde{V}^{(0)} = \text{span}(\{\tilde{\varphi}(t - k)\}_{k \in \mathbb{Z}}),$$

and the appropriate scaled spaces

$$V^{(\ell)} = \text{span}(\{\varphi_{\ell,k}\}_{k \in \mathbb{Z}}), \quad \tilde{V}^{(\ell)} = \text{span}(\{\tilde{\varphi}_{\ell,k}\}_{k \in \mathbb{Z}}), \quad (6.98)$$

for  $\ell \in \mathbb{Z}$ . For a given  $\varphi(t)$ —for example, the triangle function—we can verify that the axioms of multiresolution analysis (Exercise ??). From there, define the wavelet families as in (6.97a)–(6.97b), which then lead to the wavelet spaces  $W^{(\ell)}$  and  $\tilde{W}^{(\ell)}$ . While this seems very natural, the geometry is more complicated than in the orthogonal case. On the one hand, we have the decompositions

$$V^{(\ell)} = V^{(\ell+1)} \oplus W^{(\ell+1)}, \quad (6.99)$$

$$\tilde{V}^{(\ell)} = \tilde{V}^{(\ell+1)} \oplus \tilde{W}^{(\ell+1)}, \quad (6.100)$$

as can be verified by using the two-scale equations for the scaling functions and wavelets involved. On the other hand, unlike the orthonormal case,  $V^{(\ell)}$  is not orthogonal to  $W^{(\ell)}$ . Instead,

$$\tilde{W}^{(\ell)} \perp V^{(\ell)}, \quad W^{(\ell)} \perp \tilde{V}^{(\ell)},$$

similarly to a biorthogonal basis (see Figure 1.11). We explore these relationships in Exercise ?? to show that

$$\tilde{W}^{(\ell)} \perp W^{(m)}, \quad \ell \neq m.$$

The embedding (6.86a) has then two forms:

$$\dots \subset V^{(2)} \subset V^{(1)} \subset V^{(0)} \subset V^{(-1)} \subset V^{(-2)} \subset \dots,$$

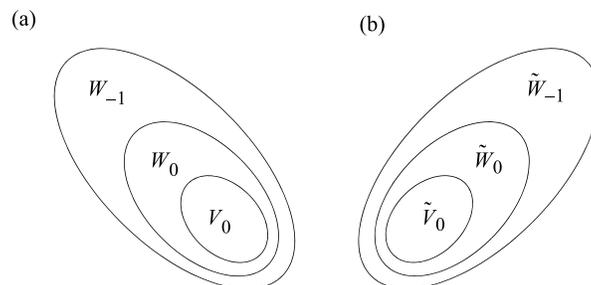
with detail spaces  $\{W^{(\ell)}\}_{\ell \in \mathbb{Z}}$ , or,

$$\dots \subset \tilde{V}^{(2)} \subset \tilde{V}^{(1)} \subset \tilde{V}^{(0)} \subset \tilde{V}^{(-1)} \subset \tilde{V}^{(-2)} \subset \dots,$$

with detail spaces  $\{\tilde{W}^{(\ell)}\}_{\ell \in \mathbb{Z}}$ . The detail spaces allow us to write

$$\mathcal{L}^2(\mathbb{R}) = \bigoplus_{\ell \in \mathbb{Z}} W^{(\ell)} = \bigoplus_{\ell \in \mathbb{Z}} \tilde{W}^{(\ell)}.$$

The diagram in Figure 6.38 illustrates these two splits and the biorthogonality between them.



**Figure 6.38:** The space  $\mathcal{L}^2(\mathbb{R})$  is split according to two different embeddings. (a) Embedding  $V^{(\ell)}$  based on the scaling function  $\varphi(t)$ . (b) Embedding  $\tilde{V}^{(\ell)}$  based on the dual scaling function  $\tilde{\varphi}(t)$ . Note that orthogonality is “across” the spaces and their duals, for example,  $\tilde{W}^{(\ell)} \perp V^{(\ell)}$ .

## 6.4 Wavelet Frame Series

### 6.4.1 Definition of the Wavelet Frame Series

### 6.4.2 Frames from Sampled Wavelet Series

## 6.5 Continuous Wavelet Transform

### 6.5.1 Definition of the Continuous Wavelet Transform

The continuous wavelet transform uses a function  $\psi(t)$  and all its shifted and scaled versions to analyze functions. Here we consider only real wavelets; this can be extended to complex wavelets without too much difficulty.

Consider a real wavelet  $\psi(t) \in \mathcal{L}^2(\mathbb{R})$  centered around  $t = 0$  and having at least one zero moment (i.e.,  $\int \psi(t) dt = 0$ ). Now, consider all its shifts and scales, denoted by

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right), \quad a \in \mathbb{R}^+, \quad b \in \mathbb{R}, \quad (6.101)$$

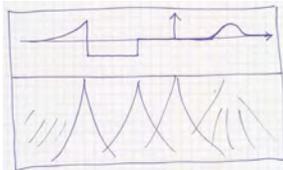
which means that  $\psi_{a,b}(t)$  is centered around  $b$  and scaled by a factor  $a$ . The scale factor  $\frac{1}{\sqrt{a}}$  insures that the  $\mathcal{L}^2$  norm is preserved, and without loss of generality, we can assume  $\|\psi\| = 1$  and thus

$$\|\psi_{a,b}\| = 1.$$

There is one more condition on the wavelet, namely the *admissibility condition* stating that the Fourier transform  $\Psi(w)$  must satisfy

$$C_\psi = \int_{\omega \in \mathbb{R}^+} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < \infty. \quad (6.102)$$

Since  $|\Psi(0)| = 0$  because of the zero moment property, this means that  $|\Psi(\omega)|$  has to decay for large  $\omega$ , which it will if  $\psi$  has any smoothness. In short, (6.102) is a very mild requirement that is satisfied by all wavelets of interest (see, for example,



**Figure 6.39:** The wavelet transform. (a) An example function. (b) The magnitude of wavelet transform  $|X(a, b)|$ .

Exercise ??). Now, given a function  $x(t)$  in  $\mathcal{L}^2(\mathbb{R})$ , we can define its continuous wavelet transform as

$$\begin{aligned} X(a, b) &= \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} \psi\left(\frac{t-b}{a}\right) x(t) dt = \int_{-\infty}^{\infty} \psi_{a,b}(t) x(t) dt \\ &= \langle f, \psi_{a,b} \rangle. \end{aligned} \quad (6.103)$$

In words, we take the inner product of the function  $x(t)$  with a wavelet centered at location  $b$ , and rescaled by a factor  $a$ , shown in Figure 6.16. A numerical example is given in Figure 6.39, which displays the magnitude  $|X(a, b)|$  as an image. It is already clear that the continuous wavelet transform acts as a singularity detector or derivative operator, and that smooth regions are suppressed, which follows from the zero moment property.

Let us rewrite the continuous wavelet transform at scale  $a$  as a convolution. For this, it will be convenient to introduce the scaled and normalized version of the wavelet,

$$\psi_a(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t}{a}\right) \quad \xleftrightarrow{\text{FT}} \quad \Psi_a(\omega) = \sqrt{a} \Psi(a\omega), \quad (6.104)$$

as well as the notation  $\bar{\psi}(t) = \psi(-t)$ . Then

$$\begin{aligned} X(a, b) &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) f(t) dt = \int_{-\infty}^{\infty} \psi_a(t-b) f(t) dt \\ &= (f * \bar{\psi}_a)(b). \end{aligned} \quad (6.105)$$

Now the Fourier transform of  $X(a, b)$  over the “time” variable  $b$  is

$$X(a, \omega) = X(\omega) \Psi_a^*(\omega) = X(\omega) \sqrt{a} \Psi^*(a\omega), \quad (6.106)$$

where we used  $\psi(-t) \xleftrightarrow{\text{FT}} \Psi^*(\omega)$  since  $\psi(t)$  is real.

### 6.5.2 Existence and Convergence of the Continuous Wavelet Transform

The invertibility of the continuous wavelet transform is of course a key result: not only can we compute the continuous wavelet transform, but we are actually able to

come back! This inversion formula was first proposed by J. Morlet.<sup>55</sup>

**THEOREM 6.13 (INVERSION OF THE CONTINUOUS WAVELET TRANSFORM)**

Consider a real wavelet  $\psi$  satisfying the admissibility condition (6.102). A function  $f \in \mathcal{L}^2(\mathbb{R})$  can be recovered from its continuous wavelet transform  $X(a, b)$  by the inversion formula

$$x(t) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty X(a, b) \psi_{a,b}(t) \frac{db da}{a^2}, \quad (6.107)$$

where equality is in the  $\mathcal{L}^2$  sense.

*Proof.* Denote the right hand side of (6.107) by  $x(t)$ . In that expression, we replace  $X(a, b)$  by (6.105) and  $\psi_{a,b}(t)$  by  $\psi_a(t-b)$  to obtain

$$\begin{aligned} x(t) &= \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty (f * \bar{\psi}_a)(b) \psi_a(t-b) \frac{db da}{a^2} \\ &= \frac{1}{C_\psi} \int_0^\infty (f * \bar{\psi}_a * \psi_a)(t) \frac{da}{a^2}, \end{aligned}$$

where the integral over  $b$  was recognized as a convolution. We will show the  $\mathcal{L}^2$  equality of  $x(t)$  and  $x(t)$  through the equality of their Fourier transforms. The Fourier transform of  $x(t)$  is

$$\begin{aligned} X(\omega) &= \frac{1}{C_\psi} \int_{-\infty}^\infty \int_0^\infty (f * \bar{\psi}_a * \psi_a)(t) e^{-j\omega t} \frac{da dt}{a^2} \\ &\stackrel{(a)}{=} \frac{1}{C_\psi} \int_0^\infty X(\omega) \Psi_a^*(\omega) \Psi_a(\omega) \frac{da}{a^2} \\ &\stackrel{(b)}{=} \frac{1}{C_\psi} X(\omega) \int_0^\infty a |\Psi(a\omega)|^2 \frac{da}{a^2}, \end{aligned} \quad (6.108)$$

where (a) we integrated first over  $t$ , and transformed the two convolutions into products; and (b) we used (6.104). In the remaining integral above, apply a change of variable  $\Omega = a\omega$  to compute:

$$\int_0^\infty |\Psi(a\omega)|^2 \frac{da}{a} = \int_0^\infty \frac{|\Psi(\Omega)|^2}{\Omega} d\Omega = C_\psi, \quad (6.109)$$

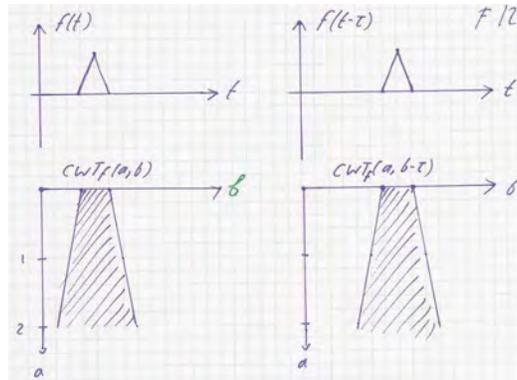
which together with (6.108), shows that  $X(\omega) = X(\omega)$ . By Fourier inversion, we have proven that  $x(t) = x(t)$  in the  $\mathcal{L}^2$  sense.

The formula (6.107) is also sometimes called the *resolution of the identity* and goes back to Calderon in the 1960's in a context other than wavelets.

### 6.5.3 Properties of the Continuous Wavelet Transform

#### Linearity

<sup>55</sup>The story goes that Morlet asked a mathematician for a proof, but only got as an answer: "This formula, being so simple, would be known if it were correct."



**Figure 6.40:** The shift property of the continuous wavelet transform.

**Shift in Time** The continuous wavelet transform has a number of properties, several of these being extensions or generalizations of properties seen already for wavelet series. Let us start with shift and scale invariance. Consider  $g(t) = x(t - \tau)$ , or a delayed version of  $x(t)$ . Then

$$\begin{aligned} X_g(a, b) &= \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} \psi\left(\frac{t-b}{a}\right) x(t-\tau) dt = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} \psi\left(\frac{t'+\tau-b}{a}\right) x(t') dt' \\ &= X_f(a, b-\tau) \end{aligned} \quad (6.110)$$

by using the change of variables  $t' = t - \tau$ . That is, the continuous wavelet transform of  $g$  is simply a delayed version of the wavelet transform of  $x(t)$ , as shown in Figure 6.40.

**Scaling in Time** For the scaling property, consider a scaled and normalized version of  $x(t)$ ,

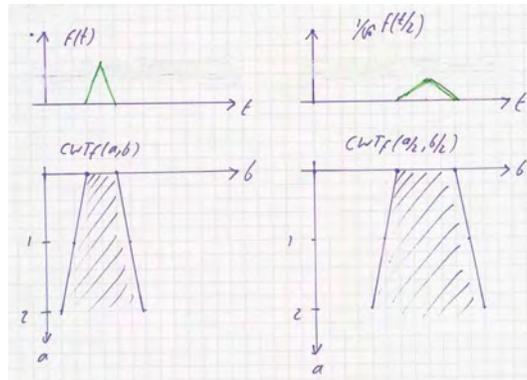
$$g(t) = \frac{1}{\sqrt{s}} f\left(\frac{t}{s}\right),$$

where the renormalization ensures that  $\|g\| = \|f\|$ . Computing the continuous wavelet transform of  $g$ , using the change of variables  $t' = t/s$ , gives

$$\begin{aligned} X_g(a, b) &= \frac{1}{\sqrt{as}} \int_{-\infty}^{\infty} \psi\left(\frac{t-b}{a}\right) f\left(\frac{t}{s}\right) dt = \frac{1}{\sqrt{as}} \int_{-\infty}^{\infty} \psi\left(\frac{st'-b}{a}\right) f(t') dt' \\ &= \sqrt{\frac{s}{a}} \int_{-\infty}^{\infty} \psi\left(\frac{t'-b/s}{a/s}\right) x(t') dt' = X\left(\frac{a}{s}, \frac{b}{s}\right). \end{aligned} \quad (6.111)$$

In words: if  $g(t)$  is a version of  $x(t)$  scaled by a factor  $s$  and normalized to maintain its energy, then its continuous wavelet transform is a scaled by  $s$  both in  $a$  and  $b$ . A graphical representation of the scaling property is shown in Figure 6.41.

Consider now a function  $x(t)$  with unit energy and having its wavelet transform concentrated mostly in a unit square, say  $[a_0, a_0 + 1] \times [b_0, b_0 + 1]$ . The continuous



**Figure 6.41:** The scaling property of the continuous wavelet transform.

wavelet transform of  $g(t)$  is then mostly concentrated in a square  $[sa_0, s(a_0 + 1)] \times [sb_0, s(b_0 + 1)]$ , a cell of area  $s^2$ . But remember that  $g(t)$  has still unit energy, while its continuous wavelet transform now covers a surface increased by  $s^2$ . Therefore, when evaluating an energy measure in the continuous wavelet transform domain, we need to renormalize by a factor  $a^2$ , as was seen in both the inversion formula (6.107) and the energy conservation formula (6.112).

When comparing the above properties with the equivalent ones from wavelet series, the major difference is that shift and scale are arbitrary real variables, rather than constrained, dyadic rationals (powers of 2 for the scale, multiples of the scale for shifts). Therefore, we obtain true time scale and shift properties.

**Parseval's Equality** Closely related to the resolution of the identity is an energy conservation formula, an analogue to Parseval's equality.

**THEOREM 6.14 (ENERGY CONSERVATION OF THE CONTINUOUS WAVELET TRANSFORM)**

Consider a function  $f \in \mathcal{L}^2(\mathbb{R})$  and its continuous wavelet transform  $X(a, b)$  with respect to a real wavelet  $\psi$  satisfying the admissibility condition (6.102). Then, the following energy conservation holds:

$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \frac{1}{C_\psi} \int_{a \in \mathbb{R}^+} \int_{b \in \mathbb{R}} |X(a, b)|^2 \frac{db da}{a^2}. \quad (6.112)$$

## 6.5. Continuous Wavelet Transform

247

*Proof.* Expand the right hand side (without the leading constant) as

$$\begin{aligned} \int_{a \in \mathbb{R}^+} \int_{b \in \mathbb{R}} |X(a, b)|^2 \frac{db da}{a^2} &\stackrel{(a)}{=} \int_{a \in \mathbb{R}^+} \int_{b \in \mathbb{R}} |(f * \bar{\psi}_a)(b)|^2 db \frac{da}{a^2} \\ &\stackrel{(b)}{=} \int_{a \in \mathbb{R}^+} \frac{1}{2\pi} \int_{b \in \mathbb{R}} |X(\omega) \sqrt{a} \Psi^*(a\omega)|^2 d\omega \frac{da}{a^2} \\ &= \int_{a \in \mathbb{R}^+} \frac{1}{2\pi} \int_{b \in \mathbb{R}} |X(\omega)|^2 |\Psi(a\omega)|^2 d\omega \frac{da}{a}, \end{aligned}$$

where (a) uses (6.105); and (b) uses Parseval's equality for the Fourier transform with respect to  $b$ , also transforming the convolution into a product. Changing the order of integration and in (c) using the change of variables  $\Omega = a\omega$  allows us to write the above as

$$\begin{aligned} \int_{a \in \mathbb{R}^+} \int_{b \in \mathbb{R}} |X(a, b)|^2 \frac{db da}{a^2} &= \int_{-\infty}^{\infty} \frac{1}{2\pi} |X(\omega)|^2 \int_{a \in \mathbb{R}^+} |\Psi(a\omega)| \frac{da}{a^2} d\omega \\ &\stackrel{(c)}{=} \int_{\omega \in \mathbb{R}} \frac{1}{2\pi} |F(\omega)|^2 \underbrace{\int_{\omega \in \mathbb{R}^+} |\Psi(\Omega)|^2 \frac{d\Omega}{\Omega}}_{C_\psi} d\omega. \end{aligned}$$

Therefore

$$\frac{1}{C_\psi} \int_{a \in \mathbb{R}^+} \int_{b \in \mathbb{R}} |X(a, b)|^2 \frac{db da}{a^2} = \frac{1}{2\pi} \int_{\omega \in \mathbb{R}} |X(\omega)|^2 d\omega,$$

and applying Parseval's equality to the right side proves (6.112).

Both the inversion formula and the energy conservation formula use  $da db/a^2$  as an integration measure. This is related to the scaling property of the continuous wavelet transform as will be shown below. Note that the extension to a complex wavelet is not hard; the integral over  $da$  has to go from  $-\infty$  to  $\infty$ , and  $C_\psi$  has to be defined accordingly.

**Redundancy** The continuous wavelet transform maps a one-dimensional function into a two-dimensional one: this is clearly very redundant. In other words, only a small subset of two-dimensional functions correspond to wavelet transforms. We are thus interested in characterizing the image of one-dimensional functions in the continuous wavelet transform domain.

A simple analogue is in order. Consider an  $M$  by  $N$  matrix  $T$  having orthonormal columns (i.e.,  $T^T T = I$ ) with  $M > N$ . Suppose  $y$  is the image of an arbitrary vector  $x \in \mathbb{R}^N$  through the operator  $T$ , or  $y = Tx$ . Clearly  $y$  belongs to a subspace  $S$  of  $\mathbb{R}^M$ , namely the span of the columns of  $T$ .

There is a simple test to check if an arbitrary vector  $z \in \mathbb{R}^M$  belongs to  $S$ . Introduce the kernel matrix  $K$ ,

$$K = TT^T, \tag{6.113}$$

which is the  $M$  by  $M$  matrix of outer products of the columns of  $T$ . Then, a vector  $z$  belong to  $S$  if and only if it satisfies

$$Kz = z. \tag{6.114}$$

Indeed, if  $z$  is in  $S$ , then it can be written as  $z = Tx$  for some  $x$ . Substituting this into the left side of (6.114) leads to

$$Kz = TT^T Tx = Tx = z.$$

Conversely, if (6.114) holds then  $z = Kz = TT^T z = Tx$ , showing that  $z$  belongs to  $S$ .

If  $z$  is not in  $S$ , then  $Kz = \hat{z}$  is the orthogonal projection of  $z$  onto  $S$  as can be verified. See Exercise ?? for a discussion of this, as well as the case of non-orthonormal columns in  $T$ .

We now extend the test given in (6.114) to the case of the continuous wavelet transform. For this, let us introduce the *reproducing kernel* of the wavelet  $\psi(t)$ , defined as

$$K(a_0, b_0, a, b) = \langle \psi_{a_0, b_0}, \psi_{a, b} \rangle. \quad (6.115)$$

This is the deterministic crosscorrelation of two wavelets at scale and shifts  $(a_0, b_0)$  and  $(a, b)$ , respectively, and is the equivalent of the matrix  $K$  in (6.113).

Call  $V$  the space of functions  $X(a, b)$  that are square integrable with respect to the measure  $(db da)/a^2$  (see also Theorem 6.14). In this space, there exists a subspace  $S$  that corresponds to bona fide continuous wavelet transforms. Similarly to what we just did in finite dimensions, we give a test to check whether a function  $X(a, b)$  in  $V$  actually belongs to  $S$ , that is, if it is the continuous wavelet transform of some one-dimensional function  $x(t)$ .

**THEOREM 6.15 (REPRODUCING KERNEL PROPERTY OF THE CONTINUOUS WAVELET TRANSFORM)**  
A function  $X(a, b)$  is the continuous wavelet transform of a function  $x(t)$  if and only if it satisfies

$$X(a_0, b_0) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty K(a_0, b_0, a, b) X(a, b) \frac{db da}{a^2}. \quad (6.116)$$

*Proof.* We show that if  $X(a, b)$  is a continuous wavelet transform of some function  $x(t)$ , then (6.116) holds. Completing the proof by showing that the converse is also true is left as Exercise ??.

By assumption,

$$X(a_0, b_0) = \int_{-\infty}^\infty \psi_{a_0, b_0}(t) x(t) dt.$$

Replace  $x(t)$  by its inversion formula (6.107), or

$$\begin{aligned} F(a_0, b_0) &= \int_{-\infty}^\infty \psi_{a_0, b_0}(t) \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty \psi_{a, b}(t) X(a, b) \frac{db da}{a^2} dt \\ &\stackrel{(a)}{=} \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty \int_{-\infty}^\infty \psi_{a_0, b_0}(t) \psi_{a, b}(t) X(a, b) dt \frac{db da}{a^2} \\ &\stackrel{(b)}{=} \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty K(a_0, b_0, a, b) X(a, b) \frac{db da}{a^2}, \end{aligned}$$

where (a) we interchanged the order of integration; and (b) we integrated over  $t$  to get the reproducing kernel (6.115).

**Characterization of Singularities** The continuous wavelet transform has an interesting localization property which is related to the fact that as  $a \rightarrow 0$ , the wavelet  $\psi_{a,b}(t)$  becomes arbitrarily narrow, performing a zoom in the vicinity of  $b$ . This is easiest to see for  $x(t) = \delta(t - \tau)$ . Then

$$X(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} \psi\left(\frac{t-b}{a}\right) \delta(t - \tau) dt = \frac{1}{\sqrt{a}} \psi\left(\frac{\tau - b}{a}\right). \quad (6.117)$$

This is the wavelet scaled by  $a$  and centered at  $b$ . As  $a \rightarrow 0$ , the continuous wavelet transform narrows exactly on the singularity and grows as  $a^{-1/2}$ .

A similar behavior can be shown for other singularities as well, which we do now. For simplicity, we consider a compactly supported wavelet with  $N$  zero moments. We have seen the most elementary case, namely the Haar wavelet (with a single zero moment) in Section 6.1. Another example is the ramp function starting at  $\tau$ :

$$x(t) = \begin{cases} 0, & t \leq \tau; \\ t - \tau, & t > \tau. \end{cases}$$

This function is continuous, but its derivative is not. Actually, its second derivative is a Dirac delta function at location  $\tau$ .

To analyze this function and its singularity, we need a wavelet with at least 2 zero moments. Given a compactly supported wavelet, its second order primitive will be compactly supported as well. To compute the continuous wavelet transform  $X(a, b)$ , we can apply integration by parts just like in (6.34) to obtain

$$X(a, b) = - \int_{-\infty}^{\infty} \sqrt{a} \theta\left(\frac{t-b}{a}\right) x'(t) dt,$$

where  $x'(t)$  is now a step function. We apply integration by parts one more time to get

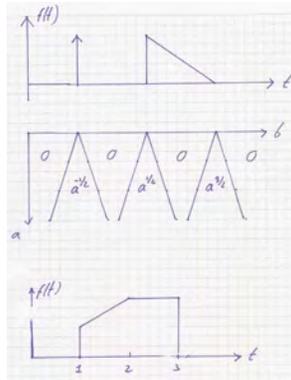
$$\begin{aligned} X(a, b) &= - \left[ a^{3/2} \theta^{(1)}\left(\frac{t-b}{a}\right) x'(t) \right]_{t \in \mathbb{R}} + a^{3/2} \int_{-\infty}^{\infty} \theta^{(1)}\left(\frac{t-b}{a}\right) x''(t) dt \\ &= a^{3/2} \int_{-\infty}^{\infty} \theta^{(1)}\left(\frac{t-b}{a}\right) \delta(t - \tau) dt = a^{3/2} \theta^{(1)}\left(\frac{\tau - b}{a}\right), \end{aligned} \quad (6.118)$$

where  $\theta^{(1)}(t)$  is the primitive of  $\theta(t)$ , and the factor  $a^{3/2}$  comes from an additional factor  $a$  due to integration of  $\theta(t/a)$ . The key, of course, is that as  $a \rightarrow 0$ , the continuous wavelet transform zooms towards the singularity and has a behavior of the order  $a^{3/2}$ . These are examples of the following general result.

**THEOREM 6.16 (LOCALIZATION PROPERTY OF THE CONTINUOUS WAVELET TRANSFORM)**

Consider a wavelet  $\psi$  of compact support having  $N$  zero moments and a function  $x(t)$  with a singularity of order  $n \leq N$  (meaning the  $n$ th derivative is a Dirac delta function; for example, Dirac delta function = 0, step = 1, ramp = 2, etc.). Then, the wavelet transform in the vicinity of the singularity at  $\tau$  is of the form

$$X(a, b) = (-1)^n a^{n-1/2} \psi^{(n)}\left(\frac{\tau - b}{a}\right), \quad (6.119)$$



**Figure 6.42:** A function with singularities of order 0, 1 and 2, and its wavelet transform.

where  $\psi^{(n)}$  is the  $n$ th primitive of  $\psi$ .

*Proof.* (Sketch) The proof follows the arguments developed above for  $n = 0, 1$ , and 2. Because  $\psi(t)$  has  $N$  zero moments, its primitives of order  $n \leq N$  are also compactly supported. For a singularity of order  $n$ , we apply integration by parts  $n$  times. Each primitive adds a scaling factor  $a$ ; this explains the factor  $a^{n-1/2}$  (the  $-1/2$  comes from the initial  $1/\sqrt{a}$  factor in the wavelet). After  $n$  integrations by parts,  $x(t)$  has been differentiated  $n$  times, is thus a Dirac delta function, and reproduces  $\psi^{(n)}$  at location  $\tau$ .

The key is that the singularities are not only precisely located at small scales, but the behavior of the continuous wavelet transform also indicates the singularity type. Figure 6.42 sketches the continuous wavelet transform of a function with a few singularities.

We considered the behavior around points of singularity, but what about “smooth” regions? Again, assume a wavelet of compact support and having  $N$  zero moments. Clearly, if the function  $x(t)$  is polynomial of degree  $N - 1$  or less, all inner products with the wavelet will be exactly zero due to the zero moment property. If the function  $x(t)$  is piecewise polynomial,<sup>56</sup> then the inner product will be zero once the wavelet is inside an interval, while boundaries will be detected according to the types of singularities that appear. We have calculated an example in Section 6.1 for Haar, which makes the above explicit, while also pointing out what happens when the wavelet does not have enough zero moments.

**Decay and Smoothness** Beyond polynomial and piecewise-polynomial functions, let us consider more general smooth functions. Among the many possible classes

<sup>56</sup>That is, the function is a polynomial over intervals  $(t_i, t_{i+1})$ , with singularities at the interval boundaries.

of smooth functions, we consider functions having  $m$  continuous derivatives, or the space  $C^m$ .

For the wavelet, we take a compactly supported wavelet  $\psi$  having  $N$  zero moments. Then, the  $N$ th primitive, denoted  $\psi^{(N)}$ , is compactly supported and

$$\int_{-\infty}^{\infty} \psi^{(N)}(t) dt = C \neq 0.$$

This follows since the Fourier transform of  $\psi$  has  $N$  zeros at the origin, and each integration removes one, leaving the Fourier transform  $\psi^{(N)}$  nonzero at the origin. For example, the primitive of the Haar wavelet is the triangle function in (6.33), with integral equal to  $1/2$ .

Consider the following scaled version of  $\psi^{(N)}$ , namely  $a^{-1}\psi^{(N)}(t/a)$ . This function has an integral equal to  $C$ , and it acts like a Dirac delta function as  $a \rightarrow 0$  in that, for a continuous function  $x(t)$ ,

$$\lim_{a \rightarrow 0} \int_{-\infty}^{\infty} \frac{1}{a} \psi^{(N)}\left(\frac{t-b}{a}\right) x(t) dt = Cx(b). \quad (6.120)$$

Again, the Haar wavelet with its primitive is a typical example, since a limit of scaled triangle functions is a classic way to obtain the Dirac delta function. We are now ready to prove the decay behavior of the continuous wavelet transform as  $a \rightarrow 0$ .

**THEOREM 6.17 (DECAY OF CONTINUOUS WAVELET TRANSFORM FOR  $x \in \mathbb{C}^N$ )**  
 Consider a compactly supported wavelet  $\psi$  with  $N$  zero moments,  $N \geq 1$ , and primitives  $\psi^{(1)}, \dots, \psi^{(N)}$ , where  $\int \psi^{(N)}(t) dt = C$ . Given a function  $x(t)$  having  $N$  continuous and bounded derivatives  $f^{(1)}, \dots, f^{(N)}$ , or  $f \in C^N$ , then the continuous wavelet transform of  $x(t)$  with respect to  $\psi$  behaves as

$$|X(a, b)| \leq C' a^{N+1/2} \quad (6.121)$$

for  $a \rightarrow 0$ .

*Proof.* (sketch) The proof closely follows the method of integration by parts as used in Theorem 6.16. That is, we take the  $N$ th derivative of  $x(t)$ ,  $f^{(N)}(t)$ , which is continuous and bounded by assumption. We also have the  $N$ th primitive of the wavelet,  $\psi^{(N)}(t)$ , which is of compact support and has a finite integral. After  $N$  integrations by parts, we have

$$\begin{aligned} X(a, b) &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) x(t) dt \\ &\stackrel{(a)}{=} (-1)^N a^N \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} \psi^{(N)}\left(\frac{t-b}{a}\right) f^{(N)}(t) dt \\ &\stackrel{(b)}{=} (-1)^N a^{N+1/2} \int_{-\infty}^{\infty} \frac{1}{a} \psi^{(N)}\left(\frac{t-b}{a}\right) f^{(N)}(t) dt, \end{aligned}$$



**Figure 6.43:** A function and its scalogram. (a) Function with various modes. (b) Scalogram with a Daubechies wavelet. (c) Scalogram with a symmetric wavelet. (d) Scalogram with a Morlet wavelet.

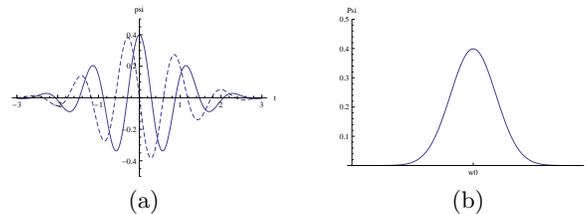
where (a)  $N$  steps of integration by parts contribute a factor  $a^N$ ; and (b) we normalize the  $N$ th primitive by  $1/a$  so that it has a constant integral and acts as a Dirac delta function as  $a \rightarrow 0$ . Therefore, for small  $a$ , the integral above tends towards  $Cf^{(N)}(b)$ , which is finite, and the decay of the continuous wavelet transform is thus of order  $a^{N+1/2}$ .

While we used a global smoothness, it is clear that it is sufficient for  $x(t)$  to be  $C^N$  in the vicinity of  $b$  for the decay to hold. The converse result, namely the necessary decay of the wavelet transform for  $x(t)$  to be in  $C^N$ , is a technical result which is more difficult to prove; it requires non-integer, Lipschitz, regularity. Note that if  $x(t)$  is smoother, that is, it has more than  $N$  continuous derivatives, the decay will still be of order  $a^{N+1/2}$  since we cannot apply more integration by parts steps. Also, the above result is valid for  $N \geq 1$  and thus cannot be applied to functions in  $C^0$ , but it can still be shown that the behavior is of order  $a^{1/2}$  as is to be expected.

**Scalograms** So far, we have only sketched continuous wavelet transforms, to point out general behavior like localization and other relevant properties. For “real” functions, a usual way of displaying the continuous wavelet transform is the density plot of the continuous wavelet transform magnitude  $|X(a, b)|$ . This is done in Figure 6.43 for a particular function and for 3 different wavelets, namely an orthogonal Daubechies wavelet, a symmetric biorthogonal wavelet, and the Morlet wavelet.

As can be seen, the scalograms with respect to symmetric wavelets (Figure 6.43 (c) and (d)) have no drift across scales, which helps identify singularities. The zooming property at small scales is quite evident from the scalogram.

**Remarks** The continuous-time continuous wavelet transform can be seen as a *mathematical microscope*. Indeed, it can zoom in, and describe the local behavior of a function very precisely. This pointwise characterization is a distinguishing feature of the continuous wavelet transform. The characterization itself is related to the wavelet being a local derivative operator. Indeed, a wavelet with  $N$  zero moments acts like an  $N$ th order derivative on the function analyzed by the wavelet transform, as was seen in the proofs of Theorems 6.16 and 6.17. Together with the



**Figure 6.44:** Morlet wavelet. (a) Time domain function, with real and imaginary parts in solid and dotted lines, respectively. (b) Magnitude spectrum of the Fourier transform.

fact that all scales are considered, this shows that the continuous wavelet transform is a multiscale differential operator.

*Compactly Supported Wavelets:* Throughout the discussion so far, we have often used the Haar wavelet (actually, its centered version) as the exemplary wavelet used in a continuous wavelet transform. The good news is that it is simple, short, and antisymmetric around the origin. The limitation is that in the frequency domain it has only a single zero at the origin; thus it can only characterize singularities up to order 1, and the decay of the continuous wavelet transform for smooth functions is limited.

Therefore, one can use higher order wavelets, like any of the Daubechies wavelets, or any biorthogonal wavelet. The key is the number of zeros at the origin. The attraction of biorthogonal wavelets is that there are symmetric or antisymmetric solutions. Thus, singularities are well localized along vertical lines, which is not the case for non-symmetric wavelets like the Daubechies wavelets. At the same time, there is no reason to use orthogonal or biorthogonal wavelets, since any functions satisfying the admissibility conditions (6.102) and having a sufficient number of zero moments will do. In the next subsection, scalograms will highlight differences between continuous wavelet transforms using different wavelets.

*Morlet Wavelet:* The classic, and historically first wavelet is a windowed complex exponential, first proposed by Jean Morlet. As a window, a Gaussian bell shape is used, and the complex exponential makes it a bandpass filter. Specifically, the wavelet is given by

$$\psi(t) = \frac{1}{\sqrt{2\pi}} e^{-j\omega_0 t} e^{-t^2/2}, \quad (6.122)$$

with

$$\omega_0 = \pi \sqrt{\frac{2}{\ln 2}},$$

where  $\omega_0$  is such that the second maximum of  $\Re(\psi(t))$  is half of the first one (at  $t = 0$ ), and the scale factor  $1/\sqrt{2\pi}$  makes the wavelet of unit norm. It is to be noted that  $\Psi(0) \neq 0$ , and as such the wavelet is not admissible. However,  $\Psi(0)$  is very small (of order  $10^{-7}$ ) and has numerically no consequence (and can be corrected by removing it from the wavelet). Figure 6.44 shows the Morlet wavelet in time and frequency domains.

It is interesting to note that the Morlet wavelet and the Gabor function are related. From (6.122) the Morlet wavelet at scale  $a \neq 0$  is

$$\psi_{a,0} = \frac{1}{\sqrt{2\pi a}} e^{-j\omega_0 t/a} e^{-(t/a)^2/2}$$

while, following (5.3) and (5.6), the Gabor function at  $\omega$  is

$$g_{\omega,0}(t) = \frac{1}{\sqrt{2\pi a}} e^{j\omega t/a} e^{-t^2/2a^2}$$

which are equal for  $\omega = \omega_0 = \pi\sqrt{2/\ln 2}$  and the same scale factor  $a$ . Thus, there is a frequency and a scale where the continuous wavelet transform (with a Morlet wavelet) and a local Fourier transform (with a Gabor function) coincide.

## 6.6 Computational Aspects

The multiresolution framework derived above is more than just of theoretical interest. In addition to allow constructing wavelets, like the spline and Meyer wavelets, it also has direct algorithmic implications as we show by deriving Mallat's algorithm for the computation of wavelet series.

### 6.6.1 Wavelet Series: Mallat's Algorithm

Given a wavelet basis  $\{\psi_{m,n}(t)\}_{m,n \in \mathbb{Z}}$ , any function  $x(t)$  can be written as

$$x(t) = \sum_{m \in \mathbb{Z}} \sum_{n \in \mathbb{Z}} \beta_k^{(\ell)} \psi_{m,n}(t).$$

where

$$\beta_k^{(\ell)} = \langle f, \psi_{m,n} \rangle. \quad (6.123)$$

Assume that only a finite-resolution version of  $x(t)$  can be acquired, in particular the projection of  $x(t)$  onto  $V^{(0)}$ , denoted  $f^{(0)}(t)$ . Because

$$V^{(0)} = \bigoplus_{m=1}^{\infty} W^{(\ell)},$$

we can write

$$f^{(0)}(t) = \sum_{m=1}^{\infty} \sum_{n \in \mathbb{Z}} \beta_k^{(\ell)} \psi_{m,n}(t). \quad (6.124)$$

Since  $f^{(0)}(t) \in V^{(0)}$ , we can also write

$$f^{(0)}(t) = \sum_{n \in \mathbb{Z}} \alpha_n^{(0)} \varphi(t-n), \quad (6.125)$$

where

$$\alpha_n^{(0)} = \langle x(t), \varphi(t-n) \rangle_t = \langle f, \varphi_{0,n} \rangle.$$

Given these two ways of expressing  $f^{(0)}(t)$ , how to go from one to the other? The answer, as to be expected, lies in the two-scale equation, and leads to a filter bank algorithm. Consider  $f^{(1)}(t)$ , the projection of  $f^{(0)}(t)$  onto  $V^{(1)}$ . This involves computing the inner products

$$\alpha_n^{(1)} = \langle f^{(0)}(t), \frac{1}{\sqrt{2}}\varphi(t/2 - n) \rangle_t, \quad n \in \mathbb{Z}. \quad (6.126)$$

From (6.48), we can write

$$\frac{1}{\sqrt{2}}\varphi(t/2 - n) = \sum_{k \in \mathbb{Z}} g_k \varphi(t - 2n - k). \quad (6.127)$$

Replacing this and (6.125) into (6.126) leads to

$$\begin{aligned} \alpha_n^{(1)} &= \sum_{k \in \mathbb{Z}} \sum_{\ell \in \mathbb{Z}} g_k \alpha_\ell^{(0)} \langle \varphi(t - 2n - k), \varphi(t - \ell) \rangle_t \\ &\stackrel{(a)}{=} \sum_{\ell \in \mathbb{Z}} g_{\ell - 2n} \alpha_\ell^{(0)} \stackrel{(b)}{=} (\tilde{g} * \alpha^{(0)})_{2n}, \end{aligned} \quad (6.128)$$

where (a) follows because the inner product is 0 unless  $\ell = 2n + k$ ; and (b) simply rewrites the sum as a convolution, with

$$\tilde{g}_n = g_{-n}.$$

The upshot is that the sequence  $\alpha_n^{(1)}$  is obtained from convolving  $\alpha_n^{(0)}$  with  $\tilde{g}$  (the time-reversed impulse response of  $g$ ) and downsampling by 2. The same development for the wavelet series coefficients

$$\beta_k^{(1)} = \langle f^{(0)}(t), \frac{1}{\sqrt{2}}\psi(t/2 - n) \rangle_t$$

yields

$$\beta_k^{(1)} = (\tilde{h} * \alpha^{(0)})_{2n}, \quad (6.129)$$

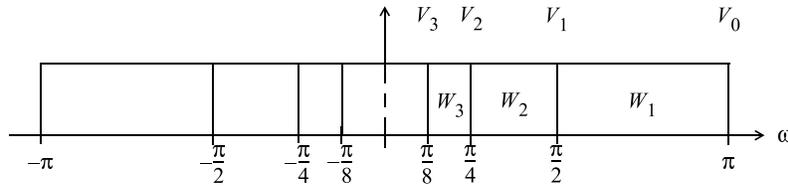
where

$$\tilde{h}_n = h_{-n}$$

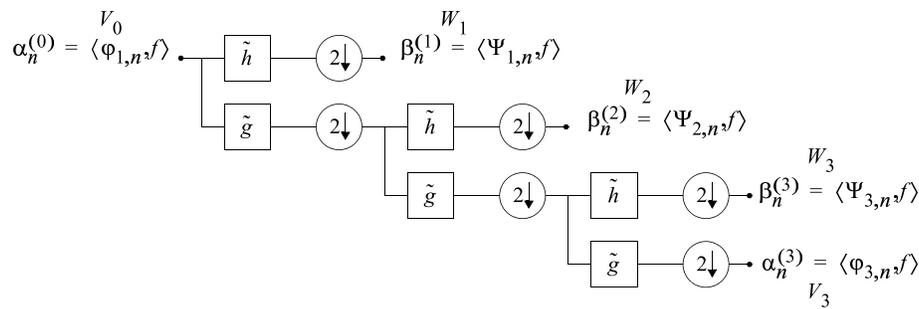
is the time-reversed impulse response of the highpass filter  $h$ . The argument just developed holds irrespectively of the scale at which we start, thus allowing to split a function  $f^{(\ell)}$  in  $V^{(\ell)}$  into its components  $f^{(m+1)}$  in  $V^{(m+1)}$  and  $d^{(m+1)}$  in  $W^{(m+1)}$ . This split is achieved by filtering and downsampling  $\alpha^{(\ell)}$  with  $\tilde{g}$  and  $\tilde{h}$ , respectively. Likewise, this process can be iterated  $k$  times, to go from  $V^{(\ell)}$  to  $V^{(m+k)}$ , while splitting off  $W^{(m+1)}$ ,  $W^{(m+2)}$ ,  $\dots$ ,  $W^{(m+k)}$ , or

$$V^{(\ell)} = W^{(m+1)} \oplus W^{(m+1)} \oplus \dots \oplus W^{(m+k)} \oplus V^{(m+k)}.$$

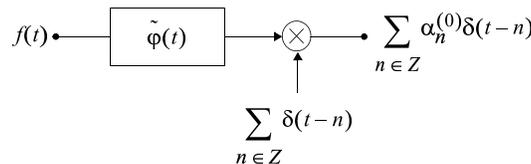
The key insight is of course that once we have an initial projection, for example,  $f^{(0)}(t)$  with expansion coefficients  $\alpha_n^{(0)}$ , then all the other expansion coefficients can be computed using discrete-time filtering. This is shown in Figure 6.46, where the sequence  $\alpha_n^{(0)}$ , corresponding to an initial projection of  $x(t)$  onto  $V^{(0)}$ , is decomposed into the expansion coefficients in  $W^{(1)}$ ,  $W^{(2)}$ ,  $W^{(3)}$  and  $V^{(3)}$ . This algorithm is known as Mallat's algorithm, since it is directly related to the multiresolution analysis of Mallat and Meyer.



**Figure 6.45:** Splitting of  $V^{(0)}$  into  $W^{(1)}, W^{(2)}, W^{(3)}$  and  $V^{(3)}$ , shown for a sinc multiresolution analysis.



**Figure 6.46:** Mallat's algorithm. From the initial sequence  $\alpha_n^{(0)}$ , all of the wavelet series coefficients are computed through a discrete filter bank algorithm.



**Figure 6.47:** Initialization of Mallat's algorithm. The function  $x(t)$  is convolved with  $\tilde{\varphi}(t) = \varphi(-t)$  and sampled at  $t = n$ .

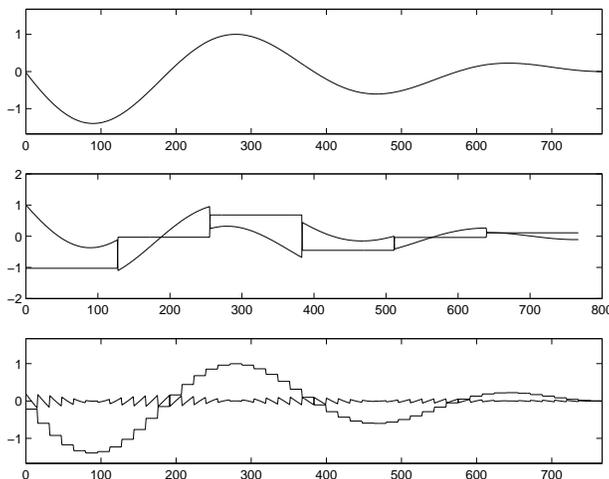
**Initialization** How do we initialize Mallat's algorithm, that is, compute the initial sequence  $\alpha_n^{(0)}$ ? There is no escape from computing the inner products

$$\alpha_n^{(0)} = \langle x(t), \varphi(t - n) \rangle_t = (\tilde{\varphi} * f)|_{t=n},$$

where  $\tilde{\varphi}(t) = \varphi(-t)$ . This is shown in Figure 6.47.

The simplification obtained through this algorithm is the following. Computing inner products involves continuous-time filtering and sampling, which is difficult. Instead of having to compute such inner products at all scales as in (6.123), only a single scale has to be computed, namely the one leading to  $\alpha_n^{(0)}$ . All the subsequent inner products are obtained from that sequence, using only discrete-time processing.

The question is: How well does  $f^{(0)}(t)$  approximate the function  $x(t)$ ? The



**Figure 6.48:** Initial approximation in Mallat's algorithm. (a) Function  $x(t)$ . (b) Approximation  $f^{(0)}(t)$  with Haar scaling function in  $V^{(0)}$  and error  $e^{(0)}(t) = x(t) - f^{(0)}(t)$ . (c) Same but in  $V^{(-3)}$ , or  $f^{(-3)}(t)$  and  $e^{(-3)}(t) = x(t) - f^{(-3)}(t)$ .

key is that if the error  $\|f^{(0)} - f\|$  is too large, we can go to finer resolutions  $f^{(\ell)}$ ,  $m < 0$ , until  $\|f^{(\ell)} - f\|$  is small enough. Because of completeness, we know that there is an  $m$  such that the initial approximation error can be made as small as we like.

In Figure 6.48, we show two different initial approximations and the resulting errors,

$$e^{(\ell)}(t) = x(t) - f^{(\ell)}(t).$$

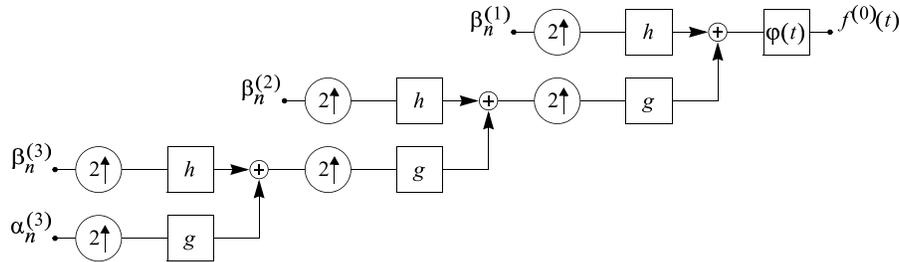
Clearly, the smoother the function, the faster the decay of  $\|e^{(\ell)}\|$  as  $m \rightarrow -\infty$ . Exercise ?? explores this further.

**The Synthesis Problem** We have considered the analysis problem, or given a function, how to obtain its wavelet coefficients. Conversely, we can also consider the synthesis problem. That is, given a wavelet series representation as in (6.124), how to synthesize  $f^{(0)}(t)$ . One way is to effectively add wavelets at different scales and shifts, with the appropriate weights (6.123).

The other way is to synthesize  $f^{(0)}(t)$  as in (6.125), which now involves only linear combinations of a single function  $\varphi(t)$  and its integer shifts. To make matters specific, assume we want to reconstruct  $f^{(0)} \in V^{(0)}$  from  $f^{(1)}(t) \in V^{(1)}$  and  $d^{(1)}(t) \in W^{(1)}$ . There are two ways to write  $f^{(0)}(t)$ , namely

$$f^{(0)}(t) = \sum_{n \in \mathbb{Z}} \alpha_n^{(0)} \varphi(t - n) \quad (6.130)$$

$$= \frac{1}{\sqrt{2}} \sum_{n \in \mathbb{Z}} \alpha_n^{(1)} \varphi(t/2 - n) + \frac{1}{\sqrt{2}} \sum_{n \in \mathbb{Z}} \beta_k^{(1)} \psi(t/2 - n), \quad (6.131)$$



**Figure 6.49:** Synthesis of  $f^{(0)}(t)$  using Mallat's algorithm. The wavelet and scaling coefficients are fed through a DWT synthesis, generating the sequence  $\alpha_n^{(0)}$ . A final continuous-time processing implementing (6.130) leads to  $f^{(0)}(t)$ .

where the latter is the sum of  $f^{(1)}(t)$  and  $d^{(1)}(t)$ . Now,

$$\alpha_\ell^{(0)} = \langle f^{(0)}(t), \varphi(t - \ell) \rangle_t.$$

Using the two-scale equation (6.127) and its equivalent for  $\psi(t/2 - n)$ ,

$$\frac{1}{\sqrt{2}}\psi(t/2 - n) = \sum_{k \in \mathbb{Z}} h_k \varphi(t - 2n - k),$$

we can write

$$\begin{aligned} \alpha_\ell^{(0)} &= \langle f^{(1)}(t), \varphi(t - \ell) \rangle_t + \langle d^{(1)}(t), \varphi(t - \ell) \rangle_t \\ &\stackrel{(a)}{=} \sum_{n \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} \alpha_n^{(1)} g_k \langle \varphi(t - \ell), \varphi(t - 2n - k) \rangle_t \\ &\quad + \sum_{n \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} \beta_k^{(1)} h_k \langle \varphi(t - \ell), \varphi(t - 2n - k) \rangle_t \\ &\stackrel{(b)}{=} \sum_{n \in \mathbb{Z}} \alpha_n^{(1)} g_{\ell - 2n} + \sum_{n \in \mathbb{Z}} \beta_k^{(1)} h_{\ell - 2n} \end{aligned} \quad (6.132)$$

where (a) follows from (6.131) using the two-scale equation; and (b) is obtained from the orthogonality of the  $\varphi$ s, unless  $k = \ell - 2n$ . The obtained expression for  $\alpha_\ell^{(0)}$  indicates that the two sequences  $\alpha_n^{(1)}$  and  $\beta_k^{(1)}$  are upsampled by 2 before being filtered by  $g$  and  $h$ , respectively. In other words, a two-channel synthesis filter bank produces the coefficients for synthesizing  $f^{(0)}(t)$  according to (6.130). The argument above can be extended to any number of scales and leads to the synthesis version of Mallat's algorithm, shown in Figure 6.49.

Again, the simplification arises since instead of having to use continuous-time wavelets and scaling functions at many scales, only a single continuous-time prototype function is needed. This prototype function is  $\varphi(t)$  and its shifts, or the basis for  $V^{(0)}$ . Because of the inclusion of all the coarser spaces in  $V^{(0)}$ , the result is intuitive, nonetheless it is remarkable that the multiresolution framework leads naturally to a discrete-time filter bank algorithm.

## 6.6.2 Wavelet Frames

### Chapter at a Glance

	Lowpass & scaling function	Highpass & wavelet
Filter	$G(z) = \left(\frac{1+z^{-1}}{2}\right)^N R(z)$	$H(z) = z^{-L+1} \left(\frac{1-z}{2}\right)^N R(-z^{-1})$
Function	$\Phi(\omega) = \prod_{i=1}^{\infty} \frac{1}{\sqrt{2}} G(e^{j\omega/2^i})$	$\Psi(\omega) = \frac{1}{\sqrt{2}} H(e^{j\omega/2}) \prod_{i=2}^{\infty} \frac{1}{\sqrt{2}} G(e^{j\omega/2^i})$
Two-scale equation	$\Phi(\omega) = 2^{-1/2} G(e^{j\omega/2}) \Phi(\omega/2)$ $\varphi(t) = \sqrt{2} \sum_n g_n \varphi(2t - n)$	$\Psi(\omega) = 2^{-1/2} H(e^{j\omega/2}) \Phi(\omega/2)$ $\psi(t) = \sqrt{2} \sum_n h_n \varphi(2t - n)$
Orthogonality	$\langle \varphi(t), \varphi(t - n) \rangle_t = \delta_n$ $\langle \varphi(t), \psi(t - n) \rangle_t = 0$	$\langle \psi(t), \psi(t - n) \rangle_t = \delta_n$ $\langle \psi(t), 2^{-m/2} \psi(2^{-m}t - n) \rangle_t = \delta_n$
Smoothness	Can be tested, increases with $N$	Same as for $\varphi(t)$
Moments	Polynomials of degree $N - 1$ are in $\text{span}(\{\varphi(t - n)\}_{n \in \mathbb{Z}})$	Wavelets has $N$ zero moments
Size and support	$\text{support}(g) = \{0, \dots, L - 1\}$ $\text{support}(\varphi) = [0, L - 1]$	$\text{support}(h) = \{0, \dots, L - 1\}$ $\text{support}(\psi) = [0, L - 1]$

**Table 6.1:** Major properties of scaling function and wavelet based on an iterated filter bank with an orthonormal lowpass filter having  $N$  zeros at  $z = -1$  or  $\omega = \pi$ .

### Historical Remarks

TBD

### Further Reading

**Books and Textbooks** Daubechies [31].

**Results on Wavelets** For the proof of completeness of Theorem 6.6, see [29, 31].



**Chapter 7****Approximation,  
Estimation, and  
Compression****Contents**

---

<b>7.1</b>	<b>Introduction</b>	<b>262</b>
<b>7.2</b>	<b>Abstract Models and Approximation</b>	<b>262</b>
<b>7.3</b>	<b>Empirical Models</b>	<b>262</b>
<b>7.4</b>	<b>Estimation and Denoising</b>	<b>262</b>
<b>7.5</b>	<b>Compression</b>	<b>262</b>
<b>7.6</b>	<b>Inverse Problems</b>	<b>262</b>
	<b>Chapter at a Glance</b>	<b>262</b>
	<b>Historical Remarks</b>	<b>262</b>
	<b>Further Reading</b>	<b>262</b>
<b>7.A</b>	<b>Elements of Source Coding</b>	<b>263</b>

---

## 7.1 Introduction

### Chapter Outline

## 7.2 Abstract Models and Approximation

### 7.2.1 Local Fourier and Wavelet Approximations of Piecewise Smooth Functions

### 7.2.2 Wide-Sense Stationary Gaussian Processes

### 7.2.3 Poisson Processes

## 7.3 Empirical Models

### 7.3.1 $\ell^p$ Models

### 7.3.2 Statistical Models

## 7.4 Estimation and Denoising

### 7.4.1 Connections to Approximation

### 7.4.2 Wavelet Thresholding and Variants

### 7.4.3 Frames

## 7.5 Compression

### 7.5.1 Audio Compression

### 7.5.2 Image Compression

## 7.6 Inverse Problems

### 7.6.1 Deconvolution

### 7.6.2 Compressed Sensing

### Chapter at a Glance

TBD

### Historical Remarks

TBD

### Further Reading

TBD

---

**Appendix****7.A Elements of Source Coding****7.A.1 Entropy Coding****7.A.2 Quantization****7.A.3 Transform Coding**



# Bibliography

- [1] T. Aach. New criteria for shift variance and wide-sense cyclostationarity in multirate filter banks. In *Proc. IEEE Int. Workshop on Spectral Methods and Multirate Signal Proc.*, pages 7–13, Florence, Italy, 2006.
- [2] T. Aach. Comparative analysis of shift variance and cyclostationarity in multirate filter banks. *IEEE Trans. Circ. and Syst.*, 54(5):1077–1087, May 2007.
- [3] N. I. Akhiezer and I. M. Glazman. *Theory of Linear Operators in Hilbert Spaces*, volume 1. Frederick Ungar Publishing, 1966.
- [4] S. Akkarakaran and P. P. Vaidyanathan. Bifrequency and bispectrum maps: A new look at multirate systems with stochastic inputs. *IEEE Trans. Signal Proc.*, 48(3):723–736, March 2000.
- [5] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Trans. Image Proc.*, 1(2):205–220, April 1992.
- [6] P. Auscher. *Wavelets: Mathematics and Applications*, chapter Remarks on the local Fourier bases, pages 203–218. CRC Press, 1994.
- [7] M. G. Bellanger and J. L. Daguët. TDM-FDM transmultiplexer: Digital polyphase and FFT. *IEEE Trans. Commun.*, 22(9):1199–1204, September 1974.
- [8] J. J. Benedetto and M. C. Fickus. Finite normalized tight frames. *Adv. Comp. Math., sp. iss. Frames*, 18:357–385, 2003.
- [9] B. G. Bodmann, P. G. Casazza, and G. Kutyniok. A quantitative notion of redundancy for finite frames. In *Journ. Appl. and Comput. Harmonic Analysis*, 2010.
- [10] H. Bölcskei and F. Hlawatsch. *Gabor Analysis and Algorithms: Theory and Applications*, chapter Oversampled modulated filter banks, pages 295–322. Birkhäuser, Boston, MA, 1998.
- [11] H. Bölcskei and F. Hlawatsch. Oversampled cosine modulated filter banks with perfect reconstruction. *IEEE Trans. Circ. and Syst. II: Analog and Digital Signal Proc.*, 45(8):1057–1071, August 1998.

- 
- [12] H. Bölcskei, F. Hlawatsch, and H. G. Feichtinger. Frame-theoretic analysis of oversampled filter banks. *IEEE Trans. Signal Proc.*, 46(12):3256–3269, December 1998.
- [13] A. P. Bradley. Shift invariance in the discrete wavelet transform. In *Proc. Digit. Image Comp.*, December 2003.
- [14] A. A. M. L. Bruckens and A. W. M. van den Enden. New networks for perfect inversion and perfect reconstruction. *IEEE Journ. Sel. Areas in Commun.*, 10(1), September 1992.
- [15] P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Trans. Commun.*, 31(4):532–540, April 1983.
- [16] E. J. Candès. Curvelet Web Site. <http://www.curvelet.org/papers.html>.
- [17] E. J. Candès, L. Demanet, D. L. Donoho, and L. Ying. Fast discrete curvelet transforms. *Preprint*, 2005.
- [18] P. G. Casazza. The art of frame theory. *Taiwanese Journ. Math.*, 4(2):129–202, 2000.
- [19] P. G. Casazza and J. Kovačević. Equal-norm tight frames with erasures. *Adv. Comp. Math., sp. iss. Frames*, 18:387–430, 2002.
- [20] P. G. Casazza and G. Kutyniok. A generalization of Gram-Schmidt orthogonalization generating all Parseval frames. *Adv. Comp. Math.*, 27:65–78, 2007. Preprint.
- [21] A. Chebira and J. Kovačević. Lapped tight frame transforms. In *Proc. IEEE Int. Conf. Acoust., Speech and Signal Proc.*, volume III, pages 857–860, Honolulu, HI, April 2007.
- [22] O. Christensen. *An Introduction to Frames and Riesz Bases*. Birkhäuser, 2002.
- [23] A. Cohen, I. Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Commun. Pure and Appl. Math.*, 45:485–560, 1992.
- [24] R. R. Coifman, Y. Meyer, S. Quake, and M. V. Wickerhauser. Signal processing and compression with wavelet packets. Technical report, Yale Univ., 1991.
- [25] A. L. Cunha, J. Zhou, and M. N. Do. The nonsubsampling contourlet transform: Theory, design, and applications. *IEEE Trans. Image Proc.*, 15(10):3089–3101, October 2006.
- [26] Z. Cvetković. Oversampled modulated filter banks and tight Gabor frames in  $\ell^2(\mathbb{Z})$ . In *Proc. IEEE Int. Conf. Acoust., Speech and Signal Proc.*, pages 1456–1459, Detroit, MI, May 1995.

- 
- [27] Z. Cvetković and M. Vetterli. Oversampled filter banks. *IEEE Trans. Signal Proc.*, 46(5):1245–1255, May 1998.
- [28] Z. Cvetković and M. Vetterli. Tight Weyl-Heisenberg frames in  $\ell^2(\mathbb{Z})$ . *IEEE Trans. Signal Proc.*, 46(5):1256–1259, May 1998.
- [29] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Commun. Pure and Appl. Math.*, 41:909–996, November 1988.
- [30] I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Trans. Inform. Th.*, 36(5):961–1005, September 1990.
- [31] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, Philadelphia, PA, 1992.
- [32] I. Daubechies, A. Grossman, and Y. Meyer. Painless nonorthogonal expansions. *Journ. Math. Phys.*, 27:1271–1283, November 1986.
- [33] M. N. Do and M. Vetterli. The finite ridgelet transform for image representation. *IEEE Trans. Image Proc.*, 12(1):16–28, January 2003.
- [34] M. N. Do and M. Vetterli. The contourlet transform: An efficient directional multiresolution image representation. *IEEE Trans. Image Proc.*, 14(12):2091–2106, December 2005.
- [35] E. Dubois. The sampling and reconstruction of time-varying imagery with application in video systems. *Proc. IEEE*, 73(4):502–522, April 1985.
- [36] R. J. Duffin and A. C. Schaeffer. A class of nonharmonic Fourier series. *Trans. Amer. Math. Soc.*, 72:341–366, 1952.
- [37] Y. Eldar and H. Bölcskei. Geometrically uniform frames. *IEEE Trans. Inform. Th.*, 49(4):993–1006, April 2003.
- [38] D. Esteban and C. Galand. Applications of quadrature mirror filters to split band voice coding schemes. In *Proc. IEEE Int. Conf. Acoust., Speech and Signal Proc.*, pages 191–195, 1995.
- [39] H. G. Feichtinger and T. Strohmer, editors. *Gabor Analysis and Algorithms: Theory and Applications*. Birkhäuser, Boston, MA, 1998.
- [40] D. Gabor. Theory of communication. *Journ. IEE*, 93:429–457, 1946.
- [41] V. K. Goyal, M. Vetterli, and N. T. Thao. Quantized overcomplete expansions in  $\mathbb{R}^N$ : Analysis, synthesis, and algorithms. *IEEE Trans. Inform. Th.*, 44(1):16–31, January 1998.
- [42] D. Han and D. R. Larson. *Frames, bases and group representations*. Number 697 in Memoirs AMS. AMS Press, Providence, RI, 2000.
- [43] C. Heil and D. Walnut. Continuous and discrete wavelet transforms. *SIAM Review*, 31:628–666, 1989.

- 
- [44] C. Herley and M. Vetterli. Biorthogonal bases of symmetric compactly supported wavelets. In M. Farge and et al., editors, *Proc. Wavelets, Fractals and Fourier Transforms*. Oxford Univ. Press, 1991.
- [45] C. Herley and M. Vetterli. Wavelets and recursive filter banks. *IEEE Trans. Signal Proc.*, August 1993.
- [46] C. Herley and M. Vetterli. Orthogonal time-varying filter banks and wavelet packets. *IEEE Trans. Signal Proc.*, 42(10):2650–2663, October 1994.
- [47] O. Herrmann. On the approximation problem in nonrecursive digital filter design. *IEEE Trans. Circ. Theory*, 18:411–413, 1971.
- [48] R. B. Holmes and V. I. Paulsen. Optimal frames for erasures. *Linear Algebra and Its Appl.*, 377:31–51, 2004.
- [49] J. D. Johnston. A filter family designed for use in Quadrature Mirror Filter Banks. In *Proc. IEEE Int. Conf. Acoust., Speech and Signal Proc.*, pages 291–294, Denver, CO, 1980.
- [50] G. Karlsson and M. Vetterli. Theory of two - dimensional multirate filter banks. *IEEE Trans. Acoust., Speech, and Signal Proc.*, 38(6):925–937, June 1990.
- [51] N. G. Kingsbury. The dual-tree complex wavelet transform: A new efficient tool for image restoration and enhancement. In *Proc. Eur. Signal Proc. Conf.*, pages 319–322, 1998.
- [52] N. G. Kingsbury. Image processing with complex wavelets. *Phil. Trans. Royal Soc. London A*, September 1999.
- [53] N. G. Kingsbury. Complex wavelets for shift invariant analysis and filtering of signals. *Journ. Appl. and Comput. Harmonic Analysis*, 10(3):234–253, May 2001.
- [54] J. Kovačević and A. Chebira. *An Introduction to Frames*. Foundations and Trends in Signal Proc. Now Publishers, 2008.
- [55] J. Kovačević and M. Vetterli. Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for  $\mathcal{R}^n$ . *IEEE Trans. Inform. Th., sp. iss. Wavelet Transforms and Multiresolution Signal Analysis*, 38(2):533–555, March 1992. Chosen for inclusion in Fundamental Papers in Wavelet Theory.
- [56] D. Labate, W-Q. Lim, G. Kutyniok, and G. Weiss. Sparse multidimensional representation using shearlets. In *Proc. SPIE Conf. Wavelet Appl. in Signal and Image Proc.*, pages 254–262, Bellingham, WA, 2005.
- [57] D. J. LeGall and A. Tabatabai. Subband coding of digital images using symmetric short kernel filters and arithmetic coding techniques. In *Proc. IEEE Int. Conf. Acoust., Speech and Signal Proc.*, pages 761–765, New York, NY, 1988.

- 
- [58] Y. Lu and M. N. Do. The finer directional wavelet transform. In *Proc. IEEE Int. Conf. Acoust., Speech and Signal Proc.*, volume 4, pages 573–576, Philadelphia, PA, March 2005.
- [59] S. Mallat. Multiresolution approximations and wavelet orthonormal bases of  $L^2(\mathbb{R})$ . *Trans. Amer. Math. Soc.*, 315:69–87, September 1989.
- [60] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, second edition, 1999.
- [61] H. S. Malvar. *Signal Processing with Lapped Transforms*. Artech House, Norwood, MA, 1992.
- [62] T. G. Marshall. U-L block-triangular matrix and ladder realizations of sub-band codes. In *Proc. IEEE Int. Conf. Acoust., Speech and Signal Proc.*, volume 3, pages 177–180, 1993.
- [63] J. L. Massey and T. Mittelholzer. *Sequences II: Methods in Communication, Security and Computer Sciences*, chapter Welch’s bound and sequence sets for code-division multiple-access systems, pages 63–78. Springer-Verlag, New York, NY, 1993.
- [64] F. Mintzer. Filters for distortion-free two-band multirate filter banks. *IEEE Trans. Acoust., Speech, and Signal Proc.*, 33(3):626–630, June 1985.
- [65] I. Newton. *Opticks or A Treatise of the Reflections, Refractions, Inflections and Colours of Light*. Royal Society, 1703.
- [66] H. J. Nussbaumer. Complex quadrature mirror filters. In *Proc. IEEE Int. Conf. Acoust., Speech and Signal Proc.*, volume II, pages 221–223, Boston, MA, 1983.
- [67] B. Porat. *Digital Processing of Random Signals*. Prentice Hall, Englewood Cliffs, NJ, 1994.
- [68] B. Porat. *A Course in Digital Signal Processing*. John Wiley & Sons, New York, NY, 1996.
- [69] K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *IEEE Trans. Image Proc.*, 2(2):160–175, April 1993.
- [70] T. A. Ramstad. IIR filter bank for subband coding of images. In *Proc. IEEE Int. Symp. Circ. and Syst.*, pages 827–830, 1988.
- [71] J. M. Renes, R. Blume-Kohout, A. J. Scot, and C. M. Caves. Symmetric informationally complete quantum measurements. *Journ. Math. Phys.*, 45(6):2171–2180, 2004.
- [72] O. Rioul. A discrete-time multiresolution theory. *IEEE Trans. Signal Proc.*, 41(8):2591–2606, August 1993.

- [73] O. Rioul and P. Duhamel. Fast algorithms for discrete and continuous wavelet transforms. *IEEE Trans. Inform. Th., sp. iss. Wavelet Transforms and Multiresolution Signal Analysis*, 38(2):569–586, March 1992.
- [74] N. Saito and R. R. Coifman. Local discriminant bases. In *Proc. SPIE Conf. Vis. Commun. and Image Proc.*, pages 2–14, 1994.
- [75] N. Saito and R. R. Coifman. Local discriminant bases and their applications. *Journ. Math. Imag. Vis.*, 5:337–358, 1995.
- [76] A. Sandryhaila, A. Chebira, C. Milo, J. Kovačević, and M. Püschel. Systematic construction of real lapped tight frame transforms. *IEEE Trans. Signal Proc.*, 58(5):2256–2567, May 2010.
- [77] I. W. Selesnick. *Wavelets in Signal and Image Analysis*, chapter The double density DWT. Kluwer Academic Publishers, 2001.
- [78] I. W. Selesnick. The double-density dual-tree DWT. *IEEE Trans. Signal Proc.*, 52(5):1304–1314, May 2004.
- [79] I. Shah and T. A. C. M. Kalker. On ladder structures and linear phase conditions for bi-orthogonal filter banks. In *Proc. IEEE Int. Conf. Acoust., Speech and Signal Proc.*, pages 181–184, 1994.
- [80] E. P. Simoncelli. Simoncelli Web Site. <http://www.cns.nyu.edu/~eero/steerpyr/>.
- [81] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger. Shiftable multiscale transforms. *IEEE Trans. Inform. Th., sp. iss. Wavelet Transforms and Multiresolution Signal Analysis*, 38(2):587–607, March 1992.
- [82] M. J. T. Smith. *IIR analysis/synthesis systems*, chapter in *Subband Image Coding*. Kluwer Academic Press, Boston, MA, 1991. J. W. Woods ed.
- [83] M. J. T. Smith and T. P. Barnwell III. Exact reconstruction for tree-structured subband coders. *IEEE Trans. Acoust., Speech, and Signal Proc.*, 34(3):431–441, June 1986.
- [84] A. K. Soman, P. P. Vaidyanathan, and T. Q. Nguyen. Linear phase paraunitary filter banks: Theory, factorizations and applications. *IEEE Trans. Signal Proc., sp. iss. Wavelets and Signal Proc.*, 41(12), December 1993.
- [85] J.-L. Starck, M. Elad, and D. L. Donoho. Redundant multiscale transforms and their application for morphological component separation. *Advances in Imaging and Electron Physics*, 132, 2004.
- [86] P. Stoica and R. Moses. *Introduction to Spectral Analysis*. Prentice Hall, Englewood Cliffs, NJ, 2000.
- [87] G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley Cambridge Press, Boston, MA, 1996.

- 
- [88] T. Strohmer. *Modern Sampling Theory: Mathematics and Applications*, chapter Finite and infinite-dimensional models for oversampled filter banks, pages 297–320. Birkhäuser, Boston, MA, 2000.
- [89] T. Strohmer and R. Heath. Grassmannian frames with applications to coding and communications. *Journ. Appl. and Comput. Harmonic Analysis*, 14(3):257–175, 2003.
- [90] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Journ. Appl. and Comput. Harmonic Analysis*, 1996.
- [91] C. W. Therrien. Issues in multirate statistical signal processing. In *Proc. Asilomar Conf. Signal, Syst. and Comp.*, volume 1, pages 573–576, Pacific Grove, CA, 2001.
- [92] J. A. Tropp, I. S. Dhillon, R. W. Heath, Jr., and T. Strohmer. Designing structured tight frames via an alternating projection method. *IEEE Trans. Inform. Th.*, 51(1):188–209, January 2005.
- [93] M. K. Tsatsanis and G. B. Giannakis. Principal component filter banks for optimal multiresolution analysis. *IEEE Trans. Signal Proc.*, 43(8):1766–1777, August 1995.
- [94] M. Unser. An extension of the Karhunen-Loève transform for wavelets and perfect reconstruction filterbanks. In *Proc. SPIE Conf. Wavelet Appl. in Signal and Image Proc.*, volume 2034, pages 45–56, San Diego, CA, July 1993.
- [95] M. Unser. Wavelets, filterbanks, and the Karhunen-Loève transform. In *Proc. Eur. Conf. Signal Proc.*, volume III, pages 1737–1740, Rhodes, Greece, 1998.
- [96] P. P. Vaidyanathan. Quadrature mirror filter banks, M-band extensions and perfect reconstruction techniques. *IEEE Acoust., Speech, and Signal Proc. Mag.*, 4(3):4–20, July 1987.
- [97] P. P. Vaidyanathan. Theory and design of M-channel maximally decimated quadrature mirror filters with arbitrary M, having the perfect reconstruction property. *IEEE Trans. Acoust., Speech, and Signal Proc.*, 35(4):476–492, April 1987.
- [98] P. P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice Hall, Englewood Cliffs, NJ, 1992.
- [99] P. P. Vaidyanathan and S. Akkarakaran. A review of the theory and applications of principal component filter banks. *Journ. Appl. and Comput. Harmonic Analysis*, 10(3):254–289, May 2001.
- [100] P. P. Vaidyanathan and P.-Q. Hoang. Lattice structures for optimal design and robust implementation of two-channel perfect reconstruction filter banks. *IEEE Trans. Acoust., Speech, and Signal Proc.*, 36(1):81–94, January 1988.

- 
- [101] P. P. Vaidyanathan and Z. Doğanata. The role of lossless systems in modern digital signal processing: A tutorial. *IEEE Trans. Educ.*, 32(3):181–197, August 1989.
- [102] R. Vale and S. Waldron. Tight frames and their symmetries. *Const. Approx.*, 21:83–112, 2005.
- [103] M. Vetterli. Filter banks allowing perfect reconstruction. *Signal Proc.*, 10(3):219–244, April 1986.
- [104] M. Vetterli. A theory of multirate filter banks. *IEEE Trans. Acoust., Speech, and Signal Proc.*, 35(3):356–372, March 1987.
- [105] M. Vetterli and C. Herley. Wavelets and filter banks: Theory and design. *IEEE Trans. Signal Proc.*, 40(9):2207–2232, September 1992.
- [106] M. Vetterli and J. Kovačević. *Wavelets and Subband Coding*. Signal Processing. Prentice Hall, Englewood Cliffs, NJ, 1995. <http://waveletsandsubbandcoding.org/>.
- [107] M. Vetterli, J. Kovačević, and V. K. Goyal. *Foundations of Signal Processing*. Cambridge Univ. Press, 2013.
- [108] M. Vetterli and D. J. LeGall. Perfect reconstruction FIR filter banks: Some properties and factorizations. *IEEE Trans. Acoust., Speech, and Signal Proc.*, 37(7):1057–1071, July 1989.
- [109] E. Viscito and J. P. Allebach. The analysis and design of multidimensional FIR perfect reconstruction filter banks for arbitrary sampling lattices. *IEEE Trans. Circ. and Syst.*, 38(1):29–42, January 1991.
- [110] P. Viswanath and V. Anantharam. Optimal sequences and sum capacity of synchronous CDMA systems. *IEEE Trans. Inform. Th.*, 45(6):1984–1991, September 1999.
- [111] S. Waldron. Generalised Welch bound equality sequences are tight frames. *IEEE Trans. Inform. Th.*, 49(9):2307–2309, September 2003.
- [112] M. V. Wickerhauser. INRIA lectures on wavelet packet algorithms. Technical report, Yale Univ., March 1991.
- [113] M. V. Wickerhauser. Lectures on wavelet packet algorithms, 1992.